



**MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ**

**SAHTE HABER TESPİTİ**

**Duhan KÖSALİ**

**LİSANS BİTİRME PROJESİ**

**İstanbul, Nisan 2023**

# **SAHTE HABER TESPİTİ**

**Duhan KÖSALİ**

**LİSANS BİTİRME PROJESİ**

**YAZILIM MÜHENDİSLİĞİ**

**LİSANS BİTİRME PROJESİ DANIŞMANI**  
Zeynep Behrin Güven AYDIN

## ÖNSÖZ

İlgili rapor, Beykoz Üniversitesi Mühendislik ve Mimarlık Fakültesi Yazılım Mühendisliği bölümü son sınıf öğrencisi Duhan Kösalı tarafından hazırlanmıştır. Projenin amacı, sahte haberlerin yayılması konusunda mevcut sorunlara çözüm sunan bir yazılım tasarlamaktır.

Bu rapor, proje boyunca yürütülen çalışmaları ve elde edilen sonuçları içermektedir. Çalışmalar, gerekli literatür araştırmalarının yapılması, gereksinim analizlerinin yapılması, tasarım sürecinin gerçekleştirilmesi, kod yazma ve test etme aşamalarını içermektedir. İlgili projenin geliştirilmesi sürecinde edindiğim tecrübe, yazılım mühendisliği alanında gelişimime katkı sağlamıştır. Proje boyunca karşılaşılan zorluklar, problemlere farklı açılardan yaklaşma becerimi geliştirmiştir.

Bu projede emeği geçen başta danışmanım Zeynep Behrin Güven Aydın olmak üzere, diğer bütün öğretim görevlilerimize ve Beykoz Üniversitesi Rektörlüğüne teşekkürlerimi sunuyorum.

Bu rapor, yazılım mühendisliği alanında ilerleyen öğrencilere rehberlik etmesi amacıyla hazırlanmıştır. Bu nedenle, projede kullanılan yöntemler, teknolojiler ve elde edilen sonuçlar detaylı bir şekilde açıklanmaktadır.

İlgili raporun, yazılım mühendisliği alanında araştırma yapmak isteyen öğrencilere faydalı bir kaynak olmasını ümit ediyorum.

Saygılarımla,

Duhan Kösalı

# İÇİNDEKİLER

ÖNSÖZ.....	iii
İÇİNDEKİLER.....	iv
ÖZET .....	vi
ŞEKİL LİSTESİ.....	vii
TABLO LİSTESİ.....	viii
1 GİRİŞ.....	1
2 PROJE İÇERİĞİ.....	2
2.1 Kullanılan Teknolojiler .....	2
2.2 Projenin Katma Değeri .....	2
3 PROJENİN YAPISI .....	4
3.1 Veri Toplama .....	4
3.2 Veri Önışleme.....	5
3.3 Model Oluşturma .....	6
2.4 Model Eğitimi.....	6
3.5 Test ve Doğrulama .....	6
3.5.1 Test Adımı.....	6
3.5.2 Doğrulama Adımı .....	7
3.6 Sonuç Analizi .....	7
4 MAKİNE ÖĞRENMESİ VE SAHTE HABER TESPİTİ ÜZERİNE YAPILAN ÇALIŞMALAR.....	8

5	PROJENİN İÇERİĞİ VE PROJEDE KULLANILAN ALGORİTMALAR.....	11
5.1	GİRİŞ VE AMAÇ .....	11
5.2	Random Forest Algoritması .....	12
5.3	PROJEDE KULLANILAN ÖZELLİK ÇIKARMA YÖNTEMLERİ.....	15
5.4	LONG SHORT TERM MEMORY ) .....	17
6	WEB UYGULAMASI.....	28
7	SONUÇ.....	30
8	Kaynakça .....	32

# ÖZET

## MAKİNE ÖĞRENMESİ İLE SAHTE HABER TESPİTİ

Bu lisans bitirme projesi, makine öğrenmesi teknikleri kullanarak sahte haberleri tespit edilmesi amaçlamaktadır. Günümüzde, sahte haberler internet ve sosyal medya platformlarında yaygınlaşarak toplumda ciddi bir etki yaratmaktadır. Bu nedenle, sahte haberleri tespit etmek önemli bir sorundur ve otomatik bir sistem geliştirmek bu sorunu çözmek için bir çözüm olabilir. Bu projede, öncelikle bir sahte haber veri seti toplanmış ve veri seti üzerinde temizleme, önişleme ve özellik çıkarma işlemleri yapılmıştır. Daha sonra, makine öğrenmesi algoritmaları kullanılarak sahte haberlerin tespiti için bir sınıflandırıcı modeli oluşturulmuştur. İlgili projenin son aşamasında eğitilmiş olan modeller ile haber tespiti yapan bir web projesi geliştirilmiştir. Bu proje, sahte haberlerin tespit edilmesine yardımcı olan ve haber kaynaklarının güvenilirliğini arttıran bir sistem oluşturma amacı taşımaktadır.

**Anahtar Kelimeler:** Makine Öğrenmesi, Sahte Haber Tespiti, Modelin Oluşturulması, Doğruluk Oranı.

**Tarih:** 2023

## ŞEKİL LİSTESİ

Şekil 1 – İyileştirme Adımları .....	14
Şekil 2 – Doğrulama Skoru .....	14
Şekil 3 – LSTM Modeli.....	18
Şekil 4 - Eğitim ve Doğrulama Kaybı Grafiği.....	20
Şekil 5 - Eğitim ve Doğrulama Doğruluk Grafiği .....	21
Şekil 6 - Karışıklık Matrisi .....	21
Şekil 7 – Model Değişiklikleri.....	22
Şekil 8 - Karışıklık Matrisi .....	23
Şekil 9 - Eğitim ve Doğrulama Kaybı (Loss) Grafiği .....	24
Şekil 10 - Eğitim ve Doğrulama Doğruluk (Accuracy) Grafiği .....	24
Şekil 11 – Grid Search.....	26
Şekil 12 - Karışıklık Matrisi .....	27
Şekil 13 - Web Sitesi.....	28
Şekil 14 - Web Sitesi.....	29

## TABLO LİSTESİ

Tablo 1 – Metrik Tablosu.....	20
Tablo 2 – Metrik Tablosu.....	23



# 1 GİRİŞ

Sahte haberler, son yıllarda özellikle internetin yaygınlaşması ve sosyal medyanın kullanımının artması nedeniyle oldukça yaygınlaşmıştır. Bu projenin amacı, sahte haberlerin toplumsal etkisini azaltmak için bir web sitesi tasarlamaktır. Bu web sitesi arka planda ilgili veri setleri ve algoritmalar ile birlikte önceden gerçekleştirmiş olduğu makine öğrenmesi yöntemleriyle beraber, kullanıcılara sahte ve gerçek haberleri ayırt etme konusunda yardımcı olmaktadır.

İlgili projede güncel algoritmalar kullanılmıştır ve büyük miktardaki veri setleri üzerinde eğitilerek geliştirilmiştir.

Projede çözülmesi hedeflenmiş olan problemler yani sahte haberlerin insanlar açısından oluşturabileceği sorunlar oldukça çeşitlidir. Bu sorunlardan bazıları:

- Yanlış Bilgilendirme.
- Toplumsal gerilim ve kutuplaşma.
- Demokratik süreçlerin etkilenmesi.
- Kişisel hakların ihlali.
- Finansal kayıplar vs.

Tüm bu sorunlar ile birlikte sahte haberler, genellikle gerçek haberlere göre daha dikkat çekici olabilirler, bu durum gerçek haberlere göre daha hızlı yayılmalarına neden olabilir. İlgili projede yayılma hızı yüksek olan sahte haberlerin önüne geçmek ve toplum bilincini sağlamak için geliştirilmiştir.

## 2 PROJE İÇERİĞİ

### 2.1 Kullanılan Teknolojiler

Projede kullanılan teknolojiler aşağıdaki şekildedir:

- Programlama dili olarak Python kullanılmıştır.
- Doğal Dil İşleme (NLP) yapısı için NLTK kütüphaneleri kullanılmıştır.
- Veri görselleştirme işlemleri için Seaborn, Matplotlib vb. kütüphaneler tercih edilmiştir.
- Veri setinin, proje yapısına uygun veri çerçevesi üzerinden incelenebilmesi için Pandas kütüphanesinden faydalanılmıştır.
- Algoritma olarak SVM, LSTM ve Random Forest kullanılmış ve sonuçlar proje dosyası ve raporuna kaydedilmiştir.

### 2.2 Projenin Katma Değeri

Ülkemizde ve Dünya’da haberlerin teyit edilmesi ile ilgili çalışmalar yürüten pek çok kuruluş bulunmaktadır. İlgili proje, sahte haberlerin tespiti ile ilgili bir çözüm sunmakta ve bu tarz çalışmalar yapan kuruluşlara daha hızlı ve alternatif bir çözüm sunmaktadır. Yani özetle projenin, medya kuruluşları ve haber siteleri tarafından kullanılması mümkündür. Bütün bunların dışında ilgili proje, benzer konular üzerinde yapılabilecek çalışmalarda kaynak olarak kullanılabilir.

