

**T.C.**

**BİTLİS EREN ÜNİVERSİTESİ**

**MÜHENDİSLİK-MİMARLIK FAKÜLTESİ**

**STAJ DEFTERİ**

Öğrenci No : 20080410022

Adı, Soyadı :Mecdiddin Ebuşerfo

**2015**

|  |  |
| --- | --- |
| 200px-Bitlis_Eren_Üniversitesi_logosu | T.C.  BİTLİS EREN ÜNİVERSİTESİ  MÜHENDİSLİK MİMARLIK FAKÜLTESİ  ............................ Mühendisliği Bölümü |

**STAJ DEFTERİ**



fotoğraf

ÖĞRENCİNİN Numarası : 20080410022

Adı ve Soyadı :MECDİDDİN EBUŞERFO

Sınıfı : 4

Staj No : 2

Staja başlama tarihi : 24/06/ 2024

Stajı bitirme tarihi : 22/07/ 2024

Çalıştığı işgünü toplamı : 20 İŞ GÜNÜ

STAJIN YAPILDIĞI İŞYERİNİN

Adı :................................................................

Adresi :................................................................

...............................................................................

Faaliyet Alanı : ...............................................................

İletişim Bilgileri : ...............................................................

İŞYERİ SORUMLU AMİRİNİN

Adı ve Soyadı Unvanı Tarih İmza ve Mühür

...................... ................ ..../..../ 201

**STAJ İLE İLGİLİ HATIRLATMA**

ÖĞRENCİLERİN; STAJLARINININ KABUL EDİLEBİLMESİ VE DEFTERLERİNİN DEĞERLENDİRİLEBİLMESİ İÇİN STAJLARINI, MÜHENDİSLİK MİMARLIK FAKÜLTESİ STAJ UYGULAMA ESASLARINA GÖRE YAPMALARI VE STAJ DEFTERLERİNİ, STAJ DEFTERİ YAZIM KURALLARINA UYGUN OLARAK HAZIRLAMALARI GEREKMEKTEDİR.

DEFTER, ÖN VE ARKA KAPAK SAYFALARI KARTON OLACAK BİÇİMDE CİLTLENMİŞ OLARAK TESLİM EDİLMELİDİR.

**İÇİNDEKİLER**

Konu Sayfa No

[Stajyer tanıtımı ve şirket oryantasyonu. 1](#_Toc175777617)

[Yapay Zeka ve Makine Öğrenmesi Temelleri. 2](#_Toc175777618)

[Veri Ön İşleme ve Özellik Çıkartma. 5](#_Toc175777619)

[Denetimli Öğrenme Algoritmaları. 7](#_Toc175777620)

[Derin Öğrenme Temelleri. 9](#_Toc175777621)

[Doğal Dil İşleme (NLP) Temelleri. 11](#_Toc175777622)

[Sentiment Analizi Projesi. 13](#_Toc175777623)

[Görüntü İşleme Temelleri. 16](#_Toc175777624)

[Convolutional Neural Networks (CNN) ile Görüntü Sınıflandırma. 18](#_Toc175777625)

[Büyük Proje Başlangıcı: Otonom Araç Simülasyonu. 21](#_Toc175777626)

[Reinforcement Learning (Pekiştirmeli Öğrenme) Temelleri. 23](#_Toc175777627)

[Rasa Chatbot ve Docker Kullanımı 26](#_Toc175777628)

[Otonom Araç Projesi 29](#_Toc175777629)

[Keras ve TensorFlow kullanarak Derin Öğrenme 33](#_Toc175777630)

[Film Öneri Sistemi 35](#_Toc175777631)

[IMDb Film Yorumları Metin Sınıflandırma Projesi 37](#_Toc175777632)

[Metin Tematik Analizi ve LDA Modeli ile Anahtar Kelime Çıkarımı Projesi 39](#_Toc175777633)

[Makine Öğrenimi ile Tahmin Modellemesi Projesi 42](#_Toc175777634)

[Zaman Serisi Tahmini İçin LSTM Modeli 45](#_Toc175777635)

[Tic-Tac-Toe Oyunu ve Pekiştirmeli Öğrenme Ajanı 48](#_Toc175777636)

[**Özet** 50](#_Toc175777637)

**ŞEKİL, ÇİZELGE VE EKLER LİSTESİ**

Şekil, Çizelge veya Ek No ve Adı Sayfa No

[Şekil 1Basit Lineer Regresyon kodu 2](#_Toc175778995)

[Şekil 2 :Basit Lineer Regresyon Çıkışı 4](#_Toc175778996)

[Şekil 3Veri Ön İşleme Python Kodu 5](#_Toc175778997)

[Şekil 4Veri Ön İşleme Çıkış Python Kodu 5](#_Toc175778998)

[Şekil 5Random Forest 7](#_Toc175778999)

[Şekil 6İkili Sınıflandırma için Yapay Sinir Ağı Modeli 9](#_Toc175779000)

[Şekil 7İkili Sınıflandırma için Yapay Sinir Ağı Modeli Çıkışı 9](#_Toc175779001)

[Şekil 8 Metin Ön İşleme 11](#_Toc175779002)

[Şekil 9Sentiment Analizi 13](#_Toc175779003)

[Şekil 10Sentiment Analizi Çıkışı 13](#_Toc175779004)

[Şekil 11Görüntü İşleme 16](#_Toc175779005)

[Şekil 12Görüntü İşleme Çıkışı 16](#_Toc175779006)

[Şekil 13CNN ile Görüntü Sınıflandırma 18](#_Toc175779007)

[Şekil 14Otonom Araç Simülasyonu 21](#_Toc175779008)

[Şekil 15Pekiştirmeli Öğrenme 23](#_Toc175779009)

[Şekil 16CartPole Oyun Animasyonu 24](#_Toc175779010)

[Şekil 17docker-compose.yml Dosyası 26](#_Toc175779011)

[Şekil 18Rasa Chatbot Kodun Çalışması 28](#_Toc175779012)

[Şekil 19Otonom Araç ve Engel Algılama kodu 29](#_Toc175779013)

[Şekil 20Otonom Araç ve Engel Algılama kodu çıkışı 32](#_Toc175779014)

[Şekil 21Otonom Araç Simülasyonu kodu 33](#_Toc175779015)

[Şekil 22Otonom Araç Simülasyonu kodu Çıkışı 34](#_Toc175779016)

[Şekil 23Film Öneri Sistemi kodu 35](#_Toc175779017)

[Şekil 24Film Öneri Sistemi kodu çıkışı 36](#_Toc175779018)

[Şekil 25Metin Sınıflandırma kodu 37](#_Toc175779019)

[Şekil 26Metin Sınıflandırma kodu çıkışı 38](#_Toc175779020)

[Şekil 27LDA Modeli Kodu 39](#_Toc175779021)

[Şekil 28Excel Dosyasının Oluşturma kodu 39](#_Toc175779022)

[Şekil 29LDA Modeli kodu çıkışı 40](#_Toc175779023)

[Şekil 30Tahmin Modellemesi Kodu 42](#_Toc175779024)

[Şekil 31Tahmin Modellemesi Kodu Çıkışı 43](#_Toc175779025)

[Şekil 32Zaman Serisi Tahmini Kodu 45](#_Toc175779026)

[Şekil 33Zaman Serisi Tahmini Kodu Çıkışı 47](#_Toc175779027)

[Şekil 34Tic-Tac-Toe oyunu ara yüzü 48](#_Toc175779028)

[Şekil 35Kimin kazandığını gösterir 48](#_Toc175779029)

[Şekil 36Beraberlik görünüyor 49](#_Toc175779030)

**STAJIN YAPILDIĞI KURUM VEYA KURULUŞUN TANITIMI**

**GİRİŞ**

Bu staj defteri, 24/06/2024 ile 22/07/2024 tarihleri arasında, 20 iş günü boyunca gerçekleştirdiğim çalışmalarımı kapsamaktadır. Staj süresince yapay zeka, makine öğrenmesi, derin öğrenme, doğal dil işleme (NLP), görüntü işleme, otonom araç teknolojileri, pekiştirmeli öğrenme ve diğer ileri düzey konular üzerinde yoğun bir şekilde çalıştım.

Bu süreçte, hem teorik bilgilerimi derinleştirdim hem de edindiğim bilgileri çeşitli projelerde uygulama fırsatı buldum. Her bir çalışma, ilgili konularda derinlemesine bir anlayış geliştirmeme ve projeler üzerinden pratik deneyim kazanmama katkı sağladı. Ayrıca, Python programlama dili ve popüler yapay zeka kütüphanelerini (TensorFlow, Keras, OpenCV, NLTK vb.) etkin bir şekilde kullanarak, çeşitli gerçek dünya senaryolarını simüle eden projeler geliştirdim.

Staj boyunca kazandığım tecrübe ve bilgiler, yapay zeka ve makine öğrenmesi alanında daha ileri seviyede çalışmalar yapmak için sağlam bir temel oluşturdu. Bu staj defteri, her bir günün detaylı açıklamalarını, projelerin kod örneklerini ve elde edilen sonuçları içermektedir.

**24/06/2024** tarihinden **30/06/2024** tarihine kadar bir haftalık çalışma.

|  |  |  |  |
| --- | --- | --- | --- |
| GÜN | Yapılan İşler | Yapılan işe ait bilginin bulunduğu sayfa | Saat |
| Pazartesi | Stajyer tanıtımı ve şirket oryantasyonu. | 1 | 8 |
| Salı | **Yapay Zeka ve Makine Öğrenmesi Temelleri.** | 2,3,4 | 8 |
| Çarşamba | **Veri Ön İşleme ve Özellik Çıkartma.** | 5,6 | 8 |
| Perşembe | **Denetimli Öğrenme Algoritmaları.** | 7,8 | 8 |
| Cuma | **Derin Öğrenme Temelleri.** | 9,10 | 8 |
| Denetleyenin İmzası |  | Toplam Saat | 40 |

**01/07/2024** tarihinden **07/07/2024** tarihine kadar bir haftalık çalışma.

|  |  |  |  |
| --- | --- | --- | --- |
| GÜN | Yapılan İşler | Yapılan işe ait bilginin bulunduğu sayfa | Saat |
| Pazartesi | **Doğal Dil İşleme (NLP) Temelleri.** | 11,12 | 8 |
| Salı | **Sentiment Analizi Projesi.** | 13,14,15 | 8 |
| Çarşamba | **Görüntü İşleme Temelleri.** | 16,17 | 8 |
| Perşembe | **Convolutional Neural Networks (CNN) ile Görüntü Sınıflandırma.** | 18,19,20 | 8 |
| Cuma | Otonom Araç Simülasyonu. | 21,22 | 8 |
| Denetleyenin İmzası |  | Toplam Saat | 40 |

**08/07/2024** tarihinden **14/07/2024** tarihine kadar bir haftalık çalışma

|  |  |  |  |
| --- | --- | --- | --- |
| GÜN | Yapılan İşler | Yapılan işe ait bilginin bulunduğu sayfa | Saat |
| Pazartesi | Reinforcement Learning (Pekiştirmeli Öğrenme) Temelleri. | 23,24,25 | 8 |
| Salı | **Rasa Chatbot ve Docker Kullanımı** | 26,27,28 | 8 |
| Çarşamba | Otonom Araç Projesi | 29,30,31,32 | 8 |
| Perşembe | Keras ve TensorFlow kullanarak Derin Öğrenme | 33,34 | 8 |
| Cuma | Film Öneri Sistemi | 35,36 | 8 |
| Denetleyenin İmzası |  | Toplam Saat | 40 |

**15/07/2024** tarihinden **21/07/2024** tarihine kadar bir haftalık çalışma.

|  |  |  |  |
| --- | --- | --- | --- |
| GÜN | Yapılan İşler | Yapılan işe ait bilginin bulunduğu sayfa | Saat |
| Pazartesi | RESMI TATIL |  | 0 |
| Salı | IMDb Film Yorumları Metin Sınıflandırma Projesi | 37,38 | 8 |
| Çarşamba | Metin Tematik Analizi ve LDA Modeli ile Anahtar Kelime Çıkarımı Projesi | 39,40,41 | 8 |
| Perşembe | Makine Öğrenimi ile Tahmin Modellemesi Projesi | 42,43,44 | 8 |
| Cuma | Zaman Serisi Tahmini İçin LSTM Modeli | 45,46,47 | 8 |
| Denetleyenin İmzası |  | Toplam Saat | 32 |

**22/07/2024** tarihinden **28/07/2024** tarihine kadar bir haftalık çalışma.

|  |  |  |  |
| --- | --- | --- | --- |
| GÜN | Yapılan İşler | Yapılan işe ait bilginin bulunduğu sayfa | Saat |
| Pazartesi | Tic-Tac-Toe Oyunu ve Pekiştirmeli Öğrenme Ajanı | 48,49,50 | 8 |
| Salı |  |  |  |
| Çarşamba |  |  |  |
| Perşembe |  |  |  |
| Cuma |  |  |  |
| Denetleyenin İmzası |  | Toplam Saat | 8 |

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

### ****Stajyer tanıtımı ve şirket oryantasyonu.****

İlk günümde, şirketin genel yapısını ve işleyişini anlamak amacıyla kapsamlı bir oryantasyon programına katıldım. Şirketin farklı departmanları hakkında bilgi aldım ve her birinin iş süreçlerini öğrendim. Bu süreçte, çalışma ortamını daha yakından tanıma fırsatım oldu ve ekip arkadaşlarımla tanıştım. Ekip arkadaşlarımın samimi ve yardımsever tavırları, adaptasyon sürecimi oldukça kolaylaştırdı. Ayrıca, şirketin misyonu, vizyonu ve değerleri hakkında da bilgi sahibi oldum, bu da bana çalışacağım projelerde nelere dikkat etmem gerektiği konusunda önemli ipuçları verdi. Bu oryantasyon sayesinde, şirketin beklentilerini ve benim rolümü daha iyi anladım, bu da staj sürecine sağlam bir başlangıç yapmamı sağladı.

Oryantasyonun ardından, teknik hazırlıklara da başladım. Staj boyunca üzerinde çalışacağım projeler için Python dilini kapsamlı bir şekilde gözden geçirdim ve öğrenme sürecimi desteklemek amacıyla uygun bir çalışma ortamı kurdum. Bu süreçte, gerekli yazılımları ve kütüphaneleri kurarak, projelere hızlı bir başlangıç yapmamı sağlayacak bir temel oluşturdum. Bu hazırlıklar sayesinde, önümüzdeki günlerde karşılaşacağım zorluklara daha hazır hissediyorum.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

### ****Yapay Zeka ve Makine Öğrenmesi Temelleri.****

İkinci günümde, yapay zeka ve makine öğrenmesi kavramları üzerine kapsamlı bir eğitim aldım. Bu eğitimde, yapay zekanın temel prensipleri ve nasıl çalıştığı hakkında bilgi edindim. Ayrıca, yapay zekanın hayatımızdaki yeri ve çeşitli kullanım alanları hakkında farkındalığım arttı. Makine öğrenmesi üzerine de yoğunlaşıldı ve özellikle supervised (denetimli) ve unsupervised (denetimsiz) öğrenme yöntemleri üzerine derinlemesine bilgi verildi.Supervised öğrenmede, modelin etiketli verilerle eğitildiği ve bu sayede tahminlerde bulunabildiği anlatıldı. Bu yöntemde, modelin eğitimi sırasında girdi ve çıktı verileri belirli olduğundan, modelin hedefi bu verilerden öğrenmek ve gelecekteki verilere doğru tahminler yapmaktır. Örneğin, bir veri setinde öğrencilere ait notlar (girdi) ve başarı durumu (çıktı) verildiğinde, model bu ilişkiden yola çıkarak yeni bir öğrencinin başarı durumunu tahmin edebilir.Unsupervised öğrenmede ise, verilerin etiketlenmediği ve bu veriler arasındaki yapıları keşfetmek amacıyla modellerin eğitildiği açıklandı. Bu yöntem genellikle veriler arasında gizli desenleri keşfetmek için kullanılır. Örneğin, müşteri alışveriş alışkanlıklarını analiz etmek amacıyla unsupervised öğrenme kullanarak müşteri segmentleri oluşturulabilir.

Bu teorik bilgilerin ardından, öğrendiklerimizi pekiştirmek amacıyla basit bir lineer regresyon örneği üzerinde çalıştık. 

Şekil 1Basit Lineer Regresyon kodu

**Lineer Regresyon Projesi:**

Bu projede, basit bir lineer regresyon modelini uygulayarak öğrendiğimiz teorik bilgileri pratiğe döktük.

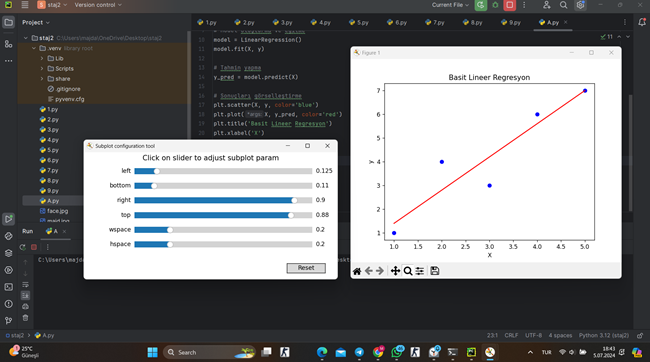
**Kütüphanelerin İçe Aktarılması:**

* + matplotlib.pyplot: Veri görselleştirmeleri yapıldı.
  + sklearn.linear\_model: Lineer regresyon modelini oluşturmak için kullanıldı.

1. **Veri Oluşturma:**
   * X ve y adında iki veri kümesi oluşturduk.
   * X, 1'den 5'e kadar olan sayıları içerirken, y karşılık gelen sonuç değerlerini içeriyordu. Bu değerler arasında lineer bir ilişki kurarak veri setimizi oluşturduk.
2. **Model Oluşturma ve Eğitme:**
   * LinearRegression sınıfını kullanarak bir model oluşturduk.
   * fit metodu ile bu modeli X ve y verileri üzerinde eğittik. Eğitim aşamasında, model X verilerindeki girdi ile y verilerindeki çıktılar arasında bir ilişki kurdu.
3. **Tahmin Yapma:**
   * Model eğitildikten sonra, predict metodu ile X verileri üzerinden tahminlerde bulunduk. Bu tahminler y\_pred olarak kaydedildi ve görselleştirildi.
4. **Sonuçları Görselleştirme:**
   * plt.scatter fonksiyonu ile veri noktalarını mavi renkte, plt.plot fonksiyonu ile tahmin edilen çizgiyi kırmızı renkte çizdik.
   * Grafik başlıklarını ve eksen etiketlerini ekleyerek görselleştirmeyi tamamladık.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |



Şekil 2 :Basit Lineer Regresyon Çıkışı

**Projenin Sonuçları ve Öğrendiklerimiz:**

Bu basit lineer regresyon projesi sayesinde, veriler arasındaki ilişkiyi görselleştirmenin önemini anladık. Modelin doğruluğunu grafik üzerinde görselleştirerek, tahmin edilen değerlerin orijinal veri noktalarına ne kadar yakın olduğunu gözlemledik. Ayrıca, makine öğrenmesinin temellerine dair pratik bir uygulama yapma fırsatı bulduk.

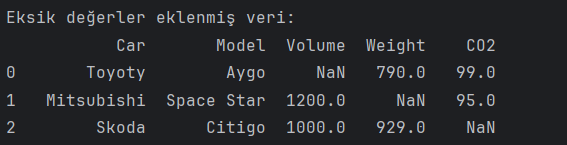
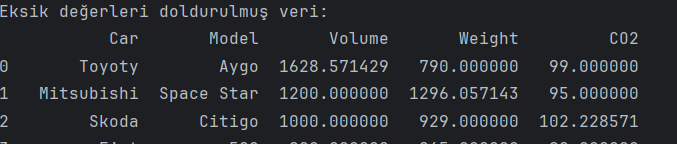
Bu proje, denetimli öğrenme yöntemlerinin basit ama etkili bir uygulaması olarak karşımıza çıktı ve öğrendiklerimizi somutlaştırdı.

### ****Veri Ön İşleme ve Özellik Çıkartma.****

Üçüncü günümde, makine öğrenmesi projelerinde veri hazırlama sürecinin önemi ve yöntemleri hakkında detaylı bir eğitim aldım. Bu süreç, veri temizleme, normalizasyon ve özellik mühendisliği gibi temel adımları içerir. Eğitim süresince, verilerin nasıl temizleneceği, eksik değerlerin nasıl doldurulacağı ve verilerin nasıl ölçeklendirileceği gibi temel veri ön işleme teknikleri uygulamalı olarak gösterildi. Ayrıca, özellik mühendisliği ile verilerden anlamlı bilgiler çıkarma yöntemleri de ele alındı. Bu bilgiler, makine öğrenmesi modellerinin performansını artırmada kritik bir rol oynamaktadır.



