METİN MADENCİLİĞİ YÖNTEMİ İLE HABER SİTELERİNDEKİ KÖŞE YAZILARININ SINIFLANDIRILMASI

2012
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ

Mehmet Fatih KARACA

METİN MADENCİLİĞİ YÖNTEMİ İLE HABER SİTELERİNDEKİ KÖŞE YAZILARININ SINIFLANDIRILMASI

Mehmet Fatih KARACA

Karabük Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalında
Yüksek Lisans Tezi
Olarak Hazırlanmıştır

KARABÜK
Haziran 2012

Mehmet Fatih KARACA tarafından hazırlanan "METİN MADENCİLİĞİ YÖNTEMİ İLE HABER SİTELERİNDEKİ KÖŞE YAZILARININ SINIFLANDIRILMASI" başlıklı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarım.

Yrd. Doç. Dr. Salih GÖRGÜNOĞLU

Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir. 28/06/2012

Ünvanı, Adı SOYADI (Kurumu)	İmzası
Başkan : Yrd. Doç. Dr. Baha ŞEN (KBÜ)	.....
Üye : Yrd. Doç. Dr. Salih GÖRGÜNOĞLU (KBÜ)	.....
Üye : Yrd. Doç. Dr. İbrahim ÇAYIROĞLU (KBÜ)	.....

...../...../2012

KBÜ Fen Bilimleri Enstitüsü Yönetim Kurulu, bu tez ile, Yüksek Lisans derecesini onamıştır.

Prof. Dr. Nizamettin KAHRAMAN

Fen Bilimleri Enstitüsü Müdürü

“Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”

Mehmet Fatih KARACA

ÖZET

Yüksek Lisans Tezi

METİN MADENCİLİĞİ YÖNTEMİ İLE HABER SİTELERİNDEKİ KÖŞE YAZILARININ SINIFLANDIRILMASI

Mehmet Fatih KARACA

Karabük Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Yrd. Doç. Dr. Salih GÖRGÜNOĞLU
Haziran 2012, 99 sayfa

İnternetin günümüzde hayatımızın her alanına girmiş olmasıyla birlikte verilerin yaşantımızdaki boyutu ve değeri artmıştır. Büyük miktarlardaki verilerden anlamlı bilgi çıkarmak günümüzde kişi ve firmaları bilgi temelli ekonomi çağında diğerlerinden bir adım öne geçirmektedir. İnternette bulunan bütün veriler bir çok kişi için gerekli veya faydalı değildir. Verilerden katkı sağlayacak bilgilerin elde edilmesi için çeşitli madencilik teknikleri uygulanmalıdır. Veri madenciliği yapısal veriler üzerinde işlemleri gerçekleştirir. Metinler yapısal veriler değildir. Metinlerin, veri madenciliği tekniklerinin uygulanabileceği yapısal veri haline dönüştürülmesi işlemi metin madenciliği ile gerçekleştirilir. Veri madenciliği teknikleri uygulanmadan önce verilerin hazırlanması, ön işlemden geçirilmesi gerekmektedir. Metinlerden bilgi çıkarmak, metni sınıflandırmak, aranan bilgiye kısa sürede ulaşmak metin madenciliğinin popülaritesini arttırmış, bu konuda çalışmalar yapılmasını gerekli hale getirmiştir.

Metin sınıflandırma, sistemin önceden tanımlanmış kategorilere eğitim dokümanlarını kullanarak verilen metnin sınıfına karar vermesi işlemidir.

Bu çalışmada haber sitelerindeki köşe yazılarının otomatik olarak alınması ve ekonomi, spor, sağlık, eğitim, yaşam olarak sınıflandırılması gerçekleştirilmiştir.

Anahtar Sözcükler : Veri madenciliği, metin madenciliği, haber siteleri, köşe yazıları.

Bilim Kodu **:** 902.1.014

ABSTRACT

M.Sc. Thesis

CLASSIFICATION OF ONLINE NEWSPAPERS ARTICLES THROUGH TEXT MINING METHOD

Mehmet Fatih KARACA

Karabük University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Thesis Advisor:
Asst. Prof. Dr. Salih GÖRGÜNOĞLU
June 2012, 99 pages

With the access of internet to every aspect of the community, the volume and value of data has increased. Extracting meaningful information out of vast volumes of data makes one or a firm go one step further than the others in this information-oriented age of economy. All data, available on the internet, is not essential or useful to most. A range of mining techniques must be implemented to acquire useful information out of data. Data mining operates on structural data. Texts are not structural data. The process that texts are transformed into structural data to which data mining can be applied is realized through data mining. It is a must that data be ready and exposed to pre-processing before data mining techniques are applied. Acquiring information out of texts and classifying them, and getting an instant access to the information have given rise to text mining, necessitating the studies on it.

Text classification is the process in which the class of text is determined through system's using the training documents given into pre-determined categories.

In this study, it has been achieved that articles of online newspapers are automatically extracted and classified into categories as economy, sports, health, education and life.

Key Word **:** Data mining, text mining, online newspapers, articles.
Science Code **:** 902.1.014

TEŞEKKÜR

Bu tez çalışmasının planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren sayın hocam Yrd. Doç. Dr. Salih GÖRGÜNOĞLU'na sonsuz teşekkürlerimi sunarım.

Ayrıca tez çalışmam boyunca fikir ve zamanlarını esirgemeyen öğretim görevlileri Ümit YILDIRIM ve Turgut ÖZSEVEN'e teşekkürü bir borç bilirim.

Bugünlere gelmemde en büyük pay sahibi olan aileme, her zaman manevi desteklerini hissettiğim sevgili eşim Şule KARACA ve kızım Zeynep İpek KARACA'ya tüm kalbimle teşekkür ederim.

İÇİNDEKİLER

Sayfa

KABUL. ii
ÖZET iv
ABSTRACT vi
TEŞEKKÜR viii
İÇİNDEKİLER ix
ŞEKİLLER DİZİNİ xii
ÇİZELGELER DİZİNİ xiv
SİMGELER VE KISALTMALAR DİZİNİ xvi

BÖLÜM 1 1
GİRİŞ 1

BÖLÜM 2 6
VERİ MADENCİLİĞİ 6
2.1. YAPISAL VERİ 7
2.2. KDD SÜRECİ 8
2.2.1. Veri Seçme 9
2.2.2. Ön İşlem 9
2.2.3. Dönüşüm 10
2.2.3.1. Veri temizleme 10
2.2.3.2. Verilerin Bütünleştirilmesi 10
2.2.3.3. Veri Dönüştürme 11
2.2.3.4. Veri İndirgeme 11
2.2.4. Veri Madenciliği 11
2.2.5. Yorumlama / Değerlendirme 12

BÖLÜM 3 13
METİN MADENCİLİĞİ 13
3.1. KULLANIM ALANLARI 14

Sayfa

3.2. METİN MADENCİLİĞİ ADIMLARI 14
3.2.1. Metin Koleksiyonu Oluşturma (Veri Seti) 15
3.2.2. Metin Ön İşleme 15
3.2.3. Sözcük Ağırlıklandırma 16
3.2.3.1. Bit Ağırlıklandırması 18
3.2.3.2. Sözcük Frekansı Ağırlıklandırması (*tf*) 19
3.2.3.3. Ters Doküman Ağırlıklandırması (*idf*) 19
3.2.3.4. Sözcük Frekansı-Ters Doküman Ağırlıklandırması (*tf-idf*) 20
3.2.4. Özellik Seçimi (Özellik Vektörünü Oluşturan Sözcüklerin Seçimi) 21
3.2.5. Vektör Uzay Modeli 22
3.3. BENZERLİKLERİN HESAPLANMASI 23
3.3.1. Euclid Mesafesi 23
3.3.2. Ağırlıklı Oylama 24
3.3.3. Cosine Benzerliği 24
3.4. METİN SINIFLANDIRMA 25
3.4.1. k-NN 26
3.4.2. Naive Bayes 27
3.4.2.1. Multi-Variate Model 27
3.4.2.2. Multi-Nominal Model 28

BÖLÜM 4 30
UYGULAMA 30
4.1. SİSTEM YAPISI 32
4.2. METİN KOLEKSİYONU OLUŞTURMA (VERİ SETİ) 33
4.3. VERİTABANI MODELİ 33
4.4. TANIMLARIN YAPILMASI 34
4.4.1. Gazete Tanımlamaları 35
4.4.2. Yazar Tanımlamaları 35
4.4.3. Sınıf (Kategori) Tanımlamaları 36
4.5. EĞİTİM DOKÜMANLARI (YAZILARI) İŞLEMLERİ 36
4.5.1. Dokümanların Alınması 38
4.5.1.1. Köşe Yazısı Bilgilerinin Alınması 38

Sayfa

4.5.1.2. Köşe Yazısı İçeriğinin Alınması 40
4.5.1.3. Köşe Yazısı Diğer İşlemleri 41
4.5.2. Metin Ön İşlem 43
4.5.2.1. İçeriğin HTML Etiketlerinden Temizlenmesi 43
4.5.2.2. İçeriğin Karakterlerden Temizlenmesi 44
4.5.2.3. İçerikteki Kelimelerin Köklerine Ayrılması 44
4.5.2.4. İçeriğin Gereksiz Kelimelerden Temizlenmesi 46
4.5.3. Kelimelerin Veritabanına Kaydedilmesi 47
4.5.4. Sözcük Ağırlıklandırma 49
4.5.5. Özellik Seçimi (Özellik Vektörünü Oluşturan Sözcüklerin Seçimi) 49
4.5.6. Dokümanların Vektörel İfadesi 57
4.6. TEST DOKÜMANLARI (YAZILARI) İŞLEMLERİ 62
4.7. METİN SINIFLANDIRMA 63
4.8. UYGULAMA SONUÇLARI VE DEĞERLENDİRİLMESİ 69
4.8.1. k-NN (*k=7*) / Bit Ağırlıklandırma 72
4.8.2. k-NN (*k=7*) / *tf-idf* Ağırlıklandırma 75
4.8.3. Naive Bayes / Multi-Variate / Bit Ağırlıklandırma 78
4.8.4. Naive Bayes / Multi-Variate / *idf* Ağırlıklandırma 80
4.8.5. Naive Bayes / Multi-Variate / *tf-idf* Ağırlıklandırma 81
4.8.6. Naive Bayes / Multi-Nominal / *tf* Ağırlıklandırma 84
4.8.7. Naive Bayes / Multi-Nominal / *idf* Ağırlıklandırma 86
4.8.8. Naive Bayes / Multi-Nominal / *tf-idf* Ağırlıklandırma 87

BÖLÜM 5 90
SONUÇ VE ÖNERİLER 90

KAYNAKLAR 93
ÖZGEÇMİŞ 99

ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.1. Cümle tablosu ve kelime kökleri tablosu eşleşmesi. 8

Şekil 2.2. KDD sürecinde yer alan adımlar..... 9

Şekil 3.1. Sözlük ve doküman tablolarının sözcük olarak gösterimi. 18

Şekil 3.2. Vektör uzay modeli..... 22

Şekil 3.3. Dokümanların *tf-idf* ağırlandırmayla vektörel ifadesi..... 23

Şekil 3.4. *k=3* ve *k=5* değerleri için k-NN sınıflandırması. 26

Şekil 4.1. Eğitim dokümanlarının hazırlanması işlemi basamakları..... 32

Şekil 4.2. Sınıflandırma işlemi basamakları. 32

Şekil 4.3. Veritabanı yapısı. 34

Şekil 4.4. Sınıf, yazar ve gazete tabloları ve ilişkileri..... 35

Şekil 4.5. Yazılım, işlem seçim ekranı..... 37

Şekil 4.6. Köşe yazısı işlemleri ekran görüntüsü ve bölgeleri. 38

Şekil 4.7. IV.Bölge açılır menü görüntüsü..... 40

Şekil 4.8. II.Bölge açılır menü görüntüsü. 41

Şekil 4.9. Sınıfa ait yazarların yazılarının alınması işlemi ekran görüntüsü..... 42

Şekil 4.10. Yazının, HTML etiketlerinden temizlenmesi. 43

Şekil 4.11. Kelimeyi fonksiyona gönderen ve fonksiyondan aldığı kökü ekrana yazdıran kod. 45

Şekil 4.12. Gereksiz kelime listesi. 47

Şekil 4.13. Kelime, yazı ve kelime-yazı tabloları ve ilişkileri. 48

Şekil 4.14. Kelime-yazı dağılımını gösteren tbl_yazi_kelime_EGITIM tablosu verilerinden bir görüntü..... 49

Şekil 4.15. Sınıfı *Ekonomi* olan özellik seçimi uygulanmamış *X, Y, Z* dokümanlarının *tf* ağırlandırılmış vektörel ifadesi. 52

Şekil 4.16. Sınıf özellik vektörünü oluşturan sözcüklerin seçiminde, kategori_id'si 1 (*Ekonomi*) olan sözcüklerin id değerini alan kod. 53

Şekil 4.17. Bütün sınıflarda, en fazla dokümanda geçen 175 sözcüğü alıp sınıf özellik vektörünü oluşturan kod..... 54

Şekil 4.18. Özellik seçiminde kullanılan ikinci yaklaşımın *Y* dokümanıyla eşleştirilmesi..... 58

Sayfa

Şekil 4.19. Sınıf özellik vektörüne göre X, Y, Z dokümanlarının bit ağırlıklı vektörel ifadesi. a)Birinci yaklaşım b)İkinci yaklaşım. 58

Şekil 4.20. Sınıf özellik vektörüne göre *X, Y, Z* dokümanlarının *tf* ağırlıklı vektörel ifadesi. a)Birinci yaklaşım b)İkinci yaklaşım. 59

Şekil 4.21. Sınıf özellik vektörünü oluşturan sözcüklerle eğitim dokümanlarını eşleştirip bit ağırlıklandırmasını gerçekleştiren kod. 61

Şekil 4.22. Ağırlıklandırılmış doküman vektörleri. a)bit, b)*tf-idf*, c)*tf*, d)*idf*. 62

Şekil 4.23. Sınıflandırma işlemi ekran görüntüsü. 66

Şekil 4.24. Bölgelerin ekran görüntüsü. a)5.Bölge, b)7.Bölge. 67

Şekil 4.25. Kullanılan benzerlik ölçütleri ve bit ağırlıklı sınıf özellik vektörlerine göre sınıflandırma sonuçları grafiksel gösterimi (k-NN). 74

Şekil 4.26. Kullanılan benzerlik ölçütleri ve *tf-idf* ağırlıklı sınıf özellik vektörlerine göre sınıflandırma sonuçları grafiksel gösterimi (k-NN). ... 77

Şekil 4.27. Multi-Variate model sınıflandırma işleminde sözcük kontrolü. 82

Şekil 4.28. Multi-Variate model ve bit, *idf*, *tf-idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırma sonuçlarının grafiksel gösterimi. 84

Şekil 4.29. Multi-Nominal model ve *tf, idf, tf-idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırma sonuçlarının grafiksel gösterimi. 89

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 3.1. Aynı köke sahip kelimeler örneği. 16
Çizelge 3.2. Düşündü kelimesi kökleri ve ekleri. 16
Çizelge 3.3. Sözlük tablosu. 17
Çizelge 3.4. Dokümanlarda geçen kelimeler tablosu. a) X, b) Y dokümanıdır. 18
Çizelge 3.5. Sözcüklerin *idf* ağırlıklandırılması. 20
Çizelge 3.6. Sözcüklerin dokümanlardaki *tf-idf* ağırlıklandırılması. 21
Çizelge 4.1. Sitelerden alınan köşe yazıları bilgileri. 33
Çizelge 4.2. Habertürk gazetesinin yazı bilgileri ve yazılar sayfası link formatları. 35
Çizelge 4.3. Sınıf (kategori) bilgileri. 36
Çizelge 4.4. Habertürk gazetesi, yazı bilgileri ifadeleri. 39
Çizelge 4.5. Zaman gazetesi yazı ifadeleri. 40
Çizelge 4.6. Küçük harfe çevrilmiş özel isimlerin kökleri. 44
Çizelge 4.7. Elde edilen kelimelerin (köklerin) yazılarda bulunma sayıları. 46
Çizelge 4.8. Sınıflara göre sözcük sayıları. 50
Çizelge 4.9. Sözcük sayılarına göre sınıflardaki yazı sayıları. 51
Çizelge 4.10. Sınıf özellik vektörlerinin oluşturulmasında kullanılan özellikler. 55
Çizelge 4.11. k-NN sınıflandırma bilgileri. 68
Çizelge 4.12. Naive Bayes sınıflandırma bilgileri. 68
Çizelge 4.13. Sınıf özellik vektörleri kullanılarak yapılan sınıflandırma işlemlerinde sınıfların ortalama başarıları. 71
Çizelge 4.14. Kullanılan benzerlik ölçütleri ve bit ağırlıklı sınıf özellik vektörlerine göre sınıflandırma sonuçları (k-NN). 73
Çizelge 4.15. Kullanılan benzerlik ölçütleri ve bit ağırlıklı sınıf özellik vektörlerine göre sınıfların ortalama sınıflandırma sonuçları (k-NN). 74
Çizelge 4.16. Kullanılan benzerlik ölçütleri ve *tf-idf* ağırlıklı sınıf özellik vektörlerine göre sınıflandırma sonuçları (k-NN). 76
Çizelge 4.17. Kullanılan benzerlik ölçütleri ve *tf-idf* ağırlıklı sınıf özellik vektörlerine göre sınıfların ortalama sınıflandırma sonuçları (k-NN). 78
Çizelge 4.18. Multi-Variate model ve bit ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırma sonuçları. 79

Sayfa

Çizelge 4.19. Multi-Variate model ve bit ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırmalarda, ortalama sınıf başarıları. 79

Çizelge 4.20. Multi-Variate model ve *idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırma sonuçları. 80

Çizelge 4.21. Multi-Variate model ve *idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırmalarda, ortalama sınıf başarıları. 81

Çizelge 4.22. Multi-Variate model ve *tf-idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırma sonuçları. 82

Çizelge 4.23. Multi-Variate model ve *tf-idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırmalarda, ortalama sınıf başarıları. 83

Çizelge 4.24. Multi-Nominal model ve *tf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırma sonuçları. 85

Çizelge 4.25. Multi-Nominal model ve *tf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırmalarda, ortalama sınıf başarıları. 85

Çizelge 4.26. Multi-Nominal model ve *idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırma sonuçları. 86

Çizelge 4.27. Multi-Nominal model ve *idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırmalarda, ortalama sınıf başarıları. 86

Çizelge 4.28. Multi-Nominal model ve *tf-idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırma sonuçları. 87

Çizelge 4.29. Multi-Nominal model ve *tf-idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırmalarda, ortalama sınıf başarıları. 88

SİMGELER VE KISALTMALAR DİZİNİ

SİMGELER

$arg\ max_{c_k}$: en yüksek değerli c_k sınıfı

c_k : k sınıfı

$C(X)$: X dokümanının atanacağı sınıfı

d : Euclid ölçümlerinde vektörler arası mesafe

d_s : bir sözcüğün bir sınıftaki dokümanlarda geçme sayısı

i : sayaç değişkeni

idf : ters doküman değeri

idf_{w_i} : w_i kelimesinin ters doküman değeri

g_s : kelimenin geçtiği doküman sayısı

$g\_s_{w_i}$: w_i kelimesinin geçtiği doküman sayısı

k : kategori sayısı

n : kelime sayısı

S_C : vektörlerin skaler çarpım sonucu

tf : sözcük frekansı

tf_{w_i} : w_i kelimesinin sözcük frekans ağırlığı

$tf\text{-}idf$: sözcük frekansı-ters doküman değeri

$tf\text{-}idf_{w_i}$: w_i sözcüğünün sözcük frekansı-ters doküman ağırlığı

t_s : bir sözcüğünün bir sınıftaki dokümanlarda geçme sayısı toplamı

V : eğitim dokümanı sayısı

w_i : i indisli sözcük

w_{w_i} : w_i sözcüğünün ağırlığı

wb_{w_i} : w_i sözcüğünün bit olarak ağırlığı

X : X dokümanı

Y : Y dokümanı

Z : Z dokümanı

KISALTMALAR

CSS : Cascading Style Sheets (Katmanlı Biçim Sayfası)

HTML : Hyper Text Markup Language (Zengin Metin İşaret Dili)

KDD : Knowledge Discovery in Databases (Veritabanlarında Bilgi Keşfi)

k-NN : k-Nearest Neighbor **(**k-En Yakın Komşu)

SQL : Structured Query Language (Yapısal Sorgulama Dili)

SVM : Support Vector Machines (DestekVektör Makineleri)

BÖLÜM 1

GİRİŞ

Bilgisayarın hayatımıza girmesiyle birlikte birçok işin kolaylaştığı, işlemlerin daha hızlı ve doğru yapıldığı bir gerçektir. Her alanda kullanılan bilgisayar, depolanan veri miktarını da arttırmıştır. Kişiler, işletmeler ve çeşitli kuruluşlar kendileriyle ilgili her türlü veriyi dosya olarak veya veritabanlarında saklamaktadırlar. Fakat bu verilerin fayda sağlaması için işlenmesi gerekmektedir. İşlenmeyen veriler veritabanı boyutunu arttırmaktan başka bir işe yaramazlar ve işlenmediği sürece anlamsız bir yığından öteye gidemezler (Han and Kamber, 2006).

Veri madenciliği, önceden bilinmeyen ve potansiyel olarak faydalı olabilecek, veri içindeki gizli bilgilerin çıkarılmasıdır (Frawley et al., 1991). Veri madenciliği yapısal veriler üzerinde çalışır. Fakat metin dosyaları yapısal olmayan verilerdir. Bu tür verilerin işlenebilmesi için yapısal hale dönüştürülmesi gerekir. Metin madenciliği bu problemlere çözüm olarak sunulan, metin formatındaki verileri kullanarak içerisindeki bilgileri gün ışığına çıkaran ve özellikle 2000'li yıllardan sonra ilginin giderek arttığı önemli bir alandır (Konchady, 2006).

Metin madenciliği, özel amaçlar için metinden bazı bilgiler çıkarmak adına, metinin analiz edilmesi işlemidir (Visa, 2001). Bu analiz işlemlerinden bir tanesi de tez çalışmasının konusu olan sınıflandırmadır. Metin sınıflandırma, önceden belirlenmiş sınıflara dokümanların atanması işlemidir (Mitchell, 1997).

Metin madenciliğinin çalışma alanı sadece bilgisayarımızda saklanılan dosyalardan ibaret değildir. E-mailler, bloglar, kişisel sayfalar, haber siteleri gibi internet ortamında bulunan verilerin de işlenmesi metin madenciliği teknikleriyle gerçekleştirilir. Metin madenciliğinin amacı bu tür verilerin, veri madenciliği tekniklerinin uygulanabileceği yapısal forma dönüştürülmesidir.

Liao et al. (2012) yaptıkları çalışmada; 2000-2011 yılları arasındaki veri madenciliği teknik ve uygulamaları incelenmiş, yayınlanmış makaleler ve yapılan çalışmalar kaynakları ile verilmiştir.

Bir diğer çalışmada, geçmiş fabrika verileri kullanılarak, halı üretim verimliliğini arttırmak için veri madenciliği teknikleri kullanılmıştır (Çiflikli ve Kahya-Özyirmidokuz, 2010).

Bao et al. (2012), yer bilimi verilerinin veri madenciliği teknikleriyle işlenmesiyle, petrol ve gaz rezervleri hakkında bilgi sahibi olunabileceği konusunu işlemişlerdir.

Hsu (2009) yaptığı çalışmada; giyim sektöründe endüstriyel standartların geliştirilmesi ve üretim-pazarlamanın arttırılmasına yönelik, standart dikimler yerine müşterilerin vücut ölçülerinin veri madenciliği teknikleriyle işlenerek elde edilen sonuçlar doğrultusunda dikimlerin yapılarak satışların arttırılması hedeflenmiştir.

Metin madenciliği ile metin konularının özetlenmesi ve benzer yazıların belirlenmesinin yapıldığı çalışmada, yüksek *idf* değerli sözcüklerle özetleme, Cosine benzerliğiyle benzer yazıları belirleme işlemini gerçekleştirilmiştir (Pons-Porrata et al., 2007).

Gieger et al. (2003) yaptıkları çalışma, günümüzde sağlık alanında metin madenciliği kullanımını öngören bir çalışma olduğunu göstermiştir. Sağlık alanında sıkça kullanılan metin madenciliğinin, gen klinik araştırmalarında kullanımının geleceği konusu işlenmiştir. Eldeki verilerin sadece yapılan deney ve gözlemlerden ibaret olmadığı, konuyla ilgili birçok bilimsel çalışma gerçekleştirildiği, bu çalışmaların metin verisi barındırdığı belirtilmiştir. Metinlerin konuyla ilgili olup olmadıklarının belirlenmesinin metin madenciliği teknikleri kullanılarak gerçekleştirilebileceği vurgulanmıştır.

Fuller et al. (2011), kandırma ve hilelerin veri ve metin madenciliği teknikleriyle bulunmasını incelemişler ve %74,00 oranında doğru tespit gerçekleştirmişlerdir.

Metin sınıflandırma, metin madenciliğinin uygulama alanlarından biridir. Metin sınıflandırma ile ilgili Türkçe dışındaki dillerde yapılan çalışmalar vardır (Cohen and Hirsh, 1998; Yang and Liu, 1999; Sebastiani, 2002).

Yang and Liu (1999) ve Mengle et al. (2007) yaptıkları çalışmada; k-NN (k-Nearest Neighbor), Naive Bayes ve SVM (Support Vector Machines) yöntemleri kullanılarak metin sınıflandırma performanslarının karşılaştırmışlardır. Çalışma sonucuna göre SVM ve k-NN'in Naive Bayes'e göre daha başarılı sınıflandırma yaptığı görülmüştür.

k-NN ve Naive Bayes'in bit ağırlıklandırma kullanılarak yapılan metin sınıflandırma işleminde, k-NN'in Cosine benzerliği ile birlikte uygulandığında, Naive Bayes'ten daha başarılı olmuştur (Soucy and Mineau, 2001).

Sanwaliya et al. (2010), *k* değerini 30,40, 50, 60, 70, 80 alarak sınıflandırma çalışması yapmışlar ve en yüksek sınıflandırmayı %90,64 başarıyla *k=50* değerinden elde etmişlerdir. Yine aynı çalışmanın başarı oranlarına bakıldığında aynı koşullarda yapılan Naive Bayes'in k-NN'den daha başarılı olduğu gözlemlenmiştir.

Li et al. (2011) yaptıkları çalışmada; küçük ve büyük boyutlu veri setleri kullanarak, test dokümanlarının 6 farklı sınıftan birine atanmasını hedeflemişlerdir. *tf* ağırlıklandırma, Naive Bayes ile birlikte kullanmışlardır. Büyük boyutlu veri setleri ile yapılan sınıflandırma işleminde, bütün sınıflarda daha başarılı sınıflandırma gerçekleştirmiştir.

Türkçe metinler üzerine de metin sınıflandırma ile ilgili çeşitli çalışmalar bulunmaktadır (Amasyalı ve Diri, 2006; Güran vd., 2009; Toraman vd., 2011).

Amasyalı ve Yıldırım (2004), 50 eğitim ve 25 test dokümanı kullanarak çalışma yapmışlar ve maksimum %76,00'lık başarı elde etmişlerdir.

Aşlıyan ve Günel (2010) yaptıkları çalışmada; k-NN ve En Yakın Komşu metotlarını farklı şekillerde uygulamışlar ve en başarılı sonucu %88,40 ile En Yakın Komşu metodunda elde etmişlerdir.

Metin sınıflandırmada boyut azaltmanın etkileri ve özellik seçimi konusunun işlendiği çalışmada, *tf-idf* olarak ağırlıklandırılmış sözcükler kullanılarak, iki farklı veri seti ile çalışılmıştır. Sınıflandırmada kullanılacak sözcük sayısı, orijinal metin sayısının %90,00'ından %10,00'una kadar kademeli olarak düşürülerek sonuçlar gözlenmiştir. Toplam sözcüklerin %10,00'u ile yapılan sınıflandırmada, başarının birinci veri setinde %6,25, ikinci veri setinde ise %11,39 arttığı gözlemlenmiştir (Durmaz ve Bilge, 2011).

Metinlerin sınıflandırılmasında olduğu gibi web sayfalarının sınıflandırılmasında da metin madenciliği yöntemleri kullanılmaktadır. Çünkü web sayfaları da metin içerikli veri barındırmaktadır ve bu verilerin, veri madenciliği tekniklerinin uygulanabileceği yapıya dönüştürülmesi işlemi metin madenciliği ile gerçekleştirilmektedir.

Yin et al. (2007), metin madenciliği tekniklerinin web verilerine uygulanmasıyla ilgili yaptıkları çalışmada, web verilerinin düz bir yazı gibi değerlendirilemeyeceğini, reklam gibi metin dışı veri barındırdığından veri madenciliği tekniklerinin web sayfalarına direkt uygulanmasının oldukça zor olduğu belirtmişlerdir.

12 sınıfa ayrılmış Çin web sayfaları, Naive Bayes kullanılarak sınıflandırılmıştır. En yüksek sınıflandırma %100,00 başarı ile sigorta sınıfından elde edilmiştir. Ortalama başarı ise %94,80'dir (Huang et al., 2007).

Ahmadi et al. (2011) yaptıkları çalışmada; içerik, profil ve görsel bilgiye göre web sayfalarının sınıflandırılmasını amaçlamışlardır. İçerik bilgisindeki geçen kelimeler ve kelime sayılarına; profil bilgisindeki link sayısı, meta taglar, sayfadaki çerçeve ve açıklama sayısına; görsel bilgi içerisindeki resimlerin ten renginin bulunduğu bölgelere göre sınıflandırma yapmışlardır. Sadece içeriğe göre yapılan sınıflandırmalardan farklı olarak görsel öğeler de kullanmışlardır.

Çeşitli sınıflandırma algoritmaları uygulanarak yapılan web sayfası sınıflandırması çalışmasında, sınıflarda sıkça geçen kelimeler için pozitif, seyrek geçen kelimeler içinde negatif ağırlık kullanarak özellik seçimi uygulanmıştır. Sonuçlara göre, en yüksek sınıflandırma başarısı %91,12'dir. Fakat çalışmada kullanılan özellik seçimiyle k-NN uygulandığında, %90,62 olan sınıflandırma %90,80'e çıkmıştır (Chen et al., 2006).

Liang et al. (2006), kimya ile ilgili web sayfalarının sınıflandırılmasını amaçlamışlardır. Kimya terimlerinden oluşan sözlük kullanılarak, k-NN ve Cosine benzerliği birlikte uygulanmıştır. Konuya ilişkin terimlerden oluşturulan sözlükle yapılan sınıflandırmaların, sınıftaki eğitim dokümanlarında geçen kelimelerle oluşturulan sözlükle yapılan çalışmalara göre daha başarılı olduğu gözlemlenmiştir.

Toraman vd. (2011) yaptıkları çalışmada; bit, *tf* ve *tf-idf* ağırlıklandırmalar, k-NN, SVM, C4.5 ve Naive Bayes ile birlikte uygulayarak, Türkçe web sitelerinin sınıflandırılması gerçekleştirmişlerdir. 2667 eğitim, 1219 test dokümanı ve 2442 eğitim, 1212 test dokümanı olan iki farklı veri seti kullanılarak yapılan çalışmada, k-NN ve Naive Bayes ile yapılan sınıflandırmalarda *tf-idf* ağırlıklandırmanın, bit ve *tf* ağırlıklandırmayla yapılan sınıflandırmalara göre daha başarılı olduğu görülmüştür.

Kendisiyle aynı kategorideki benzer haberlerin tespit edilmesini amaçlayan çalışmada, kelime köklerinin bulunması için Türkçe doğal dil işleme kütüphanesi Zemberek kullanılmıştır (Karadağ ve Takçı, 2010).

Bu tez çalışmasının amacı, metin madenciliği yöntemi ile haber sitelerindeki köşe yazılarının otomatik olarak alınıp sınıflandırılmasıdır. İnternet ortamında yayın yapan 6 farklı gazeteden 25 yazarın köşe yazıları ekonomi, spor, sağlık, eğitim ve yaşam olarak sınıflandırılması amaçlanmıştır. İşlemleri gerçekleştirmek için Visual Basic ile Visual Studio 2008 ortamında yazılım geliştirilmiştir.

Bu tez çalışması beş bölümden oluşmaktadır. İkinci bölümde veri madenciliği, üçüncü bölümde metin madenciliği, dördüncü bölümde köşe yazıları sınıflandırma uygulaması ve son olarak altıncı bölümde sonuçlar ve öneriler sunulmuştur.

