T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

DOĞAL DİL İŞLEME VE MAKİNE ÖĞRENMESİ YÖNTEMLERİ KULLANILARAK STİLOMETRİ ANALİZİ İLE TÜRKÇE METİNLERDE ESER – YAZAR EŞLEŞTİRME

İbrahim DOĞAN

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Temmuz-2023
KONYA

TEZ KABUL VE ONAYI

İbrahim DOĞAN tarafından hazırlanan "Doğal Dil İşleme ve Makine Öğrenmesi Yöntemleri Kullanılarak Stilometri Analizi ile Türkçe Metinlerde Eser – Yazar Eşleştirme" adlı tez çalışması 14/07/2023 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Başkan Doç. Dr. Mehmet Akif ŞAHMAN	……..
Danışman Dr. Öğr. Üyesi Sedat KORKMAZ	……..
Üye Doç. Dr. Ersin KAYA	……..

Yukarıdaki sonucu onaylarım.

Prof. Dr. Mevlüt UYAN
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İbrahim DOĞAN

Tarih:13.09.2023

ÖZET

YÜKSEK LİSANS TEZİ

DOĞAL DİL İŞLEME VE MAKİNE ÖĞRENMESİ YÖNTEMLERİ KULLANILARAK STİLOMETRİ ANALİZİ İLE TÜRKÇE METİNLERDE ESER – YAZAR EŞLEŞTİRME

İbrahim DOĞAN

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Sedat KORKMAZ

2023, 97 Sayfa

Jüri
Doç. Dr. Mehmet Akif ŞAHMAN
Doç. Dr. Ersin KAYA
Dr. Öğr. Üyesi Sedat KORKMAZ

Teknolojinin ve internetin yaygınlaşmasıyla birlikte birçok alanda olduğu gibi gazetecilik alanında da değişimler ve gelişmeler meydana gelmiştir. Çevrim içi gazeteler, birçok unsuru içerisinde barındırabilmesi ve daha büyük kitlelere hitap etmesi nedeniyle çok sayıda gazete sahibinin ve bu alanda hizmet veren yazarın yoğun ilgisini üzerine çekmiştir. Bu kapsamda; geleneksel yöntemlerle yayın yapmakta olan gazeteler dijital ortamda hizmet verebilmek için hızla bu değişim sürecine dahil olmuş ve böylelikle internet üzerinden hizmet veren medya kuruluşlarının sayısında kısa süre içerisinde önemli ölçüde artış meydana gelmiştir. Gazetecilik alanında meydana gelen bu gelişmeler, farklı alanlarda eser veren yazarların aynı platform üzerinde okuyucular ile buluşmasına olanak sağlayarak birçok alanda sayısız eser verilmesinde etkili olmuştur. Ancak söz konusu bu olumlu gelişmelerin yanı sıra veri sayısında meydana gelen artış istenilen ve doğru veriye erişimi zorlaştırmıştır. Bunun sonucunda özellikle anlamsız verilerden anlamlı verilerin çıkarılması ve verilerin belirli niteliklere göre sınıflandırılması konuları önem kazanmıştır. Bu çalışmada elektronik ortamda gazete yazarlığı yapan kişilerin yazmış olduğu yazılar sayısal üslup analizi, yapay zekânın bir alt dalı olan doğal dil işleme ile makine öğrenmesi yöntemleri kullanılarak analiz edilmiş ve söz konusu yazıların hangi yazarlar tarafından yazıldığının doğru eşleştirilebilmesi amaçlanmıştır. Veri seti üzerinde veri ön işleme çalışmaları tamamlanarak veri seti optimizasyonu sağlanmış ve daha sonra Zemberek kütüphanesi fonksiyonları kullanılarak doğal dil işleme süreçleri ile öznitelik çıkarma işlemleri gerçekleştirilmiştir. Son olarak çalışmada kullanılan makine öğrenmesi sınıflandırma algoritmalarından hangisinin yazar-eser eşleştirmelerinde daha başarılı olduğuna karar verebilmek için karşılaştırma yapılmış ve model seçiminin buna bağlı olarak belirlenmesi hedeflenmiştir.

Anahtar Kelimeler: Doğal Dil İşleme, Makine Öğrenmesi, Metin Sınıflandırma, Zemberek Kütüphanesi

ABSTRACT

MS THESIS

ARTİFACT – AUTHOR MATCHİNG İN TURKİSH TEXTS WİTH STYLOMETRY ANALYSİS USİNG NATURAL LANGUAGE PROCESSİNG AND MACHİNE LEARNİNG METHODS

İbrahim DOĞAN

Konya Technical University
Institute of Graduate Studies
Department of Computer Engineering

Advisor: Asst.Prof.Dr. Sedat KORKMAZ

2023, 97 Pages

Jury
Assoc.Prof.Dr. Mehmet Akif ŞAHMAN
Assoc.Prof.Dr. Ersin KAYA
Asst.Prof.Dr. Sedat KORKMAZ

With the widespread use of technology and the Internet, changes and developments have occurred in many areas, including the field of journalism. Online newspapers have attracted the attention of many newspaper owners and writers who serve in this field due to their ability to incorporate various elements and appeal to larger audiences. In this context, traditional newspapers that were operating with conventional methods quickly became part of this process of change in order to provide services in the digital environment, resulting in a significant increase in the number of media organizations operating on the Internet. These developments in journalism have enabled writers working in different fields to reach readers on the same platform, leading to numerous works being produced in various areas. However, alongside these positive developments, the increase in data volume has made it difficult to access the desired and accurate data. As a result, extracting meaningful data from meaningless data and classifying data according to specific attributes have become important topics. In this study, articles written by individuals who engage in newspaper writing in the electronic environment were analyzed using methods of numerical style analysis, natural language processing (a subfield of artificial intelligence), and machine learning to accurately match the authors of these articles. Data preprocessing was performed on the dataset to optimize it, followed by feature extraction using the functions of the Zemberek library for natural language processing. Finally, a comparison was made among the machine learning classification algorithms used in the study to determine which one was more successful in author-work matching, and the model selection was intended to be determined accordingly.

Keywords: Machine Learning, Natural Language Processing, Text Classification, Zemberek Library

ÖNSÖZ

Tez çalışmam boyunca engin bilgi ve tecrübeleriyle beni yönlendirerek çalışmamı bilimsel temeller ışığında şekillendirmeme katkı sağlayan, yüksek vizyonu ile akademik hayata farklı pencerelerden bakmama vesile olan ve bu zorlu tez sürecinde karşılaşmış olduğumuz Covid-19 pandemisi ile 100 yılın felaketi olarak adlandırılan 06 Şubat 2023 depremi sonrasında bizleri manevi anlamda destekleyerek cesaretlendiren danışman hocam Sayın Dr. Öğr. Üyesi Sedat KORKMAZ'a teşekkürü bir borç bilir saygılarımı sunarım.

Yapmış olduğum çalışmalarda maddi ve manevi desteğini hiçbir zaman esirgemeyen ve karşılaştığım her zorlukta yanımda olan Sevgili Eşim Esra Betül DOĞAN'a ve aileme en içten teşekkürlerimi sunarım.

İbrahim DOĞAN
KONYA-2023

İÇİNDEKİLER

KISALTMALAR

BOW	:	Kelime Çantası (Bag of Words)
DDA	:	Doğal Dil Anlama
DF	:	Doküman Sıklığı (Document Frequency)
DDİ	:	Doğal Dil İşleme
DDÜ	:	Doğal Dil Üretme
FN	:	Yanlış Negatif (False Negative)
FP	:	Yanlış Pozitif (False Positive)
IDF	:	Ters Doküman Sıklığı (Inverse Document Frequency)
KNN	:	K-En Yakın Komşu (K-Nearest Neighbors)
LSTM		Uzun Kısa Süreli Bellek (Long Short-Term Memory)
MLP	:	Çok Katmanlı Algılayıcılar (Multi-layer Perceptron)
NLP	:	Natural Language Processing
SÜA	:	Sayısal Üslup Analizi (Stilometri)
SVM	:	Destek Vektör Makineleri (Support Vector Machine)
TF	:	Terim Sıklığı (Term Frequency)
TN	:	Gerçek Negatif (True Negative)
TP	:	Gerçek Pozitif (True Positive)
TF-IDF	:	Terim Sıklığı - Ters Doküman Sıklığı (Term Frequency - Inverse Document Frequency)
YSA	:	Yapay Sinir Ağı

1. GİRİŞ

1.1. Dil ve Çeşitleri

Dil; insanların duygu ve düşüncelerini sözcükler veya işaretler kullanarak ifade etmelerini sağlayan gelişmiş bir iletişim aracıdır. Amerikalı ünlü filozof, tarihçi ve dilbilimci olan Avram Noam Chomsky yaptığı çalışmalarda dili *"Sözcük ve cümle birimleri aracılığıyla, düşünceyi konuşmayla ilişkilendiren çok seviyeli bir sistemdir."* olarak tanımlamıştır. (Chomsky, 2014) Diller bilgisayar bilimleri alanında makine dili (Machine language), yapay diller ve doğal diller (Natural language) olmak üzere üç temel başlık altında sınıflandırılmaktadır.

Makine kodu veya nesne kodu olarak da adlandırılan makine dili, bir makinenin okuyup yorumladığı ikili rakamlar (0/1) veya sıralanmış bitler dizisi olmakla birlikte bilgisayarın anlayabileceği en alt seviyedeki tek programlama dilidir. Bilgisayar ve diğer elektronik cihazların birçoğunda yapmış olduğumuz işlemler ve geliştirilmiş üst seviye programlama dilleri (Java, C#, C++, Python, PHP, Ruby vb.) bilgisayar birimlerinin anlayabileceği makine diline dönüştürülmektedir.

Yapay diller ise doğal dillerin aksine kaynağı belli olan, bazı insanlar veya topluluklar tarafından bir amaç doğrultusunda oluşturulmuş ve geliştirilmiş olan dillerdir. Dilbilgisi ve kuralları tarihin akışına göre zamanla değişmez, insanların kabulü ve yönelimleri çerçevesinde evrimleşmeden kurucusunun belirlemiş olduğu şekilde kullanılmaya devam eder. Yapay dilleri kendi aralarında iki farklı kategoride incelemek mümkündür. Bunlar; bilgisayar bilimlerinde kullanılan programlama dilleri (Scala, C, JavaScript, HTML, Matlab vb.) ve bazı toplulukların konuşma dili olarak kullandıkları dillerdir. Bugün dünya üzerinde en yaygın ve en popüler olarak kullanılan (Yaklaşık 1.5 milyon kişi tarafından kullanıldığı tahmin edilmektedir.) yapay dil, 28 harften oluşan ve Polonyalı göz doktoru Ludwik Lejzer Zamenhof tarafından 1887 yılında geliştirilen Esperanto'dur. (Aray, 2020) Esperanto kelimesi Fransızca kökenli olup, umut etmek manasını taşımaktadır. Günümüzde en çok kullanılan ve bilinen yapay diller; Esperanto, Bâleybelen, İdo, Elfçe, Klingon, Toki Pona olarak sıralanabilir.

Doğal dil ise insanların günlük hayatlarında fikirlerini, isteklerini ve düşüncelerini birbirleriyle paylaşmak amacıyla kullanmış oldukları belirli bir kural yapısına sahip çözümlenebilir çok seviyeli bir sistemdir. İngilizce, Almanca, İspanyolca, Fransızca, Çince dünya üzerinde yaygın olarak kullanılan doğal dillere örnek olarak verilebilir.

Dünya üzerindeki yaşayan ve hâlâ kullanılmakta olan dil sayısı tarihlere göre değişiklik göstermektedir. 1951 yılından beri Ethnologue: Languages of the World (Ethnologue: Dünya Dilleri) tarafından yaşayan dünya dillerinin hangi bölgelerde konuşulduğu, konuşmacıların sayısı, lehçeleri, dirençliliği, gelişmişliği, tehlikelilik durumu gibi istatistiki verileri tutulmakta ve yayımlanmaktadır. Söz konusu yayın kaynağının 21 Şubat 2022 tarihinde yayımlanan 25. baskısında dünya üzerinde 7151 adet yaşayan doğal dil olduğu yer almaktadır. (Saud, 2022)

Bir dilin canlı kalması o kültürün, tarihin ve topluluğun canlı kalması demektir. Bundandır ki değişik milletlerden öne çıkmış kişiler bu konunun önemine dikkat çekmek için bazı sözler dile getirmişlerdir. Bu bağlamda söylenmiş sözlerden bazılarına aşağıda yer verilmiştir.

"Türk demek, dil demektir. Millet olmanın en belirgin niteliklerinden biri dildir. "Türk milletindenim." diyen kişi, her şeyden önce kesinlikle Türkçe konuşmalıdır. Türkçe konuşmayan bir kişi, Türk kültürüne ve milletine bağlılığını öne sürerse buna inanmak doğru olmaz." (Atatürk)

"Muhtemelen dünyadaki birçok bitki ve hayvanın yok olma tehlikesiyle karşı karşıya olduğunu biliyorsunuzdur, ancak birçok insan dilinin de yok olma tehlikesiyle karşı karşıya olduğunu biliyor muydunuz? Küreselleşme, dünyayı daha küçük bir yer haline getirmeye yardımcı oldu. Ama aynı zamanda dünyadaki birçok dilin kaybına da katkıda bulundu. Her iki haftada bir başka bir dil sonsuza kadar yok oluyor! Bir dil kaybolduğunda, insanlık zengin mirasımızın bir kısmını da kaybeder. Nesli tükenmekte olan dilleri korumaya yardımcı olmak, gelenekleri canlı tutmak için önemlidir." (Robert Alan Silverstein)

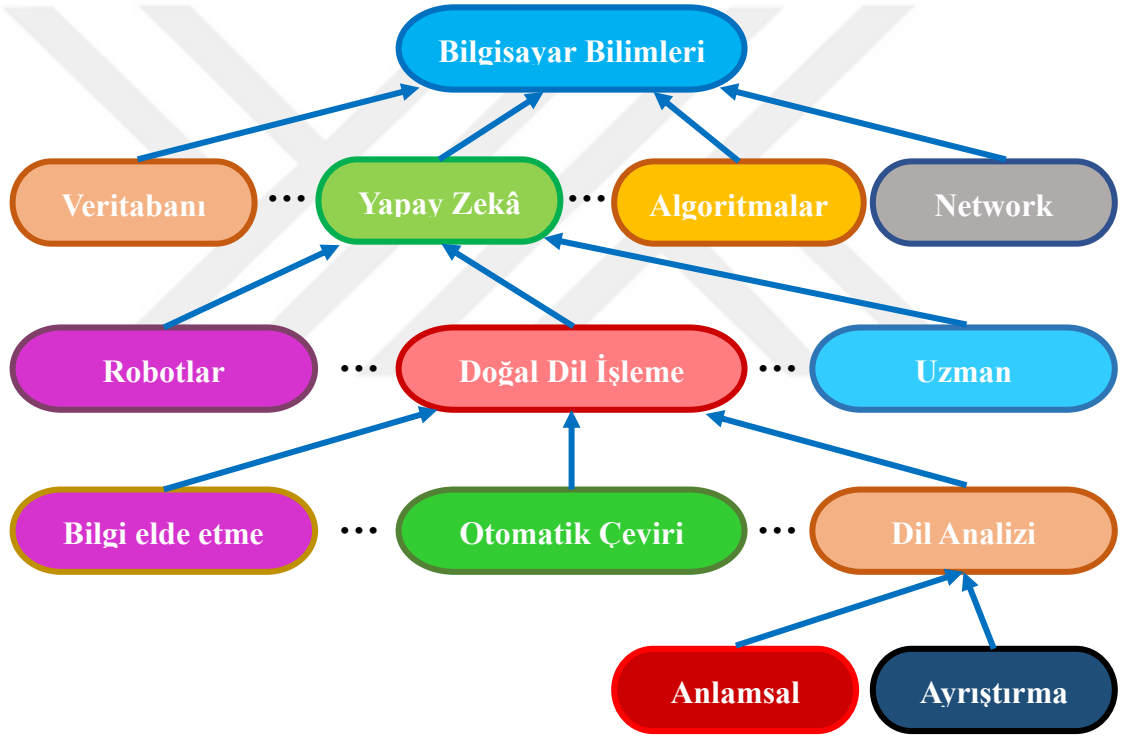
"Dilimiz kendimizin aynasıdır. Bir dil, konuşmacılarının karakterinin ve gelişiminin tam bir yansımasıdır." (Cesar Chavez)

"Dilin gelişimi, kişiliğin gelişiminin bir parçasıdır, çünkü kelimeler, düşünceleri ifade etmenin ve insanlar arasında anlayış oluşturmanın doğal araçlarıdır." (Maria Montesori)

1.2. Doğal Dil İşleme (DDİ)

İnsanlık tarihi incelendiğinde bazı gelişmelerin sağlamış olduğu sıçramalar dikkat çekmektedir. Dil bu gelişmelerin en başında gelmektedir. Dil sayesinde inanç, kültür, sanat ve pozitif bilimler gibi daha birçok alanda değişim ve ilerleme kaydedilmiştir.

Teknolojinin hızla gelişmesi ve insan-bilgisayar etkileşiminin artması sonucunda makine öğrenmesi, dil bilimi ve yapay zekâ çalışmaları artmış, buna bağlı olarak DDİ alanında birçok uygulama geliştirilmiştir. DDİ uygulamalarının geliştirilmesinde Şekil 1.1'de görüldüğü üzere çok sayıda bilim dalından ve disiplinden faydalanılmaktadır. DDİ; yapay zekâ, dil bilimi, bilişsel psikoloji ve veri bilimi alanları ile etkileşimli olarak çalışan, metin veya ses verilerini girdi olarak alıp belirli kurallar çerçevesinde istenen çıktıları sağlayarak makine dili ile konuşma dili arasındaki ilişkiyi kuran bir köprüdür. DDİ değişmeyen algoritmalar içermediğinden ve belirsiz durumlara sahip olduğundan Non-deterministic Polynomial (NP) olarak değerlendirilen bir problemdir. (Diri, 2020), (Özbilici, 2006)



Şekil 1.1. Doğal dil işlemenin bilgisayar bilimlerindeki yeri

1.3.Doğal Dil İşlemenin Tarihçesi

İngilizce literatürde Natural Language Processing (NLP) olarak karşımıza çıkan DDİ ile ilgili ilk çalışmalar 1950'li yıllarda başlamıştır. Başladığı yıllardan günümüze kadar geçen zaman aralığı gözlemlendiğinde tarihsel değişiminin 3 farklı aşamadan meydana geldiğini görmek mümkündür. Bu aşamalar Şekil 1.2'de verilmiş olup sırasıyla; Sembolik NLP, İstatiksel NLP ve Nöral NLP'dir.

Şekil 1.2. DDİ sürecinin gelişimi ve evrimi (Özmutlu, 2021)

1.3.1. Sembolik NLP (1950-1990)

İlk aşama olan Sembolik NLP, 1950 yılları ile 1990 yılları arasını kapsamakta olup bu alanda yapılan ilk girişimleri ve doğal dil işlemenin temelini oluşturan çalışmaları içermektedir. Bu dönemde kural tabanlı metotlar bilgisayarlara önceden yüklenerek dilin çözümlenmesi yapılmaya çalışılmıştır. Joseph Weizenbaum tarafından geliştirilen ve bir psikoloğu taklit eden ELIZA adlı DDİ uygulaması dönemin en başarılı ve dikkat çeken çalışmalarından biridir. (Weizenbaum, 1966) Çalışmanın amacı insan ile bilgisayar arasındaki iletişimin mümkün olduğunu gösterebilmektir. Ancak bu dönemde elle yazılmış karmaşık kurallar ile taklit yöntemi kullanılarak DDİ uygulamaları geliştirildiğinden kısıtlı düzeyde başarı sağlanmıştır.

1.3.2. İstatiksel NLP (1990-2010)

İkinci aşama olan İstatiksel NLP 1990 yılları ile 2010 yılları arasını kapsamakta olup bir önceki dönem olan Sembolik NLP'de çözülemeyen sorunların istatistiksel yaklaşımlarla çözülmesine olanak sunmuştur. Bu dönemde söz konusu yaklaşımların üzerine kurulmuş olan makine öğrenmesi algoritmaları ile DDİ alanında bir devrim yaşanmıştır. Günümüzde kullanılmakta olan modern DDİ yöntemlerinin temelleri bu dönemde atılmıştır.

1.3.3. Nöral NLP (2010-Günümüz)

Son olarak doğal dil işlemenin altın çağı olarak da adlandırabileceğimiz Nöral NLP, 2010 yılı ile günümüz arasında kalan dönemi kapsamaktadır. Bu dönemde hızla gelişen makine öğrenmesi algoritmaları, derin sinir ağları, derin öğrenme, büyük veri ve yapay zekâ DDİ alanında büyük bir sıçramaya neden olmuştur. Bunun yanı sıra sosyal

medya mecralarının ve internetin yaygınlaşmasıyla birlikte çok sayıda içerik kullanıcılar tarafından üretilerek paylaşıma sunulmuştur. Bu sayede pek çok alanda doğal dil işlemenin kullanımı yaygınlaşmış ve bu alanda her zamankinden daha fazla uygulama geliştirilmeye başlanmıştır.

1.4. Doğal Dil İşlemenin Alt Dalları ve Temel Kullanım Alanları

DDİ zaman zaman Doğal Dil Anlama (DDA / Natural Language Understanding-NLU) ve Doğal Dil Üretme (DDÜ / Natural Language Generation-NLG) kavramları ile karıştırılabilmekte ve birbirinin yerine kullanılabilmektedir. Söz konusu karışıklığın sebebi birbirine benzeyen işlem adımları olsa da Şekil 1.3'te gösterildiği gibi DDA ve DDÜ süreçlerinin bütünü Doğal Dil İşlemeyi oluşturmaktadır. (Talan ve Aktürk, 2021)

Şekil 1.3. Doğal dil işlemenin bileşenleri (Yılmaz ve Yumuşak, 2021)

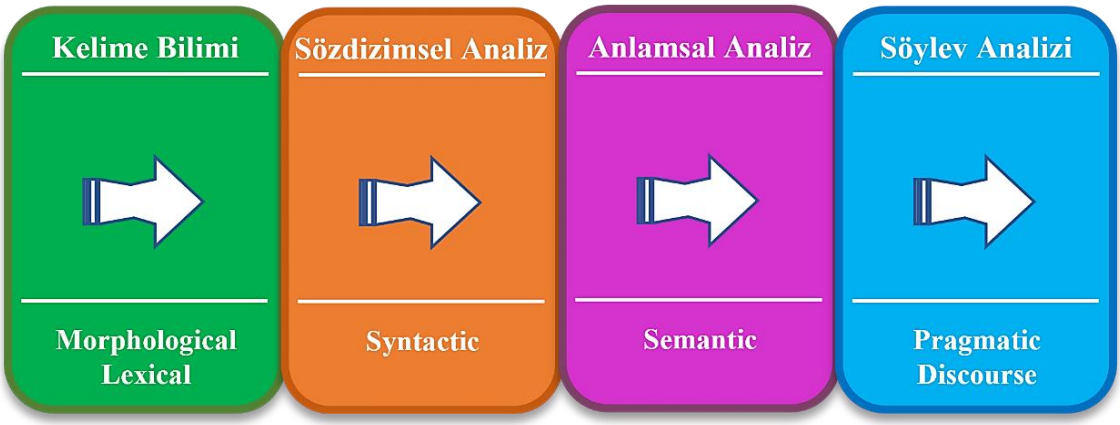
DDA, metin veya konuşma biçiminde olan bir cümlenin anlamını belirlemek amacıyla sözdizimsel ve anlamsal analiz yöntemlerine ihtiyaç duyan doğal dil işlemenin bir alt dalıdır. DDÜ ise doğal dil işlemenin başka bir alt dalıdır. DDA ile DDÜ arasındaki temel fark, birisi okuduğunu veya duyduğunu anlamaya çalışırken bir diğeri anlamlı bir cümle oluşturmayı veya seslendirmeyi hedeflemektedir. Konunun daha iyi anlaşılmasını sağlamak için; doğal dil anlamaya iPhone telefonlarda kullanılan Siri uygulamasına veya Android telefonlarda kullanılan Google Asistan uygulamasına verilen sesli bir komut karşısında uygulamanın komutu anlayarak işlem yapması, doğal dil üretmeye ise yine iPhone telefonlarda kullanılan Siri uygulamasına veya Android telefonlarda kullanılan Google Asistan uygulamasına verilen sesli bir komut karşısında uygulamanın sorulan soruya cevaben sesli olarak geri dönüş yapması örnek olarak verilebilir.

Konuşulan doğal dilin yaşayan bir sistem olmasından kaynaklı olarak dil zaman içerisinde değişikliklere uğramakta ve farklılaşmaktadır. Aynı zamanda kurallarında ve yapısında meydana gelen değişiklikler, platformlara veya içeriklere göre değişiklik gösteren kullanım alanları ve her doğal dilin kurallarının birbirinden farklı olması nedeni

ile söz konusu alanda birçok farklı uygulama ve çalışma ortaya çıkmıştır. Bu anlamda geliştirilen uygulamaların varlığı günlük hayatımızda her ne kadar fark edilmese de çok sayıda alanda sıkça kullanılmaktadır. Örneğin; etkileşimli dil öğrenme platformları (Duolingo vb.), firmaların ve kurumların kullanmış olduğu sesli yanıt sistemleri (Banka, internet servis sağlayıcıları, kargo firmaları, çağrı merkezleri vb.), işitme ve görme engeli olan bireyler için geliştirilen uygulamalar, yazım denetimi ve kelime öneri sistemi, metin özetleme uygulamaları, metinden bilgi çıkarımı DDİ uygulamalarının kullanım alanlarından sadece bazılarıdır. Yapılan çalışmalar göz önüne alındığında aslında doğal dil işlemenin hayatımızla ne kadar iç içe ve yaşantımızı kolaylaştıran bir etmen olduğu daha da iyi anlaşılmaktadır.

1.5. Doğal Dil İşleme Süreçleri

Doğal dil işlemenin çalışma adımları Şekil 1.4'te olduğu gibi dört ana başlık altında toplanmaktadır. Bunlar Kelime Bilimi (Biçimbilim / Morphological-Lexical), Sözdizimsel (Syntactic) Analiz, Anlamsal (Semantic) Analiz ve Söylev Analizi (Pragmatic-Discourse) olarak sıralanmaktadır.



Şekil 1.4. DDİ süreçleri (Seker, 2015)

1.5.1. Kelime bilimi (Morphological-lexical)

Morfoloji kelimesi, şekillerin incelenmesi anlamına gelmekle beraber dilbiliminde sözcüklerin iç yapısının incelenmesi manasını da taşımaktadır. Morfolojik analiz aşamasında kelimelerin en küçük yapı birimleri olan ek ve kökler ayrı ayrı incelenerek tür ve biçim belirlemesi yapılmaktadır. Türkçe sondan eklemeli bir dil olup

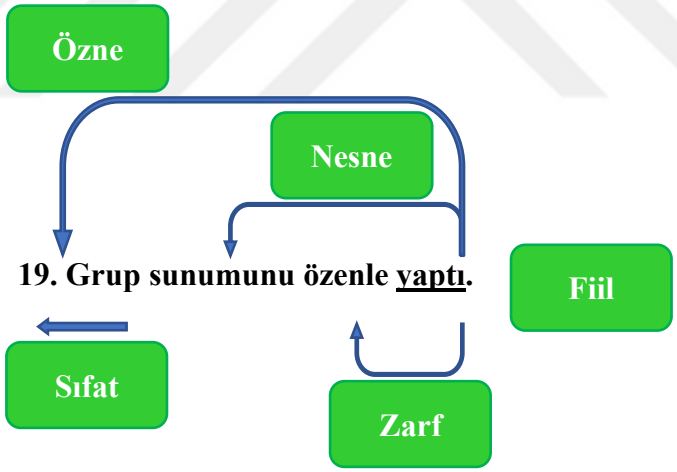
aldığı ekler ile yeni kelimeler türetilebilmektedir. Bu nedenle morfolojik analiz yapılırken ilk olarak kelimenin kökü bulunmaktadır. Bu çalışmada kelime köklerinin bulunması için Zemberek kütüphanesi kullanılmış olup konu materyal ve yöntem başlığı altında detaylandırılarak açıklanmıştır.

- Örnek: Hüseyin'in kardeşleri Hatice ve Ferhat gittiler.

Kardeş	+	ler	+	i
İsim (kök)	+	Çoğul eki	+	3'üncü tekil şahıs iyelik eki

1.5.2. Sözdizimsel (Syntactic) analiz

Sözdizimsel analiz yöntemi, tümcenin yapısal tanımını ortaya çıkarabilmek amacıyla morfolojik analiz sonucunda elde edilen verileri kullanmaktadır. Bu yöntem Şekil 1.5'te verildiği gibi cümlede yer alan ve art arda gelen anlamlı sözcük yığınının dizilim biçimi ile ilgilenir. Aynı zamanda tümcedeki kelimelerin birbiri arasındaki ilişkilerini yorumlayan analizleri içermektedir.



Şekil 1.5. Sözdizimsel analiz (Ozyurt ve Kose, 2006)

1.5.3. Anlamsal (Semantic) analiz

Metinlerde yer alan cümlelerde hangi duygu ve düşüncelerin ifade edilmek istendiğinin anlaşılması için kullanılan analiz yöntemidir. (2006) Çünkü doğal dil işlemenin sağlıklı bir şekilde çalışabilmesi, cümlelerde anlatılmak istenen ana düşüncenin doğru olarak anlaşılmasıyla doğrudan ilgilidir. Bu nedenle cümlelerde kullanılan

sözcüklerin karşılık geldiği anlamlar sistem tarafından belirlenir ve anlamlılık bakımından doğruluğu kontrol edilir.

1.5.4. Söylev (Pragmatic-discourse) analizi

Anlamsal analiz aşamasında her cümlenin ve kelimenin ne ifade etmek istediğini çözümleyen bilgisayar, bu aşamada ise cümlelerin bir araya gelerek aktarmak istediği duygu ve düşünceyi doğru olarak analiz etmeyi hedeflemektedir. Söylev analizi birçok cümle üzerinde çalışmakta, her kelimeyi veya tümceyi sırasıyla kullanıldığı sözcük, cümle, paragraf ve belge bağlamında değerlendirmektedir. (Kul, 2020) Konuya örnek olması açısından ev sahibi ve kiracı arasında geçen diyalog aşağıda verilmiştir.

- Ev sahibi : Okuyor musunuz?
- Kiracı : Evet, üniversite 3 sınıfa gidiyorum.
- Ev Sahibi : Tebrik ederim, çok güzel.
- Kiracı : Aynı zamanda gitar çalmayı da öğreniyorum.
- Ev Sahibi : Bu kötü.

Eğer diyaloğun kiracı ve ev sahibi arasında geçtiği bilinmezse “Bu kötü.” ifadesi anlaşılamaz.

Sonuç olarak; doğal dil işlemede bilgisayar öncelikli olarak kelimeyi kök ve eklerine ayırarak morfolojik analiz işlemini gerçekleştirir. İkinci adımda ise tümcelerdeki kelimelerin dizilim ve biçimine göre sözcükleri anlamlandırmaya çalışarak söz dizimsel analiz işlemini gerçekleştirir. Daha sonra anlamsal analiz yöntemi ile cümlelerin hangi duygu ve düşünceyi ifade etmeye çalıştığını çözümleyerek cümle bütünlüğünü sağlar. Son olarak pragmatik analiz ile tümcelerin bir arada ve kullanıldıkları bağlamda neyi ifade etmek istediklerini anlamlandırmaya çalışır.

1.6. Doğal Dil İşleme Çalışmalarında Karşılaşılan Zorluklar

DDİ süreçlerinde bazen kullanılan dilin belirsizliklerinden ve dilbilgisi kurallarından, bazen de yazarın yapmış olduğu hatalardan kaynaklanan birtakım zorluklarla karşılaşılmaktadır. Bu bölümde DDİ çalışmalarında sıkça karşılaşılan sorunlar ve zorluklar yer almaktadır.

1.6.1. Kuralsız ve anlaşılmaz konuşmalar

Bazı bölgelerde yöre halkı tarafından kullanılan, yazı dilinden biçim ve ses olarak farklılıklar gösteren söyleyiş biçimlerinden kaynaklanan zorluklardır. Gidiyom, napıyon len, böğün, gelcez gibi kelimeler bu duruma örnek olarak verilebilir.

1.6.2. Kuralsız ve bozuk yazılar

Bağlı bulunduğu dilin dilbilgisi kurallarına ve noktalama işaretlerine uymayan, çoğunlukla yerine kullanıldığı kelimeyi çağrıştıran sözcük veya cümlelerdir. Tmm, Tamm, kelebk gibi sözcükler bu duruma örnek olarak verilebilir.

1.6.3. Konuşmayı bölme

Doğaçlama olarak yapılan konuşmanın metne dönüştürülmesi sürecinde sıklıkla karşı karşıya kalınan bir sorundur. Hazırlıksız konuşmalarda konuşmacılar bazen noktalama işaretlerine, nefes almaya, cümleler arasında duraklamaya ve vurgulamaya dikkat etmeden konuşmaktadırlar. Bu durum, konuşmanın metne çevrilmesi aşamasında bilinmesi gereken cümle başlangıç ve bitiş noktalarının belirlenememesine sebebiyet vermektedir. Cümlelerin başlangıç ve bitiş noktalarının belirsizliği konuşmanın bölünmesini zorlaştırmakta ve DDİ çalışmalarında yanlış sonuçlar elde edilmesine yol açmaktadır. Yukarıda bahsedilen konuşma hususlarına dikkat edilmesi durumunda söz konusu sorunlar ortadan kalkacak ve konuşmanın sağlıklı bir biçimde bölünmesi sağlanacaktır.

