FLIP ROBO

HOUSING: PRICE PREDICTION

Submitted by:

Mahaboob Basha Shaik

ACKNOWLEDGMENT

I would like to express my special thanks of gratitude to my mentor Mr. Shubham Yadav as well as Fliprobo for gave me the golden opportunity to do this wonderful project on the topic price prediction for housing, which also helped me in doing a lot of Research and I came to know about so many new things I am thankful to them. Few research papers, reference link below also helped me in completion of project:

https://www.cse.ust.hk/~rossiter/independent_studies_projects/real_estate_prediction/real_estate_report.pdf

https://www.researchgate.net/publication/340939997_House_Price_Prediction_Using_Various_Regression_Techniques

https://towardsdatascience.com/data-visualization-using-matplotlib-16f1aae5ce70

https://towardsdatascience.com/machine-learning-with-python-regression-complete-tutorial-47268e546cea

INTRODUCTION

- Business Problem Framing

Houses are one of the necessary needs of each 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.

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.

The company is looking at prospective properties to buy houses to enter the market. You are required to build a model using Machine Learning to predict the actual value of the prospective properties and decide whether to invest in them or not.

- Conceptual Background of the Domain Problem

House price forecasting is an important topic of real estate. The literature attempts to derive useful knowledge from historical data of property markets. Machine learning techniques are applied to analyse historical property transactions in India to discover useful models for house buyers and sellers. Revealed is the high discrepancy between house prices in the most expensive and most affordable suburbs in the city of Australia. Moreover, experiments demonstrate that the Multiple Linear Regression that is based on mean squared error measurement is a competitive approach.

- Review of Literature

Machine learning is a form of artificial intelligence which compose available computers with the efficiency to be trained without being veraciously programmed. Machine learning interest on the extensions of computer programs which is capable enough to modify when unprotected to new-fangled data.

Machine learning has many applications out of which one of the applications is prediction of real estate. The real estate market is one of the most competitive in terms of pricing and same tends to be vary significantly based on lots of factor, forecasting property price is an important modules in decision making for both the buyers and investors in supporting budget allocation, finding property finding stratagems and determining suitable policies hence it becomes one of the prime fields to apply the concepts of machine learning to optimize and predict the prices with high accuracy.

- Motivation for the Problem Undertaken

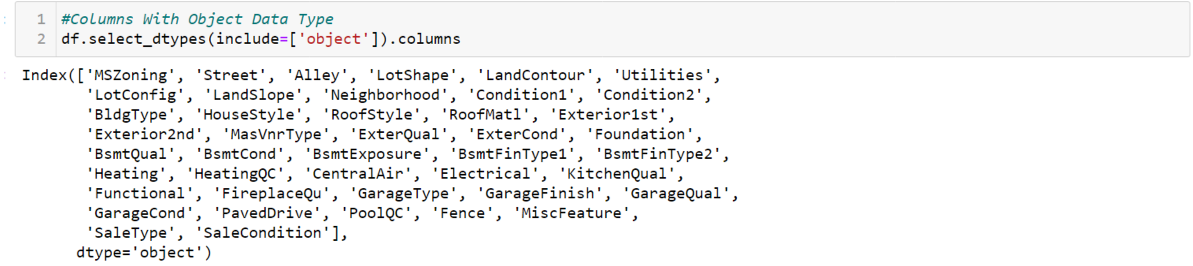
Being extremely interested in everything having a relation with the Machine Learning, the independent project was a great occasion to give me the time to learn and confirm my interest for this field. The fact that we can make estimations, predictions and give the ability for machines to learn by themselves is both powerful and limitless in term of application possibilities. We can use Machine Learning in Finance, Retail, Medicine, almost everywhere. That is why I decided to conduct my project around the Machine Learning.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem
  a. Pandas: The Python Data Analysis Library is used for storing the data in data frames and manipulation.
  b. NumPy: Python scientific computing library.
  c. Matplotlib: Python plotting library.
  d. Seaborn: Statistical data visualization based on matplotlib.
  e. Scikit-learn: Sklearn is a machine learning library for Python.
  f. SciPy. Stats: Provides a number of probability distributions and statistical functions.

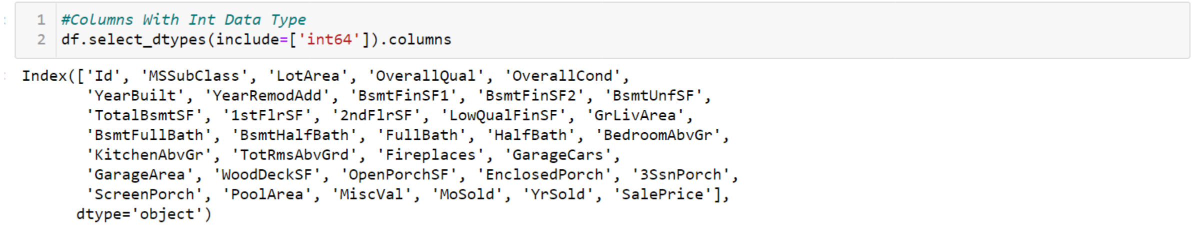
- Data Sources and their formats

  For the same purpose, the company has collected a data set from the sale of houses in Australia. The data is provided in the CSV file. Please find the data description below:



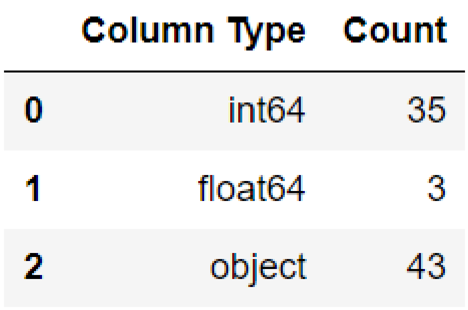
```
#Columns With Object Data Type
df.select_dtypes(include=['object']).columns
```

```
Index(['MSZoning', 'Street', 'Alley', 'LotShape', 'LandContour', 'Utilities',
       'LotConfig', 'LandSlope', 'Neighborhood', 'Condition1', 'Condition2',
       'BldgType', 'HouseStyle', 'RoofStyle', 'RoofMatl', 'Exterior1st',
       'Exterior2nd', 'MasVnrType', 'ExterQual', 'ExterCond', 'Foundation',
       'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2',
       'Heating', 'HeatingQC', 'CentralAir', 'Electrical', 'KitchenQual',
       'Functional', 'FireplaceQu', 'GarageType', 'GarageFinish', 'GarageQual',
       'GarageCond', 'PavedDrive', 'PoolQC', 'Fence', 'MiscFeature',
       'SaleType', 'SaleCondition'],
      dtype='object')
```



```
#Columns With Int Data Type
df.select_dtypes(include=['int64']).columns
```

```
Index(['Id', 'MSSubClass', 'LotArea', 'OverallQual', 'OverallCond',
       'YearBuilt', 'YearRemodAdd', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF',
       'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF', 'GrLivArea',
       'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'HalfBath', 'BedroomAbvGr',
       'KitchenAbvGr', 'TotRmsAbvGrd', 'Fireplaces', 'GarageCars',
       'GarageArea', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch',
       'ScreenPorch', 'PoolArea', 'MiscVal', 'MoSold', 'YrSold', 'SalePrice'],
      dtype='object')
```



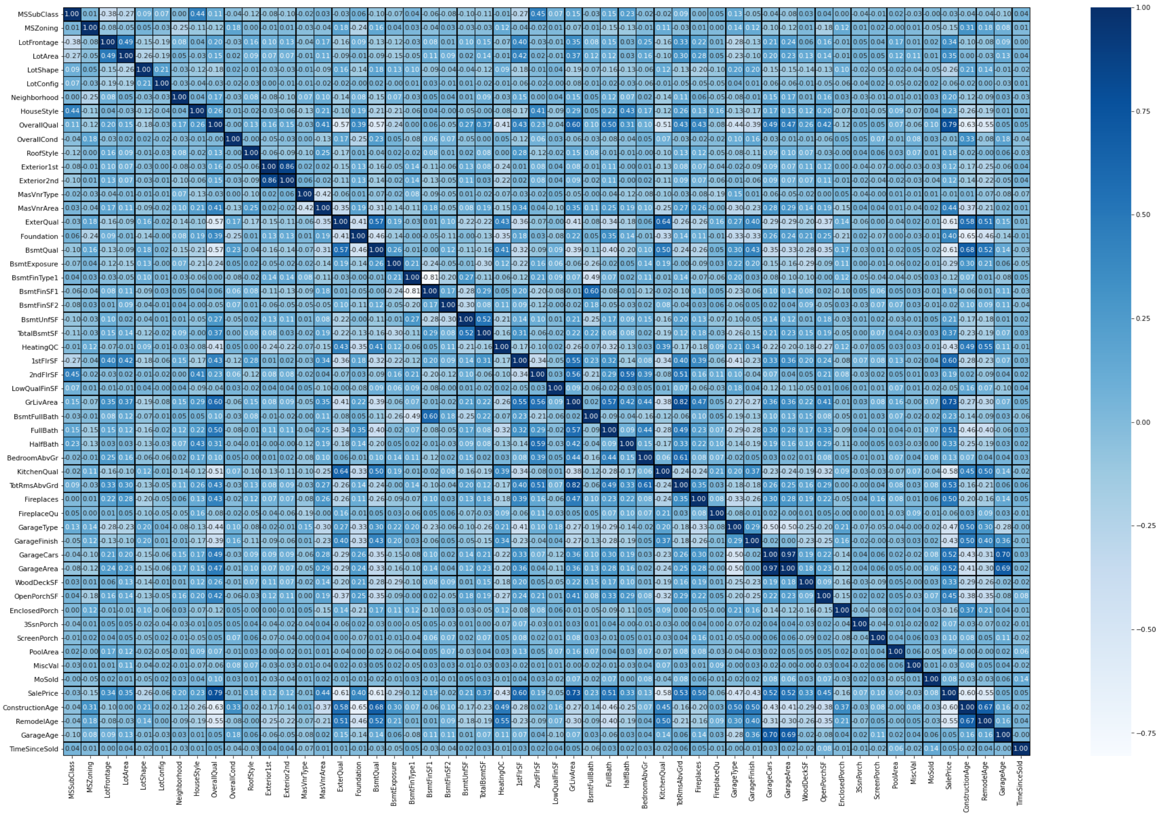
	Column Type	Count
0	int64	35
1	float64	3
2	object	43

