

BURSA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ



TRENDYOL YORUM SINIFLANDIRMA PROJESİ

LİSANS BİTİRME ÇALIŞMASI

Taner Solak

Bilgisayar Mühendisliği Bölümü

HAZİRAN, 2023

BURSA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ



TRENDYOL YORUM SINIFLANDIRMA PROJESİ

LİSANS BİTİRME ÇALIŞMASI

Taner Solak
18360859034

Bilgisayar Mühendisliği Bölümü

Danışman: Prof. Dr. Turgay Tugay Bilgin

HAZİRAN, 2023

BTÜ, Mühendislik ve Doğa Bilimleri Fakültesi Bilgisayar Mühendisliği Bölümü'nün 18360859034 numaralı öğrencisi Taner SOLAK, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "TRENDYOL YORUM SINIFLANDIRMA PROJESİ" başlıklı bitirme çalışmasını aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Danışmanı : **Prof. Dr. Turgay Tugay Bilgin**
Bursa Teknik Üniversitesi

Jüri Üyeleri : **Dr. Öğr. Üyesi Adı SOYADI**
Bursa Teknik Üniversitesi

Öğr. Gör. Dr. Adı SOYADI
Bursa Teknik Üniversitesi

Savunma Tarihi : 1 HAZİRAN 2023

BM Bölüm Başkanı : Prof. Dr. Turgay Tugay Bilgin
Bursa Teknik Üniversitesi/...../.....

İNTİHAL BEYANI

Bu bitirme alışmasında grsel, işitsel ve yazılı biçimde sunulan tüm bilgi ve sonuçların akademik ve etik kurallara uyularak tarafımdan elde edildiğini, bitirme alışması içinde yer alan ancak bu alışmaya özgü olmayan tüm sonuç ve bilgileri bitirme alışmasında kaynak göstererek belgelediğimi, aksinin ortaya ıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim.

Öğrencinin Adı Soyadı: Taner Solak

İmzası :



ÖNSÖZ

Bitirme projemi geliştirme aşamasında takip eden değerli hocam Prof. Dr. Turgay Tugay Bilgin 'e ve üniversite hayatım boyunca ders aldığım tüm saygı değer hocalarıma katkıları için teşekkür ederim.

Temmuz 2023

Taner Solak

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	v
İÇİNDEKİLER	vi
KISALTMALAR	vii
ŞEKİL LİSTESİ.....	viii
ÖZET.....	ix
SUMMARY	x
1. GİRİŞ.....	11
1.1 Hipotez	12
2. LİTERATÜR TARAMASI	13
3. KULLANILAN TEKNOLOJİLER.....	15
3.1 Python.....	15
3.1.1 Pandas ve Numpy.....	16
3.1.2 Natural Language Toolkit (NLTK).....	16
3.1.3 Pickle.....	16
3.1.4 Flask	16
3.2 Google Colab (Colaboratory).....	17
4. METODOLOJİ	18
4.1 Proje Geliştirme Süreci	18
4.2 Proje Taslağının Oluşturulması	19
4.3 Verinin toplanması ve saklanması.....	20
4.3.1 Selenium.....	20
4.4 Veri Ön İşleme	21
4.4.1 Metin Temizleme	22
4.4.2 Tokenize Etme	23
4.4.3 Stop Kelime Kaldırma	25
4.4.4 Vektörizasyon	25
4.5 Veri Görselleştirme	27
4.6 Makine Öğrenmesi	30
4.6.1 Modelin Oluşturulması	31
4.6.2 Modelin Kaydedilmesi	33
4.6.3 Model Performansı Değerlendirme.....	34
4.7 Modelin Ürünleştirilmesi	39
4.7.1 Flask ile Sunucu Oluşturma	39
5. UYGULAMA ÇIKTILARI.....	43
6. SONUÇ	45
6.1 Çalışmanın Uygulama Alanı	45
7. KAYNAKÇA	47
ÖZGEÇMİŞ.....	48

KISALTMALAR

NLP	: Natural Language Process
KNN	: K-Nearest Neighbor
NB	: Naive Bayes
MLP	: Backpropagation
SVM	: Support Vector Machine
LR	: Logistic Regression
NLTK	: Natural Language Tool Kit
WSGI	: Web Server Gateway Interface
API	: Application Programming Interface
GPU	: Graphics Processing Unit
TPU	: Tensor Processing Unit
HTML	: HyperText Markup Language
TPR	: True Positive Rate
FPR	: False Positive Rate
ROC	: Receiver operating characteristic
URL	: Uniform Resource Locator
TF-IDF	: Term Frequency-Inverse Document Frequency

ŞEKİL LİSTESİ

Sayfa

Şekil 4.1 Çalışma programı için Gantt şeması.	18
Şekil 4.2 Proje algoritmasının çalışma şeması.	19
Şekil 4.3 Verilerin saklanma şekli.	20
Şekil 4.4 Veri ön işleme yapan fonksiyon.	22
Şekil 4.5 Fonksiyonun içinde tokenize işlemini yapan kod.	24
Şekil 4.6 Metinlerin tokenize edilmesine örnek.	24
Şekil 4.7 Tokenize edilmiş metin örneği.	24
Şekil 4.8 Stop kelimelerin çıkarılması.	25
Şekil 4.9 Vektörizasyon işleminin yapılması.	26
Şekil 4.10 Wordcloud yöntemi ile ilgililik veri setindeki kelimelerin gösterimi.	28
Şekil 4.11 İlgililik durumu için eğitilecek veri setinin dağılımı.	28
Şekil 4.12 Olumluluk veri setinin wordcloud yöntemi ile görselleştirilmesi.	29
Şekil 4.13 Olumluluk durumu için eğitilecek veri setinin dağılımı.	29
Şekil 4.14 Verilerin eğitim ve test olarak ikiye ayrılması.	31
Şekil 4.15 Bernoulli Naive Bayes model eğitimi.	32
Şekil 4.16 Oluşturulan modelin kaydedilmesi.	34
Şekil 4.17 Önceden kaydedilmiş modelin yüklenmesi.	34
Şekil 4.18 İlgililik modeli Bernoulli Naive Bayes ROC eğrisi.	35
Şekil 4.19 Olumluluk modeli Bernoulli Naive Bayes ROC eğrisi.	36
Şekil 4.20 İlgililik modeli Gaussian Naive Bayes ROC eğrisi.	36
Şekil 4.21 İlgililik modeli Multinomial Naive Bayes ROC eğrisi.	37
Şekil 4.22 İlgililik modeli Logistic Regression ROC eğrisi.	37
Şekil 4.23 İlgililik modeli KNN ROC eğrisi.	38
Şekil 4.24 İlgililik modeli Support Vector Classifier Cross Validation değerleri.	38
Şekil 4.25 Flask ile yazılmış servis kodu.	39
Şekil 4.26 HTML ile yazılmış ana sayfa arayüzü.	40
Şekil 4.27 Ana sayfada girilen verilerin modelden geçirilerek oluşturulmuş sonuçlarının gösterildiği sayfa.	42
Şekil 5.1 Yorum ve yıldız sayısı girişi yapılan ana sayfa arayüzü.	43
Şekil 5.2 Ana sayfada girilen verilerin modellerden geçirildikten sonra sonuçlarının gösterildiği arayüz.	43
Şekil 5.3 Ana sayfa arayüzü için ikinci bir örnek.	44
Şekil 5.4 Sonuç arayüzü için ikinci bir örnek.	44

Trendyol Yorum Sınıflandırma Projesi

ÖZET

Bu proje, Trendyol isimli e-ticaret platformu üzerinde satışı yapılan ürünler ile ilgili kullanıcı yorumlarının, doğal dil işleme yöntemleri ile değerlendirilerek, verilen yıldız sayısı ile yapılan yorum arasındaki uyumsuzlukların ve ürünle ilgili olmayan (kargo geç geldi, bedeni uymadı gibi) yorumların tespit edilmesini amaçlamaktadır. Projede çoğunlukla Python programlama dili kullanılmaktadır. Selenium kütüphanesi ile site içerisindeki yorumlar veri madenciliği yöntemleri ile elde edilmektedir. Veriler elde edildikten sonra Naive Bayes, KNN, Decision Tree gibi makine öğrenimi modelleri eğitilmektedir. Model oluşturulduktan sonra gerekli değerlendirmeler yapılarak en performanslı çalışan model seçilerek modelin Flask kütüphanesi ile servis haline getirilmesi ve bu servisin kullanılacağı bir web arayüzü geliştirilmektedir. Bu makalede proje geliştirilirken yapılan işlemler raporlanmaktadır.

Anahtar kelimeler: Doğal Dil İşleme, Makine Öğrenmesi, Trendyol Yorum Analizi, Çevrimiçi Alışveriş, Python

MACHINE LEARNING BASED REAL-TIME SMART ROOM SYSTEM

SUMMARY

This project aims to detect user comments about the products sold on the Trendyol e-commerce platform, by evaluating with natural language processing methods, the discrepancies between the number of stars given and the comments made, and the comments that are not related to the product (such as the cargo arrived late, the size did not fit). Python programming language is mostly used in the project. With the Selenium library, the comments on the site are obtained by data mining methods. After the data is obtained, machine learning models such as Naive Bayes, KNN, and Decision Tree are trained. After the model is created, necessary evaluations are made, and the model that works with the most performance is selected, and a web interface is developed for the model to be serviced with the Flask library and to use this service. This article reports the actions taken while developing the project.

Keywords: Natural Language Processing, Machine Learning, Trendyol Comment Analysis, Online Shopping, Python

1. GİRİŞ

Günümüzde çevrimiçi alışveriş, tüketicilerin sıklıkla kullandığı bir alışveriş türü haline gelmiştir. Çevrimiçi alışverişin tercih edilmesine neden olan birçok etken mevcuttur. Bu etkenlerden bazıları kolay erişilebilirlik, geniş ürün yelpazesi, fiyat karşılaştırmasının yapılabilmesi, diğer alıcıların değerlendirmelerini görüntüleyebilmek. Getirdiği faydalarla birlikte çevrimiçi alışverişin birtakım dezavantajları da bulunmakta. Bu dezavantajlardan bazıları fiziksel deneyimden yoksun kalmak, teslimat ve iade süreçlerinin uzun ve sancılı olabilmesi, satıcıya ve alışveriş yapılan platforma kişisel ve finansal verilerin verilmesi başlıca dezavantajlardır.

Trendyol ürünlerinin yorum sınıflandırması projesi, müşterilerin daha doğru ve güvenilir bilgilere erişmesine yardımcı olmak ve tüketici için çevrimiçi alışverişin dezavantajlarını azaltmak amacıyla ortaya çıkmıştır. Yöntem olarak ise çevrimiçi alışverişin avantajlarından olan diğer alıcıların değerlendirmelerinin görüntülenebilmesi özelliğinin güçlendirilmesi tercih edilmiştir. Birçok kullanıcı, satın almadan önce diğer müşterilerin deneyimlerine dayalı geri bildirimlere güvenir. Ancak, çoğu zaman müşteriler ürün yorumlarının gerçekten ürünle ilgili olmadığını görmekte ve bundan şikayetçi olmaktadır.

Bu projenin temel amacı, Trendyol'da bulunan ürün yorumlarını gerçekten ürünle ilgili olup olmadığını tespit edebilecek bir makine öğrenmesi modeli oluşturmak ve yapılan yorumların verilen yıldız sayısı ile orantılı olup olmadığını tespit edebilecek bir model oluşturmak ve bu modelleri kullanılabilir bir ürün haline getirmektir. Proje, yapay zeka ve doğal dil işleme yöntemlerini kullanarak, yorumların içeriğini analiz eder. Karmaşık algoritmalar, yorumları inceleyerek gerçekten ürünle ilgili olanları tespit etmek için çeşitli özellikler ve kalıplar arar. Örneğin, ürünle ilgili spesifik detayları içeren yorumlar, gerçek kullanıcı deneyimlerini yansıtmaya eğilimindedir. Diğer yandan, yalnızca genel ifadeler veya sahte içerik barındıran yorumlar, dikkate alınmaz.

