T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

METİN SINIFLAMA İÇİN YENİ BİR ÖZELLİK ÇIKARIM YÖNTEMİ

GÖKSEL BİRİCİK

DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI

DANIŞMAN
PROF. DR. A. COŞKUN SÖNMEZ

İSTANBUL, 2011

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

METİN İŞLEME ALANI İÇİN YENİ BİR ÖZELLİK ÇIKARIM YÖNTEMİ

Göksel BİRİCİK tarafından hazırlanan tez çalışması 12.08.2011 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Prof. Dr. A. Coşkun SÖNMEZ
Yıldız Teknik Üniversitesi _____

Jüri Üyeleri

Prof. Dr. Selim AKYOKUŞ
Doğuş Üniversitesi _____

Doç. Dr. Şule GÜNDÜZ ÖĞÜDÜCÜ
İstanbul Teknik Üniversitesi _____

Yrd. Doç. Dr. Banu DİRİ
Yıldız Teknik Üniversitesi _____

Yrd. Doç. Dr. Songül ALBAYRAK
Yıldız Teknik Üniversitesi _____

ÖNSÖZ

Günümüzdeki bilgi üretim hızının geçmişle kıyaslanamayacak kadar fazla olduğu gerçeği, gelecekteki ivmenin bugün hayal bile edemeyeceğimiz boyutlara ulaşacağını göstermektedir. Tüm bu üretilen bilgi ise erişilemedikçe ve kullanılamadıkça değersiz bir yığın olmaktan öteye gidemez. Bu sebeple bilgiye erişim problemini çözmek, gelecekte bugün olduğundan çok daha önemli bir problem olarak karşımıza çıkacaktır.

Bilgi; metin, görüntü, ses, video veya çoklu ortam gibi farklı tür ve formlarda olabilir. Metin şeklindeki bilgiye erişimi kolaylaştırmak ve düzenlemek için metnin özelliklerine göre sınıflandırma işlemi ile kategorizasyon, kaçınılmaz ve doğal bir çözüm olarak uygulanmaktadır. Ancak burada karşımıza çıkan sorun, içeriğe göre bilgiye erişimi sağlamak üzere metnin hangi özelliklerle temsil edileceğidir.

Metin tipindeki bilgi, yapısında çok sayıda özellik barındırmaktadır. Bu sebeple metin tipindeki verileri kategorizasyon yöntemleri ile işlemek zordur. Bu problemin çözümü içinse kategorize edilecek verinin özellik boyutlarını indirgeme yöntemleri ortaya atılmıştır. Bu yöntemlerde amaç, başarımı en az düşürecek ve efektif çalışmayı arttıracak şekilde, özelliklerin sayısının azaltılması veya az sayıda yeni özellikle verinin yeniden tanımlanmasıdır.

Biz, çalışmamızda metin sınıflandırma işlemlerinde başarımı arttırmak üzere özellik sayısını indirgeyecek yeni bir yöntem tasarımı gerçekleştirdik. Yöntemimiz ile elde ettiğimiz, metni temsil eden özelliklere de soyut özellikler adını verdik. Gerçekleştirdiğimiz deney ve testlerde diğer yöntemlere kıyasla daha yüksek başarım sağladığını gördüğümüz yöntemimizin diğer çalışmalarda ve farklı disiplinlerdeki veriler üzerinde de kullanılarak literatürde yer edinmesini umut ediyoruz.

Bu çalışmanın ortaya çıkmasında bilgi, birikim, görüş ve önerileriyle destek olup yol gösteren, yardımlarını esirgemeyen değerli hocam Dr. Banu DİRİ'ye ve deneyimleriyle problemleri sorgulayıcı ve yorumlayıcı bakış açısıyla çözmemi sağlayan danışmanım Prof. Dr. A. Coşkun SÖNMEZ'e teşekkürü borç biliyorum.

Ayrıca destekleri için başta sevgili dostum Dr. Ö. Özgür BOZKURT olmak üzere çalışma arkadaşlarıma teşekkür etmeden geçemem.

Son olarak, bilgi, ilgi ve desteğini benden hiçbir zaman esirgemeyen, beni her zaman motive eden ve yüreklendiren sevgili eşim Ekin Su ve tüm aileme sonsuz teşekkürlerimi sunuyorum.

Ağustos, 2011

Göksel Biricik

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ ..... Xİİİ

KISALTMA LİSTESİ ..... XİV

ŞEKİL LİSTESİ ..... XVİ

ÇİZELGE LİSTESİ ..... XVİİİ

ÖZET XX

ABSTRACT ..... XXİİ

BÖLÜM 1

GİRİŞ 1

1.1 Literatür Özeti 3
1.2 Tezin Amacı 5
1.3 Hipotez 5
1.4 Tezin Yapısı 6

BÖLÜM 2

BİLGİYE ERİŞİM 8

2.1 Bilgi ve Erişim Sistemleri 8
2.1.1 Bilginin Depolanması 9
2.1.2 Bilgiye Erişim Sorunu 10
2.1.3 Bilgiye Erişim Sistemi Yapısı 11
2.1.4 Belge Dizinleme 13
2.2 Örün Robotları 15
2.2.1 Robot Çalışma Politikaları 15
2.2.2 Robot Mimarisi 16
2.2.3 Örün Robotu Çeşitleri 17
2.3 Terim Ağırlıklandırma 19
2.3.1 Vektör Uzayı Modeli 19
2.3.1.1 Vektör Uzayı Modelinin Eksik Yönleri 20

2.3.2 Terim Ağırlıklandırma Yöntemleri 20
2.3.2.1 Alternatif Terim Ağırlıklandırma Yöntemleri 23
2.4 Performans Ölçekleri 24
2.4.1 Duyarlılık 25
2.4.2 Anma 25
2.4.3 F-Ölçeği 26
2.4.4 Alıcı İşletim Karakteristiği 26

BÖLÜM 3

SINIFLANDIRMA VE BOYUT İNDİRGEME 28

3.1 Sınıflandırma 28
3.1.1 Örün Sayfalarının Sınıflandırılması Çalışmaları 29
3.2 Boyut İndirgeme 30
3.2.1 Özellik Seçimi 30
3.2.1.1 Karşılıklı Bilgi Özellik Seçim Yöntemi 31
3.2.1.2 Chi Kare Özellik Seçim Yöntemi 32
3.2.1.3 Korelasyon Katsayısı Özellik Seçim Yöntemi 33
3.2.2 Özellik Çıkarımı 33
3.2.2.1 Tekil Değer Çözümlemesi 34
3.2.2.2 Temel Bileşen Analizi 36
3.2.2.3 Saklı Anlamsal Çözümleme 37
3.2.2.4 Doğrusal Ayırtaç Çözümlemesi 38

BÖLÜM 4

SOYUT ÖZELLİK ÇIKARIM YÖNTEMİ 39

4.1 Soyut Özellikler 39
4.2 Soyut Özellik Çıkarım Algoritması 40
4.3 Örnek Problem Üzerinde Uygulama 42

BÖLÜM 5

VERİ KÜMELERİ 48

5.1 Ön İşleme Adımları 49
5.1.1 Belirtkeleme 49
5.1.2 Filtreleme 49
5.1.3 Gövdeleme 50
5.1.4 Kutu Çizimi Filtresi 50
5.2 DMOZ Veri Kümesi 52
5.2.1 DMOZ Veri Kümesi Önişleme Adımları 54
5.2.2 DMOZ Bağımsız Test Kümesi 56
5.3 Reuters-21578 Veri Kümesi 57
5.3.1 Reuters Veri Kümesi Önişleme Adımları 58
5.3.2 Reuters ModApte10 Versiyonu Veri Kümesi 59
5.4 20 Newsgroups Veri Kümesi 60
5.4.1 20 Newsgroups Veri Kümesi Önişleme Adımları 61

BÖLÜM 6

TEST SONUÇLARI 62

6.1 Deney Kurulumu 62
6.1.1 Uygulanan Doğrulama Yöntemleri 64
6.1.1.1 10 Kere Çapraz Doğrulama 64
6.1.1.2 Bağımsız Eğitim-Test Kümeleri 65
6.1.2 Seçilen Sınıflandırma Algoritmaları 65
6.1.3 Kullanılan Uygulama Ortamı 67
6.2 DMOZ Veri Kümesi ile Test Sonuçları 67
6.2.1 Bağımsız DMOZ Test Kümesi ile Sınıflandırma Sonuçları 72
6.3 Reuters Veri Kümesi ile Test Sonuçları 75
6.4 20-Newsgroups Veri Kümesi ile Test Sonuçları 78
6.5 ModApte-10 Veri Kümesi ile Test Sonuçları 81
6.6 Eşit Sayıda Özelliğe Sahip Veri Kümeleri ile Test Sonuçları 84
6.7 Hipotez Testi Sonuçları 93
6.8 Sınıflandırıcı Parametrelerinin Başarıma Olan Etkisinin Testi 95
6.9 Eğitim Örneği Sayısının Başarıma Olan Etkisinin Testi 97

BÖLÜM 7

SONUÇ VE ÖNERİLER 103

KAYNAKLAR 108

EK-A

VERİ KÜMELERİNDEN ÇIKARILAN SIK KULLANILAN KELİMELER 114

A-1 Türkçe Sık Kullanılan Kelimeler 114
A-2 İngilizce Sık Kullanılan Kelimeler 116

EK-B

ÇIKARILAN SOYUT ÖZELLİKLERİN VERİ KÜMELERİNDEKİ HER SINIF İÇİN ORTALAMA DEĞER GRAFİKLERİ 120

B-1 DMOZ Veri Kümesinde Soyut Özelliklerin Sınıflara Ait Olma Olasılıkları 121
B-2 Reuters Veri Kümesinde Soyut Özelliklerin Sınıflara Ait Olma Olasılıkları 122
B-3 20-Newsgroups Veri Kümesinde Soyut Özelliklerin Sınıflara Ait Olma Olasılıkları 123

EK-C

VERİ KÜMELERİNDE GERÇEKLEŞTİRİLEN TESTLERİN DUYARLILIK, ANMA, F-ÖLÇEĞİ VE ROC EĞRİSİ ALTINDAKİ ALAN SONUÇLARI 124

C-1 DMOZ Veri Kümesi İle Elde Edilen Performans Sonuçları 125
C-2 Bağımsız DMOZ Test Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen Performans Sonuçları 126
C-3 Reuters Veri Kümesi İle Elde Edilen Performans Sonuçları 127

C-4 20-Newsgroups Veri Kümesi İle Elde Edilen Performans Sonuçları 128
C-5 ModApte-10 Veri Kümesi İle Elde Edilen Performans Sonuçları 129
C-6 21 özellikli Reuters Veri Kümesi İle Elde Edilen Performans Sonuçları 130
C-7 20 Özellikli 20-Newsgroups Veri Kümesi İle Elde Edilen Performans Sonuçları 131
C-8 10 Özellikli ModApte-10 Veri Kümesi İle Elde Edilen Performans Sonuçları 132

EK-D

VERİ KÜMELERİNDE TESTLERİN EN İYİ VE EN KÖTÜ KARMAŞIKLIK MATRİSLERİ 133

D-1 DMOZ Veri Kümesinde Özellik Çıkarımı Olmadan Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle) 133
D-2 DMOZ Veri Kümesinde Özellik Çıkarımı Olmadan Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle) 133
D-3 DMOZ Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle) 134
D-4 DMOZ Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En İyi Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle) 134
D-5 DMOZ Test Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En Kötü Karmaşıklık Matrisi (SVM Algoritması İle) 134
D-6 DMOZ Test Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En İyi Karmaşıklık Matrisi (10 En Yakın Algoritması İle) 135
D-7 DMOZ Veri Kümesinde 11948 Eğitim 16660 Test Örneği İle Soyut Özellik Çıkarımıyla Elde Edilen En Kötü Karmaşıklık Matrisi (SVM Algoritması İle) 135
D-8 DMOZ Veri Kümesinde 11948 Eğitim 16660 Test Örneği İle Soyut Özellik Çıkarımıyla Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 135
D-9 DMOZ Veri Kümesinde 16660 Eğitim 11948 Test Örneği İle Soyut Özellik Çıkarımıyla Elde Edilen En Kötü Karmaşıklık Matrisi (SVM Algoritması İle) 136
D-10 DMOZ Veri Kümesinde 16660 Eğitim 11948 Test Örneği İle Soyut Özellik Çıkarımıyla Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 136
D-11 Reuters Veri Kümesinde Özellik Çıkarımı Olmadan Elde Edilen En Kötü Karmaşıklık Matrisi (Rasgele Orman Algoritması İle) 136
D-12 Reuters Veri Kümesinde Özellik Çıkarımı Olmadan Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 137
D-13 Reuters Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En Kötü Karmaşıklık Matrisi (LINEAR Algoritması İle) 137
D-14 Reuters Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En İyi Karmaşıklık Matrisleri (10 En Yakın Komşu ve SVM Algoritmaları İle) 137
D-15 Reuters Veri Kümesinde Chi-Kare İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle) 138
D-16 Reuters Veri Kümesinde Chi-Kare İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 138
D-17 Reuters Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle) 139
D-18 Reuters Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 139

D-19 Reuters Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle) 139
D-20 Reuters Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle) 140
D-21 Reuters Veri Kümesinde PCA İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle) 140
D-22 Reuters Veri Kümesinde PCA İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle) 140
D-23 Reuters Veri Kümesinde LSA İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle) 141
D-24 Reuters Veri Kümesinde LSA İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle) 141
D-25 Reuters Veri Kümesinde LDA İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle) 141
D-26 Reuters Veri Kümesinde LDA İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 142
D-27 20-Newsgroups Veri Kümesinde Özellik Çıkarımı Olmadan Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle) 142
D-28 20-Newsgroups Veri Kümesinde Özellik Çıkarımı Olmadan Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 142
D-29 20-Newsgroups Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En Kötü Karmaşıklık Matrisi (C4.5 Algoritması İle) 143
D-30 20-Newsgroups Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 143
D-31 20-Newsgroups Veri Kümesinde Chi-Kare İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle) 143
D-32 20-Newsgroups Veri Kümesinde Chi-Kare İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle) 144
D-33 20-Newsgroups Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En Kötü Karmaşıklık Matrisi (Rasgele Orman Algoritması İle) 144
D-34 20-Newsgroups Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 144
D-35 20-Newsgroups Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En Kötü Karmaşıklık Matrisi (RIPPER Algoritması İle) 145
D-36 20-Newsgroups Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle) 145
D-37 20-Newsgroups Veri Kümesinde PCA İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle) 145
D-38 20-Newsgroups Veri Kümesinde PCA İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 146
D-39 20-Newsgroups Veri Kümesinde LSA İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle) 146
D-40 20-Newsgroups Veri Kümesinde LSA İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 146
D-41 20-Newsgroups Veri Kümesinde LDA İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle) 147

D-42 20-Newsgroups Veri Kümesinde LDA İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 147
D-43 ModApte-10 Veri Kümesinde Özellik Çıkarımı Olmadan Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle) 147
D-44 ModApte-10 Veri Kümesinde Özellik Çıkarımı Olmadan Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle)..... 148
D-45 ModApte-10 Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En Kötü Karmaşıklık Matrisi (SVM Algoritması İle)..... 148
D-46 ModApte-10 Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En İyi Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle) 148
D-47 ModApte-10 Veri Kümesinde Chi-Kare İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)..... 148
D-48 ModApte-10 Veri Kümesinde Chi-Kare İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle)..... 149
D-49 ModApte-10 Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle) 149
D-50 ModApte-10 Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 149
D-51 ModApte-10 Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle) 149
D-52 ModApte-10 Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle)..... 150
D-53 ModApte-10 Veri Kümesinde PCA İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle) 150
D-54 ModApte-10 Veri Kümesinde PCA İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)..... 150
D-55 ModApte-10 Veri Kümesinde LSA İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle) 150
D-56 ModApte-10 Veri Kümesinde LSA İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle) 151
D-57 ModApte-10 Veri Kümesinde LDA İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle) 151
D-58 ModApte-10 Veri Kümesinde LDA İle Elde Edilen En İyi Karmaşıklık Matrisleri (10 En Yakın Komşu ve SVM Algoritmaları İle) 151
D-59 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En Kötü Karmaşıklık Matrisi (LINEAR Algoritması İle)..... 152
D-60 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En İyi Karmaşıklık Matrisleri (10 En Yakın Komşu ve SVM Algoritmaları İle) 152
D-61 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde Chi-Kare İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle)..... 153
D-62 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde Chi-Kare İle Elde Edilen En İyi Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)..... 153
D-63 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle)..... 153
D-64 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle)..... 154

D-65 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle) 154
D-66 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En İyi Karmaşıklık Matrisi (Rasgele Orman Algoritması İle) 154
D-67 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde PCA İle Elde Edilen En Kötü Karmaşıklık Matrisi (RIPPER Algoritması İle) 155
D-68 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde PCA İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 155
D-69 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde LSA İle Elde Edilen En Kötü Karmaşıklık Matrisi (RIPPER Algoritması İle) 155
D-70 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde LSA İle Elde Edilen En İyi Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle) 156
D-71 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En Kötü Karmaşıklık Matrisi (C4.5 Algoritması İle) 156
D-72 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 156
D-73 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde Chi-Kare İle Elde Edilen En Kötü Karmaşıklık Matrisi (RIPPER Algoritması İle) 157
D-74 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde Chi-Kare İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 157
D-75 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En Kötü Karmaşıklık Matrisi (RIPPER Algoritması İle) 157
D-76 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 158
D-77 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En Kötü Karmaşıklık Matrisi (RIPPER Algoritması İle) 158
D-78 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 158
D-79 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde PCA İle Elde Edilen En Kötü Karmaşıklık Matrisi (RIPPER Algoritması İle) 159
D-80 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde PCA İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 159
D-81 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde LSA İle Elde Edilen En Kötü Karmaşıklık Matrisi (RIPPER Algoritması İle) 159
D-82 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde LSA İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle) 160
D-83 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En Kötü Karmaşıklık Matrisi (SVM Algoritması İle) 160
D-84 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En İyi Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle) 160
D-85 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde Chi-Kare İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle) 161
D-86 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde Chi-Kare İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle) 161
D-87 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle) 161

D-88 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle) 161
D-89 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle) 162
D-90 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En İyi Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle) 162
D-91 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde PCA İle Elde Edilen En Kötü Karmaşıklık Matrisi (RIPPER Algoritması İle) 162
D-92 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde PCA İle Elde Edilen En İyi Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle) 162
D-93 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde LSA İle Elde Edilen En Kötü Karmaşıklık Matrisi (LINEAR Algoritması İle) 163
D-94 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde LSA İle Elde Edilen En İyi Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle) 163

EK-E

ARFF DOSYA FORMATI ÖRNEKLERİ 164

E-1 Örnek Bir Veri Kümesinin Standart Halinin ARFF Formatındaki Gösterimi 164
E-2 Örnek Bir Veri Kümesinin Soyut Özellik Çıkarımı Uygulanmış Halinin ARFF Formatındaki Gösterimi 164

ÖZGEÇMİŞ 165

SİMGE LİSTESİ

t_i	i. terim
d_j	j. belge
c_k	k. sınıf
q	Sorgu belgesi
$n_{t,d}$	d belgesinde t teriminin geçme sayısı
N	Belge sayısı
$tf_{t,d}$	d belgesinde t teriminin frekansı
df_t	t teriminin belge sıklığı
$w_{i,j}$	i. terimin j. belgedeki ağırlığı
$f_{i,j}$	w_i teriminin d_j belgesindeki frekansı
J_i, f_{wi}	w_i teriminin geçtiği belge sayısı
f_{dj}	d_j belgesindeki toplam terim sayısı
F	Toplam terim sayısı
p(x,y)	x ve y değişkenlerinin birleşik olasılık yoğunluk fonksiyonu
p(x)	x değişkeninin olasılığı
Σ	Kovaryans matrisi
Λ	Diyagonal matris
w	Özvektör
Σ_B	Sınıflar arası dağılım matrisi
Σ_W	Sınıf içi dağılım matrisi
μ_c	c sınıfının ortalaması
μ	Tüm sınıfların ortalamalarının ortalaması
$nc_{i,k}$	i. terimin k. sınıfta geçme sayısı
$w_{i,k}$	j. belgedeki i. terimin k. sınıfta geçme sayısı
$Y_{j,k}$	j. belgedeki tüm terimlerin k sınıfına olan toplam etkisi
$AF_{j,k}$	j. belge için k. soyut özellik

KISALTMA LİSTESİ

LSA	Latent Semantic Analysis
IG	Information Gain
MI	Mutual Information
CC	Correlation Coefficient
RELIEF	Relevance In Estimating Features
PCA	Principal Component Analysis
MDS	Multidimensional Scaling
LVQ	Learning Vector Quantization
LDA	Linear Discriminant Analysis
FA	Factor Analysis
SOM	Self-organizing Maps
ISOMAP	Isometric Feature Mapping
LLE	Local Linear Embedding
TF	Term Frequency
IDF	Inverse Document Frequency
TFIDF	Term Frequency – Inverse Document Frequency
$TF\chi^2$	Term Frequency-Chi-Squared
TFKLI	Term Frequency-Kullback-Leibler Information
TFRF	Term Frequency-Relevance Feedback
CSA	Concise Semantic Analysis
HTML	Hyper Text Markup Language
FTP	File Transfer Protocol
URL	Uniform Resource Locator
CSS	Cascading Style Sheets
OPIC	Online Page Importance Computation
HTTP	Hyper Text Transfer Protocol
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
ROC	Receiver Operating Characteristic
STC	Suffix Tree Clustering
MI	Mutual Information Feature Selection
CHI	Chi-squared Feature Selection

CC	Correlation Coefficient Feature Selection
PCA	Principal Component Analysis
LSA	Latent Semantic Analysis
LDA	Linear Discriminant Analysis
LVQ	Learning Vector Quantization
LLE	Local Linear Embedding
SOM	Self-Organizing Maps
ISOMAP	Isometric Feature Mapping
SVD	Singular Value Decomposition
AFE	Abstract Feature Extraction
DMOZ	Directory Mozilla
SVM	Support Vector Machines
EM	Expectation Maximization
WEKA	Waikato Environment for Knowledge Analysis
ARFF	Attribute-Relation File format

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 Dünyada sayısal ortamda yıllara göre üretilen veri miktarı 2
Şekil 2.1 Bilgiye erişim sistemi yapısı 11
Şekil 2.2 Bilgiye erişim sistemi çalışma şekli 12
Şekil 2.3 Örün robotu mimarisi 17
Şekil 2.4 Örnek ROC eğrisi 27
Şekil 4.1 Örnek veri kümesi 43
Şekil 4.2 Örnek veri kümesindeki soyut özelliklerin görselleştirilmiş hali 45
Şekil 4.3 Örnek veri kümesi için test belgeleri 46
Şekil 4.4 Örnek veri kümesinin üç ayrı yöntem ile çıkarılan özelliklerle gösterimi ve elde edilen sınıf ayırtaçları 47
Şekil 5.1 Açık dizin Türkçe bölümü ana kategorileri 53
Şekil 5.2 Açık dizinde alt kategorilerin bir örneği 53
Şekil 5.3 DMOZ dizinin taranması sonucu elde edilen klasör yapısı 54
Şekil 5.4 DMOZ veri kümesindeki kelime ve sayfa sayılarının dağılımı 55
Şekil 5.5 DMOZ veri kümesindeki örneklerin sınıflardaki dağılımı 56
Şekil 5.6 Bağımsız DMOZ test kümesindeki örneklerin sınıflardaki dağılımı 57
Şekil 5.7 Reuters-21578 veri kümesinde ön işleme adımları sonrası belgelerin sınıflardaki dağılımı ve alt-üst çeyrek çizgileri arasında kalan alan 58
Şekil 5.8 Reuters veri kümesindeki örneklerin sınıflardaki dağılımı 59
Şekil 5.9 ModApte-10 veri kümesindeki örneklerin sınıflardaki dağılımı 59
Şekil 5.10 20-Newsgroups veri kümesindeki sınıflar ve yakınlıklarına göre kümeleri 60
Şekil 6.1 10 kere çapraz doğrulamada veri kümesinin ayırımı 64
Şekil 6.2 Tüm kelime türlerinden oluşan DMOZ veri kümesi üzerinde özellik indirgeme yöntemlerinin sınıflandırma performansına olan etkisinin görsel olarak karşılaştırılması 69
Şekil 6.3 Soyut özellik çıkarım yönteminin sadece isim türünde kelimeleri içeren DMOZ veri kümesi üzerinde sınıflandırma performansına olan etkisi 71
Şekil 6.4 Soyut özellik çıkarım yönteminin bağımsız DMOZ test veri kümesi üzerinde sınıflandırma performansına olan etkisi 74
Şekil 6.5 Reuters veri kümesi üzerinde özellik indirgeme yöntemlerinin sınıflandırma performansına olan etkisinin görsel olarak karşılaştırılması 77
Şekil 6.6 20-Newsgroups veri kümesi üzerinde özellik indirgeme yöntemlerinin sınıflandırma performansına olan etkisinin görsel olarak karşılaştırılması 80
Şekil 6.7 ModApte-10 veri kümesi üzerinde özellik indirgeme yöntemlerinin sınıflandırma performansına olan etkisinin görsel olarak karşılaştırılması 83

Şekil 6.8 Eşit sayıda özellikle ifade edilen Reuters veri kümesi üzerinde seçilen yöntemlerin sınıflandırma performansına olan etkisinin görsel olarak karşılaştırılması 88
Şekil 6.9 Eşit sayıda özellikle ifade edilen 20-Newsgroups veri kümesinde seçilen yöntemlerin sınıflandırma performansına olan etkisinin görsel olarak karşılaştırılması 90
Şekil 6.10 Eşit sayıda özellikle ifade edilen ModApte-10 veri kümesi üzerinde seçilen yöntemlerin sınıflandırma performansına olan etkisinin görsel olarak karşılaştırılması 92
Şekil 6.11 ModApte-10 veri kümesinde SVM algoritmasının değişik çekirdek tiplerinin uygulanması ile elde edilen F_1 değerlerinin görsel olarak karşılaştırılması 97
Şekil 6.12 Soyut özellik çıkarımı yöntemi ile elde edilen sınıflandırma başarımının eğitim örneği sayısına göre değişiminin görsel olarak karşılaştırılması 98
Şekil 6.13 Soyut özellik çıkarımı yöntemi ile Naive Bayes sınıflandırıcısında elde edilen başarımın eğitim örneği sayısına göre değişimi 99
Şekil 6.14 Soyut özellik çıkarımı yöntemi ile C4.5 sınıflandırıcısında elde edilen başarımın eğitim örneği sayısına göre değişimi 99
Şekil 6.15 Soyut özellik çıkarımı yöntemi ile RIPPER sınıflandırıcısında elde edilen başarımın eğitim örneği sayısına göre değişimi 99
Şekil 6.16 Soyut özellik çıkarımı yöntemi ile 10 en yakın komşu sınıflandırıcısında elde edilen başarımın eğitim örneği sayısına göre değişimi..... 100
Şekil 6.17 Soyut özellik çıkarımı yöntemi ile rasgele orman sınıflandırıcısında elde edilen başarımın eğitim örneği sayısına göre değişimi 100
Şekil 6.18 Soyut özellik çıkarımı yöntemi ile SVM sınıflandırıcısında elde edilen başarımın eğitim örneği sayısına göre değişimi 100
Şekil 6.19 Soyut özellik çıkarımı yöntemi ile LINEAR sınıflandırıcısında elde edilen başarımın eğitim örneği sayısına göre değişimi 101

ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 Terim sıklığı bileşeni için alternatifler 21
Çizelge 2.2 Belge sıklığı bileşeni için alternatifler 22
Çizelge 2.3 Verilen bir örnek için hata matrisi 25
Çizelge 3.1 Chi kare ve korelasyon katsayısı yöntemlerinde yer alan ikili olasılıkların açıklamaları 32
Çizelge 4.1 Örnek veri kümesi için terim-belge matrisi 43
Çizelge 4.2 Örnek veri kümesi için terimlerin sınıflar üzerindeki ağırlıkları ($w_{i,k}$) 44
Çizelge 4.3 Örnek veri kümesi için belgelerin çıkarılan özellikleri ($AF_{j,k}$) 44
Çizelge 6.1 Seçilen yöntemlerle veri kümelerinin indirgenmiş özellik sayıları 63
Çizelge 6.2 Soyut özellik çıkarımı yönteminin tüm kelime türlerini içeren DMOZ veri kümesi kullanıldığında sınıflandırma performansına olan etkisinin karşılaştırılması 68
Çizelge 6.3 Sadece isim türünde kelimeleri içeren DMOZ veri kümesi kullanıldığında soyut özellik çıkarımı yönteminin sınıflandırma performansına olan etkisinin karşılaştırılması 70
Çizelge 6.4 DMOZ veri kümesinde tüm kelime türlerinden terim ve sadece isim türü terimlerin kullanılmasının sınıflandırma performansına olan etkisinin karşılaştırılması 72
Çizelge 6.5 Soyut özellik çıkarımı yönteminin bağımsız DMOZ test kümesi kullanıldığında sınıflandırma performansına olan etkisinin karşılaştırılması 73
Çizelge 6.6 Soyut özellik çıkarımı yönteminin Reuters veri kümesinde sınıflandırma performansına olan etkisinin karşılaştırılması 76
Çizelge 6.7 Soyut özellik çıkarımı yönteminin 20-Newsgroups veri kümesinde sınıflandırma performansına olan etkisinin karşılaştırılması 79
Çizelge 6.8 Soyut özellik çıkarımı yönteminin ModApte-10 veri kümesinde sınıflandırma performansına olan etkisinin karşılaştırılması 82
Çizelge 6.9 Seçilen yöntemlerin eşit sayıda özellikle ifade edilen Reuters veri kümesinde sınıflandırma performansına olan etkilerinin karşılaştırılması 87
Çizelge 6.10 Seçilen yöntemlerin eşit sayıda özellikle ifade edilen 20-Newsgroups veri kümesinde sınıflandırma performansına olan etkilerinin karşılaştırılması 89
Çizelge 6.11 Seçilen yöntemlerin eşit sayıda özellikle ifade edilen ModApte-10 veri kümesinde sınıflandırma performansına olan etkisinin karşılaştırılması .. 91
Çizelge 6.12 Reuters veri kümesinde t-testi uygulaması sonucu elde edilen ortalama F_1 değerleri, bu değerlerin ağırlıklı ortalamaları ve standart sapmaları 94

Çizelge 6.13 ModApte-10 veri kümesinde SVM algoritmasının değişik çekirdek tiplerinin uygulanması ile elde edilen F_1 değerleri 96
Çizelge 6.14 Soyut özellik çıkarımı yöntemi ile elde edilen sınıflandırma başarımının eğitim örneği sayısına göre değişimi 98

ÖZET

METİN SINIFLAMA İÇİN YENİ BİR ÖZELLİK ÇIKARIM YÖNTEMİ

Göksel BİRİCİK

Bilgisayar Mühendisliği Anabilim Dalı
Doktora Tezi

Tez Danışmanı: Prof. Dr. A. Coşkun SÖNMEZ

Metin tipindeki bilgiye erişim, verinin yapısı ve doğası gereği zorlu bir iştir. Bu sebeple bilgiye erişimi kolaylaştırmak için metnin özelliklerine göre kategorizasyon çözümleri geliştirilmiştir. Ancak metin verisi çok sayıda özellik içerdiği için kategorizasyon yöntemleri ile çalışmak da güçtür. Bu problemin çözümü için özellikleri azaltan boyut indirgeme yöntemleri ortaya atılmıştır. Boyut indirgemede ilk yaklaşım özellik seçimidir ve başarımı en az düşürecek ve efektif çalışmayı arttıracak şekilde, özelliklerin sayısının azaltılması hedeflenir. İkinci yaklaşım olan özellik çıkarımında ise amaç az sayıda yeni özellikle verinin yeniden tanımlanmasıdır.

Özellik seçim yöntemleri ile belgeleri diğerlerinden daha iyi tanımlayan terimler seçilmeye çalışılır. Bunun için çeşitli deneyler yaparak diğerlerinden daha iyi sonuç veren terim alt kümesini arayan yöntemler olduğu gibi, çeşitli değerlendirme ve dizme yöntemleriyle terimleri sıralayıp belirli bir eşik değerinin üzerinde değer alan terimleri seçen yöntemler de mevcuttur. Metin işleme uygulamalarında boyut indirgeme için genellikle özellik seçim yöntemleri tercih edilmektedir.

Özellik çıkarım yöntemleri, belgeleri terimlerin bileşkesini alarak daha düşük boyutlu yeni bir uzayda kaynaştırılmış yeni özelliklerle ifade eder. Bu sayede veri, sayıca daha az ve orijinallerinden bağımsız özelliklerle ifade edilmiş olur. Çıkarılan özellikler belgelerin karakteristikleri hakkında gözlenebilir bilgi de sunmaz.

Bu tez çalışması kapsamında, metin işleme alanı için yeni bir özellik çıkarım yöntemi geliştirilmiştir. Bir veri kümesinde yer alan terimlerin belgelerdeki dağılımları, belgelerin kategorilere ait olmasında etki sahibidir. Özellik seçiminde de terimlerin ayırt ediciliklerine bakarak seçim yapan yöntemler mevcuttur. Bu sebeple çalışmada ilk olarak terimlerin ayırt edicilikleri ağırlıklandırılarak ortaya çıkarılmıştır. Daha sonra belgeler her bir sınıf için etki değerlerinin bileşkesinden oluşan, yeni bir uzayda yer alan özelliklerle ifade edilmiştir. Çıkarılan özelliklere, belgelerdeki orijinal terimlerin her bir sınıfa olan etkisinin bileşkesini temsil ettiği için soyut özellikler adı verilmiştir. Kısaca,

soyut özellik çıkarım yöntemi ile belgelerdeki terimlerin içerdiği ayırt edicilik değerleri kullanılarak terimlerin sınıflara olan etkilerinin bileşkesi yeni bir uzayda soyut olarak ifade edilmiştir.

Soyut özellik çıkarım yönteminin başarımını test etmek ve diğer yöntemlerle karşılaştırmak üzere metin tipinde veri kümeleri üzerinde sınıflandırma testleri gerçekleştirilmiştir. Türkçe veri kümesi olarak DMOZ dizininden taranan örün sayfaları ile bir veri kümesi oluşturulmuştur. Sonuçları doğrulamak üzere bağımsız bir DMOZ test veri kümesi de hazırlanıp kontrol testleri yapılmıştır. Standart veri kümeleri olarak Reuters-21578 ve 20-Newsgroups seçilmiş ve kullanılmıştır. Bağımsız eğitim-test kümeleri ile test yapabilmek için, ModApte-10 veri kümesi ile de testler tekrarlanmıştır. Karşılaştırma için özellik seçim yöntemleri olarak chi-kare, korelasyon katsayısı ve karşılıklı bilgi, özellik çıkarım yöntemleri olarak da PCA, LSA ve LDA testlere dahil edilmiştir. Sınıflandırma testleri için değişik tasarım yaklaşımlarına sahip algoritmalar tercih edilmiştir. İstatistiki sınıflandırıcı olarak Naive Bayes, karar ağacı olarak C4.5, kural tabanlı sınıflandırıcı olarak RIPPER, örnek temelli yöntem olarak 10 en yakın komşu, kontrollü varyasyonlara sahip karar ağaçları koleksiyonu için rastgele orman, çekirdek tabanlı sınıflandırıcı olarak destek vektör makineleri, doğrusal sınıflandırıcı olarak LINEAR kullanılmıştır. Ayrıca sınıflandırma algoritmalarının parametrelerinin başarıma olan etkisini ölçmek üzere destek vektör makineleri sınıflandırma algoritması farklı çekirdek alternatifleriyle denenerek sınanmıştır. Sınıflandırma deneylerinde doğrulama için standart eğitim ve test kümesi ayırımı olan veri kümeleri haricinde 10 kere çapraz doğrulama kullanılmıştır.

Yapılan testlerin sonuçlarına göre soyut özellik çıkarım yöntemi diğer yöntemlerden daha yüksek başarım sağlamıştır. Yöntem bazında testlerin ortalama sonuçları incelendiğinde de soyut özellik çıkarım yönteminin başarımı diğerlerinden yüksektir. Bu sonuçlardan anlaşılacağı üzere soyut özellik çıkarım yöntemi veri kümelerini metin işleme uygulamalarına efektif olarak hazırlamak için kullanılabilir. Bunun yanında yöntem sınıfların ayrılabilirliği hakkında da bilgi vermektedir. Yöntem ile ortaya çıkarılan soyut özellikler, örneklerin kendi sınıfına ve diğer sınıflara ait olma olasılıkları olarak da değerlendirilebilir. Örneklerdeki soyut özelliklerin değerleri birbirine yakın olduğunda sınıfların ayrılabilirliği az olmaktadır. Soyut özelliklerin değerleri arasındaki farklar büyüdükçe sınıfları bağımsız olarak ayırt etmek daha kolaydır.

Anahtar Kelimeler: Bilgiye erişim, metin işleme, boyut indirgeme, özellik çıkarımı, sınıflandırma için ön işleme, soyut özellikler

ABSTRACT

A NEW METHOD ON FEATURE EXTRACTION FOR TEXT CLASSIFICATION

Göksel BİRİCİK

Department of Computer Engineering
PhD. Thesis

Advisor: Prof. Dr. A. Coşkun SÖNMEZ

Textual information retrieval is a challenging task due to complex structure and nature. Categorization solutions based on the features of textual data are presented in order to overcome challenges and simplify information retrieval process. Since textual data contains vast amount of features that causes the curse of dimensionality, implementing categorization becomes an arduous task. Dimension reduction techniques are introduced to overcome this problem. There are two approaches for reducing dimensions of the feature space. The first approach, feature selection, selects a subset of the original features as the new features, which is expected to increase affectivity and minimally decrease performance. The second approach, feature extraction, reduces dimension by creating new features that redefines data. This is achieved by combining or projecting the original features.

Feature selection methods try to find the features that explain data better than others, and output a smaller subset of features. Selection procedure is based on either evaluation of features on a specific classifier to find the best subset, or ranking of features by a metric and eliminating the ones that are below the threshold value. Feature selections methods have a broader usage than feature extraction for dimension reduction in text processing.

Feature extraction algorithms map the multidimensional feature space to a lower dimensional space. This is achieved by combining features to form a new description for the data with sufficient accuracy. Since the projected features are transformed into a new space, they no longer resemble the original feature set, but extract relevant information from the input set.

In this thesis, we introduce a novel feature extraction method for text classification. The distribution of terms on documents affects the membership in classes. We also

know that there are feature selection methods which evaluate the worthiness of features regarding to their distinctiveness's. Based on this fact, we weigh and reveal the distinctiveness of the features as the first step. Then the documents are represented with new extracted features that consist of the combination of their weights, in a lower dimensional space. Since the new features resemble the combined effect of original features to each class, we name the extracted features as abstract features. Briefly, we project high dimensional features of documents onto a new feature space having dimensions equal to the number of classes in order to form the abstract features.

We execute classification tasks on text datasets in order to test the performance of abstract feature extraction and compare with other methods. The first dataset is built up of Turkish web pages crawled from DMOZ directory. A separate test dataset is also prepared for validation by crawling unseen web pages from DMOZ again. Reuters-21578 and 20-Newsgroups datasets are used as standard text datasets. Tests are repeated using ModApte-10 dataset for observing the results with separate train and test samples. We select chi-squared, correlation coefficient and mutual information as feature selection and PCA, LSA and LDA as feature extraction methods for comparing the classification and clustering performances with abstract feature extractor. We use Naïve Bayes as a simple probabilistic classifier. We choose C4.5 decision tree algorithm for a basic tree based classifier, and a random forest with 10 trees to construct a collection of decision trees with controlled variations. We choose 10-nearest neighbour algorithm to test instance-based and RIPPER to test rule based classifiers. We use SVM for kernel based classifier and LINEAR as a linear classifier. In order to test the impact of classifier parameters on performance, we compare different kernel alternatives for SVM classifier. Instead of train-test split datasets, we use 10 fold cross validation to evaluate the tests.

Test results show that abstract feature extraction leads to higher performance results than the other methods. If we look at average results on method basis, abstract feature extraction is ahead of the compared methods again. These results prove that abstract feature extraction algorithm can be used to effectively prepare datasets to text processing tasks. Not only abstract feature extraction makes it possible to prepare datasets in an effective way, but also gives information about class separability. The extracted abstract features can be seen as the membership probabilities of samples to the classes. These features also describe the likelihood of a sample to other classes. We can infer that if the values of abstract features are close to each other, class separability is low. As the distances between the abstract features increase, it becomes easier to distinguish the classes.

Key words: Information retrieval, text processing, dimension reduction, feature extraction, preprocessing for classification, abstract features

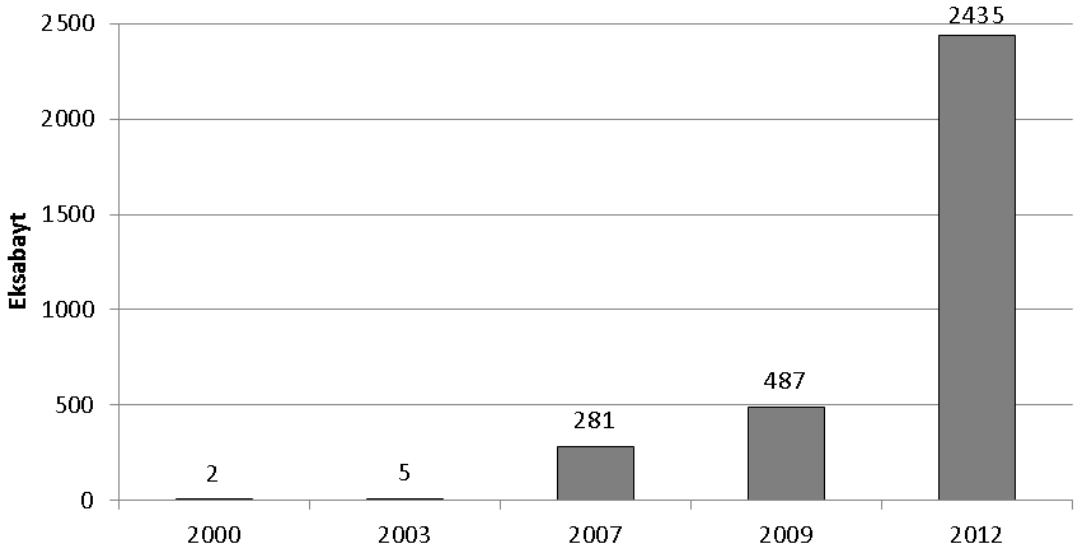
BÖLÜM 1

GİRİŞ

İnsanlar bildiklerini ve birikimlerini daha sonraki nesillere aktarabilmek ister. Bunun için de bilgilerin kaydedilmesi, depolanması gereklidir. Kayıtlı bilgiye erişim ise, depolanan bilgi miktarı arttıkça önemli bir sorun haline gelmektedir. Tarih boyunca bu soruna çözüm üretebilmek için çalışmalar yapılmıştır. Birçok düşünür ve bilim adamı bilgi erişim sorunuyla ilgilenerek, depolama ve erişim için yapılması gerekenleri tartışmışlardır. Platon, bilgi kaydetmek için kullanılan en temel eleman olan yazıyı sorgulayarak, "insanlar yazıyı öğrenirlerse akıllarına unutkanlık aşılanacağını" söylemiştir [1]. Gelişen bilim ile insan bilgi depolama konusunda çözümler ortaya koyabilmiştir. İnsanlığın ürettiği bilgilerin istediği kadarını cebinde taşıyabileceği teknolojiler ortaya çıkmıştır. Bilgi birikimini bir yonga ile vücuda, hatta beyne ilave etmenin çok uzak bir gelecekte olabileceğini tahmin etmek de zor değildir.

Günümüze kıyasla insan geçmişte çok az bilgi üretebilmekte ve az miktarda bilgiye erişim sağlayabilmekteydi. Teknolojinin gelişmesi ile ortaya çıkan depolama olanaklarının artması, sayısal ortamda çok büyük miktarda verilerin saklanmasını sağlamıştır. 2000 yılında dünya üzerinde sayısal ortamda üretilen veri miktarı 2 eksabayt (2000 katrilyon bayt) iken, 2007 yılında 281 eksabayta çıkmıştır [2]. 2011 yılındaki veri miktarı ise 2006 yılının 10 katı, yaklaşık 1800 eksabayttır. 2012'de ise bu miktarın 2435 eksabayt olması beklenmektedir. Sayısal ortamda üretilen verinin yıllara göre artımı Şekil 1.1'de verilmiştir. Bu veriler; sayısal kayıt ortamları, kablosuz akıllı kart verileri, sayısal televizyon, sayısal müzik ve ses, sayısal sabit ve hareketli görüntüler, internet üzerinden haberleşme ve ses transferi, tıbbi görüntüleme, kişisel ve kurumsal

bilgisayarlar, veri merkezi uygulamaları, oyunlar, uydu görüntüleri, küresel konumlama sistemi verileri, bankacılık işlemleri, tarayıcı ve algılayıcı verileri, uçtan uca paylaşılan dosyalar, e-posta, anında mesajlaşma, video-konferans, bilgisayar destekli modelleme ve üretim verileri gibi pek çok ve farklı kaynakça oluşturulmaktadır.



Şekil 1.1 Dünyada sayısal ortamda yıllara göre üretilen veri miktarı

Sözünü ettiğimiz veri temel olarak hareketli veya sabit görüntü, ses ya da metin tipindedir. Temel veri tiplerinin iki veya daha fazlasını bir arada barındıran veri tiplerine ise çoklu ortam adı verilmektedir. Geçmişten günümüze kalan bilgi birikiminin en büyük kısmının yazı ile saklandığını göz önüne aldığımızda, metin tipindeki bilgi hacminin ne kadar büyük olduğunu da anlayabiliriz.

Bahsettiğimiz farklı tiplerdeki bilgiye erişim için tek bir çeşit yöntem kullanılması söz konusu olamayacağı için verilerin yapılarındaki değişikliklere uygun farklı bilgi erişim yöntemleri ortaya atılmıştır. Metin tipindeki veride olduğu gibi, bu kadar fazla miktarda verinin içinden istenilenlere erişmek, “samanlıkta iğne aramak” deyimini bile yetersiz bırakacak kadar zordur. Bu sebeple insanlar bilgiye erişimi kolaylaştırmak için metnin özelliklerine göre kategorizasyon çözümleri geliştirmişlerdir. Dewey Ondalık sınıflandırma, LCC Kongre Kütüphanesi sınıflandırma sistemi gibi çözümler, metnin özelliklerine göre kategorilere ayrılmasını ve kolay erişimi sağlayan çözümlerden

standart olarak kullanılanlarına örnek verilebilir. Ancak burada karşımıza çıkan sorun, içeriğe göre bilgiye erişimi sağlamak üzere metnin hangi özelliklerle temsil edileceğidir.

Metin tipindeki bilgi, temelde kelimelerden oluşmaktadır. Bir metinde çok sayıda kelime kullanılmakta, bu sebeple de metin tipindeki veriler yapısında çok sayıda özellik barındırmaktadır. Bundan dolayı metin tipindeki verileri kategorizasyon yöntemleri ile işlemek zordur. Bu problemin çözümü içinse kategorize edilecek verinin özellik boyutlarını indirgeme yöntemleri ortaya atılmıştır. Bu yöntemlerde amaç, başarımı en az düşürecek ve efektif çalışmayı arttıracak şekilde, özelliklerin sayısının azaltılması veya az sayıda yeni özellikle verinin yeniden tanımlanmasıdır. Verinin boyutlarını indirgeme, sonrasında yapılacak sınıflandırma, kümeleme, bağlanım (*regression*) gibi işlemlere özel olarak seçilebilir ya da şekillendirilebilir. Bunun dışında değişik alanlarda kullanılabilen, genel boyut indirgeme yöntemleri de mevcuttur. Bu bölümde tezin amacı ile birlikte literatürde mevcut benzeşen örneklerden kısaca bahsedilecek, boyut indirgeme yöntemleri ve karşılaştırma için seçilen örnekleri Bölüm 3.3'te ayrıntıları ile açıklanacaktır.

1.1 Literatür Özeti

Metin işleme alanında özelliklerin boyutlarını indirgemek için genellikle özellik seçim yöntemlerinden faydalanılmaktadır. Bu yöntemlerin ortak noktası ise doğrusal modele sahip olmalarıdır. Bilgi kuramına dayanarak, özelliklerin taşıdığı bilgi miktarlarına ve ayırma derecelerine bakılıp yüksek bilgi taşıyan veya iyi ayırıcılığa sahip özelliklerin seçilmesi, kalanların da elenmesi ile boyut indirgenmektedir. Bilgi kazanımı (information gain, IG), karşılıklı bilgi (mutual information, MI), ki-kare özellik seçimi (chi-squared feature selection), korelasyon katsayısı (correlation coefficient, CC) gibi yöntemler bilgi kuramına dayanan özellik seçim yöntemleridir [3]. Bunların dışında veri bir sınıflandırma algoritması üzerinde denenerek seçiciliği fazla olan özelliklerin bulunması da özellik seçimi için kullanılabilir.

Doğrusal olmayan özellik seçim yöntemlerinin metin işleme alanında uygulamalarına karmaşıklıklarından ve uygulama zorluklarından dolayı nadiren rastlanmaktadır [4].

RELIEF ve çeşitli çekirdek yöntemleri doğrusal olmayan özellik seçim yöntemlerine örnek olarak verilebilir.

Metin işleme alanında özel olarak boyut indirgeme için yapılan çok sayıda özellik çıkarımı çalışması bulunmamaktadır. Bilinen ve en yaygın olarak kullanılan boyut indirgeme yöntemi saklı anlamsal çözümlemedir (latent semantic analysis, LSA) [5]. LSA'nın olasılıksal bir versiyonu da bulunmaktadır [6]. LSA temel olarak doğrusal bir özellik çıkarım yöntemidir. Bunun dışında temel bileşen çözümlemesi (principal component analyis, PCA), çok boyutlu ölçekleme (multidimensional scaling, MDS), öğrenen yöney nicemleme (learning vector quantization, LVQ), doğrusal ayırtaç çözümlemesi (linear discriminant analysis, LDA), etken çözümlemesi (factor analysis, FA) gibi pek çok genel amaçlı özellik çıkarım yöntemi de metin işleme alanında uygulanmıştır [7], [8]. Doğrusal olmayan özellik çıkarım yöntemlerine örnek olarak kendini düzenleyen eşlemeler (self organizing maps, SOM), eş ölçülü özellik eşleme (isometric feature mapping, ISOMAP) ve yerel doğrusal gömme (local linear embedding, LLE) örnek olarak verilebilir. Doğrusal olmayan yöntemler metin işlemeden çok verinin görselleştirilmesi çalışmalarında kullanılmıştır [9].

Metin işleme için boyut indirgeme işlemlerinde, özelliklerin verinin taşıdığı karakteristiklere uygun olacak şekilde metriklerle temsil edilmesi gereklidir. Bu konuda yapılmış pek çok ağırlıklandırma çalışması bulunmaktadır [10], [11]. Ancak TFIDF (*term frequency-inverse document frequency*) [12] en çok bilinen ve en yaygın olarak kullanılan terim ağırlıklandırma yöntemidir. Bunun dışında en bilinenleri TFKLI (*term frequency-Kullback-Leibler information*), TFRF (*text frequency-relevance feedback*) olmak üzere pek çok terim ağırlıklandırma yöntemi ortaya atılmıştır [11],[13]. Yine de TFIDF; en bilinen, en yaygın kullanılan ve yeni çıkan yöntemlerle hala rahatlıkla karşılaştırılabilir ve başa çıkabilir terim ağırlıklandırma yöntemi olmayı sürdürmektedir [11].

Metin işlemlerine hazırlık olması amacıyla özellik çıkarımı ile ilgili yapılan güncel bir çalışma özlü anlamsal çözümleme (*concise semantic analysis*, CSA) adıyla bilinmektedir [14]. Tezde önerdiğimiz yöntemle, probleme bakış açısı olarak benzeşen bu yöntem,

özelliklerin sınıfları nasıl etkilediğini dikkate almamakta, sadece verinin tümüne olan etkisine bakmaktadır. Ayrıca, iki yöntemin özellik ağırlıklandırmaları, çıkarılan özellikler ve sayıları birbirinden farklıdır. CSA yönteminde ilk başta belirlenen sayı kadar özellik ağırlıklandırılmaktadır. Tezde önerdiğimiz yöntem ise veri kümesindeki sınıf sayısı kadar özellik çıkarmaktadır. Ayrıca CSA yöntemi özellikleri ağırlıklandırdığı haliyle kullanmakta, önerdiğimiz yöntem ise kendi ağırlıklandırmasına ilave olarak lineer dönüşüm ile örnekler için tanımlayıcı yeni özellikler türetmektedir.

1.2 Tezin Amacı

Metin tipindeki veri çok fazla sayıda özellikten oluşmaktadır. Bu yapı, çok boyutluluğun laneti (*curse of dimensionality*) olarak da bilinen soruna yol açmaktadır. Bu problemi önlemek için boyut azaltma işlemlerinin uygulanması gerekmektedir.

Önceki bölümde bahsedildiği gibi, metin işleme alanı için yapılmış çok fazla özellik çıkarım yöntemi bulunmamakta, genellikle özellik seçim yöntemleri ile boyutlar azaltılmaktadır. Tezdeki amacımız, özellikle metin sınıflandırma yöntemlerinde problem teşkil eden çok boyutlu yapıyı çözen, sınıflar arası ayırımları göz önünde tutan ve belirleyen, metin işleme alanında kullanılacak yeni bir özellik çıkarım yöntemi tasarlamaktır.

Metin işleme için özellik çıkarımı alanında doğrusal olmayan (*non-linear*) yöntemlerin uygulanabilir olmaması [9] sebebiyle, önerdiğimiz özellik çıkarım yöntemi doğrusal eşleme (*linear mapping*) temellidir. Ayrıca veriyi mümkün olan en az sayıda boyuta taşımak da amaçlarımız arasındadır. Çıkardığımız özelliklerin aynı zamanda verinin görselleştirilmesi alanında da kullanılabilir olması, hedeflerimiz arasında yer almaktadır.

1.3 Hipotez

Amacımız bir metni işleyip sınıflandırma yolu ile kategorize etmek olduğunda, metnin özellikleri olarak içerdiği kelimeleri kullandığımızı varsayalım. Bu durumda bir örnekte

bulunan her bir özellik, o örneğin bir sınıfa ait olmasında az veya çok etki sahibi olacaktır. Bu etkilerin değerlerini bir çeşit ağırlıklandırma ile ortaya çıkarabiliriz.

Hedefimiz her bir kategori için etki değerlerinin karmasını bulmaktır. Özellikleri, elde ettiğimiz etki değerlerini kullanarak boyutları kategori sayısına eşit bir uzaya doğrusal olarak eşlersek (*linear mapping*), bir örnekte bulunan özelliklerin her bir kategoriye olan etkisinin bileşkesini elde edebiliriz. Bu sayede, bir örnekte her bir sınıf için ne kadar kanıt ya da bilgi bulunduğunu elde edebiliriz.

Boyutları indirgenmiş uzayda oluşan yeni değerler, verideki örnekleri tanımlayan soyut özellikler olarak kullanılabilir. Başka bir deyişle yaptığımız ağırlıklandırma ve doğrusal eşleme ile mümkün olan en az sayıdaki özellik elde edilmiş, özellik çıkarımı gerçekleştirilmiştir.

İki ya da üç kategorinin bulunduğu veri kümelerinde, çıkarılan soyut özelliklerin değerleri eksenlerde yer alacak şekilde veri kümesindeki örneklerin görselleştirilmesi de mümkündür. Örneklerden ortaya çıkarılan kategorilere aidiyet bilgileri ile verideki örneklerin kategoriler arasında nasıl dağıldığını ve kategoriler arası ayrımı izlemek olasıdır.

1.4 Tezin Yapısı

Tez kitabının yapısı şu şekildedir. İkinci bölümde bilgi ve bilgiye erişim konuları tanıtılarak mevcut problemler verilmektedir. Üçüncü bölümde bilgiye erişimde kategorizasyon çözümü için sınıflandırma ile işlemleri kolaylaştırmak için geliştirilmiş olan boyut indirgeme yöntemleri tanıtılarak literatürde yer bulmuş örneklerden bahsedilmektedir. Dördüncü bölümde tez çalışmasında geliştirdiğimiz soyut özellik çıkarım yöntemi tanıtılmaktadır. Beşinci bölümde karşılaştırma testlerinde kullanılan veri kümeleri, özellikleri ve ön işleme adımları ile birlikte açıklanmaktadır. Altıncı bölümde gerçekleştirilen testler, deney kurulumları ve elde edilen karşılaştırmalı sonuçlar sunularak tartışılmaktadır. Son bölümde ise tez çalışması, geliştirilen yöntem ve elde edilen sonuçlar karşılaştırılarak değerlendirilmektedir. Son bölümde ayrıca tezde geliştirilen yöntemle ilgili olası gelecek çalışmalara da yer verilmiştir.

Veri kümelerinden elenen Türkçe ve İngilizce sık kullanılan kelimelerin listesi, çıkarılan soyut özelliklerin veri kümelerindeki her sınıf için görselleştirilmiş grafikleri, gerçekleştirilen testlerde elde edilen detaylı performans sonuçları, testlerin en kötü ve en iyi karmaşıklık matrisleri ve deneylerde kullanılan işletim ortamının dosya formatı ekler kısmında verilmiştir.

BÖLÜM 2

BİLGİYE ERİŞİM

Bilgiye erişim (*information retrieval*) yaklaşık yarım yüzyıllık geçmişi olan bir disiplindir. Bilgi toplama, sınıflama, kataloglama, depolama, büyük miktardaki verilerden arama yapma ve bu verilerden istenilen bilgiyi üretme teknik ve süreci olarak tanımlanır [15].

Bu bölümde bilgi ve bilgiye erişim sorunu, İnternetten bilgiye erişim için kullanılan örün robotları ve özellikleri, metin tipindeki bilgiye erişimde kullanılan uzay ve terim ağırlıklandırma yöntemleri, bilgiye erişimde başarımı ölçmek için kullanılan performans ölçekleri tanıtılacaktır.

2.1 Bilgi ve Erişim Sistemleri

Bilgi için Türk Dil Kurumunun verdiği tanım şöyledir [16]:

1.İnsan aklının erebileceği olgu, gerçek ve ilkelerin bütünü, bili, malumat.

2.Öğrenme, araştırma veya gözlem yolu ile elde edilen gerçek, malumat, vukuf.

3.İnsan zekâsının çalışması sonucu ortaya çıkan düşünce ürünü, malumat, vukuf.

4.*Felsefe* Genel olarak ve ilk sezi durumunda zihnin kavradığı temel düşünceler.

5.Bilim

6.*Bilişim* Kurallardan yararlanarak kişinin veriye yönelttiği anlam.

Sözlük tanımlarından hareketle, bilgi sözcüğünün üç temel kullanımı vardır [17]:

1.Süreç olarak bilgi (*Information-as-process*): Bilgi bir “bilgilendirme etkinliği”dir.

2.Bilgi olarak bilgi (*Information-as-knowledge*): Bilgilendirme etkinliği sırasında bir konu ya da olaya ilişkin verilen haber veya bilginin ifadesidir.

3. Nesne olarak bilgi (*Information-as-thing*): Bilgi olarak adlandırılan veri ya da belgelerdir. Bu nesneler bilgilendirici niteliğe sahiptir.

Bilgi olarak bilginin ana özelliği soyut bir kavram olmasıdır. Bu yüzden bilgi iletilmek üzere açıklanmalı, tanımlanmalı, fiziksel bir şekilde temsil edilmelidir. Bu şekilde bir temsil ise, "simgeleme", bir diğer deyişle "nesne olarak bilgi" olarak bilinmektedir. Nesne olarak bilgiyi işleyerek yeni formlarda nesne olarak bilgi edilmesine ise "bilgi işleme" (information processing) adı verilir. Bilgi işleme için kullanılan araçlara da bilgi teknolojisi adı verilir [17].

2.1.1 Bilginin Depolanması

İnsan beyni yüzyıllar boyunca bilgi için tek depolama kaynağı olmuştur. Ancak doğa gereği ölümle birlikte kaydedilen tüm bilgiler de yok olur, bu yüzden geçici bir kayıt ortamıdır. Bilginin depolanarak gelecek nesillere aktarılması sayesinde insanlık gelişmiş ve diğer tüm canlılardan farklı şekilde ilerleyebilmiştir. Depolama için keşfedilen araç ise yazıdır. Yazının bulunması ile birlikte bilginin insan beyni dışında kaydedilmesi de mümkün hale gelmiştir.

Bilginin saklanması konusunda değişik görüşler bulunmaktadır. Platon kitabında, Eski Mısır Şehir Krallarından Thamus'un yazıyı bulan tanrı Theuth'a karşı çıkış gerekçesini şu şekilde yazmıştır: "İnsanlar yazıyı öğrenirlerse akıllarına unutkanlık aşılanır; bellek alıştırması yapmayı bırakırlar. Çünkü yazılı olana güvenirler; şeyleri ezbere değil, dışsal işaretler aracılığıyla hatırlamaya çalışırlar. Keşfettiğiniz şey bellek için değil, hatırlama için bir reçetedir. Ve size inananlara sunduğunuz şey gerçek bir hikmet değil, sadece onun görüntüsüdür. Çünkü size inananlara birçok şey söyleyerek, ama öğretmeden, onları çok biliyorlarmış gibi gösterebilirsiniz. Oysa onlar çoğunlukla hiçbir şey bilmezler. Ve insanlar hikmetle (*wisdom*) değil de aldatıcı hikmetle yüklenirlerse diğer insanlara yük olurlar [18]." Ancak şu da bir gerçektir ki, yazı bulunana dek birikimlerin aktarılması, sadece insanların yetenekleriyle sınırlıydı. Yazı bulunduktan sonra oluşturulan "bilgi kütüphaneleri" ile bilginin arttırılarak aktarılması mümkün olmuştur. Thamus'un düşüncesinin tersine, bilim felsefecisi Karl Popper, "Dünya uygarlığı bir savaşla yok olup, geriye kütüphanelerde saklanan nesnel bilgi içeriği kalırsa, uygarlığı yeniden kurmak mümkündür. Hâlbuki bu nesnel bilgi içeriği, yani kütüphaneler yok

olup, yalnızca öznelerin öğrenme yeteneği kalsa, çağdaş uygarlığı yeniden inşa etmek hemen hemen imkânsızdır" demektedir [19].

Nesnel bilgi içeriğine erişmek için yıllardan beri düşünürler ve bilim adamları çeşitli öneriler ortaya atmışlardır. Örneğin, H.G. Wells 1938 yılında yazdığı *World Brain* (Dünya Beyni) adlı kitabında "Dünya Ansiklopedisi" adını verdiği ve insanlığın ulaştığı uygarlık düzeyini yansıtan bütün bilgileri içinde barındıracak bir ansiklopedi yaratılmasını önermiştir. Wells'e göre modern Dünya Ansiklopedisi, konu uzmanlarının onayıyla her konuda dikkatle derlenmiş ve eleştirel bir biçimde sunulmuş seçme bilgilerden ve alıntılardan oluşacaktır. Herhangi bir konuda bilgi edinmek için, sürekli gözden geçirilen, geliştirilen, büyüyen, yaşayan bir kaynak olarak düşünülen bu ansiklopedi dışında bir kaynağa başvurmak gerekmeyecekti [20].

İkinci Dünya Savaşı sırasında Amerikan Bilimsel Araştırma ve Geliştirme Ofisi müdürü olan Dr. Vannevar Bush, "memex" (*Memory Extension*) adını verdiği bir cihazdan söz eder. Bush memex'i, bir kimsenin kitap, gazete, dergi, yazışma, resim gibi tüm kişisel verilerini depolayabileceği, gerektiğinde sorgulayabileceği mekanik aygıt olarak tarif etmiştir. Sorgulama için dizinleme yerine, insan beyninin çalışma modeli olan ilişki kurma yoluyla seçim modelini önermiştir [21]. Başka bir deyişle, yaklaşık 60 yıl önce "hiper metin" (*hypertext*) terimini kullanmadan örün modelini öngörmüştür.

Günümüzde İnternet, Wells'in Dünya Ansiklopedisi kavramıyla örtüşmektedir. Aynı zamanda Bush'un tanımladığı "memex" de hiper metnin, dolayısıyla örünün atasıdır. Bilginin depolanması, kuşaktan kuşağa aktarım için olmazsa olmaz bir ihtiyaçtır. İnternet ve örün ise bu depolama ihtiyacını gidermiştir. Ancak hakkında bilgi bulmak için bilinmeyen bir şeyin tanımlanması gereği, depolanmış olan bilgiye erişim sorununu doğurmuştur.

2.1.2 Bilgiye Erişim Sorunu

Bilginin depolanmasının gerekli olduğunu tartıştıktan sonra, sadece depolanan bilginin bir işe yaramayacağı, aynı zamanda bu bilgiye etkin bir şekilde erişimin de olması gerektiği sonucu ortaya çıkmıştır. Varian'ın vurguladığı gibi, bilgiyi üretmek, yaymak ve

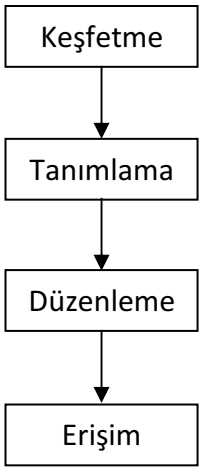
depolamak için kullanılan teknoloji, bu bilgiyi bulmak ve düzenlemek için bir yöntem yoksa yararsızdır [22].

Bilgi erişim sorunu, bilgi ihtiyacımızı tanımladığımız terimler ile bu ihtiyacımızı karşılaması muhtemel kaynaklarda bulunan terimlerin eşleştirilmesi problemidir. Bu yüzden bilginin doğru bir şekilde sınıflandırılması, düzenlenmesi gereklidir. Büyük miktarda bilgiler arasından istenilen bilgiyi bulmaya çalışmak, "yangın hortumundan su içmeye" benzetilmektedir [23].

Bilgi erişim sorununun bir diğer yönü de, hakkında bilgi bulmak için, bilinmeyen bir şeyin tanımlanması gereğidir. Sözlüğün ne olduğunu bilmeyen birisi, bir kelimenin anlamını öğrenmek için sözlüğe bakmayı akıl edemez. İnternet kaynaklı bilgiye erişim de bu problemi aynen taşımaktadır: Bir arama sonucunda, iyi tanımlanmamış, çok fazla ya da çok az bilgiye erişim söz konusudur.

2.1.3 Bilgiye Erişim Sistemi Yapısı

İdeal bir bilgi erişim sistemi, ilgili belgelerin tümüne erişim sağlamalıdır. Bunun yanında birbirine benzeyen bilgileri bir araya getirmeli, benzemeyenleri de ayırmalıdır. Bilgi erişim sisteminin genel yapısı, Şekil 2.1'de verilmiştir.

