YAPAY ÖĞRENME YÖNTEMLERİ İLE TÜRKÇE METİNLERDE DUYGU ANALİZİ

Batıbay Düven
17 14 02 103

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Yüksek Lisans Programı
Danışman: Dr. Öğr. Üyesi Volkan Tunalı

İstanbul
T.C. Maltepe Üniversitesi
Lisansüstü Eğitim Enstitüsü
Ağustos, 2021

YAPAY ÖĞRENME YÖNTEMLERİ İLE TÜRKÇE METİNLERDE DUYGU ANALİZİ

Batıbay Düven
17 14 02 103
Orcid: 0000-0002-0214-760X

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Yüksek Lisans Programı
Danışman: Dr. Öğr. Üyesi Volkan Tunalı

İstanbul
T.C. Maltepe Üniversitesi
Lisansüstü Eğitim Enstitüsü
Ağustos, 2021



JÜRİ VE ENSTİTÜ ONAYI

Bu belge, Yükseköğretim Kurulu tarafından 19.01.2021 tarihli *"Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge"* ile bildirilen 6689 Sayılı Kişisel Verilerin Korunması Kanunu kapsamında gizlenmiştir.

ETİK İLKE VE KURALLARA UYUM BEYANI

Bu belge, Yükseköğretim Kurulu tarafından 19.01.2021 tarihli *"Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge"* ile bildirilen 6689 Sayılı Kişisel Verilerin Korunması Kanunu kapsamında gizlenmiştir.

TEŞEKKÜR

Tez boyunca bilgilerini ve yardımlarını esirgemeyen danışman hocam Sayın Dr. Öğr. Üyesi Volkan Tunalı'ya, destek aldığım arkadaşım Serkan Anıl Saygılı'ya ve beni bu yaşa kadar okutan, her koşulda yanımda olan çok kıymetli aileme teşekkürlerimi sunarım.

Batıbay Düven

Ağustos, 2021

ÖZ

YAPAY ÖĞRENME YÖNTEMLERİ İLE TÜRKÇE METİNLERDE DUYGU ANALİZİ

Batıbay Düven
Yüksek Lisans Tezi
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Yüksek Lisans Programı
Danışman: Dr. Öğr. Volkan Tunalı
Maltepe Üniversitesi Lisansüstü Eğitim Enstitüsü, 2021

İnsanlar günlük hayatta yaşadıkları deneyimlerini, görüşlerini, kişisel duygularını yazılı hale getirme eğilimindedirler. Aynı zamanda insanlar herhangi bir konuda karar vermek ya da bilgi sahibi olmak için başkalarının düşüncelerine ihtiyaç duyarlar. Günümüzdeki bu bilgi alışverişi internet üzerindeki sosyal platformlarda gerçekleşmektedir. Bu platform birçok konu hakkında görüş bildirmektedir. Bu görüşlerin tek tek okunup analiz edilmesi veri çokluğundan dolayı imkansız hale gelmiştir. Buradaki bilgilerin doğru analiz edilmesi büyük önem taşımaktadır. Bu durumda devreye duygu analizi girer. Duygu analizi görüş bildiren metinleri sınıflandırmamıza ve bu görüşleri anlaşılır bir biçimde sunma işlemini ifade eder.

Bu çalışmada önceden etiketlenmiş eğitim verisi üzerinde farklı makine öğrenmesi yöntemleri ile duygu analizi yapılmış ve başarı oranları karşılaştırılmıştır. Başarı oranlarını arttırmak için çeşitli önişleme yöntemleri denenmiştir ve sonuçlar yorumlanmıştır.

Anahtar Sözcükler: Duygu analizi, makine öğrenmesi, veri önişleme.

ABSTRACT

SENTIMENT ANALYASIS IN TURKISH TEXTS USING MACHINE LEARNING TECHNIQUES

Batıbay Düven
Master's Thesis
Department of Computer Engineering
Computer Engineering Programme
Thesis Advisor: Assist. Prof. Dr. Üyesi Volkan Tunalı
Maltepe University Graduate School, 2021

People tend to write down their daily experiences, opinions, and personal feelings. At the same time, people need the opinions of others to make decisions or gain knowledge on any subject. Today, this information exchange takes place on social platforms on the internet. This platform provides opinions on many issues. It has become impossible to read and analyze these views one by one due to the abundance of data. It is very important to analyze the information here correctly. In this case, sentiment analysis comes into play. Sentiment analysis refers to the process of classifying texts expressing opinions and presenting these opinions in an understandable way.

In this study, emotion analysis was performed with different machine learning methods on previously tagged training data and success rates were compared. Various pre-treatment methods have been tried to increase the success rates and the results have been interpreted.

Keywords: Sentiment analysis, machine learning, data preprocessing.

İÇİNDEKİLER

JÜRİ VE ENSTİTÜ ONAYI ii
ETİK İLKE VE KURALLARA UYUM BEYANI iii
TEŞEKKÜR iv
ÖZ v
ABSTRACT vi
İÇİNDEKİLER vii
TABLOLAR LİSTESİ ix
ŞEKİLLER LİSTESİ x
KISALTMALAR LİSTESİ xi
ÖZGEÇMİŞ xii
BÖLÜM 1. GİRİŞ 1
BÖLÜM 2. İLGİLİ ÇALIŞMALAR 3
BÖLÜM 3.YÖNTEM 5
3.1. Makine Öğrenmesi 5
3.2. Duygu Analizi 5
3.3. Öznitelik Çıkarımı 7
3.4. Çalışmada Kullanılan Veri 9
3.5. Veri Ön İşleme 9
3.5.1. Dizgeciklere ayırma 10
3.5.2. Durak sözcük filtreleme 10
3.5.3. Kök alma 11
3.5.4. Terim Ağırlıklandırma 12
3.6. Kullanılan Araçlar 14
3.6.1. Preto 14
3.6.2. Weka 18
3.6.3. ARFF Converter 20
3.7. Kullanılan Algoritmalar 22
3.7.1. Naive Bayes 22
3.7.2. Naive Bayes Multinomial 24

3.7.3. SMO 25
3.7.4. IBK 26
3.7.5. J48 27
3.7.6. Random Forest 30
3.8. Değerlendirme Ölçütleri 31
3.8.1. Doğruluk 31
3.8.2. Kesinlik 32
3.8.3. Hassasiyet 32
3.8.4. F1 Skor 32
3.8.5. ROC Eğrisi 32
BÖLÜM 4. DENEYLER 34
4.1. Deney 1 34
4.2. Deney 2 34
4.3. Deney 3 35
4.4. Deney 4 35
4.5. Deney 5 36
4.6. Deney 6 36
4.7. Deney 7 37
4.8. Deney 8 37
BÖLÜM 5. DENEYLER VE TARTIŞMA 39
BÖLÜM 6. SONUÇ 41
KAYNAKÇA 42

TABLOLAR LİSTESİ

Tablo 3.1. n-Gram Örneği 8
Tablo 3.2. Doküman Terim Matrisi için Terim-İndis Değerleri 9
Tablo 3.3. Doküman Terim Matris Örneği 9
Tablo 3.4. Basit Bir Cümlenin Dizgelere Ayrılması 10
Tablo 3.5. Promosyonlara Göre Cinsiyet Ayrımı Örnek Veri Kümesi 23
Tablo 3.6. Naive Bayes Algoritmasıyla Sınıflandırılacak Örnek Veri 24
Tablo 3.7. IBF Algoritması İçin Örnek Veri Kümesi 27
Tablo 3.8. IBF Algoritmasıyla Sınıflandırılacak Örnek Veri 27
Tablo 3.9. Örnek Veri İçin Öklid Uzaklığının Hesaplanması 27
Tablo 3.10. J48 Algoritması İçin Örnek Veri Kümesi 29
Tablo 3.11. Hava Durumuna Göre Tenis Oynama Durumu 29
Tablo 4.1. Deney 1 Sonuçları 34
Tablo 4.2. Deney 2 Sonuçları 35
Tablo 4.3. Deney 3 Sonuçları 35
Tablo 4.4. Deney 4 Sonuçları 36
Tablo 4.5. Deney 5 Sonuçları 36
Tablo 4.6. Deney 6 Sonuçları 37
Tablo 4.7. Deney 7 Sonuçları 37
Tablo 4.8. Deney 8 Sonuçları 38
Tablo 5.1. Deneylerde Elde Edilen Öznitelik Sayıları 39

ŞEKİLLER LİSTESİ

Şekil 3.1. Duygu Analizi Basamakları 6

Şekil 3.2. Duygu Analizi Sınıflandırma Teknikleri [16]. 7

Şekil 3.3. Preto Ekran Görüntüsü 15

Şekil 3.4. Preto İşlem Adımları [22]. 16

Şekil 3.5. Preto'dan Çıkan Docbyterm.txt dosyası 18

Şekil 3.6. ARFF Veri Dosyası Örneği 20

Şekil 3.7. ARFF Converter Ekran Görüntüsü 21

Şekil 3.8. Weka Seyrek Matris Formatındaki ARFF Dosyası 22

Şekil 3.9. DVM Destek Vektörleri ve Hiper Düzlemi [28]. 25

Şekil 3.10. Veri Kümesinin J48 Algotimasına Göre Oluşan Ağaç Yapısı 30

Şekil 3.11. ROC Eğrileri ve Altında Kalan Alanlar 33

KISALTMALAR

ARRF : Attribute-Relation File Format

AUC : Area Under Curve

DVM : Destek Vektör Makineleri

K-NN : K-Nearest Neighbor

ROC : Receiver Operating Characteristics

SMO : Sequential Minimal Optimization

TF-IDF : Term Frequency – Inverse Document Frequency

Weka : Waikato Environment for Knowledge Analysis

ÖZGEÇMİŞ

Batıbay DÜVEN

Bilgisayar Mühendisliği Anabilim Dalı

Eğitim

Derece	*Yıl*	*Üniversite, Enstitü, Anabilim/Anasanat Dalı*
Y.Ls.	2021	Maltepe Üniversitesi, Lisansüstü Eğitim Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı
Ls.	2015	Kocaeli Üniversitesi, Mühendislik Fakültesi Makine Mühendisliği Anabilim Dalı
Lise	2010	Kadıköy Anadolu Lisesi

İş/İstihdam

Yıl	*Görev*
2015- 2018	Makine Mühendisi. ISIMAS A.Ş
2018 -…	Makine Mühendisi. GEPA GROUP

BÖLÜM 1. GİRİŞ

Günümüzde dünya nüfusu hızlı bir şekilde artmaktadır. En güncel tahminlere göre dünya nüfusu 7,78 milyarı geçmiştir ve her sene %1 ile %1,3 arasında nüfus artışı gerçekleşmektedir [1]. Nüfus artışı ve teknolojik gelişmelerin artmasıyla birlikte bilgisayar ve internet kullanımı hızlı bir şekilde artmaktadır. Bilgisayarın yaygın kullanılmasıyla beraber insan hayatında birçok değişiklik ve yenilik oluşmuştur. İnsanların alışveriş yapma alışkanlıklarından, bilgi edinme yöntemlerine kadar birçok konuda değişiklikler meydana gelmiştir.

Artan nüfus ve bilgisayar kullanımıyla internet ortamında çok büyük bir bilgi birikimi oluşmaktadır ve günden güne bu bilgi birikimi artmaktadır. Bu bilgi yığını arasından işe yarar bilgiyi edinmenin önemi oldukça büyüktür. Veri madenciliği, büyük ölçekli veriler arasından faydalı bilgiye ulaşmayı amaçlar. Veri madenciliğinin alt dallarından biri olan Duygu Analizi ise görüş bildiren yazılı metinlerin amaca uygun bir şekilde analiz edilip, kullanılmasına olanak sunar [2].

Duygu Analizi hakkında yayınlanmış birçok makale bulunmaktadır. Özellikle son yıllarda bu sayıda dikkat çekici bir artış söz konusudur. Girişimci yeni firmalar ve köklü firmalar, problemlerine yeni çözümler üretmek ve hizmet kalitesini arttırmak için yatırımlar yapmakta ve Duygu Analizi çalışmalarına başvurmaktadırlar.

Duygu Analizi birçok çalışmada bir ürün ve hizmetin yorumlarını analiz edebilmek ve açıklamak için basit ifadeler kullanır. Buna rağmen, diller arası farklılıklar, kelimelerin cümle içlerinde farklı anlamlarda kullanılması ve kültürel farklılıkların yazı diline yansıması Duygu Analizi çalışmaları için bir problem haline gelmektedir. Bir cümle içindeki kelimelerden anlam bütünlüğü çıkarmak bir dokümanı yorumlamaktan daha zor olabilmektedir. İnsanların duygu ve düşüncelerini barındıran yazıları analiz etmek ve doğru kararı verebilmek bilgisayarlar için her zaman kolay değildir.

Veri çokluğu, verilerin tek tek analiz edilmesini mümkün kılmamaktadır. Amaç birçok veriyi tek tek incelemeden otomatikleştirilmiş bir sistemle, metinlerdeki duyguyu

saptamaktadır. Bu esnada devreye makine öğrenmesi girer. Makine öğrenmesinin günümüzde pek çok farklı kullanım alanı bulunmaktadır. Otomotiv, tıp, eğlence, pazarlama, müşteri memnuniyeti, bankacılık makine öğrenmesinin temel kullanım alanlarıdır. Bu da makine öğrenmesinin günümüz dünyasında var olan bilgiyi iyi değerlendirebilme konusunda ne kadar etkin bir öneme sahip olduğunu göstermektedir. Bu tez çalışmasında denetimli makine öğrenmesi yöntemleri kullanılarak Türkçe metinler üzerinde Duygu Analizi yapılmıştır.

Bu tez çalışmasının ikinci bölümünde literatürde duygu analizi ile ilgili daha önce yapılmış çalışmalara değinilmiştir. Üçüncü bölümünde, duygu analizinde kullanılan temel kavramlar, çalışmada kullanılan veri, önişleme teknikleri, kullanılan araçlar, kullanılan makine öğrenmesi algoritmaları ve değerlendirme ölçütleri hakkında bilgiler verilmektedir. Dördüncü bölümünde yapılan deney ve sonuçlarına, beşinci bölümde ise deney sonuçlarının değerlendirilmesine ve sonuç kısmına yer verilmiştir.

