Mesaj Metinlerinin Sınıflandırılması

Yazılım Mühendisliği Ana Bilim

Dalı Dönem Projesi

Caner GÜLCAN

Proje Danışmanı: Prof. Dr. Aytuğ ONAN

Ocak 2024

Mesaj Metinlerinin Sınıflandırılması

Öz

Günümüzde kısa mesaj(SMS) üzerinden iletişim kurmak yaygınlaşmıştır. Bu yaygınlaşmanın sonucu olarak gelen kutusundaki kısa mesaj sayısı oldukça artmıştır. Zaman zaman hatalı sınıflandırmalar olsa da günümüzde telefonlarda ki istenmeyen(spam) mesaj filtereleri oldukça başarılı şekilde çalışmaktadır. Bu çalışma kapsamında UCI platformunda paylaşılmış olan SMS Spam Collection adındaki veri seti kullanılarak metin sınıflandırma işlemi gerçekleştirilecektir. Veri seti üzerinde metin ön işleme adımları uygulanacaktır. Makine öğrenmesi modelleri olarak Karar Destek Vektörü, Rastgele Orman, Multinominal Naive Bayes, Karar Ağacı, K-En Yakın Komşu kullanılacaktır. Makine öğrenmesi algoritmalarında hiper parametre seçimi yapılmıştır. Çapraz doğrulama k=10 seçildiği durumda Karar Destek Vektörü ile %98 oranında doğru sınıflandırma başarımı gözlemlenmiştir.

Anahtar Sözcükler: Metin Sınıflandırma, metin madenciliği, makine öğrenmesi, hiper parametre seçimi, karar destek vektörü

Classification of Message Texts

Abstract

Nowadays, communicating via short message service (SMS) has become widespread. As a result of this proliferation, the number of short messages in the inbox has significantly increased. Despite occasional misclassifications, spam message filters on phones today work quite successfully. In this study, text classification will be performed using the SMS Spam Collection dataset shared on the UCI platform. Text preprocessing steps will be applied to the dataset. Machine learning models such as Support Vector Machine, Random Forest, Multinomial Naive Bayes, Decision Tree, and K-Nearest Neighbors will be used. Hyperparameter tuning has been performed for the machine learning algorithms. With 10-fold cross-validation, a classification accuracy of 98% was observed with the Support Vector Machine.

Keywords: Text Classification, text mining, machine learning, hyperparameter tuning, support vector machine

Bu proje çalışmasını:

Bana her zaman destek olan aileme,

İthaf ediyorum.

Teşekkür

Proje çalışmasında bana destek olan aileme, proje danışmanı hocam Aytuğ Onan'a teşekkür ederim.

İçindekiler

Şekiller Listesi

Tablolar Listesi

Kısaltmalar Listesi

BoW	Bag of Words
IDF	Inverse Document Frequency
İKÇÜ	İzmir Kâtip Çelebi Üniversitesi
K-NN	K-Nearest Neighbors
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
ORCID	Open Researcher and Contributor ID
RIPPER	Repeated Incremental Pruning to Produce Error Reduction.
SMS	Short Message Service
SVM	Support Vector Machines
TF	Term Frequency

Bölüm 1

Giriş

Bu proje çalışmasında makine öğrenmesinin alt kategorisi olan denetimli öğrenme konusunda çalışma gerçekleştirilecektir. Denetimli öğrenme etiketleri bilinen veriler üzerinde yapılan çalışma türüdür. Denetimli öğrenmenin alt alanı olan sınıflandırma problemine çözüm aranacaktır. Sınıflandırma probleminde ise metin verilerinin sınıflandırılması, doğal dil işleme alanında çalışma gerçekleştirilecektir.

Veri seti olarak UCI'da (2012) senesinde paylaşılmış olan SMS Spam Collection veri seti üzerinde çalışma gerçekleştirilecektir[1]. Veri seti içerisinde toplam 5572 örnek yer almaktadır. Bu örneklerin 4825 adeti ham(geçerli), 747 adeti spam(istenmeyen) şeklinde etiketlenmiştir. Veri seti içerisinde eksik veri bulunmamaktadır. Almeida vd. (2011) SMS Spam Collection veri setini gerçek SMS mesajlarından çalışma kapsamında oluşturarak veri seti üzerinde Karar Destek Vektörü ile %97.64 oranında doğruluk başarımı elde etmiştir[2].

Proje çalışmasında amacımız SMS mesajlarını doğruluk oranı en yüksek olacak şekilde sınıflandırmaktır. İşlenmemiş SMS mesajları üzerinde öncelikli olarak metin ön işleme adımları uygulanmıştır. Metin ifadelerinin makine öğrenmesi modelinde kullanılabilmesi amacıyla sayısal verilere, vektörlere dönüştürmek için sayım vektörleme (BoW) ve kelime temsili olarak unigram, bigram, trigram kelime temsili yaklaşımları kullanılmıştır. Bununla birlikte TF-IDF vektörleme öznitelik seti olarak kullanılacaktır.

