PREDICTING HOUSE PRICE USING MACHINE LEARNING

HousePricePrediction

Introduction:

- The real estate market is a dynamic and complex arena, where property values can fluctuate significantly due to a multitude of factors.Forbothhomebuyersandsellers,accuratelydeterminingthe fair market value of a property is of paramount importance.
- In this era of technological advancement, machine learning has emergedasagame-changingtoolintherealmofrealestate.Oneof its most compelling applications is predicting house prices with remarkable accuracy.
- Traditionalmethods ofpropertyvaluation,relyingonfactorssuch as location, square footage, and recent sales data, are undoubtedly useful.However,theyoftenfallshortincapturingtheintricaciesand nuances that drive real estate market dynamics.
- Machine learning, on the other hand, has the capability to process vast volumes of data and identify patterns that human appraisers might overlook. This technology has the potential to revolutionize thewaywevaluerealestate,offeringmore preciseanddata-driven predictions.
- In this exploration, we delve into the exciting world of predicting house prices using machine learning. We will uncover how this cutting-edgetechnologyharnessesthepowerofalgorithmsanddata to create predictive models that consider an array of variables, such as neighborhood characteristics, property features, economic indicators, and even social trends.
- Bydoingso,machinelearningenablesustomakeinformed,data-backed predictions about the future value of a property.

- Thistransformationoftherealestateindustryisnotonlybeneficial for buyers and sellers but also for investors, developers, and policymakers. Accurate house price predictions can inform investment decisions, urban planning, and housing policy development, leading to a more efficient and equitable real estate market.
- Asweembarkonthisjourneyintotherealmofmachinelearningfor house price prediction, we will explore the various techniques, data sources, and challenges involved.

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

Givendata 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...

5000Rows x7Columns

Here'salistoftoolsandsoftwarecommonlyusedinthe process:

1. ProgrammingLanguage:

- Python is the most popular language for machine learning due to itsextensivelibrariesand frameworks.You can use libraries like*NumPy, pandas, scikit-learn, and more.*

2. IntegratedDevelopmentEnvironment(IDE):

- Choose an IDE for coding and running machine learning experiments.SomepopularoptionsincludeJupyterNotebook,Google Colab, or traditional IDEs like PyCharm.

3. MachineLearning Libraries:

- You'llneedvariousmachinelearninglibraries,including:
- scikit-learnforbuildingandevaluatingmachinelearningmodels.
- TensorFloworPyTorchfordeeplearning,ifneeded.
- XGBoost,LightGBM,orCatBoostforgradientboostingmodels.

4. DataVisualizationTools:

- ToolslikeMatplotlib,Seaborn,orPlotlyareessentialfordata exploration and visualization.

5. DataPreprocessingTools:

- Librarieslikepandashelpwithdatacleaning,manipulation,and preprocessing.

6. DataCollectionandStorage:

- Dependingonyourdatasource,youmightneedwebscraping tools *(e.g., BeautifulSoup or Scrapy)* or databases *(e.g., SQLite, PostgreSQL)* for data storage.

7. VersionControl:

-VersioncontrolsystemslikeGitarevaluablefortracking changes in your code and collaborating with others.

8. NotebooksandDocumentation:

- ToolsfordocumentingyourWork,suchasJupyterNotebooks or Markdown for creating *README* files and documentation.

9. HyperparameterTuning:

- ToolslikeGridSearchCVorRandomizedSearchCVfrom scikit-learn can help with hyperparameter tuning.

10. WebDevelopmentTools(forDeployment):

- Ifyouplantocreateawebapplicationformodel deployment, knowledgeofwebdevelopmenttoolslike*FlaskorDjango*forbackend development, and *HTML, CSS, and JavaScript* for the front-end can be useful.

11. CloudServices(forScalability):

-Forlarge-scaleapplications,cloudplatformslikeAWS,Google Cloud, or Azure can provide scalable computing and storage resources.

12. **ExternalDataSources(ifapplicable):**

- Dependingonyourproject'sscope,youmightrequiretoolsto access external data sources, such as APIs or data scraping tools.

13. DataAnnotationandLabelingTools(ifapplicable):

- Forspecializedprojects,toolsfordataannotationand labeling may be necessary, such as Labelbox or Supervisely.

14. GeospatialTools(forlocation-basedfeatures):

- Ifyourdatasetincludesgeospatialdata,geospatiallibraries like GeoPandas can be helpful.

1.DESIGNTHINKINGANDPRESENTINFORMOF DOCUMENT

1.Empathize:

- Understandtheneedsandchallengesofallstakeholdersinvolvedin the house price prediction process, including homebuyers, sellers, real estate professionals, appraisers, and investors.
- Conduct interviews and surveys to gather insights on what users valueinpropertyvaluationandwhatinformationismostcriticalfor their decision-making.

2.Define:

- Clearly articulate the problem statement, such as "How might we predicthousepricesmoreaccuratelyandtransparentlyusingmachine learning?"
- Identify the key goals and success criteria for the project, such as increasingpredictionaccuracy,reducingbias,orimprovingusertrust in the valuation process.

3.Ideate:

- Brainstormcreativesolutionsanddatasourcesthatcanenhancethe accuracy and transparency of house price predictions.
- Encourageinterdisciplinarycollaborationtogenerateawiderangeof ideas, including the use of alternative data, new algorithms, or improved visualization techniques.

4. Prototype:

- Createprototypemachinelearningmodelsbasedontheideas generated during the ideation phase.
- Testanditerateontheseprototypestodeterminewhichapproaches are most promising in terms of accuracy and usability.

5. Test:

- Gatherfeedbackfromusersandstakeholdersbytestingthemachine learning models with real-world data and scenarios.
- Assesshowwellthemodelsmeetthedefinedgoalsandsuccess criteria, and make adjustments based on user feedback.

6. Implement:

- Developaproduction-readymachinelearningsolutionforpredicting house prices, integrating the best-performing algorithms and data sources.
- Implementtransparencymeasures,suchasmodelinterpretability tools, to ensure users understand how predictions are generated.

7. Evaluate:

- Continuouslymonitortheperformanceofthemachinelearning model after implementation to ensure it remains accurate and relevant in a changing real estate market.
- Gatherfeedbackandinsightsfromuserstoidentifyareasfor improvement.

8. Iterate:

- Applyaniterativeapproachtorefinethemachinelearningmodel based on ongoing feedback and changing user needs.
- Continuouslyseekwaystoenhancepredictionaccuracy, transparency, and user satisfaction.

9. ScaleandDeploy:

- Oncethemachinelearningmodelhasbeenoptimizedandvalidated, deploy it at scale to serve a broader audience, such as real estate professionals, investors, and homeowners.
- Ensurethemodelisaccessiblethroughuser-friendlyinterfacesand integrates seamlessly into real estate workflows.

10. EducateandTrain:

- Providetrainingandeducationalresourcestohelpusers understand how the machine learning model works, what factors it considers, and its limitations.
- Fosteracultureofdataliteracyamongstakeholderstoenhancetrust in the technology.

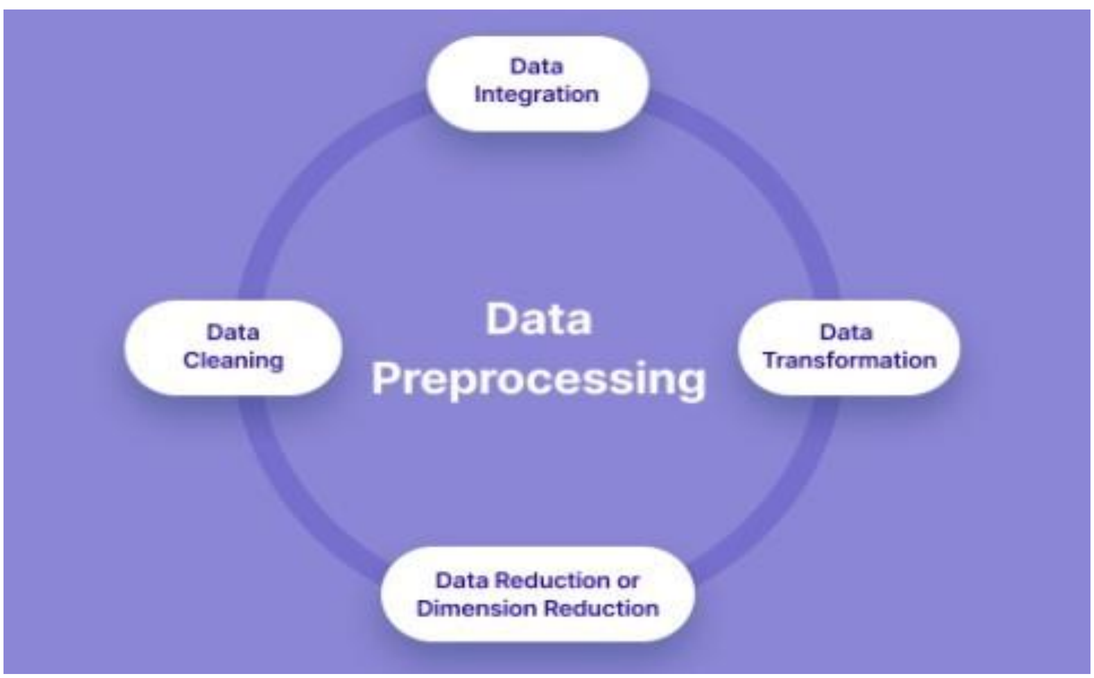
2.DESIGNINTO INNOVATION

1. DataCollection:

Gatheracomprehensivedatasetthatincludesfeaturessuchas location, size, age, amenities, nearby schools, crime rates, and other relevant variables.

2. DataPreprocessing:

Clean the data by handling missing values, outliers, and encodingcategoricalvariables.Standardizeornormalizenumerical features as necessary.



PYHONPROGRAM:

```
#Importnecessarylibraries
importpandasaspd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
fromsklearn.preprocessingimportStandardScaler

#Loadthedataset(replace'house_data.csv'withyourdatasetfile)
data=pd.read_csv('E:/USA_Housing.csv')
#Display thefirstfewrowsofthedatasettogetanoverview
print("Dataset Preview:")
print(data.head())


                #DataPre-processing
#HandleMissingValues
# Let's fill missing values in numeric columns with the mean and in
categorical columns with the most frequent value.
numeric_cols = data.select_dtypes(include='number').columns
categorical_cols = data.select_dtypes(exclude='number').columns

imputer_numeric = SimpleImputer(strategy='mean')
imputer_categorical = SimpleImputer(strategy='most_frequent')
```

```
data[numeric_cols] =
imputer_numeric.fit_transform(data[numeric_cols])
data[categorical_cols] =
imputer_categorical.fit_transform(data[categorical_cols])

#ConvertCategoricalFeaturestoNumerical
# We'll use Label Encoding for simplicity here. You can also use one-
hot encoding for nominal categorical features.
label_encoder = LabelEncoder()
for col in categorical_cols:
    data[col]=label_encoder.fit_transform(data[col])

# Split Data into Features (X) and Target (y)
X = data.drop(columns=['Price'])# Features y
= data['Price']# Target

#NormalizetheData
scaler = StandardScaler()X_scaled
= scaler.fit_transform(X)
```

```
# Split data into training and testing sets (adjust test_size as needed)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,
test_size=0.2, random_state=42)

#Displaythepreprocesseddata
print("\nPreprocessedData:")
print(X_train[:5])# Display first 5 rows of preprocessed features
print(y_train[:5])# Display first 5 rows of target values
```

OUTPUT:

DatasetPreview:

```
   Avg.AreaIncomeAvg.AreaHouseAgeAvg.AreaNumberofRoo
ms  \
0     79545.458574          5.682861                 7.009188
1     79248.642455          6.002900                 6.730821
2     61287.067179          5.865890                 8.512727
3     63345.240046          7.188236                 5.586729
4     59982.197226          5.040555                 7.839388

   Avg.AreaNumberofBedroomsAreaPopulation              Price\
0                        4.09     23086.800503   1.059034e+06
1                        3.09     40173.072174   1.505891e+06
2                        5.13     36882.159400   1.058988e+06
```

3 3.26 34310.2428311.260617e+06

4 4.23 26354.1094726.309435e+05

Address

0 208MichaelFerryApt.674\nLaurabury,NE3701...