## 2.3 Projenin Kapsamı

İlgili projede, sahte haberlerin tespiti için 3 farklı makine öğrenmesi modelleri oluşturulmuştur. Proje, veri toplama, veri ön işleme, özellik çıkarma, sınıflandırıcı model oluşturma ve doğrulama adımlarını içermektedir.

## 2.4 Projenin Özellikleri

İlgili proje bünyesinde pek çok özelliği barındırmaktadır. Bu özelliklerden başlıcalarını şu şekilde sıralayabiliriz:

- Sahte haber veri setinin toplanması ve temizlenmesi.
- Veri ön işleme ve özellik çıkarma adımlarının gerçekleştirilmesi
- Makine öğrenmesi algoritmalarının kullanılarak bir sınıflandırıcı modelinin oluşturulması.
- Modelin doğruluğunun analiz edilmesi.
- Analizler sonucunda projenin bir web uygulaması ile birlikte kullanıcı testine uygun hale getirilmesi.

## 2.5 Kısıtlamalar

Projenin kısıtları:

- Veri seti, sınırlı sayıda haber kaynağından oluşmaktadır.
- Veri seti İngilizce haberlerden oluştuğu için yalnızca İngilizce haberlerin tespiti gerçekleştirilmektedir.
- Veri setindeki haberlerin doğruluğu ve güvenilirliği konusu üzerinde çalışılmıştır. Fakat bu konuda kesin bir bilgi sahibi olmak bazı durumlar için mümkün olmayabilir.
- Modelin yüzde yüz doğru sonuç vermesi mümkün değildir.

## 3 PROJENİN YAPISI

İlgili proje, makine öğrenmesi yöntemlerinin kullanıldığı bir yapay zeka sistemi olarak kurgulanmıştır. Bu sistemde, sahte haberlerin tespit edilebilmesi için NLP olarak bilinen doğal dil işleme tekniklerinden faydalanılmıştır. Böylelikle haberlerin içeriği analiz edilmiştir. Algoritma olarak Random Forest, LSTM ve SVM gibi güçlü yapay zeka algoritmalarına başvurulmuştur.

### 3.1 Veri Toplama

Belirli amaçlar için toplanan bilgilere veri denir (Üstün, 2016). Sahte haberlerin ve gerçek haberlerin ayrımı için kullanılacak verilerin toplama işlemi gerçekleştirilmiştir.

İlgili projede 2 farklı veri seti kullanılmıştır. “Fake.csv” ve “Real.csv” bu iki veri setide 4 sütundan oluşmaktadır:

- Title: Haber başlığı
- Text: Haber içeriği
- Subject: İlgili haberin konusunu belirtmektedir.
- Date: İlgili haberin tarihi

Projede verisetinin ‘Text’ kısmından faydalanılmıştır. Ayriyeten iki veri seti birleştirilip sahte haberlere ‘0’, gerçek haberlere ‘1’ etiketi atanmıştır.

Verisetindeki haberlerin tamamı İngilizce’dir ve politik haberlerden oluşmaktadır. Veri seti seçilirken veriler arasında aşırı uyum olmamasına önem gösterilmiştir.

### 3.1.1 Aşırı Uyum (Overfitting)

Aşırı uyum, denetimli makine öğreniminde temel bir sorundur. Modelin eğitim verilerine aşırı uyum sağlamasından kaynaklanır (Ying, 2019). Overfitting, aşırı karmaşık modellerin kullanılması veya az sayıda veri örneği ile çalışma durumunda sıkça görülür.

Aşırı uyumu önlemek için, genellikle daha az karmaşık modeller kullanılır, daha fazla veri toplanır veya veriler rastgele örneklemelerle bölünerek eğitim ve test verileri olarak kullanılır. (Cook, 2016)

## 3.2 Veri Önışleme

Veri önışlemesi, makine öğrenmesi modelleri için önemli bir adımdır. Veri önışlemesi, veri kümesinin doğru bir şekilde temizlenmesi, düzenlenmesi ve hazırlanmasını sağlar (Şirin, 2017). Veri önışlemesinin bazı adımları aşağıda verilmiştir:

- Veri Temizleme: Modelin performansının arttırılabilmesi için veri kümesindeki eksik veya hatalı verilerin tespit edilip çıkarılması işlemidir.
- Veri Dönüştürme: Veri kümesindeki özelliklerin birleştirilmesi veya özelliklerin çıkarılması işlemidir. Bu işlem, modelin daha az gürültülü verilerle çalışmasını sağlar.
- Veri Bölme: Veri kümesinin eğitim, doğrulama ve test verileri olarak bölünmesi işlemidir. Bu işlem, modelin verilerle test edilmesini sağlar. Ayrıca aşırı uyum sorununu önler.

### **3.3 Model Oluřturma**

İlgili projenin başarısını etkileyen en önemli adımların birisi model oluřturma ařamasıdır. Bu adım sahte ve gerek haberlerin analiz edilmesi, model algoritmasının seilmesi, parametrelerin belirlenmesi ve modelin yapılandırılması srecini ierir.

Model oluřturma sırasında dikkat edilmesi gereken önemli unsurlardan birisi model algoritmasının seim ařamasıdır. (Chollet, 2017)

Bu nedenle projede, eřitli algoritmalar test edilmiřtir ve hangi algoritmanın başarılı olacaėı web projesi ile birlikte kullanıcının testine aık hale getirilmiřtir.

### **2.4 Model Eėitimi**

Model eėitimi, belirlenen parametreler ve veri seti kullanılarak gereleřtirilir. Bu adımda, modelin hata oranı hesaplanır ve en iyi sonucun alınacaėı algoritma kullanılarak parametremiz gncellenir. Bu iřlem belirlenen sre boyunca tekrar edilir.

Sonu olarak, model performansında deėerlendirme iřlemleri yapılır. Eėer deėerlendirme sonucu yeterli deėilse problemler analiz edilir. Gerekli deėiřiklikler yapıldıktan sonra iřlem tekrarlanır. (Chollet, 2017)

### **3.5 Test ve Doėrulama**

İlgili proje gibi makine ėrenmesi projelerinin en önemli adımlarından birisi test ve doėrulama adımıdır. Modelin genel doėruluėunun llebilmesini saėlar. (Ay, 2020)

#### **3.5.1 Test Adımı**

Test adımı, modelimizin eėitim sreci boyunca kullanılmıř olan sahte ve gerek haberler zerinden model performansının lldėė adımıdır. Modelin doėru sınıflandırma oranı, eřitli metriklerin hesaplanmasıyla belirlenir. (Ay, 2020)

### **3.5.2 Doğrulama Adımı**

Doğrulama adımı, modelimizin eğitim sürecinde kullanılmamış sahte ve gerçek haberlerden oluşan farklı bir veri seti kullanılarak modelin performansının ölçülmesini içerir. Bu adımda modelin genel performansı ölçülür. Ayrıca modelin bu adımdaki performansını arttırarak ilgili modelin gerçek dünya problemlerine karşı uygunluğu arttırılır. (Ay, 2020)

## **3.6 Sonuç Analizi**

Sonuç analizi bölümünde, projenin doğruluğu ve başarı oranı ile ilgili grafikler yer almaktadır.

Sonuç analizi aşamasında, projenin yeterli başarıyı sağlayamadığı senaryoda

- Farklı makine öğrenmesi algoritmaları kullanılarak bu problem aşılımaya çalışılmıştır.
- Farklı ve daha geniş veri setleri kullanılarak makinenin öğreniminde kullanılacak haber kaynakları genişletilmeye çalışılmıştır.

## 4 MAKİNE ÖĞRENMESİ VE SAHTE HABER TESPİTİ ÜZERİNE YAPILAN ÇALIŞMALAR

Makine öğrenmesi ve yapay zeka teknolojileri, birçok uygulama alanında olduğu gibi, sahte haber tespitlerinde de etkili bir araç olarak kullanılmaktadır. Bu alanda yapılan çalışmalar bu projede de olduğu gibi, genel olarak doğal dil işleme (NLP) tekniklerinin ve sınıflandırıcı algoritmalarının kullanılması üzerine yoğunlaşmaktadır.

