FLIP ROBO

HOUSING: PRICE PREDICTION

Submitted by:

ANURAG KUMAR

ACKNOWLEDGMENT

This project needed data provided by Flip Robo Technology. Necessary technical help. Some websites such as stackoverflow.com, Datatrained.com, guru99.com helped a lot for doubt clearing. Mentor at Flip Robo guided me through all the speed brakers I faced during this project completion.

INTRODUCTION

• Business Problem Framing

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. So, model using Machine Learning in order to predict the actual value of the prospective properties and decide whether to invest in them or not. In real life business decision taken on the basis of all the factors which could affect the decision.

• Conceptual Background of the Domain Problem

As this problem needed some basic knowledge of real estate functioning such as what are the different variables which could affect the sales price of a house. Some variables knowledge like shape of land, type of building, size of lot required for building construction, basic materials used in finishing,

• Review of Literature

There are some need to through research of Australian property market. For the reference I went through article at https://www.claytonutz.com/ArticleDocuments/178/ICLG-Real-Estate-Australia-2009.pdf. Also, there are Australia property bubble in 2010, So article in microbusiness.com (House Affordability : Australia : Markets : 1981–2010, n.d.) helped a lot.

• Motivation for the Problem Undertaken

As the worldwide property market seen downslide including India due to many reasons among which Covid is one of the major reasons. So, it makes me curious what are other factors which affect the house sales price around the globe.

Analytical Problem Framing

- ### Mathematical/ Analytical Modelling of the Problem

  Some Mathematical and Statistical modelling required for the problem solving such as Correlation, Regression, Standard deviation, mean, pivot and quantile.

- ### Data Sources and their formats

  Data was provided in csv format by flip robo which contains two csv file one for training the model and another one for testing. Training data set have 80 columns of independent variable and one column of output variable. Whereas Test data set has all independent features.

- ### Data Pre-processing Done

  First hurdle in making ML model is missing values. For solving nan value problem, we use mean value of the items. Second problem was outlier which needed to be fixed for irrational decision making by model. So, we used standard deviation method for removing outliers. Logarithmic Transformation also required for normal curve of continuous data set.

- ### Data Inputs- Logic- Output Relationships

  Relationship between Input and output variables is best described by correlation matrix. Either they are positively or negatively correlated. Some features have very weak correlation with output variables.

- ### Hardware and Software Requirements and Tools Used

  Anaconda package used for this project. Most work done at jupyter notebook for model building. Different library such as Pandas, NumPy, Matplotlib, Seaborn, SciPy, sklearn are used.

Model/s Development and Evaluation

- ## Identification of possible problem-solving approaches (methods)

  Divided Numerical data into two categories Discrete and Continuous variable for analysis purpose. Used different graphs for EDA purpose. Used Some statistical tools such as Standard Deviation and Zscore for outlier removal. Used logarithmic normalization.

- ## Testing of Identified Approaches (Algorithms)

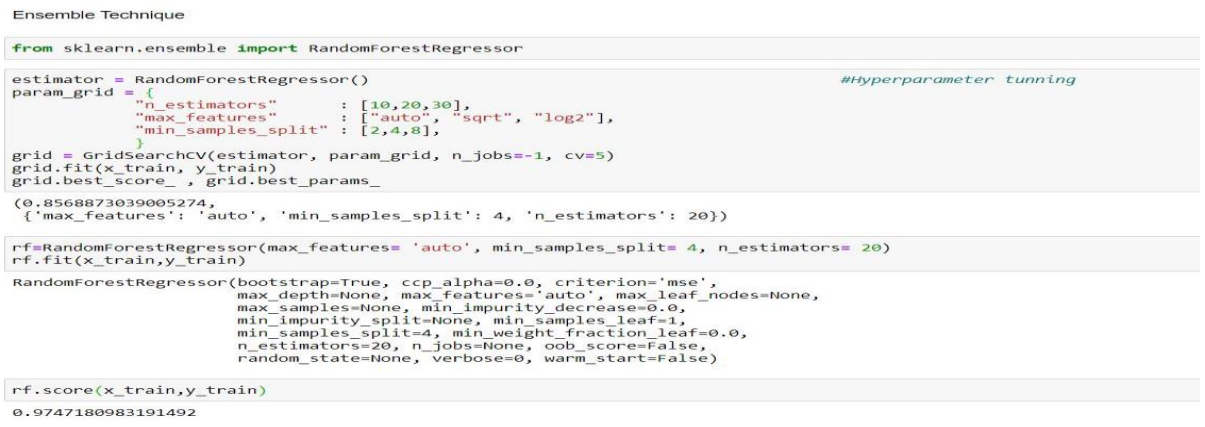
  a) Cleaning of test data set. b) Label Encoding c) Min Max Scaler

  d) Remove weak correlated variables. e) Training dataset which include dependent and Independent Variable in Lasso Regression

  f) After training the model, we give test dataset for output variable (Sales Price).

