Makine Öğrenmesi Yaklaşımlarının Spam-Mail Sınıflandırma Probleminde Karşılaştırmalı Analizi

Araştırma Makalesi/Research Article

Nuriye BAKTIR, Yılmaz ATAY

Bilgisayar Mühendisliği, Mühendislik Fakültesi, Gazi Üniversitesi, Ankara, Türkiye
nuriyebaktir@gmail.com, yilmazatay@gazi.edu.tr
(Geliş/Received:16.03.2022; Kabul/Accepted:20.07.2022)
DOI: 10.17671/gazibtd.1014764

Özet— Elektronik posta, kuruluşların, kişilerin sıklıkla kullandıkları dosya paylaşımı gibi çeşitli etkileşimlerin bulunduğu iletişim aracıdır. Bu tür araçların faydalı etkilerinin yanında istenmeyen elektronik posta paylaşımı da söz konusudur. İstenmeyen elektronik postalar 'Spam' adı ile etiketlenmektedir. Spam elektronik postalar; istenmeyen reklamlar, virüs etkileşimleri ve oltalama gibi zararlı içeriklere kaynak teşkil edebilmektedir. İletişimde güvenliğin oldukça önemli olduğu bilinmektedir. Bu sebeple elektronik posta sistemlerinin zararlı araçlardan veya yazılımlardan arındırılabilmesi için çeşitli kriterlere göre sınıflandırılması önem arz etmektedir. Literatürde bu tür çalışmalar farklı başlıklar altında sunulmaktadır. Sınıflandırma çalışmalarında makine öğrenmesi algoritmaları etkin bir şekilde kullanılmaktadır. Bu çalışma kapsamında naive bayes, lojistik regresyon, karar ağacı ve k-en yakın komşu algoritmalarının ilgili probleme uyarlanması ve karşılaştırmalı olarak analiz edilmesi amaçlanmıştır. Burada farklı metodolojilere sahip yaklaşımların ilgili problem üzerindeki etkisi detaylı olarak incelenmek istenmiştir. Bu kapsamda algoritmalar çeşitli veri setleri kullanılmıştır. Veri setlerinin farklı büyüklüklerde ve farklı ham/spam oranlarında olması çalışma üzerindeki etkisi tartışılmıştır. Farklı başarım sonuçları elde edilmiştir. Bu başarım sonuçlarının farklı metotlara göre karşılaştırması yapılarak tablolar halinde sunulmuştur. Veri seti sayısının ve spam oranının fazla olması Enron 5 veri setinde etkili sonuçların elde edilmesini sağlamıştır. Farklı özellik seçim yöntemlerinin kullanımıyla Karar ağacı algoritmasının Enron 4 veri seti üzerinde iyi performans göstermesini sağlamıştır. En iyi başarım performanslarının CS440/ECE448 veri seti üzerindeki testlere göre lojistik regresyon ve k-en yakın komşu algoritmalarıyla elde edildiği gözlemlenmiştir.

Anahtar Kelimeler—karar ağacı, k-en yakın komşu, lojistik regresyon, naive bayes, sınıflandırma, spam e-posta

Comparative Analysis of Machine Learning Approaches in the Spam-Mail Classification Problem

Abstract— Electronic mail is a communication tool where organizations and people frequently use various interactions such as file sharing. In addition to the beneficial effects of such tools, there is also the sharing of spam e-mail. Unwanted e-mails are labeled as 'Spam'. Spam emails; It can be a source of harmful content such as unwanted advertisements, virus interactions and phishing. It is known that security is very important in communication. For this reason, it is important to classify e-mail systems according to various criteria in order to be free from harmful tools or software. Such studies are presented under different headings in the literature. Machine learning algorithms are used effectively in classification studies. In this study, it is aimed to adapt naive bayes, logistic regression, decision tree and k-nearest neighbor algorithms to the related problem and analyze them comparatively. Here, the effect of approaches with different methodologies on the related problem is tried to be examined in detail. In this context, algorithms have been used in various data sets. The effect of datasets of different sizes and different raw/spam ratios on the study is discussed. Different performance results have been obtained. These performance results were compared according to different methods and presented in tables. The high number of datasets and spam rate provided effective results in the Enron 5 dataset. By using different feature selection methods, Decision tree algorithm performed well on Enron 4 dataset. It has been observed that the best performance performances are obtained with logistic regression and k-nearest neighbor algorithms according to the tests on the CS440/ECE448 dataset.

Keywords— decision tree, k-nearest neighbor, logistic regression, naive bayes, classification, spam e-mail

1. GİRİŞ (INTRODUCTION)

İnternet teknolojisinin gelişimiyle elektronik posta uygulamalarının mesaj ve dosya paylaşımı amaçlarıyla kullanımı artmıştır. Kullanılabilirliğinin kolay olması, hızlı mesaj alışverişi sağlayabilmesi, düşük iletim maliyetlerine sahip olması ve benzeri sebeplerle sıklıkla kullanılmaktadır. E-posta uygulamaları faydalı etkilerinin yanında kötü niyetli kişilerinde hedefinde olmuştur. Şüpheli mesajlar ya da kötü niyetli eklentiler sıklıkla karşılaşılan problemlerdendir. Spam e-posta olarak tanımlanan istenmeyen elektronik postalar kullanıcıya ya önemsiz bir mesaj olduğunu ya da gönderen tarafından çok sayıda alıcıya ulaştırılan şüpheli içeriğe sahip e-posta olduğunu göstermektedir. Bunun gibi şüpheli içeriğe sahip e-postalardaki bağlantılara giriş yapılması alıcıları kimlik avı web adreslerine yönlendirebilmekte veya kurbanın bilgisayarına art niyetli yazılımlara yönlendirebilmektedir [1, 2]. Teknik alt yapı tarafında da zararları mevcuttur. Örneğin ağ bant genişliği, CPU veya disk kapasitesini gereksiz yere meşgul edebilmektedir [3]. Bu mesajlara bakılması ve sonrasında yapılan işlemler kullanıcıların zamanını alabilmektedir.

Spam e-postaların sınıflandırılması ve kullanıcıya iletiminin engellenmesi önemli hale gelmiştir. Bu konuyla ilgili çalışmalar yapılmıştır ve yapılmaya devam etmektedir. Sınıflandırma probleminde kullanılmış olan naive bayes (naive bayes - NB), karar ağacı (decision tree - DT), lojistik regresyon (logistic regression - LR) ve k-en yakın komşu (k-nearest neighbors - KNN) algoritmaları incelenmiştir. E-posta sınıflandırma probleminin metin sınıflandırmaya dayanan bir geçmişi vardır [4]. Probleme uyarlanmış olan bu algoritmaların klasik çalışmalara dayanarak metin sınıflandırma veya içerik kategorize etme şeklinde çalışmalar mevcuttur. Naive bayes yaklaşımı yardımıyla 1992 yılında metin kategorize etme çalışmaları [5] yapılmış olup; 1994'te lojistik regresyon ile naive bayes yaklaşımlarının sonuçlarını karşılaştıran metin sınıflandırma çalışmaları devam etmiştir [6]. Quinlan tarafından yapılan çalışmada, karar ağaçlarının kullanılması ile elde edilen modellerden yararlanılarak sınıflandırma çalışması yapılmıştır [7]. Bir diğer algoritma olan KNN ile klasik sınıflandırma çalışmaları 1967 yılında başlanmış olup [8] ve sonraki yıllarda algoritmaların etkin bir şekilde kullanıldığı metin sınıflandırma konusu önemli bir çalışma alanı sağlamıştır.

İlgili çalışma kapsamında bahsedilen algoritmaların örnek uygulamaları incelenmiştir. Bu çalışmada dört farklı makine öğrenmesi yöntemi spam e-posta sınıflandırma problemine uyarlanmıştır. Çeşitli kaynaklardan elde edilen veri setleri bu çalışmadaki test süreçlerinde etkin bir şekilde kullanılmıştır. Deneyler beş farklı veri setine göre gerçekleştirilmiştir. Bu veri setleri öncelikli olarak standart hale getirilerek ortak bir çatı altında toplanmış ve bu standartlaştırma sürecinde farklı ön işlem adımları uygulanmıştır. Bu çalışmada doğruluk, kesinlik, duyarlılık ve F1-skor değerlendirme metrikleriyle sonuçlar tablo halinde sunulmuştur. Veri setlerinin metriklere göre en iyi çalışan algoritmanın ne kadar başarılı sonuç verdiği tablo üzerinde gösterilmiştir.

Bu çalışmanın amacı spam e-posta sınıflandırmasında veri setlerinin çeşitliliğine, büyüklüğüne ve ham/spam oranlarının farklılığına göre farklı algoritmaların çalışmasını incelemektir. Sınıflandırmanın doğru bir şekilde yapılıp zararlı durumların kullanıcı dijital platformlarına bulaşmasını engellemektir. Böylelikle kullanıcının spam etiketi ile gelen e-postayı görmezden gelerek hem zamandan kazanım sağlanmasına hem de zararlı faaliyetlerle uğraşmasının önüne geçilmesi hedeflenmiştir.

2. LİTERATÜR TARAMASI (LITERATURE REVIEW)

Bu çalışmada sınıflandırma amacıyla kullanılan NB, LR, DT ve KNN algoritmalarının öncelikle literatür araştırması yapılmıştır. Bunlardan NB algoritması yaygın olarak kullanılan denetimli öğrenme yöntemlerindendir. Naive bayes, gerekli ön bilgi ve koşullara dayalı olarak bir olayın meydana gelme olasılığını üretmeye çalışan bayes temelli yaklaşımdır [9]. Bu yaklaşımın ölçeklenebilir olması sayesinde herhangi bir probleme hızlı bir şekilde uygulanabilmesi mümkündür. Az sayıda veriye sahip örneklerde iyi performanslar sergilediği bilinmektedir [10-12]. Raza vd. [13] NB ile farklı derin öğrenme yöntemlerini kullanarak karşılaştırmalı bir çalışma yapmışlardır. Junnarkar vd. [14] ise yapmış oldukları çalışmada doğal dil işleme (natural language processing-NLP) ve tekdüzen kaynak bulucu (uniform resource locators-URL) tabanlı filtreleme yöntemlerini kullanarak derin öğrenme algoritmalarının veri seti üzerinde çalışma performansını incelemişlerdir. Burada sınıflandırmanın doğruluğunu artırabilmek amacıyla Aslantaş vd. [15] önerdikleri çalışmada, ikili ateşböceği (binary firefly algorithm-BFA) ve NB algoritmaları kullanılarak hibrit bir çalışma gerçekleştirmişlerdir. Elde edilen doğruluğa bağlı olarak performansın iyi olduğu sonucuna varılmış. Nayak vd. [16] NB ve J48 algoritmalarının hibrit torbalama yöntemini kullanarak korelasyon özellik seçimi (correlation feature selection-CFS) yöntemiyle sınıflandırma çalışması yapmışlardır. Hibrit çalışma ile NB algoritmasının aynı sonucu verdiği gözlemlenmiş. Başka bir özellik seçim yöntemi olarak kullanılan TF-IDF (term frequency-inverse document frequency) [17] fonksiyonu ile Enron [18], Ling-Spam [19] ve PU [20] veri setleri kullanılarak NB yöntemi üzerinde veri seti çeşitliliğinin önemi tartışılmış [21]. Gibson vd. [22], parçacık sürü optimizasyonu (particle swarm optimization-PSO) ve genetik algoritma (genetic algorithm-GA) ile NB yöntemini optimize eden bir yaklaşım önermişlerdir. Yedi adet veri seti kullanılarak yapılan bu çalışmada genetik algoritma temelli multinomial naive bayes genel olarak en iyi performansı sunduğu sonucuna varılmış. Veri seti çeşitliliğinin performansa etkisini inceleyen bir diğer çalışmada ise öznitelik sayısının performansa etkisi araştırılmış [23].

İncelenen algoritmalardan diğeri LR yaklaşımıdır. Lojistik regresyon evet/hayır, doğru/yanlış gibi tahminlerin ikili sınıflandırılması amacıyla kullanılan popüler bir istatistiksel makine öğrenimi algoritmasıdır. Bu çalışmada LR, gelen e-postaları ham ve spam şeklinde sınıflandırabilmek amacıyla uygulanmıştır. Dedekurt vd.