Sahte haber tespitlerine yönelik makine öğrenmesi kullanımı ile ilgili çalışmaların tarihsel sürecine bakıldığında, ilk önemli adımların 2000'lerin sonları ve 2010'ların başlarında atıldığı görülmektedir. Bu dönemde, metin sınıflandırma ve doğal dil işleme alanlarında yapılan ilerlemeler, sahte haber tespiti için de kullanılabilecek temel tekniklerin geliştirilmesini sağlamıştır. (Arkaitz Zubiaga, 2018)

2016 ABD Başkanlık seçimleri sonrasında sahte haberlerin toplum üzerindeki etkisi daha belirgin bir hal almaya başlamıştır. Bu durum, sahte haberlerin tespitine yönelik yapılan çalışmaların artmasında önemli rol oynamıştır. Bu süreçte, sınıflandırma algoritmaları ve derin öğrenme teknikleri ile sahte haberlerin tespitine yönelik çeşitli araştırmalar yapılmıştır. (Kai Shu, 2017)



Son yıllarda, özellikle transformer tabanlı modellemelerin (BERT vb.) ve bu modellerin dil anlama ve metin sınıflandırma yeteneklerinin artması, sahte haber tespitinde büyük bir potansiyel oluşturmıştır. Bu modeller, metin içerisindeki ince anlam farklılıklarını ve bağlamları anlama yetenekleri ile sahte haberleri daha etkin bir şekilde tespit edebilirler. (Alec Radford, 2019)

Aynı zamanda, sahte haber tespitinde metinlerin yanı sıra metaverilerin kullanılması da bir başka araştırma alanını oluşturmaktadır. Metaveriler, haberin yazarı, yayınlandığı tarih, içerdiği bağlantılar ve haberin bulunduğu kaynak gibi bilgileri içerir. Bu tür bilgiler, sahte haberleri tespit etmek için kullanılan algoritmaların performansını artırabilir. (Natali Ruchansky, 2017)

Makine öğrenmesi ile sahte haber tespiti üzerine yapılan güncel çalışmalara bakıldığında derin öğrenme tekniklerinin kullanımı ön plana çıkmaktadır. Bu alanda çalışma yapan bir çok araştırmacı, özellikle metin sınıflandırma ve duygu analizi gibi konularda derin öğrenme tekniklerinin etkili olduğunu belirtmektedirler. (Verónica Pérez-Rosas, Automatic Detection of Fake News, 2017)

Bu alanda çalışan bazı öne çıkan araştırmacılar ve kurumlar şunlardır:

- MIT Media Lab: Sahte haberlerin tespiti ve yayılmasının önlenmesi üzerine birçok proje yürütmektedirler.
- Stanford University: Sahte haberlerin sosyal medya üzerindeki etkisini inceleyen çalışmalar yürütmektedirler.

## **4.1 MIT Media Lab**

MIT Media Lab'daki arařtırmacılar, makine öğrenmesi ve sosyal ağ analizi tekniklerini kullanarak sahte haberlerin yayılma dinamiklerini analiz etmeye yönelik çalışmalar yapmıştır. Vosoughi, Roy ve Aral (2018) tarafından yapılan bir çalışmada, Twitter'daki sahte haberlerin gerçek haberlerden daha hızlı ve daha geniş bir kitleye yayıldığı bulunmuştur. Bu çalışma, sahte haber tespitinde kullanılan makine öğrenmesi modellemelerini geliřtirmek için önemli bir referanstır.

## **4.2 Stanford University**

Stanford Üniversitesi'nde sahte haber tespitine yönelik yapılan çalışmalardan bir tanesi “Deepfake” teknolojisi üzerine odaklanmıştır. Zahavy, Benaim ve Shmueli (2021) tarafından yapılan bu çalışmada, derin öğrenme teknikleri kullanılarak sahte video içeriklerinin tespiti üzerine yoğunlaşmıştır.

Özetle makine öğrenmesi ile sahte haber tespiti, hem geçmişteki gelişmelerle hem de güncel teknolojik yeniliklerle sürekli olarak ilerleyen bir alandır. Bu konuda yapılan arařtırmalar, toplumun sahte haberlerin olumsuz etkilerinden korunmasına yardımcı olacak tekniklerin ve yöntemlerin geliştirilmesine katkı sağlamaktadır.

## 5 PROJENİN İÇERİĞİ VE PROJEDE KULLANILAN ALGORİTMALAR

### 5.1 GİRİŞ VE AMAÇ

Bu projenin amacı, sahte haberlerin tespit edilmesi için makine öğrenmesi yöntemlerini kullanarak bir model geliştirmektir. İlk aşamada “Real.csv” ve “Fake.csv” isimli iki farklı veri seti kullanılarak birleştirilmiş bir veri çerçevesi oluşturulur. Bu veri setlerindeki veriler kelime düzeyinde ayrıştırılırken, noktalama işaretleri ve büyük harfler gibi gürültü unsurlarından arındırılmıştır.

Projenin ikinci aşamasında, üç farklı algoritma kullanılmıştır: Random Forest, LSTM (Long Short-Term Memory) ve SVM. Random Forest ve SVM algoritmaları için, Count Vectorizer ve TF-IDF isimli vektörleştirme yöntemleri kullanılmıştır. Bu yöntemler, metin verilerini sayısal vektörlere dönüştürerek algoritmanın kullanabileceği formata getirir. LSTM algoritması için ise bir ‘tokenizer’ kullanılmıştır. Tokenizer, metin verisini daha küçük parçalara ayırarak algoritmaya giriş yapmak üzere hazırlar. Daha sonra ilgili algoritmalar train-test-verification işlemlerinden geçer ve başarı oranları ortaya çıkarılır.

Son aşamada, Flask kullanılarak bir web uygulaması geliştirilmiştir. Bu uygulama, train-test-verification işlemlerinden yüksek başarı oranı ile geçmiş olan modellerin son kullanıcı tarafından test edilmesi için bir arayüz sağlamaktadır. Kullanıcılar, açılır menüler aracılığıyla istediği algoritmayı (LSTM, SVM veya Random Forest) seçebilir. Örneğin Random Forest algoritmasını seçmiş olan bir kullanıcı, ek olarak Count Vectorizer veya TF-IDF yöntemlerinden birisini seçerek bir haber metni girebilir. Makine öğrenmesi modeli, kullanıcının girdiği haberi gerçek veya sahte olarak tahmin etmek için kullanılır.

## **5.2 Random Forest Algoritması**

Random Forest algoritması, bir topluluk öğrenme yöntemi olarak, birden çok karar ağacını bir araya getirerek daha güçlü ve genellenebilir bir model oluşturmayı hedefler. 2001 yılında Leo Breiman tarafından geliştirilen bu algoritma, karar ağaçlarından oluşan bir ‘orman’ oluşturur ve bu ormandaki tüm ağaçların tahminlerini birleştirerek nihai çıktıyı üretir. (IBM, tarih yok)

### **5.2.1 Random Forest’in Çalışma Prensipleri**

Random Forest algoritması, öğrenme sürecinde bir dizi karar ağacı oluşturur. Her ağaç rastgele seçilen özelliklerin ve veri noktalarından oluşturulan bir alt küme üzerinde eğitilir. Bu süreç, bir çeşit bootstrap örnekleme olarak bilinen bir teknik ile gerçekleştirilir. Bagging yöntemi, genellenebilirliğini artırır ve aşırı uyum problemini azaltır. (Breiman, 2001)

Her ağaç tahminleri sınıflandırma veya regresyon için birleştirilir. Bu nedenle, Random Forest tek bir karar ağacına göre çok daha iyi bir sonuç verir.

## **5.2.2 Random Forest’ın Kullanım Alanları**

Random Forest algoritması, geniş bir uygulama alanına sahip olan çok yönlü bir makine öğrenme modelidir. Stok fiyat tahmini, müşteri analizi, sahte haber tespiti, görüntü ve ses analizi vs. gibi pek çok alanda kullanılır.

## **5.2.3 Random Forest’ın Projedeki Kullanımı**

İlgili projede 3 farklı model bulunmaktadır. Bu modellerden bir tanesinde Random Forest algoritması kullanılmaktadır.

Bu modelde ilk aşamada, veri seti train ve test olarak iki ayrı parçaya ayrılmıştır.. Train-Test ayrımı yapılarak, geliştirilen modelin temel perfomansı değerlendirilmiştir. Veri setinin %80’i eğitim verisi olarak kullanılmış ve geri kalan %20’si ise test verisi olarak ayrılmıştır.

Eğitim sürecinin ardından geliştirilen model test verileriyle değerlendirilmiştir. Bu değerlendirme, modelin doğru sınıflandırma oranını yani doğruluk skorunu belirlemek için yapılmıştır. Vectorizer olarak ise Count Vectorizer ve TF-IDF Vectorizer olarak iki farklı yöntem ile test edilmiştir. TF-IDF Vectorizer’ında almış olduğu %80’lik başarı oranı ile kabul edilir bir sonuç almıştır. Fakat Count Vectorizer yöntemi yeteri kadar başarılı bir sonuç vermemiştir.

Fakat projede yapılan bazı testler modelin overfitting (aşırı uyum) problemine takılmış olabileceğinin tespit edilmesiyle birlikte projede iyileştirme adımlarına gidilmesi gerektiğine karar verilmesine neden olmuştur.

