**HOUSE PRICE PREDICTION USING MACHINE LEARNING**

**PROBLEM STATEMENT**

The prediction of house prices has garnered substantial interest in both academia and industry due to its significant implications for the real estate market. This study proposes a comprehensive machine learning approach for predicting house prices based on a diverse set of features including property attributes, location data, and macroeconomic indicators.

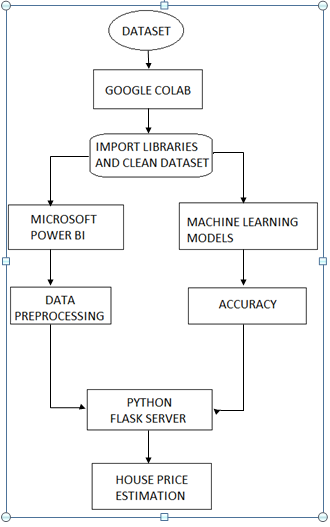
The research employs various regression techniques such as linear regression, support vector machines, and random forests to develop predictive models that can accurately estimate house prices. Additionally, feature engineering and selection methodologies are integrated to enhance the predictive capabilities of the models and mitigate the impact of multicollinearity.

The dataset used in this study encompasses a wide range of geographical locations, enabling the models to capture regional variations and specific market dynamics. The performance of the proposed models is evaluated through rigorous cross-validation techniques and evaluation metrics, such as mean squared error, mean absolute error, and R-squared.

The findings demonstrate the efficacy of the machine learning models in accurately predicting house prices, thereby providing valuable insights for real estate investors, homeowners, and policymakers to make informed decisions in the dynamic housing market. This research contributes to the existing body of knowledge by showcasing the applicability of advanced machine learning techniques in addressing complex real estate valuation challenges and offers practical implications for stakeholders in the real estate sector.

**DESIGN THINKING PROCESS:**

**System Architecture:**



**PROJECT PHASE DEVELOPMENT:**

**Steps to be followed:**

1] Data Analysis (To find out Outliers, Null Values)

2] Data Cleaning (Addressed the problem found in step 1)

3] Data Visualization (Scatter plots, Histograms, Correlation Matrix, Data distribution)

4] Data Munging (encoding of zip code, Logarithmic transformations)

6] Feature Selection (Recursive feature elimination)

7] Machine Learning Models (Linear regression, Lasso regression, Ridge regression, Polynomial regression)

**DATASET:**

**Overview of Dataset:**

This dataset has 19 Features.

|  |  |
| --- | --- |
| **Column Name** | **Description** |
| Id | to denote the house |
| Date | on which the house was sold |
| price | Price is prediction target |
| bedrooms | Number of Bedrooms/House |
| bathrooms | Number of bathrooms/House |
| sqft | living square footage of the home |
| sqft\_lot | square footage of the lot |
| floors | Total floors (levels) in house |
| waterfront | House which has a view to a waterfront |
| view | Has been viewed |
| condition | How good the condition is ( Overall ) |
| grade | overall grade given to the housing unit, based on King County grading system |
| sqft\_above | square footage of house apart from basement |
| sqft\_basement | square footage of the basement |
| yr\_built | Built Year |
| yr\_renovated | Year when house was renovated |
| zip | Zipcode |
| Lat | Latitude coordinate |
| Long | Longitude coordinate |

