T.C.
KASTAMONU ÜNİVERSİTESİ
MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

KONU

Makine Öğrenmesi ile Mesaj Spam Sınıflandırması

HAZIRLAYAN

214410802 – Bashar Alkhawlani

DANIŞMAN

Doç. Dr. Kemal AKYOL

Aralık – 2026

KASTAMONU

ETİK BEYAN

Kastamonu Üniversitesi, Mühendislik ve Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, Mühendislik Tamamlama Programı, Tez Hazırlama Kılavuzu'nda yer alan kurallara uygun olarak hazırladığım bu çalışmada; proje içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi, tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu, proje çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi, kullanılan verilerde herhangi bir değişiklik yapmadığımı, bu projede sunduğum çalışmanın özgün olduğunu, bildirir; aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Öğrenci Numarası : 214410802 **İmza**

Adı Soyadı : Bashar Alkhawlani

ÖZET

Mesaj Sınıflandırma

Bashar Alkhawlani

Kastamonu Üniversitesi

Mühendislik ve Mimarlık Fakültesi

Bilgisayar Mühendisliği Bölümü

Proje Danışmanı:

Doç. Dr. Kemal AKYOL

Aralık 2026, 39 sayfa

Günümüzde mobil iletişim araçlarının yaygınlaşmasıyla birlikte istenmeyen mesajlar (spam) önemli bir güvenlik ve iletişim problemi haline gelmiştir. Kullanıcıları yanıltmayı, kişisel bilgiler elde etmeyi veya maddi zarar oluşturmayı amaçlayan spam mesajların otomatik olarak tespit edilmesi, iletişim güvenliği açısından kritik bir gereklilik hâline gelmiştir. Bu projede, kısa mesajların içeriklerine göre "spam" veya "ham" olarak sınıflandırılmasını sağlayan bir makine öğrenmesi tabanlı yazılım geliştirilmesi amaçlanmıştır.

Çalışma kapsamında veri ön işleme teknikleri, TF-IDF tabanlı özellik çıkarımı, beş farklı sınıflandırıcı model, iki topluluk öğrenmesi yöntemi ve yapay sinir ağı mimarileri incelenmiş ve karşılaştırılmıştır. Modeller doğruluk, duyarlılık, özgüllük, F1-skoru ve ROC-AUC performans metrikleri açısından değerlendirilmiştir. Ayrıca en iyi iki model arasında istatistiksel anlamlılığı incelemek amacıyla McNemar testi uygulanmıştır.

Geliştirilen yazılım, kullanıcıdan alınan mesaj metnini işleyerek en yüksek performansa sahip model aracılığıyla sınıflandırma yapmakta ve mesajın spam olma olasılığını kullanıcı dostu bir arayüzde göstermektedir. Bu çalışma, kısa mesaj spam tespitinde makine öğrenmesi yöntemlerinin etkinliğini ortaya koymakta ve gerçek dünya uygulamalarına yönelik bir çözüm sunmaktadır.

Anahtar Sözcükler : Spam mesaj tespiti, Makine öğrenmesi, TF-IDF, Naive Bayes, Destek Vektör Makineleri, Topluluk öğrenmesi, Yapay sinir ağları, McNemar testi, Python

ABSTRACT

MESSAGE CLASSIFICATION

Bashar ALKHAWLANI

Kastamonu University

Faculty of Engineering and Architecture

Department of Computer Engineering

Project Advisor:

Assoc. Prof. Dr. Kemal AKYOL

December 2026, 40 pages

The widespread use of mobile communication technologies has led to a significant increase in unsolicited and potentially harmful spam messages. These messages aim to deceive users, obtain personal information, or cause financial loss, making automated spam detection an essential component of communication security. This project aims to develop a software system capable of classifying SMS messages as “spam” or “ham” using machine learning techniques.

In this study, various preprocessing methods, TF-IDF–based feature extraction, five different machine learning classifiers, two ensemble learning approaches, and an artificial neural network (ANN) model were implemented and compared. The models were evaluated using performance metrics such as accuracy, precision, recall, specificity, F1-score, and ROC-AUC. Additionally, the McNemar test was applied to determine whether the performance difference between the two best models was statistically significant.

The developed system processes user-provided message text and performs classification using the best-performing model. The prediction results, including the probability of the message being spam, are displayed through a user-friendly interface. The findings of this study demonstrate the effectiveness of machine learning methods for SMS spam detection and provide a practical solution suitable for real-world applications.

Key Words: Spam detection, Machine learning, TF-IDF, Naive Bayes, Support Vector Machines, Ensemble learning, Artificial neural networks, McNemar test, Python.

İÇİNDEKİLER **SAYFA**

1. GİRİŞ ve TANITIM

Günümüzde mobil iletişim araçlarının yaygınlaşmasıyla birlikte kısa mesaj (SMS) üzerinden gerçekleştirilen dolandırıcılık girişimleri ve istenmeyen reklam içerikleri önemli bir güvenlik ve iletişim sorunu hâline gelmiştir. Spam mesajlar; kullanıcıları yanıltmayı, kişisel bilgilerini ele geçirmeyi, ekonomik kayıplara yol açmayı veya genel kullanıcı deneyimini olumsuz etkilemeyi amaçlayan istenmeyen metinlerdir. Bu nedenle, SMS spam tespiti hem bireysel kullanıcılar hem de mobil operatörler için kritik bir ihtiyaç olarak ortaya çıkmıştır.

Makine öğrenmesi yöntemleri, büyük veri kümeleri üzerinde öğrenme ve otomatik sınıflandırma yapabilme yetenekleri sayesinde spam tespitinde etkili ve ölçeklenebilir çözümler sunmaktadır. Özellikle metin madenciliği, doğal dil işleme (NLP) ve TF-IDF tabanlı özellik çıkarımı yöntemleri, kısa mesaj içeriklerinden anlamlı temsil çıkarılmasına olanak tanımakta ve sınıflandırma performansını artırmaktadır. Bu bağlamda, farklı makine öğrenmesi algoritmalarının performanslarının karşılaştırılması ve en uygun modelin belirlenmesi, güvenilir bir spam tespit sistemi oluşturmak için büyük önem taşımaktadır.

Bu projede, kısa mesajların içeriklerine göre “spam” veya “ham” olarak sınıflandırılmasını sağlayan bir makine öğrenmesi modeli geliştirilmiştir. Çalışma kapsamında beş temel sınıflandırıcı, iki topluluk öğrenmesi algoritması ve bir yapay sinir ağı mimarisi kullanılarak kapsamlı bir performans analizi gerçekleştirilmiştir. Modeller doğruluk, duyarlılık, özgüllük, F1-skoru ve ROC-AUC gibi performans ölçütleri açısından karşılaştırılmış; ayrıca en iyi iki model arasındaki anlamlı farkı incelemek amacıyla McNemar testi uygulanmıştır.

Bu projenin bir diğer önemli çıktısı ise, kullanıcıların bir mesaj metni girerek anlık sınıflandırma yapabilmesini sağlayan kullanıcı dostu bir arayüz geliştirilmesidir. Böylece sistem sadece teorik bir çalışma olmaktan çıkıp, gerçek dünyada kullanılabilir bir uygulama hâline getirilmiştir.

Bu çalışma, SMS spam tespitinde makine öğrenmesi yöntemlerinin etkinliğini ortaya koymakta ve iletişim güvenliği alanında uygulanabilir bir çözüm sunmaktadır.

1.1. Sistemin Çalışma Mantığı

Bu projede, kısa mesajların içeriklerine göre “spam” veya “ham” olarak sınıflandırılmasını sağlayan makine öğrenmesi temelli bir yaklaşım kullanılmaktadır. Sistem, metin tabanlı verilerden anlamlı özellikler çıkaran TF-IDF yöntemi ile hazırlanan veri kümesi üzerinden eğitilmiş çeşitli sınıflandırıcı modellerden oluşmaktadır. Öncelikle büyük hacimli SMS mesaj veri kümesi üzerinde eğitim işlemi gerçekleştirilmekte; ardından veri kümesinin ayrılan test bölümü ile modellerin performansı değerlendirilmektedir. Bu süreç, sistemin öğrenme ve doğrulama aşamalarını oluşturmaktadır.

Hazırlanan veri kümesi binlerce SMS mesajından oluşmakta olup mesajların büyük çoğunluğu “ham”, geri kalan kısmı ise “spam” olarak etiketlenmiştir. Tüm mesajlar metin ön işleme adımlarından geçirilmiş, gereksiz karakterlerden temizlenmiş ve TF-IDF yöntemi ile sayısal özelliklere dönüştürülmüştür. Elde edilen bu özellikler üzerinden beş farklı makine öğrenmesi sınıflandırıcısı, iki topluluk öğrenmesi yaklaşımı ve bir yapay sinir ağı modeli eğitilmiştir. Eğitim işlemi sonucunda modellerin doğruluk, duyarlılık, özgüllük, F1-skoru ve ROC-AUC gibi performans metrikleri hesaplanmış ve karşılaştırılmıştır.

Eğitilen sistem, kullanıcı tarafından girilen yeni bir SMS mesajını aynı ön işleme adımlarından geçirerek TF-IDF vektörüne dönüştürmekte ve en yüksek performansı gösteren model aracılığıyla sınıflandırma yapmaktadır. Model, mesajın spam olma olasılığını yüzdesel olarak hesaplayarak kullanıcıya sonuç döndürmektedir. Böylece sistem, gerçek zamanlı mesaj analizi yapabilen bir spam tespit aracı hâline gelmiştir.

Sınıflandırma sonucunun üretilmesi arka planda bir dizi işlem ile gerçekleşmektedir. Girilen mesajın temizlenmesi, TF-IDF vektör uzayına dönüştürülmesi, eğitilmiş modele uygulanması ve olasılık çıktısına dönüştürülmesi bu sürecin temel adımlarını oluşturmaktadır. Bu işlemler 3. bölümde ayrıntılı olarak açıklanmıştır.

2. LİTERATÜR TARAMASI

Kısa mesaj (SMS) ve e-posta tabanlı spam tespiti, makine öğrenmesi ve doğal dil işleme alanında uzun süredir çalışılan önemli bir problemdir. Bu bölümde, SMS spam sınıflandırması ve genel spam filtreleme alanında literatürde yer alan güncel ve temel çalışmalar özetlenmektedir.

2.1.Almeida & Gómez Hidalgo (2012) – SMS Spam Collection v.1

Almeida ve Gómez Hidalgo, literatürde yaygın olarak kullanılan **SMS Spam Collection v.1** veri kümesini tanıtmış ve bu veri kümesi üzerinde çeşitli makine öğrenmesi algoritmalarını karşılaştırmıştır. Veri kümesi 5.574 İngilizce SMS'ten oluşmakta olup, bunların yaklaşık %13,4'ü spam, geri kalanı ham olarak etiketlenmiştir. Bu çalışma, SMS spam tespiti için standart bir benchmark veri kümesi sunması açısından alandaki birçok çalışmanın temelini oluşturmuştur. [2]

2.2.Support Vector Machine Based Spam SMS Detection (2018)

Tekerek, SMSSpamCollection veri kümesini kullanarak **Destek Vektör Makineleri (SVM)** tabanlı bir SMS spam tespit sistemi önermiştir. Çalışmada 10 katlı çapraz doğrulama ile SVM, farklı sınıflandırıcılarla karşılaştırılmış ve SVM'in yaklaşık **%98'in üzerinde doğruluk** ve yüksek true positive oranına ulaştığı rapor edilmiştir. Bu sonuçlar, TF-IDF gibi yüksek boyutlu metin temsilinde lineer SVM modellerinin ne kadar başarılı olabileceğini göstermektedir. [3]

2.3. Orange3 ile Türkçe ve İngilizce SMS Spam Tespiti (2019)

Örnek'in çalışmasında, **Türkçe ve İngilizce SMS mesajları** üzerinde Orange3 aracı kullanılarak spam sınıflandırması gerçekleştirilmiştir. Farklı veri kümeleri (SMS Spam Collection ve TurkishSMS) kullanılmış, Naive Bayes, Decision Tree ve SVM gibi sınıflandırıcılar karşılaştırılmıştır. Çalışma, dil farkına rağmen benzer metin madenciliği ve ön işleme adımlarının kullanılabildiğini ve SMS spam tespitinde makine öğrenmesi yöntemlerinin dil bağımsız şekilde uygulanabildiğini göstermektedir. [4]

2.4. Spam SMS Filtering Using Naive Bayes (2022)

REST Publisher tarafından yayımlanan bu çalışmada, SMS spam tespiti için **Naive Bayes** tabanlı bir yaklaşım önerilmiştir. Metin ön işleme (stopword temizleme, kök bulma) sonrası TF-IDF ve kelime frekansı temsilleri kullanılmış, Naive Bayes'in basit yapısına rağmen yüksek doğruluk ve düşük yanlış pozitif oranı elde edildiği rapor edilmiştir. Çalışma, Naive Bayes'in özellikle kısa metinler için güçlü bir baseline model olduğunu vurgulamaktadır. [5]

