T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

TASARIM LOGUNUN SINIFLANDIRMA İLE İNCELENMESİ

Onur ÜLKÜ

DANIŞMAN
Doç. Dr. Muhammed Ali AYDIN

II. DANIŞMAN
Prof. Dr. Abdül Halim ZAİM

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

İSTANBUL-2021

Bu çalışma, 17.09.2021 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Mühendisliği Programında Yüksek Lisans tezi olarak kabul edilmiştir.

Tez Jürisi

Doç. Dr. Muhammed Ali AYDIN(Danışman)
İstanbul Üniversitesi-Cerrahpaşa
Mühendislik Fakültesi

Doç. Dr. Selçuk SEVGEN
İstanbul Üniversitesi-Cerrahpaşa
Mühendislik Fakültesi

Dr. Öğr. Üyesi Zeynep TURGUT AKGÜN
İstanbul Medeniyet Üniversitesi
Mühendislik ve Doğa Bilimleri Fakültesi

Unvan Adı SOYADI
Üniversite
Fakülte

Unvan Adı SOYADI
Üniversite
Fakülte

20.04.2016 tarihli Resmi Gazete'de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi-Cerrahpaşa'nın abonesi olduğu intihal yazılım programı kullanılarak Lisansüstü Eğitim Enstitüsü'nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

ÖNSÖZ

Bilgi çağındaki gelişmelerin hızla ilerlemesi ile birlikte bilgilerin loglanmasının da önemi paralelinde artmaktadır. Gerçekleştirilen işlemlerin loglanması durumu ileri de yaşanılacak olan kritik sorunların çözümü için kolaylık sağlayacaktır. Bu sayede yapılan tüm işlemleri denetleme imkanı yakalanmış olunacaktır.

Logların giderek arttığı ve karmaşıklaştığı durumda analiz edilmesi zor olmaktadır. Bu durumu yönetmek içinse çeşitli çalışmalar yapılmaktadır. Son dönemde görülen çalışmalarda ise yapay zeka ön planda olduğu görülmektedir. Yapay zeka ile logların analiz edilmesi ve sınıflandırılması konusu ele alınacaktır.

Tez çalışması esnasında değerli destekleri için değerli hocalarım Doç. Dr. Muhammed Ali AYDIN'a ve Prof. Dr. Abdül Halim ZAİM'e teşekkür ederim.

Eğitim hayatım süresince bize eğitimin önemini, bilincini ve çalışma disiplinini kazandıran, hep yanımda olan aileme teşekkür ederim.

Tezimi babam Abdulbaki ÜLKÜ'ye armağan ediyorum.

Eylül 2021 Onur ÜLKÜ

İÇİNDEKİLER

Sayfa No

ÖNSÖZ ..... **iv**
İÇİNDEKİLER ..... **v-vi**
ŞEKİL LİSTESİ ..... **vii**
TABLO LİSTESİ ..... **viii**
SİMGE VE KISALTMA LİSTESİ ..... **ix**
ÖZET ..... **x**
SUMMARY ..... **xi**
1. GİRİŞ **1**
2. GENEL KISIMLAR **2**
2.1. LOGLAMA VE TÜRLERİ 2
2.2. LOGLAMA SİSTEMLERİNİN ÖNEMİ 2
2.3. LOG ANALİZİNDE YAPAY ZEKA 3
2.4. İLGİLİ ÇALIŞMALAR 3
3. MALZEME VE YÖNTEM **18**
3.1. TASARIM LOG YAPISI 18
3.2. VERİ ÖN İŞLEME 18
3.2.1. Noktalama İşaretlerinin Kaldırılması 19
3.2.2. Tokenize Etme 19
3.2.3. İstenmeyen Kelimelerin Çıkartılması 19
3.2.4. Kelime Kök Analizi 20
3.3. TERİM FREKANS-TERS BELGE SIKLIĞI YÖNTEMİ 20
3.4. KELİME ÇANTASI 21
3.5. ÇAPRAZ DOĞRULAMA 21
3.6. DOĞRULUK, KESİNLİK, DUYARLILIK VE F1 PUANI 22
3.7. KULLANILAN SINIFLANDIRMA ALGORİTMALAR 22
3.7.1. Doğrusal Sınıflandırıcılar: Naive Bayes Sınıflandırıcısı 22
3.7.2. Doğrusal Sınıflandırıcılar: Lojistik Regresyon 24
3.7.3. Destek Vektör Makinesi 26
3.7.4. Karar Ağacı 29
3.7.5. Rastgele Orman 31

3.7.6. Arttırma 31
3.7.7. K-En Yakın Komşu Sınıflandırıcısı 32
3.7.8. Çok Katmanlı Algılayıcı.....33
4. BULGULAR.....36
5. TARTIŞMA VE SONUÇ 44
KAYNAKLAR.....48
EKLER.....55
ÖZGEÇMİŞ.....56

ŞEKİL LİSTESİ

Sayfa No

Şekil 3.1 Anonim log yapısı 18
Şekil 3.2 Noktalama işaretlerinin kaldırılması örneği 19
Şekil 3.3 Tokenize etme işlemi 19
Şekil 3.4 İstenmeyen kelimelerin çıkartılması örneği 20
Şekil 3.5 Kelime kök analizi yapılması 20
Şekil 3.6 Naive Bayes sınıflandırıcısı akış 24
Şekil 3.7 Lojistik regresyon akış 25
Şekil 3.8 Destek vektör makineleri [58] 26
Şekil 3.9 Sert marj ile yumuşak marj karşılaştırması [58] 27
Şekil 3.10 Marj sınırlarının belirlenmesi [58] 27
Şekil 3.11 Doğrusal olmayan destek vektör makineleri [59] 28
Şekil 3.12 Destek vektör makineleri akış 29
Şekil 3.13 Karar ağaçları akış 30
Şekil 3.14 Rastgele orman akış 31
Şekil 3.15 AdaBoost akış 32
Şekil 3.16 K-En yakın komşu akış 33
Şekil 3.17 Çok katmanlı yapay sinir ağının genel hali 34
Şekil 3.18 Çok katmanlı algılayıcı akış 35
Şekil 4.1 Karmaşıklık matrisi 37
Şekil 5.1 Lojistik regresyon sonuç 45

TABLO LİSTESİ

Sayfa No

Tablo 4.1 Algoritma doğruluk sonuçları....36
Tablo 4.2 Çapraz doğrulama sonuçları....36
Tablo 4.3 Navie Bayes- Kelime çantası sonuçları....38
Tablo 4.4 Navie Bayes- Terim frekans-Ters belge sıklığı yöntemi sonuçları....38
Tablo 4.5 Lojistik Regresyon - Kelime çantası sonuçları....38
Tablo 4.6 Lojistik Regresyon - Terim frekans-Ters belge sıklığı yöntemi sonuçları....39
Tablo 4.7 Destek Vektör Makinesi - Kelime çantası sonuçları....39
Tablo 4.8 Destek Vektör Makinesi - Terim frekans-Ters belge sıklığı yöntemi sonuçları ...39
Tablo 4.9 Karar Ağacı - Kelime çantası sonuçları....40
Tablo 4.10 Karar Ağacı - Terim frekans-Ters belge sıklığı yöntemi sonuçları....40
Tablo 4.11 Rastgele Orman - Kelime çantası sonuçları....41
Tablo 4.12 Rastgele Orman - Terim frekans-Ters belge sıklığı yöntemi sonuçları....41
Tablo 4.13 Arttırma - Kelime çantası sonuçları....41
Tablo 4.14 Arttırma - Terim frekans-Ters belge sıklığı yöntemi sonuçları....42
Tablo 4.15 K-En Yakın Komşu - Kelime çantası sonuçları....42
Tablo 4.16 K-En Yakın Komşu - Terim frekans-Ters belge sıklığı yöntemi sonuçları....42
Tablo 4.17 Çok Katmanlı Algılayıcı - Kelime çantası sonuçları....43
Tablo 4.18 Çok Katmanlı Algılayıcı - Terim frekans-Ters belge sıklığı yöntemi sonucu 43
Tablo 5.1 Lojistik Regresyon sonuç....46

SİMGE VE KISALTMA LİSTESİ

Kısaltmalar **Açıklama**

TF-IDF : Term Frequency-Inverse Document Frequency (Terim Frekans-Ters Belge Sıklığı Yöntemi)
MLP : Multilayer Perceptron (Çok Katmanlı Algılayıcı)
SVM : Support Vektor Machine (Destek Vektör Makinesi)
CNN : Convolution Neural Network (Evrişimli sinir ağları)
KNN : K-Nearest Neighbors (K-En Yakın Komşuları)
NN : Nearest Neighbors (Yakın Komşular)
NB : Naive Bayes
DVM : Destek Vektör Makineleri
RBF : Radial Basis Function (Radyal Temel Fonksiyonu)
IDS : Intrusion Detection System (İzinsiz Giriş Tespit Sistemi)

ÖZET

TASARIM LOGUNUN SINIFLANDIRMA ile İNCELENMESİ

YÜKSEK LİSANS TEZİ

Onur ÜLKÜ

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman : Doç. Dr. Muhammed Ali AYDIN

II. Danışman : Prof. Dr. Abdül Halim ZAİM

Yazılım sistem yöneticileri, sistem davranışını anlamak, anormallikleri izlemek, yazılım hatalarını izlemek ve arıza tespiti için log verilerine bağımlıdır. Makine öğrenimi tekniklerine dayalı günlük analizi, ham günlüklerin DevOps ekibinin ve yöneticilerin sorunları çözmesine yardımcı olan anlamlı bilgilere dönüştürülmesini sağlar. Yapay zeka, benzer günlükleri bir arada gruplandırmayı sağlar ve periyodik günlükleri daha düzenli ve sıralı tutarak bakmamız gereken yere daha hızlı ulaşmamızı sağlar. Bu yazıda, VoIP (İnternet Protokolü Üzerinden Ses) soft-switch ürünü tarafından üretilen log verileri üzerinde bir log sınıflandırma sistemi sunuyoruz. Bu sayede sorunu tespit etmeyi, ilgili birime yönlendirmeyi, kaynak tahsisini, yazılım hatalarını daha hızlı ve verimli bir şekilde çözmeyi hedefledik. Log sınıflandırması için Destek Vektör Makineleri, Doğrusal Sınıflandırıcılar, Karar Ağacı, Rastgele Orman, Arttırma, K-En Yakın Komşuları ve Çok Katmanlı Algılayıcı gibi makine öğrenme algoritmaları kullanılmaktadır.

Eylül 2021, 55 sayfa.

Anahtar kelimeler: Log Analizi, TF-IDF, Kelime Çantası, Lojistik Regresyon

SUMMARY

EXAMINATION OF DESIGN LOG with CLASSIFICATION

M.Sc. THESIS

Onur ÜLKÜ

Istanbul University-Cerrahpasa

Institute of Graduate Studies

Department of Computer Engineering

Supervisor : Assoc. Prof. Dr. Muhammed Ali AYDIN

Co-Supervisor : Prof. Dr. Abdul Halim ZAİM

Software system admins depend on log data for understanding system behavior, monitoring anomalies, tracking software bugs, and malfunctioning detection. Log analysis based on machine learning techniques enables to transform of raw logs into meaningful information that helps the DevOps team and administrators to solve problems. AI ensures to group similar logs together and keeps periodic logs more organized and sorted, allowing us to get to where we need to look faster. In this paper, we present a log classification system on log data generated by VoIP (Voice over Internet Protocol) soft-switch product. In this way, we targeted to detect the problem, direct it to the relevant department, allocate resources, and solve software bugs faster and more efficiently. Machine learning algorithms such as Support Vector Machines, Linear Classifiers, Decision Tree, Random Forest, Boosting, K-Nearest Neighbors, and Multilayer Perceptron are used for log classification.

September 2021, 55 pages.

Keywords: Log Analytics, TF-IDF, Bag of Words, Logistic Regression

1. GİRİŞ

Gelişen teknoloji ile beraber bilgi alışverişi de beraberinde artmaktadır. Bilgilerin ise kayıt altında tutulmasının önemi ise her geçen gün artmaktadır. Bunun içinse loglama işlemi gerçekleştirilmelidir. Sistemlerin ya da cihazların yaptığı işlemleri tutan veya bu sistemleri, cihazları kullanan kullanıcıların işlemlerinin tutulmasına loglama denir [1]. Loglama ile tek bir yerden gelen kayıtlar tutulmamakta olup birden çok farklı yerden gelen kayıtlar bir arada tutulmaktadır. Günümüzdeki kurumlar logları sadece kayıt altına almaktadır ve analizlerini yapmamaktadır. Logların analizinin ve sınıflandırmasının yapılması gerekliliğinin yanı sıra doğru bir şekilde yapılması önemlidir. Karmaşıklaşan logların analizi ve sınıflandırması ise zordur. Kurumlarda log yönetiminin önemi ve gerekliliği dikkat edilmesi gereken en önemli unsurlardan biridir.

Tez kapsamında ise bir şirket ile anlaşılmıştır ve bu şirket tarafından iletilen tasarım logların sınıflandırılması için çalışmada bulunulmuştur. Tasarım logları şirketteki DevOps, operasyon, bakım personelleri ve tasarım ekipleri tarafından incelenmektedir. Tasarım log sistemi, telekom anahtarında çalışan uygulamalar ve işlemler için sorunlar ve durum ile ilgili bilgilendirici verileri içerir. Bu loglar, sistem üzerinde yazılım uygulamaları çalıştırılarak oluşturulur. Genel bir tasarım logu örneği, zaman damgası, donanım konumu, işlem adı, uygulama adı, uygulama dizini ve uygulama hakkında bilgiler içeren bilgiler içerir. Tasarım logları sınıflandırma yapılırken; error, debug veya info şeklinde 3 farklı şekilde sınıflandırılmaktadır. Logları bu şekilde etiketlemek, organize etmek ve bunları farklı kategoriler halinde düzenlemek ileri verileri filtrelemeyi de kolaylaştıracaktır. Bu sayede sorunun daha hızlı bir şekilde adreslenerek ilgili departmanlara yönlendirilmesi sürecini hızlandıracaktır. Sınıflandırma yaparken de makine öğrenmesi algoritmalarından (Destek Vektör Makineleri, Doğrusal Sınıflandırıcılar, Karar Ağaçları, Rastgele Orman, Arttırma, K-En Yakın Komşu ve Çok Katmanlı Algılayıcı) faydalanılmıştır.

2. GENEL KISIMLAR

2.1.LOGLAMA VE TÜRLERİ

Log kelime anlamı olarak kayıt demektir. Bilişim sistemlerinin çeşitli amaçlar için (hata giderme, güvenlik, performans, denetim bulguları vs.) ürettiği kayıt bilgilerinin tutulmasına loglama denir [1]. Gelişen teknoloji ile birlikte kullanıcıların yapmış olduğu işlemler veya sistemin kendisi tarafından yapılan işlemler kayıt altında tutulmaktadır. İnternetin yoğun kullanılması ve teknoloji ile beraber dijital ortamdaki gerçekleşen faaliyetlerin artması ise yapılan işlemlerin loglanmasını zorunlu hale getirmektedir.

Logların bir çok türü vardır. Bazı log türlerinden bahsetmek gerekirse; veritabanındaki değişikliklerin kaydı için işlem logları, Windows sistemler üzerinde gerçekleşen oturum açma, uygulama hataları gibi farklı tür de olayları kaydetmek için olay logları ve sistemin günlük loglarını toplayabilmek için Linux tabanlı sistemlerde sistem logları tutulmaktadır [2]. Tez kapsamında incelenen log ise bir şirket tarafından sağlanan ve anonimleştirilen loglardır. Şirket logu, şirketin uygulamaları ile birlikte işlemler sonucunda oluşan sorunlar ve durum ile ilgili bilgilendirici verileri içermektedir.

2.2.LOGLAMA SİSTEMLERİNİN ÖNEMİ

Sisteme yapılan giriş ve çıkışların tutulması, sistemde yapılan işlemlerin tutulması, hangi kullanıcılar tarafından işlemlerin yapıldığı bilgisinin kayıt altına alınması ileri de yaşanılabilecek herhangi bir güvenlik sorunu olduğunda ya da farklı bir sebepten dolayı incelenmek istenildiğinde loglamanın önemi ortaya çıkmaktadır. Sadece oluşan bir sorunun çözümü için log incelenmesi yapılmamaktadır. İleri de oluşabilecek sorunlarında önceden tespit edilebilmesi için de tutulan loglar üzerinden analiz yapılabilir ve oluşabilecek sorunlar öncesinde tespit edilebilir.

Tez kapsamında yapmış olduğumuz çalışmada giderek artan şirket loglarının ayrıştırılması zor olmaktadır ve logların sınıflandırılması için yapay zeka yöntemleri kullanılarak çalışılmıştır.

2.3.LOG ANALİZİNDE YAPAY ZEKA

Günümüzde hızla ilerleyen teknoloji ile beraber veri çeşitliğinde ve hacminde de büyük oranda artık gözlenmektedir. Verilerin analiz edilmesinde ise yapay zeka ve makine öğrenmesi teknikleri kullanımlarına başvurulmaktadır. Tez kapsamında ise bu tekniklerden biri olan sınıflandırma tekniği detaylı olarak ele alınması hedeflenmektedir. Sınıflandırma ile bir logun hangi sınıfa ait olduğunu yani error, debug veya info sınıflarından hangisinin kapsamında olduğunun tahmin edilmesi sağlanmıştır. Sınıflandırma, veri setindeki ayırt etmemize yarayan belirli özelliklere bakarak hedefimizi kategorilere ayırmamızı sağlayacaktır.

Şirket loglarının sınıflandırılması için Lojistik Regresyon, Destek Vektör Makinesi, Naive Bayes, Karar Ağacı, Arttırma, Rastgele Orman, K-En Yakın Komşu ve Çok Katmanlı Algılayıcı yapay zeka algoritmaları kullanılmıştır.

2.4.İLGİLİ ÇALIŞMALAR

Logların sınıflandırılması ile ilgili çeşitli güncel literatür araştırmaları yapılmıştır ve bu bölümde tez ile ilgili yapılan güncel araştırmalara yer verilmiştir. Bu çalışmalardan bazıları aşağıdaki şekilde açıklanmıştır.

Verileri sınıflandırmak için yapay sinir ağları ve lojistik regresyon yöntemleri karşılaştırılmıştır. Karşılaştırma sonunda ise yapay sinir ağlarının lojistik regresyona göre verileri sınıflandırırken daha etkin olduğu görülmüştür. Çalışma da veri seti kümesi, Yüzüncü Yıl Üniversitesi'nde okuyan beden eğitimi ögretmenliği bölümü için özel yetenek ile ilgili sınava giren 467 tane erkekten oluşturulmuştur. Çalışma 2005 ve 2006 tarihleri arasında yapılmıştır. Analiz sürecinde ise Lojistik Regresyon için Minitap, yapay sinir ağları için R istatistiksel analiz programları kullanılmıştır [3].

Rastgele ormanlar yönteminin kullandığı çalışmada bir rehber turu üzerine incelemelerde bulunulmuştur. Bu çalışma da teorik ve metodolojik gelişmeler gözden geçirilmiştir. Parametrelerin seçimine, yeniden örnekleme mekanizmasına ve değişken önem ölçülerine özel önem verilerek, algoritmayı yönlendiren matematiksel kuvvetlere vurgu yapılmıştır. Ayrıca bu inceleme ile uzman olmayanların ana fikirlere kolayca erişim sağlaması amaçlanmaktadır [4].

