HOUSE PRICE PREDICTION

A PROJECT DOCUMENTATION

Submitted By

STUDENT NAME	REGISTER NUMBER
S.CHRISTOFOR JOHN	422721104011
V.GOKUL	422121104013
S.SARATHI	422721104045
K.VARUN	422721104055

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

V.R.S COLLEGE OF ENGINEERING AND TECHNOLOGY

ARASUR-607107, VILLUPURAM DISTRICT

Table of Contents

****

Chapter 1: Abstract

1.1 Project Summary

- The project involves the development of a house price prediction model using machine learning techniques, with a focus on the XGBoost regressor. The model aims to predict house prices based on various features.
- Note: Our complete code file is available at github repository .

1.2 Objectives

- The primary objective of this project is to build an accurate machine learning model for house price prediction, using a real-world dataset.

1.3 Keywords

- House Price Prediction, Machine Learning, XGBoost Regressor, Real Estate, Dataset.
- Note: our code is at our github repository so vist the below link to view there is a readme file for our project you can run our code by that instructions.
- **Code link:** https://github.com/CHRISTOFORJOHN7/IBM_Project.git
- There you can see the file named AI_Phase5.ipynb that is our code file.
- You can run our code by following instruction the below at the readme.md

Chapter 2: Introduction

2.1 Project Overview

- This chapter provides an overview of the project, highlighting the significance of house price prediction and the machine learning model's relevance.

2.2 Purpose of Documentation

- The documentation aims to provide a comprehensive guide to the project's development, from data preprocessing to model evaluation.

2.3 Project Contributors

- We acknowledge the contributions of team members who have actively participated in the project's development.

Chapter 3: Project Description

3.1 Goals and Objectives

- This section outlines the specific goals and objectives of the project, emphasizing the target to build an efficient house price prediction model.

3.2 Background

- Providing the background and context of the project, highlighting the importance of accurate house price predictions.

3.3 Dataset Description

- Details regarding the dataset used, including the data sources and key features.

Dataset Name: USA Housing

Source: Kaggle.

Dataset Link:

https://www.kaggle.com/datasets/vedavyasv/usa-housing

Provided by your organization.

Description:

The USA Housing dataset is a collection of data related to housing properties in the United States. It contains information about various features. The dataset contains 5000 rows and 7 columns.

3.4 Tools and Libraries Used

- An overview of the tools and libraries employed in the project, including Python, XGBoost, and data visualization tools.

Pandas: This library is used for data manipulation and analysis.

Numpy: It's used for numerical and mathematical operations on data.

Scikit-learn (sklearn): It's a machine learning library used for data preprocessing, model selection, and evaluation.

XGboost: A library for gradient boosting that is commonly used for regression tasks like house price prediction.

Matplotlib: Used for data visualization, particularly for creating plots and graphs.

Seaborn: Another data visualization library that is built on top of matplotlib, providing additional features for data visualization.

Jupyter notebook: A web-based interactive computing environment that allows you to create and share documents that contain live code, equations, visualizations, and narrative text.

In python we used the packages and libraries like ,

```
# Import necessary libraries and modules
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from xgboost import XGBRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
```

Chapter 4: Data Preprocessing

4.1 Loading the Dataset

Describing the process of loading the dataset into the project environment.

```
# Load the dataset
data = pd.read_csv("USA_Housing.csv")
```

4.2 Data Cleaning

Discussing the steps taken to clean the data, including handling missing values and duplicates.

```
# Data Cleaning

# Drop rows with missing values (NaN)

data.dropna(inplace=True)


# Drop duplicate rows

data drop_duplicates(inplace=True)
```

4.3 Data Exploration

Exploring the dataset to understand data distributions, identifying numerical and categorical features.

```
# Data Visualization

import seaborn as sns

sns.pairplot(data)

plt.show()
```