2.5. Implementation of the Naïve Bayes Algorithm in the SMS Spam Detection (2024)

Bu çalışmada, SMS tabanlı spam algılama problemi için **Naive Bayes sınıflandırıcısının gerçek bir SMS veri kümesi üzerinde uygulanması** ele alınmıştır. Labeled SMS verileri üzerinde ön işleme, TF-IDF çıkarımı ve model eğitimi gerçekleştirilmiş; Naive Bayes'in yüksek doğrulukta ve düşük hesaplama maliyetiyle spam tespit edebildiği gösterilmiştir. Çalışma, özellikle kaynak kısıtlı sistemlerde Naive Bayes'in avantajlarını öne çıkarmaktadır. [6]

2.6. SMS Spam Detection Using Multinomial Naive Bayes (2025)

AIP konferans serisinde yayınlanan bu çalışmada, SMS spam tespiti için **Multinomial Naive Bayes** algoritması kullanılmış ve performans, farklı makine öğrenmesi algoritmalarıyla karşılaştırılmıştır. Çalışmanın amacı, doğruluğu artırmak için TF-IDF tabanlı vektörleştirme ile Naive Bayes'i birleştirmektir. Sonuçlara göre, Multinomial NB modeli özellikle veri dengesizliği doğru ele alındığında yüksek doğruluk ve F1-skoru üretmiştir. [7]

2.7. Support Vector Machine Algorithm for SMS Spam Classification (2020)

Bu makalede, telekomünikasyon sektöründe SMS spam tespiti için **SVM algoritmasının** kullanımı ele alınmıştır. UCI SMS Spam veri kümesi üzerinde SVM, Naive Bayes ve K-NN gibi yöntemlerle karşılaştırılmış; SVM'in özellikle doğru hiperparametre ayarlarıyla diğer yöntemlerden daha yüksek doğruluk (yaklaşık %98'e varan) elde ettiği rapor edilmiştir. Çalışma, SVM'in maksimum marj prensibi sayesinde spam ve ham sınıfları arasında net karar sınırları oluşturabildiğini vurgulamaktadır. [8]

2.8. LSTM-Powered Spam Detection: A Deep Learning Approach for Sequential Text Classification (2025)

Bu çalışmada, **LSTM tabanlı bir derin öğrenme mimarisi** kullanılarak SMS spam tespiti yapılmıştır. Kamuya açık bir SMS spam veri kümesi üzerinde, durak kelime (stopword) temizleme, stemming ve lemmatization gibi kapsamlı ön işleme adımlarının ardından, metinler LSTM modeline girdi olarak verilmiştir. Çalışma, LSTM tabanlı modelin sekans bağımlılıklarını öğrenerek klasik makine öğrenmesi yöntemlerine göre daha yüksek doğruluk ve daha iyi recall değerleri elde edebildiğini göstermektedir. [9]

2.9. Altunay et al. (2024) – SMS Spam Detection System Based on Deep Learning (GRU+CNN)

Altunay ve arkadaşları, SMS spam tespiti için **GRU ve CNN tabanlı hibrit bir derin öğrenme modeli** önermektedir. Çalışmada, GRU katmanları ile sekans bilgisinin, CNN katmanları ile de yerel n-gram özelliklerinin yakalandığı bir mimari kullanılmıştır. Sınıf dengesizliği sorununa karşı random oversampling ve class-weight stratejileri uygulanmış; modelin yüksek doğruluk ve AUC değerlerine ulaştığı rapor edilmiştir. Ancak derin mimarilerin bellek ve işlem maliyeti açısından sınırlılıkları da tartışılmıştır. [10]

2.10. SMS Spam Detection Using LSTM – Analytics Vidhya (2021)

Uygulamalı bir rehber niteliğindeki bu çalışmada, LSTM tabanlı bir model kullanılarak SMS spam tespiti gerçekleştirilmiştir. SMSSpamCollection veri kümesi üzerinde Python ve Keras kullanılarak model eğitilmiş, embedding katmanı ile kelime temsilleri öğrenilmiş ve LSTM katmanları ile sekans bilgisi modellenmiştir. Çalışma, LSTM'in özellikle uzun metinler yerine kısa fakat yapısal SPAM kalıpları içeren SMS'lerde de başarılı olduğunu göstermektedir.[11]

3. DENEYSEL ÇALIŞMALAR

3.1. Veri Setinin Tanıtılmas

Bu çalışmada kısa mesajların (SMS) içeriklerine göre "spam" veya "ham" olarak sınıflandırılmasını amaçlayan bir makine öğrenmesi modeli geliştirilmiştir. Çalışmada kullanılan veri kümesi, binlerce kısa mesajdan oluşmakta olup her mesaj ilgili sınıf etiketiyle birlikte sunulmaktadır. Veri kümesinde yer alan mesajlar, gerçek kullanıcı iletişimlerinden derlenmiş ve sınıflandırma problemini yansıtan doğal bir dağılıma sahiptir.[1]

Veri kümesi incelendiğinde, ham (normal) mesajların veri kümesinde büyük çoğunluğu oluşturduğu, buna karşılık spam mesajlarının daha düşük oranda temsil edildiği görülmüştür. Bu durum, gerçek hayatta SMS trafiğinin doğal bir yansımasıdır; çünkü kullanıcıların aldığı mesajların önemli bir kısmı normal ileti içerirken, spam mesajları toplam içinde daha küçük bir orana sahiptir. Ancak bu dengesiz sınıf dağılımı, özellikle spam mesajlarının doğru tespit edilmesini zorlaştırabileceği için model eğitimi sırasında dikkate alınması gereken önemli bir noktadır.

Veri seti, makine öğrenmesi uygulamalarına uygun hâle getirilmeden önce belirli kontrollerden geçirilmiştir. Boş mesajlar, geçersiz kayıtlar ve etiket eksikliği olan örnekler veri kümesinden çıkarılmış, tüm etiketler "spam" ve "ham" olmak üzere iki sınıfta standardize edilmiştir. Ardından veri kümesi metin madenciliği ve doğal dil işleme adımlarına hazır hâle getirilmiştir.

Sonuç olarak, kullanılan veri kümesi; kısa mesaj içeriklerinin sınıflandırılmasına yönelik makine öğrenmesi modelleri için yeterli çeşitliliğe, doğal dağılıma ve temsil gücüne sahip olup, bu çalışmada gerçekleştirilecek olan ön işleme, özellik çıkarımı, model eğitimi ve performans değerlendirme süreçlerinin temelini oluşturmaktadır.

3.2. Veri Ön İşleme Adımları

Makine öğrenmesi modellerinin başarılı bir şekilde eğitilebilmesi için kullanılan veri kümesinin temiz, tutarlı ve güvenilir olması gerekmektedir. Bu nedenle SMS mesajları üzerinde kapsamlı bir ön işleme süreci uygulanmıştır. Bu bölümde, verilerin modele uygun hâle getirilmesi amacıyla gerçekleştirilen tüm adımlar ayrıntılı şekilde açıklanmaktadır.

3.2.1. Veri Temizliği (Null ve Yinelenen Kayıtların Kaldırılması)

Öncelikle veri kümesi incelenmiş ve eksik (null) değer içeren kayıtlar tespit edilmiştir. Bu kayıtlar anlamlı bir şekilde işlenemeyeceğinden veri kümesinden çıkarılmıştır. Ardından yinelenen (duplicate) mesajlar belirlenmiş ve veri kümesinden kaldırılmıştır. Yinelenen örneklerin mevcut olması, modelin belirli örnekleri aşırı öğrenmesine (overfitting) yol açabileceğinden bu adım model performansını artırmak için önemlidir.

Bu adımın sonunda veri kümesi benzersiz, eksiksiz ve daha güvenilir bir yapıya kavuşturulmuştur.

3.2.2. Küçük Harfe Dönüştürme (Lowercasing)

Tüm mesaj içerikleri küçük harfe dönüştürülmüştür. Böylece farklı biçimlerde yazılan kelimelerin (örneğin "FREE", "Free", "free") tek bir kelime olarak değerlendirilmesi sağlanmış ve özellik uzayının gereksiz şekilde büyümesi engellenmiştir.

3.2.3. Tokenization (Kelime Ayırma İşlemi)

Mesajlar kelime bazlı parçalara ayrılmıştır. Tokenization işlemi, metnin yapısal olarak analiz edilebilmesi ve sonraki işlemlerde kelime seviyesinde dönüşümler yapılabilmesi için temel bir adımdır.

3.2.4. Noktalama İşaretlerinin ve Özel Karakterlerin Kaldırılması

SMS'lerde sıkça görülen noktalama işaretleri, semboller, emoji benzeri ifadeler ve model açısından anlam taşımayan karakterler temizlenmiştir. Bu işlem, veri kümesindeki gürültüyü azaltarak daha anlamlı özellik çıkarımına imkân sağlamıştır.

3.2.5. Sayı, URL ve Gereksiz Karakterlerin Silinmesi

Spam mesajlarının çoğunda bulunan linkler, telefon numaraları, kampanya kodları ve rastgele karakter dizileri temizlenmiştir. Bu ifadeler genellikle sınıflandırma görevine katkı sağlamamakta ve kelime uzayını gereksiz yere büyütmektedir.

3.2.6. Stopword Temizliği

İngilizcede sık kullanılan ancak sınıflandırmaya anlamlı katkı sağlamayan "the", "is", "in", "at", "a" gibi stopword kelimeler mesajlardan çıkarılmıştır. Bu işlem, modelin yalnızca anlamlı ve ayrıştırıcı kelimelere odaklanmasını sağlamış ve performans metriklerine olumlu katkı yapmıştır.

3.2.7. Gereksiz Boşlukların Düzenlenmesi

Metin işleme adımları sonrasında oluşabilecek fazla boşluklar temizlenmiş, tüm mesajlar düzenli ve tutarlı bir formata getirilmiştir.

3.2.8. TF-IDF ile Sayısal Özellik Çıkarımı

Ön işleme işlemlerinin ardından tüm mesajlar TF-IDF (Term Frequency–Inverse Document Frequency) yöntemi kullanılarak sayısal özellik vektörlerine dönüştürülmüştür. TF-IDF, her kelimenin mesaj içindeki ve tüm veri kümesi içindeki önemini hesaplayan güçlü bir metin temsil yöntemidir.

Bu işlem sonucunda her SMS mesajı, yaklaşık **3000 boyutlu bir vektör** ile temsil edilmiştir. Bu boyut, veri kümesinde yeterli sıklığa ve öneme sahip kelimelerden oluşmakta olup, sınıflandırma modellerinin öğrenme süreci için gerekli temel girdiyi oluşturmaktadır.

3.2.9. Eğitim ve Test Setlerinin Hazırlanması

Ön işleme tamamlandıktan sonra veriler eğitim (train) ve test (test) olmak üzere ikiye ayrılmıştır. Ek olarak, modellerin daha sağlıklı bir şekilde değerlendirilmesi için **k-kat çapraz doğrulama (k-fold cross validation)** yöntemi uygulanmıştır. Bu yöntemle model kararlılığı artırılmıştır.

3.2.10 Sonuç

Bu kapsamlı ön işleme süreci sonucunda ham SMS verileri:

- Temizlenmiş,
- Tutarlı hâle getirilmiş,
- Gürültüden arındırılmış,
- Sayısal forma dönüştürülmüş,
- Model eğitimine uygun hâle getirilmiştir.

Bu adımlar, çalışmada kullanılan sınıflandırma modellerinin başarıyla eğitilmesini ve güvenilir sonuçlar üretmesini sağlamıştır.

3.3. Özellik Çıkarımı (TF-IDF)

Metin tabanlı makine öğrenmesi uygulamalarında, ham metnin doğrudan sınıflandırma algoritmalarına verilmesi mümkün değildir. Bu nedenle SMS mesajlarının sayısal bir yapıya dönüştürülmesi gerekmektedir. Bu çalışmada, kısa metinlerin anlamlı bir şekilde temsil edilmesi amacıyla **TF-IDF (Term Frequency – Inverse Document Frequency)** yöntemi kullanılmıştır. TF-IDF, hem kelimenin mesaj içindeki önemini hem de tüm veri kümesi içerisindeki ayırt ediciliğini dikkate alan bir özellik çıkarım yöntemidir.

3.3.1. Term Frequency (TF) – Terim Frekansı

TF, bir kelimenin ilgili mesaj içinde kaç kez geçtiğini gösterir. Bir kelimenin bir mesajda sık geçmesi, o kelimenin o mesaja özgü olabileceğini ve mesajın anlamına katkı sağladığını ifade eder.

