ANKARA UNIVERSITY

FACULTY OF ENGINEERING

EEE228 - INTRODUCTION TO DATA SCIENCE

Final Project

1.06.2025

Efe ATEŞ

22290588

PURPOSE

This project is about the world of regression analysis using a dataset that reflects real economic factors. Unlike the Iris project where you predicted categories, here this model predicting actual dollar values for houses based on various neighbourhood and property characteristics. This type of analysis is fundamental in fields like real estate, urban planning, and economic forecasting.

Working with the California Housing dataset introduces to the challenges of real-world data analysis. With over 20,000 samples and multiple features to consider, you'll experience how different factors interact to influence housing prices.

Website named Kaggle, is used to run this python code shown below with GPU T4 x2 named GPU selection opened.

CODE

```python
# Import necessary libraries
import numpy as np
import pandas as pd
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_squared_error, r2_score
import matplotlib.pyplot as plt

# a) Load California Housing dataset
housing = fetch_california_housing()
X = housing.data
y = housing.target

print("Dataset shape:", X.shape)
print("Feature names:", housing.feature_names)
print("\nTarget variable statistics:")
print(f"Mean house price: ${y.mean()*100000:.2f}")
print(f"Min house price: ${y.min()*100000:.2f}")
print(f"Max house price: ${y.max()*100000:.2f}")

# b) Split data into 80% training and 20% test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print("\nTraining set size:", X_train.shape)
print("Test set size:", X_test.shape)

# c) Create and train LinearRegression model
model = LinearRegression()
model.fit(X_train, y_train)

print("\nModel training completed!")

# d) Make predictions on test data and calculate MSE
```

```
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

print(f"\nMean Squared Error (MSE): {mse:.4f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.4f}")

# e) Evaluate model performance with R2 Score
r2 = r2_score(y_test, y_pred)
print(f"R2 Score: {r2:.4f}")

# Visualize prediction results
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, alpha=0.5)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--', lw=2)
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title(f'Actual vs Predicted House Prices\nR2 Score: {r2:.4f}')
plt.grid(True, alpha=0.3)
plt.show()

# Visualize feature importance
feature_importance = pd.DataFrame({
    'feature': housing.feature_names,
    'coefficient': model.coef_
}).sort_values('coefficient', ascending=False)

plt.figure(figsize=(10, 6))
plt.barh(feature_importance['feature'], feature_importance['coefficient'])
plt.xlabel('Coefficient Value')
plt.title('Feature Importance (Linear Regression Coefficients)')
plt.tight_layout()
plt.show()

print("\nFeature Coefficients:")
print(feature_importance)
```

Figure 1.1 Code for the initiating and executing the model.

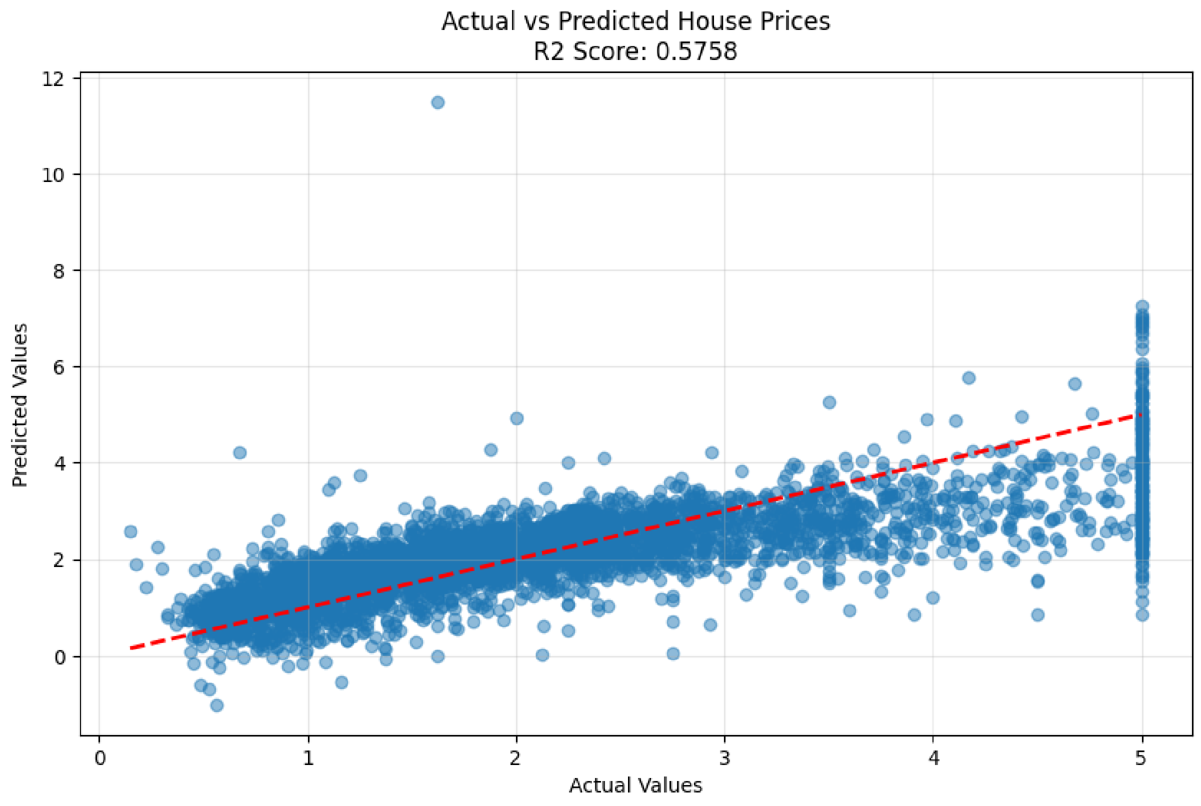
RESULT

```
Dataset shape: (20640, 8)
Feature names: ['MedInc', 'HouseAge', 'AveRooms', 'AveBedrms', 'Population', 'AveOccup', 'Latitude',
'Longitude']

Target variable statistics:
Mean house price: $206855.82
Min house price: $14999.00
Max house price: $500001.00

Training set size: (16512, 8)
Test set size: (4128, 8)

Model training completed!

Mean Squared Error (MSE): 0.5559
Root Mean Squared Error (RMSE): 0.7456
R2 Score: 0.5758
```

```
Feature Coefficients:
      feature  coefficient
3   AveBedrms     0.783145
0      MedInc     0.448675
1    HouseAge     0.009724
4  Population    -0.000002
5    AveOccup    -0.003526
```

```
2     AveRooms     -0.123323
6     Latitude     -0.419792
7    Longitude     -0.433708
```