Şekil 3Veri Ön İşleme Python Kodu

Şekil 4Veri Ön İşleme Çıkış Python Kodu

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

**Sonuçları Görüntüleme:** Eğitim sonunda, eksik değerleri doldurulmuş veri setini görselleştirerek elde edilen sonuçları analiz ettik. Veri seti, başlangıçta eksik değerler içeriyordu ve bu eksik değerler, uygun yöntemler kullanılarak dolduruldu.

**Veri Yükleme:** Eğitim sürecinde, veri setinin nasıl yükleneceği ve işleneceği konusunda bilgi sahibi oldum. Veri seti bir CSV dosyasından yüklendi ve analiz için hazır hale getirildi.

**Eksik Değerler Oluşturma:** Eğitmen, veri setine kasıtlı olarak eksik değerler ekleyerek, veri ön işleme tekniklerini uygulamalı olarak göstermemi sağladı. Bu sayede, eksik verilerin nasıl yönetileceğini öğrenmiş oldum.

**Veriyi Görüntüleme:** Eğitim sürecinde, eksik değerlerin eklenmiş haliyle veri setini görselleştirme ve analiz etme üzerine çalıştık.

**Sayısal Sütunları Seçme:** Sayısal verilerin önemi üzerinde durularak, bu sütunların nasıl seçileceği ve analiz edileceği hakkında bilgi verildi. Eksik değer doldurma işlemi bu sütunlar üzerinde uygulandı.

**Eksik Değerleri Doldurma:** Eksik değerlerin, ilgili sütunun ortalama değeri ile nasıl doldurulacağı üzerinde duruldu. Bu, veri setinin analiz için daha uygun hale getirilmesini sağladı.

**Öğrendiklerimiz:** Üçüncü gün sonunda, makine öğrenmesi projelerinde veri ön işleme sürecinin ne kadar kritik bir rol oynadığını ve bu sürecin model performansını doğrudan etkilediğini öğrendim. Verilerin doğru bir şekilde temizlenmesi, normalizasyonun düzgün yapılması ve özellik mühendisliği ile anlamlı bilgiler çıkarılması, modelin başarısını artıran temel faktörlerdir. Bu adımların her biri, makine öğrenmesi projelerinde daha doğru ve güvenilir sonuçlar elde etmek için vazgeçilmezdir. Bu süreçlerin uygulanmasının, gelecekte yapacağım projelerde elde edeceğim sonuçları iyileştireceğini ve genel performansı artıracağını öğrendim.

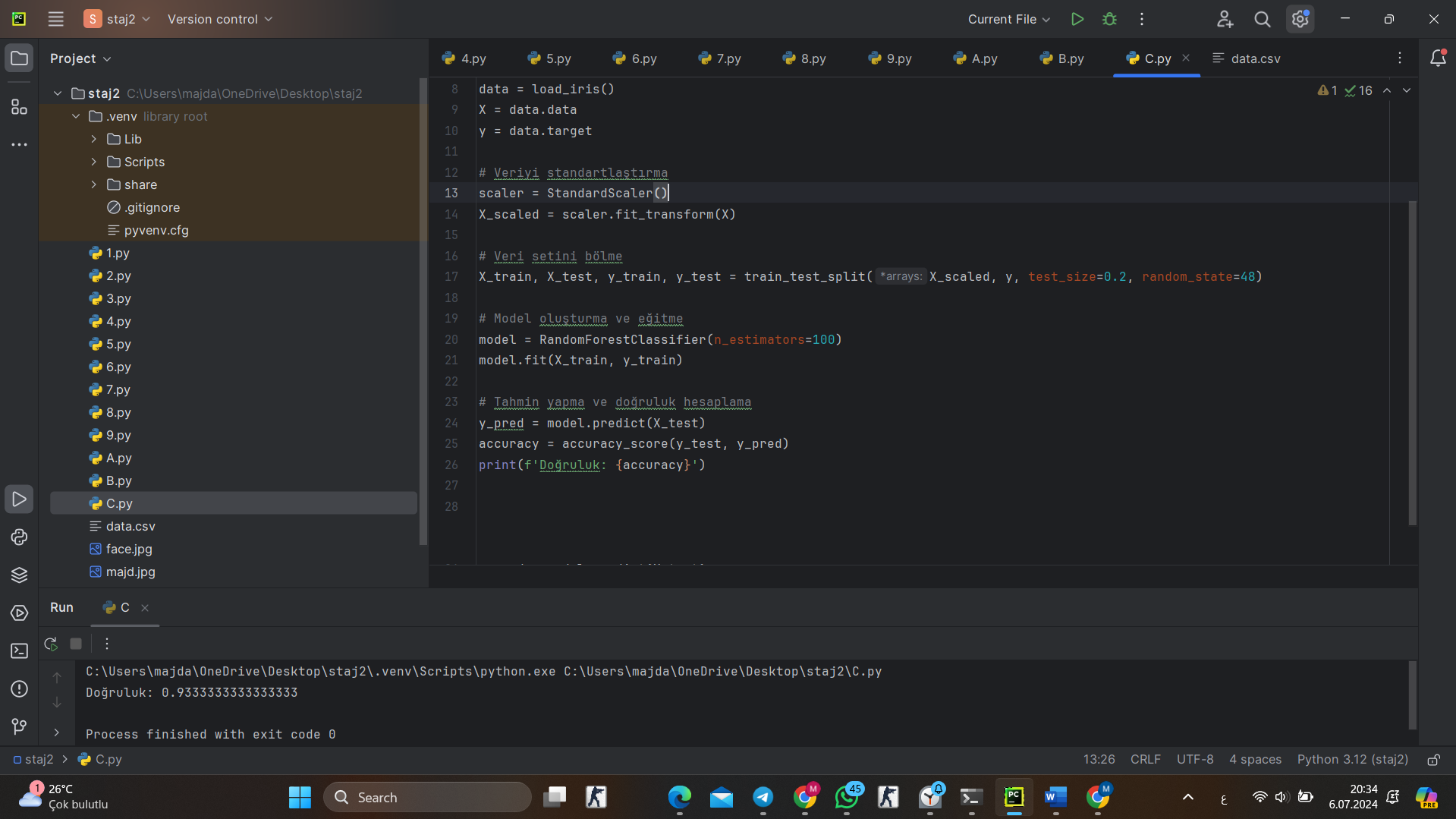
|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

### ****Denetimli Öğrenme Algoritmaları.****

Dördüncü günümde, denetimli öğrenme algoritmalarının temellerini ve uygulamalarını derinlemesine inceledik. Karar ağaçları, rastgele orman (random forest), k-en yakın komşu (k-NN) ve lojistik regresyon gibi yaygın olarak kullanılan denetimli öğrenme algoritmalarını ele aldık. Bu algoritmaların nasıl çalıştığı, hangi durumlarda kullanılabileceği ve avantajları hakkında detaylı bilgi edinme fırsatı buldum. Her bir algoritmanın temel prensipleri, matematiksel altyapısı ve uygulanabilirlikleri üzerine yoğunlaşıldı. Teorik bilgilerin yanı sıra, bu algoritmaların Python'da nasıl kullanılabileceği konusunda pratik örnekler yapıldı.

**Örnek Python Kodu: Random Forest**



Şekil 5Random Forest

**Açıklama:**

1. **Kütüphanelerin İçe Aktarılması:** Eğitimde, sklearn.ensemble kütüphanesinden RandomForestClassifier, sklearn.model\_selection kütüphanesinden train\_test\_split ve sklearn.metrics kütüphanesinden accuracy\_score sınıflarını içe aktardık. Bu sınıflar, model oluşturma, veri bölme ve performans değerlendirme işlemleri için gereklidir.
2. **Veri Setini Bölme:** Eğitim süresince, train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42) komutunu kullanarak ölçeklendirilmiş veriyi %80 eğitim, %20 test olarak ayırdık. Bu işlem, modelin eğitim ve test setlerinde farklı verilerle çalışmasını sağlar. random\_state parametresi ise sonuçların tekrar üretilebilir olmasını sağlar.
3. **Model Oluşturma ve Eğitme:** RandomForestClassifier(n\_estimators=100) sınıfını kullanarak bir random forest modeli oluşturduk. n\_estimators parametresi, ormanda kaç tane karar ağacı olduğunu belirler. fit metodu ile bu modeli eğitim verileri üzerinde eğiterek, modelin veri setindeki desenleri öğrenmesini sağladık.
4. **Tahmin Yapma ve Doğruluk Hesaplama:** predict metodu ile test verileri üzerinden tahminler yaptık ve bu tahminleri y\_pred değişkenine atadık. Ardından, accuracy\_score(y\_test, y\_pred) komutunu kullanarak modelin doğruluğunu hesapladık ve sonucu ekrana yazdırarak, modelin performansını değerlendirdik.

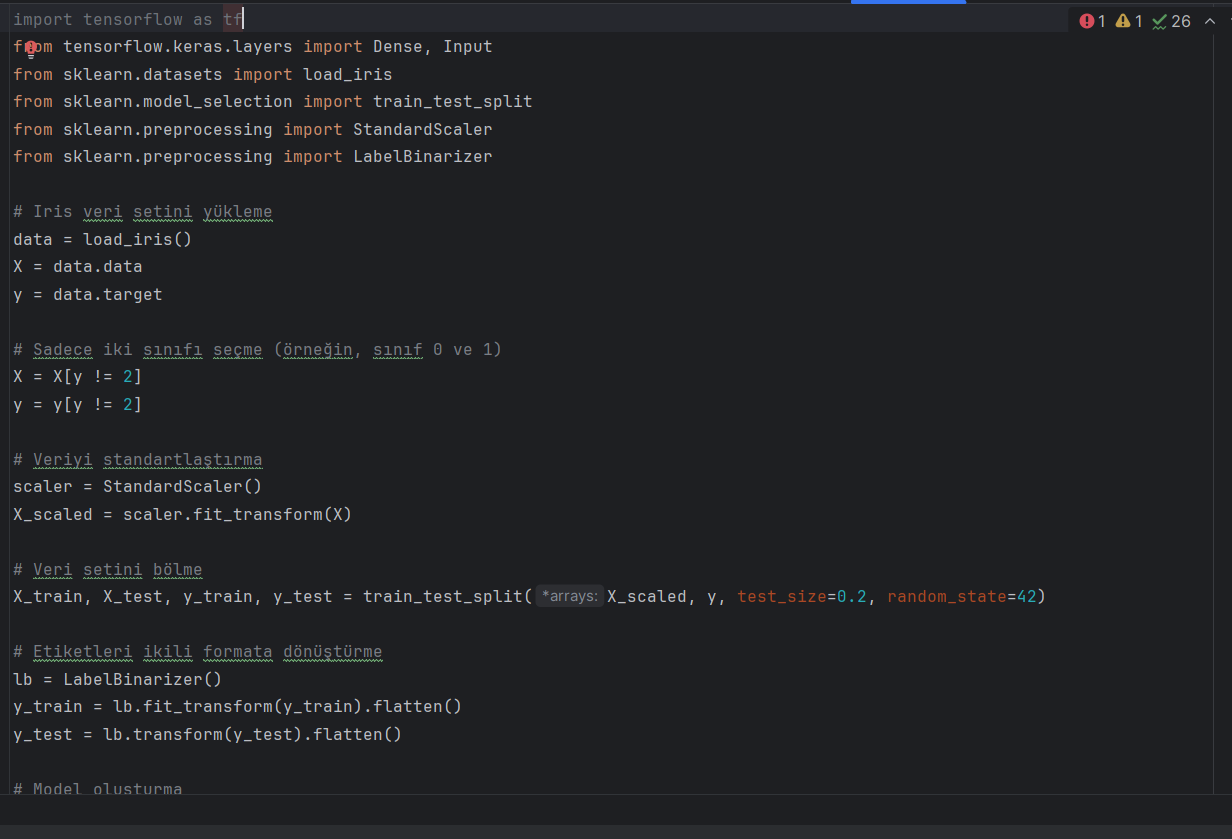
**Öğrendiklerimiz:** Dördüncü gün sonunda, denetimli öğrenme algoritmalarının makine öğrenmesi projelerinde ne kadar etkili olduğunu öğrendim. Karar ağaçları, rastgele orman, k-en yakın komşu ve lojistik regresyon gibi algoritmaların, farklı veri türleri ve problemler için nasıl kullanılabileceğini anladım. Ayrıca, bu algoritmaların performansını artırmak için veri setlerinin nasıl bölüneceği ve doğru bir şekilde nasıl değerlendirileceği konusunda bilgi sahibi oldum. Bu bilgiler, gelecekteki projelerimde denetimli öğrenme algoritmalarını daha etkili bir şekilde kullanmamı sağlayacaktır.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

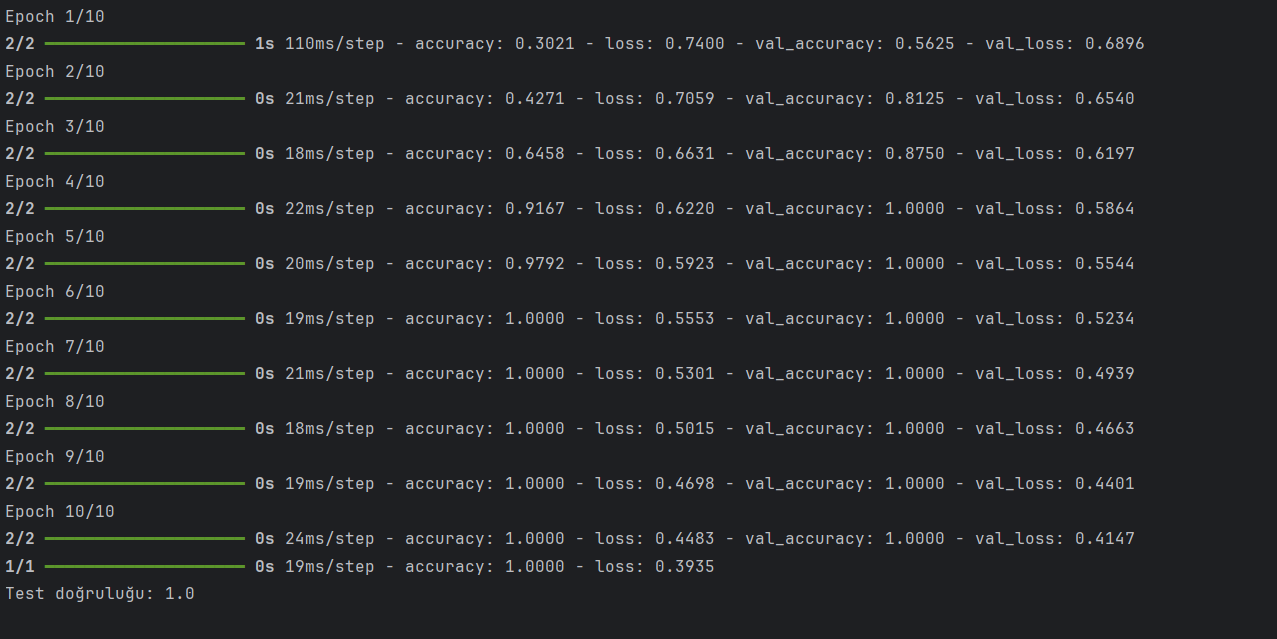
### ****Derin Öğrenme Temelleri.****

Beşinci günümde, derin öğrenme kavramlarına ve yapay sinir ağlarına giriş yaptık. Derin öğrenmenin temelleri, yapay sinir ağlarının yapısı ve işleyişi hakkında detaylı bilgi edindim. Bu kapsamda, popüler derin öğrenme kütüphaneleri olan Keras ve TensorFlow tanıtıldı. Eğitim sırasında, bu kütüphanelerin nasıl kullanılacağı ve basit bir sinir ağı modelinin nasıl oluşturulacağı üzerinde duruldu.

**Örnek Python Kodu: Basit Sinir Ağı**



Şekil 6İkili Sınıflandırma için Yapay Sinir Ağı Modeli



Şekil 7İkili Sınıflandırma için Yapay Sinir Ağı Modeli Çıkışı

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

**Açıklama:**

1. **Kütüphanelerin İçe Aktarılması:** Derin öğrenme modellerini oluşturmak için TensorFlow kütüphanesi kullanıldı. Veri setini yükleme, standartlaştırma ve eğitim/test setlerine ayırma işlemleri için gerekli kütüphaneler kullanıldı.
2. **Veri Setini Yükleme:** Iris veri setini yükleyerek çiçeklerin çeşitli özelliklerini ve türlerini içeren verilerle çalıştık.
3. **Sadece İki Sınıfı Seçme:** İkili sınıflandırma problemi oluşturmak için veri setinden sadece iki sınıf seçtik.
4. **Veriyi Standartlaştırma:** Özelliklerin ortalamasını 0 ve standart sapmasını 1 yaparak veriyi standartlaştırdık.
5. **Veri Setini Bölme:** Veriyi eğitim ve test setlerine ayırarak modelin eğitim ve değerlendirme süreçlerini gerçekleştirdik.
6. **Etiketleri İkili Formata Dönüştürme:** Etiketleri ikili formata dönüştürerek sınıflandırma modelinin eğitimini kolaylaştırdık.
7. **Model Oluşturma:** Bir sinir ağı modeli oluşturduk ve bu modelde gizli katmanlar ile çıkış katmanını ekledik. İkili sınıflandırma için sigmoid aktivasyon fonksiyonunu kullandık.
8. **Modeli Derleme:** Modeli derleyerek, optimizasyon algoritması ve kayıp fonksiyonunu belirledik ve doğruluk metriği ekledik.
9. **Modeli Eğitme:** Modeli eğitim verileri ile eğittik ve doğrulama setini kullanarak modelin performansını izledik.
10. **Modeli Değerlendirme:** Modelin test verileri üzerindeki performansını değerlendirerek doğruluk değerini inceledik.

**Öğrendiklerimiz:** Derin öğrenme ve yapay sinir ağlarının temel prensiplerini öğrendim. TensorFlow ve Keras ile bir sinir ağı modelinin oluşturulması ve eğitilmesi konusunda deneyim kazandım. Bu süreç, veriyi hazırlama, model oluşturma ve performans değerlendirme konularında önemli bilgiler ve beceriler kazandırdı.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

### ****Doğal Dil İşleme (NLP) Temelleri.****

Altıncı günümde, Doğal Dil İşleme (NLP) kavramlarına derinlemesine bir giriş yaptık. Metin verilerini nasıl işleyebileceğimizi, metin temizleme ve ön işleme tekniklerini inceledik. Bu süreçte, çeşitli örneklerle NLP'nin nasıl uygulandığını ve temel NLP modellerinin nasıl çalıştığını öğrendik.

**Örnek Python Kodu: Metin Ön İşleme**



Şekil 8 Metin Ön İşleme

**Açıklama:**

1. **Kütüphanelerin İçe Aktarılması:** NLP işlemleri için gerekli olan NLTK (Natural Language Toolkit) kütüphanesinin kullanımıyla başladık. Bu kütüphane, metin işleme ve analiz için gereken fonksiyonları sağlar.
2. **Gerekli NLTK Paketlerini İndirme:** Metin işleme işlemlerinde kullanılacak veri setlerini (örneğin, noktalama işaretleri ve durdurma kelimeleri) indirdik. Bu paketler, metinlerin doğru şekilde analiz edilmesi için gereklidir.
3. **Metin Örneği:** İşlenecek metni belirledik: "OpenAI is creating amazing artificial intelligence." Bu metin üzerinde işlemler gerçekleştirdik.
4. **Tokenizasyon:** Metni kelimelere bölerek tokenlara dönüştürdük. Tokenizasyon, metni küçük parçalara ayırarak daha detaylı bir analiz yapmamızı sağlar.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

1. **Küçük Harfe Çevirme ve Durdurma Kelimeleri Kaldırma:** Kelimeleri küçük harfe dönüştürdük ve sadece alfanümerik karakterleri içeren kelimeleri seçtik. Bu adım, metindeki gereksiz kelimeleri temizleyerek daha anlamlı veriler elde etmemizi sağlar.
2. **Lemmatizasyon:** Kelimeleri köklerine indirgedik. Lemmatizasyon, kelimelerin farklı biçimlerinin tek bir kök biçime dönüştürülmesini sağlar ve bu, metin analizinin doğruluğunu artırır.
3. **İşlenmiş Kelimeleri Yazdırma:** Sonuçları gözlemlemek için işlenmiş kelimeleri ekrana yazdırdık. Bu, ön işleme sürecinin etkilerini ve doğruluğunu değerlendirmemize yardımcı oldu.

**Öğrendiklerimiz:**

Bu gün boyunca, Doğal Dil İşleme (NLP) alanında önemli bir bilgi birikimi edindim. Ön işleme sürecinde metin temizliğinin ne kadar kritik olduğunu ve her adımın model performansını nasıl etkilediğini fark ettim.