Makine öğrenmesi sınıflandırma algoritmaları olarak Karar Destek Vektörü, Rastgele Orman, Multinominal Naive Bayes, Karar Ağacı ve K-En yakın komşu algoritmaları kullanılacaktır. Algoritmalara doğruluk başarımını arttırmaya yönelik hiper parametre seçimi gerçekleştirilmiştir. Modellerin başarımlarının ölçülmesinde k=10 Çapraz doğrulama sonuçları esas olarak alınmıştır. Modellerin doğruluk(accuracy), kesinlik(precision), duyarlılık(recall) ve F1-ölçütü(F1 Score) değerleri raporlanmıştır.

Çalış vd. (2013) Türkçe 400 normal, 400 reklam içerikli e-posta ile toplam 800 örnekten oluşan veri seti toplamışlardır. Toplanan bu veri seti üzerinde reklam epostalarını K-NN algoritması ile %96.5 doğruluk oranında sınıflandırma başarısı elde etmiştir[3].

Sasaki ve Shinnou (2005) spam ve geçerli e-posta sınıflandırmada k-means ortalamalar algoritması kullanarak e-posta mesajlarını kümelemiştir. Naive Bayes ve Destek Vektör Makineleri ile yapılan sınıflandırma işlemi gerçekleştirip sonuçları raporlamıştır. Destek Vektör Makineleri ile %97 doğruluk oranında başarılı sınıflandırma işlemi gerçekleştirmiştir[4].

Androutsopoulos vd. (2000) spam ve geçerli e-posta filtreleme çalışması gerçekleştirmiştir. Naive Bayes sınıflandırma algoritması ile Ling-Spam veri seti üzerinde farklı öznitelik setleri için sınıflandırma işlemi gerçekleştirmiştir. Çapraz doğrulama k=10 seçildiği durumda ağırlıklı başarım olarak %99.5 oranında doğruluk başarımı elde etmiştir[5].

Cohen (1996) spam SMS filtreleme çalışmasında TF-IDF ve RIPPER (Repeated Incremental Pruning to Produce Error Reduction) algoritmasının karşılaştırılmasını yapmış sonuçlarını paylaşmıştır[6]. RIPPER: IREP algoritmasının geliştirilmesi ile elde edilmiş kural tabanlı sınıflandırma algoritmasıdır. Temel olarak inşa sürecinde kuralların oluşturulup budama işleminin yapılması ile algoritma işletilir[7].

Li vd. (2002) spam mesajlarının filtrelenmesinde karar destek vektörleri(SVM) algoritmasınnı detaylı bir şekilde açıklamasını gerçekleştirmiştir[8].

Drucker (1999) geçerli ve spam mesajlarının sınıflandırılmasında karar destek vektörlerini(SVM), Ripper algoritmasını, Boosting karar ağaçlarını uygulamıştır. Farklı öznitelik setleri ile sınıflandırma performanslarını hata oranları(error rate) şeklinde çalışmalarında paylaşmıştır. En düşük hata oranını Boosting ve terim sıklığı (TF) ile 0.0180 şeklinde elde etmiştir[9].

Lai ve Tsai (2004) makine öğrenmesi yöntemleri ile spam ve geçerli e-posta mesajlarını sınıflandırmıştır. Farklı ön işleme adımları üzerinde Naive Bayes(NB), K-NN, SVM algoritmaları ile sınıflandırma işlemi gerçekleştirmiştir. En başarılı sınıflandırmayı TF-IDF öznitelik setinde Naive Bayes algoritması ile %95.30 doğruluk oranında elde etmiştir[10].

Lin ve Pantel (1998) spam filtrelemesine yönelik yapmış oldukları çalışmada Naive Bayes algoritması ile %92 doğruluk oranında başarım elde etmiştir[11].

Androutsopoulos vd. (2006) spam ve geçerli e-posta sınıflandırmada genellikle kullanılan Naive Bayes algoritmasının hangi türünün bu problemde daha iyi performans göstereceğine yönelik çalışma gerçekleştirmiştir. Gaussian, Bernolli ve Multinominal Naive Nayes yöntemlerini sınıflandırmada kullanmıştır. En iyi sonucu Multinominal Naive Bayes ile %97.53 doğruluk oranında başarım elde etmiştir[12].

Rahman ve Qamar (2016) metin sınıflandırmada 3 farklı veri seti üzerinde Naive Bayes, Multinominal Naive Bayes, K-NN, SVM, Bernoulli Naive Bayes(BN) ve Karar Ağacı algoritmasını kullanmıştır. Her bir veri seti için algoritmaların performans sonuçlarını tablo şeklinde paylaşmıştır[13]. Algoritmaların sınıflandırma performanslarının farklı veri setleri üzerinde oldukça farklı sonuç ürettiği gözlemlenmiştir.

Eryılmaz vd. (2020) spam mesajlarının tespit edilmesine yönelik yapmış oldukları çalışmada “TurkishEmail” veri seti üzerinde MLP algoritması ile %98 doğruluk oranında başarım elde etmiştir[14].

