4222 -SURYA GROUP OF INSTITUTION

VIKRAVANDI -605 652

SB3001 - EXPERIENCE BASED PRACTICAL LEARNING

ARTIFICAL INTELLIGENCE -TECHNOLOGY

PREDICTION HOUSE PRICES USING MACHINE LEARNING

NAAN MUDHALVAN PROJECT

PREPARED BY :

D.RAJASEKAR

REG NO :422221106015

ECE DEPARTMENT

3RD YEAR

INTRODUCTION:

House price prediction using machine learning is a common and well-established task in the field of data science and real estate. You can use various machine learning algorithms to predict house prices based on historical data and a set of relevant features. Here's a step-by-step guide on how to approach this task:

Data Collection:

Gather a dataset that includes historical information about houses, such as the number of bedrooms, bathrooms, square footage, location, age, and, most importantly, the actual selling prices.

Data Preprocessing:

Handle missing data: Identify and fill in missing values in the dataset.

Encode categorical variables: Convert categorical features like "location" into numerical values using techniques like one-hot encoding or label encoding.

Feature scaling: Normalize or standardize numerical features to ensure they are on a similar scale.

Feature Selection/Engineering:

Select relevant features that are likely to influence the house price. This may involve domain knowledge and statistical analysis.

Create new features if necessary, such as the price per square foot or the age of the house at the time of sale.

Split Data:

Divide the dataset into two parts: a training set and a testing set. A common split is 70-80% for training and the remainder for testing.

Choose a Machine Learning Algorithm:

Select a regression algorithm suitable for predicting house prices. Common choices include Linear Regression, Decision Trees, Random Forest, Support Vector Machines, and Gradient Boosting algorithms like XG Boost or Linear regression.

Train the Model:

Use the training data to train the chosen algorithm. The algorithm will learn the relationships between the features and the house prices in the training dataset.

Evaluate the Model:

Use the testing dataset to evaluate the model's performance. Common regression metrics for evaluation include Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2).

Hyperparameter Tuning:

Fine-tune the model's hyperparameters to optimize its performance. This can be done using techniques like grid search or random search.

Make Predictions:

Once you are satisfied with the model's performance, you can use it to make predictions on new, unseen data.

Deployment:

If you plan to make your model available for others to use, you can deploy it as a web application or API.

Regular Maintenance:

Keep the model updated with new data if you want it to provide accurate predictions over time.

Interpret Results:

Analyze the importance of features in predicting house prices to gain insights into the factors that influence the market.

Remember that house price prediction is a complex task influenced by many factors, and no model will be perfect. It's important to continuously improve and update your model to ensure its accuracy. Additionally, ethical considerations and fairness in pricing should be taken into account when developing such models to avoid bias and discrimination.

PROBLEM DEFINITION :

OBJECTIVE:

- People looking to buy a new home tend to be more conservative with their budgets and market strategies.

- This project aims to analyze various parameters like average income, average area etc. and predict the house price accordingly.

- This application will help customers to invest in an estate without approaching an agent

- To provide a better and fast way of performing operations. • To provide proper house price to the customers.

- To eliminate need of real estate agent to gain information regarding house prices.

- To provide best price to user without getting cheated.

- To enable user to search home as per the budget.

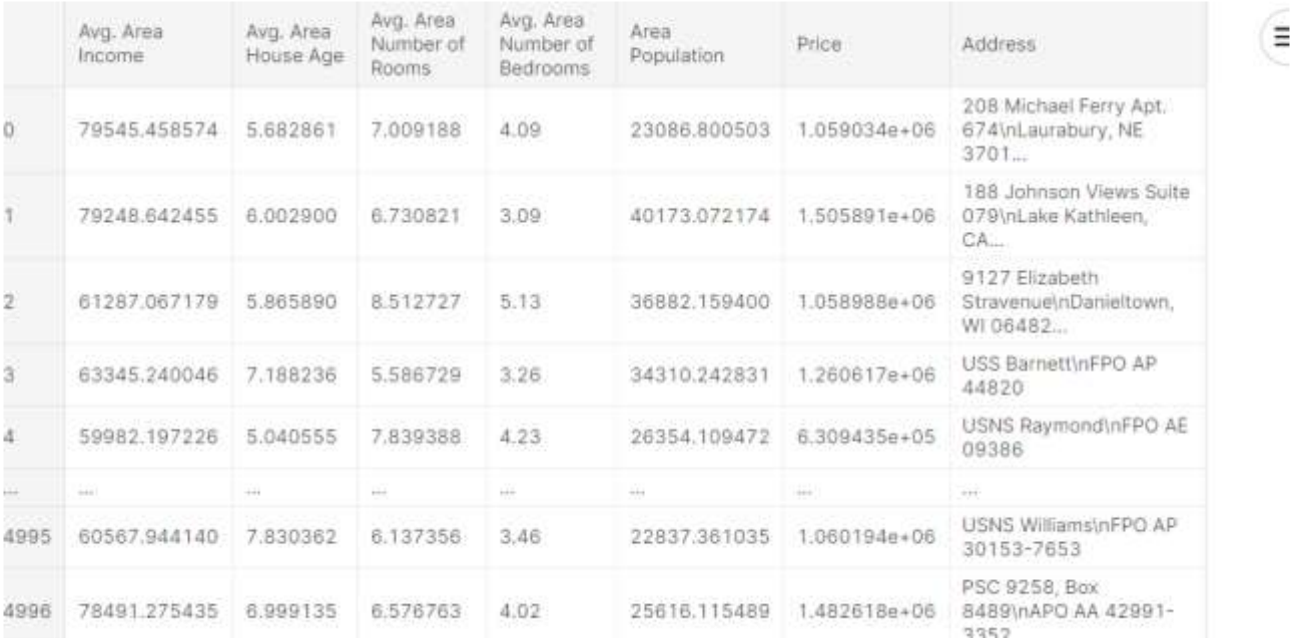
- The aim is to predict the efficient house pricing for real estate customers with respect to their budgets and priorities. By analyzing previous market trends and price ranges, and also upcoming developments future prices will be predicted.

- House prices increase every year, so there is a need for a system to predict house prices in the future.

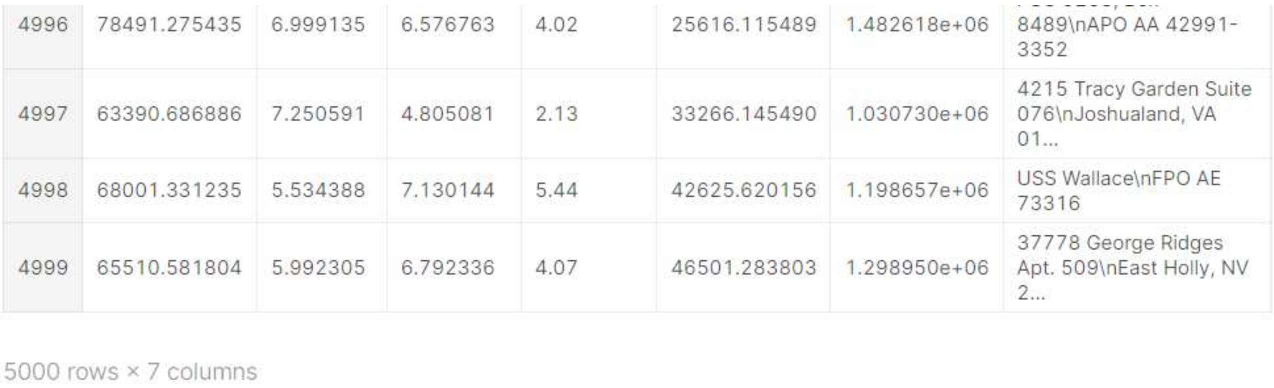
- House price prediction can help the developer determine the selling price of a house and can help the customer to arrange the right time to purchase a house.

