

Student's Name

Department, Institutional Affiliation

Course Name and Number

Professor's Name

Due Date

Report on a Linear Regression Model for the Prediction of House Prices

Introduction to Machine Learning

Machine learning is a subfield of artificial intelligence that focuses on the study of building models and algorithms that let computers learn from data and use that data to make predictions or choices (Hopkins, Emily 2022). In the context of this report, we will discuss the implementation of a linear regression model for the prediction of house prices.

The Linear Regression Model

A basic machine learning approach called linear regression is used to predict the connection between a dependent variable and one or more independent variables (James, Gareth, et al.2023). In this research, we forecast home prices using a linear regression model based on two independent features: the property's square footage and the number of bedrooms.

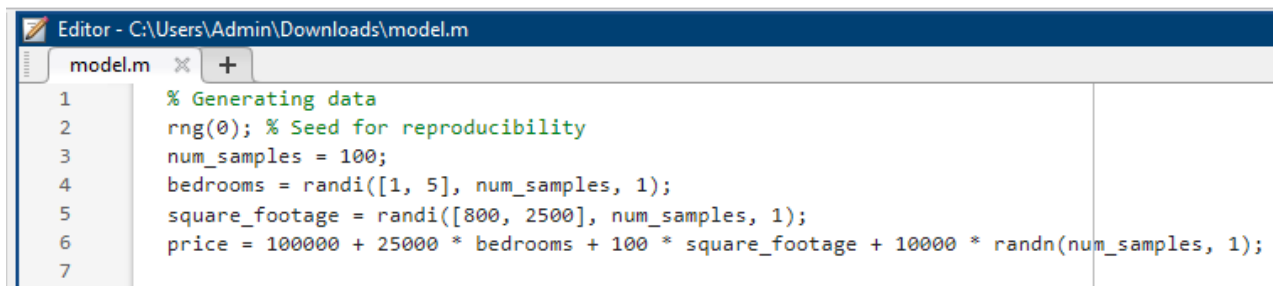
The implementation of a linear regression model for house price predictions offers simplicity, transparency, insights, and adaptability. It can provide accurate predictions and serve as a valuable tool in the real estate industry, helping stakeholders make informed decisions and better understand the factors influencing property prices. While linear regression may have its limitations, it remains a powerful and accessible tool for house price estimation.

Data Preparation

Before building the model, we generate a synthetic dataset with 100 samples to simulate house price data. The dataset is created by adding a random amount of noise to a linear relationship between the number of bedrooms, the square footage, and the price. The data is then split into input features (X) and the target variable (y), which represents the house prices.

Figure 1:

The screenshot below shows the implementation of Data Generation in the matlab model



```

Editor - C:\Users\Admin\Downloads\model.m
model.m  x  +
1  % Generating data
2  rng(0); % Seed for reproducibility
3  num_samples = 100;
4  bedrooms = randi([1, 5], num_samples, 1);
5  square_footage = randi([800, 2500], num_samples, 1);
6  price = 100000 + 25000 * bedrooms + 100 * square_footage + 10000 * randn(num_samples, 1);
7

```

Model Building/Implementation

The linear regression model is built using the normal equation approach. The model aims to find the coefficients (theta values) that minimize the mean squared error (MSE) between the predicted house prices and the actual house prices in the training dataset.

The equation for the linear regression model is as follows:

$$y_{pred} = \theta_0 + \theta_1 * bedrooms + \theta_2 * square_footage$$

Where:

- θ_0 is the intercept,
- θ_1 is the coefficient for the number of bedrooms, and
- θ_2 is the coefficient for square footage.

The coefficients (θ_0 , θ_1 , and θ_2) are calculated using the normal equation, which is used in the code as follows:

$$theta = (X' * X) \setminus (X' * y);$$

This normal equation calculates the optimal coefficients that define the linear relationship between the features and the target variable. These coefficients are used to make predictions and understand the influence of each feature on the target variable in the context of the linear regression model.

