FLIP ROBO

HOUSING: PRICE PREDICTION

Submitted by:

Nahid Anjum Gouri

ACKNOWLEDGMENT

The project entitled "Housing: Price Prediction" is done by me during my internship with Flip Robo Technologies. I am grateful to Data Trained and Flip Robo Technologies for their guidance during this project.

INTRODUCTION

- Houses are one of the necessary need of each and every person around the globe and therefore housing and real estate market is one of the markets which is one of the major contributors in the world’s economy. It is a very large market and there are various companies working in the domain.

- Data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, profits, improving their marketing strategies and focusing on changing trends in house sales and purchases. Predictive modelling, Market mix modelling, recommendation systems are some of the machine learning techniques used for achieving the business goals for housing companies. Our problem is related to one such housing company.

- We are required to model the price of houses with the available independent variables. This model will then be used by the management to understand how exactly the prices vary with the variables. They can accordingly manipulate the strategy of the firm and concentrate on areas that will yield high returns. Further, the model will be a good way for the management to understand the pricing dynamics of a new market

- A US-based housing company named Surprise Housing has decided to enter the Australian market. The company uses data analytics to purchase houses at a price below their actual values and flip them at a higher price. For the same purpose, the company has collected a data set from the sale of houses in Australia.

- The company is looking at prospective properties to buy houses to enter the market. We have to build a model using Machine Learning in order to predict the actual value of the prospective properties and decide whether to invest in them or not. For this company wants to know:
  - Which variables are important to predict the price of variable?
  - How do these variables describe the price of the house?

Analytical Problem Framing

- ## Mathematical/ Analytical Modeling of the Problem

In this project we have performed various mathematical and statistical analysis such as we checked description or statistical summary of the data using describe, checked correlation using corr and also visualized it using heatmap. Then we have used zscore to plot outliers and remove them.

Statistical Summary

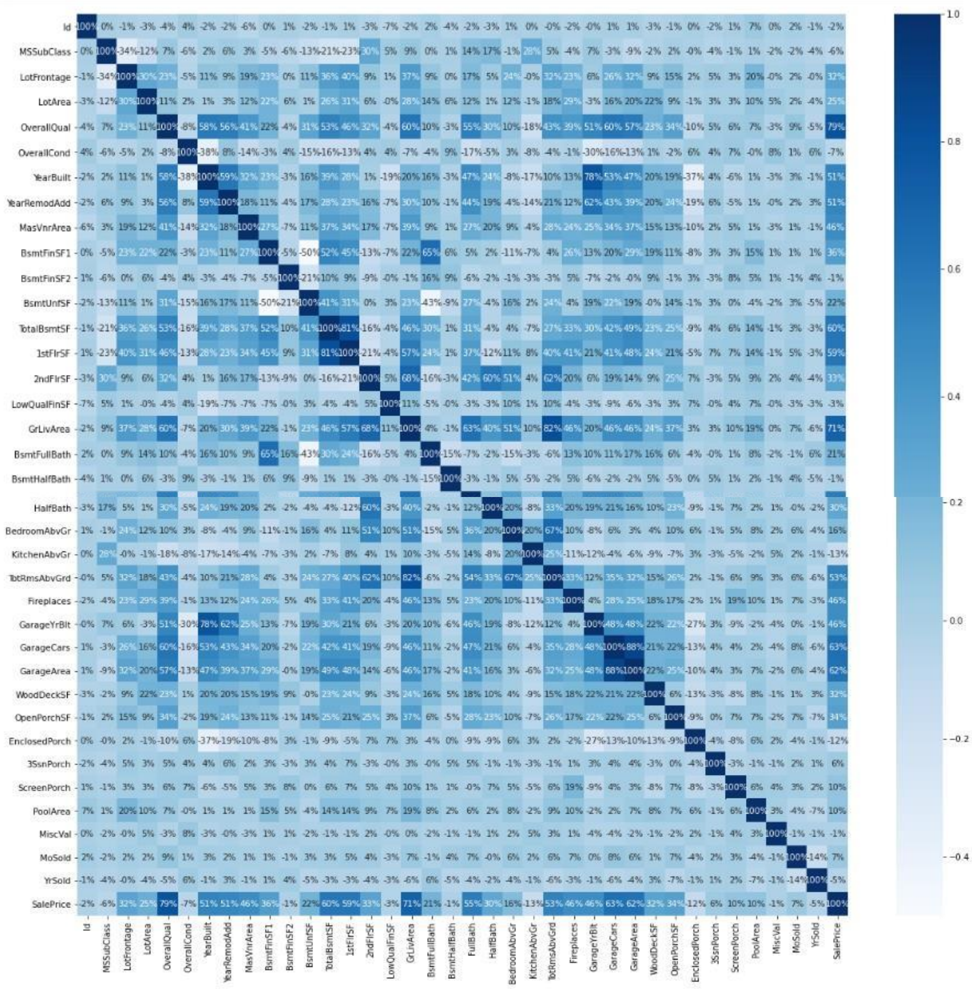
In [18]:
```
#checking the description/summary of the dataset
df.describe()
```

Out[18]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	...	WoodDec
count	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	...	1168.00
mean	724.136130	56.767979	70.988470	10484.749144	6.104452	5.595890	1970.930651	1984.758562	102.310078	444.726027	...	96.20
std	416.159877	41.940650	22.437056	8957.442311	1.390153	1.124343	30.145255	20.785185	182.047152	462.664785	...	126.15
min	1.000000	20.000000	21.000000	1300.000000	1.000000	1.000000	1875.000000	1950.000000	0.000000	0.000000	...	0.00
25%	360.500000	20.000000	60.000000	7621.500000	5.000000	5.000000	1954.000000	1966.000000	0.000000	0.000000	...	0.00
50%	714.500000	50.000000	70.988470	9522.500000	6.000000	5.000000	1972.000000	1993.000000	0.000000	385.500000	...	0.00
75%	1079.500000	70.000000	79.250000	11515.500000	7.000000	6.000000	2000.000000	2004.000000	160.000000	714.500000	...	171.00
max	1460.000000	190.000000	313.000000	164660.000000	10.000000	9.000000	2010.000000	2010.000000	1600.000000	5644.000000	...	857.00

