FLIP ROBO

HOUSING PRICE PREDICTION PROJECT

Submitted by:

CHETHANA M

ACKNOWLEDGMENT

I express.my sincere gratitude to **Flip Robo Technologies** for giving me this opportunity to carry out the project work.

A special thanks to my mentor **Mohd Kashif** for guiding me in completing this project and being available to resolve my doubts whenever I raise any tickets.

I also would love to take this moment to thank **DataTrained** for giving me all the knowledge about how to build an effective machine learning model and providing this opportunity to work as intern in Flip Robo Technologies.

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.

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 data is provided in the CSV file below.

The company is looking at prospective properties to buy houses to enter the market. Our goal is 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?

Our main aim today is to make a model which can give us a good prediction on the price of the house based on other variables. We are going to use Linear Regression, Support vector regressor, Decision Tree Regressor, K Neighbors Regressor, Ridge and Lasso, Random forest classifier to build the different models for this dataset and see which model gives us a good accuracy.

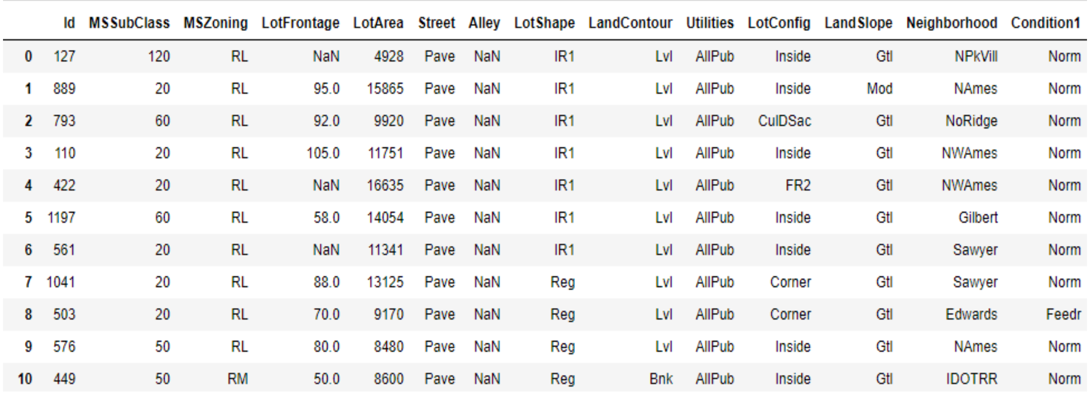
The Motivation behind it is I just want to know about the house prices in Australia as well as which are the instances that gives major impact on deciding price of a house.

Analytical Problem Framing

In this project, house prices will be predicted using given explanatory variables that cover many aspects of residential houses. The goal of this project is to create a regression model that is able to accurately estimate the price of the house under the given features.

In this dataset is made for predicting the House Price of Australia. Here I just show all of the feature for each house separately. Such as Width of the street, Area covered, Shape of the house, Material used to build the house, Rating given for the house, Condition of the house, No of Room, Kitchen type, No of bathrooms, Pool, No of stories and so on. We'll about the variables in the upcoming part.

Data contains 1460 entries each having 81 variables.
Data contains Null values. These null values should be treated using the domain knowledge and own understanding of the dataset.
Extensive EDA has to be performed to gain relationships of important variable and price.
Data contains numerical as well as categorical variable which needs to handled accordingly.
Machine Learning models is needed to be built, apply regularization and determine the optimal values of Hyper Parameters.
You need to find important features which affect the price positively or negatively.
Two datasets are being provided to you (test.csv, train.csv). train.csv dataset is used to train the model and predict on test.csv file.



