FLIP ROBO

Housing Project

Submitted by:

Varagantham Gamya

ACKNOWLEDGMENT

- I would like to thank Flip Robo Technologies for providing me with the opportunity to work on this project from which I have learned a lot.
- Some of the reference sources are as follows:
- • Coding Ninjas
- • Medium.com
- • StackOverflow

INTRODUCTION

● Business Problem Framing

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.

● Conceptual Background of the Domain Problem

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.

● Review of Literature

Linear Regression is evaluated for their ability to predict house prices for the company which is trying to get into the market and the final model which gradient regressor gives the best accuracy.

● Motivation for the Problem Undertaken

This project was highly motivated project as it includes the real time problem for The real estate company which is using the machine learning model for the prediction of house prices based on different factors. The better the model the better of chances of profit for the business.

Analytical Problem Framing

● Mathematical/ Analytical Modeling of the Problem

```
train['SalePrice'].describe()

count      1460.000000
mean     180921.195890
std       79442.502883
min       34900.000000
25%      129975.000000
50%      163000.000000
75%      214000.000000
max      755000.000000
Name: SalePrice, dtype: float64
```

The above image shows the Statistics analysis of the variable Sale Price.

```
SalePrice        1.000000
Skewed_SP        0.948374
OverallQual      0.790982
GrLivArea        0.708624
GarageCars       0.640409
GarageArea       0.623431
TotalBsmtSF      0.613581
1stFlrSF         0.605852
FullBath         0.560664
TotRmsAbvGrd     0.533723
YearBuilt        0.522897
YearRemodAdd     0.507101
GarageYrBlt      0.486362
MasVnrArea       0.477493
Fireplaces       0.466929
BsmtFinSF1       0.386420
LotFrontage      0.351799
WoodDeckSF       0.324413
2ndFlrSF         0.319334
OpenPorchSF      0.315856
HalfBath         0.284108
LotArea          0.263843
BsmtFullBath     0.227122
BsmtUnfSF        0.214479
BedroomAbvGr     0.168213
ScreenPorch      0.111447
PoolArea         0.092404
MoSold           0.046432
3SsnPorch        0.044584
BsmtFinSF2      -0.011378
BsmtHalfBath    -0.016844
MiscVal         -0.021190
Id              -0.021917
LowQualFinSF    -0.025606
YrSold          -0.028923
OverallCond     -0.077856
MSSubClass      -0.084284
EnclosedPorch   -0.128578
KitchenAbvGr    -0.135907
Name: SalePrice, dtype: float64
```

The correlation of Sale price with all the other variables is given above.

● Data Sources and their formats

Data contains 1460 entries each having 81 variables.

Data contains Null values. You need to treat them using the domain knowledge and your own understanding.

Extensive EDA has to be performed to gain relationships of important variable and price.

