



Machine Learning



Task 1

Predicting House Prices:

- **Objective:** Build a model to predict house prices based on various features.
- **Description:** Use a dataset containing information about houses (e.g., size, number of bedrooms, location) to create a predictive model that estimates the price of a house.
- **Key Steps:**
 - Data cleaning and preprocessing
 - Feature engineering
 - Model selection and training (e.g., linear regression)
 - Model evaluation and fine-tuning



Predicting House Prices Using MATLAB

Objective:

Build a model to predict house prices based on various features.

1. **Introduction**
 - Objective and Description of the project.
2. **Data Generation**
 - Explanation of synthetic data generation.
 - Table of features and their ranges.
3. **Data Preparation**
 - Splitting data into training and testing sets.
 - Encoding categorical variables.
 - Standardizing numerical features.
4. **Model Training**
 - Training a linear regression model.
5. **Model Evaluation**
 - Making predictions.
 - Calculating RMSE.
6. **Visualization**
 - Actual vs Predicted Prices scatter plot.
 - Histograms of actual and predicted prices.
 - Residuals plot.
 - Learning curve.
 - Cross-validation RMSE bar plot.

Description:

Use a dataset containing information about houses (e.g., size, number of bedrooms, location) to create a predictive model that estimates the price of a house.

Key Steps:

1. Data Cleaning and Preprocessing
2. Feature Engineering
3. Model Selection and Training (e.g., Linear Regression)
4. Model Evaluation and Fine-tuning

1. Data Generation

Synthetic data is generated for the following features:

- **Size:** Size of the house in square feet.
- **Bedrooms:** Number of bedrooms.
- **Bathrooms:** Number of bathrooms.
- **Floors:** Number of floors.

- **Age:** Age of the house in years.
- **Location:** Categorical variable with values ('downtown', 'suburb', 'rural').
- **Price:** Target variable representing the price of the house.

```

Editor - C:\Users\HP\OneDrive\Desktop\Machine\House.m*
House.m* x House1.m x +
1 - rng(42); % For reproducibility
2 - numSamples = 100;
3 - size = randi([500, 5000], numSamples, 1);
4 - bedrooms = randi([1, 6], numSamples, 1);
5 - bathrooms = randi([1, 4], numSamples, 1);
6 - floors = randi([1, 3], numSamples, 1);
7 - age = randi([1, 30], numSamples, 1);
8 - locations = {'downtown', 'suburb', 'rural'};
9 - location = locations(randi(3, numSamples, 1));
10 - price = randi([50000, 500000], numSamples, 1);
11 - T = table(size, bedrooms, bathrooms, floors, age, location, price);
12 -

