

MACHINE LEARNING (PROJECTS)

Project ① (House Price Prediction)

Adımlar

- ① Veri setini segme
- ② Veriyi keşfetme
- ③ Veri ön işleme
- ④ Model seisme eğitme
- ⑤ Modeli değerlendirme
- ⑥ Modelin tahmin yapmasını sağlama

Dosya yapısı

↳
VS Code

+ saved-model.py

| HOUSE_PRICE_PREDICTION | |
|------------------------|---|
| data | |
| .gitkeep.txt | A |
| test.csv | U |
| train.csv | U |
| models | |
| .gitkeep.txt | A |
| house_price_model.pkl | U |
| notebooks | |
| .gitkeep.txt | A |
| 1_data_explorations.py | U |
| 2_model_training.py | U |
| src | |
| __pycache__ | |
| .gitkeep.txt | A |
| load_data.py | U |
| predict.py | U |
| preprocess.py | U |
| train.py | U |
| .gitignore.txt | A |
| README.md.txt | A |
| requirements.txt.txt | A |

① Veri Setini Seçme

Veri setini Kaggle'dan alabiliriz veya Scikit-learn'ın (California Housing) veri setini kullanabiliriz.

② load_data.py dosyası ile veri setini projeye import ettim.

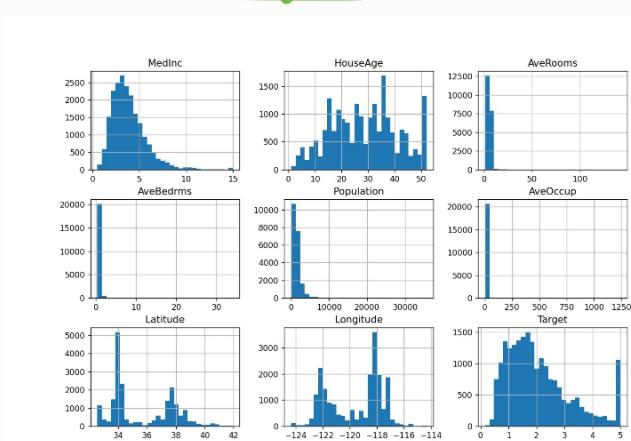
③ 1-data-explorations dosyası ile veri keşfi yaptım.

- eksik veri kontrolü
- veri dağılımı
- Korelasyon matrisi

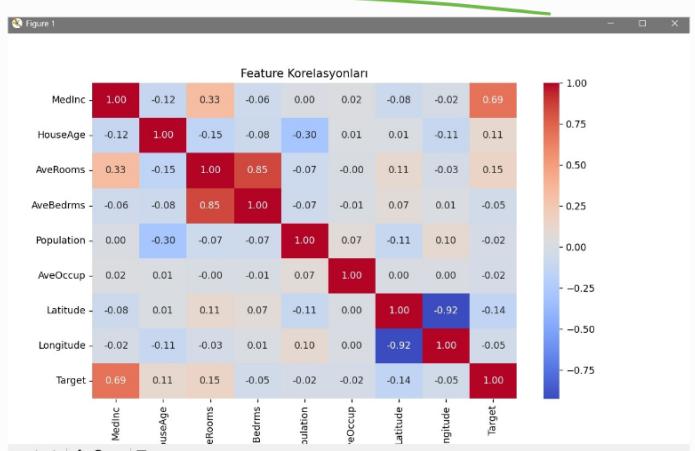
yüklediğim veri hakkında genel bilgi edinimi

Şangi değişkenlerin hedef değişken ile ilişkili olduğunu anlamamızı sağlar.

Veri Dağılım.



Korelasyon Matrisi



① Sıradaki adım veri önisleme

↳ Normalizasyon: "özellikleri aynı ölçekte getirme"

↳ Veri Bölme: eğitim + test verisi olarak ayırma

"train-test-split" { %80 eğitim
%20 test ile

→ "StandardScaler" ile veriyi standart hale getiriyoruz.

```
PS C:\Users\Yvzsa\OneDrive\Desktop\MACHINE_LEARNING\House_Price_Prediction & C:/Users/Yvzsa/AppData/Local/Programs/Python/Python311/python.exe c:/Users/Yvzsa/OneDrive/Desktop/MACHINE_LEARNING/House_Price_Prediction/src/preprocess.py
Eğitim verisi boyutu: (16512, 8)
Test verisi boyutu: (4128, 8)
```

↳ veri, model eğitimi için hazır!

② Model Kurma ve Eğitme

train.py dosyasında ayırdığımız eğitim verisi ile model eğitiyorum.

LinearRegression() ile modeli kurup

fit() ile eğitiyorum

↳ modeli kurdugumuz algoritma

mean_squared_error() fonk. ile modelin tahmin performansını değerlendirmiyorum.

RMSE (root mean squared error) değeri ile modelin doğruluğunu ölçüyorum.

```
PS C:\Users\Yvzsa\OneDrive\Desktop\MACHINE_LEARNING\House_Price_Prediction> & C:/Users/Yvzsa/AppData/Local/Programs/Python/Python311/python.exe c:/Users/Yvzsa/OneDrive/Desktop/MACHINE_LEARNING/House_Price_Prediction/src/train.py
Model Değerlendirmesi:
RMSE: 0.7455813830127764
```

↳ modelin tahmin hatası :

↳ ne kadar küçükse o kadar iyi

④ Model Kaydetme

saved-model.py dosyasında modeli kaydedelim. → house_price_model.pkl dosyasına

İlerleyen aşamalarda tekrar kullanılabilir.

④ Son Aşama

Modelin yüklenmesi ve Tahmin

`joblib.load()` ile kaydedilen modeli `house_price-model.pkl` dosyasına kaydettim.

→ `predict` metodu ile yüklenen model ile birlikte tahmin yapılır.

```
# Örnek veriyle tahmin yapma
new_data = np.array([[5.0, 20.0, 6.0, 2.0, 2000.0, 3.0, 37.0, -122.0]]) # Yeni örnek veri
prediction = model.predict(new_data)
print(f"Tahmin Edilen Konut Fiyatı: {prediction[0]}")
```

Örnek bir evin sahip olabileceği özellikleri alarak model üzerinde evin sahip olabileceği fiyat üzerinde tahmin yapar.

```
PS C:\Users\Yvzsa\OneDrive\Desktop\MACHINE_LEARNING\House_Price_Prediction & C:/Users/Yvzsa/AppData/Local/Programs/Python/Python311/python.exe c:/Users/Yvzsa/OneDrive/Desktop/MACHINE_LEARNING/House_Price_Prediction/src/predict.py
Tahmin Edilen Konut Fiyatı: 75.90323957087394
```

SONUÇ! Özellikleri ve bu özelliklere bağlı fiyatları bulunan 9 örnek evi veri setini eğiterek özellikleri girilen bir evin fiyat tahminini yapmış oldum !!