BÖLÜM 2. İLGİLİ ÇALIŞMALAR

Kaynar, Yıldız, Görmez ve Albayrak IMDB'de bulunan film yorumlarından elde ettikleri veri seti üzerinde çeşitli makine öğrenmesi yöntemleriyle duygu analizi çalışması yapmışlardır. Veri setinde toplam 2.000 yorum bulunmaktadır. 1.000 tanesi negatif 1.000 tanesi pozitif olarak etiketlenmiştir. Veri setini analiz etmek için MATLAB'de Naive Bayes, Merkez Tabanlı Sınıflayıcı, Çok Katmanlı Yapay Sinir Ağları ve Destek Vektör Makineleri sınıflandırma algoritmalarından yararlanılmıştır. Sınıflandırma algoritmalarında verinin %75'i eğitim için ayrılmış, kalan %25'lik bölümü ise test amacıyla kullanılmıştır. Algoritma başarıları incelendiğinde en yüksek değerler Yapay sinir ağları ve Destek vektör makinelerinde elde edilmiştir. Eğitim veri setinde, yapay sinir ağları %89,73, destek vektör makineleri ise %84,07 oranında doğru sınıflama yapmışlardır. Test veri setinde ise her iki sınıflayıcı %75 civarında doğru sınıflama oranıyla hemen hemen aynı performansı göstermişlerdir [3].

Çoban ve arkadaşları Twitter üzerinden elde ettikleri Türkçe veri seti üzerinde Duygu Analizi çalışması yapmışlardır. Twitter mesajlarının içerdiği his simgelerine göre, mesajları negatif ve pozitif olmak üzere iki ayrı gruba ayırmışlardır. Ön işleme adımlarından sonra, kelime torbası ve *n*-gram modelleri kullanılarak öznitelikler çıkartılmıştır. Özniteliklerin ağırlıklandırılmasında TF, Boolean ve TD-IDF olmak üzere üç farklı yöntem kullanılmıştır ve en iyi sonucu Boolean ağırlıklandırma yöntemi vermiştir. Deneysel sonuçlar SVM, Naive Bayes, Naive Bayes Multinomial ve KNN algoritmalarıyla elde edilmiştir. En başarılı sonucu kelime torbası modelinde %62,48'lik doğru sınıflandırma yüzdesi ile Naive Bayes Multinomial, *n*-gram modelinde ise %66,06'lık doğru sınıflandırma yüzdesi ile Naive Bayes Multinomial vermiştir [4].

Sevindi, Türkçe film yorumları üzerinde yaptığı çalışmada hem makine öğrenmesi hem de sözlük tabanlı yaklaşım yöntemleri ile duygu sınıflaması yapmıştır. Makine öğrenmesi yönteminde alınan en iyi sonuç 0,8258 F-skor değeri ile SVM sınıflandırıcısı ile elde ediliştir. Bu sonuca *n*-gram uzunluğu bir iken ulaşılmış olup *n*-gram uzunluğu arttıkça başarı oranı düşmüştür. Sözlük tabanlı yaklaşım yönteminde alınan en iyi sonuç 0,5969 F-skor değeri olmuştur. Bu değere skor sınırı kullanılmadan

ulaşılmıştır. Skor sınır değeri arttıkça F-skor değerinin gittikçe düştüğü tespit edilmştir [5].

Kumaş, 16.000 negatif, 16.000 pozitif olmak üzere toplam 32.000 adet yorumun bulunduğu, Twitter'dan elde etmiş olduğu veri seti üzerinde makine öğrenmesi yöntemini kullanarak duygu analizi çalışması yapmıştır. TF-IDF terim ağırlıklandırma yöntemini kullanan Kumaş, sınıflandırma algoritmalarında veri setinin %80'ini eğitim %20'sini test için kullanmıştır. Elde edilen sonuçlara göre, %73'lük F-skor değeri ile en başarılı algoritma destek vektör makineleri olmuştur [6].

Köksal, Erdem, Türkeli ve Öztürk, yaptıkları çalışmada, Twitterda Bitcoin ile ilgili yazılan tweetlerden yola çıkarak Bitcoin değer tahminlemesi yapmışlardır. 'Bitcoin' anahtar kelimesini içeren günlük pozitif tweet oranıyla Bitcoin günlük açılış değeri beraber kullanılarak, Bitcoin kapanış değeri tahmin edilmiştir. Doğrusal regresyonun uygulandığı yöntemde r^2 değeri %88,97 iken rastgele orman regresyonun uygulandığı yöntemde bu değer %94,16 çıkmıştır [7].

Yoldaş'ın SentiTurkNet sözlüğünü kullanarak yaptığı sözlük tabanlı duygu analizi çalışmasında pozitif içerikli cümlelerin negatif içerikli cümlelere nazaran daha iyi sınıflandırıldığı ve yoruma açık cümlelerin sınıflandırılmasında sözlük tabanlı yöntemlerin eksiklikleri bulunduğu tespit edilmiştir [8].

Aytekin ve Bayram yaptıkları çalışmada eğitim verisi için 'Karma Model' uygulaması geliştirmişlerdir. Bu sayede her mecra için farklı eğitim verisine gerek kalmadan duygu analizi yapılması hedeflenmiştir. Bağlamdan bağımsız daha genel bir model oluşturulabilmek için, her bir mecradan rasgele belli bir sayıda alınan verilerle karma bir veri kümesi oluşturulmuştur. Karma model ile, mecraya özgü eğitim verilerinin kullanıldığı deneylerde başarı oranlarının birbirine yakın bulunduğu tespit edilmiştir [9].

BÖLÜM 3.YÖNTEM

3.1. Makine Öğrenmesi

Makine öğrenmesi, özel olarak programlanmadan sonuçları tahmin etmede yazılım uygulamalarının doğruluğunu artıran bir yapay zeka türüdür. Sistemlerin verilerden öğrenebileceği, kalıpları belirleyebileceği ve minimum insan müdahalesi ile karar verebileceği fikrine dayanır [10]. Makine öğrenmesi denetimli ve denetimsiz olmak üzere iki ana gruba ayrılır. Pratik makine öğrenmesinin çoğu, denetimli öğrenmeyi kullanır. Denetimli öğrenme, giriş değişkenlerine (X) ve bir çıkış değişkenine (Y) sahip olunan ve girişten çıkışa eşleme işlevini öğrenmek için bir algoritma kullanılan makine öğrenmesi yöntemidir. Amaç, eşleme işlevine o kadar iyi yaklaşmaktır ki, yeni girdi verileri (X) olduğunda, bu veriler için çıktı değişkenlerini (Y) tahmin edebilmek amaçlanır. Algoritma kabul edilebilir bir performans seviyesine ulaştığında öğrenme durur. Denetimli öğrenme sınıflandırma ve regresyon problemleri olarak gruplandırabilir. Eğer çıktı değişkeni kategorik ise sınıflandırma (classification), numerik ise regresyon (regression) denir [11].

Denetimsiz öğrenme, yalnızca giriş verilerinin (X) ve karşılık gelen çıktı değişkenlerinin bulunmadığı bir makine öğrenme yöntemidir. Denetimsiz öğrenmenin amacı, veriler hakkında daha fazla bilgi edinmek için verilerin temel yapısını veya dağılımını modellemektir. Denetimli öğrenmenin aksine doğru cevaplar ve eğitici yoktur. Algoritmalar, verilerdeki ilginç yapıyı keşfetmek ve sunmak için kendi tasarımlarına bırakılır. Denetimsiz öğrenme problemleri ayrıca kümeleme (clustering) ve ilişkilendirme (Association) problemleri olarak gruplandırılabilir [11].

3.2. Duygu Analizi

Duygu Analizi esas olarak, metin işleme (text processing) yöntemidir. Buna bağlı olarak verilen metnin duygusal olarak anlatmaya çalıştığı kısmı belirlemeyi hedefler. Duygusal kutupsallık (sentimental polarity) Duygu Analizinin birinci çalışma aşamasıdır. Belirlenen bir yazıyı olumlu (positive), olumsuz (negative) ve nötr olarak bölümlendirmeyi hedeflemektedir [12]. Duygu Analizi, sanal ortamda bulunan her türlü

bilgi, konu, düşünce, görüş, ürün yelpazesi, organizasyonlar vs. hakkında insanların düşüncelerini ve duygularını belirlemeyi hedeflemektedir [13]. Duygu Analizi basamakları Şekil 3.1.'deki gibi ifade edilebilir.



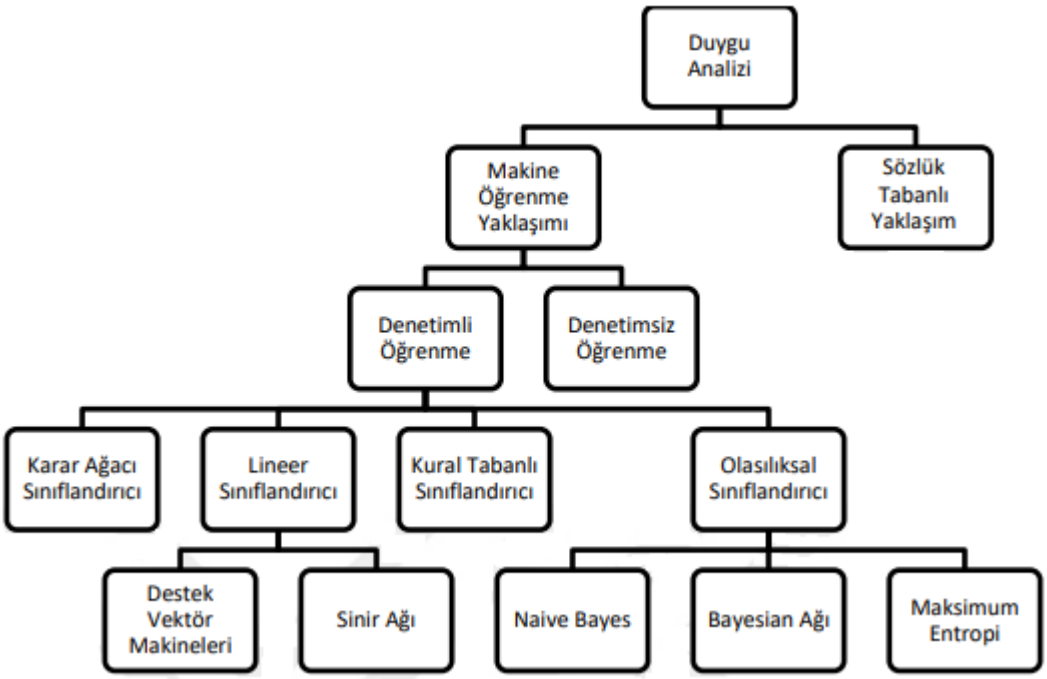
Şekil 3.1. Duygu Analizi Basamakları

Duygu analizinde sözlük tabanlı yaklaşım ve makine öğrenmesi yaklaşımı olmak üzere iki farklı yöntem bulunur. Sözlük tabanlı yaklaşımda cümlelerdeki duygu anlatımlarını belirlemek için duygu terimleri sözlüğü kullanılır. Sözlükteki her terim pozitif ya da negatif olarak etiketlenmiştir. Sınıflandırılacak metinlerdeki sözcükler, duygu terimleri sözlüğünde aranır ve sınıflandırma işlemi bu aramadan elde edilen sonuçlara göre yapılır. Buna göre metindeki sözcükler pozitif ve negatif olarak kodlanır. Skorlama yapılarak cümlenin hangi kutupta yer aldığına karar verilir [14]. Makine öğrenmesine dayalı metotlarda ise daha önce etiketlenmiş eğitim verisi ile sistem eğitilir böylece eğitilmiş sistem ile duygu sınıflandırması yapılır.

Makine öğrenmesine dayalı yöntemlerle sözlük tabanlı yöntemlerin birbirine karşı avantaj ve dezavantajları mevcuttur. Makine öğrenmesi yönteminin en büyük dezavantajı çalışma yapılacak konuyla ilgili etiketlenmiş eğitim verisine ihtiyaç duyulmasıdır. Ayrıca bir çalışmada kullanılan eğitim verisi başka bir çalışmada kullanılamaz yani her çalışma alanına özgü eğitim verisine ihtiyaç vardır. Örneğin sinema konulu bir eğitim verisi bilgisayar konulu bir çalışmada kullanılırsa işe yaramaz. Her konuyla ilgili eğitim verisinin bulunması oldukça zordur [13].

Sözlük tabanlı yöntemler, etiketlenmiş eğitim verisine ihtiyaç duymazlar. Bu yönüyle makine öğrenmesi yöntemine göre üstünlük sağlamaktadır. Ancak sözlük tabanlı

yöntemde duygu terimleri sözlüğüne ve doğal dil işleme tekniklerine ihtiyaç duyulur. Duygu terimleri sözlüğü ve doğal dil işleme teknikleri dilden dile farklılık göstermektedir. Her dile özgü doğal dil işleme teknikleri ve duygu terimleri sözlüğü mevcuttur. Bu konuda en avantajlı dil İngilizcedir. İngilizcede birçok hazır duygu terimleri sözlükleri ve doğal dil işleme araçları bulunmaktadır. Diğer dildeki araçlar İngilizcedeki kadar etkin değildir [15].



Şekil 3.2. Duygu Analizi Sınıflandırma Teknikleri [16].

3.3. Öznitelik Çıkarımı

Var olan verilerin işlenmesi ve bilgisayar tarafından anlaşılır hale gelmesi için veriyi bir öznitelik vektörü şeklinde ifade etmek gerekir. Bir metni iyi bir şekilde temsil edebilecek öznitelik vektörüne dönüştürmek, duygu analizinin temel adımlarından biridir. Genellikle *n*-gramları içeren öznitelikler üzerinden eğitilen sınıflandırıcı, yeni gelen örnekleri bölümlendirir. *n*-gram, *n* uzunluğundaki kelime dizisine verilen addır. Kelimeler, basitçe bir uzunluktaki *n*-gram öznitelikler olarak düşünülebilir. İfadeler ise bir veya daha fazla kelime gruplarının oluşturduğu kelime gruplarıdır (2-gram, 3-gram, 4-gram, 5-gram) [5]. Tablo 3.1.'de bir cümlenin *n*-gramlara ayrılmış hali örnek olarak verilmiştir.

