Tomasz MARKIEWICZ Nr albumu: 136917

Lab 3 - **Sprawozdanie z analizy regresji liniowej na zestawie danych California Housing**

1. Wstęp

Celem tego sprawozdania było przeprowadzenie analizy regresji liniowej na zestawie danych dotyczącym nieruchomości w Kalifornii. Zadaniem było zbadanie zależności pomiędzy poszczególnymi cechami (kolumnami danych) a wartością docelową (cena nieruchomości). Do tego celu wykorzystano bibliotekę *scikit-learn,* z pakietu *sklearn.datasets* wczytana została funkcja *fetch_california_housing*. Zbiór zawiera informacje dotyczące różnych cech nieruchomości, na Rysunku 1. przedstawiony został podział danych na zbiór uczący (*70% danych*) i zbiór testowy (*30% danych*) za pomocą funkcji *train_test_split* (Kod 1.)

```
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error
from tabulate import tabulate

data = fetch_california_housing()
X = data.data
y = data.target
feature_names = data.feature_names

'''Podział danych na zbiór uczący i testowy'''
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)
print(f"Wartośc zbioru testującego: ", len(X_test), "obietków", "\nWartość
zbiór uczącego: ", len(X_train), "obiektów")
```

Kod 1. Fragment kodu przy zachowaniu stałości danych wejściowych.

```
Wartośc zbioru testującego:  6192 obietków
Wartość zbiór uczącego:  14448 obiektów
```

Rysunek 1. Podział próbek.

2. Analiza regresji liniowej

Przy pomocy biblioteki *matplotlib* wygenerowany został wykres i podzielony przy pomocy *subplot* (Kod 2.), na którym przedstawiono zależność między daną cechą a wartością „*Przewidzianą*” (cena nieruchomości). Uwzględniłem również dodatkowo dla każdego modelu wartości błędów oraz dodałem także krzywą regresji liniowej.

Poniżej przedstawiam dalszą część kodu oraz wykres z regresją liniową dla wybranych cechy. Na wykresie przedstawiono rzeczywiste wartości (punkty niebieskie) oraz krzywą regresji liniowej (linia czerwona) (Rysunek 2).

```
mae_values = []
mse_values = []

plt.figure(figsize=(12, 9))

for i in range(X.shape[1]):
    plt.subplot(3, 3, i + 1)
    plt.grid(color='grey', linestyle="--", linewidth=1.15, alpha=0.5)

    '''Wybór jednej zmiennej (kolumny)'''
    X_train_single = X_train[:, i].reshape(-1, 1)
    X_test_single = X_test[:, i].reshape(-1, 1)

    '''Uczenie modelu regresji liniowej'''
    model = LinearRegression()
    model.fit(X_train_single, y_train)

    '''Predykcja na zbiorze testowym'''
    y_pred = model.predict(X_test_single)

    '''Wykres'''
    plt.scatter(X_train_single, y_train, alpha=0.7, color='blue', label='zbiór
uczący')
    plt.scatter(X_test_single, y_test, alpha=0.2, color='green', label='zbiór
testujący')
    plt.plot(X_test_single, y_pred, color='red', label='linia regresji')

    plt.xlabel(f'{feature_names[i]}')
    plt.ylabel('Przewidywane')
    plt.title(f'Linia regresji ceny - {feature_names[i]} vs Przewidywane')
    plt.legend()

    '''Obliczenie i wypisanie błędów MAE i MSE'''
    mae = mean_absolute_error(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)

    mae_values.append(mae)
    mse_values.append(mse)
    mae_values = [round(val, 4) for val in mae_values]
    mse_values = [round(val, 4) for val in mse_values]
    # print(f"{feature_names[i]} - MAE: {mae:.4f}, MSE: {mse:.4f}")

plt.tight_layout()
plt.show()

'''Nauczenie modelu regresji liniowej na całym zbiorze uczącym'''
final_model = LinearRegression()
final_model.fit(X_train, y_train)
```

