M.KUMARASAMY
COLLEGE OF ENGINEERING
NAAC Accredited Autonomous Institution
Approved by AICTE & Affiliated to Anna University
ISO 9001:2015 & ISO 14001:2015 Certified Institution
Thalavapalayam, Karur – 639 113.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

HOUSE PRICE PREDICTION

Submitted by

M.POOJA SHREE - 927621BEC146

T.SASIREKHA - 927621BEC189

Elite Training Project Report

Submission Date : **12/08/2023**

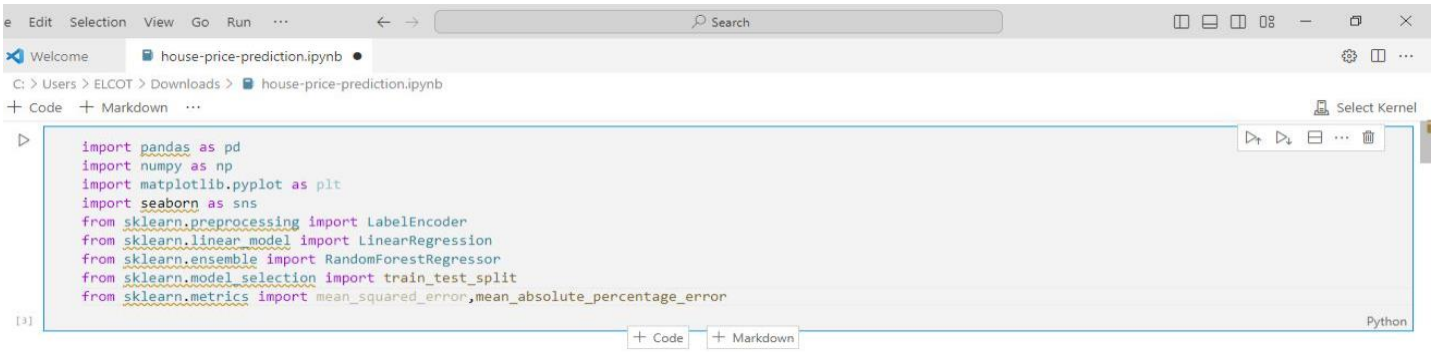
Signature of Staff Incharge **Signature of HOD**

Signature of Elite Training Coordinator
Dr.D.Pradeep, AP/CSE

Summary

The Project aims to predict the House Price by using machine learning with python. It is based on data analysis , it will try to guess the most accurate price. Taking the sample dataset for houses, and considering its various attributes, the prices for houses have been predicted by employing machine learning methods of regression-for predicting the price of estate using prior data, and clustering-for inspecting the quality of the solution or output. We will be implementing our project in jupyter Notebook ,furthermore it could be attached with a website or an app , to provide users a more customised and user friendly experience. We will import linear regression model from Sklearn. Use features identified from heatmap and label to create training and testing set. Sklearn is a open source and efficient tool for predictive data analysis. Built on NumPy, SciPy and Matplotlib. Buying a house is a stressful thing. Buyers are generally not aware of factors that influence the house prices. Many problems are faced during buying a house. Hence real estate agents are trusted with the communication between buyers and sellers as well as laying down a legal contract for the transfer. This just creates a middle man and increases the cost of houses. The real estate market is one of the most price-driven, but it is still affected by volatility. This is one of the main uses of machine learning ideas to improve and predict costs with high precision. As housing prices are fluctuating, People are cautious when trying to buy a new house based on their budget and marketing strategy. The purpose of the paper is to forecast consistent home prices for non-owners based on their financial dispositions and aspirations. The paper involves predictions using various Regression techniques like linear regression, random forest regression, polynomial regression, robust regression, lasso regression, elastic net regression, stochastic gradient descent, svm regression, artificial neural network. On a data set, house price prediction has been done by combining all of the above-mentioned strategies to determine which is the most effective. The purpose of the project is to assist the seller in accurately estimating the selling price of a house. Physical circumstances, and location, among other things, were all taken into account while determining the cost.