1.6.4. Metni bölme

Noktalama işaretlerinin sistematik bir biçimde kullanıldığı dillerde yazılı metin içerisindeki her bir cümle kesin ve net olarak ayırt edilebilmektedir. Buna karşın Çince, Japonca gibi bazı dillerde kelimelerin ve cümlelerin birbirinden ayırt edilmesi oldukça zordur. Arapça gibi bazı dillerde ise cümlelerin ortalama beş on satır kapsadığını ve Türkçeye göre adeta paragraf uzunluğunda olduğunu gözlemlemek mümkündür. (Adalı, 2012) Benzer niteliklere sahip dillerin yazılı metinleri üzerinde çalışma yapılabilmesi için

her cümlenin kendi içerisinde anlamlı bir ifade oluşturabilecek şekilde metnin parçalara bölünmesi gerekmektedir.

1.6.5. Dilde belirsizlik – muğlaklık durumu (Ambiguity)

Doğal dillerde belirsizliğin oluşmasına neden olan faktörleri üç temel başlık altında ifade etmek mümkündür. Bunlar sözcüksel belirsizlik, sözdizimsel belirsizlik ve anlamsal belirsizlik olarak sıralanmaktadır.

1.6.5.1. Sözcüksel belirsizlik (Lexical)

Türkçe gibi bitişik ve sondan eklemeli dil ailesine (Agglutinative) üye dillerde köke uygulanan ek veya ekler, sözcüğün anlamını ve hatta niteliğini bile değiştirebilmektedir. Bu nedenle bitişik ve sondan eklemeli dillerde sözcüğün kökünü ve kelimelerin anlamlarını bulmak bir hayli zordur.

1.6.5.2. Sözdizimsel belirsizlik (Syntactic)

Cümle içerisinde yer alan kelimelerin birbiri arasında yer değiştirmesiyle cümlenin dilbilgisi yapısı ve kelime sayısı değişmez, ancak cümlenin anlamı ve vurgusu değişebilmektedir. Bu durum insanlar tarafından rahatlıkla anlaşılabilse de aynı şey bilgisayarlar için söz konusu değildir. Doğal dil işlemenin en doğru sonuçları verebilmesi için durumun bilgisayar tarafından algılanarak bu çerçevede işlem yapılması gerekmektedir.

1.6.5.3. Anlam belirsizliği

Yazılışları aynı fakat anlamları farklı kelimeler sesteş kelime olarak adlandırılmaktadır. Sesteş kelimeler genellikle anlam belirsizliklerine yol açmaktadır. Çünkü sesteş kelime ile ne ifade edilmek istendiği o kelimeye bakılarak anlaşılamaz, bunun için bulunduğu cümleye hatta paragrafın tamamına bakılmasına ihtiyaç duyulabilir. Anlam belirsizlikleri, özellikle bir dilden başka bir dile çeviri yapılırken önemli bir sorun olarak karşımıza çıkmaktadır. Buna örnek olarak, ülkemizde çok satan ve saygın bir gazetede yapılan haber başlığında “Köprücüler İstanbul'da toplanıyor”

ifadesi kullanılmıştır. Haberin başlığını okuyan gazete okurlarının akıllarında muhtemelen ilk canlanacak şey köprü inşaatı ile ilgilenen bazı şirket yetkilerinin toplantı yapmak amacı ile İstanbul'da buluşacağı olacaktır. Ancak haberin içeriği okunmaya başlandığında konunun köprü inşaatı ile ilgilenen firma yetkilerinin bir araya gelmesiyle ilgili olmadığı, aslında bir oyundan bahsedildiği anlaşılacaktır. Yabancı kaynaktan alınmış bu İngilizce haberi dilimize çeviren yazar, briç oyunu ve köprü anlamlarını taşıyan "bridge" sözcüğünün sesteş olmasından kaynaklı bu anlam belirsizliği hatasına düşmüştür.

1.7. Metin Sınıflandırma

Metin sınıflandırma, denetimli öğrenim yöntemleri kullanılarak metinlerin önceden belirlenmiş kategorilere atanması olarak tanımlanmaktadır. E-posta adreslerine kötü niyetli kişi ya da guruplar tarafından gönderilen zararlı mesajların spam filtreleri aracılığı ile ayıklanarak başka bir klasörde tutulması metin sınıflandırmaya örnek olarak gösterilebilir.

Dokümanların metin sınıflandırma yöntemi ile belirlenen kategorilere ayırılabilmesi için Şekil 1.6'da gösterilen işlem adımlarından geçmesi gerekmektedir.

Şekil 1.6. Metin sınıflandırma için gerekli olan temel adımlar

Metin sınıflandırma yapabilmek için gerekli olan ilk işlem adımı çeşitli özniteliklere sahip verilerin toplanarak veri setinin oluşturulasıdır.

Toplanan verilerin sınıflandırmaya uygun hale getirilmesi için; aykırı verilerin veri setinden çıkartılması, metnin küçük harfe çevrilmesi, kelimelerin kök ve eklerine ayrılması, metinde yer alan satır boşluklarının, durak kelimelerin ve noktalama

işaretlerinin temizlenmesi gibi veri ön işleme tekniklerinin kullanılması gerekmektedir. (Uysal ve Gunal, 2014)

Ön işleme ile gereksiz verilerinden temizlenerek kelime köklerine ayrılan metinden daha fazla öznitelik türetilerek sınıflandırma algoritmalarının başarısını arttırabilmek için; DDİ araçlarından, makine öğrenmesi kütüphanelerinin bazı fonksiyonlarından, Kelime Çantası (Bag of Words - BOW) ve Terim Sıklığı - Ters Doküman Sıklığı (Term Frequency - Inverse Document Frequency - TF-IDF) gibi öznitelik ağırlıklandırma yöntemlerinden faydalanılmaktadır.

Öznitelik ağırlıklandırma, veri kümesini oluşturan her bir kelimenin bulunmuş olduğu metni ne kadar ifade ettiğinin hesaplanarak sözcüğün önem derecesinin belirlenmesi olarak tanımlanmaktadır. Öznitelik ağırlıklandırma, metin sınıflandırma çalışmalarında doküman içerisindeki ayırt edici kelimelerin belirlenerek başarının arttırılmasında önemli rol oynayan bir öznitelik çıkarım işlem adımıdır. Bu işlemler sonucunda öznitelikler sayısallaştırılarak ağırlıklandırılmış ve makine öğrenmesi sınıflandırma algoritmalarının kullanımı için uygun hale getirilmiştir. Ayrıca bu aşama sonucunda elde edilen bazı sayısal değerler normalize edilerek veri optimizasyonu sağlanmıştır.

Metin sınıflandırma çalışmalarında kullanılan dokümanlar genellikle yüksek boyutludur. Öyle ki sıradan bir metin veri kümesi bile on binlerce nitelikten oluşabilmektedir. Sınıflandırma algoritmalarının bu veya buna benzer veri kümeleri üzerinde çalışarak yüksek başarı oranı elde etme olasılığı oldukça düşüktür. Bu gibi durumlarda öznitelik seçim yöntemleri ile sınıflandırmaya etkisi olmayan niteliklerin tespit edilerek veri kümesinden çıkartılması ve veri kümesi boyutunun optimum seviyeye düşürülmesi önemlidir. (Uysal ve Gunal, 2012), (Agnihotri ve ark., 2017)

Son olarak, sınıflandırma algoritmalarının yardımı ile metin önceden belirlenmiş kategorilere ayrılmakta ve böylece sınıflandırma için gereken işlem adımları tamamlanmaktadır.

1.8. Çalışmanın Amacı

Teknolojinin hızla gelişmesi ve iletişim ağlarının globalleşmesi ile bitlikte internet kullanımı da yaygınlaşmıştır. İnternet ağlarının ve kullanımının yaygınlaşması, beraberinde birçok hizmetin bu alanda da verilmesine, veriye ulaşımın kolaylaşmasına ve veri sayısının her geçen gün katlanarak artmasına sebep olmuştur. Bu durum aslında

birçok alanda hayatımızı kolaylaştırmış olsa da veri sayısının bu denli artmasına bağlı olarak yapay zekâ ve makine öğrenmesi alanlarında yeni çalışmaların yapılması gerekliliğini ortaya çıkarmıştır. Çünkü aranan verileri bulabilmek, içerisinden doğru seçimler yapabilmek ve nitelikli çalışmalar üretebilmek oldukça zorlaşmıştır. Bu nedenle özellikle anlamsız verilerden anlamlı verilerin çıkarılması ve söz konusu verilerin belirli niteliklere göre sınıflandırılması konuları önem kazanmıştır.

Günümüzde birçok haber kanalı ve gazete okuyucularının veya izleyicilerinin televizyonda verilen veya gazetede basılan güncel bilgilere bulundukları yerden anlık olarak ulaşabilmelerini sağlamak için dijital platformda da hizmet vermektedir. Bu çalışmada gazetelerin internet sitesinden alınan haberler incelenerek yazar-eser eşleştirmesi yapılmıştır.

Stilometri (Sayısal Üslup Analizi - SÜA), eserlere ait çeşitli istatistiksel metriklerin üslup belirlemesi yapabilmek amacıyla kullanılması anlamına gelmekte olup genellikle edebiyat alanındaki yazılı metinler üzerinde uygulanmakta, aynı zamanda resim ve müzik alanlarında da başarılı bir biçimde kullanılmaktadır. (İşi ve ark., 2013)

Bu çalışmada farklı alanlarda gazete yazarlığı yapan kişilerin yazmış olduğu yazılar SÜA, yapay zekânın bir alt dalı olan DDİ ile makine öğrenmesi yöntemleri kullanılarak analiz edilmiş ve söz konusu yazıların hangi yazarlar tarafından yazıldığının doğru eşleştirilebilmesi amaçlanmıştır. Veri seti üzerindeki veri ön işleme çalışmaları tamamlandıktan sonra Zemberek kütüphanesi fonksiyonları kullanılarak DDİ süreçleri ve öznitelik çıkarma işlemleri gerçekleştirilmiştir. Sonrasında çalışmada kullanılan makine öğrenmesi sınıflandırma algoritmalarından hangisinin yazar-eser eşleştirmelerinde daha başarılı olduğuna karar verebilmek için karşılaştırma yapılmış ve model seçiminin buna bağlı olarak belirlenmesi hedeflenmiştir. Çalışmanın son bölümünde, yapılan deneysel analizler üzerinde durulmuş ve elde edilen analiz sonuçları göz önünde bulundurularak çalışmanın başarısını arttırmaya yönelik işlem adımları yürütülmüştür.

1.9. Çalışmanın Önemi

İnternetin yaygınlaşmasıyla birlikte birçok alanda olduğu gibi gazetecilik alanında da değişimler ve gelişmeler meydana gelmiştir. Çevrim içi gazeteler, birçok unsuru içerisinde barındırabilmesi ve daha büyük kitlelere hitap etmesi nedeniyle çok sayıda gazete sahibinin ve bu alanda hizmet veren yazarın yoğun ilgisini üzerine çekmiştir. Bu kapsamda; 1990 yılından itibaren geleneksel yöntemlerle yayın yapmakta olan gazeteler

dijital ortamda hizmet verebilmek için hızlı bir değişim içerisine girmiş ve bununla birlikte internet üzerinden hizmet veren medya kuruluşlarının sayısında önemli ölçüde artış meydana gelmiştir.

Gazetecilik alanında meydana gelen bu gelişmeler, farklı alanlarda eser veren yazarların aynı platform üzerinde okuyucular ile buluşmasına olanak sağlayarak birçok alanda sayısız eser verilmesinde etkili olmuştur. Böylelikle okuyucular merak etmiş oldukları alanlarda farklı yazarlar tarafından kaleme alınmış eserlere, gündeme dair merak edilen konulara ve takip etmiş oldukları yazarların olaylar karşısındaki bakış açılarına kolayca ve kısa yoldan erişebilmiş, bu konularda bilgi sahibi olmuştur. Ancak söz konusu olumlu gelişmelerin yanı sıra veri sayısında meydana gelen bu ciddi artış istenilen ve doğru veriye erişimi bir hayli zorlaştırmıştır. Veri çokluğuna bağlı olarak istenilen veriye erişimde yaşanan zorluk ile birlikte söz konusu verilerin işlenerek anlamlı hale getirilmesi ve verilerin sınıflandırılmasına duyulan ihtiyaç, makine öğrenmesi alanında birçok uygulama geliştirilmesine sebep olmuştur.

Yazılan eserlerin yazarlarına göre sınıflandırılması işlemi metin sınıflandırması alanına girmekte ve bunun için DDİ ile makine öğrenmesi yöntemlerinden faydalanılmaktadır. Metin sınıflandırmasının ilk aşaması metnin yazılmış olduğu dilin bilgisayar tarafından algılanması ve parçalara bölünerek makine tarafından anlamlandırılması adımlarından oluşmaktadır. Bahse konu işlemlerin yabancı diller üzerinde gerçekleştirilebilmesi amacıyla birçok yöntem geliştirilmiş ve bu alanda pek çok uygulama literatüre kazandırılmıştır. Ancak söz konusu metin sınıflandırma işlemi Türkçe gibi sondan eklemeli dillerde kelimenin sonuna gelen yapım eklerinin kelimenin anlamını değiştirmesinden ve Türkçe için geliştirilmiş DDİ yöntemlerinin yetersizliğinden dolayı zorlaşmaktadır.

Bu çalışma:

DDİ kütüphanesi olan Zemberek'in çeşitli fonksiyonlarının Türkçe metinler üzerinde kullanılması,

TF-IDF, BOW ve kelime-cümle dağılım vektör modellerinin kullanılması,

Öznitelik seçim yöntemleriyle sınıflandırma için uygun olmayan niteliklerin belirlenerek veri setinden çıkartılması sonucunda; veri boyutunun azaltılması, sınıflandırma başarısının arttırılması ve sınıflandırma süresinin düşürülmesi,

Makine öğrenmesi sınıflandırma algoritmalarının yazar-eser eşleştirmeleri sonucunda elde edeceği başarı oranlarının karşılaştırılması, parametre analizlerinin

yapılarak yöntemin maksimum seviyede başarı elde etmesinin sağlanması ve uygun sınıflandırma yönteminin tercih edilmesi açısından örnek teşkil edecektir.

Dinamik ve anlık olarak değişebilen internet ortamında bulunan eserlerde yer alan kelime kökleri, sözcüklerin kullanım sıklığı ve noktalama işaretlerinin kullanımı yazım tarzının tespit edilmesine ve eser ile yazar arasındaki bağlantının kurularak yazarın tahmin edilmesine olanak sağlayacaktır. Aynı zamanda bu çalışma; eserler arası benzerliklerin bulunması, yazarları belli olmayan eserlerin yazarlarının tahmin edilmesi, yazarların yazım tekniklerinin ve alışkanlıklarının tespiti, eserlerinde benzer yazım tarzı kullanan yazarların tespiti, yazara ait eserlerin sınıflandırılarak bir araya toplanması, istenen yazara ve veriye ulaşmanın kolaylaşması, metinde kullanılan noktalama işareti sayısı, en çok kullanılan kelime, kelime zenginliği ölçüsü gibi yazara ait birçok istatistiki veriye ulaşmak için kullanılabilecektir.

1.10. Çalışmanın Organizasyonu

Bu tez çalışması beş ana bölümden oluşmaktadır. Bölümlere ait içerikler aşağıda listelenmiştir.

Birinci bölümde dilin tanımı yapılmış ve dil çeşitlerinden (Makine dili, yapay dil ve doğal dil) bahsedilmiştir. Doğal dil işlemenin tanımı, açıklanması, bilgisayar bilimindeki yeri, etkileşimli çalıştığı bilim dalları, kullanım alanları, günlük hayatımızdaki yeri, hangi süreçlerden meydana geldiği, hangi yıllarda ortaya çıktığı, tarihçesi, gelişim aşamaları, alt dalları ve zorlukları detaylandırılarak bu hususlar hakkında bilgi verilmiştir. Ayrıca çalışmanın amacı, önemi ve organizasyonu bu bölümde ele alınmıştır.

İkinci bölümde benzer çalışmalar incelenerek elde edilen literatür taramalarına yer verilmiştir. DDİ ile ilgili yapılan temel uygulamalar, doğal dil işlemenin kullanıldığı sınıflandırma uygulamaları, Zemberek kütüphanesi ile yapılan çalışmalar ve son olarak Türkçe dili üzerine yapılmış olan DDİ çalışmaları hakkında bilgi verilmiştir.

Üçüncü bölümde çalışmada kullanılan materyal ve yöntemlere yer verilmiştir. Çalışmada kullanılan veri setinin nitelikleri, veri setine uygulanan veri ön işleme teknikleri ve Zemberek kütüphanesi hakkında bilgi verilmiştir. Veri temsil ve ağırlıklandırma yöntemleri ile öznitelik çıkarımı, öznitelik seçimi, hiper parametre ayarlaması, çapraz doğrulama yöntemleri ve kullanılan makine öğrenmesi sınıflandırma algoritmalarının çalışma prensipleri açıklanmıştır.

Dördüncü bölümde çalışmanın gerçekleştirilmesi için uygulanan işlem adımları detaylandırılmıştır. Veri setinin toplanması için yapılan çalışmalara, veri ön işleme adımlarının uygulanmasına, özniteliklerin çıkarılmasına ve makine öğrenmesi algoritmalarının veri seti üzerinde elde etmiş oldukları sınıflandırma başarısına bu bölümde yer verilmiştir.

Beşinci ve son bölümde; yapılan çalışmanın kısa özetine, uygulanan yöntemler sonucunda elde edilen verilerin karşılaştırılmasına ve analiz edilmesine yer verilmiştir. Yapılan analizler sonucunda ulaşılan gözlemler dikkate alınarak bazı çıkarımlarda ve değerlendirmelerde bulunulmuştur. Ayrıca çalışmanın daha ileri noktalara taşınabilmesi amacıyla sunulan bazı öneriler de bu bölümde yer almaktadır.

2. KAYNAK ARAŞTIRMASI

Bu çalışmada kullanılacak yöntemler ile işlem adımlarının belirlenmesi ve ön bilgi sahibi olunabilmesi amacıyla benzer konularda daha önceden hazırlanmış kitaplardan, tezlerden ve makalelerden faydalanılmıştır. Bu kapsamda yararlanılan kaynaklar hakkında aşağıda kısaca bilgi verilmiştir.

Nur Banu'nun 2011 yılında yaptığı çalışmada Türkçenin kullanımı ve psikolojik durum arasındaki ilişki araştırılmıştır. (Albayrak, 2011) Çalışmada psikolojik sorunları olan kişilerden ve herhangi bir psikolojik problemi bulunmayan kişilerden Türkçe metinler toplanmıştır. Elde edilen veri setine veri ön işleme adımları uygulanmıştır. Toplanan metinler üzerinde Zemberek kütüphanesi kullanılarak morfolojik analiz ile kelimeler kök ve eklerine ayrılmış, öznitelik çıkarma işlemiyle veri setine yeni nitelikler eklenmiştir. Makine öğrenmesi ve veri ön işleme için Java programlama dili ile geliştirilmiş WEKA isimli program kullanılmıştır.

1980'li yıllara kadar DDİ kural tabanlı olarak yapılmaktaydı. Bu dönemde yapılan DDİ çalışmaları sembolik olmakla birlikte genellikle taklide dayanmaktaydı. (Guida ve Mauri, 1986) Makine öğrenmesi, yapay zekâ ve derin öğrenme uygulamalarının yaygınlaşmasıyla birlikte kural tabanlı DDİ yöntemleri ile çözülemeyen problemler üzerinde yeniden çalışılmaya başlanmış ve üstün başarıya sahip sonuçlar elde edilmiştir.

İnternetin hızla yaygınlaşmasıyla birlikte veriye ulaşım da kolaylaşmıştır. Ancak veri sayısının fazlalığından ve internet ortamında paylaşılmasından kaynaklı olarak eser intihali gibi sorunlar sıklıkla yaşanmaya başlanmıştır. Bir edebi eser üzerinde birden çok kişinin hak iddia etmesi durumunda yazar tanıma sistemi kullanılarak eserin kime ait olduğunun tespit edilmesi büyük önem arz etmektedir. Bu ve benzer uygulamalarda söz konusu işlemin gerçekleştirilebilmesi için DDİ teknikleri kullanılmaktadır. Eserde yer alan kelime kökleri, sözcüklerin kullanım sıklığı ve noktalama işaretlerinin kullanımı yazım tarzının tespit edilmesine ve eser ile yazar arasındaki bağlantının kurularak yazarın tahmin edilmesine yardımcı olmaktadır. 2021 yılında Serkan KORKMAZ tarafından makine öğrenmesi yöntemleri kullanılarak DDİ tabanlı şair tanıma uygulaması geliştirilmiştir. (Korkmaz, 2021)

Çevrim içi platformların, bilgisayarların ve mobil cihazların sayısının artması ile birlikte internet ortamında bulunan verilerin sayısında da ciddi oranda artış meydana gelmiştir. Benzer şekilde organizasyonlara yapılan şikâyet ve önerilerde artık çevrim içi platformlar vasıtası ile yapılmaktadır. Ancak şikayetlerin çokluğu nedeni ile inceleme ve

değerlendirme aşamasında güçlükler yaşanmaktadır. Söz konusu sorunun çözümüne ilişkin olarak 2019 yılında Ahmed Enis ERKAYA tarafından çeşitli makine öğrenmesi yöntemleri kullanılarak şikayetler sınıflandırılmıştır. (Erkaya, 2019) Çalışmada veri kümesi olarak bir organizasyona yapılan şikayetler kullanılmıştır. Veri setinde bulunan yazım yanlışlarının giderilebilmesi için veri ön işleme adımları uygulanmış ve biçimbilimsel analiz yöntemleri kullanılarak verinin sadeleştirilmesi sağlanmıştır. Yapılan çalışmanın özü metin sınıflandırma işlemi olduğundan dolayı veri kümesinde çokça tekrar eden ve sınıflandırma aşamasında katkısı olmayacağı değerlendirilen kelime grupları tespit edilerek veri setinden çıkarılmıştır. Veri seti üzerinde veri ön işleme adımları öncesinde ve sonrasında Rastgele Orman, Long Short-Term Memory (LSTM), Naive Bayes, Lojistik Regresyon ve Destek Vektör Makineleri (Support Vector Machine - SVM) sınıflandırma algoritmaları kullanılmış ve elde edilen sonuçların karşılaştırması yapılmıştır. Kullanılan makine öğrenmesi algoritmaları arasında en doğru sonucu LSTM sınıflandırma algoritması vermiştir.

Elektronik ortamda (Sosyal medya platformları, oyunlar veya cep telefonu uygulamaları) kişinin şahsına yönelik yapılan her türlü taciz veya taciz girişimi siber zorbalık olarak adlandırılmaktadır. Siber zorbalığa maruz kalan bireylerin bir kısmı bu durumu önemsemezken bazı bireyler psikolojik anlamda problemler yaşamaktadır. Bu nedenle siber zorbalık amacıyla elektronik ortamdan gönderilen mesajların tespit edilerek filtrelenmesi ve mağdura zarar vermeden engellenmesi büyük önem taşımaktadır. Bu kapsamda; Türkçe dili için geçmiş yıllarda yapılmış olan çalışmalar mevcuttur. Ancak, yapılan çalışmalar incelendiğinde kullanılan veri setlerinin siber zorbalığa örnek metin sayısı açısından yeterli boyutta olmadığı gözlemlemiştir. Bu nedenle 2019 yılında Erhan ÖZTÜRK tarafından yeterli veriye sahip örnek veri setinin oluşturulması ve kullanılan sınıflandırma yöntemlerinin bahse konu veri kümesi üzerindeki başarı oranlarının incelenmesi amacıyla çalışma yürütülmüştür. (Öztürk, 2019) Siber zorbalığa örnek teşkil edebilecek hakaretler, cinsel içerikli veya küfürlü ifadeler Facebook, Twitter ve Youtube gibi mecralardan elle tek tek toplanırken, Instagram verileri yazılan bir kod parçacığı sayesinde otomatik olarak toplanmıştır. Sonuç olarak; 8 bine yakın siber zorbalığa örnek metin ve benzer sayıda siber zorbalık içermeyen metin toplanmış ve sınıflandırma yapılabilmesi amacıyla negatif veya pozitif olarak etiketlenmiştir. Veri ön işleme ve makine örenmesi işlemleri için Java tabanlı WEKA uygulaması kullanılmış olup bütün işlemler söz konusu uygulama üzerinden gerçekleştirilmiştir. (Topuz, 2021), (Alı, 2022) Veri setinde yer alan özel karakterler temizlenmiş, metin TF-IDF yöntemi ile

ağırlıklandırılmış ve kelimeler kök bulma yöntemi ile köklerine ayrılmıştır. Nitelik kümesinin küçültülebilmesi için nitelik seçim yöntemlerinden yaygın olarak kullanılan Ki-kare (Chi-Square - CHI2) ve Bilgi Kazancı yöntemleri kullanılmış ve Ki-kare yönteminin daha başarılı olduğu sonucuna varılmıştır. Elde edilen sonuca göre, Ki-kare ya da Bilgi Kazancı değeri 0.001'in altında olan nitelikler veri setinden çıkartılmıştır. Ayrıca çalışmada K-katmanlı Çapraz Doğrulama yöntemi kullanılmış, K değeri 10 olarak belirlenmiş ve veri seti eğitim ve test olmak üzere iki parçaya bölünmüştür. Çalışmada; Karar Ağaçları – J48, SVM, Multinomial Naive Bayes ve Rastgele Orman sınıflandırma algoritmaları kullanılmış, sınıflandırma başarısı ve modelleme süresi bakımından Multinomial Naive Bayes sınıflandırma algoritması en iyi sonucu verirken Rastgele Orman algoritması en kötü sonucu vermiştir.

Dünyanın en çok kullanılan ikinci dili olan İspanyolca, bugün 500 milyondan fazla insan tarafından konuşulmaktadır. İspanyolca her ne kadar dünyanın en fazla konuşulan ikinci dili olsa da metin sınıflandırma alanında İngilizce ile kıyaslandığında yeteri kadar çalışma bulunmadığı görülecektir. Bu nedenle 2018 yılında Semuel FRANKO tarafından 10 farklı konuda gazete ve dergilerden toplanan metinlerle bir veri seti oluşturulmuş, veri kümesine veri ön işleme adımları uygulanarak SVM, Naive Bayes, Maksimum Entropi ve Karar Ağaçları gibi sınıflandırma yöntemleri ile başarı oranları hesaplanmıştır. %89 başarı oranı ile en iyi sonucu SVM sınıflandırma algoritması verirken, onu %88 başarı oranı ile Maksimum Entropi ve %87 doğruluk oranı ile Naive Bayes takip etmiştir. Çalışmanın sonucunda sınıflandırma algoritmalarının tahmin süresi ve doğruluk oranları karşılaştırılarak gerekli incelemeler yapılmıştır. (Franko, 2018)

Elektronik belge miktarında meydana gelen artış metin sınıflandırma alanında yapılan çalışmaların önemini arttırmıştır. Metin sınıflandırma alanında karşılaşılan en büyük sorunlardan birisi de veri kümesi öznitelik uzayının çok geniş olmasından kaynaklı yapılan hatalı sınıflandırmalardır. 2013 yılı ve sonrasında Alper Kürşat UYSAL ve arkadaşları tarafından, söz konusu problemin ve problemin sınıflandırmanın başarısına olan olumsuz etkilerinin giderilmesine yönelik çalışmalar yürütülmüştür. (Uysal, 2015), (Uysal, 2016), (Tiryaki ve Uysal, 2023)

Dijital ortamda hizmet veren gazeteler tarafından yayımlanan haberlerin otomatik olarak kategorilere ayrılması aranan konu başlığına ait haberlerin bulunmasında okuyuculara büyük kolaylık sağlamaktadır. Ancak birçok haberin kategori bilgisi bulunmamakta, kapsamlı olarak belirtilmemekte veya yanlış atanmaktadır. Otomatik metin sınıflandırma; sık kullanılan kelimelerin veri setinden temizlenmesi, metnin

köklerine ayrılması ve terim ağırlıklandırma gibi birçok yöntemi içermektedir. Bu kapsamda Çağrı TORAMAN tarafından 2011 yılında gerçekleştirilen çalışmada; Bilkent Haber Portalından farklı konularda haber metinleri toplanarak veri kümesi hazırlanmış ve sınıflandırma yöntemi olarak K-En Yakın Komşu (K-Nearest Neighbors-KNN), C4.5, Naive Bayes ve SVM kullanılmıştır. Algoritmaların veri seti üzerindeki başarı oranları hesaplanmış ve elde edilen sonuçlar karşılaştırılarak incelemeler yapılmıştır. (Toraman, 2011)

Metin sınıflandırmada öznitelik boyutunun yüksek olması sınıflandırma başarısını etkileyen önemli bir problem olarak karşımıza çıkmaktadır. 2011 yılında Ece ÖZBİLEN tarafından yapılan çalışmada, öznitelik seçme yöntemleri birleştirilerek metin boyutunun düşürülmesi ve sınıflandırma performansının arttırılması hedeflenmiştir. Beş farklı öznitelik seçme yöntemi farklı özelliklerdeki beş veri seti üzerinde ilk olarak tek tek daha sonra çeşitli ikili birleştirmeler ile SVM sınıflandırıcısı kullanılarak test edilmiştir. Çalışmanın sonucunda elde edilen veriler incelendiğinde öznitelik seçme metotlarının çeşitli şekillerde birleştirilmesi sonucunda elde edilen başarı oranının, metotların tek tek kullanılması sonucunda elde edilen başarı oranından fazla olduğunu görülmüştür. (Özbilen, 2011)

Dijital kütüphane, e-posta mesajları ve elektronik kitap gibi dijital dokümanların sayısının hızla artması beraberinde metin sınıflandırma alanında yapılan çalışmaların da artmasına neden olmuştur. Sınıflandırma işleminde kullanılan verinin çok boyutlu olması nitelik seçme işleminin önemini arttırmakta ve algoritmanın sınıflandırma başarısını doğrudan etkilemektedir. 2008 yılında Şerafettin TAŞCI ve arkadaşları tarafından boyutları, çarpıklıkları ve karmaşıklıkları farklılık gösteren yedi adet veri seti kullanılarak öznitelik seçme işleminin sınıflandırmaya olan etkisi incelenmiştir. Çalışmada ağırlıklandırma yöntemi olarak TF-IDF kullanılırken, sınıflandırma algoritması olarak SVM kullanılmıştır. (Tasci ve Gungor, 2008)

Bekir PARLAK tarafından 2021 yılında yapılan doktora çalışmasında, metinlerin otomatik sınıflandırılmasının öneminden ve bu işlemin doğruluk oranını etkileyen parametrelerden bahsedilmiştir. Dokümanları içeriklerine uygun sınıflara atayabilmek için sınıflandırma algoritmaları kullanılmaktadır. Ancak, söz konusu sınıflandırma algoritmalarının gerçeğe yakın sonuçlar vermesini sağlamak amacıyla veri setine sınıflandırma işleminden önce bir dizi işlem uygulanması gerekmektedir. Bu kapsamda çalışmada; nitelik seçimi, nitelik çıkartma ve nitelik ağırlıklandırma yöntemlerine yer verilmiştir. Bahsedilen yöntemlerden öznitelik seçimi son zamanlarda popülerliğini

giderek arttırmış ve daha da önem kazanmıştır. Yapılan çalışmada bilindik ve güncel olarak kullanılan dokuz farklı öznitelik seçme yöntemi kullanılmıştır. Bunlar; Ki-kare, Sınıf Ayırıcı Ölçütü (Class Discriminating Measure - CDM), Ayrımcı Güç Ölçütü (Discriminative Power Measure - DPM), Olasılık Oranı (Odds Ratio - OR), Ayırt Edici Öznitelik Seçici (Distinguishing Feature Selector - DFS), Kapsamlı Öznitelik Seçim Ölçütü (Comprehensively Measure Feature Selection - CMFS), Ayrımcı Öznitelik Seçimi (Discriminative Feature Selection - DFSS), Normalleştirilmiş Fark Ölçütü (Normalized Difference Measure - NDM), Max-Min Oranı (Max-Min Ratio-MMR) olarak sıralanabilir. Ancak literatürde bulunan bu dokuz geleneksel yöntemin dışında yeni olarak önerilen Ayrıntılı Öznitelik Seçim yöntemi, söz konusu diğer dokuz yöntemden daha yüksek performans sağlayarak çalışma sonucunda önerilen öznitelik seçim yöntemi olmuştur. Bunun yanı sıra metnin cümlelerine ve kelimelerine ayrılması, kelime köklerinin çıkarılması, kullanılmayan kelimelerin ayıklanması ve metnin karakterlerinin küçük harfe çevrilmesi gibi veri ön işleme adımları uygulanmıştır. Ayrıca TF-IDF vektör temsil yöntemi kullanılarak kelime çıkarımı yapılmıştır. Son olarak çalışmada Multinomal Naive Bayes, SVM, Karar Ağaçları, KNN sınıflandırma algoritmaları kullanılmıştır. Sınıflandırma sonucunda elde edilen veriler incelendiğinde, SVM algoritmasının çoğu durumda Karar Ağaçlarına göre daha başarılı olduğu gözlemlenmiştir. (Parlak, 2021), (Parlak ve Uysal, 2021)