1 188JohnsonViewsSuite079\nLakeKathleen,CA...

2 9127ElizabethStravenue\nDanieltown,WI06482...

3 USSBarnett\nFPOAP44820

4 USNSRaymond\nFPOAE09386

PreprocessedData:

[[-0.19105816-0.13226994-0.139692930.12047677-0.83757985-1.0 0562872]

[-1.39450169 0.427867360.79541275-0.552125091.157290181.61 946754]

[-0.35137865 0.463944891.701995090.03133676-0.326712131.63 886651]

[-0.13944143 0.1104872 0.22289331-0.75471601-0.90401197-1.54 810704]

[0.62516685 2.209696660.42984356-0.454881440.125662160.98 830821]]

4227 1.094880e+06

4676 1.300389e+06

800 1.382172e+06

3671 1.027428e+06

4193 1.562887e+06

Name:Price,dtype:float64

3. **FeatureEngineering:**

Create new features or transform existing ones to extract more valuable information. For example, you can calculate the distance to the nearestpublictransportation,orcreateafeaturefortheoverallcondition of the house.

4. **ModelSelection:**

Choose the appropriate machine learning model for the task. Common models for regression problems like house price prediction include*LinearRegression,DecisionTrees,RandomForest,Gradient Boosting, and Neural Networks.*

5. **Training:**

Splitthedatasetintotrainingandtestingsetstoevaluatethe model's performance. Consider techniques like cross-validation to prevent overfitting.

6. **HyperparameterTuning:**

Optimizethemodel'shyperparameterstoimproveitspredictive accuracy. Techniques like grid search or random search can help with this.

7. EvaluationMetrics:

Selectappropriateevaluationmetricsforregressiontasks,such as *Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE)*. Choose the metric that aligns with the specific objectives of your project.

8. Regularization:

ApplyregularizationtechniqueslikeL1(Lasso)orL2(Ridge) regularization to prevent overfitting.

9. FeatureSelection:

Usetechniqueslikefeatureimportancescoresorrecursive feature elimination to identify the most relevant features for the prediction.

10. Interpretability:

Ensure that the model's predictions are interpretable and explainable. This is especially important for real estate applications wherestakeholderswanttounderstandthefactorsaffectingpredictions.

11. Deployment:

Developauser-friendlyinterfaceorAPIforend-userstoinput property details and receive price predictions.

12. ContinuousImprovement:

Implementafeedbackloopforcontinuousmodelimprovement based on user feedback and new data.

13. EthicalConsiderations:

Bemindfulofpotentialbiasesinthedataandmodel.Ensure fairness and transparency in your predictions.

14. Monitoringand Maintenance:

Regularlymonitorthemodel'sperformanceintherealworldand update it as needed.

15. Innovation:

Considerinnovativeapproachessuchasusingsatelliteimageryor IoT data for real-time property condition monitoring, or integrating natural language processing for textual property descriptions.

3.BUILDLOADINGANDPREPROCESSINGTHEDATASET

1. **DataCollection:**

Obtainadatasetthatcontainsinformationabouthousesand their correspondingprices. This dataset can be obtainedfrom sources like real estate websites, government records, or other reliable data providers.

2. LoadtheDataset:

- Importrelevantlibraries,suchaspandasfordatamanipulationand numpy for numerical operations.
- LoadthedatasetintoapandasDataFrameforeasydatahandling. You can use *pd.read_csv()* for CSV files or other appropriate functions for different file formats.

Program:

```
import pandas as pd

import numpy as np

importseabornassns

importmatplotlib.pyplotasplt

fromsklearn.model_selectionimporttrain_test_split

from sklearn.preprocessing import StandardScaler
```

from sklearn.metrics import r2_score,
mean_absolute_error,mean_squared_error

fromsklearn.linear_modelimportLinearRegression

from sklearn.linear_model import Lasso

fromsklearn.ensembleimportRandomForestRegressor from

sklearn.svm import SVR

importxgboostasxg

%matplotlibinline

import warnings

warnings.filterwarnings("ignore")

/opt/conda/lib/python3.10/site-packages/scipy/___init__.py:146:
UserWarning:ANumPyversion>=1.16.5and<1.23.0isrequiredfor this
version of SciPy (detected version 1.23.5

warnings.warn(f"ANumPyversion>={np_minversion}and
<{np_maxversion}"

LoadingDataset:

dataset=pd.read_csv('E:/USA_Housing.csv')

Output:

	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 ?

3. DataExploration:

Explore the dataset to understand its structure and contents. Checkforthepresenceofmissingvalues,outliers,anddatatypes of each feature.

4. DataCleaning:

Handlemissingvaluesbyeitherremovingrowswithmissing data or imputing values based on the nature of the data.

5. FeatureSelection:

Identifyrelevantfeaturesforhousepriceprediction.Featureslike the number of bedrooms, square footage, location, and amenities are often important.

We are selecting numerical features which have more than 0.50 or less than -0.50 correlation rate based on Pearson Correlation Method—which is the defaultvalueof parameter "method" in corr()function. Asfor selectingcategoricalfeatures,I selectedthe categoricalvalueswhichIbelievehave significanteffect on the target variable such as Heating and MSZoning.

In[1]:

```
important_num_cols=list(df.corr()["SalePrice"][(df.corr()["SalePrice"]>0.5
0)|(df.corr()["SalePrice"]<-0.50)].index)

cat_cols=["MSZoning","Utilities","BldgType","Heating","KitchenQual","
SaleCondition","LandSlope"]

important_cols=important_num_cols+cat_cols

df= df[important_cols]
```

Checkingforthemissingvalues

In[2]:

```
print("MissingValues byColumn")

print("-"*30)

print(df.isna().sum())
```

```
print("-"*30)

print("TOTALMISSINGVALUES:",df.isna().sum().sum())
```
Missing Values by Column

OverallQual 0

YearBuilt 0

YearRemodAdd 0

TotalBsmtSF 0

1stFlrSF 0

GrLivArea 0

FullBath 0

TotRmsAbvGrd 0

GarageCars 0

GarageArea 0

SalePrice 0

MSZoning 0

Utilities 0

BldgType 0

Heating 0

KitchenQual 0

SaleCondition   0

LandSlope   0

dtype: int64

TOTALMISSINGVALUES: 0

6. FeatureEngineering:

Create new features or transform existing ones to capture additionalinformationthatmayimpacthouseprices.*Forexample,you can calculate the price per square foot.*

7. DataEncoding:

Convertcategoricalvariables*(e.g.,location)*intonumerical format using techniques like one-hot encoding.

8. Train-TestSplit:

Splitthedatasetintotrainingandtestingsetstoevaluatethe machine learning model's performance.

Program:

```
X=df.drop('price',axis=1)#Features y =

df['price']# Target variable

X_train,X_test,y_train,y_test=train_test_split(X,y,
test_size=0.2, random_state=42)
```

4.PERFORMINGDIFFERENTACTIVITIESLIKEFEATURE ENGINEERING, MODEL TRAINING, EVALUATIONetc.,

1. FeatureEngineering:

- Asmentionedearlier,featureengineeringiscrucial.Itinvolves creating new features or transforming existing ones to provide meaningful information for your model.
- Extractinginformationfromtextualdescriptions*(e.g.,presenceof keywords like "pool" or "granite countertops").*
- Calculatingdistancestokeylocations*(e.g.,schools,parks)*ifyou have location data.

2. DataPreprocessing&Visualisation:

Continue data preprocessing by handling any remaining missingvaluesoroutliersbasedoninsights fromyourdataexploration.

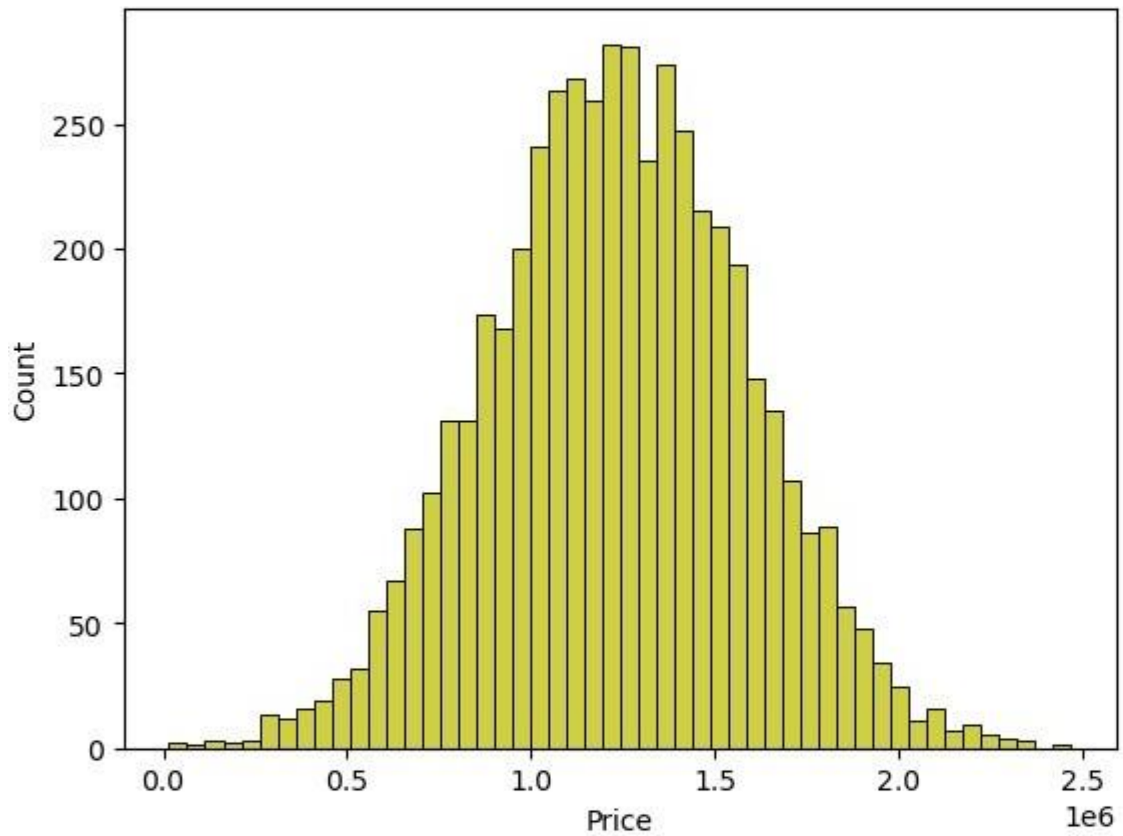
VisualisationandPre-ProcessingofData:

In[1]:

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

Out[1]:

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

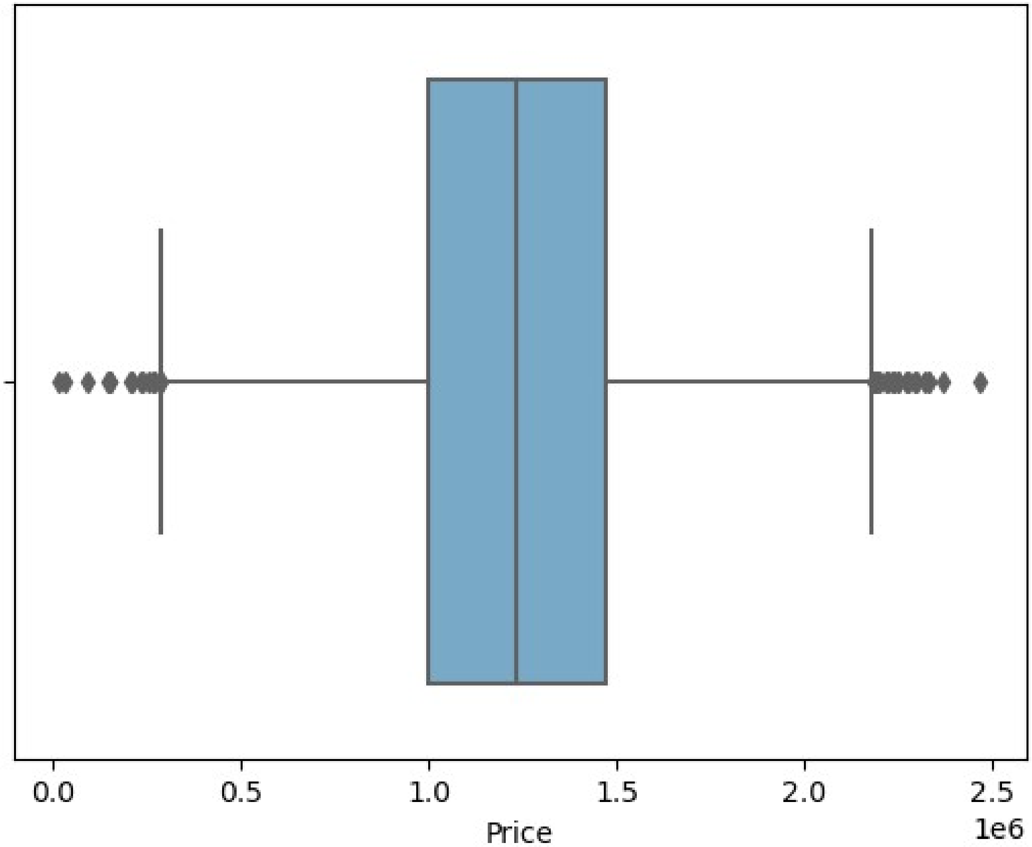


In[2]:

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

Out[2]:

<Axes:xlabel='Price'>

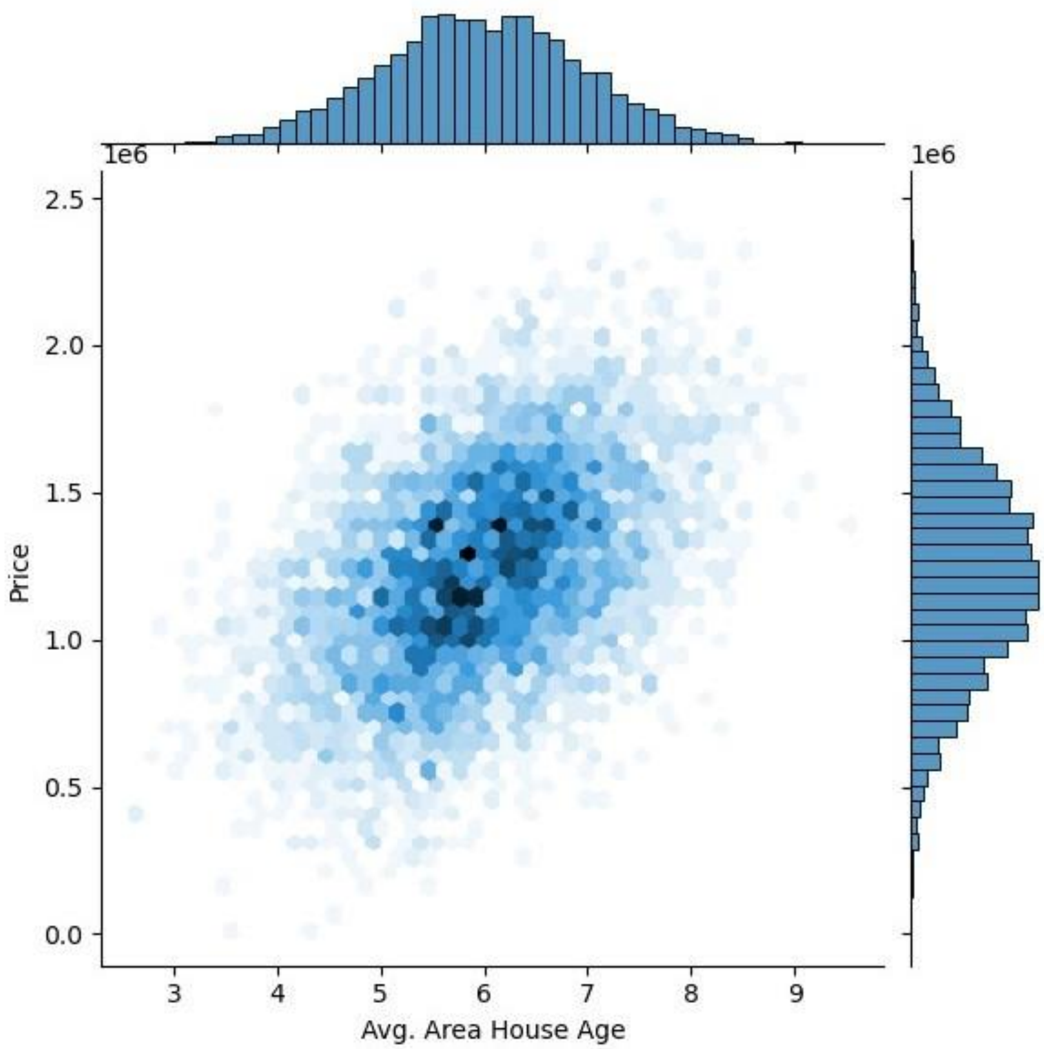


In[3]:

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

Out[3]:

<seaborn.axisgrid.JointGridat0x7caf1d571810>

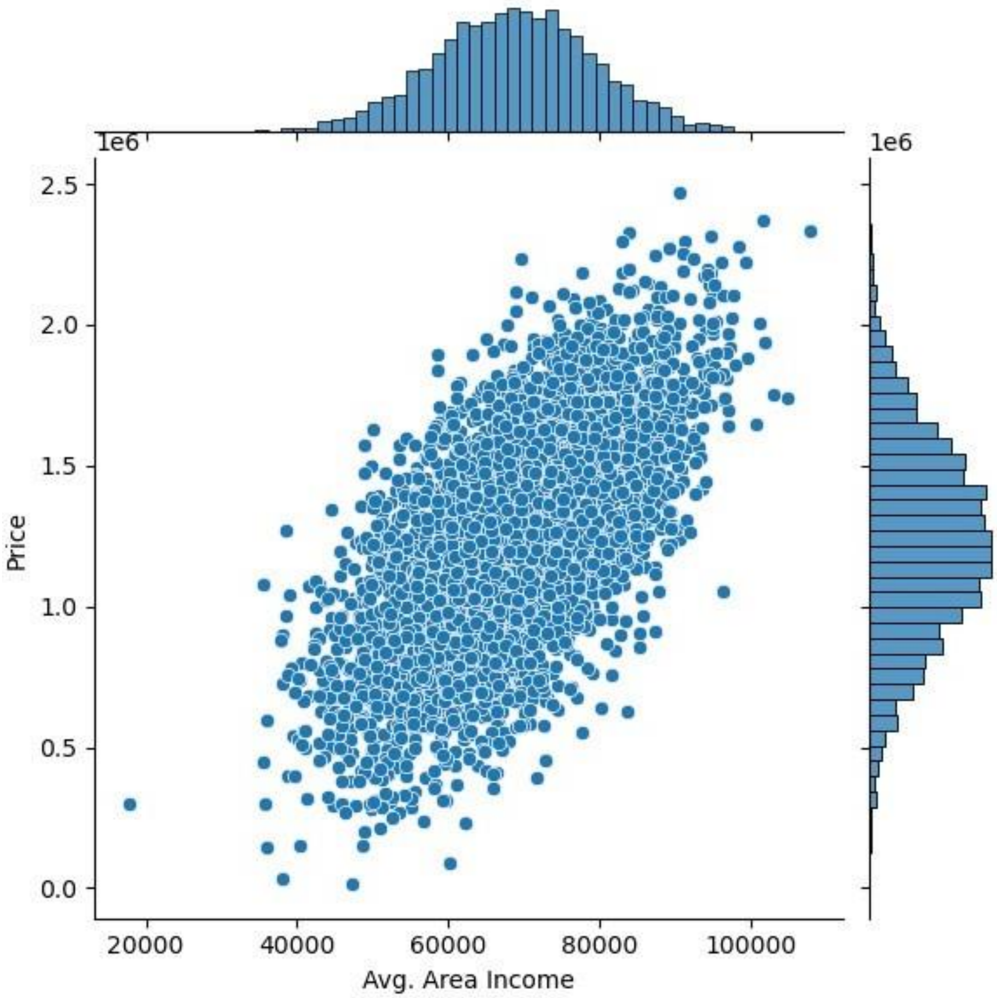


In[4]:

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

Out[4]:

<seaborn.axisgrid.JointGridat0x7caf1d8bf7f0>



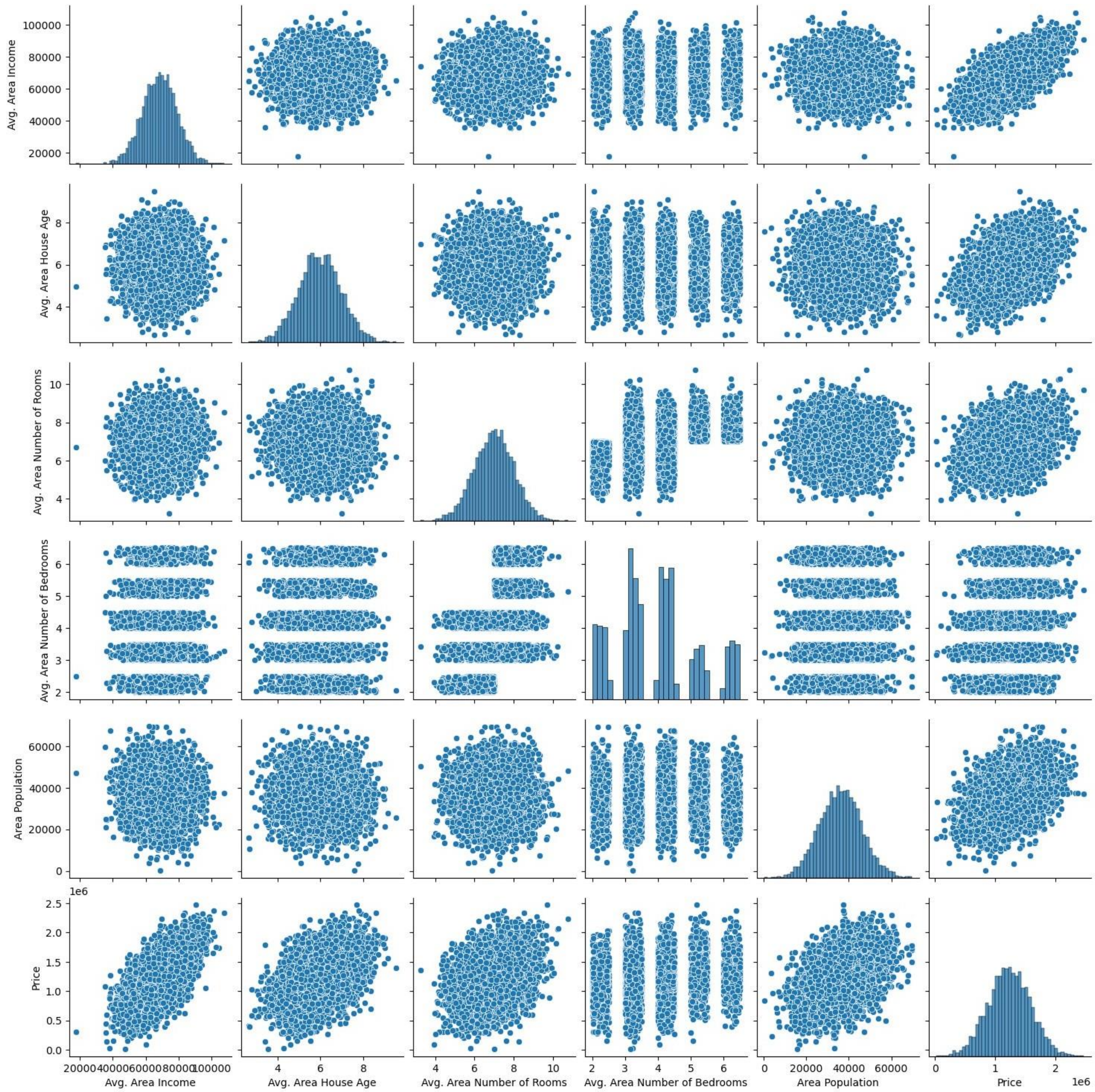
In[5]:

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

Out[5]:

<seaborn.axisgrid.PairGridat0x7caf0c2ac550>

<Figuresize1200x800with0Axes>



In[6]:

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

Out[6]:

array([[<Axes:title={'center':'Avg.AreaIncome'}>,

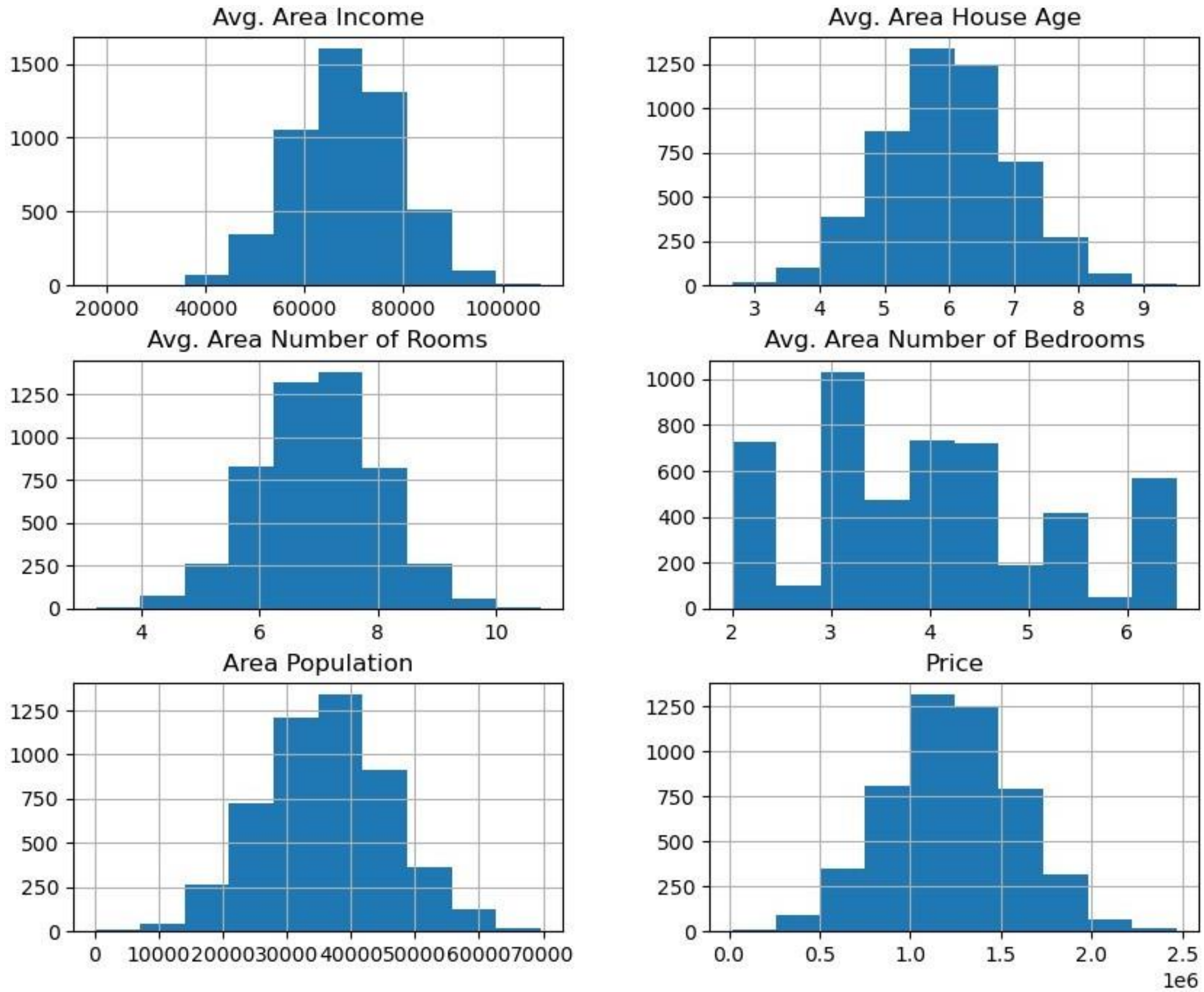
<Axes: title={'center': 'Avg. Area House Age'}>],

[<Axes:title={'center':'Avg.AreaNumberofRooms'}>,

<Axes:title={'center':'Avg.AreaNumberofBedrooms'}>], [<Axes:

title={'center': 'Area Population'}>,

<Axes:title={'center':'Price'}>]],dtype=object)



VisualisingCorrelation:

In[7]:

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

Out[7]:

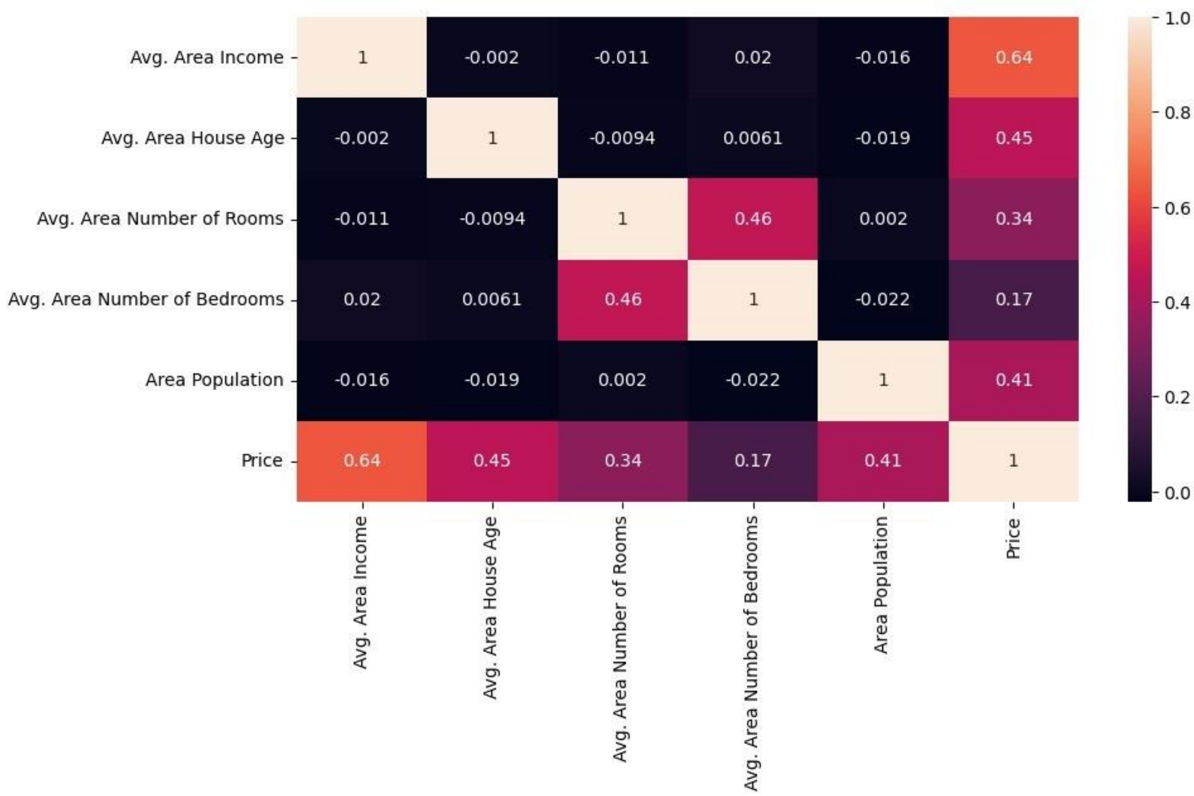
	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg.Area Numberof Bedrooms	Area Population	Price
Avg.Area Income	1.000000	-0.002007	-0.011032	0.019788	-0.016234	0.639734
Avg.Area House Age	-0.002007	1.000000	-0.009428	0.006149	-0.018743	0.452543
Avg. Area Numberof Rooms	-0.011032	-0.009428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

In[8]:

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

Out[8]:

<Axes:>



3. ModelSelection:

Chooseanappropriatemachinelearningmodelforyour regression task. *Common choices include:*