En yakın komşu tekniklerinin incelendiği bir çalışmada en yakın komşu tekniğinin örüntü tanıma, metin kategorizasyonu, nesne tanıma gibi alanlarda yüksek verimli ve etkili olduğu savunulmuştur. Avantajlarının yanı sıra dezavantajları da açıklanmıştır. Çalışmada da birçok teknik açıklanmıştır. Ağırlıklı KNN, Model Bazlı KNN, Yoğunlaştırılmış NN, İndirgenmiş NN, Genelleştirilmiş NN yapısı ile ilgili konulara yer verilmiştir. K-D Ağacı, En Yakın Özellik Çizgisi, Ayarlanabilir NN, Ortogonal Arama ağacı KNN temelinde geliştirilen yapı tabanlı algoritmalar da anlatılmaktadır [5].

5 farklı kategorinin olduğu 2 çeşit haber kaynağı üzerine yapılan araştırmada, vektör olarak azaltılmış özellik vektörü kullanılmıştır. Ayrıca Destek Vektör Makineleri, Naive Bayes, C4.5 ve Rastgele Orman sınıflandırma metotları uygulanarak elde edilen sonuçları karşılaştırmıştır. Genellikle daha yüksek olmakla birlikte en yüksek başarı sonucu %92,73 başarı sonucu ile Naive Bayes algoritması sayesinde sağlanmıştır [6].

Yapay sinir ağları ve yapay zekanın gelişmiş uygulamaları üzerinde bir incelemede bulunulmuştur. Verilerin yorumlanma yeteneğini geliştirmek için sinir ağlarının bulanık mantık gibi diğer hesaplama yöntemleri incelenmiştir. Sinir ağlarının problem çözme için en yaygın olarak kullanıldığı en genel uygulamalar örüntü tanıma, veri analizi, kontrol ve kümelemedir. Yapay sinir ağları, yüksek işlem hızları ve bir problemin çözümünü bir dizi örnekten öğrenme yeteneği gibi birçok özelliğe sahiptir. Bu makalede yapay sinir ağları ve yapay zekanın farklı alanlarda oynadığı kritik roller tartışılmaktır [6].

Başka çalışmada veri seti olarak sosyal ağdaki 12 gruba ayrılan haber metinleri sınıflandırılmıştır. Haber kaynakları üzerinde yöntem olarak kelime çantası yöntemi kullanılmıştır. Bununla birlikte vektörize edilmiştir. Destek vektör makineleri algoritması uygulanarak %75 başarı değerinin üstünde değer elde edilmiştir [6].

Endonezya'daki 15 farklı kategoride haber makalelerinden oluşan veri setini ele alarak bu haber makalelerini sınıflandırılma ile ilgili çalışmada terim frekans-ters belge sıklığı yöntemi ile %98,3 başarı oranı değeri elde edilmiştir [6].

Hindistan'da yapılan çalışmada Naive Bayes algoritması kullanılarak üniversite verileri üzerinde deney gerçekleştirilmiştir. Deney kapsamında kullanıcıların sistemi kullanırken arama tarama süresini azaltarak üniversitede kullanılan sistemin verimliliğinin ve etkinliğinin arttırılması hedeflenmiştir. %80 oranında doğruluk sonucu vermesinin yanı sıra Naive Bayes

sınıflandırıcısının tanımsız sınıfları sınıflandıramaması da bir dezavantaj olarak görülmüştür [7].

Başka çalışmada karar ağaçları incelenmiştir. Ek olarak daha çok tahmin ve sınıflandırmada kullanılan veri madenciliği yaklaşımları incelenmiştir. Anket düzenlenmiştir. Anketin içeriği ise internetin ve bilgisayarın güvenliği üzerine olan konulardan oluşmaktadır. Çeşitli demografik niteliklerindeki bireyler için karar ağaçları yöntemi uygulanarak çıkarımlar yapılması istenmiştir. Anket, toplamda rassal olarak seçilen 300 kişiye uygulanmıştır. Çalışma esnasında program olarak SPSS Clementine (Modeler) tercih edilmiştir. Bu programın tercih edilme sebeplerinin başında ise istatistik ile ilgili algoritmalarda görsel bir ara yüz sunmasıdır [8].

Ağ saldırı tespit için rastgele orman modellemesi kullanılması ile ilgili çalışma yapılmıştır. İzinsiz giriş tespiti, gelişmekte olan bir araştırma alanı haline gelmektedir. İzinsiz Giriş Tespit Sistemi (IDS), kullanıcıların faaliyetlerini normal veya anormal olarak tanımlamaya ve bildirmeye çalışır. IDS, doğrusal olmayan ve karmaşık bir problemdir. Ayrıca ağ trafiği verileri ile de ilgilenir. Birçok IDS yöntemi önerilmiştir ve farklı doğruluk seviyeleri üretmektedir. Bu nedenle etkili saldırı tespit sisteminin geliştirilmesi gereklidir. Çalışmada rastgele orman sınıflandırıcısı kullanarak saldırı tespit sistemi için bir model oluşturulmuştur. Rastgele orman bir topluluk sınıflandırıcıdır ve saldırıların etkili bir şekilde sınıflandırılması için diğer geleneksel sınıflandırıcılara kıyasla iyi performans gösterir. Modelin performansını değerlendirmek için NSL-KDD veri seti üzerinde deneyler yapılmıştır. Önerilen modelin düşük yanlış alarm oranı ve yüksek algılama oranı ile verimli olduğunu göstermektedir [9].

Metin inceleyip sınıflandırmak için Naive Bayes, rastgele orman, karar ağacı yöntemleri denenmiştir. Ek olarak destek vektör makineleri ile lojistik regresyon sınıflandırıcıları kullanarak karşılaştırılmıştır. Makalenin odak noktası, eğitim veri setlerinin boyutuna ve n-gram sayısına dayalı olarak sınıflandırma doğruluğunu değerlendirerek bu sınıflandırıcıları karşılaştırmaktır. Deneylerde, Amazon'dan ürün inceleme verileri için kısa metinler analiz edilmiştir. Tüm sınıflandırma yöntemleri için sınıf başına 5000 incelemek için eğitim seti boyutunun yeterli olduğu belirtilmiştir. Ürün incelemeleri için lojistik regresyon çok sınıflı sınıflandırma yönteminin, Naive Bayes, rastgele orman, karar ağacı ve destek vektör makineleri sınıflandırma yöntemlerine kıyasla en yüksek sınıflandırma doğruluğuna ulaştığını göstermiştir [10].

Aynı olmayan kategorilere ait Türkçe metinleri üzerinde Word2Vec algoritması uygulayıp sınıflandırılmıştır. Çalışma kapsamında ilk olarak metinlerde kelime vektörleri bulunmuştur ve dokümanda bulunan sözcüklerin vektör ortalaması alınmıştır. Destek vektör makineleri algoritması uygulanılarak sınıflandırma işlemi gerçekleşmiştir. Word2Vec yöntemi ile beraber TF-IDF'inde olduğu bir sınıflandırma işlemleri yapılmıştır. Sonuçlar karşılaştırıldığında diğerlerine göre çok iyi sonuç elde edildiği belirtilmiştir [11].

Metin tanıma, yüz tanıma ve parmak izi tanıma gibi problemlerin çözümlerinde çoğunlukla yapay sinir ağları kullanılması konusundan yola çıkarak bir çalışmada bulunulmuştur. Metin türü belirleme işlemi için seçilen veri kümesi üzerinde öncelikle k-en yakın komşu algoritması uygulanmıştır. Sonra ise elde edilen hatalı durumlar yapay sinir ağları ile tekrar ele alınarak birlikte çalışma tekniği uygulanmıştır. Böylelikle başarımı artıracak bir çözüm önerisi sunulmaya çalışılmıştır. Veri seti olarak 3 ayrı konu incelenmiştir. Konu içeriklerinden ilki akışkanlar dinamiğidir. Diğer konular ise bilimsel endeksler ile tıbbi konulardır. Yaklaşık 3900 tane çalışmayı bulunduran bu veri seti üzerinde detaylı olarak çalışılmıştır. Çalışma sonucunda problem çözerken sadece bir algoritma ile problemler ele alınmayıp bunun ile birlikte farklı bir teknik daha uygulanırsa başarı oranının daha da yüksek olacağı ortaya konulmuştur [12].

Teknolojinin hızla ilerlemesi ile birlikte kontrol edilemeyecek miktarda veriler hızlıca üretildiği için ve kontrolü de paralelinde zorlaştığı için yapay zeka çalışmaları üzerine ayrı bir çalışmada bulunulmuşlardır. Sınıflandırma ve kümeleme hakkında çeşitli makalelerde oldukça fazladır. Ayrıca web madenciliği ve yapay sinir ağları hakkında da bir çok makale vardır. Yapay zeka ile makine öğrenmesi yöntemleri detaylı olarak anlatılmaktadır [13].

Dağınık küme sistemlerinde günlük logların sınıflandırılması çalışmasında bulunulmuştur ve sınıflandırma algoritmaları kullanılmıştır. Çalışmalarının sonucunda evrişimli sinir ağlarına dayalı sınıflandırma sistemi modeli tercih edip bulut sistemlerdeki loglar sınıflandırılmıştır [14].

Türkçe için karşılaştırmalı olarak metin sınıflandırma analizi yapılan bir çalışmada bulunulmuştur. Çalışmada geleneksel yaklaşım olarak öne sürülen kelime çantası yöntemi ve yapay sinir ağı güncel yaklaşımlar ile beraber metinlerin sınıflandırılması yönünden

karşılaştırılmış ve bunun sonucunda geleneksel yöntemin kelime gömme anlamına gelen word embedding yöntemi ile kıyaslanacak düzeyde olduğu vurgulanmıştır [15].

İşletme veri tabanındaki müşteri yorumlarını şikayet, talep ve teşekkür olmak üzere sınıflandırmak üzerine karar ağacı modeli oluşturulmuştur. Çok fazla sayıdaki verinin işletme tarafından manuel olarak sınıflandırılması zor olacağı için otomatik olarak sınıflandırma yöntemi tercih edilmiştir. Müşteri yorumları kullanılmıştır ve çeşitli etiketler oluşturulmuştur. Müşteri temsil eden özellikteki kelimeler belirlenmiş ve bu kelimeler birbirlerine düğümler ile bağlantı kurarak tespit etme sağlanmıştır. Araştırma kapsamında 12 tane müşterinin yorumu değerlendirilmiştir. Yorum tipleri de 3 ayrı şekilde 4'er 4'er bölünmüştür. Yorum tipleri sırasıyla; şikayet, talep ve teşekkür tipleridir. Örnekler seçilirken kullanılan örnekleme yöntemi kararsal örneklemedir. Ayrıca yazım denetimi otomatik olmadan yapılmıştır [16].

İletişim araçlarımızdan birisi olan e-posta hayatımızı kolaylaştırdığı gibi amacı dışında gelen e-postalar hem alıcının zamanını çalmaktadır hem de maddi ve manevi olarak zararlar verebilmektedir diye düşünüp istenmeyen e-postaların sınıflandırılması için makine öğrenmesi yöntemi ile beraber kelime kümesi teknikleri kullanılarak çalışmada bulunulmuştur. Çalışma sonucunda Naive Bayes yaklaşık %99 değer ile istenmeyen e-posta sınıflandırılmasında yüksek orana sahip algoritma olmuştur. Ancak karar ağaçlarının bu konuda Naive Bayes kadar iyi olmadığı görülmüştür [17].

Saldırı tespit sistemleri ve izinsiz giriş önleme sistemleri ile birlikte sistemlerin güvenliğini sağlanmaya yönelik bir çalışmada bulunulmuştur. İzinsiz giriş tespitinde destek vektör makinesi, rastgele orman algoritmaları ve aşırı öğrenme makine teknikleri kullanılmıştır. Çalışmada NSL veri seti kullanılmıştır. Sonuçlar karşılaştırıldığında ise aşırı öğrenme makine tekniği daha iyi performans göstermiştir [18].

E-ticaret sitesi kullanıcılarının müşteri anlaşmasına uymamasını tahmin etmek için yapay zeka sistemleri kullanılmıştır. E-ticarette ödemede temerrüde düşmeyi önlemek için artan risk yönetimine yol açmaktadır. Ödemede temerrüt ifadesinin anlamı ise bir müşterinin faturayı aldıktan sonra 90 gün içinde ödememesidir. Müşterilerin temerrüt olasılığını belirlemek için kredi puanlaması sistemi kullanılır. İncelenen çalışmada temel amacı, müşterilerin temerrütlerini tahmin etmek için en çok kullanılan sistemlerden biri olan e-ticaret risk yönetim sistemi, risk çözüm hizmetlerinin risk öncesi kontrolünün yerini alacak bir kredi

puanlaması modeli geliştirmektir. Ön risk kontrolü, sipariş sürecinden gelen verileri kullanır ve hariç tutma kurallarını ve genel bir kredi puanlaması modelini içerir. Yeni modelin tüm ön risk kontrolünün yerini alması ve risk çözüm hizmetleri ana risk kontrolü ile hem izole hem de entegre olarak çalışması gerekiyor. Bu çalışmada genetik programlamanın kredi puanlama üzerine uygulaması incelenmiştir. Model, tanınmış bir Alman finansal çözümler şirketi tarafından sağlanan gerçek dünya veri kümesi üzerinde geliştirilmiştir. Veri kümesi, risk çözüm hizmetleri tarafından işlenen sipariş isteklerini içerir. Sonuçlar, genetik programlamanın hem sınıflandırma doğruluğu hem de kar açısından risk öncesi kontrolün genel kredi puanlaması modelinden daha iyi performans gösterdiğini göstermektedir. Genetik programlama, lojistik regresyon, destek vektör makineleri ve güçlendirilmiş ağaçlar gibi birkaç son teknoloji makine öğrenimi yöntemiyle rekabetçi sınıflandırma doğruluğu elde etmiştir. Ayrıca, genetik programlama modeli, daha da yüksek ayrım gücü olan bir model oluşturmak için risk çözüm hizmetleri ana risk kontrolü ile birlikte kullanılabilir [19].

Bir çalışmada sinir-kas sistemleri hastalık çeşidi olan Nöromüsküler hastalıkları yapay zeka yöntemleri ile sınıflandırılmıştır. EMG (Elektromiyogram) sinyalleri önce ön işleme tabi tutulup sonrasında sınıflandırılma işlemi gerçekleştirilmiştir. Daha sonra ise bölme, küme haline getirmek ile birlikte özellik çıkartma adımları uygulanmıştır. Küme haline getirirken hibrit yöntem seçilmiştir. Ayrıca frekans ile beraber zaman konularındaki özellik vektörleri ve bunların çoklu özellik vektörlerinin çeşitli kombinasyonları (toplam 25 özellik vektörü) kullanılmıştır. Diğer adımda da veriler üzerinde diskriminant analizi algoritmaları ve destek vektör makineleri yöntemleri uygulanmıştır. Ek olarak k-en yakın komşu algoritması da kullanılarak sınıflandırma yapılmıştır. Doğrulama olarak çapraz doğrulama yönteminin bir ölçüsü olarak kullanılmıştır ve 10'un k-değerinin çapraz doğrulaması tercih edilmiştir. Deneyin sonucunda, öznitelik vektörlerinin, tek öznitelik vektörün çoklu öznitelik vektörlerinden başarısız olduğu gözlenmiştir. Tekli olmayan yani çoklu öznitelik vektörleri kullanıldığında; destek vektör makineleri, EMG sinyallerini k-en yakın komşuları yöntemine kıyasla yüksek doğruluk elde edilmiştir. Sonuç olarak Amyotrofik Lateral Skleroz hastalığı için %97.39 ve miyojenik için de yaklaşık %87 doğruluk elde edilmiştir. Araştırma sonucunda çıkartılan sonuç ise; sınıflandırmanın performansı, sınıflar arasındaki ayrılabilirliğinin öznitelik vektörlerine bağlıdır [20].

Patlama kaynaklı hava aşırı basıncının tahmin edilmesi için yapay sinir ağlarına ve rastgele ormana dayalı sağlam bir yapay zeka sistemi tasarlanmıştır. Açık ocak madenlerinde kaya parçalanması için en popüler yöntem patlamadır. Yapay sinir ağları (YSA) ve rastgele orman (RF), kod adı ANNs-RF'ye dayalı patlama kaynaklı AOp(hava açık basıncı)'yi tahmin etmek için sağlam bir yapay zeka sistemi geliştirilmiştir. Önce 5 tane yapay sinir ağları modeli geliştirilmiştir; daha sonra bunları birleştirmek için rasgale orman algoritması kullanılmıştır. Vietnam'daki Nui Beo açık ocak kömür madeninde 114 patlatma olayı kaydedilmiştir. AOp'yi tahmin etmek için girdi değişkenleri olarak maksimum patlayıcı şarj kapasitesi, izleme mesafesi, dikey mesafe, barut faktörü, yük, boşluk ve gövde uzunluğu kullanılmıştır [21].

Türkçe metinler üzerinde yine bir çalışmada bulunulmuştur ve Doc2Vec algoritması uygulanmıştır. Metinleri sınıflandırırken destek vektör makineleri ile birlikte k-en yakın komşu kullanılmıştır. Bu sınıflandırıcılar üzerinde ise Doc2Vec ve TF-IDF uygulanıp, elde edilen sonuçlar karşılaştırılmıştır [22].

Köşe yazarlarının yazılarını, yaklaşık 2000 veri, analiz edilip yazarı tahmin etmek amaçlanmıştır. DeepLearning4J kütüphanesi kullanılmıştır. PV-DBOW ile PV-DM mimarileri Doc2Vec algoritmasının mimarilerindendir ve çalışma kapsamında kullanılmıştır. Çalışma sonucunda bazı yazarların diğer yazarlara göre farklı olduğunu bulmak için bu yöntemlerin kullanılabileceği görülmüştür. Yaklaşım ile yazarların profillerini belirleme ile birlikte intihallerin tespitinde ve ayrıca yazar kalemlerinin birbiri ile ilişkili olduğunu bulma gibi amaçlar doğrultusunda kullanılabileceğini belirtmişlerdir [23].

Bir ön işleme tekniği ile birleştirilen dinamik seçimin, dengesiz sınıflandırma problemleri için statik topluluktan daha yüksek performans elde edebileceği argumanını deneysel olarak değerlendiren bir çalışma yapılmıştır. Değişen derecelerde sınıf dengesizliğine sahip 84 adet iki sınıflı ve 26 adet çok sınıflı veri seti toplanmıştır. Dinamik topluluğun, statik topluluğa kıyasla F-ölçümünü ve G-ortalamasını iyileştirdiğini göstermektedir. Ayrıca, farklı dengesizlik seviyeleri göz önüne alındığında, dinamik seçim yöntemleri diğer alternatiflere göre daha üst sıralarda yer aldığı gözlenmiştir [24].

Bir çalışmada veri seti olarak 2 tane online haber kaynağından çokça miktarda metinler toplanmıştır. Araştırmada bir yapay sinir ağı kullanılmıştır. Bu sinir ağı kelime vektörleri ile birlikte doküman sınıflandırmasını yapabilmektedir. Yapay sinir ağı kullanılarak 4 kategoriye

ayrılmış olan bir haber kaynağı için yaklaşık %88 başarı oranı elde edilmiş iken diğer haber kaynağı için %74,31 başarı sonucu elde edilmiştir [25].