- We use Random forest regression algorithm in machine learning for predicting the house price trends

GIVEN DATA SET:



	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Address
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson Views Suite 079\nLake Kathleen, CA...
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Elizabeth Stravenue\nDanieltown, WI 06482...
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nFPO AP 44820
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\nFPO AE 09386
...	...	...	...	...	...	...	...
4995	60567.944140	7.830362	6.137356	3.46	22837.361035	1.060194e+06	USNS Williams\nFPO AP 30153-7653
4996	78491.275435	6.999135	6.576763	4.02	25616.115489	1.482618e+06	PSC 9258, Box 8489\nAPO AA 42991-3352



4996	78491.275435	6.999135	6.576763	4.02	25616.115489	1.482618e+06	8489\nAPO AA 42991-3352
4997	63390.686886	7.250591	4.805081	2.13	33266.145490	1.030730e+06	4215 Tracy Garden Suite 076\nJoshualand, VA 01...
4998	68001.331235	5.534388	7.130144	5.44	42625.620156	1.198657e+06	USS Wallace\nFPO AE 73316
4999	65510.581804	5.992305	6.792336	4.07	46501.283803	1.298950e+06	37778 George Ridges Apt. 509\nEast Holly, NV 2...

5000 rows × 7 columns

DATA SOURCE : Collection of data processing techniques and processes are numerous. We collected data for USA real estate properties from various real estate websites. The data would be having attributes such as Location, carpet area, built-up area, age of the property, zip code, price, no of bed rooms etc. We must collect the quantitative data which is structured and categorized. Data collection is needed before any kind of machine learning research is carried out. Dataset validity is a must otherwise there is no point in analyzing the data.

Data preprocessing :

Data preprocessing is the process of cleaning our data set. There might be missing values or outliers in the dataset. These can be handled by data cleaning. If there are many missing values in a variable we will drop those values or substitute it with the average value

```
dataset = pd.read_csv('/kaggle/input/usa-housing/USA_Housing.csv') dataset.info()
dataset.describe()

    dataset.columns
```

VISUALISATION AND PRE PROCESSING DATA

```
    sns.histplot(dataset, x='Price', bins=50, color='y')
sns.boxplot(dataset, x='Price', palette='Blues') sns.jointplot(dataset,
x='Avg. Area House Age', y='Price', kind='hex') sns.jointplot(dataset,
x='Avg. Area Income', y='Price')
    plt.figure(figsize=(12,8))
sns.pairplot(dataset)
```

```
dataset.hist(figsize=(10,8))

dataset.corr(numeric_only=True

plt.figure(figsize=(10,5))

     sns.heatmap(dataset.corr(numeric_only = True), annot=True)
```

DIVIDE THE DATA SET

```
X = dataset[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of
  Rooms',

       'Avg. Area Number of Bedrooms', 'Area Population']]
Y = dataset['Price']
```

USING TRAIN TEST SPLIT

```
         X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,
random_state=101)

Y_train.head()

Y_train.shape
Y_test.head()

Y_test.shape
 sc =
StandardScaler()
X_train_scal =
sc.fit_transform(X_tr
ain)
X_test_scal = sc.fit_transform(X_test)
```

MODEL BULIDING AND EVALUTION OF PREDICATED DATA

```
    model_lr=LinearRegression()

odel_lr.fit(X_train_scal, Y_train)
```

```
   Prediction1 = model_lr.predict(X_test_scal)

plt.figure(figsize=(12,6))

    plt.plot(np.arange(len(Y_test)), Y_test, label='Actual Trend')
plt.plot(np.arange(len(Y_test)), Prediction1, label='Predicted Trend')
plt.xlabel('Data')     plt.ylabel('Trend')     plt.legend()
plt.title('Actual vs Predicted')     ns.histplot((Y_test-Prediction1),
bins=50)

  print(r2_score(Y_test,

Prediction2))

 print(mean_absolute_error(Y_test, Prediction2))
print(mean_squared_error(Y_test, Prediction2))


print(r2_score(Y_test, Prediction1)) print(mean_absolute_error(Y_test,
Prediction1)) print(mean_squared_error(Y_test, Prediction1))



 Model_rf = RandomForestRegressor(n_estimators=50)


  model_rf.fit(X_train_scal, Y_train)
```

AI_ PHASE 2:

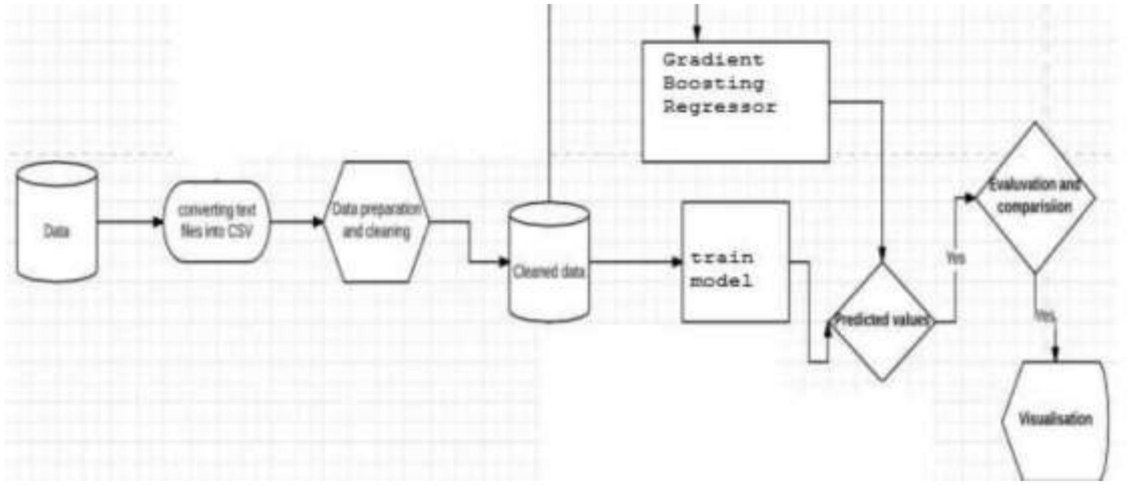
Consider exploring advanced regression techniques like XG Boost for Improved prediction accuracy.

PROPOSED SYSTEM:

In this proposed system, we focus on predicting the house price values using machine learning

algorithms like XG Boost regression model. We proposed the system “House price prediction using Machine Learning” we have predicted the House price using XG boost regression model. In this proposed system, we were able to train the machine from the various attributes of data points from the past to make a future prediction. We took data from the previous year stocks to train the model .The data set we used was from the official organization. Some of data was used to train the machine and the rest some data is used to test the data. The basic approach of the supervised learning model is to learn the patterns and relationships in the data from the training set and then reproduce them for the test data. We used the python pandas library for data processing which combined different datasets into a data frame. The raw data makes us to prepare the data for feature identification. The attributes were stories, no. of bed rooms, bath rooms, Availability of garage, swimming pool, fire place, year built, area in soft, sale price for a particular house. We used all these features to train the machine on XG boost regression and predicted the house price, which is the price for a given day. We also quantified the accuracy by using the predictions for the test set and the actual values. The proposed system gives the Predicted price.

ARCHITECTURE:


Data
converting text files into CSV
Data preparation and cleaning
Cleaned data
Gradient Boosting Regressor
train model
Predicted values
Yes
Evaluation and comparision
Yes
Visualisation

ALOGORITHM:

We used the python pandas library for data processing which combined different datasets into a data frame. The raw data makes us to prepare the data for feature identification. XG for regression builds an additive model in a forward stage wise fashion. It allows for the optimization of arbitrary differentiable loss functions. In each stage, a regression tree is fit on the negative XG of the given loss function. The idea of boosting came out of the idea of whether a weak learner can be modified to become better. A weak hypothesis is defined as one whose performance is at least slightly better than random chance. The Objective is to minimize the loss of the model by adding weak hypothesis using a XG descent like procedure. This class of algorithms was described as a stage-wise additive model. This is because one new weak learner is added at a time and existing weak learners in the model are frozen and left unchanged.

Step 1: Load the data set df = pd.read csv(“ml_house_data_set.csv”)

Step 2: Replace categorical data with one-hot encoded data

Step 3: Remove the sale price from the feature data

Step 4: Create the features and labels X and Y arrays.

Step 5: Split the data set in a training set (70%) and a test set (30%). Step 6: Fit regression model.