Figure 2:

The screenshot below shows the implementation of Model Building

```
% Separating the dataset into input features (X) and output target (y)
X = [ones(num_samples, 1), bedrooms, square_footage];
y = price;

% Performing linear regression using the normal equation
theta = (X' * X)\(X' * y);

% Display the results
fprintf('Theta values: %f %f %f\n', theta(1), theta(2), theta(3));

% Plotting the predicted values against the actual values
y_pred = X * theta;
figure;
scatter(y, y_pred, 'filled');
xlabel('Actual Price');
ylabel('Predicted Price');
title('House Price Prediction');

% using the model to make predictions for new data
new_data = [1, 3, 2000]; % 3 bedrooms, 2000 square footage
predicted_price = new_data * theta;
fprintf('Predicted Price for new data: %f\n', predicted_price);
```

Model Evaluation

To assess the model's performance, several evaluation metrics are implemented:

1. Mean Absolute Error (MAE):

MAE is a measure of the average absolute difference between the predicted and actual house prices.

2. Mean Squared Error (MSE):

MSE calculates the average squared difference between the predicted and actual house prices.

3. Root Mean Squared Error (RMSE):

RMSE is the square root of the MSE and provides a more interpretable measure of prediction error.

4. R-squared (R^2):

R-squared is a statistical measure indicating how well the model fits the data. It represents the proportion of the variance in the target variable that is predictable from the independent variables.

Figure 3:

Screenshot showing the implementation of Evaluation Metrics

```
% Calculate Mean Absolute Error (MAE)
mae = mean(abs(y_pred - y));

% Calculate Mean Squared Error (MSE)
mse = mean((y_pred - y).^2);

% Calculate Root Mean Squared Error (RMSE)
rmse = sqrt(mse);

% Calculate R-squared (R2)
SSR = sum((y_pred - mean(y)).^2); % Regression sum of squares
SST = sum((y - mean(y)).^2);      % Total sum of squares
r_squared = SSR / SST;

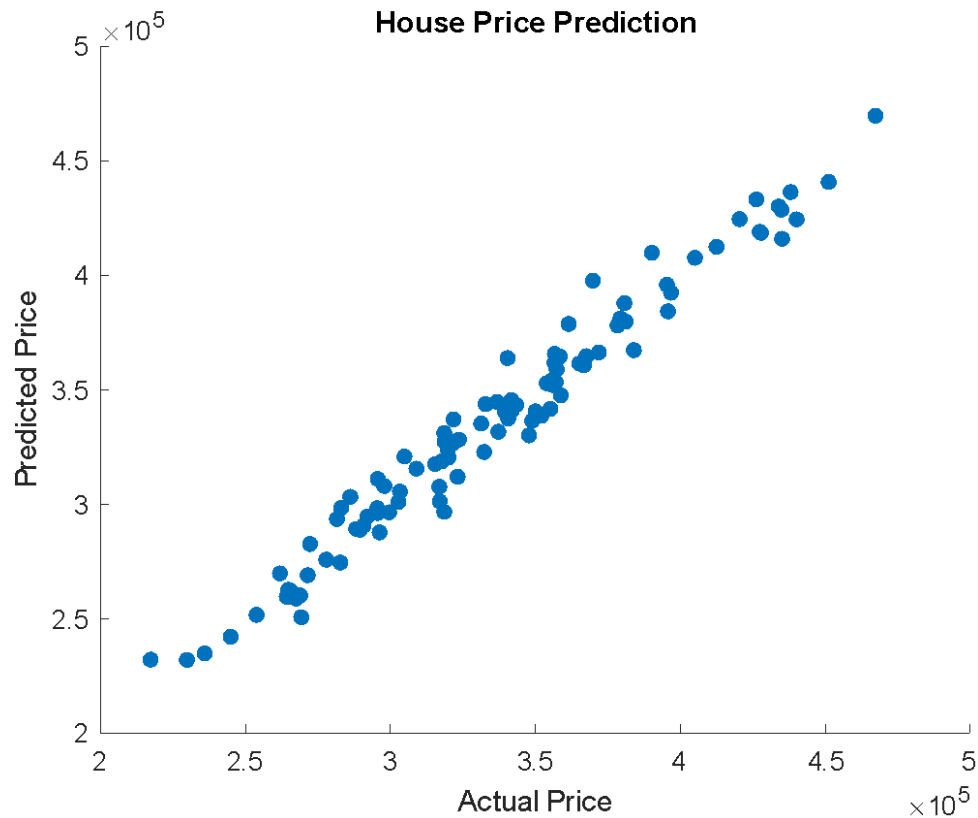
% Display the results including the metrics
fprintf('Mean Absolute Error (MAE): %f\n', mae);
fprintf('Mean Squared Error (MSE): %f\n', mse);
fprintf('Root Mean Squared Error (RMSE): %f\n', rmse);
fprintf('R-squared (R2): %f\n', r_squared);
```