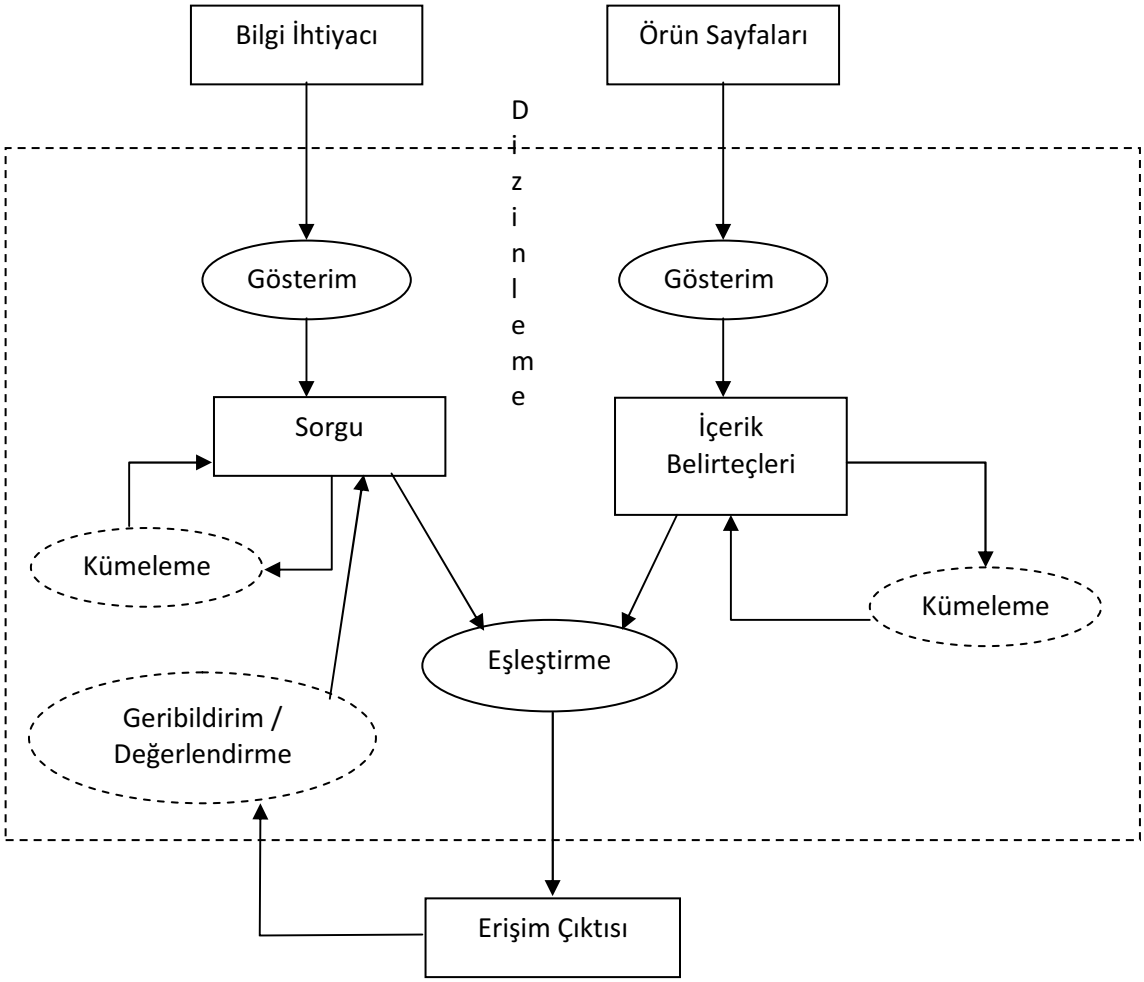


Şekil 2.1 Bilgiye erişim sistemi yapısı

Bilgi erişim sistemini oluşturmak için ilk olarak belgelerin keşfedilmesi gereklidir. Örün sayfalarının keşfedilmesi için örün robotlarından yararlanılır. Bu konu Bölüm 2.2'de detaylı olarak açıklanacaktır. Tanımlama aşamasında her bir kaynak için belirleyici özellikler bulunur ve çıkarılır. Bu özelliklere içerik belirteci adı da verilmektedir.

Düzenleme aşamasında belgeler ve onları temsil eden içerik belirteçleri dizinlenir. Konulara göre sınıflandırma da yine düzenleme aşamasıdır. Bir sorgu için de içerik belirteçlerinin çıkarılması ve dizinlenmiş kaynaklarda eşleşme aramasının yapılması ise erişim ayağını oluşturur.

Genel bir bilgi erişim sisteminin işlevsel birimleri ve çalışma şekli Şekil 2.2'de verilmiştir [24]. Dikdörtgen şekilde gösterilenler kavramlar, oval şekilde gösterilenler ise temel süreçlerdir. Kesikli oval şekiller ise seçenekli süreçlerdir. Bilgi ihtiyacı, örün sayfaları ve erişim çıktısı sistemin ön yüzünü oluşturur. Sorgular ve içerik belirteçleri ise sistemin arka yüzünü oluşturur.



Şekil 2.2 Bilgiye erişim sistemi çalışma şekli

2.1.4 Belge Dizinleme

İçerik belirteçlerinin belirlenmesi süreci, belge dizinleme olarak da bilinir. Bir belge için oluşturulan söz konusu içerik belirteçleri, belgenin tamamını temsil etmek için kullanılır. Bu yüzden de bilgi erişim sisteminin en önemli adımlarından birini oluşturur.

Erişim sisteminde belge olarak örün sayfaları kullanıldığında, belge dizinleme süreci adımları şu şekildedir [25]:

1.Belge Doğrusallaştırma

a.Yapısal metin (*markup*) ve şekil bilgisinin temizlenmesi

b.Belirtkeleme (*tokenization*)

2.Filtreleme

3.Gövdeleme (*Stemming*)

4.Ağırlıklandırma

Örün belgelerinin özelliği, içerisinde yapısal metin dilinin kullanılıyor olması ve şekil bilgisi taşımasıdır. Bu gibi düzenleme ve görünüme yönelik yapılar içeriğe yönelik bilgi taşımadıkları için, dizinleme işleminde yerleri yoktur ve doğrusallaştırma aşamasında elenirler. Veri tabanı kaynaklı ya da düz metin belgelerinin dizinlenmesi için bu aşamaya gerek yoktur.

Yapısal metin ve şekil bilgisinin temizlenmesi aşamasında, örün belgelerini oluşturan HTML etiketleri ve taşıdıkları değerler temizlenir. Belge erişim sisteminin yapısına göre tüm etiketler temizlenebileceği gibi, içerik belirlemede destek sağlayabilecek veri taşıyan etiket bilgileri (başlık, yardımcı veri (*metadata*), görüntü açıklayıcı metinler gibi) saklanabilir.

Belirtkeleme aşamasında, temizlenen örün sayfasındaki metin kelimelere ayrıştırılır, küçük harfe çevrilir ve noktalama işaretleri temizlenir. Kelimeler içerik belirteci olacakları için, heceleme kontrollerine özen gösterilir. Standart dışı karakterler temizlenir.

Belge doğrusallaştırma aşamasında ayrıca belgede kullanılan stil tanımları da temizlenir. Ancak CSS (*Cascading Style Sheets*) tanımları ile üst üste veya yan yana

katmanlar tanımlanabilir. Bu durumda sayfada kullanıcı baktığında görsel olarak yakın duran, birbiri ile ilgili içerik belirteçleri, aslında sayfanın kaynak kodunda birbirinden uzakta bulunabilir. Bu da bir örün sayfası incelendiğinde insan algısı ile makinenin algıladığı içeriğin birbirinden farklı olması sonucunu doğurur. Bu yüzden doğrusallaştırma aşamasında stil tanımları temizlenirken dikkatli davranılması gerekmektedir.

Filtreleme aşamasında örün sayfasını en iyi şekilde temsil edecek olan içerik belirteçleri süzülür. Buradaki amaç, sayfanın içeriğini en iyi temsil eden ve sayfayı diğer sayfalardan en çok ayıran belirteçlerin seçilmesidir. Belgelerin çoğunda yer alan belirteçler, ayırt edici değildir. Bu yüzden durma listesi (*stopword list*) olarak bilinen, sık kullanılan kelime ve terimler atılır. Durma listesi sık kullanılan sözcüklerden oluşan bir kütüphane olabileceği gibi, dizinlenmek üzere toplanan örün sayfalarının genelinde yapılan bir inceleme sonucu da oluşturulabilir. Bu sayede koleksiyondaki tüm belgeler için geçerli bir durma listesi oluşturulabilir.

Gövdeleme işlemi, bir kelimeden çekim eklerinin ayrılarak kelime kökünün ortaya çıkarılması aşamasıdır. Bilgi erişim sistemlerinde gövdelemenin avantaj ve dezavantajları konusunda değişik fikirler bulunmaktadır. Gövdeleme ile bir sorgunun biraz değişik hallerinin de sonuç kümesinde bulunması sağlanır. Ayrıca belirteçlerin boyutunun da küçülmesini sağlar. Ancak kelimelere gövdeleme işleminin fazla uygulanması, kafa karıştırıcı derecede ilgisiz sonuçların da dönmesine yol açabilir [25]. Gövdeleme işleminde başarıyı arttıran en önemli unsur, hedefteki dilin yapısıdır. Türkçe gibi sondan eklemeli dillerde gövdeleme işlemi, anma ve duyarlılık değerlerinde %20 ila %25 oranında artış sağlayabilir [26]. İngilizcedeki etkisi ise daha azdır. Bu yüzden dil seçimi bilgi erişim sistemlerinde önemli bir aşamadır.

Belge dizinlemenin son aşaması ise elde edilen terimlere ağırlık vermedir. Bir terimin bir belgedeki ağırlığını, o belgede bulunması belirler. Terim belgede yer alıyorsa bir, almıyorsa sıfır ağırlığındadır. Bu yaklaşıma Boole Modeli adı verilir. Bunun dışında terimlerin bir belgedeki frekansı ile dokümanların genelinde bulunma frekansından oluşan vektör tabanlı karma bir model daha popülerdir. Bu modele *tf*idf* adı verilir [27]. Burada tf terimin ilgili belgede geçme sıklığı, idf ise terimin geçtiği belge sayısının

devrik sıklığıdır. Bu çarpım sonucunda bir terimin ilgili belgede geçme sıklığı yüksek ve diğer belgelerde geçme sıklığı düşükse, göreceli ağırlığının yüksek olması (ayırt edici terim olması) sağlanır. Terim ağırlıklandırma ile ilgili detaylı bilgi Bölüm 2.3'te detaylarıyla açıklanacaktır.

2.2 Örün Robotları

Bilgi erişim sistemini oluşturmak için ilk olarak belgelerin keşfedilmesinin gerekli olduğu Bölüm 2.1.3'te tartışılmıştı. Örün sayfalarının keşfedilmesi için örün robotlarından yararlanılır. Örünü otomatik olarak gezip sayfaların içeriğini bilgi erişim sistemi oluşturmak üzere inceleyerek alan program veya betiklere örün robotu adı verilir [28]. Sürünücü (crawler), örümcek (spider), karınca (ants), otomatik endeksleyici gibi değişik isimler de örün robotu yerine daha az sıklıkla kullanılmaktadır. Örün robotunun gezintisi, özellikle arama motorlarında güncel verilerin sunulması için kullanılmaktadır. Robot gezdiği örün sayfalarının bir kopyasını içeriğe göre indeksleme yapılması ve arama sırasında hızlı erişim için saklarlar. İndeksleme dışında site yöneticileri tarafından bağlantıların (link) kontrolü için, ya da öründen belirli bir konuda bilgi toplamak için de kullanılabilir.

Bir robotun gezintisi, örünün çok büyük boyutu (yaklaşık 5 milyon terabayt) [29], hızlı değişim oranı ve isteğe göre dinamik yaratılan sayfalar gibi nedenler yüzünden oldukça zordur. Örünün çok büyük boyutu yüzünden bir robot, verilen bir zaman aralığında sadece belirli bir sayıdaki sayfayı gezebilir. Gezdiği zamandaki sayfaların içeriğinin, zaman içerisinde değişme olasılığı da çok yüksektir. Birçok sayfa parametrelere dayalı olarak, çok sayıda kombinasyona sahip olabilir. Bu sebeple robot gezintisi için mutlaka bir politika belirlemeli, her aşamada sırada ziyaret edeceği sayfanın hangisi olduğunu belirlemelidir [30].

2.2.1 Robot Çalışma Politikaları

Yapılan çalışmalar, robotlar yardımıyla taranarak örünün ancak %16'lık bir kısmının indekslenebildiğini göstermiştir [31]. En basit seçim politikası, "enine arama" olarak da bilinen bağlantıların görüldükleri sıra ile işlenmesidir. O yüzden öründe rastgele seçilmiş sayfalar yerine, ilgili sayfalarda arama yapılması daha önemlidir. İlgili sayfaları

ayırmak için sayfalara verilen bağlantılar, yapılan atıflar, ziyaret sayıları gibi kıstaslardan oluşan bir değer ile robotun gezeceği sayfaların seçim sıralaması yapılabilir. Bir sayfaya verilen bağlantıların sayısı ile değerlilik hesaplayan PageRank algoritması [32], tarama sürdükçe değişen çevrimiçi sayfa önemlilik algoritması (OPIC) yaygın kullanılan seçim politikalarıdır [33].

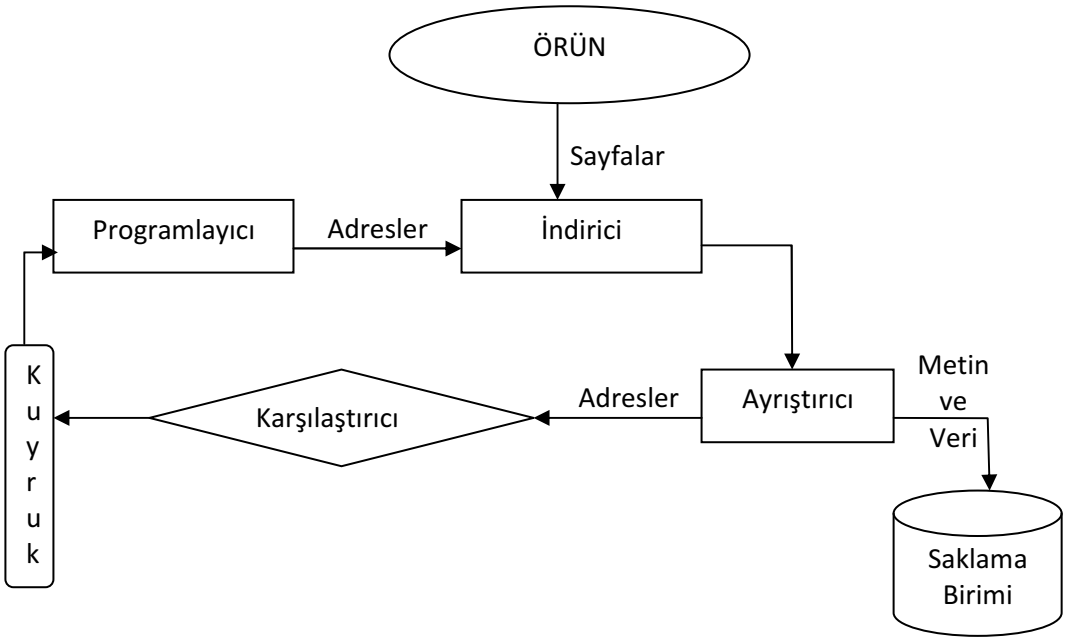
Örün sayfaları çok sık olarak değişmekte ve güncellenmektedir. Tahminlere göre örün sayfalarının %40'ı her ay değişmekte [34], İnternet ortamındaki bir bağlantının ortalama ömrü 44 gün sürmektedir [35]. Bu yüzden daha önceden ziyaret edilerek indekslenen sayfaların tekrar ziyaret edilerek indeksin güncelleştirilmesi gereklidir. Tekrar ziyaret için iki farklı politika geliştirilmiştir [36]. Birincisi sayfaların değiştirilme sürelerine bakmaksızın hepsini aynı zaman aralığında güncellemek, ikincisi ise değişmelerin günlüğünü tutarak sık değişen sayfaları daha sık ziyaret etmektir. Sık değişen sayfaları daha sık ziyaret etmek daha mantıklı görünebilir ancak Cho ve Garcia-Molina, düzenli tekrar ziyaretin daha başarılı olduğunu kanıtlamıştır [36]. Çok sık değişen sayfaları dışlayarak, sayfaların değişme sıklığına göre tekrar ziyaret frekansının ayarlanması ise en uygun çözümü getirmektedir.

Robotların örünü gezme hızları, insanlarınkinden çok daha yüksektir. Bu yüzden kibarlık politikası kullanmayan bir robot örün sunucularının bant genişliklerini kolaylıkla tüketebilir. Örün sitesi yöneticileri için geliştirilen bir protokol (robots.txt) ile robotların ziyaret edebileceği sayfalar sınırlandırılabilir [37]. Ancak robotların bu protokole uymak gibi bir zorunluluğu yoktur.

2.2.2 Robot Mimarisi

Bir robotun iyi seçilmiş politikalar dışında iyi bir mimariye sahip olması da önemlidir. Amaca yönelik olarak, ne kadar sayfa indirileceği, ne kadar zamanda gezileceği gibi kıstaslarla robotlar tasarlanır. Örün robotunun genel mimari yapısı Şekil 2.3'te verilmiştir. Programlayıcı, gezilecek adresleri sıralar. İndirici, genellikle paralel işlem birimleri şeklinde gerçeklenir ve öründen sayfaları indirerek saklar. Ayrıştırıcı, dizinlenecek bilgileri ve gezilecek yeni adresleri indirilen kaynaktan elde eder. Gezilecek yeni adresler, daha önceden gezilip gezilmediklerinin kontrolü için karşılaştırıcıya girer.

Elde edilen yeni adresler programlayıcının kuyruğuna atılır. Bu döngü, gezilebilecek yeni adres kalmayana kadar ya da istenen hedefe ulaşılana kadar devam eder.



Şekil 2.3 Örün robotu mimarisi

2.2.3 Örün Robotu Çeşitleri

İsimleri "AbachoBOT"'dan "ZyBorg"'a kadar değişen robotlar, özellikle arama motorları olmak üzere, birçok uygulama tarafından kullanılmaktadır. Örneğin; AltaVista arama motoru "Scooter", Google ise "Googlebot" robotlarını kullanmaktadır. Bazı arama motorları yeni sayfaları bulmak için bir robot, sayfa bağlantılarını kontrol etmek için başka bir robot da kullanabilir. Arama motorları dışında robotlar sayfa bağlantılarının canlı olup olmadığını kontrol etmek, sayfa değişimlerini denetlemek, sayfanın HTML kodunun doğruluğunu ve standartlara uyumluluğunu kontrol etmek, FTP istemcisi olarak indirilecek olan dosyaların yönetimi, sayfalardan belirli bir bilgi toplamak gibi işlerde de kullanılmaktadır.

WebCrawler [38] örünün ilk kamuya açık, tam metin indeksini oluşturmak için kullanılmıştır. Sayfaları indirmek için "lib-WWW" kütüphanesini, öğelere ayrıştırma ve adresleri sıralama için başka programları kullanır. Örün ağ yapısını enine arama ile tarar.

Google Robotu [32] hakkında biraz bilgi verilmekle birlikte referans sadece mimarisinin ilk zamanlarını göstermektedir. C++ ve Phyton dilleri kullanılarak yazılmıştır. Metin parçalama işlemi hem tam metin indeksleme hem de adres çıkarma yordamları için kullanılır. Bu yüzden gezici ile indeksleme işlemleri birlikte çalışır. Birden fazla gezici yordamın adresleri işleyebilmesi için bir adres sunucusu kullanılır. Adres sunucusuna parçalama işlemi sonucunda elde edilen adresler yollanır. Sunucu adresin daha önceden gezilip gezilmediğini kontrol eder, yeni bir adres ise listeye ekler.

Mercator [39] Java dili ile yazılmış, dağıtık ve modüler bir örün robotudur. Örün sayfalarının hangi protokolle gezileceği (örneğin HTTP) ve nasıl işleneceği, değiştirilebilen protokol ve işlem modülleri ile belirlenir. Standart işlem modülü ile sayfalarda yer alan adresler çıkarılır. Diğer işlem modülleri ile sayfalardaki metinlerin indekslenmesi, istatistik çıkarma gibi işlemler de yapılabilir.

Ubicrawler [40] Java dili ile yazılmış, merkezi işlem birimi olmayan, birbirinin aynı ajanlardan oluşan dağıtık bir örün robotudur. Bir ajanın gezdiği sayfa başka bir ajan tarafından gezilmez, bu sayede gezinti sırasında üst üste binme engellenir. Robot mümkün olduğunca ölçeklenebilir ve hatalara karşı dayanıklı olması için tasarlanmıştır.

Nutch [41] yine Java dili ile yazılmış, Apache Lisansı ile dağıtılan bir örün robotudur. Lucene metin indeksleme paketi ile birlikte kullanılabilmesi için yazılmıştır. Belirli adresleri tarayabildiği gibi, tüm örün üzerinde tarama da gerçekleştirebilir. Barındırdığı parçalayıcı birimi gezilecek yeni adresleri ayıklarken, sayfalardaki metinleri de indekslenmek üzere ayrıştırır.

WEbSPHINX [42] kişisel ve siteye özgü bir robot olarak Java dili ile gerçekleştirilmiştir. Çoklu alt sürece sahip sayfa indirici ve HTML ayrıştırıcı, bir grafik arabiriminden verilen adresteki sayfaları işler. İşlenen sayfalardan elde edilen veri ile metin tabanlı bir arama motoru da oluşturulur.

WIRE [43] C++ dili ile yazılmış, genel kamu lisansı ile dağıtılan bir örün robotudur. Sayfaların indirilmesini zamanlamak ve sıralamak için çeşitli politikaları bulunur. İndirilen sayfalar hakkında rapor üreten ve istatistik tutan çeşitli parçaları ile örün nitelendirme için kullanılmaktadır.

2.3 Terim Ağırlıklandırma

Bir sayfanın içerisinde bulunan bir terimin istatistikî özellikleri o terimin sayfa hakkındaki tanımlayıcılığını ve sayfayı diğerlerinden ayırt ettirme özelliğini yansıtır [44]. Diğer bir deyişle, bir terim bir konu hakkında ne kadar ayırt edici ise, ilgili sayfalarda bulunma olasılığı yüksek, diğer sayfalarda bulunma olasılığı düşüktür. Terimleri ağırlıklandırma işleminde de bu özellik dikkate alınmaktadır.

2.3.1 Vektör Uzayı Modeli

Metin tipindeki veriler için, belgeleri terimlerden oluşan vektörler halinde ifade eden cebirsel modele vektör uzayı modeli adı verilir. Bilgi erişim sistemlerinde tekil belgelerden gelen yerel ve belge koleksiyonundan gelen genel bilgiler göz önüne alınarak ağırlıklandırma yapılır. Klasik ağırlıklandırma yaklaşımı Salton'un Vektör Uzayı Modeli olarak bilinir [45].

Vektör uzayı modelinde her boyut ayrı bir terime denk düşmektedir. Bir belgede bir terim bulunuyorsa, oluşan vektörde ilgili boyutun değeri sıfırdan farklı olacaktır. Belge vektöründe yer alan terimlerin ağırlıklandırılması içinse birçok yöntem ortaya atılmıştır. Bölüm 2.3.2'de bu yöntemlere çeşitli örnekler verilecektir.

Vektörlerin terimleri olarak kelimeler, seçilmiş anahtar kelimeler ya da daha uzun kelime bileşke yapıları kullanılabileceği gibi; 2-gram, 3-gram ya da n-gram gibi hece yapıları da kullanılabilir. Terim olarak kullanılacak yapı, uygulamaya bağlı olarak seçilebilir. Belgeler, terimleri kelimelerden oluşan vektörler şeklinde gösterildiğinde buna kelime torbası gösterimi (*bag-of-words representation*) adı verilmektedir.

Vektör uzayı modelinin örnek bir uygulaması olarak belge benzerliğinin saptanması gösterilebilir. Uzayda bir d_j belgesi $d_j=(t_{1,j}, t_{2,j}, ..., t_{i,j})$ şeklinde, benzerliği saptanacak q belgesi de $q=(t_{1,q}, t_{2,q}, ..., t_{i,q})$ şeklinde gösterilmiş olsun. Bu iki belge arasındaki benzerlik, vektör uzayı modelinde kosinüs benzerliği ile bulunabilir. Kosinüs benzerliği 2.1 ile hesaplanır.

$$Cos\theta_{D_i} = Benzerlik\left(D_j, Q\right) = \frac{\sum_i t_{i,j} t_{i,q}}{\sqrt{\sum_i t_{i,j}^2}\sqrt{\sum_i t_{i,q}^2}} \tag{2.1}$$

2.3.1.1 Vektör Uzayı Modelinin Eksik Yönleri

Terim vektör şemasının bazı kısıtları ve zorlukları vardır. Bunlar aşağıda listelenmiştir.

1. Hesaplama hatalarına karşı oldukça duyarlıdır. Bunun yanında işlem zamanı da fazladır.
2. Terim uzayına yeni bir eleman eklendiğinde tüm vektörlerin güncellenmesi gereklidir.
3. Uzun belgelerde (çok fazla terim içeren belgeler) benzerlik ölçmek zordur.
4. Benzer içeriğe sahip ancak farklı kelimeler kullanan belgelerde yanlış negatif sonuçlar yüksek olabilir. Kelime tabanlı bilgi erişim sistemlerindeki genel sorun budur.
5. Ön işlemeye bağlı olarak, gövdeleme işlemi yanlış pozitif sonuçlar doğurabilir.

Bu gibi engellerden fazla etkilenmemek için belgeleri en iyi tanımlayan kelimeler seçilmeli, durma listelerinde yer alan çok sık kullanılan kelimeler elenmeli, isim, fiil, sıfat gibi tanımlayıcı kelimeler kullanılmalı, çok farklı boyutlarda belgeler kullanılmamalı, belirli bir eşik değerinin altındaki belgeler elenmelidir.

Kelime tabanlı sistemlerde terimlerin birbirinden bağımsız oldukları düşünülür. Ancak çok anlamlı sözcükler (kara, yurt gibi) yüzünden ilgisiz belgeler benzer çıkabilir. Bu da duyarlılığı etkiler. Eş anlamlı sözcükler (misafir-konuk, ihtiyar-yaşlı) yüzünden de ilgili belgeler benzer terimleri içermedikleri için ayrı düşerler. Bu da anma başarısını düşürür.

2.3.2 Terim Ağırlıklandırma Yöntemleri

Vektör uzayı modelinde belge vektörlerini terimlerle ifade ederek işlem yaparken, terimlerin sadece belgede yer alıp almamasına bakmak yeterli işlem hassasiyetini sağlamayabilir. Bu sebeple çeşitli terim ağırlıklandırma yöntemleri önerilmiştir. Terim ağırlıklandırmada iki esas bileşen bulunmaktadır. Bu iki bileşen tek başlarına da ağırlıklandırma için kullanılabilir ancak bileşkeleri daha etkin sonuçlar üretmektedir.

Terim ağırlıklandırmada yer alan ilk bileşen terim sıklığıdır (*term frequency, TF*). Bir belge içinde diğer terimlere göre daha fazla bulunan bir terimin öneminin daha fazla

olması esasına dayanmaktadır. Bir *d* belgesinde *t* teriminin yer alma sayısı $n_{t,d}$, belgedeki toplam terim sayısı $|d|$ olsun. Bu durumda terim sıklığı 2.2 ile belirlenebilir.

$$TF_{t,d} = \frac{n_{t,d}}{|d|} \tag{2.2}$$

Bu sayede bir belgede sıklıkla geçen terimin değeri arttırılmış olur. Terim sıklığı için genellikle kullanılan alternatifler ve 2.2'ye göre hesaplanma formülleri Çizelge 2.1'de verilmiştir [46].

Çizelge 2.1 Terim sıklığı bileşeni için alternatifler

Terim Sıklığı Alternatifi	**Formülü**
Normal	$TF_{t,d}$ (2.3)
Logaritmik	$1+\log(TF_{t,d})$ (2.4)
Arttırılmış	$0{,}5+\frac{0{,}5xTF_{t,d}}{\max_t(TF_{t,d})}$ (2.5)
Mantıksal	$\begin{cases}1 & TF_{t,d}>0\\ 0 & \text{diger\_haller}\end{cases}$ (2.6)
Logaritmik Ortalama	$\frac{1+\log(TF_{t,d})}{1+\log(\text{avg}_{t\in d}(TF_{t,d}))}$ (2.7)

Bir terimin bir belgede çok fazla geçiyor olması ayırt ediciyken, veri kümesindeki bir çok belgede çok fazla geçmesi, bu terimin ayırt ediciliğini azaltır. Bu sebeple terim ağırlıklandırmada ikinci bileşen olarak belge sıklığı (document frequency) kullanılır. Veri kümesinde daha az sayıda belgede geçen bir terimin daha ayırt edici olması düşüncesinden hareketle, çok sayıda belgede geçen terimlerin ağırlığını azaltabiliriz. Bu işleme devrik belge sıklığı (*inverse document frequency, IDF*) adı verilir. Bir veri kümesinde *N* adet doküman varken, *t* teriminin belge sıklığı (terimin geçtiği belge sayısı) df_t ile verilmiş olsun. Devrik belge sıklığı 2.8 ile hesaplanır. Belge sıklığı için

genellikle kullanılan alternatifler ve hesaplanma formülleri Çizelge 2.2'de verilmiştir [46].

$$IDF_t = \log \frac{N}{df_t} \tag{2.8}$$

Çizelge 2.2 Belge sıklığı bileşeni için alternatifler

Belge Sıklığı Alternatifi	**Formülü**
Yok	1 (2.9)
Devrik Belge Sıklığı	$\log(\frac{N}{df_t})$ (2.10)
Olasılıksal Devrik Belge Sıklığı	$\max\left\{0, \log \frac{N - df_t}{df_t}\right\}$ (2.11)

Bir belgede çok bulunan, buna karşın diğer belgelerde az görülen bir terimin ayırt ediciliğinin daha yüksek olacağı açıktır. Bu sebeple terim sıklığı ve devrik belge sıklığı bileşenleri bir arada kullanılır ve tam ağırlıklandırma (*TFIDF*) adı verilir. Bir *t* teriminin *d* belgesindeki TFIDF ağırlığı 2.12 ile hesaplanır. 2.12'de, *TFIDF* ve *TF* doğru orantılı olduğu için, eşit uzunlukta belgeler olduğunda sorgu terimini en çok içerenler daha değerlidir. Değişik uzunluktaki belgelerde ise daha fazla sorgu örneği içerebileceği için uzun belgeler daha değerli hale gelir. Denklemde belge frekansı ile ağırlık arasında ters orantı olduğu da görülmektedir. Bu sayede bir terim ne kadar az belgede geçiyorsa o kadar değerli hale gelir.

$$TFIDF_{t,d} = TF_{t,d} \times IDF_t \tag{2.12}$$

Bir belgede, bir anahtar kelime çok fazla sayıda tekrar ederse terim frekansı bundan çok fazla etkilenir. Bunu engellemek için terim frekansları normalize edilebilir. 2.2 ile hesaplanan terim sıklığı yerine 2.13 kullanılırsa, normalize terim sıklığı TFIDF hesabında yer alır. Burada $TF_{max(D)}$, belgedeki en yüksek frekanslı terimin terim sıklığı değeridir.

$$NTF_{t,d} = \frac{TF_{t,d}}{TF_{\max}(D)} \tag{2.13}$$

2.3.2.1 Alternatif Terim Ağırlıklandırma Yöntemleri

Terim ağırlıklandırma için TFIDF dışında literatürde pek çok çalışma bulunmaktadır. Bunlar genellikle TFIDF yöntemini iyileştirmeyi ya da ondan daha başarılı olmayı hedeflemektedir. Ancak yine de TFIDF en çok bilinen ve en yaygın olarak kullanılan, hatta alternatif yöntemlerle başabaş karşılaştırılabilen bir terim ağırlıklandırma yöntemidir [11]. Bu bölümde alternatif terim ağırlıklandırma yöntemleri hakkında kısa bilgi verilecektir.

Sanderson ve Ruthven'in [47] Glasgow Bilgi Erişim Raporunda sundukları ağırlık formülü 2.14'te verilmiştir. Burada $w_{i,j}$ *i*. terimin *j*. belgedeki ağırlığını, $freq_{i,j}$ *i*. terimin *j*. belgedeki terim frekansını, $length_j$ *j*. belgedeki toplam anahtar kelime sayısını, *N* belge sayısını, n_i ise *i.* terimin geçtiği belge sayısını göstermektedir. Frekans için 2.13'te verilen normalize edilmiş frekanslar kullanılabilir.

$$w_{i,j} = \frac{\log\left(freq_{i,j}+1\right)}{\log\left(length_j\right)} \times \log\left(\frac{N}{n_i}\right) \tag{2.14}$$

Terim frekanslarındaki değişimlerden korunmak için yapılan değişikliklerin yanında, belgelerin genelinde olan özelliklerden yararlanmak için modellerde mutlaka devrik belge sıklığı kullanılmaktadır. Glasgow modeli sayesinde çok uzun belgelerin değeri düşürülmekte, baskın olmaları önlenmektedir.

Terimlerin içerdiği bilgi miktarlarına bakarak ağırlıklandırma konusunda yapılan çalışmaların en bilineni Kullback-Leibler bilgisini (*Kullback-Leibler Information*) kullanan çalışmadır. TFKLI olarak bilinen ağırlıklandırma yöntemi 2.15'te verilmiştir. Burada $f_{i,j}$, w_i teriminin d_j belgesindeki frekansı; f_{w_i}, w_i teriminin geçtiği belge sayısı; f_{d_j} d_j belgesindeki toplam terim sayısı; F ise toplam terim sayısını ifade etmektedir.

$$TFKLI_{w_i} = \frac{f_{w_i}}{F} \sum_{d_j \in D} \frac{f_{i,j}}{f_{w_i}} \log \frac{f_{i,j} \big/ f_{w_i}}{f_{d_j} \big/ F} \tag{2.15}$$

Etiketli bir veri kümesi kullanıldığında, örneklerin etiket bilgilerinden ağırlıklandırma için faydalanmak başarımı artırabilir. Terimlerin ilgili sınıfta olmalarına değer verip ilgisiz sınıflardaki terimlerin etkileri azaltılarak TFRF olarak bilinen ağırlıklandırma yöntemi ortaya atılmıştır. Bir *d* belgesindeki *t* terimi için; *a*, *t*'yi içeren ve pozitif kategoride olan belge sayısı; *c*, *t*'yi içeren ve negatif kategoride olan belge sayısı olarak verilmiş olsun. Bu durumda TFRF ağırlıklandırması 2.16 ile hesaplanabilir.

$$TFRF_{t,d} = \frac{n_{t,d}}{|d|} \times \log(2 + \frac{a}{\max(1, c)}) \quad (2.16)$$

2.4 Performans Ölçekleri

Bilgi erişim sistemlerinin performanslarını belirlemek üzere kullanılan ölçeklerdir. Bunların kullanılması için belgelerden oluşan bir veri kümesi veya bir sorgu sonucu olması gereklidir. Sınıflandırma ya da kümeleme açısından bakıldığında ise belgenin ilgili sınıfa veya kümeye ait olmasının doğruluğunu ölçerler. Performans ölçekleri değerlendirme için karmaşıklık matrisinde (*confusion matrix*) tanımlı değerleri kullanır. Bu değerler, verilen bir veri kümesinde gerçekte ait olduğu kategori ve tahmin sonucu öngörülen kategorisi uyuşan ve uyuşmayan örneklerin sayısını göstermektedir.

Bir veri kümesinde gerçekte pozitif sınıfta olup, sınıflandırmada pozitif sınıfta yer alan bir örneğe doğru pozitif (*true positive, TP*) adı verilir. Örnek, gerçekte negatif sınıfa ait olup sınıflandırma sonucunda negatif sınıfta yer alırsa doğru negatif (*true negative, TN*) adını alır. Gerçekte negatif sınıfta olmasına rağmen pozitif sınıfta tahmin edilen örneğe yanlış pozitif (*false positive, FP*) denir. Pozitif sınıfta olmasına rağmen negatif sınıfta tahmin edilen örnekler ise yanlış negatif (*false negative, FN*) olarak isimlendirilmektedir. Çizelge 2.3'te örneğin doğru sınıfta olması ile ilgili tüm durumlar verilmiştir. Hata matrisindeki değerlerden yararlanarak geliştirilen performans ölçekleri alt bölümlerde açıklanacaktır.

Çizelge 2.3 Verilen bir örnek için hata matrisi

		Gerçek Kategori	
Tahmin Edilen Kategori		Kategori 1	Kategori 2
	Kategori 1	Doğru Pozitif (TP)	Yanlış Pozitif (FP)
	Kategori 2	Yanlış Negatif (FN)	Doğru Negatif (TN)

2.4.1 Duyarlılık

Kategorizasyon işlemleri için duyarlılık (*precision*) ile bir örneğin doğru kategoride olması durumu ölçülür. Gerçekten doğru sınıfta olan örnek sayısının test sonucu doğruymuş gibi bulanan örnek sayısına oranı ile bulunur. Bilgiye erişim sistemlerinde ise duyarlılık, sorgu sonuçlarından ne kadarının kullanıcının istediği gibi olduğunu ölçen orandır. Başka bir deyişle duyarlılık, getirilen sonuçlar içinden rastgele seçilen bir belgenin ilgili olma olasılığıdır. Duyarlılık için genel formül 2.17'de verilmiştir.

$$precision = \frac{TP}{TP + FP} \tag{2.17}$$

2.4.2 Anma

Kategorizasyonun hassasiyetini ölçmek için kullanılan ölçeğe anma (*recall*) adı verilir. Doğru pozitif örneklerin oranı olarak da bilinir. 2.18'de verilen anma ölçeği, bilgiye erişim sistemlerinde ise sorguyla ilgili sonuçların getirilme oranını verir. Rasgele seçilen bir belgenin sorgu sonuçları arasında olma olasılığı olarak da düşünülebilir.

Tek başına anma ölçeği kullanmak performansı ölçmek için yeterli değildir. Örneğin veri kümesindeki tüm örnekler bir sınıftaymış gibi bir sonuç elde edilirse, anma değeri %100 olur. Bu sebeple tek başına duyarlılık ya da anma yerine ikisinin bileşkesi yöntemleri kullanmak daha sağlıklı ölçüm sonuçları doğurmaktadır.

$$recall = \frac{TP}{TP + FN} \tag{2.18}$$

2.4.3 F-Ölçeği

Duyarlılık ve anma ölçeklerinin ağırlıklı harmonik ortalamasına F-ölçeği (*F-measure*) adı verilir. F_1 ölçeği 2.19'da verilmiştir. Duyarlılık ve anma eşit ağırlıkta olduğu için bu ölçek F_1 ölçeği olarak da bilinir.

$$F_1 = \frac{2 \times precision \times recall}{(precision + recall)} \tag{2.19}$$

F-Ölçeği için genel formül ise 2.20'de verilmiştir. $F_{0.5}$ ve F_2 de bu formüle göre sık kullanılan ölçeklerdir. F_2 ölçeğinde anma değerinin ağırlığı duyarlılık ölçeğine oranla iki kat ağırlıklandırılır. $F_{0.5}$ ölçeğinde ise duyarlılık değeri sonuca anma ölçeğinden iki kat daha fazla etki eder.

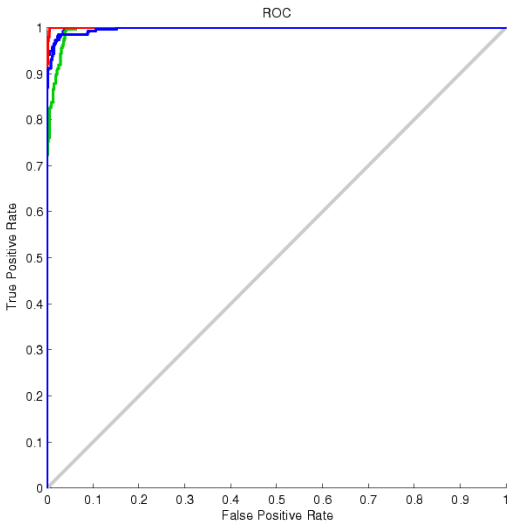
$$F_\alpha = \frac{(1 + \alpha^2) \times precision \times recall}{(\alpha^2 \times precision + recall)} \tag{2.20}$$

Genel olarak kategorizasyon performansının ölçülmesinde 2.19'da verilmiş olan F_1 ölçeği kullanılır. Kategoriler arası dağılımı homojen olmayan ya da belirli bir kategoride çok az sayıda örnek barındıran veri kümelerindeki kategorizasyon işlemlerinde başarımı ölçmek için de bu ölçeğin kullanımı daha uygundur.

2.4.4 Alıcı İşletim Karakteristiği

Doğru pozitif örnek sayısının yanlış pozitif örnek sayısına olan oranı hassasiyet (*sensitivity*) olarak bilinir. Eğer bu oran grafik eğri olarak çizilirse, bu eğri alıcı işletim karakteristiği (*receiver operating characteristic*, ROC) adını alır. ROC eğrisi aynı zamanda doğru pozitif oranının yanlış pozitif oranına bölümünden elde edilen değerler ile de çizdirilebilir. Burada doğru pozitif oranı, doğru pozitiflerin tüm gerçek pozitiflere (doğru pozitif ve yanlış negatif toplamı) oranı, yanlış pozitif oranı ise yanlış pozitiflerin tüm gerçek negatiflere (yanlış pozitif ve doğru negatif toplamı) oranı ile hesaplanır.

Şekil 2.4'te örnek bir ROC eğrisi çizimi yer almaktadır. Eğrinin (0,1) noktasından geçmesi, hassasiyetin maksimum olduğunu, yani tüm örneklerin doğru kategorize edildiğini gösterir. Diğer bir deyişle ROC eğrisi (0,1) noktasına yaklaştığında modelin daha iyi kategorizasyon yaptığı, uzaklaştığında daha başarısız sonuçlar ürettiği anlaşılır.



Şekil 2.4 Örnek ROC eğrisi

ROC eğrisi altında kalan alanın büyük olması modelin kategorizasyonda başarılı olduğunu gösterdiğinden, genellikle modellerin performanslarının karşılaştırılmasında kullanılır [48]. Ancak son yıllarda yapılan çalışmalarla ROC eğrisi altındaki alan ölçeğinin kategorizasyon işlemleri için gürültüye oldukça duyarlı olduğu, ayrıca bunun gibi birtakım ölçüm problemlerini de beraberinde getirdiği anlaşılmıştır [49].

BÖLÜM 3

SINIFLANDIRMA VE BOYUT İNDİRGEME

Metin tipindeki bilgiye erişimi kolaylaştırmak üzere kategorizasyon çözümlerinin geliştirildiğinden giriş bölümünde bahsedilmişti. Bu kategorizasyon, eğitim kümesinin etiketli veya etiketsiz olması, ya da etiket bilgisinin kullanılıp kullanılmamasına göre ikiye ayrılır. Eğer etiket bilgisi var ve kategorizasyonda kullanılıyorsa, bu işleme sınıflandırma adı verilir. Etiket bilgisi kullanılmadan benzer özelliklere bakılarak yapılan kategorizasyon işlemi ise kümeleme adını alır.

Sınıflandırma işleminin karmaşıklığı örneklerin barındırdıkları özellik boyutuna (sayısına) çok bağlıdır. Özellik sayısı arttıkça işlem zamanı uzar, işlem için gereken işlemci ve bellek gibi sistem kaynakları ihtiyacı artar. Bu gibi problemler özellikle metin tipindeki verilerde, yapısı gereği içerdiği yüksek özellik boyutları sebebiyle kaçınılmaz olarak oluşmaktadır. Çok boyutluluğun laneti olarak da bilinen bu problemi çözebilmek için boyut indirgeme yöntemleri geliştirilmiştir.

Bu bölümde sınıflandırma işlemi ile ilgili genel bilgi ve tanımlar verilecek, ardından boyut indirgeme yöntemlerinin iki yaklaşımı; özellik seçimi ve özellik çıkarımı metotları incelenecektir.

3.1 Sınıflandırma

Sınıflandırma, sınıf etiketleri belli bir eğitim kümesi yardımıyla yeni örneklerin ait oldukları kategorileri belirlemeyi amaçlayan bir öğrenme çeşididir. Söz konusu metinlerin sınıflandırılması olduğunda, eğitim kümesi sınıfları belirli belgelerden oluşur. Belgelerde bulunan kelimeler de örneklerin içerdiği özellikler olarak ele alınır. Belgelerin kelime içeriğine dayalı olarak yapılan sınıflandırmada Bölüm 2.3'te

bahsedilen vektör uzayı modeli ve TFIDF değerleri esas alınır. Bu yaklaşıma aynı zamanda kelime torbası (*bag-of-words*) modeli de denmektedir. Sayfalardaki özelliklerin değerlerinin belirlenmesi ve sayfalar arası benzerliklerin hesaplanması için de yine Bölüm 2.3.2'de bahsedilen terim ağırlıklandırma yöntemleri kullanılır.

3.1.1 Örün Sayfalarının Sınıflandırılması Çalışmaları

Örün sayfalarının sınıflandırmasında da her bir sayfa bir belge olarak ele alınır, sayfanın içeriğinden elde edilen anahtar kelimeler de terimleri oluşturur. Örün sayfalarının sınıflandırılması için yapılan çalışmalar internetin yaygınlaştığı ve örün sayfalarını sayılarının arttığı ilk yıllarda daha çok yapılmıştır.

K en yakın komşu [50] [51] [52], Destek vektör makineleri [53], Naive Bayes [54] [55], yapay sinir ağları [56] [57], doğrusal en küçük kareler yöntemi [58] gibi birçok algoritma metin sınıflandırma işlemlerinde denenmiş ve kullanılmıştır.

Fürnkranz, WebKB veri kümesi üzerinde, metindeki bağlantılar ve bağlantılara yakın metinleri kullanarak hedef örün sayfasının sınıfını bulmak üzere bir çalışma yapmış, bu sayede metin sınıflandırma başarısında artış elde etmiştir [59]. Ancak burada hedef sayfaya verilen bağlantıları göz önüne almış, sayfadaki metin bilgisinden faydalanmamıştır.

Joachims ve arkadaşları destek vektör makinelerinde değişik çekirdek fonksiyonları kullanarak WebKB veri kümesi üzerinde bir çalışma yapmışlardır [60]. Bir çekirdek fonksiyonunu belgedeki terimler için, başka bir çekirdek fonksiyonunu da sayfalardaki bağlantılar için kullanarak daha iyi sonuçlar elde ettiklerini raporlamışlardır.

Dumais ve Chen LookSmart örün klasöründeki girdiler üzerinde destek vektör makineleri kullanarak hiyerarşik sınıflandırma fikrini uygulamışlardır [61]. Mladenic, Yahoo örün klasöründeki sayfalar üzerinde Naive Bayes metodunu uygulamıştır [62]. Holden ve Freitas, örün kümesi üzerinde karınca kolonisi algoritması ile sınıflandırma denemişlerdir [63]. Literatürde metin sınıflandırma ve özellikle örün sayfalarının sınıflandırılması konularında bunlara benzer birçok çalışma bulmak mümkündür.

3.2 Boyut İndirgeme

Vektör uzayı yaklaşımı çok yüksek boyutlu ve seyrek bir özellik uzayı oluşturur. Bu özelliklere sahip bir özellik uzayında çalışmak ise sınıflandırma algoritmalarının efektifliğini düşürür, işlem zamanının ve ihtiyaç duyulan çalışma alanı ve belleğin artmasına sebep olur. Kategorizasyon işlemine geçmeden önce veri kümesinde yer alan örneklerin özelliklerinin boyutlarını indirgemek ise bu problemlerin çözümüne yardımcı olabilir. Özelliklerin boyutlarını indirgemek üzere temel olarak iki yaklaşım bulunmaktadır:

- Özellik seçimi (*feature selection*)
- Özellik çıkarımı (*feature extraction*).

3.2.1 Özellik Seçimi

Özellik seçimi ile özelliklerin tümü değil, seçilen bir alt kümesi sınıflandırma için kullanılır. Bu da daha düşük boyutlu bir uzayda çalışmak anlamına gelir. Özellik seçimi için iki genel yaklaşım bulunmaktadır. Sarmalayıcı (*wrapper*) yöntemler olarak bilinen özellik seçim metotları sınıflandırıcılar üzerinde denemeler yaparak özelliklerin en iyi alt kümesini elde etmeye çalışır [51]. İkinci yaklaşıma ait yöntemler ise filtre yöntemleri olarak bilinir ve özellikleri dizme yöntemleri kullanarak belirli bir eşik değerinin üzerinde değer alan özellikleri seçerler. Hangi yaklaşımla çalışırsa çalışsın, ayırt ediciliği iyi olan bir özellik alt kümesi elde edildiğinde kategorizasyon işlemlerinin maliyeti düşer ve hassasiyeti artar.

En bilinen ve en çok kullanılan özellik seçimi algoritmaları arasında belge frekansı (*document frequency*), chi istatistiği (*chi statistic*), bilgi kazanımı (*information gain*), terim dayanıklılığı (*term strength*) ve karşılıklı bilgi (*mutual information*) yer alır [67]. Sayılan yöntemlerin hepsi filtre yöntemlerdir ve özellikleri birbiriyle benzeşen entropi temelli değer hesabına göre süzerler. Ayrıca, yine popüler metotlar olan chi kare (chi-square) ve korelâsyon katsayısı (correlation coefficient) metotlarının belge frekansından daha iyi sonuçlar verdiği de ispatlanmıştır [68].

Şimdiye kadar tanıtılan yöntemlerin hepsi doğrusaldır. Doğrusal olmayan özellik seçimi yöntemlerine örnek olarak *Relief* ve *doğrusal olmayan çekirdek çarpımsal artımları*

(*nonlinear kernel multiplicative updates*) yöntemleri verilebilir [69]. Doğrusal olmayan özellik çıkarım yöntemlerini gerçeklemek karmaşık olduğu ya da fazlaca işlem gücü gerektirdiği için, metin işleme alanında doğrusal yöntemler kadar sık kullanılmazlar.

Özellik seçimi yöntemlerinin kötü yanı ise seçim işleminin belirli bir sınıflandırıcı üzerinde denenerek veya belirli bir dizme yöntemi kullanılarak yapılmasıdır. Bu yüzden seçim yapılan sınıflandırıcı ile elde edilen özellik alt kümesi, başka sınıflandırıcılar için uygun olmayabilir ve performansı arttırmayabilir. Aynı şekilde uygulanan filtre sonucu elde edilen azaltılmış özellikler, belirli kategorizasyon işlemlerinin performansına kötü etki edebilir.

Bu bölümün alt bölümlerinde en bilinen ve en yaygın olarak kullanılan özellik seçim yöntemleri olan karşılıklı bilgi, chi kare ve korelasyon katsayısı yöntemlerinden bahsedilecektir. Tanıtılacak olan yöntemler aynı zamanda tezde önerilen yöntemin performansını karşılaştırmak üzere de kullanılacaktır.

3.2.1.1 Karşılıklı Bilgi Özellik Seçim Yöntemi

Sınıf etiketleri belirli bir veri kümesinde, bir sınıfın örneklerinde bulunan terimlerin, sınıf hakkında tanımlayıcı değer içermeleri beklenir. Bu tanımlayıcı değerler karşılıklı bilgi değeri ile bulunabilir. İki ayrık rasgele değişken arasındaki karşılıklı bilgi 3.1 ile hesaplanır. Burada *p(x,y)*; *x* ve *y* değişkenlerinin birleşik olasılık yoğunluk fonksiyonu, $p_1(x)$ ve $p_2(y)$ ise *x* ve *y* değişkenlerinin bileşen olasılığını göstermektedir.

$$I(X,Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p_1(x)\, p_2(y)} \right) \tag{3.1}$$

Karşılıklı bilgi değerini hesaplayacağımız iki bağımsız değişkeni *T* ve *C* olarak seçelim. Seçilen bir belge içerisinde *t* terimi varsa *T=1*, yoksa *T=0* değeri alacaktır. Benzer şekilde seçilen belge *c* sınıfına aitse *C=1*, değilse *C=0* değeri alır. Bir *t* terimi ile *c* sınıfı arasındaki karşılıklı bilgi, 3.2 ile hesaplanabilir.

$$MI(T,C) = \sum_{T \in \{1,0\}} \sum_{C \in \{1,0\}} p(T,C) \log\left(\frac{p(T,C)}{p_1(T)\, p_2(C)} \right) \tag{3.2}$$

Veri kümesindeki tüm terimler için karşılıklı bilgi değerleri hesaplandıktan sonra, en yüksek değere sahip olan *k* adet terim seçilip diğerleri veri kümesinden çıkarılırsa, karşılıklı bilgi özellik seçimi uygulanmış olur. Sonuçta veri kümesindeki örnekler en yüksek karşılıklı bilgi değerine sahip *k* adet özellik ile temsil edilir.

3.2.1.2 Chi Kare Özellik Seçim Yöntemi

Bu yöntemde özelliklerin sınıflara göre chi kare istatistikleri hesaplanır. Bu istatistiklere bakılarak her özellik teker teker değerlendirilir [70]. Bir sınıfta yer alan bir özellik için hesaplanan chi kare skoru, o terim ile o sınıf arasındaki bağımlılığı ölçmektedir. Eğer özellik sınıftan bağımsızsa, skoru sıfır olur. Yüksek bir chi kare skoruna sahip olan özellik, sınıf için daha tanımlayıcıdır.

N adet belgeden oluşan bir veri kümesinde, c_i sınıfında yer alan *t* terimi için χ^2 chi kare skoru Denklem 3.3 ile hesaplanır [71]. 3.3'deki ikili olasılıkların açıklaması Çizelge 3.1'de verilmiştir.

$$\chi^2(t,c_i)=\frac{N\left[P(t,c_i)P(\bar{t},\bar{c}_i)-P(t,\bar{c}_i)P(\bar{t},c_i)\right]^2}{P(t)P(\bar{t})P(c_i)P(\bar{c}_i)} \tag{3.3}$$

Çizelge 3.1 Chi kare ve korelasyon katsayısı yöntemlerinde yer alan ikili olasılıkların açıklamaları

	c_i Sınıfında Bulunma	**c_i Sınıfında Bulunmama**
***t* Terimi Var**	(t,c_i)	$(t,\bar{c}_i)$
***t* Terimi Yok**	$(\bar{t},c_i)$	$(\bar{t},\bar{c}_i)$

Veri kümesindeki tüm terimler için hesaplanan chi kare istatistik değerleri en yüksek olan *k* adet terim seçilerek diğerleri atıldığında, veri kümesi seçilen *k* adet terim ile ifade edilmiş olur.

3.2.1.3 Korelasyon Katsayısı Özellik Seçim Yöntemi

Bu yöntem chi kare yönteminin $CC^2 = \chi^2$ olduğu bir varyantıdır. Korelasyon katsayısı yöntemi, bir özellik alt kümesinin değerini her terimin tekil kestirim kabiliyeti yanında aralarındaki artıklığın derecesini de göz önünde bulundurarak ölçer (Hall, 1999). Tercih edilen özellik alt kümesi sınıf içi korelasyonu yüksek, sınıflar arası korelasyonu düşük tutmalıdır.

N adet belgeden oluşan bir veri kümesinde, c_i sınıfında yer alan *t* terimi için korelasyon katsayısı $cc(t,c_i)$ 3.4 ile hesaplanır [71]. 3.4'teki ikili olasılıkların açıklaması Çizelge 3.1'de verilmiştir.

$$cc(t,c_i) = \frac{\sqrt{N}\left[P(t,c_i)P(\bar{t},\bar{c}_i) - P(t,\bar{c}_i)P(\bar{t},c_i)\right]}{\sqrt{P(t)P(\bar{t})P(c_i)P(\bar{c}_i)}} \tag{3.4}$$

Yöntem ile, istenen *k* adet özellikten oluşan ve korelasyon katsayısı en yüksek olan özellik alt kümesi belirlenir. Bu alt kümede yer alan özellikler dışında kalan özelliklerin veri kümesinden çıkarılmasıyla özellik seçimi tamamlanmış olur.

3.2.2 Özellik Çıkarımı

Özelik çıkarımı yöntemleri çok boyutlu bir veri kümesini tanımlamak için gerekli kaynakları basitleştirmek için kullanılır. Özellikle metin kategorizasyon işlemleri göz önüne alındığında, vektör uzayı modelinin oluşturduğu yüksek boyutlu ve seyrek yapının gerektirdiği yüksek işlem gücü ve hafıza miktarını azaltmak, özellik çıkarım yöntemlerinin temel amacı halini alır.

Özellik çıkarımı yöntemlerinin hedefi, terimleri kaynaştırarak veri için yeterli hassasiyete sahip yeni bir tanım oluşturmaktır. Bu hedefe ulaşmak için de çok boyutlu özellik uzayının yeni ve daha düşük boyutlu bir hiper uzaya olan iz düşümünü alır. Özellik seçimi yöntemlerinden farklı olarak, çıkarılan özellikler orijinal özelliklerden değildir. Tamamen yeni ve bağımsız bir uzayda yer alırlar.

En bilinen ve en çok kullanılan özellik çıkarımı yöntemleri temel bileşen analizi (*principal component analysis*, PCA) ve saklı anlamsal çözümlemedir (*latent semantic analysis*, LSA). Günümüzde LSA özellikle metin madenciliği uygulamalarında sıklıkla

kullanılan bir yöntem olmuştur. Bu iki yöntem dışında öğreticili bir özellik çıkarım yöntemi olan doğrusal ayırtaç analizi (*linear discriminant analysis*, LDA) de çok kullanılan yöntemlerdendir.

Bu bölümün alt bölümlerinde bilinen ve yaygın olarak kullanılan özellik çıkarım yöntemleri olan PCA, LSA ve LDA'dan bahsedilecektir. Tanıtılacak olan yöntemler aynı zamanda tezde önerilen yöntemin performansını karşılaştırmak üzere de kullanılacaktır. Yöntemlerin tanıtımına geçmeden önce ise PCA ve LSA'da ortak olarak kullanılan tekil değer çözümlemesi (*singular value decomposition,* SVD) işleminin nasıl gerçekleştirildiğinden bahsedilecektir.

3.2.2.1 Tekil Değer Çözümlemesi

Tekil değer çözümlemesi tek başına bir boyut indirgeme yöntemi değildir. Lineer matris cebri işlemleri ile bir matrisin üç farklı matrisin çarpımı şeklinde ifade edilmesini sağlar.

A, *m*x*n* boyutunda (*m*≥*n* olmak üzere) bir matris olsun. 3.5'te gösterildiği gibi *A* matrisini üç matrise ayırabiliriz. Bu matrislerden ilki, kolonları sol tekil vektörler olarak adlandırılan *m*x*n* boyutlu kolon dikey birim matris *U*'dur ($U^TU=I$). İkinci matris, köşegenlerinde negatif olmayan değerlerin büyükten küçüğe sıralanmış bir şekilde dizildiği *m*x*n* boyutlu köşegen matris *Λ* olarak tanımlanır. Üçüncü matris ise kolonları sağ tekil vektörler (right singular vectors) olarak adlandırılan *n*x*n* boyutlu birim dikey matris *V*'dir. Bu çözümlemeye tekil değer çözümlemesi adı verilmektedir.

$$A = U \wedge V^T \tag{3.5}$$

Çözümlemeyi, bileşenlerini tanımlayarak ispatlayabiliriz. Eğer *A*, *m*x*n* boyutunda bir matris ise A^TA, *n*x*n* boyutunda simetrik matris olur. Bu yüzden özdeğerleri reeldir ve *A* matrisini köşegenleştirebilen bir dik *V* matrisine sahiptir. Ayrıca, tüm özdeğerleri pozitiftir. *λ*, A^TA'nın bir özdeğeri ve *x* de *λ*'ya ait olan özvektör olsun. 3.6 tanımlandığında, 3.7 doğru olur. Bu da özdeğerlerin pozitif olduğunu ispatlar.

$$\|Ax\|^2 = x^T A^T Ax = \lambda x^T x = \lambda \|x\|^2 \tag{3.6}$$

$$\lambda = \frac{\|Ax\|^2}{\|x\|^2} \geq 0 \tag{3.7}$$

Eğer A^TA'nın özdeğerlerini sıralar ve bunlara ait olan özvektörleri V matrisi olarak tanımlarsak, tekil değerleri 3.8 ile ifade edebiliriz.

$$\sigma_j = \sqrt{\lambda_j}, j = 1,...,n \tag{3.8}$$

Eğer A'nın rankı r ise, A^TA'nın rankı da r olur. A^TA simetrik matris olduğu için, rankı sıfırdan farklı pozitif özdeğerlerin sayısına eşittir. Bu eşitlikse $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ ve $\sigma_{r+1}=\sigma_{r+2}=\ldots=\sigma_n=0$ olduğunu ispatlar. V matrisini $V_1=(v_1,v_2,\ldots,v_r)$, $V_2=(v_{r+1}, v_{r+2},\ldots,v_n)$ olarak ve Λ_1 matrisini rxr köşegen matris olarak alırsak, Λ ve A matrislerini 3.9 ve 3.10 ile ifade edebiliriz.

$$\Lambda = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.9}$$

$$\begin{aligned} I &= VV^T = V_1V_1^T + V_2V_2^T \\ A &= AI = AV_1V_1^T + AV_2V_2^T = AV_1V_1^T \end{aligned} \tag{3.10}$$

Şimdi $AV=U\Lambda$ olduğunu gösterelim. İlk r sütun için $Av_j=\sigma_j u_j$ tanımını yapıp $U_1=(u_1,u_2,\ldots,u_r)$, $AV_1=U_1\Lambda_1$ eşitliklerini tanımlayabiliriz. Geriye kalan m-r adet birim dikey kolon vektörleri de $U_2=(u_{r+1}, u_{r+2},\ldots,u_m)$ olarak tanımlanabilir. $U=(U_1\ U_2)$ olduğu için, 3.5'i 3.11'deki gibi yeniden yazabiliriz. 3.12'de verilen 3.11'in çözümü, $A=U\Lambda V^T$ olduğunu kanıtlamaktadır.

$$U\Lambda V^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Lambda_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \tag{3.11}$$

$$\begin{aligned} U\Lambda V^T &= U_1\Lambda_1 V_1^T \\ &= AV_1V_1^T \\ &= A \end{aligned} \tag{3.12}$$

Sıradaki iki alt bölümde tanımlanan yöntemler temel olarak tekil değer çözümlemesini kullanmaktadır. Aradaki farkı ise çözümlenen girdi matrisi yaratmaktadır. İlgili bölümlerde daha detaylı olarak açıklanacağı gibi; tekil değer çözümlemesini PCA kovaryans matrisine, LSA ise terim-belge matrisine uygulamaktadır. Bu durumda eğer terim-belge matrisinin değerleri ortalanırsa, iki yöntemin de gerçekleştirdiği çözümleme birbirinin aynısı olur.

3.2.2.2 Temel Bileşen Analizi

Temel bileşen analizi, orijinal değişkenleri, temel bileşen adı verilen ve daha az sayıda olan ilişkili değişkene dönüştürür. Pearson tarafından bulunan yöntem, genellikle araştırma amaçlı veri analizi için kullanılır [72].

PCA veri kümesinin varyansını en iyi yansıtan ve bilginin önemli yönlerini gösteren özellikler olan alt düzey temel bileşenlerin karakteristiğini koruyarak özellik çıkarımı yapar. Bu da özellik kümesinin yeni bir hiper uzaya iz düşümünün özdeğerleri (*eigen values*) ve özvektörleri (*eigen vectors*) kullanılarak alınması ile gerçekleşir. Birinci temel bileşen en yüksek varyansa sahip özelliklerin doğrusal kombinasyonundan oluşur. Bu aynı zamanda en yüksek öz değere sahip olan özvektördür. İkinci temel bileşenin varyansı daha düşüktür ve birinci temel bileşen ile dik açı yapar. Esasen orijinal özellik sayısı kadar temel bileşen mevcuttur ve bunlar en yüksek özdeğerliden en düşük özdeğerliye doğru sıralanırlar. Veri kümesinin en önemli karakteristiklerini koruyacak şekilde, varyansın %95'ini kapsayacak kadar temel bileşen kullanılması ile boyut indirgeme sağlanır.

Temel bileşenler, tekil değer çözümlemesi (*singular value decomposition*, SVD) ile bulunur. *X*, *mxn* boyutlu veri kümesi matrisi olsun. İlk olarak kovaryans matrisi Σ elde edilir. SVD ile Σ 'in özvektörler ve özdeğerleri elde edilir. Tekrar vurgulayacak olursak, PCA yönteminde tekil değer çözümlemesi kovaryans matrisi (Σ) üzerinde gerçekleştirilmektedir. Bu değerler, 3.13'te verilmiş olan *pxp* boyutlu dik açılı özvektörlerin oluşturduğu *U* matrisinde ifade edilir. 3.14 ile *pxn* boyutundaki *S* matrisinden hedeflenen kesinliği sağlayacak varyansı kapsayacak şekilde ilk *p* adet vektörün seçilmesi ile boyut indirgenmesi sağlanmış olur.

$$\Sigma = U \Lambda U^T \tag{3.13}$$

$$S = U^T X \tag{3.14}$$

PCA şekil ve örüntü tanımada popüler bir teknik olsa da sınıf ayırımı için optimize edilmemiş olmasından dolayı sınıflandırmada kullanımı çok yaygın değildir [73]. Genellikle görüntü işleme uygulamalarında kullanılır.

3.2.2.3 Saklı Anlamsal Çözümleme

LSA bir belge kümesi ve içerdiği terimler üzerindeki ilişkileri analiz ederek bunlara ait bir kavram kümesi oluşturmaya yarayan, 1988'de patenti alınmış bir yöntemdir [5]. LSA da PCA gibi SVD ile belgeler arasındaki ilişkileri ortaya çıkarır.

İsminden de anlaşılacağı gibi tekil değer çözümlemesi, terim-belge matrisini daha küçük bileşenlere ayrıştırır. Vurgulamak gerekirse, PCA'den farklı olarak çözümleme terim-belge matrisi üzerinde gerçekleştirilir. LSA algoritması çözümlenen bileşenlerden birini değiştirerek boyutlarını küçültür, sonra orijinali ile aynı şekilli bir matriste bileşenleri tekrar birleştirir. Oluşan yeni matristeki değerler ilk matristekinden hayli farklıdır ve orijinal matrisi gerçeğe en uygun şekilde temsil eder.

LSA'da ilk olarak verilen *X* terim-belge matrisi SVD ile 3.15'de gösterilen daha küçük bileşenlere ayrıştırılır. Belgelerdeki terimler arası korelasyonları XX^T, terimler üzerinde belgelerin korelasyonunu da X^TX ile gösterilirse, 3.15'teki matrisler 3.16 ve 3.17 ile ifade edilebilir. Σ matrisinden *k* tekil değer ve *U* ile *V* matrislerinden bu değerlere karşılık gelen vektörler alınırsa, *X* için en az hata ile *k* rankından yakınsama elde edilmiş olur. Alınan bu yakınsama ile boyut indirgemesi sağlanmış olur. Eğer Σ, *U* ve *V*'yi tekrar birleştirilip 3.18'de gösterilen $\hat{X}$ oluşturulursa, tekrar terim-belge matrisiymiş gibi kullanılabilir.

$$X = U\Sigma V^T \tag{3.15}$$

$$XX^T = U\Sigma\Sigma^T U^T \tag{3.16}$$

$$X^T X = V\Sigma^T\Sigma V^T \tag{3.17}$$

$$\hat{X}_k = U_k\Sigma_k V_k^T \tag{3.18}$$

Saklı anlamsal çözümleme metin madenciliğinde en çok örün sayfalarının benzerliklerinin bulunmasında ve belge kümeleme işlemlerinde kullanılır. Ayrıca oylama ve bilgi filtreleme (*information filtering*) gibi yöntemler ile birlikte belge sınıflama için de kullanılabilir. Birçok algoritma daha basit bir girdi uzayı ile çalışmak için özellik çıkarımı yöntemi olarak saklı anlamsal çözümlemeyi uygulamaktadır.