### **İyileştirme Adımları:**

İlk aşamada elde edilen sonuçlar üzerine iyileştirme adımları atılmıştır. Modelin performansını arttırmak için çeşitli hiperparametre ayarlamaları yapılmıştır. Öncelikle, “n\_estimators” ve “max\_depth” gibi parametreler belirtilmiş ve bunlara belli değerler verilmiştir.

```

# Random Forest algoritması ile test-train yapıyoruz ve %98 lik bir başarı elde ettik.
# Yukarıda kütüphaneyi import ettik.
n_estimators_options = [50, 100, 200, 300]
max_depth_options = [10, 20, 30, 40, 50]

best_val_score = 0
best_n_estimators = None
best_max_depth = None

for n_estimators in n_estimators_options:
    for max_depth in max_depth_options:
        rfc = RandomForestClassifier(n_estimators=n_estimators, max_depth=max_depth, random_state=0)
        rfc.fit(xv_train, y_train)
        val_score = rfc.score(xv_val, y_val)

        if val_score > best_val_score:
            best_val_score = val_score
            best_n_estimators = n_estimators
            best_max_depth = max_depth

print(f"En iyi doğrulama skoru: {best_val_score}")
print(f"En iyi n_estimators: {best_n_estimators}")
print(f"En iyi max_depth: {best_max_depth}")

```

**Şekil 1 – İyileştirme Adımları**

Farklı parametre kombinasyonları kullanılarak modelin doğruluk skoru değerlendirilmiştir. Deneme ve hata yaklaşımlarıyla, en iyi doğruluk skoru elde etmek için en iyi parametre değerleri tespit edilmiştir. Bu süreçte, doğruluk skorunun en üst seviyeye çıkarmak için iterative bir yaklaşım benimsenmiştir.

Bütün bunların dışında son olarak overfitting (aşırı uyum) probleminin önüne geçilmesi için train-test olarak ayrılmış olan veri seti train, test ve verifaction olmak üzere 3 farklı şekilde ayrılmıştır.

Daha sonra bir önceki modelde de olduğu gibi 2 farklı vectorizer yöntemi ile proje test edilmiş ve yapılan testlerde TF-IDF'in daha başarılı olduğu sonucuna varılmıştır. Son olarak alınan çıktılarda başarı oranının %99.46 gibi yüksek derecelere kadar ulaştığı gözlemlenmiştir.

```

[44899 rows x 2 columns]
En iyi doğrulama skoru: 0.9946547884187082
En iyi n_estimators: 200
En iyi max_depth: 50
Test skoru: 0.9908685968819599

```

**Şekil 2 – Doğrulama Skoru**

## 5.3 PROJEDE KULLANILAN ÖZELLİK ÇIKARMA YÖNTEMLERİ

Makine öğrenmesinde özellik çıkarma yöntemleri, veri setlerindeki bilgileri sayısal özelliklere dönüştürmek için kullanılan tekniklerdir. Bu yöntemler, makine öğrenimi algoritmalarının çalışabilmesi için veri setlerini anlamlı bir şekilde temsil eden özelliklerin elde edilmesini sağlar. Bizim projemizde kullanmakta olduğumuz **TF-IDF** ve **Count Vectorizer** metin verilerinden özellik çıkarmak için kullanılmaktadırlar. Bu yöntemlerle kelime frekanslarına veya kelime varlığına dayalı sayısal temsiller elde edilebilir. Ayrıca, önceden eğitilmiş kelime gömme (Word embedding) modelleri kullanılarak metin verileri vektörlere dönüştürülebilir. (Aggarwal, 2015)

### 5.3.1 TF IDF (Term Frequency-Inserve Document Frequency)

Metin madenciliği ve bilgi erişimi alanlarında sıkça kullanılan bir istatistiksel yöntemdir. Bu yöntem, belirli bir belgedeki bir kelimenin önemini belirlemek için kullanılır. TF-IDF belirli bir belgedeki bir kelimenin önemini belirlemek için o kelimenin iki özelliğini birleştirir. (Zhang, 2020)

- TF (Term Frequency), bir kelimenin belgedeki yaygınlığını temsil eder. Bir kelimenin ne kadar yaygın olduğu, o kelimenin ayırt edicilik özelliğini yansıtabilir.
- IDF, belirli bir kelimenin belgelerde kaç kez geçtiğini ve toplam belge sayısına oranını hesaplar. Bu değer daha sonra tersine çevrilir ve logaritması alınır.

TF-IDF, kelimenin belgedeki önemini hesaplamak için tek tek bulunan TF ve IDF değerlerini birleştirir. Yani TF-IDF değeri TF ve IDF değerlerinin çarpılmasıyla hesaplanır. Bu bulunan değer bir kelimenin bir dökümanda ne kadar önemli olduğunu gösterir.

TF-IDF genellikle bilgi erişimi, metin sınıflandırma, özetleme ve benzerlik eşleştirme gibi alanlarda kullanılır.

İlgili sahte haber tespiti modelimizde TF-IDF'in tercih edilmesinin bir kaç nedeni vardır. Bu nedenlerden bazıları:

- **Önem Saptama:** TF-IDF, bir kelimenin belgedeki önemini belirlemek için kullanılır. Bu özellik, sahte haberleri ayırt etmek için yararlı olabilir.
- **Nadir Terimlerin Önemi:** TF-IDF, nadir terimlerin önemini vurgular. Sahte haberlerde, özel terimler veya belirli kelime kombinasyonları sıkça kullanılabilir. Bu terimler, sahte haberlerin belirli temalarını veya içeriğini yansıtabilir. TF-IDF vektörizer, nadir terimlere yüksek ağırlık vererek bu ayırt edici terimlerin model tarafından yakalanmasını sağlayabilir. (Jurafsky, 2019)
- **Daha Az Örtüşme:** TF-IDF, yaygın kelimelerin ağırlığını azaltır. Yaygın kelimeler genellikle gerçek ve sahte haberlerde de benzer şekilde kullanılır. Bu durumda, yaygın kelimelerin sıklığına dayalı bir özellik temsili kullanmak, modelin gerçek ve sahte haberleri ayırt etmede zorlanmasına neden olabilir. TF-IDF, yaygın kelimelerin ağırlığını düşürerek daha özgün ve ayırt edici özelliklere odaklanılmasını sağlar.

### 5.3.2 Count Vectorizer

Makine öğrenmesi alanında, özellik çıkarma, ham verileri öğrenme algoritmaları tarafından etkili bir şekilde kullanılabilen bir formata dönüştürmede önemli bir rol oynamaktadır. Count Vectorizer, özellikle metin sınıflandırma ve bilgi erişimi gibi doğal dil işleme (NLP) görevlerinde sıkça kullanılan bu yöntemlerden biridir.

Count Vectorizer, aynı zamanda Bag of Words modeli olarak da bilinen, metin verilerini sayısal olarak temsil etmek için makine öğrenmesinde yaygın olarak kullanılan bir tekniktir. Bu yöntem, bir dizi metin belgesini bir matrise dönüştürür, burada her satır bir belgeye karşılık gelirken, her sütun tüm korpus içindeki benzersiz bir kelimeyi temsil eder. Bu yaklaşım, yalnızca kelimelerin frekansına odaklanır ve



sıralamalarını dikkate almaz, bu da seyrek bir matris temsiliyle sonuçlanır. (Manning, 2008)

Count Vectorizer'ın pek çok kullanım alanı vardır,

- **Metin Sınıflandırma:** Count Vectorizer, metin belgelerini özellik matrisine dönüştürerek metin sınıflandırma görevlerinde kullanılabilir. Örneğin, belirli bir metnin hangi kategoriye ait olduğunu tahmin etmek için sınıflandırma algoritmalarıyla birlikte kullanılabilir. Örnek olarak spam tespiti, duygu analizi veya haber kategorizasyonu gibi alanlar gösterilebilir.
- **Bilgi Erişimi ve Metin Madenciliği:** Count Vectorizer, bilgi erişimi ve metin madenciliği gibi alanlarda da kullanılabilir. Örneğin, belirli bir metindeki anahtar kelimeleri veya önemli terimleri belirlemek, belgeler arasındaki benzerlikleri veya ilişkileri keşfetmek için kullanılabilir.
- **Bilgi Çıkarma:** Count Vectorizer, metinlerden anlamlı bilgileri çıkarmak için kullanılabilir. Örneğin, bir metindeki önemli terimleri, konuları ve trendleri belirlemek için kullanılabilir. İlgili projemizde Count Vectorizer'ın buradaki kullanım alanının önemli bir yeri vardır.

