## PREDICTINGHOUSEPRICEUSINGMACHINELEARNING

**TEAMMEMBER**

**953721106028:NEERATHILINGAM.A**

# Phase2SubmissionDocument

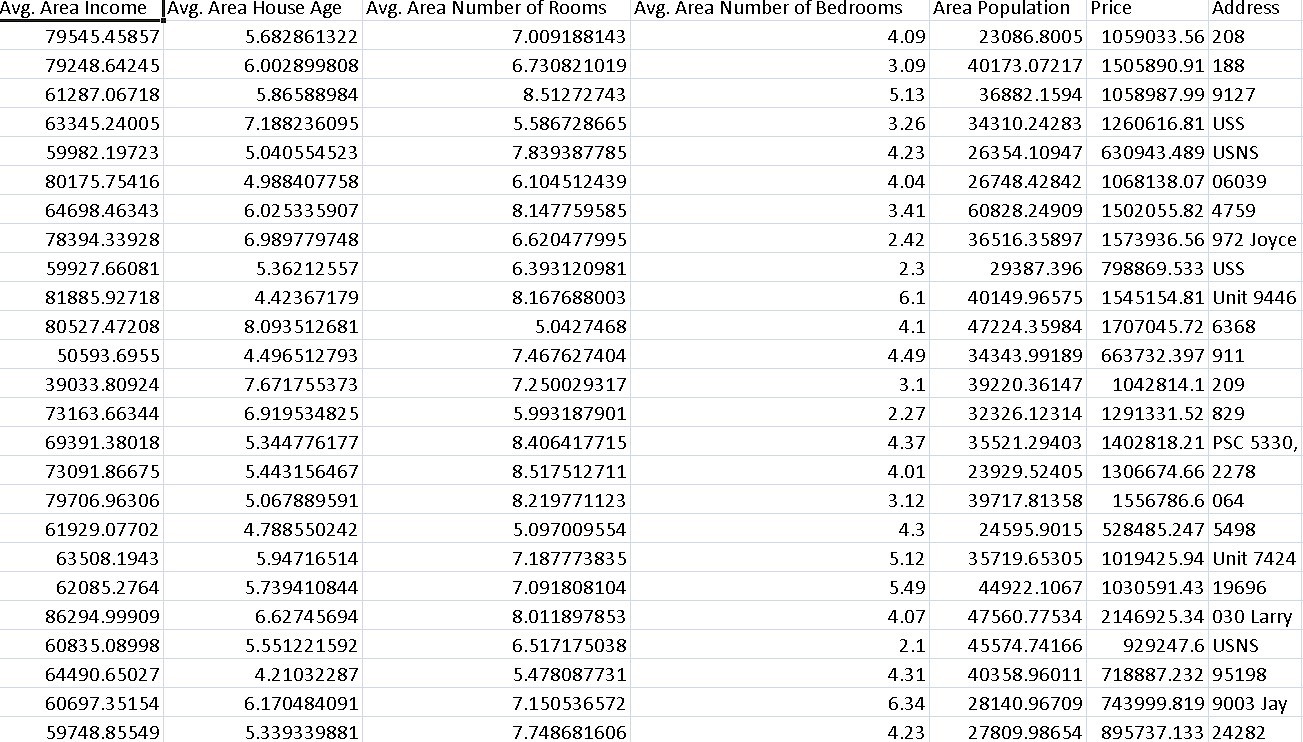
**Project:**HousePricePrediction



# Introduction:

* The real estate market is one of the most dynamic and lucrative sectors, with houseprices constantly fluctuating based on various factors such as location, size, amenities, andeconomic conditions. Accurately predicting house prices is crucial for both buyers andsellers, as it can help make informed decisions regarding buying, selling, or investing inproperties.
* Traditional linear regression models are often employed for house price prediction.However, they may not capture complex relationships between predictors and the targetvariable, leading to suboptimal predictions. In this project, we will explore advancedregression techniques to enhance the accuracy and robustness of house price predictionmodels.
* Briefly introduce the real estate market and the importance of accurate house priceprediction.

Highlight the limitations of traditional linear regression models in capturing complexrelationships.