Ablel-Rheem vd. (2020) spam mesaj sınıflandırılmasına yönelik UCI'da paylaşılmış olan Spambase veri seti üzerinde sınıflandırma işlemi gerçekleştirmiştir. Hybrid Ensemble ile ağırlıklı ortalama F-ölçütü ile %94 oranında başarım elde etmiştir[15].

Bassiouni vd. (2018) makine öğrenmesi yöntemleri kullanarak UCI'da paylaşılmış olan Spambase veri seti üzerinde çalışma gerçekleştirmiştir. Rastgele Orman algoritması ile %95.45 doğruluk oranında başarım elde etmiştir[16].

Özdemir vd. (2013) spam e-postalar üzerine Türkçe veri seti üzerinde yapmış oldukları çalışmada Bayes ve liner olmayan sınıflandırıcıların sınıflandırma performanslarını incelemiştir. CSCA algoritması ile %86 oranında doğruluk başarımı elde etmiştir[17]. CSCA algoritması giriş verileri ve karşılık gelen sınıfı gösteren bir etiket özniteliklerini temsil eden d boyutlu vektör tarafından oluşan antikorlar tarafından temsil edilir[18].

Güven (2023) Türkçe e-posta spam sınıflandırılmasında temel makine öğrenmesi modelleri ve dil modellerinin sınıflandırma başarımlarını karşılaştırmıştır. Proje çalışmasında BERT ve ELECTRA dil modelleri ile %94.08 oranında doğruluk başarımı elde etmiştir[19].

Parlak (202) Türkçe e-posta veri seti üzerinde 3 farklı öznitelik seçimi metodu ile Naive Bayes, K-NN, Destek Vektör Makineleri algoritmalarının sınıflandırma performanslarını karşılaştırmıştır. En başarılı F skor değerini %0.991 olarak Gini Endeksi(GI) ve Naive Bayes kombinasyonu ile elde etmiştir[20].

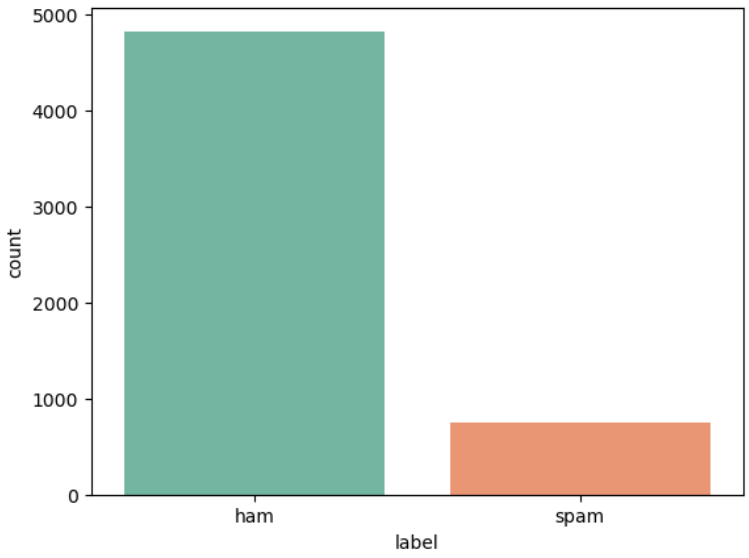
Bölüm 2

Yöntemler

2.1 Veri Seti

Proje çalışması için hazır bir veri seti kullanılmıştır. Bu veri seti UCI'da 2012 senesinde paylaşılmış olan SMS Spam Collection veri setidir. Veri seti içerisinde iki kolon bulunmaktadır. Bunlardan birisi SMS mesajlarının yer aldığı metin ifadesinin kolonudur. Diğer kolon ise sınıf etiketlerinin yer aldığı etiket kolonudur. İki farklı sınıf etiketi ile veriler etiketlenmiştir. Bu etiketler ham(geçerli) ve spam(istenmeyen) şeklindedir. Veri seti içerisinde eksik satır, kolon yoktur.

Veri seti içerisinde toplam 5572 adet örnek yer almaktadır. Örneklerin 4825 adeti ham(geçerli), 747 adeti spam(geçersiz) SMS mesajlarından oluşmaktadır. Aşağıdaki Şekil 2.1'de veri seti içerisindeki sınıf etiketlerine göre sayım grafiği gösterilmiştir.



Şekil 2. 1: Veri Setindeki Etiketlere Göre Örnek Sayıları

2.2 Veri Ön İşleme

Proje içerisindeki mesaj kolonundaki SMS mesajları için ön işlenmemiş metin ifadeleri oluklarını söyleyebiliriz. Doğal dil işleme çalışmalarında metinlerin öznitelikleri çıkarılmadan önce metin ön işleme adımları gerçekleştirilir. Yapılacak olan proje ile ve veri setindeki örneklere göre bu ön işleme adımları projeden projeye farklılık gösterebilmektedir.

Veri seti içerisindeki ilk SMS mesaj örneği aşağıdaki gibidir:
'Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat....'