Gerçek olmayan web sitelerini sınıflandırıp bu siteleri tespit etmek için bir araştırma makalesi yayınlanmıştır. Sınıflandırmada rastgele orman, k-en yakın komşu, Naive Bayes ve destek vektör makineleri kullanmışlardır. Detaylı olarak sonuçlar karşılaştırılıp başarı oranı en yüksek algoritmanın bulunması istenilmiştir. Algoritmalar içerisinde sonuçlar detaylı karşılaştırılması sonucunda yaklaşık %95 oranla rastgele orman algoritması seçilmiştir [26].

Bir araştırma makalesinde bir online ticaret sitesi üzerinden elde edilen veriler üzerinde sınıflandırma yöntemleri kullanarak çıkan sonuçların doğruluk oranları karşılaştırılmıştır. Bunun sonucunda ise destek vektör makineleri ve çok katmanlı algılayıcı algoritmalarının yaklaşık %97 başarı oranı ile diğer kullanılan algoritmaların sonuçlarından daha iyi olduğu görülmüştür [27].

Türkçe haber metinlerini sınıflandırırken evrişimli sinir ağları ve Word2Vec metodu kullanılmıştır. Türkçe Metin Sınıflandırma olan TTC-3600 veri seti ile metin sınıflandırma ile ilgili çalışma yapılmıştır. Çalışmada Zemberek yazılımı kullanılmıştır. Bununla birlikte düzenlenmiş veri seti ile birden çok evrişimli sinir ağları eğitilmiş ve testleri gerçekleştirilmiştir. Evrişimli sinir ağları ile birlikte Word2Vec metodu, diğer makine öğrenmesi sınıflandırma algoritmalarından daha çok performans olup doğruluk değeri yaklaşık %93 olarak belirlenmiştir. Doğal dil işleme üzerine yapılan çalışmalar az sayıdadır. Bu yüzden nitelik çıkartma yollarının sınırlı olması sebebiyle, sözcüklerin semantik değerlerinin öncesinde eğitilmiş olan Word2Vec ağı ile beraber sınıflandırmaya dahil olması sayesinde evrişimli sinir ağları modelinin başarı oranını yükseltmiştir [28].

Başka çalışmada da Türkçe metin sınıflandırma çalışmasında bulunulmuştur. Bu çalışmada veri kümesi Türkçedir. Yazılar ise etiketsizdir. Öncelikle veri kümesi ile beraber kelime vektörü kullanılmıştır. Kelime vektörleri farklı bir veri seti üzerine aktarılmıştır. Bu veri seti ise farklı haber siteleri kaynaklarından oluşmaktadır ve transfer öğrenme uygulanmıştır. Veri kümesi üzerinde çeşitli sınıflandırma algoritmaları uygulanmıştır. Bu algoritmalar: Naive Bayes ve destek vektör makineleri. Araştırmada kelime vektörlerinin transfer öğrenme ile beraber aktarımı incelenmiştir. Bununla birlikte doğruluk değerleri değişimleri ve makine öğrenmesi yöntemlerine etkileri analiz edilerek gözlenmiştir. Deney sonucunda transfer

öğrenme yönteminin başarı sonucunu arttırdığı ve destek vektör makinelerinde iyi sonuç verdiği görülmüştür. Araştırmada toplam iki ayrı Türkçe veri kümesi oluşturulduğuna değinilmiştir. İlk veri seti kelime vektörlerinin öğretilmesi amacı ile beraber Türkçe diliyle oluşturulan etiketsiz metinler kullanılarak derlem oluşturulması sağlanmıştır. Ayrıca bu derlem Wikipedia ortamındaki bulunan Türkçe araştırma makalelerinden seçilerek oluşmuştur. Diğer veri kümesinde farklı gazete sitelerinde olan yazılar seçilmiştir. 4 farklı kategorideki (ekonomi, siyaset, teknoloji, spor) haber metinleri alınarak veri kümesi elde edilmiştir. Ek olarak kelime çantası ile Word2Vec yöntemi karşılaştırılmıştır [29].

Bölgedeki afetlerin önüne geçilmesi ve bölgedeki su kaynaklarının tasarruflu bir şekilde kullanmak için yağış ve akış verilerinin ilişkisinin tahmin edilmesinde iki yöntem kullanılmıştır. Bu yöntemlerden ilki destek vektör makineleridir. Diğer ise M5 karar ağacı yöntemi kullanılmıştır. Çalışmada ABD'de bulunan bir nehirde yaklaşık 730 günlük sıcaklık, yağış ve akış gibi bilgiler ile modeller oluşturulmuştur. Tahminleme de kullanılan iki yöntem arasından en iyi sonuç verenin M5 karar ağaçları olduğu görülmüştür [30].

Elektronik belge yönetim sistemi olan EBYS sisteminde çok fazla sayıdaki kullanıcıların olması ile birlikte bu kullanıcılar üzerindeki davranışsal verileri toplanmıştır. Veriler üzerinde ise log analizi yapılmıştır. Veri kümesi yaklaşık yüz bin satır kayıttan oluşmaktadır. Pandas, Plotly ve Scikit-Learn kütüphaneleri kullanılmıştır. Çalışmada veriler anonim hale getirilmiştir. Deney sonucuna göre kayıtların elektronik belge yönetim sistemi açısından işlem ile birlikte aksiyonları hakkında fikir verilmesi konusundaki olumlu etkileri görülmüştür. Ayrıca kurumsal planlamada da stratejik yapılar için etkili olduğu gözlenmiştir. İstatistik bilgiler ile kayıt dosyalarının, şirketlerin stratejik planlar oluştururken karar almasında şirketlere büyük ölçüde destek olacağı tahmin edilmektedir [31].

Twitter'da Türkçe ve İngilizce yazılar üzerinde metin sınıflandırması yapılmıştır. Veri seti üzerinde kelime çantası ve Word2Vec ile oluşturulan modeller kıyaslanmıştır. Ek olarak sözcükleri analiz edip hem köklerine ayırıp ve hem de köklerine ayırmadan 2 versiyon ile testler yapılmıştır. Modeller üzerinde lojistik regresyon ve destek vektör makineleri kullanılarak sınıflandırma yapılmıştır. Elde edilen deney sonucu kıyaslanmıştır [32].

Türkçe mailler ile metin sınıflandırma yapılmıştır. Sahte mailler tespit edilmeye çalışılmıştır. Mailler üzerinde Doc2Vec algoritması uygulanmıştır. Ayrıca farklı sınıflandırma algoritmaları da kullanılmıştır ve sonuçlar karşılaştırılmıştır [33].

Türkçe metinler üzerinde duygu analizi uygulaması gerçekleştirilmiştir. Hem sözlük tabanlı hem de hibrit yaklaşım yöntemi ile duyguların analizi yapılmıştır. Hibrit yaklaşım kullanıldığı zaman ortalama başarı oranının yaklaşık %7 yükseldiği gözlemlenmiştir [34].

Metin sınıflandırma işlemlerinde başarım değeri arttırılmaya çalışılmıştır. Bunun için de boyut indirgeme ve özellik seçimi işlemleri yapılmıştır. Veri kümesi ön işlem aşamalarından geçirilmiştir. Doc2Vec metodunu kullanılmıştır. Bu sayede veri seti oluşturan dokümanlar üzerinde vektörleştirme yöntemi kullanarak sınıflandırma işlemi yapılmıştır. Sonrasında nitelik seçimi işlemini yapmışlardır. Nitelik seçimi sayesinde sonuçların daha iyi olduğu görülmüştür [35].

Türkçe haber dokümanları ayrı bir çalışmada da incelenmiştir. Veri kümesi yaklaşık 5 Milyon haber dokümanından oluşmaktadır. Veri kümesi ön işlem aşamalarından geçirilmiştir. Modeller ilk başta Word2Vec ve Doc2Vec metotları ile eğitilmiştir. Ek olarak fastText metodu ile de eğitilmiştir. Kelime vektörleri kullanılarak dokümanların sayısallaştırılması sağlanmışlardır. 1 milyondan fazla haber kaynağı da hem test hem de eğitim amacıyla ikiye ayrılmıştır. Çeşitli makine öğrenmesi yöntemleri ile sınıflandırma yapılmıştır. FastText metodu ile en iyi sonuç alınmıştır [36].

Ayrı bir dergide sınıflandırma ile ilgili araştırma makalesi yayınlanmıştır. Kurumsal müşterilerin bilgilerinden oluşan veri kümesi üzerinde yapay sinir ağları, destek vektör makineleri ve lojistik regresyon metotları denenmiştir. Kredi başvurusunda bulunan müşteriler için risk kategorilerine göre sınıflandırılmışlardır. Böylelikle kredi başvurusu olumlu olan müşteriler belirlenmiştir. Riskli müşteriler ise ayrıca sınıflandırılıp kredi başvuruları olumlu olarak sonuçlandırılmamıştır. Kısacası kurumsal müşteriler risk yönünden riskli ve risksiz şeklinde iki sınıfa ayrılmıştır. Destek vektör makineleri yaklaşık %96 oranla en iyi sonucu vermiştir [37].

Çevrimiçi kullanıcı yorumlarını duygu analizi ile sınıflandırılmıştır. Çalışmada çok katmanlı algılayıcı algoritması kullanılmıştır. Daha sonrasında ise RapidMiner yöntemi kullanmıştır. Beraberinde ise Naive Bayes algoritması ile destek vektör makineleri algoritması, lojistik

regresyon algoritması kullanılmıştır. Çok katmanlı algılayıcı en iyi sonucu veren algoritma olduğu gözlemlenmiştir. Çalışmada her dilin kendine özgü kuralları olduğu ve birbirinden farklı olduklarından dolayı duygu analizi çalışmalarındaki zorluklardan bahsedilmiştir [38].

Orman yangınlarını önceden tahmin edebilmek için uydulardan alınan verilerden faydalanılarak çalışma yapılmıştır. Bu sayede ise hızlı müdahale ve önlem alınması amaçlanmıştır. Uydudan alınan verilerden hesaplanan arazi yüzeyi sıcaklığı ve termal anomali gibi veriler kullanılıp ormanlarda çıkan yangınları tahmin edilmesi sağlanmıştır. Çalışma kapsamında bu veriler üzerinde tahmin yapabilmek için karar ağaçları algoritmaları uygulanmıştır. Veri setinin %70 i ile karar ağaçları eğitilmiştir ve geriye kalan veri seti üzerinde modelin testi yapılmıştır. Birden çok tekrar edilerek yapılan ve uygulanan yöntemlerin performansları belirlenmiştir. Gerçekleştirilen denemeler sonucunda yaklaşık %98 hassaslık oranı ile ortaya çıkan yangınlar hatasız tahmin edilmiştir. Deney sonucunda %93 başarı ile olası gerçek durumlar tespit edilmiştir [39].

Lojistik sektöründe de yapay zeka ve makine öğrenmesi teknikleri konuları ele alınmıştır. İnceledikleri çalışmada bu konuyu ele alınarak geniş bir perspektif ile incelenmeyi hedeflemişlerdir. İncelemeler sonucunda lojistik sektöründe yapay zeka kullanıldığında işletmelerin daha verimli işler ortaya çıkarttığı ve lojistik süreçlerini kolaylaştırdığı belirtilmiştir. Müşteri kaybı tahmininde bulunurken, talep tahminlerinde, envanter yönetiminde, satın alma, lojistik operasyon planlama, lojistik takip, sipariş toplama, depo sistemleri, sevkiyat ve araç rotalama gibi uygulamalar da yapay zeka ve makine öğrenmesinin önemi vurgulanmıştır [40].

Regresyon analizleri ile karar ağaçlarının karşılaştırılması için aynı veri seti üzerinde çeşitli yöntemler uygulanarak karşılaştırmalar yapılmıştır. Bu yöntemler ile konut fiyatını hesaplayan değişkenler netleştirilmiştir. Ayrıca modellerin tahmin oranları ve performans yüzdeleri karşılaştırılıp en iyi model seçilmesi hedeflenmiştir. Çalışmada yaklaşık 400 gayrimenkul bilgisini içeren veriler kullanılmıştır. Sonuç olarak ise yaklaşımların sınıflandırma ağacına göre daha kötü performans gösterdiği gözlenmiştir [41].

6 farklı kategoride Türkçe haber kaynakları üzerinde sınıflandırma çalışmada bulunulmuştur. İlk başta haber kaynakları verileri ön işlemden geçirilmiştir. Bu metinler üzerinde ilk başta TF-IDF yöntemi kullanılmıştır. Ayrıca Word2Vec yöntemi ve fastText yöntemi de

kullanılmıştır. Python'ın Scikit-learn kütüphanesi kullanılmıştır. Çeşitli sınıflandırma yöntemleri kullanılmıştır. Bunlar sırasıyla: Naive Bayes algoritması, destek vektör makinesi algoritması, lojistik regresyon algoritması, rastgele orman algoritması ve yapay sinir ağı yöntemleri ile sınıflandırılmıştır. Araştırma sonucunda yaklaşık %95 başarı sonucu ile fastText yöntemi ve destek vektör makinesi ile kombinasyonundan sağlanmıştır [42].

Yükseköğretim kurumlarının yapay zekaya ne kadar hazır olup olmadığı ile ilgili makale yazılmıştır. Makalede yapay zekanın yükseköğretimdeki uygulamaları sorgulanmaktadır. Ayrıca bu kurumların yapay zeka ile geleceğe hazırlanmasında iyimser bakış açışı ile tartışılmıştır [43].

Bir dosyadaki yazı tipi stilinin doğrulanması ve önerilen yazı tipi stilinin daha kısa sürede sınıflandırılması için yapay zeka tabanlı algoritma ağı olan kapsül ağı kullanılmıştır. İncelenen çalışmada kapsül ağı yapısı ayrıca Naive Bayes, karar ağacı ve k-en yakın komşu algoritmaları ile de karşılaştırılmıştır. Çalışma kapsamında ise veri seti olarak 702 yazı tipi, 468 eğitim veri seti ve 234 test veri seti kullanılmıştır. Sonuç olarak kapsül ağa dayalı önerilen yazı tipi stili sınıflandırma modelinin daha iyi doğruluk ve F1 puanları ile görüntüler [44].

Duygu analizi çalışmasında sözcükleri temsil eden metotlar belirlenmiştir. Kelime çantası ile fastText yöntemleri uygulanıp sınıflandırma başarı yüzdeleri incelenmiştir. İnceleme yapılırken duygu analiz işlemlerinde ön hazırlık aşamasında sözcükleri köküne ayırmak ile sözcükleri pozitif yapmamak sınıflandırmada iyi sonuç alındığını göstermektedir. Çalışmada negatif ve pozitif duygu inceleme ile birlikte nötr, negatif ve pozitif duygu incelemeleri yapılmıştır. Toplam altı tane veri kümesi seçilmiştir. Bu veri kümeleri üzerinde ilk başta Naive Bayes ve Multinom Naive Bayes algoritmaları uygulanmıştır. Daha sonra k-en yakın komşu ile destek vektör makinesi algoritmaları da uygulanarak kelime çantası temsil metodu ile beraber WEKA uygulamasıyla sınıflandırma yapılmıştır. Veri kümeleri kelime çantası ile sözcük temsil metoduna göre test sonucu sonrasında fastText sözcük temsil metodu üzerindeki test de python yazılım dili ve fastText kütüphanesi kullanıp yapılmıştır. F-skor sonuç değeri hesaplanırken, sınıflandırma işlemi tam 10 tekrarlı olacak şekilde çapraz doğrulama metodu uygulanarak hesaplanmıştır. Sonuçta ise fastText kelime temsil metodu, kelime çantası yöntemine göre iki ayrı sınıflı duygu incelemesinde daha başarısız olduğu görülmüştür. Ayrı bir çalışmada da kelime çantası yönteminin, fastText kelime yöntemine

göre üç ayrı sınıflı duygu incelemesinde daha başarısız olduğu görülmüştür. Duygu incelemesinde sözcükleri köküne ayırmak ve olumlu yapmama işleminin fastText sözcük temsil metodu ile yapılan sınıflandırmada olumsuz veya olumlu bir şekilde fayda sağlamadığı görülmektedir. Fakat kelime çantası kullanılarak yapılan duygu incelemesinde faydası olduğu görülmüştür. Yaklaşık 0.90 f-skor sonucuyla destek vektör algoritması ile oluşturulan model ile istenilen elde edilmiştir. Üç ayrı sınıflı duygu incelemesinde en başarılı olan sınıflandırma sonucu incelendiğinde yaklaşık 0.70 f-skoru başarı değeri ile fastText kullanıp oluşturulan model ile istenilen elde edilmiştir [45].

Kitle fonlaması, bireylerin büyük miktarlarda olmayan bağışlarına dayalı yatırım ve fonlama sistemlerine denir. Kitle fonlaması projeleri üzerinde de karar ağacı ve rastgele orman algoritmaları ile sınıflandırma yapılmıştır. Kitle fonlaması olan bu platformlar ile internet vasıtasıyla iş üzerine oluşan fikrin hayata geçirilmesi ve destek alabilmesi sağlanmaktadır. Bahsedilen platformda destek ihtiyacı olan projelerin başarı yüzdesi, almış oldukları finansal destek ile birlikte artmaktadır. Ancak finansal yardım almak için proje destekleyicilerine iyi şekilde sunmak gerekmektedir. Çalışma kapsamında, farklı birçok kategorideki Kickstarter projesi makine öğrenmesi metotları ile sınıflandırma yapılarak web arayüzünde son kullanıcıya gösterilmiştir. İlgili projelerin sınıflandırılırken karar ağacı ve rastgele orman algoritmaları yöntemleri uygulanmıştır. Karar ağacı ile yaklaşık %72 ve rastgele orman ile de yaklaşık %81 başarılı şekilde sınıflandırma yapabilmiştir [46].

Metinlerin duygu sınıflarının hangisine ait olduklarını bulmak için duygu derin öğrenme ile sınıflandırılmıştır. İncelenen çalışmada e-ticaret sitesi olan Hepsiburada şirketine ait yaklaşık 243 bin kullanıcının yorumları ile oluşan veri setleri kullanılmıştır. Dengesiz veri seti üzerinde derin öğrenme yöntemleri kullanılmıştır. Yanlış pozitif oran değeri yaklaşık olarak %70'den %90 başarı oranı değerine yükselmiştir. Ayrıca doğruluk değeri de yaklaşık %95'ten %99 başarı oranı değerine yükseltilerek iyileştirme sağlanmıştır [47].

Derin öğrenme yöntemleri ve TF-IDF algoritması kullanılarak internetteki haberlerin gerçek olup olmadığının analizini yapılmıştır. İngilizce haber için Katharine Jarmul tarafından oluşturulan veri seti kullanılmıştır. Veri seti toplam 6336 haber öğesi içeriyordu. Veri setinin dağılımı ise %50 yalan, %50 gerçekti. Metin sınıflandırmasında kullanılan yöntem TF-IDF idi. Sınıflandırma, kullanılan veri seti ile yapılmış ve analiz sonucunda %93,88 başarı ve %6,12 hata elde edilmiştir [48].

Sindirim sistemi ile ilgili bilim dalı olan Gastroenteroloji ve karaciğer hastalıklarının tanı ve tedavisi ile ilgili bilim dalı olan Hepatoloji alanlarında da yapay zeka uygulanmıştır. Bu çalışmalar sayesinde ise yapay zeka uygulamaları doktorların teşhis koymasında yardımcı olacaktır [49].