Step 7: Save the trained model to a file trained_house_classifier_model.pkl

Step 8: Predict house worth using predict function

MODEL XG BOOST REGRESSOR :

INPUT:

```
model_xg = xg.XGBRegressor()
```

```
model_xg.fit(X_train_scal, Y_train)
```

OUTPUT:

```
XGB Regressor
XGB Regressor (base_score=None, booster=None, callbacks=None,
colsample_bylevel=None, colsample_bynode=None,
colsample_bytree=None, early_stopping_rounds=None,
enable_categorical=False, eval_metric=None, feature_types=None,
gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
interaction_constraints=None, learning_rate=None, max_bin=None,
max_cat_threshold=None, max_cat_to_onehot=None,                max_delta_step=None,
max_depth=None, max_leaves=None,                         min_child_weight=None, missing=nan,
monotone_constraints=None,                          n_estimators=100, n_jobs=None,
num_parallel_tree=None,                        predictor=None, random_state=None, ...)
```

PREDICTING PRICES:

```
                              Prediction5 = model_xg.predict(X_test_scal)
```

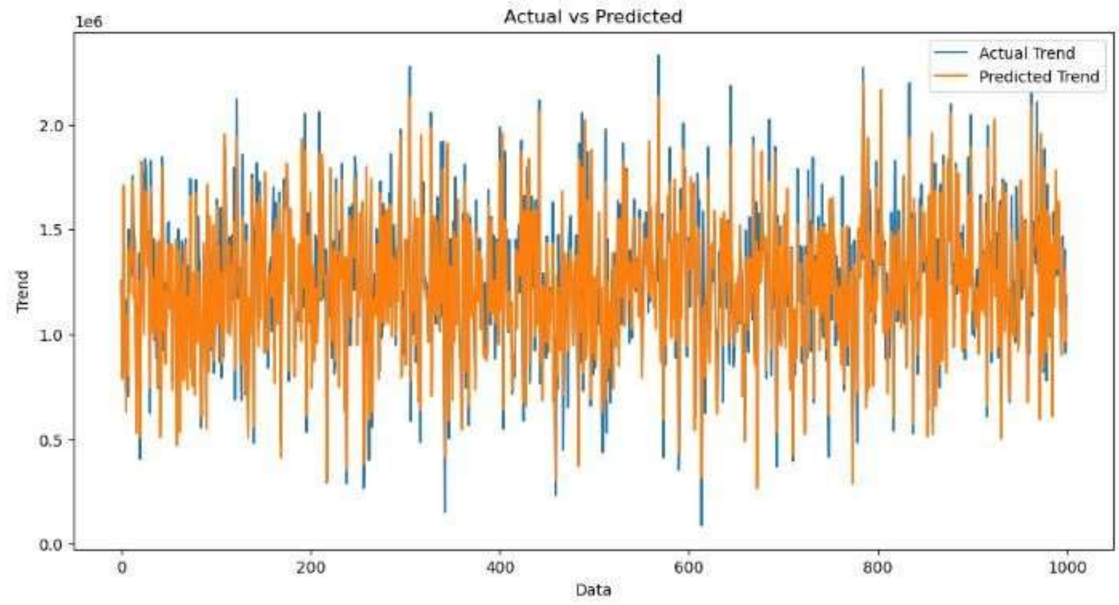
EVALUATION OF PREDICTING DATA:

INPUT:

```
                plt.figure(figsize=(12,6))                    plt.plot(np.arange(len(Y_test)), Y_test, label='Actual Trend')
plt.plot(np.arange(len(Y_test)), Prediction5, label='Predicted Trend')                    plt.xlabel('Data')
plt.ylabel('Trend')                    plt.legend()
                plt.title('Actual vs Predicted')
```

OUTPUT:

```
               Text(0.5, 1.0, 'Actual vs Predicted')
```

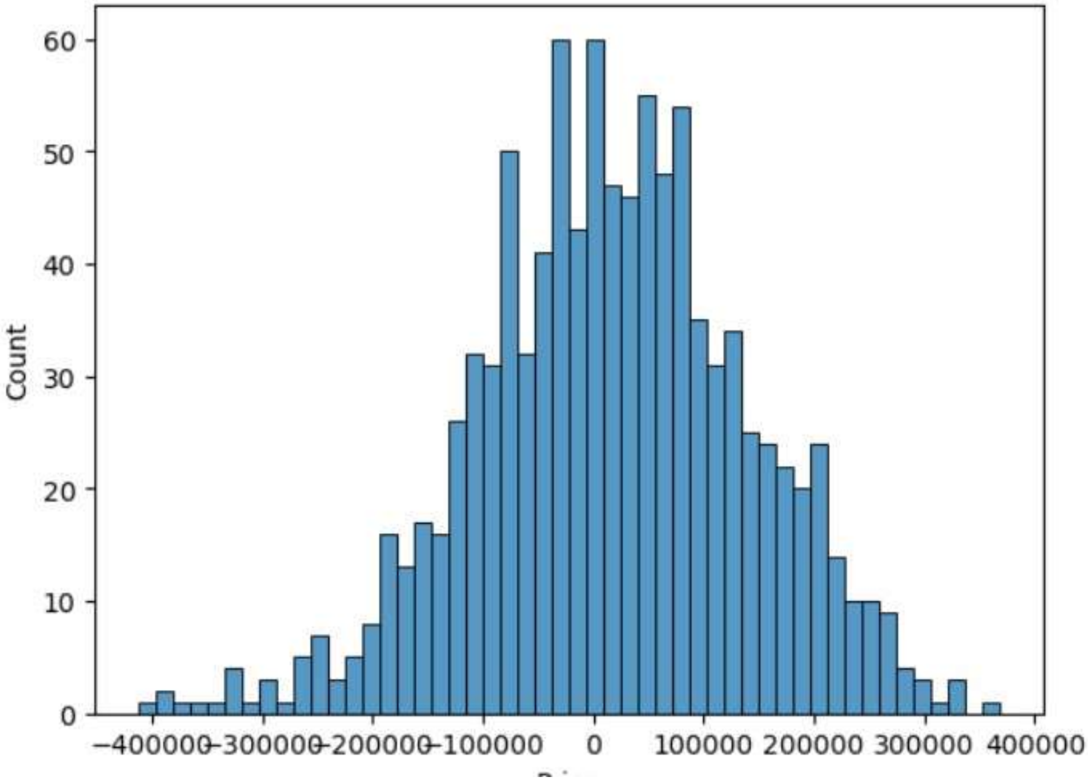


INPUT;

```
        sns.histplot((Y_test-Prediction4), bins=50)
```

OUTPUT:

```
<Axes: xlabel='Price', ylabel='Count'>
```



Count
60
50
40
30
20
10
0
−400000
−300000
−200000
−100000
0
100000
200000
300000
400000