* Emphasize the need for advanced regression techniques like Gradient Boosting andXGBoostto enhanceprediction accuracy.

# ContentforProjectPhase2 :

Consider exploring advanced regression techniques like Gradient Boosting or XGBoost forimprovedPredictionaccuracy.

**DataSource**

A good data source for house price prediction using machine learning should beAccurate,Complete,Coveringthegeographicareaofinterest,Accessible.

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

## DataCollectionandPreprocessing:

* Importing the dataset: Obtain a comprehensive dataset containing relevant featuressuchassquarefootage, numberofbedrooms, location,amenities,etc.
* Data preprocessing: Clean the data by handling missing values, outliers, andcategoricalvariables.Standardizeornormalizenumericalfeatures.

## ExploratoryDataAnalysis(EDA):

* Visualize and analyze the dataset to gain insights into the relationships betweenvariables.
* Identifycorrelationsandpatternsthatcaninformfeatureselectionandengineering.
* Presentvariousdatavisualizationstogaininsightsintothedataset.
* Explorecorrelationsbetweenfeaturesandthetargetvariable(houseprices).
* DiscussanysignificantfindingsfromtheEDAphasethatinform featureselection.

## FeatureEngineering:

* Createnewfeaturesortransformexistingonestocapturevaluableinformation.
* Utilize domain knowledge to engineer features that may impact house prices, such asproximitytoschools, transportation, orcrime rates.
* Explaintheprocessofcreatingnewfeaturesortransformingexistingones.
* Showcase domain-specific feature engineering, such as proximity scores or compositeindicators.
* Emphasizetheimpactofengineeredfeatures onmodelperformance.

## AdvancedRegressionTechniques:

* **Ridge Regression:** Introduce L2 regularization to mitigate multicollinearity andoverfitting.
* **Lasso Regression:** Employ L1 regularization to perform feature selection andsimplifythemodel.
* **ElasticNet Regression:** Combine both L1 and L2 regularization to benefit from theirrespectiveadvantages.
* **Random Forest Regression:** Implement an ensemble technique to handle non-linearityandcapturecomplexrelationshipsinthedata.
* **Gradient Boosting Regressors (e.g., XGBoost, LightGBM):** Utilize gradientboostingalgorithms forimproved accuracy.

**ModelEvaluationandSelection:**

* Splitthedatasetintotrainingandtestingsets.
* Evaluate models using appropriate metrics (e.g., Mean Absolute Error, Mean SquaredError,R-squared)toassesstheirperformance.
* Usecross-validationtechniquestotunehyperparametersandensuremodelstability.
* Compare the results with traditional linear regression models to highlightimprovements.
* Selectthebest-performingmodelforfurtheranalysis.

## ModelInterpretability:

* Explain how to interpret feature importance from Gradient Boosting and XGBoostmodels.
* Discuss the insights gained from feature importance analysis and their relevance tohousepriceprediction.
* Interpret feature importance from ensemble models like Random Forest and GradientBoostingtounderstandthefactors influencinghouseprices.

## DeploymentandPrediction:

* Deploythechosenregressionmodeltopredicthouseprices.
* Develop a user-friendly interface for users to input property features and receive pricepredictions.

# Program:

## HousePrice Prediction

Importing Dependenciesimport pandas as pdimport numpy as npimportseabornassns

importmatplotlib.pyplotasplt

from sklearn.model\_selection import train\_test\_splitfromsklearn.preprocessingimportStandardScaler

fromsklearn.metricsimportr2\_score,mean\_absolute\_error,mean\_squared\_error

from sklearn.linear\_model import LinearRegressionfromsklearn.linear\_modelimportLasso

from sklearn.ensemble import RandomForestRegressorfromsklearn.svmimport SVR

importxgboostasxg

%matplotlibinline

import warningswarnings.filterwarnings("ignore")

/opt/conda/lib/python3.10/site-packages/scipy/\_init\_.py:146: UserWarning: A NumPyversion >=1.16.5 and <1.23.0 is required for this version of SciPy (detected version1.23.5

warnings.warn(f"ANumPyversion>={np\_minversion}and<{np\_maxversion}"

LoadingDataset

dataset=pd.read\_csv('E:/USA\_Housing.csv')