- ✓ LinearRegression
- ✓ DecisionTrees
- ✓ RandomForest
- ✓ GradientBoosting*(e.g.,XGBoostorLightGBM)*
- ✓ NeuralNetworks(DeepLearning)

Program:

ImportingDependencies

import pandas as pd

import numpy as np

import seaborn as sns

importmatplotlib.pyplotasplt

fromsklearn.model_selectionimporttrain_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import r2_score,
mean_absolute_error,mean_squared_error

fromsklearn.linear_modelimportLinearRegression

from sklearn.linear_model import Lasso

```
fromsklearn.ensembleimportRandomForestRegressor

from sklearn.svm import SVR

importxgboostasxg

%matplotlibinline


import warnings

warnings.filterwarnings("ignore")

/opt/conda/lib/python3.10/site-packages/scipy/_init_.py:146: UserWarning:ANumPyversion>=1.16.5and<1.23.0 isrequired for this version of SciPy (detected version 1.23.5

  warnings.warn(f"ANumPyversion>={np_minversion}and <{np_maxversion}"
```

LoadingDataset

```
dataset=pd.read_csv('E:/USA_Housing.csv')
```

Model1 -LinearRegression

In[1]:

```
model_lr=LinearRegression()
```

In[2]:

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

Out[2]:

```
▾ LinearRegression
LinearRegression()
```

PredictingPrices

In[3]:

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

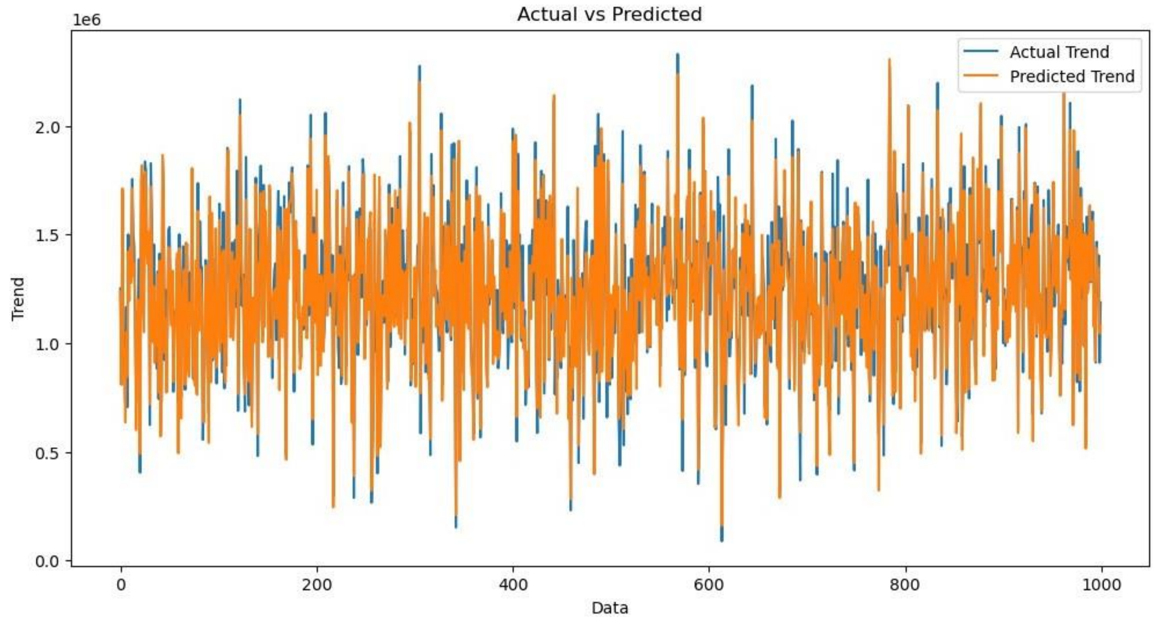
EvaluationofPredictedData

In[4]:

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

Out[4]:

Text(0.5,1.0,'ActualvsPredicted')

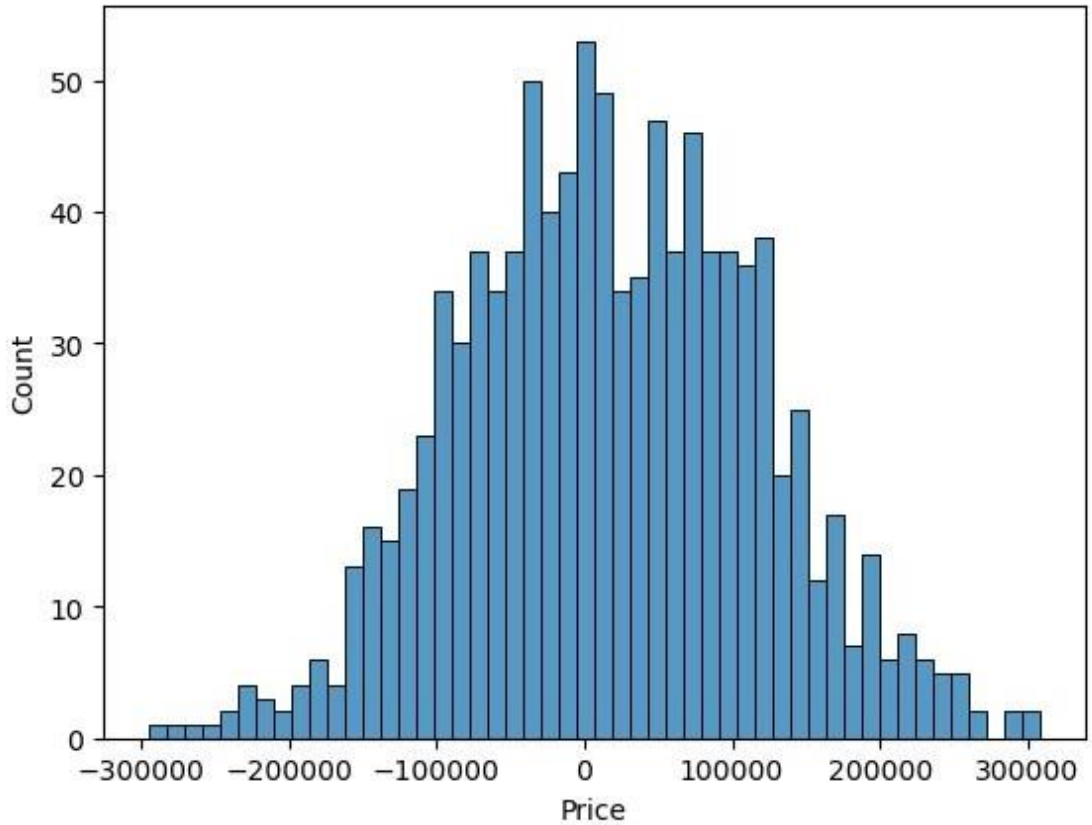


In[5]:

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

Out[5]:

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



In[6]:

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

Out[6]:

0.9182928179392918

82295.49779231755

10469084772.975954

Model2 -SupportVectorRegressor

In[7]:

```
model_svr=SVR()
```

In[8]:

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

Out[8]:

PredictingPrices

In[9]:

```
Prediction2=model_svr.predict(X_test_scal)
```

EvaluationofPredictedData

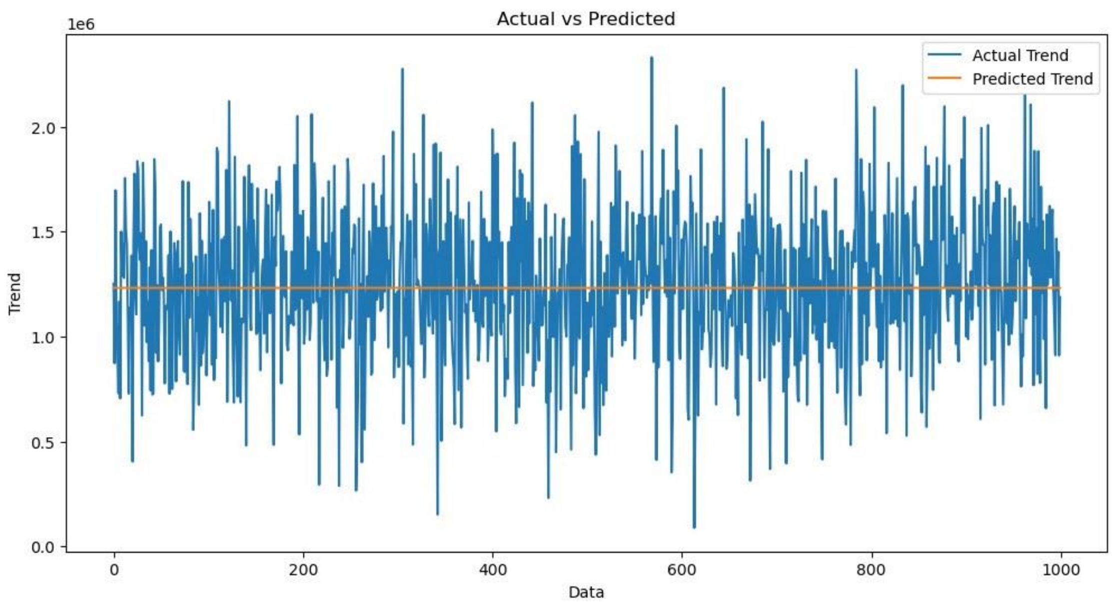
In[10]:

```
plt.figure(figsize=(12,6))
plt.plot(np.arange(len(Y_test)),Y_test,label='ActualTrend')
```

```
plt.plot(np.arange(len(Y_test)),Prediction2,label='PredictedTrend')
plt.xlabel('Data')
plt.ylabel('Trend')
plt.legend()
plt.title('ActualvsPredicted')
```

Out[10]:

Text(0.5,1.0,'ActualvsPredicted')

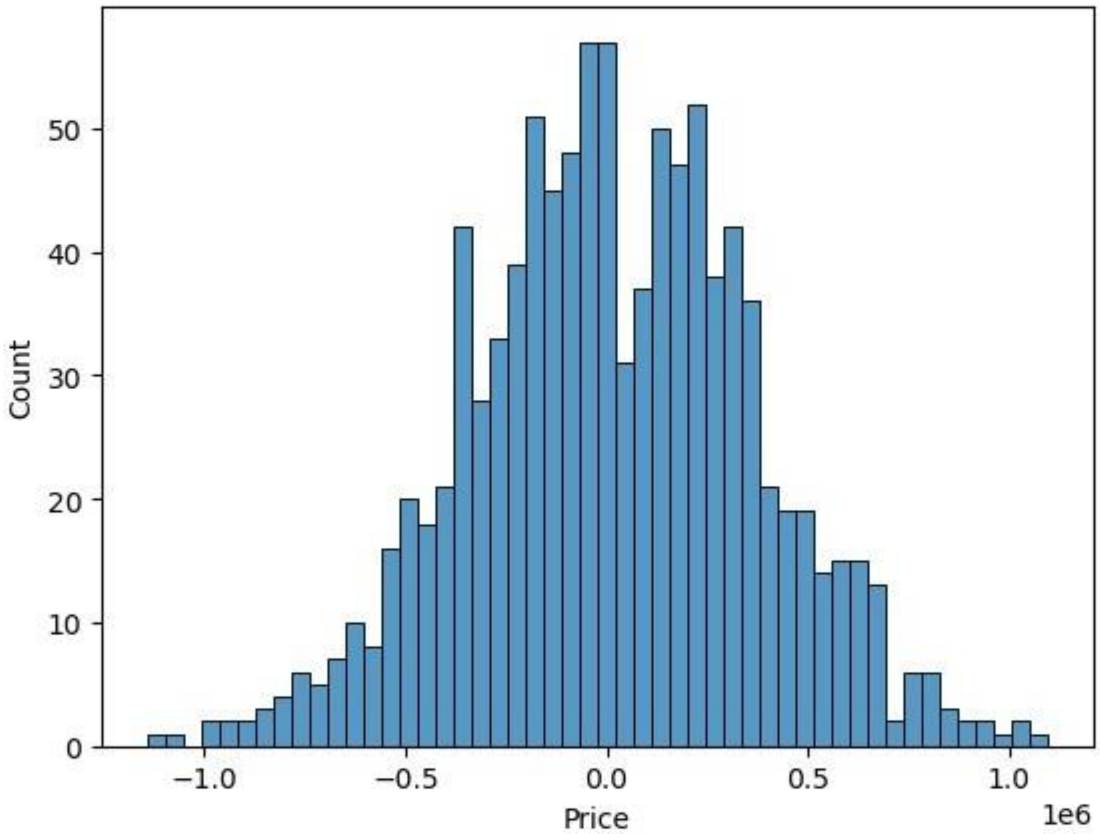


In[11]:

```
sns.histplot((Y_test-Prediction2),bins=50)
```

Out[12]:

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



In[12]:

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

-0.0006222175925689744

286137.81086908665

128209033251.4034

Model3 -LassoRegression

In[13]:

```
model_lar=Lasso(alpha=1)
```

In[14]:

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

Out[14]:

```
▾ Lasso
Lasso(alpha=1)
```

PredictingPrices

In[15]:

```
Prediction3=model_lar.predict(X_test_scal)
```

EvaluationofPredictedData

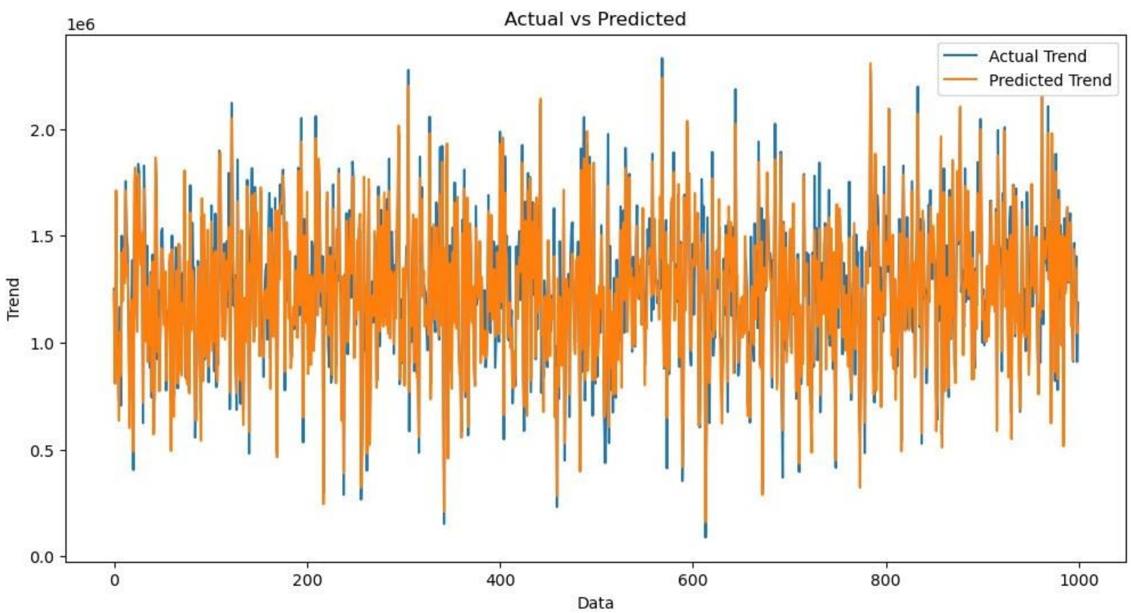
In[16]:

```
plt.figure(figsize=(12,6))
plt.plot(np.arange(len(Y_test)),Y_test,label='ActualTrend')
```

```
plt.plot(np.arange(len(Y_test)),Prediction3,label='PredictedTrend')

plt.xlabel('Data')

plt.ylabel('Trend')

plt.legend()

plt.title('ActualvsPredicted')
```

Out[16]:

Text(0.5,1.0,'ActualvsPredicted')

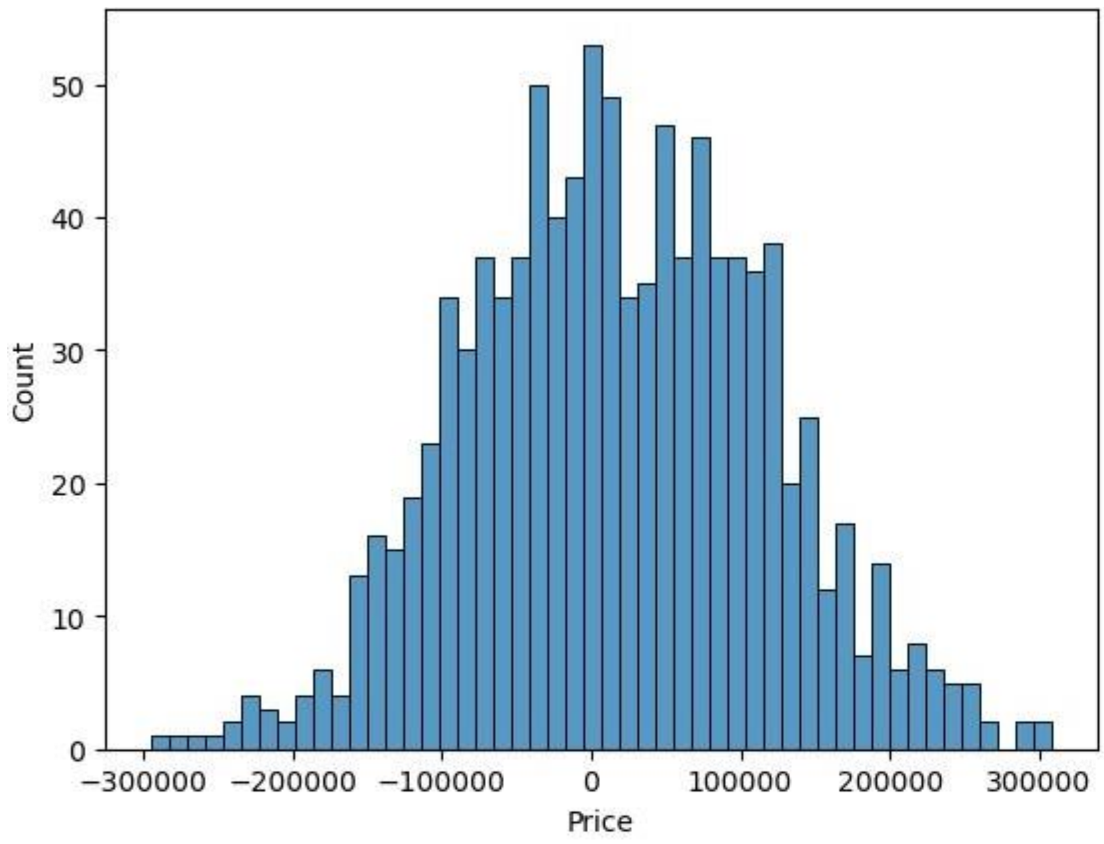


In[17]:

```
sns.histplot((Y_test-Prediction3),bins=50)
```

Out[17]:

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



In[18]:

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

-0.0006222175925689744

286137.81086908665

128209033251.4034

Model4-RandomForestRegressor

In[19]:

```
model_rf=RandomForestRegressor(n_estimators=50)
```

In[20]:

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

Out[20]:

```
RandomForestRegressor
RandomForestRegressor(n_estimators=50)
```

PredictingPrices

In[21]:

```
Prediction4=model_rf.predict(X_test_scal)
```

Model5-XGboostRegressor

In[25]:

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

In[26]:

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

Out[26]:

XGBRegressor

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,...)

4. **ModelTraining:**

Splityourdatasetintotrainingandtestingsets(asshownearlier) and train the selected model on the training data. Here's an example using Linear Regression:

5. ModelEvaluation:

Evaluateyourmodel'sperformanceusingappropriateregression metrics, such as *Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE)*. For example:

PYTHONPROGRAM:

```
#Importnecessarylibraries

fromsklearn.feature_selectionimportSelectKBest,f_regression

from sklearn.linear_model import LinearRegression

from sklearn.ensemble import RandomForestRegressor

fromsklearn.metricsimportmean_squared_error,r2_score

import numpy as np

selector = SelectKBest(score_func=f_regression, k=k)

X_train_selected=selector.fit_transform(X_train,y_train)
```

#ModelSelection

#Let'schoosebothLinearRegressionandRandomForestRegressorfor comparison.

linear_reg_model=LinearRegression()

random_forest_model=RandomForestRegressor(n_estimators=100, random_state=42)

 # Train the models on the selected features

linear_reg_model.fit(X_train_selected, y_train)

random_forest_model.fit(X_train_selected,y_train)

 # Evaluate the models on the test set

X_test_selected = selector.transform(X_test)

#Makepredictions

linear_reg_predictions=linear_reg_model.predict(X_test_selected)

random_forest_predictions = random_forest_model.predict(X_test_selected)

#Evaluatemodelperformance

defevaluate_model(predictions,model_name):

```
mse=mean_squared_error(y_test,predictions)

r2 = r2_score(y_test, predictions)

print(f"{model_name} Model Evaluation:")

print(f"Mean Squared Error (MSE): {mse}")

print(f"R-squared (R2) Score: {r2}\n")

#PerformanceMeasure

elr_mse=mean_squared_error(y_test,pred)

elr_rmse = np.sqrt(lr_mse)

elr_r2=r2_score(y_test,pred)

#ShowMeasures

result = '''

MSE:{}

RMSE:{}

R^2:{}

'''.format(lr_mse,lr_rmse,lr_r2)

print(result)

#ModelComparision
```

```
    names.append("elr")

    mses.append(elr_mse)

    rmses.append(elr_rmse)

    r2s.append(elr_r2)

evaluate_model(linear_reg_predictions, "Linear Regression")

evaluate_model(random_forest_predictions,"RandomForestRegressor")
```

OUTPUT:

LinearRegressionModelEvaluation:

MeanSquaredError(MSE):10089009300.893988

R-squared (R2) Score: 0.9179971706834331

Random Forest Regressor Model Evaluation:

MeanSquaredError(MSE):14463028828.265167

R-squared (R2) Score: 0.8824454166872736

MSE:10141766848.330585

RMSE:100706.33966305491

R^2:0.913302484308253

ModelComparison:

ThelesstheRootMean SquaredError(RMSE),Thebetterthe model is.