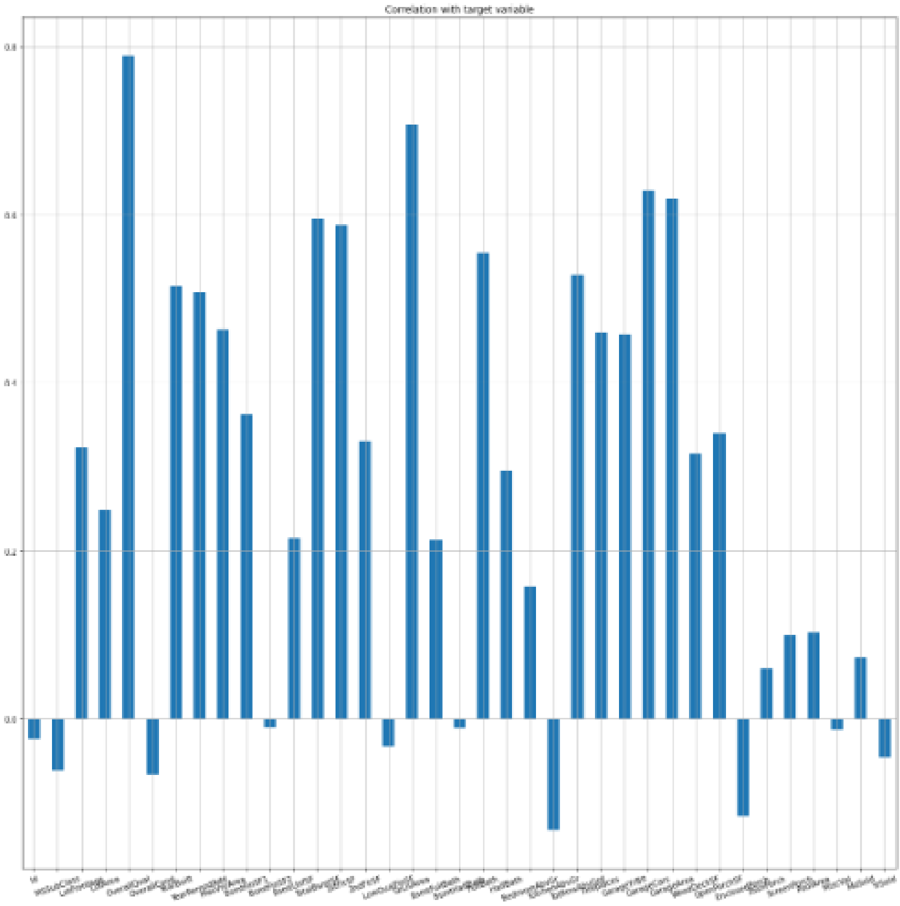
8 rows × 38 columns

```
#checking correlation via visualization (heatmap)
plt.figure(figsize=(20,20))
sns.heatmap(df.corr(),annot=True,fmt='.0%',cmap='Blues')
plt.show()
```



In [21]:
```
plt.figure(figsize=(20,20))
df.drop('SalePrice',axis=1).corrwith(df['SalePrice']).plot(kind='bar',grid=True)
plt.xticks(rotation=20)
plt.title("Correlation with target variable")
```

Out[21]:
```
Text(0.5, 1.0, 'Correlation with target variable')
```



Removing the Outliers using Z-score

In [39]:
```
from scipy.stats import zscore
z=np.abs(zscore(df))
z
```

Out[39]:
```
array([[1.43548658, 1.50830058, 0.02164599, ..., 0.33003329, 0.20793187,
        0.67631017],
       [0.39632483, 0.87704243, 0.02164599, ..., 0.33003329, 0.20793187,
        1.09423443],
       [0.16554544, 0.07709478, 0.02164599, ..., 0.33003329, 0.20793187,
        1.11687211],
       ...,
       [1.26961389, 2.46243779, 0.02164599, ..., 0.33003329, 0.20793187,
        0.41705186],
       [1.66626597, 0.31562908, 4.76211672, ..., 0.33003329, 0.20793187,
        1.78922393],
       [0.25755011, 0.07709478, 0.02164599, ..., 0.33003329, 0.20793187,
        0.02179027]])
```

In [40]:
```
threshold=3
print(np.where(z>3))
```

```
(array([   1,    1,    1, ..., 1166, 1166, 1166], dtype=int64), array([ 8, 19, 33, ..., 38, 61, 62], dtype=int64))
```

In [41]:
```
df_new=df[(z<3).all(axis=1)]
df_new
```

In [42]:
```
df.shape
```

Out[42]:
```
(1168, 75)
```

In [43]:
```
df_new.shape
```

Out[43]:
```
(483, 75)
```

In [ ]:
```
#685 rows have been removed
```

In [44]:
```
df=df_new
```

In [45]:
```
#checking skewness
df.skew()
```

- <u>Data Sources and their formats</u>

The sample data is provided to us from our client database. It is provided in csv format and hence we import it using pandas. Then we further checked more about data using info, checked data types using dtypes, shapes using .shape, columns using .columns, null values using .isnull.sum, and further visualize it through heatmap as follows:

In [1]:
```
#importing the Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.impute import SimpleImputer
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
#Loading the dataset both train and test
df=pd.read_csv('train_housing.csv')
df1=pd.read_csv('test_housing.csv')
```