Sosyal değerler ile ilgili Schwarzt modelinde bireysel olan değerlere tüketici kişilerin çevreye karşı olan duyarlı tüketim hareketleri isimsel olan lojistik regresyon analizi ile incelenmiştir. İnsanların yaşamış olduğu çevre problemlerinin daha iyi anlaşılması için ekolojik olan bakış açısı gerektirse bile problemlerin olmasında birey etkileri göz önüne alındığı zaman davranışların üzerinde çok etkisi olan bu değer yapısının incelenmesi gereklidir. Araştırmada ise bir ölçeğin diğer eksen olan bireysel değerler olan boyuttaki "Bireyin kendini geliştirmesi" ve "Bireyin kendini aşması" değer söz öbekleri ile; Geri dönüşüme destekleme, boşa harcamadan tüketme gibi çeşitli davranış hareketleri eğilimlerine göre sınıflandırılma yapılması istenilmiştir. Amaç doğrultusunda "İyilikseverlik", "Başarı", "Evrensel", "Kudret" gibi çeşitli değer yapısının çevreci farklı davranış tipleri ile beraber etkilerinin anlaşılması için yaklaşık 200 veri ile lojistik regresyon analizi ile çalışılmıştır [50].

Lisansüstü derslerin içeriğinin belirlenmesinde doğal dil işleme teknikleri kullanılmıştır. İlgili ders için IEEE'nin uzun zamandır düzenlemiş olduğu konferanslarda yayımlanan makaleler ile dersler hakkında bilgilerin netleşmesinde kullanılacak veri kümesi oluşturulmuştur. Yıl olarak 2015 ve 2018 yılları arasında kabul edilen ve konferansa gönderilen makaleler ile veri setleri oluşturulmuştur. Doğal dil işleme yöntemleri ile birlikte analizler yapılıp derslerin içerikleri kolayca netleştirilmesi sağlanmıştır. İlgili veri kümesi oluşturulduktan sonra ise önce veriler üzerinde temizleme işlemleri yapılmıştır. Sonrasında makalelerin başlıkları kelimelere ayrılmıştır. Kelimelere ayrılan veri kümesinde kelimelerin frekans değerleri belirlenip frekanslarına göre yaklaşık yirmi kelime belirlenmiştir. Bu süreçte NTK paketi olan Apache Spark çalışmada kullanılmış. Derslerin konu başlığı kolayca belirlenerek akademisyenlere büyük ölçüde kolaylık sağlamıştır [51].

Günümüzde yabancı ve en çok kullanılan diller üzerinde bir çalışma yapılmıştır. İçeriği zengin olan diller üzerinde analiz yaparken bazen iyi analiz yapılamadığı görülmüştür. Bu sorunlar üzerinde araştırma yapılmıştır. Çalışmada Word2Vec modeli kullanılmıştır. Sözcüklerin ek ve sözcüklerin köküne ayırma işleminde benzer olmayan şekilde yapıldığı

Türkçe olan metin ile eğitilmiştir. FastText modeli eğitilmiştir. Sonrasında duygu analizi ve metin sınıflandırma gibi sonuçlar karşılaştırılmıştır [52].

İnovasyon şirketler için verimlilik ve ekonomik büyüme açısından en önemli kaynaktır. Büyük olmayan firma yeniliğin yaratıcısı olarak ilk başta görülse de, bu firma ölçeği ve yenilik arasındaki ilişki her zaman tartışma konusu olmuştur. TÜİK İnovasyon Araştırması istatistikleri incelendiğinde ise yenilikçi firmaların en düşük payının küçük işletmelerde olduğu görülmektedir. Buna dayanarak, araştırma imalat sanayisinde daha çok işlev gösteren küçük olan işletmede teknolojik yeniliklerin belirleyicilerini sürekli araştırmaktadırlar. 2014 TÜİK İnovasyon Araştırması bu mikro veri kümesinden çıkartılan örneklem ile regresyonu yasaklama ve karar ağacı teknikleri ile ürün ve süreç yeniliklerini etkileyen faktörler analiz yapılmıştır [53].

Türkçe bir yazıyı sınıflandırırken, sınıflandırma başarı oranı değerini arttırmak için ardışık sözcük grubu ile tespit etmeye çalışıp bu değerin artması için TF-IDF ile Doc2Vec yöntemleri kullanılmıştır. TF-IDF ile metinler içerisindeki kelimelerin geçme sıklığı bilgisi ile beraber bu kelimeler arasındaki anlamsal ilişkileri incelemektedir. Doc2Vec yöntemi ile sözcükler ve sözcüklerin bulunduğu dokümanların arasındaki ilişkileri vektörler ile belirlemişledir. Çalışma kapsamında geleneksel olan TF-IDF ile terimlerin ağırlıklandırma ölçümü ve Doc2Vec ile beraber başarı yüzdeleri karşılaştırılmıştır. Veri kümesi, 4 ayrı haber dokümanından oluşmaktadır. Bu çalışmada sınıflandırma aşamasına geçmeden önce TF-IDF ve Doc2Vec yöntemleri kullanılmasının başarı oranını arttırdığı belirtilmiştir [54].

Birçok makale incelendiğinde farklı sınıflandırma modellerinin uygulandığını ve farklı sonuçlar aldığı gözlemlenmiştir.

3. MALZEME VE YÖNTEM

3.1. TASARIM LOG YAPISI

Şirket tarafından iletilen tasarım logları, telekom anahtarında çalışan uygulamalar ve işlemler için sorunlar ve durum ile ilgili bilgilendirici verileri içeren ve sistem üzerinde yazılım uygulamaları çalıştırılarak oluşturulur. Genel bir tasarım logu örneğinde; zaman damgası, donanım konumu, işlem adı, uygulama adı, uygulama dizini ve uygulama hakkında bilgiler içerir. Uygulama hakkındaki bilgiler kısmı dikkate alınarak, ilgili logun error, debug ya da info sınıflarından hangisine ait olduğuna karar verilecektir. Şirket tarafından sağlanan log üzerinde çalışılmıştır.

Date	Hardware	Process	Subsystem	Code Path	Data	Class
Dec 29 00:16:14	Hardware 1	Process 4	Subsystem 13	Path 20	Failed to add netlink route. error: (-101)	Error
Dec 29 00:16:19	Hardware 3	Process 5	Subsystem 14	Path 21	Unable to open '/tmp/.sessionLock' for locking -- Permission de	Error
Dec 29 00:16:51	Hardware 3	Process 6	Subsystem 15	Path 22	Error: got rc:731 from trans mode set	Error
Dec 29 00:18:01	Hardware 4	Process 7	Subsystem 16	Path 23	CN: No subscriber were found for pe-change-notification versio	Error
Dec 29 00:18:18	Hardware 3	Process 8	Subsystem 17	Path 24	SCSI: sgHlpModeSense10CmdRead failed sghErr(-2005): Sense I	Error
Dec 29 00:19:07	Hardware 1	Process 2	Subsystem 2	Path 4	rc:1 = call agent command successful	Debug
Dec 29 00:19:07	Hardware 1	Process 2	Subsystem 2	Path 4	call agent command successful	Debug
Dec 29 00:19:07	Hardware 1	Process 2	Subsystem 2	Path 7	Cannot remove timer 0, unexpected timerId	Debug
Dec 29 00:19:07	Hardware 1	Process 2	Subsystem 2	Path 7	Started timer: 1, timeoutValSec: 60, delayedStartSec: 0	Debug
Dec 29 00:19:07	Hardware 1	Process 2	Subsystem 2	Path 9	setupIp: Added IP address A.B.C.D/26 on interface eth1	Debug
Dec 29 00:19:41	Hardware 2	Process 10	Subsystem 19	Path 26	Issuing instantiation cmd with systemd: /usr/bin/systemd-run --	Info
Dec 29 00:19:41	Hardware 2	Process 10	Subsystem 19	Path 26	VCA_SU_1:caoam: INSTANTIATING: "/run/opt/private/ca/20.0.(	Info
Dec 29 00:19:41	Hardware 2	Process 10	Subsystem 19	Path 26	VCA_SU_1: exit from swdld	Info
Dec 29 00:19:42	Hardware 2	Process 2	Subsystem 2	Path 8	TIPC socket created. FD: 35	Info
Dec 29 00:19:42	Hardware 2	Process 2	Subsystem 12	Path 19	==========> Sync Execute got response!!	Debug
Dec 29 00:19:42	Hardware 2	Process 2	Subsystem 12	Path 19	Sync Execute got messageType 0X1aab	Debug
Dec 29 00:19:42	Hardware 2	Process 2	Subsystem 12	Path 19	found STAMP: len=8, val=1609219183	Debug
Dec 29 00:19:42	Hardware 2	Process 2	Subsystem 12	Path 19	found STAMP in SyncResponseMessage	Debug
Dec 29 00:19:42	Hardware 2	Process 2	Subsystem 12	Path 19	got good Sess Req Id: reqi:1609219183, msg:1609219183. n	Debug
Dec 29 00:19:42	Hardware 2	Process 2	Subsystem 12	Path 19	Current Wait perioid to wait 5000	Debug
Dec 29 00:19:43	Hardware 2	Process 2	Subsystem 2	Path 4	main: AMF registration starting	Info
Dec 29 00:19:43	Hardware 2	Process 2	Subsystem 2	Path 4	main: Moving-Master registration starting	Info
Dec 29 00:19:43	Hardware 2	Process 2	Subsystem 2	Path 4	main: caoam init OK, entering main loop with timers	Info
Dec 29 00:19:43	Hardware 2	Process 1	Subsystem 1	Path 3	TIPC socket created. FD: 25	Debug
Dec 29 00:19:43	Hardware 2	Process 1	Subsystem 1	Path 2	Init HA state to: SA_AMF_HA_UNASSIGNED	Info

Şekil 3.1: Anonim log yapısı.

3.2. VERİ ÖN İŞLEME

Veriler tamamlanmamış, tutarsız ya da güncelliğini kaybetmiş olabileceği için kaliteli veri için ön işleme yapılmalıdır. Şirket tarafından sağlanan tasarım log üzerinde ise ilk başta veri ön işleme işlemleri adım adım yapılmıştır [51].

3.2.1.Noktalama İşaretlerinin Kaldırılması

Tasarım log içerisindeki data kolonundaki her cümlenin içerisindeki noktalama işaretleri aşağıdaki örnekteki gibi kaldırıldı.

Body_Text	Body_Text_Clean
TIPC socket created. FD: 25	TIPC socket created FD 25
Cannot remove timer 0, unexpected timerId	Cannot remove timer 0 unexpected timerId
fsman AMF init succeeded.	fsman AMF init succeeded
a current _entries. size: for 1	a current entries size for 1
main: AMF registration starting	main AMF registration starting

Şekil 3.2: Noktalama işaretlerinin kaldırılması örneği.

3.2.2.Tokenize Etme

Data kolonundaki her cümle artık noktalama işaretlerinden arındırılmıştır. Tokenize işlemi ile ise aşağıdaki örnekte olduğu gibi noktalama işaretlerinden arındırılmış cümleler için kelimeleri virgül ile ayırma işlemi yapılmaktadır.

Body_Text_Clean	Body_Text_Tokenized
TIPC socket created FD 25	[tipc, socket, created, fd, 25]
Cannot remove timer 0 unexpected timerId	[cannot, remove, timer, 0, unexpected, timerid]
fsman AMF init succeeded	[fsman, amf, init, succeeded]
a current entries size for 1	[a, current, entries, size, for, 1]
main AMF registration starting	[main, amf, registration, starting]

Şekil 3.3: Tokenize etme işlemi.

3.2.3.İstenmeyen Kelimelerin Çıkartılması

Algoritmanın daha iyi sonuç vermesi için okunan log dosyası içerisindeki cümleler üzerinde yapılması gereken bir işlem de istenmeyen kelimelerin çıkartılmasıdır ve bu işleme kısacası veri temizliği diye nitelendirebiliriz. Veri temizliği yapmak, sistemin başarım oranını artıran en önemli adımlardan bir tanesidir. Bu çalışmada veri temizlemesi yapılırken cümleye olumlu ya da olumsuz olacak şekilde herhangi bir anlam katmayan ve bilgi içermeyen durdurma kelimeleri (stopwords) kaldırıldı. Bu kelimelere örnek olarak: "in", "on", "at", "from", "until", "this", … gibi kelimeler gösterilebilir.

Tasarım log içerisindeki data kolonundaki cümlelerden bu kelimeler aşağıdaki gibi çıkartılmıştır.

Body_Text_Tokenized	Body_Text_Nostop
[tipc, socket, created, fd, 25]	[tipc, socket, created, fd, 25]
[cannot, remove, timer, 0, unexpected, timerid]	[remove, timer, 0, unexpected, timerid]
[fsman, amf, init, succeeded]	[fsman, amf, init, succeeded]
[a, current, entries, size, for, 1]	[current, entries, size, 1]
[main, amf, registration, starting]	[main, amf, registration, starting]

Şekil 3.4: İstenmeyen kelimelerin çıkartılması örneği.

3.2.4.Kelime Kök Analizi

Data kolonundaki cümlelerin kelime köklerine erişebilmek için stemming (kök bulma) ve lemmatization (alma) yöntemleri uygulanmıştır. Bu işlemlerin amacı her kelimeyi eklerinden arındırıp ortak temel ya da köke dönüştürmektir. Aşağıdaki örnekteki gibi bu işlemler uygulanarak kelimelerin köklerine bakılmıştır. Kök bulma yöntemi ile -ing, -ly ve -s ekleri kaldırılmıştır. Ancak İngilizce gitmek anlamına gelen "goes" kelimesindeki "s" kaldırılması ile anlamlı bir köke ulaşılmadığı görülmüştür ve bu tarz örnekler ile karşılaşmamak için alma yöntemi ardına uygulanılmıştır.

Body_Text_Nostop	Body_Text_Stemmed	Body_Text_Lemmatized
[tipc, socket, created, fd, 25]	[tipc, socket, creat, fd, 25]	[tipc, socket, create, fd, 25]
[remove, timer, 0, unexpected, timerid]	[remov, timer, 0, unexpect, timerid]	[remove, timer, 0, unexpected, timerid]
[fsman, amf, init, succeeded]	[fsman, amf, init, succeed]	[fsman, amf, init, succeed]
[current, entries, size, 1]	[current, entri, size, 1]	[current, entry, size, 1]
[main, amf, registration, starting]	[main, amf, registr, start]	[main, amf, registration, start]

Şekil 3.5: Kelime kök analizi yapılması.

3.3. TERİM FREKANS-TERS BELGE SIKLIĞI YÖNTEMİ

TF-IDF(Terim frekansı-ters doküman sıklığı), metin analizlerinde doğal dil işleme için makine öğrenimi algoritmalarında kelimeleri puanlamak için kullanılan yöntemlerden birisidir. Bir belgedeki her kelimeyi o belgedeki her kelime ile ne kadar alakalı olduğunun ilişkisini kurmaktadır ve numerik olarak ilişkilendirmektedir. Bir belgenin bir belge koleksiyonundaki bir belgeyle ne kadar alakalı olduğunu değerlendiren istatistiksel bir ölçüdür. Belge arama ve bilgi erişimleri için kullanılmaktadır. Metin içerisindeki anahtar sözcüklerin çıkartılmasında da faydalanılmaktadır.

TF-IDF hesaplanırken; TF (Terim frekansı) ve IDF (Ters Belge Sıklığı) metrikleri ayrı ayrı hesaplanır ve çarpılır. Terim frekansı, bir belgedeki kelime sıklığını ifade etmektedir. Sıklığı hesaplarken bir kelimenin bir belgede göründüğü örneklerin ham sayımıdır. Ters belge sıklığı

ise bir kelimenin tüm belge setinde ne kadar yaygın olduğu ya da nadir olduğu anlamına gelmektedir. Bu metrik ile toplam belge sayısı alınarak, bir sözcük içeren belgelerin sayısına bölünerek ve logaritma hesaplanır. Kelime çok yaygınsa ve birçok belgede yer alıyorsa bu sayı 0'a yaklaşacaktır. Aksi taktirde 1'e yaklaşacaktır. TF ve IDF metriklerinin çarpılması ile bir belgedeki bir kelimenin TF-IDF puanı hesaplanır. Puan ne kadar yüksekse, söz konusu olan kelime o belge ile o kadar alakalıdır şeklinde doğru orantı kurulmaktadır [54].

Belge seti (D), belge setindeki belge (d), belgedeki kelime (t) ve toplam belge sayısı(N) temsil edecek şekilde matematiksel olarak aşağıdaki şekilde ifade edilmektedir.

$$tf\ idf\ (t, d, D) = tf\ (t, d)\ .\ idf\ (t, D)$$

$$tf\ (t, d) = log\ (1 + freq\ (t, d))$$

$$idf\ (t, D) = log\ (\ \frac{N}{count\ (d \in D : t \in d)}\ )$$

Denklem 3.1: Terim frekans-Ters belge sıklığı formülü [55].

3.4. KELİME ÇANTASI

Metin belgelerinden özellik çıkartılması için kullanılan ve makine öğrenme algoritmalarının eğitiminde de yaygın olarak kullanılan bir yöntem olan kelime çantası, her bir girişin önceden tanımlandığı kelime sözlüklerinde her kelimeye karşılık gelen sabit uzunlukta bir vektör tanımlanması işlemleridir. Vektör boyutu, sözlüğün boyutuna eşittir. Sonrasında bu vektörü kullanarak bir metni temsil etmek için, sözlükte her kelimenin metinde kaç kez görüldüğü sayılır ve bu sayıya karşılık gelen vektörler bilinir. Örneğin; sözlüğümüz {Araba, çok, biraz, güzel} kelimelerini içeriyorsa ve "Araba çok güzel" metnini vektörleştirmek istenilirse, vektör (1,1,0,1) olarak belirlenir [45].

3.5. ÇAPRAZ DOĞRULAMA

Çapraz doğrulama ile bütün veri kümesini eğitim ve test verisi olarak kullanabilmektedir. Model kurulduğunda veri belirli oranlarda eğitim ve test olarak ayırmaktadır. Ancak modelin öğrenmesini istediğimiz bazı uç durumlar, test için ayırmış olunan bölümde kalmış da olabilmektedir. Değerleri eğitim veri kümesi modeli eğitilirken olamayacağı için algoritma

test ile kontrol yapıldığında sonuçlar istediği şekilde çıkmayacaktır. Çapraz doğrulama ise bu tarz durumlar ile karşılaşmayıp optimum sonuç yakalayabilmek için kullanılmaktadır. Tez çalışması kapsamında eğitim için %70 olacak şekilde ve test içinse %30 olacak şekilde model kuruldu. Model kurulduktan sonra test veri seti içerisinde çapraz doğrulama sonuçlarına bakılmıştır ve yüzdelik oranlar karşılaştırıldı. Sonuçların karşılaştırılması ile çalışmada kullanılan algoritmaların sonuçlarının tutarlı olup olmama durumları kontrol edilmiştir.

3.6. DOĞRULUK, KESİNLİK, DUYARLILIK VE F1 PUANI

En doğru modelin hangisi olduğuna karar verilmesi gereken durumlarda model seçimi için doğruluk (accuracy) değerine bakılmaktadır. Yalnız doğruluk değerine bakılarak bir modelin başarısını ölçmek tek başına yeteli değildir. Bu metriğe bakılmasının yanı sıra kesinlik (precision), duyarlılık (recall) ve F1 puanı metriklerine de bakılması gerekmektedir.