Data contains numerical as well as categorical variable. You need to handle them accordingly.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
Id               1460 non-null int64
MSSubClass       1460 non-null int64
MSZoning         1460 non-null object
LotFrontage      1201 non-null float64
LotArea          1460 non-null int64
Street           1460 non-null object
Alley            91 non-null object
LotShape         1460 non-null object
LandContour      1460 non-null object
Utilities        1460 non-null object
LotConfig        1460 non-null object
LandSlope        1460 non-null object
Neighborhood     1460 non-null object
Condition1       1460 non-null object
Condition2       1460 non-null object
BldgType         1460 non-null object
HouseStyle       1460 non-null object
OverallQual      1460 non-null int64
OverallCond      1460 non-null int64
YearBuilt        1460 non-null int64
YearRemodAdd     1460 non-null int64
RoofStyle        1460 non-null object
RoofMatl         1460 non-null object
Exterior1st      1460 non-null object
Exterior2nd      1460 non-null object
MasVnrType       1452 non-null object
MasVnrArea       1452 non-null float64
ExterQual        1460 non-null object
ExterCond        1460 non-null object
Foundation       1460 non-null object
BsmtQual         1423 non-null object
BsmtCond         1423 non-null object
BsmtExposure     1422 non-null object
BsmtFinType1     1423 non-null object
BsmtFinSF1       1460 non-null int64
BsmtFinType2     1422 non-null object
BsmtFinSF2       1460 non-null int64
BsmtUnfSF        1460 non-null int64
TotalBsmtSF      1460 non-null int64
Heating          1460 non-null object
HeatingQC        1460 non-null object
CentralAir       1460 non-null object
Electrical       1459 non-null object
1stFlrSF         1460 non-null int64
2ndFlrSF         1460 non-null int64
LowQualFinSF     1460 non-null int64
GrLivArea        1460 non-null int64
BsmtFullBath     1460 non-null int64
BsmtHalfBath     1460 non-null int64
FullBath         1460 non-null int64
HalfBath         1460 non-null int64
BedroomAbvGr     1460 non-null int64
KitchenAbvGr     1460 non-null int64
KitchenQual      1460 non-null object
TotRmsAbvGrd     1460 non-null int64
Functional       1460 non-null object
Fireplaces       1460 non-null int64
FireplaceQu      770 non-null object
GarageType       1379 non-null object
GarageYrBlt      1379 non-null float64
GarageFinish     1379 non-null object
GarageCars       1460 non-null int64
GarageArea       1460 non-null int64
GarageQual       1379 non-null object
GarageCond       1379 non-null object
PavedDrive       1460 non-null object
WoodDeckSF       1460 non-null int64
OpenPorchSF      1460 non-null int64
EnclosedPorch    1460 non-null int64
3SsnPorch        1460 non-null int64
ScreenPorch      1460 non-null int64
PoolArea         1460 non-null int64
PoolQC           7 non-null object
Fence            281 non-null object
MiscFeature      54 non-null object
MiscVal          1460 non-null int64
MoSold           1460 non-null int64
YrSold           1460 non-null int64
SaleType         1460 non-null object
SaleCondition    1460 non-null object
SalePrice        1460 non-null int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
```

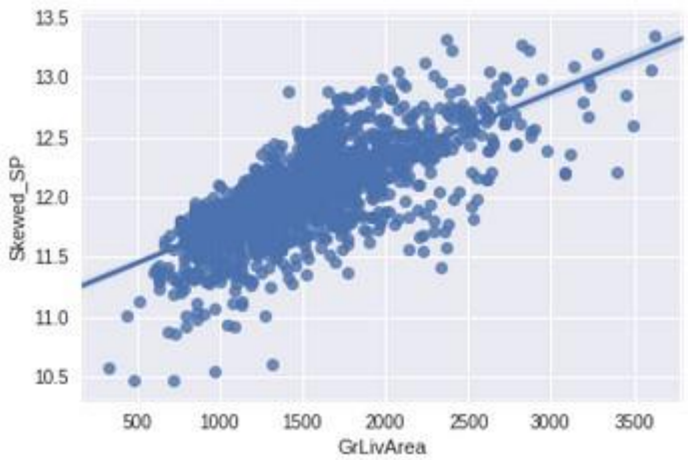
● Data Pre-processing Done

- We treated the skewness using Log transformation.
- We imputed the missing values.
- We encoded the categorical values using One hot encoding
- We trained the model on the train set
- We tested the model on test set

- Applied hyperparameters for improving the performance.

● Data Inputs- Logic- Output Relationships

SalePrice varies directly with the Overall quality



SalePrice increases as the GrLivArea increases. We will also get rid of the outliers which severely affect the prediction of the survival rate.

● Hardware and Software Requirements and Tools Used

Hardware: 8GB RAM, 64-bit, i7 processor.

Software: Excel, Jupyter Notebook, python 3.6., google colab

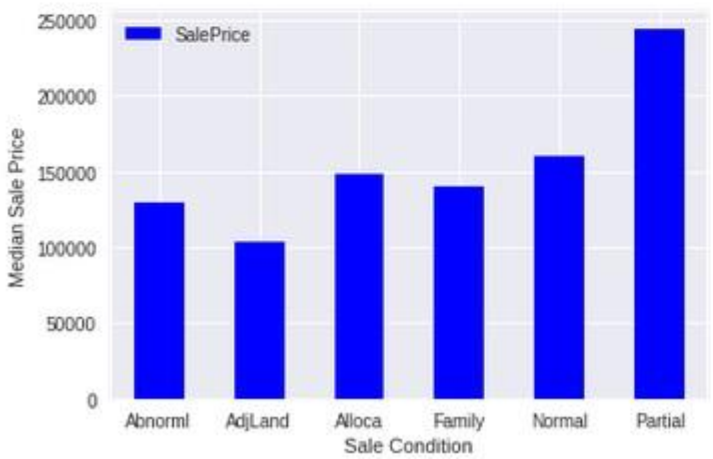
Libraries Used:-

```python
# Import libraries

# Pandas
import pandas as pd
from pandas import Series,DataFrame

# Numpy and Matplotlib
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
#sns.set_style('whitegrid')
%matplotlib inline

# Machine Learning
from sklearn import preprocessing
```

```python
from sklearn import linear_model
from sklearn import ensemble
```

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)



The Sale price is highly affected by sale condition.

- Testing of Identified Approaches (Algorithms)

Linear Regression
Random forest Regressor
Lasso Regressor
Ridge Regressor
Gradient Boost Regressor

- Run and Evaluate selected models

```
#lr =  ensemble.RandomForestRegressor(n_estimators = 100, oob_score = True, n_jobs = -1,random_state =50,max_feature
#lr = linear_model.LinearRegression()
lr = ensemble.GradientBoostingRegressor()
#lr = linear_model.TheilSenRegressor()
#lr = linear_model.RANSACRegressor(random_state=50)
```

```
model = lr.fit(X_train, y_train)
```

- Key Metrics for success in solving problem under consideration

```
print ("R^2 is: \n", model.score(X_test, y_test))
```

```
R^2 is:
 0.999768628635
```

```
from sklearn.metrics import mean_squared_error
print ('RMSE is: \n', mean_squared_error(y_test, predictions))
```

```
RMSE is:
 3.57545004789e-05
```

The R2 score and the RMSE is given above

- Interpretation of the Results

From the above visualization and matrices found that the Gradient boost regressor performed the best 99% R2 score, with least root mean square error which we were able to achieve from dataset provided.

CONCLUSION

- Key Findings and Conclusions of the Study

From the above visualisation and model building we analysed that Gradient boost regressor performed better when this type of dataset was given and based on the model performance it can be used to predict the house price of the house based on various factors.

Based on the final model the Real estate company can make decisions and there is a higher possibility that the decisions will be profitable.