

**T.C
KONYA TEKNİ ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ
BİLGİSAYAR MÜHENDİSLİĞİ UYGULAMASI – 1/2
(BITİRME PROJESİ-1/2) FİNAL RAPOR FORMU**

Öğrencinin Adı- Soyadı	Yusuf Tunahan ETLİK		
Numarası:	201213081	İmzası:	
Proje Danışmanı Adı Soyadı:	Doç. Dr. Ersin KAYA	İmzası:	
Sınav Tarihi:	12.06.2024		

Projenin Adı: Uçak bilet, otel ve ulaşım fiyatının tahmini bütçesini çıkartan yapay zekâ

PROJEDE YAPILAN ÇALIŞMALARIN ÖZETİ

Dönem içi yapılan çalışmalara bitirme projesinin seçilmesi için gerekli kaynakları toplayarak ve proje danışmanına proje önerileri sunarak başlandı. En son olarak danışman ile ortak karar kılarkak uçak bilet, otel ve ulaşım fiyatının tahmini bütçesinin yapay zeka ile çıkartılması projesi seçildi. Projenin ana amacı ve kapsamı belirlendi. Projenin ana amacı kısa süreli ve hızlı planlanması gereken tatillerde kullanıcının bütçesine uygun tatil seçeneklerini göstermektir. Projenin kapsamı enflasyon ve özel günleri yok sayarak ilk olarak Hindistan'ın yerel bölgelerinde uçak bilet, otel ve ulaşım fiyatlarını tahmin etmeyi içerir.

Hindistan için uçak bilet verilerinin veri kazıma yardımıyla çekilerek fiyat tahmini yapılabilmesi üzerine uygun bir web sitesi arandı. Skyscanner, Google Uçuşlar, Expedia, Kayak ve makemytrip gibi pek çok seyahat sitesi incelendi. Veri setindeki tüm özelliklerin bulunması, veri kalitesinin yüksek olması ve kullanıcı dostu arayüzüyle veri kazımını kolaylaştıracığı için www.makemytrip.com sitesi seçildi.

Veri kazıma işlemi önceden yapılmadığından, birçok veri kazımı yapan eğitici videolar izlenerek ne şekilde yapılacağı öğrenildi. Veri kazıma işleminde sıkça kullanılan Python, Javascript, Ruby ve Java dillerine bakılarak hangisine ihtiyaç duyulduğuna karar verildi. Hem önceden deneyim sahibi olunduğundan hem de daha anlaşılır olduğundan Python dilinde karar kılındı. Python web kazıma çalışmaları internet kaynaklarından izlenerek hangi scraping kütüphanesinin kullanılacağı belirlendi. Web kazıma işleminde sıkça kullanılan Beautiful Soup, Scrapy ve Selenium WebDriver araştırılarak ihtiyaca yönelik karşılaştırma yapıldı. Bunun sonucunda, internet sitelerini otomatikleştirmek için kullanılan ve dinamik, kullanıcı etkileşimi gerektiren, karmaşık Javascript işlemlerini içeren internet sitelerinde bile ideal çalışan Selenium WebDriver seçildi. Seçilen Python kodundaki Selenium kütüphanesi ile çalışılarak web kazıma işlemi gerçekleştirildi. Yapay zeka eğitimi için kullanılacağı için "csv" kütüphanesi eklenerek .csv uzantılı veriseti dosyaları olarak kaydedilmesi sağlandı. "time" kütüphanesi de eklenerek açılan Chrome sayfasındaki verilerin düzgün yüklenmesi sağlandı. Tarayıcı seçenekleri tanımlanarak Chrome tarayıcısının yapılandırılması sağlandı.

Python kodu şu şekilde yazıldı: Webdriver.Chrome() yöntemiyle Chrome tarayıcısı, önceden belirlenmiş ayarlarla başlatıldı. Tarayıcı www.makemytrip.com sitesindeki belirli bir URL'ye yönlendirildi ve açıldı. Sayfanın tamamen yüklenmesi için 25 saniye beklenildi. Verilerin tutulacağı csv dosyasının yolu belirlendi. Değişkenler oluşturularak bir döngü başlatıldı. Bu döngü, belirli sayıda uçuş bloğunu taramak için kullanıldı. “try-except” bloklarıyla hatalar kontrol edildi. Önce bloğun var olup olmadığı kontrol edildi. Blok yoksa tarayıcı ve kod sonlandırıldı. Her uçuş bloğundan çeşitli bilgileri almak için XPath ve/veya CSS seçenekleri kullanılarak belirli elementler bulundu ve bu bilgiler yazdırıldı. Yazdırılan bilgiler listelendi ve “csv.writerows()” ile csv dosyasına yazıldı. Adımların sonunda yeni adıma geçerken 2 saniye beklenildi. Döngü değişken sayısına ulaştığında tarayıcı ve kod sonlandırıldı. Bu şekilde, www.makemytrip.com sayfasındaki veriler web scraping yöntemiyle csv dosyasına aktarıldı.

Resim 1.a (Python selenium kütüphanesiyle yazılmış web kazıma kod parçası)

```

arrtime = block.find_element(By.XPATH, './div[1]/div[2]/div[1]/div[3]/label/div/div/div[3]/p[1]').text
print("Arrival Time:", arrtime)
arrcity = block.find_element(By.XPATH, './div[1]/div[2]/div[1]/div[3]/label/div/div/div[3]/p[2]').text
print("Arrival City:", arrcity)
duration = block.find_element(By.XPATH, './div[1]/div[2]/div[1]/div[3]/label/div/div/div[2]/p').text
print("Duration:", duration)

price = block.find_element(By.XPATH, './div[1]/div[2]/div/div').text
price = price.split('\n')
price = price[0][2:]
print("Price:", price)

except Exception as e:
    print("Error:", e)
data = [[fname, fcode, depcity, deptime, arrcity, arrtime, duration, price]]
with open(CSV_PATH, 'a', newline='', encoding="utf-8") as file:
    writer = csv.writer(file)
    writer.writerows(data)
    time.sleep(2)

driver.close()

```

Resim 1.b (Python selenium kütüphanesiyle yazılmış web kazıma kod parçası)

Dönem içi yapılan çalışmaların ilki uçak fiyat tahmini yapabilmek için veri seti kaynakları araştırıldı. Bu kaynaklardan birleşebilecek ve özellikleri nümerik değerlere dönüştürülebilecek veri setleri seçildi.

(Kaynaklar – 1 – 2 ve 3)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	date,airline,code,num,code,dep_time,from,to,arr_time,to,price																						
2	11-02-2022,Air India,AI,868,18:00,Delhi,02h 00m,non-stop,20:00,Mumbai,"25,612"																						
3	11-02-2022,Air India,AI,624,19:00,Delhi,02h 15m,non-stop,21:15,Mumbai,"25,612"																						
4	11-02-2022,Air India,AI,531,20:00,Delhi,24h 45m,"1-stop																						
5	"20:45,Mumbai,"42,220"																						
6	11-02-2022,Air India,AI,839,21:25,Delhi,26h 30m,"1-stop																						
7	"23:55,Mumbai,"44,450"																						
8	11-02-2022,Air India,AI,544,17:15,Delhi,06h 40m,"1-stop																						
9	"23:55,Mumbai,"46,690"																						
10	11-02-2022,Vistara,UK,985,19:50,Delhi,02h 10m,non-stop,22:00,Mumbai,"50,264"																						
11	11-02-2022,Air India,AI,473,18:40,Delhi,22h 45m,"1-stop																						
12	"15:00,Mumbai,"50,669"																						
13	11-02-2022,Vistara,UK,977,19:00,Delhi,02h 15m,non-stop,21:15,Mumbai,"53,288"																						
14	11-02-2022,Air India,AI,504,21:35,Delhi,11h 00m,"1-stop																						
15	"08:35,Mumbai,"56,081"																						
16	11-02-2022,Air India,AI,807,17:20,Delhi,15h 15m,"1-stop																						
17	"14:30,Mumbai,"51,731"																						
18	11-02-2022,Vistara,UK,977,19:00,Delhi,02h 15m,non-stop,21:15,Mumbai,"53,288"																						
19	11-02-2022,Air India,AI,504,21:35,Delhi,11h 00m,"1-stop																						
20	"17:25,Mumbai,"51,059"																						
21	11-02-2022,Vistara,UK,871,20:35,Delhi,17h 55m,"1-stop																						
22	"14:30,Mumbai,"51,731"																						
23	11-02-2022,Vistara,UK,977,19:00,Delhi,02h 15m,non-stop,21:15,Mumbai,"53,288"																						
24	11-02-2022,Air India,AI,504,21:35,Delhi,11h 00m,"1-stop																						
25	"08:35,Mumbai,"56,081"																						
26	11-02-2022,Air India,AI,807,17:20,Delhi,15h 15m,"1-stop																						
27	"14:30,Mumbai,"51,731"																						

Resim 2.a (İlk veri setine ait .csv uzantılı dosyası bussines class bilgileri)

The screenshot shows a Microsoft Excel spreadsheet titled "economy.csv". The data consists of approximately 27 rows of flight information. The columns include:

- date
- airline
- ch_code
- num
- code
- dep_time
- from
- time_taken
- stop
- arr_time
- to
- price