[24] tarafından yapılan çalışmada LR ve yapay arı koloni (artificial bee colony-ABC) algoritmaları TF-IDF yönteminin avantajlarıyla birleştirilerek yeni bir spam filtreleme yaklaşımı önerilmiş. Diğer çalışmalardan farklı olarak Turkish Spam V01 [25], CSDMC 2010 Spam Corpus [26] veri setleri kullanılmış. Önerilen ABC-LR algoritması klasik LR algoritmasından daha iyi sonuç verdiği gözlemlenmiş. Janez-Martino tarafından önerilen TF-IDF ve kelime çantası (bag of words-BOW) [27] yöntemlerinin LR ile kombinasyonunu değerlendirebilmek amacıyla SPEMC-11K (spam email classification) veri seti testlerde kullanılmış [28]. Turkish Spam V01 veri seti kullanan başka bir çalışmada DT algoritması ile sınıflandırma yapılmış [3]. Aynı veri setini kullanılarak yapılan farklı bir çalışmada ayırt edici özellikler elde edildikten sonra belirlenen BOW histogramları üzerinde 3 ayrı algoritmayla sınıflandırma işlemleri yapılmış [29]. Başka bir çalışmada spam e-posta tespit edilirken içeriği analiz edebilmek amacıyla LR algoritması Movie Reviews, CSDMC 2010 Spam Corpus ve TREC 2007 Public Corpus [30] veri setlerine ayrı ayrı uygulanmış [31]. Bassiouni vd. [32] LR algoritmasının da içerisinde olduğu 10 farklı algoritmanın Spambase [33] test veri seti üzerinde karşılaştırmalı analiz yapmışlardır. Doğruluğu sağlamak amacıyla 10 kat çapraz doğrulama kullanılmış. Özellik seçimi aşamasında sonsuz gizli özellik seçme (infinite latent feature selection-ILFS) yaklaşımı kullanılmış.

Literatürde etkin bir şekilde kullanılan diğer yaklaşım DT algoritmalarıdır. Yapılan çalışmada kullanılan karar ağacı algoritması probleme uyarlanmasının kolay olması, açıklamalara ve görselleştirmelere uygun olması açısından sıklıkla tercih edilmektedir. Bu algoritma farklı boyuttaki veri setleriyle birlikte rahatlıkla kullanılabilmektedir. Hem numerik hem de kategorik verileri işleme yeteneğine sahip olması da diğer avantajlarındandır [34,35]. İncelenen çalışmalar arasında Spambase veri seti kullanılarak DT algoritması ve diğer klasik makine öğrenimi algoritmalarının çalışması gerçekleştirilmiş [36]. Spambase veri setini içeren diğer bir çalışmada DT algoritmasının da olduğu 10 adet yaklaşım üzerinde değerlendirmeler yapılmış. Doğruluğu sağlayabilmek amacıyla 10'ar kez çalıştırma yapıldığı gözlemlenmiştir [22]. Yağanoğlu vd. [37] tarafından yapılan çalışmada, doğal dil işleme ile ön işlem adımları üzerine çalışılarak DT ve farklı makine öğrenmesi algoritmaları karşılaştırmalı olarak incelenmiş. Veri seti olarak 5574 adet veri içeren SMS Spam Collection Data [38] kullanılmış. Burada en yüksek başarıma DT yöntemi ile ulaşıldığı sonucuna varılmıştır. E-posta sınıflandırma amacıyla yapılan diğer bir çalışmada ise özellik seçimi yöntemiyle karar ağaçları türlerinden biri olan yinelemeli dikotomizer 3 (iterative dichotomiser 3-ID3) yöntemi kullanılmış. Ayrıca burada eğitim ve test veri kümelerinin bölünme oranlarının performansa etkisi de tartışılmış [39].

Son olarak bu başlık altında incelenen yöntem, k-en yakın komşu algoritmasıdır. Bu algoritma, geliştirilen sistem için hem sınıflandırma hem de regresyon çıktıları oluşturmak amacıyla kullanılmaktadır. Bu algoritmanın en büyük dezavantajı, veri setindeki aykırı değerlere karşı oldukça hassas olmasıdır. Bunun dışında bu yöntemin hesaplama maliyetinin de genelde yüksek olduğu belirtilmektedir [35,40]. Spambase veri seti kullanılarak KNN ile biyoloji esinlenmeli yöntemlerin hibrit çalışması gerçekleştirilmiş [41]. Al-Tahrawi vd. [42] e-posta sınıflandırma probleminde, metin sınıflandırmada sıklıkla kullanılan algoritmalarından olan polinomal sinir ağları (polynomial neural networks-PNN) yaklaşımını KNN algoritmasıyla birlikte kullanmışlardır. Ling-Spam veri seti kullanılarak yapılan çalışmada algoritmaların performans değerlendirmesi sunulmuş. Amjad vd. [43] özellik seçim aşamasının sınıflandırma performansına etkisini incelemişler ve dağılım arama algoritması (scatter search algorithm-SSA) ile KNN algoritmalarının hibrit çalışmasını sunmuşlardır. Burada Spambase veri seti kullanılarak yapılan çalışmada hem özellik seçimi ile hem de özellik seçimi yapılmadan gerçekleştirilen deneylerin sonuçları sunulmuş. Burada sınıflandırmada öznitelik seçim işleminin performansa etkisi analiz edilmiş. Daha önce yapılan çalışmalarda yaygın olarak kullanılan sınıflandırıcılardan KNN, NB ve bunun gibi yöntemlerin sonuçlarını optimize etmek ve önerilen spam algılamayı değerlendirebilmek amacıyla hibrit su döngüsü (water cycle feature selection-WCFS) ve benzetimli (simüle) tavlama algoritmaları ile özellik seçiminin performansa etkisi Al-Rawashdeh vd. [44] tarafından analiz edilmiş ve sonuçlar tartışılmış.

3. METOT VE MATERYALLER (METHOD AND MATERIALS)

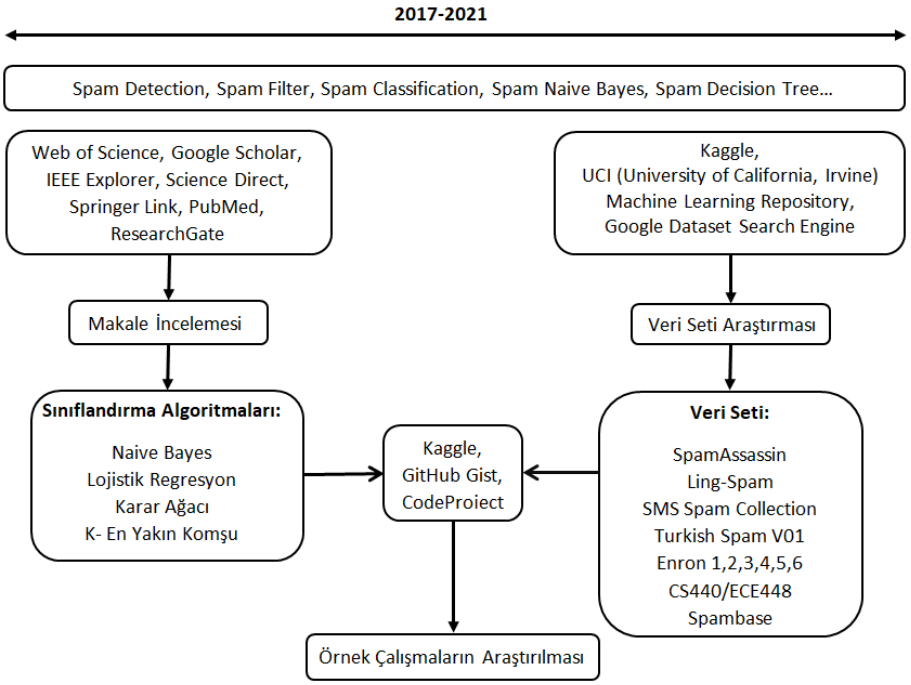
Bu bölümde araştırma metodolojisi, kullanılan yöntemler ve test verileri hakkında özet bilgiler verilmiştir. Şekil 1'de gösterilen araştırma metodolojisinde son yıllara ait çalışmalar gösterilmektedir. Spam e-posta sınıflandırma problemi geniş bir konu yelpazesine sahip olduğu için metin madenciliği ve içerik analizi gibi farklı alanlardan da faydalanılmıştır. Sınıflandırmanın yanında filtreleme, özellik seçimine ya da veri seti çeşidine göre araştırmalara devam edilmiştir. Çalışma içerisinde kullanılmış olan naive bayes, lojistik regresyon, karar ağacı ve k-en yakın komşu algoritmalarının çalışma mantığı, makalelerde hangi algoritmalarla karşılaştırıldığı, başarım durumları, hangi başarım değerlendirme metrikleri ile sonuçların verildiği araştırma konularından bazılarıdır.

Veri seti araştırması yapılırken öncelikli olarak makalelerde kullanılmış olan veri setlerinden kaynakları mevcut olanlara erişim sağlanmış ve özellikleri bu başlık altında sunulmuştur. Veri seti ile ilgili literatür incelemesi sonucunda farklı sayılarda veri kümelerinin kullanıldığı, büyüklüklerinin farklı olduğu, eğitim-test setleri olarak çeşitli ayrımların yapıldığı çalışmalar incelenmiştir. Bu çalışma kapsamında, SpamAssassin, Enron 4-5-6 ve CS440/ECE448 veri setleri kullanılarak veri çeşitliliğin, veri sayısının ve ham/spam oranının sınıflandırma performansına etkisi araştırılmak istenmiştir. Elde edilen veri setlerinin farklı algoritmalarda nasıl sonuçlar verdiği incelenmiştir.

3.1. Veri Seti (Dataset)

Çalışma kapsamında veri seti araştırması yapılırken öncelikli olarak makalelerde kullanılan ve erişimi açık olan veri setleri temin edilmiştir. Erişim sağlanan veri setlerinin dışında e-posta ve SMS (kısa mesaj hizmeti-short message service) veri setleri şeklinde araştırma yapılarak farklı veriler elde edilmiştir. Bazı veri setleri ilgili dokümanlarla beraber web sayfalarından erişim sağlanmıştır [45].

Gerçekleştirilen çalışma içerisinde kullanılmış olan veri setlerinden biri olan SpamAssassin verisi, Apache SpamAssassin web sayfasından temin edilmiştir. Bu veri seti ile yapılmış olan farklı çalışmalarda sunulmaktadır. Burada ham ve spam olarak ayrı klasörlerde sunulmuş olan veriler bu çalışmada birleştirilerek işleme alınmıştır [46].



Şekil 1. Araştırma metodolojisi (Research methodology)

Çalışmalarda sıklıkla kullanılan Enron veri setine Ion Androutsopoulos'un çalışmasından erişilebilmektedir [47]. Enron veri seti içerisinde alt başlıklara ayrılmış 6 adet veri seti bulunmaktadır [48]. Farklı büyüklükteki bu veri setlerinin ham ve spam oranları da birbirinden farklıdır. Metsis vd. [47] tarafından yapılan çalışmada bu altı adet veri seti kullanılarak ham/spam sınıflandırması gerçekleştirilmiş.

CS440/ECE448, Enron 2 veri setinden elde edilen bir veri setidir. Bu veriye, kendisine kaynak olan web sayfasından erişim sağlanmıştır [49]. Burada veri seti ile ilgili özellikler sunulmuş olup Naive bayes ile yapılan çalışma hakkında CS440/ECE448 veri seti incelenmiş.

Tablo 1'de kullanılmış olan veri setlerinin özellikleri sunulmuştur. Veri seti içerisinde kaç adet ham ve spam verilerinin bulunduğu gösterilmiştir. Bu niceliklere göre spam oranları yazılmıştır. Tablodaki veriler veri setlerinin yayınlanma tarihine göre sıralanmıştır.

