

**İkinci El  
Otomobil Fiyat  
Tahmini için**

**MAKİNE ÖĞRENMESİ  
MODELİ**

---

# Giriş:

Projemizin amacı, belirli bir otomobilin marka, model, yıl, kilometre, motor hacmi gibi özelliklerine dayalı olarak piyasa değerini tahmin edebilecek bir makine öğrenmesi modeli geliştirmektir.

- 1 **Veri Kazıma**
  - 2 **Verileri işleme**
  - 3 **Modelin Kurulması ve Değerlendirilmesi**
-

# Veri Kazıma



Autoscout24.com Veri Kazıma Seçimimizin Nedenleri:

- Spam Engeli Yok: Veri erişimine izin veriyor, bu nedenle veri toplama süreci sorunsuz.
- Basit ArayüzYapısı: Doğrudan ulaşılabilen bilgilere sahip basit bir yapı, veri çekmeyi kolaylaştırıyor.
- Geniş İlan Havuzu: Avrupa pazarında yeterli veri sunarak anlamlı ve güvenilir sonuçlar elde etmemizi sağlıyor.

**Auto  
Scout24**

# Veri Kazıma



UiPath'in veri kazıma için sağladığı avantajlar:

- Görsel Otomasyon: Web sitelerinin kullanıcı arayüzünü taklit ederek veri çekmeyi kolaylaştırır.
- Esneklik: Farklı kaynaklardan veri toplamak için özelleştirilebilir iş akışları sunar.
- Hız ve Verimlilik: Otomasyon ile manuel veri toplama işlemlerine göre daha hızlı sonuçlar elde edilir.
- Doğruluk: İyi yapılandırılmış robotlar insan hatası olmadan veri toplar.
- Entegrasyon: Diğer araçlarla kolayca entegre olabilir, ek işlemlere gerek kalmaz.
- Yenilik: Yeni bir şeyler öğrenmek istememiz



2024 Fiat Egea Fiyatları & Modelleri x

https://www.sahibinden.com/flat-egca?a5\_min=2024&pagingSize=50&a5\_max=2024

OBS 30.2.3 - Profil: İsimsiz - Sahneler: İsimsiz

Dosya Düzenle Görünüm Paneller Profil Sahne Koleksiyonu Araçlar Yardım

Kaynak seçilmemiş Özellikler Filtreler

Sahneler

Sahne

Kaynaklar

Ekran

Ses Karıştırıcı

Masaüstü Ses 0.0 dB

Mic/Aux -inf dB

Sahne Geçişleri

Soldur

Süre 300 ms

Kontroller

Yayını Başlat

Kaydı Durdur

Sanal Kamerayı Başlat

Stüdyo Modu

Ayarlar

Çıkış

CPU: 0.4% 60.00 / 60.00 FPS

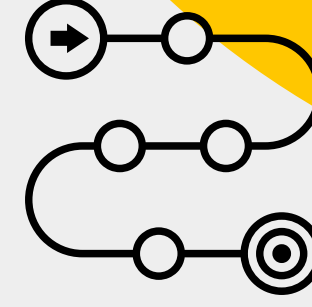
1.6 Multijet Limited FIAT ANA BAYI EFSANE EGEA 1.6 DTC LIMITED HATASIZ BOYASIZ

2024 8.000 Siyah 1.349.000 TL 27 Kasım 2024 Kocaeli Gebze

Ara

Haberler 19:11 27.11.2024

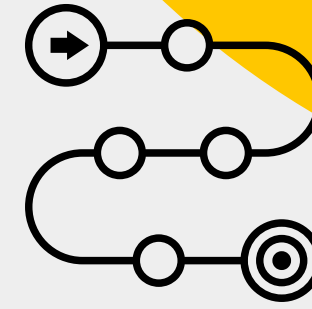
# Verileri İşleme



- Gereksiz Verilerden Kurtulma:
  - İşimize yaramayan sütunları ve satırları çıkardık.
  - Tekrar eden verileri temizledik.
- Sapan Veriler ve Normalizasyon:
  - Sapan değerleri bulup ya kaldırdık ya da düzelttik.
  - Verileri aynı seviyeye getirmek için normalize ettik.
- Boş Değerlerin Doldurulması:
  - Eksik olan yerleri tespit ettik.
  - Bu boşlukları ortalama, medyan veya mod ile doldurduk.
- Aykırı Değerleri Kaldırma:
  - Aykırı değerleri bulup ya kaldırdık ya da dönüştürdük.
- Yeni Dosyaya Aktarma:
  - Temizlediğimiz veriyi yeni bir dosyaya kaydettik.