Kesinlik, pozitif olarak tahmin edilen değerin kaç tanesinin pozitif olacağını göstermektedir. Duyarlılık ise pozitif olarak tahmin edilen işlemlerin kaç tanesinin pozitif tahmin edildiğini gösteren bir metriktir. F1 puan değeri ise kesinlik değerinin ve duyarlılık değerinin harmonik ortalamasını ifade etmektedir. Her sınıfa karşılık gelen puanlar, diğer tüm sınıflara kıyasla o sınıftaki veri noktalarının sınıflandırılmasında sınıflandırıcının doğruluğunu söylemektedir. Destek ise o sınıfta yer alan gerçek yanıtın örnek sayısıdır. Makro ortalama da metriği her sınıf için bağımsız olarak hesaplamaktadır ve daha sonra ortalama alınmaktadır. Bu nedenle tüm sınıfları eşit şekilde işleme tabi tutmaktadır. Fakat mikro ortalama da tüm sınıfların ortalama metriği hesaplamak için katkısını toplar. Çok sınıflı bir sınıflandırma kurulumunda, sınıf dengesizliği olabileceği düşünülen durumlarda mikro ortalama tercih edilmektedir.

3.7. KULLANILAN SINIFLANDIRMA ALGORİTMALAR

3.7.1. Doğrusal Sınıflandırıcılar: Naive Bayes Sınıflandırıcısı

Naive Bayes sınıflandırması aslında Bayes Teoremi'ne dayanarak oluşturulmuştur. Teorem ise bir olayın olma olasılığının, elde edilen yeni bilgilerden nasıl etkilendiğini göstermektedir. Bayes Teoremi'nin mantığı sırasıyla ise P(A) ifadesi A'nın olma olasılığını, P(B) ifadesi B'nin olma olasılığını, P(A|B) ifadesi B'nin doğru olduğu bilindiğinde A'nın olma olasılığını ve P(B|A) ifadesi A'nın doğru olduğu bilindiğinde B'nin olma olasılığını ifade edecek şekilde aşağıdaki şekilde ifade edilmiştir.

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

Denklem 3.2: Naive Bayes formülü I [56].

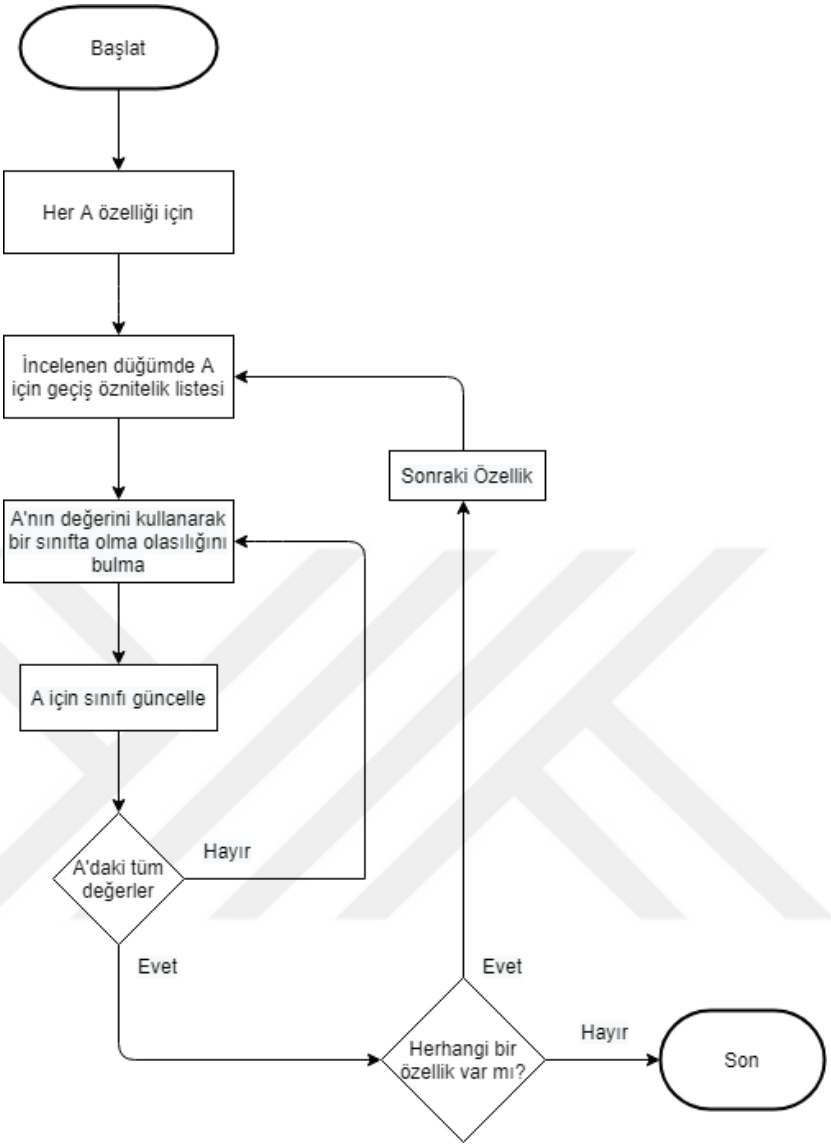
Bir veri setinde bağımsız değişkenlerin oluşturduğu seti X ile, bağımlı değişkenlerin Y ile ifade edildiği düşünülürse, Bayes Formülü'ne göre P(A|B) yerine P(Y|X) hesaplanacaktır. Eğer birden fazla bağımsız değişkene göre hesaplanmak istenilirse Naive Bayes formülü ile hesaplama yapılabilir. Naive Bayes'te her özellik (X1, X2, X3, …, Xn) birbirinden bağımsız olarak değerlendirilip hesaplama yapılmaktadır.

$$P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

Denklem 3.3: Naive Bayes formülü II [56].

Naive Bayes'in avantajlı yönlerine bakılacak olursa özellikleri birbiri ile bağımlı olmadığı kabul edildiğinden Lojistik regresyon gibi benzer modellerden yüksek başarı sonuç vermektedir. Ayrıca dengesiz veri setlerinde kullanılabilmektedir ve yüksek boyutlu verilerde iyi çalışabilmektedir. Gerçek hayatta her özellik bir noktada birbirine bağımlı olduğu için bir dezavantajdır.

Tez çalışmasında ise scikit-learn kütüphanesi naive_bayes modülü GaussianNB sınıfından yaratacağımız classifier nesnesi ile model oluşturulmuştur.



Şekil 3.6: Naive Bayes sınıflandırıcısı akış.

3.7.2. Doğrusal Sınıflandırıcılar: Lojistik Regresyon

İstatistiksel metot olan lojistik regresyon, bir ya da birden fazla bağımlı olmayan değişkenleri bulunduran veri setlerini incelerken kullanılmaktadır. Sonuç ise ikili değişkenlerle ölçülmektedir. Doğru-Yanlış, 1-0 gibi yalnızca iki olası sonuç vardır.

Lojistik regresyon hedefi iki yönlü karakteristiği olan bir dizi bağımlı olmayan değişkenlerin birbiriyle ilişkisini tanımlayabilmek için uygun olan modeli bulmaya çalışmaktır. Karakteristiklerin varlık olasılıklarını logit dönüşümü hakkında tahminde bulunmak için formül katsayısı üretmektedir.

Formüllerin içerisinde olasılık(odds), olasılık oranı(odds ratio) ve logit terimler geçmektedir. Odds tanım olarak başarı veya görülme olasılığının, başarısızlık yada görülmeme olasılığına (1-p) oranıdır. Odds ratio tanım olarak iki odds'un birbirine oranına denir ve iki değişken arasındaki ilişkinin özet ölçüsüdür denilebilir. Logit, odds rationun doğal logaritmasına denir. Odds ratio ise asimetriktir ve odds ratio'nun doğal logaritması alınarak simetrik şekle dönüşür.

$$logit(p) = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \ldots + b_kX_k$$

Denklem 3.4: Lojistik regresyon formülü I [57].

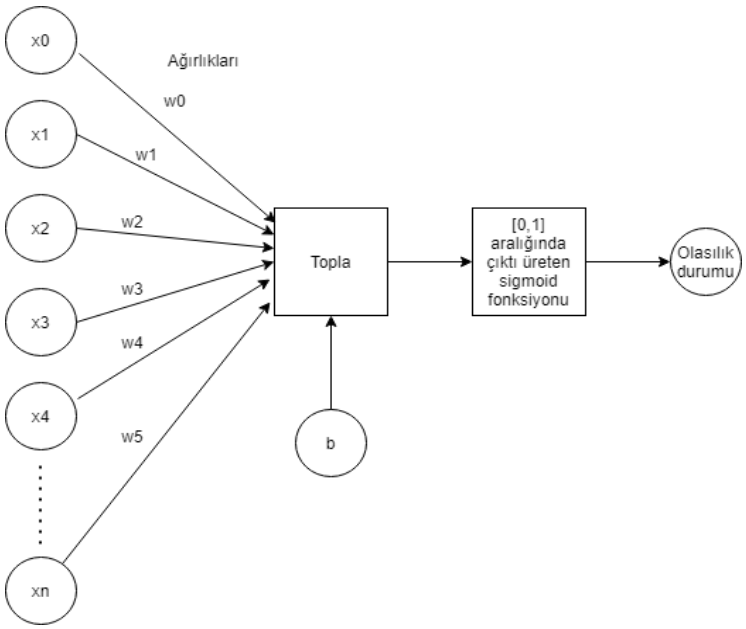
Formüldeki p karakteristik özelliğin var olma durumunu ifade etmektedir.

$$odds = \frac{p}{1-p} = \frac{probability\ of\ presence\ of\ characteristic}{probability\ of\ absence\ of\ characteristic}$$

$$logit(p) = \ln\left(\frac{p}{1-p}\right)$$

Denklem 3.5: Lojistik regresyon formülü II [57].

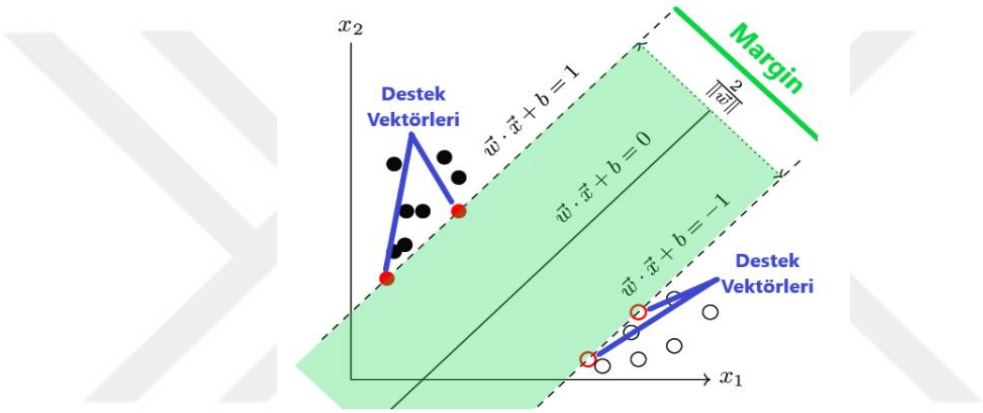
Lojistik regresyon tahmin değerini, örnek verilen değerlerin gözlem olasılık değerini arttırmaya yönelik parametreyi seçmektedir.



Şekil 3.7: Lojistik regresyon akış.

3.7.3. Destek Vektör Makinesi

Gözetimli öğrenme yöntemlerinden biri olan destek vektör makineleri(support vector machine) sınıflandırma da kullanılmaktadır. Destek vektör makineleri iki ayrı sınıfa ait olan verileri ayırmada kullanılmaktadır. Doğru ile düzleme yerleşen noktalar ayrılmaktadır. Doğru üzerinde iki sınıf noktalar içerisinde maksimum konumda olması amaçlanmaktadır. Karmaşık olan veri kümeleri ve küçük-orta ölçekteki veri kümeleri için uygun bir seçimdir.



Şekil 3.8: Destek vektör makineleri [58].

Şekilde siyah ve beyaz olmak üzere iki sınıf görülmektedir. Sınıflandırma yaparken de verinin hangi sınıfta yer alacağına karar verilirken destek vektör makineleri kullanılabilir. İki sınıfı ayıran bir doğru çizilmektedir. Doğrunun +1 ve -1 arasındaki yeşil olan bölgeye marj denilmektedir. Marj ne kadar geniş ise sınıflar o kadar iyi ayrıştırılmıştır anlamanı gelmektedir.

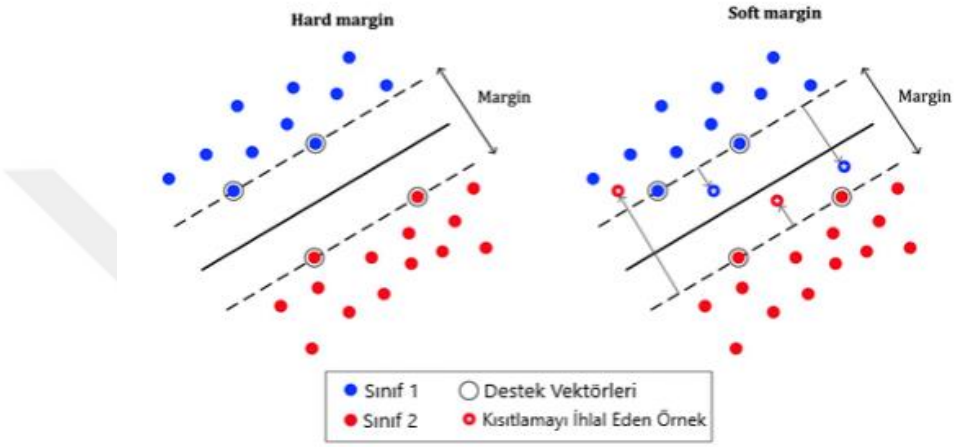
Destek vektör makineleri formülü aşağıdaki gibi olup; w ağırlık vektörü, x girdi vektörü, b ise sapmayı ifade etmektedir.

$$\hat{y} = \begin{cases} 0 \text{ if } \mathbf{w}^T \cdot \mathbf{x} + b < 0, \\ 1 \text{ if } \mathbf{w}^T \cdot \mathbf{x} + b \geq 0 \end{cases}$$

Denklem 3.6: Destek vektör makineleri formülü [58].

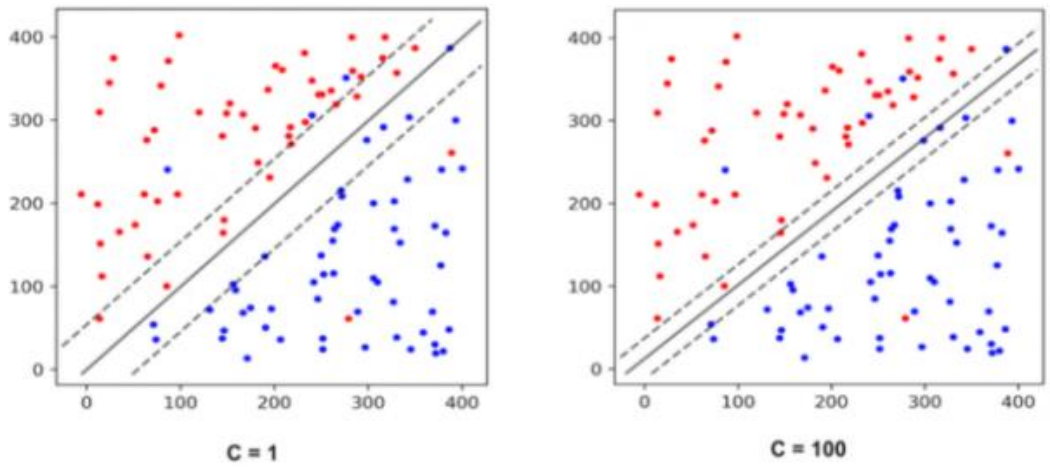
Formüle göre sonuç 0 dan küçük çıkarsa beyaz noktalara yakın olacaktır ve sonuç 0 dan büyük veya eşitse siyah noktalara yakın olacaktır.

Marj bölgesi içerisinde sınıflandırmaya dahil olacak örnekler görülürse bu duruma yumuşak marj denilmektedir. Eğer verilerimiz doğrusal olarak ayrılabiliyorsa aykırı değerlere karşı duyarlıdır ve bu duruma sert marj denilmektedir.



Şekil 3.9: Sert marj ile yumuşak marj karşılaştırması [58].

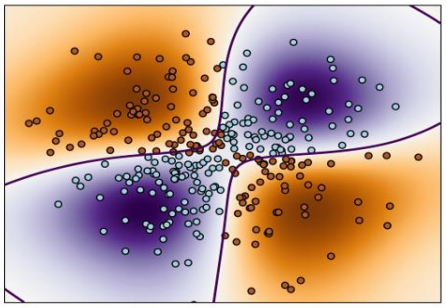
Marjdaki sınırlar C parametresi ile kontrol edilmekte olup, C parametresi ne kadar büyükse Marj o kadar dardır denilmektedir. Ayrıca modelin aşırı uyum göstermiş olduğu durumda C'nin azaltılması gerekmektedir.



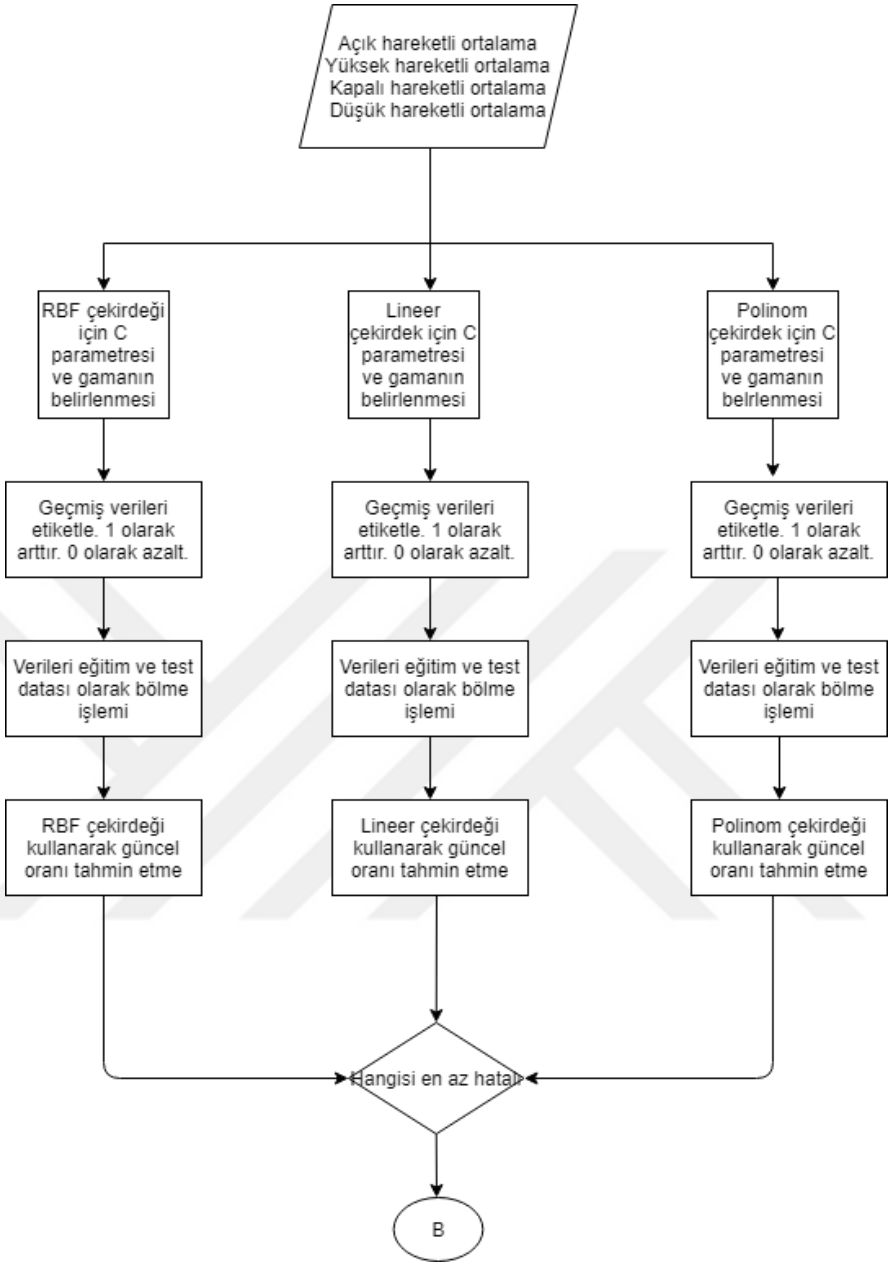
Şekil 3.10: Marj sınırlarının belirlenmesi [58].

