T.C.
BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ

YAPAY ZEKA DERSİ FİNAL PROJESİ

SPAM E-POSTA SINIFLANDIRMA

Şule MEŞE
191502001

Bilgisayar Mühendisliği Bölümü

ARALIK 2023

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 lojistik regresyon, k-en yakın komşu algoritmalarının ilgili probleme uyarlanması ve karşılaştırmalı olarak analiz edilmesi amaçlanmıştır.Bu kapsamda algoritmalar veri setinde kullanılmıştır. Farklı algoritmalarla farklı değerlendirme metrikleri sonuçları elde edilmiştir. Bu sonuçlarının karşılaştırması yapılarak tablolar halinde sunulmuştur. En iyi başarım performansının veri seti üzerindeki testlere göre lojistik regresyon algoritmasıyla elde edildiği gözlemlenmiştir.

Giriş

Bu rapor, e-posta spam filtresi tasarımı ve analizi üzerine odaklanmaktadır. Temel hedef, "Message" sütunundaki e-posta içeriklerini analiz ederek, bu e-postaları "spam" veya "ham" kategorisine ayırt etmektir.

Problem Tanımı

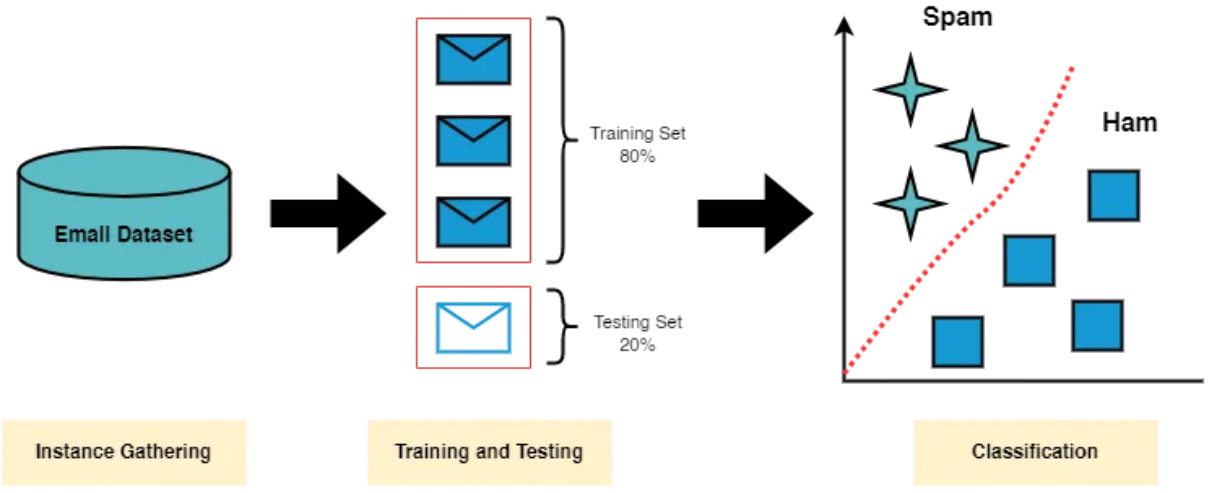
E-posta spam filtresi tasarımı, "Message" sütunundaki e-posta içeriklerine dayalı olarak e-postaları iki ana kategoriye ayırmayı amaçlar: "spam" ve "ham". Spam e-postalar istenmeyen, zararlı veya rahatsız edici içeriklere sahiptir, bu nedenle bu e-postalar doğru bir şekilde tanımlanmalı ve filtrelenmelidir.

Yöntem

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, stopwords kelimeler, köklendirme süreçleridir. Ele alınan algoritmalar, probleme uyarlanarak 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 5 tekrarla çalıştırılmıştır. Kullanılan sayısal değerler değiştirilmeden çalıştırılmıştır. Elde edilen sonuçlar Tablo üzerinde gösterilmiştir.