BÖLÜM 2

VERİ MADENCİLİĞİ

Bilgisayarın hayatımızın her alanında kullanılıyor olması eldeki veri miktarının artmasına neden olmuştur. Bilgisayardaki herhangi bir dosyayı bulmak bile oldukça zaman alıcı ve zor bir iş iken, arama işleminin dosya adına yerine içeriğine göre yapılması, disk kapasitelerinin büyük boyutlara eriştiği günümüzde bu işlemin zorluk derecesini daha da arttırmaktadır. Eldeki veriler, veritabanlarında tutulan kayıtlardan ibaret değildir. Metin dosyaları ve internet ortamında bulunan içerikler gibi yapısal olmayan verilerin de işlenmesi gereklidir.

Kullanılabilen iş bilgilerinin %80'inden fazlası yapısal olmayan verilerden çıkarılmaktadır (Shilakes and Tylman, 1998). Eldeki verileri kullanarak anlamlı bilgiler çıkarmak kişi, işletme ve kuruluşları, diğerlerinden bir adım öne geçirmektedir. Veri madenciliği, büyük miktarlardaki verilerden fayda sağlayıcı bilgileri ortaya çıkararak veriye anlam kazandırma işlemidir (Han and Kamber, 2006). Veri madenciliği ile şirketler stratejik adımlar atarken çok büyük veri yığınları arasından kendilerine yol gösterecek kritik verileri ayıklayarak analiz edebilirler (Alpaydın, 2000).

Günümüzde veri madenciliğinin kullanıldığı bir çok alan vardır. Telefon şirketleri, aboneliklerini iptal eden müşterilerin iptal nedenlerini belirleyerek, var olan müşterileri elde tutmak için kullanmaktadırlar. Gelen bir mailin spam olup olmadığının belirlenmesi için kullanılmaktadır. Öğrencilerin başarılı veya başarısız olduğu derslerdeki ortak özelliklerini ortaya çıkarmak için kullanılmaktadır. Müşterilerin kredi kartı alış veriş hareketlerinin incelenerek kart hırsızlıklarında kartın kilitlenmesi için kullanılmaktadır. Bir hastalığa ait özelliklerin belirlenmesi suretiyle risk grubuna girenlerin belirlenmesinde kullanılmaktadır. Veri madenciliği, verilerden bilgi edilmek istenilen her yerde kullanılmaktadır.

Veri madenciliği ve KDD (Knowledge Discovery in Databases - Veritabanlarında Bilgi Keşfi) kavramları, her ne kadar farklı anlamlarda terimler olsa da aynı anlamda da kullanılmaktadırlar.

KDnuggets, 2011 yılında 228 oy kullanılarak elde edilen sonuçlara göre veri madenciliği kullanımı sektörel olarak şöyledir; %25 müşteri ilişkileri yönetimi, %18,9 bankacılık, %16,7 sağlık hizmetleri, %16,2 eğitim, %14 dolandırıcılık tespiti, %13,6 bilimsel çalışmalar, %13,2 sosyal ağlar ve %12,7 kredi derecelendirmesi (http://www.kdnuggets.com/polls/2011/industries-applied-anaytics-data-mining.html, 2011).

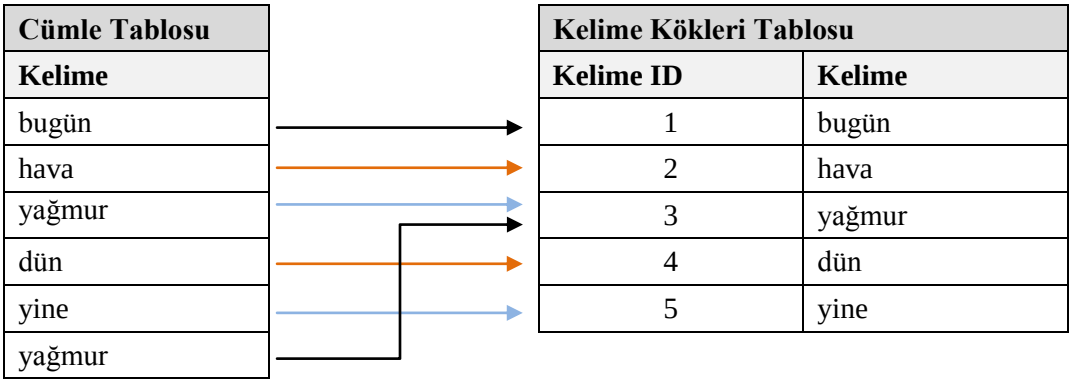
2.1. YAPISAL VERİ

Yapısal veri, üzerinde terim olarak işlem yapmaya olanak sağlayan verilerdir. Veritabanlarında tutulan veriler genelde yapısal verilerdir. Bilineceği üzere veritabanlarında çalıştırılacak sorgu deyimleriyle bazı bilgilere ulaşılabilir. Sorgu, eldeki verilerden belirli kısıtlara uyanların çekilmesi şeklinde ifade edilebilir. Sorgu sonucu elde edilen kayıtlarla ilgili olarak sorgu dili çıkan bilgileri yorumlamaz ve anlamlandırmaz. Verilerden anlam çıkarma işini veri madenciliği gerçekleştirir. Veri madenciliği, bir sorgu dili değildir.

Metin madenciliğinin amacı, yapısal olmayan verinin veri madenciliğinde kullanılacak yapısal veri haline dönüştürülmesidir (Dolgun vd., 2009; Karadağ ve Takçı, 2010). Yapısal olmayan veriler işlenmeden önce hazırlanmalı, yapısal veri haline getirilmelidir. Veri madenciliği, yapılandırılmış sayısal verilerle çalışırken metin madenciliği yapısal olmayan veri olan metinlerle çalışır. Metin madenciliği sonucunda, kategorilerin oluşturulması ile yapısal olmayan veri yapısal hale dönüşmektedir (Fan et al., 2005). Veri madenciliğinde veriler, metin madenciliği teknikleriyle elde edilir ve kelime köklerini temsil eden sayısal verilere dönüştürülerek yapısallık elde edilmiş olur.

Yapısal olmayan "Bugün hava yağmurlu. Dün yine yağmurluydu." cümlesinin öncelikle kelime kökleri bulunur ve "bugün, hava, yağmur, dün, yine, yağmur"

şeklinde ifade edilir. Veri madenciliğinde kullanılacak verilerin yapısal olması için verilerin sayısal olarak ifade edilmesi gerekir. Cümle, "bugün (1), hava (2), yağmur (3), dün (4), yine (5), yağmur (3)" kelimelerinden oluşan kelime kökleriyle değil, "(1, 2, 3, 4, 5, 3)" şeklinde kökü ifade eden sayılarla temsil edilmelidir. Bu uygulama Şekil 2.1.'de temsil edilmiştir.



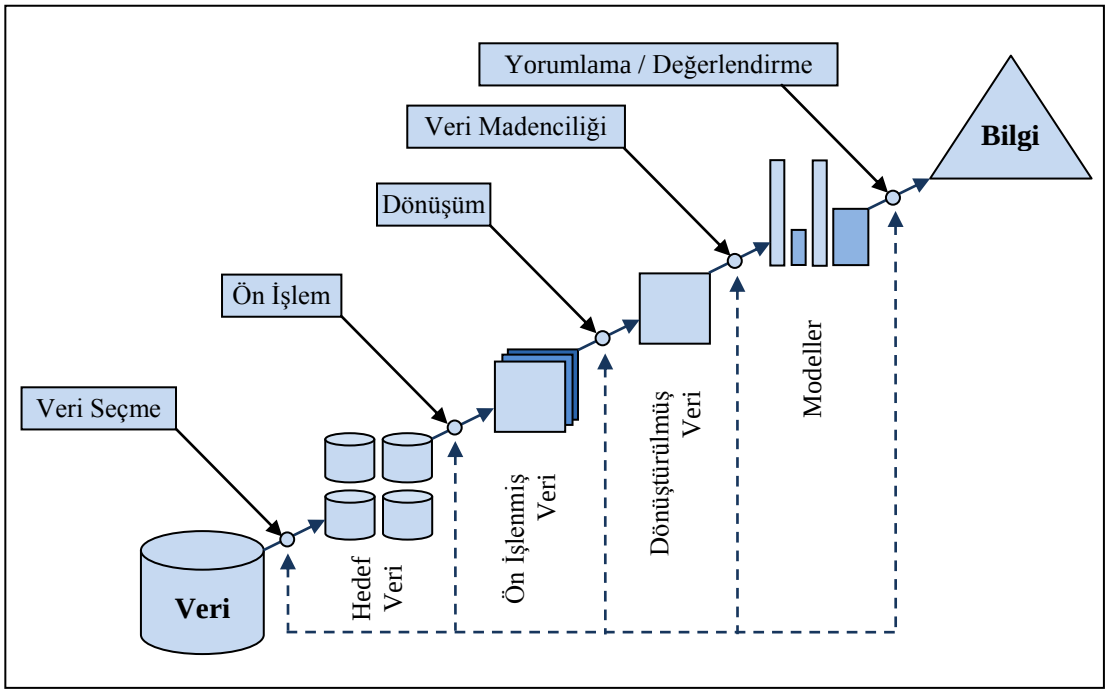
Şekil 2.1. Cümle tablosu ve kelime kökleri tablosu eşleşmesi.

2.2. KDD SÜRECİ

KDD, verideki geçerli, kullanılabilecek ve anlaşılır örüntüleri tanımlama işlemidir (Frawley et al., 1991). Veritabanlarında bilgi bulma, veriden faydalı bilginin keşfi işlemi için baştan sona kadar olan bütün aşamaları kapsadığı, veri madenciliğinin ise bu aşamalarda belirli bir adımı ifade ettiği belirtilmiştir (Fayyad et al., 1996).

KDD, ilk olarak 1989'da KDD konferansında bilginin, veriye dayalı keşfin son ürünü olduğunu vurgulamak için ortaya çıktı (Piatetsky-Shapiro, 1991). KDD, Şekil 2.2.'de gösterildiği gibi bir süreç olup verinin alınmasından bilgi çıkarmaya kadar olan bütün aşamaları ifade eder. Fakat günümüzde veri madenciliği ile veritabanlarında bilgi bulma aynı anlamda kullanılmaktadır.

KDD süreci, şu adımlardan oluşmaktadır; veri seçme, ön işlem, dönüşüm, veri madenciliği, yorumlama / değerlendirme. Ayrıca dönüşüm adımı da kendi içerisinde veri temizleme, verilerin bütünleştirilmesi, veri dönüştürme ve veri indirgeme basamaklarını barındırmaktadır.



Şekil 2.2. KDD sürecinde yer alan adımlar (Fayyad et al., 1996).

2.2.1. Veri Seçme

Bir şirkette tutulan binlerce kayıt bulunabilir ve bu kayıtlar sadece veritabanı dosyalarından ibaret değildir. Raporlar, yazışmalar, anketler, müşterilerden gelen mailler, şikayet formları da şirket kaydı olarak değerlendirilir. Sadece yapısal verilerden bilgi çıkarmak, yapısal olmayan verileri göz ardı etmek anlamına gelir. Fakat yapısal olmayan veri içeren kayıtlarda da gün ışığına çıkarılmayı bekleyen bilgiler vardır. Veri seçme, yapılacak analizler için veri yığınları içerisinden kullanılacak verilerin ayıklanmasıdır sürecidir.

2.2.2. Ön İşlem

Kullanılacak verilerin işlenmeden önce ön işlemden geçirilmesi gerekir. Ön işlem, veri madenciliği işlemlerinde veri içerisinde bilgi barındırmayan, temizlenmediğinde yanlış sonuçlar çıkmasına sebep olabilecek sözcüklerden arındırma ve sözcükleri bir sonraki aşama olan veriyi sayısal formata dönüştürme adımında kullanılabilecek yapı haline getirme işlemidir. Ön işlemde kendi içerisinde adımları barındırabilir. Ön işlem, KDD adımlarından en uzunu olabilir (Susan, 2001). Ön işlem aşaması

sonucunda elde edilen veriler ne kadar iyi hazırlanırsa, yapılacak madencilik çalışmalarından o kadar iyi sonuçlar alınır.

Web içeriklerinin ön işlemine örnek olarak şunlar verilebilir;

- Metinler üzerinde yapılan veri madenciliği çalışmalarında kelimelerin köklerine ayrılması.
- Web sayfaları üzerinde yapılan çalışmalarda HTML (Hyper Text Markup Language) etiketlerinden metinlerin temizlenmesi.

2.2.3. Dönüşüm

Eldeki verilerin dönüşümlerinin yapıldığı süreçtir.

Dönüşüm işlemleri şunlardır;

2.2.3.1. Veri temizleme

Eldeki veriden eksik, hatalı veya fazladan girilen verilerin çıkarılması işlemidir.

Örneğin, veritabanındaki cinsiyet alanı Erkek için "*E*", Kadın için "*K*" iken bunlar dışında veri girilmesi gürültü oluşturur ve bunların temizlenmesi gerekir.

Veritabanında doğum yılı ve yaş bilgilerini tutan alanlar varsa bu alanlardan bir tanesini bilerek diğeri de bulunabileceğinden sadece bir tanesinin dikkate alınması gerekir.

2.2.3.2. Verilerin Bütünleştirilmesi

Farklı veritabanlarından alınan verilerin ortak bir şekilde ifade edilmesi işlemidir.

Veritabanında bulunan tablodaki evlilik durumu bilgisi alanı verileri "Evli, Bekar, Dul" şeklinde bulunabileceği gibi "E, B, D", "0, 1, 2" gibi de bulunabilir.

2.2.3.3. Veri Dönüştürme

Verilerin direkt olarak veri madenciliği çalışmalarına katılması yanlış sonuçlar elde edilmesine neden olur. Sayılar üzerinde işlem yapılırken çok büyük sayılar sonucu daha çok etkileyecek, küçük sayıların sonucu etkilemesi çok daha az olacaktır. Bu işlemler için verilerin normalleştirilmesi gerekir. Çeşitli teknikler uygulanarak veriler normalleştirilir; min-max, Z-score.

Örneğin, bir tablodaki bir alanın 5-10, diğer alanın 1-10 000 arası değer aldığını düşünelim. Bu durumda verilerin sonuç üzerindeki etkisi farklı olacaktır. Bu verilere min-max normalleştirme uygulanması, bütün alanların sonucu aynı oranda etkilemesi, aynı alandaki verilerin 0-1 aralığındaki karşılıklarına dönüştürülmesi şeklinde gerçekleştirilir: 5-10 aralığındaki verilerde 5 değeri 0'a eşitlenirken 10 değeri 1'e eşitlenecek ve aradaki değerler belirlenecek, 1-10 000 aralığındaki verilerde ise 1 değeri 0'a eşitlenirken 10 000 değerş 1'e eşitlenecek ve aradaki değerler belirlenecektir. Bu sayede veriler, sonuç üzerinde aynı etkiyi göstermesi için normalleştirilmiş olacaktır.

2.2.3.4. Veri İndirgeme

Özellik seçimi adıyla da ifade edilir. Bu adımda eldeki verilerin tamamını veri madenciliği işlemlerinde kullanmak zaman kaybına neden olur. Bu verilerin işlemler için önemli olanlarının dikkate alınması veri indirgeme işlemidir.

Veri madenciliği sonuçlarından doğru sonuçlar elde edilebilmesi, verilerin hazırlanmasına bağlıdır.

2.2.4. Veri Madenciliği

Bu adıma kadar verilerin hazırlanması işlemi gerçekleştirilmiş olur. Artık verilere, veri madenciliği yöntemleri uygulanabilir. Hazırlanan veriler üzerinde çeşitli algoritmalar çalıştırılarak sınıflandırma, kümeleme veya birliktelikler çıkarılabilir. Sonuçlar bu adımda elde edilir.

2.2.5. Yorumlama / Değerlendirme

Veri madenciliği uygulandıktan sonra elde edilen verilerin beklenen veriler olup olmadığı ve uygun modelin seçilip seçilmediğinin değerlendirilmesi yapılır.

BÖLÜM 3

METİN MADENCİLİĞİ

İnternet ortamındaki doküman ve web sayfalarında büyük miktarlarda yapısal olmayan veri vardır. Tahminlere göre işletmeler, bilgilerinin %80'ini e-mail, iç yazışma, müşteri yazışmaları, raporlar, metin dokümanları şeklinde tutmaktadır (Wakil, 2002; Tan, 1999). Bu kullanılmamış kaynaklardaki bilgiyi çıkarmak, bilgi temelli ekonomi çağında işletmelere benzersiz rekabet fırsatı sunacaktır (Wakil, 2002).

Teknolojik gelişim ve internet kullanımı, bütün sektörlerin kendilerini bu duruma entegre etmelerini zorunlu hale getirmiştir. Zira globalleşen dünyada işletme ve çeşitli kuruluşlar büyümek için değil, ayakta kalabilmek için reklam bütçelerinden internet reklamlarına pay ayırmak zorunda kalmışlardır. Elbette bu durumu büyüme fırsatı olarak görenler işlerini daha da büyütüp, bütün işlemlerini (sipariş, ödeme, yazışma, bilgilendirme yazıları) internet ortamından gerçekleştirir hale gelmişlerdir.

İnternet kullanımın yaygınlaşması, internet ortamında o denli veri olduğu anlamına gelir. Bu verilerin veri madenciliği teknikleriyle incelenmesi yapısal veri olmadıklarından mümkün değildir. Metin veri madenciliği, metin koleksiyonlarından bilgiye erişen, bireysel metinlerden bilgi çıkaran, veritabanlarından bilgi keşfeden, organizasyonlarda bilgi yönetimini ve veri ile bilginin görselleştirilmesi aşamalarını birleştiren bir mimaridir (Losiewicz et al., 2000).

Metin madenciliği, veri madenciliğinin veya veritabanında bilgi keşfinin uzantısı olan farklı bir uygulaması olarak görülebilir (Fan et al., 2005). Metinlerin işlenebilir hale getirilmeleri için kullanılan bir uygulamadır. Metnin hazırlanması metin madenciliğiyle, işlenmesi ise veri madenciliği ile gerçekleştirilir.

Metin madenciliği, metin veri madenciliği veya doküman madenciliği diye adlandırılır.

3.1. KULLANIM ALANLARI

Metinsel veriler barındıran hemen her yerde kullanılabilir. Kullanımlarına örnek olarak şunlar verilebilir;

- Müşteri ilişkileri yönetimi,
- Sahtekarlık tespiti,
- Sağlık alanı,
- Pazar araştırmaları,
- Metinlerden bilgi çıkarımı,
- Doküman özetleme,
- Doküman sınıflandırma (Aşlıyan ve Günel, 2010; Gongde et al., 2006),
- Benzer içerikleri belirleme (Karadağ ve Takçı, 2010),
- Web içerikleri sınıflandırma (İlhan, 2001; Huang et al., 2007; Sanwaliya at al., 2010),
- Yazar tanıma sistemleri (Amasyalı ve Diri, 2006),
- Soru-cevap sistemleri (Erhardt et al., 2006).

3.2. METİN MADENCİLİĞİ ADIMLARI

Metin madenciliği adımları; kullanılacak verilerin alınması, ön işlemden geçirilmesi, metni temsil edecek kelimelerin seçilmesi ve vektör oluşturulması şeklinde ifade edilir.

Binlerce veya daha fazla metin içerisinden fayda sağlayıcı bilgilerin çıkarılması işlemi, metin madenciliği ile metinlerin hazırlanması ile mümkündür. Bu aşamada yapılanlar olumlu veya olumsuz olarak sonuçları etkileyeceğinden yapılan işlemlerde dikkatli olunmalıdır.

3.2.1. Metin Koleksiyonu Oluşturma (Veri Seti)

Veri seti kavramı, veriler içerisinden yapılacak çalışmada kullanılacak olanlarını ifade eder. Bilgisayarımızda kayıtlı dosyalar, mail kutumuzdaki mailler, bir şirketin her ay sonunda hazırladığı raporlar, forum sitelerinde yazılan yazılar, hastaların tahlil sonuçları, veri setine örnektir.

Örneğin, bir gazetedeki spor yazarlarının yazıları üzerine çalışma yapılacaksa, veri, bütün yazarların yazıları iken veri seti spor yazarlarının yazılarıdır. Hatta spor yazarlarının, 01.01.2012 ile 31.01.2012 tarihleri arasındaki yazılarıyla çalışma yapılacaksa veri seti spor yazarlarının bu aralıktaki yazılar olacaktır.

3.2.2. Metin Ön İşleme

Verilerin alındıktan sonra temizleme, bütünleştirilme, dönüştürme ve indirgenme işlemleri yapılır (Han and Kamber, 2006). Bu adım eldeki verilere göre farklılıklar gösterebilir: Soru-cevap sistemlerinde soru kelimeleri çok kullanıldığından bunların dahil edilmemesi; web sayfaları üzerinde işlem yapılırken HTML etiketlerinin temizlenmesi; kelimelerin küçük harfe çevrilmesi; kelime köklerinin elde edilmesi; noktalama işaretlerinin kaldırılması. Gereksiz kelimelerden metni arındırmak ve kelime köklerini bulmak bütün çalışmalarda yapılan ön işlem aşamalarıdır.

Türkçe yapı bakımından sondan eklemeli dillerdendir. Kelimeler ek almış veya almamış olarak bulunabilir. Çekim ve yapım eki olmak üzere iki ek vardır. Yapım ekleri alan kelimeler yeni anlam kazanırken çekim ekleri kelimenin anlamını değiştirmez.

Çalışmalarda kelime yerine kelime kökü kullanılır. Kelime kökleri ile çalışılmazsa aynı kelimeyi temsil eden farklı çekim eki almış kelimeler, farklı kelimeler gibi değerlendirilir. Böyle olunca hem sözcük hem vektör boyutu artmış olur hem de uygulamalarda yanlış sonuçlar elde edilir. Çizelge 3.1.'de aynı köke sahip çekim eki almış kelimeler görülmektedir.

Çizelge 3.1. Aynı köke sahip kelimeler örneği.

Kelime	Kök
gör	gör
gördüm	gör
görmüştüm	gör
görmedim	gör
görsem	gör

Kelime kökünün bulunmasında Zemberek kullanılır (http://code.google.com/p/zemberek/downloads/detail?name=zemberek-2.1.1.zip, 2011). Zemberek, Çizelge 3.2.'de gösterildiği gibi verilen kelimenin kökünü, tipini, aldığı ekleri ve ek türlerini veren açık kaynak kodlu Türkçe doğal dil işleme kütüphanesidir.

Çizelge 3.2. Düşündü kelimesi kökleri ve ekleri.

Kök	Tip	Aldığı Ekler
düşün	Fiil	FIIL_GECMISZAMAN_DI
düşün	İsim	IMEK_HIKAYE_DI
düş	İsim	ISIM_TAMLAMA_IN, IMEK_HIKAYE_DI
düş	İsim	ISIM_SAHIPLIK_SEN_IN, IMEK_HIKAYE_DI

3.2.3. Sözcük Ağırlıklandırma

Sözcük diye ifade edilen kelime kökleri, elde edildikten sonra sözcük ağırlıklandırma işlemine geçilir. Ağırlıklandırma işlemine sözcüklerin doküman üzerindeki etkisi de denilebilir (Karaca ve Görgünoğlu, 2012). Sözcükler, dokümanlarda bulunma durumlarına göre değer alıp, sayısal olarak ifade edilerek yapısallığı elde etmiş olurlar.

Bit, sözcük frekansı, ters doküman frekansı ve sözcük frekansı-ters doküman frekansı sözcük ağırlıklandırılmasında kullanılır. Eğer sözcük doküman içerisinde bulunuyorsa sözcüğün ağırlıklandırılmış değerini, bulunmuyorsa 0 değerini alır.

Metin madenciliği çalışmalarında, özellik seçimi bölümünde anlatılacağı üzere sözcüklerin tamamı yerine dokümanlardan bazı özelliklere göre seçilen sözcüklerle çalışmalar yapılır.

Çizelge 3.3.'te, dokümanlardan seçilen sözcükler sözlük tablosunda gösterilmiştir. Bu örnekteki sözcüklerin seçiminde herhangi bir kriter uygulanmamış, rastgele seçilmiştir ve dokümanlarda olmayan kelimelerde bulunmaktadır.

Çizelge 3.3. Sözlük tablosu.

Sözlük Tablosu	
Kelime ID	**Kelime**
1	*veri*
2	*metin*
3	*maden*
4	*içerik*
5	*köşe*
6	*yazı*
7	*benzerlik*
8	*eğitim*
9	*bilgisayar*
10	*mühendis*

Örnekte X ve Y adında iki doküman kullanılacaktır. Bu dokümanlarda yer alan sözcükler Çizelge 3.4.'te gösterilmiştir. Dokümanlar, sözlük tablosundaki sözcükleri bulundurma durumlarına göre ağırlıklandırılmış değerleriyle vektörel olarak ifade edilecektir.

Çizelge 3.4. Dokümanlarda geçen kelimeler tablosu. a) X, b) Y dokümanıdır.

(a)

Doküman1 (*X*) Tablosu	
Kelime ID	**Kelime**
1	*veri*
3	*maden*
1	*veri*
5	*köşe*
6	*yazı*
5	*köşe*
4	*içerik*
4	*içerik*
5	*köşe*

(b)

Doküman2 (*Y*) Tablosu	
Kelime ID	**Kelime**
2	*metin*
4	*içerik*
1	*veri*
9	*bilgisayar*
2	*metin*
1	*veri*
9	*bilgisayar*
10	*mühendis*

Sözlük ve doküman tablolarındaki sözcükler Şekil 3.1.de şu şekilde gösterilmiştir;

SÖZLÜK={"*veri*", "*metin*", "*maden*", "*içerik*", "*köşe*", "*yazı*", "*benzerlik*", "*eğitim*", "*bilgisayar*", "*mühendis*"}

X={"*veri*", "*maden*", "*veri*", "*köşe*", "*yazı*", "*köşe*", "*içerik*", "*içerik*", "*köşe*"}

Y={"*metin*", "*içerik*", "*veri*", "*bilgisayar*", "*metin*", "*veri*", "*bilgisayar*", "*mühendis*"}

Şekil 3.1. Sözlük ve doküman tablolarının sözcük olarak gösterimi.

3.2.3.1. Bit Ağırlıklandırması

Bit, boolean ve binary ağırlıklandırma olarak da isimlendirilir. Sözcüklerin dokümanda bulunup bulunmadığıyla ilgilenir. Sözcüğün alacağı değer 0 veya 1'dir. Eğer sözcük dokümanda bulunuyorsa 1, bulunmuyorsa 0 değerini alır. Dokümanda bulunan bütün sözcükler eşit değerdedir (Jackson and Moulinier, 2002). Sözcüğün doküman içerisinde bir kez geçmesiyle birden çok geçmesi arasında bir fark yoktur.

Örneğin, *veri* kelimesi, iki dokümanda da bulunduğu için 1 değerini alacaktır. Fakat *metin* kelimesi *X*'de bulunmadığından 0, *Y*'de bulunduğundan 1 değerini alacaktır.

3.2.3.2. Sözcük Frekansı Ağırlıklandırması (*tf*)

Bit ağırlıklandırma gibi sadece kendi dokümanıyla ilgilenir. Sözcüğün diğer dokümanlarda geçmesiyle ilgilenmez. Sözcük, doküman içerisinde kaç kez geçiyorsa geçme sayısıyla ağırlıklandırılır. Sözcüğün doküman içerisinde birden fazla geçmesi, o doküman için değerli olduğu anlamını çıkarır.

Örneğin *veri* kelimesi *X*'de 1 kez geçtiği için 1, *Y*'de 2 kez geçtiği için 2 değerini alacaktır. *metin* kelimesi *X*'de bulunmadığından 0, *Y*'de 2 defa bulunduğundan 2 değeriyle ağırlıklandırılacaktır.

3.2.3.3. Ters Doküman Ağırlıklandırması (*idf*)

Sözcüğü, bütün eğitim dokümanlarını inceleyerek, geçtiği doküman sayısına göre ağırlıklandırılır. Sözcüğün doküman için ne kadar belirleyici olduğunu ölçer. Eğer sözcük, sadece bir dokümanda geçiyorsa yüksek değerli, birçok dokümanda geçiyorsa düşük değerli olur. Bir sözcüğün bütün dokümanlarda geçmesiyle hiçbir dokümanda geçmemesi aynı değerdedir ve 0 olarak alınır.

w_i sözcüğünün ağırlığı olan idf_{w_i}, Eşitlik 3.1'de gösterildiği gibi hesaplanır. Bu eşitlikte eğitim doküman sayısı V ile w_i sözcüğünün geçtiği doküman sayısı $g\_s_{w_i}$ ile ifade edilir (Salton and Buckley, 1988).

$$idf_{w_i} = \log\left(\frac{V}{g\_s_{w_i}}\right) \tag{3.1}$$

Kelimelerin *idf* ağırlıkları Çizelge 3.5.'te verilmiştir. Görüldüğü gibi sözcük, bütün dokümanlarda geçiyorsa 0 değerini alırken 1 dokümanda geçtiğinde 0,30102 değerini almıştır. Hiçbir dokümanda geçmeyen sözcükler içinse değer hesaplaması yapılmamıştır.

Çizelge 3.5. Sözcüklerin *idf* ağırlıklandırılması.

Kelime ID	**Kelime**	**Toplam Doküman Sayısı**	**Geçtiği Doküman Sayısı**	***idf* Değeri**
1	*veri*	2	2	**0**
2	*metin*	2	1	**0,30102**
3	*maden*	2	1	**0,30102**
4	*içerik*	2	2	**0**
5	*köşe*	2	1	**0,30102**
6	*yazı*	2	1	**0,30102**
7	*benzerlik*	2	0	**-**
8	*eğitim*	2	0	**-**
9	*bilgisayar*	2	1	**0,30102**
10	*mühendis*	2	1	**0,30102**

3.2.3.4. Sözcük Frekansı-Ters Doküman Ağırlıklandırması (*tf-idf*)

idf ağırlıklandırmada sözcüğün dokümandaki frekansının (*tf*) önemi yoktur. Fakat *tf-idf* ağırlıklandırmada sözcüğün diğer dokümanlarda bulunma durumlarının yanı sıra bulunduğu dokümandaki frekansı da önemlidir. *tf-idf* fonksiyonunun çeşitli versiyonları mevcuttur (Soucy and Mineau, 2005).

tf-idf şu şekilde açıklanmıştır (Zobel and Moffat, 1998):

- Farklı dokümanlarda sık geçmeyen sözcükler sık geçenlere göre daha değerlidir.
- Bir dokümanda bir sözcüğün sık gözükmesi seyrek gözükmesinden daha değerlidir.
- Dokümanın uzunluğu sözcüğün değerini etkilememektedir.

w_i sözcüğünün ağırlığı olan w_{w_i}, Eşitlik 3.2'de gösterildiği gibi ifade edilir; (Salton and Buckley, 1988);

$$w_{w_i} = tf_{w_i} \times idf_{w_i} \tag{3.2}$$

Kelimelerin *tf-idf* ağırlıkları Çizelge 3.6.'da verilmiştir.

Çizelge 3.6. Sözcüklerin dokümanlardaki *tf-idf* ağırlıklandırılması.