SpamAssassin ve Enron 4-5-6 veri setlerinin boyutları birbirine yakın sayıdadır. Enron 4-5-6 veri setlerinde ham/spam oranında spam veri sayısı daha fazladır. SpamAssassin veri setinde ham veriler daha fazladır. Enron 2 veri setinden elde edilen CS440/ECE448 veri seti veri sayısı diğerlerinden daha azdır ve ham/spam oranı eşit olarak bölünmüştür. Çalışma içerisinde bu farklılıkların sınıflandırmaya etkisi incelenmiştir.

Tablo 1. Veri setleri (Datasets)

Veri Seti Adı	Veri Seti Bilgisi			
	Referans	Ham + Spam	Spam oranı	Yayın tarihi
SpamAssassin	[56]	4150 + 1897 = 6047	%31	2002
Enron 4	[45]	1500 + 4500 = 6000	%75	2006
Enron 5	[45]	1500 + 3675 = 5175	%71	2006
Enron 6	[45]	1500 + 4500 = 6000	%75	2006
CS440/ECE448	[25]	465 + 465 = 930	%50	2018

3.2. Veri Setlerinin Standart Hale Getirilmesi (Standardizing Datasets)

Çalışma kapsamında temin edilen veri setlerinin ham haliyle kullanılmasındaki karmaşıklıklar sebebiyle tüm veriler standart hale getirilmiştir. Bu aşamada aşağıdaki işlem adımları uygulanmıştır.

- İlk olarak başlıklar değiştirilerek mail içeriğinin bulunduğu sütunun adı '*mail*'; ilgili epostanın ham/spam olduğunu gösteren sütunun adı '*label*' olarak değiştirilmiştir.
- Çalışmada kullanılmayacak olan sütunlar silinmiştir.
- Sütunların sıralaması 'label' ve 'mail' şeklinde düzenlenmiştir.
- 'Dataset' ismiyle oluşturulan dizin 'dataset1' şeklinde kaydedilmiştir.

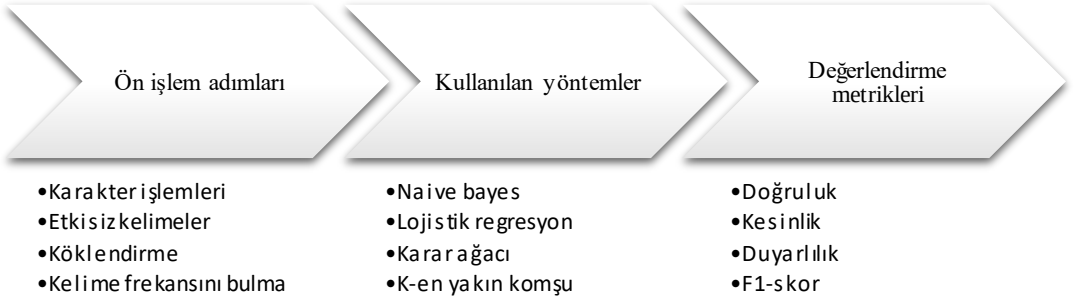
Bazı veriler eğitim ve test veri setleri olarak ayrı klasörler halinde tutulmuştur. Bu klasörlerin içerisinde ham ve spam verileri de ayrı klasörlerde sunulmuştur. Bu veri setleri spam ve ham olarak ayrı ayrı okunmuştur. Okuma işleminden sonra birleştirme işlemi gerçekleştirilmiş ve bu veri setlerinin aynı kriterlere göre ayrılabilmesi için bu işlemler tekrarlanmıştır. Daha sonra standart hale getirme işlem adımları uygulanmıştır. Bu dosyalar .txt uzantısıyla sunulmuş olup tek bir dizin içerisinde .csv dosya uzantısıyla kayıt altına alınmıştır. Enron 4, 5, 6 ve CS440/ECE448 veri setleri üzerinde bu işlem adımları gerçekleştirilmiştir. 'dataset2', 'dataset3', 'dataset4' ve 'dataset5' şeklinde kayıt edilerek işlem kolaylığı sağlanmıştır.

4. DENEYSEL ÇALIŞMALAR (EXPERIMENTAL STUDIES)

Bu bölümde seçilen sınıflandırıcıların gerçek test verileri üzerindeki deneysel çalışmaları, elde edilen test sonuçları ve analiz çıktıları hakkında genel bilgiler verilmiştir. Ayrıca veri setlerinin test verileri olarak kullanılabilmesi için yapılmış olan birtakım ön işlem adımları da açıklanmıştır. Bahsi geçen adımlar: karakter işlemleri, etkisiz kelimeler, köklendirme ve belirteçleştirme süreçleridir. Ele alınan algoritmalar, probleme uyarlanarak beş veri seti üzerinde test edilmiştir. Deneysel sonuçlar tablolar halinde doğruluk (accuracy), kesinlik (precision), duyarlılık (recall) ve F1-skor (F1-score) değerlendirme metrikleriyle sunulmuştur.

Çalışma içerisinde her bir algoritma üzerinde her bir veri seti 10 tekrarla çalıştırılmıştır. Kullanılan sayısal değerler değiştirilmeden çalıştırılmıştır. Tekrarlı çalışmanın yapılmasındaki amaç uygulama içerisinde yüksek standart sapmaların olup olmadığını izlemektir. Elde edilen sonuçlar Tablo 2-5 üzerinde gösterilmiştir. Yüksek standart sapmalar elde edilmemiştir. Bu da çalışmanın iyi performansla çalıştığını göstermiştir. Test sonuçlarında en iyi, en kötü, ortalama, medyan ve standart sapma değerleri farklı fonksiyonlarla hesaplanarak tablolar üzerinde gösterilmiştir.

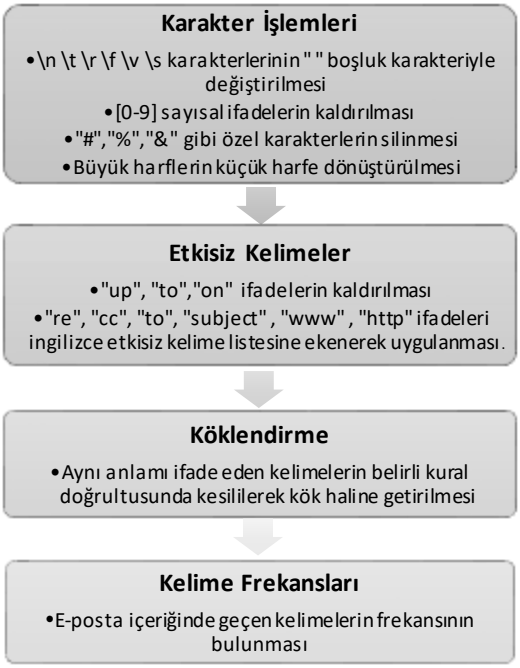
Şekil 2'de proje üzerinde gerçekleştirilen işlem adımları gösterilmiştir. İşlemler ardışık üç farklı adımda gerçekleştirilmiştir. Bunlar ön işlem adımları, kullanılan yöntemler ve değerlendirme metrikleri başlıkları altında açıklanmıştır. Deneysel çalışmalar ön işlem adımları ile başlamıştır. Bu adım, veri setlerinin temin edilmesi ve veri setlerinin standart bir yapıya getirilmesi işlemlerini kapsamaktadır. Verilerdeki hata, tutarsızlık ve gürültüler azaltılarak veri temizleme işlemi gerçekleştirilmiştir. Ön işlem sonrası elde edilen son veriler sınıflandırma algoritmalarının performanslarının ölçülmesi aşamasında kullanılmıştır. Ele alınan probleme uyarlanan yöntemler sınıflandırma çalışmalarında kullanılmıştır. Her bir algoritmanın en iyi performansla çalışabilmesi için örnek kütüphaneler kullanılmamıştır. Sıfırdan probleme uyarlanarak ve kod optimizasyonu yapılarak en iyi performans elde edilmeye çalışılmıştır. Uygulamalar gerçekleştirilirken kullanılan sayısal değerler sabit değerler olarak ele alınmamıştır. Deneme yapılarak en iyi sonuçları veren değerler seçilmiştir. Test adımında algoritma performanslarının ölçülebilmesi için doğruluk, kesinlik, duyarlılık ve F1-skor değerlendirme metrikleri hesaplanmıştır.



Şekil 2. Proje adımları (Project steps)

4.1. Ön İşlem Adımları (Preprocessing Steps)

Ön işlem adımları dört başlık altında toplanmıştır. Şekil 3'te işlem adımları gösterilmiştir. Her bir temel konunun açıklaması ilgili başlıklarda sunulmuştur.



Şekil 3. Ön işlem adımları (Preprocessing steps)

4.1.1. Karakter İşlemleri (Character Manipulation)

E-posta içerisinde bulunan fazla boşluklar silinerek tek bir boşluk bırakılmıştır. Sayısal ifadeler silinerek kelimeler üzerinde sınıflandırma işlemleri yapılmıştır. Ayrıca '@', '|', '<', '>', '#', '$', '%' ve bunlar gibi özel karakterler çıkarılmıştır. Bazı harflerin büyük harf ve küçük harf ile yazım şekilleri farklı olabilmektedir. Anlam bütünlüğünü sağlayabilmek ve aynı kelimenin farklı bir kelime gibi işlem yapılmasını engelleyebilmek amacıyla büyük-küçük harf uyumu sağlanmıştır. Büyük harfler küçük harflere dönüştürülerek standartlaştırma yapılmıştır.

4.1.2. Etkisiz Kelimeler (StopWords)

Etkisiz kelimeler tek başına kullanıldığında anlam ifade etmeyen kalıpları karşılayan listeyi temsil etmektedir. Burada, hazır olarak sunulan etkisiz kelimeler kütüphanesi kullanılarak bu kalıplar e-postalardan arındırılmıştır. Bu kütüphanenin 2018 versiyonu kullanılmıştır. Kullanılan veri setleri İngilizce kökenli olduğu için etkisiz kelimeler listesi olarak '*english'* kütüphanesi kullanılmıştır. Etkisiz kelime listesine "*re*", "*cc*", "*to*", "*subject*", "*www*", "*http*" ifadeleri de eklenmiştir. Bu kelimeler hem ham hem de spam maillerde yaygın olarak kullanıldığından ham ve spam ayrımı yaparken etkisi olmamaktadır. Bu kalıplar da eklendikten sonra etkisiz kelimeler belirlenmiştir.

4.1.3.Köklendirme (Rooting)

Aynı anlama sahip olan kelimelerin köklerine indirgenme işlemi burada anlatılmıştır. Bunlar kelimenin eşit ya da daha küçük halini göstermektedir. Köklendirme algoritmaları kural tabanlıdır. Kelimeleri kısaltan bir süreçtir. Köklendirme işlemleri için Python üzerinde '*Porter Stemmer'* kütüphanesi kullanılmıştır. Bu ön işlemin amacı e-posta içerisinde geçen kelime sayısını azaltıp frekansını artırmaktır. Bu işlemler, sınıflandırma yapılırken e-postanın ham ve spam sınıflarından hangisine ait olduğunu tespit etmede kolaylık sağlamaktadır.

4.1.4. Kelimelerin Frekansını Bulma (Find the Frequency of Words)

Kelimelerin hangi e-posta içerisinde ne kadar sık geçtiğini bulma işlemidir. Öncelikli olarak ön işleme aşamasındaki kelimelerin listesi oluşturulmuştur. Kelime listesi sütun isimleri haline dönüştürülerek yeni bir veri çerçevesine dönüştürülmüştür. Her bir kelimenin sırasıyla gelen mail içerisinde bulunup bulunmadığına bakılmıştır. Kelime varsa değişkenin sayısal değeri artırılarak araştırmaya devam edilmiştir. Kelimelerin frekansını bulma işlemi kelime vektörleştirme (Count Vectorizer) ile gerçekleştirilmiştir. Bu sayede hangi kelimenin hangi mail içerisinde ne kadar sık geçtiği belirlenmiştir. Oluşturulan tablo üzerinde label (etiket) ve mail (içerik) içeren sütunlar birleştirilmiştir. Veri setinin eğitim işlemine geçmeden önce elde edilen veri çerçevesi bir matrise dönüştürülmüştür. Matris üzerinden eğitim işlemi gerçekleştirilmiştir. En sık geçen kelimenin etiketine bakılarak daha sonra incelenecek e-postalar içerisindeki kelimelere göre sınıfın spam mı ham mı olduğu tespit edilmiştir. Burada *x_test* verileri gönderilerek tahminlerde bulunulmuştur. Bu tahmin değerleriyle elde bulunan *y_test* değerleri karşılaştırılmıştır. Son olarak elde edilen sonuçlara bakılarak başarım durumları hesaplanmıştır.

