HOUSING: PRICE PREDICTION

Submitted by

Steffin Varghese

Batch - 26

In partial fulfillment of Data Science – Internship

At

Flip Robo Technologies

AI and Software Development company

Flip Robo Technologies | Indiranagar, Bangalore - 560 038, Karnataka, India

Month of Submission

June 2022

Acknowledgement

I am highly indebted to FlipRobo Technologies for giving me this opportunity to work on a project and for the guidance and persistent supervision, as well as for providing necessary project information and assistance in completing the project.

I would want to convey my gratitude to the members of FlipRobo Technologies for their kind encouragement and support in completing this project.

I would like to extend my heartfelt gratitude and appreciation to SME, Miss. Khushboo Garg for spending such close attention to me and assisting me during the project's completion, as well as towards others who have volunteered to assist me with their skills.

I acknowledge my gratitude towards the authors of papers : “Machine Learning for Property Price Prediction and Price Valuation”, “Housing Price Prediction with Machine Learning” and “Real Estate Price Prediction” for the insights I could attain through the extensive research which enhanced my knowledge in development of the project.

CONTENTS

INTRODUCTION

Analytical Problem Framing

Model/s Development and Evaluation

CONCLUSION

INTRODUCTION

Business Problem Framing

Houses are one of the necessary needs 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. 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.

Conceptual Background of the Domain Problem

Housing and Real Estate is a very large market and there are various companies working n 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. For the same purpose, **Surprise Housing** has collected a data set from the sale of houses in Australia. The company requirement 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. With this Machine Learning model, the company should be able to derive the insights on:

- Which variables are important to predict the price of variable?
- How do these variables describe the price of the house?

The aim is 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.

Review of Literature

MACHINE LEARNING FOR PROPERTY PRICE PREDICTION AND PRICE VALUATION(2021), (Nur Shahirah Ja'afar1, Junainah Mohamad2, Suriatini Ismail3).

This paper reviews published articles with machine learning techniques that uses optimal solutions in predicting the price of house properties and its valuation. The application of machine learning is used in predicting the house property price and valuation based on location, land and property size, number of rooms and other various factors. The study shows that there is extensive infusion of machine learning in the real estate market which is a crucial factor in predicting the house price and its valuation. This article summaries few suggestions on the potential of machine learning algorithms in the real estate market.

Housing Price Prediction with Machine Learning(2022), (Amena Begum, Nishad Jahan Kheya, Md. Zahidur Rahman).

This study is based on prediction of house price using major three machine learning algorithms : Linear Regression, Decision Tree and Random Forest Regression. The performance of these three algorithms are compared and analyzed based on their pros and cons in using them as the model for house price prediction. This study is looking to draw inferences on methodological and practical contributions to property appraisal and automation through machine learning in the valuation of housing costs. This study also provides a scope for further improvements in the machine learning enables house price predictions in future with more features, historical housing data patterns beyond housing development.

Real Estate Price Prediction(2021), (Smith Dabreo , Shaleel Rodrigues , Valiant Rodrigues , Parshvi Shah)

The primary aim of this study is the adoption of Machine Learning Techniques and curate them into ML models which can then serve the users in the real estate market. This study is primarily oriented on buyers in a way to buy house properties which are not over priced and for this purpose machine learning algorithms are used for house valuation. Along with that, this paper gives an insights on breaking the complexity in house price valuation through machine learning which will help the real estate companies from a lot of hassles which can help to reduce time and cost on research and analysis of the price valuation of properties.

Motivation for the Problem Undertaken

Real Estate has become more than a necessity in this 21st century, it represents something much more nowadays. The demand for house properties are rising year by year. This has subsequently increased the price for the prospect and potential house properties. Most of the times, these increase in prices are overvalued by the market makes or undervalued by the genuine sellers. These are some heuristic approaches of house property valuation. This manual process is always time consuming and inappropriate as this can always omits the major factors that needs to be considered in the price prediction of house properties.

With the help of machine learning and data science, on in simple terms, with the help of statistics, the market makers, buyers, or all the parties involved in the real estate market will be benefitted in their businesses. Machine learning enables to analyze the valuation of a property with the help of powerful machine learning algorithms that can learn the patterns and correlations of various factors affecting the valuation of houses with the help of historical data.

These models always have an option for recalibrations with the changing trends and patterns of the real estate markets. Because the real estate markets are always fluctuating and subject to various economic conditions, money supply, weather and are influenced by various other factors.

With the right machine learning models, it is possible to make a change in the potential real estate market in predicting the price predictions and its valuations. This will also allow us to the automation of house price predictions without human interventions with more accuracy and less time consuming.

Analytical Problem Framing

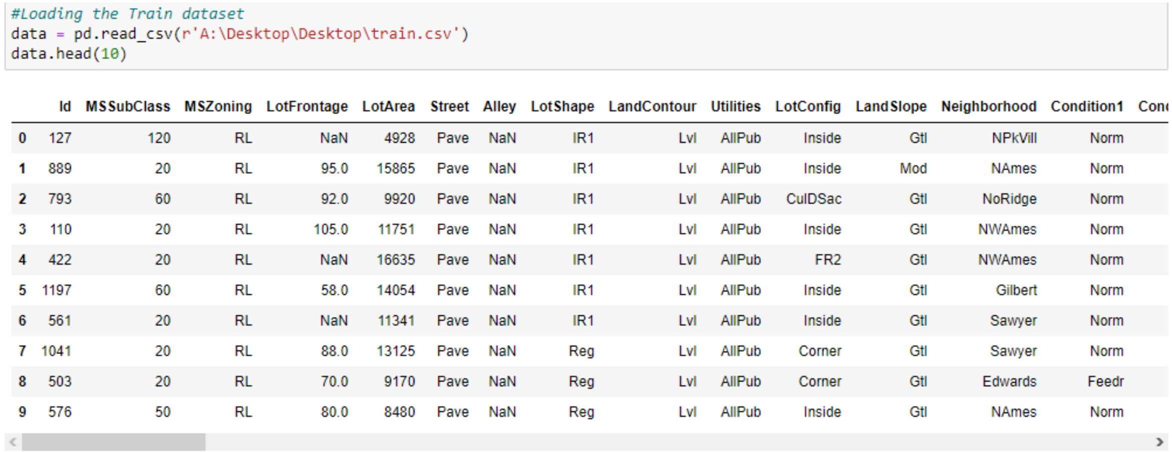
Mathematical/ Analytical Modeling of the Problem

As per the data source and requirements, we have to predict the house prices using various features regarding the house properties. Since we have provided with the historical transactions and data of house which are sold, we have been given with the output variable for training the machine. Supervised learning is the best machine learning approach in this problem. Since the price of the house is numerical in nature, we can use the regression approaches as regression methods are best in analyzing numerical output predictions based on various independent features. We have used various regression techniques such as linear regression models, regularization techniques to identify the over fitting of the models, various boosting algorithms and ensemble methods which consists of a fusion of these algorithms which makes them powerful in machine learning and making the predictions. We have also used several preprocessing techniques to process the raw data to be ready for the machine for learning. In the process as per the requirements, we have also made several inferences on the relation of various features of the houses or the independent information about the houses with the target variable or the sale price of the house. This includes the identification of features which are making an impact on the sale price of a house and the importance of each independent variables that are influencing the valuation of a house property.

Data Sources and their formats

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. This is the dataset which we have used for the analysis and mode building through machine learning.

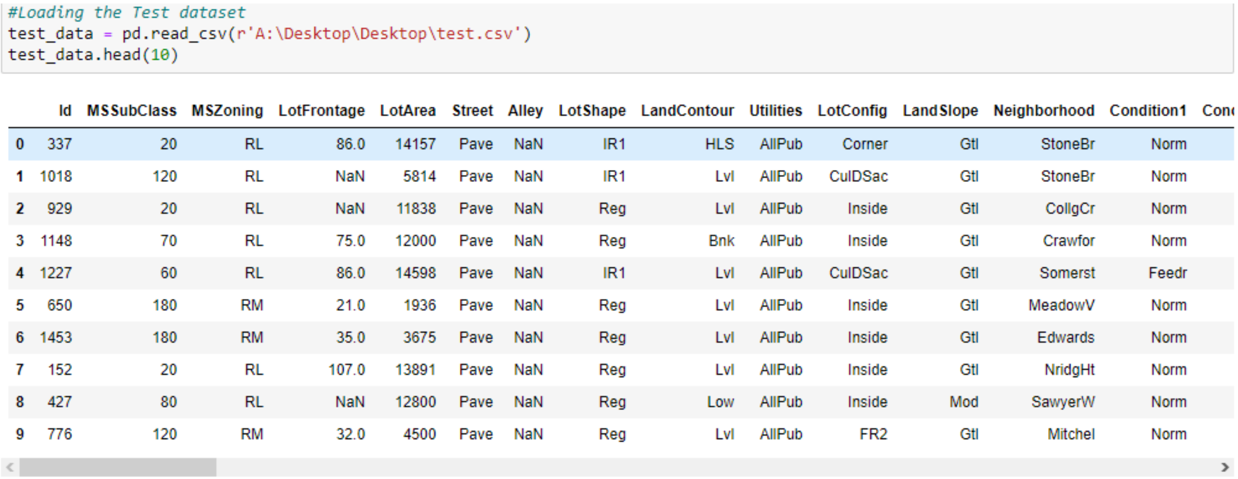
Two datasets are being provided to you (test.csv, train.csv). You will train on train.csv dataset and predict on test.csv file.



```
#Loading the Train dataset
data = pd.read_csv(r'A:\Desktop\Desktop\train.csv')
data.head(10)
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Con
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	

Snapshot of Train Dataset



```
#Loading the Test dataset
test_data = pd.read_csv(r'A:\Desktop\Desktop\test.csv')
test_data.head(10)
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Con
0	337	20	RL	86.0	14157	Pave	NaN	IR1	HLS	AllPub	Corner	Gtl	StoneBr	Norm	
1	1018	120	RL	NaN	5814	Pave	NaN	IR1	Lvl	AllPub	CulDSac	Gtl	StoneBr	Norm	
2	929	20	RL	NaN	11838	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	
3	1148	70	RL	75.0	12000	Pave	NaN	Reg	Bnk	AllPub	Inside	Gtl	Crawfor	Norm	
4	1227	60	RL	86.0	14598	Pave	NaN	IR1	Lvl	AllPub	CulDSac	Gtl	Somerst	Feedr	
5	650	180	RM	21.0	1936	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	MeadowV	Norm	
6	1453	180	RM	35.0	3675	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Edwards	Norm	
7	152	20	RL	107.0	13891	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	NridgHt	Norm	
8	427	80	RL	NaN	12800	Pave	NaN	Reg	Low	AllPub	Inside	Mod	SawyerW	Norm	
9	776	120	RM	32.0	4500	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl	Mitchel	Norm	

Snapshot of Test Dataset

- Data contains 1460 entries each having 81 variables
- We have string, float and integer type of data in the train and test dataset.
- We have 1168 non-null values in all the variables in train data except the columns ['Alley', 'LotFrontage', 'MasVnrType', 'MasVnrArea', 'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2', 'FireplaceQu', 'GarageType', 'GarageYrBlt', 'GarageFinish', 'GarageQual', 'GarageCond', 'PoolQC', 'Fence', 'MiscFeature'].
- We have 292 non-null values in all the variables in test data except the columns ['Alley', 'LotFrontage', 'MasVnrType', 'MasVnrArea', 'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2', 'Electrical', 'FireplaceQu', 'GarageType', 'GarageYrBlt', 'GarageFinish', 'GarageQual', 'GarageCond', 'PoolQC', 'Fence', 'MiscFeature'].

Description of the dataset

Features in Dataset(Independent Variable)

Id - Unique number to identify each property

MSSubClass - Identifies the type of dwelling involved in the sale.

MSZoning - Identifies the general zoning classification of the sale.

LotFrontage - Linear feet of street connected to property

LotArea - Lot size in square feet

Street - Type of road access to property

Alley - Type of alley access to property

LotShape - General shape of property

LandContour - Flatness of the property

Utilities - Type of utilities available

LotConfig - Lot configuration

LandSlope - Slope of property

Neighborhood - Physical locations within Ames city limits

Condition1 - Proximity to various conditions

Condition2 - Proximity to various conditions (if more than one is present)

BldgType - Type of dwelling

HouseStyle - Style of dwelling

OverallQual - Rates the overall material and finish of the house

OverallCond - Rates the overall condition of the house

YearBuilt - Original construction date

YearRemodAdd - Remodel date (same as construction date if no remodeling or additions)

RoofStyle - Type of roof

RoofMatl - Roof material

Exterior1st - Exterior covering on house

Exterior2nd - Exterior covering on house (if more than one material)

MasVnrType - Masonry veneer type

MasVnrArea - Masonry veneer area in square feet

ExterQual - Evaluates the quality of the material on the exterior

ExterCond - Evaluates the present condition of the material on the exterior

Foundation - Type of foundation

BsmtQual - Evaluates the height of the basement

BsmtCond - Evaluates the general condition of the basement

BsmtExposure - Refers to walkout or garden level walls

BsmtFinType1 - Rating of basement finished area

BsmtFinSF1 - Type 1 finished square feet

BsmtFinType2 - Rating of basement finished area (if multiple types)

BsmtFinSF2 - Type 2 finished square feet

BsmtUnfSF - Unfinished square feet of basement area

TotalBsmtSF - Total square feet of basement area

Heating - Type of heating

HeatingQC - Heating quality and condition

CentralAir - Central air conditioning

Electrical - Electrical system

1stFlrSF - First Floor square feet

2ndFlrSF - Second floor square feet

LowQualFinSF - Low quality finished square feet (all floors)

GrLivArea - Above grade (ground) living area square feet

BsmtFullBath - Basement full bathrooms

BsmtHalfBath - Basement half bathrooms

FullBath - Full bathrooms above grade

HalfBath - Half baths above grade

BedroomAbvGr - Bedrooms above grade (does NOT include basement bedrooms)

KitchenAbvGr - Kitchens above grade

KitchenQual - Kitchen quality

TotRmsAbvGrd - Total rooms above grade (does not include bathrooms)

Functional - Home functionality (Assume typical unless deductions are warranted)

Fireplaces - Number of fireplaces

FireplaceQu - Fireplace quality

GarageType - Garage location

GarageYrBlt - Year garage was built

GarageFinish - Interior finish of the garage

GarageCars - Size of garage in car capacity

GarageArea - Size of garage in square feet

GarageQual - Garage quality

GarageCond - Garage condition

PavedDrive - Paved driveway

WoodDeckSF - Wood deck area in square feet

OpenPorchSF - Open porch area in square feet

EnclosedPorch - Enclosed porch area in square feet

3SsnPorch - Three season porch area in square feet

ScreenPorch - Screen porch area in square feet

PoolArea - Pool area in square feet

PoolQC - Pool quality

Fence - Fence quality

MiscFeature - Miscellaneous feature not covered in other categories

MiscVal - $Value of miscellaneous feature

MoSold - Month Sold (MM)

YrSold - Year Sold (YYYY)

SaleType - Type of sale

SaleCondition - Condition of sale

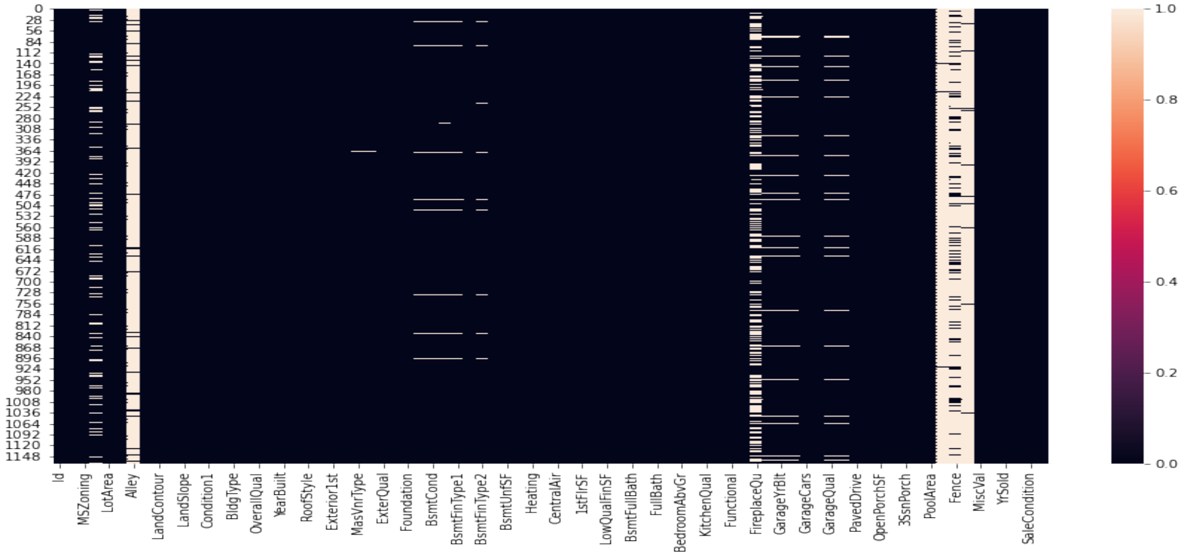
Target in dataset(Dependent Variable)

SalePrice - The $sale price of the property

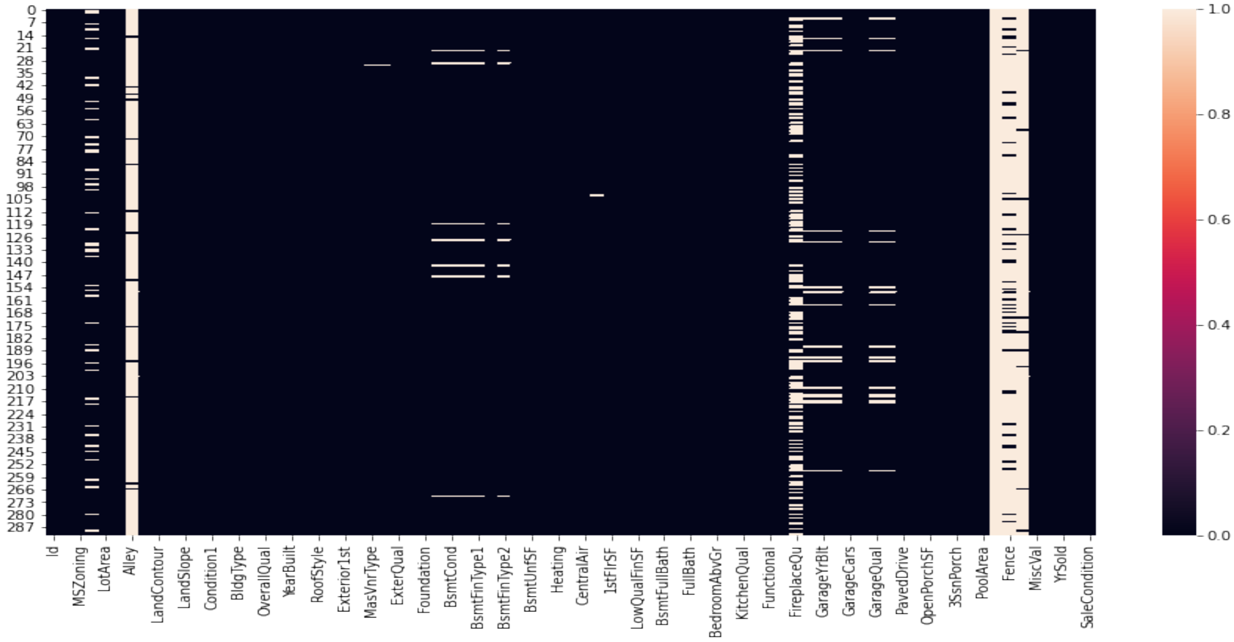
Explored the categorical and numerical variables to check whether we have any unusual data values present in the test and train data. The column 'Utilities' in train dataset is having only one category. So, it will not be helpful to train the machine with other categories.