Günümüzde bakım ve idame işlemleri genellikle bakım yönetim otomasyonları üzerinden takip edilmekte olup sistem üzerinden geçmiş bakım kayıtlarına, planlanan bakım takvimlerine, sistemlerle ilgili ihtiyaç duyulabilecek önemli notlara, karşılaşılan arızalara ve yapılan müdahalelere ilişkin bilgilere erişilebilmektedir. İbrahim Burak TOSUN tarafından bir iplik firmasına ait varlık ve bakım yönetim programında yer alan arıza kayıtları kullanılarak otomatik arıza sınıflandırma çalışması gerçekleştirilmiştir. Yapılan çalışma ile karşılaşılan arızaların otomatik olarak ilgili arıza kategorisine atanarak kayıt sürecinin hızlandırılması, kolaylaştırılması, insan kaynaklı oluşabilecek hataların önüne geçilmesi, sistemin daha etkin ve verimli kullanılması amaçlanmıştır. Öznitelik çıkarımı için BOW ve TF-IDF vektör temsil yöntemleri kullanılmıştır. Sınıflandırma algoritması olarak Lojistik Regresyon, Lineer Destek Vektörü, Stokastik Gradyan Azalma Sınıflandırıcısı ve Naive Bayes kullanılmıştır. Lineer Destek Vektörü sınıflandırma algoritması bakım yönetim otomasyonundaki tüm arızaların yaklaşık %72'sini %87 oranında doğru tahmin ederek en başarı sınıflandırma algoritması olmuştur. (Tosun, 2021)

Metin sınıflandırma problemlerine makine öğrenmesi yöntemleri aracılığıyla çözüm üretmek amacıyla Sümeyra Nur ALTAN tarafından 2018 yılında bir çalışma yürütülmüştür. Çalışmada BOW ve TF-IDF metin temsil yöntemleri kullanılmıştır. Ayrıca öznitelik seçimi için Sarmalama Tabanlı öznitelik seçim yönteminden faydalanılmıştır. Çalışma WEKA programı üzerinden yürütülmüş olup sınıflandırma için Simple Logistic, Sequential Minimal Optimization, Naive Bayes, Decision Table ve Sınıflandırıcı Toplulukları yöntemleri kullanılmıştır. Simple Logistic yöntemi, TF-IDF ve Kazanç Oranı algoritması ile birlikte kullanıldığında %96.98 doğruluk oranı ile en başarılı sınıflandırma algoritması olmuştur. (Altan, 2018)

Önder ÇOBAN tarafından duygu simgesi kullanılarak paylaşılan Türkçe Twitter mesajları toplanmış ve metinler iki farklı kategori olacak şekilde etiketlenerek veri seti oluşturulmuştur. Çalışma ile Türkçe mesajlarda yer alan duygu durumlarının doğru olarak tahmin edilmesi amaçlanmaktadır. Veri seti kısa boyutlu metinlerden oluştuğu için çalışmada aynı zamanda DDİ ve makine öğrenmesi yöntemleri de kullanılmıştır. Çalışmada DDİ için Zemberek kütüphanesinden faydalanılmıştır. Kelime köklerinin bulunması, durak kelimelerin veri setinden çıkartılması ve TF-IDF ağırlıklandırma gibi işlemler ile veri ön işleme çalışmaları tamamlanmıştır. Metin sınıflandırma çalışmalarında sıklıkla başvurulan öznitelik seçim yöntemleri (Ki-kare, Karşılıklı Bilgi, Bilgi Kazanımı ve Saklı Anlam Analizi) kullanılarak veri boyutunun düşürülmesi sağlanmıştır. Sınıflandırma algoritması olarak Naive Bayes, Multinomal Naive Bayes, KNN, SVM ve Maksimum Entropi yöntemleri kullanılmıştır. Multinomal Naive Bayes, duygu analizi konusunda %92.5'lik doğruluk oranı ile en başarılı sınıflandırma algoritması olmuştur. (Çoban, 2016), (Çoban ve ark., 2015)

Metin sınıflandırmanın temelinde, metinde anlatılmak istenen duygu ve düşüncenin algılanarak metnin en uygun kategoriye atanması yer almaktadır. Ancak metinlerde yer alan bütün ifadeler metin içeriği hakkında bilgi taşımayabilir, bu gibi ifadeler içerik bütünlüğünün sağlanması amacı ile kullanılmakta ve çoğunlukla bütün metinlerde bulunduğundan ayırt edici olmamaktadır. Bu durum veri setinin boyutunun artmasına ve sınıflandırma algoritmalarının başarı oranlarının düşmesine neden olmaktadır. Bu nedenle metin sınıflandırması alanında yapılan çalışmalarda bahse konu durumun önüne geçmek için öznitelik seçme yöntemleri kullanılmaktadır. Bu kapsamda; Durmuş Özkan ŞAHİN tarafından yapılan çalışmada en doğru özniteliklerin seçilebilmesi hedeflenmiştir. DDİ yöntemleri için Zemberek kütüphanesi kullanılırken, öznitelik seçimi için ise Ki Kare, Bilgi Kazancı, Relevance Frequency, Acc ve Acc2 teknikleri

kullanılmıştır. Ayrıca sınıflandırma aşamasında SVM, KNN ve Naive Bayes algoritmaları kullanılmıştır. (Şahin, 2016)

Bilgisayar sistemlerinin gelişimine paralel olarak bu alandaki veri sayısı da hızla artış göstermiştir. Öyle ki bu verilerin depolanması için veritabanı olarak adlandırılan sistemler geliştirilmiştir. Veritabanı; kurum, kuruluş veya kişilere ait bilgisayar ortamında elektronik olarak tutulan verilerin belirli bir düzen içerisinde depolanmasına ve bu verilerin çok amaçlı kullanımına olanak sağlayan sistem olarak nitelendirilebilir. (Soyuyüce ve ark., 2003), (Köseoğlu, 2008) Buna yazarlar tarafından gündeme dair çeşitli alanlarda kaleme alınan haber niteliğindeki metinlerin dijital ortamda yayımlanması ve geçmişe dönük kaynak oluşturması açısından depolanması örnek olarak verilebilir. Benzer şekilde Mustafa Kemal Atatürk'ün talimatıyla kurulan ve ülkemizin resmi haber ajansı olarak hizmet vermekte olan Anadolu Ajansı tarafından da haber niteliğindeki yazılar yayımlanmakta ve depolanmaktadır. (Başboğa, 2010), (Bengi, 2019) İsmail Ferhat PİLAVCILAR tarafından Anadolu Ajansı'nın yayımlamış olduğu dört farklı kategoriden yaklaşık 2600 adet haber metni toplanarak bir veri seti oluşturulmuştur. Veri setinin boyutunun düşürülmesi için Joker (Wild Card) yöntemi kullanılmış ve kelimeler gövdelerine ayırılmıştır. Yapı olarak gövdelerine ayırılmış kelimelerin ağırlıklandırılması için ise TF-IDF yöntemi kullanılmıştır. Çalışmada metin sınıflandırma alanında sıkça kullanılan KNN ve Naive Bayes algoritmaları ile bu algoritmalara ait çeşitli ağırlıklandırma yöntemleri kullanılmıştır. Çalışma sonucunda elde edilen veriler değerlendirildiğinde %92.6 doğruluk oranı ile en başarılı yöntemin Naive Bayes olduğu görülmüştür. (Pilavcılar, 2007)

Elektronik ortamda haberleşme ve bilgi paylaşımı için birçok yöntem mevcuttur. En çok kullanılan elektronik haberleşme ve bilgi paylaşımı platformlarından birisi de şüphesiz ki e-posta sistemleridir. Ancak e-posta uygulamaları kullanılırken siber saldırılara karşı çok dikkatli olunması gerekmektedir. Zira internet ortamındaki kötü niyetli kişi veya guruplar özel bilgilere erişim ve dolandırıcılık gibi çeşitli sebeplerle e-posta hesaplarına spam olarak adlandırılan istenmeyen ve genellikle toplu olarak gönderilen mesajlar yollamaktadırlar. Bu tür kötü niyetli mesajları engellemenin henüz kesin bir yolu bulunmamakla beraber e-posta firmaları tarafından geliştirilmiş filtreleme teknikleri kullanılarak büyük oranda korunma sağlanmaktadır. Dolayısıyla spam mesajlarına karşı korunmak için filtreleme tekniklerinin daha da geliştirilmesi büyük önem arz etmektedir. Bunu gerçekleştirmenin en temel yollarından birisi de spam mesajlarının sınıflandırılmasında kullanılan algoritmaların daha geniş bir veri seti

üzerinde eğitilmesinden geçmektedir. Spam mesajlarının engellenmesine yönelik olarak veri sayısının arttırılması ve sınıflandırma algoritmaları üzerindeki etkilerinin analiz edilmesi amacıyla Ayşenur AKSOY tarafından 2022 yılında bir çalışma yürütülmüştür. Çalışmada DDİ kütüphanesi Zemberek ile SVM, KNN, Multinominal Naive Bayes, Karar Ağaçları, Lojistik Regresyon ve Rastgele Orman sınıflandırma algoritmalarından faydalanılmıştır. Çalışmanın sonucunda spam mesajlarının sayısını arttırmanın sınıflandırma algoritmalarının verimliliğini arttırdığı gözlemlenmiştir. (Aksoy, 2022)

Doğal dil işlemenin hukuk alanına uyarlanması ile ortaya çıkmış çok sayıda çalışma mevcuttur. Emre MUMCUOĞLU tarafından, Türk hukukunun vermiş olduğu kararlar üzerine uygulanmış herhangi bir DDİ çalışmasının bulunmadığı fark edilmiş ve bunun sonucunda dava metinlerinden mahkeme sonuçlarının tahmin edilmesini amaçlayan bir çalışma gerçekleştirilmiştir. Bu bağlamda DDİ kütüphanesi Zemberek kullanılarak çalışma için derlenen mahkeme verileri cümlelerine, kelimelerine ve köklerine ayırılmıştır. Mahkeme kararlarının tahmin edilmesi için ise SVM, Rastgele Orman ve Karar Ağaçları sınıflandırma algoritmalarının yanı sıra Geçitli Mükerrer Hücreler, tek yönlü ve çift yönlü Uzun-Kısa Vade Hafıza Ağları gibi derin öğrenme yöntemleri kullanılmıştır. Elde edilen sonuçlar diğer ülke mahkemeleri için yapılan benzer çalışma sonuçlarıyla kıyaslandığında başarı oranlarının birbirine yakın olduğu görülmüştür. (Mumcuoğlu, 2022)

Gelişen teknoloji eğitim alanında da bazı yenilikleri beraberinde getirmiştir. Buna Nimet AKSOY'un açık uçlu sınav sistemlerinden elde edilen sınav verilerinin DDİ ve makine öğrenmesi yöntemleri kullanılarak öğrenci notunun otomatik olarak belirlenmesini hedefleyen çalışması örnek olarak verilebilir. Çalışma kapsamında farklı alanlarda yapılmış olan açık uçlu sınav verileri toplanmış, veriler Bloom taksonomisine göre kategorize edilmiş ve böylelikle veri seti oluşturulmuştur. Çalışmanın asıl amacı açık uçlu sınav sistemi değerlendirmelerinin bilgisayar destekli yapılarak objektif ve hızlı sonuçlar alınmasını sağlamaktır. Veri ön işleme aşamasında metin küçük harfe çevrilmiş, veri setinde yer alan durak kelimeler ve özel karakterler çıkarılmıştır. Veri setinde yer alan metinsel ifadelerin sayısal ifadelere dönüştürülmesi, gereksiz olan özniteliklerin temizlenmesi ve veri boyutunun düşürülmesi için Kelime Sayma, TF-IDF, Doc2Vec ve Word2Vec öznitelik çıkarım yöntemleri kullanılmıştır. Çalışmada beş farklı sınıflandırma yöntemi (SVM, KNN, Naive Bayes, Karar Ağaçları ve Lojistik Regresyon) kullanılmıştır. Bloom taksonomisine göre alt düzey kategori sorularında Kelime Sayma yöntemi ile birlikte kullanılan KNN algoritması %97, üst düzey kategori sorularda ise

Doküman Vektörü yöntemi ile birlikte kullanılan Naive Bayes algoritması %86 doğruluk oranı ile en başarılı sınıflandırma algoritmaları seçilmiştir. (Aksoy, 2021)

Mobil cihazlarda yer alan ve sesli komutlarla yönlendirebildiğimiz kişisel asistanlar aslında doğal dil işlemenin bir sonucu olarak ortaya çıkmış uygulamalardır. Apple Siri, Google Asistan ve Microsoft Cortana bilinen en popüler kişisel dijital asistanlardır. Gökhan ÇELİKKAYA mobil cihazlarda kullanılabilecek Türkçe destekli kişisel asistan geliştirebilmek için ileri seviye DDİ teknikleri kullanarak 2015 yılında bir çalışma gerçekleştirmiştir. Çalışma kapsamında tasarlanan kişisel asistan komutları sesli olarak alabilmekte, ancak sesli yanıt döndürememektedir. Mobil asistan sms gönderimi, e-posta gönderimi, numarayı veya kayıtlı kişiyi arama, yüklü uygulamaları çalıştırma, internet üzerinden arama yapma, hava durumu bilgisi döndürme, harita hizmetleri verme, döviz hesaplama ve trafik yoğunluk bilgisi döndürme gibi işlemleri yerine getirebilmektedir. Bahse konu proje mobil uygulama ve sunucu tarafı olmak üzere iki temel kısımdan meydana gelmektedir. Mobil uygulama kısmında kullanıcıdan gelen sesli komutlar yazıya dönüştürülmekte, sunucuya yazılı komutun gönderimi ve sunucudan gelen yanıtın ekrana yansıtılması gerçekleştirilmektedir. Sunucu tarafında ise gelen yazılı istekler DDİ yöntemleri kullanılarak algılamakta ve mobil uygulamaya cevap döndürülmektedir. Sunucuya gelen istekler doğrudan DDİ araçlarına aktarılmadan önce olası hataların giderilebilmesi için metin normalleştiriciden geçirilmektedir. Ön işlemden geçirilen yazılı komut içerisinde DDİ araçlarının da bulunduğu melez bir modele aktarılarak gerekli cevabın döndürülmesi sağlanmaktadır. Geliştirilen melez model test verisi üzerinde yaklaşık %98 oranında başarı elde ederken, tüm sistemin başarı oranı yaklaşık %71 olmuştur. (Çelikkaya, 2015)

Yapay Sinir Ağı (YSA)'nın yaygınlaşmasıyla birlikte Google çalışanı Thomas Mikolov ve ekip arkadaşları tarafından Word2Vec yöntemi geliştirilmiştir. (Mikolov ve ark., 2013) Word2Vec, tahmin tabanlı bir kelime temsil yöntemi olup arka planda YSA ile birlikte CBOW (Continuous Bag of Words) ve Skip-Gram Model'lerini kullanarak gerekli işlemleri gerçekleştirmektedir. DDİ alanında metinlerin sınıflandırılmasında kullanılan YSA tabanlı bu sistem, metni makinenin anlayabileceği bir vektöre dönüştürerek sanallaştırmaktadır. Sözcüklerin vektörize işlemi sonrasında hala anlam bütünlüklerini koruyabilmelerini sağlamak amacıyla kelimelerin sağındaki ve solundaki ifadeler de dikkate alınmaktadır.

Samet KAYA tarafından 2018 yılında yapılan çalışmada 20 yazara ait Türkçe dilinde yazılmış 120 eser üzerinde yazar tahmini çalışması yapılmıştır. (Kaya, 2018)

Kitaplardan 20 tanesi eğitim verisi olarak kullanılırken 100 adedi ise test verisi olarak kullanılmıştır. Eserler veri ön işleme adımlarına tabi tutulduktan sonra N-gram algoritması ile kelime bazında öznitelik ve yazarın SÜA çıkartılmıştır. Sınıflandırma aşamasında ise Navie Bayes sınıflandırıcı kullanılarak sınıflandırma işlemi gerçekleştirilmiştir. Eğitim verisi olarak ayrılan eserler yazar etiketlemesi yapılarak sınıflandırma algoritması eğitilmiş ve test için ayrılan etiketsiz veriler üzerinde tahmin işlemi yapılmıştır.

Salimkan Fatma TAŞKIRAN tarafından 2021 yılında yapılan çalışmada DDİ teknikleri ile akademik metinler üzerinde kümeleme yapılmıştır. (Taşkıran, 2021) DDİ teknikleri olarak; Snowball Stemmer ailesinin Türkçe Gövdeleme algoritması, kök çözümleme için ise Turkish-Lemmatizer kullanılmıştır. Frekans ve YSA tabanlı metin temsil yöntemleri kullanılarak farklı kümeleme algoritmalarından alınan sonuçlar karşılaştırılmış ve analiz edilmiştir. Frekans tabanlı metin temsil yöntemi olarak TF-IDF yöntemi kullanılmış, YSA tabanlı metin temsil yöntemi olarak da Word2Vec yöntemi kullanılmıştır. Kümeleme işlemi için ise K-Means, K-Medoids, İngilizce literatürde OPTICS olarak bilinen “Ordering Points To Identify Cluster Structure” ve Affinity Propagation, algoritmaları kullanılmıştır. Elde edilen sonuçlar karşılaştırılarak çıkarımlarda bulunulmuştur.

3. MATERYAL VE YÖNTEM

Birinci bölümde genel olarak bahsedilen metin sınıflandırma süreci, bu bölümde detaylandırılmış ve kullanılan yöntemler ile birlikte açıklanmıştır.

Bu tez çalışmasında; Anaconda versiyon 1.9.0, Python versiyon 3.9.12, Jupyter Notebook versiyon 6.4.11 ve ekstra işlem gücü gerektiren durumlarda Google Colab platformu kullanılmıştır. Ayrıca Türkçe DDİ araçlarından faydalanmak için Java tabanlı Zemberek kütüphanesinin 0.17.1 versiyonu kullanılmıştır.

3.1. Veri Seti

Veri seti gazetelerin internet sitesinden veri kazıma (Web scraping) yöntemleri kullanılarak elde edilmiştir. Veri setinde farklı alanlarda (Gündem, spor, pazar vb.) yazarlık yapan 51 adet köşe yazarı yer almakta olup söz konusu yazarlara ait eser sayısı Çizelge 3.1'de sunulmuştur. İlgili çizelge incelendiğinde her yazara ait en az 2, en fazla 100 adet olmak üzere toplamda 4923 adet köşe yazısı bulunduğu görülmektedir.

Çizelge 3.1. Veri setinde yer alan yazarlar ve yazarlara ait eser sayıları

No	Yazar	Eser Sayısı	No	Yazar	Eser Sayısı
1	Yazar 1	99	27	Yazar 27	100
2	Yazar 2	100	28	Yazar 28	100
3	Yazar 3	100	29	Yazar 29	100
4	Yazar 4	100	30	Yazar 30	100
5	Yazar 5	100	31	Yazar 31	100
6	Yazar 6	100	32	Yazar 32	100
7	Yazar 7	99	33	Yazar 33	99
8	Yazar 8	100	34	Yazar 34	100
9	Yazar 9	100	35	Yazar 35	100
10	Yazar 10	100	36	Yazar 36	100
11	Yazar 11	95	37	Yazar 37	100
12	Yazar 12	99	38	Yazar 38	100
13	Yazar 13	100	39	Yazar 39	31
14	Yazar 14	100	40	Yazar 40	100
15	Yazar 15	100	41	Yazar 41	100
16	Yazar 16	100	42	Yazar 42	100
17	Yazar 17	100	43	Yazar 43	100
18	Yazar 18	100	44	Yazar 44	100
19	Yazar 19	100	45	Yazar 45	100
20	Yazar 20	100	46	Yazar 46	100
21	Yazar 21	100	47	Yazar 47	100
22	Yazar 22	100	48	Yazar 48	100
23	Yazar 23	100	49	Yazar 49	100
24	Yazar 24	100	50	Yazar 50	2
25	Yazar 25	99	51	Yazar 51	100
26	Yazar 26	100			

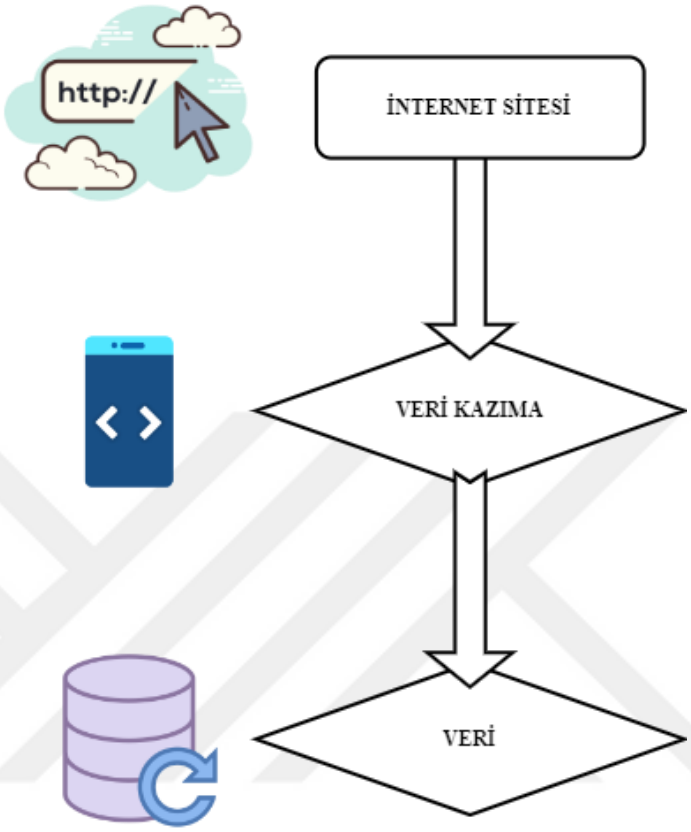
Kazıma yöntemi ile elde edilen veriler “csv” formatında tablo olarak tutulmakta olup toplamda 5 sütundan meydana gelmektedir. Tablo sütunları Şekil 3.1’de gösterildiği gibi sırasıyla yazının yazıldığı tarih (Tarih), yazarın adı (Yazar), yazının manşeti (Başlık), yazının alındığı gazete sitesinin bağlantısı (Link) ve son olarak köşe yazısının metni (Metin) bilgilerini içermektedir.

	Tarih	Yazar	Başlık	Link	Metin
0	28 Nisan 2023	Yazar_1	Ruhani lider Papa Francesco’nun Macaristan'ı z...	Link_1	\r\n\r\nVatikan Devlet Başkanı ve Katolik Kili...
1	20 Nisan 2023	Yazar_1	Dünyanın en kalabalık ülkesi Hindistan	Link_2	\r\n\r\nBirleşmiş Milletler tarafından yapılan...
2	04 Nisan 2023	Yazar_1	NASA Artemis 2 misyonu için Ay'a gidecek astro...	Link_3	\r\n\r\nAmerikan Ulusal Havacılık ve Uzay Dair...
3	04 Nisan 2023	Yazar_1	Yeni İngiltere Kralı Charles'ın silüetinin bul...	Link_4	\r\n\r\nİngiltere Kraliçesi II. Elizabeth’in ö...
4	01 Nisan 2023	Yazar_1	Sudan'da devam eden siyasi kriz 17 ay sonra im...	Link_5	\r\n\r\nSudan'da askerler ve siviller arasında...

Şekil 3.1. Veri seti sütunları

Veri setinin gazete sitelerinden veri kazıma yöntemleri ile alınmasında Python programlama dili, Requests, BeautifulSoup, Pandas, Numpy ve Selenium kütüphaneleri kullanılmıştır. Requests kütüphanesi ile http veya https protokolü kullanan internet siteleriyle kurulan bağlantı sayesinde temel web işlemleri (Get, post vb.) gerçekleştirilebilmektedir. Kütüphanenin “get” fonksiyonu kullanılarak ilgili gazete sitesinin bilgilerine erişim sağlanmıştır. BeautifulSoup kütüphanesi, HTML ve XML dilleri kullanılarak oluşturulmuş internet sitelerinden ihtiyaç duyulan etiketlerin (div, h, p vb.) bilgileri alınarak verinin elde edilmesine olanak sağlamaktadır. (Jin, 2021) Akış şeması Şekil 3.2’de görüldüğü üzere ilk olarak Requests kütüphanesi kullanılarak verilerin çekileceği internet sitesine erişim sağlanmış, daha sonra BeautifulSoup ve Selenium kütüphaneleri yardımıyla veri kazıma işlemi yapılarak ihtiyaç duyulan veriler

elde edilmiş ve son olarak Pandas kütüphanesi ile veriler tablo formatında kaydedilerek veri seti oluşturulmuştur.



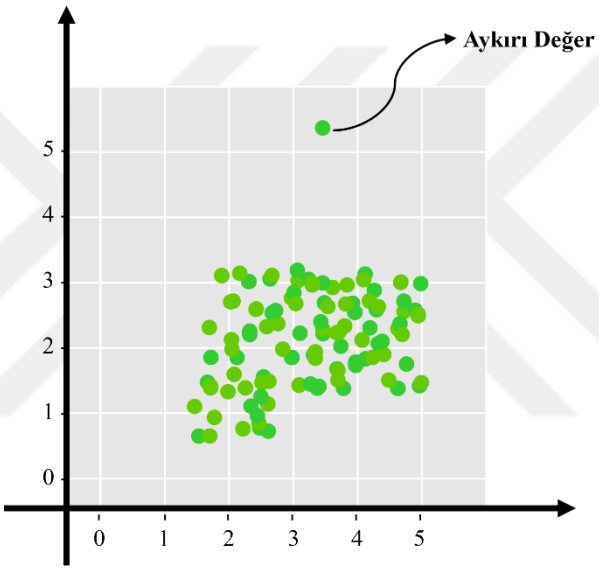
Şekil 3.2. Veri kazıma işlem adımları (Jin, 2021)

3.2. Veri Ön İşleme

Veri ön işleme, makine öğrenmesi algoritmalarının veri setinden en yüksek düzeyde faydalanabilmesini ve en doğru sonucu elde edebilmesini sağlamak amacıyla veri setiyle çalışmaya başlanmadan önce uygulanması gereken yöntemler ve teknikler bütünü olarak adlandırılmaktadır. (Oğuzlar, 2003), (Famili ve ark., 1997) Söz konusu veri ön işleme teknik ve yöntemleri kullanılarak veride gerekli bazı değişiklikler yapılmakta ve veri seti çalışma için en uygun biçime dönüştürülmektedir. Bu değişiklikler; aykırı verilerin veri setinden çıkartılması, veri normalizasyonu, metnin küçük harfe çevrilmesi, kelimelerin kök ve eklerine ayrılması, metinde yer alan satır boşluklarının, durak kelimelerin ve noktalama işaretlerinin temizlenmesi olarak özetlenebilir.

3.2.1. Aykırı / Uç verilerin veri setinden çıkarılması

Şekil 3.3'te görüldüğü üzere veri setindeki diğer örneklemlerden belirgin derecede farklılık gösteren gözlem, aykırı veya uç değer olarak tanımlanmaktadır. Aykırı gözlemler, veri setinin geri kalanından farklı davranışlar sergilediğinden dolayı veri bütünlüğünün bozulmasına ve sınıflandırma başarısının düşmesine neden olmaktadır. Z-score ve Tukey's gibi yöntemler kullanılarak aykırı değer tespiti yapılabilmektedir. (Shiffler, 1988) Tespit edilen aykırı değerler veri kümesinden çıkartılarak veya yeni değerler atanarak elimine edilebilmektedir.



Şekil 3.3. Aykırı değer

3.2.2. Metnin sözcüklerine ayrılması (Tokenization)

Sözcüklere ayırma, metnin daha küçük ve anlamlı alt birimlere bölünmesi sonucunda cümlenin yapı taşı olan kelimelerin elde edilmesini ifade etmektedir. Bu işlem Çizelge 3.2'de gösterildiği üzere cümle içerisinde yer alan boşluklar ve noktalama işaretleri referans alınarak gerçekleştirilmektedir.

Çizelge 3.2. Metnin sözcüklerine ayırılası

Dil	Cümle	Kelimeler
Türkçe	Doğal dil işleme	Doğal, dil, işleme
İngilizce	Natural language processing	Natural, language, processing

3.2.3. Zemberek

Zemberek, Ahmet Afşın AKIN tarafından Java programlama dili kullanılarak geliştirilmiş olup Türkçe metinler üzerinde DDİ yöntemlerinin uygulanmasına olanak sağlayan bir kütüphanedir. (Akın ve Akın, 2007) Kütüphane kullanılarak yazım denetimi, cümle ayırma, yanlış yazılmış kelimelerin düzeltilmesi, yazı dili tespiti, morfolojik analiz yöntemi ile kelime kök ve eklerinin bulunması gibi işlemler kolaylıkla gerçekleştirilebilmektedir. Çizelge 3.3'te kütüphane modülleri ve kullanım alanları hakkında kısaca bilgi verilmiştir.

Çizelge 3.3. Zemberek modülleri (Karasoy, 2019)

Modül Adı	Maven ID	Açıklama
Core	zemberek-core	Özel veri yapılarını ve yardımcı sınıfları içermektedir.
Morphology	zemberek-morphology	Türkçe dili için morfolojik analiz, belirsizlik çözümü, anlam ayrımı ve kelime üretimi amacıyla kullanılmaktadır.
Tokenization	zemberek-tokenization	Türkçe sınıflandırma ve cümle sınırlarının belirlenmesini (cümle çıkarımı) sağlamaktadır.
Normalization	zemberek-normalization	Basit yazım denetleyicisi, hatalı metin düzeltme ve doğru kelime önerileri için kullanılmaktadır.
NER	zemberek-ner	Varlık ismi çıkarımı (NER) için kullanılmaktadır.
Classification	zemberek-classification	Türkçe metinlerin sınıflandırılmasını sağlayan modüldür.
Language Identification	zemberek-lang-id	Hızlı metin dili tanıma için kullanılmaktadır.
Language Modelling	zemberek-lm	Dil modeli karşılaştırma algoritmasıdır.
Applications	zemberek-apps	Konsol uygulamaları için kullanılan modüldür.
gRPC Server	zemberek-grpc	Diğer diller için gRPC sunucusuna erişimi sağlamaktadır.
Examples	zemberek-examples	Kütüphane kullanım örneklerinin yer aldığı modüldür.

3.2.3.1. Morfolojik analiz

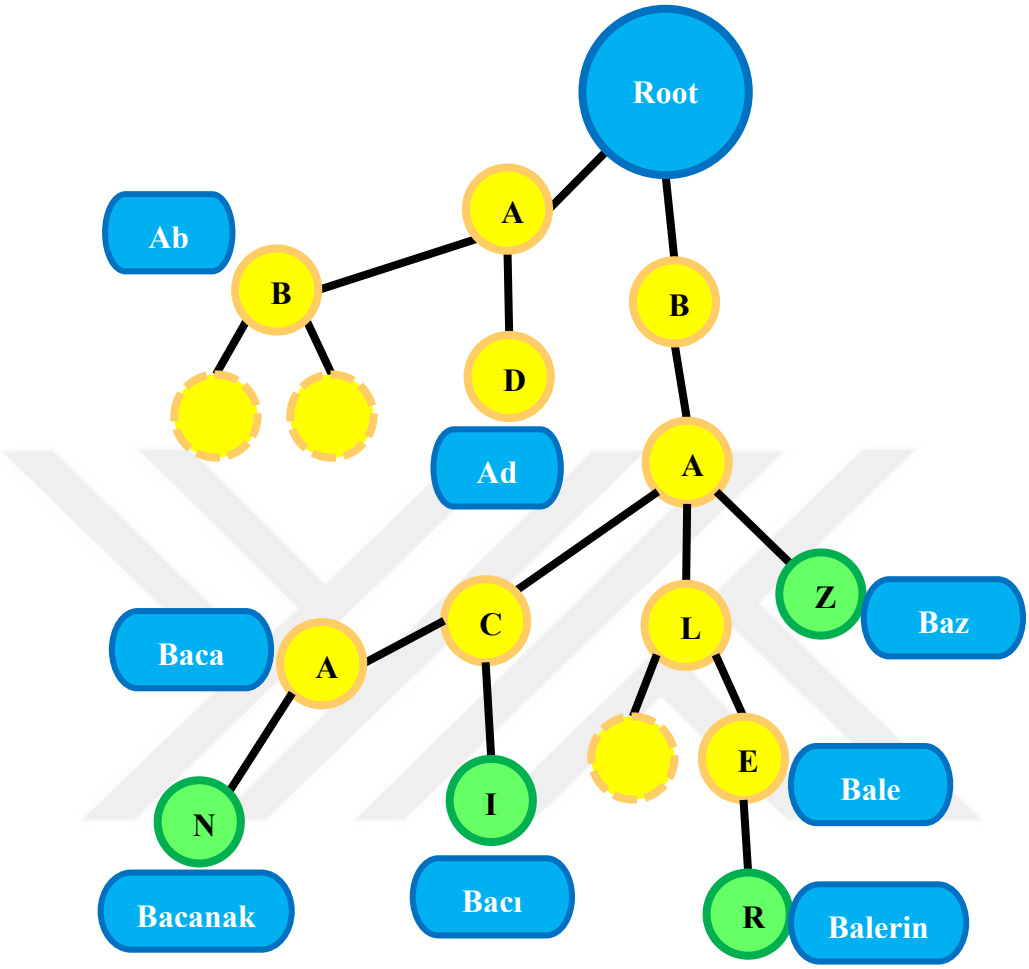
Morfolojik analiz, bir kelimenin biçim yapısının belirlenebilmesi için kök ve eklerinin birbirinden ayrılması olarak tanımlanmaktadır. Kök, kelimenin en küçük yapı taşı ve anlamlı en küçük parçasıdır. Ek ise tek başına bir anlam ifade etmemekle birlikte kelime köklerine getirilerek farklı amaçlar doğrultusunda kullanılmaktadır. Türkçe sondan eklemeli bir dil olduğundan kelimeye gelen ekler kelimenin yapısında değişikliklere neden olabilmekte ve anlamını değiştirebilmektedir. Buna örnek olarak; Çizelge 3.4'te isim kök olan "masa" ve fiil kök olan "oku" sözcüklerinin yapım eki almadan ve sırasıyla 1, 2, 3 adet yapım eki aldığında elde edilebilecek farklı kelimelerin sayısı verilmiştir. Söz konusu çizelgeden de görülebileceği üzere, sadece "oku" fiiline 3 ek getirilmesi durumunda yaklaşık olarak 1.5 milyon yeni sözcük türetilebilmektedir. Bu nedenle Türkçe gibi sondan eklemeli dillerde kelimelerin köklerini ve eklerini bulabilmek bir hayli zorlaşmaktadır. (Gündoğdu ve Duru, 2016)

Çizelge 3.4. Yapım ekleriyle "masa" ve "oku" köklerinden elde edilebilecek farklı kelime sayısı (Oflazer, 2016)

Kök	Yapım Eki	Sözcük	Toplam
Masa	0	112	112
	1	4,663	4,775
	2	49,640	54,415
	3	493,975	548,390
Oku	0	702	702
	1	11,366	12,068
	2	112,877	124,945
	3	1,336,266	1,461,211

Zemberek kütüphanesinde Türkçe kelimelerin kökünü bulabilmek için tasarlanmış olan özel bir ağaç modeli bulunmaktadır. Tasarlanan model sayesinde kelime kökleri son derece hızlı ve kolay bir biçimde bulunabilmektedir. Kökler Şekil 3.4'te sunulduğu gibi harf bazında ağaç üzerine yerleştirilmekte ve etiketlenmektedir. Örneğin "balo" isim kökü sırasıyla B-A-L-O harfleri ile ağaca eklenmiş ve etiketlenmiştir. Modelin en önemli avantajlarından biri ise her kök için ayrı ayrı düğüm oluşturmaya gerek duymadan var olan düğümlerin kullanılabilmesidir. Bu yöntem ile sadece ağaçta eksik olan kök düğümler oluşturulduğundan gereksiz düğümlerin oluşturulması

engellenir ve böylelikle hafızadan tasarruf edilmiş olunur. (Ekin, 2020) Örneğin “balerin” kökü B-A-L-E düğümlerinden sonra gelen R düğümüyle devam ettirilmiştir.