Deneysel çalışmalar ön işlem adımları ile başlamıştır. Bu adım, veri setinin temin edilmesi ve veri setinin eğitime hazır hale 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. Test adımında algoritma performanslarının ölçülebilmesi için doğruluk, kesinlik, duyarlılık ve F1-skor değerlendirme metrikleri hesaplanmıştır.



1. Veri Seti (*Dataset*)

Çalışmada, Kaggle platformunda erişime açık, dili İngilizce olan bir veri seti kullanılmıştır. Kullanılan veri setinde 4825 ham ve 747 spam elektronik posta örneği bulunmaktadır. Veri seti içeriği ".csv" formatında olup iki sütundan oluşmaktadır. İlk sütunda "ham" veya "spam" şeklinde, elektronik postanın spam mı yoksa normal mi olduğunu belirten etiket, diğer sütunda ise elektronik posta içeriği bulunmaktadır. Çalışmada ilgili veri setinin %80'i eğitim, %20'si ise test verileri olarak kullanılmıştır. Veri setinde ham/spam oranında ham veriler daha fazladır.

Veri Seti	Ham	Spam	Spam Oranı
mail_data.csv	4825	747	%13,4

2. Veri Keşfi (Data Exploration)

Veri keşfi (data exploration), bir veri setini anlama ve içerdiği bilgileri ortaya çıkarma sürecidir.

2.1. Veri Setinin Yüklenmesi

Projenin başlangıcında CSV dosyasındaki veriler okunarak bir DataFrame'e dönüştürülür. Elde edilen dataframe iki ana sütundan oluşmaktadır. İlk sütun olan "Category", e-posta mesajlarının kategorisini belirtir. Bu kategoriler "ham" ve "spam" olarak iki temel değere sahiptir. "ham" değeri, bir e-postanın spam olmadığını, "spam" değeri ise bir e-postanın spam olduğunu gösterir. İkinci sütun ise "Message" olarak adlandırılmıştır ve e-posta mesajlarının metin içeriğini içerir. Her bir satır, bir e-posta mesajını temsil eder.

```
     Category                                            Message
0         ham  Go until jurong point, crazy.. Available only ...
1         ham                      Ok lar... Joking wif u oni...
2        spam  Free entry in 2 a wkly comp to win FA Cup fina...
3         ham  U dun say so early hor... U c already then say...
4         ham  Nah I don't think he goes to usf, he lives aro...
...       ...                                                ...
5567     spam  This is the 2nd time we have tried 2 contact u...
5568      ham              Will ü b going to esplanade fr home?
5569      ham  Pity, * was in mood for that. So...any other s...
5570      ham  The guy did some bitching but I acted like i'd...
5571      ham                        Rofl. Its true to its name

[5572 rows x 2 columns]
```

2.2.Veri Seti İncelenmesi:

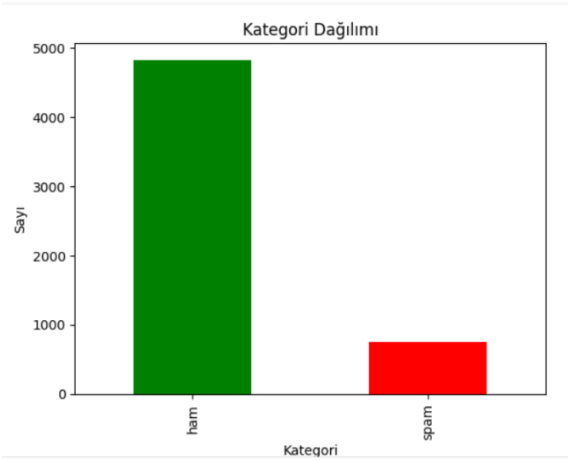
Veri setinin incelenmesi, projenin devamındaki veri analizi ve makine öğrenimi aşamalarına geçiş için temel bir adımdır. Bu aşamada veri setinin ilk birkaç satırı incelendi. Ekran Çıktısı, her bir satırın bir e-posta mesajını temsil ettiği ve bu mesajların kategorilerinin (spam veya ham) belirtildiği bir yapıya sahiptir. Bu adımı gerçekleştirerek, veri setindeki örnek verilere hızlı bir göz atıldı ve verilerin genel yapısı anlaşıldı. Bu sayede, projenin ilerleyen aşamalarında kullanılacak olan veri setinin temel özellikleri hakkında bir ön fikir elde edildi.

