ÇAĞRI MERKEZİNE GELEN ARAMALARIN METİN TABANLI SINIFLANDIRILMASI

2024
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ

Muammer ÖZDEMİR

Tez Danışmanı
Dr. Öğr. Üyesi Yasin ORTAKCI

ÇAĞRI MERKEZİNE GELEN ARAMALARIN METİN TABANLI SINIFLANDIRILMASI

Muammer ÖZDEMİR

Tez Danışmanı
Dr. Öğr. Üyesi Yasin ORTAKCI

T.C.
Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalında
Yüksek Lisans Tezi
Olarak Hazırlanmıştır

KARABÜK
Ocak 2024

Muammer ÖZDEMİR tarafından hazırlanan "ÇAĞRI MERKEZİNE GELEN ARAMALARIN METİN TABANLI SINIFLANDIRILMASI" başlıklı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarım.

Dr. Öğr. Üyesi Yasin ORTAKCI .....
Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından Oy Birliği ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir. 12/01/2024

Ünvanı, Adı SOYADI (Kurumu)	İmzası
Başkan : Dr. Öğr. Üyesi Kürşat Mustafa KARAOĞLAN (KBÜ)	.....
Üye : Dr. Öğr. Üyesi Yasin ORTAKCI (KBÜ)	.....
Üye : Dr. Öğr. Üyesi Emel SOYLU (SAMÜ)	ONLİNE

KBÜ Lisansüstü Eğitim Enstitüsü Yönetim Kurulu, bu tez ile, Yüksek Lisans derecesini onamıştır.

Doç. Dr. Zeynep ÖZCAN .....
Lisansüstü Eğitim Enstitüsü Müdürü

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

Muammer ÖZDEMİR

ÖZET

Yüksek Lisans Tezi

ÇAĞRI MERKEZİNE GELEN ARAMALARIN METİN TABANLI SINIFLANDIRILMASI

Muammer ÖZDEMİR

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Dr. Öğr. Üyesi Yasin ORTAKCI
Ocak 2024, 58 sayfa

Bu tez çalışmasında, Ticaret Bakanlığı Çağrı Merkezine gelen çağrıların yönlendirilmesi gereken birimi tahmin etmek için çeşitli makine öğrenme algoritmaları ve farklı metin temsil yöntemleri beraber kullanılmış ve bu modellerin başarımları kıyaslanmıştır. Türkiye Ticaret Bakanlığı Çağrı Merkezi, uzman personel eksikliği ve yanlış yönlendirilen çağrılar nedeniyle çağrıları hızlı bir şekilde çözmekte zorluklarla karşılaşabilmektedir. Bu durum ülke ekonomisi üzerinde olumsuz sonuçlar doğurabilmektedir. Bu tez çalışması, Word2Vec, GloVe ve TF-IDF gibi metin temsil yöntemleri ve çeşitli makine öğrenmesi algoritmalarını detaylı bir şekilde inceleyerek, çağrıları en etkili şekilde uygun departmana yönlendirmenin yollarını belirlemeyi ve bu modellerin performansını karşılaştırmayı amaçlamaktadır. Word2Vec, GloVe ve TF-IDF metin temsil yöntemlerini kullanarak, K-En Yakın Komşu, Naive Bayes, Destek Vektör Makineleri, Adaptive Boosting, Karar Ağacı ve Rastgele Orman gibi çeşitli makine öğrenmesi algoritmalarını içeren kapsamlı bir analiz yapılmıştır.

Performans değerlendirmesi doğruluk, kesinlik, duyarlılık ve f1-skor gibi ölçütler kullanılarak gerçekleştirilmiştir. Deneysel sonuçlar, Rastgele Orman ve Word2Vec kombinasyonunun, eğitim ve çalışma zamanı açısından çağrıları yönlendirmeyi başarabilen en uygun model olduğunu göstermiştir.

Anahtar Sözcükler : Metin sınıflandırma, Çağrı merkezi, Word2Vec, GloVe, TF-IDF

Bilim Kodu : 92431

ABSTRACT

Master Thesis

TEXT BASED CLASSIFICATION OF INCOMING CALLS TO THE CALL CENTER

Muammer ÖZDEMİR

Karabük University
Institute of Graduate Programs
Department of Computer Engineering

Thesis Advisor:
Assist. Prof. Dr. Yasin ORTAKCI
January 2024, 58 pages

In this thesis, various machine learning algorithms and different text representation methods are used together to predict the department to which incoming calls to the Ministry of Trade Call Center should be routed and the performance of these models are compared. The Call Center of the Ministry of Trade in Türkiye has been facing challenges in resolving issues promptly due to a lack of skilled staff and misdirected calls. This has had negative consequences on the nation's economy. The main objective of this study is to thoroughly examine different machine learning algorithms and word representation techniques, including Word2Vec, GloVe, and TF-IDF, determine the most effective way to route calls to the appropriate department, and compare the performance of these models. Using Word2Vec, GloVe, and TF-IDF methods, we conduct a comprehensive analysis that incorporates various machine learning algorithms, such as K-Nearest Neighbors, Naive Bayes, Support Vector Machines, Adaptive Boosting, Decision Tree, and Random Forest. Performance evaluation is performed using metrics such as accuracy, precision, recall, and f1-score.

The results indicate that a combination of the Random Forest and Word2Vec is the optimal model that can manage to route calls in learning and running time.

Key Word **:** Text classification, Word2Vec, GloVe, TF-IDF, Call center
Science Code : 92431

TEŞEKKÜR

Bu tez çalışmasının planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren sayın hocam Dr. Öğr. Üyesi Yasin ORTAKCI'ya sonsuz teşekkürlerimi sunarım.

Eğitim hayatım boyunca maddi ve manevi desteğini esirgemeyen sevgili babama, desteğini hep yanımda hissettiğim sevgili anneme ve kardeşlerime, tezimin hazırlanma süresince göstermiş olduğu sabır ve desteklerinden dolayı sevgili eşime teşekkürlerimi sunarım.

Tezimi varlığı ile hayatımda en büyük destekçim olan biricik kızıma ithaf ederim.

İÇİNDEKİLER

Sayfa

Sayfa

ŞEKİLLER DİZİNİ

Sayfa

ÇİZELGELER DİZİNİ

Sayfa

KISALTMALAR DİZİNİ

KISALTMALAR

ANN	: Artificial Neural Networks (Yapay Sinir Ağları)
ADABOOST	: Ada Boost (Ada Boost Algoritması)
BOW	: Bag of Words (Kelime Torbası)
CBOW	: Continuous Bag of Words (Sürekli Kelime Torbası)
CNN	: Convolutional Neural Network (Evrişimli Sinir Ağı)
DDİ	: Doğal Dil İşleme
DT	: Decision Trees (Karar Ağaçları)
K-NN	: K Nearest Neighbor (K En Yakın Komşu)
LR	: Logistic Regression (Lojistik Regresyon)
LSTM	: Long Short – Term Memory (Uzun Kısa Süreli Bellek)
MİY	: Müşteri İlişkileri Yönetimi
MM	: Müşteri Memnuniyeti
NB	: Naive Bayes
NLTK	: Natural Language Tool Kit (Doğal Dil Araç Kiti)
RF	: Random Forest (Rastgele Orman)
SVC	: Support Vector Classifier (Destek Vektör Sınıflandırıcı)
SVM	: Support Vector Machines (Destek Vektör Makineleri)
TBÇM	: Ticaret Bakanlığı Çağrı Merkezi
TF-IDF	: Term Frequence-Inverse Document Frequence (Terim Frekansı-Ters Doküman Frekansı)

BÖLÜM 1

GİRİŞ

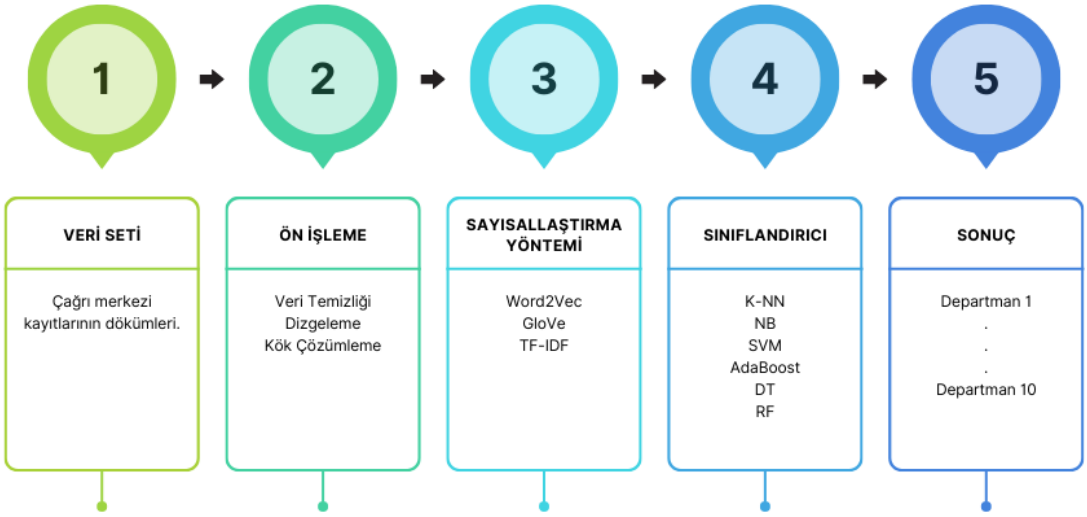
Teknolojide yaşanan hızlı gelişmeler ve internetin yaygın kullanılması, dijital dünyada veri çeşitliliğinin genişlemesini ve ele alınan verilerin büyük boyutlu olmasını beraberinde getirmiştir. Büyük verinin etkin kullanılması ile birlikte, bu verileri kullanan kurum ve kuruluşların veri üzerinde doğru analiz yapma ve anlamlı sonuçlar elde etme istekleri artmıştır [1]. Artan veri boyutlarının, anlamlı veriye dönüştürülmesi ve bu anlamlı verinin iş gücü maliyetlerinin azaltılması, enerji kaybının en aza indirilmesi ve zaman maliyetlerinin düşürülmesi gibi konularda yapay zeka ve makine öğrenmesi teknikleri büyük rol almaya başlamıştır. Başlıca makine öğrenmesi uygulamaları Şekil 1.1'de gösterilmiştir. Çağrı merkezleri de bu uygulama alanlarından biri olarak karşımıza çıkmaktadır.

Şekil 1. 1. Makine öğrenmesi uygulama alanları.

Çağrı merkezi, telefonla çok sayıda sorguyu almak veya iletmek için kullanılan, merkezi veya uzak olarak yönetilebilen bir etkileşim merkezidir. Kuruluş ve firmaların bulundukları sektörde tutunabilmeleri veya hizmetleri kesintisiz şekilde yürütebilmeleri için müşteriyi memnun etmek zorundadırlar. Bu amaçla, müşteri memnuniyetini sağlamanın kilit noktası, müşterilerin istek ve beklentilerinin yerine getirilmesidir. Bu hedefi gerçekleştirmek için kuruluşlar, müşterileriyle çeşitli iletişim kanalları aracılığıyla etkileşimde bulunmakta, müşteri istekleri, beklentileri ve tercihleri hakkında bilgi edinmeye çalışmaktadır. Bu iletişim kanallarından en yaygın ve etkili olanı çağrı merkezleridir [2]. Çağrı merkezleri, kuruluşların müşterilerine daha hızlı, daha kaliteli ve daha kişiselleştirilmiş hizmet sunmasını sağlar. Böylece, müşteri sadakati, memnuniyeti ve bağlılığı artar. Ayrıca, çağrı merkezleri, kuruluşların müşterilerinin ihtiyaçlarını, davranışlarını ve tercihlerini daha iyi anlamasına ve analiz etmesine yardımcı olur. Bu da, Müşteri İlişkileri Yönetimi'nin (MİY) daha etkin bir şekilde uygulanmasına imkan verir.

MİY, modern iş dünyasının vazgeçilmez bir unsuru haline gelmiştir. MİY, müşteri memnuniyetini sağlamak amacıyla doğru bilgi sunarak, hızlı soru cevaplayarak ve etkili bir şekilde hizmet taleplerine yanıt vererek müşterilere odaklanmaktadır. MİY'nin başarısı, büyük ölçüde çağrı merkezlerinin performansına bağlıdır. Son yıllarda, teknolojik ilerlemeler ve ticaret hacimlerindeki artışların bir sonucu olarak çağrı merkezi hizmetlerine olan talepte de bir artış olmuştur. Bununla birlikte, birçok şirket ve kurum, ilk soruları ele alacak nitelikli personel eksikliği nedeniyle müşteri ilişkileri yönetiminde zorluklarla karşılaşmaktadır [3]. Bu durum, sorunların çözümünün gecikmesine ve hatalı olmasına yol açarak daha uzun bekleme sürelerine ve müşterilerin basit sorunlar için bile tekrar tekrar arama yapmak zorunda kalmasına neden olmaktadır. Bu olumsuz sonuçlar, hem müşterilerin hem de çağrı merkezi temsilcilerinin memnuniyetsizliğine yol açmaktadır. Ayrıca, sık sık fazla iş yükü altında kalan çağrı merkezi temsilcileri aşırı strese maruz kalmakta ve öfkeli müşterilerle uğraşmak zorunda kalmaktadır [4,5]. Bu nedenle, çağrı merkezi temsilcilerinin performansını ve motivasyonunu artırmak için akıllı yardım sistemlerine acil ihtiyaç vardır. Bu faktörler, otomasyon ve akıllı destek araçlarının çağrı merkezlerine entegre edilmesinin öneminin altını çizmektedir.

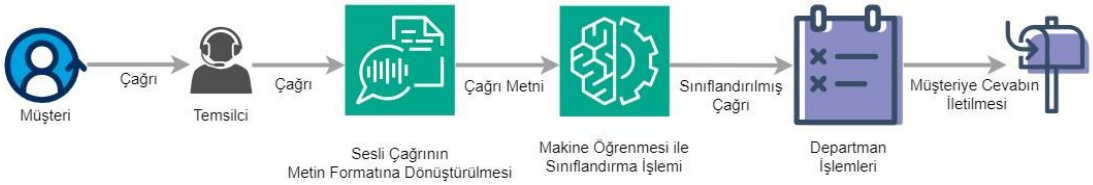
Bu tez çalışmasında, çağrı merkezlerinde karşılaşılan zorlukları ele almak amacıyla yapay zeka güdümlü bir metin sınıflandırma modeli geliştirilmiştir. Modelin uygulama aşamaları Şekil 1.2'de gösterilmiştir. Bu model, Türkiye Cumhuriyeti Ticaret Bakanlığı Çağrı Merkezi'ne (TBÇM) gelen çağrıları, çağrı metni dikkate alınarak ilgili birimlere otomatik yönlendirmektedir. TBÇM'ye günde ortalama 10.000 çağrı gelmekte ve bu çağrılar ilk olarak çağrı merkezi temsilcileri tarafından cevaplanmaktadır. Temsilciler çağrıyı kendileri mi ele alacaklarına yoksa ilgili birime mi yönlendireceklerine karar verirler. Ancak, çağrı içeriğinin çeşitliliği ve çağrı merkezi temsilcilerinin bilgi ve deneyim düzeylerinin farklılık göstermesinden dolayı, bazen müşterilere yanlış çözümler sunulabilmekte veya onları yanlış departmanlara yönlendirebilmektedirler. Bu hatalar, çözülemeyen sorunlara veya geciken çözümlere neden olmaktadır. Çağrıların çözüm süresinin uzaması örneğin sınır kapısında veya limanda ihraç olacak ve belirli bir raf ömrüne sahip eşyaların bozulmasına sebep olabilmektedir. Bu durum ihracatçı ve ülke için olumsuz ticaret görünümü oluşturmakta ve ülke ekonomisi üzerinde olumsuz sonuçlar doğurabilmektedir.



Şekil 1. 2. Metin çağrılarının sınıflandırılması için önerilen modelin aşamaları.

Bu sorunların üstesinden gelebilmek için çalışmada, gelen çağrıları TBÇM'deki ilgili departmanlara otomatik olarak yönlendirmek için çeşitli makine öğrenmesi tekniklerini kullanan yenilikçi bir çözüm sunulmuştur. Bu sayede, müşterilerin sorunlarının daha hızlı ve doğru bir şekilde çözülmesi ve müşteri memnuniyetinin artması hedeflenmektedir. Bunu başarmak için, TBÇM tarafından alınan 20.000 örnek

çağrı metninden ve bunların doğru yönlendirildiği departmanlardan oluşan bir veri kümesi kullanılmıştır. Bu metinler ön işlem süreçlerinden geçirilip, TF-IDF (Terim Frekansı - Ters Belge Frekansı), Word2Vec ve GloVe metin temsil teknikleri kullanılarak sayısal temsillere dönüştürülmüştür. Daha sonra bu sayısallaştırılmış çağrı metinlerini sınıflandırmak ve çağrıları uygun departmanlara yönlendirmek için K-En Yakın Komşu (K-NN), Destek Vektör Makineleri (SVM), Naive Bayes (NB), Karar Ağacı (DT), Adaptive Boosting (AdaBoost) ve Rastgele Orman (RF) dahil olmak üzere çeşitli makine öğrenmesi algoritmaları uygulanmıştır. Önerilen modelin genel iş akışı da Şekil 1.3'te gösterilmiştir.



Şekil 1. 3. Önerilen model genel iş akışı.

Bu tez çalışmasında önerilen modelin diğer modellerinden başlıca farklılıkları şunlardır:

- Bazı çağrı merkezi destek sistemleri sınırlı sınıf seçeneği sunarak tercih işlemi ile işlemlere devam etmektedir. Önerilen modelde ise, sistem ilgili sınıfa otomatik yönlendirme yapmaktadır.
- Diğer bir fark ise, anahtar kelime dinleme yöntemi kullanılmamasıdır. Çağrılardaki anahtar kelime dinlenerek sınıf eşleştirme işlemi yapılabilmektedir. Bu yöntem yanlış sınıflandırmalar yapabilmektedir. Önerilen modelde bir öğrenme işlemi gerçekleştirildiğinden sınıflandırma işlemi yüksek doğruluk oranları ile sonuçlanmaktadır.
- Önerilen modelin diğer çağrı merkezi sistemlerinden bir diğer farkı da, sürekli veri giriş ile kendini geliştirme imkanına sahip olmasıdır.