	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1
0	127	120	RL	NaN	4928	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	NPkVill	Norm
1	889	20	RL	95.0	15865	Pave	NaN	IR1	Lvl	AllPub	Inside	Mod	NAmes	Norm
2	793	60	RL	92.0	9920	Pave	NaN	IR1	Lvl	AllPub	CulDSac	Gtl	NoRidge	Norm
3	110	20	RL	105.0	11751	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	NWAmes	Norm
4	422	20	RL	NaN	16635	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl	NWAmes	Norm
5	1197	60	RL	58.0	14054	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Gilbert	Norm
6	561	20	RL	NaN	11341	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Sawyer	Norm
7	1041	20	RL	88.0	13125	Pave	NaN	Reg	Lvl	AllPub	Corner	Gtl	Sawyer	Norm
8	503	20	RL	70.0	9170	Pave	NaN	Reg	Lvl	AllPub	Corner	Gtl	Edwards	Feedr
9	576	50	RL	80.0	8480	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	NAmes	Norm
10	449	50	RM	50.0	8600	Pave	NaN	Reg	Bnk	AllPub	Inside	Gtl	IDOTRR	Norm

Data Pre-processing Done

There were presence of missing data, outliers, columns that doesn't create any impact on output variable. Missing values in few columns were filled with NA based on assumptions made in EDA and rest were treated with simple imputer. The outliers were removed from continuous data using z-score method. Multicollinearity was checked and few columns were dropped which were creating high multicollinearity in the dataset.

Data Inputs- Logic- Output

As discussed before there are 81 variables in the data set, 1 output variable and 80 input variables

INPUT VARIABLES : Id, MSSubClass, MSZoning, LotFrontage, LotArea, Street, Alley, LotShape, LandContour, Utilities, LotConfig, LandSlope, Neighborhood, Condition1, Condition2, BldgType, HouseStyle, OverallQual, OverallCond, YearBuilt, YearRemodAdd, RoofStyle, RoofMatl, Exterior1st, Exterior2nd, MasVnrType, MasVnrArea, ExterQual, ExterCond, Foundation, BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinSF1, BsmtFinType2, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF, Heating, HeatingQC, CentralAir, Electrical, 1stFlrSF, 2ndFlrSF, LowQualFinSF, GrLivArea, BsmtFullBath, BsmtHalfBath, FullBath, HalfBath, BedroomAbvGr, KitchenAbvGr, KitchenQual, TotRmsAbvGrd, Functional, Fireplaces, FireplaceQu, GarageType, GarageYrBlt, GarageFinish, GarageCars, GarageArea, GarageQual, GarageCond, PavedDrive, WoodDeckSF, OpenPorchSF, EnclosedPorch, 3SsnPorch, ScreenPorch, PoolArea, PoolQC, Fence, MiscFeature, MiscVal, MoSold, YrSold, SaleType, SaleCondition

OUTPUT VARIABLE : SalePrice

About the Algorithms used

The major aim in this project is to predict the house prices based on the features using some of the regression techniques and algorithms.

1. Linear Regression
2. Support vector Regressor
3. Decision Tree Regressor
4. KNeighbors Regressor
5. Ridge and Lasso Regressors
6. Random Forest Classifier

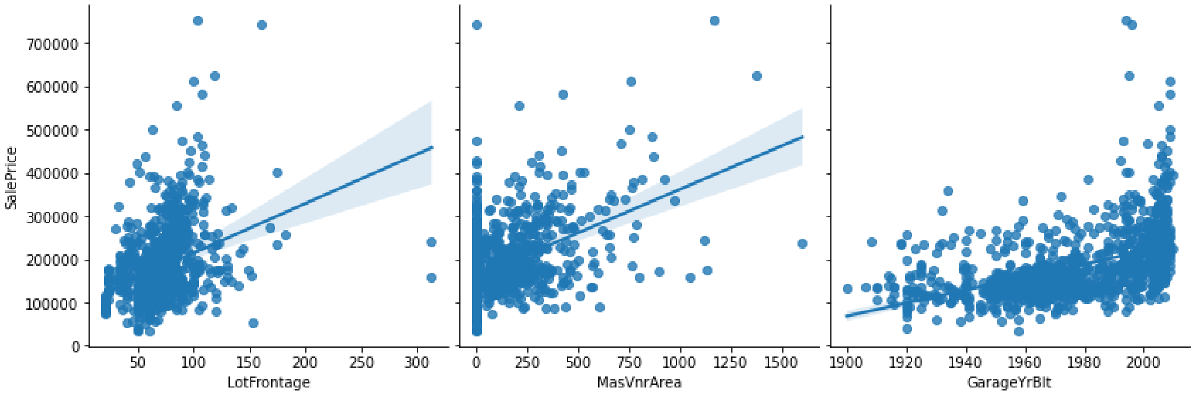
Machine Learning Packages are used for in this Project

The dataset was available in the zip folder which had train and test datasets separately in the form of csv files, these csv files were uploaded to Jupyter and read using pandas library. Once the dataset is read DataFrame is created and further EDA process is done on the dataset using different functions available in pandas.