Data Preprocessing

Checking for Missing Values

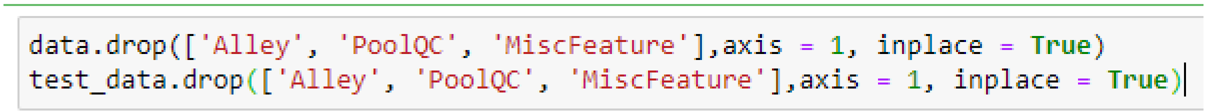


Missing data in train dataset



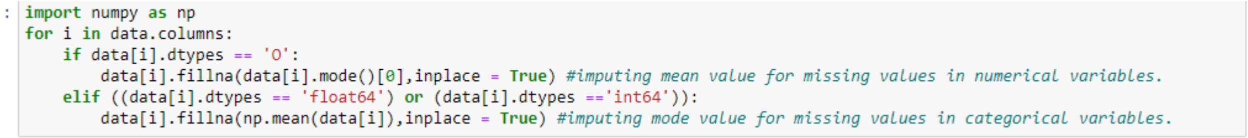
Missing data in test dataset

Since the columns ['Alley', 'PoolQC', 'MiscFeature'] are having more than 90% missing values, we dropped these columns from our train and test data as imputing a different value for the missing values in these columns will become biased in model prediction.



```
data.drop(['Alley', 'PoolQC', 'MiscFeature'],axis = 1, inplace = True)
test_data.drop(['Alley', 'PoolQC', 'MiscFeature'],axis = 1, inplace = True)
```

Dropped the column with more than 90% missing values



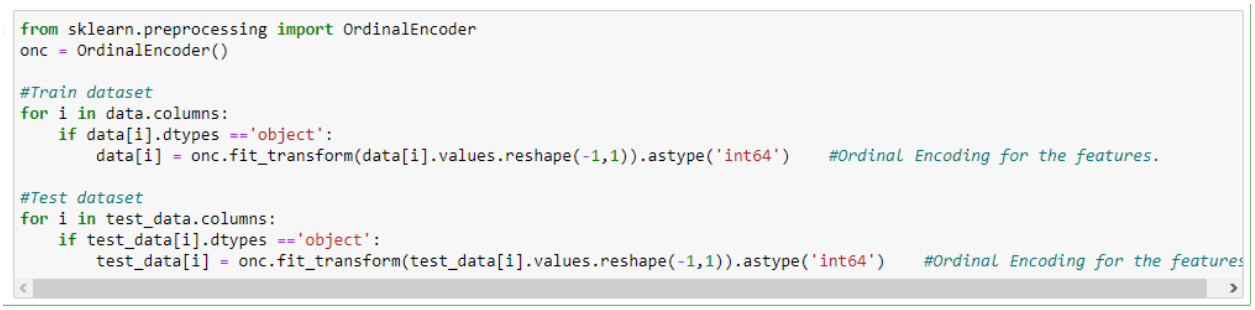
```
: import numpy as np
  for i in data.columns:
      if data[i].dtypes == 'O':
          data[i].fillna(data[i].mode()[0],inplace = True) #imputing mean value for missing values in numerical variables.
      elif ((data[i].dtypes == 'float64') or (data[i].dtypes =='int64')):
          data[i].fillna(np.mean(data[i]),inplace = True) #imputing mode value for missing values in categorical variables.
```

Imputation of mean and mode value for missing values in dataset

We imputed the mean value for missing values in numerical columns and mode value of column for missing values in categorical columns in train dataset.

Encoded the Categorical Columns

In Machine Learning, the machine can only learn numbers. For this purpose, we encode our categorical variables. Here we have used Ordinal Encoder as we only need to encode some of the features in the train and test datasets.

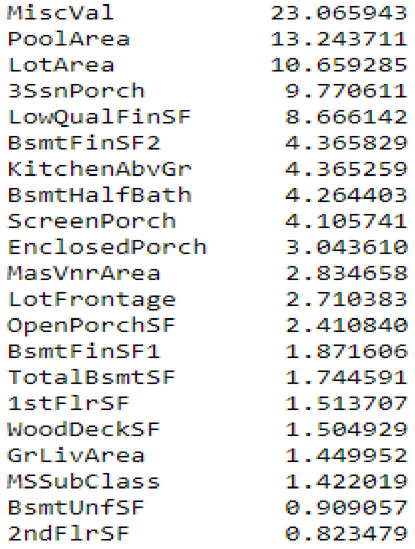


```
from sklearn.preprocessing import OrdinalEncoder
onc = OrdinalEncoder()

#Train dataset
for i in data.columns:
    if data[i].dtypes =='object':
        data[i] = onc.fit_transform(data[i].values.reshape(-1,1)).astype('int64')    #Ordinal Encoding for the features.

#Test dataset
for i in test_data.columns:
    if test_data[i].dtypes =='object':
        test_data[i] = onc.fit_transform(test_data[i].values.reshape(-1,1)).astype('int64')    #Ordinal Encoding for the features
```

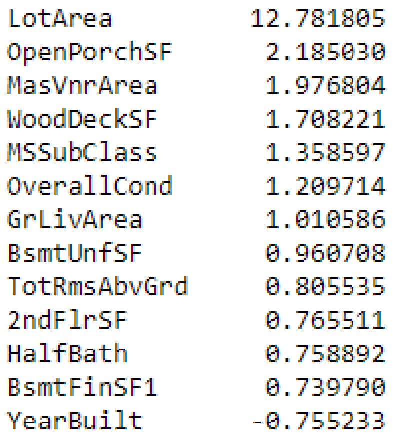
Encoding the Categorical Variable data

Data Cleansing

Removing the skewness with power transform



```
MiscVal          23.065943
PoolArea         13.243711
LotArea          10.659285
3SsnPorch         9.770611
LowQualFinSF      8.666142
BsmtFinSF2        4.365829
KitchenAbvGr      4.365259
BsmtHalfBath      4.264403
ScreenPorch       4.105741
EnclosedPorch     3.043610
MasVnrArea        2.834658
LotFrontage       2.710383
OpenPorchSF       2.410840
BsmtFinSF1        1.871606
TotalBsmtSF       1.744591
1stFlrSF          1.513707
WoodDeckSF        1.504929
GrLivArea         1.449952
MSSubClass        1.422019
BsmtUnfSF         0.909057
2ndFlrSF          0.823479
```

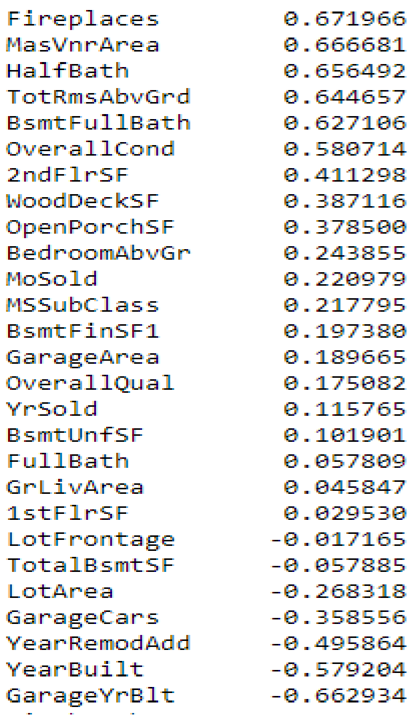


```
LotArea         12.781805
OpenPorchSF      2.185030
MasVnrArea       1.976804
WoodDeckSF       1.708221
MSSubClass       1.358597
OverallCond      1.209714
GrLivArea        1.010586
BsmtUnfSF        0.960708
TotRmsAbvGrd     0.805535
2ndFlrSF         0.765511
HalfBath         0.758892
BsmtFinSF1       0.739790
YearBuilt       -0.755233
```

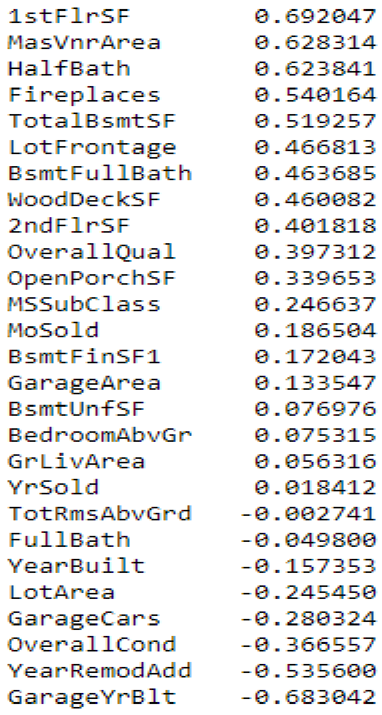
Numerical columns in test data with skewness beyond +/-0.7

Numerical columns in train data with skewness beyond +/-0.7

After removing the skewness from the data using standard scaler and power transform our data looks like this:



```
Fireplaces        0.671966
MasVnrArea        0.666681
HalfBath          0.656492
TotRmsAbvGrd      0.644657
BsmtFullBath      0.627106
OverallCond       0.580714
2ndFlrSF          0.411298
WoodDeckSF        0.387116
OpenPorchSF       0.378500
BedroomAbvGr      0.243855
MoSold            0.220979
MSSubClass        0.217795
BsmtFinSF1        0.197380
GarageArea        0.189665
OverallQual       0.175082
YrSold            0.115765
BsmtUnfSF         0.101901
FullBath          0.057809
GrLivArea         0.045847
1stFlrSF          0.029530
LotFrontage      -0.017165
TotalBsmtSF      -0.057885
LotArea          -0.268318
GarageCars       -0.358556
YearRemodAdd     -0.495864
YearBuilt        -0.579204
GarageYrBlt      -0.662934
```



```
1stFlrSF          0.692047
MasVnrArea        0.628314
HalfBath          0.623841
Fireplaces        0.540164
TotalBsmtSF       0.519257
LotFrontage       0.466813
BsmtFullBath      0.463685
WoodDeckSF        0.460082
2ndFlrSF          0.401818
OverallQual       0.397312
OpenPorchSF       0.339653
MSSubClass        0.246637
MoSold            0.186504
BsmtFinSF1        0.172043
GarageArea        0.133547
BsmtUnfSF         0.076976
BedroomAbvGr      0.075315
GrLivArea         0.056316
YrSold            0.018412
TotRmsAbvGrd     -0.002741
FullBath         -0.049800
YearBuilt        -0.157353
LotArea          -0.245450
GarageCars       -0.280324
OverallCond      -0.366557
YearRemodAdd     -0.535600
GarageYrBlt      -0.683042
```

Skewness of Train data

Skewness of Test data

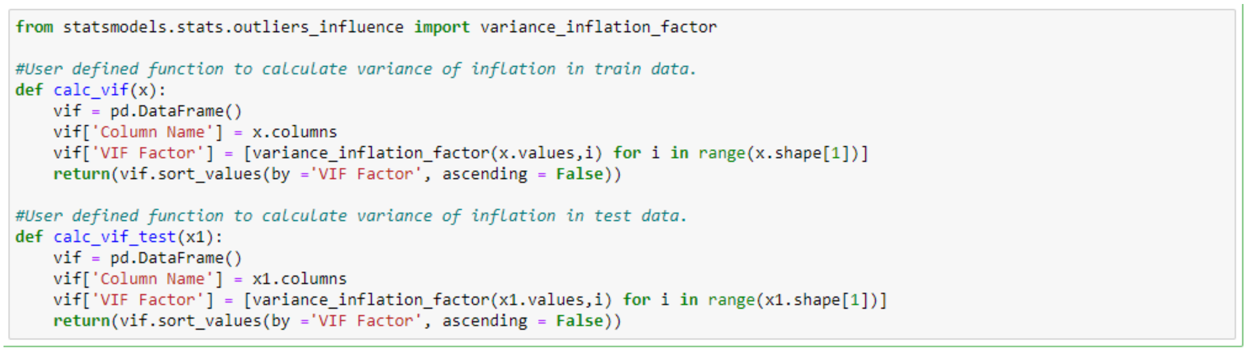
Removed Outliers in the datasets

We have used two methods for outlier removal method:

- Outlier removal using ZScore Technique
- Outlier removal using IQR(Inter Quartile Range) Technique.

  By using the IQR method for outlier removal from the dataset, we have incurred a loss of data of 28% which is not acceptable as we are losing a major portion of the data. This would make our data biased. But after using the ZScore outlier removal method we are losing only 10.87% of data from the train dataset. So, we are considering this method for removing outliers from our train and test dataset.

Checked and Removed Multicollinearity from the Datasets.

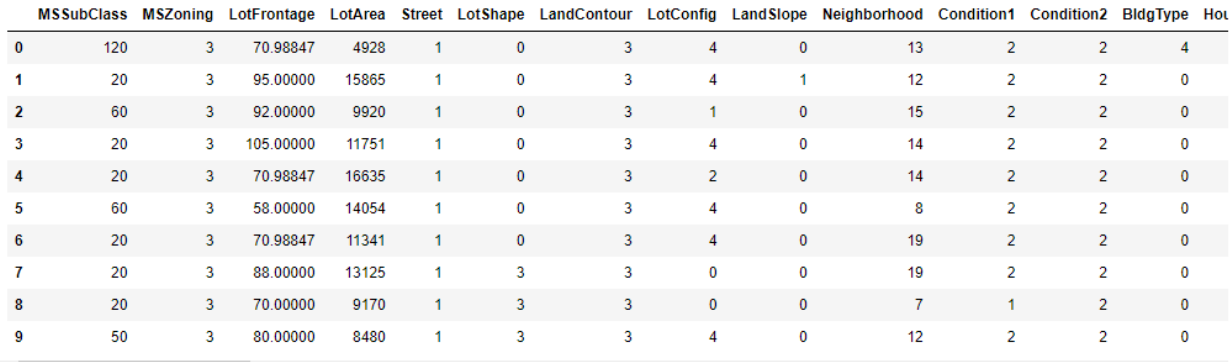


```
from statsmodels.stats.outliers_influence import variance_inflation_factor

#User defined function to calculate variance of inflation in train data.
def calc_vif(x):
    vif = pd.DataFrame()
    vif['Column Name'] = x.columns
    vif['VIF Factor'] = [variance_inflation_factor(x.values,i) for i in range(x.shape[1])]
    return(vif.sort_values(by ='VIF Factor', ascending = False))

#User defined function to calculate variance of inflation in test data.
def calc_vif_test(x1):
    vif = pd.DataFrame()
    vif['Column Name'] = x1.columns
    vif['VIF Factor'] = [variance_inflation_factor(x1.values,i) for i in range(x1.shape[1])]
    return(vif.sort_values(by ='VIF Factor', ascending = False))
```

User defined function to calculate variance of inflation

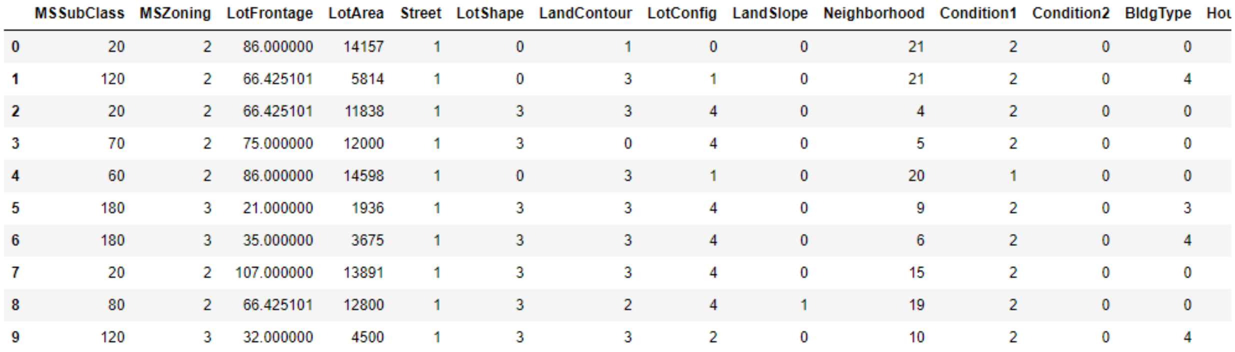
After calculating the VIF of the variables in the datasets, we found that the columns ['GrLivArea','TotalBsmtSF', 'YearBuilt','2ndFlrSF', 'BsmtFinSF1', 'GarageCars'] are having high variance of inflation. Thus, we had to drop these columns from the test and train dataset.

Final Dataset



	MSSubClass	MSZoning	LotFrontage	LotArea	Street	LotShape	LandContour	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType	Hou
0	120	3	70.98847	4928	1	0	3	4	0	13	2	2	4	
1	20	3	95.00000	15865	1	0	3	4	1	12	2	2	0	
2	60	3	92.00000	9920	1	0	3	1	0	15	2	2	0	
3	20	3	105.00000	11751	1	0	3	4	0	14	2	2	0	
4	20	3	70.98847	16635	1	0	3	2	0	14	2	2	0	
5	60	3	58.00000	14054	1	0	3	4	0	8	2	2	0	
6	20	3	70.98847	11341	1	0	3	4	0	19	2	2	0	
7	20	3	88.00000	13125	1	3	3	0	0	19	2	2	0	
8	20	3	70.00000	9170	1	3	3	0	0	7	1	2	0	
9	50	3	80.00000	8480	1	3	3	4	0	12	2	2	0	

Snapshot of Final Train Dataset



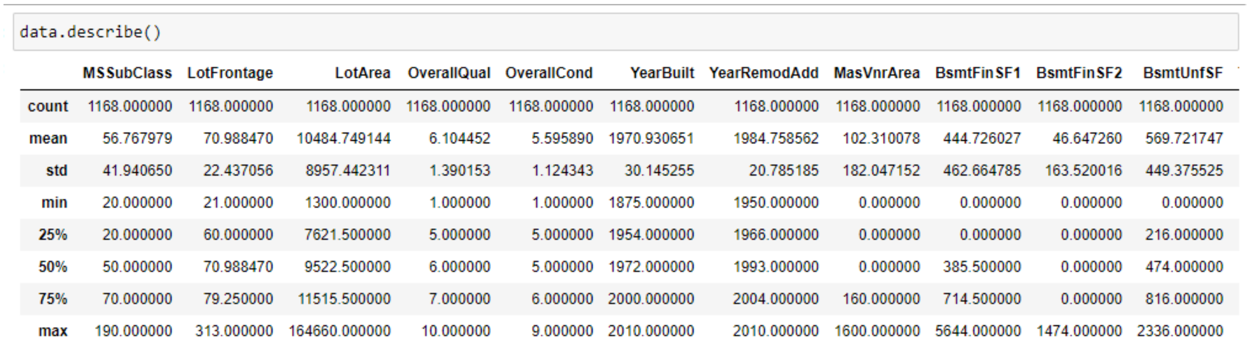
	MSSubClass	MSZoning	LotFrontage	LotArea	Street	LotShape	LandContour	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType	Hou
0	20	2	86.000000	14157	1	0	1	0	0	21	2	0	0	
1	120	2	66.425101	5814	1	0	3	1	0	21	2	0	4	
2	20	2	66.425101	11838	1	3	3	4	0	4	2	0	0	
3	70	2	75.000000	12000	1	3	0	4	0	5	2	0	0	
4	60	2	86.000000	14598	1	0	3	1	0	20	1	0	0	
5	180	3	21.000000	1936	1	3	3	4	0	9	2	0	3	
6	180	3	35.000000	3675	1	3	3	4	0	6	2	0	4	
7	20	2	107.000000	13891	1	3	3	4	0	15	2	0	0	
8	80	2	66.425101	12800	1	3	2	4	1	19	2	0	0	
9	120	3	32.000000	4500	1	3	3	2	0	10	2	0	4	