- Data Pre-processing

  Cleaning is the first module called to clean the item and verify that all the information in it correspond to the pattern used to extract it. For instance, suppose that we have several estates on a given web page. Each estate is presented using images and many basic information such as, the gross area, the saleable area, the price, etc. When the spider extracts them, it is not unlikely that some noise (like extra characters) was present with the value, or just the value does not exist. The cleaning module removes the noise, and check that all the values are not empty, otherwise the item is dropped. This is done for simplicity; indeed, it could be better to try to inference them later. After the cleaning part done, the item is sent to the formatting module.

Data exploration is the first step in data analysis and typically involves summarizing the main characteristics of a data set, including its size, accuracy, initial patterns in the data and other attributes. It is commonly conducted by data analysts using visual analytics tools, but it can also be done in more advanced statistical software, Python**.** Before it can conduct analysis on data collected by multiple data sources and stored in data warehouses, an organization must know how many cases are in a data set, what variables are included, how many missing values there are and what general hypotheses the data is likely to support. An initial exploration of the data set can help answer these questions by familiarizing analysts with the data with which they are working.

- Data Inputs- Logic- Output Relationships



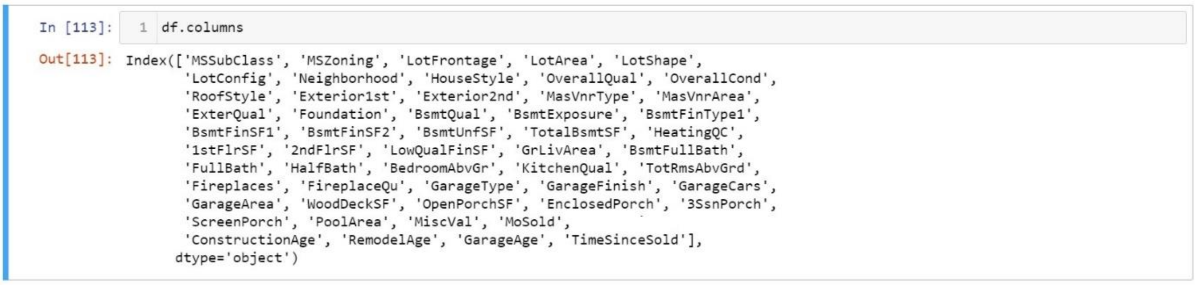
- Hardware and Software Requirements and Tools Used
    a. Pandas: The Python Data Analysis Library is used for storing the data in data frames and manipulation.
    b. NumPy: Python scientific computing library.
    c. Matplotlib: Python plotting library.
    d. Seaborn: Statistical data visualization based on matplotlib.
    e. Scikit-learn: Sklearn is a machine learning library for Python.
    f. SciPy. Stats: Provides a number of probability distributions and statistical functions.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)
  a. Yeo-Johnson is used to scaling of the data.
  b. Pandas: The Python Data Analysis Library is used for storing the data in data frames and manipulation.
  c. NumPy: Python scientific computing library.
  d. Matplotlib: Python plotting library.
  e. Seaborn: Statistical data visualization based on matplotlib.
  f. Scikit-learn: Sklearn is a machine learning library for Python.
  g. SciPy. Stats: Provides a number of probability distributions and statistical functions.
  h. Decision Tree, Random Forest, SVR, Lasso, Ridge Model were uses to predict the model.
  i. Cross Validation and R2 Score were used to identify the models.

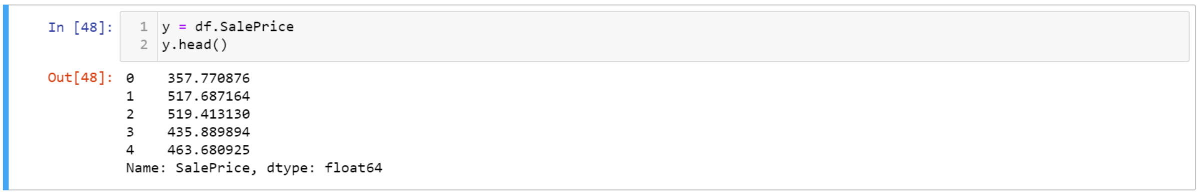
- Testing of Identified Approaches (Algorithms)

X – Variable:



```
In [113]:  df.columns

Out[113]: Index(['MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'LotShape',
                 'LotConfig', 'Neighborhood', 'HouseStyle', 'OverallQual', 'OverallCond',
                 'RoofStyle', 'Exterior1st', 'Exterior2nd', 'MasVnrType', 'MasVnrArea',
                 'ExterQual', 'Foundation', 'BsmtQual', 'BsmtExposure', 'BsmtFinType1',
                 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'HeatingQC',
                 '1stFlrSF', '2ndFlrSF', 'LowQualFinSF', 'GrLivArea', 'BsmtFullBath',
                 'FullBath', 'HalfBath', 'BedroomAbvGr', 'KitchenQual', 'TotRmsAbvGrd',
                 'Fireplaces', 'FireplaceQu', 'GarageType', 'GarageFinish', 'GarageCars',
                 'GarageArea', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch',
                 'ScreenPorch', 'PoolArea', 'MiscVal', 'MoSold',
                 'ConstructionAge', 'RemodelAge', 'GarageAge', 'TimeSinceSold'],
                dtype='object')
```

Y-Variable:

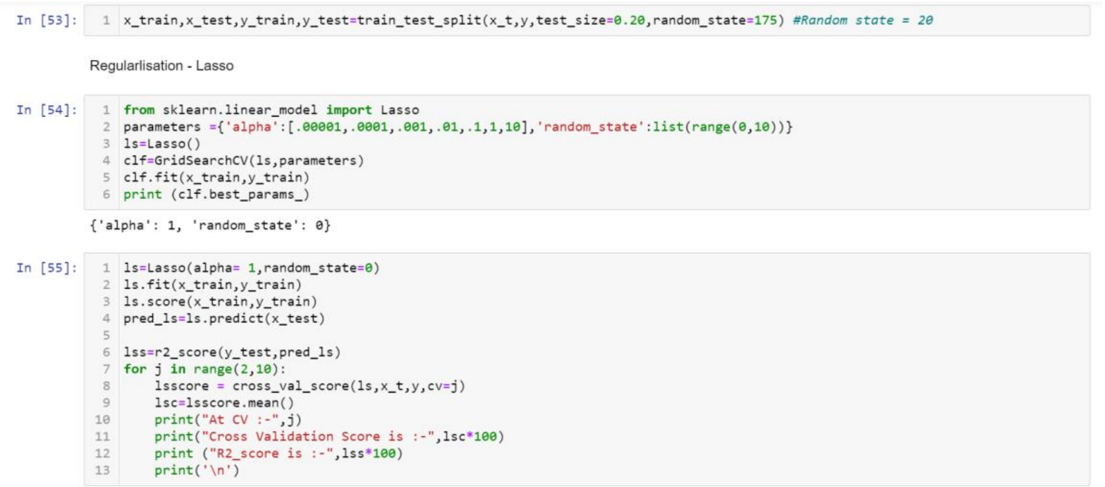


```
In [48]:  y = df.SalePrice
          y.head()

Out[48]:  0    357.770876
          1    517.687164
          2    519.413130
          3    435.889894
          4    463.680925
          Name: SalePrice, dtype: float64
```

Decision Tree, Random Forest, SVR, Lasso, Ridge Model were uses to predict the model.