3.2.2.4 Doğrusal Ayırtaç Çözümlemesi

Bundan önce tanıtılan iki özellik çıkarım yöntemi de eğiticisiz yöntemlerdir ve bu sebeple sınıf etiketi taşıyan bir veri kümesine ihtiyaç duymazlar. LDA ise bunlardan farklı olarak ayırtaçların bulunması için veri kümesindeki örneklerin sınıf bilgilerine de ihtiyaç duyar, bu sebeple de eğiticili bir yöntemdir. LDA sınıflar arası ayrımı belirleyecek ayırtaçlar oluşturmak için orijinal özelliklerin doğrusal bir bileşkesini oluşturur. Amaç sınıflar arası varyans ile sınıf içi varyans arasındaki oranı en yükseğe çıkarmaktır. *w* yönündeki sınıf ayrımı 3.19 ile hesaplanabilir. 3.19'daki Σ_B sınıflar arası dağılım matrisi 3.20 ile, Σ_w sınıf içi dağılım matrisi de 3.21 ile hesaplanır [74]. Bu denklemlerde μc, c sınıfının ortalaması, μ ise tüm sınıfların ortalamalarının ortalamasıdır.

$$S = \frac{w^T \Sigma_B w}{w^T \Sigma_W w} \tag{3.19}$$

$$\Sigma_B = \sum_c (\mu_c - \mu)(\mu_c - \mu)^T \tag{3.20}$$

$$\Sigma_W = \sum_c \sum_{i \in c} (x_i - \mu_c)(x_i - \mu_c)^T \tag{3.21}$$

LDA ile elde edilen dönüşüm 3.19'de verilen sınıf ayrımını maksimuma çıkartır. Eğer *w* $\Sigma_w^{-1}\ \Sigma_B$ matrisinin bir özvektörü ise, sınıf ayırtaçları özdeğerlere eşit olur ve doğrusal dönüşüm matrisi *U* 3.22 ile hesaplanabilir. U matrisinin kolonları $\Sigma_w^{-1}\ \Sigma_B$ matrisinin özvektörlerinden oluşur. 3.23'ün çözülmesi ile elde edilen özvektörler, özellikler arası değişimleri ifade eden bir vektör alt uzayı oluşturduğu için boyut indirgeme amacıyla kullanılabilir.

$$\begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_k \end{bmatrix} = \begin{bmatrix} u_1^T \\ u_2^T \\ \dots \\ u_k^T \end{bmatrix} (x - \mu) = U^T (x - \mu) \tag{3.22}$$

$$\Sigma_B u_k = \lambda_k \Sigma_w u_k \tag{3.23}$$

BÖLÜM 4

SOYUT ÖZELLİK ÇIKARIM YÖNTEMİ

Bir önceki bölümde sınıflandırma işleminde çok boyutluluğun doğurduğu problemlerin üstesinden gelmek ve başarımı arttırmak üzere özellik seçimi ya da özellik çıkarımı yoluyla boyut indirgeme işlemi uygulandığından bahsetmiş, ayrıca sık kullanılan ve en bilinen yöntemleri tanıtmıştık. Metin işleme alanında genellikle özellik seçimi yöntemlerinin kullanıldığını, özellik çıkarımı ile ilgili yapılan çalışmaların çok sınırlı olduğundan da Bölüm 1.1'de bahsedilmişti. Bu bölümde ise metin işleme alanı için geliştirdiğimiz özellik çıkarım yöntemini tanıtılacak, algoritması verilecek ve iki sınıflı küçük bir örnek veri kümesi üzerinde uygulaması yapılacaktır. Bu uygulama ile soyut özellik çıkarım yönteminin veri görselleştirme için de kullanılabileceği gösterilecektir.

4.1 Soyut Özellikler

Orijinal özelliklerin veri kümesindeki örneklerdeki dağılımları, örneklerin sınıflara ait olmasında etki sahibidir [75]. Bu noktadan hareketle özellik seçim yöntemleri, ayırt ediciliği en fazla olan özellikleri seçmeye, özellik çıkarım yöntemleri ise özelliklerin bulunduğu uzayı değiştirerek daha ayırt edici ve tanımlayıcı özellikler oluşturmaya çalışırlar.

Özelliklerin içerdiği ayırt edicilik değerleri, Bölüm 2.3'te incelenen ağırlıklandırmalara benzer bir çalışma ile ortaya çıkarılıp her bir sınıf için etki değerlerinin bileşkesi oluşturulsun. Bu bileşke ifadeler, örnekleri özelliklerin sınıflara olan etkilerinden doğan ayırt edicilikleri ile orantılı olacak şekilde yeni bir uzayda ifade eder. Bu yeni uzayda yer alan çıkarılan özelliklere, soyut özellikler (*abstract features*) adını veriyoruz. Soyut özellikler, orijinal özellikleri ağırlıklandırarak elde edilen etki değerlerini kullanıp,

boyutları sınıf sayısına eşit bir uzaya doğrusal olarak eşleme ile elde edilir. Bir örnek için çıkarılan soyut özellikler, örnekte bulunan orijinal özelliklerin her bir sınıfa olan etkisinin bileşkesi olacaktır. Bu sayede, soyut özellikleri çıkararak bir örnekte her bir sınıf için ne kadar kanıt ya da bilgi bulunduğu elde edilmiş olur.

Soyut özelliklerin oluşturulması için belirli bir dilde ya da tipte özellikler olması gerekmemektedir. Orijinal özellikler kelime, n-gram ya da cümle gibi istendiği şekilde seçilebilir. Bu sebeple soyut özellikler belirli bir dile bağlı olarak oluşturulmak zorunda değildir. Herhangi bir dilde hazırlanmış veri kümesi üzerinde, istenen orijinal özelliklerin bileşkesi alınarak, soyut özellikler elde edilebilir.

4.2 Soyut Özellik Çıkarım Algoritması

Soyut özellik çıkarım yöntemi (*abstract feature extraction,* AFE) özvektörleri ya da özdeğerleri kullanmadığı gibi tekil değer ayrıştırımı da yapmaz, bu yönüyle literatürde yer alan diğer özellik çıkarımı yöntemlerinden farklıdır. Bu yöntemde terimlerin ağırlıkları ve sınıflar üzerindeki olasılıksal dağılımları göz önüne alınmaktadır. Terim olasılıklarının sınıflara olan izdüşümünü alıp bu olasılıkları toplayarak, her terimin sınıfları ne kadar etkilediği bulunmaktadır. Veri kümesindeki kategori etiketleri işlemde kullanıldığı için, soyut özellik çıkarım yöntemi tıpkı LDA gibi eğiticili bir özellik çıkarım yöntemidir.

I adet terim, *J* adet belge ve *K* adet sınıf olduğu varsayılsın. $n_{i,j}$ t_i teriminin d_j belgesinde kaç kere geçtiğini göstersin. Metin işleme alanı söz konusu olduğu için $n_{i,j}$ aynı zamanda t_i teriminin d_j belgesindeki terim frekansıdır. J_i de veri kümesinde t_i terimine sahip olan belge sayısı olsun. Soyut özellik çıkarım yönteminin adımları aşağıda listelenmiştir:

1. t_i teriminin c_k sınıfında kaç kere geçtiğini gösteren $nc_{i,k}$ değerini 4.1 ile hesapla.

$$nc_{i,k} = \sum_{j} n_{i,j} \quad , \quad d_j \in c_k \tag{4.1}$$

2. d_j belgesinde yer alan t_i teriminin c_k sınıfını ne kadar etkilediğini 4.2 ile bulup ağırlıklandır.

$$w_{i,k} = \log(nc_{i,k} + 1) \times \log(\frac{J}{J_i}) \quad (4.2)$$

3. Tüm belgeler için tekrarla:

Belgedeki tüm terimlerin c_k sınıfına olan toplam etkisini 4.3 ile hesapla.

$$Y_{j,k} = \sum_i w_{i,k} \quad , \quad t_i \in d_j \quad (4.3)$$

4.Çıkarılan soyut özellikleri 4.4 ile normalize et.

$$AF_{j,k} = \frac{Y_{j,k}}{\sum_k Y_{j,k}} \quad (4.4)$$

Sonuçta, *I* adet terim *K* boyutlu bir uzaya yansıtılır. Bu işlem tüm belgeler için uygulandığında *J* satır (her belge için bir satır) ve *K* sütundan oluşan (çıkarılan özelliklerin sayısı sınıf sayısına eşittir) indirgenmiş sonuç matrisi elde edilir.

Terimleri 4.2 ile ağırlıklandırırken kullanılan logaritmik fonksiyon, toplanırlığı (*additivity*) ve türdeşliği (*homogeneity*) sağlamadığı için doğrusal olmayan bir fonksiyon türü olarak bilinir. Ancak doğrusal olmayan fonksiyon algoritmada direk olarak kullanılmayıp fonksiyon çıktıları veri kümesinden elde edilen sabit ağırlık değerleri olarak algoritmada kullanıldığından doğrusal olmayan fonksiyonla ağırlıklandırma soyut özellik çıkarım yönteminin doğrusallığını bozmamaktadır. Buna karşın 4.4 ile yapılan normalizasyon işlemi sonucunda doğrusallık bozulmaktadır. Bu sebeple soyut özellik çıkarım yöntemi yeni uzaya doğrusal eşleme yapsa da, sonuçta oluşan soyut özellikler doğrusal değildir.

Soyut özellik çıkarım yönteminin doğrusal olarak sınıf sayısı kadar boyutlu uzaya eşleme yaptığını matris cebiri ile de gözlemlemek mümkündür. *K* sınıflı, *J* belgeden oluşan ve toplamda *I* terim içeren bir veri kümesi verilmiş olsun. *J*x*I* boyutlu terim-belge matrisi *X*, 4.1 ve 4.2 ile ağırlıklandırılan terim-sınıf matrisi de *Y* olsun. Soyut özellik çıkarım yöntemi, X^TY matris çarpımını sütun normalizasyonu ile gerçekleştirerek

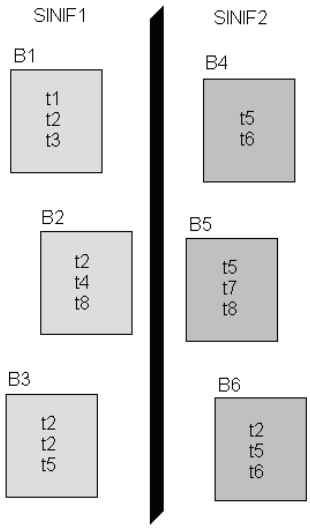
özellikleri çıkarır. Bu matris aynı zamanda terim-sınıf dağılım matrisidir. Vektör uzayı modelindeki *X* matrisi ile eğitim için XX^TY işlemi gerçekleştirilir. Bir v test belgesinin boyutlarını indirgemek için vX^TY işlemi uygulanır. Eğitim ve test işlemlerinde de matrislere sütun normalizasyonu uygulanır. Soyut özellik çıkarım yöntemini matris çarpımları şeklinde tanımlamak, *I* boyutlu bir uzaydan *K* boyutlu bir uzaya doğrusal bir eşleme yapıldığını daha açık bir şekilde göstermekte, son adımdaki normalizasyon işlemi de yöntemin doğrusal olmamasını sağlamaktadır.

Daha önceden de belirttiğimiz gibi, soyut özellik çıkarım yöntemi eğiticili bir yöntemdir ve etiketli veri kümeleri ile çalışmaktadır. SVD temelli yöntemler çıkarılan özellikleri özdeğerlere göre sıralayıp hedeflenen kesinliğe yetecek kadarını kullanır. Soyut özellik çıkarım yöntemi ise orijinal özelliklerin sınıflara olan bileşke etkisini ortaya çıkarmayı hedeflediğinden sınıf sayısı kadar özellik çıkarmaktadır. Özelliklerin etkilerinin dağılımı 4.1 ve 4.2 ile belirlendikten sonra, veri kümesindeki örnekler için soyut özellikleri 4.3 ve 4.4 ile bulunabilir. Bir belge için çıkarılan *K* adet AF_k soyut özellikleri, belgenin *K* adet sınıfa ait olma olasılıkları olarak da değerlendirilebilir.

4.3 Örnek Problem Üzerinde Uygulama

Bu bölümde soyut özellik çıkarım yönteminin küçük bir veri kümesi üzerinde örnek uygulaması yapılacaktır. Şekil 4.1'deki gibi, iki sınıfta toplam 6 belge ve 8 terimden oluşan küçük ve basit bir örnek veri kümesi verilsin. Veri kümesine ait olan terim-belge matrisi de Çizelge 4.1'de görülmektedir.

Terimlerin sınıflar üzerindeki ağırlıkları 4.2 ile hesaplanır. Bu işlem sonunda *I*x*K* (8x2) boyutlarında, Çizelge 4.2'de verilen matris elde edilir. Örnek olarak Terim5'in Sınıf1 ve Sınıf2 üzerindeki ağırlıklarını hesaplayalım. i=5, k=1 için 4.1 ve 4.2'yi kullandığımızda w5,1 = log(1+1) x log(6/4) = 0,053 değeri elde edilir. Aynı şekilde i=5, k=2 için 4.1 ve 4.2'yi kullanırsak w5,2 = log(1+3) x log(6/4) = 0,106 değerini buluruz.



Şekil 4.1 Örnek veri kümesi

4.3 ve 4.4 uygulandıktan sonra *JxK* (6x2) boyutlarındaki indirgenmiş matris elde edilir. İndirgenmiş sonuç matrisi Çizelge 4.3'te verilmiş, Şekil 4.2'de görselleştirilmiştir. Örnek olarak Belge4 için Sınıf1 ve Sınıf2 üzerindeki ağırlıkları hesaplayalım. *j*=4, *k*=1 için 4.3 ve 4.4'ü kullandığımızda Y_1 = 0,053 + 0 = 0,053 olarak bulunur. AF_1 = 0,053/(0,053+0+0,106+0,228) = 0,137 olarak hesaplanır. Aynı şekilde *j*=4, *k*=2 için 4.3 ve 4.4'ü kullanırsak Y_2 = 0,106 + 0,228 = 0,334 olarak bulunur. AF_2 = 0,334/(0,053+0+0,106+0,228) = 0,863 olarak bulunur.

Çizelge 4.1 Örnek veri kümesi için terim-belge matrisi

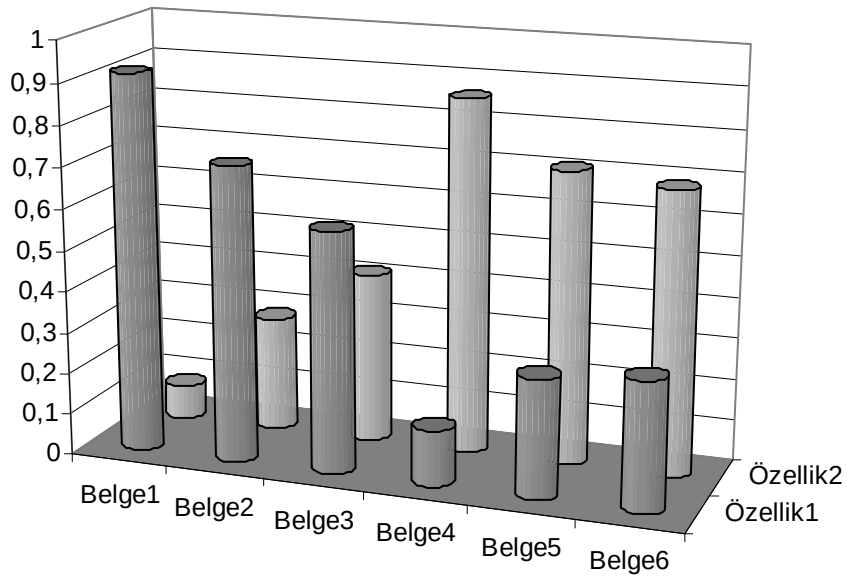
	Belge1	**Belge2**	***Belge3***	***Belge4***	***Belge5***	***Belge6***
Terim1	1	0	0	0	0	0
Terim2	1	1	2	0	0	1
Terim3	1	0	0	0	0	0
Terim4	0	1	0	0	0	0
Terim5	0	0	1	1	1	1
Terim6	0	0	0	1	0	1
Terim7	0	0	0	0	1	0
Terim8	0	1	0	0	1	0

Çizelge 4.2 Örnek veri kümesi için terimlerin sınıflar üzerindeki ağırlıkları ($w_{i,k}$)

	Sınıf1	**Sınıf2**
Terim1	0,234	0
Terim2	0,123	0,053
Terim3	0,234	0
Terim4	0,234	0
Terim5	0,053	0,106
Terim6	0	0,228
Terim7	0	0,234
Terim8	0,144	0,144

Çizelge 4.3 Örnek veri kümesi için belgelerin çıkarılan özellikleri ($AF_{j,k}$)

	Soyut Özellik 1	**Soyut Özellik 2**
Belge1	0,918	0,082
Belge2	0,718	0,282
Belge3	0,585	0,415
Belge4	0,137	0,863
Belge5	0,289	0,711
Belge6	0,313	0,687

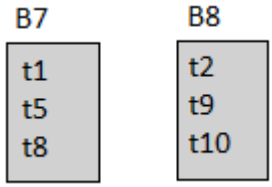


Şekil 4.2 Örnek veri kümesindeki soyut özelliklerin görselleştirilmiş hali

Şekil 4.2 incelendiğinde, çıkarılan özelliklerin aynı zamanda belgelerin sınıflara olan üyelik olasılıkları olarak da okumanın mümkün olduğu görülmektedir. Örneğin Belge 4'ün içeriğine bakıldığında, %13,7 olasılıkla birinci sınıfa, %86,3 olasılıkla da ikinci sınıfa ait olduğu anlaşılmaktadır. Buradan hareketle, önerilen özellik çıkarımı yönteminin aynı zamanda bir sınıflandırıcı olarak da kullanılmasının mümkün olduğu sonucuna ulaşılmaktadır.

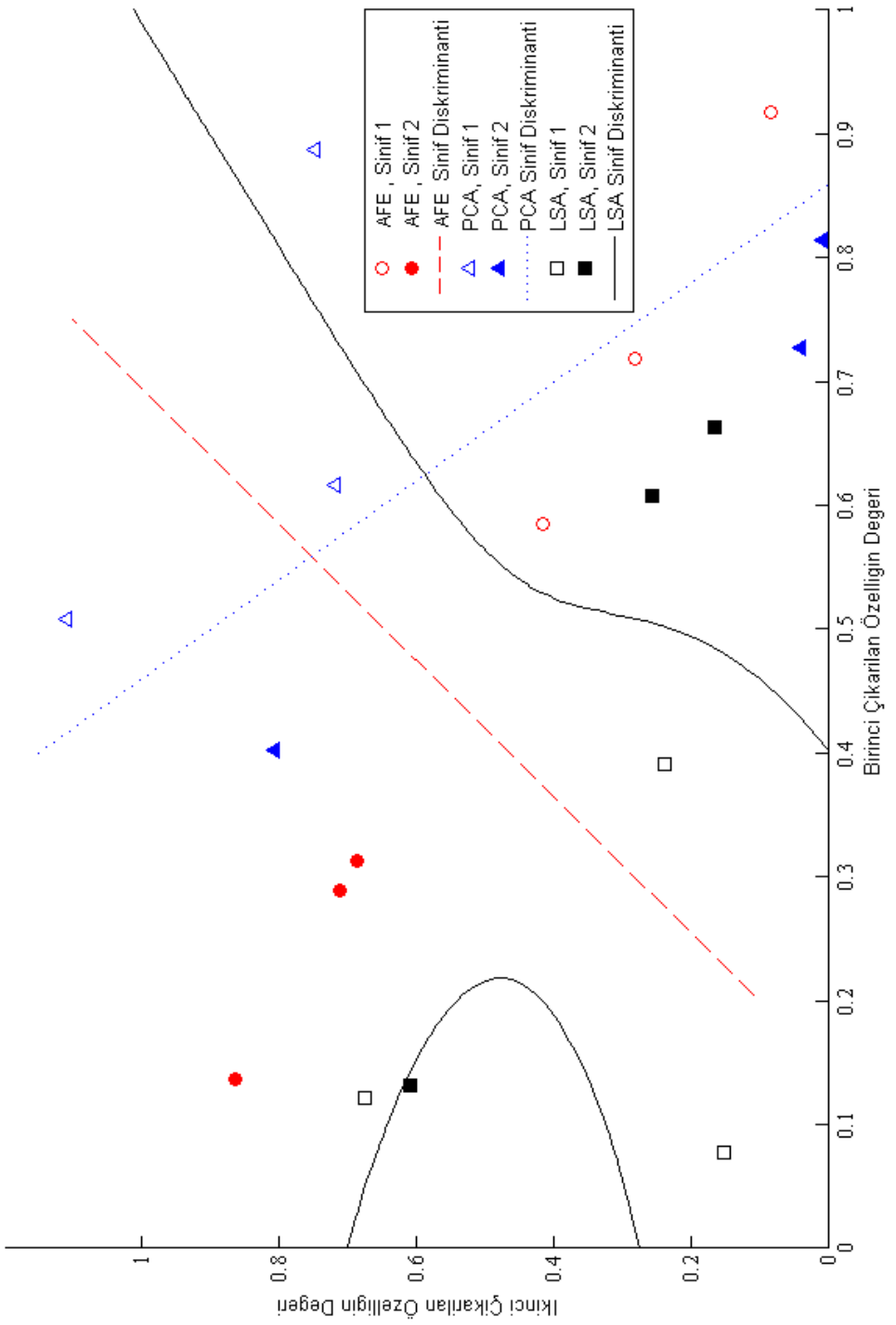
Yeni bir belge geldiğinde belgenin hangi sınıfa ait olacağı 4.3 ve 4.4'ten elde edilen sonuçlara göre belirlenebilir. Şekil 4.3'te verilen iki yeni belge geldiğini varsayalım. B7 belgesi eğitim kümesinde yer alan t1, t5 ve t8 terimlerini içermektedir. Bu terimlerin sınıflar üzerindeki ağırlıkları Çizelge 4.2'den alınır. 4.3 ile Y_1 = 0,234 + 0,053 + 0,144 = 0,431 ve Y_2 = 0 + 0,106 + 0,144 = 0,250 bulunur. 4.4 ile soyut özellikler hesaplandığında, B7 için AF_1 = 0,431 / 0,681 = 0,633 ve AF_2 = 0,250 / 0,681 = 0,367 değerleri elde edilir. B8 belgesi ise eğitim kümesinde yer alan t2 terimini ve yer almayan t9 ve t10 terimlerini içermektedir. Bu belge için sadece eğitim kümesinde yer alan t2 terimi kullanılarak soyut özellikler hesaplanır. Çizelge 4.2'den Y1 = 0,123 ve Y2 = 0,053 değerleri alınır. 4.4 ile B8 için AF1 = 0,123 / 1,176 = 0,699 ve AF2 = 0,053 / 0,176 = 0,301 değerleri bulunur. Eğitim kümesinde yer almayan özellikleri de hesaba katmak

istenirse, vektör uzayı modelinin doğası gereği yeni belgelerin de eğitim kümesine katılarak özellik değerlerinin tekrar hesaplanması gerekir.



Şekil 4.3 Örnek veri kümesi için test belgeleri

Soyut özelliklerin sınıfları ayırmadaki başarısını ölçmek üzere örnek veri kümesine PCA ve LSA yöntemleri de uygulanarak her yöntem ile ikişer özellik çıkarılmıştır. Soyut özellik çıkarım yöntemi, PCA ve LSA ile çıkarılan özellikler Şekil 4.4'te aynı düzlem üzerinde karşılaştırılmıştır. Örneklerin dağılımlarını incelendiğinde, soyut özelliklerin diğer yöntemlerle çıkarılan özelliklere kıyasla en kesin ve belirgin ayırımı sağladığı rahatlıkla görülebilir. PCA ile çıkarılan özellikler de doğrusal ayırım sağlasa da, soyut özellikler kadar temiz ve ayrık diskriminantlar oluşturamamaktadır. LSA ile boyutları indirgenen örnekler ise doğrusal olarak ayrıştırılamamaktadır. LSA'nın ikinci dereceden diskriminantının olması, sınıflandırma algoritmalarının başarımını karmaşıklığı nedeniyle düşürmektedir.



Şekil 4.4 Örnek veri kümesinin üç ayrı yöntem ile çıkarılan özelliklerle gösterimi ve elde edilen sınıf ayırtaçları

BÖLÜM 5

VERİ KÜMELERİ

Soyut özellik çıkarım yönteminin başarımını test etmek ve diğer yöntemlerle karşılaştırmak üzere metin tipinde veri kümeleri üzerinde sınıflandırma testleri gerçekleştirilmiştir. Bölüm 6'da detaylarıyla anlatılacak olan bu testlerde kullanılmak üzere üç ayrı veri kümesi hazırlanmış ve kullanılmıştır. Bu bölümde testler için kullanılan veri kümeleri ve hazırlanmaları için yapılan ön işleme adımları tanıtılacaktır.

Testlerde kullanılan ilk veri kümesi DMOZ örün dizininin "World/Türkçe" başlığı altında yer alan örün sayfalarının taranması ile oluşturulmuştur. Bu veri kümesi 12 kategori altında tamamıyla Türkçe örün sayfalarından oluşmaktadır.

İkinci veri kümesi metin işleme uygulamalarında standart olarak kullanılan Reuters-21578 veri kümesinden elde edilmiştir. Reuters-21578 veri kümesinin ayrıca ModApte-10 olarak bilinen versiyonu da testler için hazırlanarak kullanılmıştır. Hazırlanan Reuters veri kümesi 21 kategori, ModApte veri kümesi ise 10 kategoriden oluşmaktadır.

Üçüncü veri kümesi ise yine metin işleme uygulamalarının testlerinde standart olarak kullanılan ve 20 kategoriden oluşan 20-Newsgroups veri kümesidir. Bu bölümün alt bölümlerinde her bir veri kümesinin genel özellikleri tanıtılacak, hazırlık için uygulanan ön işleme adımları açıklanacak ve testlerde kullanılan veri kümelerinin detayları anlatılacaktır.

5.1 Ön İşleme Adımları

Metin tipinde bir veri kümesini yöntemleri karşılaştırmak üzere yapılan testlerde olduğu gibi kullanmak neredeyse imkânsızdır. Öncelikle veri kümesi üzerinde çeşitli ön işleme adımları gerçekleştirilerek yöntem ve algoritmaların girdilerine ve beklentilerine uygun formata getirilmeleri gerekmektedir. Bu amaçla veri kümesinde yer alan örnekler terimlere ayırma, filtreleme, gövdeleme ve aykırı değerlerin ayıklanması gibi çeşitli işlemlere tabi tutulur. Bu alt bölümde ön işleme adımlarının neler olduğu ve nasıl gerçekleştirildikleri açıklanacaktır.

5.1.1 Belirtkeleme

Metin tipindeki verinin sınıflama ve kümeleme algoritmalarınca işlenebilmesi için öncelikle özellik olabilecek atomik parçalara ayrılmış olması gerekir. Metni atomik parçalara ayırma işlemine belirtkeleme (*tokenization*) adı verilir. Uygulamanın tipine göre belirtkeçler cümle, kelime öbeği, kelime, n-gram, hece ya da harf olabilir. Belirtkecin ne olacağı ve nasıl seçileceği tamamen uygulamaya bağlıdır. Sınıflama ve kümeleme çalışmalarında metin belirtkeci olarak genellikle kelime ve n-gramlar kullanılır. Bu çalışmada veri kümelerindeki metinler kelimelerine ayırıştırılıp, terim olarak kullanılmıştır.

5.1.2 Filtreleme

Bir veri kümesinde bazı terimler çok sayıda belgede geçiyor ve fazlaca tekrarlanıyor olabilir. Bu terimler sınıflandırma veya kümeleme performansını düşürürler. Metin işleme işlemlerine başlanmadan önce sık geçen terimlerin bulunması ve veri kümesindeki belgelerden temizlenmesi işlemine filtreleme adı verilir. Filtrelemeye sık kullanılan sözcüklerin elenmesi (*stopword removal*) de denmektedir.

Veri kümesinde bulunan sık kullanılan sözcüklerin elenmesi için standart bir yöntem bulunmamaktadır. Çalışılacak veri kümesinde yapılacak bir çalışma ile geçtiği belge sayısı belirli bir değerin üzerinde olan terimler, sık kullanılan sözcük olarak

değerlendirilip elenebilir. Buradaki filtre limitinin ne olacağı yine tamamen kullanıcıya bağlı bir parametredir.

Bir dilde yazılmış ve metin işleme uygulamalarında test için kullanılan çeşitli veri kümeleri üzerinde yapılan çalışmalarla, o dilde ortalama olarak en sık kullanılan sözcükler belirlenebilir. Bu çalışmada İngilizce ve Türkçe için derlenmiş olan sık kullanılan sözcükler listesinden yararlanılmıştır. Veri kümelerinden filtrelenen sık kullanılan kelimelerin listesi EK-A'da verilmiştir.

5.1.3 Gövdeleme

Gövdeleme (*stemming*) işlemi ile kelimelerin biçimbirimsel (*morphological*) analizi gerçekleştirilir. Kelimelerin eklerinin ayrılması ve köklerinin bulunmasını sağlar.

İngilizce ön ekli (*prefix*) bir dil olduğu için kelimelerin köklerine ulaşmak kolaydır ve soyut makineler ile tasarlanabilir. Porter'ın gövdeleme algoritması [76] İngilizce için kök bulma işlemini sorunsuzca yerine getirmektedir ve neredeyse bir standart olarak İngilizce gövdeleme işlemlerinde kullanılmaktadır. Türkçe gibi sondan eklemeli (suffix) dillerde ise gövdeleme işlemi başlı başına bir sorun teşkil etmektedir. Gerek çekim eklerinin fazlalığı, gerekse yapım eklerinin üretkenlikleriyle yeni anlamlar üretebilmeleri dolayısıyla kelimelerin gerçek köklerinin bulunması hala tam olarak çözülebilmiş değildir. Türkçe için biçimbirimsel çözümleme yapabilen Zemberek isimli uygulama paketi [77] gövdeleme için en yaygın kullanılan araçtır.

Bu çalışmada İngilizce veri kümelerindeki örneklerin kelimelerini gövdelemek için Porter'ın gövdeleme algoritması, Türkçe veri kümelerindekiler için ise Zemberek paketi kullanılmıştır.

5.1.4 Kutu Çizimi Filtresi

Kutu çizimi (*Box-Plot*) filtresi bir veri kümesinin dağılımını göstermek ve aykırı değerleri kolayca yakalamak için kullanılan bir yöntemdir [78]. Daha gelişmiş olan histogram ve

çekirdek yoğunlukölçer (*kernel density estimator*) yöntemlerinden daha basit olsa da, daha hızlı ve pratiktir.

Standart kutu çiziminde veri kümesinin dağılımı beş sayısal özellik ile temsil edilir:

- Veri kümesinde gözlenen en küçük değerli örnek
- Alt çeyrek (Lower Quartile)
- Medyan
- Üst çeyrek (Upper Quartile)
- Veri kümesinde gözlenen en büyük değerli örnek

Bu beş sayısal özellik noktası, verinin üzerinde çizilerek gösterilir. Özellikler arasındaki mesafeler verinin ne kadar dağınık olduğunu gösterir. Normal kutu filtresinde alt ve üst çeyrek çizgileri verinin %25 ve %75'lik kısmını belirler. Bunun dışında kalan veriler aykırı değer olarak kabul edilir ve temizlenir. Ancak uygulamalara ve veri kümesinin yapısına bağlı olarak alt ve üst çeyrek sınırları çeşitli şekillerde seçilebilir [79]:

- Tüm verinin en küçük ve en büyük değeri
- Medyandan 1,5 çeyrek değer kadar aşağısı ve yukarısı
- Ortalamadan k kat standart sapma kadar aşağısı ve yukarısı
- Verinin alttan %9'luk kısmı ile üstten %91'lik kısmı
- Verinin alttan %2'lik kısmı ile üstten %98'lik kısmı

Bu çalışmada kutu çizimi filtresi DMOZ veri kümesi hazırlanırken çok fazla ya da çok az sayıda terim içeren örnekleri elemek, Reuters veri kümesi hazırlanırken de içerdiği örnek sayısı belirli sınırlar arasında yer alan sınıfları seçmek için kullanılmıştır.

5.2 DMOZ Veri Kümesi

Metin işleme alanında Türkçe standart bir veri kümesi bulunmamaktadır. Bu sebeple Türkçe örün sayfalarından oluşan bir koleksiyon hazırlanarak soyut özellik çıkarım yöntemini test etmek üzere kullanılmıştır.

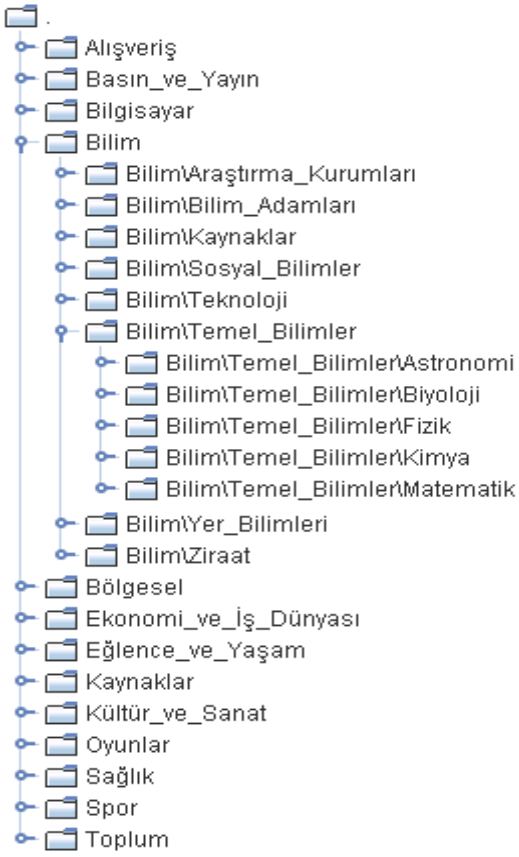
Açık Dizin Projesi (*Directory Mozilla*, DMOZ) öründeki en büyük ve en kapsamlı, insanlar tarafından düzenlenen dizindir. Dünyanın her tarafından katılımda bulunan geniş bir gönüllü editörler topluluğu tarafından inşa edilmiştir ve varlığı onlar tarafından sürdürülmektedir. DMOZ, Açık Dizin Projesi (Open Directory Project) adıyla da anılmaktadır [80].

Açık dizin, insanlar tarafından sınıflandırılmış örün içeriğini barındıran en geniş kapsamlı veritabanıdır. İnternet kullanıcılarından oluşan editör kadrosu, örün kaynaklarının keşfini sağlayan kolektif beyni oluşturur. Açık dizin, en büyük ve en popüler arama motorlarının dizin hizmetlerini de güçlendirmektedir. Bunlara Google, Lycos, AOL Search, HotBot, DirectHit ve daha yüzlercesi dâhildir.

Açık dizinin yapısı bir ağaç şeklindedir. Bu ağaçta temel konular en üst düğümleri, o konulara ait alt konular ve başlıklar ise alt düğümleri oluşturur. Her düğümde o kategori ile ilgili örün sayfalarının adresleri yer alır. Şekil 5.1'de açık dizinin "World/Türkçe" başlığında yer alan ana düğümleri verilmiştir. Bunların dışında büyüklere yönelik içeriği barındıran "Adult", çocuklara özel içeriği sınıflayan "Kids and Teens", Dünya üzerindeki coğrafi bölgelere göre sayfaları sınıflayan "Regional", değişik dillerdeki sayfaları sınıflayan "World" kategorileri de bulunmaktadır. Alt kategoriler de kendi içlerinde alt kategoriler barındırabilir. Buna bir örnek olarak Şekil 5.2'de "News" dizini gösterilmiştir.

Mayıs 2011 tarihi itibarıyla açık dizinde 4,8 milyondan fazla sayfa sınıflandırılmıştır. Tüm alt sınıflarla birlikte toplam kategori sayısı ise 1.000.000'dan fazladır. Tüm bu sınıflandırma işlemleri gönüllülük esasına göre çalışan 91.000'den fazla editör sayesinde yapılmaktadır. İnsan faktörü gözlem gücünü arttırmakla birlikte, çalışma hızını çok büyük oranda düşürmektedir. Tezde tasarlanan özellik çıkarım yöntemi ile örün sayfalarının otomatik sınıflandırılması yapılarak, sadece insan emeğine dayalı yapıyı bir nebze olsun rahatlatmak hedeflenebilir.

Şekil 5.1 Açık dizin Türkçe bölümü ana kategorileri



Şekil 5.2 Açık dizinde alt kategorilerin bir örneği

Bölüm 2.2'de detaylarıyla açıklandığı üzere, örün sayfalarını gezmek ve içeriklerini almak için örün robotları kullanılır. Soyut özellik çıkarım yönteminin eğitimi aşamasında sabit bir veri kümesi ile çalışılması gerekliliğinden, sürekli yeni sayfaları bulan bir örün

robotuna ihtiyaç yoktur. Kullanılan örün veritabanını oluşturmak için daha basit ve pratik bir çözüm olacak şekilde bir robot tasarlanmış ve gerçeklenmiştir.

Gezilecek adresler olarak DMOZ veritabanının "World/Türkçe" başlığı altında yer alan, Türkçe örün sayfalarının adres ve tanımlarının bulunduğu kayıtlar seçilmiştir. Bu kayıtlar geliştirilen robot ile gezilerek kategorilerine göre kaydedilmiştir. Örün tarama işlemi sonucunca Şekil 5.3'te görülen klasör yapısına sahip veri toplanmıştır. Toplam 13 kategoride 17.885 örün sayfası gezilerek kayıt edilmiştir. Kayıt edilen toplam veri 831 MB boyutundadır.

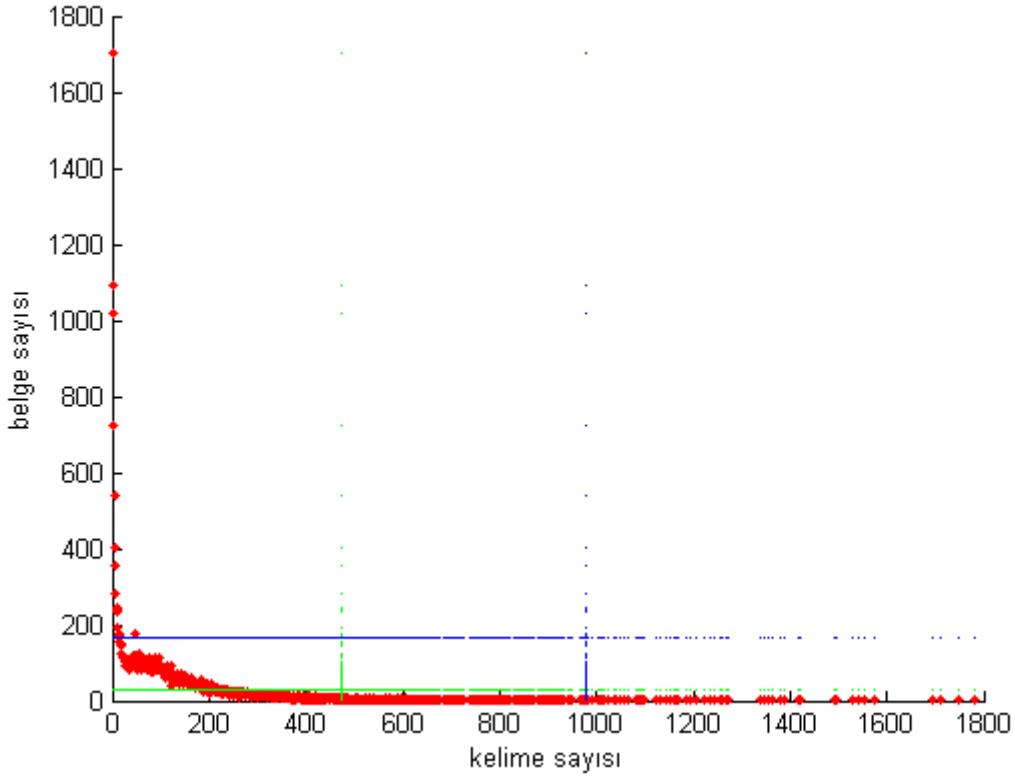
Şekil 5.3 DMOZ dizinin taranması sonucu elde edilen klasör yapısı

5.2.1 DMOZ Veri Kümesi Önişleme Adımları

Düz HTML formatında kayıt edilen örün sayfalarının içerikleri *HTML Parser* [81] isimli ayrıştırıcı yardımı ile alınmıştır. Toplanan sayfalar incelendiğinde bazı sayfalarda çok az, bazı sayfalarda ise çok fazla kelime bulunduğu görülmüştür. Şekil 5.4'te sayfalardaki kelimelerin dağılımları verilmiştir. Sayfaların sahip olduğu kelime sayıları ile belli bir miktar kelimeye sahip olan sayfa sayıları görünmektedir. Şekil 5.4 incelendiğinde bazı sayfalarda çok fazla kelime olduğu, birçok sayfada ise çok az kelime bulunduğu görülebilir. Veri kümesinin genelinde ise sayfalarda 200'den az sayıda kelime bulunmaktadır.

Sayfalardaki içeriği oluşturan kelimelerin sayılarındaki bu dengesiz dağılım sınıflandırma performansını düşüreceği için Bölüm 5.1.4'te anlatılan kutu çizimi filtresi uygulanarak normal sınırların dışında sayıda kelime içeren sayfalar elenmiştir. Kelime-

sayfa dağılımlarının ortalamaları ve standart sapmaları hesaplanarak dağılımda ortalama değer merkezli, standart sapma sınırlı bir kutu çizilmiştir. Bu sınırlar da Şekil 5.4'te verilmiştir. Kutunun içinde kalan sayfalar veri kümesi olarak kullanılmış, dışında kalanlar ise normalin dışında veri içeren örnekler oldukları için elenmiştir.

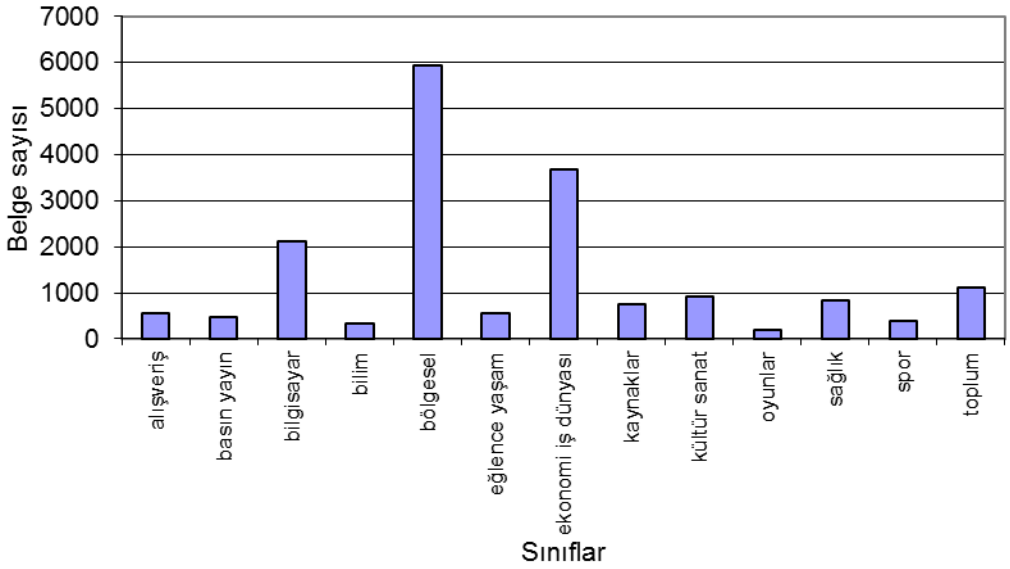


Şekil 5.4 DMOZ veri kümesindeki kelime ve sayfa sayılarının dağılımı

Veri kümesinde kalan sayfalardaki kelimeler *Zemberek* [77] aracı ile köklerine ayrılmıştır. Türkçe sondan eklemeli bir dil olduğu için, bir köke eklenen yapım ekleri ile anlam değişebilir. Mümkün olan en fazla çeşitliliği sağlamak için Zemberek aracının bulduğu kökler arasından en fazla yapım ekine sahip olan halleri kullanılmıştır. Elde edilen köklerin türlerinin sınıflandırma performansına etkisini ölçmek için iki ayrı veri kümesi oluşturulmuştur. Birincisinde örün sayfalarında bulunan isim, sıfat, yüklem vb. her çeşit kök yer almaktadır. İkincisinde ise sadece isim türündeki köklere yer verilmiş, diğer tüm türlere ait kökler elenmiştir.

Türkçe'de sık kullanılan kelimeler de sayfalardan temizlenerek ön işleme adımları tamamlanmıştır. Ön işleme sonunda toplanan örün sayfalarından filtreleme ve temizlik işlemleri sonucunda seçilen 17.885 adedi veri kümesi olarak hazırlanmıştır. Bu

sayfalarda tüm kelime türlerini içeren veri kümesinde toplam 18.363 adet tekil kelime bulunmaktadır. Sadece isim türündeki kelime köklerini içeren veri kümesinde ise 11.922 tekil kelime yer almaktadır. Hazırlanan veri kümesinin World/Türkçe başlığı altında yer alan 13 sınıftaki dağılımları Şekil 5.5'te verilmiştir. Burada dikkati çeken, *Bölgesel* ve *Ekonomi_İş_Dünyası* başlıkları altında diğer sınıflardan çok daha fazla sayıda örnek olmasıdır. Bu dengesiz dağılım sınıflandırma performansının düşmesine yol açabilecek bir etkendir.

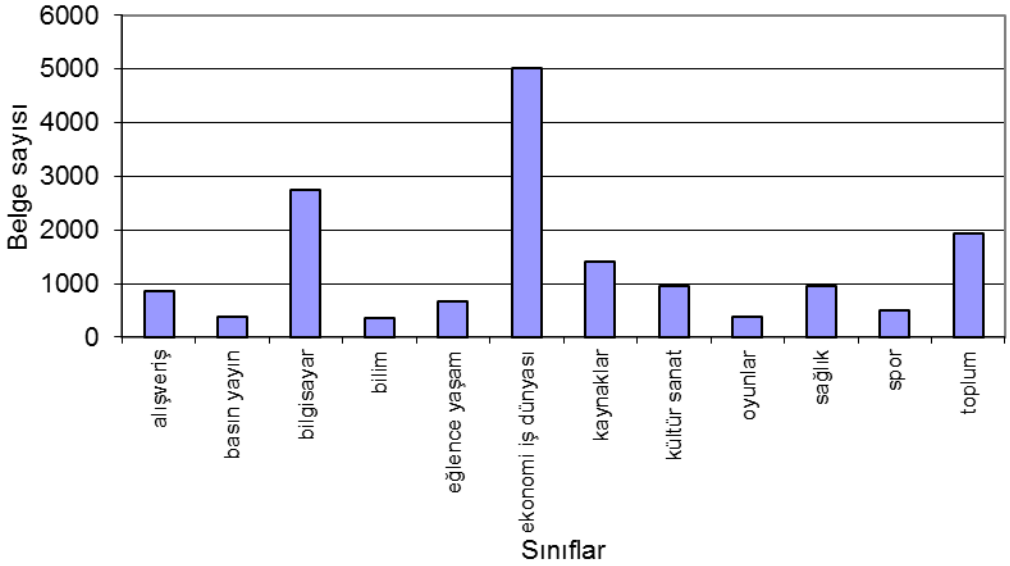


Şekil 5.5 DMOZ veri kümesindeki örneklerin sınıflardaki dağılımı

5.2.2 DMOZ Bağımsız Test Kümesi

Oluşturulan DMOZ veri kümesinin yanında, sistemi sadece çapraz geçerleme değil, aynı zamanda bağımsız bir test kümesiyle de test edebilmek adına, DMOZ veritabanının "World/Türkçe" başlığı değişik zamanlarda tekrar ziyaret edilerek, yeni eklenen sayfalar da indekslenmiştir. Bölüm 5.2.1'de anlatılan ön işleme adımları bağımsız DMOZ test kümesindeki sayfalar için de aynen uygulanmıştır. Bu işlemler sonucunda, çalışmaya esas olarak hazırlanan DMOZ veri kümesinin haricinde, o veri kümesinde hiç yer almamış toplam 16.660 adet örün sayfasından oluşan bağımsız DMOZ test kümesi oluşturulmuştur. "Bölgesel" sınıfında çok büyük sayıda sayfa bulunduğu için bu sınıf da işleme katılmamıştır. DMOZ bağımsız test veri kümesi, 12 sınıfta toplam 18.289

terimden oluşmaktadır. Hazırlanan bağımsız test veri kümesine ait örün sayfalarının 12 sınıftaki dağılımları Şekil 5.6'da verilmiştir.



Şekil 5.6 Bağımsız DMOZ test kümesindeki örneklerin sınıflardaki dağılımı

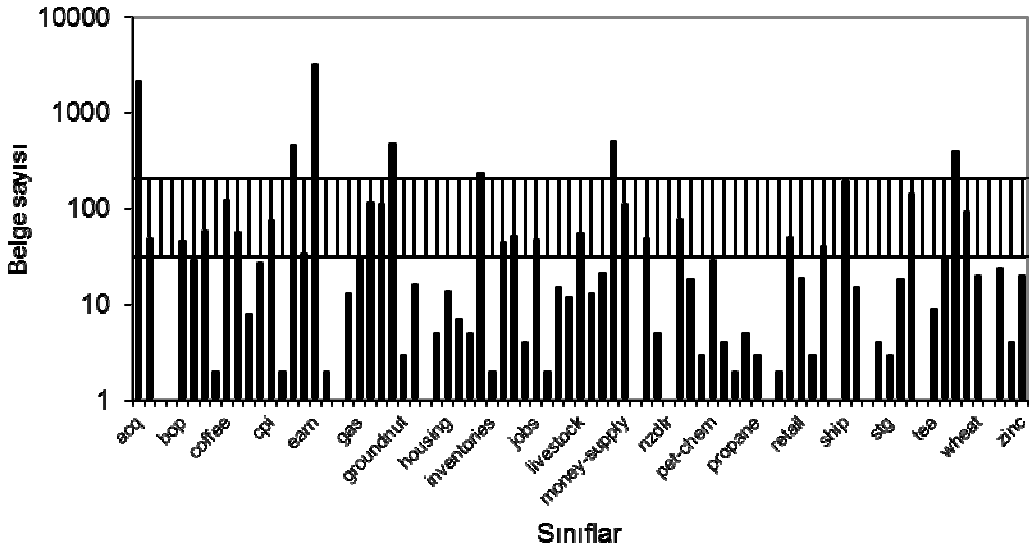
5.3 Reuters-21578 Veri Kümesi

Reuters-21578 bilgi erişimi ve makine öğrenmesi disiplinlerinde kullanılan metin sınıflandırma işlemleri için standart bir veri kümesidir [82]. 1987 yılında Reuters haber ajansının yayımladığı 21578 adet İngilizce haber metninden oluşmaktadır.

Reuters veri kümesinde haberleri sınıflandırmak için 135 adet konu başlığı bulunmaktadır. Bazı haberlerde birden fazla başlık olabildiği gibi, bazı haberlerde de hiçbir başlık bilgisi bulunmamaktadır. Bu özellikleriyle Reuters veri kümesi olduğu gibi kullanılmaz, bunun yerine benzer tipteki sınıflandırma çalışmalarının başarımlarını karşılaştırmak üzere belirli özelliklere göre bölümlenmiş versiyonları tercih edilir. Bunun dışında belirli sınıfları ya da belirli özellik sahibi örnekleri kullanarak Reuters temelli veri kümeleri de çalışmalarda sıklıkla kullanılmaktadır. Bu çalışmada kullanmak üzere Reuters veri kümesi uyarlanırken sadece tek bir başlığı olan ve haber metni bulunan belgeler kullanılmıştır.

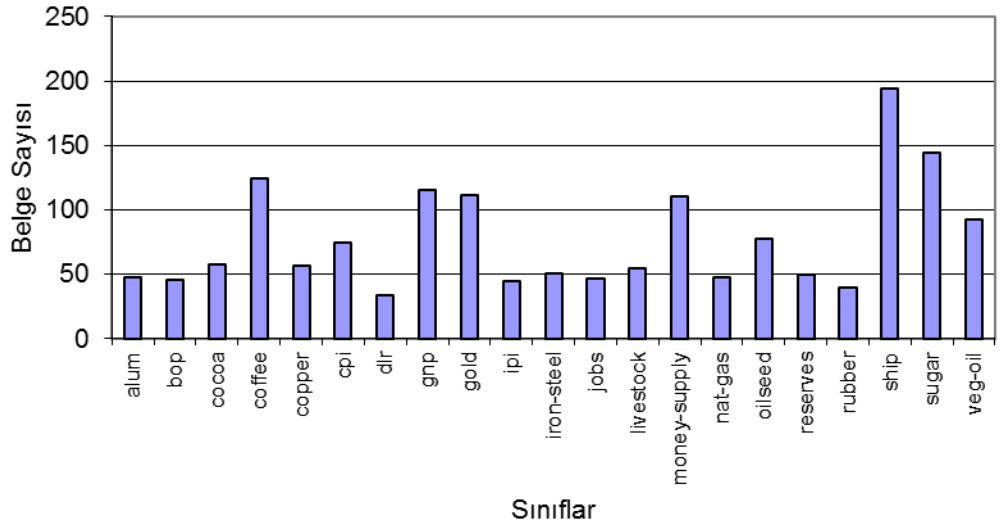
5.3.1 Reuters Veri Kümesi Önişleme Adımları

Porter'ın [76] kök bulma yöntemi ile veri kümesindeki belgelerde yer alan kelimelerin kökleri bulunmuştur. Ek-B'de listelenmiş olan İngilizce sık kullanılan kelimeler temizlenmiş, sayılar ve noktalama işaretleri kaldırılmıştır. Bu ön işleme adımlarından sonra veri kümesinde 81 sınıfa ait 9554 tekil belge kalmıştır. Belgelerin sınıflardaki dağılımları Şekil 5.7'de verilmiştir. Buradaki dağılım da düzgün değildir. Sınıflarda mümkün olduğunca eşit sayıda belge olmasını sağlamak amacıyla kutu çizimi filtresi sınıflardaki belge sayıları üzerinde uygulanmıştır. Kutu çizimi filtresinin alt ve üst çeyrek çizgileri ortalamanın 0,2 standart sapma üst ve altından geçecek şekilde çizilmiş, bu bölge içinde kalan sınıflar kullanılmış, bu alanın dışında kalan sınıflar ise aykırı olarak kabul edilmiş ve temizlenmiştir. Alt ve üst çeyrek çizgileri ile arada kalan alan da Şekil 5.7'de görülmektedir. Şekil 5.7'deki Y ekseni sınıflardaki örnek sayılarını göstermektedir ve logaritmik ölçekte çizilmiştir.



Şekil 5.7 Reuters-21578 veri kümesinde ön işleme adımları sonrası belgelerin sınıflardaki dağılımı ve alt-üst çeyrek çizgileri arasında kalan alan

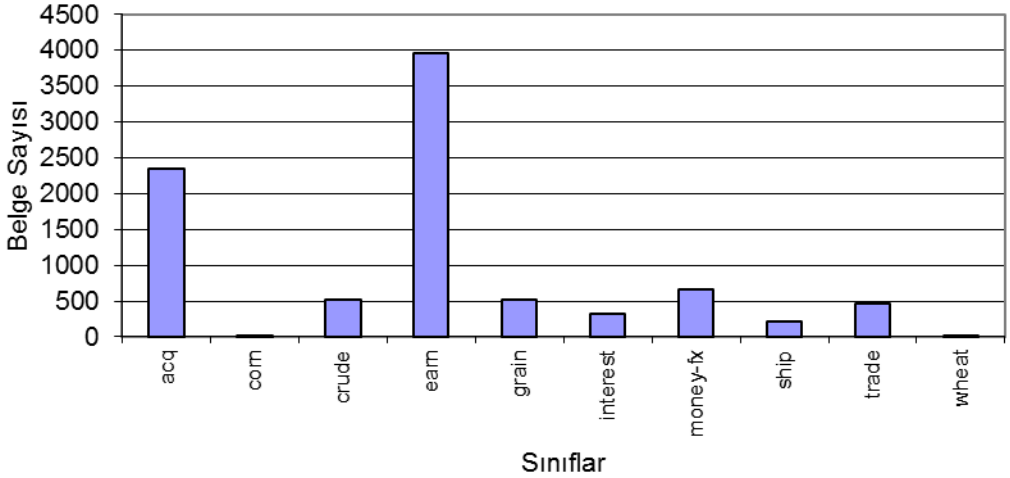
Ön işleme ve filtreleme adımları sonucunda 21 sınıfta 1623 adet belge veri kümesi olarak hazırlanmıştır. Filtrelenmiş veri kümesinde 8120 adet tekil terim yer almaktadır. Hazırlanan veri kümesinin sınıfları ve belgelerin sınıflardaki dağılımları Şekil 5.8'de verilmiştir.



Şekil 5.8 Reuters veri kümesindeki örneklerin sınıflardaki dağılımı

5.3.2 Reuters ModApte10 Versiyonu Veri Kümesi

Reuters veri kümesi üzerinde tıpkı Bölüm 5.2.2'de tanıtılan çapraz geçerleme yerine bağımsız test örnekleriyle başarımı ölçebilmek için, Reuters veri kümesinin ModApte-10 olarak bilinen versiyonu kullanılmıştır.



Şekil 5.9 ModApte-10 veri kümesindeki örneklerin sınıflardaki dağılımı

ModApte-10 veri kümesi eğitim ve test örnekleri olmak üzere hazır olarak ikiye bölünmüş durumdadır. Veri kümesinde 9603 eğitim, 3299 test örneği yer almaktadır.

ModApte-10 aşırı derecede heterojen bir yapıya sahiptir. Bir sınıfta çok az eğitim ve test örneği varken, başka bir sınıfta çok sayıda örnek bulunabilmektedir. Bu durum Şekil 5.9'da verilmiş olan örneklerin sınıflardaki dağılım grafiğinde de açıkça görülmektedir. Şekil 5.9'da eğitim ve test kümelerindeki örneklerin tamamı bir arada gösterilmiştir.

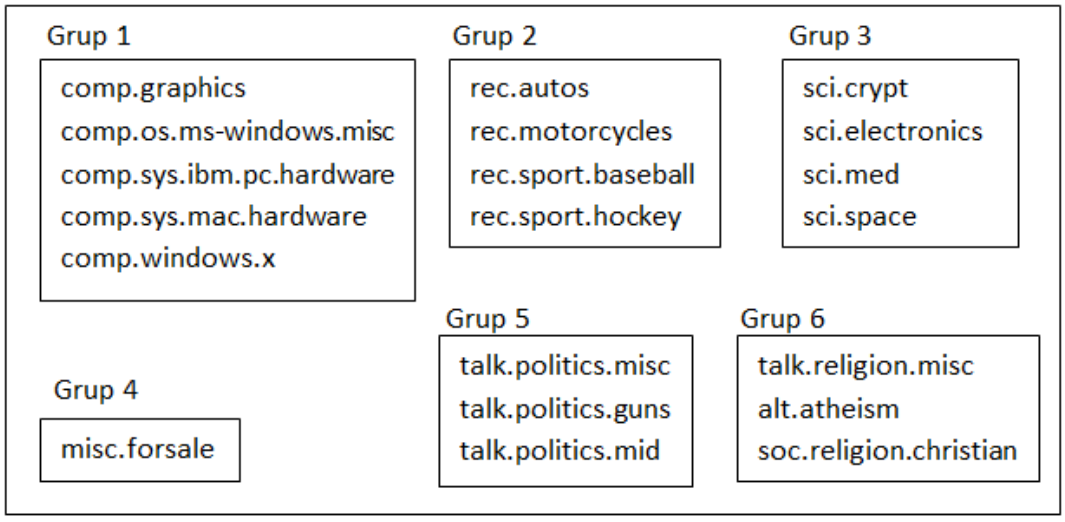
ModApte-10 veri kümesinin testlerde kullanımı, geliştirilen soyut özellik çıkarım yönteminin aşırı heterojen veri kümelerindeki davranışını test edebilmeyi sağlayacaktır.

5.4 20 Newsgroups Veri Kümesi

20 Newsgroups veri kümesi, metin sınıflandırma ve kümeleme işlemleri için sıklıkla kullanılan bir veri kümesidir [83]. 20 değişik başlık altında toplanmış yaklaşık 20.000 adet haber grubu postası koleksiyonundan oluşmaktadır.

Bazı başlıklardaki konular birbirine oldukça yakınken (ör: comp.sys.ibm.pc.hardware / comp.sys.mac.hardware), bazıları da hiç alakasız konulardaki başlıklardır (ör: misc.forsale / soc.religion.christian). Genel olarak veri kümesindeki başlıkların 6 grupta toplandığı görülmektedir. Veri kümesinde yer alan başlıklar ve bunların birbirleriyle olan alakaları ile oluşturdukları kümeler Şekil 5.10'da verilmiştir.

Veri kümesinin orijinalinde yaklaşık 20.000 doküman bulunmaktadır. Çalışmada tüm veri kümesi yerine, her sınıfta yüzer adet örnek barındıran "20 Newsgroups Mini" [84] versiyonu kullanılmıştır.



Şekil 5.10 20-Newsgroups veri kümesindeki sınıflar ve yakınlıklarına göre kümeleri

5.4.1 20 Newsgroups Veri Kümesi Önişleme Adımları

20 Newsgroups veri kümesi, XML benzeri yapıya sahip girdilerden oluşmaktadır. Bu girdileri çözümleyerek haber grubu postasının kod numarası, ait olduğu sınıf ve içeriğini ayıran bir uygulama geliştirilerek veri kümesi kullanıma hazır hale getirilmiştir. Porter'ın [76] kök bulma yöntemi ile veri kümesindeki belgelerde yer alan kelimelerin kökleri bulunmuştur. Ek-B'de verilmiş olan İngilizce sık kullanılan kelimeler temizlenmiş, sayılar ve noktalama işaretleri kaldırılmıştır. Bu işlemler sonucunda 20 sınıfta yüzer adet belgeden oluşan, toplam 2000 belge ve 25.204 terimden oluşan deneysel veri kümesi oluşturulmuştur.

BÖLÜM 6

TEST SONUÇLARI

Önerilen özellik çıkarım yöntemi, hazırlanan DMOZ, Reuters ve 20-Newsgroups veri kümeleri üzerinde yedi farklı sınıflandırma algoritması ile denenmiştir. Ayrıca geliştirilen soyut özellik çıkarım yöntemi Bölüm 3.3.1 ve 3.3.2'de anlatılan özellik seçim ve özellik çıkarım yöntemleri ile bahsi geçen veri kümeleri üzerinde sınıflandırma işleminden önce uygulanarak başarımları karşılaştırılmıştır. Bu bölümde elde edilen sonuçlar sunularak yorumlanmaktadır.

6.1 Deney Kurulumu

Soyut özellik çıkarım yönteminin etkisini test etmek ve diğer özellik seçim ve çıkarım yöntemleri ile karşılaştırabilmek için değişik türlerde sınıflandırma algoritmaları seçilmiştir. İstatistik temelli, karar ağacı, kural tabanlı, örnek temelli ve çekirdek tabanlı sınıflandırıcı türlerinin en çok bilinen ve kullanılan algoritmalarına yer verilmiştir. Seçilen algoritmalar ve testlerde kullanılan parametreleri, seçilen doğrulama yöntemleri ve uygulama ortamı alt bölümlerde verilmiştir.

Seçilen yöntemlerin veri kümelerinde karşılaştırmalı test sonuçları Bölüm 6.2, 6.3, 6.4 ve 6.5'te incelenmiştir. Karşılaştırmak üzere seçilen özellik seçim ve çıkarım yöntemleriyle indirgenen özelliklerin sayıları farklı olmaktadır. Karşılaştırılan yöntemlerin ürettiği indirgenmiş özellik sayıları Çizelge 6.1'de verilmiştir. LDA yöntemi sınıflar arası ayırtaçları özellik olarak çıkardığı için, ürettiği özellik sayıları veri kümesindeki sınıf sayısından bir eksiktir. Yöntemlerin başarısının indirgenen özellik sayılarına bağlı olup olmadığını test etmek için, tüm yöntemlerle veri kümeleri sınıf

sayısı kadar boyuta indirgenerek karşılaştırma testleri yinelenmiştir. Bu testlerle ilgili sonuçlar Bölüm 6.6'da verilmiştir.

Testlerin istatistik analizini yapmak için, Reuters veri kümesi kullanılarak hipotez testi gerçekleştirilmiştir. Eşleştirilmiş t-testi kullanılarak yapılan bu analiz Bölüm 6.7'de açıklanmıştır.

Soyut özellik çıkarım yöntemi kullanıldığında, eğitim örneği sayısının sınıflandırma başarımına olan etkisini görmek için 20-Newsgroups veri kümesi kullanılarak yapılan testler Bölüm 6.8'de incelenmiştir.

Çizelge 6.1 Seçilen yöntemlerle veri kümelerinin indirgenmiş özellik sayıları

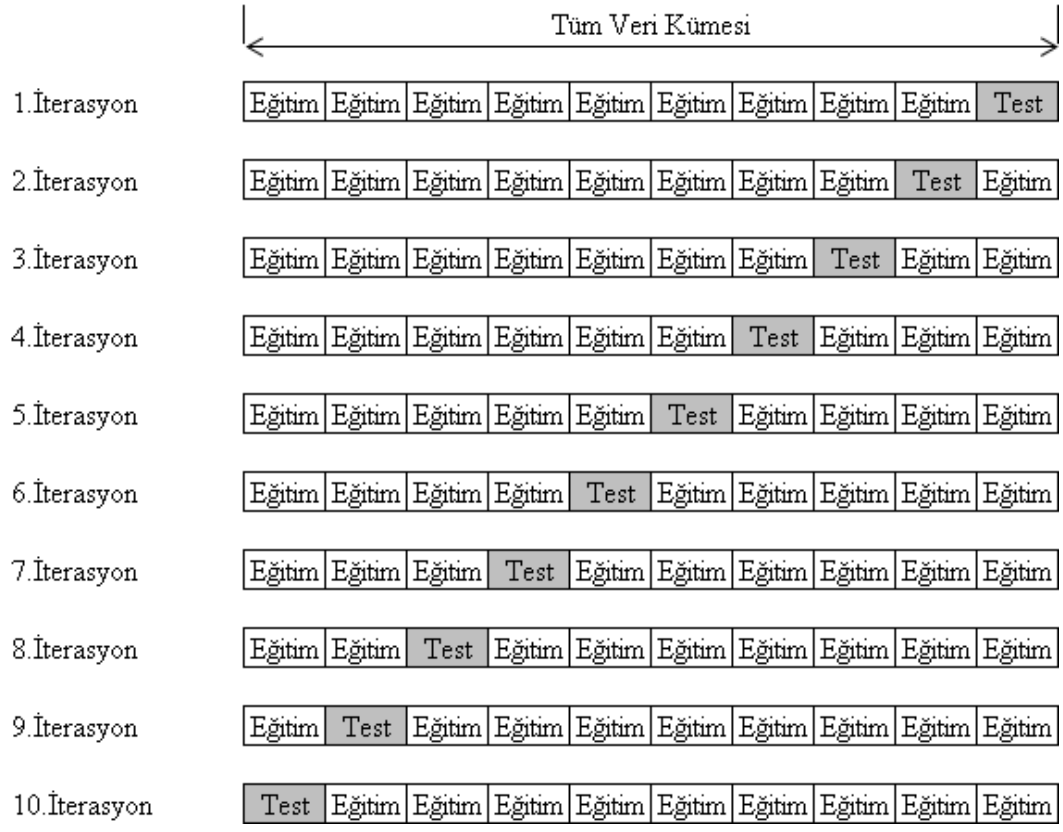
	DMOZ	**Reuters**	**20 Newsgroups**	**ModApte10**
İndirgeme Olmadan	17465	8120	25204	16435
Soyut Özellik Çıkarım Yöntemi	12	21	20	10
Karşılıklı Bilgi Yöntemi	3680	326	326	2019
Chi-Kare Yöntemi	3680	326	326	2019
Korelasyon Katsayısı Yöntemi	85	38	69	38
PCA	276	286	1422	1886
LSA	1926	1145	1056	1172
LDA	11	20	19	9

6.1.1 Uygulanan Doğrulama Yöntemleri

Yöntemleri karşılaştırmak üzere yapılan testlerde iki ayrı doğrulama yönteminden yararlanılmıştır. Boyut indirgeme metotlarının sınıflandırma algoritmalarının başarımlarına olan etkilerini doğrulamak için ilk olarak 10 kere çapraz doğrulama ve ikinci olarak bağımsız eğitim-test kümeleri yöntemleri seçilmiştir.