In [3]:
```
#checking the first five rows of the train dataset
df.head()
```

Out[3]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal
0	127	120	RL	NaN	4928	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0
1	889	20	RL	95.0	15865	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0
2	793	60	RL	92.0	9920	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0
3	110	20	RL	105.0	11751	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0
4	422	20	RL	NaN	16635	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0

5 rows × 81 columns

In [4]:
```
#checking the first five rows of the test dataset
df1.head()
```

Out[4]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	ScreenPorch	PoolArea	PoolQC	Fence	MiscFea
0	337	20	RL	86.0	14157	Pave	NaN	IR1	HLS	AllPub	...	0	0	NaN	NaN	
1	1018	120	RL	NaN	5814	Pave	NaN	IR1	Lvl	AllPub	...	0	0	NaN	NaN	
2	929	20	RL	NaN	11838	Pave	NaN	Reg	Lvl	AllPub	...	0	0	NaN	NaN	
3	1148	70	RL	75.0	12000	Pave	NaN	Reg	Bnk	AllPub	...	0	0	NaN	NaN	
4	1227	60	RL	86.0	14598	Pave	NaN	IR1	Lvl	AllPub	...	0	0	NaN	NaN	

5 rows × 80 columns

In [5]:
```
#checking the information of the train dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1168 entries, 0 to 1167
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             1168 non-null   int64
 1   MSSubClass     1168 non-null   int64
 2   MSZoning       1168 non-null   object
 3   LotFrontage    954 non-null    float64
 4   LotArea        1168 non-null   int64
 5   Street         1168 non-null   object
 6   Alley          77 non-null     object
 7   LotShape       1168 non-null   object
 8   LandContour    1168 non-null   object
 9   Utilities      1168 non-null   object
 10  LotConfig      1168 non-null   object
 11  LandSlope      1168 non-null   object
 12  Neighborhood   1168 non-null   object
 13  Condition1     1168 non-null   object
 14  Condition2     1168 non-null   object
 15  BldgType       1168 non-null   object
 16  HouseStyle     1168 non-null   object
 17  OverallQual    1168 non-null   int64
 18  OverallCond    1168 non-null   int64
 19  YearBuilt      1168 non-null   int64
 20  YearRemodAdd   1168 non-null   int64
 21  RoofStyle      1168 non-null   object
 22  RoofMatl       1168 non-null   object
 23  Exterior1st    1168 non-null   object
 24  Exterior2nd    1168 non-null   object
 25  MasVnrType     1161 non-null   object
 26  MasVnrArea     1161 non-null   float64
 27  ExterQual      1168 non-null   object
 28  ExterCond      1168 non-null   object
 29  Foundation     1168 non-null   object
 30  BsmtQual       1138 non-null   object
 31  BsmtCond       1138 non-null   object
 32  BsmtExposure   1137 non-null   object
 33  BsmtFinType1   1138 non-null   object
 34  BsmtFinSF1     1168 non-null   int64
 35  BsmtFinType2   1137 non-null   object
 36  BsmtFinSF2     1168 non-null   int64
 37  BsmtUnfSF      1168 non-null   int64
 38  TotalBsmtSF    1168 non-null   int64
 39  Heating        1168 non-null   object
 40  HeatingQC      1168 non-null   object
 41  CentralAir     1168 non-null   object
 42  Electrical     1168 non-null   object
 43  1stFlrSF       1168 non-null   int64
 44  2ndFlrSF       1168 non-null   int64
 45  LowQualFinSF   1168 non-null   int64
 46  GrLivArea      1168 non-null   int64
 47  BsmtFullBath   1168 non-null   int64
 48  BsmtHalfBath   1168 non-null   int64
 49  FullBath       1168 non-null   int64
 50  HalfBath       1168 non-null   int64
 51  BedroomAbvGr   1168 non-null   int64
 52  KitchenAbvGr   1168 non-null   int64
 53  KitchenQual    1168 non-null   object
 54  TotRmsAbvGrd   1168 non-null   int64
 55  Functional     1168 non-null   object
 56  Fireplaces     1168 non-null   int64
 57  FireplaceQu    617 non-null    object
 58  GarageType     1104 non-null   object
 59  GarageYrBlt    1104 non-null   float64
 60  GarageFinish   1104 non-null   object
 61  GarageCars     1168 non-null   int64
 62  GarageArea     1168 non-null   int64
 63  GarageQual     1104 non-null   object
 64  GarageCond     1104 non-null   object
 65  PavedDrive     1168 non-null   object
 66  WoodDeckSF     1168 non-null   int64
 67  OpenPorchSF    1168 non-null   int64
 68  EnclosedPorch  1168 non-null   int64
 69  3SsnPorch      1168 non-null   int64
 70  ScreenPorch    1168 non-null   int64
 71  PoolArea       1168 non-null   int64
 72  PoolQC         7 non-null      object
 73  Fence          237 non-null    object
 74  MiscFeature    44 non-null     object
 75  MiscVal        1168 non-null   int64
 76  MoSold         1168 non-null   int64
 77  YrSold         1168 non-null   int64
 78  SaleType       1168 non-null   object
 79  SaleCondition  1168 non-null   object
 80  SalePrice      1168 non-null   int64
dtypes: float64(3), int64(35), object(43)
memory usage: 739.2+ KB
```

In [7]:
```
#checking the shape of train dataset
df.shape
```

Out[7]: (1168, 81)

- <u>Data Preprocessing Done</u>

  First we will determine whether there are any null values and since there were null values as well as NaN vales present in the dataset we proceeded further by imputing them using Simple Imputer with mean and most frequent as strategies respectively. Next we did Label encoding using label encoder. Then we performed some data visualization in which we observed certain attributes were having skewness and outliers that were plotted using distplot and boxplot. Outliers were removed with the help of Zscore in which 685 rows were removed.