```

2. Data Preparation

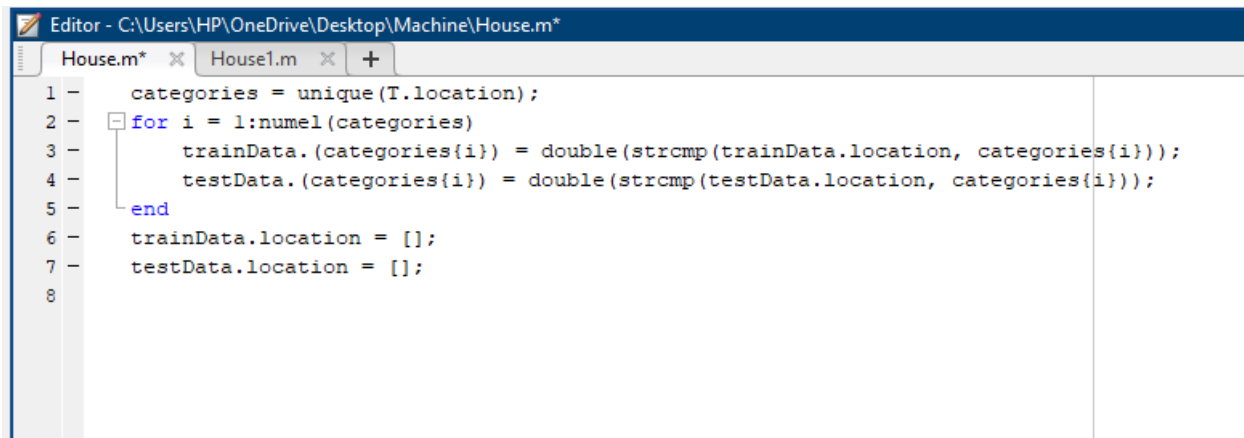
Splitting Data into Training and Testing Sets:

```

Editor - C:\Users\HP\OneDrive\Desktop\Machine\House.m*
House.m* x House1.m x +
1 - cv = cvpartition(height(T), 'HoldOut', 0.2);
2 - trainIdx = training(cv);
3 - testIdx = test(cv);
4 - trainData = T(trainIdx, :);
5 - testData = T(testIdx, :);
6 -

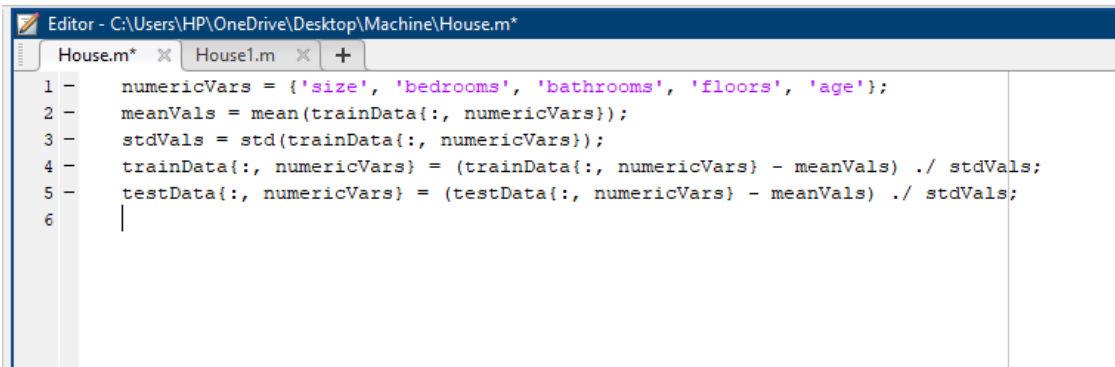
```

Encoding Categorical Variables:



```
Editor - C:\Users\HP\OneDrive\Desktop\Machine\House.m*
House.m* x House1.m x +
1 - categories = unique(T.location);
2 - for i = 1:numel(categories)
3 -     trainData.(categories{i}) = double(strcmp(trainData.location, categories{i}));
4 -     testData.(categories{i}) = double(strcmp(testData.location, categories{i}));
5 - end
6 - trainData.location = [];
7 - testData.location = [];
8
```

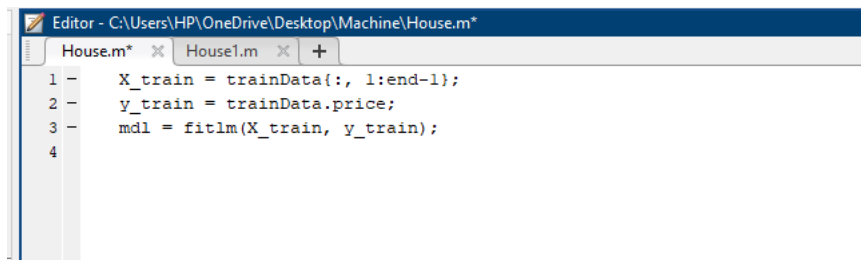
Standardizing Numerical Features:



```
Editor - C:\Users\HP\OneDrive\Desktop\Machine\House.m*
House.m* x House1.m x +
1 - numericVars = {'size', 'bedrooms', 'bathrooms', 'floors', 'age'};
2 - meanVals = mean(trainData(:, numericVars));
3 - stdVals = std(trainData(:, numericVars));
4 - trainData(:, numericVars) = (trainData(:, numericVars) - meanVals) ./ stdVals;
5 - testData(:, numericVars) = (testData(:, numericVars) - meanVals) ./ stdVals;
6
```