Deneyler boyunca, metin sınıflandırma algoritmaları ve metin sayısallaştırma yöntemleri üzerinde kapsamlı analizler gerçekleştirilmiştir. Öncelikli hedeflerden biri, TBÇM çağrı metnini sınıflandırmada üstün performans gösteren sınıflandırıcılar ve metin sayısallaştırma yöntemlerinin en etkili kombinasyonunu bulmaktır. Ek olarak,

aşağıdaki araştırma sorularının (AS) yanıtları araştırılmıştır: (AS1) Metin sınıflandırmada kullanılacak en uygun sınıflandırıcı hangisidir? (AS2) Hangi metin sayısallaştırma tekniği sınıflandırıcılar arasında daha iyi performans sağlar? (AS3) Hangi sınıflandırıcı ve metin sayısallaştırma yöntemi kombinasyonu en yüksek metin sınıflandırma sonuçlarını üretir? (AS4) Bu çözümleri uygun bir çalışma zamanıyla uygulamak mümkün müdür? Bu konulara ilişkin gözlem ve bulgular Bölüm 5.4'te ayrıntılı olarak ele alınmıştır. Sonuç olarak, bu çalışmanın temel katkıları aşağıdaki gibi özetlenebilir:

- TBÇM içerisinde telefon çağrılarının yönlendirilmesini kolaylaştırmak ve böylece çağrı merkezi operasyonlarının verimliliğini artırmak için tasarlanmış otomatik bir sistem önerilmektedir.
- Bu tez çalışması, TF-IDF, Word2Vec ve GloVe metin sayısallaştırma yöntemleri ile birlikte K-NN, AdaBoost, NB, DT, SVM ve RF gibi sınıflandırıcıları kullanarak Türkçe metinler üzerinde metin sınıflandırma performansının kapsamlı bir karşılaştırmalı analizini sunmaktadır.
- Önerilen yapay zeka tabanlı sistemin performansını artırmak için müşteri hizmetleri temsilcilerine bazı pratik bilgiler sağlanmıştır.

Bu tez çalışması 6 bölümden oluşmaktadır. Birinci bölümde, çalışma konusu ve kullanılan yöntemler hakkında bilgiler verilmiştir. Metin sınıflandırma, çağrı sınıflandırma konusunda daha önce yapılan çalışmalar sunulmuştur. İkinci bölümde; sınıflandırma, sınıflandırma türleri ve veri ön işleme yöntemleri anlatılmaktadır. Üçüncü bölümde, bu tez çalışmasının temel konularından biri olan metin sayısallaştırma yöntemleri hakkında bilgi verilmektedir. Dördüncü bölümde, tez çalışmasında TBÇM çağrılarının sınıflandırılması için kullanılan altı farklı sınıflandırma algoritması ayrıntılı bir şekilde açıklanmaktadır. Beşinci bölümde, TBÇM çağrılarının sınıflandırılmasında kullanılan çalışma ortamı, veri seti, değerlendirme metrikleri, deneysel çalışmalar ve sonuç analizi şekil ve çizelgelerle açıklanmaktadır. Altıncı ve son bölümde, uygulama sonucundaki performansların değerlendirilmesi yapılmaktadır. Bu değerlendirme sonucunda en optimum sonucu elde eden sınıflandırıcı ve kelime temsil yöntemi ikilisi önerilmektedir. Son olarak tezin literatüre yaptığı katkılara ve ileride yapılabilecek çalışmalara değinilmiştir.

1.1. LİTERATÜR TARAMASI

Müşteri Memnuniyeti (MM), müşterilerin şirketlerle kurdukları ilişkiler sonucunda ürün ve hizmetlerden aldıkları tatmin seviyelerini ölçmeye yönelik pazarlama araştırmalarının temel bir amacıdır [6,7]. Bilgi teknolojilerindeki gelişmeler, MM'nin değerlendirilmesinde yeni fırsatlar sunmuştur. Özellikle, MM'yi sağlamak için, birçok çalışma çağrı merkezi departmanlarının performansını geliştirmek için farklı yöntemler kullanmıştır. Örneğin, Park ve Gates çağrı metinlerinin analiziyle MM'yi otomatik olarak belirleme becerisini gösteren bir çalışma yapmıştır. Bu yöntem, şirketlerin her bir çağrı için memnuniyet seviyesini neredeyse anlık olarak hesaplamasına imkan vermektedir [8]. Benzer şekilde, Chowdhury vd. sesli çağrılarda sıra almanın kullanıcı memnuniyetini etkileyen kritik bir faktör olduğunu ve bunun tahmin edilebilirliğini incelemiştir [9]. Luque vd., müşteri hizmetleri diyaloglarında MM'yi akustik özellikler kullanarak tahmin etmek için bir çalışma yürütmüşlerdir. Bu çalışmada, konuşmacıların vurgu düzeylerine göre memnuniyet seviyelerini belirlemek için evrişimsel sinir ağlarından yararlanılmıştır. Önerilen yöntem, AUC ve f-skoru gibi performans ölçütlerinde geleneksel yöntemlere göre daha üstün sonuçlar vermiştir [10]. Chatterjee vd., ses özelliklerini temel alan bir SVM sınıflandırıcı geliştirmiş ve bu sınıflandırıcıyı sorunlu ve sorunlu olmayan telefon görüşmelerini ayırt etmek için kullanmışlardır. Ses özellikleri arasında mel-frekans cepstral katsayıları, enerji, ses ve sıfır geçiş oranı bulunmaktadır. Sınıflandırıcı, sorunlu çağrıları %87,5 oranında doğru bir şekilde belirlemiştir [11].

MİY çalışmalarında, metin sınıflandırması için çağrı merkezlerinde genellikle makine öğrenmesi teknikleri kullanılmaktadır. Bu bağlamda, Meinzer vd., otomotiv endüstrisindeki müşterilerin memnuniyetsizlik seviyelerini tespit etmek için AdaBoost, K-NN, SVM ve RF gibi dört farklı makine öğrenmesi algoritmasını uygulamışlardır. Araştırmaları, radyal temel fonksiyonu çekirdek olarak kullanan SVM yönteminin, %88,8'lik bir doğrulukla memnuniyetsizliği belirlemede diğer algoritmalardan daha üstün olduğunu ortaya koymuştur [12]. Liu vd., çağrı merkezlerindeki hizmet görüşmelerinin nasıl sınıflandırılacağına dair bir model önermiştir. Bu model, anahtar ifade analizi ve LR algoritmalarının birleştirilmesiyle oluşturulmuştur. Modelin performansı, özellikle kısıtlı sayıda eğitim verisi kullanarak

yaptıkları deneylerle kanıtlanmıştır [13]. Busemann vd., çağrı merkezlerinde gelen e-postaları içeriklerine göre sınıflandırmak için çeşitli makine öğrenmesi tekniklerini bir araya getiren sistematik bir yöntem önermiştir. Kullandıkları teknikler arasında sığ metin işleme, tembel öğreniciler, SVM ve sembolik istekli öğreniciler bulunmaktadır. Geliştirdikleri model, çağrı merkezi temsilcilerine yardımcı olacak bir sistemle sorunsuz bir şekilde bütünleştirilmiş ve verilen yanıtların niteliğini yükseltmiştir [14]. Galanis vd., bir çağrı merkezi sohbet veritabanını SVM ile analiz ederek duygusal kısımları elde etmişlerdir [15]. Emmanuela vd., farklı pazar yerlerindeki 549 müşterinin kullanıcı anketlerine metin sınıflandırma teknikleri uygulayarak müşteri memnuniyetini değerlendirmişlerdir. Çalışmada DT, NB, K-NN makine öğrenmesi yöntemleri karşılaştırılmış ve DT algoritmasının en doğru sonuçları verdiği belirlenmiştir [16].

Metin sınıflandırma uygulamalarında, metinlerin sayısal vektörlere dönüştürülmesi önemli bir aşamadır [17]. Salminen vd., çevrimiçi nefret içeren yorumları tespit etmek için farklı platformlardan (Wikipedia, Twitter, YouTube, Reddit) topladıkları veriler üzerinde kapsamlı bir sınıflandırma deneyi gerçekleştirmiştir. Çalışmalarında, Kelime Torbası (BoW), TF-IDF, Word2Vec ve BERT gibi farklı özellik çıkarma yöntemleri ile LR, NB, SVM, XGBoost ve Yapay Sinir Ağı (ANN) gibi farklı sınıflandırma algoritmalarını birlikte kullanmışlardır. En iyi sonucu, 0,92 F1 Puanı ile XGBoost modelinin verdiği görülmüştür [18]. Alaoui ve Nfaoui ise Word2Vec ile CNN ve LSTM ağlarının birleştirilmesiyle dört adet derin öğrenme tabanlı metin sınıflandırma modeli önermişlerdir. Bu modelleri kötü amaçlı HTTP web isteklerini tespit etmek için kullanmışlar ve LSTM'nin hem sınıflandırma metrikleri hem de eğitim süresi bakımından diğer modellerden üstün olduğunu göstermişlerdir [19]. Cahyani ve Patasik de banliyö hattı tweetlerinin duygu analizi için SVM ve Multinomial NB yöntemleri ile TF-IDF ve Word2Vec yöntemlerini karşılaştırmışlardır. İlk olarak tweetleri duygu içerip içermediğine göre ayırmışlar, sonra da kızgın, şaşırmış, korkmuş, mutlu ve üzgün olmak üzere beş duygu kategorisine göre sınıflandırmışlardır. Elde ettikleri sonuçlar, TF-IDF'nin Word2Vec'e göre daha yüksek sınıflandırma performansı sağladığını ve literatürdeki diğer çalışmalara göre daha iyi sonuçlar elde edildiğini göstermiştir [20]. Akuma vd., canlı tweetlerdeki kötü ifadeleri belirlemek için TF-IDF ve BoW gibi özellik çıkarım tekniklerinin karşılaştırmalı bir

analizini yapmışlardır. LR, DT, NB ve K-NN gibi farklı makine öğrenmesi algoritmalarını TF-IDF ve BoW ile birlikte kullanarak Kaggle Nefret Söylemi ve Saldırgan Dil veri setinde denemişlerdir. Çalışmada kullanılan dört performans metriği olan doğruluk, kesinlik, duyarlılık ve f1-skoru açısından değerlendirilen sonuçlara göre, DT en iyi makine öğrenmesi algoritması olarak ortaya çıkmıştır. Ayrıca, TF-IDF bir özellik çıkarım yöntemi olarak BoW'dan daha üstün bir performans sergilemiştir [21].

Türkçe metin verileri üzerinde yapılan araştırmalarda; Ekici ve Takcı, Türkçe spam veri kümesinde Word2Vec ve TF-IDF yöntemlerinin Gradient Boosting algoritmasıyla birleştirilmesinin performansını karşılaştırmışlardır. Yapılan analizlere göre TF-IDF ve Gradient Boosting ikilisinin Word2Vec&Gradient Boosting ikilisine ve CNN modeline kıyasla daha yüksek başarı sağladığı görülmüştür [22]. Koruyan ve Ekeryilmaz, Sikayetvar.com sitesinden alınan Türkiye'nin önde gelen üç tüketici elektroniği satıcısına ait şikayet verilerini sınıflandırdıkları çalışmalarında TF-IDF yönteminin yanında SVM, Stochastic Gradient Descent ve LR gibi sınıflandırma algoritmalarını kullanmışlardır. Çalışmaları LR'nin %80 doğruluk oranıyla en iyi sonucu verdiğini ortaya koymuştur [23]. Çelik ve Koç, farklı kaynaklardan elde edilen Türkçe haber verilerini altı farklı kategoriye ayırmak için bir araştırma yapmışlardır. Araştırmada FastText, Word2Vec ve TF-IDF gibi metin temsil yöntemleri ile ANN, LR, NB, RF ve SVM gibi sınıflandırıcılar kullanılmıştır. SVM&FastText ikilisinin %95,75 doğruluk oranı ile diğer yöntemlerden daha iyi sonuç verdiği tespit edilmiştir [24]. Literatür taramasında geçen çalışmaların kullandıkları metod ve veri setleri de Çizelge 1.1’de gösterilmiştir.

Çizelge 1. 1. Literatür taraması özeti.

Kaynak	Kullanılan Metod	Kullanılan Veri Seti
[12]	AdaBoost, K-NN, SVM ve RF	Bir otomotiv şirketinin 19.008 gerçek servis müşteri yorum veri seti
[13]	Anahtar ifade analizi ve LR	Telekomünikasyon şirketi müşteri hizmet görüşmeleri veri seti
[14]	Sığ metin işleme ve Makine öğrenimi teknikleri	4777 e-posta
[15]	SVM	Bir Yunan telekom şirketine ait 135 çağrı
[16]	DT, NB, K-NN makine öğrenmesi yöntemleri	Farklı pazar yerlerindeki 549 müşterinin kullanıcı anketleri
[18]	BoW, TF-IDF, Word2Vec, BERT & ile LR, NB, SVM, XGBoost ve ANN	Wikipedia, Twitter, YouTube, Reddit platform verileri
[19]	Word2Vec ile CNN ve LSTM ağlarının birleştirilmesi	223.585 örnek içeren HTTP CSIC 2010 veri seti
[20]	SVM ve Multinomial NB yöntemleri ile TF-IDF ve Word2Vec	Banliyö hattı tweetlerinin duygu analizi
[21]	TF-IDF ve BoW ile LR, DT, NB ve K-NN makine öğrenmesi algoritmaları	Kaggle Nefret Söylemi ve Saldırgan Dil veri seti
[22]	Word2Vec ve TF-IDF ile Gradient Boosting algoritması	Türkçe spam mail veri seti
[23]	SVM, Stochastic Gradient Descent ve LR	Üç tüketici elektroniği satıcısına ait şikayetvar.com verileri
[24]	FastText, Word2Vec ve TF-IDF ile ANN, LR, NB, RF ve SVM	Türkçe haber verileri

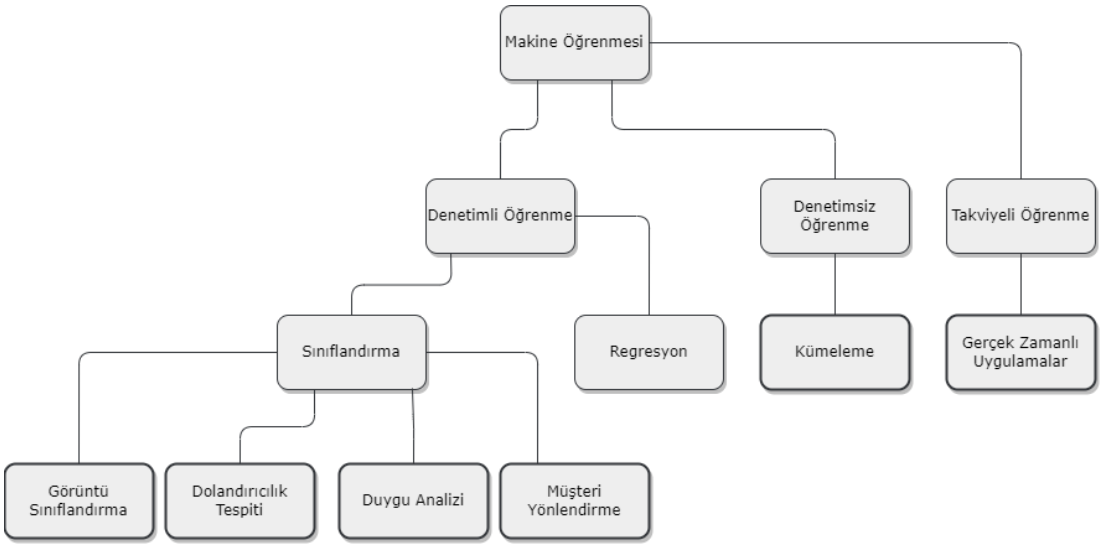
BÖLÜM 2

SINIFLANDIRMA

Makine öğrenmesinin bir alt dalı olan sınıflandırma, verileri birden çok sınıfa, yani ayrık değerlere ayırmaya yardımcı olan bir model veya işlevi bulma veya keşfetme sürecidir. Makine öğrenmesindeki sınıflandırma algoritmaları, sonraki verilerin önceden belirlenmiş kategorilerden birine girme olasılığını tahmin etmek için girdi olarak eğitim verilerini kullanır. Sınıflandırma, gelecekteki veri kümelerinde aynı modeli (benzer kelimeler veya duygular, sayı dizileri, vb.) bulmak için eğitim verilerine uygulanan sınıflandırma algoritmaları ile bir örüntü tanıma biçimidir.

2.1. MAKİNE ÖĞRENMESİ TÜRLERİ

Verilerin niteliğine ve istenilen sonuca bağlı olarak Makine Öğrenmesi algoritmalarının üç ana kategorisi bulunmaktadır; Denetimsiz Öğrenme, Denetimli Öğrenme ve Takviyeli Öğrenme. (Şekil 2.1).



Şekil 2. 1. Makine öğrenmesi kategorileri.

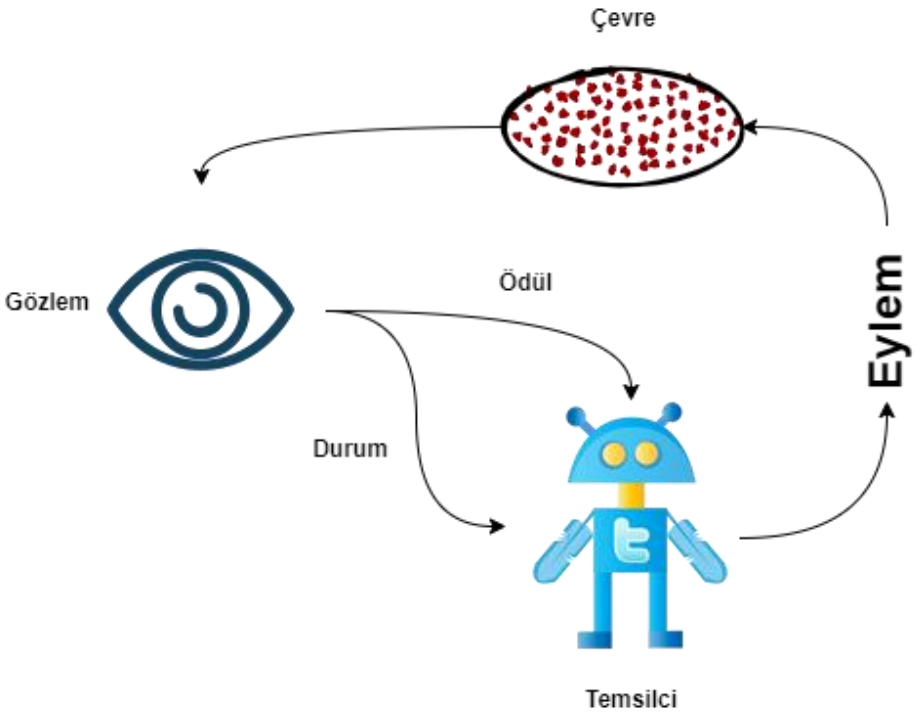
Denetimsiz Öğrenme: Etiketlenmemiş veri gruplarını incelemek ve sınıflandırmak için makine öğrenmesi algoritmalarını kullanan türdür. Bu algoritmalar, insan müdahalesine ihtiyaç duymadan gizli desenleri veya veri kümelerini ortaya çıkarabilir [25]. Denetimsiz öğrenme, bilgilerdeki benzerlikleri ve farklılıkları keşfetme yeteneği ile öne çıkar, bu da onu keşif amaçlı veri analizi, satış stratejileri, müşteri segmentasyonu ve görüntü tanıma gibi alanlarda ideal bir çözüm haline getirir [26].