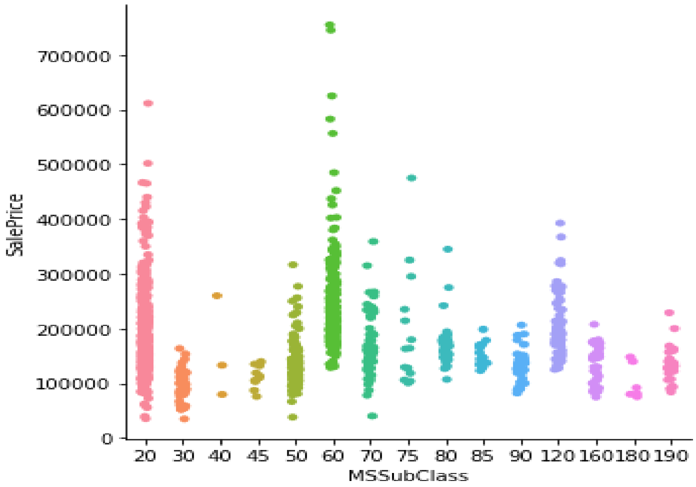
The Seaborn and Matplotlib are used to plot the different graphs and understand the relationship between each variable and output variable.

Model/s Development and Evaluation

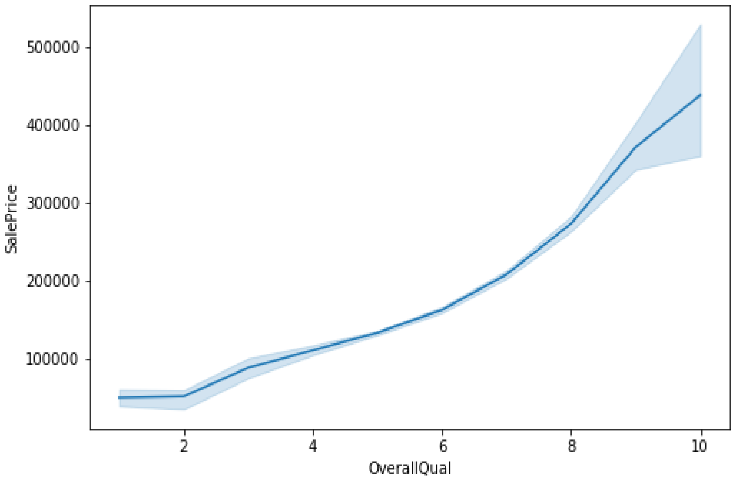
To see about the relation between Output variable and few input variables.



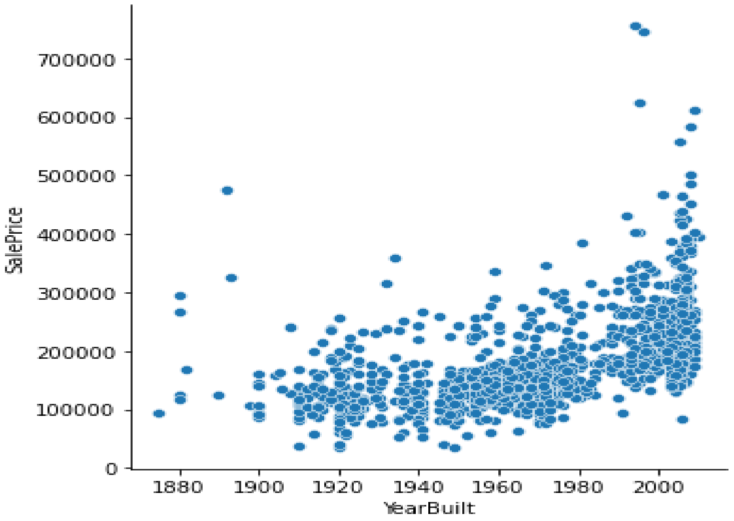
- In the plot LotFrontage Vs SalePrice we can observe that the linear feet of street which is connected to property is mostly ranging from 10 to 120 and for these ranges the sale price is ranging from 2000 to 40000.
- The Masonry veneer area is ranging from 0 to 1000 in most of the cases and price for this range is from 5000 to 40000.
- In the plot GarageYrBlt Vs SalePrice we can observe that as the year when the garage was built goes increasing price of the property increases.