1.1 Hipotez

Trendyol ürünlerinin yorum sınıflandırması projesi, ürünle ilgili yorumları diğer yorumlardan etkili bir şekilde ayırt edebilir. Bu hipotez, Trendyol platformunda bulunan ürün yorumlarının içeriğini inceleyen bir projenin, gerçekten ürünle ilgili olan yorumları diğer yorumlardan başarıyla ayırabileceğini öngörüyor. Proje, yapay zeka ve doğal dil işleme tekniklerini kullanarak yorumları analiz edecek ve ürünle ilgili olan yorumları diğerlerinden ayırt etmek için belirli özellikler ve kalıplar arayacaktır. Ayrıca kullanıcı değerlendirmesinde verilen yıldız sayısı ile yapılan yorum karşılaştırılarak yorumun gerçek yıldız seviyesi tespit edecektir.

2. LİTERATÜR TARAMASI

Havva Yılmaz, Semih Yumuşak, “Açık Kaynak Doğal Dil İşleme Kütüphaneleri” (2021), bu çalışmada doğal dil işleme alanı, yapay zeka ve dilbilim alt kategorileriyle birlikte incelenmektedir. Doğal dil işleme yöntemleri sürekli güncellenmekte ve yeni yöntemler geliştirilmektedir. Çalışmanın amacı, doğal dil işleme projeleri için uygun kütüphanelerin doğru ve hızlı bir şekilde seçilmesini sağlamaktır. Bu amaç doğrultusunda, popüler doğal dil işleme kütüphaneleri özetlenmektedir. Farklı yöntem ve kütüphaneler karşılaştırmalı olarak açıklanmaktadır.[1]

Kemal Oflazer, “Türkçe ve Doğal Dil İşleme” (2012), Bu makalede Türkçe’nin doğal dil işleme açısından ilginç olan özellikleri ve karşılaşılan sorun ve bulunan çözümlerin kuş bakışı bir taraması yapılmıştır. Çoğu zorluklar dilin karmaşık sözcük yapısından ve bu yapının sözdizim ve istatistiksel modellemeyle olan ilişkisinden kaynaklanmaktadır. Bu taramanın sonrasında da Türkçe doğal dil işleme için geliştirilmiş olan önemli kaynakların bir özeti verilmiştir.[2]

Seda Tuzcu, “Çevrimiçi Kullanıcı Yorumlarının Duygu Analizi ile Sınıflandırılması” (2020), Duygu analizi, ilgilenilen metin kaynağının kutupsallığını sınıflandırmak için bir yaklaşımdır. Günümüzde internet kullanım oranının yüksek olması nedeniyle hacmi giderek artan çevrimiçi kullanıcı yorumları, erişilebilirlik ve çeşitlilik açısından duygu analizi çalışmaları için önemli bir veri kaynağı haline gelmektedir. Bu çalışmada öncelikle Python programlama dili kullanılarak duygu analizi için çevrimiçi bir kitapçının çevrimiçi kullanıcı incelemelerine Çok Katmanlı Algılayıcı (MLP) algoritması uygulanmıştır. Daha sonra RapidMiner veri bilimi yazılımı kullanılarak aynı veri seti üzerinde Naïve Bayes (NB), Destek Vektör Makineleri (SVM) ve Lojistik Regresyon (LR) algoritmaları uygulanmıştır. Algoritmaların incelemeleri sınıflandırmadaki başarıları karşılaştırıldı ve Multi-Layer Perceptron bu veri setinde en iyi sonuçları gösteren algoritma oldu.[3]

Gülşen Eryiğit, “ITU Turkish NLP Web Service” (2014), İstanbul Teknik Üniversitesi doğal dil işleme grubu tarafından geliştirilen “İTÜ Türkçe NLP Web Service” adlı bir doğal dil işleme (NLP) platformu sunuyoruz. Platform (tools.nlp.itu.edu.tr adresinde mevcuttur) bir SaaS (Software as a Service) olarak çalışır ve araştırmacılara ve öğrencilere birçok katmanda son teknoloji NLP araçları sağlar: ön işleme, morfoloji, sözdizimi ve varlık tanıma. Kullanıcılar platformla üç kanal üzerinden iletişim kurabilirler: 1. kullanıcı dostu bir web arayüzü aracılığıyla, 2. dosya yüklemeleri yoluyla ve 3. daha üst düzey uygulamalar oluşturmak için kendi kodlarında sağlanan Web API' lerini kullanarak.[4]

Burhan Bilen, Fahrettin Horasan, “LSTM Network based Sentiment Analysis for Customer Reviews” (2022), Çalışmada, tekil etiket-çoğul sınıf yaklaşımıyla ikili sınıflandırma yapılmıştır. Bir LSTM ağı ve birkaç makine öğrenimi modeli kullanılarak testler gerçekleştirilmiştir. Türkçe veri seti ve Stanford Büyük Film İncelemeleri veri seti kullanılarak çalışma yürütülmüştür. Veri setindeki gürültü nedeniyle metinlerin normalleştirilmesi ve temizlenmesi için Zemberek NLP Kütüphanesi Türk Dilleri ve Düzenli Anlatım teknikleri kullanılmıştır. Ardından, veriler vektör dizilerine dönüştürülmüştür. Ön işleme sürecinin, Türkiye Müşteri Yorumları veri setinde model performansında %2'lik bir artış sağladığı belirtilmiştir. Model, LSTM ağı kullanılarak oluşturulmuştur ve makine öğrenimi tekniklerinden daha iyi performans göstermiştir. Türkiye veri setinde %90,59, IMDB veri setinde ise %89.02 doğruluk elde edilmiştir. Bu çalışma, duygu analizi alanındaki ilerlemeleri ve Türkçe veri setleri üzerinde yapılan sınıflandırma çalışmalarını özetlemektedir.[5]

3. KULLANILAN TEKNOLOJİLER

Doğal Dil İşleme (Natural Language Processing, NLP): Doğal Dil İşleme teknikleri, metin verilerini analiz etmek ve anlamak için kullanılır. Trendyol yorumlarındaki metinleri işlemek ve içerdikleri bilgileri anlamak için NLP yöntemleri kullanılır. Örneğin, metin sınıflandırması, duygu analizi ve varlık tanıma gibi NLP yöntemleri, yorumların ürünle ilgili olup olmadığını belirlemek için kullanılır.

Makine Öğrenmesi (Machine Learning): Makine öğrenmesi algoritmaları, belirli bir amaca yönelik olarak eğitilen modeller oluşturmak için kullanılabilir. Trendyol yorumlarının analizi için makine öğrenmesi algoritmaları kullanılarak, gerçekten ürünle ilgili yorumları tespit etmek için bir sınıflandırma modeli geliştirilebilir. Bu model, eğitim verileri üzerinde öğrenerek, yeni yorumların ürünle ilgili olup olmadığını tahmin edebilir.

Veri Madenciliği (Data Mining): Veri madenciliği teknikleri, büyük veri setlerinden anlamlı bilgiler çıkarmak için kullanılır. Trendyol yorumları üzerinde veri madenciliği yöntemleri uygulanarak, yorumlardaki kalıpları ve özellikleri belirlemek mümkündür. Bu yöntemler, ürünle ilgili spesifik terimlerin varlığını, kullanıcı deneyimini yansıtan ifadeleri ve diğer önemli özellikleri belirlemeye yardımcı olabilir.

3.1 Python

Python, genel amaçlı bir programlama dilidir. 1991 yılında Guido van Rossum tarafından geliştirilen Python, kolay okunabilir ve anlaşılır bir sözdizimine sahip olmasıyla bilinir. Özellikle açık kaynak kodlu yazılım geliştirme, veri analizi, yapay zeka, web geliştirme ve bilimsel hesaplama gibi birçok alanda popüler bir tercih haline gelmiştir.

Python, açık kaynak kodlu bir dildir, yani kullanıcılar tarafından ücretsiz olarak indirilebilir, kullanılabilir ve değiştirilebilir. Ayrıca Python, geniş bir topluluğa sahiptir ve sürekli olarak gelişmektedir. Çok sayıda kaynak, dokümantasyon ve eğitim materyali bulunmaktadır, bu da öğrenme sürecini kolaylaştırır. Python, birçok farklı alanda kullanılan güçlü ve çok yönlü bir programlama dilidir.

3.1.1 Pandas ve Numpy

Bu projede Python dilinin birçok kütüphanesi kullanılmaktadır. Bunlardan Pandas ve NumPy veri bilimi için kullanılan kütüphanelerdir. Pandas, Python için açık kaynaklı bir veri analizi kütüphanesidir. Özellikle büyük ve karmaşık veri setlerinin işlenmesinde kolaylık sağlar. Veri işleme, temizleme, analiz ve modelleme için birçok fonksiyon içermektedir. Bu kütüphane projenin kodlama sürecinde oldukça fazla kullanılmaktadır.

3.1.2 Natural Language Toolkit (NLTK)

Python dilinde doğal dil işleme (NLP) uygulamaları geliştirmek için kullanılan bir kütüphanedir. NLTK, metin verilerini işlemek, analiz etmek ve dilbilimsel araştırmalar yapmak için bir dizi araç ve kaynak sunar. Python dilinde doğal dil işleme projeleri geliştirmek isteyen araştırmacılar, öğrenciler ve yazılım geliştiriciler için güçlü ve kullanımı kolay bir araçtır. Kapsamlı dokümantasyonu ve topluluk desteği ile NLTK, NLP alanında yaygın olarak kullanılan bir kütüphanedir.

3.1.3 Pickle

Python dilindeki "pickle" kütüphanesi, nesneleri serileştirmek (pickle etmek) ve serileştirilmiş nesneleri geri yüklemek için kullanılan bir modüldür. Pickle, Python nesnelerini diskte veya ağ üzerindeki bir dosyaya yazmak ve daha sonra geri yüklemek için kullanılır. Bu işlem, nesnelerin yapılarını ve durumlarını korur ve ileride tekrar kullanılabilir hale getirir. Bu sayede oluşturduğumuz model ve vektörleri ileride tekrar tekrar oluşturmadan kullanmamız mümkün olmaktadır.

3.1.4 Flask

Flask, Python programlama dilinde web uygulamaları geliştirmek için kullanılan hafif bir web framework'üdür. Flask, minimal ve basit bir tasarıma sahiptir ve esneklik sunar. Flask, WSGI uyumlu web sunucuları üzerinde çalışabilir ve HTTP isteklerini işleyerek dinamik web sayfaları oluşturmanıza olanak tanır. Flask, basit ve küçük ölçekli web projelerinden daha karmaşık uygulamalara kadar geniş bir yelpazede kullanılabilir. Hızlı prototipleme için idealdir ve RESTful API'ler, web tabanlı uygulamalar, mikro hizmetler ve daha fazlası için tercih edilen bir seçenektir. Flask'ın kolay kullanımı ve öğrenilmesi, Python geliştiricileri arasında popüler hale getirmiştir.

3.2 Google Colab (Colaboratory)

Google Colab, Google tarafından sunulan ücretsiz bir Jupyter notebook hizmetidir. Google Colab, tarayıcınızda çalışan bulut tabanlı bir Python geliştirme ortamı sunar. Bu platformda, Python kodu yazabilir, çalıştırabilir ve sonuçları hızlı bir şekilde görebilirsiniz. Google Colab, GPU ve TPU gibi yüksek performanslı işlem birimlerini kullanma imkanını ücretsiz olarak sağlar. Bu imkan, makine öğrenmesi ve derin öğrenme gibi hesaplama yoğun işlemleri hızlandırmak için önemlidir. Bu projede yapılan kodlama işlemleri de Google Colab üzerinde yapılmaktadır.

4. METODOLOJİ

Bu projenin metodolojisi; İlk adım, Trendyol'dan ürün yorumlarını veri madenciliği yöntemleri ile toplamaktır. İkinci adım, toplanan verilerin çeşitli yöntemlerle ön işleme tabii tutulması. Üçüncü adım, işlenen yorum verilerinin ilgililik durumuna göre sınıflandırılması. Dördüncü adım, sınıflandırılan verilerin model ile eğitilmesi ve doğruluk değerlerinin kontrol edilmesi. Beşinci adım, oluşturulan modelin kullanılabilir bir ürün haline getirilmesi. Son adım, sonuçların analiz edilip raporlanması. Bu adımların her biri, projenin başarılı bir şekilde tamamlanması için önemlidir.