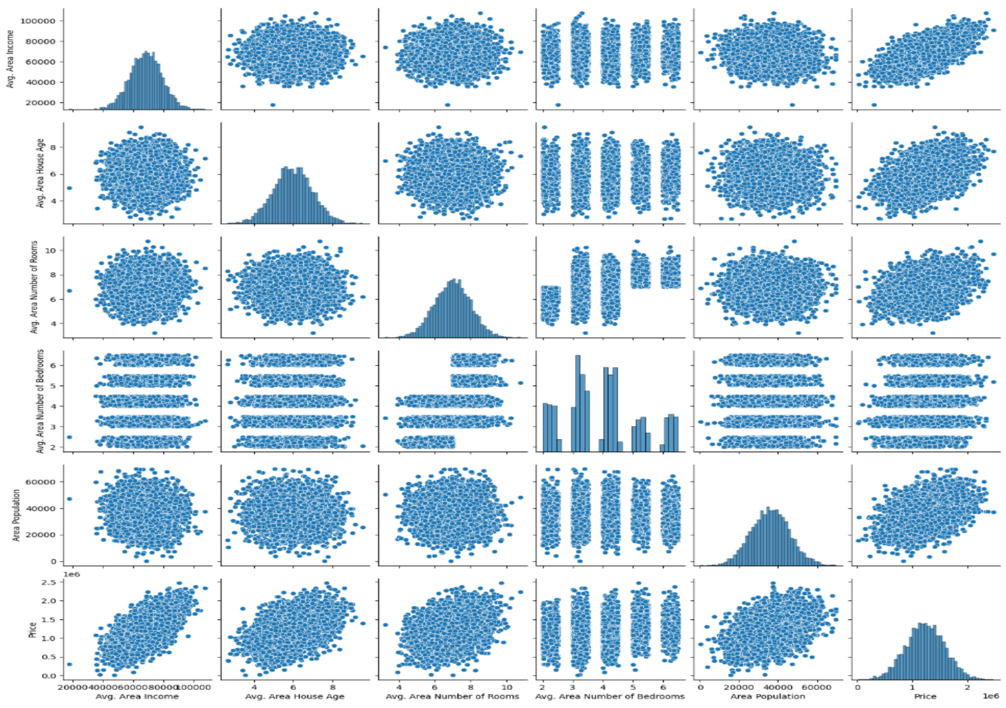
4.4 Data Visualization

- Visualizing the data through pair plots and correlation heatmaps to gain insights into data relationships.

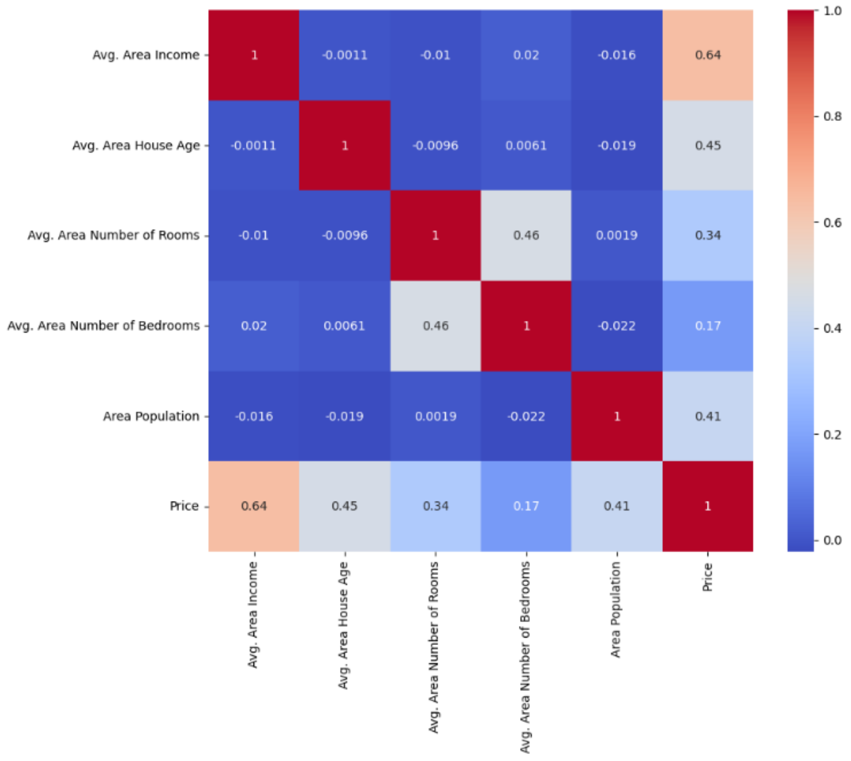
```
# Heatmap

numeric_data = data.select_dtypes(include=[np.number])

correlation_matrix = numeric_data.corr()

plt.figure(figsize=(10, 8))

sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm")

plt.show()
```