Snapshot of Final Test Dataset

Data Inputs- Logic- Output Relationships

Statistical Summary

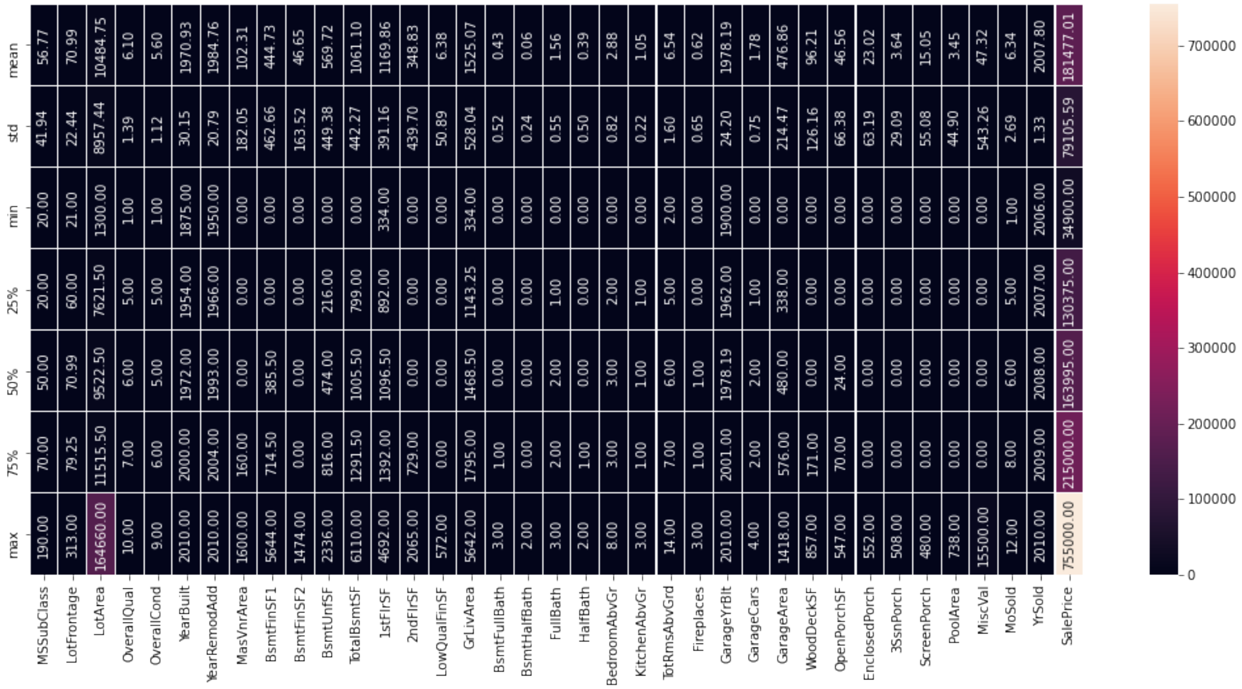
Describe of the data



```
data.describe()
```

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	BsmtUnfSF
count	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000	1168.000000
mean	56.767979	70.988470	10484.749144	6.104452	5.595890	1970.930651	1984.758562	102.310078	444.726027	46.647260	569.721747
std	41.940650	22.437056	8957.442311	1.390153	1.124343	30.145255	20.785185	182.047152	462.664785	163.520016	449.375525
min	20.000000	21.000000	1300.000000	1.000000	1.000000	1875.000000	1950.000000	0.000000	0.000000	0.000000	0.000000
25%	20.000000	60.000000	7621.500000	5.000000	5.000000	1954.000000	1966.000000	0.000000	0.000000	0.000000	216.000000
50%	50.000000	70.988470	9522.500000	6.000000	5.000000	1972.000000	1993.000000	0.000000	385.500000	0.000000	474.000000
75%	70.000000	79.250000	11515.500000	7.000000	6.000000	2000.000000	2004.000000	160.000000	714.500000	0.000000	816.000000
max	190.000000	313.000000	164660.000000	10.000000	9.000000	2010.000000	2010.000000	1600.000000	5644.000000	1474.000000	2336.000000

Snapshot of describe of the data



	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	BsmtUnfSF	TotalBsmtSF	1stFlrSF	2ndFlrSF	LowQualFinSF	GrLivArea	BsmtFullBath	BsmtHalfBath	FullBath
mean	56.77	70.99	10484.75	6.10	5.60	1970.93	1984.76	102.31	444.73	46.65	569.72	1061.10	1169.86	348.83	6.38	1525.07	0.43	0.06	1.56
std	41.94	22.44	8957.44	1.39	1.12	30.15	20.79	182.05	462.66	163.52	449.38	442.27	391.16	439.70	50.89	528.04	0.52	0.24	0.55
min	20.00	21.00	1300.00	1.00	1.00	1875.00	1950.00	0.00	0.00	0.00	0.00	0.00	334.00	0.00	0.00	334.00	0.00	0.00	0.00
25%	20.00	60.00	7621.50	5.00	5.00	1954.00	1966.00	0.00	0.00	0.00	216.00	799.00	892.00	0.00	0.00	1143.25	0.00	0.00	1.00
50%	50.00	70.99	9522.50	6.00	5.00	1972.00	1993.00	0.00	385.50	0.00	474.00	1005.50	1096.50	0.00	0.00	1468.50	0.00	0.00	2.00
75%	70.00	79.25	11515.50	7.00	6.00	2000.00	2004.00	160.00	714.50	0.00	816.00	1291.50	1392.00	729.00	0.00	1795.00	1.00	0.00	2.00
max	190.00	313.00	164660.00	10.00	9.00	2010.00	2010.00	1600.00	5644.00	1474.00	2336.00	6110.00	4692.00	2065.00	572.00	5642.00	3.00	2.00	3.00

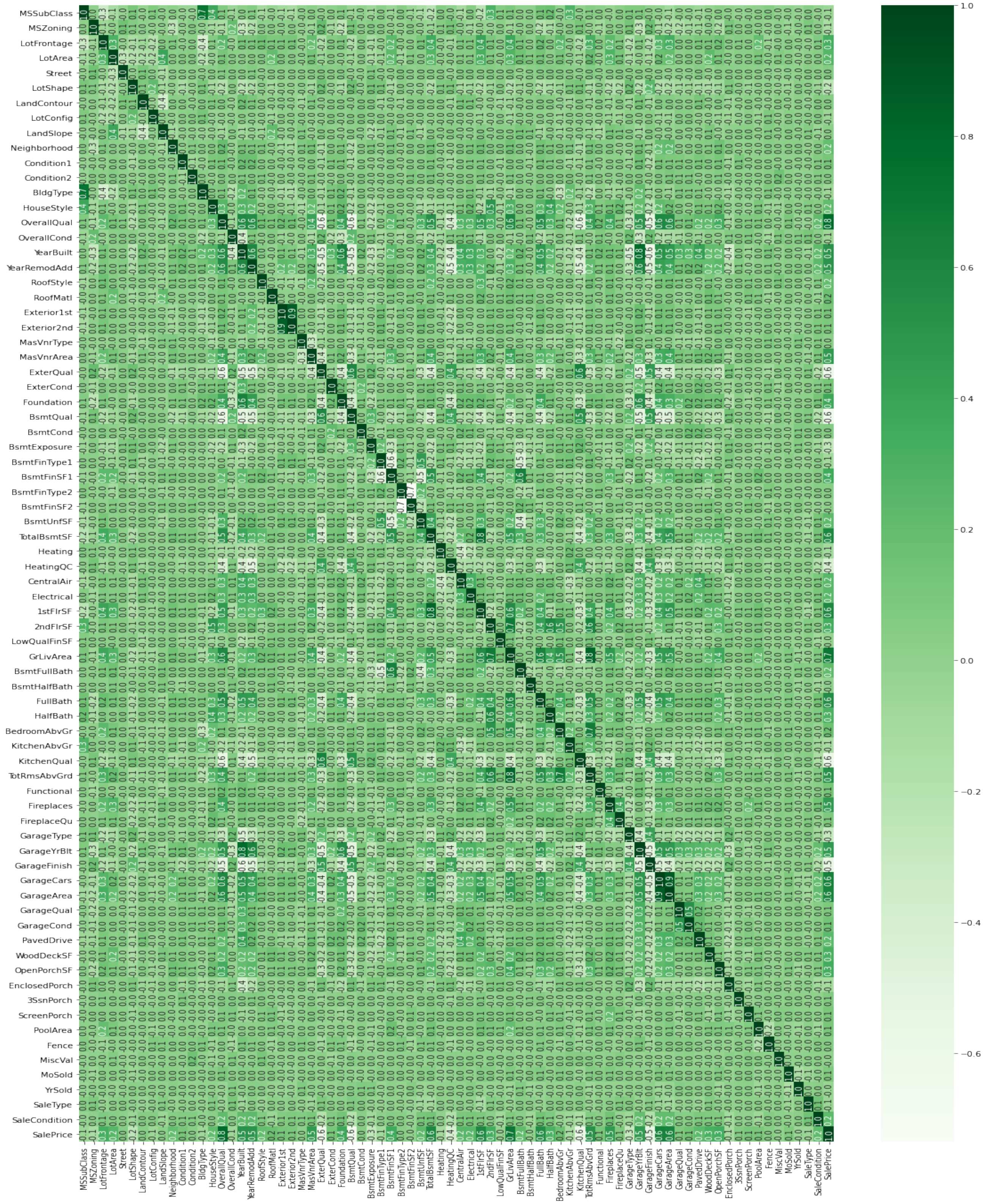
	HalfBath	BedroomAbvGr	KitchenAbvGr	TotRmsAbvGrd	Fireplaces	GarageYrBlt	GarageCars	GarageArea	WoodDeckSF	OpenPorchSF	EnclosedPorch	3SsnPorch	ScreenPorch	PoolArea	MiscVal	MoSold	YrSold	SalePrice
mean	0.39	2.88	1.05	6.54	0.62	1978.19	1.78	476.86	96.21	46.56	23.02	3.64	15.05	3.45	47.32	6.34	2007.80	181477.01
std	0.50	0.82	0.22	1.60	0.65	24.20	0.75	214.47	126.16	66.38	63.19	29.09	55.08	44.90	543.26	2.69	1.33	79105.59
min	0.00	0.00	0.00	2.00	0.00	1900.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	2006.00	34900.00
25%	0.00	2.00	1.00	5.00	0.00	1962.00	1.00	338.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.00	2007.00	130375.00
50%	0.00	3.00	1.00	6.00	1.00	1978.19	2.00	480.00	0.00	24.00	0.00	0.00	0.00	0.00	0.00	6.00	2008.00	163995.00
75%	1.00	3.00	1.00	7.00	1.00	2001.00	2.00	576.00	171.00	70.00	0.00	0.00	0.00	0.00	0.00	8.00	2009.00	215000.00
max	2.00	8.00	3.00	14.00	3.00	2010.00	4.00	1418.00	857.00	547.00	552.00	508.00	480.00	738.00	15500.00	12.00	2010.00	755000.00

Heatmap of the statistical summary of train data

Observations:

- The columns ['MSSubClass', 'LotArea', 'OverallQual', 'OverallCond', 'MasVnrArea', 'BsmtFinSF1'. 'BsmtFinSF2', 'BsmtUnfSF','TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'HalfBath', 'KitchenAbvGr', 'TotRmsAbvGrd', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'MiscVal', 'MoSold', 'SalePrice'] are having higher mean value than the median value. That means skewness is present in the distribution of data of these columns.
- Most of the columns are having high amount of difference between the 75% and the maximum value of the columns. That means, outliers are present in the data of these columns.

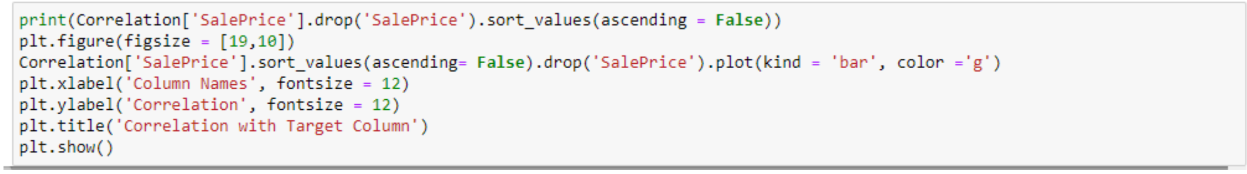
Correlation



Snapshot of Correlation of data variables

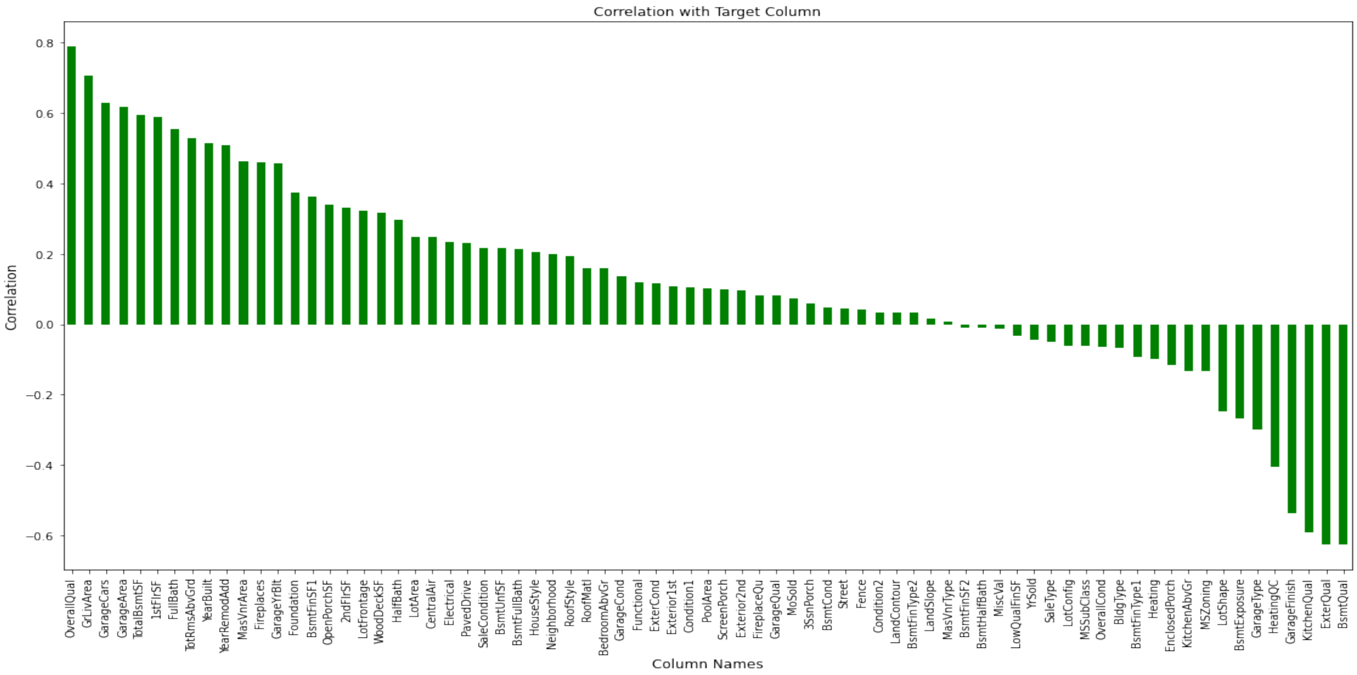
Since there are large number of variables, inferences from correlation of all the columns will not give us any insights. So, we can consider the correlation of features with the target variable.

Correlation with Target Variable



```
print(Correlation['SalePrice'].drop('SalePrice').sort_values(ascending = False))
plt.figure(figsize = [19,10])
Correlation['SalePrice'].sort_values(ascending= False).drop('SalePrice').plot(kind = 'bar', color ='g')
plt.xlabel('Column Names', fontsize = 12)
plt.ylabel('Correlation', fontsize = 12)
plt.title('Correlation with Target Column')
plt.show()
```

Code snippet for plotting correlation of variables with target variable

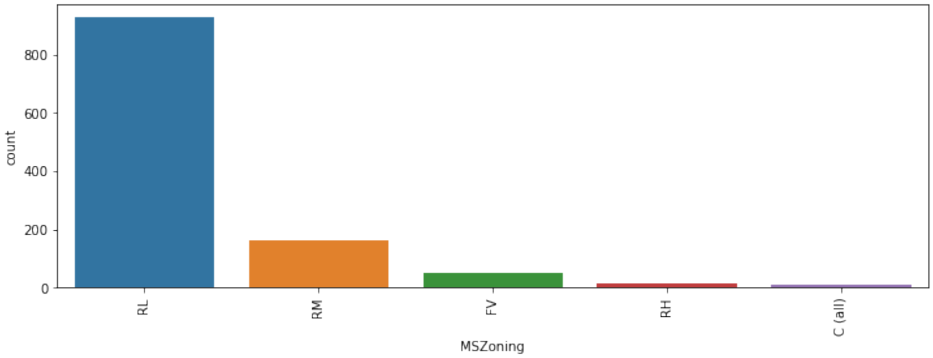


Correlation of Variables with Target Variable

Observations:

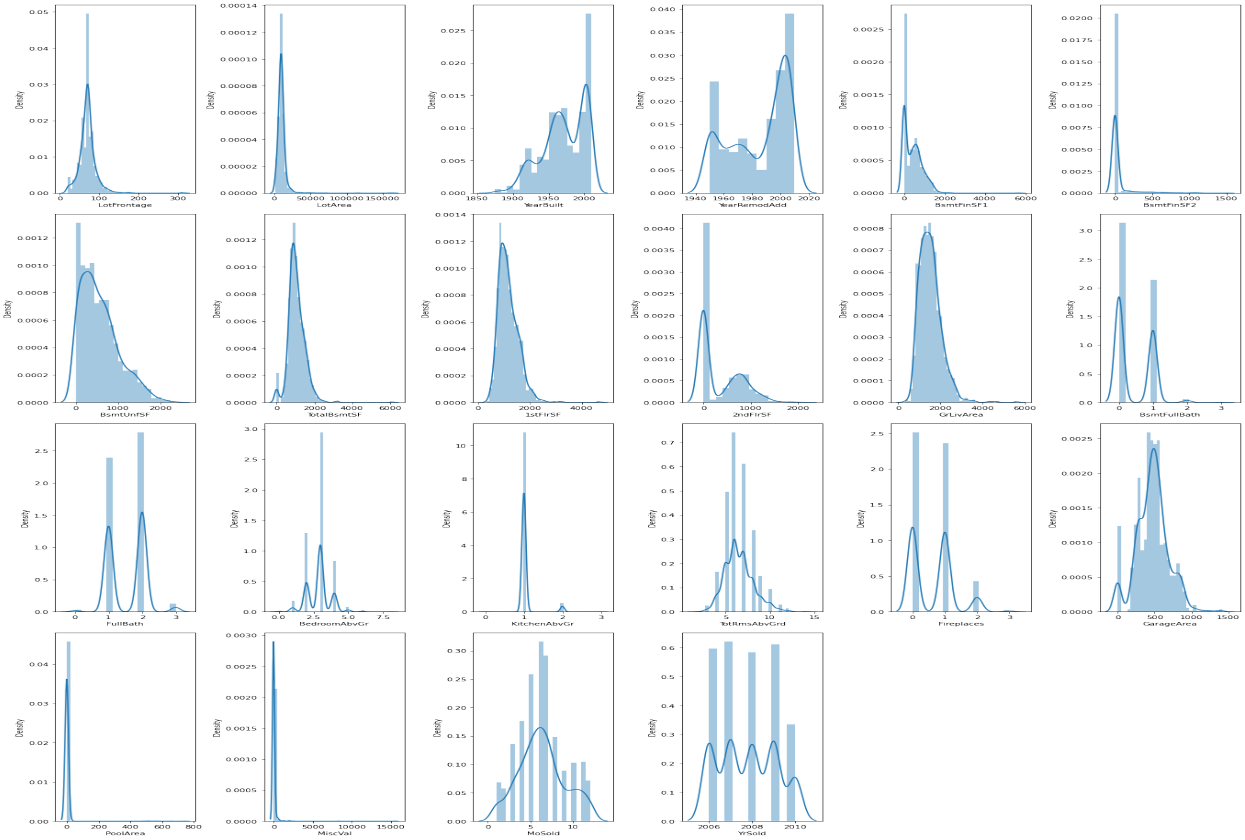
- The columns ['BsmtFinSF2', 'BsmtHalfBath', 'MiscVal', 'LowQualFinSF', 'YrSold', 'SaleType', 'LotConfig', 'MSSubClass', 'OverallCond', 'BldgType', 'BsmtFinType1', 'Heating', 'KitchenAbvGr', 'MSZoning', 'LotShape', 'BsmtExposure', 'GarageType', 'HeatingQC', 'GarageFinish', 'KitchenQual', 'ExterQual', 'BsmtQual'] are negatively correlated to the target variable 'SalePrice'.
- The column 'OverallQual' is having highest positve correlation to the target variable 'SalePrice', while the column 'BsmtQual' is having highest negative correlation to the target variable 'SalePrice'.
- The column 'MasVnrType' is having the least positive correlation to the target variable 'SalePrice' and the column 'BsmtFinSF2' is having the least negative correlation to the target variable 'SalePrice'.