Tablo 3.1. *n*-Gram Örneği

Cümle	Bugün Okullar tatil olduğu için mutluyum
1-gram (Unigrams)	{Bugün}, {okullar}, {tatil}, {olduğu}, {için}, {mutluyum}
2-gram (Bigrams)	{Bugün okullar}, {okullar tatil}, {tatil olduğu}, {olduğu için}, {için mutluyum}
3-gram (Trigrams)	{Bugün okullar tatil}, {okullar tatil olduğu}, {tatil olduğu için}, {olduğu için mutluyum}
4-gram (Fourgrams)	{Bugün okullar tatil olduğu}, {okullar tatil olduğu için}, {tatil olduğu için mutluyum}
5-gram (Fivegrams)	{Bugün okullar tatil olduğu için}, {okullar tatil olduğu için mutluyum}

Veri setinde kullanılan kelimeler çıkartılarak bir kelime torbası oluşturulur. Bu kelimeler mevcut metinde geçiyorsa ‘1’ geçmiyorsa ‘0’ olarak kodlanır. Bu sayede doküman terim matrisi elde edilir. Ayrıca yazı içerisindeki sözcüklerin sırası da vektör uzay modelinde önemini yitirmektedir. Yani bir sözcüğün nerede geçtiğinin önemi yoktur.

Doküman terim matrisine aşağıda örnek olarak üç cümle verilmiştir [12].

- D_1 = Adam likes to play volleyball.
- D_2 = Rose likes volleyball too.
- D_3 = Adam also likes basketball.

Tablo 3.2. Doküman Terim Matrisi için Terim-İndis Değerleri

Terim	İndis
Adam	1
likes	2
to	3
play	4
volleyball	5
Rose	6
too	7
also	8
basketball	9

Tablo 3.2.'de verilen terim dizinlerin birer vektöre dönüşmüş hali Tablo 3.3'te sunulmuştur.

Tablo 3.3. Doküman Terim Matris Örneği

	Adam	likes	to	play	volleyball	Julia	too	also	basketball
D_1	1	1	1	1	1	0	0	0	0
D_2	0	1	0	0	1	1	1	0	0
D_3	1	1	0	0	0	0	0	1	1

Tablo 3.3.'teki örnekte gösterildiği gibi artık her dokümanı ikilik tabanda bir vektör ile anlatmak mümkündür. Örneğin D_1 ''111110000'' şeklinde ifade edilebilir.

3.4. Çalışmada Kullanılan Veri

Çalışmadaki kullanılan veri, bir internet sayfasında otomobiller hakkında yapılan yorumlardan oluşmaktadır. Yorumlar internet sayfasından tek tek elle toplanmıştır. Her bir yorum farklı bir TXT dosyasına kaydedilmiştir. Yorumlar tek tek incelenip pozitif ve negatif olarak ayrılmıştır. 750 adet pozitif yorum, 500 adet negatif yorum olmak üzere toplam 1.250 yorum çekilmiştir.

3.5. Veri Ön İşleme

Veri madenciliği uygulamalarında veri elde edilmesinden sonra, veri önişleme adımı uygulanmaktadır. Önişleme adımının temel amacı, örnek kategorileri metinsel verilerden ayırt edebilen ve bunları uygun biçime dönüştürebilen önemli öznitelikleri

ortaya çıkarmaktır [17]. Buradaki formatla kastedilen, verilerin sayısallaştırılmasıdır. Bu sayede veri, makine öğrenmesi algoritmaları tarafından işlenebilir ve yorumlanabilir. Veri önişleme, veriyi daha sade ve anlaşılır hale getirilmesini ve veri herhangi bir algoritmaya girdi olarak verildiğinde başarı oranın artmasını sağlamaktadır. Metin belgelerini önişlemeden geçirmek için metin madenciliğinde sıklıkla kullanılan metotlardan kimisi dizgeciklere ayırma (tokenization), durak sözcük filtreleme (stopword filtering), kök bulma (stemming) ve terim ağırlıklandırmadır (term weighting) [18].

3.5.1. Dizgeciklere ayırma

Metin sınıflandırmada; dizgelere ayırma, metnin dizge (token) adı verilen kelimelere, öbeklere veya anlamlı parçalara ayrılması işlemidir. Bir cümleyi daha küçük anlamlı birimlere ayırmak olarak tanımlanabilir. Token'lar, anlamlı küçük birimlerdir. Semboller, kelimeler, deyimler token'lere örnek verilebilir. Kullanılan tokenizer'a göre ayrıştırma değişir. En çok kullanılan ayırma işlemi ise kelimelere ayırma işlemidir. Bütün bir yazıyı oluşturan her bir kelime tek tek ayrılır [17]. Bu çalışmada dokümanlar kelime bazında dizgeciklere ayrılmıştır. Tablo 3.4.'te basit bir cümlenin dizgelere ayrılması gösterilmektedir.

Tablo 3.4. Basit Bir Cümlenin Dizgelere Ayrılması

Cümle	**Dizgeler (virgüllerle ayrılmış)**
Bugün hava çok güzel	Bugün, hava, çok, güzel

3.5.2. Durak sözcük filtreleme

Durak sözcük kavramı, belirli bir sınıfa bağımlı olmayan, metinlerde sıklıkla karşılaşılabilecek terimleri veya kelimeleri (bağlaçlar, edatlar) ifade etmektedir. Bu kelimeleri kaldırmadaki amaç daha temiz bir veri seti elde etmek ve öznitelikleri doğru çıkarabilmektir. Türkçe durak kelimelerine örnek olarak ise "acaba, ama, ancak, artık, bazen, bazı, bazıları, bazısı, belki, gene, gibi, hangi, hangisi, elbette, için, ile" verilebilir. [19]. Bu çalışmada Türkçe durak kelimelerin listelendiği bir dosya durak sözcük filtreleme için kullanılmıştır. Bunun yanı sıra otomobil marka ve modellerinin duygu

analizi değerlendirilmesinde herhangi bir etkisi bulunmadığından durak kelimelerin bulunduğu listeye otomobil marka ve modelleri de eklenmiştir.

3.5.3. Kök alma

Türkçe gibi eklemeli dillerde bir kelime birçok kelimeye dönüşebilir, ancak bu kelime kaynağında aynıdır. Kök alma yöntemi, metin madenciliği önişleme esnasında en çok kullanılan yöntemlerden biridir. Kök alma, sözcükleri basit haline çevirme işlemidir. Kök alma işlemiyle öznitelik uzay boyutunun olabildiğince azaltılması ve doğru sonuçların elde edilmesi sağlanır. Türkçe için çeşitli kök alma yöntemleri bulunmaktadır. Bunlar; Zemberek Kütüphanesi kök bulma metodu, ek çıkaran kök bulucu (affix stripping stemmer), sabit önek kök bulma (fixed prefix stemming) yöntemleridir [20]. Bu çalışmada farklı kök alma yöntemi denenmiş ve sonuçları karşılaştırılmıştır.

3.2.3.1. Zemberek kütüphanesi kök bulma

Zemberek, açık kaynak kodlu ve Java ile geliştirilmiş bir kütüphanedir. Zemberek kütüphanesini kullanarak Türkçe yazım hataları düzeltilebilir ve kelimeler eklerine ve köklerine ayrılabilir. Zemberek bir Türkçe Doğal dil işleme kütüphanesidir.

Zemberek kütüphanesi bir kelimenin Türkçe olup olmadığına göre çalışır. Bundaki temel mantık, bir kelime kök ve eklerine ayrılabiliyorsa Türkçedir, ayrılamıyorsa Türkçe değildir. Zemberek Türkiye Türkçesi kütüphanesi neredeyse 30.000 kök içeren sözlüğü de bir arada taşır, bu sözlükte her kök çeşidine ve özel durumlarına göre etiketlenmiş biçimde bulunur. Burada bahsi geçen özel durumlar sondaki yumuşama, ünlü düşmesi gibi ses olaylarıdır. Zemberek çok kolay bir yöntemle verilen bir sözcüğü morfolojik biçimde inceler. Önce iletilen sözcüğün kökü olabilecek adayları tespit eder, sonra da olabilecek ekleri elverişli sırayla bu köke eklemeye çalışır. Eğer bu işlem dizininde girişteki sözcüğün aynısını sağlamışsa, o zaman yerinde kök ve ekleri de bulmuş demektir ve sözcük Türkçedir. Şayet kök adaylarının hiçbirinden netice elde edilememişse o zaman sözcük Türkçe değildir [21].

3.2.3.2. Ek çıkaran kök bulucu

Ek çıkaran kök bulucu (affix stripping stemmer) olarak isimlendirilen ikinci kök bulma metodu, sözlük tabanlı kök bulma metotlarından farklı olarak Türkçenin kural tabanlı biçiminden faydalanmaktadır. Bu kök bulma metodu, kelimelerin biçim bilimsel (morfolojik) yapısını, kelimelere getirilen eklerin sondan başlanarak başa doğru kelimelerden çıkarılması (affix stripping) yaklaşımıyla çözümleyen bir çözümleyiciye dayalı olarak ilerletilmiştir. Bu yöntemin avantajı, bir sözlüğe ihtiyaç duymaması ve bu nedenle sözlük tabanlı kök bulma yöntemlerinden daha az işlemci zamanı kullanmasıdır. [18].

3.2.3.3. Sabit Önek

Kullanılan son kök bulma metodu olan sabit önek (fixed prefix stemming) yöntemi sözde bir kök bulma metodudur. Temel olarak, kelimelerin ilk n karakterlerinin kelimenin kökü olarak kullanıldığı kolay ve hızlı bir yöntemdir. *n* karakter ve daha kısa kelimeler ise aynı şekilde kullanılırlar. Bu yöntem ayrıyeten sözcük kesme (word truncation) metodu olarak da bilinmektedir [18].

3.5.4. Terim Ağırlıklandırma

Terim ağırlıklandırma bir dokümanda geçen terimlerin (sözcüklerin) etki değerinin hesaplanması şeklinde açıklanabilir. Doküman terim matrisi düşünüldüğünde her kelime aynı etki değerine sahiptir. Yalnız bir kelime seyrek ya da çok sık kullanılıyorsa, bu durum kelimenin sonuca olan etkisini değiştirmektedir. Terim ağırlıklandırma, terimlerin (sözcüklerin) etki değerinin saptanıp kullanılan algoritmalarda başarı oranının artmasını sağlamaktadır. Bunun için kullanılan en yaygın yöntem TF-IDF hesabıdır.

İngilizcedeki Term Frequency – Inverse Document Frequency (Terim frekansı – ters metin frekansı) olarak geçen sözcüklerin baş harflerinden oluşan terim basitçe bir metinde geçen terimlerin çıkarılması ve bu terimlerin geçtiği sıklığa göre farklı hesapların yapılması üzerine kuruludur. Kısacası TF-IDF hesabı sırasında iki önemli sayı bulunmaktadır. Bunlardan ilki, incelenmekte olan belgedeki terimin sayısı, diğeri ise bu

terimi içeren toplam belge sayısıdır [19]. Formülü denklem 3.1., 3.2. ve 3.3.'teki gibi ifade edilmiştir.

$$TF = \frac{Terimin\ dok\ddot{u}manda\ geçme\ sayısı}{Dok\ddot{u}mandaki\ terim\ sayısı} \tag{3.1}$$

$$IDF = \log \frac{Toplam\ dok\ddot{u}man\ sayısı}{Terimin\ geçtiği\ dok\ddot{u}man\ sayısı} \tag{3.2}$$

$$TF\text{-}IDF = TF \times IDF \tag{3.3}$$

Bu hesaba göre terimin geçtiği doküman sayısı ile TF-IDF skoru ters orantılıdır. Bu sayede birçok dokümanda geçen, dolayısıyla ayırt edici özelliği olmayan terimlerin skorları, ayırt edici terimlere göre daha düşük olacaktır [5].

TF-IDF hesabı için aşağıda üç adet örnek cümle verilmiştir.

- Mandalina sarı ya da turuncu renklidir çoğunlukla sarı olur.
- Muz yeşil ve uzundur, sarı renkte de olabilir.
- Karpuzun içi kırmızı renklidir.

'Sarı' kelimesi için ilk sıradaki cümle içinde hesaplama yapıldığında, $TF = \left(\frac{2}{8}\right) = 0{,}25$ ve $IDF = \log\frac{3}{2} = 0{,}18$ olarak hesaplanır. TF-IDF $= 0{,}25 \times 0{,}18 = 0{,}044$ bulunur. Burada cümle sayı arttırılıp dört yapılırsa IDF değeri 0,3 olur. TF-IDF değeri ise 0,075 olacaktır. Yani 'sarı' teriminin tüm metinler içinde seyrekliği yükseldi ve daha ayırt edilen bir özellik kazandı.

TF-IDF hesabında bir terimin toplam kullanılma sayısı kadar, kaç farklı dokümanda kullanıldığı da önem kazanır. Örneğin sadece bir dokümanda 100 kere geçen bir terimle 10 farklı dokümanda 10'ar kere geçen terimin ikisi de aslında toplamda 100 kere geçmiştir ancak TF-IDF ikincisine yani daha çok dokümanda olana önem verir [19].