Sample data from the spreadsheet:

```

1 11-02-2022,Spicejet,SG,56,709,18:55,Delhi,O2h 10m,non-stop,21:05,Mumbai,"5,953"
2 11-02-2022,Spicejet,SG,5157,06:20,Delhi,O2h 20m,non-stop,08:40,Mumbai,"5,953"
3 11-02-2022,AirAsia,5,764,04:25,Delhi,O2h 10m,non-stop,06:35,Mumbai,"5,956"
4 11-02-2022,Vistara,UK,995,10:20,Delhi,O2h 15m,non-stop,12:35,Mumbai,"5,955"
5 11-02-2022,Vistara,UK,963,08:50,Delhi,O2h 20m,non-stop,11:10,Mumbai,"5,955"
6 11-02-2022,Vistara,UK,945,11:40,Delhi,O2h 20m,non-stop,14:00,Mumbai,"5,955"
7 11-02-2022,Vistara,UK,927,09:30,Delhi,O2h 05m,non-stop,11:35,Mumbai,"6,060"
8 11-02-2022,Vistara,UK,951,11:40,Delhi,O2h 10m,non-stop,16:30,Mumbai,"6,060"
9 11-02-2022,Vistara,UK,951,14:20,Delhi,O2h 02m,non-stop,10:10,Mumbai,"5,954"
10 11-02-2022,GO FIRST,GB,334,08:00,Delhi,O2h 10m,non-stop,10:10,Mumbai,"5,954"
11 11-02-2022,GO FIRST,GB,336,14:20,Delhi,O2h 15m,non-stop,16:35,Mumbai,"5,954"
12 11-02-2022,GO FIRST,GB,392,15:00,Delhi,O2h 15m,non-stop,17:15,Mumbai,"5,954"
13 11-02-2022,GO FIRST,GB,338,10:30,Delhi,O2h 20m,non-stop,12:50,Mumbai,"5,954"
14 11-02-2022,Indigo,6E,5001,07:15,Delhi,O2h 10m,non-stop,09:25,Mumbai,"5,955"
15 11-02-2022,Indigo,6E,6202,12:00,Delhi,O2h 02m,non-stop,14:10,Mumbai,"5,955"
16 11-02-2022,Indigo,6E,549,14:20,Delhi,O2h 15m,non-stop,16:35,Mumbai,"5,955"
17 11-02-2022,Indigo,6E,6278,08:45,Delhi,O2h 20m,non-stop,11:05,Mumbai,"5,955"
18 11-02-2022,Air India,A887,07:00,Delhi,O2h 05m,non-stop,09:05,Mumbai,"5,955"
19 11-02-2022,Air India,A665,08:00,Delhi,O2h 10m,non-stop,10:10,Mumbai,"5,955"
20 11-02-2022,AirAsia,5,747,19:00,Delhi,12h 15m,"1-stop
21
22 "07:15 Mumbai,"5,949"
23 11-02-2022,AirAsia,5,747,19:00,Delhi,16h 20m,"1-stop
24
25 "11:30 Mumbai,"5,949"
26 11-02-2022,GO FIRST,GB,266,06:25,Delhi,11h 45m,"1-stop
27
28

```

Resim 2.b (İlk veri setine ait .csv uzantılı dosyası economy class bilgileri)

The screenshot shows four Microsoft Excel windows side-by-side, each displaying a .csv file with flight data:

- economy**: Shows flight data for the economy class across various routes.
- Delhi**: Shows flight data specifically for routes originating from or ending at Delhi.
- Mumbai**: Shows flight data specifically for routes originating from or ending at Mumbai.
- Kolkata**: Shows flight data specifically for routes originating from or ending at Kolkata.

Each window has its own set of columns and data, reflecting the specific flight routes and times for that city.

Resim 3.a (İkinci veri setine ait her şehrin uçuş rotasına göre ayrılmış .csv uzantılı dosyaları)

Airline	e_of_Jour	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	total_Stop	additional_Ir	Price
IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEI	22:20	01:10 22.4h 50m	non-stop	No info		3897
IndiGo	1/05/2019	Kolkata	Banglore	CCU → KFB	05:50	13:15 7h 25m	2 stops	No info		7662
IndiGo	9/06/2019	Banglore	Delhi	DEL → LKC	09:25	04:25 10 J1 19h	2 stops	No info		13882
IndiGo	12/05/2019	Kolkata	Banglore	CCU → NA	18:05	23:30 5h 25m	1 stop	No info		6218
IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAI	16:50	21:35 4h 45m	1 stop	No info		13302
SpiceJet	24/06/2019	Kolkata	Banglore	CCU → BL	09:00	11:25 2h 25m	non-stop	No info		3873
Jet Airways	12/03/2019	Banglore	New Delhi	BLR → BO	18:55	10:25 13 N15h 30m	1 stop	In-flight m		11087
Jet Airways	01/03/2019	Banglore	New Delhi	BLR → BO	08:00	05:02 N 21h 5m	1 stop	No info		22270
Jet Airways	12/03/2019	Banglore	New Delhi	BLR → BO	08:55	10:25 13 N 25h 30m	1 stop	In-flight m		11087
Multiple c	27/05/2019	Delhi	Cochin	DEL → BO	11:25	19:15 7h 50m	1 stop	No info		8625
Air India	1/06/2019	Delhi	Cochin	DEL → BLF	09:45	23:00 13h 15m	1 stop	No info		8907
IndiGo	18/04/2019	Kolkata	Banglore	CCU → BL	20:20	22:55 2h 35m	non-stop	No info		4174
Air India	24/06/2019	Chennai	Kolkata	MAA → CC	11:40	13:55 2h 15m	non-stop	No info		4667
Jet Airways	9/05/2019	Kolkata	Banglore	CCU → BC	21:10	09:20 10 N 12h 10m	1 stop	In-flight m		9663
IndiGo	24/04/2019	Kolkata	Banglore	CCU → BL	17:15	19:50 2h 35m	non-stop	No info		4804
Air India	3/03/2019	Delhi	Cochin	DEL → AM	16:45	19:15 04 N 26h 35m	2 stops	No info		14011
SpiceJet	15/04/2019	Delhi	Cochin	DEL → PNW	08:45	13:15 4h 30m	1 stop	No info		5830
Jet Airways	12/06/2019	Delhi	Cochin	DEL → BO	14:00	12:35 13 J1 22h 35m	1 stop	In-flight m		10262
Air India	12/06/2019	Delhi	Cochin	DEL → CCI	20:15	19:15 13 J1 23h	2 stops	No info		13381
Jet Airways	27/05/2019	Delhi	Cochin	DEL → BO	16:45	12:35 28 N 20h 35m	1 stop	In-flight m		12898
GoAir	6/03/2019	Delhi	Cochin	DEL → BO	14:10	19:20 5h 10m	1 stop	No info		19495
Air India	21/03/2019	Banglore	New Delhi	BLR → CO	22:45	13:20 19 N 15h 20m	1 stop	No info		6955
IndiGo	3/04/2019	Banglore	Delhi	BLR → DEI	04:45	06:50 2h 50m	non-stop	No info		3943
IndiGo	1/05/2019	Banglore	Delhi	BLR → DEI	18:55	21:50 2h 55m	non-stop	No info		4823
Jet Airways	6/05/2019	Kolkata	Banglore	CCU → BC	18:55	08:15 07 N 13h 20m	1 stop	In-flight m		7757
Jet Airways	9/06/2019	Delhi	Cochin	DEL → IDR	21:25	12:35 09 J1 15h 10m	2 stops	No info		13292

Resim 4.a (Üçüncü veri setine ait .xlsx uzantılı dosyası)

Bu birbirlerine ana hatlar olarak yakın olan veri setlerinin işlenmesi ve analizi için Burak ERGENE'nin yazısından (Kaynaklar - 4) edinilen veri analizi ve veri işleme teknikleri ile 3 veri seti işlenerek amaca uygun kullanabilecek tek bir veri setine dönüştürüldü. Bu işlem yapılırken gün-ay-yıl verilerinden gün, ay ve yıl verileri ayrılarak 3 türetilmiş özellik oluşturuldu ve nümerik olarak değiştirildi. Aynı işlem kalkış saatı verisini saat ve dakika olarak ikiye ayırarak ve nümerik yapıya çevirerek tekrarlandı. Aynı işlem iniş saatı verisini saat ve dakika olarak ikiye ayırarak ve nümerik yapıya çevirerek tekrarlandı. Yapılan bu işlevlerden sonra kalkış ve iniş saatinde saatin veya dakikanın olmaması durumundan ortaya çıkan boş veri değerleri düzeltildi. Havada kalma süresi de saat ve dakika olarak ikiye ayrıldı ve nümerik değere dönüştürüldü. Fiyat bilgisine etkisinin az olduğu bilinen ve veri seti farklılıklarından dolayı bazı özellikler kaldırıldı. Uçak bilet fiyati farklı veri çeşidine sahip değerler nümerik değişkenlere çevrildi. Airline, Destination ve Source gibi değerler sözlük yapısıyla anlaşıldığı verileri nümerik yapıya çevirdi. Çevirme islevinde kullanılan sözlük yapıları:

```
In [ ]: airlines = [
    'Trujet',
    'SpiceJet',
    'Air Asia',
    'IndiGo',
    'GoAir',
    'Vistara',
    'Vistara Premium economy',
    'Air India',
    'Multiple carriers',
    'Multiple carriers Premium economy',
    'Jet Airways',
    'Jet Airways Business',
    'Akasa Air',
    'GO FIRST',
    'StarAir'
]

dict_airlines = {key: index for index, key in enumerate(airlines, 0)}
```