Assumptions Related to the Problem Under Consideration



General zoning classification

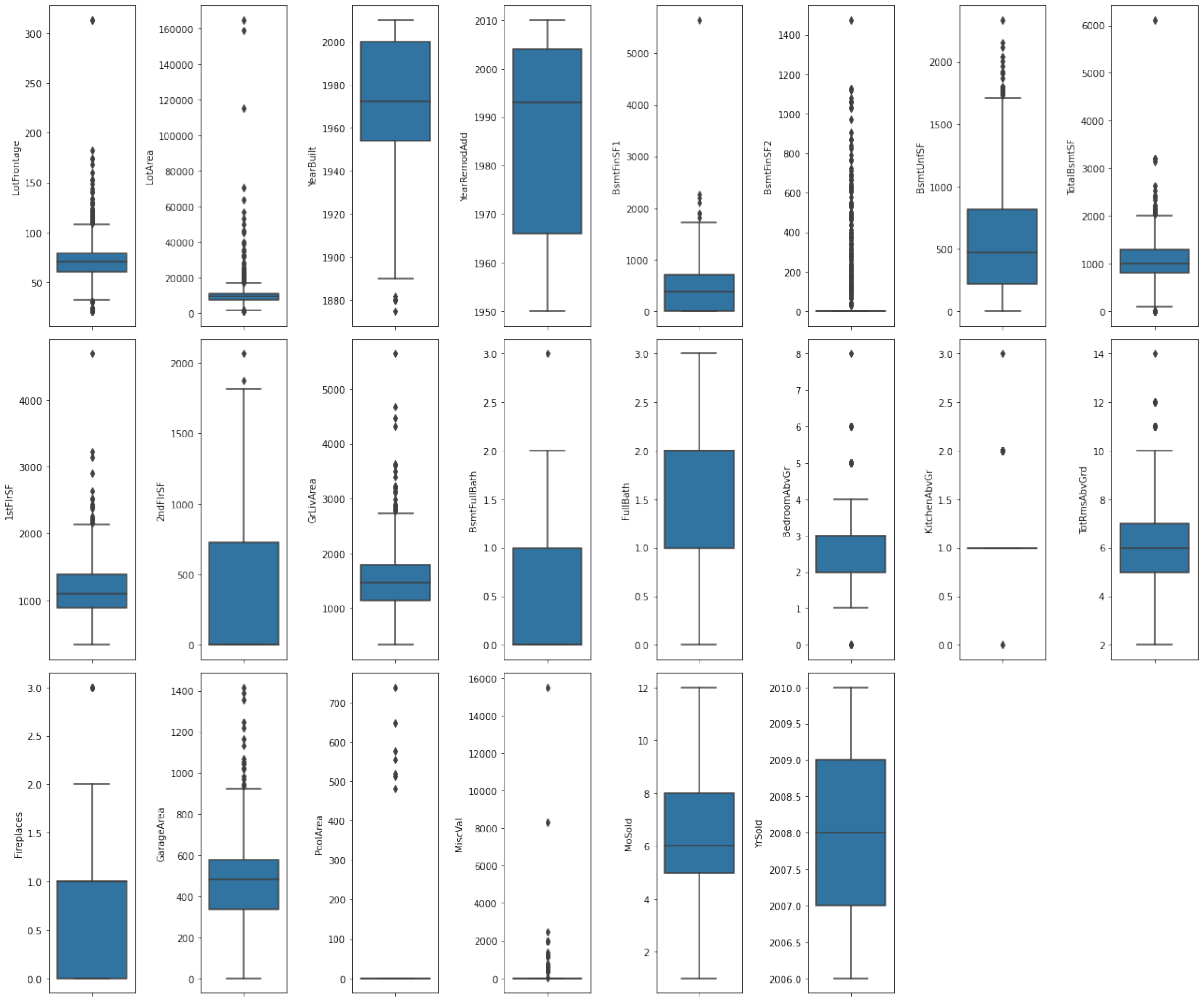
We can see that most of the properties are in low density residential areas. That means, most of the properties are having high prospect. Because since the area is low dense and residential area, most of the people will be attracted to the area. So, there will be demand for these properties.



Distribution of Data in Numerical Columns

The numerical columns in the dataset are not having a normal distribution. So skewness is present in the data distribution.

Outliers

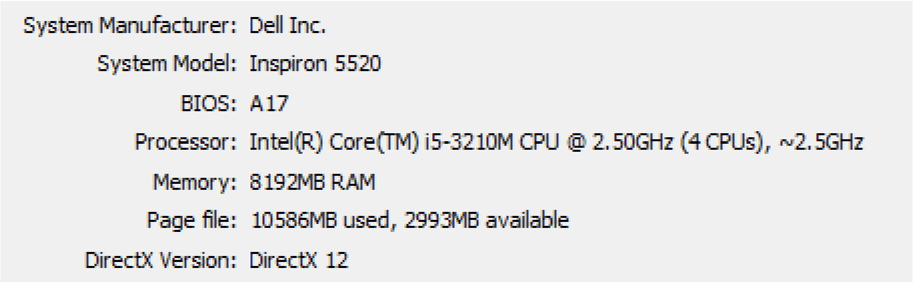


Presence of Outliers in Numerical Variables

From the observations, except the columns ['YearRemodAdd', 'FullBath', 'MoSold', 'YrSold'], rest of the columns are having outliers present in the train dataset.

Hardware and Software Requirements and Tools Used

Hardware Requirement:



System Manufacturer:	Dell Inc.
System Model:	Inspiron 5520
BIOS:	A17
Processor:	Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz (4 CPUs), ~2.5GHz
Memory:	8192MB RAM
Page file:	10586MB used, 2993MB available
DirectX Version:	DirectX 12

Hardware Configuration

Software Requirements:

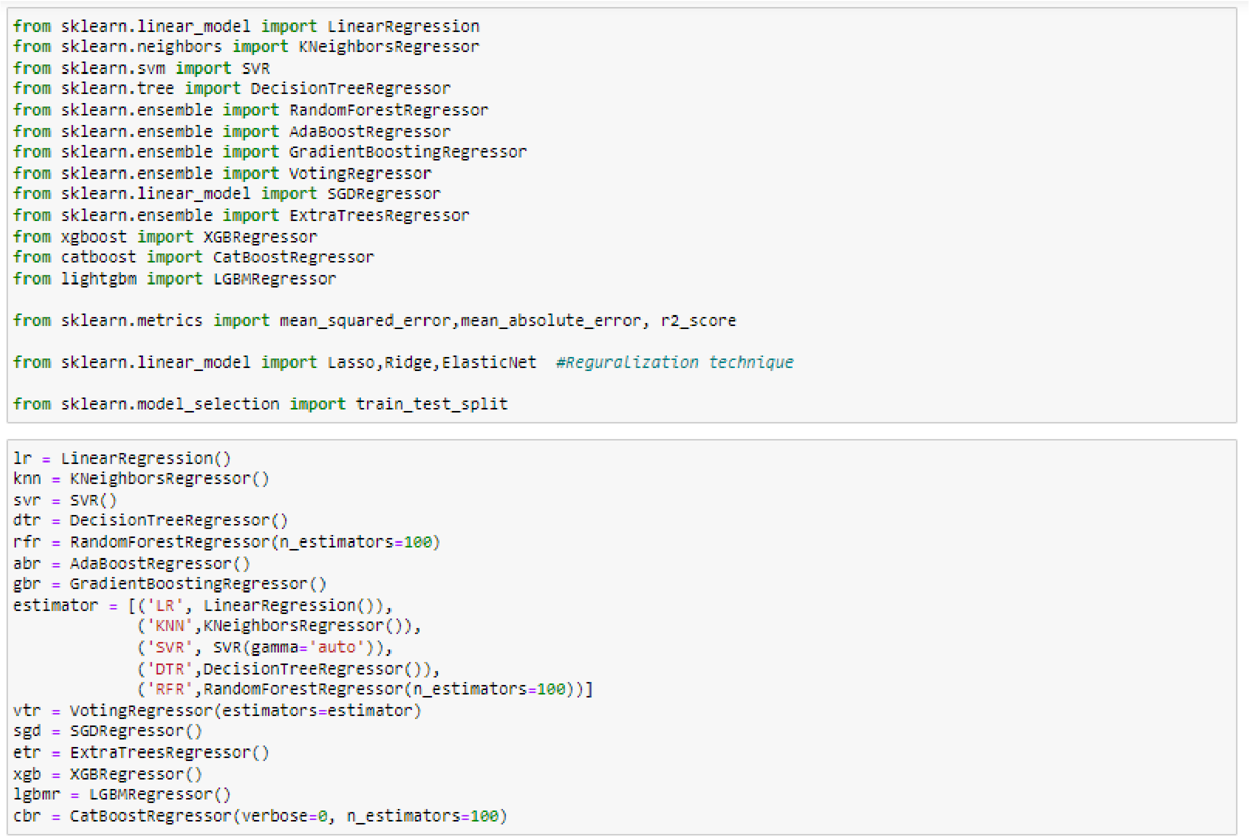
- Windows Version : Windows 10 Pro
- Anaconda Navigator : 2.0.3
  Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine.
- Jupyter Notebook : 6.3.0
  Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine.
- Python3 : Python 3.9.9
  Python3 is used as the base environment for performing the machine learning and data analysis.

Python Libraries Used:

- Pandas : Data manipulation and analysis
- Numpy : Adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
- Matplotlib, Seaborn : For visualization of variable relations and data distribution, and analysis.
- Sklearn : Simple and efficient tools for predictive data analysis.
- Scipy : SciPy provides algorithms for optimization, integration, interpolation, eigenvalue problems, algebraic equations, differential equations, statistics and many other classes of problems.
- Statsmodels : Provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration.
- Xgboost, catboost, lightgbm : Gradient boosting framework that uses tree-based learning algorithms.
- Pickle : Implements binary protocols for serializing and de-serializing

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods)



```
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import AdaBoostRegressor
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.ensemble import VotingRegressor
from sklearn.linear_model import SGDRegressor
from sklearn.ensemble import ExtraTreesRegressor
from xgboost import XGBRegressor
from catboost import CatBoostRegressor
from lightgbm import LGBMRegressor

from sklearn.metrics import mean_squared_error,mean_absolute_error, r2_score

from sklearn.linear_model import Lasso,Ridge,ElasticNet  #Reguralization technique

from sklearn.model_selection import train_test_split
```

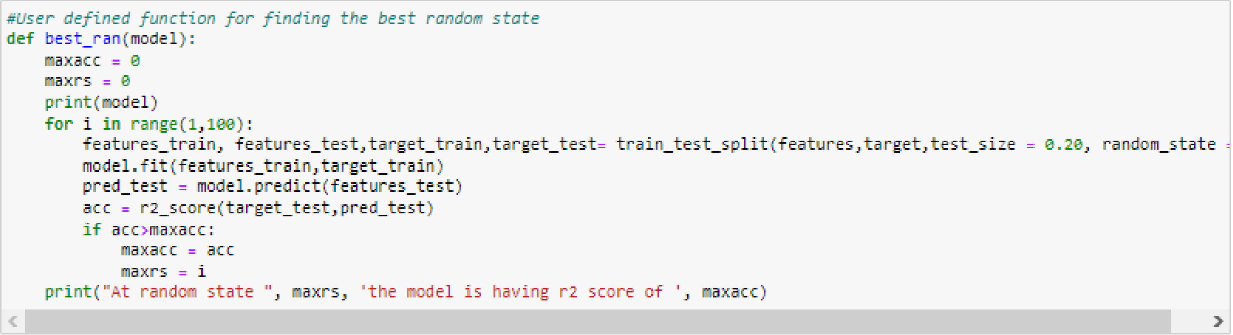
```
lr = LinearRegression()
knn = KNeighborsRegressor()
svr = SVR()
dtr = DecisionTreeRegressor()
rfr = RandomForestRegressor(n_estimators=100)
abr = AdaBoostRegressor()
gbr = GradientBoostingRegressor()
estimator = [('LR', LinearRegression()),
             ('KNN',KNeighborsRegressor()),
             ('SVR', SVR(gamma='auto')),
             ('DTR',DecisionTreeRegressor()),
             ('RFR',RandomForestRegressor(n_estimators=100))]
vtr = VotingRegressor(estimators=estimator)
sgd = SGDRegressor()
etr = ExtraTreesRegressor()
xgb = XGBRegressor()
lgbmr = LGBMRegressor()
cbr = CatBoostRegressor(verbose=0, n_estimators=100)
```

Imported the required models and created instances for the models

We created three functions for testing the model and for cross validations:

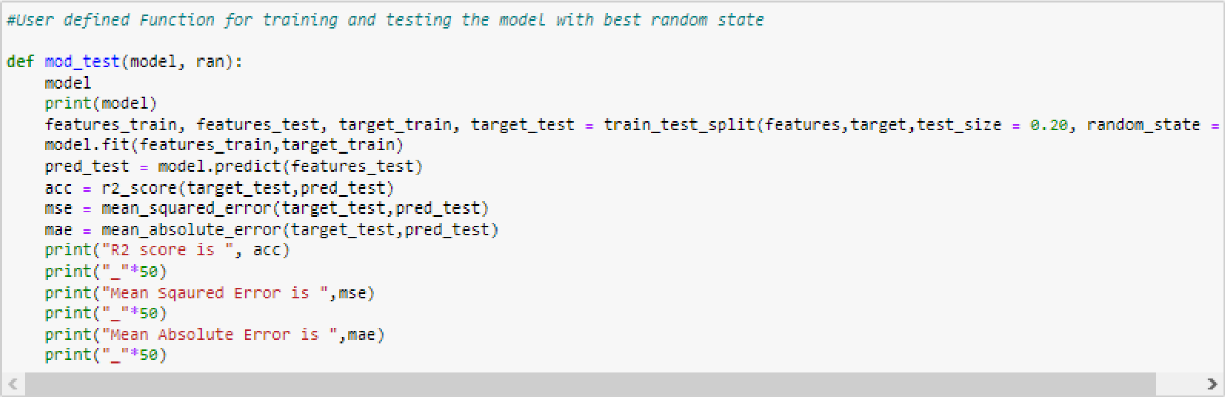
- best_ran : Finding the best random state for the selected model
- mod_test : Ttraining the model with the train data using the best random state.
- cross_val : Finding the best cross validation mean score for each model.

Testing of Identified Approaches (Algorithms) and evaluation of selected models



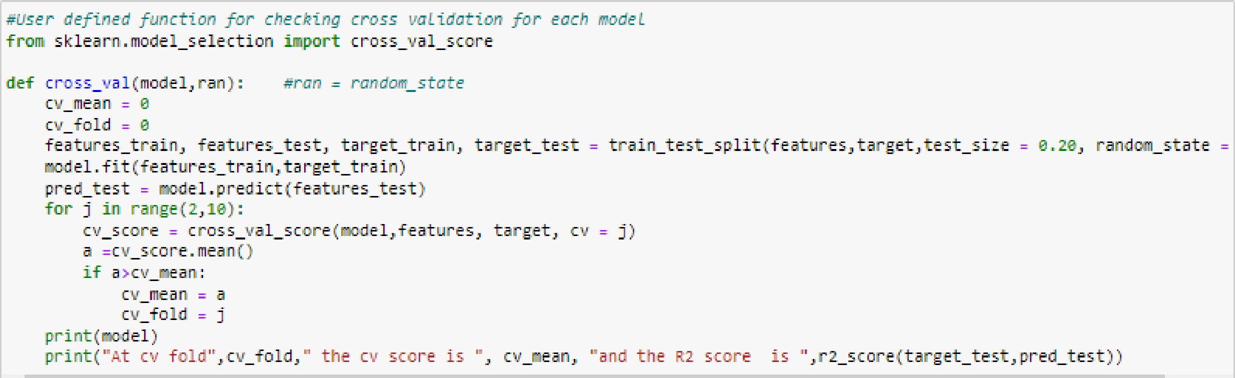
```
#User defined function for finding the best random state
def best_ran(model):
    maxacc = 0
    maxrs = 0
    print(model)
    for i in range(1,100):
        features_train, features_test,target_train,target_test= train_test_split(features,target,test_size = 0.20, random_state =
        model.fit(features_train,target_train)
        pred_test = model.predict(features_test)
        acc = r2_score(target_test,pred_test)
        if acc>maxacc:
            maxacc = acc
            maxrs = i
    print("At random state ", maxrs, 'the model is having r2 score of ', maxacc)
```

Code Snippet for function to find best random state



```
#User defined Function for training and testing the model with best random state

def mod_test(model, ran):
    model
    print(model)
    features_train, features_test, target_train, target_test = train_test_split(features,target,test_size = 0.20, random_state =
    model.fit(features_train,target_train)
    pred_test = model.predict(features_test)
    acc = r2_score(target_test,pred_test)
    mse = mean_squared_error(target_test,pred_test)
    mae = mean_absolute_error(target_test,pred_test)
    print("R2 score is ", acc)
    print("_"*50)
    print("Mean Sqaured Error is ",mse)
    print("_"*50)
    print("Mean Absolute Error is ",mae)
    print("_"*50)
```

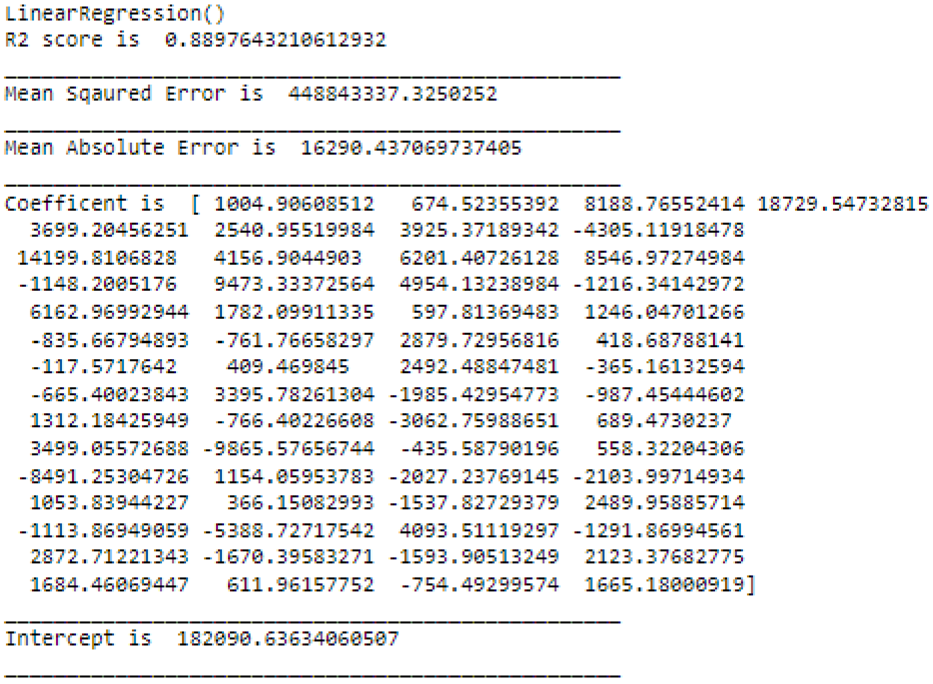
Code Snippet for function to test the model



```
#User defined function for checking cross validation for each model
from sklearn.model_selection import cross_val_score

def cross_val(model,ran):    #ran = random_state
    cv_mean = 0
    cv_fold = 0
    features_train, features_test, target_train, target_test = train_test_split(features,target,test_size = 0.20, random_state =
    model.fit(features_train,target_train)
    pred_test = model.predict(features_test)
    for j in range(2,10):
        cv_score = cross_val_score(model,features, target, cv = j)
        a =cv_score.mean()
        if a>cv_mean:
            cv_mean = a
            cv_fold = j
    print(model)
    print("At cv fold",cv_fold," the cv score is ", cv_mean, "and the R2 score  is ",r2_score(target_test,pred_test))
```

Code Snippet for function to find the cross validation mean score

- Linear Regression



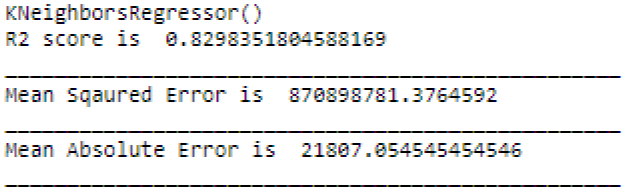
```
LinearRegression()
R2 score is  0.8897643210612932
__________________________________________________
Mean Sqaured Error is  448843337.3250252
__________________________________________________
Mean Absolute Error is  16290.437069737405
__________________________________________________
Coefficent is  [ 1004.90608512   674.52355392  8188.76552414 18729.54732815
  3699.20456251  2540.95519984  3925.37189342 -4305.11918478
 14199.8106828   4156.9044903   6201.40726128  8546.97274984
 -1148.2005176   9473.33372564  4954.13238984 -1216.34142972
  6162.96992944  1782.09911335   597.81369483  1246.04701266
  -835.66794893  -761.76658297  2879.72956816   418.68788141
  -117.5717642    409.469845    2492.48847481  -365.16132594
  -665.40023843  3395.78261304 -1985.42954773  -987.45444602
  1312.18425949  -766.40226608 -3062.75988651   689.4730237
  3499.05572688 -9865.57656744  -435.58790196   558.32204306
 -8491.25304726  1154.05953783 -2027.23769145 -2103.99714934
  1053.83944227   366.15082993 -1537.82729379  2489.95885714
 -1113.86949059 -5388.72717542  4093.51119297 -1291.86994561
  2872.71221343 -1670.39583271 -1593.90513249  2123.37682775
  1684.46069447   611.96157752  -754.49299574  1665.18000919]
__________________________________________________
Intercept is  182090.63634060507
__________________________________________________
```