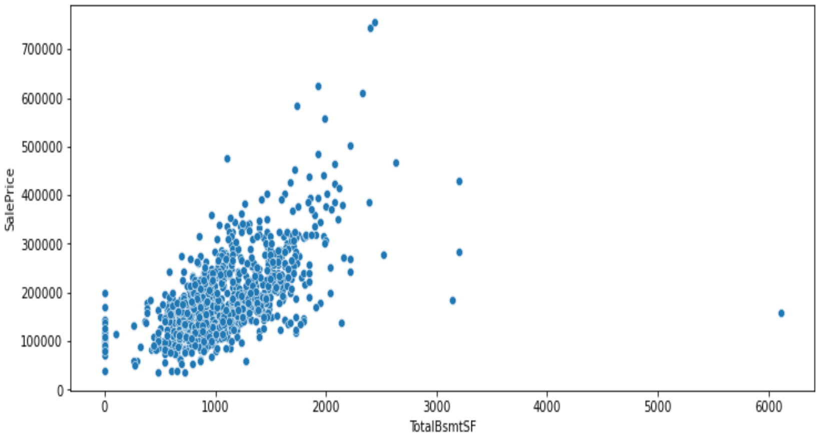
From the about category plot we can observe that the sale price is highest for those properties which are 2-STORY 1946 & NEWER



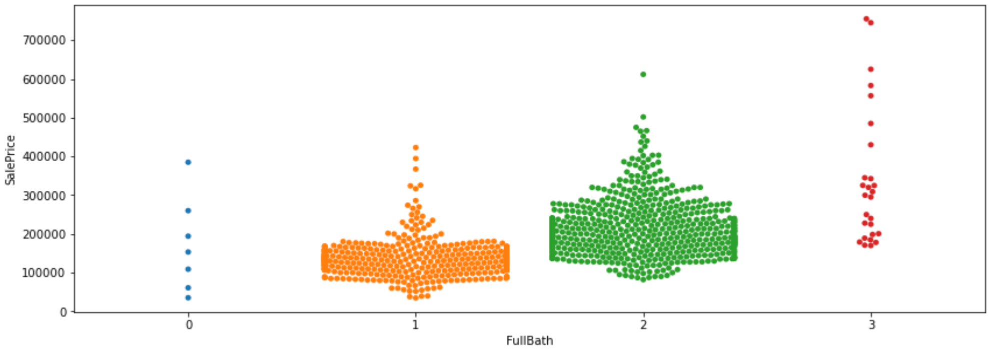
From the above plot it is clear that Price increases as the overall ratings of the property increase



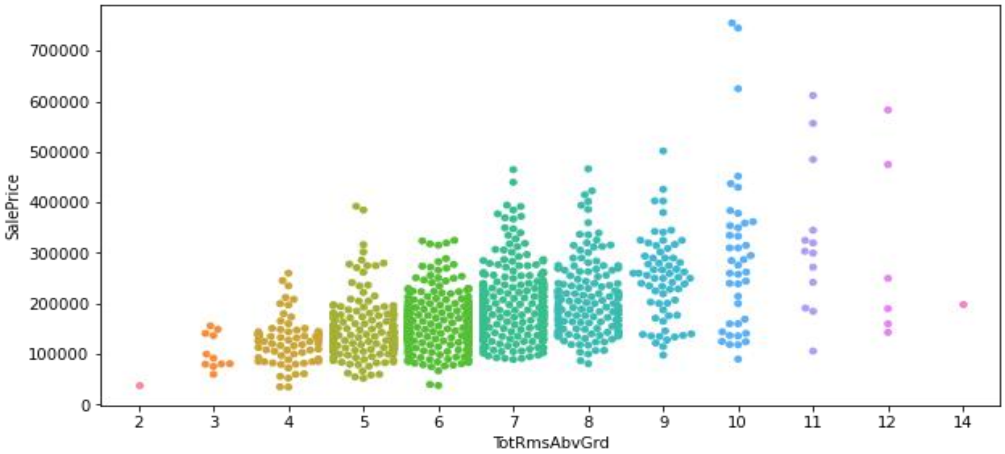
From the above plot we can conclude that the Price is high for those properties bulit in latest years than the older properties