- <u>Data Inputs- Logic- Output Relationships</u>

  The data consists of 80 inputs and one output-“SalePrice”. MSSubClass,OverallCond,KitchenAbvGr,EnclosedPorch and Yr Sold are the least/negatively correlated column with target('SalePrice') variable. OverallQual is highly correlated column with target variable followed by GrLivArea and other attributes.

- <u>Hardware and Software Requirements and Tools Used</u>

  In this project we have used HP Pavilion PC with 64-bit operating system and have Windows 10 pro. We have used python to develop this project in which we have used various libraries such as numpy, pandas, matplotlib, seaborn for handling data or arrays and their visualization. For statistical purpose we have used zscore from scipy.stats to remove outliers. Lastly, to develop the model we have used various libraries and metrics from sklearn such as train_test_split, Linear Regression, Lasso, Ridge, Elastic Net, SVR, Decision Tree Regressor, KNeighbors Regressor, Random Forest Regressor, AdaBoost Regressor, Gradient Boosting Regressor, mean_squared_error, mean_absolute_error and r2_score.

```
In [57]: #Importing all the libraries,metrices required for ML
from sklearn.linear_model import LinearRegression,Lasso,Ridge,ElasticNet
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.neighbors import KNeighborsRegressor

from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import AdaBoostRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_squared_error,mean_absolute_error,r2_score
from sklearn.model_selection import train_test_split,GridSearchCV,cross_val_score
```

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  We have performed various mathematical and statistical analysis such as we checked description or statistical summary of the data using describe, checked correlation using corr and also visualized it using heatmap. Then we have used zscore to plot outliers and remove them. We have used distplot to find the distribution of all attributes.

- Testing of Identified Approaches (Algorithms)

  We have used following algorithms such as: LinearRegression, Lasso, Ridge, ElasticNet, SVR, DecisionTreeRegressor, KNeighborsRegressor, RandomForestRegressor, AdaBoostRegressor and GradientBoostingRegressor.

- Run and Evaluate selected models

  We have formed a loop where all the algorithms will be used one by one and their corresponding Score, Mean Absolute Error, Mean Squared Error, RMSE and r2_score will be evaluated.

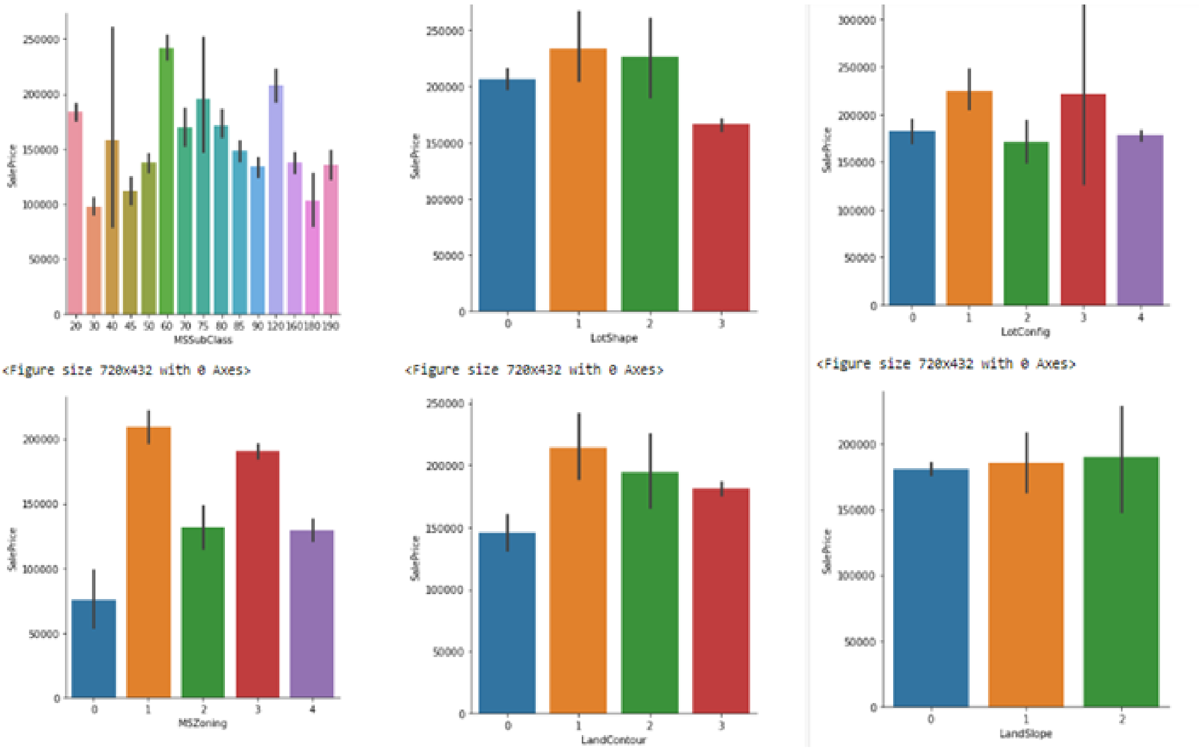
- I chose GradientBoostingRegressor as our best model since it's giving us best score and it's performing well. It's r2_score is also satisfactory and it shows that our model is neither underfitting/overfitting. Then we performed hyperparamter tuning using GridSearchCV on GradientBoostingRegressor from which got 'learning_rate': 0.1, 'n_estimators': 300 as best parameters. We got score : 0.9976896375406933 after performing hyperparameter tuning and earlier it was 0.9846658425719441. Its r2_score is also satisfactory.

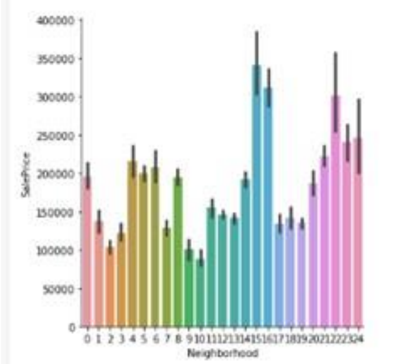
  Hence we saved GradientBoostingRegressor as our final model using joblib.