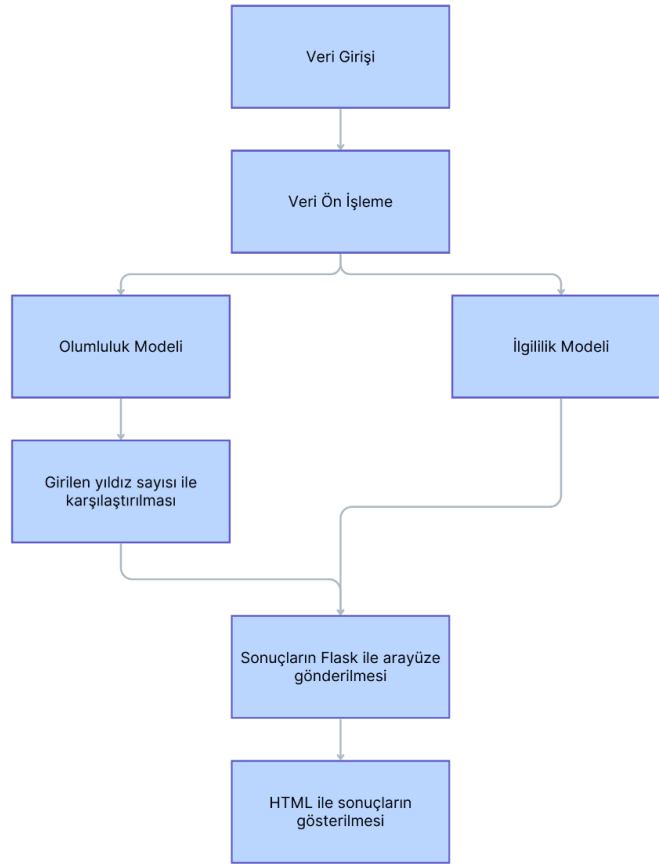
Görevler\Haftalar		Şubat		Mart				Nisan				Mayıs				Haziran			
		3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Hazırlık	Projenin belirlenmesi ve araştırılması																		
	Proje teknolojilerinin araştırılması																		
	Veri toplama yönteminin belirlenmesi																		
	Veri toplama işleminin yapılacağı örnek ürünlerin seçilmesi																		
	Projenin belirlendiğine dair rapor teslim edilmesi																		
	Çalışma programının oluşturulması																		
Veri İşleme	Verilerin toplanması																		
	Verilerin saklanacağı formatın belirlenmesi																		
	Verilerin sınıflandırma kriterlerinin belirlenmesi																		
	Oluşturulan veri setinin sınıflandırılması																		
Model Eğitimi	Proje için uygun makine öğrenmesi modellerinin incelenmesi																		
	Ara rapor teslimi																		
	Model eğitimi için örnek projelerin incelenmesi																		
	Modellerin eğitilip test edilmesi																		
Uygulama	Optimum modelin seçilip geliştirilmesi																		
	Projenin nasıl bir ürün haline getirileceğinin belirlenmesi																		
	Projenin ürünleştirilmesinde kullanılacak teknolojinin belirlenmesi																		
	Ürünün geliştirilmesi																		
Raporlama	Proje hatalarının tespit edilip düzeltilmesi																		
	Bitirme projesi raporunun yazılması																		
	Raporun teslimi																		
	Proje sunumu																		

Şekil 4.1 Çalışma programı için Gantt şeması.

4.1 Proje Geliştirme Süreci

Bu projede yaşanabilecek birçok versiyon uyumsuzluğu, kendi bilgisayar ortamı yerine Google Colab kullanılarak giderilmiştir. Bu durumda hızlı bir şekilde çözüm üretme kabiliyetine sahip olmak, proje sürecinin ilerleyişini olumlu olarak etkilemiştir.

4.2 Proje Taslağının Oluşturulması



Şekil 4.2 Proje algoritmasının çalışma şeması.

Projenin ilk aşamasında, oluşturulan arayüz sayfasından bir adet yorum ve bu yoruma verilen yıldız sayısının girişi yapılır. Girilen veriler modelin anlayabileceği şekilde ön işlemeden geçirilir. Ön işlemeden alınan veriler, önceden eğitilmiş modele gerçek zamanlı olarak gönderilir. Modelden çıkan sonuçlar, Flask kütüphanesi ile oluşturulan yerel sunucudan arayüze gönderilir ve HTML ile oluşturulan sonuçların gösterimi yapılır.

4.3 Verinin toplanması ve saklanması

Projenin ana amacı olan yorumların analizinin yapılabilmesi için öncelikle elimizde modelimizi eğitebileceğimiz verilerin olması gerekir. Bu veriler Trendyol sitesinin kendisinde bulunan cep telefonu, kazak, pantolon, ayakkabı gibi popüler ürünlere yapılmış yorumlar Python'un bir web kazıma kütüphanesi olan Selenium ile elde edildi. [8]

Selenium ile toplanan veriler Şekil ... de gösterildiği gibi aralarında bir “\n” (tab) boşluk bırakılacak şekilde “**num**, **text**, **star**” isimli sütunlar halinde toplanmıştır. Burada “**num**” veri numarasını, “**text**” yorum metnini, “**star**” ise yoruma verilen yıldız sayısını belirtmektedir.

```
num text star
1 Kesinlikle harika bir ürün camın uzun süre temiz kalmasını sağlıyor sürekli stokluyorum 5
2 Stok yapılacak harika bi ürün. 5
3 Kargolama güzeldi teşekkür ediyorum ilk kullanımda memnun kaldım stoklanması gereken bir ürün 5
4 süper bi ürün uğraşmaya gerek yok anında pırıl pırıl camlar,aynalar ben indimden aldım. indirim zamanlarında stoklamalık ürün 5
5 Stok yaparak aldım temizliğini seviyorum 5
6 Gerçekten çok memnunuz stok yapmak için aldım sizde alın aldırın 5
7 severek kullanıyorum arkadaşlarıma da aldım. ayrıca hızlı kargo ve iyi paketleme için de teşekkürler 5
8 Ay çok güzel beğeniyorum,özellikle aynaları pırıl pırıl yapıyor. 4
9 Araba için aldık. Memnunuz 5
10 İndirimde stoklanacak ürünler arasında. Elinizin altında mutlaka olmalı 5
11 cidden süper ürün evde küçük bebeği olan anneler bilir aynalar hep el izi bu ürün süper bişey alın mutlaka deneyin 5
12 Çokkkk harikaaaa stoklayın 5
13 cam ve aynalarda kullanıyorum..pırıl pırıl yapıyor..kokusu da çok güzel, odayı mis gibi kokutuyor..her eve lazım.. 5
14 Cam silmek artık daha kolay. İndirim zamanlarında stoklanacak bir ürün. 5
15 Sorunsuz paketleme denedim gayet güzel. 5
16 kutu kutu stok yaptığım ürün kendileri bayılıyor 5
17 Hemen stok yapacağım bayıldım 5
```

Şekil 4.3 Verilerin saklanma şekli.

4.3.1 Selenium

Selenium, web tarayıcı otomasyonu için kullanılan bir araçtır. Web scraping ise web sayfalarından veri çekme veya bilgi kazıma işlemidir. Bu nedenle, Selenium'i web scraping için kullanabilirsiniz.

Selenium, tarayıcıyı otomatik olarak kontrol etmenizi sağlar. Bu, bir web tarayıcısını başlatmanıza, web sayfalarına erişmenize, sayfalarda gezinmenize, kullanıcı etkileşimlerini taklit etmenize ve sayfa öğelerini bulmanıza olanak tanır. Web scraping için, Selenium'i kullanarak web sayfalarını otomatik olarak ziyaret edebilir, belirli öğeleri (metin, resim, tablo vb.) bulabilir ve bu öğelerden veri çekebilirsiniz.

Selenium, dinamik web sayfalarıyla da etkileşim kurabilme yeteneğine sahiptir. Dinamik web sayfaları, JavaScript veya AJAX gibi teknolojiler kullanarak sayfa içeriğini yükler veya günceller. Bu nedenle, statik verileri çekmekten daha karmaşık

bir işlem gerektirebilir. Selenium'in bu durumda tarayıcıyı kontrol ederek sayfanın yüklenmesini beklemesi ve ardından gerekli verileri alması önemlidir.

Selenium ile web scraping yaparken, web sayfalarına erişmek, form doldurmak, düğmelere tıklamak, sayfa kaydırmak, tablo verilerini çekmek gibi birçok işlemi gerçekleştirebilirsiniz. Bu sayede, istediğiniz verileri toplayabilir ve analiz veya başka işlemler için kullanabilirsiniz.

Web scraping işlemlerini gerçekleştirirken, hedef web sitesinin kullanım şartlarına ve izinlere uymanız önemlidir. Bazı web siteleri, web scraping'i yasaklayabilir veya belirli kısıtlamalar getirebilir. Bu nedenle, web scraping yaparken hedef web sitesinin **robots.txt** dosyasını kontrol etmek ve gerektiğinde izinleri almak önemlidir. Ayrıca, etik davranış kurallarına uymak ve web scraping işlemlerini aşırı yüklenmeye yol açacak şekilde düzenlememek önemlidir.

4.4 Veri Ön İşleme

Veri ön işleme, doğal dil işleme (NLP) veya makine öğrenmesi gibi veriye dayalı analitik çalışmalarda kullanılan bir aşamadır. Veri ön işleme, ham veri kümesini temizleme, düzenleme ve dönüştürme sürecini ifade eder. Bu aşama, veri setinin daha işlenebilir, analiz edilebilir ve modele uygun hale getirilmesini sağlar. Bu projede dört adet veri ön işleme yöntemi tercih edilmiştir.[6]

Metin Temizleme: Gereksiz karakterlerin ve noktalama işaretlerinin kaldırılması. Özel sembollerin temizlenmesi. Sayıların çıkarılması veya yerine belirli bir sembolün atanması. Büyük/küçük harf dönüşümü yapılması.

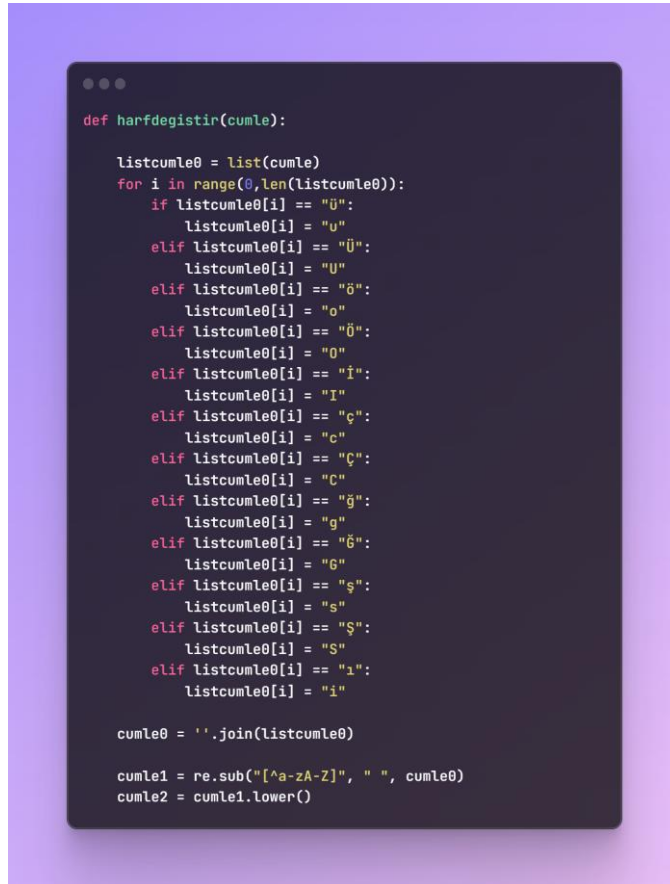
Tokenization (Tokenize Etme): Metni anlamlı birimlere, yani kelimelere veya alt cümlelere (tokenlara) ayırma. Kelimelerin veya cümlelerin ayraçlarla ayrılması.

Stop Kelime Kaldırma: Yaygın ve anlamsız kelimelerin (stop kelimelerin) çıkarılması. Stop kelimeleri genellikle analiz için önemsiz kabul edilir, örnek olarak "ve", "veya", "ama" gibi kelimeler.