4.2. Kullanılan Yöntemler (Studied Method)

Bu başlık altında gerçekleştirilen çalışmalarda kullanılan algoritmaların çalışma mantıkları, parametrik değerleri ve hesaplama formülleri detaylı olarak anlatılmaktadır. Veri setleri üzerinde yapılan ön işlemlerden sonra problemin algoritmalara uyarlanması gerçekleştirilmiştir. Veri seti eğitim ve test verileri olarak ikiye bölünmüştür. Bölme oranları %20 test ve %80 eğitim verisi şeklindedir. Python her defasında farklı yerlerden bölme işlemi yapmaktadır. Random_state değeri belirlenerek rasgele sayıların aynı sırada üretilmesi sağlanır. Bundan dolayı *random_state=42* sabit değeri belirlenerek her çalıştırmada test ve eğitim verilerinin sabit olması sağlanmıştır. Böylece testlerde aynı eğitim ve test veri setleri kullanılmıştır. Bazı veri setlerinin *label* sıralaması ham ve spam şeklinde verilmiştir. Dağılımların homojen olabilmesi amacıyla burada karıştırma işlemi yapılmıştır. Bu işlem sonrasında ayırma işlemi gerçekleştirilmiştir. Böylece tüm veriler sınıflandırıcı algoritmalar üzerinde test edilmeye hazır hale getirilmiştir.

4.2.1. Naive Bayes (Naive Bayes)

Naive bayes, sınıflandırma görevi için kullanılan olasılıklı bir makine öğrenme algoritmasıdır. Bu sınıflandırıcının temel noktası bayes teoremine dayanmaktadır [50].

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \quad (1)$$

Denklem 1'deki Y değişkeni e-postanın ham ya da spam olduğunu gösteren sınıf değişkenidir. X ise kelimeleri temsil eden değişkenleri ifade etmektedir. E-postanın içerisinde geçen kelimelerin sıklığına göre ham/spam olup olmadığı araştırılmıştır. Sütunlar mail içerisinde geçen kelimeleri temsil etmektedir. Satırlar kelimelerin mail içerisinde bulunup bulunmadığını ve varsa ne kadar olduğunu temsil etmektedir. Mail içerisinde geçen kelimenin sıklığına göre ilgili mailin ham ya da spam olduğu tahmininde bulunulmaktadır. Bu aşamada iki farklı varsayım kullanılmaktadır. Birincisi, tahmin edicilerin birbirinden bağımsız olmasıdır. Yani kelime 1'in olması kelime 2'yi etkilememektedir. İkincisi ise tüm tahmin edicilerin sonuç üzerinde aynı etkiye sahip olmasıdır.

$$P(y\_train|x\_train) = \frac{P(x\_train|y\_train)P(y\_train)}{P(x\_train)} \quad (2)$$

Tüm girdiler eğitim ve test verisi olarak ayrıldığı için öncelikli olarak veriler eğitilmiştir. Örneğin 'kelime1' spam mail içerisinde çok tekrarlandıysa, test verisinde gelen kelime1'in sınıfı spam mail olduğu tahmininde bulunulmuştur. Test verisi olarak geriye kalan verilerin diğer kelimelere göre sınıflandırılması yapılarak etkin tahminlerde bulunulmuştur. Denklem 2'deki eşitlik kullanılarak tahmini değerler elde edilmiştir. Bu değerler ayrı bir çerçevede tutulmuştur. Bu veri çerçevesiyle *y_test* şeklinde ayrılan test değerleri ile karşılaştırılmıştır. Karşılaştırmanın sonucunda doğruluk, kesinlik, duyarlılık ve F1-skor değerlendirme metrikleri elde edilmiştir.

4.2.2. Lojistik Regresyon (Logistic Regression)

Bu algoritma, sonuçları kategorik olarak değerlendiren (evet/hayır, geçti/kaldı vb.) makine öğrenmesi yaklaşımı sunmaktadır ve birden fazla sonuca dayalı sınıflandırma çıktısı üretmektedir. Bu çalışma kapsamında LR yaklaşımı ikili sınıflandırma amacıyla kullanılmıştır. Sonucun ham veya spam olarak sınıflandırılması şeklinde ele alınmıştır.

Yöntemin uygulanmasında eğitilmiş veri seti verildiğinde;

$$B = \{\langle B_1, Y_1\rangle, \dots, \langle B_i, Y_i\rangle, \dots, \langle B_m, Y_m\rangle\} \quad (3)$$

i. veri için $P(Y_i = 1|B_i)$ parametresi spam olma olasılığını sunarken; $(Y_i = 0|B_i)$, çıktının ham olma olasılığını vermektedir. Burada Y_i, ikili hesaplamanın nihai sonucunu göstermektedir (örneğin spam veya ham gibi). Burada binom log-olasılık işlevi Denklem 4'te sunulmuştur.

$$L = \sum_{i=1}^{m}[Y_i \log P(Y_i = 1|B_i) + (1 - Y_i) \log P(Y_i = 0|B_i)] \quad (4)$$

Lojistik regresyonun kullanıldığı tek bir örnek uzayında, bir X_i örneği verildiğinde, X_i'nin ikili sonucunun beklenen değeri ilgili Sigmoid fonksiyonu ile modellenmektedir. Burada $P(Y_i = 1|X_i) = (\exp(p.X_i + b)/(1 + \exp(p.X_i + b)))$ hesaplamasının ardından lojistik regresyon fonksiyonunu maksimize eden p ve b parametreleri tahmin edilmektedir. İkili düzey sınıf olasılıklarını veren $P(Y_{ij} = 1|X_{ij})$ fonksiyonu Denklem 5'teki gibi hesaplanmaktadır.

$$P(Y_{ij} = 1|X_{ij}) = \frac{\exp(p.X_{ij}+b)}{1+\exp(p.X_{ij}+b)} \quad (5)$$

Burada X_{ij}, i. e-postadaki j. Örneği göstermektedir. Parametrelerden p ve b girdileri tahmin edilmesi gereken içerikleri ifade etmektedir. Böylece girdilere göre örnek düzeyinde verilen sınıf olasılıklarının sunulduğu $P(Y_i = 0|B_i)$ değeri, Denklem 6'daki gibi hesaplanmaktadır [51]:

$$P(Y_{ij} = 0|X_{ij}) = \frac{1}{1+\exp(p.X_{ij}+b)} \quad (6)$$

Denklem 5 ve 6 da gösterilen sigmoid fonksiyonu tahmini değerlerin olasılıksal eşleştirilebilmesi için kullanılmıştır.

$$weight = full\ (x\_train.T\ , 0{,}01) \quad (7)$$

$$x = dot(weight.T, x\_train) + bias \quad (8)$$

Vektör parametresi ile hesaplanan *x_train* matrisinin öncelikli olarak transpozu alınmıştır. Ağırlıklandırma değeri olarak seçilen *0,01* değeriyle matrisin ağırlık değeri hesaplanmıştır. Bu değerin de transpozu alınmış ve *x_train* matrisi ile çarpımı gerçekleştirilmiştir. İşlem sonucu *bias* değeriyle toplanarak *x* değeri elde edilmiştir.

$$loss = -(1 - y\_t) * log(1 - f(x)) - y\_t * \log(f(x)) \quad (9)$$

$$cost = (sum(loss))/x\_train \quad (10)$$

Hesaplanan sigmoid değerine göre lojistik regresyon hipotezi hesaplanmıştır. Ardından maliyet hesaplaması yapılmıştır. Minimum hatayla doğru bir model oluşturabilmek için maliyet en aza indirilmiştir. Kayıp (*loss*) değeri Denklem 9'a göre hesaplanmıştır. Burada *y_train* parametresi, *y_t* ile gösterilmiştir. Denklem 10'da ise maliyet hesaplaması gösterilmiştir. Maliyet azaltma gradyan hesaplama ile yapılmıştır. Her maliyette gradyan fonksiyonu çalıştırılmıştır. Öğrenme oranı olarak 4 değeri seçilmiştir. Tekrarlama sayısı olarak 200 kullanılmıştır. Bu değerler deneme yanılma yaklaşımı sonucunda bulunmuştur. En iyi başarım değerleri bu sayılarla elde edilmiştir. Veriler eğitildikten sonra *x_test* ile gönderilen test verileriyle tahmini değerler elde edilmiştir. Bu tahmini değerlerle *y_test* değerleri karşılaştırılmıştır. Eşitliklere göre değerlendirme metrikleri elde edilmiştir.

4.2.3. Karar Ağacı (Decision Tree)

Karar ağacı, tahmine dayalı modelleme yaklaşımı sunan makine öğrenmesi algoritmasıdır. Birkaç girdi değişkeni temelinde bir hedef değişkenin değerini tahmin eden bir model oluşturmaktadır. Her bölmede aşamalı olarak bir ağaç oluşturan ve yaygın olarak kullanılan bir yöntemdir. Karar ağacının her bir düğümü farklı öznitelikler üzerinde bir kararı gösterirken; dallar, kararların yönünü ve yaprak düğümler ise sınıf etiketlerinin nihai çıktılarını temsil etmektedir. Ağaç oluşturulurken ayırma işlemleri, bilgi kazancına (information gain-IG) ve gini indeksi fonksiyonlarına göre gerçekleştirilmektedir [22]. Ergin ve diğerlerinin Turkish Spam V01 veri seti üzerinde yapmış oldukları çalışmada özellik seçimi olarak bilgi kazancı, gini indeksi ve CHI karesi (CHI2) yöntemlerini kullanmışlardır. Uyguladıkları veri seti üzerinde gini indeksinin ve CHI2 yönteminin bilgi kazancından daha etkili sonuçlar verdiği gözlemlenmiş [52].

Yapılan çalışmada mevcut verilere göre eğitimler yapılarak yeni gelen verilere göre uygun öğrenme modeli geliştirilmiştir. Çalışma içerisinde veri setine vektör işlemi uygulanarak kullanılacak olan veri matrise dönüştürülmüştür. Bu matris ile veri setinin *label* sütunu birleştirilmiştir. Elde edilen veri seti *en iyi bölme* fonksiyonuna gönderilmiştir. Bu algoritmadaki bölünmeleri kontrol edebilmek amacıyla gini indeksi yaklaşımı kullanılmıştır. Gini indeksi hesaplamasının yapılabilmesi için ayrı bir sütun oluşturulmuştur. O sütuna özellik olarak verilen kelimelerin sayısı sayılıp atama işlemi yapılmıştır. Bulunan bu değerler gini indeksi dizininde kullanılmıştır.

$$gini = 1 - p^2 \qquad (11)$$

Denklem 11'de gösterilen gini indeksi fonksiyonunu hesaplayan formülde p değeri yeni oluşturulan sütunun her bir satırdaki değerin veri setinin uzunluğuna bölünmesiyle elde edilmiştir. Gini indeksi çalışmanın başlangıcında bir (1) olarak belirlenmiştir. Gini indeksini hesaplayabilmek için başlangıç değerinden çıkarma işlemi yapılmıştır. Formül uygulandıktan sonra yeni gini indeksi elde edilmiştir. Burada *en iyi bölme* fonksiyonundaki sütunlarda tekrar eden veriler göz ardı edilmiştir. Geriye kalan sütunlarda en fazla geçen kelime bulunmuştur. Bulunan kelimeler ağacın bölme noktasını göstermektedir. Elde edilen değer, bu çalışmada *question* olarak tanımlanmış ve "en iyi özellik/değer" ikilisi olarak tanımlanmıştır. Uygulamada yeni ayrılan parçalardan bilgi kazancı (gain) hesaplaması yapılmıştır. Denklem 12'de satırlar (rows), R ile temsil edilmiştir.

$$gain = float(len(left_R))/(len(left_R) + (len(right_R) \qquad (12)$$

$$newGain = g - ga * (left_g) - (1 - ga) * (right_g) \qquad (13)$$

Denklem 13'te elde edilen değer başlangıçta verilen en iyi değerden büyük veya eşit ise burada hesaplanan *newGain* değişkeni yeni en iyi değer olarak atanmıştır. Bu denklemde gini parametresi g ile gösterilmiştir. Gain ise ga ile temsil edilmiştir. Bulunan *question* değeri de en iyi soru olarak atanmıştır. Bu şekilde en iyi değer hesaplamasına devam edilmiştir. Sınıflandırma algoritması ile hangi düğümün tercih edileceğine karar verilmiştir. Bölünmüş olan test verileri çalıştırılarak tahmini değerler elde edilmiştir. Bu tahmini değerlere göre başarım oranları hesaplanmıştır.