Denetimsiz öğrenme üç amaç için kullanılmaktadır;

- **Kümeleme**: Etiketlenmemiş verileri benzerliklerine veya farklılıklarına göre gruplandıran bir tekniktir. Kümeleme algoritmaları, ham ve sınıflandırılmamış veri nesnelerini, bilgideki yapılar veya kalıplarla temsil edilen gruplar halinde işlemek için kullanılır. Bu algoritmalar genellikle dışlayıcı, örtüşen, hiyerarşik ve olasılıksal olmak üzere farklı türlerde gruplandırılabilirler [26].
- **Birliktelik Kuralları**: Veri kümesindeki değişkenler arasındaki ilişkileri ortaya çıkarmak için kural tabanlı bir yöntem olan birliktelik kuralları, özellikle pazar sepeti analizi uygulamalarında işletmelere büyük fayda sağlar. Bu yöntemler sayesinde, farklı ürünler arasındaki bağlantıları daha iyi kavrayabilir, müşterilerin tüketim davranışlarını anlayabilir ve daha etkin çapraz satış stratejileri ve öneri motorları oluşturabilmektedir. Örneğin, bir markette "Bu Ürünü Alan Müşteriler Ayrıca Şunu da Satın Aldı" şeklindeki ifadeler bu yöntemlerin sonucudur. Bu alanda en çok kullanılan algoritmalardan biri Apriori algoritmasıdır.
- **Boyut Azaltma**: Boyut azaltma, çok fazla özelliğe veya boyuta sahip bir veri kümesini daha küçük bir boyuta indirgemek için kullanılan bir tekniktir. Veri kümesinin temel yapısını ve bilgisini mümkün olduğunca bozmadan, veri girişlerinin sayısını daha kolay işlenebilir bir seviyeye düşürür. Veri ön işleme sürecinde sıkça başvurulan bir yöntemdir.

Takviyeli Öğrenme: Takviyeli öğrenme, bir problemi çözmek için doğru kararları nasıl alacağını öğrenen, kendi kendine hareket edebilen ve çevresini algılayabilen bir sistemdir [27]. Bu yöntem, oyun geliştirme, finans, sağlık, akıllı ulaşım gibi çeşitli alanlarda yaygın olarak kullanılır.

Takviyeli öğrenme, daha geniş çapta çalışılan denetimli öğrenme yönteminden birkaç yönden farklılık göstermektedir. En önemli fark, belirli girişlerle doğrudan eşleşen sonuçlar bulunmamasıdır. Başka bir ifadeyle, takviyeli öğrenme problemlerinde genellikle bir giriş/çıkış çifti verisiyle eğitim yapılmaz. Bu yöntem yerine, yeni durumun hedef durumunu belirtmek için bir ödül/ceza ile destek işlemi gerçekleştirilir, sistem kendi deneyimleri ve elde ettiği geri bildirimlerden yararlanarak öğrenmeyi sürdürür. Takviyeli öğrenmedeki temel amaç, öğrenen sistemin faaliyet gösterdiği potansiyel durumları belirlemek ve denenen her durumun doğruluğunu veya yanlışlığını anlamak, böylece sistem performansını geliştirmektir. Sistemdeki yer alan temsilcinin, optimum şekilde hareket etmesi için olası sistem durumları, eylemleri, geçişleri ve ödülleri hakkında yararlı deneyimler toplaması gerekmektedir (Şekil 2.2). Denetimli öğrenmeden başka bir farkı ise, çevrimiçi performansın önemli olmasıdır, sistemin değerlendirilmesi genellikle öğrenme ile eşzamanlıdır.

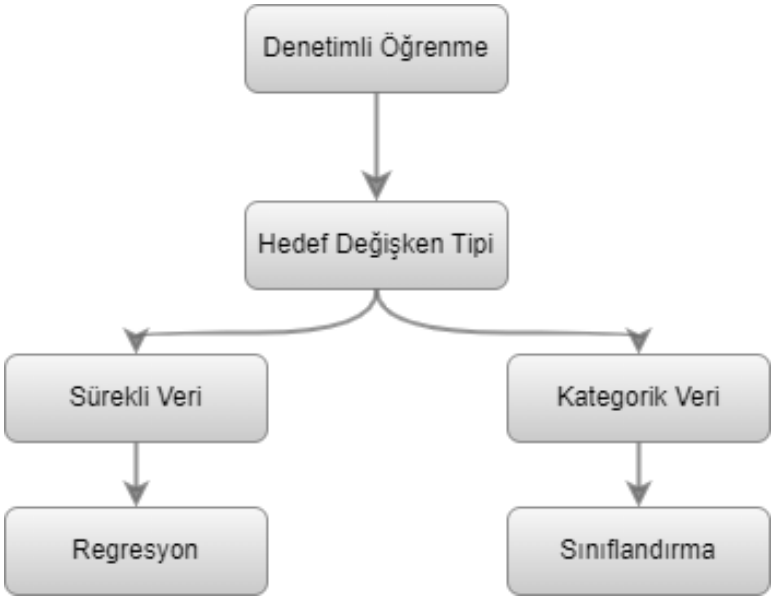


Şekil 2. 2. Takviyeli öğrenme modeli.

Denetimli Öğrenme: Denetimli öğrenme, eğitim seti ve bir test seti olmak üzere iki veri seti ile gerçekleştirilen öğrenmedir [28,29]. Bu yöntemdeki amaç, eğitim setindeki öğrenicilerin, eğitim setindeki bir dizi etiketli örnekten "öğrenmesi" ve böylece test

setindeki etiketlenmemiş örnekleri mümkün olan en yüksek doğrulukla tanımlayabilmesidir.

Sınıflandırma ve regresyon denetimli öğrenme kategorisinden olsa da aynı görevi yerine getirmemektedir (Şekil 2.3). Tahmin görevi, hedef değişken ayrık olduğunda bir sınıflandırmadır. Kitap içeriklerinden kategorisini tahmin etmeye çalışmak veya sosyal ağlardaki paylaşımlardan duygu tahmininde bulunma sınıflandırmaya örnek verilebilir. Tahmin görevi, hedef değişken sürekli olduğunda bir regresyondur. Bir kişinin eğitim derecesi, önceki iş deneyimi, coğrafi konumu ve uzmanlık düzeyi göz önünde bulundurularak alacağı ücretin tahmini regresyona örnek verilebilir.

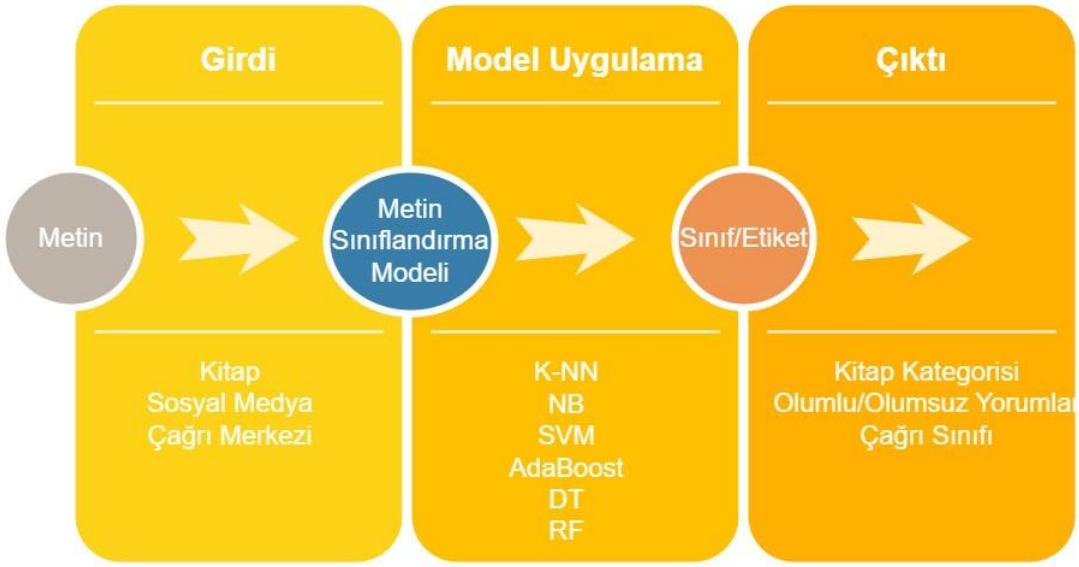


Şekil 2. 3. Denetimli öğrenme.

2.2. METİN SINIFLANDIRMA

Otomatik metin sınıflandırılması, belgeleri genellikle makine öğrenmesi algoritmaları kullanarak önceden belirlenmiş sınıflar halinde düzenleme görevidir [30,31]. Şekil 2.4'te basit bir metin sınıflandırmasının nasıl yapıldığı gösterilmektedir. Metin sınıflandırmaya örnek olarak; Duygu Analizi, Etiket Sınıflandırma, Doküman Etiketleme verilebilir. Metinsel sınıflandırma da görevleri bakımından üç farklı türe

ayrılabilir; ikili sınıflandırma, çok sınıflı sınıflandırma, çok sınıflı çok etiketli sınıflandırma.



Şekil 2. 4. Metin sınıflandırma genel yapısı.

İkili Sınıflandırma: İçeriklerin ilişkilendirileceği sınıfların/etiketlerin sayısı iki adettir; evet/hayır, olumlu/olumsuz vb. örnek verilebilir.

Çok Sınıflı Sınıflandırma: İçeriğin ilişkilendirileceği sınıfların/etiketlerin sayısı n>2 olmak üzere, n adet olabilir.

Çok Sınıflı Çok Etiketli Sınıflandırma: Her içeriğin birden fazla sınıfla ilişkilendirilebildiği ve bu nedenle birden fazla etikete sahip olduğu çok sınıflı sınıflandırma türüdür.

Bu türlerden birine göre bir metinsel sınıflandırma yapabilmek için üç adımı gerçekleştirmek gerekir:

- **Verileri etiketlemek**: Sınıflandırma yapılmak istenen özelliklerle etiketlenmiş metin içeriklerinden oluşan bir eğitim seti oluşturmak.

- **Metni dönüştürme ve işleme**: Metin içeriklerinin, gerçekleştirilecek görevde daha uygun bir forma sokulabilmesi için gerekli olan ön işleme aşamasıyla birlikte, aynı zamanda makine öğrenmesi algoritmaları tarafından daha kolay okunabilir ve analiz edilebilir hale getirilebilmeleri için içeriklerin bir dönüşüm aşamasından geçmeleri gerekir.
- **Model eğitimi**: Sınıflandırmada sıklıkla kullanılan algoritmalar ile model eğitiminin gerçekleştirilmesi.

2.3. VERİ ÖN İŞLEME

Veri ön işleme, makine öğrenmesi için verileri hazırlama sürecidir. Ham verilerdeki eksiklik, yanlışlık, tekrarlar, tutarsızlık, çeşitlilik gibi sorunları gidermek ve verileri daha anlamlı ve kullanılabilir hale getirmek için bir dizi işlem içerir.

Bu çalışmada kullanılan veri seti, müşteriler ile TBÇM'deki çağrı merkezi temsilcileri arasındaki telefon görüşmelerinin metne dönüştürülmüş formatlarından oluşmaktadır. Veri seti için veri ön işleme adımları; önce veri temizleme, sonra dizgeleme ve en son kök çözümleme görevlerini kapsayacak biçimde sıralanmıştır (Şekil 2.5).

Şekil 2. 5. Veri ön işleme adımları.

Çalışmada, veri temizleme süreci birkaç adımdan oluşmaktadır. Öncelikle, veri kümesi yazım hatalarından, boş metinlerden, tekrarlardan ve noktalama işaretlerinden arındırılmıştır. Bunun yanı sıra, kimlik numaraları, telefon numaraları ve gümrük beyanname numaraları gibi kişisel veriler çıkarılmıştır. Ayrıca, cümleye anlam katmayan, sıklıkla kullanılan ve genelde cümle yapısı ve uyum için gerekli olan etkisiz kelimeler temizlenmiştir [30,32]. Şekil 2.6'da Python programlama dili ile yazılan fonksiyonla gerçekleştirilen veri ön işleme yöntemleri sunulmuştur. Ön işlemenin ikinci aşamasında, metin kelimelere ayrılarak dizgelenmiştir. Son olarak, kelimelerin köklerini bulmak için kök çözümleme işlemi yapılmıştır. Çizelge 2.1'de de örnek bir metin için ön işlemeden önce ve sonrası gösterilmiştir.

Çizelge 2. 1. Veri ön işleme öncesi ve sonrası.

Ön İşlemeden Önce	Ön İşlemeden Sonra
İnternetden aldığım ürünler bozuk çıktı. 1000 lira ödemem geri yatırılmadı.	internet al ürün boz çık lira öde geri yat

```
def norm_doc(single_doc):
    single_doc = single_doc.lower()
    single_doc = re.sub('\W+', ' ', single_doc)
    single_doc = re.sub('\d+', '', single_doc)
    single_doc = single_doc.strip()
    tokens = WPT.tokenize(single_doc)
    new_token=[]
    for token in tokens:
     new_doc=simplemma.lemmatize(token, langdata)
     new_token.append(new_doc)
    tokens=[]
    tokens=new_token
    # TR: Stop-word listesindeki kelimeler hariç al
    # EN: Filter out the stop-words
    filtered_tokens = [token for token in new_token if token not in stop_word_list]
    # TR: Dokümanı tekrar oluştur
    # EN: Reconstruct the document
    single_doc = ' '.join(filtered_tokens)
    #print(single_doc)
    return single_doc
norm_docs = np.vectorize(norm_doc)
normalized_documents = norm_docs(docs)
```

Şekil 2. 6. Python ile veri ön işlemenin uygulanması.

Etkisiz Kelimeler (Stopwords): Etkisiz kelimeler; bir dilde sıklıkla kullanılan ve önemli bir anlam içermeyen sözcüklerdir. Etkisiz kelimelerle başa çıkmak için en yaygın teknik, onları metinlerden ve belgelerden çıkarmaktır [30,31,33]. Bu çalışmada örneğin; 'ama', 'gibi', 'fakat', 'yani', 'ya', 'mı', 'mi', 'ne', 'nerede' vb. 53 adet etkisiz kelime 'NLTK' kütüphanesinin 'Turkish' sınıfı kullanılarak filtrelenmiştir [34].

Kök Çözümleme (Lemmatization): Kök çözümleme, 'lemma' olarak ta bilinen temel formunu elde etmek için bir kelime üzerindeki son ekleri değiştiren veya ortadan kaldıran temel bir DDİ tekniğidir [35,36]. Örneğin; ‘gözlükçüler’ kelimesinin lemma formu ‘gözlükçü’ olarak bulunmaktadır. Çalışmada, kelimelerin kökünü veya temel biçimini belirlemek için tasarlanmış açık kaynaklı bir Python kütüphanesi olan Simplemma'dan '*tr*' sınıfı kullanılmıştır [37]. Kök çözümleme işlemiyle kelimelerin türediği form bulunarak dizgeleme aşamasına iletilmektedir.

Gövdeleme (Stemming): Bir kök/temel kelimenin morfolojik çeşitlerini üretme sürecidir. Gövdeleme programlarına genellikle Gövdeleme algoritmaları veya gövdeleyici denir [38]. Gövdeleme algoritması, 'chocolates', 'chocolatey', 'choco' kelimelerini kök kelimeye, 'chocolate' ve 'retrival', 'retried', 'retrievs' kelimelerini 'retrive' köküne getirir. Gövdeleme, DDİ’de ardışık düzen oluşturma sürecinin önemli bir parçasıdır. Bu tez çalışmasında hem Gövdeleme hem Kök Çözümleme işlemi uygulamak yerine sadece kök çözümleme uygulanmıştır. Gövdeleme yöntemin uygulanmasının Türkçe sözcüklerde anlam kaybına yol açtığı gözlemlendiğinden çalışmada kullanılmamıştır.

Dizgeleme (**Tokenization)**: Dizgeleme, metni kelimeler, kelime öbekleri veya semboller gibi dizge olarak adlandırılan anlamlı birimlere ayıran önemli bir ön işleme tekniğidir [39]. Metin kelime, boşluk, noktalama işaretleri vb. kriterlere göre dizgelere ayrılır. Bu çalışmada, veri ön işlemenin son aşamasında dizgeleme ile metinler boşluklara göre parçalanarak kelime dizisine dönüştürülmüştür. Dizgeleme işlemi için NLTK kütüphanesi kullanılmıştır. Veri ön işlemeyi takip eden sonraki aşama olan metin temsil yöntemi için veri hazır hale getirilmiştir.

BÖLÜM 3

METİN SAYISALLAŞTIRMA

Kelimeler, tek başına anlam taşıyan ya da birbirine bağlı birçok biçimbirimden (morfem) oluşan ve ses değeri içeren dil birimleri olarak ifade edilir [40]. Duygu, düşünce, istek, haber, durum, olay gibi unsurları ifade etmek amacıyla oluşturulan ve kendi içinde anlam ve mantık bütünlüğü taşıyan sözcük veya sözcük gruplarına cümle denir. Metin sınıflandırma görevlerinde de, modellerin metni daha iyi anlamasına, sınıflandırmasına veya sınıf oluşturmasına yardımcı olabilecek kelimeleri doğru bir şekilde temsil etmek önemli bir adım haline gelmektedir [41]. Metin belgeleri genellikle çok çeşitli bilgiler içerir, ancak geleneksel veritabanlarının sağladığı işlevsel yapıya sahip değillerdir. Bu nedenle, yapılandırılmamış verilerin, özellikle serbest metin verilerinin, yapılandırılmış verilere dönüştürülmesi gereklidir. Bu dönüşümü gerçekleştirmek için literatürde birçok ön işleme tekniği kullanılmaktadır [42].

Yapılandırılmış ve gürültüden arındırılmış veriler, doğru ve hızlı sınıflandırma yapabilmek için kelime sayısallaştırma yöntemlerine tabi tutulur. Sayısallaştırılan metin verileri hem daha fazla sayıda modelle çalışmaya imkan verir hem de veri boyutu azaldığından hızlı sonuçlar elde etmeye katkıda bulunur. Literatürde çeşitli kelime sayısallaştırma yöntemi ve türevleri mevcuttur. Bazı popüler kelime sayısallaştırma yöntemlerine örnek; Word2Vec, GloVe, FastText, Doc2Vec, BoW, TF-IDF. Bu yöntemler metinsel verileri sayısal temsillere dönüştürerek DDİ sorunlarını matematiksel yaklaşımlarla ele almamızı sağlamaktadır.

3.1. KELİME TORBASI (BAG OF WORDS)

BoW, makine öğrenmesi algoritmalarında kullanılmak üzere metinden özellik çıkarma tekniğidir. BoW'a dilbilimsel bağlamda ilk atıf Zellig Harris'in Dağıtımsal Yapı üzerine 1954 tarihli makalesinde rastlanmaktadır [43]. BoW, daha ileri DDİ analizi için metinden özellik çıkarmaya yönelik bir tekniktir. BoW, semantik anlamlarını, bağlamsal ilgilerini veya sıralarını dikkate almadan bir metin içindeki kelimelerin oluşumunu ve tekrarını dikkate alarak çalışır.

Örneğin; bu çalışmada kullanılan çağrı kayıtlarındaki her bir benzersiz kelime ayrı bir özelliğe karşılık gelmektedir. Çağrı metninde bir kelime bulunduğunda, bu kelimeye sıfır olmayan bir değer atanır. Buna ek olarak, bir kelime metin içinde tekrar ediyorsa, bu kelime için BoW değeri metindeki sıklığına karşılık gelir. Bir kelime bir metinde iki kez tekrarlanıyorsa, bu kelimenin BoW vektöründeki değeri ikidir [44]. Örnek olarak 5 farklı metnin BoW değerleri Çizelge 3.1'de hesaplanmıştır.