- ## Run and evaluate selected models.

  Random Forest Regression



Ensemble Technique

```
from sklearn.ensemble import RandomForestRegressor
```

```
estimator = RandomForestRegressor()                                    #Hyperparameter tunning
param_grid = {
            "n_estimators"        : [10,20,30],
            "max_features"        : ["auto", "sqrt", "log2"],
            "min_samples_split" : [2,4,8],
            }
grid = GridSearchCV(estimator, param_grid, n_jobs=-1, cv=5)
grid.fit(x_train, y_train)
grid.best_score_ , grid.best_params_
```

```
(0.8568873039005274,
 {'max_features': 'auto', 'min_samples_split': 4, 'n_estimators': 20})
```

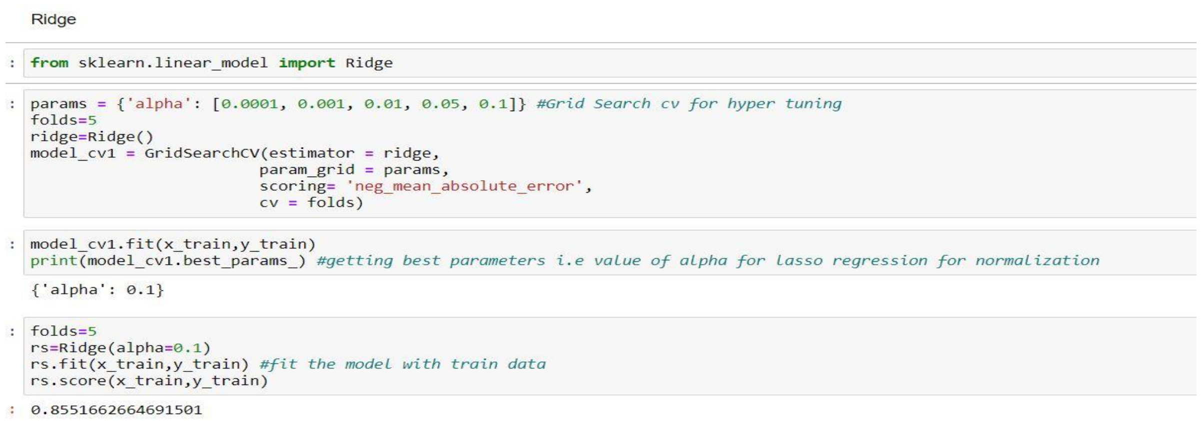
```
rf=RandomForestRegressor(max_features= 'auto', min_samples_split= 4, n_estimators= 20)
rf.fit(x_train,y_train)
```

```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=None, max_features='auto', max_leaf_nodes=None,
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=4, min_weight_fraction_leaf=0.0,
                      n_estimators=20, n_jobs=None, oob_score=False,
                      random_state=None, verbose=0, warm_start=False)
```

```
rf.score(x_train,y_train)
```

```
0.9747180983191492
```

Ridge Regression



Ridge

```
: from sklearn.linear_model import Ridge
```

```
: params = {'alpha': [0.0001, 0.001, 0.01, 0.05, 0.1]} #Grid Search cv for hyper tuning
  folds=5
  ridge=Ridge()
  model_cv1 = GridSearchCV(estimator = ridge,
                           param_grid = params,
                           scoring= 'neg_mean_absolute_error',
                           cv = folds)
```