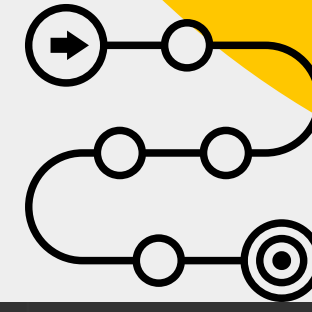
# Verileri İşleme



```
1 import pandas as pd
2 from sklearn.preprocessing import StandardScaler
3
4 def veri_yukle(dosya):
5     data = pd.read_excel("cleaned_data.xlsx")
6     print("Veriler yüklendi")
7     print(data.head())
8     return data
9
10 def veri_incele(data):
11
12
13
14
15
16 def kayip_degerlerle_ilgilen(data):
17
18
19
20
21
22
23 def aykiri_degerleri_kaldir(data):
24
25
26
27
28
29
30
31
32 def ozellikleri_normalize_et(data, features):
33
34
35
36
37
38
39 if __name__ == "__main__":
40     dosya = 'cleaned_data.xlsx'
41     data = veri_yukle(dosya)
42     veri_incele(data)
43     data = kayip_degerlerle_ilgilen(data)
44     data = aykiri_degerleri_kaldir(data)
45     data = ozellikleri_normalize_et(data, ['Km', 'hp', 'year'])
46
47     output_file = 'processed_data.xlsx'
48     data.to_excel(output_file, index=False)
49     print(f"\nVerinin hazirlanmasi tamamlandi. Hazır veri bu sekilde kaydedildi '{output_file}'")
```



# Verileri İşleme



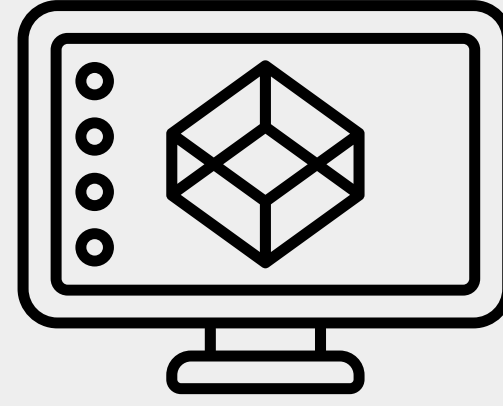
```
1 import pandas as pd
2 from sklearn.model_selection import train_test_split
3
4 # verileri on isleme tabi tutan fonksiyon
5 def veri_onislem(dosya):
6     # islenmis veriyi yukle
7     data = pd.read_excel("processed_data.xlsx")
8
9     # ihtiyacimiz olan ozellikleri ve hedefi belirle
10    X = data[['Km', 'hp', 'year']] # input
11    y = data['price'] # hedef
12
13    # datayi train ve test icin bolumle (80% train, 20% test)
14    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
15
16    print("Veri on isleme tamamlandi")
17    print(f"train verisi icin shape: {X_train.shape}")
18    print(f"testing verisi icin shape: {X_test.shape}")
19
20    # bolumlenmis veriyi model trainingi icin returnle
21    return X_train, X_test, y_train, y_test
22
23 # dosya direkt execute edilirse on islemeyi yap
24 if __name__ == "__main__":
25     X_train, X_test, y_train, y_test = veri_onislem('processed_data.xlsx')
26
```

```
Veri on isleme tamamlandi
train verisi icin shape: (1273, 3)
testing verisi icin shape: (319, 3)
```

Bu Python kodu, bir veri dosyasını işleyerek makine öğrenmesi modeli için eğitim ve test verilerini hazırlıyor



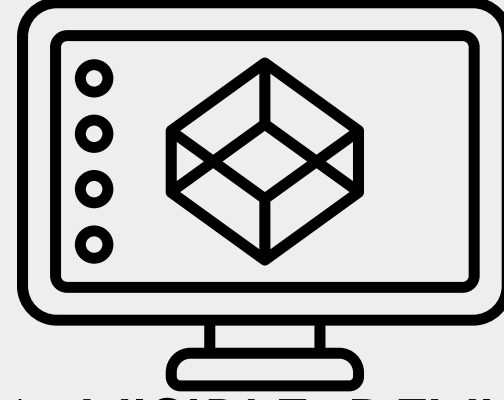
# Modelin Kurulması ve Değerlendirilmesi



## Kullanılan Kütüphaneler ve Özellikleri

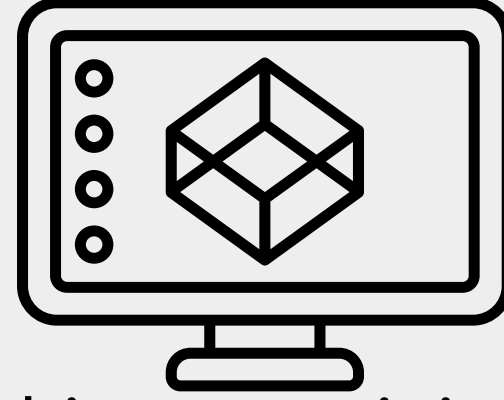
- Pandas
- Scikit-learn
- NumPy
- TensorFlow ve Keras
  - Sequential: Katmanlı bir sinir ağı modeli oluşturma.
  - Dense: Tam bağlantılı (fully connected) katmanların eklenmesi.
  - Adam: Optimize edici algoritma.
  - `model.compile()`: Modelin öğrenme parametrelerini ayarlama.