4.2.4. K-En Yakın Komşu (K-Nearest Neighbors)

K-en yakın komşu algoritması, sınıflandırma, veri madenciliği ve diğer birçok alanda kullanılan denetimli öğrenme algoritmasıdır. KNN sınıflandırıcısı öklid, manhattan ve chebyshev gibi uzaklık belirleme yöntemlerine göre çalışmaktadır. Bu çalışmada kullanılmış olan öklid hesaplama yöntemi Denklem 14'e göre hesaplanır.

$$X_i \; ve \; X_j : X_i = (X_{i1}, X_{i2}, , X_{in}) \text{ ve } X_j = (X_{j1}, X_{j2}, \dots , X_{jn});$$

$$D(X_i, X_j) = \sqrt{\sum_{K=1}^{n}((X_{ik} - X_{jk}))^2} \qquad (14)$$

KNN yönteminin en önemli dezavantajlarından biri büyük ölçekli ve yüksek boyutlu veri kümeleri için verimsiz olmasıdır [53]. Dezavantajının arkasındaki temel dayanak diğer algoritmalardan farklı olarak öğrenme aşamasının olmamasıdır. Bir tahmin işlemi sırasında eğitim veri setindeki en yakın komşular aranmaktadır. Yeni bir veri noktası geldiğinde KNN algoritması bu yeni veri noktasının en yakın komşularını bularak başlamaktadır. Daha sonra bu komşuların değerlerini almaktadır ve bunları yeni veri noktası için bir tahmin olarak kullanmaktadır.

Kullanılan veri setlerinde vektör işlemleri diğer algoritmalarda sınıflandırmanın dışında yapılmıştır. K-en yakın komşu algoritmasında vektör işlemleri KNN sınıfı içerisinde yapılmıştır. Bu algoritmanın çalışmasında bir k değeri belirlenmiştir. Bu k değeri ile yeni gelen değerin kaç adet komşu arasındaki uzaklığa bakılarak hangi sınıfa ait olacağını bulmaktadır. Minimum k değeri 1 olarak belirlenmektedir. Bu, tahmin için yalnızca bir komşu kullanılması anlamına gelmektedir. Maksimum k değeri ise sahip olunan tüm veri noktasını göstermektedir. Örneğin; başlangıç değeri olarak k=5 alındığında, yeni gelen veri için uygun sınıfın tahmin edilmesinde 5 komşu seçilmiş olacaktır. Öklid mesafe hesaplama fonksiyonu ile komşuların uzaklıkları hesaplanmıştır. Bu ölçüt, iki nokta arasındaki uzaklığın kareleri toplamının karekökünü eşittir. İlgili formül, Denklem 14'te verilmiştir. Bunun sonucunda da yeni gelen verinin sınıfı tahmin edilmiştir. K=9 ve k=11 değerleri verilerek tekrardan komşuların uzaklıkları hesaplanmıştır. Farklı k değerleri verilerek başarılı sonuçlara hangi k değeri ile elde edilebileceği bulunmuştur. Sonuç olarak k=9 değeriyle daha başarılı sonuçlar elde edilmiş ve test çalışmalarında k=9 değerinin kullanılmasına devam edilmiştir. Uzaklık değerlerine göre hesaplanan mesafeler artan düzende sıralanmıştır. Sıralanan dizinde en üstteki k satırı alınarak en sık

göründüğü sınıf değeri tahmini değer olarak geri döndürülmüştür. Bu tahmini değerlerle *y_test* değerleri karşılaştırılarak değerlendirme metrikleri hesaplanmıştır.

4.3. Değerlendirme Metrikleri (Evaluation Metrics)

İncelenen çalışmalarda performansların değerlendirilmesi, çeşitli metrikler aracılığıyla yapıldığı gözlemlenmiştir. Bu değerlendirme metrikleri arasından doğruluk, kesinlik, duyarlılık ve F1-skor metrikleri proje çalışması içerisinde kullanılarak en başarılı sonuca hangi algoritma ile ulaşıldığı, hangi veri seti ile başarı çalışmanın gerçekleştirildiği gözlemlenmiştir [54].

Doğruluk: $\frac{TP+TN}{TP+TN+FP+FN}$

Kesinlik : $\frac{TP}{TP+FP}$

Duyarlılık: $\frac{TP}{TP+FN}$

F1-skor: $2 * \frac{kesinlik * duyarlılık}{kesinlik + duyarlılık}$

Bu çalışma içerisinde veri setlerinin algoritma üzerinde stabil olarak çalışıp çalışmadığını anlayabilmek amacıyla uygulamalar 10 kez çalıştırılmıştır. Her bir algoritma üzerinde aynı veri setinin aynı ham/spam oranıyla ve aynı sabit değerleriyle çalıştırılması gerçekleştirilmiştir. Bunun sonucunda en iyi, en kötü, ortalama, metrik ve standart sapma değerleri farklı bir fonksiyonla hesaplanmıştır. Yüksek bir standart sapma değerinin olmadığı ve en iyi en kötü değerlerin farklı olmadığı gözlemlenmiştir.

En iyi değer: *Elde edilen sonuçlar arasında en başarılı sonucu verir.*

En kötü değer: *Elde edilen sonuçlar arasında en kötü sonucu verir.*

Ortalama değer: $x_1 + x_2 + \cdots + x_{10}/10$

Medyan: $p = 10/2\,,\ \ (x_p + x_{p+1})/2$

Standart sapma: $\sqrt{\sum_{i=1}^{10}(x_i - \mu)^2/10}$

5. DENEYSEL SONUÇLAR (EXPERIMENTAL RESULTS)

Bu çalışmada öncelikle testlerde kullanılacak veri setleri standart hale getirilmiştir. Standartlaştırılan bu veri setleri üzerinde ön işlem adımları gerçekleştirilerek özellik olarak kullanılacak olan kelimelerin sayısı azaltılmaya çalışılmıştır. Algoritmalarda özellik seçimi yapılarak sınıflandırma çalışmaları ayrı ayrı her bir veri seti için tekrar çalıştırılmıştır. Algoritmaların başarım durumlarını gözlemleyebilmek için çeşitli değerlendirme metrikleri kullanılmıştır. Testlerde literatürde çok sık rastlanmayan veri setleri tercih edilmiştir. Enron veri setini ayrı ayrı değerlendirerek içerisinden 4 adet veri seti kullanılmıştır. Farklı büyüklükte girdi verilerine sahip olması ve farklı dağılımlarda ham/spam çıktısına sahip olan veri setlerinin kullanılması durumunda algoritmaların performansları incelenmiştir. Beş farklı veri seti üzerinde algoritmalar ayrı ayrı uygulanarak bunların çalışma performansları karşılaştırılmıştır. Algoritmaların deneysel çıktılarının genelleştirilebilmesi için testler 10 kez tekrarlanmıştır. Burada eğitim ve test verilerinin k-fold cross validation yaklaşımına göre oluşturulması sağlanmış ve böylelikle daha nesnel değerlendirmelerin yapılması sağlanmıştır. Test çalışmaları sonucunda farklı ölçütlerle çalışma performansları değerlendirilmiştir. Sonuçlar Tablo 2-5 arasında gösterilmiştir. Tablolar üzerinde görüldüğü gibi algoritma çıktılarının yüksek standart sapmalara sahip olmadığı gözlemlenmiştir. Çalışmaların benzer sonuçları verdiği gözlemlenmiştir. Elde edilen tahmini sonuçlar doğruluk, kesinlik, duyarlılık ve F1-skor değerlendirme metriklerine göre ilgili tablolarda sunulmuştur.

Tablo 2-5'te gösterilen tablolar minimize edilerek yeni değerlendirme tabloları elde edilmiştir. Tablo 6-9'da çalıştırılan algoritmalar ile elde edilen doğruluk, kesinlik, duyarlılık ve F1-skor başarım oranlarından en iyi ve en kötü değerlerin hangi veri seti ile elde edildiği gösterilmiştir. NB ile elde edilen metrik değerlerinden en iyi sonucun %99,8 ile kesinlik metriğiyle Enron 5 veri setine ait olduğu gözlemlenmiştir. En kötü değerin duyarlılık metriğinde %58,2 değeri ile SpamAssassin veri setinde elde edildiği anlaşılmıştır. LR algoritması ile elde edilen metrik değerlerinden en iyi sonucun %99,58 ile F1-skor metriğiyle elde edilmiştir. Bu başarı değerinin Enron 5 veri setine ait olduğu gözlemlenmiştir. En kötü değerlerin tüm metriklerde SpamAssassin veri seti ile ulaşılmıştır. En düşük sonucun kesinlik metriğinde %90,19 olduğu görülmektedir. DT'nin uygulanması ile elde edilen değerlerden en iyi sonucun %98,96 ile duyarlılık metriğiyle CS440/ECE448 veri seti üzerindeki testler sonucunda elde edilmiştir. Bu veri dışında diğer metriklere göre Enron 4 veri setinde de başarılı sonuçlar elde edilmiştir. Ayrıca en kötü değerlerin tüm metriklerde SpamAssassin veri setinde elde edildiği gözlemlenmiştir. %83,51 ile kesinlik ölçütünde en düşük değer elde edilmiştir. KNN ile elde edilen metrik değerlerinden en iyi sonucun %98,89 ile duyarlılık ve F1-skor metrikleriyle Enron 5 veri seti üzerinde elde edilmiştir. En kötü değerin kesinlik dışındaki metriklerle SpamAssassin veri setinde elde edildiği gözlemlenmiştir.

Başarım oranları açısından karşılaştırma tablolarına göre DT yöntemi dışındaki diğer algoritmalarda F1-skor metriğinde Enron 5 veri seti üzerinde en uygun sonuçlara ulaşıldığı gözlemlenmiştir. Diğer metriklerde de Enron 5 veri seti üzerinde başarılı sonuçlara ulaşıldığı çıkarımı yapılabilir. Enron 5 veri setinde ham/spam oranında spam oranın yüksek olması ve özellik seçiminde kullanılan kelimelerin sınıflandırmanın doğru çalışmasında etkili olduğu sonucuna varılmıştır. DT yaklaşımında duyarlılık metriği dışında en başarılı sonuçlar Enron 4 üzerinde elde edilmiştir. DT algoritmasında farklı özellik seçim

çalışmasının yapılması Enron 4 veri seti üzerinde olumlu sonuçların elde edilmesini sağlamıştır. NB' de doğruluk, KNN' de kesinlik metrikleri dışındaki testlerde SpamAssassin verisi için genellikle başarısız sonuçlara ulaşılmıştır. SpamAssassin veri setinin ham oranın fazla olması sınıflandırmanın diğer veri setlerine göre daha düşük başarım değerleriyle sonuçlanmasına sebep olmuştur.