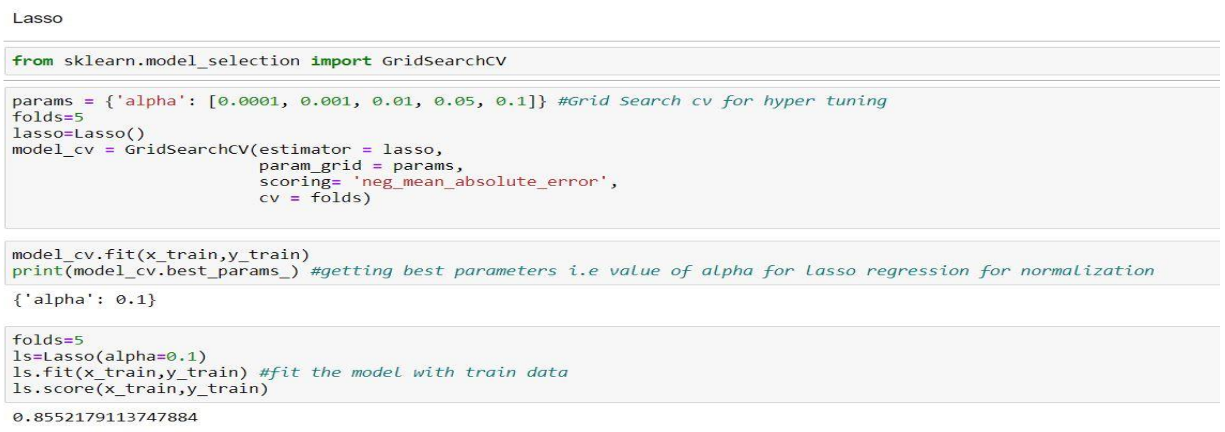
```
: model_cv1.fit(x_train,y_train)
  print(model_cv1.best_params_) #getting best parameters i.e value of alpha for lasso regression for normalization
```

```
  {'alpha': 0.1}
```

```
: folds=5
  rs=Ridge(alpha=0.1)
  rs.fit(x_train,y_train) #fit the model with train data
  rs.score(x_train,y_train)
```

```
: 0.8551662664691501
```

Lasso Regression



Lasso

```
from sklearn.model_selection import GridSearchCV
```

```
params = {'alpha': [0.0001, 0.001, 0.01, 0.05, 0.1]} #Grid Search cv for hyper tuning
folds=5
lasso=Lasso()
model_cv = GridSearchCV(estimator = lasso,
                        param_grid = params,
                        scoring= 'neg_mean_absolute_error',
                        cv = folds)
```

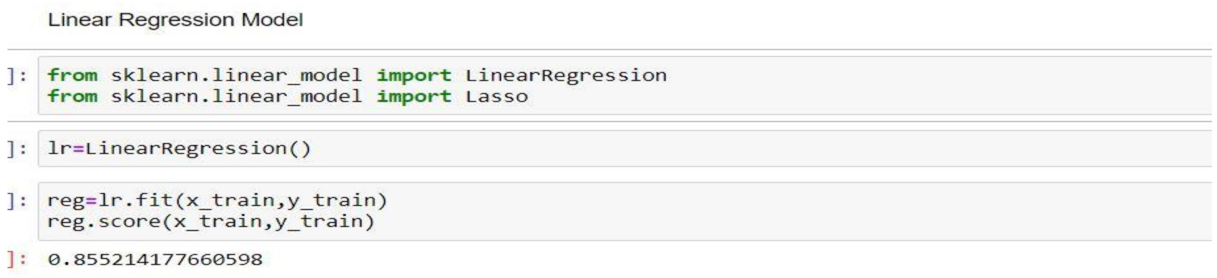
```
model_cv.fit(x_train,y_train)
print(model_cv.best_params_) #getting best parameters i.e value of alpha for lasso regression for normalization
```

```
{'alpha': 0.1}
```

```
folds=5
ls=Lasso(alpha=0.1)
ls.fit(x_train,y_train) #fit the model with train data
ls.score(x_train,y_train)
```

```
0.8552179113747884
```

Linear Regression



Linear Regression Model

```
]: from sklearn.linear_model import LinearRegression
   from sklearn.linear_model import Lasso
```

```
]: lr=LinearRegression()
```

```
]: reg=lr.fit(x_train,y_train)
   reg.score(x_train,y_train)
```