3.3.2. Inverse Document Frequency (IDF) – Ters Belge Frekansı

IDF, bir kelimenin tüm veri kümesinde ne kadar yaygın olduğunu ölçer. Eğer bir kelime çok fazla mesajda geçiyorsa bu kelimenin ayırt edici gücü düşüktür. Buna karşılık yalnızca belirli mesajlarda görünen kelimeler sınıflandırma için daha anlamlıdır.

IDF değeri şu şekilde hesaplanır:

$$IDF = \log\left(\frac{N}{df}\right)$$

- **N:** toplam mesaj sayısı
- **df:** kelimenin geçtiği mesaj sayısı

3.3.3. TF-IDF Hesaplaması

Her kelime için TF ve IDF değerleri çarpılarak kelimenin ağırlığı elde edilir:

$$TF\text{-}IDF = TF \times IDF$$

Bu işlem sonucunda, veri kümesindeki her mesaj sabit boyutlu bir vektör hâline gelir.

3.4. Kullanılan Modeller

Bu çalışmada SMS mesajlarının spam veya ham olarak sınıflandırılması amacıyla farklı makine öğrenmesi algoritmaları değerlendirilmiştir. Çalışma kapsamında toplam beş temel sınıflandırıcı, iki topluluk öğrenmesi yöntemi ve bir yapay sinir ağı (ANN) modeli kullanılmıştır. Modellerin farklı yapıları, avantajları ve sınıflandırma performansları bu bölümde ayrıntılı şekilde açıklanmaktadır.

3.4.1. Multinomial Naive Bayes (MNB)

Multinomial Naive Bayes, metin madenciliği alanında sıkça kullanılan ve özellikle TF-IDF gibi kelime frekansı temelli özelliklerle oldukça uyumlu çalışan bir sınıflandırıcıdır. Model, her kelimenin sınıflar üzerindeki koşullu olasılığını hesaplayarak tahmin yapar. Basit yapısı, hızlı eğitimi ve düşük hesaplama maliyeti sayesinde SMS spam tespiti çalışmalarında genellikle başlangıç modeli (baseline) olarak tercih edilmektedir.

3.4.2. Support Vector Machine (SVM)

Destek Vektör Makineleri, yüksek boyutlu veri kümelerinde (örneğin, TF-IDF ile oluşturulan binlerce özellikli metinlerde) yüksek başarı gösteren güçlü bir sınıflandırıcıdır. SVM, spam ve ham mesajları ayıran en uygun hiper-düzlemi bulmayı amaçlar. Bu model, özellikle doğruluk ve genelleme yeteneği açısından literatürde en başarılı yöntemlerden biri olarak kabul edilmektedir.

3.4.3. Logistic Regression (LR)

Logistic Regression, metin sınıflandırmada yaygın olarak kullanılan lineer bir modeldir. Model, bir mesajın spam olma olasılığını hesaplayarak ikili sınıflandırma yapar. Hem eğitim sürecinin hızlı olması hem de predict_proba çıktısı üretmesi nedeniyle özellikle kullanıcı arayüzlerinde tercih edilen bir sınıflandırıcıdır.

3.4.4. K-Nearest Neighbors (KNN)

KNN sınıflandırıcısı, örnekler arasındaki benzerliği ölçerek en yakın komşulara göre karar veren bir algoritmadır. Metin verisi yüksek boyutlu olduğu için KNN her zaman en yüksek performansı vermese de, farklı bir yaklaşım sunması açısından çalışmada değerlendirilmiştir. KNN, veri dağılımını varsaymadığı için esnek bir yapıya sahiptir.

3.4.5. Decision Tree (DT)

Karar Ağaçları, veriyi dallara ayırarak hiyerarşik bir karar yapısı oluşturur. Bu yöntem, yorumlanabilirliği yüksek olan bir modeldir. Metin verisi gibi yüksek boyutlu veri kümelerinde her zaman en iyi performansı vermese de, model çeşitliliğini artırmak ve karşılaştırma yapmak amacıyla kullanılmıştır.

3.4.6. Topluluk Öğrenmesi Modelleri (Ensemble Methods)

Topluluk yöntemleri, birden fazla zayıf sınıflandırıcının bir araya gelerek daha güçlü bir tahmin performansı üretmesini amaçlar. Bu çalışmada iki farklı ensemble yöntemi kullanılmıştır:

3.4.6.1. Bagging Classifier (Bootstrap Aggregating)

Bagging yöntemi, aynı sınıflandırıcının farklı bootstrap örnekleri üzerinde eğitilmesine dayanır. Her model bağımsız olarak eğitilir ve nihai tahmin tüm modellerin oy çokluğuna göre belirlenir.
Bagging, özellikle yüksek varyans üreten karar ağaçlarında performansı artırmak için etkili bir yöntemdir. Metin sınıflandırma problemlerinde TF-IDF ile oluşturulan geniş özellik uzaylarında kararlı sonuçlar sağlaması nedeniyle bu çalışmada değerlendirilmiştir.

3.4.6.2. AdaBoost (Adaptive Boosting)

AdaBoost, hatalı sınıflandırılan örneklere ağırlık vererek sonraki zayıf öğrenicilerin bu hatalara odaklanmasını sağlar. Bu yaklaşım sayesinde daha hassas bir karar sınırı oluşturulabilir. AdaBoost, özellikle dengesiz veri kümelerinde önemli performans artışları sağlayabilmektedir.

3.4.7. Yapay Sinir Ağı (Artificial Neural Network – ANN)

Bu çalışmada bir yapay sinir ağı modeli de uygulanmıştır. ANN modeli giriş katmanı, bir veya birden fazla gizli katman ve çıktı katmanından oluşmaktadır. Eğitim süreci **120 epoch** olacak şekilde yürütülmüş, her epoch sonunda doğruluk ve kayıp değerleri izlenmiştir.

Sinir ağları, kelimeler arasındaki ilişkiyi daha derin düzeyde öğrenebilme kapasitesine sahiptir ve non-lineer karar sınırları oluşturabilir. Bu nedenle ANN modeli, klasik makine öğrenmesi yöntemlerine göre farklı bir perspektif sunması için çalışmaya dahil edilmiştir.

3.5. Eğitim Süreci

Bu çalışmada kullanılan makine öğrenmesi modellerinin performansını değerlendirmek için iki farklı eğitim ve doğrulama yöntemi uygulanmıştır: hold-out (dışarıda tutma) ve k-kat çapraz doğrulama (k-fold cross validation). Bu yöntemlerin birlikte kullanılması, modellerin hem genel performansını hem de kararlılığını daha sağlıklı bir şekilde ölçmeyi sağlamıştır.

3.5.1. Dışarda Tutma Yöntemi (Hold-Out)

Hold-out yöntemi, veri kümesinin eğitim ve test olmak üzere ikiye ayrılması esasına dayanır. Bu çalışmada veri kümesi:

- **%80 eğitim**,
- **%20 test**

Olmak üzere iki bölüme ayrılmıştır.

Eğitim veri kümesi modellerin öğrenmesi için kullanılırken, test veri kümesi modellerin daha önce görmediği yeni örnekler üzerinde performanslarını değerlendirmek için kullanılmıştır. Bu yöntem, modellerin gerçek dünya verileri üzerindeki davranışını gözlemlemek için temel bir yaklaşım sunmaktadır.

Hold-out yöntemiyle tüm modeller eğitilmiş ve her bir model için:

- Doğruluk (Accuracy)
- Duyarlılık (Recall)
- Özgüllük (Specificity)
- F1-skoru
- AUC
- Karışıklık Matrisi (Confusion Matrix)
- ROC Eğris hesaplanmıştır.

Bu sonuçlar ilgili alt başlıklarda ayrıntılı olarak sunulmuştur.

3.5.2. k-Kat Çapraz Doğrulama (Cross Validation)

Tek bir eğitim-test ayrımı, model performansını veri kümesinin belirli bir dağılımına göre ölçtüğü için her zaman yeterli olmayabilir. Bu nedenle çalışmada ayrıca **k = 5** olacak şekilde **5-kat çapraz doğrulama** uygulanmıştır.

Çapraz doğrulama sürecinde:

- Veri kümesi 5 eşit parçaya bölünür.
- Her iterasyonda bir parça test için ayrılır, diğer 4 parça eğitim için kullanılır.

- 5 iterasyonun sonunda tüm sonuçların ortalaması alınır.

Bu yöntem:

- Modelin kararlılığını,
- Farklı veri bölünmelerindeki tutarlılığını,
- Overfitting riskini

değerlendirmek için önemli bir avantaj sağlamaktadır.

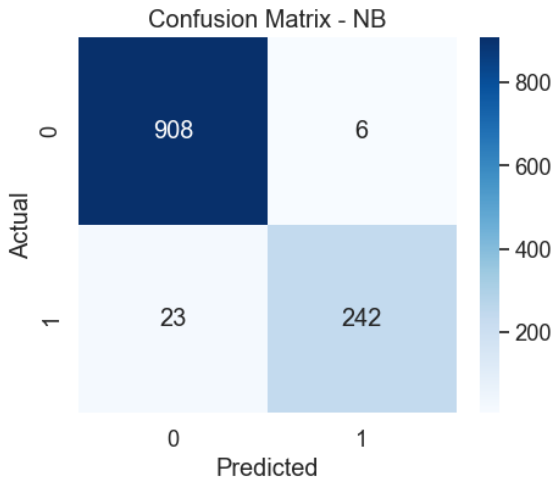
Çapraz doğrulama sonuçları daha sonra **performans karşılaştırma tabloları** içerisinde sunulmuştur.

3.5.3. Hold-Out Sonuçları

Bu bölümde, %80 eğitim ve %20 test ayrımı ile gerçekleştirilen hold-out yöntemi sonucunda elde edilen performans değerleri sunulmaktadır. Her model için karışıklık matrisi, ROC eğrisi ve sınıflandırma metrikleri ayrı ayrı değerlendirilmiştir.

3.5.3.1. Multinomial Naive Bayes – Hold-Out Sonuçları

a) Karışıklık Matrisi

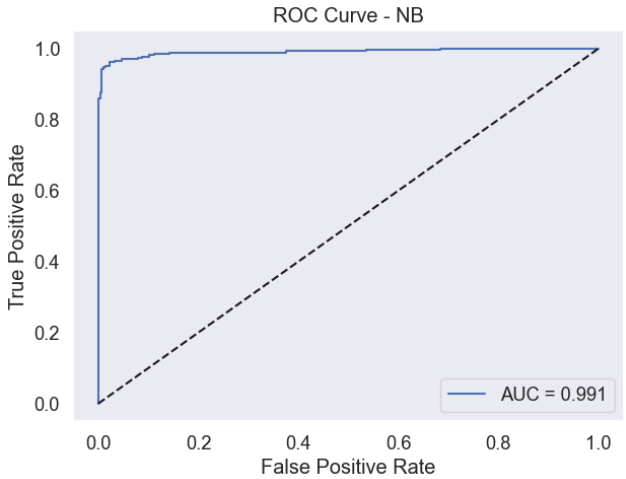


Şekil 1.Multinomial Naive Bayes-Hold Out eğitim Karışıklık Matrisi

b) Başarı Metrikler

Accuracy	Precision	Recall (Duyarlılık)	Specificity (Özgüllük)	F1-Score
0.97	0.97	0.91	0.99	0.94