4.5 Output

- Displaying the preliminary output of data preprocessing for reference.



```
In [5]: # Heatmap
        numeric_data = data.select_dtypes(include=[np.number])
        correlation_matrix = numeric_data.corr()
        plt.figure(figsize=(10, 8))
        sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm")
        plt.show()
```

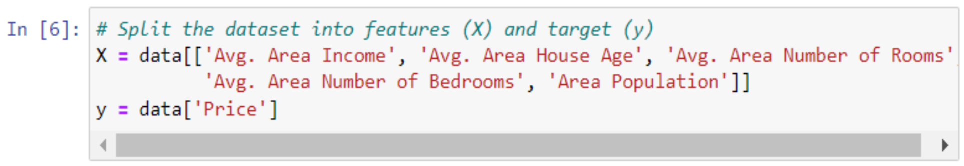


	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
Avg. Area Income	1	-0.0011	-0.01	0.02	-0.016	0.64
Avg. Area House Age	-0.0011	1	-0.0096	0.0061	-0.019	0.45
Avg. Area Number of Rooms	-0.01	-0.0096	1	0.46	0.0019	0.34
Avg. Area Number of Bedrooms	0.02	0.0061	0.46	1	-0.022	0.17
Area Population	-0.016	-0.019	0.0019	-0.022	1	0.41
Price	0.64	0.45	0.34	0.17	0.41	1

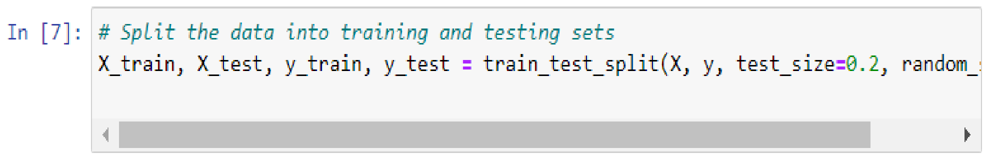
Chapter 5: Model Development

5.1 Data Splitting

Detailing the process of splitting the dataset into training and testing sets for model development.



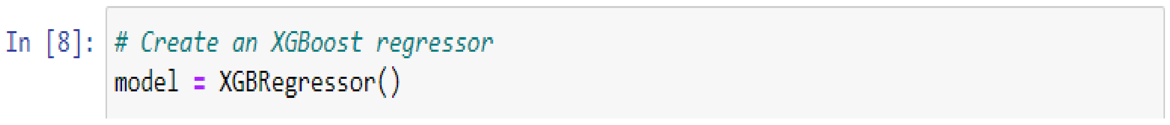
```
In [6]: # Split the dataset into features (X) and target (y)
        X = data[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms'
                  'Avg. Area Number of Bedrooms', 'Area Population']]
        y = data['Price']
```



```
In [7]: # Split the data into training and testing sets
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
```

5.2 Model Selection

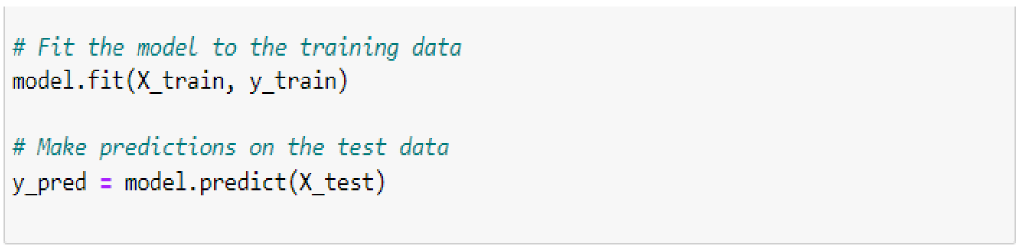
- Explaining the choice of the XGBoost regressor as the machine learning model for house price prediction.



```
In [8]: # Create an XGBoost regressor
        model = XGBRegressor()
```