### Model1 -LinearRegression

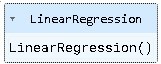
**In[1]:**

model\_lr=LinearRegression()

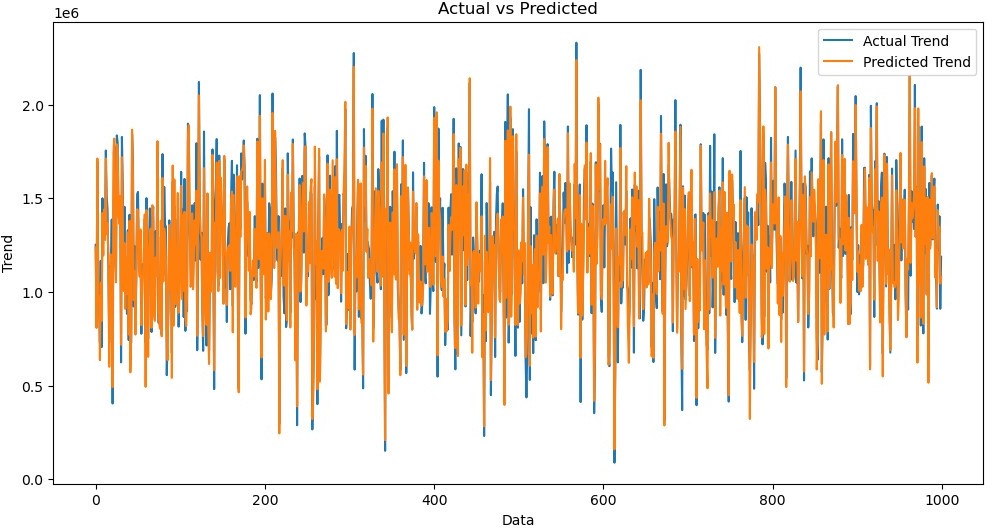
### In[2]:

model\_lr.fit(X\_train\_scal,Y\_train)

**Out[2]:**



## PredictingPrices



### In[3]:

Prediction1=model\_lr.predict(X\_test\_scal)

### Evaluation of Predicted DataIn[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')

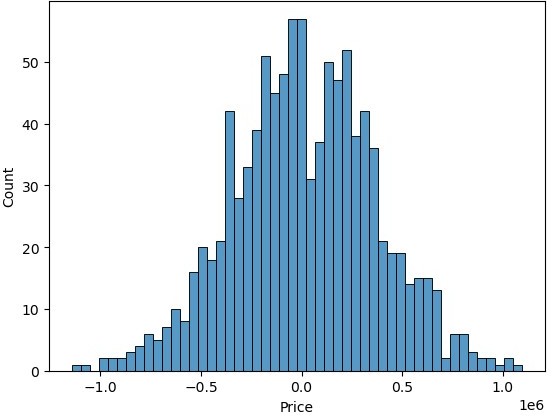
plt.xlabel('Data')

plt.ylabel('Trend')

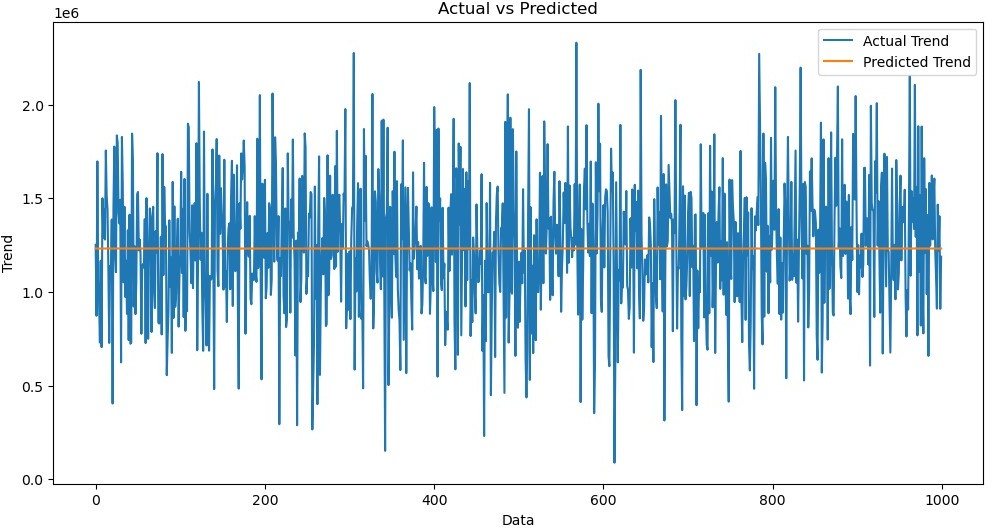
plt.legend()

