House Price Prediction Model

Overview

This project implements a machine learning model to predict house prices using linear regression. The model considers various factors like house size, number of rooms, location rating, and age to estimate property values.

Project Description

As a first-year student of JIIT exploring machine learning, I built this house price prediction system to understand how different features affect property values. The project uses synthetic data to simulate real-world housing market conditions and applies linear regression for price prediction.

Features

- **Data Generation**: Creates synthetic house data with realistic features
- **Linear Regression**: Uses scikit-learn's LinearRegression model
- **Feature Analysis**: Shows how each feature impacts house prices
- **Model Evaluation**: Includes MSE, RMSE, and R-squared metrics
- **Example Predictions**: Tests the model with sample houses

Technologies Used

- Python 3.x
- pandas - for data manipulation
- numpy - for numerical operations
- scikit-learn - for machine learning algorithms
- matplotlib - for potential visualizations

Dataset Features

The synthetic dataset includes:

- **Size**: House area in square feet

- **Rooms**: Number of rooms (2-5)
- **Location**: Location rating (1-10 scale)
- **Age**: House age in years
- **Price**: Target variable (house price in dollars)

Setup and Installation

How to Run

1. **Navigate to the project directory**

   cd house-price-prediction

2. **Run the main script**

python house_price_prediction.py

3. **Output**

   The program will display:

- Dataset statistics
- Model training progress
- Evaluation metrics
- Example predictions
- Feature coefficients

Expected Output

When you run the program, you should see output similar to: House Price Prediction Model

==========

Creating data for 1000 houses...

Sample of our data:

	size	rooms	location	age	price
0	1819.65	4	8.44	23	234567.89

MADE BY: KARTIK VIRMANI, JIIT NOIDA

1 1234.56 3 5.67 15 178901.23

...

FLOW OF THE WHOLE PROGRAMME :

1. **Data Generation**
   The program starts by creating a synthetic dataset with 1,000 samples. Each house in the dataset has features such as size (in square feet), number of rooms, location rating, and age. The target variable, house price, is simulated using a formula that weights these features with added random noise to mimic real-world variability.

2. **Data Preparation**
   The dataset is organized into features (input variables) and the target variable (price). The features include numerical values like house size and age. The data is then split into training and testing sets (80% training, 20% testing) using stratified sampling to maintain representative distributions.

3. **Model Training**
   A Linear Regression model from scikit-learn is created and trained using the training data. The model learns the relationship between house features and prices by minimizing the prediction error.

4. **Prediction**
   After training, the model predicts the prices of houses in the test set using the learned coefficients.

5. **Evaluation**
   The program computes evaluation metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared to quantify the model's accuracy. These metrics indicate how close the predicted prices are to the actual prices.

6. **Model Interpretation**
   The coefficients of the linear regression model are displayed. Each coefficient represents how much the house price increases or decreases with a unit change in a feature, providing insights into feature importance.

7. **Testing with Examples**
   The program includes examples of specific houses with known features for which it outputs predicted prices. This step helps verify that the model's predictions are reasonable.

8. **Results Summary**
   A comparison between actual and predicted prices for samples in the test set is printed, displaying prediction errors and helping visualize model performance

   Training set: 800 houses
   Testing set: 200 houses

   Model Results:
   Mean Squared Error: $123,456,789.00

Root Mean Squared Error: $11,111.11

R-squared score: 0.8500

Project Structure

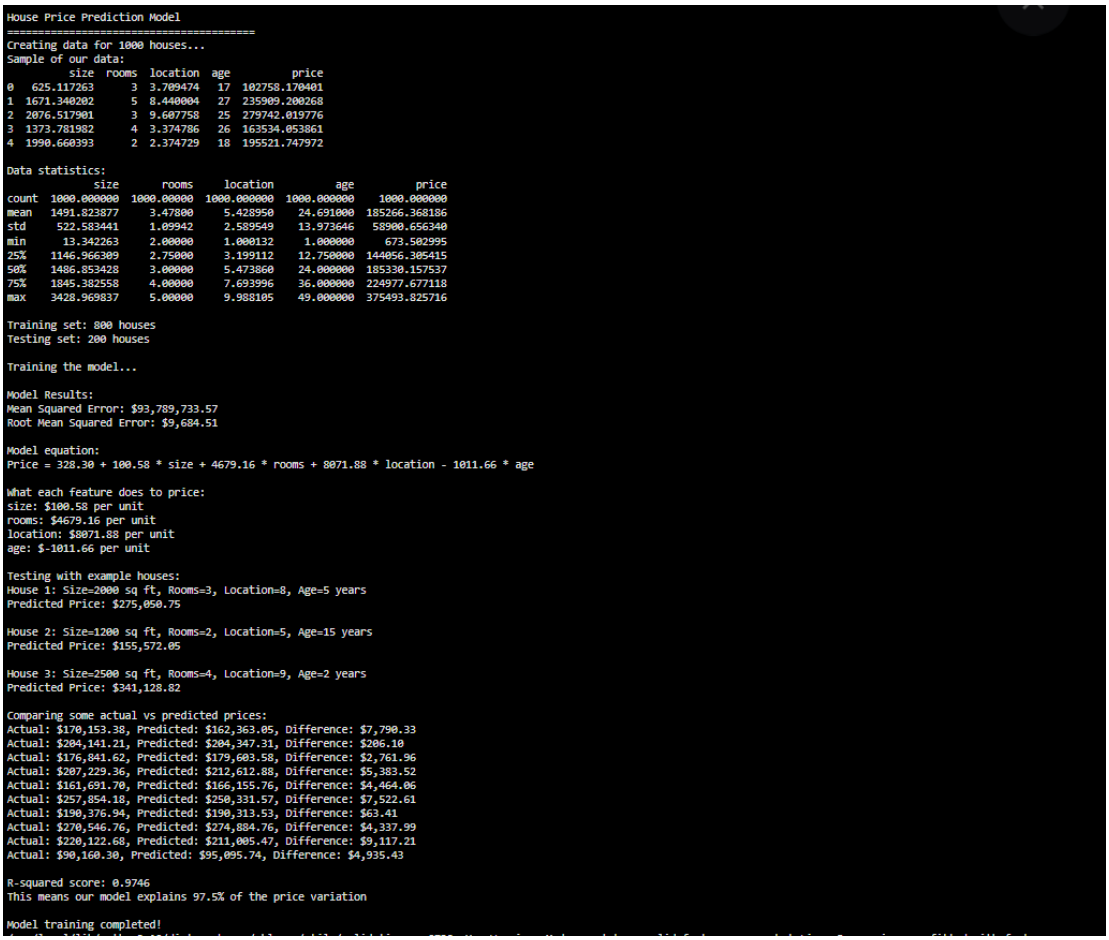
```
house-price-prediction/
│
├── house_price_prediction.py    # Main Python script
├── README.md                    # Project documentation
├── screenshots/                 # Output screenshots
│   ├── output_example.png
│   └── model_results.png
└── requirements.txt             # Python dependencies
```