- <u>Key Metrics for success in solving problem under consideration</u>

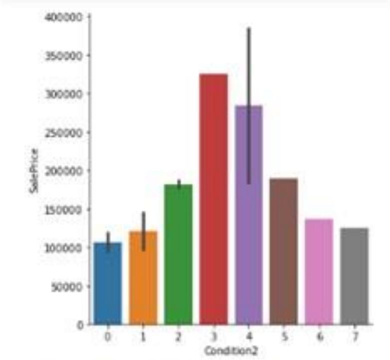
  Key metrics used for finalising the model was Score and r2_score. Since in case of GradientBoostingRegressor it's giving us good score among all other models and it's performing well. It's r2_score is also satisfactory and it shows that our model is neither underfitting/overfitting .

- <u>Visualizations</u>

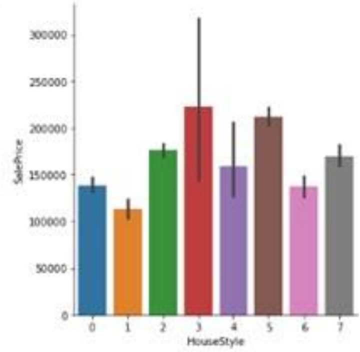




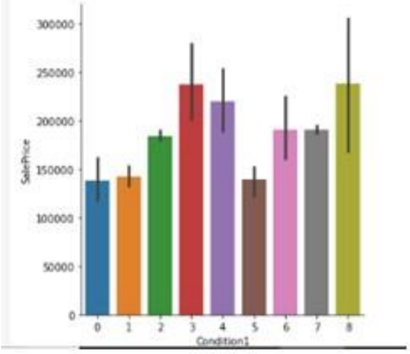
<Figure size 720x432 with 0 Axes>

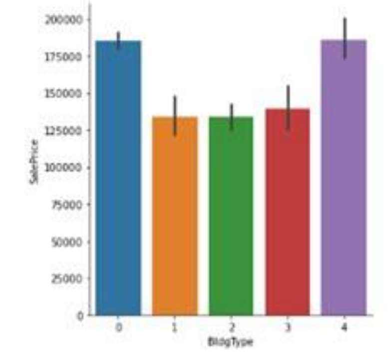


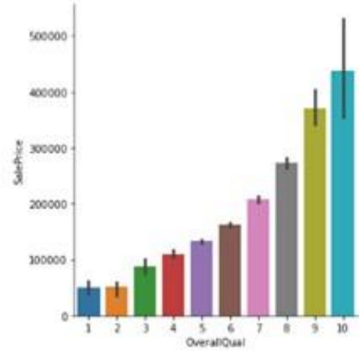
<Figure size 720x432 with 0 Axes>

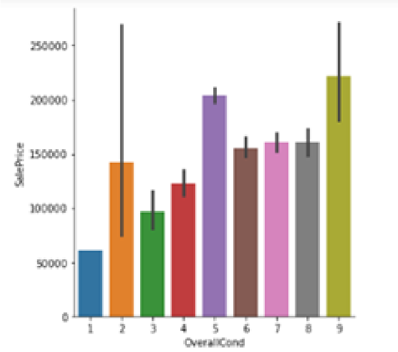


<Figure size 720x432 with 0 Axes>

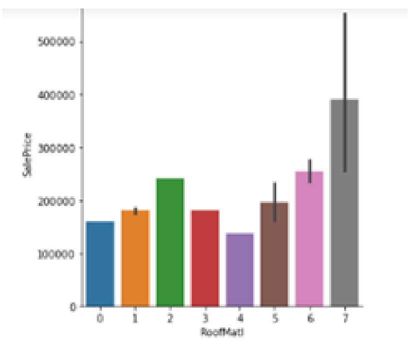




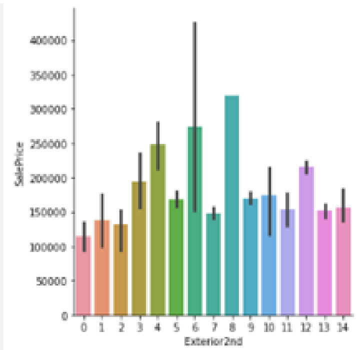




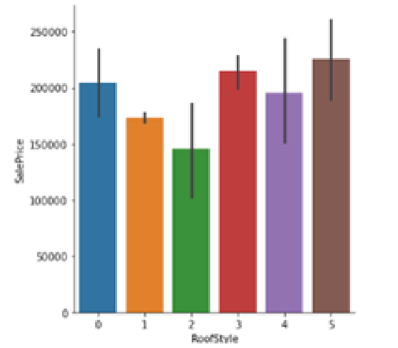
<Figure size 720x432 with 0 Axes>

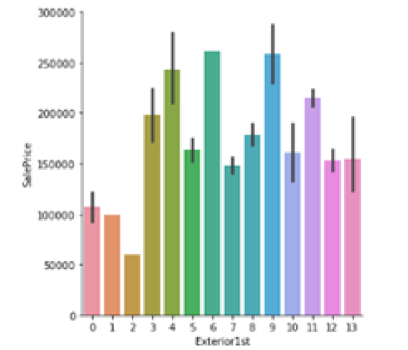


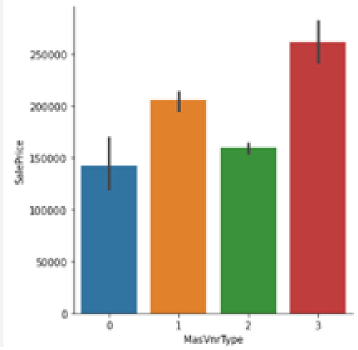
<Figure size 720x432 with 0 Axes>

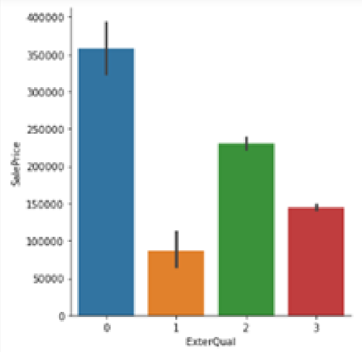


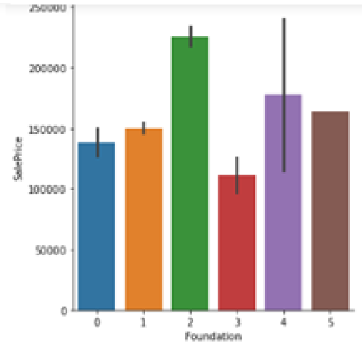
<Figure size 720x432 with 0 Axes>

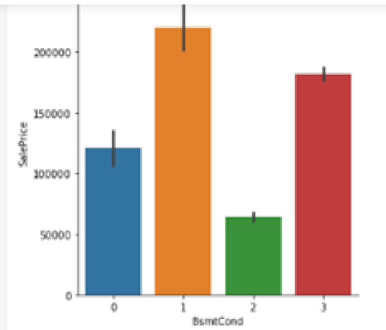








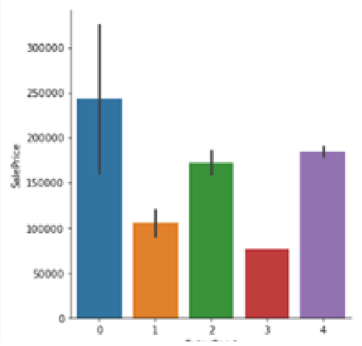


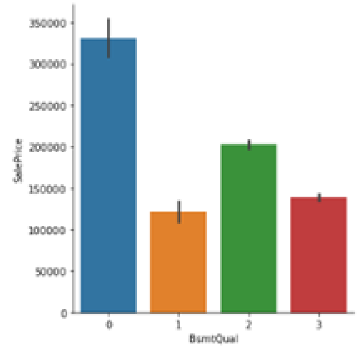


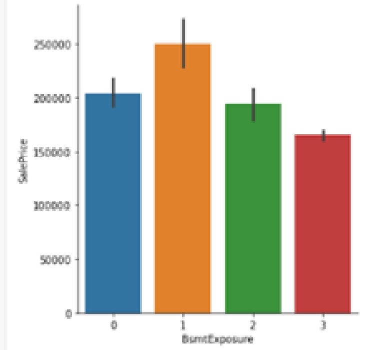
<Figure size 720x432 with 0 Axes>

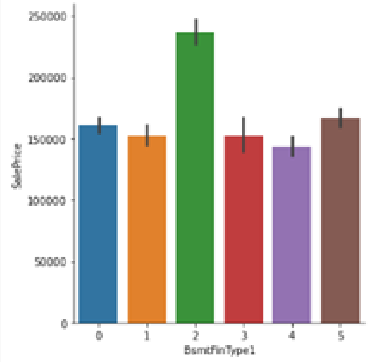
<Figure size 720x432 with 0 Axes>

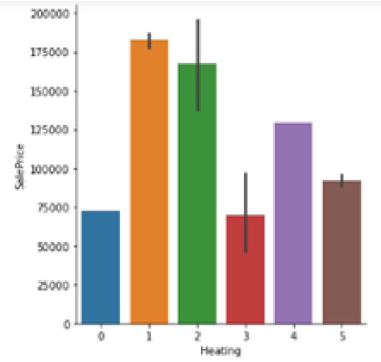
<Figure size 720x432 with 0 Axes>

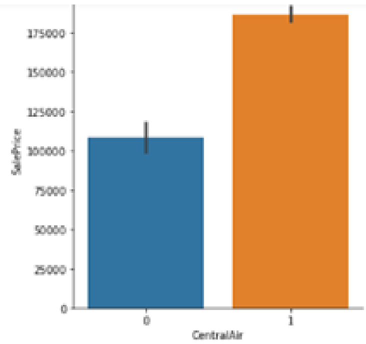








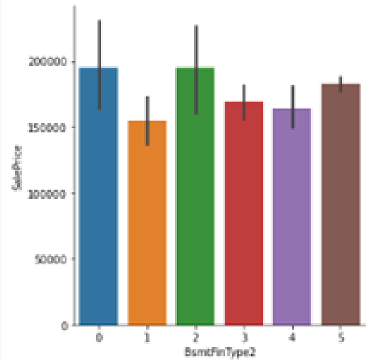


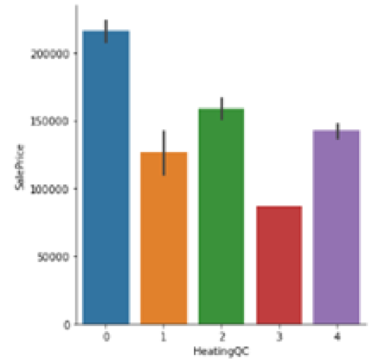


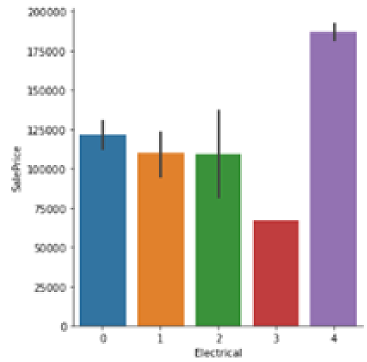