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,Prediction2))

print(mean\_squared\_error(Y\_test,Prediction2))

-0.0006222175925689744

286137.81086908665

128209033251.4034

## Model3- LassoRegression

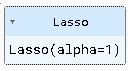
### In[13]:

model\_lar=Lasso(alpha=1)

### In[14]:

model\_lar.fit(X\_train\_scal,Y\_train)

**Out[14]:**



## 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')

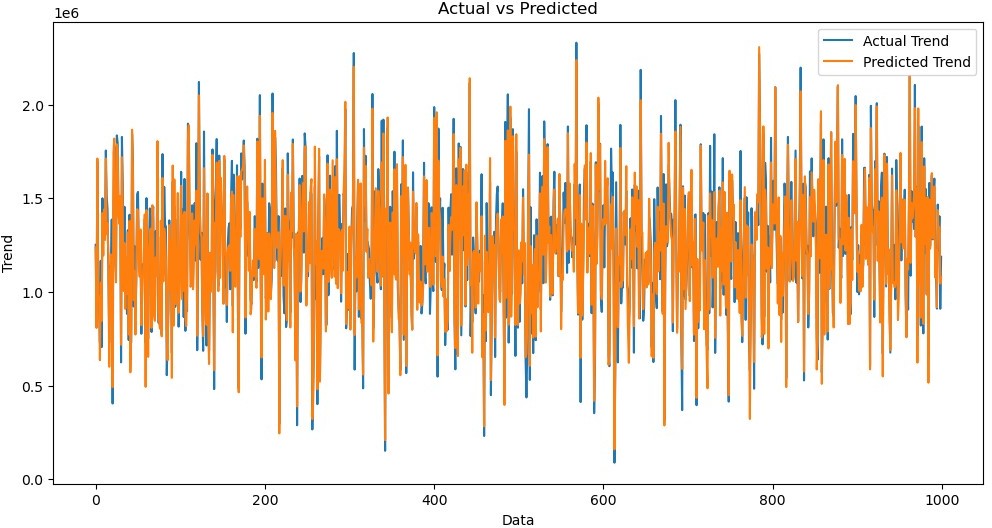
plt.ylabel('Trend')

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,Prediction2))

print(mean\_squared\_error(Y\_test,Prediction2))

-0.0006222175925689744

286137.81086908665

128209033251.4034

### Model4-RandomForestRegressor

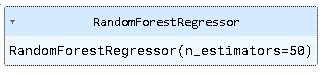
**In[19]:**

model\_rf=RandomForestRegressor(n\_estimators=50)

### In[20]:

model\_rf.fit(X\_train\_scal,Y\_train)

**Out[20]:**



## PredictingPrices

### In[21]:

Prediction4=model\_rf.predict(X\_test\_scal)