Uygulanan Metin Önişleme Adımları

1. **Küçük Harfe Çevirme:** Tüm metni küçük harfe çevirerek büyük/küçük harf duyarlılığını ortadan kaldırırız.
2. **Sayıları Kaldırma**: Sayılar bazen yararlı olabilir, ancak genellikle metin sınıflandırma için gereksizdir.
3. **Noktalama İşaretlerini Kaldırma:** Noktalama işaretleri genellikle metin sınıflandırma modelleri için gürültü oluşturabilir.
4. **Durak Kelimelerini Kaldırma** (Stop Words Removal): İngilizce'de cümle içerisinde sıkça geçen fakat anlamsal olarak katkısı olmayan ifadelerdir. Örnek olarak "am" ,"is", "are" vs. verilebilir.

Yukarıdaki 4 ön işleme adımı proje kapsamında sırasıyla uygulanmıştır.

TF-IDF öznitelik setinde sayıların silinmediği durumda 9437 öznitelik sayısı silindiğinde 8501'e düşmüştür. Sınıflandırma başarımlarında çok fark olmamıştır. Bu nedenle sayılar veri ön işleme adımında silinmiştir.

Genellikle uygulanan ön işleme adımı olan kökten ayırma (Lemmatization) projede uygulanmamıştır. SMS mesajları içerisinde çok fazla kısaltılmış ifade olması köke inmenin başarılı şekilde yapılmasını zorlaştırmaktadır. Kelimeler köklerine indirildiğinde sınıflandırma başarımının düştüğü gözlemlendiği için bu adım çıkarılmıştır.

Kelime durak kelimelerinin kaldırılması için NLTK kütüphanesinde yer alan İngilizce Stopwords(durak kelimeleri) listesi kullanılmıştır. Durak kelimeleri listesindeki ilk 10 örnek 'i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're" şeklindedir.

2.3 Öznitelik Çıkarımı

Öznitelik çıkarımı, metin verilerini makine öğrenimi modelleri için uygun biçimde sayısal vektörlere dönüştürme işlemidir. Öznitelikler ön işlenmiş metin ifadelerinden çıkarılmaktadır. Proje kapsamında uygulamış olduğum öznitelik çıkarımları yaklaşımları aşağıda belirtilmiştir.

2.3.1 Kelime Çantası (BOW)

Kelime Çantası (BOW), metin belgelerinden özellikleri ayıklama yöntemidir. Ayrıca bu özellikler, makine öğrenimi algoritmalarını eğitmek için kullanılabilir. Kelime Çantası, Eğitim veri kümesindeki tüm belgede bulunan tüm benzersiz sözcüklerin sözlüğünü oluşturur[21].

Proje kapsamında kelime çantası N-gram (unigram, bigram, trigram) şeklinde ayrı ayrı ele alınmıştır.

2.3.2 TF-IDF (Term Frequency- Inverse Document Frequency)

Her kelimenin belge içindeki sıklığı ile tüm dökümanlar içinde nadir olma derecesinin çarpımını alır. TF temel olarak bir kelimenin metin içerisinde geçme sıklığıdır. IDF değeri ise bu kelimenin bütün metinler içerisindeki geçme sıklığıdır. TF ve IDF değerlerini hesaplamak için kullanılan formüller Denklem (2.1) ve Denklem (2.2)'de verilmiştir [22-24].

Her bir kelime için TF ve IDF değerleri bulunduktan sonra, Denklem (2.3)'te verilen formül yardımıyla her bir kelimenin ağırlığı hesaplanarak doküman terim matrisi oluşturulur [23,24].

Her bir kelime için yukarıdaki işlemlerin yapılmasından sonra, Doküman Terim Matrisi (Document Term Matrix – DTM) Şekil (2.2)'de gösterildiği gibi oluşturulmuş olur. T değerleri bir terimi, D değeri ise bir dokümanı simgelemektedir.

Denklem 2. 1

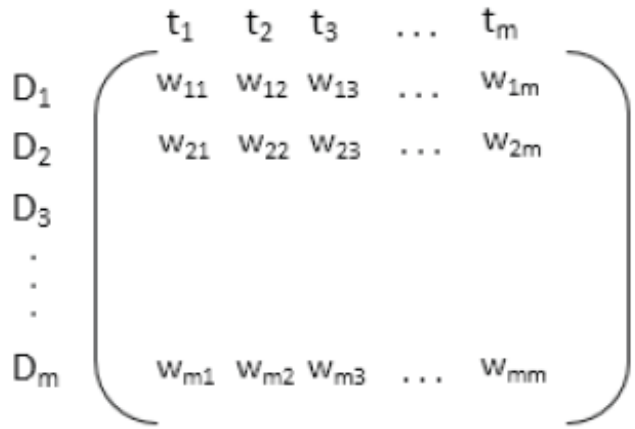
$$TF_{(d,m)} = \frac{\text{m kelimesinin d dokümanında geçme sayısı}}{\text{Dokümandaki toplam kelime sayısı}}$$

Denklem 2. 2

$$IDF_{(m)} = \ln \frac{\text{Vektor modelindeki toplam dokuman sayısı}}{\text{İçerisinde m kelimesi bulunduran toplam dokuman sayısı}}$$

Denklem 2. 3

$$w_{(d,m)} = TF_{(d,m)} * TDF_{(m)}$$



Şekil 2. 2 Döküman Terim Matrisi

2.4 Python ve Kütüphaneleri

Python sahip olduğu hazır kütüphaneler sayesinde veri bilimi alanında günümüzde popüler bir şekilde kullanılmaktadır. Proje kapsamında Python 3.7 sürümü kullanılmıştır[25].