Vectorization (Vektörleştirme): Metni sayısal verilere dönüştürmek için vektörleştirme yöntemleri kullanılır. Örneğin, kelime seviyesindeki vektörleştirme yöntemleri (TF-IDF, Count Vectors) veya kelime gömme modelleri (Word2Vec, GloVe) kullanılabilir.

4.4.1 Metin Temizleme

Bu veri işleme yönteminde öncelikle Türkçe karakterlerin İngilizce karakterlere çevrilmesi işlemi yapılmaktadır. Bunu yapacak Python fonksiyonu Şekil 4.4’de verildiği gibidir. Bu kod bloğunda “ü, ö, ç, İ, ğ, ş” gibi İngilizcede bulunmayan karakterler İngilizcede bulunan yakın karakterlere çevrilmektedir. Ardından **re.sub** fonksiyonu ile a’den z’ye kadar olan İngilizce harfler dışındaki bütün karakterlerin yerine “ ” (boşluk) koyarak gereksiz karakterlerden kurtulmaktadır. Gereksiz karakterlerden kurtulunca büyük harfler küçük harfe dönüştürülmektedir.



```
def harfdegistir(cumle):  
  
    listcumle0 = list(cumle)  
    for i in range(0, len(listcumle0)):  
        if listcumle0[i] == "ü":  
            listcumle0[i] = "u"  
        elif listcumle0[i] == "ö":  
            listcumle0[i] = "o"  
        elif listcumle0[i] == "ç":  
            listcumle0[i] = "c"  
        elif listcumle0[i] == "İ":  
            listcumle0[i] = "I"  
        elif listcumle0[i] == "ğ":  
            listcumle0[i] = "g"  
        elif listcumle0[i] == "ş":  
            listcumle0[i] = "s"  
        elif listcumle0[i] == "ı":  
            listcumle0[i] = "i"  
  
    cumle0 = ''.join(listcumle0)  
  
    cumle1 = re.sub("[^a-zA-Z]", " ", cumle0)  
    cumle2 = cumle1.lower()
```

Şekil 4.4 Veri ön işleme yapan fonksiyon.

4.4.2 Tokenize Etme

Kelimeleri tokenize etmek, metni anlamlı birimlere, yani kelimelere veya alt cümlelere (tokenlara) ayırmak anlamına gelir. Kelime tokenize işlemi, doğal dil işleme (NLP) uygulamalarında önemli bir adımdır ve metin verilerinin işlenmesi için birçok fayda sağlar.

Dilbilgisi Analizi: Kelime tokenize, metindeki kelime yapısını ve cümle yapısını analiz etmek için bir temel sağlar. Bu şekilde, cümledeki kelime sıralaması, dilbilgisi yapıları ve anlam ilişkileri gibi dilbilgisi analizlerini gerçekleştirebilirsiniz.


Bilgi Çıkarma: Kelimeleri tokenize etmek, metinden anlamlı bilgileri çıkarmayı kolaylaştırır. Tokenlara ayrılmış kelimeleri kullanarak metindeki anahtar kelimeleri belirleyebilir, konuları tanımlayabilir veya metindeki önemli ifadeleri algılayabilirsiniz.

Kelime Frekansı Hesaplama: Tokenize edilmiş kelimeleri kullanarak, metindeki kelime frekansını hesaplayabilirsiniz. Bu, metinde hangi kelimelerin ne sıklıkta kullanıldığını belirlemek ve kelime dağılımını analiz etmek için önemlidir. Kelime frekansı, metnin içeriği hakkında genel bir fikir edinmek için kullanılabilir.

Dil Modeli Oluşturma: Tokenize edilmiş kelimeler, dil modeli oluşturma için kullanılabilir. Dil modeli, dildeki kelime ve cümle yapısını temsil eden istatistiksel bir modeldir. Tokenize edilmiş kelimeleri kullanarak dil modeli eğitebilir ve ardından metindeki dilbilgisi yapılarını veya yeni cümleleri oluşturabilirsiniz.

Makine Öğrenmesi Uygulamaları: Tokenize edilmiş kelimeler, makine öğrenmesi modellerine giriş olarak kullanılabilir. Kelimeleri sayısal vektörlere dönüştürerek, metni makine öğrenmesi algoritmalarına besleyebilir ve metin tabanlı sınıflandırma, duygu analizi veya metin üretimi gibi NLP görevlerini gerçekleştirebilirsiniz.

Bu nedenlerle, kelimeleri tokenize etmek, metin verilerinin daha işlenebilir, analiz edilebilir ve makine öğrenmesi uygulamalarıyla kullanılabilir hale gelmesini sağlar. Ayrıca, tokenizasyon, dilbilgisi analizi, bilgi çıkarma ve dil modeli oluşturma gibi birçok NLP görevinin temelini oluşturur.

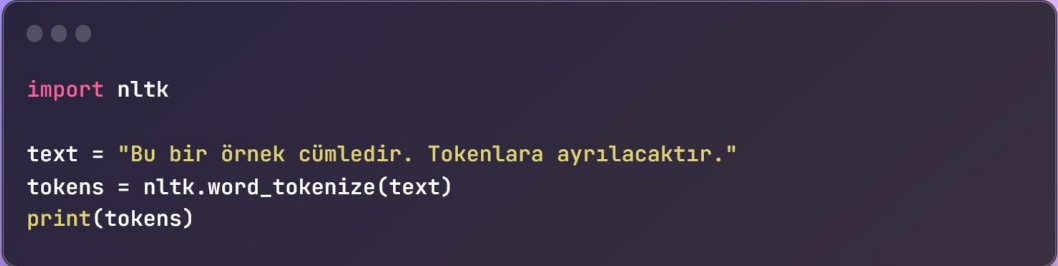


```
cumle2 = nltk.word_tokenize(cumle2)
```

Şekil 4.5 Fonksiyonun içinde tokenize işlemini yapan kod.

Projede NLTK kütüphanesinin **word_tokenize** fonksiyonunu Şekil 4.5'deki gibi kullanarak **harfdegistir** adlı fonksiyonda önceden temizlenen metinler tokenize edilmektedir.

Metni **nltk.word_tokenize()** fonksiyonuna girdi olarak verildiğinde, fonksiyon metni küçük parçalara bölerek kelimeleri veya alt cümleleri (tokenları) döndürür. Bu parçalama işlemi, metin içerisindeki boşlukları ve noktalama işaretlerini kullanarak gerçekleştirilir.



```
import nltk

text = "Bu bir örnek cümledir. Tokenlara ayrılacaktır."
tokens = nltk.word_tokenize(text)
print(tokens)
```

Şekil 4.6 Metinlerin tokenize edilmesine örnek.

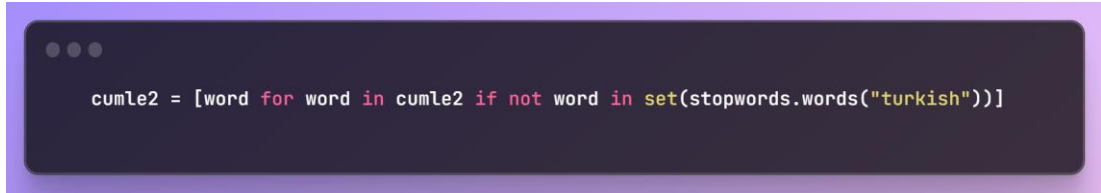


```
['Bu', 'bir', 'örnek', 'cümledir', '.', 'Tokenlara', 'ayrılacaktır', '.']
```

Şekil 4.7 Tokenize edilmiş metin örneği.

4.4.3 Stop Kelime Kaldırma

Bu yöntemde, metin analizinde anlamsal olarak az öneme sahip olan ve genellikle dilbilgisi yapılarını ifade eden kelimeler, yani "stop words" çıkarılır. Stop kelimeler, genellikle bağlaçlar, zamirler, edatlar ve sık kullanılan kelimeler gibi dilbilgisi yapılarını ifade eden kelimelerdir. Örnek olarak, "bir", "ve", "ama", "için", "oldu" gibi kelimeler stop kelimeler arasında yer alır. Bu kelimeler metin analizinde pek fazla anlam taşımazlar ve daha çok gereksiz gürültü olarak kabul edilirler.



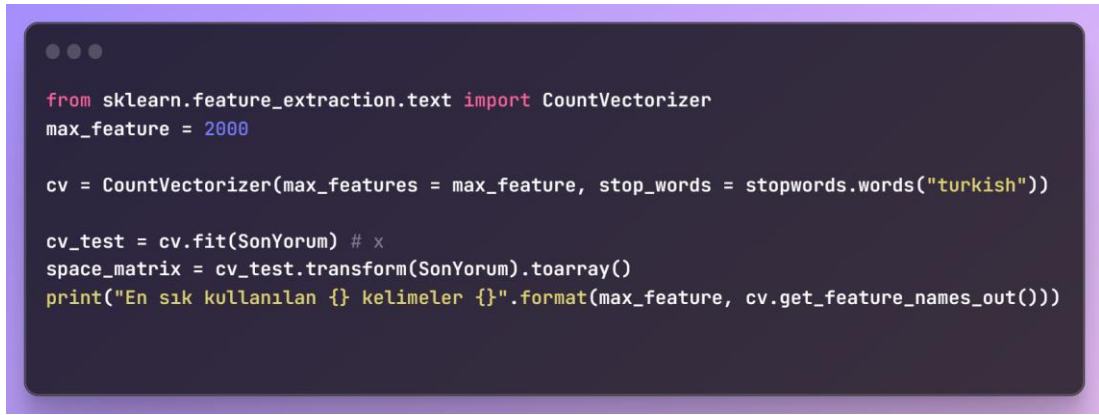
```
cumle2 = [word for word in cumle2 if not word in set(stopwords.words("turkish"))]
```

Şekil 4.8 Stop kelimelerin çıkarılması.

Şekil 4.8'deki kod cümleler içerisindeki stop kelimeleri çıkararak, yalnızca anlamlı kelimeleri içeren bir liste elde etmeyi sağlar. Böylece, stop kelimelerin metin analizi veya diğer doğal dil işleme uygulamalarında olumsuz etkileri azaltabilir ve önemli kelimeler üzerinde daha doğru işlemler yapmamız sağlamaktadır.

4.4.4 Vektörizasyon

Vektörizasyon, doğal dil işleme (NLP) uygulamalarında metin verilerini sayısal vektörlerle temsil etme işlemidir. Metin verileri, makine öğrenmesi algoritmaları veya diğer istatistiksel işlemlerle kullanılabilecek şekilde sayısal formata dönüştürülür. Metin verileri, insanlar arasında anlamlı olan dilbilgisi yapılarından oluşurken, makine öğrenmesi algoritmaları genellikle sayısal verilerle çalışır. Bu nedenle, metin verilerini sayısal vektörlerle temsil etmek, metin tabanlı NLP uygulamalarında yaygın bir gerekliliktir.



```

from sklearn.feature_extraction.text import CountVectorizer
max_feature = 2000

cv = CountVectorizer(max_features = max_feature, stop_words = stopwords.words("turkish"))

cv_test = cv.fit(SonYorum) # x
space_matrix = cv_test.transform(SonYorum).toarray()
print("En sık kullanılan {} kelimeler {}".format(max_feature, cv.get_feature_names_out()))

```

Şekil 4.9 Vektörizasyon işleminin yapılması.

Şekil 4.9'daki kod bloğu aşağıda belirtilen işlemleri sırasıyla gerçekleştirerek vektörizasyon işlemini gerçekleştirir.

from sklearn.feature_extraction.text import CountVectorizer: CountVectorizer sınıfını scikit-learn kütüphanesinden import eder.

max_feature = 2000: En fazla kullanılacak özellik sayısını belirler. Bu durumda, en fazla 2000 özellik kullanılacaktır.

cv=CountVectorizer(max_features=max_feature,stop_words=stopwords.words("turkish")): CountVectorizer sınıfının bir örneği oluşturulur. **max_features** parametresi, belirli bir sayıda en sık kullanılan özelliği seçmek için kullanılır. **stop_words** parametresi ise Türkçe durak kelimelerinin çıkarılmasını sağlayan bir liste kullanır.

cv_test = cv.fit(SonYorum): CountVectorizer nesnesi, veri seti **SonYorum** üzerinde uyumlaştırılır ve özelliklerin öğrenilmesini sağlar. Bu adımda, özelliklerin frekansları ve sıralamaları belirlenir.

space_matrix = cv_test.transform(SonYorum).toarray(): **SonYorum** veri setindeki metinler, **CountVectorizer** tarafından öğrenilen özelliklere göre sayısal vektörlere dönüştürülür. **transform** fonksiyonu, metin verilerini vektörler haline getirir. **toarray()** fonksiyonu, sonucu bir NumPy dizisine dönüştürür.

print("En sık kullanılan {} kelimeler {}".format(max_feature, cv.get_feature_names_out())): En sık kullanılan **max_feature** sayısındaki kelime özelliklerini ekrana basar. **get_feature_names_out()** fonksiyonu, öğrenilen kelime özelliklerinin listesini döndürür.