6.1.1.1 10 Kere Çapraz Doğrulama

Bağımsız DMOZ test veri kümesi ve ModApte-10 veri kümesi dışındaki tüm veri kümelerinde gerçekleştirilen testler için 10 kere çapraz doğrulama (*10-fold cross validation*) yöntemi seçilmiştir. Adı anılan veri kümelerinde uygulanan doğrulama yöntemi Bölüm 6.1.1.2'de anlatılmıştır. Bu yöntemde veri kümesi 10 eşit parçaya bölünür. Her bir iterasyonda veri kümesinin 9/10'luk kısmı eğitim için, ayrılan 1/10'luk kısmı da test için kullanılır. Her iterasyonda farklı bir parça test için ayrılarak, tüm veri kümesinin eğitim ve test için kullanımı sağlanır. 10 kere çapraz doğrulamanın genel yapısı Şekil 6.1'de verilmiştir.



Şekil 6.1 10 kere çapraz doğrulamada veri kümesinin ayırımı

Toplam hata hesaplanırken, 6.1 kullanılır. Burada test örneklerinin hatalarının ortalamaları alınarak 10 iterasyondaki toplam hata değerine ulaşılmaktadır.

$$E = \frac{1}{K} \sum_{i=1}^{K} E_i \tag{6.1}$$

10 kere çapraz doğrulamanın en önemli avantajı, veri kümesindeki tüm örneklerin eğitim ve test aşamalarında kullanılmasıdır. Bu sayede bazı örneklerin eğitim sürecinde yapmış olabilecekleri pozitif veya negatif etkiler bertaraf edilmiş olur. Her örneğin toplam 9 kere eğitim, 1 kere de test için kullanılması garanti edilmiş olur.

6.1.1.2 Bağımsız Eğitim-Test Kümeleri

Bağımsız DMOZ test veri kümesi ile yapılan testlerde ilk olarak daha önceden hazırlanan DMOZ veri kümesindeki 11.948 adet örnekle (“Bölgesel” sınıfı hariç) eğitilip, DMOZ test kümesindeki 16.660 örnek ile test edilmiştir. Ayrıca eğitim ve test kümeleri yer değiştirilerek sistem test kümesindeki 16.660 örnekle eğitilip eğitim kümesindeki 11.948 adet örnekle de test edilmiştir. Bu sayede eğitim ve test için ayrılan örnek öbekleri her iki işlem için de birer kere kullanılmıştır.

ModApte-10 veri kümesinde ise standart eğitim ve test ayırımı korunarak kullanılmıştır. Bu veri kümesinde 6489 örnek eğitim için, 2545 örnekse test için kullanılmıştır.

6.1.2 Seçilen Sınıflandırma Algoritmaları

Soyut özellik çıkarım yönteminin sınıflandırma algoritmalarından önce kullanıldığındaki başarımını, diğer yöntemlerin performanslarıyla kıyaslamak üzere değişik türlerde sınıflandırma algoritmaları kullanılmıştır. Kullanılan sınıflandırma algoritmaları ve parametreleri aşağıda listelenmiştir. Seçilen parametreler, sınıflandırma algoritmalarının varsayılan ayarlarındadır. En yakın komşu sınıflandırıcısında ise yapılan deneylerin sonucunda komşu sayısı 10, uzaklık ağırlıklandırma tipi de *1/uzaklık* olarak seçilmiştir. Algoritmaların varsayılan parametreleri dışındaki ayarlar ile çalıştırılmasının sonuçlara olan etkisini ölçmek üzere, destek vektör makineleri yöntemi dört farklı çekirdek türü ile de çalıştırılarak elde edilen sonuçlar karşılaştırılmıştır. Bu testler ile ilgili bilgi ve elde edilen sonuçlar Bölüm 6.8’de verilmiştir.

- İstatistikî sınıflandırıcı olarak Naive Bayes tercih edilmiştir. Naive Bayes sınıflandırıcısı Bayes teoremini güçlü bağımsız varsayımlara uygulamaya dayalıdır [54].
- Quinlan'ın [85] C4.5 algoritmasının uygulaması olan J48 ağacı, karar ağacı sınıflandırıcısına örnek olarak seçilmiştir.
    - Güven faktörü (*confidence factor*) 0,25 olarak kullanılmıştır.
    - Her yapraktaki minimum örnek sayısı 2 olarak alınmıştır.
- Kural tabanlı sınıflandırıcı olarak RIPPER algoritması kullanılmıştır [86].
    - Her bir kuralda örneklerin minimum toplam ağırlığı 2,0 olarak ayarlanmıştır.
    - Budama için 3 parça (3 fold) veri kullanılmıştır.
    - Optimizasyon aşaması ikişer kez çalıştırılmıştır.
- Örnek temelli sınıflandırıcılardan 10 en yakın komşu algoritması seçilmiştir.
    - Örnekler arası uzaklık ağırlıklandırma için, *1/uzaklık* yöntemi seçilmiştir.
- Kontrollü varyasyonlara sahip karar ağaçları koleksiyonu [87] için de 10 ağaçlı bir rastgele orman tercih edilmiştir.
    - Ağacın derinliği için herhangi bir sınır verilmemiştir.
- Verinin seyrekliğine dayanıklı çekirdek tabanlı sınıflandırıcı olarak destek vektör makineleri (*Support Vector Machines*, SVM) [88] seçilmiştir.
    - Modelleme için doğrusal çekirdek ($U^T xV$) seçilmiştir.
    - Maliyet parametresi 1,0 olarak ayarlanmıştır.
    - Sonlandırma toleransı olan epsilon değeri 0,001 olarak ayarlanmıştır.
- Doğrusal sınıflandırıcı olarak da geniş ve seyrek veri kümelerinde başarılı olduğu bilinen LINEAR [89] algoritmasına yer verilmiştir.
    - Maliyet parametresi 1,0 olarak ayarlanmıştır.
    - Sonlandırma toleransı olan epsilon değeri 0,01 olarak ayarlanmıştır.

6.1.3 Kullanılan Uygulama Ortamı

Deney kurulumunda yer alan testleri gerçekleştirmek üzere WEKA [92] adlı uygulama ortamı kullanılmıştır. WEKA, pek çok makine öğrenmesi yönteminin gerçeklendiği ve çeşitli şekillerde test edilebildiği, JAVA platformunda geliştirilmiş açık kaynak kodlu bir veri madenciliği uygulama programıdır. Sınıflandırma, kümeleme, ilişkilendirme işlemleri dışında, verinin ön işlenmesi için çeşitli filtreler ve boyut indirgeme yöntemleri de yine WEKA'da mevcuttur.

WEKA veri kümeleri için virgülle ayrılmış metin dosyalarını ya da ".arff" uzantılı kendi dosya tipindeki veri kümesi tanım dosyalarını kullanabilir. Bunların dışında uygun veritabanı bağlantı arayüzleri tanımlanırsa, veritabanı tablolarından da veri alabilmektedir. ARFF (*attribute-relation file format*) veri dosyaları temel olarak tanımlayıcı bir başlık, veri kümesinde mevcut olan tüm özellikler ve türlerinin tanımlandığı alan ve örneklerin yer aldığı veri alanından oluşmaktadır. Örnekler, tanımlanan özelliklerin aldığı değerler ile ifade edilmektedir. Bölüm 5'te açıklanan veri kümelerinin hepsi de çalışma kapsamında ARFF formatında hazırlanarak kullanılmışlardır. ARFF formatının içeriğini göstermek üzere; Bölüm 4.3'te kullanılan örnek veri kümesi, hem standart, hem de soyut özellik çıkarım yöntemi uygulanmış halleriyle ARFF dosyaları olarak EK-E'de sunulmuştur.

6.2 DMOZ Veri Kümesi ile Test Sonuçları

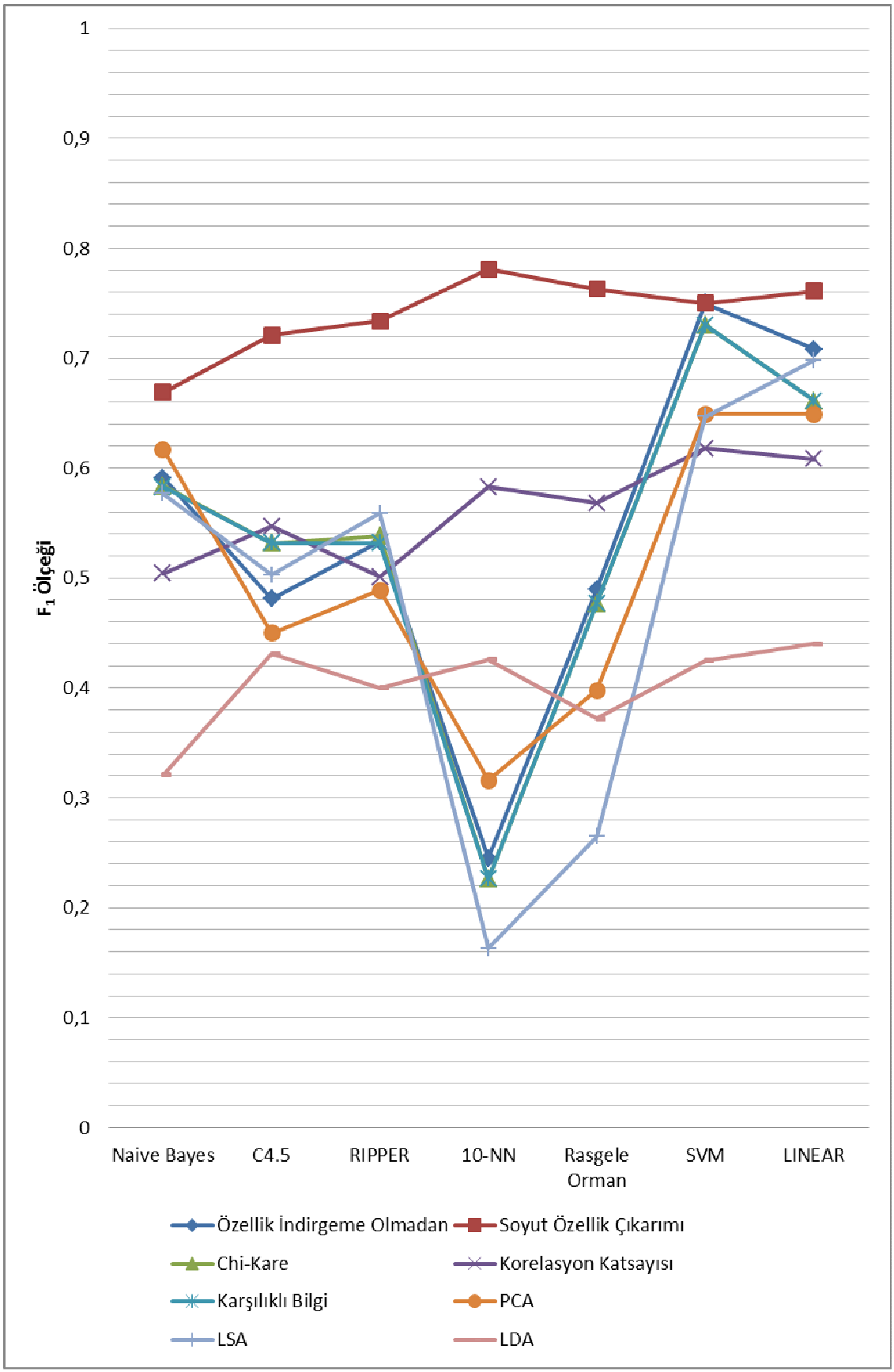
Soyut özellik çıkarım yönteminin sınıflandırma algoritmaları üzerindeki etkisini ölçmek amacıyla hazırlanan ve detayları Bölüm 5.2'de verilmiş olan DMOZ veri kümesi üzerinde testler yapılmıştır [93]. Hem tüm kelime türlerini içeren terimlere sahip olan, hem de sadece isim türündeki terimlerden oluşan veri kümeleri üzerinde testler yapılarak kelime türlerinin de sınıflandırma performansına olan etkisi ölçülmüştür.

Testler 10 kere çapraz doğrulama ile gerçekleştirilmiş, sınıflandırıcıların performansı 2.19'da verilen F_1 ölçeği ile ölçülmüştür. Çizelge 6.2'de seçilen yedi çeşit sınıflandırıcı üzerinde karşılaştırılan yöntemlerin uygulanmasıyla elde edilen sınıflandırma performansı sonuçları karşılaştırmalı olarak görülmektedir. Şekil 6.2'de sonuçlar karşılaştırmayı kolaylaştırmak üzere görselleştirilmiştir. Elde edilen sonuçlarla ilgili

detaylı performans sonuçları EK-C'de, en iyi ve en kötü sonucu veren deney kurulumları için karmaşıklık matrisleri EK-D'de sunulmuştur.

Çizelge 6.2 Soyut özellik çıkarımı yönteminin tüm kelime türlerini içeren DMOZ veri kümesi kullanıldığında sınıflandırma performansına olan etkisinin karşılaştırılması

	Özellik İndirgeme Olmadan	**Soyut Özellik Çıkarımı**	**Chi-Kare**	**Korelasyon Katsayısı**	**Karşılıklı Bilgi**	**PCA**	**LSA**	**LDA**
Naive Bayes	0,591	0,669	0,584	0,504	0,583	0,617	0,577	0,321
C4.5	0,481	0,721	0,532	0,547	0,532	0,450	0,503	0,431
RIPPER	0,533	0,734	0,538	0,501	0,532	0,489	0,559	0,400
10-NN	0,245	**0,781**	0,226	0,583	0,226	0,316	0,163	0,426
Rasgele Orman	0,490	0,763	0,477	0,568	0,477	0,398	0,265	0,372
SVM	**0,750**	0,750	**0,730**	**0,618**	**0,730**	**0,649**	0,647	0,425
LINEAR	0,708	0,761	0,661	0,608	0,661	**0,649**	**0,698**	**0,440**
Ortalama	0,543	**0,740**	0,535	0,561	0,534	0,510	0,487	0,402



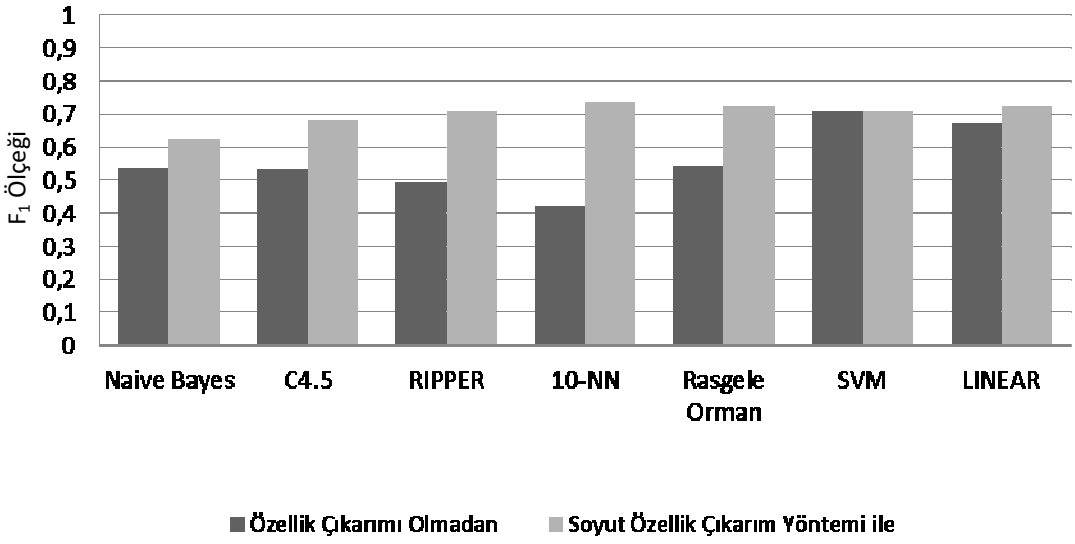
Şekil 6.2 Tüm kelime türlerinden oluşan DMOZ veri kümesi üzerinde özellik indirgeme yöntemlerinin sınıflandırma performansına olan etkisinin görsel olarak karşılaştırılması

Tüm kelime türlerini içeren DMOZ veri kümesindeki sonuçlar incelendiğinde, en yüksek F_1 değerinin 0,781 ile 10 en yakın komşu sınıflandırıcısından önce kullanılan soyut özellik çıkarımı ile elde edildiği görülmektedir. Bunun yanında tüm sınıflandırıcılar; soyut özellik çıkarımı ile birlikte kullanıldıklarında, diğer yöntemlere kıyasla en yüksek performansı göstermektedir. Ortalama F_1 değerlerine göre, soyut özellik çıkarım yöntemi en yakın yöntemden 0.179 daha yüksek F_1 değeri üretmiştir.

Sadece isim türünü içeren terimlerden oluşan DMOZ veri kümesi kullanıldığında, sınıflandırma algoritmaları üzerinde soyut özellik çıkarım yöntemi uygulanmadan ve uygulanarak elde edilen F_1 değerleri Çizelge 6.3'te verilmiştir. Şekil 6.3'te karşılaştırmalı F_1 değerleri okunabilirliği arttırmak için grafik olarak gösterilmiştir.

Çizelge 6.3 Sadece isim türünde kelimeleri içeren DMOZ veri kümesi kullanıldığında soyut özellik çıkarımı yönteminin sınıflandırma performansına olan etkisinin karşılaştırılması

	Soyut Özellik Çıkarımı Olmadan	**Soyut Özellik Çıkarımı Uygulanarak**
Naive Bayes	0,538	0,622
C4.5	0,534	0,680
RIPPER	0,497	0,710
10-NN	0,420	**0,736**
Rastgele Orman	0,543	0,726
SVM	**0,708**	0,709
LINEAR	0,673	0,723
Ortalama	0,559	**0,701**



Şekil 6.3 Soyut özellik çıkarım yönteminin sadece isim türünde kelimeleri içeren DMOZ veri kümesi üzerinde sınıflandırma performansına olan etkisi

DMOZ veri kümesinin sadece isim türünden kelimeler içeren versiyonunda soyut özellik çıkarım yöntemi kullanıldığında; tüm sınıflandırıcıların F_1 değerlerinin arttığı gözlemlenmiştir. Soyut özellik çıkarım yöntemi kullanılarak 10 en yakın komşu sınıflandırma algoritması kullanıldığında, özellik çıkarımı yapılmadan çalıştırılan destek vektör makineleri yönteminden 0,028 daha yüksek F_1 değeri elde edilmiştir. Yedi sınıflandırıcının ortalamasına bakıldığında, soyut özellik çıkarım yönteminin ortalama F_1 değerinde 0,142 artış getirdiği gözlemlenmiştir.

Tüm kelime türlerinden terimler içeren ve sadece isim türünde terimler içeren DMOZ veri kümesinde soyut özellik çıkarım yöntemi kullanılarak ve kullanılmadan elde edilen yedi sınıflandırıcıya ait ortalama ve en iyi F_1 değerleri Çizelge 6.4'te verilmiştir. Boyut indirgeme olmadan; sadece isim türünden kelimeler içeren veri kümesinde, tüm kelime türlerinden terimler içeren veri kümesinden ortalamada 0,016 daha iyi bir F_1 değeri gözlemlenmiştir. Ancak en iyi değerlere bakıldığında, sadece isim türünden kelimelerden oluşan veri kümesi 0,042 daha kötü performans göstermiştir. Buna karşın soyut özellik çıkarım yöntemi uygulanarak sadece isim türünden kelimelerden oluşan veri kümesi kullanılırsa; tüm kelimelerden oluşan veri kümesindeki sonuçlara göre F_1 değerinin ortalamada 0,039, en iyi değerlere göre de 0,045 düştüğü gözlemlenmiştir. Bu test sonucunda, sadece isim türünden kelimeleri kullanmak algoritmaların genelinde bir performans artışı sağlamadığı ve hatta başarımı düşürdüğü için, sadece

isim türündeki kelimeleri ayıklamakla uğraşmanın pratikte başarıma katkı sağlamadığı sonucuna ulaşılmıştır.

Çizelge 6.4 DMOZ veri kümesinde tüm kelime türlerinden terim ve sadece isim türü terimlerin kullanılmasının sınıflandırma performansına olan etkisinin karşılaştırılması

	F_1 Değeri	Soyut Özellik Çıkarımı Uygulanmadan	Soyut Özellik Çıkarımı Uygulanarak
Tüm Kelime Türlerinden Terimler	Ortalama	0,543	0,740
	En İyi	0,750 (SVM)	0,781 (10 en yakın komşu)
Sadece İsim Türünden Terimler	Ortalama	0,559	0,701
	En İyi	0,708 (SVM)	0,736 (10 en yakın komşu)

6.2.1 Bağımsız DMOZ Test Kümesi ile Sınıflandırma Sonuçları

10 kere çapraz doğrulama yöntemi ile yapılan deneylerin yanında, Bölüm 5.2.2'de detayları verilmiş olan, bağımsız DMOZ test kümesi ile de soyut özellik çıkarım yönteminin sınıflandırma algoritmaları üzerindeki etkisi ölçülmüş, bu sayede hiç tanınmayan büyük miktarda test verisi ile de yöntemin efektifliği test edilmiştir. Bunun için söz edilen bağımsız test kümesi, ilk olarak soyut özellik çıkarım işlemine tabi tutularak boyutları indirgenmiştir. Elde edilen sonuçlarla ilgili detaylı performans sonuçları EK-C'de, en iyi ve en kötü sonucu veren deney kurulumları için karmaşıklık matrisleri EK-D'de sunulmuştur.

Bu aşamada üç çeşit test yapılmıştır. İlk testte sadece bağımsız test kümesi kullanılarak, 10 kere çapraz doğrulama yapılmıştır. İkinci testte, eğitim kümesindeki 11.948 örnek (test kümesinde "Bölgesel" sınıfı olmadığı için, bu sınıftakiler hariç tüm örnekler) ile eğitilen sistem, 16.660 adet bağımsız (hiç görülmemiş) örnek ile test edilmiştir. Üçüncü testte ise eğitim ve test kümeleri değiştirilerek sistem 16.660 örnek ile eğitilip 11.948 örnek ile test edilmiştir. Bu testlerle ilgili F_1 ölçeği sonuçları Çizelge 6.5'te verilmiştir.

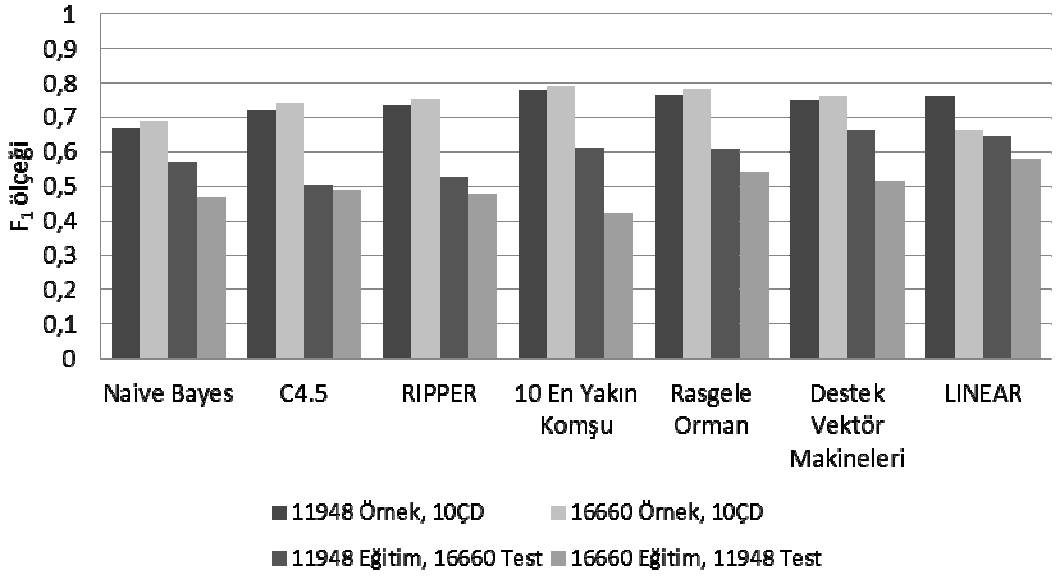
Çizelge 6.5 Soyut özellik çıkarımı yönteminin bağımsız DMOZ test kümesi kullanıldığında sınıflandırma performansına olan etkisinin karşılaştırılması

	Sadece test kümesi, 10 kere Çapraz Doğrulama ile	**Eğitim kümesi 11948 örnek, Test kümesi 16660 örnek**	**Eğitim kümesi 16660 örnek, Test kümesi 11948 örnek**
Naive Bayes	0,690	0,572	0,468
C4.5	0,741	0,503	0,491
RIPPER	0,753	0,528	0,478
10-NN	**0,789**	0,611	0,423
Rastgele Orman	0,783	0,607	0,540
SVM	0,760	**0,664**	0,514
LINEAR	0,663	0,647	**0,578**
Ortalama	**0.740**	0,590	0,499

Çizelge 6.5 incelendiğinde, soyut özellik çıkarım yöntemi ile boyutları indirgenmiş bağımsız test kümesi üzerinde yapılan 10 kere çapraz doğrulamada en yüksek F_1 değerinin 0,789 ile 10 en yakın komşu algoritmasında elde edildiği, bunu ise sadece 0,006 farkla 10 ağaçlı rasgele orman algoritmasının takip ettiği görülmektedir. Bu sonuçların, Bölüm 6.2.1'de verilen DMOZ veri kümesi ile elde edilen test sonuçları ile tutarlı ve yakın sonuçlar olduğu görülmektedir. Buna karşın 10 kere çapraz doğrulama yapmaksızın, sınıflandırma algoritmaları 11.948 örnek ile eğitilip hiç görülmemiş 16.660 örnek ile test edildiğinde en yüksek F_1 değeri 0,664 ile SVM algoritmasında elde edilmiştir. Bu sonuç, bir önceki testteki en iyi değerden 0,125 daha düşük bir F_1 değerine işaret etmektedir. 10 kere çapraz doğrulama yapmaksızın, eğitim ve test kümeleri yer değiştirilerek test edildiğinde ise en yüksek F_1 değeri 0,578 ile LINEAR algoritmasında elde edilmiştir. Bu sonuç da ilk testten 0,211 daha düşük bir F_1 değeri demektir.

Ortalama F_1 değerleri incelendiğinde, soyut özellik çıkarım yöntemi ile boyutları indirgenmiş bağımsız test kümesi üzerinde yapılan 10 kere çapraz doğrulamada 0,740; çapraz doğrulama yapmaksızın 11.948 örnek ile eğitim yapılıp hiç görülmemiş 16.660 örnek ile yapılan testlerde 0,590; eğitim ve test kümeleri yer değiştirilerek test edildiğinde ise 0,499 değerleri görülmektedir. Çapraz doğrulama kullanılarak yapılan test ile bağımsız eğitim-test kümeleri ile yapılan testler arasında F_1 değerlerinin sırasıyla ortalama 0,150 ve 0,241 düştüğü gözlemlenmiştir. Bu farkın sebebi, 10 kere çapraz doğrulama yapıldığında her adımda sistemin 14.994 örnek ile eğitilip sadece 1666 örnek ile test edilmesi ve bu işlemin Bölüm 6.1.1.1'de detaylarıyla anlatıldığı şekilde değiştirilerek 10 kez tekrarlanıp, bu adımlarda elde edilen F_1 değerlerinin ortalamasının nihai F_1 değeri olarak alınmasıdır. Bu sebeple, bu kadar fazla sayıda bağımsız örnekle yapılan testler sonucunda elde edilen değerin, az sayıda örnekle 10 kere yapılan test ortalamasından daha düşük çıkması beklenen ve olağan bir durum olarak değerlendirilmiştir.

Şekil 6.4'te karşılaştırmalı F_1 değerleri, okunabilirliği arttırmak için grafik olarak gösterilmiştir.



Şekil 6.4 Soyut özellik çıkarım yönteminin bağımsız DMOZ test veri kümesi üzerinde sınıflandırma performansına olan etkisi

6.3 Reuters Veri Kümesi ile Test Sonuçları

Soyut özellik çıkarım yönteminin metin işleme algoritmaları üzerindeki etkisini ölçmek amacıyla hazırlanan ve detayları Bölüm 5.3'te verilmiş olan Reuters veri kümesi üzerinde testler yapılmıştır [94]. Bölüm 6.2'de detayları verilen sınıflandırma algoritmaları üzerinde, seçilen boyut indirgeme metotları ile soyut özellik çıkarım yönteminin sınıflandırma performansına olan etkileri karşılaştırılmıştır.

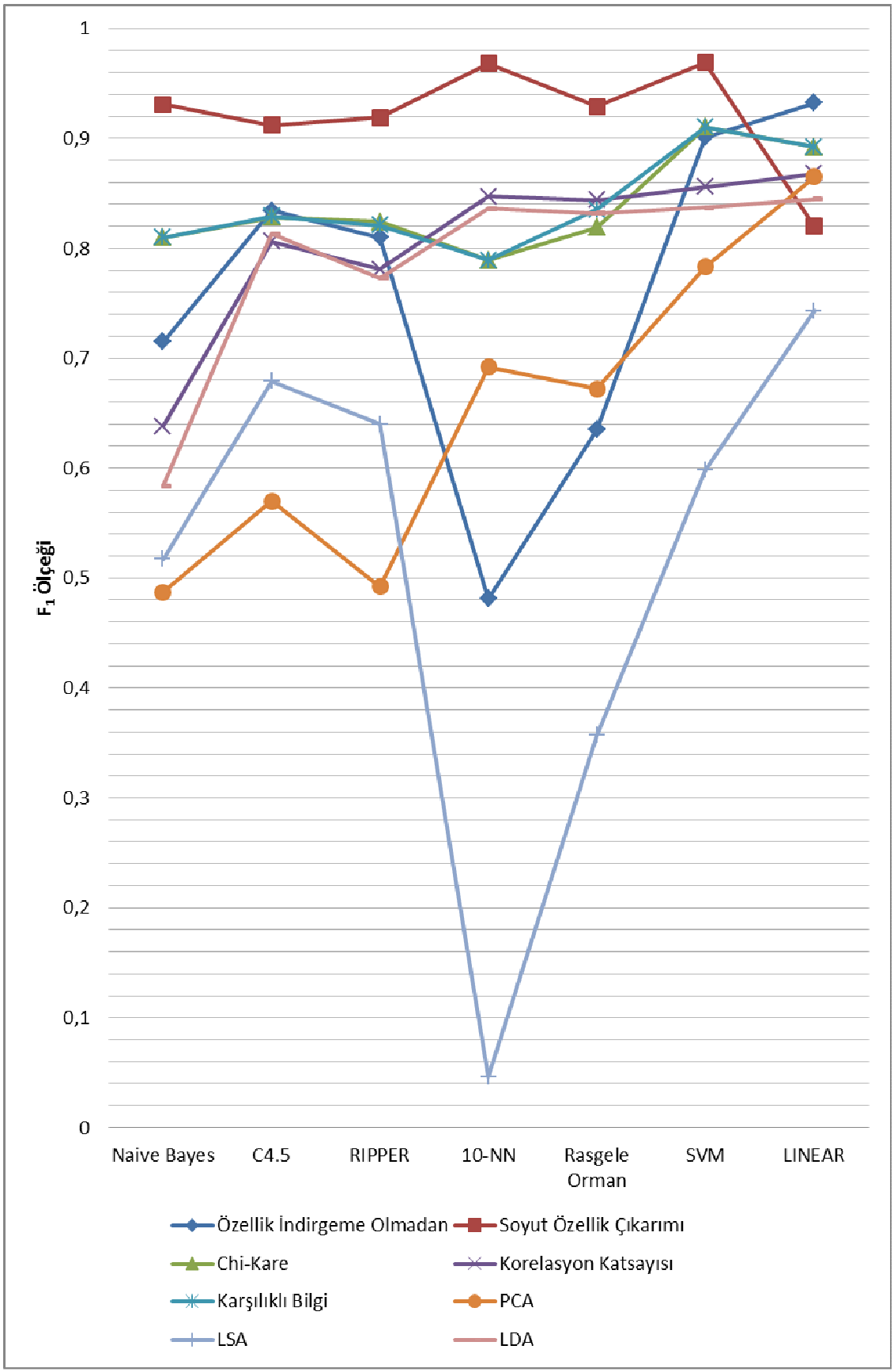
Testler 10 kere çapraz doğrulama ile gerçekleştirilmiştir. Sınıflandırıcıların performansı 2.19'da verilen F_1 ölçeği ile ölçülmüştür. Çizelge 6.6'da seçilen yedi çeşit sınıflandırıcı üzerinde karşılaştırılan boyut indirgeme yöntemlerin uygulanmasıyla elde edilen sınıflandırma performansı sonuçları F_1 değerleri ile karşılaştırmalı olarak görülmektedir. Şekil 6.5'te sonuçlar karşılaştırmayı kolaylaştırmak üzere görselleştirilmiştir.

Elde edilen sonuçlarla ilgili detaylı performans sonuçları EK-C'de, en iyi ve en kötü sonucu veren deney kurulumları için karmaşıklık matrisleri EK-D'de sunulmuştur.

Reuters veri kümesinde elde edilen sonuçlar incelendiğinde yedi sınıflandırıcının en yüksek ortalama F_1 değerini soyut özellik çıkarım algoritmasının ürettiği ve bu değerin en yakın yöntemin ortalamasından 0,080 daha yüksek olduğu görülmektedir. Testlerde elde edilen en yüksek F_1 değeri, 0,969 olarak soyut özellik çıkarımı uygulandıktan sonra çalıştırılan SVM algoritması ile elde edilmiştir. Sonuçlar sınıflandırıcı bazında incelendiğinde, soyut özellik çıkarım yönteminin LINEAR hariç tüm sınıflandırıcılarda diğer yöntemlerden daha iyi F_1 değerleri ürettiği, LINEAR algoritmasında ise sadece LSA'dan daha iyi olduğu görülebilir. Soyut özellik çıkarım yönteminin bu veri kümesinde diğerlerinden daha başarısız olmasının sebebi, LINEAR algoritmasının "bop" sınıfındaki tüm örnekleri yanlış sınıflandırmasıdır. Bu sınıftaki 14 örnek "gnp", 32 örnek "money-supply" olarak sınıflandırılmıştır. Soyut özellik çıkarım yöntemi ile "bop" sınıfı için çıkarılan özellikler incelendiğinde, "gnp" ve "money-supply" sınıflarına çok benzediği görülmüştür.

Çizelge 6.6 Soyut özellik çıkarımı yönteminin Reuters veri kümesinde sınıflandırma performansına olan etkisinin karşılaştırılması

	Özellik İndirgeme Olmadan	**Soyut Özellik Çıkarımı**	**Chi-Kare**	**Korelasyon Katsayısı**	**Karşılıklı Bilgi**	**PCA**	**LSA**	**LDA**
Naive Bayes	0,715	0,931	0,810	0,638	0,810	0,487	0,517	0,584
C4.5	0,834	0,912	0,828	0,806	0,829	0,570	0,679	0,813
RIPPER	0,810	0,919	0,824	0,781	0,821	0,492	0,640	0,773
10-NN	0,481	0,968	0,789	0,847	0,789	0,692	0,046	0,836
Rasgele Orman	0,635	0,929	0,819	0,844	0,835	0,672	0,357	0,832
SVM	0,901	**0,969**	**0,910**	0,856	**0,910**	0,783	0,598	0,837
LINEAR	**0,932**	0,820	0,892	**0,868**	0,892	**0,865**	**0,743**	**0,845**
Ortalama	0,758	**0,921**	0,839	0,806	0,841	0,652	0,511	0,789



Şekil 6.5 Reuters veri kümesi üzerinde özellik indirgeme yöntemlerinin sınıflandırma performansına olan etkisinin görsel olarak karşılaştırılması

6.4 20-Newsgroups Veri Kümesi ile Test Sonuçları

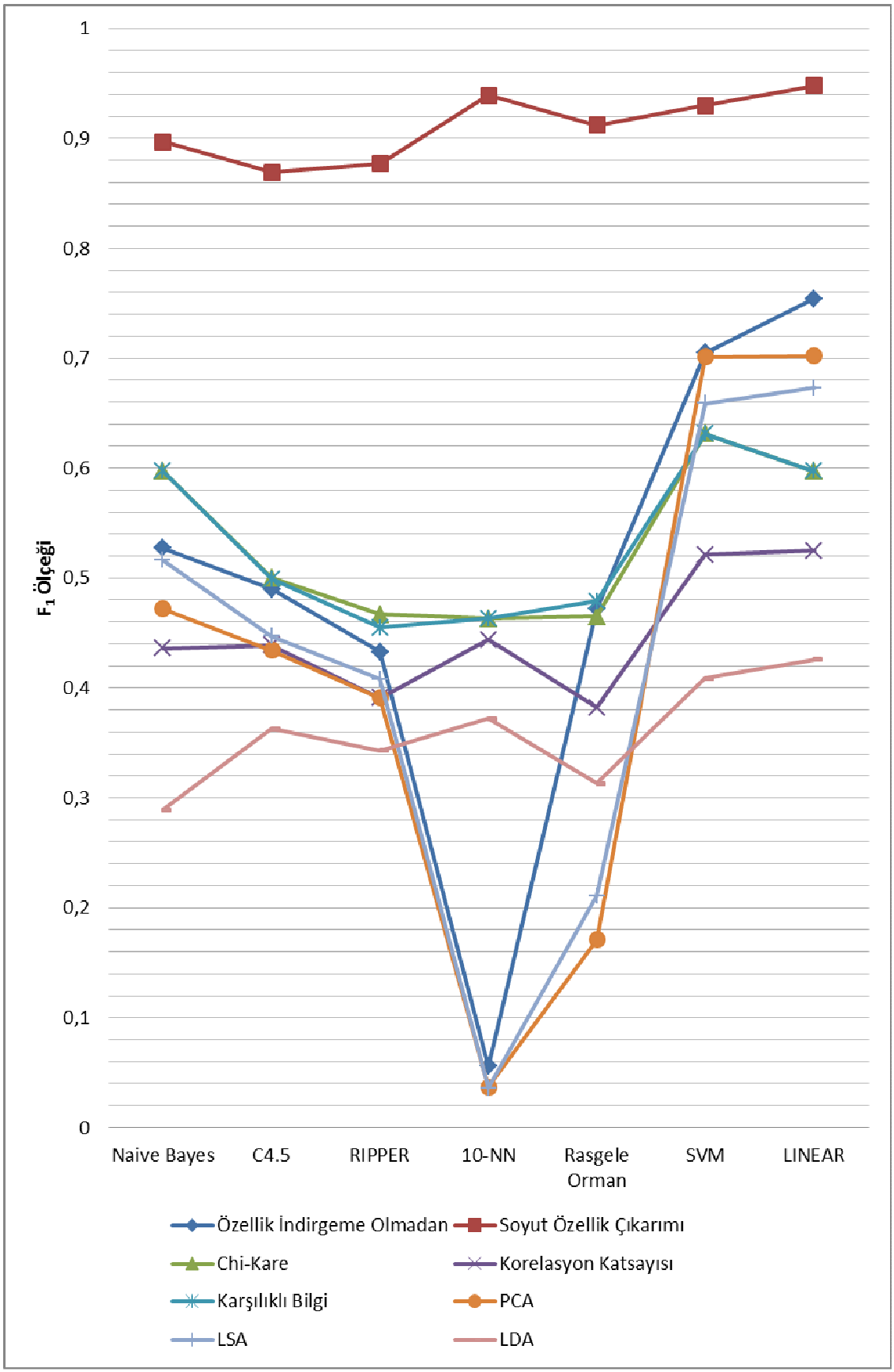
Soyut özellik çıkarım yönteminin metin işleme algoritmaları üzerindeki etkisini ölçmek amacıyla hazırlanan ve detayları Bölüm 5.4'te verilmiş olan 20-Newsgroups veri kümesi üzerinde testler yapılmıştır. Bölüm 6.2'de detayları verilen sınıflandırma algoritmaları üzerinde seçilen boyut indirgeme metotları ile soyut özellik çıkarım yönteminin sınıflandırma performansına olan etkileri karşılaştırılmıştır.

Testler 10 kere çapraz doğrulama ile gerçekleştirilmiş, sınıflandırıcıların performansı 2.19'da verilen F_1 ölçeği ile ölçülmüştür. Çizelge 6.7'de seçilen yedi çeşit sınıflandırıcı üzerinde karşılaştırılan yöntemlerin uygulanmasıyla elde edilen sınıflandırma performansı sonuçları karşılaştırmalı olarak görülmektedir. Şekil 6.6'da sonuçlar karşılaştırmayı kolaylaştırmak üzere görselleştirilmiştir. Elde edilen sonuçlarla ilgili detaylı performans sonuçları EK-C'de, en iyi ve en kötü sonucu veren deney kurulumları için karmaşıklık matrisleri EK-D'de sunulmuştur.

20-Newsgroups veri kümesinde elde edilen sonuçlar incelendiğinde yedi sınıflandırıcının en yüksek ortalama F_1 değerini soyut özellik çıkarım algoritmasının ürettiği ve bu değerin en yakın yöntemin ortalamasından 0,378 daha yüksek olduğu görülmektedir. Testlerde elde edilen en yüksek F_1 değeri, 0,948 olarak soyut özellik çıkarımı uygulandıktan sonra çalıştırılan LINEAR algoritması ile elde edilmiştir. Sonuçlar sınıflandırıcı bazında incelendiğinde, soyut özellik çıkarım yönteminin diğer tüm yöntemlerden çok daha iyi performans gösterdiği görülebilir. Bunun sebebi, çıkarılan soyut özelliklerin sınıflar için çok tanımlayıcı olması ve sınıflar arası farklılıkları belirgin hale getirmesidir. Performans farkı Şekil 6.6'da açıkça görülmektedir.

Çizelge 6.7 Soyut özellik çıkarımı yönteminin 20-Newsgroups veri kümesinde sınıflandırma performansına olan etkisinin karşılaştırılması

	Özellik İndirgeme Olmadan	**Soyut Özellik Çıkarımı**	**Chi-Kare**	**Korelasyon Katsayısı**	**Karşılıklı Bilgi**	**PCA**	**LSA**	**LDA**
Naive Bayes	0,527	0,897	0,597	0,436	0,597	0,472	0,516	0,289
C4.5	0,490	0,869	0,500	0,439	0,499	0,434	0,447	0,363
RIPPER	0,433	0,877	0,467	0,391	0,455	0,391	0,408	0,343
10-NN	0,056	0,939	0,463	0,444	0,463	0,037	0,036	0,372
Rasgele Orman	0,472	0,912	0,465	0,382	0,479	0,171	0,211	0,313
SVM	0,705	0,930	**0,631**	0,521	**0,631**	0,701	0,659	0,409
LINEAR	**0,754**	**0,948**	0,597	**0,525**	0,597	**0,702**	**0,673**	**0,426**
Ortalama	0,491	**0,910**	0,531	0,448	0,532	0,415	0,421	0,359



Şekil 6.6 20-Newsgroups veri kümesi üzerinde özellik indirgeme yöntemlerinin sınıflandırma performansına olan etkisinin görsel olarak karşılaştırılması

6.5 ModApte-10 Veri Kümesi ile Test Sonuçları

Soyut özellik çıkarım yönteminin etkinliğini hiç görülmemiş bir test kümesiyle sınamak amacıyla, detayları Bölüm 5.3.2'de verilmiş olan Reuters'in ModApte-10 versiyonu veri kümesi üzerinde testler yapılmıştır. Bölüm 6.2'de detayları verilen sınıflandırma algoritmaları üzerinde seçilen boyut indirgeme metotları ile soyut özellik çıkarım yönteminin sınıflandırma performansına olan etkileri karşılaştırılmıştır.

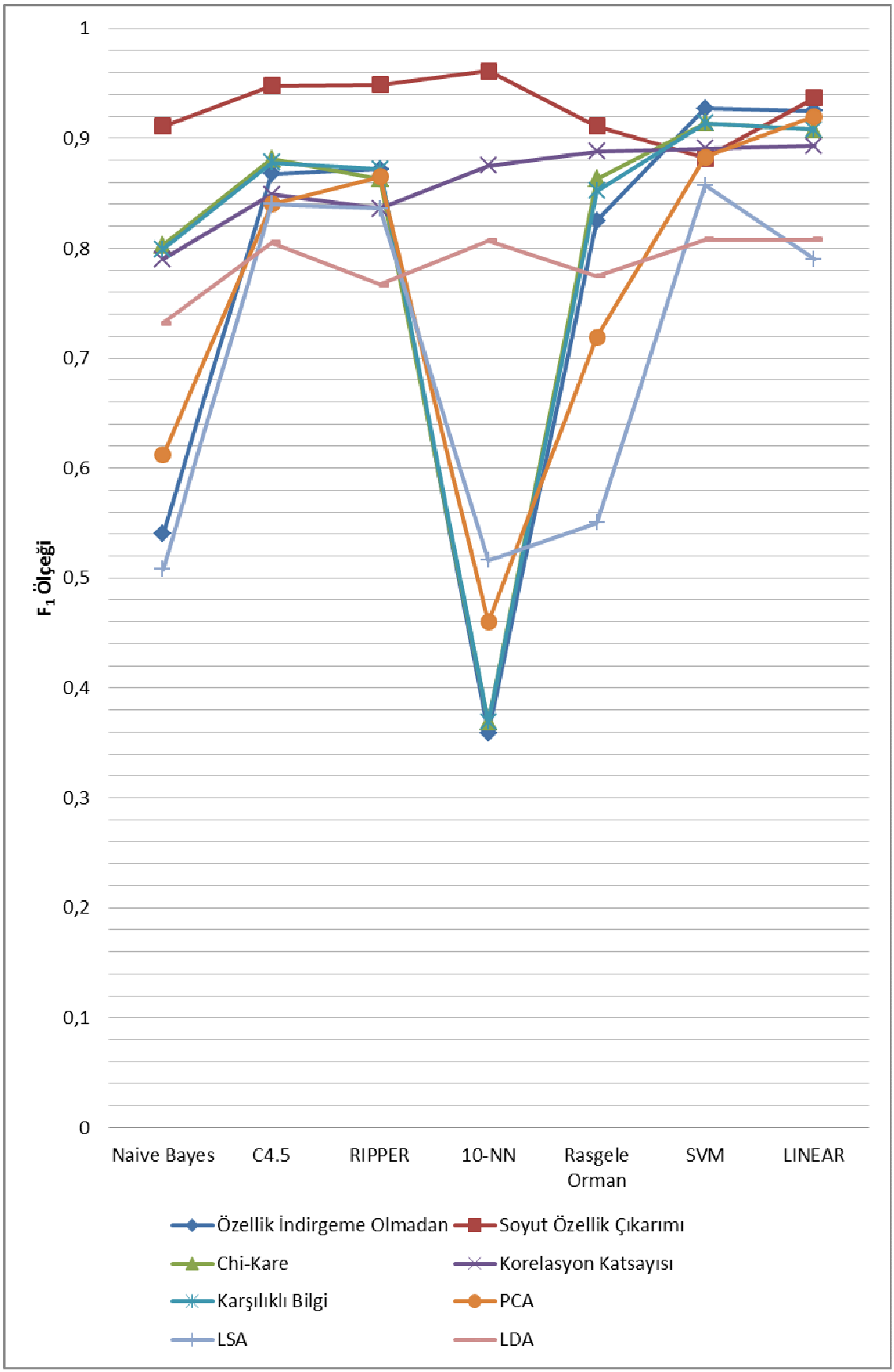
Sınıflandırıcılar ayrı eğitim kümeleri ile eğitilmiş, deneyler bağımsız test kümesi ile doğrulanmıştır. Sınıflandırıcıların performansı 2.19'da verilen F_1 ölçeği ile ölçülmüştür. Çizelge 6.8'de seçilen yedi çeşit sınıflandırıcı üzerinde karşılaştırılan yöntemlerin uygulanmasıyla elde edilen sınıflandırma performansı sonuçları karşılaştırmalı olarak görülmektedir. Şekil 6.7'de sonuçlar karşılaştırmayı kolaylaştırmak üzere görselleştirilmiştir. Elde edilen sonuçlarla ilgili detaylı performans sonuçları EK-C'de, en iyi ve en kötü sonucu veren deney kurulumları için karmaşıklık matrisleri EK-D'de sunulmuştur.

ModApte-10 veri kümesinde elde edilen sonuçlar incelendiğinde yedi sınıflandırıcının en yüksek ortalama F_1 değerini soyut özellik çıkarım algoritmasının ürettiği ve bu değerin en yakın yöntemin ortalamasından 0,068 daha yüksek olduğu görülmektedir. Testlerde elde edilen en yüksek F_1 değeri, 0,961 olarak soyut özellik çıkarımı uygulandıktan sonra çalıştırılan 10 en yakın komşu algoritması ile elde edilmiştir.

Sonuçlar sınıflandırıcı bazında incelendiğinde, soyut özellik çıkarım yönteminin SVM hariç tüm sınıflandırıcılarda diğer yöntemlerden yüksek performans sergilediği görülmüştür. SVM algoritmasında ise LSA ve LDA'dan daha iyi sonuçlar üretmiştir. Bunun bir sebebi, ModApte-10 veri kümesinin aşırı derecede heterojen olması sonucunda çok az örnek içeren iki sınıfa ait ("corn" ve "wheat") örneklerin SVM'in yüksek sayıda örnek ve özellikle iyi çalışma özelliğini bozmasıdır. İkinci sebep ise "interest", "trade" ve "ship" sınıflarını temsil eden soyut özelliklerin, bu sınıflardaki örneklerin birbirine benzemesi dolayısı ile birbirine yakın değerler almış olmasıdır. Bu durum, Reuters veri kümesinde başarımın nispeten düşük olduğu LINEAR algoritmasındaki durum ile benzeşmektedir.

Çizelge 6.8 Soyut özellik çıkarımı yönteminin ModApte-10 veri kümesinde sınıflandırma performansına olan etkisinin karşılaştırılması

	Özellik İndirgeme Olmadan	**Soyut Özellik Çıkarımı**	**Chi-Kare**	**Korelasyon Katsayısı**	**Karşılıklı Bilgi**	**PCA**	**LSA**	**LDA**
Naive Bayes	0,540	0,911	0,803	0,790	0,799	0,612	0,508	0,732
C4.5	0,868	0,948	0,881	0,849	0,878	0,840	0,840	0,805
RIPPER	0,872	0,949	0,863	0,836	0,872	0,865	0,836	0,767
10-NN	0,359	**0,961**	0,369	0,875	0,369	0,460	0,516	0,807
Rasgele Orman	0,825	0,911	0,863	0,888	0,852	0,719	0,550	0,775
SVM	**0,927**	0,882	**0,914**	0,891	**0,914**	0,883	**0,857**	**0,808**
LINEAR	0,925	0,937	0,908	**0,893**	0,908	**0,920**	0,790	**0,808**
Ortalama	0,759	**0,928**	0,800	0,860	0,799	0,757	0,700	0,786



Şekil 6.7 ModApte-10 veri kümesi üzerinde özellik indirgeme yöntemlerinin sınıflandırma performansına olan etkisinin görsel olarak karşılaştırılması

Veri kümeleri üzerinde yapılan testlerin geneli sınıflandırma algoritmaları temelinde incelendiğinde; 10 en yakın komşu algoritması, tüm veri kümelerinde, diğer sınıflandırıcılarla karşılaştırıldığında en kötü performansı sergilemiştir. 10 en yakın komşu sınıflandırıcısı için tüm boyut indirgeme yöntemlerinin ortalama F_1 değeri, DMOZ için 0,371; Reuters için 0,681; 20-Newsgroups için 0,351 ve ModApte-10 için 0,590 olmuştur. Buna karşın soyut özellikler, sınıfların birbirlerine benzerliklerini ve örneklerin sınıflara ait olma olasılıklarını gösterdiği için en yakın komşu algoritması için uygun özelliklerdir. Bu sebeple soyut özellik çıkarım yöntemi, 10 en yakın komşu sınıflandırıcısının başarımını tüm veri kümelerinde oldukça yükseltmiştir.

Veri kümelerinde gerçekleştirilen testlerin en yüksek performans gösteren sınıflandırma algoritması ise LINEAR olmuştur. Reuters, 20-Newsgroups ve ModApte-10 veri kümelerinde, tüm boyut indirgeme yöntemlerinin ortalama F_1 değerine göre LINEAR algoritması en yüksek başarımı sergilemiştir. LINEAR sınıflandırıcısı için tüm boyut indirgeme yöntemlerinin ortalama F_1 değeri, Reuters veri kümesinde 0,857; 20-Newsgroups veri kümesinde 0,653 ve ModApte-10 veri kümesinde 0,886 olmuştur. Sadece DMOZ veri kümesinde SVM'in ortalama başarımı LINEAR'dan yüksektir. 0,662 ortalama F_1 değeri üreten SVM algoritmasına karşın LINEAR 0,648 ortalama F_1 değeri üreterek, sadece 0,014 daha başarısız olmuştur.

Özet olarak; soyut özellik çıkarım yöntemi diğer yöntemlere göre daha yüksek başarım sergilemiş, sınıflandırma algoritmalarının performanslarını genel olarak oldukça arttırmıştır. Buna ilaveten, en düşük başarımı sergileyen 10 en yakın komşu algoritmasının performansını en yüksek seviyelere çıkarmıştır.

6.6 Eşit Sayıda Özelliğe Sahip Veri Kümeleri ile Test Sonuçları

Yöntemlerin başarısının indirgenen özellik sayılarına bağlı olup olmadığını test etmek için Reuters, 20-Newsgroups ve ModApte-10 veri kümeleri; chi-kare, korelasyon katsayısı, karşılıklı bilgi, temel bileşen analizi ve saklı anlamsal çözümleme yöntemleriyle, soyut özellik çıkarım yönteminin çıkardığı özellik sayısı olan veri kümesindeki sınıf sayısı kadar boyuta indirgenmiş, karşılaştırma testleri yinelenmiştir. Doğrusal ayırtaç çözümlemesi yöntemi sınıf sayısından bir eksik sayıda özellik çıkardığı için, bu karşılaştırma testlerine dahil edilmemiştir.

Reuters veri kümesi 21 sınıftan oluşmaktadır. Bu sebeple, özellik seçim yöntemleri ile veri kümesinden 21 özellik seçilmiş, özellik çıkarım yöntemleri ile de veri kümesi 21 özellikle ifade edilmiştir. Aynı şekilde 20-Newsgroups veri kümesi seçilen yöntemlerle 20 özelliğe indirgenmiştir. ModApte-10 veri kümesi ise karşılaştırma için seçilen yöntemler kullanılarak 10 özellik ile temsil edilmiştir.

21 özelliğe sahip Reuters veri kümesinde yapılan karşılaştırmalı test sonuçları, sınıflandırma algoritmalarının F_1 değerleri ile Çizelge 6.9'da verilmiş, Şekil 6.8 ile görselleştirilmiştir. 20 özellikle ifade edilen 20-Newsgroups veri kümesindeki karşılaştırmalı F_1 değeri sonuçları Çizelge 6.10'da verilmiş, Şekil 6.9'da görselleştirilmiştir. 10 özellikle ifade edilen ModApte-10 veri kümesindeki F_1 değeri sonuçları da Çizelge 6.11'de verilmiştir. ModApte-10 veri kümesinin sonuçları Şekil 6.10'da görselleştirilmiştir. Gerçekleştirilen testler sonucunda elde edilen detaylı performans sonuçları ise EK-C bölümünde verilmiştir. En iyi ve en kötü sonucu veren deney kurulumları için karmaşıklık matrisleri EK-D'de sunulmuştur.

21 özellikle ifade edilen Reuters veri kümesinde elde edilen sonuçlar incelendiğinde yedi sınıflandırıcının en yüksek ortalama F_1 değerini soyut özellik çıkarım algoritmasının ürettiği ve bu değerin en yakın yöntemin ortalamasından 0,123 daha yüksek olduğu görülmektedir. Testlerde elde edilen en yüksek F_1 değeri, 0,968 olarak soyut özellik çıkarımı uygulandıktan sonra çalıştırılan SVM ve 10 en yakın komşu algoritmaları ile elde edilmiştir. Sonuçlar sınıflandırıcı bazında incelendiğinde, soyut özellik çıkarım yönteminin LINEAR hariç tüm sınıflandırıcılarda diğer yöntemlerden daha iyi F_1 değerleri ürettiği, LINEAR algoritmasında ise sadece PCA'nın çok az daha iyi olduğu görülebilir. LINEAR algoritmasında soyut özellik çıkarım yöntemi ile PCA arasındaki performans farkı sadece 0,008'dir. Soyut özellik çıkarım yönteminin diğer yöntemlerden yüksek olan performansı Şekil 6.8'de izlenebilir.

20 özellikle ifade edilen 20-Newsgroups veri kümesinde elde edilen sonuçlar incelendiğinde yedi sınıflandırıcının en yüksek ortalama F_1 değerini soyut özellik çıkarım algoritmasının ürettiği ve bu değerin en yakın yöntemin ortalamasından 0,383 daha yüksek olduğu görülmektedir. Testlerde elde edilen en yüksek F_1 değeri, 0,947 olarak soyut özellik çıkarımı uygulandıktan sonra çalıştırılan LINEAR algoritması ile elde

edilmiştir. Sonuçlar sınıflandırıcı bazında incelendiğinde, soyut özellik çıkarım yönteminin diğer tüm yöntemlerden çok daha iyi performans gösterdiği görülebilir. Performans farkı Şekil 6.9'da açıkça görülmektedir.

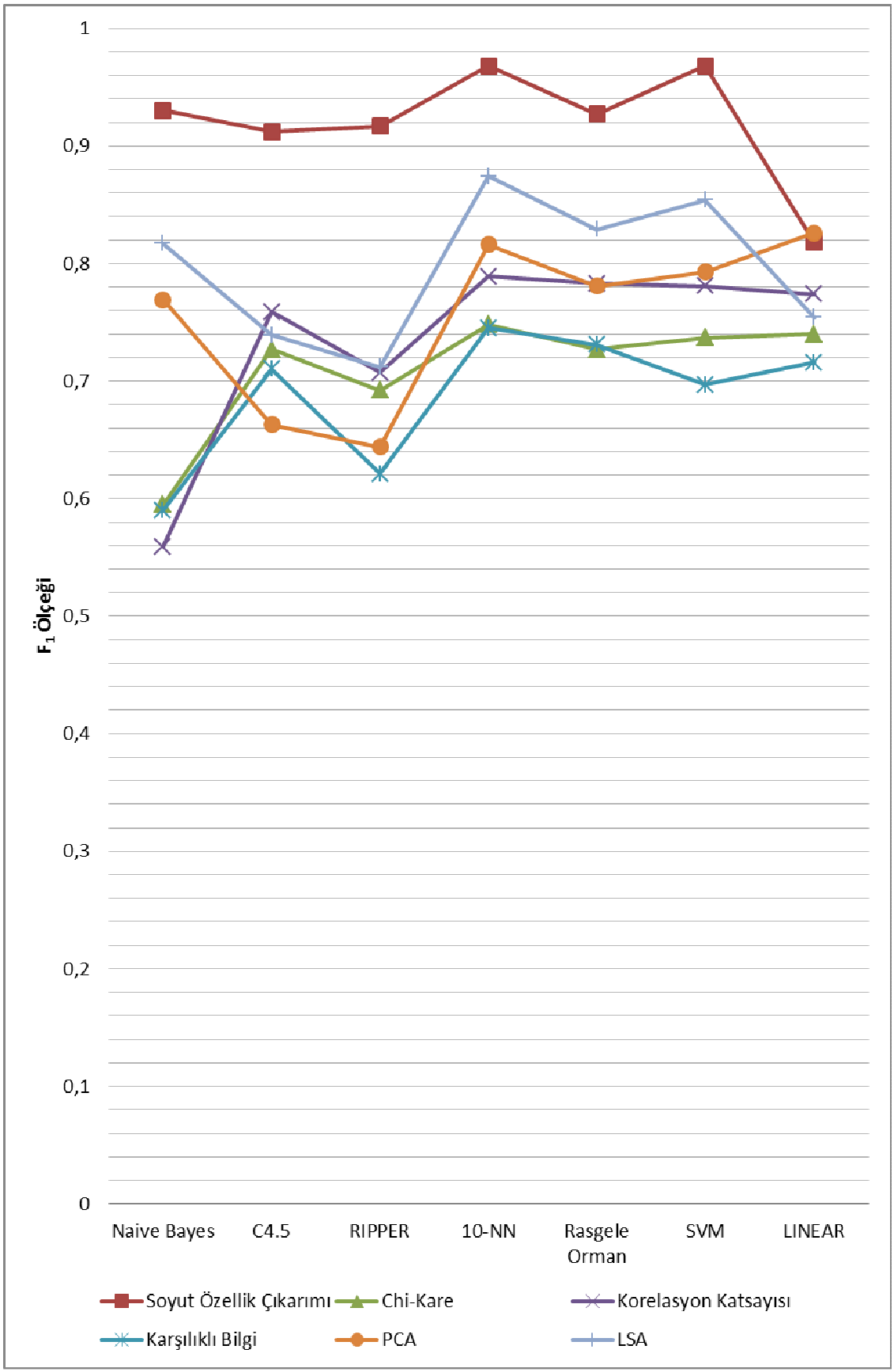
10 özellikle ifade edilen ModApte-10 veri kümesinde elde edilen sonuçlar incelendiğinde yedi sınıflandırıcının en yüksek ortalama F_1 değerini soyut özellik çıkarım algoritmasının ürettiği ve bu değerin en yakın yöntemin ortalamasından 0,060 daha yüksek olduğu görülmektedir. Testlerde elde edilen en yüksek F_1 değeri, 0,961 olarak soyut özellik çıkarımı uygulandıktan sonra çalıştırılan 10 en yakın komşu algoritması ile elde edilmiştir. Sonuçlar sınıflandırıcı bazında incelendiğinde, soyut özellik çıkarım yönteminin diğer tüm yöntemlerden daha iyi performans gösterdiği görülebilir. Performans farkı Şekil 6.10'da görülmektedir.

Sonuçlar incelendiğinde soyut özellik çıkarım yönteminin diğer yöntemlerden daha yüksek başarıma sahip olduğu anlaşılmıştır. Temel bileşen analizi ve saklı anlamsal çözümleme yöntemleri, az sayıda özellikle ifade edilen veri kümelerinde önceki testlere göre daha başarılı sonuçlar üretmiştir. Chi-kare, korelasyon katsayısı ve karşılıklı bilgi yöntemleri ise özellik seçim yöntemleridir. Çok az sayıda seçilen özellikle veri kümesindeki örneklerin sınıflardaki dağılımlarının karakteristikleri yeterince temsil edilemediği için bu yöntemlerin başarılarının çok düşük kaldığı sonucuna ulaşılmıştır.

Bu sonuçlara ilave olarak; her veri kümesinde, varsayılan sayıda özellik ve eşit sayıda özellikle yapılan testlerdeki sonuç sıralamasının genel olarak değişmediği görülmüştür. İyi performans gösteren yöntem ve algoritmaların her iki durumda da daha iyi performans gösterdiği, kötü performans gösteren deney kurulumlarının da her iki durumda da kötü performans sergiledikleri gözlemlenmiştir.