Destek vektör makineleri doğrusal ve doğrusal olmayan şekilde ikiye ayrılmaktadır. Doğrusal destek vektör makinelerinde iki sınıfa ait örnekler doğrusal olarak dağılmaktadır ve bu veri setini ikiye ayıran doğru karar doğrusudur. Doğrusal olmayan destek vektör makinelerinde doğrusal bir düzlem çizilemez ve çekirdek hilesi kullanılmaktadır. Doğrusal olmayan verilerde çekirdek metodu kullanılmaktadır ve bu kullanım sayesinde makine öğrenimini önemli ölçüde arttığı görülmektedir.

Sık kullanılan çekirdek metotları ise; Polynomial Kernel ve Gaussian Radial Basis Function Kernel'dir. 2 boyuttan çıkıp 3 veya daha fazla boyutta işlem yapılırken Polynomial Kernel yöntemi kullanılmaktadır. Gaussian RBF (Radial Basis Function- Radyal Temel Fonksiyonu) Kernel yönteminde ise sonsuz boyuttaki destek vektör makinelerini bulur ve noktaların belirli noktaya benzeme oranını normal dağılım yöntemi ile hesaplar ve sınıflandırır [58].



Şekil 3.11: Doğrusal olmayan destek vektör makineleri [59].



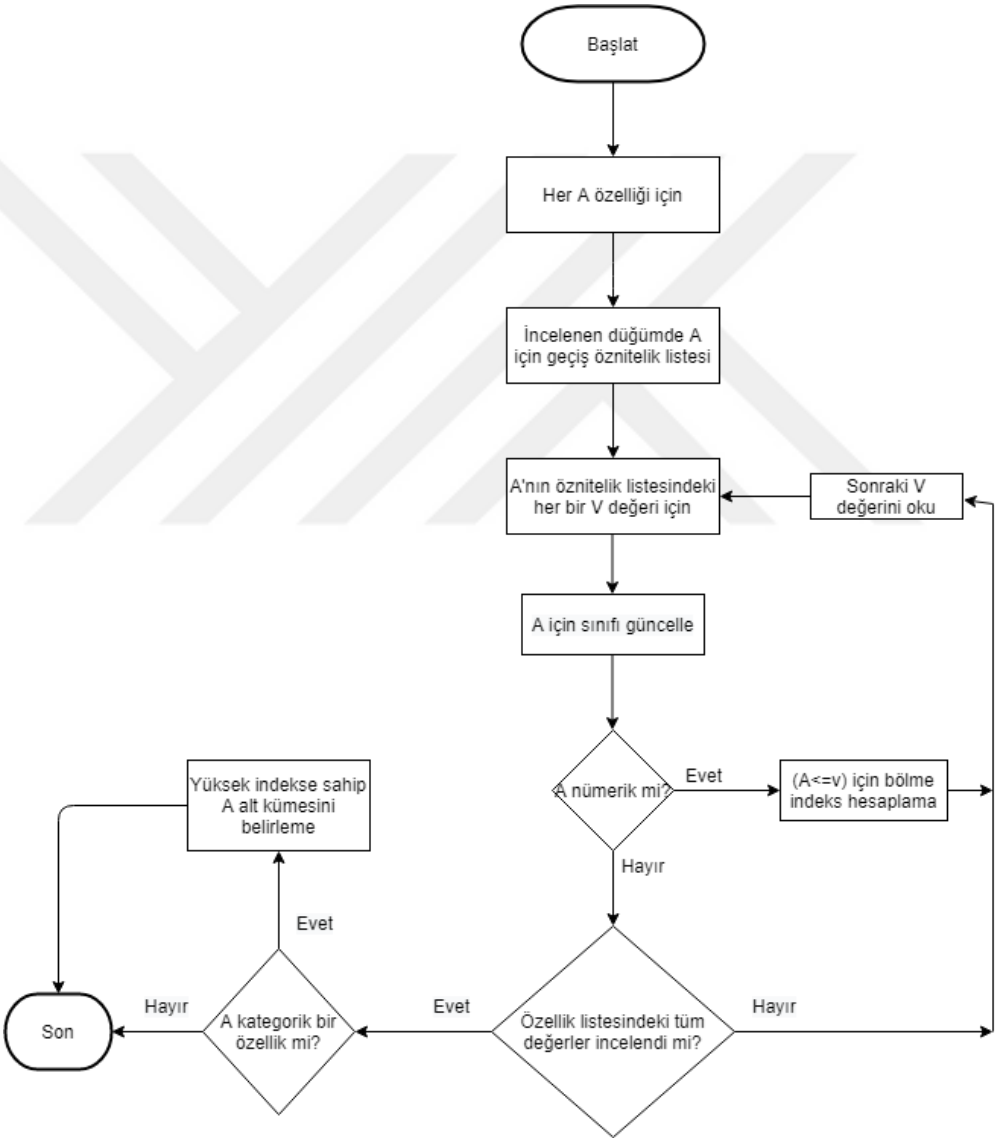
Şekil 3.12: Destek vektör makineleri akış.

3.7.4. Karar Ağacı

Karar ağaçları, bilinen sınıflandırma makine öğrenmesi algoritmalarındandır. Verilerin nitelikleri içerisinden kurallar çıkartıp kuralları öğrenip değişkenlerin değerini tahminen bulan model oluşturmaktadır. Algoritmanın bir başka özelliği ise eksik değer desteklememektedir ve model eğitilmeden önce eksik değerlerin çözülmesi gerekmektedir. Karar ağacı yönteminde öncelikle sorun tanımlanmalıdır. Sonrasında karar ağacı çizilmesi ve yapılandırılması

gerçekleştirilir. Olayların oluşma olasılıkları atanır ve işlemler ile karşılaştırmalar yapılarak tahmin sonucu sunulmaktadır.

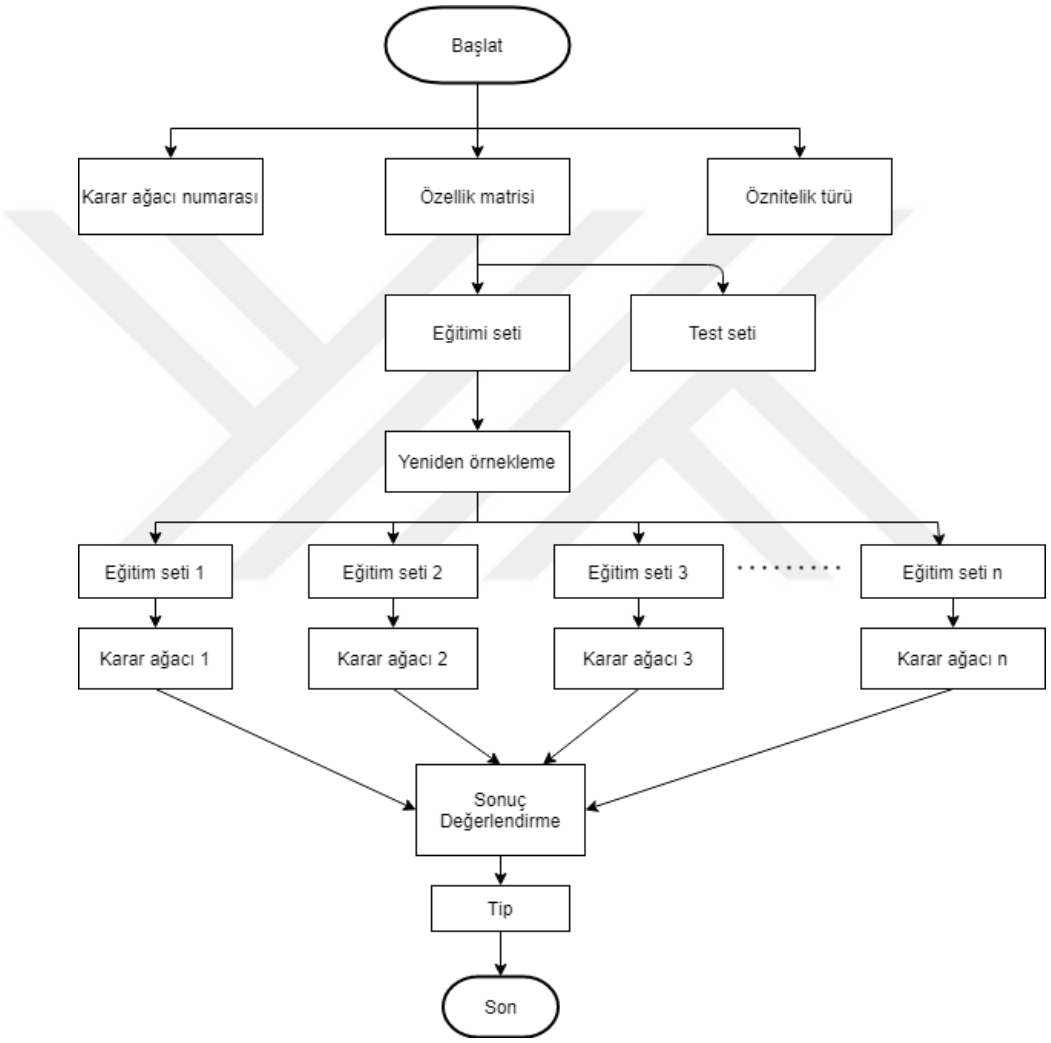
Küçük karar ağaçlarını yorumlamak kolaydır. Anlaşılabilir kurallar oluşturulabilir. Yer karmaşıklığı ve zaman belli parametrelere bağlıdır. Bunlardan bazıları; öğrenme küme örnek sayısı, oluşan ağaç yapısına ve nitelik sayısına bağlıdır. Sürekli ile ayrık nitelik değerleri için karar ağaçları kullanılabilir.



Şekil 3.13: Karar ağaçları akış.

3.7.5. Rastgele Orman

Rastgele orman çalışma mantığı olarak birden fazla karar ağacı oluşturur. Burada bağlı olunan veri kümesi içerisinden rastgele seçilen karar ağaçlarıdır. Sonuç üretildiğinde ise karar ağaçlarında ortalama olarak değerler alınmaktadır. Sonrasında ise sonuçlar üretilmektedir. Karar ormanları oluşumu anında sonuçların üzerinden tahminde bulunulmaktadır.

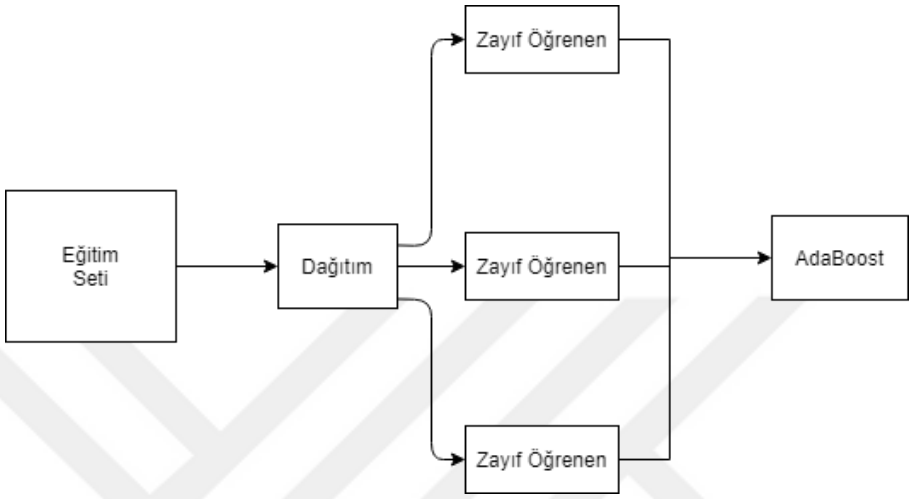


Şekil 3.14: Rastgele orman akış.

3.7.6. Arttırma

Arttırma, zayıf modelleri güçlendirme eğilimindedir. Çeşitli arttırma algoritmaları vardır: XGBoost, GradientBoost, AdaBoost ve CatBoost. Tez kapsamındaki çalışmada ise AdaBoost algoritması kullanılmıştır. Ağaç tabanlı algoritmalardan oluşan AdaBoost'u güçlü kılan oluşturulan bir ağaç yapısını, sonraki ağacın önceki ağaçtan gelen hataları aza indirme

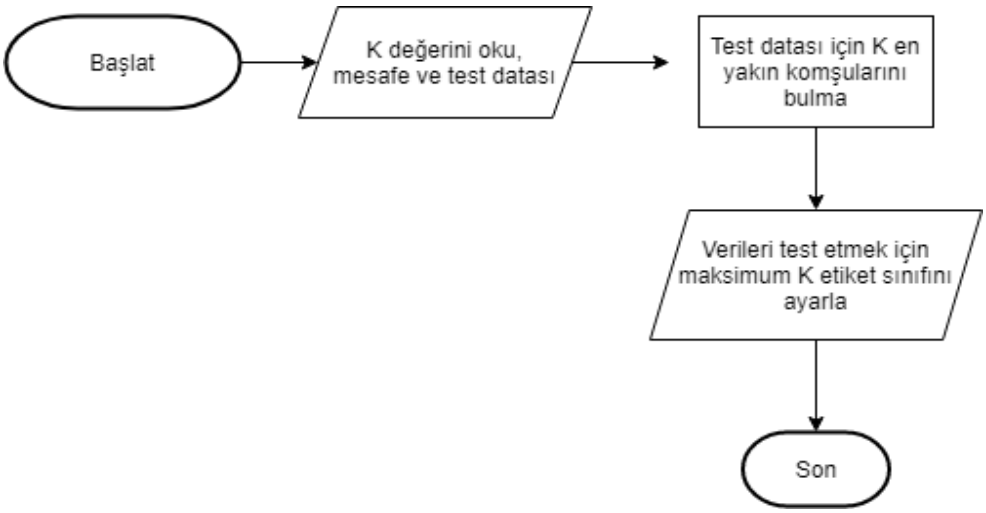
eğiliminde olması ve algoritma sayesinde zayıf model güçlü kural haline getirilme amacını gerçekleştirir. AdaBoost yöntemi ile son sınıflandırmaya karar verebilmek için ormandaki karar ağaçlarının tahminlerini belirlerken rastgele ormana benzemektedir. Bazı farklılıklar da bulunmaktadır. Örnek: AdaBoost'ta karar ağaçlarının derinlik sayısı 1'dir ve 2 yapraktır.



Şekil 3.15: AdaBoost akış.

3.7.7. K-En Yakın Komşu Sınıflandırıcısı

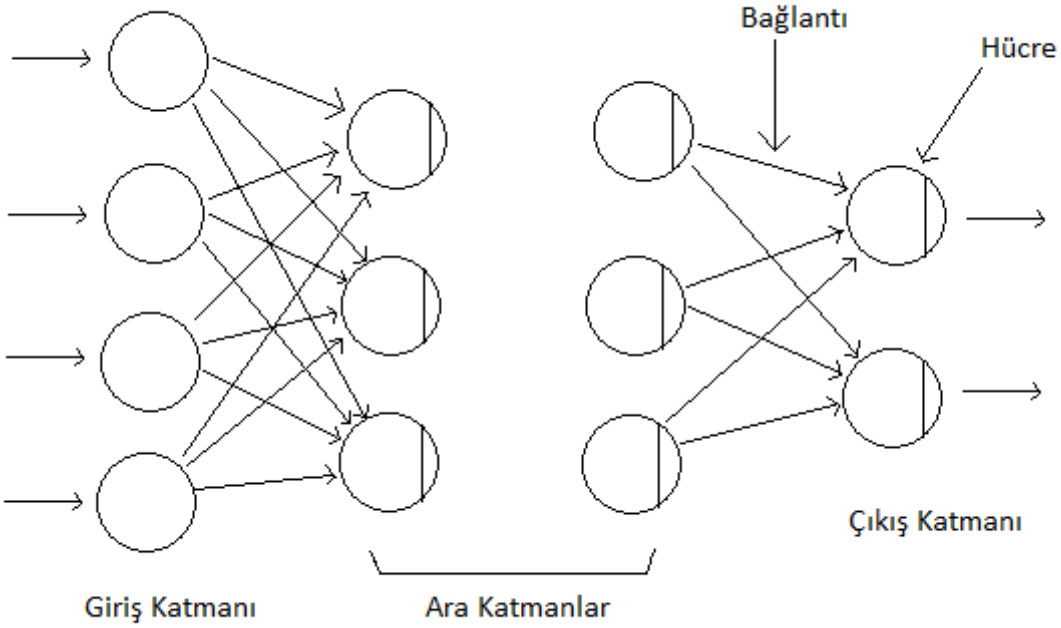
Gözetimli öğrenme metotlarından sınıflandırma işleminde kullanılan bir yöntem de KNN (K Nearest Neighborhood - K En Yakın Komşu) algoritmasıdır. KNN yöntemi ile yeni olan noktalara en yakın noktalar aranmaktadır. Bilinmeyen noktaların en yakın komşularını miktarını K temsil etmektedir. Sonuç tahmininde bulunmak için algoritma için K miktarı seçilir. İlk olarak K değeri belirlenmektedir ve sonrasında başka nesnelerden hedef nesnelere olan öklit uzaklıkları hesaplanmaktadır. Daha sonra uzaklıklar sıralanmaktadır. Minimum uzaklıklara bağlı olarak en yakın komşuları bulmaktadır. Son olarak en yakın komşu kategorileri toplanmaktadır ve uygun olan komşu kategorisi seçilmektedir.



Şekil 3.16: K-En yakın komşu akış.

3.7.8. Çok Katmanlı Algılayıcı

Basit bir sinir ağını tanımlamak gerekirse tek nörondan oluşan algılayıcılara denir. İleri beslemeli sinir ağları ise çok katmanlı algılayıcılar olarak bilinmektedir. Birden çok nörondan oluşmaktadır ve karmaşık ağlardır. Katmanlar arasında girdi katmanını, gizli katmanını ve çıktı katmanını ayırt edilebilmektedir. Genellikle çok katmanlı algılayıcılar birbirlerine tam bağlıdır. Belirli katmandaki algılayıcı ile bir diğer katmanda bulunan algılayıcı arasında bağlantısı var olduğu anlamına gelmektedir. Algılayıcılar sadece sınıfların birbiri ile doğrusal olarak ayrımlarını ifade ederken, çok katmanlı algılayıcılar sınırlamanın üstesinden gelmektedir ve ayrıca karmaşık karar sınırını da temsil edebilmektedir.



Şekil 3.17: Çok katmanlı yapay sinir ağının genel hali.

Tek katmanlı olan yapay sinir ağı genel olarak lineer ayrılabilen sınıflandırma probleminde kullanılmaktadır. Lineer ayrılamayan sistemlerde iyi sonuçlar vermez yani sistemi tam sınıflandıramamaktadır. Çok katmanlı olan yapay sinir ağları da lineer olmayan sistemleri çözerken iyi sonuçlar üretir. Sonucun iyi olma durumu; kullanılan ağın yapısına, kullanılan gizli katmanların sayısına ve kullanılan aktivasyon fonksiyonlarına bağlı olarak değişir.



