



Çanakkale Onsekiz Mart Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

BLM3012 - İLERİ VERİTABANI SİSTEMLERİ DERSİ
FİNAL PROJESİ RAPORU

Baha YOLAL – 200401031

İrfan KARA – 210401075

Ali İhsan REY – 210401115

Öğr. Gör. İSMAİL KAHRAMAN

Bu raporda, bir Volkswagen araba fiyat tahmini uygulaması geliştirilmiş olup, veriler ilişkisel veri tabanı kullanılarak depolanmıştır. Veri tabanı işlemlerini kolaylaştırmak için ORM yapısı oluşturulmuş ve XAF kullanılarak veriler web ortamında gözlemlenmiştir. Aşağıda verilen aşamalar tek tek izlenmiş ve kontroller sağlanmıştır. (Yapılan aşamalar en altta verilen youtube linkinde özetlenmiştir.)

1) Data Mining Project

Volkswagen Araba Fiyat Tahmini:

Açıklama:

- Araba markası, modeli, yaş, kilometre, motor özellikleri gibi özellikler kullanılarak bir regresyon modeli geliştirilir ve araç fiyatları tahmin edilir.

Dataset Shape:

- Veri kümesinde toplam 15157 örnek (instance) bulunmaktadır.
- Her bir örnek 9 özellik (feature) içermektedir.

Feature Türleri:

Marka (Categoric), Model (Categoric), Yaş (Numeric), Kilometre (Numeric)
Vites tipi (Categoric), Yakıt tipi (Categoric), Vergi (Numeric), Yakıt tüketimi (Numeric)
Motor özellikleri (Numeric)

Kullanılan Yöntem:

- Kullanılan yöntem, bir bağımlı değişkenin sürekli bir değerini tahmin etmek için kullanılan bir gözetimli öğrenme (supervised learning) tekniği olan XGBoost yöntemidir.
- Bir çok gözetimli öğrenme tekniği denenmiş olup arasından XGBoost tekniği en iyi sonucu vermiştir. **XGBoost (Extreme Gradient Boosting)**, gözetimli öğrenme algoritmalarından biridir ve özellikle yüksek doğruluk, hız ve verimlilik için optimize edilmiştir.

- 1) VwCar.csv dosyasındaki verilerin yapısına uygun olarak ORM (Object-Relational Mapping) yapısı oluşturulmuştur. Bu yapı, veri tabanı nesnelerini kod seviyesinde temsil eder ve veri tabanı işlemlerini kolaylaştırmak için yapılmıştır.

```
public virtual string model { get; set; } // 15157 non-null object
public virtual int? year { get; set; } // 15157 non-null int64
public virtual int? price { get; set; } //15157 non-null int64
public virtual string transmission { get; set; } //15157 non-null object
public virtual int? mileage { get; set; } //15157 non-null int64
public virtual string fuelType { get; set; } //15157 non-null object
public virtual int? tax { get; set; } //15157 non-null int64
public virtual double? mpg { get; set; } //15157 non-null float64
public virtual double? engineSize { get; set; } //15157 non-null float64
```

2) Local db ye erişip 2.adımda oluşan db ye erişim sağlandı.

SQLQuery1.sql - B... (BAHAY\baha_ (66))*				BahaY.CarPricePrediction - dbo.VwCars	
	Column Name	Data Type	Allow Nulls		
▶	ID	uniqueidentifier	<input type="checkbox"/>		
	model	nvarchar(MAX)	<input checked="" type="checkbox"/>		
	year	int	<input checked="" type="checkbox"/>		
	price	int	<input checked="" type="checkbox"/>		
	transmission	nvarchar(MAX)	<input checked="" type="checkbox"/>		
	mileage	int	<input checked="" type="checkbox"/>		
	fuelType	nvarchar(MAX)	<input checked="" type="checkbox"/>		
	tax	int	<input checked="" type="checkbox"/>		
	mpg	float	<input checked="" type="checkbox"/>		
	engineSize	float	<input checked="" type="checkbox"/>		
			<input type="checkbox"/>		