Model Test Performance

LinearRegression()

At cv fold 4 the cv score is 0.841956481601463 and the R2 score is 0.8897643210612932

- KNeighborsRegressor



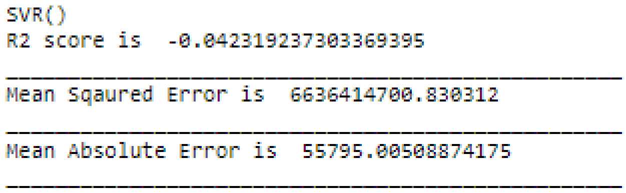
```
KNeighborsRegressor()
R2 score is  0.8298351804588169
__________________________________________________
Mean Sqaured Error is  870898781.3764592
__________________________________________________
Mean Absolute Error is  21807.054545454546
__________________________________________________
```

Model Test Performance

KNeighborsRegressor()

At cv fold 9 the cv score is 0.7671794540397973 and the R2 score is 0.8298351804588169

- SVR

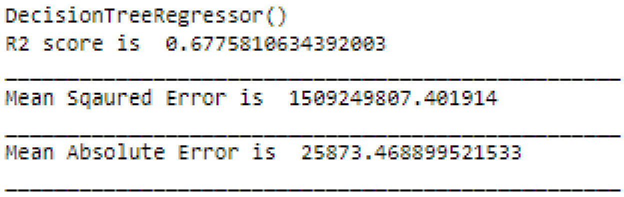


```
SVR()
R2 score is  -0.042319237303369395
_____________________________________________________
Mean Sqaured Error is  6636414700.830312
_____________________________________________________
Mean Absolute Error is  55795.00508874175
_____________________________________________________
```

Model Test Performance

SVR()
At cv fold 0 the cv score is 0 and the R2 score is -0.042319237303369395

- DecisionTree Regressor

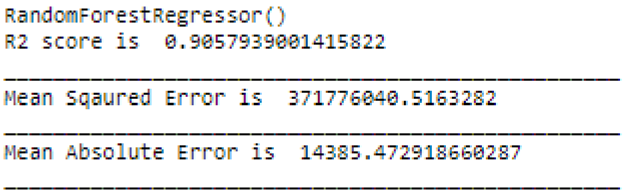


```
DecisionTreeRegressor()
R2 score is  0.6775810634392003
_____________________________________________________
Mean Sqaured Error is  1509249807.401914
_____________________________________________________
Mean Absolute Error is  25873.468899521533
_____________________________________________________
```

Model Test Performance

DecisionTreeRegressor()
At cv fold 5 the cv score is 0.6701646628382395 and the R2 score is 0.6746053292322107

- RandomForest Regressor

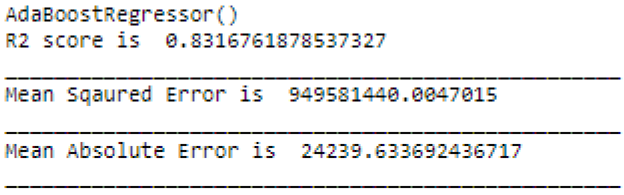


```
RandomForestRegressor()
R2 score is  0.9057939001415822
_____________________________________________________
Mean Sqaured Error is  371776040.5163282
_____________________________________________________
Mean Absolute Error is  14385.472918660287
_____________________________________________________
```

Model Test Performance

RandomForestRegressor()
At cv fold 9 the cv score is 0.843297454804246 and the R2 score is 0.8953098729820348

- AdaBoostRegressor

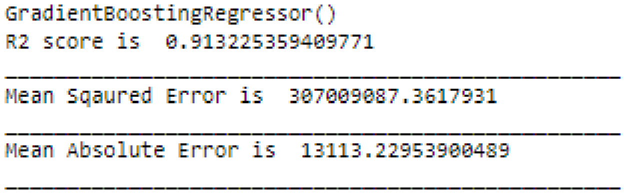


```
AdaBoostRegressor()
R2 score is  0.8316761878537327
__________________________________________________
Mean Sqaured Error is  949581440.0047015
__________________________________________________
Mean Absolute Error is  24239.633692436717
__________________________________________________
```

Model Test Performance

AdaBoostRegressor()
At cv fold 2 the cv score is 0.7855981890265346 and the R2 score is 0.8233835066834014

- GradientBoostingRegressor

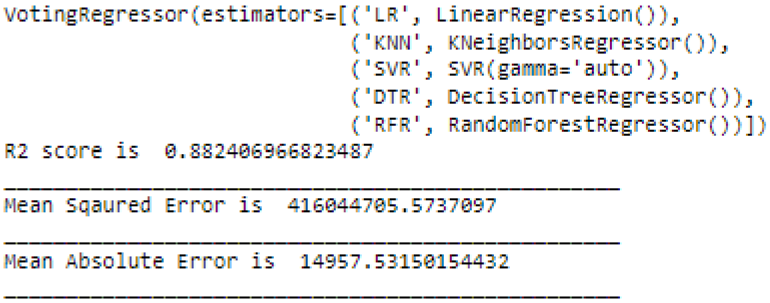


```
GradientBoostingRegressor()
R2 score is  0.913225359409771
__________________________________________________
Mean Sqaured Error is  307009087.3617931
__________________________________________________
Mean Absolute Error is  13113.22953900489
__________________________________________________
```

Model Test Performance

GradientBoostingRegressor()
At cv fold 9 the cv score is 0.8707080769738818 and the R2 score is 0.9111654918760981

- VotingRegressor

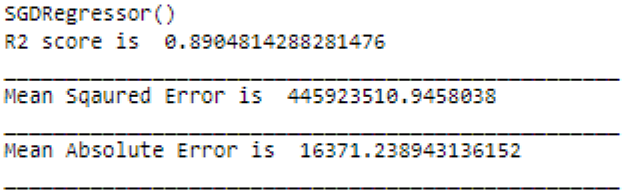


```
VotingRegressor(estimators=[('LR', LinearRegression()),
                            ('KNN', KNeighborsRegressor()),
                            ('SVR', SVR(gamma='auto')),
                            ('DTR', DecisionTreeRegressor()),
                            ('RFR', RandomForestRegressor())])
R2 score is  0.882406966823487
__________________________________________________
Mean Sqaured Error is  416044705.5737097
__________________________________________________
Mean Absolute Error is  14957.53150154432
__________________________________________________
```

Model Test Performance

VotingRegressor()
At cv fold 9 the cv score is 0.7954568264992761 and the R2 score is 0.867350264164061

- SGDRegressor

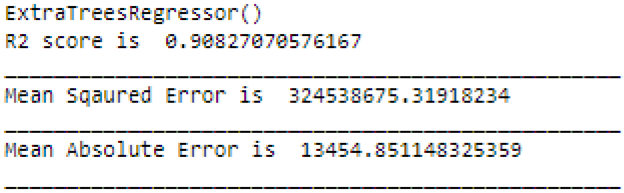


```
SGDRegressor()
R2 score is  0.8904814288281476
____________________________________________________
Mean Sqaured Error is  445923510.9458038
____________________________________________________
Mean Absolute Error is  16371.238943136152
____________________________________________________
```

Model Test Performance

SGDRegressor()
At cv fold 8 the cv score is 0.8423369543752157 and the R2 score is 0.8882865363787905

- ExtraTreesRegressor

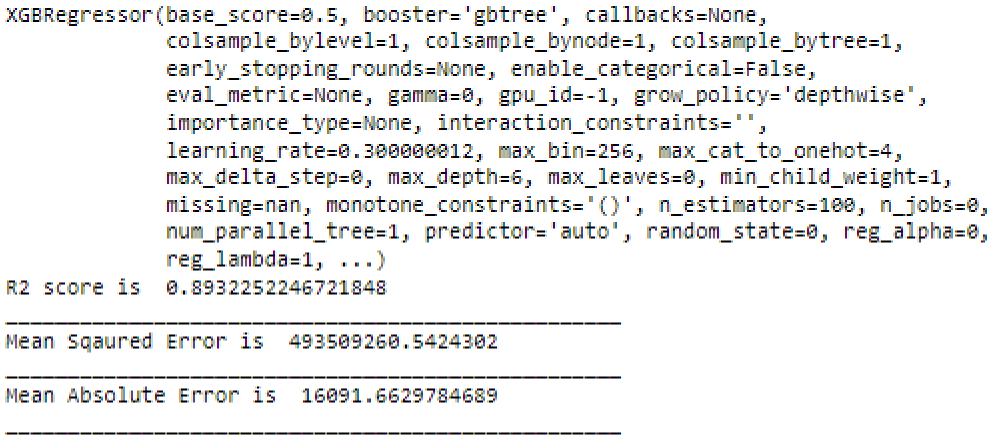


```
ExtraTreesRegressor()
R2 score is  0.90827070576167
____________________________________________________
Mean Sqaured Error is  324538675.31918234
____________________________________________________
Mean Absolute Error is  13454.851148325359
____________________________________________________
```

Model Test Performance

ExtraTreesRegressor()
At cv fold 8 the cv score is 0.8399658063380535 and the R2 score is 0.9067967751346855

- XGBRegressor



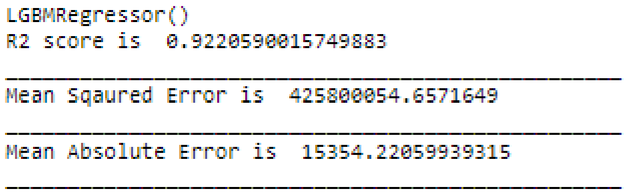
```
XGBRegressor(base_score=0.5, booster='gbtree', callbacks=None,
             colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
             early_stopping_rounds=None, enable_categorical=False,
             eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
             importance_type=None, interaction_constraints='',
             learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
             max_delta_step=0, max_depth=6, max_leaves=0, min_child_weight=1,
             missing=nan, monotone_constraints='()', n_estimators=100, n_jobs=0,
             num_parallel_tree=1, predictor='auto', random_state=0, reg_alpha=0,
             reg_lambda=1, ...)
R2 score is  0.8932252246721848
__________________________________________________
Mean Sqaured Error is  493509260.5424302
__________________________________________________
Mean Absolute Error is  16091.6629784689
__________________________________________________
```

Model Test Performance

XGBRegressor()

At cv fold 9 the cv score is 0.8533124887375951 and the R2 score is 0.8932252246721848

- LGBMRegressor



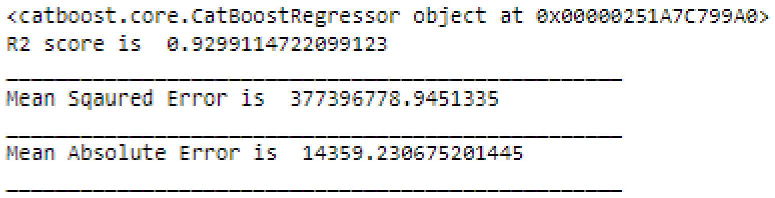
```
LGBMRegressor()
R2 score is  0.9220590015749883
__________________________________________________
Mean Sqaured Error is  425800054.6571649
__________________________________________________
Mean Absolute Error is  15354.22059939315
__________________________________________________
```

Model Test Performance

LGBMRegressor()

At cv fold 4 the cv score is 0.8706445896499 and the R2 score is 0.9220590015749883

- CatBoostRegressor



```
<catboost.core.CatBoostRegressor object at 0x00000251A7C799A0>
R2 score is  0.9299114722099123
__________________________________________________
Mean Sqaured Error is  377396778.9451335
__________________________________________________
Mean Absolute Error is  14359.230675201445
__________________________________________________
```

Model Test Performance

CatBoostRegressor()

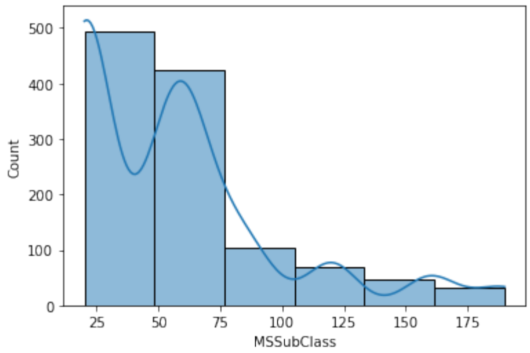
At cv fold 4 the cv score is 0.8760469431105571 and the R2 score is 0.9299114722099123

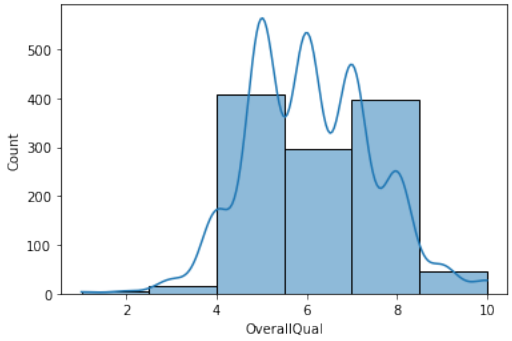
Key Metrics for success in solving problem under consideration

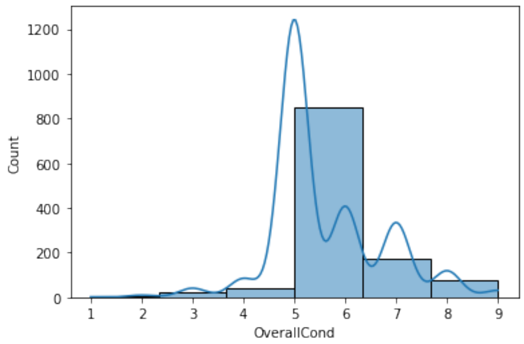
- R2 Score
  The proportion of the variance in the dependent variable that is predictable from the independent variable(s). It is the one of the key metrics for analyzing the performance of regression models.
- Mean Squared Error
  It is the average of the square of the errors. The larger the number the larger the error. **Error** means the difference between the observed value and predicted values.
- Mean Absolute Error
  It is the average of difference between the measured value and “true” value.
- Regularization
  Regularization is a technique used to reduce the errors by fitting the function appropriately on the given training set and avoid overfitting.

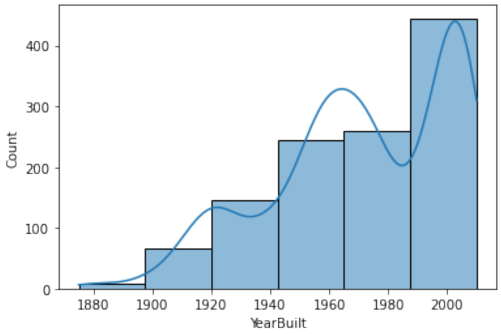
Visualization

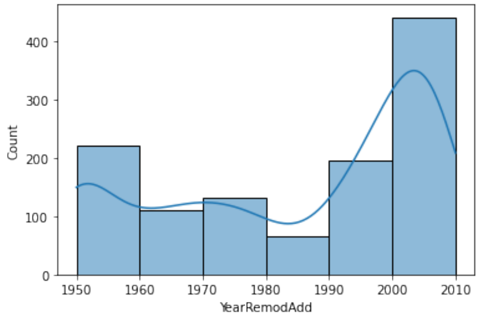
Univariate Analysis

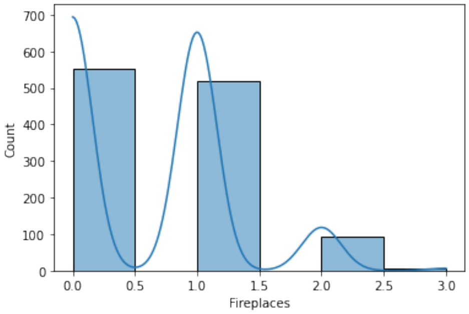


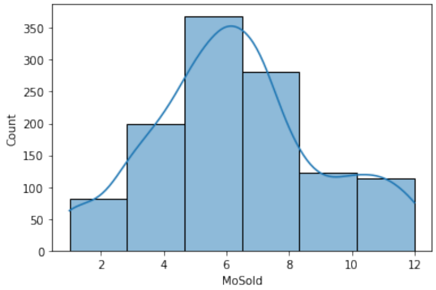


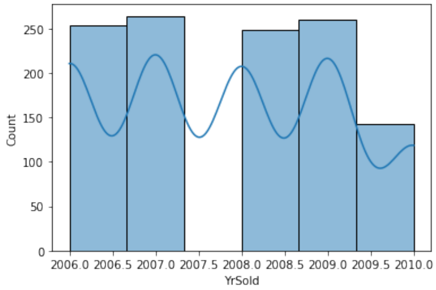








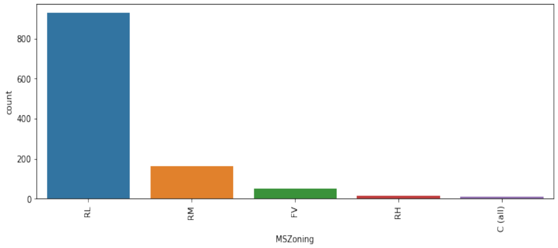


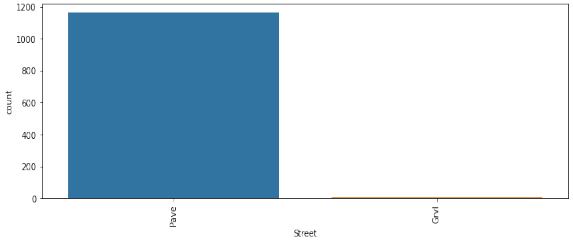


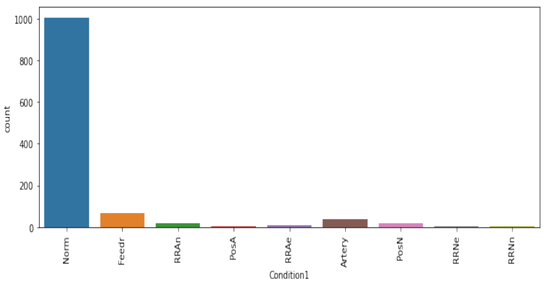
Observations:

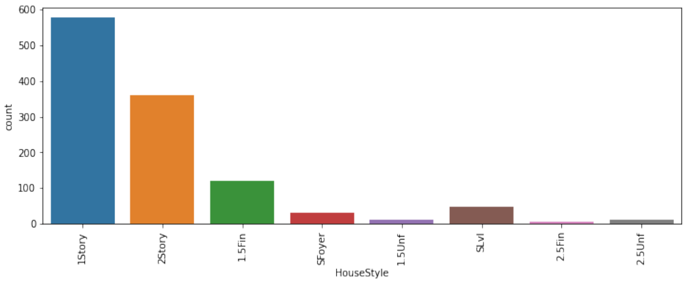
- Most of the house properties are 1 story or 1 1/2 story houses.
- Most of the house properties have a rating of 4-8 out of 10 for overall quality which is a good sign for a prospect house.
- Most of the house properties have a rating of 5-6 for overall condition of the house. This seems a good fit for a potential house property.
- Most of the houses were built between the year 1987-2010 which seems good. Because most of the houses are not too old.
- Most of the houses have been remodeled between the year 2000-2010. That means most of the house properties are not too old and can be good for living for furthermore years.
- Most of the houses are having zero or 1 fireplace facility.
- Most of the house properties are sold on from May to August
- Most of the houses were sold between 2006-2007 and between 2008-2009.