Şekil 3.18: Çok katmanlı algılayıcı akış.

4. BULGULAR

Tasarım logları için belirtilen sınıflandırma algoritmaları uygulanılmıştır. Algoritmaların doğruluk sonuçlarının yanı sıra çapraz doğrulama, doğruluk, kesinlik, duyarlılık ve F1 puanları incelenmiştir. Ayrıca kelime çantası ve terim frekans-ters belge sıklığı yöntemleri ayrı ayrı kullanarak tüm sonuçlar incelenmiştir. Sonuçlar aşağıdaki gibidir.

	Bag of Words	TF-IDF
Naive Bayes	91,92	89,89
Logistic Regression	94,95	91,91
Support Vector Machines	59,60	59,59
Decision Tree	89,90	87,87
Random Forest	85,86	84,84
Boosting	71,72	71,71
K-Nearest Neighbor	69,70	71,71
Multilayer Perceptron	90,91	59,59

Tablo 4.1: Algoritma doğruluk sonuçları.

	Bag of Words		TF-IDF	
Naive Bayes	[0.82	0.85714286]	[0.88	0.85714286]
Logistic Regression	[0.88	0.81632653]	[0.84	0.73469388]
Support Vector Machines	[0.58	0.59183673]	[0.58	0.59183673]
Decision Tree	[0.74	0.7755102]	[0.84	0.75510204]
Random Forest	[0.78	0.73469388]	[0.82	0.79591837]
Boosting	[0.8	0.67346939]	[0.8	0.67346939]
K-Nearest Neighbor	[0.56	0.79591837]	[0.56	0.75510204]
Multilayer Perceptron	[0.78	0.81632653]	[0.6	0.59183673]

Tablo 4.2: Çapraz doğrulama sonuçları.

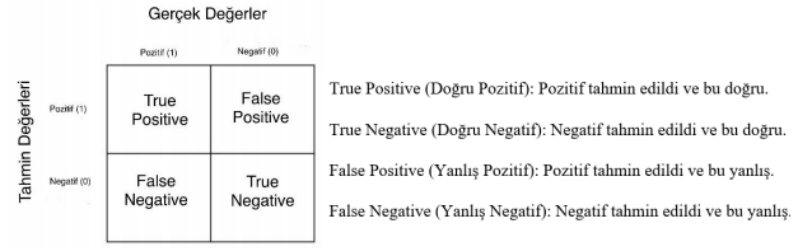
3. Bölümde bahsedilen metotların deneysel uygulama sonuçları sunulmuş ve analiz edilmiştir. Uygulama sonuçları değerlendirilirken bazı başarı ölçüm terimleri baz alınır. Bu terimler: doğruluk, F-ölçüsü, keskinlik ve duyarlılık terimleridir. Doğruluk, sınıflandırıcının doğru değeri ölçme yeteneğidir. Başka bir deyişle, ölçülen değerin standart veya gerçek bir değere yakınlığıdır. F-ölçü hesaplaması, keskinlik ve duyarlılık değerlerine dayanır. Kesinlik, doğru tahminlenen değerlerin gerçekte ne kadarının doğru olduğu gösterir ve bir sınıftaki doğru

tahmin edilen öğelerin sayısının o sınıfta olacağı tahmin edilen toplam öğe sayısına bölünmesiyle elde edilen orandır. Duyarlılık, bir sınıftaki öğelerin ne kadarının doğru bir şekilde tahmin edildiği kavramını temsil eder. F-ölçüsü, keskinlik ve duyarlılığın harmonik ortalamasıdır ve aşağıdaki şekilde hesaplanır.

$$F = 2 \times \frac{keskinlik \times duyarlılık}{keskinlik + duyarlılık}$$

Denklem 4.1: F1 Skor formülü [60].

Sınıflandırma metotlarının performansını değerlendirmek için kullanılan bir diğer ölçümleme terimi ise karmaşıklık matrisidir. Karmaşıklık matrisi tablosunda bir sınıflandırma probleminde gerçekleşen ve tahminlenen değerler gösterilmektedir.



Şekil 4.1: Karmaşıklık matrisi [60].

$$Keskinlik\ (Precision) = \frac{True\ Positive}{True\ Positive + False\ Positive}$$
$$Duyarlılık\ (Recall) = \frac{True\ Positive}{True\ Positive + False\ Negative}$$
$$Accuracy\ (Doğruluk) = \frac{True\ Positive + True\ Negative}{TP + FP + TN + FN}$$

Denklem 4.2: Keskinlik, duyarlılık ve doğruluk formülleri [60].

Kesinlik, Duyarlılık ve F1 Puanı değerleri:

Öncelikle Naive Bayes algoritması kullanılarak alınmış F-ölçüsü, keskinlik ve duyarlılık değerleri, kelime çantası ve terim frekans-ters belge sıklığı yöntemleri ile ayrı ayrı hesaplanarak değerleri aşağıdaki tablolarda gösterilmiştir.

Bag of Words

	precision	recall	f1-score	support
0	0.91	0.89	0.90	36
1	0.67	0.50	0.57	4
2	0.93	0.97	0.95	59
accuracy			0.92	99
macro avg	0.84	0.78	0.81	99
weighted avg	0.92	0.92	0.92	99

Tablo 4.3: Navie Bayes- Kelime çantası sonuçları.

TF-IDF

	precision	recall	f1-score	support
0	0.89	0.86	0.87	36
1	0.50	0.25	0.33	4
2	0.92	0.97	0.94	59
accuracy			0.90	99
macro avg	0.77	0.69	0.72	99
weighted avg	0.89	0.90	0.89	99

Tablo 4.4: Navie Bayes- Terim frekans-Ters belge sıklığı yöntemi sonuçları.

Naive Bayes sınıflandırıcısının uygulaması kolaydır ve az miktarda eğitim verisi Naive Bayes için yeterlidir. Boyutu yüksek olan verilerde de iyi çalışabilir. Hızlı olduğu için gerçek zamanlı sistemlerde kullanılabilir. Kesikli ve sürekli veriler ile kullanılabilmektedir. Gerçek hayatta değişkenler birbiri ile bağımlıdır ancak Naive Bayes'te öznitelikler birbirinden bağımsız kabul edildiği için değişkenler arasındaki ilişkiler modellenememektedir.

Bag of Words

	precision	recall	f1-score	support
0	0.97	0.94	0.96	36
1	1.00	0.25	0.40	4
2	0.94	1.00	0.97	59
accuracy			0.95	99
macro avg	0.97	0.73	0.77	99
weighted avg	0.95	0.95	0.94	99

Tablo 4.5: Lojistik Regresyon - Kelime çantası sonuçları.

TF-IDF

	precision	recall	f1-score	support
0	0.94	0.89	0.91	36
1	0.00	0.00	0.00	4
2	0.91	1.00	0.95	59
accuracy			0.92	99
macro avg	0.62	0.63	0.62	99
weighted avg	0.88	0.92	0.90	99

Tablo 4.6: Lojistik Regresyon - Terim frekans-Ters belge sıklığı yöntemi sonuçları.

Lojistik regresyonun uygulanması ve yorumlanması daha kolaydır ve eğitilmesi çok verimlidir. Gözlem sayısı, öznitelik sayısından fazla olması lojistik regresyonun daha iyi sonuç vermesini sağlayacaktır. Ayrıca lojistik regresyon, sadece bir tahmin edicinin (katsayı boyutu) ne kadar uygun olduğunun bir ölçüsünü sağlamakla kalmaz, aynı zamanda ilişki yönünü de (pozitif veya negatif) sağlar. Yalnızca ayrık işlevleri tahmin etmek için kullanılabilir. Bu nedenle, lojistik regresyonun bağımlı değişkeni ayrık sayı kümesine bağlıdır. Bilinmeyen kayıtları sınıflandırmada çok hızlıdır. Birçok basit veri kümesi için doğruluk değeri sonucu iyi hesaplanmaktadır ve veri kümesi doğrusal olarak ayrılabilir olduğundan iyi performans göstermektedir.

Bag of Words

	precision	recall	f1-score	support
0	0.00	0.00	0.00	36
1	0.00	0.00	0.00	4
2	0.60	1.00	0.75	59
accuracy			0.60	99
macro avg	0.20	0.33	0.25	99
weighted avg	0.36	0.60	0.45	99

Tablo 4.7: Destek Vektör Makinesi - Kelime çantası sonuçları.

TF-IDF

	precision	recall	f1-score	support
0	0.00	0.00	0.00	36
1	0.00	0.00	0.00	4
2	0.60	1.00	0.75	59
accuracy			0.60	99
macro avg	0.20	0.33	0.25	99
weighted avg	0.36	0.60	0.45	99

Tablo 4.8: Destek Vektör Makinesi - Terim frekans-Ters belge sıklığı yöntemi sonuçları.

Destek vektör makineleri karmaşık karar sınırlarını modelleyebilmektedir ve karar aşaması anında bir alt eğitim noktası kümesi kullandığından dolayı daha az bellek kullanmaktadırlar. Ayrıca net bir ayırma marjı ve yüksek boyutsal boşlukla daha iyi çalışmaktadır. Destek vektör makinelerinin dezavantajı ise, eğitim süresinin uzun olması sebebi ile büyük veri kümeleri için uygun değildir ve Naive Bayes ile karşılaştırıldığında, Naive Bayes'e kıyas ile model eğitimi sırasında eğitimde daha fazla zaman harcamaktadır. Çakışan sınıflar ile kötü çalışmaktadır ve ayrıca kullanılan çekirdek türüne karşı duyarlıdır.

Bag of Words

	precision	recall	f1-score	support
0	1.00	0.75	0.86	36
1	0.75	0.75	0.75	4
2	0.87	1.00	0.93	59
accuracy			0.90	99
macro avg	0.87	0.83	0.85	99
weighted avg	0.91	0.90	0.90	99

Tablo 4.9: Karar Ağacı - Kelime çantası sonuçları.

TF-IDF

	precision	recall	f1-score	support
0	0.96	0.72	0.83	36
1	1.00	0.75	0.86	4
2	0.84	0.98	0.91	59
accuracy			0.88	99
macro avg	0.93	0.82	0.86	99
weighted avg	0.89	0.88	0.87	99

Tablo 4.10: Karar Ağacı - Terim frekans-Ters belge sıklığı yöntemi sonuçları.

Karar ağaçlarını yorumlamak kolaydır ve anlaşılabilir kurallar oluşturulabilmektedir. Ayrık ve sürekli nitelikli değerler için kullanılabilir. Sınıfların sayısı normalden fazla ise modelin oluşturulma başarısı düşmektedir. Ayrıca öğrenme kümesinin örnek sayıları da normalden az olduğu zamanda da modellerin oluşturulma başarıları düşmektedir. Zaman ve yer karmaşıklığı için de öğrenme kümesi örneklerinin sayısına, nitelik sayısına ve oluşan ağacın yapısına bağlı olarak değişmektedir.

Bag of Words

	precision	recall	f1-score	support
0	0.93	0.72	0.81	36
1	0.40	0.50	0.44	4
2	0.86	0.97	0.91	59
accuracy			0.86	99
macro avg	0.73	0.73	0.72	99
weighted avg	0.87	0.86	0.86	99

Tablo 4.11: Rastgele Orman - Kelime çantası sonuçları.

TF-IDF

	precision	recall	f1-score	support
0	0.96	0.67	0.79	36
1	0.25	0.25	0.25	4
2	0.84	1.00	0.91	59
accuracy			0.85	99
macro avg	0.68	0.64	0.65	99
weighted avg	0.86	0.85	0.84	99

Tablo 4.12: Rastgele Orman - Terim frekans-Ters belge sıklığı yöntemi sonuçları.

Rastgele orman, makine öğrenmesi sistemlerinin çoğunu oluşturan sınıflandırma ve regresyon problemlerinde kullanılabilmektedir. Ayrıca kullanışlıdır. Kullanımı ise kolay olan algoritmadır. Kısa sürede model geliştirilmesi istenildiği zaman, bu algoritma iyi bir seçimdir. Varsayılan hiperparametreler ile genelde iyi tahmin sonucları oluşturmaktadır ve hiperparametrelerin sayısı yüksek olmayıp anlaşılması kolaydır. Random forest algoritması hızlı eğitilebilir. Ancak bir kez eğitildikten sonra tahminleri oluşturmak için yavaştır.

Bag of Words

	precision	recall	f1-score	support
0	1.00	0.28	0.43	36
1	1.00	0.50	0.67	4
2	0.68	1.00	0.81	59
accuracy			0.72	99
macro avg	0.89	0.59	0.64	99
weighted avg	0.81	0.72	0.67	99

Tablo 4.13: Arttırma - Kelime çantası sonuçları.

TF-IDF

	precision	recall	f1-score	support
0	1.00	0.28	0.43	36
1	1.00	0.50	0.67	4
2	0.68	1.00	0.81	59
accuracy			0.72	99
macro avg	0.89	0.59	0.64	99
weighted avg	0.81	0.72	0.67	99

Tablo 4.14: Arttırma - Terim frekans-Ters belge sıklığı yöntemi sonuçları.

Öncesinde de bahsedildiği üzere birden çok arttırım algoritmaları vardır; XGBoost, GradientBoost, AdaBoost ve CatBoost. Tez kapsamındaki çalışmada kullanılan arttırım algoritması ise AdaBoost algoritmasıdır. Adaboost algoritması farklı zayıf algoritmaları bir araya getirilerip karma makine öğrenmesi gerçekleştirmektedir. Bu sayede ise daha güçlü bir model oluşturmaktadır. Kendisinden önce geliştirilmiş çeşitli makine öğrenmesi algoritmalarını karşılaştırıp ve birleştirip öğrenmede iyi sonuca ulaşılmasını hedeflemektedir.

Bag of Words

	precision	recall	f1-score	support
0	0.66	0.53	0.58	36
1	0.00	0.00	0.00	4
2	0.71	0.85	0.78	59
accuracy			0.70	99
macro avg	0.46	0.46	0.45	99
weighted avg	0.66	0.70	0.67	99

Tablo 4.15: K-En Yakın Komşu - Kelime çantası sonuçları.

TF-IDF

	precision	recall	f1-score	support
0	0.72	0.50	0.59	36
1	0.00	0.00	0.00	4
2	0.72	0.90	0.80	59
accuracy			0.72	99
macro avg	0.48	0.47	0.46	99
weighted avg	0.69	0.72	0.69	99

Tablo 4.16: K-En Yakın Komşu - Terim frekans-Ters belge sıklığı yöntemi sonuçları.

Diğer algoritmalara kıyasa K-NN algoritmasının eğitiminin olmaması ve gerçekleştiriminin daha kolay olması önemli avantajlarındandır. Ayrıca analitik olarak izlenebilir olması gibi avantajlarından dolayı sınıflandırma uygulamalarında tercih edilmektedir. Fakat yüksek

miktarda bellek alanına ihtiyaç duymaktadır. Ayrıca veri kümesi ve öznitelik boyutları arttıkça işlem yüklerinin ve maliyetlerinin önemli boyutta yükseldiği gözlenmektedir. Performans k komşu sayısına, uzaklık ölçütüne ve öznitelik sayısına bağlı olarak değişmektedir. Bu gibi parametreler ve özelliklerle etkilenmesi söz konusudur.

Bag of Words

	precision	recall	f1-score	support
0	0.94	0.86	0.90	36
1	0.00	0.00	0.00	4
2	0.89	1.00	0.94	59
accuracy			0.91	99
macro avg	0.61	0.62	0.61	99
weighted avg	0.87	0.91	0.89	99

Tablo 4.17: Çok Katmanlı Algılayıcı - Kelime çantası sonuçları.

TF-IDF

	precision	recall	f1-score	support
0	0.00	0.00	0.00	36
1	0.00	0.00	0.00	4
2	0.60	1.00	0.75	59
accuracy			0.60	99
macro avg	0.20	0.33	0.25	99
weighted avg	0.36	0.60	0.45	99

Tablo 4.18: Çok Katmanlı Algılayıcı - Terim frekans-Ters belge sıklığı yöntemi sonucu.

Çok katmanlı, sinir ağlarının çoğu derin öğrenmeyi de etkilemektedir. Bir veya birden çok gizli katman kullanılabilmektedir. Zor ve karmaşık problemler için kullanılabilir. Örnekleri kullanarak öğrenirler. Daha önce karşılaşmamış örnekler hakkında bilgi verebilirler. Bazen uzun eğitim zamanı gereklidir. Optimizasyon zordur.

5. TARTIŞMA VE SONUÇ

Log sınıflandırmak için birden çok algoritma kullanılmaktadır. Tez kapsamında ise şirket tarafından sağlanan logları sınıflandırabilmek için Naive Bayes, destek vektör makineleri, lojistik regresyon, karar ağacı, arttırım, k-en yakın komşu ve çok katmanlı algılayıcı kullanılarak sonuçları karşılaştırılmıştır.

İlk başta kelime çantası ve terim frekans-ters belge sıklığı olmak üzere iki ayrı yöntem denenmiş olup sonuçlar gözlemlenmiştir ve k-en yakın komşu harici kullanılan tüm algoritmalarda doğruluk değerleri, kelime çantası yöntemi ile çalışıldığında daha yüksek olduğu görülmüştür. Kelime çantası kullanıldığı durum da her ne kadar k-en yakın komşunun doğruluk değerinde düşüş denilse de yaklaşık %2 oranda düşüş olmuştur. En çok göze çarpan ise kelime çantası kullanıldığında çok katmanlı algılayıcının doğruluk değeri %59,59 dan %90,91 e çıkmış olup yaklaşık %31 oranda artış gözlenmiştir. Son durumda tez çalışmasında kelime çantası yöntemi kullanılmaya karar verilmiştir.

Kelime çantası kullanıldığı durumda doğruluk değeri yüksekten düşüğe doğru algoritma sonuçları şu şekilde olmuştur: Lojistik Regresyon (%94,95), Navie Bayes (%91,92), Çok Katmanlı Algılayıcı (%90,91), Karar Ağacı (%89,90), Rastgele Orman (%85,86), Arttırma (%71,72), K-En Yakın Komşu (%69,70) ve Destek Vektör Makinesi (%59,60).

Destek vektör makinesi büyük veri kümeleri için uygun değildir ve model eğitimi sırasında da eğitim anında çok fazla zaman harcadığı için tercih edilmemiştir. Ayrıca çakışan sınıflar ile çalıştığında kötü performans sergilemektedir. Çalışmada kullanılan algoritmalar arasında %59.60 ile en düşük sonucu veren algoritma olmuştur.

Benzer şekilde k-en yakın komşu algoritması ile deneme yapıldığında veri seti boyutu arttırıldıkça maliyetin yükseldiği fark edilmiştir. Ancak destek vektör makinesi ile karşılaştırıldığında yaklaşık %10 daha başarılı olduğu görülmüştür.

Adabost algoritması ile çeşitli zayıf algoritmalar bir araya getirilip güçlü bir model sunulması hedeflenmiş ise de bu çalışma kapsamında istenilen sonuca erişmekte yeterli olmamıştır.