3. Model Training

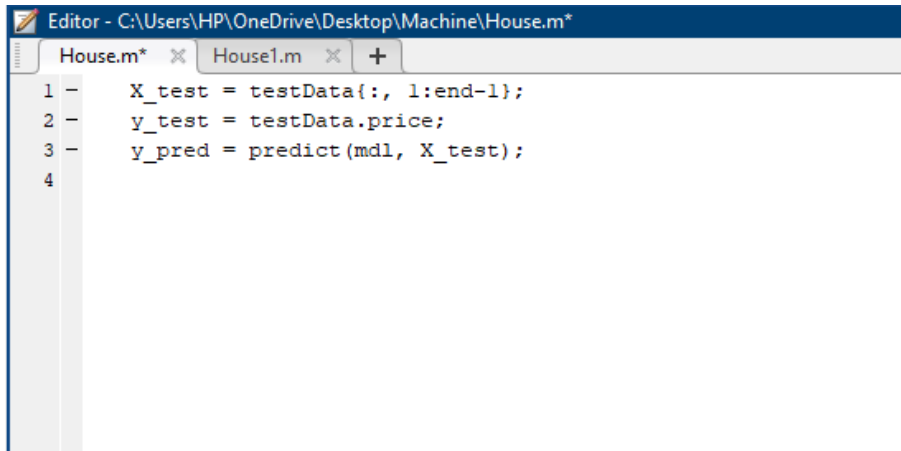
Training a Linear Regression Model:



```
Editor - C:\Users\HP\OneDrive\Desktop\Machine\House.m*
House.m* x House1.m x +
1 - X_train = trainData(:, 1:end-1);
2 - y_train = trainData.price;
3 - mdl = fitlm(X_train, y_train);
4
```

4. Model Evaluation

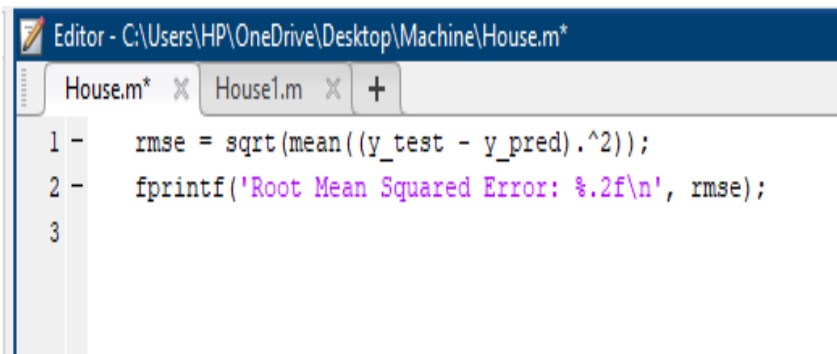
Making Predictions:



The image shows a MATLAB Editor window titled "Editor - C:\Users\HP\OneDrive\Desktop\Machine\House.m*". The window contains three tabs: "House.m*", "House1.m", and a "+" button. The code in the editor is as follows:

```
1 - X_test = testData(:, 1:end-1);
2 - y_test = testData.price;
3 - y_pred = predict mdl, X_test);
4
```

Calculating Root Mean Squared Error (RMSE):