Çizelge 6.9 Seçilen yöntemlerin eşit sayıda özellikle ifade edilen Reuters veri kümesinde sınıflandırma performansına olan etkilerinin karşılaştırılması

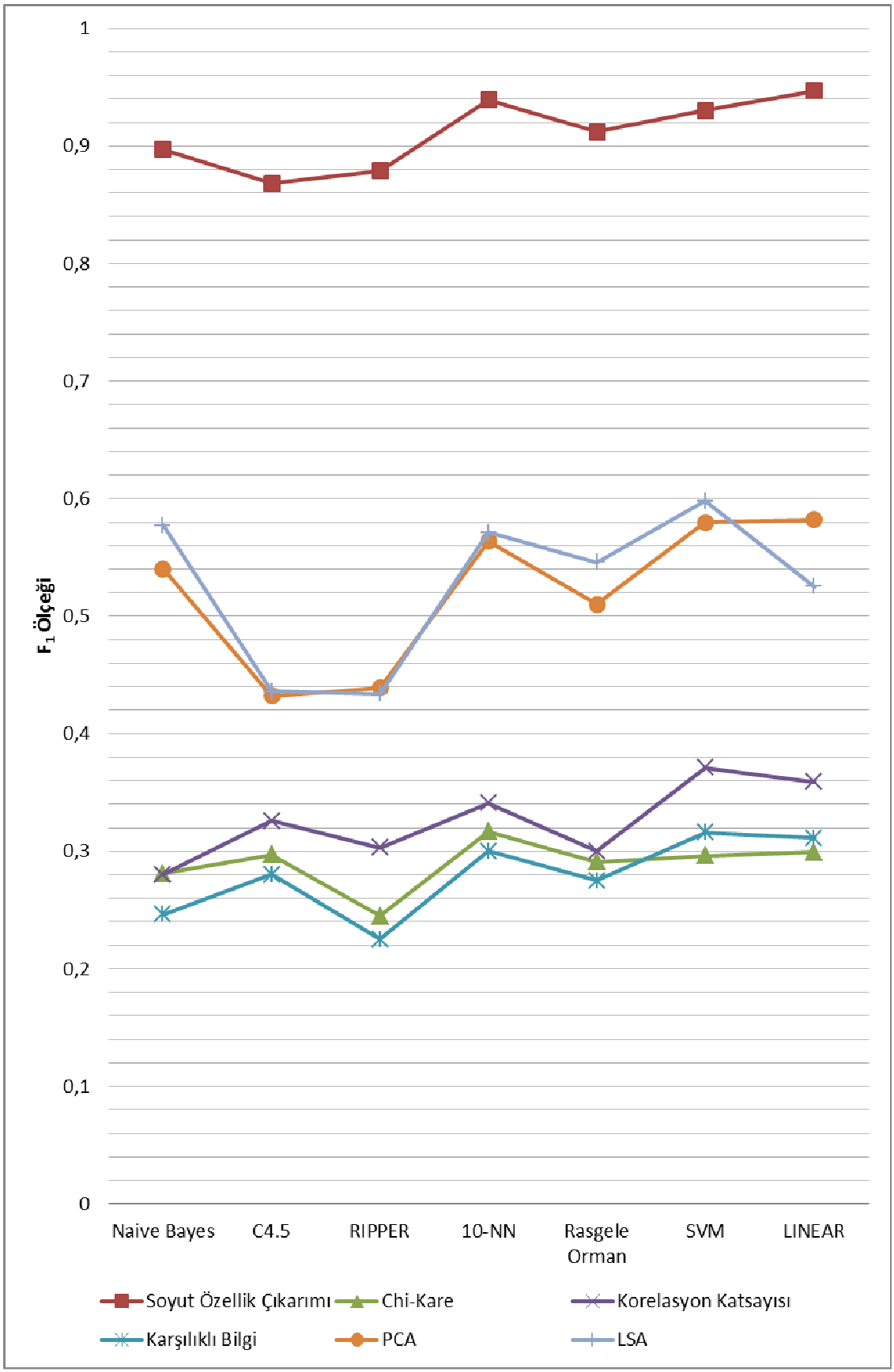
	Soyut Özellik Çıkarımı	Chi-Kare	Korelasyon Katsayısı	Karşılıklı Bilgi	PCA	LSA
Naive Bayes	0,930	0,595	0,559	0,590	0,769	0,817
C4.5	0,912	0,727	0,759	0,710	0,663	0,739
RIPPER	0,917	0,692	0,707	0,621	0,644	0,712
10-NN	**0,968**	**0,748**	**0,789**	**0,745**	0,816	**0,874**
Rasgele Orman	0,927	0,727	0,783	0,731	0,781	0,829
SVM	**0,968**	0,737	0,781	0,697	0,793	0,854
LINEAR	0,818	0,740	0,774	0,716	**0,826**	0,754
Ortalama	**0,920**	0,709	0,736	0,687	0,756	0,797



Şekil 6.8 Eşit sayıda özellikle ifade edilen Reuters veri kümesi üzerinde seçilen yöntemlerin sınıflandırma performansına olan etkisinin görsel olarak karşılaştırılması

Çizelge 6.10 Seçilen yöntemlerin eşit sayıda özellikle ifade edilen 20-Newsgroups veri kümesinde sınıflandırma performansına olan etkilerinin karşılaştırılması

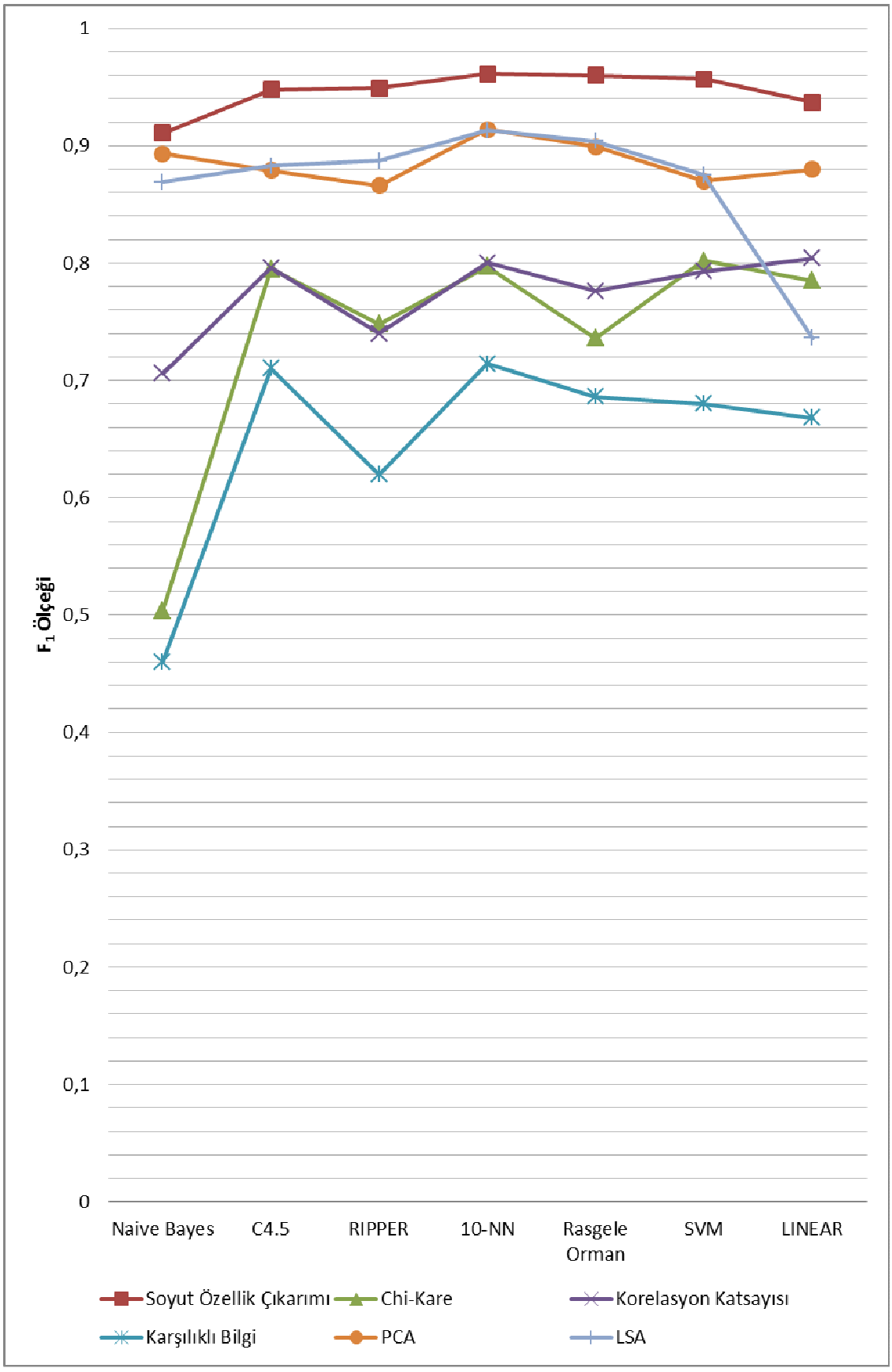
	Soyut Özellik Çıkarımı	Chi-Kare	Korelasyon Katsayısı	Karşılıklı Bilgi	PCA	LSA
Naive Bayes	0,897	0,281	0,280	0,246	0,540	0,577
C4.5	0,868	0,297	0,326	0,280	0,432	0,436
RIPPER	0,879	0,245	0,303	0,225	0,439	0,434
10-NN	0,939	0,317	0,341	0,300	0,564	0,571
Rasgele Orman	0,912	0,291	0,300	0,275	0,510	0,546
SVM	0,930	0,296	**0,371**	**0,316**	0,580	**0,598**
LINEAR	**0,947**	**0,299**	0,359	0,311	**0,582**	0,525
Ortalama	**0,910**	0,289	0,326	0,279	0,521	0,527



Şekil 6.9 Eşit sayıda özellikle ifade edilen 20-Newsgroups veri kümesinde seçilen yöntemlerin sınıflandırma performansına olan etkisinin görsel olarak karşılaştırılması

Çizelge 6.11 Seçilen yöntemlerin eşit sayıda özellikle ifade edilen ModApte-10 veri kümesinde sınıflandırma performansına olan etkisinin karşılaştırılması

	Soyut Özellik Çıkarımı	Chi-Kare	Korelasyon Katsayısı	Karşılıklı Bilgi	PCA	LSA
Naive Bayes	0,911	0,504	0,706	0,460	0,893	0,869
C4.5	0,948	0,795	0,796	0,710	0,879	0,883
RIPPER	0,949	0,748	0,740	0,620	0,866	0,887
10-NN	**0,961**	0,797	0,800	**0,714**	**0,914**	**0,913**
Rasgele Orman	0,960	0,736	0,776	0,686	0,899	0,904
SVM	0,957	**0,802**	0,793	0,680	0,870	0,875
LINEAR	0,937	0,785	**0,804**	0,668	0,880	0,736
Ortalama	**0,946**	0,738	0,774	0,648	0,886	0,867



Şekil 6.10 Eşit sayıda özellikle ifade edilen ModApte-10 veri kümesi üzerinde seçilen yöntemlerin sınıflandırma performansına olan etkisinin görsel olarak karşılaştırılması

6.7 Hipotez Testi Sonuçları

Yapılan testlerin istatistiki açıdan doğruluğunun değerlendirmesini yapmak üzere, Bölüm 5.3'te detayları verilmiş olan Reuters veri kümesi üzerinde seçilen sınıflandırma algoritmaları kullanılarak soyut özellik çıkarım yöntemi ve karşılaştırılan yöntemler arasında hipotez testi uygulanmıştır. Bunun için soyut özellik çıkarım yöntemi baz alınıp, diğer yöntemler ile karşılıklı olarak, t-testi olarak bilinen eşleştirilmiş iki grup arasındaki farkların testi (*paired-samples t-test*) gerçekleştirilmiştir. Bu sayede, farklı iki yöntem ile elde edilen sonuçlar arasındaki farkın belirli bir güven düzeyinde istatistiksel olarak anlamlı olup olmadığı, yoksa rastlantısal mı olduğu test edilmiştir.

T-testini gerçekleştirmek üzere Bölüm 6.3'te detaylarıyla incelenen karşılaştırma testleri 10 kez tekrar edilmiştir. Elde edilen sonuçlar üzerinde yapılan t-testi sonucunda deneylerden elde edilen ortalama F_1 değerleri, bu değerlerin ağırlıklı ortalamaları ve standart sapmaları Çizelge 6.12'de verilmiştir. T-testi için seçilen güven düzeyi %95'tir. Çizelge 6.12'de hücrelerde yer alan değerler 10 testin sonucunda elde edilen ortalama F_1 değerlerini, parantez içinde yer alan değerler de standart sapmaları göstermektedir. Ortalama satırındaki değerler de testler sonucunda elde edilen F_1 değerlerinin ağırlıklı ortalamalarıdır.

Çizelge 6.12'de soyut özellik çıkarım yöntemi ile karşılaştırmalı olarak test edilen diğer yöntemlerin her bir sınıflandırıcı için %95 güven düzeyinde daha iyi, benzer ya da daha kötü başarım sergilediği sonuçları da yer almaktadır. %95 güven düzeyinde anlamlı olmak üzere, karşılaştırılan tüm yöntemlerin; Naive Bayes, C4.5, RIPPER, 10 en yakın komşu, rasgele orman ve SVM sınıflandırıcılarında soyut özellik çıkarım yönteminden daha kötü başarım sergiledikleri t-testi ile kanıtlanmıştır. LINEAR sınıflandırma algoritmasında ise soyut özellik çıkarım yöntemi %95 güven düzeyinde PCA ve LDA ile benzer başarım sergilemiş, geri kalan yöntemlerden ise daha başarısız olmuştur. Bu sonuçlar, Bölüm 6.3'te elde edilen test sonuçlarının doğruluğunu %95 güven düzeyi için kanıtlamaktadır.

Çizelge 6.12 Reuters veri kümesinde t-testi uygulaması sonucu elde edilen ortalama F_1 değerleri, bu değerlerin ağırlıklı ortalamaları ve standart sapmaları

	Soyut Özellik Çıkarımı	İndirgeme Olmadan	Chi-Kare	Korelasyon Katsayısı	Karşılıklı Bilgi	PCA	LSA	LDA
Naive Bayes	0,930 (0,021)	0,724 (0,045) k	0,809 (0,040) k	0,636 (0,025) k	0,809 (0,040) k	0,475 (0,071) k	0,712 (0,028) k	0,579 (0,043) k
C4.5	0,912 (0,020)	0,818 (0,023) k	0,827 (0,030) k	0,803 (0,036) k	0,828 (0,029) k	0,562 (0,058) k	0,688 (0,040) k	0,809 (0,038) k
RIPPER	0,917 (0,023)	0,808 (0,023) k	0,817 (0,029) k	0,773 (0,031) k	0,818 (0,024) k	0,477 (0,063) k	0,681 (0,028) k	0,765 (0,050) k
10-NN	0,968 (0,015)	0,617 (0,031) k	0,780 (0,031) k	0,843 (0,027) k	0,780 (0,031) k	0,677 (0,040) k	0,202 (0,036) k	0,833 (0,040) k
Rasgele Orman	0,927 (0,021)	0,644 (0,035) k	0,815 (0,029) k	0,841 (0,021) k	0,832 (0,033) k	0,661 (0,041) k	0,591 (0,044) k	0,828 (0,034) k
SVM	0,968 (0,015)	0,909 (0,022) k	0,909 (0,025) k	0,852 (0,025) k	0,909 (0,025) k	0,770 (0,053) k	0,858 (0,023) k	0,831 (0,043) k
LINEAR	0,818 (0,013)	0,928 (0,016) i	0,891 (0,020) i	0,865 (0,025) i	0,891 (0,020) i	0,857 (0,041) b	0,906 (0,022) i	0,841 (0,031) b
Ortalama	0,920	0,778	0,835	0,802	0,838	0,640	0,662	0,784
(Daha iyi / Benzer / Daha kötü)	(i / b / k)	(1/0/6)	(1/0/6)	(1/0/6)	(1/0/6)	(0/1/6)	(1/0/6)	(0/1/6)

6.8 Sınıflandırıcı Parametrelerinin Başarıma Olan Etkisinin Testi

Soyut özellik çıkarım yöntemini diğer boyut indirgeme yöntemleri ile karşılaştırmak üzere yapılan testlerde sınıflandırıcılar, parametrelerinin varsayılan ayarları ile kullanılmıştı. Ancak sınıflandırıcı parametrelerinin probleme uygun olarak ayarlanması, bazı durumlarda başarımı arttırabilir. Örneğin, çekirdek temelli bir sınıflandırıcı olan SVM, değişik çekirdeklerle kullanıldığında farklı türdeki problemlere daha iyi uyum sağlayabilmektedir. Tez çalışması kapsamında gerçekleştirilen çalışma ve yapılan karşılaştırmalarda amaç, sınıflandırıcı parametrelerinin en iyi halini bulmak değil, standart parametrelerle çalışan sınıflandırıcılardan önce boyut indirgeyerek başarımı arttırmak olsa da, parametrelerin başarıma olan etkisi de test edilmiştir.

Testi gerçekleştirmek üzere, SVM sınıflandırma algoritması dört farklı çekirdek tipi ile çalıştırılarak soyut özellik çıkarım yöntemi ve diğer yöntemlerle boyutları indirgenen ModApte-10 veri kümesine uygulanmıştır. Karşılaştırması yapılan çekirdek tiplerinden doğrusal çekirdek 6.2'de, çokterimli (*polynomial*) çekirdek 6.3'te, radyal tabanlı fonksiyon çekirdeği 6.4'te ve sigmoid çekirdek 6.5'te verilmiştir.

$$K(u,v) = u^T v \tag{6.2}$$

$$K(u,v) = (\gamma u^T v + r)^d, \gamma > 0 \tag{6.3}$$

$$K(u,v) = \exp(-\gamma \left\| u^T v \right\|^2), \gamma > 0 \tag{6.4}$$

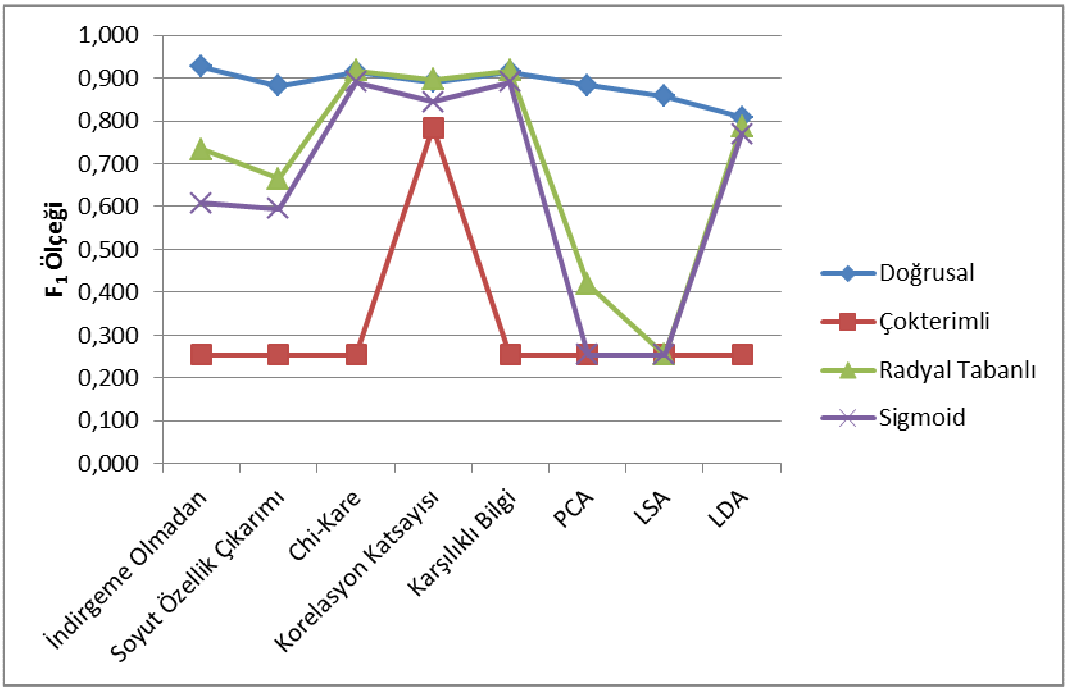
$$K(u,v) = \tanh(\gamma u^T v + r) \tag{6.5}$$

SVM çekirdek tiplerinin sınıflandırma başarımına olan etkisini görmek üzere, karşılaştırılan yöntemlerle boyutları indirgenmiş ModApte-10 veri kümesi üzerinde yapılan testler sonucunda elde edilen F1 değerleri Çizelge 6.13'te verilmiş, Şekil 6.11'de görselleştirilmiştir. Çizelge 6.13 incelendiğinde, beş yöntemin karşılaştırmalarda kullanılan doğrusal çekirdek ile en yüksek başarımı sağladığı, üç yöntemin ise radyal tabanlı fonksiyon çekirdeği ile çok az farkla doğrusal çekirdekten daha başarılı olduğu görülmektedir. Aradaki başarım farkı 0,004 ile 0,006 arasındadır. Bu kadar az bir fark göz ardı edilirse, doğrusal çekirdeğin tüm boyut indirgeme yöntemleri için en uygun

çekirdek tipi olduğu görülmektedir. Bu test sonucunda SVM sınıflandırıcısı için karşılaştırmalı testlerde doğru çekirdek tipinin kullanıldığı da kanıtlanmıştır.

Çizelge 6.13 ModApte-10 veri kümesinde SVM algoritmasının değişik çekirdek tiplerinin uygulanması ile elde edilen F_1 değerleri

	Soyut Özellik Çıkarımı	İndirgeme Olmadan	Chi-Kare	Korelasyon Katsayısı	Karşılıklı Bilgi	PCA	LSA	LDA
Doğrusal	**0,882**	**0,927**	0,914	0,891	0,914	**0,883**	**0,857**	**0,808**
Çokterimli	0,254	0,254	0,254	0,784	0,254	0,254	0,254	0,254
Radyal Tabanlı	0,665	0,734	**0,918**	**0,897**	**0,918**	0,419	0,254	0,788
Sigmoid	0,595	0,608	0,891	0,845	0,891	0,254	0,254	0,769



Şekil 6.11 ModApte-10 veri kümesinde SVM algoritmasının değişik çekirdek tiplerinin uygulanması ile elde edilen F_1 değerlerinin görsel olarak karşılaştırılması

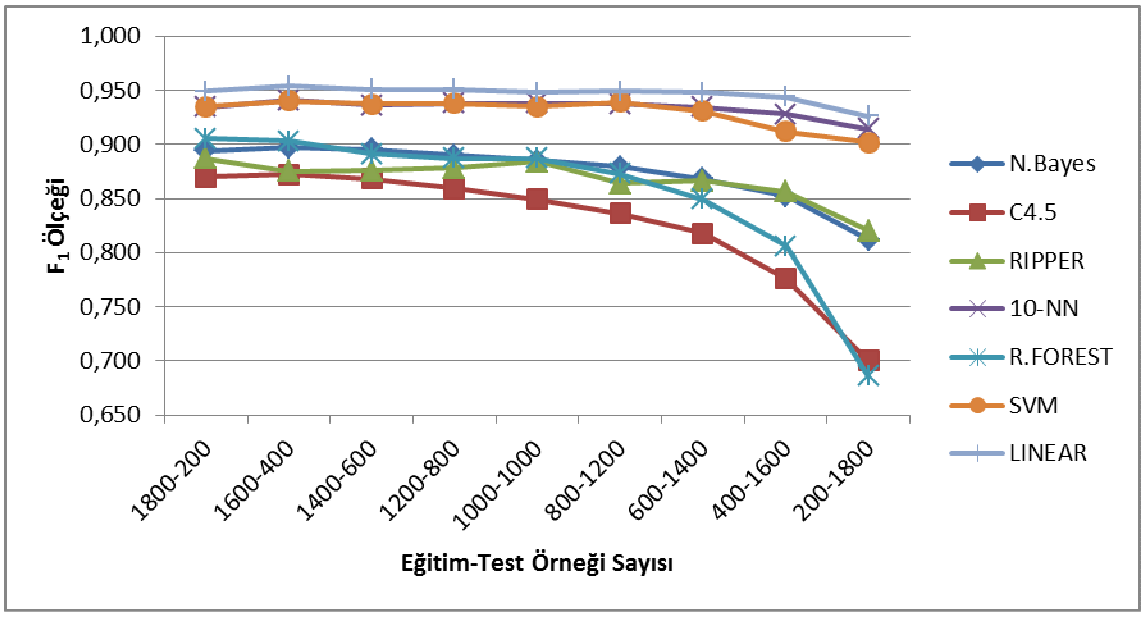
6.9 Eğitim Örneği Sayısının Başarıma Olan Etkisinin Testi

Soyut özellik çıkarım yönteminin sağladığı sınıflandırma başarımının eğitim örneği sayısına olan bağlılığını test etmek için 20-Newsgroups veri kümesi kullanılarak bir deney kurgulanmıştır. Bu deneyde soyut özellik çıkarım yöntemi ile boyutları indirgenmiş olan veri kümesi, 1800 eğitim 200 test belgesinden başlayıp 200 eğitim 1800 test belgesine kadar değişen bir yelpazede eğitim-test demeti şeklinde ayrıştırılmıştır. Seçilen yedi sınıflandırma algoritması, ayrıştırılan veri kümeleri üzerinde onar kez çalıştırılarak elde edilen F_1 değerlerinin ortalaması alınmıştır.

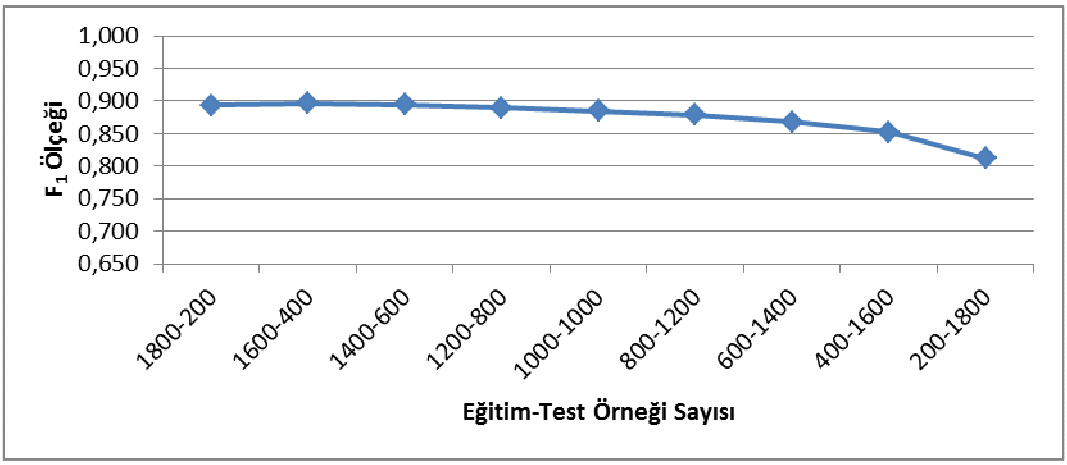
Bu deney sonucunda elde edilen ortalama F_1 değerleri Çizelge 6.14'te verilmiş, Şekil 6.12'de görselleştirilmiştir. Soyut özellik çıkarım yöntemi ile boyutları indirgenmiş olan, değişen sayılarda eğitim ve test örneği içeren veri kümeleri üzerinde sınıflandırma algoritmalarının eğitim örneği sayısına göre başarımlarının değişimleri; Naive Bayes için Şekil 6.13'te, C4.5 için Şekil 6.14'te, RIPPER için Şekil 6.15'te, 10 en yakın komşu için Şekil 6.16'da, rasgele orman için Şekil 6.17'de, SVM için Şekil 6.18'de, LINEAR için Şekil 6.19'da verilmiştir.

Çizelge 6.14 Soyut özellik çıkarımı yöntemi ile elde edilen sınıflandırma başarımının eğitim örneği sayısına göre değişimi

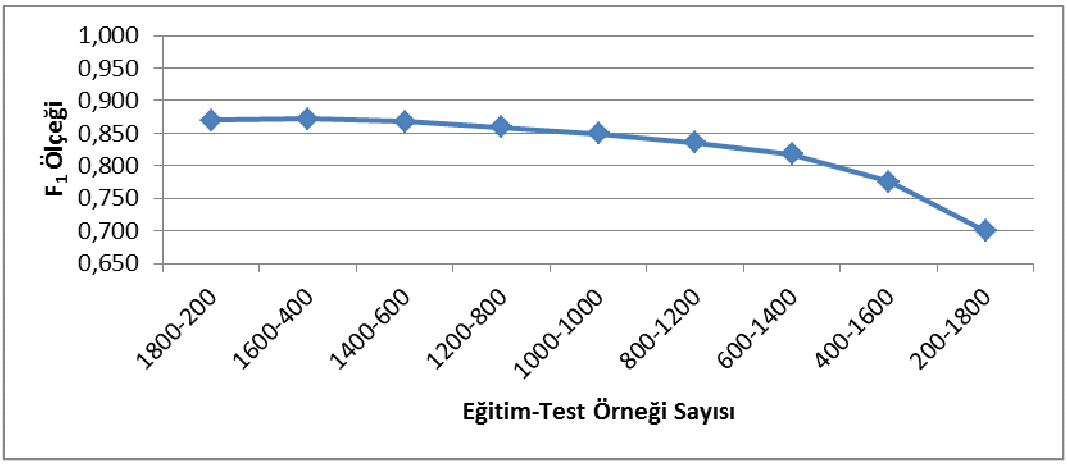
Eğitim Örneği	Test Örneği	**Naive Bayes**	**C4.5**	**RIPPER**	**10-NN**	**Random Forest**	**SVM**	**LINEAR**
1800	**200**	0,894	0,870	0,887	0,934	0,905	0,935	0,949
1600	**400**	0,897	0,872	0,875	0,941	0,903	0,940	0,954
1400	**600**	0,895	0,868	0,876	0,936	0,891	0,937	0,951
1200	**800**	0,890	0,859	0,878	0,938	0,887	0,938	0,951
1000	**1000**	0,885	0,849	0,884	0,938	0,887	0,935	0,948
800	**1200**	0,879	0,836	0,864	0,937	0,873	0,939	0,949
600	**1400**	0,868	0,818	0,866	0,934	0,849	0,931	0,948
400	**1600**	0,852	0,776	0,856	0,928	0,806	0,912	0,944
200	**1800**	0,812	0,700	0,820	0,914	0,686	0,902	0,926



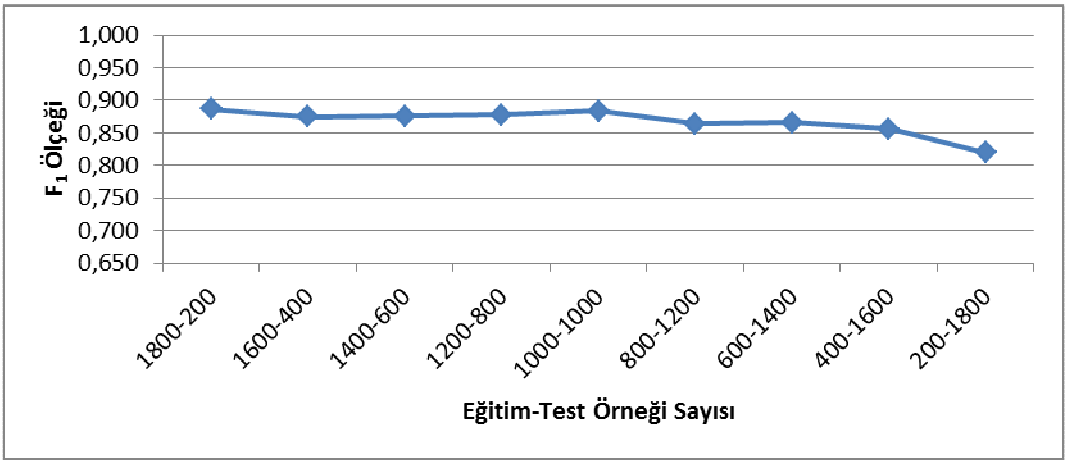
Şekil 6.12 Soyut özellik çıkarımı yöntemi ile elde edilen sınıflandırma başarımının eğitim örneği sayısına göre değişiminin görsel olarak karşılaştırılması



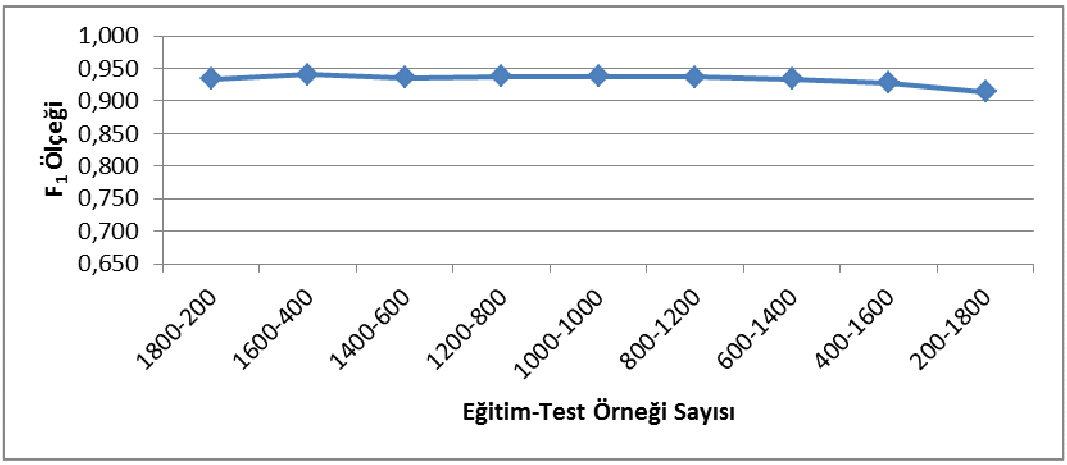
Şekil 6.13 Soyut özellik çıkarımı yöntemi ile Naive Bayes sınıflandırıcısında elde edilen başarımın eğitim örneği sayısına göre değişimi



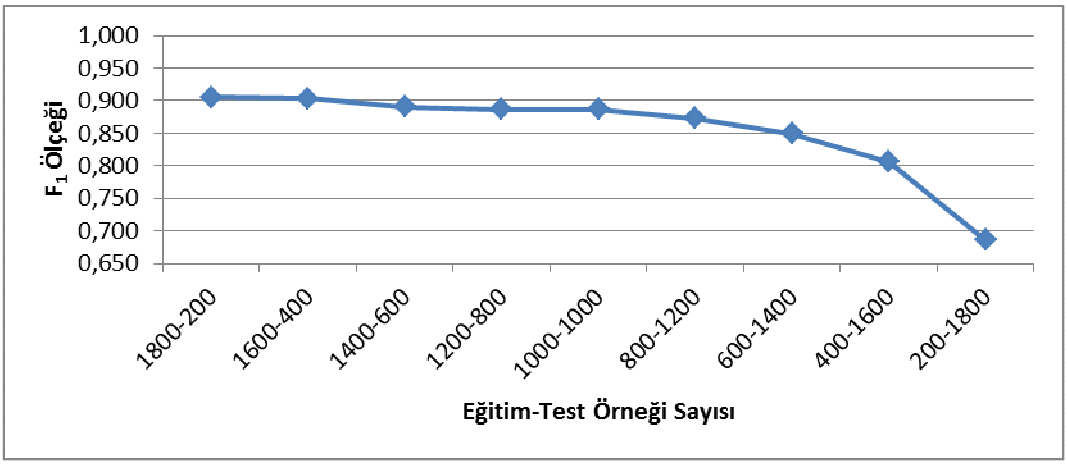
Şekil 6.14 Soyut özellik çıkarımı yöntemi ile C4.5 sınıflandırıcısında elde edilen başarımın eğitim örneği sayısına göre değişimi



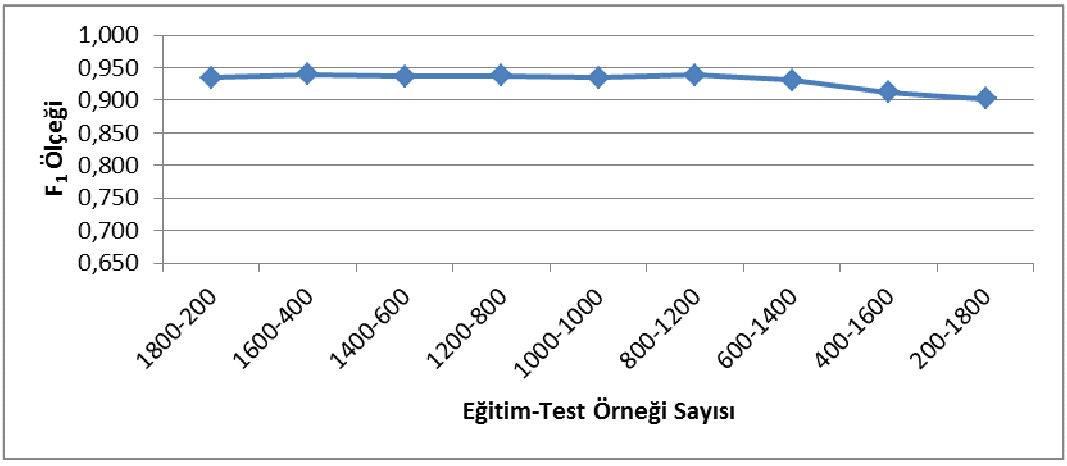
Şekil 6.15 Soyut özellik çıkarımı yöntemi ile RIPPER sınıflandırıcısında elde edilen başarımın eğitim örneği sayısına göre değişimi



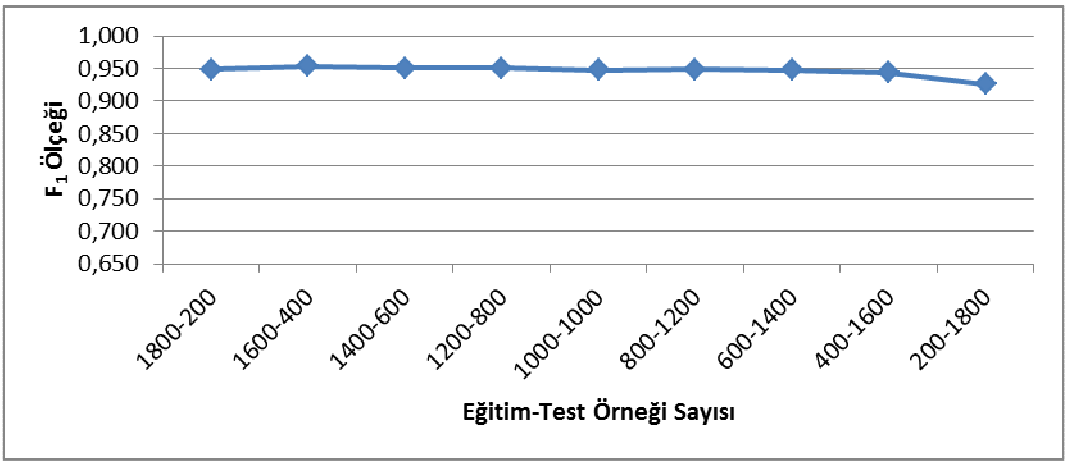
Şekil 6.16 Soyut özellik çıkarımı yöntemi ile 10 en yakın komşu sınıflandırıcısında elde edilen başarımın eğitim örneği sayısına göre değişimi



Şekil 6.17 Soyut özellik çıkarımı yöntemi ile rasgele orman sınıflandırıcısında elde edilen başarımın eğitim örneği sayısına göre değişimi



Şekil 6.18 Soyut özellik çıkarımı yöntemi ile SVM sınıflandırıcısında elde edilen başarımın eğitim örneği sayısına göre değişimi



Şekil 6.19 Soyut özellik çıkarımı yöntemi ile LINEAR sınıflandırıcısında elde edilen başarımın eğitim örneği sayısına göre değişimi

Soyut özellik çıkarım yöntemi ile boyutları indirgenmiş olan veri kümesinde gerçekleştirilen testlerin sonuçları incelendiğinde, ağaç temelli sınıflandırıcılar olan C4.5 ve rasgele orman algoritmalarının başarımlarının eğitim örneği sayısından en çok etkilenen sınıflandırıcılar olduğu görülebilir. Bu algoritmalarda eğitim örneği sayısı test örneği sayısından daha az olduğunda başarım fazlaca düşmektedir.

Kural tabanlı RIPPER ve istatistiki sınıflandırıcı olan Naive Bayes de eğitim örneği sayısındaki düşüşten önceki iki algoritma kadar olmasa da etkilenmektedir. Bu algoritmalarda eğitim örneği sayısı toplam örnek sayısının üçte birinden daha az olduğunda başarımları diğer durumlara kıyasla düşmektedir.

Çekirdek tabanlı sınıflandırıcı olan SVM, doğrusal LINEAR ve örnek temelli en yakın komşu algoritmaları ise soyut özellik çıkarım yöntemi ile birlikte kullanıldıklarında eğitim örneği sayısından en az etkilenen sınıflandırıcılar olarak öne çıkmıştır. Bu sınıflandırıcılar, eğitim örneği sayısı toplam örnek sayısının %10'unun altına düşmediği durumda oldukça başarılı sonuçlar üretmektedir. Toplam örnek sayısının %10'undan daha az örnekle eğitilseler bile önceden anılan dört yöntemden iyi başarım sergileyebilmişlerdir.

Bu deney sonucunda; sınıflandırma işleminden önce boyut indirgeme için soyut özellik çıkarım yöntemi kullanıldığında, eğitim ve test örneği sayısının eşit olması durumunda her çeşit sınıflandırıcının en iyi performansı gösterebileceği sonucuna ulaşılmıştır. SVM,

LINEAR ya da en yakın komşu sınıflandırıcıları kullanıldığında ise, eğitim örneği sayısının çok daha az olması bile başarılı sonuçlar almak için yeterlidir.

BÖLÜM 7

SONUÇ VE ÖNERİLER

Metin işleme uygulamalarında performansı etkileyen ve süreci zorlaştıran en önemli engel verinin yüksek boyutlu olmasıdır. Çok sayıda terimle ifade edilen belgeler üzerinde sınıflandırma gerçekleştirebilmek için gereken işlem gücü ve kaynak miktarı çoğu zaman bu işlemlerin yapılmasını imkansızlaştırmaktadır. Yüksek boyutluluğun getirdiği sorun için uygulanan çözüm, verinin boyutlarının indirgenmesidir. Boyut indirgeme için özellik seçimi ve özellik çıkarımı yaklaşımları bulunmaktadır.

Özellik seçimi ile veriyi diğerlerinden daha iyi tanımlayan özellikler bulunmaya çalışılır. Özellik seçimi yapmak için, sınıflandırıcılar üzerinde denemeler yaparak özelliklerin en iyi alt kümesini elde etmeye çalışan sarmalayıcı yöntemler ve özellikleri dizme yöntemleri kullanarak belirli bir eşik değerinin üzerinde değer alan özellikleri seçen filtre yöntemleri mevcuttur. Hangi yaklaşım olursa olsun, elde edilen iyi ayırt edici bir özellik alt kümesi kategorizasyon işlemlerinin maliyetini düşürür. Metin tipindeki verilerde özellik seçimi yöntemleri belgeleri daha iyi ayırt eden ve daha iyi tanımlayan terimlerin bulunması şeklinde uygulanmaktadır.

Özellik çıkarımı yöntemleri verideki orijinal özelliklerin bileşkesini alarak daha düşük boyutlu yeni bir uzaya taşır. Böylece veri daha az sayıda özellikle ifade edilmiş olur. Ancak oluşan bu yeni özellikler orijinal özelliklerle birebir aynı değildir. Özellik çıkarımı ile özellikler kaynaştırılarak veri için yeterli hassasiyete sahip yeni bir tanım oluşturulur. Metin tipindeki verilerde özellik çıkarımı uygulandığında elde edilen özellikler, terimler kullanılarak elde edilen yeni ve bağımsız tanımlayıcılardır. Çıkarılan özellikle belgelerin karakteristikleri hakkında görülebilir ayırt edici bilgi sunmazlar.

Önceden de belirttiğimiz gibi, orijinal terimlerin veri kümesindeki belgelerdeki dağılımları, belgelerin kategorilere ait olmasında etki sahibidir. Bu noktadan hareketle özellik seçim yöntemleri, ayırt ediciliği en fazla olan terimleri seçmeye, özellik çıkarım yöntemleri ise terimleri kaynaştırıp çeşitli dönüşümler uygulayarak daha düşük sayıda tanımlayıcı özellikler oluşturmaya çalışırlar. Metin işleme için boyut indirgeme çalışmalarında genellikle özellik seçim yöntemleri kullanılmakta, özellik çıkarımı yöntemleri pek tercih edilmemektedir.

Bu tez çalışması kapsamında metin işleme alanı için yeni bir özellik çıkarım yöntemi geliştirmek üzere, belgelerdeki terimlerin içerdiği ayırt edicilik değerlerini kullanarak sınıflara olan etkileri yeni bir uzayda soyut olarak ifade edilmiştir. Bunu gerçekleştirmek üzere ilk olarak terimlerin ayırt edicilikleri ağırlıklandırarak ortaya çıkarılmıştır. Daha sonra belgeler, her bir sınıf için etki değerlerinin bileşkesinden oluşan ve terimlerin ayırt edicilikleri ile orantılı olacak şekilde yeni bir uzayda yer alan özellikler ile ifade edilmiştir. Bu yeni uzayda yer alan çıkarılan özelliklere, belgede bulunan orijinal terimlerin her bir sınıfa olan etkisinin bileşkesini temsil ettiği için soyut özellikler adı verilmiştir. Soyut özellikler, orijinal terimleri ağırlıklandırarak elde edilen etki değerlerini kullanıp boyutları sınıf sayısına eşit bir uzaya doğrusal olarak eşleme ile elde edilmiştir. Bu sayede, soyut özellikler bir belgede her bir sınıf için ne kadar kanıt ya da bilgi bulunduğunu, başka bir deyişle belgenin içerdiği terimlere göre sınıflara ait olma olasılığını göstermektedir.

Soyut özellik çıkarım yönteminin başarımını test etmek ve diğer yöntemlerle karşılaştırmak üzere metin tipinde veri kümeleri üzerinde sınıflandırma testleri gerçekleştirilmiştir. Bu amaçla DMOZ örün dizininin “World/Türkçe” başlığı altında yer alan örün sayfalarının taranması ile Türkçe bir veri kümesi hazırlanmıştır. Ayrıca diğer yöntemlerin başarımları ile standart bir karşılaştırma yapabilmek üzere metin işleme uygulamalarında standart olarak kullanılan Reuters-21578 ve 20-Newgroups veri kümeleri üzerinde testler gerçekleştirilmiştir. Reuters-21578 veri kümesinin ayrıca ModApte-10 olarak bilinen versiyonu da testler için hazırlanarak kullanılmıştır. Tüm veri kümelerinde ön işleme adımları olarak belirtkeleme, filtreleme ve gövdeleme işlemleri gerçekleştirilmiştir. Ayrıca Reuters veri kümesinde sınıfları seçmek ve DMOZ

veri kümesinde gereksiz örnekleri elemek amacıyla kutu çizimi filtresinden ön işleme adımları içinde faydalanılmıştır.

Metin işleme uygulamalarındaki başarımı test etmek ve karşılaştırmak üzere seçilen boyut indirgeme yöntemleri ve soyut özellik çıkarım yöntemi sınıflandırma işlemlerinden önce uygulanmıştır. Özellik seçim yöntemleri olarak chi-kare, korelasyon katsayısı ve karşılıklı bilgi, özellik çıkarım yöntemleri olarak da PCA, LSA ve LDA karşılaştırmalı testlere dahil edilmiştir. Sınıflandırma işlemlerine olan etkileri ölçmek için; istatistiki sınıflandırıcı olarak Naive Bayes, karar ağacı olarak C4.5, kural tabanlı sınıflandırıcı olarak RIPPER, örnek temelli yöntem olarak 10 en yakın komşu, kontrollü varyasyonlara sahip karar ağaçları koleksiyonu için rastgele orman, çekirdek tabanlı sınıflandırıcı olarak destek vektör makineleri, doğrusal sınıflandırıcı olarak LINEAR kullanılmıştır. Deneylerde ModApte-10 veri kümesine standart eğitim-test kümeleri kullanılmış, bunun dışındaki veri kümelerinde 10 kere çapraz doğrulama ile testler gerçekleştirilmiştir.

DMOZ veri kümesindeki sınıflandırma testlerinin sonuçlarına göre soyut özellik çıkarımı kullanıldığında elde edilen en yüksek F_1 değeri, kullanılmadığı zamana göre 0,031 daha yüksektir. Sınıflandırıcıların tümü göz önüne alındığında, soyut özellik çıkarım yönteminin ortalama olarak F_1 değerinde 0,179 artış sağladığı gözlemlenmiştir.

Reuters veri kümesinde yapılan karşılaştırma testlerine göre soyut özellik çıkarım yöntemi en başarılı F_1 değerini üreten takipçisinden 0,037 daha yüksek bir F_1 değeri üretmiştir. Üstelik soyut özellik çıkarımı uygulandığında neredeyse bütün sınıflandırıcıların en iyi performansı sergilediği görülmüştür. Tüm sınıflandırıcıların ortalaması değerlendirildiğinde soyut özellik çıkarımının ürettiği ortalama F_1 değeri, diğer tüm yöntemlerin ortalamasından yüksek, en yakın yöntemin ortalamasından da 0,080 daha iyidir. Eşit sayıda özellikte yöntemlerin başarımları karşılaştırıldığında, 21 özellikle ifade edilen Reuters veri kümesindeki performans sonuçlarına göre soyut özellik çıkarım yöntemi en yakın takipçisinden ortalamada 0,123 daha yüksek F_1 değeri üretmiştir.

20-Newsgroups veri kümesinde yapılan karşılaştırma testlerinde tüm sınıflandırıcıların ortalamasına göre soyut özellik çıkarımı ile elde edilen ortalama F_1 değeri en yakın

yöntemden 0,378 daha yüksektir. Bu veri kümesinde soyut özellik çıkarım yöntemi, en yakın takipçisinden 0,194 daha fazla olacak şekilde en yüksek F_1 değerini de üretmiştir. Eşit sayıda özellikte yöntemlerin başarımları karşılaştırıldığında, 20 özellikle ifade edilen 20-Newsgroups veri kümesindeki ortalama F_1 değerlerine göre, soyut özellik çıkarım yöntemi en yakın takipçisinden 0,383 daha yüksek F_1 değeri sağlamıştır.

ModApte-10 veri kümesinde soyut özellik çıkarım yöntemi en yüksek F_1 değerini üreten takipçisinden 0,034 daha yüksek F_1 değeri üretmiştir. Ortalama F_1 değerlerine göre de soyut özellik çıkarım yöntemi 0,068 daha yüksek sonuç vermiştir. Eşit sayıda özellikte yöntemlerin başarımları karşılaştırıldığında, 10 özellikle ifade edilen ModApte-10 veri kümesindeki F_1 değeri sonuçlarına göre soyut özellik çıkarım yöntemi en yakın takipçisinden ortalamada 0,06, daha yüksek F_1 değeri üretmiştir.

Yapılan test sonuçlarından anlaşılacağı üzere soyut özellik çıkarım yöntemi veri kümelerini metin işleme uygulamalarına efektif olarak hazırlamak için kullanılabilir. Bunun yanında yöntem sınıfların ayrılabilirliği hakkında da bilgi vermektedir. Bir veri kümesindeki eğitim örnekleri, ait oldukları sınıflar hakkında taşıdıkları özelliklerin bileşkesinde gizli olan bilgiyi içermektedir. Soyut özellik çıkarım yöntemi ile bu bilgi açığa çıkarılmaktadır. Yöntem ile ortaya çıkarılan soyut özellikler, örneklerin kendi sınıfına ve diğer sınıflara ait olma olasılıkları olarak da değerlendirilebilir. Örneklerdeki soyut özelliklerin değerleri birbirine yakın olduğunda sınıfların ayrılabilirliği az olmaktadır. Soyut özelliklerin değerleri arasındaki farklar büyüdükçe sınıfları bağımsız olarak ayırt etmek kolaylaşmaktadır. Bu gözleme dayanarak, soyut özelliklere bakılarak sınıfların ayrılabilirliği konusunda yorum yapmak mümkün olmaktadır.

Soyut özellik çıkarım yöntemini daha da geliştirmek üzere yapılabilecek gelecek çalışmalar, yöntemi metin işleme dışında görüntü işleme, ses işleme gibi alanlarda kullanılabilecek şekilde uyarlamaktır. Bunun dışında hiyerarşik sınıflandırma ve çok sınıflı verilerin sınıflandırılması için uyarlamalar yapılabilir. Soyut özellik çıkarım yöntemi, veri kümesinde sınıf etiketinin eksik olduğu örneklerin kestirimi için de kullanılabilir. Veri kümesini sınıf sayısı kadar boyuta indirgemek yerine farklı özelliklerin de sınıf bilgisi gibi devreye alınmasıyla veriyi farklı boyut sayılarına indirgemek ve

sonuçları test ederek başarımı karşılaştırmak, soyut özellik çıkarım yöntemini daha da geliştirmek üzere gerçekleştirilebilecek bir çalışmadır.

KAYNAKLAR

[1] Akverdi, H., (1997), "Eflâtun Phaidros", 274 e; 275 a-b. Milli Eğitim Bakanlığı, İstanbul.

[2] IDC, The Diverse.and Exploding Digital Universe, http://www.emc.com/collateral/analyst-reports/diverse-exploding-digital-universe.pdf, 18 Nisan 2011.

[3] Guyon, I., (2003). "An Introduction to Variable and Feature Selection", J. Of Machine Learning Research, 3: 1157-1182.

[4] Chen, X. ve Wasikowski, M., (2008). "FAST: A Roc-Based Feature Selection Metric for Small Samples and Imbalanced Data Classification Problems", ACM SIGKDD Conference, 24-27 August 2008, Las Vegas-Nevada.

[5] Landauer, T. K. ve Dumais, S. T., (1997). "A Solution to Pluto's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representetion of Knowledge", Psychological Review, 104(2): 211–240.

[6] Hofmann, T., (1999). "Probabilistic Latent Semantic Indexing", SIGIR-99 Conference , 15-19 August 1999, Berkeley-California.

[7] Fodor, I., (2002). A Survey of Dimension Reduction Techniques, https://computation.llnl.gov/casc/sapphire/pubs/148494.pdf, 22 Nisan 2011.

[8] Dragos, A. M., (1998). "Feature Extraction – A Pattern for Information Retrieval", Proceedings of 5th Pattern Languages of Programs Conference, Monticello-Illionis, August 1998.

[9] Tsai, F. S., (2011). "Dimensionality Reduction Techniques for Blog Visualization", Expert Systems with Applications, 38(3):2766-2773.

[10] Salton, G. ve Buckley, C., (1988). "Term-weighting approaches in automatic text retrieval". Information Processing & Management 24(5): 513-523.

[11] Lan, M., Tan, C. L., Su, J. ve Lu, Y., (2009). "Supervised and Traditional Term Weighting Methods for Automatic Text Categorization", IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(4):721-735.

[12] Salton, G. ve McGill, M. J., (1983). Introduction to Modern Information Retrieval, McGraw-Hill, New York.

[13] Aizawa, A., (2003). “An Information-Theoretic Perspective of TFIDF Measures”, Information Processing & Management, 39:45-65.

[14] Li, Z., Xiong, Z., Zhang, Y., Liu, C. ve Li, K., (2011). “Fast Text Categorization Using Concise Semantic Analysis”, Pattern Recognition Letters, 32(3):441-448.

[15] Tonta, Y., (2001). “Bilgi Erişim Sorunu, 21.Yüzyıla Girerken Enformasyon Olgusu”, Ulusal Sempozyum, 19-20 Nisan 2001, Hatay.

[16] Türk Dil Kurumu, Genel Türkçe Sözlük, http://www.tdk.gov.tr/TR/Genel/SozBul.aspx?F6E10F8892433CFFAAF6AA849816B2EF4376734BED947CDE&Kelime=bilgi, 27 Nisan 2011.

[17] Buckland, M., (1991). Information and Information Systems, Praeger, New York.

[18] Dabney, D.P., (1986). “The Curse of Thamus: An Analysis of Full-Text Legal Document Retrieval”, Law Library Journal, 78(5):5-40.

[19] Tonta, Y., (1988). “Kütüphaneler İnsanlığın Ortak Belleğidir”, Öğretmen Dünyası, 99:25-26.

[20] Kochen, M., (1967). “The Growth of Knowledge: Readings on Organization and Retrieval of Information” içinde “Wells, H.G., World Encyclopedia, 11-22”, Wiley, New York.

[21] Bush, V. “As We May Think”, http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm, 27 Nisan 2011.

[22] Varian, H., (1995). “The Information Economy”, Scientific American, 273:161-162.

[23] Kapor, M., “Quotations About the Internet”, http://www.quotegarden.com/internet.html, 27 Nisan 2011.

[24] Tonta, Y., Bitirim, Y., ve Sever, H., (2002). “Türkçe Arama Motorlarında Performans Değerlendirme”, Total Bilişim Ltd. Şti., Ankara.

[25] Garcia, E., “Document Indexing Tutorial”, http://www.miislita.com/information-retrieval-tutorial/indexing.html, 27 Nisan 2011.

[26] Sezer, E., (1999). SMART Bilgi Erişim Sisteminin Türkçe Yerelleştirilmesi ve Otomatik Gömü Üretimi, Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara.

[27] Garcia, E., “Term Vector Theory and Keyword Weights”, http://www.miislita.com/term-vector/term-vector-1.html, 27 Nisan 2011.

[28] Kobayashi, M. ve Takeda, K., (2000). “Information Retrieval on the Web”, ACM Computing Surveys 32(2):144-173.

[29] Plesu, A., “How Big is the Internet”, http://news.softpedia.com/news/How-Big-Is-the-Internet-10177.shtml, 27 Nisan 2011.

[30] Edwards, J., McCurley, K.S. ve Tomlin, J.A., (2001). “An Adaptive Model for Optimizing Performance of an Incremental Web Crawler”, Proceedings of 10th Conference on World Wide Web, 2001, 1-5 May 2001, Hong Kong.

[31] Lawrence, S. ve Giles, C.L., (2000). “Accesibility of Information on the Web”, Intelligence, 11(1):32-39.

[32] Brin, S. ve Page, L., (1998). “The Anatomy of a Large-Scale Hypertextual Web Search Engine”, Computer Networks and ISDN Systems, 30(1-7):107-117.

[33] Abiteboul, S., Preda, M. ve Cobena, G., (2003). “Adaptive Online Page Importance Computation”, WWW-2003 Conference, 20-24 May 2003, Budapest, Hungary.

[34] Kahle, B., “Archiving The Internet”, http://www.archive.org/sciam_article.html, 27 Nisan 2011.

[35] Kahle, B., (1997). “Preserving the Internet”, Scientific American, 276(3):82-83.

[36] Cho, J. ve Garcia-Molina, H., (2003). “Effective Page Refresh Policies for Web Crawlers”, ACM Transitions on Database Systems, 28(4): 390-426.

[37] Koster, M, “A Standard for Robot Exclusion”, http://www.robotstxt.org/orig.html, 27 Nisan 2011.

[38} Pinkerton, B, (1994). “Finding What People Want: Experiences with the WebCrawler”, Proceedings of 1st WWW Conference, 25-27 May 1994, Geneva, Switzerland.

[39] Heydon, A. ve Najork, M., (1999). “Mercator: A Scalable, Extensible Web Crawler”, World Wide Web, 2(4): 219-229.

[40] Boldi, P., Codenotti, B., Santini, M. ve Vigna, S., (2004). “UbiCrawler: A Scalable Fully Distributed Web Crawler”, Software, Practice and Experience, 34(8), 711-726.

[41] Apache Nutch Projesi Ana Sayfası, http://nutch.apache.org/, 27 Nisan 2011.

[42] Miller, R. ve Bharat, K., (1998), “Sphinx: A Framework for Creating Personal, Site-Specific Web Crawlers”, Proceedings of 7th WWW Conference, 1998, 14-18 Nisan 1998, Brisbane, Australia.

[43] Baeza-Yates, R. ve Castillo, C., (2002). “Soft Computing Systems – Design, Management and Applications“ içinde “Balancing Volume, Quality and Freshness in Web Crawling, 565-572”, IOS Press, Santiago, Şili.

[44] Nanas, N., Uren, V. ve De Roeck, A., (2004). “A Comparative Evaluation of Term Weighting Methods for Information Filtering”, Proceedings of 15th DEXA-04, 30 August - 3 September 2004, Saragosa, Spain.

[45] Salton, G., Wong, A. ve Yang, C.S., (1975). “A Vector Space Model for Automatic Indexing”, Communications of the ACM, 18(11): 613-620.

[46] Manning, C.D., Raghavan, P. ve Schütze, H., (2009). An Introduction to Information Retrieval, Cambridge University Press, Cambridge, England.

[47] Sanderson, M. ve Ruthven, I., (1996). “Report on the Glasgow IR Group Submission”, Proceedings of 5th TREC Conference, 20-22 November 2011, Gaithersburg, Maryland.

[48] Hanczar, B., Hua, J., Sima, C., Weinstein, J., Bittner, M. ve Dougherty, E.R., (2010). "Small-sample Precision of ROC-related Estimates", Bioinformatics, 26(6): 822-830.

[49] Lobo, J. M., Jiménez-Valverde, A. ve Real, R., (2008). "AUC: A Misleading Measure of the Performance of Predictive Distribution Models", Global Ecology and Biogeography, 17: 145–151.

[50] Masand, B., Linoff, G. ve Waltz, D., (1992). "Classifying News Stories Using Memory Based Reasoning", 15th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval, 21-24 June 1992, Copenhagen, Denmark.

[51] Yang, Y. ve Pedersen, J.O., (1997). "A Comperative Study on Feature Selection in Text Categorization", 14th International Conference on Machine Learning, 8-12 July 1997, Nashville, Tennessee, USA.

[52] Yang, Y., (1999). "An Evaluation of Statistical Approaches to Text Categorization", Journal of Information Retrieval , 1(1-2): 69–90.

[53] Joachims, T., (1998). "Text Categorization with Support Vector Machines: Learning with Many Relevant Features", Proceedings of 10th European Conference on Machine Learning (ECML), 21-23 April 1998, Chemnitz, Germany.

[54] McCallum, A., Nigam, K., (1998). "A Comparison of Event Models for Naive Bayes Text Classification". Proceedings of AAAI/ICML-98 Workshop on Learning for Text Categorization, 24-27 July 1998, Madison, Wisconsin, USA.

[55] Koller, D., Sahami, M., (1997). "Hierarchically Classifying Documents Using Very Few Words", Proceedings of 14th International Conference on Machine Learning, 8-12 July 1997, Nashville, Tennessee, USA.

[56] Ng, H. T., Goh, W.B., ve Low, K.L., (1997). "Feature Selection, Perceptron Learning, and a Usability Case Study for Text Categorization", 20th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval, 27-31 July 1997, Philadelphia, USA.

[57] Wiener, E., Pedersen, J.O., ve Weigend, A.S., (1995). "A Neural Network Approach to Topic Spotting", Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval, 24-26 April 1995, Las Vegas, USA.

[58] Yang, Y., Liu, X., (1999). "A Re-examination of Text Categorization Methods", Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval, 15-19 August 1999, Berkeley, USA.

[59] Fürnkranz, J., (1999). "Exploiting Structural Information for Text Classification on the WWW", Proceedings of IDA-99, 3rd Symposium on Intelligent Data Analysis, August 1999, Amsterdam, Netherlands.

[60] Joachims, T., Cristianini, N., ve Shawe-Taylor, J., (2001). "Composite Kernels for Hypertext Categorisation", Proceedings of the 18th International Conference on Machine Learning, 28 June 1 July 2001, Williamstown, USA.

[61] Dumais, S., ve Chen, H., (2000), "Hierarchical Classification of web Content", In Proc. of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval, 24-28 July 2000, Athens, Greece.

[62] Mladenic, D., (1998). "Turning Yahoo into an Automatic Web-Page Classifier", 13th European Conference on Artificial Intelligence Young Researcher Paper, 23-28 August 1998, Brihton, UK.

[63] Holden, N., Freitas, A.A., (2004). "Web Page Classification with an Ant Colony Algorithm", Parallel Problem Solving from Nature – PPSN VIII, vol.3242, Springer Berlin-Heidelberg, 1092-1102.

[64] Zamir, O., Etzioni, O., (1998). "Web Document Clustering: A Feasibility Demonstration", Proceedings of 21st Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval, 24-28 August 1998, Melbourne, Avusturalia.

[65] Schenker, A., Last, M., Kandel, A., (2005). "Design and Implementation of a Web Mining System for Organizing Search Engine Results", International Journal Of Intelligent Systems, 20:607-625.

[66] Zhang, D., Dong, Y., (2004). "Semantic, Hierarchical, Online Clustering of Web Search Results", Proceedings of the 6th Asia Pacific Web Conference, 14-17 April 2004, Hangzhou, China.

[67] Zhu, J., Wang, H., ve Zhang, X., (2006). "Discrimination-Based Feature Selection for Multinomial Naïve Bayes Text Classification", LNAI, 4285: 149-156.

[68] Jensen, R., Shen, Q., (2008). "Computational Intelligence and Feature Selection Rough and Fuzzy Approaches", IEEE-Wiley, New Jersey.

[69] Guyon, I., Bitter, H.M., Ahmed, Z., Brown, M. Ve Heller, J., (2005). "Multivariate Non-Linear Feature Selection with Kernel Methods", Studies in Fuzziness and Soft Computing, 164: 313-326.

[70] Hall, M.A., Smith, L.A. (1998). "Practical Feature Subset Selection for Machine Learning", 21st Australian Computer Science Conference, 4-6 February 1998, Perth, Australia.

[71] Zheng, Z., Wu, X. ve Srihari, R., (2004). "Feature Selection for Text Categorization on Imbalanced Data", ACM SIGKDD Exploraitons Newsletter, 6(1):80-89.

[72] Pearson, K., (1901). "On Lines and Planes of Closest Fit to Systems of Points in Space", Philosophical Magazine, 2(6): 559-572.

[73] Fukunaga, K., (1990), Introduction to Statistical Pattern Recognition, Academic Press, Londra.

[74] Martinez, A. M. ve Kak, A.C., (2001). "PCA versus LDA", IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(2):228-233.

[75] Kaban, Z. ve Diri, B., (2008). "Recognizing Type and Author in Turkish Documents Using Artificial Immune Systems", 16.Sinyal İşleme, İletişim ve Uygulamaları Kurultayı, 20-22 Nisan 2008, ODTÜ, Aydın, Türkiye.

[76] Porter, M.F., (1980). "An Algorithm for Suffix Stripping", Program, 14(3):130-137.

[77] Zemberek, http://code.google.com/p/zemberek, 26 Mayıs 2011.

[78] Tukey, J.W. (1977). “Exploratory Data Analysis”, Addison Wesley, Reading, Massachuttes.

[79] Frigge, M., Hoaglin, D.C., Iglewicz, B., (1989). “Some Implementations of the Boxplot”, The American Statistician, 43(1):50-54.

[80] DMOZ Ana Sayfası, http://www.dmoz.org, 24 Mayıs 2011.

[81] HTML Parser, http://htmlparser.sourceforge.net, 26 Mayıs 2011.

[82] Reuters-21578 Veri kümesi, http://www.daviddlewis.com/resources/testcollections/reuters21578/, 26 Mayıs 2011.

[83] 20 Newsgroups veri kümesi, http://people.csail.mit.edu/jrennie/20Newsgroups/, 26 Mayıs 2011.

[84] 20 Newsgroups özet veri kümesi, http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html, 26 Mayıs 2011.

[85] Quinlan, J.R., (1993). C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers.

[86] Cohen, W.W., (1995). “Fast Effective Rule Induction”, 21st International Conference on Machine Learning, 4-8 July 2004, Banff, Alberta, Canada.

[87] Breiman, L., (2001). "Random Forests". Machine Learning, vol:45, no:1, 5-32.

[88] Cortes, C., Vapnik, V., (1995), “Support-Vector Networks”, Machine Learning, 20(3):273-297.

[89] Fan, R.E., Chang, K.W., Hsieh, X,R., Wang, Lin, C.J., (2008). “LIBLINEAR: A Library for Large Linear Classification”, Journal of Machine Learning Research, 9:1871-1874.

[90] Bishop, Christopher M. (2006). Pattern Recognition and Machine Learning, Springer.

[91] Lloyd, S.P., (1982). Least squares quantization in PCM, IEEE Transactions on Information Theory, 28(2):129–136.

[92] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. ve Witten, I.H., (2009). The WEKA Data Mining Software: An Update, SIGKDD Explorations, 11(1):10-18.

[93] Biricik, G. ve Diri, B., (2009). “Impact Of a New Attribute Extraction Algorithm on Web Page Classification”, 5th International Conference on Data Mining,13-16 July 2009, Las Vegas, USA.

[94] Biricik, G., Diri, B. ve Sönmez, A.C., “A New Method for Attribute Extraction with Application on Text Classification”, ICSCCW 2009, 2-4 September 2009, Famagusta, TRNC.

EK-A

VERİ KÜMELERİNDEN ÇIKARILAN SIK KULLANILAN KELİMELER

Veri kümelerindeki belgelerde yer alan ve başarımı düşürdüğü için çıkarılan sık kullanılan kelimelerin listeleri, Türkçe için A-1'de, İngilizce için A-2'de verilmiştir.

A-1 Türkçe Sık Kullanılan Kelimeler

ait	ama	amma	anca	ancağ
ancak	bari	başkası	be	ben
bende	bide	birazı	birçoğ	birçoğu
birçok	biri	birisi	bu	buna
dandini	çoğu	çünkü	da	değ
dek	değin	diye	doğrusu	eğer
eh	en	etkili	fakad	fakat
falan	falanca	filan	filanca	gereğ
gerek	gibi	göre	görece	hakeza
hakkında	halbuki	hangisi	hasebiyle	hatime
hatta	hele	hem	hepsi	hiçbiri

hoş	için	içre	ile	ila
indinde	intağ	intak	imdi	ise
ister	kaçı	kadar	kah	karşın
keşke	keşki	keza	kezaliğ	kezalik
ki	kimisi	kimse	kelli	kendi
kim	kimi	lakin	madem	mademki
mamafih	meğer	meğerki	meğerse	mu
mı	mi	mü	neden	nedense
neye	neyi	nitekim	ne	nere
nerede	nereden	nereli	neresi	nereye
ona	onda	ondan	onlar	onu
onun	oysa	oysaki	öbürkü	öbürü
ötekisi	pad	pat	peki	sana
sakın	sen	siz	ta	şu
şuna	şunda	şundan	şunlar	şunu
şunun	uyarınca	üsd	üst	ve
vesselam	veya	veyahud	veyahut	ya
velev	velhasıl	velhasılıkelam	yazığ	yazık
yekdiğeri	yoksa	yukarda	yukardan	yukarıda
yukarıdan	zinhar	zira		

A-2 İngilizce Sık Kullanılan Kelimeler

a	about	above	across	after
afterwards	again	against	all	almost
alone	along	already	also	although
always	am	among	amongst	amoungst
amount	an	and	another	any
anyhow	anyone	anything	anyway	anywhere
are	around	as	at	back
be	became	because	become	becomes
becoming	been	before	beforehand	behind
being	below	beside	besides	between
beyond	bill	both	bottom	but
by	call	can	cannot	cant
co	computer	con	could	couldnt
cry	de	describe	detail	do
done	down	due	during	each
eg	eight	either	eleven	else
elsewhere	empty	enough	etc	even
ever	every	everyone	everything	everywhere
except	few	fifteen	fify	fill

find	fire	first	five	for
former	formerly	forty	found	four
from	front	full	further	get
give	go	had	has	hasnt
have	he	hence	her	here
hereafter	hereby	herein	hereupon	hers
herself	him	himself	his	how
however	hundred	i	ie	if
in	inc	indeed	interest	into
is	it	its	itself	keep
last	latter	latterly	least	less
ltd	made	many	may	me
meanwhile	might	mill	mine	more
moreover	most	mostly	move	much
must	my	myself	name	namely
neither	never	nevertheless	next	nine
no	nobody	none	noone	nor
not	nothing	now	nowhere	of
off	often	on	once	one
only	onto	or	other	others

otherwise	our	ours	ourselves	out
over	own	part	per	perhaps
please	put	rather	re	same
see	seem	seemed	seeming	seems
serious	several	she	should	show
side	since	sincere	six	sixty
so	some	somehow	someone	something
sometime	sometimes	somewhere	still	such
system	take	ten	than	that
the	their	them	themselves	then
thence	there	thereafter	thereby	therefore
therein	thereupon	these	they	thick
thin	third	this	those	though
three	through	throughout	thru	thus
to	together	too	top	toward
towards	twelve	twenty	two	un
under	until	up	upon	us
very	via	was	we	well
were	what	whatever	when	whence
whenever	where	whereafter	whereas	whereby

wherein	whereupon	wherever	whether	which
while	whither	who	whoever	whole
whom	whose	why	will	with
within	without	would	yet	you
your	yours	yourself	yourselves	a
b	c	d	e	f
g	h	i	j	k
l	m	n	o	p
q	r	s	t	u
v	w	x	y	z

EK-B

ÇIKARILAN SOYUT ÖZELLİKLERİN VERİ KÜMELERİNDEKİ HER SINIF İÇİN ORTALAMA DEĞER GRAFİKLERİ

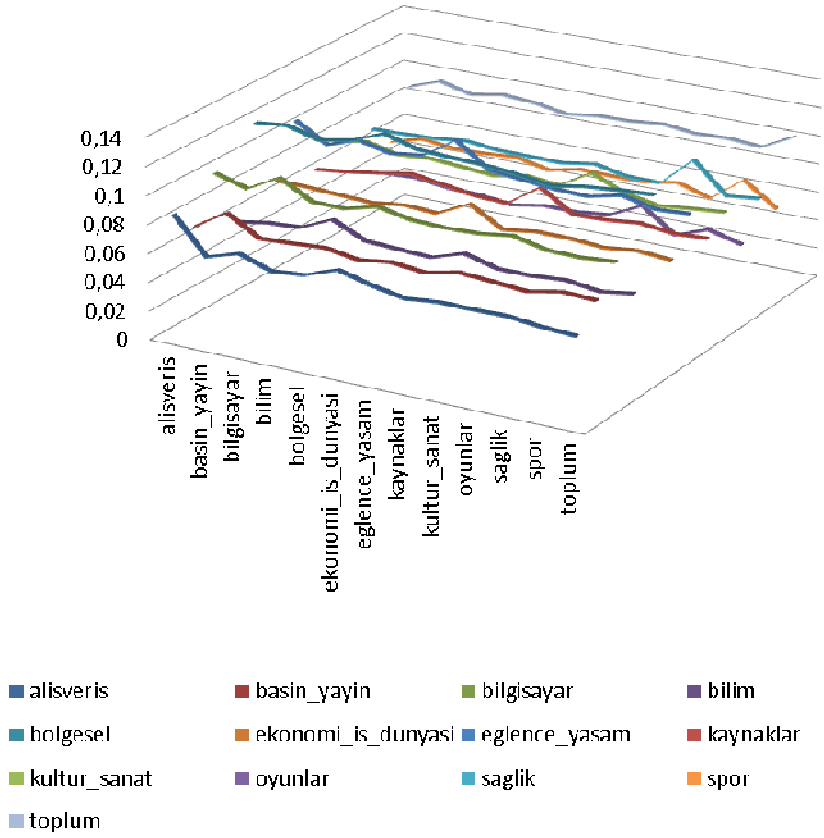
DMOZ, Reuters, ve 20-Newsgroups veri kümelerinde soyut özellik çıkarım yöntemi uygulandıktan sonra elde edilen soyut özelliklerin sınıflara göre dağılımlarını gösteren grafikler bu bölümde yer almaktadır. Grafiklerde her bir seri, bir sınıfa ait tüm örneklerin özelliklerinin ortalama değerlerini göstermektedir. Her bir örneğin değerlerini ayrı ayrı incelemek yerine, bir sınıftaki tüm örneklerin çıkarılan özelliklerinin ortalama değerleri gösterilerek, o sınıfın geneli için sınıflara ait olma olasılıkları görsel hale getirilmiştir. Grafikler incelendiğinde, her sınıf için o sınıfa ait soyut özelliğin en yüksek değeri taşıdığı görülebilir. ModApte-10 veri kümesi ayrık eğitim ve test kümelerinden oluştuğu için, grafiği çizilmemiştir.