Tablo 2. NB başarım tablosu (NB achievement table)

Veri Seti Adı	Metrik	En İyi Değer	En Kötü Değer	Medyan	Ortalama Değer	Standart Sapma
SpamAssassin	Doğruluk	0,8347	0,8347	0,8347	0,8347	1,11022E-16
	Kesinlik	0,8796	0,8796	0,8796	0,8796	0
	Duyarlılık	0,582	0,582	0,582	0,582	1,11022E-16
	F1-skor	0,7005	0,7005	0,7005	0,7005	1,11022E-16
Enron 4	Doğruluk	0,7583	0,7583	0,7583	0,7583	1,11022E-16
	Kesinlik	0,9849	0,9849	0,9849	0,9849	1,11022E-16
	Duyarlılık	0,677	0,677	0,677	0,677	0
	F1-skor	0,8024	0,8024	0,8024	0,8024	0
Enron 5	Doğruluk	0,7835	0,7835	0,7835	0,7835	1,11022E-16
	Kesinlik	0,998	0,998	0,998	0,998	0
	Duyarlılık	0,6911	0,6911	0,6911	0,6911	1,11022E-16
	F1-skor	0,8166	0,8166	0,8166	0,8166	1,11022E-16
Enron 6	Doğruluk	0,7216	0,7216	0,7216	0,7216	0
	Kesinlik	0,9926	0,9926	0,9926	0,9926	2,22045E-16
	Duyarlılık	0,6206	0,6206	0,6206	0,6206	0
	F1-skor	0,7637	0,7637	0,7637	0,7637	0
CS440/ECE448	Doğruluk	0,7849	0,7849	0,7849	0,7849	1,11022E-16
	Kesinlik	0,9672	0,9672	0,9672	0,9672	2,22045E-16
	Duyarlılık	0,6082	0,6082	0,6082	0,6082	1,11022E-16
	F1-skor	0,7468	0,7468	0,7468	0,7468	1,11022E-16

Tablo 3. LR başarım tablosu (LR achievement table)

Veri Seti Adı	Metrik	En İyi Değer	En Kötü Değer	Medyan	Ortalama Değer	Standart Sapma
SpamAssassin	Doğruluk	0,9454	0,9454	0,9454	0,9454	1,11022E-16
	Kesinlik	0,9019	0,9019	0,9019	0,9019	1,11022E-16
	Duyarlılık	0,9378	0,9378	0,9378	0,9378	0
	F1-skor	0,9195	0,9195	0,9195	0,9195	2,22045E-16
Enron 4	Doğruluk	0,9908	0,9908	0,9908	0,9908	0
	Kesinlik	0,9897	0,9897	0,9897	0,9897	0
	Duyarlılık	0,9977	0,9977	0,9977	0,9977	1,11E-16
	F1-skor	0,9937	0,9937	0,9937	0,9937	2,22E-16
Enron 5	Doğruluk	0,9942	0,9942	0,9942	0,9942	1,11E-16
	Kesinlik	0,9944	0,9944	0,9944	0,9944	0
	Duyarlılık	0,9972	0,9972	0,9972	0,9972	0
	F1-skor	0,9958	0,9958	0,9958	0,9958	0
Enron 6	Doğruluk	0,9783	0,9783	0,9783	0,9783	1,11022E-16
	Kesinlik	0,9741	0,9741	0,9741	0,9741	1,11022E-16
	Duyarlılık	0,9965	0,9965	0,9965	0,9965	1,11022E-16
	F1-skor	0,9852	0,9852	0,9852	0,9852	1,11022E-16
CS440/ECE448	Doğruluk	0,9892	0,9892	0,9892	0,9892	2,22E-16
	Kesinlik	0,9797	0,9797	0,9797	0,9797	0
	Duyarlılık	1	1	1	1	0
	F1-skor	0,9897	0,9897	0,9897	0,9897	2,22E-16

Tablo 4. DT başarım tablosu (DT achievement table)

Veri Seti Adı	Metrik	En İyi Değer	En Kötü Değer	Medyan	Ortalama Değer	Standart Sapma
SpamAssassin	Doğruluk	0,9165	0,9165	0,9165	0,9165	1,11022E-16
	Kesinlik	0,8351	0,8351	0,8351	0,8351	0
	Duyarlılık	0,9328	0,9328	0,9328	0,9328	0
	F1-skor	0,8813	0,8813	0,8813	0,8813	0
Enron 4	Doğruluk	0,97	0,97	0,97	0,97	0
	Kesinlik	0,9771	0,9771	0,9771	0,9771	1,11022E-16
	Duyarlılık	0,9816	0,9816	0,9816	0,9816	0
	F1-skor	0,9793	0,9793	0,9793	0,9793	0
Enron 5	Doğruluk	0,9603	0,9603	0,9603	0,9603	0
	Kesinlik	0,9696	0,9696	0,9696	0,9696	1,11022E-16
	Duyarlılık	0,9736	0,9736	0,9736	0,9736	1,11022E-16
	F1-skor	0,9716	0,9716	0,9716	0,9716	0
Enron 6	Doğruluk	0,95	0,95	0,95	0,95	0
	Kesinlik	0,9571	0,9571	0,9571	0,9571	0
	Duyarlılık	0,9747	0,9747	0,9747	0,9747	1,11022E-16
	F1-skor	0,9658	0,9658	0,9658	0,9658	1,11022E-16
CS440/ECE448	Doğruluk	0,9462	0,9462	0,9462	0,9462	2,22045E-16
	Kesinlik	0,9142	0,9142	0,9142	0,9142	1,11022E-16
	Duyarlılık	0,9896	0,9896	0,9896	0,9896	0
	F1-skor	0,9504	0,9504	0,9504	0,9504	1,11022E-16

Tablo 5. KNN başarım tablosu (KNN achievement table)

Veri Seti Adı	Metrik	En İyi Değer	En Kötü Değer	Medyan	Ortalama Değer	Standart Sapma
SpamAssassin	Doğruluk	0,9057	0,9057	0,9057	0,9057	0
	Kesinlik	0,9832	0,9832	0,9832	0,9832	0
	Duyarlılık	0,7288	0,7288	0,7288	0,7288	1,11022E-16
	F1-skor	0,8371	0,8371	0,8371	0,8371	0
Enron 4	Doğruluk	0,9616	0,9616	0,9616	0,9616	0
	Kesinlik	0,9768	0,9768	0,9768	0,9768	0
	Duyarlılık	0,9701	0,9701	0,9701	0,9701	2,22045E-16
	F1-skor	0,9734	0,9734	0,9734	0,9734	1,11022E-16
Enron 5	Doğruluk	0,9845	0,9845	0,9845	0,9845	1,11022E-16
	Kesinlik	0,9889	0,9889	0,9889	0,9889	1,11022E-16
	Duyarlılık	0,9889	0,9889	0,9889	0,9889	1,11022E-16
	F1-skor	0,9889	0,9889	0,9889	0,9889	1,11022E-16
Enron 6	Doğruluk	0,9441	0,9441	0,9441	0,9441	1,11022E-16
	Kesinlik	0,9695	0,9695	0,9695	0,9695	0
	Duyarlılık	0,9528	0,9528	0,9528	0,9528	0
	F1-skor	0,9611	0,9611	0,9611	0,9611	1,11022E-16
CS440/ECE448	Doğruluk	0,9784	0,9784	0,9784	0,9784	0
	Kesinlik	1	1	1	1	0
	Duyarlılık	0,9587	0,9587	0,9587	0,9587	1,11022E-16
	F1-skor	0,9789	0,9789	0,9789	0,9789	2,22045E-16

Tablo 6. NB en iyi ve en kötü değerlerin karşılaştırılması (NB comparison of best and worst values)

Metrik	En İyi Değer	Veri Seti	En Kötü Değer	Veri Seti
Doğruluk	0.8347	SpamAssassin	0.7216	Enron 6
Kesinlik	0.998	Enron 5	0.8796	SpamAssassin
Duyarlılık	0.6911	Enron 5	0.582	SpamAssassin
F1-skor	0.8166	Enron 5	0.7005	SpamAssassin

Tablo 7. LR en iyi ve en kötü değerlerin karşılaştırılması (LR comparison of best and worst values)

Metrik	En İyi Değer	Veri Seti	En Kötü Değer	Veri Seti
Doğruluk	0.9942	Enron 5	0.9454	SpamAssassin
Kesinlik	0.9944	Enron 5	0.9019	SpamAssassin
Duyarlılık	1.0	CS440/ECE448	0.9378	SpamAssassin
F1-skor	0.9958	Enron 5	0.9195	SpamAssassin

Tablo 8. DT en iyi ve en kötü değerlerin karşılaştırılması (DT comparison of best and worst values)

Metrik	En İyi Değer	Veri Seti	En Kötü Değer	Veri Seti
Doğruluk	0.97	Enron 4	0.9165	SpamAssassin
Kesinlik	0.9771	Enron 4	0.8351	SpamAssassin
Duyarlılık	0.9896	CS440/ECE448	0.9328	SpamAssassin
F1-skor	0.9793	Enron 4	0.8813	SpamAssassin

Tablo 9. KNN en iyi ve en kötü değerlerin karşılaştırılması (KNN comparison of best and worst values)

Metrik	En İyi Değer	Veri Seti	En Kötü Değer	Veri Seti
Doğruluk	0.9845	Enron 5	0.9057	SpamAssassin
Kesinlik	1.0	CS440/ECE448	0.9695	Enron 6
Duyarlılık	0.9889	Enron 5	0.7288	SpamAssassin
F1-skor	0.9889	Enron 5	0.8371	SpamAssassin

Bu çalışma sonucunda algoritmalara göre hangi veri seti ile başarılı sonuç elde edildiği gözlemlenmiştir. Tablo 6-9'da algoritmalar üzerinde hangi veri seti nasıl çalışmış değerlendirme metrikleriyle gösterilmiştir. Veri setine göre hangi algoritmanın başarılı bir şekilde çalıştığı gözlemlenmiştir. Tablo 10'da test sonuçlarına göre elde edilen değerler gösterilmiştir.

Deneysel sonuçlarla ilgili Tablo 10'da veri setlerinde değerlendirme metriklerine göre en başarı sonucun hangi algoritma ile elde edildiği gösterilmiştir. Burada veri setlerindeki en başarılı sonuçların hangi algoritma ile elde edildiği bilgisi de koyu renkle vurgulanmıştır. SpamAssassin veri setinde kesinlik metriği dışında diğer metriklerde en iyi değerin LR ile elde edildiği anlaşılmıştır. SpamAssassin veri setinde %98,32 oranıyla en iyi sonucun KNN ile elde edildiği gözlemlenmiştir. Bu sonuç kesinlik metriğine göre elde edilmiştir. Enron 4 veri setindeki testlere göre tüm metriklerde en iyi değerler LR yöntemi ile elde edilmiştir. Buna göre %99,77 ile duyarlılık metriğinin en başarılı sonucuna ulaşıldığı gözlemlenmiştir. Enron 5 veri setinde kesinlik metriği dışındaki metriklerde en iyi değerin lojistik regresyon ile elde edildiği gözlemlenmiştir. Bu veri üzerinde yapılan çalışmalarda %99,8 ile en iyi skoru veren yaklaşımın NB ile kesinlik metriğine göre elde edildiği anlaşılmıştır. Enron 6 veri seti ile yapılan çalışmalarda en başarılı sonuçların kesinlik metriği dışındaki metriklerde lojistik regresyon algoritmasının verdiği görülmektedir. Bu veri setindeki deneylerde kesinlik metriğine göre en iyi sonuç NB ile elde edilmiştir. Metrik değerleri arasında duyarlılık metriği ile %99,65 sonucunun LR yöntemiyle elde edildiği gözlemlenmiştir.

CS440/ECE448 veri seti üzerinde yapılan deneysel çalışmalarda kesinlik metriği dışındaki tüm metriklerde lojistik regresyon algoritmasının daha başarılı sonuçlar verdiği gözlemlenmiştir. Kesinlik metriğinde ise başarılı sonucun k-en yakın komşu algoritması ile elde edildiği gözlemlenmiştir. Bu veri seti ile yapılan çalışmalarda %100 başarım ile en iyi sonucun kesinlik metriğinde KNN ile elde edildiği ve duyarlılık metriğinde ise LR ile elde edildiği sonucuna varılmıştır.