4.5 Veri Görselleştirme

Veri görselleştirme, verilerin grafikler, tablolar veya diğer görsel öğeler aracılığıyla sunulması işlemidir. Veri görselleştirme, aşağıdaki amaçlar için kullanılır:

Verilerin daha anlaşılır hale getirilmesi: Veri tabloları veya sayısal veriler genellikle karmaşık olabilir. Görsel grafikler kullanarak verileri daha anlaşılır bir şekilde sunmak, verileri daha kolay okunur ve anlaşılır hale getirebilir. Grafikler ve görseller, verilerin trendlerini, dağılımlarını, ilişkilerini veya desenlerini daha net bir şekilde gösterir.

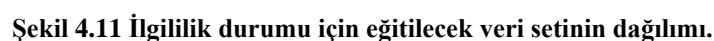
Trendleri ve ilişkileri keşfetme: Görsel grafikler, verilerdeki trendleri, ilişkileri ve desenleri daha hızlı ve daha kolay bir şekilde keşfetmeyi sağlar. Örneğin, çizgi grafikleri, zaman serilerindeki değişimleri gösterirken, scatter grafikleri iki değişken arasındaki ilişkiyi görselleştirebilir.

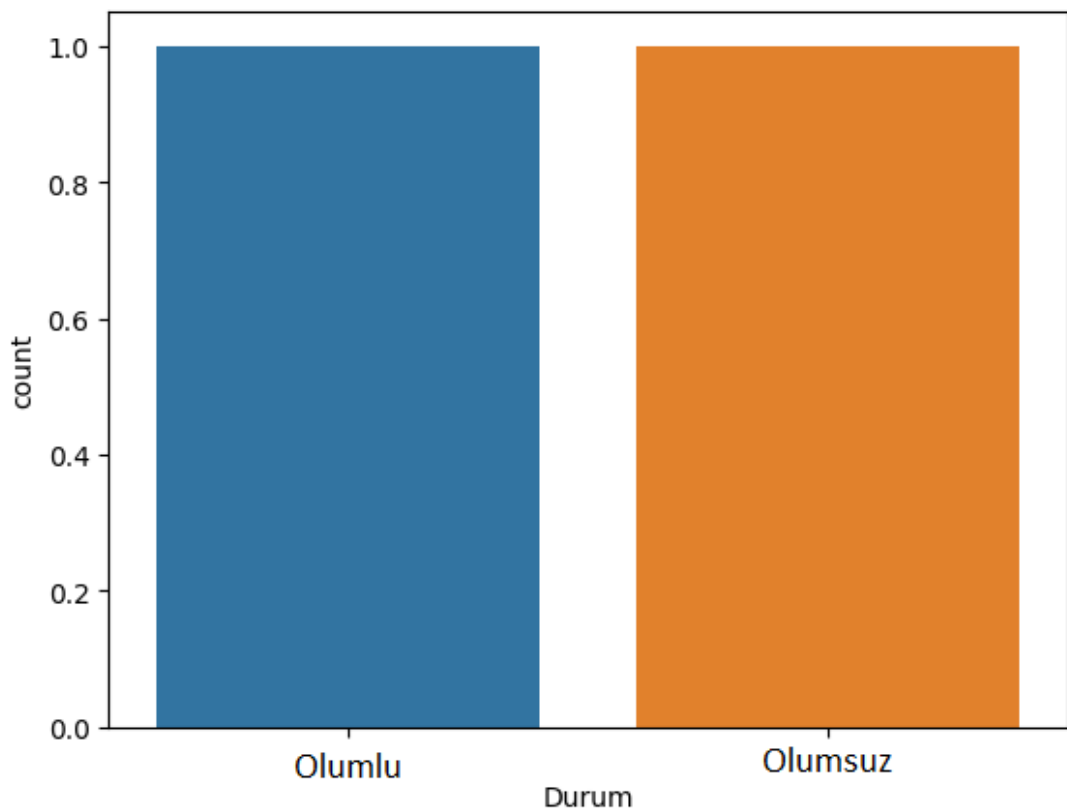
Bilgilerin etkili bir şekilde iletilmesi: Görsel öğeler, insan beyninin görsel algısına daha uygun olduğu için bilgileri etkili bir şekilde iletmek için kullanılır. Grafikler, tablolar veya infografikler, karmaşık veri kümelerini özetleyerek hızlı bir şekilde anlamamızı sağlar.

Karar verme süreçlerinin desteklenmesi: Veri görselleştirmesi, karar verme süreçlerinde önemli bir rol oynar. Grafikler ve görseller, verilere dayalı analizler yaparken daha doğru ve bilinçli kararlar almayı destekler. Verilerin görsel olarak sunulması, trendleri, desenleri veya aykırı değerleri daha kolay tespit etmeyi ve kararlarımızı buna göre şekillendirmeyi sağlar.

Hikaye anlatımı: Verileri görselleştirmek, verileri bir hikaye anlatma şekline dönüştürmeyi sağlar. Grafikler, görseller ve infografikler, verileri bir bağlam içinde anlatarak, bir hikayeyi anlatmaya yardımcı olur.

Veri görselleştirme, verilerin daha anlaşılır hale getirilmesi, trendlerin ve ilişkilerin keşfedilmesi, bilgilerin etkili bir şekilde iletilmesi ve karar verme süreçlerinin desteklenmesi için önemli bir araçtır. Görsel grafikler ve görseller, verilerin anlaşılması ve analiz edilmesi sürecini kolaylaştırır ve verilerden anlamlı içgörüler elde etmemizi sağlar.





4.6 Makine Öğrenmesi

Makine öğrenmesi, bilgisayar sistemlerinin veri üzerinde öğrenme yapabilmesi için kullanılan bir yapay zeka dalıdır. Makine öğrenmesi, algoritmaların verilerden örüntüleri çıkarabilmesini ve gelecekteki verilere dayanarak tahminler yapabilmesini sağlar.

Makine öğrenmesi modelleri, makine öğrenmesi algoritmalarının uygulandığı yapısal ve matematiksel modellerdir. Bu modeller, veri setlerinin içerdiği özellikleri kullanarak, girdi verilerinden çıktıları tahmin etmek veya sınıflandırmak için kullanılır. Makine öğrenmesi modelleri genellikle aşağıdaki kategorilere ayrılır:

Denetimli Öğrenme Modelleri: Bu modeller, etiketlenmiş veri setleri üzerinde eğitilir. Her bir örneğin giriş verileri ve hedef çıktıları (etiketleri) bilinir. Denetimli öğrenme modelleri, girdi verilerinden doğru çıktıları tahmin etmek için kullanılır. Örnekler arasında karar ağaçları, destek vektör makineleri (SVM), doğrusal regresyon ve sinir ağları bulunur.

Denetimsiz Öğrenme Modelleri: Bu modeller, etiketlenmemiş veri setleri üzerinde eğitilir. Denetimsiz öğrenme modelleri, veri setindeki gizli yapıları veya desenleri keşfetmek için kullanılır. Örnekler arasında kümeleme (clustering) algoritmaları ve boyut indirgeme (dimensionality reduction) yöntemleri bulunur.

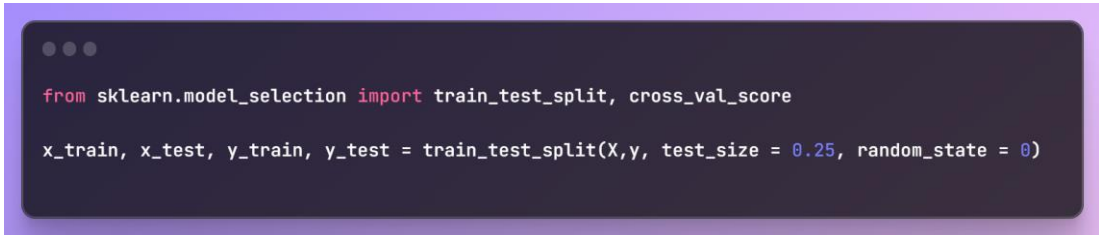
Yarı Denetimli Öğrenme Modelleri: Bu modeller hem etiketlenmiş hem de etiketlenmemiş veri setlerini kullanır. Yarı denetimli öğrenme, etiketli verilerin sınıflandırma veya tahmin performansını artırmak için etiketlenmemiş verilerden yararlanır.

Pekiştirmeli Öğrenme Modelleri: Bu modeller, bir ajanın bir çevreyle etkileşimde bulunarak ödül veya ceza sinyallerine dayanarak en iyi eylemleri öğrenmesini sağlar. Pekiştirmeli öğrenme modelleri, belirli bir hedefi optimize etmek için deney-yanılma sürecini kullanır.

Bu sadece bazı temel makine öğrenmesi modeli kategorileridir ve her kategori altında birçok farklı algoritma ve model bulunabilir. Makine öğrenmesi modelleri, veri analizi, sınıflandırma, regresyon, kümeleme, öneri sistemleri, doğal dil işleme ve görüntü işleme gibi çeşitli uygulamalarda kullanılır.

4.6.1 Modelin Oluşturulması

Model oluşturma işlemi için kullanım kolaylığı ve stabilitesi sayesinde **scikit-learn** kütüphanesi çoğu projede tercih edilmektedir. **scikit-learn**, kullanıcıların veri analizi, model eğitimi ve tahmin yapma gibi makine öğrenmesi görevlerini kolaylaştırmak için bir dizi algoritma ve araç sağlar. **scikit-learn**, makine öğrenmesi uygulamaları için kapsamlı bir araç seti sunan güçlü bir kütüphanedir. Geniş bir algoritma yelpazesi ve kullanıcı dostu arayüzü sayesinde hem yeni başlayanlar hem de deneyimli veri bilimciler için tercih edilen bir seçenek olmaktadır.[6]



```
from sklearn.model_selection import train_test_split, cross_val_score

x_train, x_test, y_train, y_test = train_test_split(X,y, test_size = 0.25, random_state = 0)
```

Şekil 4.14 Verilerin eğitim ve test olarak ikiye ayrılması.

Şekil 4.14’de sklearn kütüphanesinin **train_test_split** fonksiyonu veri setini eğitim ve test kümelerine ayırmak için kullanılır. **train_test_split** fonksiyonu, **X** veri özelliklerini ve **y** hedef değişkenini alır ve belirtilen oranda (**test_size**) rastgele bir şekilde eğitim ve test kümelerine böler.

Model eğitimi için KNN, SVC, Gaussian, Bernoulli ve Multinomial Naïve Bayes yöntemlerinde Bernoulli yöntemi seçilmiştir çünkü Bernoulli sınıflandırıcı bu projede olduğu gibi sınıflandırmanın ikili değerlerde yapıldığı veri setlerinde daha iyi çalışmaktadır. Bu projenin yorum verilerinde çoğunlukla anahtar kelime olan kargo, hızlı, beden gibi kelimelerin varlığı veya yokluğu üzerine bir etiketleme yapılmıştır. Bu nedenle Bernoulli Naïve Bayes sınıflandırıcısı bizim veri setimiz için daha iyi bir model olarak işlev görmektedir.

```

from sklearn.naive_bayes import BernoulliNB

bnb= BernoulliNB()
bnb.fit(x_train,y_train)
y_pred_bnb=bnb.predict(x_test)

bnb_cvs = cross_val_score(estimator=bnb, X = x_train, y = y_train, cv = 5)
y_pred_proba = bnb.predict_proba(x_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(y_test, y_pred_proba)

#create ROC curve
plt.plot(fpr,tpr)
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()

```

Şekil 4.15 Bernoulli Naive Bayes model eğitimi.