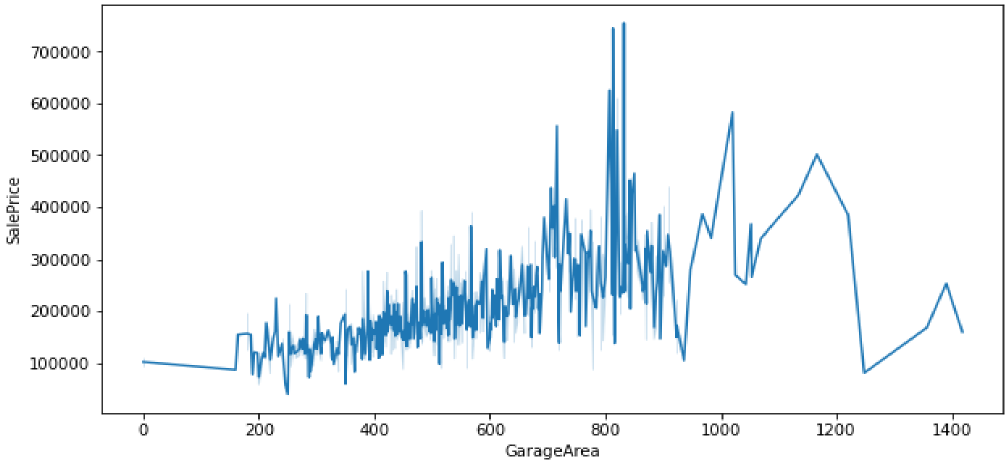
From the above plot we can observe that the Price increases with the increase of total basement area increase and from 2500 the price slightly reduces for few properties



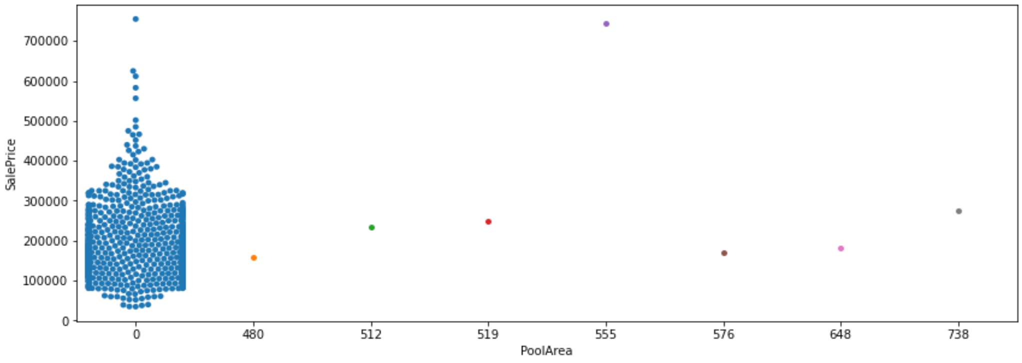
From the above plot we can observe that most of the properties has 1 or 2 Full bathrooms above grade and as the number of bathrooms increase the Price of the property increases