DMOZ veri kümesinde soyut özelliklerin sınıflara göre dağılımları B-1'de verilmiştir. Yüksek değerler, benzeşen-karışabilecek, birbirine yakın sınıfları göstermektedir. Örneğin alışveriş sınıfı, ekonomi ve iş dünyası isimli sınıfla benzeştiği için o sınıfa ait olma olasılığı diğerlerine nazaran daha yüksek çıkmıştır.

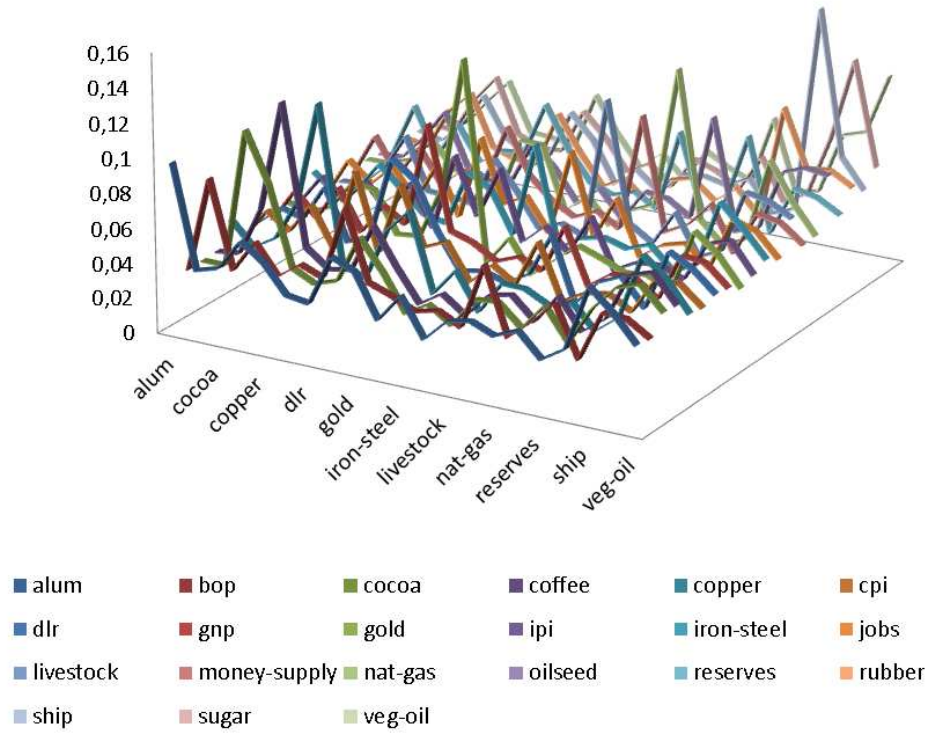
Reuters veri kümesinde soyut özelliklerin sınıflara göre dağılımları B-2'de verilmiştir. Burada sınıflar arası benzeşim DMOZ veri kümesi ile elde edilen sonuçlardan daha fazladır. Bu sebeple şekildeki aidiyet grafikleri daha karışık ve yakın görünmektedir.

20 Newsgroups veri kümesinde soyut özelliklerin sınıflara göre dağılımları B-3'te verilmiştir. Bu veri kümesinde sınıflar arası benzeşim DMOZ veya Reuters veri kümeleri ile elde edilen sonuçlardan daha azdır. Bu sebeple şekildeki olasılıklar sınıf için fazla, diğer sınıflar için daha az olarak görünmektedir.

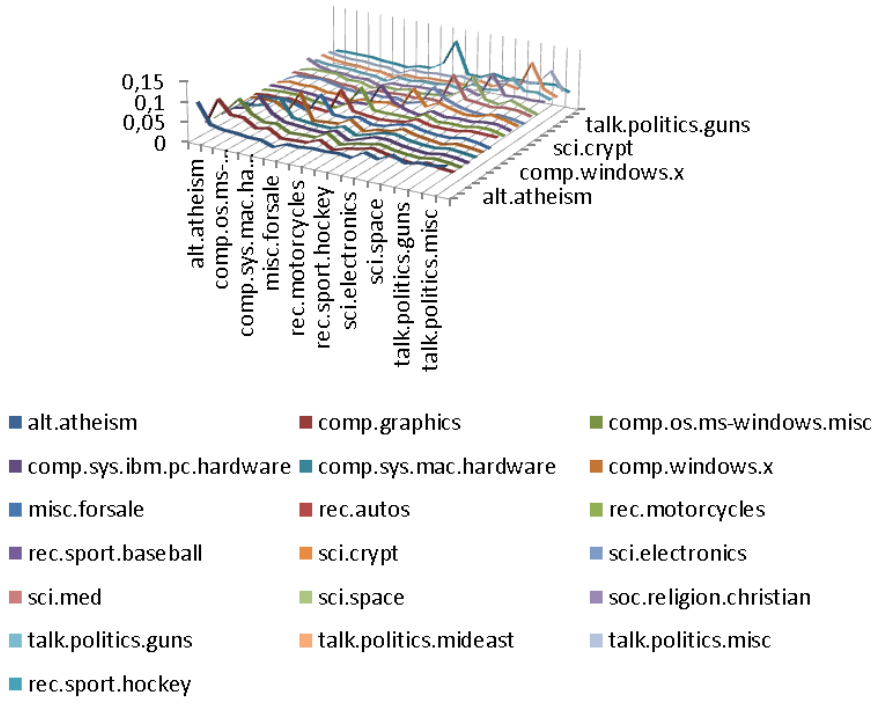
B-1 DMOZ Veri Kümesinde Soyut Özelliklerin Sınıflara Ait Olma Olasılıkları



B-2 Reuters Veri Kümesinde Soyut Özelliklerin Sınıflara Ait Olma Olasılıkları



B-3 20-Newsgroups Veri Kümesinde Soyut Özelliklerin Sınıflara Ait Olma Olasılıkları



EK-C

VERİ KÜMELERİNDE GERÇEKLEŞTİRİLEN TESTLERİN DUYARLILIK, ANMA, F-ÖLÇEĞİ VE ROC EĞRİSİ ALTINDAKİ ALAN SONUÇLARI

Bölüm 6'da yer alan sınıflandırma test sonuçları sadece F_1 ölçeği ile verilmişti. Bu bölümde ise, testlerde elde edilen tüm performas ölçekleri olan duyarlılık, anma, F_1ölçeği ve ROC eğrisi altındaki alan değerleri sunulmuştur.

C-1 DMOZ Veri Kümesi İle Elde Edilen Performans Sonuçları

(10 Kere Çapraz Doğrulama)		Özellik Çıkarımı Yok	Soyut Özellik Çıkarımı	Chi-Kare	Korelas. Katsayısı	Karşılıklı Bilgi	PCA	LSA	LDA
Naive Bayes	Duyarlılık	0,665	0,696	0,684	0,630	0,676	0,620	0,599	0,615
	Anma	0,557	0,668	0,543	0,454	0,545	0,617	0,574	0,261
	F-Ölçeği	0,591	0,669	0,584	0,504	0,583	0,617	0,577	0,321
	ROC Alanı	0,831	0,926	0,816	0,855	0,817	0,888	0,847	0,709
C4.5	Duyarlılık	0,480	0,720	0,529	0,543	0,530	0,450	0,504	0,505
	Anma	0,485	0,723	0,535	0,558	0,536	0,451	0,503	0,477
	F-Ölçeği	0,481	0,721	0,532	0,547	0,532	0,450	0,503	0,431
	ROC Alanı	0,700	0,852	0,723	0,774	0,722	0,682	0,715	0,703
RIPPER	Duyarlılık	0,576	0,739	0,580	0,570	0,575	0,512	0,557	0,542
	Anma	0,555	0,739	0,559	0,535	0,554	0,520	0,566	0,456
	F-Ölçeği	0,533	0,734	0,538	0,501	0,532	0,489	0,559	0,400
	ROC Alanı	0,759	0,884	0,763	0,742	0,761	0,743	0,781	0,671
10-NN	Duyarlılık	0,503	0,782	0,580	0,594	0,580	0,569	0,453	0,498
	Anma	0,318	0,783	0,347	0,593	0,347	0,402	0,289	0,471
	F-Ölçeği	0,245	0,781	0,226	0,583	0,226	0,316	0,163	0,426
	ROC Alanı	0,629	0,949	0,725	0,859	0,725	0,809	0,541	0,714
Rasgele Orman	Duyarlılık	0,503	0,764	0,497	0,563	0,495	0,419	0,280	0,419
	Anma	0,520	0,767	0,510	0,580	0,510	0,435	0,321	0,423
	F-Ölçeği	0,490	0,763	0,477	0,568	0,477	0,398	0,265	0,372
	ROC Alanı	0,826	0,942	0,822	0,851	0,822	0,747	0,633	0,679
SVM	Duyarlılık	0,753	0,759	0,730	0,618	0,730	0,649	0,650	0,533
	Anma	0,753	0,768	0,732	0,631	0,732	0,654	0,649	0,474
	F-Ölçeği	0,750	0,750	0,730	0,618	0,730	0,649	0,647	0,425
	ROC Alanı	0,851	0,857	0,840	0,775	0,840	0,794	0,789	0,647
LINEAR	Duyarlılık	0,707	0,747	0,661	0,621	0,661	0,646	0,719	0,540
	Anma	0,710	0,780	0,663	0,631	0,663	0,657	0,710	0,485
	F-Ölçeği	0,708	0,761	0,661	0,608	0,661	0,649	0,698	0,440
	ROC Alanı	0,828	0,864	0,801	0,771	0,801	0,796	0,815	0,652

C-2 Bağımsız DMOZ Test Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen Performans Sonuçları

		10 Kere Çapraz Doğrulama	11948 Eğitim 16660 Test	16660 Eğitim 11948 Test
Naive Bayes	Duyarlılık	0,713	0,655	0,665
	Anma	0,691	0,568	0,495
	F-Ölçeği	0,690	0,572	0,468
	ROC Alanı	0,934	0,903	0,900
C4.5	Duyarlılık	0,740	0,549	0,562
	Anma	0,743	0,505	0,487
	F-Ölçeği	0,741	0,503	0,491
	ROC Alanı	0,864	0,762	0,711
RIPPER	Duyarlılık	0,757	0,713	0,679
	Anma	0,758	0,501	0,440
	F-Ölçeği	0,753	0,528	0,478
	ROC Alanı	0,894	0,758	0,758
10-NN	Duyarlılık	0,789	0,675	0,641
	Anma	0,790	0,634	0,450
	F-Ölçeği	0,789	0,611	0,423
	ROC Alanı	0,949	0,882	0,803
Rasgele Orman	Duyarlılık	0,784	0,667	0,613
	Anma	0,788	0,614	0,558
	F-Ölçeği	0,783	0,607	0,540
	ROC Alanı	0,949	0,888	0,867
SVM	Duyarlılık	0,499	0,357	0,377
	Anma	0,365	0,316	0,340
	F-Ölçeği	0,248	0,158	0,201
	ROC Alanı	0,541	0,504	0,526
LINEAR	Duyarlılık	0,679	0,622	0,623
	Anma	0,708	0,697	0,638
	F-Ölçeği	0,663	0,647	0,578
	ROC Alanı	0,815	0,807	0,767

C-3 Reuters Veri Kümesi İle Elde Edilen Performans Sonuçları

(10 Kere Çapraz Doğrulama)		Özellik Çıkarımı Yok	Soyut Özellik Çıkarımı	Chi-Kare	Korelas. Katsayısı	Karşılıklı Bilgi	PCA	LSA	LDA
Naive Bayes	Duyarlılık	0,738	0,932	0,821	0,726	0,821	0,564	0,656	0,723
	Anma	0,708	0,931	0,808	0,649	0,808	0,481	0,519	0,580
	F-Ölçeği	0,715	0,931	0,810	0,638	0,810	0,487	0,517	0,584
	ROC Alanı	0,872	0,996	0,978	0,964	0,977	0,830	0,874	0,958
C4.5	Duyarlılık	0,835	0,914	0,830	0,807	0,830	0,578	0,680	0,820
	Anma	0,835	0,913	0,829	0,807	0,829	0,567	0,680	0,814
	F-Ölçeği	0,834	0,912	0,828	0,806	0,829	0,570	0,679	0,813
	ROC Alanı	0,923	0,961	0,924	0,924	0,924	0,806	0,845	0,939
RIPPER	Duyarlılık	0,824	0,921	0,838	0,805	0,835	0,528	0,650	0,806
	Anma	0,808	0,918	0,822	0,776	0,818	0,483	0,638	0,769
	F-Ölçeği	0,810	0,919	0,824	0,781	0,821	0,492	0,640	0,773
	ROC Alanı	0,938	0,970	0,940	0,935	0,945	0,786	0,841	0,927
10-NN	Duyarlılık	0,774	0,969	0,870	0,861	0,870	0,779	0,350	0,847
	Anma	0,506	0,969	0,762	0,844	0,762	0,687	0,088	0,837
	F-Ölçeği	0,481	0,968	0,789	0,847	0,789	0,692	0,046	0,836
	ROC Alanı	0,928	0,999	0,960	0,969	0,960	0,945	0,722	0,968
Rasgele Orman	Duyarlılık	0,649	0,931	0,824	0,846	0,842	0,684	0,370	0,841
	Anma	0,642	0,929	0,821	0,845	0,837	0,678	0,366	0,833
	F-Ölçeği	0,635	0,929	0,819	0,844	0,835	0,672	0,357	0,832
	ROC Alanı	0,908	0,992	0,974	0,967	0,975	0,910	0,769	0,964
SVM	Duyarlılık	0,911	0,969	0,913	0,871	0,913	0,819	0,761	0,857
	Anma	0,900	0,969	0,909	0,855	0,909	0,781	0,610	0,839
	F-Ölçeği	0,901	0,969	0,910	0,856	0,910	0,783	0,598	0,837
	ROC Alanı	0,946	0,983	0,951	0,921	0,951	0,880	0,784	0,912
LINEAR	Duyarlılık	0,934	0,838	0,893	0,869	0,893	0,867	0,792	0,858
	Anma	0,932	0,852	0,892	0,868	0,892	0,866	0,739	0,847
	F-Ölçeği	0,932	0,820	0,892	0,868	0,892	0,865	0,743	0,845
	ROC Alanı	0,964	0,920	0,943	0,929	0,943	0,929	0,857	0,917

C-4 20-Newsgroups Veri Kümesi İle Elde Edilen Performans Sonuçları

(10 Kere Çapraz Doğrulama)		Özellik Çıkarımı Yok	Soyut Özellik Çıkarımı	Chi-Kare	Korelas. Katsayısı	Karşılıklı Bilgi	PCA	LSA	LDA
Naive Bayes	Duyarlılık	0,559	0,899	0,612	0,481	0,612	0,514	0,577	0,419
	Anma	0,521	0,898	0,605	0,446	0,605	0,470	0,504	0,298
	F-Ölçeği	0,527	0,897	0,597	0,436	0,597	0,472	0,516	0,289
	ROC Alanı	0,788	0,992	0,939	0,884	0,938	0,871	0,859	0,784
C4.5	Duyarlılık	0,501	0,869	0,506	0,446	0,506	0,438	0,453	0,407
	Anma	0,484	0,869	0,498	0,438	0,497	0,432	0,444	0,353
	F-Ölçeği	0,490	0,869	0,500	0,439	0,499	0,434	0,447	0,363
	ROC Alanı	0,762	0,944	0,788	0,751	0,787	0,718	0,731	0,726
RIPPER	Duyarlılık	0,504	0,878	0,515	0,471	0,518	0,405	0,413	0,511
	Anma	0,417	0,877	0,451	0,391	0,437	0,385	0,407	0,303
	F-Ölçeği	0,433	0,877	0,467	0,391	0,455	0,391	0,408	0,343
	ROC Alanı	0,788	0,968	0,818	0,797	0,815	0,728	0,749	0,731
10-NN	Duyarlılık	0,442	0,940	0,553	0,488	0,553	0,330	0,525	0,421
	Anma	0,082	0,939	0,449	0,431	0,449	0,066	0,065	0,356
	F-Ölçeği	0,056	0,939	0,463	0,444	0,463	0,037	0,036	0,372
	ROC Alanı	0,572	0,995	0,824	0,802	0,824	0,580	0,744	0,745
Rasgele Orman	Duyarlılık	0,535	0,913	0,473	0,392	0,484	0,183	0,227	0,344
	Anma	0,459	0,912	0,466	0,378	0,482	0,173	0,209	0,306
	F-Ölçeği	0,472	0,912	0,465	0,382	0,479	0,171	0,211	0,313
	ROC Alanı	0,817	0,988	0,848	0,800	0,850	0,665	0,685	0,726
SVM	Duyarlılık	0,731	0,932	0,664	0,581	0,664	0,714	0,692	0,566
	Anma	0,695	0,930	0,614	0,496	0,614	0,696	0,644	0,377
	F-Ölçeği	0,705	0,930	0,631	0,521	0,631	0,701	0,659	0,409
	ROC Alanı	0,839	0,963	0,797	0,735	0,797	0,840	0,813	0,672
LINEAR	Duyarlılık	0,772	0,950	0,600	0,545	0,600	0,703	0,677	0,557
	Anma	0,749	0,949	0,597	0,523	0,597	0,703	0,671	0,394
	F-Ölçeği	0,754	0,948	0,597	0,525	0,597	0,702	0,673	0,426
	ROC Alanı	0,868	0,973	0,788	0,749	0,788	0,843	0,827	0,681

C-5 ModApte-10 Veri Kümesi İle Elde Edilen Performans Sonuçları

(Standart Eğitim ve Test Kümeleri)		Özellik Çıkarımı Yok	Soyut Özellik Çıkarımı	Chi-Kare	Korelas. Katsayısı	Karşılıklı Bilgi	PCA	LSA	LDA
Naive Bayes	Duyarlılık	0,860	0,918	0,891	0,867	0,887	0,671	0,541	0,797
	Anma	0,407	0,911	0,750	0,755	0,746	0,628	0,547	0,743
	F-Ölçeği	0,540	0,911	0,803	0,790	0,799	0,612	0,508	0,732
	ROC Alanı	0,743	0,992	0,908	0,973	0,903	0,818	0,772	0,931
C4.5	Duyarlılık	0,867	0,949	0,881	0,850	0,878	0,839	0,840	0,819
	Anma	0,870	0,949	0,882	0,851	0,879	0,841	0,840	0,808
	F-Ölçeği	0,868	0,948	0,881	0,849	0,878	0,840	0,840	0,805
	ROC Alanı	0,935	0,973	0,946	0,955	0,945	0,896	0,911	0,923
RIPPER	Duyarlılık	0,874	0,950	0,867	0,841	0,873	0,868	0,840	0,779
	Anma	0,875	0,949	0,868	0,841	0,876	0,864	0,832	0,777
	F-Ölçeği	0,872	0,949	0,863	0,836	0,872	0,865	0,836	0,767
	ROC Alanı	0,937	0,988	0,931	0,918	0,938	0,931	0,914	0,872
10-NN	Duyarlılık	0,741	0,962	0,728	0,875	0,728	0,679	0,824	0,821
	Anma	0,468	0,962	0,483	0,876	0,483	0,538	0,522	0,810
	F-Ölçeği	0,359	0,961	0,369	0,875	0,369	0,460	0,516	0,807
	ROC Alanı	0,861	0,997	0,928	0,983	0,928	0,894	0,903	0,932
Rasgele Orman	Duyarlılık	0,828	0,918	0,868	0,887	0,857	0,735	0,566	0,785
	Anma	0,837	0,911	0,872	0,890	0,861	0,752	0,602	0,780
	F-Ölçeği	0,825	0,911	0,863	0,888	0,852	0,719	0,550	0,775
	ROC Alanı	0,973	0,992	0,985	0,983	0,980	0,928	0,833	0,904
SVM	Duyarlılık	0,927	0,878	0,914	0,890	0,914	0,881	0,858	0,828
	Anma	0,929	0,905	0,917	0,893	0,917	0,886	0,864	0,810
	F-Ölçeği	0,927	0,882	0,914	0,891	0,914	0,883	0,857	0,808
	ROC Alanı	0,958	0,946	0,946	0,934	0,946	0,927	0,910	0,879
LINEAR	Duyarlılık	0,926	0,953	0,907	0,893	0,907	0,920	0,824	0,830
	Anma	0,925	0,945	0,911	0,895	0,911	0,921	0,810	0,808
	F-Ölçeği	0,925	0,937	0,908	0,893	0,908	0,920	0,790	0,808
	ROC Alanı	0,956	0,968	0,944	0,934	0,944	0,953	0,847	0,879

C-6 21 özellikli Reuters Veri Kümesi İle Elde Edilen Performans Sonuçları

(10 Kere Çapraz Doğrulama)		Soyut Özellik Çıkarımı	Chi-Kare	Korelas. Katsayısı.	Karşılıklı Bilgi	PCA	LSA
Naive Bayes	Duyarlılık	0,938	0,650	0,669	0,702	0,798	0,843
	Anma	0,931	0,603	0,572	0,587	0,771	0,816
	F-Ölçeği	0,930	0,595	0,559	0,590	0,769	0,817
	ROC Alanı	0,997	0,945	0,956	0,945	0,975	0,985
C4.5	Duyarlılık	0,923	0,743	0,774	0,722	0,695	0,765
	Anma	0,913	0,743	0,767	0,726	0,670	0,743
	F-Ölçeği	0,912	0,727	0,759	0,710	0,663	0,739
	ROC Alanı	0,961	0,924	0,915	0,905	0,849	0,878
RIPPER	Duyarlılık	0,926	0,713	0,747	0,654	0,716	0,755
	Anma	0,918	0,717	0,720	0,654	0,643	0,712
	F-Ölçeği	0,917	0,692	0,707	0,621	0,644	0,712
	ROC Alanı	0,971	0,919	0,922	0,898	0,884	0,907
10-NN	Duyarlılık	0,972	0,761	0,808	0,767	0,838	0,890
	Anma	0,969	0,766	0,795	0,756	0,819	0,874
	F-Ölçeği	0,968	0,748	0,789	0,745	0,816	0,874
	ROC Alanı	0,999	0,951	0,964	0,943	0,976	0,984
Rasgele Orman	Duyarlılık	0,938	0,736	0,800	0,744	0,802	0,852
	Anma	0,929	0,744	0,789	0,742	0,786	0,832
	F-Ölçeği	0,927	0,727	0,783	0,731	0,781	0,829
	ROC Alanı	0,992	0,930	0,949	0,935	0,976	0,984
SVM	Duyarlılık	0,972	0,753	0,797	0,709	0,825	0,879
	Anma	0,969	0,759	0,795	0,727	0,801	0,858
	F-Ölçeği	0,968	0,737	0,781	0,697	0,793	0,854
	ROC Alanı	0,983	0,867	0,889	0,850	0,892	0,924
LINEAR	Duyarlılık	0,823	0,744	0,782	0,713	0,844	0,759
	Anma	0,852	0,765	0,791	0,745	0,831	0,789
	F-Ölçeği	0,818	0,740	0,774	0,716	0,826	0,754
	ROC Alanı	0,920	0,872	0,888	0,863	0,909	0,887

C-7 20 Özellikli 20-Newsgroups Veri Kümesi İle Elde Edilen Performans Sonuçları

(10 Kere Çapraz Doğrulama)		Soyut Özellik Çıkarımı	Chi-Kare	Korelas. Katsayısı.	Karşılıklı Bilgi	PCA	LSA
Naive Bayes	Duyarlılık	0,910	0,350	0,374	0,315	0,575	0,598
	Anma	0,898	0,301	0,295	0,265	0,556	0,587
	F-Ölçeği	0,897	0,281	0,280	0,246	0,540	0,577
	ROC Alanı	0,992	0,765	0,814	0,806	0,935	0,945
C4.5	Duyarlılık	0,881	0,370	0,365	0,296	0,446	0,452
	Anma	0,869	0,304	0,321	0,291	0,438	0,439
	F-Ölçeği	0,868	0,297	0,326	0,280	0,432	0,436
	ROC Alanı	0,944	0,735	0,722	0,684	0,740	0,741
RIPPER	Duyarlılık	0,898	0,334	0,447	0,329	0,547	0,540
	Anma	0,877	0,251	0,289	0,229	0,421	0,412
	F-Ölçeği	0,879	0,245	0,303	0,225	0,439	0,434
	ROC Alanı	0,967	0,691	0,728	0,689	0,818	0,821
10-NN	Duyarlılık	0,944	0,414	0,386	0,330	0,594	0,586
	Anma	0,939	0,318	0,333	0,307	0,584	0,607
	F-Ölçeği	0,939	0,317	0,341	0,300	0,564	0,571
	ROC Alanı	0,995	0,747	0,753	0,732	0,895	0,912
Rasgele Orman	Duyarlılık	0,919	0,380	0,339	0,296	0,525	0,562
	Anma	0,912	0,288	0,291	0,282	0,519	0,551
	F-Ölçeği	0,912	0,291	0,300	0,275	0,510	0,546
	ROC Alanı	0,988	0,726	0,731	0,712	0,882	0,889
SVM	Duyarlılık	0,936	0,379	0,503	0,422	0,621	0,622
	Anma	0,930	0,303	0,350	0,307	0,584	0,607
	F-Ölçeği	0,930	0,296	0,371	0,316	0,580	0,598
	ROC Alanı	0,963	0,633	0,658	0,635	0,781	0,793
LINEAR	Duyarlılık	0,953	0,340	0,428	0,349	0,597	0,562
	Anma	0,948	0,323	0,364	0,331	0,604	0,566
	F-Ölçeği	0,947	0,299	0,359	0,311	0,582	0,525
	ROC Alanı	0,973	0,644	0,665	0,648	0,792	0,772

C-8 10 Özellikli ModApte-10 Veri Kümesi İle Elde Edilen Performans Sonuçları

(Standart Eğitim ve Test Kümeleri)		Soyut Özellik Çıkarımı	Chi-Kare	Korelas. Katsayısı.	Karşılıklı Bilgi	PCA	LSA
Naive Bayes	Duyarlılık	0,918	0,755	0,859	0,514	0,896	0,881
	Anma	0,911	0,516	0,657	0,435	0,892	0,869
	F-Ölçeği	0,911	0,504	0,706	0,460	0,893	0,869
	ROC Alanı	0,992	0,941	0,952	0,909	0,988	0,991
C4.5	Duyarlılık	0,949	0,788	0,798	0,711	0,886	0,886
	Anma	0,949	0,809	0,806	0,739	0,881	0,883
	F-Ölçeği	0,948	0,795	0,796	0,710	0,879	0,883
	ROC Alanı	0,973	0,932	0,948	0,896	0,951	0,951
RIPPER	Duyarlılık	0,950	0,742	0,747	0,641	0,872	0,889
	Anma	0,949	0,769	0,765	0,675	0,869	0,890
	F-Ölçeği	0,949	0,748	0,740	0,620	0,866	0,887
	ROC Alanı	0,988	0,861	0,853	0,778	0,946	0,961
10-NN	Duyarlılık	0,962	0,796	0,799	0,722	0,916	0,915
	Anma	0,962	0,806	0,809	0,741	0,915	0,914
	F-Ölçeği	0,961	0,797	0,800	0,714	0,914	0,913
	ROC Alanı	0,997	0,939	0,956	0,911	0,991	0,986
Rasgele Orman	Duyarlılık	0,961	0,733	0,773	0,677	0,904	0,905
	Anma	0,960	0,742	0,782	0,702	0,900	0,907
	F-Ölçeği	0,960	0,736	0,776	0,686	0,899	0,904
	ROC Alanı	0,996	0,916	0,940	0,891	0,989	0,987
SVM	Duyarlılık	0,961	0,804	0,796	0,655	0,907	0,892
	Anma	0,959	0,814	0,808	0,734	0,891	0,891
	F-Ölçeği	0,957	0,802	0,793	0,680	0,870	0,875
	ROC Alanı	0,977	0,883	0,876	0,821	0,933	0,937
LINEAR	Duyarlılık	0,953	0,790	0,803	0,638	0,895	0,765
	Anma	0,945	0,808	0,813	0,732	0,894	0,792
	F-Ölçeği	0,937	0,785	0,804	0,668	0,880	0,736
	ROC Alanı	0,968	0,875	0,884	0,818	0,933	0,836

EK-D

VERİ KÜMELERİNDE TESTLERİN EN İYİ VE EN KÖTÜ KARMAŞIKLIK MATRİSLERİ

Bölüm 6'da yer alan sonuçlara göre yapılan sınıflandırma testlerinde elde edilen en iyi ve en kötü karmaşıklık matrisleri bu bölümde verilmiştir.

D-1 DMOZ Veri Kümesinde Özellik Çıkarımı Olmadan Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
68	0	254	0	1	222	0	0	7	0	0	1	a = alisveris
0	6	276	0	1	180	0	0	14	0	0	1	b = basin_yayin
4	1	1182	0	7	886	0	2	42	0	0	1	c = bilgisayar
0	0	179	11	0	127	13	5	11	0	0	0	d = bilim
0	0	302	0	19	217	0	1	16	0	0	1	e = eglence_yasam
7	2	1209	0	11	2375	0	2	72	0	0	5	f = ekonomi_is_dunyasi
0	0	343	6	1	315	62	0	15	1	0	0	g = kaynaklar
0	0	492	0	2	362	0	20	36	0	0	2	h = kultur_sanat
0	0	113	0	2	58	0	0	26	0	0	0	i = oyunlar
0	0	494	0	1	318	2	0	22	5	0	1	j = saglik
0	0	230	0	0	154	0	1	11	0	1	1	k = spor
0	1	639	0	1	423	3	2	21	1	0	19	l = toplum

D-2 DMOZ Veri Kümesinde Özellik Çıkarımı Olmadan Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
383	2	33	3	7	95	3	17	2	2	1	5	a = alisveris
2	271	27	4	6	54	1	47	0	5	13	48	b = basin_yayin
28	14	1630	5	14	264	26	58	10	6	2	68	c = bilgisayar
0	3	10	133	3	57	56	18	0	14	0	52	d = bilim
6	10	19	2	381	61	2	31	0	5	7	32	e = eglence_yasam
58	15	199	11	25	3186	32	49	0	41	4	63	f = ekonomi_is_dunyasi
1	3	42	33	4	56	499	19	2	29	3	52	g = kaynaklar
4	18	40	5	12	73	13	678	1	1	1	68	h = kultur_sanat
0	2	20	0	3	12	2	10	124	1	15	10	i = oyunlar
2	3	13	6	9	57	27	14	0	678	2	32	j = saglik
2	5	9	2	17	23	5	8	0	3	306	18	k = spor
3	25	86	19	14	103	25	78	0	25	9	723	l = toplum

D-3 DMOZ Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j    k    l   <-- classified as
 147    0    6    0    4    0   11    1   21    8    0    1 |    a = oyunlar
   1  155   17   11    1  102   18    0    0    4   37    0 |    b = bilim
  13   10 2241   97   66   62  919   17    6   61   78  113 |    c = ekonomi_is_dunyasi
   1    0   12  709   13   47   15    1    4    4   32    5 |    d = saglik
   3    0   24    3  416    0   33    1   18   43   10    5 |    e = eglence_yasam
   3    8   25   20    1  542   70    1    3    9   61    0 |    f = kaynaklar
  72    1  128    1   26   29 1637   20    3  128   36   44 |    g = bilgisayar
   9    3   12    3    8    1   29  290   22   58   37    6 |    h = basin_yayin
  11    0    6    2   22    8    7    7  325    3    7    0 |    i = spor
  31    3   12    4   14   10   54   13    5  686   57   25 |    j = kultur_sanat
  16   28   27   23   18   73   75   36   14  165  633    2 |    k = toplum
   3    2  116   15   32    5  106    7    0   41   24  202 |    l = alisveris
```

D-4 DMOZ Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En İyi Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j    k    l   <-- classified as
 154    0    3    0    4    0   20    0    9    5    4    0 |    a = oyunlar
   1  211   30    9    2   47   11    2    0    8   25    0 |    b = bilim
   1    3 3071   51   34   41  250   14   10   48   62   98 |    c = ekonomi_is_dunyasi
   0    1   44  716   10   26   11    1    3    7   22    2 |    d = saglik
   0    0   54    1  434    2   16    2    9   19   12    7 |    e = eglence_yasam
   2   11   39   23    0  581   45    3    3   10   26    0 |    f = kaynaklar
   9    3  205    2   17   25 1747   27    1   37   38   14 |    g = bilgisayar
   1    3   31    3    6    1   30  338    2   24   31    8 |    h = basin_yayin
   1    0    8    2   11    5   11    5  333    3   16    3 |    i = spor
   1    7   45    5   10    6   32   21    2  740   32   13 |    j = kultur_sanat
   3   11   80   25   15   46   70   32   12   54  760    2 |    k = toplum
   1    1  156    9    9   10   54    6    1   33    7  266 |    l = alisveris
```

D-5 DMOZ Test Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En Kötü Karmaşıklık Matrisi (SVM Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j    k    l   <-- classified as
   0    0  247    0    0    0  123    0    0    0    0    0 |    a = oyunlar
   0    0  327    0    0   11    1    0    0    0    9    0 |    b = bilim
   0    0 5012    3    0    0    0    0    0    0    1    0 |    c = ekonomi_is_dunyasi
   0    0  824  113    0    7    0    0    0    0    2    0 |    d = saglik
   0    0  660    0    0    0    1    0    0    0    2    0 |    e = eglence_yasam
   0    0 1264    3    0  134    0    0    0    0    5    0 |    f = kaynaklar
   0    0 2431    0    0    0  297    0    0    0    6    0 |    g = bilgisayar
   0    0  362    0    0    0    0    0    0    0   25    0 |    h = basin_yayin
   0    0  504    0    0    1    1    0    0    0    3    0 |    i = spor
   0    0  907    0    0    0    0    0    0    1   48    0 |    j = kultur_sanat
   0    0 1597    1    0    8    6    0    0    0  322    0 |    k = toplum
   0    0  852    0    0    0    0    0    0    0    2    0 |    l = alisveris
```

D-6 DMOZ Test Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En İyi Karmaşıklık Matrisi (10 En Yakın Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
324	2	4	0	0	0	23	2	8	2	5	0	a = oyunlar
1	187	38	13	5	45	15	0	1	3	39	1	b = bilim
3	6	4176	53	51	43	311	20	16	56	120	161	c = ekonomi_is_dunyasi
0	1	32	826	5	29	7	0	7	3	32	4	d = saglik
3	3	53	1	538	1	16	1	1	8	29	9	e = eglence_yasam
1	16	56	25	8	1171	39	0	2	16	70	2	f = kaynaklar
22	2	244	5	12	29	2198	23	5	43	105	46	g = bilgisayar
0	1	29	0	4	1	20	265	4	18	44	1	h = basin_yayin
1	1	10	3	20	6	9	4	426	1	26	2	i = spor
3	5	53	1	6	14	29	12	6	741	68	18	j = kultur_sanat
11	14	93	31	28	96	125	28	17	55	1428	8	k = toplum
0	2	241	8	17	5	73	4	3	17	20	464	l = alisveris

D-7 DMOZ Veri Kümesinde 11948 Eğitim 16660 Test Örneği İle Soyut Özellik Çıkarımıyla Elde Edilen En Kötü Karmaşıklık Matrisi (SVM Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
0	0	358	0	0	0	12	0	0	0	0	0	a = oyunlar
0	0	348	0	0	0	0	0	0	0	0	0	b = bilim
0	0	5016	0	0	0	0	0	0	0	0	0	c = ekonomi_is_dunyasi
0	0	917	29	0	0	0	0	0	0	0	0	d = saglik
0	0	662	0	0	0	1	0	0	0	0	0	e = eglence_yasam
0	0	1405	1	0	0	0	0	0	0	0	0	f = kaynaklar
0	0	2711	0	0	0	23	0	0	0	0	0	g = bilgisayar
0	0	387	0	0	0	0	0	0	0	0	0	h = basin_yayin
0	0	509	0	0	0	0	0	0	0	0	0	i = spor
0	0	955	0	0	0	1	0	0	0	0	0	j = kultur_sanat
0	0	1899	0	0	0	10	0	0	0	25	0	k = toplum
0	0	854	0	0	0	0	0	0	0	0	0	l = alisveris

D-8 DMOZ Veri Kümesinde 11948 Eğitim 16660 Test Örneği İle Soyut Özellik Çıkarımıyla Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
0	0	8	1	0	1	297	0	58	1	4	0	a = oyunlar
0	0	91	13	0	123	46	0	0	2	73	0	b = bilim
0	0	4597	37	2	48	220	0	11	22	79	0	c = ekonomi_is_dunyasi
0	0	41	825	0	57	7	0	4	2	10	0	d = saglik
0	0	244	9	278	5	77	0	19	4	27	0	e = eglence_yasam
0	0	108	15	0	1166	85	0	2	5	25	0	f = kaynaklar
0	0	382	7	2	24	2247	0	4	16	52	0	g = bilgisayar
0	0	76	1	1	8	55	0	24	28	194	0	h = basin_yayin
0	0	35	3	2	23	16	0	421	1	8	0	i = spor
0	0	101	3	1	48	118	0	9	552	124	0	j = kultur_sanat
0	0	240	52	7	193	245	0	21	23	1153	0	k = toplum
0	0	676	4	7	9	113	0	0	10	35	0	l = alisveris

D-9 DMOZ Veri Kümesinde 16660 Eğitim 11948 Test Örneği İle Soyut Özellik Çıkarımıyla Elde Edilen En Kötü Karmaşıklık Matrisi (SVM Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
0	0	178	0	0	0	21	0	0	0	0	0	a = oyunlar
0	0	330	0	0	6	0	0	0	0	10	0	b = bilim
0	0	3677	1	0	0	0	0	0	0	5	0	c = ekonomi_is_dunyasi
0	0	803	36	0	1	0	0	0	0	3	0	d = saglik
0	0	553	0	0	0	0	0	0	0	3	0	e = eglence_yasam
0	0	715	1	0	18	1	0	0	0	8	0	f = kaynaklar
0	0	2058	0	0	0	58	0	0	0	9	0	g = bilgisayar
0	0	436	0	0	0	0	0	0	0	42	0	h = basin_yayin
0	0	394	0	0	0	0	0	0	0	4	0	i = spor
0	0	801	0	0	0	8	0	0	0	105	0	j = kultur_sanat
0	0	834	0	0	0	4	0	0	0	272	0	k = toplum
0	0	550	0	0	0	1	0	0	0	2	0	l = alisveris

D-10 DMOZ Veri Kümesinde 16660 Eğitim 11948 Test Örneği İle Soyut Özellik Çıkarımıyla Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	<-- classified as
0	0	7	0	0	0	170	0	5	10	7	0	a = oyunlar
0	0	122	13	0	54	74	0	0	12	71	0	b = bilim
0	0	3419	15	0	2	178	0	0	30	39	0	c = ekonomi_is_dunyasi
0	0	95	658	0	18	38	0	0	12	22	0	d = saglik
0	0	278	7	54	0	105	0	1	46	65	0	e = eglence_yasam
0	0	165	18	0	319	175	0	0	21	45	0	f = kaynaklar
0	0	408	0	0	1	1660	0	0	28	28	0	g = bilgisayar
0	0	100	4	0	0	64	7	11	100	192	0	h = basin_yayin
0	0	79	3	2	3	56	0	166	47	42	0	i = spor
0	0	123	2	0	0	95	0	0	638	56	0	j = kultur_sanat
0	0	183	19	1	12	151	0	0	38	706	0	k = toplum
0	0	418	6	0	2	77	0	0	26	24	0	l = alisveris

D-11 Reuters Veri Kümesinde Özellik Çıkarımı Olmadan Elde Edilen En Kötü Karmaşıklık Matrisi (Rasgele Orman Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
20	0	0	3	4	1	0	0	2	1	4	1	0	0	0	1	1	2	4	4	0	a = alum
1	34	0	1	0	0	0	5	1	0	0	0	0	2	0	0	1	0	1	0	0	b = bop
0	1	31	12	1	1	0	0	1	0	1	0	1	0	0	1	0	1	3	4	0	c = cocoa
0	1	6	99	0	1	0	4	2	0	1	0	0	0	1	2	0	1	4	2	0	d = coffee
2	1	2	2	33	0	0	1	7	0	2	0	0	0	0	0	0	0	6	0	1	e = copper
0	5	0	1	0	56	0	4	0	2	0	1	0	1	2	0	0	0	0	2	1	f = cpi
1	0	2	2	0	0	18	8	0	0	0	0	0	0	0	1	1	0	1	0	0	g = dlr
2	7	0	2	1	1	0	83	5	2	0	0	0	2	0	1	3	0	4	1	1	h = gnp
1	1	0	5	5	2	0	2	88	0	0	0	0	0	1	0	0	1	4	0	1	i = gold
1	1	1	1	0	8	0	7	1	20	0	1	1	1	0	0	0	0	1	1	0	j = ipi
4	1	2	4	0	1	1	1	6	1	16	0	0	2	2	1	0	0	8	1	0	k = iron-steel
0	2	1	0	1	3	1	2	1	1	0	31	0	1	1	0	0	0	2	0	0	l = jobs
0	2	2	4	0	0	0	0	3	0	3	1	21	0	1	1	1	1	9	4	2	m = livestock
0	2	0	2	0	1	0	9	0	0	0	0	0	91	0	0	4	0	1	0	0	n = money-supply
3	2	0	0	1	2	0	0	2	1	0	0	0	1	32	0	0	0	3	1	0	o = nat-gas
0	1	1	3	2	0	0	2	1	0	1	1	0	1	0	44	0	0	5	4	12	p = oilseed
0	2	0	2	0	2	0	4	3	0	0	1	0	6	0	0	27	0	3	0	0	q = reserves
2	0	3	5	4	0	0	3	2	0	2	0	1	3	0	0	0	11	1	2	1	r = rubber
2	2	4	11	2	1	0	2	4	0	1	1	1	4	0	3	2	1	151	1	1	s = ship
3	1	4	13	2	1	0	1	4	0	2	0	2	1	0	4	1	0	13	90	2	t = sugar
0	2	2	5	1	2	1	2	2	1	2	1	5	1	0	12	0	1	2	5	46	u = veg-oil

D-12 Reuters Veri Kümesinde Özellik Çıkarımı Olmadan Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
46	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	a = alum
0	39	0	0	0	0	0	6	0	0	0	0	0	0	0	0	1	0	0	0	0	b = bop
0	0	55	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	c = cocoa
0	0	2	120	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	d = coffee
0	0	0	0	56	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	e = copper
0	0	0	0	0	69	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	f = cpi
0	1	0	0	0	0	32	1	0	0	0	0	0	0	0	0	0	0	0	0	0	g = dlr
0	1	0	1	0	0	0	109	0	1	0	0	0	2	0	0	0	0	1	0	0	h = gnp
0	0	0	0	0	0	0	0	110	0	0	0	0	0	0	0	0	0	1	0	0	i = gold
0	0	0	0	0	0	0	3	0	42	0	0	0	0	0	0	0	0	0	0	0	j = ipi
2	0	0	0	0	0	0	0	1	0	46	0	0	0	0	1	0	0	1	0	0	k = iron-steel
0	0	0	0	0	0	0	3	0	1	0	43	0	0	0	0	0	0	0	0	0	l = jobs
0	0	0	0	0	0	0	0	0	1	0	0	52	0	0	0	0	0	1	0	1	m = livestock
0	1	0	0	0	0	0	3	0	0	0	0	0	105	0	0	1	0	0	0	0	n = money-supply
0	0	0	0	0	0	0	0	0	0	0	0	0	0	46	0	0	0	2	0	0	o = nat-gas
0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	57	0	0	2	1	16	p = oilseed
0	0	0	0	0	1	0	3	0	0	0	0	0	3	0	0	43	0	0	0	0	q = reserves
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	r = rubber
1	0	0	0	0	0	0	0	2	0	2	0	1	0	2	1	0	0	183	1	1	s = ship
0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	2	139	0	t = sugar
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	9	0	0	1	1	81	u = veg-oil

D-13 Reuters Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En Kötü Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	a = alum
0	0	0	0	0	0	0	14	0	0	0	0	0	32	0	0	0	0	0	0	0	b = bop
0	0	53	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	c = cocoa
0	0	0	123	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d = coffee
0	0	0	1	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	e = copper
0	0	0	0	0	65	0	7	0	0	0	0	0	1	0	0	0	0	2	0	0	f = cpi
0	0	0	0	0	0	11	12	1	0	0	0	0	9	0	0	0	0	1	0	0	g = dlr
0	0	0	0	0	0	0	115	0	0	0	0	0	0	0	0	0	0	0	0	0	h = gnp
0	0	0	0	0	0	0	0	111	0	0	0	0	0	0	0	0	0	0	0	0	i = gold
0	0	0	0	0	24	0	14	0	5	0	0	0	2	0	0	0	0	0	0	0	j = ipi
0	0	0	0	0	0	0	0	0	0	48	0	0	0	0	0	0	0	3	0	0	k = iron-steel
0	0	0	0	0	0	0	10	0	0	0	36	0	0	0	0	0	0	0	0	1	l = jobs
0	0	0	0	0	0	0	0	0	0	0	0	53	0	0	0	0	0	0	1	1	m = livestock
0	0	0	0	0	0	0	0	0	0	0	0	0	110	0	0	0	0	0	0	0	n = money-supply
0	0	0	0	0	0	0	0	0	0	0	0	0	0	47	0	0	0	1	0	0	o = nat-gas
0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	44	0	0	4	10	18	p = oilseed
0	0	0	0	0	0	0	1	0	0	0	0	0	49	0	0	0	0	0	0	0	q = reserves
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38	0	2	0	r = rubber
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	192	1	1	s = ship
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	143	0	t = sugar
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	2	88	u = veg-oil

D-14 Reuters Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En İyi Karmaşıklık Matrisleri (10 En Yakın Komşu ve SVM Algoritmaları İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	a = alum
0	45	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	b = bop
0	0	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	c = cocoa
0	0	0	121	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	d = coffee
0	0	0	1	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	e = copper
0	0	0	0	0	73	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	f = cpi
0	0	0	0	0	0	32	1	0	0	0	0	0	0	0	0	1	0	0	0	0	g = dlr
0	0	0	0	0	0	0	113	0	2	0	0	0	0	0	0	0	0	0	0	0	h = gnp
0	0	0	0	0	0	0	0	110	0	0	0	0	0	0	0	0	0	0	1	0	i = gold
0	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0	0	0	0	0	j = ipi
0	0	0	0	0	0	0	0	0	0	51	0	0	0	0	0	0	0	0	0	0	k = iron-steel
0	0	0	0	0	0	0	0	0	0	0	47	0	0	0	0	0	0	0	0	0	l = jobs
0	0	0	0	0	0	0	0	0	0	0	0	53	0	0	0	0	0	0	1	1	m = livestock
0	1	0	0	0	0	0	0	0	0	0	0	0	109	0	0	0	0	0	0	0	n = money-supply
0	0	0	0	0	0	0	0	0	0	0	0	0	0	47	0	0	0	1	0	0	o = nat-gas
0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	63	0	0	0	2	11	p = oilseed
0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	46	0	0	0	0	q = reserves
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	1	0	r = rubber
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	190	2	1	s = ship
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	143	0	t = sugar
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	88	u = veg-oil

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	a = alum
0	44	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	b = bop
0	0	55	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	c = cocoa
0	0	0	122	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	d = coffee
0	0	0	3	54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	e = copper
0	0	0	0	0	74	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	f = cpi
0	0	0	0	0	0	33	0	0	0	0	0	0	0	0	0	1	0	0	0	0	g = dlr
0	0	0	0	0	0	0	113	0	2	0	0	0	0	0	0	0	0	0	0	0	h = gnp
0	0	0	0	0	0	0	0	108	0	0	0	0	2	0	0	0	0	0	1	0	i = gold
0	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0	0	0	0	0	j = ipi
0	0	0	0	0	0	0	0	0	0	51	0	0	0	0	0	0	0	0	0	0	k = iron-steel
0	0	0	0	0	0	0	0	0	0	0	47	0	0	0	0	0	0	0	0	0	l = jobs
0	0	0	0	0	0	0	0	0	0	0	0	53	0	0	0	0	0	0	1	1	m = livestock
0	1	0	0	0	0	0	0	0	0	0	0	0	109	0	0	0	0	0	0	0	n = money-supply
0	0	0	0	0	0	0	0	0	0	0	0	0	0	47	0	0	0	1	0	0	o = nat-gas
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	66	0	0	0	0	11	p = oilseed
0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	46	0	0	0	0	q = reserves
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	38	0	1	0	r = rubber
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	189	2	1	s = ship
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	143	0	t = sugar
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	88	u = veg-oil

D-15 Reuters Veri Kümesinde Chi-Kare İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
26	0	0	18	0	0	0	0	0	1	0	0	0	0	0	0	0	3	0	0	0	a = alum
0	34	0	5	0	0	1	4	0	0	1	0	0	0	0	0	0	1	0	0	0	b = bop
0	0	39	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = cocoa
0	0	1	121	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	d = coffee
0	0	0	14	41	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	e = copper
0	0	0	16	0	54	0	3	0	0	0	0	0	0	1	0	0	0	1	0	0	f = cpi
0	0	0	6	0	0	26	2	0	0	0	0	0	0	0	0	0	0	0	0	0	g = dlr
0	0	0	17	0	0	1	93	0	1	0	0	1	1	0	0	0	0	1	0	0	h = gnp
0	0	0	28	2	0	0	0	79	0	0	0	0	0	0	0	0	0	2	0	0	i = gold
0	0	0	10	0	2	1	1	0	31	0	0	0	0	0	0	0	0	0	0	0	j = ipi
1	0	0	15	1	0	0	0	1	0	28	0	0	0	0	0	0	0	4	1	0	k = iron-steel
0	0	0	10	0	1	0	3	0	0	0	31	0	1	0	0	0	0	0	0	1	l = jobs
0	0	0	19	0	0	0	0	0	0	0	0	35	0	0	0	0	0	1	0	0	m = livestock
0	0	0	6	0	0	1	2	3	0	0	0	0	96	0	0	1	0	1	0	0	n = money-supply
0	0	0	3	0	0	0	0	0	0	0	0	0	0	41	0	0	0	4	0	0	o = nat-gas
0	0	0	17	0	0	0	0	0	0	0	0	3	1	0	42	0	0	1	2	12	p = oilseed
0	1	0	4	0	1	1	1	0	0	0	0	0	1	0	0	41	0	0	0	0	q = reserves
0	0	0	16	0	0	0	0	2	0	0	0	0	0	0	0	0	20	2	0	0	r = rubber
0	0	1	21	0	0	0	0	0	0	0	0	2	0	0	0	0	0	169	1	0	s = ship
0	0	0	19	0	0	0	0	0	0	0	0	2	0	0	1	0	0	2	120	0	t = sugar
0	0	0	15	0	0	0	0	0	0	0	0	2	0	0	4	0	0	2	0	70	u = veg-oil

D-16 Reuters Veri Kümesinde Chi-Kare İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
46	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	a = alum
0	39	0	1	0	0	0	4	0	1	0	0	0	0	0	0	1	0	0	0	0	b = bop
0	0	53	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	c = cocoa
0	0	1	119	0	0	0	2	0	0	0	0	1	0	0	0	0	0	1	0	0	d = coffee
0	0	0	0	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	e = copper
1	0	0	1	0	63	0	5	0	0	0	0	0	2	1	0	1	0	1	0	0	f = cpi
0	1	0	1	0	0	31	1	0	0	0	0	0	0	0	0	0	0	0	0	0	g = dlr
0	1	0	0	0	4	4	96	0	2	1	2	0	2	0	0	2	1	0	0	0	h = gnp
0	0	0	0	0	0	0	0	0	110	0	1	0	0	0	0	0	0	0	0	0	i = gold
1	0	0	0	0	1	1	3	0	38	0	0	1	0	0	0	0	0	0	0	0	j = ipi
2	0	0	1	1	0	0	0	1	0	41	0	0	0	0	0	0	0	5	0	0	k = iron-steel
0	0	0	0	0	1	0	2	0	1	0	42	0	0	0	1	0	0	0	0	0	l = jobs
0	0	0	0	0	0	0	0	0	0	0	0	51	0	0	1	0	0	2	0	1	m = livestock
0	1	0	1	0	0	1	5	0	0	0	0	0	100	0	0	1	0	1	0	0	n = money-supply
0	0	0	0	0	0	0	0	0	0	0	0	0	1	44	0	0	0	3	0	0	o = nat-gas
1	0	1	1	0	0	0	0	0	1	0	1	1	1	0	52	0	1	2	1	16	p = oilseed
0	2	0	0	0	1	1	1	0	0	0	0	0	1	0	0	44	0	0	0	0	q = reserves
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	39	0	0	0	r = rubber
3	1	0	4	0	0	0	1	0	0	3	0	2	0	1	3	0	0	175	1	0	s = ship
1	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	2	138	0	t = sugar
0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	17	1	0	0	1	71	u = veg-oil

D-17 Reuters Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
34	0	0	0	0	0	0	0	0	0	4	0	1	0	3	0	0	6	0	0	0	a = alum
0	38	0	0	0	0	4	3	0	0	0	0	0	1	0	0	0	0	0	0	0	b = bop
0	0	50	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = cocoa
0	0	75	36	1	0	0	3	0	0	0	0	2	1	0	0	0	3	1	0	2	d = coffee
0	0	0	0	44	0	0	0	11	0	0	0	1	0	0	0	0	1	0	0	0	e = copper
0	1	0	1	0	56	1	8	0	0	0	2	0	2	3	0	0	0	0	0	1	f = cpi
1	0	0	0	0	0	28	3	0	0	0	0	0	0	0	0	2	0	0	0	0	g = dlr
0	5	0	1	0	1	10	71	0	13	0	1	1	7	0	0	5	0	0	0	0	h = gnp
1	0	0	0	0	0	0	0	103	0	2	0	0	0	0	0	4	1	0	0	0	i = gold
1	0	0	0	1	0	0	6	0	25	2	2	2	0	1	0	0	4	1	0	0	j = ipi
3	0	0	0	1	0	0	0	0	0	1	39	0	0	0	1	0	0	5	1	0	k = iron-steel
0	0	0	0	0	0	0	4	0	1	1	38	0	0	0	0	1	2	0	0	0	l = jobs
0	0	0	1	0	0	1	1	0	0	1	1	39	0	1	0	0	4	0	2	4	m = livestock
0	0	0	0	0	0	4	3	0	0	0	0	0	101	0	0	2	0	0	0	0	n = money-supply
1	1	0	0	0	0	0	0	0	2	2	1	0	0	31	0	0	10	0	0	0	o = nat-gas
3	0	0	0	0	1	0	1	0	0	2	0	2	0	3	31	0	16	0	0	19	p = oilseed
0	0	0	0	0	0	2	4	0	0	0	0	0	12	1	0	31	0	0	0	0	q = reserves
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	37	0	0	1	r = rubber
14	1	0	0	0	1	1	2	0	0	11	1	7	0	94	5	1	25	26	5	0	s = ship
0	0	1	7	1	0	0	0	0	0	0	0	3	0	2	0	0	0	1	129	0	t = sugar
2	0	0	0	0	0	1	2	0	0	0	0	2	0	2	14	0	2	1	0	67	u = veg-oil

D-18 Reuters Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
34	0	0	0	0	1	0	0	0	0	4	0	1	1	0	2	0	0	5	0	0	a = alum
0	39	0	0	0	0	0	4	0	1	0	0	0	1	0	0	1	0	0	0	0	b = bop
0	0	56	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = cocoa
0	0	2	117	0	0	0	1	0	0	0	0	0	0	0	0	0	0	4	0	0	d = coffee
0	0	0	0	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	e = copper
0	1	0	1	0	65	0	5	0	1	0	0	0	0	2	0	0	0	0	0	0	f = cpi
0	0	0	0	0	0	32	2	0	0	0	0	0	0	0	0	0	0	0	0	0	g = dlr
0	2	0	1	0	2	1	101	0	1	0	1	0	3	0	0	2	0	1	0	0	h = gnp
0	0	0	0	0	0	0	0	110	0	0	0	0	0	0	0	0	0	1	0	0	i = gold
0	0	0	0	0	1	1	2	0	35	1	1	2	0	0	1	0	0	1	0	0	j = ipi
1	0	0	0	0	0	0	0	1	0	42	0	1	0	0	3	0	0	3	0	0	k = iron-steel
0	0	0	0	0	0	0	2	0	2	0	41	0	0	0	0	0	0	1	0	1	l = jobs
1	0	0	0	0	0	1	0	0	1	1	0	41	0	1	2	0	0	5	1	1	m = livestock
1	3	0	0	0	1	0	3	0	1	0	0	0	97	0	0	3	0	1	0	0	n = money-supply
0	0	0	0	0	0	0	0	0	0	0	1	0	0	33	1	0	0	11	0	2	o = nat-gas
1	0	0	0	0	0	0	0	0	0	0	0	3	0	1	45	2	0	9	0	17	p = oilseed
0	0	0	0	0	1	1	1	0	0	0	0	0	5	1	0	40	0	0	0	1	q = reserves
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	38	1	0	0	r = rubber
1	0	0	1	0	1	0	0	0	1	2	0	2	0	2	4	1	0	174	1	4	s = ship
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	3	140	0	t = sugar
1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	13	0	0	4	0	73	u = veg-oil

D-19 Reuters Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
26	0	0	18	0	0	0	0	0	0	1	0	0	0	0	0	0	0	3	0	0	a = alum
0	34	0	5	0	0	1	4	0	0	1	0	0	0	0	0	0	0	1	0	0	b = bop
0	0	39	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = cocoa
0	0	1	121	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	d = coffee
0	0	0	14	41	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	e = copper
0	0	0	16	0	54	0	3	0	0	0	0	0	0	1	0	0	0	1	0	0	f = cpi
0	0	0	6	0	0	26	2	0	0	0	0	0	0	0	0	0	0	0	0	0	g = dlr
0	0	0	17	0	0	1	93	0	1	0	0	1	1	0	0	0	0	1	0	0	h = gnp
0	0	0	28	2	0	0	0	79	0	0	0	0	0	0	0	0	0	2	0	0	i = gold
0	0	0	10	0	2	1	1	0	31	0	0	0	0	0	0	0	0	0	0	0	j = ipi
1	0	0	15	1	0	0	0	1	0	28	0	0	0	0	0	0	0	4	1	0	k = iron-steel
0	0	0	10	0	1	0	3	0	0	0	31	0	1	0	0	0	0	0	0	1	l = jobs
0	0	0	19	0	0	0	0	0	0	0	0	35	0	0	0	0	0	1	0	0	m = livestock
0	0	0	6	0	0	1	2	3	0	0	0	0	96	0	0	1	0	1	0	0	n = money-supply
0	0	0	3	0	0	0	0	0	0	0	0	0	0	41	0	0	0	4	0	0	o = nat-gas
0	0	0	17	0	0	0	0	0	0	0	0	3	1	0	42	0	0	1	2	12	p = oilseed
0	1	0	4	0	1	1	1	0	0	0	0	0	1	0	0	41	0	0	0	0	q = reserves
0	0	0	16	0	0	0	0	2	0	0	0	0	0	0	0	0	20	2	0	0	r = rubber
0	0	1	21	0	0	0	0	0	0	0	0	2	0	0	0	0	0	169	1	0	s = ship
0	0	0	19	0	0	0	0	0	0	0	0	2	0	0	1	0	0	2	120	0	t = sugar
0	0	0	15	0	0	0	0	0	0	0	0	2	0	0	4	0	0	2	0	70	u = veg-oil

D-20 Reuters Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle)

```
  a   b   c   d   e   f   g   h   i   j   k   l   m   n   o   p   q   r   s   t   u   <-- classified as
 41   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   1   5   0   0 |   a = alum
  0  38   0   0   0   0   1   4   0   1   0   0   0   0   0   0   1   0   1   0   0 |   b = bop
  0   0  52   5   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0 |   c = cocoa
  1   0   2 113   0   0   0   1   0   0   0   0   0   0   0   0   0   0   7   0   0 |   d = coffee
  0   0   0   0  55   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0 |   e = copper
  0   0   0   0   0  66   0   7   0   0   0   0   0   0   1   0   0   0   1   0   0 |   f = cpi
  0   1   0   0   0   0  32   1   0   0   0   0   0   0   0   0   0   0   0   0   0 |   g = dlr
  0   0   0   0   0   0   0 110   0   1   0   0   0   2   0   0   1   0   0   0   1 |   h = gnp
  0   0   0   0   1   0   0   0 107   0   0   0   0   0   1   0   0   0   2   0   0 |   i = gold
  0   0   0   0   0   2   0   2   0  40   0   0   1   0   0   0   0   0   0   0   0 |   j = ipi
  3   0   0   0   0   0   0   0   0   0  42   0   0   0   0   1   0   0   5   0   0 |   k = iron-steel
  0   0   0   0   0   2   0   3   0   1   0  40   0   0   0   0   0   0   1   0   0 |   l = jobs
  0   0   0   0   0   0   0   0   0   0   0   0  50   0   0   0   0   0   3   1   1 |   m = livestock
  0   0   0   1   0   0   0   5   0   0   0   0   1 100   0   0   2   0   1   0   0 |   n = money-supply
  0   0   0   0   0   1   0   0   0   0   0   0   0   0  44   0   0   0   3   0   0 |   o = nat-gas
  1   0   0   0   0   0   0   0   0   0   0   0   1   1   0  55   0   0   3   0  17 |   p = oilseed
  0   1   0   0   0   1   0   1   0   0   0   0   0   0   1   0   0   0  46   0   0 |   q = reserves
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  40   0 |   r = rubber
  1   0   0   0   0   0   0   0   0   0   2   0   2   0   0   1   0   0 187   1   0 |   s = ship
  0   0   0   0   0   0   0   0   0   0   0   0   3   0   0   1   0   0   2 138   0 |   t = sugar
  0   0   0   1   0   0   0   0   0   0   0   0   0   0   0  11   0   0   1   0  80 |   u = veg-oil
```

D-21 Reuters Veri Kümesinde PCA İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle)

```
  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t  u   <-- classified as
 23  0  0  2  1  0  0  0  1  0  0  0  0  0  1  1  1  0  1  0  0 |  a = alum
  0 21  0  1  0  0  0  1  0  0  0  0  0  0  0  0  1  0  1  0  1 |  b = bop
  1  0 12  0  1  0  0  0  0  1  0  0  0  0  0  1  2  3  5  2  2 |  c = cocoa
  0  0  2 30  1  0  1  1  1  0  0  4  2  0  0  6  2  1  5  1  1 |  d = coffee
  1  0  0  0 20  0  0  0  1  1  0  0  0  0  0  0  1  0  4  0  1 |  e = copper
  0  0  0  2  0 17  0  1  1 11  0  1  0  0  0  0  2  1  4  0  0 |  f = cpi
  0  0  0  2  0  0  4  0  1  0  0  0  0  0  0  1  1  1  4  1  0 |  g = dlr
  3  0  2  3  0  1  0 19  0 12  0  0  1  2  0  0  0  1  6  3  0 |  h = gnp
 15  0  3  1  4  0  1  0 27  5  0  0  0  2  2  1  2  0  6  0  1 |  i = gold
  2  0  0  1  1  0  0  0  0 17  0  1  0  0  0  0  0  0  0  0  0 |  j = ipi
  8  0  0  0  1  0  0  0  1  1  8  0  1  0  3  0  0  0  1  0  0 |  k = iron-steel
  0  0  0  1  0  0  0  2  0  1  0 15  0  0  1  0  0  0  1  0  0 |  l = jobs
  1  0  1  2  2  0  1  0  3  0  0  0 13  0  0  4  0  1  0  1  0 |  m = livestock
  0  0  0  4  0  0  0  3  0  0  0  0  0 18  0  0 28  0  4  0  0 |  n = money-supply
  5  0  1  0  0  0  0  0  0  1  0  0  0  0 10  0  0  1  2  1  0 |  o = nat-gas
  7  1  0  0  2  0  0  0  1  1  0  0  1  0  1 21  0  0  4  0  6 |  p = oilseed
  0  0  0  0  0  0  0  1  0  0  0  0  0  0  1  0 16  0  1  1  0 |  q = reserves
  1  0  0  1  2  0  0  0  0  1  0  0  0  0  1  0  0 16  0  0  0 |  r = rubber
  8  0  1  2  1  0  0  1  9  1  2  0  3  0  1  6  2  0 42  3  4 |  s = ship
  6  1  1  1  2  0  0  0  0  4  2  1  2  0  1  8  0  0  9 23  1 |  t = sugar
  4  0  1  2  2  0  0  0  0  2  1  0  1  0  0  9  1  0  9  0 18 |  u = veg-oil
```

D-22 Reuters Veri Kümesinde PCA İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle)