Örnek metinler:

- Metin 1: "Tır gümrükten çıkış yaptı."
- Metin 2: "Kapıkule gümrüğünde tırlar kuyruk oluşturdu."
- Metin 3: "Limandan aldığımız eşyaların gümrük işlemleri bitmedi."
- Metin 4: "Bayiden aldığım eşya bozuk çıktı. Firma eşyayı geri almıyor."
- Metin 5: "Firmadan aldığım eşya bozuk çıktı."

Çizelge 3.1. BoW yöntemi.

Metinler	al	bitmek	bozuk	çık	eşya	firma	geri	gümrük	bayi	işlem	tır
1	0	0	0	1	0	0	0	1	0	0	1
2	0	0	0	0	0	0	0	1	0	0	1
3	1	1	0	0	1	0	0	1	0	1	0
4	1	0	1	1	2	1	1	0	1	0	0
5	1	0	1	1	1	1	0	0	0	0	0

3.2. TERİM FREKANSI-TERS DOKÜMAN FREKANSI (TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY)

BoW'un temel dezavantajlarından biri, sık kullanılan kelimelerin metin vektörlerinde baskın hale gelmesidir. Metinlerde önemli bir anlamsal etkiye sahip olmayan bu sık kullanılan kelimeler, diğer özellikleri bastırabilir ve etkilerini azaltabilir. Bu sorunu çözmek için TF-IDF, metin vektörleri için hem terim sıklığını hem de ters belge sıklığını hesaplar. TF-IDF sadece bir kelimenin bir belgede kaç kez geçtiğini değil, aynı zamanda tüm derlemdeki önemini de ölçer. Terim sıklığı (TF) bir kelimenin bir belgedeki sıklığını hesaplarken, ters belge sıklığı (IDF) kelimenin tüm belgelerdeki nadirliğini ölçer [45]. Bir belgedeki bir kelime için atanan puan, 0 ile 1 arasında değişen TF ve IDF puanlarının çarpımı ile orantılıdır. Bir belgede sıkça geçen ancak tüm belgelerde nadiren geçen sözcükler, önemlerini gösteren yüksek bir puan alır. Tersine, tüm belgelerde sıkça geçen kelimeler düşük puan alır ve bu da önemsiz olduklarını gösterir.

TF, Denklem 3.1'de gösterildiği gibi, bir terimin bir belgede bulunma sayısının o belgedeki kelime sayısına bölünmesiyle belirlenir. Öte yandan IDF, bir terimin belge koleksiyonunda ne sıklıkta bulunduğunu gösterir. Belgelere özgü terimler, ortak sözcüklere kıyasla daha yüksek IDF değerlerine sahiptir. IDF'yi hesaplamak için, toplam belge sayısının bir kelimeyi içeren belge sayısına bölünmesi ve ardından Denklem 3.2'de formüle edildiği gibi logaritmasının alınması gerekmektedir. TF-IDF değeri, Denklem 3.3'te gösterildiği gibi TF ve IDF'nin çarpılmasıyla elde edilir.

Belirli bir metin içinde basit bir TF-IDF değerinin pratik hesaplamasını bir örnekle gösterirsek; örneğin, 40 kelimelik bir metinde "liman" kelimesi iki kez geçiyorsa, TF değeri 0,05'dir. Benzer şekilde, "liman" kelimesinin külliyatta bu terimi içeren metin sayısına bakılır; 400 metnin 20'sinde geçiyorsa terim, çıkan sonucun logaritması alındığında IDF değeri 1,30 olarak elde edilir. Bu iki değerin çarpılmasıyla TF-IDF değeri 0,065 olarak hesaplanır. Çizelge 3.2’de TF-IDF yöntemine ait örnek bir puan tablosu gösterilmiştir.

$$TF = \frac{terimin\ belgede\ geçme\ sayısı}{belgedeki\ toplam\ terim\ sayısı} \quad 3.1$$

$$IDF = \log\left(\frac{külliyattaki\ belgelerin\ sayısı}{külliyatta\ terimi\ içeren\ belgelerin\ sayısı+1}\right) \quad 3.2$$

$$TF - IDF = TF * IDF \quad 3.3$$

Bu çalışmada, TF-IDF özellik matrisini oluşturabilmek için *scikit-learn* kütüphanesinin *TfidfVectorizer* metodundan yararlanılmıştır.

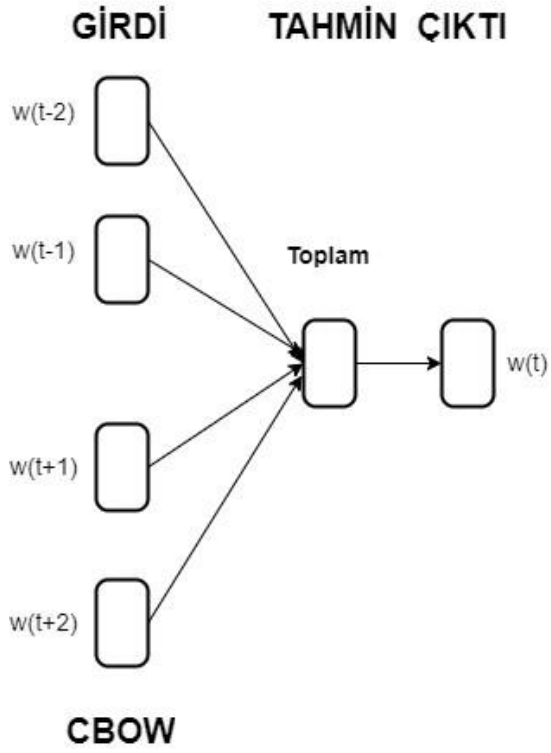
Çizelge 3.2. TF-IDF yöntemi.

Metin	**al**	**almanya**	**ara**	**kapıkule**	**bir**	**cam**	**liman**	**kooperatif**
0	0	0	0.290	0	0.280	0	0	0
1	0	0	0	0	0.255	0	0	0.396
2	0	0	0	0	0	0	0,065	0
3	0	0.319	0.478	0	0.325	0	0	0
4	0.474	0	0	0	0	0	0	0
5	0	0	0	0	0	0.454	0	0.482
6	0	0,280	0	0.120	0	0	0	0

3.3. WORD2VEC

Word2Vec, Mikolov vd. 2013 yılında Google arama motorunun performansını yükseltmek amacıyla geliştirdikleri bir metin temsil yöntemidir [46]. Bu yöntem, sığ yapay sinir ağlarını tahmin temelli bir biçimde kullanarak kelime vektörleri oluşturur. Word2Vec'in sığ sinir ağı, anlamsal benzerlikleri hızlı bir şekilde tanıyabilir ve lojistik regresyon yöntemlerini kullanarak eşanlamlı kelimeleri belirleyebilir, bu da onu derin sinir ağlarından daha hızlı hale getirmektedir. TF-IDF gibi metni yüksek boyutlu seyrek matrislerle kodlamak yerine Word2Vec, metnin her kelimesi için yoğun vektörlerden oluşan kelime gömmeleri üretir. Ayrıca vektör uzayında benzer kelimeleri birbirine yakın yerleştirir. Bu gömmeler, vektör uzayında anlamsal olarak yakın olan kelimeleri bir araya getirir. Kelime gömmeleri, daha az hesaplama maliyeti ve DDİ uygulamalarında daha iyi sonuçlar sağlayan iki önemli fayda sağlar [47,48]. Word2Vec, kelimelerin metin içindeki bağlamlarını dikkate alarak kelime gömmeleri oluşturur. Bunun için iki farklı sinir ağı modeli kullanır: Continuous Bag-of-Words (CBOW) ve Skip-gram [49].

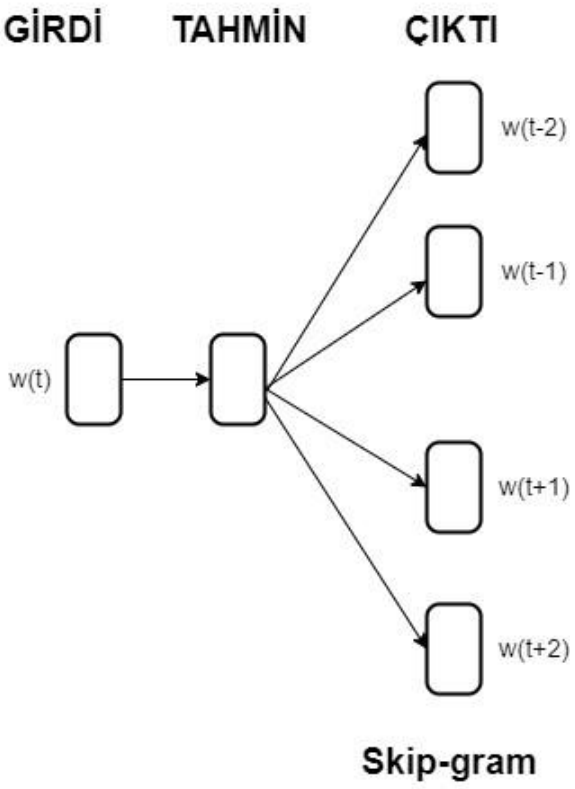
CBOW modeli, merkezi kelimeyi tahmin etmek için, hedeflenen kelimeye yakın olan metindeki birkaç bağlamsal kelimeyi kullanır. CBOW, komşu kelimelerin sayısını belirlemek için sabit bir pencere boyutu kullanır. Bu penceredeki kelimeler, ilgili kelimenin tahminine eşit şekilde katılır. Şekil 3.1, CBOW yapısını görsel olarak göstermektedir; w(*t*) mevcut kelimeyi ve w(*t-2*) ile w(*t+2*) arasındaki kelimeler, w(*t*) içindeki pencere boyutu olan ilgili kelimenin bağlamsal kelimeleridir. Bu sinir ağında, gizli katman normal, tam bağlı, yoğun bir katmandır ve çıktı katmanı kelime dağarcığındaki hedef kelime için olasılıkları belirler.



Şekil 3.1. CBOW genel yapısı.

Skip-gram modeli, bir metindeki bir kelimeyi pencere boyutunun ortasına yerleştirir ve bu pencerede bulunan diğer kelimeleri çıktı olarak tahmin etmeye çalışır. Penceredeki ortadaki kelime w(*t*) ile ifade edilir ve çevresindeki kelimeler w(*t-2*) ile w(*t+2*) arasındaki bir aralıkta Şekil 3.2'de görüldüğü gibi belirtilir. Skip-gram modeli, tek bir kelimenin bağlamına dayanarak yakınındaki kelimeleri tahmin edebilir. Bu özellik, Skip-gram modelinin CBOW'dan farklı olarak nadir veya seyrek kelimeleri temsil etmede daha başarılı olmasını sağlar. Skip-gram ayrıca büyük veri setlerini

işlemede CBOW'dan daha verimlidir. Bu sebeplerle ve önceki araştırmalarda Skip-gram'ın CBOW'ya göre daha hızlı ve daha iyi sonuçlar vermesi nedeniyle bu çalışmada da Skip-gram modeli tercih edilmiştir.



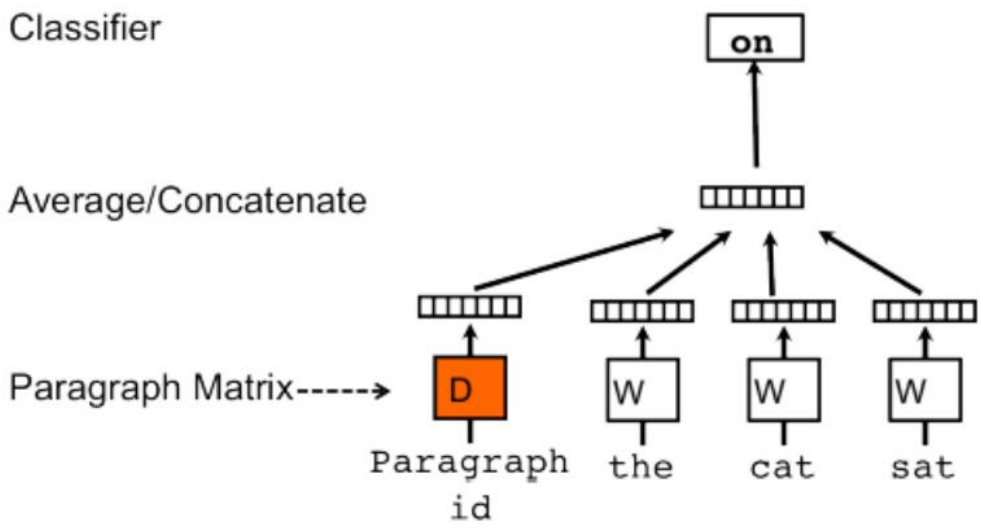
Şekil 3.2. Skip-gram genel yapısı.

3.4. DOC2VEC

Doc2Vec, yapay sinir ağlarına dayalı bir yaklaşım olup, 2014 yılında Mikolov tarafından geliştirilmiştir. Bu yöntem, kelimeler ve bu kelimeleri içeren dokümanlar arasındaki anlamsal ilişkiyi içerir ve yönetilebilir boyutlu doküman vektörlerinin elde edilmesini sağlar [50]. Doc2Vec yönteminin temel amacı, n boyutlu bir vektör uzayında her dokümanı temsil eden bir vektör elde etmektir. Bu şekilde, dokümanlar arasındaki anlamsal benzerlikleri tespit etmek mümkün olur [51].

Word2Vec'ten farklı olarak Doc2Vec'te kelime vektörleri oluşturulurken bu vektörlerin yanına bir de paragraf veya doküman vektörü eklenmesi prensibine göre çalışır (Şekil 3.3). *D* matrisi bir vektöre denk gelen paragraf temsilcisi olduğu

varsayılırsa, bu *D* vektörünün üç kelimelik bir bağlamla değerlendirilmesi ve bağlamdaki eksik bilgiyi tamamlaması, dördüncü kelimeyi tahmin etmekte kullanılır.



Şekil 3.3. Paragraf vektörünü öğrenme modeli [50].

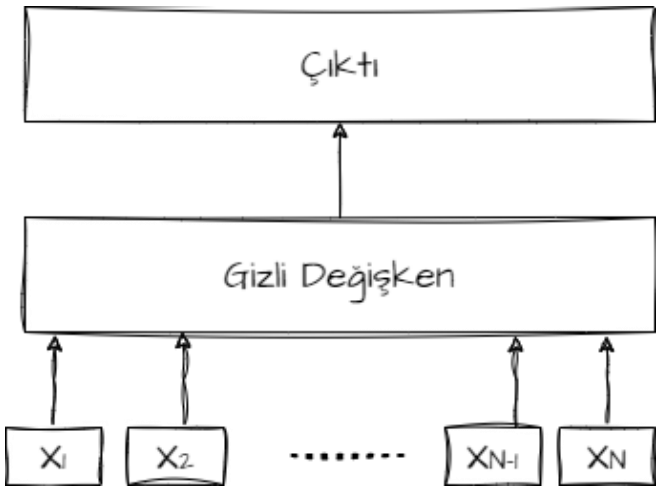
3.5. FASTTEXT

Facebook'taki araştırma ekibi tarafından geliştirilen bir kelime temsil tekniği olan FastText, Word2Vec tekniği tarafından kullanılan skip-gram yöntemi üzerine inşa edilmiştir. Word2Vec, tek tek kelimeleri analiz için en küçük birimler olarak görürken, FastText kelimeleri n-gram koleksiyonları olarak ele alarak farklı bir yaklaşım benimsemektedir. Bireysel n-gramların birleştirilmesiyle tespit edilen bu n-gramlar, vektör temsilleri sağlamaktadır. Diğer birçok yöntem, kelimeleri vektörlere dönüştürme süreci sırasında kelimelerin morfolojisini yeterince hesaba katmamaktadır. Bu dezavantaj, geniş bir kelime dağarcığına ve nadir kelimelerin önemli bir varlığına sahip dillerle uğraşırken özellikle belirginleşir. Bununla birlikte, FastText, n-gram yaklaşımını benimseyerek bu sınırlamanın üstesinden gelir ve böylece altta yatan dilsel yapının kapsamlı bir şekilde anlaşılmasını sağlar [52].

Almanca gibi dillerde belirli ifadeler tek kelime ile ifade edilmektedir. Örneğin masa tenisi ifadesi *Tischtennis* olarak yazılır. Word2Vec'te, *masa tenisi* ve *tischtennis* temsilleri ayrı ayrı öğrenilir. Bu da, masa tenisi ve tischtennisin ilişkili olduğu

çıkarımını yapmayı zorlaştırır. FastText, bu kelimelerin karakter n-gram temsilini öğrenerek, masa tenisi ve tischtennisi, örtüşen n-gramları paylaşarak ve vektör uzayında daha ilişkili hale getirerek, ilgili kavramların daha kolay ortaya çıkarılmasını kolaylaştırmaktadır.

Şekil 3.4'te N adet n-gram özelliği olan bir cümle gösterilmektedir. Özelliklerin (X_1, ..., X_N) vektörleri çıkarılır ve gizli değişkeni bulmak için ortalamaları alınır.



Şekil 3.4. FastText genel yapısı.

Özet olarak; Word2Vec ve FastText kavramsal olarak aynı amaca sahip olsalar da bilinmeyen sözcükleri tahmin etmek için sözcükleri kullanan Word2Vec'in aksine, FastText karakter n-gramlarıyla daha ayrıntılı bir düzeyde çalışır. FastText'in Word2Vec'den daha iyi olduğunu söylemek mümkün değil ve dilden dile bu sayısallaştırma yöntemi performans farklılığı gösterebilmektedir.

3.6. GLOVE

GloVe (Global Vectors), Pennington vd., önerdiği bir kelime gömme yöntemidir [32]. GloVe, V boyutunda bir kelime dağarcığına sahip bir metin kümesinden birliktelik matrisi (X) oluşturur. Bu matris, V×V boyutunda olup, her hücresi (X_{ij}) i ve j kelimelerinin birlikte görülme sayısını belirtir. Örneğin, "Bir elma ağacın dalında

kalmıştı" cümlesinde belirli bir pencere büyüklüğü kullanılarak hesaplanan birliktelik matrisi Şekil 3.5'te gösterilmiştir.

Bu yöntem, Word2Vec'ten farklı olarak, kelime tahmin etmek için olasılık dağılımlarını temel alan yeni bir yansız amaç fonksiyonu sunar. GloVe, hem cümle düzeyinde elde edilen yerel bilgileri hem de metin kümesi düzeyinde elde edilen küresel bilgileri dikkate alarak, bir kayıp fonksiyonu vasıtasıyla tahmin hatalarını minimize eder ve böylece doğru kelime vektörleri elde eder. Diğer yandan, çalışmada kullanılan GloVe yöntemi için ilgili parametreler Çizelge 3.3'te verilmiştir.