**PROGRAM CODE FOR LOADING AND PREPROCESSING DATASET:**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

import chardet#for encoding

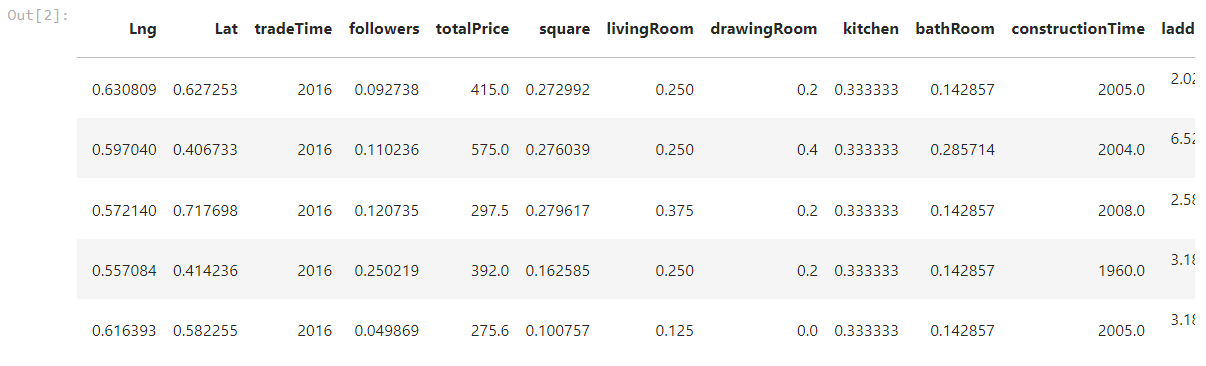
import warnings# to avoid the warnings

warnings.filterwarnings('ignore')

pd.pandas.set\_option('display.max\_columns',0)

data=pd.read\_csv("dataset.csv",index\_col=0)

data.head()



Choosing Best Machine Learning Model, top features for our model are

1.tradeTime

2.CommunityAverage

3. square

4.livingRoom

5. BathRoom

6. DrawingRoom

7.RenovationCondition

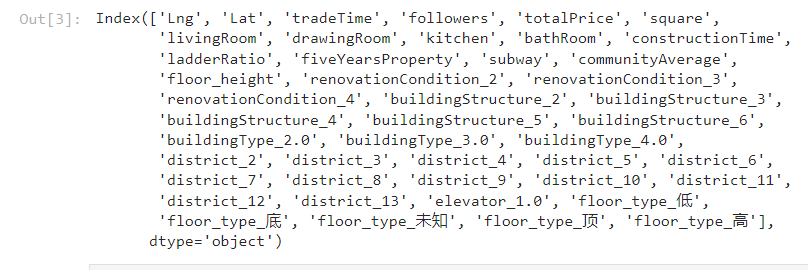
8. BuildingStructure

9. Elevator

10.constructionTime

11. Followers

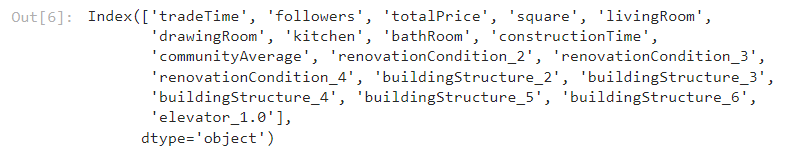
data.columns

****

data.drop(columns=["Lng","Lat","ladderRatio","fiveYearsProperty","subway","floor\_height”,"buildingType\_2.0","buildingType\_3.0","buildingType\_4.0","district\_2","district\_3","district\_4","district\_5","district\_6","district\_7", "district\_8","district\_9","district\_10","district\_11","district\_12","district\_13","floor\_type\_low","floor\_type\_bottom","floor\_type\_unknown","floor\_type\_top","floor\_type\_high"],inplace=True,axis=1)

data.shape



data.columns

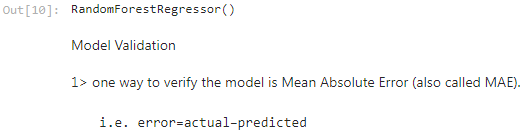
**PROGRAM CODE FOR FEATURE-EXTRACTION AND CLASSIFICATION:**

**(Choice of machine learning algorithm, model training, and evaluation metrics)**

from sklearn.ensemble import RandomForestRegressor

rfm=RandomForestRegressor()

rfm.fit(X\_train,y\_train)



print(rfm.score(X\_train,y\_train))

print(rfm.score(X\_test,y\_test))



pred**=**rfm**.**predict(X\_test)

pred



from sklearn.metrics import mean\_absolute\_error

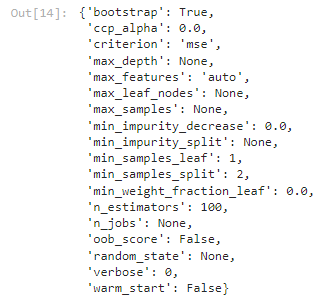
from sklearn import metrics

print(mean\_absolute\_error(y\_test,pred))

print(metrics.mean\_absolute\_error(y\_test,pred))



rfm.get\_params()