## 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')

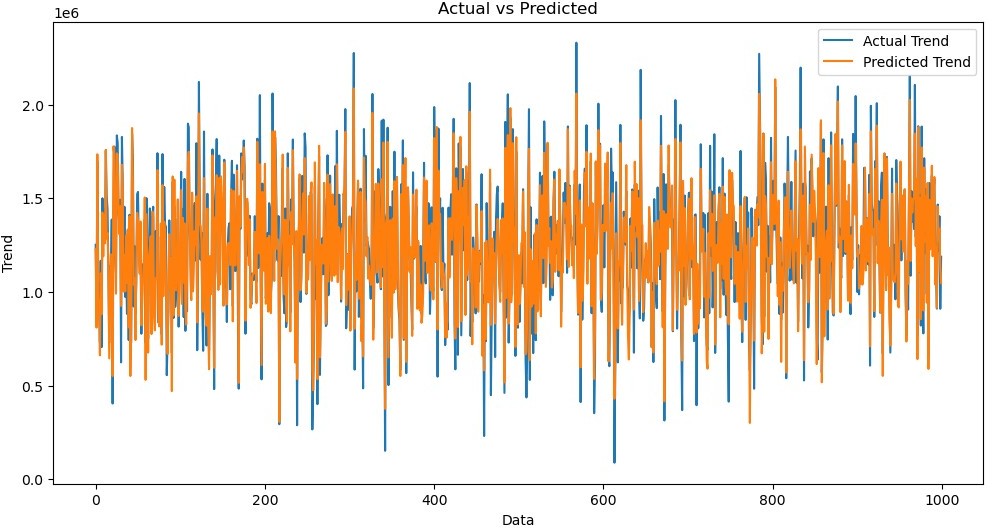
plt.ylabel('Trend')

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

## 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,...)

## PredictingPrices

### In[27]:

Prediction5=model\_xg.predict(X\_test\_scal)

## EvaluationofPredictedData

### In[28]:

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

plt.plot(np.arange(len(Y\_test)),Y\_test,label='ActualTrend')

plt.plot(np.arange(len(Y\_test)),Prediction5,label='PredictedTrend')

plt.xlabel('Data')

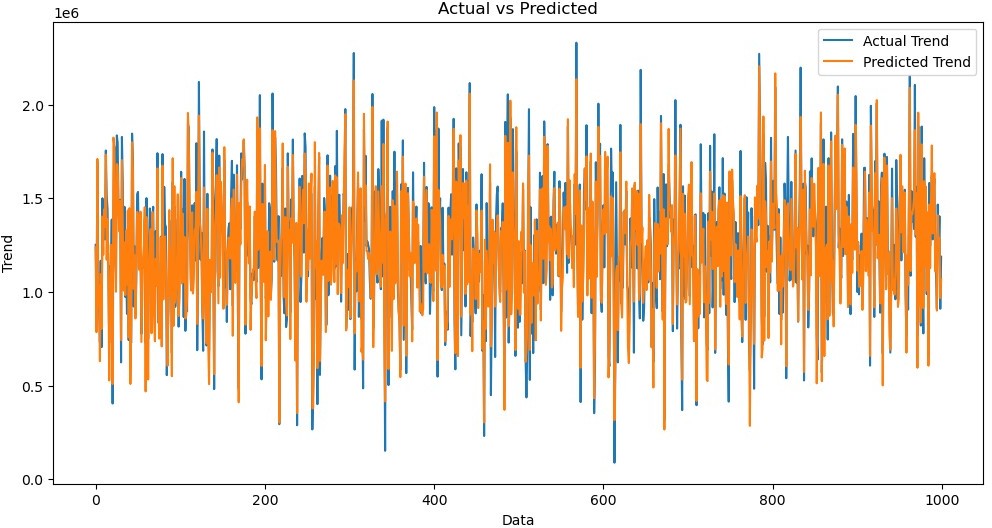
plt.ylabel('Trend')

plt.legend()

plt.title('ActualvsPredicted')

### Out[28]:

Text(0.5,1.0, 'ActualvsPredicted')





### In[29]:

sns.histplot((Y\_test-Prediction4),bins=50)

### Out[29]:

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

### In[30]:

print(r2\_score(Y\_test,Prediction2))

print(mean\_absolute\_error(Y\_test,Prediction2))

print(mean\_squared\_error(Y\_test,Prediction2))

### Out[30]:

-0.0006222175925689744

286137.81086908665

128209033251.4034

# Conclusion andFutureWork(Phase2):

## ProjectConclusion:

* In the Phase 2 conclusion, we will summarize the key findings and insights from theadvanced regression techniques. We will reiterate the impact of these techniques onimprovingtheaccuracyandrobustnessofhousepricepredictions.
* Future Work: We will discuss potential avenues for future work, such as incorporatingadditional data sources (e.g., real-time economic indicators), exploring deep learning modelsfor prediction, or expanding the project into a web application with more features andinteractivity.