2.3. Kategorilerin Dağılımının İncelenmesi

Veri setinin kategorik dağılımını incelediğinizde, "Category" sütununda iki farklı değer olduğunu gözlemlendi. Bunlar "ham" ve "spam" olarak 2 türdür. Bu iki kategoriye ait örnek sayılarını içeren çıktı ham kategorisinde 4825 spam kategorisinde ise 747 adettir . Bu çıktı, veri setinizdeki e-posta mesajlarının kategorilerine göre dağılımını göstermektedir. "ham" kategorisi, spam olmayan e-postaları temsil ederken, "spam" kategorisi spam e-postaları temsil etmektedir. Çıktıdan elde edilen sonuçlar analiz edilirse sınıf dengesizliği olduğu tespit edilir.

2.4. Kategorilerin Dağılımının Görselleştirilmesi

Bu aşamada, e-posta kategorilerinin dağılımını daha iyi anlamak için bir çubuk grafiği oluşturuldu. Grafik incelenerek "ham" kategorisindeki e-posta sayısının "spam" kategorisine göre oldukça fazla olduğunu gözlemlendi. Bu durum, veri setinin dengesiz olduğunu ve sınıflar arasında belirgin bir sayısal farkın olduğunu göstermektedir.



Şekil 1.Kategori Dağılımını Gösteren Çubuk Grafiği

2.5. Eksik Değerlerin İncelenmesi

Bu aşama veri setinde eksik değer olup olmadığını kontrol etmek amacıyla gerçekleştirildi.Sonuçlar incelendiğinde "Category" ve "Message" sütunlarında herhangi bir eksik değer bulunmadığı tespit edildi. Yapılan bu kontrol, veri setinizin eksiksiz olduğunu ve her iki sütunda da herhangi bir eksik veya boş veri olmadığını doğrulamak adına önemli bir adımdır.

3. Veri Ön İşleme (Data Preprocessing)

Spam tespitinde başarılı bir model oluşturmanın ilk ve temel adımı elde bulunan verilen ön işlemden geçirilmesidir. Veri ön işleme, veri setini temizleme ve modele uygun hale getirme sürecidir. Bu aşamada yapılan işlemler elektronik postaların “spam” ya da “normal” olarak sınıflandırılmasındaki başarı oranının artırılmasına yöneliktir.

Ön işlem adımları 6 başlık altında toplanmıştır. Her bir temel konunun açıklaması ilgili başlıklarda sunulmuştur.

- [0-9] gibi sayısal ifadelerin kaldırılması (Character Manipulation)
- "#", "%", "&" gibi özel karakterlerin kaldırılması (Character Manipulation)
- Büyük harflerin küçük harfe dönüştürülmesi (Character Manipulation)
- Etkisiz kelimelerin (Stopwords) çıkarılması
- Kelime köklerinin bulunması
- Label Encoding

3.1. Sayısal İfadelerin Kaldırılması

Veri setindeki metinlerdeki sayısal ifadeler, genellikle spam analizi gibi metin madenciliği görevleri için önemli değildir. Bu nedenle, bu ifadeleri kaldırmak, modelin metni daha iyi anlamasına yardımcı olabilir. Bu amaçla sayısal ifadeler silinerek kelimeler üzerinde sınıflandırma işlemleri yapılmıştır.