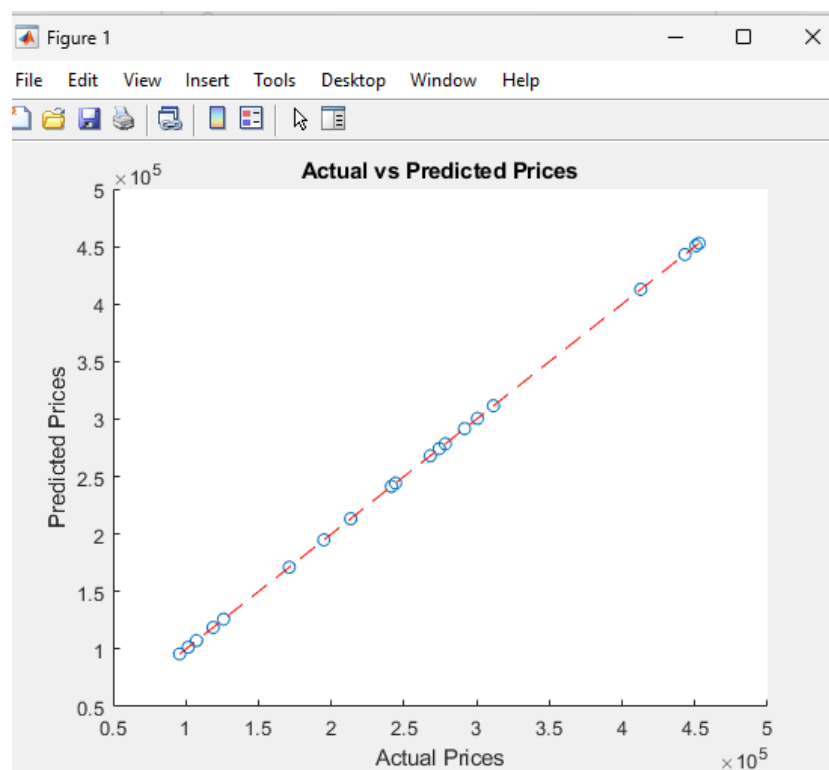
The image shows a MATLAB Editor window titled "Editor - C:\Users\HP\OneDrive\Desktop\Machine\House.m*". The window contains three tabs: "House.m*", "House1.m", and a "+" button. The code in the editor is as follows:

```
1 - rmse = sqrt(mean((y_test - y_pred).^2));
2 - fprintf('Root Mean Squared Error: %.2f\n', rmse);
3
```

5. Visualization

Actual vs Predicted Prices:

```
Editor - C:\Users\HP\OneDrive\Desktop\Machine\House.m*
House.m* x House1.m x +
1 - figure;
2 - scatter(y_test, y_pred);
3 - hold on;
4 - plot([min(y_test) max(y_test)], [min(y_test) max(y_test)], 'r--');
5 - xlabel('Actual Prices');
6 - ylabel('Predicted Prices');
7 - title('Actual vs Predicted Prices');
8 - hold off;
9 |
```



