FLIP ROBO

Housing Price Prediction

Submitted by:

Nomaan Sayed

ACKNOWLEDGMENT

INTRODUCTION

- Business Problem Framing
  To predict the price and build the model with available independent variables which will help the management to understand how exactly prices vary with the

variables. Even in the real world it is important to understand the prices variation in different scenario in real estate to get the maximum profit.

- Conceptual Background of the Domain Problem

  In real estate, prices varies with different parameters, doesn’t matter the size of the apartment. Like, Zones, Pool facing etc decides the prices.

- Review of Literature

  The price prediction is done on the basis of the factors like zones, pool facing, utilities etc. Its price varies from factor to factor.

- Motivation for the Problem Undertaken

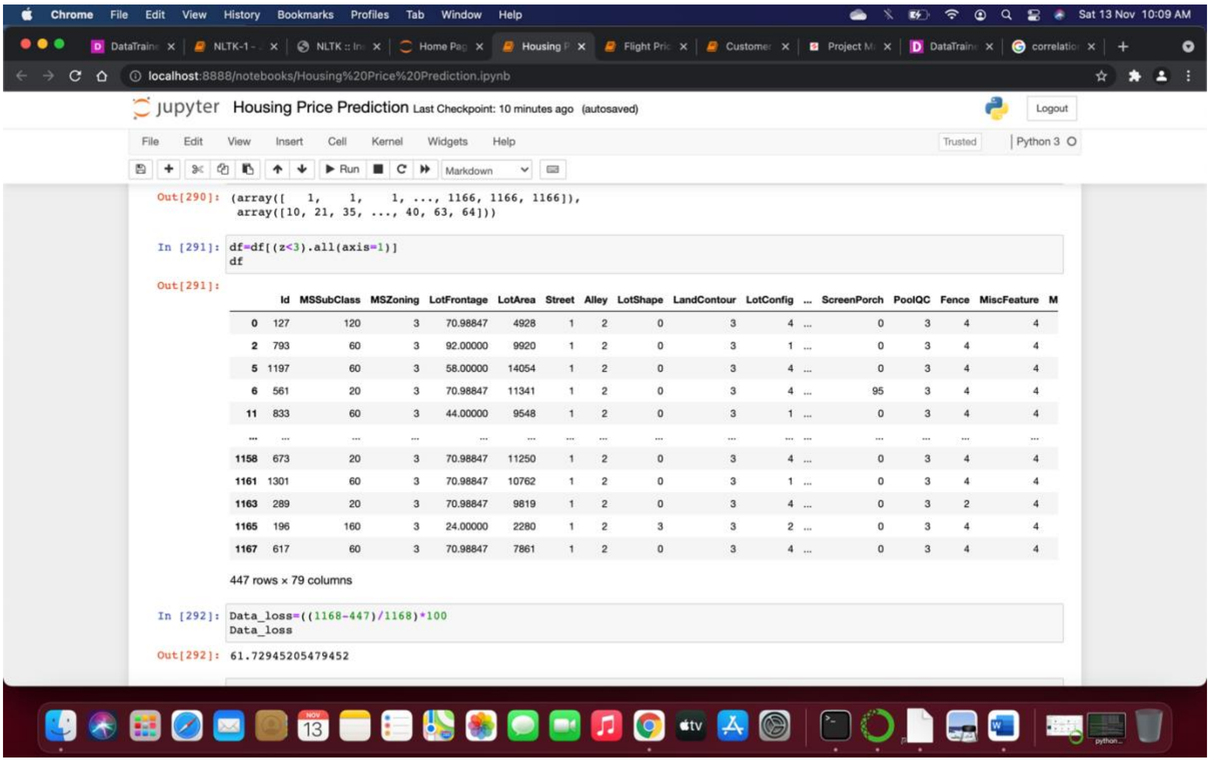
  As India is a developing country, real estate is in boom. Because as the increasing globalisation many foreigners and even Indians are investing in real estate.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

Histogram and Countplot is used in this problem to get the insights of data. Data was present in categorical and continuous type. Correlation is used to see the changes between the two data.

- Data Sources and their formats



```
Out[290]: (array([   1,    1,    1, ..., 1166, 1166, 1166]),
           array([10, 21, 35, ..., 40, 63, 64]))
```

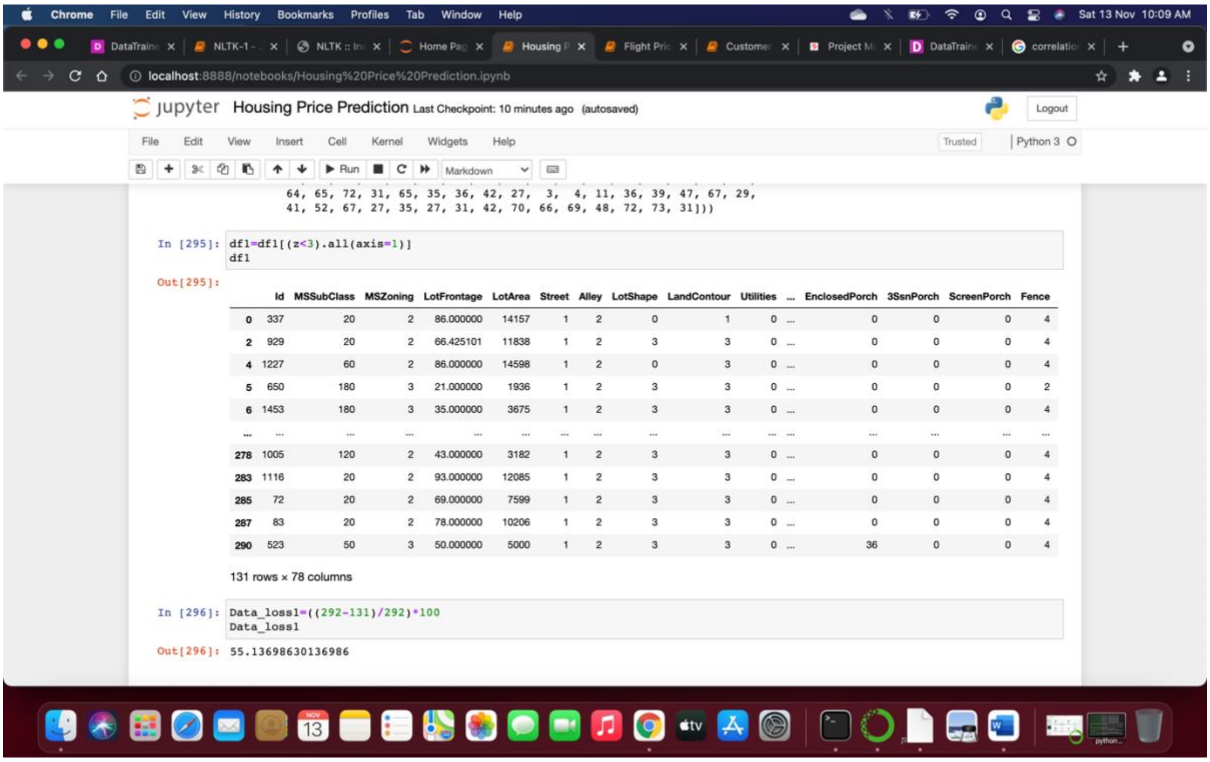
```
In [291]: df=df[(z<3).all(axis=1)]
          df
```