3.2. Özel Karakterlerin Kaldırılması

Metin verilerindeki özel karakterler, modelin kelime veya cümle yapılarını anlamasını zorlaştırabilir. Özel karakterlerin kaldırılması, metin verilerindeki gereksiz veya anlamsız karakterleri temizleyerek, modelin daha iyi öğrenmesine ve daha doğru sonuçlar üretmesine yardımcı olur. Bu amaçla *"@', '|', '<', '>', '#', '$', '%"* ve bunlar gibi özel karakterler çıkarılmıştır.

3.3.Büyük Harflerin Küçük Harfe Dönüştürülmesi

Metindeki tüm harfleri küçük harfe dönüştürmek, modelin metni büyük ve küçük harf duyarlılığı olmadan öğrenmesine yardımcı olur.Model metni daha iyi anlar. 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 harfler küçük harflere dönüştürülerek standartlaştırma yapılmıştır.

3.4. Etkisiz Kelimelerin (Stopwords) Çıkarılması

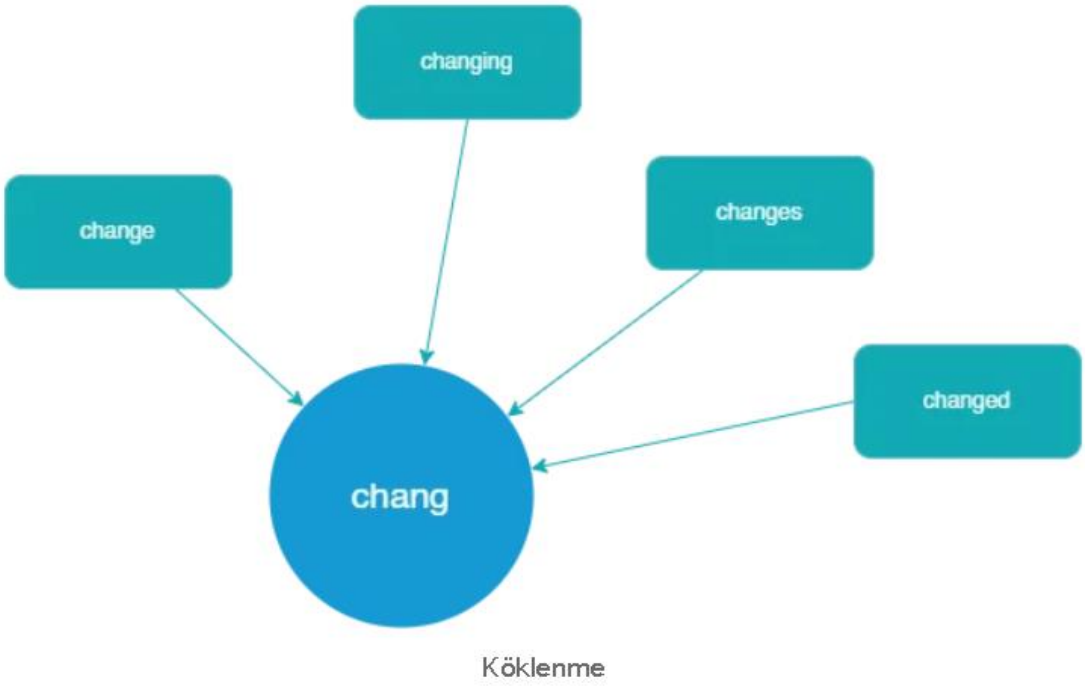
Dil içerisinde sıkça kullanılan ancak genellikle anlam taşımayan kelimeler (stopwords), modelin performansını olumsuz etkileyebilir. Etkisiz kelimeler hem ham hem de spam maillerde yaygın olarak kullanıldığından ham ve spam ayrımı yaparken etkisi olmamaktadır. Bu tür kelimelerin kaldırılması, modelin daha önemli bilgileri öğrenmesine odaklanmasını sağlar. Metni daha anlamlı kılar. Projede kullanılan veri seti İngilizce kökenli olduğu için etkisiz kelimeler listesi olarak 'english' kütüphanesi kullanılmıştır.