Resim 5.a (Airline(Havayolu şirketi) özelliği için kullanılmış sözlük yapısı)

```
In [ ]: dest = ['Kolkata',
   'Hyderabad',
   'Delhi',
   'Banglore',
   'Cochin',
   'Mumbai',
   'Chennai'
]

dict_dest = {key: index for index, key in enumerate(dest, 0)}
```

Resim 5.b (Destination(Varış noktası) özelliği için kullanılmış sözlük yapısı)

```
In [ ]: dest = ['Kolkata',
   'Hyderabad',
   'Delhi',
   'Banglore',
   'Cochin',
   'Mumbai',
   'Chennai'
]

dict_dest = {key: index for index, key in enumerate(dest, 1)}
```

Resim 5.c (Source(Kalkış noktası) özelliği için kullanılmış sözlük yapısı)

Veri seti son halini aldığında veri setinin en iyi halini bulmak için one-hot teknğini uygulayarak ve aykırı verileri yok ederek yapılan veri işlemlerinde değerlerin 2 aralıkta kaldığı görüldü. Bunun nedeninin fiyat bilgisine az etki ettiği gereklisiyle kaldırılan class özelliğinin (bussines - economy) aslında 1 veri seti için fiyat bilgisine çok etkisi olduğu anlaşılırak class özelliği yeniden eklendi. Eklenirken class özelliği için yeniden küçük bir sözlük yapısı uygulandı:

```
In [ ]: class = [
   'economy',
   'business'
]

dict_airlines = {key: index for index, key in enumerate(class, 0)}
```

Resim 5.d (class(sınıf) özelliği için kullanılmış sözlük yapısı)

Bu şekilde hem 3 veri seti birleştirilmiş hem de tüm değerleri nümerik hale getirilmiş oldu. Sonradan fark edilen bazı takvimlerin normal yazış biçiminde yazılıp bazlarının farklı yazılması hatası da ortadan kalktıktan sonra elimizdeki veri seti alttaki resimde görüldüğü gibi oldu.

Resim 6.a (Veri setinin kullanıma hazır olan son hali)

Bununla beraber class özelliğin eklenmesiyle beraber uçak fiyatını en çok etkileyen özelliklere bakıldığından:

Özellik sıralaması:

```

class          0.879398
Duration_hours 0.045506
Airline         0.011029
Journey_day    0.010394
Source          0.009382
Destination     0.008226
Journey_month   0.007248
Arrival_Time_hour 0.006918
Dep_Time_hour   0.006348
Arrival_Time_minute 0.005625
Dep_Time_minute 0.004657
Duration_mins   0.004257
Journey_year    0.001012
dtype: float64

```

Resim 7.a (Veri setindeki özelliklerin uçak fiyatına etkileri)

Python içerisindeki hazır Karar ağaçları, regresyon gibi modelleri kullanarak hangi özelliğe one-hot uygulayıp uygulanamayacağına bakıldı. Verilen modellerden Airline, Destination ve Source özelliklerine one-hot uygulanabilir mi o araştırıldı. Airline içerisindeki özellik adet sayısı fazla olduğu için böyle bir özelliğe one-hot uygulamak sütun sayısını gereğinden fazla uzatacağı için ve Destination ile Source özelliklerinin ikisine de one-hot uygulamak aynı nedenden gereksiz olacağının düşünüldü. Bu yüzden Python içerisinde hazır modellere göre hangisi uzayınca daha mantıklı oluyor ona bakıldı ve bunun sonucunda Source özelliğinin one-hot yapılmasına karar kılındı. En son hale gelen veri seti:

Resim 8.a (Veri setine one-hot uygulanmış hali)

Elde edilen veri seti ile farklı modeller değerlendirilip en iyi model seçilmeye çalışıldı fakat veri setinin büyülüğu ve model eğitiminin zorluğu nedeniyle veri setinin hazırlanıldığı ortam olan jupyter notebook'ta bu adımı gerçekleştiremeyerek farklı python yazım alanları araştırıldı. Bununla beraber Google colab kullanılarak ayrı bir ram ve disk sağlanılıp model eğitimi devam edilebildi. İlk olarak verilerin hepsinin nümerik olma nedenlerinden biri olan Çoklu lineer regresyon modeli kullanılarak ilk model eğitimi yapıldı.

Coklu Lineer Regresyon modeli kodları istenildiği gibi çalışması için eğitim veri seti ve test veri setine ayrılmıştır. Bu ayrı çok fazla veri olduğu için eğitim verisine verinin %80'i ve test verisine verinin %20'si olacak şekilde ayrılmıştır. Çoklu regresyon modelinin parametreleri belirlenmiştir. Öğrenme katsayısı 0.01, iterasyon sayısı 1000 ve hata eşiği 0.0001 olarak belirlenmiştir. Öğrenme katsayısunın küçük ve iterasyon sayısının fazla olması çoklu regresyon modelinde optimum katsayırlara daha yakın olmasını amaçlayarak belirlenmiştir. Hata eşiği, istenilen katsayıların elde edilmesi veya istenilenlere yakın katsayıların elde edilmesi aşamasında regresyon modelini sonlandıran bir parametredir. Çoklu lineer regresyon modelini bu parametrelerle çalıştırılmasıyla elde edilen sonuç, test verisinde r^2 parametresinin 0.853 olmasını göstermesidir. Bu 0.853 parametresi de bize çoklu lineer regresyon modelimizin %85'lik bir başarı oranı gösterdiğini ifade eder.

Karar ağacı modeli kodlarının istenildiği gibi çalışması için veri seti, eğitim ve test veri seti olarak ayrıldı. Bu ayrı, büyük miktarda veri olduğu için, eğitim verisine %80 ve test verisine %20 olacak şekilde yapıldı. Karar ağacı modelinin parametreleri belirlendi. Modelin ne kadar örnekte alt dallara ayrılacağını belirleyen `min_samples_split` parametresi 10 olarak belirlendi. Modelin ne kadar derinleşeceğini gösteren `max_depth` parametresi 5 olarak belirlendi.

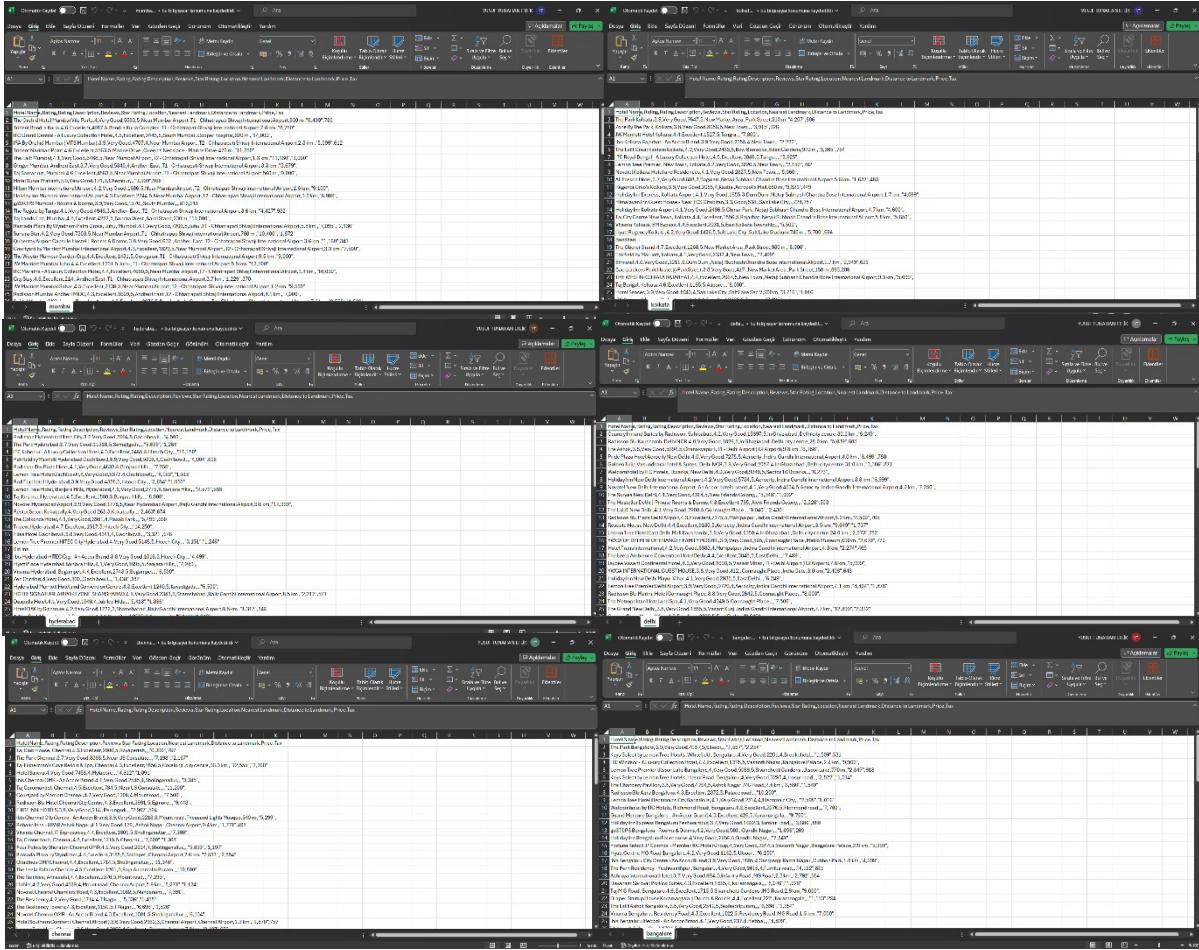
Bu parametrelerle çalıştırılan karar ağaç modelinde, eğitim setindeki r^2 parametresi 0.980, test setindeki r^2 parametresi ise 0.234 olarak ölçüldü. Bu değerler, aşırı öğrenme (overfitting) olduğunu gösterdi. Max_depth parametresini artırmak veya azaltmak aşırı öğrenme sorununu çözmedi. Bununla birlikte, aşırı öğrenme sorununun üstesinden gelmek için daha zor olan rastgele orman modeli denendi.