Results and Discussion

After implementing the model and evaluating its performance, the following results were obtained:

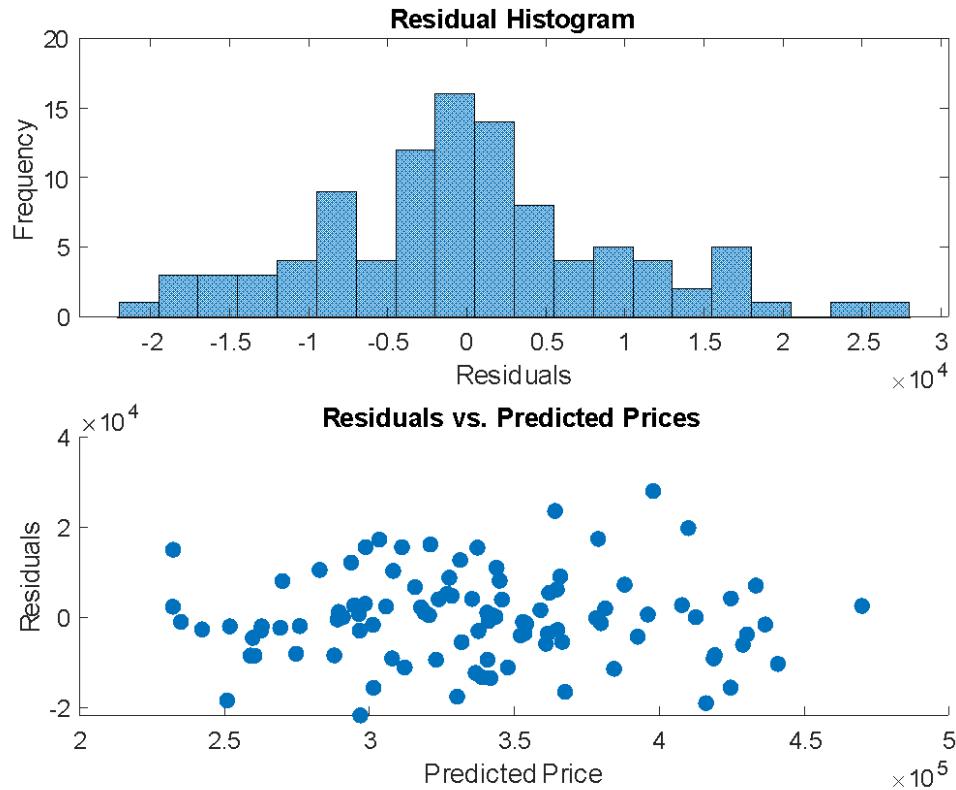
1. Theta values (model coefficients):
 - Intercept (θ_0): 106,103.60
 - Bedrooms coefficient (θ_1): 24,200.90
 - Square footage coefficient (θ_2): 97.33

Figure 4:

Plotting the predicted values against the actual values

2. Predicted Price for New Data (3 bedrooms, 2000 square footage): 373,372.28

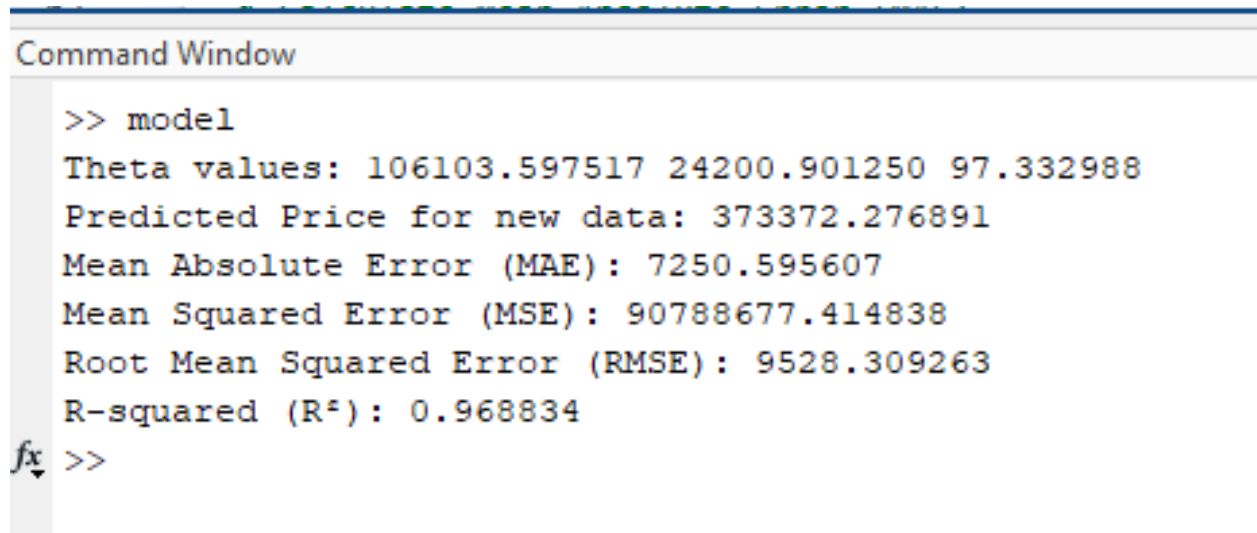
Figure 5:

Residual Plot and Scatter plot of residuals vs. predicted values**3. Evaluation Metrics:**

- Mean Absolute Error (MAE): 7,250.60
- Mean Squared Error (MSE): 90,788,677.41
- Root Mean Squared Error (RMSE): 9,528.31
- R-squared (R^2): 0.9688

These results indicate that the linear regression model has been successfully trained and performs well on the given dataset. The R^2 value of approximately 0.9688 suggests that the model explains a significant portion of the variance in house prices, and the low RMSE and MAE values demonstrate the model's accuracy in predicting prices.

Figure 6:

Screenshot showing the Model Results

```
Command Window

>> model
Theta values: 106103.597517 24200.901250 97.332988
Predicted Price for new data: 373372.276891
Mean Absolute Error (MAE): 7250.595607
Mean Squared Error (MSE): 90788677.414838
Root Mean Squared Error (RMSE): 9528.309263
R-squared (R2): 0.968834
fx >>
```

Conclusion

In conclusion, the linear regression model developed for house price prediction demonstrated strong performance on the synthetic dataset. The model's coefficients were estimated using the normal equation, and it provided accurate predictions for new data. The low MAE and RMSE values indicate that the model's predictions closely matched the actual house prices. With an R-squared value of 0.9688, the model explained a significant portion of the variance in house prices, making it a valuable tool for real estate price estimation.

However, it is important to note that this model was trained on synthetic data, and its performance may vary with real-world datasets. Further refinement and testing on real data are recommended for practical applications.

References

Hopkins, Emily. "Machine learning tools, algorithms, and techniques." *Journal of*

Self-Governance and Management Economics 10.1 (2022): 43-55.

James, Gareth, et al. "Linear regression." *An Introduction to Statistical Learning: With*

Applications in Python. Cham: Springer International Publishing, 2023. 69-134.