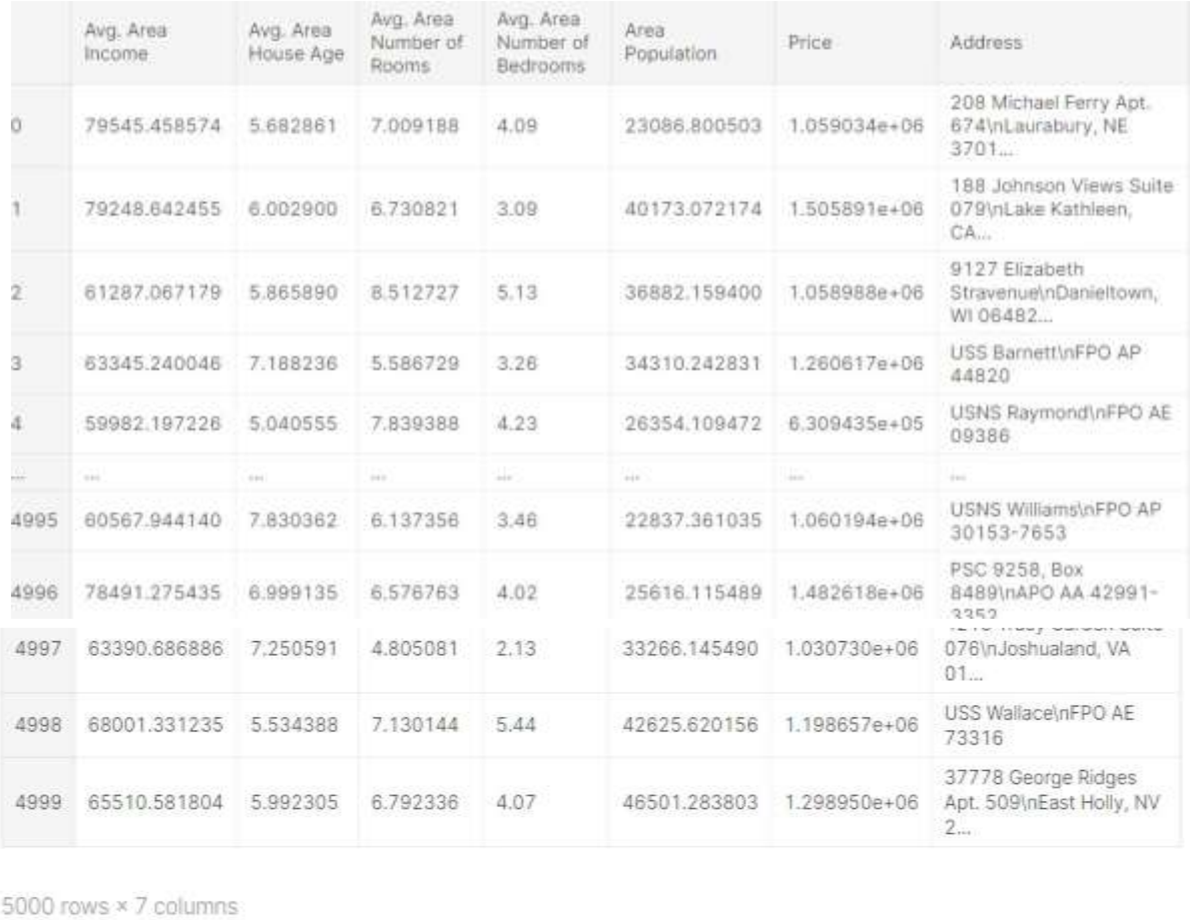
PHASE 3 DEVELOPMENT PART 1

PREDICTION HOUSE PRICES USING MACHINE LEARNING

AI_ PHASE 3:

Data preprocessing is the process of cleaning our data set. There might be missing values or outliers in the dataset. These can be handled by data cleaning. If there are many missing values in variable we will drop those values or substitute it with the average value.

PREPROCESS THE GIVEN DATA SET:



	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Address
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson Views Suite 079\nLake Kathleen, CA...
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Elizabeth Stravenue\nDanieltown, WI 06482...
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nFPO AP 44820
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\nFPO AE 09386
...	...	...	...	...	...	...	...
4995	60567.944140	7.830362	6.137356	3.46	22837.361035	1.060194e+06	USNS Williams\nFPO AP 30153-7653
4996	78491.275435	6.999135	6.576763	4.02	25616.115489	1.482618e+06	PSC 9258, Box 8489\nAPO AA 42991-3352
4997	63390.686886	7.250591	4.805081	2.13	33266.145490	1.030730e+06	4215 Tracy Garden Suite 076\nJoshualand, VA 01...
4998	68001.331235	5.534388	7.130144	5.44	42625.620156	1.198657e+06	USS Wallace\nFPO AE 73316
4999	65510.581804	5.992305	6.792336	4.07	46501.283803	1.298950e+06	37778 George Ridges Apt. 509\nEast Holly, NV 2...

5000 rows × 7 columns

THE DATA SET INFO :

Input: dataset.info()

Output:

```
<class 'panda.core.frame.DataFrame'>
Range Index: 5000 entries, 0 to 4999 Data
columns (total 7 columns):
 #   Column                          Non-Null Count  D type
---  ------                          --------------  -----
0   Avg. Area Income               5000 non-null    float64
```

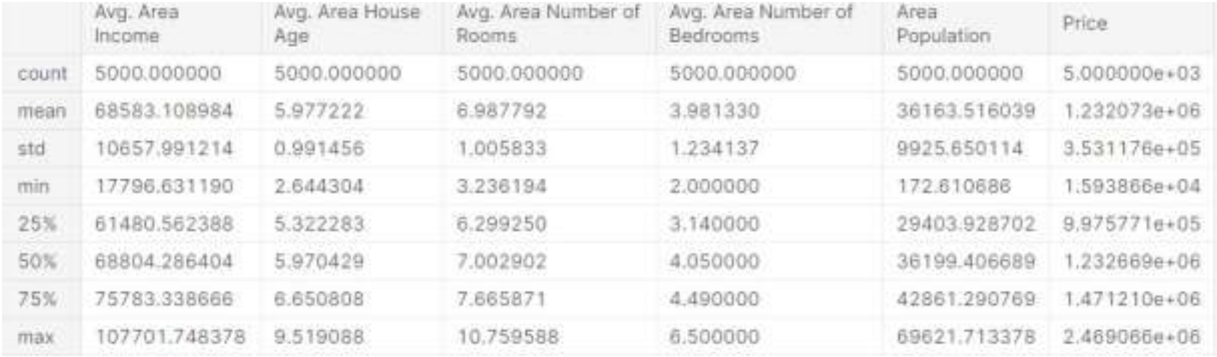
```
1   Avg. Area House Age           5000 non-null   float64
2   Avg. Area Number of Rooms      5000 non-null   float64
3   Avg. Area Number of Bedrooms  5000 non-null   float64
4   Area Population               5000 non-null   float64
5   Price                         5000 non-null   float64  6   Address
    5000 non-null   object
dtypes: float64(6), object(1)
memory usage: 273.6+ KB
```

DESCRIBE:

Input:

```
dataset.describe()
```

Output:



	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5.000000e+03
mean	68583.108984	5.977222	6.987792	3.981330	36163.516039	1.232073e+06
std	10657.991214	0.991456	1.005833	1.234137	9925.650114	3.531176e+05
min	17796.631190	2.644304	3.236194	2.000000	172.610686	1.593866e+04
25%	61480.562388	5.322283	6.299250	3.140000	29403.928702	9.975771e+05
50%	68804.286404	5.970429	7.002902	4.050000	36199.406689	1.232669e+06
75%	75783.338666	6.650808	7.665871	4.490000	42861.290769	1.471210e+06
max	107701.748378	9.519088	10.759588	6.500000	69621.713378	2.469066e+06

```
dataset.columns
```

```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Address'],              dtype='object')
```

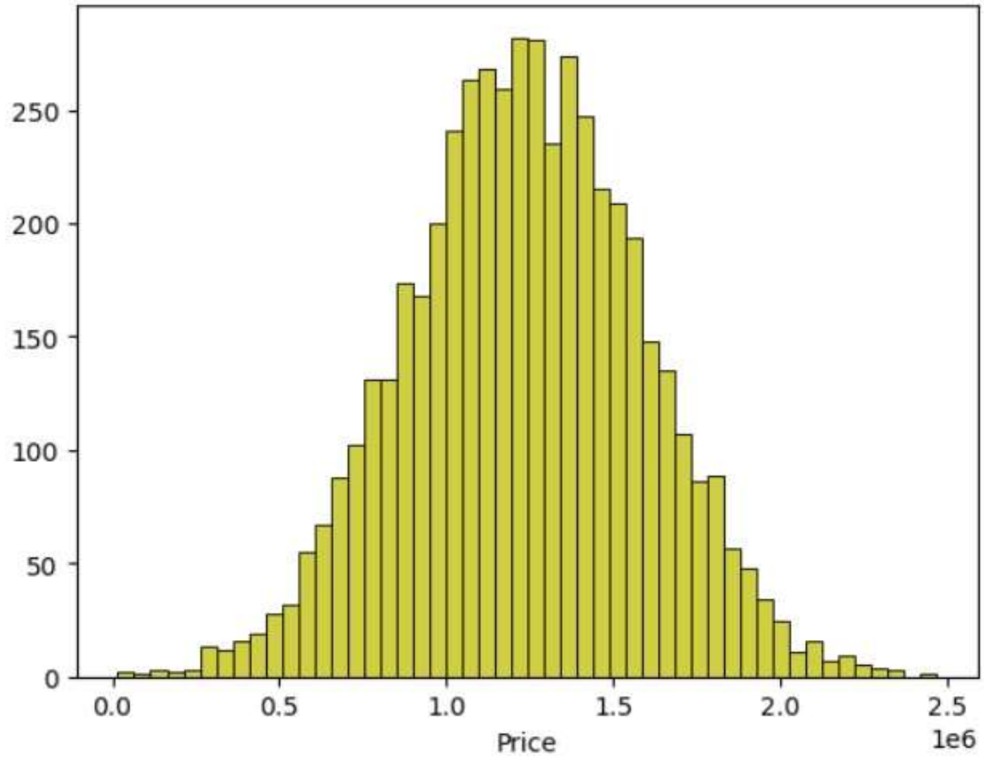
VISUALIZATION AND PRE PROCESSING THE DATA:

Input:

```
sns.histplot(dataset, x='Price', bins=50, color='y')
```

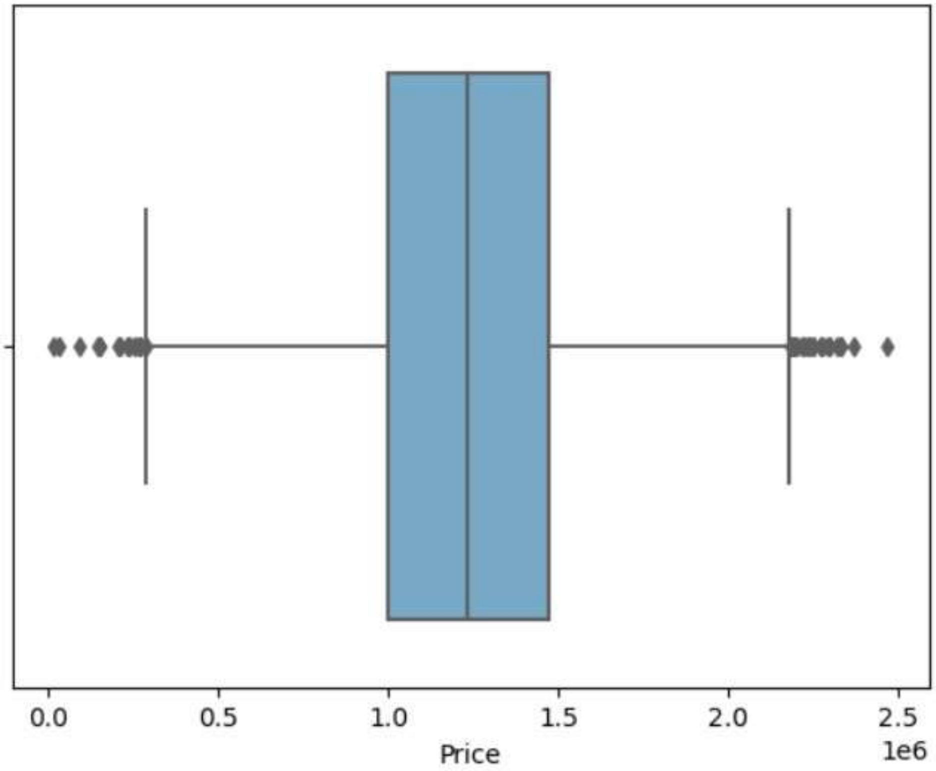
Output :

```
<Axes: xlabel='Price', ylabel='Count'>
```



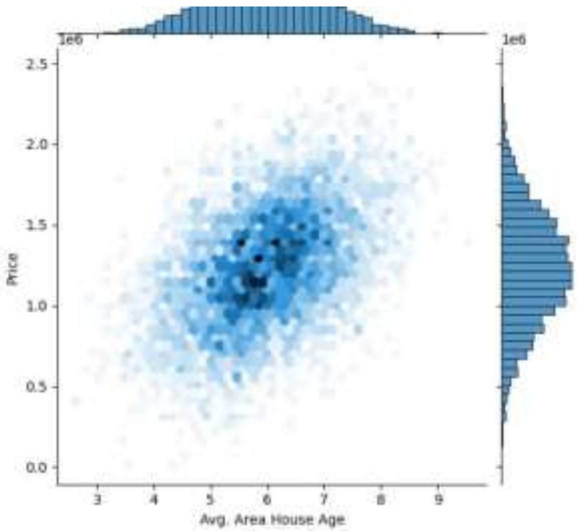
```
sns.boxplot(dataset, x='Price',  palette='Blues')
```

```
 <Axes: xlabel='Price'>
```



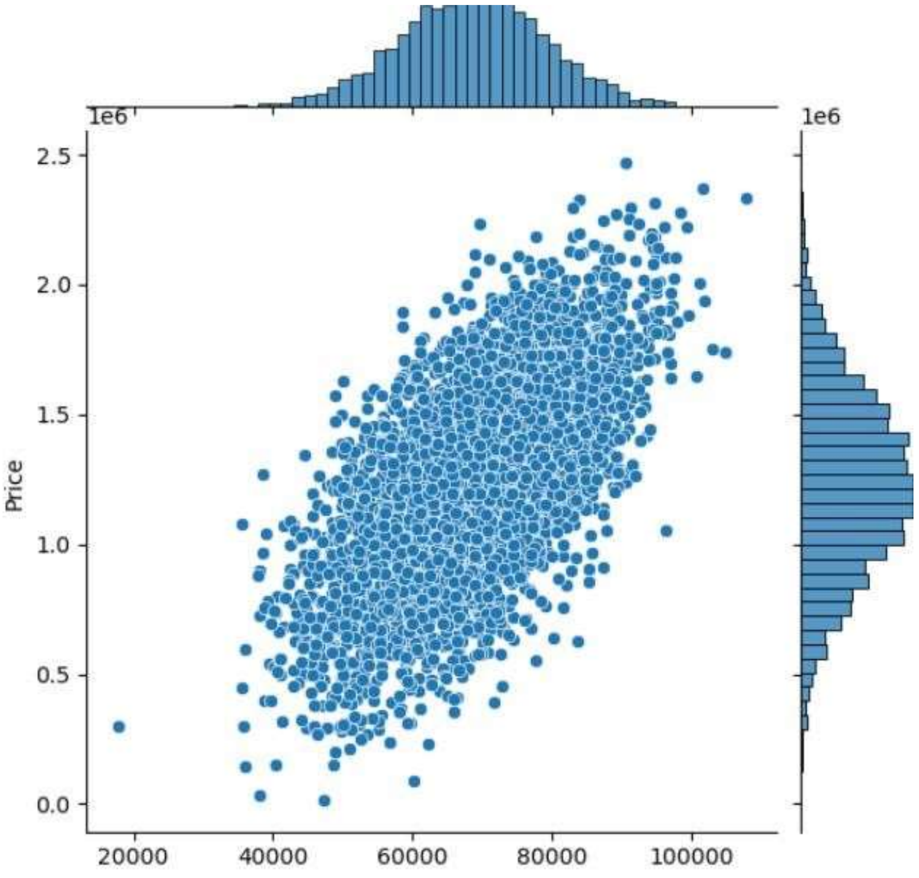
```
sns.jointplot(dataset, x='Avg. Area House Age', y='Price', kind='hex')
```

```
<seaborn.axisgrid.JointGrid at 0x7a2b71bd29b0>
```



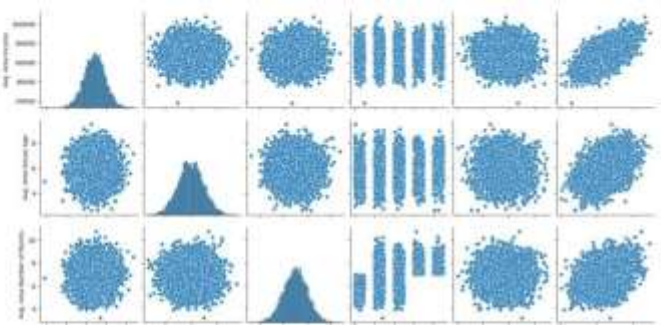
```
sns.jointplot(Income', y='Price')dataset, x='Avg. Area
```

```
<seaborn.axisgrid.JointGrid at 0x7a2b5e50cbb0>
```