Şekil 4.15’deki kod bloğu ile **BernoulliNB** sınıfından bir gnb nesnesi oluşturulur. **bnb** nesnesi, eğitim verilerini (**x_train**) ve hedef değişkenleri (**y_train**) kullanarak eğitilir. Eğitilmiş model kullanılarak, test verileri (**x_test**) üzerinde tahminler (**y_pred**) yapılır. **predict** fonksiyonu, verilen özelliklerin sınıflandırma tahminlerini döndürür.

cross_val_score fonksiyonu kullanılarak, eğitim verileri üzerinde çapraz doğrulama yapılır. Bu, modelin eğitim verileri üzerindeki performansını tahmin etmek için kullanılır. **cv** parametresi ile belirtilen sayıda katlamaya (fold) bölünür ve her bir katlamada modelin performansı ölçülür. **bnb_csv** değişkeni, her bir katlamadaki doğruluk skorlarını içerir.

predict_proba fonksiyonu kullanılarak, test verileri üzerinde sınıf olasılıkları hesaplanır. **bnb.predict_proba(x_test)** ifadesi, test verilerindeki her bir örneğin sınıf olasılıklarını döndürür. Burada, **(::,1)** dilimleme işlemi kullanılarak sadece pozitif sınıfa (1) ait olasılıklar alınır.

metrics.roc_curve fonksiyonu kullanılarak, ROC eğrisi için **false positive rate** (FPR) ve **true positive rate** (TPR) değerleri hesaplanır. **y_test** ve **y_pred_proba** parametreleri, gerçek sınıf etiketleri ve sınıf olasılıklarını içerir.

plt.plot fonksiyonu ile FPR ve TPR değerleri kullanılarak ROC eğrisi çizilir. **plt.ylabel** ve **plt.xlabel** fonksiyonları ile eksen etiketleri belirlenir. **plt.show** fonksiyonu ile ROC eğrisi görüntülenir.

4.6.2 Modelin Kaydedilmesi

Eğitilen makine öğrenmesi modelinin kaydedilmesi, modelin tekrar kullanılabilirliğini, dağıtılabilirliğini, çevrimdışı kullanımını, sürekliliğini, güvenilirliğini ve yedeklemesini sağlar. Bu da modelin pratik uygulamalarda daha etkili ve verimli bir şekilde kullanılmasını sağlar.

Tekrar kullanılabilirlik: Modelin kaydedilmesi, gelecekte aynı veya benzer türdeki veriler üzerinde yeniden kullanılabilmesini sağlar. Modeli her seferinde yeniden eğitmek yerine kaydedilmiş modeli kullanmak, zaman ve kaynak maliyetlerini azaltır.

Dağıtılabilirlik: Modelin kaydedilmesi, başkalarıyla paylaşılmasını ve dağıtılmasını kolaylaştırır. Başkaları, eğitim veri setine erişim veya eğitim sürecini tekrarlamak zorunda kalmadan kaydedilmiş modeli kullanabilir.

Çevrimdışı kullanım: Kaydedilmiş bir model, internet bağlantısı olmadığında bile kullanılabilir. Bu özellik, modelin yerel uygulamalarda veya düşük bant genişliği ortamlarında kullanılmasını mümkün kılar.

Süreklilik: Modelin kaydedilmesi, modelin belirli bir zaman noktasındaki durumunu korumasını sağlar. Böylece, ilerideki analizlerde veya karşılaştırmalı çalışmalarda aynı modelin sonuçlarını elde edebilirsiniz.

Güvenilirlik: Kaydedilmiş bir model, modelin kaynak koduna veya eğitim veri setine erişim olmadan sonuçları üretebilmenizi sağlar. Bu, modelin güvenilirliğini artırır ve sonuçların tekrar edilebilirliğini sağlar.

Yedekleme: Kaydedilmiş model, veri kaybı veya sistem hatası gibi beklenmedik durumlarda modelin yedeklenmesini sağlar. Bu, modelin güvenli bir şekilde korunmasını ve geri yüklenebilmesini sağlar.

```
import pickle

filename = '/content/drive/MyDrive/Veri/bnb_model3.sav'
pickle.dump(mnb, open(filename, 'wb'))
```

Şekil 4.16 Oluşturulan modelin kaydedilmesi.

Şekil 4.16'deki kod bloğunda **pickle** kütüphanesinin **dump** fonksiyonu ile belirlenen dosya yoluna, oluşturulan model kaydedilmektedir ve Şekil 4.17'deki **load** fonksiyonu ile **loaded_model** değişkenine atanmaktadır. Yüklenen model normal bir şekilde kullanılmaya devam edilebilmektedir.

```
import pickle

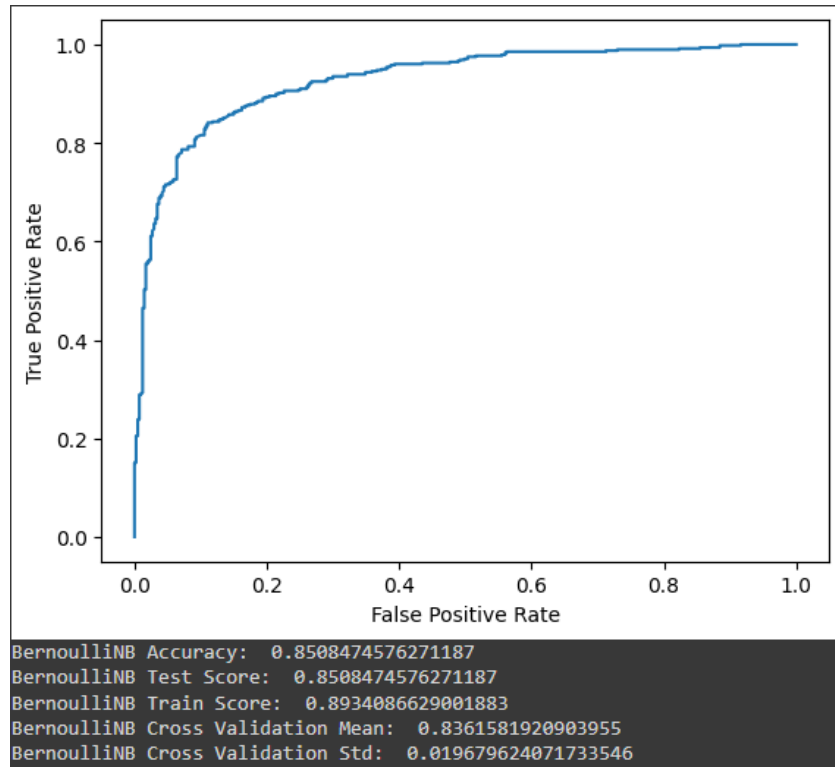
filename = '/content/drive/MyDrive/Veri/bnb_model.sav'
loaded_model = pickle.load(open(filename, 'rb'))
result = loaded_model.predict(space_matrix.transform(text))
print(result)
```

Şekil 4.17 Önceden kaydedilmiş modelin yüklenmesi.

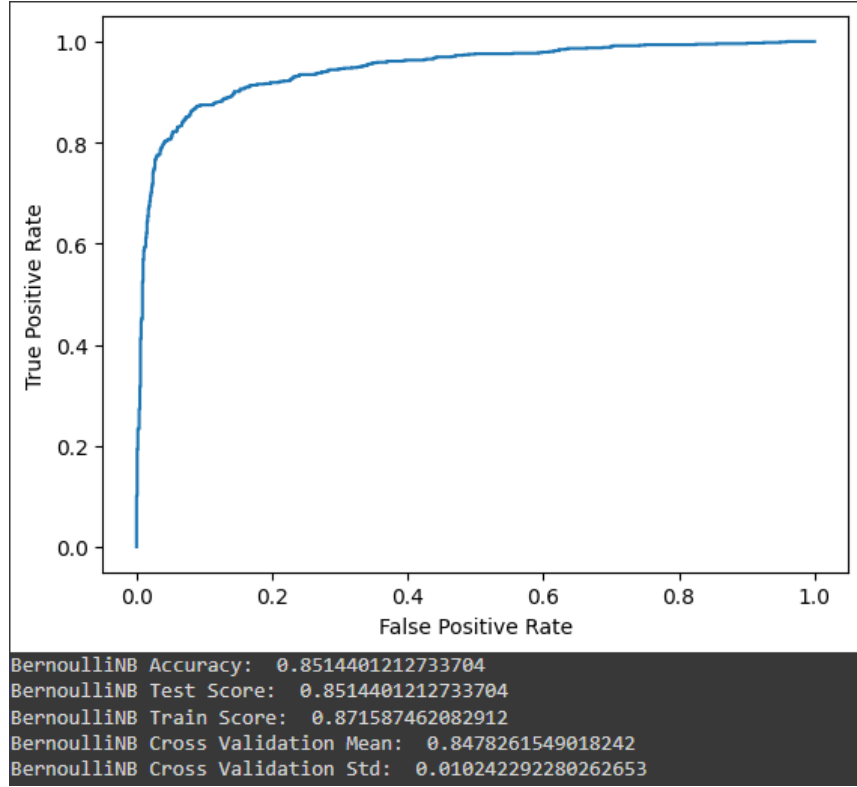
4.6.3 Model Performansı Değerlendirme

Projede eğitilen birçok makine öğrenmesi modelinin performans değerlendirme için ROC eğrisi kullanılması tercih edilmektedir. ROC eğrisi, sınıflandırma modellerinin performansını değerlendirmek için kullanılan bir görselleştirme aracıdır. Özellikle projedeki veri seti gibi ikili sınıflandırma problemlerinde yaygın olarak kullanılır. ROC eğrisi, sınıflandırma modelinin hassasiyet (True Positive Rate, TPR) ile özgüllük (True Negative Rate, TNR) arasındaki ilişkiyi gösterir. TPR, gerçek pozitif oranını, yani doğru olarak tahmin edilen pozitif örneklerin toplam pozitif örnekler içindeki oranını ifade eder. TNR ise gerçek negatif oranını, yani doğru olarak tahmin edilen negatif örneklerin toplam negatif örnekler içindeki oranını ifade eder.

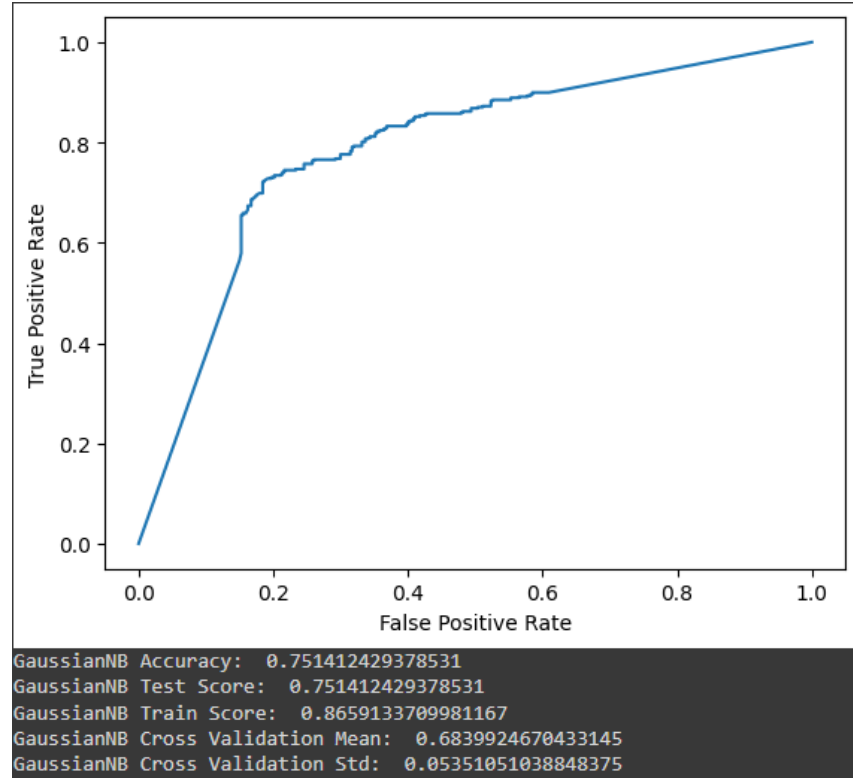
ROC eğrisi, farklı sınırlama (threshold) değerlerine göre TPR ve TNR değerlerini çizgi grafiği olarak gösterir. Eğri, sol alt köşeden başlayarak yukarı sağ köşeye doğru çizilir. Eğrinin altında kalan alan, AUC (Area Under the Curve) olarak adlandırılır ve modelin performansını ölçmek için kullanılır. AUC değeri 0 ile 1 arasında değişir ve 1'e ne kadar yakınsa, modelin performansı o kadar iyidir. ROC eğrisi, sınıflandırma modelinin duyarlılık (sensitivity) ve özgüllük (specificity) arasındaki dengeyi gösterir. İdeal bir ROC eğrisi, sol üst köşeye yakın bir noktada yer alır, yani yüksek TPR değerleriyle birlikte yüksek TNR değerlerine sahiptir. Buna karşılık, eğri yatay çizgiye yakınsarsa (diagonal çizgi), modelin performansı rastgele tahmin yapmaktan farksızdır.[9]