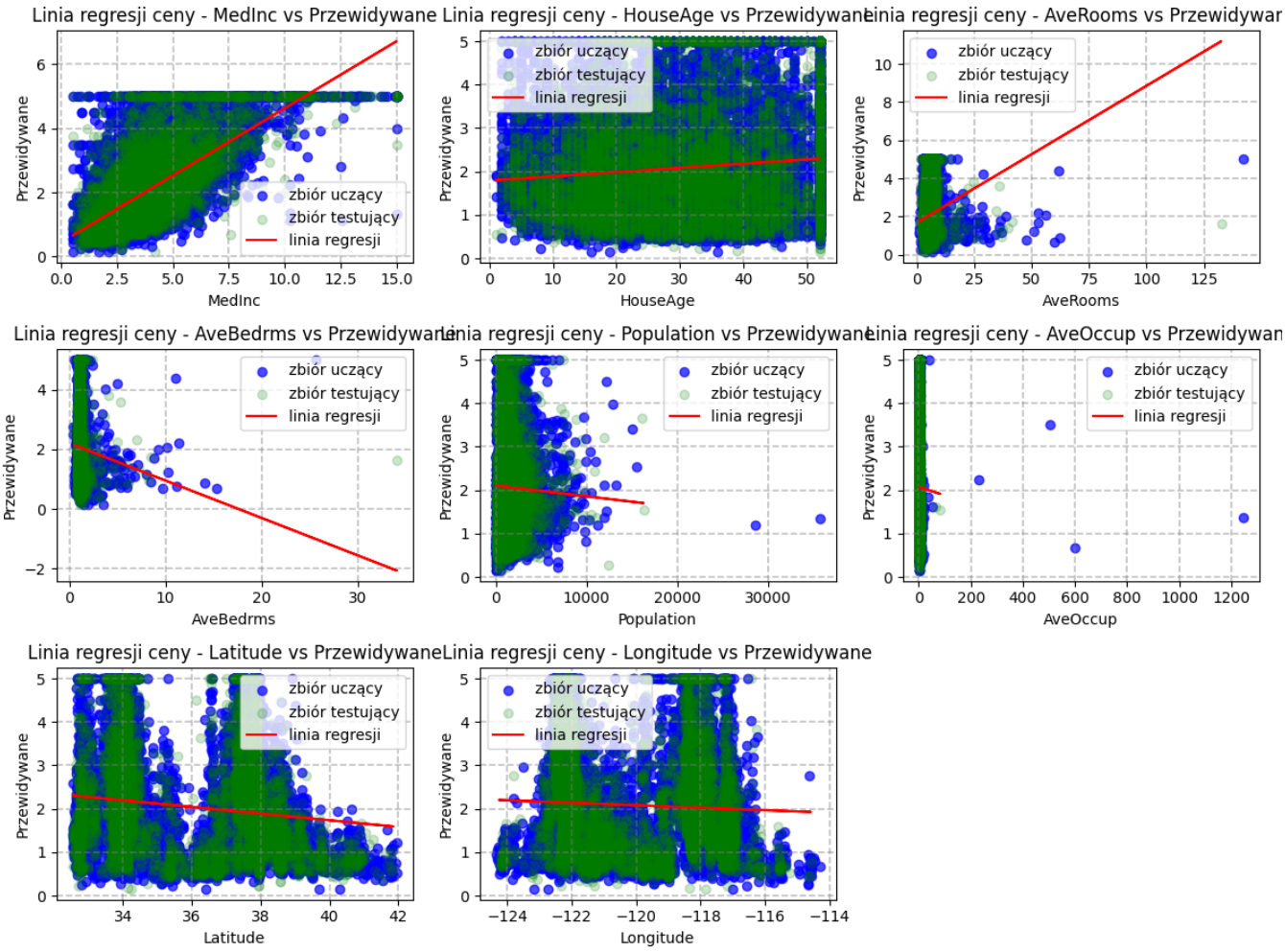
```python
'''Prognoza na danych uczących i testowych'''
y_train_pred = final_model.predict(X_train)
y_test_pred = final_model.predict(X_test)

'''Obliczenie błędów MAE i MSE na danych uczących i testowych'''
mae_train = mean_absolute_error(y_train, y_train_pred)
mse_train = mean_squared_error(y_train, y_train_pred)
mae_test = mean_absolute_error(y_test, y_test_pred)
mse_test = mean_squared_error(y_test, y_test_pred)

'''Ocena jakości modelu'''
print("\nOcena jakości modelu:")
print(f"Dane uczące - MAE: {mae_train:.4f}, MSE: {mse_train:.4f}")
print(f"Dane testujące - MAE: {mae_test:.4f}, MSE: {mse_test:.4f}")

'''Tworzenie tabeli z wynikami'''
table = tabulate(
    zip(feature_names, mae_values, mse_values),
    headers=['Feature', 'MAE', 'MSE'],
    tablefmt='double_grid'
)
print(table)
```

Kod 2. Fragment kodu rysującego .