Tablo 10. Veri setlerinin değerlendirme metrik sonuçları (Evaluation metric results of datasets)

Veri seti / Metrik	SpamAssassin		Enron 4		Enron 5		Enron 6		CS440/ECE448	
Doğruluk	LR	0.9454	LR	0.9908	LR	0.9942	LR	0.9783	LR	0.9892
Kesinlik	**KNN**	**0.9832**	LR	0.9897	**NB**	**0.9980**	NB	0.9926	**KNN**	**1.0**
Duyarlılık	LR	0.9378	**LR**	**0.9977**	LR	0.9972	**LR**	**0.9965**	**LR**	**1.0**
F1-skor	LR	0.9195	LR	0.9937	LR	0.9958	LR	0.9852	LR	0.9897

Tablo 11'de literatürde yapılmış olan Enron veri seti ile çalışma içerisinde kullanılmış olan Enron 4-5-6 ve CS440/ECE448 veri setlerinin performans karşılaştırması yapılmıştır. Yapılmış olan Enron veri seti ile ayrı ayrı Enron veri setleri olarak değerlendirildiğinde farklı sonuçlara erişildiği gözlemlenmiştir. En çok naive bayes algoritması ile çalışma yapıldığı görülmektedir. Proje çalışması kapsamında farklı algoritmalarla çalışma gerçekleştirilmiştir. İncelenen çalışmalar arasında lojistik regresyon algoritmasına denk gelinmemiştir. Bu da çalışmamızın farkını göstermektedir. Burada algoritma tabanlı inceleme yapıldığında naive bayes algoritmasında [10] referanslı çalışmada %93 değerin üstünde değerlendirme metrikleri sonucuyla iyi bir sonuç göstermiştir. Yapılan çalışmada kesinlik metriğinde yüksek başarım oranları elde edilmiştir. Diğer metriklerde naive bayes algoritmasında yüksek oranlar elde edilememiştir. Karar ağacı algoritmasında değerlendirme metriklerinde %97 üzerinde sonuç elde ederek Enron 5 veri seti üzerinde algoritmanın iyi çalıştığı sonucuna varılmıştır. K-en yakın komşu algoritmasında CS440/ECE448 veri seti üzerinde yapılan çalışma sonucunda kesinlik metriğinde %100 başarım değeri elde edilmesine rağmen diğer değerlendirme metrikleriyle beraber düşünüldüğünde Enron 5 veri setinde %98 ve üstü sonucuyla daha başarılı olarak değerlendirilebilir.

SpamAssassin veri seti ile yapılmış olan çalışmaların performans karşılaştırması Tablo 12 ile verilmiştir. [10] referanslı çalışmada naive bayes algoritmasının yapılan çalışmaya göre doğruluk, kesinlik ve duyarlılık metriklerinde iyi sonuçların elde edildiği gözlemlenmiştir. [22] referanslı çalışmada KNN algoritmasında doğruluk ve duyarlılık metriklerinde %96 üstünde başarılı sonuçlar verdiği gözlemlenmiştir. Aynı çalışmada DT algoritması ile de çalışılmıştır. Doğruluk metriğinde verdiği sonucun yapılan çalışmadan daha iyi olduğu gözlemlenmiştir. Proje çalışmasında SpamAssassin veri seti ile çalıştırılan algoritmaların iyi sonuçlar vermediği gözlemlenmiştir. Bu durumun spam oranlarının az olmasından kaynaklandığı düşünülmektedir. Diğer çalışmalara kıyasla bu çalışmada lojistik regresyon algoritması da incelenmiştir. Bu yaklaşıma göre elde edilen sonuçların da kabul edilebilir düzeyde olduğu görülmektedir. Bu veri seti üzerinde LR ile çalışıldığında uygun sonuçlara ulaşıldığı anlaşılmıştır.

6.1. Sonuçların Değerlendirilmesi

Çalışmanın sonucunda farklı ölçütlere göre değerlendirme yapılmıştır. Veri seti olarak kullanılan CS440/ECE448 veri seti hariç diğerlerinde birbirine yakın sayılarda veri içeren setler seçilmiştir. Enron 4,5,6 ve SpamAssassin veri setlerinin veri sayıları birbirine yakındır. Ham/spam oranı olarak SpamAssassin veri setinin ham oranı diğerlerine kıyasla daha fazladır. Ham/spam oranının başarım değerlendirmesi üzerine etkisi izlenmiştir. Spam oranının %71 olduğu Enron 5 veri setinin iyi performans gösterdiği gözlemlenmiştir. Burada spam oranın fazla olmasının çalışmaya etkisinin olumlu olduğu sonucuna varılmıştır.

Ham/spam oranı birbirine eşit olan CS440/ECE448 veri setinde lojistik regresyon algoritmasında duyarlılık değerlendirme metriğiyle %100 başarım elde edilmiştir. K-en yakın komşu algoritmasında da kesinlik değerlendirme metriğiyle %100 başarılı sonuç elde edilmiş. Veri seti sayısının az olması ve ham/spam oranın eşit olması performansa olumlu etki etmiştir. Veri sayısının yanında algoritmada kullanılan fonksiyonlarında performansa etkisi incelenmiştir. Karar ağacı algoritması dışındaki algoritmalarda özellik seçimi olarak TF-IDF kullanılmıştır. Başarım değerlendirme metriklerinde Enron 5 veri seti ile iyi sonuçlar elde edilmiştir. Karar ağacı algoritmasında ise özellik seçimi olarak bilgi kazancı ve gini indeksi kullanılmıştır. Bu algoritma ile Enron 4 veri setinde daha başarılı sonuçlar elde edilmiştir.

6. SONUÇ VE ÖNERİ (CONCLUSION AND SUGGESTION)

Makine öğrenmesi algoritmaları spam tespiti gibi gerçek dünya problemlerinin çözüm aşamasında genellikle kullanılmaktadır. Bununla ilgili NB, LR, DT ve KNN algoritmaları uygulanan yaklaşımlardan bazılarıdır. Bu çalışma kapsamında seçilen algoritmalar spam e-posta ile ilgili sınıflandırma problemine uyarlanmıştır. Veri seti çeşitliliği, veri seti sayısının farklılığı ve ham/spam oranının farklılığı üzerine çalışma gerçekleştirilmiştir. Bu kriterlerin başarım performansına etkisi tartışılmıştır. SpamAssassin, Enron 4, Enron 5, Enron 6 ve CS440/ECE448 veri setleri kullanılarak yaklaşık 25.000 e-posta deney verisi test edilmiştir. Proje çalışmasında kullanılan spam külliyatı hem numerik hem alfabetik bileşenleri içermektedir. Öncelikle sayısal içerikler ve problemin tanımlanmasında önemli olmadığı düşünülen karakterler temizlenerek önişlem adımları gerçekleştirilmiştir. Sonuçların doğru tahmin edilmesi açısından ön işlem adımlarının uygulanması çalışmada ele alınmıştır. Karşılaştırmada doğruluğu etkin kılmak için her algoritma ilgili veri setleri üzerinde sırasıyla 10'ar kez çalıştırılmış ve sonuçlar istatistiksel bilgilere göre kaydedilmiştir.

Python programlama dili ile Jupyter Notebook ortamında yazılan algoritmalarla gerçekleştirilen testlerde genel anlamda hazır kütüphane yapıları kullanılmadan tüm çalışmalar gerçekleştirilmiştir. Test veri setleri üzerinde uygulanan algoritmalarla doğruluk, kesinlik, duyarlılık ve F1-skor ölçütlerine göre detaylı performans değerlendirmeleri yapılmıştır. Elde edilen başarımlara göre farklı metotlarla kıyaslamalar yapılmış ve karşılaştırmalı değerlendirmeler sunulmuştur.

Tablo 11. Enron veri seti ile karşılaştırma (Comparison with Enron Dataset)

Enron Veri Seti					
Çalışmalar	*Algoritma*	*Doğruluk*	*Kesinlik*	*Duyarlılık*	*F1-skor*
[40]	NB	%84,98	%90,01	%36,75	-
	KNN	%78,22	%85,89	%38,06	-
[11]	NB	%93,08	%95,1	**% 93,1**	**% 93,6**
[34]	NB	%79,2	%95,6	%69	%80,1
	DT	%93,4	%93,4	%93,4	%93,4
	KNN	%89,8	%89,8	%89,9	%89,8
[21]	NB	-	%74,02	%64,34	%68,84
	DT	-	%82	%82	%82
[22]	NB	%93,26	-	-	-
	DT	%96,05	-	-	-
[55]	NB	**% 95,48**	%97	%68	%80
	DT	%90,90	%83	%41	%55
Enron 4	NB	%75,83	%98,49	%67,7	%80,24
	LR	%99,08	%98,97	%99,77	%99,37
	DT	**% 97**	**% 97,71**	%98,16	**% 97,93**
	KNN	%96,16	%97,68	%97,01	%97,34
Enron 5	NB	%78,35	**% 99,8**	%69,11	%81,66
	LR	%99,42	%99,44	%99,72	%99,58
	DT	%96,03	%96,96	%97,36	%97,16
	KNN	**% 98,45**	%98,89	**% 98,89**	**% 98,89**
Enron 6	NB	%72,16	%99,26	%62,06	%76,37
	LR	%97,83	%97,41	%99,65	%98,52
	DT	%95	%95,71	%97,47	%96,58
	KNN	%94,41	%96,95	%95,28	%96,11
CS440/ECE448	NB	%78,49	%96,72	%60,82	%74,68
	LR	%98,92	%97,97	%100	%98,97
	DT	%94,62	%91,42	**% 98,96**	%95,04
	KNN	%97,84	**% 100**	%95,87	%97,89

Tablo 12. SpamAssassin veri seti ile karşılaştırma (Comparison with SpamAssassin Dataset)

SpamAssassin Veri Seti					
Çalışmalar	*Algoritma*	*Doğruluk*	*Kesinlik*	*Duyarlılık*	*F1-skor*
[10]	NB	%99,46	%99,66	%98,46	-
[40]	NB	%86,54	%86,23	%34,38	-
	KNN	%77,63	%81,64	%37,86	-
[22]	KNN	%96,20	%87	%97,14	-
	DT	%98,47	-	-	-
Proje çalışmasında	NB	%83,47	%87,96	%58,2	%70,05
	LR	%94,54	%90,19	%93,78	%91,95
	DT	%91,65	%83,51	%93,28	%88,13
	KNN	%90,57	%98,32	%72,88	%83,71

Veri setlerinin değerlendirme metriklerine göre kaydedilen sonuçlarında LR algoritmasının genel olarak veri setlerinde daha iyi bir performans gösterdiği gözlemlenmiştir. En iyi performansın CS440/ECE448 verisinde %100'lük tam başarı oranına göre KNN algoritmasının kesinlik; LR algoritmasının ise duyarlılık metriklerinde ulaştığı gözlemlenmiştir. Çalışma kapsamında ele alınan yöntemlerin Enron 5 veri seti üzerindeki testlerde uygun performanslara ulaştıkları anlaşılmıştır. Buna ek olarak Enron 5 veri setinde üzerinde test edilen NB yaklaşımı ile kesinlik ölçütüne göre %99,8 oranında iyi performansa ulaşılmıştır. Ham ve spam oranlarının farklılık gösterdiği veri kümelerinde başarım performansları ile ilgili detaylı değerlendirmeler yapılmıştır. %71 oranında spam verisine sahip olan Enron 5 veri kümesi üzerindeki testlerde %31 spam verisi bulunduran SpamAssassin veri setindeki testlere göre daha başarılı sonuçlara ulaşıldığı görülmüştür. Eğitim ve test veri setleri olarak kullanılan verilerin spam oranın daha fazla olması bilgisinin sınıflandırmanın doğru yapılabilmesi açısından önemli olduğu anlaşılmıştır. Bu çalışmada veri sayısının fazla olması veri içerisinde

seçilmiş olan özelliklerin ham/spam olarak sınıflamada etkisi ve ham/spam oranında spam verilerinin fazla olması sınıflandırmada önemli etkiye sahip olduğu gözlemlenmiştir. Karar ağacı algoritmasında farklı özellik seçimi yöntemlerinin kullanılması Enron 4 veri setinin daha iyi sonuçlar vermesini sağlamıştır. Enron 5 veri setinin SpamAssassin veri setine göre spam veri oranın fazla olması sınıflandırmada önemli etkiye sahip olmuştur. Genel olarak algoritmalarda Enron 5 veri setinin iyi sonuçlar vermesi bu çıkarımı doğrulamaktadır. Burada mevcut performanslara ek olarak bazı ek ön işlem adımlarının geliştirilmesiyle çalışma performansının artırılabileceği düşünülmektedir. Ön işlem adımlarından sonra özellik seçiminde farklı algoritmalar kullanılarak özellik seçiminin etkisi üzerine çalışma geliştirilebilir. Ayrıca ele alınan algoritmaların bazı mekanizmalarının ortak kullanıldığı hibrit yöntemleri içeren yeni çalışmalar da yapılabilir. Veri setinin eğitim ve test veri setleri olarak ayrılmasında farklı yaklaşımlar kullanılarak bunun performansa etkisi analiz edilebilir.