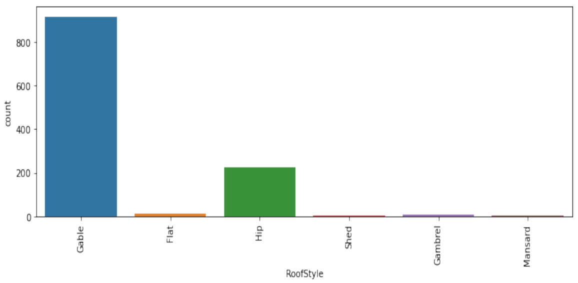
Categorical Variables

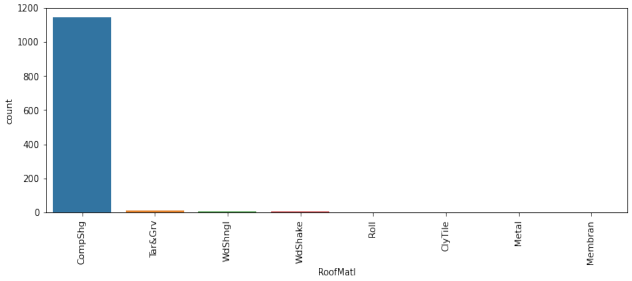


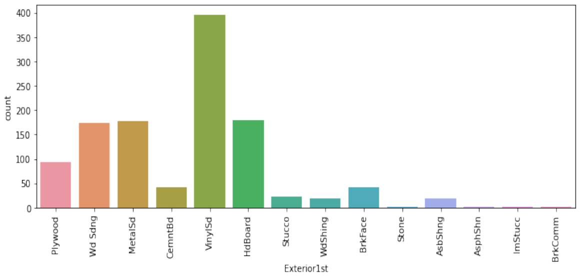


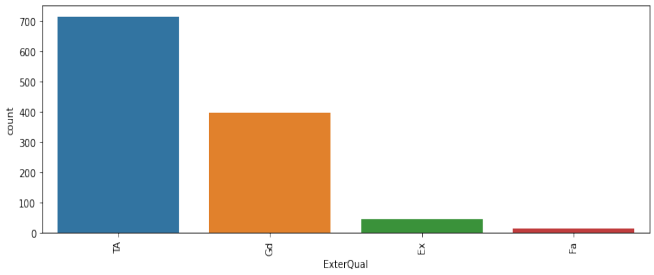

count
1000
800
600
400
200
0
Norm
Feedr
RRAn
PosA
RRAe
Artery
PosN
RRNe
RRNn
Condition1

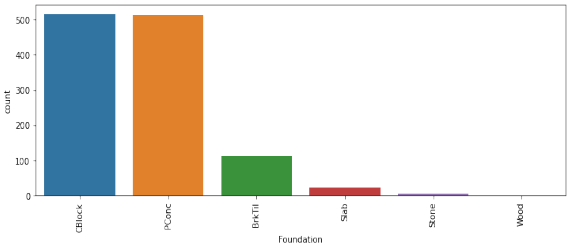

count
600
500
400
300
200
100
0
1Story
2Story
1.5Fin
SFoyer
1.5Unf
SLvl
2.5Fin
2.5Unf
HouseStyle

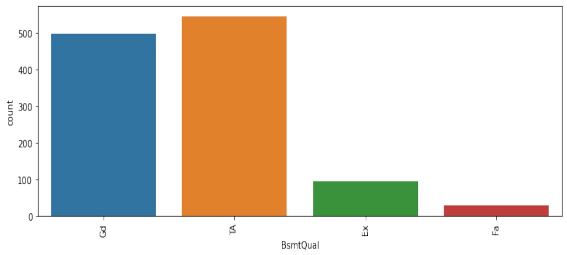

count
800
600
400
200
0
Gable
Flat
Hip
Shed
Gambrel
Mansard
RoofStyle

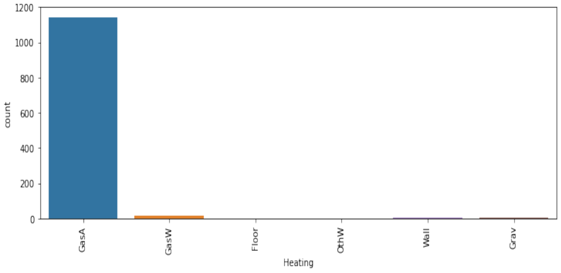

count
1200
1000
800
600
400
200
0
CompShg
Tar&Grv
WdShngl
WdShake
Roll
ClyTile
Metal
Membran
RoofMatl

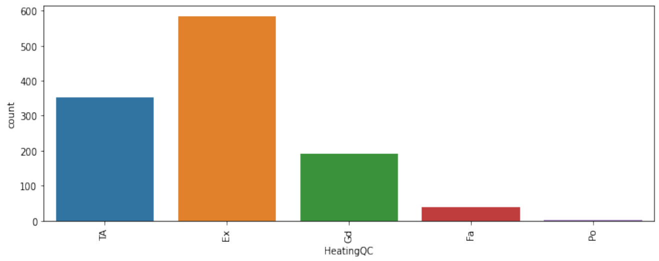

count
400
350
300
250
200
150
100
50
0
Plywood
Wd Sdng
MetalSd
CemntBd
VinylSd
HdBoard
Stucco
WdShing
BrkFace
Stone
AsbShng
AsphShn
ImStucc
BrkComm
Exterior1st

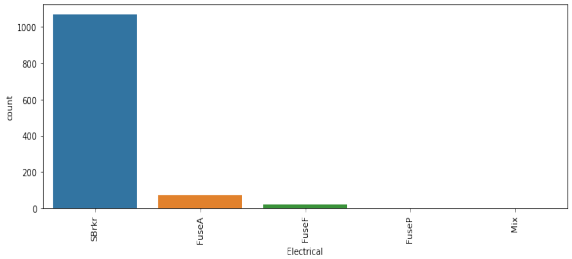

count
700
600
500
400
300
200
100
0
TA
Gd
Ex
Fa
ExterQual

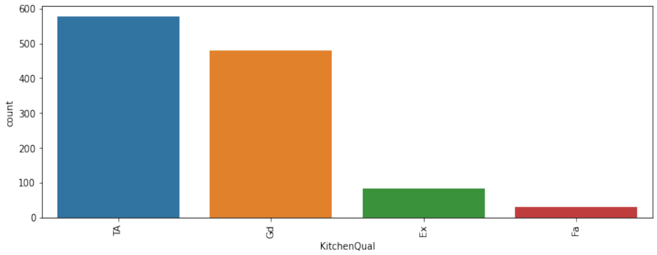

count
500
400
300
200
100
0
CBlock
PConc
BrkTil
Slab
Stone
Wood
Foundation

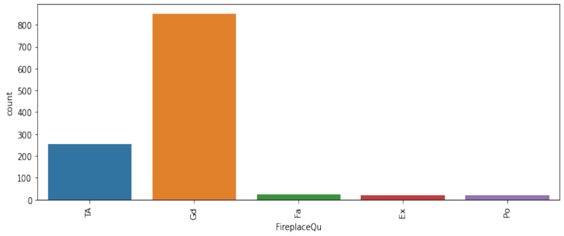

count
500
400
300
200
100
0
Gd
TA
Ex
Fa
BsmtQual

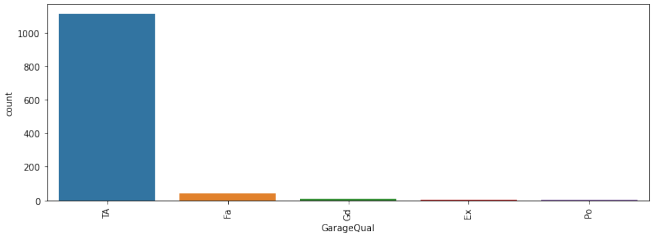

count
1200
1000
800
600
400
200
0
GasA
GasW
Floor
OthW
Wall
Grav
Heating

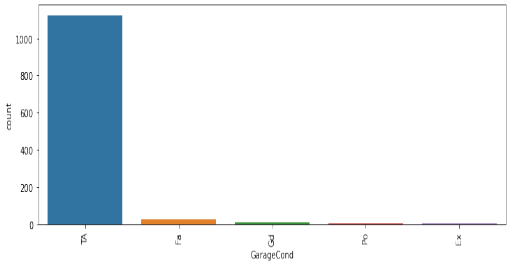

count
600
500
400
300
200
100
0
TA
Ex
Gd
Fa
Po
HeatingQC

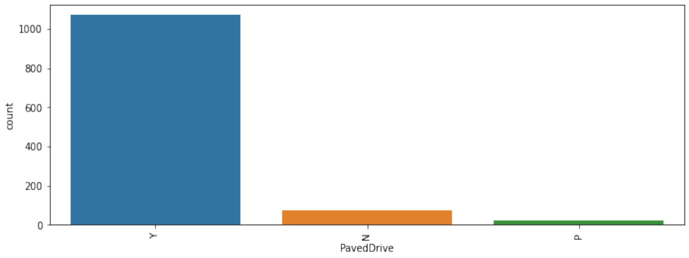


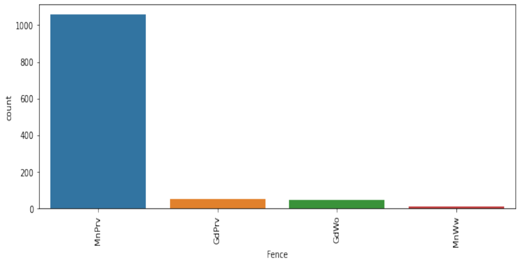


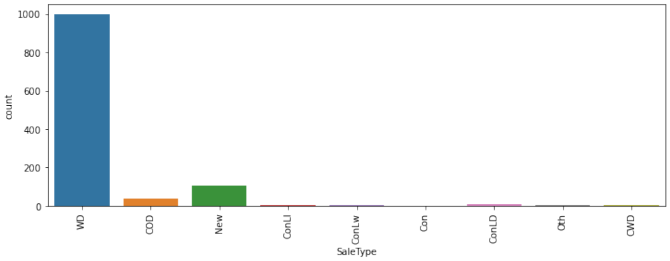


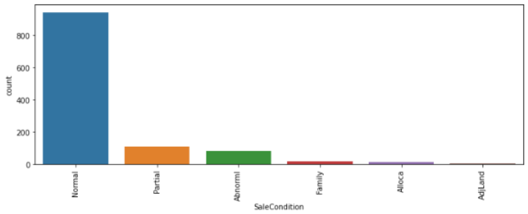












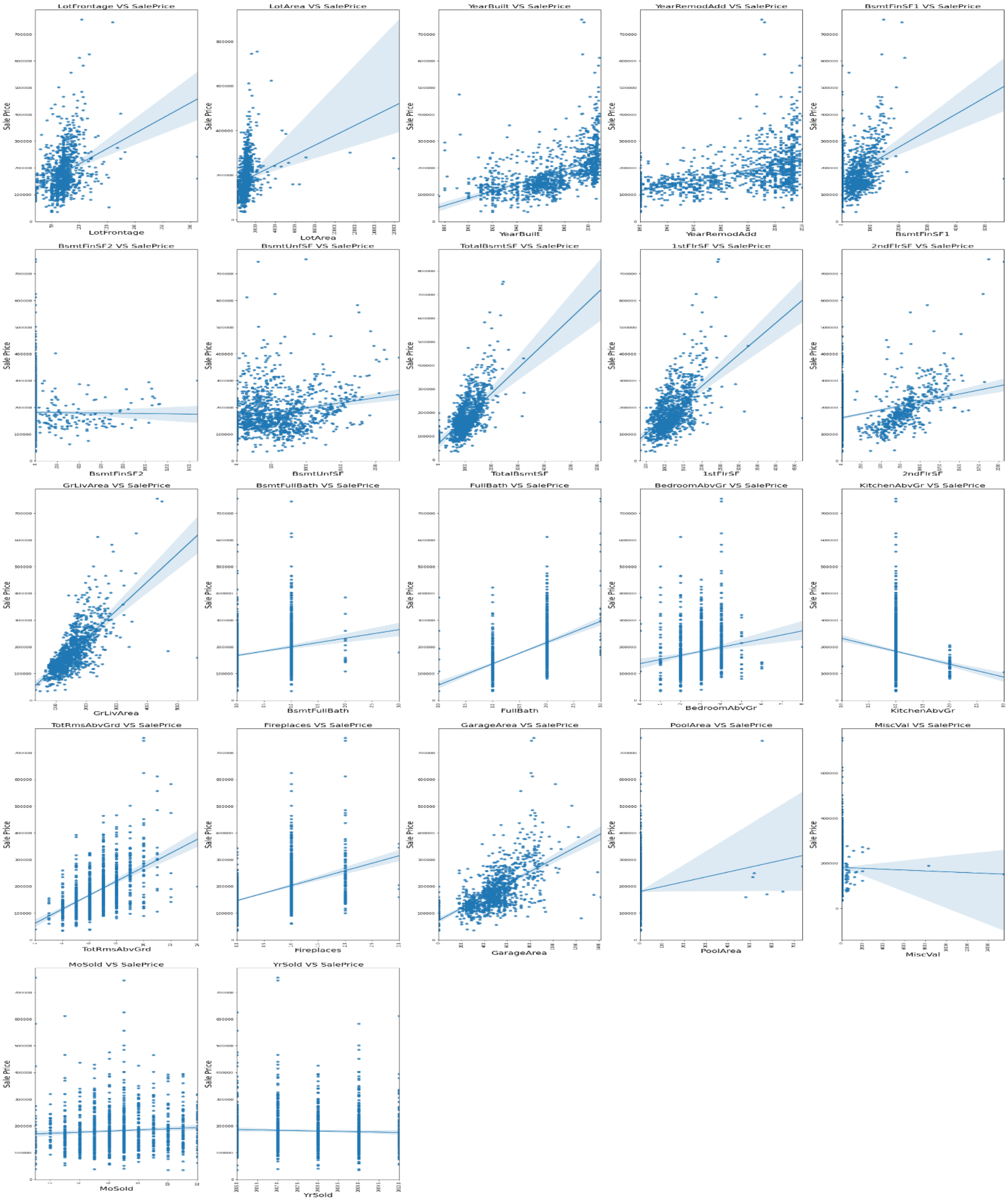
Observations:

- We can see that most of the properties are in low density residential areas. That means, most of the properties are having high prospect. Because since the area is low densed and residential area, most of the people will be attracted to the area. So, there will be demand for these properties.
- Most of the houses have paved road access to the property.
- Most of the house properties are not close to any conditions like railroad, arterial street or feeder street etc. Most of the properties are residing in the normal areas.

- Most of the properties are 1 story building which is suitable for 1 family.
- Most of the house properties are having gable roof style.
- Most of the houses were having roof tops made of standard(composite) shingle.
- From the above observations, most of the properties are having Vinyl Siding for exterior covering of the house.
- Most of the houses have been rated as avearge or typical in quality of exterior material.
- Most of the houses were having Cinder Block or Poured Concrete as material for foundation.
- Most of the houses are having 80 to 99 inches basement height. Which are good and typical
- Most of the houses are having gas forced warm air furnace for heating.
- Most of the houses are having excellent or typical heating quality and condition.
- Most of the houses are having Standard Circuit Breaker and Romex electrical system.
- Most of the houses are having good and typical or average quality rating for kitchen.
- Most of the houses are having Good - Masonry Fireplace in main level for the fireplace quality.
- Most of the houses are having typical or average rating for garage available in the house property.
- Most of the houses are having typical or average rating for the condition of the garage available in the house property.
- Most of the house properties are having paved house driveway.
- Most of the hosues are getting minimum privacy from the quality of the fence on the property.
- Most of the house properties are sold on warranty deed(Conventional)
- Most of the house properties were normally sold.

Multivariate Analysis

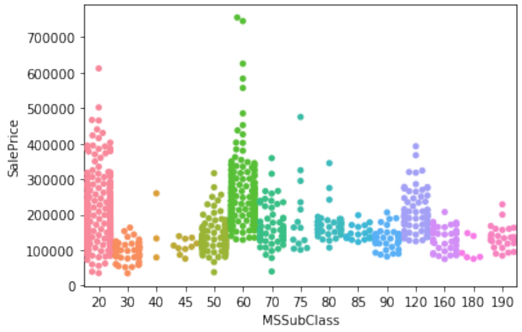
Plotting the relation of numerical variables with target variable

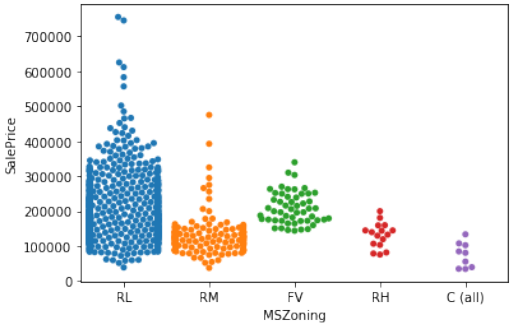


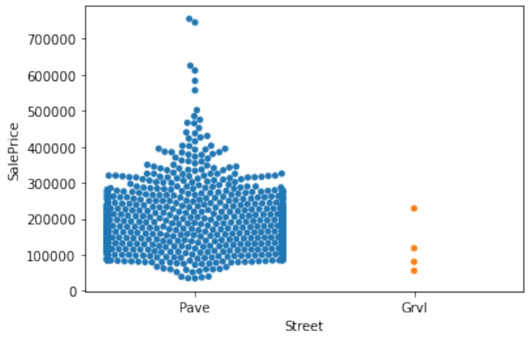
Observations:

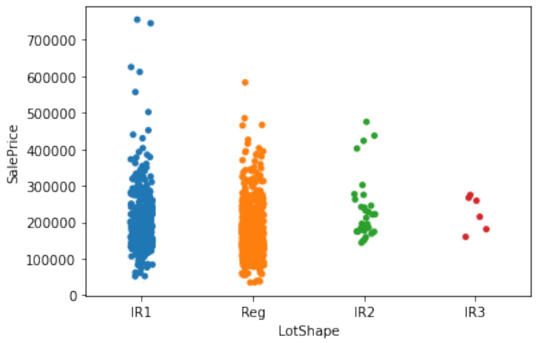
- It is visible that, as the distance from the street connected to the property increases, the sale price is also increasing.
- The house sale price is increasing as the lot area of the propoerties increases.
- Houses that were built recently are having higher sale price. The sale price of the house is reducing for older house properties.
- Most of the houses which were recently remodeled are having slightly higher sale price.
- The price of house is increasing as the square feet of type 1 finished basement of house increases.
- The price of house is decreasing as the square feet of type 2 finished basement of house increases.
- The price of house is increasing as the square feet of unfinished basement is increasing.
- The price of house is increasing as the total square feet of the basement of house is increasing.
- It shows that the price of house is increasing as the first floor and second floor square feet is increased.
- The price of house is high for houses with larger above grade living area.
- Houses with 1 basement full bathroom is sold more but price of house increases as there is two full bathrooms in basements
- The house of price is increasing as the number of above grade full bathroom increases.
- The price of house is increasing as there are more number of above grade bedrooms and total rooms above grade available in the house.
- The price of house is decreasing as the number of above grade kitchens are increased in the house.
- The house price is increasing as the number of fireplaces in house is increased.
- The house price is higher for houses with higher garage area and/or pool area.
- The price of house is reducing as $value of miscellaneous features is increased.
- Most of the houses were sold in the month between May - August and in the year 2009.