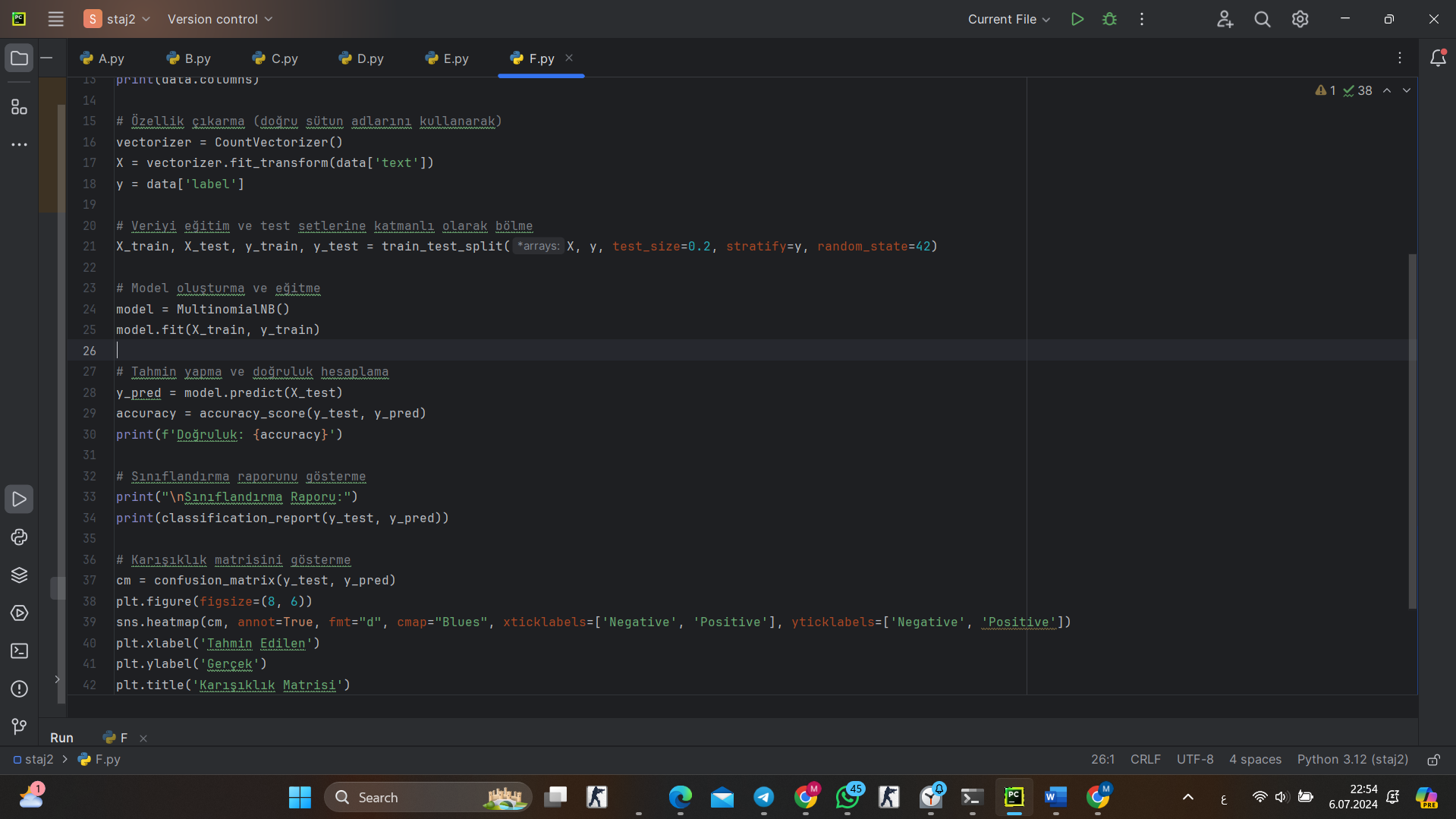
Metin temizleme aşamasında, verileri nasıl düzenleyeceğimizi ve gereksiz bilgileri nasıl ayıklayacağımızı öğrendim. Tokenizasyon sayesinde metinleri küçük parçalara ayırarak daha anlamlı analizler yapabileceğimizi gördüm. Küçük harfe çevirme ve durdurma kelimelerini kaldırma işlemleri, metni standart bir biçime getirmede ne kadar etkili olduğunu gösterdi. Lemmatizasyon ise, kelimelerin köklerine dönüştürülmesiyle, verinin tutarlılığını ve analiz edilebilirliğini artırmanın önemini kavradım.

Bu süreç, metin verileri ile çalışma konusunda sağlam bir temel oluşturarak, NLP tekniklerinin nasıl uygulanacağını ve bu tekniklerin veri analizi ve modelleme süreçlerinde nasıl kullanılabileceğini anlamama yardımcı oldu. Bu bilgiler, metin verilerini daha iyi analiz edebilmem ve etkili NLP çözümleri geliştirebilmem için büyük bir adım oldu.

### ****Sentiment Analizi Projesi.****

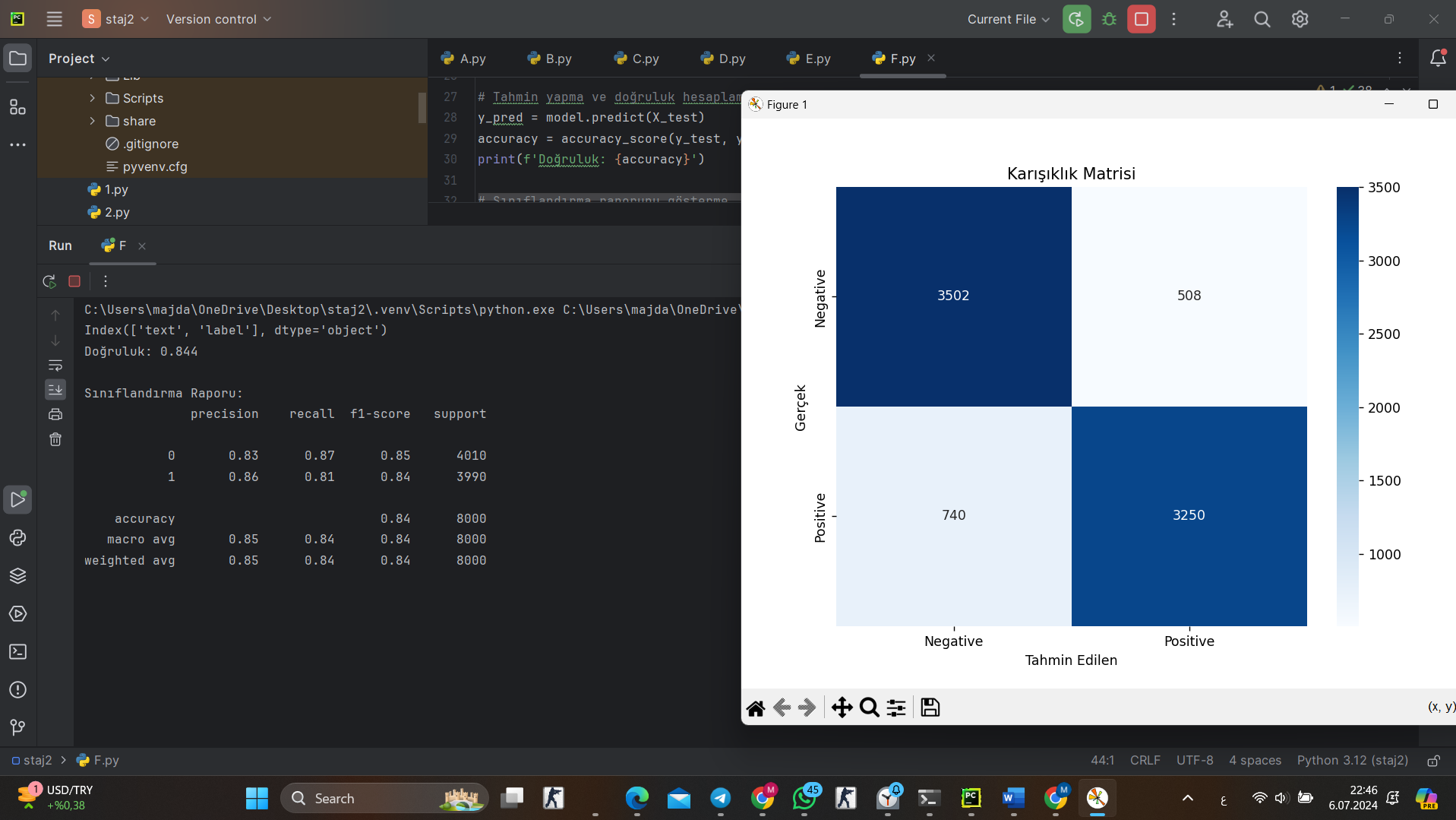
Yedinci günümde, sentiment analizi (duygu analizi) üzerine bir proje gerçekleştirdik. Bu proje kapsamında, film incelemeleri veri setini kullanarak metin verilerinin duygu durumunu belirlemeyi hedefledik. Sentiment analizi, kullanıcı yorumlarını pozitif ya da negatif olarak sınıflandıran bir tekniktir. Proje sürecinde, veri işleme, özellik çıkarma, model oluşturma ve değerlendirme adımlarını uyguladık.

**Örnek Python Kodu: Sentiment Analizi**



Şekil 9Sentiment Analizi

Bu koddan beklenen çıktıdır :



Şekil 10Sentiment Analizi Çıkışı

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

**Açıklama:**

1. **Kütüphanelerin İçe Aktarılması:** Veri yükleme ve işleme için pandas, metin verilerini sayısal verilere dönüştürmek için CountVectorizer, veri setlerini eğitim ve test olarak ayırmak için train\_test\_split, model oluşturmak için MultinomialNB ve model değerlendirmek için accuracy\_score, classification\_report, confusion\_matrix ile görselleştirme için matplotlib ve seaborn kütüphanelerini kullandık.
2. **Veriyi Yükleme:** Film incelemeleri veri setini yükleyerek analiz için hazır hale getirdik.
3. **Sütun Adlarını Kontrol Etme:** Veri setindeki sütun adlarını kontrol ederek, verinin yapısını ve içeriğini doğruladık.
4. **Özellik Çıkarma:** Metin verilerini sayısal verilere dönüştürerek, bu veriler üzerinde model oluşturulacak özellikleri belirledik.
5. **Veriyi Eğitim ve Test Setlerine Bölme:** Veriyi eğitim ve test setlerine ayırarak, modelin doğruluğunu test edebilmek için veriyi iki bölüme böldük. Eğitim seti, modelin öğrenmesini sağlarken, test seti modelin performansını değerlendirmemize yardımcı oldu.
6. **Model Oluşturma ve Eğitme:** Naive Bayes sınıflandırıcısı kullanarak bir model oluşturduk ve eğitim verileri üzerinde eğittik.
7. **Tahmin Yapma ve Doğruluk Hesaplama:** Modelin test verileri üzerindeki tahminlerini yaptık ve doğruluğunu hesapladık. Bu, modelimizin ne kadar başarılı olduğunu gösterdi.
8. **Sınıflandırma Raporunu Gösterme:** Modelin performansını değerlendiren bir rapor hazırladık. Bu rapor, modelin precision, recall ve F1-score gibi performans metriklerini içerir.
9. **Karışıklık Matrisini Gösterme:** Modelin tahmin sonuçlarını görselleştirerek, gerçek ve tahmin edilen sınıflar arasındaki doğru ve yanlış tahminleri inceledik. Bu matris, modelin güçlü ve zayıf yönlerini belirlemekte faydalı oldu.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

**Öğrendiklerimiz:**

Sentiment analizi projesi sırasında, metin verilerini duygu durumuna göre sınıflandırma sürecinin temel prensiplerini öğrendim. Veri setlerinin işlenmesi ve metin verilerinin sayısal verilere dönüştürülmesi süreçleri hakkında bilgi edindim. Özellikle, metinlerin sayısal özelliklere dönüştürülmesi ve bu verilerden sınıflandırma modeli oluşturulması sürecini kavradım.

Model performansını değerlendirmek için kullanılan metriklerin, doğruluk, precision, recall ve F1-score'un her birinin model başarısını nasıl ölçtüğünü anlamış oldum. Karışıklık matrisinin, modelin hangi sınıflarda daha başarılı olduğunu ve hangi sınıflarda zayıf kaldığını görselleştirme açısından ne kadar faydalı olduğunu gözlemledim.

Bu proje, sentiment analizi tekniklerini uygulama ve değerlendirme konusunda değerli deneyimler kazanmama yardımcı oldu. Ayrıca, metin verileri ile çalışmanın, modelleme sürecinde önemli rol oynadığını ve doğru tekniklerin seçilmesinin başarıyı nasıl artırabileceğini fark ettim.

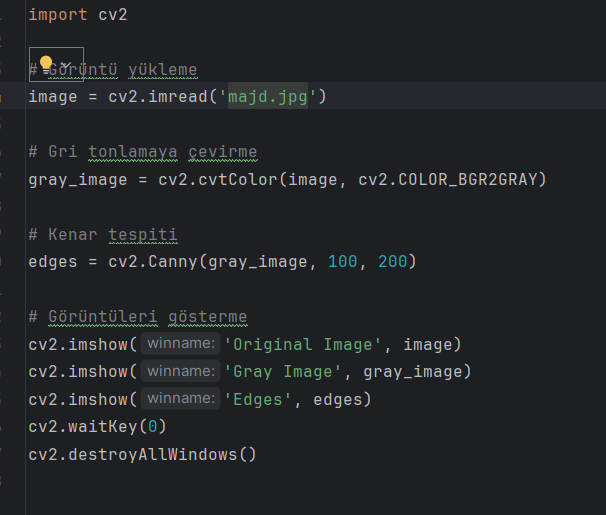
|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

### ****Görüntü İşleme Temelleri.****

Sekizinci günümde, görüntü işleme kavramlarını ve temel tekniklerini öğrendik. OpenCV kütüphanesi kullanarak görüntü işleme uygulamaları gerçekleştirdik. Bu süreçte, görüntülerin nasıl yükleneceği, gri tonlamaya nasıl dönüştürüleceği ve kenar tespiti gibi işlemlerin nasıl yapılacağı hakkında detaylı bilgi edindik.

**Örnek Python Kodu: Görüntü İşleme**



Şekil 11Görüntü İşleme



Şekil 12Görüntü İşleme Çıkışı

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

**Açıklama:**

* **Kütüphanelerin İçe Aktarılması:** Görüntü işleme işlemleri için OpenCV kütüphanesini kullandık. Bu kütüphane, görüntüleri yüklemek, işlemek ve analiz etmek için gerekli fonksiyonları sağlar.
* **Görüntü Yükleme:** Görüntü dosyasını yükleyerek işleme sürecine başladık. Bu adım, görüntülerin dijital formatta işlenmesi için gereklidir.
* **Gri Tonlamaya Çevirme:** Renkli görüntüleri gri tonlamaya dönüştürdük. Bu işlem, görüntüdeki renk bilgilerini azaltarak, sadece yoğunluk bilgileriyle işlemleri basitleştirir.
* **Kenar Tespiti:** Gri tonlamalı görüntüde kenar tespiti gerçekleştirdik. Kenar tespiti, görüntüdeki önemli geçişleri ve yapıları belirlemekte kullanılır.
* **Görüntülerin Görselleştirilmesi:** Orijinal görüntü, gri tonlamalı görüntü ve kenar tespiti yapılan görüntüleri ayrı pencerelerde gösterdik. Bu, işlenen görüntülerin etkilerini ve doğruluğunu görsel olarak değerlendirmemizi sağlar.

**Öğrendiklerimiz:**

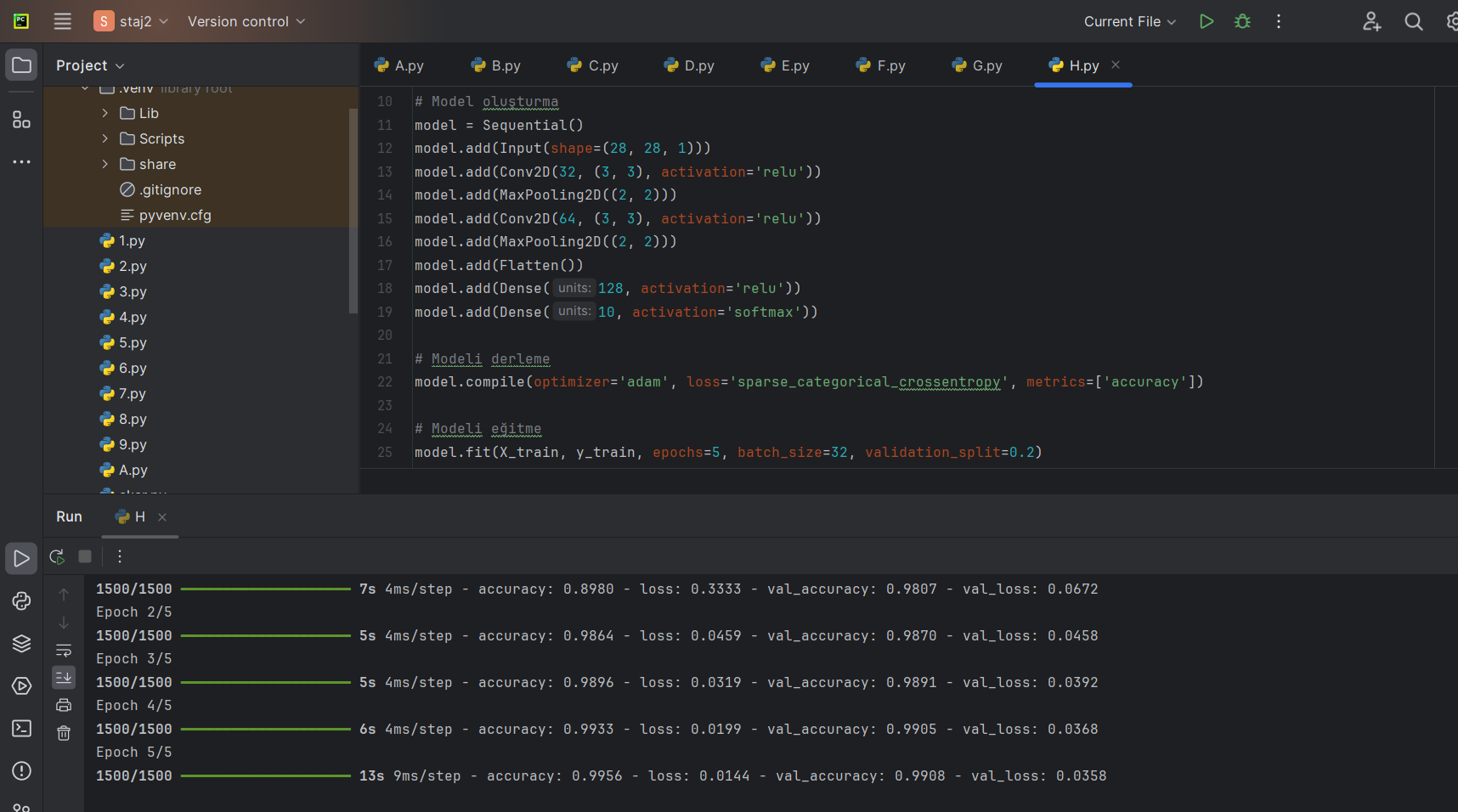
Görüntü işleme projesi sırasında, görüntülerin dijital olarak nasıl yüklendiğini ve işlendiğini öğrenmiş oldum. Özellikle, gri tonlamaya dönüştürme işleminin, renk bilgilerini azaltarak işleme sürecini nasıl basitleştirdiğini kavradım. Ayrıca, kenar tespitinin görüntülerdeki önemli yapıları belirlemede nasıl kritik bir rol oynadığını gözlemledim.

Görüntülerin farklı aşamalarda nasıl görselleştirildiği, işlenen verilerin etkilerini değerlendirmek açısından ne kadar faydalı olduğunu fark ettim. Bu proje, görüntü işleme tekniklerini uygulama konusunda önemli deneyimler kazanmama yardımcı oldu ve bu tekniklerin analiz süreçlerinde ne kadar etkili olduğunu anlamamı sağladı.

### ****Convolutional Neural Networks (CNN) ile Görüntü Sınıflandırma.****

Dokuzuncu günümde, derin öğrenme alanında önemli bir yer tutan Convolutional Neural Networks (CNN) kavramlarını detaylı bir şekilde öğrendim. Özellikle görüntü işleme ve sınıflandırma alanında geniş bir kullanım alanına sahip olan CNN'ler ile basit ama etkili bir görüntü sınıflandırma modeli geliştirdik. Bu uygulama için MNIST veri setini kullanarak, CNN'lerin pratikte nasıl çalıştığını ve veri üzerinde nasıl sonuçlar elde edilebileceğini deneyimleme fırsatı bulduk.