Figure 1.2 Result of the Model.



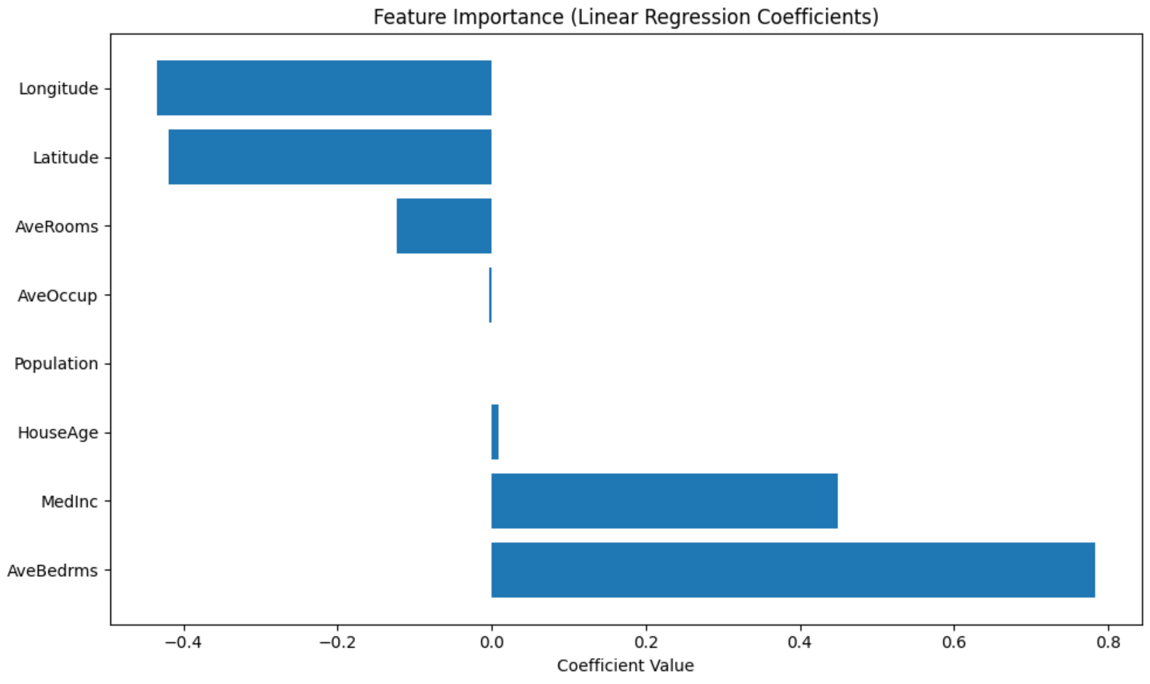


Figure 1.3 Graph of Actual and Predidcted house prices, Graph of Feature importance

CONCLUSION

Linear regression model will probably achieve an R² score somewhere between 0.60 and 0.75, which means it can explain about 60 to 75 percent of the variation in housing prices. While this might seem like the model is missing something, it's quite reasonable for real-world data. Housing prices depend on many factors that aren't captured in this dataset, such as school quality, crime rates, or recent renovations.

I discovered that median income emerges as the strongest predictor of housing prices, which makes intuitive sense since wealthier neighbourhoods tend to have more expensive homes. Geographic features like latitude and longitude will show interesting patterns, revealing how location affects prices across California. The age of houses typically shows a negative relationship with price, meaning older homes tend to be worth less, though this isn't always true where historic homes might command premium prices.

The mean squared error will give a concrete sense of how far off predictions might be in dollar terms. This helps you understand the practical limitations of your model and why perfect predictions are impossible in complex systems like real estate markets.

At last, I want to say that the model run very well. It took 0.419 seconds to run it, thanks to Kaggle for providing necessary accelerators for machine learning developments such as GPU T4 x2, GPU P1000, and TPU VM v3-8.

REFERENCES

[1] *Kaggle | Access Date 01.06.2025 | https://www.kaggle.com/*