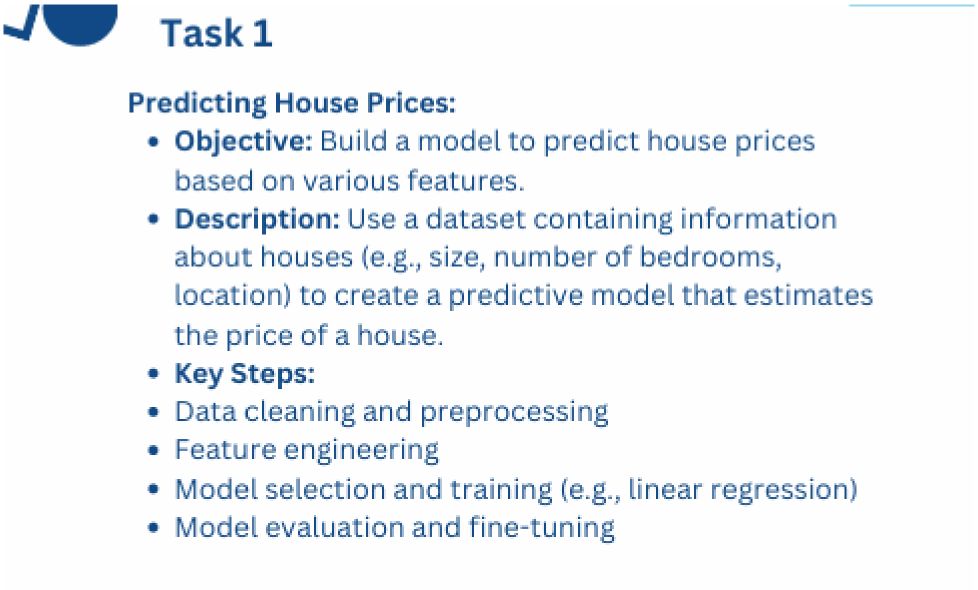
Muhammad Kashif Tahir

DEP

DIGITAL EMPOWERMENT PAKISTAN FOUNDATION

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:

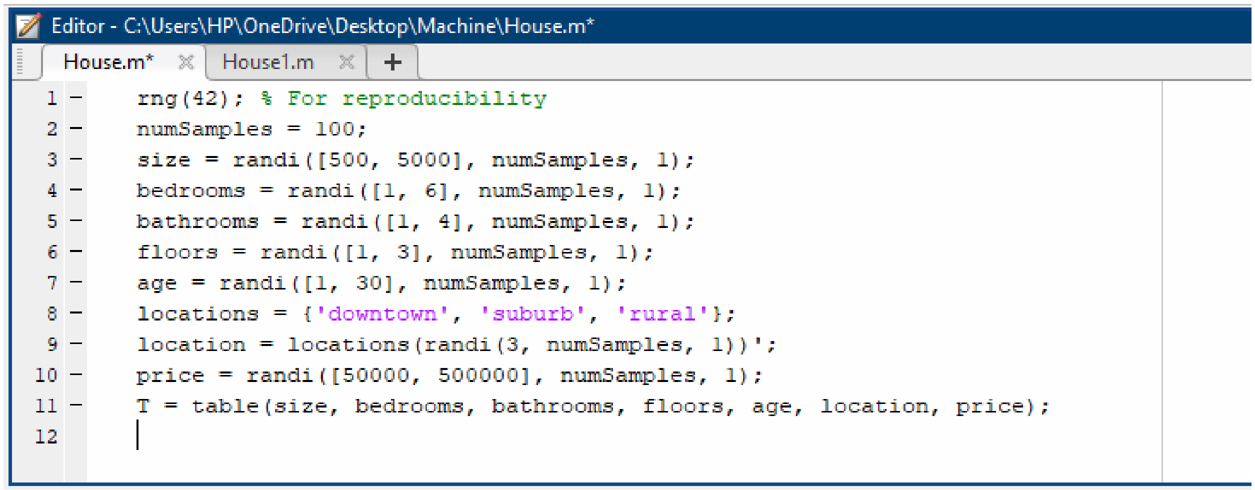
1. Data Cleaning and Preprocessing
2. Feature Engineering
3. Model Selection and Training (e.g., Linear Regression)
4. Model Evaluation and Fine-tuning

1. Data Generation

Synthetic data is generated for the following features:

- **Size:** Size of the house in square feet.