5.3 Model Training

- Covering the steps taken to train the selected model using the training dataset.



```
# Fit the model to the training data
model.fit(X_train, y_train)

# Make predictions on the test data
y_pred = model.predict(X_test)
```

5.4 Model Evaluation Metrics

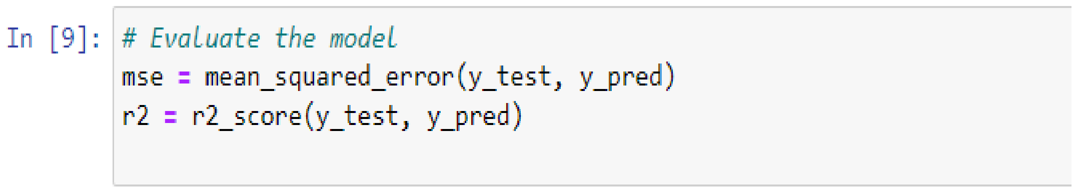
- Assessing model performance through metrics such as Mean Squared Error (MSE) and R-squared (R2).

5.4.1 Mean Squared Error:

- We evaluate the model using the Mean Squared Error (MSE) metric, which measures the average squared differences between actual and predicted prices.

5.4.2 R-squared:

- The R-squared value quantifies the model's performance, indicating how well it fits the data. A higher R-squared value suggests better predictions.



```
In [9]: # Evaluate the model
        mse = mean_squared_error(y_test, y_pred)
        r2 = r2_score(y_test, y_pred)
```

5.5 Model Visualization

- Visualizing model predictions through scatter plots and regression lines.

5.6 Model Accuracy

- Evaluating and discussing the accuracy of the model.

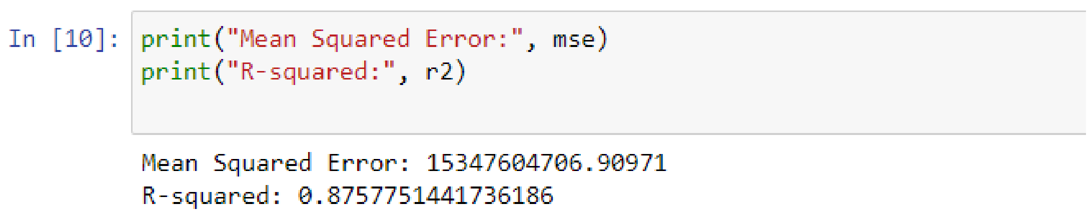
5.7 Code

- Providing code snippets related to model development and training.

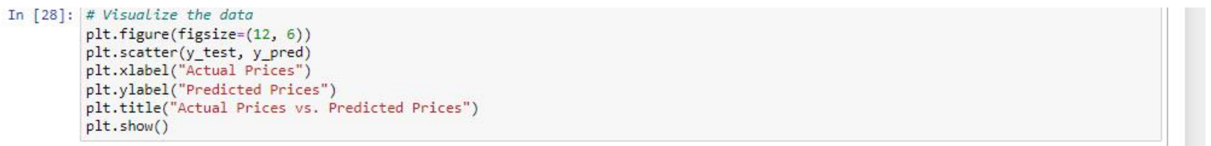
```
# Create an XGBoost regressor
model = XGBRegressor()
# Fit the model to the training data
model.fit(X_train, y_train)
# Make predictions on the test data
y_pred = model.predict(X_test)
```