- Run and evaluate selected models:

a) Lasso Model

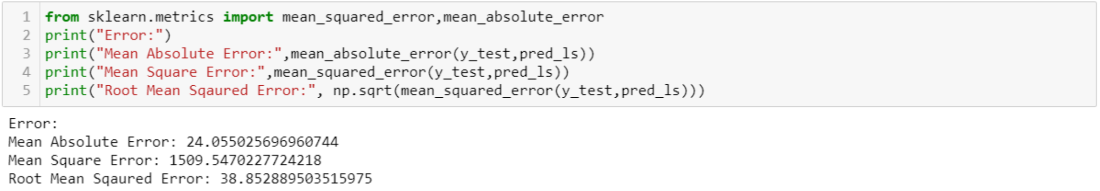


```
In [53]: x_train,x_test,y_train,y_test=train_test_split(x_t,y,test_size=0.20,random_state=175) #Random state = 20
```

Regularlisation - Lasso

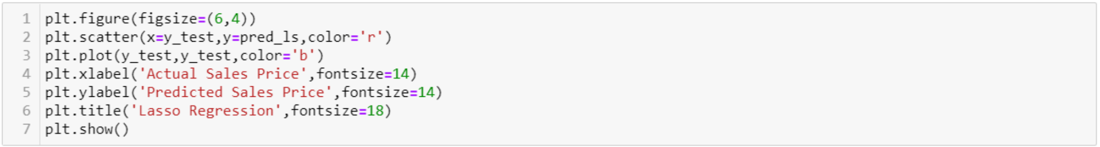
```
In [54]: from sklearn.linear_model import Lasso
         parameters ={'alpha':[.00001,.0001,.001,.01,.1,1,10],'random_state':list(range(0,10))}
         ls=Lasso()
         clf=GridSearchCV(ls,parameters)
         clf.fit(x_train,y_train)
         print (clf.best_params_)

         {'alpha': 1, 'random_state': 0}
```

```
In [55]: ls=Lasso(alpha= 1,random_state=0)
         ls.fit(x_train,y_train)
         ls.score(x_train,y_train)
         pred_ls=ls.predict(x_test)

         lss=r2_score(y_test,pred_ls)
         for j in range(2,10):
             lsscore = cross_val_score(ls,x_t,y,cv=j)
             lsc=lsscore.mean()
             print("At CV :-",j)
             print("Cross Validation Score is :-",lsc*100)
             print ("R2_score is :-",lss*100)
             print('\n')
```

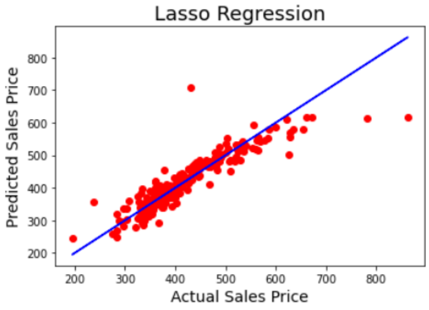


```
from sklearn.metrics import mean_squared_error,mean_absolute_error
print("Error:")
print("Mean Absolute Error:",mean_absolute_error(y_test,pred_ls))
print("Mean Square Error:",mean_squared_error(y_test,pred_ls))
print("Root Mean Sqaured Error:", np.sqrt(mean_squared_error(y_test,pred_ls)))
```

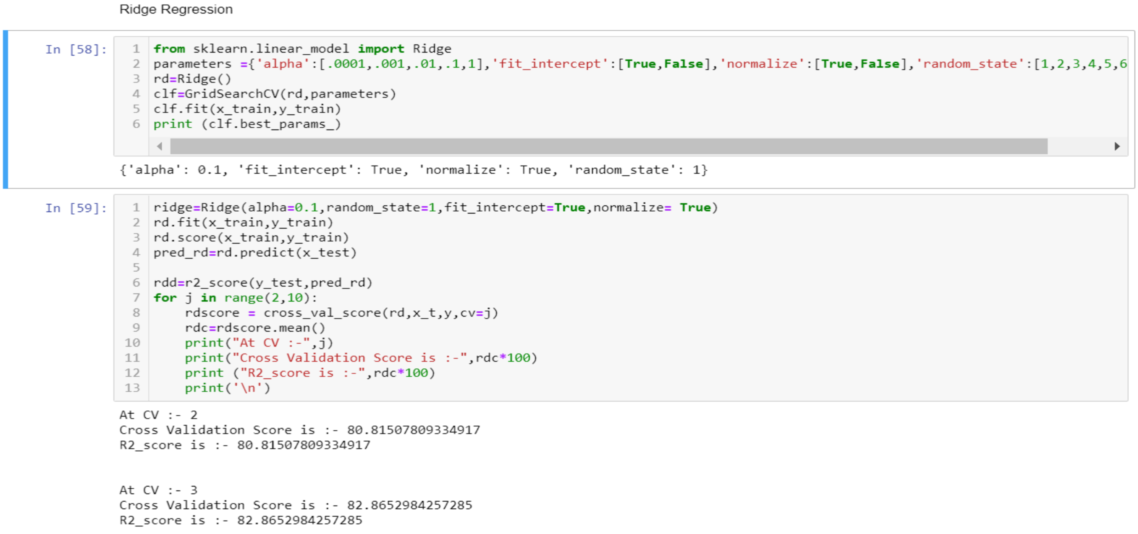
```
Error:
Mean Absolute Error: 24.055025696960744
Mean Square Error: 1509.5470227724218
Root Mean Sqaured Error: 38.852889503515975
```



```
plt.figure(figsize=(6,4))
plt.scatter(x=y_test,y=pred_ls,color='r')
plt.plot(y_test,y_test,color='b')
plt.xlabel('Actual Sales Price',fontsize=14)
plt.ylabel('Predicted Sales Price',fontsize=14)
plt.title('Lasso Regression',fontsize=18)
plt.show()
```



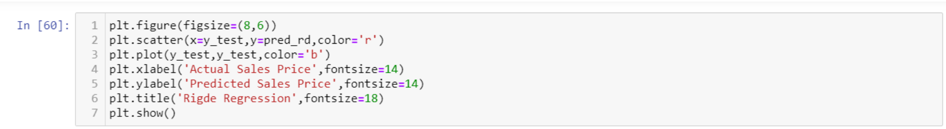
b) Ridge Regression:



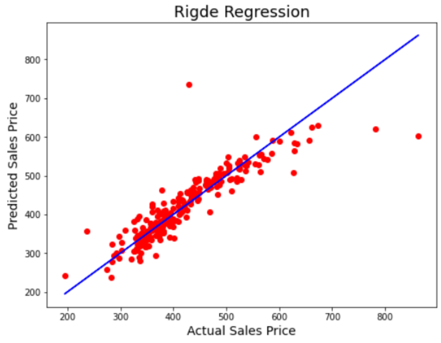
Ridge Regression

```
In [58]: from sklearn.linear_model import Ridge
         parameters ={'alpha':[.0001,.001,.01,.1,1],'fit_intercept':[True,False],'normalize':[True,False],'random_state':[1,2,3,4,5,6
         rd=Ridge()
         clf=GridSearchCV(rd,parameters)
         clf.fit(x_train,y_train)
         print (clf.best_params_)

         {'alpha': 0.1, 'fit_intercept': True, 'normalize': True, 'random_state': 1}
```

```
In [59]: ridge=Ridge(alpha=0.1,random_state=1,fit_intercept=True,normalize= True)
         rd.fit(x_train,y_train)
         rd.score(x_train,y_train)
         pred_rd=rd.predict(x_test)

         rdd=r2_score(y_test,pred_rd)
         for j in range(2,10):
             rdscore = cross_val_score(rd,x_t,y,cv=j)
             rdc=rdscore.mean()
             print("At CV :-",j)
             print("Cross Validation Score is :-",rdc*100)
             print ("R2_score is :-",rdc*100)
             print('\n')

         At CV :- 2
         Cross Validation Score is :- 80.81507809334917
         R2_score is :- 80.81507809334917


         At CV :- 3
         Cross Validation Score is :- 82.8652984257285
         R2_score is :- 82.8652984257285
```

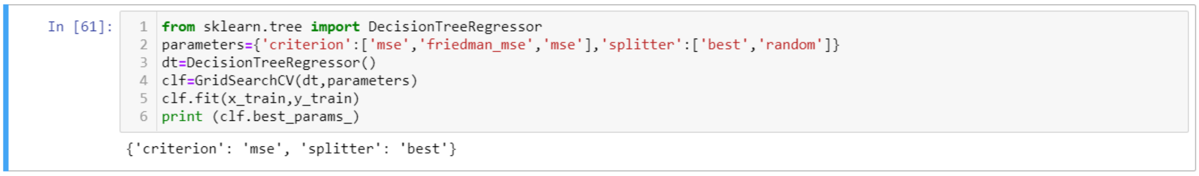


```
In [60]:
plt.figure(figsize=(8,6))
plt.scatter(x=y_test,y=pred_rd,color='r')
plt.plot(y_test,y_test,color='b')
plt.xlabel('Actual Sales Price',fontsize=14)
plt.ylabel('Predicted Sales Price',fontsize=14)
plt.title('Rigde Regression',fontsize=18)
plt.show()
```