# Modelin Kurulması ve Değerlendirilmesi



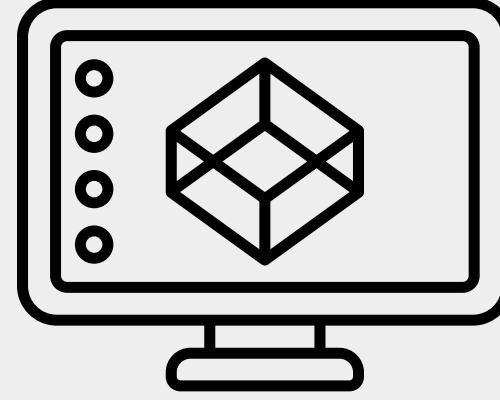
- CUDA Ayarları: İlk olarak, `os.environ["CUDA_VISIBLE_DEVICES"] = "-1"` satırı ile GPU kullanımını devre dışı bıraktım. Bu, modelin sadece CPU üzerinde çalışmasını sağlıyor.
- Kütüphanelerin İçerik Aktarılması: `pandas`, `tensorflow`, `sklearn`, `numpy`, ve `matplotlib` gibi kütüphaneleri içere aktardım. Bunlar veri işleme, model oluşturma ve sonuçları görselleştirme için gerekli.
- Veri Yükleme ve Hazırlama: `processed_data.xlsx` dosyasından verileri yükledim. Özellikler olarak `Km`, `hp`, ve `year` sütunlarını, hedef olarak ise `price` sütununu seçtim. Veriyi %80 eğitim ve %20 test olarak böldüm.
- Veri Normalizasyonu: `StandardScaler` kullanarak verileri normalize ettim. Bu, modelin daha hızlı ve etkili öğrenmesini sağlıyor.

# Modelin Kurulması ve Değerlendirilmesi



- Modelin İnşası: `model_insasi` fonksiyonu ile bir yapay sinir ağı modeli oluşturdum.
- Model, üç gizli katmana sahip: İlk katman: 64 nöron İkinci katman: 32 nöron Üçüncü katman: 16 nöron Çıkış katmanı ise tek bir nöron içeriyor, çünkü fiyat tahmini yapıyoruz.
- Modelin Eğitimi: Modeli 100 epoch boyunca eğittim. Eğitim ve doğrulama kayıplarını matplotlib ile görselleştirdim.
- Modelin Değerlendirilmesi: Test verisi üzerinde modeli değerlendirdim ve kayıp değerini yazdırdım. `r2_score` fonksiyonu ile modelin doğruluk skorunu hesapladım.
- Tahminler ve Kullanıcı Deneyimi: Test verisi üzerinde bazı tahminler yaptım ve bunları gerçek değerlerle karşılaştırdım.

# Modelin Kurulması ve Değerlendirilmesi



**10/10** ————— **0s** 3ms/step - loss: 4.2730 - mae: 1.5813

Test kaybı: [4.776244640350342, 1.6725867986679077]

**10/10** ————— **0s** 6ms/step

Modelin doğruluk skoru ( $R^2$ ): 90.14%

Tahmini fiyat: 12.730452537536621, Asil fiyat: 9.7

Tahmini fiyat: 18.15683364868164, Asil fiyat: 14.979

Tahmini fiyat: 25.158283233642578, Asil fiyat: 29.94

Tahmini fiyat: 13.213232040405273, Asil fiyat: 12.9

Tahmini fiyat: 27.215438842773438, Asil fiyat: 22.444

Arabanin ozelliklerini girin:

Arabanin yili: 2022

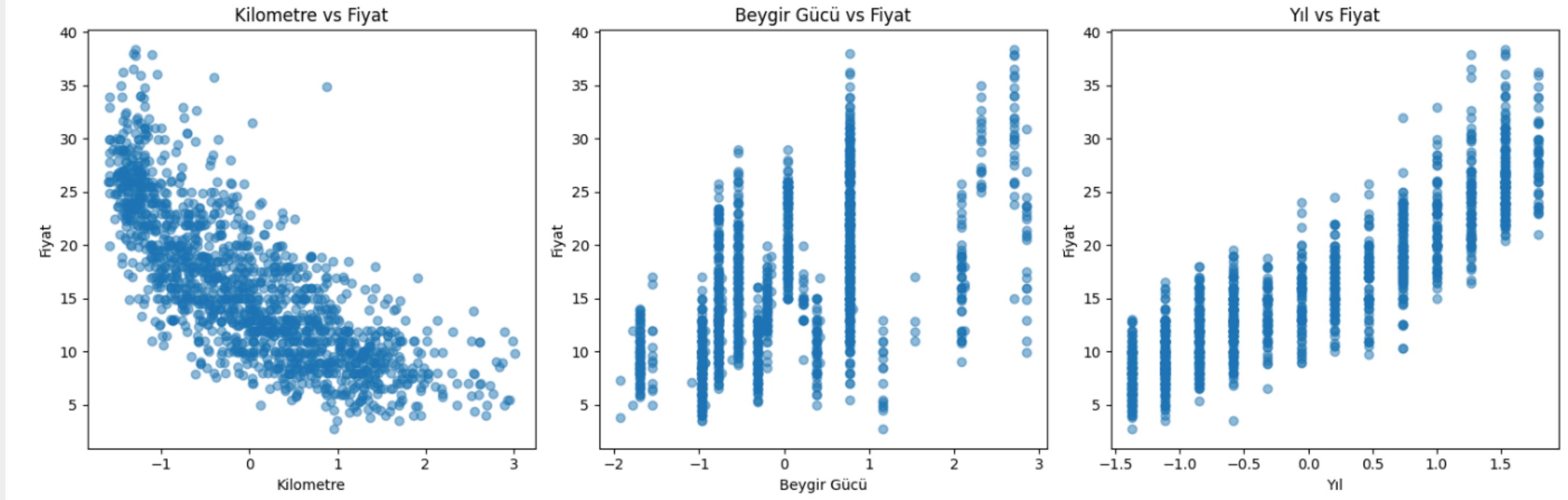
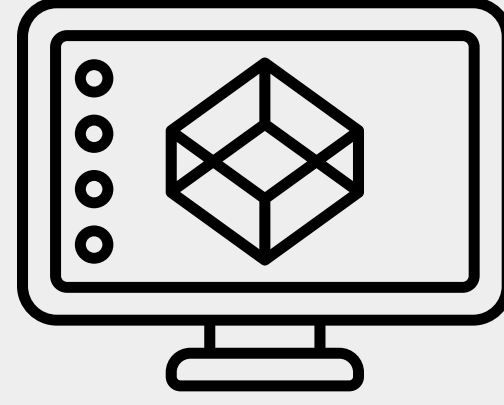
Arabanin km'sini yazin: 50000