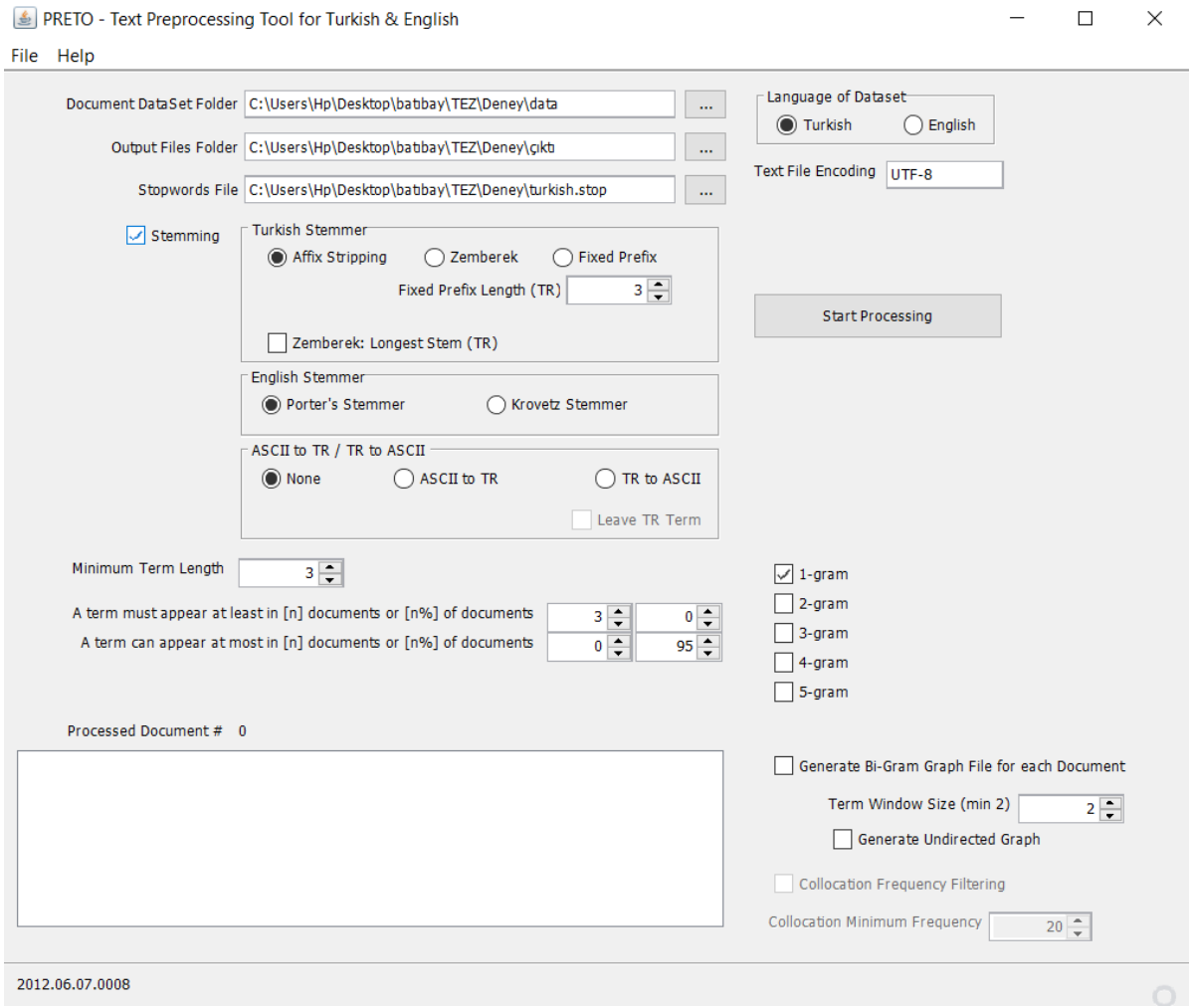
3.6. Kullanılan Araçlar.

Bu kısımda tez çalışmasında kullanılan araçlardan söz edilmiştir.

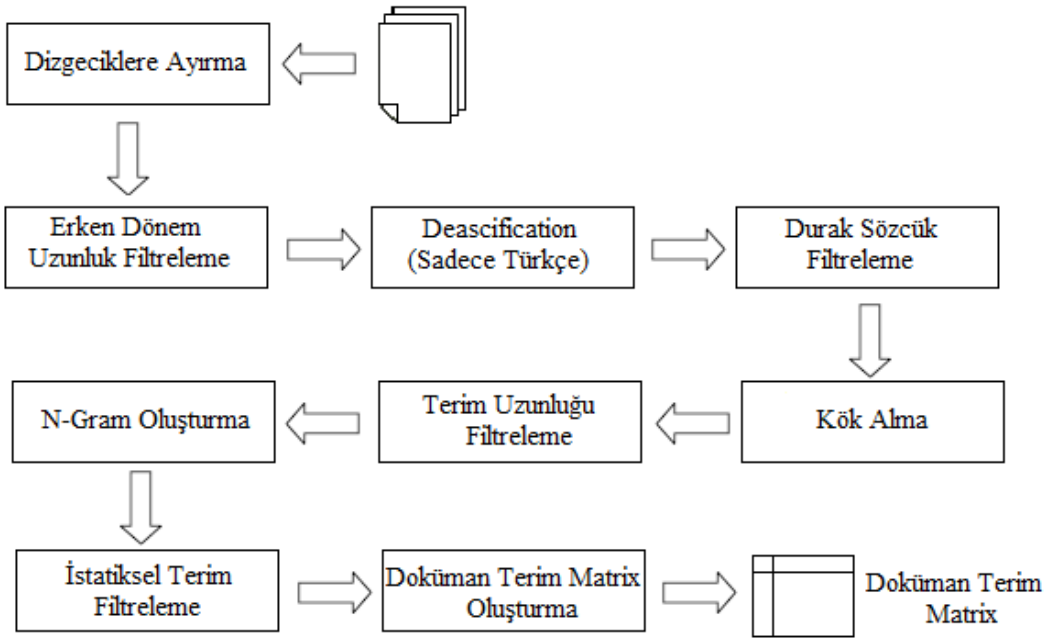
- Preto: Veri önişleme ve doküman terim matrisin elde edilme işlemleri için kullanılmıştır.
- Weka: Veriyi çeşitli algoritmalarla sınıflandırmak için kullanılmıştır.
- ARFF Converter: Preto Çıktısını ARFF formatına çevirmek için kullanılmıştır.

3.6.1. Preto

Preto, Türkçe metinlerde Metin Madenciliğinin önişleme operasyonları için ilerletilmiş açık kaynak kodlu ve temelinden ayrı çalışabilen bir araçtır. Preto geniş bir önişleme yelpazesi sunar. Bunlar; kök alma (stemming), durak sözcük filtreleme (stopword filtering), istatistiksel terim filtreleme ve *n*-gram oluşturma gibi [22].



Şekil 3.3. Preto Ekran Görüntüsü



Şekil 3.4. Preto İşlem Adımları [22].

Şekil 3.4.'te Preto'nun işlem adımları verilmiştir. Buna göre Preto dokümanlarda yazılanları kelime bazında dizgeciklere (token) ayırarak veri önişlemeye başlar. Deascification özelliği ise Türkçe olup ama Türkçe yazılmayan kelimeleri Türkçeye çevirmeye yarar. Örneğin; 'isildama' olarak yazılmış aslında 'ışıldama' olan kelimenin doğru yazılmasını sağlar. Preto bu işlemi Zemberek Kütüphanesi altyapısı kullanarak gerçekleştirir.

Preto çeşitli terim filtreleme seçeneklerine sahiptir. Bunlardan ilki durak kelimeleri filtrelemedir. Diğer filtreleme seçeneği ise kelimelerin uzunluklarına göredir. Örneğin kelime uzunluğu $n = 4$ karakterden az olan kelimeler filtrelenebilir. Aynı zamanda Preto istatiksel terim filtreleme özelliğine de sahiptir. Örneğin bir terim en az n dokümanda ya da %n dokümanda geçmelidir yoksa filtrelenebilir. Bir diğer istatiksel filtreleme ise bir terim en fazla n dokümanda ya da %n dokümanda geçebilir yoksa filtrelenebilir [22].

n-gram oluşturma işlemi 5-grama kadar Preto'da mümkündür. 1-gram dan 5-grama kadar çoklu n-gram seçebilmek mümkündür.

Preto Türkçe için çeşitli kök bulma yöntemleri içermektedir. Bunlar; Zemberek Kütüphanesi kök bulma metodu, ek çıkaran kök bulucu (affix stripping stemmer), sabit önek kök bulma (fixed prefix stemming) yöntemleridir.

Tüm önişleme işlemleri tamamlandıktan sonra Preto, TF, TF-IDF ve TF-IDF norm olmak üzere üç farklı terim ağırlıklandırma yöntemleri ile doküman terim matrisini oluşturur.

Preto'nun çıktı olarak verdiği dosyalar;

- **Docbyterm.mat:** Doküman Terim Matrisinin MAT uzantılı olarak bulunduğu dosyadır.
- **Docbyterm.tfidf.mat:** Doküman Terim Matrisinin TF-IDF formunda MAT uzantılı olarak bulunduğu dosyadır.
- **Docbyterm.tfidf.norm.mat:** Doküman Terim Matrisinin TF-IDF formunun normalize edilmiş halinin MAT uzantılı olarak bulunduğu dosyadır.
- **Docbyterm.txt:** Doküman Terim Matrisinin bulunduğu TXT uzantılı dosyadır. Şekil 3.5'te gösterilmiştir.
- **Docbyterm.tfidf.txt:** Doküman Terim Matrisinin TF-IDF formunda TXT uzantılı olarak bulunduğu dosyadır.
- **Docbyterm.tfidf.norm.txt:** Doküman Terim Matrisinin TF-IDF formunun normalize edilmiş halinin TXT uzantılı olarak bulunduğu dosyadır.
- **Documents_fullname.txt:** Üzerinde çalışan dokümanların o makine üzerindeki açık adreslerini listeleyen dosyadır.
- **Documents.txt:** Üzerinde çalışılan dokümanların isimlerinin listelendiği dosyadır.
- **Term_detailed.txt:** Hangi kelimeden toplamda kaç adet ve kaç tane doküman üzerinde geçtiğini tutan dosyadır.

- **Terms.txt:** İşlenen dokümanda geçen tüm sözcüklerin liste halinde bulunduğu dosyadır [20].

```
1250 6508 24698
1 1 1.0
1 2 1.0
1 3 1.0
1 4 1.0
2 5 1.0
2 6 1.0
2 7 1.0
2 8 1.0
2 9 1.0
2 10 1.0
2 11 1.0
2 12 1.0
2 13 1.0
2 14 1.0
2 15 1.0
2 16 1.0
2 17 1.0
2 18 1.0
2 19 1.0
2 20 1.0
2 21 1.0
2 22 1.0
```

Şekil 3.5. Preto'dan Çıkan Docbyterm.txt dosyası

Şekil 3.5.'e bakıldığında en üstteki üç rakam sırasıyla; incelenen doküman sayısını, öznitelik sayısını ve devamında okunması gereken veri sayısını ifade eder. Altındaki üç rakam ise sırasıyla; hangi dokümanda, hangi kelimenin, kaç defa geçtiğini ifade eder.

3.6.2. Weka

Weka, makine öğrenmesi, Waikato Üniversitesinde geliştirilmiş ve "Waikato Environment for Knowledge Analysis" kelimelerinin baş harflerinin birleştirilmesi ile meydana gelmiş yazılımın adıdır. Zamanımızda artık çok sık kullanılmakta olan büyük bir kısmı makine öğrenmesi algoritmalarını ve yöntemlerini kapsamaktadır. Java Programlama Dili sayesinde ilerletilmiştir ayrıca açık kaynak kodludur [23].

Weka, bütünüle modüler bir yapıya sahip olduğundan, kapsadığı niteliklerle veri grupları üzerinde görselleştirme, iş zekası, veri analizi kullanımları ve veri madenciliği gibi bir takım işlemler yapabilmektedir. Esas olarak üç Veri Madenciliği işlemi Weka sayesinde yapılabilmektedir.

- Classification (Sınıflandırma): Sınıflandırmak veya tahmin etmek için öğrenme yöntemini eğitir ve değerlendirir.
- Clustering (Kümeleme): Veri kümeleri için kümeleri öğrenmeyi sağlar.
- Association (Birliktelik Analizi): Veri seti için birliktelik kurallarını öğrenir ve bunları değerlendirir.

  Bu işlemlere ek olarak, veri gruplarında ön ve son işlemler yapılabilir.

- Data Pre-Processing (Veri Önişleme): Veri setinin seçilmesi ve bunu farklı yöntemlerle düzenlenmesini sağlar.
- Visualization (Görselleştirme): Veri kümesinin iki boyutlu farklı grafiklerinin görülmesini ve bunlar arasındaki etkileşimi belirlenmesini sağlamaktadır [24].