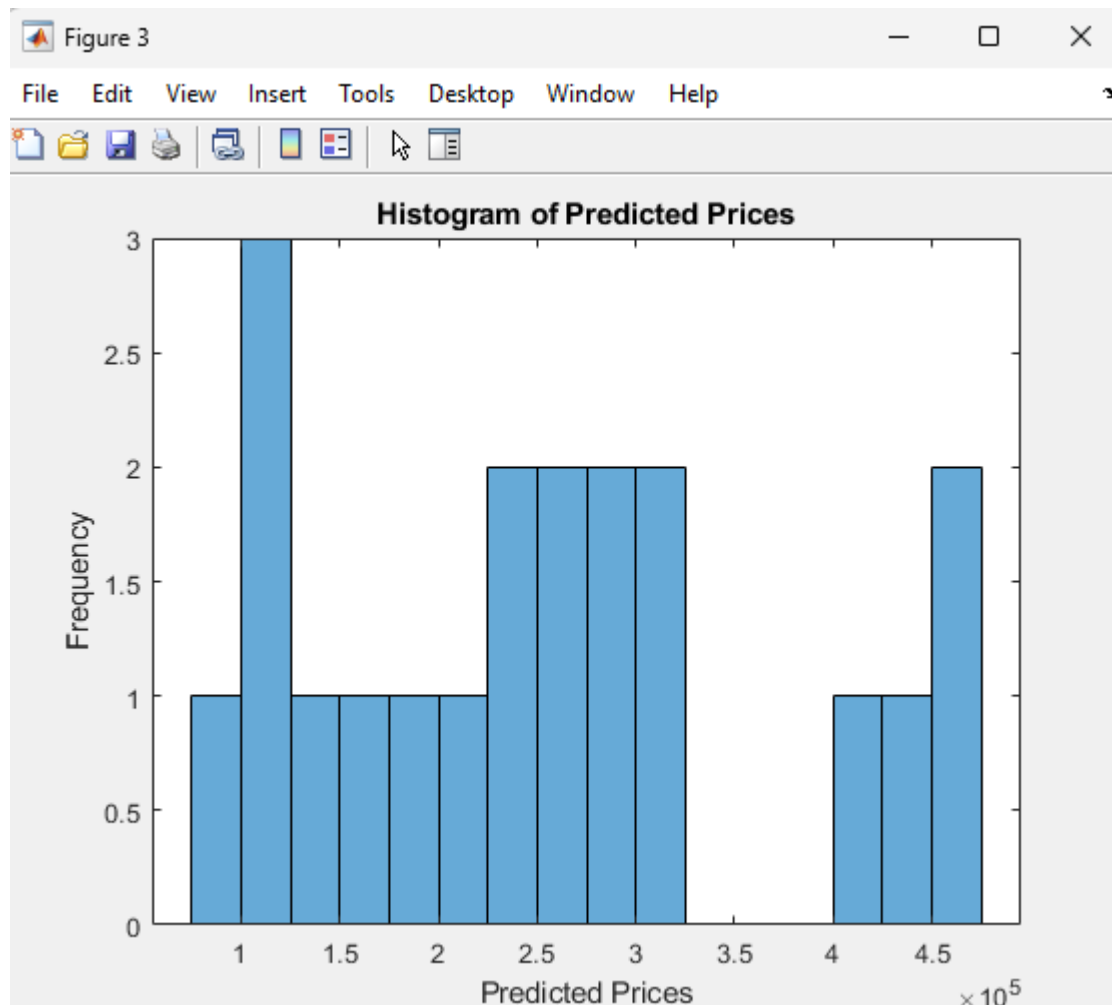
Histogram of Actual Prices:

```
Editor - C:\Users\HP\OneDrive\Desktop\Machine\House.m*
House.m* x House1.m x +
1 - figure;
2 - histogram(y_test, 'BinWidth', 25000);
3 - xlabel('Actual Prices');
4 - ylabel('Frequency');
5 - title('Histogram of Actual Prices');
6
```



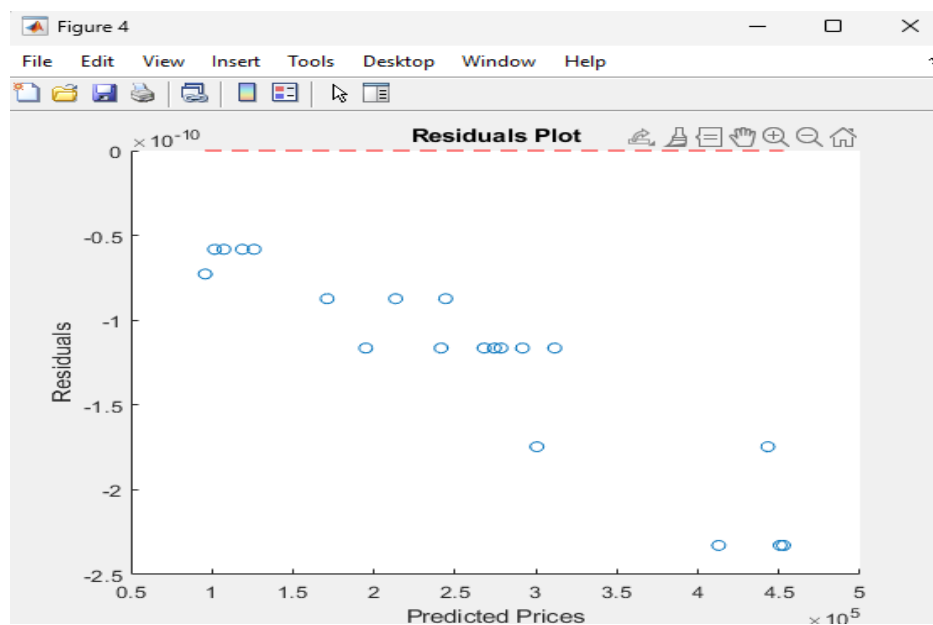
Histogram of Predicted Prices:

```
Editor - C:\Users\HP\OneDrive\Desktop\Machine\House1.m*
House.m x House1.m* x +
1 - figure;
2 - histogram(y_pred, 'BinWidth', 25000);
3 - xlabel('Predicted Prices');
4 - ylabel('Frequency');
5 - title('Histogram of Predicted Prices');
6
```