Rastgele Orman modeli ile yapılan denemedede, diğer yazılan kodlardan daha uzun bir süre çalışmasına rağmen istenilen sonuca yakın sonuç elde edildi. MSE değeri 1635.785619 ve r^2 değeri 0.7481493 olarak çıktı. Rastgele Orman modelinde, modele verilen tek parametre olan n_estimators (Toplam karar ağaçları sayısı) kullanıldı. Bu parametre, Rastgele Orman modelindeki karar ağaçları sayısını gösterir. Elde edilen sonuçlar incelendiğinde, en iyi sonucu veren modelin Çoklu Lineer Regresyon modeli olduğu tespit edildi.

Kullanıcı dostu ve kullanılabilir bir arayüz sağlamak amacıyla, çoklu lineer regresyon modelinde fiyat tahmini için gerekli olan tüm verilerin kullanıcı tarafından girilmesi gerekmektedir. Ancak, bu durum, kullanıcıların genellikle bilmediği veya ilgilenmediği verilerin girdirmesine neden olacağından, arayüzü karmaşık hale getirebilirdi. Bu nedenle, kullanıcı dostu bir arayüz hedeflenerek, sadece fiyat tahmini yapmak yerine varış saatı ve dakikası, uçakta kalma süresi gibi verileri de tahmin ederek başarı oranının biraz azalması göze alınarak çok değişkenli lineer regresyon modeli oluşturulmaya karar verildi.

Uçak bilet fiyatlını tahmin etmek için gerekli olan veri seti ve model işlemlerinden sonra, otel veri setinin bulunması ve işlenmesi gerekiyordu. Bu veri seti, uçak bilet veri setini bulduğum scraper yardımıyla bir internet sitesinden çekilmiş şekilde bulundu. Otel veri seti fazla değildi ve içerisinde yapay zeka modeli eğitimi için uygun olmayan veriler bulunuyordu; aynı zamanda otellerin en yakın havaalanına uzaklığını gibi bilgiler de bu veriler arasında dayandı. Bu nedenle, hem taksi verisinin az olmasından dolayı hem de havaalanı ve otel mesafe verisinin mevcut olmasından dolayı, bu verileri birleştirerek tek bir yapay zeka modelinde hem taksi hem de otel fiyat tahminini yapmak amaçlandı. Öncelikle otel verisi, şehirlere göre farklı farklı veri setleri içeriyeındı ve bu veriler tek bir veri setinde birleştirildi.

Bu veri setleri birleştirildikten sonra veri setindeki otellerin bulunduğu şehre göre veri setlerine yeni şehir sütunu eklenildi. Model geliştirilirken kullanılmayacak bilgiler hariç kalan kategorik verilere one-hot encoding yöntemi uygulandı ve veri seti taksi bilgilerini de ekleyerek yeniden düzenlenildi. Veri setinin son hali bu duruma getirildi.



Resim 9.a (Otel veri setinin bulunduğu şehre göre ayrılmış veri setleri)

A1	Rating_Reviews	Star_Rating	Price	city_Bangalore	city_Chennai	city_Delhi	city_Hyderabad	city_Kolkata	city_Mumbai	Rating_Description_Average	Rating_Description_Excellent	Rating_Description_Good	Rating_Description_Poor	Rating_Description_Very_Good	Hotel_Name	Location	Nearest_Landmark	Distance_to_Landmark	taxi_Start_x	taxi_Start_y	km
2	4.0	3.5	7279	False	False	False	True	False	False	True	False	True	False	True	The Orchid Hotel Mumbai Vile Parle	Near Mumbai Airport	Chhatrapati Shivaji International Airport	320,29,0,30,0			
4	4.0	5.1044	7279	False	False	False	False	False	False	True	False	False	False	True	ITC Grand Central - A Luxury Collection Hotel, South Mumbai	Near Mumbai Airport	Chhatrapati Shivaji International Airport	17500,29,28,30,0			
5	4.5	4957.5	15448	False	False	False	False	False	False	True	False	False	False	True	ITC Maratha	Kundan Kunj	Colaba	Chhatrapati Shivaji International Airport	320,29,0,30,0		
3	4.7077	4.7111	7279	False	False	False	True	False	False	True	False	True	False	True	ITC Grand Central - A Luxury Collection Hotel, South Mumbai	Near Mumbai Airport	Chhatrapati Shivaji International Airport	2300,29,29,30,0			
6	4.6	5565.5	12030	False	False	False	False	False	False	True	False	False	False	True	Taj Mahal Palace & Tower	Santacruz	Marine Drive	Chhatrapati Shivaji International Airport	23470,29,29,30,0		
1	4.1	6469.5	14169	False	False	False	False	False	False	True	False	False	False	True	Taj Mahal Palace & Tower	Near Mumbai Airport	Chhatrapati Shivaji International Airport	1300,29,0,30,0			
7	3.7	5340.4	5197	False	False	False	True	False	False	True	False	False	False	True	'Ginger' Mumbai, Andheri East	Andheri East	Chhatrapati Shivaji International Airport	3300,29,29,30,0			
8	4.6	4565.5	10124	False	False	False	True	False	False	True	False	False	False	True	Taj Samudra	Bandra West	Chhatrapati Shivaji International Airport	560,29,0,30,0			
9	3.8	121.3	4907	False	False	False	True	False	False	True	False	False	False	True	Hyatt Regency Mumbai	Near Mumbai Airport	Chhatrapati Shivaji International Airport	3500,29,0,30,0			
10	4.2	1686.5	5999.1	False	False	False	True	False	False	True	False	False	False	True	Hilton Mumbai International Airport	Near Mumbai Airport	Chhatrapati Shivaji International Airport	1600,29,0,30,0			
12	4.3	2744.5	7932	False	False	False	True	False	False	True	False	False	False	True	Hilton Mumbai International Airport	Near Mumbai Airport	Chhatrapati Shivaji International Airport	3000,29,0,30,0			
8	3.9	1376.3	31027	False	False	False	True	False	False	True	False	False	False	True	goSTOPS Mumbai	Rooms & Deems, South Mumbai	Chhatrapati Shivaji International Airport	2200,29,0,30,0			
11	4.1	4946.3	33539	False	False	False	True	False	False	True	False	False	False	True	The Regale by Tunga	Andheri East	Chhatrapati Shivaji International Airport	3600,29,0,30,0			
5	4.5	4222.5	12324	False	False	False	True	False	False	True	False	False	False	True	Taj Lands End, Mumbai	Bandra West	Chhatrapati Shivaji International Airport	8600,29,0,30,0			
11	4.1	7206.5	51301	False	False	False	True	False	False	True	False	False	False	True	Ramada Plaza By Wyndham Palm Grove Juhu	Juhu	Chhatrapati Shivaji International Airport	5800,29,0,30,0			
8	4.6	6322.5	15301	False	False	False	True	False	False	True	False	False	False	True	Radisson Blu Hotel, Mumbai	Near Mumbai Airport	Chhatrapati Shivaji International Airport	3600,29,0,30,0			
3	4.3	1629.5	8105	False	False	False	True	False	False	True	False	False	False	True	Capella Hotel, Mumbai	Near Mumbai Airport	Chhatrapati Shivaji International Airport	5500,29,0,30,0			
4	4.4	1431.5	9008	False	False	False	True	False	False	True	False	False	False	True	The Westin Mumbai Garden City, Gurgaon	Chhatrapati Shivaji International Airport	5800,29,0,30,0				
4	4.4	1731.5	10307	False	False	False	True	False	False	True	False	False	False	True	Marriott Mumbai Juhu	Juhu	Chhatrapati Shivaji International Airport	5500,29,0,30,0			
4	4.4	4660.5	10642	False	False	False	True	False	False	True	False	False	False	True	ITC Maratha - A Luxury Collection Hotel	Near Mumbai Airport	Chhatrapati Shivaji International Airport	1400,29,29,30,0			
6	4.6	214.3	11598	False	False	False	True	False	False	True	False	False	False	True	City Stay Andheri East	Andheri East	Chhatrapati Shivaji International Airport	3700,29,0,30,0			
5	4.5	2130.5	10490	False	False	False	True	False	False	True	False	False	False	True	JW Marriott Mumbai Sahar	Near Mumbai Airport	Chhatrapati Shivaji International Airport	4100,29,0,30,0			
5	4.3	3609.5	7855	False	False	False	True	False	False	True	False	False	False	True	Radisson Mumbai Andheri MIDC	Andheri East	Chhatrapati Shivaji International Airport	4100,29,0,30,0			
1	4.2	2327.5	16000	False	False	False	True	False	False	True	False	False	False	True	Marriott Mumbai Juhu	Juhu	Chhatrapati Shivaji International Airport	5700,29,0,30,0			

Resim 10.a (Otel ve taksi veri setinin son hali)

Otel verisi için çoklu regresyon, Karar Ağacı ve Karmaşık Orman yapay zeka modelleri uygulanarak %78,941 başarı oranı elde edilmiş ve sonuç olarak çoklu regresyon modeli seçilmiştir. Çoklu lineer regresyon modelinde otel fiyat tahmini için gerekli olan verilerin kullanıcı tarafından girilmesi gerekmektedir, ancak bu kullanıcı dosta olmayan bir arayüz sağlayabilirdi. Bu nedenle, sadece fiyat tahmini yapmak yerine otel fiyatı, otel internet sitesi yıldızı, otel internet sitesi olumlu yorumları gibi verileri de tahmin ederek başarı oranının az da olsa azalması göze alınarak çok değişkenli lineer regresyon modeli oluşturulmaya karar verildi. Bu şekilde, kullanıcı dosta ve kullanılır bir arayüz hedeflenmiştir. Her iki yapay zeka modelinde de çoklu regresyon modeli kullanılarak işlem yapıldığında yapay zeka modelinin %59.632'li başarı oranıyla düşüş yaptığı fark edildi ve ayrı ayrı kullanılmaya karar verildi.