Şekil 3.4. Zemberek kök ağacı yapısı (Akın ve Akın, 2007)

3.2.3.2. Cümle çıkarma / bölme

Zemberek kütüphanesi metni kelime, kök ve eklerine ayırabildiği gibi cümlelerine de ayırabilmektedir. Kütüphane bu işlemi gerçekleştirebilmek için içerisinde yer alan TurkishSentenceExtractor sınıfını kullanmaktadır. Sınıf girdi olarak bir metin alırken, çıktı olarak metnin birbirinden ayrılmış cümlelerini döndürmektedir. (Uludogan ve ark., 2017) Söz konusu sınıf cümlenin başlangıç ve bitiş noktalarını bulabilmek için kural tabanlı basit kombinasyonları (Cümlelerin sonunda nokta bulunması, cümlelerin büyük harfle başlaması vb.) ve İkili Ortalamalı Algılayıcı modelini kullanmaktadır. Sınıfı kullanmak için üç farklı yöntem bulunmakta olup bahse konu yöntemler aşağıda sıralanmıştır. (Özker, 2019)

1. fromParagraph(): Bu fonksiyon ile sadece tek paragraftan oluşan metin üzerinde işlem yapılabilmektedir. İşlem sonucunda metin içerisinde yer alan cümleler birbirinden ayrılmış durumda bir liste ile döndürülmektedir.

2. fromParagraphs(): Bu fonksiyon ile birden çok paragraftan oluşan metin üzerinde işlem yapılabilmektedir. İşlem sonucunda metin içerisinde yer alan cümleler birbirinden ayrılmış durumda bir liste ile döndürülmektedir.

3. fromDocument(): Bu fonksiyon ile bir dosyada yer alan her paragraf için fromParagraph() fonksiyonu çalıştırılarak işlem yapılmaktadır. İşlem sonucunda dosya içerisinde yer alan cümleler birbirinden ayrılmış durumda bir liste ile döndürülmektedir.

3.2.4. Satır boşluklarının ve noktalama işaretlerinin temizlenmesi

Sınıflandırmada kullanılacak metin içerisinde yer alan noktalama işaretleri ve satır boşlukları dilbilgisi kuralları gereği kullanılmak zorunda olduğundan, yazarın üslubunun tespit edilmesine yönelik herhangi bir bilgi taşımamaktadır. Bu nedenle sınıflandırma işlemi öncesinde metinde bulunan noktalama işaretleri ve satır boşlukları Çizelge 3.5'te gösterildiği gibi metin içerisinden temizlenerek algoritmaların daha doğru sonuç vermesi sağlanmaktadır.

Çizelge 3.5. Satır boşluklarının ve noktalama işaretlerinin temizlenmesi

Veri kazıma sonucu elde edilen metin	Satır boşluklarının ve noktalama işaretlerinin temizlenmesi sonucunda elde edilen metin
\r\n\r\nSudan'da askerler ve siviller arasında çıkan anlaşmazlık üzerine ordu 25 Ekim 2021 tarihinde olağanüstü hâl ilan etmiş ve yaklaşık olarak 17 aylık bir süredir ülke yönetimini ele geçirmiştir. \r\nBu durum askerler ile siviller arasında çatışmaların çıkmasına yol açmış ve birçok kişinin hayatını kaybetmesine neden olmuştur.	sudanda askerler ve siviller arasında çıkan anlaşmazlık üzerine ordu 25 ekim 2021 tarihinde olağanüstü hal ilan etmiş ve yaklaşık olarak 17 aylık bir süredir ülke yönetimini ele geçirmiştir bu durum askerler ile siviller arasında çatışmaların çıkmasına yol açmış ve birçok kişinin hayatını kaybetmesine neden olmuştur

3.2.5. Küçük harf dönüşümü

Veri ön işlemenin en önemli adımlarından birisi olan küçük harf dönüşümü dokümanda yer alan bütün büyük harflerin küçük harfe dönüştürülmesini hedeflemektedir. Çünkü metin içerisinde hem büyük hem de küçük harflerle yazılmış olan aynı kelimelerin değişik kombinasyonlarının her biri bilgisayar tarafından farklı bir öznitelikmiş gibi değerlendirilerek sınıflandırma işlemi gerçekleştirilmektedir. Bu durum

toplam öznitelik sayısının artmasına, veri boyutunun gereksiz yere yükselmesine, sınıflandırma süresinin uzamasına ve elde edilen başarı oranının düşmesine neden olmaktadır. Bu nedenle; metinde yer alan ve bünyesinde büyük harf barındıran bütün ifadelerin veri ön işleme aşamasında küçük harfe dönüştürülmesi sağlanmaktadır. Ayrıca Çizelge 3.6'da görüldüğü üzere Türkçe ve İngilizce gibi farklı dillerin yapısal özelliklerine bağlı olarak söz konusu dönüşüm farklılıklar gösterebilmektedir.

Çizelge 3.6. Türkçe ve İngilizce için küçük harf dönüşümü

Dil	Orijinal form	Küçük harf dönüşümü
Türkçe	K, I	k, ı
İngilizce	K, I	k,i

3.2.6. Durak kelimelerin (Stop words) veri setinden çıkartılması

Cümle bütünlüğünün sağlanması amacıyla kullanılan ve metnin konusu kapsamında herhangi bir bilgi taşımayan etkisiz kelimeler durak kelime ya da stop words olarak adlandırılmaktadır. Çizelge 3.7'den de görülebileceği üzere dillere göre değişiklik gösteren bu kelimeler veri boyutunu azaltmak ve metnin sınıflandırma başarısını arttırmak amacıyla veri ön işleme aşamasında veri kümesinden çıkartılmaktadır.

Çizelge 3.7. Örnek durak kelimeler

Dil	Kelimeler
Türkçe	bazen, elbet, gerçi, hiçbir, ile, madem, oysa
İngilizce	above, best, can, every, has

3.2.7. Normalizasyon

Normalizasyon işlemi, makine öğrenmesi uygulamaları için kullanılan bir veri ön işleme tekniği olup sadece veri setindeki sayısal özniteliklere uygulanabilmektedir. Normalizasyon; veri boyutunu küçültmek, birbirine çok uzak sayısal değerler arasındaki farkı oransal olarak bozmadan bir ölçek dahilinde biçimlendirme yaparak veri dağınıklığını azaltmak, veri setini daha anlamlı ve yorumlanabilir duruma getirerek makine öğrenmesi uygulamalarının daha iyi sonuçlar elde etmesini sağlamak amacıyla kullanılmaktadır. (Dondurmacı ve Çınar, 2014) Ancak her veri seti normalizasyon işlemi gerektirmeyebilir. Literatürde birçok normalizasyon yöntemi bulunmakta olup bunlardan

en çok kullanılan ve bilinenleri Minimum-Maksimum (Min-Max), Z-skor, Ondalık Ölçeklendirme, Medyan ve Sigmoid normalizasyon yöntemleridir. (Jayalakshmi ve Santhakumaran, 2011), (Munkhdalai ve ark., 2019)

Bu çalışmada Min-Max normalizasyon tekniği kullanılmış ve veri seti 0 ile 1 arasında ölçeklendirilerek normalize edilmiştir. Min-Max normalizasyon tekniği verileri doğrusal olarak normalize etmektedir. Minimum verinin alabileceği en küçük değeri, maksimum ise verinin alabileceği en büyük değeri ifade etmektedir. (Yavuz ve Deveci, 2012) Bu işlemi gerçekleştirebilmek için aşağıda yer alan Denklem 3.1 kullanılmaktadır. (Gültepe, 2019), (Erdal ve Yapraklı, 2016)

$$x' = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{3.1}$$

Denklem 3.1'de bulunan;

x' = Normalize edilmiş veriyi,

x_i = Girdi değerini (Normalize edilecek veriyi),

x_{min} = Veri seti içerisinde bulunan en küçük sayıyı,

x_{max} = Veri seti içerisinde bulunan en büyük sayıyı ifade etmektedir.

3.3. Öznitelik Çıkartma

Öznitelik çıkarma, sınıflandırma algoritmalarından daha yüksek başarı elde edebilmek için veri setinde bulunan işlenmemiş ve anlamsız verilerden çeşitli yöntemlerle algoritmanın anlayabileceği ve sınıflandırma aşamasında kullanabileceği yeni veriler çıkartılması olarak ifade edilmektedir. Metin tabanlı sınıflandırma uygulamalarında metin içerisinden çok sayıda istatistiki öznitelik çıkartılabilir (Kullanılan noktalama işaretlerinin sayısı, büyük harflerin sayısı, cümlelerin uzunluk ortalaması, cümle içerisinde geçen durak kelimelerin sayısı vb.), ancak bütün bu öznitelikler arasında metnin kapsamına ilişkin en fazla bilgi yine de sözcükler tarafından tutulmaktadır. Bu nedenle, metin tabanlı sınıflandırma uygulamalarında dokümanı oluşturulan kelimelerin bir şekilde sınıflandırmaya dahil edilmesi gerekmektedir. Bunun için metin temsil yöntemleri kullanılmaktadır. Metin temsil yöntemleriyle metni oluşturan kelimeler bazı kurallar çerçevesinde sayısallaştırılarak sınıflandırmaya uygun hale getirilmektedir. Bu çalışmada metin temsil yöntemleri olarak BOW ve TF-IDF kullanılmıştır.

3.3.1. Kelime çantası (Bag of words-BOW)

BOW yönteminde metin kelimelerine ayırılmakta ve her bir terim bir özniteliği ifade etmektedir. BOW oluşturulurken dilbilgisi kuralları ve hatta kelimelerin metin içerisindeki sıralaması göz ardı edilmektedir. Metni oluşturan kelimeler BOW olarak adlandırılan yapıda tutularak kelime hazinesini oluşturmakta ve metnin vektör temsil yöntemi ile sayısallaştırılmasında kullanılmaktadır.

Aşağıdaki örnekte metinden BOW oluşturulması ve bu yöntem ile metnin sayısallaştırılması gösterilmiştir.

- Metin = [‘Ayşe bugün doğum gününü kutlayacak’
  ‘Mert bir sonraki hafta doğum gününü bir daha kutlayacak’]
- BOW = [‘Ayşe’,
  ‘bir’,
  ‘gününü’,
  ‘bugün’,
  ‘hafta’,
  ‘daha,
  ‘Mert’,
  ‘kutlayacak’,
  ‘doğum’,
  ‘sonraki’]
- Çıktı = [[1, 0, 1, 1, 0, 0, 0, 1, 1, 0],
  [0, 2, 1, 0, 1, 1, 1, 1, 1, 1]]

Yukarıdaki örnekten de görülebileceği üzere BOW yöntemi veri setine uygulandıktan sonra veri setinde yer alan benzersiz kelime sayısı kadar öznitelik oluşmaktadır.

3.3.2. TF-IDF (Term Frequency- Inverse Document Frequency)

Terim Sıklığı-Ters Doküman Sıklığı ya da kısaltılmış haliyle TF-IDF, veri setinde yer alan herhangi bir metnin içerisinde geçen bir terimin ilgili metin için önem derecesinin istatistiksel yöntemlerle belirlendiği temsil şeklidir. Burada terim sıklığı (TF) ilgili terimin işlem yapılan metindeki sıklığını veya frekansını ifade etmektedir. TF değeri

hesaplanırken metinde geçen bütün sözcüklerin önem derecesinin aynı olduğu kabul edilmektedir. (Taşkıran, 2021)

TF değerinin hesaplanmasında birkaç farklı yöntem bulunmakta olup bunlardan bazılarına aşağıda (Denklem 3.2 – 3.4) yer verilmiştir. (Manning ve ark., 2010)

$$Ham\ Frekans = \frac{Terimin\ belgedeki\ tekrar\ sayısı}{Belgedeki\ toplam\ kelime\ sayısı} = f_{t,d} \tag{3.2}$$

$$Binary\ Frekans = \ Belge\ içerisinde\ terimin\ olup\ olmadığı = (0,1) \tag{3.3}$$

$$Logaritmik\ Olarak\ Ölçeklenmiş\ Frekans = \ \log(1 + f_{t,d}) \tag{3.4}$$

Doküman sıklığı (DF) ilgili kelimenin diğer metinlerdeki frekansını, Ters doküman sıklığı (IDF) ise DF değerinin logaritmasını ifade etmektedir. IDF, tüm belgede sıklıkla geçen ve dolayısıyla frekansı metindeki diğer kelimelere göre daha yüksek olan terimlerin ağırlığını azaltıp, belgede nadiren geçen frekansı düşük ve belge için önem arz eden kelimelerin ağırlığını arttıran faktördür. IDF'in sıfıra yaklaşması o kelimenin veri seti için önemsiz olduğunu göstermektedir. Denklem 3.5'te görüldüğü üzere IDF değeri toplam doküman sayısının ilgili kelimenin geçtiği doküman sayısına bölünmesinden elde edilen sonucun logaritması alınarak hesaplanmaktadır. (Christian ve ark., 2016)

$$IDF = \log\left(\frac{Veri\ setindeki\ toplam\ belge\ sayısı}{Terimin\ geçtiği\ belge\ sayısı}\right) \tag{3.5}$$

TF-IDF bu iki değerin çarpımının sonucunda elde edilen ve 0 ile 1 arasında değişkenlik gösteren bir frekanstır. Aynı zamanda terimin bağlı bulunduğu dokümanı ne kadar temsil ettiğinin de bir göstergesidir. TF-IDF değerinin yüksek olması kelimenin metin için daha önemli olduğu ve metni daha fazla temsil ettiği anlamını taşımaktadır.

Örneğin; Çizelge 3.8'deki gibi 3 farklı doküman içeren bir veri setinin bazı terimleri için hesaplanan TF-IDF değerlerine aşağıda yer verilmiştir.

- Birinci dokümanda yer alan "havada" kelimesi için;

  $$TF = \frac{1}{5} = 0.2$$

  $$IDF = \log\left(\frac{3}{1}\right) = 0.47$$

$TF - IDF = (0.2\ x\ 0.47) = 0.095$ olarak hesaplanmıştır.

- Birinci dokümanda yer alan "güzel" kelimesi için;

  $TF = \frac{2}{5} = 0.4$

  $IDF = \log\left(\frac{3}{2}\right) = 0.17$

  $TF - IDF = (0.4\ x\ 0.17) = 0.068$ olarak hesaplanmıştır.

- İkinci dokümanda yer alan "güzel" kelimesi için;

  $TF = \frac{1}{6} = 0.16$

  $IDF = \log\left(\frac{3}{2}\right) = 0.17$

  $TF - IDF = (0.16\ x\ 0.17) = 0.027$ olarak hesaplanmıştır.

Hesaplanan TF-IDF değerleri göz önünde bulundurulduğunda: Birinci dokümanda yer alan "havada" kelimesinin TF-IDF değeri (0.095) ile aynı dokümanda yer alan "güzel" kelimesinin TF-IDF değeri (0.068) karşılaştırıldığında "havada" kelimesinin ilgili metin içerisinde bulunan "güzel" kelimesinden az geçmesine rağmen veri seti içerisinde yer alan metinlerde bulunma sayısı az olduğundan TF-IDF skoru yüksektir, bu da metin için daha önemli olduğu sonucunu ortaya çıkarmaktadır. Ancak birinci dokümanda yer alan "güzel" kelimesiyle (TF-IDF = 0.068) ikinci dokümanda yer alan "güzel" kelimesinin (TF-IDF = 0,027) TF-IDF değerleri karşılaştırıldığında, kelimenin birinci metinde iki defa geçmesine bağlı olarak TF-IDF değerinin daha yüksek çıktığı sonucuna ulaşılmaktadır.

Çizelge 3.8. TF-IDF için doküman örneği (Kaya, 2016)

Metin
Bülbül güzel güzel uçuyormuş havada.
Bülbül güzel koku gelen tarafa uçmuş.
Mis gibi koku duymuş o sahada.

3.4. Öznitelik Seçimi

Öznitelik çıkarımı aşamasında kullanılan BOW ve TF-IDF yöntemleri metne uygulandıktan sonra öznitelik sayısı on binleri ve hatta veri kümesinin genişliğine bağlı olarak yüz binleri bulabilmektedir. Öznitelik sayısının bu denli artış göstermesi önemsiz öznitelik sayısının ve veri boyutunun artmasına neden olmaktadır. Bu durum

sınıflandırma algoritmalarının başarısını düşürmekte ve aynı zamanda sınıflandırma süresini arttırmaktadır. Çünkü veri setinde bulunan her öznitelik aynı oranda bilgi taşımamakta ve buna bağlı olarak sınıflandırma aşamasında ayırt ediciliği farklılık göstermektedir. Ayırt ediciliği fazla olan öznitelikler metnin kapsamına ilişkin daha fazla bilgi taşımakta ve sınıflandırma başarısını arttırmaktadır. Bu nedenle metin sınıflandırma uygulamalarında doğru özniteliklerin seçilmesi büyük önem taşımaktadır.

Literatürde farklı yaklaşımlarla doğru özniteliğin belirlenmesini hedefleyen birçok öznitelik seçim yöntemi bulunmaktadır. Bu çalışmada öznitelik seçim yöntemi olarak Ki-kare ve Korelasyon Matrisi tekniklerinden faydalanılmıştır.

3.4.1. Ki-kare (Chi-square-CHI2)

Ki-kare yöntemini kullanılacak verilerin durumuna göre Ki-kare Uygunluk ve Ki-kare Bağımsızlık olmak üzere ikiye ayırmak mümkündür. Bu çalışmada metin tabanlı sınıflandırma yapılması nedeniyle kategorik veriler kullanılmış ve buna bağlı olarak Ki-kare Bağımsızlık yöntemi tercih edilmiştir. Ki-kare Bağımsızlık yöntemi iki olayın bağımsızlığını ölçmek için kullanılan istatistiki bir test yöntemidir. Yöntem girdi parametresi olarak hedef özniteliği ve sınıflandırma eğitiminde kullanılacak olan diğer öznitelikleri almaktadır. Ki-kare Bağımsızlık testi, her bir benzersiz özniteliğin hedef öznitelik ile olan bağımsızlık derecesini hesaplayarak niteliklerin birbiri arasında olan ilişkilerini tespit etmektedir. Bunun için Denklem 3.6'da yer alan formül kullanılmaktadır. Test sonucunda elde edilen bağımsızlık derecesinin yüksekliği, ilgili özniteliğin hedef öznitelik ile arasındaki ilişkinin fazlalığına işaret etmektedir. Bu nedenle bağımlılık derecesi en yüksek olan öznitelikler seçilerek sınıflandırma algoritmalarının başarısının arttırılması amaçlanmaktadır.

$$X^2(t,c) = \frac{N.(AD-CB)^2}{(A+D).(B+D).(A+B).(C+D)} \tag{3.6}$$

Denklem 3.6'da bulunan;

$X^2(t,c) = t$ teriminin c sınıfıyla arasındaki Ki-kare skorunu,

N = Toplam doküman sayısını,

$A = t$ terimini içeren ve c sınıfında olan doküman sayısını,

$B = t$ terimini içeren ancak c sınıfında olmayan doküman sayısını,

$C = t$ terimini içermeyen ancak c sınıfında olan doküman sayısını,

$D = t$ terimini içermeyen ancak c sınıfında olmayan doküman sayısını ifade etmektedir. (Yang ve Pedersen, 1997)

3.4.2. Korelasyon matrisi

Korelasyon Matrisi, iki veya daha fazla öznitelik arasındaki ilişkinin şiddetini ve yönünü hesaplamak için kullanılan istatistiksel bir yöntemdir. Korelasyon Matrisine girdi olarak verilen bütün öznitelikler matrisin satır ve sütunlarına yerleştirilir. Daha sonra her bir öznitelik çifti için korelasyon değeri hesaplanır ve sonuç ilişkinin yönünü belirleyen işaret ile birlikte matrisin ilgili sütununa kaydedilir. Denklem 3.7 ile hesaplanan korelasyon katsayısı -1 ile +1 arasında değişmektedir. Burada -1 değeri öznitelik çifti arasında negatif yönde güçlü bir ilişki olduğunu, +1 değeri söz konusu öznitelik çifti arasında pozitif yönde güçlü bir ilişki olduğunu ve son olarak 0 değeri ise iki öznitelik arasında herhangi bir ilişki olmadığını ifade etmektedir. Ayrıca elde edilen hesaplama sonuçlarına göre korelasyon matrisinin hücreleri renklendirilmektedir. Renklendirmede genellikle koyu kırmızı renk pozitif yönde mükemmel korelasyonu, mavi renk negatif yönde mükemmel korelasyonu ve gri renk korelasyon olmadığını ifade etmektedir. Gösterimin tablo şeklinde olması okumayı, yorumlamayı ve anlamayı kolaylaştırdığından bu yöntem öznitelik seçimi aşamasında sıklıkla tercih edilmektedir.

$$r = \frac{n\,(\Sigma xy) - \Sigma x\,\Sigma y}{\sqrt{(n(\Sigma x^2 - (\Sigma x)^2))\;(n\,(\Sigma y^2 - (\Sigma y)^2))}} \tag{3.7}$$

Denklem 3.7'de yer alan;

r = Korelasyon katsayısını,

n = Toplam gözlem sayısını,

x = Birinci özniteliği,

y = İkinci özniteliği ifade etmektedir. (Bland ve Altman, 1995)

3.5. Sınıflandırma Algoritmaları

Makine öğrenmesi sınıflandırma algoritmaları kullanılarak veri setinin önceden belirlenmiş kategorilere göre bölümlenmesi sınıflandırma olarak tanımlanmaktadır.

Sınıflandırma işleminin sağlıklı bir biçimde gerçekleştirilebilmesi için veri setinin yukarıdaki bölümlerde anlatılan işlem adımlarından geçirilmesi gerekmektedir. Metin tabanlı sınıflandırma uygulamalarında sınıflandırma algoritmalarının veri seti hakkında bilgi sahibi olabilmesi ve sınıflandırmayı en doğru şekilde yapabilmesi amacıyla, veri setinin alt kümesi niteliğindeki bir parçası üzerinde eğitilmesi gerekmektedir. Böylelikle sınıflandırma algoritmaları veri setinin kalan bölümünü tahmin ederken daha isabetli kararlar verecektir.

Bu çalışmada; Gaussian Naive Bayes, Multinomial Naive Bayes, Karar Ağaçları, KNN, Lojistik Regresyon, SVM, Çok Katmanlı Algılayıcılar (Multi-layer Perceptron - MLP) ve Rastgele Orman sınıflandırma algoritmaları kullanılmıştır.

3.5.1. Naive bayes

Temeli Bayes teoremine dayanan Naive Bayes sınıflandırma algoritması veri madenciliği, duygu analizi ve metin sınıflandırma alanlarında oldukça başarılı bir algoritmadır. Naive Bayes bütün özniteliklerin aynı önem derecesine sahip ve özniteliklerin birbirinden bağımsız olduğunu kabul ederek işlemleri gerçekleştirir. (Kim ve ark., 2006) Sınıflandırmada kullanılan her öznitelik için olası bütün durumlar hesaplanır ve elde edilen en yüksek olasılık değerine göre ilgili öznitelik kümesinin sınıflandırma işlemi gerçekleştirir. Bu sayede küçük boyutlu veri kümelerinde bile yüksek oranda başarı sağlanabilmektedir.

Naive Bayes algoritmasının sınıflandırma için kullanmış olduğu Bayes'in olasılık teoremi Denklem 3.8'de verilmiştir. (Alkış, 2016)

$$P(C_k|X) = \frac{P(X|C_k)P(C_k)}{P(X)} \tag{3.8}$$

Denklemde yer alan;

$P(X|C_k) = k$ duygu sınıfına ait bir örneğin X olma ihtimalini,

$P(C_k) = k$ sınıfının ilk olasılığını,

$P(X)$ = Seçilen bir örneğin X olma olasılığını,

$P(C_k|X) = X$ duygu sınıfına ait bir örneğin k olma ihtimalini ifade etmektedir.

Naive Bayes sınıflandırma algoritması Bernoulli Naive Bayes, Multinominal Naive Bayes ve Gaussian Naive Bayes olmak üzere üç ana başlık altında incelenmektedir.

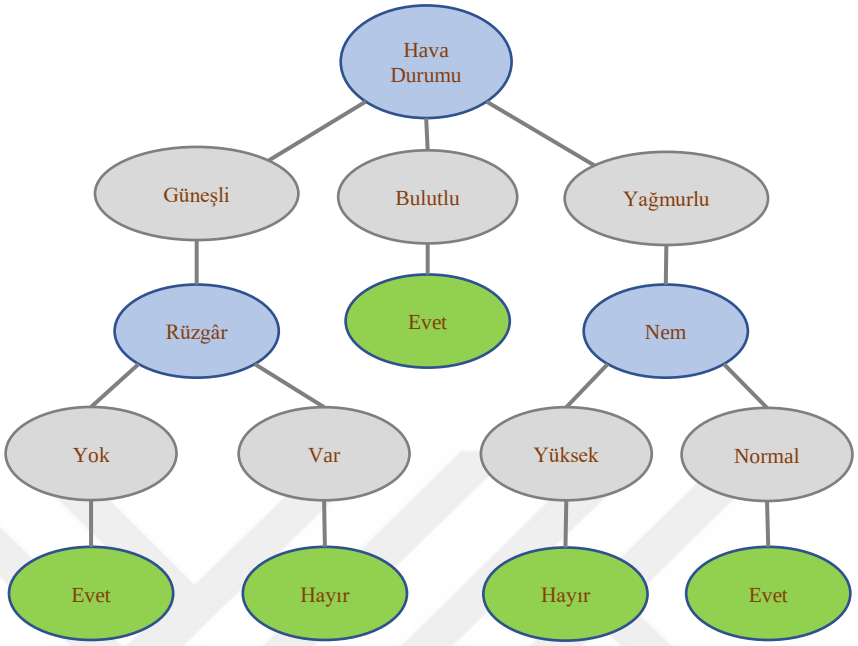
3.5.2. Karar ağaçları

Karar Ağaçları, çok sayıda gözleme sahip olan bir veri setini kümeleme algoritmaları yardımıyla daha küçük kümelere bölerek benzer kayıtların bir arada gruplandırılmasını sağlayan bir sınıflandırma algoritmasıdır. Karar Ağaçları hem regresyon hem de sınıflandırma problemlerinin çözümünde kullanılabildiği için oldukça popülerdir. Bunun yanı sıra Karar Ağaçları sayısal ve kategorik veriler üzerinde de çalışabilmektedir. Karar Ağaçlarında problemin çözümü için yöntemin adından da anlaşılabileceği üzere ağaç yapıları kullanılmaktadır. Bu yapı çok sayıda karar düğümünden ve yaprak düğümlerden meydana gelmektedir. Bir karar düğümü bir veya daha fazla dallanma ile farklı alt kümeleri oluşturabilmektedir. Bu bağlamda en yaygın olarak kullanılan Karar Ağacı türü İkili Karar Ağacıdır. Karar Ağacının en tepesinde yer alan ilk düğüm kök olarak adlandırılırken, ağacın en altında kalan düğümler yaprak olarak ifade edilmektedir. Karar Ağaçlarının anlatımında sıklıkla başvurulan örnek veri seti Çizelge 3.9'da verilmiştir. Söz konusu örnek çizelgede değişik hava koşullarında futbol oynanıp oynanamayacağının durum değerlendirilmesi yapılarak karar verilmesi sağlanmıştır.

Çizelge 3.9. Karar ağaçları örnek veri seti (Dalkılıç ve Dalkılıç, 2015)

Özellikler				Hedef
Hava Durumu	Sıcaklık	Nem	Rüzgâr	Futbol Oyna
Yağmurlu	Sıcak	Yüksek	Yok	Hayır
Yağmurlu	Sıcak	Yüksek	Var	Hayır
Bulutlu	Sıcak	Yüksek	Yok	Evet
Güneşli	Ilık	Yüksek	Yok	Evet
Güneşli	Soğuk	Normal	Yok	Evet
Güneşli	Soğuk	Normal	Var	Hayır
Bulutlu	Soğuk	Normal	Var	Evet
Yağmurlu	Ilık	Yüksek	Yok	Hayır
Yağmurlu	Soğuk	Normal	Yok	Evet
Güneşli	Ilık	Normal	Yok	Evet
Yağmurlu	Ilık	Normal	Yok	Evet
Bulutlu	Ilık	Yüksek	Var	Evet
Bulutlu	Sıcak	Normal	Yok	Evet
Güneşli	Ilık	Yüksek	Var	Hayır

Çizelge 3.9'da yer alan veri setinin örnek karar ağaçları gösterim şeması Şekil 3.5'te sunulmuştur.



Şekil 3.5. Çizelge 3.9'da yer alan veri setinin karar ağaçlarıyla gösterimi (Dalkılıç ve Dalkılıç, 2015)

Entropiye ve regresyona dayalı farklı sınıflandırma ağaçları bulunmaktadır. ID3 ve C4.5 sınıflandırma ağaçları entropiye, CART ise regresyona dayalı Karar Ağaçlarından en çok bilinenlerdir. Entropi bir sistemdeki düzensizliğin ölçüsünü temsil etmekte olup Denklem 3.9'da bulunan formülle veri setinde yer alan bütün değerler için hesaplanmaktadır. Bir olayın gerçekleşme olasılığına göre veri kümesinin homojenliğini hesaplamaya çalışan entropi 0 ile 1 arasında bir değer almaktadır. Veri kümesinin homojenliği arttıkça entropi değeri de sıfıra yaklaşmaktadır. Bu çalışmada entropiye dayalı Karar Ağacı kullanılmıştır.

$$H(S) = -\sum_{i=1}^{n} P_i \log_2 P_i \tag{3.9}$$

Denklemde yer alan;

S = Hedef değişkenin entropisini,

n = Veri setinde bulunan gözlem sayısını,

P = Gereken şartı ifade etmektedir. (Maszczyk ve Duch, 2008)

3.5.3. K-en yakın komşu (K-nearest neighbors-KNN)

KNN hem sınıflandırma hem de regresyon problemlerinin çözümünde kullanılabilen gözetimli öğrenme algoritmasıdır. Algoritmanın temelini birbirine yakın olan örneklemlerin benzer olduğu varsayımı oluşturmaktadır. Bu nedenle KNN algoritmasında verilerin birbirine olan mesafesi ve komşu gözlem sayısı büyük önem taşımaktadır. KNN algoritması eğitim veri setinden edinmiş olduğu bilgiler doğrultusunda test veri setini sınıflandırmayı hedeflemektedir. (Taşcı ve Onan, 2016) Bu kapsamda sınıflandırma aşamasında takip edilen yol Şekil 3.6'da gösterilmiştir.