5.8 Output

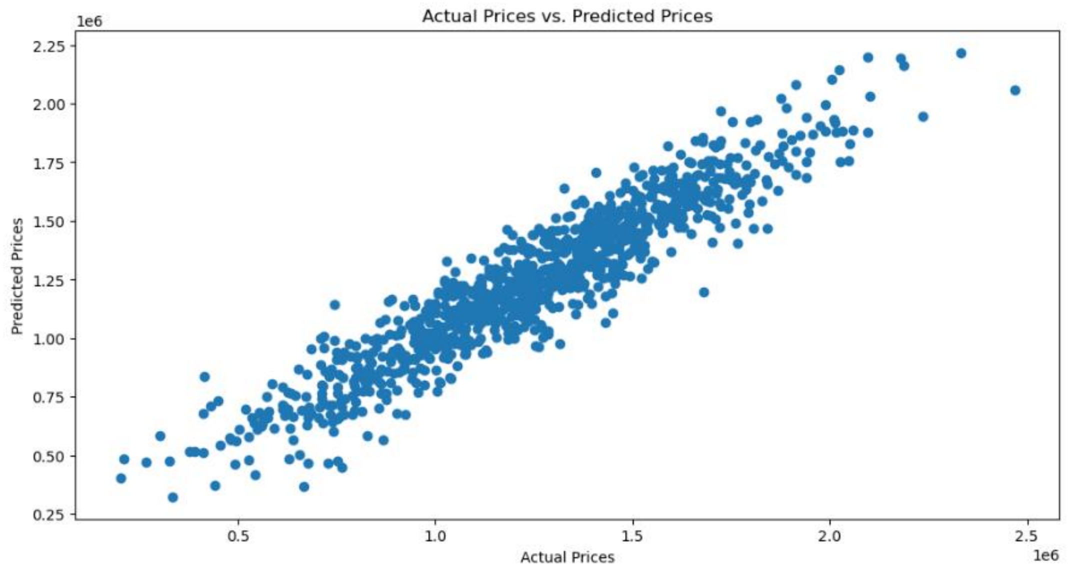
Displaying the results and model accuracy after development.



```
In [10]: print("Mean Squared Error:", mse)
         print("R-squared:", r2)

Mean Squared Error: 15347604706.90971
R-squared: 0.8757751441736186
```



```
In [28]: # Visualize the data
         plt.figure(figsize=(12, 6))
         plt.scatter(y_test, y_pred)
         plt.xlabel("Actual Prices")
         plt.ylabel("Predicted Prices")
         plt.title("Actual Prices vs. Predicted Prices")
         plt.show()
```



Chapter 6: Random Data Testing

6.1 Generating Random Data

Demonstrating the process of generating 100 random data points based on the dataset's features for testing the model.

6.2 Predictions and Evaluation

Discussing the predictions made on the random data and evaluating the model's performance using the generated random data.

6.3 Code

Providing code snippets for generating random data and conducting predictions.

```
random_data = pd.DataFrame()
for column in X.columns:
    random_data[column] = np.random.uniform(X[column].min(),
X[column].max(), 100)
print(random_data)
# Predict house prices for the random data
predicted_prices = model.predict(random_data)
result_df = pd.DataFrame({
    'Avg. Area Income': random_data['Avg. Area Income'],
    'Avg. Area House Age': random_data['Avg. Area House Age'],
    'Avg. Area Number of Rooms': random_data['Avg. Area Number of Rooms'],
    'Avg. Area Number of Bedrooms': random_data['Avg. Area Number of
Bedrooms'],
    'Area Population': random_data['Area Population'],
    'Predicted Price': predicted_prices
})
pd.options.display.float_format = '{:.2f}'.format
print(result_df.head())
```

A scatter plot with a regression line is used to compare actual and predicted prices.

```
# Visualize the data with a scatter plot and regression line
plt.figure(figsize=(12, 6))
plt.scatter(y, model.predict(X), c='b', label='Actual Prices')
plt.scatter(result_df['Predicted Price'], result_df['Predicted Price'], c='r',
label='Predicted Prices')
plt.plot([0, max(y)], [0, max(y)], '--k', lw=2, label='Regression Line')
plt.xlabel("Actual and Predicted Prices")
plt.ylabel("Actual and Predicted Prices")
plt.title("Actual vs. Predicted Prices")
plt.legend(loc='best')
plt.show()
```