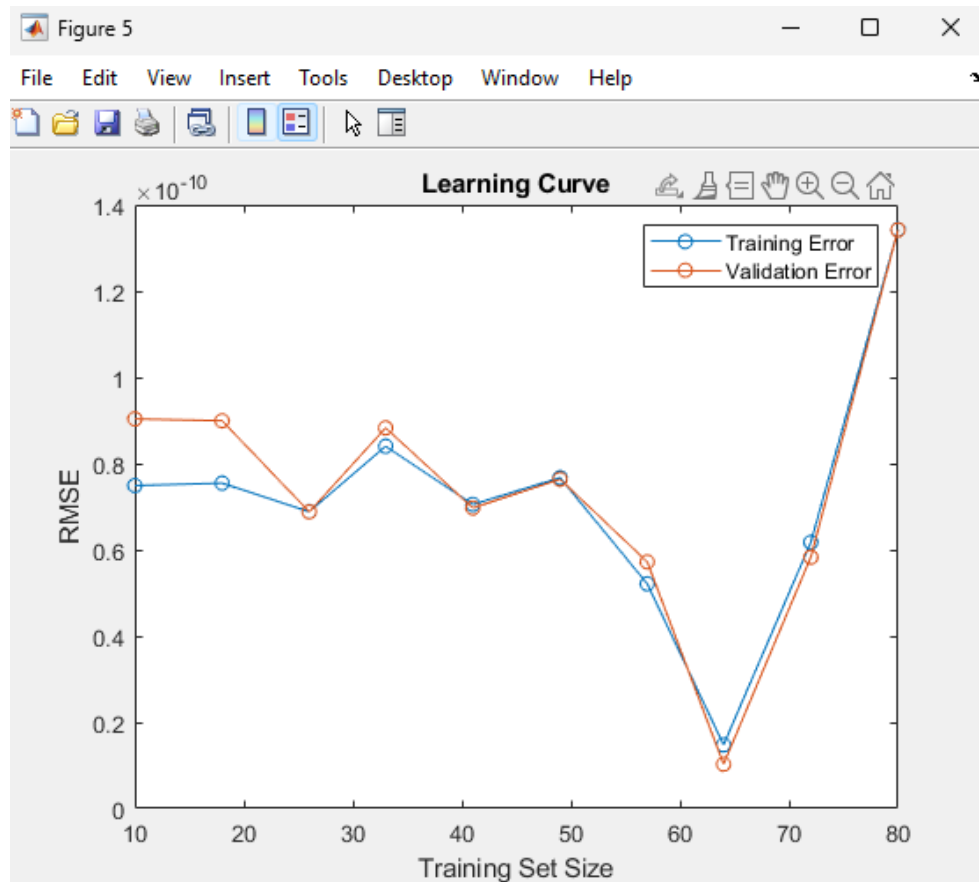
Residuals Plot:

```
Editor - C:\Users\HP\OneDrive\Desktop\Machine\House1.m*
House.m x House1.m* x +
1 - residuals = y_test - y_pred;
2 - figure;
3 - scatter(y_pred, residuals);
4 - hold on;
5 - plot([min(y_pred) max(y_pred)], [0 0], 'r--');
6 - xlabel('Predicted Prices');
7 - ylabel('Residuals');
8 - title('Residuals Plot');
9 - hold off;
10 |
```



