import pandas as pd

from sklearn.datasets import fetch\_openml

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import StandardScaler

boston = fetch\_openml(data\_id=531)

df = pd.DataFrame(data=boston.data, columns=boston.feature\_names)

df['PRICE'] = boston.target # Ev fiyatları (TARGET)

print("Eksik değerler:")

print(df.isnull().sum())

X = df.drop("PRICE", axis=1)

y = df["PRICE"]

imputer = SimpleImputer(strategy='mean')

X = imputer.fit\_transform(X)

scaler = StandardScaler()

X = scaler.fit\_transform(X)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print(f"Mean Squared Error: {mse}")

print(f"R-squared: {r2}")

print("Sonuçların Yorumlanması:")

print(f"Modelin Mean Squared Error (MSE) değeri: {mse}, bu değer modelin tahminlerinde ortalama karesel hata miktarını gösterir. MSE'nin düşük olması, modelin daha iyi performans gösterdiği anlamına gelir.")

print(f"Modelin R-squared (R²) değeri: {r2}, bu değer bağımsız değişkenlerin bağımlı değişkeni ne kadar açıkladığını gösterir. R² değeri 1'e ne kadar yakınsa, model o kadar iyi demektir.")