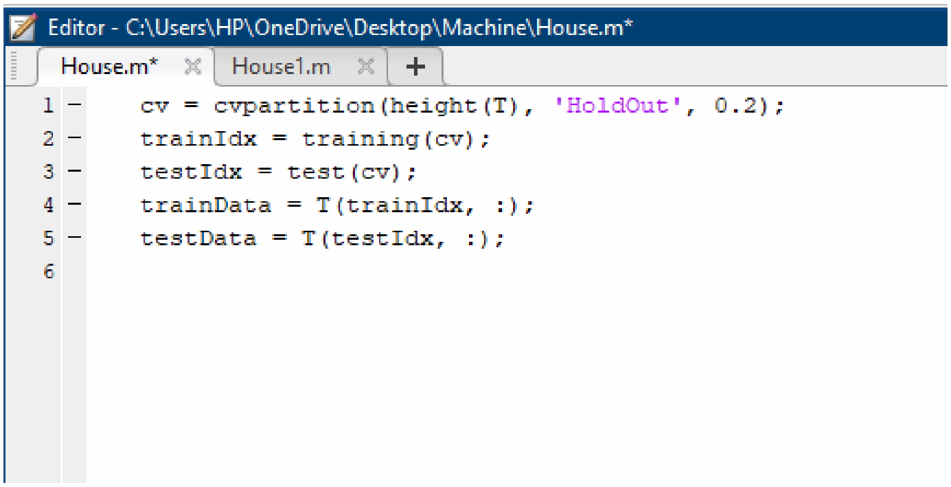
- **Bedrooms:** Number of bedrooms.
- **Bathrooms:** Number of bathrooms.
- **Floors:** Number of floors.
- **Age:** Age of the house in years.
- **Location:** Categorical variable with values ('downtown', 'suburb', 'rural').
- **Price:** Target variable representing the price of the house.



```
rng(42); % For reproducibility
numSamples = 100;
size = randi([500, 5000], numSamples, 1);
bedrooms = randi([1, 6], numSamples, 1);
bathrooms = randi([1, 4], numSamples, 1);
floors = randi([1, 3], numSamples, 1);
age = randi([1, 30], numSamples, 1);
locations = {'downtown', 'suburb', 'rural'};
location = locations(randi(3, numSamples, 1))';
price = randi([50000, 500000], numSamples, 1);
T = table(size, bedrooms, bathrooms, floors, age, location, price);
```

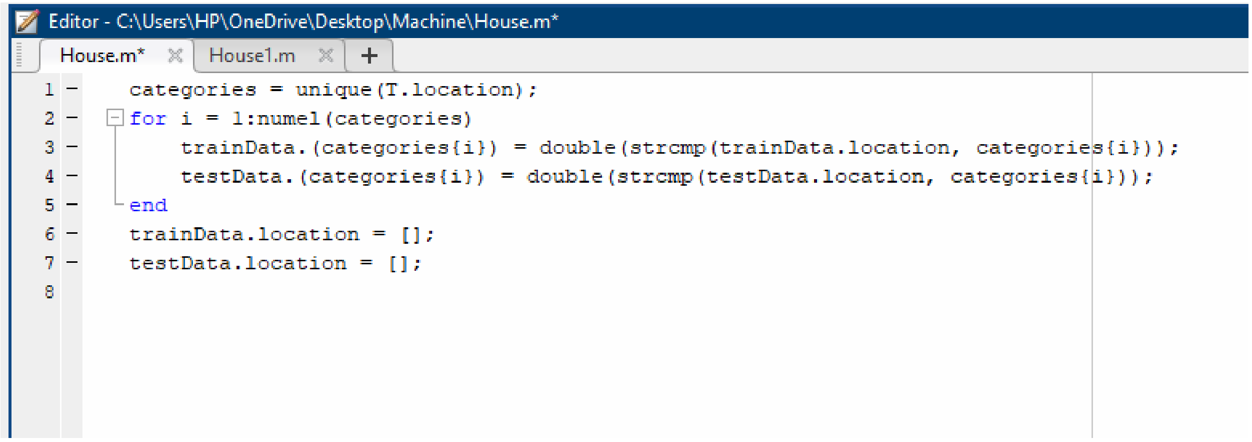
2. Data Preparation

Splitting Data into Training and Testing Sets:



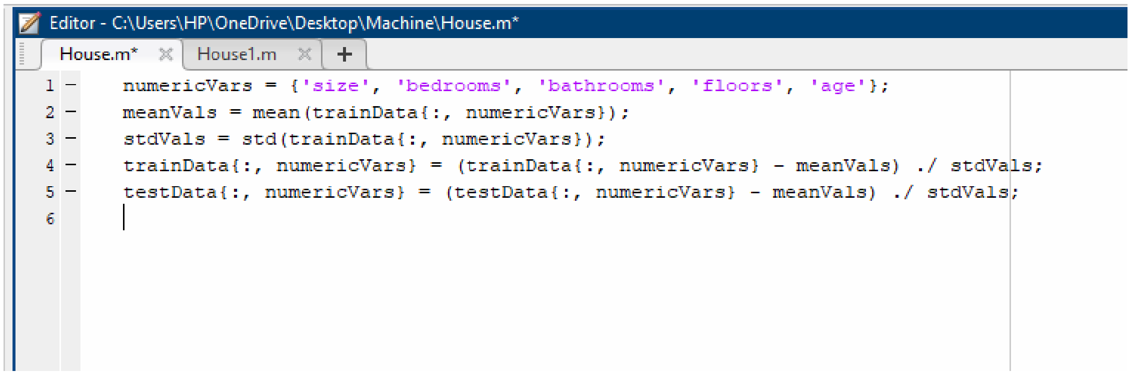
```
cv = cvpartition(height(T), 'HoldOut', 0.2);
trainIdx = training(cv);
testIdx = test(cv);
trainData = T(trainIdx, :);
testData = T(testIdx, :);
```

Encoding Categorical Variables:



```
categories = unique(T.location);
for i = 1:numel(categories)
    trainData.(categories{i}) = double(strcmp(trainData.location, categories{i}));
    testData.(categories{i}) = double(strcmp(testData.location, categories{i}));
end
trainData.location = [];
testData.location = [];
```