Kelime ID	***idf* Değeri**	***X* *tf***	***X* *tf-idf***	***Y* *tf***	***Y* *tf-idf***
1	0	2	**0**	2	**0**
2	0,30102	0	**0**	1	**0,30102**
3	0,30102	1	**0,30102**	0	**0**
4	0	2	**0**	1	**0**
5	0,30102	2	**0,60204**	0	**0**
6	0,30102	1	**0,30102**	0	**0**
7	0	0	**0**	0	**0**
8	0	0	**0**	0	**0**
9	0,30102	0	**0**	1	**0,30102**
10	0,30102	0	**0**	1	**0,30102**

3.2.4. Özellik Seçimi (Özellik Vektörünü Oluşturan Sözcüklerin Seçimi)

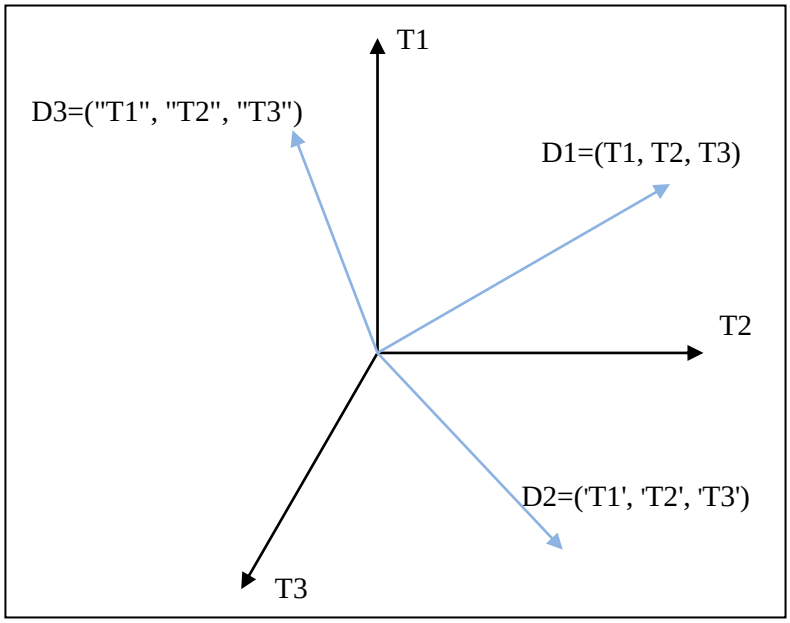
Sınıflandırıcının, kendisine hangi anahtar sözcüklerle sınıflandırma işlemine karar vereceğini sormaktadır. Bu anahtar sözcükler ile bütün eğitim dokümanlarını ele alarak hangisiyle daha çok ilişkili ise o yönde karar vermektedir (Soergel, 1985). Dokümanı temsil edecek kelimelerin seçimi veya diğer bir deyişle özellik seçimi kavramı, metin madenciliği çalışmalarında dokümanın hangi sözcüklerle ifade edileceğinin belirlendiği adımdır.

Metin dokümanlarının onlarca veya yüzlerce sayfadan oluşmaları, bu dokümanların tamamının yapılacak işleme dahil edilemeyeceğini göstermiştir. Özellik seçiminin amacı, sınıf özellik vektörünü oluşturacak sözcüklerin seçimini gerçekleştirmek, vektör boyutunu azaltmak ve metin hakkında bilgi verici niteliği bulunmayan sözcükleri çıkarmaktır (Yang and Pedersen, 1997).

Özellik seçiminin doküman vektör boyutunun azaltılması sonucu olarak işlem süresini düşürdüğü ve çalışmalarda daha doğru sonuçlar elde edilmesini sağladığı görülmüştür (Durmaz ve Bilge, 2011).

3.2.5. Vektör Uzay Modeli

Ön işlem, ağırlıklandırma ve özellik seçimi uygulandıktan sonraki aşama, dokümanların ağırlıklandırılmış sözcüklerle ifadesidir. Dokümanların, dokümanları oluşturan sözcüklerle ifade edilmesine vektör uzay modeli denir ve Şekil 3.2.'de gösterilmiştir (Salton et al., 1975). Bilgi çıkarımı, bilgi filtreleme, indeksleme gibi alanlarda kullanılan vektör uzay modeli, cebirsel bir modeldir (Pilavcılar, 2007).



Şekil 3.2. Vektör uzay modeli (Salton et al., 1975).

Şekil 3.1.'de, *tf-idf* ağırlıklandırma uygulanan dokümanların vektörel ifadesi Şekil 3.3.'te gösterilmiştir.

X= (2*0·0*0,30102·1*0,30102· 2*0·2*0,30102·1*0,30102·0*0·0*0·0*0,30102· 0*0,30102)

***X*=(0·0·0,30102·0·0,60204·0,30102·0·0·0·0)**

Y=(2*0·1*0,30102·0*0,30102·1*0·0*0,30102·0*0,30102·0*0·0*0·1*0,30102· 1*0,30102)

***Y*=(0·0,30102·0·0·0·0·0·0·0,30102·0,30102)**

Şekil 3.3. Dokümanların *tf-idf* ağırlandırmayla vektörel ifadesi.

3.3. BENZERLİKLERİN HESAPLANMASI

Sınıflandırma ve doküman benzerliği bulma işlemlerinde dokümanların birbirleriyle ne kadar ilişkili olduğunun tespit edilmesi dokümanların yakınlıklarıyla ilgilidir. Dokümanların benzerliklerini bulmak, vektörel ifadeleri kullanılarak gerçekleştirilir. Benzerlik, dokümanlar arası mesafenin ölçülmesi ve değerlendirilmesidir (Nanopoulos et al., 2001), Euclid mesafe ölçümü ve Cosine benzerliği en çok kullanılan benzerlik tespit yöntemleridir.

3.3.1. Euclid Mesafesi

$X = \{x_1, x_2, x_3, ..., x_n\}$ ve $Y = \{y_1, y_2, y_3, ..., y_n\}$ iki vektör olmak üzere bu iki vektör arasındaki mesafe olan *d*'nin hesaplanması Eşitlik 3.3'te formüle edilmiştir (Han and Kamber, 2006; Hand et al., 2001). *d* değerinin düşüklüğü vektörlerin birbirlerine yakın olduğunu belirtir.

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3.3}$$

Euclid için bir sözcüğün iki vektörde bulunması ile iki vektörde bulunmaması aynı değerdedir. Euclid, bir sözcüğün sadece iki vektörde bulunmasıyla değil ikisinde bulunmamasıyla da ilgilenir. Dokümandaki bütün sözcükler dikkate alınarak yapılan sınıflandırma çalışmalarında düşük performanslıdır.

3.3.2. Ağırlıklı Oylama

k-NN, *k* komşu içerisinden en fazla tekrar eden sınıfla ilgilenir. Bu durumda komşular içerisinden çok yakın veya çok uzak eğitim dokümanı arasında değerlendirme açısından bir farklılık yoktur, ikisi de karar vermek için eşit değerdedir. Ağırlıklı oylama, Eşitlik 3.4'te formüle edildiği gibi uzaklık değerlerinin karelerinin çarpmaya göre tersi alınıp sınıf bazında toplanarak elde edilen en yüksek değere sahip sınıfın doküman sınıfı olarak belirlenmesi ilkesine dayanır. Bu şekilde *k* komşu içerisindeki en fazla tekrarlayan sınıf değil en yüksek ağırlığa sahip sınıf doküman sınıfı olarak atanır.

$$d_w = \frac{1}{d^2} \tag{3.4}$$

3.3.3. Cosine Benzerliği

$X = \{x_1, x_2, x_3, ..., x_n\}$ ve $Y = \{y_1, y_2, y_3, ..., y_n\}$ iki vektör olmak üzere bu iki vektörün benzerliği *s* şeklinde Eşitlik 3.5'te formüle edilmiştir (Hand et al., 2001). *s* değerinin büyüklüğü vektörlerin birbirlerine yakın olduğunu belirtir. *s* değeri, iki doğru arasındaki açının kosinüsüdür. *s* değeri ile iki doğru arasındaki açı değeri ters orantılıdır.

s benzerlik değeri skaler çarpımların iki vektörün normlarının çarpımlarına oranıyla bulunur (Salton and Buckley, 1988). *s*, maksimum 1 değerini alır. *s* değerinin 1'e yakınlığı vektörlerin birbirine benzediğini gösterirken 0'a yakınlığı vektörler arasında ortak sözcüğün az olduğunu ve vektörlerin benzerliğinin düşük değerde olduğunu gösterir.

$$s = \frac{\sum_{i=1}^{n} X_i Y_i}{\sqrt{\sum_{i=1}^{n} (X_i)^2} \sqrt{\sum_{i=1}^{n} (Y_i)^2}} \tag{3.5}$$

Eşitlik 3.6'da, vektörlerin skaler çarpımlarını ifade etmektedir. Eğer bir sözcük bir vektörde var diğer vektörde yoksa skaler çarpım sonucu o sözcük için 0'dır.

$$S\_C = \sum_{i=1}^{n} X_i Y_i \tag{3.6}$$

Cosine, sözcüğün vektörde varlığını araştırır. İki vektörde de bulunan sözcük değerlendirmeye alınır. Bu durum, Euclid'in aksine, dokümandaki bütün sözcükler alınsa bile sözcük varlığını araştırdığı için daha doğru sonuçlar üretir.

3.4. METİN SINIFLANDIRMA

Metin sınıflandırma, önceden belirlenmiş sınıflara doküman atamayı hedefler (Mitchell, 1997). Sınıflandırma yapılmadan önce sınıfların belirlenmesi gerekir. Dokümanların ağırlıklandırılmış değerli vektörel ifadeleri kullanılarak elde edilen benzerlik ölçüm sonuçlarına ve uygulanan algoritmaya göre sınıflandırılması gerçekleştirilir. Metin sınıflandırma, doğal dil metinleriyle çalışan bir sınıflandırmadır (Soucy and Mineau, 2001).

Metin dokümanlarının uzunluğu metin madenciliği çalışmalarındaki en büyük sıkıntıdır. Dokümanların ön işlemden geçirilmesi ve özellik seçimi uygulanması boyutu azaltarak bu sıkıntıyı gideren işlemlerdir.

Makine öğrenmesine ihtiyaç duyulmasının nedeni, el ile kategorizasyonun pahalı ve zaman tüketen bir iş oluşudur ki, ayrıca elle sınıflandırmada, sınıflandırmayı yapan uzmanların vermiş oldukları kararlara bağlı olarak sonuçlar da değişmektedir (İlhan, 2001). Bu sebeple otomatik metotlar, algoritmalar ve büyük miktarlardaki verilerle çalışan araçlar önemli bir hale gelmiştir (Lagus, 2000).

Metin sınıflandırma işlemlerinde eğitim dokümanları vardır. Sınıflandırma yapılırken bu eğitim dokümanları kullanılarak sonuca gidilir. Eğitim dokümanlarının sınıfları, sınıflandırma işlemlerinin karar vermesine yardımcı olur ve özellik seçimindeki sözcükler bu dokümanlardan seçilir.

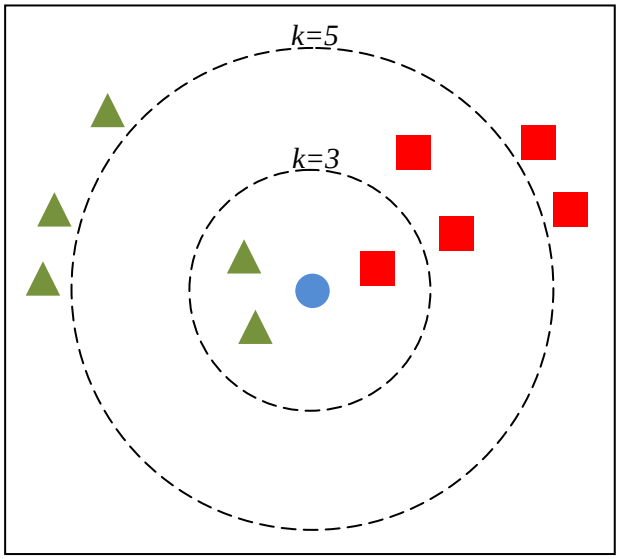
Kullanılan eğitim doküman sayısının azlığı (Toraman vd., 2011), haber metinlerinin kısalığı, her haberin farklı konulardan bahsetmesi sınıfı yansıtan sözcüklerin tespit edilememesine neden olur ve sınıflandırma başarısını düşürür.

Metin sınıflandırma işleminde k-NN, Naive Bayes, SVM ve yapay sinir ağları en çok kullanılan sınıflandırma yöntemleridir.

3.4.1. k-NN

k-NN algoritması ile sınıflandırma, önceden belirlenmiş *k* değerine göre uzaklıkları hesaplanmış eğitim dokümanları içerisinden en yakın *k* dokümandaki en yüksek frekansa sahip sınıfa göre test dokümanının sınıfını belirleme işlemidir (Dasarathy, 1991; Han and Kamber, 2006). Bütün eğitim dokümanları ile test dokümanlarının uzaklıkları tek tek hesaplanır ve belirlenecek *k* değerine göre sınıflandırma sonucuna karar verilir.

Şekil 3.4.'te görüldüğü gibi *k=3* alındığında dokümanın sınıfı üçgen olacakken *k=5* alındığında kare olmuştur. *k* değerinin yüksek seçilmesi benzemeyen dokümanların işleme dahil edilmesine, düşük seçilmesi benzeyenlerin dahil edilmemesine neden olur.



Şekil 3.4. *k=3* ve *k=5* değerleri için k-NN sınıflandırması.

Çoğunluğun seçimi ilkesine dayanan k-NN algoritmasının dezavantajı, eğitim dokümanlarındaki dengesizliktir: Bir sınıfa ait eğitim dokümanı sayısının başka bir sınıftaki doküman sayısından fazla olmasıdır (Coomans and Massart, 1982; Sanwaliya, 2010). Bu durumda, diğerlerine göre daha fazla sayıda eğitim dokümanı bulunduran sınıftan *k* içerisine doküman girme olasılığı artmaktadır.

3.4.2. Naive Bayes

Kolay uygulanabilmesi ve başarılı sonuçlar üretmesi nedeniyle en çok tercih edilen sınıflandırma metotlarından biridir. Hesaplamalar, sınıflar düzeyinde gerçekleştirilir. Her bir sınıf için olasılık değeri hesaplanarak en yüksek olasılık değerine sahip sınıf, sınıflandırılması yapılacak dokümanın sınıfıdır.

Sınıflandırma işleminde, her bir sözcük sınıftan bağımsızdır (Eyheramendy et al., 2003). Ayrıca sözcükler, aynı değerde olup birbirinden bağımsız olduğu kabul edilerek hesaplama yapılır.

3.4.2.1. Multi-Variate Model

Sınıflandırma işlemlerinde her sınıf ayrı olmak üzere, sınıfı oluşturan dokümanların sözcükleriyle işlem yapılır. Diğer sınıfların dokümanları ve sözcükleri hesaplama işlemlerinde dikkate alınmaz. Sözcüğün, sınıfın dokümanlarında geçip geçmediğiyle ilgilenir.

$X = \{x_1, x_2, x_3, ..., x_n\}$ sınıflandırılması yapılacak olan vektördür. Eşitlik 3.7'de w_i sözcüğünün c_k sınıfında geçme olasılığı formüle edilmiştir (Eyheramendy et al., 2003). $g\_s_{w_i}$, w_i sözcüğünün c_k sınıfındaki dokümanlarda geçme sayısı; d_s, c_k sınıfındaki toplam doküman sayıdır. Eşitlik 3.7'de bulunan paydaki 1 ve paydadaki 5 ifadeleri bir sözcüğün sınıf içerisinde geçmemesi durumunda sonucu 0 yapmasını engellemek içindir (Roiger and Geatz, 2003). Bütün sınıflardaki doküman sayıları eşit ve 5 sınıf olduğu için pay 1, payda 5 verilmiştir ve bu sonucu etkilememektedir. Çeşitli sıfır olasılık sorunu gidericiler kullanılmıştır (Eyheramendy et al., 2003).

$$p(w_i|c_k) = \frac{1 + g\_s_{w_i}}{5 + d\_s} \tag{3.7}$$

Eşitlik 3.8'de gösterilen formül Eşitlik 3.9'daki gibi revize edilmiştir. *X* vektörünün c_k sınıfında olma olasılığı ve test dokümanının sınıfı olan *C(X)* formüle edilmiştir. En yüksek $p(X|c_k)$ değerine sahip sınıf test dokümanının sınıfıdır (Eyheramendy et al., 2003). wb_{w_i}, w_i sözcüğünün bit olarak ağırlığını ifade etmektedir.

$$C(X) = \arg\max_{c_k} p(X|c_k) = \arg\max_{c_k} \prod_{i=1}^{n} p(w_i|c_k)^{wb_{w_i}} (1 - p(w_i|c_k))^{1-wb_{w_i}} \tag{3.8}$$

$$C(X) = \arg\max_{c_k} p(X|c_k) = \arg\max_{c_k} \sum_{i=1}^{n} \log(p(w_i|c_k)^{wb_{w_i}} (1 - p(w_i|c_k))^{1-wb_{w_i}}) \tag{3.9}$$

3.4.2.2. Multi-Nominal Model

$X = \{x_1, x_2, x_3, ..., x_n\}$ sınıflandırılması yapılacak olan vektördür. Eşitlik 3.10'da w_i sözcüğünün c_k sınıfında geçme olasılığı formüle edilmiştir (Eyheramendy et al., 2003). $g\_s_{w_i}$, w_i sözcüğünün c_k sınıfında geçme sıklığını; *t_s*, c_k sınıfındaki toplam sözcük sayısını; *V*, c_k sınıfındaki toplam doküman sayısını ifade eder.

$$p(w_i|c_k) = \frac{1 + g\_s_{w_i}}{V + t\_s} \tag{3.10}$$

X vektörünün c_k sınıfında olma olasılığı, her sınıfta eşit sayıda doküman olduğundan Eşitlik 3.11'de gösterilen formül revize edilerek Eşitlik 3.12'deki gibi formüle edilmiştir. $p(c_k)$, c_k sınıfının olasılığını; *k* ise toplam kategori sayısını ifade etmektedir. Olasılığı gösteren en yüksek değerli $p(X|c_k)$, test dokümanının sınıfıdır ve *C(X)* şeklinde ifade edilir.

$$C(X) = \arg\max_{c_k} \; p(X|c_k) = \arg\max_{c_k} \; p(c_k) k! \prod_{i=1}^{n} \frac{p(w_i|c_k)^{tf_{w_i}}}{tf_{w_i}!} \tag{3.11}$$

$$C(X) = \arg\max_{c_k} \; p(X|c_k) = \arg\max_{c_k} \sum_{i=1}^{n} \frac{\log(p(w_i|c_k))^{tf_{w_i}}}{tf_{w_i}!} \tag{3.12}$$

BÖLÜM 4

UYGULAMA

Türkiye'de gazetecilik, teknolojik gelişmelerden yararlanan alanlardan bir tanesidir. Dünyada ilk olarak The Washington Times, New York Times gibi gazeteler 1995 yılında internette yayın yapmaya başlamışlardır. Türkiye'de ilk olarak Zaman gazetesi, 2 Aralık 1995 tarihinde haberi ve köşe yazılarını başlıklar halinde internet ortamında sunmuştur. 27 Kasım 1996 tarihinde ise Milliyet, gazetenin tamamını internet ortamında veren ilk günlük gazete olmuştur. 1997 yılında ise yüksek tirajlı gazeteler okuyucularına sanal ortamdan ulaşmaya başlamışlardır.

Günümüzde, insanların yoğun iş temposu ve bilgisayar kullanımının artması, haber takibinde ve köşe yazarlarının yazılarını okunmasında internet gazeteciliğini önemli bir yere getirmiştir. Fakat internet gazeteciliği sadece geleneksel gazetecilik hizmeti verenlerle değil geleneksel gazetecilik yapmayan (sadece internetten yayın yapan) haber siteleriyle de sürdürülmektedir. İnternet gazetelerinin bu kadar rağbet görmesi bu alanda çalışmalar yapılmasını gerektirmiştir. İnternet gazetelerine olan ilginin başlıca sebepleri şunlardır; içerik farklılığı (metin, görüntü, resim, ses barındırması), istenilen yerden ulaşılabilmesi, arşivlere erişim kolaylığı, son dakika haberlerinin anında yayınlanması, detaylara izin vermesi, özgürleştirici olması, haberlere yorum yapılabilmesi. İnternet gazetelerinin geleneksel medyadan ayıran en önemli dört özelliği ise hızı, geri dönülebilir olması, detaylara izin vermesi ve yayıncı ve okuyucu açısından özgürlükçü olmasıdır (Dilmen, 2005).

Yapılan araştırma, kişilerin %87,90'ının bilgisayar sahibi olduğunu, %83,50'sinin bilgisayarı evde, %66,80'inin işyerinde kullandığını, %85,00'inin haber sitelerini takip ettiğini ortaya koymuştur (Çalışkan, 2008). İnternetin günlük hayatımızın önemli bir parçası haline gelmesi, bu ortamdaki veriler üzerinde çalışmalar yapılmasını gerekli kılmıştır.

İnsanlar, haber sitelerini sadece haber edinme amaçlı kullanmamaktadırlar. Farklı gazetelerdeki köşe yazarlarının takip edilmesi internet gazeteciliği ile daha rahat bir hal almıştır. Dahası köşe yazarlarının günlük yazılarının taranarak oluşturulduğu web siteleri vardır. Bu işlem, insanların veriye erişmesinde rahatlık imkanı kazandırmıştır. Kişilerin aradığı bilgiye rahat bir şekilde ulaşabilmeleri için verilerin filtrelenmesi, yönetilmesi ve sınıflandırılması gerekmektedir (Aas and Eikvil, 1999).

İnternetteki yazıların sınıflandırılması, kişilerin yazıyı okumadan bilgi sahibi olmaları açısından önemlidir. Bilgisayar sistemlerinden önce sınıflandırma işlemi, manüel olarak gerçekleştirilmekteydi ve bu işlem hem yavaş, hem pahalı hem de tutarlı değildi (Hayes et al., 1988). Veri miktarı düşünüldüğünde, günümüzde sınıflandırma işlemlerinin manüel olarak yapılması pekte mümkün görülmemektedir. Sınıflandırma işleminin otomatik olarak yapılması, insanlara bilgiye erişimde hız, kolaylık ve rahatlık gibi avantajlar sağlamıştır.

Türkçede haber metinleri sınıflandırma ile ilgili çalışmalar vardır (Amasyalı ve Yıldırım, 2004; Toraman vd., 2011). Fakat internet gazeteciliğinin önemli bir unsuru olan köşe yazılarının içerik bakımından sınıflandırılmasıyla ilgili bir çalışma olmaması ve başarılı sonuçlar elde etmiş olması bu çalışmayı önemli bir hale getirmiştir.

Ülke gündeminin değişken olması, köşe yazarlarının alanları dışında yazıyor olmaları, insanları yazıları okumadan hangi alanla ilgili yazı olduğunu bilmeleri, bilgi edinmenin yanında bilgiye erişim hızının da önemli bir unsur olduğunu ortaya koymaktadır.

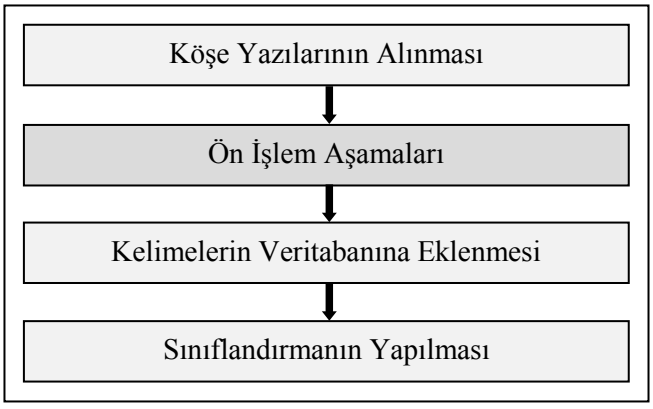
Bu çalışmanın gerçekleştirilmesi için Visual Studio 2008 ortamında Visual Basic dili ile hazırlanmış yazılım ve Access veritabanı kullanılmıştır.

Verilerin alınması, ön işlemden geçirilmesi, sözcük ağırlıklandırmaları, özellik seçimi, vektör oluşturulması, benzerliklerinin hesaplanması ve sınıflandırma işlemi kodlar aracılığıyla gerçekleştirilmiştir.

4.1. SİSTEM YAPISI

Şekil 4.1. Eğitim dokümanlarının hazırlanması işlemi basamakları.

Şekil 4.1., sistemin eğitilmesi işleminin işlem basamaklarını göstermektedir. Sistem önce eğitilmeli, eğitim dokümanlarının sınıflandırma işleminde karar vericiye yol gösterecek şekilde hazırlanması gerekmektedir.



Şekil 4.2. Sınıflandırma işlemi basamakları.

Şekil 4.2., sınıflandırma işleminin işlem basamaklarını göstermektedir. Sınıflandırılacak yazı alınır, ön işlemden geçirilir ve algoritma çalıştırılıp eğitim dokümanlarının yardımıyla sınıflandırma işlemi gerçekleşir.

4.2. METİN KOLEKSİYONU OLUŞTURMA (VERİ SETİ)

Veri seti, günlük yayın yapan gazetelerin sitelerinden alınmıştır. 6 farklı gazeteden 25 yazarın yazıları kullanılarak çalışma gerçekleştirilmiştir. Çizelge 4.1.'de, köşe yazılarının alınmaya başlandığı ve bittiği tarihler ile alındığı gazetenin bilgileri verilmiştir.

Çizelge 4.1. Sitelerden alınan köşe yazıları bilgileri.

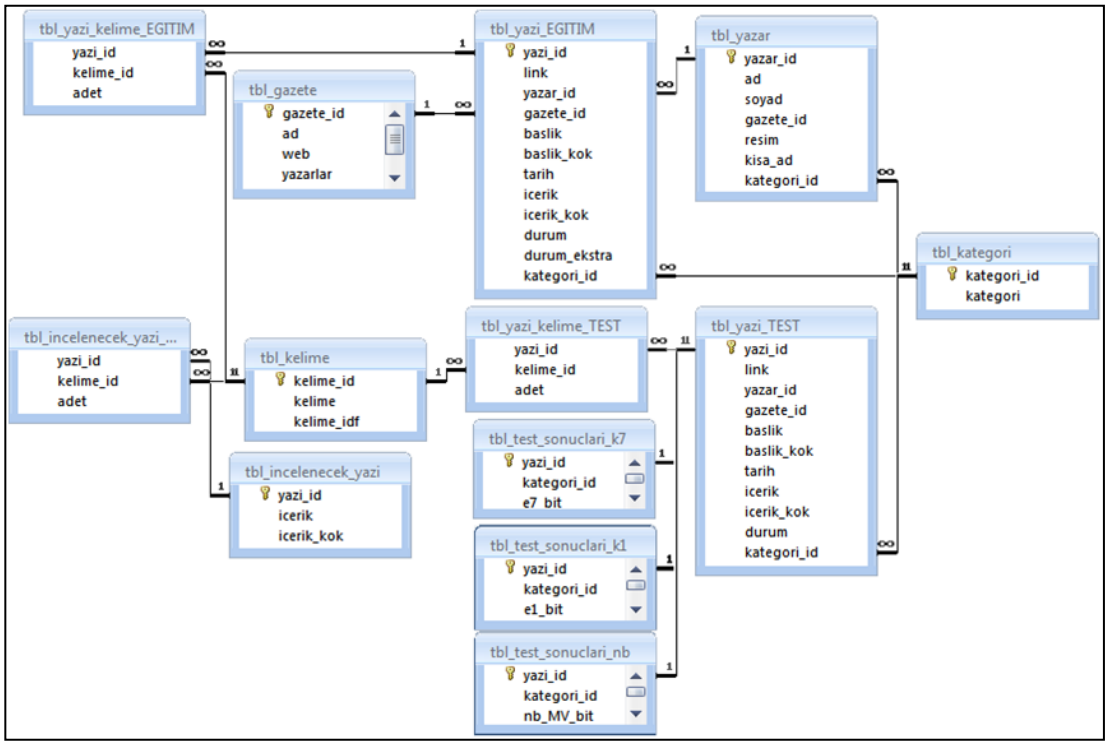
Gazete ID	**Gazete Adı**	**Başlangıç**	**Bitiş**
1	Posta Gazetesi	26.07.2011	15.02.2012
2	Milliyet Gazetesi	09.02.2011	22.02.2012
3	Habertürk Gazetesi	28.02.2011	27.02.2012
4	Zaman Gazetesi	04.06.2011	22.02.2012
5	Akşam Gazetesi	07.10.2010	22.02.2012
6	Star Gazetesi	24.07.2011	22.02.2012

Veri seti, 500 eğitim ve 250 test dokümanından oluşur. Sistemin eğitilmesinde eğitim, test işleminde test dokümanları kullanılmaktadır. Her sınıfta 100'er eğitim ve 50'şer test dokümanı bulunmaktadır. Yazarların, sitede bulunan bütün yazıları eklenerek eğitim ve test dokümanı olarak kullanılabilir.

Her ne kadar sisteme, yapılacak sınıflandırma çalışmalarında kullanılmak üzere 250 test dokümanı alınmış olsa da yazılım ara yüzü kullanılarak 25 yazara ek olarak tanımlamaları sisteme yapılan 25 yazarla farklı yazarların yazıları da takip edilip sınıflandırma çalışmaları yapılabilmektedir.

4.3. VERİTABANI MODELİ

Çalışmada kullanılan veritabanının yapısı, tablolar ve ilişkileri Şekil 4.3.'te görüntülenmektedir. Test dokümanlarının sınıflandırılma sonuçları, bütün sınıf özellik vektörleri için ayrı ayrı olarak *tbl_test_sonuclari_k7 (*k-NN, *k=7* değeri için) ve *tbl_test_sonuclari_nb* tablolarında (Naive Bayes) tutulmaktadır.

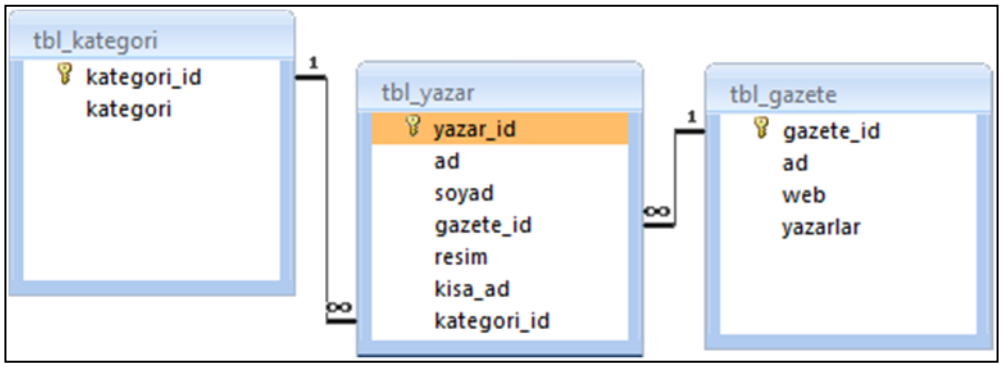


Şekil 4.3. Veritabanı yapısı.

4.4. TANIMLARIN YAPILMASI

Gazetelerin internet sayfalarının farklı HTML etiketleriyle meydana gelmeleri, verilerin her gazetede farklı şekilde sunulmaları anlamına gelir. Verilerin sitelerden alınması, bu farklılıklar nedeniyle standart değildir ve her bir gazete için ayrı tanımlamalar yapılmasını gerektirmiştir.

Sistemde bulunacak gazete, yazar ve sınıf tanımlamaları yapılarak gazetelerden veri alma işleminin standartlaştırılması sağlanmıştır. Tanımların tutulduğu tablolar ve tablolar arası ilişki Şekil 4.4.'te gösterilmiştir.



Şekil 4.4. Sınıf, yazar ve gazete tabloları ve ilişkileri.

4.4.1. Gazete Tanımlamaları

tbl_gazete tablosu, gazetenin id, ad, web ve yazarlar bilgilerini tutmak için oluşturulmuştur. Gazetelerin, yazı bilgileri (köşe yazısının link, başlık, içerik ve tarih bilgisi) ile yazı içerikleri (köşe yazısı) farklı link yapısında bulunmaktadır. Şekil 4.5.'teki *tbl_gazete* tablosundaki web alanı yazı içeriklerini, yazarlar alanı ise yazı bilgilerinin görüntülenmesi için kullanılmaktadır.

Çizelge 4.2.'de Habertürk gazetesinin yazı bilgileri ve yazı içeriklerinin elde edilmesi için kullanılan link formatları bulunmaktadır.

Çizelge 4.2. Habertürk gazetesinin yazı bilgileri ve yazılar sayfası link formatları.

Yazarlar Sayfası (Yazı Bilgileri)	`http://www.haberturk.com/htyazar/kisa_ad/`
Yazılar Sayfası	`http://www.haberturk.com/yazarlar/kisa_ad/`

4.4.2. Yazar Tanımlamaları

Çalışmada 25 yazar alınsa dahi link yapısı belirlenen gazeteden istenilen sayıda yazar sisteme dahil edilebilir. Gazeteler birbirlerinden farklı yapılar kullansalar da, yazarları için ortak yapı kullanmaktadırlar, yazara göre link formatı değiştirmezler. *tbl_yazar* tablosunda bulunan kisa_ad alanı yazarın gazetesindeki isimlendirilmesidir. Yazarlar, sitelerde veritabanında bulunan kisa_ad alanıyla işlem görürler.

