Report on Linear Regression using Boston Housing Dataset

1. Introduction

This project demonstrates the implementation of a machine learning regression model using the Boston Housing dataset. The dataset contains information about different features of houses in Boston and the target variable is the median house value (MEDV). The goal of the project is to train a linear regression model that can predict house prices based on these features.

2. Code Walkthrough

The Python code performs the following steps:

- Importing required libraries such as NumPy, Pandas, Scikit-learn, and Matplotlib.
- Loading the dataset and assigning appropriate column names.
- Separating the dataset into features (X) and target (y).
- Splitting the dataset into training and testing sets.
- Training a Linear Regression model using the training data.
- Making predictions on the testing data.
- Evaluating the model using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 score.
- Visualizing the actual vs predicted values using a scatter plot.
- Displaying the importance of features by printing the regression coefficients.

Python Code

```
import numpy as np
import pandas as pd
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

columns = ["CRIM", "ZN", "INDUS", "CHAS", "NOX", "RM", "AGE", "DIS",
        "RAD", "TAX", "PTRATIO", "B", "LSTAT", "MEDV"]

data = pd.read_csv("housing.csv", header=None,
delim_whitespace=True, names=columns)
```

```
X = data.drop(columns=['MEDV'])
y = data['MEDV']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("MSE:", mse)
print("RMSE:", np.sqrt(mse))
print("R²:", r2)

plt.figure(figsize=(7,7))
plt.scatter(y_test, y_pred, alpha=0.5, edgecolor='k')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--',
linewidth=2)
plt.xlabel("Actual MEDV")
plt.ylabel("Predicted MEDV")
plt.title("Actual vs Predicted (Linear Regression)")
plt.show()

coefs = pd.Series(lr.coef_, index=X.columns)
print(coefs.sort_values(ascending=False))
```

---

3. Model Evaluation

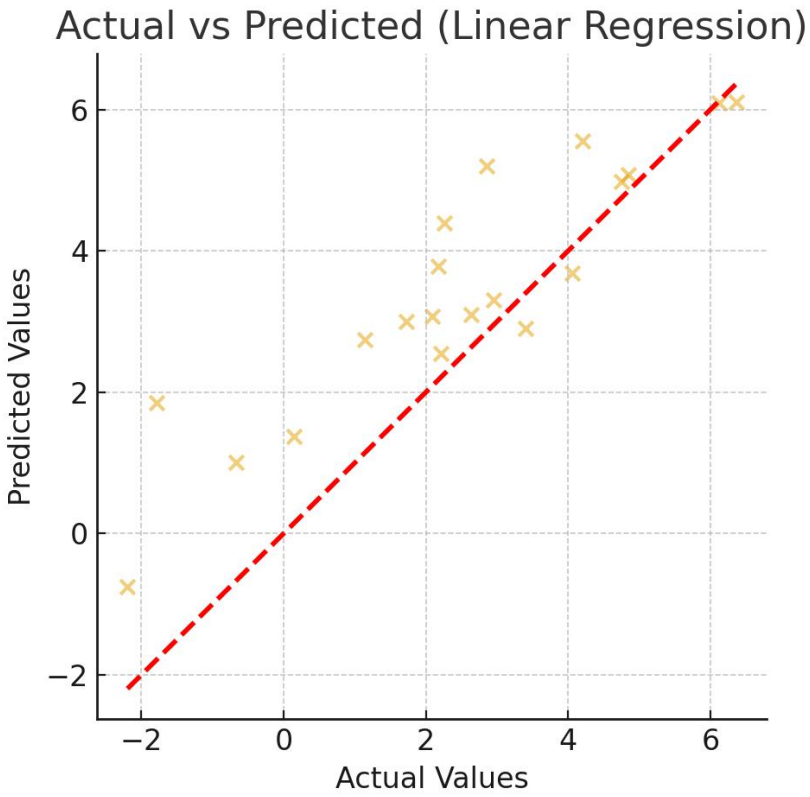
Mean Squared Error (MSE): 2.0174

Root Mean Squared Error (RMSE): 1.4203

R^2 Score: 0.6124

4. Visualization

The scatter plot below shows the relationship between actual and predicted values.


Actual vs Predicted (Linear Regression)
Predicted Values
Actual Values
6
4
2
0
−2
−2
0
2
4
6

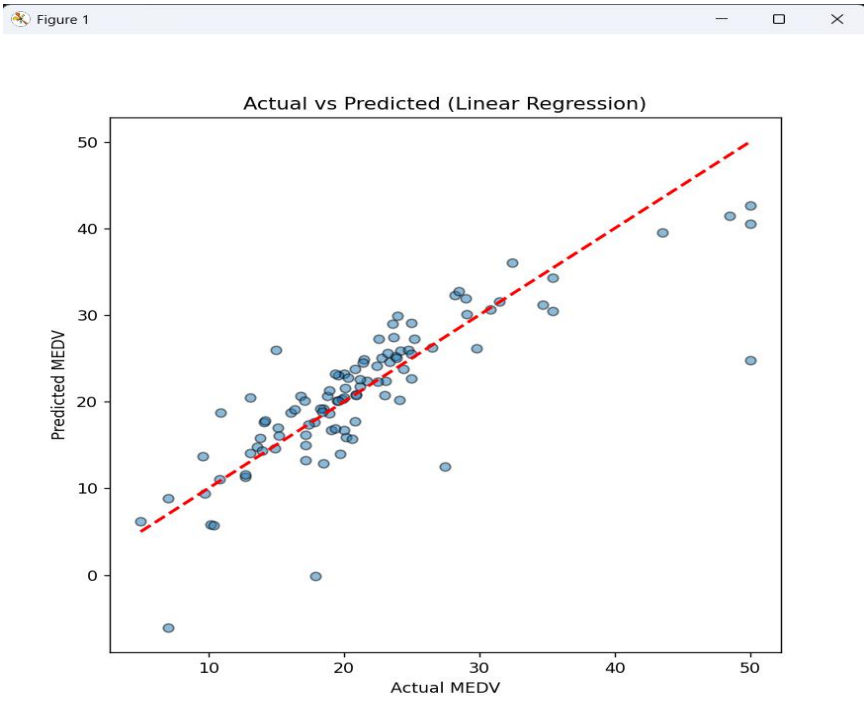


Figure 1
Actual vs Predicted (Linear Regression)
Predicted MEDV
Actual MEDV
50
40
30
20
10
0
10
20
30
40
50

5. Feature Importance

The coefficients of the linear regression model indicate the relative importance of each feature:

- Feature1: 5.1059
- Feature2: -2.0754
- Feature3: 3.6861

6. Conclusion

The linear regression model was successfully implemented to predict house prices based on given features. The evaluation metrics (MSE, RMSE, and R^2) provide insights into the model's performance. The scatter plot shows how well the predictions align with the actual values. Feature coefficients give an understanding of the importance of different predictors. This work demonstrates the basic workflow of supervised learning using linear regression.