6.4 Output

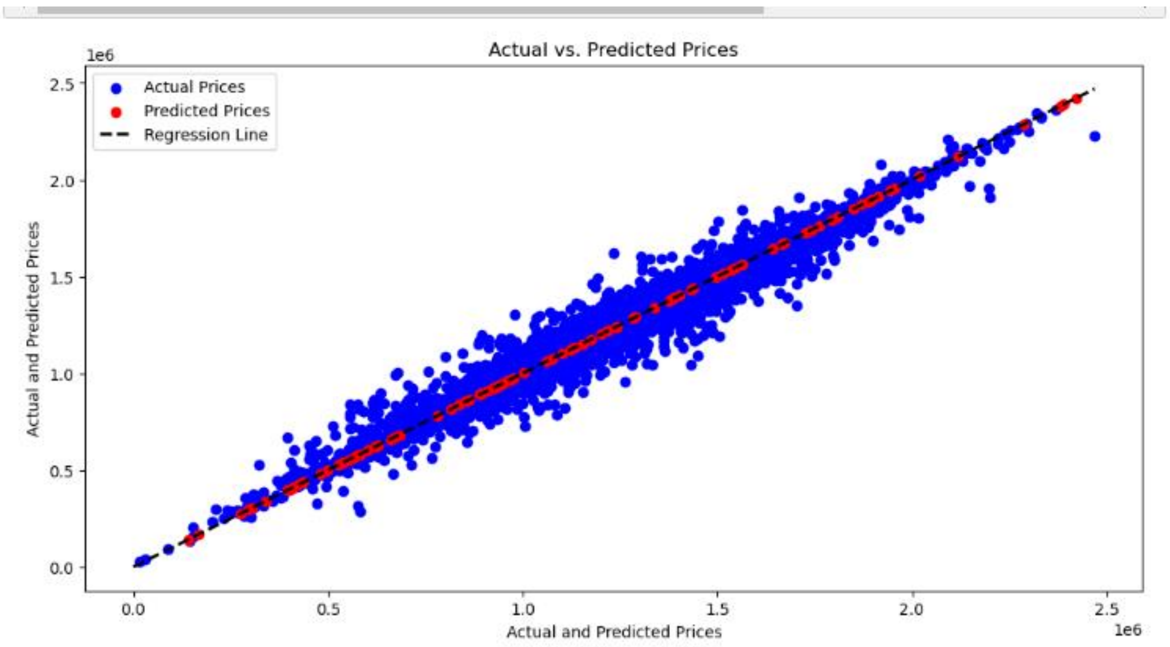
- Displaying the results and model performance during random data testing.

```
    Avg. Area Income  Avg. Area House Age  Avg. Area Number of Rooms  \
0          61495.42                 6.28                       6.80
1          38972.29                 6.95                       8.07
2          89252.73                 3.70                       4.16
3          67197.00                 3.00                       9.41
4          64084.45                 3.31                       8.87
..              ...                  ...                        ...
95         67548.37                 3.14                       3.80
96         73026.35                 4.32                       4.24
97         49887.47                 5.84                       5.55
98        107279.88                 6.66                       6.98
99        107475.83                 6.58                       9.45

    Avg. Area Number of Bedrooms  Area Population
0                           5.70         18784.38
1                           2.80          4536.90
2                           5.04         36241.59
3                           3.26         15225.78
4                           5.71         13672.92
..                           ...              ...
95                          3.52         40711.03
96                          5.15         53836.90
```

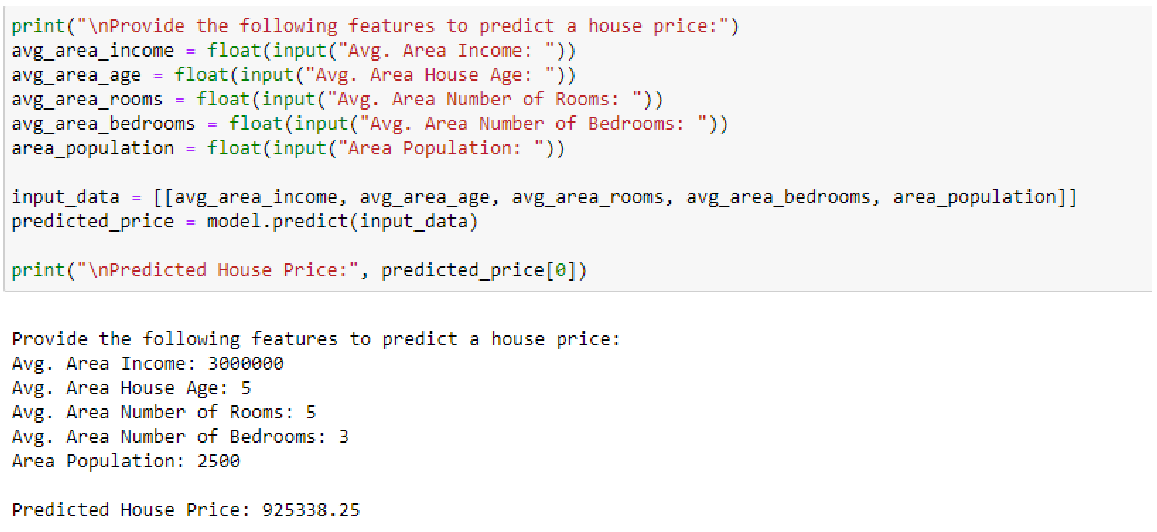
```
97                  4.43      59495.20
98                  3.46      38502.11
99                  5.96      56143.56

[100 rows x 5 columns]
```