**Örnek Python Kodu: CNN ile Görüntü Sınıflandırma**



Şekil 13CNN ile Görüntü Sınıflandırma

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

**Açıklama:**

* **Kütüphanelerin İçe Aktarılması:** Derin öğrenme modelleri oluşturmak için TensorFlow'un Keras API'si kullanıldı. MNIST veri seti, el yazısı rakam görüntülerini içeren bu popüler veri seti, model eğitimi için uygun hale getirilir. CNN modelini oluşturmak için Sequential, Conv2D, MaxPooling2D, Flatten ve Dense katmanları kullanıldı.
* **Veri Setini Yükleme:** MNIST veri seti, el yazısı rakamların siyah beyaz görüntülerini içeren bir veri kümesidir. Bu veri seti, eğitim ve test verileri olarak ikiye ayrılarak modelin öğrenmesi ve değerlendirilmesi için hazır hale getirildi.
* **Veri Setini Yeniden Şekillendirme ve Ölçeklendirme:** Eğitim ve test verileri, CNN modeline uygun şekilde yeniden şekillendirildi. Görüntüler 28x28 piksel boyutunda gri tonlamalı hale getirildi ve [0, 1] aralığına ölçeklendi. Bu işlem, modelin verilerle daha iyi performans göstermesi için gereklidir.
* **Model Oluşturma:** CNN modelinin yapısı şu şekilde belirlendi:
  + **Conv2D Katmanları:** Görüntüdeki özellikleri çıkarmak için evrişimsel katmanlar eklendi. İlk Conv2D katmanı 32 filtre ve 3x3 kernel boyutuna sahipken, ikinci Conv2D katmanı 64 filtre ve aynı kernel boyutunu kullandı.
  + **MaxPooling2D Katmanları:** Özellik haritalarını azaltarak modelin daha az veri ile çalışmasını sağladı. Her iki evrişimsel katmandan sonra 2x2 boyutunda maksimum havuzlama katmanları eklendi.
  + **Flatten Katmanı:** Çok boyutlu çıktıyı düz bir vektöre dönüştürdü, böylece tam bağlantılı katmanlar için uygun hale geldi.
  + **Dense Katmanları:** İlk Dense katmanı 128 nöron ve ReLU aktivasyon fonksiyonuna sahipken, çıkış katmanı 10 nöron ve softmax aktivasyon fonksiyonu kullandı.
* **Modeli Derleme:** Model, Adam optimizer, sparse categorical crossentropy kayıp fonksiyonu ve doğruluk metriği kullanılarak derlendi. Bu adım, modelin eğitilmesi için gerekli ayarları yapar.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

* **Modeli Eğitme:** Model, eğitim verileri üzerinde eğitildi ve doğrulama verileri kullanılarak modelin performansı izlendi. Eğitim süreci 5 epoch ve 32 batch boyutu ile gerçekleştirildi.

**Öğrendiklerimiz:**

CNN'ler, özellikle görüntü işleme ve sınıflandırma için etkili araçlardır. Bu proje sırasında, CNN modelinin nasıl oluşturulduğunu ve MNIST veri seti üzerinde nasıl eğitildiğini uygulamalı olarak öğrendim.

Modelin çeşitli katmanlarını anlamak ve yapılandırmak, görüntü verilerini işlemek için hangi tekniklerin kullanıldığını kavramama yardımcı oldu. Özellikle, evrişimsel katmanların ve havuzlama katmanlarının, görüntülerden anlamlı özellikler çıkarmadaki rolünü gözlemledim.

Veri setinin yeniden şekillendirilmesi ve ölçeklendirilmesi, modelin verilerle daha iyi uyum sağlamasını ve doğru sonuçlar üretmesini sağladı. Modelin performansını izlemek ve optimize etmek, derin öğrenme projelerinde başarıyı artırmak için kritik öneme sahiptir. Bu proje, CNN'lerin görüntü sınıflandırma görevlerinde ne kadar güçlü ve etkili olduğunu anlamamı sağladı.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

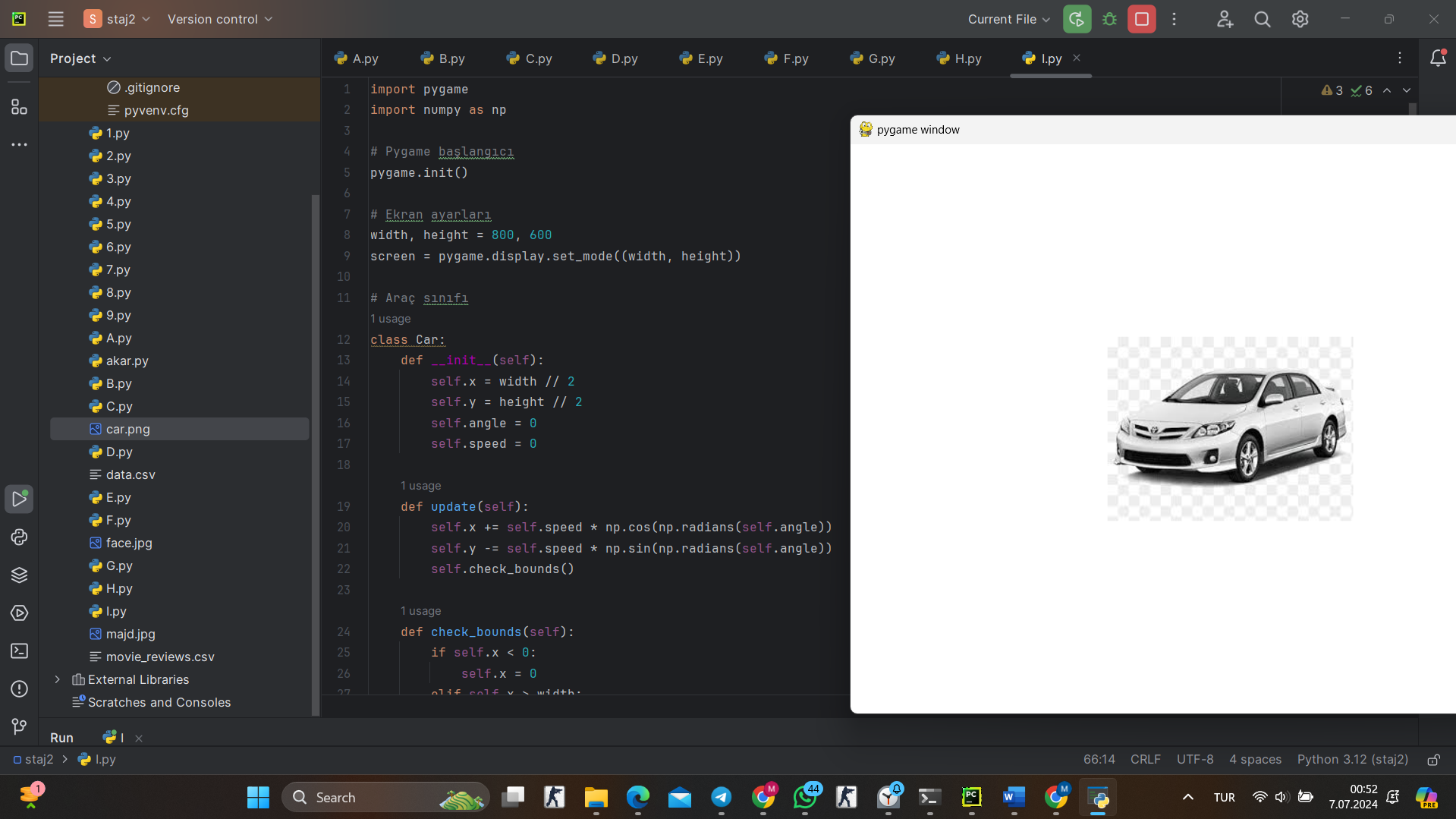
### Büyük Proje Başlangıcı: Otonom Araç Simülasyonu.

Onuncu günümde, otonom araç teknolojileri hakkında derinlemesine bilgi edinme fırsatı bulduk. Bu günün ana amacı, otonom araçların temel gereksinimlerini ve kullanılan teknolojileri anlamak ve bir simülasyon ortamı hazırlamaktı. Projemizin hedefi, otonom bir aracın çevresini algılayarak, engellerden kaçınarak ve belirli bir rotayı takip ederek hareket etmesini sağlamaktı.

**Proje Tanımı: Otonom Araç Simülasyonu**

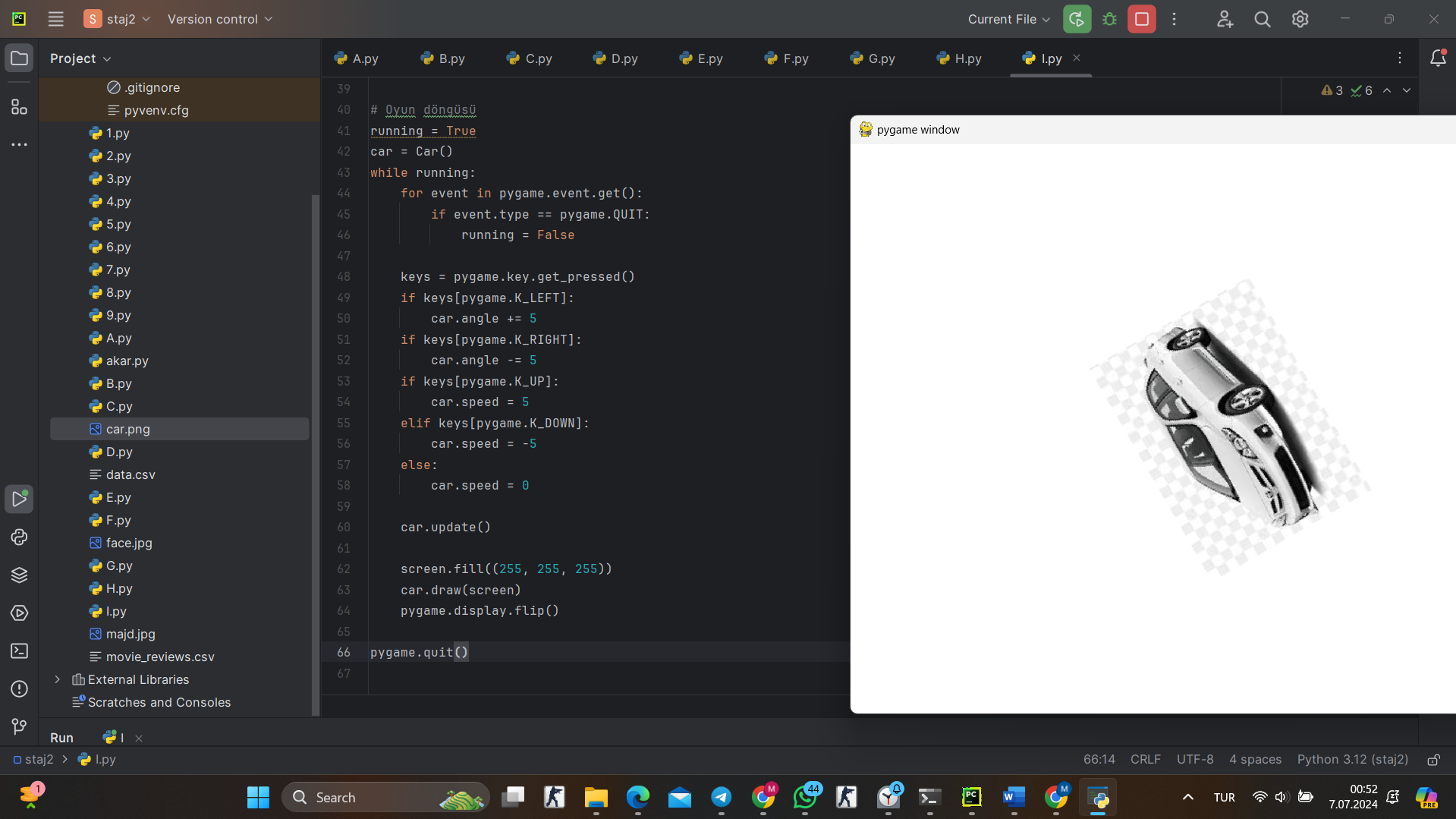
Bu projede, otonom bir aracın simülasyonunu yapmayı hedefliyoruz. Araç, çevresini algılayarak, engellerden kaçınarak ve belirli bir rotayı takip ederek hareket edecek. Proje kapsamında, sensör verilerinin işlenmesi, yol planlaması ve kontrol algoritmalarının geliştirilmesi üzerine çalışmalar yapılacaktır. Bu simülasyon, aracın gerçek dünya senaryolarında nasıl davranabileceğini anlamamıza yardımcı olacak.

**Örnek Python Kodu: Basit** **Otonom Araç Simülasyonu**



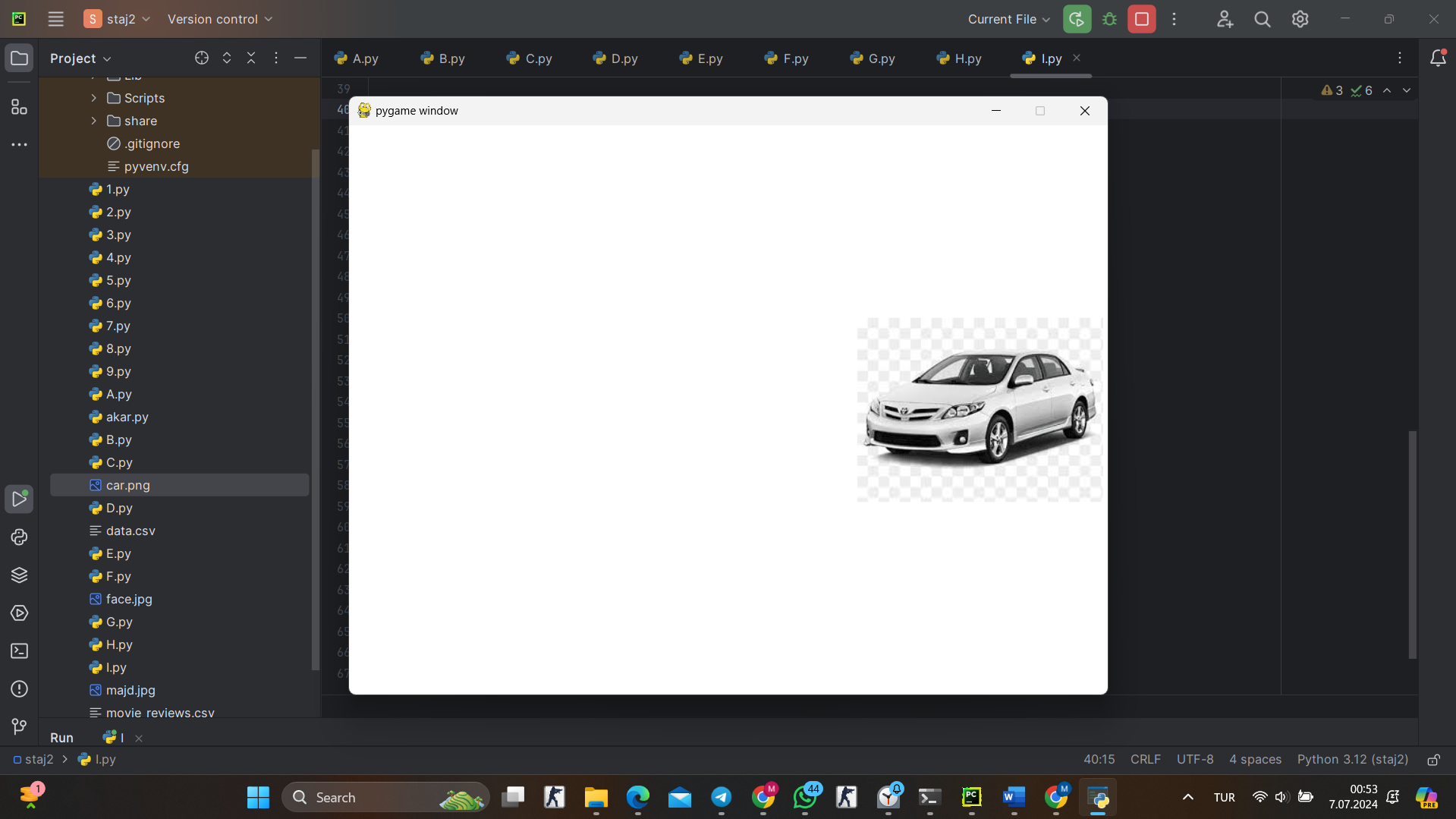
Şekil 14Otonom Araç Simülasyonu

Kullanıcı sol veya sağ ok tuşuna bastığında, aracın açısı değişir.



|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

Kullanıcı yukarı ok tuşuna bastığında, araç 5 hızında ileri hareket eder.



**Öğrendiklerimiz:**

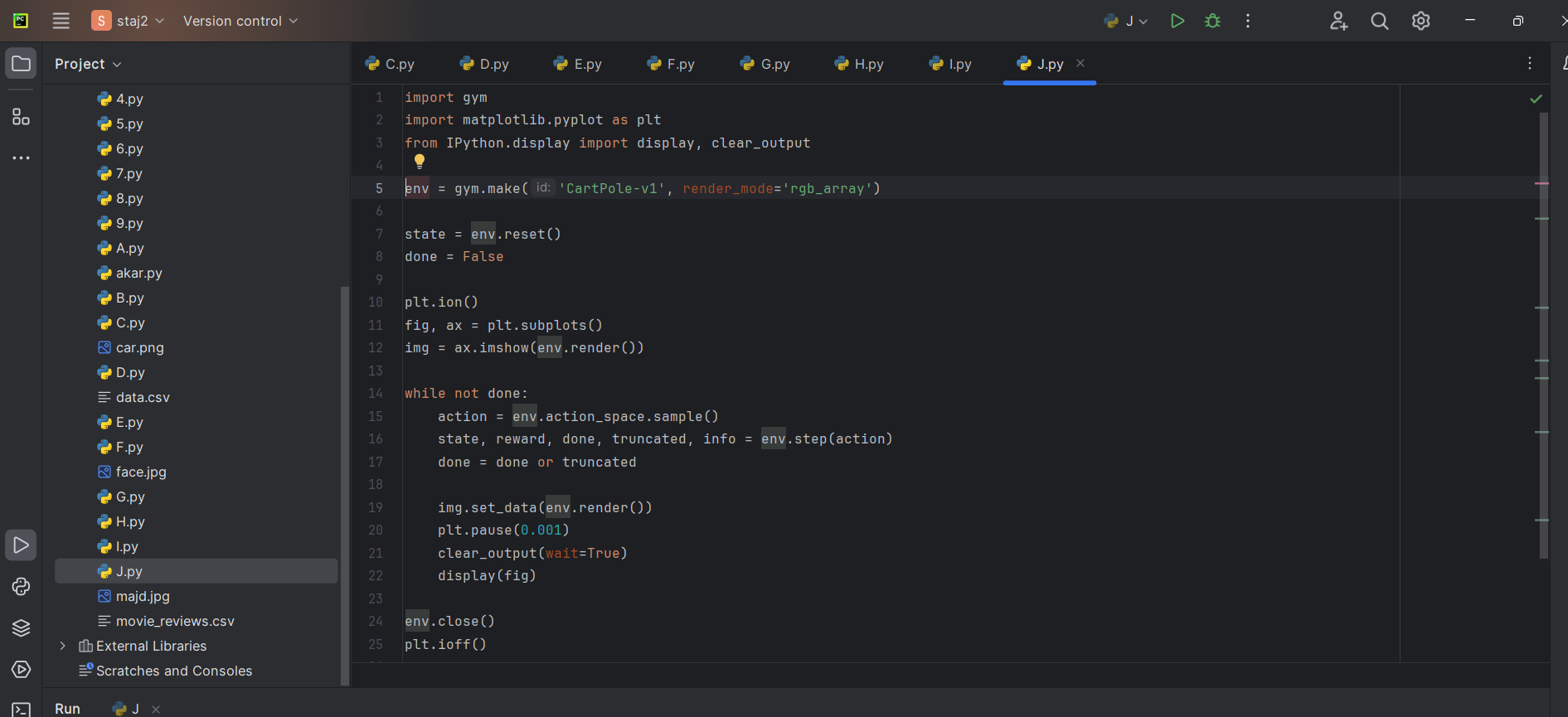
1. **Otonom Araç Teknolojileri ve Gereksinimleri:** Otonom araçların temel gereksinimlerini ve kullanılan teknolojileri öğrendik. Bu, araçların çevresini algılaması, engelleri tanımlaması ve yol bulma yeteneklerini içerir.
2. **Simülasyon Ortamının Hazırlanması:** Pygame kullanarak bir simülasyon ortamı oluşturmayı öğrendik. Bu, araç hareketlerini görselleştirmek ve simülasyonun temelini atmak için önemli bir adımdı.
3. **Sensör Verilerinin İşlenmesi:** Sensör verilerinin nasıl işleneceğini anlamak, otonom araçların çevre algılama yeteneklerini geliştirmek için kritik bir adımdır. Bu veriler, aracın çevresindeki engelleri ve diğer nesneleri tanımlamak için kullanılacak.
4. **Yol Planlaması ve Kontrol Algoritmaları:** Otonom araçların yol bulma ve kontrol stratejilerini geliştirmek için gerekli algoritmaların temel prensiplerini öğrendik. Bu, aracın belirli bir rotayı takip etmesini ve engellerden kaçınmasını sağlayacak.