Out[291]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	LotConfig	...	ScreenPorch	PoolQC	Fence	MiscFeature	M
0	127	120	3	70.98847	4928	1	2	0	3	4	...	0	3	4	4	
2	793	60	3	92.00000	9920	1	2	0	3	1	...	0	3	4	4	
5	1197	60	3	58.00000	14054	1	2	0	3	4	...	0	3	4	4	
6	561	20	3	70.98847	11341	1	2	0	3	4	...	95	3	4	4	
11	833	60	3	44.00000	9548	1	2	0	3	1	...	0	3	4	4	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1158	673	20	3	70.98847	11250	1	2	0	3	4	...	0	3	4	4	
1161	1301	60	3	70.98847	10762	1	2	0	3	1	...	0	3	4	4	
1163	289	20	3	70.98847	9819	1	2	0	3	4	...	0	3	2	4	
1165	196	160	3	24.00000	2280	1	2	3	3	2	...	0	3	4	4	
1167	617	60	3	70.98847	7861	1	2	0	3	4	...	0	3	4	4	

447 rows × 79 columns

```
In [292]: Data_loss=((1168-447)/1168)*100
          Data_loss

Out[292]: 61.72945205479452
```



```
        64, 65, 72, 31, 65, 35, 36, 42, 27,  3,  4, 11, 36, 39, 47, 67, 29,
        41, 52, 67, 27, 35, 27, 31, 42, 70, 66, 69, 48, 72, 73, 31]))
```

```
In [295]: df1=df1[(z<3).all(axis=1)]
          df1
```

Out[295]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	EnclosedPorch	3SsnPorch	ScreenPorch	Fence
0	337	20	2	86.000000	14157	1	2	0	1	0	...	0	0	0	4
2	929	20	2	66.425101	11838	1	2	3	3	0	...	0	0	0	4
4	1227	60	2	86.000000	14598	1	2	0	3	0	...	0	0	0	4
5	650	180	3	21.000000	1936	1	2	3	3	0	...	0	0	0	2
6	1453	180	3	35.000000	3675	1	2	3	3	0	...	0	0	0	4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
278	1005	120	2	43.000000	3182	1	2	3	3	0	...	0	0	0	4
283	1116	20	2	93.000000	12085	1	2	3	3	0	...	0	0	0	4
285	72	20	2	69.000000	7599	1	2	3	3	0	...	0	0	0	4
287	83	20	2	78.000000	10206	1	2	3	3	0	...	0	0	0	4
290	523	50	3	50.000000	5000	1	2	3	3	0	...	36	0	0	4

131 rows × 78 columns

```
In [296]: Data_loss1=((292-131)/292)*100
          Data_loss1

Out[296]: 55.13698630136986
```

Almost more than 50% data is removed to get the best output result.

- Data Preprocessing Done

  1- Used histogram and Countplot to get the insights of the data.

  2- Described the data and uses Correlation to get the result between two data

  3- Checked the outliers and skewness

  4- Removed the skewness using zscore

  5- Uses power transform to transform the data

  6- Scale the data as ML works much better in scaled data.

- Data Inputs- Logic- Output Relationships

  In this problem, the output which is price of the house is depended on the parameters like Pool, land, etc. Output is directly proportional to input.

- State the set of assumptions (if any) related to the problem under consideration

  No such assumptions.

- Hardware and Software Requirements and Tools Used

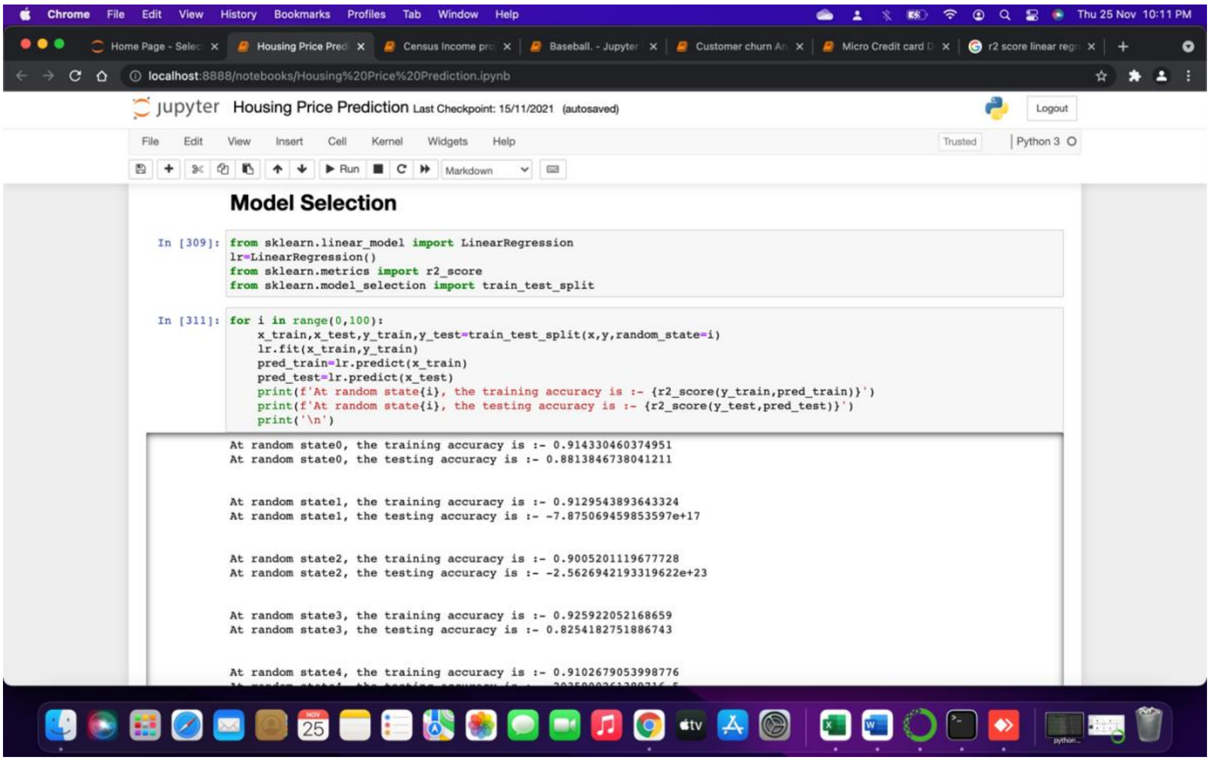
  Hardware :- Macbook Air- i5 9th gen,8gb ram.