KAYNAKLAR (REFERENCES)

[1] J. Hong, "The State of Phishing Attacks", *Communications of the ACM*, 55(1), 74-81, 2012.

[2] E. M. Rudd, A. Rozsa, M. Günther, T. E. Boult, "A Survey of Stealth Malware Attacks, Mitigation Measures, and Steps Toward Autonomous Open World Solutions", *IEEE Communications Surveys & Tutorials*, 19(2), 1145-1172, 2016.

[3] S. Ergin, S. Işık, "The İnvestigation on the Effect of Feature Vector Dimension for Spam Email Detection with a New Framework", **In 2014 9th Iberian Conference on Information Systems and Technologies (CISTI)**, IEEE, 1-4, 2014.

[4] M. E. Maron, "Automatic İndexing: an Experimental İnquiry", *Journal of the ACM (JACM)*, 8(3), 404-417, 1961.

[5] J. R. Anderson, M. Matessa, "Explorations of an İncremental, Bayesian Algorithm for Categorization", *Machine Learning*, 9(4), 275-308, 1992.

[6] D. D. Lewis, W. A. Gale, "A Sequential Algorithm for Training Text Classifiers", *SIGIR'94*. Springer, London, 3-12, 1994.

[7] J. R. Quinlan, "Generating Production Rules from Decision Trees", *ijcai*., 87, 304-307, 1987.

[8] T. Cover, P. Hart, "Nearest Neighbor Pattern Classification", *IEEE Transactions on Information Theory*, 13(1), 21-27, 1967.

[9] L. Melian, A. Nursikuwagus, "Prediction Student Eligibility in Vocation School with Naïve-Byes Decision Algorithm", **IOP Conference Series: Materials Science and Engineering**, Bandung, Indonesia, 407(1), 012140, 9 May 2018.

[10] W. A. Awad, S. M. ELseuofi, "Machine Learning Methods for Spam E-Mail Classification", *International Journal of Computer Science & Information Technology (IJCSIT)*, 3(1), 173–184, 2011.

[11] A. Sharaff, N. K. Nagwani, A. Dhadse, "Comparative Study of Classification Algorithms for Spam Email Detection", *Emerging research in computing, information, communication and applications*, Springer, New Delhi, 237–244, 2016.

[12] T. Lv, P. Yan, H. Yuan, W. He, "Spam Filter Based on Naive Bayesian Classifier", **Journal of Physics: Conference Series,** Zhejiang, China, 1575(1), 012054, 22-23 May 2020.

[13] M. Raza, N. D. Jayasinghe, M. M. A. Muslam, "A Comprehensive Review on Email Spam Classification using Machine Learning Algorithms", **2021 International Conference on Information Networking (ICOIN),** IEEE, Jeju Island, Korea (South), 327-332, 13-16 January 2021.

[14] A. Junnarkar, S. Adhikari, J. Fagania, P. Chimurkar, D. Karia, "E-Mail Spam Classification via Machine Learning and Natural Language Processing", **2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV),** IEEE, Tirunelveli, India, 693-699, 4-6 February 2021.

[15] B. Ahmed, "Wrapper Feature Selection Approach Based on Binary Firefly Algorithm for Spam E-mail Filtering", *Journal of Soft Computing and Data Mining*, 1(2), 44-52, 2020.

[16] R. Nayak, S. A. Jiwani, B. Rajitha, "Spam Email Detection using Machine Learning Algorithm", *Materials Today: Proceedings*, 2021.

[17] G. Salton, C. S. Yang, C. T. Yu, "Contribution to the Theory of Indexing", *Cornell University*, 1973.

[18] İnternet: D. Galanis, J. Koutsikakis, Natural Language Proc. Group, nlp.cs.aueb.gr/software_and_datasets/Enron-Spam/index.html, 16.11.2021.

[19] İnternet: I. Androutsopoulos, aueb.gr/users/ion/data/lingspam_public, 09.11.2021.

[20] İnternet: I. Androutsopoulos, aueb.gr/users/ion/publications.html, 23.11.2021.

[21] D. Gaurav, S. M. Tiwari, A. Goyal, N. Gandhi, A. Abraham, "Machine Intelligence-Based Algorithms for Spam Filtering on Document Labeling", *Soft Computing*, 24(13), 9625-9638, 2020.

[22] S. Gibson, B. Issac, L. Zhang, S. M. Jacob, "Detecting Spam Email with Machine Learning Optimized with Bio-Inspired Meta-Heuristic Algorithms", *IEEE Access*, 8, 187914- 187932, 2020.

[23] N. F. Rusland, N. Wahid, S. Kasim, H. Hafit, "Analysis of Naïve Bayes Algorithm for Email Spam Filtering Across Multiple Datasets", **IOP Conference Series: Materials Science and Engineering**, Melaka, Malaysia, 226(1), 6–7 May 2017.

[24] B. K. Dedeturk, B. Akay, "Spam Filtering Using a Logistic Regression Model Trained by an Artificial Bee Colony Algorithm", *Applied Soft Computing*, 91, 106229, 2020.

[25] İnternet: C. Özdemir, UCI Machine L. Repository, https://archive.ics.uci.edu/ml/datasets/Turkish+Spam+V01, 16.10.2021.

[26] İnternet: M. Kirk, Github, github.com/hexgnu/spam_filter/tree/master/data, 22.11.2021.

[27] G. Salton, C. S. Yang, "On the Specification of Term Values in Automatic Indexing", *Journal of Documentation*, 29(4), 351-372, 1973.

[28] F. Jánez-Martino, E. Fidalgo, S. González-Martínez, J. Velasco-Mata, "Classification of Spam Emails Through Hierarchical Clustering and Supervised Learning", *arXiv preprint arXiv:2005.08773*, 2020.

[29] S. Isik, Z. Kurt, Y. Anagun, K. Ozkan, "Recurrent Neural Networks for Spam E-mail Classification on an Agglutinative Language", *International Journal of Intelligent Systems and Applications in Engineering*, 8(4), 221-227, 2020.

[30] İnternet: G. V. Cormack, T. R. Lynam, TREC 2007 Public Corpus, https://plg.uwaterloo.ca/cgi-bin/cgiwrap/gvcormac/foo07, 22.11.2021.

[31] E. Ezpeleta, I. Velez de Mendizabal, J. M. G. Hidalgo, U. Zurutuza, "Novel Email Spam Detection Method using Sentiment Analysis and Personality Recognition", *Logic Journal of the IGPL*, 28(1), 83-94, 2020.

[32] M. Bassiouni, M. Ali, E. A. El-Dahshan, "Ham and Spam E-Mails Classification using Machine Learning Techniques", *Journal of Applied Security Research*, 13(3), 315-331, 2018.

[33] İnternet: M. Hopkins, E. Reeber, G. Forman, J. Suermondt, UCI Machine Learning Repository, archive.ics.uci.edu/ml/datasets/Spambase, 18.10.2021.

[34] A. I. Taloba, S. S. I. Ismail, "An Intelligent Hybrid Technique of Decision Tree and Genetic Algorithm for E-Mail Spam Detection", **2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)**, Cairo, Egypt, 99–104, 8-10 December 2019.

[35] A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti, M. Alazab, "A Comprehensive Survey for Intelligent Spam Email Detection", *IEEE Access*, 7, 168261-168295, 2019.

[36] S. Nandhiniand, J. M. KS. "Performance Evaluation of Machine Learning Algorithms for Email Spam Detection", **2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)**, IEEE, Vellore, India, 1-4, 24-25 February 2020.

[37] M. Yağanoğlu, E. Irmak, "Separation of Incoming E-Mails Through Artificial Intelligence Techniques", *Avrupa Bilim ve Teknoloji Dergisi*, (21), 690-696, 2021.

[38] İnternet: Tiago A. Almeida, UCI Machine Learning Repo., archive.ics.uci.edu/ml/datasets/sms+spam+collection, 18.10.2021.

[39] I. Čavor, "Decision Tree Model for Email Classification", **2021 25th International Conference on Information Technology (IT)**, IEEE, Zabljak, Montenegro, 1-4, 16-20 February 2021.

[40] T. Kumaresan, S. Sanjushree, K. Suhasini, C. Palanisamy, "Image spam filtering using support vector machine and particle swarm optimization", **National Conference on Information Processing and Remote Computing(NCIPRC),** 17-21, 2015.

[41] J. Batra, R. Jain, V. A. Tikkiwal, A. Chakraborty, "A Comprehensive Study of Spam Detection in E-Mails Using Bio-Inspired Optimization Techniques", *International Journal of Information Management Data Insights*, 1(1), 100006, 2021.

[42] M. Al-Tahrawi, M. Abualhaj, S. Al-Khatib, "Polynomial Neural Networks Versus Other Spam Email Filters: An Empirical Study", *TEM Journal*, 9(1), 136-143, 2020.

[43] S. Amjad, F. S. Gharehchopogh, "A Novel Hybrid Approach for Email Spam Detection Based on Scatter Search Algorithm and K-Nearest Neighbors", *Journal of Advances in Computer Engineering and Technology*, 5(3), 181-194, 2019.

[44] G. Al-Rawashdeh, R. Mamat, N. H. B. Abd Rahim, "Hybrid Water Cycle Optimization Algorithm with Simulated Annealing for Spam E-Mail Detection", *IEEE Access*, 7, 143721-143734, 2019.

[45] İnternet: Kaggle, www.kaggle.com, 15.10.2021.

[46] İnternet: Apache SpamAssassin, *spamassassin.apache.org/old/publiccorpus*, 04.12.2021.

[47] V. Metsis, I. Androutsopoulos, G. Paliouras, "Spam Filtering with Naive Bayes-Which Naive Bayes?", **CEAS 2006 - Third Conference on Email and Anti-Spam**, Mountain View, California, USA, 17, 28-69, 27-28 July 2006.

[48] İnternet: I. Androutsopoulos, http://www2.aueb.gr/users/ion/data/enron-spam, 09.11.2021.

[49] İnternet: K. Studer, The Grainger College of Engineering, https://courses.grainger.illinois.edu/cs440/fa2018/MPs/mp4/assignment4.html, 02.12.2021.

[50] K. A. Vidhya, G. Aghila, "A Survey of Naïve Bayes Machine Learning Approach in Text Document Classification", *(IJCSIS) International Journal of Computer Science and Information Security*, 7(2), 206-211, 2010.

[51] Z. Jorgensen, Y. Zhou, M. Inge, "A Multiple Instance Learning Strategy for Combating Good Word Attacks on Spam Filters", *Journal of Machine Learning Research*, 9(6), 1115-1146, 2008.

[52] S. Ergin, S. Işık, "The Assessment of Feature Selection Methods on Agglutinative Language for Spam Email Detection: A Special Case for Turkish", **In 2014 IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA) Proceedings,** IEEE, 122-125, June 2014.

[53] L. H. Lee, C. H. Wan, T. F. Yong, H. M. Kok, "A Review of Nearest Neighbor-Support Vector Machines Hybrid Classification Models", *Journal of Applied Sciences*, 10(17), 1841-1858, 2010.

[54] H. Satılmış, S. Akleylek, "IoT Güvenliği İçin Kullanılan Makine Öğrenimi ve Derin Öğrenme Modelleri Üzerine bir Derleme", *Bilişim Teknolojileri Dergisi*, 14(4), 457-481, 2021.

[55] A. Junnarkar, S. Adhikari, J. Fagania, P. Chimurkar, D. Karia, "E-Mail Spam Classification via Machine Learning and Natural Language Processing", **2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)**, IEEE, Tirunelveli, India, 693-699, 4-6 February 2021.

[56] Z. Yong, L. Youwen, X. Shixiong, "An Improved KNN Text Classification Algorithm Based on Clustering", *Journal of computers*, 4(3), 230-237, 2009.