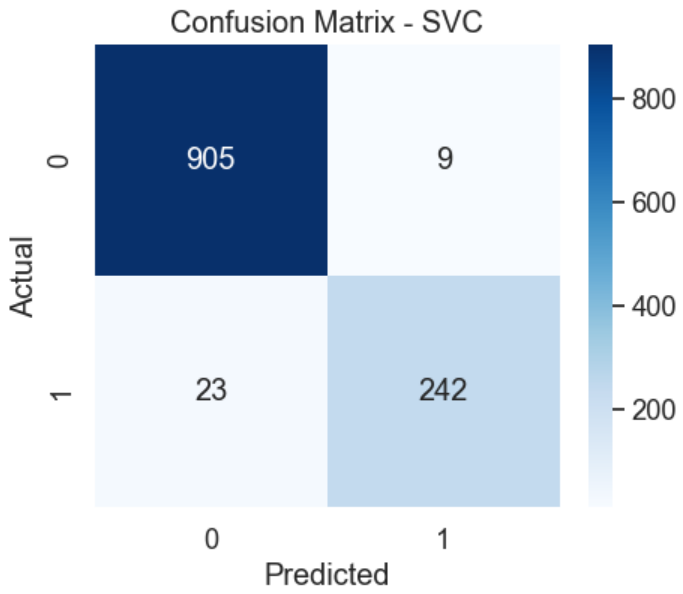
c) ROC Eğrisi



Şekil 2.Multinomial Naive Bayes -Hold Out eğitim Roc eğrisi

3.5.3.2. Support Vector Machine – Hold-Out Sonuçları

a) Karışıklık Matrisi

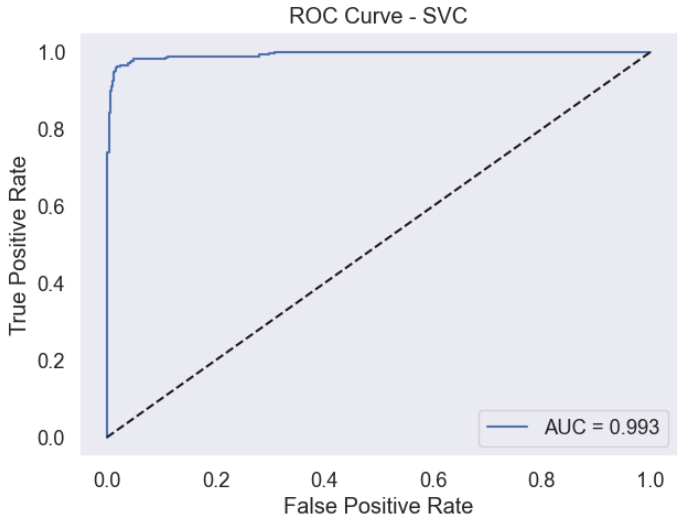


Şekil 3.SVM-Hold Out eğitim Karışıklık Matrisi

b) Başarı Metrikler

Accuracy	Precision	Recall (Duyarlılık)	Specificity (Özgüllük)	F1-Score
0.97	0.96	0.91	0.99	0.93

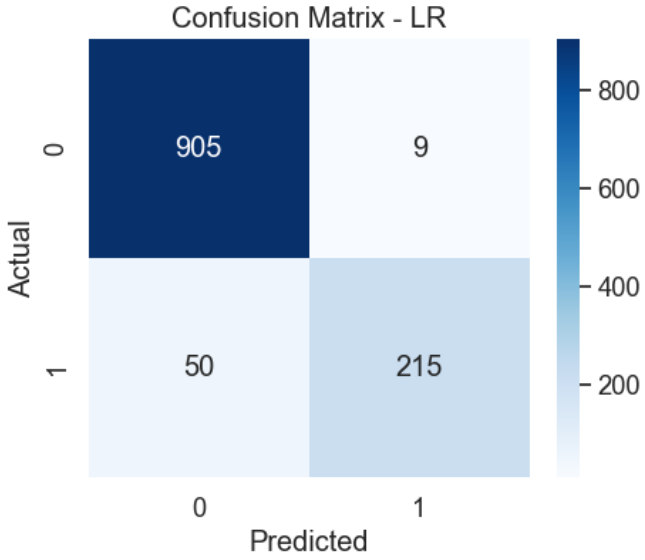
c) ROC Eğrisi



Şekil 4.SVC -Hold Out eğitim Roc eğrisi

3.5.3.3. Logistic Regression – Hold-Out Sonuçları

a) Karışıklık Matrisi

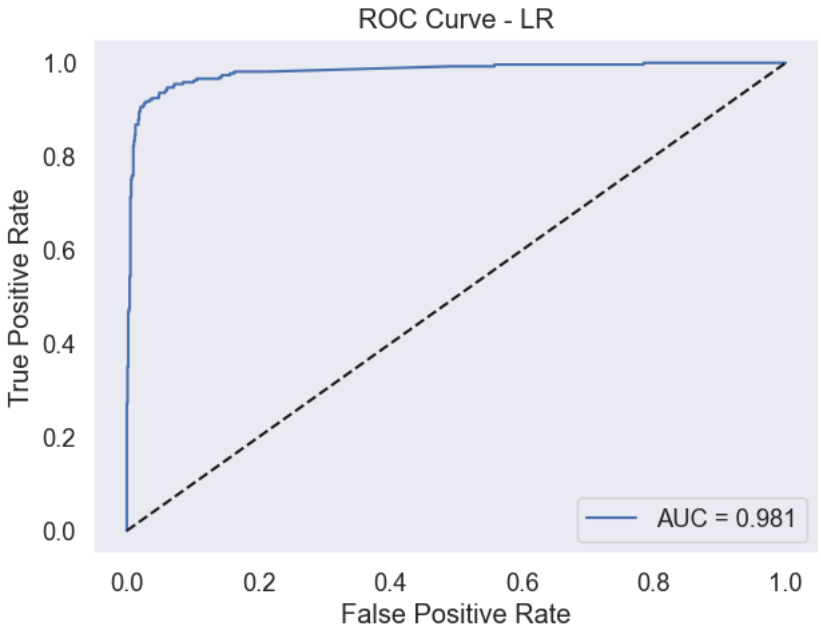


Şekil 5.Logisitic Regression-Hold Out eğitim Karışıklık Matrisi

b) Başarı Metrikler

Accuracy		Precision	Recall (Duyarlılık)	Specificity (Özgüllük)	F1-Score
0.94		0.95	0.81	0.99	0.87

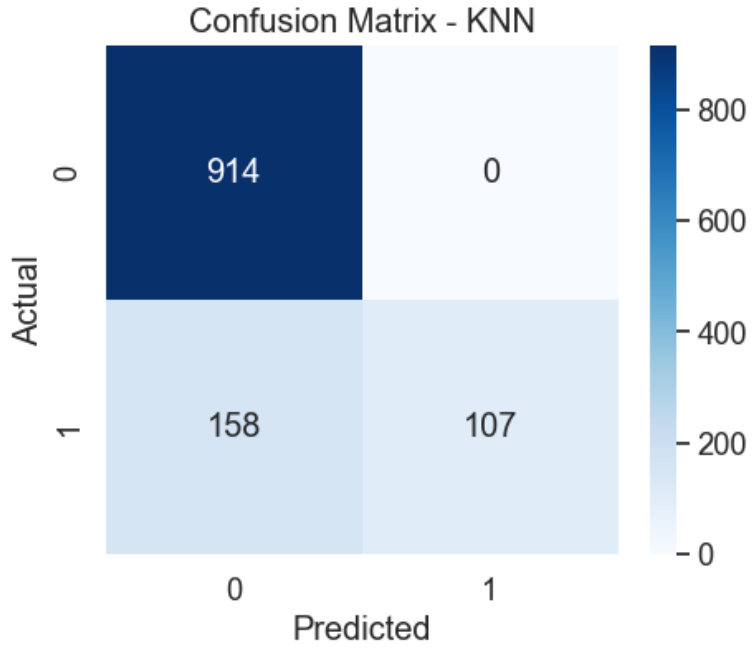
c) ROC Eğrisi



Şekil 6.Logisitic Regression-Hold Out eğitim Roc eğrisi

3.5.3.4. KNN – Hold-Out Sonuçları

a) Karışıklık Matrisi

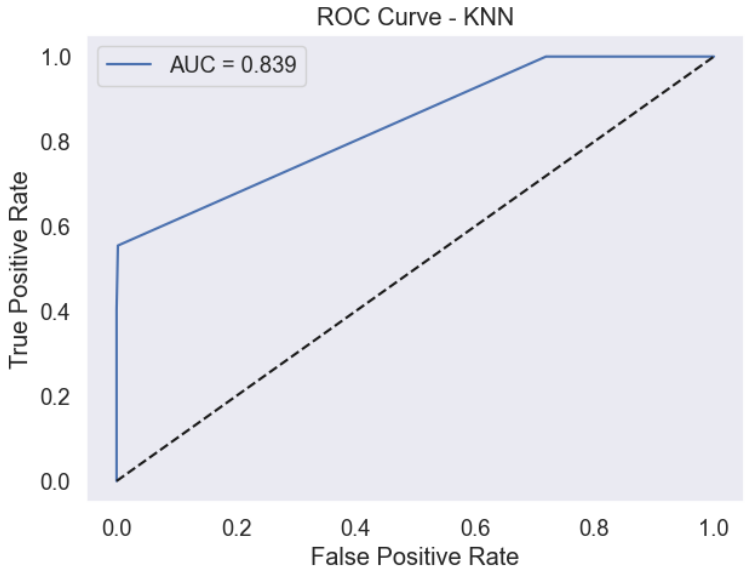


Şekil 7.KNN-Hold Out eğitim Karışıklık Matrisi

b) Başarı Metrikler

Accuracy	Precision	Recall (Duyarlılık)	Specificity (Özgüllük)	F1-Score
0.86	1	0.4	1	0.5

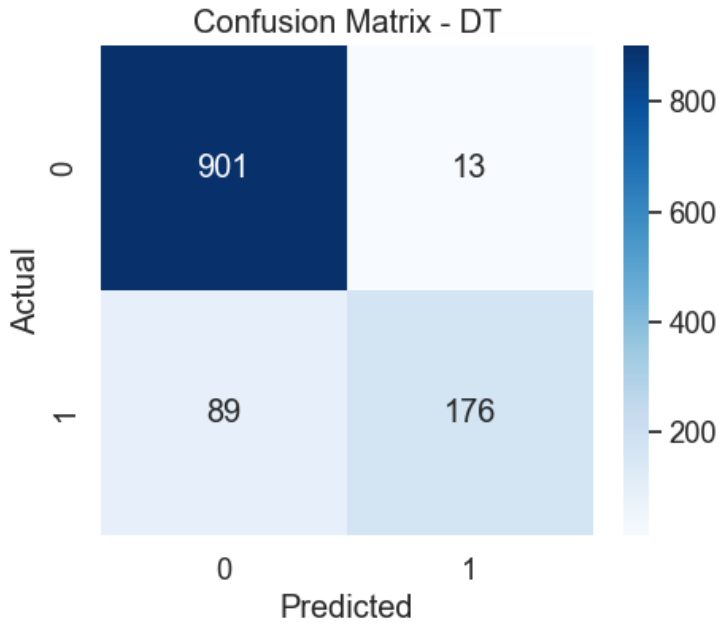
c) ROC Eğrisi



Şekil 8.KNN-Hold Out eğitim Roc eğrisi

3.5.3.5. Decision Tree– Hold-Out Sonuçları

a) Karışıklık Matrisi

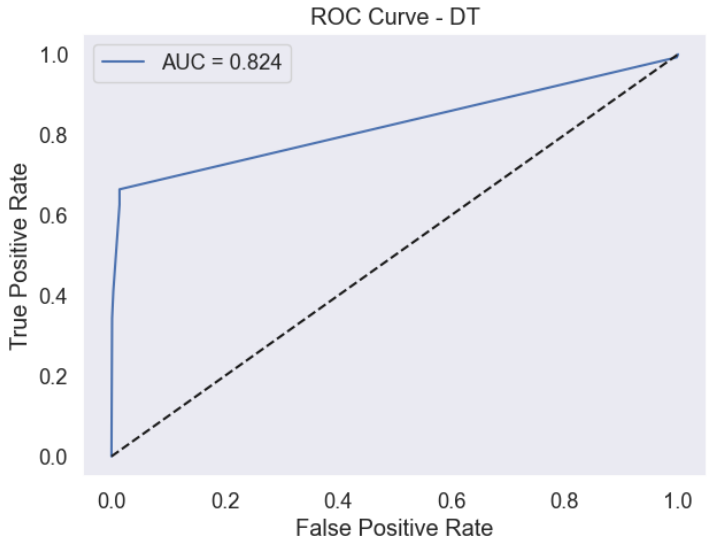


Şekil 9. Decision Tree -Hold Out eğitim Karışıklık Matrisi

b) Başarı Metrikler

Accuracy	Precision	Recall (Duyarlılık)	Specificity (Özgüllük)	F1-Score
0.91	0.93	0.66	0.98	0.77