Proje kapsamında, kullanıcı arayüzünün oluşturulması için React kullanıldı. React, kullanıcı etkileşimlerine hızlı tepki verebilen, bileşen tabanlı bir kütüphane olarak tercih edildi. Kullanıcı tarafından sağlanan seyahat bilgilerini toplamak ve sunucuya etkileşim kurmak amacıyla useState, useEffect, ve useCallback kancaları (hooks) kullanıldı.

Seyahat Bütçe Tahmini
Uygulaması

Hava Yolu Şirketi:
Seçiniz

Kalkış Noktası:
Seçiniz

Varış Noktası:
Seçiniz

Seyahat Sınıfı:
Seçiniz

Tarih ve Saat:
gg . aa . yyyy - - : --

Otel Yıldızları:
Seçiniz

Derecelendirme:
Seçiniz

Gönder

Resim 11.a (Projenin temel arayüzü)

Bu projede, sade ve kullanıcı dostu bir tasarım hedeflenmiştir. Arka plan rengi açık gri, metin rengi ise koyu gri olarak belirlenmiş; Arial ve sans-serif font aileleri tercih edilmiştir. Başlıklar mavi tonlarıyla renklendirilmiş ve ortalanmıştır. Form elemanları flexbox ile dikey hizalanmış, label etiketleri mavi tonlarıyla vurgulanmıştır.

The screenshot shows a flight search form titled "Seyahat Bütçe Tahmini Uygulaması". The form includes dropdown menus for "Hava Yolu Şirketi" (Air Asia), "Kalkış Noktası" (Hyderabad), "Varış Noktası" (Delhi), "Seyahat Sınıfı" (Economy), and a date input field showing "01.06.2024 05:45". Below these are dropdowns for "Otel Yıldızları" (Select) and a star rating component showing 5 stars.

Resim 11.b (Projenin form doluyken arayüzü)

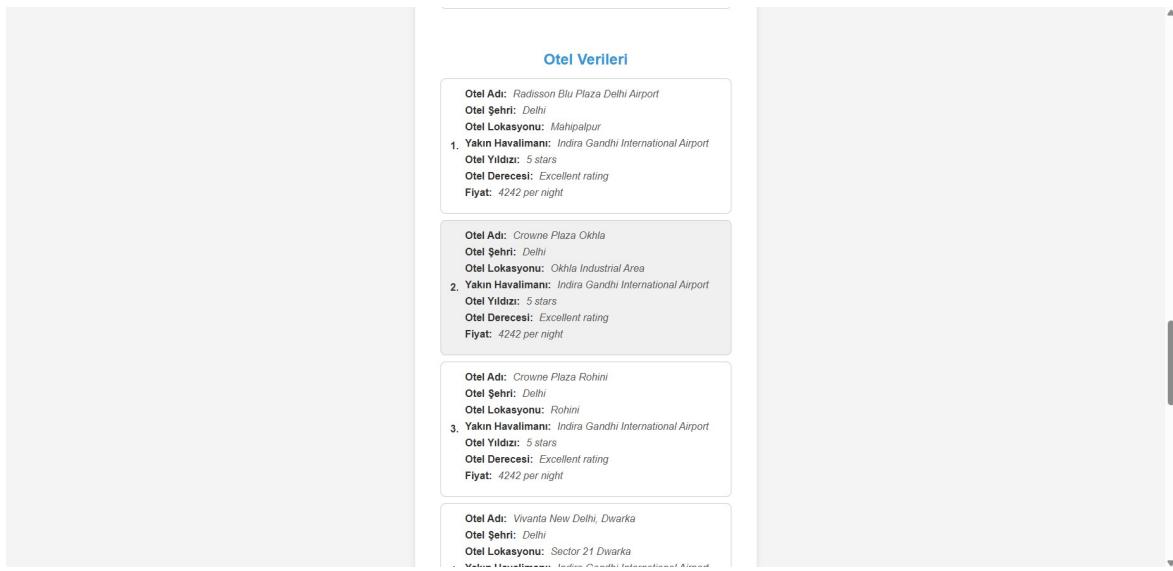
Kullanıcı etkileşiminin kolaylaşımak amacıyla seçim kutuları ve metin giriş alanları geniş tutulmuş, submit butonları mavi tonlarla renklendirilmiş ve hover etkisiyle biraz daha koyu maviyeye dönüştürülmüş. Otel ve uçuş verilerini gösteren liste elemanları, hover etkisiyle arka plan renkleri değişecek şekilde tasarlanmış, seçili elemanlar belirgin hale getirilmiştir

The screenshot displays a table titled "Uçak Bileti Verileri" containing four rows of flight information. Each row includes a "Delete" button at the bottom right. The columns represent flight details: Hava Yolu Şirketi, Kalkış Noktası, Varış Noktası, Kalkış Tarihi, Kalkış Saati, and Fiyat.

	Hava Yolu Şirketi: Air Asia	Kalkış Noktası: Hyderabad	Varış Noktası: Delhi	Kalkış Tarihi: 2024-06-01	Kalkış Saati: 05:45	Fiyat: 1736.67
1	Hava Yolu Şirketi: Air Asia	Kalkış Noktası: Hyderabad	Varış Noktası: Delhi	Kalkış Tarihi: 2024-06-01	Kalkış Saati: 05:45	Fiyat: 1736.67
2	Hava Yolu Şirketi: Air Asia	Kalkış Noktası: Hyderabad	Varış Noktası: Delhi	Kalkış Tarihi: 2024-06-01	Kalkış Saati: 05:45	Fiyat: 1736.67
3	Hava Yolu Şirketi: Air Asia	Kalkış Noktası: Hyderabad	Varış Noktası: Delhi	Kalkış Tarihi: 2024-06-01	Kalkış Saati: 05:45	Fiyat: 1736.67
4	Hava Yolu Şirketi: Air Asia	Kalkış Noktası: Hyderabad	Varış Noktası: Delhi	Kalkış Tarihi: 2024-06-01	Kalkış Saati: 05:45	Fiyat: 1736.67
Hava Yolu Şirketi: Air Asia						

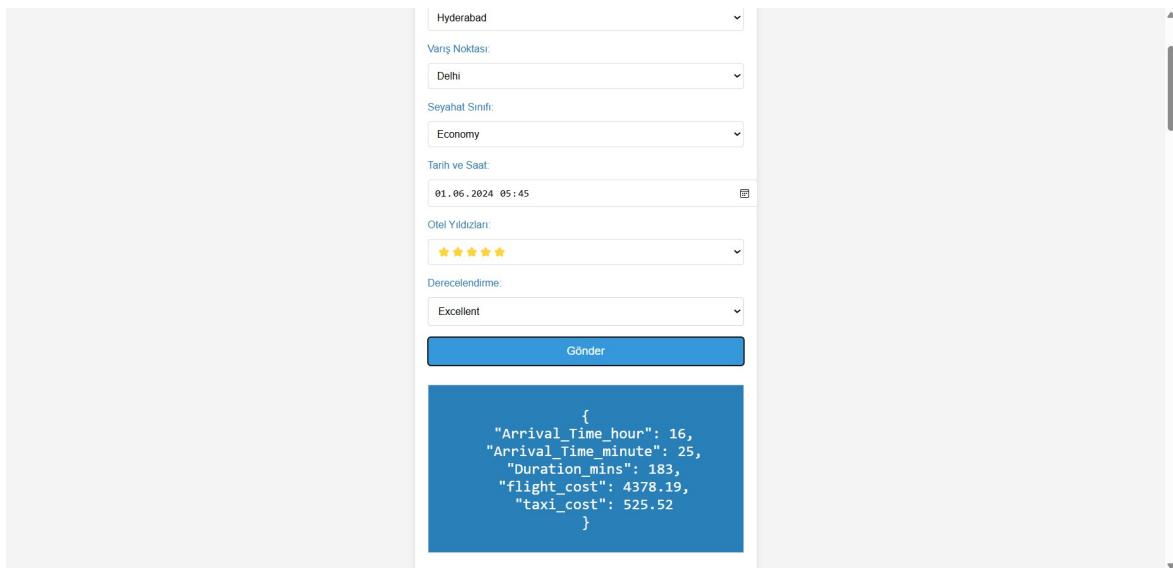
Resim 11.c (Veri tabanından alınan uçak bileti verileri)

Kullanıcıdan alınan veriler, useState kancası ile yönetilen formData adlı bir durum nesnesinde saklandı. Form bileşeni yüklenliğinde, useEffect kullanılarak uçuş ve otel verilerinin sunucudan çekilmesi sağlandı. Bu veriler, fetchFlightData ve fetchHotelData fonksiyonları aracılığıyla sunucudan alındı ve ilgili durum değişkenlerinde (flightData ve hotelData) saklandı. Bu sayede, kullanıcıya güncel uçuş ve otel bilgileri sunuldu.