<Figure size 720x432 with 0 Axes>

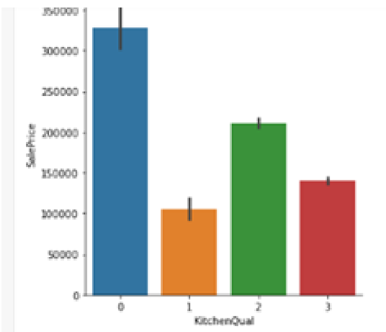
<Figure size 720x432 with 0 Axes>

<Figure size 720x432 with 0 Axes>

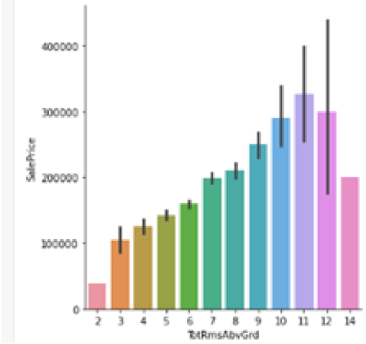


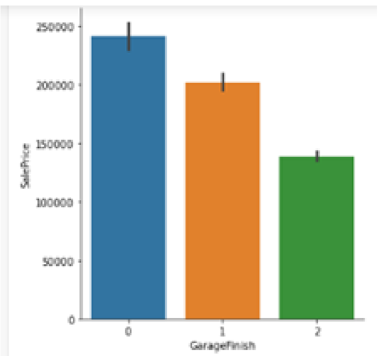




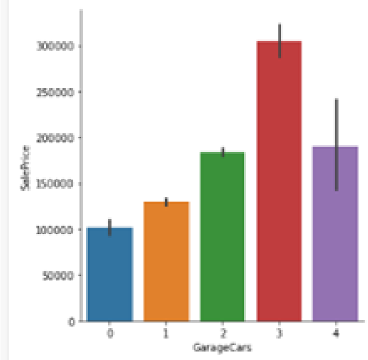


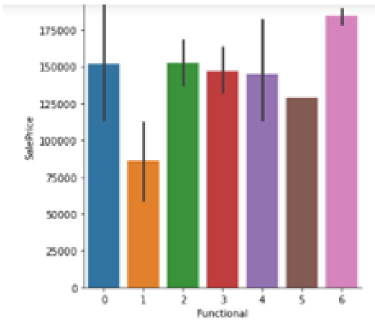
<Figure size 720x432 with 0 Axes>



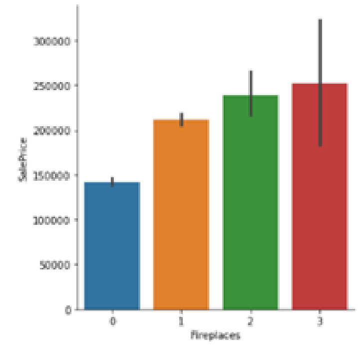


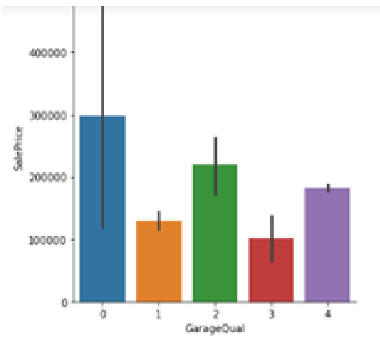
<Figure size 720x432 with 0 Axes>



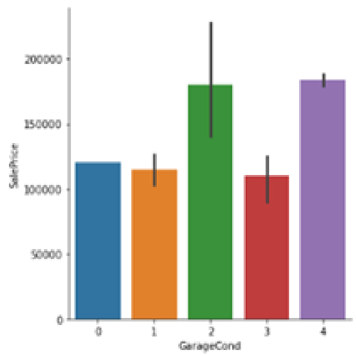


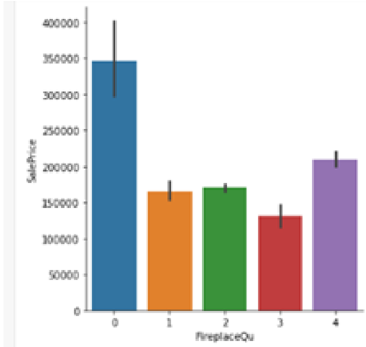
<Figure size 720x432 with 0 Axes>



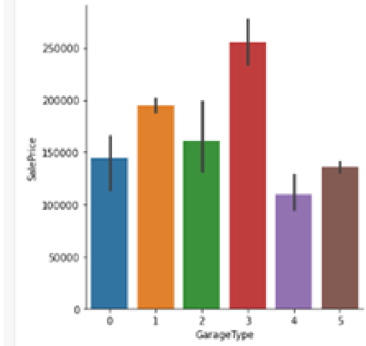


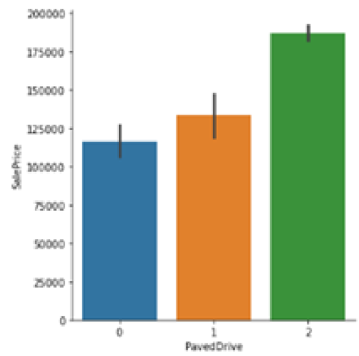
<Figure size 720x432 with 0 Axes>



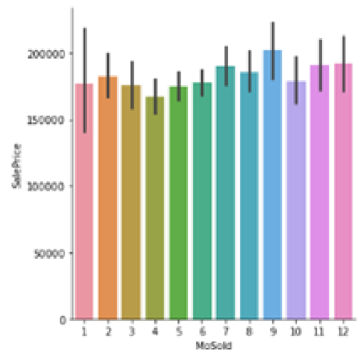


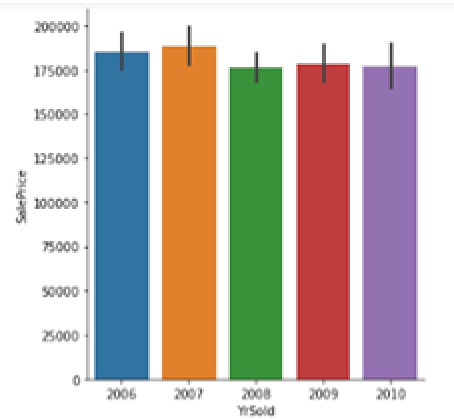
<Figure size 720x432 with 0 Axes>

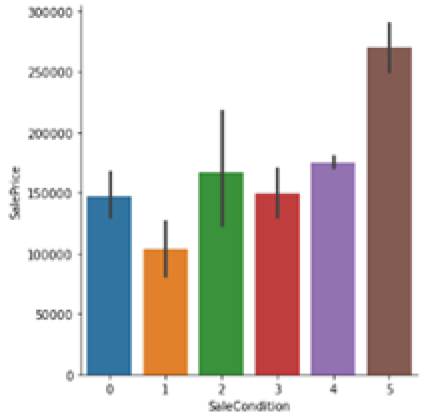




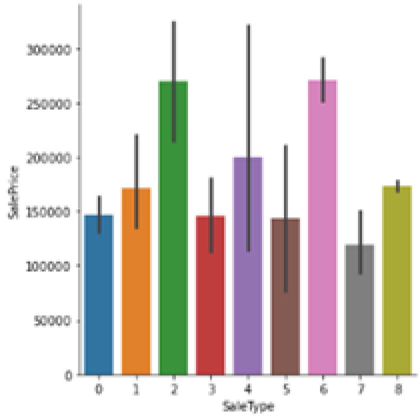
<Figure size 720x432 with 0 Axes>







<Figure size 720x432 with 0 Axes>



- <u>Interpretation of the Results</u>

- Least SalePrice is for 30:1-STORY 1945 & OLDER and maximum for 60:2-STORY 1946 & NEWER
- In MSZoing maximum is for category 1 i.e, Floating Village Residential
- Lotshape 1 and 2 have almost similar price and 3 has least.
- Landconotur corresponding to 1 i.e, HLS Hillside - Significant slope from side to side has maximum price.

- Lotconfig corresponding to 1 and 3 have similar price.

- Neighborhoot with (15)NPkVill Northpark Villa has maximum sales price and (10)IDOTRR Iowa DOT and Rail Road has least.

- Normal condition houses have highest saleprice

- 1Fam Single-family Detached and TwnhsI Townhouse Inside Unit have maximum saleprice.