  Software :-Jupyter Notebook and the libraries – Numpy, Pandas, Scikit, Scipy , power_transform ,Standard Scaler.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)
  Firstly, we had seen the insights and cleaned the data using various techniques and finally used the Linear regression model as the label is continuous data. Coefficient of evaluate score(r2_score) was used to evaluate the performance of linear regression model.
- Testing of Identified Approaches (Algorithms)
  Linear Regression
  R2_score

- Run and Evaluate selected models



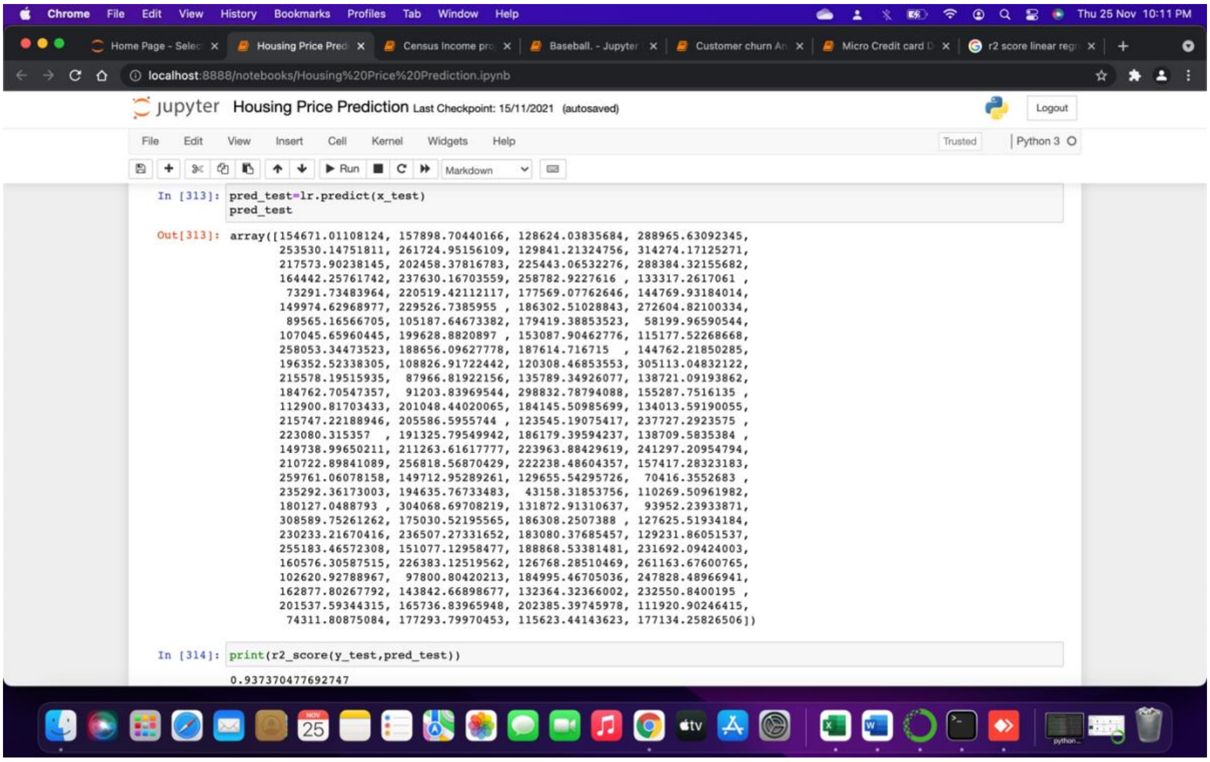
jupyter Housing Price Prediction Last Checkpoint: 15/11/2021 (autosaved)
localhost:8888/notebooks/Housing%20Price%20Prediction.ipynb
Thu 25 Nov 10:11 PM

Model Selection

```
In [309]: from sklearn.linear_model import LinearRegression
          lr=LinearRegression()
          from sklearn.metrics import r2_score
          from sklearn.model_selection import train_test_split
```

```
In [311]: for i in range(0,100):
              x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=i)
              lr.fit(x_train,y_train)
              pred_train=lr.predict(x_train)
              pred_test=lr.predict(x_test)
              print(f'At random state{i}, the training accuracy is :- {r2_score(y_train,pred_train)}')
              print(f'At random state{i}, the testing accuracy is :- {r2_score(y_test,pred_test)}')
              print('\n')
```

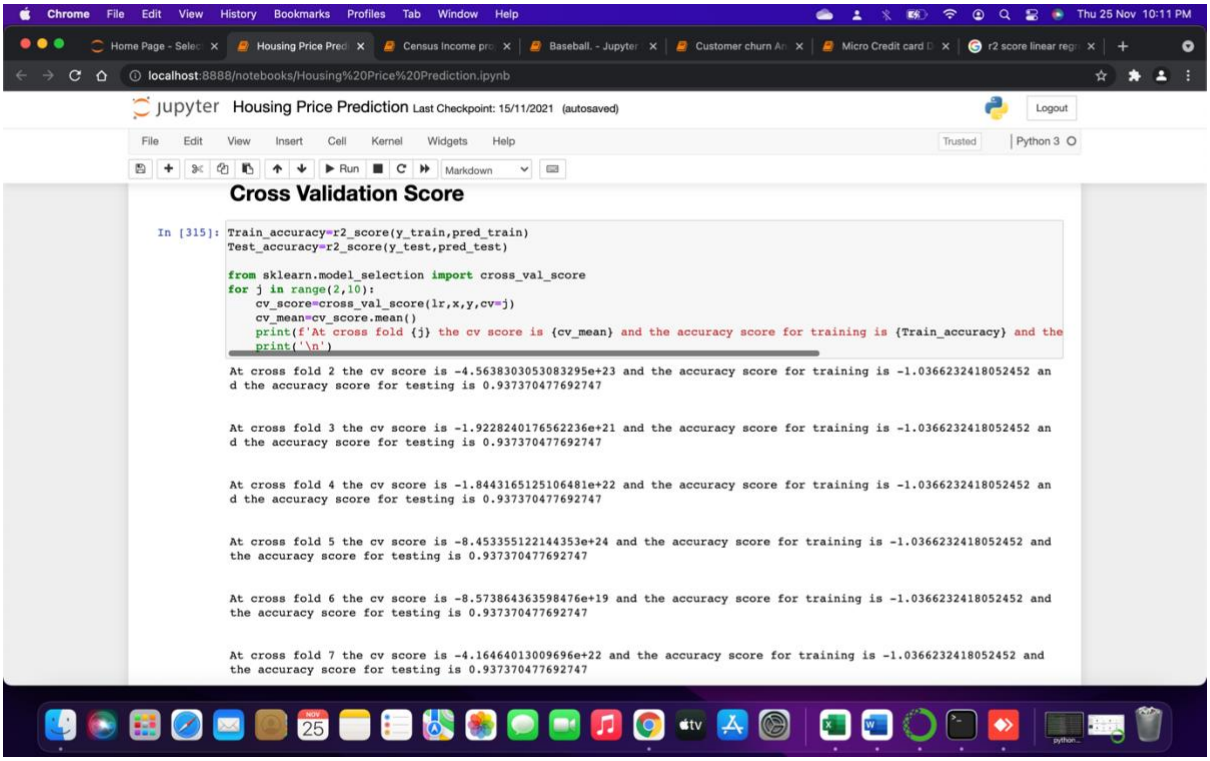
```
At random state0, the training accuracy is :- 0.914330460374951
At random state0, the testing accuracy is :- 0.8813846738041211


At random state1, the training accuracy is :- 0.9129543893643324
At random state1, the testing accuracy is :- -7.875069459853597e+17


At random state2, the training accuracy is :- 0.9005201119677728
At random state2, the testing accuracy is :- -2.5626942193319622e+23


At random state3, the training accuracy is :- 0.925922052168659
At random state3, the testing accuracy is :- 0.8254182751886743


At random state4, the training accuracy is :- 0.9102679053998776
```