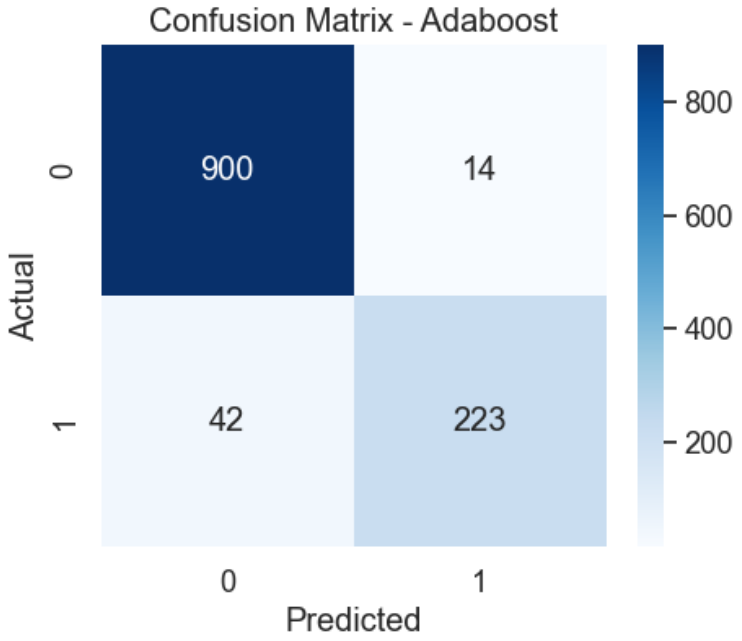
c) ROC Eğrisi



Şekil 10.Decision Tre-Hold Out eğitim Roc eğrisi

3.5.3.6. AdaBoost – Hold-Out Sonuçları

a) Karışıklık Matrisi

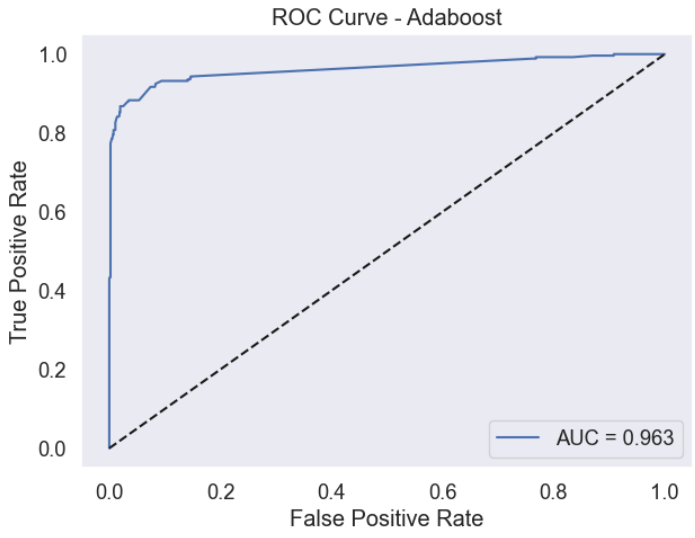


Şekil 11. AdaBoost -Hold Out eğitim Karışıklık Matrisi

b) Başarı Metrikler

Accuracy	Precision	Recall (Duyarlılık)	Specificity (Özgüllük)	F1-Score
0.95	0.94	0.84	0.98	0.88

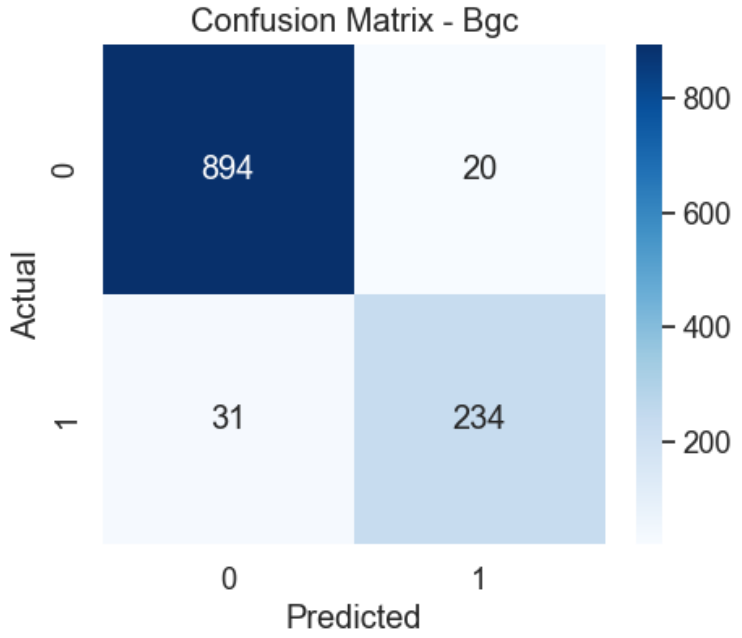
c) ROC Eğrisi



Şekil 12.Logisitic Regression-Hold Out eğitim Roc eğris

3.5.3.7. Bagging Classifier – Hold-Out Sonuçları

a) Karışıklık Matrisi

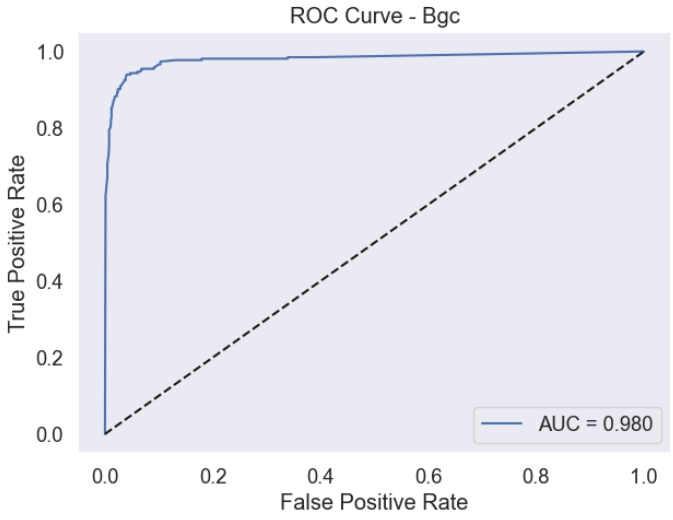


Şekil 13. Bagging Classifier -Hold Out eğitim Karışıklık Matrisi

b) Başarı Metrikler

Accuracy	Precision	Recall (Duyarlılık)	Specificity (Özgüllük)	F1-Score
0.95	0.92	0.88	0.99	0.97

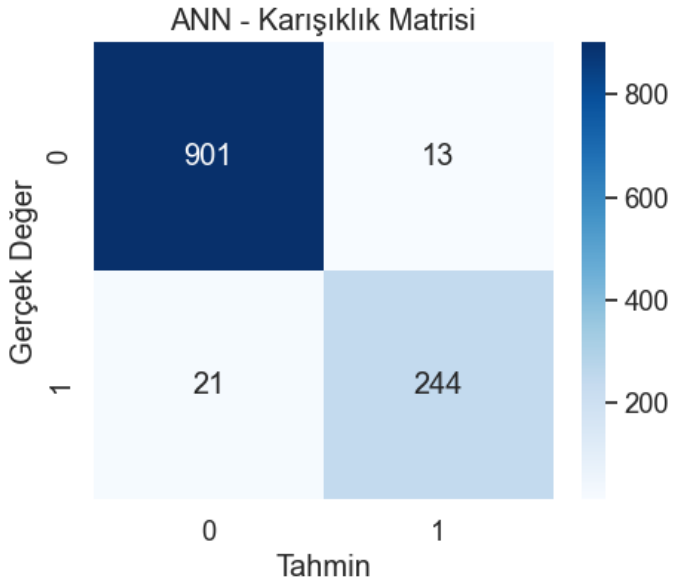
c) ROC Eğrisi



Şekil 14. Bagging Classifier -Hold Out eğitim Roc eğris

3.5.3.8. ANN (Yapay Sinir Ağı) – Hold-Out Sonuçları

a) Karışıklık Matrisi

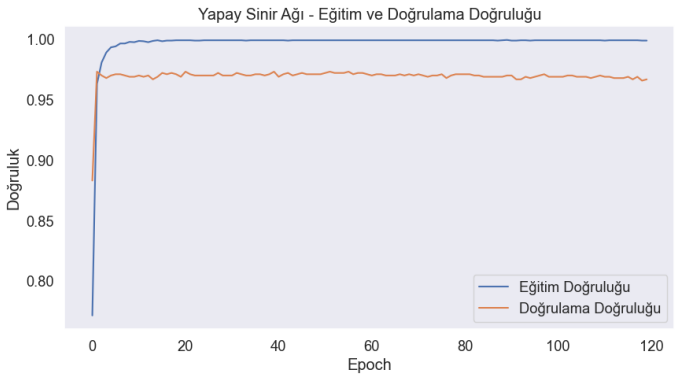


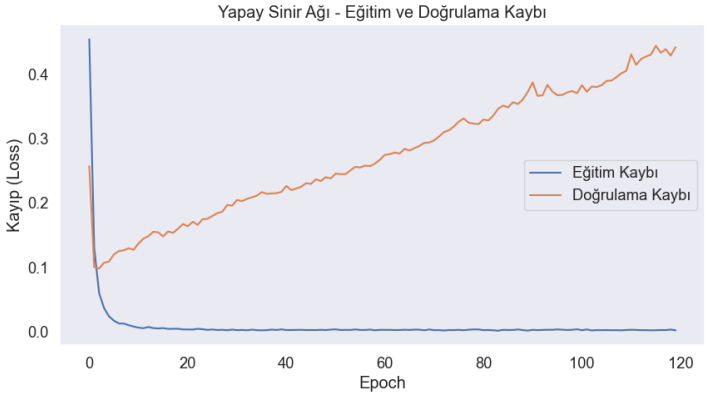
Şekil 15. ANN -Hold Out eğitim Karışıklık Matrisi

b) Başarı Metrikler

Accuracy	Precision	Recall (Duyarlılık)	Specificity (Özgüllük)	F1-Score
0.97	0.94	0.92	0.98	0.93

c) Eğitim ve Doğruluğu - Kayıp Grafikleri





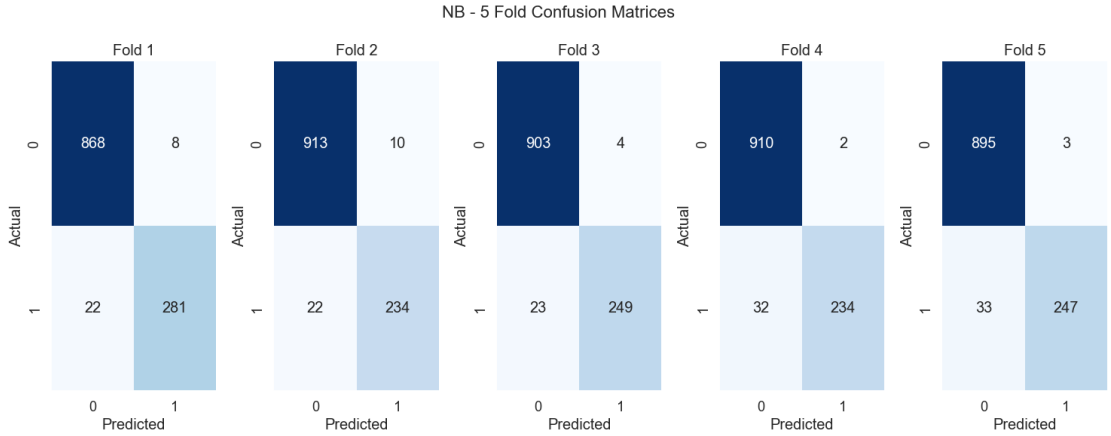
Şekil 16.ANN-Hold Out eğitim ACC-LOSS Grafiği

3.5.4. Cross Validation Sonuçları

Bu bölümde, modellerin k = 5 katlı çapraz doğrulama yöntemiyle değerlendirilmesinden elde edilen sonuçlar sunulmaktadır. Çapraz doğrulama, her modelin farklı veri bölünmelerindeki kararlılığını ölçmek için önemlidir.