- In HouseStyle category 3: 2Story Two story has max sale price.
- In OverallQual: SalePrice increase as Ratings increase.
- Similary for OverallCond 5 and 9 have max sale price
- In RoofStyle 5:Shed has maximum.
- In Exterior1st 6:HardBoard and 9:Other have Saleprice
- In Exterior2nd 8:MetalSd Metal Siding
- In MasVnrType, 3:stone has max saleprice and 0:BrkCmn Brick Common has least
- In ExterQual 0:Excellent has maximum price. Similary for ExterCond
- In Foundation 2:PConc Poured Contrete has max price
- In BsmtQual 0: Ex Excellent (100+ inches), In BsmtCond 1: Gd Good, In BsmtExposure 1: Av Average Exposure (split levels or foyers typically score average or above) have max sale prices
- In BsmtFinType1: Rating of basement finished area - 2:GLQ Good Living Quarters has max price
- In HeatingQC: Heating quality and condition 0:Ex Excellent has max price.
- Houses with CentralAir has higher saleprice
- In FireplaceQu: Fireplace quality 0:Ex Excellent - Exceptional Masonry Fireplace has max saleprice
- GarageType 3:BuiltIn Built-In (Garage part of house - typically has room above garage) has max saleprice
- Finished Garage has more price
- Paved Driveway has more price
- In 2007 maximum houses are sold followed by 2006

- In saletype category 2 and 6 have max sale price
- Normal sale condition has max price.

CONCLUSION

- Key Findings and Conclusions of the Study
- Lotshape 1 and 2 have almost similar price and 3 has least.
- Landconotur corresponding to 1 i.e, HLS Hillside - Significant slope from side to side has maximum price.
- Neighborhoot with (15)NPkVill Northpark Villa has maximum sales price and (10)IDOTRR Iowa DOT and Rail Road has least.