jupyter Housing Price Prediction Last Checkpoint: 15/11/2021 (autosaved)
localhost:8888/notebooks/Housing%20Price%20Prediction.ipynb
Thu 25 Nov 10:11 PM

```
In [313]: pred_test=lr.predict(x_test)
          pred_test

Out[313]: array([154671.01108124, 157898.70440166, 128624.03835684, 288965.63092345,
                 253530.14751811, 261724.95156109, 129841.21324756, 314274.17125271,
                 217573.90238145, 202458.37816783, 225443.06532276, 288384.32155682,
                 164442.25761742, 237630.16703559, 258782.9227616 , 133317.2617061 ,
                  73291.73483964, 220519.42112117, 177569.07762646, 144769.93184014,
                 149974.62968977, 229526.7385955 , 186302.51028843, 272604.82100334,
                  89565.16566705, 105187.64673382, 179419.38853523,  58199.96590544,
                 107045.65960445, 199628.8820897 , 153087.90462776, 115177.52268668,
                 258053.34473523, 188656.09627778, 187614.716715  , 144762.21850285,
                 196352.52338305, 108826.91722442, 120308.46853553, 305113.04832122,
                 215578.19515935,  87966.81922156, 135789.34926077, 138721.09193862,
                 184762.70547357,  91203.83969544, 298832.78794088, 155287.7516135 ,
                 112900.81703433, 201048.44020065, 184145.50985699, 134013.59190055,
                 215747.22188946, 205586.5955744 , 123545.19075417, 237727.2923575 ,
                 223080.315357  , 191325.79549942, 186179.39594237, 138709.5835384 ,
                 149738.99650211, 211263.61617777, 223963.88429619, 241297.20954794,
                 210722.89841089, 256818.56870429, 222238.48604357, 157417.28323183,
                 259761.06078158, 149712.95289261, 129655.54295726,  70416.3552683 ,
                 235292.36173003, 194635.76733483,  43158.31853756, 110269.50961982,
                 180127.0488793 , 304068.69708219, 131872.91310637,  93952.23933871,
                 308589.75261262, 175030.52195565, 186308.2507388 , 127625.51934184,
                 230233.21670416, 236507.27331652, 183080.37685457, 129231.86051537,
                 255183.46572308, 151077.12958477, 188868.53381481, 231692.09424003,
                 160576.30587515, 226383.12519562, 126768.28510469, 261163.67600765,
                 102620.92788967,  97800.80420213, 184995.46705036, 247828.48966941,
                 162877.80267792, 143842.66898677, 132364.32366002, 232550.8400195 ,
                 201537.59344315, 165736.83965948, 202385.39745978, 111920.90246415,
                  74311.80875084, 177293.79970453, 115623.44143623, 177134.25826506])
```

```
In [314]: print(r2_score(y_test,pred_test))

          0.937370477692747
```

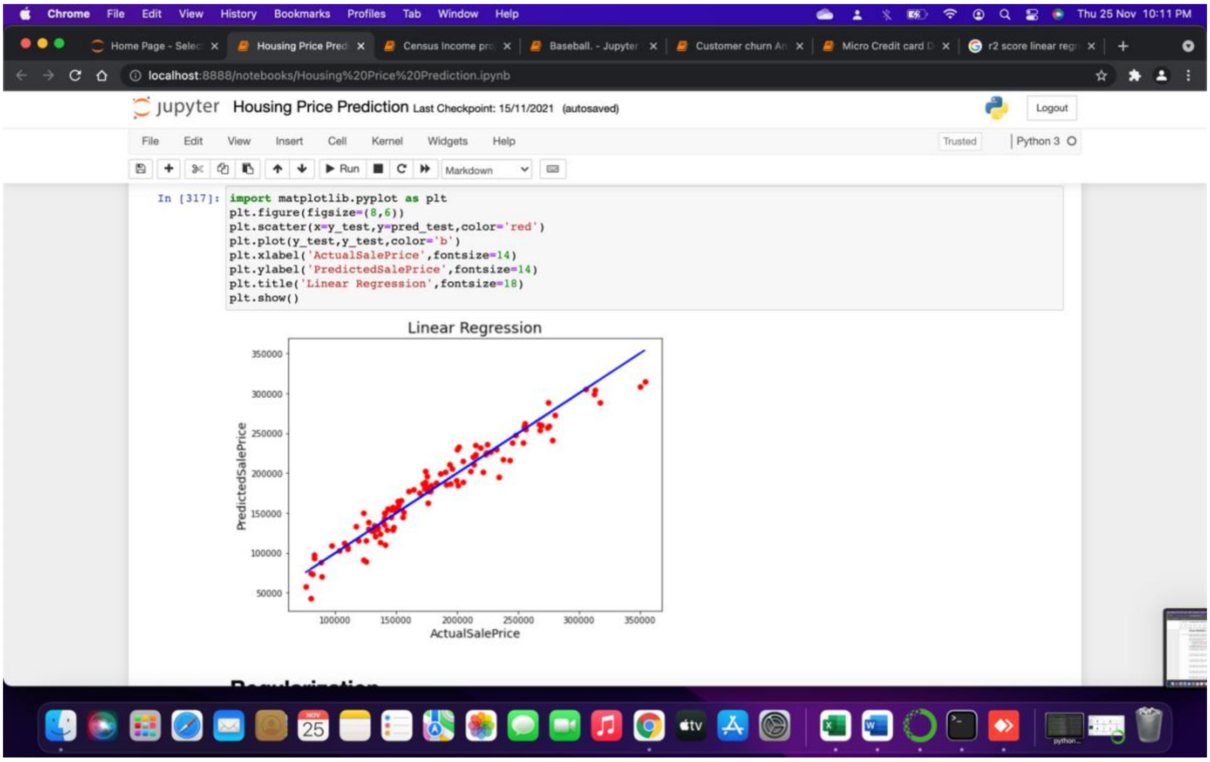


Housing Price Prediction Last Checkpoint: 15/11/2021 (autosaved)