Proje kapsamında kullanılan Python kütüphaneleri:

- Numpy
- Pandas
- Matplotlib
- Seaborn
- Scikit-learn
- Re
- NLTK

2.5 Sınıflandırma Yöntemleri

Proje çalışması içerisinde 5 farklı makine öğrenmesi algoritması sınıflandırma işlemi için kullanılmıştır. Bu 5 farklı makine öğrenmesi algoritmasına hiper parametre seçimi işlemi gerçekleştirilmiştir. Hiper parametre belirleme, kısaca, bir makine öğrenimi modelinin en iyi hiper parametreler kombinasyonunun belirlenmesi işlemidir. Bu işlem, birçok hiper parametre kombinasyonunun test edilmesini gerektirir[26]. Proje içerisinde hiper parametrelerin belirlenmesinde Gridsearch hiper parametre belirleme yöntemi uygulanmıştır.

2.5.1 Makine Öğrenmesi Modelleri

- **Rastgele Orman(RF)**

Rastgele Orman algoritması makine öğrenmesinin alt dalı olan denetimli öğrenmede oldukça popüler olan sınıflandırma ve regresyon problemlerinde kullanılan bir algoritmadır. Rastgele Orman sınıflandırıcısı farklı boyutlardaki karar ağaçlarının bir araya gelmesiyle oluşmaktadır. Her bir karar ağacı, veri setinin rastgele bir alt örneği üzerinde eğitilir ve bu süreçte her düğümde de rastgele bir öznitelik alt kümesi kullanılır. Bu çeşitlendirme, modellerin birbirinden bağımsız olmasını sağlar ve bu sayede modelin genelleme yeteneğini artırır. Sonuç olarak her bir ağacın çıktısı ortalama alınarak veya çoğunluk oyu ile nihai tahmin yapılarak belirlenir.

- **Multinominal Naive Bayes (MNB)**

Multinomial Naive Bayes algoritması, özellikle metin sınıflandırma gibi ayrık veri kümeleriyle çalışmak için kullanılan bir olasılıksal sınıflandırma yöntemidir. Naive Bayes ilkesine dayanmaktadır. Bu nedenle her özelliğin bağımsız olduğunu varsayar. Multinomial Naive Bayes, sınıflandırılacak her sınıf için özelliğin meydana gelme olasılığını hesaplar ve bu olasılıkları kullanarak en yüksek olasılığa sahip sınıfı seçer. Özellikle kelime sıklığına dayalı olarak çalıştığından, metin verilerinde kelime sayısını veya kelime frekansını kullanarak sınıflandırma yapar.

- **Karar Destek Vektörleri (SVM)**

Karar Destek Vektörleri, makine öğrenmesinde lineer ve lineer olmayan sınıflandırma ve regresyon problemlerinde kullanılan güçlü bir algoritmadır. Karar Destek vektörleri, verileri sınıflandırmak için en iyi ayrımı sağlayan bir hiper düzlem bulmayı hedefler. Bu hiper düzlem, sınıflar arasındaki marjini maksimize edecek şekilde konumlandırılır. Sınıflandırma işlemi sırasında, marjinleri belirleyen kritik veri noktalarına destek vektörleri denir. SVM, doğrusal olarak ayrılabilen verilerde doğrudan hiper düzlem kullanırken, doğrusal olmayan veriler için çekirdek (kernel) triklerini kullanarak verileri daha yüksek boyutlu uzaylara dönüştürür ve bu şekilde ayrım sağlar.

- **Karar Ağacı (DT)**

Karar ağacı algoritması, hem sınıflandırma hem de regresyon problemlerinde kullanılan basit ama güçlü bir makine öğrenimi yöntemidir. Algoritma verileri karar düğümleri aracılığıyla dallara ayırarak sınıflandırma veya tahmin yapar. Her düğümde, veri seti belirli bir özelliğe göre iki veya daha fazla alt kümeye ayrılır. Bu işlem, veri seti tamamen sınıflandırılana veya belirli bir durma kriterine ulaşılana kadar devam eder. Karar ağacı, verileri kolayca anlaşılabilir ve görselleştirilebilir bir yapıya dönüştürdüğünden, özellikle karmaşık karar verme süreçlerini açıklamada faydalıdır.