In[30]:

```
models.sort_values(by="RMSE(Cross-Validation)")
```

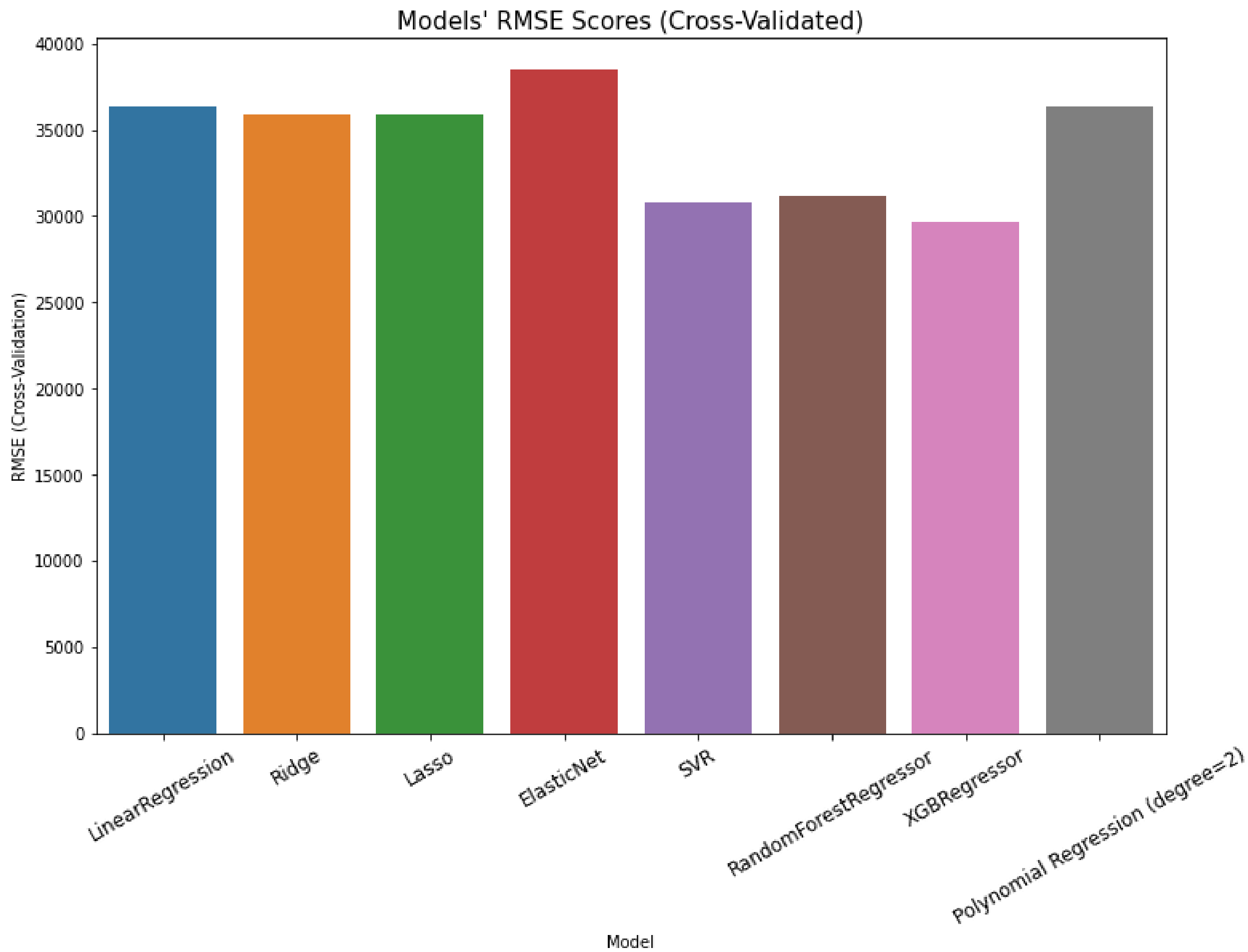
Out[30]:

	Model	MAE	MSE	RMSE	R2Score	RMSE (Cross-Validation)
6	XGBRegressor	1.743992 e+04	7.165790 e+08	2.676899 e+04	9.065778 e-01	29698.849618
4	SVR	1.784316 e+04	1.132136 e+09	3.364723 e+04	8.524005 e-01	30745.475239
5	RandomForestRegressor	1.811511 e+04	1.004422 e+09	3.169262 e+04	8.690509 e-01	31138.863315
1	Ridge	2.343550 e+04	1.404264 e+09	3.747351 e+04	8.169225 e-01	35887.852792
2	Lasso	2.356046 e+04	1.414338 e+09	3.760768 e+04	8.156092 e-01	35922.769369
0	LinearRegression	2.356789 e+04	1.414931 e+09	3.761557 e+04	8.155318 e-01	36326.451445
7	Polynomial Regression (degree=2)	2.382228 e+15	1.513991 e+32	1.230443 e+16	-1.973829 e+22	36326.451445

	Model	MAE	MSE	RMSE	R2Score	RMSE (Cross-Validation)
3	ElasticNet	2.379274e+04	1.718446e+09	4.145414e+04	7.759618e-01	38449.008646

In[31]:

```
plt.figure(figsize=(12,8))

sns.barplot(x=models["Model"],y=models["RMSE(Cross-Validation)
"])

plt.title("Models'RMSEScores(Cross-Validated)",size=15)

plt.xticks(rotation=30,size=12)plt.

show()
```



EvaluationofPredictedData

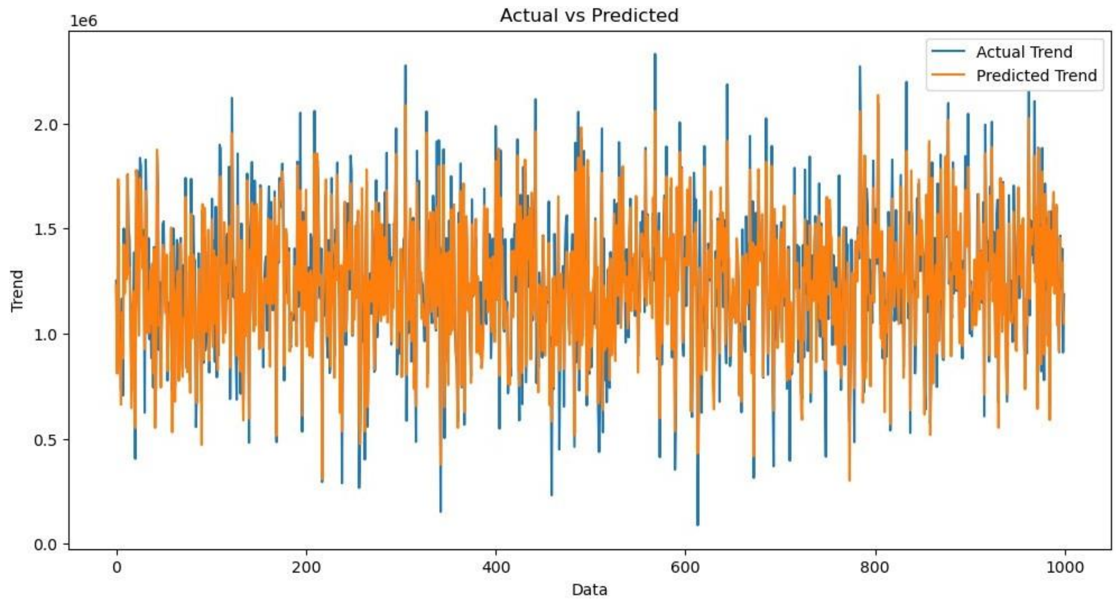
In[22]:

```
plt.figure(figsize=(12,6))
plt.plot(np.arange(len(Y_test)),Y_test,label='ActualTrend')
plt.plot(np.arange(len(Y_test)),Prediction4,label='PredictedTrend')
plt.xlabel('Data')
```

```
plt.ylabel('Trend')

plt.legend()

plt.title('ActualvsPredicted')
```

Out[22]:

Text(0.5,1.0,'ActualvsPredicted')

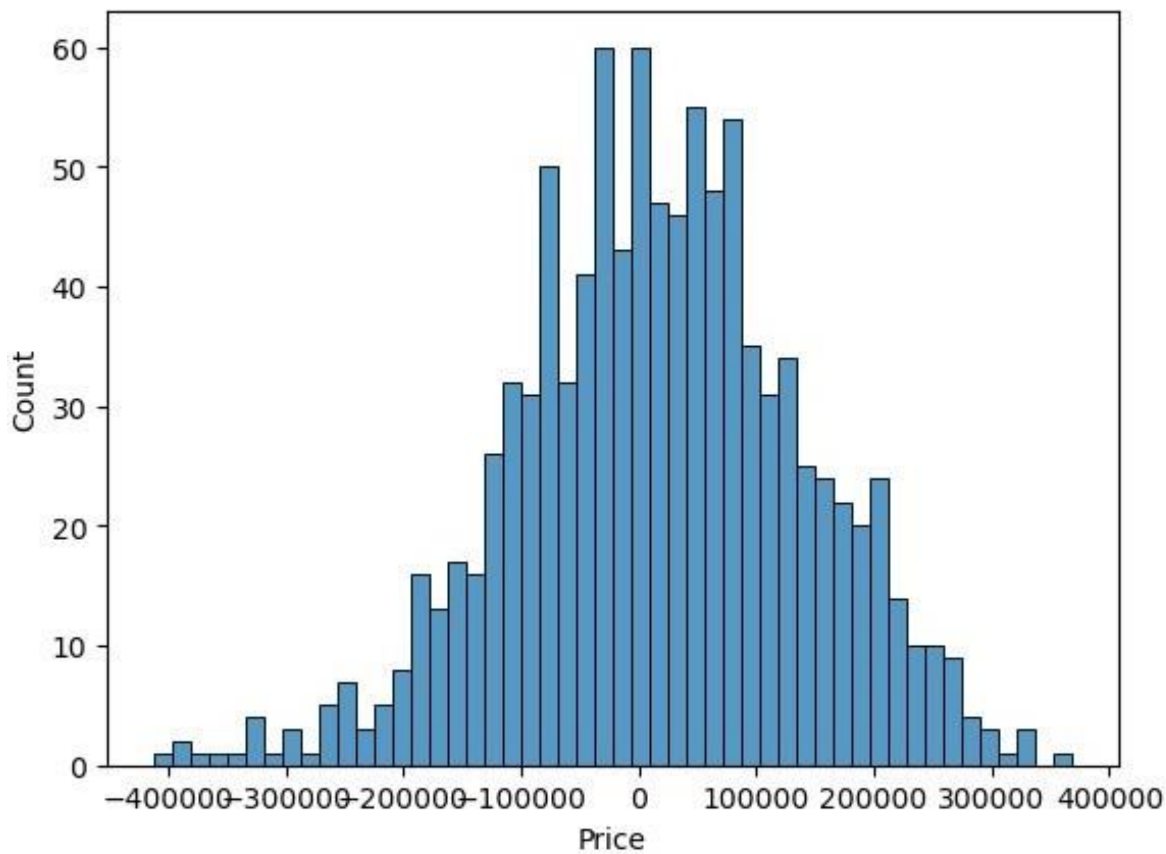


In[23]:

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

Out[23]:

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



In[24]:

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

Out[24]:

-0.0006222175925689744

286137.81086908665

128209033251.4034

6. **HyperparameterTuning:**

Optimize the model's hyperparameters to improve its performance.Dependingonthemodel,youcanusetechniqueslikegrid search or random search.