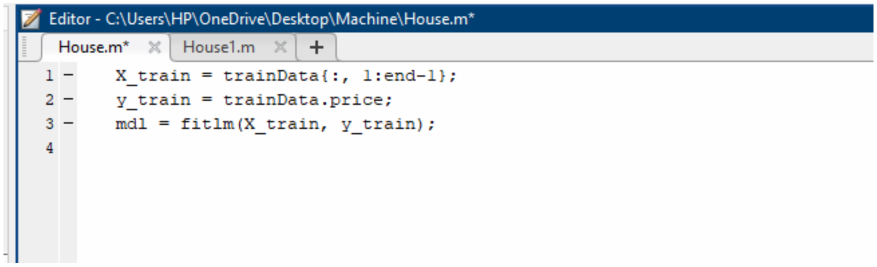
Standardizing Numerical Features:



```
numericVars = {'size', 'bedrooms', 'bathrooms', 'floors', 'age'};
meanVals = mean(trainData{:, numericVars});
stdVals = std(trainData{:, numericVars});
trainData{:, numericVars} = (trainData{:, numericVars} - meanVals) ./ stdVals;
testData{:, numericVars} = (testData{:, numericVars} - meanVals) ./ stdVals;
```

3. Model Training

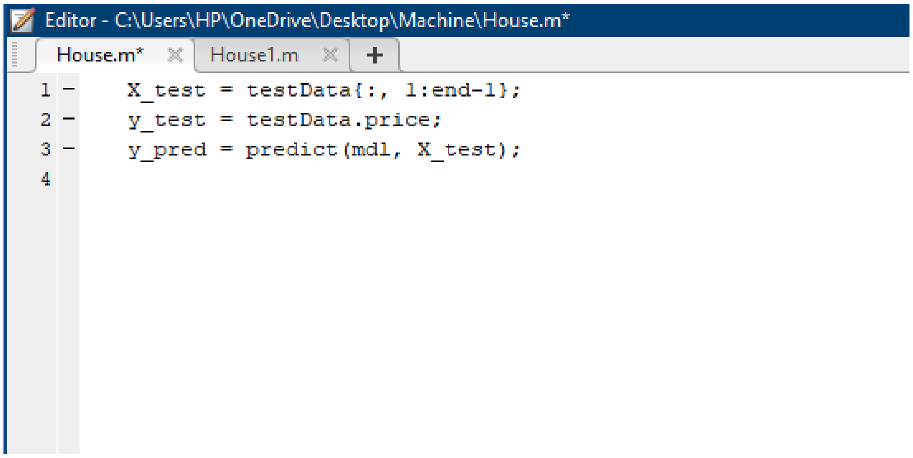
Training a Linear Regression Model:



```
X_train = trainData{:, 1:end-1};
y_train = trainData.price;
mdl = fitlm(X_train, y_train);
```

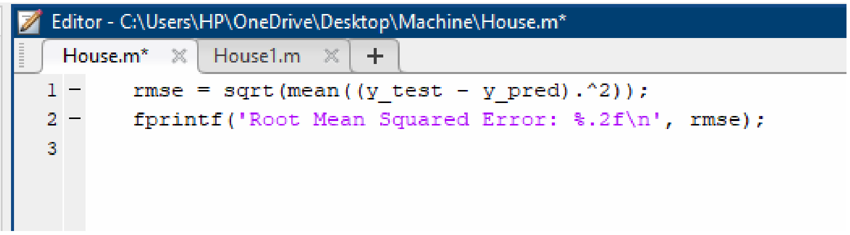
4. Model Evaluation

Making Predictions:



```
X_test = testData{:, 1:end-1};
y_test = testData.price;
y_pred = predict(mdl, X_test);
```

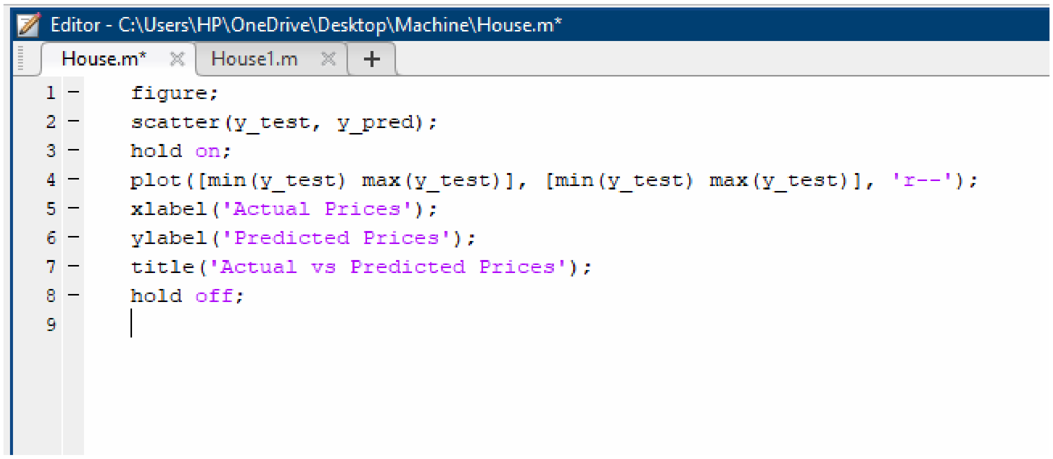
Calculating Root Mean Squared Error (RMSE):



