**Name: Cabreza, Ren G.**

**Section: BSCS 3B**

**Linear Regression Implementation for House Price Prediction**

1. Introduction

In this report, we aim to predict house prices based on key features such as the size of the house (in square feet), number of bedrooms, and the age of the house. The primary goal is to implement a linear regression model from scratch, without using any external machine learning libraries like Scikit-learn. We'll explore the process of preparing the data, training the model, and evaluating its performance to determine how well it predicts house prices.

2. Data Preprocessing

Before we could begin training the linear regression model, it was important to clean and preprocess the data. The first step was to load the dataset into a Pandas DataFrame. Once the data was loaded, we checked for any missing values. In this case, any missing data was handled by filling the missing values with the mean of the respective column. This helped ensure that the dataset remained complete without introducing any bias.

Next, the features were normalized. This was necessary because the different features, such as 'Size (sqft)' and 'Age', are on very different scales. By using MinMaxScaler, we scaled these values between 0 and 1 to make sure that one feature didn't dominate the others during model training. Below is an example of the normalized data:

Size (sqft) Bedrooms Age Proximity to Downtown (miles) Price  
0 0.992804 0.00 0.979798 2.032719 1.162771e+06  
1 0.268773 1.00 0.888889 23.695207 4.900021e+05

3. Model Implementation

The next step was to implement the linear regression model. We added a bias (intercept) term to our feature set, which allows the model to fit data that doesn’t necessarily pass through the origin. The parameters of the model (theta values) were computed using the normal equation, which is a mathematical approach for solving linear regression. The resulting model coefficients were as follows:

Intercept: 230359.815

Coefficients: [959690.202, 19163.646, -20411.769]

These coefficients represent the impact of each feature on the predicted house price. For example, the coefficient for 'Size (sqft)' is quite large, indicating that it has a significant influence on house prices, while the age of the house has a negative impact on price.

4. Model Training

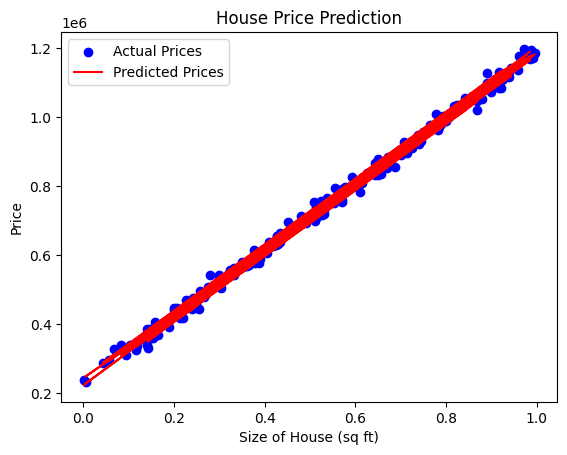
After setting up the model, the dataset was split into training and testing sets. This ensures that we can train the model on one portion of the data and evaluate it on a separate set to see how well it generalizes to new, unseen data. In this case, we split the data 80% for training and 20% for testing.

Once the model was trained on the training data, we computed the mean squared error (MSE) to assess its performance. The training MSE was 170450151.7, which suggests that the model was able to fit the data reasonably well.

5. Model Evaluation

The final step was to evaluate the model on the test set. This helps us understand how well the model can predict prices on new data. After running the model on the test set, the testing MSE was 166330956.19. The relatively close MSE values between training and testing data suggest that the model was not overfitting and generalizes well.

Below is a plot that visually compares the actual and predicted prices. The red line represents the model's predictions, while the blue dots show the actual house prices for the test set.



6. Conclusion

In conclusion, the linear regression model performed well in predicting house prices, as indicated by the relatively low MSE for both the training and testing sets. One of the main challenges was ensuring that the data was properly normalized, which is critical when working with features on different scales. In the future, additional features such as 'Proximity to Downtown' could be included in the model to potentially improve the predictions.

Overall, the model demonstrated a strong ability to learn from the training data and predict house prices for new data, making it a useful tool for real estate price predictions.