- **K-En Yakın Komşu (K-NN)**

K-En Yakın Komşu algoritması, makine öğrenmenin alt dalı olan denetimli öğrenmede hem sınıflandırma hem de regresyon problemlerinde kullanılan basit ve etkili bir yöntemdir Algoritma çalıştırılırken bir k değeri belirlenir. K değerinin anlamı incelenecek komşu sayısıdır. Bu algoritma, bir veri noktasının sınıfını veya değerini belirlemek için en yakın k sayıdaki komşularının etiketlerine veya değerlerine bakar. KNN, eğitim verisini doğrudan kullanır ve her bir yeni veri noktası için en yakın K komşusunu belirleyerek sınıflandırma veya tahmin yapar. Yeni bir değer en yakın k kadar eleman alınarak gelen değerler arasındaki uzaklık hesaplanır. Uzaklık hesaplama yönteminde birçok yöntem olsa da genellikle Öklid uzaklığı yöntemi kullanılır. Uzaklıklar hesaplandıktan sonra sıralanır ve yeni değer en uygun olan sınıf etkiketi ile etiketlenir.

2.5.2 Modellerin Hiper Parametre Seçimi

Proje içerisinde kullanılan makine öğrenmesi modelleri için scikit-learn kütüphanesi içerisindeki GridSearch yöntemi kullanılmıştır. Hiper parametre seçimi yapıldıktan sonra modellerin belirlenen parametreleri aşağıdaki Tablo (2.1)'de gösterilmiştir.

Tablo 2. 1 Makine Öğrenmesi Modellerinin Parametreleri

	Belirlenen Parametreler
Rastgele Orman (RF)	'balanced_subsample', 'entropy', 'max_depth : 14'
Multinominal Naive Bayes (MNB)	'alpha : 0.1'
Destek Vektör Makineleri(SVM)	'linear'
Karar Ağacı (DT)	'ccp_alpha : 0.001', 'entropy' , 'max_depth :17', 'best'
K-En Yakın Komşu(K-NN)	'n_neighbors : 1', 'p : 2', 'uniform'

2.6 Kullanılan Metrikler

2.6.1 Doğruluk (Accuracy)

Model başarımının ölçülmesinde kullanılan en popüler ve basit yöntem, modele ait doğruluk oranıdır. Doğru sınıflandırılmış örnek sayısının (TP +TN), toplam örnek sayısına (TP+TN+FP+FN) oranıdır. İlgili hesaplama aşağıdaki Denklem (2.4)'te gösterilmiştir.

Denklem 2. 4

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

2.6.2 Kesinlik (Precision)

Kesinlik, sınıfı 1 olarak tahmin edilmiş True Pozitif (TP) örnek sayısının, sınıfı 1 olarak tahmin edilmiş tüm örnek sayısına (TP+FP) oranıdır [27]. İlgili hesaplama aşağıdaki Denklem (2.5)'te gösterilmiştir.

Denklem 2. 5

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

2.6.3 Duyarlılık (Recall)

Doğru sınıflandırılmış pozitif örnek (TP) sayısının, toplam pozitif örnek sayısına (TP+FN) oranıdır.[27]. İlgili hesaplama aşağıdaki Denklem (2.6)'da gösterilmiştir.

Denklem 2. 6

$$\text{Duyarlılık} = \frac{TP}{(TP + FN)}$$

2.6.4 F1-Ölçütü (F1 Score)

Doğru Kesinlik ve duyarlılık ölçütleri tek başına anlamlı bir karşılaştırma sonucu çıkarmamıza yeterli değildir. Her iki ölçütü beraber değerlendirmek daha doğru sonuçlar verir. Bunun için f-ölçütü (F) tanımlanmıştır. F1-ölçütü, kesinlik (K) ve duyarlılığın (D) harmonik ortalamasıdır[27]. İlgili hesaplama aşağıdaki Denklem (2.7)'de gösterilmiştir.

Denklem 2. 7

$$F_1 = \frac{2DK}{(D + K)}$$

Bölüm 3

Bulgular

Proje içerisinde ön işlenmiş ve öznitelikleri çıkarılmış olan metin verilerine ilişkin sınıflandırma işlemi gerçekleştirilmiştir. Veri seti içerisinde 2 farklı sınıf etkileti olduğu için ikili (binary) sınıflandırma yapıldığı söylenebilir. Makine öğrenmesi modellerinin sonuçlarında k=10 çapraz doğrulama sonuçları dikkate alınacaktır. Modellerin performansları doğruluk, kesinlik, duyarlılık, f1 ölçütü şeklinde incelenmiştir.

3.1 SMS Geçerli(Ham) , Geçersiz(Spam) Sınıflandırma Sonuçları

Aşağıdaki Tablo (3.1)'de kelime çantası(bow) ve unigram temsili ile modellerin k=10 çapraz geçerleme performansları gösterilmiştir. Doğruluk ve F1 ölçütü göz önüne alındığında en başarılı sınıflandırma işlemini Karar Destek Vektörleri(SVM) gerçekleştirmiştir.