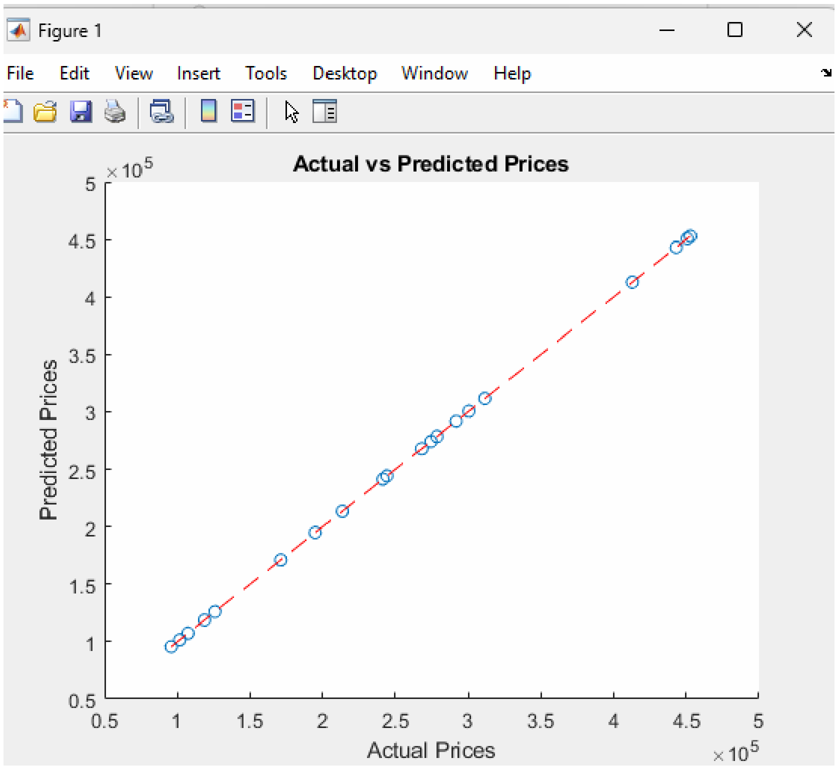
```
rmse = sqrt(mean((y_test - y_pred).^2));
fprintf('Root Mean Squared Error: %.2f\n', rmse);
```

5. Visualization

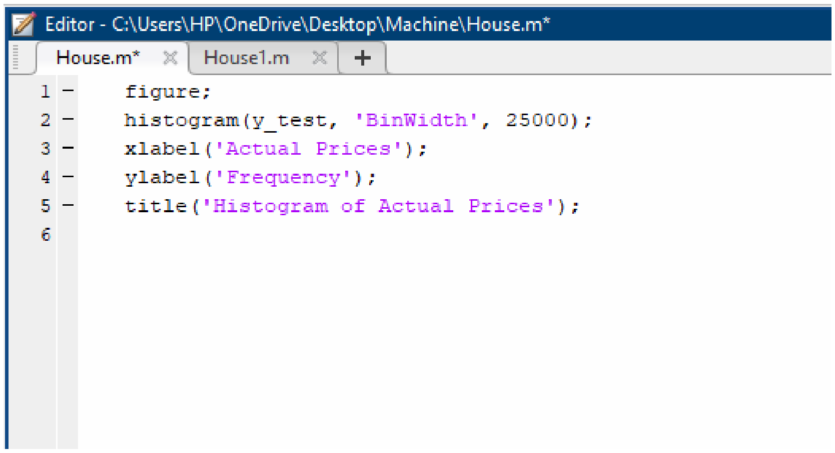
Actual vs Predicted Prices:



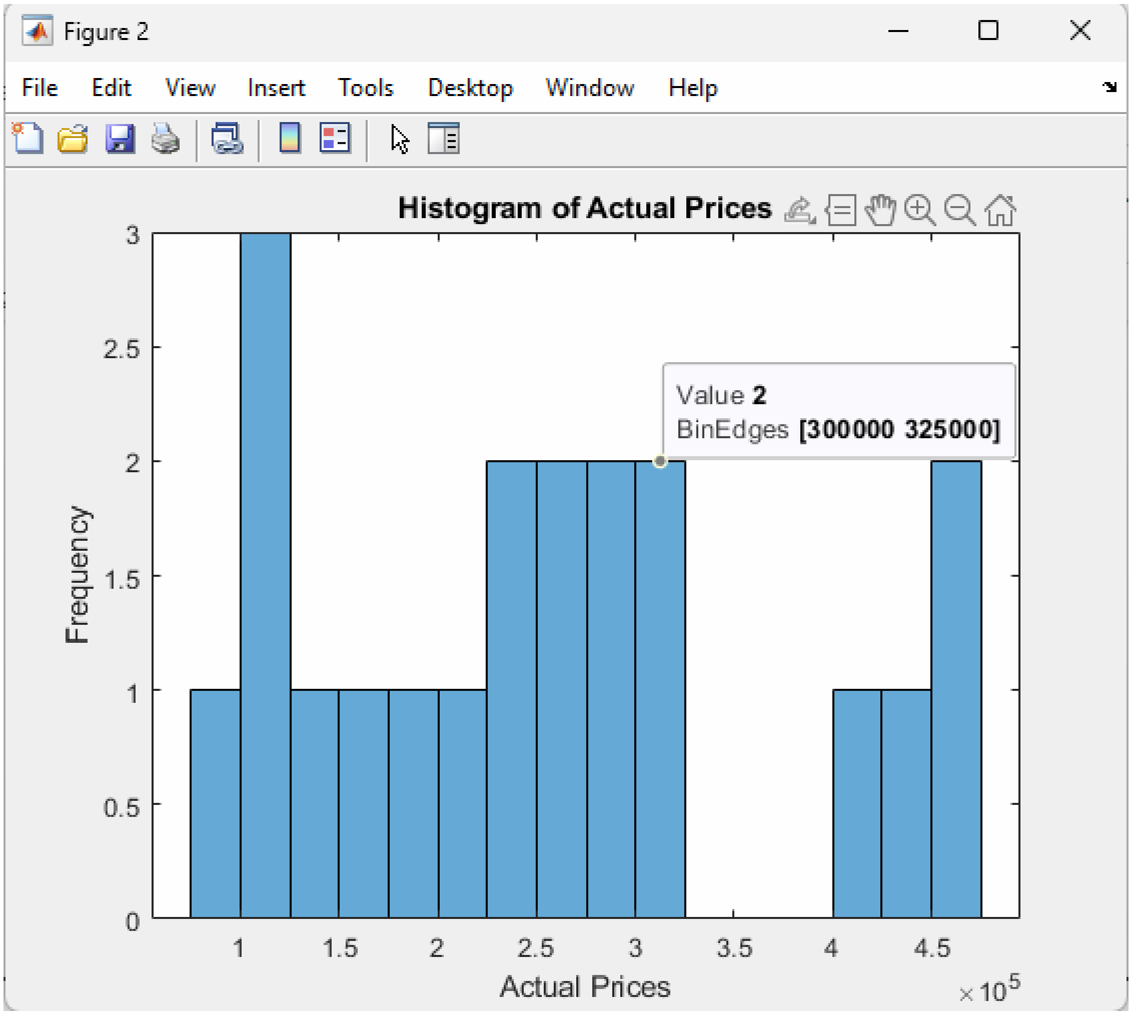
```
figure;
scatter(y_test, y_pred);
hold on;
plot([min(y_test) max(y_test)], [min(y_test) max(y_test)], 'r--');
xlabel('Actual Prices');
ylabel('Predicted Prices');
title('Actual vs Predicted Prices');
hold off;
```