Cross Validation Score

```
In [315]: Train_accuracy=r2_score(y_train,pred_train)
          Test_accuracy=r2_score(y_test,pred_test)

          from sklearn.model_selection import cross_val_score
          for j in range(2,10):
              cv_score=cross_val_score(lr,x,y,cv=j)
              cv_mean=cv_score.mean()
              print(f'At cross fold {j} the cv score is {cv_mean} and the accuracy score for training is {Train_accuracy} and the
              print('\n')
```

```
At cross fold 2 the cv score is -4.5638303053083295e+23 and the accuracy score for training is -1.0366232418052452 an
d the accuracy score for testing is 0.937370477692747


At cross fold 3 the cv score is -1.9228240176562236e+21 and the accuracy score for training is -1.0366232418052452 an
d the accuracy score for testing is 0.937370477692747


At cross fold 4 the cv score is -1.8443165125106481e+22 and the accuracy score for training is -1.0366232418052452 an
d the accuracy score for testing is 0.937370477692747


At cross fold 5 the cv score is -8.453355122144353e+24 and the accuracy score for training is -1.0366232418052452 and
the accuracy score for testing is 0.937370477692747


At cross fold 6 the cv score is -8.573864363598476e+19 and the accuracy score for training is -1.0366232418052452 and
the accuracy score for testing is 0.937370477692747


At cross fold 7 the cv score is -4.16464013009696e+22 and the accuracy score for training is -1.0366232418052452 and
the accuracy score for testing is 0.937370477692747
```



Housing Price Prediction Last Checkpoint: 15/11/2021 (autosaved)

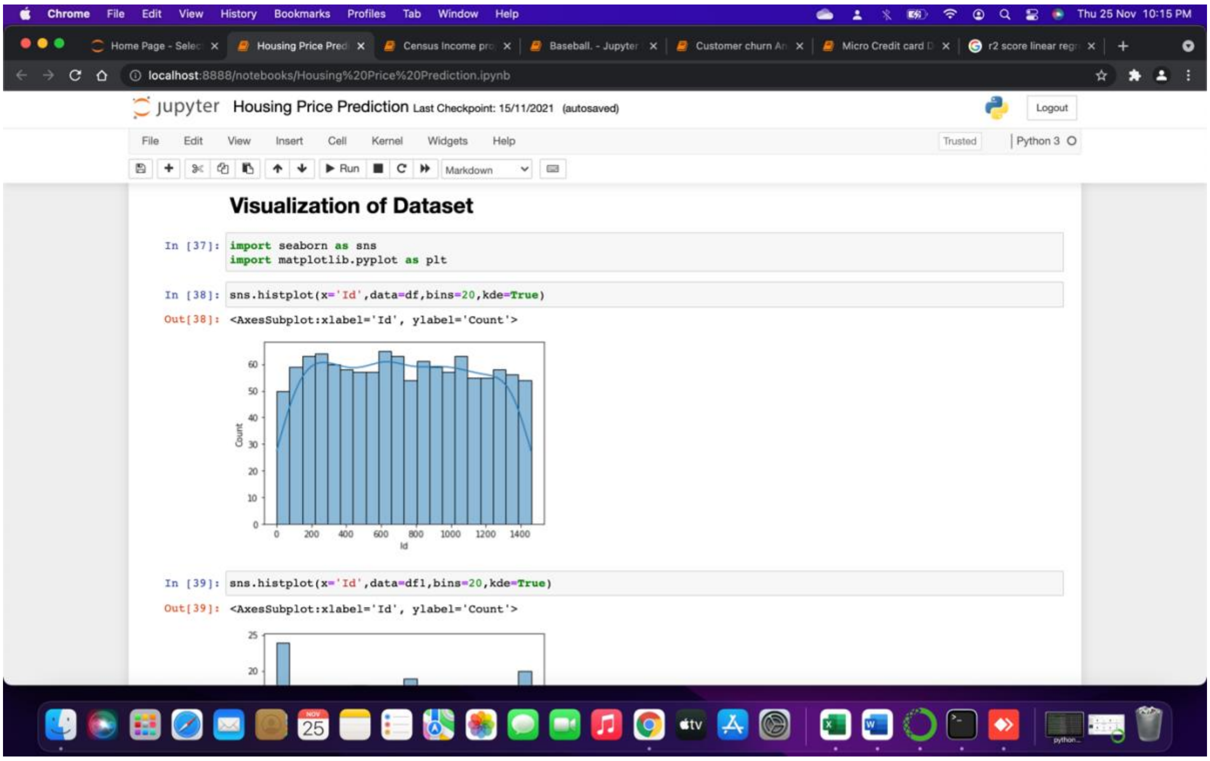
```
In [317]: import matplotlib.pyplot as plt
          plt.figure(figsize=(8,6))
          plt.scatter(x=y_test,y=pred_test,color='red')
          plt.plot(y_test,y_test,color='b')
          plt.xlabel('ActualSalePrice',fontsize=14)
          plt.ylabel('PredictedSalePrice',fontsize=14)
          plt.title('Linear Regression',fontsize=18)
          plt.show()
```

- Key Metrics for success in solving problem under consideration

  R2_score metrics is used.

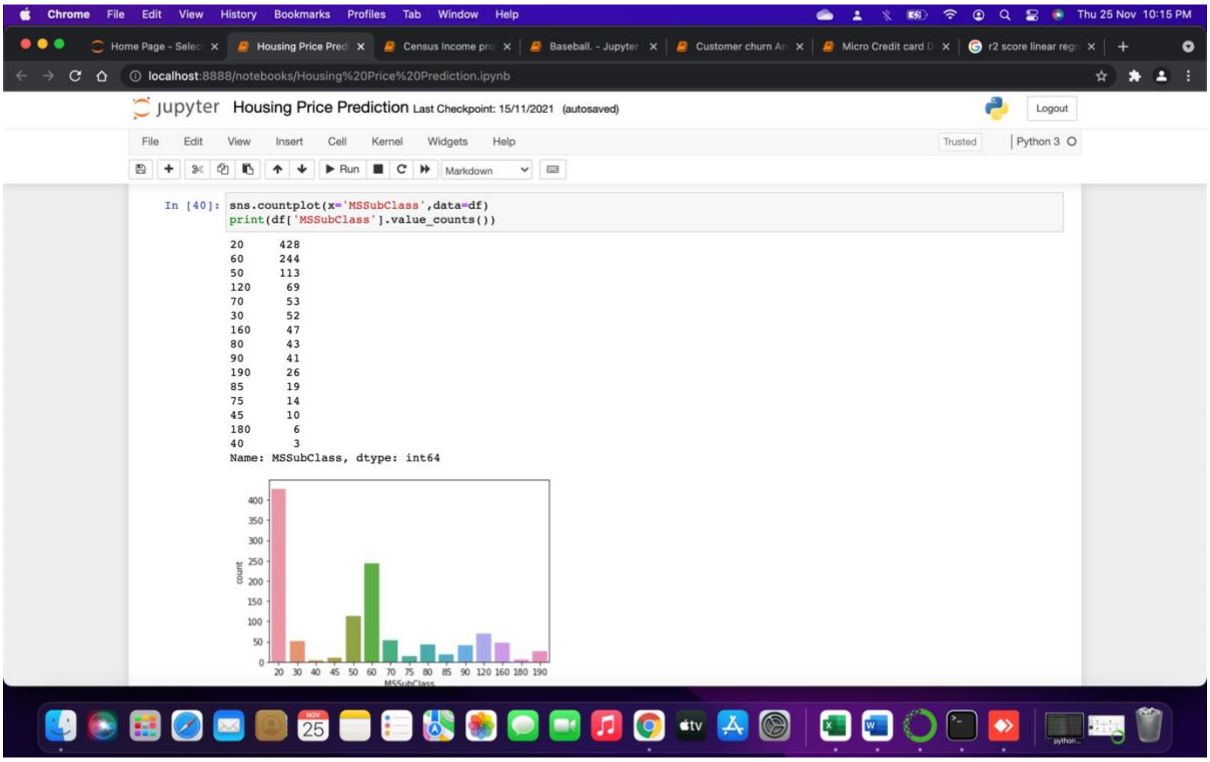
- Visualizations
- Interpretation of the Results

There are many visualization plots in the project, it is difficult to put them all so only few images are displayed rest can be interpreted from inside the programm.