Şekil 3.6. KNN algoritması sınıflandırma işlem adımları

K parametresi kullanıcı tarafından belirlenmekte olup bu parametre, kategorisi belli olmayan gözlem verisine en yakın komşu sayısını ifade etmektedir. Algoritma kategorisi bilinmeyen test verisini sınıflandırabilmek için gözleme en yakın olan k sayısı kadar komşu eğitim verisinin kategorisini göz önünde bulundurarak sınıflandırmayı gerçekleştirir.

Komşuların test verisine olan mesafesi üç farklı uzaklık hesaplama yöntemi ile yapılabilmektedir. Bunlar; Euclidean, Manhattan ve Minkowski uzaklık hesaplama yöntemleridir. Söz konusu yöntemlere ait denklemlere aşağıda (Denklem 3.10, 3.11 ve 3.12) yer verilmiştir. (Dilki ve Başar, 2020)

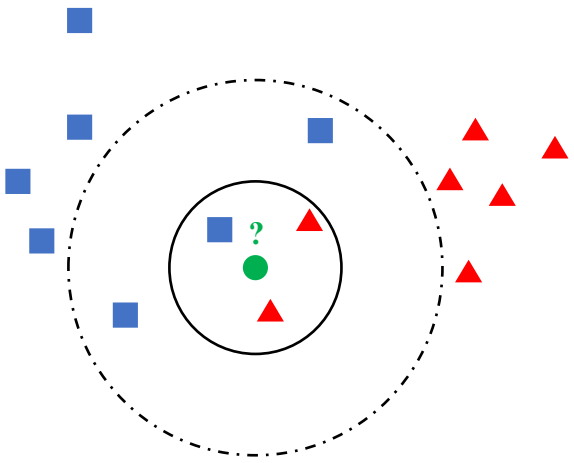
$$Euclidean = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \tag{3.10}$$

$$Manhattan = \sum_{i=1}^{k} |x_i - y_i| \tag{3.11}$$

$$Minkowski = \left(\sum_{i=1}^{k} (|x_i - y_i|)^q \right)^{1/q} \tag{3.12}$$

KNN sınıflandırma yönteminde k parametresi sınıflandırma başarısını doğrudan etkilemektedir. Bu nedenle k parametresinin optimum değer olarak belirlenmesi büyük önem arz etmektedir. Şekil 3.7'de k parametresinin sınıflandırmaya olan etkisi bir örnek üzerinden gösterilmeye çalışılmıştır. Ayrıca parametrenin belirlenmesinde aşağıda yer alan hususlarında dikkate alınması sınıflandırma başarısını etkileyecektir.

1. K parametresinin tek sayı olarak belirlenmesi test verisine en yakın farklı kategoriden komşu gözlemlerin sayısının eşit olması ihtimalini ortadan kaldıracak ve algoritmanın daha stabil çalışmasını sağlayacaktır.
2. K parametresinin 1 olarak seçilmesi modelin aşırı öğrenme ihtimalini arttıracaktır.
3. Değerin çok büyük seçilmesi durumunda model genel sonuçlar üretecek ve sınıflandırma başarısı düşecektir.

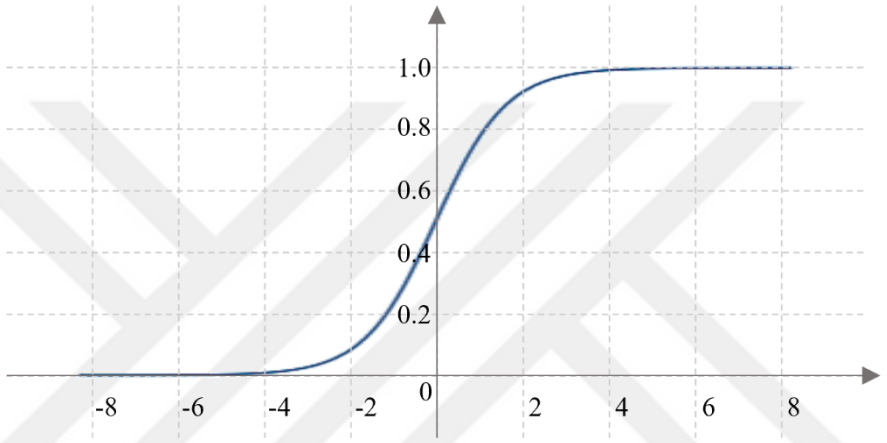


Şekil 3.7. K parametresinin önemi (Peterson, 2009), (Altunkaynak ve ark., 2020)

Şekil 3.7'de görüldüğü üzere; yeşil yuvarlak nokta k parametresinin 3 olarak belirlenmesi durumunda kırmızı üçgen olarak tahmin edilirken, parametrenin 5 olarak belirlenmesi durumunda mavi kare olarak tahmin edilecektir.

3.5.4. Lojistik regresyon (Logistic regression)

Makine öğrenmesi algoritmalarından olan Lojistik Regresyon, bağımlı ve bağımsız değişkenler arasındaki nedensellik ilişkisini inceleyerek bir model oluşturmaktadır. Oluşturulan model bir olayın gerçekleşme olasılığını tahmin etmekte veya kategorik verilerin sınıflandırılmasında kullanılmaktadır. Bunu gerçekleştirebilmek için Şekil 3.8'de yer alan Sigmoid fonksiyonundan yararlanılmaktadır.



Şekil 3.8. Sigmoid fonksiyonu (Han ve Moraga, 1995)

Yukarıda grafiği bulunan Sigmoid fonksiyonu Denklem 3.13'te paylaşılmıştır. Formül incelendiğinde Lojistik Regresyonun her zaman 0 ile 1 arasında bir değer alacağı görülecektir. Şekil 3.8'den görülebileceği üzere bağımlı değişkeni sembolize eden x değeri artı sonsuza yakınsadıkça hedef değişkeni sembolize eden y değeri bire yaklaşmaktadır. Benzer şekilde x değerinin eksi sonsuza yakınsaması durumunda y değerinin de sıfıra yaklaştığı görülmektedir.

Hesaplanan Lojistik Regresyon değerinin kullanıcı tarafından belirlenen eşik değerinin (Genellikle 0.5 seçilmektedir.) altında kalması durumunda sonuç 0 olarak kabul edilmekte ve ilgili verinin o sınıfa ait olmadığı varsayılmaktadır. Aksi durumda hesaplanan Lojistik Regresyon değeri eşik değerinin üzerinde olacağından ilgili verinin mevcut sınıfa ait olduğuna karar verilecektir.

$$sig(t) = \frac{1}{1 + e^{-t}} \tag{3.13}$$

Denklemde yer alan;

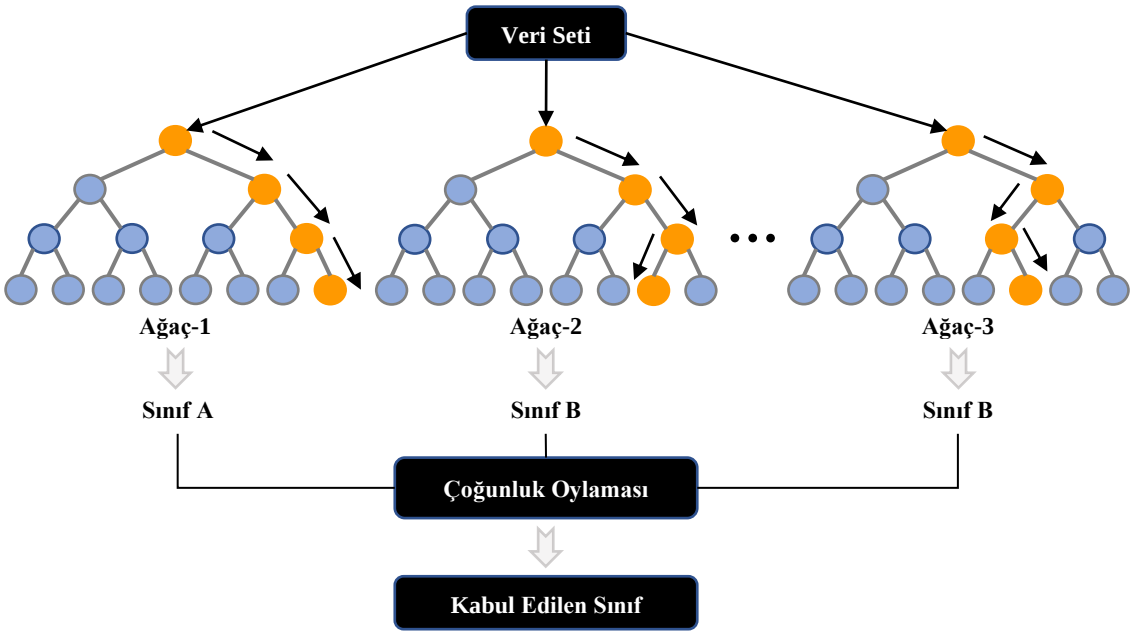
$sig(t)$ = Hedef değişkenin lojistik regresyon değerini,

t = Bağımlı değişkene ait veriyi,

e = Euler sayısını (2.71) ifade etmektedir. (Han ve Moraga, 1995)

3.5.5. Rastgele orman algoritması (Random forest)

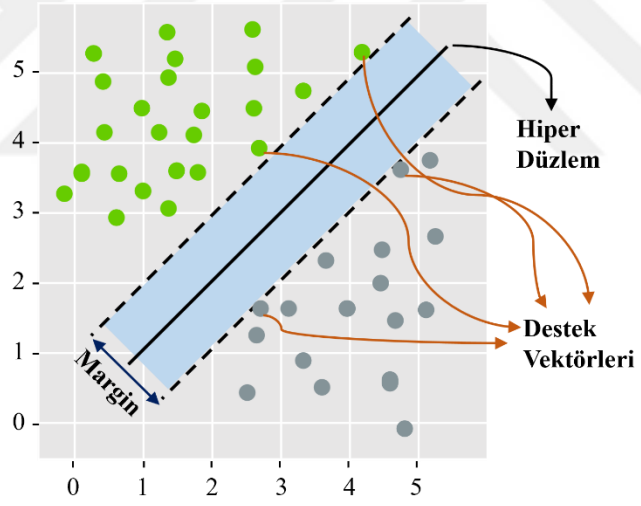
Rastgele Orman, regresyon ve sınıflandırma problemlerinin çözümünde kullanılabilen denetimli bir makine öğrenmesi algoritmasıdır. Algoritma temelde kullanmış olduğu yöntem itibariyle Karar Ağaçlarıyla benzerlik göstermektedir. Ancak Rastgele Orman algoritması problemin çözümünde birden fazla ağaç kullanması, kök düğümlerinin bulunması ve düğümlerin bölünmesi işlemlerini rastgele yapması nedeniyle Karar Ağaçlarından farklılık göstermektedir. Şekil 3.9'dan da görülebileceği üzere algoritma sınıflandırma yaparken birbirinden bağımsız olarak çalışan birçok karar ağacından aldığı veriyi birleştirerek elde edilen en yüksek skora göre sınıflandırma işlemini gerçekleştirmektedir. Bu yöntem sayesinde Rastgele Orman algoritmasının çalışma hızı, öğrenmeye karşı dayanıklılığı ve başarı oranı artmaktadır. Algoritmanın başlatılabilmesi için kullanıcı tarafından iki adet parametrenin belirlenmesi gerekmektedir. Bunlar; oluşturulacak karar ağaçlarının sayısı ve her bir düğümde yer alacak özniteliklerin sayısı olarak tanımlanmaktadır. (Akar ve Güngör, 2012)



Şekil 3.9. Rastgele orman algoritması çalışma biçimi (Bonissone ve ark., 2010)

3.5.6. Destek vektör makineleri (Support vector machine - SVM)

SVM algoritması hem sınıflandırma hem de regresyon problemlerinin çözümünde kullanılabilen gözetimli bir makine öğrenmesi algoritmasıdır. Ancak çoğunlukla sınıflandırma problemlerinin çözümünde kullanılmaktadır. Algoritma belirli öznitelikler etrafında kümelenmiş verileri birbirinden ayırmak ve sınıflandırma yapabilmek için vektörlerden faydalanmaktadır. Sınıfları birbirinden ayıran bu vektörler hiper düzlem olarak nitelendirilmektedir. SVM, sınıflandırmanın başarısını arttırabilmek amacıyla bütün sınıfların verilerine en uzak mesafede olan vektörü tercih etmektedir. Tercih edilen bu vektöre optimum hiper düzlem adı verilmektedir. Optimum hiper düzleme en yakın olan noktalara veya verilere ise destek (Support) vektörü denilmektedir. Destekler arasında kalan boşluk ise margin olarak adlandırılmaktadır. Şekil 3.10'da SVM algoritmasında kullanılan terimler ve algoritmanın çalışma biçimi yer almaktadır.



Şekil 3.10. SVM algoritmasında kullanılan terimler ve algoritmanın çalışma biçimi (Kavzoğlu ve Çölkesen, 2010), (Rueping, 2010)

Şekil 3.10'da görüldüğü üzere noktalar arasındaki fark oldukça belirgindir. Bu ve buna benzer durumlarda algoritmanın çekirdek parametresinin "linear" olarak seçilmesi sınıflandırma başarısının artmasını sağlayacaktır. Yukarıdaki şekilde olduğu gibi doğrusal olarak ayrılabilen iki sınıflı bir sınıflandırma probleminde optimum hiper düzleme ait eşitsizlikler Denklem 3.14 ve 3.15 kullanılarak hesaplanmaktadır. (Kavzoğlu ve Çölkesen, 2010)

$w.x_i + b \geq +1$ her $y = +1$ için (3.14)

$w.x_i + b \leq +1$ her $y = -1$ için (3.15)

Denklem 3.14 ve 3.15'te bulunan;

x_i = R^N elemanı olup N-boyutlu bir uzayı,

$y = \{-1, +1\}$ elemanı olup sınıf etiketlerini,

w = Ağırlık vektörünü (Hiper düzlem normalini),

b = Eğilim değerini ifade etmektedir. (Osuna ve ark., 1997)

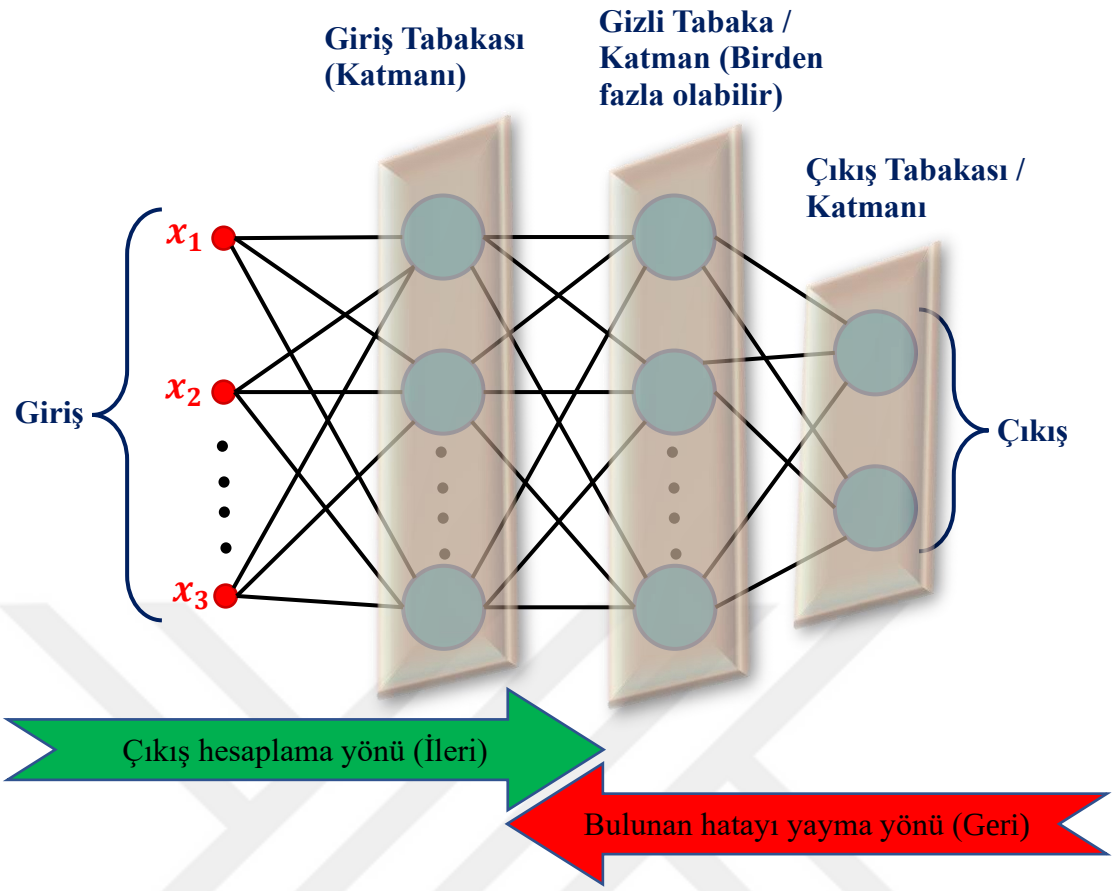
Her veri setinde bulunan değerler yukarıdaki durumda olduğu gibi doğrusal bir vektör ile birbirinden ayrılmayabilir. Bu gibi durumlarda çekirdek tipi en uygun seçenek ile değiştirilmelidir. Çekirdek parametresi; "linear", "poly", "rbf", "sigmoid" ve "precomputed" değerlerini alabilmektedir. SVM algoritmasında kullanılan temel çekirdek fonksiyonları ve parametreleri Çizelge 3.10'da sunulmuştur.

Çizelge 3.10. SVM'de kullanılan temel çekirdek fonksiyonları ve parametreleri (Das ve ark., 2019)

Çekirdek Fonksiyonu	Formüller	Parametreler
Polinom Kerneli	$K(x,y) = \left((x.y)+1\right)^d$	d: Polinom derecesi
Normalleştirilmiş Polinom Kerneli	$K(x,y) = \frac{\left((x.y)+1\right)^d}{\sqrt{\left((x.y)+1\right)^d \left((y.y)+1\right)^d}}$	d: Polinom derecesi
Radyal Tabanlı Fonksiyon (RBF) Kerneli	$K(x,y) = e^{-\gamma\|x-x_i\|^2}$	γ : Kernel Boyutu
Pearson Evrensel Kernel (PUK)	$K(x,y) = \frac{1}{\left[1+\left(\frac{2.\sqrt{\|x-y\|^2\sqrt{2^{1/\omega}-1}}}{\sigma}\right)^2\right]^\omega}$	ω ,σ : Pearson genişliği parametreleri

3.5.7. Çok katmanlı algılayıcılar (Multi-layer perceptron-MLP)

İngilizcede Multi-layer Perceptron olarak ifade edilen Çok Katmanlı Algılayıcılar Şekil 3.11'de gösterildiği gibi 3 temel bölümden oluşmaktadır. Bunlar sırasıyla: Giriş katmanı, gizli katman ve çıkış katmanı olarak ifade edilmektedir.



Şekil 3.11. MLP modeli (İşeri ve Arıman, 2019)

Giriş katmanında yer alan nöronların sayısı girdi olarak alınan vektörün uzunluğuna eşittir. Çıkış katmanında yer alan nöronların sayısı modelin sonucunda elde edilmek istenen vektörün uzunluğuna eşit olacak şekilde tasarlanmaktadır. Giriş ile çıkış katmanı arasında bulunan ve en az bir katmandan oluşması gereken ara katmanlar ise gizli katman olarak nitelendirilmektedir. Gizli katmanda en az bir nöron bulunması gerekmektedir ancak, bir nöron ile model istenen oranda başarı sağlamayabilir. Bu nedenle; modelin gizli katmanında bulunması gereken optimum nöron sayısını belirlemek için farklı denemeler yapılmaktadır. Denemeler sonucunda en yüksek başarı oranını sağlayan gizli katman nöron sayısı tasarım modele uygulanmaktadır. Yani gizli katman (ara katman) sayısı ve ara katmanlardaki nöron sayısı deneme yanılma yolu ile tespit edilmektedir. (Ataseven, 2013)

Denklem 3.16'da yer alan eşitlik, MLP ağ modelini ifade etmektedir. (İşeri ve Arıman, 2019)

$$Y_n = f_0\left\{b_0 + \sum_{k=1}^{h}\left[w_k * f_h\left(b_{hk} + \sum_{i=1}^{m} w_{ik}X_{ni}\right)\right]\right\} \quad (3.16)$$

Denklem 3.16'da bulunan;

Y_n : Normalize edilmiş çıktıyı,

f_0 : Çıkış katman transfer fonksiyonunu,

b_0 : Çıkış katmanının bias terimlerini,

w_k : K'nıncı gizli katmandan çıkış katmanına olan bağlantı ağırlığını,

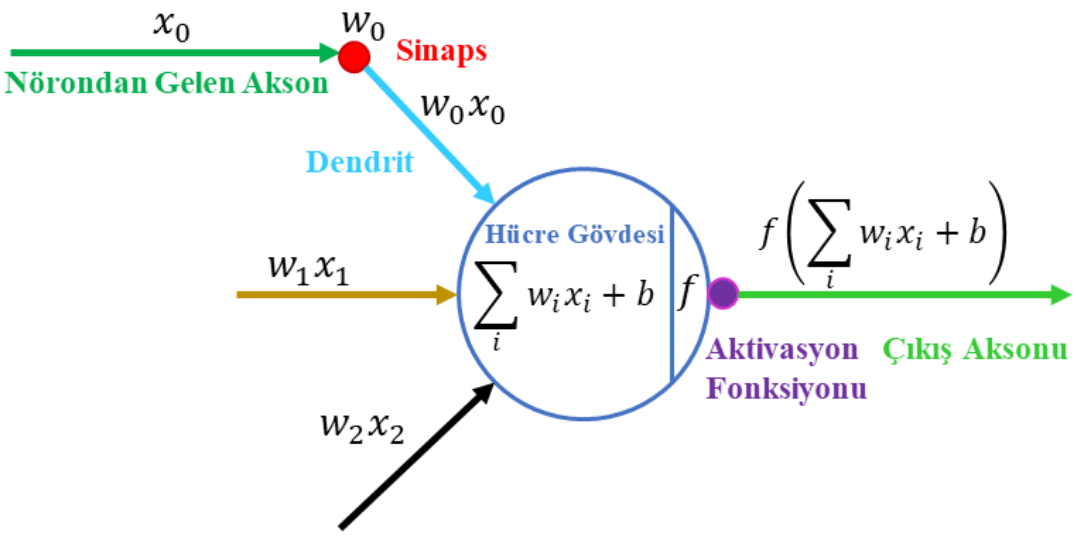
f_h : Gizli katmanın transfer fonksiyonunu,

b_{hk} : K'nıncı gizli katmanın bias terimlerini,

w_{ik} : Girdi katmanındaki i'ninci nörondan gizli katmandaki k'nıncı nöron bağlantısını,

X_{ni} : Normalize edilmiş girdi vektörünü ifade etmektedir.

Her katmanda bulunan nöron, kendisinden bir sonra gelen katmandaki bütün nöronlarla tek yönlü ve ileri beslemeli olacak şekilde bağlantılı olup sistemde geri besleme yoktur. Bu nedenle bu tip ağlar İleri Beslemeli Sinir Ağı modeli olarak ifade edilmektedir. MLP sınıflandırma algoritmalarında kullanılan yapay sinir hücresinin (Nöron) yapısı Şekil 3.12'de verilmiştir. Şekil 3.11'den de görülebileceği üzere modelde yer alan her katmanın çıkışı kendisinden bir sonraki katmanın girişini oluşturacak şekilde tasarlanmıştır.



Şekil 3.12. Yapay sinir hücresinin yapısı (Ataseven, 2013)

3.6. Başarı Arttırma ve Gösterim Yöntemleri

Makine öğrenmesi algoritmalarından birçok farklı alanda faydalanılmaktadır. Bu kapsamda yürütülen çalışmalarda algoritmaların maksimum performansla çalışabilmesini sağlamak için parametre ayarlamasına ihtiyaç duyulmaktadır. Parametrelerin sayısı ve kullanım alanları algoritmalara göre değişiklik gösterebilmektedir. Bununla birlikte yapılan çalışmanın türüne ve kullanılan veri setinin teknik özelliklerine bağlı olarak birçok parametrenin ayarlanmasına ihtiyaç duyulabilmektedir. Bu gibi durumlarda olası parametre kombinasyonu sayısının çok fazla olması sebebiyle optimum parametre setinin manuel olarak bulunabilme ihtimali oldukça düşüktür. Bu nedenle, algoritmalar için optimum parametrelerin bulunmasını sağlayan yöntemler geliştirilmiştir. Söz konusu yöntemler ile hiper parametre analizi yapılarak algoritmalar için en iyi parametre kombinasyonunun otomatik olarak bulunması sağlanmaktadır. Bu çalışmada hiper parametre analizi için sıkça tercih edilen RandomizedSearchCV ve GridSearchCV yöntemlerinden yararlanılmıştır.

3.6.1. GridSearchCV

GridSearchCV yöntemi, tanımlanan her bir parametrenin diğer parametrelerle oluşturabileceği bütün kombinasyon çeşitlerini algoritma üzerinde ayrı ayrı deneyerek en yüksek başarı oranının bulunmasını amaçlamaktadır. GridSearchCV yöntemi bütün kombinasyon olasılıklarını teker teker denediği için en yüksek başarı oranını sağlayacak hiper parametre setinin bulunmasını garanti etmektedir. Ayrıca yöntem hiper parametre ayarlaması ile birlikte kullanılan veri setinden maksimum bilgi çıkarımını sağlamak ve dolayısıyla en yüksek verimi elde etmek amacıyla çapraz doğrulama yöntemini de kullanmaktadır.

GridSearchCV yönteminin çalışma sistematiği bütün kombinasyonların teker teker denenmesine dayandığı için küçük veri setleri üzerinde gayet kullanışlı ve başarılıdır. Ancak bu durum büyük boyutlu veri setleri için aynı olmayabilir. Çünkü, veri seti çaprazlama sayısı ile olası bütün parametre kombinasyonlarının çarpımı sonucunda elde edilecek eğitilmesi gereken algoritma sayısı göz önünde bulundurulduğunda işlemin tamamlanma süresinin çok fazla olacağı görülecektir. Bu nedenle GridSearchCV yöntemi parametre kombinasyonu sayısının fazla ve veri seti boyutunun yüksek olduğu durumlarda pek fazla tercih edilmemektedir. Bu gibi durumlarda genellikle

GridSearchCV yöntemine kıyasla daha hızlı olan RandomizedSearchCV yönteminden faydalanılmaktadır. Böylece hem hiper parametre analizi yapılarak algoritmanın sınıflandırma başarısı arttırılmakta hem de bu aşamada harcanan süre büyük ölçüde düşürülerek zaman maliyeti sağlanmaktadır.

3.6.2. RandomizedSearchCV

RandomizedSearchCV yöntemi, tıpkı GridSearchCV yönteminde olduğu gibi algoritmalar için en doğru hiper parametre setinin belirlenmesinde kullanılmaktadır. Ancak RandomizedSearchCV yöntemi GridSearchCV yönteminden farklı olarak bütün parametre kombinasyonlarını veri seti üzerinde denemek yerine, rastgele seçtiği belirli sayıda parametre kombinasyonunu veri seti üzerinde test etmektedir. Bu nedenle GridSearchCV yönteminde olduğu gibi en iyi performansı sağlayan hiper parametre setini bulmayı garanti etmez.

RandomizedSearchCV yönteminde en başarılı hiper parametre setinin bulunması için yöntemin kaç defa deneme yapacağı hususu iterasyon sayısı ile belirlenmektedir. İterasyon sayısı kullanıcı tarafından belirlenmekte olup RandomizedSearchCV yöntemi ve hiper parametre analizi için büyük önem taşımaktadır. Optimum hiper parametre analizi yapılan algoritmanın veri seti üzerindeki eğitim sayısı, çaprazlama sayısı ile iterasyon sayısının çarpımı sonucunda ortaya çıkmaktadır. İterasyon sayısının tercih edilen sınıflandırma algoritmasının çalışma biçimi ve veri setinin teknik özellikleri dikkate alınarak kullanıcı tarafından belirlenmesi aslında zaman maliyetinin de kullanıcı tarafından belirlendiği anlamını taşımaktadır. Bu nedenle büyük boyutlu veri setleri üzerinde RandomizedSearchCV yöntemi GridSearchCV yöntemine göre hem daha esnek hem de daha avantajlıdır.

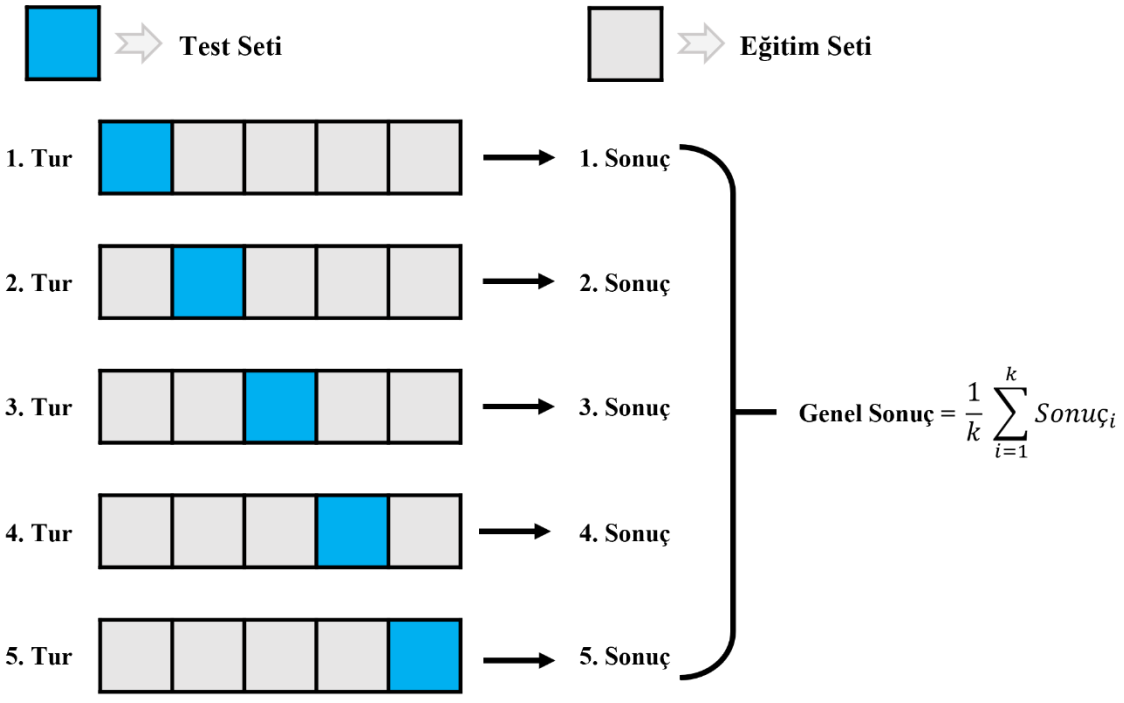
3.6.3. Çapraz doğrulama (Cross validation)

Sınıflandırma algoritmalarında veri seti eğitim ve test verisi olmak üzere ikiye bölünmektedir. Bölme işlemi her ne kadar rastgele yapılıyor olsa da veri seti yeterince homojen karışmamış olabilir. Bu durum algoritmaların daha önce eğitim verisinde hiç karşılaşmadığı bir veriyi yanlış sınıflandırmasına neden olacaktır. Bu nedenle; bu gibi istenmeyen durumların oluşmasını engellemek için çapraz doğrulama yöntemlerinden

faydalanılmaktadır. Ayrıca yukarıdaki bölümlerde de bahsedildiği üzere çapraz doğrulama hiper parametre analizlerinde sıklıkla kullanılan bir yöntemdir.

Literatürde çapraz doğrulama için birçok farklı teknik mevcuttur. Rastgele Çapraz Doğrulama, Zaman Serisi Çapraz Doğrulama, K-katlı Çapraz Doğrulama, Birini Dışarıda Bırakarak Çapraz Doğrulama, Basit Çapraz Doğrulama ve Tekrarlanan Çapraz Doğrulama yöntemleri bunlardan bazılarıdır. Bu tez çalışmasında çapraz doğrulama yöntemleri arasında oldukça popüler olan K-katlı Çapraz Doğrulama yöntemi kullanılmıştır.

K-katlı Çapraz Doğrulama yönteminin temeli veri setinin belirli parçalara bölünerek modelin her bir parça üzerinde eğitim ve değerlendirme yapmasına dayanmaktadır. Bunun nedenle veri seti ilk olarak kendi içerisinde daha homojen bir dağılımın yakalanabilmesi amacıyla karıştırılır. Daha sonra veri seti kullanıcı tarafından belirlenen k sayısı kadar eşit parçaya bölünerek her parçanın hem eğitim hem de test verisi olarak kullanılması sağlanır. Böylelikle veri setinin dağılımından kaynaklanan sapmalar minimuma indirgenmiş olur. Ayrıca yöntem algoritmanın aşırı öğrenmesini tespit ederek modelin gerçek başarısının belirlenmesine de yardımcı olmaktadır. Ancak bu yöntem modelin başarı oranının arttırılmasında avantaj sağlarken, veri setinin büyüklüğüne bağlı olarak hesaplama süresi ve zaman maliyeti yönünden dezavantaja dönüşebilmektedir.