3) VwCar.csv dosyası, pandas DataFrame objesine aktarılmış ve ardından oluşan veriyi SQL ile veritabanına eklemek için INSERT komutu kullanılmıştır.

```
df = pd.read_csv(r"C:\Users\baha_\OneDrive\Belgeler\GitHub\VolkswagenCarPricePredictionMLApplication\vw.csv")

server = 'BAHAY'
database = 'CarPricePrediction'
username = 'sa'
password = 'B4b4yorgun.31'

connection = pyodbc.connect('DRIVER={SQL Server};SERVER='+server+';DATABASE='+database+';UID='+username+';PWD='+password)
cursor = connection.cursor()

for index, row in df.iterrows():
    id_value = uuid.uuid4() # Her satır için benzersiz bir değer sağlayın (örneğin, UUID kullanabilirsiniz)
    cursor.execute("INSERT INTO dbo.VwCars (ID, model, year, price, transmission, mileage, fuelType, tax, mpg, engineSize) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)",
        id_value, row['model'], row['year'], row['price'], row['transmission'], row['mileage'], row['fuelType'], row['tax'], row['mpg'], row['engineSize'])
    for ind, value in row.items():
        if pd.isnull(value):
            row[ind]=None

connection.commit()
connection.close()
```

4) XAF' i run edip db ye yazılanların db de olduğundan eminiz ve artık arayüzde verileri görüyoruz.

	ID	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
1	242C5066-0628-472F-B06A-000D4F9C8324	Golf	2018	16490	Manual	13741	Diesel	145	67.3	2
2	F6FCAE9-E4B0-4F60-A5F1-000EB9871A35	Touran	2018	18995	Semi-Auto	22034	Diesel	150	65.7	1.6
3	566A4E5-CFB5-449B-9F4C-00117AD4B2C5	Sharan	2017	17990	Manual	20856	Diesel	125	56.5	2
4	974ACEE7-CB49-4174-B140-0012F9230F55	Tiguan	2019	25000	Manual	5000	Diesel	145	47.9	2
5	3E7AC246-FB72-44C8-AD36-0013F5FEFF4	Golf SV	2017	12950	Manual	19231	Petrol	30	55.4	1.4
6	0A004C59-0993-4836-BABC-001490302719	Touran	2016	14750	Manual	33500	Diesel	30	64.2	2
7	7842439A-F884-48EF-9C47-001694E83E44	Tiguan	2019	27500	Semi-Auto	2153	Diesel	145	39.8	2
8	B72FE82A-4C91-4E23-AD32-001C1DFFC9B8	Tiguan	2017	17950	Semi-Auto	34059	Diesel	145	49.6	2
9	F51BFF6D-FC89-4985-8201-001E0B273780	Golf	2019	17390	Manual	5886	Diesel	145	57.7	1.6
10	B5ADDC98-7DA0-4323-8153-001EADC42DC2	Caddy Maxi Life	2019	19995	Semi-Auto	8481	Diesel	150	54.3	2
11	4E255071-63EE-484D-8A43-001F7062BA2B	Sharan	2019	22880	Automatic	13656	Diesel	145	41.5	2
12	EDDDC987-85D2-431D-B134-0025C89A6EAA	Golf	2018	20990	Semi-Auto	19700	Diesel	145	57.7	2
13	13729819-9ACC-4063-B1C5-0027EEF88DF0	Tiguan	2017	22000	Semi-Auto	21500	Petrol	235	38.2	2
14	B88A2A22-9E64-43EB-8BD4-002AA1CB8700	Golf	2019	31990	Automatic	2152	Petrol	145	36.2	2
15	40BFA961-142E-4C57-BACD-00327E37FB66	Touran	2016	16998	Semi-Auto	13465	Diesel	30	65.7	1.6
16	14BC9137-D379-48DE-86A2-003C5A683734	Beetle	2016	10941	Manual	21930	Petrol	125	51.4	1.2

Query executed successfully.

BAHAY (16.0 RTM)

BAHAY\baha_ (66)