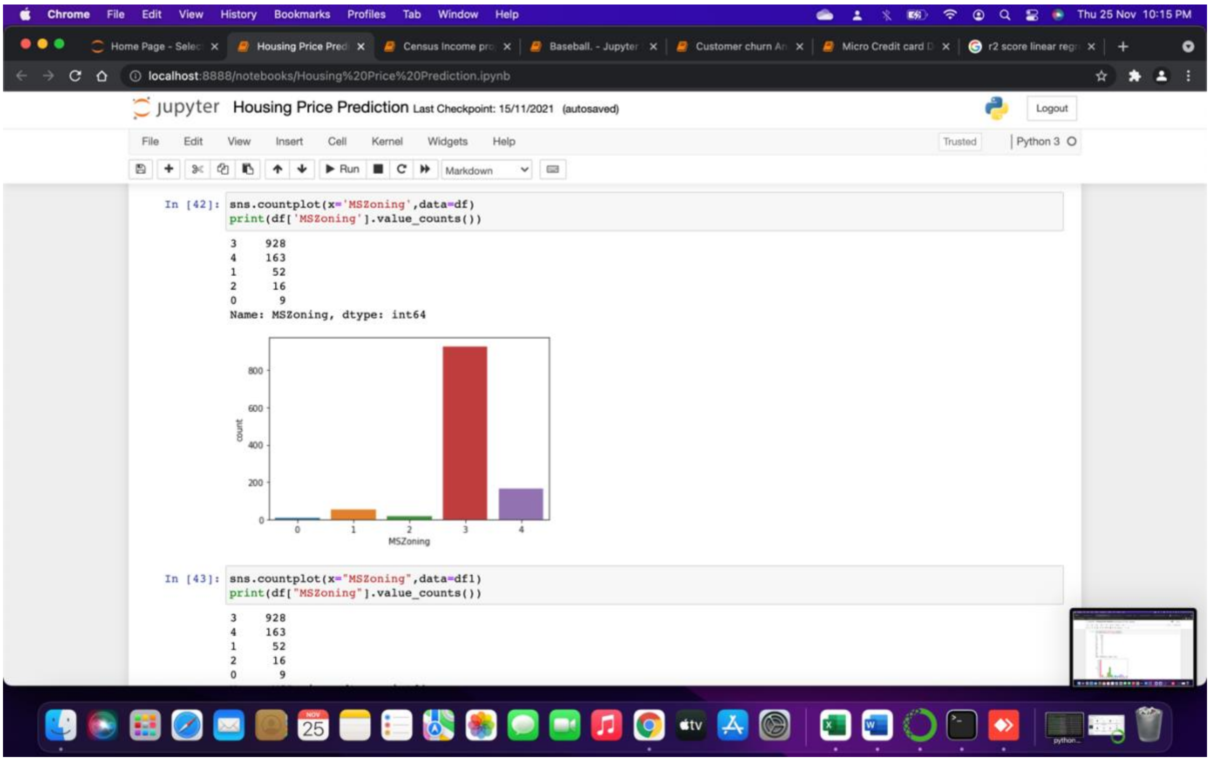

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Thu 25 Nov 10:15 PM
Home Page - Selec
Housing Price Pred
Census Income pro
Baseball. - Jupyter
Customer churn An
Micro Credit card D
r2 score linear regr
localhost:8888/notebooks/Housing%20Price%20Prediction.ipynb
jupyter Housing Price Prediction Last Checkpoint: 15/11/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
Count
Id
0 10 20 30 40 50 60
0 200 400 600 800 1000 1200 1400
25
20

Visualization of Dataset

```
In [37]: import seaborn as sns
         import matplotlib.pyplot as plt

In [38]: sns.histplot(x='Id',data=df,bins=20,kde=True)

Out[38]: <AxesSubplot:xlabel='Id', ylabel='Count'>

In [39]: sns.histplot(x='Id',data=df1,bins=20,kde=True)

Out[39]: <AxesSubplot:xlabel='Id', ylabel='Count'>
```

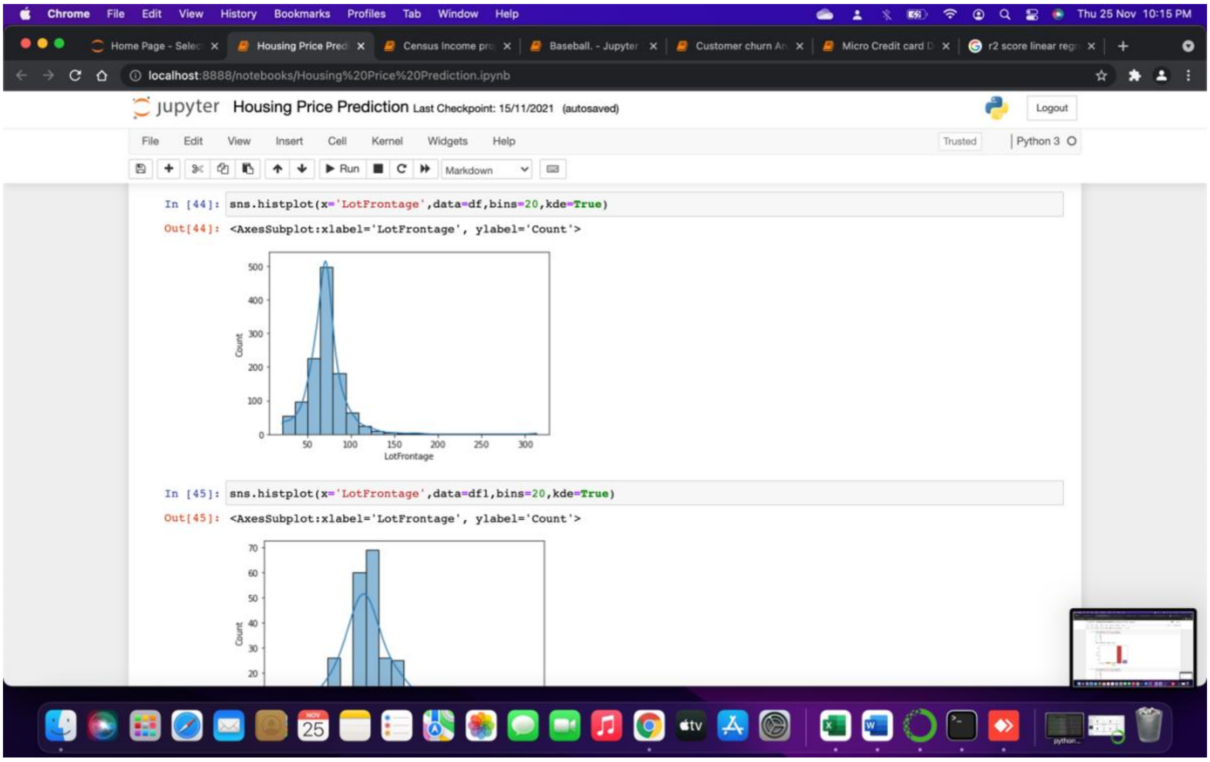


```
In [40]: sns.countplot(x='MSSubClass',data=df)
         print(df['MSSubClass'].value_counts())

20     428
60     244
50     113
120     69
70      53
30      52
160     47
80      43
90      41
190     26
85      19
75      14
45      10
180      6
40       3
Name: MSSubClass, dtype: int64
```