4.4.3. Sınıf (Kategori) Tanımlamaları

Ana sınıflar (ekonomi, spor, sağlık, eğitim, yaşam) ve yardımcı sınıflar (karışık, aynı içerik, atanmamış, test) vardır. Çizelge 4.3.'te sınıfların hangi durumda kullanıldıkları belirtilmiştir. Ana sınıflar, sınıflandırılacak yazıya atanması beklenen ve doğru sınıflandırma yapıldığının belirlenmesinde kullanılacak sınıflardır. Yardımcı sınıflar ise yazıların alınmasından sınıflandırılması işlemleri sürecinde yazıların sınıfı hakkında bilgi veren sınıflar olarak belirlenmiştir.

Çizelge 4.3. Sınıf (kategori) bilgileri.

Sınıf ID	Sınıf (Kategori)	Açıklama
1	Ekonomi	Ekonomi dokümanlarını ifade eder.
2	Spor	Spor dokümanlarını ifade eder.
3	Sağlık	Sağlık dokümanlarını ifade eder.
4	Eğitim	Eğitim dokümanlarını ifade eder.
5	Yaşam	Yaşam dokümanlarını ifade eder.
6	Karışık	Farklı sınıflara ait olma ihtimali olan dokümanları ifade eder.
7	Aynı İçerik	Aynı içeriğe sahip birden fazla doküman bulunduğunu ifade eder.
8	Atanmamış	Hiçbir sınıf belirleyemediğini ifade eder.
9	Test	Sınıf belirleme işleminin yapılmadığını ifade eder.

4.5. EĞİTİM DOKÜMANLARI (YAZILARI) İŞLEMLERİ

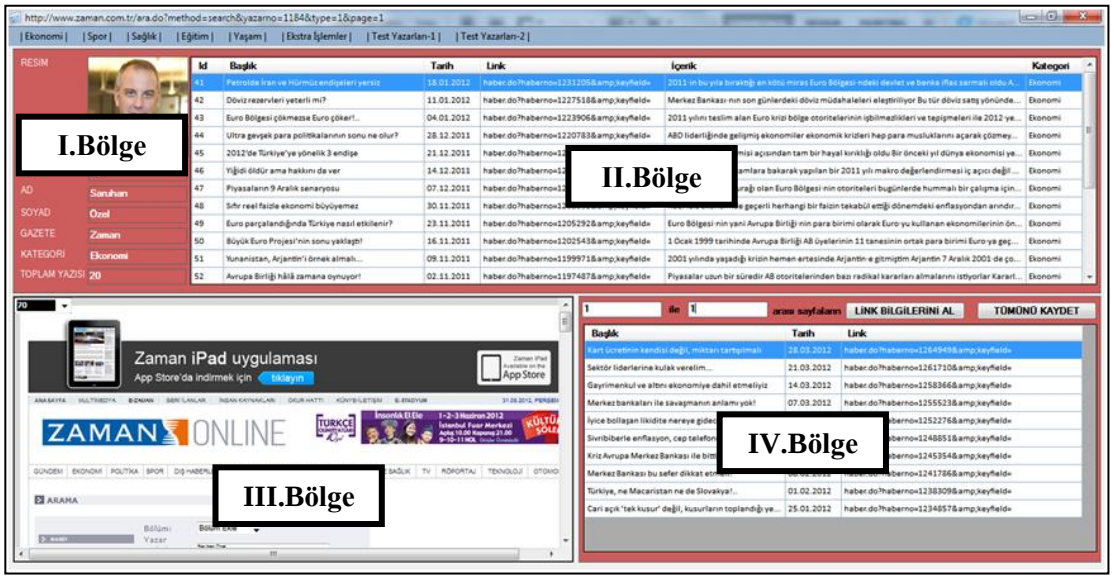
Yazıların alınmasından sınıflandırma yapılmasına kadar olan -tanımlar haricindeki- bütün işlemler menüler ve seçenekler yardımıyla gerçekleştirilir. Program çalıştırıldığında Şekil 4.5.'te gösterildiği gibi "*Köşe Yazısı İşlemleri*" ve "*Madencilik İşlemleri*" olmak üzere iki düğme aracılığıyla seçim gerçekleşmektedir. Köşe yazısı işlemlerinde, kullanılacak eğitim ve test dokümanlarının alınması, ön işlemden geçirilmesi, kelime ve yazı-kelime tablolarına verilerin kaydedilmesi gerçekleştirilir. Madencilik işlemlerinde ise sözcük ağırlıklandırılması, özellik seçimi, vektörlerin oluşturulması ve sınıflandırma işlemleri vardır. Dokümanların ve yazıdaki kelimelerin sayısı, işlem süresini arttırmaktadır.

Yazıların alınması, ön işlemden geçirilmesi, köklerin veritabanına kaydedilmesi işlemi, hem eğitim hem de test dokümanı olarak kullanılan köşe yazılarında uygulanan ortak işlemdir. Ayrıca yapılacak bütün çalışmalardan önce yazar seçimi yapılmalıdır.

Şekil 4.5. Yazılım, işlem seçim ekranı.

Şekil 4.6.'da "*Köşe Yazısı İşlemleri*" ekran görüntüsü verilmiştir. Ekran görüntüsü, 4 bölge ve menüden oluşmaktadır:

- 1.Bölge; seçilen yazarın bilgilerinin görüntülendiği alandır.
- 2.Bölge; seçilen yazarın veritabanında kayıtlı yazılarının ve yazılar üzerine yapılacak işlemlerin açılır menü ile görüntülendiği alandır.
- 3.Bölge; web sayfalarının görüntülendiği alandır.
- 4.Bölge; seçilen yazarın yazı bilgilerinin sitesinden alındığı, görüntülendiği ve veritabanına kaydetme işleminin gerçekleştiği alandır.
- Menü; yazar seçimi, yazılardaki kelime köklerinin bulunması, yazı-kelime dağılımlarının yapılması, aynı içerikleri yazıların belirlenmesi, kategorilerdeki dokümanların özet bilgilerinin görüntülenmesi işlemlerinin yapıldığı alandır.



Şekil 4.6. Köşe yazısı işlemleri ekran görüntüsü ve bölgeleri.

4.5.1. Dokümanların Alınması

Gazetelerin web sayfalarını profesyonel birim veya kişiler hazırlamaktadırlar. Bu durum, web sayfalarını oluşturan HTML etiketleri (div, p, img, a, li vb) ve CSS (Cascading Style Sheets) gibi web nesnelerinin kullanımında düzeni sağlamıştır. Yazı bilgileri ve içerikleri alınırken sayfanın kaynak kodu üzerinden işlemler gerçekleştirilir.

Veritabanına kaydedilecek eğitim ve test dokümanlarının alınması 2 adımdan oluşur; öncelikle yazarın sayfasındaki yazı bilgileri (Bkz. Şekil 4.6. IV Bölge), sonrasında yazı içerikleri alınır (Bkz. Şekil 4.6. II Bölge). Bu çalışmada, aynı yazılar kullanılarak farklı sınıflandırma çalışmalar yapılması amaçlandığından test dokümanları veritabanına kaydedilmiştir fakat anlık sınıflandırma yapılacaksa test dokümanlarının veritabanına kaydedilmesi zorunlu değildir.

4.5.1.1. Köşe Yazısı Bilgilerinin Alınması

Verilerin kaynak koddan alınması, ifadelerin aranmasıyla elde edilir. Çizelge 4.4.'te, Habertürk gazetesinin yazı bilgilerinin alınması için kaynak kodda aranan ifadeler verilmiştir.

Çizelge 4.4. Habertürk gazetesi, yazı bilgileri ifadeleri.

İfade	Açıklama
<A class=header	Yazı bilgisi başlangıç ifadesidir.
</DIV>	Yazı bilgisi bitiş ifadesidir.
href="	Link bilgisi başlangıç ifadesidir.
">	Link bilgisi bitiş ifadesidir.
">	Başlık bilgisi başlangıç ifadesidir.
</A>	Başlık bilgisi bitiş ifadesidir.
<DIV id=HaberTarih class=date>	Tarih bilgisi başlangıç ifadesidir.

Sayfadaki başlangıç ve bitiş ifadeleriyle yazı bilgisi elde edilir. *InStr* komutuyla, bu bilgi içerisinden link, başlık ve tarih bilgisinin çekilmesi işlemi gerçekleştirilir. Bu işlem sayfada bulunan yazı sayısı kadar tekrarlanır. Bütün gazetelerde sıralama link, başlık ve tarih şeklinde değildir bazılarında ise link, tarih ve başlık veya farklı şekillerde bulunabilir.

Köşe yazıları, sitelerde tarih sırasına göre sıralanmış halde sayfa sayfa görüntülenmektedir. Yazı bilgilerinin alınması işlemi, Şekil 4.6.'daki IV.Bölgede gerçekleştirilir. IV.Bölgedeki "*LİNK BİLGİLERİNİ AL*" düğmesi, girilen başlangıç ve bitiş arasındaki sayfaları gezerek yazı bilgilerini alır. Yazılar alınır alınmaz veritabanına kaydedilmez. Kullanıcı bu yazılar içerisinden istediklerini veya tümünü veritabanına kaydedebilir. Yazıların tümünün kaydedilmesi "*TÜMÜNÜ KAYDET*" düğmesiyle gerçekleştirilirken, Şekil 4.7.'de gösterilen IV.Bölgenin (Bkz. Şekil 4.6.) açılır menüsü kullanılarak yazı üzerindeki diğer işlemler gerçekleştirilir. Bu işlemler şu şekildedir:

- Veritabanına Kaydet; yazarın seçilen yazısının bilgilerinin veritabanına kaydedilmesini gerçekleştirir.
- Listeden Kaldır; yazarın seçilen yazısının listeden kaldırılmasını gerçekleştirir.
- Listeyi Temizle; bulunan köşe yazılarının listelendiği alanın temizlenmesini gerçekleştirir.
- Köşe Yazısını Görüntüle; yazarın seçilen yazısının tarayıcıda (III.Bölge) görüntülenmesini gerçekleştirir.

Şekil 4.7. IV.Bölge açılır menü görüntüsü.

4.5.1.2. Köşe Yazısı İçeriğinin Alınması

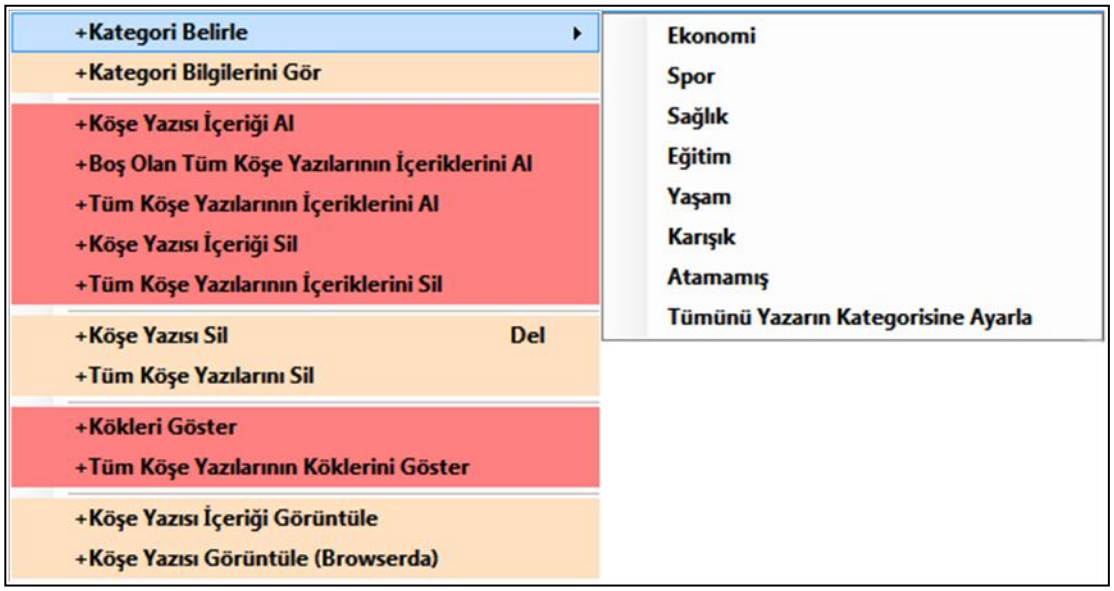
Yazı bilgileri alındıktan sonra yazıların alınmasına geçilebilir. Yazı bilgilerinin alınması, üç farklı veri (link, başlık, tarih) çekilmesi nedeniyle yazıların (yazı içerikleri) alınmasına göre daha karmaşık bir işlemdir. Yazı içeriği, yazı bilgilerinin alınmasına benzer bir metotla fakat farklı ifadelerin aranmasıyla elde edilir. Örneğin, Zaman gazetesinden yazıyı almak Çizelge 4.5.'teki başlangıç ve bitiş ifadeleri *InStr* komutuyla arattırılarak yazı başlangıç ve bitişi değerleri arasındaki içerik çekilir.

Çizelge 4.5. Zaman gazetesi yazı ifadeleri.

İfade	**Açıklama**
`<DIV id=news-detail-spot>`	Yazı başlangıç ifadesidir.
`</DIV>`	Yazı bitiş ifadesidir.

II.Bölgenin (Bkz. Şekil 4.5.) açılır menüsü kullanılarak Şekil 4.8.'de gösterilen yazı işlemleri gerçekleştirilir. Bu işlemler içerisinden içeriklerin alınmasıyla ilgili olanlar şunlardır:

- Köşe Yazısı İçeriği Al; yazarın seçilen yazısının içeriğinin alınıp veritabanına kaydedilmesini gerçekleştirir.
- Boş Olan Tüm Köşe Yazılarının İçeriklerini Al; yazarın içeriği boş olan yazılarının veritabanına kaydedilmesini gerçekleştirir.
- Tüm Köşe Yazılarının İçeriklerini Al; yazarın tüm yazılarının veritabanına kaydedilmesini gerçekleştirir.



Şekil 4.8. II.Bölge açılır menü görüntüsü.

4.5.1.3. Köşe Yazısı Diğer İşlemleri

Köşe yazıları veritabanına alındıktan sonra bunlar üzerinde uygulanacak işlemler Şekil 4.8.'de gösterildiği gibi şunlardır:

- Kategori Bilgilerini Gör; yazarın kategorilere göre yazılarının sayısını ekrana getirir.
- Köşe Yazısı İçeriği Sil; yazarın seçilen yazısının içeriğinin veritabanından silinmesi işlemini gerçekleştirir.
- Tüm Köşe Yazılarının İçeriklerini Sil; yazarın tüm yazılarının içeriklerinin veritabanından silinmesi işlemini gerçekleştirir.
- Köşe Yazısı Sil; yazarın seçilen yazısının veritabanından silinmesi işlemini gerçekleştirir.
- Tüm Köşe Yazılarını Sil; yazarın tüm yazılarının veritabanından silinmesi işlemini gerçekleştirir.
- Kökleri Göster; yazarın seçilen yazısının içeriğinin köklerinin görüntülenmesi işlemini gerçekleştirir.
- Tüm Köşe Yazılarının Köklerini Göster; yazarın sırayla, tüm yazılarının içeriklerinin köklerinin görüntülenmesi işlemini gerçekleştirir.

- Köşe Yazısı İçeriği Görüntüle; yazarın seçilen yazısının içeriğinin görüntülenmesini gerçekleştirir.
- Köşe Yazısı Görüntüle; yazarın seçilen yazısının tarayıcıda (III.Bölge) görüntülenmesini gerçekleştirir.

Köşe yazılarının alınması işlemi yazarların tek tek seçilmesiyle gerçekleştirilebilir. Fakat sınıfa ait bütün yazarlar için yazı bilgileri ve yazı içerikleri alma işlemi Şekil 4.9.'da gösterilen menüler (Bkz. Şekil 4.6.) kullanılarak tek seçimle gerçekleştirilebilir. Bu işlemler şunlardır;

- Bütün Köşe Yazarlarının Yazılarını Kaydet; seçilen sınıf yazarlarının yazı bilgilerinin, II.Bölgedeki (Bkz. Şekil 4.6.) başlangıç ve bitiş arasındaki sayfaların gezilmesi suretiyle alınması ve veritabanına kaydedilmesi işlemini gerçekleştirir.
- Bütün Köşe Yazarlarının Yazılarını Sil; seçilen sınıf yazarlarının yazı bilgilerinin veritabanından silinmesi işlemini gerçekleştirir.
- Bütün Köşe Yazarlarının Yazılarının İçeriklerini Kaydet; seçilen sınıf yazarlarının yazı içeriklerinin alınması ve veritabanına kaydedilmesi işlemini gerçekleştirir.
- Bütün Köşe Yazarlarının Yazılarının İçeriklerini Sil; seçilen sınıf yazarlarının yazı içeriklerinin silinmesi işlemini gerçekleştirir.

Şekil 4.9. Sınıfa ait yazarların yazılarının alınması işlemi ekran görüntüsü.

4.5.2. Metin Ön İşlem

Yazıların, ön işlemden geçirilmesi bir takım işlemleri barındırır. İnternet sayfaları, HTML etiketlerinden ve CSS ifadelerinden meydana geldiğinden salt bir metine uygulanan ön işlem aşamalarından farklı olarak bazı işlemlere tabi tutulurlar.

Metni HTML etiketlerinden arındırmak ön işlem aşaması için yeterli değildir. Metin madenciliği çalışmalarında ön işlem, kelimelerin köklerine ayrılmasıyla sonlanır. Sınıflandırma çalışmalarında yazı içerikleri kullanıldığından ve yazı bilgileri kullanılmadığından, yazı bilgilerine ön işlem uygulanmamış sadece yazı içeriklerine uygulanmıştır. İçeriğin HTML etiketlerinden ve karakterlerden temizlenmesi, yazı içeriği alınırken gerçekleşirken köklerine ayrılması ve gereksiz kelimelerden temizlenmesi kök bulma işlemi yapılırken gerçekleşir.

Aynı zamanda boyut azaltma işlemi de olan ön işlem aşamasının amacı, yazının sınıflandırılması işleminde sınıflandırıcının karar vermesine yardımcı olmayan, temizlenmediğinde yanlış sonuçlar çıkmasına neden olabilecek verilerin yazıdan temizlenmesi ve kelime köklerinin elde edilmesidir.

4.5.2.1. İçeriğin HTML Etiketlerinden Temizlenmesi

İçeriğin HTML etiketlerinden arındırılması işlemi, içerikteki HTML ifadelerinin aranıp temizlenmesiyle mümkün olsa da bu uzunca bir işlemdir. Bunun yerine, kullanılan köşe yazısı alınır ve Şekil 4.10. uygulanarak metin HTML etiketlerinden arındırılır (Karaca ve Görgünoğlu, 2012).

```
w_TMP.DocumentText = kose_yazisi
kose_yazisi=w_TMP.Document.Body.InnerText
```

Şekil 4.10. Yazının, HTML etiketlerinden temizlenmesi.

4.5.2.2. İçeriğin Karakterlerden Temizlenmesi

Harfler ve sayılar aynen alınır. Bütün farklı şekillerde yazılmış çift tırnaklar, “ " ” şeklindeki çift tırnağa dönüştürülür. Bunlar dışındaki karakterler ise “ ” ile değiştirilir. Bu şekilde bütün karakterler gözden geçirilmiş ve yazı, kök bulma işlemine hazır hale getirilmiş olur.

4.5.2.3. İçerikteki Kelimelerin Köklerine Ayrılması

Türkçe gibi bitişken (bükümlü) dillerde ise kelimeler, en küçük anlamlı parçasının sınırlarına dair bir belirti göstermez, üstelik bu parçalar, morfolojik ve fonolojik şartlara bağlı olarak şekil alırlar. Türkçede bir kelimenin son ekine bir tane daha ekleyerek, nispeten uzun kelimeler elde edilebilir, üstelik, sadece bir tek Türkçe kelimeden çok miktarda değişik anlamlı kelimeler oluşturulabilir. Bu karmaşık morfolojik yapı yüzünden, Türkçe, İngilizce ve benzeri dillerden daha farklı metin işleme teknikleri gerektirir. Bu nedenle, bütün kelimelerin küçük harfe çevrilmesi ve noktalama işaretlerinin kaldırılması dışında joker kelimeler ile anahtar kelimelerin oluşturulması gibi bazı ön hazırlıklar yapılması gerekmektedir (İlhan, 2001).

Dokümandaki bütün karakterlerin küçük harfe dönüştürerek gerçekleştirilen çalışmalar vardır (Aşlıyan ve Günel, 2010). Fakat bu çalışmada kullanılan kelime kök bulucusu, Çizelge 4.6.’da görüldüğü gibi küçük harfe çevrilmiş özel isimleri bulamamaktadır. Bu sebeple karakterlerin küçük harfe dönüştürülmesi tercih edilmemiştir.

Çizelge 4.6. Küçük harfe çevrilmiş özel isimlerin kökleri.

Kelime	**Kök (Sözcük)**
ankara	Kök bulunamadı.
ankara’da	Kök bulunamadı.
Ankara	Ankara
Ankara’da	Ankara

Java'da hazırlanan bir yazılımla, Türkçe doğal dil işleme kütüphanesi Zemberek kullanılarak kelime köklerinin bulunması işlemi gerçekleştirilmiştir (http://code.google.com/p/zemberek/downloads/detail?name=zemberek-2.1.1.zip, 2011). Hazırlanan yazılım *jar* dosyası olarak kaydedilip programdaki menü seçeneklerinden (Bkz. Şekil 4.6.) "*Ekstra İşlemler>Kelimeleri Köklerine Ayır*" seçeneğiyle çalıştırılıp kelimeler köklerine ayrılır.

Yazılardan alınan kelimeler diziye aktarılır ve elde edilen kelime dizisinin her bir elemanı Şekil 4.11.'de görüldüğü gibi kökleri bulunmak üzere *kelimeCozumle("kökü_bulunacak_kelime")* fonksiyonuna gönderilir. Fonksiyondan kelimenin kökleri, kökün hangi tipte (sıfat, fiil, özel vb) olduğu ve aldığı ekler döndürülür. Bu şekilde kök elde edilmiş olur.

```
Kelime[] cozumler=z.kelimeCozumle("kökü_bulunacak_kelime ");
for (Kelime kelime : cozumler)
     {
     System.out.println
     (kelime.kok().icerik());
     }
```

Şekil 4.11. Kelimeyi fonksiyona gönderen ve fonksiyondan aldığı kökü ekrana yazdıran kod.

Yeni kelimeler ekler alarak türetildiğinden, ekler teker teker kaldırılarak kökler elde edilip, Çizelge 3.2.'de gösterilen "düşündü" kelimesinin kökleri içerisinden "düşün" kökünün seçilmesi örneğinde olduğu gibi, en uzun kök, kelime kökü olarak kabul edilir. Hazırlanan yazılım ve kök bulucusu, kökleri, en uzun kökten en kısa köke doğru (eklerin sırayla kaldırılmasıyla elde edilen kökler) sıralı olarak vermektedir. Elde edilen ilk kök hem en uzun kök hem de aranılan köktür. Kök bulma işlemi yazı içerikleri için bu şekilde bütün kelimeler için gerçekleşir.

Çizelge 4.7.'de sınıflara göre kök bulma işleminden elde edilen sonuçlar görüntülenmektedir. Gruplandırma işlemi yapılmamıştır ve her bir kelimenin yazıda her bulunması ayrı bir kelime gibi değerlendirilerek sonuçlar elde edilmiştir.

Çizelge 4.7. Elde edilen kelimelerin (köklerin) yazılarda bulunma sayıları.

Özellik	Ekonomi	Spor	Sağlık	Eğitim	Yaşam
Toplam Kelime Sayısı	62252	41910	43044	42828	49859
Kökü Bulunabilen Kelime Sayısı	57501	38570	40746	40456	47196
Kökü Bulunamayan Kelime Sayısı	4751	3340	2568	2372	2663
Tipi Sayı Olan Kelime Sayısı	2518	1822	1202	1337	2483
Gereksiz Kelime (Stop Words) Sayısı	12944	9217	8445	10001	12466
1 Harfli Kelime Sayısı	159	256	51	137	343
2 Harfli Kelime Sayısı	8159	4939	5815	6210	6498
3 Harfli Kelime Sayısı	11023	8896	6836	7506	10794
4 Harfli Kelime Sayısı	10011	6931	7372	7275	8887
5 Harfli Kelime Sayısı	14993	9844	10600	9482	12511
6 Harfli Kelime Sayısı	6942	3916	5031	4551	4278
7 Harfli Kelime Sayısı	3752	1851	2534	2869	2271
8 Harfli Kelime Sayısı	1675	1308	1249	1358	1179
9 Harfli Kelime Sayısı	400	280	355	388	208
10 ve Daha Fazla Harfli Kelime Sayısı	387	349	633	680	227

4.5.2.4. İçeriğin Gereksiz Kelimelerden Temizlenmesi

Türkçede ve diğer dillerde tek başına anlamı olmayan, birçok metinde sıkça geçen gereksiz kelimelerin (stop words) anlama bir katkısı olmaması nedeniyle metinlerden kaldırılır. Gereksiz kelimelerin kaldırılması, kök bulma işlemi sonrasındaki kontrolde gerçekleştirilir. Eğer kök, gereksiz kelimelerden biri ise veritabanına kaydedilmez. Şekil 4.12.'de gereksiz kelimelerin listesi, Çizelge 4.7.'de ise sınıflara göre dokümanlarda bu kelimelerin geçme sayıları verilmiştir. Türkçede kullanılan gereksiz kelimeler Şekil 4.12.'de gösterilenden daha fazla olmasına rağmen kelime kökü bulunduktan sonra gereksiz kelime kontrolü yapıldığından liste normale göre kısadır. Gereksiz kelimelerin de kökleri bulunarak çalışılmıştır.

Yazı ön işlem aşaması, içeriğin gereksiz kelimelerden temizlenmesiyle son bulur. Bundan sonraki aşamada yazıyı oluşturan kelimelerin kökleri bir boşluk karakteri

bırakılarak birleştirilir ve veritabanındaki Şekil 4.13.'te gösterilen *tbl_yazi_EGITIM* tablosunun *icerik_kok* alanına kaydedilir.

"acaba", "altı", "ama", "ancak", "artık", "asıl", "asla", "az", "bazen", "bazı", "belki", "ben", "beş", "bile", "bir", "birçok", "biri", "birisi", "birkaç", "biz", "böyle", "böylece", "bu", "buna", "bura", "bütün", "çoğu", "çok", "çünkü", "da", "daha", "de", "değil", "diğer", "diye", "dokuz", "dolayı", "dört", "elbette", "en", "fakat", "falan", "felan", "filan", "gene", "gibi", "hala", "hangi", "hani", "hatta", "hem", "henüz", "hep", "hepsi", "her", "herkes", "hiç", "hiçbiri", "için", "içinde", "iki", "ile", "ise", "işte", "kaç", "kadar", "kendi", "ki", "kim", "kimi", "kimisi", "l", "m", "madem", "mı", "mi", "mu", "mü", "nasıl", "ne", "neden", "nere", "ney", "niçin", "niye", "o", "on", "ona", "onda", "onlar", "onu", "onun", "ora", "oysa", "oysaki", "öbür", "ön", "önce", "ötürü", "öyle", "rağmen", "sekiz", "sen", "siz", "son", "sonra", "şayet", "şey", "şimdi", "şöyle", "şu", "tabi", "tamam", "tüm", "üç", "üzere", "var", "ve", "veya", "veyahut", "ya", "yani", "yedi", "yerin", "yine",

Şekil 4.12. Gereksiz kelime listesi.

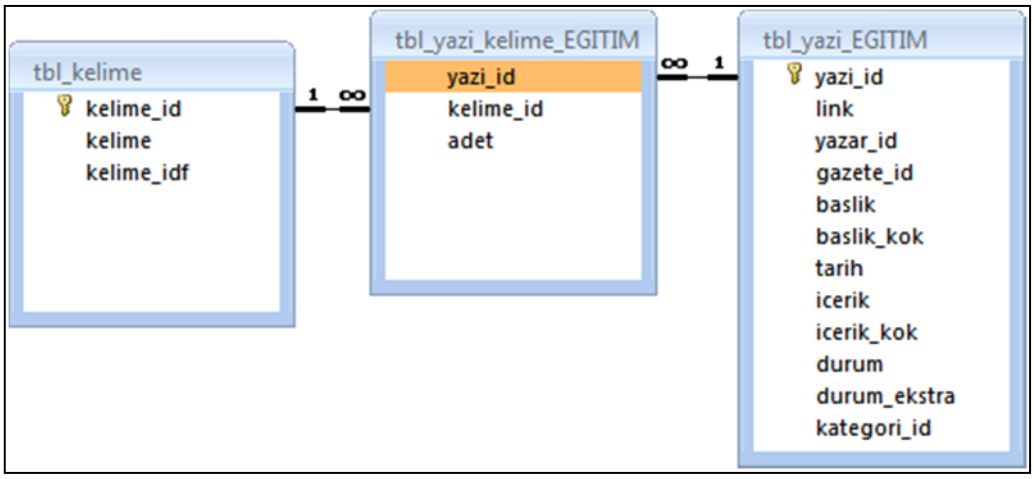
4.5.3. Kelimelerin Veritabanına Kaydedilmesi

Yazıda geçen kelimelerin en yakın kökleri yerine kendilerinin direk kullanılmaları hem boyutu arttırır hem de yanlış sonuçlar alınmasına neden olur. Ön işlem aşaması, alınan yazının sınıflandırılabilir yazı haline dönüştürülmesindeki en önemli adımlardan biridir. Fakat bu yeterli değildir.

Sözcük, sınıflandırma işleminde kökünü temsil eden sayısal değerle tutulmalıdır. Her bir sayı, farklı bir kökü (sözcüğü) ifade eder. Şekil 4.13.'te kelime, yazı ve yazı-kelime dağılımlarının tutulduğu tablolar ve ilişkileri görülmektedir. Menü (Bkz. Şekil 4.6.) seçeneklerinden "*Ekstra İşlemler>Köşe Yazısı Kelime Dağılımını Yap*" seçeneğiyle köşe yazısı dağılımı gerçekleşirken sözcükler de veritabanına kaydedilir.

İçeriği boş olmayıp kategorisi belirlenmiş olan içerik, *tbl_yazi_EGITIM* tablosunun yazının içeriğinin kök halinde tutulduğu *icerik_kok* alanından alınır. *Dim kelime() = Split(Trim(dtst("icerik_kok")), " ")* komutuyla, *icerik_kok* alanı oluşturulurken kökleri ayıran boşluk karakteri kullanılarak yazıyı oluşturan kelime kökleri *kelime* dizisine aktarılır.

Dizi elemanları sırayla önce köklerin bulunduğu *tbl_kelime* sonra kelime-yazı dağılımlarının tutulduğu *tbl_yazi_kelime_EGITIM* tablosuna kaydedilir. Bu şekilde sözcüklerin sayısal olarak ifade edilmesi işlemi gerçekleştirilmiş olur. Yazı içeriğindeki sözcükler, *kelime* dizisine aktarıldıktan sonra tek tek alınıp *tbl_kelime* tablosuna bakılır. Eğer tabloda kayıtlı değilse kaydedilir ve böylece yazılarda geçen bütün sözcüklerin tutulduğu bir havuz oluşturulmuş olur. *tbl_kelime* tablosuna kaydedilen sözcüğün dokümanda her geçmesinde *tbl_yazi_kelime_EGITIM* tablosundaki, sözcüğün dokümanda kaç kez geçtiği bilgisinin tutulduğu *adet* alanı bir arttırılır ve bu işlem her yazı ve her kök için tekrarlanır.



Şekil 4.13. Kelime, yazı ve kelime-yazı tabloları ve ilişkileri.

Şekil 4.14.'te *tbl_yazi_kelime_EGITIM* tablosundaki verilerden *kelime_id* değeri 1 olan kökün yazılarda geçme durumunu gösteren bir kesit görülmektedir. Bu kesite bakarak 1 id'li yazı içerisinde 7, 8 id'li yazı içerisinde 3, 462 id'li yazı içerisinde 2, 18 id'li yazı içerisinde 5, 19 id'li yazı içerisinde 15, 27 id'li yazı içerisinde 10 ve 28 id'li yazı içerisinde 6 kez geçtiği görülmektedir.

tbl_kelime tablosunda 10823, *tbl_yazi_kelime_EGITIM* tablosunda 94947 veri bulunmaktadır. Eğitim dokümanlarında geçen sözcük sayısı 7056, test dokümanlarında geçen sözcük sayısı 5756'dır.