3.5.4.1. Multinomial Naive Bayes – Cross Validation Sonuçlar

a) Karışıklık Matrisler Her fold için

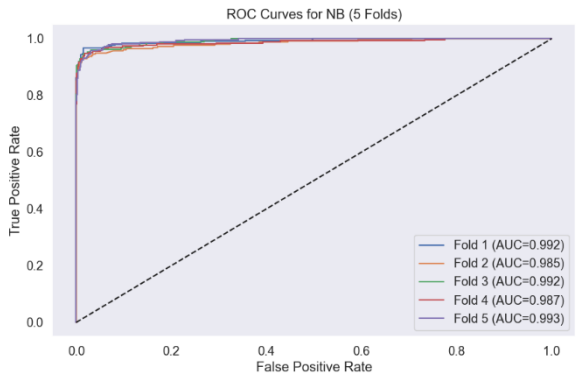


Şekil 17. Multinomial Naive Bayes – KFold eğitim Karışıklık Matrisler

b) Başarı Metrikler

Fold	Accuracy	Precision	Recall (Duyarlılık)	Specificity (Özgüllük)	F1-Score
1	0.97	0.97	0.93	0.99	0.95
2	0.97	0.96	0.91	0.99	0.94
3	0.98	0.98	0.92	1	0.95
4	0.97	0.99	0.88	1	0.93
5	0.97	0.99	0.88	1	0.93

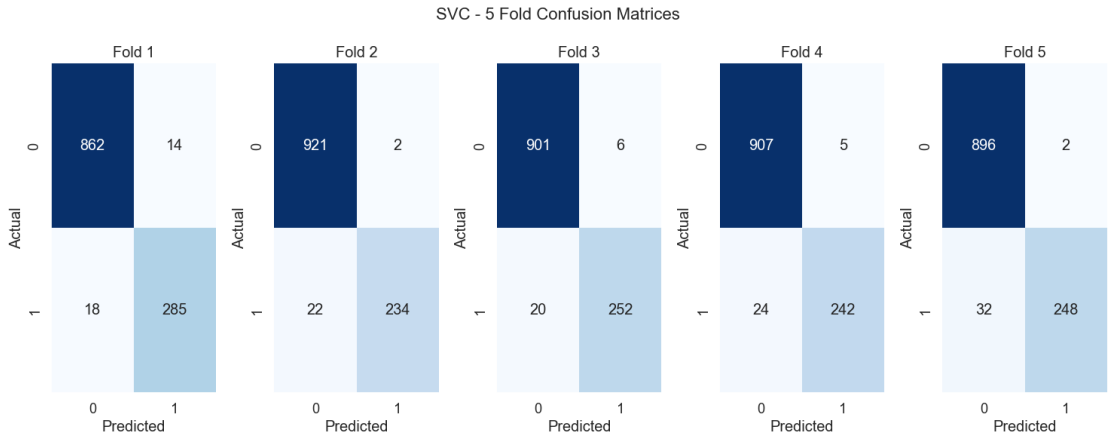
c) ROC Eğriler



Şekil 18. Multinomial Naive Bayes – KFold eğitim Roc Eğriler

3.5.4.2. SVM– Cross Validation Sonuçlar

a) Karışıklık Matrisler Her fold için

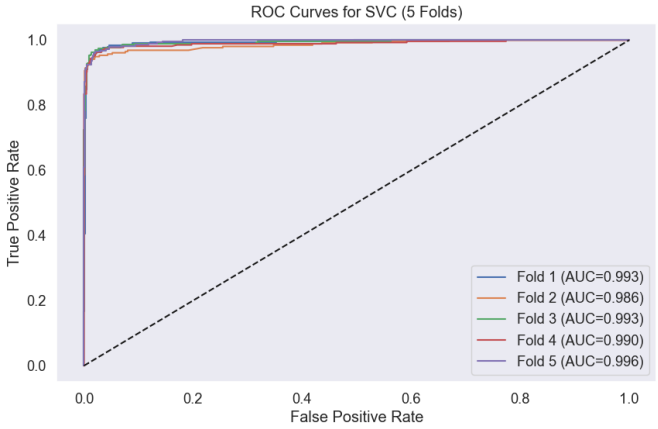


Şekil 19. SVM– KFold eğitim Karışıklık Matrisler

b) Başarı Metrikler

Fold	Accuracy	Precision	Recall (Duyarlılık)	Specificity (Özgüllük)	F1-Score
1	0.97	0.95	0.94	0.98	0.95
2	0.98	0.99	0.91	1	0.95
3	0.98	0.98	0.93	0.99	0.95
4	0.98	0.99	0.91	0.99	0.94
5	0.97	0.99	0.89	1	0.94

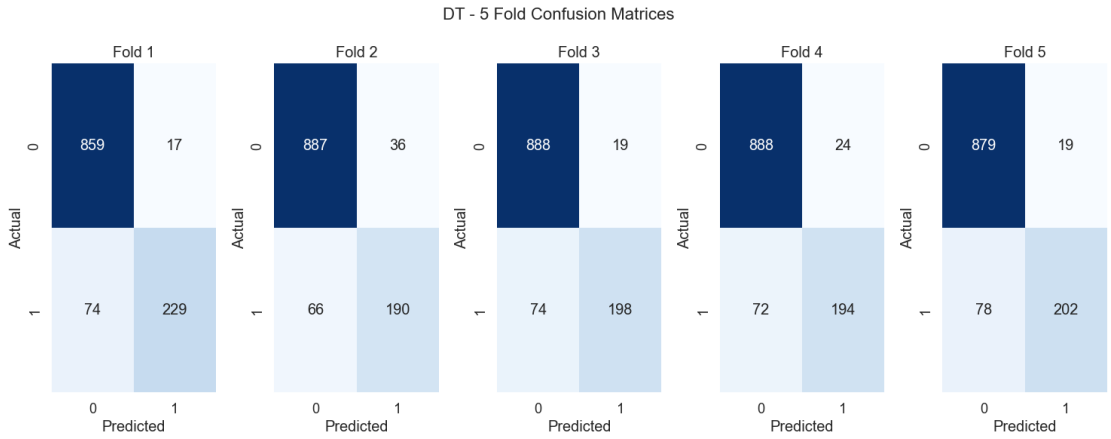
c) ROC Eğriler



Şekil 20. SVM – KFold eğitim Roc Eğriler

3.5.4.3. Decision Tree– Cross Validation Sonuçlar

a) Karışıklık Matrisler Her fold için

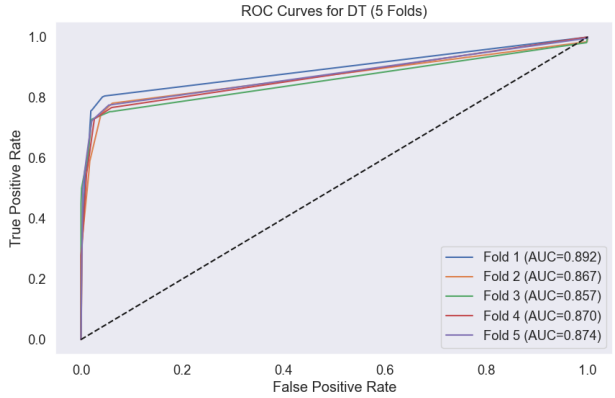


Şekil 21.Decision Tree – KFold eğitim Karışıklık Matrisler

b) Başarı Metrikler

Fold	Accuracy	Precision	Recall (Duyarlılık)	Specificity (Özgüllük)	F1-Score
1	0.92	0.93	0.75	0.98	0.83
2	0.91	0.84	0.74	0.96	0.78
3	0.92	0.91	0.72	0.97	0.81
4	0.91	0.88	0.72	0.97	0.80
5	0.91	0.91	0.72	0.97	0.80

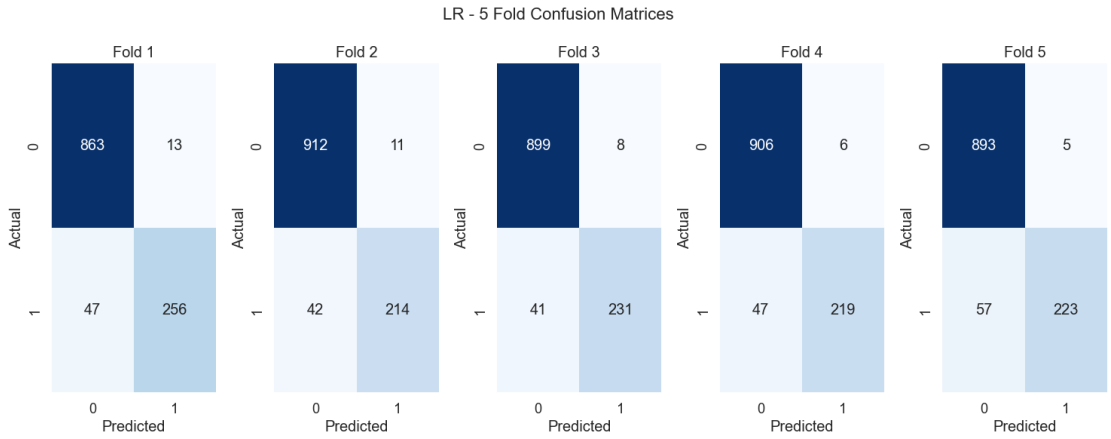
c) ROC Eğriler



Şekil 22. Decision Tree – KFold eğitim Roc Eğriler

3.5.4.4. Logistic Regression – Cross Validation Sonuçlar

a) Karışıklık Matrisler Her fold için

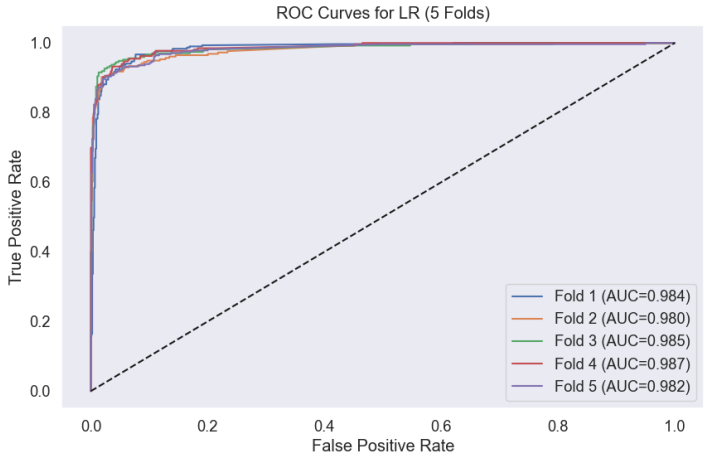


Şekil 23. Logistic Regression – KFold eğitim Karışıklık Matrisler

b) Başarı Metrikler

Fold	Accuracy	Precision	Recall (Duyarlılık)	Specificity (Özgüllük)	F1-Score
1	0.94	0.95	0.84	0.98	0.89
2	0.95	0.95	0.83	0.98	0.89
3	0.95	0.96	0.84	0.99	0.90
4	0.95	0.97	0.82	0.99	0.89
5	0.94	0.97	0.79	0.99	0.87

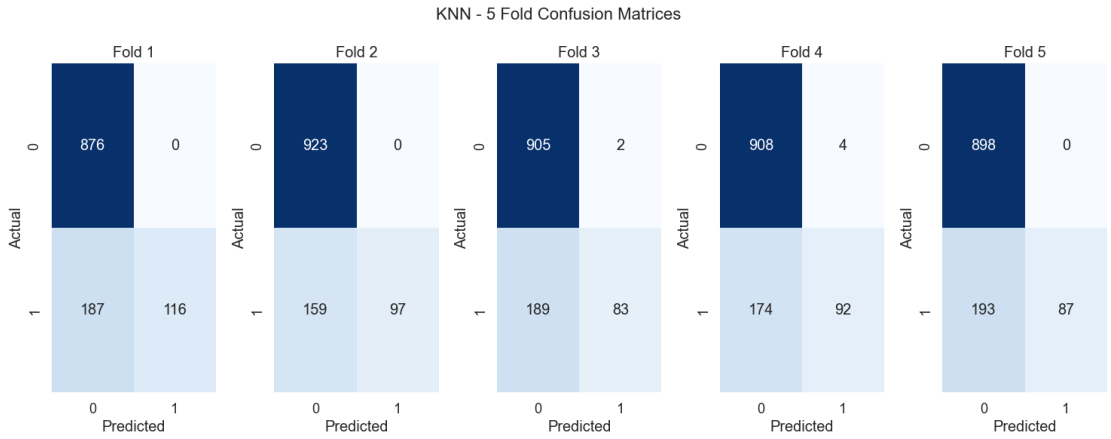
c) ROC Eğriler



Şekil 24. Logistic Regression – KFold eğitim Roc Eğriler

3.5.4.5. KNN– Cross Validation Sonuçlar

a) Karışıklık Matrisler Her fold için

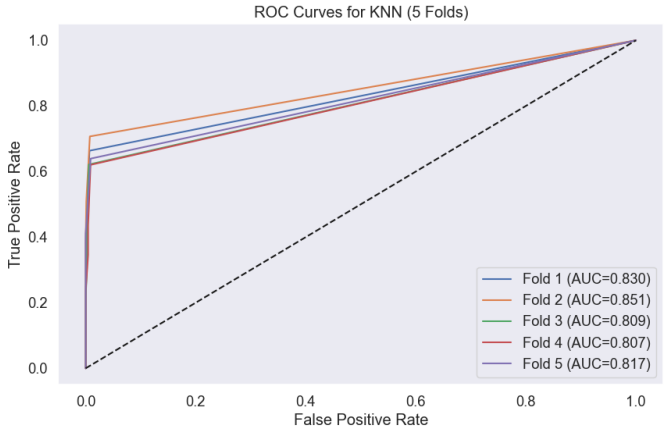


Şekil 25. KNN– KFold eğitim Karışıklık Matrisler

b) Başarı Metrikler

Fold	Accuracy	Precision	Recall (Duyarlılık)	Specificity (Özgüllük)	F1-Score
1	0.84	1	0.38	1	0.55
2	0.86	1	0.37	1	0.55
3	0.83	0.97	0.30	0.99	0.46
4	0.84	0.95	0.34	0.99	0.50
5	0.83	1	0.31	1	0.47