CarPricePrediction

00:00:00

15.157 rows

6.) Test case yarattık ve bunu db'ye yazdık.

```
def predict_car_price(model, year, transmission, mileage, fueltype, tax, mpg, engineSize):  
    # Veri çerçevesi oluşturma  
    dataframe = pd.DataFrame({  
        'model': [model],  
        'year': [year],  
        'transmission': [transmission],  
        'mileage': [mileage],  
        'fueltype': [fueltype],  
        'tax': [tax],  
        'mpg': [mpg],  
        'engineSize': [engineSize]  
    })  
  
    # Tahmin yapma  
    prediction = model_xgb.predict(dataframe)  
  
    return prediction[0] # Tahmini döndürme  
  
# Rastgele değerler oluşturma  
model = np.random.randint(0, 27) # 1 ile 20 arasında rastgele bir model seçme  
year = np.random.randint(2000, 2023) # 2000 ile 2022 arasında rastgele bir yıl seçme  
transmission = np.random.randint(0, 3) # 1 ile 2 arasında rastgele bir şanzıman seçme  
mileage = np.random.randint(1000, 100000) # 1000 ile 100000 arasında rastgele bir kilometre seçme  
fueltype = np.random.randint(0, 4) # 1 ile 3 arasında rastgele bir yakıt türü seçme  
tax = np.random.randint(0, 100) # 0 ile 100 arasında rastgele bir vergi seçme  
mpg = np.random.uniform(10, 100) # 10 ile 100 arasında rastgele bir yakıt tüketimi seçme  
engineSize = np.random.uniform(1, 5) # 1 ile 5 arasında rastgele bir motor hacmi seçme  
  
# Tahmin işlemi  
predicted_price = predict_car_price(model, year, transmission, mileage, fueltype, tax, mpg, engineSize)  
print("model:", listCarModel[model], "year:", year, "transmission:", listTransmission[transmission], "mileage:", mileage, "fueltype:", listfuelType[fueltype], "tax:", tax, "mpg:", mpg,  
      "Predicted car price:", predicted_price)
```

7.) XAF web projesi çalıştırılarak, oluşturulan verilerin ve test senaryolarının doğru bir şekilde işlendiği ve arayüzde görüntülediği kontrol edilmiştir.

eXpressApp Framework

Vw Cars

Filter...

Default

Vw Cars

+ New

Delete

Text to search...

Q

+

-

<input type="checkbox"/>	model	year	price	transmission	mileage	fuel Type	tax	mpg	engine Size	
<input type="checkbox"/>	Golf	2018	16,490	Manual		13,741	Diesel	145	67.3	2
<input type="checkbox"/>	Touran	2018	18,995	Semi-Auto		22,034	Diesel	150	65.7	1.6
<input type="checkbox"/>	Sharan	2017	17,990	Manual		20,856	Diesel	125	56.5	2
<input type="checkbox"/>	Tiguan	2019	25,000	Manual		5,000	Diesel	145	47.9	2
<input type="checkbox"/>	Golf SV	2017	12,950	Manual		19,231	Petrol	30	55.4	1.4
<input type="checkbox"/>	Touran	2016	14,750	Manual		33,500	Diesel	30	64.2	2
<input type="checkbox"/>	Tiguan	2019	27,500	Semi-Auto		2,153	Diesel	145	39.8	2
<input type="checkbox"/>	Tiguan	2017	17,950	Semi-Auto		34,059	Diesel	145	49.6	2
<input type="checkbox"/>	Golf	2019	17,390	Manual		5,886	Diesel	145	57.7	1.6
<input type="checkbox"/>	Caddy Maxi Life	2019	19,995	Semi-Auto		8,481	Diesel	150	54.3	2
<input type="checkbox"/>	Sharan	2019	22,880	Automatic		13,656	Diesel	145	41.5	2
<input type="checkbox"/>	Golf	2018	20,990	Semi-Auto		19,700	Diesel	145	57.7	2
<input type="checkbox"/>	Tiguan	2017	22,000	Semi-Auto		21,500	Petrol	235	38.2	2
<input type="checkbox"/>	Golf	2019	31,990	Automatic		2,152	Petrol	145	36.2	2
<input type="checkbox"/>	Touran	2016	16,998	Semi-Auto		13,465	Diesel	30	65.7	1.6
<input type="checkbox"/>	Beetle	2016	10,941	Manual		21,930	Petrol	125	51.4	1.2
<input type="checkbox"/>	Golf	2016	16,800	Manual		24,712	Diesel	20	67.3	2
<input type="checkbox"/>	Sharan	2015	13,689	Manual		33,052	Diesel	165	50.4	2
<input type="checkbox"/>	Golf	2015	16,998	Manual		41,427	Petrol	200	39.8	2
<input type="checkbox"/>	Passat	2017	14,999	Manual		21,421	Diesel	145	68.9	2

<<

<

1

>

>>

of 758

Page Size: 20

Youtube Sunum Link:

<https://youtu.be/ABoHSXg50P0>

GitHub Proje Link:

<https://github.com/Bahayll/VolkswagenCarPricePredictionMLApplication>