Bir sözcük birden fazla yazıda geçebilmektedir. *ol* fiili bir yazı dışında bütün yazılarda geçerek en yüksek frekanslı ve en düşük *idf* değerine sahip kelime

olmuştur. *ol* sözcüğünü yap, et, ver, al, yıl ve gel izlemektedir. Sadece bir yazıda geçen sözcük sayısı 2130'dir.

tbl_yazi_kelime_EGITIM

yazi_id	kelime_id	adet
1	1	7
8	1	3
462	1	2
18	1	5
19	1	15
27	1	10
28	1	6

Şekil 4.14. Kelime-yazı dağılımını gösteren tbl_yazi_kelime_EGITIM tablosu verilerinden bir görüntü.

4.5.4. Sözcük Ağırlıklandırma

Sözcükler (kökler), sayısal veri haline dönüştürülmüş ve yapısallık elde edilmiştir. Yazılar, vektörel olarak ifade edilirler. Bu vektörler, yazıları oluşturan sözcüklerden meydana gelir. Vektörler oluşturulurken köklerin vektörde hangi değerle temsil edileceği sözcük ağırlıklandırma ile belirlenir. Sözcükler, yazılarda bulunma durumlarına göre bit, *tf*, *idf* ve *tf-idf* ağırlıklandırılmalar kullanılarak ağırlıklandırılıp vektörlerde temsil edilirler. Şekil 4.22., eğitim dokümanları vektörlerinin bit, *tf* ve *tf-idf* ağırlıklandırılmış olarak ifadelerine örnek olarak verilmiştir.

idf ve *tf-idf* ağırlıklandırma için *idf* değerinin hesaplanması gerekmektedir. Sözcüklerin *idf* değerlerinin hesaplanması, menüdeki (Bkz. Şekil 4.6.) "*Ekstra İşlemler>Kelime idf Hesapla*" seçeneğiyle gerçekleştirilir. Bu işlem, bütün sözcüklerin *idf* değerlerini hesaplayıp *tbl_kelime* tablosundaki *kelime_idf* alanına kaydeder.

4.5.5. Özellik Seçimi (Özellik Vektörünü Oluşturan Sözcüklerin Seçimi)

Ön işlem aşamasıyla başlayan boyut azaltma teknikleri özellik seçim işlemiyle son bulur. Özellik vektörünü oluşturan sözcüklerin seçimi veya özellik seçimi işlemi,

yazıyı temsil edecek sözcüklerin belirlenmesidir. Bu işlemle büyük boyutlardaki yazıların boyutu indirgenmiş olur ve işlemlerin daha hızlı gerçekleştirilmesi sağlanır.

Özellik seçimini sadece yazıların vektör boyutunun azaltılması olarak görmek, konuyu eksik değerlendirmek olur. Bazı sınıflandırma işlemleri sonuçları, düşük boyutlu özellik vektörleri ile yapılan çalışmalarda daha başarılı sonuçlar elde edildiğini ortaya koymuştur (Durmaz ve Bilge, 2011).

Çizelge 4.8.'de sınıflara göre yazıların sözcük sayılarıyla ilgili veriler bulunmaktadır. *Eğitim* sınıfı, 78 sözcük ile en düşük ve 946 sözcük ile en yüksek sözcük sayısına sahip sınıftır. Bütün yazılar dikkate alındığında ortalama sözcük sayısı 341'dir. Ayrıca sınıfları oluşturan sözcük sayıları verilmiştir; en az sözcük sayısına sahip sınıf *Eğitim*, en fazla sözcük sayısına sahip sınıf *Yaşam* sınıfıdır.

Çizelge 4.8. Sınıflara göre sözcük sayıları.

Özellik	**Ekonomi**	**Spor**	**Sağlık**	**Eğitim**	**Yaşam**
Yazıların Oluştuğu Sözcük Sayısı	21123	18892	17430	15296	22106
Yazıların Kaç Farklı Sözcükle Oluştuğu	2712	3440	3119	2558	4270
Minimum Sözcük Sayısı	180	148	103	*78*	125
Maksimum Sözcük Sayısı	700	660	692	*946*	809
Sınıfların Ortalama Sözcük Sayısı	445	293	319	304	346
Ortalama Sözcük Sayısı	341				

Çizelge 4.8.'de eğik olanlar, minimum ve maksimum sözcük sayısına sahip sınıfı göstermektedir.

Çizelge 4.9.'da sözcük sayılarına göre sınıflardaki eğitim dokümanları yazı sayıları gösterilmektedir. *Ekonomi*, *Spor* ve *Sağlık* sınıflarındaki yazılar, 101 ile 700 arası; *Eğitim* sınıfındaki yazılar, 1 yazı haricinde 1 ile 700 arası; *Yaşam* sınıfındaki yazılar ise 101 ile 900 arası sözcükten meydana geldiği görülmektedir. 201-300 arası sözcük

aralığı 187 yazı ile en fazla yazı bulunduran aralıktır. 500 eğitim dokümanından 248 tanesi, 1-300 arası sözcükten; 252 tanesi, 301 ve daha fazla sayıda sözcükten meydana gelmektedir. 289 yazı, ortalama sözcük sayısının altında; 211 yazı, ortalama sözcük sayısının üstünde sözcüğe sahiptir.

Çizelge 4.9. Sözcük sayılarına göre sınıflardaki yazı sayıları.

Yazıyı Oluşturan Sözcük Sayısı	**Ekonomi**	**Spor**	**Sağlık**	**Eğitim**	**Yaşam**	**Toplam**
1-100 Arası	-	-	-	3	-	**3**
101-200 Arası	2	22	16	8	10	**58**
201-300 Arası	13	45	44	43	42	**187**
301-341 Arası	12	3	9	8	9	**41**
342-400 Arası	17	6	8	22	5	**58**
401-500 Arası	16	14	4	9	19	**62**
501-600 Arası	25	8	16	3	10	**62**
601-700 Arası	15	2	3	3	1	**24**
701-800 Arası	-	-	-	-	3	**3**
801-900 Arası	-	-	-	-	1	**1**
901'den Fazla	-	-	-	1	-	**1**
Toplam	**100**	**100**	**100**	**100**	**100**	**500**

Yazılardaki bütün sözcüklerin çalışmalarda kullanılması, bazı çalışmalarda yanlış sonuçlar alınmasına, sınıflandırma işlem süresinin uzamasına sebebiyet verir. Bu sebeple, bütün sözcüklerin çalışmalara dahil edilmesi yerine yazıyı en iyi temsil edecek sözcüklerin kullanılması tercih edilir. Bu çalışmada, yazıları ifade edecek sözcüklerin seçimi için 4 farklı yaklaşım kullanılmıştır: Birinci yaklaşım, sözcüklerin geçtiği eğitim doküman sayısının dikkate alındığı; ikinci yaklaşım, sözcüklerin eğitim dokümanlarında geçme sayısı toplamının dikkate alındığı; üçüncü yaklaşım, en yüksek *idf* değerine sahip sözcüklerin dikkate alındığı; dördüncü yaklaşım ise bütün sözcüklerin dikkate alındığı yaklaşımdır. Birinci ve ikinci yaklaşımda sadece kendi sınıfındaki yazılarla ilgilenirken; üçüncü ve dördüncü yaklaşım, bütün eğitim dokümanlarını dikkate alır.

X=(1·0·2·2·0·3·1·1·5·0)

Y=(0·2·0·3·0·4·1·2·0·8)

Z=(0·1·0·0·0·2·1·1·2·0)

Şekil 4.15. Sınıfı *Ekonomi* olan özellik seçimi uygulanmamış *X, Y, Z* dokümanlarının *tf* ağırlandırılmış vektörel ifadesi.

Sözcüklerin geçtiği eğitim dokümanı sayısını dikkate alan birinci yaklaşımda, sözcüğün yazı içerisinde birden fazla kez geçmesi dikkate alınmaz. Önemli olan fazla yazıda geçmesidir. Şekil 4.15.'teki *X, Y* ve *Z* vektörleri üzerinden örneklendirme yapılırsa; 0 indisli sözcük 1; 1 indisli sözcük 2; 2 indisli sözcük 1; 3 indisli sözcük 2; 4 indisli sözcük 0; 5 indisli sözcük 3; 6 indisli sözcük 3; 7 indisli sözcük 3; 8 indisli sözcük 2 ve 9 indisli sözcük 1 yazıda geçmiştir. Bu bilgiler ışığında, en fazla yazıda geçme sayısına sahip 3 sözcük şeklinde bir özellik seçimi uygulanacaksa, bu sözcükler 3 yazıda geçen 5, 6 ve 7 indisli sözcükler olacaktır.

Sözcüklerin eğitim dokümanlarında geçme sayısı toplamının dikkate alındığı ikinci yaklaşımda, sözcüğün yazılarda geçme sayıları toplanır. Fazla yazıda geçmesinin önemi yoktur. Önemli olan, yazılarda geçme sayısı toplamının fazla olmasıdır. Şekil 4.15.'teki *X, Y* ve *Z* vektörleri üzerinden örneklendirme yapılırsa; 0 indisli sözcük 1; 1 indisli sözcük 3; 2 indisli sözcük 2; 3 indisli sözcük 5; 4 indisli sözcük 0; 5 indisli sözcük 9; 6 indisli sözcük 3; 7 indisli sözcük 4; 8 indisli sözcük 7 ve 9 indisli sözcük 8 geçme sayısına sahiptir. Yazılarda en fazla geçme sayısına sahip 3 sözcük değerlendirmeye alınarak işlem yapılacaksa, seçilecek olan bu sözcükler; 9 kez geçen 5 indisli; sadece *Y* yazısında geçmesine karşın bu yazıda 8 kez geçtiği için 9 indisli; 7 kez geçen 8 indisli sözcükler olacaktır.

Hem geçtiği yazı sayısı hem de yazılarda geçme sayısı toplamı yüksek olduğundan iki yaklaşımda da 5 indisli sözcük seçilmiştir. Birinci özellik seçimi yaklaşımı sonucu 5, 6 ve 7 indisli, ikinci yaklaşım sonucu 5, 8 ve 9 indisli sözcükler sınıf özellik vektörlerinin oluşturulmasında kullanılacaktır.

idf değerinde sözcüklerin geçtiği eğitim dokümanı sayısı dikkate alınır. Eğer bir sözcük, az sayıda eğitim dokümanında geçiyorsa yüksek, çok sayıda dokümanda

geçiyorsa düşük *idf* değerli sözcük şeklinde değerlendirilir ve yüksek *idf* değerli sözcükler seçilir. Sözcüklerin az sayıda yazıda geçmesi dikkate alındığından birinci yaklaşımın tam tersi şeklinde çalışmaktadır.

Sadece *Ekonomi* sınıfı yazılarına uygulanan birinci ve ikinci yaklaşın örneklendirmesinin, diğer sınıflara da uygulanması gerekir. Her sınıftan bu şekilde sözcükler alınarak birleştirilir ve sınıf özellik vektörü oluşturulur. Bir sözcük farklı iki sınıfta olsa dahi bir defa alınır.

Şekil 4.14.'te köşe yazısında geçen sözcüklerin yazi_id, kelime_id ve adet bilgileri *tbl_yazi_kelime_EGITIM* tablosunda tutulmaktadır. Bu tablodaki adet alanı kullanılarak, özellik seçimi sonucu seçilecek sözcükler elde edilir.

Örneklendirmenin, çalışmaya uyarlaması şu şekildedir. Birinci yaklaşım olan sözcüğün geçtiği yazı sayısı tespitinde C kısaltmasıyla *COUNT* kullanılır. İkinci yaklaşımda ise yazılarda geçme sayısı toplamını ifade eden S kısaltmasıyla *SUM* kullanılarak sınıf özellik vektörünü oluşturacak sözcükler elde edilir. Örneklendirmede 3 sözcük dikkate alınmıştır. Fakat bu çalışmada her sınıftan 175, 350, 500 sözcük alınmıştır.

```
SELECT TOP 175 * FROM

(SELECT kelime_id FROM tbl_yazi_kelime_EGITIM
LEFT JOIN tbl_yazi_EGITIM
ON tbl_yazi_kelime_EGITIM.yazi_id = tbl_yazi_EGITIM.yazi_id
GROUP BY kelime_id, kategori_id
HAVING kategori_id=1
ORDER BY COUNT(adet) DESC)
```

Şekil 4.16. Sınıf özellik vektörünü oluşturan sözcüklerin seçiminde, kategori_id'si 1 (*Ekonomi*) olan sözcüklerin id değerini alan kod.

Şekil 4.16., *kategori_id*'si 1 (*Ekonomi* sınıfı) olan eğitim dokümanlarında geçme sayısına göre (C), en fazla yazıda geçen ilk 175 sözcüğün *id* değerini alır. Bu işlem sadece *Ekonomi* sınıfının özellik vektörünü elde eder. Şekil 4.16.'daki kategori_id değeri değiştirilerek diğer sınıflar (kategori_id) içinde *kelime_id*'leri alınır, SQL

(*UNION*) ifadesi ile birleştirilir ve sınıf özellik vektörü oluşturulmuş olur. *UNION*, aynı sözcüklerden sadece bir tanesini alır. Çizelge 4.10. incelenciğinde C_175_bit sınıf özellik vektöründe normalde 875 (175*5) sözcük olması gerekirken 437 sözcük vardır. Bu da bazı sözcüklerin birden fazla sınıfta bulunduğunu ve bir kez alındığını gösterir.

```
SELECT kelime_id FROM

(SELECT TOP 175 * FROM (SELECT kelime_id FROM
tbl_yazi_kelime_EGITIM LEFT JOIN tbl_yazi_EGITIM ON
tbl_yazi_kelime_EGITIM.yazi_id = tbl_yazi_EGITIM.yazi_id
GROUP BY kelime_id, kategori_id HAVING kategori_id=1 ORDER
BY COUNT(adet) DESC) UNION

SELECT TOP 175 * FROM (SELECT kelime_id FROM
tbl_yazi_kelime_EGITIM LEFT JOIN tbl_yazi_EGITIM ON
tbl_yazi_kelime_EGITIM.yazi_id = tbl_yazi_EGITIM.yazi_id
GROUP BY kelime_id, kategori_id HAVING kategori_id=2 ORDER
BY COUNT(adet) DESC) UNION

SELECT TOP 175 * FROM (SELECT kelime_id FROM
tbl_yazi_kelime_EGITIM LEFT JOIN tbl_yazi_EGITIM ON
tbl_yazi_kelime_EGITIM.yazi_id = tbl_yazi_EGITIM.yazi_id
GROUP BY kelime_id, kategori_id HAVING kategori_id=3 ORDER
BY COUNT(adet) DESC) UNION

SELECT TOP 175 * FROM (SELECT kelime_id FROM
tbl_yazi_kelime_EGITIM LEFT JOIN tbl_yazi_EGITIM ON
tbl_yazi_kelime_EGITIM.yazi_id = tbl_yazi_EGITIM.yazi_id
GROUP BY kelime_id, kategori_id HAVING kategori_id=4 ORDER
BY COUNT(adet) DESC) UNION

SELECT TOP 175 * FROM (SELECT kelime_id FROM
tbl_yazi_kelime_EGITIM LEFT JOIN tbl_yazi_EGITIM ON
tbl_yazi_kelime_EGITIM.yazi_id = tbl_yazi_EGITIM.yazi_id
GROUP BY kelime_id, kategori_id HAVING kategori_id=5 ORDER
BY COUNT(adet) DESC))
```

Şekil 4.17. Bütün sınıflarda, en fazla dokümanda geçen 175 sözcüğü alıp sınıf özellik vektörünü oluşturan kod.

Şekil 4.17., sınıfında geçtiği doküman sayısı en fazla olan, her bir sınıftan 175'er sözcüğü alıp birleştirir ve C (*COUNT*) için sınıf özellik vektörünü oluşturmuş olur. Çalışmada, sınıflardan 175, 350 ve 500 sözcük alınarak işlemler gerçekleştirilmiştir. Şekil 4.17.'de bulunan sorgu, her bir sınıftan 175 sözcük almaktadır. *SELECT TOP*

175 ifadesindeki 175 sayısı 350 veya 500 yapılabilir. Bu işlem, sınıflandırma işleminde daha fazla sayıda sözcüğün kullanılmasını sağlar. Fakat sözcük sayısının artması, sınıf özellik vektörünün ve dolayısıyla doküman vektörlerinin boyutunun artmasına neden olur.

Şekil 4.17.'deki sorguda bulunan *COUNT(adet)* ifadesi *SUM(adet)* yapılarak ikinci yaklaşım olan yazılarda geçme sayısı toplamına göre sınıf özellik vektörünün oluşturulması sağlanır. Sınıf özellik vektörü isimlendirmesi şu şekilde meydana gelir. İsimlendirmede özellikler _ (alt çizgi) ile ayrılmaktadır. İlk kısım, kullanılan sözcük seçiminin kısaltmasını (*COUNT*-C veya *SUM*-S); ikinci kısım alınacak sözcük sayısını; üçüncü kısım ağırlıklandırma şeklini göstermektedir. Örneğin, C_175_bit; *COUNT* kullanılarak, her bir sınıftan 175'er sözcüğün bit ağırlıklandırma yapılacak şekilde alınacağını, S_350_*tf-idf*; *SUM* kullanılarak, her bir sınıftan 350'şer sözcüğün *tf-idf* ağırlıklandırma yapılacak şekilde alınacağını ifade etmektedir.

37 farklı özellik seçimi uygulanarak sınıflara ait sözcükler dolayısıyla sınıf özellik vektörleri elde edilmiş olur. Doküman vektörlerinin oluşturulmasında kullanılacak sınıf özellik vektörünün elde edilme şekilleri Çizelge 4.10.'da verilmiştir.

Sınıf özellik vektörünün oluşturulması şu şekilde gerçekleşmiştir; *COUNT*, *SUM*, bütün sözcükler veya yüksek *idf* değerli sözcüklerden; 175, 350, 500 sözcük veya bütün sözcükleri alarak; bit, *tf-idf*, *tf* veya *idf* ağırlıklandırma yapılabilecek şekilde düzenlenmiştir.

Çizelge 4.10. Sınıf özellik vektörlerinin oluşturulmasında kullanılan özellikler.

Sözcüklerin Seçimi	**Alınacak Sözcük Sayısı**	**Sözcük Ağırlıklandırma**	**Sınıf Özellik Vektörü İsmi**	**Sınıf Özellik Vektörü Boyutu**
COUNT	175	bit	C_175_bit	437
	350		C_350_bit	876
	500		C_500_bit	1229
SUM	175		S_175_bit	497
	350		S_350_bit	945
	500		S_500_bit	1299
Bütün Sözcükler			_bit	7056

Çizelge 4.10. (devam ediyor).

COUNT	175	*tf-idf*	C_175_*tf-idf*	437
	350		C_350_*tf-idf*	876
	500		C_500_*tf-idf*	1229
SUM	175		S_175_*tf-idf*	497
	350		S_350_*tf-idf*	945
	500		S_500_*tf-idf*	1299
Yüksek *idf* Değerli Sözcükler	175		175_*tf-idf*	2130
	350		350_*tf-idf*	2589
	500		500_*tf-idf*	3243
Bütün Sözcükler			_ *tf-idf*	7056
COUNT	175	*tf*	C_175_*tf*	437
	350		C_350_*tf*	876
	500		C_500_*tf*	1229
SUM	175		S_175_*tf*	497
	350		S_350_*tf*	945
	500		S_500_*tf*	1299
Yüksek *idf* Değerli Sözcükler	175		175_*tf*	2130
	350		350_*tf*	2589
	500		500_*tf*	3243
Bütün Sözcükler			_*tf*	7056
COUNT	175	*idf*	C_175_*idf*	437
	350		C_350_*idf*	876
	500		C_500_*idf*	1229
SUM	175		S_175_*idf*	497
	350		S_350_*idf*	945
	500		S_500_*idf*	1299
Yüksek *idf* Değerli Sözcükler	175		175_*idf*	2130
	350		350_*idf*	2589
	500		500_*idf*	3243
Bütün Sözcükler			_*idf*	7056

Çizelge 4.10.'da ayrıca sınıf özellik vektörlerinin boyutları da görülmektedir. Vektörü boyutu, sınıf özellik vektörünün oluşmasında kullanılan sözcük sayısıdır. Örneğin, sözcük seçiminde COUNT, alınacak sözcük sayısında 175 ile oluşturulan ve ismi C_175 ile başlayan bütün sınıf özellik vektörlerinin boyutları 437'dir. Vektör boyutuyla ilgili bu durum, diğer sınıf özellik vektörleri için de geçerlidir.

Yüksek *idf* değerli sözcüklerin seçimiyle oluşan sınıf özellik vektörleri boyutlarının, diğer sınıf özellik vektörleri boyutlarından daha fazla olduğu görülmektedir. Örneğin, 175_*idf* sınıf özellik vektörünün boyutu 2130'dur. Boyutun en fazla 175*5=875 olması gerekirken 2130 olması, 2130 sözcüğün bütün eğitim dokümanları içerisinde sadece bir kez geçmesindendir. Bu sözcükler 2,69897 *idf* değerine sahiptirler. 175_*idf* sınıf özellik vektörüne seçilecek kelimelerin 2130 tanesi aynı *idf* değerine sahip olduğundan, bütün sözcükler alınmak durumunda kalınmıştır. 2,69897 *idf* değerine sahip *Ekonomi* sınıfında 206, *Spor* sınıfında 480, *Sağlık* sınıfında 450, *Eğitim* sınıfında 182, *Yaşam* sınıfında 812 sözcük vardır. Buna karşın, ortak sözcük var olduğundan, C_175_*tf-idf*'deki sözcük sayısı, 875'ten 437'ye düşmüştür.

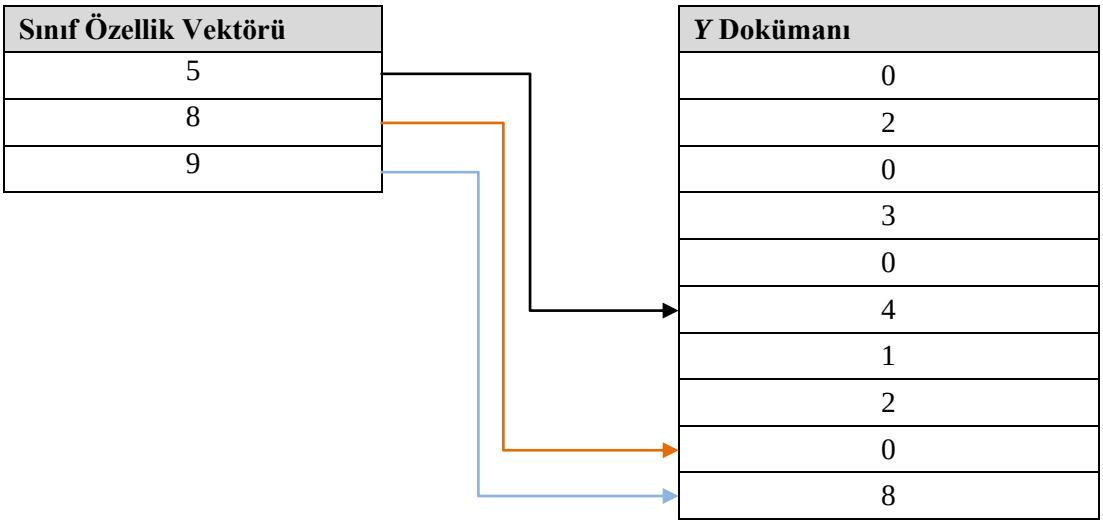
Özellik seçimi işlemi sonunda, sınıflardan alınan sözcüklerin oluşturduğu sınıf özellik vektörleri oluşturulmuş olur.

4.5.6. Dokümanların Vektörel İfadesi

Oluşturulan sınıf özellik vektörleri ile dokümanlar eşleştirilerek doküman vektörleri elde edilir.

Şekil 4.15. örneğinde iki farklı özellik seçimi yaklaşımı uygulanarak, birinci yaklaşımda 5, 6 ve 7 indisli, ikinci yaklaşımda 5, 8 ve 9 indisli sözcükler elde edilmişti. İki yaklaşımda da 3 sözcük seçilmişti. Bu sebeple sınıf özellik vektörü, *X*, *Y* ve *Z* dokümanlarının vektör boyutu 3 olacaktır. Bu 3 sözcük dokümanlarla eşleştirilecek ve tercih edilen ağırlıklandırmaya göre vektör oluşturulacaktır.

Özellik seçimi ikinci yaklaşımıyla 5, 8 ve 9 indisli sözcükler elde edilmişti. Şekil 4.18.'de, bu sözcüklerin oluşturduğu sınıf özellik vektörü ile *Y* dokümanının eşleştirilmesi gösterilmiştir. 5 indisli sözcük, *Y* dokümanında 4 defa geçmiştir. Sınıf özellik vektörleri belirlendikten sonra, vektör elemanlarının dokümanlarda varlığı araştırılır. Elemanlar, vektörlerde ağırlıklarıyla ifade edilirler. Bu örnekte, eğer ağırlıklandırma bit olarak yapılacaksa 1 değerini, *tf* olarak yapılacaksa 4 değerini alacaktır.



Şekil 4.18. Özellik seçiminde kullanılan ikinci yaklaşımın *Y* dokümanıyla eşleştirilmesi.

Şekil 4.19., birinci ve ikinci yaklaşımın bit ağırlıklandırmayla kullanılmasıyla elde edilmiş doküman vektörlerini göstermektedir. Bit ağırlıklandırmada sözcüklerin dokümanlarda varlığı araştırıldığından bütün elemanları 0 veya 1'dir. Şekil 4.19. (a), sınıf özellik vektöründeki bütün elemanlar dokümanlarda da bulunduğundan bütün elemanları 1 değerini almıştır. Fakat Şekil 4.19. (b)'de bazı vektör elemanlarının 0 değerini aldığı görülmektedir. 0 değeri, o sözcüğün dokümanda geçmediğini ifade etmektedir.

X=(1·1·1)
Y=(1·1·1)
Z=(1·1·1)

(a)

X=(1·1·0)
Y=(1·0·1)
Z=(1·1·0)

(b)

Şekil 4.19. Sınıf özellik vektörüne göre *X*, *Y*, *Z* dokümanlarının bit ağırlıklı vektörel ifadesi. a)Birinci yaklaşım, b)İkinci yaklaşım.

Şekil 4.20., Şekil 4.19.'da verilen vektörlerin, *tf* ağırlıklandırma ile gösterimidir. Şekil 4.20.'de sözcüğün varlığı yanında frekansı da önem taşır ve vektörde frekansıyla değer alır.

X=(3·1·1)
Y=(4·1·2)
Z=(2·1·1)

(a)

X=(3·5·0)
Y=(4·0·8)
Z=(2·2·0)

(b)

Şekil 4.20. Sınıf özellik vektörüne göre *X, Y, Z* dokümanlarının *tf* ağırlıklı vektörel ifadesi. a)Birinci yaklaşım, b)İkinci yaklaşım.

Şekil 4.22. dokümanların vektörel olarak ifadelerine örnektir. Vektör oluşturma işleminin zaman alması nedeniyle sınıflandırma işleminden önce vektörler oluşturulur ve veritabanına kaydedilir. Aksi takdirde her sınıflandırma işleminden önce, bütün eğitim dokümanları için vektörler oluşturulmalıdır.

Şekil 4.5.'te gösterilen "*Madencilik İşlemleri*" seçildikten sonra ekrana gelen penceredeki "*Ekstra İşlemler>Vektörleri Oluştur*" menü seçeneği kullanılarak 37 farklı özellik seçimi ve bit, *tf, idf, tf-idf* ağırlandırılmalar kullanılarak 37 farklı sınıf özellik vektörü oluşturma işlemi gerçekleştirilmiştir.

Özellik seçimi sonucu elde edilen sözcüklerle *tbl_yazi_kelime_TEST* tablosu, her bir yazı için ayrı ayrı olmak üzere eşleştirilir. Bu şekilde sözlükteki sözcüklerden yazı içerisinde geçenler bulunur, ağırlıklandırması yapılır, vektör oluşturulur ve veritabanına kaydedilir. Vektör elemanlarının birbirinden ayrılması için ön işlem aşamasında karakter değişimi için de kullanılan "·" karakteri kullanılır.

Yeni bir eğitim dokümanı eklendiğinde veya silindiğinde özellik seçimi sonucu meydana gelen sözlük değişebileceğinden sınıf özellik vektörü ve dolayısıyla doküman vektörü oluşturma işlemi yinelenmelidir. Eğitim dokümanları işlemlerinin son adımı olan vektörlerinin oluşturulmasıyla yazılar sınıflandırma işleminde kullanılmak üzere hazırlanmış olmaktadırlar.

Bu çalışmada sözcük ağırlıklandırma işlemi, doküman vektörlerinin oluşturulması esnasında gerçekleşir.

Şekil 4.21.'de, yazıyı bit olarak ifade edebilmek için C_175_bit özellik seçiminden meydana gelmiş sözlük ile kelime-yazı dağılımlarının tutulduğu *tbl_yazi_kelime_EGITIM* tablosundaki yazının eşleştirilebilmesi için gerekli olan SQL ifadesi bulunmaktadır. Bu ifade bütün eğitim dokümanları için tekrarlanır. Bu şekilde bir sınıf özellik vektörü için bütün yazıların doküman vektörleri oluşturulur ve veritabanına kaydedilir. Diğer özellik seçimlerinin sözlüğünü oluşturmak içinde farklı SQL ifadeleri kullanılmıştır.

Şekil 4.17. uygulanması sonucunda sınıf özellik vektörünü oluşturan sözcüklerin id'leri alınmıştır. Sonraki aşama, sınıf özellik vektörüyle Şekil 4.21.'de *yazi_id*'si verilen eğitim dokümanıyla karşılaştırılması ve *ISNULL(adet)* yardımıyla *True/False* şeklinde değer döndürülmesi sağlanır. Eğer değer null ise *True*, değilse *False* değerini alır.

Şekil 4.21., dokümanı bit olarak ağırlıklandırdığı için sözcüğün doküman içerisindeki varlığı veya yokluğu araştırılır. Eğer dönen değer *False* ise 0, *True* ise -1 değerini alır.

Bit ağırlıklandırma, 1 ve 0 kullanılarak yapıldığı için 0 ve -1 değerleri uygun değildir. *ISNULL(adet)+1* kullanılarak null olmayan için 1 ve null olan için 0 değer alması sağlanıp ağırlıklandırmaya uygun hale getirilir.

```
SELECT ISNULL(adet)+1 AS sayac FROM

(SELECT kelime_id FROM

(Şekil 4.17. ile elde edilen sınıf özellik vektörü) AS
tablo1

LEFT JOIN tbl_yazi_kelime_EGITIM
ON (tablo1.kelime_id=tbl_yazi_kelime_EGITIM.kelime_id AND
yazi_id=" & yazi_id & ")
ORDER BY tablo1.kelime_id
```

Şekil 4.21. Sınıf özellik vektörünü oluşturan sözcüklerle eğitim dokümanlarını eşleştirip bit ağırlıklandırmasını gerçekleştiren kod.

Şekil 4.21.'de, sınıf özellik vektöründeki sözcüklerin, dokümanda varlığını araştıran sorgu görülmektedir. Şekil 4.22.'de, eğitim dokümanları vektörlerinin tutulduğu tablolardan kesitler bulunmaktadır.

Sınıf özellik vektörü ve köşe yazısı eşleştirilerek sözcüklerin varlığı, yazı içerisinde araştırılır ve "·" karakteri ile ağırlıklar birleştirilir. *tbl_vektor_C_175_bit* tablosundaki, yazı id alanına, köşe yazısının id değerini; vektör alanına, birleştirilmiş ağırlıkları kaydeder. Şekil 4.22. (a), C_175_bit sınıf özellik vektörü uygulanarak elde edilen doküman vektörlerini göstermektedir.

Şekil 4.22. (a)'daki *yazi_id*'si 4 olan köşe yazısı vektörü incelendiğinde (0·0·0·1·1...) şeklinde devam ettiği görülmektedir. Burada, C_175_bit sınıf özellik vektöründeki 0, 1 ve 2 indisli sözcüklerin bu dokümanda geçmediği, 3 ve 4 indisli sözcüklerin geçtiği söylenebilir.