3.5. Kelime Köklerinin Bulunması

Stemming, kelime köklerini bulma işlemidir ve metindeki kelimeleri temel köklerine indirgeme sürecidir. Stemming, kelime köklerini kullanarak benzer anlamlı kelimelerin ortak bir temsilini elde eder. Bu, özellikle büyük veri setlerinde boyutu azaltmaya yardımcı olur.
Spam analizi gibi görevlerde, kelime kökleri genellikle bir kelimenin temel anlamını korur. Bu nedenle, kelime köklerini kullanmak, anlamı bozmadan metni daha basitleştirebilir. Stemming,

metin verilerindeki kelime çeşitliliğini azaltarak dil modellemesi görevlerinde kullanışlı olabilir.

Köklendirme işlemleri için Python üzerinde 'Porter Stemmer' kütüphanesi kullanılmıştır. Bu ön işlemin amacı e-posta içerisinde geçen kelime sayısını azaltmaktır. Bu işlemler, sınıflandırma yapılırken e-postanın ham ve spam sınıflarından hangisine ait olduğunu tespit etmede kolaylık sağlamaktadır.



Şekil 2. Köklendirme İçin Örnek

3.6. Label Encoding:

Label encoding, kategorik verileri sayısal formata dönüştürerek, bu verilerin makine öğrenimi modelleri tarafından işlenebilir hale gelmesini sağlar.

Veri setindeki kategorik olan "Category" sütunundaki verileri ("ham" ve "spam" gibi) sayısal değerlere dönüştürmek için kullanılır. Gerçekleştirilen label encoding işlemi, "spam" etiketlerini 0'a ve "ham" etiketlerini 1'e dönüştürmüştür. Bu sayede, "Category" sütunundaki etiketler artık sayısal formatta temsil edilmektedir. Örneğin, bir model "spam" etiketini 0 olarak, "ham" etiketini 1 olarak algılayabilir.

4. Veri Setinin X ve Y ye bölünmesi:

Bu aşama, veri setini bağımlı değişken (hedef) ve bağımsız değişkenler (özellikler) olarak ayırmayı amaçlar. Veri seti eğitime uygun formata getirilir. Bu projede "Category" sütunu bağımlı değişken (y) olarak kabul edilebilirken, "Message" sütunu bağımsız değişken (X) olarak kabul edilir.

5. Özellik Çıkarımı (Feature Extraction):

Özellik çıkarımı için TF-IDF yöntemi ele alınmıştır.

TF-IDF (Term Frequency-Inverse Document Frequency), metin madenciliği ve bilgi algoritması gibi alanlarda sıkça kullanılan bir metin madenciliği yöntemidir. Bu yöntem, bir belgedeki bir kelimenin önemini ölçmeye ve belge kümesi içindeki bir kelimenin genel önemini belirlemeye çalışır. TF-IDF, özellikle belge sınıflandırma, metin özetleme ve bilgi çıkarma gibi uygulamalarda yaygın olarak kullanılır.

Terim Frekansı (TF- Term Frequency), bir belgedeki belirli bir terimin (kelimenin) ne kadar sık geçtiğini ölçer. Bir kelimenin bir belgedeki frekansı arttıkça, o kelimenin belgedeki önemi artar. Ancak, sıklıkla geçen kelimeler genellikle genel bir öneme sahip olmayabilir.

Belge Frekansı (Document Frequency), bir kelimenin belge koleksiyonundaki kaç belgede bulunduğunu ölçer. Bir kelimenin birçok belgede geçmesi durumunda, bu kelimenin genel bir öneme sahip olduğu söylenebilir.

TF-IDF, bu iki ölçümü birleştirir ve bir kelimenin bir belgedeki önemini, aynı zamanda bu kelimenin genel koleksiyondaki önemini dikkate alarak hesaplar

TF-IDF, bir kelimenin bir belgedeki sıklığı ile o kelimenin genel belge koleksiyonundaki nadirliğini dikkate alarak bir terimin ağırlığını belirler. Spam e-postalar genellikle belirli anahtar kelimeleri aşırı kullanma eğilimindedir. TF-IDF, bu tür anahtar kelimeleri belirlemede yardımcı olabilir.