## 5.4 LONG SHORT TERM MEMORY )

Derin öğrenme, makine öğrenmesinin bir alt dalıdır ve genellikle karmaşık ve büyük veri kümeleri üzerinde çalışmak için kullanılır. Derin öğrenme, verilerin karmaşıklığını ve çeşitliliğini yönetebilen çok katmanlı yapay sinir ağlarından yararlanır (Ian Goodfellow, 2016).

Derin öğrenme, özellikle doğal dil işleme, görüntü tanıma, ses tanıma gibi karmaşık veri tiplerini işlemek için kullanılan yapay sinir ağlarına dayanan bir öğrenme tekniğidir. LSTM, derin öğrenmenin bir formudur ve genellikle zaman serileri analizi ve dil modellemesi gibi sıralı verilerle çalışmak için kullanılır (Yann LeCun, 2015)

LSTM, özellikle sıralı veri gerektiren alanlarda etkilidir. Bu, doğal dil işleme,

konuşma tanıma, metin oluşturma ve zaman serisi tahmini gibi bir dizi uygulamayı içerir.

Bununla birlikte sahte haber tespitinde de son derece yararlıdır. Sahte haberler, genellikle yanıltıcı, yanlış veya taraflı bilgiler içeren haberlerdir ve toplum üzerinde önemli etkileri olabilir. LSTM, metin verilini ve özellikle sıralı bilgileri işlemek konusunda etkili olduğundan, bir metnin dil yapısını ve içeriğini analiz etmek ve böylece sahte olup olmadığını belirlemek için kullanılabilir. (Xinyi Zhou, 2020)

Sahte haber tespiti için kullanılan LSTM'ler, metnin dilini, söz dizimini ve sözcüklerin sıralı kullanımını modelleyerek, bir haber metninin sahte olup olmadığını tespit etme yeteneğine sahiptir. Bu, özellikle sahte haberlerin genellikle belirli bir dil ve söz dizimi örüntüsüne sahip olduğu durumlarda etkilidir.

Ayrıca, LSTM'ler, metnin zamanla nasıl değiştiğini modellemek için de kullanılabilir, bu da sahte haberlerin genellikle belirli bir süre zarfında yayıldığı durumlarda önemlidir. Bu, sahte haber tespit sistemlerinin, bir haberin yayılma hızını ve biçimini de dikkate alarak daha doğru sonuçlar vermesine yardımcı olabilir.

### 5.4.1 LSTM Algoritması ve Python

Python'da LSTM algoritması kullanmak için genellikle **Keras** kütüphanesi tercih edilmektedir. Keras, TensorFlow'un yüksek seviye bir API'sidir ve derin öğrenme modellerini kolayca oluşturmayı ve eğitmeyi sağlar.

İlgili projemin içerisinde bulunan **lstm\_DeepLearning.py** isimli dosyasında LSTM modelini oluşturmak için ilk olarak Keras kütüphanelerimi dahil ediyorum ve LSTM modeli oluşturulur.

```
# LSTM modeli
model = Sequential()
model.add(Embedding(max_vocab, 128, input_length=max_len))
model.add(Bidirectional(LSTM(32)))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Şekil 3 – LSTM Modeli

Yukarıdaki oluşturduğum modelde yaptığım işlemler şu şekildedir:

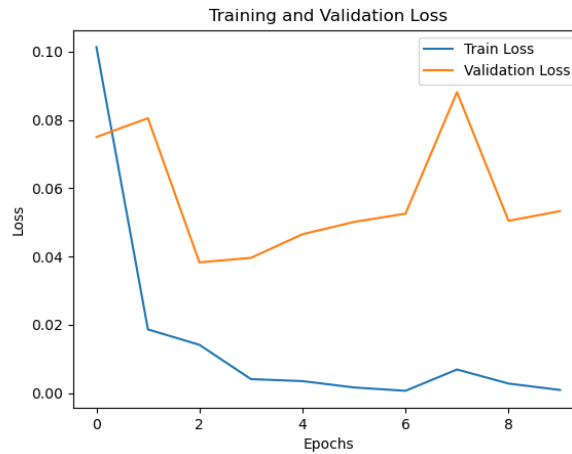
- Sequential methodu, katmanların sıralı bir şekilde yığıldığı bir model türüdür.
- Modelimizdeki ilk katman “Embedding” katmanıdır. Bu katman, kelime dağılımı “max\_vocab” yani 2000, vektör temsilleri 128, giriş verilerindeki her bir örneğin boyutu “input\_length” yani 250 olarak verilmiştir.
- İkinci katmanımız Bidirectional, bizim LSTM katmanımızdır. Bu katman, LSTM’nin hem geçmiş bilgiyi hem de gelecek bilgiyi modellemesini sağlar. LSTM’in hücre sayısı 32 olarak verilmiştir.
- Üçüncü katmanımız Dropout katmanıdır. Bu katman modelimizin aşırı uyumunu engellemek için kullanılmaktadır. Her eğitim adımında rastgele bir şekilde belirtilen orandaki giriş birimlerini sıfıra ayarlar. Bu modelimizde ilgili değer %20 olarak verilmiştir.
- Dördüncü katmanımız olan Dense katmanı, tam bağlantılı bir katmandır ve genellikle bir sinir ağı modelinin çıktı katmanı olarak kullanılmaktadır. Buradaki durumda çıktı boyutu 1’dir, aktivasyon fonksiyonu olarakta “sigmoid” kullanılmıştır. Böyle yapmamızın nedeni ilgili projemizin ikili bir sınıflandırma modeli olmasından kaynaklanmaktadır. Bir haber kaynağı “Gerçek” veya “Sahte” olabilmektedir.
- Son kısımda model derlemesi yapılmıştır. Optimizasyon algoritması olarak “adam”, kayıp fonksiyonu olarak “binary\_crossentropy” ve “accuracy” metriği kullanılmıştır.

İterasyon	Loss	Accuracy	Valudation Loss	Valudation Accuracy
1	0.1014	0.9602	0.0750	0.9744
2	0.0186	0.9944	0.0805	0.9755
3	0.0141	0.9955	0.0383	0.9898
4	0.0041	0.9988	0.0396	0.9902
5	0.0035	0.9989	0.0465	0.9893
6	0.0016	0.9996	0.0501	0.9889
7	6.4705e-04	0.9999	0.0526	0.9882
8	0.0069	0.9983	0.0882	0.9824
9	0.0028	0.9992	0.0505	0.9891
10	8.8569e-04	0.9998	0.0533	0.9889

**Tablo 1 – Metrik Tablosu**

Ayriyeten ilgili modelin başarı durumunu daha iyi gözlemleyebilmek için 3 farklı grafik oluşturuldu bu grafikler:

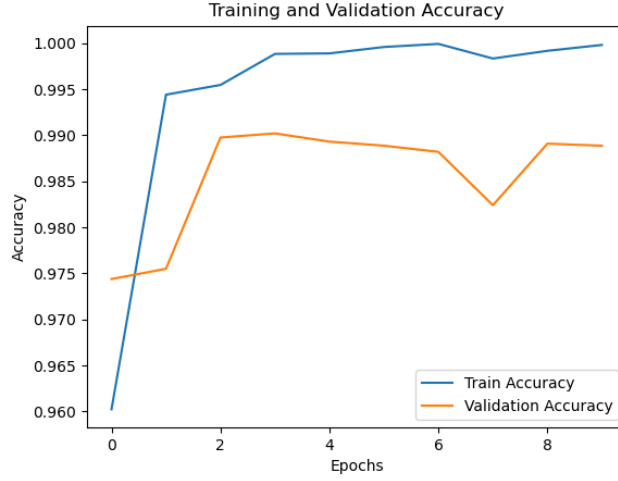
- Eğitim ve Doğrulama Kaybı (Loss) Grafiği: Bu, modelin eğitim ve doğrulama setleri üzerindeki kaybını her bir iterasyon boyunca gösterir. Bu, modelin aşırı uyuma (overfitting) başlayıp başlamadığını belirlemenin bir yoludur.



**Şekil 4 - Eğitim ve Doğrulama Kaybı Grafiği**

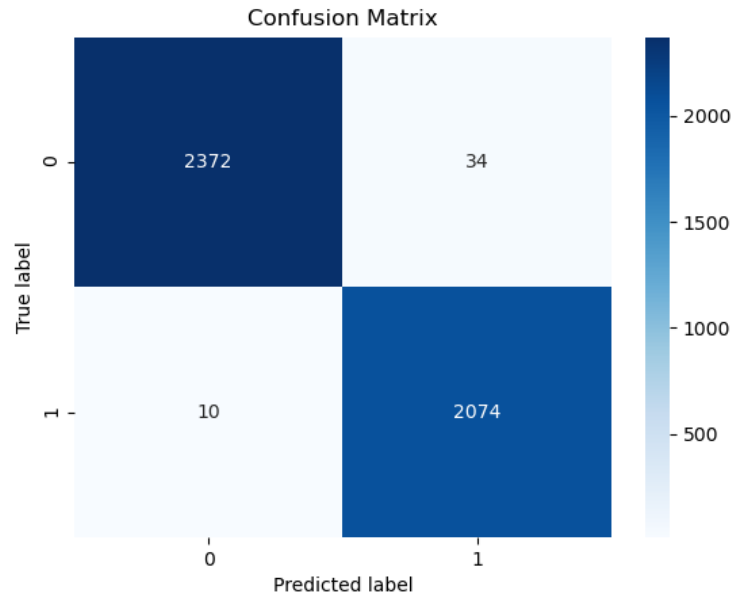
- Eğitim ve Doğrulama Doğruluk (Accuracy) Grafiği: Bu, modelin eğitim ve

doğrulama setleri üzerindeki doğruluğunu her bir iterasyon boyunca gösterir. Ayrıca modelin aşırı uyuma (overfitting) başlayıp başlamadığını belirlemenin bir başka yoludur.



Şekil 5 - Eğitim ve Doğrulama Doğruluk Grafiği

- Karışıklık Matrisi (Confusion Matrix): Bu, modelin test veri setindeki performansını gösterir. Doğru ve yanlış tahminlerin dağılımını gösterir, bu da modelin hangi sınıfları doğru olarak tahmin ettiğini ve hangi hatta hangi hataları yaptığını anlamana yardımcı olabilir.



Şekil 6 - Karışıklık Matrisi

İlgili modelden alınan çıktıları yorumlayacak olursak, eğitim ve doğrulama veri

setlerinde oldukça yüksek doğruluk elde edilmiş gözüküyor. Test veri setindeki %99.02 ile oldukça iyi bir başarı var. Ancak burada dikkate almamız gereken bazı hususlar bulunmaktadır.

Eğitim doğruluğunun çok yüksek olmasına rağmen doğrulama setindeki doğruluk oranı biraz düşüyor gibi görünüyor. Bu durum genellikle modelin aşırı uydurma (overfitting) olduğunun bir işareti olabilir. Bu nedenle modelimizde bir kaç değişikliğe gidiyoruz.