Şekil 4.22. (b), (c) ve (d) ise Şekil 4.22. (a)'nın, farklı ağırlıklandırmalarla oluşturulan vektörlerine örnek gösterilebilir. Şekil 4.22. (b), (c) ve (d)'de 4 *yazi_id*'li köşe yazısının 0, 1 ve 2 indisli sözcüklerinin 0 değeri aldığı görülmektedir. Aynı sözcük seçimi ve sözcük sayısı kullanılarak elde edilen vektörlerde, aynı sözcükler farklı ağırlıklandırmalarla kullanılmaktadır. *tf* ağırlıklandırmayla elde edilen Şekil 4.22. (c)'de 4 *yazi_id*'li köşe yazısının 0, 1 ve 2 indisli sözcüklerinin 0, 3 indisli sözcüğün 1, 4 indisli sözcüğün 5 *tf* değeriyle ağırlıklandırıldığı görülmektedir.

tbl_vektor_C_175_bit

yazi_id	
1	1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·1·
2	1·1·0·1·0·0·0·0·1·1·0·1·1·0·1·0·1·1·1·0·0·0·1·0·1·0·0·1·1·0·0·0·0·1·1·1·0·0·0·0·1·1·1·1·1·0·1·1·1·0·0·0·
3	1·1·1·1·1·0·1·1·1·1·0·0·1·1·0·1·0·1·1·0·1·1·0·1·1·1·0·0·1·0·1·0·1·1·1·1·0·1·1·0·1·1·1·1·1·1·1·1·1·1·0·0·0·
4	0·0·0·1·1·0·0·0·0·1·1·1·0·0·1·1·1·0·0·1·0·0·0·1·1·0·1·0·1·0·1·0·0·0·0·1·0·1·0·0·0·0·1·1·1·1·0·1·1·0·0·0·
5	0·0·0·1·0·0·0·1·0·1·0·1·1·0·0·1·1·0·0·0·0·1·0·1·0·0·0·0·0·1·1·0·1·0·0·1·0·0·0·0·1·0·1·1·1·0·1·1·0·0·0·0·

(a)

tbl_vektor_C_175_tf_idf

yazi_id	
1	4,773557·2,267287·2,899729·0,9069146·8,937272·0,7166988·1,9828·0,7786796·0,627088·0,008
2	0,6819367·0,9069146·0·1,360372·0·0·0·0·1,881264·0,004347294·0·0,04866248·0,9019934·0·0,4
3	0,6819367·2,267287·1,933152·0,4534573·4,874876·0·2,9742·0,7786796·0,627088·0,01999755·C
4	0·0·0·0,4534573·4,062397·0·0·0·0·0,002608376·0,4412914·0,2919749·0·0·0,4412914·0,537426·2
5	0·0·0·0,9069146·0·0·0·0,3893398·0·0,008694587·0·0,04866248·1,35299·0·0·0,89571·1,415488·0

(b)

tbl_vektor_C_175_tf

yazi_id	
1	7·5·3·2·11·1·2·2·1·10·1·5·3·2·1·4·1·2·5·1·3·3·1·3·2·1·1·1·4·4·4·7·1·1·2·1·1·3·1·4·2·1·7·4·3·2·1·1·6·1·
2	1·2·0·3·0·0·0·0·3·5·0·1·2·0·1·0·5·1·2·0·0·0·1·0·1·0·0·2·5·0·0·0·0·1·2·2·0·0·0·0·1·7·1·8·12·0·1·4·2·0·0
3	1·5·2·1·6·0·3·2·1·23·0·0·3·1·0·2·0·25·1·0·3·3·0·1·2·1·0·0·1·0·1·0·1·3·1·11·0·4·2·0·5·1·6·5·5·4·3·8·1·C
4	0·0·0·1·5·0·0·0·0·3·1·6·0·0·1·3·3·0·0·1·0·0·0·1·2·0·1·0·1·0·1·0·0·0·0·6·0·1·0·0·0·0·6·8·10·12·0·4·3·0·
5	0·0·0·2·0·0·0·1·0·10·0·1·3·0·0·5·2·0·0·0·0·1·0·1·0·0·0·0·0·1·1·0·1·0·0·3·0·0·0·0·2·0·1·22·3·0·1·15·0·C

(c)

tbl_vektor_C_175_idf

yazi_id	
1	0,6819367·0,4534573·0,9665762·0,4534573·0,8124793·0,7166988·0,9913998·0,3893398·0,6270
2	0,6819367·0,4534573·0·0,4534573·0·0·0·0·0,627088·0,0008694587·0·0,04866248·0,4509967·0·0
3	0,6819367·0,4534573·0,9665762·0,4534573·0,8124793·0·0,9913998·0,3893398·0,627088·0,0008
4	0·0·0·0,4534573·0,8124793·0·0·0·0·0,0008694587·0,4412914·0,04866248·0·0·0,4412914·0,1791
5	0·0·0·0,4534573·0·0·0·0,3893398·0·0,0008694587·0·0,04866248·0,4509967·0·0·0,179142·0,7077

(d)

Şekil 4.22. Ağırlıklandırılmış doküman vektörleri. a)bit, b)*tf-idf*, c)*tf*, d)*idf*.

4.6. TEST DOKÜMANLARI (YAZILARI) İŞLEMLERİ

Test dokümanları, yazarların sistemin eğitilmesi için kullanılan yazıları dışındakilerden oluşur. Sınıflandırma işleminde kullanılacak test dokümanlarını veritabanında kayıtlı olanlar ve olmayanlar olarak ikiye ayırabiliriz. Kayıtlı olan dokümanlar, veritabanında tutulmaktadır. Ayrıca eğitim dokümanları işlemlerinin vektörlerin veritabanına kaydedilmesi haricindeki bütün işlemler test dokümanları içinde uygulanır; çünkü test dokümanları için vektör oluşturma işlemi, sınıflandırmaya başlandığında yapılır.

Kayıtlı dokümanlar için, eğitim dokümanlarında yazıların olduğu *tbl_yazi_EGITIM* tablosunun yerine test dokümanlarında *tbl_yazi_TEST* ve kelime-yazı dağılımlarının tutulduğu *tbl_yazi_kelime_EGITIM* yerine *tbl_yazi_kelime_TEST* tablosu kullanılmaktadır.

Kayıtlı olmayan dokümanlar ise anlık sınıflandırma işlemleri için kullanılan dokümanlardır. Bu dokümanların sadece içerikleri, sınıflandırma anında veritabanına kaydedilirler ve ardından kökleri elde edilir. Yeni bir test işlemi başladığında eski doküman silinir.

Sınıflandırılacak köşe yazısı eğer veritabanında kayıtlı ise yazının, özellik seçimi sonucu oluşan sınıf özellik vektörüyle *tbl_yazi_kelime_TEST* tablosu eşleştirilir, elde edilen kelime-yazı dağılımı bilgileri *tbl_incelenecek_yazi_kelime* tablosuna aktarılır ve işlemler bu tablo üzerinden yürütülür.

Doküman kayıtlı değilse, yazı içeriği *tbl_incelenecek_yazi* tablosuna kaydedilip kelime kökleri elde edilir. *tbl_kelime* tablosunda olmayan kökler bu tabloya eklenir. Elde edilen kökler ile kelime-yazı dağılımı yapılıp veriler *tbl_incelenecek_yazi_kelime* tablosuna kaydedilir. Özellik seçimi sonucu oluşan sınıf özellik vektörüyle test dokümanı karşılaştırılarak ağırlıklandırma yapılır ve vektör oluşturulur. Görüldüğü gibi eğitim ve test dokümanları aynı işleme tabi tutulurlar.

Kayıtlı olan ve olmayan dokümanlar için kullanılan tablolar ile diğer tabloların ilişkileri, veritabanı yapısının verildiği Şekil 4.3.'te görülmektedir.

4.7. METİN SINIFLANDIRMA

Bu çalışmada, sınıflandırma işlemi için k-NN (*k=7*) ile Naive Bayes'in Multi-Variate ve Multi-Nominal modelleri kullanılmıştır. k-NN için *k* değeri sabit olsa da çalışma anında değişkenlik gösterecek şekilde programlanmıştır. Şöyle ki; *k* değerinin 7 seçildiği varsayıldığında, e0ğer sınıflandırmaya dahil edilecek son vektör olan 7. vektörden sonra 7. vektörle aynı benzerliğe ve farklı sınıfa sahip vektör veya vektörler varsa *k* sayısı vektör sayısı kadar arttırılır. Bu işlem, *k=7* için

uygulanmıştır. Bu, sınıflandırmaya dahil edilecek vektörleri belirler ve bu vektörlere göre karar verme işlemini gerçekleştirir.

Bilindiği üzere k-NN, sınıflandırma işleminde kararı eldeki dokümanlardan en fazla sayıda var olan sınıf yönünde vermektedir. Sınıflandırma aşamasında k-NN için eğer maksimum doküman sayısını bulunduran birden fazla sınıf varsa dokümanın sınıfını belirleyememiştir; örneğin sınıflandırma aşamasında $k=7$ için 3 spor, 3 sağlık ve 1 eğitim sınıfı belirlenmiş olduğunu varsayalım; maksimum sınıf sayısı olan 3'ten 2 adet sınıf vardır. Bu tür sınıf ataması yapılmamış dokümanların sınıfı *Karışık* olarak atanmıştır. Bir sınıflandırıcının sınıf belirleyememesi nedeniyle sınıfını *Karışık* olarak atadığı bir dokümana başka bir sınıflandırıcı farklı metotlar kullanıldığından sınıf ataması gerçekleştirebilir.

k-NN, $k=7$ alınarak şu şekilde uygulanmıştır:

- Bit ağırlıklandırması Euclid, Cosine ve ağırlıklı oylamaları ile,
- *tf-idf* ağırlıklandırması Euclid, Cosine ve ağırlıklı oylamaları ile uygulanmıştır.

Naive Bayes'in, Multi-Variate ve Multi-Nominal modelleri şu şekilde uygulanmıştır:

- Bit ağırlıklandırması Multi-Variate modelle,
- *tf* ağırlıklandırması Multi-Nominal modelle,
- *idf* ağırlıklandırması Multi-Variate ve Multi-Nominal modelleriyle,
- *tf-idf* ağırlıklandırması Multi-Variate ve Multi-Nominal modelleriyle uygulanmıştır.

Eşitlik 3.9 bit, Eşitlik 3.12 ise *tf* olarak ağırlıklandırılmış sözcükler ile çalışılmak üzere tasarlanmışlardır. Fakat çalışmamamızda Multi-Variate model ile bit ağırlıklandırmaya, Multi-Nominal model ile *tf* ağırlıklandırmaya ek olarak *tf-idf* ağırlıklandırmaları kullanılmıştır. Bu sebeple ve Eşitlik 3.9'daki sözcüğün bit ağırlığı olan wb_{w_i} yerine, *idf* ağırlıklandırma için idf_{w_i},*tf-idf* ağırlıklandırma için tf_{w_i} kullanılarak sınıflandırma gerçekleştirilmiştir. Eşitlik 3.12'deki Multi-Nominal

modelin *tf* ağırlığı ile yapılan sınıflandırmada, sözcüğün frekans ağırlığı olan tf_{w_i} yerine, *idf* ağırlılığı için idf_{w_i}, *tf-idf* ağırlılığı için tf_{w_i} kullanılmıştır.

Aynı benzerlik oranına sahip iki sınıf varsa dokümanın sınıfı *Karışık* olarak belirlenmiştir. k-NN ve Naive Bayes'in bazı uygulamalarında, test dokümanıyla herhangi bir eğitim dokümanı arasında benzerlik bulunamamıştır. Bu tür dokümanların sınıfı *Atanmamış* olarak belirlenmiştir.

Farklı sınıf özellik vektörleri kullanılarak 105 adet sınıflandırma sonucu elde edilmiştir. Başarı sonuçları % cinsinden verilmiştir.

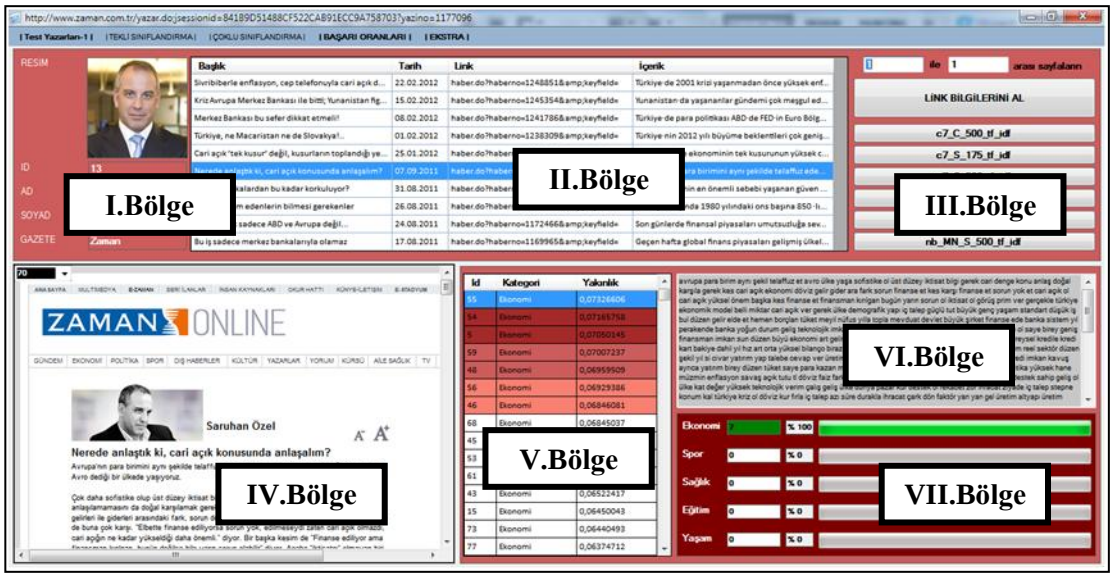
Sınıflandırma işlemi ekranına Şekil 4.4.'te gösterilen "*Madencilik İşlemleri*" seçeneğiyle ulaşılır. İşlem, Şekil 4.23.'teki ekran görüntüsü verilen menülerden "*Tekli Sınıflandırma*" ve "*Çoklu Sınıflandırma*" ile gerçekleştirilir. "*Tekli Sınıflandırma*" sadece bir yazının anlık sınıflandırılmasında, "*Çoklu Sınıflandırma*" ise veritabanında kayıtlı olan test yazılarının tamamının sınıflandırılmasında kullanılır.

Ekonomi alanında yazılar yazan yazarın, 07.09.2011 tarihli yazısının k-NN (*k=7*) algoritması, Cosine benzerliği ve *tf-idf* ağırlıklandırmasıyla sınıflandırılması işlemi ekran görüntüsü Şekil 4.23.'te verilmiştir.

Bu ekran görün 7 bölge ve menüden oluşmaktadır:

- 1.Bölge; seçilen yazarın bilgilerinin görüntülendiği alandır.
- 2.Bölge; seçilen yazarın veritabanında kayıtlı yazılarının görüntülendiği veya sitesinden alınan yazı bilgilerinin görüntülendiği alandır.
- 3.Bölge; seçilen yazarın yazı bilgilerinin sitesinden alınma işleminin gerçekleştiği ve en yüksek sınıflandırma başarısına sahip k-NN (*tf-idf* ağırlandırmanın, *k=7* değerine göre Cosine benzerlik hesabı kullanılarak ve Naive Bayes'in Multi-Nominal modeliyle yapılan sınıflandırma) sınıflandırmaların bulunduğu alandır.
- 4.Bölge; web sayfalarının görüntülendiği alandır.

- 5.Bölge; Naive Bayes sınıflandırıcıda bir fonksiyonu olmayan bu alan, k-NN sınıflandırıcısında eğitim yazılarının test yazısına olan uzaklıklarını yakından uzağa doğru göstermektedir.
- 6.Bölge; Sınıflandırılacak yazı içeriğini oluşturan kelimelerin köklerinin gösterildiği alandır.
- 7.Bölge; Sınıflandırmada ortaya çıkan sonuç burada gösterilmektedir.
- Menü; yazar seçimi, sözcük *idf* değerinin hesaplanması, vektör oluşturulması, tekil ve çoklu sınıflandırma yapılması ve sınıflandırma başarılarının görüntülenme işlemlerinin yapıldığı alandır.



Şekil 4.23. Sınıflandırma işlemi ekran görüntüsü.

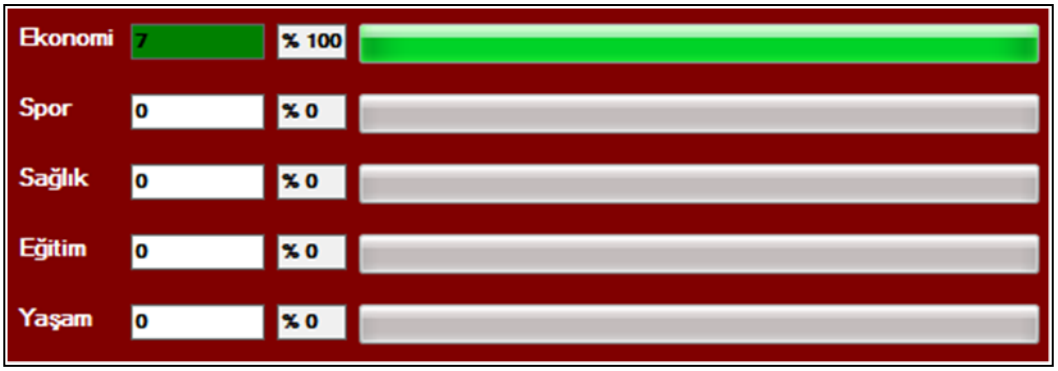
k-NN ve Cosine kullanılarak yapılan sınıflandırma işleminin görüntülendiği Şekil 4.24. (a), eğitim yazılarının *id*'sini, sınıflarını ve sınıflandırılacak yazı arasındaki benzerlik değerini göstermektedir. *yazi_id* 5 olan eğitim yazısı test yazısına en yakın yazı olduğu görülmektedir. Renklendirme işlemi ise *k=3*, *k=5* ve *k=7* değerleri için gerçekleştirilmiştir. Renkli alan karar verme işleminde kullanılan dokümanlardır.

Şekil 4.24. (b), sınıflandırma sonucunun görüntülendiği alandır. Şekil, yapılan sınıflandırmada dikkate alınacak 7 dokümanın *Ekonomi* yazısı olduğunu ve % alanı ise yazının *Ekonomi* sınıfında olma ihtimalini göstermektedir. Arka planı

renklendirilmiş olan sınıf test yazısının sınıflandırma sonucu karar verdiği sınıfını göstermektedir.

Id	Kategori	Yakınlık
5	Ekonomi	0,0549867
54	Ekonomi	0,05489668
48	Ekonomi	0,0532975
55	Ekonomi	0,05316354
68	Ekonomi	0,0528596
45	Ekonomi	0,05210762
46	Ekonomi	0,05190745
59	Ekonomi	0,05116191
61	Ekonomi	0,05087343
53	Ekonomi	0,05067036
56	Ekonomi	0,04983103

(a)



(b)

Şekil 4.24. Bölgelerin ekran görüntüsü. a) 5.Bölge, b) 7.Bölge.

Sınıflandırma için kullanılan teknikler Çizelge 4.11. ve Çizelge 4.12.'de verilmiştir. Çizelge 4.11., k-NN sınıflandırıcısının kullandığı *k* değerini, sözcük ağırlıklandırmasını, benzerlik hesabını ve kullanılan sınıf özellik vektörlerini göstermektedir.

Çizelge 4.11. k-NN sınıflandırma bilgileri.

k	**Sözcük Ağırlığı**	**Benzerlik Hesabı**	**Kullanılan Sınıf Özellik Vektörleri**
7	Bit	Euclid	C_175_bit, C_350_bit, C_500_bit, S_175_bit, S_350_bit, S_500_bit, _bit
7	Bit	Euclid (Ağırlıklı Oylama)	C_175_bit, C_350_bit, C_500_bit, S_175_bit, S_350_bit, S_500_bit, _bit
7	*tf-idf*	Euclid	C_175_*tf-idf*, C_350_*tf-idf*, C_500_*tf-idf*, S_175_*tf-idf*, S_350_*tf-idf*, S_500_*tf-idf*, 175_*tf-idf*, 350_*tf-idf*, 500_*tf-idf*, _*tf-idf*
7	Bit	Cosine	C_175_bit, C_350_bit, C_500_bit, S_175_bit, S_350_bit, S_500_bit, _bit
7	Bit	Cosine (Ağırlıklı Oylama)	C_175_bit, C_350_bit, C_500_bit, S_175_bit, S_350_bit, S_500_bit, _bit
7	*tf-idf*	Cosine	C_175_*tf-idf*, C_350_*tf-idf*, C_500_*tf-idf*, S_175_*tf-idf*, S_350_*tf-idf*, S_500_*tf-idf*, 175_*tf-idf*, 350_*tf-idf*, 500_*tf-idf*, _*tf-idf*

Çizelge 4.12. Naive Bayes sınıflandırma bilgileri.

Model	**Sözcük Ağırlığı**	**Kullanılan Sınıf Özellik Vektörleri**
Multi-Variate	Bit	C_175_bit, C_350_bit, C_500_bit, S_175_bit, S_350_bit, S_500_bit, _bit
Multi-Variate	*idf*	C_175_*idf*, C_350_*idf*, C_500_*idf*, S_175_*idf*, S_350_*idf*, S_500_*idf*, 175_*idf*, 350_*idf*, 500_*idf*, _*idf*
Multi-Variate	*tf-idf*	C_175_*tf-idf*, C_350_*tf-idf*, C_500_*tf-idf*, S_175_*tf-idf*, S_350_*tf-idf*, S_500_*tf-idf*, 175_*tf-idf*, 350_*tf-idf*, 500_*tf-idf*, _*tf-idf*
Multi-Nominal	*tf*	C_175_*tf*, C_350_*tf*, C_500_*tf*, S_175_*tf*, S_350_*tf*, S_500_*tf*, 175_*tf*, 350_*tf*, 500_*tf*, _*tf*
Multi-Nominal	*idf*	C_175_*idf*, C_350_*idf*, C_500_*idf*, S_175_*idf*, S_350_*idf*, S_500_*idf*, 175_*idf*, 350_*idf*, 500_*idf*, _*idf*
Multi-Nominal	*tf-idf*	C_175_*tf-idf*, C_350_*tf-idf*, C_500_*tf-idf*, S_175_*tf-idf*, S_350_*tf-idf*, S_500_*tf-idf*, 175_*tf-idf*, 350_*tf-idf*, 500_*tf-idf*, _*tf-idf*

Çizelge 4.12., Naive Bayes sınıflandırıcısının kullandığı modeli, sözcük ağırlıklandırmasını ve kullanılan sınıf özellik vektörlerini göstermektedir.

4.8. UYGULAMA SONUÇLARI VE DEĞERLENDİRİLMESİ

Kullanılan sınıf özellik vektörleri ve sınıflandırma algoritmalarının elde ettiği sonuçlar çizelgelerle (Bkz. Çizelge 4.13. ile Çizelge 4.29. arası ve Şekil 4.25. ile Şekil 4.29. arası) verilmiştir.

k-NN'nin (*k=7*) bit ağırlıklandırmayla, Euclid uygulanarak yapılan sınıflandırmalarında en yüksek sınıflandırma başarısı %96,40, Cosine uygulanarak yapılan sınıflandırmalarında ise %97,60'dır. Aynı *k* değerinin *tf-idf* ağırlıklandırma kullanılarak yapılan sınıflandırmalarında başarı Euclid için %87,20 iken Cosine için dört farklı sınıf özellik vektöründe %100,00'dür.

Naive Bayes, %100,00'lük başarıyı, C_350_*tf-idf* ve S_500_*tf-idf* sınıf özellik vektörlerinin Multi-Nominal modelle birlikte uygulanmasıyla yakalamıştır.

Bit ağırlıklandırma ile yapılan sınıflandırmaların genel ortalamasında, en yüksek sınıflandırmayı %97,43 başarı ile Naive Bayes'in Multi-Variate modeli gerçekleştirmiştir.

tf-idf ağırlıklandırmalarla yapılan sınıflandırmalarda, en yüksek sınıflandırma ortalaması Naive Bayes'in Multi-Nominal modeliyle ve en düşük sınıflandırmanın *k=7* değeri kullanılarak Euclid ile uygulandığı sınıflandırmalarda gerçekleştirdiği görülmüştür. 175_*tf-idf*, 350_*tf-idf* ve 500_*tf-idf* sınıf özellik vektörleri dikkate alınmadığında en başarılı sınıflandırma, %99,73 olarak *k=7* değeri için Cosine ile ve %99,65 olarak Naive Bayes'in Multi-Nominal modeli ile sağlanmıştır.

idf ağırlıklandırma, Multi-Variate ve Multi-Nominal modellerle uygulanmıştır. %99,20'lik en yüksek sınıflandırma, S_500_*idf* sınıf özellik vektörünün Multi-Nominal modelle; %47,20'lik en düşük sınıflandırma ise 175_*idf* sınıf özellik vektörünün Multi-Variate modelle uygulanmasıyla gerçekleştirilmiştir.

Genel ortalama değerlerine bakıldığında *idf* ağırlıklandırmada Multi-Nominal modelin Multi-Variate modelden daha başarılı olduğu görülmektedir.

Sınıflandırma başarısı hesaplanırken sınıfı *Karışık* ve *Atanamamış* olan dokümanlar başarısız olarak değerlendirilmiştir. 175_*tf-idf*, 350_*tf-idf* ve 500_*tf-idf* sınıf özellik vektörleri kullanarak yapılan sınıflandırma işlemlerinde, sınıf belirleyemediği test dokümanları olmuştur. Bunun nedeni test dokümanı ile eğitim dokümanları arasında benzerlik tespit edememesidir. Test dokümanı vektörü ile eğitim dokümanları vektörlerinde ortak hiçbir sözcük yoktur.

Yapılan uygulamalarda, genelde aynı sınıf özellik vektörleri farklı sınıflandırma işlemleriyle kullanılarak, kullanılan yöntemlerin etkinliklerinin ölçülmesi hedeflenmiştir.

Çizelgelerdeki altı çizili veriler sütundaki, eğik olanlar satırdaki, eğik ve altı çizili olanlar ise hem sütun hem de satırdaki en başarılı sonucu, kalınlar ise sütun ortalamalarını göstermektedir.

Çizelge 4.13., kullanılan sınıf özellik vektörlerinin, algoritmalarla uygulanarak gerçekleşen sınıflandırmalarda, sınıflara göre ortalama başarılarını göstermektedir.

Bit ağırlıklandırma Multi-Nominal dışında bütün sınıflandırmalarla kullanılmasına karşın *idf* ağırlıklandırma Naive Bayes ile ve *tf* ağırlıklandırma ise sadece Multi-Nominal modellerle kullanılmıştır.

175_*tf-idf*, 350_*tf-idf* ve 500_*tf-idf* sınıf özellik vektörlerini kullanarak yapılan sınıflandırma işlemlerinde, sınıf belirleyemediği test dokümanları bulunması nedeniyle başarının düşük olduğu görülmektedir. Sınıf belirleyememesi, test dokümanı ile eğitim dokümanları arasında benzerlik tespit edilememiş olmasındandır: test dokümanı vektörü ile eğitim dokümanları vektörlerinde ortak hiçbir eleman (sözcük) yoktur.

Çizelge 4.13.'te, sınıf özellik vektörlerinin sınıflar bazında ortalama başarıları görüntülenmektedir. %100,00 sınıflandırma, 26 adet *tf*, 9 adet *idf* ve 1 adet *tf-idf* ağırlıklandırma kullanılarak yapılan sınıflandırmalarda gerçekleşmiştir. Bit ağırlıklandırmalarda hiçbir sınıf için %100,00 başarı sağlanamamıştır. *idf* ağırlıklandırma %100,00 sınıflandırma başarısını *Ekonomi, Sağlık* ve *Yaşam* sınıflarında, *tf* ağırlıklandırma *Spor* sınıfları dışındaki bütün sınıflarda, *tf-idf* ağırlıklandırmada ise *Yaşam* sınıfı dışındaki sınıflarda elde etmiştir.

Çizelge 4.13. Sınıf özellik vektörleri kullanılarak yapılan sınıflandırma işlemlerinde sınıfların ortalama başarıları.

Sınıf Özellik Vektörü	**Ekonomi**	**Spor**	**Sağlık**	**Eğitim**	**Yaşam**
C_175_bit	97,20	93,60	98,00	93,60	98,00
C_350_bit	90,80	92,40	97,20	96,80	85,60
C_500_bit	88,80	89,20	94,00	97,20	80,00
S_175_bit	96,80	95,60	98,80	93,20	98,00
S_350_bit	90,00	92,40	96,80	96,80	88,00
S_500_bit	88,40	89,20	92,00	97,60	80,00
_bit	79,20	62,80	74,00	96,80	52,80
Ortalama	**90,17**	**87,89**	**92,97**	**96,00**	**83,20**
C_175_*tf-idf*	92,50	96,00	95,00	94,50	99,50
C_350_*tf-idf*	91,50	89,00	87,00	93,50	96,50
C_500_*tf-idf*	89,00	89,00	83,50	95,50	91,00
S_175_*tf-idf*	90,50	95,50	85,00	92,50	97,00
S_350_*tf-idf*	90,50	97,00	84,00	99,50	100,00
S_500_*tf-idf*	89,50	95,00	82,00	99,00	91,50
175_*tf-idf*	27,00	35,00	54,00	32,00	60,00
350_*tf-idf*	51,00	30,00	81,50	38,50	54,00
500_*tf-idf*	64,50	49,50	62,50	65,00	54,50
_*tf-idf*	83,00	86,50	77,00	97,50	77,00
Ortalama	**76,90**	**76,25**	**79,15**	**80,75**	**82,10**
C_175_*idf*	99,00	97,00	97,00	95,00	100,00
C_350_*idf*	98,00	97,00	98,00	95,00	100,00
C_500_*idf*	99,00	98,00	99,00	96,00	100,00
S_175_*idf*	99,00	98,00	96,00	96,00	100,00

Çizelge 4.13. (devam ediyor).

S_350_*idf*	99,00	98,00	99,00	97,00	100,00
S_500_*idf*	99,00	99,00	99,00	97,00	100,00
175_*idf*	30,00	43,00	68,00	12,00	85,00
350_*idf*	66,00	46,00	79,00	55,00	89,00
500_*idf*	77,00	72,00	88,00	51,00	92,00
_*idf*	100,00	98,00	100,00	90,00	100,00
Ortalama	**86,60**	**84,60**	**92,30**	**78,40**	**96,60**
Genel Ortalama *(tf dahil edilmiştir)*	**86,12**	**83,58**	**89,61**	**85,44**	**88,58**

4.8.1. k-NN (*k=7*) / Bit Ağırlıklandırma

Farklı sınıf özellik vektörlerine göre k-NN algoritması, bit ağırlıklandırma kullanılarak *k=7* için Euclid, Euclid (Ağırlıklı Oylama), Cosine ve Cosine (Ağırlıklı Oylama) ile uygulanmış ve sonuçlar Çizelge 4.14. ve Çizelge 4.15.'de gösterilmiştir.

Euclid, bütün sınıflarda, sınıfını belirleyemediği yazılar için kullandığı *Karışık* sınıfına atadığı yazı bulunmasına rağmen ağırlıklı oylaması, bütün yazılarda sınıf belirleme işlemini gerçekleştirmiştir. Euclid, C_175_bit sınıf özellik vektörü ile 11 dokümanın sınıfını *Karışık* olarak atamış ve başarı oranı %93,20 olmuştur. Buna rağmen Euclid ağırlıklı oylaması, bütün dokümanlara sınıf atamasını yaparak başarı oranını %96,4'e çıkarmıştır. Fakat, başarının artması sınıflandırma yaptığı için değil doğru sınıflandırma yaptığı içindir. Buna karşın, yanlış sınıflandırma yapılması durumunda başarı oranı düşecektir. Sınıflandırma başarısında, *Karışık* olan dokümanlar başarısız olarak değerlendirilmediği takdirde sınıflandırma başarısı %97,60'ye çıkacaktır. Cosine, C_175_bit ile S_350_bit sınıf özellik vektörlerinde ve ağırlıklı oylamasında bütün sınıf özellik vektörlerinde tüm yazıların sınıf atamasını gerçekleştirmiştir.

Aynı sınıf özellik vektörlerinin, farklı uygulamalarda aynı sonuçları vermediği görülmektedir. Örneğin, C_175_bit sınıf özellik vektörü için Euclid %93,20 ve Cosine %97,20 başarı elde etmişlerdir. Bunun sebebi, uygulamaların farklı benzerlik hesabı kullanmalarıdır.

Euclid uygulamalarında en başarılı sonuç, %94,80 ile S_175_bit'ten ve ağırlıklı oylamasında %96,40 ile C_175_bit'ten elde edilmiştir. Cosine ve ağırlıklı oylaması uygulamalarında en başarılı sonuç, %97,60 ile C_350_bit ve S_350_bit'ten elde edilmiştir. Çizelge 4.14.'te, Cosine ve ağırlıklı oylamasında, bütün sözcüklerin dahil edildiği _bit dışındaki sınıflandırmalarda başarı oranının değişmediği görülmektedir.