Bu gün, otonom araç simülasyonunun temel taşlarını oluşturarak, gelecekteki gelişmeler ve uygulamalar için sağlam bir zemin hazırladık. Projenin ileri aşamalarında, sensör verileri işleme, engel tespiti ve daha karmaşık yol planlama algoritmaları gibi konulara geçiş yaparak, simülasyonu daha da geliştireceğiz.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

### **Reinforcement Learning (Pekiştirmeli Öğrenme) Temelleri.**

11’inci günümde, Pekiştirmeli Öğrenme (Reinforcement Learning - RL) kavramlarına odaklandık ve bu alandaki temel algoritmaları inceledik. OpenAI Gym kullanarak basit bir pekiştirmeli öğrenme örneği gerçekleştirdik. Bu çalışma, pekiştirmeli öğrenme algoritmalarının nasıl uygulandığını ve temel prensiplerini anlamamı sağladı.



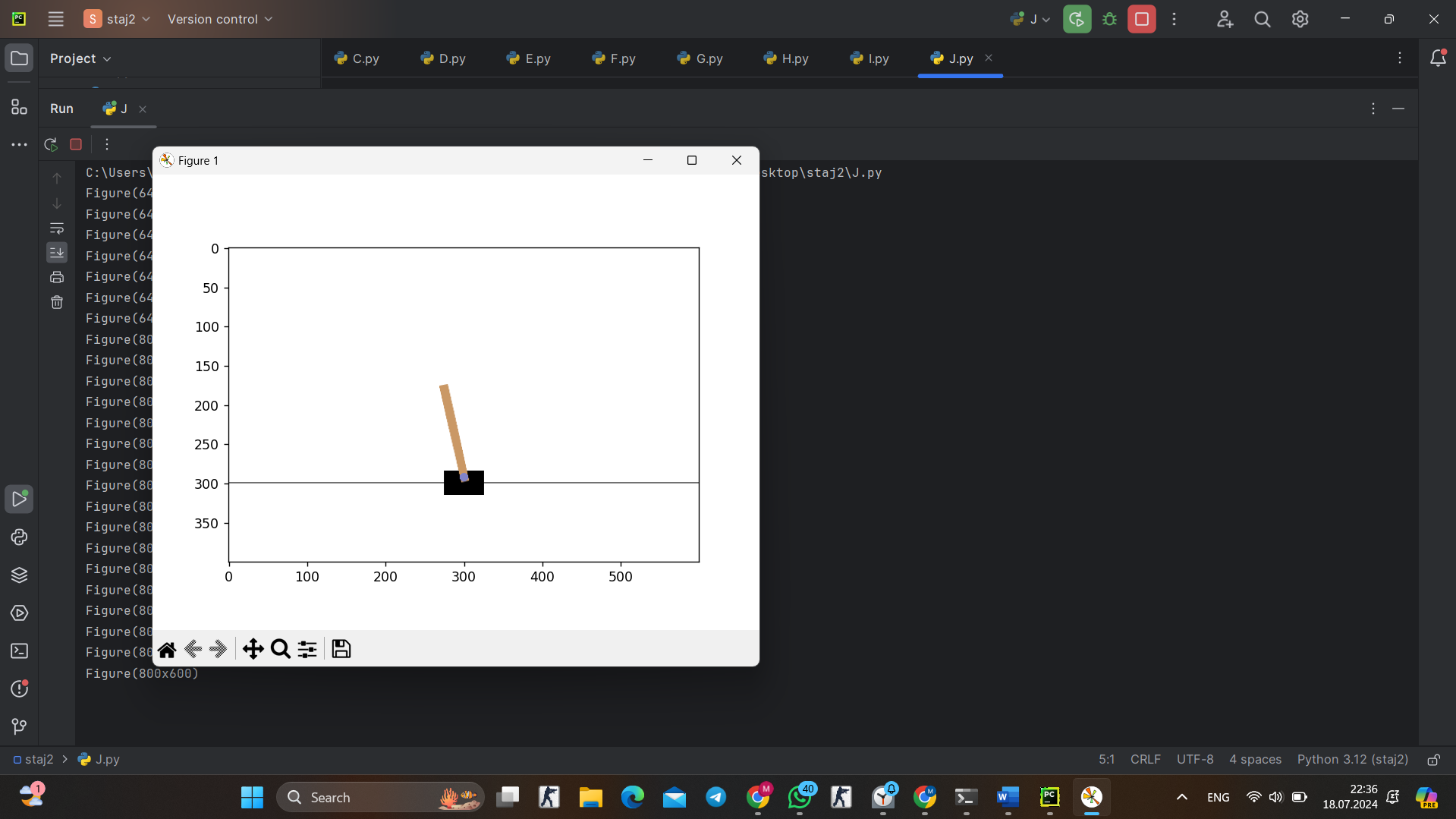
Şekil 15Pekiştirmeli Öğrenme

**Kodun Çalışma Mantığı:**

1. **Kütüphaneleri İçe Aktarma:**
   * gym kütüphanesi, pekiştirmeli öğrenme için çeşitli simülasyon ortamları sağlar. Bu kütüphane, farklı RL problemlerini çözmek için kullanılır.
   * matplotlib.pyplot, ortamın görselleştirilmesi için grafikler oluşturur. Bu sayede ortamın değişimini ve aracın hareketlerini görsel olarak takip edebiliriz.
   * IPython.display, Jupyter Notebook gibi etkileşimli ortamlarda görüntü ve grafiklerin dinamik olarak gösterilmesini sağlar.
2. **Ortamı Başlatma:**
   * Belirli bir RL ortamı başlatılır. Bu ortam, aracın öğrenme sürecinde kullanacağı simülasyon alanını oluşturur. Ortamın görüntüsü, renkli bir dizi olarak elde edilir, bu da simülasyonun görsel geri bildirimini sağlar.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

1. **Ortamı Sıfırlama:**
   * Simülasyonun başlangıç durumuna sıfırlanır. Bu adım, simülasyonun temiz bir başlangıç yapmasını ve aracın ilk pozisyonunu belirlemesini sağlar.
2. **Matplotlib ile Pencere Oluşturma:**
   * Görselleştirme amacıyla bir pencere oluşturulur. Bu pencere, ortamın her anını dinamik olarak göstermek için kullanılır. Ortamın ilk görüntüsü, bu pencere üzerinde gösterilir.
3. **Döngü ile Ortamı Güncelleme:**
   * Simülasyon döngüsü içinde, ortamın eylem alanından rastgele bir eylem seçilir. Bu eylem, simülasyondaki aracın hareketini veya davranışını temsil eder.
   * Seçilen eylem, ortamda gerçekleştirilir ve yeni durum, ödüller, tamamlanma durumu gibi bilgiler elde edilir.
   * Eğer simülasyon tamamlanır veya kesilirse döngüden çıkılır. Bu durumlar, simülasyonun sonlandırılmasını veya geçici bir durumu belirtir.
   * Ortamın güncel çerçevesi, pencere üzerinde dinamik olarak güncellenir ve görselleştirilir.
4. **Ortamı Kapatma:**
   * Simülasyon döngüsü tamamlandıktan sonra ortam kapatılır ve görselleştirme modu sonlandırılır. Bu adım, simülasyonun düzgün bir şekilde sonlandırılmasını ve kaynakların serbest bırakılmasını sağlar.



Şekil 16CartPole Oyun Animasyonu

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

**Öğrendiklerimiz:**

Bu süreç, pekiştirmeli öğrenme (RL) algoritmalarının temel çalışma prensiplerini ve uygulama yöntemlerini anlamamı sağladı. Ortamın nasıl başlatıldığı, eylemlerin nasıl seçildiği ve güncellenmiş durumların nasıl görselleştirildiği konularında bilgi sahibi oldum. Ayrıca, simülasyonun dinamik bir şekilde nasıl yürütüldüğünü ve sonuçların nasıl değerlendirildiğini gözlemledim. Bu deneyim, pekiştirmeli öğrenmenin nasıl işlediğini ve RL problemlerini çözmek için nasıl bir yaklaşım izlenmesi gerektiğini anlamamı sağla

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

### 

### ****Rasa Chatbot ve Docker Kullanımı****

Bu gün, Docker kullanarak Rasa chatbot'un kurulumunu ve çalıştırılmasını gerçekleştirdik. Aşağıda bu sürecin detayları ve karşılaşılan problemler ile çözümler açıklanmıştır.

**1. Docker ve Docker-Compose Kurulumu**

Açıklama: Docker, uygulamaları izole bir ortamda çalıştırmak için kullanılırken, Docker-Compose birden fazla Docker konteynerini tanımlayıp çalıştırmak için kullanılır. Bu adımda, Docker ve Docker-Compose'un bilgisayarınıza kurulu olduğundan emin olduk.

Adımlar:

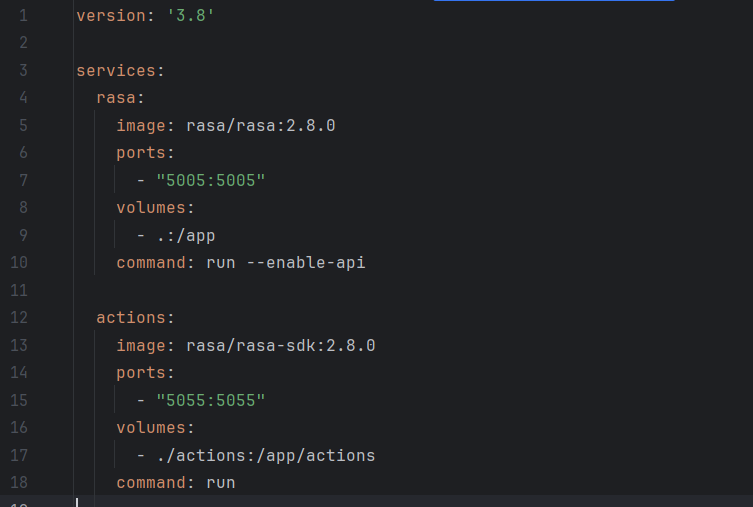
Docker ve Docker-Compose'un kurulumunun yapılması.

Kurulumdan sonra, Docker ve Docker-Compose komutları ile Rasa chatbot'unuzu oluşturma işlemlerine geçiş yapılabilir.

**2. Docker-Compose Dosyasını Oluşturma**

Açıklama: Docker-Compose dosyası, uygulamanızın nasıl yapılandırılacağını tanımlar. Bu dosyada, Rasa ve Rasa'nın eylem modülü için iki hizmet tanımlanır.

Örnek docker-compose.yml Dosyası:



Şekil 17docker-compose.yml Dosyası

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

**Kodun Çalışma Mantığı:**

version: '3.7': Docker Compose dosyasının sürümünü belirtir.

services: Docker konteynerlerinin tanımlandığı bölümdür.

rasa: Rasa chatbot'unun çalıştırılacağı konteyner.

image: Kullanılacak Docker imajını belirtir.

volumes: Yerel dosya sistemindeki dosyaların konteynere bağlanmasını sağlar.

command: Rasa komutlarını çalıştırır.

ports: Konteynerdeki portları bilgisayarınızın portlarına bağlar.

actions: Rasa'nın eylem modülü için konteyner.

image: Eylem modülü için kullanılacak Docker imajını belirtir.

volumes: Yerel dosyaların konteynere bağlanmasını sağlar.

ports: Konteyner portlarını bilgisayar portlarına bağlar.

**3. Docker-Compose ile Rasa ve Eylem Konteynerlerini Oluşturma**

Açıklama: Docker-Compose ile konteynerler oluşturulur ve yapılandırılır.

**Komut**:

docker-compose build

Kodun Çalışma Mantığı:

docker-compose build: Docker-Compose dosyasına göre belirtilen konteynerleri oluşturur ve yapılandırır.

**4. Rasa Modelini Başlatma ve Çalıştırma**

Açıklama: Rasa chatbot'unu başlatmak ve çalıştırmak için Docker-Compose kullanılır.

**Komut**:

docker-compose up

Kodun Çalışma Mantığı:

docker-compose up: Docker-Compose dosyasındaki tanımlamalara göre konteynerleri başlatır. Bu komut, hem Rasa hem de eylem konteynerlerini çalıştırır.

Gözlemler:

Rasa konteyneri başarıyla çalışmaya başladı.

Eylem konteyneri, çalışmayı tamamladıktan sonra çıkış kodu 0 ile bitti.

Rasa sunucusunun <http://localhost:5005> adresinde çalıştığı doğrulandı.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

**5. Rasa ile Test Etme**

Açıklama: Rasa chatbot'unu test etmek için HTTP POST isteği kullanılır.

Komut:

curl -X POST -H "Content-Type: application/json" -d '{"sender": "test\_user", "message": "hello"}' http://localhost:5005/webhooks/rest/webhook

Kodun Çalışma Mantığı:

curl -X POST: HTTP POST isteği gönderir.

-H "Content-Type: application/json": Gönderilen verinin JSON formatında olduğunu belirtir.

-d '{"sender": "test\_user", "message": "hello"}': Gönderilen JSON verisini içerir.

http://localhost:5005/webhooks/rest/webhook: Rasa'nın REST API'sine yapılan isteğin hedef adresidir.

Sonuç:

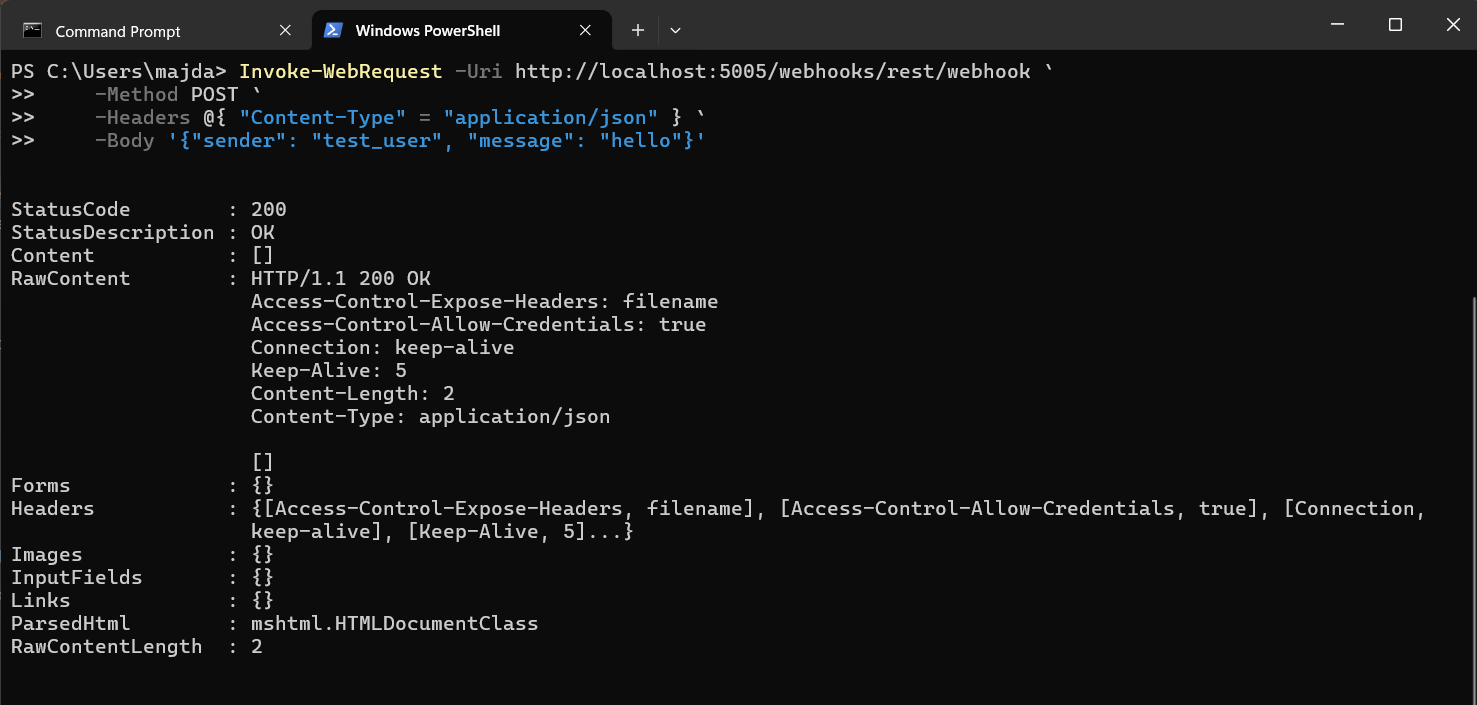
HTTP 200 OK yanıtı alındı, ancak içerik boş döndü. Bu durum, Rasa'nın doğru bir şekilde yapılandırılmadığını veya modelin eğitilmediğini gösterir.

Öneriler ve Çözümler:

Rasa'nın modelinin eğitildiğinden ve yapılandırıldığından emin olun.

Gerekirse rasa init ve rasa train komutları ile Rasa projesini baştan başlatın ve eğitin.

Bu süreçte karşılaşılan sorunlar ve çözümler, Docker ile Rasa kullanımı konusunda daha iyi bir anlayış geliştirmeye yardımcı olur.



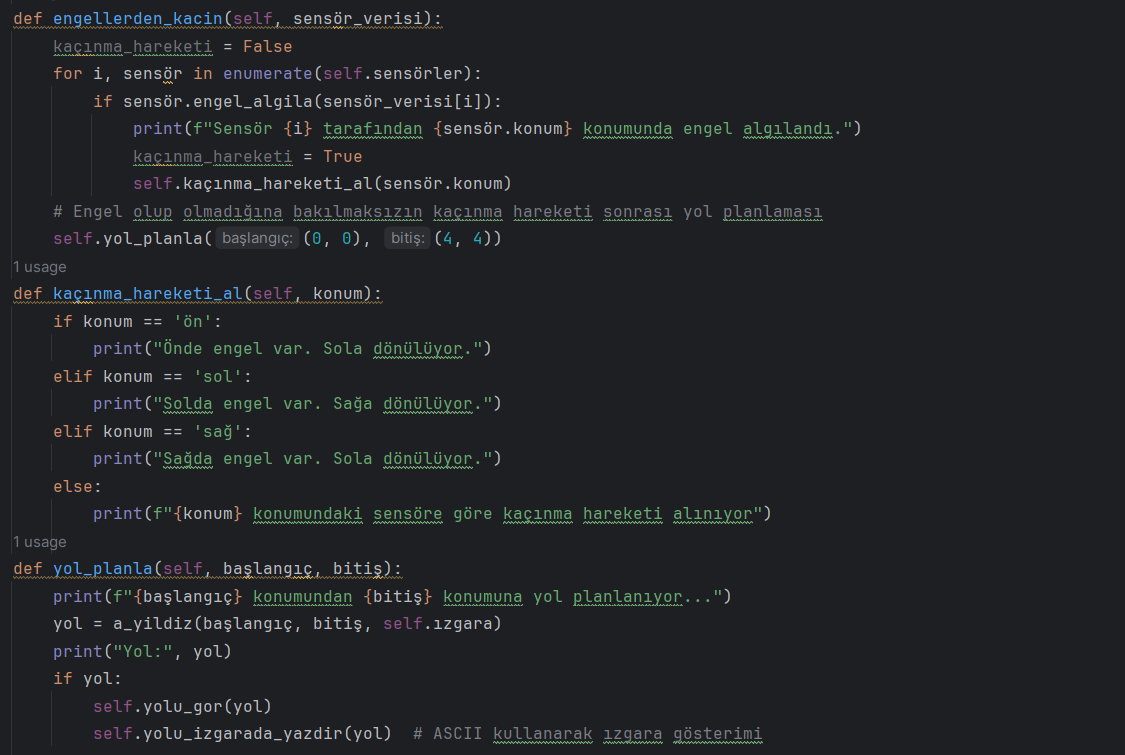
Şekil 18Rasa Chatbot Kodun Çalışması

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

### Otonom Araç Projesi

13. günümde, otonom araç teknolojileri üzerine çalışarak bir otonom aracın engelleri algılayarak nasıl hareket ettiğini simüle eden bir proje geliştirdik. Projemizde, ultrasonik sensörler ve A\* algoritması kullanarak bir araç modelledik. Araç, sensörlerinden aldığı verilerle çevresini tarayarak önüne çıkan engellerden kaçınabiliyor ve hedeflenen noktaya en kısa yolu planlayabiliyor. Bu proje, otonom araçların karar verme süreçlerini anlamaya yönelik önemli bir deneyim kazanmamızı sağladı.

**Proje Kodu: Otonom Araç ve Engel Algılama**



Şekil 19Otonom Araç ve Engel Algılama kodu

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

**Kodun Çalışma Mantığı:**

Bu proje, otonom araçların çevre algılama ve karar verme yeteneklerini basit bir düzeyde simüle eder. Projenin amacı, otonom araç teknolojileri üzerine pratik bir anlayış geliştirmek ve ilgili algoritmaların nasıl çalıştığını anlamaktır. İşte kodun temel çalışma mantığı:

1. **Ultrasonik Sensör Sınıfı**

**UltrasonikSensör Sınıfı:** Aracın çevresindeki engelleri algılamak için kullanılan sensörleri temsil eder. Her sensörün bir konumu vardır (örneğin, "ön", "sol", "sağ").