```
  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t  u   <-- classified as
 21  0  0  0  0  0  0  0  1  0  1  0  0  0  0  0  0  1  4  3  0 |  a = alum
  0 23  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  3  0  0 |  b = bop
  0  0 20  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  8  0  0 |  c = cocoa
  0  0  0 49  0  0  0  0  0  0  0  0  0  0  0  0  0  1  8  0  0 |  d = coffee
  0  0  0  0 26  0  0  0  2  0  0  0  0  0  0  0  0  0  1  0  0 |  e = copper
  0  0  0  0  0 38  0  1  0  0  0  0  0  0  0  0  0  0  1  0  0 |  f = cpi
  0  0  0  1  0  0  7  1  1  0  0  0  0  2  0  0  0  0  3  0  0 |  g = dlr
  0  0  0  0  0  0  0 45  0  1  0  0  0  0  0  0  0  0  5  2  0 |  h = gnp
  0  0  0  2  0  0  1  0 59  0  0  0  0  1  0  0  1  0  5  1  0 |  i = gold
  0  0  0  0  0  5  0  1  0 14  0  0  0  0  0  0  0  0  2  0  0 |  j = ipi
  0  0  0  0  0  0  0  0  2  2  1 12  0  0  0  0  1  0  0  6  0 |  k = iron-steel
  0  0  0  0  0  0  0  0  2  0  1  0 16  0  0  0  0  0  0  2  0 |  l = jobs
  0  0  0  2  0  0  0  0  1  0  0  0 16  0  1  0  0  0  6  1  2 |  m = livestock
  0  0  0  1  0  0  0  1  0  0  0  0  0 51  0  0  0  0  4  0  0 |  n = money-supply
  0  0  0  0  0  0  0  0  0  2  0  0  0  0  1 11  0  0  1  6  0  0 |  o = nat-gas
  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0 31  0  0  2  4  7 |  p = oilseed
  0  0  0  0  0  0  0  0  1  0  0  0  0  0  7  0  0 12  0  0  0  0 |  q = reserves
  0  0  0  2  1  0  0  0  0  0  0  0  0  0  0  0  0  0 15  2  1  1 |  r = rubber
  0  0  2  3  0  0  0  0  1  1  0  0  1  0  0  0  0  0  0 74  2  2 |  s = ship
  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  1  0  0  4 56  0 |  t = sugar
  0  0  1  0  0  0  0  0  1  0  0  0  0  0  0  0  4  0  0  6  1 37 |  u = veg-oil
```

D-23 Reuters Veri Kümesinde LSA İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
0	0	0	0	0	0	0	0	0	0	0	0	0	48	0	0	0	0	0	0	0	a = alum
0	0	0	0	0	0	0	0	0	0	0	0	0	46	0	0	0	0	0	0	0	b = bop
0	0	0	0	0	0	0	0	0	0	0	0	0	58	0	0	0	0	0	0	0	c = cocoa
0	0	0	4	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	d = coffee
0	0	0	0	2	0	0	0	0	0	0	0	0	55	0	0	0	0	0	0	0	e = copper
0	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0	0	0	0	0	f = cpi
0	0	0	0	0	0	0	0	0	0	0	0	0	34	0	0	0	0	0	0	0	g = dlr
0	0	0	0	0	0	0	0	0	0	0	0	0	115	0	0	0	0	0	0	0	h = gnp
0	0	0	0	0	0	0	0	18	0	0	0	0	93	0	0	0	0	0	0	0	i = gold
0	0	0	0	0	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0	j = ipi
0	0	0	0	0	0	0	0	0	0	0	0	0	51	0	0	0	0	0	0	0	k = iron-steel
0	0	0	0	0	0	0	0	0	0	0	3	0	44	0	0	0	0	0	0	0	l = jobs
0	0	0	0	0	0	0	0	0	0	0	0	0	55	0	0	0	0	0	0	0	m = livestock
0	0	0	0	0	0	0	0	0	0	0	0	0	110	0	0	0	0	0	0	0	n = money-supply
0	0	0	0	0	0	0	0	0	0	0	0	0	48	0	0	0	0	0	0	0	o = nat-gas
0	0	0	0	0	0	0	0	0	0	0	0	0	74	0	4	0	0	0	0	0	p = oilseed
0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	q = reserves
0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	r = rubber
0	0	0	0	0	0	0	0	0	0	0	0	0	194	0	0	0	0	0	0	0	s = ship
0	0	0	0	0	0	0	0	0	0	0	0	0	142	0	0	0	0	0	2	0	t = sugar
0	0	0	0	0	0	0	0	0	0	0	0	0	93	0	0	0	0	0	0	0	u = veg-oil

D-24 Reuters Veri Kümesinde LSA İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
9	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	30	6	0	a = alum
0	21	0	0	0	0	0	9	1	0	0	0	0	12	0	0	0	0	2	1	0	b = bop
0	0	33	14	0	0	0	0	0	0	0	0	0	0	0	1	0	0	7	2	1	c = cocoa
0	0	1	107	0	0	0	1	0	0	0	0	0	0	0	0	0	0	7	8	0	d = coffee
0	0	0	0	22	0	0	1	4	0	0	0	0	1	0	0	0	0	26	3	0	e = copper
0	0	0	1	0	47	0	6	0	0	0	0	0	5	0	0	0	0	15	1	0	f = cpi
0	0	0	1	0	0	6	7	1	0	0	0	0	1	0	0	0	0	10	8	0	g = dlr
0	0	0	1	0	2	0	84	4	0	0	0	0	6	0	0	0	0	8	10	0	h = gnp
0	0	0	0	0	0	0	10	63	0	0	0	0	0	0	0	0	0	28	10	0	i = gold
0	0	0	0	0	2	0	7	0	20	0	0	0	7	0	0	0	0	7	2	0	j = ipi
0	0	0	0	0	0	0	2	3	0	16	0	0	0	0	1	0	0	24	4	1	k = iron-steel
0	0	0	0	0	0	0	3	1	0	0	26	0	5	0	0	0	0	11	0	1	l = jobs
0	0	0	0	0	0	0	1	3	0	0	0	11	0	0	1	0	0	37	1	1	m = livestock
0	0	0	0	0	0	0	4	0	0	0	0	0	97	0	1	2	0	4	1	1	n = money-supply
0	0	0	0	0	0	0	1	0	0	0	0	0	1	4	1	0	0	35	4	2	o = nat-gas
0	0	0	0	0	0	0	1	0	0	0	0	0	2	0	33	0	0	21	9	12	p = oilseed
0	0	0	0	0	2	0	5	0	0	0	0	0	13	0	0	28	0	1	1	0	q = reserves
0	0	0	8	0	0	0	0	2	0	0	0	0	0	0	0	0	13	13	4	0	r = rubber
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	189	2	2	s = ship
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	123	1	t = sugar
0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	4	0	0	38	11	38	u = veg-oil

D-25 Reuters Veri Kümesinde LDA İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
28	0	0	0	0	0	0	0	0	0	5	1	0	0	2	0	0	12	0	0	0	a = alum
0	27	0	0	0	1	11	2	0	0	1	0	0	2	1	0	0	1	0	0	0	b = bop
0	0	50	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = cocoa
0	0	73	42	0	0	0	1	0	0	0	0	0	1	2	0	0	4	0	0	1	d = coffee
1	0	0	0	49	0	0	0	5	0	0	0	0	0	0	0	0	1	0	1	0	e = copper
0	2	0	0	0	34	0	7	0	0	0	3	0	15	2	1	0	10	0	0	1	f = cpi
1	0	0	0	0	0	28	2	0	0	1	0	0	0	0	0	2	0	0	0	0	g = dlr
0	0	0	2	0	2	3	65	0	19	1	1	0	17	0	0	4	1	0	0	0	h = gnp
0	0	0	0	0	0	1	2	69	0	0	0	0	0	2	0	36	1	0	0	0	i = gold
0	0	0	0	0	0	3	2	0	9	3	6	1	0	3	0	0	18	0	0	0	j = ipi
4	0	0	0	0	0	0	0	0	0	36	0	0	0	0	0	1	8	2	0	0	k = iron-steel
0	0	0	0	0	0	0	4	0	1	1	37	0	0	0	0	0	4	0	0	0	l = jobs
3	0	0	0	0	0	0	0	0	0	0	1	30	0	0	1	0	15	0	4	1	m = livestock
0	0	0	0	0	0	1	1	0	1	3	1	0	76	8	0	4	15	0	0	0	n = money-supply
1	0	0	0	0	0	0	0	0	0	0	0	0	0	45	0	0	2	0	0	0	o = nat-gas
0	0	0	0	0	2	0	0	0	0	0	0	1	0	1	42	0	24	1	1	6	p = oilseed
0	0	0	0	0	0	2	0	0	0	1	0	0	13	4	0	29	1	0	0	0	q = reserves
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	r = rubber
16	1	1	0	0	1	0	0	0	0	7	0	1	1	86	0	1	53	21	4	1	s = ship
0	0	2	4	0	0	0	0	0	0	1	0	2	0	2	0	0	1	0	132	0	t = sugar
0	1	0	1	0	0	0	0	0	0	1	0	0	0	10	20	0	6	0	0	54	u = veg-oil

D-26 Reuters Veri Kümesinde LDA İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
29	0	0	0	0	0	0	0	0	1	5	0	0	0	0	0	0	0	13	0	0	a = alum
0	38	0	0	0	0	2	3	1	0	0	0	0	0	0	0	0	0	2	0	0	b = bop
0	0	56	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	c = cocoa
0	1	2	116	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4	0	0	d = coffee
0	0	0	0	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	e = copper
0	2	0	1	0	63	1	5	0	0	0	0	0	1	1	0	0	0	1	0	0	f = cpi
0	1	0	0	0	0	30	2	0	0	0	0	0	1	0	0	0	0	0	0	0	g = dlr
0	0	0	2	0	4	3	99	0	0	0	1	0	4	0	0	0	1	1	0	0	h = gnp
0	0	0	0	0	0	0	0	106	0	0	0	0	0	0	0	4	0	1	0	0	i = gold
0	0	0	0	0	0	2	3	0	36	0	1	0	0	1	0	0	0	2	0	0	j = ipi
0	0	0	0	0	0	1	0	1	1	40	0	0	0	0	0	0	0	8	0	0	k = iron-steel
0	0	0	0	0	1	0	2	0	2	0	41	0	0	0	0	0	0	1	0	0	l = jobs
0	0	0	0	0	0	2	0	0	1	0	0	32	1	0	1	0	0	16	1	1	m = livestock
0	0	0	0	0	1	0	3	0	0	0	0	0	87	0	0	15	0	4	0	0	n = money-supply
0	0	0	0	0	1	0	0	0	0	0	0	0	0	45	0	0	0	2	0	0	o = nat-gas
0	1	0	0	0	0	0	0	0	0	0	0	2	0	0	35	0	0	25	0	15	p = oilseed
0	0	0	0	0	1	1	1	4	0	1	0	0	8	0	0	34	0	0	0	0	q = reserves
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	38	1	0	0	r = rubber
0	1	0	1	0	3	0	0	0	3	2	0	1	1	1	0	0	0	175	1	5	s = ship
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2	141	0	t = sugar
0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	7	0	0	7	0	77	u = veg-oil

D-27 20-Newsgroups Veri Kümesinde Özellik Çıkarımı Olmadan Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
0	0	0	0	9	0	0	90	0	0	0	0	0	0	0	0	0	0	0	1	a = alt.atheism
0	0	0	0	12	0	0	88	0	0	0	0	0	0	0	0	0	0	0	0	b = comp.graphics
0	0	1	0	7	0	0	91	0	0	0	0	0	0	1	0	0	0	0	0	c = comp.os.ms-windows.misc
0	0	0	1	14	0	0	84	0	0	0	0	0	0	1	0	0	0	0	0	d = comp.sys.ibm.pc.hardware
0	0	0	0	20	0	0	80	0	0	0	0	0	0	0	0	0	0	0	0	e = comp.sys.mac.hardware
0	0	0	0	12	0	0	88	0	0	0	0	0	0	0	0	0	0	0	0	f = comp.windows.x
0	0	0	0	15	0	0	84	0	0	0	0	0	0	1	0	0	0	0	0	g = misc.forsale
0	0	0	0	6	0	0	93	0	0	0	0	0	0	1	0	0	0	0	0	h = rec.autos
0	0	0	0	10	0	0	86	2	0	0	0	0	0	2	0	0	0	0	0	i = rec.motorcycles
0	0	0	0	6	0	0	92	0	1	1	0	0	0	0	0	0	0	0	0	j = rec.sport.baseball
0	0	0	0	7	0	0	69	0	0	22	0	0	0	2	0	0	0	0	0	k = rec.sport.hockey
0	0	0	0	8	0	0	91	0	0	0	0	0	0	1	0	0	0	0	0	l = sci.crypt
0	0	0	0	14	0	0	85	0	0	0	0	0	0	1	0	0	0	0	0	m = sci.electronics
0	0	0	0	10	0	0	88	0	0	0	0	0	0	2	0	0	0	0	0	n = sci.med
0	0	0	0	4	0	0	83	0	0	0	0	0	0	13	0	0	0	0	0	o = sci.space
0	0	0	0	4	0	0	94	0	0	0	0	0	0	0	2	0	0	0	0	p = soc.religion.christian
0	0	0	0	9	0	0	86	0	0	0	0	0	0	0	0	5	0	0	0	q = talk.politics.guns
0	0	0	0	7	0	0	90	0	0	0	0	0	0	1	0	0	2	0	0	r = talk.politics.mideast
0	0	0	0	6	0	0	89	0	0	0	0	0	0	0	0	1	0	2	2	s = talk.politics.misc
0	0	0	0	8	0	0	89	0	0	0	0	0	0	0	0	1	0	2	0	t = talk.religion.misc

D-28 20-Newsgroups Veri Kümesinde Özellik Çıkarımı Olmadan Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
67	0	0	0	0	0	0	7	0	0	0	0	0	0	2	6	0	1	0	17	a = alt.atheism
1	68	5	3	2	3	1	10	0	0	0	1	5	0	1	0	0	0	0	0	b = comp.graphics
0	4	63	13	1	4	3	9	0	1	0	0	1	0	1	0	0	0	0	0	c = comp.os.ms-windows.misc
0	8	10	65	2	3	5	4	0	1	0	1	0	0	0	1	0	0	0	0	d = comp.sys.ibm.pc.hardware
0	1	4	4	81	1	0	5	0	0	0	0	0	1	1	0	1	0	0	1	e = comp.sys.mac.hardware
0	7	10	1	0	72	0	4	1	0	0	0	3	0	1	0	0	1	0	0	f = comp.windows.x
0	1	0	5	3	0	71	12	0	1	1	0	4	0	1	0	0	0	0	1	g = misc.forsale
0	1	2	0	0	0	3	83	2	0	0	0	5	0	0	1	1	0	2	0	h = rec.autos
0	0	0	0	0	0	1	8	89	1	0	0	1	0	0	0	0	0	0	0	i = rec.motorcycles
0	0	0	0	0	1	3	6	0	84	5	0	0	1	0	0	0	0	0	0	j = rec.sport.baseball
1	0	1	0	0	0	0	2	0	2	90	0	0	0	0	0	0	1	2	1	k = rec.sport.hockey
0	0	2	0	0	1	0	6	0	0	0	87	2	0	0	0	2	0	0	0	l = sci.crypt
0	0	3	7	2	3	4	17	0	0	0	2	59	1	2	0	0	0	0	0	m = sci.electronics
1	2	0	0	0	0	1	8	1	0	0	0	1	84	1	0	1	0	0	0	n = sci.med
0	1	0	1	0	0	0	6	0	0	0	0	0	1	91	0	0	0	0	0	o = sci.space
0	0	0	0	0	0	0	9	0	0	0	0	1	2	0	82	0	3	1	2	p = soc.religion.christian
1	0	1	0	0	0	0	4	0	0	0	3	0	1	1	1	81	0	5	2	q = talk.politics.guns
1	1	1	0	0	0	0	6	0	0	0	0	0	0	0	1	1	87	1	1	r = talk.politics.mideast
1	0	0	0	0	0	0	5	1	1	2	0	0	1	1	2	11	6	54	15	s = talk.politics.misc
19	0	0	1	0	0	0	9	0	0	0	0	0	1	0	15	5	0	10	40	t = talk.religion.misc

D-29 20-Newsgroups Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En Kötü Karmaşıklık Matrisi (C4.5 Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
86	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	3	0	8	a = alt.atheism
0	85	3	7	2	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	b = comp.graphics
0	6	81	3	0	5	1	0	1	1	0	0	1	0	0	1	0	0	0	0	c = comp.os.ms-windows.misc
0	5	4	82	2	2	1	0	0	0	0	0	4	0	0	0	0	0	0	0	d = comp.sys.ibm.pc.hardware
0	0	1	7	87	1	1	1	0	0	0	0	0	0	0	1	1	0	0	0	e = comp.sys.mac.hardware
0	3	3	1	0	89	1	0	1	0	0	0	0	1	0	1	0	0	0	0	f = comp.windows.x
1	0	1	4	3	3	78	2	2	1	0	0	1	1	1	2	0	0	0	0	g = misc.forsale
1	0	0	0	0	0	1	97	0	0	0	0	0	0	0	1	0	0	0	0	h = rec.autos
0	0	0	0	0	2	1	1	94	1	0	1	0	0	0	0	0	0	0	0	i = rec.motorcycles
0	1	1	0	0	1	1	0	3	85	6	1	0	0	0	1	0	0	0	0	j = rec.sport.baseball
1	0	1	0	1	0	2	1	0	4	85	0	0	1	0	1	0	1	2	0	k = rec.sport.hockey
0	0	0	0	0	0	3	0	0	3	0	90	3	0	0	0	1	0	0	0	l = sci.crypt
0	2	1	1	0	0	2	2	1	1	0	4	82	1	0	2	1	0	0	0	m = sci.electronics
0	0	1	0	0	0	1	0	1	0	1	1	1	91	1	1	0	1	0	0	n = sci.med
0	0	0	0	0	0	0	0	1	0	0	0	0	0	95	2	2	0	0	0	o = sci.space
1	1	0	0	0	2	1	1	0	1	0	0	0	2	0	87	1	2	0	1	p = soc.religion.christian
1	0	0	0	1	2	0	0	0	0	0	2	0	0	2	1	88	1	1	1	q = talk.politics.guns
0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	3	1	92	2	0	r = talk.politics.mideast
0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	92	4	s = talk.politics.misc
12	0	0	0	1	0	0	1	0	1	1	0	0	0	0	3	2	0	7	72	t = talk.religion.misc

D-30 20-Newsgroups Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
97	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	1	0	a = alt.atheism
0	96	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b = comp.graphics
0	2	91	3	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = comp.os.ms-windows.misc
0	2	0	94	3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	d = comp.sys.ibm.pc.hardware
0	0	0	0	97	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	e = comp.sys.mac.hardware
0	0	1	0	0	96	1	0	1	1	0	0	0	0	0	0	0	0	0	0	f = comp.windows.x
0	0	0	0	0	0	99	0	1	0	0	0	0	0	0	0	0	0	0	0	g = misc.forsale
0	0	0	0	0	0	0	98	1	0	0	0	0	0	0	0	0	0	0	1	h = rec.autos
0	0	0	0	0	0	1	0	97	0	1	1	0	0	0	0	0	0	0	0	i = rec.motorcycles
0	0	0	0	0	0	0	0	1	92	5	2	0	0	0	0	0	0	0	0	j = rec.sport.baseball
0	0	0	0	1	1	0	0	0	0	98	0	0	0	0	0	0	0	0	0	k = rec.sport.hockey
0	0	0	0	0	0	0	0	0	0	0	96	4	0	0	0	0	0	0	0	l = sci.crypt
0	1	0	1	1	0	0	0	0	0	0	0	94	3	0	0	0	0	0	0	m = sci.electronics
0	0	0	0	0	0	1	0	0	0	0	0	0	94	4	1	0	0	0	0	n = sci.med
0	0	0	0	0	1	0	0	0	0	0	0	0	1	95	1	1	1	0	0	o = sci.space
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	p = soc.religion.christian
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	96	3	0	0	q = talk.politics.guns
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	99	0	0	r = talk.politics.mideast
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	98	1	0	s = talk.politics.misc
16	0	0	0	0	0	0	0	1	0	0	0	0	0	0	5	3	0	5	70	t = talk.religion.misc

D-31 20-Newsgroups Veri Kümesinde Chi-Kare İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
43	0	0	0	0	0	7	10	0	0	0	0	1	12	1	6	0	0	2	18	a = alt.atheism
3	34	10	1	0	9	6	19	0	0	0	0	7	4	4	0	0	0	0	3	b = comp.graphics
2	4	41	2	1	20	4	14	2	1	0	0	6	1	2	0	0	0	0	0	c = comp.os.ms-windows.misc
2	4	3	24	7	7	15	18	1	0	0	0	14	1	1	0	0	0	3	0	d = comp.sys.ibm.pc.hardware
6	0	1	10	21	3	11	19	3	0	0	2	16	3	2	0	0	0	0	3	e = comp.sys.mac.hardware
3	4	11	0	0	44	2	20	1	0	0	0	7	3	1	2	0	0	0	2	f = comp.windows.x
1	0	2	2	1	4	57	22	1	1	1	0	5	1	0	2	0	0	0	0	g = misc.forsale
0	0	0	0	0	3	2	75	3	2	0	0	4	4	1	3	0	0	1	2	h = rec.autos
0	0	2	0	0	1	9	14	70	0	0	0	2	0	0	0	0	0	2	0	i = rec.motorcycles
3	0	0	0	0	1	11	16	0	44	11	0	3	4	1	0	1	0	1	4	j = rec.sport.baseball
2	0	1	1	0	0	6	9	0	9	57	0	1	4	0	5	1	0	2	2	k = rec.sport.hockey
0	4	2	0	0	2	5	11	1	2	0	38	10	7	4	1	1	0	7	5	l = sci.crypt
5	3	1	3	0	4	8	25	0	3	1	0	34	5	3	3	0	0	0	2	m = sci.electronics
4	0	0	0	0	0	6	18	0	0	0	0	3	55	5	2	0	0	3	4	n = sci.med
1	0	2	0	1	0	4	17	0	0	0	0	4	5	59	0	1	0	4	2	o = sci.space
8	0	0	0	0	1	0	17	0	1	1	0	5	2	2	51	0	0	1	11	p = soc.religion.christian
5	0	0	1	0	0	7	13	0	0	0	3	2	10	2	1	30	0	15	11	q = talk.politics.guns
8	0	0	0	0	2	4	10	0	0	0	0	4	8	1	1	0	52	6	4	r = talk.politics.mideast
5	0	4	0	0	0	4	5	0	1	0	0	3	12	2	3	6	4	37	14	s = talk.politics.misc
19	0	1	0	0	0	9	7	0	1	0	0	5	7	1	6	3	0	10	31	t = talk.religion.misc

D-32 20-Newsgroups Veri Kümesinde Chi-Kare İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
47	1	0	0	0	0	1	2	0	0	0	0	7	2	1	7	0	1	5	26	a = alt.atheism
1	61	5	4	1	8	1	2	0	0	0	0	13	2	2	0	0	0	0	0	b = comp.graphics
0	6	44	11	0	15	1	2	0	1	0	0	15	2	1	0	0	0	1	1	c = comp.os.ms-windows.misc
0	5	10	48	8	1	6	1	1	0	0	0	17	0	0	0	0	0	1	2	d = comp.sys.ibm.pc.hardware
0	1	4	10	63	2	3	2	0	0	0	1	11	1	2	0	0	0	0	0	e = comp.sys.mac.hardware
0	7	9	0	0	62	3	0	0	0	0	0	14	1	3	0	0	0	0	1	f = comp.windows.x
2	2	3	3	4	3	58	3	0	2	0	0	15	0	1	0	0	0	1	3	g = misc.forsale
2	1	0	1	0	0	2	67	2	1	0	0	16	1	0	1	2	0	3	1	h = rec.autos
1	0	0	0	0	0	1	8	79	0	0	0	8	0	0	0	0	0	1	2	i = rec.motorcycles
2	0	1	0	1	0	3	0	0	73	3	0	9	2	0	0	0	0	2	4	j = rec.sport.baseball
0	0	0	0	0	1	1	0	0	10	72	0	10	2	0	0	1	1	1	1	k = rec.sport.hockey
2	1	0	1	1	2	2	0	0	0	0	73	6	3	0	1	4	0	4	0	l = sci.crypt
1	2	2	8	1	4	6	5	0	2	1	2	62	2	2	0	0	0	0	0	m = sci.electronics
3	3	0	0	0	1	2	0	1	0	0	0	15	67	1	1	0	0	3	3	n = sci.med
1	2	0	2	1	4	0	1	0	0	0	0	10	3	73	0	1	1	0	1	o = sci.space
4	0	0	0	0	0	1	1	0	0	0	0	9	1	0	72	1	0	2	9	p = soc.religion.christian
1	0	0	0	0	1	0	1	0	1	0	2	7	0	1	0	56	1	14	15	q = talk.politics.guns
4	0	0	0	0	1	0	3	0	0	0	1	8	0	0	1	2	70	10	0	r = talk.politics.mideast
4	0	1	0	0	0	2	4	0	1	0	0	7	3	1	1	16	5	49	6	s = talk.politics.misc
20	0	0	0	0	0	1	1	1	0	0	0	11	3	0	15	7	0	9	32	t = talk.religion.misc

D-33 20-Newsgroups Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En Kötü Karmaşıklık Matrisi (Rasgele Orman Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
22	1	2	4	1	2	2	2	1	4	2	2	3	10	5	12	2	3	6	14	a = alt.atheism
1	27	12	8	4	17	0	3	3	2	1	1	7	5	5	0	1	0	1	2	b = comp.graphics
1	10	40	8	3	17	2	1	1	1	2	1	6	5	1	1	0	0	0	0	c = comp.os.ms-windows.misc
4	6	17	19	13	9	5	4	2	2	0	0	10	2	3	0	4	0	0	0	d = comp.sys.ibm.pc.hardware
3	10	2	16	32	2	5	1	0	1	1	1	15	3	3	0	2	1	1	1	e = comp.sys.mac.hardware
2	16	24	7	2	23	2	1	0	2	0	2	9	7	2	0	1	0	0	0	f = comp.windows.x
1	1	3	2	4	5	57	4	0	5	2	0	10	3	1	1	0	0	1	0	g = misc.forsale
3	1	2	1	0	2	0	60	5	3	0	1	8	3	0	2	2	1	3	3	h = rec.autos
2	2	0	1	2	2	0	6	75	0	2	1	1	5	0	0	0	0	1	0	i = rec.motorcycles
4	2	2	1	2	2	6	1	1	33	24	2	5	4	2	1	0	2	4	2	j = rec.sport.baseball
3	1	1	0	1	1	0	1	0	22	56	0	5	1	4	0	1	0	3	0	k = rec.sport.hockey
5	4	1	1	2	4	2	2	1	3	1	56	4	2	2	2	4	0	3	1	l = sci.crypt
3	6	3	8	8	10	5	7	3	2	2	5	18	12	4	1	2	0	0	1	m = sci.electronics
13	3	4	2	4	5	3	1	3	1	2	0	12	23	7	2	4	6	3	2	n = sci.med
5	4	4	1	2	7	2	0	1	2	2	1	6	9	47	0	4	0	3	0	o = sci.space
13	1	1	2	1	2	1	0	0	2	1	2	7	3	1	47	2	3	4	7	p = soc.religion.christian
7	2	0	0	4	0	1	4	1	7	3	4	4	5	3	0	38	0	10	7	q = talk.politics.guns
5	1	0	0	1	2	2	5	0	3	3	3	4	8	1	4	5	41	7	5	r = talk.politics.mideast
14	0	4	1	2	4	1	4	2	3	0	3	2	9	3	4	13	10	18	3	s = talk.politics.misc
14	1	2	0	2	2	0	6	1	2	0	0	4	6	4	12	8	2	10	24	t = talk.religion.misc

D-34 20-Newsgroups Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
41	0	0	3	1	3	1	2	0	0	0	0	1	8	2	15	0	5	3	15	a = alt.atheism
1	50	10	1	7	18	1	2	0	1	0	0	1	2	4	0	0	1	1	0	b = comp.graphics
4	11	36	8	4	21	1	1	1	0	2	2	1	2	3	1	0	0	2	0	c = comp.os.ms-windows.misc
1	5	11	40	17	9	7	4	0	0	0	0	2	2	1	0	0	0	0	1	d = comp.sys.ibm.pc.hardware
2	3	4	7	57	10	2	0	0	1	0	0	4	2	2	3	0	0	1	2	e = comp.sys.mac.hardware
1	13	18	1	1	52	4	0	0	0	0	1	1	1	5	0	1	0	0	1	f = comp.windows.x
3	2	2	1	3	9	65	4	1	0	3	1	1	2	0	1	0	0	0	2	g = misc.forsale
3	0	0	1	6	9	4	62	3	0	0	1	0	3	0	4	1	1	1	1	h = rec.autos
2	0	0	2	2	2	1	3	83	0	0	0	0	1	1	0	0	1	0	2	i = rec.motorcycles
6	0	0	0	1	4	4	0	0	53	13	1	2	8	0	2	0	1	2	3	j = rec.sport.baseball
5	1	0	0	2	3	1	0	0	14	66	0	1	2	1	2	1	1	0	0	k = rec.sport.hockey
6	1	1	0	2	8	3	0	0	1	0	66	1	1	1	1	2	1	3	2	l = sci.crypt
1	5	1	5	7	15	9	7	1	1	1	6	20	8	8	2	1	0	1	1	m = sci.electronics
10	5	0	0	5	10	4	0	1	0	0	1	3	44	3	4	1	0	6	3	n = sci.med
5	1	0	1	0	5	1	1	1	1	0	1	1	1	74	1	2	0	2	2	o = sci.space
8	0	0	0	1	6	1	1	0	1	1	1	0	5	0	64	1	2	3	5	p = soc.religion.christian
2	0	0	2	0	1	1	2	0	1	1	5	0	6	2	2	46	1	15	13	q = talk.politics.guns
11	0	0	0	1	3	0	1	0	0	0	1	1	3	0	1	1	70	5	2	r = talk.politics.mideast
7	1	2	1	0	3	0	2	1	3	0	2	0	6	2	4	10	6	37	13	s = talk.politics.misc
24	0	0	0	0	1	0	1	2	2	2	0	0	11	1	19	7	1	9	20	t = talk.religion.misc

D-35 20-Newsgroups Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En Kötü Karmaşıklık Matrisi (RIPPER Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
29	0	10	0	7	0	7	1	0	1	1	0	1	5	2	8	1	7	7	13	a = alt.atheism
5	42	14	0	3	3	9	2	0	1	0	0	0	12	1	0	0	1	4	3	b = comp.graphics
6	6	36	11	5	6	10	0	0	1	0	1	1	8	2	1	0	0	3	3	c = comp.os.ms-windows.misc
6	6	7	32	13	5	12	1	0	1	0	0	0	6	1	0	2	0	4	4	d = comp.sys.ibm.pc.hardware
4	2	8	8	46	2	13	1	0	1	0	0	3	7	0	0	0	0	3	2	e = comp.sys.mac.hardware
3	8	22	0	1	34	11	0	0	0	2	1	0	8	2	1	0	0	3	4	f = comp.windows.x
4	3	11	3	5	1	50	6	0	3	0	0	0	5	1	0	0	0	5	3	g = misc.forsale
2	0	6	0	4	0	6	53	4	0	0	1	1	10	0	0	2	0	5	6	h = rec.autos
1	1	4	0	2	1	6	7	68	1	1	0	1	4	1	0	1	0	0	1	i = rec.motorcycles
3	0	4	1	5	0	6	0	0	53	10	1	0	7	1	0	1	0	5	3	j = rec.sport.baseball
2	0	5	0	3	0	7	1	0	5	57	0	0	11	3	0	0	0	3	3	k = rec.sport.hockey
1	2	3	1	3	3	7	0	0	0	0	60	2	3	1	0	4	0	7	3	l = sci.crypt
6	3	12	3	7	1	15	5	1	1	1	4	18	10	4	0	0	0	6	3	m = sci.electronics
6	3	8	0	5	1	11	0	1	0	1	0	1	56	0	0	0	0	4	3	n = sci.med
4	3	6	0	4	0	5	0	0	0	3	2	0	12	58	0	1	0	1	1	o = sci.space
5	0	7	0	2	0	8	0	0	0	0	0	0	4	0	53	1	2	5	13	p = soc.religion.christian
5	0	8	1	2	2	6	0	0	2	0	1	1	13	2	0	41	1	11	4	q = talk.politics.guns
5	0	3	0	2	0	6	0	1	1	0	0	0	10	0	2	1	61	4	4	r = talk.politics.mideast
8	0	12	0	9	0	11	1	0	1	1	0	0	16	1	3	12	4	16	5	s = talk.politics.misc
19	0	8	1	6	1	14	1	0	0	1	1	0	8	0	13	6	2	9	10	t = talk.religion.misc

D-36 20-Newsgroups Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
47	1	0	0	0	0	1	2	0	0	0	0	7	2	1	7	0	1	5	26	a = alt.atheism
1	61	5	4	1	8	1	2	0	0	0	0	13	2	2	0	0	0	0	0	b = comp.graphics
0	6	44	11	0	15	1	2	0	1	0	0	15	2	1	0	0	0	1	1	c = comp.os.ms-windows.misc
0	5	10	48	8	1	6	1	1	0	0	0	17	0	0	0	0	0	1	2	d = comp.sys.ibm.pc.hardware
0	1	4	10	63	2	3	2	0	0	0	1	11	1	2	0	0	0	0	0	e = comp.sys.mac.hardware
0	7	9	0	0	62	3	0	0	0	0	0	14	1	3	0	0	0	0	1	f = comp.windows.x
2	2	3	3	4	3	58	3	0	2	0	0	15	0	1	0	0	0	1	3	g = misc.forsale
2	1	0	1	0	0	2	67	2	1	0	0	16	1	0	1	2	0	3	1	h = rec.autos
1	0	0	0	0	0	1	8	79	0	0	0	8	0	0	0	0	0	1	2	i = rec.motorcycles
2	0	1	0	1	0	3	0	0	73	3	0	9	2	0	0	0	0	2	4	j = rec.sport.baseball
0	0	0	0	0	1	1	0	0	10	72	0	10	2	0	0	1	1	1	1	k = rec.sport.hockey
2	1	0	1	1	2	2	0	0	0	0	73	6	3	0	1	4	0	4	0	l = sci.crypt
1	2	2	8	1	4	6	5	0	2	1	2	62	2	2	0	0	0	0	0	m = sci.electronics
3	3	0	0	0	1	2	0	1	0	0	0	15	67	1	1	0	0	3	3	n = sci.med
1	2	0	2	1	4	0	1	0	0	0	0	10	3	73	0	1	1	0	1	o = sci.space
4	0	0	0	0	0	1	1	0	0	0	0	9	1	0	72	1	0	2	9	p = soc.religion.christian
1	0	0	0	0	1	0	1	0	1	0	2	7	0	1	0	56	1	14	15	q = talk.politics.guns
4	0	0	0	0	1	0	3	0	0	0	1	8	0	0	1	2	70	10	0	r = talk.politics.mideast
4	0	1	0	0	0	2	4	0	1	0	0	7	3	1	1	16	5	49	6	s = talk.politics.misc
20	0	0	0	0	0	1	1	1	0	0	0	11	3	0	15	7	0	9	32	t = talk.religion.misc

D-37 20-Newsgroups Veri Kümesinde PCA İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
0	0	0	0	0	4	0	62	0	0	0	0	0	0	30	0	0	0	0	4	a = alt.atheism
0	0	0	0	0	4	2	64	0	0	1	0	0	0	29	0	0	0	0	0	b = comp.graphics
0	0	3	0	0	3	1	67	0	0	0	0	0	0	26	0	0	0	0	0	c = comp.os.ms-windows.misc
0	0	0	0	0	1	4	57	0	0	0	0	0	0	38	0	0	0	0	0	d = comp.sys.ibm.pc.hardware
0	0	0	0	5	1	1	61	0	0	0	0	0	0	32	0	0	0	0	0	e = comp.sys.mac.hardware
0	0	0	0	0	4	2	65	0	0	0	0	0	0	29	0	0	0	0	0	f = comp.windows.x
0	0	0	0	0	1	8	51	0	0	0	0	0	0	40	0	0	0	0	0	g = misc.forsale
0	0	0	0	0	0	3	69	0	0	0	0	0	0	28	0	0	0	0	0	h = rec.autos
0	0	0	0	0	3	1	64	0	0	0	0	0	0	32	0	0	0	0	0	i = rec.motorcycles
0	0	0	0	0	1	3	62	0	1	1	0	0	0	32	0	0	0	0	0	j = rec.sport.baseball
0	0	0	0	0	3	0	59	0	0	3	0	0	0	35	0	0	0	0	0	k = rec.sport.hockey
0	0	0	0	0	6	1	65	0	0	1	0	0	0	27	0	0	0	0	0	l = sci.crypt
0	0	0	0	0	2	2	67	0	0	0	0	1	0	28	0	0	0	0	0	m = sci.electronics
0	0	0	0	0	5	0	62	0	0	0	0	0	0	33	0	0	0	0	0	n = sci.med
0	0	0	0	0	6	1	62	0	0	0	0	0	1	30	0	0	0	0	0	o = sci.space
0	0	0	0	0	4	0	63	0	0	0	0	0	0	32	0	0	0	0	1	p = soc.religion.christian
0	0	0	0	0	5	2	59	0	0	0	0	0	0	32	0	1	0	0	1	q = talk.politics.guns
0	0	0	0	0	5	0	61	0	0	0	0	0	0	30	0	0	4	0	0	r = talk.politics.mideast
0	0	0	0	0	3	2	70	0	0	0	0	0	0	22	0	0	0	0	3	s = talk.politics.misc
5	0	0	0	0	1	0	62	0	0	0	0	0	0	26	0	1	0	3	2	t = talk.religion.misc

D-38 20-Newsgroups Veri Kümesinde PCA İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
60	0	1	0	0	0	0	1	1	1	2	0	1	0	3	4	2	1	1	22	a = alt.atheism
2	63	8	5	3	6	0	2	0	0	0	2	6	0	0	0	1	0	0	2	b = comp.graphics
1	9	55	12	4	5	4	0	0	2	0	2	1	0	2	1	1	0	1	0	c = comp.os.ms-windows.misc
0	4	9	58	7	5	7	3	1	0	1	1	4	0	0	0	0	0	0	0	d = comp.sys.ibm.pc.hardware
0	4	6	7	68	2	2	2	0	1	0	0	6	0	1	1	0	0	0	0	e = comp.sys.mac.hardware
0	7	12	2	0	65	1	0	0	0	0	2	7	0	2	0	1	1	0	0	f = comp.windows.x
1	2	1	5	4	1	72	3	1	0	0	2	4	0	2	1	0	1	0	0	g = misc.forsale
0	1	2	0	1	0	5	76	2	1	0	0	7	0	0	1	1	1	1	1	h = rec.autos
0	1	0	0	0	1	1	2	89	1	2	1	0	0	0	0	0	1	1	1	i = rec.motorcycles
1	1	1	0	0	1	1	0	2	82	5	0	0	2	1	0	0	1	1	1	j = rec.sport.baseball
1	0	1	0	0	1	0	3	0	4	88	0	1	0	0	0	0	0	1	0	k = rec.sport.hockey
0	2	3	2	1	3	0	0	0	0	1	79	3	0	0	1	3	0	2	0	l = sci.crypt
1	2	4	6	3	5	7	6	0	0	1	1	55	2	4	0	2	0	1	0	m = sci.electronics
2	0	2	1	0	0	1	0	0	1	0	0	3	84	3	1	1	0	0	1	n = sci.med
1	0	2	1	0	0	0	3	0	0	0	1	0	4	86	0	0	0	1	1	o = sci.space
2	2	1	1	0	1	1	3	0	0	0	1	2	0	0	78	2	1	1	4	p = soc.religion.christian
1	1	0	0	0	0	0	2	0	1	0	2	0	3	2	3	68	1	11	5	q = talk.politics.guns
1	1	2	1	0	1	0	0	0	1	0	1	1	0	0	0	2	85	4	0	r = talk.politics.mideast
5	1	0	1	0	0	0	1	0	1	2	3	1	1	4	3	10	5	47	15	s = talk.politics.misc
18	1	0	2	0	0	0	2	1	0	0	0	0	0	0	10	4	3	12	47	t = talk.religion.misc

D-39 20-Newsgroups Veri Kümesinde LSA İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	75	a = alt.atheism
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	65	b = comp.graphics
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	77	c = comp.os.ms-windows.misc
0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	63	d = comp.sys.ibm.pc.hardware
0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	25	73	e = comp.sys.mac.hardware
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	32	67	f = comp.windows.x
0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	39	59	g = misc.forsale
0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	30	67	h = rec.autos
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	30	69	i = rec.motorcycles
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	74	j = rec.sport.baseball
0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	35	63	k = rec.sport.hockey
0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	27	70	l = sci.crypt
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	63	m = sci.electronics
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	26	72	n = sci.med
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	30	68	o = sci.space
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	30	70	p = soc.religion.christian
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	31	69	q = talk.politics.guns
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	32	65	r = talk.politics.mideast
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	32	67	s = talk.politics.misc
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	25	70	t = talk.religion.misc

D-40 20-Newsgroups Veri Kümesinde LSA İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
66	1	0	0	0	1	0	1	0	1	1	1	1	1	1	5	1	0	0	19	a = alt.atheism
0	62	7	4	2	4	2	1	0	3	0	1	6	1	0	2	0	0	3	2	b = comp.graphics
0	5	54	9	3	5	5	1	0	0	2	2	8	1	2	1	0	0	1	1	c = comp.os.ms-windows.misc
0	6	12	58	4	4	5	2	1	1	0	1	4	1	0	1	0	0	0	0	d = comp.sys.ibm.pc.hardware
0	2	6	8	64	1	3	1	0	1	0	1	7	0	0	1	1	1	2	1	e = comp.sys.mac.hardware
0	9	5	4	1	57	3	1	1	0	1	0	4	3	6	0	1	1	1	2	f = comp.windows.x
0	3	2	5	6	1	60	0	0	2	0	0	8	0	1	1	1	2	3	5	g = misc.forsale
0	2	2	1	1	2	2	67	3	1	0	1	6	2	2	0	2	2	2	2	h = rec.autos
0	1	0	0	1	1	2	5	81	0	1	0	2	1	0	0	1	0	3	1	i = rec.motorcycles
0	4	0	0	0	1	4	1	1	79	4	1	0	2	0	1	0	0	2	0	j = rec.sport.baseball
1	1	3	2	1	1	1	0	0	2	81	0	2	0	1	0	0	1	2	1	k = rec.sport.hockey
0	3	3	1	3	1	0	0	0	0	2	80	4	0	0	0	1	0	2	0	l = sci.crypt
2	2	4	5	3	4	6	5	0	1	0	3	50	2	4	2	1	1	4	1	m = sci.electronics
0	2	2	0	1	0	2	1	0	1	0	0	6	76	2	2	2	1	0	2	n = sci.med
1	0	2	2	1	0	0	2	0	0	1	0	3	3	83	0	1	0	1	0	o = sci.space
3	2	1	2	1	0	0	1	1	0	0	0	0	2	1	74	4	2	1	5	p = soc.religion.christian
1	0	2	0	0	0	2	0	1	0	1	5	0	5	0	2	67	0	7	7	q = talk.politics.guns
2	3	0	1	1	1	0	0	1	0	0	0	0	0	0	0	1	1	82	5	2	r = talk.politics.mideast
1	0	2	0	1	0	2	2	2	1	3	2	0	0	2	1	9	5	53	14	s = talk.politics.misc
18	1	0	3	0	0	0	0	1	1	1	0	1	3	0	10	4	0	9	48	t = talk.religion.misc

D-41 20-Newsgroups Veri Kümesinde LDA İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle)

```
  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t   <-- classified as
  4  0  0  3 12  0  0  7  0 17  0  1  1  6  3 37  0  1  1  7 |  a = alt.atheism
  1 13  2  7 16 23  1  6  0 14  0  0  1  1  6  7  0  1  0  1 |  b = comp.graphics
  0  3  7 17 12 37  0  2  0 10  0  0  1  0  2  8  0  0  0  1 |  c = comp.os.ms-windows.misc
  0  0  2 13 30  6  1 11  0 20  0  0  1  2  0 14  0  0  0  0 |  d = comp.sys.ibm.pc.hardware
  0  1  2 10 50  4  2  5  0 10  0  0  0  3  2 11  0  0  0  0 |  e = comp.sys.mac.hardware
  0  3  4 12 11 42  0  3  0 12  0  0  0  2  3  7  0  0  1  0 |  f = comp.windows.x
  0  1  0 10 16  3 28 12  0 18  0  0  1  0  2  9  0  0  0  0 |  g = misc.forsale
  0  0  0  6 12  0  2 60  2  6  0  0  4  0  1  6  1  0  0  0 |  h = rec.autos
  0  0  0  4  8  1  1  9 48 15  0  0  0  5  1  8  0  0  0  0 |  i = rec.motorcycles
  0  0  0  2 12  0  1  3  0 64  6  0  0  0  0 11  0  0  0  1 |  j = rec.sport.baseball
  0  0  0  4  9  0  2  5  0 69  2  0  0  1  1  5  0  0  1  1 |  k = rec.sport.hockey
  0  0  0  6 16  1  1  3  1 14  0 41  3  2  2  6  1  0  3  0 |  l = sci.crypt
  0  0  0 12 22  1  1 10  2 16  0  3 17  1  4 11  0  0  0  0 |  m = sci.electronics
  0  2  0  9 20  0  0  6  1 17  0  0  1 23  7 11  0  1  1  1 |  n = sci.med
  0  4  0  1 16  0  0  4  0 15  1  0  0  3 43 10  1  0  1  1 |  o = sci.space
  1  0  0  5 10  0  0  3  0  6  0  0  0  2  1 68  0  0  0  4 |  p = soc.religion.christian
  0  0  1  2 15  1  0 17  0 14  1  2  0  3  1 11 25  0  5  2 |  q = talk.politics.guns
  2  3  0  2 15  0  0  6  0 21  0  2  0  2  2  8  0 35  1  1 |  r = talk.politics.mideast
  0  0  0  5 19  0  1  9  1 23  0  0  0  5  3 13  4  5 10  2 |  s = talk.politics.misc
  6  0  0  7 12  0  0  8  0 13  0  0  0 13  3 31  2  1  2  2 |  t = talk.religion.misc
```

D-42 20-Newsgroups Veri Kümesinde LDA İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

```
  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t   <-- classified as
 10  1  1  9  1  0  0  2  0  0  0  0  1  3  1 29  0  2 30 10 |  a = alt.atheism
  1 32  7 30  4  5  1  2  0  0  0  0  0  1  4  0  0  1 11  1 |  b = comp.graphics
  3  2 50 25  3  2  1  0  3  0  0  0  1  1  3  1  0  0  4  1 |  c = comp.os.ms-windows.misc
  3  2 12 61  1  0  2  1  0  0  0  0  1  3  1  1  2  0 10  0 |  d = comp.sys.ibm.pc.hardware
  1  1  4 33 37  1  1  0  0  0  1  0  2  3  1  0  0  0 15  0 |  e = comp.sys.mac.hardware
  1  3 29 27  0 23  0  0  0  0  2  0  0  1  2  0  0  0 10  2 |  f = comp.windows.x
  1  0  4 40  5  1 31  6  0  2  2  0  1  0  2  0  1  0  4  0 |  g = misc.forsale
  1  0  0 22  1  0  4 61  1  0  0  0  0  1  0  0  2  0  7  0 |  h = rec.autos
  5  0  0 18  0  2  0  6 52  0  1  0  0  3  1  0  1  0 11  0 |  i = rec.motorcycles
  7  0  0 15  1  0  2  0  0 33 28  0  0  1  0  0  0  0 13  0 |  j = rec.sport.baseball
  4  0  0 16  0  0  1  0  0  8 59  0  0  0  0  0  1  0 10  1 |  k = rec.sport.hockey
  4  0  1 19  1  1  0  0  0  0  2 47  1  2  0  2  1  0 18  1 |  l = sci.crypt
  4  2  0 40  4  0  0  5  1  0  4  3 19  1  4  0  0  0 13  0 |  m = sci.electronics
  8  5  0 27  0  1  1  0  1  0  1  0  1 26  1  3  0  0 19  6 |  n = sci.med
  7  2  0 15  2  1  0  0  0  0  0  0  0  3 45  0  1  0 22  2 |  o = sci.space
  6  1  0 25  1  0  0  0  0  0  1  0  0  1  1 59  0  0  3  2 |  p = soc.religion.christian
  6  1  1 13  0  1  1  2  0  1  2  2  0  4  1  2 40  0 23  0 |  q = talk.politics.guns
  6  1  0 20  0  0  0  1  0  0  0  0  0  1  0  2  1 39 26  3 |  r = talk.politics.mideast
 11  0  1 16  0  0  2  1  0  1  1  0  0  5  1  2  7  3 46  3 |  s = talk.politics.misc
  8  1  1 15  0  0  0  1  0  0  1  0  0  6  0 29  4  1 16 17 |  t = talk.religion.misc
```

D-43 ModApte-10 Veri Kümesinde Özellik Çıkarımı Olmadan Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle)

```
   a   b   c   d   e   f   g   h   i   j   <-- classified as
 287 327   9  28   1   7   0   7   9  42 |   a = acq
   0   1   0   0   0   0   0   0   0   0 |   b = corn
   0  46  91   0   1   0   0  22   1   6 |   c = crude
  14 614   9 431   0   1   1   0   8   5 |   d = earn
   0  65   0   0  31   0   0   3   5  33 |   e = grain
   0  36   2   0   0  26  26   0  10   1 |   f = interest
   0  59   0   0   1  20  62   3  18   1 |   g = money-fx
   0   9   9   0   0   0   0  32   5   5 |   h = ship
   0  25   0   0   2   1   5   2  74   3 |   i = trade
   0   2   0   0   1   0   0   0   0   0 |   j = wheat
```

D-44 ModApte-10 Veri Kümesinde Özellik Çıkarımı Olmadan Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
 699    0    4   12    0    0    0    0    2    0 |    a = acq
   0    0    0    0    1    0    0    0    0    0 |    b = corn
  13    0  149    1    0    0    0    3    1    0 |    c = crude
  11    0    3 1066    0    0    0    3    0    0 |    d = earn
   2    0    0    1  119    0    0    8    5    2 |    e = grain
   0    0    0    1    0   63   33    0    4    0 |    f = interest
   7    0    0    0    0   15  136    0    6    0 |    g = money-fx
   5    0   19    1    0    0    0   33    2    0 |    h = ship
   4    0    1    0    5    0    3    0   99    0 |    i = trade
   0    0    0    0    3    0    0    0    0    0 |    j = wheat
```

D-45 ModApte-10 Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En Kötü Karmaşıklık Matrisi (SVM Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
 710    0    0    4    1    0    1    0    1    0 |    a = acq
   0    0    0    0    1    0    0    0    0    0 |    b = corn
   5    0  148    8    1    0    0    1    4    0 |    c = crude
   6    0    0 1070    6    0    0    1    0    0 |    d = earn
   0    0    0    0  130    0    0    0    7    0 |    e = grain
   1    0    0    1    1    0   98    0    0    0 |    f = interest
   0    0    0    1    0    0  163    0    0    0 |    g = money-fx
   3    0   28    4   10    0    1    3   11    0 |    h = ship
   0    0    0    3    1    0   29    0   79    0 |    i = trade
   0    0    0    0    3    0    0    0    0    0 |    j = wheat
```

D-46 ModApte-10 Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En İyi Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
 701    0    5    7    1    0    1    0    2    0 |    a = acq
   0    0    0    0    1    0    0    0    0    0 |    b = corn
   1    0  156    1    1    0    0    8    0    0 |    c = crude
   4    0    1 1070    4    0    0    4    0    0 |    d = earn
   0    0    0    0  132    0    0    3    2    0 |    e = grain
   0    0    0    0    0   73   27    0    1    0 |    f = interest
   0    0    0    1    0    4  157    0    2    0 |    g = money-fx
   0    0    7    0    0    0    0   53    0    0 |    h = ship
   0    0    1    0    1    0    4    0  106    0 |    i = trade
   0    0    0    0    3    0    0    0    0    0 |    j = wheat
```

D-47 ModApte-10 Veri Kümesinde Chi-Kare İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
  64    0    0  653    0    0    0    0    0    0 |    a = acq
   0    0    0    1    0    0    0    0    0    0 |    b = corn
   0    0   24  143    0    0    0    0    0    0 |    c = crude
   0    0    1 1079    0    0    0    3    0    0 |    d = earn
   0    0    0  123   13    0    0    1    0    0 |    e = grain
   0    0    0   78    0   17    6    0    0    0 |    f = interest
   0    0    0  135    0    2   27    0    0    0 |    g = money-fx
   0    0    0   59    0    0    0    1    0    0 |    h = ship
   0    0    0  109    0    0    0    0    3    0 |    i = trade
   0    0    0    3    0    0    0    0    0    0 |    j = wheat
```

D-48 ModApte-10 Veri Kümesinde Chi-Kare İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
 674    0    4   39    0    0    0    0    0    0 |    a = acq
   0    0    0    0    1    0    0    0    0    0 |    b = corn
   4    0  148   10    0    0    0    4    1    0 |    c = crude
   8    0    2 1069    0    1    0    3    0    0 |    d = earn
   0    0    0    3  123    0    0    7    3    1 |    e = grain
   0    0    0    1    0   62   35    0    3    0 |    f = interest
   5    0    0    5    0   15  132    0    7    0 |    g = money-fx
   4    0   24    3    1    0    0   27    1    0 |    h = ship
   3    0    1    3    3    0    2    0  100    0 |    i = trade
   0    0    0    0    3    0    0    0    0    0 |    j = wheat
```

D-49 ModApte-10 Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
 566   38   14   14    0   50    6   10    0   19 |    a = acq
   0    1    0    0    0    0    0    0    0    0 |    b = corn
   2   26  121    1    0    1    1    8    1    6 |    c = crude
  33   12    8 1012    0    4    1    2    0   11 |    d = earn
   0   64    1    0   17    1    1    5    3   45 |    e = grain
   0    4    0    0    0   66   27    0    0    4 |    f = interest
   1   19    0    0    0   24  109    0    5    6 |    g = money-fx
   0   37    3    0    1    1    1    5    0   12 |    h = ship
   0   39    4    0    6    4   24    5   22    8 |    i = trade
   0    0    0    0    1    0    0    0    0    2 |    j = wheat
```

D-50 ModApte-10 Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
 673    0    6   30    0    0    3    0    5    0 |    a = acq
   0    0    0    0    1    0    0    0    0    0 |    b = corn
  10    0  144    1    0    1    0   11    0    0 |    c = crude
  37    0    1 1041    0    0    0    4    0    0 |    d = earn
   3    0    2    1  117    1    0    4    9    0 |    e = grain
   7    0    0    0    0   57   32    0    5    0 |    f = interest
  21    0    0    0    1   20  112    0   10    0 |    g = money-fx
  10    0    5    3    2    0    0   38    2    0 |    h = ship
   4    0    2    0    3    0    5    1   97    0 |    i = trade
   0    0    0    0    3    0    0    0    0    0 |    j = wheat
```

D-51 ModApte-10 Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
  64    0    0  653    0    0    0    0    0    0 |    a = acq
   0    0    0    1    0    0    0    0    0    0 |    b = corn
   0    0   24  143    0    0    0    0    0    0 |    c = crude
   0    0    1 1079    0    0    0    3    0    0 |    d = earn
   0    0    0  123   13    0    0    1    0    0 |    e = grain
   0    0    0   78    0   17    6    0    0    0 |    f = interest
   0    0    0  135    0    2   27    0    0    0 |    g = money-fx
   0    0    0   59    0    0    0    1    0    0 |    h = ship
   0    0    0  109    0    0    0    0    3    0 |    i = trade
   0    0    0    3    0    0    0    0    0    0 |    j = wheat
```

D-52 ModApte-10 Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
 674    0    4   39    0    0    0    0    0    0 |    a = acq
   0    0    0    0    1    0    0    0    0    0 |    b = corn
   4    0  148   10    0    0    0    4    1    0 |    c = crude
   8    0    2 1069    0    1    0    3    0    0 |    d = earn
   0    0    0    3  123    0    0    7    3    1 |    e = grain
   0    0    0    1    0   62   35    0    3    0 |    f = interest
   5    0    0    5    0   15  132    0    7    0 |    g = money-fx
   4    0   24    3    1    0    0   27    1    0 |    h = ship
   3    0    1    3    3    0    2    0  100    0 |    i = trade
   0    0    0    0    3    0    0    0    0    0 |    j = wheat
```

D-53 ModApte-10 Veri Kümesinde PCA İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
 123    0    0  552    0    1    5   36    0    0 |    a = acq
   0    0    0    0    1    0    0    0    0    0 |    b = corn
   0    0   32  125    1    2    1    6    0    0 |    c = crude
   2    0    0 1076    0    0    0    5    0    0 |    d = earn
   0    0    0   95   34    3    0    5    0    0 |    e = grain
   1    0    0   43    0   45    9    3    0    0 |    f = interest
   1    0    0   96    0   15   45    7    0    0 |    g = money-fx
   0    0    0   44    1    1    0   14    0    0 |    h = ship
   0    0    0  103    0    1    3    5    0    0 |    i = trade
   0    0    0    3    0    0    0    0    0    0 |    j = wheat
```

D-54 ModApte-10 Veri Kümesinde PCA İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
 688    0    5   21    0    0    1    0    2    0 |    a = acq
   0    0    0    0    1    0    0    0    0    0 |    b = corn
   6    0  152    3    2    1    0    0    3    0 |    c = crude
   9    0    1 1070    0    0    0    3    0    0 |    d = earn
   2    0    0    1  112    3    3    5    9    2 |    e = grain
   0    0    0    1    1   64   31    0    4    0 |    f = interest
   6    1    0    1    1   17  131    0    7    0 |    g = money-fx
   2    0   21    0    1    0    2   32    2    0 |    h = ship
   1    0    3    1    8    0    2    1   96    0 |    i = trade
   0    0    0    0    3    0    0    0    0    0 |    j = wheat
```

D-55 ModApte-10 Veri Kümesinde LSA İle Elde Edilen En Kötü Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
  72    0    0  176    0    1    0  468    0    0 |    a = acq
   0    0    0    0    1    0    0    0    0    0 |    b = corn
   0    0   31   31    1    2    1  101    0    0 |    c = crude
   1    0    0 1038    0    0    0   44    0    0 |    d = earn
   0    0    0   33   48    0    0   56    0    0 |    e = grain
   0    0    0    5    0   32   22   42    0    0 |    f = interest
   0    0    0   22    0    5   61   76    0    0 |    g = money-fx
   1    0    0   12    1    0    0   46    0    0 |    h = ship
   5    0    0   20    0    0    4   82    1    0 |    i = trade
   0    0    0    1    1    0    0    1    0    0 |    j = wheat
```

D-56 ModApte-10 Veri Kümesinde LSA İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
 664    0   11   34    0    2    1    0    5    0 |    a = acq
   0    0    0    0    1    0    0    0    0    0 |    b = corn
  25    0  112   17    0    2    1    4    5    1 |    c = crude
  13    0    4 1063    0    2    0    0    1    0 |    d = earn
   4    0    0    9  115    0    1    3    3    2 |    e = grain
   9    0    0    4    0   51   28    0    9    0 |    f = interest
  23    0    3    8    2    8  100    0   20    0 |    g = money-fx
  23    0    6    8    2    1    1   19    0    0 |    h = ship
  10    0    1    5   11    1    8    0   76    0 |    i = trade
   0    0    0    0    3    0    0    0    0    0 |    j = wheat
```

D-57 ModApte-10 Veri Kümesinde LDA İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle)

```
  a   b   c   d   e   f   g   h   i   j   <-- classified as
664   3  19   1   0  25   2   0   2   1 |   a = acq
  0   0   0   0   0   0   1   0   0   0 |   b = corn
 34   6 115   0   0   3   0   7   2   0 |   c = crude
113   2   5 939   1  16   6   0   1   0 |   d = earn
 41  18   0   8   3   7   3   4   1  52 |   e = grain
  9   0   0   0   0  78  14   0   0   0 |   f = interest
 44   3   0   0   0  47  69   0   1   0 |   g = money-fx
 15  21   8   0   0   2   1  12   0   1 |   h = ship
 19  20   4   0   0  30  31   0   8   0 |   i = trade
  0   0   0   0   0   0   0   0   0   3 |   j = wheat
```

D-58 ModApte-10 Veri Kümesinde LDA İle Elde Edilen En İyi Karmaşıklık Matrisleri (10 En Yakın Komşu ve SVM Algoritmaları İle)

```
  a   b   c   d   e   f   g   h   i   j   <-- classified as
658   0  28   6   5   1   5   0  14   0 |   a = acq
  0   0   0   0   0   0   0   0   1   0 |   b = corn
 19   0 135   3   0   1   0   5   4   0 |   c = crude
112   0   9 946   1   2  10   0   3   0 |   d = earn
 42   0   2   5  76   2   1   4   5   0 |   e = grain
  9   0   0   6   0  59  24   0   3   0 |   f = interest
 47   0   0  12   0  17  84   0   4   0 |   g = money-fx
 15   0   9   2   1   0   0  30   3   0 |   h = ship
 23   0   6   2   2   1   5   0  73   0 |   i = trade
  0   0   0   0   3   0   0   0   0   0 |   j = wheat

  a   b   c   d   e   f   g   h   i   j   <-- classified as
658   0  28   7   1   0   4   0  19   0 |   a = acq
  0   0   0   0   0   0   0   0   1   0 |   b = corn
 15   0 143   1   0   1   0   3   4   0 |   c = crude
112   0   8 945   0   1  11   1   5   0 |   d = earn
 42   0   4   4  70   1   0   6  10   0 |   e = grain
 10   0   0   5   0  56  25   0   5   0 |   f = interest
 46   0   1   8   0  13  82   0  14   0 |   g = money-fx
 17   0  11   0   0   0   0  28   4   0 |   h = ship
 20   0   6   0   2   1   5   0  78   0 |   i = trade
  0   0   0   0   2   0   0   0   0   1 |   j = wheat
```

D-59 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En Kötü Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	a = alum
0	0	0	0	0	0	0	14	0	0	0	0	0	32	0	0	0	0	0	0	0	b = bop
0	0	53	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	c = cocoa
0	0	0	123	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d = coffee
0	0	0	1	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	e = copper
0	0	0	0	0	65	0	7	0	0	0	0	0	1	0	0	0	0	2	0	0	f = cpi
0	0	0	0	0	0	11	12	1	0	0	0	0	9	0	0	0	0	1	0	0	g = dlr
0	0	0	0	0	0	0	115	0	0	0	0	0	0	0	0	0	0	0	0	0	h = gnp
0	0	0	0	0	0	0	0	111	0	0	0	0	0	0	0	0	0	0	0	0	i = gold
0	0	0	0	0	24	0	14	0	5	0	0	0	2	0	0	0	0	0	0	0	j = ipi
0	0	0	0	0	0	0	0	0	0	48	0	0	0	0	0	0	0	3	0	0	k = iron-steel
0	0	0	0	0	0	0	10	0	0	0	36	0	0	0	0	0	0	0	0	1	l = jobs
0	0	0	0	0	0	0	0	0	0	0	0	53	0	0	0	0	0	0	1	1	m = livestock
0	0	0	0	0	0	0	0	0	0	0	0	0	110	0	0	0	0	0	0	0	n = money-supply
0	0	0	0	0	0	0	0	0	0	0	0	0	0	47	0	0	0	1	0	0	o = nat-gas
0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	44	0	0	4	10	18	p = oilseed
0	0	0	0	0	0	0	1	0	0	0	0	0	49	0	0	0	0	0	0	0	q = reserves
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38	0	2	0	r = rubber
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	192	1	1	s = ship
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	143	0	t = sugar
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	2	88	u = veg-oil

D-60 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En İyi Karmaşıklık Matrisleri (10 En Yakın Komşu ve SVM Algoritmaları İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	a = alum
0	45	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	b = bop
0	0	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	c = cocoa
0	0	0	121	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	d = coffee
0	0	0	1	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	e = copper
0	0	0	0	0	73	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	f = cpi
0	0	0	0	0	0	32	1	0	0	0	0	0	0	0	0	1	0	0	0	0	g = dlr
0	0	0	0	0	0	0	113	0	2	0	0	0	0	0	0	0	0	0	0	0	h = gnp
0	0	0	0	0	0	0	0	110	0	0	0	0	0	0	0	0	0	0	1	0	i = gold
0	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0	0	0	0	0	j = ipi
0	0	0	0	0	0	0	0	0	0	51	0	0	0	0	0	0	0	0	0	0	k = iron-steel
0	0	0	0	0	0	0	0	0	0	0	47	0	0	0	0	0	0	0	0	0	l = jobs
0	0	0	0	0	0	0	0	0	0	0	0	53	0	0	0	0	0	0	1	1	m = livestock
0	1	0	0	0	0	0	0	0	0	0	0	0	109	0	0	0	0	0	0	0	n = money-supply
0	0	0	0	0	0	0	0	0	0	0	0	0	0	47	0	0	0	1	0	0	o = nat-gas
0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	63	0	0	0	2	11	p = oilseed
0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	46	0	0	0	0	q = reserves
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	1	0	r = rubber
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	190	2	1	s = ship
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	143	0	t = sugar
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	88	u = veg-oil

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	a = alum
0	44	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	b = bop
0	0	55	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	c = cocoa
0	0	0	122	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	d = coffee
0	0	0	3	54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	e = copper
0	0	0	0	0	74	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	f = cpi
0	0	0	0	0	0	33	0	0	0	0	0	0	0	0	0	1	0	0	0	0	g = dlr
0	0	0	0	0	0	0	113	0	2	0	0	0	0	0	0	0	0	0	0	0	h = gnp
0	0	0	0	0	0	0	0	108	0	0	0	0	2	0	0	0	0	0	1	0	i = gold
0	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0	0	0	0	0	j = ipi
0	0	0	0	0	0	0	0	0	0	51	0	0	0	0	0	0	0	0	0	0	k = iron-steel
0	0	0	0	0	0	0	0	0	0	0	47	0	0	0	0	0	0	0	0	0	l = jobs
0	0	0	0	0	0	0	0	0	0	0	0	53	0	0	0	0	0	0	1	1	m = livestock
0	1	0	0	0	0	0	0	0	0	0	0	0	109	0	0	0	0	0	0	0	n = money-supply
0	0	0	0	0	0	0	0	0	0	0	0	0	0	47	0	0	0	1	0	0	o = nat-gas
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	66	0	0	0	0	11	p = oilseed
0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	46	0	0	0	0	q = reserves
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	38	0	1	0	r = rubber
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	189	2	1	s = ship
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	143	0	t = sugar
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	88	u = veg-oil

D-61 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde Chi-Kare İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
31	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	14	0	0	0	a = alum
0	39	0	0	0	0	3	2	0	0	0	0	0	1	0	0	1	0	0	0	0	b = bop
0	0	56	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = cocoa
1	0	17	97	1	0	0	1	0	0	0	0	0	0	0	0	0	4	0	0	3	d = coffee
0	0	0	0	50	0	0	0	5	0	0	0	0	0	0	0	0	1	0	1	0	e = copper
0	1	0	1	0	43	1	6	0	0	0	10	0	5	2	3	0	3	0	0	0	f = cpi
2	1	0	0	0	0	28	1	0	0	1	0	0	0	0	0	1	0	0	0	0	g = dlr
2	0	0	0	0	2	1	72	0	10	2	1	3	9	0	2	4	3	4	0	0	h = gnp
0	0	0	0	0	0	1	0	104	0	0	0	0	0	1	0	4	1	0	0	0	i = gold
0	1	0	0	0	0	2	1	0	0	3	9	0	2	3	1	0	22	1	0	0	j = ipi
1	0	0	0	0	0	0	0	1	1	39	0	0	0	0	0	0	8	1	0	0	k = iron-steel
0	0	0	0	0	0	0	2	0	1	2	39	0	0	0	0	0	3	0	0	0	l = jobs
2	0	0	0	0	0	0	0	0	0	0	0	3	0	0	2	0	47	0	1	0	m = livestock
1	1	0	0	0	1	1	1	0	0	1	1	0	85	2	1	4	9	1	0	1	n = money-supply
0	1	0	0	0	0	0	0	0	1	0	0	0	0	43	0	0	3	0	0	0	o = nat-gas
1	1	0	0	0	0	0	1	0	1	0	0	2	0	2	12	0	55	0	0	3	p = oilseed
0	0	0	0	0	0	0	2	1	0	1	0	0	14	1	0	31	0	0	0	0	q = reserves
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	r = rubber
3	0	0	1	0	0	2	1	0	3	3	0	5	3	2	40	0	122	3	1	5	s = ship
0	0	2	3	0	0	0	0	0	0	0	0	0	0	1	0	0	3	0	135	0	t = sugar
0	2	0	1	0	0	0	0	0	1	1	0	0	0	2	50	0	7	0	0	29	u = veg-oil