```
# LSTM modeli
model = Sequential()
model.add(Embedding(max_vocab, 128, input_length=max_len))
model.add(Bidirectional(LSTM(32, return_sequences=True))) # return_sequences=True ekleyerek daha fazla LSTM katmanı eklemeye izin verir
model.add(Dropout(0.5)) # Dropout miktarını artırdık
model.add(LSTM(32)) # İkinci bir LSTM katmanı ekledik
model.add(Dropout(0.5)) # İkinci Dropout katmanı
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# EarlyStopping callback'ini kullanın
early_stop = EarlyStopping(monitor='val_loss', patience=3) # 3 epoch boyunca iyileşme olmazsa eğitimi durdurur

# Modeli eğitimi
epochs = 10
batch_size = 32
history = model.fit(x_train_pad, y_train, epochs=epochs, batch_size=batch_size,
                    validation_data=(x_val_pad, y_val),
                    callbacks=[early_stop]) # callbacks parametresine early_stop ekledik
```

### Şekil 7 – Model Değişiklikleri

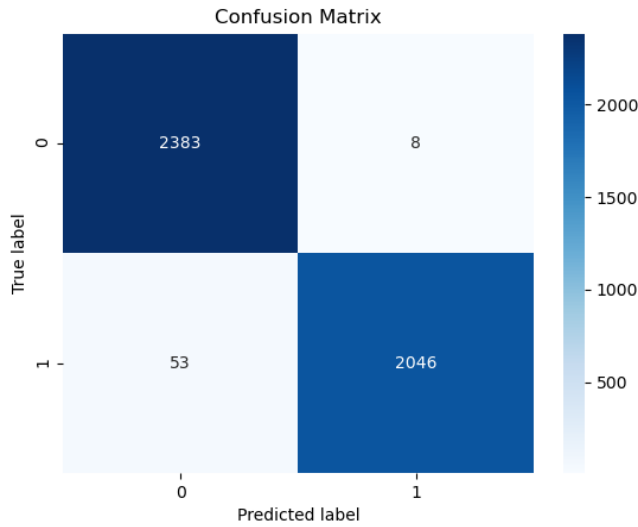
Yukarıdaki kod bloğunda modelimizin üzerinde yapılan değişiklikleri görebilirsiniz.

- "return\_sequences" değeri true olarak güncellenmiştir. Bu özelliği kullanarak 2. bir LSTM katmanı ekledik.
- Dropout miktarı artırıldı. Dropout, rastgele belirli nöronları "kapatarak" (yani onların ağırlıklarını geçici olarak sıfıra ayarlayarak) modelin öğrenme sırasında belirli nöronlara fazlasıyla bağımlı hale gelmesini engeller. Bu, modelin genelleme yeteneğini artırır ve overfitting'i azaltır. Dropout miktarını artırmak, daha fazla nöronun "kapatılmasına" yol açar, bu da genellikle daha güçlü bir düzenleştirme sağlar. Ancak, çok yüksek bir dropout oranı, modelin öğrenme yeteneğini olumsuz yönde etkileyebilir.
- İkinci bir Dropout katmanı ekledik.
- Tensorflow kütüphanesinin bize sunmuş olduğu EarlyStopping yöntemini kullandık. EarlyStopping bir geri çağırma (callback) türüdür ve modelin eğitimini, belirli bir koşul karşılandığında erken bir şekilde durdurmayı sağlar.

İterasyon	Loss	Accuracy	Valudation Loss	Valudation Accuracy
1	0.2883	0.8703	0.1883	0.9441
2	0.1534	0.9495	0.0697	0.9804
3	0.0997	0.9683	0.0571	0.9835
4	0.0300	0.9930	0.0420	0.9886
5	0.0183	0.9957	0.0431	0.9891
6	0.0068	0.9989	0.0491	0.9906
7	0.0123	0.9975	0.0370	0.9920
8	0.0126	0.9974	0.0498	0.9873
9	0.0049	0.9991	0.0494	0.9913
10	0.0079	0.9980	0.0612	0.9857

**Tablo 2 – Metrik Tablosu**

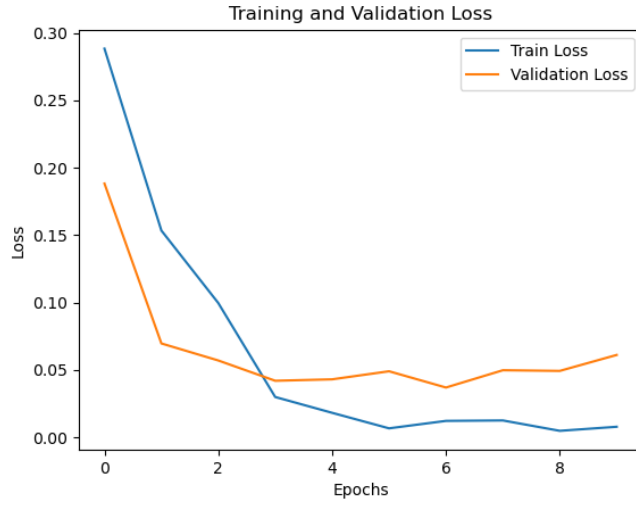
Karışıklık Matrisi (Confusion Matrix)



**Şekil 8 - Karışıklık Matrisi**

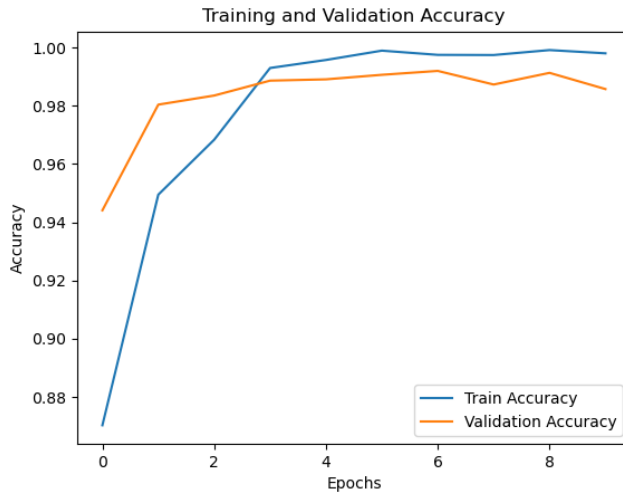
Bu matris incelendiğinde modelimin tahmin yeteniğinin bir önceki modeldeki kadar başarılı olduğunu söyleyebiliriz. Fakat bir önceki modelde yapılan testler ve diğer grafikler incelendiğinde “Overfitting” olduğu açıkça gözüküyordu. Bu nedenle ek olarak doğruluk ve kayıp grafiklerini incelememiz gerekiyor.

Eğitim ve Doğrulama Kaybı (Loss) Grafiği



**Şekil 9 - Eğitim ve Doğrulama Kaybı (Loss) Grafiği**

**Eğitim ve Doğrulama Doğruluk (Accuracy) Grafiği**



**Şekil 10 - Eğitim ve Doğrulama Doğruluk (Accuracy) Grafiği**

Yukarıdaki grafikler incelendiğinde 7. İterasyondan sonra doğrulama kaybı başladığını açık bir şekilde gözlemleyebiliyoruz.

Örneğin kayıp grafiğini incelediğimizde 7. itasyondan sonra Validation Loss değerinin artmaya başlaması overfitting oluştuğunun işaretçisi olabilir. Bu nedenle modelimizi 10 iterasyon yerine 7 iterasyon ile geliştirmemiz daha başarılı sonuçlar elde edebileceğimizi göstermektedir.



## 5.5 Support Vector Machines (SVM)

Destek Vektör Makineleri (Support Vector Machines - SVM), bir sınıflandırma ve regresyon analizi yöntemi olan denetimli öğrenme modelidir. Genel anlamda, iki sınıf arasında en geniş 'marj'ı sağlayan hiperdüzlemi bulmaya çalışır. Bu, öğrenme esnasında modelin genelleme kabiliyetini artırır ve overfitting'i önler.

SVM'ler, bir veri kümesindeki sınıflar arasında en geniş marjla ayırım yapabilecek bir hiperdüzlem bulmaya çalışır. Bir hiperdüzlem, iki sınıfı ayıran bir çizgi (iki boyutlu) veya bir düzlem (üç boyutlu) olabilir. Boyut sayısı arttıkça, bu ayırım şekli hiperdüzlem halini alır.

Bir SVM'nin temel amacı, marjı maksimize ederken, hiperdüzlem ve veri noktaları arasındaki hataları (yanlış sınıflandırmalar) minimize etmektir.

SVM algoritmaları doğal dil işleme (NLP) projelerinde kullanılabilir. Özellikle metin sınıflandırma, duygu analizi, sahte haber tespiti gibi problemler SVM'ler ile çözülebilir. Fakat SVM algoritması yavaş çalışmaktadır. Bu nedenle model eğitimleri uzun sürebilir.

### 5.5.1 Sahte Haber Tespitinde NLP

SVM, makine öğrenmesi ile sahte haber tespiti yapan bir model geliştirmek için kullanılabilir. Özellikle, metin verileri üzerinde çalışırken, SVM'nin yüksek boyutlu giriş verilerini işleyebilme yeteneği bu tür projeler için oldukça faydalıdır. Sahte haber tespiti genellikle bir metin sınıflandırma problemi olarak görülür. Bununla birlikte, bu tür bir proje başarılı olmak için geniş bir etiketli veri setine ihtiyaç duyar. Yani, her haberin 'sahte' veya 'gerçek' olduğunu belirten bir etiketleme gereklidir. İlgili proejde bu etiketleme verisetlerindeki sahte haberlerin '0', gerçek haberlerin '1' şeklinde etiketlenmesi ile sağlanmıştır. Bu veri seti üzerinde SVM modeli eğitilir ve daha sonra yeni, görülmemiş haberleri sınıflandırmak için kullanılır.

Son olarak, sahte haber tespiti gibi karmaşık problemler, çözüm için birçok farklı özellik ve algoritmanın birleştirilmesini gerektirebilir. SVM, bu tür bir çözümün

yalnızca bir parçası olabilir ve diğer NLP teknikleri ve algoritmaları ile birlikte kullanılabilir.

## Grid Search

İlgili SVM modelimizi Grid Search ile birlikte kullandık. Grid Search, her bir model konfigürasyonu için bir dizi model eğitir ve en iyi sonucu veren modeli seçer. Bu süreç, modelin karmaşıklığına, belirtilen parametrelerin sayısına ve veri setinin boyutuna bağlı olarak uzun sürebilir.