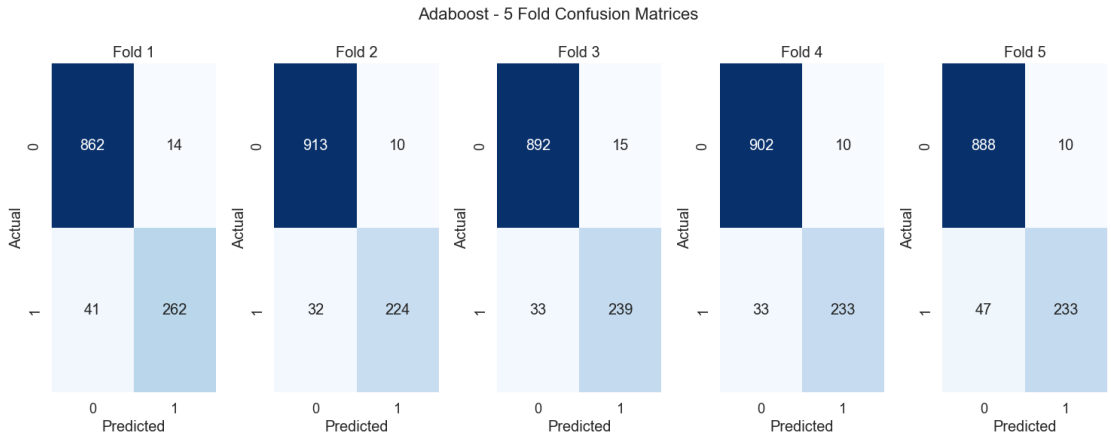
c) ROC Eğriler



Şekil 26. KNN – KFold eğitim Roc Eğriler

3.5.4.6. AdaBoost– Cross Validation Sonuçlar

a) Karışıklık Matrisler Her fold için

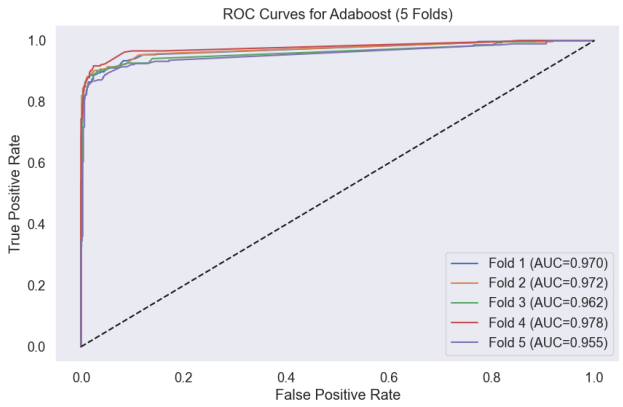


Şekil 27.AdaBoost – KFold eğitim Karışıklık Matrisler

b) Başarı Metrikler

Fold	Accuracy	Precision	Recall (Duyarlılık)	Specificity (Özgüllük)	F1-Score
1	0.95	0.94	0.86	0.98	0.90
2	0.96	0.95	0.87	0.98	0.91
3	0.96	0.94	0.87	0.98	0.90
4	0.96	0.95	0.87	0.98	0.91
5	0.95	0.95	0.83	0.98	0.89

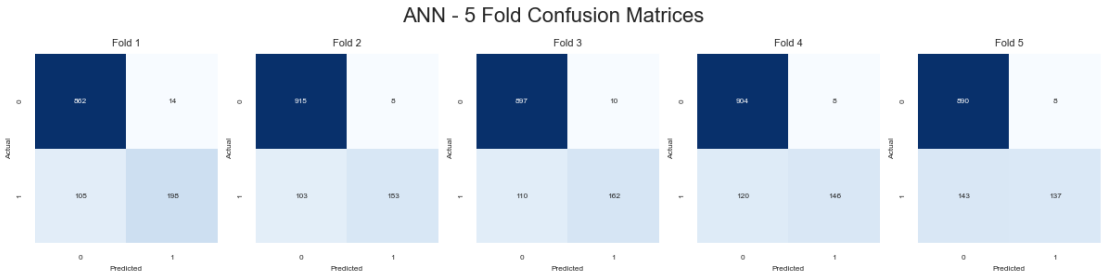
c) ROC Eğriler



Şekil 28. AdaBoost – KFold eğitim Roc Eğriler

3.5.4.7. Bagging Classifier – Cross Validation Sonuçlar

a) Karışıklık Matrisler Her fold için

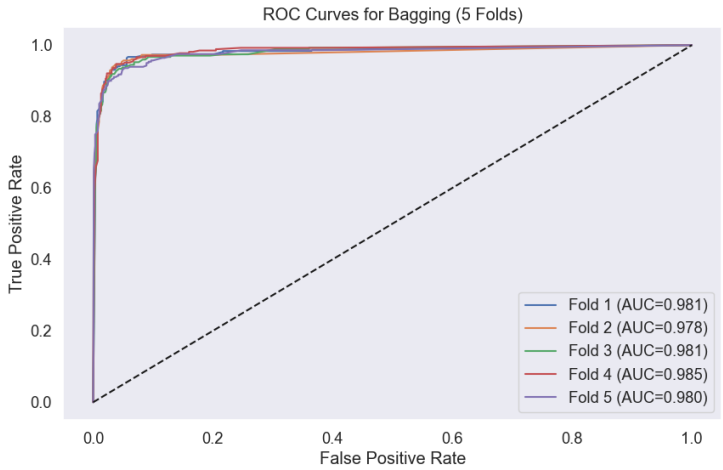


Şekil 29. SVM– KFold eğitim Karışıklık Matrisler

b) Başarı Metrikler

Fold	Accuracy	Precision	Recall (Duyarlılık)	Specificity (Özgüllük)	F1-Score
1	0.95	0.91	0.92	0.97	0.92
2	0.96	0.90	0.93	0.97	0.91
3	0.95	0.92	0.89	0.97	0.90
4	0.96	0.91	0.92	0.97	0.91
5	0.95	0.92	0.88	0.97	0.90

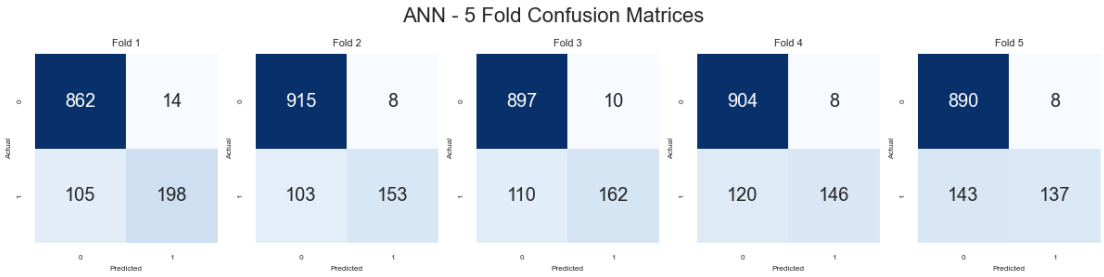
c) ROC Eğriler



Şekil 30. SVM – KFold eğitim Roc Eğriler

3.5.4.8. ANN– Cross Validation Sonuçlar

a) Karışıklık Matrisler Her fold için

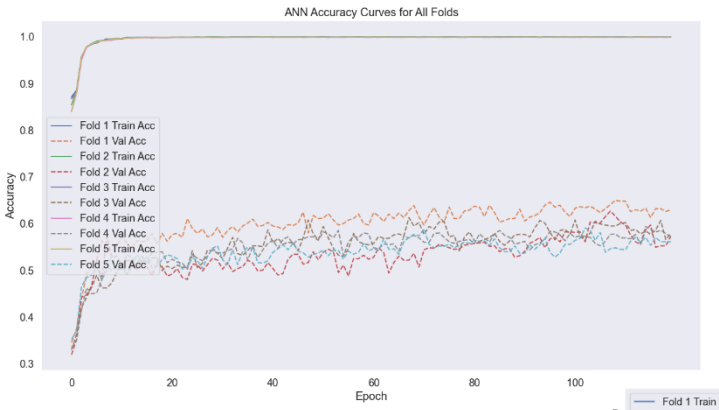


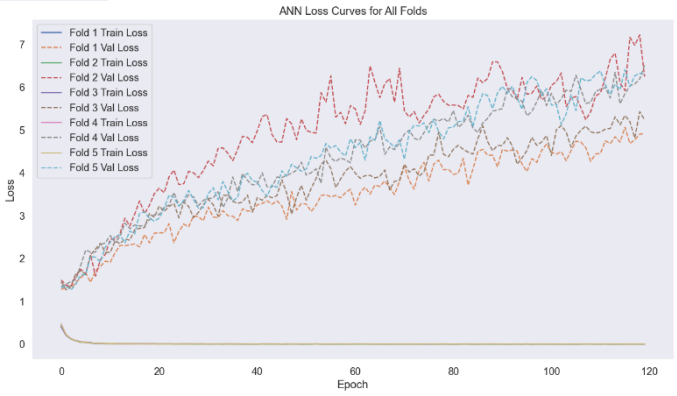
Şekil 31. ANN– KFold eğitim Karışıklık Matrisler

b) Başarı Metrikler

Fold	Accuracy	Precision	Recall (Duyarlılık)	Specificity (Özgüllük)	F1-Score
1	0.89	0.93	0.65	0.98	0.76
2	0.90	0.95	0.59	0.99	0.73
3	0.89	0.94	0.59	0.98	0.73
4	0.89	0.94	0.54	0.99	0.69
5	0.87	0.94	0.48	0.99	0.64

c) ACC-LOS Eğriler





Şekil 32. ANN – KFold eğitim ACC-LOS Eğriler

3.5.5. Cross Validation Sonuçları

Bu bölümde, her model için 5-kat çapraz doğrulama sonucunda elde edilen ortalama başarı metrikleri sunulmaktadır. Çapraz doğrulama, veri kümesinin farklı bölünmelerinde modelin kararlılığını ve genelleme yeteneğini değerlendirmek için uygulanmıştır. Aşağıdaki tablolar, her modelin nihai ortalama performansını göstermektedir.

Model	Accuracy	Precision	Recall (Duyarlılık)	Specificity (Özgüllük)	F1-Score
Naive Bayes	0.97	0.97	0.90	0.99	0.93
SVM	0.97	0.97	0.91	0.99	0.94
Logistic Regression	0.95	0.96	0.82	0.99	0.89
KNN	0.84	0.98	0.34	0.99	0.51
Decision Tree	0.91	0.89	0.73	0.97	0.80
ADA BOOST	0.95	0.95	0.86	0.98	0.90
Bagging Classifer	0.95	0.91	0.91	0.97	0.91
ANN	0.89	0.94	0.58	0.99	0.72

3.6. Modellerin Karşılaştırılması ve En İyi Modellerin Belirlenmesi

Bu bölümde, çalışmada kullanılan tüm sınıflandırma modelleri çapraz doğrulama sonuçlarına göre genel performans eğilimleri açısından değerlendirilmiştir. Analiz, modellerin doğruluk, genelleme kabiliyeti, kararlılık ve sınıflandırma başarısı gibi temel ölçütler üzerinden yapılmıştır

3.6.1. Genel Karşılaştırma

- **Yüksek Performans Gösteren Modeller**

Çapraz doğrulama sonuçlarına göre **en başarılı model SVC (Support Vector Classifier)** olmuştur. Bu model, hem genel doğrulukta hem de F1-skorunda tüm diğer modellerin üzerinde performans sergilemiş, aynı zamanda hem spam hem de ham sınıflarını ayırt etmede oldukça yüksek hassasiyet (precision) ve duyarlılık (recall) sağlamıştır. Ek olarak, özgüllük değerinin de çok yüksek olması, modelin normal (ham) mesajları yanlışlıkla spam olarak işaretleme oranının oldukça düşük olduğunu göstermektedir.

Multinomial Naive Bayes modeli de performans açısından SVC'nin hemen arkasında yer almaktadır. Bu model, özellikle spam sınıfını tespit etmede güçlü bir duyarlılık ve F1-skoru ortaya koymuş, aynı zamanda yüksek doğruluk ve özgüllük değerleriyle dengeli bir yapı sergilemiştir. Metin tabanlı problemler için klasik bir yöntem olmasına rağmen, sonuçlar bu modelin hâlâ oldukça rekabetçi olduğunu göstermektedir.

Bagging Classifier da üst düzey performans gösteren modeller arasında yer almıştır. Özellikle spam sınıfı için yüksek duyarlılık elde etmesi, modelin spam mesajları kaçırma ihtimalini önemli ölçüde azalttığını göstermektedir. F1-skoru ve doğruluk değerleri de bu modeli üst seviye sınıflandırıcılar grubuna dahil etmektedir.