Kelimeler	***Bir***	***elma***	***ağacın***	***dalında***	***kalmıştı***
Bir	0	**1**	0	0	0
elma	**1**	0	**1**	0	0
ağacın	0	**1**	0	**1**	0
dalında	0	0	**1**	0	**1**
kalmıştı	0	0	0	**1**	0

Şekil 3.5. GloVe örnek cümle matrisi.

Çizelge 3.3. GloVe yönteminde kullanılan parametre değerleri.

Parametre	**Değer**
input_file	'./veri.xlsx'
vocab_size	default=100,000
max_size	default = 0
min_count	default = 5
window	default = 15
embed_size	default = 50
epoch	default = 25
threads	default = 8
memory_limit	default = 4
lr	default = 0.05

Bu tez çalışmasında, veri ön işleme adımının ardından, çağrı metinlerini vektör temsillerine dönüştürmek için üç farklı yöntem uygulanmıştır: TF-IDF, Word2Vec ve GloVe.

BÖLÜM 4

SINIFLANDIRMA ALGORİTMALARI

Bu tez çalışmasında popüler denetimli öğrenme sınıflandırma algoritmaları metin sınıflandırmada analiz amacıyla kullanılmış ve bu bölümde sıklıkla kullanılan altı sınıflandırma algoritması; K-NN, NB, SVM, AdaBoost, DT ve RF detaylı açıklanmıştır.

4.1. K EN YAKIN KOMŞU (K-NN)

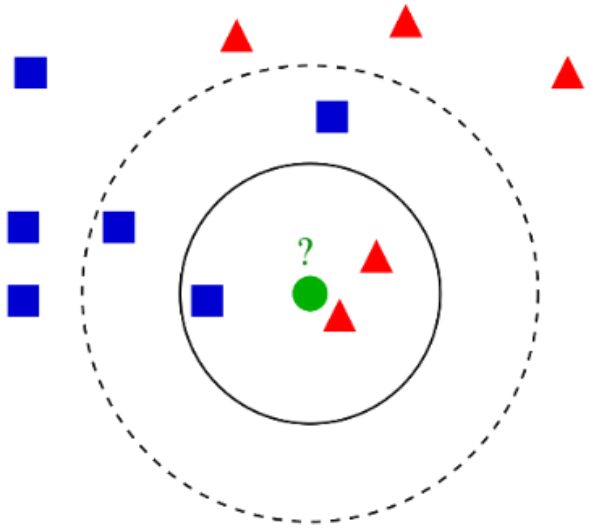
K en yakın komşu algoritması (K-NN), ilk olarak 1951'de Evelyn Fix ve Joseph Hodges tarafından geliştirilen ve daha sonra Thomas Cover tarafından genişletilen, parametrik olmayan bir denetimli öğrenme yöntemi sınıflandırma ve regresyon için kullanılmaktadır [53]. Denetimli makine öğrenme yöntemi olarak ifade edilmektedir. Büyük verilerin ortaya çıkmasıyla birlikte, geleneksel K-NN algoritmasının performansı ve verimliliği hızla kritik bir konu haline gelmiştir [54].

K-NN algoritması, önceden etiketlenmiş bir veri kümesini kullanarak, dışarıdan gelecek yeni bir verinin, mevcut veri noktalarına olan uzaklığı hesaplanarak, en yakın *K* adet komşusuna bakılarak sınıflandırma yapar. Bu algoritma, temel olarak iki faktör olan uzaklık ve komşuluk sayısı üzerinden tahminler gerçekleştirir.

Uzaklık: Belirli bir noktanın diğer noktalardan uzaklığını belirlemeyi ifade eder. Bu uzaklığı ölçmek için, Öklid, Manhattan ve Minkowski gibi yöntemler kullanılmaktadır. Bu yaklaşımlar, noktalar arasındaki farklı özellikleri dikkate alarak uzaklık değerini ortaya çıkarmaktadır.

K (Komşuluk Sayısı)*:* En yakın kaç komşu üzerinden hesaplamanın yapılacağını belirlemek için kullanılır. *K,* 1 olursa aşırı öğrenme gerçekleşebilir. Daha yüksek değerler verilmesi halinde daha olası sonuçlar vermesi mümkündür. Bu yüzden, en

uygun *K* değerini tahmin etmek te kendi başına bir zorluktur. Şekil 4.1'de eğer *K* değeri 3 olarak seçilirse, sınıflandırma algoritması "?" işareti ile gösterilen yeşil noktayı kırmızı üçgen sınıfı olarak tanımlayacaktır. Ancak, *K* değeri 5 olarak seçilirse, sınıflandırma algoritması "?" işareti ile gösterilen yeşil noktayı mavi kare sınıfı olarak tanımlayacaktır.



Şekil 4.1. K-NN örneği [55].

Çalışmada, Scikit-learn kütüphanesinin *KNeighborsClassifier* sınıflandırıcısı kullanılmıştır.

K-NN algoritmasının avantajları:

- Eğitim işlemlerinin diğer algoritmalara göre daha kolay olması.
- Süreçlerin ve analizlerin analitik/sayısal olarak takibinin kolay olması.
- Gürültülü eğitim verilerine karşı etkili olması.
- Uygulamanın kolay olması.

K-NN algoritmasının dezavantajları:

- Uygun uzaklık hesabı algoritmasının bulunmaya çalışılması zaman kaybına yol açmaktadır.

- Geniş ölçekli verilere karşı dirençli olsamısına rağmen işlem adımı fazla olduğundan zaman almaktadır.
- İşlem hacmi ve işlem adımı fazla olduğundan dolayı güçlü bilgisayarlara ihtiyaç duyar.

4.2. NAIVE BAYES (NB)

Adını matematikçi Thomas Bayes'ten alan NB, verileri olasılık ilkelerini kullanarak sınıflandırır [56]. Naive Bayes algoritması, çoğunlukla DDİ'de kullanılan olasılık temelli bir algoritmadır. NB, verilen bir örnek için her sınıfın olasılığını ayrı ayrı hesaplar ve örneği en yüksek olasılığa sahip sınıfa atar. Bu hesaplama Bayes Formülü aracığılıyla gerçekleştirilir (Denklem 4.1).

$$P(c\,|x) = P(x|c)\,P(c)/P(x) \tag{4.1}$$

c: Tahmin edilmeye çalışılan sınıf

x: Tahmin eden sınıf

P(c|x): x olayı gerçekleştiğinde c olayının gerçekleşme olasılığı

P(x|c): c olayı gerçekleştiğinde x olayının gerçekleşme olasılığı

P(c): c olayının gerçekleşme olasılığı

P(x): x olayının gerçekleşme olasılığı

Naive Bayes Türleri:

- Bernoulli Naive Bayes: Bernoulli NB'de sınıflar ikili olarak değerlendirilmektedir.
- Multinomial Naive Bayes: MNB algoritması çok sınıflı kategorilerde sınıflandırma için kullanılmaktadır.

- Gaussian Naive Bayes**:** Özellikler sürekli verilerden oluştuğunda, genellikle bu verilerin Gauss dağılımına, yani normal dağılıma uyduğu varsayılır. Gauss Naive Bayes, her biri Gauss dağılımına uyan sürekli değerli özellikleri ve bu özellikleri destekleyen modelleri içeren bir yöntemdir.

Bu tez çalışmasında Multinominal NB model kullanılmıştır, Multinominal NB yalnızca Bayes teoreminden elde edilen olasılığa dayalı karar fonksiyonunun değerlendirilmesine dayanır [57].

NB sınıflandırıcıların avantajları:

- Lojistik regresyon gibi modellerin aksine, özelliklerin birbirine bağımlı olmadığını varsayarak daha iyi performans sergileyebilir.
- Veri miktarı az olsa bile başarılı sonuçlar elde edebilir.
- Hem kesikli hem de sürekli verilerle çalışabilir.
- Yüksek boyutlulukta verilerle de iyi sonuçlar verebilir.

Naive Bayes sınıflandırıcıların dezavantajları:

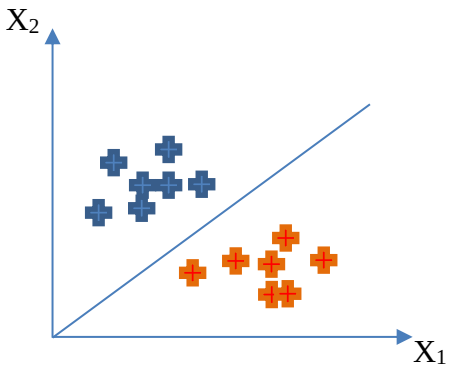
- Özelliklerin birbirinden ayrı olduğunu varsayarak işlem yapıldığında, değişkenlerin nasıl etkileştiğini modellemek mümkün olmaz.
- Sıfır olasılık problemiyle karşılaşılabilir. Naive Bayes sınıflandırıcılarında, incelenen örneğin veri setinde hiç bulunmaması durumuna sıfır olasılık problemi denir. Bu sorunun çözümü Laplace gibi yöntemlerin kullanılmasıdır.

4.3. DESTEK VEKTÖR MAKİNELERİ (SVM)

Destek Vektör Makineleri, günümüzde çoğu sınıflandırma probleminin çözümünde başarılı sonuç almış, verimli makine öğrenmesi algoritmalarından biri olarak literature girmiştir [58]. Cortes ve Vapnik tarafından önerilen SVM, sınıflandırma problemlerini ele almak için araştırmacılar tarafından yaygın olarak kullanılmaktadır. SVM, sınıflandırma ve regresyon görevleri için kullanılan denetimli bir makine öğrenmesi algoritmasıdır [59,60]. SVM başlangıçta ikili sınıflandırma görevleri için

tasarlanmıştır. Bununla birlikte, birçok araştırmacı bu baskın tekniği kullanarak çok sınıflı problemler üzerinde çalışmaktadır [61]. Birden çok sürekli ve kategorik değişkeni kolayca işleyebilir. SVM, çekirdekleri kullanarak verileri yüksek boyutlu bir özellik uzayına yansıtır ve verileri ayrı kategorilere bölen hiper düzlemleri tanımlar. Bu hiper düzlemler belirlenirken amaç, farklı kategorilerdeki en yakın noktalar arasındaki mesafeyi maksimize etmektir. Bu yaklaşım, gürültüye ve aykırı değerlere karşı daha dirençli olan sağlam bir sınıflandırma modeli sağlar. Şekil 4.2'de görüldüğü gibi, hiper düzlem veri setini doğrusal olarak böler ve sınıflar. Veriler ne kadar hiper düzlemden uzak olursa, sınıflandırma o kadar doğru olur. Bu yüzden, verilerin hiper düzleme en uzak doğrularda olması performansı yükseltir. Verileri bir düzlemle ayırmak gerekir, eğer düzlemle ayırma işlemi yapılamıyorsa boyut arttırma işlemi yapılabilir. Bunu yapmak için literatürde çeşitli çekirdek fonksiyonları vardır. Örneğin; Doğrusal, Polinom, Gauss, Laplace, Sigmoid, Bessel çekirdek fonksiyonları.

SVM iki tür olabilir; Doğrusal SVM ve Doğrusal Olmayan SVM. Doğrusal SVM, verilerin bir düz çizgi ile iki sınıfa ayrılabildiği durumlarda kullanılır. Bu verilere doğrusal ayrılabilir veriler denir ve sınıflandırma işlemi için Linear SVM sınıflandırıcı uygulanır. Diğer bir taraftan, Doğrusal Olmayan SVM; verilerin düz bir çizgi ile ayrılamadığı durumlar için geçerlidir. Başka bir deyişle, veriler doğrusal bir çizgi ile sınıflandırılamıyorsa, bu veri tipine doğrusal olmayan veri denir ve Non-Linear SVM sınıflandırıcı bu tür verileri sınıflandırmak için kullanılır.



Şekil 4.2. Doğrusal SVM modeli.

Scikit-learn kütüphanesinde SVM için 3 tür sınıflandırıcı bulunmaktadır. SVC (Support Vector Classifier), NuSVC ve LinearSVC bir veri kümesi üzerinde ikili ve

çok sınıflı sınıflandırma yapabilen sınıflardır. SVC ve NuSVC benzer yöntemlerdir, ancak farklı parametre gruplarını kabul ederler ve farklı matematiksel formülasyonlara sahiptirler.

Çalışmada scikit-learn kütüphanesinin SVC sınıfı kullanılmış olup SVC sınıflandırıcısında kernel parametresi olarak '*poly'* seçilmiştir.

SVM'nin avantajları:

- Farklı çekirdek fonksiyonları (kernel) kullanılır.
- Bellekten tasarruf ederek verimli bir şekilde çalışır.
- Yüksek boyutlu uzaylarda verileri iyi sınıflandırabilir.
- Boyut sayısının örnek sayısından fazla olduğu durumlarda etkilidir.

SVM'nin dezavantajları:

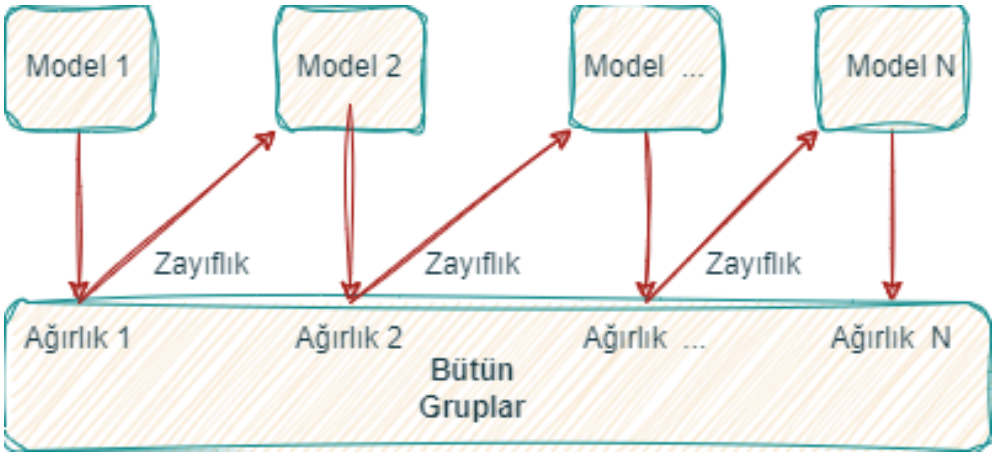
- SVM'nin, çok büyük veri kümelerinde uygulanması zor ve zaman alıcıdır.
- Gürültülü verilerde iyi performans göstermez ve aşırı öğrenme gerçekleşebilir.
- Her bir veri noktası için özellik sayısının, eğitim veri örneklerinin sayısından fazla olduğu durumlarda, SVM düşük performans gösterir.
- SVM, veri noktalarını sınıflandırma hiper düzleminin üstüne ve altına yerleştirerek çalıştığından, sınıflandırma için olasılıksal bir açıklama sunmaz.

4.4. ADAPTIVE BOOSTING (ADABOOST)

AdaBoost, sınıflandırma problemleri için kullanılan popüler bir topluluk makine öğrenme algoritmasıdır. Birden fazla zayıf sınıflandırıcıyı birleştirerek sağlam bir sınıflandırma modeli oluşturur [62]. AdaBoost, her bir veri kümesine eşit ağırlık vererek başlar. Sonra, her karar ağacı sonrasında veri noktalarının ağırlıklarını kendiliğinden değiştirir [63]. Doğru sınıflandırılamayan nesneler için daha yüksek ağırlık verir ve bir sonraki turda onları düzeltmeye çalışır. Artık hata veya gerçek değer ile tahmin değeri arasındaki fark, belirlenen bir sınırın altına inene kadar bu işlemi yineler.

Bu tez çalışmasında scikit-learn kütüphanesinin *AdaBoostClassifier* sınıfından yararlanılmıştır. Sınıflandırıcının temel tahmin algoritması olarak *ExtraTreeClassifier* seçilmiştir.

Extra Trees Classifier temelde karar ağaçlarına dayalı bir topluluk öğrenme yöntemidir. ExtraTrees algoritması, klasik yukarıdan aşağıya prosedüre göre budanmamış karar veya regresyon ağaçlarından oluşan bir grup oluşturur [64]. ExtraTreesClassifier, RF gibi verilerden aşırı öğrenmeyi ve aşırı uydurmayı en aza indirgemek için belirli kararları ve veri alt kümelerini rastgele gerçekleştirir. Şekil 4.3'te AdaBoost'un çalışma şekli gösterilmiştir.



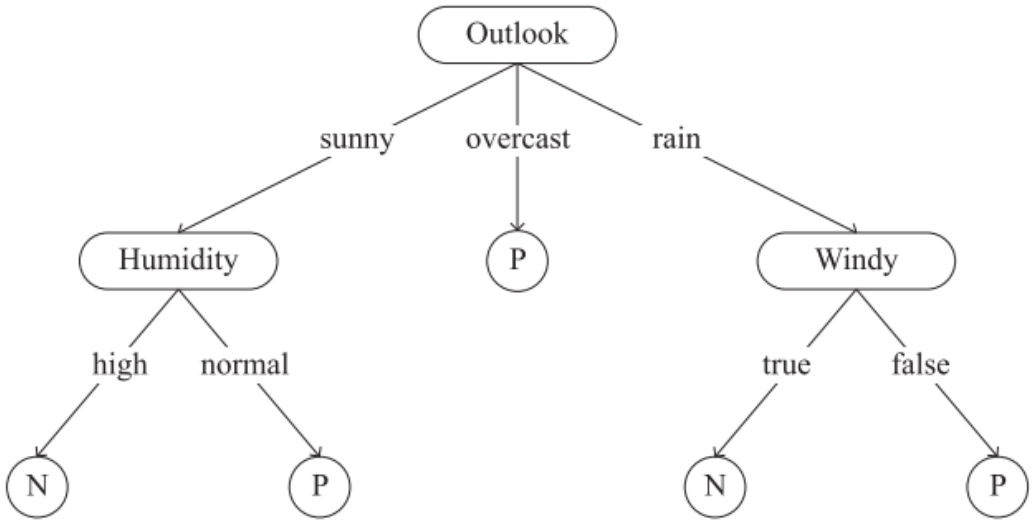
Şekil 4.3. AdaBoost çalışma mimarisi.

4.5. KARAR AĞAÇLARI (DT)

DT, parametrik olmayan bir denetimli öğrenme yöntemidir ve sınıflandırma ve regresyon için kullanılır [65]. Temel strateji, veri noktalarını özniteliklere göre sınıflandırmak için bir ağaç yapısı kurmaktır. Bir karar ağacının ana zorluğu, hangi özniteliğin veya özelliklerin üst düzeyde veya alt düzeyde yer alacağına karar vermektir. Amaç, veri özelliklerinden elde edilen basit karar kurallarını kullanarak bir hedef değişkenin değerini tahmin eden bir model geliştirmektir. Bir ağaç, parçalı sabit bir yaklaşım olarak görülebilir. Karar ağacı öğrenme ve tahmin için çok hızlı bir algoritmadır, ancak aynı zamanda verilerdeki küçük bozulmalara karşı son derece hassastır ve aşırı öğrenme gerçekleşebilir [66]. Modeli genelleştirmek için kökteki tüm

verilerle başlayan ve kademeli olarak farklı özelliklere bölünen karar ağaçlarını kullanır. Karar ağacındaki ayrımların sağlamlığı, entropi değeri veya verilerdeki gürültü derecesi ile ölçülür. Basit ve anlaşılması kolaydır ve hem sürekli hem de kategorik verileri işleyebilir ve nispeten daha az eğitim süresine sahiptir.