3.2.2.1. ARFF Dosya Formatı

ARFF, Attribute Relationship File Format sözcüklerinin baş harflerinin bir araya gelmesidir. ARFF dosya yapısı, Weka için özel olarak ayrı düzenlenmiştir. Dosya, metin yapısında mevcuttur ve Weka yazılımında; CSV dosyalarını ARFF formatına çevirmeye yarayan araçlar bulunmaktadır.

```
@RELATION iris
@ATTRIBUTE sepallength NUMERIC
@ATTRIBUTE sepalwidth  NUMERIC
@ATTRIBUTE petallength NUMERIC
@ATTRIBUTE petalwidth  NUMERIC
@ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}

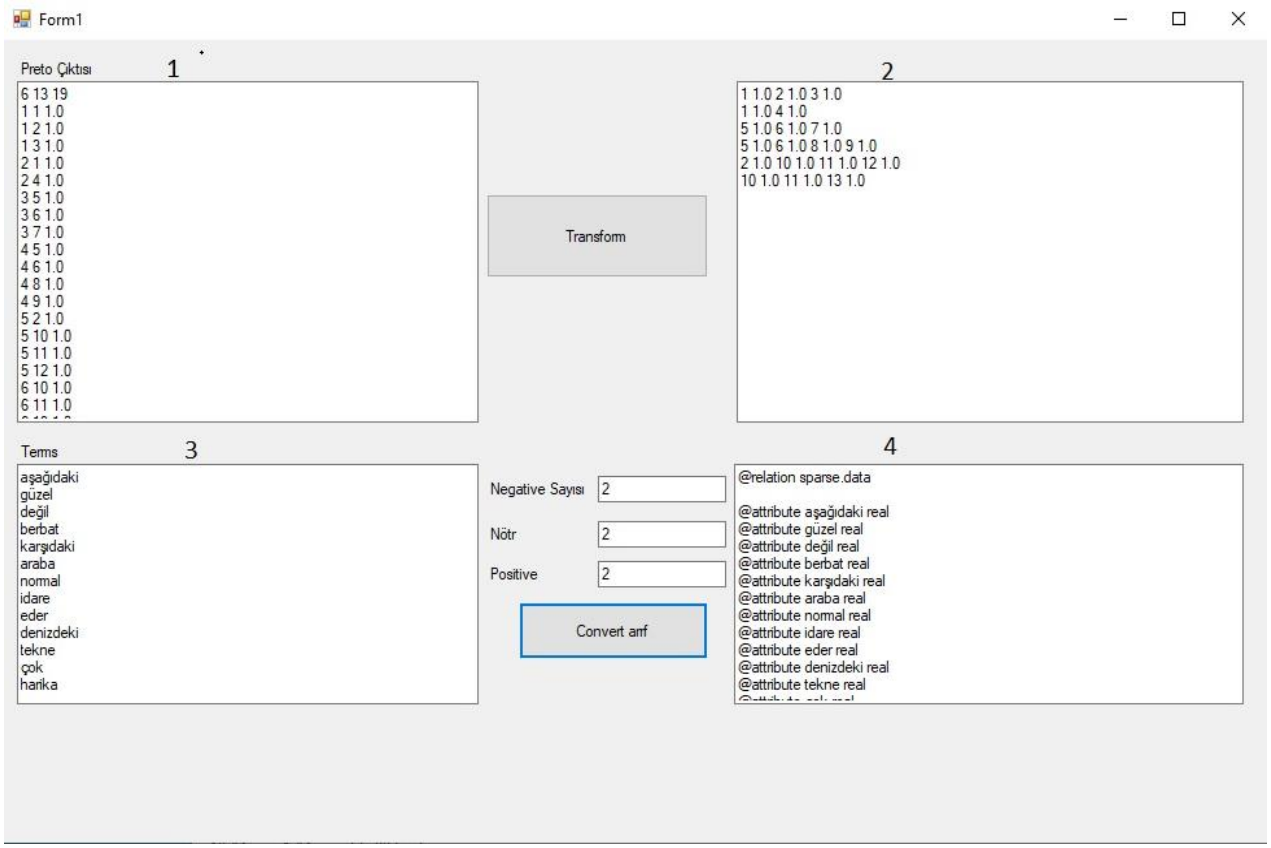
@DATA
5.1,3.5,1.4,0.2,Iris-setosa 4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa 4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa 5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa 5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa 4.9,3.1,1.5,0.1,Iris-setosa
```

Şekil 3.6. ARFF Veri Dosyası Örneği

Dosyanın birinci satırında, ilişki tipi (relation) bulunmaktadır. Daha sonraki satırdan itibaren veri grubundaki özellikler (attributes) yazılmaktadır. Özelliklerin hemen arkasında veri kümesi bulunur. Veri kümesindeki tüm satırlar bir örneğe (instance) işaret eder. Ayriyeten veri kümesinde her örnekteki özellikleri ayırmak için virgül kullanılmaktadır.

3.6.3. ARFF Converter

Visual Studio ile yazılan Windows uygulaması olan ARFF Converter Preto çıktısını Weka programının çalıştırabileceği ARFF formatına çeviren programdır. Preto çıktısını Weka seyrek matris formatına çevirir.



Şekil 3.7. ARFF Converter Ekran Görüntüsü

Bir nolu kısma Preto çıktısı olan docbyterm dosyası girdi olarak verilir. Transforma basıldığı zaman iki nolu kısımda görüldüğü gibi program docbyterm dosyasını ARFF formatı için dosyayı düzenler. Preto çıktısı olan terms dosyası üç nolu kısma girdi olarak verilir. Bunlar ARFF dosya formatındaki özellikler (attribute) olacaktır. ARFF dosya formatında data kısmında cümlelerin etiketlenebilmesi için sırası ile negatif, nötr ve pozitif cümle sayıları girilir. Etiketleme işi sırasıyla negatif, nötr ve pozitif olarak yapılmaktadır. Şekil 3.7.'deki örneğe bakıldığında altı cümleden, Preto'ya girdi olarak verilen ilk iki cümle negatif sonraki iki cümle nötr sonraki iki cümle ise pozitiftir. Son olarak convert ARFF sekmesine basıldığı zaman Weka seyrek matris formatında ARFF dosyası elde edilir.

```
@relation sparse.data

@attribute aşağıdaki real
@attribute güzel real
@attribute değil real
@attribute berbat real
@attribute karşıdaki real
@attribute araba real
@attribute normal real
@attribute idare real
@attribute eder real
@attribute denizdeki real
@attribute tekne real
@attribute çok real
@attribute harika real
@attribute class {negatif, nötr, pozitif}
@data
{0 1.0, 1 1.0, 2 1.0, 13 negatif}
{0 1.0, 3 1.0, 13 negatif}
{4 1.0, 5 1.0, 6 1.0, 13 nötr}
{4 1.0, 5 1.0, 7 1.0, 8 1.0, 13 nötr}
{1 1.0, 9 1.0, 10 1.0, 11 1.0, 13 pozitif}
{9 1.0, 10 1.0, 12 1.0, 13 pozitif}
```

Şekil 3.8. Weka Seyrek Matris Formatındaki ARFF Dosyası

3.7. Kullanılan Algoritmalar

3.7.1. Naive Bayes

Naive Bayes algoritması olasılık tabanlı bir sınıflandırıcıdır. Naive Bayes sınıflandırıcısı Bayes teoremine dayanmaktadır. Bayes teoremi Thomas Bayes tarafından bulunan koşullu olasılık hesaplama teoremidir. Algoritmanın çalışma biçimi, bir eleman için her koşulun olasılığını hesaplar ve olasılığı en fazla olana göre sınıflandırır. Naive Bayes Sınıflandırıcı örüntü tanıma sorununa bakıldığında oldukça kısıtlayıcı gözüken bir önerme ile kullanılabilen olasılıkçı yaklaşımdır. Bu önerme, örüntü tanımada kullanılacak her tanımlayıcı öznitelik veya parametrenin istatistiksel olarak bağımsız olması gerektiğidir. Bayes kuralı denklem 3.4.'te belirtilmiştir [25].

$$P(C_i \mid X) = \frac{P(X \mid C_i).P(C_i)}{P(X)} \tag{3.4}$$

$P(C_i \mid X)$ = X olan bir örneğin sınıf i'den olma olasılığıdır. (Son olasılık)

$P(X \mid C_i)$ = Sınıf i'den bir örneğin X olma olasılığıdır.

$P(C_i)$ = Sınıf i'nin ilk olasılığıdır.

$P(X)$ = Herhangi bir örneğin X olma olasılığıdır.

$P(X)$ olasılığı bütün sınıflar için sabit olduğuna göre, sadece $(C_i \mid X) = P(X \mid C_i) \times P(C_i)$ olasılığı için en büyük değer aranır.

Eğer bu basitleştirilmiş ifadede bütün özellikler bağımsız ise $P(X \mid C_i)$ denklem 3.5.'teki gibi yazılabilir.

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) = P(x_1 \mid C_i) \times P(x_2 \mid C_i) \times \ldots \times P(x_n \mid C_i) \quad (3.5)$$

Promosyon çeşitlerinin kullanımına göre cinsiyet ayrımları Tablo 3.5.'te ifade edilmiştir. Bu tablodan yola çıkarak Tablo 3.6.'daki durumda cinsiyet tahmini Naive Bayes algoritmasıyla yapılabilir.

Tablo 3.5. Promosyonlara Göre Cinsiyet Ayrımı Örnek Veri Kümesi

Magazin	**Saat**	**Hayat**	**Kredi**	**Cinsiyet**
Evet	Hayır	Hayır	Hayır	Erkek
Evet	Evet	Evet	Evet	Kadın
Hayır	Hayır	Hayır	Hayır	Erkek
Evet	Evet	Evet	Evet	Erkek
Evet	Hayır	Evet	Hayır	Kadın
Hayır	Hayır	Hayır	Hayır	Kadın
Evet	Evet	Evet	Evet	Erkek
Hayır	Hayır	Hayır	Hayır	Erkek
Evet	Hayır	Hayır	Hayır	Erkek
Evet	Evet	Evet	Hayır	Kadın

Tablo 3.6. Naive Bayes Algoritmasıyla Sınıflandırılacak Örnek Veri

Magazin	Saat	Hayat	Kredi	Cinsiyet
Evet	Evet	Hayır	Hayır	?

Cinsiyet 'erkek' için olasılık hesabı

Erkek olma önsel olasılığı: 6/10

Magazin evet iken erkek olma olasılığı: 4/6

Saat evet iken erkek olma olasılığı: 2/6

Hayat sigortası hayır iken erkek olma olasılığı: 4/6

Kredi kartı hayır iken erkek olma olasılığı: 4/6

$P(E) = 6/10 \times 4/6 \times 2/6 \times 4/6 \times 4/6 = 0{,}0593$

Cinsiyet 'kadın' için olasılık hesabı

Kadın olma önsel olasılığı: 4/10

Magazin evet iken kadın olma olasılığı: 3/4

Saat evet iken kadın olma olasılığı: 2/4

Hayat sigortası hayır iken kadın olma olasılığı: 1/4

Kredi kartı hayır iken kadın olma olasılığı: 3/4

$P(K) = 6/10 \times 3/4 \times 2/4 \times 1/4 \times 3/4 = 0{,}0281$

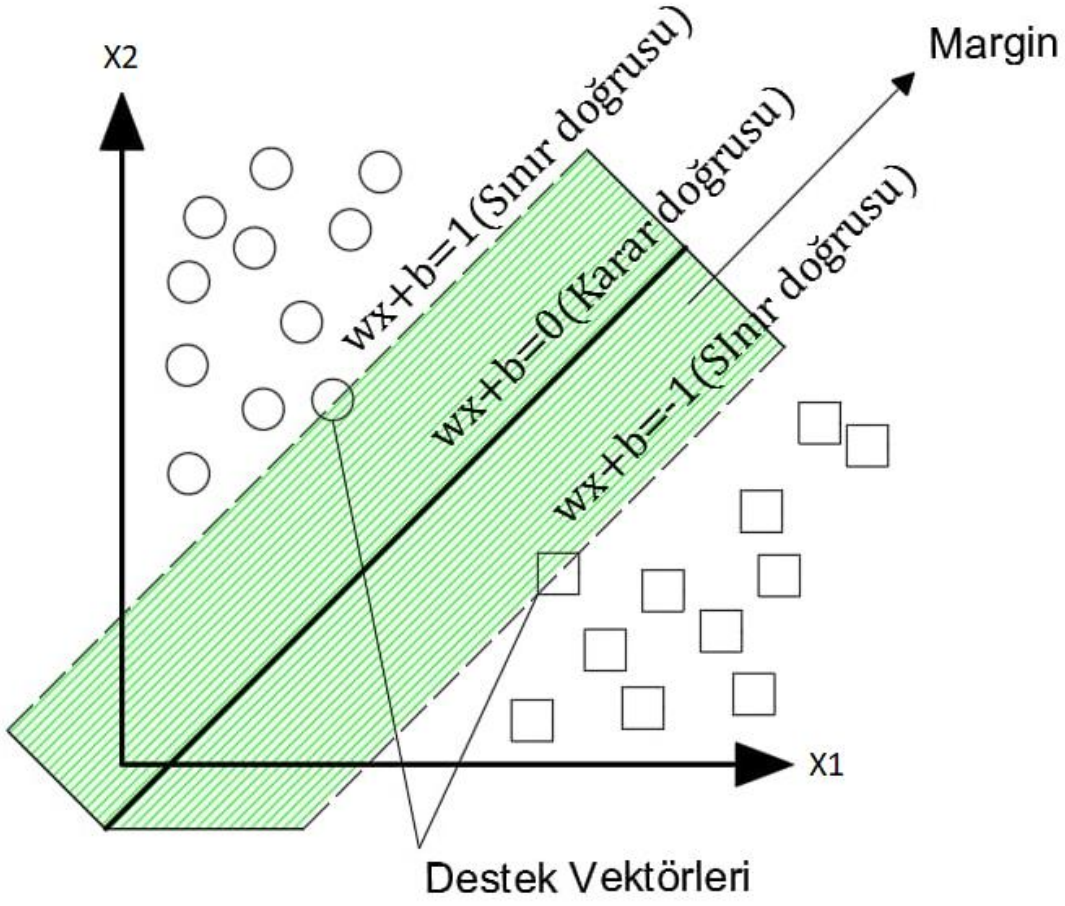
Erkek olma olasılığı, kadın olma olasılığından büyük olduğundan algoritmanın öğrendiği verilerden tahmini, erkek olma yönündedir.

3.7.2. Naive Bayes Multinomial

Naive Bayes Multinomial algoritması, Naive Bayes algoritmasının, metin belgelerinin sınıflandırılması için geliştirilen bir algoritmadır. Naive Bayes'den farklı olarak sözcükler için çok terimli dağılım kullanan Naive Bayes Multinomial, metin sınıflandırmada Naive Bayes'e göre daha yüksek başarı oranı sergilemektedir [26].

3.7.3. SMO

SMO (Sequential Minimal Optimization) Türkçe ismiyle Destek Vektör Makineleri sınıfları birbirinden ayıran doğrusal bir düzlem bulmaya çalışan sınıflandırıcıdır. Eğer iki özniteliğe sahip bir veri olursa, sınıfları birbirinden ayıran düzlem değil doğrusal bir çizgi olacaktır. Öznitelik sayısı üç tane olan bir veride ise sınıfları birbirinden ayırmak için bir düzleme ihtiyaç olacaktır. Eğer öznitelik sayısı dört ve üstü olan bir veride örnekleri birbirinden ayırmak için bir hiper düzleme ihtiyaç olacaktır. DVM doğrusal düzlemi, destek vektörlerine göre tanımlanan maksimum aralık ile bulmaya çalışır [27]. Destek vektörleri ve ayırıcı düzlem kavramları Şekil 3.9. üzerinden anlatılmıştır.