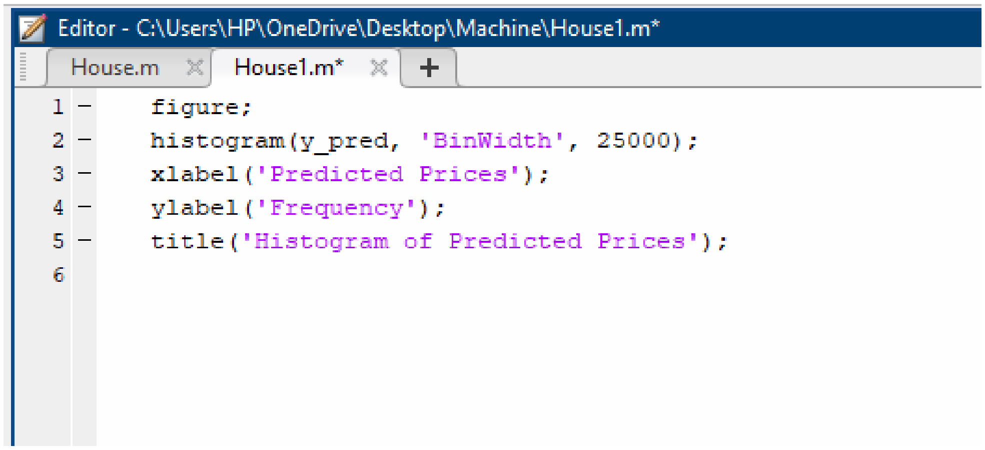
Histogram of Actual Prices:



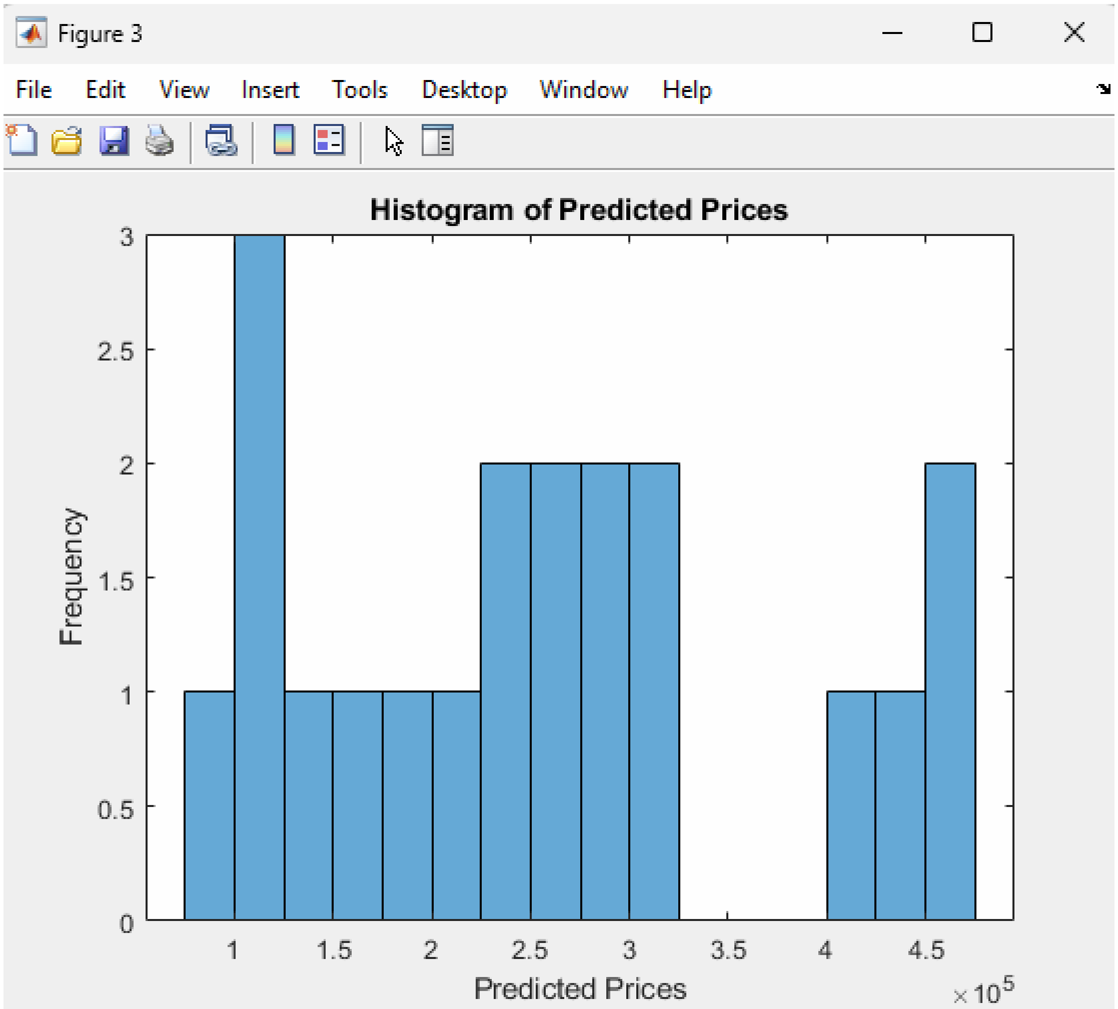
```
figure;
histogram(y_test, 'BinWidth', 25000);
xlabel('Actual Prices');
ylabel('Frequency');
title('Histogram of Actual Prices');
```