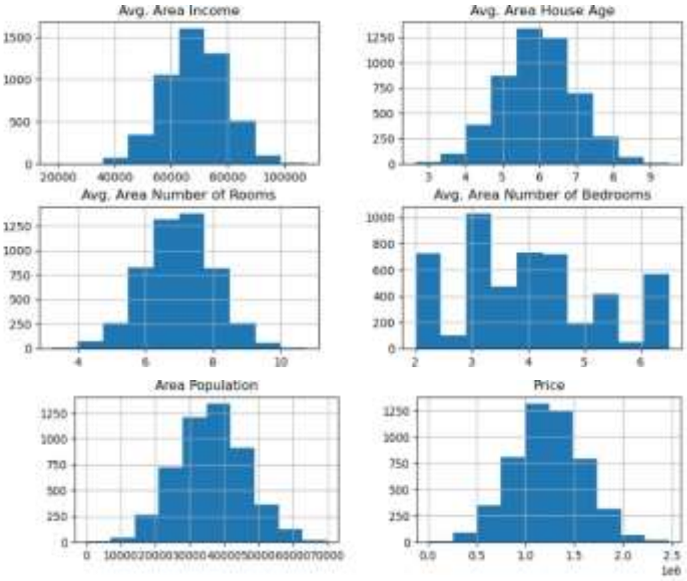
```
plt.figure(figsize=(12,8))
sns.pairplot(dataset)
```

```
<seaborn.axisgrid.PairGrid at 0x7a2b5e5ce350>
```



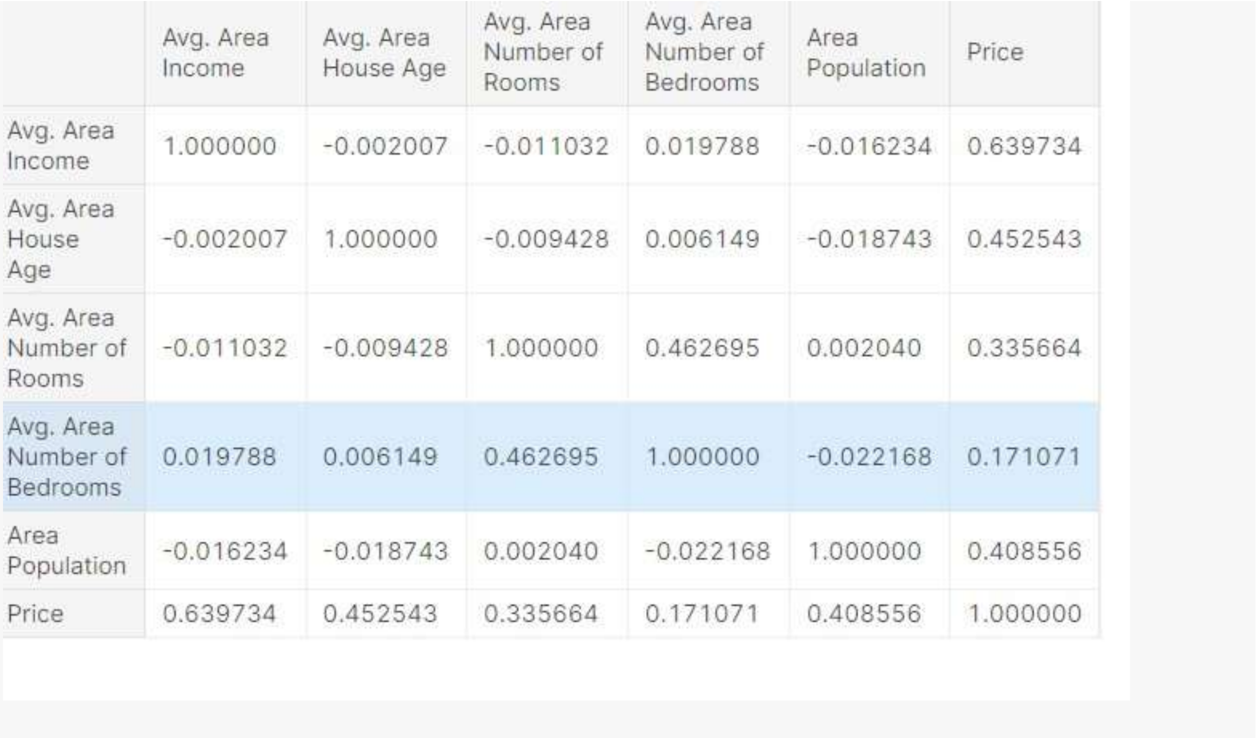
```
dataset.hist(figsize=(10,8))
```

```
array([[<Axes: title={'center': 'Avg. Area Income'}>,
<Axes: title={'center': 'Avg. Area House Age'}>],
       [<Axes: title={'center': 'Avg. Area Number of Rooms'}>,
        <Axes: title={'center': 'Avg. Area Number of Bedrooms'}>],
       [<Axes: title={'center': 'Area Population'}>,
        <Axes: title={'center': 'Price'}>]], dtype=object)
```



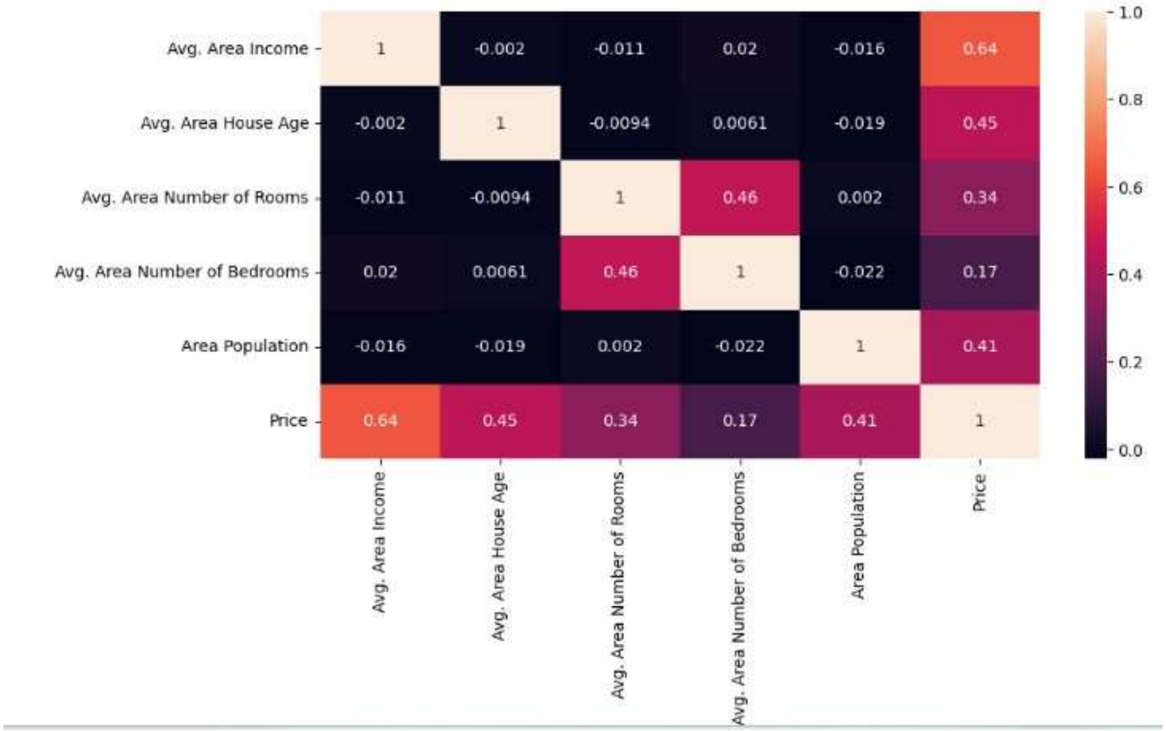
VISUALISING CORRELATION

```
dataset.corr(numeric_only=True)
```



	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
Avg. Area Income	1.000000	-0.002007	-0.011032	0.019788	-0.016234	0.639734
Avg. Area House Age	-0.002007	1.000000	-0.009428	0.006149	-0.018743	0.452543
Avg. Area Number of Rooms	-0.011032	-0.009428	1.000000	0.462695	0.002040	0.335664
Avg. Area Number of Bedrooms	0.019788	0.006149	0.462695	1.000000	-0.022168	0.171071
Area Population	-0.016234	-0.018743	0.002040	-0.022168	1.000000	0.408556
Price	0.639734	0.452543	0.335664	0.171071	0.408556	1.000000

```
plt.figure(figsize=(10,5))
sns.heatmap(dataset.corr(numeric_only = True), annot=True)
```



AI PHASE 4

To Develop the project development part 2 is build a model evaluation, model training by using Linear Regression and XG boost regression.