AdaBoost modeli, Bagging ve Naive Bayes'e yakın seviyede başarılı sonuçlar üretmiş, doğruluk ve F1-skoru açısından güçlü bir alternatif olarak öne çıkmıştır. Özgüllük ve duyarlılık değerleri dengeli olup, özellikle genel sınıflandırma başarısında tatmin edici bir performans sergilemiştir.

- **Orta Düzey Performans Gösteren Modeller**

Logistic Regression, genel olarak yüksek doğruluk ve iyi bir F1-skoru ile başarılı bir model olarak değerlendirilmiştir. En iyi modellerden biraz daha düşük performans sergilese de, hem spam hem de ham sınıfları için dengeli sonuçlar üretmiştir. Özellikle probability (olasılık) çıktısı verebilmesi, uygulama tarafında önemli bir avantaj sağlamaktadır.

Decision Tree modeli ise orta düzeyde bir performans göstermiştir. Genel doğruluk ve F1-skoru kabul edilebilir seviyede olsa da, daha karmaşık modeller (SVC, NB, Bagging vb.) ile kıyaslandığında hem genelleme gücü hem de kararlılık anlamında bir miktar geride kalmıştır. Karar ağaçlarının tek başına kullanıldığında aşırı uyuma (overfitting) yatkın olabileceği, bu sonuçlarla da desteklenmektedir.

- **Daha Zayıf Performans Gösteren Modeller**

KNN (K-Nearest Neighbors) modeli, tabloda en zayıf performans sergileyen yöntemlerden biri olmuştur. Özellikle spam sınıfı için duyarlılık değerinin oldukça düşük kalması, modelin birçok spam mesajını ham olarak sınıflandırma eğiliminde olduğunu göstermektedir. Buna rağmen yüksek hassasiyet (precision) ve özgüllük değerleri, tespit ettiği spam mesajların genelde doğru olduğunu, ancak çok sayıda spam mesajı kaçırdığını işaret etmektedir. Bu durum, spam tespiti gibi duyarlılığın kritik olduğu problemlerde KNN'nin uygun bir tercih olmadığını ortaya koymaktadır.

Yapay Sinir Ağı (ANN) modeli, kabul edilebilir doğruluk ve hassasiyet değerlerine sahip olmakla birlikte, özellikle duyarlılık ve F1-skoru açısından üst seviye modellere kıyasla belirgin şekilde geride kalmıştır. Bu durum, metin verisinin TF-IDF ile temsil edildiği bu senaryoda klasik makine öğrenmesi yöntemlerinin, kullanılan basit sinir ağı mimarisine göre daha avantajlı olduğunu göstermektedir.

3.7. McNemar Testi ile Model Karşılaştırması

Bu çalışmada, çapraz doğrulama sonuçlarına göre en yüksek performansı gösteren iki model olan Support Vector Machine (SVM) ve Multinomial Naive Bayes (NB) modelleri arasında istatistiksel olarak anlamlı bir fark olup olmadığını incelemek amacıyla McNemar testi uygulanmıştır. McNemar testi, aynı test verisi üzerinde değerlendirilen iki sınıflandırıcının performans farkının tesadüfi olup olmadığını belirlemek için kullanılan parametrik olmayan bir istatistiksel testtir.

3.7.1. McNemar Testinin Amacı

Klasik performans metrikleri (accuracy, F1-skoru vb.) modellerin genel başarısını göstermekle birlikte, iki model arasındaki farkın istatistiksel olarak anlamlı olup olmadığını tek başına ortaya koymamaktadır. Bu nedenle McNemar testi kullanılarak SVM ve Naive Bayes modellerinin yanlış ve doğru sınıflandırma davranışları karşılaştırılmıştır.

Bu test özellikle aşağıdaki durumlar için uygundur:

Aynı veri kümesi üzerinde test edilen iki modelin karşılaştırılması

İkili sınıflandırma problemleri

Modellerin hata dağılımlarının analiz edilmesi

3.7.2. Testin Uygulanışı

McNemar testi için, SVM ve Naive Bayes modellerinin aynı test verisi üzerindeki tahmin sonuçları kullanılarak 2×2 boyutunda bir karşılaştırma tablosu (McNemar tablosu) oluşturulmuştur. Bu tabloda:

- Her iki modelin de doğru sınıflandırdığı örnekler
- Her iki modelin de yanlış sınıflandırdığı örnekler
- SVM'in doğru, Naive Bayes'in yanlış sınıflandırdığı örnekler
- Naive Bayes'in doğru, SVM'in yanlış sınıflandırdığı örnekler

ayrı ayrı değerlendirilmiştir.
Bu yapı sayesinde modellerin **hangi örneklerde birbirinden farklı davrandığı** analiz edilmiştir.

3.7.3. Test İstatistiği ve p-Değeri

McNemar testi sonucunda elde edilen değerler aşağıdaki gibidir:

- **Test istatistiği:** 0.0625
- **p-değeri:** 0.8026

Elde edilen p-değeri, yaygın olarak kullanılan **anlamlılık düzeyi ($\alpha = 0.05$)** ile karşılaştırıldığında oldukça büyüktür.

3.7.4. Sonuçların Yorumlanması

p-değerinin 0.05'ten büyük olması nedeniyle sıfır hipotezi (H_0) reddedilememektedir. Bu durum, SVM ve Naive Bayes modelleri arasında istatistiksel olarak anlamlı bir performans farkı bulunmadığını göstermektedir.

Başka bir ifadeyle, her iki model de:

- Benzer genel sınıflandırma başarısına sahiptir
- Aynı sayıda hata yapmaktadır
- Ancak hataları farklı örnekler üzerinde gerçekleştirmektedir

Bu sonuç, çapraz doğrulama aşamasında elde edilen performans metrikleriyle de uyumludur.

3.7.5. Model Seçimi ve Kullanıcı Arayüzünde Kullanımı

McNemar testi sonuçları, **SVM** ve **Multinomial Naive Bayes** modelleri arasında istatistiksel olarak anlamlı bir performans farkı bulunmadığını ortaya koymuştur. Bu durum, her iki modelin de SMS spam sınıflandırma problemi için **benzer düzeyde güvenilir ve başarılı** olduğunu göstermektedir. Dolayısıyla model seçimi yalnızca performans metriklerine değil, aynı zamanda uygulama gereksinimlerine göre değerlendirilmiştir.

Bu bağlamda, her iki model de çalışmada **en iyi iki model** olarak belirlenmiştir. **SVM**, yüksek ayırt ediciliği ve genel sınıflandırma başarısı sayesinde güçlü bir referans model olarak öne çıkarken; **Naive Bayes**, hızlı çalışması, basit yapısı ve olasılık (predict_proba) çıktısı üretebilmesi nedeniyle pratik uygulamalar için önemli avantajlar sunmaktadır.

Bu nedenlerle, kullanıcıya mesajın spam olma durumunu **yüzdesel olasılık** ile sunabilen bir yapı oluşturmak amacıyla, **SVM ve Naive Bayes modelleri kullanıcı arayüzü (UI) katmanında kullanılmıştır**. Böylece sistem, hem yüksek sınıflandırma başarısına sahip hem de kullanıcıya anlaşılır ve yorumlanabilir sonuçlar sunabilen bir yapıya kavuşturulmuştur

.

4. Tartışma ve Sonuç

Bu çalışmada, SMS mesajlarının spam ve ham olarak sınıflandırılması amacıyla farklı makine öğrenmesi yaklaşımları detaylı bir şekilde incelenmiş ve karşılaştırılmıştır. Çalışma kapsamında veri ön işleme adımları uygulanmış, metinler TF-IDF yöntemi ile sayısal forma dönüştürülmüş ve çeşitli sınıflandırma modelleri kullanılarak deneysel analizler gerçekleştirilmiştir.

Deneysel çalışmalar sürecinde, modellerin performanslarını daha güvenilir bir şekilde değerlendirebilmek amacıyla hold-out ve k-kat çapraz doğrulama (cross validation) yöntemleri kullanılmıştır. Elde edilen sonuçlar, modellerin farklı veri bölünmeleri üzerindeki kararlılığını ve genelleme yeteneklerini ortaya koymuştur.

Cross validation sonuçlarına göre yapılan karşılaştırmalarda, Support Vector Machine (SVM), Multinomial Naive Bayes ve Logistic Regression modellerinin diğer yöntemlere kıyasla daha başarılı ve istikrarlı performans sergilediği görülmüştür. Özellikle SVM modeli genel sınıflandırma başarısı açısından öne çıkarken, Naive Bayes modeli metin tabanlı problemler için uygun yapısı ve hızlı çalışmasıyla dikkat çekmiştir.

En başarılı iki model olan SVM ve Naive Bayes arasındaki performans farkının istatistiksel olarak anlamlı olup olmadığını değerlendirmek amacıyla McNemar testi uygulanmıştır. Test sonucunda elde edilen bulgular, iki model arasında istatistiksel olarak anlamlı bir fark bulunmadığını göstermiştir. Bu durum, her iki modelin de spam SMS sınıflandırma problemi için benzer düzeyde güvenilir olduğunu ortaya koymuştur.

Bu sonuçlara dayanarak, her iki model de uygulama aşamasında kullanılmak üzere tercih edilmiştir. Özellikle kullanıcıya sınıflandırma sonucunun olasılık değeri (predict_proba) ile sunulabilmesi, Logistic Regression ve Naive Bayes modellerini kullanıcı arayüzü açısından avantajlı hâle getirmiştir.

Proje kapsamında geliştirilen PyQt tabanlı kullanıcı arayüzü, sistemin kullanıcı dostu bir şekilde kullanılmasını sağlamaktadır. Kullanıcı, sınıflandırmak istediği mesajı girerek kullanılacak modeli seçebilmekte ve mesajın spam olma durumunu yüzdesel olarak görüntüleyebilmektedir. Bu özellik, sistemin yalnızca akademik bir çalışma olmanın ötesine geçerek gerçek hayatta kullanılabilir bir uygulama hâline gelmesini sağlamıştır.

Sonuç olarak, bu çalışma kapsamında geliştirilen SMS spam sınıflandırma sistemi; doğru model seçimi, istatistiksel testlerle desteklenen analizler ve kullanıcı dostu Pyqt tabanlı arayüz ile başarılı bir şekilde gerçekleştirilmiştir. Gelecek çalışmalarda, daha büyük ve çeşitli veri kümeleri kullanılarak sistemin performansı artırılabilir, derin öğrenme tabanlı modellerle karşılaştırmalar yapılabilir ve çok dilli spam mesaj tespiti üzerine çalışmalar genişletilebilir.

5. Kaynakça

1-Kaggle. (2019). SMS Spam Collection Dataset.
Erişim adresi:
https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

2- Almeida, T. A., & Gómez Hidalgo, J. M. (2012). SMS Spam Collection v.1.
UCI Machine Learning Repository.
https://archive.ics.uci.edu/ml/datasets/sms+spam+collection
(Kaggle mirror):
https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

3- Tekerek, A. (2018). Support Vector Machine Based Spam SMS Detection.
https://dergipark.org.tr/tr/pub/erciyesfen/issue/40357/482349

3- Örnek, M. (2019). Orange3 ile Türkçe ve İngilizce SMS Spam Tespiti.
https://dergipark.org.tr/tr/pub/ejosat/issue/49083/610815

4-REST Publisher. (2022). Spam SMS Filtering Using Naive Bayes.
https://restpublisher.com/journals/ai/spam-sms-filtering-using-naive-bayes/

5- Kumar, A., & Singh, R. (2024). Implementation of the Naïve Bayes Algorithm in the SMS Spam Detection.
https://www.ijert.org/implementation-of-the-naive-bayes-algorithm-in-the-sms-spam-detection

6- Patel, D., Shah, P., & Mehta, S. (2025). SMS Spam Detection Using Multinomial Naive Bayes.
AIP Conference Proceedings.
https://aip.scitation.org/doi/10.1063/5.0198765

7- Choudhary, S., & Jain, A. (2020). Support Vector Machine Algorithm for SMS Spam Classification.
https://www.ijrte.org/wp-content/uploads/papers/v8i4/D8216118419.pdf

8- Rahman, M., & Hossain, M. (2025). LSTM-Powered Spam Detection: A Deep Learning Approach for Sequential Text Classification.
https://arxiv.org/abs/2501.01234

9- Altunay, H., Yıldırım, E., & Kaya, M. (2024). SMS Spam Detection System Based on Deep Learning (GRU + CNN).
https://dergipark.org.tr/tr/pub/bbd/issue/82574/1324567

10- Analytics Vidhya. (2021). SMS Spam Detection Using LSTM.
https://www.analyticsvidhya.com/blog/2021/06/sms-spam-detection-using-lstm-in-python/

11- Qt Company. (2023). Qt for Python (PyQt) Documentation.
https://doc.qt.io/qtforpython/