# from sklearn.model\_selection import RandomizedSearchCV

# rf\_grid= {'n\_estimators': np.arange(80,120,10),

# 'criterion':["mse", "mae"],

# 'max\_features': ["auto", "sqrt", "log2"]

# }

# print(rf\_grid)

# rfm\_randomcv=RandomizedSearchCV(estimator=rfm,param\_distributions=rf\_grid,cv=3,n\_jobs=-1,random\_state=20,verbose=1)

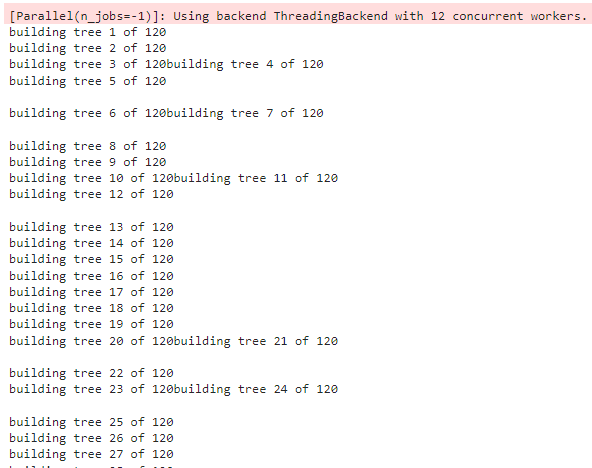
# rfm\_randomcv.fit(X\_train,y\_train)

# rfm\_randomcv.best\_params\_

rfm=RandomForestRegressor(max\_depth=None,min\_samples\_leaf=5,min\_samples\_split=6,

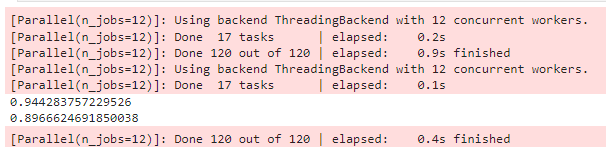
n\_estimators=120,verbose=2,n\_jobs=-1)

rfm.fit(X\_train,y\_train)

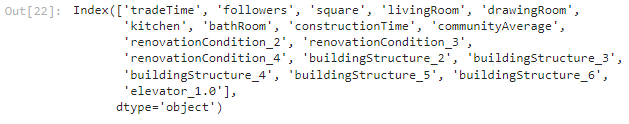


print(rfm.score(X\_train,y\_train))

print(rfm.score(X\_test,y\_test))

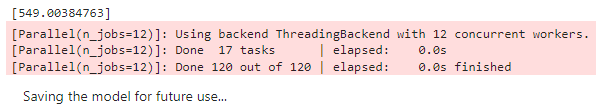


X.columns



test\_data=[[2010,120,130,3,1,2,2,2005,45000,0,0,1,0,0,0,0,0,1]]

print(rfm.predict(test\_data))



import pickle

with open('Housing\_Model','wb') as f:

pickle.dump(rfm,f)

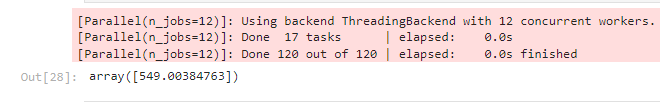
#Testing the model

with open('Housing\_Model','rb') as f:

mod=pickle.load(f)

x=mod.predict([[2010,120,130,3,1,2,2,2005,45000,0,0,1,0,0,0,0,0,1]])

x



print(float(x))