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Thu 25 Nov 10:15 PM
Home Page - Selec
Housing Price Pred
Census Income pro
Baseball. - Jupyter
Customer churn An
Micro Credit card D
r2 score linear regr
localhost:8888/notebooks/Housing%20Price%20Prediction.ipynb
jupyter
Housing Price Prediction Last Checkpoint: 15/11/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
800
600
400
200
0
count
0
1
2
3
4
MSZoning

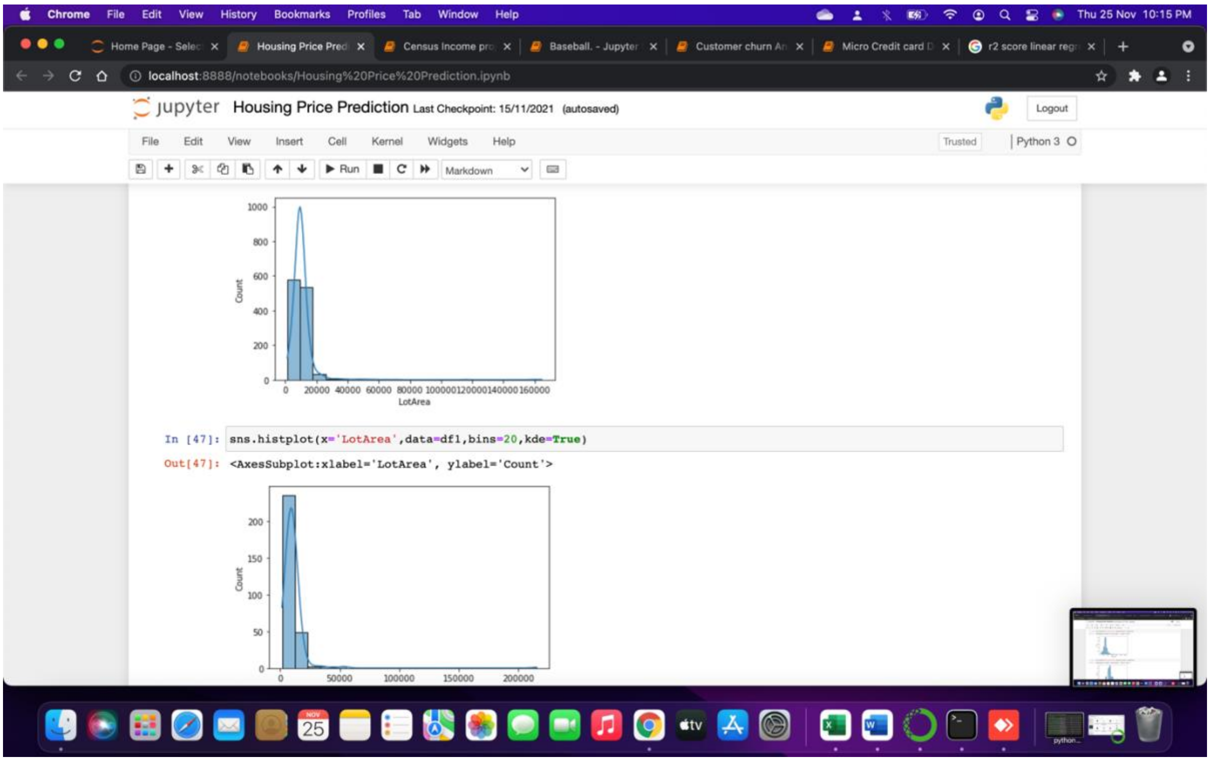
```
In [42]: sns.countplot(x='MSZoning',data=df)
         print(df['MSZoning'].value_counts())

         3    928
         4    163
         1     52
         2     16
         0      9
         Name: MSZoning, dtype: int64
```

```
In [43]: sns.countplot(x="MSZoning",data=df1)
         print(df["MSZoning"].value_counts())

         3    928
         4    163
         1     52
         2     16
         0      9
```

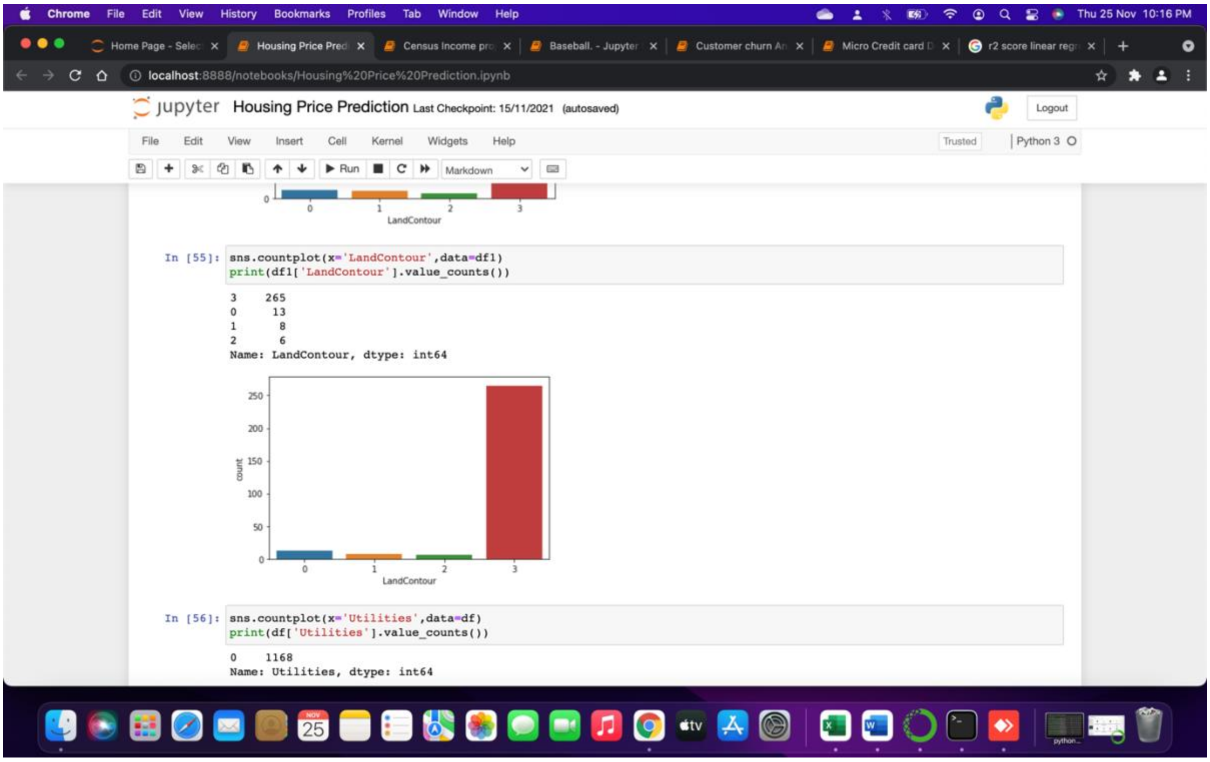


```
In [44]: sns.histplot(x='LotFrontage',data=df,bins=20,kde=True)
Out[44]: <AxesSubplot:xlabel='LotFrontage', ylabel='Count'>
```

```
In [45]: sns.histplot(x='LotFrontage',data=df1,bins=20,kde=True)
Out[45]: <AxesSubplot:xlabel='LotFrontage', ylabel='Count'>
```