Şekil 3.9. DVM Destek Vektörleri ve Hiper Düzlemi [28].

Şekil 3.9.'da iki farklı sınıfa ait örnekler bulunmaktadır. Şekilde her örnek X_1 ve X_2 olmak üzere iki öznitelik değerine sahiptir. Bir örneğin X ile temsil edildiğini kabul edersek eğer, X = (X_1, X_2) şeklinde tanımlanabilir. DVM'nin amacı en optimal doğruyu çizmektir. Bunun için sınıflar arasındaki birbirine en yakın iki nokta seçilir. Bu noktalara destek vektörleri denir. Destek vektörlerinin bulunduğu noktadan geçecek şekilde birer doğru çizilir. Bu doğrulara ise sınır doğruları denir. İki sınır doğrusu arasında kalan bölgeye margin adı verilir. Margin ne kadar geniş ise iki veya daha fazla sınıf o kadar iyi ayrıştırılmış demektir. Karar doğrusu diğer adıyla hiper düzlem ise her sınıfın sınır doğrusuna eşit uzaklıkta bulunan doğrudur. DVM'lerde sınıflar +1 ve -1 olarak etiketlenir. Karar doğrusunun üst kısmında kalan doğrunun denklemine $wx + b = 1$ olarak alt kısmında kalan doğrunun denklemine ise $wx + b = -1$ olur. Karar doğrusunun denklemi ise $wx + b = 0$ şeklindedir. Her sınır doğrusunun formülü karar doğrusuna eşit uzaklıkta olacak şekilde tasarlanmıştır [28].

3.7.4. IBK

IBK diğer adıyla KNN en yakın komşu algoritması olarak bilinir. KNN sınıflandırıcısı, etiketli veri seti üzerinden sınıflandırılacak X örneğine en yakın 'K' tanesini belirleyerek sınıflandırma yapar. K değeri kadar elaman incelenir. Bir değer geldiğinde en yakın K kadar eleman alınarak gelen değer arasındaki uzaklık hesaplanır. Uzaklık hesaplama işleminde genelde Öklid fonksiyonu kullanılır. Yapılan Öklid uzaklığı hesabında en yakın değere sahip K tane elamanının sınıfına bakılıp en çok hangi sınıfta elaman mevcutsa ona göre sınıflama yapar [29]. Öklid uzaklığının formülü denklem 3.6.'da verilmiştir.

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \tag{3.6}$$

Tablo 3.7.'de beş adet örneğe ait beş tane öznitelik değerleri verilmiştir. Bu değerlere göre bir ürünün satın alınıp alınmayacağına göre sınıflandırma yapılmıştır. Bu veri seti baz alındığında Tablo 3.8.'deki 'f' örneğinin K = 3 alınarak sınıfının ne olacağı

KNN algoritması ile hesaplanabilir. Tablo 3.9.'da her örnek için Öklid uzaklığı hesaplanmıştır.

Tablo 3.7. IBF Algoritması İçin Örnek Veri Kümesi

	t1	t2	t3	t4	t5	Class
a	2	3	4	5	5	Buy
b	0	1	2	3	5	Buy
c	2	2	2	2	4	Sell
d	3	3	3	3	3	Sell
e	4	2	4	4	4	Sell

Tablo 3.8. IBF Algoritmasıyla Sınıflandırılacak Örnek Veri

	t1	t2	t3	t4	t5	Class
f	1	2	3	4	4	?

Tablo 3.9. Örnek Veri İçin Öklid Uzaklığının Hesaplanması

	t1 farkı	t2 farkı	t3 farkı	t4 farkı	t5 farkı	
a	1	1	1	1	1	$\sqrt{1^2+1^2+1^2+1^2+1^2} = \sqrt{5}$
b	1	1	1	1	1	$\sqrt{1^2+1^2+1^2+1^2+1^2} = \sqrt{5}$
c	1	0	1	2	0	$\sqrt{1^2+0^2+1^2+2^2+0^2} = \sqrt{6}$
d	2	1	0	1	1	$\sqrt{2^2+1^2+0^2+1^2+1^2} = \sqrt{7}$
e	3	0	1	0	0	$\sqrt{3^2+0^2+1^2+0^2+0^2} = \sqrt{10}$

Öklid uzaklıkları dikkate alındığı zaman en yakın üç örnek a, b ve c örnekleridir. Bu üç örnekten iki tanesi satın alma bir tanesi satın almama sınıfındadır. En yakın üç örnekten çoğunluğu satın alma sınıfında bulunduğu için Tablo 3.8.'deki durumda algoritmanın yapacağı sınıflandırma satın alma yönünde olacaktır.

3.7.5. J48

J48 algoritması diğer adıyla C4.5 algoritması bir karar ağacı ortaya koyar. Bu karar ağacı modelinde, daha anlaşılır ve işlevsel bir ağaç olması için mümkün oldukça küçük bir ağaç olmaya çalışır. Bunun için ağacı oluşturmaya başlamak için hangi değişkenin öncelikli olarak alınacağını belirlemek için information gain ve entropi gibi

değerler kullanılır. Herhangi bir değişkenin information gain değeri ne kadar büyükse dallandırma konusunda o kadar etkindir. Ağacın ilk dallandırması information gain'in en fazla olduğu yerden başlanır. Alt dallanmalar için aynı hesap yapılarak ağaç oluşturulur. Information gain hesaplama şekli denklem 3.7'de gösterilmektedir [30].

$$Gain(A) = Info(D) - Info_A(D) \tag{3.7}$$

Entropi, bir veri setinde örneklerin sınıflara göre dağılımının ne kadar düzgün olduğunu ifade eder. Denklem 3.7.'de Info(D) ifadesi, bir veri setinin entropisini gösterir. $Info_A(D)$ ise A özniteliğinin aldığı değerlere göre, örneklerin bu değerlerdeki entropisini ifade eder. Yani information gain veri setinin entropisinden, özniteliğin entropi değerinin çıkartılması ile elde edilir [30]. Denklem 3.8'de Info(D) ve $Info_A(D)$ ifadelerinin hesaplama formülleri gösterilmektedir.

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$
$$Info_A(D) = -\sum_{j=1}^{y} \frac{|D_j|}{|D|} \times Info(D_j) \tag{3.8}$$

Info(D) eşitliğinde m değeri sınıf sayısını, p_i değeri ise her sınıfın önsel olasılığını ifade eder. $Info_A(D)$ eşitliğinde y değeri, A özelliğinin aldığı farklı değerlerin sayısıdır. $|D_j|/|D|$ ifadesi ise, A özelliğinin D_j değerine sahip olan örneklerin toplam örneklere olan oranını gösterir. Info(Dj) ise A özelliğinin D_j değerine sahip olan örneklerin entropisini ifade etmektedir. Tablo 3.10.'da günlere göre; hava durumu, sıcaklık, nem ve rüzgar özniteliklerine göre tenis oynama durumunu gösteren veri seti mevcuttur. Bu veri esas alınarak J48 algoritmasına göre ağacın hangi öznitelikten başlayacağı hesaplanabilir.

Tablo 3.10. J48 Algoritması İçin Örnek Veri Kümesi

Günler	**Hava**	**Sıcaklık**	**Nem**	**Rüzgar**	**Durum**
Gün 1	Güneşli	Sıcak	Yüksek	Az	Hayır
Gün 2	Güneşli	Sıcak	Yüksek	Güçlü	Hayır
Gün 3	Bulutlu	Sıcak	Yüksek	Az	Evet
Gün 4	Yağmurlu	Ilık	Yüksek	Az	Evet
Gün 5	Yağmurlu	Soğuk	Normal	Az	Evet
Gün 6	Yağmurlu	Soğuk	Normal	Güçlü	Hayır
Gün 7	Bulutlu	Soğuk	Normal	Güçlü	Evet
Gün 8	Güneşli	Ilık	Yüksek	Az	Hayır
Gün 9	Güneşli	Soğuk	Normal	Az	Evet
Gün 10	Yağmurlu	Ilık	Normal	Az	Evet
Gün 11	Güneşli	Ilık	Normal	Güçlü	Evet
Gün 12	Bulutlu	Ilık	Yüksek	Güçlü	Evet
Gün 13	Bulutlu	Sıcak	Normal	Az	Evet
Gün 14	Yağmurlu	Ilık	Yüksek	Güçlü	Hayır

Tablo 3.10.'daki duruma bakıldığında toplam 14 durumdan 9 tanesi evet 5 tanesi hayırdır. Bu durumda Info(D)'i hesaplanırsa eğer;

$Info(D) = I(9,5) = -\left(\frac{9}{14}.\log_2\frac{9}{14}\right) - \left(\frac{5}{14}.\log_2\frac{5}{14}\right) = 0{,}94$ değeri elde edilir. (Veri setinin entropi değeri)

Tablo 3.11.'de Hava durumuna göre tenis oynama durumu gösterilmiştir.

Tablo 3.11. Hava Durumuna Göre Tenis Oynama Durumu

		Durum	
		Evet	Hayır
Hava Durumu	Güneşli	2	3
	Bulutlu	4	0
	Yağmurlu	3	2

Tablo 3.11.'e göre Info $_{hava}$(D) değeri hesaplanabilir.

$Info_{Hava}(D) = p(güneşli)$.I(2,3) + p(bulutlu).I(4,0) + p(yağmurlu).I(3,2)

$Info_{Hava}(D) = \frac{5}{14}.I(2,3) + \frac{4}{14}.I(4,0) + \frac{5}{14}.(3,2) = 0{,}694$

Hesaplamalar sonucunda;

$Gain(hava) = 0{,}94 - 0{,}694 = 0{,}246$ değeri elde edilir.

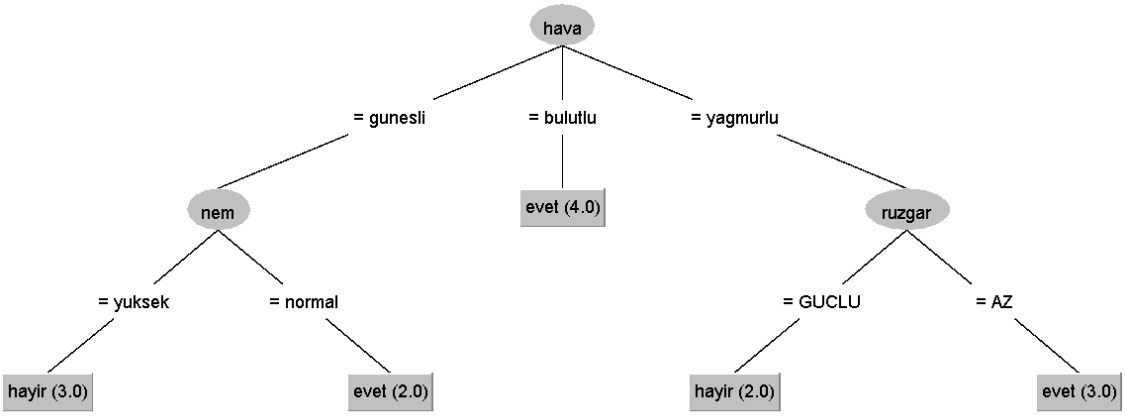
Aynı hesaplamalar sıcaklık, nem ve rüzgar durumları için yapılıp sonuçlar belirtilmiştir.

$Gain(sıcaklık)$ =0,048

$Gain(nem)$ =0,152

$Gain(ruzgar)$ =0,029

Information gain değerlerine bakıldığı zaman en büyük değer havaya aittir. Buna göre ağacımız dallanmaya 'Hava' özniteliğinden başlayacaktır. Şekil 3.10.'da bu verinin J48 algoritmasına göre ağaç yapısı gösterilmiştir.



Şekil 3.10. Veri Kümesinin J48 Algotimasına Göre Oluşan Ağaç Yapısı

3.7.6. Random Forest

Topluluk öğrenme yöntemi olan Random Forest (Rastgele Orman) algoritması, sınıflandırma işlemi sırasında birden daha fazla karar ağacı üreterek sınıflandırma başarısını arttırmayı amaçlayan bir algoritmadır. Bireysel olarak oluşturulan karar ağaçları bir araya gelerek bir karar ormanı oluşturur. Bu ormandaki karar ağaçları bağlı bulundukları veri setinden gelişi güzel seçilmiş birer alt kümedir. Dallanma ölçülerinin belirlenmesinde, karar ağacında olduğu gibi en çok kullanılan yöntem Gini indeksidir. Gini indeksi, sınıfların homojenliğini belirlemek için de kullanılır. Gini indeksi ne kadar

az olursa sınıf bir o kadar homojendir. Dalın başarılı olabilmesi için, bir alt düğümün Gini indeksi, bir üst düğümün Gini indeksinden daha az olması yeterlidir [19].

3.8. Değerlendirme Ölçütleri

Algoritmalardan elde edilen sonuçlar değerlendirilirken bazı değerlere bakılır. Bu değerler;

-Doğruluk (Accuracy)

-Kesinlik (Precision)

-Hassasiyet (Recall)

-F1 Skor (F-Skor)

-ROC alanı (ROC area) değerleridir.

Bu değerlerin hesaplanabilmesi için belirtilen tanımların bilinmesi gerekmektedir.