Örnek olarak tfidf sonuçları yorumlanırsa ilk parametre belgenin indeksini, ikinci parametre o belgede geçen kelimenin indeksini ve karşısındaki sonuç tfidf ağırlığını vermektedir.

```
(0, 6574)     0.19702290780713422
(0, 204)      0.3546607748984538
(0, 2362)     0.1644114271559457
(0, 1043)     0.2996188558139331
(0, 773)      0.33856204558236747
(0, 3215)     0.290759085739021
(0, 6771)     0.24101894405056848
(0, 2398)     0.19739925519222162
(0, 775)      0.2996188558139331
(0, 391)      0.2696908349561709
(0, 1257)     0.27466035642293457
(0, 4537)     0.24216987541391052
(0, 3077)     0.3546607748984538
```

Şekil 3. *TF-IDF Sonuçlarından Bir Kesit*

6. Modelin Eğitilmesi

Bu başlık altında gerçekleştirilen çalışmalarda kullanılan algoritmaların çalışma mantıkları 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 tüm veriler sınıflandırıcı algoritmalar üzerinde test edilmeye hazır hale getirilmiştir.

6.1. KNN

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 1'e göre hesaplanır.

$$Xi\ ve\ Xj : Xi = (Xi1\ ,\ Xi2\ ,\ ,\ Xin\ ) \text{ ve } Xj = (Xj1\ ,\ Xj2\ ,\ \dots\ ,\ Xjn\ )$$

$$D(Xi\ ,\ Xj\ ) = \sqrt{\sum}\ ((Xik - Xjk\ ))\ \textbf{(1)}$$

KNN yönteminin en önemli dezavantajlarından biri büyük ölçekli ve yüksek boyutlu veri kümeleri için verimsiz olmasıdır. 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.

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. Farklı k değerleri verilerek tekrardan komşuların uzaklıkları hesaplanmıştır. Bu hesap yapılırken k=1 den k=20'ye kadar tüm k değerleri için eğitim ve test yapılmıştır ve en yüksek skoru veren k değeri bulunmuştur. Sonuç olarak "k=1" değeriyle daha başarılı sonuçlar elde edilmiş ve test çalışmalarında "k=1" değerinin kullanılmasına devam edilmiştir.

6.1.1. En iyi K değerinin Hesaplanması

K değerleri 1'den 20'ye kadar denemek üzere bir liste oluşturulmuştur. Her k değeri için KNN modeli eğitilmiştir. Elde edilen accuracy değerleri tablo 1'de kayıt edilmiştir. En yüksek accuracy değeri k=1 de alınmıştır. Bu nedenle k=1 değeri seçilmiştir.

K=1	0.947085201793722
K=2	0.947085201793722
K=3	0.9255605381165919
K=4	0.9255605381165919
K=5	0.9103139013452914
K=6	0.9103139013452914
K=7	0.8968609865470852
K=8	0.8968609865470852
K=9	0.8869955156950673
K=10	0.8869955156950673
K=11	0.8789237668161435
K=12	0.8789237668161435
K=13	0.874439461883408
K=14	0.874439461883408
K=15	0.8708520179372198
K=16	0.8708520179372198
K=17	0.8699551569506726
K=18	0.8699551569506726
K=19	0.8672645739910314
K=20	0.8672645739910314

Tablo 1. *(Her Bir "K" Değeri İçin Accuracy Değeri)*

6.1.2 Modelin Cross Validation Sonuçları

Her biri farklı veri parçaları üzerinde yapılan 5 katlamalı çapraz doğrulama sonuçları tablo 2'de verilen değerlere sahiptir.Tüm sonuçlar karşılaştırıldığında fold 1 için accuracy, recall, precision ve f1 score değerleri en yüksektir.