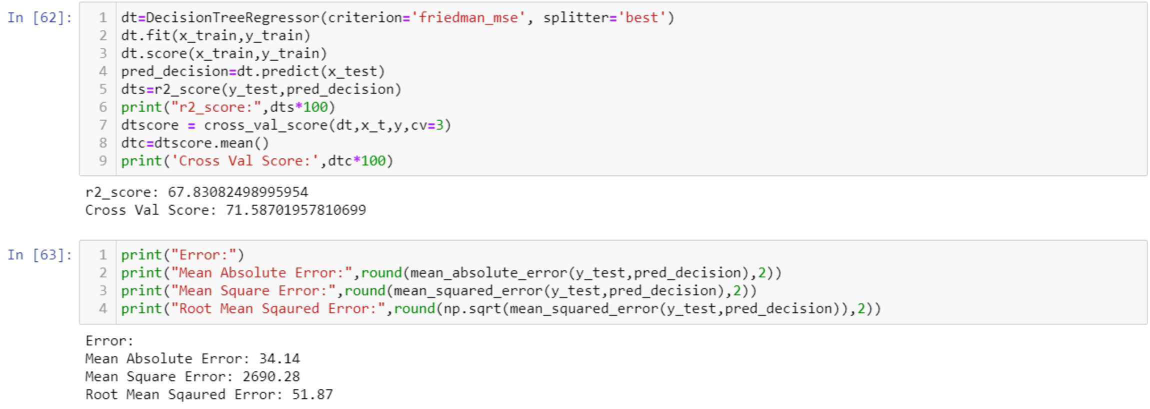
Same as Lasso Model all Data points are interacting with the predictd line

c) Decision Tree:



```
In [61]:
from sklearn.tree import DecisionTreeRegressor
parameters={'criterion':['mse','friedman_mse','mse'],'splitter':['best','random']}
dt=DecisionTreeRegressor()
clf=GridSearchCV(dt,parameters)
clf.fit(x_train,y_train)
print (clf.best_params_)
```

```
{'criterion': 'mse', 'splitter': 'best'}
```

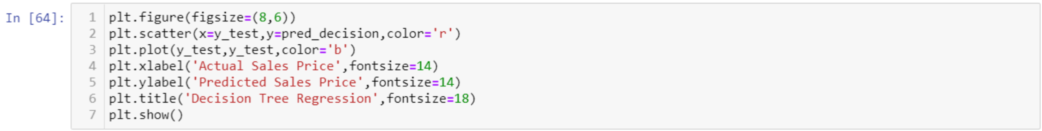


```
In [62]:
dt=DecisionTreeRegressor(criterion='friedman_mse', splitter='best')
dt.fit(x_train,y_train)
dt.score(x_train,y_train)
pred_decision=dt.predict(x_test)
dts=r2_score(y_test,pred_decision)
print("r2_score:",dts*100)
dtscore = cross_val_score(dt,x_t,y,cv=3)
dtc=dtscore.mean()
print('Cross Val Score:',dtc*100)
```

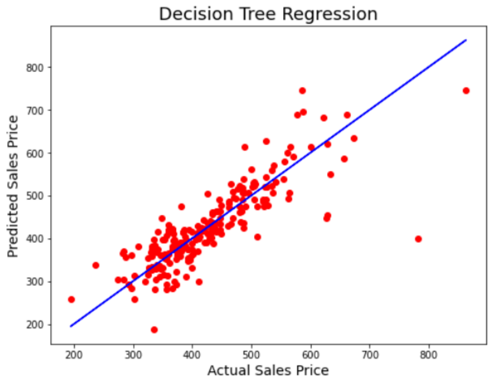
```
r2_score: 67.83082498995954
Cross Val Score: 71.58701957810699
```

```
In [63]:
print("Error:")
print("Mean Absolute Error:",round(mean_absolute_error(y_test,pred_decision),2))
print("Mean Square Error:",round(mean_squared_error(y_test,pred_decision),2))
print("Root Mean Sqaured Error:",round(np.sqrt(mean_squared_error(y_test,pred_decision)),2))
```

```
Error:
Mean Absolute Error: 34.14
Mean Square Error: 2690.28
Root Mean Sqaured Error: 51.87
```

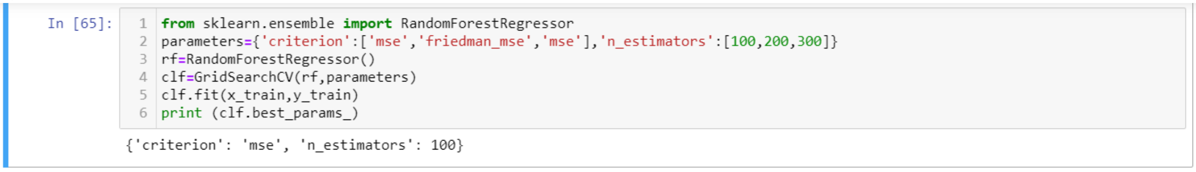


```
In [64]:
plt.figure(figsize=(8,6))
plt.scatter(x=y_test,y=pred_decision,color='r')
plt.plot(y_test,y_test,color='b')
plt.xlabel('Actual Sales Price',fontsize=14)
plt.ylabel('Predicted Sales Price',fontsize=14)
plt.title('Decision Tree Regression',fontsize=18)
plt.show()
```



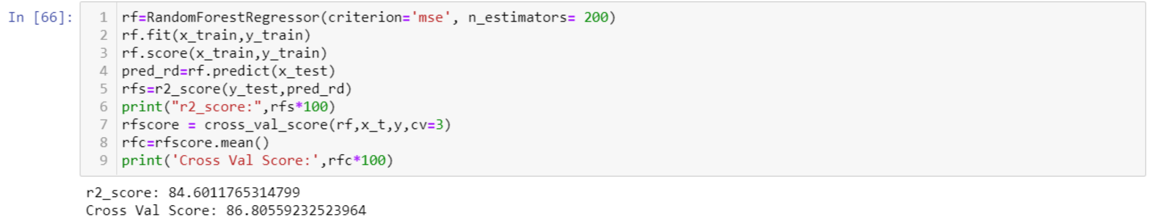
Very Less Data Points Are Intracting with the predicted Line

d) Random Forest:



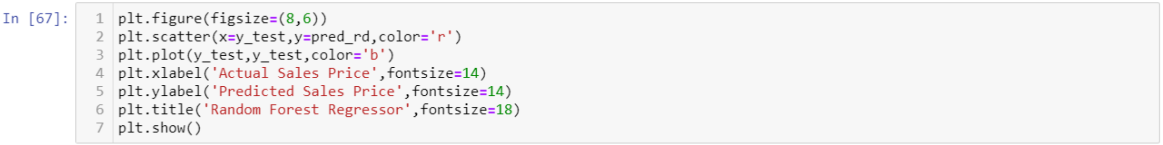
```
In [65]:
from sklearn.ensemble import RandomForestRegressor
parameters={'criterion':['mse','friedman_mse','mse'],'n_estimators':[100,200,300]}
rf=RandomForestRegressor()
clf=GridSearchCV(rf,parameters)
clf.fit(x_train,y_train)
print (clf.best_params_)
```

```
{'criterion': 'mse', 'n_estimators': 100}
```

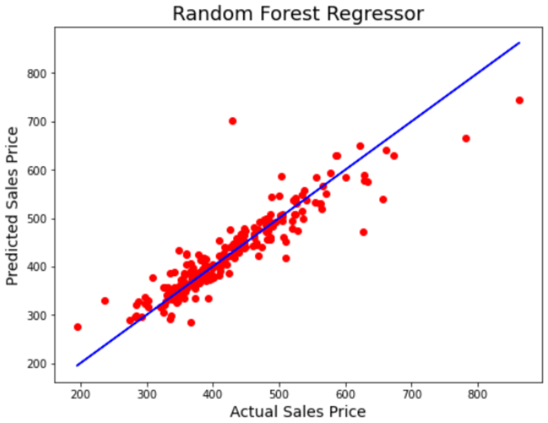


```
In [66]:
rf=RandomForestRegressor(criterion='mse', n_estimators= 200)
rf.fit(x_train,y_train)
rf.score(x_train,y_train)
pred_rd=rf.predict(x_test)
rfs=r2_score(y_test,pred_rd)
print("r2_score:",rfs*100)
rfscore = cross_val_score(rf,x_t,y,cv=3)
rfc=rfscore.mean()
print('Cross Val Score:',rfc*100)
```

```
r2_score: 84.6011765314799
Cross Val Score: 86.80559232523964
```

