Name: Vanesse V. Reyes
Y&S: BSCS 3B IS
Course: CSST 102 | Basic Machine Learning
Topic: Topic 2: Supervised Learning Fundamentals
Laboratory Exercise 1: Linear Regression Implementation

1. Data Preprocessing

Introduction

Data preprocessing is a crucial step in preparing our dataset for analysis and modeling. It involves transforming raw data into a clean and suitable format for our linear regression model.

Loading the Dataset

To begin, we loaded the dataset into a Pandas DataFrame, which provides a structured and tabular format for our data. The dataset comprises the following features:

- Size (sqft): The house size in square feet.
- Bedrooms: Number of bedrooms in the house.
- Age: The age of the house in years.
- Proximity to Downtown (miles): Distance from the downtown area.
- Price: The price of the house in thousands of dollars.

Here are the first few rows of the dataset:

Size	Bedroom	Age	Proximity to Downtown(miles)	Price
3974	1	97	2.032719	1162771
1660	5	88	23.695207	490002.1
2094	4	49	6.440232	640073.7
1930	2	28	8.129315	563788.1
1895	1	56	5.358837	565128.9

Handling Missing Values

Upon inspection, we found that there were no missing values in the dataset, ensuring that our data is complete and ready for analysis.

Normalization

To prepare the features for modeling, we normalized them using StandardScaler from sklearn.preprocessing. This process standardizes the features to have a mean of zero and a standard deviation of one, which is crucial for enhancing model performance.

2. Model Implementation

Linear Regression Class

We implemented a SimpleLinearRegression class that includes methods for fitting the model to the training data and making predictions based on new input data.

- **Initialization (__init__):** This method sets up empty coefficients and an intercept.
- **Fitting (fit):** This method employs the least squares method to compute parameters based on the training data.
- **Prediction (predict):** This method estimates house prices based on input features.

Model Training

The dataset was divided into training (80%) and testing (20%) sets using train_test_split. We trained our linear regression model on the training data and evaluated its performance using Mean Squared Error (MSE).

3. Model Evaluation

Assessing Model Performance

To evaluate how well our linear regression model performs on new, unseen data, we assessed it using the testing set. The primary metric used was Mean Squared Error (MSE), which measures the average squared difference between predicted and actual values.

- **Mean Squared Error on the Testing Set: 103,564,728.18**

This MSE value is comparable to that obtained from the training set, indicating that our model has successfully generalized from the training data rather than merely memorizing it. A similar MSE across both datasets suggests that our model effectively captures underlying patterns in house pricing.

Visual Representation of Results

To illustrate the model's effectiveness visually, we created a plot showing actual house prices against their respective sizes (in square feet) alongside the regression line generated by our model.In this visualization, 'Size (sq. ft.)' serves as the primary feature for analysis. The scatter plot displays individual data points representing actual prices, while the regression line indicates how our model predicts prices based on house size.The following code snippet demonstrates how this visualization was created:

```
import matplotlib.pyplot as plt

# Predict house prices on the testing set
predicted_test_prices = linear_model.predict(features_test)

# Calculate Mean Squared Error for testing data
mse_testing = mean_squared_error(target_test, predicted_test_prices)
print("Mean Squared Error on the testing set:", mse_testing)
# Create a plot to visualize predictions vs actual prices
plt.figure(figsize=(12, 6))
plt.scatter(features_test[:, 0], target_test, color='teal', label='Actual Prices')

plt.plot(size_values, predicted_prices_line, color='coral', label='Regression Line')

plt.xlabel('Size (sq. ft.)')

plt.ylabel('Price (in thousands)')

plt.title('Relationship Between House Size and Price')

plt.legend()

plt.grid(True)

plt.show()
```

This graphical representation effectively conveys how well our linear regression model captures the relationship between house size and price. Analyzing this plot allows us to gain insights into prediction accuracy and identify potential areas for improvement in future modeling efforts. This revised report maintains clarity while providing detailed explanations of each section related to your linear regression implementation project.