In this proposed system, we focus on predicting the house price values using machine learning algorithms like XG Boost regression model and Linear Regression . We proposed the system "House price prediction using Machine Learning" we have predicted the House price using XG boost regression and linear regression model. In this proposed system, we were able to train the machine from the various attributes of data points from the past to make a future prediction. We took data from the previous year stocks to train the model .The data set we used was from the official organization. Some of data was used to train the machine and the rest some data is used to test the data. The basic approach of the supervised learning model is to learn the patterns and relationships in the data from the training set and then reproduce them for the test data. We used the python pandas library for data processing which combined different datasets into a data frame. The raw data makes us to prepare the data for feature identification. The attributes were stories, no. of bed rooms, bath rooms, Availability of garage, swimming pool, fire place, year built, area in soft, sale price for a particular house. We used all these features to train the machine on XG boost regression and predicted the house price, which is the price for a given day. We also quantified the accuracy by using the predictions for the test set and the actual values. The proposed system gives the Predicted price

ALOGORITHM:

We used the python pandas library for data processing which combined different datasets into a data frame. The raw data makes us to prepare the data for feature identification. XG for regression builds an additive model in a forward stage wise fashion. It allows for the optimization of arbitrary differentiable loss functions. In each stage, a regression tree is fit on the negative XG of the given loss function. The idea of boosting came out of the idea of whether a weak learner can be modified to become better. A weak hypothesis is defined as one whose performance is at least slightly better than random chance. The Objective is to minimize the loss of the model by adding weak hypothesis using a XG descent like procedure. This class of algorithms was described as a stage-wise additive model. This is because one new weak learner is added at a time and existing weak learners in the model are frozen and left unchanged.

Step 1: Load the data set df = pd.read csv("ml_house_data_set.csv")

Step 2: Replace categorical data with one-hot encoded data

Step 3: Remove the sale price from the feature data

Step 4: Create the features and labels X and Y arrays.

Step 5: Split the data set in a training set (70%) and a test set (30%).

Step 6: Fit regression model.

Step 7: Save the trained model to a file trained_house_classifier_model.pkl

Step 8: Predict house worth using predict function

Model Building and evaluation:

Model 1 : Linear regression

Input;

```
model_lr=LinearRegression() model_lr.fit(X_train_scal,
Y_train)
```

Output:

```
LinearRegression()
```

Predicting Prices

```
Prediction1 = model_lr.predict(X_test_scal)
```

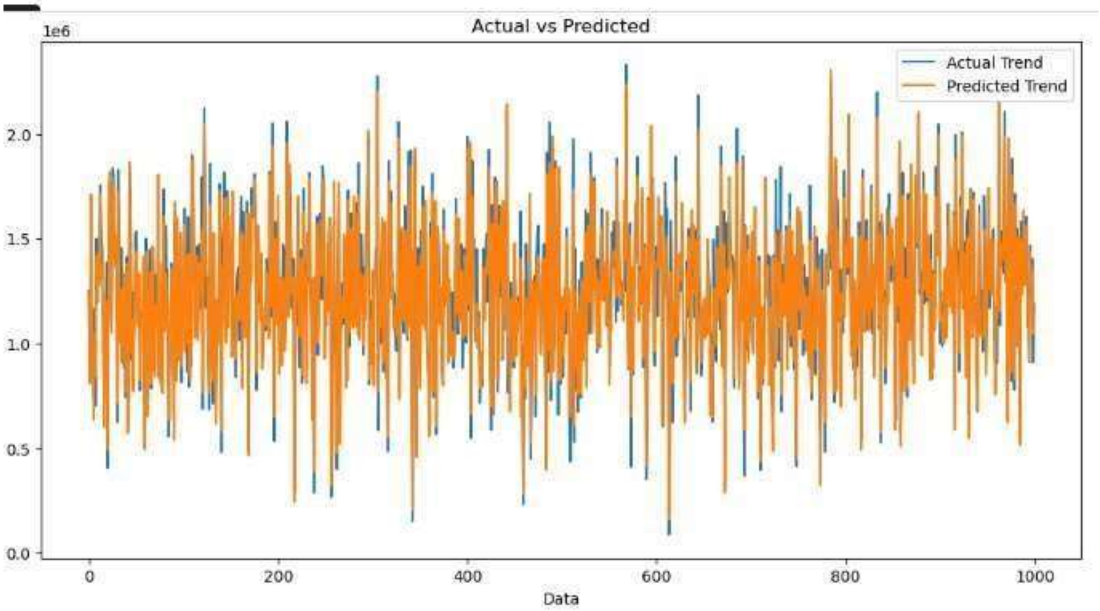
Evaluation of predicted data:

Input :

```
 plt.figure(figsize=(12,6))     plt.plot(np.arange(len(Y_test)), Y_test,
label='Actual Trend')     plt.plot(np.arange(len(Y_test)), Prediction1,
label='Predicted Trend')     plt.xlabel('Data')     plt.ylabel('Trend')     plt.legend()
plt.title('Actual vs Predicted')
```

Output:

```
Text(0.5, 1.0, 'Actual vs Predicted')
```



To Find a Mean absolute Error:

Input:

```
sns.histplot((Y_test-Prediction1), bins=50)
```

Output:

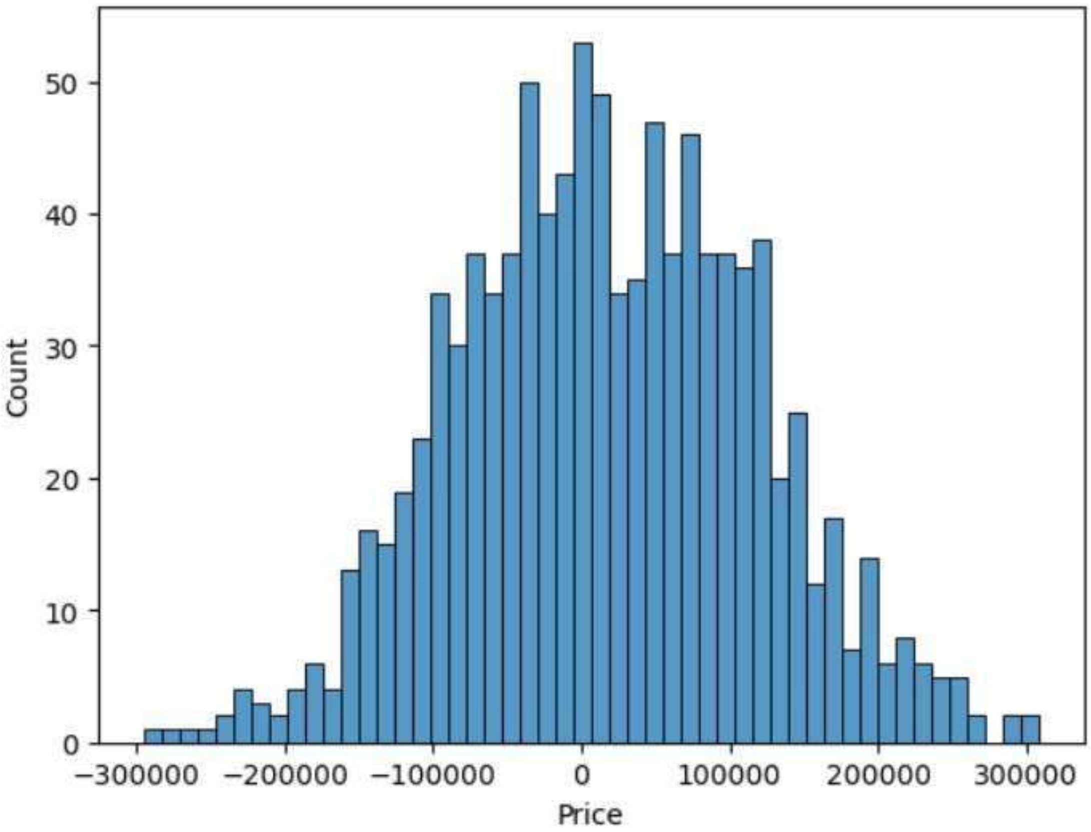
```
<Axes: xlabel='Price', ylabel='Count'>
```