Şekil 3.13. K = 5 için K-katlı çapraz doğrulama yöntemi (Vergili ve Orhan, 2022)

K-katlı Çapraz Doğrulama yönteminin çalışma biçimi Şekil 3.13'te k = 5 değeri için gösterilmiştir. Şekilden de anlaşılabileceği üzere veri seti k adet parçaya bölünmüş ve her bir parçanın hem test hem de eğitim verisi olarak kullanılması sağlanmıştır. Modelin genel sonucunun belirlenebilmesi için her turun sonunda elde edilen başarı oranları toplanarak k sayısına bölünmektedir. Bunun için Denklem 3.17'den yararlanılmaktadır. (Namlı ve ark., 2019)

$$Genel\ Sonuç = \frac{1}{k}\sum_{i=1}^{k} Sonuç_i \tag{3.17}$$

Denklemde bulunan;

k : Kullanıcı tarafından belirlenen bölünme sayısını,

$Sonuç_i$: Tur sonucunda elde edilen başarı oranını ifade etmektedir.

3.6.4. Karmaşıklık matrisi (Confusion matrix)

Karmaşıklık Matrisi, makine öğrenmesi sınıflandırma algoritmalarının yapmış olduğu tahminlerin gerçek değerler ile karşılaştırarak sonuçların Şekil 3.14'te görüldüğü gibi bir matris üzerinde ifade edildiği yöntemdir. Karmaşıklık Matrisi sayesinde modelin hangi veriler üzerinde sıklıkla hata yaptığı gözlemlenebilmekte ve modele ilişkin daha detaylı bilgi sahibi olunabilmektedir. Karmaşıklık Matrisi oluşturulurken dört farklı parametreden faydalanılmaktadır. Bunlar sırasıyla gerçek pozitif, gerçek negatif, yanlış pozitif ve yanlış negatif değerleridir.

		Tahmin	
		Negatif	Pozitif
Gerçek	Negatif	TN	FP
	Pozitif	FN	TP

Şekil 3.14. Karmaşıklık matrisi gösterimi (Visa ve ark., 2011)

Gerçek Pozitif (True Positive — TP): Gerçekte doğru olan bir durumun algoritma tarafından da doğru olarak tahmin edilmesini,

Gerçek Negatif (True Negative — TN): Gerçekte yanlış olan bir durumun algoritma tarafından da yanlış olarak tahmin edilmesini,

Yanlış Pozitif (False Positive — FP): Gerçekte yanlış olan bir durumun algoritma tarafından doğru olarak tahmin edilmesini,

Yanlış Negatif (False Negative — FN): Gerçekte doğru olan bir durumun algoritma tarafından yanlış olarak tahmin edilmesini ifade etmektedir.

Yukarıda bulunan değerler Denklem 3.18, 3.19 ve 3.20'de yerine yazılarak sınıflandırma algoritmalarına ait Doğruluk, Kesinlik ve Hassasiyet gibi performans ölçüm metrikleri hesaplanmaktadır. (Bilgin, 2017)

$$Doğruluk = \frac{(TP + TN)}{(TP + TN + FN + FP)} \tag{3.18}$$

$$Kesinlik = \frac{TP}{(TP + FP)} \tag{3.19}$$

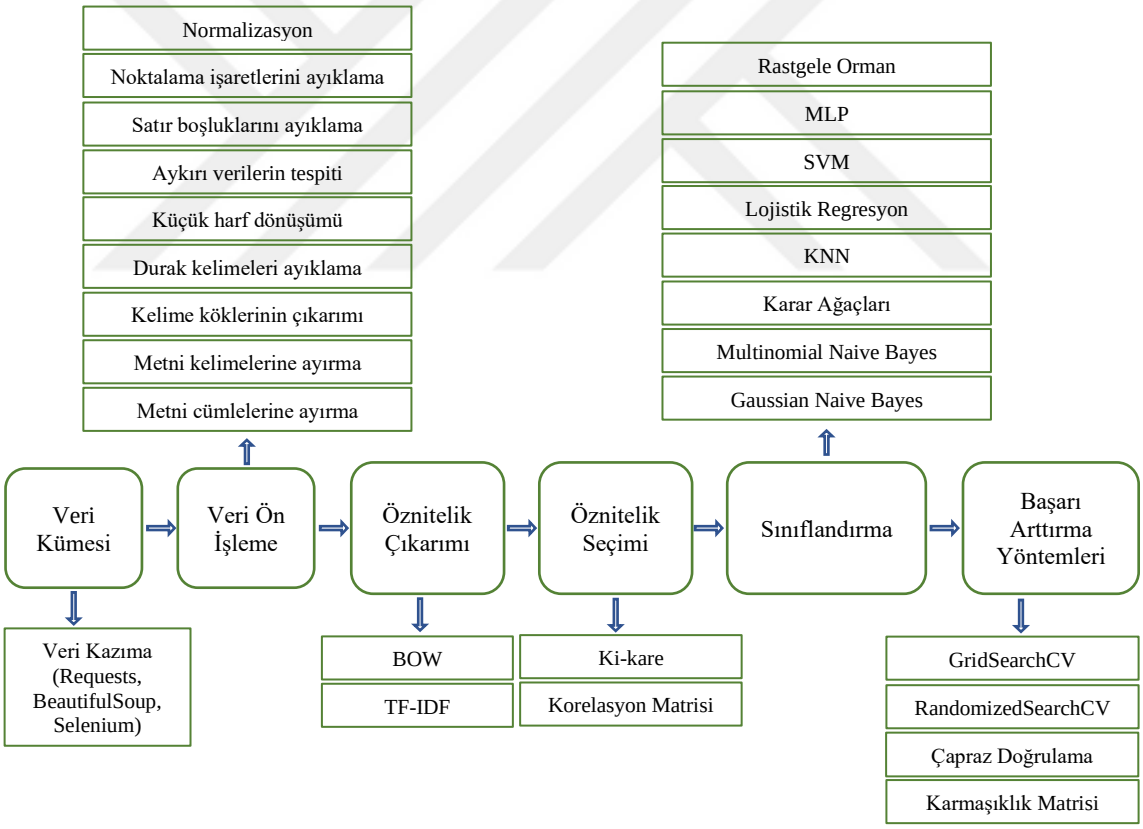
$$Hassasiyet = \frac{TP}{(TP + FN)} \tag{3.20}$$

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu tez çalışması Şekil 4.1'de görüldüğü üzere 6 temel işlem adımından meydana gelmektedir. Bunlar;

1. Verilerin toplanarak veri setinin oluşturulması,
2. Veri ön işleme,
3. Öznitelik çıkarma,
4. Öznitelik seçimi,
5. Makine öğrenmesi algoritmaları ile sınıflandırma,
6. Hiper parametre analizi ve çapraz doğrulama olarak sıralanabilir.

Bu bölümde yukarıda sıralanan adımlara ilişkin yapılan çalışmalar ve elde edilen sonuçlar detaylandırılarak açıklanmıştır.



Şekil 4.1. Çalışmanın işlem adımları

4.1. Verilerin Toplanması ve Veri Setinin Oluşturulması

Çalışmada kullanılan veriler; ülkemizde hizmet vermekte olan bilindik gazetelerin internet sitesinden alınmıştır. Veri çeşitliliğinin ve sayısının arttırılabilmesi için farklı

alanlarda gazete yazarlığı yapan birçok köşe yazarı tarafından kaleme alınmış haberler derlenmiştir. Haberlerin güncelliğinin sağlanabilmesi amacıyla, her köşe yazarının yazmış olduğu son 100 yazı (Eğer köşe yazarının 100'den daha az yayımlanan yazısı var ise olan yazıların tamamı alınmıştır.) veri setine eklenmiştir.

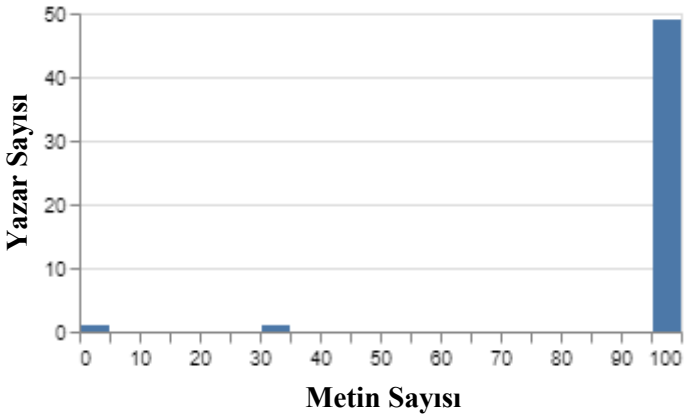
Veri setinde 51 yazara ait toplam 4923 köşe yazısı bulunmaktadır. Çizelge 3.1'de yazarların etiketleri ve yayımlanmış köşe yazılarının sayısı yer almaktadır. Veri setinin oluşturulmasında çoğunlukla Requests, BeautifulSoup ve Selenium kütüphanelerinden faydalanılmıştır. Request kütüphanesinin "get" fonksiyonu sayesinde gazete sitesinin verilerine erişim sağlanırken BeautifulSoup kütüphanesinin "find_all" fonksiyonu kullanılarak ihtiyaç duyulan HTML etiketlerinin verileri çekilmiş ve veri seti oluşturulmuştur. Ancak internet sitesinin yapısından kaynaklanan sebeplerden dolayı bu yöntemlerle sadece ilk 20 köşe yazısına ulaşılabilmiştir. Gazetenin daha fazla köşe yazısına erişebilmek ve veri çeşitliliğini sağlayabilmek amacıyla Selenium kütüphanesi kullanılmıştır. Böylelikle istenilen sayıda köşe yazısı derlenmiş ve veri setinin nihai halini alması sağlanmıştır.

4.2. Veri Ön İşleme Çalışmaları

Oluşturulan veri seti, makine öğrenmesi algoritmalarının doğrudan üzerinde çalışması için uygun değildir. Bu nedenle veri ön işleme tekniklerinin uygulanmasına ihtiyaç duyulmaktadır. Bu kapsamda; uygulanan bazı veri ön işleme tekniklerine bu bölümde yer verilirken diğer veri ön işleme tekniklerine konu bütünlüğünün sağlanması açısından öznitelik çıkarma ve sınıflandırma bölümlerinde kullanım nedenleriyle birlikte detaylı şekilde yer verilmiştir.

Şekil 4.2'de yazar ve eser sayılarının dağılımını gösteren grafik verilmiştir. Grafik incelendiğinde verilerin büyük bir çoğunluğunun belirli bir alanda kümelendiği, ancak bununla birlikte bazı verilerin dengesiz dağılım göstererek aykırı veri oluşumuna neden olduğu görülmüştür. Aykırı verilerin tespiti için veri kümesinde yer alan yazar ve eser sayıları dikkate alınmıştır. Oluşturulan veri setinde toplam 51 köşe yazarı bulunmakta olup 15'ten daha az köşe yazısı yayımlamış 1 adet yazar bulunmaktadır. Bu yazara ait toplam köşe yazısı sayısı ise 2'dir. İlgili yazara ait köşe yazısı sayısının veri seti ortalamasının altında kaldığı ve algoritmaların eğitimi için yeterli sayıda olmadığı dikkate alınarak sınıflandırma başarısını olumsuz etkileyeceği sonucuna varılmıştır. Bu nedenle; söz konusu yazara ait köşe yazıları veri setinden çıkartılarak aykırı verilerin veri setinden

temizlenmesi sağlanmıştır. Bu işlemin sonucunda veri setinde toplam 50 yazar ve 4291 köşe yazısı kalmıştır.



Şekil 4.2. Veri dağılım grafiği

Metnin kelimelerine ve cümlelerine ayrılması aşamasında durak kelimeler metinden atılmıştır. Durak kelimelerin belirlenmesinde Zemberek kütüphanesi içerisinde yer alan "Stop-words" dosyasından faydalanılmıştır. Ayrıca sınıflandırmanın başarısını arttırmak için veri seti içerisindeki benzersiz kelimeler çıkarılarak her kelimenin metin içerisinde kaç defa kullanıldığı yani frekansı hesaplanmıştır. Elde edilen kelime bazlı frekans hesaplaması sonucuna göre en çok kullanılan ve metin içerisinde en az öneme sahip sözcükler durak kelimeler listesine dahil edilmiştir. Durak kelimeler listesinde yer alan bazı sözcükler Çizelge 4.1'de sunulmuştur.

Çizelge 4.1. Zemberek kütüphanesinde yer alan bazı durak kelimeler (Akın, 2019)

Durak kelimeler		
Acaba	Filan	Otuz
Ama	Geçenlerde	Oysa
Aynen	Günbegün	Öteki
Ancak	Hâlihazırda	Pek
Bebekçe	Helalinden	Rağmen
Bilakis	Hoyrat	Sadece
Birkaçı	İçin	Sonra
Buradan	İvedi	Şöyle
Çabuk	Kaç	Tamam
Çokluk	Kasten	Taptaze
Dolayı	Malum	Tümü
Evvel	Meğer	Veya
Evet	Müthiş	Yani

4.3. Öznitelik Çıkarma

Öznitelik çıkarma yöntemleri metin tipindeki verilerin sınıflandırmaya uygun hale getirilmesi ve veri setinden daha fazla bilgi çıkarımı yapılabilmesi amacıyla kullanılmaktadır. Bu çalışmada; metin içerisinde yer alan kelimelerin listesi, kelime kökleri, metin içerisinde yer alan cümlelerin listesi, harf bazında kelime uzunluk dağılımı, kelime bazında cümle uzunluk dağılımı, kelime çeşitliliği, köklerin ortalama uzunluğu, kelime bazında ortalama cümle uzunluğu, her metin için kullanılan noktalama işaretlerinin sayısı, durak kelime olarak değerlendirilen toplam sözcük sayısı, yazarların etiketlenmesi, TF-IDF vektörünün oluşturulması, BOW vektörünün oluşturulması, harf bazında kelime uzunluk dağılımı vektörü ve kelime bazında cümle uzunluk dağılımı vektörü gibi öznitelikler çıkartılarak sınıflandırma yapılmıştır. Söz konusu özniteliklerin çıkartılması için DDİ kütüphanesi Zemberek, makine öğrenmesi temel fonksiyonları, TF-IDF ve BOW vektör modelleri kullanılmıştır. Yukarıda bahsi geçen öznitelikler ve çıkarım yöntemleri Çizelge 4.2'de nitelik bazında detaylandırılarak açıklanmıştır.

Çizelge 4.2. Veri ön işleme teknikleri kullanılarak elde edilen öznitelikler ve açıklamaları

Kısa Açıklama	Açıklama	Öznitelik
Veri setinin temizlenmesi	Veriler internet sitelerinden elde edildiğinden içerisinde satır boşlukları, büyük harflerle yazılmış sözcükler ve farklı noktalama işaretleri bulunmaktadır. DDİ tekniklerinin ve sınıflandırma için kullanılacak makine öğrenmesi algoritmalarının düzgün çalışabilmesi için kelimelerin tamamı küçük harfe dönüştürülmüş, satır boşlukları ve noktalama işaretleri kaldırılmıştır.	Temizlenmiş Metin
Metnin kelimelerine ayrılması	Temizlenmiş metin sözcüklerine ayrılmış ve sözcükler arasında yer alan durak kelimeler listeden çıkarılmıştır. Bu işlem sonucunda metnin kelimelerinden oluşan bir dizi elde edilmiştir.	Kelimeler
Metnin cümlelerine ayrılması	Veri ön işleme teknikleri uygulanmamış köşe yazılarına Zemberek kütüphanesinin TurkishSentenceExtractor sınıfının fromDocument() fonksiyonu uygulanmış ve metin cümlelerine ayrılmıştır. Daha sonra cümlelerine ayrılan metin içerisindeki durak kelimeler cümlelerden atılmıştır. Burada veri ön işleme teknikleri uygulanmamış köşe yazılarının seçilmesinin	Cümleler

	nedeni; metni cümlelerine ayırma işlemlerinde noktalama işaretlerine ve büyük/küçük harflere ihtiyaç duyulmasındandır. Bu işlemin sonucunda metnin cümlelerinden oluşan bir dizi elde edilmiştir.	
Kelime köklerinin çıkarılması	Kelimelerin bir dizi halinde tutulduğu öznitelik verisine Zemberek kütüphanesinde bulunan morfolojik analiz yöntemi uygulanarak kelime köklerinin çıkartılması sağlamıştır. Bu işlemin sonucunda kelime köklerinin yer aldığı bir dizi oluşturulmuştur.	Kökler
Kök sıklığının belirlenmesi	Köklerin yer aldığı dizide her kökten kaç adet bulunduğu tespit edilmiş ve sonuç liste tipinde kaydedilmiştir.	Kök Dağılımı
Harf bazında kelime uzunluk dağılımı	Kelimelerin dizi halinde tutulduğu öznitelik verisi kullanılarak harf bazında kelime uzunluk dağılımı çıkartılmaktadır. Oluşan vektörün her bir elemanı, belirli uzunluktaki kelimelerin metinde kaç defa geçtiğini temsil etmektedir.	Kelime Uzunluk Dağılımı
Kelime bazında cümle uzunluk dağılımı	Cümlelerin dizi halinde tutulduğu öznitelik verisi kullanılarak kelime bazında cümle uzunluk dağılımı çıkartılmaktadır. Oluşan vektörün her bir elemanı, belirli uzunluktaki cümlelerin metinde kaç defa geçtiğini belirtmektedir.	Cümle Uzunluk Dağılımı
Kelime zenginliği ölçüsü	Kelimelerine ayrılmış dizi içerisindeki benzersiz kelime sayısının toplam kelime sayısına olan oranını ifade etmektedir.	Kelime Zenginliği
Kök zenginliği ölçüsü	Köklerine ayrılmış dizi içerisindeki benzersiz kök sayısının toplam kök sayısına olan oranını belirtmektedir.	Kök Zenginliği
Harf bazında ortalama kelime uzunluğu	Kelimelerine ayrılmış dizi içerisindeki kelimelerin ortalama uzunluğunu temsil etmektedir. Ortalama kelime uzunluğu, dizi içerisindeki kelimelerin toplam uzunluğunun kelime sayısına bölünmesi ile hesaplanmaktadır.	Ort. Kelime Uzunluğu
Kelime bazında ortalama cümle uzunluğu	Cümlelerine ayrılmış dizi içerisindeki cümlelerin ortalama uzunluğunu temsil etmektedir. Ortalama cümle uzunluğu, dizi içerisindeki cümlelerin toplam uzunluğunun yine aynı dizi içerisindeki cümle sayısına bölünmesi ile hesaplanmaktadır.	Ort. Cümle Uzunluğu
Noktalama işareti sayısı	Veri ön işleme teknikleri uygulanmamış metin içerisindeki noktalama işaretlerinin sayısını belirtmektedir.	Noktalama İşareti Sayısı
Durak kelime sayısı	Temizlenmiş verinin içerisinde bulunan toplam durak kelime sayısını ifade etmektedir.	Stopwords Sayısı

Tamamı büyük harfle yazılmış kelime sayısı	Veri ön işlemeye tabi tutulmamış metin içerisindeki tamımı büyük harf ile yazılmış kelimelerin sayısını belirtmektedir.	Büyük yazılmış Kel. Say.
BOW vektörü	Köklerin yer aldığı dizinin BOW yöntemi kullanılarak sayısallaştırılmasını ifade etmektedir.	BOW Vektörü
TF-IDF vektörü	Köklerin yer aldığı dizinin TF-IDF yöntemi kullanılarak sayısallaştırılmasını ifade etmektedir.	TF-IDF Vektörü
Harf bazında kelime uzunluk dağılımı vektörü	Harf bazında kelime uzunluk dağılımı özniteliğinin DictVectorizer() yöntemi kullanılarak sayısallaştırılmasını ifade etmektedir.	Kelime Uzunluk Dağılımı Vektörü
Kelime bazında cümle uzunluk dağılımı vektörü	Kelime bazında cümle uzunluk dağılımı verisinin DictVectorizer() yöntemi ile sayısallaştırılmasını ifade etmektedir.	Cümle Uzunluk Dağılımı Vektörü
Yazarların sayısal olarak etiketlenmesi	Yazarların isimleriyle değil tam sayılar ile temsil edildiği kodlama sonucunda meydana gelen yazar-tam sayı sözlüğünü ifade etmektedir.	Hedef Öznitelik

Öznitelik çıkarma işlemi sonrasında oluşan veri setine ait görüntülere Şekil 4.3 (a, b ve c)'te yer verilmiştir.

	Tarih	Yazar	Başlık	Link	Metin	Temizlenmiş Metin	Kelimeler
0	28 Nisan 2023	Yazar_1	Ruhani lider Papa Francesco'nun Macaristan'ı z...	Link_1	\r\n\r\nVatikan Devlet Başkanı ve Katolik Kili...	vatikan devlet başkanı ve katolik kiliselerini...	[vatikan, devlet, başkanı, katolik, kiliseleri...
1	20 Nisan 2023	Yazar_1	Dünyanın en kalabalık ülkesi Hindistan	Link_2	\r\n\r\nBirleşmiş Milletler tarafından yapılan...	birleşmiş milletler tarafından yapılan bir çal...	[birleşmiş, milletler, yapılan, çalışmada, hin...
2	04 Nisan 2023	Yazar_1	NASA Artemis 2 misyonu için Ay'a gidecek astro...	Link_3	\r\n\r\nAmerikan Ulusal Havacılık ve Uzay Dair...	amerikan ulusal havacılık ve uzay dairesi nasa...	[amerikan, ulusal, havacılık, uzay, dairesi, n...
3	04 Nisan 2023	Yazar_1	Yeni İngiltere Kralı Charles'ın silüetinin bul...	Link_4	\r\n\r\nİngiltere Kraliçesi II. Elizabeth'in ö...	ingiltere kraliçesi ii elizabethin ölümünden s...	[ingiltere, kraliçesi, ii, elizabethin, ölümün...
4	01 Nisan 2023	Yazar_1	Sudan'da devam eden siyasi kriz 17 ay sonra im...	Link_5	\r\n\r\nSudan'da askerler ve siviller arasında...	sudanda askerler ve siviller arasında çıkan an...	[sudanda, askerler, siviller, arasında, çıkan,...

Şekil 4.3. Veri ön işleme sonucunda elde edilen veri seti (a)

	Tarih	Yazar	Başlık	Cümleler	Kökler	Kök Dağılımı	Kelime Uzunluk Dağılımı	Cümle Uzunluk Dağılımı	Kelime Zenginliği
0	28 Nisan 2023	Yazar_1	Ruhani lider Papa Francesco'nun Macaristan'ı z...	[vatikan devlet başkanı katolik kiliselerinin ...	[vatikan, devlet, başkan, katolik, kilise, ruh...	{'başla': 1, 'barış': 1, 'devlet': 1, 'UNK': 1...	{0: 0, 1: 1, 2: 1, 3: 1, 4: 2, 5: 2, 6: 3, 7: ...	{0: 0, 1: 0, 2: 1, 3: 5, 4: 2, 5: 3, 6: 3, 7: ...	0.842491
1	20 Nisan 2023	Yazar_1	Dünyanın en kalabalık ülkesi Hindistan	[birleşmiş milletler yapılan çalışmada hindist...	[birleş, millet, yap, çalış, hindistan, çin, U...	{'yarı': 1, '142': 1, 'UNK': 2, 'yaş': 1, 'mil...	{0: 0, 1: 0, 2: 1, 3: 2, 4: 2, 5: 5, 6: 7, 7: ...	{0: 0, 1: 0, 2: 0, 3: 2, 4: 4, 5: 10, 6: 0, 7:...	0.769784
2	04 Nisan 2023	Yazar_1	NASA Artemis 2 misyonu için Ay'a gidecek astro...	[amerikan ulusal havacılık uzay dairesi nasa u...	[amerikan, ulusal, havacılık, uzay, daire, nas...	{'dolaş': 1, 'amerikan': 1, 'tamamla': 1, 'in'...	{0: 0, 1: 1, 2: 1, 3: 3, 4: 3, 5: 1, 6: 3, 7: ...	{0: 0, 1: 0, 2: 0, 3: 7, 4: 1, 5: 2, 6: 2, 7: ...	0.821678
3	04 Nisan 2023	Yazar_1	Yeni İngiltere Kralı Charles'ın silüetinin bul...	[ingiltere kraliçesi ii elizabethin ölümünden ...	[ingiltere, kraliçe, UNK, elizabeth, ölüm, eyl...	{'eylül': 1, 'UNK': 2, 'kraliçe': 1, '3': 1, '...	{0: 0, 1: 1, 2: 1, 3: 1, 4: 3, 5: 5, 6: 3, 7: ...	{0: 0, 1: 0, 2: 1, 3: 9, 4: 3, 5: 5, 6: 4, 7: ...	0.804178
4	01 Nisan 2023	Yazar_1	Sudan'da devam eden siyasi kriz 17 ay sonra im...	[sudanda askerler siviller arasında çıkan anla...	[sudan, asker, sivil, ara, çık, anlaş, ordu, 2...	{'2021': 1, 'hal': 1, 'barış': 1, 'hayat': 1, ...	{0: 0, 1: 1, 2: 2, 3: 4, 4: 7, 5: 7, 6: 4, 7: ...	{0: 0, 1: 0, 2: 0, 3: 6, 4: 2, 5: 4, 6: 3, 7: ...	0.800000

Şekil 4.3. Veri ön işleme sonucunda elde edilen veri seti (b)

	Tarih	Yazar	Başlık	Kök Zenginliği	Ort. Kelime Uzunluğu	Ort. Cümle Uzunluğu	Noktalama İşareti Sayısı	Stopwords Sayısı	Büyük yazılmış Kel. Say.	Hedef Öznitelik
0	28 Nisan 2023	Yazar_1	Ruhani lider Papa Francesco'nun Macaristan'ı z...	0.608059	7.564103	6.658537	97	79	33	0
1	20 Nisan 2023	Yazar_1	Dünyanın en kalabalık ülkesi Hindistan	0.539568	7.467626	7.722222	82	113	50	0
2	04 Nisan 2023	Yazar_1	NASA Artemis 2 misyonu için Ay'a gidecek astro...	0.643357	7.174825	7.944444	79	118	31	0
3	04 Nisan 2023	Yazar_1	Yeni İngiltere Kralı Charles'ın silüetinin bul...	0.600522	7.671018	8.326087	109	131	17	0
4	01 Nisan 2023	Yazar_1	Sudan'da devam eden siyasi kriz 17 ay sonra im...	0.622857	6.897143	9.459459	87	76	43	0

Şekil 4.3. Veri ön işleme sonucunda elde edilen veri seti (c)

4.4. Makine Öğrenmesi Algoritmaları ile Sınıflandırma

Çalışmanın bu bölümünde; farklı özniteliklerin, temsil yöntemlerinin ve öznitelik seçimlerinin sınıflandırmaya olan etkisi, hiper parametre çalışmaları sonucunda elde edilen sınıflandırma başarılarının karşılaştırılması ve sonuçların görselleştirilerek incelenmesi yer almaktadır.

Veri ön işleme ve öznitelik çıkarma işlemlerinin tamamlanmasının ardından, veri setini bağımlı ve hedef değişkenlerine ayırabilmek için öznitelik değerleri x ve y olmak üzere iki farklı değişkene atanmıştır. Burada x değişkenine; Metin, Temizlenmiş Metin, Kelimeler, Cümleler, Kökler, Kök Dağılımı, Kelime Uzunluk Dağılımı, Cümle Uzunluk Dağılımı, Kelime Zenginliği, Kök Zenginliği, Ort. Kelime Uzunluğu, Ort. Cümle Uzunluğu, Noktalama İşareti Sayısı, Stopwords Sayısı ve Büyük yazılmış Kel. Say. sütunlarında bulunan veriler atanırken, y değişkenine sadece yazarların tam sayı olarak kodlanmış olduğu Hedef Öznitelik sütunundaki değerler atanmıştır.

Bağımlı ve hedef değişkenler olmak üzere ikiye ayrılan veri seti, makine öğrenmesi algoritmaları ile kelime temsil yöntemlerinin eğitilebilmesi ve test edilebilmesi amacıyla yeniden iki parçaya bölünmüştür. Toplam 50 yazardan ve 4921 köşe yazısından oluşan veri setinin %75'i (3690 adet köşe yazısı) eğitim verisi, %25'i (1231 adet köşe yazısı) ise test verisi olacak şekilde ayarlanmıştır.

Metin tabanlı sınıflandırma çalışmalarında metnin kapsamına ilişkin en fazla bilgi genellikle kelime köklerinde yer almaktadır. Bu nedenle kelime kökleri metin sınıflandırma uygulamalarında ayırt edici öznitelik olarak önemli bir yere sahiptir. Ancak sınıflandırma algoritmalarının büyük bir kısmının kategorik veriler üzerinde çalışamaması nedeniyle bu tür verilerin dönüştürülmesi gerekmektedir. Eğitim veri setinde yer alan kelime kökleri TF-IDF ve BOW vektör modellerinin eğitiminde kullanılmış ve aynı zamanda bu modeller vasıtasıyla sayısallaştırılmıştır. Eğitilen TF-IDF ve BOW vektör modelleri test verisi içerisinde bulunan kelime kökleri için de kullanılarak veri setinde yer alan bütün kelime köklerinin vektörize edilmesi sağlanmıştır. Benzer şekilde DictVectorizer() yöntemi kullanılarak kelime ve cümle uzunluk dağılımı öznitelikleri de vektörize edilmiştir.

Öznitelik çıkartma işlemi sonucunda kelime zenginliği ölçüsü, kök zenginliği ölçüsü, ortalama kelime ve cümle uzunluğu, noktalama işareti sayısı, durak kelime sayısı gibi bazı istatistiki değerler elde edilerek veri setine kaydedilmiştir.

	Kelime Zenginliği	Kök Zenginliği	Ort. Kelime Uzunluğu	Ort. Cümle Uzunluğu	Noktalama İşareti Sayısı	Stopwords Sayısı	Büyük yazılmış Kel. Say.
0	0.842491	0.608059	7.564103	6.658537	97	79	33
1	0.769784	0.539568	7.467626	7.722222	82	113	50
2	0.821678	0.643357	7.174825	7.944444	79	118	31
3	0.804178	0.600522	7.671018	8.326087	109	131	17
4	0.800000	0.622857	6.897143	9.459459	87	76	43

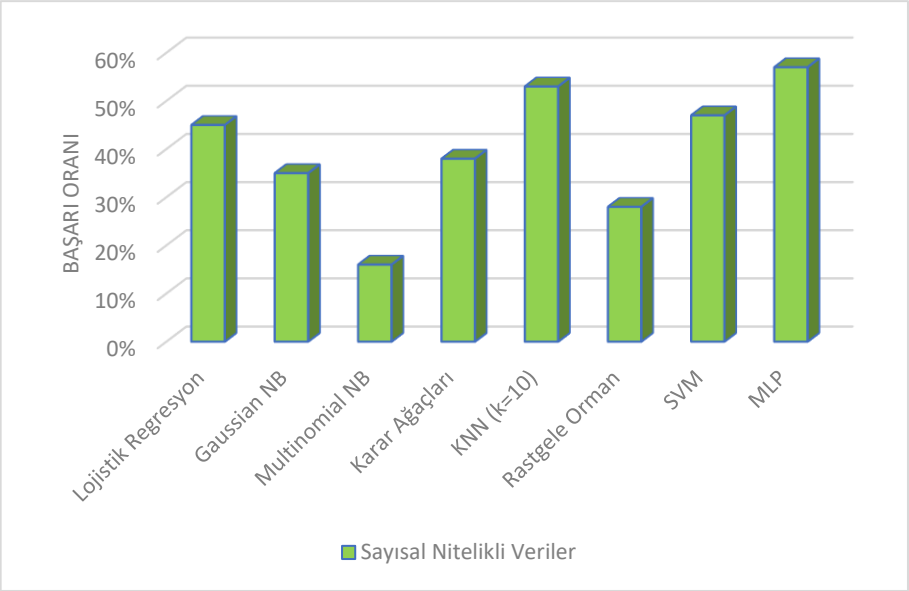
Şekil 4.4. Veri setinde yer alan sayısal veriler

Ancak Şekil 4.4'ten de görülebileceği üzere verilerin değerleri arasında farklılıklar mevcuttur. Bu durum veri dağınıklığına neden olacağından sınıflandırma başarısını düşürecektir. Veri dağınıklığının giderilebilmesi için eğitim ve test verisinde yer alan sayısal öznitelikler Min-Max yöntemi ile normalize edilmiştir. Böylelikle eğitim ve test verileri hem vektörize hem de normalize edilerek sınıflandırmaya uygun hale getirilmiştir.