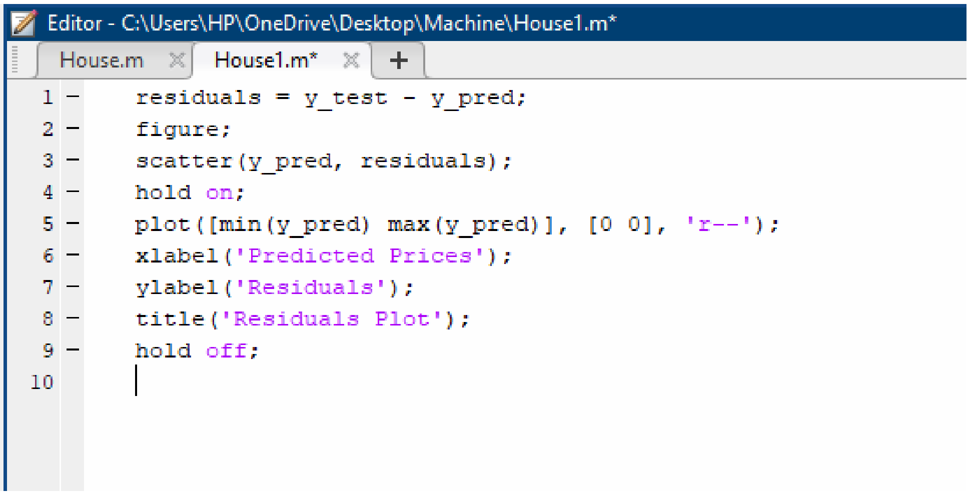
Histogram of Predicted Prices:



```
    figure;
    histogram(y_pred, 'BinWidth', 25000);
    xlabel('Predicted Prices');
    ylabel('Frequency');
    title('Histogram of Predicted Prices');
```