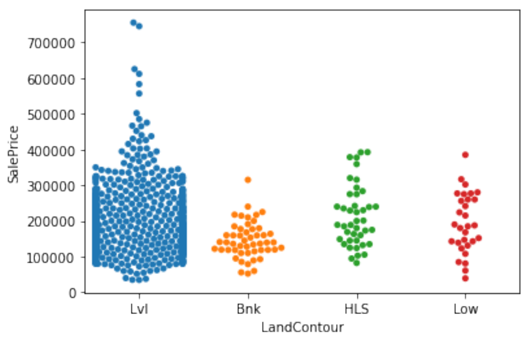
Plotting the relation of categorical variable with the target variable

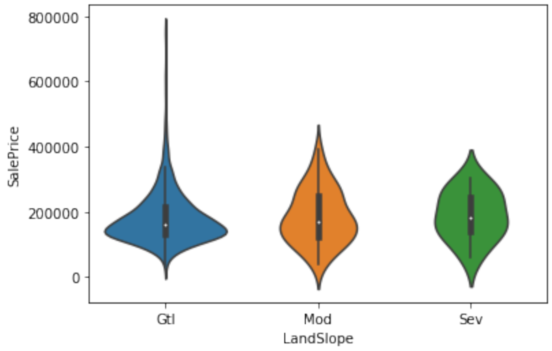


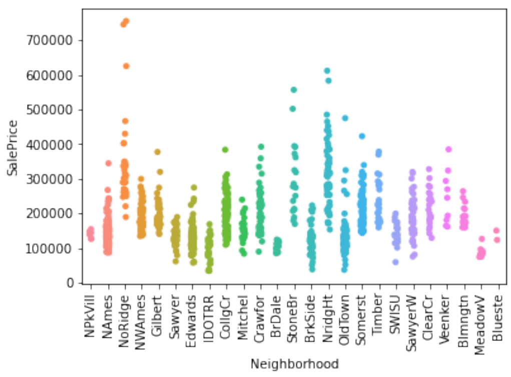


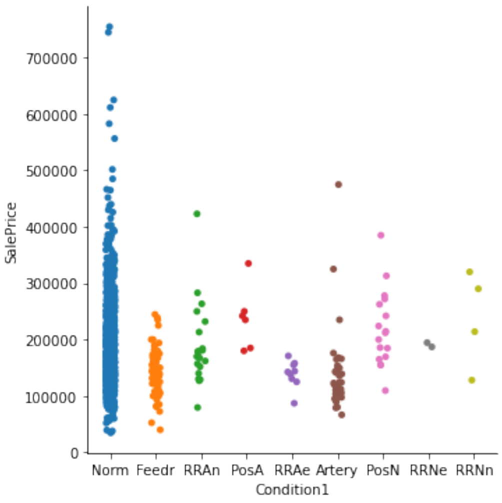

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
Pave
Grvl
Street

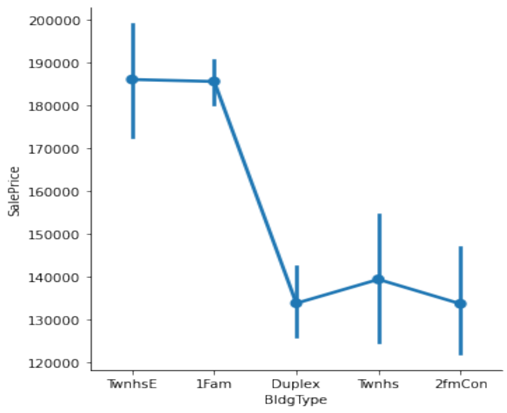

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
IR1
Reg
IR2
IR3
LotShape

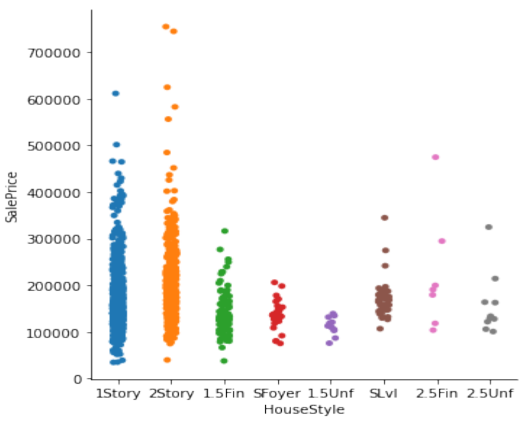

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
Lvl
Bnk
HLS
Low
LandContour

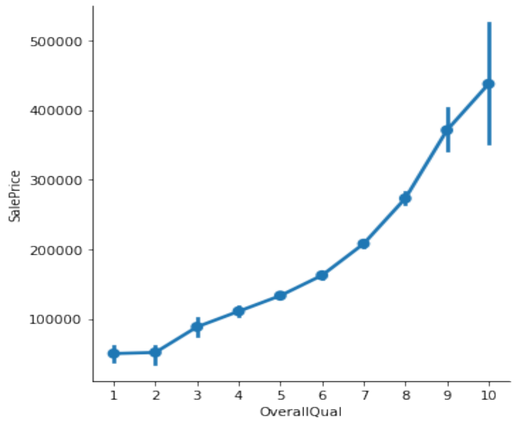

SalePrice
800000
600000
400000
200000
0
Gtl
Mod
Sev
LandSlope

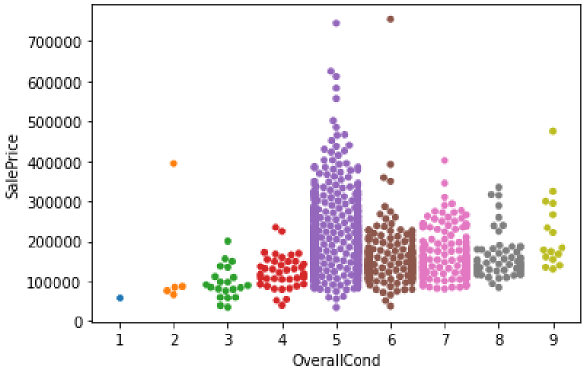

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
NPkVill
NAmes
NoRidge
NWAmes
Gilbert
Sawyer
Edwards
IDOTRR
CollgCr
Mitchel
Crawfor
BrDale
StoneBr
BrkSide
NridgHt
OldTown
Somerst
Timber
SWISU
SawyerW
ClearCr
Veenker
Blmngtn
MeadowV
Blueste
Neighborhood

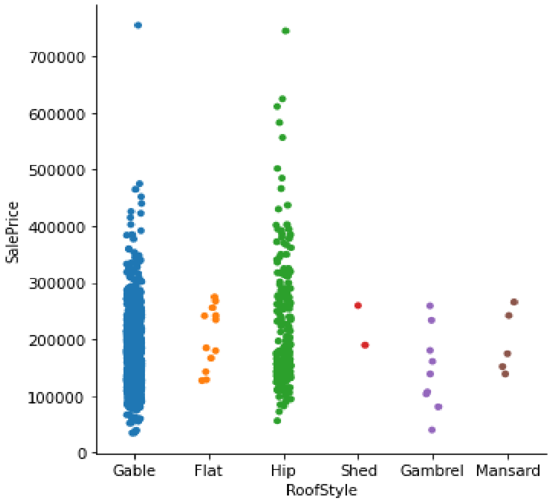

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
Norm
Feedr
RRAn
PosA
RRAe
Artery
PosN
RRNe
RRNn
Condition1

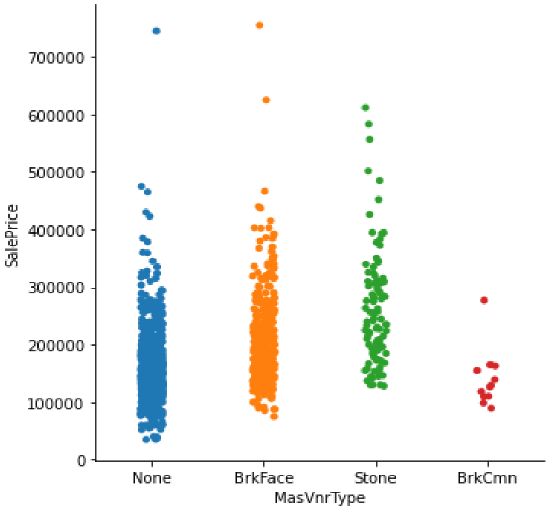

SalePrice
200000
190000
180000
170000
160000
150000
140000
130000
120000
TwnhsE
1Fam
Duplex
Twnhs
2fmCon
BldgType

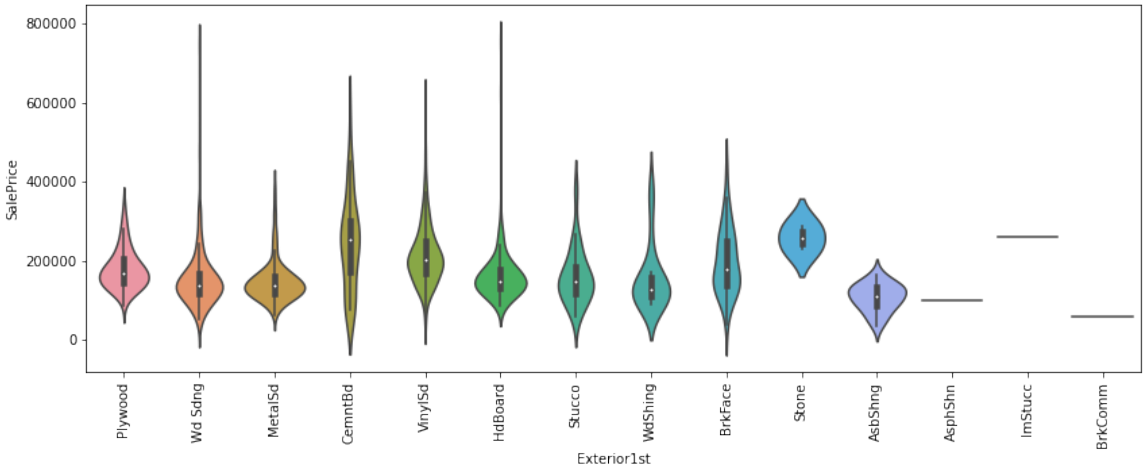

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
1Story
2Story
1.5Fin
SFoyer
1.5Unf
SLvl
2.5Fin
2.5Unf
HouseStyle

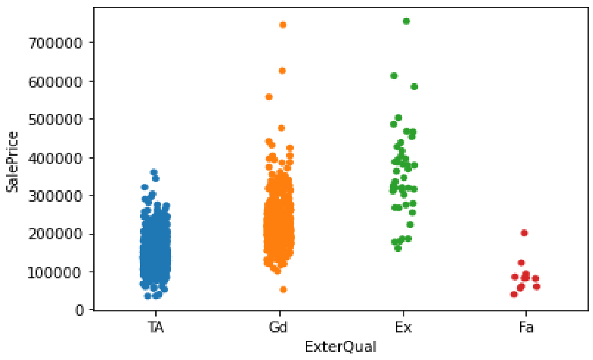

SalePrice
500000
400000
300000
200000
100000
1
2
3
4
5
6
7
8
9
10
OverallQual

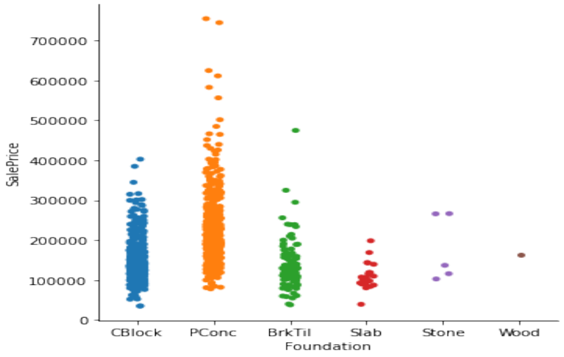

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
1
2
3
4
5
6
7
8
9
OverallCond

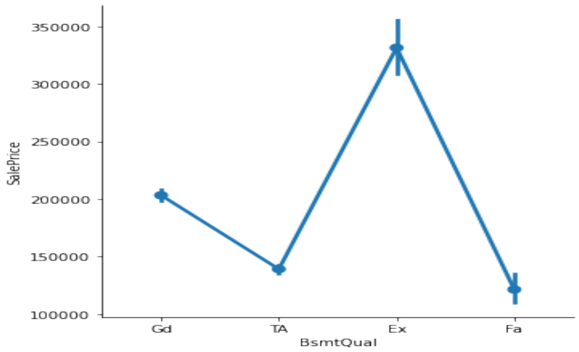

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
Gable
Flat
Hip
Shed
Gambrel
Mansard
RoofStyle

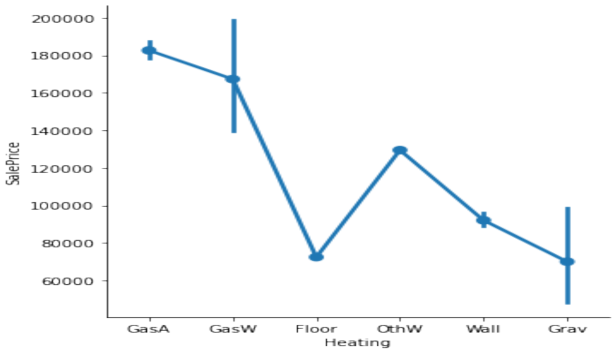

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
None
BrkFace
Stone
BrkCmn
MasVnrType

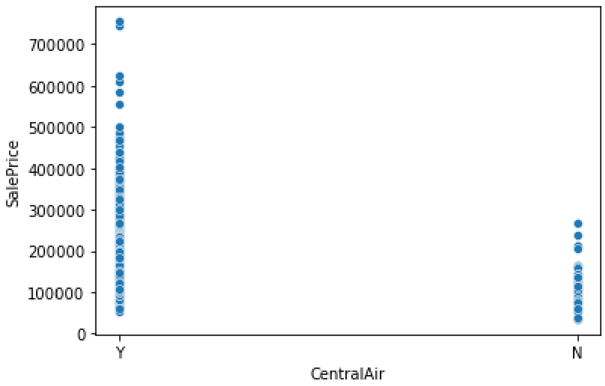

SalePrice
800000
600000
400000
200000
0
Plywood
Wd Sdng
MetalSd
CemntBd
VinylSd
HdBoard
Stucco
WdShing
BrkFace
Stone
AsbShng
AsphShn
ImStucc
BrkComm
Exterior1st

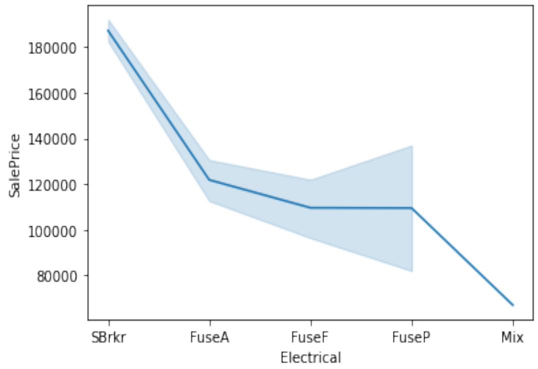

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
TA
Gd
Ex
Fa
ExterQual

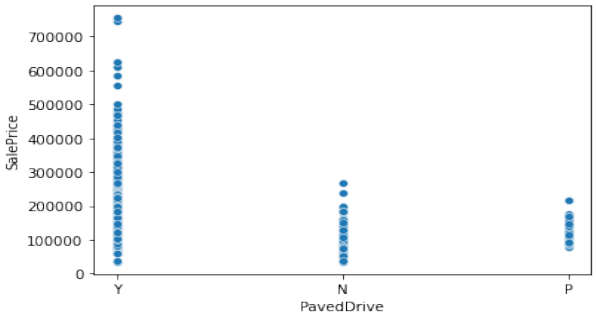

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
CBlock
PConc
BrkTil
Slab
Stone
Wood
Foundation

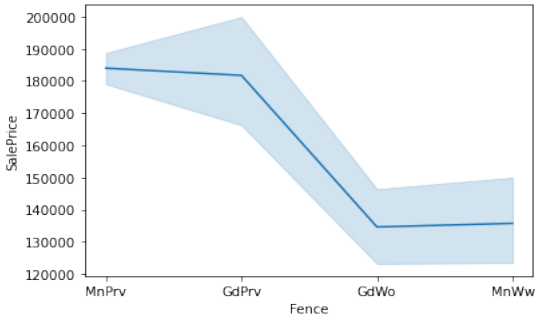

SalePrice
100000
150000
200000
250000
300000
350000
Gd
TA
Ex
Fa
BsmtQual

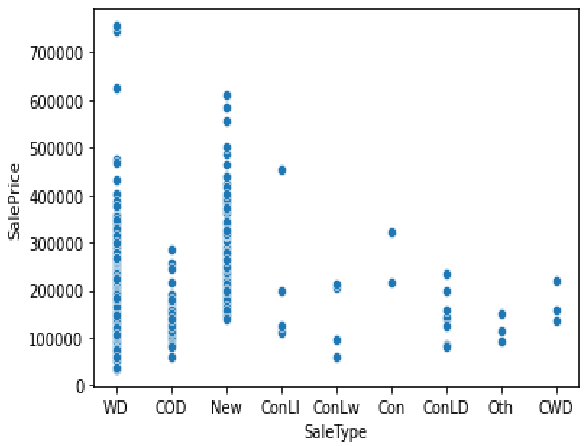

SalePrice
60000
80000
100000
120000
140000
160000
180000
200000
GasA
GasW
Floor
OthW
Wall
Grav
Heating

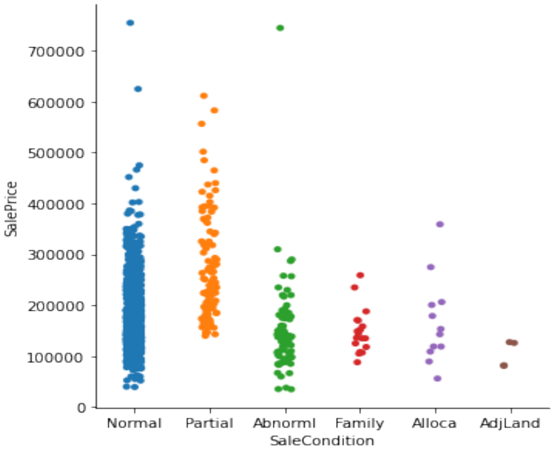

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
Y
N
CentralAir


SalePrice
80000
100000
120000
140000
160000
180000
SBrkr
FuseA
FuseF
FuseP
Mix
Electrical


SalePrice
0
100000
200000
300000
400000
500000
600000
700000
Y
N
P
PavedDrive


SalePrice
120000
130000
140000
150000
160000
170000
180000
190000
200000
MnPrv
GdPrv
GdWo
MnWw
Fence





Observations:

- Most of the houses having 2-STORY 1946 & NEWER as dwelling type were sold for higher price. Also 1-STORY 1946 & NEWER ALL STYLES dwelling type houses were most sold.
- The house properties residing in residential and low densed area were sold for higher price and mostly sold compared to houses residing in other zoing classifications.
- From the above observations, most of the houses sold are having paved type of road access to the property.
- From the above observations, slightly irregular shaped property were sold for higher price.
- Most of the houses sold were having flat or nearly flat property. The price for these types of houses were also comparitively high.
- Most of the houses sold and which were having higher sale price were gentle slopped property.
- Most of the houses in Northridge and Northridge heights are sold for higher price.
- Most of the houses which are in normal conditions and which are not in proximity to arterial street, feeder street or railrods etc are sold for higher price.
- Most of the houses having single family detached or town house end unit as dwelling type are sold for higher price compared to other type of dwelling houses.
- Most of the houses sold were one or two story dwelling style. These were also the categories having higher sale price houses.
- The house price is increasing for the properties with high rating for overall material and finish of the house.
- Most of the properties sold were having 5-6 as rating for the overall condition of the house. Also houses with rating 5 were sold for higher prices compared to other rating houses.
- Most of houses having gable or hip roof type were sold more compared to other roof type houses. Houses with these type of roofs were also sold for higher prices.
- The price of house is higher for houses with exterior covering of wood siding and hardboard.
- Houses with Stone as Masonry veneer type were priced higher.
- Most of the houses which were rated as excellent in quality of exterial material are having high sale price.
- Most of the houses with poured concrete as foundation are having higher sale volume and higher sale price.
- Houses with 100+ inches basement height are having higher sale price.
- The house price is higher for houses with heating type as Gas forced warm air furnace.

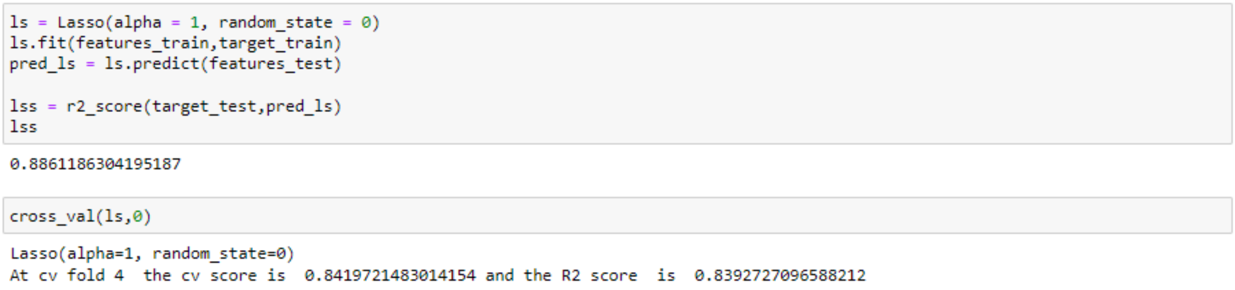
- The sale price is higher for houses with central air conditioning.
- Most of the houses sold and were having higher sale price is for houses with Standard Circuit Breakers & Romex as electrical system.
- Most of the houses that are having paved driveway are having higher sale price and are mostly sold.
- Most of the houses with minimum fence quality privacy are having higher sale price.
- Most of the houses sold with warranty deed(Conventional) or sold after construction were having higher sale price.
- Most of the houses which are normally sold are more in the dataset. But the sale price is higher for houses which were not completed wehn last assessed(associated with new homes).