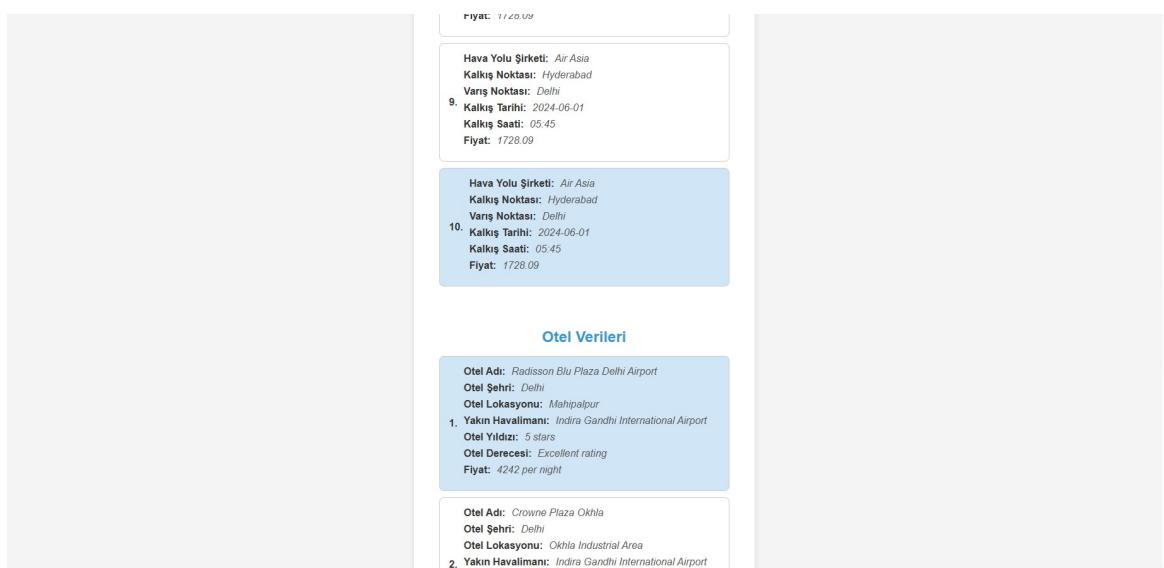
Resim 11.d (Veri tabanından alınan otel verileri)

Form gönderildiğinde, kullanıcı verileri JSON formatında sunucuya iletti ve sunucudan gelen yanıt serverResponse durum değişkeninde saklandı. Kullanıcıya seçilen uçuş ve otel bilgilerinin yanı sıra, sunucudan gelen yanıtlar gösterildi.



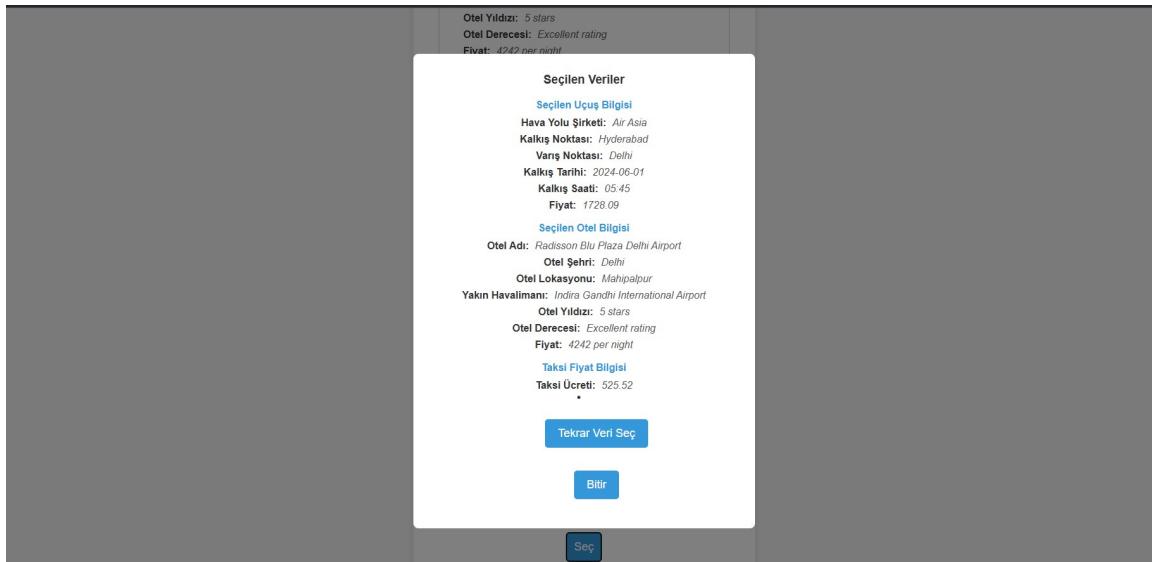
Resim 11.e (Sunucudan gelen yanıt verileri)

Seçim yapma, veri temizleme ve formun yeniden doldurulması gibi kullanıcı etkileşimleri, handleSelect, handleClearData, ve handleContinue gibi fonksiyonlarla yönetildi. Bu fonksiyonlar, ilgili durum değişkenlerini güncelleyerek arayüzdeki değişikliklerin yanında yansıtılmasını sağladı.



Resim 11.f (Gelen verilerden seçim yapma işlemi)

Pop-up bileşeni ekranın ortasında belirgin şekilde görünmesi için yerleştirilmiş, sunucu yanıtları mavi arka plan ve beyaz metinle ayrıstırılmıştır. Bu stillendirme işlemleri, kullanıcı deneyimini iyileştirmek ve etkileşimleri daha akıcı hale getirmek amacıyla yapılmıştır. Tüm detaylar, tasarımin sade, anlaşılır ve kullanıcı dostu olmasını sağlamak için gerçekleştirilmiştir.



Resim 11.g (Seçilen verilerin pop-up menüde gösterilmesi)

Sonuç olarak, proje kapsamında React kullanılarak dinamik ve kullanıcı dostu bir arayüz oluşturuldu. Sunucu ile etkileşimler asenkron fonksiyonlarla sağlanarak kullanıcı deneyimi iyileştirildi ve verilerin anlık olarak güncellenmesi sağlandı. Bu yapılarak, kullanıcıların seyahat planlaması için gerekli verileri kolayca yönetebilmeleri mümkün kılındı.

Yine bu proje kapsamında, Python ve Flask kullanılarak bir web sunucusu oluşturulmuştur. Sunucuya bağlanmak için MySQL veri tabanı kullanılmıştır. Veri tabanı bağlantısı mysql.connector kütüphanesi ile sağlanmıştır. Sunucu, kullanıcıdan gelen seyahat bilgilerini alarak bu bilgiler doğrultusunda bazı işlemler gerçekleştirmiştir.

İlk olarak, kullanıcıdan alınan seyahat tarihi ve saatı gibi bilgiler ayrıstırılmış ve doğrulanmıştır. Ardından, uçuş verileri işlenerek tahmini uçuş süresi ve maliyeti hesaplanmıştır. Bu hesaplamalar, önceden eğitilmiş bir model (joblib ile yüklenen) kullanılarak yapılmıştır. Uçuş verileri işlendikten sonra, uygun uçuşlar MySQL veri tabanına eklenmiştir. Eklenen veriler daha sonra geri alınarak kullanıcıya sunulmak üzere düzenlenmiştir.

The screenshot shows the MySQL Workbench interface with the 'seyahat_bileti' table selected. The table has columns: Hava_Yolu_Noktası, Kalkış_Noktası, Varış_Noktası, Seyahat_Tarifi, Kalkış_Tarifi, Varış_Saat, Varış_Saatı, Dovenme_Saat, Fiyat. The data grid displays 15 rows of flight information, such as Economy class flights from Air Asia to various destinations like Delhi, Mumbai, and Bangalore at different times and prices.

Resim 12.a (Seyahat veri tabanı uçak_bileti_tablosu)

Eğer kullanıcı otel verilerini de görmek isterse, otel bilgileri üzerinde benzer bir işlem gerçekleştirilmiştir. Kullanıcıdan alınan otel yıldız sayısı ve değerlendirme bilgileri doğrultusunda, otel fiyatları ve değerlendirmeleri tahmin edilmiştir. Bu veriler de aynı şekilde MySQL veritabanına eklenmiş ve geri alınarak kullanıcıya sunulmuştur. Ayrıca, otel ve havaalanı arasındaki taksi maliyetleri de hesaplanarak ek bir bilgi olarak kullanıcıya iletilmiştir.

Son olarak, veritabanındaki mevcut verilerin temizlenmesi ve sıfırlanması için gerekli işlemler yapılmıştır. Bu işlem, önceki verilerin silinmesi ve yeni verilerin eklenmesi ile gerçekleştirilmiştir. Tüm bu işlemler, kullanıcı deneyimini iyileştirmek ve doğru verileri sağlamak amacıyla titizlikle yapılmıştır.