Tablo 3. 1: Countvector k=10 Çapraz Doğrulama Model Performansları

	Accuracy	Precision	Recall	F1 Score
(RF)	%97	%97	%81	%88
(MNB)	%97	%84	%94	%89
(SVM)	%98	%98	%87	%92
(DT)	%96	%94	%74	%83
(K-NN)	%95	%100	%66	%80

Aşağıdaki Tablo (3.2)'de kelime çantası(bow) ve bigram temsili ile modellerin k=10 çapraz geçerleme performansları gösterilmiştir. Bigram temsili ile bütün modellerin sınıflandırma performanslarında düşüş gözlemlenmiştir. Özellikle F1 ölçüt değerlerine bakıldığı zaman en iyi F1 ölçütü, unigram temsili ile sınıflandırmada ki en kötü F1 ölçütü ile eşit olduğu gözlemlenmiştir.

Tablo 3. 2: CountVector+Bigram k=10 Çapraz Doğrulama Model Performansları

	Accuracy	Precision	Recall	F1 Score
(RF)	%93	%99	%48	%65
(MNB)	%54	%23	%99	%37
(SVM)	%95	%100	%67	%80
(DT)	%91	%97	%33	%49
(K-NN)	%94	%100	%58	%47

Aşağıdaki Tablo (3.3)'de kelime çantası(bow) ve trigram temsili ile modellerin k=10 çapraz geçerleme performansları gösterilmiştir. Unigram, bigram ve trigram kelime temsil yaklaşımları içerisinde en kötü sınıflandırma performansları trigram kelime temsilinde gerçekleşmiştir.

Tablo 3. 3: CountVector+Trigram k=10 Çapraz Doğrulama Model Performansları

	Accuracy	Precision	Recall	F1 Score
(RF)	%90	%100	%29	%45
(MNB)	%42	%19	%99	%31
(SVM)	%94	%100	%58	%73
(DT)	%89	%100	%17	%29
(K-NN)	%94	%100	%55	%71

Aşağıdaki Tablo (3.4)'te TF-IDF temsili ile modellerin k=10 çapraz geçerleme performansları gösterilmiştir. TF-IDF öznitelik setinde doğruluk ve f1 ölçütü değerlerine bakıldığında en başarılı sınıflandırma performansını Destek Vektör Makineleri göstermiştir.

Tablo 3. 4: TF-IDF k=10 Çapraz Doğrulama Model Performansları

	Accuracy	Precision	Recall	F1 Score
(RF)	%97	%97	%81	%88
(MNB)	%98	%90	%92	%91
(SVM)	%98	%99	%86	%92
(DT)	%96	%88	%79	%83
(K-NN)	%95	%100	%64	%78

Bölüm 4

Sonuçlar

Bu çalışmada UCI'da paylaşılmış olan SMS Spam Collection veri seti üzerinde doğal dil işleme çalışması gerçekleştirilmiştir. Denetimli öğrenmenin alt kategorisi olan sınıflandırma problemi ile ilgili çalışma gerçekleştirilmiştir. Veri setindeki SMS mesajları ön işleme adımları gerçekleştikten sonra öznitelikleri çıkarılmıştır. Bu öznitelik setleri N-gram kelime çantası(bow) ve TF-IDF şeklindedir.

Öznitelik setlerinde makine öğrenmesi modelleri sonuçlarına bakıldığında en başarılı sınıflandırma performansları **unigram** ve **TF-IDF** üzerinde olmuştur. Bigram ve trigram öznitelik setlerinde modellerin performansları ciddi oranda düşmüştür.

Toplamda 5 farklı makine öğrenmesi modeli kullanılmıştır. Bu modeller Rastgele Orman(RF), Multinominal Naive Bayes(MNB), Destek Vektör Makineleri(SVM), Karar Ağacı(DT), K-En Yakın Komşu(K-NN) şeklindedir. Her bir makine öğrenmesi modeli için gridsearch yöntemi ile hiper parametre seçimi gerçekleştirilmiştir.
Modellerin performanslarına bakıldığında en iyi sınıflandırma performansını **Destek Vektör Makineleri(SVM)** gerçekleştirmiştir.

Unigram öznitelik setinde: SVM %**98** doğruluk(accuracy) oranında sınıflandırma başarımı gerçekleştirmiştir. Bununla birlikte SVM %**98** kesinlik(precision), %**87** duyarlılık(recall), **%92** F1 ölçütü sonucu elde etmiştir.

TF-IDF öznitelik setinde: SVM %**98** doğruluk(accuracy) oranında sınıflandırma başarımı gerçekleştirmiştir. Bununla birlikte SVM %**99** kesinlik(precision), %**86** duyarlılık(recall), **%92** F1 ölçütü sonucu elde etmiştir.

Karar Destek Vektörleri(SVM) Unigram ve TF-IDF öznitelik setlerinde oldukça yakın ve başarılı sınıflandırma performansı göstermiştir.

Kaynaklar

[1] Internet: UCI Veri seti, https://archive.ics.uci.edu/dataset/228/sms+spam+collection

[2] Almeida, T. A. Hidalgo, J. M. G., & Yamakami, A. (2011, September). Contributions to the study of SMS spam filtering: new collection and results. In *Proceedings of the 11th ACM symposium on Document engineering* (pp. 259-262). https://doi.org/10.1145/2034691.2034742

[3] Çalış, K. Gazdağı, O., & Yıldız, O. (2013). Reklam içerikli epostaların metin madenciliği yöntemleri ile otomatik tespiti. Bilişim Teknolojileri Dergisi, 6(1), 1-7.