SOURCE CODE WITH OUTPUT




```
data  = data.drop(['date','country',"street","statezip","yr_built","yr_renovated"],axis=1)
data.head()
```

[5] Python

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	city
0	313000.0	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	Shoreline
1	2384000.0	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	Seattle
2	342000.0	3.0	2.00	1930	11947	1.0	0	0	4	1930	0	Kent
3	420000.0	3.0	2.25	2000	8030	1.0	0	0	4	1000	1000	Bellevue


```
cols  = ['bedrooms',"bathrooms","floors","price"]

for  col in cols :
    data[col] = data[col].astype(int)

# Because how can 1.50 bathroom exists
```

[8] Python

```
data.head()
```

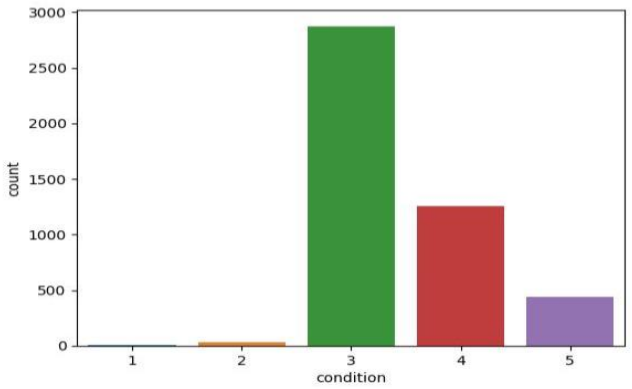
[9] Python

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	city	city_new
0	313000	3	1	1340	7912	1	0	0	3	1340	0	Shoreline	36
1	2384000	5	2	3650	9050	2	0	4	5	3370	280	Seattle	35
2	342000	3	2	1930	11947	1	0	0	4	1930	0	Kent	18
3	420000	3	2	2000	8030	1	0	0	4	1000	1000	Bellevue	3

```
[10] sns.countplot(data=data,x=data['condition'])
```

Python

```
<Axes: xlabel='condition', ylabel='count'>
```




```
[11] data['city'].value_counts()
```

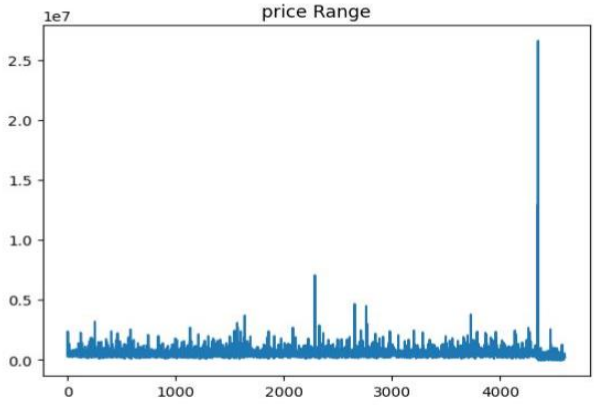
Python

```
city
Seattle                1573
Renton                  293
Bellevue                286
Redmond                 235
Issaquah                187
Kirkland                187
Kent                    185
Auburn                  176
Sammamish               175
Federal Way             148
Shoreline               123
Woodinville             115
Maple Valley             96
Mercer Island            86
Burien                   74
Snoqualmie               71
Kenmore                  66
Des Moines               58
North Bend               50
Covington                43
Duvall                   42
Lake Forest Park         36
Bothell                  33
Newcastle                33
...
Milton                    2
```

```
[12] data['price'].plot(title='price Range')
```

Python

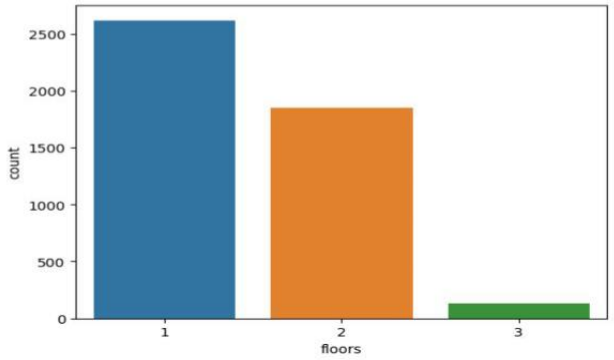
```
<Axes: title={'center': 'price Range'}>
```



```
[14] sns.countplot(data=data,x=data['floors'])
```

Python

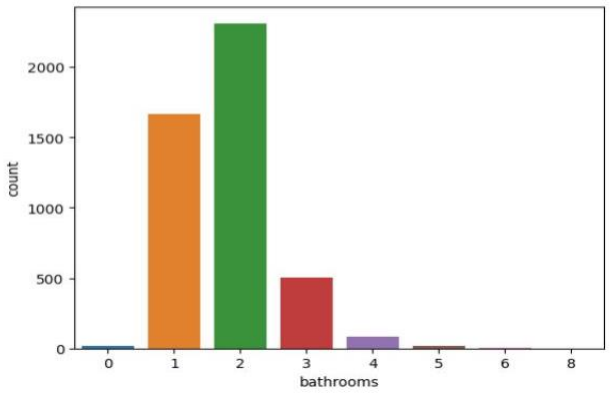
```
<Axes: xlabel='floors', ylabel='count'>
```




```
[15] sns.countplot(data=data,x=data['bathrooms'])
```