Arabanin beygir gucunu yazin: 120

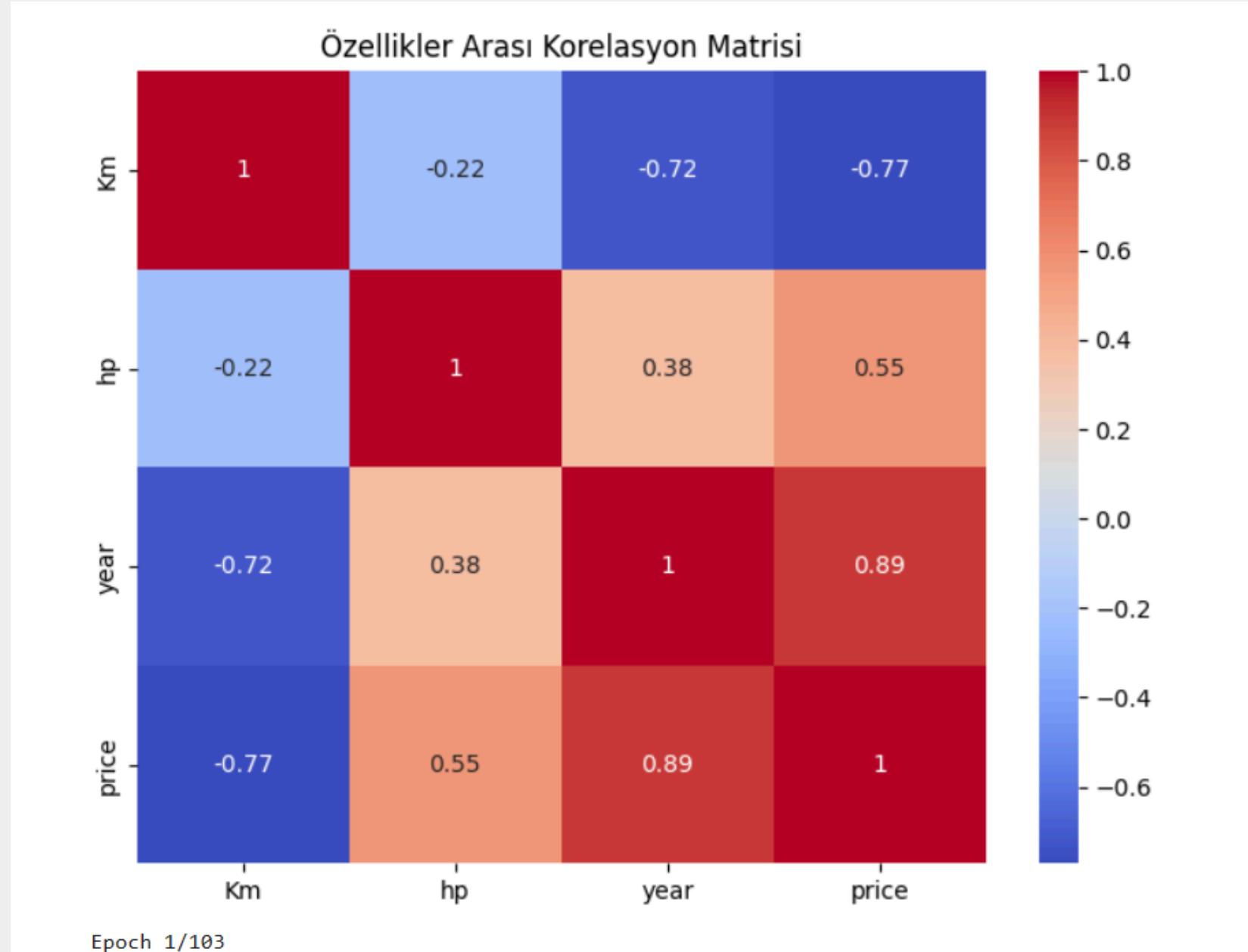
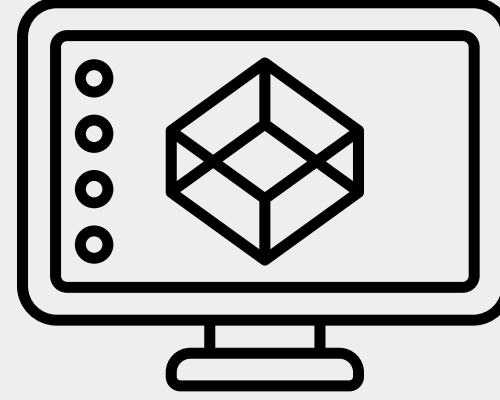
**1/1** ————— **0s** 30ms/step

Tahmini fiyat: \$2194.04

# Modelin Kurulması ve Değerlendirilmesi



# Modelin Kurulması ve Değerlendirilmesi

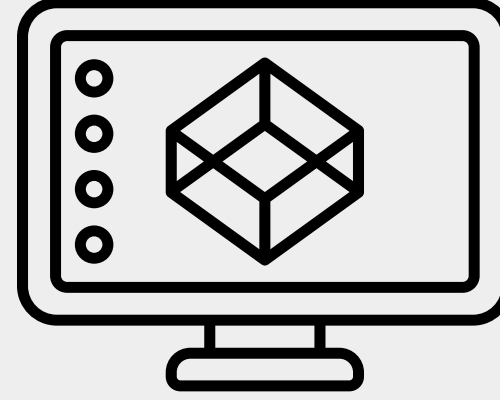


- km ve price: -0.77 değeri, kilometre artarken fiyatın genellikle düştüğünü gösterir. Bu, beklenen bir durumdur çünkü yüksek kilometreli araçlar genellikle daha düşük fiyatlıdır.
- year ve price: 0.89 değeri, üretim yılı arttıkça fiyatın da genellikle arttığını gösterir. Bu da beklenen bir durumdur, çünkü yeni model araçlar genellikle daha pahalıdır.