```
# GridSearch için parametreleri belirler
param_grid = {'C': [0.1, 1, 10, 100], 'gamma': [1, 0.1, 0.01, 0.001], 'kernel': ['linear', 'rbf']}

# GridSearchCV oluşturur ve eğitir
grid = GridSearchCV(model, param_grid, refit=True, verbose=2)
grid.fit(x_train, y_train)
```

### Şekil 11 – Grid Search

Belirttiğimiz parametre ızgarasında, dört farklı C değeri, dört farklı gamma değeri ve iki farklı kernel değeri bulunuyor. Bu durumda, her bir kombinasyon için bir model eğitiliyor. Dolayısıyla, toplamda 4 (C değerleri) \* 4 (gamma değerleri) \* 2 (kernel değerleri) = 32 farklı model eğitiliyor. Her bir model için 5-fold cross-validation (CV) kullanılıyor, yani her bir model için 5 kez eğitim gerçekleştiriliyor. Bu durumda, toplam eğitim sayısı  $32 * 5 = 160$  oluyor. Eğitim sayısını model üzerinden azaltarak modelimizin eğitim süresinin kısılmasını sağlayabiliriz fakat bu durum modelin başarı oranına zarar verebilir veya aşırı öğrenmeye (overfitting) yol açabilir.

Burada kullanılan ‘C’, ‘gamma’ ve ‘kernel’ parametreleri ve aldıkları değerlerin açıklamaları aşağıdaki şekildedir:

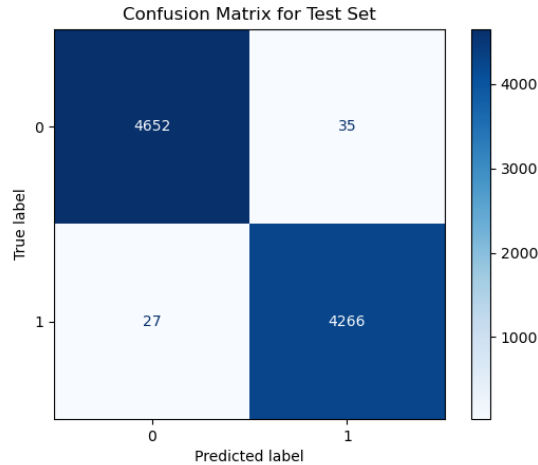
- ‘C’ parametresi: Bu, hata teriminin katsayısıdır ve modelin aşırı uyum arasındaki dengesini kontrol eder. Daha büyük bir C, daha az yanlılıkla sonuçlanır ve modelin bireysel veri noktalarına daha yakın olmasını sağlar, ancak bu durum aşırı uyuma neden olabilir. Daha küçük bir C, daha fazla yanlılıkla sonuçlanır ancak modelin genelizasyonunu artırır, yani yeni veri noktalarına daha iyi uyarlanabilir.

- ‘Gamma’ parametresi: Bu, radyal temel fonksiyon (RBF) çekirdeği için bir parametredir ve tek bir eğitim örneğinin etkisinin ne kadar olduğunu belirler. Büyük bir gamma, bir eğitim örneğinin etkisinin uzak mesafelere kadar uzandığı, daha kompleks bir karar sınırı oluşturduğu anlamına gelir, ancak bu da aşırı uyuma neden olabilir. Daha küçük bir gamma, bir eğitim örneğinin etkisinin kısa mesafeleri kapsadığı anlamına gelir ve daha basit bir karar sınırı oluşturur.
- ‘Kernel’ parametresi: Linear ve rbf olarak ikiye ayrılır. 'linear' çekirdek, doğrusal bir karar sınırı oluştururken, 'rbf' (Radyal Temel Fonksiyon), daha karmaşık, doğrusal olmayan karar sınırları oluşturabilir.

Modelimi çalıştırdıktan sonra bir takım sonuçlar elde ettim aldığım sonuçlara göre Grid Search’ın belirlediği en iyi parametreler ‘C’ = 100, ‘gamma’ = 0.1, ‘kernel’ = ‘rbf’ olarak belirlenmiştir.

Doğrulama başarı oranı 0.9918, test başarı oranı ise 0.9930 gibi yüksek bir sonuç göstermiştir.

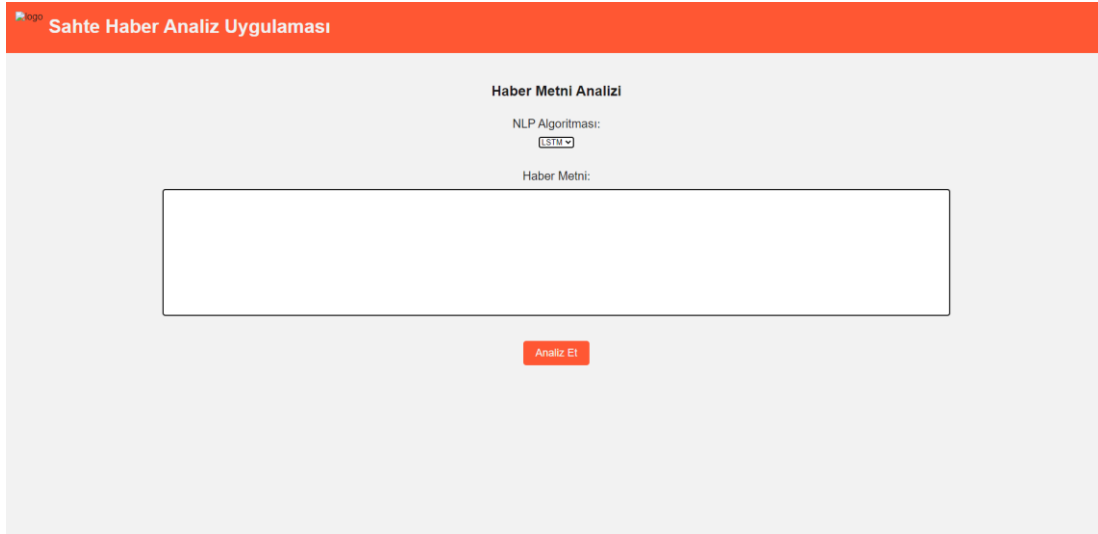
#### Karışıklık Matrisi (Confusion Matrix)



Şekil 12 - Karışıklık Matrisi

## 6 WEB UYGULAMASI


Web sitesinin geliştirme aşaması boyunca HTML, CSS, Javascript ve back-end işlemleri için ise Flask Framework’den faydalanılmıştır.



The screenshot shows a web application titled 'Sahte Haber Analiz Uygulaması' (Fake News Analysis Application). The main heading is 'Haber Metni Analizi' (News Text Analysis). Below this, there is a label 'NLP Algoritması:' followed by a dropdown menu currently set to 'LSTM'. Underneath is a label 'Haber Metni:' followed by a large, empty text input box. At the bottom center, there is a red button labeled 'Analiz Et' (Analyze).

Şekil 13 - Web Sitesi

Web sitesini genel tasarımı yukarıdaki gibidir. Kullanıcı ‘dropdown’ üzerinden Random Forest, LSTM veya SVM algoritmalarından birini seçer, daha sonra ‘textbox’ alanına teyit etmek istediği haberin içeriğini girer ve analiz et butonuna tıklar.

 **Sahte Haber Analiz Uygulaması**

**Analiz Sonuçları**

**Haber Metni:**

Russia may have effectively captured the symbolic prize of Bakhmut, but in many ways the battle for the city might only just be beginning. Moscow declared a triumphant victory in Bakhmut over the weekend, its first in nearly a year, with state media extolling its 'liberation' and President Vladimir Putin promising 'state rewards' to those who 'distinguished' themselves in the war's longest and bloodiest battle. However, Putin's troops — exhausted and depleted by the sort of fighting not seen in Europe since World War II — may struggle to push deeper into the eastern Donbas region while Kyiv's military will seek to take advantage of recent gains by trying to encircle them, according to Ukrainian officials and Western military analysts. 'Mission accomplished?' Russian state media headlines on Monday declared Bakhmut was already being de-mined after the country's defense ministry and mercenary chief Yevgeny Prigozhin said Saturday that forces led by Wagner fighters had taken full control of the battered city. Prigozhin posted images of his fighters raising flags over partially-destroyed buildings in the city, which has been left in ruins by months of conflict that has seen both sides suffer huge losses.