D-62 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde Chi-Kare İle Elde Edilen En İyi Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
30	0	0	0	0	0	1	0	0	0	3	0	1	0	0	0	0	13	0	0	0	a = alum
0	37	0	0	0	1	1	4	0	0	0	0	0	1	0	0	1	0	1	0	0	b = bop
0	0	51	3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2	1	0	c = cocoa
0	0	1	117	0	0	0	1	0	0	0	0	0	0	0	0	0	0	5	0	0	d = coffee
0	0	0	0	56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	e = copper
0	1	0	1	0	53	0	4	0	8	0	0	0	4	2	0	0	0	2	0	0	f = cpi
0	1	0	0	0	0	31	2	0	0	0	0	0	0	0	0	0	0	0	0	0	g = dlr
0	1	0	1	0	5	0	92	0	1	0	2	1	4	0	0	1	1	6	0	0	h = gnp
0	0	0	0	0	0	0	0	105	0	0	0	0	0	1	0	0	0	4	0	1	i = gold
0	0	0	0	0	6	1	4	0	21	0	1	0	1	1	0	0	0	9	0	1	j = ipi
0	0	0	0	0	0	0	0	1	0	40	0	1	0	0	0	0	0	9	0	0	k = iron-steel
0	0	0	0	0	0	0	1	0	2	0	40	1	0	0	0	0	0	3	0	0	l = jobs
0	0	0	0	0	0	0	1	0	0	1	0	4	0	0	2	0	0	46	1	0	m = livestock
0	0	0	0	0	4	1	2	0	7	0	0	0	89	0	0	1	0	6	0	0	n = money-supply
0	0	0	0	0	0	0	0	0	1	0	0	1	0	45	0	0	0	1	0	0	o = nat-gas
1	0	0	0	0	0	0	0	0	2	0	0	2	2	0	0	0	0	56	0	15	p = oilseed
0	0	0	0	0	1	1	1	0	0	0	0	0	14	0	0	33	0	0	0	0	q = reserves
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38	2	0	0	r = rubber
2	0	0	2	0	0	0	7	0	3	3	0	7	2	0	0	1	0	158	1	8	s = ship
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2	141	0	t = sugar
0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	4	0	1	23	0	63	u = veg-oil

D-63 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
7	0	0	0	0	0	0	0	0	0	2	4	7	0	0	0	0	26	2	0	0	a = alum
0	28	0	0	0	0	8	2	1	1	0	0	0	5	0	0	1	0	0	0	0	b = bop
0	0	51	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = cocoa
0	0	37	79	0	0	0	1	0	0	0	1	1	0	1	0	0	2	1	0	1	d = coffee
0	0	0	0	51	0	0	0	4	0	0	0	0	0	0	0	0	1	0	1	0	e = copper
0	0	0	2	0	46	0	7	0	1	0	2	0	6	4	0	0	7	0	0	0	f = cpi
0	1	0	0	0	1	10	0	0	0	1	3	2	14	0	0	1	1	0	0	0	g = dlr
7	12	1	0	0	6	5	36	0	0	5	7	3	14	0	0	4	14	0	0	1	h = gnp
0	0	0	0	0	0	0	0	79	0	0	0	0	0	2	0	29	1	0	0	0	i = gold
3	0	0	0	0	0	0	3	0	3	1	2	3	0	2	2	0	26	0	0	0	j = ipi
4	0	0	0	0	0	0	0	1	0	35	1	1	0	0	0	0	7	2	0	0	k = iron-steel
0	0	0	0	0	0	0	2	0	1	0	42	1	1	0	0	0	0	0	0	0	l = jobs
1	0	0	0	0	0	0	0	0	0	1	4	35	0	0	3	0	10	0	1	0	m = livestock
2	0	0	0	0	1	3	1	0	0	0	0	0	100	0	0	3	0	0	0	0	n = money-supply
1	0	0	0	0	0	0	0	0	0	0	1	0	0	45	0	0	1	0	0	0	o = nat-gas
2	0	0	0	0	1	0	1	0	0	1	5	17	0	2	6	0	31	0	2	10	p = oilseed
0	0	0	0	0	0	1	1	1	0	0	0	0	21	0	0	26	0	0	0	0	q = reserves
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	1	r = rubber
26	0	0	0	0	0	2	0	1	0	3	8	21	1	66	2	0	26	33	4	1	s = ship
1	0	0	4	0	0	0	0	0	0	0	0	3	0	1	1	0	0	0	134	0	t = sugar
2	0	0	1	0	0	0	0	0	0	0	0	4	0	12	24	0	5	1	0	44	u = veg-oil

D-64 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
0	0	0	0	0	0	2	1	0	1	1	0	0	2	0	12	0	0	29	0	0	a = alum
0	35	0	0	0	0	2	4	0	0	0	0	0	5	0	0	0	0	0	0	0	b = bop
0	0	53	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	c = cocoa
0	0	1	116	0	1	0	0	0	0	0	0	0	2	0	0	0	0	4	0	0	d = coffee
0	0	0	0	55	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	e = copper
0	0	0	0	0	65	0	7	0	0	0	0	0	0	1	1	0	0	1	0	0	f = cpi
0	0	0	0	0	0	11	4	0	0	0	0	0	4	0	0	0	0	15	0	0	g = dlr
2	0	0	0	0	2	1	97	0	6	0	0	0	2	0	0	1	0	4	0	0	h = gnp
0	0	0	0	0	0	0	0	109	0	0	0	0	0	0	1	0	0	1	0	0	i = gold
0	0	0	0	0	1	0	13	0	29	0	0	1	0	0	0	0	0	1	0	0	j = ipi
4	0	0	0	0	0	0	1	0	0	37	0	0	0	0	3	0	0	5	0	1	k = iron-steel
0	0	0	0	0	0	0	2	0	0	0	42	0	0	0	0	0	0	3	0	0	l = jobs
1	0	0	0	0	0	0	0	0	4	0	0	30	0	0	3	0	0	15	1	1	m = livestock
0	0	0	0	0	1	3	7	0	0	0	0	0	98	0	0	1	0	0	0	0	n = money-supply
0	0	0	0	0	0	0	1	0	0	0	0	0	0	44	2	0	0	1	0	0	o = nat-gas
1	0	0	0	0	0	0	2	0	0	0	0	12	1	0	24	0	0	21	0	17	p = oilseed
0	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	29	0	0	0	0	q = reserves
0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	37	1	0	0	r = rubber
0	0	0	2	0	0	1	4	0	0	3	0	4	1	0	5	0	0	169	1	4	s = ship
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	2	140	0	t = sugar
0	0	0	1	0	0	1	0	0	0	0	0	4	0	0	6	0	0	10	0	71	u = veg-oil

D-65 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
3	0	1	0	0	0	2	0	1	0	1	0	28	0	0	0	0	11	1	0	0	a = alum
0	39	0	0	0	0	3	1	1	0	0	0	0	2	0	0	0	0	0	0	0	b = bop
0	0	52	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = cocoa
0	0	63	52	1	1	0	0	0	0	0	0	4	0	0	0	0	1	1	0	1	d = coffee
0	0	0	0	51	0	0	0	5	0	0	0	1	0	0	0	0	0	0	0	0	e = copper
0	1	0	1	0	60	0	5	0	1	1	2	0	1	1	0	0	2	0	0	0	f = cpi
0	0	0	0	0	0	20	2	0	0	1	0	6	1	1	0	0	3	0	0	0	g = dlr
0	2	0	0	0	2	3	80	0	10	0	4	1	10	0	0	3	0	0	0	0	h = gnp
1	0	0	0	0	0	0	0	99	1	1	0	1	0	0	0	8	0	0	0	0	i = gold
1	0	0	0	0	0	0	5	0	18	5	1	3	0	0	1	0	9	2	0	0	j = ipi
3	0	0	0	0	0	0	0	1	1	6	1	21	0	2	1	0	15	0	0	0	k = iron-steel
0	0	0	0	0	0	0	4	0	0	1	40	2	0	0	0	0	0	0	0	0	l = jobs
0	0	0	0	0	0	0	0	0	0	3	1	44	0	0	1	0	4	0	1	1	m = livestock
1	0	0	0	0	1	3	3	0	0	0	0	0	99	0	0	3	0	0	0	0	n = money-supply
1	0	0	0	0	0	0	0	1	1	4	1	21	0	11	1	0	5	1	0	1	o = nat-gas
0	1	0	0	0	0	0	0	0	0	7	2	24	0	2	4	1	25	0	0	12	p = oilseed
0	0	0	0	0	0	2	3	0	0	0	0	0	19	1	0	25	0	0	0	0	q = reserves
3	0	1	0	0	0	0	0	0	0	0	0	20	0	1	0	0	15	0	0	0	r = rubber
16	1	0	0	0	0	3	1	0	0	10	0	54	0	51	4	0	4	48	1	1	s = ship
0	0	1	7	0	0	0	0	0	0	1	0	3	0	0	0	0	1	1	130	0	t = sugar
1	0	0	1	0	0	0	0	0	1	1	0	6	0	18	7	0	2	0	0	56	u = veg-oil

D-66 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En İyi Karmaşıklık Matrisi (Rasgele Orman Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
12	0	0	0	1	0	1	0	0	0	5	0	3	2	1	9	0	4	10	0	0	a = alum
0	35	0	0	0	0	1	6	0	0	1	0	0	2	0	0	1	0	0	0	0	b = bop
0	0	54	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	c = cocoa
0	0	2	116	0	0	0	1	0	0	0	0	0	0	0	0	0	0	5	0	0	d = coffee
0	0	0	0	54	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	e = copper
0	0	0	0	0	64	0	4	0	2	0	0	1	0	0	0	0	1	2	1	0	f = cpi
1	1	0	0	0	0	21	2	0	0	0	0	1	0	3	0	0	0	5	0	0	g = dlr
1	0	0	1	0	1	1	101	0	1	0	1	1	6	0	0	0	0	1	0	0	h = gnp
0	0	0	0	0	0	0	0	109	0	0	0	0	0	0	0	0	1	1	0	0	i = gold
0	1	0	0	0	2	0	4	0	32	2	0	0	0	2	0	0	0	2	0	0	j = ipi
4	1	0	0	0	0	0	0	1	1	19	0	6	1	2	3	0	1	8	1	3	k = iron-steel
0	0	0	0	0	1	1	3	0	0	0	37	0	0	4	0	0	0	1	0	0	l = jobs
3	0	0	0	0	0	1	0	0	0	4	0	12	0	4	1	0	3	23	2	2	m = livestock
0	0	0	0	0	1	4	10	0	0	0	1	0	88	0	0	5	0	1	0	0	n = money-supply
1	0	0	0	0	0	3	2	0	2	2	2	5	0	8	3	0	4	13	0	3	o = nat-gas
6	1	0	0	0	0	1	0	0	0	4	0	2	0	6	26	0	1	14	0	17	p = oilseed
0	0	0	0	0	0	0	1	1	0	0	0	0	11	0	0	37	0	0	0	0	q = reserves
6	0	0	0	0	1	0	0	0	0	5	0	6	0	1	3	0	10	6	0	2	r = rubber
7	0	1	2	0	1	1	1	0	0	1	1	0	1	5	0	0	4	163	1	5	s = ship
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	141	0	t = sugar
2	0	0	1	0	1	0	0	0	0	3	0	0	0	3	8	0	0	9	0	66	u = veg-oil

D-67 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde PCA İle Elde Edilen En Kötü Karmaşıklık Matrisi (RIPPER Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
20	1	1	0	3	0	0	0	0	0	2	0	0	0	0	0	0	2	18	0	1	a = alum
0	32	0	0	0	0	1	4	0	0	0	1	0	3	0	0	2	0	3	0	0	b = bop
0	0	34	1	0	0	0	1	2	0	0	0	0	0	0	0	0	0	15	5	0	c = cocoa
1	0	2	81	0	0	0	2	2	0	0	1	3	0	0	1	0	1	25	5	0	d = coffee
5	1	0	0	31	0	0	0	2	0	2	1	0	0	0	0	0	2	13	0	0	e = copper
0	0	0	1	0	52	0	7	1	2	0	0	0	2	0	0	0	1	9	0	0	f = cpi
1	1	0	0	0	0	22	3	0	0	0	0	0	0	0	0	0	1	6	0	0	g = dlr
0	3	0	0	0	3	1	82	1	5	1	0	0	1	0	0	0	0	16	0	2	h = gnp
0	0	0	1	1	0	2	1	78	2	1	0	0	2	0	0	0	0	23	0	0	i = gold
0	1	0	0	0	1	0	2	0	31	0	1	0	1	0	0	1	0	6	1	0	j = ipi
4	1	0	0	1	1	0	0	2	0	18	0	0	0	0	0	0	0	24	0	0	k = iron-steel
0	0	0	0	0	1	0	3	0	1	0	35	0	0	0	0	1	1	5	0	0	l = jobs
3	0	0	2	1	0	1	0	0	0	2	0	28	1	0	0	0	0	15	0	2	m = livestock
0	0	0	0	1	0	3	4	2	0	0	0	0	85	1	0	7	1	6	0	0	n = money-supply
0	0	0	1	0	0	0	1	0	0	0	0	0	0	40	0	0	0	5	1	0	o = nat-gas
0	1	0	0	1	0	0	0	0	0	2	0	5	0	0	21	1	0	20	6	21	p = oilseed
0	4	0	0	0	0	2	2	0	0	0	0	0	9	0	0	23	0	9	0	1	q = reserves
2	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	25	10	0	1	r = rubber
4	0	3	1	4	0	0	2	2	0	4	3	1	0	1	0	3	1	156	6	3	s = ship
0	0	2	3	0	0	0	0	0	1	0	0	4	0	0	3	0	1	41	87	2	t = sugar
0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	5	0	0	19	4	63	u = veg-oil

D-68 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde PCA İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
41	0	0	0	1	0	0	0	2	0	3	0	0	0	0	0	0	0	1	0	0	a = alum
0	38	0	0	0	0	1	3	1	0	0	0	0	1	0	0	1	0	1	0	0	b = bop
1	0	48	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	6	0	c = cocoa
0	0	4	107	1	0	0	1	1	0	1	0	0	0	0	0	0	0	6	3	0	d = coffee
5	0	0	1	39	0	0	0	4	0	1	0	0	0	0	0	0	3	4	0	0	e = copper
0	0	0	0	0	65	0	7	0	0	0	0	0	2	1	0	0	0	0	0	0	f = cpi
0	1	0	2	0	0	28	1	0	0	0	0	0	1	0	0	0	0	1	0	0	g = dlr
0	1	0	1	0	1	1	105	0	3	0	0	0	1	0	0	0	0	2	0	0	h = gnp
0	0	0	0	0	0	0	0	104	0	0	0	1	0	0	0	0	0	5	1	0	i = gold
0	0	0	0	0	1	0	4	0	36	1	1	0	0	0	1	0	0	1	0	0	j = ipi
4	0	0	1	2	0	0	1	1	0	33	0	0	0	0	2	0	0	5	2	0	k = iron-steel
0	0	0	0	0	0	0	2	0	0	1	42	0	1	0	0	0	0	0	1	0	l = jobs
0	0	0	1	1	0	0	0	0	0	1	0	42	0	0	4	0	0	2	3	1	m = livestock
0	1	0	0	0	2	2	2	0	0	0	0	0	102	0	0	1	0	0	0	0	n = money-supply
1	0	0	0	0	0	0	0	0	0	0	0	0	0	45	0	0	0	2	0	0	o = nat-gas
0	0	0	1	0	0	0	1	1	0	0	0	6	1	0	33	0	0	10	4	21	p = oilseed
0	1	0	0	0	0	2	2	0	0	0	0	0	11	0	0	34	0	0	0	0	q = reserves
0	0	0	0	1	1	0	1	0	0	2	0	0	0	0	0	0	33	2	0	0	r = rubber
3	1	1	3	1	0	0	0	1	1	5	0	1	0	1	1	0	0	173	1	1	s = ship
0	0	1	6	1	0	0	0	0	0	0	0	5	0	0	0	0	1	6	123	1	t = sugar
0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	4	0	1	3	4	78	u = veg-oil

D-69 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde LSA İle Elde Edilen En Kötü Karmaşıklık Matrisi (RIPPER Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
19	0	0	0	5	0	0	0	0	0	10	0	0	0	0	1	0	0	13	0	0	a = alum
0	31	0	0	0	0	1	3	0	2	0	1	0	4	0	0	2	0	1	0	1	b = bop
0	0	44	2	0	0	0	0	0	0	0	0	0	0	0	0	0	3	8	1	0	c = cocoa
1	2	1	99	1	1	1	0	2	0	0	0	0	0	0	2	2	0	10	1	1	d = coffee
6	0	0	0	38	0	0	0	1	1	2	0	0	0	0	0	0	0	9	0	0	e = copper
2	0	0	0	0	47	0	9	0	6	0	1	0	1	1	0	0	0	6	2	0	f = cpi
3	0	0	0	0	0	23	2	1	0	0	0	0	2	0	0	0	0	3	0	0	g = dlr
1	3	0	0	0	8	1	82	0	6	0	1	1	4	0	0	1	0	7	0	0	h = gnp
0	0	0	1	0	0	3	0	84	1	1	0	2	0	0	0	0	0	18	1	0	i = gold
1	0	0	1	0	2	0	4	0	24	0	0	1	3	0	0	0	1	8	0	0	j = ipi
5	0	0	0	0	0	0	0	2	0	22	0	0	0	0	1	0	1	17	1	2	k = iron-steel
0	0	0	0	0	0	0	2	0	1	0	41	0	0	0	0	0	0	3	0	0	l = jobs
1	0	0	1	0	0	0	0	0	0	0	0	30	0	0	3	0	0	19	1	0	m = livestock
1	0	0	0	0	0	2	7	0	0	2	0	0	88	0	1	3	0	6	0	0	n = money-supply
0	0	0	0	0	0	0	0	1	0	0	0	0	0	40	0	0	1	6	0	0	o = nat-gas
0	0	2	1	0	0	0	0	0	0	0	0	6	1	0	31	0	0	19	2	16	p = oilseed
0	1	0	0	0	0	3	1	0	0	0	0	0	7	0	0	33	0	5	0	0	q = reserves
2	0	4	0	0	0	0	0	0	0	1	0	1	0	0	0	0	22	10	0	0	r = rubber
4	0	1	0	0	2	0	0	6	1	1	1	2	0	0	3	0	0	168	4	1	s = ship
0	0	2	2	0	0	0	0	0	1	1	1	1	0	0	2	0	0	9	125	0	t = sugar
0	0	1	0	0	0	0	0	0	0	0	0	3	0	0	13	0	0	11	0	65	u = veg-oil

D-70 21 Özelliğe İndirgenmiş Reuters Veri Kümesinde LSA İle Elde Edilen En İyi Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	<-- classified as
36	0	0	0	5	0	0	0	0	0	5	0	0	0	0	0	0	0	2	0	0	a = alum
0	37	0	0	0	0	3	4	0	0	0	0	0	0	0	0	2	0	0	0	0	b = bop
3	0	45	3	0	0	0	0	0	0	0	0	0	0	0	0	0	5	1	1	0	c = cocoa
0	0	1	116	1	0	0	1	0	0	0	0	0	0	0	1	0	0	4	0	0	d = coffee
7	0	0	0	45	0	0	0	2	0	1	0	1	0	0	0	0	0	1	0	0	e = copper
0	0	0	0	0	69	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	f = cpi
0	1	0	0	0	0	31	2	0	0	0	0	0	0	0	0	0	0	0	0	0	g = dlr
1	1	0	0	0	1	1	103	0	3	1	0	0	1	0	0	2	0	0	0	1	h = gnp
1	0	0	0	0	0	0	0	107	0	0	0	0	0	0	0	0	0	3	0	0	i = gold
0	0	0	0	0	1	0	1	0	42	0	0	1	0	0	0	0	0	0	0	0	j = ipi
9	0	0	0	0	0	0	0	1	0	37	0	0	0	0	0	0	0	4	0	0	k = iron-steel
0	0	0	0	0	0	0	2	0	0	1	43	0	0	0	0	0	0	0	0	1	l = jobs
0	0	0	0	0	0	0	0	0	1	2	0	48	0	0	1	0	0	1	0	2	m = livestock
0	0	0	0	0	0	1	4	0	0	0	0	1	99	0	0	5	0	0	0	0	n = money-supply
0	0	0	0	0	0	0	0	0	0	0	0	0	0	44	0	0	0	4	0	0	o = nat-gas
0	0	0	1	0	0	0	0	0	0	0	0	6	1	0	50	0	2	1	1	16	p = oilseed
0	0	0	0	0	0	2	2	0	0	0	0	0	1	0	0	45	0	0	0	0	q = reserves
4	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	31	3	0	0	r = rubber
6	0	0	0	0	0	0	0	0	2	2	0	5	0	0	3	0	0	174	2	0	s = ship
1	0	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0	1	2	137	0	t = sugar
0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	7	0	0	1	1	79	u = veg-oil

D-71 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En Kötü Karmaşıklık Matrisi (C4.5 Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
86	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	3	0	8	a = alt.atheism
0	85	3	7	2	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	b = comp.graphics
0	6	81	3	0	5	1	0	1	1	0	0	1	0	0	1	0	0	0	0	c = comp.os.ms-windows.misc
0	5	4	82	2	2	1	0	0	0	0	0	4	0	0	0	0	0	0	0	d = comp.sys.ibm.pc.hardware
0	0	1	7	87	1	1	1	0	0	0	0	0	0	0	1	1	0	0	0	e = comp.sys.mac.hardware
0	3	3	1	0	89	1	0	1	0	0	0	0	1	0	1	0	0	0	0	f = comp.windows.x
1	0	1	4	3	3	78	2	2	1	0	0	1	1	1	2	0	0	0	0	g = misc.forsale
1	0	0	0	0	0	1	97	0	0	0	0	0	0	0	1	0	0	0	0	h = rec.autos
0	0	0	0	0	2	1	1	94	1	0	1	0	0	0	0	0	0	0	0	i = rec.motorcycles
0	1	1	0	0	1	1	0	3	85	6	1	0	0	0	1	0	0	0	0	j = rec.sport.baseball
1	0	1	0	1	0	2	1	0	4	85	0	0	1	0	1	0	1	2	0	k = rec.sport.hockey
0	0	0	0	0	0	3	0	0	3	0	90	3	0	0	0	1	0	0	0	l = sci.crypt
0	2	1	1	0	0	2	2	1	1	0	4	82	1	0	2	1	0	0	0	m = sci.electronics
0	0	1	0	0	0	1	0	1	0	1	1	1	91	1	1	0	1	0	0	n = sci.med
0	0	0	0	0	0	0	0	1	0	0	0	0	0	95	2	2	0	0	0	o = sci.space
1	1	0	0	0	2	1	1	0	1	0	0	0	2	0	87	1	2	0	1	p = soc.religion.christian
1	0	0	0	1	2	0	0	0	0	0	2	0	0	2	1	88	1	1	1	q = talk.politics.guns
0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	3	1	92	2	0	r = talk.politics.mideast
0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	92	4	s = talk.politics.misc
12	0	0	0	1	0	0	1	0	1	1	0	0	0	0	3	2	0	7	72	t = talk.religion.misc

D-72 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
97	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	1	0	a = alt.atheism
0	96	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b = comp.graphics
0	2	91	3	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = comp.os.ms-windows.misc
0	2	0	94	3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	d = comp.sys.ibm.pc.hardware
0	0	0	0	97	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	e = comp.sys.mac.hardware
0	0	1	0	0	96	1	0	1	1	0	0	0	0	0	0	0	0	0	0	f = comp.windows.x
0	0	0	0	0	0	99	0	1	0	0	0	0	0	0	0	0	0	0	0	g = misc.forsale
0	0	0	0	0	0	0	98	1	0	0	0	0	0	0	0	0	0	0	1	h = rec.autos
0	0	0	0	0	0	1	0	97	0	1	1	0	0	0	0	0	0	0	0	i = rec.motorcycles
0	0	0	0	0	0	0	0	1	92	5	2	0	0	0	0	0	0	0	0	j = rec.sport.baseball
0	0	0	0	1	1	0	0	0	0	98	0	0	0	0	0	0	0	0	0	k = rec.sport.hockey
0	0	0	0	0	0	0	0	0	0	0	96	4	0	0	0	0	0	0	0	l = sci.crypt
0	1	0	1	1	0	0	0	0	0	0	0	94	3	0	0	0	0	0	0	m = sci.electronics
0	0	0	0	0	0	1	0	0	0	0	0	0	94	4	1	0	0	0	0	n = sci.med
0	0	0	0	0	1	0	0	0	0	0	0	0	1	95	1	1	1	0	0	o = sci.space
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	p = soc.religion.christian
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	96	3	0	0	q = talk.politics.guns
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	99	0	0	r = talk.politics.mideast
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	98	1	s = talk.politics.misc
16	0	0	0	0	0	0	0	1	0	0	0	0	0	0	5	3	0	5	70	t = talk.religion.misc

D-73 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde Chi-Kare İle Elde Edilen En Kötü Karmaşıklık Matrisi (RIPPER Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
1	0	0	0	27	0	10	2	0	0	0	0	0	47	1	6	0	2	0	4	a = alt.atheism
0	0	7	0	27	2	9	1	0	0	0	1	0	47	5	0	0	1	0	0	b = comp.graphics
1	0	27	0	21	6	9	0	2	0	0	2	0	28	3	1	0	0	0	0	c = comp.os.ms-windows.misc
0	0	4	0	26	3	10	0	0	0	0	0	0	55	2	0	0	0	0	0	d = comp.sys.ibm.pc.hardware
0	0	3	0	28	0	10	0	0	0	0	1	0	57	1	0	0	0	0	0	e = comp.sys.mac.hardware
0	1	14	0	22	9	9	0	0	1	0	3	0	39	2	0	0	0	0	0	f = comp.windows.x
0	0	2	0	27	2	10	1	2	3	1	0	0	51	1	0	0	0	0	0	g = misc.forsale
0	1	0	0	22	0	10	29	6	0	0	1	0	31	0	0	0	0	0	0	h = rec.autos
0	0	0	0	8	0	2	2	72	0	1	0	0	13	2	0	0	0	0	0	i = rec.motorcycles
1	0	0	0	17	0	7	0	0	35	3	1	0	36	0	0	0	0	0	0	j = rec.sport.baseball
0	0	0	0	16	0	7	0	0	5	42	0	0	29	1	0	0	0	0	0	k = rec.sport.hockey
0	0	1	0	10	0	3	0	0	0	0	59	0	27	0	0	0	0	0	0	l = sci.crypt
0	0	0	0	26	0	10	5	1	0	0	3	0	50	5	0	0	0	0	0	m = sci.electronics
0	0	0	0	27	0	10	0	1	0	1	0	0	60	0	0	0	0	0	1	n = sci.med
0	0	0	0	13	0	9	0	0	0	2	1	0	29	46	0	0	0	0	0	o = sci.space
5	0	0	0	20	0	3	0	0	0	0	0	0	23	0	46	0	0	0	3	p = soc.religion.christian
2	0	0	0	29	0	10	2	0	1	0	2	0	53	1	0	0	0	0	0	q = talk.politics.guns
0	0	0	0	17	0	2	0	0	0	1	0	0	40	0	3	0	37	0	0	r = talk.politics.mideast
1	0	0	0	26	0	9	1	0	1	0	0	0	57	1	1	0	3	0	0	s = talk.politics.misc
2	1	0	0	26	0	8	0	0	0	0	0	0	49	2	10	0	1	0	1	t = talk.religion.misc

D-74 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde Chi-Kare İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
20	0	1	0	12	0	0	2	0	0	0	1	0	42	3	17	0	2	0	0	a = alt.atheism
1	0	15	0	15	0	0	3	0	1	1	1	0	57	5	0	0	1	0	0	b = comp.graphics
1	0	58	0	8	1	0	0	0	0	1	2	0	25	2	2	0	0	0	0	c = comp.os.ms-windows.misc
1	0	12	0	15	2	1	1	0	0	0	2	0	65	1	0	0	0	0	0	d = comp.sys.ibm.pc.hardware
0	0	6	0	15	0	0	0	0	0	1	2	0	75	1	0	0	0	0	0	e = comp.sys.mac.hardware
0	0	39	0	11	0	0	0	0	1	1	3	0	43	2	0	0	0	0	0	f = comp.windows.x
0	0	8	0	15	0	0	5	2	6	6	1	0	54	3	0	0	0	0	0	g = misc.forsale
0	0	0	0	5	0	0	62	4	0	0	0	0	29	0	0	0	0	0	0	h = rec.autos
1	0	0	0	1	0	1	2	85	0	0	0	0	9	1	0	0	0	0	0	i = rec.motorcycles
1	0	0	0	5	0	0	0	0	50	10	2	0	32	0	0	0	0	0	0	j = rec.sport.baseball
0	0	0	0	6	0	1	0	0	20	51	0	0	22	0	0	0	0	0	0	k = rec.sport.hockey
1	0	1	0	5	0	0	0	0	0	1	72	0	19	0	1	0	0	0	0	l = sci.crypt
1	0	1	0	14	0	0	6	1	2	2	8	0	59	6	0	0	0	0	0	m = sci.electronics
3	0	1	0	19	0	1	1	3	2	0	0	0	68	1	1	0	0	0	0	n = sci.med
1	0	1	0	11	0	0	0	0	0	0	0	0	27	60	0	0	0	0	0	o = sci.space
9	0	0	0	3	0	0	0	0	0	0	0	0	22	0	64	0	0	0	2	p = soc.religion.christian
4	0	1	0	15	0	0	6	0	4	2	2	0	63	1	2	0	0	0	0	q = talk.politics.guns
1	1	0	0	11	0	0	1	0	0	1	0	0	41	0	4	0	39	0	1	r = talk.politics.mideast
1	0	1	0	18	0	0	2	1	3	0	0	0	66	1	4	0	3	0	0	s = talk.politics.misc
11	0	1	0	11	0	0	1	1	1	1	2	0	47	2	19	0	1	0	2	t = talk.religion.misc

D-75 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En Kötü Karmaşıklık Matrisi (RIPPER Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
7	0	0	18	2	0	0	1	0	0	0	0	0	54	1	13	1	2	0	1	a = alt.atheism
0	24	5	20	2	0	1	1	0	0	1	0	0	42	3	0	0	1	0	0	b = comp.graphics
0	3	26	22	2	1	0	0	1	0	0	0	0	41	3	1	0	0	0	0	c = comp.os.ms-windows.misc
0	1	1	27	1	0	1	1	0	0	0	0	0	65	1	0	2	0	0	0	d = comp.sys.ibm.pc.hardware
0	3	1	23	29	0	1	0	0	0	0	0	0	42	1	0	0	0	0	0	e = comp.sys.mac.hardware
0	3	16	25	0	5	0	0	0	1	0	0	0	48	2	0	0	0	0	0	f = comp.windows.x
0	4	3	20	4	0	24	3	1	0	1	0	0	39	1	0	0	0	0	0	g = misc.forsale
0	0	0	11	1	0	2	50	0	0	0	0	0	33	0	0	3	0	0	0	h = rec.autos
1	0	0	7	0	0	0	3	67	0	0	0	0	19	1	0	2	0	0	0	i = rec.motorcycles
0	0	0	17	1	0	2	0	0	8	26	0	0	46	0	0	0	0	0	0	j = rec.sport.baseball
0	0	0	13	2	0	0	0	0	14	41	0	0	28	1	0	1	0	0	0	k = rec.sport.hockey
0	0	0	16	1	0	0	0	0	0	1	45	0	34	0	0	3	0	0	0	l = sci.crypt
1	3	0	23	4	0	1	3	1	0	1	2	0	57	4	0	0	0	0	0	m = sci.electronics
1	2	0	27	0	0	1	0	1	0	1	0	0	66	1	0	0	0	0	0	n = sci.med
0	0	0	16	0	0	0	0	0	3	0	0	0	43	37	0	1	0	0	0	o = sci.space
3	0	0	18	1	0	0	0	0	0	0	0	0	34	0	42	0	1	0	1	p = soc.religion.christian
2	0	0	16	0	0	1	0	0	0	0	1	0	39	0	1	40	0	0	0	q = talk.politics.guns
0	0	0	18	0	0	0	1	0	0	0	0	0	37	0	2	3	39	0	0	r = talk.politics.mideast
0	0	0	23	0	0	1	1	0	1	0	0	0	61	0	1	8	4	0	0	s = talk.politics.misc
5	0	0	22	2	0	0	1	0	0	1	0	0	44	0	19	4	1	0	1	t = talk.religion.misc

D-76 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
28	0	1	7	2	0	0	2	0	0	0	0	0	18	1	28	0	2	11	0	a = alt.atheism
5	30	10	29	3	0	1	2	0	0	1	0	1	10	4	2	0	1	1	0	b = comp.graphics
3	7	49	21	3	1	1	0	0	0	1	0	2	5	3	3	0	0	1	0	c = comp.os.ms-windows.misc
7	5	9	53	2	0	3	1	0	0	0	0	5	10	1	1	2	0	1	0	d = comp.sys.ibm.pc.hardware
5	2	4	38	36	0	2	0	0	0	1	0	4	5	1	0	0	0	2	0	e = comp.sys.mac.hardware
6	6	37	32	0	0	0	0	0	0	2	0	3	8	2	0	0	0	4	0	f = comp.windows.x
4	2	4	34	6	0	31	5	1	1	2	0	2	0	3	0	1	0	4	0	g = misc.forsale
4	0	0	21	1	0	4	60	2	0	0	0	1	2	0	2	2	0	1	0	h = rec.autos
6	0	0	12	1	0	0	3	73	0	0	0	1	2	1	0	1	0	0	0	i = rec.motorcycles
17	1	0	15	1	0	2	0	0	1	53	0	0	6	0	2	0	0	2	0	j = rec.sport.baseball
5	0	0	9	1	0	1	0	0	1	74	0	1	3	0	0	1	0	4	0	k = rec.sport.hockey
10	0	1	11	1	0	0	0	0	0	2	49	1	10	0	2	4	0	9	0	l = sci.crypt
8	3	1	42	5	0	2	6	1	0	4	3	2	15	4	2	0	0	2	0	m = sci.electronics
14	2	0	28	0	0	2	1	1	0	2	0	4	22	1	8	1	0	14	0	n = sci.med
14	1	0	10	1	0	0	0	0	0	1	0	2	21	45	0	2	0	3	0	o = sci.space
8	0	0	10	1	0	1	0	0	0	1	0	3	5	1	63	0	0	7	0	p = soc.religion.christian
12	0	1	8	0	0	1	0	0	0	1	3	2	8	1	2	43	0	18	0	q = talk.politics.guns
12	0	0	9	0	0	0	2	0	0	1	0	1	12	0	3	3	39	18	0	r = talk.politics.mideast
14	0	1	8	1	0	2	1	1	0	1	3	1	20	1	3	11	3	29	0	s = talk.politics.misc
23	0	1	10	0	0	0	1	0	0	1	0	0	18	0	29	4	1	11	1	t = talk.religion.misc

D-77 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En Kötü Karmaşıklık Matrisi (RIPPER Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
10	0	0	0	16	0	0	2	0	0	0	0	0	44	0	23	0	2	0	3	a = alt.atheism
0	0	4	0	17	3	0	1	0	0	0	0	0	62	4	8	0	1	0	0	b = comp.graphics
0	0	22	1	14	5	0	0	2	0	0	0	0	45	3	8	0	0	0	0	c = comp.os.ms-windows.misc
0	0	5	3	22	1	0	0	0	0	0	0	0	59	1	9	0	0	0	0	d = comp.sys.ibm.pc.hardware
0	0	1	5	20	0	0	0	0	0	0	0	0	64	2	8	0	0	0	0	e = comp.sys.mac.hardware
0	0	8	0	16	7	0	0	0	0	1	4	0	55	2	7	0	0	0	0	f = comp.windows.x
0	0	3	2	17	0	0	3	1	0	3	0	0	60	2	9	0	0	0	0	g = misc.forsale
0	0	1	1	16	0	0	44	1	0	0	0	0	33	0	4	0	0	0	0	h = rec.autos
1	0	0	1	5	0	0	4	66	0	1	0	0	19	3	0	0	0	0	0	i = rec.motorcycles
0	0	0	0	9	0	0	0	0	8	27	1	0	49	0	5	0	0	1	0	j = rec.sport.baseball
0	0	0	0	11	0	0	1	0	13	37	0	0	34	1	3	0	0	0	0	k = rec.sport.hockey
1	0	0	0	11	0	0	0	0	0	0	54	0	29	1	2	0	0	2	0	l = sci.crypt
0	0	0	0	18	0	0	4	1	0	1	4	0	56	6	10	0	0	0	0	m = sci.electronics
0	0	0	0	20	0	0	0	1	0	1	0	0	67	0	10	0	0	0	1	n = sci.med
0	0	0	0	10	0	0	0	0	1	2	0	0	41	38	6	2	0	0	0	o = sci.space
6	0	0	0	11	0	0	0	0	0	0	0	0	34	0	43	0	0	0	6	p = soc.religion.christian
3	0	0	0	20	0	0	1	0	0	0	2	0	64	1	9	0	0	0	0	q = talk.politics.guns
0	0	0	0	11	0	0	1	0	0	0	0	0	39	0	11	1	36	0	1	r = talk.politics.mideast
1	0	0	0	18	0	0	1	0	0	0	1	0	60	1	12	2	3	1	0	s = talk.politics.misc
3	0	0	0	17	0	0	0	0	1	0	1	0	50	0	24	0	1	0	3	t = talk.religion.misc

D-78 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
18	2	1	0	0	0	6	2	0	0	0	0	1	2	1	28	0	5	22	12	a = alt.atheism
4	8	14	0	3	1	40	3	0	0	1	0	2	8	5	0	1	1	4	5	b = comp.graphics
1	1	57	3	1	1	15	0	1	0	1	1	0	5	2	2	5	0	2	2	c = comp.os.ms-windows.misc
5	4	12	12	14	1	32	1	0	0	0	2	0	8	1	0	2	0	3	3	d = comp.sys.ibm.pc.hardware
9	3	6	22	8	0	31	0	0	0	1	2	0	4	0	1	3	0	5	5	e = comp.sys.mac.hardware
3	5	38	0	0	1	33	0	0	0	2	3	1	4	2	0	2	0	3	3	f = comp.windows.x
2	0	8	4	3	0	50	5	1	0	10	1	1	2	3	1	4	0	2	3	g = misc.forsale
1	2	0	2	3	0	14	62	2	0	0	0	0	1	0	1	6	0	4	2	h = rec.autos
2	1	0	3	6	0	5	3	73	0	0	0	0	2	1	0	0	0	0	4	i = rec.motorcycles
10	2	0	0	1	0	14	0	0	3	49	2	1	0	0	1	3	0	6	8	j = rec.sport.baseball
2	0	0	0	0	0	10	0	0	1	75	0	0	1	0	0	1	1	7	2	k = rec.sport.hockey
4	0	1	0	1	0	6	0	0	0	2	66	2	3	0	1	7	1	5	1	l = sci.crypt
5	6	0	0	8	1	30	6	1	1	4	8	0	7	6	1	6	0	5	5	m = sci.electronics
9	3	1	1	2	0	25	1	1	1	1	0	2	10	1	4	5	0	25	8	n = sci.med
9	3	1	0	0	0	11	0	0	0	1	1	1	2	46	0	7	0	8	10	o = sci.space
5	1	0	1	1	0	10	0	0	0	1	0	0	4	1	67	4	0	4	1	p = soc.religion.christian
11	3	1	0	2	0	8	5	0	1	2	2	2	5	1	4	19	0	29	5	q = talk.politics.guns
9	3	0	0	1	0	8	1	0	0	1	0	0	4	0	5	9	39	16	4	r = talk.politics.mideast
12	3	1	0	1	0	6	2	1	0	1	0	0	5	1	2	21	3	35	6	s = talk.politics.misc
19	0	1	1	0	0	2	1	0	0	1	2	0	8	0	29	8	1	14	13	t = talk.religion.misc

D-79 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde PCA İle Elde Edilen En Kötü Karmaşıklık Matrisi (RIPPER Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
32	0	13	0	2	4	0	0	2	2	0	4	8	0	3	7	1	5	8	9	a = alt.atheism
4	14	21	1	4	19	4	1	0	0	0	7	13	0	0	0	1	5	6	0	b = comp.graphics
1	2	35	9	10	15	1	1	0	1	0	7	6	0	0	0	0	4	8	0	c = comp.os.ms-windows.misc
0	2	13	33	21	7	4	1	0	0	0	5	8	1	0	0	0	1	4	0	d = comp.sys.ibm.pc.hardware
1	0	18	15	26	6	3	0	0	0	0	6	12	1	0	1	0	7	4	0	e = comp.sys.mac.hardware
0	10	24	1	6	29	2	0	0	0	0	7	10	1	1	0	0	4	5	0	f = comp.windows.x
0	0	8	5	8	2	51	6	0	0	0	5	11	0	0	0	0	2	2	0	g = misc.forsale
0	0	14	0	5	6	1	24	11	1	0	3	13	5	0	0	2	6	8	1	h = rec.autos
0	0	2	0	3	1	3	9	65	2	2	4	4	0	0	0	0	2	3	0	i = rec.motorcycles
2	1	7	0	1	2	0	1	0	59	14	3	6	1	0	1	0	1	1	0	j = rec.sport.baseball
1	0	2	0	2	1	1	0	2	13	70	1	0	0	0	1	0	2	4	0	k = rec.sport.hockey
0	2	8	0	5	2	0	0	1	0	0	64	9	0	0	0	3	3	3	0	l = sci.crypt
0	3	19	6	10	6	4	6	3	0	0	5	30	0	1	0	0	4	3	0	m = sci.electronics
3	0	7	0	5	3	1	0	2	1	1	1	10	55	2	2	0	0	6	1	n = sci.med
0	1	8	0	6	4	0	0	3	1	0	3	8	1	53	1	2	2	6	1	o = sci.space
6	0	12	0	1	0	0	1	3	0	1	4	7	2	0	56	0	3	3	1	p = soc.religion.christian
1	0	11	0	5	1	0	1	1	0	0	7	6	1	1	0	30	6	25	4	q = talk.politics.guns
3	1	3	0	3	0	0	1	0	2	0	4	5	1	3	1	1	68	3	1	r = talk.politics.mideast
2	0	8	1	4	2	0	1	1	2	0	6	6	1	1	0	24	6	34	1	s = talk.politics.misc
16	0	9	1	5	1	0	2	3	1	0	2	3	0	1	16	14	4	8	14	t = talk.religion.misc

D-80 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde PCA İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
60	0	0	1	0	0	1	0	2	2	0	0	1	3	2	13	2	6	2	5	a = alt.atheism
1	47	4	5	3	18	5	2	0	1	0	4	3	2	4	1	0	0	0	0	b = comp.graphics
1	9	37	11	5	19	3	2	0	2	0	3	1	3	1	2	0	0	0	1	c = comp.os.ms-windows.misc
1	6	8	52	15	1	5	3	2	1	0	1	2	1	2	0	0	0	0	0	d = comp.sys.ibm.pc.hardware
0	0	3	31	44	3	6	1	0	0	0	2	6	1	2	0	0	0	1	0	e = comp.sys.mac.hardware
0	15	11	0	0	61	2	1	0	1	1	0	0	1	3	1	1	1	1	0	f = comp.windows.x
2	2	1	3	2	0	77	1	3	1	1	0	2	0	4	0	0	0	0	1	g = misc.forsale
0	0	0	3	0	1	15	38	22	0	0	1	5	4	4	2	3	0	2	0	h = rec.autos
0	0	1	1	0	0	2	0	92	1	0	1	0	0	0	0	0	0	2	0	i = rec.motorcycles
3	0	0	0	0	0	3	1	1	58	22	1	0	1	4	2	2	1	1	0	j = rec.sport.baseball
1	0	1	2	0	0	1	0	0	3	89	0	0	1	0	0	0	1	1	0	k = rec.sport.hockey
0	0	1	0	0	1	1	0	1	2	0	83	3	0	1	0	2	1	3	1	l = sci.crypt
2	5	4	4	9	2	8	5	0	1	0	4	45	2	6	1	1	0	0	1	m = sci.electronics
8	3	0	1	0	0	1	0	1	2	0	0	0	77	3	2	1	0	1	0	n = sci.med
1	1	0	0	1	1	2	1	0	0	1	1	1	3	85	1	0	0	1	0	o = sci.space
2	0	0	0	0	0	1	1	1	1	0	0	1	2	0	82	2	3	3	1	p = soc.religion.christian
2	0	0	0	0	1	1	1	1	2	2	5	3	3	4	1	51	3	20	0	q = talk.politics.guns
4	0	0	1	0	0	0	0	1	1	0	0	0	2	1	1	3	85	1	0	r = talk.politics.mideast
3	0	0	0	0	0	0	1	0	2	0	4	0	2	4	2	29	9	44	0	s = talk.politics.misc
36	0	1	0	0	0	1	0	2	3	0	2	1	3	1	22	20	1	5	2	t = talk.religion.misc

D-81 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde LSA İle Elde Edilen En Kötü Karmaşıklık Matrisi (RIPPER Algoritması İle)

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	<-- classified as
31	0	2	4	0	6	0	5	7	0	6	0	7	7	2	10	0	2	6	5	a = alt.atheism
0	31	7	7	2	15	1	3	4	0	6	0	7	6	1	1	0	2	7	0	b = comp.graphics
1	10	35	14	2	14	0	2	2	0	5	1	4	7	1	0	0	0	2	0	c = comp.os.ms-windows.misc
0	4	9	26	21	3	2	6	6	0	4	0	11	6	0	0	0	0	2	0	d = comp.sys.ibm.pc.hardware
0	3	7	20	26	2	11	3	7	0	3	2	5	8	1	0	0	0	2	0	e = comp.sys.mac.hardware
0	6	20	5	1	39	2	4	4	0	4	0	5	2	3	1	0	0	4	0	f = comp.windows.x
0	0	2	10	4	2	43	7	3	2	5	0	9	3	4	0	2	0	4	0	g = misc.forsale
0	0	4	6	1	2	8	43	9	1	4	1	11	2	0	0	2	2	4	0	h = rec.autos
3	0	4	5	0	1	2	4	66	0	3	0	8	2	0	0	0	0	2	0	i = rec.motorcycles
0	0	4	2	0	2	2	1	1	33	42	0	3	4	1	0	1	1	3	0	j = rec.sport.baseball
0	0	1	3	0	0	0	2	3	20	59	0	6	1	0	1	0	0	4	0	k = rec.sport.hockey
0	0	2	6	0	7	0	3	6	0	4	58	8	3	0	0	0	0	3	0	l = sci.crypt
0	7	5	13	2	5	3	9	8	1	6	3	20	10	3	0	0	0	5	0	m = sci.electronics
1	1	3	1	0	5	2	2	5	1	4	0	8	57	3	1	0	0	5	1	n = sci.med
0	0	2	2	0	4	2	1	3	0	2	1	5	5	68	0	1	0	4	0	o = sci.space
6	0	4	3	0	2	0	4	2	0	5	0	7	5	1	55	0	0	5	1	p = soc.religion.christian
1	0	2	2	0	5	1	9	6	0	6	5	4	8	0	0	35	1	14	1	q = talk.politics.guns
2	0	2	2	0	0	1	0	4	1	3	1	1	6	1	0	4	66	5	1	r = talk.politics.mideast
1	0	8	4	0	3	1	2	5	2	6	2	5	7	1	1	21	2	28	1	s = talk.politics.misc
21	0	2	5	0	4	0	3	9	0	2	1	6	6	2	13	13	1	8	4	t = talk.religion.misc

D-82 20 Özelliğe İndirgenmiş 20-Newsgroups Veri Kümesinde LSA İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle)

```
  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t   <-- classified as
 61  0  0  0  0  0  0  1  2  1  0  0  1  0  2 14  3  2  2 11 |  a = alt.atheism
  3 61  5  4  0  7  2  0  0  1  0  0 15  0  2  0  0  0  0  0 |  b = comp.graphics
  3 10 45  9  1 13  3  0  1  2  0  3  8  1  1  0  0  0  0  0 |  c = comp.os.ms-windows.misc
  1  4  8 39 22  1  6  1  1  0  0  0 15  1  1  0  0  0  0  0 |  d = comp.sys.ibm.pc.hardware
  0  2  1 30 45  1  6  1  0  1  0  0 12  0  1  0  0  0  0  0 |  e = comp.sys.mac.hardware
  0 17 10  0  0 57  2  1  0  0  0  0  8  0  3  0  2  0  0  0 |  f = comp.windows.x
  0  1  0  1  5  2 79  1  3  1  1  0  5  0  1  0  0  0  0  0 |  g = misc.forsale
  0  0  0  0  1  0  6 66  4  0  0  0 12  3  3  0  2  0  3  0 |  h = rec.autos
  2  0  0  0  1  0  5  6 82  1  0  0  2  0  0  0  0  0  1  0 |  i = rec.motorcycles
  4  0  0  0  0  0  4  0  0 47 34  0  4  2  0  0  1  1  1  2 |  j = rec.sport.baseball
  2  0  0  1  0  0  1  0  0 15 77  0  1  0  0  0  2  1  0  0 |  k = rec.sport.hockey
  1  0  0  0  0  2  1  2  1  2  0 70 10  0  1  0  7  1  1  1 |  l = sci.crypt
  0  4  1  8  7  2  7  6  2  3  0  4 52  2  0  0  1  0  0  1 |  m = sci.electronics
  6  1  0  0  0  2  1  0  0  3  0  0  5 78  1  2  0  0  1  0 |  n = sci.med
  2  1  0  0  0  4  2  0  0  0  0  0  5  4 80  0  1  0  0  1 |  o = sci.space
  5  0  0  1  0  0  2  0  0  0  0  0  3  3  0 78  2  2  4  0 |  p = soc.religion.christian
  3  0  0  0  0  1  1  0  1  1  0  2  2  1  2  1 72  1  7  5 |  q = talk.politics.guns
  7  0  0  0  0  0  0  0  0  3  0  0  1  1  1  2  4 78  2  1 |  r = talk.politics.mideast
  6  0  0  0  1  0  1  1  0  1  0  1  2  1  3  1 38  3 39  2 |  s = talk.politics.misc
 38  0  0  0  0  1  1  1  1  3  0  1  1  0  0 20 20  0  5  8 |  t = talk.religion.misc
```

D-83 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En Kötü Karmaşıklık Matrisi (SVM Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
 706    0    2    6    1    0    1    0    1    0 |    a = acq
   0    0    0    0    1    0    0    0    0    0 |    b = corn
   2    0  150    1    3    0    0   11    0    0 |    c = crude
   4    0    1 1071    7    0    0    0    0    0 |    d = earn
   0    0    0    0  135    0    0    0    2    0 |    e = grain
   0    0    0    0    0   62   38    0    1    0 |    f = interest
   0    0    0    1    0    3  159    0    1    0 |    g = money-fx
   2    0    2    0    2    0    0   53    1    0 |    h = ship
   0    0    1    0    2    0    5    0  104    0 |    i = trade
   0    0    0    0    3    0    0    0    0    0 |    j = wheat
```

D-84 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde Soyut Özellik Çıkarımı İle Elde Edilen En İyi Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
 701    0    5    7    1    0    1    0    2    0 |    a = acq
   0    0    0    0    1    0    0    0    0    0 |    b = corn
   1    0  156    1    1    0    0    8    0    0 |    c = crude
   4    0    1 1070    4    0    0    4    0    0 |    d = earn
   0    0    0    0  132    0    0    3    2    0 |    e = grain
   0    0    0    0    0   73   27    0    1    0 |    f = interest
   0    0    0    1    0    4  157    0    2    0 |    g = money-fx
   0    0    7    0    0    0    0   53    0    0 |    h = ship
   0    0    1    0    1    0    4    0  106    0 |    i = trade
   0    0    0    0    3    0    0    0    0    0 |    j = wheat
```

D-85 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde Chi-Kare İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle)

```
   a   b   c   d   e   f   g   h   i   j   <-- classified as
  12   3  27   2   0   1 657  15   0   0 |   a = acq
   0   1   0   0   0   0   0   0   0   0 |   b = corn
   0   6 106   2   0   0   9  43   1   0 |   c = crude
   8   0  36 965   0   1  72   0   1   0 |   d = earn
   0  51   1   0   7   1  23   2   0  52 |   e = grain
   0   0   0   0   0  63  38   0   0   0 |   f = interest
   0   0   1   0   0  23 139   0   0   1 |   g = money-fx
   0  11   6   0   0   0  29  12   1   1 |   h = ship
   0   9   5   0   0   0  90   2   6   0 |   i = trade
   0   1   0   0   0   0   0   0   0   2 |   j = wheat
```

D-86 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde Chi-Kare İle Elde Edilen En İyi Karmaşıklık Matrisi (SVM Algoritması İle)

```
   a   b   c   d   e   f   g   h   i   j   <-- classified as
 652   0  31   9   1   0   7   0  17   0 |   a = acq
   0   0   0   0   1   0   0   0   0   0 |   b = corn
  14   0 145   3   0   0   1   0   4   0 |   c = crude
  77   0   7 988   0   1   5   0   5   0 |   d = earn
  20   0   3   1 103   1   0   0   9   0 |   e = grain
   9   0   0   1   0  49  38   0   4   0 |   f = interest
  71   0   2   3   0   8  58   0  22   0 |   g = money-fx
  38   0  13   0   6   0   0   0   3   0 |   h = ship
  19   0   6   0   4   0   7   0  76   0 |   i = trade
   0   0   0   0   3   0   0   0   0   0 |   j = wheat
```

D-87 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle)

```
   a   b   c   d   e   f   g   h   i   j   <-- classified as
 459 207  11  14   0  19   0   5   0   2 |   a = acq
   0   1   0   0   0   0   0   0   0   0 |   b = corn
   3  27  98   3   0   1   0  33   2   0 |   c = crude
  41  35   9 995   0   2   0   0   1   0 |   d = earn
   0  60   1   3  10   3   0   1   0  59 |   e = grain
   0  13   1   0   0  83   1   0   1   2 |   f = interest
   1  81   2   0   0  59  10   0   4   7 |   g = money-fx
   0  49   4   0   0   1   0   3   1   2 |   h = ship
   3  76   1   0   0   3   6   8  11   4 |   i = trade
   0   1   0   0   0   0   0   0   0   2 |   j = wheat
```

D-88 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde Korelasyon Katsayısı İle Elde Edilen En İyi Karmaşıklık Matrisi (LINEAR Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
 619    0   18   23    0    2   47    0    8    0 |    a = acq
   0    0    0    0    0    0    0    0    1    0 |    b = corn
   8    0  143    6    0    1    6    0    3    0 |    c = crude
  52    0    1 1020    0    1    9    0    0    0 |    d = earn
  30    0    2    5   72    1   15    0   12    0 |    e = grain
   3    0    0    1    0   72   19    0    6    0 |    f = interest
  40    0    2    2    0   34   60    0   26    0 |    g = money-fx
  30    0   18    1    0    0    5    0    6    0 |    h = ship
  11    0    5    2    2    1    7    0   84    0 |    i = trade
   0    0    0    0    3    0    0    0    0    0 |    j = wheat
```

D-89 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En Kötü Karmaşıklık Matrisi (Naive Bayes Algoritması İle)

```
   a   b   c   d   e   f   g   h   i   j   <-- classified as
   0 598  54   6   0   0   7   0   2  50 |   a = acq
   0   1   0   0   0   0   0   0   0   0 |   b = corn
   0 107  48   0   0   0   4   0   0   8 |   c = crude
   1  39  67 957   2   0   8   0   1   8 |   d = earn
   0 106   4  10   0   5   3   0   0   9 |   e = grain
   0  12   1   0   0  70   7   0   0  11 |   f = interest
   0  72   3   0   0  37  20   0   2  30 |   g = money-fx
   0  55   1   0   0   0   0   0   0   4 |   h = ship
   0  81   1   0   0   0   9   0  12   9 |   i = trade
   0   3   0   0   0   0   0   0   0   0 |   j = wheat
```

D-90 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde Karşılıklı Bilgi İle Elde Edilen En İyi Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
 617    0    9   53   10    0   11    3   14    0 |    a = acq
   0    0    0    0    0    0    0    0    1    0 |    b = corn
  93    0   30   23    3    0    8    6    4    0 |    c = crude
  52    0    2 1018    1    0    6    0    4    0 |    d = earn
  74    0    7    9   26    2    7    1   11    0 |    e = grain
   8    0    0    3    0   48   40    0    2    0 |    f = interest
  48    0    0    7    7   16   71    1   14    0 |    g = money-fx
  44    0    3    1    5    0    2    3    2    0 |    h = ship
  25    0    0    2    2    0    9    0   74    0 |    i = trade
   1    0    0    0    2    0    0    0    0    0 |    j = wheat
```

D-91 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde PCA İle Elde Edilen En Kötü Karmaşıklık Matrisi (RIPPER Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
 641    0    8   54    0    2    5    7    0    0 |    a = acq
   0    0    0    0    1    0    0    0    0    0 |    b = corn
   3    0  132    5    0    0    1   26    0    0 |    c = crude
  21    0    2 1046    0    0    1   12    1    0 |    d = earn
   0    0    0    3  118    0    1    6    9    0 |    e = grain
   0    0    2   18    1   37   39    1    3    0 |    f = interest
   2    0    1   21    0   11  124    0    5    0 |    g = money-fx
   2    0   14    5    0    0    0   39    0    0 |    h = ship
   0    0    1   18    6    2    8    2   75    0 |    i = trade
   0    0    0    0    3    0    0    0    0    0 |    j = wheat
```

D-92 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde PCA İle Elde Edilen En İyi Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

```
   a    b    c    d    e    f    g    h    i    j   <-- classified as
 676    0    9   27    0    0    1    3    1    0 |    a = acq
   0    0    0    0    1    0    0    0    0    0 |    b = corn
   3    0  139    4    0    0    0   21    0    0 |    c = crude
  24    0    0 1051    0    0    0    7    1    0 |    d = earn
   0    0    0    0  126    0    0    3    8    0 |    e = grain
   0    0    0    1    1   59   38    0    2    0 |    f = interest
   3    0    0    2    0   13  140    0    6    0 |    g = money-fx
   3    0   11    1    2    0    0   42    1    0 |    h = ship
   1    0    2    0    5    1    6    2   95    0 |    i = trade
   0    0    0    0    3    0    0    0    0    0 |    j = wheat
```

D-93 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde LSA İle Elde Edilen En Kötü Karmaşıklık Matrisi (LINEAR Algoritması İle)

a	b	c	d	e	f	g	h	i	j	<-- classified as
691	0	1	25	0	0	0	0	0	0	a = acq
0	0	0	1	0	0	0	0	0	0	b = corn
21	0	107	39	0	0	0	0	0	0	c = crude
20	0	0	1062	0	0	1	0	0	0	d = earn
8	0	0	125	1	0	0	0	3	0	e = grain
1	0	0	27	0	0	73	0	0	0	f = interest
7	0	0	36	0	0	121	0	0	0	g = money-fx
14	0	3	42	0	0	0	0	1	0	h = ship
13	0	0	61	0	0	5	0	33	0	i = trade
0	0	0	3	0	0	0	0	0	0	j = wheat

D-94 10 Özelliğe İndirgenmiş ModApte-10 Veri Kümesinde LSA İle Elde Edilen En İyi Karmaşıklık Matrisi (10 En Yakın Komşu Algoritması İle)

a	b	c	d	e	f	g	h	i	j	<-- classified as
680	0	6	22	1	0	1	5	2	0	a = acq
0	0	0	0	1	0	0	0	0	0	b = corn
2	0	144	1	5	0	0	15	0	0	c = crude
13	0	0	1063	0	0	0	7	0	0	d = earn
0	0	0	0	125	0	1	5	6	0	e = grain
0	0	0	0	1	51	45	1	3	0	f = interest
2	0	0	0	0	15	139	0	8	0	g = money-fx
1	0	10	2	10	0	0	35	2	0	h = ship
0	0	1	0	13	2	5	2	89	0	i = trade
0	0	0	0	3	0	0	0	0	0	j = wheat

EK-E

ARFF DOSYA FORMATI ÖRNEKLERİ

Bölüm 6'da tanıtımı yapılan WEKA adlı uygulama ortamının kullandığı ARFF dosya formatının içeriğini tanıtmak üzere örnekler bu bölümde verilmiştir.

E-1 Örnek Bir Veri Kümesinin Standart Halinin ARFF Formatındaki Gösterimi

```
@relation '2class_sample_dataset_raw'
@attribute class {class1,class2}
@attribute t1 numeric
@attribute t2 numeric
@attribute t3 numeric
@attribute t4 numeric
@attribute t5 numeric
@attribute t8 numeric
@attribute t6 numeric
@attribute t7 numeric
@data
{1 0.808019,2 0.18285,3 0.808019}
{2 0.219212,4 0.968703,6 0.593957}
{2 0.978728,5 0.617508}
{0 class2,5 0.400688,7 1.085667}
{0 class2,5 0.219212,6 0.593957,8 0.968703}
{0 class2,2 0.378634,5 0.378634,7 1.025913}
```

E-2 Örnek Bir Veri Kümesinin Soyut Özellik Çıkarımı Uygulanmış Halinin ARFF Formatındaki Gösterimi

```
@relation ''2class_sample_dataset_afe'
@attribute attr1 numeric
@attribute attr2 numeric
@attribute class {class1, class2}
@data
0.918, 0.082, class1
0.718, 0.282, class1
0.585, 0.415, class1
0.137, 0.863, class2
0.289, 0.711, class2
0.313, 0.687, class2
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	:Göksel BİRİCİK
Doğum Tarihi ve Yeri	:İstanbul, 10 Ocak 1981
Yabancı Dili	:İngilizce
E-posta	:goksel@ce.yildiz.edu.tr

ÖĞRENİM DURUMU

Derece	**Alan**	**Okul/Üniversite**	**Mezuniyet Yılı**
Y. Lisans	Bilgisayar Mühendisliği	Yıldız Teknik Üniversitesi	2004
Lisans	Bilgisayar Bilimleri Mühendisliği	Yıldız Teknik Üniversitesi	2002
Lise	Fen Bilimleri	Vefa Anadolu Lisesi	1998

İŞ TECRÜBESİ

Yıl	**Firma/Kurum**	**Görevi**
2003-2011	Yıldız Teknik Üniversitesi, Elektrik-Elektronik Fakültesi, Bilgisayar Mühendisliği Bölümü, Yazılım Anabilim Dalı	Araştırma Görevlisi
2001	Borusan Holding, BNet	Donanım Stajyeri, Yazılım Mühendisi
1999	32 Bit Bilgisayar Hizmetleri Ltd.Şti.	Yazılım Stajyeri

YAYINLARI

Makale

1.(2011) Biricik, G., Diri, B. ve Sönmez, A.C., "Abstract Feature Extraction For Text Classification", Turkish Journal of Electrical Engineering & Computer Sciences dergisinde yayımlanacak (kabul edildi).

Bildiri

1.(2011) Biricik, G. ve Diri, B., "Comparing the Efficiency of Abstract Feature Extractor with Other Dimension Reduction Methods on Reuters-21578 Dataset", Computer Science Student Workshop, Nisan 2011, Sabancı University, İstanbul.

2.(2011) Türkmen, H.İ., Diri, B., Biricik, G. ve Doğan, R., "Konuşma Dili Kullanılarak Demografik Bilgilerin Sınıflandırılması", 19. Sinyal İşleme, İletişim ve Uygulamaları Kurultayı (SIU 2011), Nisan 2011, Antalya.

3.(2010) Dağdaş, S., Biricik, G. ve Diri, B., "A Focused Web Spider Specialized for Turkish Language", International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2010), Kayseri.

4.(2009) Biricik, G., Diri, B. ve Sönmez, A.C., "A New Method for Attribute Extraction with Application on Text Classification5th International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control (ICSCCW 2009), Eylül 2009, KKTC.

5.(2009) Biricik, G. ve Diri, B., "Impact of A New Attribute Extraction Algorithm on Web Page Classification", 5th International Conference on Data Mining (DMIN 2009), Temmuz 2009, Las Vegas, ABD.

6.(2006) Biricik G., "Bir ve İki Boyutlu Kendini Örgütleyen Ağlarla Renkli Görüntü Kuantalama", Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu (ASYU 2006), Haziran 2006, Yıldız Teknik Üniversitesi,

İstanbul.

7.(2006) Uğurlu, E.S. ve Biricik, G., "Q-Öğrenme Yöntemi İçin Olasılık Temelli Sekiz Yönlü Hareket Stratejisi Geliştirilmesi ve Uygulanması", Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu (ASYU 2006), Haziran 2006, Yıldız Teknik Üniversitesi, İstanbul.

8.(2006) Uğurlu, E.S. ve Biricik, G., "Olasılık Temelli Hareket Stratejisi Kullanan Q-Öğrenme", 14.IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SIU06), Nisan 2006, Sabancı Üniversitesi, Antalya.

9.(2005) Biricik G., Yavuz, A.A., Kartal, Ö. ve Kalıpsız, O., "Developing Information System with N-Tier Architecture: Hospital Management Information System", International Informatics Congress (BILTEK2005), Haziran 2005, TBD, Eskişehir.

10.(2005) Biricik G. ve Albayrak, S., "Content-Based Image Retrieval System Using Color Features", International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2005), Haziran 2005, Yıldız Teknik Üniversitesi, İstanbul.

11.(2004) Biricik G., Albayrak S. ve Kalıpsız O., "Renk Özelliğine Dayalı Bir İçerik Tabanlı Görüntü Erişim Sistemi", Havacılıkta İleri Teknolojiler (HİTEK2004) Sempozyumu, Aralık 2004, Hava Harp Okulu, İstanbul.

12.(2003) Biricik G. ve Bozkurt, Ö.Ö., "Farklı Yazılım Teknolojilerinde Mobil Kod Güvenliğini Sağlama Yöntemleri", 1.Ulusal Yazılım Mühendisliği Sempozyumu UYMS'03, Ekim 2003, Ege Üniversitesi, İzmir.

13.(2003) Biricik, G., Buharalı, A. ve Kalıpsız, O., "The Impact of Object Oriented Development Methods on Software Quality", Impact of Software Process on Quality Workshop (IMPROQ03), June 2003, Bilkent University, Ankara.

Kitap

1.(2009) Kalıpsız, O., Dikenelli, O., Diri, B. ve Biricik, G. (Editörler), 4. Ulusal Yazılım Mühendisliği Sempozyumu Bildiriler Kitabı, Elektrik Mühendisleri Odası –Yıldız Teknik Üniversitesi, İstanbul.

2.(2006) Kalıpsız, O., Buharalı, A. ve Biricik, G., Bilgisayar Bilimlerinde Sistem Analizi ve Tasarımı, Papatya Yayıncılık, İstanbul.

Proje

1.(2004) Kalıpsız, O., Albayrak, S., Kalay, M.U., Biricik, G. ve Altun, O., Çoklu Ortam Veritabanlarında İçeriğe Dayalı Sorgulama Sistemi, Yıldız Teknik Üniversitesi Bilimsel Araştırma Projesi, 2002-2004.