The screenshot shows the MySQL Workbench interface with the 'otel' table selected. The table has columns: Otel_Adı, Otel_Sehir, Otel_IkinciAdres, Otel_Jakobs_havaborsası, Otel_Yıldızlı, Otel_Deneyimci, Otel_İstirat, Otel_reviewer. The data grid displays 15 rows of hotel information, including details like 'Otel İkinci Adres' and 'Otel İstirat' for various hotels in cities like Delhi, Mumbai, and Bangalore.

Resim 12.a (Seyahat veri tabanı otel_tablosu)

PROJENİN AMACI ve ÖNEMİ

Projenin Amacı: Projenin amacı, kullanıcıların uçak biletı, otel ve taksi fiyatlarını tahmin etme sürecini kolaylaştırmak ve kişiselleştirmek için yapay zeka destekli bir platform sunmaktadır. Bu platform, kullanıcıların seyahat planları yaparken uygun fiyatlı seçenekler bulmakta zorlandıkları veya farklı ulaşım ve konaklama kategorileri arasında gezinirken karar vermekte güçlük çektikleri durumlarda yardımcı olmayı hedeflemektedir. Kullanıcılar, uçak biletlerinden otellere ve taksi hizmetlerine kadar farklı kategorilerde zengin bir seçenek yelpazesi aracılığıyla, tercihlerine ve ihtiyaçlarına uygun öneriler alabileceklerdir.

Ayrıca, isteklerine göre bu kategorileri düzenleyebilecek, önerilen seçenekleri özelleştirebilecek ve hatta tahmin edilen fiyatlara uygun gerçek uçak biletı, otel ve taksi verilerini seçerek rezervasyonlarını kolaylaştırabileceklerdir. Bu şekilde, kullanıcılar daha verimli ve keyifli bir seyahat planlama deneyimi yaşayacaklar ve seyahatlerini daha ekonomik ve konforlu hale getirebileceklerdir.

Projenin Önemi: Bu proje, akademik, ticari ve sosyal açılarından büyük önem sahiptir. Akademik olarak, yapay zekâ teknolojisinin seyahat ve ulaşım sektöründeki uygulamalarını araştırmak ve geliştirmek için bir fırsat sunar. Bu proje, aynı zamanda turizm ve ulaşım endüstrisindeki yenilikçi yaklaşımları keşfetmek ve kullanıcı deneyimini artırmak için bir model olabilir.

Ticari olarak, bu proje, seyahat ve ulaşım sektöründe faaliyet gösteren işletmeler için müşteri memnuniyetini artırma ve gelirlerini artırma potansiyeli sunar. Uçak biletı, otel ve taksi fiyatlarını tahmin etme sistemi, kullanıcıların seyahat planlama sürecini kolaylaştırırken aynı zamanda turizm ve ulaşım sektörüne yönelik tüketici taleplerini anlamak için değerli veriler sağlayabilir.

Sosyal olarak, bu projenin çıktıları, seyahat planlamasına yardımcı olmak ve bireylerin daha uygun ve verimli seyahat seçenekleri ile karar vermelerini teşvik etmek açısından değerlidir. Bu çözüm, özellikle zamanı kısıtlı olan ve seyahat planlamasında yardıma ihtiyacı olan kişiler için değerli bir araç olabilir.

KAYNAK ARAŞTIRMASI

Bitirme projesi kapsamında yapılan kaynak araştırması öncelikle, uçak bilet fiyat tahminleri için gerekli veri setleri temin edilmiştir. Bu amaçla Kaggle sitesinde Shubham Bathwal tarafından hazırlanan "Flight Price Prediction" veri seti kullanıldı [1]. Ek olarak, Andrew Geeks tarafından GitHub'da paylaşılan "MakeMyTrip-scraper/flight_datasets" ve "MakeMyTrip-scraper/hotel_datasets" veri setlerinden de yararlanıldı [2][3]. Jillani Soft Tech'in Kaggle'da yayımladığı "Flight Price Prediction DataSet" ise diğer bir veri kaynağı olarak değerlendirildi [4].

Veri setlerinin işlenmesi ve analizi için gerekli bilgi ve beceriler edinildi. Burak ERGENE tarafından hazırlanan "Predict_Fare_of_Airlines_Ticket_w_Detailed_EDA_RF" adlı rehber, veri madenciliği sürecinde önemli bir kaynak olarak kullanıldı [5]. Python programlama dilini daha iyi anlayabilmek için Yazılım Bilimi kanalındaki Python ders videoları izlendi [6] ve Python'un resmi dokümantasyonu detaylı bir şekilde incelendi [7]. Google Colab üzerinde veri seti yükleme ve analiz yapabilmek için Mahesh Hudar'in hazırladığı YouTube videolarından faydalandırıldı [8].

Çevrimiçi eğitim almak için Miuul sitesinden dersler alındı [9]. Flask API geliştirme sürecinde, "Build A Full-Stack Web Application with Flask And ReactJS - Part 1 Flask API Development" adlı YouTube video serisi rehber olarak kullanıldı [10]. JavaScript diline dair ihtiyaç duyulan bilgiler, Mozilla'nın resmi dokümantasyonundan elde edildi [11]. Proje süresince, yapay zeka destekli dil modeli ChatGPT ile çeşitli sorular yanıtları ve gerekli bilgiler sağlandı [12].

Makine öğrenimi süreçlerinde kullanılan Scikit-learn kütüphanesinin resmi dokümantasyonu incelendi [13]. React ile ilgili bilgiler, React'in resmi dokümantasyonundan öğrenildi [14]. Modelin saklanması ve yüklenmesi işlemlerinde Joblib kütüphanesi kullanıldı ve ilgili dokümantasyon incelendi [15]. Web scraping işlemleri için Selenium kütüphanesi kullanılarak veri çekme işlemleri gerçekleştirildi ve bu süreçte resmi dokümantasyon rehber olarak alındı [16].

Bu şekilde, proje boyunca kullanılan tüm kaynaklar ve yapılan araştırmalar sayesinde kapsamlı ve detaylı bir çalışma gerçekleştirilmiştir.

MATERYALLER:

1. Python: Python, basit ve anlaşılabilir sözdizimi ile öne çıkan yüksek seviyeli bir programlama dilidir. Genel amaçlı olarak kullanılan Python, modüler tasarıımı sayesinde geniş bir kütüphane ve araç yelpazesi sunar. Dinamik tip sistemine sahip olan dil, nesne yönelimli programlama paradigmalarını destekler. Python, veri analizi, yapay zeka, web geliştirme ve bilimsel hesaplamalar gibi birçok alanda popüler bir tercih haline gelmiştir.



Resim 13.a (Python logosu)

2. React.js: Facebook tarafından geliştirilen ve kullanıcı arayüzlerini oluşturmak için kullanılan bir JavaScript kütüphanesidir. React, bileşen tabanlı bir yaklaşım benimser ve bu sayede kullanıcı arayüzlerini daha modüler ve yönetilebilir hale getirir. React uygulamaları, veri değişikliklerini algılayarak sadece etkilenen bileşenleri günceller, bu da performansı artırır. JSX adı verilen bir dil kullanarak JavaScript'e HTML benzeri bir sözdizimi ekleyerek, kullanıcı arayüzüne tanımlamak daha kolay ve anlaşılır hale gelir. React, büyük ve karmaşık uygulamaların geliştirilmesini daha kolaylaştırarak, geliştiricilere güçlü bir araç seti sunar.



Resim 13.b (React JS logosu)

3. Visual Studio Code: Microsoft tarafından geliştirilen ücretsiz ve açık kaynaklı bir kod düzenleyici ve geliştirme ortamıdır. Hafif yapısı ve geniş eklenti desteği ile geliştiricilere esnek bir çalışma ortamı sunar. VS Code, birçok programlama dilini destekler ve zengin özelliklere sahiptir, bunlar arasında otomatik tamamlama, hata denetimi, entegre git desteği ve kullanıcı dostu bir arayüz bulunmaktadır. Kullanıcılar, VS Code'u ihtiyaçlarına göre özelleştirebilir ve geniş eklenti galerisinden faydalananabilir. Hem hızlı hem de güçlü olmasıyla, geliştiriciler arasında popüler bir tercih haline gelmiştir.



Resim 13.c (Visual Studio Code logosu)

4. Python Flask: Web uygulamaları geliştirmek için kullanılan hafif ve esnek bir web çerçevesidir. Flask, minimal ve modüler bir tasarıma sahiptir, bu da geliştiricilere ihtiyaçlarına göre özelleştirebilecekleri bir yapı sunar. Jinja2 şablon motoru ve Werkzeug WSGI toolkit gibi bileşenleri kullanarak, temel web uygulamalarından daha karmaşık projelere kadar geniş bir yelpazede kullanılabilir. Flask, RESTful API'lerden tam teşekkülü web uygulamalarına kadar çeşitli kullanım senaryolarını destekler. Ayrıca, geniş bir topluluk ve dökümantasyon desteği sayesinde Flask, Python geliştiricileri arasında popüler bir tercih haline gelmiştir.



Resim 13.d (Python Flask logosu)

5. XAMPP: XAMPP, Apache Friends tarafından geliştirilen, açık kaynaklı ve platformlar arası uyumlu bir web sunucusu paketidir. İçerisinde Apache HTTP Server, MariaDB (önceki MySQL), PHP ve Perl gibi yazılımlar barındırır. Geliştiricilerin yerel ortamlarında web uygulamaları geliştirebilmesi ve test edebilmesi için tasarlanmıştır. XAMPP, kolay kurulumu ve kullanımı sayesinde hem yeni başlayanlar hem de deneyimli geliştiriciler tarafından yaygın olarak tercih edilmektedir.