Şekil 4.4'te bir karar ağacı örneği gösterilmiştir. Bu örnekte iç düğümler (kök düğüm dahil) öznitelik adlarıyla temsil edilir ve iç düğümlerden oluşan dallar, bir düğümdeki özniteliğin farklı değerlerine karşılık gelir; yaprak düğümleri, farklı sınıf etiketleriyle temsil edilir. Bu örnekte üç özellik mevcuttur: Görünüm {güneşli, bulutlu, yağmur}, Nem {yüksek, normal} ve Rüzgarlı {doğru, yanlış}; ve iki sınıf etiketi {N, P} [67].



Şekil 4.4. Hava durumunu gösteren karar ağacı [67].

Bu tez çalışmasında, scikit-learn kütüphanesinin *DecisionTreeClassifier* sınıflandırıcısı kullanılmıştır.

DT'nin avantajları:

- Kolay anlaşılır ve yorumlanabilir bir yöntemdir. Ağaçlar görsel olarak sunulabilir.

- Veri ön işleme gerektirmez. Diğer yöntemler genellikle verinin normalleştirilmesini, sahte değişken oluşturulmasını ve eksik değerlerin silinmesini gerektirir.
- Çoklu sınıflandırma için uygundur.
- İstatistiksel testlerle bir modelin doğruluğunu test etmeyi sağlar. Bu, modelin güvenilirliğini gösterir.
- Verilerin gerçek modele uygunluğunu varsaymaz, verilerin gerçek modelden biraz sapması durumunda bile iyi çalışır.

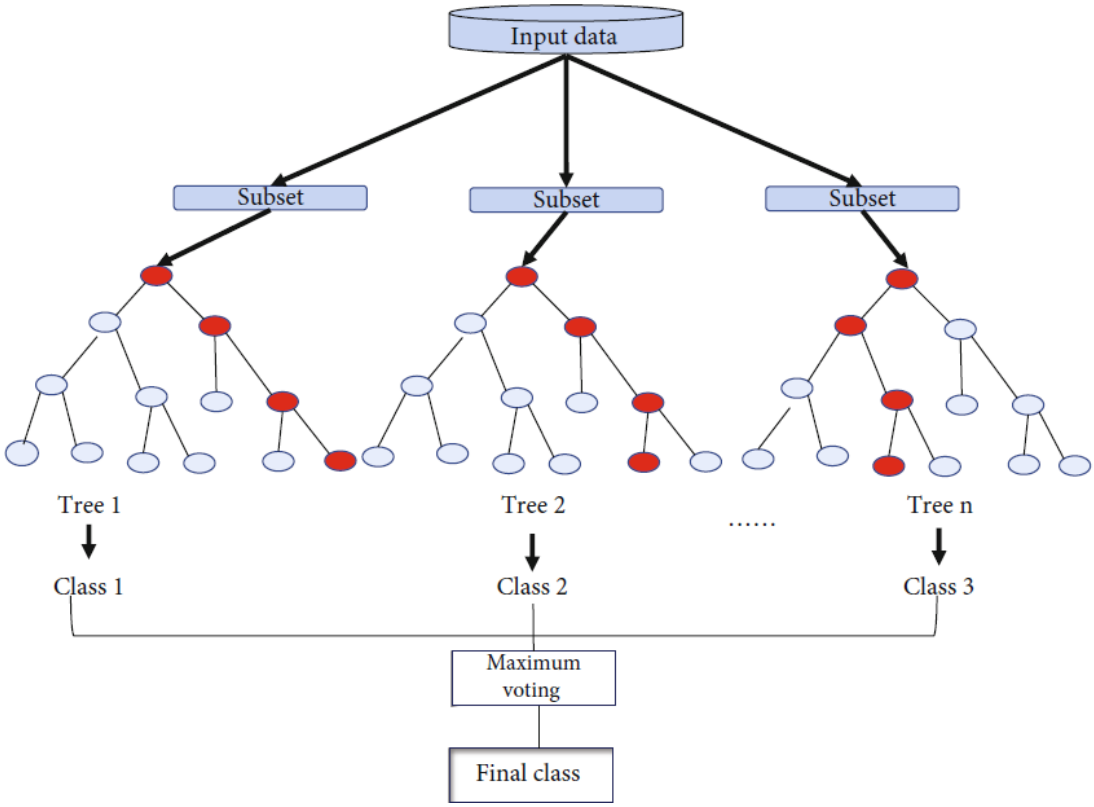
DT'nin dezavantajları:

- Karar ağaçları öğrenme sırasında, verileri genelleştirmeyen çok karmaşık ağaçlar üretebilir. Bu sorunu çözmek için budama, yaprak düğümlerindeki minimum örnek sayısını belirleme veya ağacın maksimum derinliğini sınırlama gibi teknikler kullanılabilir.
- Verilerdeki küçük değişikliklere çok duyarlıdır ve farklı ağaçlar oluşturabilir. Bu sorun, karar ağaçlarını bir arada kullanan topluluk yöntemleriyle giderilebilir.
- Karar ağacı öğrenme sırasında, bazı sınıflar diğerlerinden daha fazlaysa, ağaçlar önyargılı olabilir. Bu yüzden, karar ağacı öğrenmeden önce veri setinin dengeli olması gerekir.

4.6. RASTGELE ORMAN (RF)

RF, çok sayıda karar ağacının birleştirilerek en iyi sonucu veren değerin seçildiği bir yöntemdir [68]**.** RF, denetimli öğrenme kapsamında, sınıflandırma, regresyon gibi çeşitli görevler için eğitim sürecinde birçok karar ağacı oluşturur ve problemin türüne göre sınıflandırma veya regresyon tahmini yapar. RF'nin temel mantığı, rastgele ve birden fazla karar ağacı üreterek sınıflandırma performansını artırmaktır. RF'de, her düğümde en iyi bölünmeyi sağlayan tahmin edici, rastgele seçilen tahmin edicilerin bir alt kümesi içinden belirlenir [69]. Bu şekilde RF, karar ağaçlarının aşırı öğrenme problemine bir çözüm sunar. RF, birçok alanda başarıyla kullanılmış, görüntü ve konuşma tanıma gibi pek çok sınıflandırma problemine etkili bir şekilde uygulanmıştır [70].

Şekil 4.5'te RF yönteminin çalışma mantığı gösterilmiştir. Ağaç sayısı ne kadar fazla olursa, modelin doğru tahmin yapma olasılığı da o kadar yükselmektedir. RF, n adet karar ağacından oluşur. Bu n adet karar ağacı, modelin son aşamasında bir oylamaya tabi tutulur. En çok tekrar eden tahmin, başarılı tahmin olarak kabul edilir.



Şekil 4.5. RF mimarisi [71].

RF avantajları:

- İyi tahminleme oluşturulmasını sağlayan güçlü hiperparametreler kullanır.
- Aynı anda birden fazla işlem yürütülebilir.
- Sınıflandırma ve Regresyon problemlerinde kullanılabilmektedir.
- Aşırı öğrenmeyi azaltır.
- Büyü veri setlerine uygulanabilir.
- Gürültüden fazla etkilenmemektedir.

RF dezavantajları:

- Çok sayıda orman oluşturulması iyi bir tahmin yapmasını engelleyebilir.
- Öğrenme hızlı, tahmin yavaş gerçekleşir.
- Veriler çok seyrek olduğunda iyi sonuçlar üretmemektedir,
- Tanımlamadan çok tahminde bulunmaya dayalı kullanılmaktadır.

Bu tez çalışmasında scikit-learn kütüphanesinin *RandomForestClassifier* sınıfı kullanılmıştır.

BÖLÜM 5

DENEYSEL ÇALIŞMALAR

Bu bölümde, tez çalışmasının amaçladığı metin sınıflandırma modelinin performansını ölçmek için yapılan deneyler ve elde edilen sonuçlar anlatılmıştır. Sonuçların güvenilirliğini sağlamak için, her bir sınıflandırıcı farklı üç metin sayısallaştırma yöntemiyle 20 defa çalıştırılmış ve ortaya çıkan ortalama değerler, standart sapmalarıyla birlikte gösterilmiştir.

Bu çalışmada, sınıflandırıcı modellerin sağlamlığını artırmak ve makine öğrenmesi algoritmalarının performansını objektif bir şekilde ölçmek amacıyla k-kat çapraz doğrulama yöntemi kullanılmıştır. Bu yöntem, veri kümesini k farklı eğitim/test alt kümesine ayırarak, modeli tek bir eğitim/test alt kümesi yerine birden fazla alt kümede test eder. Bu sayede, modelin aşırı öğrenme riskini azaltarak daha güvenilir sonuçlar vermesi sağlanır. Model, veri kümesinin farklı eğitim/test bölümlerinde eğitilerek, çapraz doğrulama işleminin daha istikrarlı ve sağlam olması amaçlanmıştır. Bu çalışmada, 5 katlı çapraz doğrulama gerçekleştirilmiştir; yani model oluşturma süreci beş defa yinelenmiştir. Her katlamada, veri kümesi %80 eğitim ve %20 test alt kümelerine ayrılmıştır.

5.1. ÇALIŞMA ORTAMI

Çalışmalar, üzerinde Windows 10 Enterprise 64-bit işletim sistemi olan HP ProOne 440 All-in-One bilgisayarda gerçekleştirildi. Bilgisayar Inte Core i7-10700 CPU 2.90GHz, 8 Core(s) 16 mantıksal işlemci, 32 GB RAM ve SSD'ye sahiptir. Bu çalışmada kodlama Spyder (Anaconda3 4.10.3 versiyon) arayüzünde Python programlama dili ile gerçekleştirilmiştir. Veri temizleme, düzenleme, harf küçültme, sayılar ve çekim eklerinin kaldırılması gibi işlemler Pandas 1.3.4 versiyonu, Numpy 1.20.3 versiyonu ve nltk 3.6.2 versiyonu kütüphaneleri kullanılmıştır [72]. Sınıflandırma algoritmaları, çapraz doğrulama ve değerlendirme metrikleri için ise

scikit-learn kütüphanesi 0.24.2 versiyonu kullanılmıştır [73]. Deneylerde kullanılan sınıflandırıcıların optimum parametreleri Çizelge 5.1'de listelenmiştir.

Çizelge 5.1. Deneyde kullanılan her sınıflandırıcının optimum parametre değerleri.

Metod	**Parametre**	**Değer**
K-NN	n-neighbors	5
NB	alpha	1.0
SVM	kernel	poly
	degree	3
	c	1.75
	gamma	scale
	coef0	0.0
AdaBoost	base_estimator	ExtraTreeClassifier
	n_estimators	50
	learning_rate	1.0
DT	min_samples_split	default=2
	min_samples_leaf	default=1
RF	min_samples_split	default=2
	min_samples_leaf	default=1
	n_estimators	15

5.2. VERİ SETİ

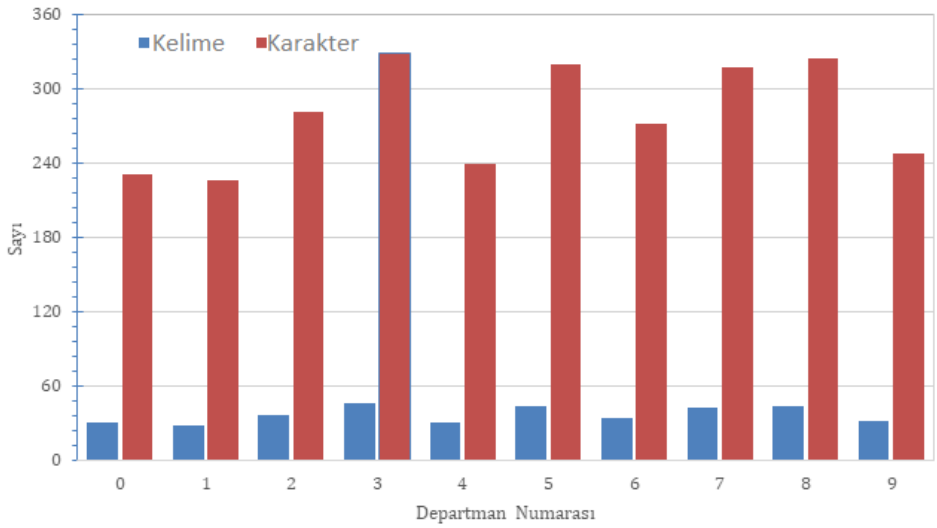
Bu çalışmada kullanılan veri seti, Ticaret Bakanlığı Çağrı Merkezi'nden gelen telefon görüşmelerinin metin çıktılarını da içeren 20.000 adet çağrı kaydından oluşmaktadır. Bu kayıtların tamamı işlenmemiş ve düzeltilmemiş haldedir ve herhangi bir uzunluk kısıtlaması bulunmamaktadır.

Çağrı kayıtları, Ticaret Bakanlığı bünyesinde faaliyet gösteren 10 farklı birimden eşit oranda seçilmiştir. Bu kayıtların birimlere göre dağılımını ve her birimin kayıtlarındaki ortalama kelime ve karakter sayısı Çizelge 5.2'de sunulmuştur. Şekil 5.1, Çizelge 5.2'de belirtilen 10 birimin telefon görüşmesi metinlerindeki ortalama

kelime ve karakter sayısını görselleştirmektedir. Tüm birimlerdeki ortalama kelime ve karakter sayıları karşılaştırıldığında, çağrı metinlerinin sınıflandırılmasını zorlaştırabilecek anlamlı bir farklılık olmadığı tespit edilmiştir. Yeni oluşturulan veri kümesindeki bu dengeli dağılım, metin sınıflandırmasının performansını yükseltmeye ve güvenilir bir model geliştirmeye yardımcı olmuştur.

Çizelge 5.2. Çağrı sayılarının departman genelinde dağılımı.

Departman Numarası	**Çağrı Sayısı**	**Ortalama Kelime Sayısı**	**Ortalama Karakter Sayısı**
0	2000	31	231
1	2000	29	226
2	2000	37	281
3	2000	46	329
4	2000	31	239
5	2000	44	320
6	2000	34	272
7	2000	43	318
8	2000	44	325
9	2000	33	248

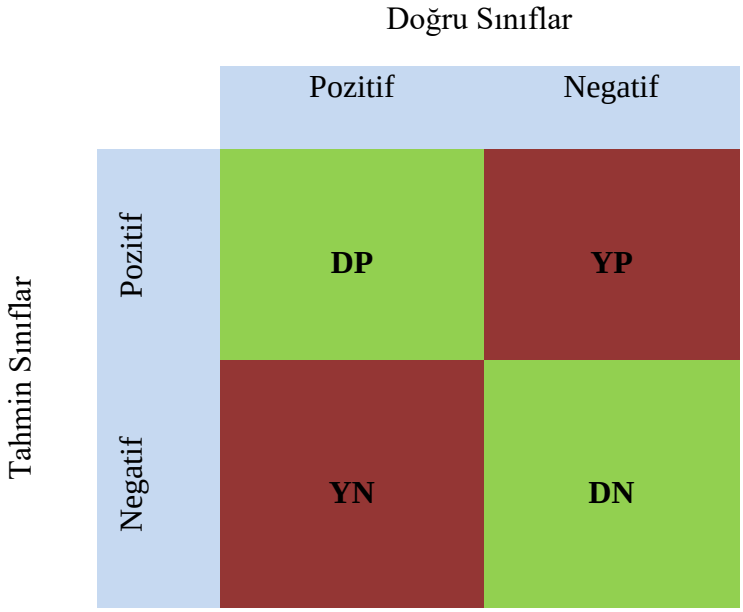


Şekil 5.1. Çağrı metnindeki ortalama kelime ve karakter sayısının 10 departmandaki dağılımı.

5.3. DEĞERLENDİRME METRİKLERİ

Bu tez çalışmasının deneylerinde metin sınıflandırma algoritmalarının etkinliğini değerlendirmek için karmaşıklık matrisinden türetilen doğruluk, kesinlik, duyarlılık ve f1-skor metriklerini kullanılmıştır.

Karmaşıklık Matrisi: Sınıflandırma problemlerinin performansını değerlendirmek için yaygın olarak kullanılan bir tablodur [74]. İki sınıflı veya çok sınıflı sınıflandırma problemlerine uygulanabilir. Karmaşıklık matrisleri, gerçek değerler ile tahmin edilen değerler arasındaki uyumu sayısal olarak gösterir. Doğru Negatif (DN) değeri, negatif sınıfa ait olan örneklerin doğru bir şekilde negatif olarak tahmin edilmesini belirtir. Aynı şekilde Doğru Pozitif (DP) değeri, pozitif sınıfa ait olan örneklerin doğru bir şekilde pozitif olarak tahmin edilmesini belirtir. Yanlış Pozitif (YP) değeri, negatif sınıfa ait olan örneklerin yanlış bir şekilde pozitif olarak tahmin edilmesini belirtir. Yanlış Negatif (YN) değeri ise, pozitif sınıfa ait olan örneklerin yanlış bir şekilde negatif olarak tahmin edilmesini belirtir (Şekil 5.2).



Şekil 5.2. Karmaşıklık matrisi.

Doğruluk: Doğruluk, sınıflandırma modellerini değerlendirmek için kullanılan metriklerden biridir (Denklem 5.1). Sistemde doğru olarak yapılan tahminlerin tüm tahminlere oranıdır.

$$Doğruluk = (DP + DN) / (DP + DN + YN + YP) \quad (5.1)$$

Kesinlik: Kesinlik, pozitif tahmin değeri anlamına gelir. Modelin tahmin ettiği pozitiflerin sayısına kıyasla tahmin ettiği gerçek pozitiflerin sayısının bir ölçüsüdür (Denklem 5.2).

$$Kesinlik = DP /(DP + YP) \quad (5.2)$$

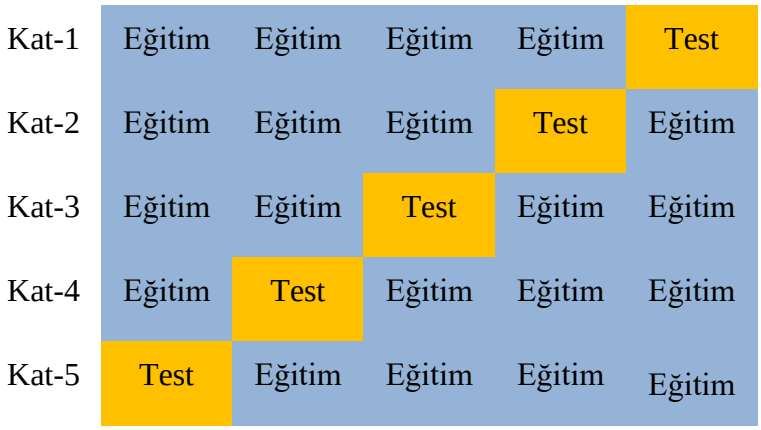
Duyarlılık: Bu metrik, pozitif durumların tahmin edilme başarısını ölçer. Tahmin edilen durumların gerçekleşen durumlarla ne kadar uyumlu olduğunu gösterir. Metrik ne kadar yüksekse, tahminlerin doğruluğu da o kadar yüksektir (Denklem 5.3).

$$Duyarlılık = DP/(DP + YN) \quad (5.3)$$

F1-Puanı: Duyarlılık ve Kesinliğin harmonik ortalamasıdır. İkili sınıflandırma için F1 Puanı Denklem 5.4'de gösterilmiştir.

$$F1 - Puanı = 2 * Kesinlik * Duyarlılık/(Kesinlik + Duyarlılık) \quad (5.4)$$

Çapraz Doğrulama: Makine öğrenmesi modelinin yeni verilerde nasıl davrandığını objektif ve doğru bir şekilde ölçmek için istatistiksel bir yeniden örnekleme yöntemi olan Çapraz Doğrulama kullanılır [75]. Bu yöntem, modelin genelleştirme yeteneğini test etmek ve aşırı öğrenme veya seçim yanlılığı gibi sorunları belirlemek için tasarlanmıştır. Bu çalışmada, scikit-learn kütüphanesindeki *cross_val_score* fonksiyonu ile k - katlı çapraz doğrulama gerçekleştirilmiştir (Şekil 5.3). K değeri 5 olarak alınmış ve model beş defa eğitilmiştir. Her seferinde, veri kümesi %80 eğitim ve %20 test olacak şekilde ikiye ayrılmıştır.