# Ekstra

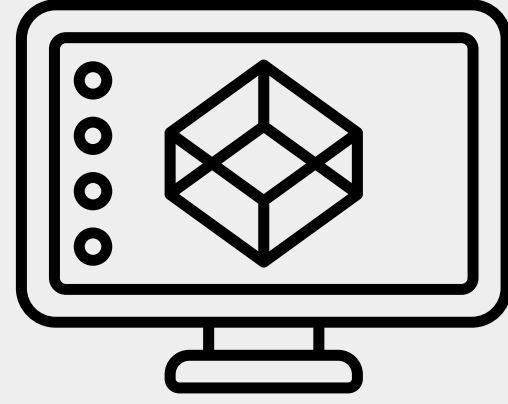
## Model 2



- Kütüphanelerin İçerik Aktarılması: pandas, numpy, matplotlib, seaborn, ve sklearn gibi kütüphaneleri içerik aktardım. Bunlar veri işleme, model oluşturma ve sonuçları görselleştirme için gerekli.
- Veri Yükleme ve Ön İşleme: cleaned\_data.xlsx dosyasından verileri yükledim.
- preprocess\_data fonksiyonu ile veriyi temizledim ve dönüştürdüm: Vites ve yakıt tiplerini standartlaştırdım. Kategorik değişkenleri LabelEncoder ile sayısal hale getirdim. Arabanın yaşını hesapladım ve kilometreyi ölçek küçültmek için 1000'e böldüm.
- Özellik ve Hedef Değişkenlerin Belirlenmesi: Modelin girdi olarak kullanacağı özellikleri (Km\_thousands, gear\_encoded, year, fuel\_type\_encoded, hp) ve hedef değişkeni (price) belirledim.

# Ekstra

## Model 2

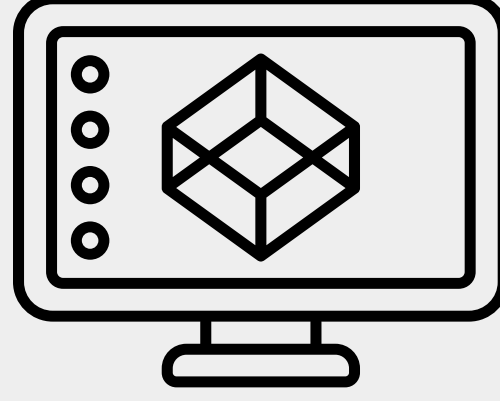


- Veri Ölçeklendirme: StandardScaler kullanarak verileri ölçeklendirdim.
- Modelin Oluşturulması ve Eğitimi: RandomForestRegressor modelini oluşturdum ve eğittim. Bu model, karar ağaçları kullanarak tahmin yapar ve genellikle iyi bir performans gösterir.
- Tahmin Fonksiyonu: predict\_price fonksiyonu, kullanıcıdan alınan verilerle fiyat tahmini yapıyor. Girdi verilerini hazırlayıp ölçeklendirdikten sonra modelden tahmin alıyor.



# Ekstra

## Model 2



```
=== Volkswagen Golf Fiyat Tahmini ===  
Kilometreyi girin: 50000  
Yılı girin (örn. 2013): 2022  
Beygir gücünü girin: 120
```

```
Vites tipi için:  
1 - Manuel  
2 - Otomatik  
Vites tipini seçin (1 veya 2): 1
```