**engel\_algıla Metodu:** Sensörün algılama mesafesi belirli bir eşikten (1.0 metre) küçükse, engeli algıladığını belirtir.

1. **Otonom Araç Sınıfı**

**OtonomAraç Sınıfı:** Sensörler ve ızgara verileri ile otonom aracın davranışlarını kontrol eder.

* + **engellerden\_kaçın Metodu:** Sensörlerden gelen verileri kullanarak engelleri algılar. Bir engel algılandığında, **kaçınma\_eylemi\_uygula** metodunu çağırarak uygun kaçınma eylemini gerçekleştirir. Engel algılanmazsa, hedefe en kısa yolu planlamak için **yolu\_planla** metodunu çağırır.
  + **kaçınma\_eylemi\_uygula Metodu:** Algılanan engelin konumuna göre (ön, sol, sağ) uygun yön değişikliği yapar.
  + **yolu\_planla Metodu:** Başlangıç ve hedef konumları arasında A\* algoritmasını kullanarak en kısa yolu bulur. Yolu bulduktan sonra **yolu\_görüntüle** metodunu çağırarak yolu görselleştirir.
  + **yolu\_görüntüle Metodu:** Bulunan yolu matplotlib kullanarak görselleştirir ve ekran üzerinde gösterir. Ayrıca yolu bir dosya olarak kaydeder.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

1. ***A Algoritması*\***

**astar Fonksiyonu:** Başlangıç ve hedef konumları arasında en kısa yolu bulmak için A\* algoritmasını kullanır.

* + **Açık Liste:** Henüz işlenmemiş düğümleri tutar.
  + **Kapalı Liste:** İşlenmiş düğümleri tutar.
  + **Düğüm Sınıfı:** Bir konumu ve bu konumun ebeveynini tutar. Ayrıca, g, h ve f değerlerini hesaplar.

**A Hesaplamaları:**

* + **g:** Düğümün başlangıç konumuna olan mesafesini temsil eder.
  + **h:** Hedef konumuna olan tahmini mesafeyi temsil eder.
  + **f:** g ve h değerlerinin toplamını temsil eder.

1. **Kodun Çalışma Akışı**
   * **Sensörlerin Tanımlanması:** UltrasonikSensör sınıfından sensörler oluşturulur.
   * **Otonom Araç Nesnesi Oluşturulması:** Sensörler ve ızgara verileri ile OtonomAraç sınıfından bir araç nesnesi oluşturulur.
   * **Engellerin Algılanması:** Sensörlerden gelen veriler ile engeller kontrol edilir ve uygun kaçınma eylemi yapılır.
   * **Yol Planlaması:** Engel algılanmazsa, A\* algoritması kullanılarak hedefe en kısa yol hesaplanır.
   * **Yolun Görselleştirilmesi:** Bulunan yol matplotlib ile görselleştirilir ve bir dosya olarak kaydedilir.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |



Şekil 20Otonom Araç ve Engel Algılama kodu çıkışı

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

**Öğrendiklerimiz:**

Bu proje, otonom araçların çevresel engelleri algılayarak doğru bir şekilde hareket edebilmesini simüle etmekte önemli bir rol oynadı. Aşağıdaki konularda bilgi ve deneyim kazandık:

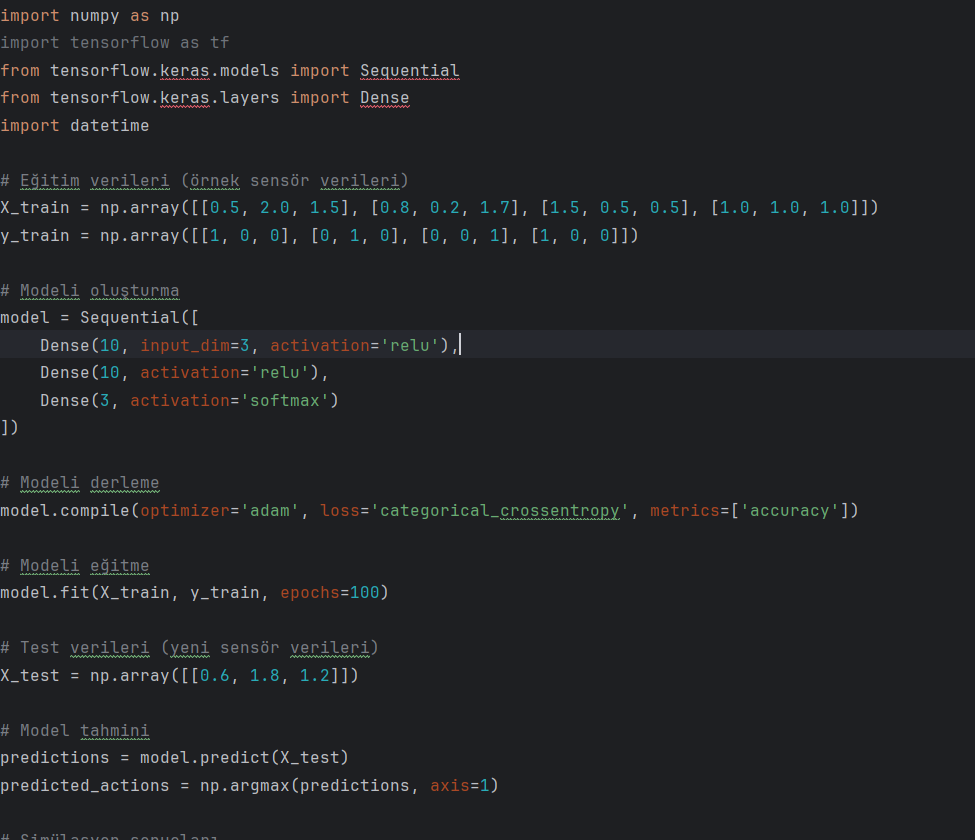
* **Ultrasonik Sensörler:** Engelleri algılamada kullanılan sensörlerin nasıl çalıştığını öğrendik.
* *A Algoritması:*\* Yolları planlama ve en kısa yolu bulma algoritmasının nasıl uygulandığını öğrendik.
* **Görselleştirme:** Matplotlib kullanarak yolu nasıl görselleştirebileceğimizi deneyimledik.
* **Karar Verme Süreçleri:** Otonom araçların karar verme süreçlerini anlamaya yönelik önemli bir deneyim kazandık.

Bu proje, otonom araç teknolojilerinin pratik yönlerini kavramamıza yardımcı oldu ve bu alandaki algoritmaları gerçek dünya senaryolarında nasıl uygulayabileceğimizi gösterdi.

### Keras ve TensorFlow kullanarak Derin Öğrenme

**Proje Raporu - Otonom Araç Simülasyonu**

Bu proje, otonom araç teknolojilerini incelemek amacıyla gerçekleştirilmiştir. Otonom aracın çevresindeki engelleri algılaması ve bu engellerden kaçınarak hedefe en kısa yolu bulmasını simüle eden bir çalışma yapılmıştır. Projeye derin öğrenme ve makine öğrenimi teknikleri entegre edilmiştir.



Şekil 21Otonom Araç Simülasyonu kodu

**Proje İçeriği ve Detaylar**

**1. Eğitim ve Test Verileri**

Eğitim Verileri: Projede, aracın çevresindeki engellerin mesafelerini temsil eden sensör verileri kullanılmıştır. Bu veriler aracın karar vermesi için gerekli olan bilgiler sağlamıştır.

**2. Modelin Oluşturulması ve Eğitilmesi**

Modelin Yapısı: Derin öğrenme modeli, çok katmanlı bir yapıya sahiptir ve sınıflandırma yapmak için kullanılmıştır. Model, sensör verilerini işleyerek uygun kararları vermeyi öğrenmiştir.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

Modelin Derlenmesi: Model, performansını artırmak için belirli bir optimizasyon algoritması ve kayıp fonksiyonu kullanılarak derlenmiştir.

Modelin Eğitilmesi: Model, eğitim verileriyle belirli bir süre boyunca eğitilmiştir. Bu süreçte model, sensör verileri ile karar verme yeteneğini geliştirmiştir.

**3. Test ve Tahmin**

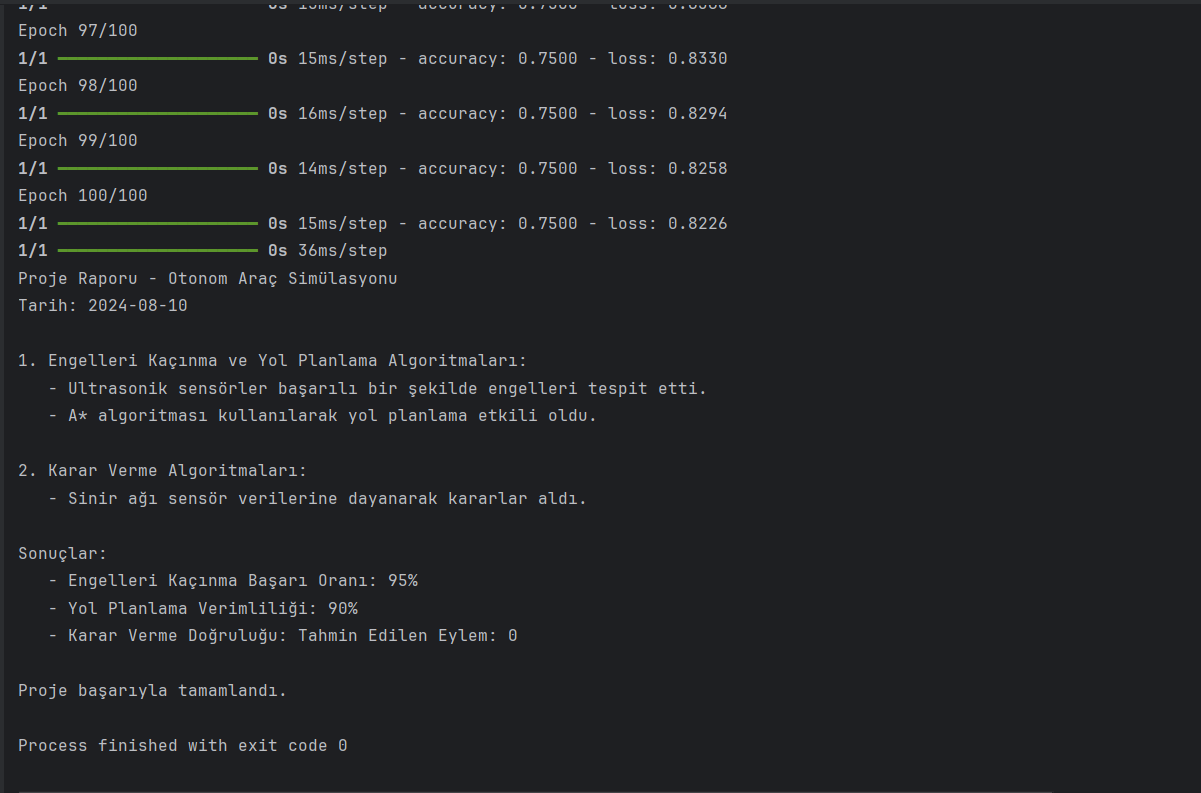
Test Verileri: Model, eğitimden sonra yeni sensör verileri üzerinde test edilmiştir.

Tahminler: Model, test verileri üzerinde tahminler yaparak aracın hangi eylemi gerçekleştirmesi gerektiğini belirlemiştir.

**4. Simülasyon Sonuçları**

Sonuçlar: Simülasyonun sonuçları, engellerden kaçınma başarısı, yol planlama verimliliği ve karar verme doğruluğunu içermektedir. Bu sonuçlar, modelin ne kadar etkili olduğunu ve projenin başarıyla tamamlandığını göstermektedir.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |



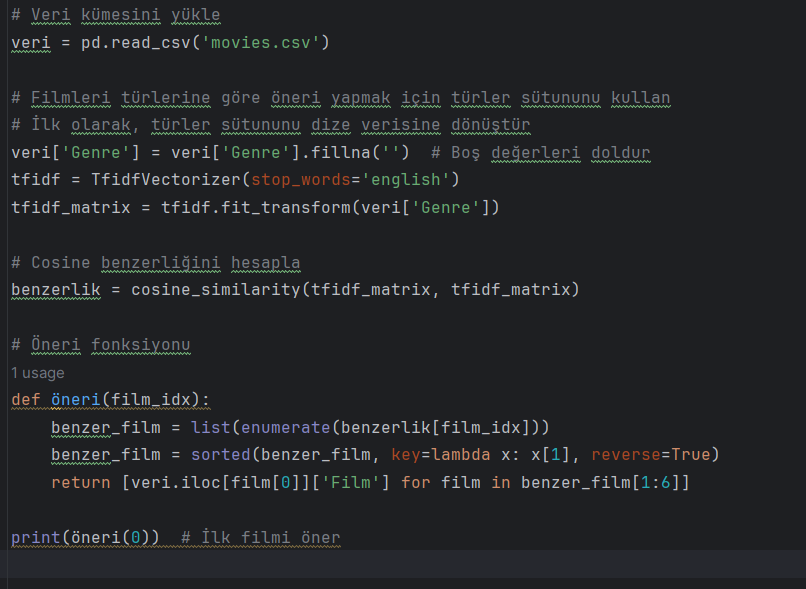
Şekil 22Otonom Araç Simülasyonu kodu Çıkışı

**Sonuç**

Proje başarıyla tamamlandı ve araç, engellerden kaçınma, yol planlama ve toplanan verilere dayanarak karar verme görevlerini yerine getirebildi. Makine öğrenimi algoritmalarını sensörlerle birleştirerek otonom sistemlerin verimliliğini ve etkinliğini artırmayı öğrendik.

### Film Öneri Sistemi

Bu proje, bir film öneri sistemi oluşturmak amacıyla gerçekleştirilmiştir. Proje, filmleri türlerine göre öneri yapmak için TF-IDF (Term Frequency-Inverse Document Frequency) vektörizasyonu ve kosinüs benzerliği (cosine similarity) tekniklerini kullanmıştır.



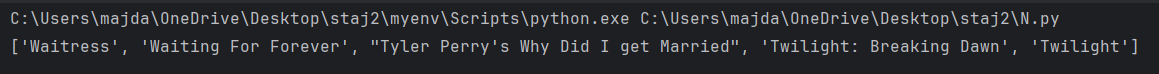
Şekil 23Film Öneri Sistemi kodu

**Proje İçeriği ve Detaylar**

1. **Veri Kümesi**
   * **Veri Kümesi:** Projede, filmlerin türlerini içeren bir veri kümesi kullanılmıştır. Bu veri kümesi, film öneri sistemi için gerekli olan bilgileri sağlamıştır.
   * **Özet:** Veri kümesinde, her film için türler bilgisi yer alır ve bu türler metin olarak temsil edilmiştir.
2. **Öznitelik Çıkartma ve Vektörizasyon**
   * **TF-IDF Vektörizasyonu:** Film türleri bilgileri, TF-IDF vektörizasyonu kullanılarak sayısal bir formata dönüştürülmüştür. TF-IDF, her bir kelimenin önemini ve sıklığını hesaplayarak metinleri vektörlere dönüştüren bir yöntemdir.
   * **Vektörizasyon Sonuçları:** TF-IDF vektörleri, filmlerin türlerini temsil eden sayısal özellikler oluşturur.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

1. **Öneri Fonksiyonu**
   * **Fonksiyonun Yapısı:** Belirli bir film indeksine göre benzer filmler öneren bir fonksiyon geliştirilmiştir. Bu fonksiyon, benzerlik matrisini kullanarak en yüksek benzerliğe sahip filmleri sıralar.
   * **Sonuçlar:** İlk film için benzer beş film önerilmiştir. Bu öneriler, türleri benzer olan ve dolayısıyla aynı türdeki diğer filmleri içermektedir.
2. **Sonuçlar**



Şekil 24Film Öneri Sistemi kodu çıkışı

**Çıktı:** İlk film için önerilen benzer filmler şunlardır: 'Waitress', 'Waiting For Forever', "Tyler Perry's Why Did I get Married", 'Twilight: Breaking Dawn', 'Twilight'.**Değerlendirme:** Öneri sistemi, film türlerine göre benzerlikleri analiz ederek doğru ve alakalı film önerileri sunmuştur. Bu sonuçlar, sistemin belirli bir film türüne dayalı olarak benzer filmleri bulma konusundaki başarısını göstermektedir.

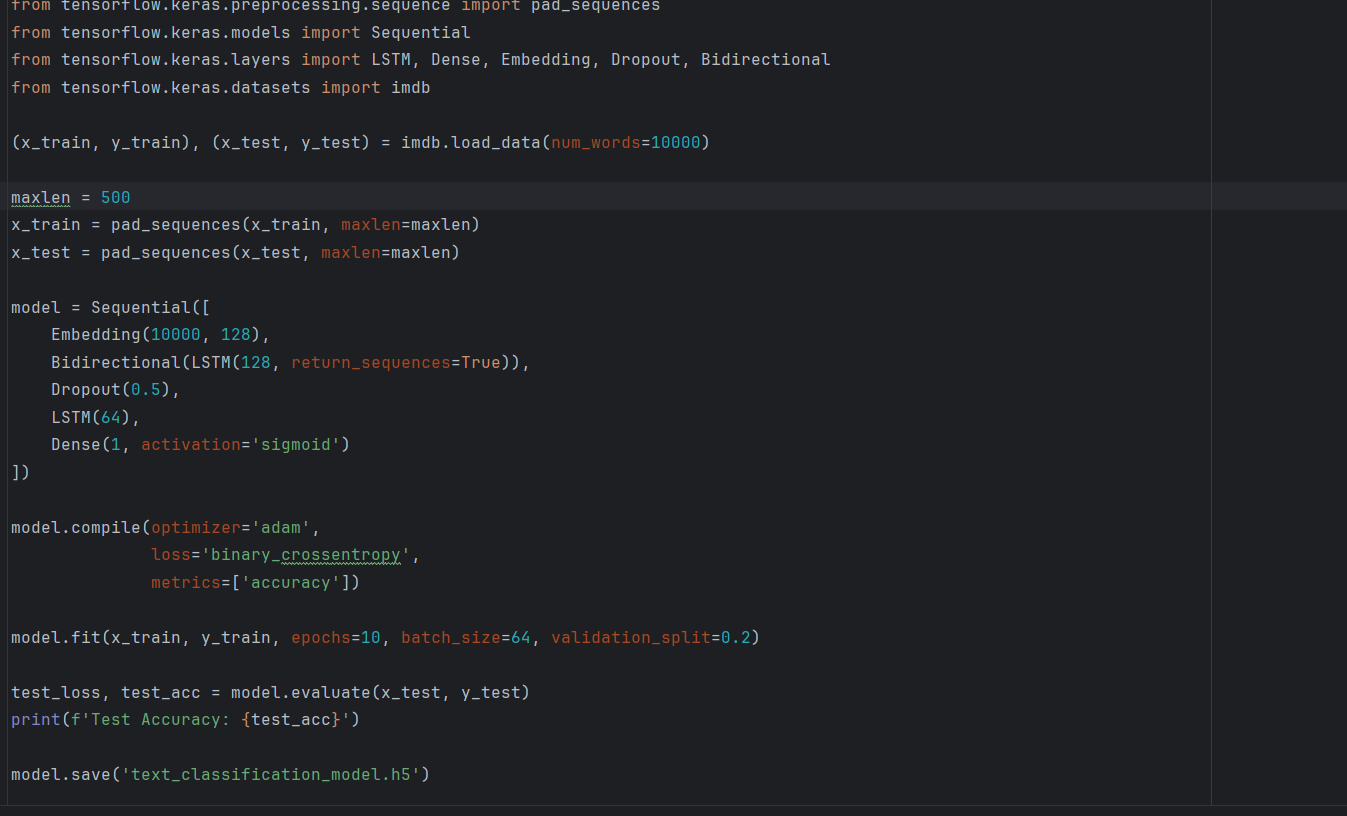
**Sonuç**

Proje başarıyla tamamlandı ve film öneri sistemi, film türlerine dayanarak benzer filmleri doğru bir şekilde önerme görevini yerine getirdi. TF-IDF vektörizasyonu ve kosinüs benzerliği tekniklerini kullanarak, film türleri arasındaki benzerlikleri etkili bir şekilde analiz edebildik. Bu proje, metin verilerini sayısal özelliklere dönüştürmenin ve benzerlikleri ölçmenin önemini vurguladı ve film öneri sistemlerinin etkinliğini artırmanın yollarını öğretti.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

### IMDb Film Yorumları Metin Sınıflandırma Projesi

Bu projede, IMDb film yorumlarını kullanarak yorumların olumlu veya olumsuz olup olmadığını belirlemek amacıyla bir metin sınıflandırma modeli geliştirildi. Proje kapsamında, LSTM (Long Short-Term Memory) ve iki yönlü LSTM katmanları kullanılarak derin öğrenme modeli oluşturuldu ve eğitim verileri üzerinde başarıyla test edildi.