Şekil 5.3. Çapraz doğrulama.

Çalışmada Pyhton ile yazılmış KNN için hazırlanmış çapraz doğrulamanın, doğruluk ve standart sapma değerini gösteren kod parçası Şekil 5.4'te gösterilmiştir.

```
113  cv = StratifiedKFold(n_splits=5, shuffle=True)
114
115  X=Tfidf_df
116  y=sinif
117
118  """K-fold Cross Validation using scikit learn
119  implemented 5 fold cross-validation"""
120
121  clfKNN = KNeighborsClassifier(n_neighbors=5)
122  cross_resultKNN = cross_val_predict(clfKNN, X, y, cv = cv)
123  # results.append(cross_resultKNN)
```

Şekil 5.4. Python çapraz doğrulama kod parçası.

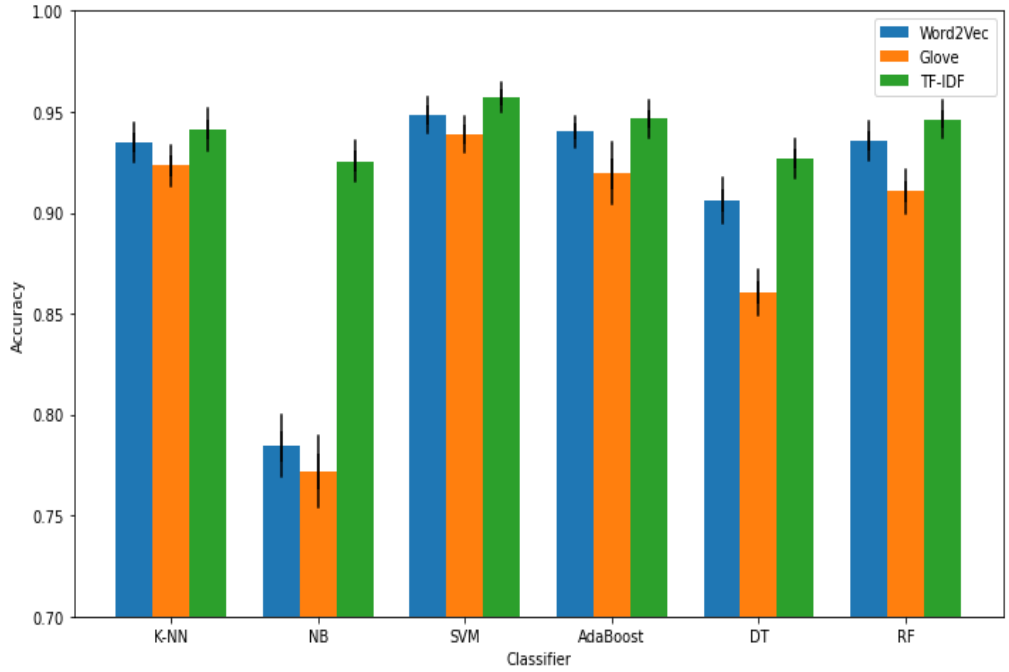
5.4. SONUÇ ANALİZİ

TBÇM çağrı metinleri üzerinde 18 farklı metin sınıflandırma modeli test edilmiştir. Bu modeller, bir metin vektörleştirme yöntemi ile bir sınıflandırıcı algoritmasının kombinasyonundan oluşmaktadır. Şekil 5.5, tüm modellerin hata çubuklarıyla birlikte doğruluk değerlerini göstermektedir. Hata çubuklarının düşük olması, modellerin veri kümesi üzerinde istikrarlı sonuçlar ürettiğini ifade etmektedir. TF-IDF ve Word2Vec için sınıflandırıcıların doğruluk değerleri şu şekilde sıralanmıştır: SVM, AdaBoost, RF, K-NN, DT ve NB. GloVe için ise doğruluk değerleri şöyle sıralanmıştır: SVM, K-NN, AdaBoost, RF, DT ve NB. AS1'i incelediğimizde, SVM'nin her üç metin vektörleştirme yöntemi için de en yüksek doğruluğu sağladığı, NB'nin ise en düşük

doğruluğu sağladığı görülmektedir. Ayrıca, Şekil 5.5'te SVM, K-NN, AdaBoost ve RF'nin yakın sınıflandırma performansına sahip olduğu ve SVM'nin diğerlerine göre daha iyi performans sergilediği tespit edilmiştir.

Çizelge 5.3, doğruluk ölçütünün yanı sıra, sınıflandırma sonuçlarının kesinlik, duyarlılık, f1-skor gibi ölçütlerle ve çalışma süreleriyle birlikte ayrıntılı bir şekilde göstermektedir. Metin vektörleştirme yöntemlerinin etkinliğiyle ilgili olarak, TF-IDF her bir sınıflandırıcı için en yüksek metrik değerlerine ulaşmış, ardından Word2Vec ve GloVe gelmiştir. Metin vektörleştirme yöntemlerinin sıralaması tüm sınıflandırıcılarda TF-IDF, Word2Vec ve GloVe şeklinde olmuştur. Bu sonuç, metin vektörleştirme yöntemlerinin seçimine ilişkin güvenilir bir bilgi sağlamanın yanı sıra AS2'nin de yanıtını vermektedir.

AS3 sorusuna cevap verebilmek amacıyla, 18 farklı sınıflandırıcı ve metin vektörleştirme yönteminin deneysel sonuçları analiz edilmiştir. Bu analize göre, SVM&TF-IDF yöntemi, çağrı metninin ilgili departmanlara en yüksek doğrulukla sınıflandırılmasında %95,7 doğruluk, %95,6 duyarlılık, %95,8 kesinlik ve %95,7 f1-skoru ile en iyi performansı sağlamıştır. Buna karşılık, NB sınıflandırıcısı üç metin vektörleştirme yöntemi için de tüm sınıflandırma ölçütlerinde en düşük performansı göstermiştir. Özellikle, NB'nin Word2Vec ve GloVe yöntemleriyle bir araya gelmesi, diğer yöntemlere göre çok daha kötü bir performans sergilemiştir. Diğer yandan, NB&GloVe, DT&GloVe ve NB&Word2Vec yöntemleri dışında, çalışmadaki çerçeve modellerinin sınıflandırma performansı birbirine yakın olmuştur.



Şekil 5.5. Her model için hata çubuklarıyla birlikte doğruluk değerleri.

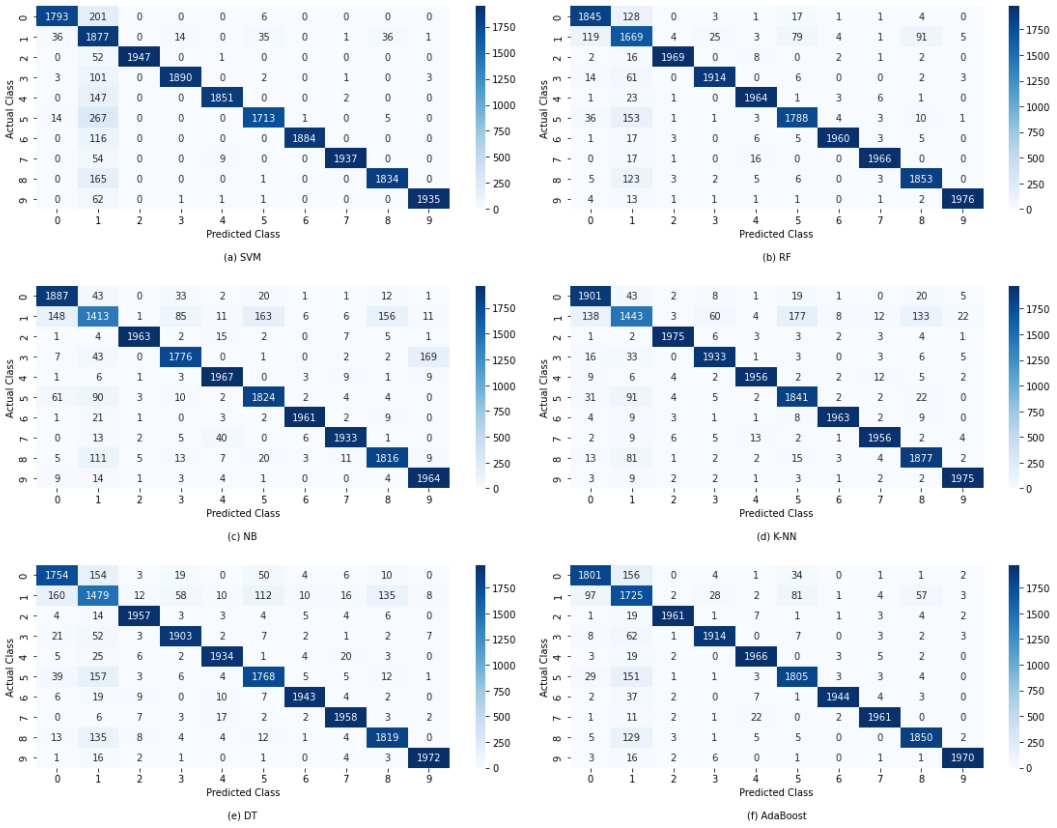
Eğitim ve çalışma süreleri göz önüne alındığında, SVM&TF-IDF ikilisinin en yüksek başarıyı elde etmesine rağmen, diğer yöntemlere kıyasla çok daha fazla zaman harcadığı görülmektedir. Ayrıca, TF-IDF yönteminin eğitim ve çalışma süresi, NB dışındaki tüm sınıflandırıcılar için diğer iki yöntemden daha uzun sürmektedir. Başarı ve zaman kriterlerine göre değerlendirildiğinde, RF&Word2Vec ikilisi sınıflandırma görevinde en uygun alternatif olarak belirlenmiştir. Bu sonuç, RF&Word2Vec ile SVM&TF-IDF ikilileri arasında doğruluk açısından sadece %2'lik bir fark olmasına rağmen, eğitim ve çalışma sürelerinde 9435 katlık büyük bir farkın ortaya çıkmasından kaynaklanmaktadır. SVM yüksek doğruluk sağlamasına rağmen, çağrı merkezlerinde gelen çağrıların hızlı bir şekilde ilgili birime yönlendirilmesi gerektiği için, çalışma süresi bakımından uygun bir sınıflandırıcı olmadığı söylenebilir. AS4 ile ilgili olarak, RF&Word2Vec yönteminin SVM tabanlı yöntemlere göre daha pratik bir çözüm sunduğu deneysel olarak kanıtlanmıştır. Şekil 5.6, 5.7 ve 5.8'de TF-IDF, Word2Vec ve GloVe metin temsil yöntemleri ile her bir sınıflandırıcı için elde edilen karmaşıklık matrisleri verilmiştir. Bu matrislerde, Tablo 2'de belirtilen departmanlara 0 ila 9 arasında sayısal etiketler atanmıştır. Satırlar gerçek departman numaralarını, sütunlar ise tahmin edilen departman numaralarını ifade etmektedir. Bu matrisler, sınıflandırma

modellerinin performanslarını departmanlar arasındaki çağrı dağılımı açısından ayrıntılı bir şekilde analiz etmeyi mümkün kılmaktadır.

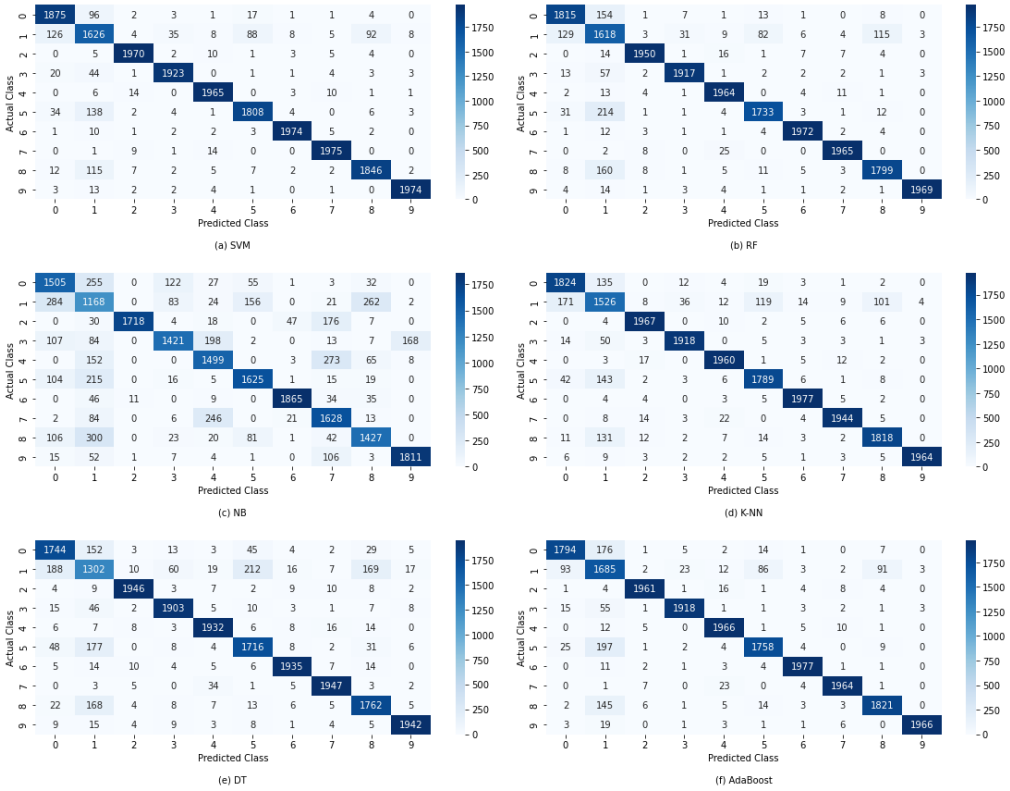
Çizelge 5.3. Performans sonuçları.

Algoritma	Metod	Doğruluk	Kesinlik	Duyarlılık	F1-skor	Eğitim-Çalışma Zamanı (sn)
SVM	TF-IDF	**0,957**	**0,958**	**0,957**	**0,957**	6133.00
	Word2Vec	0,948	0,949	0,948	0,948	45.97
	GloVe	0,939	0,939	0,939	0,939	74.60
RF	TF-IDF	0,946	0,948	0,946	0,947	37.05
	Word2Vec	0,936	0,938	0,936	0,937	**0.65**
	GloVe	0,910	0,918	0,910	0,912	21.67
NB	TF-IDF	0,926	0,925	0,926	0,925	4.74
	Word2Vec	0,784	0,796	0,784	0,788	77.13
	GloVe	0,772	0,786	0,771	0,775	0.64
K-NN	TF-IDF	0,941	0,940	0,941	0,940	91.18
	Word2Vec	0,934	0,935	0,934	0,934	12.57
	GloVe	0,923	0,923	0,923	0,921	12.63
DT	TF-IDF	0,927	0,927	0,927	0,927	87.18
	Word2Vec	0,906	0,906	0,906	0,906	14.80
	GloVe	0,861	0,860	0,862	0,860	43.92
AdaBoost	TF-IDF	0,946	0,950	0,947	0,948	279.80
	Word2Vec	0,940	0,944	0,940	0,941	31.35
	GloVe	0,919	0,927	0,917	0,920	33.48

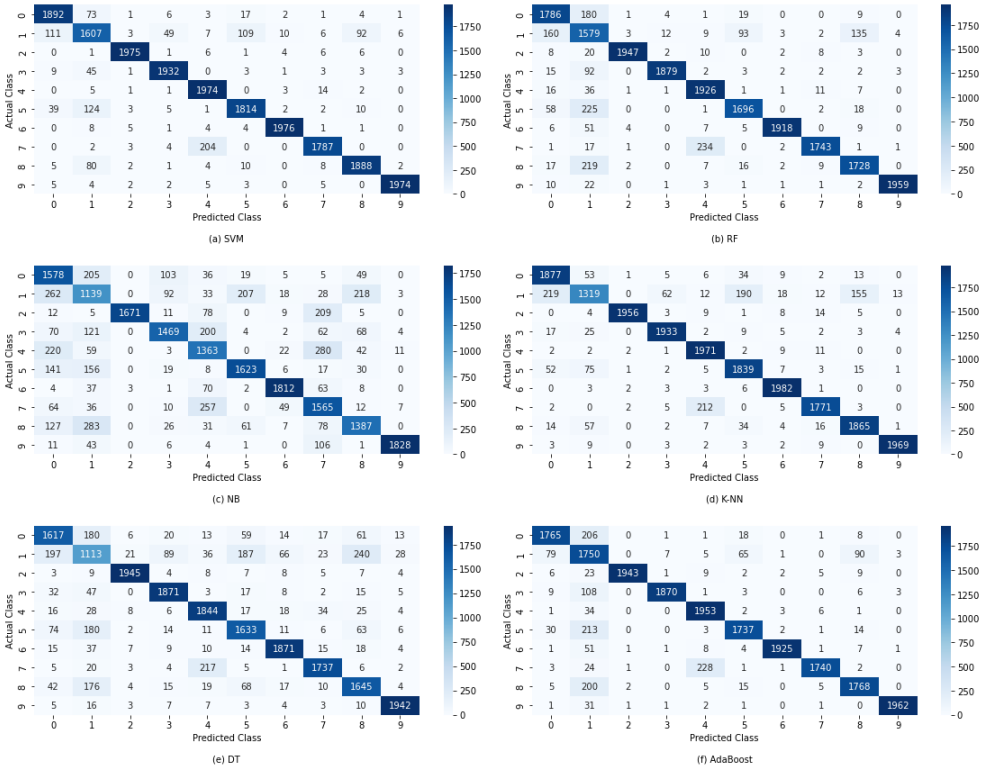
SVM&TF-IDF kombinasyonunun en yüksek doğruluğu sağladığı Şekil 5.6.a'da görüldüğü gibi, 0. Departman için 2000 çağrıdan 1793'ü doğru bir şekilde sınıflandırılmış, ancak 207'si yanlış sınıflandırılmıştır. Bu yanlış sınıflandırılan çağrıların 201'i Departman 1'e, 6'sı ise Departman 5'e hatalı olarak tahmin edilmiştir. Şekil 5.6, 5.7 ve 5.8'de yer alan karmaşıklık matrislerinin tümüne bakıldığında, modellerin Departman 1'i tahmin etmekte çok zorlandığı görülmektedir. Bu durum, çağrı merkezi çalışanlarının Departman 1'e yönelik ifadelerinde daha net ve belirgin bir dil kullanmaları gerektiğini ortaya koymaktadır. Böylelikle modellerin performansı artacak ve daha doğru ve verimli sınıflandırma sonuçları elde edilebilecektir.