TP= Pozitif olanın, pozitif olarak etiketlenmesi

TN=Negatif olanın, negatif olarak etiketlenmesi

FP=Negatif olanın, pozitif olarak etiketlenmesi

FN=Pozitif olanın, negatif olarak etiketlenmesi

3.8.1. Doğruluk

Doğruluk; "Sınıflandırıcı bütün örnekleri sınıflandırmada ne kadar başarılıdır" sorusunun cevabıdır.

$$Doğruluk = \frac{TP+TN}{TP+TN+FP+FN} \text{ olarak hesaplanır} \quad (3.9)$$

3.8.2. Kesinlik

Kesinlik; *"*Getirilen bilginin ne kadarı istenilen bilgiyle ilgilidir" sorusunun cevap değeridir. Diğer bir deyişle yanlış pozitifleri eleme kabiliyetidir.

$$Kesinlik = \frac{TP}{TP+FP} \text{ olarak hesaplanır.} \quad (3.10)$$

3.8.3. Hassasiyet

Hassasiyet; *"*Getirilmesi gereken bilginin ne kadarı getirilmiştir" sorusunun cevap değeridir. Diğer bir deyişle doğru pozitifleri tespit etme kabiliyetidir.

$$Hassasiyet = \frac{TP}{TP+FN} \text{ olarak hesaplanır.} \quad (3.11)$$

3.8.4. F1 Skor

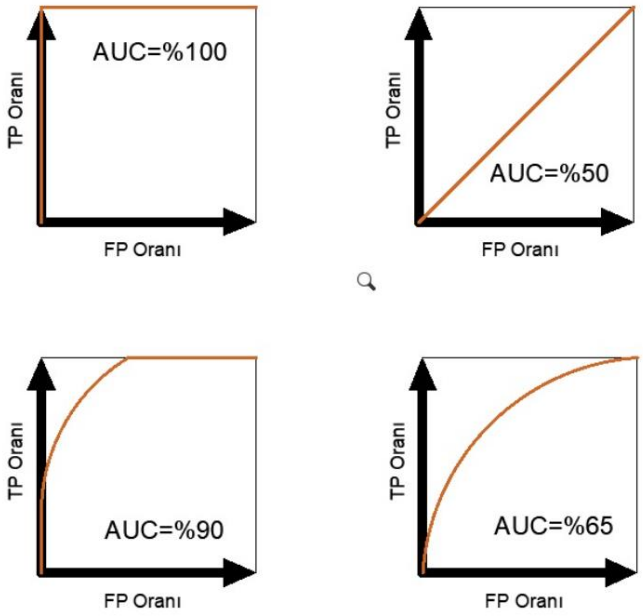
F1 Skor*;* Sınıflandırıcının hem kesinliğini hem de duyarlılığını göz önünde bulunduran değerlendirme metriğidir. Kesinlik ve hassasiyet değerlerinin harmonik ortalamasıdır.

$$F1\ skor = 2\ .\ \frac{Kesinlik\ .\ Duyarlılık}{Kesinlik+\ Duyarlılık} \text{ olarak hesaplanır.} \quad (3.12)$$

3.8.5. ROC Eğrisi

Her ayırma işlemlerinde olduğu gibi, yöntemler duyarlılık ve keskinlik arasındaki dengeyi inşa etmekle ilgilenmektedir. Veri guruplarındaki olumlu veya olumsuz modeller, denk olacak gibi bir dağılım göstermediği için, direkt doğruluk ve hassasiyet kriterlerinden önce, ROC (Receiver Operating Characteristics) olarak isimlendirilen eğri doğruluk ve hassasiyet arasındaki dengeyi yorumlamak için kullanılır. ROC eğrisi altındaki alan AUC (Area Under Curve) olarak tanımlanır. ROC eğrisi değişen gruplandırma taban verilerine göre, doğru pozitiflerin sayısının, yanlış pozitiflerin bir işlevi olarak çizilmesiyle oluşur. ROC sayısı '1' olduğunda, pozitifler çok iyi bir şekilde negatiflerden ayrılmış demektir. Şayet ROC sayısı '0' olduğunda ise rastgele bir pozitif bulunamadı manasına gelir. ROC basitçe doğru pozitiflerin yanlış pozitiflere olan kesri

olarak söylenebilir. İçeriğindeki alan büyüklüğü arttıkça, makine öğrenme modellerinin sınıfları ayırt etme yeteneği artar [31] .



Şekil 3.11. ROC Eğrileri ve Altında Kalan Alanlar

BÖLÜM 4. DENEYLER

Belgeleri önişlemeden geçirdikten sonra Weka ile deneyler yapılmıştır. Elde edilen veri grubu üstünde Naive Bayes, Naive Bayes Multinomial, SMO (Support Vector Machine), IBK (KNN), J48 ve Random Forest gibi çeşitli sınıflandırma algoritmalarıyla test edilmiştir. Deneyler yapılırken kullanılan sınıflandırıcılar, Weka'daki kabul edilen parametreler ile çalıştırılmıştır. Veri guruplarındaki örnek sayının düşük olmasından dolayı bütün deneylerde 10 kat çapraz doğrulama uygulanmıştır.

4.1. Deney 1

Bu deneyde durak sözcük filtreleme ve kök alma önişleme yöntemleri kullanılmamış ve *n*-gram uzunluğu 1 olarak alınmıştır. Toplam 6.508 farklı öznitelik oluşmuştur. Deney sonuçları Tablo 4.1.'de gösterilmiştir.

Tablo 4.1. Deney 1 Sonuçları

Algoritma	TP Oranı		FP Oranı		Kesinlik		Hassasiyet		F Skor		ROC Alanı	
	TF	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*
NaiveBayes	0,838	0,842	0,153	0,152	0,845	0,848	0,838	0,842	0,839	0,843	0,898	0,899
NaiveBayesMultinomial	**0,899**	0,894	0,132	**0,115**	**0,902**	0,894	**0,899**	0,894	**0,898**	0,894	**0,970**	0,958
SMO	0,887	0,883	0,119	0,123	0,888	0,884	0,887	0,883	0,887	0,883	0,884	0,880
IBK(k=3)	0,470	0,537	0,361	0,321	0,707	0,730	0,470	0,537	0,378	0,487	0,657	0,701
J48	0,779	0,780	0,241	0,241	0,779	0,780	0,779	0,780	0,779	0,780	0,848	0,849
RandomForest	0,894	0,888	0,125	0,128	0,894	0,888	0,894	0,888	0,894	0,888	0,958	0,956

4.2. Deney 2

Bu deneyde durak sözcük listesi kullanılmıştır. Kök alma yöntemi uygulanmamış ve *n*-gram uzunluğu bir olarak alınmıştır. Toplam 6.324 farklı öznitelik oluşmuştur. Deney sonuçları Tablo 4.2.'de gösterilmektedir.

Tablo 4.2. Deney 2 Sonuçları

Algoritma	TP Oranı		FP Oranı		Kesinlik		Hassasiyet		F Skor		ROC Alanı	
	TF	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*
NaiveBayes	0,844	0,845	0,153	0,156	0,849	0,848	0,844	0,845	0,845	0,846	0,910	0,907
NaiveBayesMultinomial	0,896	0,890	0,135	0,116	0,898	0,890	0,896	0,890	0,895	0,890	**0,969**	0,957
SMO	0,881	0,884	0,127	0,123	0,881	0,884	0,881	0,884	0,881	0,884	0,877	0,880
IBK(k=3)	0,455	0,512	0,369	0,339	0,707	0,710	0,455	0,512	0,348	0,452	0,631	0,670
J48	0,786	0,786	0,240	0,240	0,784	0,784	0,786	0,786	0,785	0,785	0,856	0,856
RandomForest	0,903	**0,909**	0,111	**0,099**	0,903	**0,909**	0,903	**0,909**	0,903	**0,909**	0,963	0,963

4.3. Deney 3

Bu deneyde durak sözcük listesi kullanılmıştır. Ek çıkaran kök bulucu, kök alma işlemi uygulanmış ve *n*-gram uzunluğu 1 olarak alınmıştır. Toplam 3.912 farklı öznitelik oluşmuştur. Deney sonuçları Tablo 4.3.'te gösterilmektedir.

Tablo 4.3. Deney 3 Sonuçları

Algoritma	TP Oranı		FP Oranı		Kesinlik		Hassasiyet		F Skor		ROC Alanı	
	TF	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*
NaiveBayes	0,846	0,836	0,171	0,187	0,846	0,835	0,846	0,836	0,846	0,835	0,846	0,894
NaiveBayesMultinomial	**0,898**	0,893	0,134	**0,120**	**0,900**	0,893	**0,898**	0,893	0,896	0,893	**0,967**	0,956
SMO	0,875	0,878	0,133	0,129	0,875	0,878	0,875	0,878	0,875	0,878	0,871	0,874
IBK(k=3)	0,576	0,706	0,332	0,294	0,673	0,716	0,576	0,706	0,562	0,708	0,678	0,753
J48	0,773	0,773	0,267	0,267	0,771	0,771	0,773	0,773	0,770	0,770	0,833	0,833
RandomForest	0,891	**0,898**	0,129	0,122	0,891	0,898	0,891	**0,898**	0,890	**0,897**	0,961	0,959

4.4. Deney 4

Bu deneyde durak sözcük listesi kullanılmıştır. Zemberek kütüphanesi kök bulma, kök alma işlemi uygulanmış ve *n*-gram uzunluğu bir olarak alınmıştır. Toplam 2.385 farklı öznitelik oluşmuştur. Deney sonuçları Tablo 4.4.'te gösterilmektedir.

Tablo 4.4. Deney 4 Sonuçları

Algoritma	TP Oranı		FP Oranı		Kesinlik		Hassasiyet		F Skor		ROC Alanı	
	TF	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*
NaiveBayes	0,826	0,822	0,198	0,204	0,825	0,821	0,826	0,822	0,825	0,820	0,871	0,869
NaiveBayesMultinomial	0,868	0,869	0,173	0,149	0,872	0,868	0,868	0,869	0,865	0,868	**0,949**	0,935
SMO	0,858	0,862	0,150	**0,145**	0,858	0,863	0,858	0,862	0,858	0,862	0,854	0,858
IBK(k=3)	0,678	0,694	0,331	0,288	0,685	0,715	0,678	0,694	0,680	0,697	0,709	0,747
J48	0,767	0,767	0,252	0,252	0,767	0,767	0,767	0,767	0,767	0,767	0,802	0,802
RandomForest	**0,878**	0,877	**0,145**	0,146	**0,877**	**0,877**	**0,878**	0,877	**0,877**	0,876	0,946	0,948

4.5. Deney 5

Bu deneyde durak sözcük listesi kullanılmıştır. Sabit önek (*n*=3), kök alma işlemi uygulanmış ve *n*-gram uzunluğu bir olarak alınmıştır. Toplam 1.142 farklı öznitelik oluşmuştur. Deney sonuçları Tablo 4.5.'te gösterilmektedir.

Tablo 4.5. Deney 5 Sonuçları

Algoritma	TP Oranı		FP Oranı		Kesinlik		Hassasiyet		F Skor		ROC Alanı	
	TF	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*
NaiveBayes	0,814	0,813	0,206	0,202	0,814	0,813	0,814	0,813	0,814	0,813	0,860	0,860
NaiveBayesMultinomial	0,856	0,834	0,183	0,189	**0,858**	0,833	0,856	0,834	0,853	0,833	**0,935**	0,908
SMO	0,848	0,845	**0,162**	0,164	0,848	0,846	0,848	0,845	0,848	0,845	0,843	0,840
IBK(k=3)	0,670	0,702	0,338	0,303	0,677	0,710	0,670	0,702	0,672	0,705	0,704	0,742
J48	0,730	0,730	0,294	0,294	0,729	0,729	0,730	0,730	0,730	0,730	0,747	0,747
RandomForest	0,851	**0,858**	0,182	0,172	0,852	**0,858**	0,851	**0,858**	0,849	**0,856**	0,932	**0,935**

4.6. Deney 6

Bu deneyde durak sözcük listesi kullanılmıştır. Sabit önek (*n*=4), kök alma işlemi uygulanmış ve *n*-gram uzunluğu 1 olarak alınmıştır. Toplam 1.993 farklı öznitelik oluşmuştur. Deney sonuçları Tablo 4.6.'da gösterilmektedir.

Tablo 4.6. Deney 6 Sonuçları

Algoritma	TP Oranı		FP Oranı		Kesinlik		Hassasiyet		F Skor		ROC Alanı	
	TF	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*
NaiveBayes	0,828	0,838	0,186	0,174	0,828	0,839	0,828	0,838	0,828	0,838	0,874	0,878
NaiveBayesMultinomial	**0,886**	0,876	0,147	0,143	**0,888**	0,876	**0,886**	0,876	**0,884**	0,875	**0,953**	0,934
SMO	0,865	0,868	0,146	0,141	0,865	0,868	0,865	0,868	0,865	0,868	0,859	0,864
IBK(k=3)	0,712	0,726	0,299	0,267	0,717	0,738	0,712	0,726	0,714	0,728	0,714	0,772
J48	0,761	0,761	0,255	0,255	0,762	0,762	0,761	0,761	0,761	0,761	0,815	0,815
RandomForest	0,882	0,883	0,137	**0,135**	0,882	0,883	0,882	0,883	0,882	0,883	0,950	0,948

4.7. Deney 7

Bu deneyde durak sözcük listesi kullanılmıştır. Ek çıkaran kök bulucu, kök alma işlemi uygulanmış ve *n*-gram uzunluğu iki olarak alınmıştır. Toplam 14.912 farklı öznitelik oluşmuştur. Deney sonuçları Tablo 4.7.'de gösterilmektedir.

Tablo 4.7. Deney 7 Sonuçları

Algoritma	TP Oranı		FP Oranı		Kesinlik		Hassasiyet		F Skor		ROC Alanı	
	TF	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*
NaiveBayes	0,809	0,676	0,280	0,234	0,844	0,782	0,809	0,676	0,794	0,669	0,899	0,820
NaiveBayesMultinomial	0,838	0,742	**0,141**	0,385	**0,853**	0,816	0,838	0,742	0,840	0,705	0,931	**0,940**
SMO	0,852	**0,858**	0,169	0,187	0,851	0,864	0,852	**0,858**	0,851	**0,855**	0,842	0,836
IBK(k=3)	0,402	0,426	0,398	0,382	0,760	0,764	0,402	0,426	0,234	0,284	0,615	0,503
J48	0,599	0,599	0,601	0,601	0,360	0,360	0,599	0,599	0,450	0,450	0,523	0,523
RandomForest	0,800	0,782	0,184	0,178	0,814	0,816	0,800	0,782	0,802	0,783	0,905	0,902

4.8. Deney 8

Bu deneyde durak sözcük listesi kullanılmıştır. Zemberek kütüphanesi kök bulma, kök alma işlemi uygulanmış ve *n*-gram uzunluğu iki olarak alınmıştır. Toplam 13.336 farklı öznitelik oluşmuştur. Deney sonuçları Tablo 4.8.'de gösterilmektedir.