	Accuracy	Recall	Precision	F1 score
Fold 1	0.946188	1.0	0.941463	0.969849
Fold 2	0.942600	1.0	0.937803	0.967903
Fold 3	0.938061	1.0	0.933268	0.965482
Fold 4	0.965482	1.0	0.941463	0.969849
Fold 5	0.940754	0.998963	0.936831	0.966900

Tablo.2. *(Modelin Cross Validation Sonuçları)*

Accuracy	Recall	Precision	F1 score
0.94274887087295	0.9997927461139897	0.9381662513164372	0.967997129414543

Tablo 3. *(Ortalama Skor Sonuçları)*

6.1.3. Modelin Test Verileri Üzerinde Tahmin Sonuçları

Model test verileri üzerinde accuracy, recall, f1-score, precision metrikleri ile değerlendirilmiştir. Değerlendirme sonuçları tablo 4'te yer almaktadır. Model %94 doğrulukla spam mesajları doğru tahmin edebilmektedir.

Accuracy	Recall	Precision	F1 score
0.947085201793722	0.9989648033126294	0.9433040078201369	0.9703368526897939

Tablo 4. *(Modelin Değerlendirme Metrik Sonuçları)*

6.2. Lojistik Regresyon

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. Veriler eğitildikten sonra x_test ile gönderilen test verileriyle tahmini değerler elde edilmiştir. Bu tahmini değerlerle y_test değerleri karşılaştırılmıştır.

6.2.1. Modelin 5 katlı Cross Validation Sonuçları

Her biri farklı veri parçaları üzerinde yapılan 5 katlamalı çapraz doğrulama sonuçları tablo 5'te verilen değerlere sahiptir.

	Accuracy	Recall	Precision	F1 score
Fold 1	0.957847	0.998963	0.954455	0.976202
Fold 2	0.950672	1.0	0.946078	0.972292
Fold 3	0.955116	0.998963	0.951628	0.974721
Fold 4	0.948833	0.997927	0.945972	0.971255
Fold 5	0.953321	0.997927	0.950641	0.973710

Tablo5. *(Modelin Cross Validation Sonuçları)*

Accuracy	Recall	Precision	F1 score
0.9531582549049601	0.9987564766839379	0.9497553711434519	0.9736366313335465

Tablo 6. *(Ortalama Skor Sonuçları)*

6.2.2. Modelin Test Verileri Üzerinde Tahmin Sonuçları

Model test verileri üzerinde accuracy, recall, f1-score, precision metrikleri ile değerlendirilmiştir. Değerlendirme sonuçları tablo 7'de yer almaktadır.

Accuracy	Recall	Precision	F1 score
0.95695067264574	1.0	0.9526627218934911	0.9757575757575757

Tablo 7. *(Modelin Değerlendirme Metrik Sonuçları)*

7. Değerlendirme Metrikleri (Evaluation Metrics)

7.1Doğruluk (Accuracy)

Doğru tahmin edilen örneklerin toplam örnek sayısına oranını ifade eder. Genellikle dengesiz sınıflara sahip veri setlerinde tek başına yeterli olmayabilir. Accuracy açısından yapılan karşılaştırmada lojistik regresyon algoritması knn'den daha iyi çıkmıştır.

7.2. Hassasiyet (Precision)

Belirli bir sınıfa ait olarak tahmin edilen örneklerin gerçekten o sınıfa ait olma olasılığını ifade eder. Precision açısından yapılan karşılaştırmada lojistik regresyon algoritması knn'den daha iyi çıkmıştır.