Çizelge 4.14.'te, bütün sözcükler dikkate alınarak Euclid, Cosine ve ağırlıklı oylaması ile yapılan sınıflandırmalarda başarının düştüğü görülmektedir. Ayrıca her sınıf özellik vektöründe Cosine benzerliğinin Euclid'den daha başarılı sonuçlar elde ettiğini görülmektedir.

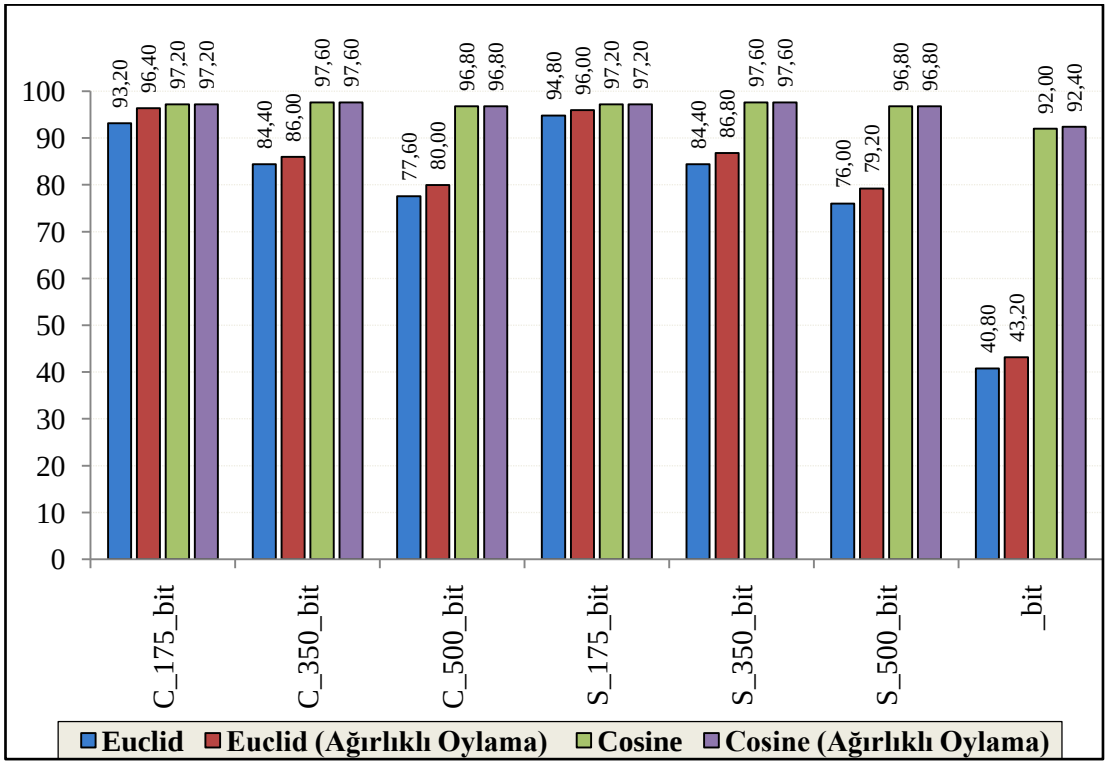
Çizelge 4.14. Kullanılan benzerlik ölçütleri ve bit ağırlıklı sınıf özellik vektörlerine göre sınıflandırma sonuçları (k-NN).

Sınıf Özellik Vektörü	**Euclid**	**Euclid (Ağırlıklı Oylama)**	**Cosine**	**Cosine (Ağırlıklı Oylama)**
C_175_bit	93,20	96,40	*97,20*	*97,20*
C_350_bit	84,40	86,00	*97,60*	*97,60*
C_500_bit	77,60	80,00	*96,80*	*96,80*
S_175_bit	94,80	96,00	*97,20*	*97,20*
S_350_bit	84,40	86,80	*97,60*	*97,60*
S_500_bit	76,00	79,20	*96,80*	*96,80*
Ortalama	**85,07**	**87,40**	**97,20**	**97,20**
_bit	40,80	43,20	92,00	*92,40*
Genel Ortalama	**78,74**	**81,09**	**96,46**	**96,51**

Şekil 4.25., Euclid ve ağırlıklı oylaması ile Cosine ve ağırlıklı oylamasını göstermektedir. Grafikte Cosine ile yapılan sınıflandırma işlemlerinin Euclid'e göre bütün sınıf özellik vektörlerinde daha başarılı olduğu görülmektedir.

Euclid, en yüksek sınıflandırmayı, sınıf özellik vektörünün her sınıftan 175'er kelimenin alınarak gerçekleştiği C_175_bit ve S_175_bit'ten elde etmiştir. Sınıf özellik vektörünün 350, 500 ve bütün sözcüklerden oluştuğu özellik seçimleriyle yapılan sınıflandırmalarda başarı sürekli düşüş göstermiştir. Ağırlıklı oylama

kullanılarak yapılan sınıflandırmalar, Euclid'e göre daha başarılıdır. Cosine'de de, bütün sözcüklerle yapılan sınıflandırmada başarısı düşmüştür.



Şekil 4.25. Kullanılan benzerlik ölçütleri ve bit ağırlıklı sınıf özellik vektörlerine göre sınıflandırma sonuçları grafiksel gösterimi (k-NN).

Çizelge 4.15., sınıflar bazında sınıflandırma sonuçlarının ortalamalarını göstermektedir. Sınıflar içerisindeki en yüksek sonuç, *Ekonomi* sınıfında %98,57 ile Cosine ve ağırlıklı oylaması ile elde edilirken en düşük sonuç, Euclid ağırlıklı oylaması ile gerçekleşen *Yaşam* sınıfındadır. Genel ortalamalara bakıldığında en yüksek sınıflandırma başarısı, *Eğitim* sınıfında olduğu görülmektedir.

Çizelge 4.15. Kullanılan benzerlik ölçütleri ve bit ağırlıklı sınıf özellik vektörlerine göre sınıfların ortalama sınıflandırma sonuçları (k-NN).

Kullanılan Benzerlik Ölçütü	**Ekonomi**	**Spor**	**Sağlık**	**Eğitim**	**Yaşam**
Euclid Sınıflandırmaların Ortalaması	**74,29**	**76,86**	**85,71**	**96,86**	**60,00**

Çizelge 4.15. (devam ediyor).

Euclid (Ağırlıklı Oylama) Sınıflandırmaların Ortalaması	**79,43**	**77,14**	**87,43**	**97,43**	**64,00**
Cosine Sınıflandırmaların Ortalaması	**98,57**	**93,71**	**97,43**	**96,57**	**96,00**
Cosine (Ağırlıklı Oylama) Sınıflandırmaların Ortalaması	**98,57**	**94,00**	**97,43**	**96,57**	**96,00**
Genel Ortalama	**87,72**	**85,43**	**92,00**	**96,86**	**79,00**

4.8.2. k-NN (*k=7*) / *tf-idf* Ağırlıklandırma

k=7 değeri için k-NN algoritması, *tf-idf* ağırlıklandırmasıyla Euclid ve Cosine ile uygulanmış ve sonuçlar Çizelge 4.16. ve Çizelge 4.17.'de gösterilmiştir. Euclid ve Cosine ile veya ağırlıklı oylamalarıyla yapılan sınıflandırmalarda farklı sonuçlar üretilmemesi nedeniyle *tf-idf* ağırlıklandırmayla yapılan sınıflandırmalarda, ağırlıklı oylama uygulanmamıştır.

Euclid ile yapılan sınıflandırmalarda, 175_*tf-idf* sınıf özellik vektörüyle 29 *Atanmamış*, 350_*tf-idf*'te ise 7 *Atanmamış* yazı bulunmaktadır. Cosine ile yapılan sınıflandırmalarda ise 175_*tf-idf* sınıf özellik vektörüyle 29 *Atanmamış*, 350_*tf-idf*'te ise 7 *Atanmamış* yazı bulunmaktadır. Bu sınıf özellik vektörleri dışındakilerle yapılan sınıflandırma çalışmalarında, bütün yazılara sınıf atama işlemi gerçekleştirilmiştir. 175_*tf-idf*, 350_*tf-idf* ve 500_*tf-idf* sınıf özellik vektörlerinde sınıfı atanamamış yazılar bulunması, bu sınıf özellik vektörlerinin sözlük oluştururken sınıflarda en yüksek *idf* değerine sahip sözcükleri sınıflandırma işlemlerinde kullanmalarından ve eğitim yazıları ile sınıflandırılacak test yazıları arasında ortak sözcük bulunmamasından kaynaklanır.

Çizelge 4.16.'ya göre Euclid, *k=7* değeri için en yüksek sınıflandırma başarısını %87,20 ile C_175_*tf-idf* sınıf özellik vektörüyle gerçekleştirmiştir. Fakat, Euclid'in *tf-idf* ağırlıklandırma kullanılarak yapılan sınıflandırmalarda başarılı olduğu söylenemez. Genel ortalama değerlerine bakıldığında sınıflandırma işlemi sonuçlarının %53,08 gibi çok düşük bir seviyede olduğu görülmektedir.

Cosine, *k=7* değeri için %100'lük başarı C_500_*tf-idf*, S_175_*tf-idf* ve S_500_*tf-idf* ve *tf-idf*'ten elde edilmiştir. C_175_*tf-idf* sınıf özellik vektöründe %99,20'lik başarı elde edilmiştir. Ayrıca, C_350_*tf-idf* ve S_350_*tf-idf* sınıf özellik vektörlerinde %99,60 başarılı sınıflandırma sağlanmıştır. Çalışmalar boyunca en yüksek sınıflandırma %100 başarı ile Cosine ve *tf-idf* ağırlıklandırmayla elde edilmiştir.

Çizelge 4.14. ve Çizelge 4.16. incelendiğinde Cosine'nin aynı boyuttaki vektörlerin farklı ağırlıklandırmalarla farklı sonuçlar ürettiğini görebiliriz. _bit ve _*tf-idf* ile oluşturulmuş vektörler aynı sözcükleri seçen, dolayısıyla aynı boyutta vektörlerdir. Bu ikisi arasındaki fark, sözcüklerin farklı yöntemlerle ağırlandırılmış olmalarıdır. Cosine, _bit özelliği ile %92 başarı elde ederken aynı işlem _*tf-idf* ağırlıklandırma ve *k=7* için %100 başarı elde etmiştir. Bu da, sözcük ağırlıklandırmanın, sözcüğün nasıl temsil edileceğinin, ne kadar önemli olduğunu ve sonucu nasıl etkilediğini göstermesi açısından önemlidir.

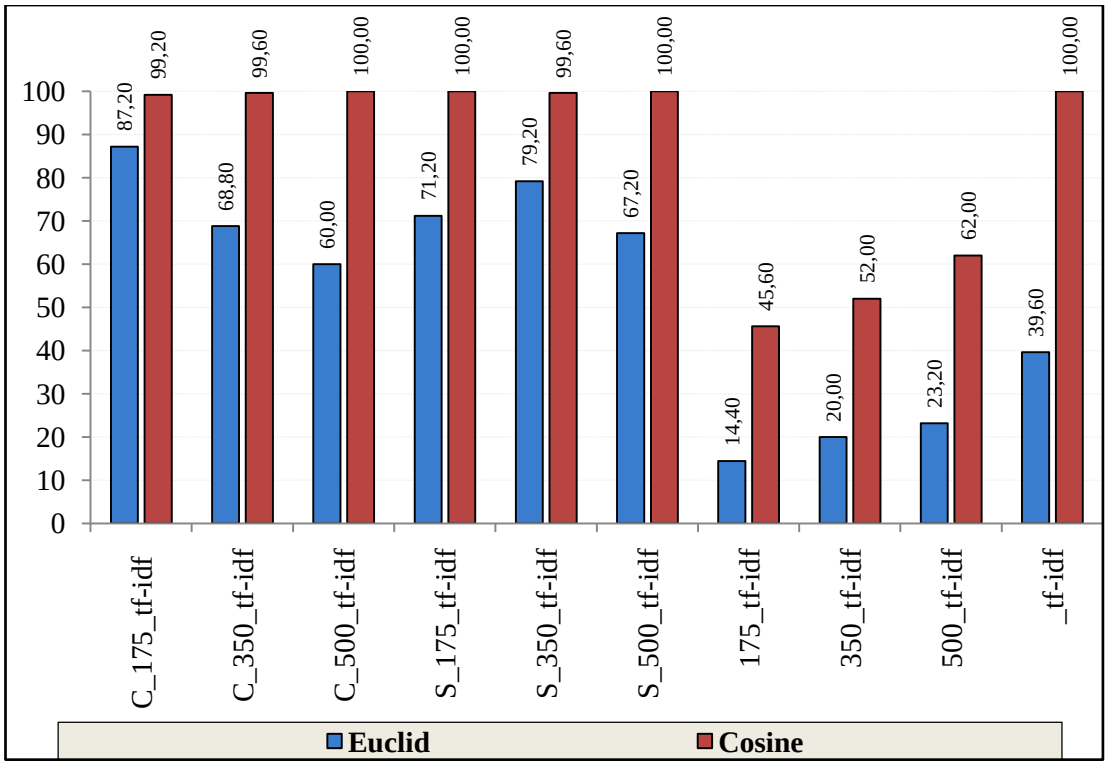
Çizelge 4.16. Kullanılan benzerlik ölçütleri ve *tf-idf* ağırlıklı sınıf özellik vektörlerine göre sınıflandırma sonuçları (k-NN).

Sınıf Özellik Vektörü	**Euclid**	**Cosine**
C_175_*tf-idf*	87,20	*99,20*
C_350_*tf-idf*	68,80	*99,60*
C_500_*tf-idf*	60,00	*100,00*
S_175_*tf-idf*	71,20	*100,00*
S_350_*tf-idf*	79,20	*99,60*
S_500_*tf-idf*	67,20	*100,00*
Ortalama	**72,27**	**99,73**
175_*tf-idf*	14,40	*45,60*
350_*tf-idf*	20,00	*52,00*
500_*tf-idf*	23,20	*62,00*
Ortalama	**19,20**	**53,20**
_*tf-idf*	39,60	*100,00*
Genel Ortalama	**53,08**	**85,80**

Şekil 4.25.'teki verilere göre Euclid'te sözcük sayısı arttıkça sınıflandırma başarısının düştüğü görülmüştü. Şekil 4.26.'da da, S_350_*tf-idf* haricinde başarı

düşmüştür. Ayrıca Cosine'nin aksine, Euclid mesafe ölçümü kullanılarak yapılan sınıflandırmalarda bit ağırlıklandırılmış vektörlerin *tf-idf* ağırlıklandırılmış vektörlere göre daha başarılı olduğu görülmektedir.

Şekil 4.25.'te, Euclid ve Cosine'de 175_*idf*, 350_*idf* ve 500_*idf* sınıf özellik vektörleriyle yapılan sınıflandırmaların başarısı oldukça düşüktür. Bunun sebebinin, bu özellik vektörlerinin oluşturulmasında, sınıfta en yüksek *idf* değerine göre seçim yapılması, dolayısıyla eğitim ve test doküman vektörleri arasında ortak sözcük bulunamaması olduğu ifade edilmişti.



Şekil 4.26. Kullanılan benzerlik ölçütleri ve *tf-idf* ağırlıklı sınıf özellik vektörlerine göre sınıflandırma sonuçları grafiksel gösterimi (k-NN).

Çizelge 4.17., *tf-idf* ağırlıklandırma ve *k=7* değeri için, sınıflar bazında sınıflandırma sonuçlarının ortalamalarını göstermektedir. Sınıflar içerisindeki en yüksek sonuç, sınıflandırma sonuçlarındaki bariz farklılıkları nedeniyle Euclid ve Cosine için ayrı değerlendirilmiştir. Euclid, en yüksek başarıyı %77,80 ile *Eğitim*, en düşük başarıyı ise %40,80 ile *Sağlık* sınıfında elde etmiştir. Cosine ise en yüksek başarıyı %90,40 ile *Sağlık*, en düşük başarıyı ise %81,00 ile *Eğitim* sınıfında elde etmiştir. Genel

ortalamalara bakıldığında, en yüksek sınıflandırma başarısının *Eğitim* sınıfında olduğu görülmektedir. Ayrıca, bundan önceki sınıflandırmalarda da en yüksek sınıf *Eğitim* olmuştur.

Çizelge 4.17. Kullanılan benzerlik ölçütleri ve *tf-idf* ağırlıklı sınıf özellik vektörlerine göre sınıfların ortalama sınıflandırma sonuçları (k-NN).

Kullanılan Benzerlik Ölçütü	**Ekonomi**	**Spor**	**Sağlık**	**Eğitim**	**Yaşam**
Euclid Sınıflandırmaların Ortalaması	**42,80**	**52,80**	**40,80**	**77,80**	**51,20**
Cosine Sınıflandırmaların Ortalaması	**86,40**	**82,40**	**90,40**	**81,00**	**88,80**
Genel Ortalama	**64,60**	**67,60**	**65,60**	**79,40**	**70,00**

4.8.3. Naive Bayes / Multi-Variate / Bit Ağırlıklandırma

Bu modelle, bit ağırlıklı olarak ifade edilmiş vektörlerle köşe yazısı sınıflandırması uygulanmış ve sonuçlar Çizelge 4.18.'de görüntülenmiştir. Bu uygulamada bütün dokümanlara sınıf ataması gerçekleştirilmiştir; sınıfı *Karışık* veya *Atanmamış* olan köşe yazısı bulunmamaktadır.

Çizelge 4.18. incelendiğinde, en başarılı sınıflandırmayı S_500_bit sınıf özellik vektörüyle ve en düşük sınıflandırmayı C_175_bit sınıf özellik vektörüyle gerçekleştiği görülmektedir.

Yapılan bütün ağırlıklandırma ve sınıflandırmalar dikkate alındığında, genel ortalama değerlerine göre en yüksek sınıflandırma başarısının, %97,43 oranıyla Multi-Variate modelin bit ağırlıklandırmadan ile elde edildiği görülmektedir. Ayrıca bit ağırlıklandırma ile yapılan sınıflandırmalar içerisinde de S_500_bit sınıf özellik vektörünün %98,40 ile en başarılı sınıflandırmayı gerçekleştirmiştir.

Sınıflandırma sonuçlarına göre sınıf özellik vektörleri arasında sınıflandırma başarılarının birbirlerine yakın olduğu, en yüksek ile en düşük sınıflandırma başarısı arasında %2,00 gibi düşük bir fark bulunduğu görülmektedir. Bit ağırlıklandırma ile

yapılan diğer sınıflandırmalarda ise bu farkın en az %5,20 ve en fazla %54,00, *tf-idf* ağırlıklandırma ile yapılan sınıflandırmada en az %43,20 ve en fazla %72,80, *idf* ağırlıklandırma ile yapılan sınıflandırmada %51,20, *tf* ağırlıklandırma ile yapılan sınıflandırmada %42,40 olduğu görülmektedir.

Çizelge 4.18.'deki sonuçların, Çizelge 4.14.'teki sonuçlardan, Euclid ve ağırlıklı oylamasında bütün sınıf özellik vektörlerinde, Cosine ve ağırlıklı oylamasında C_500_bit, S_500_bit ve _bit sınıf özellik vektörlerinde daha başarılı olduğu görülmüştür. Ayrıca, bu iki çizelgedeki en yüksek sınıflandırmayı Multi-Variate gerçekleştirmiştir.

Çizelge 4.18. Multi-Variate model ve bit ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırma sonuçları.

Sınıf Özellik Vektörü	**Multi-Variate**
C_175_bit	96,40
C_350_bit	97,20
C_500_bit	98,00
S_175_bit	97,20
S_350_bit	97,60
S_500_bit	98,40
Ortalama	**97,47**
_bit	97,20
Genel Ortalama	**97,43**

Eğitim sınıfı, bu sınıflandırma işleminden önceki sınıflandırmaların aksine en yüksek sınıflandırma başarısının elde edildiği sınıf olamamıştır. Sonuçları Çizelge 4.19.'da verilen bu sınıflandırmada, *Ekonomi* ve *Yaşam*, %100,00 ile en yüksek, *Eğitim* ise %88,29 ile en düşük başarıyı yakalayan sınıf olmuştur.

Çizelge 4.19. Multi-Variate model ve bit ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırmalarda, ortalama sınıf başarıları.

	Ekonomi	**Spor**	**Sağlık**	**Eğitim**	**Yaşam**
Genel Ortalama	**100,00**	**97,71**	**96,86**	**92,57**	**100,00**

4.8.4. Naive Bayes / Multi-Variate / *idf* Ağırlıklandırma

idf ağırlıklandırmanın, Naive Bayes'in Multi-Variate modeliyle uygulandığında elde edilen sonuçlar Çizelge 4.20. ve 4.21.'de gösterilmiştir.

Bu modelde sınıflandırma gerçekleşirken sözcük ağırlığı bit olarak değil *idf* olarak alınmıştır. *idf* ağırlıklandırma ile yapılan işlemlerde, sözcüğün vektör oluşturmadan önce hesaplanan *idf* değeri kullanılır. 175_*idf*, 350_*idf* ve 500_*idf* sınıf özellik vektörleri dikkate alınmadığında, *idf* ağırlıklandırma ile elde edilen Çizelge 4.20.'nin, %97,67 ortalama sınıflandırma başarısı ve sınıf özellik vektörleri uygulanarak alınan sonuçlar bakımından Çizelge 4.18.'e göre daha başarılı olduğu görülmektedir.

Çizelge 4.20. Multi-Variate model ve *idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırma sonuçları.

Sınıf Özellik Vektörü	**Multi-Variate**
C_175_*idf*	96,80
C_350_*idf*	97,20
C_500_*idf*	98,00
S_175_*idf*	97,20
S_350_*idf*	98,40
S_500_*idf*	98,40
Ortalama	**97,67**
175_*idf*	47,20
350_*idf*	64,00
500_*idf*	75,20
Ortalama	**62,13**
_*idf*	96,40
Genel Ortalama	**86,88**

Çizelge 4.21.'de görüldüğü gibi en başarılı ve en başarısız sınıflar *Yaşam* ve *Eğitim* sınıflarıdır. Çizelge 4.20.'de olduğu gibi *Eğitim* sınıfı en başarısız sınıf olmuştur.

Çizelge 4.21. Multi-Variate model ve *idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırmalarda, ortalama sınıf başarıları.

	Ekonomi	**Spor**	**Sağlık**	**Eğitim**	**Yaşam**
Genel Ortalama	**87,00**	**83,40**	**90,80**	**76,80**	**96,40**

4.8.5. Naive Bayes / Multi-Variate / *tf-idf* Ağırlıklandırma

Sözcük ağırlıklandırmanın sonuç üzerine etkisi Cosine ve Multi-Variate modelde görülmüştür. Çizelge 4.22.'nin, Çizelge 4.18. ve Çizelge 4.20. ile incelendiğinde, *tf-idf* ağırlıklandırmanın sonuca olumlu katkısı daha belirginleşmiştir. Bit ağırlıklandırma bir sözcüğün yazıdaki varlığıyla, *idf* ağırlıklandırma, yazıda kaç kez geçtiğiyle değil sadece diğer dokümanlarda geçip geçmediğiyle ilgilenmektedir. Buna karşın *tf-idf* ağırlıklandırma, hem sözcüğün yazı içerisinde kaç kez geçtiğiyle hem de diğer yazılarda geçip geçmediğiyle ilgilenir. *tf-idf* ağırlıklandırma, sözcüğün yazıdaki değerini daha iyi yansıttığından sınıflandırma başarısını arttırmaktadır.

Multi-Variate modelle uygulanan ağırlıklandırmalar şu şekilde değerlendirmeye alınmıştır: Bit ağırlıklandırmada *topla_ekonomi = topla_ekonomi + 1*, *idf* ağırlıklandırmada *topla_ekonomi = topla_ekonomi + idf_degeri*, *tf-idf* ağırlıklandırmada *topla_ekonomi = topla_ekonomi +(tf*idf)_degeri* şeklinde kullanılır.

Şekil 4.44.'te görülen *v_bit(i1, cls)=1* ifadesi, *i1* indisli *v_bit* eğitim dokümanı vektörünün sınıfının (*cls*) *Ekonomi* (1-*Ekonomi*) olup olmadığının kontrolünde kullanılır. *v_bit(i1,i2)*'deki *i2* sözcük indisini, *v_TMP* ise test dokümanını tutan vektörü ifade eder. Eğer eğitim dokümanının sınıfı 1 ise ve sözcük hem eğitim hem de test dokümanında bulunuyorsa sınıftaki değerleri toplayan *topla_ekonomi* değişkeninin değeri, eğitim dokümanının değeri kadar arttırılır. Bu işlem bütün eğitim dokümanları ve sözcükler için tekrarlanır.

```
If v_bit(i1,cls)=1 And v_bit(0,i2)>0 And v_bit(i1,i2)>0 Then
   topla_ekonomi = topla_ekonomi + v_bit(i1,i2)
Else If v_bit(i1,cls)=2 And v_TMP(0,i2)>0 And v_bit(i1,i2)>0  Then
   topla_spor = topla_spor + v_bit(i1,i2)
Else If v_bit(i1,cls)=3 And v_TMP(0,i2)>0 And v_bit(i1,i2)>0  Then
   topla_saglik = topla_saglik + v_bit(i1,i2)
Else If v_bit(i1,cls)=4 And v_TMP(0,i2)>0 And v_bit(i1,i2)>0  Then
   topla_egitim = topla_egitim + v_bit(i1,i2)
Else If v_bit(i1,cls)=5 And v_TMP(0,i2)>0 And v_bit(i1,i2)>0  Then
   topla_yasam = topla_yasam + v_bit(i1,i2)
End If
```

Şekil 4.27. Multi-Variate model sınıflandırma işleminde sözcük kontrolü.

Sınıf özellik vektörlerine göre uygulanarak elde edilen sonuçlar Çizelge 4.22.'de, sınıflara göre sonuçlar ise Çizelge 4.23.'te verilmiştir. Çizelge 4.22.'deki sonuçların, Euclid ve Cosine'in *tf-idf* ağırlıklandırma ve *k=7* alınarak yapılan sınıflandırma işlemi sonuçlarının görüntülendiği Çizelge 4.16. ile kıyaslanması halinde, Euclid ile yapılan sınıflandırmalarının tamamından, Cosine'in 175_*tf-idf*, 350_*tf-idf* ve 500_*tf-idf* sınıf özellik vektörleriyle yapılan sınıflandırmalarından daha başarılı olduğu görülmüştür.

Çizelge 4.22. Multi-Variate model ve *tf-idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırma sonuçları.

Sınıf Özellik Vektörü	**Multi-Variate**
C_175_*tf-idf*	96,80
C_350_*tf-idf*	97,60
C_500_*tf-idf*	98,80
S_175_*tf-idf*	97,60
S_350_*tf-idf*	98,40
S_500_*tf-idf*	98,40
Ortalama	**97,93**
175_*tf-idf*	49,60
350_*tf-idf*	64,80
500_*tf-idf*	76,80
Ortalama	**63,73**

Çizelge 4.22. (devam ediyor).

_tf-idf	98,40
Genel Ortalama	**87,72**

Çizelge 4.23.'te verilen sonuçlara göre en başarılı sınıf *Yaşam*, en başarısız sınıf ise Çizelge 4.19.'da olduğu gibi *Eğitim* sınıfı olmuştur.

Çizelge 4.23. Multi-Variate model ve *tf-idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırmalarda, ortalama sınıf başarıları.

	Ekonomi	**Spor**	**Sağlık**	**Eğitim**	**Yaşam**
Genel Ortalama	**87,80**	**83,60**	**91,60**	**79,60**	**96,00**

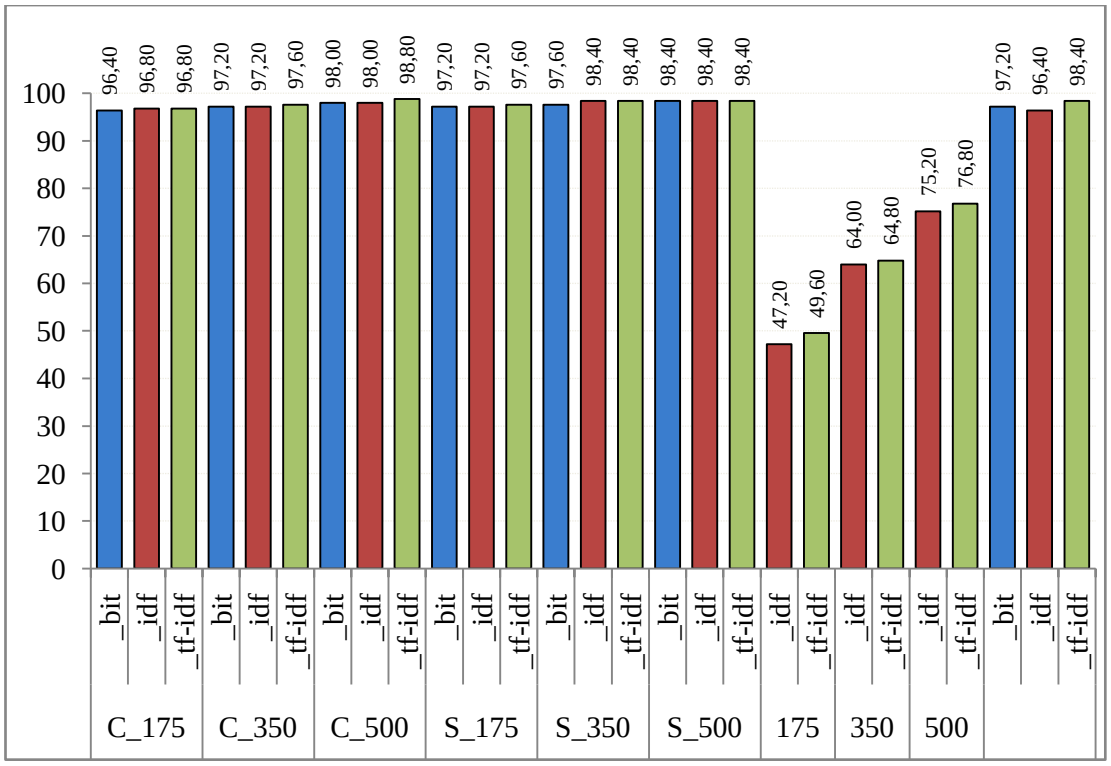
Multi-Variate model ile *tf-idf* ağırlıklandırma kullanırken, Şekil 4.27.'de gösterilen *v_bit(0,i2)>0 And v_bit(i1,i2)>0* kullanılmıştır. Eğer bu ifade yerine, *v_bit(0,i2)=v_bit(i1,i2)* kullanılsaydı sözcük eşleştirmesini gerçekleştiremezdi. *v_bit(0,i2)>0 And v_bit(i1,i2)>0* ifadesi, sözcüğün hem eğitim hem test dokümanlarında geçtiğini belirler. 175*_tf-idf*, 350*_tf-idf* ve 500*_tf-idf* sınıf özellik vektörleri dikkate alınmadığında, Multi-Variate modelle en yüksek sınıflandırmayı gerçekleştiren ağırlıklandırma olur.

Şekil 4.28.'de, Multi-Variate modelin bit, *idf* ve *tf-idf* ağırlıklandırmayla elde edilen doküman vektörleri kullanılarak yapılan sınıflandırma sonuçları görüntülenmektedir. Sonuçlar; *COUNT* (C), *SUM* (S) ve alınacak sözcük sayısı şeklinde gruplandırılmıştır. Gruplandırmalar, aynı sözcüklerin farklı şekillerde ağırlıklandırmalarını kullanmaktadırlar. Örneğin, C_175 grubunda, 437 sözcükle işlemler gerçekleşir. Bu grubun alt sınıflandırmaları arasındaki fark, sözcüklerin bit, *tf* ve *tf-idf* olarak ağırlıklandırılmış olmalarıdır.

Şekil 4.28. incelendiğinde, en başarılı grubun S_500 olduğu görülmektedir. Bu grubun üç alt sınıflandırmasında da %98,40 oranında sınıflandırma

gerçekleştirilmiştir. Şekil 4.28.'deki isimsiz olan en son grupta bütün sözcükler çalışmaya dahil edilmiştir. Bu sınıflandırmalardaki yüksek başarı dikkat çekicidir.