Tablo 4.8. Deney 8 Sonuçları

Algoritma	TP Oranı		FP Oranı		Kesinlik		Hassasiyet		F Skor		ROC Alanı	
	TF	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*	*TF*	*TF-IDF*
NaiveBayes	0,791	0,791	0,295	0,295	0,816	0,816	0,791	0,791	0,776	0,776	0,879	0,879
NaiveBayesMultinomial	0,833	0,794	0,165	0,167	0,838	0,827	0,833	0,794	0,834	0,796	**0,918**	0,904
SMO	**0,841**	0,840	0,166	**0,163**	**0,842**	**0,842**	**0,841**	0,840	**0,841**	**0,841**	0,837	0,838
IBK(k=3)	0,426	0,426	0,412	0,412	0,541	0,541	0,426	0,426	0,329	0,329	0,573	0,573
J48	0,608	0,616	0,586	0,571	0,688	0,702	0,608	0,616	0,472	0,493	0,517	0,516
RandomForest	0,781	0,788	0,181	0,177	0,813	0,817	0,781	0,788	0,783	0,790	0,887	0,889

BÖLÜM 5. DENEYLER VE TARTIŞMA

Bu tez kapsamında toplam sekiz farklı deney yapılmış ve sonuçlar TF ve TF-IDF olmak üzere iki farklı terim ağırlıklandırma yöntemi ile verilmiştir. Tablo 5.1'de farklı deneylerde elde edilen öznitelik sayıları verilmiştir.

Tablo 5.1. Deneylerde Elde Edilen Öznitelik Sayıları

No	Durak Sözcükleri	Kök Bulma	*n*-Gram	Öznitelik Sayısı
1	Hayır	Hayır	1	6.508
2	Evet	Hayır	1	6.324
3	Evet	Ek Çıkaran	1	3.912
4	Evet	Zemberek	1	2.385
5	Evet	Sabit Önek (*n*=3)	1	1.142
6	Evet	Sabit Önek (*n*=4)	1	1.993
7	Evet	Ek Çıkaran	2	14.912
8	Evet	Zemberek	2	13.336

Tablo 5.1'e bakıldığında kök alma yöntemleri, öznitelik sayısında ciddi bir azalmaya sebep olmaktadır. Bu sayede daha sade bir veri seti elde edilir. Zemberek kütüphanesi kök bulma yöntemi kullanıldığı zaman ek çıkaran kök bulma yöntemine göre öznitelik sayısını daha fazla azalttığı görülmektedir. *n*-gram sayısı arttığı zaman ise öznitelik sayısında ciddi oranda artış bulunmaktadır.

Deney 1 ve Deney 2 karşılaştırıldığında aynı koşullar altında durak sözcük kullanma durumunun sonuca etkisi gözlemlenebilir. Elde edilen F-Skor değerlerine bakıldığında, durak sözcük kullanımı bazı algoritmalarda pozitif etki gösterirken bazı algoritmalarda da negatif etki göstermektedir. Durak sözcük kullanımıyla öznitelik sayısı

azalarak daha temiz bir veri elde edilmiş ama sonuca etkisi hakkında kesin bir şey söylenememektedir.

Deney 3-4-5-6 karşılaştırıldığında aynı koşullar altında farklı kök alma yöntemlerinin sonuca etkisi gözlemlenebilir. Elde edilen F-Skor değerlerine bakıldığında, ek çıkaran kök bulucu, kök bulma yönteminin IBK hariç diğer algoritmalarda daha başarılı olduğu gözükmektedir. Buna göre en başarılı kök bulma yöntemi ek çıkaran kök bulucu, kök bulma yöntemi denebilir. Sabit önek yönteminde karakter uzunluğunun dört olduğunda, üç olmasına göre daha başarılı sonuçlar vermektedir. Elde edilen deney sonuçlarına göre kök bulma yöntemleri başarı sırasına göre, ek çıkaran kök bulucu, sabit önek (4), zemberek kütüphanesi ve sabit önek (3) şeklinde sıralanabilir. Basit bir kök bulma yöntemi olan sabit önek, karakter uzunluğu dört seçildiğinde, sözlük alt yapısı gerektiren Zemberek kütüphanesiyle karşılaştırıldığında üstünlük sağlaması dikkat çekicidir.

Deney 7 ve Deney 8 de *n*-gram sayısı 2 olarak seçilmiş ve sonuçlara bakıldığında başarı oranını azaldığı görülmektedir. *n*-gram sayısının arttırılması başarı oranına katkısı bulunmadığı gibi, negatif yönde etkilemektedir.

Tüm deney sonuçlarına bakıldığında IBK algoritmasının duygu analizinde en başarısız algoritma olduğu gözükmektedir. Naive Bayes ve Naive Bayes Multinomial deney sonuçlarına bakıldığında Naive Bayes Multinomial'ın Naive Bayes'e göre üstünlük sağladığı gözükmektedir. Naive Bayes Multinomial'ın yürütme zamanının daha kısa olması, başarısının yanında artı bir değer katmaktadır.

Terim ağırlıklandırma yöntemlerinin başarıya katkısı karşılaştırıldığında, her iki yöntemin birbirine karşı net bir üstünlük sağlamadığı gözükmektedir.

Tüm Deneylerdeki en başarılı sonuç Deney 2'de TF-IDF formda %90,9 F-Skor değeriyle Random Forest algoritmasıdır. Diğer deney sonuçlarına da bakıldığında en başarılı sonuçlar veren algoritma Random Forest algoritmasıdır.

BÖLÜM 6. SONUÇ

Bu tez çalışmasında Türkçe otomobil yorumlarının duygu analizi yapılmıştır. Çeşitli önişleme yöntemleri, farklı algoritmalarla denenmiş ve çıkan sonuçlar yorumlanmıştır.

Sonuçlar yorumlandığında Random Forest algoritmasının diğer algoritmalara göre başarı üstünlüğü sağladığı görülmüştür. Kök alma işlemi sayesinde öznitelik uzay boyutu azaltılmıştır. Ancak algoritma başarısındaki etkisine bakıldığında anlamlı bir katkı sağlamadığı görülmüştür. Kök alma yöntemleri karşılaştırıldığında en başarılı yöntem ek çıkaran kök bulma yöntemi olmuştur. En başarılı sonuçlar *n*-gram sayısı bir iken elde edilmiştir *n*-gram sayısının arttırılmasıyla birlikte deney sonuçlarında net bir düşüş olduğu saptanmıştır.

Tez çalışmasındaki deney sonuçlarına göre makine öğrenmesi yöntemiyle yapılacak duygu analizi çalışmalarında, kök bulma yöntemi olarak ek çıkaran kök bulma yöntemi, algoritma olarak Random Forest algoritmasının *n*-gram bir olacak şekilde kullanılması tavsiye edilir.

KAYNAKÇA

[1] Dünya Nüfusu, [Çevrimiçi]. Available: https://tr.wikipedia.org/wiki/Dünya_nüfusu. [Erişildi: 10 01 2021].

[2] M. Ö. Dolgun, T. G. Özdemir ve D. Oğuz, "Veri madenciliği'nde yapısal olmayan verinin analizi: Metin ve web madenciliği," *İstatikçiler Dergisi,* pp. 48-58, 2009.

[3] O. Kaynar, Y. Görmez, M. Yıldız ve A. Albayrak, " Makine Öğrenmesi Yöntemleri ile Duygu Analizi, "*International Artificial Intelligence and Data Processing Symposium*," pp. 234-241, 2016.

[4] Ö. Çoban, B. Özyer ve G. Özyer, "Sentiment analysis for Turkish Twitter feeds" *23nd Signal Processing and Communications Applications Conference (SIU),* pp. 2388-2391, 2015.

[5] B. İ. Sevindi, "Türkçe metinlerde denetimli ve sözlük tabanlı duygu analizi yaklaşımlarının karşılaştırılması," *Gazi Üniversitesi Fen Bilimleri Enstitüsü, Yayımlanmış Yüksek Lisans Tezi,* 2013.

[6] E. Kumaş, "Türkçe Twitter Verilerinden Duygu Analizi Yapılırken Sınıflandırıcıların Karşılaştırılması," *ESTUDAM Bilişim Dergisi,* cilt 2, no. 2, pp. 1-5, 2021.

[7] B. Köksal, G. Erdem, C. Türkeli ve Z. Öztürk, "Twitter'da Duygu Analizi Yöntemi Kullanılarak Bitcoin Değer Tahminlemesi," *Düzce Üniversitesi Bilim ve Teknoloji Dergisi,* pp. 280-297, 09.2021.

[8] İ. N. Yoldaş, "Türkçe Metinlerde Duygu Analizi: Sözlük Tabanlı Yaklaşım ve İnsanların Tepkilerinin Karşılaştırılması," *ESTUDAM Bilişim Dergisi,* cilt 2, no. 1, pp. 1-6, 2021.

[9] Ç. Aytekin ve M. A. Bayram, "A Research on Developing Context Free Models on Communication With Sentiment Approach for Turkish Texts: A Mixed Data Model," *e-Journal of New Media / Yeni Medya Elektronik Dergi – eJNM,* cilt 5, no. 1, pp. 12-25, January 2021.

[10] "Türkiye Cumhuriyeti Cumhurbaşkanlığı Dijital Dönüşüm Ofisi," [Çevrimiçi]. Available: https://cbddo.gov.tr/sss/yapay-zeka/. [Erişildi: 18 04 2021].

[11] "Yeni Başlayanlar için Makine Öğrenmesi Algoritmaları," 13 02 2018. [Çevrimiçi]. Available: medium.com/türkiye/yeni-başlayanlar-için-makine-öğrenmesi-algoritmaları-ae22f794af2f. [Erişildi: 15 02 2021].

[12] S. E. Şeker, "Duygu Analizi (Sentimentel Analysis)," *YBS Ansiklopedi,* cilt 3, no. 3, 2016.

[13] B. Öztürk ve M. A. Akcayol, "Fikir Madenciliği ve Duygu Analizi, Yaklaşımlar, Yöntemler Üzerine Bir Araştırma," *Selcuk Univ. J. Eng. Sci. Tech,* cilt 6, no. 4, pp. 668-693, 2018.

[14] Ö. Demir, A. Chawaı ve B. Doğan, "Sentiment Analysis Using Dictionary Based Approach in Turkish Texts," *Int. Per. of Recent Tech. in App. Eng.,* pp. 58-66, 2019.

[15] A. Uçan, "Automatic Senttiment Dictionary Translation," *Hacettepe Üniversitesi,Bilgisayar Mühendisliği Ana Bilim Dalı, Yayımlanmış Yüksek Lisans Tezi,* 2014.

[16] Ü. Can ve B. Alataş, "Duygu Analizi ve Fikir Madenciliği Algoritmalarının İncelenmesi," *Int. J. Pure Appl. Sci,* pp. 75-111, 2017.

[17] H. K. Üçükkartal, "Twitter'daki Verilere Metin Madenciliği Yöntemlerinin Uygulanması," *ESTUDAM Bilişim Dergisi,* cilt 1, no. 2, pp. 10-13, 2020.

[18] V. Tunalı ve T. T. Bilgin, "Türkçe Metinlerin Kümelenmesinde Farklı Kök Bulma Yöntemlerinin Etkisinin Araştırılması," *ELECO '2012 Elektrik - Elektronik ve Bilgisayar Mühendisliği Sempozyumu,* 2012.

[19] A. A. Karcıoğlu, "*İngilizce ve Türkçe Twitter Mesajlarının Word2vec Moldeli ile Sınıflandırılması,"* Erzurum, 2018.

[20] E. A. Yurt, "*Türkçe Metinlerde Duygu Analizi,"* İstanbul, 2015.

[21] [Çevrimiçi]. Available: http://zembereknlp.blogspot.com. [Erişildi: 11 01 2021].

[22] V. Tunalı ve T. T. Bilgin, "PRETO: A High-performance Text Mining Tool for Preprocessing Turkish Texts," *International Conference on Computer Systems and Technologies (CompSysTech),* Ruse, Bulgaria, 2012 June 22-23.

[23] Y. İşler ve A. Narin, "Diagnosis of the Patients with Congestive Heart Failure using k-Means Algorithm in WEKA Software," *SDU JOURNAL OF TECHNICAL SCIENCES,* cilt 2, no. 4, pp. 21-29, 2012.

[24] G. Aksu ve N. Doğan, "Veri madenciliğinde kullanılan bir analiz programı: WEKA," *Eğitimde ve Pskilojide Ölçme ve Değerlendirme Dergisi,* pp. 80-95, 2019 Kasım.

[25] S. A. Arpacı ve O. Kalıpsız, "A Comparison of Different Naive Bayes Technıques for Software Defect Classaification," *Omer Halisdemir University Journal of Engineering Sciences,* cilt 7, no. 1, pp. 1-13, 2018.

[26] V. Tunalı ve C. M. Tekin, "Prioritization of software development demands with text mining techniques," *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi,* pp. 615-620, 2019.

[27] N. Atasoy ve D. Tabak, "Face Recognition Application Development Using Ssupport Vector Machines," *Engineering Sciences (NWSAENS),* pp. 119-127, 2018.

[28] Z. Karakaynak, "*Destek Vektör Maikineleri ile Sınıflandırma ve Görüntü Tanıma Üzerine Bir Uygulama,"* İstanbul, Haziran 2014.

[29] B. İ. Sevindi, "k-EN Yakın Komşuluk Algoritmasının hile denetiminde kullanımı," *Muhasebe ve Vergi Uygulamaları Dergisi,* cilt 5, no. 1, pp. 21-40, 2012.

[30] M. Alan ve C. Yeşilyurt, "Farklı Veri Setleri Üzerinde SMO VE J48 Algoritmalarının Sınıflandırma Sonuçlarının Karşılaştırılması," *Sakarya Üniversitesi İşletme Bilimi Dergisi,* cilt 6, no. 3, pp. 199-213, 2018.

[31] S. Kılıç, "Klinik Karar Vermede ROC Analizi," *Journal of Mood Disorders,* cilt 3, no. 3, pp. 135-140, 2013.