[4] Sasaki, M. & Shinnou, H. (2005, November). Spam detection using text clustering. In 2005 International Conference on Cyberworlds (CW'05) (pp. 4-pp). IEEE.

[5] Androutsopoulos, I. Koutsias, J., Chandrinos, K. V., Paliouras, G., & Spyropoulos, C. D. (2000). An evaluation of naive bayesian anti-spam filtering. arXiv preprint cs/0006013.

[6] Cohen, W. W. (1996, March). Learning rules that classify e-mail. In AAAI spring symposium on machine learning in information access (Vol. 18, p. 25).

[7] Koşan, M. A., Yıldız, O., & Karacan, H. (2018). Kimlik avı web sitelerinin tespitinde makine öğrenmesi algoritmalarının karşılaştırmalı analizi. Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 24(2), 276-282.

[8] Li, K. L., Li, K., Huang, H. K., & Tian, S. F. (2002, November). Active learning with simplified SVMs for spam categorization. In Proceedings. International Conference on Machine Learning and Cybernetics (Vol. 3, pp. 1198-1202). IEEE.

[9] Drucker, H., Wu, D., & Vapnik, V. N. (1999). Support vector machines for spam categorization. IEEE Transactions on Neural networks, 10(5), 1048-1054.

[10] Lai, C. C., & Tsai, M. C. (2004, December). An empirical performance comparison of machine learning methods for spam e-mail categorization. In Fourth International Conference on Hybrid Intelligent Systems (HIS'04) (pp. 44-48). IEEE.

[11] Lin, D. & Pantel, P. (1998, July). Spamcop: A spam classification & organization program. In Proceedings of AAAI-98 workshop on learning for text categorization (pp. 95-98).

[12] Androutsopoulos, I, Metsis, V., & Paliouras, G. (2006, July). Spam filtering with naive bayes-which naive bayes?. In CEAS (Vol. 17, pp. 28-69).

[13] Rahman, A., & Qamar, U. (2016, August). A Bayesian classifiers based combination model for automatic text classification. In 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS) (pp. 63-67). IEEE.

[14] Eryılmaz, E. E., Şahin, D. Ö., & Kılıç, E. (2020). Türkçe istenmeyen e-postaların farklı öznitelik seçim yöntemleri kullanılarak makine öğrenmesi algoritmaları ile tespit edilmesi. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 13(2), 57-77.

[15] Ablel-Rheem, D. M., Ibrahim, A. O., Kasim, S., Almazroi, A. A., & Ismail, M. A. (2020). Hybrid feature selection and ensemble learning method for spam email classification. International Journal, 9(1.4), 217-223.

[16] Bassiouni, M., Ali, M., & El-Dahshan, E. A. (2018). Ham and spam e-mails classification using machine learning techniques. Journal of Applied Security Research, 13(3), 315-331.

[17] Özdemir, C., Ataş, M., & Özer, A. B. (2013, April). Classification of Turkish spam e-mails with artificial immune system. In 2013 21st Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.

[18] Oliveira, L. O. V. B., Mota, R. L. M., & Barone, D. A. C. (2012, June). Clonal selection classifier with data reduction: Classification as an optimization task. In 2012 IEEE Congress on Evolutionary Computation (pp. 1-7). IEEE.

[19] Güven, Z. A. (2023). Türkçe e-postalarda spam tespiti için makine öğrenme yöntemlerinin ve dil modellerinin analizi. Avrupa Bilim ve Teknoloji Dergisi, (47), 1-6.

[20] Parlak, B. Makine Öğrenmesi Algoritmaları kullanarak Türkçe istenmeyen e-posta filtreleme Turkish spam e-mail filtering using Machine Learning Algorithms.

[21] Kumar, N., & Sonowal, S. (2020, July). Email spam detection using machine learning algorithms. In 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 108-113). IEEE.

[22] Kılınç, D., Bozyiğit, F., Kut, A., & Kaya, M. (2015). Overview of source code plagiarism in programming courses. International Journal of Soft Computing and Engineering (IJSCE), 5(2), 79-85.

[23] Can, F., Kocberber, S., Balcik, E., Kaynak, C., Ocalan, H. C., & Vursavas, O. M. (2008). Information retrieval on Turkish texts. Journal of the American Society for Information Science and Technology, 59(3), 407-421.

[24] Salton, G. (1975). A vector space model for information retrieval. Journal of the ASIS, 613-620.

[25] Internet: Python https://www.python.org/downloads/release/python-370/ 04.06.2024

[26] Emeç, M., & ÖZCANHAN, M. H. Makine Öğrenmesi Algoritmalarında Hiper Parametre Belirleme.

[27] Nizam, H., & Akın, S. S. (2014). Sosyal medyada makine öğrenmesi ile duygu analizinde dengeli ve dengesiz veri setlerinin performanslarının karşılaştırılması. XIX. Türkiye'de İnternet Konferansı, 1(6), 873-883.