Resim 13.e (Python Flask logosu)

METOTLAR:

1. One-hot yöntemi: One-hot kodlama yöntemi, kategorik verileri makine öğrenimi modelleri için uygun bir formata dönüştürmek amacıyla kullanılan bir tekniktir. Bu yöntem, özellikle sınıflandırma problemleri için yaygın olarak kullanılır. Veri setinde bulunan kategorik bir değişken, benzersiz kategorilerine göre ayrılır ve her bir kategori için bir vektör oluşturulur. Bu vektörde, ilgili kategoriye ait indeks 1, diğer indeksler ise 0 değerini alır. Böylece, model kategorik bilgileri daha iyi anlayabilir ve öğrenebilir. One-hot kodlama, kategorik verilerin modelin anlayabileceği sayısal bir forma dönüştürülmesini sağlayarak, sınıflandırma veya regresyon gibi makine öğrenimi görevlerinde daha etkili sonuçlar elde etmeye yardımcı olur.
2. Çoklu Lineer Regresyon modeli: Çoklu Lineer Regresyon, istatistiksel bir modelleme tekniğidir ve bir bağımlı değişken ile bir veya daha fazla bağımsız değişken arasındaki ilişkiyi analiz etmek için kullanılır. Bu yöntem, birden fazla özellik veya değişkenin bağımlı değişken üzerindeki etkilerini ölçmek amacıyla tasarlanmıştır. Veri setindeki değişkenler arasındaki ilişkiyi modellemek ve bu ilişkileri kullanarak bağımlı değişkenin tahminini yapmak için kullanılır. Örneğin, pazarlama harcamaları, reklam bütçesi ve satışlar arasındaki ilişkiyi anlamak veya açıklamak için çoklu lineer regresyon kullanılabilir. Bu yöntem, gerçek dünyadaki karmaşık ilişkileri anlamak ve tahminlerde bulunmak için güçlü bir araçtır.

3. R^2 (R kare): Genellikle bir regresyon modelinin performansını ölçmek için kullanılan bir istatistiksel bir terimdir. R^2 , bağımsız değişkenlerin bağımlı değişken üzerindeki varyansı ne kadar iyi açıkladığını gösteren bir ölçütür. Bir regresyon modeli, bir bağımlı değişkenin değerini tahmin etmek için bir veya daha fazla bağımsız değişkeni kullanır. R^2 , bu tahminin gerçek değerlere ne kadar yakın olduğunu ölçer. Değerler genellikle 0 ile 1 arasında olup, 1'e ne kadar yakınsa, modelin o kadar iyi olduğunu gösterir.
4. Flask CORS (Cross-Origin Resource Sharing): Flask uygulamalarında aynı kaynaktan olmayan isteklere izin vermek için kullanılan bir Flask eklentisidir. Bu eklenti, tarayıcı güvenlik politikaları nedeniyle ortaya çıkabilecek CORS hatalarını yönetmeye yardımcı olur. CORS, tarayıcı tarafından uygulanan bir güvenlik önlemidir ve web uygulamalarının farklı kökenlerden (origin) gelen kaynaklara erişimini kısıtlar. Flask CORS, bu kısıtlamaları aşmak için gerekli başlık bilgilerini ekler ve böylece uygulamanın farklı kaynaklardan gelen isteklere yanıt vermesini sağlar. Bu, özellikle API'lerin başka domainlerden erişilmesi durumunda, uygulamanın düzgün çalışmasını sağlamak için önemlidir.
5. Selenium: Selenium, web tarayıcılarını otomatize etmek için kullanılan bir Python kütüphanesidir. Web scraper olarak da bilinen bu araç, web sayfalarını otomatik olarak gezerek veri toplamanıza olanak tanır. Selenium'un temel işlevi, kullanıcı etkileşimi simüle etmek ve dinamik web sayfalarından veri kazımak için web tarayıcılarını otomatik olarak kontrol etmektir. Bu kütüphane sayesinde, HTML içeriğini analiz edebilir, form doldurabilir, tıklama yapabilir ve hatta JavaScript tarafından oluşturulan içerikleri de alabilirsiniz. Bu sayede, web scraperlar geliştirmek ve çeşitli veri kaynaklarından bilgi toplamak oldukça kolaylaşır.
6. Joblib: Joblib, Python programlama dilinde verimli bir şekilde büyük veri setlerini ve hesaplamaları işlemek için kullanılan bir kütüphanedir. Özellikle bilimsel hesaplamalar ve makine öğrenimi alanında yaygın olarak tercih edilir. İşlevsel olarak, görevlerin paralelleştirilmesi, verilerin disk tabanlı belleklemesi ve hesaplamaların önbelleğe alınması gibi özellikler sunar. Bu sayede, büyük veri setleriyle çalışırken performans artışı ve işlem sürelerinde ciddi iyileşme sağlar. Joblib, basit API'si ile kullanım kolaylığı sunar ve diğer popüler Python kütüphaneleri ile sorunsuz bir şekilde entegre edilebilir.

7. MySQL Connector: Python programlama dili ile MySQL veritabanı arasında bağlantı kurmayı sağlayan resmi bir kütüphanedir. Bu kütüphane, MySQL veritabanına bağlanma, veri sorgulama, veri güncelleme ve yönetme gibi işlemleri kolaylaştırır. MySQL Connector/Python, kullanıcı dostu API'si ve kapsamlı özellikler ile güvenli ve verimli veri tabanı işlemleri gerçekleştirilmesine olanak tanır. Aynı zamanda, hem bağlantı havuzlama hem de SSL desteği gibi ileri düzey özellikler sunarak, uygulamaların güvenli ve ölçülebilir bir şekilde veritabanı ile etkileşimde bulunmasını sağlar.
8. SciPy: SciPy, bilimsel ve teknik hesaplama için kullanılan, açık kaynaklı bir Python kütüphanesidir. Sayısal hesaplama, lineer cebir, optimizasyon, integrasyon, interpolasyon, Fourier dönüşümleri, sinyal işleme, görüntü işleme, diferansiyel denklemler ve istatistik gibi birçok alanda geniş bir fonksiyon yelpazesine sahiptir. NumPy kütüphanesine dayanarak geliştirilen SciPy, bu algoritmaları etkili bir şekilde uygulamak için güçlü veri yapıları ve fonksiyonlar sunar. Bilimsel hesaplama, mühendislik, istatistik, biyoinformatik ve diğer disiplinlerde yaygın olarak kullanılmaktadır.
9. MySQL: MySQL kütüphanesi, MySQL veritabanıyla etkileşim kurmak için kullanılan bir yazılım bileşeni veya arayüzüdür. Bu kütüphane, programlarınızın MySQL veritabanına bağlanmasını, veri eklemesini, güncellemesini, silmesini veya sorgulamasını sağlar. Genellikle çeşitli programlama dillerinde (Python, PHP, Java vb.) kullanılabilen API'ler aracılığıyla erişilebilir ve geliştiricilere veritabanı işlemlerini programlamada büyük bir esneklik sunar. MySQL kütüphanesi, veritabanı işlemlerini programlama dili bağımsız olarak gerçekleştirmenizi sağlar, böylece farklı platformlarda ve sistemlerde uygulamanızın taşınabilirliğini sağlar.

KAYNAKLAR

- [1] <https://www.kaggle.com/datasets/shubhambathwal/flight-price-prediction>, Erişim, Eylül 2023
- [2] https://github.com/andrew-geeks/MakeMyTrip-scraper/tree/main/flight_datasets, Erişim, Eylül 2023
- [3] <https://github.com/andrew-geeks/MakeMyTrip-scraper>, Erişim, Eylül 2023
- [4] <https://www.kaggle.com/datasets/jillanisofttech/flight-price-prediction-dataset>, Erişim, Eylül 2023
- [5] <https://www.kaggle.com/code/burakergene/predict-fare-of-airlines-tickets-w-detailed-eda-rf>, Erişim, Eylül 2023
- [6] <https://www.youtube.com/watch?v=R75O5Q4&list=PLIHume2cwmHehcxE1XZieL21syR3m3tR>, Erişim, Kasım 2023
- [7] <https://docs.python.org/3/>, Erişim, Ocak 2024
- [8] <https://www.youtube.com/watch?v=uueWB3M54M4>, Erişim, Şubat 2024
- [9] <https://miuul.com/>, Erişim, Mart 2024
- [10] <https://www.youtube.com/watch?v=5aDfgcI8MHM>, Erişim, Nisan 2024
- [11] <https://developer.mozilla.org/en-US/docs/Web/JavaScript>, Erişim, Nisan 2024
- [12] <https://openai.com/chatgpt>, Erişim, Mart 2023
- [13] <https://scikit-learn.org/stable/>, Erişim, Nisan 2024
- [14] <https://reactjs.org/docs/getting-started.html>, Erişim, Mayıs 2024
- [15] <https://joblib.readthedocs.io/en/stable>, Erişim, Mayıs 2024
- [16] <https://medium.com/@muratgulcan/selenium-i%C87le-veri%C87-%C3%A7ekme-%C87%C5%9Flemi%C87-d468214234e4>, Erişim, Mayıs 2024