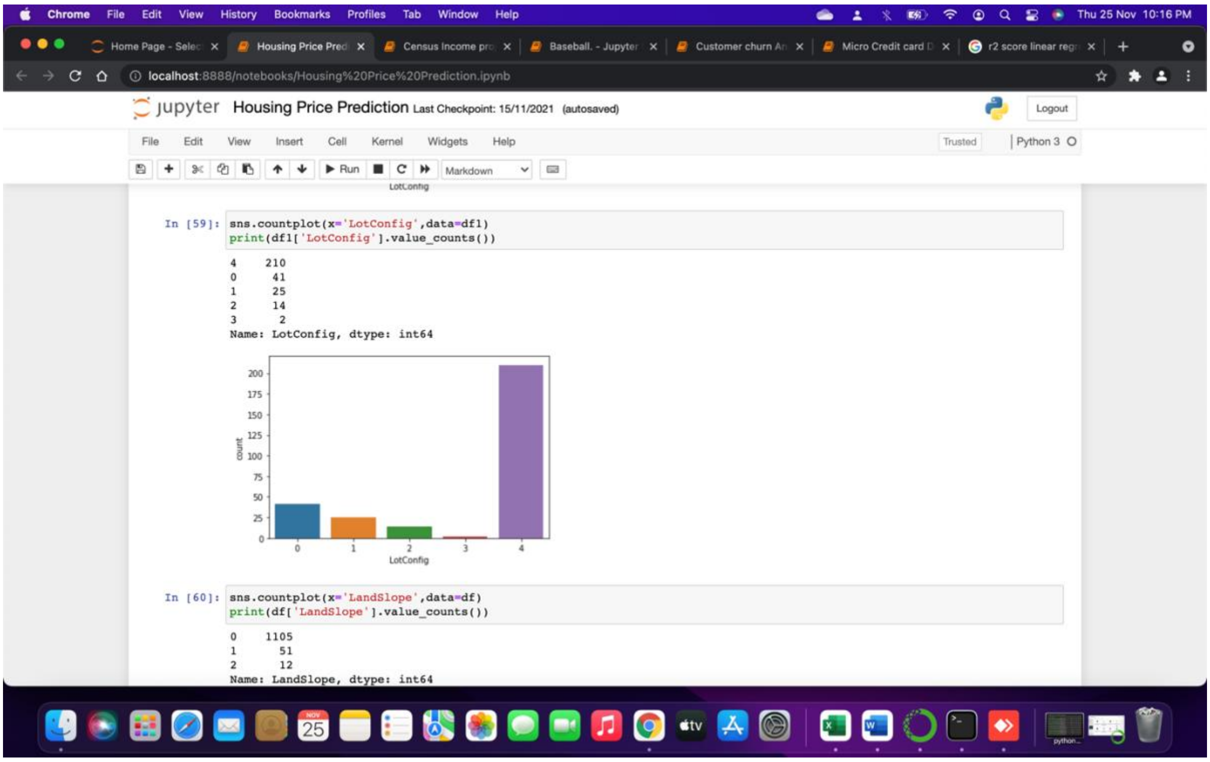

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Thu 25 Nov 10:15 PM
localhost:8888/notebooks/Housing%20Price%20Prediction.ipynb
jupyter Housing Price Prediction Last Checkpoint: 15/11/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
Count
LotArea
0 20000 40000 60000 80000 100000120000140000160000
0 200 400 600 800 1000
0 50 100 150 200
0 50000 100000 150000 200000

```
In [47]: sns.histplot(x='LotArea',data=df1,bins=20,kde=True)
Out[47]: <AxesSubplot:xlabel='LotArea', ylabel='Count'>
```


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Thu 25 Nov 10:16 PM
localhost:8888/notebooks/Housing%20Price%20Prediction.ipynb
jupyter Housing Price Prediction Last Checkpoint: 15/11/2021 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
LandContour
count
0 50 100 150 200 250
0 1 2 3

```
In [55]: sns.countplot(x='LandContour',data=df1)
         print(df1['LandContour'].value_counts())

         3    265
         0     13
         1      8
         2      6
         Name: LandContour, dtype: int64
```

```
In [56]: sns.countplot(x='Utilities',data=df)
         print(df['Utilities'].value_counts())

         0    1168
         Name: Utilities, dtype: int64
```



```
In [59]: sns.countplot(x='LotConfig',data=df1)
         print(df1['LotConfig'].value_counts())

4    210
0     41
1     25
2     14
3      2
Name: LotConfig, dtype: int64
```

```
In [60]: sns.countplot(x='LandSlope',data=df)
         print(df['LandSlope'].value_counts())

0    1105
1      51
2      12
Name: LandSlope, dtype: int64
```

Give a summary of what results were interpreted from the visualizations, preprocessing and modelling.

From the visualizations we can makeout the exact insights of the data and can decide on how to remove the outliers and clean up the data. From the model itself we can get the accuracy of the data.

CONCLUSION

- Key Findings and Conclusions of the Study
  Model is working good and giving the best results from our assumptions.
- Learning Outcomes of the Study in respect of Data Science
  Before visualization it is very difficult to come to the conclusion just by seeing the data, as it is very huge. But

using visualization it shows the insights which we cannot assume and further there are many unwanted data which should be removed so using the data cleaning it makes it pure and gives the best results. It is like finding the needle in grass using a magnet. Data cleaning works as a magnet.

- Limitations of this work and Scope for Future Work
  It doesn’t gives the 100% result so a slight dicey for the company to invest a huge amount or not. But yes gave a satisfactory result where the loss would be very less.