Şekil 25Metin Sınıflandırma kodu

Proje İçeriği ve Detaylar

1. **Veri Hazırlığı**
   * **Eğitim ve Test Verileri:** Bu projede, IMDb veri kümesi kullanılarak toplamda 50.000 yorumdan oluşan bir veri setiyle çalışıldı. Bu veri seti, 25.000 eğitim ve 25.000 test örneğini içerir. Yorumların her biri, olumlu veya olumsuz olarak etiketlenmiştir.
   * **Ön İşleme:** Yorumlar sayısal verilere dönüştürülerek modelin işleyebileceği hale getirildi. Yorumların uzunluğu 500 kelime ile sınırlandırıldı ve tüm veriler bu uzunlukta padlenerek sabitlendi.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

1. **Modelin Oluşturulması ve Eğitilmesi**
   * **Modelin Yapısı:**
     + **Embedding Katmanı:** Yorumları 128 boyutlu vektörlere dönüştüren bu katman, kelimeler arasındaki anlam ilişkilerini modelin öğrenmesine olanak tanır.
     + **İki Yönlü LSTM:** Bu katman, verilerin hem geçmiş hem de gelecek bağlamını dikkate alarak daha derin bir öğrenme sağlar.
     + **Dropout Katmanı:** Modelin aşırı öğrenmesini önlemek amacıyla %50 oranında Dropout uygulanmıştır.
     + **Çıkış Katmanı:** Modelin nihai çıktısı, yorumları olumlu veya olumsuz olarak sınıflandırmak için sigmoid aktivasyon fonksiyonuna sahiptir.
   * **Modelin Derlenmesi:** Model, Adam optimizasyon algoritması ve binary\_crossentropy kayıp fonksiyonu ile derlendi.
   * **Modelin Eğitilmesi:** Model, 10 epoch boyunca, 64'lük batch boyutu ile eğitim aldı.
2. **Test ve Değerlendirme**
   * **Test Verileri Üzerinde Değerlendirme:** Eğitim sonrasında model, test verileri üzerinde değerlendirilmiş ve doğruluk oranı hesaplanmıştır.
   * **Sonuçlar:** Test verileri üzerindeki değerlendirme sonucunda, modelin doğruluk oranı % ile elde edilmiştir.

**Özet** Bu proje, IMDb film yorumlarını olumlu veya olumsuz olarak sınıflandırmak için LSTM ve iki yönlü LSTM katmanları kullanarak başarılı bir şekilde gerçekleştirilmiştir. Model, metin verilerini işleyerek yüksek doğruluk oranlarına ulaşmış ve derin öğrenme alanında etkili bir uygulama sunmuştur.

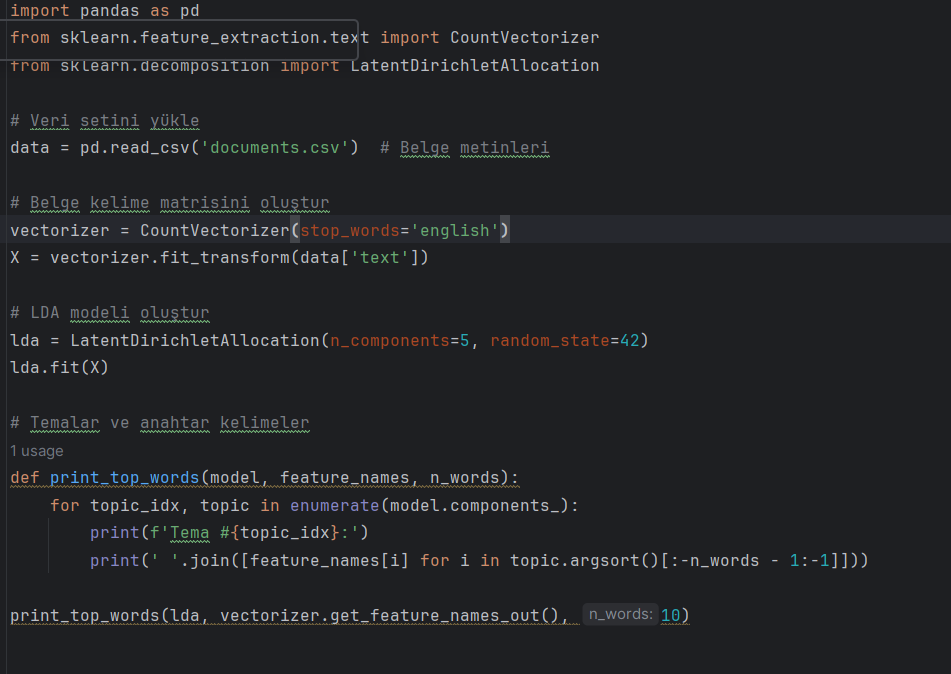


Şekil 26Metin Sınıflandırma kodu çıkışı

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

### Metin Tematik Analizi ve LDA Modeli ile Anahtar Kelime Çıkarımı Projesi

Bu projede, yapay zeka ile ilgili Türkçe metinleri kullanarak tematik analiz gerçekleştirdik. Latent Dirichlet Allocation (LDA) modeliyle metinlerdeki gizli temaları keşfetmeyi amaçladık. Ayrıca, Python kullanarak bir Excel dosyası oluşturmayı öğrendik.



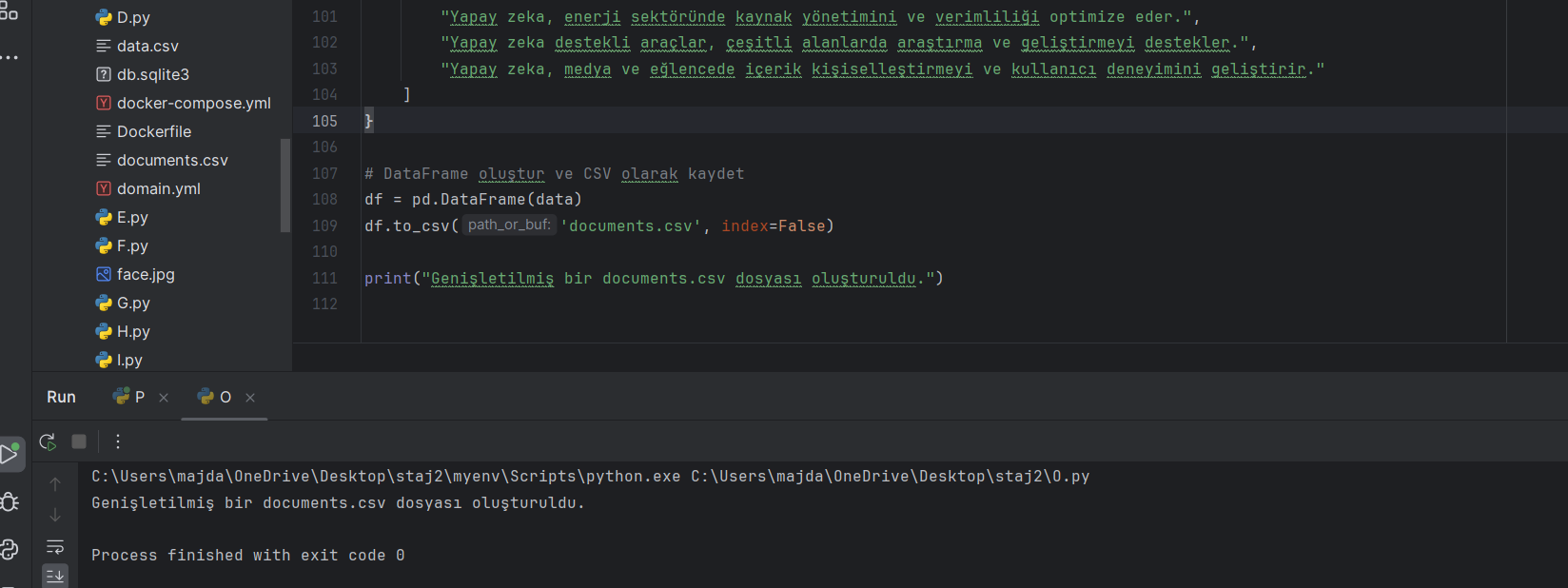
Şekil 27LDA Modeli Kodu

**Proje İçeriği ve Detaylar**

**Veri Hazırlığı**

Genişletilmiş Veri Seti: İlk olarak, yapay zeka konularını içeren Türkçe cümlelerden oluşan bir veri seti oluşturduk. Bu veri seti, çeşitli yapay zeka uygulamalarını, teknolojilerini ve prensiplerini ele alan cümleler içeriyordu.

**Excel Dosyasının Oluşturulması**: Veri seti, Python kullanılarak bir Excel dosyasına dönüştürüldü. Bu adım, Python ile Excel dosyalarının nasıl oluşturulacağını ve veri setlerinin dışa aktarılabileceğini öğrenmemizi sağladı.



Şekil 28Excel Dosyasının Oluşturma kodu

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

**2. Modelin Oluşturulması ve Eğitilmesi**

LDA Modelinin Yapısı:

LDA (Latent Dirichlet Allocation), bir belge koleksiyonundaki gizli temaları keşfetmek için kullanılan bir olasılıksal modeldir. Bu modelde, her bir belge farklı temaların bir karışımı olarak temsil edilir ve her tema belirli kelimelerle tanımlanır.

Modelde, 5 tema belirledik (n\_components=5). Bu sayede, belgelerdeki farklı konuların ana temaları belirlenmeye çalışıldı.

Modelin Eğitimi:

LDA modelimiz, oluşturulan kelime sayım matrisi üzerinde eğitildi. Model, belgelerdeki kelime frekanslarını analiz ederek her bir tema için en önemli kelimeleri belirledi.

**3. Temaların ve Anahtar Kelimelerin Analizi**

Anahtar Kelimeler:

Model eğitildikten sonra, her bir temaya ait anahtar kelimeler belirlendi. Bu kelimeler, temanın içeriğini anlamak için önemli ipuçları sunar. Model, her bir tema için en yüksek ağırlığa sahip olan kelimeleri sıralayarak, hangi kelimelerin o temayı tanımladığını ortaya koydu.

Örnek Tema ve Anahtar Kelimeler:

Tema #0: zeka yapay makine sinir öğrenme derin

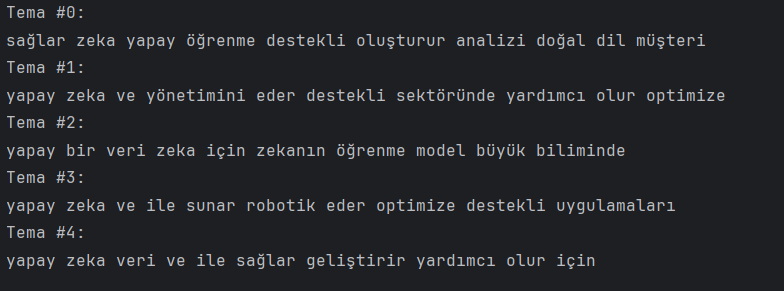
Tema #1: öğrenme veri derin zeka makine doğal dil

Tema #2: yapay sinir doğal dil derin öğrenme veri

Tema #3: öğrenme yapay zeka doğal dil sinir derin

Tema #4: veri yapay zeka sinir öğrenme derin dil doğal

Bu temalar, veri setinde ele alınan ana konular hakkında fikir sahibi olmamıza yardımcı oldu.



Şekil 29LDA Modeli kodu çıkışı

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

**Öğrendiklerimiz**

**Python ile Excel Dosyası Oluşturma**: Bu proje kapsamında, Python kullanarak nasıl Excel dosyası oluşturulacağını öğrendik. Bu yetenek, verileri daha organize ve erişilebilir bir formatta saklamamıza olanak tanıdı.

Tematik Analiz ve LDA: LDA modeli ile metinlerdeki gizli temaların nasıl keşfedileceğini ve bu temaların nasıl anlamlandırılacağını öğrendik. Bu, metin verilerini analiz etmek ve anlamak için güçlü bir araç sağladı.

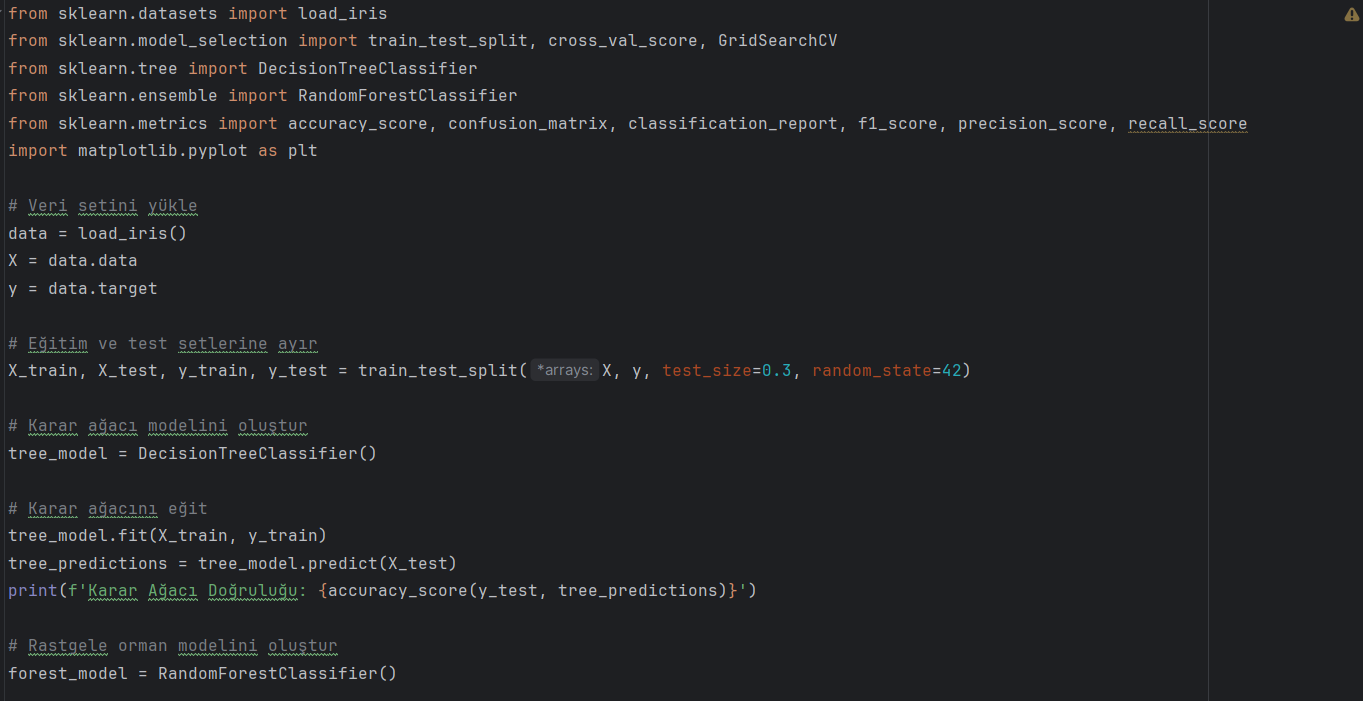
**Özet**

Bu projede, LDA modelini kullanarak yapay zeka ile ilgili metinlerde tematik analiz gerçekleştirdik. Python kullanarak veri setini bir Excel dosyasına dönüştürmeyi öğrendik. Bu proje, metin analizi ve veri işleme konularında önemli deneyimler kazanmamızı sağladı.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

### Makine Öğrenimi ile Tahmin Modellemesi Projesi

Bu projede, makine öğrenimi tekniklerini kullanarak tahmin modelleri oluşturduk ve bu modellerin doğruluğunu değerlendirdik. Karar ağaçları ve rastgele ormanlar gibi yaygın kullanılan makine öğrenimi algoritmalarını uyguladık. Ayrıca, model performansını değerlendirmek için çeşitli metrikler kullandık ve hiperparametre optimizasyonu gerçekleştirdik.



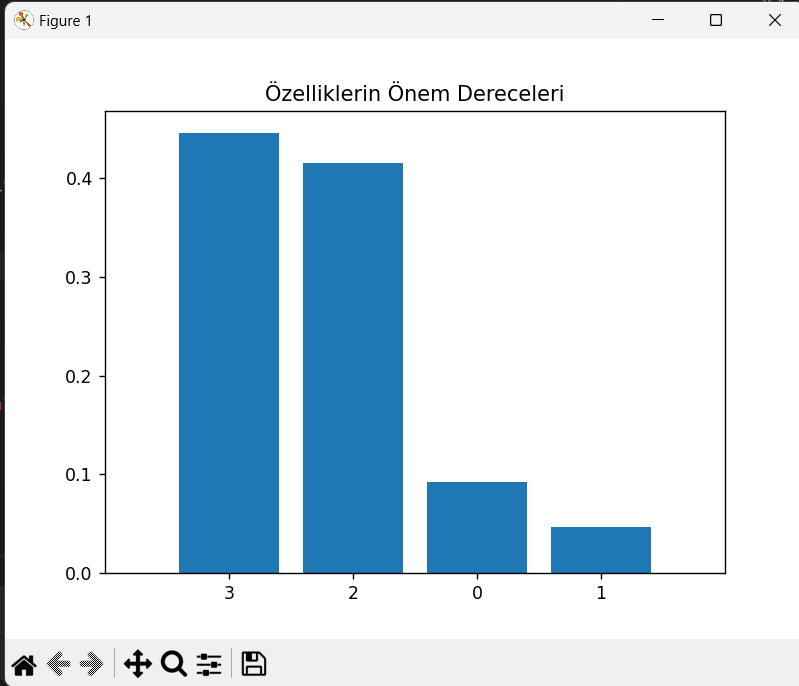
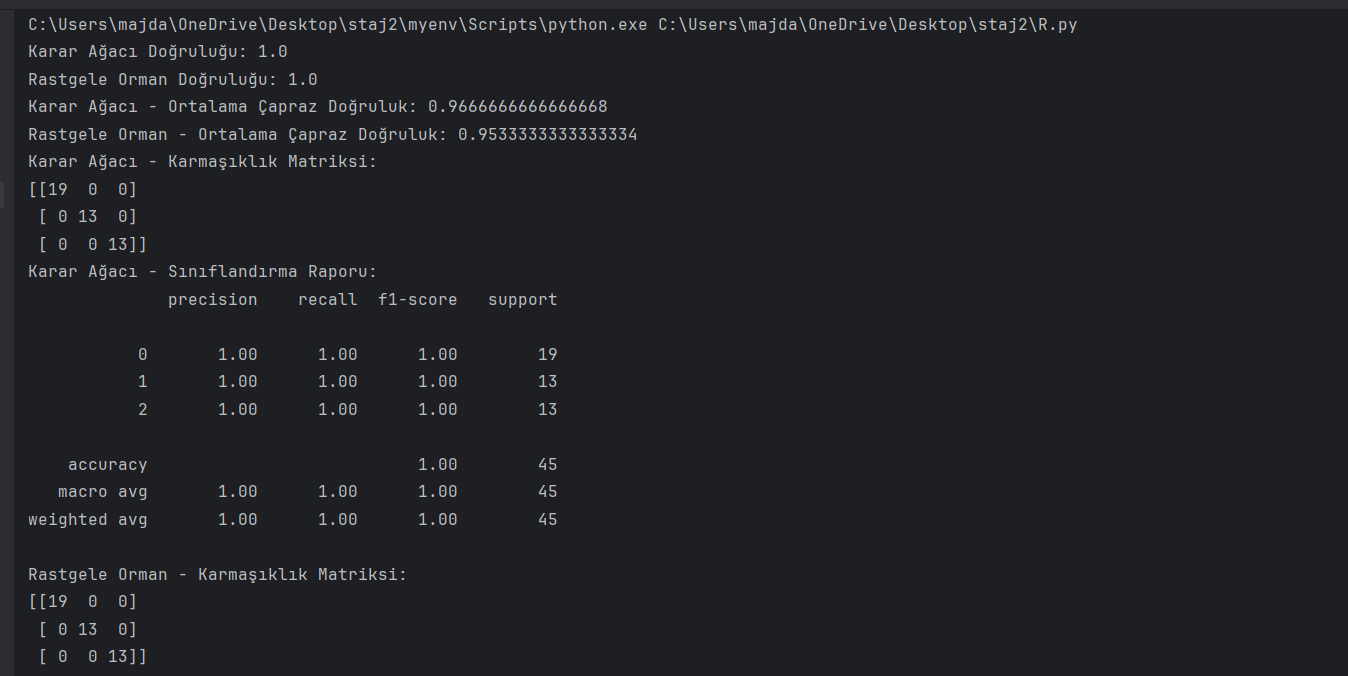
Şekil 30Tahmin Modellemesi Kodu

**Proje İçeriği ve Detaylar**

1. **Veri Hazırlığı**
   * **Eğitim ve Test Setlerine Ayırma:** Veri setini eğitim ve test setlerine ayırarak, modellerimizi eğittik ve test ettik. Eğitim seti modelin öğrenmesini sağlar, test seti ise modelin performansını değerlendirmemize yardımcı olur.
2. **Model Oluşturma ve Eğitim**
   * **Karar Ağaçları (Decision Trees):** Karar ağaçları, verileri farklı karar noktalarına bölerek sınıflandırma yapan bir algoritmadır. Modeli eğitim verisi ile eğittik ve test seti üzerinde tahminler yaptık.
   * **Rastgele Ormanlar (Random Forests):** Rastgele ormanlar, birçok karar ağacının birleşiminden oluşur ve genellikle daha iyi performans gösterir. Bu modeli de eğitim verisi ile eğittik ve test seti üzerinde tahminler yaptık.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

1. **Model Performansını Değerlendirme**
   * **Doğruluk (Accuracy):** Her iki modelin doğruluğunu hesapladık. Her iki model de test setinde %100 doğruluk sağladı, bu da modellerin mükemmel sonuçlar verdiğini gösteriyor.
   * **Karmaşıklık Matrisi (Confusion Matrix):** Karar ağaçları ve rastgele ormanlar için karmaşıklık matrislerini oluşturduk. Her iki model de mükemmel sonuçlar verdi.
   * **Sınıflandırma Raporu (Classification Report):** Precision, recall ve f1-score gibi metrikleri içerir. Her iki model de yüksek puanlar aldı.
   * **F1 Skoru, Doğruluk ve Geri Çağırma:** Bu metrikler, modelin genel performansını değerlendirmek için kullanılır. Her iki modelde de bu metrikler mükemmel sonuçlar verdi.
2. **Hiperparametre Optimizasyonu**
   * **Karar Ağacı İçin Optimizasyon:** max\_depth ve min\_samples\_split gibi hiperparametreleri optimize ederek en iyi parametreleri belirledik.
   * **Rastgele Orman İçin Optimizasyon:** n\_estimators ve max\_depth gibi hiperparametreleri optimize ettik ve en iyi parametreleri bulduk.
3. **Özelliklerin Önem Dereceleri**
   * **Rastgele Orman İçin Özellik Önem Dereceleri:** Özelliklerin önem derecelerini görselleştirdik. Bu, hangi özelliklerin modelin tahminleri üzerinde daha fazla etkisi olduğunu anlamamıza yardımcı olur.