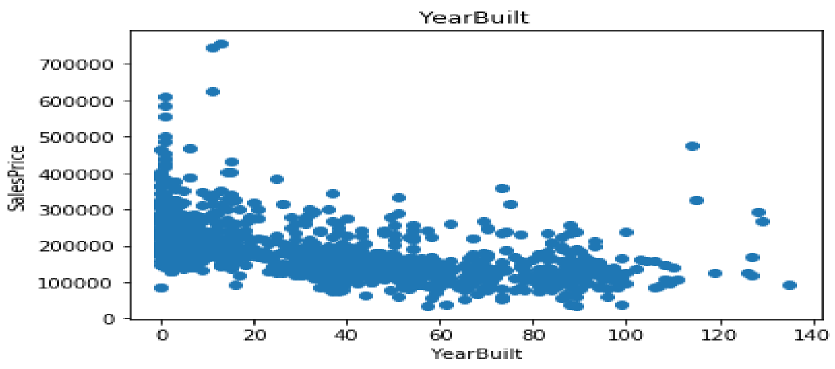
```
]: 0.855214177660598
```

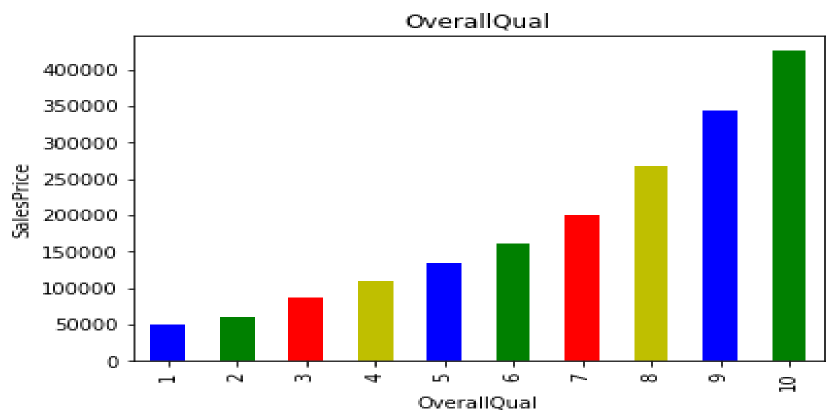
- ## Key Metrics for success in solving problem under consideration.

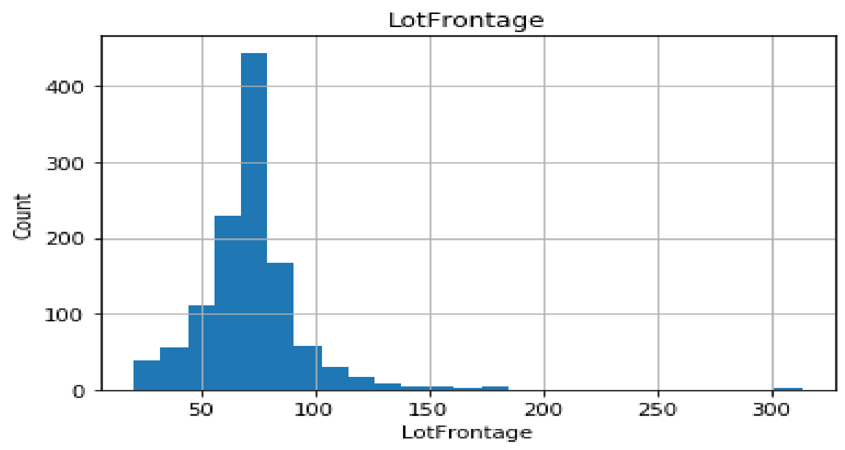
  Statistical Metrics (Correlation) used and for its visualization seaborn heatmap used. Which gives very good idea about correlation value to establish relationship between dependent and independent variable.

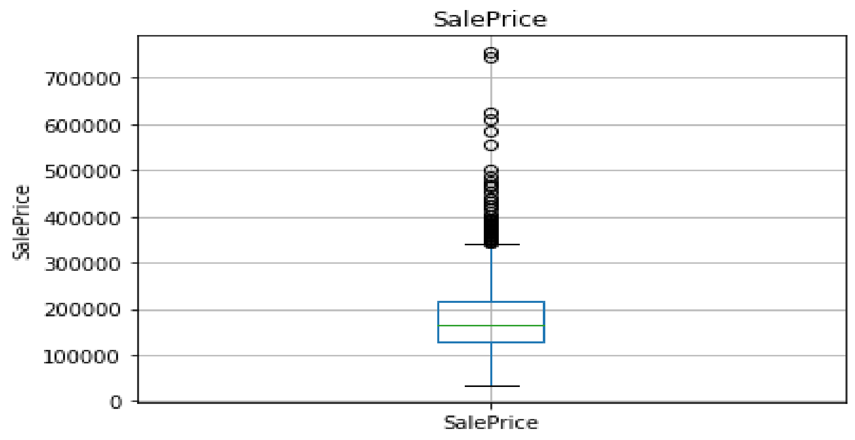
- ## Visualizations

  For visualizations seaborn and matplotlib library are used. And some of plots were used Box plot, Line graph. Histogram, Count plot, Heatmap, distplot and scatterplot. Some of the graphs are depicted below-









- ## Interpretation of the Results-

  Some of the features are not that much correlated are needed to be removed for robust result. Tried few linear regression models to get optimal score. Out of those Random Forest give good score ie. 0.9747. This model predicts the sales price with very close difference with actual value. While doing EDA its visible there are some features are strongly correlated to output prediction model such as OverallQual (Rates the overall material and finish of the house), GrLivArea (Above grade (ground) living area square feet) etc.