7. **Cross-Validation:**

Implement cross-validation to ensure that your model's performanceisconsistentacrossdifferentsubsetsofyourdata.This helps prevent overfitting.

8. **Regularization:**

ApplyregularizationtechniqueslikeL1(Lasso)orL2(Ridge) if needed to prevent overfitting and improve model generalization.

FeatureSelection:

Use feature importance scores from your model or techniques likerecursivefeatureeliminationtoidentifythemostimportantfeatures for predictions.

Interpretability:

Ensure that the model's predictions are interpretable and explainable.Stakeholdersmaywanttounderstandhoweachfeature impacts the predicted house price.

Deployment:

Deployyourtrainedmodelinareal-worldsetting,whetherit's through a web application, API, or any other user-friendly interface. Userscaninputpropertydetails,andthemodelprovidesprice predictions.

Monitoringand Maintenance:

Continuouslymonitorthemodel'sperformanceandupdateitas needed.Realestatemarkets change,so it's essential to retrain the model with new data periodically.

EthicalConsiderations:

Ensurethatyourmodeldoesn'tintroduceorperpetuatebiases in pricing. Implement fairness and transparency measures.

Innovation:

Exploreinnovativeapproachessuchasincorporatingexternal data sources *(e.g., satellite imagery, IoT data)* for better predictions.

ADVANTAGES:

Predictinghousepricesusingmachinelearningofferasseveral significant advantages:

1. **Accuracy:**

Machinelearningmodelscanprocessandanalyzevastamounts of data, including various property and market factors. This results in more accurate house price predictions compared to traditional methods that rely on a limited set of variables.

2. **ComplexData Handling:**

Machine learning algorithms can handle complex, non-linear relationships in the data. They can recognize patterns and interactions amongdifferentfeatures,allowingforamorecomprehensiveassessment of a property's value.

3. **ContinuousLearning:**

Machinelearningmodelscanbecontinuallyupdatedwithnew data, enabling them to adapt to changing market conditions and trends. Thisensuresthatpredictionsremainrelevantandup-to-date.

4. **Efficiency:**

Automated valuation models powered by machine learning can evaluate properties rapidly. This efficiency is beneficial for both propertyappraisersandindividualslookingtodeterminethevalueofa property quickly.

5. **DataIntegration:**

Machine learning models can incorporate a wide range of data sources, including property characteristics, neighborhood information, economic indicators, and even social trends. This holistic approach providesamorecompletepictureofthefactorsinfluencinghouseprices.

6. **ReducedBias:**

Machine learning can help reduce human bias in property valuation.Itevaluatespropertiesobjectivelybasedondata,whichcan lead to fairer and more consistent pricing.

7. **MarketInsights:**

Byanalyzing historicaldataand current marketconditions, machinelearningcanoffervaluableinsightsintomarkettrends,helping investors and developers make informed decisions.

8. **RiskAssessment:**

Machine learning can assess the risk associated with a property,whichiscrucialformortgagelendersandinvestors.Ithelps identifypotential issues or opportunitiesrelated to a property'svalue.

9. **Transparency:**

Machinelearningmodelscanprovideclearandtransparent explanations for their predictions, which is essential for building trust among stakeholders in the real estate market.

10. Scalability:

Machinelearningmodelscanbedeployedatscale,makingit possible to assess property values in large real estate portfolios, entire neighborhoods, or even across entire cities.

11. TimeandCostSavings:

Using machine learning for property valuation can save time and reduce costs associated with manual appraisals, making it an efficientandcost-effectivesolutionforbothbusinessesandindividuals.

12. Customization:

Machine learning models can be customized to cater to specificmarkets,types ofproperties,orregionalvariations,allowingfor more tailored and precise predictions.

DISADVANTAGES:

Whilepredictinghousepricesusingmachinelearningoffers numerous advantages, it also comes with several disadvantages and challenges:

1. DataQuality:

Machinelearningmodelsheavilyrelyondataquality.Inaccurate or incomplete data can lead to unreliable predictions. Ensuring the data used for training and evaluation is of high quality is essential.

2. Overfitting:

Machine learning models can be prone to overfitting, where theyperformexceptionallywellonthetrainingdatabutstrugglewith new, unseen data. This can result in overly optimistic or inaccurate predictions.

3. DataPrivacyandSecurity:

Handling sensitive property and financial data for training modelsraisesprivacyandsecurityconcerns.Protectingthisinformation from unauthorized access and breaches is critical.

4. ModelInterpretability:

Somemachinelearningmodels,suchasdeepneuralnetworks, can be challenging to interpret. Understanding why a model makes a specific prediction is crucial for trust and accountability.

5. BiasandFairness:

Machine learning models can inherit biases present in the trainingdata,potentiallyleadingtounfairordiscriminatorypredictions, especially in areas where historical biases exist in real estate practices.

6. LackofTransparency:

While some machine learning models offer interpretability, othersareconsidered"blackboxes," makingitdifficulttounderstandthe logic behind their predictions. This can be a barrier to trust and regulatory compliance.

7. Maintenanceand Updates:

Machinelearningmodelsrequireongoingmaintenanceand updatestoremain accurateandrelevant.Thisincludesupdatingthem with new data and retraining as market conditions change.

8. **HighComputationalRequirements:**

Trainingandrunningsophisticatedmachinelearningmodels can demand significant computational resources, which can be costly and require advanced infrastructure.

9. **CostofImplementation:**

Integratingmachinelearningintorealestateworkflowscanbe expensive, particularly for smaller businesses or organizations that lack the resources for extensive data science and engineering teams.

10. **MarketVolatility:**

Machinelearningmodelsmaynotalwaysperformwellduring timesofextreme marketvolatilityorsignificanteconomicshifts,as they rely on historical data for predictions.

11. **Legaland Regulatory Compliance:**

Theuseofmachinelearninginreal estatemustcomplywith variouslegalandregulatorystandards.Ensuringthatmodelsadhereto fair housing laws and other regulations is crucial.

12. **LimitedDataAvailability:**

Insomeregionsorforcertainpropertytypes,high-quality data may be limited, making it challenging to build accurate models.

13. HumanExpertise:

Whilemachinelearningcanenhancethevaluationprocess,it doesn't eliminate the need for human expertise entirely. Appraisers and real estate professionals are still crucial for verifying model predictions and considering unique property characteristics.

14. ModelDegradation:

Overtime,machinelearningmodelsmayloseaccuracydue to shifts in market dynamics, and retraining is necessary to maintain performance.

BENEFITS:

Predicting house prices using machine learning offers a wide range of benefits, which can positively impact various stakeholders in thereal estateindustryandbeyond.Here aresome ofthekeybenefitsof using machine learning for house price prediction:

1. Accuracy:

Machine learning models can provide more accurate property valuationsbyconsideringabroadersetofvariablesandpatternswithin the data, leading to more precise price predictions.

2. Data-DrivenInsights:

Machinelearningmodelsuncovervaluableinsightsintothereal estate market by identifyingtrends, factorsinfluencingproperty values, andneighborhoodcharacteristics.Thisinformationcaninformstrategic decisions for investors, developers, and policymakers.

3. Efficiency:

Automatedvaluationmodelspoweredbymachinelearningcan rapidlyassess propertyvalues,saving time and effortforappraisers and individuals looking to determine a property's worth quickly.

4. **ContinuousLearning:**

Machine learning models can adapt to changing market conditionsandincorporatenewdata,ensuringthatpredictionsremain relevant and up-to-date over time.

5. **MarketTransparency:**

Machinelearningcancontributetoamoretransparentand efficient real estate market by reducing overvaluation and undervaluation, thereby promoting fair pricing and reducing market inefficiencies.

6. **RiskAssessment:**

Machine learning can evaluate the risk associated with a property,whichiscrucialformortgagelenders,insurers,andinvestors. It helps identify potential issues or opportunities related to a property's value.

7. **Customization:**

Machinelearningmodelscanbetailoredtospecificmarkets, property types, or regional variations, enabling more accurate and context-specific predictions.

8. **CostSavings:**

Usingmachinelearningforpropertyvaluationcanreducethe costs associated with manual appraisals, benefiting both businesses and individuals in terms of appraisal expenses.

9. Scalability:

Machinelearningmodelscanbeappliedatscale,makingit possible to assess property valuesin large real estate portfolios, entire neighborhoods, or even entire cities.

10. FairnessandConsistency:

Machine learning models evaluate properties objectively basedondata,reducingpotentialhumanbiasinpropertyvaluationand promoting fairness and consistency in pricing.

11. Real-TimeMonitoring:

Machinelearningmodelscanprovidereal-timemonitoringof property values, allowing stakeholders to react promptly to market changes or anomalies.

12. MarketForecasting:

Byanalyzinghistoricaldataandcurrentmarketconditions, machine learning models can make forecasts about future property values, enabling more informed investment decisions.

13. UrbanPlanning:

Accurate property valuations can inform urban planning and developmentdecisions,ensuringthatcommunitiesare builtinawaythat aligns with market dynamics and housing needs.

14. MarketCompetitiveness:

Real estate professionals can gain a competitive edge by usingmachinelearningtoprovidemoreaccuratepropertyvaluationsand better serve clients.

CONCLUSION:

Predictinghousepricesusingmachinelearningisatransformative and promising approach that has the potential to revolutionize the real estate industry. Throughout this exploration, we have uncovered the remarkable capabilities of machine learning in providing more accurate, data-driven, and nuanced predictions for property values. As we conclude, several key takeaways and implications emerge:

ImprovedAccuracy: Machinelearningmodelsconsideramyriadof variables, many of which may be overlooked by traditionalmethods. This results in more accurate predictions, benefiting both buyers and sellers who can make informed decisions based on a property's true value.

Data-Driven Insights: These models providevaluable insights into the real estate market by identifying trends, neighborhood characteristics, andotherfactorsthatinfluencepropertyprices.Thisinformationcanbe invaluablefor investors, developers, and policymakers seeking to make strategic decisions.

MarketEfficiency: Theincreasedaccuracyinpricingpredictionscan lead to a more efficientrealestate market,reducing overvaluationand undervaluation of properties. This contributes to a fairer and more transparent marketplace.

ChallengesandConsiderations: Machinelearningforhouseprice prediction is not without its challenges. Data quality, model interpretability, and ethical concerns are important considerations. Addressing these issues is crucial for the responsible and ethical deployment of this technology.

Continual Advancement: The field of machine learning is continually evolving, and as it does, so will the accuracy and capabilities of predictive models. As more data becomes available and algorithms improve,wecanexpectevenmoresophisticatedpredictionsinthefuture.

In conclusion, the application of machine learning in predicting house prices is a groundbreaking development with far-reaching implications. It empowers individuals, businesses, and governments to navigate the real estate market with more confidenceandprecision.However,itisessentialtoapproachthistechnology witha clear understanding of its potential and limitations, ensuring that its benefits are harnessed responsibly for the betterment of the real estate industry and society as a whole. As machine learning continues to advance, we can look forward to a future where property valuation becomes increasingly precise and data-informed.