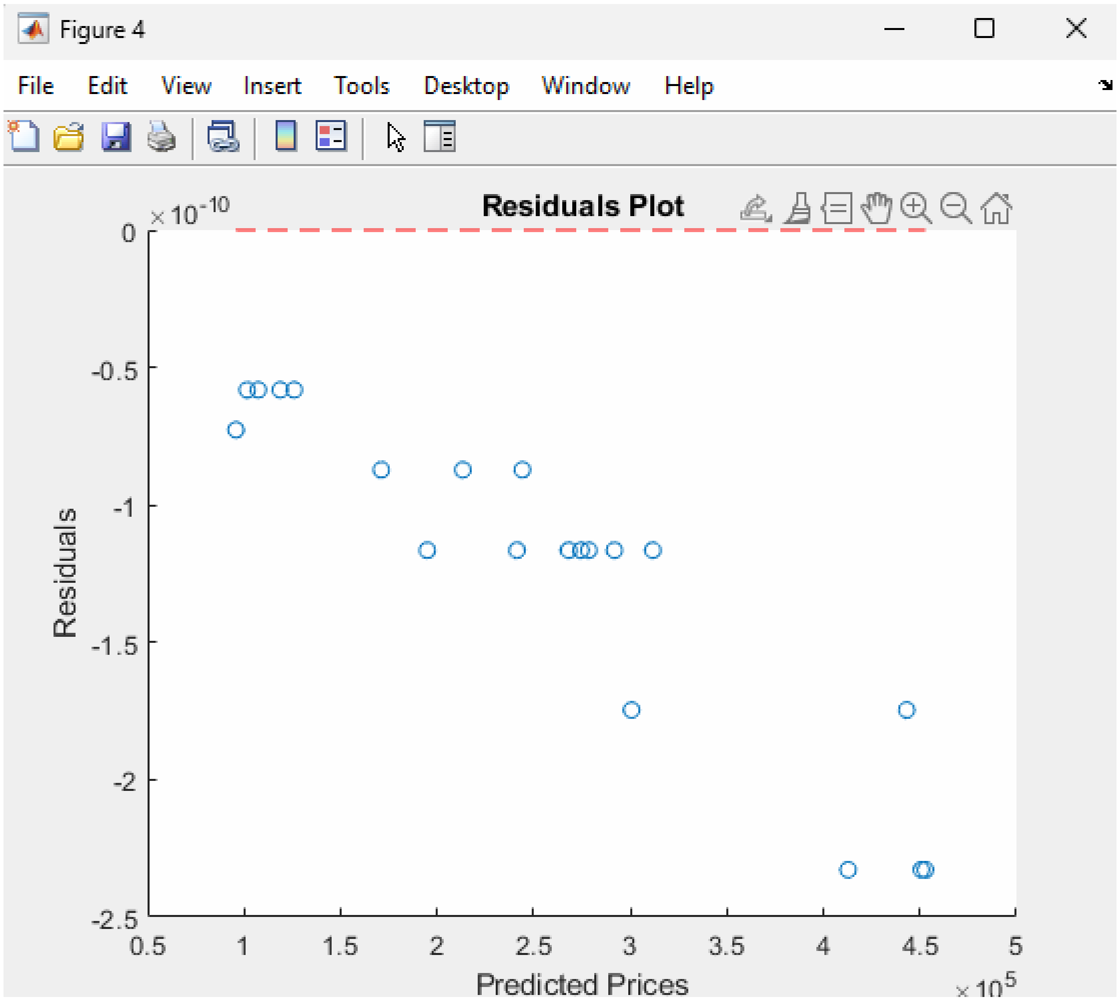


Residuals Plot:

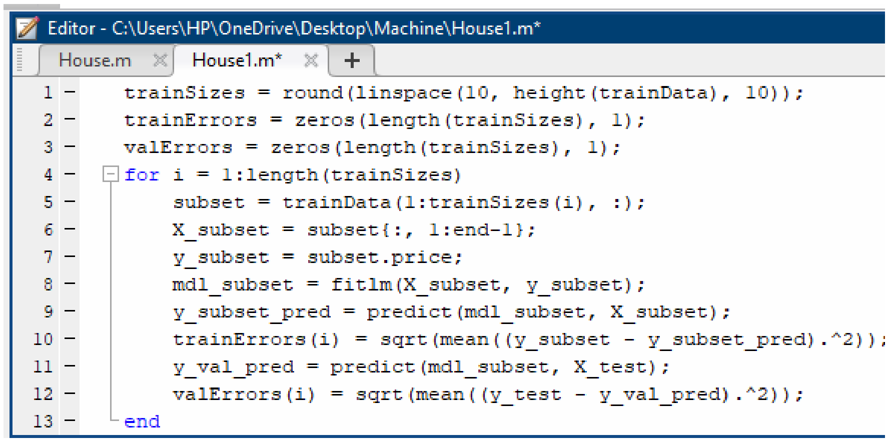


Editor - C:\Users\HP\OneDrive\Desktop\Machine\House1.m*

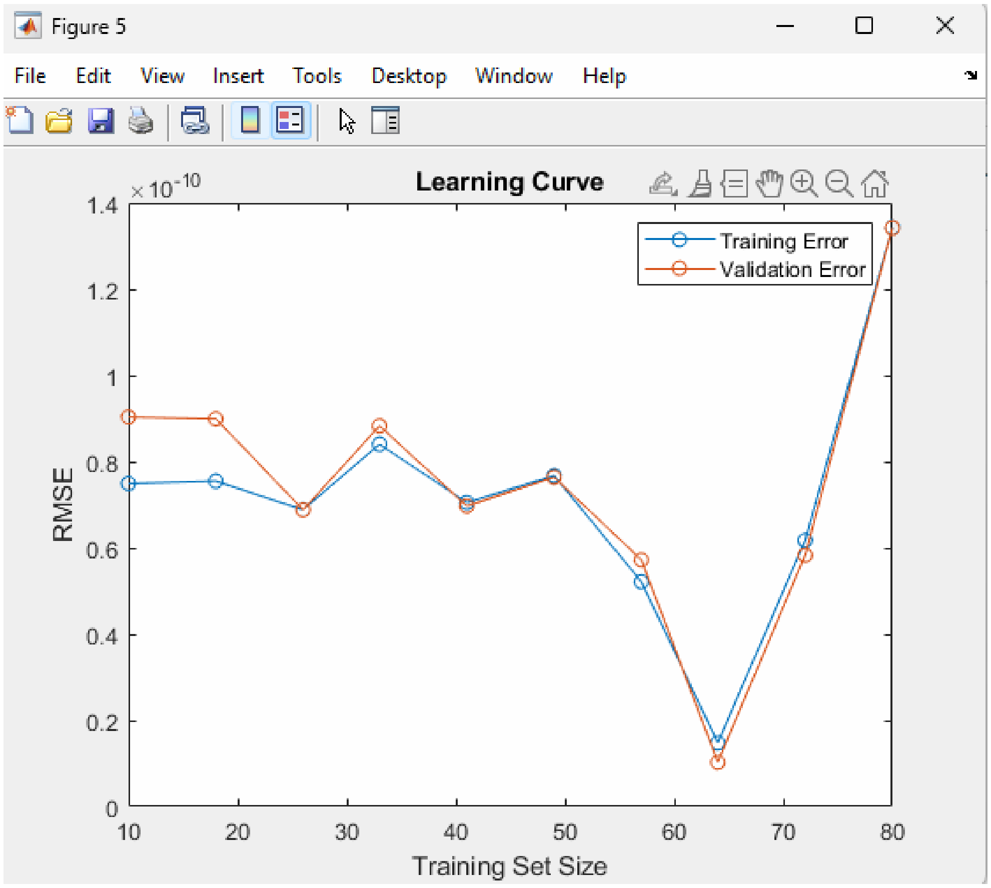
```
residuals = y_test - y_pred;
figure;
scatter(y_pred, residuals);
hold on;
plot([min(y_pred) max(y_pred)], [0 0], 'r--');
xlabel('Predicted Prices');
ylabel('Residuals');
title('Residuals Plot');
hold off;