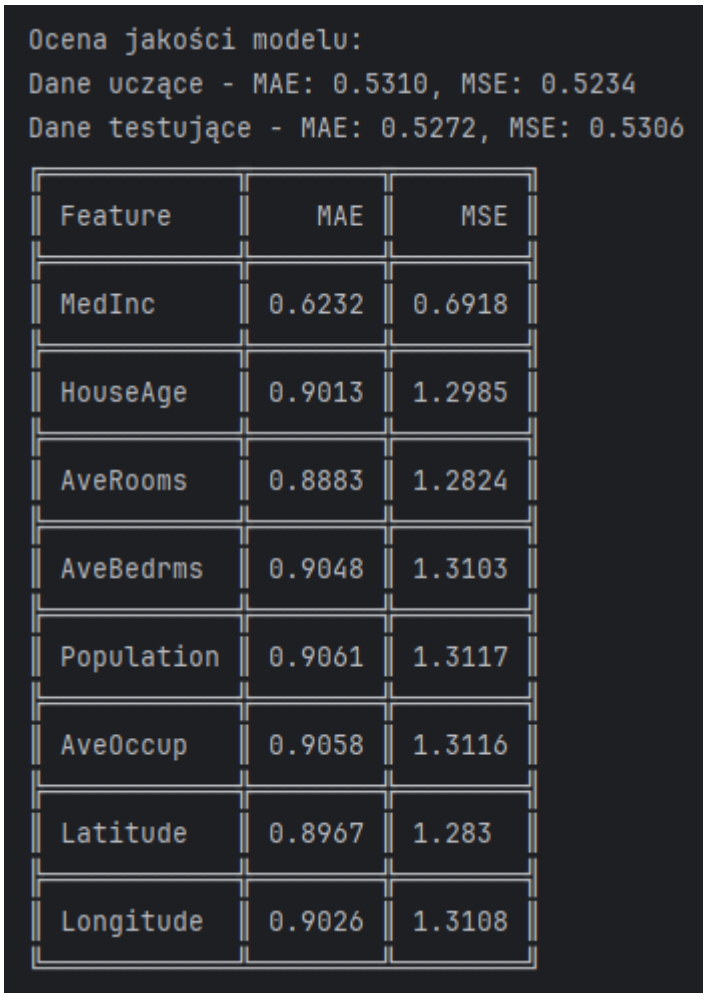


Rysunek 2. Reprezentacja graficzna modelu z linią regresji.

3. Ocena jakości modelu i wnioski

Modele MAE i MSE są powszechnie używane jako metryki oceny w modelach regresji. MAE mierzy średnią odległość między danymi rzeczywistymi a danymi przewidywanymi, podczas gdy MSE mierzy średnią kwadratową różnicę między wartościami szacunkowymi a wartością rzeczywistą.

Po analizie cech przeprowadziłem trenowanie ostatecznego modelu regresji liniowej na całym zbiorze uczącym. Następnie *print* (Rysunek 3.) jakości tego modelu na danych uczących oraz testowych za pomocą błędów MAE(prognozowanie, średni błąd absolutny) i MSE(średni błąd kwadratowy). Im wyniki są mniejsze, tym dokładność modelu jest większa.



Feature	MAE	MSE
MedInc	0.6232	0.6918
HouseAge	0.9013	1.2985
AveRooms	0.8883	1.2824
AveBedrms	0.9048	1.3103
Population	0.9061	1.3117
AveOccup	0.9058	1.3116
Latitude	0.8967	1.283
Longitude	0.9026	1.3108

Rysunek 3. Ocena jakości modelu..

MAE	MSE
• Założeniem błędu bezwzględnego jest uniknięcie wzajemnego kasowania się błędów dodatnich i ujemnych; • Błąd bezwzględny ma tylko wartości nieujemne;	• Błąd kwadratowy opiera się na tej samej idei, co błąd bezwzględny unikając ujemnych wartości błędów; • Ze względu na kwadrat uwypuklane są duże błędy i mają relatywnie większy wpływ na wartości metryki;

• Nie da się określić wzajemnego zniesienia – skośności; • Zachowuje te same jednostki miary, co analizowane dane i nadaje im tą samą wagę; • Odległość tą można zagregować na średni błąd arytmetyczny; • Użycie wartości bezwzględnej może powodować trudności w obliczaniu gradientu parametrów modelu Wykorzystywany w metryce MdAE	• Wpływ stosunkowo małych błędów będzie jeszcze mniejszy; • Jest określany penalizujący ekstremalne błędy; • Podatny na błędy odstające; • W przypadku danych odstających MSE stanie się większe w porównaniu od MAE; • Od momentu podniesienia błędu do kwadratu, każdy błąd przewidywania jest surowo karany