Python

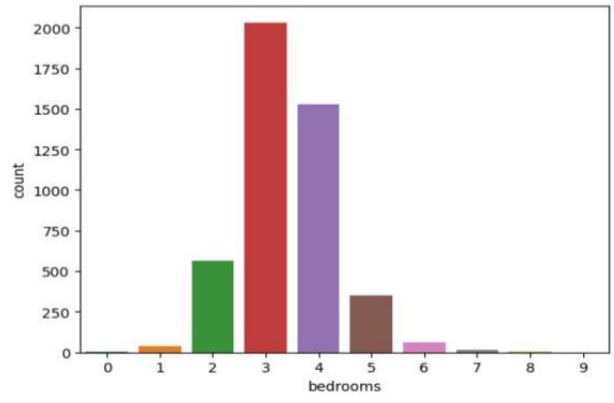
```
<Axes: xlabel='bathrooms', ylabel='count'>
```




```
[16] sns.countplot(data=data,x=data['bedrooms'])
```

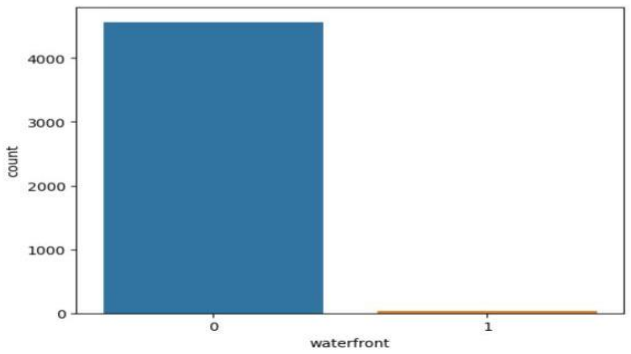
Python

```
<Axes: xlabel='bedrooms', ylabel='count'>
```



```
[17] sns.countplot(data=data,x=data['waterfront'])
```

```
<Axes: xlabel='waterfront', ylabel='count'>
```



Model Creation

```
[18] data['sqft_basement'].value_counts()
```

```
sqft_basement
0       2745
500       53
600       45
800       43
900       41
         ...
2300       1
265        1
1610       1
862        1
1640       1
Name: count, Length: 207, dtype: int64
```

```
[19] data = data.drop(["city","view","waterfront","sqft_basement"],axis=1)
```

```
[20] x  = np.array(data.loc[:,data.columns != "price"].values)
     y = np.array(data["price"].values)
```

```
[21] x
```

```
array([[   3,    1, 1340, ...,    3, 1340,   36],
       [   5,    2, 3650, ...,    5, 3370,   35],
       [   3,    2, 1930, ...,    4, 1930,   18],
       ...,
       [   3,    2, 3010, ...,    3, 3010,   32],
       [   4,    2, 2090, ...,    3, 1070,   35],
       [   3,    2, 1490, ...,    4, 1490,    9]], dtype=int64)
```

```
[22] y
```

```
array([ 313000, 2384000,  342000, ...,  416904,  203400,  220600])
```

```
[23] x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.1,random_state=42)
```

```
[24] x
```

```
array([[   3,    1, 1340, ...,    3, 1340,   36],
       [   5,    2, 3650, ...,    5, 3370,   35],
       [   3,    2, 1930, ...,    4, 1930,   18],
```

```
model_lr = LinearRegression()
model_lr.fit(x_train,y_train)
```
[25] Python

```
LinearRegression()
```

```
predictions_lr = model_lr.predict(x_test)
```
[26] Python

```
mean_absolute_percentage_error(predictions_lr,y_test)
```
[27] Python

```
0.3183144703701215
```

```
model = RandomForestRegressor(n_estimators=10)
model.fit(x_train,y_train)
predictions = model.predict(x_test)
```
[28] Python

```
mean_absolute_percentage_error(predictions,y_test)
```
[29] Python

```
data.head()
```
[32] Python

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition	sqft_above	city_new
0	313000	3	1	1340	7912	1	3	1340	36
1	2384000	5	2	3650	9050	2	5	3370	35
2	342000	3	2	1930	11947	1	4	1930	18
3	420000	3	2	2000	8030	1	4	1000	3
4	550000	4	2	1940	10500	1	4	1140	31

```
a  = [1,1,200,200,1,2,200,36]
model.predict([a])
```
[33] Python

```
array([284900.])
```

```
b =[3,1,1340,7912,1,3,1340,36]
model.predict([b])
```
[35] Python

```
array([314950.])
```

The Output shows the price of 3 bedroom, 1 bathroom, 1340 sqft_living, 7912 sqft_lot, 1 floor 3 condition, 1340 sqft_above, 36 city_ new for this inputs , the output for the house price is 314950.