Şekil 4.25., sınıflandırma işleminin bit ağırlıklandırılmış; Şekil 4.26. ise *tf-idf* ağırlıklandırılmış vektörlerle sınıflandırılma sonuçlarını gösterir. Şekil 4.26.'da Cosine ile yapılan sınıflandırmaların Şekil 4.25.'ten daha başarılıdır. Şekil 4.28.'de de, bütün sınıflandırma işlemlerinde, *tf-idf* ağırlıklandırılarak oluşturulan vektörlerle yapılan sınıflandırmaların, diğer ağırlıklandırmalardan aynı veya daha yüksek sınıflandırma başarısına sahip olduğu görülmektedir.



Şekil 4.28. Multi-Variate model ve bit, *idf*, *tf-idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırma sonuçlarının grafiksel gösterimi.

4.8.6. Naive Bayes / Multi-Nominal / *tf* Ağırlıklandırma

Bu model, *tf* olarak ifade edilmiş vektörlere uygulanmış ve sonuçlar Çizelge 4.24.'te gösterilmiştir. *tf* ağırlıklandırma, sadece Naive Bayes'in Multi-Nominal modeliyle uygulanmıştır. Çizelge 4.24.'ün, *tf-idf* ağırlıklandırmayla elde edilen sonuçları gösteren Çizelge 4.16. ile kıyaslandığında bütün sınıf özellik vektörlerinde

Euclid'den, 175_*tf-idf*, 350_*tf-idf* ve 500_*tf-idf* sınıf özellik vektörlerinde ise Cosine'den daha iyi olduğu görülmektedir.

Multi-Nominal modelin 175_*tf*, 350_*tf*, 500_*tf* sınıf özellik vektörleri dışındaki bütün sınıf özellik vektörlerinde, *tf* ağırlıklandırma ile yapılan sınıflandırmaları, Multi-Variate modelin bit, *idf* ve *tf-idf* ağırlıklandırma ile yapılan sınıflandırmalarına göre daha başarılı olmuştur.

Çizelge 4.24. Multi-Nominal model ve *tf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırma sonuçları.

Sınıf Özellik Vektörü	**Multi-Nominal**
C_175_*tf*	98,80
C_350_*tf*	99,60
C_500_*tf*	99,20
S_175_*tf*	99,60
S_350_*tf*	99,60
S_500_*tf*	99,60
Ortalama	**99,40**
175_*tf*	57,20
350_*tf*	68,40
500_*tf*	77,20
Ortalama	**67,60**
_*tf*	99,60
Genel Ortalama	**89,88**

Çizelge 4.25.'te Multi-Nominal modelin *tf* ağırlıklandırmasıyla elde edilen sınıflara göre ortalama sonuçlar verilmiştir.

Çizelge 4.25. Multi-Nominal model ve *tf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırmalarda, ortalama sınıf başarıları.

	Ekonomi	**Spor**	**Sağlık**	**Eğitim**	**Yaşam**
Genel Ortalama	**90,80**	**85,60**	**94,00**	**86,60**	**92,40**

4.8.7. Naive Bayes / Multi-Nominal / *idf* Ağırlıklandırma

Bu model, *idf* ağırlıklandırma ile sınıflandırılmış olup ve sonuçlar Çizelge 4.26.'da, sınıfların ortalama sınıflandırma başarıları Çizelge 4.27.'de gösterilmiştir. Çizelge 4.26. ile Çizelge 4.16. kıyaslandığında bütün sınıf özellik vektörlerinde Euclid'den, 175_*tf-idf*, 350_*tf-idf* ve 500_*tf-idf* sınıf özellik vektörlerinde Cosine'den daha doğru sınıflandırma gerçekleştirmiştir.

Çizelge 4.26. ile Çizelge 4.24. kıyaslandığında ise sadece 350_*idf* sınıf özellik vektöründe başarılı olmuştur.

Çizelge 4.26. Multi-Nominal model ve *idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırma sonuçları.

Sınıf Özellik Vektörü	**Multi-Nominal**
C_175_*idf*	98,40
C_350_*idf*	98,00
C_500_*idf*	98,80
S_175_*idf*	98,40
S_350_*idf*	98,80
S_500_*idf*	99,20
Ortalama	**98,60**
175_*idf*	48,00
350_*idf*	70,00
500_*idf*	76,80
Ortalama	**64,93**
_*idf*	98,80
Genel Ortalama	**88,52**

Çizelge 4.27. Multi-Nominal model ve *idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırmalarda, ortalama sınıf başarıları.

	Ekonomi	**Spor**	**Sağlık**	**Eğitim**	**Yaşam**
Genel Ortalama	**86,20**	**85,80**	**93,80**	**80,00**	**96,80**

4.8.8. Naive Bayes / Multi-Nominal / *tf-idf* Ağırlıklandırma

Multi-Nominal model, *tf-idf* olarak ifade edilmiş vektörlerle uygulanmış ve sınıf özellik vektörlerine göre sonuçlar Çizelge 4.28.'de, sınıflara göre sonuçlar ise Çizelge 4.29.'da gösterilmiştir.

Çizelge 4.28., Multi-Nominal modelin *tf-idf* ağırlıklandırmasıyla elde edilmiş sonuçlardır. k-NN (*k=7*) ve Cosine'in *tf-idf* ağırlıklandırmayla, bütün test dokümanlarının %100,00 doğrulukla sınıflandırılması işlemi, Multi-Nominal'in *tf-idf* ağırlıklandırmasıyla da iki farklı sınıf özellik vektöründe gerçekleştirilmiştir.

Çizelge 4.28. Multi-Nominal model ve *tf-idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırma sonuçları.

Sınıf Özellik Vektörü	**Multi-Nominal**
C_175_*tf-idf*	98,80
C_350_*tf-idf*	100,00
C_500_*tf-idf*	99,60
S_175_*tf-idf*	99,60
S_350_*tf-idf*	99,60
S_500_*tf-idf*	100,00
Ortalama	**99,65**
175_*tf-idf*	56,80
350_*tf-idf*	67,20
500_*tf-idf*	74,80
Ortalama	**66,27**
_*tf-idf*	98,80
Genel Ortalama	**89,52**

Çizelge 4.29.'da verilen sonuçların, Çizelge 4.19.'da ki sonuçların tamamına göre, Çizelge 4.25.,'teki sonuçların *Spor* dışındaki sınıflarına göre başarısız olmuştur. Çizelge 4.21. ve Çizelge 4.23.'teki sonuçlarda *Yaşam* sınıfı dışındaki sınıflarda başarılı olmuştur.

Çizelge 4.29. Multi-Nominal model ve *tf-idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırmalarda, ortalama sınıf başarıları.

	Ekonomi	**Spor**	**Sağlık**	**Eğitim**	**Yaşam**
Genel Ortalama	**90,60**	**86,20**	**93,80**	**84,60**	**92,40**

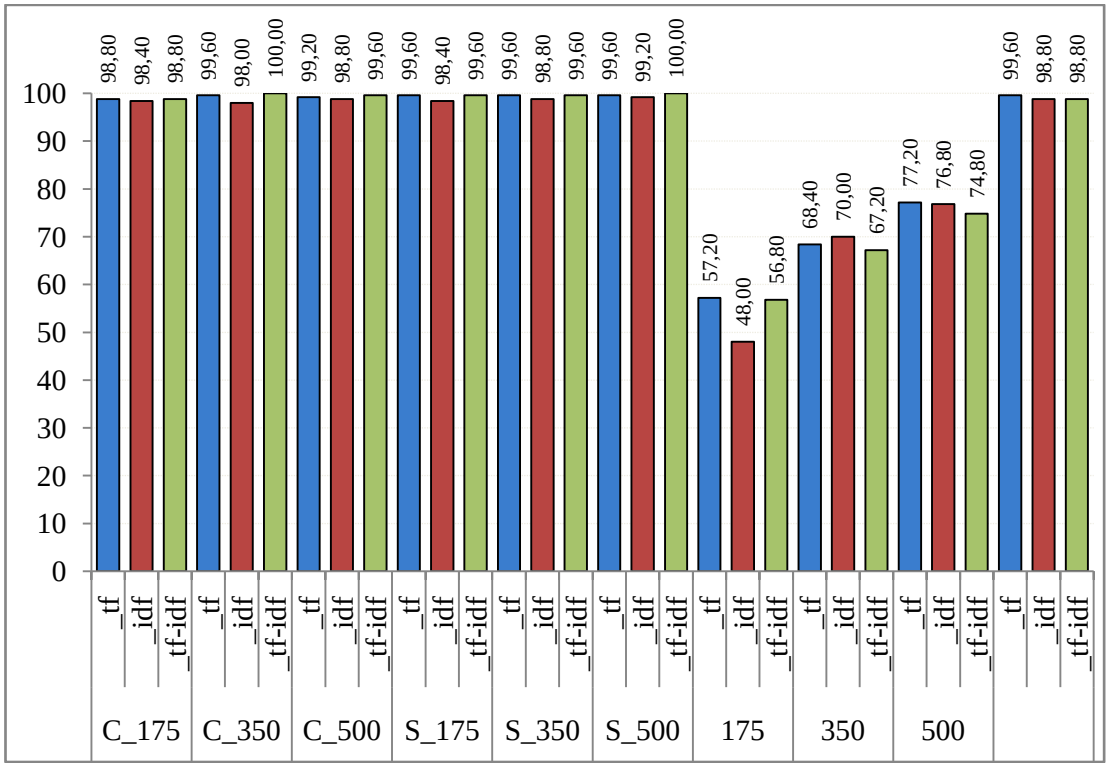
Bit ağırlıklandırılmış vektörleri Multi-Variate, *tf* ağırlıklandırılmış vektörleri Multi-Nominal, *idf* ve *tf-idf* ağırlıklandırılmış vektörler ise her iki model kullanmıştır.

Şekil 4.29.'da da sonuçlar; Şekil 4.28.'de olduğu gibi *COUNT* (C), *SUM* (S) ve alınacak sözcük sayısı şeklinde gruplandırılmıştır. İsimsiz olan son grupta bütün sözcükler sınıflandırma işleminde kullanılmıştır.

Her ne kadar Multi-Nominal model *tf*'e göre tasarlanmış olsa da, *tf-idf* ağırlıklandırmanın *tf* ve *idf* ağırlıklandırmayla, C ve S gruplarının tamamında daha başarılı olduğu görülmektedir. Şekil 4.29. incelendiğinde, Şekil 4.28.'de olduğu gibi en başarılı grubun S_500 olduğu görülmektedir.

Bu çalışmada, toplam 6 sınıflandırma işleminde %100,00 başarı sağlanmıştır. Bu sınıflandırmaların tamamında *tf-idf* ağırlıklandırılmış vektörler kullanılmıştır. 6 sınıflandırmanın 4 tanesi, Cosine benzerlik hesabının k-NN ($k=7$) ile uygulandığında elde edilmiştir. Diğer 2 sınıflandırma ise Multi-Nominal model uygulanarak sağlanmıştır. Bu, *tf-idf* ağırlıklandırmanın ne kadar etkili bir ağırlıklandırma olduğunu ortaya koymaktadır. Ayrıca bu çalışma, bir ağırlıklandırmaya göre tasarlanan bir sınıflandırma metodunun, diğer ağırlıklandırmalara da uygulanarak başarılı sonuçlar alınabileceğini ortaya koymuştur.

Sınıflandırma başarısında *tf-idf* ağırlıklandırmayı *tf* ağırlıklandırma izlemektedir. Multi-Nominal'de *idf*'in diğer ağırlıklandırmalara göre başarısız olmasının nedeni, sözcüğün yazı içerisinde kaç kez geçtiğinin dikkate alınmaması, işlemleri sadece *idf* değeri üzerinden gerçekleştirmesidir.



Şekil 4.29. Multi-Nominal model ve *tf, idf, tf-idf* ağırlıklı sınıf özellik vektörleriyle gerçekleşen sınıflandırma sonuçlarının grafiksel gösterimi.

Multi-Nominal model ile elde edilen Şekil 4.29.'daki sonuçlarla, Multi-Variate model ile elde edilen Şekil 4.28.'deki sonuçlar kıyaslanmıştır.

Aynı ağırlıklandırma kullanılarak yapılan sınıflandırmalar kıyaslandığında, Multi-Variate modelin, sadece 500 grubundaki *tf-idf* ağırlıklandırmayla yapılan sınıflandırma işleminde daha başarılı olduğu görülmektedir. Grupların genel ortalama değerlerine bakıldığında da bütün gruplarda Multi-Nominal modelle yapılan sınıflandırmaların daha iyi sınıflandırma gerçekleştirdiği görülmektedir. Aynı ağırlıklandırma kullanılarak gerçekleştirilen sınıflandırma ortalamaları şöyledir; *idf* ile Multi-Variate model %86,88, Multi-Nominal model %88,52; *tf-idf* ile Multi-Variate model %87,72, Multi-Nominal model %89,52; bit ile Multi-Variate model %86,88, Multi-Nominal model %88,52. Görüldüğü gibi ağırlıklandırmalarda da Multi-Nominal model oldukça başarılıdır.

Bu sonuçlar, sınıflandırma işleminde Multi-Nominal modelin, Multi-Variate modelden daha etkin olduğunu ortaya koymaktadır.

BÖLÜM 5

SONUÇ VE ÖNERİLER

İnternet kullanımının her geçen gün artması, internet ortamında bulunan verilerin miktarını arttırmıştır. Bu veriler, metinsel içerik barındırdığı için işlenmeden önce hazırlanması gerekmektedir. İnternette bulunan haber, blog ve köşe yazıları gibi metinsel içeriğe sahip verilerin de hazırlanması metin madenciliği yöntemi ile yapılmaktadır. Metin madenciliği, bu tür verileri veri madenciliği tekniklerinin uygulanabileceği yapıya dönüştürme işlemini gerçekleştirir.

Metinleri, önceden tanımlanmış sınıflara atama işlemine metin sınıflandırma denir. Elle sınıflandırma işleminin yavaş ve pahalı olması ile sınıflandırmada sürekli aynı sonucun alınamama ihtimali, manüel sınıflandırmayı tercih edilen sınıflandırma olmaktan çıkarmaktadır. Ayrıca veri miktarının büyüklüğü, sınıflandırma işleminin bilgisayar programları aracılığıyla yapılmasını gerekli hale getirmiştir.

Ülke gündeminin değişken olması, köşe yazarlarının alanları dışında yazıyor olmaları, insanları yazıları okumadan hangi alanla ilgili yazı olduğunu bilmeleri, bilgi edinmenin yanında bilgiye erişim hızının da önemli bir unsur olduğunu ortaya koymaktadır.

Ülke gündeminin değişkenliği, köşe yazarlarının alanları dışında yazmaları ve yazılarının okunmadan önce hangi sınıfla ilgili olduklarının belirlenmesi gibi nedenler, bilgi edinme ve bilgiye erişim hızının önemli olduğu günümüzde, sınıflandırma işleminin haber sitelerine ve köşe yazılarına uygulanması gerekliliğini ortaya koymuştur.

Bu çalışma, metin madenciliği yöntemi ile haber sitelerindeki köşe yazılarının sınıflandırılmasını gerçekleştirmektedir. Öncelikle veri madenciliği ve metin

madenciliği konuları alt başlıkları ile verilmiş, ve bir uygulama yazılımı geliştirilmiştir.

Bu yazılımda eğitim ve test dokümanlarının alınmasından sınıflandırılmasına kadar olan bütün işlemler gerçekleştirilmiştir. Köşe yazılarındaki kelime köklerinin bulunması için ayrıca bir yazılım geliştirilmiş ve program üzerinden çağrılarak çalıştırılmıştır.

6 farklı gazeteden 25 yazar ile sistem eğitilmiştir. Sınıflandırma işlemi, 50 yazar ile gerçekleştirilebilir. Gazetelerin kendi sitelerinden, 07.10.2010 ile 27.02.2012 tarihleri arasındaki köşe yazıları, eğitim ve test dokümanı olarak kullanılmıştır. Eğitim dokümanı olarak her sınıftan eşit sayıda olmak üzere toplam 500 köşe yazısı kullanılmıştır. Test dokümanı için her sınıfta 50'şer köşe yazısı kullanılmıştır.

Bit, *tf*, *idf* ve *tf-idf* ağırlıklandırma, 37 sınıf özellik vektörü ve farklı sınıflandırma algoritmaları kullanılarak 105 sınıflandırma gerçekleştirilmiştir. k-NN algoritmasında, Euclid ve Cosine ile birlikte *k=7* değeri alınarak sınıflandırma gerçekleştirmiştir. Naive Bayes, sınıflandırma işlemi için Multi-Variate ve Multi-Nominal modelleri kullanılmıştır. *Ekonomi*, *Spor*, *Sağlık*, *Eğitim* ve *Yaşam* sınıfları, yazıların atanacağı sınıflardır.

Uygulama sonuçlarına göre *tf-idf* ağırlıklandırma, Cosine ve *k=7* değeri kullanılarak yapılan sınıflandırmalarda 4 farklı sınıf özellik vektöründe bütün yazıların sınıflandırılması %100,00 doğrulukla gerçekleştirilmiştir. 2 farklı sınıf özellik vektöründe, *tf-idf* ağırlıklandırmanın Multi-Nominal modelle uygulandığı sınıflandırmalarda da %100,00 başarı sağladığı görülmüştür.

Bundan sonraki çalışmalarda, yönelik olarak, sınıflandırma işleminde karar vericiye yol gösteren eğitim dokümanlarının ve sınıflandırma başarısının değerlendirilmesinde kullanılan test dokümanlarının sayısı arttırılabilir. Bu sayede, sınıflandırma işlemlerinde kullanılan yöntemlerin etkinliği daha iyi gözlemlenebilir.

Çalışmada kullanılan bit, *tf-idf*, *tf* veya *idf* dışında sözcük ağırlıklandırma yöntemleri ve özellik seçim teknikleri kullanılarak sınıflandırma işlemleri gerçekleştirilebilir.

Sözcüklerin bulunduğu bölüme göre, bölüm katsayısı kadar değerlendirilmesi şeklinde çalışmalar yapılabilir. Örneğin, köşe yazısı bölümlere ayrılarak katsayılarla bölüm önem sırası oluşturulur. Bölümlerin oluşturulmasında ilk ve son paragrafın yüksek önemli, orta paragrafların düşük önemli bölüm olarak atanması ve ağırlıklandırılması gibi bir yaklaşım benimsenebilir. Bu yaklaşımdan farklı olarak, yazının bölümlendirilmesinde, yazı içeriğindeki sözcük sayısına göre oransal bir bölüm oluşturma tekniği de kullanılabilir. Köşe yazısının, yazıdaki sözcük sayısına göre 5'e bölündüğü varsayılırsa, ilk ve son bölümde bulunan sözcüklerin, diğer bölümlerdeki sözcüklere göre daha değerli olacak şekilde katsayı uygulanması şeklinde bir uygulama geliştirilebilir.

KAYNAKLAR

Aas, K. and Eikvil, L., "Text categorization: A survey". ***Norwegian Computing Center***, 1-37 (1999).

Ahmadi, A., Fotouhi, M. and Khaleghi, M., "Intelligent classification of web pages using contextual and visual features", ***Applied Soft Computing***, 11 (2): 1638-1647 (2011).

Alpaydın, E., "Zeki veri madenciliği: Ham veriden altın bilgiye ulaşma yöntemleri", ***Bilişim 2000 Eğitim Semineri***, İstanbul, (2000).

Amasyalı, M.F. ve Yıldırım, T., "Otomatik haber metinleri sınıflandırma", ***Signal Processing and Communications Applications (SIU 2004), 2004 IEEE 12th Conference on***, Aydın, 224-226 (2004).

Amasyalı, M.F. ve Diri, B., "Automatic Turkish text categorization in terms of author, genre and gender", ***11th International Conference on Applications of Natural Language to Information Systems (NLDB 2006)***, Klagenfurt, Austria, 221-226 (2006).

Aşlıyan, R. ve Günel, K., "Metin içerikli Türkçe doküman sınıflandırılması", ***Akademik Bilişim 2010***, Muğla Üniversitesi, Muğla, (2010).

Bao, F., He, X. and Zhao, F., "Applying data mining to the geosciences gata", ***Physics Procedia***, 33: 685-689 (2012).

Chen, C.-M., Lee, H.-M. and Tan, C.-C., "An intelligent web-page classifier with fair feature-subset selection", ***Engineering Applications of Artificial Intelligence***, 19 (8): 967-978 (2006).

Cohen, W.W. and Hirsh, H., "Joins that generalize: Text classification using WHIRL", ***In Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining***, Wuhan, China, 169-173 (1998).

Coomans, D. and Massart, D.L., "Alternative k-nearest neighbour rules in supervised pattern recognition : Part 1. k-Nearest neighbour classification by using alternative voting rules", ***Analytica Chimica Acta***, 136: 15-27 (1982).

Çalışkan, B., "Metin madenciliği ile metin sınıflandırma", Yüksek Lisans Tezi, ***Marmara Üniversitesi Sosyal Bilimleri Enstitüsü***, İstanbul, 200-204 (2008).

Çiflikli, C. ve Kahya-Özyirmidokuz, E., "Implementing a data mining solution for enhancing carpet manufacturing productivity", ***Knowledge-Based Systems***, 23 (8): 783-788 (2010).

Dasarathy, B.V., "Nearest-neighbor classification techniques", ***IEEE Computer Society Press,*** Los Alamitos, California, (1991).

Dilmen, N.E., "Yönetenler açısından Türkiye'deki internet gazeteleri ve haber portalları üzerine bir değerlendirme", ***İstanbul Üniversitesi İletişim Fakültesi Dergisi***, 22: 96 (2005).

Dolgun, M.Ö., Özdemir, T.G. ve Oğuz, D., "Veri madenciliği'nde yapısal olmayan verinin analizi: Metin ve web madenciliği", ***İstatistikçiler Dergisi***, 2 (8): 48-58 (2009).

Durmaz, O. ve Bilge, H.Ş., "Metin sınıflandırmada boyut azaltmanın etkileri ve özellik seçimi", ***Signal Processing and Communications Applications (SIU 2011), 2011 IEEE 19th Conference on***, Antalya, 21-24 (2011).

Erhardt, R.A., Schneider, R. and Blaschke, C., "Status of text mining techniques applied to biomedical text", ***Drug Dicovery Today***, 11 (7-8): 315-25 (2006).

Eyheramendy, S., Lewis, D.D. and Madigan, D., "On the naive bayes model for text categorization", ***In Proceedings of Artificial Intelligence and Statistics***, 3-6 (2003).

Fan, W., Wallace, L., Rich, L. and Zhang, Z., "Tapping into the power of text mining", ***Communications of the ACM***, 49 (9): 76-82 (2005).

Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P., "From data mining to knowledge discovery in databases", ***AI Magazine***, 17 (3): 37-54 (1996).

Frawley, W.J., Piatetsky-Shapiro, G. and Matheus, C. J., "Knowledge discovery in databases: an overview", ***AI Magazine***, 13 (3): 57-70 (1991).

Fuller, C.M., Biros, D.P. and Delen, D., "An investigation of data and text mining methods for real world deception detection", ***Expert Systems with Applications***, 38 (7): 8392-8398 (2011).

Gieger, C., Deneke, H. and Fluck, J., "The future of textmining in genome-based clinical research", ***BIOSILICO***, 1 (3): 97-102 (2003).

Gongde, G., Wang, H., Bell, D., Bi, Y. and Greer, K., "Using knn model for automatic text categorization", ***Soft Computing-A Fusion of Foundations, Methodologies and Applications***, 10 (5): 423-430 (2006).

Güran A., Akyokuş S., Güler N. ve Gürbüz Z., "Turkish text categorization using n-gram words", ***International Symposium on INnovations in Intelligent SysTems and Applications (INISTA 2009)***, Trabzon, (2009).

Han, J. and Kamber, M., "Data Mining: Concepts and Techniques 2nd ed.", ***Morgan Kaufmann Publishers***, San Francisco, 5-7, 105-106, 348-350 (2006).

Hand, D., Mannila, H. and Smyth, P., "Principles of Data Mining", ***MIT Press***, Cambridge, 31-33, 456-464 (2001).

Hayes, P.J., Knecht, L.E. and Cellio, M.J., "A news story categorization system", ***2nd Applied Natural Language Processing Conference (ANLP 1988)***, Austin, Texas, 9-17 (1988).

Hsu, C.-H., "Data mining to improve industrial standards and enhance productionand marketing: An empirical study in apparel industry", ***Expert Systems with Applications***, 36 (3-1): 4185-4191 (2009).

Huang, W., Xu, L., Duan, J. and Lu, Y., "Chinese web-page classification study", ***International Conference on Control and Automation (ICCA 2007), IEEE International Conference on***, Hong Kong, 1553-1558 (2007).

İlhan, U., "Application Of KNN and FPTC based text categorization algorithms to Turkish news reports", ***Bilkent Üniversitesi***, (2001).

İnternet: Akşam Gazetesi, **http://www.aksam.com.tr**, (2012).

İnternet: Data Mining Community's, "KDNuggets polls-industries / fields where you applied data mining in 2011", **http://www.kdnuggets.com/polls/2011/industries-applied-anaytics-data-mining.html**, (2012).

İnternet: Habertürk Gazetesi, **http://www.haberturk.com**, (2012).

İnternet: Milliyet Gazetesi, **http://www.milliyet.com.tr**, (2012).

İnternet: Posta Gazetesi, **http://www.posta.com.tr**, (2012).

İnternet: Star Gazetesi, **http://www.stargazete.com.tr**, (2012).

İnternet: Zaman Gazetesi, **http://www.zaman.com.tr**, (2012).

İnternet: Zemberek, "Açık Kaynak Kodlu Türkçe Doğal Dil İşleme Kütüphanesi", **http://code.google.com/p/zemberek/downloads/detail?name=zemberek-2.1.1.zip**, (2011).

Jackson, P. and Moulinier, I., "Natural Language Processing for Online Applications Text Retrieval Extraction and Categorization", ***John Benjamins Publishing Company***, 32-35 (2002).

Karaca, M.F. ve Görgünoğlu, S., "ColumnREADY: İnternet gazeteleri köşe yazılarını hazırlama uygulama yazılımı", ***Akademik Bilişim 2012***, Uşak Üniversitesi, Uşak, (2012).

Karadağ, A. ve Takçı, H., "Metin madenciliği ile benzer haber tespiti", ***Akademik Bilişim 2010***, Muğla Üniversitesi, Muğla, (2010).

Konchady, M., "Text Mining Application Programming 1st ed.", ***Charles River Media***, Boston, (2006).

Lagus, K., "Text Mining with the WEBSOM", ***Helsinki University of Technology***, Finland, (2000).

Li, L., Huang, Y.-G. and Liu, Z.-W., "Chinese text classification for small sample set", ***The Journal of China Universities of Posts and Telecommunications,*** 18 (1): 83-89 (2011).

Liang, C.-Y., Guo, L., Xia, Z.-J., Nie, F.-G., Li, X.-X., Su, L. and Yang, Z.-Y., "Dictionary-based text categorization of chemical webpages", ***Information Processing & Management,*** 42 (4): 1017-1029 (2006).

Liao, S.-H., Chu, P.-H. and Hsiao, P.-Y., "Datamining techniques and applications – A decade review from 2000 to 2011", ***Expert Systems with Applications,*** 39 (12): 11303-11311 (2012).

Losiewicz, P., Oard, D.W. and Kostoff, R.N., "Textual data mining to support science and technology management", ***Journal of Intelligent Information Systems 2000***, 15 (2): 99-119 (2000).

Mengle, S.S.R., Goharian, N. and Platt, A., "FACT: Fast algorithm for categorizing Text", ***5th IEEE International Conference on Intelligence and Security Informatics***, New Brunswick, New Jersey, USA, 308-315 (2007).

Mitchell, T.M., "Machine Learning 1st ed.", ***McCraw Hill***, New York, (1997).

Nanopoulos, A., Theodoridis, Y. and Manolopoulos, Y., "C2P: Clustering based on closest pairs", ***Proceedings of the 27th International Conference on Very Large Data Bases (VLDB 2001)***, Roma, Italy, 331-340 (2001).

Piatetsky-Shapiro, G., "Knowledge discovery in real databases: A report on the IJCAI-89 Workshop", ***AI Magazine***, 11 (5): 68-70 (1991).

Pilavcılar, İ.F., "Metin madenciliği ile metin sınıflandırma", Yüksek Lisans Tezi, ***Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü***, İstanbul, 4 (2007).

Pons-Porrata, A., Berlanga-Llavorib, R. and Ruiz-Shulcloper, J. "Topic discovery based on textmining techniques", ***Information Processing & Management***, 43 (3): 752-768 (2007).

Roiger, R.J. and Geatz, M.W., "Data Mining: A Tutorial-Based Primer", ***Addison Wesley***, (2003).

Salton, G., Wong, A. and Yang, C.S., "A vector space model for automatic indexing" ***Communications of the ACM***, New York, 18 (11): 613-620 (1975).

Salton, G. and Buckley, C., "Term-weighting approaches in automatic text retrieval", ***Information Processing and Management***, 24 (5): 513-523 (1988).

Sanwaliya, A., Shanker, K. and Misra, S.C., "Categorization of news articles: A model based on discriminative term extraction method", ***Advances in Databases Knowledge and Data Applications (DBKDA 2010), 2010 Second International Conference on***, French Alps, France, 145-154 (2010).

Sebastiani, F., "Machine learning in automated text categorization", ***ACM Computing Surveys***, New York, 34 (1): 1-47 (2002).

Shilakes, C.C. and Tylman, J., "Enterprise information portals", ***Merrill Lynch***, (1998).

Soergel, D., "Organizing Information: Principles of Data Base and Retrieval Systems", ***Academic Press***, Florida, (1985).

Soucy, P. and Mineau, G.W., "A simple knn algorithm for text categorization". ***Proceedings IEEE International Conference on Data Mining (ICDM '01)***, California, 647-648 (2001).

Soucy, P. and Mineau, G.W., "Beyond *TFIDF* weighting for text categorization in the vector space Model", ***Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2005)***, Edinburgh, 1130-1135 (2005).

Susan, P., "Effective use of the KDD process and data mining for computer performance professionals", ***27th International Computer Measurement Group Conference***, California, 611-620 (2001).

Tan, A.-H., "Text Mining: The state of the art and the challenges", ***In Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases***, Beijing, 71-76 (1999).

Toraman, Ç., Can, F. ve Koçberber, S., "Developing a text categorization template for Turkish news portals", ***International Symposium on INnovations in Intelligent SysTems and Applications (INISTA 2011)***, İstanbul, 379-383 (2011).

Visa, A., "Technology of Text Mining", ***Tampare University of Technology***, 1-11 (2001).

Wakil, M.E., "Introducing text mining", ***IEEE International Symposium on Information Theory (ISIT 2002)***, Lausanne, (2002).

Yang, Y. and Pedersen, J.O., "A comparative study on feature selection in text categorization", ***Proceedings Fourteenth International Conference on Machine Learning (ICML'97)***, Nashville, Tennessee, 412-420 (1997).

Yang, Y. and Liu, X., "A re-examination of text categorization methods", ***Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999)***, Berkeley, CA, USA, 42-49 (1999).

Yin, S., Qui, Y. and Ge, J., "Research and realization of text mining algorithm on web", ***Computational Intelligence and Security Workshops (CISW 2007). International Conference on***, Silicon Vally, USA, 413-416 (2007).

Zobel, J. and Moffat, A., "Exploring the similarity space", ***SIGIR Forum***, 32 (1): 18-34 (1998).

ÖZGEÇMİŞ

Mehmet Fatih KARACA 1980 yılında Tokat'ta doğdu; ilk ve orta öğrenimini aynı şehirde tamamladı. 1998 yılında Kocaeli Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi Bilgisayar Öğretmenliği Bölümü'nde öğrenime başlayıp 2002 yılında mezun oldu ve aynı yıl göreve başladı. 2002-2009 yılları arasında Milli Eğitim Bakanlığı bünyesinde öğretmenlik görevini sürdürdü. Askerliğini Kara Kuvvetleri Komutanlığı Balıkesir Astsubay Hazırlama Okulu'nda öğretmen teğmen olarak yaptı. 2009 yılında Gaziosmanpaşa Üniversitesi Erbaa Meslek Yüksekokulu Bilgisayar Teknolojisi ve Programlama Programı'nda öğretim görevlisi olarak göreve başladı ve halen aynı yerde çalışmaya devam etmektedir.

ADRES BİLGİLERİ

Adres : Gaziosmanpaşa Üniversitesi
Erbaa Meslek Yüksekokulu
Erbaa / TOKAT

Tel : (505) 312 1737
E-posta : mfkaraca@gmail.com