- Normal condition houses have highest saleprice
- 1Fam Single-family Detached and TwnhsI Townhouse Inside Unit have maximum saleprice.
- In HouseStyle category 3: 2Story Two story has max sale price.
- In RoofStyle 5:Shed has maximum.
- In Exterior1st 6:HardBoard and 9:Other have Sale price
- In MasVnrType, 3:stone has max sale price and 0:BrkCmn Brick Common has least
- Houses with Centra-lAir has higher saleprice
- GarageType 3:BuiltIn Built-In (Garage part of house - typically has room above garage) has max saleprice
- In 2007 maximum houses are sold followed by 2006
- In LotArea, initially the price keep on increasing as LotArea increases but after 70000 it becomes constant till 160000 and then drops.
- In MasVnrArea, at 1200 saleprice is maximum and then it drops drastically.

- For 1stFlrSF:first floor square feet till 2500 the price is increasing uniformly but after that it decreases and drops after 3000
- For 2ndFlrSF:Second floor square feet the price is increasing as the area increases.

- <u>Learning Outcomes of the Study in respect of Data Science</u>

  With the help of visualization tools such as matplotlib and seaborn we have visualized the impact of each attributes on our target variable. For cleaning the data and plotting outliers we have used distplot and boxplot and for removing outliers we have used zscore which is a statistical tool. At last we got GradientBoostingRegressor as our best model.

- <u>Limitations of this work and Scope for Future Work</u>

  The model is working well and we have performed hyperparameter tuning and we have concluded our project by choosing GradientBoostingRegressor as our best model.