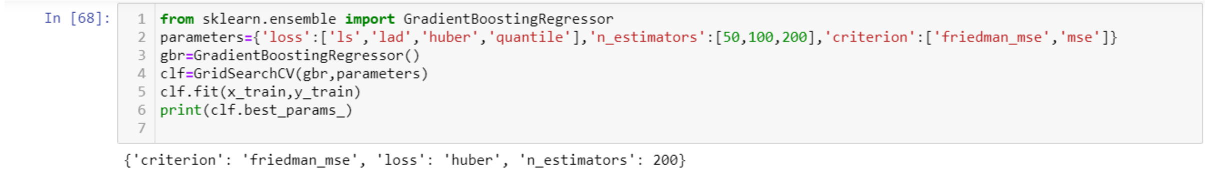


```
In [67]:
plt.figure(figsize=(8,6))
plt.scatter(x=y_test,y=pred_rd,color='r')
plt.plot(y_test,y_test,color='b')
plt.xlabel('Actual Sales Price',fontsize=14)
plt.ylabel('Predicted Sales Price',fontsize=14)
plt.title('Random Forest Regressor',fontsize=18)
plt.show()
```



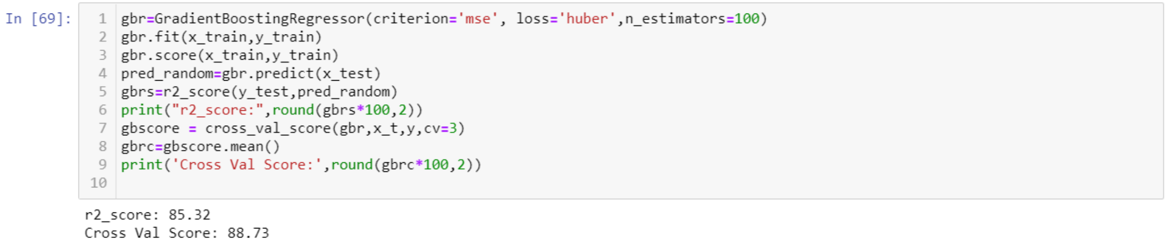
More Data points are intracting with the predicted lines when compare with above models

e) Gradient SearchCV:



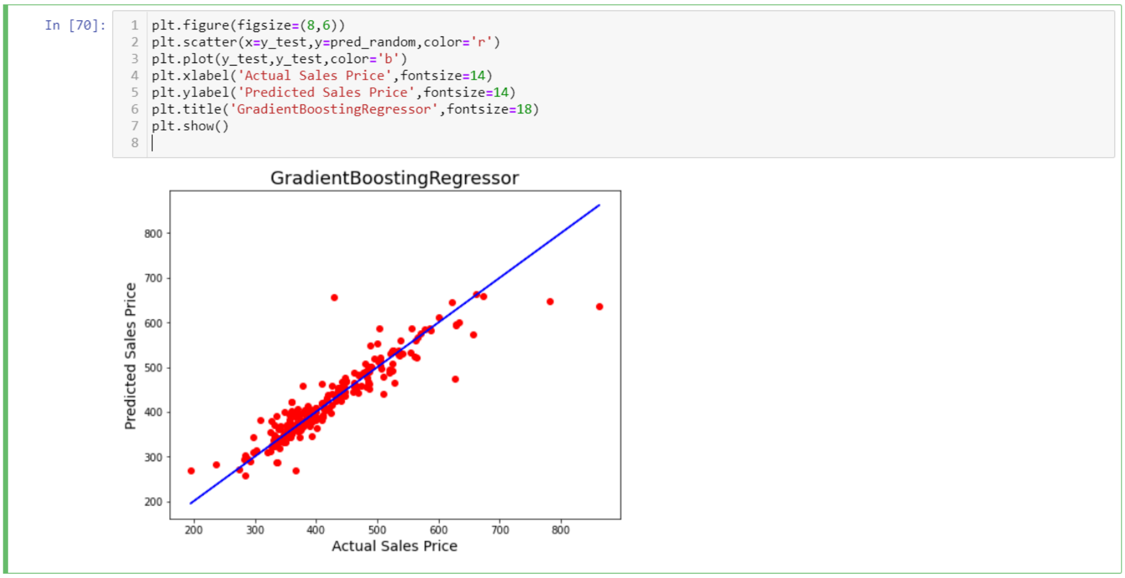
```
In [68]:
from sklearn.ensemble import GradientBoostingRegressor
parameters={'loss':['ls','lad','huber','quantile'],'n_estimators':[50,100,200],'criterion':['friedman_mse','mse']}
gbr=GradientBoostingRegressor()
clf=GridSearchCV(gbr,parameters)
clf.fit(x_train,y_train)
print(clf.best_params_)
```

```
{'criterion': 'friedman_mse', 'loss': 'huber', 'n_estimators': 200}
```



```
In [69]:
gbr=GradientBoostingRegressor(criterion='mse', loss='huber',n_estimators=100)
gbr.fit(x_train,y_train)
gbr.score(x_train,y_train)
pred_random=gbr.predict(x_test)
gbrs=r2_score(y_test,pred_random)
print("r2_score:",round(gbrs*100,2))
gbscore = cross_val_score(gbr,x_t,y,cv=3)
gbrc=gbscore.mean()
print('Cross Val Score:',round(gbrc*100,2))
```

```
r2_score: 85.32
Cross Val Score: 88.73
```



In [70]:

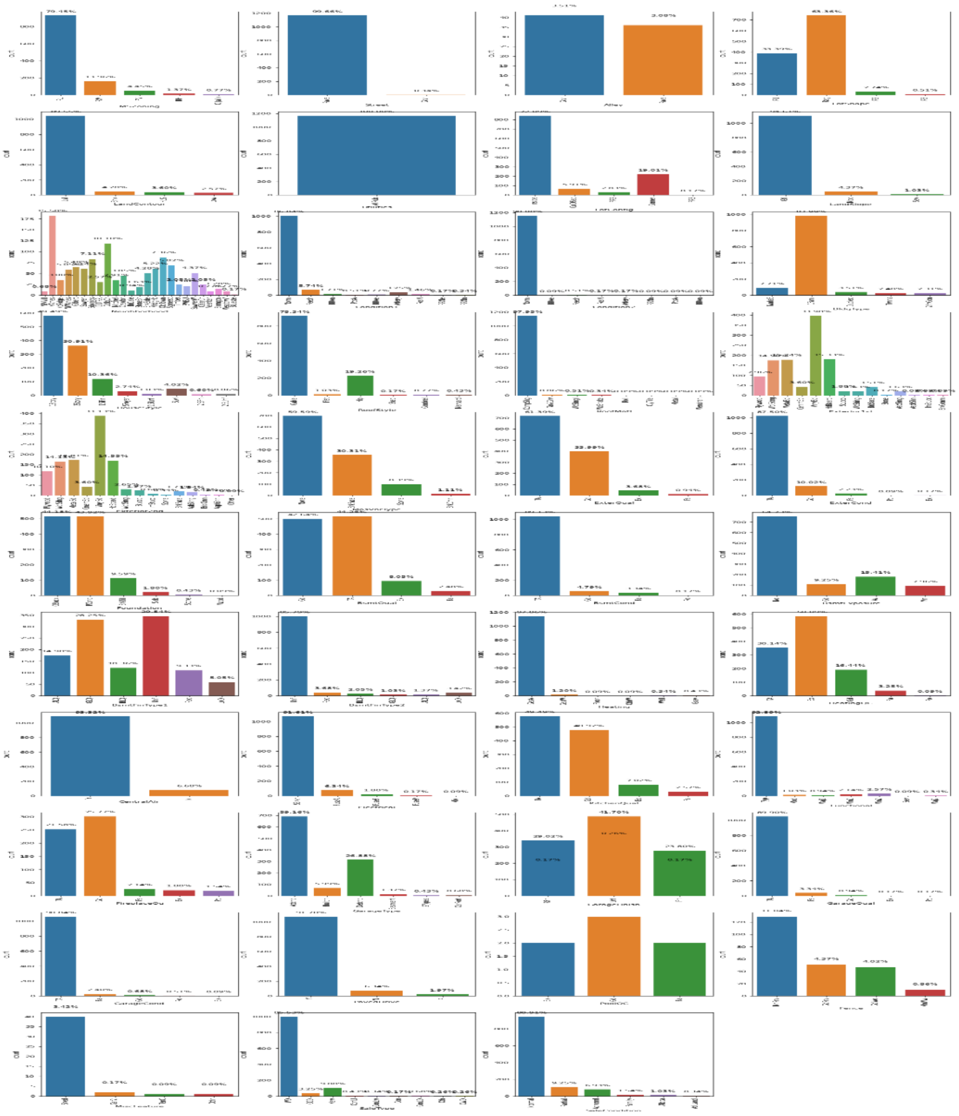
```
plt.figure(figsize=(8,6))
plt.scatter(x=y_test,y=pred_random,color='r')
plt.plot(y_test,y_test,color='b')
plt.xlabel('Actual Sales Price',fontsize=14)
plt.ylabel('Predicted Sales Price',fontsize=14)
plt.title('GradientBoostingRegressor',fontsize=18)
plt.show()

