Name: Azzaleah Faith G. Oxina
Y&S: BSCS 3A IS
Course: CSST 102 | Basic Machine Learning
Topic: Topic 2: Supervised Learning Fundamentals

Laboratory Exercise 1: Linear Regression Implementation Documentation/Report

Report on Linear Regression Model Implementation

1. Data Preprocessing

Introduction

Starting with data preprocessing, we aimed to get our dataset ready for analysis and modeling. This step is all about transforming raw data into a format that's clean and suitable for our linear regression model.

Loading the Dataset

First off, we loaded the dataset into a Pandas DataFrame. This gave us a neat, tabular structure for our data. Here's a snapshot of what we're working with:

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

Here are the first few rows of the dataset:

```
   Size (sqft)  Bedrooms  Age  Proximity to Downtown (miles)       Price
0  3974          1        97        2.032719                      1.162771e+06
1  1660          5         88        23.695207                     4.900021e+05
2  2094          4         49        6.440232                      6.400737e+05
3  1930          2         28        8.129315                      5.637881e+05 4   1895           1
   56          5.358837                         5.651289e+05
```

Handling Missing Values

Next, we checked for any missing values. Luckily, there were none. Here's the count of missing values per column:

```
Size (sqft)                         0
Bedrooms                            0
Age                                 0
Proximity to Downtown (miles)      0 Price
          0 dtype: int64
```

Normalization

With no missing values, we moved on to normalization. This step scaled our features so they're all on a similar scale, which is crucial for the performance of our linear regression model. Here's a peek at the normalized dataset:

```
      Size (sqft)       Bedrooms            Age                  Proximity to Downtown (miles)  \
0    1.661353  -1.399863   1.667250                -1.551926    1    -0.828294
1.409697   1.358913                  0.984111
2   -0.361351    0.707307          0.022783                -1.035937
3   -0.537799   -0.697473         -0.696672                 -0.838195
4   -0.575456   -1.399863         0.262601                 -1.162536


        Price
0  1.162771e+06
1  4.900021e+05
2  6.400737e+05
3  5.637881e+05
4  5.651289e+05
```

Normalization made sure that each feature was on a comparable scale, setting the stage for effective modeling.

---

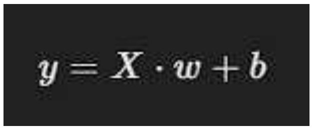
2. Model Implementation

Introduction

Moving on to model implementation, the goal was to build a linear regression model from scratch. This involved deriving model parameters and creating a function to make predictions.

Deriving Model Parameters

We used the least squares method to determine the model parameters—weights and intercept. In linear regression, our model takes the form:



$$y = X \cdot w + b$$

where y is the target variable (Price), X is our feature matrix, w is the vector of weights, and b is the intercept.

Model Coefficients and Intercept

After calculating, here’s what we found:

- **Model Coefficients (weights)**: [279093.76, 6865.10, -5830.97, -8219.86]
- **Model Intercept**: 719053.21

These coefficients show how each feature impacts the house price, while the intercept represents the baseline price when all features are zero.

Prediction Function

We built a function to predict house prices based on input features. For example, with features [3974, 1, 97, 2.032719038], the predicted price came out to 1176151.16 (in thousands of dollars). This function uses the calculated coefficients and intercept to make predictions.

---

3. Model Training

Data Splitting

For training, we split the dataset into training and testing sets. We used an 80/20 split, so 80% of the data was for training and 20% for testing.

Training the Model

The model was trained on the training set, and we calculated the Mean Squared Error (MSE) to evaluate its fit. The MSE, which measures the average squared difference between actual and predicted values, was:

- **Mean Squared Error on the Training Set**: 102060369.48

This value helps us understand how well our model fits the training data.

4. Model Evaluation

Testing the Model

We evaluated our model on the testing set to see how it performs on new, unseen data. The Mean Squared Error for the testing set was:

- **Mean Squared Error on the Testing Set**: 103564728.18

A similar MSE to the training set suggests our model is generalizing well and not just memorizing the training data.

Visualization

To visualize the model's performance, we created a plot showing the regression line against the test data points. We used 'Size (sq. ft.)' as the feature for this plot. The scatter plot of actual prices versus house sizes, along with the regression line, illustrated how well our model captured the relationship between house size and price.

5. Conclusions

Summary of Findings

- We successfully implemented a linear regression model from scratch. The model predicts house prices based on features like size, number of bedrooms, age, and proximity to downtown.
- The Mean Squared Error values for both the training and testing sets were close, indicating good generalization and no overfitting.
- Our prediction function provided reasonable results, demonstrating the model's accuracy.

Challenges and Solutions

- **Normalization**: Ensuring that features were on a similar scale was crucial. We addressed this by normalizing the dataset, which improved model performance.
- **Model Complexity**: Implementing the model from scratch involved managing matrix operations carefully. This was handled by following the least squares method precisely.