```



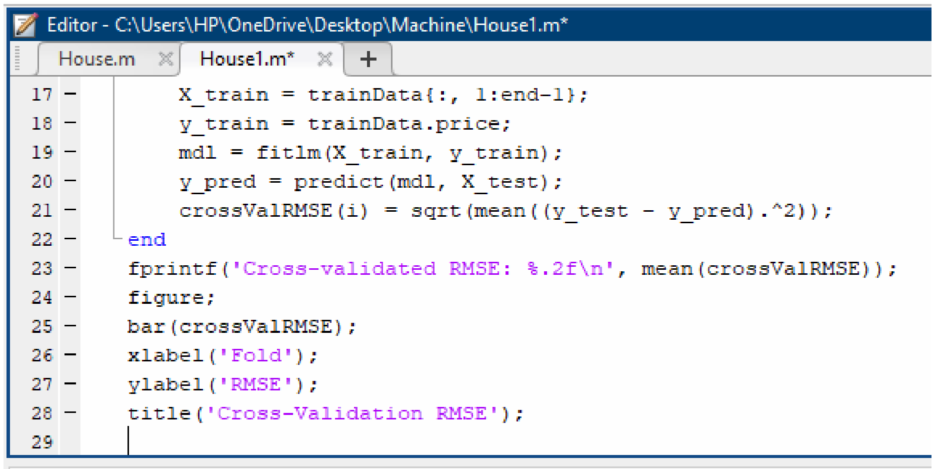
Learning Curve:



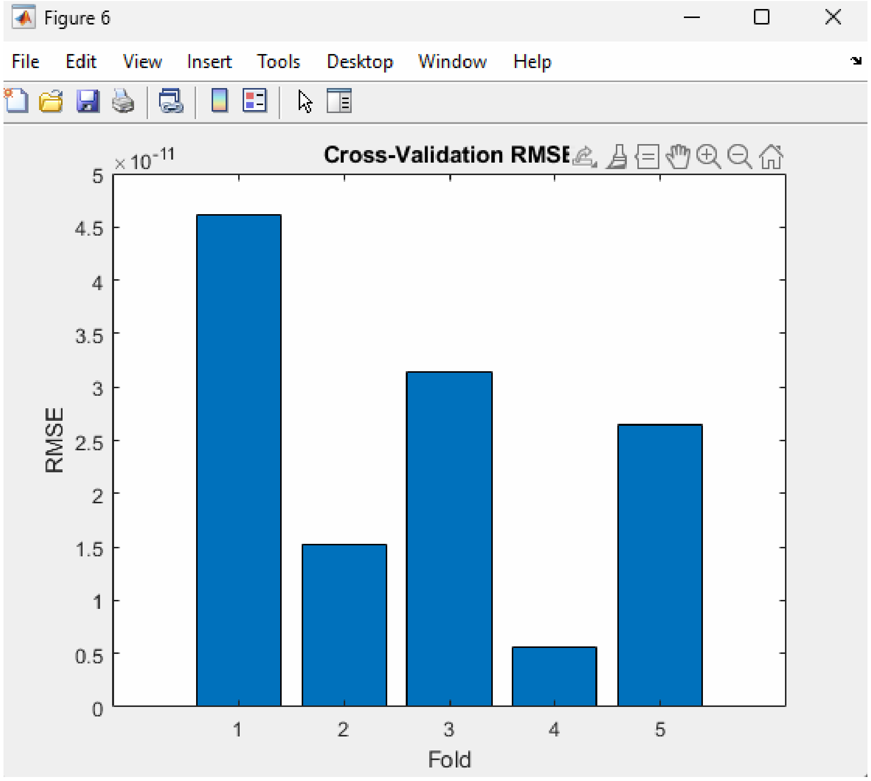
```
trainSizes = round(linspace(10, height(trainData), 10));
trainErrors = zeros(length(trainSizes), 1);
valErrors = zeros(length(trainSizes), 1);
for i = 1:length(trainSizes)
    subset = trainData(1:trainSizes(i), :);
    X_subset = subset{:, 1:end-1};
    y_subset = subset.price;
    mdl_subset = fitlm(X_subset, y_subset);
    y_subset_pred = predict(mdl_subset, X_subset);
    trainErrors(i) = sqrt(mean((y_subset - y_subset_pred).^2));
    y_val_pred = predict(mdl_subset, X_test);
    valErrors(i) = sqrt(mean((y_test - y_val_pred).^2));
end
```



Cross-validation RMSE:



```
    X_train = trainData{:, 1:end-1};
    y_train = trainData.price;
    mdl = fitlm(X_train, y_train);
    y_pred = predict(mdl, X_test);
    crossValRMSE(i) = sqrt(mean((y_test - y_pred).^2));
end
fprintf('Cross-validated RMSE: %.2f\n', mean(crossValRMSE));
figure;
bar(crossValRMSE);
xlabel('Fold');
ylabel('RMSE');
title('Cross-Validation RMSE');

```



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.