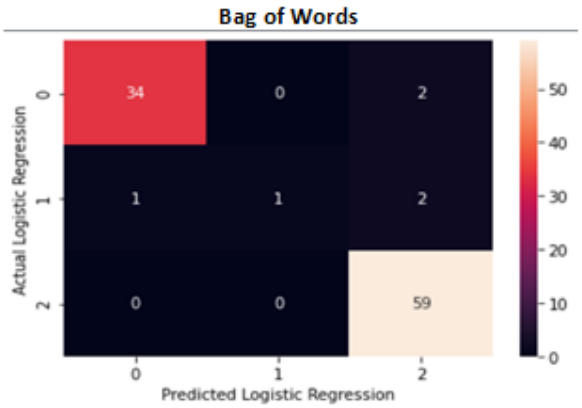
Rastgele orman ile kısa sürede model geliştirebilmektedir ve hızlı bir şekilde eğitme de kullanılmaktadır. Fakat bir kez eğitildikten sonra daha öncesinde de bahsedildiği üzere tahminleri oluştururken yavaştır. Yaklaşık yüzde %85 başarılı ve ilk başta hızlı olarak görülse de sonrasındaki tahminleri oluştururken yavaştır.

Karar ağaçlarını yorumlamak kolay olup ayrık ve sürekli değerlerle ile birlikte kullanabilmektedir ve yaklaşık %90 başarılı olmuştur. Rastgele orman ile karşılaştırıldığında ise yaklaşık %5 daha başarılı olduğu çalışma kapsamında görülmüştür.

Çok katmanlı algılayıcılar, karmaşık problemlerde kullanılan ve iyi sonuç veren algoritmadır. Örnekleri ise kullanarak öğrenmektedir. Daha öncesinde karşılaşmamış örnekler hakkında da bilgi verebilmektedir. Eğitim sırasında diğerlerine kıyasla daha uzun zaman almaktadır. Çalışma kapsamında karar ağacı ile yaklaşık aynı sonuç vermektedir. Karar ağacında sınıf sayısında artış olduğunda ve öğrenme küme örnekleri sayısı az olduğunda çok katmanlı algılayıcılara göre model oluşturma başarısı daha çok düşmektedir.

Bir diğer algoritmalardan biri olan Naive Bayes, çalışma esnasında çok kolay uygulanmıştır ve ilk başta az miktarda eğitim serisi ile çalışıldığında iyi sonuç vermiştir. Sonrasında boyutu yüksek olan verilerde de iyi çalıştığı görülmüştür ve %91,92 ile en iyi sonuç veren ikinci algoritma olmuştur.

Tez çalışmasında, %94,95 ile lojistik regresyonun en yüksek sonuç vermiştir. Diğer tüm yöntemler ile karşılaştırıldığında hem uygulanması hem de yorumlanması çok kolay olmuştur. Ayrıca gözlem sayısı, öznitelik sayısından fazla olması daha da iyi sonuçlar vermesini sağlayacaktır.



Şekil 5.1: Lojistik regresyon sonuç.

Lojistik regresyonun değerinin yüksek çıkması ile beraber kesinlik, duyarlılık, F1 puanları da incelenmiştir ve bu sonuçların da diğer algoritma sonuçlarına göre iyi olduğu gözlenmiştir.

Bag of Words

	precision	recall	f1-score	support
0	0.97	0.94	0.96	36
1	1.00	0.25	0.40	4
2	0.94	1.00	0.97	59
accuracy			0.95	99
macro avg	0.97	0.73	0.77	99
weighted avg	0.95	0.95	0.94	99

Tablo 5.1: Lojistik Regresyon sonuç.

Çapraz doğrulama değerleri de karşılaştırılmıştır ve yakındır. Son durumda şirket loglarının sınıflandırılması için lojistik regresyon kullanılmıştır.

Sonuç olarak tez kapsamında makine öğrenmesi yöntemleri kullanarak logların sınıflandırılması çalışması yapılmıştır. Ayrıntılı olarak veri analiz çalışmalarında bulunulmuştur. Veri seti setine dillerin yapısal özelliklerine uygun olacak şekilde çeşitli doğal dil işleme adımları uygulanılmıştır ve detaylı bir ön işleme çalışması yapılmıştır. Ön işleme çalışmaları ile birlikte terim frekans-ters belge sıklığı ve kelime çantası yöntemleri de kullanarak özellik çıkarımında bulunuldu. Bütün bu adımlar tamamlandıktan sonra Navie Bayes, lojistik regresyon, karar ağacı, destek vektör makineleri, rastgele orman, arttırma algoritması olan AdaBoost, k-en yakın komşu ve çok katmanlı algılayıcı algoritmaları kullanıldı ve bu algoritmalar matematiksel temellere dayandırılmak suretiyle detaylı bir şekilde incelendi. Algoritmalardan elde edilen sonuçlar değerlendirildi ve analiz edildi. Bu sonuçları destekleyici grafik ve tablolar eklendi. Elde edilen sonuçlar ışığında, sınıflandırma algoritmalarında ön işleme adımları yapıldığında performansın arttığı ortaya konuldu. Başarı oranı açısından lojistik regresyon diğer algoritmalardan daha yüksek bir başarı elde ettiği gözlemlendi.

Tez çalışmasında yapılan çalışmalardan farklı dataset kullanılıp, 8 farklı algoritma yöntemi uygulanıp, her bir yöntemde daha iyi sonuç alabilmek için parametreler değiştirilip, max sonuç alınması hedeflendi. Ayrıca verilerin çeşitli ön işlemelerden geçirilip optimal sonuçlar alınması üzerine çalışıldı. Tez çalışması benzer çalışmalara geçerli sonuçlar açısından yol

gösterici olacaktır ve farklı datasetler ile de çalışıp, sonuçları incelenebilir. Çalışmada kullanılan veri setine makine öğrenmesi uygulaması yanı sıra derin öğrenmede uygulanabilir. Sonuçların görselleştirilmesi üzerine ek çalışmalarda bulunulabilir.

KAYNAKLAR

[1]. Özseven, T., & Düğenci, M. (2011). Log Analiz: Erişim Kayıt Dosyaları Analiz Yazılımı ve GOP Üniversitesi Uygulaması. Bilişim Teknolojileri Dergisi, 4(2).

[2]. Jans, M., Soffer, P., & Jouck, T. (2019). Building a valuable event log for process mining: an experimental exploration of a guided process. Enterprise Information Systems, 13(5), 601-630.

[3]. Kaya, Y. & Yeşilova, A. (2009). Yapay Sinir Ağı ve Lojistik Regresyon Kullanılarak Kategorik Verilerin Modellenmesi. Yüzüncü Yıl Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 14(2), 112-116.

[4]. Biau, G., & Scornet, E. (2016). A random forest guided tour. Test, 25(2), 197-227.

[5]. Bhatia, N. (2010). Survey of nearest neighbor techniques. arXiv preprint arXiv:1007.0085.

[6]. Çelik, Ö., & Koç, B. C. (2020). TF-IDF, Word2vec ve Fasttext Vektör Model Yöntemleri ile Türkçe Haber Metinlerinin Sınıflandırılması. Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi, 23(67), 121-127.

[7]. Kotiyal, B., Kumar, A., Pant, B., & Goudar, R. H. (2014). Classification Technique for Improving User Access on Web Log Data. In Intelligent Computing, Networking, and Informatics (pp. 1089-1097). Springer, New Delhi.

[8]. Çalış, A., Kayapınar, S., & Çetinyokuş, T. (2014). Veri Madenciliğinde Karar Ağacı Algoritmaları İle Bilgisayar Ve İnternet Güvenliği Üzerine Bir Uygulama. Endüstri Mühendisliği, 25(3), 2-19.

[9]. Farnaaz, N., & Jabbar, M. A. (2016). Random forest modeling for network intrusion detection system. Procedia Computer Science, 89, 213-217.

[10]. Pranckevičius, T., & Marcinkevičius, V. (2017). Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. Baltic Journal of Modern Computing, 5(2), 221.

[11]. Şahin, G. (2017). Turkish document classification based on Word2Vec and SVM classifier. In 2017 25th signal processing and communications applications conference (SIU) (pp. 1-4). IEEE.

[12]. Kutlugün, M. A., Çakır, M. Y., & Kıanı, F. (2017). Yapay Sinir Ağları ve K-En Yakın Komşu Algoritmalarının Birlikte Çalışma Tekniği (Ensemble) ile Metin Türü Tanıma.

[13]. Atalay, M., & Çelik, E. (2017). Büyük veri analizinde yapay zekâ ve makine öğrenmesi uygulamalari-artificial intelligence and machine learning applications in big data analysis. Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 9(22), 155-172.

[14]. Ren, R., Cheng, J., Yin, Y., Zhan, J., Wang, L., Li, J., & Luo, C. (2018, December). Deep convolutional neural networks for log event classification on distributed cluster systems. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 1639-1646). IEEE.

[15]. Yıldırım, S., & Yıldız, T. (2018). Türkçe için karşılaştırmalı metin sınıflandırma analizi. Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 24(5), 879-886.

[16]. Aytekin, Ç., Sütçü, C. S., & Özfidan, U. (2018). Karar ağacı algoritması ile metin sınıflandırma: müşteri yorumları örneği. Uluslararası Sosyal Araştırmalar Dergisi, 11(55), 1307-9581.

[17]. Şahin, E. (2018). Makine Öğrenme Yöntemleri ve Kelime Kümesi Tekniği ile İstenmeyen e-Postasınıflandırması. Hacettepe Üniversitesi.

[18]. Ahmad, I., Basheri, M., Iqbal, M. J., & Rahim, A. (2018). Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE access, 6, 33789-33795.

[19]. Vanneschi, L., Horn, D. M., Castelli, M., & Popovič, A. (2018). An artificial intelligence system for predicting customer default in e-commerce. Expert Systems with Applications, 104, 1-21.

[20]. Küçük, H., Eminoğlu, İ., & Balcı, K. (2019). Classification of neuromuscular diseases with artificial intelligence methods. Journal of the Faculty of Engineering and Architecture of Gazi University, 34(4), 1725-1741.

[21]. Nguyen, H., & Bui, X. N. (2019). Predicting blast-induced air overpressure: a robust artificial intelligence system based on artificial neural networks and random forest. Natural Resources Research, 28(3), 893-907.

[22]. Çelenli, H. İ., Öztürk, S. T., Şahin, G., Gerek, A., Ganiz, M. C. (2018). Document Embedding Based Supervised Methods for Turkish Text Classification. In 2018 3rd International Conference on Computer Science and Engineering (UBMK) (pp. 477-482). IEEE.

[23]. Sarı, M., Özbayoğlu, A. M. (2018). Classification of Turkish Documents Using Paragraph Vector. In 2018 International Conference on Artificial Intelligence and Data Processing (IDAP) (pp. 1-5). IEEE.

[24]. Roy, A., Cruz, R. M., Sabourin, R., & Cavalcanti, G. D. (2018). A study on combining dynamic selection and data preprocessing for imbalance learning. Neurocomputing, 286, 179-192.

[25]. Sen, M. U., & Yanıkoğlu, B. (2018, May). Document classification of SuDer Turkish news corpora. In 2018 26th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.

[26]. Korkmaz, A., & Büyükgöze, S. (2019). Sahte Web Sitelerinin Sınıflandırma Algoritmaları İle Tespit Edilmesi. Avrupa Bilim ve Teknoloji Dergisi, (16), 826-833.

[27]. Kazan, S., & Karakoca, H. (2019). Makine Öğrenmesi ile Ürün Kategorisi Sınıflandırma. Sakarya University Journal of Computer and Information Sciences, 2(1), 18-27.

[28]. Acı, Ç. & Çırak, A. (2019). Türkçe haber metinlerinin konvolüsyonel sinir ağları ve Word2Vec kullanılarak sınıflandırılması. Bilişim Teknolojileri Dergisi, 12(3), 219-228.

[29]. Aydoğan, M., & Karcı, A. (2019). Makine Öğrenmesi Ve Transfer Öğrenme İle Türkçe Metin Sınıflandırma Turkish Text Classification with Machine Learning and Transfer Learning.

[30]. Demirci, M. (2019). Destek Vektör Makineleri ve M5 Karar Ağacı Yöntemleri Kullanılarak Yağış Akış İlişkisinin Tahmini. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 10(3), 1113-1124.

[31]. Yalçınkaya, B., Gedikli, M. E., & Cibaroğlu, M. O. (2019). Elektronik Belge Yönetim Sisteminde Log Analizi: İstatistiksel Bir Değerlendirme.

[32]. Karcioğlu, A. A., Aydin, T. (2019). Sentiment Analysis of Turkish and English Twitter Feeds Using Word2Vec Model. In 2019 27th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.

[33]. Deniz, E., Erbay, H., Coşar, M. (2019). Classification of Turkish E-Mails with Doc2Vec. In 2019 1st International Informatics and Software Engineering Conference (UBMYK) (pp. 1-4). IEEE.

[34]. Erşahin, B., Aktaş, Ö., Kilinc, D., Erşahin, M. (2019). A hybrid sentiment analysis method for Turkish. Turkish Journal of Electrical Engineering & Computer Sciences, 27(3), 1780-1793.

[35]. Sel, İ., Karci, A., Hanbay, D. (2019, September). Feature Selection for Text Classification Using Mutual Information. In 2019 International Artificial Intelligence and Data Processing Symposium (IDAP) (pp. 1-4). IEEE.

[36]. Erdinç, H. Y., Güran, A. (2019). Semi-supervised Turkish Text Categorization with Word2Vec, Doc2Vec and FastText Algorithms. In 2019 27th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.

[37]. Korkmaz, G. (2020). Yapay Zekâ Yöntemleriyle Sınıflandırma Ve Finans Sektöründe Bir Uygulama. Akademik Yaklaşımlar Dergisi, 11(2), 91-109.

[38]. Tuzcu, S. (2020). Çevrimiçi Kullanıcı Yorumlarının Duygu Analizi ile Sınıflandırılması. Eskişehir Türk Dünyası Uygulama ve Araştırma Merkezi Bilişim Dergisi, 1(2), 1-5.

[39]. Beşli, N., & Tenekeci, E. (2020). Uydu verilerinden karar ağaçları kullanarak orman yangını tahmini. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 11(3), 899-906.

[40]. Aylak, B. L., Oral, O., & Yazıcı, K. (2021). Yapay Zeka ve Makine Öğrenmesi Tekniklerinin Lojistik Sektöründe Kullanımı. El-Cezeri Journal of Science and Engineering, 8(1), 74-93.

[41]. Gacar, B. K., & Kocakoç, İ. D. (2020). Regresyon Analizleri mi Karar Ağaçları mı?. Manisa Celal Bayar Üniversitesi Sosyal Bilimler Dergisi, 18(4), 251-260.

[42]. Çelik, Ö., & Koç, B. C. TF-IDF, Word2vec ve Fasttext Vektör Model Yöntemleri ile Türkçe Haber Metinlerinin Sınıflandırılması. Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi, 23(67), 121-127.

[43]. Taşçı, G., & Çelebi, M. (2020). Eğitimde yeni bir paradigma: "Yükseköğretimde yapay zeka". OPUS Uluslararası Toplum Araştırmaları Dergisi, 16(29), 2346-2370.

[44]. Vijayakumar, T., & Vinothkanna, R. (2020). Capsule network on font style classification. Journal of Artificial Intelligence, 2(02), 64-76.

[45]. Aksu, M. Ç., & Karaman, E. (2020). FastText ve Kelime Çantası Kelime Temsil Yöntemlerinin Turistik Mekanlar İçin Yapılan Türkçe İncelemeler Kullanılarak Karşılaştırılması. Avrupa Bilim ve Teknoloji Dergisi, (20), 311-320.

[46]. Kılınç, M., Tarhan, Ç., & Aydın, C. Kitle Fonlaması Projelerinin Karar Ağacı ve Rastgele Orman Algoritmalarıyla Sınıflandırılması. Journal of Information Systems and Management Research, 2(2), 16-25.

[47]. Santur, Y. (2020). Derin Öğrenme ve Aşağı Örnekleme Yaklaşımları Kullanılarak Duygu Sınıflandırma Performansının İyileştirilmesi. Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 32(2), 561-570.

[48]. Korkmaz, T., Çetinkaya, A., Aydın, H., & Barışkan, M. A. (2021). Analysis of whether news on the Internet is real or fake by using deep learning methods and the TF-IDF algorithm. International Advanced Researches and Engineering Journal, 5(1), 31-41.

[49]. Le Berre, C., Sandborn, W. J., Aridhi, S., Devignes, M. D., Fournier, L., Smaïl-Tabbone, M., Danese, S., & Peyrin-Biroulet, L. (2020). Application of artificial intelligence to gastroenterology and hepatology. Gastroenterology, 158(1), 76-94.

[50]. Benli, T., & Kızgın, Y. (2020). Schwarzt'ın sosyal değerler modelindeki bireysel değerlere göre tüketicilerin çevreye duyarlı tüketim davranışlarının isimsel lojistik regresyon analizi ile incelenmesi. Manas Sosyal Araştırmalar Dergisi, 9(2), 995-1009.

[51]. Albayrak, A. (2020). Doğal Dil İşleme Teknikleri Kullanılarak Disiplinler Arası Lisansüstü Ders İçeriği Hazırlanması. Bilişim Teknolojileri Dergisi, 13(4), 373-383.

[52]. Güler, G., Tantuğ, A. C. (2020). Comparison of Turkish Word Representations Trained on Different Morphological Forms. arXiv preprint arXiv:2002.05417.

[53]. Tuncel, C. O., & Oktay, D. (2021). Analysis of Determinants of Innovation in Small Businesses with Decision Tree. International Conference on Economics.

[54]. Güran, A., & Kınık, D. (2021). TF-IDF ve Doc2Vec Tabanlı Türkçe Metin Sınıflandırma Sisteminin Başarım Değerinin Ardışık Kelime Grubu Tespiti ile Arttırılması. Avrupa Bilim ve Teknoloji Dergisi, (21), 323-332.

[55]. R-bloggers, 2014, The TF-IDF Statistic For Keyword Extraction, Available: http://www.r-bloggers.com/2014/02/the-tf-idf-statistic-for-keyword-extraction/, [Ziyaret tarihi: 15 Aralık 2020].

[56]. Devhunter, 2019, Naive-Bayes Sınıflandırıcı, Available: http://devhunteryz.wordpress.com/2019/12/02/naive-bayes-siniflandirici/, [Ziyaret tarihi: 02 Ocak 2021].

[57]. Veribilimcisi, 2017, Lojistik Regresyon, Available: https://veribilimcisi.com/2017/07/18/lojistik-regresyon/ [Ziyaret tarihi: 20 Aralık 2020].

[58]. Akça, M.F., 2020, Nedir Bu Destek Vektör Makineleri?, Available:

https://medium.com/deep-learning-turkiye/nedir-bu-destek-vekt%C3%B6r-makineleri-makine-%C3%B6%C4%9Frenmesi-serisi-2-94e576e4223e, [Ziyaret tarihi: 12 Şubat 2021].

[59]. Ulgen, E.K., 2017, Destek Vektör Makineleri, Available: https://medium.com/@k.ulgen90/makine-%C3%B6%C4%9Frenimi-b%C3%B6l%C3%BCm-4-destek-vekt%C3%B6r-makineleri-2f8010824054, [Ziyaret tarihi: 13 Şubat 2021].

[60]. Öğündür, G., 2019, Doğruluk (Accuracy) , Kesinlik(Precision) , Duyarlılık(Recall) ya da F1 Score, Available: https://medium.com/@gulcanogundur/do%C4%9Fruluk-accuracy-kesinlik-precision-duyarl%C4%B1l%C4%B1k-recall-ya-da-f1-score-300c925feb38, [Ziyaret tarihi: 05 Mart 2021].

EKLER