**Kullanılan Algoritma:**

SVM

**Analiz Sonucu:**

Girilen metin gerçek bir haber olabilir

**Geri Dön**

**Şekil 14 - Web Sitesi**

Kullanıcı analiz et butonuna tıkladıktan sonra ‘dropdown’ üzerinden seçmiş olduğu algoritmanın yedeği devreye girer ve ilgili metin analiz edilir. Analiz sonucu ‘0’ döndüğünde “Girilen metin sahte bir haber olabilir”, ‘1’ döndüğünde ise “Girilen metin gerçek bir haber olabilir” şeklinde geri dönüş alınır.

İlgili modellerin web uygulamasına dönüştürülmesindeki amaç projenin son kullanıcı tarafından anlaşılır ve doğrudan test edilebilir olması için geliştirilmiştir. Ayrıca web uygulaması projenin test edilebilirliğini doğrudan arttırmak içinde önemli bir yoldur.

## 7 SONUÇ

Bu çalışmada, en son makine öğrenmesi ve doğal dil işleme teknolojilerinin kullanılmasıyla bir haber analiz sistemi geliştirilmiştir. Bu süreçte, Random Forest, LSTM ve SVM olmak üzere makine öğrenmesinin üç farklı algoritması kullanılarak üç ayrı model tasarlanmış ve eğitilmiştir. Bu üç farklı modelin geliştirilmesinin amacı, bir platform oluşturarak kullanıcıların karşılaştıkları haberlerin gerçekliğini doğrulamalarını sağlamak ve böylece haberlerin güvenilirliği artırılmıştır.

Modellerin eğitim süreci oldukça detaylı bir şekilde yürütülmüştür. Eğitim boyunca, Confusion Matrix, Loss Graph, Accuracy Graph gibi değerlendirme teknikleri kullanılarak modellerin performansları analiz edilmiş ve böylece modellerin başarı oranları belirlenmiş ve eğitim sürecinde herhangi bir problem yaşanıp yaşanmadığı saptanmıştır. Bu detaylı analizler, projenin son raporuna eklenmiştir.

Eğitim sürecinin tamamlanmasının ardından, modeller 'pickle' kütüphanesi kullanılarak yedeklenmiştir. Bu adım, modellerin web projesine kolaylıkla entegre edilebilmesi amacıyla yapılmıştır. Projenin geliştirme aşaması boyunca oluşturulan tüm yedekler, güvenlik ve kolay erişim için proje dosyasında saklanmıştır. Bu yedekler daha sonra, Python'un popüler bir web çerçevesi olan Flask kullanılarak geliştirilen web sitesinin oluşturulmasında kullanılmıştır.

Sonu olarak, bu sistem, kullanıcıların bilgiye erişimindeki doğruluęu ve güvenilirlięi arttırmak için geliştirilmiştir. Ayrıca, bu sistem, bilgiye dayalı bir toplumda daha bilinli ve bilgiye dayalı kararlar almayı teşvik eden bir temel oluşturulmasında önemli bir rol oynamıştır. Bu alandaki ileri çalışmaların hedefi, bu tür bir platformun daha da geliştirilmesi ve böylece daha fazla insanın bu tür kararlar almasına olanak sağlanması olacaktır. Bilgiye dayalı ve bilinli bir toplumun oluşturulmasında bu tür sistemlerin geliştirilmesi kritik bir önem taşımaktadır.

## 8 Kaynakça

- (1) Alec Radford, J. W. (2019). *Language Models are Unsupervised Multitask Learners*. [www.cloudfront.net](http://www.cloudfront.net).
- (2) Arkaitz Zubiaga, A. A. (2018). *Detection and Resolution of Rumours in Social Media: A Survey*. ACM Digital Library.
- (3) Ay, Ş. (2020, Nisan 30). *Model Performansını Değerlendirmek - Metrikler*. Medium.com: <https://medium.com/deep-learning-turkiye/model-performansini-degerlendirmek-metrikler-cb6568705b1> adresinden alındı
- (4) Breiman, L. (2001). *Random Forests*.
- (5) Chollet, F. (2017). Deep Learning with Python. F. Chollet içinde, *Deep Learning With Python*. Perlego.
- (6) Geron, A. (tarih yok). Concepts, Tools, and Techniques to Build Intelligent Systems. A. Geron içinde, *Hands-on Machine Learning with Scikit-Learn, Keras & Tensorflow*.
- (7) Ian Goodfellow, Y. B. (2016). *Deep Learning*. MIT Press.
- (8) IBM. (tarih yok). *IBM*. IBM: <https://www.ibm.com/topics/random-forest> adresinden alındı
- (9) Kai Shu, A. S. (2017). *Fake News Detection on Social Media: A Data Mining Perspective*. Cornell University.



- (10) Manning, C. D. (2008). *Introduction to information retrieval*. Cambridge University.
- (11) Şirin, E. (2017). Büyük Veri Ön-İşleme. *Veri Bilimi Okulu*.
- (12) *TfidfVectorizer*. (tarih yok). Scikit-Learn: [https://scikit-learn.org/stable/modules/generated/sklearn.feature\\_extraction.text.TfidfVectorizer.html](https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html) adresinden alındı
- (13) Üstün, P. (2016). Veri Toplama Yöntemleri. *Ph Derneği*.
- (14) Verónica Pérez-Rosas, B. K. (2017). *Automatic Detection of Fake News*. Cornell University.
- (15) Xinyi Zhou, R. Z. (2020). *A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities*. ACM Computing Surveys.
- (16) Yann LeCun, Y. B. (2015). *Deep Learning*. [www.nature.com](http://www.nature.com).
- (17) Ying, X. (2019). *An Overview of Overfitting and its Solutions*. Iop Science.