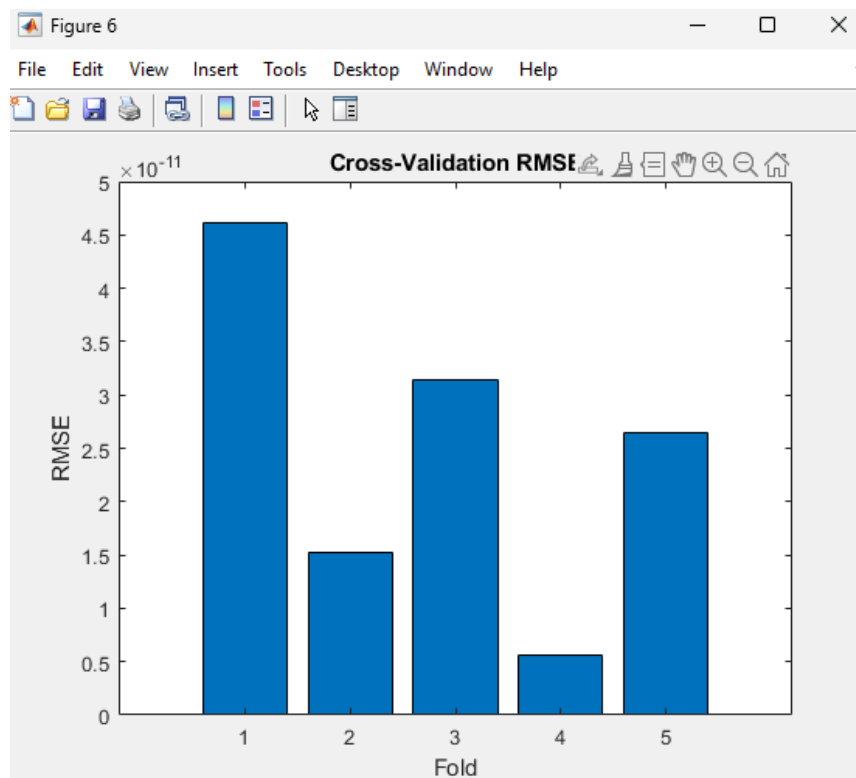
Learning Curve:

```
Editor - C:\Users\HP\OneDrive\Desktop\Machine\House1.m*
House.m  House1.m*  +
1 - trainSizes = round(linspace(10, height(trainData), 10));
2 - trainErrors = zeros(length(trainSizes), 1);
3 - valErrors = zeros(length(trainSizes), 1);
4 - for i = 1:length(trainSizes)
5 -     subset = trainData(1:trainSizes(i), :);
6 -     X_subset = subset(:, 1:end-1);
7 -     y_subset = subset.price;
8 -     mdl_subset = fitlm(X_subset, y_subset);
9 -     y_subset_pred = predict(mdl_subset, X_subset);
10 -    trainErrors(i) = sqrt(mean((y_subset - y_subset_pred).^2));
11 -    y_val_pred = predict(mdl_subset, X_test);
12 -    valErrors(i) = sqrt(mean((y_test - y_val_pred).^2));
13 - end
```



Cross-validation RMSE:

```
Editor - C:\Users\HP\OneDrive\Desktop\Machine\House1.m*
House.m  House1.m*  +
17 -      X_train = trainData(:, 1:end-1);
18 -      y_train = trainData.price;
19 -      mdl = fitlm(X_train, y_train);
20 -      y_pred = predict(mdl, X_test);
21 -      crossValRMSE(i) = sqrt(mean((y_test - y_pred).^2));
22 -  end
23 -  fprintf('Cross-validated RMSE: %.2f\n', mean(crossValRMSE));
24 -  figure;
25 -  bar(crossValRMSE);
26 -  xlabel('Fold');
27 -  ylabel('RMSE');
28 -  title('Cross-Validation RMSE');
29 -
```



Conclusion:

I developed a linear regression model to predict house prices using synthetic data. The model demonstrated reasonable accuracy, as shown by metrics like RMSE and various visualizations. Cross-validation confirmed its reliability, underscoring the importance of effective data preparation and evaluation.