Şekil 5.6. Tüm sınıflandırıcılar için TF-IDF karmaşıklık matrisi.



Şekil 5.7. Tüm sınıflandırıcılar için Word2Vec karmaşıklık matrisi.



Şekil 5.8. Tüm sınıflandırıcılar için GloVe karmaşıklık matrisleri.

BÖLÜM 6

SONUÇLAR

Bu tez çalışmasında, çağrı merkezlerindeki yanlış yönlendirme ve müşteri memnuniyeti sorunlarını çözmek için bir akıllı çağrı merkezi sistemi tasarlanmıştır. Bu sistem, TBÇM'nin her gün ortalama 10.000 çağrısının verisini analiz ederek, gelen çağrıların metin içeriklerini GloVe, TF-IDF ve Word2Vec gibi çeşitli metin temsil teknikleriyle sayısal biçimlere çevirmektedir. Daha sonra, AdaBoost, K-NN, SVM, DT, NB ve RF gibi altı farklı makine öğrenmesi algoritması, çağrı metinlerini doğru birime yönlendirmek için sınıflandırma gerçekleştirmektedir.

Bu çalışmada, çeşitli sınıflandırma ve metin vektörleştirme yöntemlerinin performansı ve uygulanabilirliği karşılaştırılmıştır. Bu yöntemlerin çağrı merkezi verilerine uygulanmasıyla elde edilen sonuçlar, bu alanda kullanılabilecek en uygun ve etkili sınıflandırma ve metin temsil teknikleri hakkında değerli bilgiler sunmuştur.

- SVM en yüksek sınıflandırma başarısını gösterirken, eğitim ve çalışma zamanı açısından gerçek zamanlı uygulamalarda yeterli olmadığı görülmüştür.
- Hem eğitim-çalışma zamanı hem de sınıflandırma başarısı dikkate alındığında, RF&Word2Vec kombinasyonu TBÇM'de akıllı bir yardımcı olarak kullanılmak için en uygun model olarak ortaya çıkmıştır.

Böylece, önerilen sistem çağrı merkezleri için gerçek zamanlı, otomatik, Türkçe dil destekli bir çözüm sunabilmiştir.

Çalışma sürecinde karşılaşılan zorluklardan biri, son eklerin kelimelerin anlamlarını değiştirmesinden dolayı Türkçe'nin karmaşıklığının artmasıdır. Bu durum, Word2Vec ve GloVe gibi kelime gömme teknikleri üzerinde önemli bir kısıtlama oluşturmaktadır. Türkçe metinlerin işlenmesindeki bu kısıtlamanın ele alınması için daha fazla

araştırma yapılması ve Türkçe veri ön işleme araçlarının sayısının artması gerekmektedir. Öte yandan, ileriye yönelik bir plan olarak, son zamanlarda oldukça popüler hale gelen büyük dil modelleri, Türkçe metinlerin sınıflandırılmasındaki başarılarını araştırmak için metin gömme yöntemleri olarak kullanılabilir.

KAYNAKLAR

1. Yu, B., Deng, C., and Bu, L., "Policy Text Classification Algorithm Based on Bert", ***Proceedings - 2022 11th International Conference Of Information And Communication Technology, ICTech 2022***, 488–491 (2022).

2. Kocabaş, İ., "Çağrı Merkezi Müşteri Temsilcisinin İmajının Müşteri Memnuniyeti Üzerindeki Rolü", ***Gümüşhane Üniversitesi İletişim Fakültesi Elektronik Dergisi***, 5 (1): 118–118 (2017).

3. Ananthram, S., Xerri, M. J., Teo, S. T. T., and Connell, J., "High-performance work systems and employee outcomes in Indian call centres: a mediation approach", ***Personnel Review***, 47 (4): 931–950 (2018).

4. Chaudhary, S., Nasir, N., Ur Rahman, S., and Masood Sheikh, S., "Impact of Work Load and Stress in Call Center Employees: Evidence from Call Center Employees", ***Pakistan Journal Of Humanities And Social Sciences***, 11 (1): 160–171 (2023).

5. Keser, A. and Yilmaz, G., "Workload , Burnout , and Job Satisfaction Among Call Center Employees", ***Journal Of Social Policy Conferences***, (66–67): 1–13 (2014).

6. Patterson, P. G., Johnson, L. W., and Spreng, R. A., "Modeling the Determinants of Customer Satisfaction for Business-to-Business Professional Services", ***Journal Of The Academy Of Marketing Science***, 25 (1): 4–17 (1996).

7. Pallotta, V., Delmonte, R., Vrieling, L., and Walker, D., "Interaction Mining: The new frontier of Call Center Analytics", ***CEUR Workshop Proceedings***, 771 (October): (2011).

8. Park, Y. and Gates, S. C., "Towards real-time measurement of customer satisfaction using automatically generated call transcripts", ***International Conference On Information And Knowledge Management, Proceedings***, 24754: 1387–1396 (2009).

9. Chowdhury, S. A., Stepanov, E. A., and Riccardi, G., "Predicting User Satisfaction from Turn-Taking in Spoken Conversations.", ***Interspeech***, 2910–2914 (2016).

10. Luque, J., Segura, C., Sanchez, A., Umbert, M., and Galindo, L. A., "The role of linguistic and prosodic cues on the prediction of self-reported satisfaction in contact centre phone calls", ***Proceedings Of The Annual Conference Of The***

International Speech Communication Association, INTERSPEECH, 2017-Augus: 2346–2350 (2017).

11. Chatterjee, J., Saxena, A., and Vyas, G., "An automatic and robust system for identification of problematic call centre conversations", ***Proceedings - 2016 International Conference On Micro-Electronics And Telecommunication Engineering, ICMETE 2016***, 325–330 (2016).

12. Meinzer, S., Jensen, U., Thamm, A., Hornegger, J., and Eskofier, B. M., "Can machine learning techniques predict customer dissatisfaction? A feasibility study for the automotive industry", ***Artificial Intelligence Research***, 6 (1): 80 - 90 (2016).

13. Liu, Y., Cao, B., Ma, K., and Fan, J., "Improving the classification of call center service dialogue with key utterences", ***Wireless Networks***, 27 (5): 3395–3406 (2021).

14. Busemann, S., Schmeier, S., and Arens, R. G., "Message classification in the call center", ***ArXiv Preprint Cs/0003060***, (2000).

15. Galanis, D., Karabetsos, S., Koutsombogera, M., Papageorgiou, H., Esposito, A., and Riviello, M. T., "Classification of emotional speech units in call centre interactions", ***4th IEEE International Conference On Cognitive Infocommunications, CogInfoCom 2013 - Proceedings***, 403–406 (2013).

16. Emmanuela, E. P., Tjendra, F. K., Kezia, S., and Suryani, D., "Classification of Customer Satisfaction in Marketplace", ***2023 International Conference On Computer Science, Information Technology And Engineering (ICCoSITE)***, 392–397 (2023).

17. Mousavi, A., Rezaee, M., and Ayanzadeh, R., "A survey on compressive sensing: Classical results and recent advancements", ***Journal Of Mathematical Modeling***, 8 (3): 309–344 (2020).

18. Salminen, J., Hopf, M., Chowdhury, S. A., Jung, S. gyo, Almerekhi, H., and Jansen, B. J., "Developing an online hate classifier for multiple social media platforms", ***Human-Centric Computing And Information Sciences***, 10 (1): 1–34 (2020).

19. Alaoui, R. L. and Nfaoui, E. H., "Web attacks detection using stacked generalization ensemble for LSTMs and word embedding", ***Procedia Computer Science***, 215: 687–696 (2022).

20. Cahyani, D. E. and Patasik, I., "Performance comparison of tf-idf and word2vec models for emotion text classification", ***Bulletin Of Electrical Engineering And Informatics***, 10 (5): 2780–2788 (2021).

21. Akuma, S., Lubem, T., and Adom, I. T., "Comparing Bag of Words and TF-IDF with different models for hate speech detection from live tweets",

International Journal Of Information Technology (Singapore), 14 (7): 3629–3635 (2022).

22. Ekici, B. and TAKCI, H., "Spam Tespitinde Word2Vec ve TF-IDF Yöntemlerinin Karşılaştırılması ve Başarı Oranının Artırılması Üzerine Bir Çalışma", ***Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi***, 8 (2): 646–655 (2021).

23. Koruyan, K. and EKERYILMAZ, A., "Makine Öğrenmesi ile Müşteri Şikayetlerinin Sınıflandırılması", ***AJIT-E: Academic Journal Of Information Technology***, 13 (50): 168–183 (2022).

24. Çelik, Ö. and 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 (2021).

25. Oja, E., "Unsupervised learning in neural computation", ***Theoretical Computer Science***, 287 (1): 187–207 (2002).

26. Internet: IBM Cloud Education, "What Is Unsupervised Learning? | IBM", **https://www.ibm.com/topics/unsupervised-learning** (2023).

27. L. P. Kaelbling, M. L. Littman, and A. W. Moore, "View of Reinforcement Learning: A Survey", ***Journal Of Artificial Intelligence Research***, 4: 237–285 (1996).

28. Verdhan, V., "Supervised learning with python", ***Okänd. Irland: Apress***, (2020).

29. Nasteski, V., "An overview of the supervised machine learning methods", ***Horizons.B***, 4: 51–62 (2017).

30. Saif, H., Fernandez, M., He, Y., and Alani, H., "On stopwords, filtering and data sparsity for sentiment analysis of twitter", ***Proceedings Of The 9th International Conference On Language Resources And Evaluation, LREC 2014***, (i): 810–817 (2014).

31. Tantuğ, A. C., "Metin sınıflandırma", ***Türkiye Bilişim Vakfı Bilgisayar Bilimleri Ve Mühendisliği Dergisi***, 5 (2): (2016).

32. Silva, C. and Ribeiro, B., "The Importance of Stop Word Removal on Recall Values in Text Categorization", ***Proceedings Of The International Joint Conference On Neural Networks***, 3: 1661–1666 (2003).

33. Madatov, K., Bekchanov, S., and Vičič, J., "Dataset of stopwords extracted from Uzbek texts", ***Data In Brief***, 43: 108351 (2022).

34. İnternet: Xia, M. X., "GitHub - Xiamx/Node-Nltk-Stopwords",

https://github.com/xiamx/node-nltk-stopwords (2022).

35. Korenius, T., Laurikkala, J., Järvelin, K., and Juhola, M., "Stemming and lemmatization in the clustering of finnish text documents", ***Proceedings Of The Thirteenth ACM International Conference On Information And Knowledge Management***, 625–633 (2004).

36. Nutu, M., "Deep Learning Approach for Automatic Romanian Lemmatization", ***Procedia Computer Science***, 192: 49–58 (2021).

37. Internet: Barbaresi, A., "Simplemma", **http://doi.org/10.5281/zenodo.4673264** .

38. Rianto, Mutiara, A. B., Wibowo, E. P., and Santosa, P. I., "Improving the accuracy of text classification using stemming method, a case of non-formal Indonesian conversation", ***Journal Of Big Data***, 8 (1): 1–16 (2021).

39. Gupta, G., "Text Document Tokenization for Word Frequency Count using Rapid Miner (Taking Resume as an Example)", ***International Journal Of Computer Applications***, 1 (March 2015): 60–768887 (2009).

40. İnternet: "Kelime - Vikipedi", **https://tr.wikipedia.org/wiki/Kelime** (2023).

41. Liu, Z., Lin, Y., and Sun, M., "Representation Learning for Natural Language Processing", ***Springer Nature***, (2023).

42. Harish, B. S., Guru, D. S., and Manjunath, S., "Representation and classification of text documents: A brief review", ***IJCA, Special Issue On Recent Trends In Image Processing And Pattern Recogniton***, (2): 110–119 (2010).

43. Harris, Z. S., "Distributional Structure", ***Distributional Structure, WORD***, 10 (3): 146–162 (1954).

44. Zhao, R. and Mao, K., "Fuzzy Bag-of-Words Model for Document Representation", ***IEEE Transactions On Fuzzy Systems***, 26 (2): 794–804 (2018).

45. Aljedaani, W., Rustam, F., Mkaouer, M. W., Ghallab, A., Rupapara, V., Washington, P. B., Lee, E., and Ashraf, I., "Sentiment analysis on Twitter data integrating TextBlob and deep learning models: The case of US airline industry", ***Knowledge-Based Systems***, 255: 109780 (2022).

46. Mikolov, T., Chen, K., Corrado, G., and Dean, J., "Efficient estimation of word representations in vector space", ***1st International Conference On Learning Representations, ICLR 2013 - Workshop Track Proceedings***, 1–12 (2013).

47. Singh, A. K. and Shashi, M., "Vectorization of text documents for identifying unifiable news articles", ***International Journal Of Advanced Computer Science And Applications***, 10 (7): 305–310 (2019).

48. Yesiltas, G. and Gungor, T., "Intrinsic and Extrinsic Evaluation of Word Embedding Models", ***Proceedings - 2020 Innovations In Intelligent Systems And Applications Conference, ASYU 2020***, (2020).
49. Jatnika, D., Bijaksana, M. A., and Suryani, A. A., "Word2vec model analysis for semantic similarities in English words", ***Procedia Computer Science***, 157: 160–167 (2019).

50. Le, Q. and Mikolov, T., "Distributed representations of sentences and documents", ***International Conference On Machine Learning***, 1188–1196 (2014).

51. Arslan, H., Kaynar, O., and Şahİn, S., "Classification of Customer Demands by Using Doc2Vec Feaure Extraction Method", ***2019 27th Signal Processing And Communications Applications Conference (SIU)***, 1–4 (2019).

52. Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T., "Enriching Word Vectors with Subword Information", ***Transactions Of The Association For Computational Linguistics***, 5: 135–146 (2017).

53. Cover, T. M. and Hart, P. E., "Nearest Neighbor Pattern Classification", ***IEEE Transactions On Information Theory***, 13 (1): 21–27 (1967).

54. Ali, M., Jung, L. T., Abdel-Aty, A. H., Abubakar, M. Y., Elhoseny, M., and Ali, I., "Semantic-k-NN algorithm: An enhanced version of traditional k-NN algorithm", ***Expert Systems With Applications***, 151: (2020).

55. İnternet: Wikipedia contributors, "K-Nearest Neighbors Algorithm-Wikipedia", **https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm** (2023).

56. Lakoumentas, J., Drakos, J., Karakantza, M., Sakellaropoulos, G., Megalooikonomou, V., and Nikiforidis, G., "Optimizations of the naïve-Bayes classifier for the prognosis of B-Chronic Lymphocytic Leukemia incorporating flow cytometry data", ***Computer Methods And Programs In Biomedicine***, 108 (1): 158–167 (2012).

57. McCallum, A. and Nigam, K., "A Comparison of Event Models for Naive Bayes Text Classification", ***AAAI/ICML-98 Workshop On Learning For Text Categorization***, 41–48 (1998).

58. Ayhan, S. and Erdoğmuş, Ş., "Destek vektör makineleriyle sınıflandırma problemlerinin çözümü için çekirdek fonksiyonu seçimi", ***Eskişehir Osmangazi Üniversitesi İktisadi Ve İdari Bilimler Dergisi***, 9 (1): 175–201 (2014).

59. Cortes, C. and Vapnik, V., "Support-Vector Networks", ***Machine Learning***, 20 (3): 273–297 (1995).

60. Metlek, S. and Kayaalp, K., "Destek Vektör Makineleri", Makine Öğreniminde Sınıflandırma Yöntemleri ve R Uygulamaları, ***İKSAD***, Ankara, (2020).

61. Boser, B. E., Guyon, I. M., and Vapnik, V. N., "Training algorithm for optimal margin classifiers", ***Proceedings Of The Fifth Annual ACM Workshop On Computational Learning Theory***, 144–152 (1992).

62. Freund, Y. and Schapire, R. E., "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting", ***Journal Of Computer And System Sciences***, 55 (1): 119–139 (1997).

63. Wang, R., "AdaBoost for Feature Selection, Classification and Its Relation with SVM, A Review", ***Physics Procedia***, 25: 800–807 (2012).

64. Geurts, P., Ernst, D., and Wehenkel, L., "Extremely randomized trees", ***Machine Learning***, 63 (1): 3–42 (2006).

65. Mehmed Kantardzic, "Data Mining: Concepts, Models, Methods, and Algorithms, Second Edition", ***Data Mining: Concepts, Models, Methods, And Algorithms: Second Edition***, (2002).

66. Quinlan, J. R., "Simplifying decision trees", ***International Journal Of Man-Machine Studies***, 27 (3): 221–234 (1987).

67. Otero, F. E. B., Freitas, A. A., and Johnson, C. G., "Inducing decision trees with an ant colony optimization algorithm", ***Applied Soft Computing Journal***, 12 (11): 3615–3626 (2012).

68. Pavlov, Y. L., "Random Forests", ***Random Forests***, 1–122 (2019).

69. Liaw, A. and Wiener, M., "Classification and Regression by randomForest", ***R News***, 2 (3): 18–22 (2002).

70. Scornet, E., Biau, G., and Vert, J. P., "Consistency of random forests", ***Annals Of Statistics***, 43 (4): 1716–1741 (2015).

71. Allehaibi, K., Daanial Khan, Y., and Khan, S. A., "ITAGPred: A Two-Level Prediction Model for Identification of Angiogenesis and Tumor Angiogenesis Biomarkers", ***Applied Bionics And Biomechanics***, 2021: (2021).

72. İnternet: "NLTK :: Natural Language Toolkit", **https://www.nltk.org/** (2022).

73. İnternet: "User Guide: Contents — Scikit-Learn 1.2.0 Documentation", **https://scikit-learn.org/stable/user_guide.html** (2022).

74. Stehman, S. V., "Selecting and interpreting measures of thematic classification accuracy", ***Remote Sensing Of Environment***, 62 (1): 77–89 (1997).

75. Kohavi, R., "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection", ***International Joint Conference Of Artificial Intelligence***, (1995).

ÖZGEÇMİŞ

2014 yılında Fırat Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nden mezun oldu. 2015-2016 yılları arasında Ziraat Teknoloji'de Yazılım Mühendisi, 2016-2023 yılları arasında Ticaret Bakanlığı Bilgi Teknolojileri Genel Müdürlüğü'nde Ticaret Uzmanı olarak çalıştı. Şu an TÜRKSAT'da IT Proje Yöneticisi olarak görev yapmaktadır. 2021 yılından itibaren Karabük Üniversitesi Lisanüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans eğitimini sürdürmektedir.