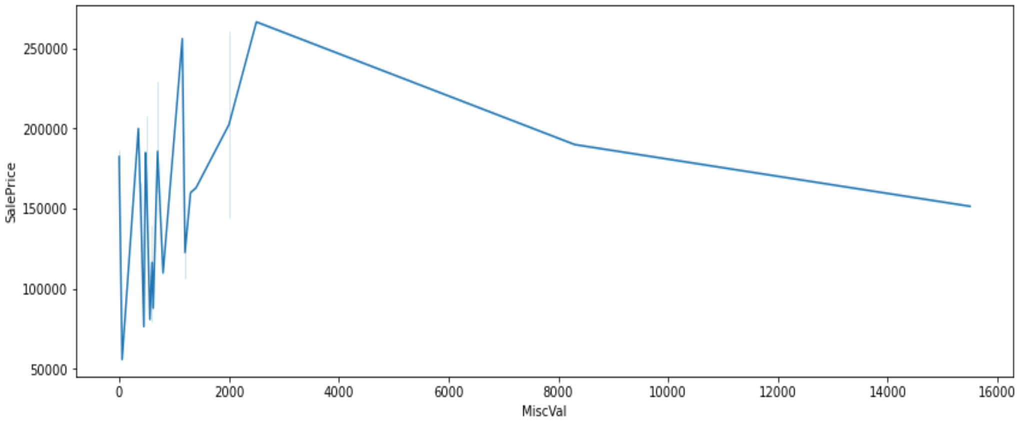
The price is highest for those properties which has total rooms above grade as 10 and it is least for the property which has 2 rooms.



Price of the property increases with the increase in area of garage



We can observe that maximum properties have no pool area and for those countable number of properties which has pool price ranges from 150000 to 300000



From 0 to 2500 we can see that the price increases with the increase in the $Value of miscellaneous feature and it goes reducing gradually from 250

Few Key observations from the dataset :

Key Observations1 :

- Mean > median (50th percentile) in the columns MSSubClass, LotArea, MasVnrArea, BsmtFinType1, BsmtFinSF1, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF, Heating, HeatingQC, 1stFlrSF, 2ndFlrSF, LowQualFinSF, GrLivArea, OpenPorchSF, EnclosedPorch, 3SsnPorch, ScreenPorch, PoolArea, MiscVal, SalePrice have left skewness in the data
- We can observe that there is a huge gap between 75th percentile and max in the columns Id, MSSubClass, MSZoning, LotFrontage, LotArea, LandSlope, Neighborhood, Condition1, Condition2, BldgType, MasVnrArea, BsmtFinSF1, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF, 1stFlrSF, 2ndFlrSF, LowQualFinSF, GrLivArea, TotRmsAbvGrd, GarageArea, WoodDeckSF, EnclosedPorch, 3SsnPorch, ScreenPorch, PoolArea, MiscVal, SalePrice we can tell that outliers are present in these columns
- In almost all the columns we can see that there is a high gap between mean and std hence the data is highly spread.

Key Observations2 :

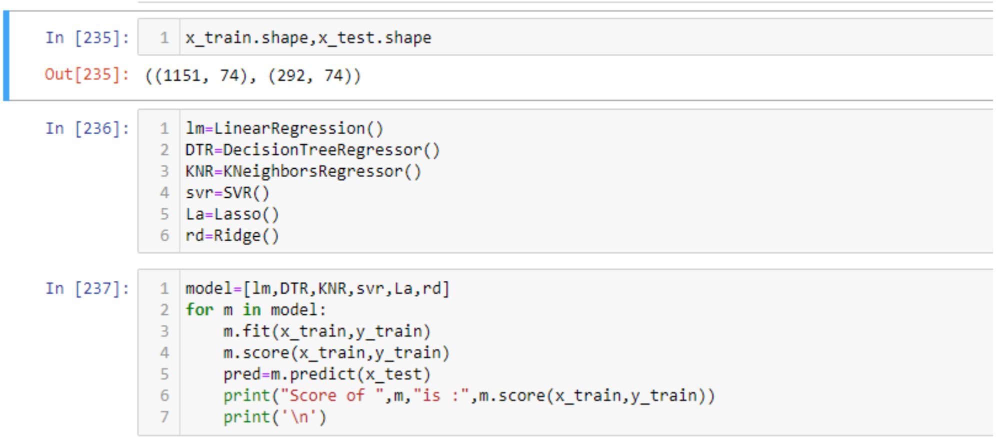
- From the heat map of correlation we can observe that SalePrice is highly correlated with variables like OverallQual, YearBuilt, TotalBsmtSF, 1stFlrSF, GrLivArea, FullBath, TotRmsAbvGrd, Fireplaces, GarageYrBlt, GarageArea
- SalePrice have negative correlation with GarageFinish, GarageType, KitchenQual, HeatingQC, BsmtQual, ExterQual

As we know that output data is continuous data we can use the algorithms

- Linear Regression
- Support vector Regressor
- Decision Tree Regressor
- KNeighbors Regressor
- Ridge and Lasso Regressors
- Random Forest Classifier

To build the model.