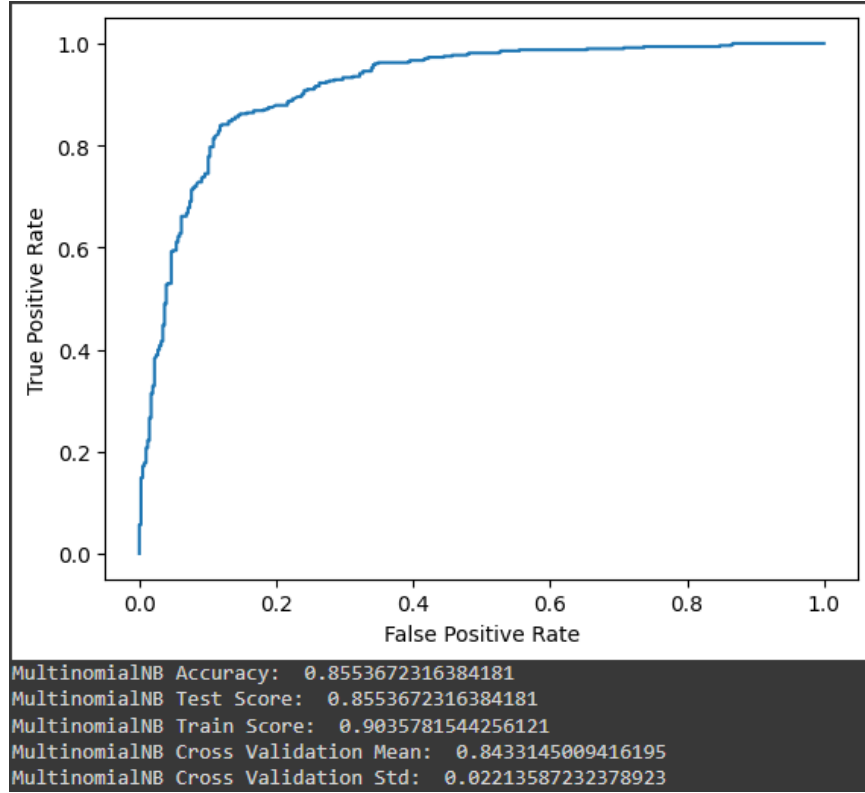
Şekil 4.18 İlgillilik modeli Bernoulli Naive Bayes ROC eğrisi.



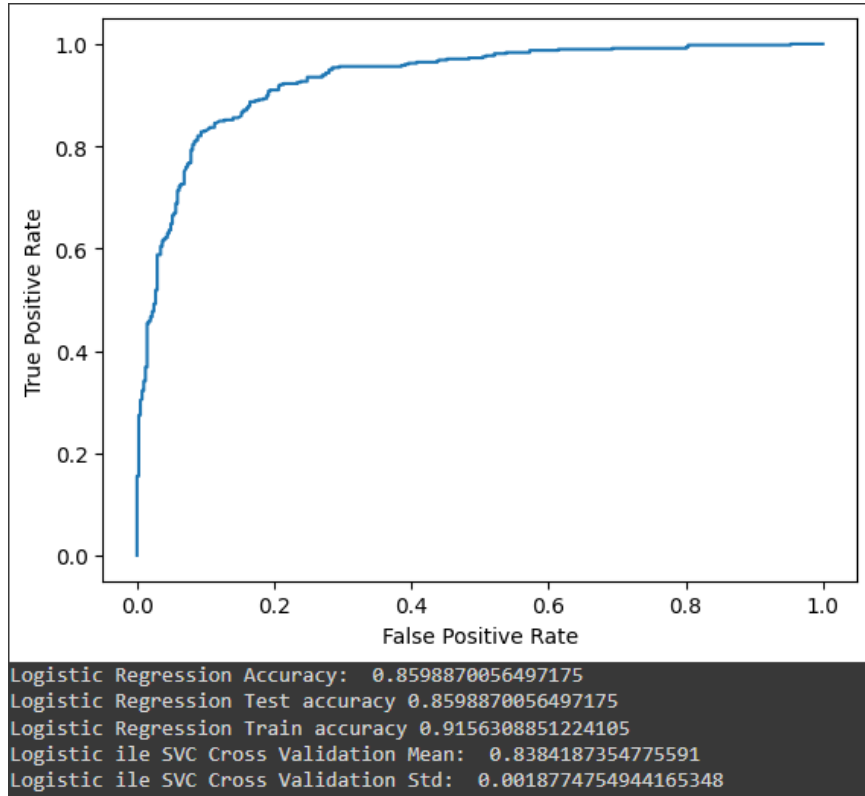
Şekil 4.19 Olumluluk modeli Bernoulli Naive Bayes ROC eğrisi.



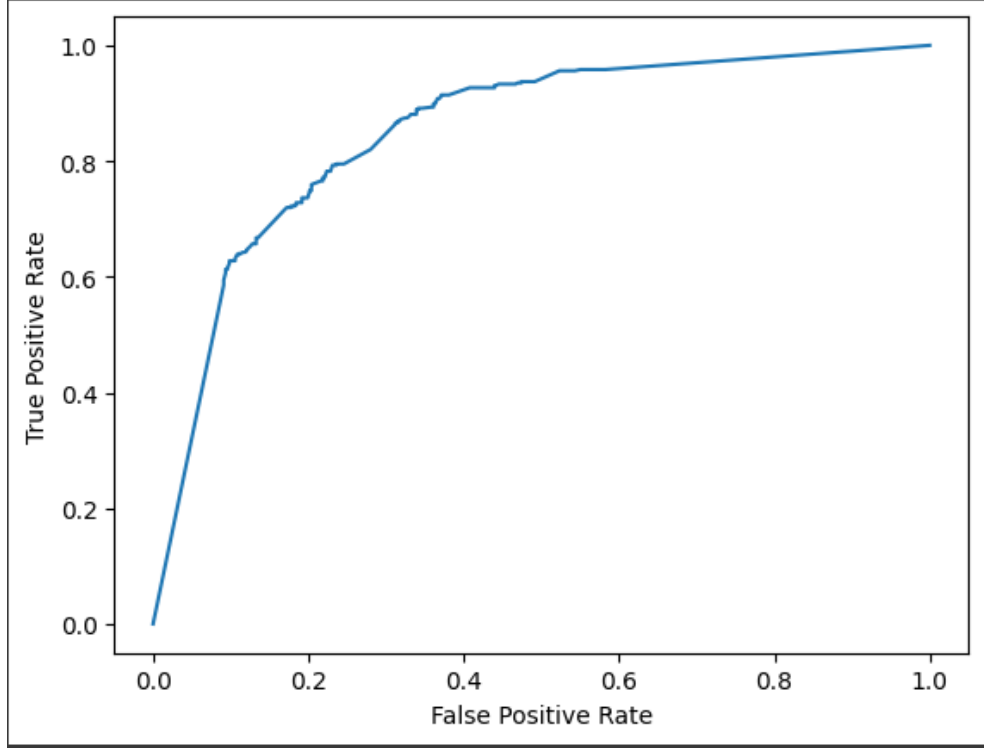
Şekil 4.20 İlgililik modeli Gaussian Naive Bayes ROC eğrisi.



Şekil 4.21 İlgililik modeli Multinomial Naive Bayes ROC eğrisi.



Şekil 4.22 İlgililik modeli Logistic Regression ROC eğrisi.



Şekil 4.23 İlgililik modeli KNN ROC eğrisi.

```
SVC Accuracy: 0.8406779661016949  
SVC Test accuracy: 0.8406779661016949  
SVC Train accuracy: 0.9337099811676083  
SVC Cross Validation Mean: 0.8320150659133709  
SVC Cross Validation Std: 0.015037638250604269
```

Şekil 4.24 İlgililik modeli Support Vector Classifier Cross Validation değerleri.

Bu grafikler sonucunda Bernoulli Naive Bayes ve Logistic Regression modelleri yakın değerlerde sonuçlar göstermektedir. Bu iki model arasında bir seçim yapılması gerektiğinden iki modelin de avantaj ve dezavantajları karşılaştırıldı. Bernoulli Naive Bayes modelinin kullanım kolaylığı ve doğal dil işleme projesine daha uygun olması sonucu bu modelin kullanılması kararlaştırıldı.

Bu modeller aynı zamanda yorumlara verilen yıldız sayısının orantılı olup olmadığını kontrol etmek için eğiteceğimiz veri seti için de aynı şekilde test edilip sonuçlandırıldıktan sonra yine Bernoulli Naive Bayes modelinin kullanılmasına karar verildi.

4.7 Modelin Ürünleştirilmesi

Oluşturulan ilgililik ve olumluluk Bernoulli Naive Bayes modelleri Flask kütüphanesi ile yerel bir web servis hizmeti haline getirilerek bir ürün oluşturmak istenmektedir. Bu ürün ile kullanıcı yapılan bir yorumun almak istenilen ürünle ilgili olup olmadığını öğrenebilecek ve girilen yıldız sayısının yorumla uyuşup uyuşmadığını da gözlemleyebilecek. [10]

4.7.1 Flask ile Sunucu Oluşturma

```
@app.route('/')
def index():
    return render_template("home.html")

@app.route('/sonuc', methods=["GET", "POST"])
def toplam():
    if request.method == "POST":
        yorum= request.form.get("yorum")
        text = [harfdegistir(yorum)]
        star = int(request.form.get('rating'))

        transformed_text=loaded_vector.transform(text).toarray()
        transformed_olumlu_text=loaded_vector_olumlu.transform(text).toarray()

        result = loaded_model.predict(transformed_text)
        ilgililik = ilgililik_dict[result[0]]
        result_olumlu = loaded_olumlu.predict(transformed_olumlu_text)
        olumluluk = olumluluk_dict[result_olumlu[0]]
        print(star)
        if(star>=3 and result_olumlu==1):
            uyum = "Verilen yıldız sayısı ile cümle uyuyor"
            return render_template("sonuc.html", total = [ilgililik, olumluluk, uyum])
        elif(star<=3 and result_olumlu==1):
            uyum = "Verilen yıldız sayısı ile cümle uyumuyor"
            return render_template("sonuc.html", total = [ilgililik, olumluluk, uyum])
        elif(star>=3 and result_olumlu==0):
            uyum = "Verilen yıldız sayısı ile cümle uyumuyor"
            return render_template("sonuc.html", total = [ilgililik, olumluluk, uyum])
        elif(star<=3 and result_olumlu==0):
            uyum = "Verilen yıldız sayısı ile cümle uyuyor"
            return render_template("sonuc.html", total = [ilgililik, olumluluk, uyum])
        else:
            return render_template("sonuc.html")
    else:
        return render_template("sonuc.html")

if __name__ == '__main__':
    app.run(debug=True)
```

Şekil 4.25 Flask ile yazılmış servis kodu.

Şekil 4.25’de gösterilen kod bloğunda **index** fonksiyonu, ana sayfayı **home.html** şablonuyla render etmektedir ve **toplamlam** fonksiyonu, **/sonuc** URL'sine GET ve POST isteklerini yönlendirir. POST isteği alındığında, kullanıcıdan gelen yorumu (**yorum**) ve yıldız değerini (**rating**) alır.