```

Most of data points are interacting with predicted line

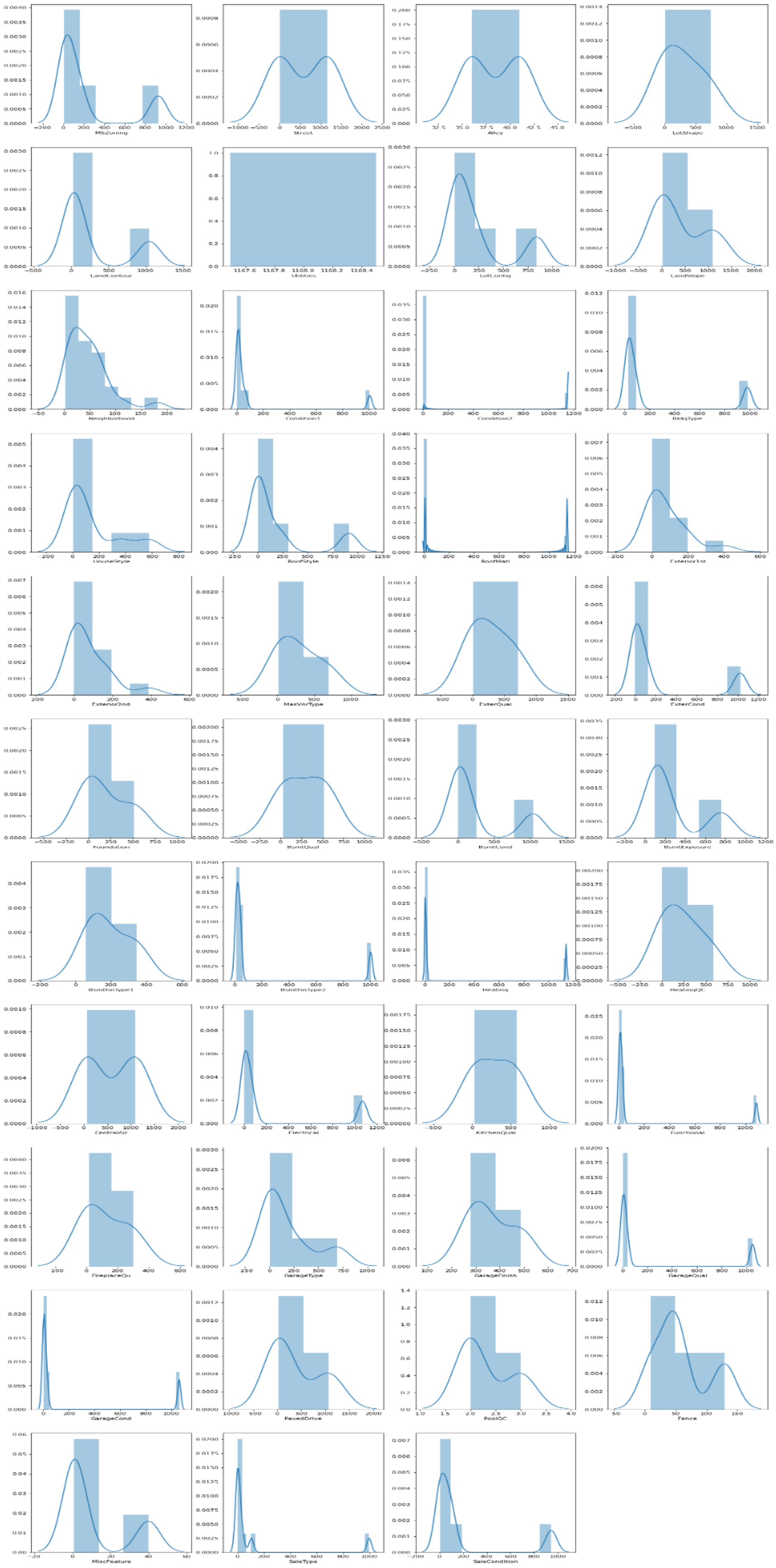
- Visualizations
  1. a) Univariant Analysis
  2. b) Bi Variant Analysis
  3. c) Multi Variant Analysis

Please find the graphs below:

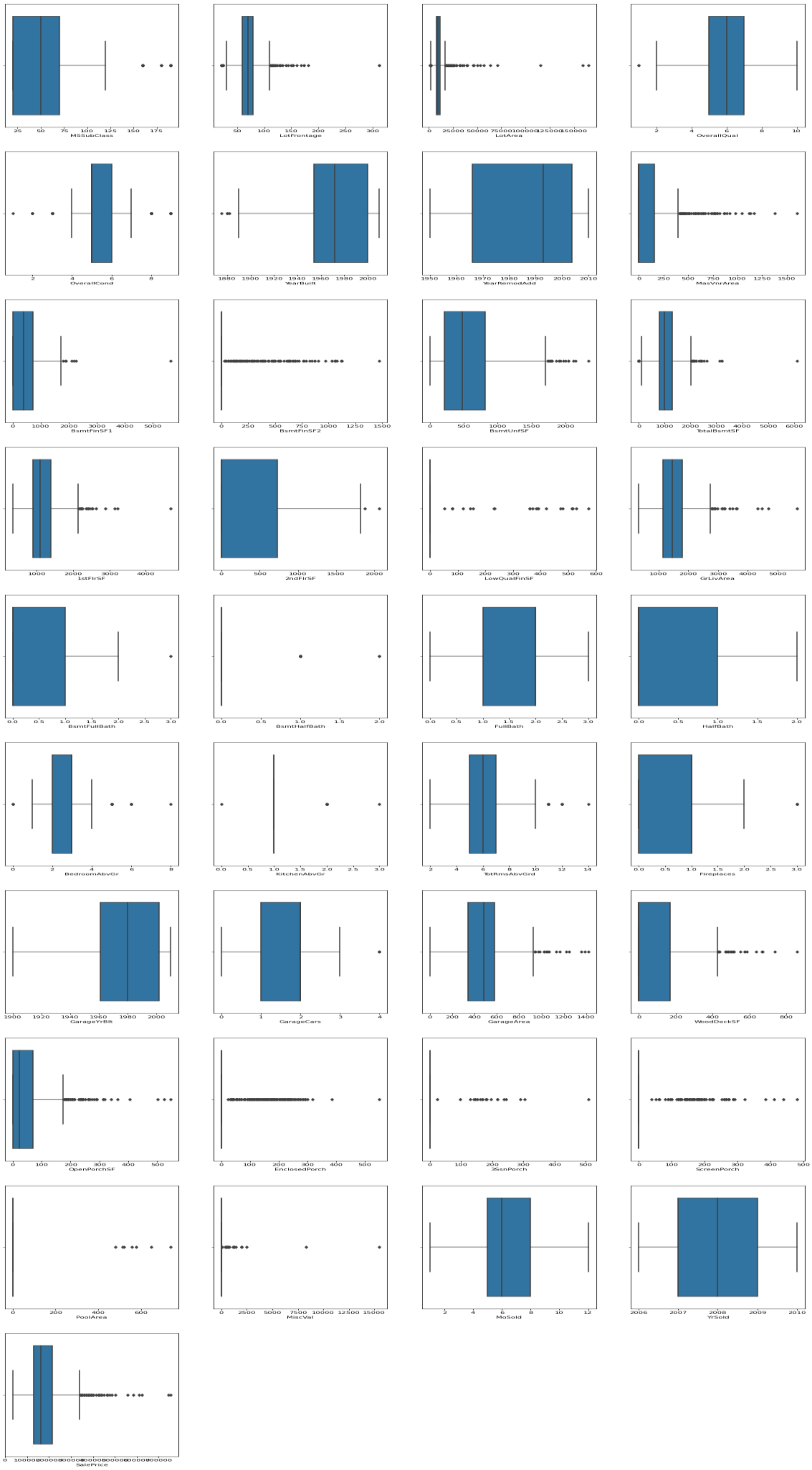
i) Bar Graphs



ii) Histogram

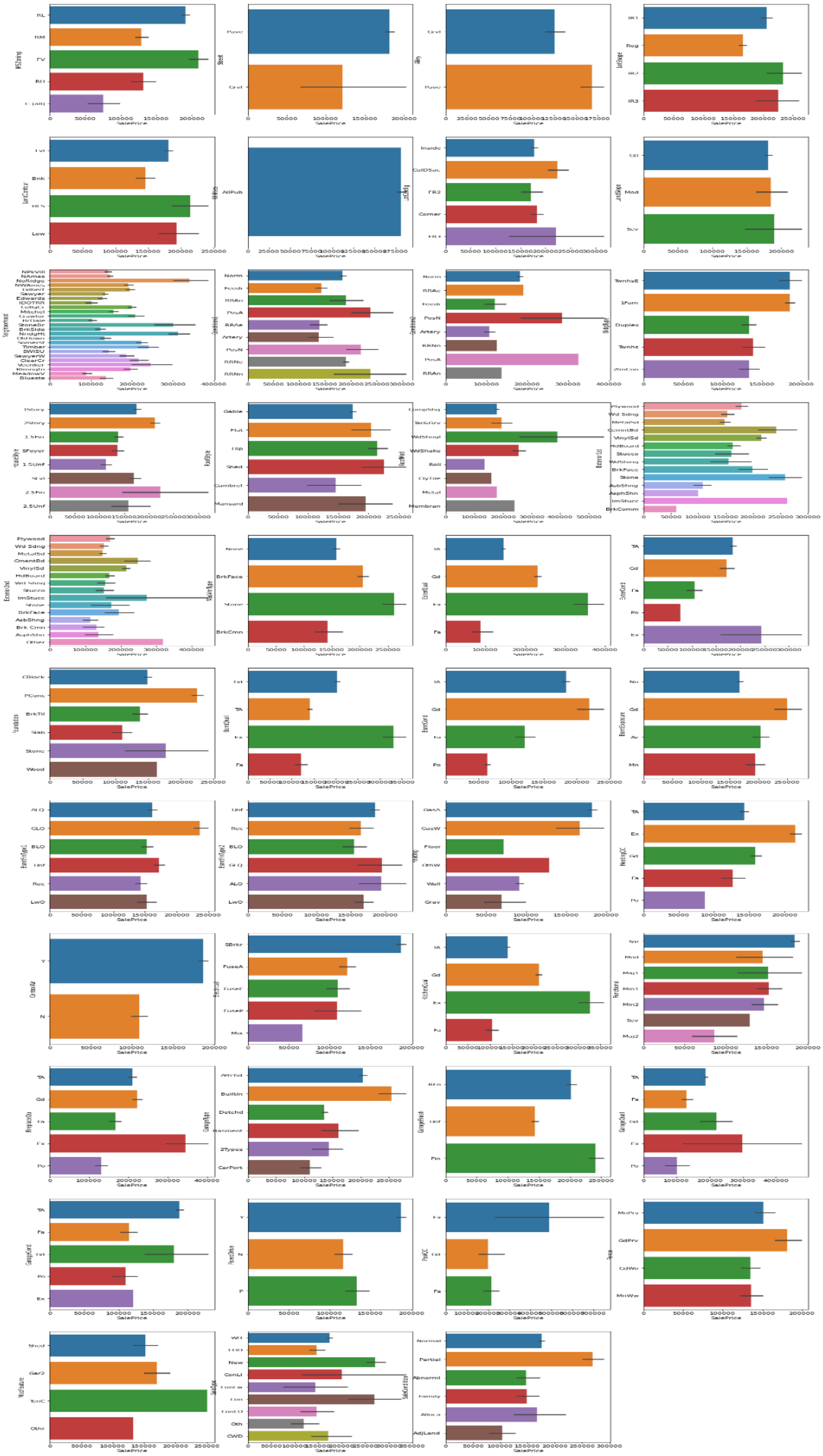


iii) Box plot



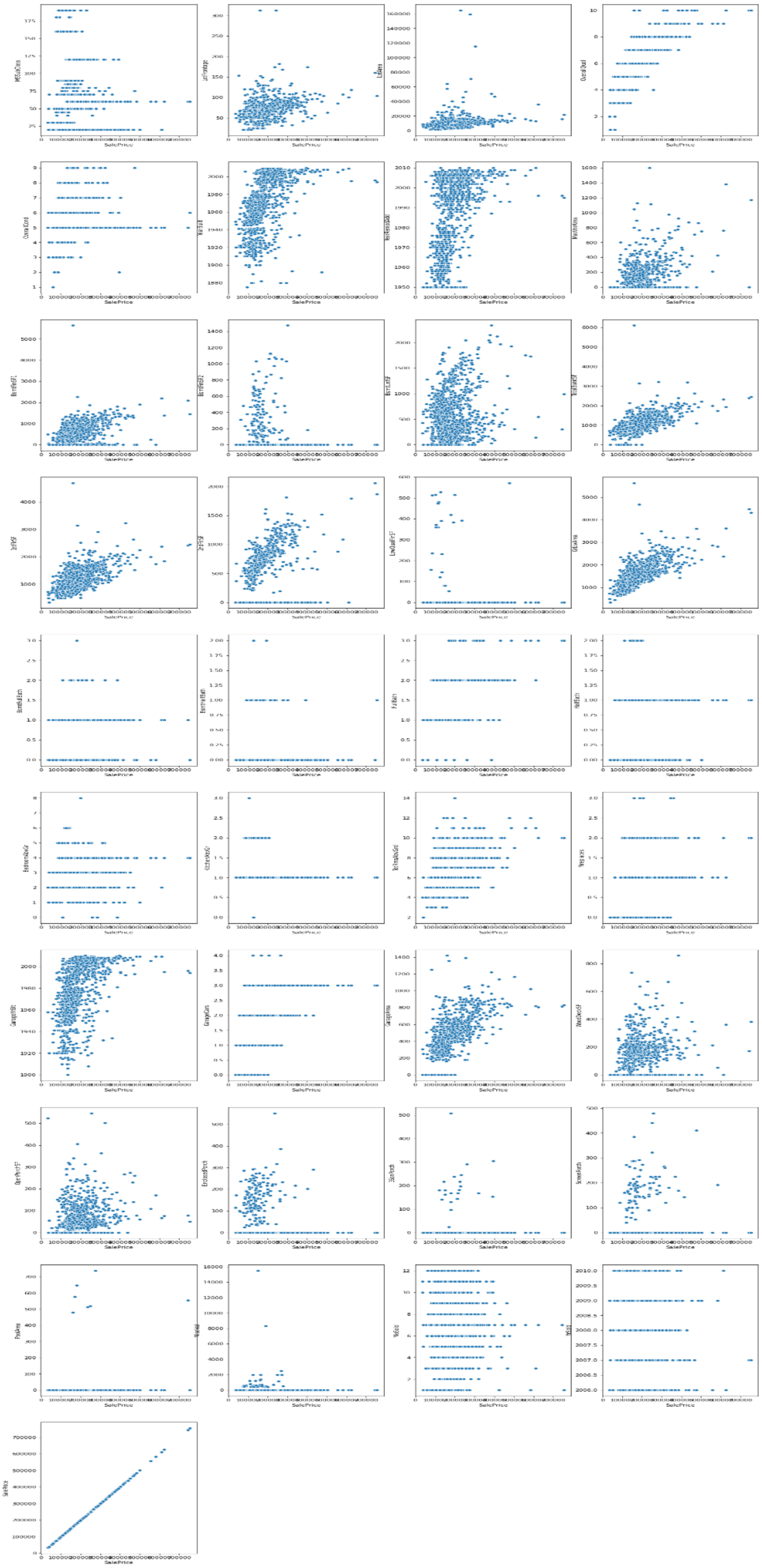
25 50 75 100 125 150 175
MSSubClass
50 100 150 200 250 300
LotFrontage
0 25000 50000 75000 100000 125000 150000
LotArea
2 4 6 8 10
OverallQual
2 4 6 8
OverallCond
1880 1900 1920 1940 1960 1980 2000
YearBuilt
1950 1960 1970 1980 1990 2000 2010
YearRemodAdd
0 250 500 750 1000 1250 1500
MasVnrArea
0 1000 2000 3000 4000 5000
BsmtFinSF1
0 250 500 750 1000 1250 1500
BsmtFinSF2
0 500 1000 1500 2000
BsmtUnfSF
0 1000 2000 3000 4000 5000 6000
TotalBsmtSF
1000 2000 3000 4000
1stFlrSF
0 500 1000 1500 2000
2ndFlrSF
0 100 200 300 400 500 600
LowQualFinSF
1000 2000 3000 4000 5000
GrLivArea
0.0 0.5 1.0 1.5 2.0 2.5 3.0
BsmtFullBath
0.0 0.5 1.0 1.5 2.0
BsmtHalfBath
0.0 0.5 1.0 1.5 2.0 2.5 3.0
FullBath
0.0 0.5 1.0 1.5 2.0
HalfBath
0 2 4 6 8
BedroomAbvGr
0.0 0.5 1.0 1.5 2.0 2.5 3.0
KitchenAbvGr
2 4 6 8 10 12 14
TotRmsAbvGrd
0.0 0.5 1.0 1.5 2.0 2.5 3.0
Fireplaces
1900 1920 1940 1960 1980 2000
GarageYrBlt
0 1 2 3 4
GarageCars
0 200 400 600 800 1000 1200 1400
GarageArea
0 200 400 600 800
WoodDeckSF
0 100 200 300 400 500
OpenPorchSF
0 100 200 300 400 500
EnclosedPorch
0 100 200 300 400 500
3SsnPorch
0 100 200 300 400 500
ScreenPorch
0 200 400 600
PoolArea
0 2500 5000 7500 10000 12500 15000
MiscVal
2 4 6 8 10 12
MoSold
2006 2007 2008 2009 2010
YrSold
0 100000 200000 300000 400000 500000 600000 700000
SalePrice