	Kelime Zenginliği	Kök Zenginliği	Ort. Kelime Uzunluğu	Ort. Cümle Uzunluğu	Noktalama İşareti Sayısı	Stopwords Sayısı	Büyük yazılmış Kel. Say.
0	0.738034	0.577333	0.672535	0.164333	0.088267	0.077326	0.135802
1	0.609167	0.460632	0.641663	0.203982	0.072121	0.121887	0.205761
2	0.701146	0.637478	0.547967	0.212266	0.068891	0.128440	0.127572
3	0.670127	0.564492	0.706747	0.226492	0.101184	0.145478	0.069959
4	0.662722	0.602549	0.459110	0.268740	0.077503	0.073394	0.176955

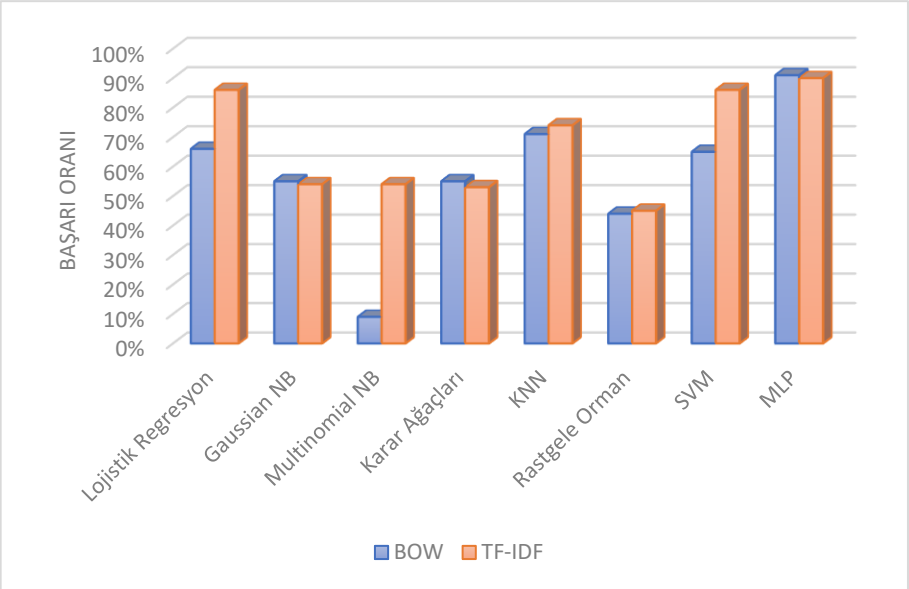
Şekil 4.5. Sayısal verilerin normalize edilmesi

Şekil 4.5'te yer alan normalize edilmiş sayısal nitelikteki verilerin sınıflandırılması sonucunda ulaşılan başarı oranları Şekil 4.6'da verilmiştir. Şekilden de görülebileceği üzere sayısal nitelikteki veriler tek başına sınıflandırıldığında düşük başarı oranı elde edilmektedir. Bu nedenle söz konusu sayısal nitelikteki verilerin diğer öznitelikler ile birlikte kullanılarak sınıflandırılmaya dahil edilmesine karar verilmiştir.



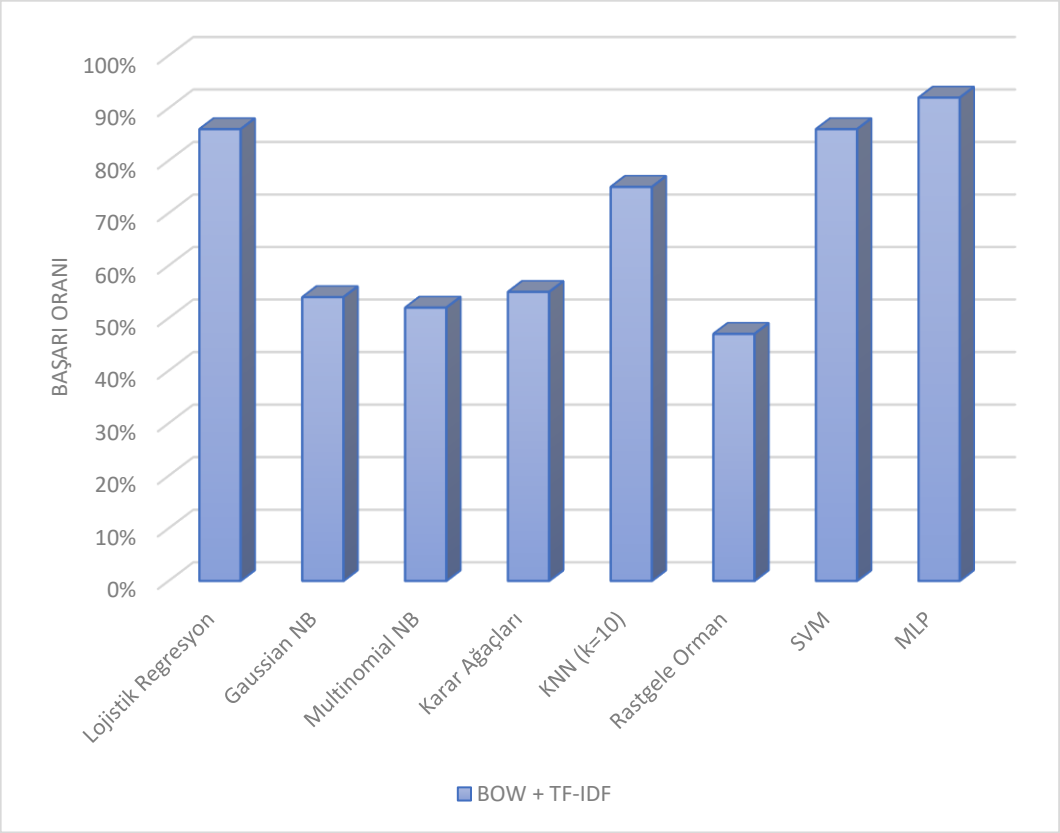
Şekil 4.6. Sayısal nitelikteki istatistiksel verilerin sınıflandırılması

Kelime köklerinin sayısallaştırılması sonucunda elde edilen BOW ve TF-IDF vektörleri birbirinden ayrı olacak şekilde diğer öznitelikler ile birleştirilmiş ve ortaya çıkan veri, sınıflandırma algoritmaları yardımıyla kategorize edilmiştir. Bu aşamada sınıflandırma için kullanılan algoritmalara herhangi bir parametre ayarlaması yapılmamış ve varsayılan parametreler kullanılarak sınıflandırma gerçekleştirilmiştir. KNN algoritmasında yer alan k parametresi, 2 ile 11 arasındaki bütün değerler için teker teker denenerek en yüksek başarı oranını sağlayan değer olarak belirlenmektedir. Şekil 4.7'de k parametresi BOW vektörü için 4, TF-IDF vektörü için ise 10 olarak belirlenmiştir.



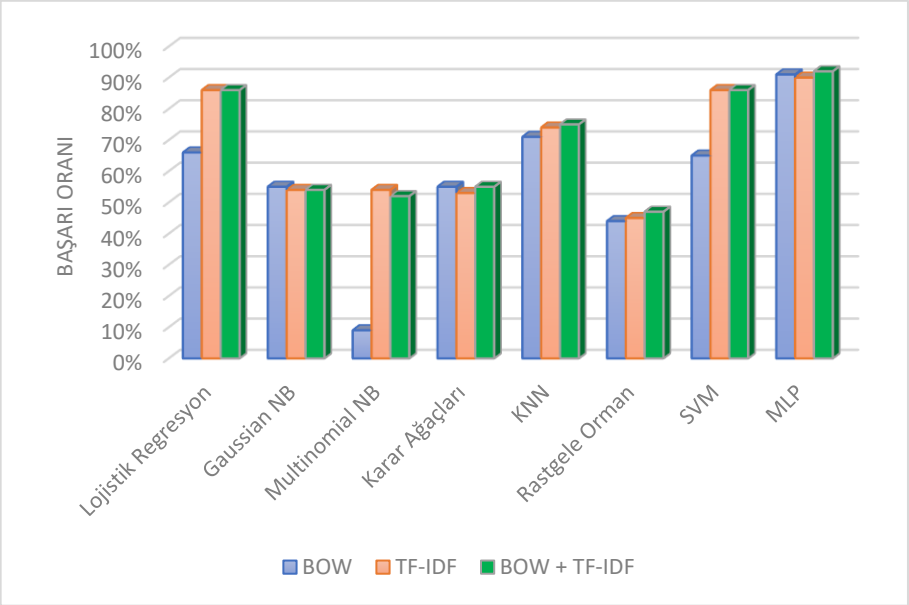
Şekil 4.7. BOW ve TF-IDF vektörlerinin sınıflandırma sonuçları

Yukarıdaki grafikte yer alan başarı oranları TF-IDF ve BOW vektör verilerinin birbirinden ayrı olarak sınıflandırılması sonucunda elde edilmiştir. Söz konusu verilerin birleştirilmesinin sınıflandırma sonucunu hangi yönde etkileyeceğini görmek için veriler birleştirilmiş ve sınıflandırma işlemi yeniden yapılmıştır. TF-IDF ve BOW vektör verilerinin birleştirilmesi sonucunda ortaya çıkan sınıflandırma başarısı Şekil 4.8'de sunulmuş olup burada KNN algoritması için k değeri 10 olarak belirlenmiştir.



Şekil 4.8. BOW ve TF-IDF vektör verilerinin birleştirilmesi sonucunda elde edilen başarı oranları

TF-IDF ile BOW vektör verileri tek başına ve birleştirilerek sınıflandırıldığında elde edilen sonuçlar yukarıdaki grafiklerde (Şekil 4.7 ve Şekil 4.8) ayrı ayrı gösterilmiştir. Sonuçlar arasında karşılaştırma ve analiz yapılabilmesi amacıyla ulaşılan başarı oranları Şekil 4.9'da ve Çizelge 4.3'te birlikte verilmiştir.



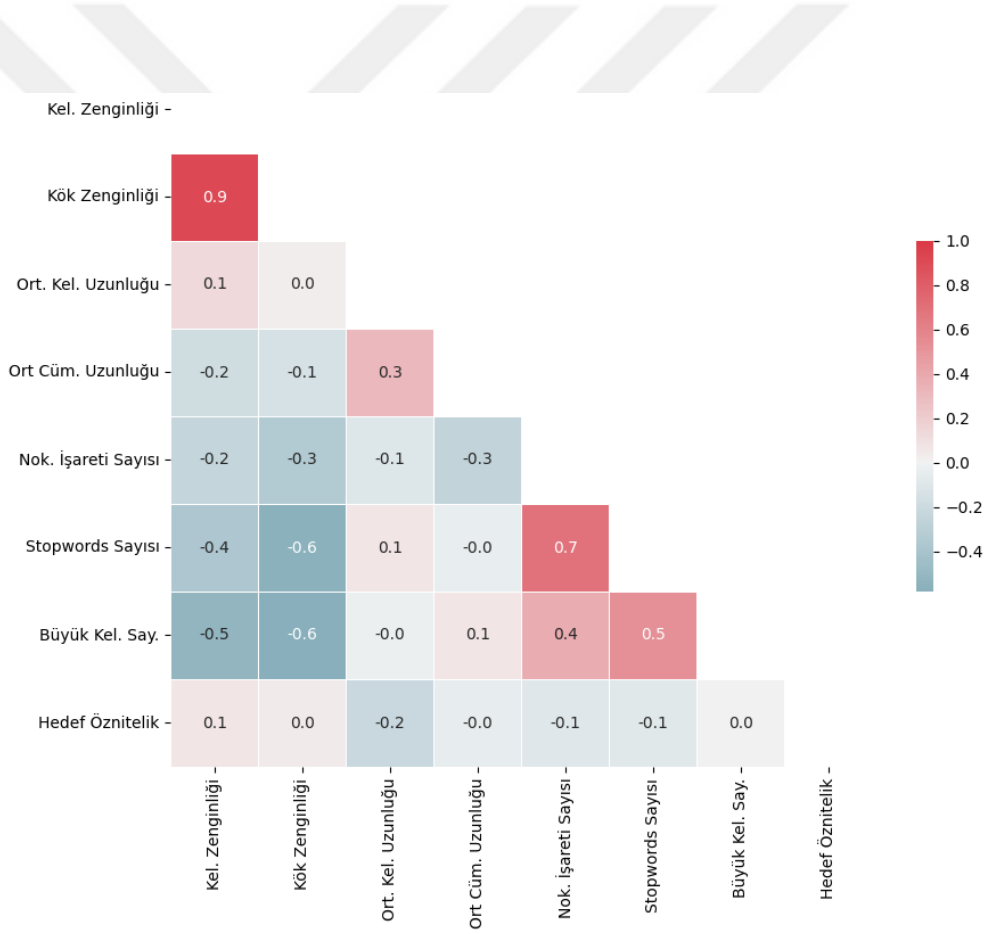
Şekil 4.9. Sınıflandırma sonuçlarının karşılaştırılması

Çizelge 4.3. Başarı oranlarının karşılaştırılması

Algoritmalar	**Sayısal Nitelikli Veriler**	**BOW**	**TF-IDF**	**BOW + TF-IDF**
Lojistik Regresyon	%45.73	%66.28	%86.10	**%86.83**
Gaussian NB	%35.01	**%55.15**	%54.91	%54.91
Multinomal NB	%16.57	%9.34	**%54.50**	%52.72
Karar Ağaçları	%38.74	**%55.40**	%53.45	%55.07
KNN	%53.77	%71.24	%74.41	**%75.22**
Rastgele Orman	%28.43	%44.3	%45.97	**%47.03**
SVM	%47.84	%65.88	%86.43	**%86.83**
MLP	%57.27	%91.71	%90.65	**%92.36**

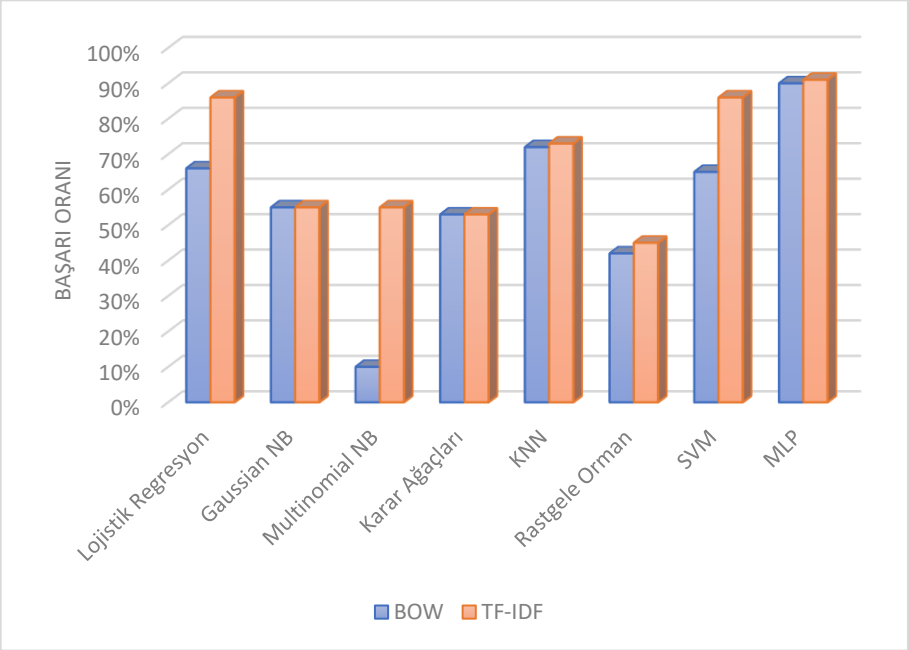
Yukarıda bahsedildiği üzere BOW ve TF-IDF yöntemleri veri setinde yer alan kelime köklerinin vektörize edilmesi için kullanılmıştır. Bu nedenle, BOW ve TF-IDF modelleri veri setinde yer alan benzersiz kelime sayısı kadar öznitelik üretmektedir. Oluşturulan her özniteliğin aslında bir kelime köküne denk geldiği göz önünde bulundurulduğunda, bütün özniteliklerin aynı seviyede bilgi taşımadığı ve metnin sınıflandırılması aşamasında ihtiyaç duyulan ayırt edicilik düzeylerinin farklılık gösterdiği anlaşılacaktır. Düşük seviyede bilgi taşıyan önemsiz özniteliklerin veri setinden çıkartılması hem veri boyutunu azaltacak hem de sınıflandırma başarısını arttıracaktır. Bu nedenle ayırt ediciliği yüksek ve önemsiz özniteliklerin belirlenmesi gerekmektedir. Bunun için birçok faklı yöntem bulunmakta olup bu çalışmada Ki-kare öznitelik seçim yöntemi kullanılmıştır. BOW ve TF-IDF vektör verilerinin her biri

yaklaşık olarak 24 bin adet öznitelikten meydana gelmektedir. Bu öznitelikler Ki-kare yöntemi ile ayırt ediciliği yüksek olandan düşük olana doğru sıralanmış ve yaklaşık olarak üçte bir oranında önemsiz veri ayıklanarak öznitelik sayısı 17 bine düşürülmüştür. BOW ve TF-IDF vektör verilerinin dışında kalan diğer özniteliklerin seçimi için ise Korelasyon Matrisi kullanılmıştır. Korelasyon Matrisi bağımlı öznitelikler ile hedef öznitelik arasındaki ilişkinin yönünün ve şiddetinin belirlenmesinde kullanılmaktadır. Şekil 4.10'da normalize edilmiş sayısal nitelikteki veriler ile hedef değişken arasındaki ilişkinin Korelasyon Matrisi yer almaktadır. Şekilde verilen Korelasyon Matrisi incelendiğinde bağımlı değişkenlerin arasında belirgin bir ilişki olduğu, ancak hedef değişken ile bağımlı değişkenler arasında güçlü bir ilişki olmadığı görülmektedir. Bu nedenle söz konusu sayısal nitelikteki veriler arasından herhangi bir öznitelik seçimi yapılmamıştır.



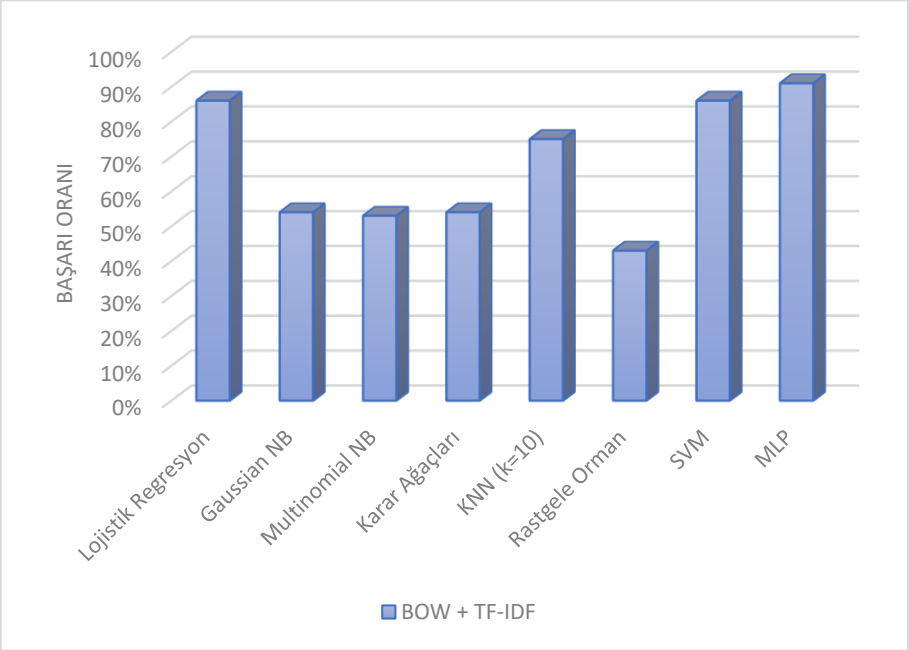
Şekil 4.10. Öznitelik seçimi için korelasyon matrisi

Öznitelik seçimi sonrasında nitelik sayısı önemli ölçüde azalan TF-IDF ve BOW vektör verileri yeniden sınıflandırılmıştır. Sınıflandırma sonucuna ait başarı oranları Şekil 4.11'de sunulmuş olup KNN algoritmasının k değeri BOW vektörü için 4, TF-IDF vektörü için ise 10 olarak belirlenmiştir.



Şekil 4.11. Öznitelik seçimi sonrasında TF-IDF ve BOW vektörlerinin yeniden sınıflandırılması

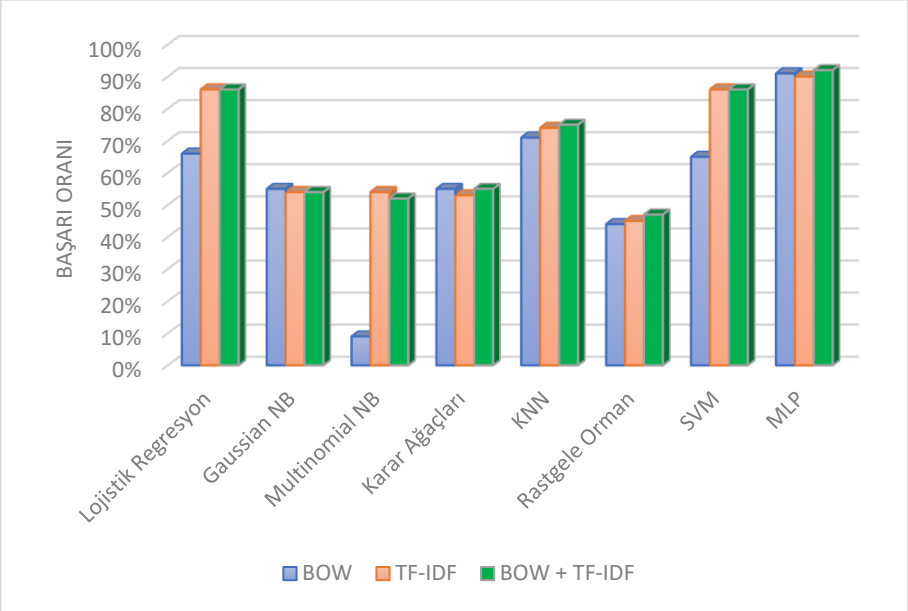
Öznitelik seçiminden geçirilen BOW ve TF-IDF vektörleri ayrı ayrı sınıflandırıldığında Şekil 4.11'de bulunan başarı oranları elde edilmiştir. Vektörlerin birleştirilmesi durumunda sınıflandırma başarısının hangi yönde değişeceğinin gözlemlenebilmesi amacıyla vektörler birleştirilerek yeniden sınıflandırma yapılmıştır.



Şekil 4.12. Öznitelik seçimi sonrasında TF-IDF ve BOW vektörlerinin birleştirilerek sınıflandırılması

Vektörlerin birleştirilerek sınıflandırılması sonucunda Şekil 4.12'de görülen başarı oranları elde edilmiştir. Sınıflandırmada kullanılan KNN algoritmasının k değeri 10 olarak belirlenmiştir.

Veri öznitelik seçimi sonrasında yapılan sınıflandırmalarda elde edilen başarı oranları yukarıda ayrı ayrı verilmiştir. Sonuçların kıyaslanabilmesi ve gerekli çıkarımların yapılabilmesi amacıyla elde edilen başarı oranları Şekil 4.13'te bir arada verilmiştir.



Şekil 4.13. Öznitelik seçimi sonrasında yapılan sınıflandırma sonuçlarının karşılaştırılması

Çalışmanın bu aşamasına kadar yapılan sınıflandırma işlemlerinde ulaşılan ortalama başarı oranlarına Çizelge 4.4'te yer verilmiştir.

Çizelge 4.4. Sınıflandırma algoritmalarının doğruluk oranları

Algoritmalar				**Ki-kare**		
	BOW	**TF-IDF**	**BOW + TF-IDF**	**BOW**	**TF-IDF**	**BOW + TF-IDF**
Lojistik Regresyon	%66.28	%86.10	%86.83	%66.28	%86.10	**%86.83**
Gaussian NB	%55.15	%54.91	%54.91	%55.07	**%55.72**	%54.91
Multinomal NB	%9.34	%54.50	%52.72	%10.50	**%55.07**	%53.53
Karar Ağaçları	**%55.40**	%53.45	%55.07	%53.85	%53.69	%54.67
KNN	%71.24	%74.41	**%75.22**	%72.81	%73.92	%75.14
Rastgele Orman	%44.3	%45.97	**%47.03**	%42.40	%45.81	%43.13
SVM	%65.88	%86.43	%86.83	%65.88	%86.59	**%86.83**
MLP	%91.71	%90.65	**%92.36**	%90.82	%91.04	%91.87

Yukarıda yer alan çizelge incelendiğinde bazı algoritmaların diğerlerine oranla daha az başarılı olduğu görülmüştür. Bu nedenle algoritmalar için hiper parametre analizi yapılarak başarı oranlarının yükseltilmesi hedeflenmiştir. Bu kapsamda; algoritmaların çalışma hızı ve olası parametre kombinasyonları göz önünde bulundurularak RandomizedSearchCV veya GridSearchCV yöntemlerinden uygun olan seçilmiş ve ilgili algoritmanın hiper parametre analizinde kullanılmıştır. Algoritmalar için hesaplanan optimum hiper parametre değerleri ve kullanılan parametre analiz yöntemleri Çizelge 4.5'te verilmiştir.

Çizelge 4.5. Hiper parametre analizleri

Algoritmalar	**Parametre Havuzu**	**Seçilen Değerler (Sırasıyla)**	**Başarı Oranı**	**Hiper Parametre Yöntemi**
Lojistik Regresyon + Ki-kare + BOW ve TF-IDF	Penalty=['l1', 'l2', 'elasticnet', 'none'], C=np.logspace(-4, 4, 20), Solver=['lbfgs', 'newton-cg', 'liblinear', 'sag', 'saga'], max_iter=[100, 1000]	Solver='newton-cg', penalty='l2', max_iter=1000, C=3792.690190732246	%88.2	RandomizedSearchCV
Gaussian NB + Ki-kare + TF-IDF	Priors=[None, [0.1,]*len(n_classes),], var_smoothing=[1e-9, 1e-6, 1e-12],	Priors=None, var_smoothing=1e-6,	%56	GridSearchCV
Gaussian NB+ Ki-kare + BOW ve TF-IDF	Priors=[None, [0.1,]*len(n_classes),], var_smoothing=[1e-9, 1e-6, 1e-12],	Priors=None, var_smoothing=1e-6,	%55	RandomizedSearchCV
Multinomal NB+ Ki-kare + TF-IDF	Alpha=[0.0001, 0.001, 0.01, 0.1, 0.5, 1.0, 10.0, 100], fit_prior=[True, False], class_prior=[None, [0.1,]* len(n_classes),]	fit_prior=False, class_prior'[0.1,..,01], alpha=0.01	%85.2	RandomizedSearchCV
Multinomal NB+ Ki-kare + BOW ve TF-IDF	Alpha=[0.0001, 0.001, 0.01, 0.1, 0.5, 1.0, 10.0, 100], fit_prior=[True, False], class_prior=[None, [0.1,]* len(n_classes),]	fit_prior=False, class_prior=None, alpha=0.01	%84.5	GridSearchCV
Karar Ağaçları + Ki-kare + BOW ve TF-IDF	max_leaf_nodes=[2,3,…,100], min_samples_split=[2, 3, 4], Random_state=42	min_samples_split=3, max_leaf_nodes=87, Random_state=42	%53.5	RandomizedSearchCV
Karar Ağaçları + BOW	max_leaf_nodes=[2,3,…,100], min_samples_split=[2, 3, 4], Random_state=42	min_samples_split=4, max_leaf_nodes=98, Random_state=42	%54	RandomizedSearchCV
KNN + Ki-kare + BOW ve TF-IDF	Scikit-Learn varsayılan parametreleri	Varsayılan K=10	%75.3	-
Rastgele Orman + BOW ve TF-IDF	n_estimators=[int(x) for x in np.linspace(start=10,stop=80,num=10)], max_features=['auto', 'sqrt'], max_depth=[2,4], min_samples_split=[2, 5], min_samples_leaf'=[1, 2] bootstrap=[True, False]	Bootstrap=False, max_depth=4, max_features='sqrt', min_samples_leaf'=2, min_samples_split=5, n_estimators=80	%51.7	GridSearchCV
Rastgele Orman + Ki-kare + BOW ve TF-IDF	n_estimators=[int(x) for x in np.linspace(start=10,stop=80,num=10)], max_features=['auto', 'sqrt'],	Bootstrap=False, max_depth=4, max_features='sqrt', min_samples_leaf'=2,	%52.0	GridSearchCV

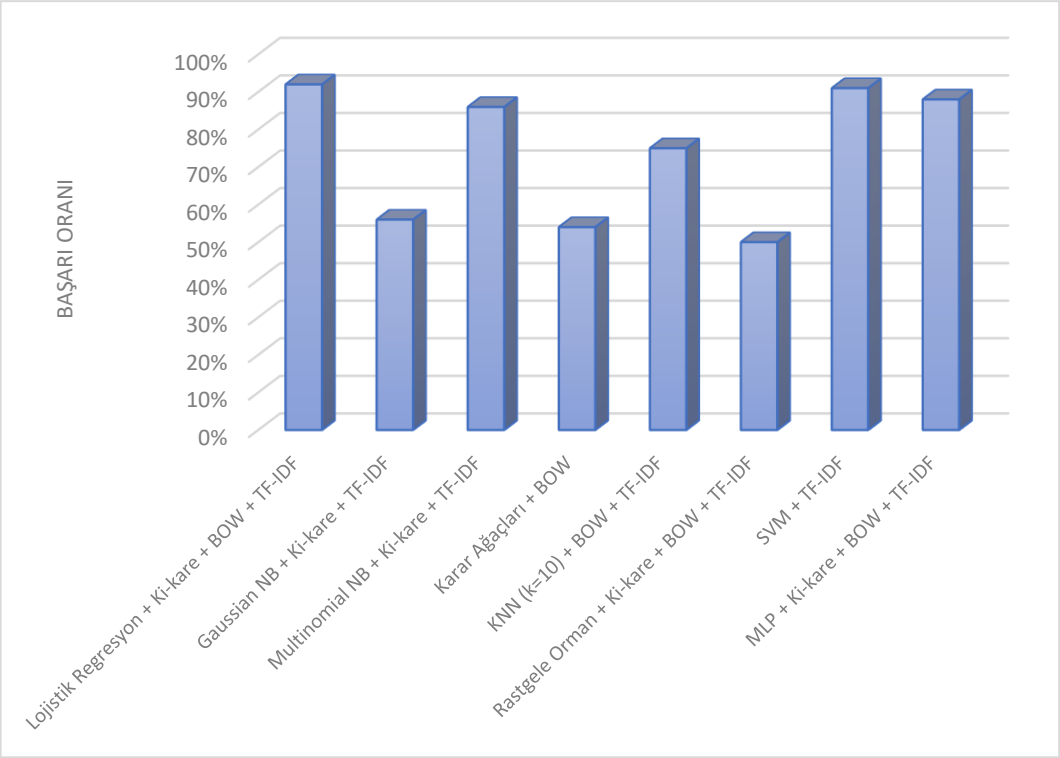
	max_depth=[2,4], min_samples_split=[2, 5], min_samples_leaf'=[1, 2] bootstrap=[True, False]	min_samples_split=2, n_estimators=72		
SVM + Ki-kare + BOW ve TF-IDF	Kernel=['rbf', 'linear', 'sigmoid', 'poly'], C=loguniform(1e0, 1e3), Gamma=loguniform(1e-4, 1e-2), class_weight=['balanced', None], degree=[2, 3, 4]	C = 216, Kernel = linear	%90	RandomizedSearchCV
SVM + TF-IDF	Kernel=['rbf', 'linear', 'sigmoid', 'poly'], C=loguniform(1e0, 1e3), Gamma=loguniform(1e-4, 1e-2), class_weight=['balanced', None], degree=[2, 3, 4]	C = 18, Kernel = linear	%90	RandomizedSearchCV
MLP + Ki-kare + BOW ve TF-IDF	hidden_layer_sizes=[(10,30,10),(20,)], activation=['tanh', 'relu'], solver=['sgd', 'adam'], alpha=[0.0001,0.001,0.01,0.1,0.5,1.0, 10.0,100], learning_rate=['constant','adaptive']	Solver='adam', learning_rate='adaptive', hidden_layer_sizes=(20,), alpha=0.1, activation='tanh'	%87.3	RandomizedSearchCV
MLP + Ki-kare + BOW ve TF-IDF	hidden_layer_sizes=[(10,30,10),(20,)], activation=['tanh', 'relu'], solver=['sgd', 'adam'], alpha=[0.0001,0.001,0.01], learning_rate=['constant','adaptive']	Activation='relu', alpha=0.001, hidden_layer_sizes=(20,), learning_rate='constant', solver='adam'	%85.9	GridSearchCV

Hiper parametre analizlerinde RandomizedSearchCV yöntemi kullanılan algoritmalar için iterasyon sayısı olabildiğince yüksek tutularak sınıflandırma başarısının maksimuma çıkartılması amaçlanmıştır. Aynı zamanda çapraz doğrulama sayısı 10 seçilerek sınıflandırma algoritmalarının veri dağılımından kaynaklı yapacağı hataların önüne geçilmiştir.

Kullanılan sınıflandırma algoritmalarının parametreleri Çizelge 4.5'te yer alan değerlere göre ayarlanmış ve çalışmanın bu aşamasından sonra K-katlı Çapraz Doğrulama yöntemi ile birlikte veri setinin yeniden sınıflandırılmasında kullanılmıştır. Sınıflandırma sonucuna ilişkin verilere Çizelge 4.6'da ve Şekil 4.14'te yer verilmiştir.