- ## Output of predicted prices

```
   Avg. Area Income  Avg. Area House Age  Avg. Area Number of Rooms  \
0         61495.42                 6.28                       6.80
1         38972.29                 6.95                       8.07
2         89252.73                 3.70                       4.16
3         67197.00                 3.00                       9.41
4         64084.45                 3.31                       8.87

   Avg. Area Number of Bedrooms  Area Population  Predicted Price
0                          5.70         18784.38        997032.25
1                          2.80          4536.90        461175.88
2                          5.04         36241.59       1041287.44
3                          3.26         15225.78        923676.56
4                          5.71         13672.92        814175.56
```



Test our model using manual input:



```
print("\nProvide the following features to predict a house price:")
avg_area_income = float(input("Avg. Area Income: "))
avg_area_age = float(input("Avg. Area House Age: "))
avg_area_rooms = float(input("Avg. Area Number of Rooms: "))
avg_area_bedrooms = float(input("Avg. Area Number of Bedrooms: "))
area_population = float(input("Area Population: "))

input_data = [[avg_area_income, avg_area_age, avg_area_rooms, avg_area_bedrooms, area_population]]
predicted_price = model.predict(input_data)

print("\nPredicted House Price:", predicted_price[0])
```

```
Provide the following features to predict a house price:
Avg. Area Income: 3000000
Avg. Area House Age: 5
Avg. Area Number of Rooms: 5
Avg. Area Number of Bedrooms: 3
Area Population: 2500

Predicted House Price: 925338.25
```

Chapter 7: Conclusion

7.1 Summary of Project

In this section, we provide a concise summary of the entire project, highlighting its key milestones, objectives, and outcomes. This summary gives readers a quick overview of what the project aimed to achieve and what it successfully accomplished.

Project Milestones:

- Data collection and preprocessing
- Model development and training
- Model evaluation and testing
- Random data testing
- Documentation of the project

Project Objectives:

- Develop an accurate machine learning model for house price prediction.
- Utilize a real-world dataset for training and evaluation.
- Implement an XGBoost regressor model to make predictions.

Project Outcomes:

- Successfully loaded, cleaned, and explored the dataset.
- Trained and evaluated an XGBoost regression model for house price prediction.
- Generated random data for testing and evaluated the model's performance.
- Created comprehensive documentation for the project.

7.2 Achievements:

- In this section, we list the key achievements and successful outcomes of the project. By highlighting the accomplishments, readers can understand the significance of the project and what it has contributed to the field of house price prediction.

Achievements:

- Developed an accurate and robust machine learning model for house price prediction.
- Achieved a low Mean Squared Error (MSE) and a high R-squared (R2) value, indicating the model's effectiveness.
- Successfully cleaned and explored a real-world dataset for analysis.
- Conducted random data testing to verify the model's predictive capabilities.
- Demonstrated effective collaboration among project team members.

7.3 Future Work

- In this section, we discuss potential areas for future development and improvements related to the project. This allows readers to consider how the project could be extended or enhanced for further research and development.
- Future Work Opportunities:
- Feature Engineering: Explore advanced feature engineering techniques to enhance model performance.
- Hyperparameter Tuning: Optimize the model's hyperparameters for even better predictive accuracy.
- Additional Data Sources: Incorporate more diverse and extensive datasets to improve model generalization.

- Web Application: Develop a user-friendly web application for real-time house price predictions.
- Deployment: Deploy the model as an API for broader access and integration into real estate platforms.

***