7.3. Geri Çağrı (Recall veya Sensitivity)

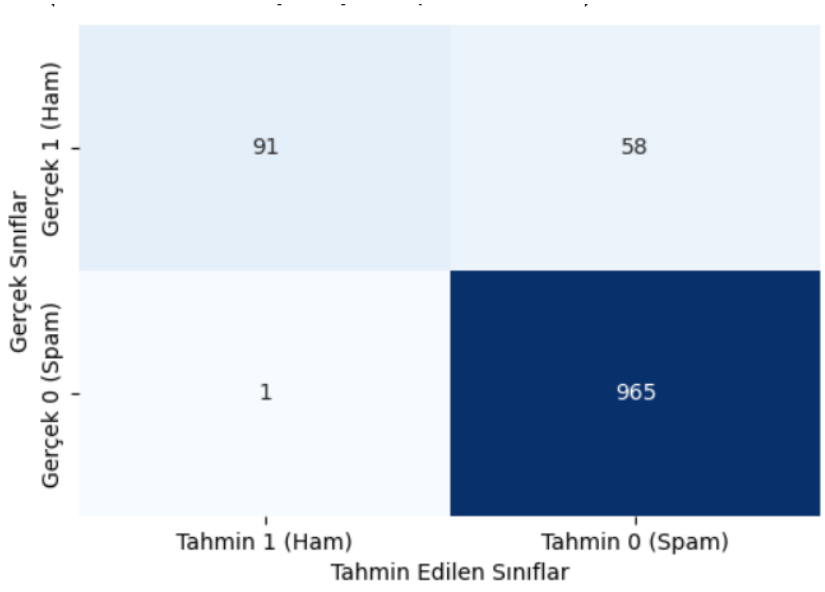
Belirli bir sınıfa ait olan tüm örneklerin kaç tanesinin doğru bir şekilde tahmin edildiğini gösterir. Recall açısından yapılan karşılaştırmada lojistik regresyon algoritması knn'den daha iyi çıkmıştır.

7.4. F1-Score

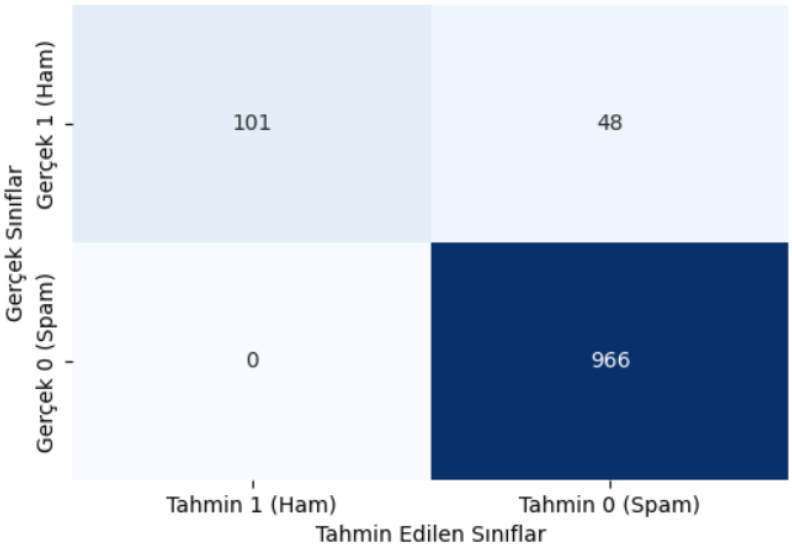
Hassasiyet ve geri çağrının harmonik ortalamasıdır. Bu metrik hem hassasiyeti hem de geri çağrıyı dengeler ve özellikle dengesiz sınıflara sahip veri setlerinde kullanışlıdır. İki model arasında F1 skorları benzer, ancak Lojistik Regresyon bir miktar daha yüksek.

8. Sonuçları Karşılaştırma

Test verileri üzerinde değerler karşılaştırıldığında lojistik regresyon algoritmasının k en yakın komşuluk algoritmasından değerlendirme metriklerine bakıldığında daha iyi olduğu sonucu elde edilir.



Şekil 4. Knn için confusion matrix



Şekil 5. Lojistik Regresyon için confusion matrix