Example Use Cases

This model can be used to:

- Estimate house prices for real estate analysis
- Understand which factors most influence property values
- Practice machine learning fundamentals
- Learn about linear regression implementation

Screenshots of the output



```
House Price Prediction Model
==================================
Creating data for 1000 houses...
Sample of our data:
         size  rooms  location  age         price
0   625.117263      3  3.709474   17  102758.170401
1  1671.340202      5  8.440004   27  235909.200268
2  2076.517901      3  9.607758   25  279742.019776
3  1373.781982      4  3.374786   26  163534.053861
4  1990.660393      2  2.374729   18  195521.747972

Data statistics:
              size       rooms     location          age          price
count  1000.000000  1000.00000  1000.000000  1000.000000    1000.000000
mean   1491.823877     3.47800     5.428950    24.691000  185266.368186
std     522.583441     1.09942     2.589549    13.973646   58900.656340
min      13.342263     2.00000     1.000132     1.000000     673.502995
25%    1146.966309     2.75000     3.199112    12.750000  144056.305415
50%    1486.853428     3.00000     5.473860    24.000000  185330.157537
75%    1845.382558     4.00000     7.693996    36.000000  224977.677118
max    3428.969837     5.00000     9.988105    49.000000  375493.825716

Training set: 800 houses
Testing set: 200 houses

Training the model...

Model Results:
Mean Squared Error: $93,789,733.57
Root Mean Squared Error: $9,684.51

Model equation:
Price = 328.30 + 100.58 * size + 4679.16 * rooms + 8071.88 * location - 1011.66 * age

What each feature does to price:
size: $100.58 per unit
rooms: $4679.16 per unit
location: $8071.88 per unit
age: $-1011.66 per unit

Testing with example houses:
House 1: Size=2000 sq ft, Rooms=3, Location=8, Age=5 years
Predicted Price: $275,050.75

House 2: Size=1200 sq ft, Rooms=2, Location=5, Age=15 years
Predicted Price: $155,572.05

House 3: Size=2500 sq ft, Rooms=4, Location=9, Age=2 years
Predicted Price: $341,128.82

Comparing some actual vs predicted prices:
Actual: $170,153.38, Predicted: $162,363.05, Difference: $7,790.33
Actual: $204,141.21, Predicted: $204,347.31, Difference: $206.10
Actual: $176,841.62, Predicted: $179,603.58, Difference: $2,761.96
Actual: $207,229.36, Predicted: $212,612.88, Difference: $5,383.52
Actual: $161,691.70, Predicted: $166,155.76, Difference: $4,464.06
Actual: $257,854.18, Predicted: $250,331.57, Difference: $7,522.61
Actual: $190,376.94, Predicted: $190,313.53, Difference: $63.41
Actual: $270,546.76, Predicted: $274,884.76, Difference: $4,337.99
Actual: $220,122.68, Predicted: $211,005.47, Difference: $9,117.21
Actual: $90,160.30, Predicted: $95,095.74, Difference: $4,935.43

R-squared score: 0.9746
This means our model explains 97.5% of the price variation

Model training completed!
```

Future Improvements

Potential enhancements for this project:

- Add more features (neighborhood, amenities, etc.)
- Implement data visualization with plots
- Try other regression algorithms (polynomial, ridge regression)
- Use real housing dataset from Kaggle
- Add cross-validation for better model evaluation
- Create a simple web interface for predictions

MADE BY: KARTIK VIRMANI, JIIT NOIDA

Learning Outcomes

Through this project, I learned:

- How to generate synthetic datasets for ML projects
- Linear regression implementation using scikit-learn
- Model evaluation techniques and metrics
- The importance of train-test split
- How different features affect target variables

Author

Kartik Virmani

JIIT Noida

Email: 992501230039@mail.jiit.ac.in