iv) Sub Plot

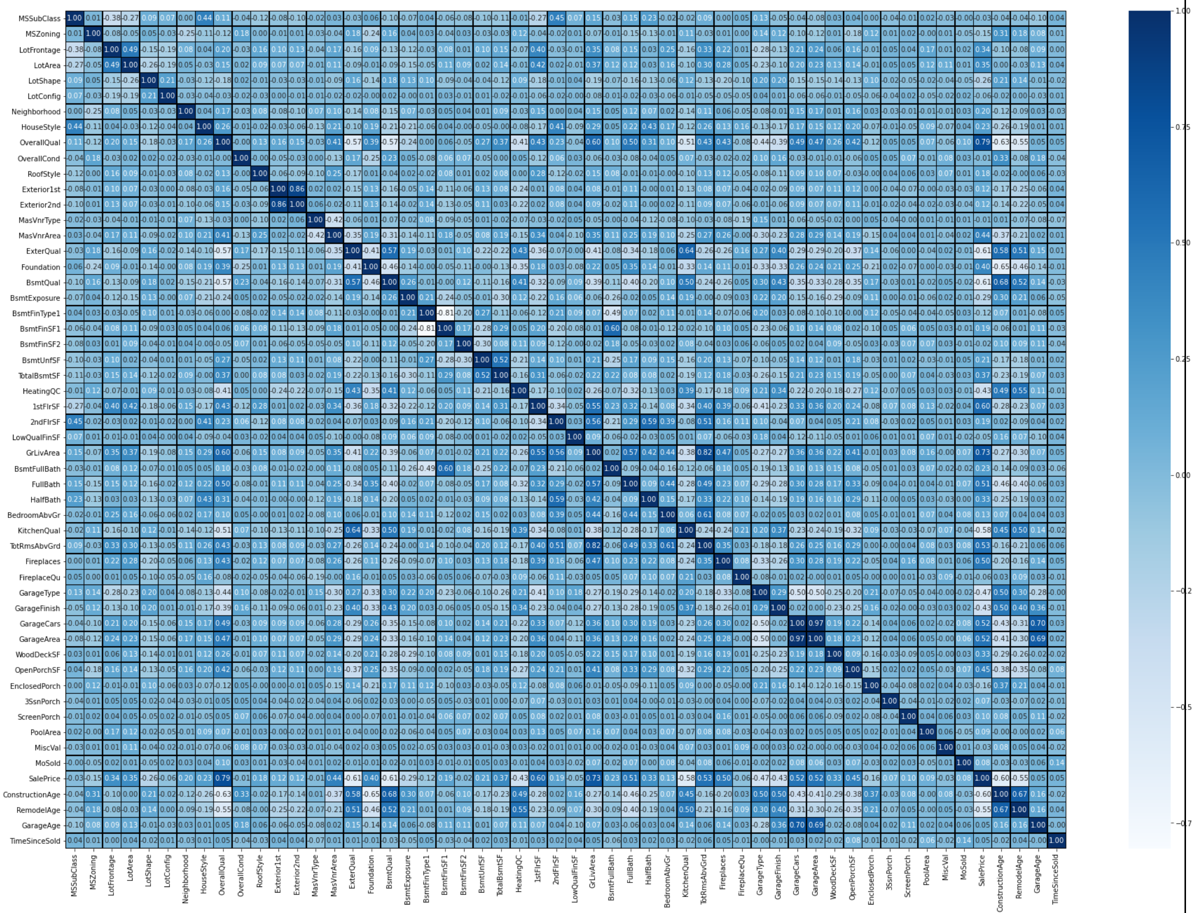


SalePrice
MSZoning
RL
RM
FV
RH
C (all)
Street
Pave
Grvl
Alley
LotShape
Reg
IR1
IR2
IR3
LandContour
Lvl
Bnk
HLS
Low
Utilities
AllPub
LotConfig
Inside
CulDSac
FR2
Corner
FR3
LandSlope
Gtl
Mod
Sev
Neighborhood
Condition1
Condition2
Norm
Feedr
PosN
Artery
RRAe
RRNn
RRAn
PosA
RRNe
BldgType
1Fam
2fmCon
Duplex
TwnhsE
Twnhs
HouseStyle
2Story
1Story
1.5Fin
1.5Unf
SFoyer
SLvl
2.5Unf
2.5Fin
RoofStyle
Gable
Hip
Gambrel
Mansard
Flat
Shed
RoofMatl
CompShg
WdShngl
Metal
WdShake
Membran
Tar&Grv
Roll
ClyTile
Exterior1st
Exterior2nd
VinylSd
MetalSd
Wd Sdng
HdBoard
BrkFace
WdShing
CemntBd
Plywood
AsbShng
Stucco
BrkComm
AsphShn
Stone
ImStucc
CBlock
Other
MasVnrType
BrkFace
None
Stone
BrkCmn
ExterQual
ExterCond
Gd
TA
Ex
Fa
Po
Foundation
PConc
CBlock
BrkTil
Wood
Slab
Stone
BsmtQual
BsmtCond
BsmtExposure
No
Gd
Mn
Av
BsmtFinType1
BsmtFinType2
GLQ
ALQ
Unf
Rec
BLQ
LwQ
Heating
GasA
GasW
Grav
Wall
OthW
Floor
HeatingQC
CentralAir
Y
N
Electrical
SBrkr
FuseF
FuseA
FuseP
Mix
KitchenQual
Functional
Typ
Min1
Maj1
Min2
Mod
Maj2
Sev
FireplaceQu
GarageType
Attchd
Detchd
BuiltIn
CarPort
Basment
2Types
GarageFinish
RFn
Unf
Fin
GarageQual
GarageCond
PavedDrive
P
PoolQC
Fence
MnPrv
GdWo
GdPrv
MnWw
MiscFeature
Shed
Gar2
Othr
TenC
SaleType
WD
New
COD
ConLD
ConLI
CWD
ConLw
Con
Oth
SaleCondition
Normal
Abnorml
Partial
AdjLand
Alloca
Family

v) Scatter Plot:



vi) Correlation – Heatmap



- Interpretation of the Results

a. Here we see correlation of all features based on saleprice or target.

b. GarageCars and GarageArea both are high correlation with each others (0.88)

c. TotRmsAbvGrd and GrLivArea both are high correlation with each others(0.83)

d. TotalBsmtSF and 1stFlrSF both are high correlation with each others(0.82)

e. Gradient Regressor is having the Accuracy of 86% which is quite a good score in predicting the model when compare with other model.

f. The suburb of South Melbourne has the least negative coefficient and a house bought there would cost more.

g. The suburb of North Melbourne has the most negative coefficient and a house bought there would cost less.

h. The predictive model i.e. Gradient Regressor accounts for 86% of the variation in prices, which indicates a good accuracy in the predictions.

i. The model's accuracy and reliability can be improved by excluding the outliers from the model data in future work.

CONCLUSION

- Key Findings and Conclusions of the Study

  a. The study shows a comparison between the regression algorithms and artificial neural network when predicting house prices.

  b. The local data gave a worse outcome when the same pre-processing strategy was implemented due to it being in a different shape compared with the public data in terms of the number of features and the correlation strength.

  c. Hence, the local data needs more features to be added preferably with a strong correlation with the house price.

  d. However, Gradient Regression got the best R2 score overall. The final results of this study showed that Gradient Regression better prediction compared to other used algorithms.

- Learning Outcomes of the Study in respect of Data Science

  a. Here we see correlation of all features based on saleprice or target.

  b. GarageCars and GarageArea both are high correlation with each others (0.88)

  c. TotRmsAbvGrd and GrLivArea both are high correlation with each others(0.83)

  d. TotalBsmtSF and 1stFlrSF both are high correlation with each others(0.82)

  e. Gradient Regressor is having the Accuracy of 86% which is quite a good score in predicting the model when compare with other model.

  f. The suburb of South Melbourne has the least negative coefficient and a house bought there would cost more.

  g. The suburb of North Melbourne has the most negative coefficient and a house bought there would cost less.

  h. The predictive model i.e. Gradient Regressor accounts for 86% of the variation in prices, which indicates a good accuracy in the predictions.

  i. The model's accuracy and reliability can be improved by excluding the outliers from the model data in future work.

- Limitations of this work and Scope for Future Work

  a. A list of features, that matches the public dataset's features, that is desired to be available when the data is sent.

  b. There is no guarantee that the data will be available in time nor contains the exact requested list of features.

  c. Thus, there might be a risk that the access will be denied or delayed. If so, the study will be accomplished based only on the public dataset.

  d. Moreover, this study will not cover all regression algorithms; instead, it is focused on the chosen algorithm, starting from the basic regression techniques to the advanced ones.

  e. Likewise, the artificial neural network that has many techniques and a wide area and several training methods that do not fit in this study.