Alınan yorumu vektörize etmek için önceden eğitilmiş vektörleme modellerini (**loaded_vector**, **loaded_vector_olumlu**) kullanır. Yorumun ilgililik değerini (**ilgililik**) ve olumluluk değerini (**olumluluk**) tahmin etmek için önceden eğitilen modelleri (**loaded_model**, **loaded_olumlu**) kullanır.

Yıldız değeriyle olumluluk sonucunu karşılaştırarak, cümlemin yıldız değeriyle uyumlu olup olmadığını belirler. Sonuçları **sonuc.html** şablonuna aktarır ve render eder. Uygun sonucun yanı sıra, ilgililik ve olumluluk değerlerini de gösterir. Ana uygulama **__main__** bölümünde çalıştırılır ve Flask uygulamasını başlatır.

```
<body>
  <div class="container">
    <h1>Yorum ve Yıldız Değerlendirmesi</h1>
    <form action="/sonuc" method="POST">
      <label for="yorum">Yorumunuz:</label>
      <textarea name="yorum" id="yorum" required></textarea>
      <label for="rating">Değerlendirme:</label>
      <div class="stars">
        <input type="radio" name="rating" id="star1" value="1"><label
for="star1">&#9733;</label>
        <input type="radio" name="rating" id="star2" value="2"><label
for="star2">&#9733;</label>
        <input type="radio" name="rating" id="star3" value="3"><label
for="star3">&#9733;</label>
        <input type="radio" name="rating" id="star4" value="4"><label
for="star4">&#9733;</label>
        <input type="radio" name="rating" id="star5" value="5"><label
for="star5">&#9733;</label>
      </div>
      <button type="submit" class="submit-btn">Gönder</button>
    </form>
  </div>
</body>
```

Şekil 4.26 HTML ile yazılmış ana sayfa arayüzü.

Şekil 4.26’de gösterilen HTML kodu, projenin web tarayıcısında açıldığında kullanıcıya yorum ve yıldız değerlendirmesi yapma imkanı sunan bir sayfayı görüntülemesini sağlar. Formun gönderildiği URL **/sonuc** olarak belirtilmektedir ve veriler **POST** metoduyla gönderilmektedir.

Şekil 4.27’de gösterilen HTML kodunda, üç farklı durumu temsil eden **if-else** blokları bulunmaktadır. İlk olarak, yorumun olumluluk durumu kontrol edilir. Eğer olumlu ise, yeşil bir **"alert-success"** bileşeni görüntülenir ve olumluluk durumu mesajı kullanıcıya gösterilmektedir. Eğer olumsuz ise, kırmızı bir **"alert-danger"** bileşeni görüntülenmektedir ve olumluluk durumu mesajı kullanıcıya gösterilmektedir.

İkinci blokta yıldız değerlendirmesi ile cümle uyuşumluluk durumu kontrol edilmektedir. Eğer verilen yıldız sayısı ile cümle uyuşuyorsa, yeşil bir **"alert-success"** bileşeni görüntülenir ve uyumluluk durumu mesajı kullanıcıya gösterilir. Eğer uyuşmuyorsa, kırmızı bir **"alert-danger"** bileşeni görüntülenir ve uyumluluk durumu mesajı kullanıcıya gösterilmektedir.

Son olarak, ilgililik durumu kontrol edilmektedir. Eğer ilgili ise, yeşil bir **"alert-success"** bileşeni görüntülenir ve ilgililik durumu mesajı kullanıcıya gösterilmektedir. Eğer ilgisiz ise, kırmızı bir **"alert-danger"** bileşeni görüntülenir ve ilgililik durumu mesajı kullanıcıya gösterilmektedir. Bu şekilde, kullanıcının yaptığı değerlendirme ve yorum sonucunda elde edilen sonuçlar kullanıcıya anlamlı bir şekilde iletilmektedir.

```

<body>
  <div class="container">
    {% if total[1]=="Olumlu" %}
      <div class="alert alert-success" role="alert">
        <p><strong>Olumluluk Durumu:</strong> {{total[1]}}</p>
      </div>
    {% elif total[1]=="Olumsuz" %}
      <div class="alert alert-danger" role="alert">
        <p><strong>Olumluluk Durumu:</strong> {{total[1]}}</p>
      </div>
    {% else %}
      <div class="alert alert-danger" role="alert">
        <p>Bu bir GET request olduğu için total değeri yok.</p>
      </div>
    {% endif %}
  </div>

  <div class="container">

    {% if total[2]=="Verilen yıldız sayısı ile cümle uyuyor" %}
      <div class="alert alert-success" role="alert">
        <p><strong>Uyumluluk Durumu:</strong> {{total[2]}}</p>
      </div>
    {% elif total[2]=="Verilen yıldız sayısı ile cümle uyumuyor" %}
      <div class="alert alert-danger" role="alert">
        <p><strong>Uyumluluk Durumu:</strong> {{total[2]}}</p>
      </div>
    {% else %}
      <div class="alert alert-danger" role="alert">
        <p>Bu bir GET request olduğu için total değeri yok.</p>
      </div>
    {% endif %}
  </div>

  <div class="container">
    {% if total[0]=="İlgili" %}
      <div class="alert alert-success" role="alert">
        <p><strong>İlgililik Durumu:</strong> {{total[0]}}</p>
      </div>
    {% elif total[0]=="İlgisiz" %}
      <div class="alert alert-danger" role="alert">
        <p><strong>İlgililik Durumu:</strong> {{total[0]}}</p>
      </div>
    {% else %}
      <div class="alert alert-danger" role="alert">
        <p>Bu bir GET request olduğu için total değeri yok.</p>
      </div>
    {% endif %}
  </div>
</body>

```

Şekil 4.27 Ana sayfada girilen verilerin modelden geçirilerek oluşturulmuş sonuçlarının gösterildiği sayfa.

5. UYGULAMA ÇIKTILARI

Yorum ve Yıldız Değerlendirmesi

Yorumunuz:

çok güzel bir ürün

Değerlendirme:



Gönder

Şekil 5.1 Yorum ve yıldız sayısı girişi yapılan ana sayfa arayüzü.

Olumluluk Durumu: Olumlu

Uyumluluk Durumu: Verilen yıldız sayısı ile cümle uyuyor

İlgililik Durumu: İlgili

Şekil 5.2 Ana sayfada girilen verilerin modellerden geçirildikten sonra sonuçlarının gösterildiği arayüz.

Yorum ve Yıldız Değerlendirmesi

Yorumunuz:

çok kötü bir ürün

Değerlendirme:



Gönder

Şekil 5.3 Ana sayfa arayüzü için ikinci bir örnek.

Olumluluk Durumu: Olumsuz

Uyumluluk Durumu: Verilen yıldız sayısı ile cümle uyuşmuyor

İlgililik Durumu: İlgili

Şekil 5.4 Sonuç arayüzü için ikinci bir örnek.

6. SONUÇ

Sonuç olarak bu proje, Trendyol ürün yorumlarının sınıflandırılarak filtrelenmesi için makine öğrenmesi modellerinin kullanılmasına dayalı yenilikçi bir yaklaşım göstermektedir. Makine öğrenimi modellerinin doğal dil işleme konusu için kullanılmasına da katkıda bulunmaktadır. Ayrıca proje veri setine ilerleyen zamanlarda daha fazla ürünün yorumlarının sınıflandırılarak eklenmesi, bu projenin daha da gelişebilecek potansiyele sahip olduğunu göstermektedir.

Projede kullanılan Flask, Scikit-learn, Pandas, Numpy, NLTK, Pickle gibi açık kaynak kodlu teknolojilerin birlikte kullanılmasının getirdiği avantajlar ile projede çeşitlilik sağlandı ve daha fazlasının entegre edilebilmesi için proje modüler olarak kodlandı.

6.1 Çalışmanın Uygulama Alanı

Bu projenin kullanım alanları müşteri hizmetlerinden, pazarlama ve satışa kadar birçok alanda kullanılmaya uygundur. Trendyol ve diğer e-ticaret platformlarındaki ürün yorumlarının etkili bir şekilde analiz edilerek, müşteri deneyimi, pazarlama stratejileri ve işletme kararlarına katkıda bulunmasını sağlar.

Ürün Değerlendirme: Trendyol gibi e-ticaret platformlarında satılan ürünler için kullanıcıların yaptığı yorumların ürünle ilgili olup olmadığını kontrol etmek, daha güvenilir ürün değerlendirmeleri sunabilir. İlgili yorumlar, gerçek kullanıcı deneyimlerini yansıtarak alışveriş yapan müşterilere daha doğru bilgiler sağlar.

Müşteri Hizmetleri: Trendyol veya diğer benzer platformlarda müşteri hizmetleri ekipleri, kullanıcıların yorumlarını kontrol ederek hızlı ve etkili geri bildirimler sağlayabilir. İlgili yorumlar, müşteri sorunlarını çözmek veya yardımcı olmak için daha hızlı tepkiler verilmesini sağlar.

Pazarlama ve Satış Analizi: Ürünle ilgili yorumları otomatik olarak analiz etmek, markaların pazarlama ve satış stratejilerini geliştirmelerine yardımcı olabilir. İlgili yorumlar, ürünlerin neden beğenildiğini veya beğenilmediğini anlamak için değerli veriler sunar. Bu veriler, ürün iyileştirmeleri, pazarlama kampanyaları ve müşteri memnuniyeti odaklı stratejilerin belirlenmesinde kullanılabilir.

Ürün Sınıflandırması ve Filtreleme: İlgili yorumları otomatik olarak tespit etmek, Trendyol gibi platformlarda ürün sınıflandırması ve filtreleme işlemlerini geliştirebilir.

Bu sayede kullanıcılar, ürünleri daha doğru bir şekilde arama, filtreleme ve karşılaştırma imkanına sahip olabilirler.

Sentiment Analizi ve Trend Analizi: Yorumları otomatik olarak analiz etmek, kullanıcıların duygusal tepkilerini anlamak ve trendleri belirlemek için kullanılabilir. Olumlu ve olumsuz yorumlar üzerinde yapılan analizler, markaların ürünlerine ilişkin genel algıyı ve müşteri memnuniyetini ölçmelerine yardımcı olur.

7. KAYNAKÇA

- [1] Havva Yılmaz, Semih Yumuşak, “Açık Kaynak Doğal Dil İşleme Kütüphaneleri” (2021)
<https://dergipark.org.tr/en/download/article-file/1573501>
- [2] Kemal Oflazer, “Türkçe ve Doğal Dil İşleme” (2012)
<https://dergipark.org.tr/en/pub/tbbmd/issue/22245/238795>
- [3] Seda Tuzcu, “Çevrimiçi Kullanıcı Yorumlarının Duygu Analizi ile Sınıflandırılması” (2020)
<https://dergipark.org.tr/tr/pub/estudambilisim/issue/53654/676052>
- [4] Gülşen Eryiğit, “ITU Turkish NLP Web Service” (2014)
<https://web.itu.edu.tr/gulsenc/papers/itunlp.pdf>
- [5] Burhan Bilen, Fahrettin Horasan, “LSTM Network based Sentiment Analysis for Customer Reviews” (2022)
<https://dergipark.org.tr/tr/pub/politeknik/issue/73018/844019>
- [6] <https://www.kaggle.com/code/ihsncnkz/techcareer-project#Veri-Seti-Inceleme>
- [7] Sinem Tokcaer, Türkçe Metinlerde Duygu Analizi(2021)
<https://dergipark.org.tr/tr/download/article-file/1736703>
- [8] https://github.com/tanersolak/Trendyol_Yorum_Scraping
- [9] <https://www.statology.org/plot-roc-curve-python/>
- [10] <https://www.geeksforgeeks.org/retrieving-html-from-data-using-flask/>

ÖZGEÇMİŞ

TARANMIŞ
VESİKALIK
FOTOĞRAF

Ad-Soyad : Taner Solak

Doğum Tarihi ve Yeri : 04.07.2000 / İzmir/Konak

E-posta : taner2164@gmail.com

BİTİRME ÇALIŞMASINDAN TÜRETİLEN MAKALE, BİLDİRİ VEYA SUNUMLAR:

-
-
-