Çizelge 4.6. Hiper parametre analizi sonucunda algoritmaların başarı oranları

Algoritmalar				**Ki-kare**			
	BOW	**TF-IDF**	**BOW + TF-IDF**	**BOW**	**TF-IDF**	**BOW + TF-IDF**	**Hiper Parametre**
Loj. Regresyon	%66.28	%86.10	%86.83	%66.28	%86.10	%86.83	**%91.8**
Gaussian NB	%55.15	%54.91	%54.91	%55.07	%55.72	%54.91	**%56.2**
Multinomal NB	%9.34	%54.50	%52.72	%10.50	%55.07	%53.53	**%86.1**
Karar Ağaçları	**%55.40**	%53.45	%55.07	%53.85	%53.69	%54.67	%53.5
KNN	%71.24	%74.41	%75.22	%72.81	%73.92	%75.14	**%75.5**
Rastgele Orman	%44.3	%45.97	%47.03	%42.40	%45.81	%43.13	**%50**
SVM	%65.88	%86.43	%86.83	%65.88	%86.59	%86.83	**%90.5**
MLP	%91.71	%90.65	**%92.36**	%90.82	%91.04	%91.87	%88



Şekil 4.14. Hiper parametre analizi sonucunda elde edilen doğruluk oranları

Hiper parametre analizi sonucunda kullanılan algoritmaların performansı en üst seviyeye çıkartılmış ve bu aşamadan sonra elde edilen tahmin değerlerinin daha detaylı incelenebilmesi için sonuçlar performans ölçüm metrikleriyle birlikte Çizelge 4.7'de verilmiştir. Ayrıca çizelgede yer alan sınıflandırma algoritmaları başarı oranlarına göre sıralanmıştır.

Çizelge 4.7. Sınıflandırma algoritmalarının performans ölçüm metrikleri

Algoritmalar	**Doğruluk (Accuracy)**	**Hassasiyet (Precision)**	**Duyarlılık (Recall)**	**F1-Skoru**
Lojistik Regresyon + Ki-kare + BOW + TF-IDF	%91.8	%92	%92	%92
SVM + TF-IDF	%90.5	%91	%90	%90
MLP + Ki-kare + BOW + TF-IDF	%88	%88	%88	%88
Multinomial NB + Ki-kare + TF-IDF	%86.1	%86	%86	%85
KNN (k=10) + BOW + TF-IDF	%75.5	%77	%76	%75
Gaussian NB + Ki-kare + TF-IDF	%56.2	%71	%56	%58
Karar Ağaçları + BOW	%53.5	%55	%54	%53
Rastgele Orman + Ki-kare + BOW + TF-IDF	%50	%47	%50	%44

Çizelge 4.7'de çalışmada tercih edilen 8 adet sınıflandırma algoritmasının hangi öznitelik seçimi ve temsil yöntemi ile birlikte kullanıldığında doğruluk metriğine göre

maksimum başarıyı elde ettiği yer almaktadır. Çizelge incelendiğinde; Lojistik Regresyon algoritmasının Ki-kare öznitelik seçim yöntemiyle birlikte BOW ve TF-IDF temsil yöntemleri ile sayısallaştırılan veri seti üzerinde, SVM algoritmasının TF-IDF temsil yöntemi ile sayısallaştırılan veri seti üzerinde, MLP algoritmasının Ki-kare öznitelik seçim yöntemiyle birlikte BOW ve TF-IDF temsil yöntemleri ile sayısallaştırılan veri seti üzerinde, Multinomal Naive Bayes algoritmasının Ki-kare öznitelik seçim yöntemiyle birlikte TF-IDF temsil yöntemi ile sayısallaştırılan veri seti üzerinde, KNN algoritmasının (k değeri 10 olarak seçilmiştir.) BOW ve TF-IDF temsil yöntemleri ile sayısallaştırılan veri seti üzerinde, Gaussian Naive Bayes algoritmasının Ki-kare öznitelik seçim yöntemiyle birlikte TF-IDF temsil yöntemi ile sayısallaştırılan veri seti üzerinde, Karar Ağaçları algoritmasının BOW temsil yöntemi ile sayısallaştırılan veri seti üzerinde ve son olarak Rastgele Orman algoritmasının Ki-kare öznitelik seçim yöntemiyle birlikte BOW ve TF-IDF temsil yöntemleri ile sayısallaştırılan veri seti üzerinde en yüksek başarıyı elde ettiği görülmektedir.

Sonuç olarak doğruluk oranı, hassasiyet, duyarlılık ve F1-skoru metrikleri göz önünde bulundurulduğunda; en başarılı sınıflandırma kombinasyonunun Lojistik Regresyon, Ki-kare, BOW ve TF-IDF yöntemlerinin birleşimi sonucunda elde edildiği gözlemlenmiştir.

5. SONUÇLAR VE ÖNERİLER

Bu bölümde çalışmanın adımlarına ve kullanılan yöntemlere kısaca değinilerek bilgilendirme yapılmıştır. Ayrıca çalışma sonucuna ait veriler, yapılan değerlendirmeler ve öneriler bu bölümde yer almaktadır.

5.1 Sonuçlar

Bu çalışmada 50 yazardan ve 4921 köşe yazısından oluşan bir veri seti kullanılarak metin tabanlı sınıflandırma işlemi yapılmıştır. Veri setine dördüncü bölümde değinilen bir dizi veri ön işleme adımı uygulanarak verinin sınıflandırılmaya uygun hale getirilmesi sağlanmıştır. Gerekli ön işlemlerden geçirilen veri seti, makine öğrenmesi algoritmalarının eğitilmesi ve test edilmesi için %75 eğitim, %25 test verisi olacak şekilde iki parçaya bölünmüştür.

Çalışmada kullanılan bazı sınıflandırma algoritmalarının kategorik veriler üzerinde işlem yapamaması nedeniyle veri setinde yer alan kelime kökleri TF-IDF ve BOW vektör temsil yöntemleriyle sayısallaştırılmıştır. Böylelikle veri hem sınıflandırma algoritmalarının kullanımına uygun hale getirilmiş hem de vektör modelleri ile ağırlıklandırılarak daha anlamlı hale dönüştürülmüştür.

Çalışmada toplam 8 adet makine öğrenmesi sınıflandırma algoritması kullanılmış olup bunlar sırasıyla; Gaussian Naive Bayes, Multinomial Naive Bayes, Karar Ağaçları, KNN, Lojistik Regresyon, Rastgele Orman, MLP ve SVM'dir. Söz konusu algoritmalara; veri setinde yer alan sayısal öznitelikler, BOW yöntemi ile oluşturulan vektör, TF-IDF yöntemi ile oluşturulan vektör, BOW ve TF-IDF yöntemleriyle oluşturulan vektörlerin birleşimi, Ki-kare yöntemi ile öznitelik sayısı düşürülen BOW vektörü, Ki-kare yöntemi ile öznitelik sayısı düşürülen TF-IDF vektörü, Ki-kare yöntemi ile öznitelik sayısı düşürülen BOW ve TF-IDF vektörlerinin birleşimi sırasıyla verilerek elde edilen başarı oranları kıyaslanmış ve gerekli incelemeler yapılmıştır. Bu doğrultuda yapılan analizler sonucunda aşağıda yer alan çıkarımlar elde edilmiştir.

a. Normalize edilmiş sayısal öznitelikler kullanılarak yapılan sınıflandırmaların sonucunda ulaşılan başarı oranları ile TF-IDF ve BOW vektörleri kullanılarak ulaşılan başarı oranları kıyaslandığında, sayısal içerikli özniteliklerin anılan diğer öznitelikler kadar ayırt edici olmadığı görülmüştür. Ayrıca sayısal öznitelikler ile hedef değişken arasındaki ilişkinin yorumlanabilmesi için Korelasyon Matrisi kullanılmış ve elde edilen

matris sonuçları incelendiğinde sayısal öznitelikler ile hedef değişken arasında ayırt edici düzeyde bir ilişki bulunmadığını gözlemlenmiştir.

b. Sayısal niteliklerin veri setinden öznitelik çıkarım yöntemleriyle elde edilen istatistiki veriler olduğu, vektör temsil yöntemlerinin ise kelime köklerinden oluşturulduğu bilinmektedir. Buradan hareketle sayısal öznitelikler ile vektör temsil yöntemleri arasındaki sınıflandırma başarısı dikkate alındığında; kelime köklerinin istatistiki verilere göre daha fazla bilgi taşıdığı ve sınıflandırma için daha önemli olduğu sonucuna varılmaktadır.

c. BOW ve TF-IDF vektörleri ayrı ayrı olacak şekilde çalışmada kullanılan sekiz farklı algoritma yöntemi ile sınıflandırılmıştır. Sınıflandırma sonuçları karşılaştırıldığında; TF-IDF vektörünün 5 adet sınıflandırma yönteminde, BOW vektörünün ise 3 adet sınıflandırma yönteminde daha başarılı olduğu görülmüştür. Öznitelik seçimi sonrasında elde edilen yeni BOW ve TF-IDF vektörleri bütün algoritmalar ile tekrardan sınıflandırılmıştır. İkinci defa yapılan bu sınıflandırma sonucunda; TF-IDF vektörünün 7 adet sınıflandırma yönteminde daha başarılı olduğu görülürken, BOW vektörünün sadece Karar Ağaçları yönteminde daha başarılı olduğu görülmüştür. Buradan yola çıkılarak veri setinin TF-IDF yöntemi ile daha iyi temsil edildiği sonucuna varılmıştır.

d. Daha önce ayrı ayrı başarı hesaplamaları yapılan BOW ve TF-IDF vektörlerinin birleştirildiğinde hangi yönde başarı sergileyeceğinin hesaplanabilmesi için vektörler birleştirilerek sınıflandırma yapılmıştır. Sınıflandırma sonucunda elde edilen başarı oranları daha önce en yüksek başarı performansını gösteren TF-IDF vektörünün sonuçlarıyla karşılaştırılmıştır. Yapılan karşılaştırmada; birleştirilerek oluşturulan BOW ve TF-IDF vektörünün 6 adet sınıflandırma yöntemi üzerinde daha fazla, Gaussian Naive Bayes yöntemi üzerinde ise eşit oranda başarı sağladığı, TF-IDF vektörünün sadece Multinomal Naive Bayes yöntemi üzerinde daha fazla başarı elde ettiği tespit edilmiştir. Buradan hareketle BOW ve TF-IDF vektörlerinin birleştirilmesi ile elde edilen yeni vektörün TF-IDF vektörüne kıyasla metni daha fazla temsil ettiği sonucuna varılmaktadır.

e. Birleştirilerek mevcut en yüksek başarı oranının elde edildiği BOW ve TF-IDF vektörü Ki-kare öznitelik seçim yöntemi ile önemsiz niteliklerden temizlenmiş ve öznitelik sayısı düşürülmüştür. Öznitelik sayısı yaklaşık olarak üçte bir oranında azaltılan birleştirilmiş vektör çalışmada kullanılan 8 adet sınıflandırma algoritması tarafından sınıflandırılmıştır. Vektörün öznitelik seçim yönteminin öncesinde ve sonrasında göstermiş olduğu sınıflandırma başarıları kıyaslandığında sonuçların hemen hemen aynı

olduğu gözlemlenmiştir. Öznitelik seçim yönteminin birleştirilen vektör verisinin sınıflandırma başarısının arttırılmasında herhangi bir katkısının olmadığı, ancak sınıflandırma süresinin azaltılmasında önemli derecede etkili olduğu görülmüştür.

f. Çalışmada kullanılan KNN sınıflandırma algoritmasının k değeri 2 ile 11 arasında başarı oranının yüksekliğine göre otomatik olarak belirlenmektedir. Yapılan çalışmalar sonucunda k değerinin BOW vektörü sınıflandırılmalarında 4 olarak belirlendiği, TF-IDF vektörü sınıflandırılmalarında ise 10 olarak belirlendiği gözlemlenmiştir. İki vektörün birleştirilmesi durumunda ise k parametresinin TF-IDF vektöründe olduğu gibi 10 olarak belirlendiği görülmektedir. Buradan yola çıkarak, KNN algoritması için belirleyici nitelikteki verinin TF-IDF vektörü olduğu sonucuna varılmaktadır.

g. Hiper parametre yöntemleri, sınıflandırma algoritmalarının başarısını arttırmak için optimum parametre setinin bulunmasını hedeflemektedir. Bu kapsamda çalışmada kullanılan bütün sınıflandırma yöntemleri için hiper parametre analizi yapılmıştır. Elde edilen hiper parametre setleri kullanılarak algoritmalar yeniden eğitilmiş ve başarı oranları tekrardan hesaplanmıştır. Hesaplanan yeni başarı oranları ile birlikte daha önceden hesaplanmış başarı oranları da çizelgeye kaydedilerek sonuçlar kıyaslanmıştır. Yapılan karşılaştırma sonucunda; MLP ve Karar Ağaçları dışında kalan bütün sınıflandırma algoritmalarının başarısını belirli bir oranda arttırdığı görülmektedir. Söz konusu iki sınıflandırma algoritmasının hiper parametre analizi öncesinde başarı oranlarının daha yüksek çıkmasına çapraz doğrulama yönteminin kullanılmamasının ve aşırı öğrenmenin neden olduğu değerlendirilmektedir.

Sonuç olarak; sayısal nitelikli istatistiki verilerin sınıflandırmada etkili olmadığı ve bunun yerine kelime köklerinin sınıflandırmada belirleyici rol oynadığı, TF-IDF yönteminin BOW yöntemine nazaran veri setini daha iyi temsil ettiği ve sınıflandırmada ayırt edici olduğu, sınıflandırmada en başarılı verinin BOW ve TF-IDF vektörlerinin birleşimi sonucunda ortaya çıkan vektör olduğu, öznitelik seçim yönteminin sınıflandırma başarısına fazla katkısının olmadığı ancak zaman maliyeti açısından faydalı olduğu, hiper parametre analizinin sınıflandırma yöntemlerinin başarısını gözle görülür derecede arttırdığı ve son olarak çapraz doğrulama yönteminin sınıflandırma algoritmalarında aşırı öğrenmeyi tespit ettiği gözlemlenmiştir.

Hiper parametre analizi ve çapraz doğrulama işlem adımları sonucunda veri seti üzerinde gerçekleştirilen en başarılı sınıflandırma %91.8 doğruluk, %92 hassasiyet, %92

duyarlılık ve %92 F1-skor oranları ile Lojistik Regresyon, Ki-kare, BOW ve TF-IDF yöntemleri kullanılarak oluşturulan kombinasyon sonucunda elde edilmiştir.

5.2 Öneriler

Bu çalışmada gazetelerin internet sitesinde yer alan köşe yazıları alınarak eser ve yazar eşleştirilmesi yapılmıştır. Veri setinin metin ve yazar sayısı bakımından zenginleştirilmesi sınıflandırma algoritmalarının da başarı oranlarını etkileyecektir. Bu bakımdan çalışmada kullanılan veri seti boyutu arttırılarak değişimlerin gözlemlenmesi faydalı olabilir.

Metin içerisinde genellikle bağlaç olarak kullanılan ve metnin sınıflandırılmasında herhangi bir yarar sağlamayan kelimeler durak kelime olarak adlandırılmaktadır. İnternette Türkçe dili için hazırlanmış olan birçok farklı durak kelimeler listesi bulunmaktadır. Bu çalışmada metin içerisindeki durak kelimelerin tespiti için Zemberek kütüphanesinde yer alan durak kelimeler listesi genişletilerek kullanılmıştır. Bu liste Türkçe dili için birçok veri seti analiz edilerek daha da genişletilebilir. Böylelikle veri seti içerisindeki anlamsız kelimelerin sayısı azalacak, dolayısıyla sınıflandırma başarısı artacaktır.

Sınıflandırma algoritmalarının kategorik verileri işleyebilmesi için geliştirilen birçok faklı metin temsil yöntemi bulunmaktadır. Metin temsil yöntemleri kategorik verileri sayısallaştırarak sınıflandırma için uygun formata dönüştürmektedir. Bunu yaparken kendilerine özgü teknikler kullandıklarından farklı başarı sonuçları elde etmektedirler. Bu çalışmada metin temsil yöntemleri arasında en çok bilinenlerden BOW ve TF-IDF vektör temsil yöntemleri kullanılmıştır. Kullanılan bu geleneksel yöntemlere literatürde bulunan farklı metin temsil yöntemleri eklenerek sınıflandırma başarısının hangi yönde değiştiği gözlemlenebilir ve incelemeler sonucunda çalışmaya yeni yorumlamalar katılabilir.

Metin tabanlı sınıflandırma uygulamalarında her bir kelime aslında bir özniteliği simgelemektedir. Bu nedenle önemsiz niteliklerin veri setinden çıkarılması sınıflandırma başarısının artmasına, veri boyutunun ve sınıflandırma süresinin azalmasına neden olacaktır. Literatürde önemsiz verilerin tespit edilmesinde kullanılan birçok öznitelik seçim yöntemi bulunmaktadır. Bu yöntemlerin çalışma biçimine ve sistematiğine bağlı olarak birbirlerine karşı sağlamış oldukları avantaj ve dezavantajları mevcuttur. Bu

çalışmada öznitelik seçim yöntemi olarak Ki-kare ve Korelasyon Matrisi kullanılmış olup daha fazla öznitelik seçim yöntemi kullanılarak farklı sonuçlar elde edilebilir.

KAYNAKLAR

Adalı, E., 2012, Doğal dil işleme, *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 5 (2).

Agnihotri, D., Verma, K. ve Tripathi, P., 2017, Variable global feature selection scheme for automatic classification of text documents, *Expert systems with Applications*, 81, 268-281.

Akar, Ö. ve Güngör, O., 2012, Rastgele orman algoritması kullanılarak çok bantlı görüntülerin sınıflandırılması, *Jeodezi ve Jeoinformasyon Dergisi* (106), 139-146.

Akın, A. A. ve Akın, M. D., 2007, Zemberek, an open source NLP framework for Turkic languages, *Structure*, 10 (2007), 1-5.

Akın, A. A., 2019, Zemberek kütüphanesi stopwords kelimeler, Web adresi: https://github.com/ahmetaa/zemberek-nlp/blob/master/experiment/src/main/resources/stop-words.tr.txt [Ziyaret Tarihi: 26.03.2023]:

Aksoy, A., 2022, Augmentıng a Turkısh dataset for spam fılterıng usıng natural language processıng technıques, *Middle East Technical University*.

Aksoy, N., 2021, Türkçe dilinde yapılmış açık uçlu sınavların doğal dil işleme ile otomatik olarak değerlendirilmesi, *Balıkesir Üniversitesi Fen Bilimleri Enstitüsü*.

Albayrak, N. B., 2011, Doğal dil işleme teknikleri kullanarak görüş ve duygu analizi, *Fatih Üniversitesi*.

Alı, S. H. K., 2022, Examının effect of data preprocessıng to market basket analyze by usıng aprıorı algorıtm ın WEKA.

Alkış, N., 2016, Bayes Yapısal Eşitlik Modellemesi: kavramlar ve genel bakış, *Gazi İktisat ve İşletme Dergisi*, 2 (3), 105-116.

Altan, S. N., 2018, Metin sınıflandırma için makine öğrenmesi tekniklerine dayalı bir yöntem geliştirme, *Ege Üniversitesi, Fen Bilimleri Enstitüsü*.

Altunkaynak, A., Başakın, E. E. ve Kartal, E., 2020, Dalgacik K-En Yakin Komşuluk Yöntemi İle Hava Kirliliği Tahmini, *Uludağ Üniversitesi Mühendislik Fakültesi Dergisi*, 25 (3), 1547-1556.

Aray, B., 2020, Aradiller ve Aradilbilim, *Trakya Üniversitesi Edebiyat Fakültesi Dergisi*, 10 (19), 127-146.

Ataseven, B., 2013, Yapay sinir ağlari ile öngörü modellemesi, *Öneri Dergisi*, 10 (39), 101-115.

Başboğa, S., 2010, Anadolu Ajansı (1920-1922), *Atatürk İlkeleri ve İnkılap Tarihi Enstitüsü*.

Bengi, S. H., 2019, Ülkelerin Bağımsızlık Mücadeleleri ve Haber Ajansları İlişkisi–Anadolu Ajansı Örneği, *Atatürk Araştırma Merkezi Dergisi*, 35 (100), 449-478.

Bilgin, M., 2017, Gerçek veri setlerinde klasik makine öğrenmesi yöntemlerinin performans analizi, *Breast*, 2 (9), 683.

Bland, J. M. ve Altman, D. G., 1995, Calculating correlation coefficients with repeated observations: Part 2—Correlation between subjects, *Bmj*, 310 (6980), 633.

Bonissone, P., Cadenas, J. M., Garrido, M. C. ve Díaz-Valladares, R. A., 2010, A fuzzy random forest, *International Journal of Approximate Reasoning*, 51 (7), 729-747.

Chomsky, N., 2014, The minimalist program, MIT press, p.

Christian, H., Agus, M. P. ve Suhartono, D., 2016, Single document automatic text summarization using term frequency-inverse document frequency (TF-IDF), *ComTech: Computer, Mathematics and Engineering Applications*, 7 (4), 285-294.

Çelikkaya, G., 2015, Doğal Dil İşleme Teknikleri Kullanılarak Türkçe Mobil Asistan Yazılımı Geliştirilmesi, *Fen Bilimleri Enstitüsü*.

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

Çoban, Ö., 2016, Metin sınıflandırma teknikleri ile türkçe twitter duygu analizi, *Yüksek Lisans Tezi, Atatürk Üniversitesi, Fen Bilimleri Enstitüs*.

Dalkılıç, H. ve Dalkılıç, F., 2015, Karar ağaçları destekli vadeli mevduat analizi, *Akademik Bilişim*.

Das, M., Balpetek, N., Akpinar, E. ve Akpinar, S., 2019, Investigation of wind energy potential of different provinces found in Turkey and establishment of predictive model using support vector machine regression with the obtained results, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 34 (4).

Dilki, G. ve Başar, Ö. D., 2020, İşletmelerin iflas tahmininde K-en yakın komşu algoritması üzerinden uzaklık ölçütlerinin karşılaştırılması, *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, 19 (38), 224-233.

Diri, B., 2020, Doğal Dil İşleme (DDİ) Natural Language Processing (NLP).

Dondurmacı, G. A. ve Çınar, A., 2014, Finans sektöründe veri madenciliği uygulaması, *Akademik Sosyal Araştırmalar Dergisi*, 2 (1), 258-271.

Ekin, M. F., 2020, Türkçe morfolojik analiz için yeni bir yöntem, *Maltepe Üniversitesi, Lisansüstü Eğitim Enstitüsü*.

Erdal, H. ve Yapraklı, T. Ş., 2016, Firma başarısızlığı tahminlemesi: Makine öğrenmesine dayalı bir uygulama, *Bilişim Teknolojileri Dergisi*, 9 (1), 21.

Erkaya, A. E., 2019, Text classification based on organizational data using machine learning, *Fen Bilimleri Enstitüsü*.

Famili, A., Shen, W.-M., Weber, R. ve Simoudis, E., 1997, Data preprocessing and intelligent data analysis, *Intelligent data analysis*, 1 (1), 3-23.

Franko, S., 2018, Multiclass analysis of automatic text classification techniques, *Fen Bilimleri Enstitüsü*.

Guida, G. ve Mauri, G., 1986, Evaluation of natural language processing systems: Issues and approaches, *Proceedings of the IEEE*, 74 (7), 1026-1035.

Gültepe, Y., 2019, Makine öğrenmesi algoritmaları ile hava kirliliği tahmini üzerine karşılaştırmalı bir değerlendirme, *Avrupa Bilim ve Teknoloji Dergisi* (16), 8-15.

Gündoğdu, Ö. E. ve Duru, N., 2016, Türkçe Metin Özetlemede Kullanılan Yöntemler, *18. Akademik Bilişim Konferansı-AB'16*.

Han, J. ve Moraga, C., 1995, The influence of the sigmoid function parameters on the speed of backpropagation learning, *International workshop on artificial neural networks*, 195-201.

İşeri, İ. ve Arıman, S., 2019, Sedimandaki Ağır Metal Konsantrasyonunun Çoklu Değişken Regresyon Modelleri ve Çok Katmanlı Algılayıcı Ağ Modeli ile Tahmini, *Avrupa Bilim ve Teknoloji Dergisi*, 389-397.

İşi, A., Çemrek, F. ve Yıldız, Z., 2013, İstatistik'ten Edebiyat'a Bir Köprü: Stilometri Analizi, *Uşak Üniversitesi Sosyal Bilimler Dergisi*, 6 (3), 271-281.

Jayalakshmi, T. ve Santhakumaran, A., 2011, Statistical normalization and back propagation for classification, *International Journal of Computer Theory and Engineering*, 3 (1), 1793-8201.

Jin, D., 2021, Image Information Collection System Based on Python Web Crawler Technology, *CONVERTER*, 606-612.

Karasoy, O., 2019, Makine öğrenmesi yöntemleri ile içerik tabanlı SMS filtreleme uygulaması geliştirilmesi, *Muğla Sıtkı Koçman Üniversitesi*.

Kavzoğlu, T. ve Çölkesen, İ., 2010, Destek vektör makineleri ile uydu görüntülerinin sınıflandırılmasında kernel fonksiyonlarının etkilerinin incelenmesi, *Harita Dergisi*, 144 (7), 73-82.

Kaya, H., 2016, Gül ile Bülbülün Aşkı, Web adresi: https://www.antoloji.com/gul-ile-bulbulun-aski-2-siiri/ [Ziyaret Tarihi: 15.01.2023]:

Kaya, S., 2018, Doğal dil işleme teknikleriyle yazar-kitap tanıma, *İstanbul Aydın Üniversitesi*.

Kim, S.-B., Han, K.-S., Rim, H.-C. ve Myaeng, S. H., 2006, Some effective techniques for naive bayes text classification, *IEEE transactions on knowledge and data engineering*, 18 (11), 1457-1466.

Korkmaz, S., 2021, Makine öğrenme yöntemleri kullanılarak doğal dil işleme tabanlı şair tanıma, *Erciyes Üniversitesi*.

Köseoğlu, K., 2008, Veritabanı mantığı, Pusula, p.

Kul, S., 2020, Türkçe Ders Anlatan Yapay Zekaya Giden Yolda Doğal Dil İşleme, *Yönetim Bilişim Sistemleri Dergisi*, 6 (2), 43-56.

Manning, C., Raghavan, P. ve Schütze, H., 2010, Introduction to information retrieval, *Natural Language Engineering*, 16 (1), 100-103.

Maszczyk, T. ve Duch, W., 2008, Comparison of Shannon, Renyi and Tsallis entropy used in decision trees, *Artificial Intelligence and Soft Computing–ICAISC 2008: 9th International Conference Zakopane, Poland, June 22-26, 2008 Proceedings 9*, 643-651.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. ve Dean, J., 2013, Distributed representations of words and phrases and their compositionality, *Advances in neural information processing systems*, 26.

Mumcuoğlu, E., 2022, Prediction of Outcomes in Higher Courts of Turkey Using Natural Language Processing, *Bilkent Universitesi (Turkey)*.

Munkhdalai, L., Munkhdalai, T., Park, K. H., Lee, H. G., Li, M. ve Ryu, K. H., 2019, Mixture of activation functions with extended min-max normalization for forex market prediction, *IEEE Access*, 7, 183680-183691.

Namlı, E., Ünlü, R. ve Gül, E., 2019, Fiyat tahminlemesinde makine öğrenmesi teknikleri ve doğrusal regresyon yöntemlerinin kıyaslanması; Türkiye'de satılan ikinci el araç fiyatlarının tahminlenmesine yönelik bir vaka çalışması, *Konya Journal of Engineering Sciences*, 7 (4), 806-821.

Oflazer, K., 2016, Türkçe ve Doğal Dil İşleme, *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 5 (2).

Oğuzlar, A., 2003, Veri ön işleme, *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi* (21).

Osuna, E., Freund, R. ve Girosi, F., 1997, An improved training algorithm for support vector machines, *Neural networks for signal processing VII. Proceedings of the 1997 IEEE signal processing society workshop*, 276-285.

Ozyurt, O. ve Kose, C., 2006, Turkce tabanli diyalog sistemi tasarimi ve kodlanmasi, *Elektrik, Elektronik ve Bilgisayar Muhendisliği Sempozyumu ve Fuari (ELECO 2006)*, 364-368.

Özbilen, E., 2011, Improving text categorization performance by combining feature selection methods, *Master of Science, Boğaziçi University, Istanbul*.

Özbilici, A., 2006, Türkçe doğal dili anlamada ilişkisel ayrık bilgiler modeli ve uygulaması, *Sakarya Universitesi (Turkey)*.

Özker, U., 2019, Zemberek - Doğal Dil İşleme, Web adresi: https://ugurozker.medium.com/zemberek-nlp-7add032881e9 [Ziyaret Tarihi: 09.02.2023]:

Özmutlu, A. E., 2021, Doğal dil işleme, *Bilgisayar Bilimlerinde Teorik Ve Uygulamalı Araştırmalar*, 129.

Öztürk, E., 2019, Türkçe dili için metin sınıflandırma kullanarak siber zorbalık tespiti, *Çukurova Üniversitesi*.

Parlak, B., 2021, Metin sınıflandırma için öznitelik seçimi ve globalleştirmenin etkisi, *Eskişehir Teknik Üniversitesi*.

Parlak, B. ve Uysal, A. K., 2021, The effects of globalisation techniques on feature selection for text classification, *Journal of Information Science*, 47 (6), 727-739.

Peterson, L. E., 2009, K-nearest neighbor, *Scholarpedia*, 4 (2), 1883.

Pilavcılar, İ. F., 2007, Metin madenciliği ile metin sınıflandırma, *Yıldız Teknik Üniversitesi*.

Rueping, S., 2010, SVM classifier estimation from group probabilities, *ICML*.

Saud, M. S., 2022, Linguistic hybridity: The use of code mixing in Nepali folk pop songs, *Journal of NELTA Gandaki*, 5 (1-2), 43-54.

Seker, S. E., 2015, Doğal Dil İşleme (Natural Language Processing), *YBS Ansiklopedi*, 2 (4), 14-31.

Shiffler, R. E., 1988, Maximum Z scores and outliers, *The American Statistician*, 42 (1), 79-80.

Soyuyüce, E., Hünkar, T. ve Tabanlıoğlu, S., 2003, Veri Tabanı Nedir? Veri Tabanının Oluşum Süreci, *Sağlık Bilimlerinde Süreli Yayıncılık Ulusal Sempozyumu*.

Şahin, D. Ö., 2016, Metin sınıflandırmada öznitelik seçimi üzerine bir çalışma, *Ondokuz Mayıs Üniversitesi, Fen Bilimleri Enstitüsü*.

Talan, T. ve Aktürk, C., 2021, Bilgisayar Bilimlerinde Teorik Ve Uygulamalı Araştırmalar, p.

Tasci, S. ve Gungor, T., 2008, An evaluation of existing and new feature selection metrics in text categorization, *2008 23rd International Symposium on Computer and Information Sciences*, 1-6.

Taşcı, E. ve Onan, A., 2016, K-en yakın komşu algoritması parametrelerinin sınıflandırma performansı üzerine etkisinin incelenmesi, *Akademik Bilişim*, 1 (1), 4-18.

Taşkıran, S. F., 2021, Doğal dil işleme ile akademik metinlerin kümelenmesi, *Konya Teknik Üniversitesi*.

Tiryaki, H. ve Uysal, A. K., 2023, Dengesiz Metin Sınıflandırmada Öznitelik Seçim Yöntemlerinin Etkililiği, *Afyon Kocatepe Üniversitesi Fen Ve Mühendislik Bilimleri Dergisi*, 23 (2), 370-379.

Topuz, S., 2021, Eğitsel Verilerde Weka ve Orange Veri Madenciliği Yazılımlarından Elde Edilen Analiz Sonuçlarının Karşılaştırılması, *Hacettepe Üniversitesi*

Toraman, Ç., 2011, Text categorization and ensemble pruning in Turkish news portals, *Bilkent Universitesi (Turkey)*.

Tosun, İ. B., 2021, Makine öğrenmesi ile metin sınıflandırma: Bakım yönetim sistemi örneği, *Sakarya Üniversitesi*.

Uludogan, G., Özçelik, R., Parlar, S., Ercan, G. ve Yıldız, O. T., 2017, Türkçe Dogal DilIsleme için Arayüzler User Interfaces for Turkish Natural Language Processing, *IEEE*.

Uysal, A. K. ve Gunal, S., 2012, A novel probabilistic feature selection method for text classification, *Knowledge-Based Systems*, 36, 226-235.

Uysal, A. K. ve Gunal, S., 2014, The impact of preprocessing on text classification, *Information processing & management*, 50 (1), 104-112.

Uysal, A. K., 2015, New approaches to enhancing the performance on text classification, *Anadolu University (Turkey)*.

Uysal, A. K., 2016, An improved global feature selection scheme for text classification, *Expert systems with Applications*, 43, 82-92.

Vergili, M. ve Orhan, H., 2022, Genomik Veri Setlerinin LASSO ve Elastik Net Regresyon Yöntemleri ile Analizi, *Süleyman Demirel Üniversitesi Sağlık Bilimleri Dergisi*, 13 (3), 485-496.

Visa, S., Ramsay, B., Ralescu, A. L. ve Van Der Knaap, E., 2011, Confusion matrix-based feature selection, *Maics*, 710 (1), 120-127.

Weizenbaum, J., 1966, ELIZA—a computer program for the study of natural language communication between man and machine, *Communications of the ACM*, 9 (1), 36-45.

Yang, Y. ve Pedersen, J. O., 1997, A comparative study on feature selection in text categorization, *Icml*, 35.

Yavuz, S. ve Deveci, M., 2012, İstatiksel Normalizasyon Tekniklerinin Yapay Sinir Ağin Performansina Etkisi, *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi* (40), 167-187.

Yılmaz, H. ve Yumuşak, S., 2021, Açık kaynak doğal dil işleme kütüphaneleri, *İstanbul Sabahattin Zaim Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 3 (1), 81-85.