Input:

```
print(r2_score(Y_test, Prediction1)) print(mean_absolute_error(Y_test,
Prediction1)) print(mean_squared_error(Y_test, Prediction1))
```

Output:

```
0.9182928179392918
82295.49779231755
10469084772.975954
```



Model 2: XG Boost Regressor :

Input:

```
model_xg = xg.XGBRegressor()  model_xg.fit(X_train_scal,

Y_train)
```

Output:

```
  XGBRegressor (base_score=None, booster=None, callbacks=None,
colsample_bylevel=None, colsample_bynode=None,              colsample_bytree=None,
early_stopping_rounds=None,              enable_categorical=False, eval_metric=None,
feature_types=None,              gamma=None, gpu_id=None, grow_policy=None,
importance_type=None,              interaction_constraints=None, learning_rate=None,
max_bin=None,              max_cat_threshold=None, max_cat_to_onehot=None,
max_delta_step=None, max_depth=None, max_leaves=None,              min_child_weight=None,
missing=nan, monotone_constraints=None,              n_estimators=100, n_jobs=None,
num_parallel_tree=None,              predictor=None, random_state=None, ...)
```

Predicting Prices :

```
Prediction5 = model_xg.predict(X_test_scal)
```

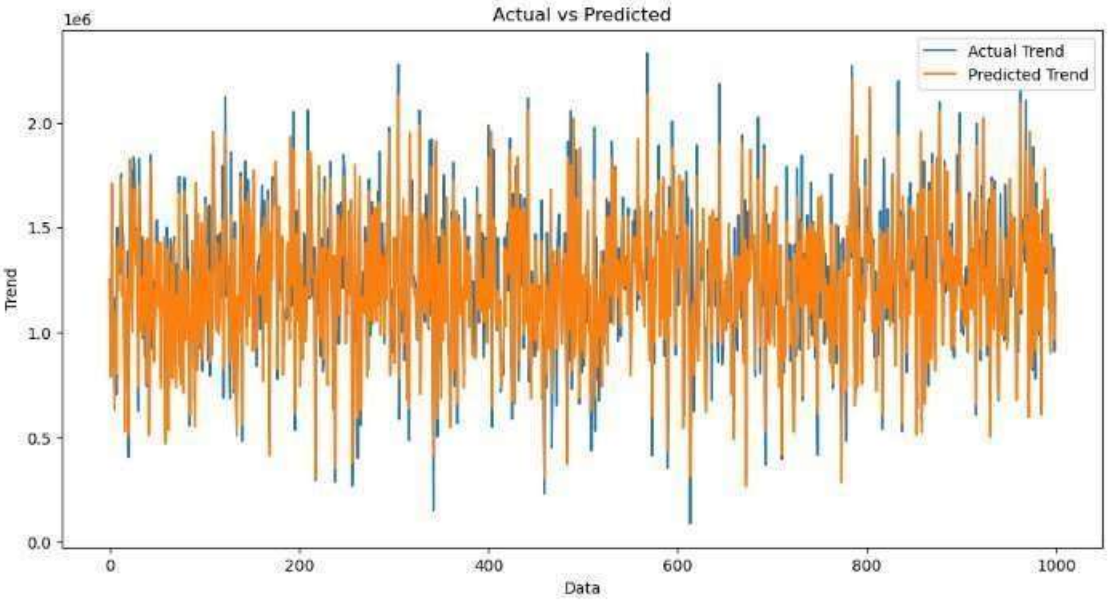
Evaluation of Predicting Prices:

Input:

```
plt.figure(figsize=(12,6))
plt.plot(np.arange(len(Y_test)), Y_test, label='Actual Trend')
plt.plot(np.arange(len(Y_test)), Prediction5, label='Predicted Trend')
plt.xlabel('Data')                  plt.ylabel('Trend')                  plt.legend()
plt.title('Actual vs Predicted')
```

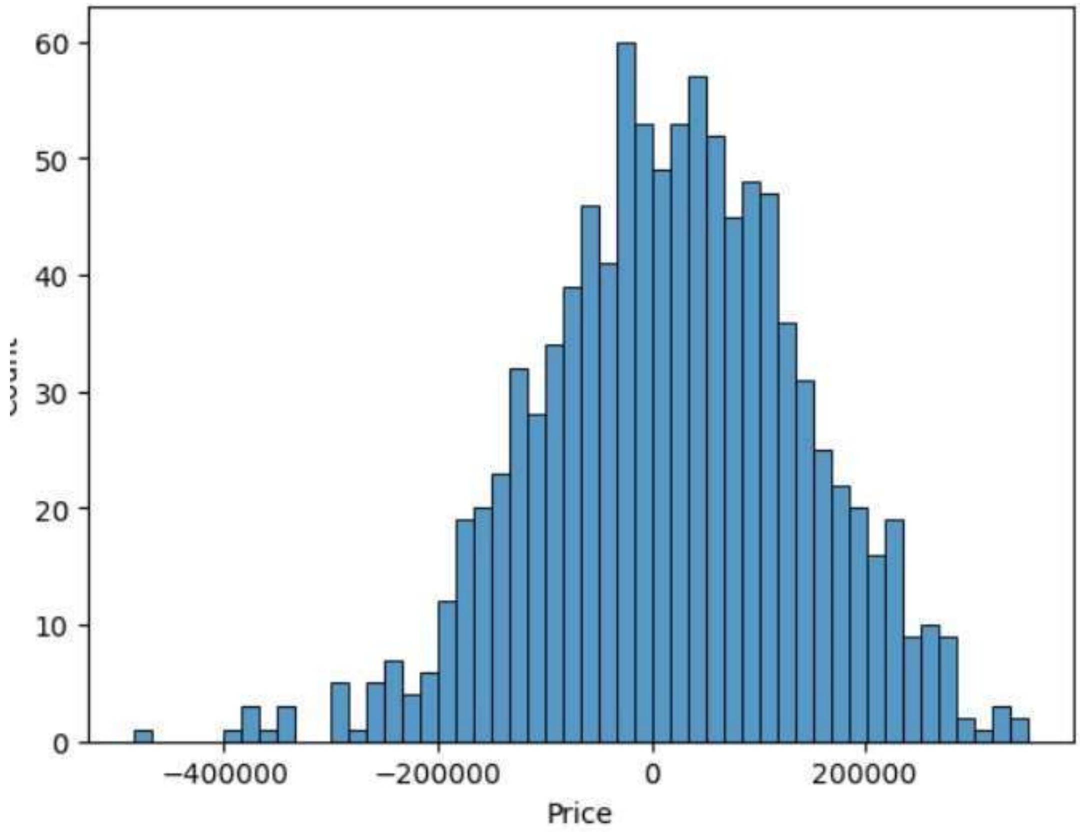
Output:

```
Text(0.5, 1.0, 'APrediction5 = model_xg.predict(X_test_scal)
```



```
print(r2_score(Y_test, Prediction2)) print(mean_absolute_error(Y_test,
Prediction2)) print(mean_squared_error(Y_test, Prediction2))

-0.0006222175925689744
286137.81086908665 128209033251.4034 sns.histplot((Y_test-Prediction4),
bins=50)
<Axes: xlabel='Price', ylabel='Count'>
```



CONCLUSION:

This project entitled “House Price Prediction Using XG Boost Regression Model.” is useful in buying the houses, by predicting house prices, and thereby to guide their buyers accordingly. The proposed system is also useful to the buyers to predict the cost of house according to the area it is present. XG boosting algorithm has high accuracy value when compared to all other algorithms regarding house price prediction.