To build the model for above mentioned algorithms we have to follow the following steps.



```
In [235]: x_train.shape,x_test.shape
Out[235]: ((1151, 74), (292, 74))

In [236]: lm=LinearRegression()
          DTR=DecisionTreeRegressor()
          KNR=KNeighborsRegressor()
          svr=SVR()
          La=Lasso()
          rd=Ridge()

In [237]: model=[lm,DTR,KNR,svr,La,rd]
          for m in model:
              m.fit(x_train,y_train)
              m.score(x_train,y_train)
              pred=m.predict(x_test)
              print("Score of ",m,"is :",m.score(x_train,y_train))
              print('\n')
```

Output of these models were as :

Score of LinearRegression() is : 0.8652413853354501
Score of DecisionTreeRegressor() is : 1.0
Score of KNeighborsRegressor() is : 0.731103114938408
Score of SVR() is : -0.04677351659363316
Score of Lasso() is : 0.8652413443715304
Score of Ridge() is : 0.8652406950756619

We can see the accuracy of the model as

- LinearRegression() is 86.52%
- DecisionTreeRegressor() is 100%
- Lasso() is 86.52%
- Ridge() is 86.52%

and consider these models to work well

Now we can do parameter tuning for these above mentioned models and choose the best m odel

Parameter tuning for Decision Tree Regressor :

Best Score : 0.6856317815754264
Best Param : 'criterion': 'absolute_error'

Parameter tuning for Random Forest Classifier :

Best Score : 0.025191040843214756
Best Param : 'criterion': 'log_loss'

Parameter tuning for Lasso :

Best Score : 0.7939513771171223
Best Param : 'alpha': 10

Parameter tuning for Ridge :

Best Score : 0.7933266023539222
Best Param : 'alpha': 0.1

From the above observations we can consider Lasso and Ridge to be the best models. We can finalize either of the models now.

Saving the model

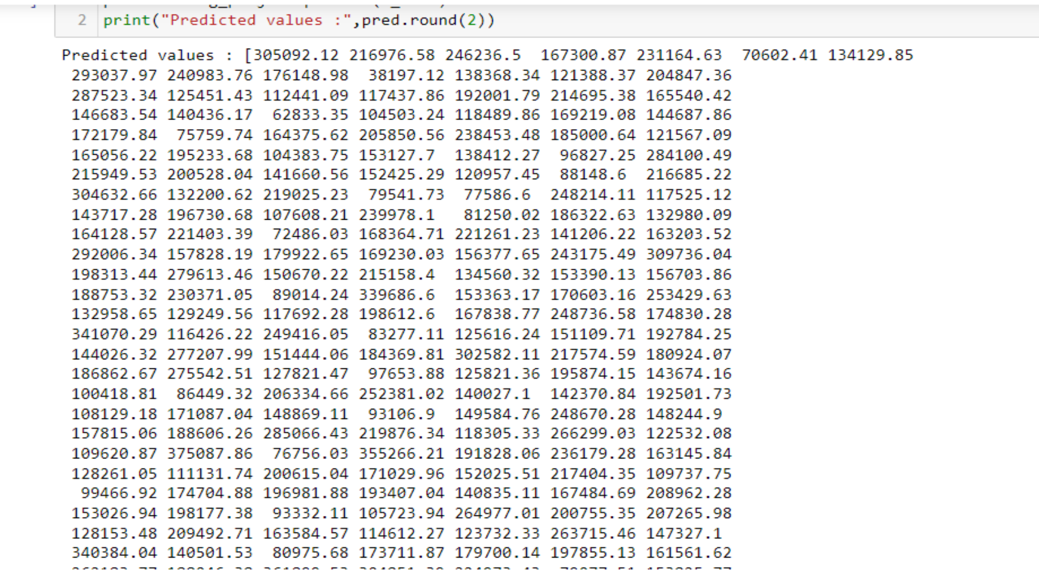
```
final_model=Lasso(alpha=10)

final_model.fit(x_train,y_train)

pred=final_model.predict(x_test)


print('Score:',final_model.score(x_train,y_train)*100)
```

```
Score: 86.5237320653474
```

```
import joblib

joblib.dump(final_model,'Housing_price_Prediction.obj')
```