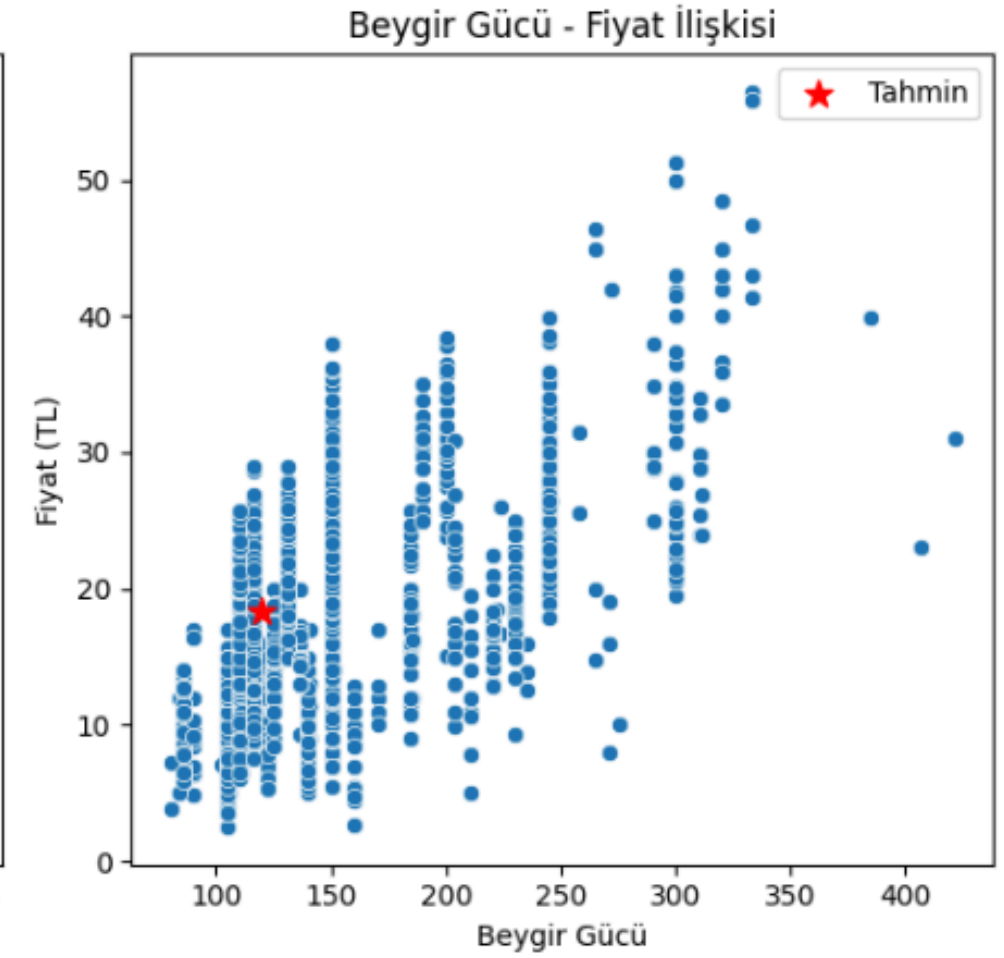
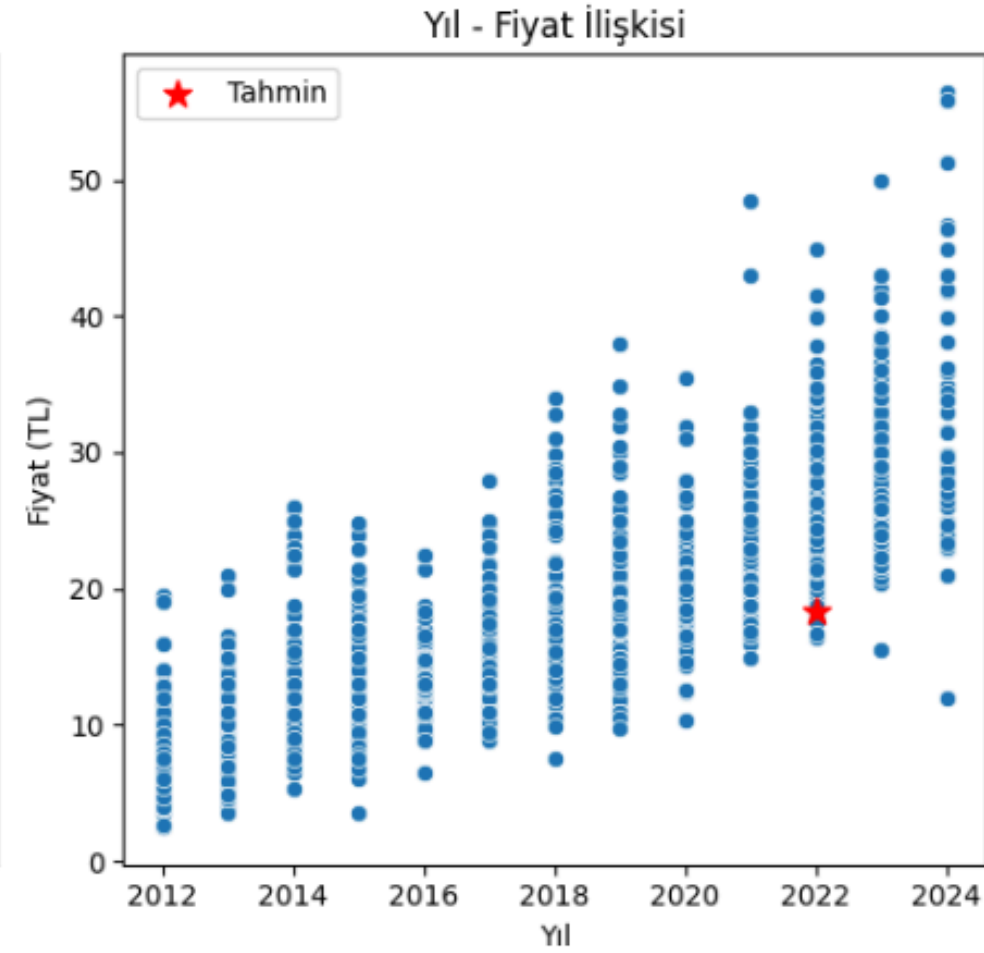
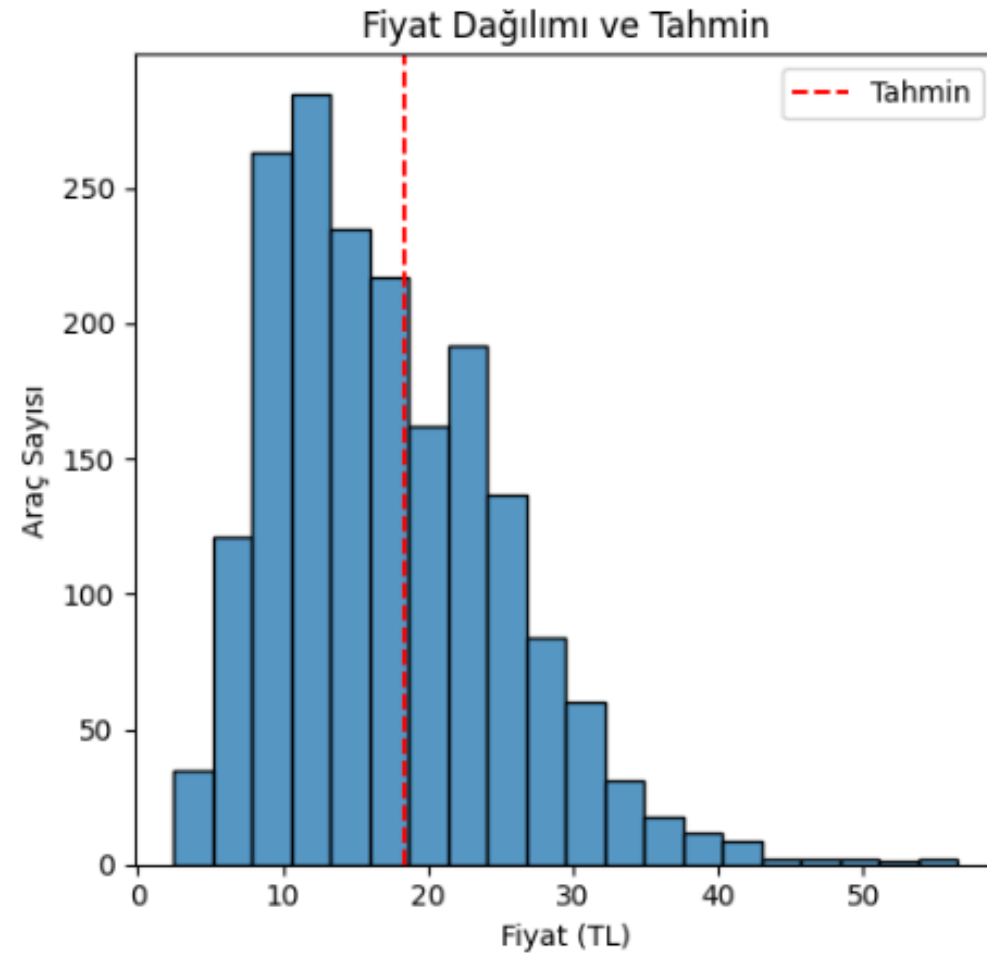
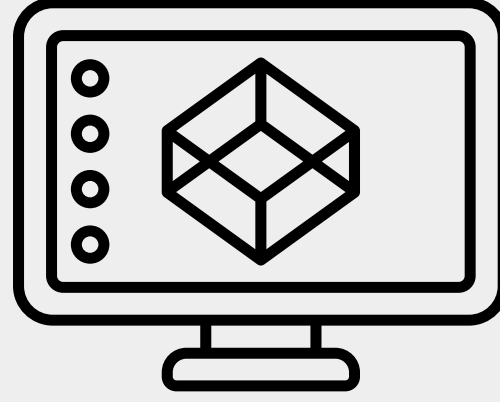
```
Yakıt tipi için:  
1 - Benzin  
2 - Dizel  
Yakıt tipini seçin (1 veya 2): 1
```

```
C:\Users\GOK\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning:  
e names  
warnings.warn(
```

```
=== Tahmin Sonuçları ===  
Girilen Değerler:  
Kilometre: 50,000  
Yıl: 2022  
Beygir Gücü: 120.0  
Vites: Manual  
Yakıt: Gasoline
```

```
Tahmini Fiyat: 18.36 TL
```

# Ekstra Model 2



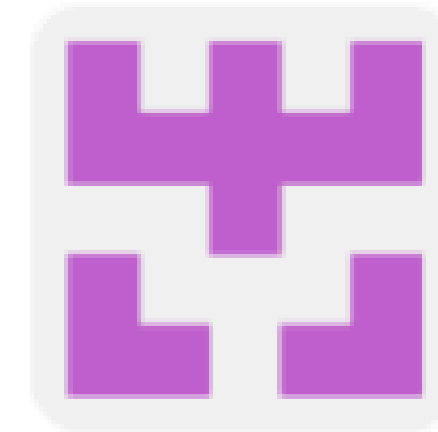
Model Performans Metrikleri:  
R2 Score: 0.9343  
RMSE: 2.16 TL



**teşekkürler.**

---

# safagoek/ ML\_Car\_price\_prediction



1

Contributor



0

Issues



0

Stars



0

Forks



## **safagoek/ML\_Car\_price\_prediction**

Contribute to safagoek/ML\_Car\_price\_prediction development by creating an account on GitHub.



GitHub