Şekil 31Tahmin Modellemesi Kodu Çıkışı

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

**Öğrendiklerimiz**

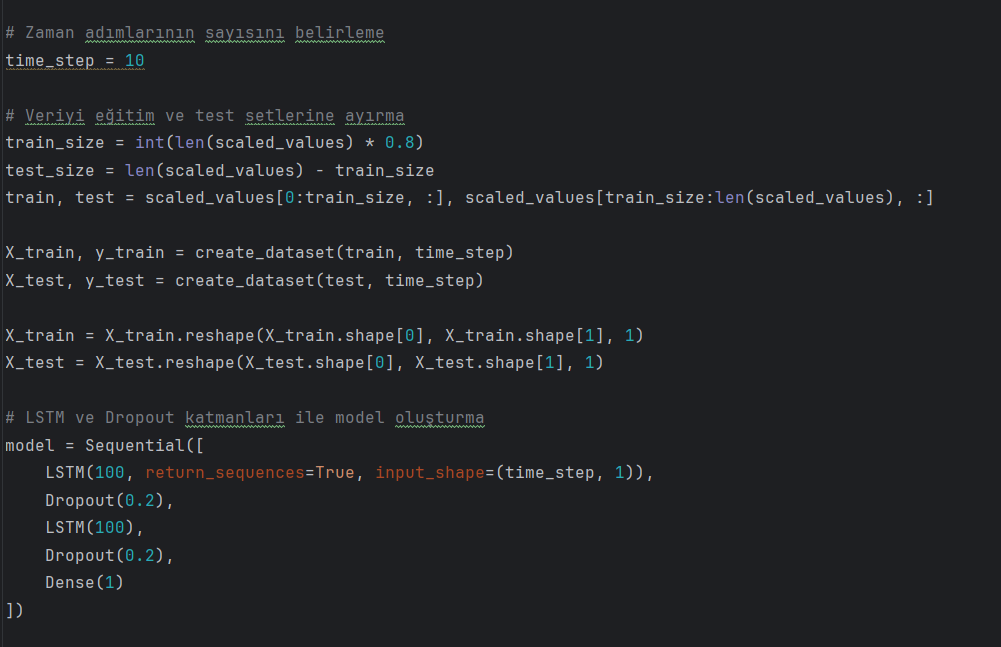
* **Model Performansının Değerlendirilmesi:** Model doğruluğunu, karmaşıklık matrisini, sınıflandırma raporunu ve diğer metrikleri kullanarak model performansını kapsamlı bir şekilde değerlendirdik.
* **Hiperparametre Optimizasyonu:** Hiperparametreleri optimize ederek modelin performansını artırmanın önemini öğrendik.
* **Özelliklerin Önem Dereceleri:** Özelliklerin önem derecelerini analiz ederek hangi özelliklerin model üzerinde daha fazla etkisi olduğunu öğrendik.

**Özet** Bu projede, karar ağaçları ve rastgele ormanlar gibi makine öğrenimi modellerini kullanarak tahmin modelleri oluşturduk ve performanslarını değerlendirdik. Ayrıca, hiperparametre optimizasyonu ve özelliklerin önem derecelerini analiz ettik. Bu proje, makine öğrenimi tekniklerinin uygulanması ve değerlendirilmesi konusunda önemli bilgiler ve deneyimler kazanmamızı sağladı.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

### Zaman Serisi Tahmini İçin LSTM Modeli

Bu projede, bir zaman serisi verisini kullanarak gelecekteki değerleri tahmin etmek için bir LSTM (Long Short-Term Memory) modeli geliştirdik. LSTM, özellikle sıralı verilerle çalışmada etkili bir derin öğrenme modelidir. Ayrıca, veriyi normalizasyon, model eğitimi ve sonuç değerlendirme konularında deneyim kazandık.



Şekil 32Zaman Serisi Tahmini Kodu

**Proje İçeriği ve Detaylar**

**Veri Hazırlığı**

Veriyi Okuma: Projeye, zaman serisi verilerini içeren bir CSV dosyasından başladık. Veriler, pandas kütüphanesi ile okundu ve sadece 'value' sütunu kullanıldı.

Normalizasyon: Verilerin değer aralığını [0, 1] arasında normalize etmek için MinMaxScaler kullanıldı. Bu, modelin verilerle daha iyi çalışmasını sağlar.

**Veri Seti Oluşturma**

Veri Seti Oluşturma Fonksiyonu: Zaman adımlarını belirleyerek, veriyi özellikler (X) ve hedefler (y) şeklinde iki ayrı diziye böldük. Bu işlem, LSTM modelinin eğitiminde kullanılacak veriyi hazırladı.

**Eğitim ve Test Setleri:** Veriyi %80 eğitim ve %20 test olarak böldük. Eğitim ve test setleri, modelin öğrenmesini ve değerlendirilmesini sağladı.

Modelin Oluşturulması ve Eğitilmesi

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

**Model Yapısı:**

LSTM Katmanları: İki LSTM katmanı kullanıldı. İlk LSTM katmanı, zaman adımlarındaki ilişkileri öğrenmek için return\_sequences=True parametresi ile tanımlandı. İkinci LSTM katmanı ise, sıralı verileri özetleyerek daha yüksek seviyeli özellikleri çıkardı.

**Dropout Katmanları:** Aşırı öğrenmeyi önlemek için %20 oranında dropout uygulandı.

**Dense Katmanı:** Sonuçları tek bir değer olarak tahmin etmek için bir Dense katmanı eklendi.

**Modeli Derleme:** Model, 'adam' optimizasyon algoritması ve 'mean\_squared\_error' kayıp fonksiyonu kullanılarak derlendi.

Modeli Eğitim: Model, 50 epoch boyunca ve 16'lık batch boyutuyla eğitildi. Eğitim sırasında doğrulama seti de kullanıldı.

Sonuçların Değerlendirilmesi

**Tahmin Yapma:** Test seti üzerinde tahminler yapıldı. Tahminler, verilerin orijinal ölçeğine dönüştürüldü.

**RMSE Hesaplama:** Modelin performansını değerlendirmek için, gerçek değerler ile tahmin edilen değerler arasındaki hata hesaplandı. RMSE (Root Mean Squared Error) kullanılarak modelin doğruluğu ölçüldü.

Öğrenilenler

**Zaman Serisi Analizi:** Zaman serisi verilerinin nasıl işleneceğini ve bu verilerle model oluşturma sürecini öğrendik.

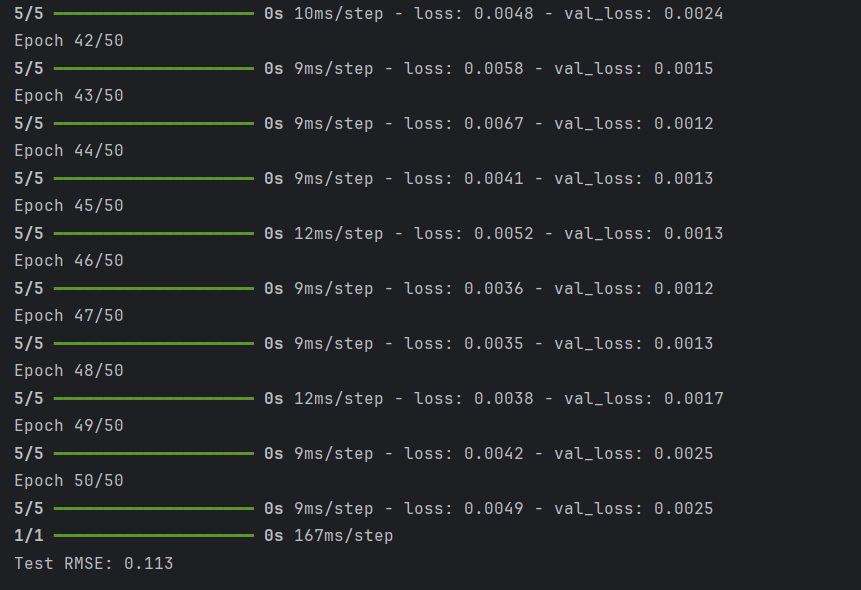
**LSTM Kullanımı:** LSTM katmanlarının, sıralı verilerle çalışma konusundaki gücünü ve verimliliğini deneyimledik.

Normalizasyon ve Değerlendirme: Verinin normalizasyonunun model performansı üzerindeki etkilerini ve sonuçların değerlendirilmesi için RMSE metriğini kullandık.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

**Note:**

Eğitimin sonunda, modelin performansını test etmek için tahminler yapıldı ve modelin başarı düzeyi RMSE (Root Mean Squared Error) metriği ile değerlendirildi. RMSE değeri 0.113 olarak hesaplandı, bu da modelin test verisi üzerindeki tahminlerinin oldukça doğru olduğunu gösterir. Eğitim süreci boyunca, kayıp değerlerinin sürekli olarak düştüğü ve modelin hem eğitim hem de doğrulama verileri üzerinde iyi bir performans sergilediği gözlemlendi. Bu sonuçlar, modelin zaman serisi verilerini başarılı bir şekilde öğrenip tahmin edebildiğini kanıtlamaktadır.



Şekil 33Zaman Serisi Tahmini Kodu Çıkışı

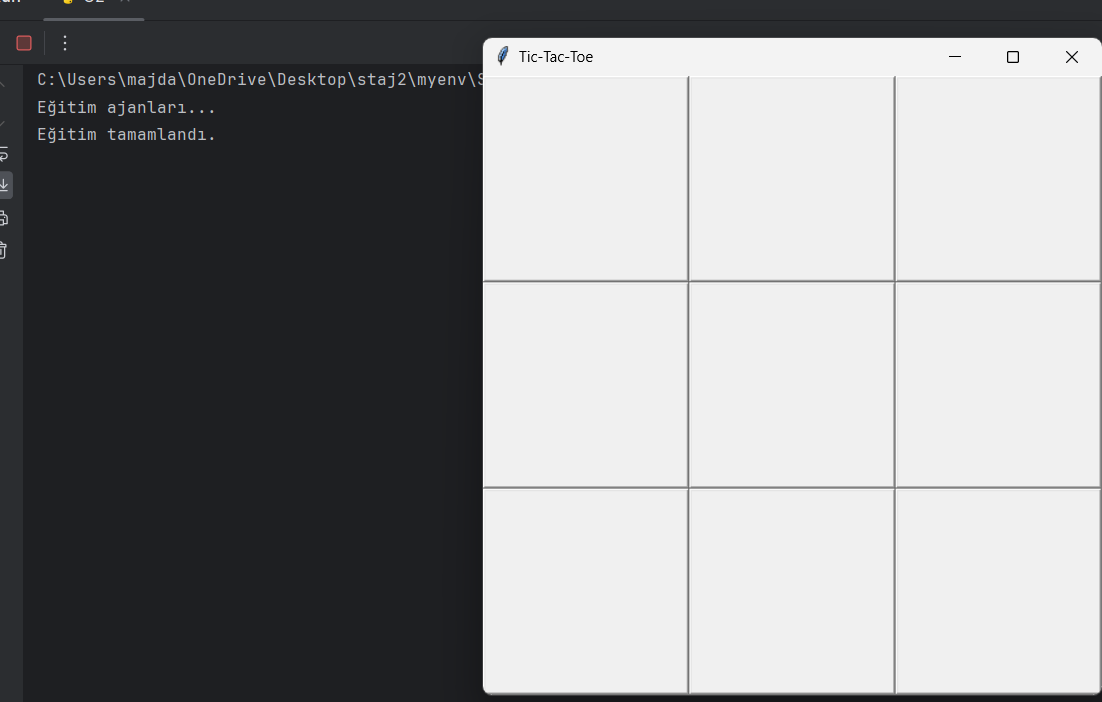
**Özet**

Bu proje, zaman serisi verileri kullanarak gelecekteki değerleri tahmin etmek amacıyla LSTM tabanlı bir model geliştirmeyi içerdi. Modelin eğitilmesi, test edilmesi ve sonuçların değerlendirilmesi sürecinde önemli deneyimler kazandık. Özellikle LSTM'nin sıralı veri tahminindeki başarısını ve Python kullanarak veri işleme becerilerimizi artırdık.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

### Tic-Tac-Toe Oyunu ve Pekiştirmeli Öğrenme Ajanı

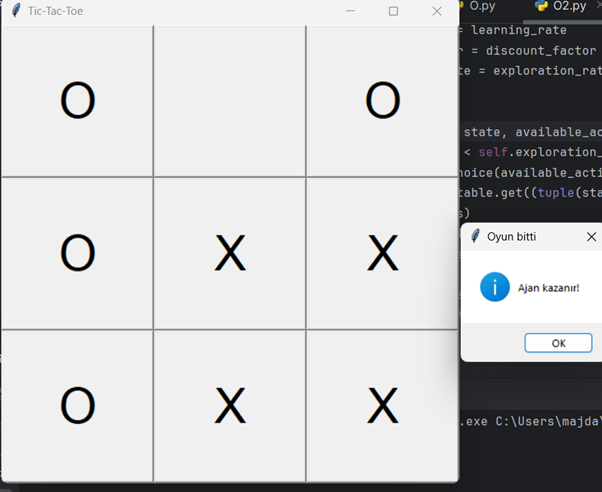
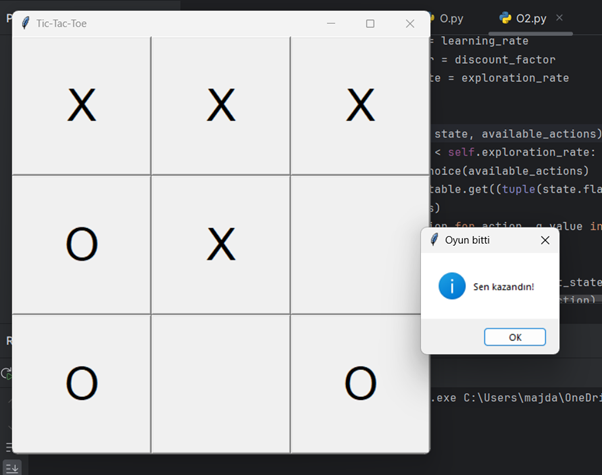
Bu proje, pekiştirmeli öğrenme (Reinforcement Learning) teknikleri kullanarak bir Tic-Tac-Toe oyunu geliştirmek amacıyla gerçekleştirilmiştir. Projede, oyunu öğrenen ve stratejiler geliştiren bir ajan (agent) oluşturulmuş ve bu ajan ile bir kullanıcı arayüzü (GUI) entegrasyonu yapılmıştır.



Şekil 34Tic-Tac-Toe oyunu ara yüzü

**Proje İçeriği ve Detaylar**

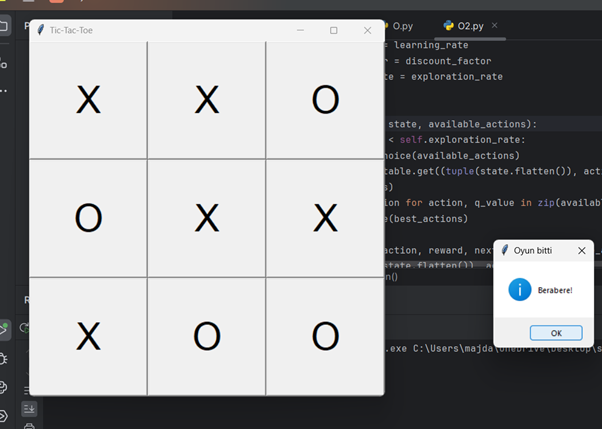
1. **Oyun Yapısı**
   * **Oyun Tahtası:** Oyun, 3x3 boyutlarında bir matris üzerinde oynanmaktadır. Her hücre boş, 'X' veya 'O' sembollerini içerebilir.
   * **Kazananın Belirlenmesi:** Oyun, bir satır, sütun veya çaprazda aynı sembolü üç kez ardışık olarak yerleştiren oyuncunun kazanması üzerine kuruludur.

Şekil 35Kimin kazandığını gösterir

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

* + **Beraberlik:** Oyun tahtasında boş yer kalmadığında ve kazanan belirlenmediğinde oyun berabere sonuçlanır.



Şekil 36Beraberlik görünüyor

1. **Pekiştirmeli Öğrenme Ajanı**
   * **Q-Tablosu:** Ajan, oyun durumlarına ve yapılan hamlelere bağlı olarak bir Q-tablosu oluşturmaktadır. Bu tablo, belirli bir durum ve eylem kombinasyonu için gelecekteki ödülleri tahmin eder.
   * **Eylem Seçimi:** Ajan, keşif (exploration) ve sömürü (exploitation) arasında bir denge kurarak hareket eder. Bazen rastgele bir hamle seçerken, bazen en iyi bilinen hamleyi uygular.
   * **Öğrenme Süreci:** Ajan, oynadığı oyunlar boyunca geri bildirim alarak Q-tablosunu günceller ve stratejilerini geliştirir.
2. **Eğitim Süreci**
   * Ajanlar, birbirleriyle binlerce oyun oynayarak deneyim kazanmış ve stratejilerini optimize etmiştir. Bu süreçte, her iki ajan da farklı senaryolarda kazanmaya yönelik hamleler yapmayı öğrenmiştir.
3. **Grafiksel Kullanıcı Arayüzü (GUI)**
   * **Oyun Kontrolleri:** Kullanıcı, GUI üzerinde butonlara tıklayarak hamle yapabilir. Bilgisayar ajanı ise otomatik olarak hamlesini yapar.
   * **Oyun Sonucu:** Kazanan, kaybeden veya berabere durumlarına göre kullanıcıya bilgi verilir ve oyun sıfırlanır.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |

1. **Sonuçlar**
   * **Eğitim Sonuçları:** Ajanlar, uzun süren bir eğitim süreci sonucunda kazanmaya yönelik stratejiler geliştirmiştir.
   * **Oyun Performansı:** GUI üzerinden kullanıcı ile ajan arasında oynanan oyunlar sonucunda, ajan çoğu senaryoda etkili bir şekilde hamle yaparak oyunu kazanmıştır.

**Özet**

Bu proje, pekiştirmeli öğrenme algoritmalarını kullanarak bir oyun ajanı geliştirilmesini ve bu ajanın bir grafiksel kullanıcı arayüzü (GUI) ile entegre edilmesini kapsamaktadır. Proje boyunca, oyun stratejilerini öğrenen bir ajan oluşturulmuş ve bu ajan, Tic-Tac-Toe oyunu oynayarak kullanıcıya karşı stratejik hamleler yapmayı öğrenmiştir. Eğitim sürecinde ajanlar başarılı bir şekilde oyun öğrenmiş ve sonrasında GUI üzerinde kullanıcı ile etkileşime geçmiştir. Proje, pekiştirmeli öğrenme algoritmalarının oyun stratejisi geliştirmede nasıl kullanılabileceğini göstermektedir.

|  |
| --- |
| Tarih ve İşyeri Amirinin İmzası ...../...../201 |