Loading the saved model and predicting the values

```
Housing_project=joblib.load('Housing_price_Prediction.obj')


pred=Housing_project.predict(x_test)

print("Predicted values :",pred)
```



```python
print("Predicted values :",pred.round(2))
```

```
Predicted values : [305092.12 216976.58 246236.5  167300.87 231164.63  70602.41 134129.85
 293037.97 240983.76 176148.98  38197.12 138368.34 121388.37 204847.36
 287523.34 125451.43 112441.09 117437.86 192001.79 214695.38 165540.42
 146683.54 140436.17  62833.35 104503.24 118489.86 169219.08 144687.86
 172179.84  75759.74 164375.62 205850.56 238453.48 185000.64 121567.09
 165056.22 195233.68 104383.75 153127.7  138412.27  96827.25 284100.49
 215949.53 200528.04 141660.56 152425.29 120957.45  88148.6  216685.22
 304632.66 132200.62 219025.23  79541.73  77586.6  248214.11 117525.12
 143717.28 196730.68 107608.21 239978.1   81250.02 186322.63 132980.09
 164128.57 221403.39  72486.03 168364.71 221261.23 141206.22 163203.52
 292006.34 157828.19 179922.65 169230.03 156377.65 243175.49 309736.04
 198313.44 279613.46 150670.22 215158.4  134560.32 153390.13 156703.86
 188753.32 230371.05  89014.24 339686.6  153363.17 170603.16 253429.63
 132958.65 129249.56 117692.28 198612.6  167838.77 248736.58 174830.28
 341070.29 116426.22 249416.05  83277.11 125616.24 151109.71 192784.25
 144026.32 277207.99 151444.06 184369.81 302582.11 217574.59 180924.07
 186862.67 275542.51 127821.47  97653.88 125821.36 195874.15 143674.16
 100418.81  86449.32 206334.66 252381.02 140027.1  142370.84 192501.73
 108129.18 171087.04 148869.11  93106.9  149584.76 248670.28 148244.9
 157815.06 188606.26 285066.43 219876.34 118305.33 266299.03 122532.08
 109620.87 375087.86  76756.03 355266.21 191828.06 236179.28 163145.84
 128261.05 111131.74 200615.04 171029.96 152025.51 217404.35 109737.75
  99466.92 174704.88 196981.88 193407.04 140835.11 167484.69 208962.28
 153026.94 198177.38  93332.11 105723.94 264977.01 200755.35 207265.98
 128153.48 209492.71 163584.57 114612.27 123732.33 263715.46 147327.1
 340384.04 140501.53  80975.68 173711.87 179700.14 197855.13 161561.62
```

CONCLUSION

From the Exploratory Data Analysis, we could generate insight from the data. How each of the features relates to the target. Also, while training the model we could observe that Linear Regression, Lasso and Ridge were giving almost equal accuracy score of 86.5 %.

However while doing parameter tuning we could observe that Lasso was working better compared to other models with the final score of 86.5% and we have considered this model to fit best with alpha value 10.

From the EDA we could also observe that the Price of the property increases with few major factors like Over all quality of the property, year in which the house was built, Total basement surface, Number of bathrooms and rooms in the house, Garage area etc.

Few variables like Basement quality, External quality, Kitchen quality, garage type was negatively correlated with the price of the house.

While building the model we could observe that it is always good to build 4 to 5 models for the same train test data and compare the scores of the models then pick the best model instead of sticking onto single model.

To make the model more accurate I think it is good to use SMOTE and PCA techniques if applicable. Do scaling of the input variables so that models give better output.