Interpretation of the Results

- The columns ['1stFlrSF', 'Exterior2nd', 'Exterior1st', 'OverallQual', 'TotRmsAbvGrd', 'GarageYrBlt', 'LotArea', 'FullBath', 'YearRemodAdd', 'BsmtUnfSF', 'GarageArea', 'CentralAir', 'BedroomAbvGr', 'Foundation', 'HouseStyle', 'Fireplaces', 'MasVnrArea', 'HalfBath', 'BsmtFullBath', 'LotFrontage', 'GarageQual', 'MasVnrType', 'LandSlope', 'GarageCond', 'PavedDrive', 'WoodDeckSF', 'SaleCondition', 'FireplaceQu', 'LandContour', 'Electrical', 'RoofStyle', 'Street', 'Functional', 'Neighborhood', 'OpenPorchSF', 'Condition2', 'ExterCond', 'MoSold', 'Fence', 'BsmtFinType2', 'Condition1', 'BsmtCond', 'RoofMatl'] are positively correlated to the target variable 'SalePrice'. That means increase in the contribution of these variables for a house will increase the sale price of the house property.
- The columns ['YrSold', 'SaleType', 'LotConfig', 'MSSubClass', 'OverallCond', 'BldgType', 'BsmtFinType1', 'Heating', 'MSZoning', 'LotShape', 'BsmtExposure', 'GarageType', 'HeatingQC', 'GarageFinish', 'KitchenQual', 'ExterQual', 'BsmtQual'] are negatively correlated to the Target Variable 'SalePrice'. That increase in the contribution of these variables for a house will reduce it's sale price.

Regularization

Lasso(L1)



```
ls = Lasso(alpha = 1, random_state = 0)
ls.fit(features_train,target_train)
pred_ls = ls.predict(features_test)

lss = r2_score(target_test,pred_ls)
lss
```

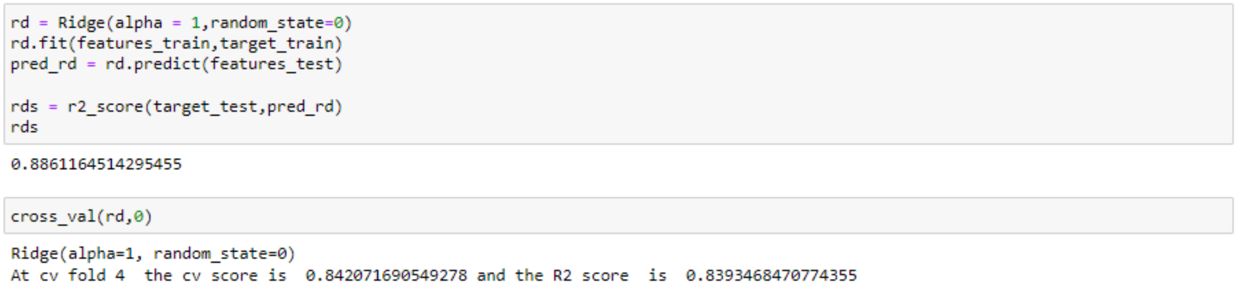
```
0.8861186304195187
```

```
cross_val(ls,0)
```

```
Lasso(alpha=1, random_state=0)
At cv fold 4  the cv score is  0.8419721483014154 and the R2 score  is  0.8392727096588212
```

Lasso Regularization Performance

Ridge(L2)



```
rd = Ridge(alpha = 1,random_state=0)
rd.fit(features_train,target_train)
pred_rd = rd.predict(features_test)

rds = r2_score(target_test,pred_rd)
rds
```

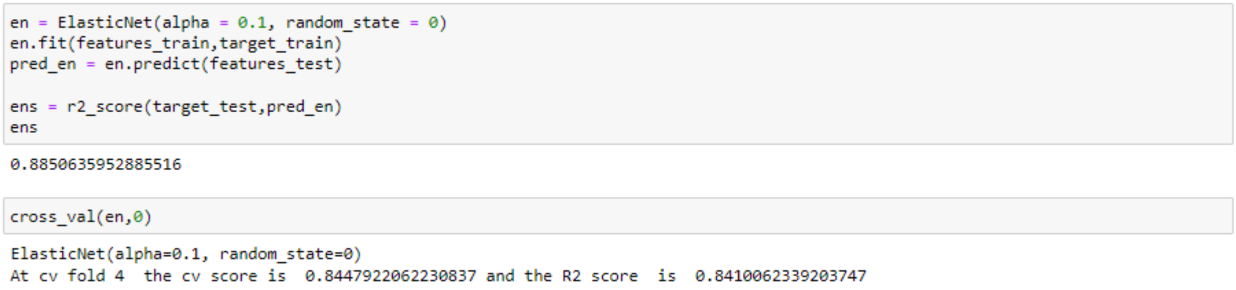
```
0.8861164514295455
```

```
cross_val(rd,0)
```

```
Ridge(alpha=1, random_state=0)
At cv fold 4  the cv score is  0.842071690549278 and the R2 score  is  0.8393468470774355
```

Ridge Regularization Performance

ElasticNet



```
en = ElasticNet(alpha = 0.1, random_state = 0)
en.fit(features_train,target_train)
pred_en = en.predict(features_test)

ens = r2_score(target_test,pred_en)
ens
```

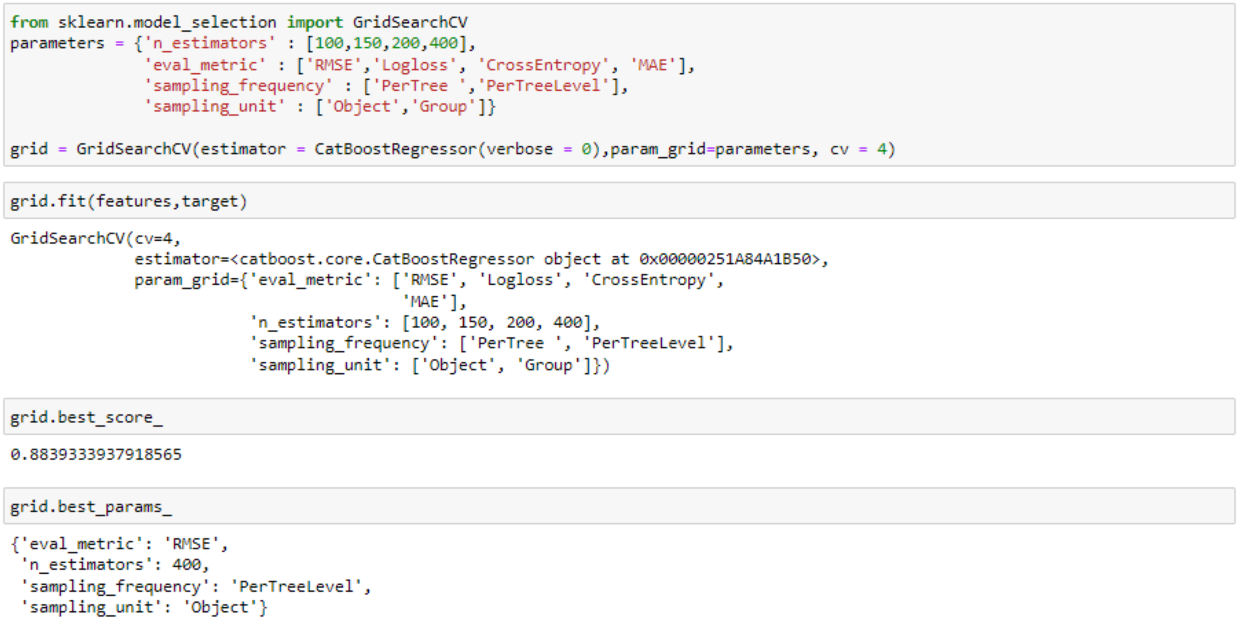
```
0.8850635952885516
```

```
cross_val(en,0)
```

```
ElasticNet(alpha=0.1, random_state=0)
At cv fold 4  the cv score is  0.8447922062230837 and the R2 score  is  0.8410062339203747
```

ElasticNet Regularization Performance

Hyperparameter Tuning



```
from sklearn.model_selection import GridSearchCV
parameters = {'n_estimators' : [100,150,200,400],
              'eval_metric' : ['RMSE','Logloss', 'CrossEntropy', 'MAE'],
              'sampling_frequency' : ['PerTree ','PerTreeLevel'],
              'sampling_unit' : ['Object','Group']}

grid = GridSearchCV(estimator = CatBoostRegressor(verbose = 0),param_grid=parameters, cv = 4)
```

```
grid.fit(features,target)
```

```
GridSearchCV(cv=4,
             estimator=<catboost.core.CatBoostRegressor object at 0x00000251A84A1B50>,
             param_grid={'eval_metric': ['RMSE', 'Logloss', 'CrossEntropy',
                                         'MAE'],
                         'n_estimators': [100, 150, 200, 400],
                         'sampling_frequency': ['PerTree ', 'PerTreeLevel'],
                         'sampling_unit': ['Object', 'Group']})
```

```
grid.best_score_
```

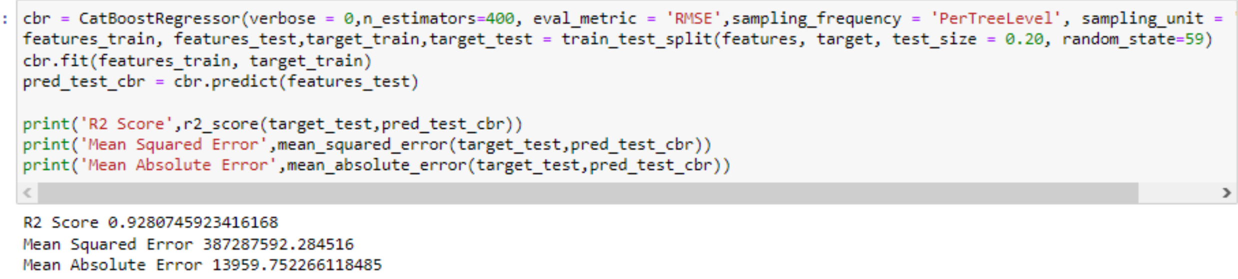
```
0.8839333937918565
```

```
grid.best_params_
```

```
{'eval_metric': 'RMSE',
 'n_estimators': 400,
 'sampling_frequency': 'PerTreeLevel',
 'sampling_unit': 'Object'}
```

Hyper Parameter Tuning of the Best Performing Model

After all the tests, cross validations, regularizations and hyper parameter tuning, the model is performing slighly better. The final model, CatBoost Regressor(cbr) is providing an R2 Score of 92.8%.

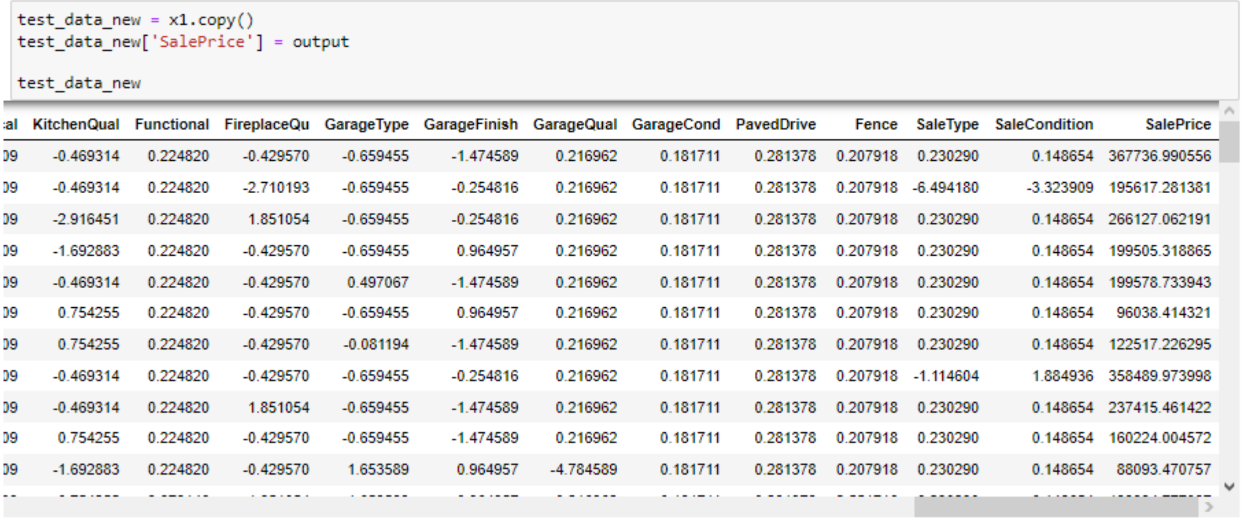


```
cbr = CatBoostRegressor(verbose = 0,n_estimators=400, eval_metric = 'RMSE',sampling_frequency = 'PerTreeLevel', sampling_unit = 
features_train, features_test,target_train,target_test = train_test_split(features, target, test_size = 0.20, random_state=59)
cbr.fit(features_train, target_train)
pred_test_cbr = cbr.predict(features_test)

print('R2 Score',r2_score(target_test,pred_test_cbr))
print('Mean Squared Error',mean_squared_error(target_test,pred_test_cbr))
print('Mean Absolute Error',mean_absolute_error(target_test,pred_test_cbr))
```

```
R2 Score 0.9280745923416168
Mean Squared Error 387287592.284516
Mean Absolute Error 13959.752266118485
```

Snapshot of performance of the final prediction model

Predicting the Target Data for the Test Data



```
test_data_new = x1.copy()
test_data_new['SalePrice'] = output

test_data_new
```

:al	KitchenQual	Functional	FireplaceQu	GarageType	GarageFinish	GarageQual	GarageCond	PavedDrive	Fence	SaleType	SaleCondition	SalePrice
09	-0.469314	0.224820	-0.429570	-0.659455	-1.474589	0.216962	0.181711	0.281378	0.207918	0.230290	0.148654	367736.990556
09	-0.469314	0.224820	-2.710193	-0.659455	-0.254816	0.216962	0.181711	0.281378	0.207918	-6.494180	-3.323909	195617.281381
09	-2.916451	0.224820	1.851054	-0.659455	-0.254816	0.216962	0.181711	0.281378	0.207918	0.230290	0.148654	266127.062191
09	-1.692883	0.224820	-0.429570	-0.659455	0.964957	0.216962	0.181711	0.281378	0.207918	0.230290	0.148654	199505.318865
09	-0.469314	0.224820	-0.429570	0.497067	-1.474589	0.216962	0.181711	0.281378	0.207918	0.230290	0.148654	199578.733943
09	0.754255	0.224820	-0.429570	-0.659455	0.964957	0.216962	0.181711	0.281378	0.207918	0.230290	0.148654	96038.414321
09	0.754255	0.224820	-0.429570	-0.081194	-1.474589	0.216962	0.181711	0.281378	0.207918	0.230290	0.148654	122517.226295
09	-0.469314	0.224820	-0.429570	-0.659455	-0.254816	0.216962	0.181711	0.281378	0.207918	-1.114604	1.884936	358489.973998
09	-0.469314	0.224820	1.851054	-0.659455	-1.474589	0.216962	0.181711	0.281378	0.207918	0.230290	0.148654	237415.461422
09	0.754255	0.224820	-0.429570	-0.659455	-1.474589	0.216962	0.181711	0.281378	0.207918	0.230290	0.148654	160224.004572
09	-1.692883	0.224820	-0.429570	1.653589	0.964957	-4.784589	0.181711	0.281378	0.207918	0.230290	0.148654	88093.470757

Predicted Sale Price of houses from the Test data using the built model.

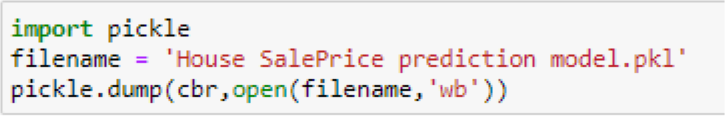
The CatBoost Regression model have predicted the house prices for with the test data.

Our model was able to predict the SalePrice of the house using the test data with R2 score of 92.8%.

We have saved the new dataset with predicted saleprice as :

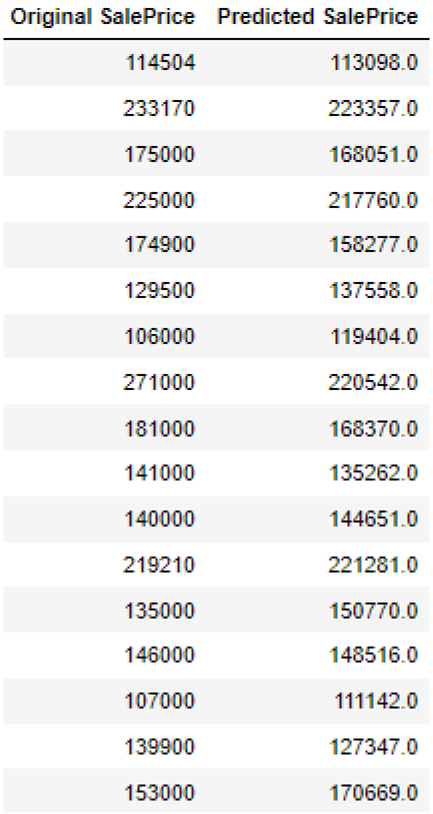
House SalePrice predicted with test data.csv

We have also saved the machine learning model for future predictions



```
import pickle
filename = 'House SalePrice prediction model.pkl'
pickle.dump(cbr,open(filename,'wb'))
```

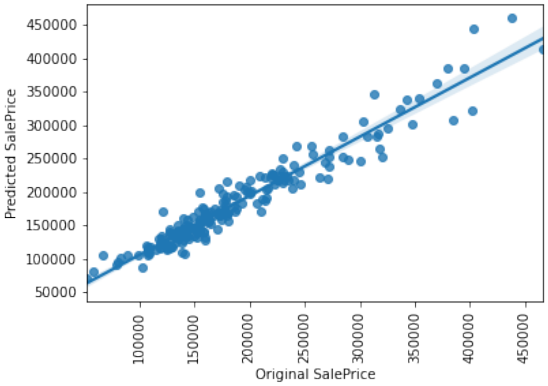
To check whether the model is performing well with predictions, we have made predictions on train data which already have the exact sale price. So we can check the variance of predicted and actual price of the house and evaluate the performance of the model in predictions.



Original SalePrice	Predicted SalePrice
114504	113098.0
233170	223357.0
175000	168051.0
225000	217760.0
174900	158277.0
129500	137558.0
106000	119404.0
271000	220542.0
181000	168370.0
141000	135262.0
140000	144651.0
219210	221281.0
135000	150770.0
146000	148516.0
107000	111142.0
139900	127347.0
153000	170669.0

1Sample Snapshot of predicted and actual sale price

Our model is performing well with predictions and provided almost accurate results.



2Plotting the Actual and predicted data

From the above observation, we can see that our model has almost predicted all the results with slight variations. Our model is providing an R2 Score of 92.8%.

CONCLUSION

Key Findings and Conclusions of the Study

The successful creation of a powerful machine learning model that can predict the sale price of house using the important features was done using the CatBoost algorithm. Now this model can be used to predict the price of houses in Australia for the company Surprise Housing with the following variable information about the house.(**Important Variables**)

* `MSSubClass` * `BldgType` * `1stFlrSF` * `Exterior2nd` * `Exterior1st`

* `OverallQual` * `TotRmsAbvGrd` * `GarageYrBlt` * `ExterQual` * `LotArea`

* `FullBath` * `YearRemodAdd` * `BsmtUnfSF` * `GarageArea` * `CentralAir`

* `BedroomAbvGr` * `BsmtQual` * `KitchenQual` * `Foundation` * `HouseStyle`

* `Fireplaces` * `OverallCond` * `MasVnrArea` * `HalfBath`

*`BsmtFinType1` *`BsmtFullBath` * `LotFrontage` * `GarageFinish`* `GarageQual`

* `MasVnrType` * `BsmtExposure` * `GarageType` * `LandSlope` * `Heating`

* `HeatingQC` * `GarageCond` * `MSZoning` * `PavedDrive`

* `WoodDeckSF` * `SaleCondition` * `FireplaceQu` * `LandContour` * `LotShape`

* `Electrical` * `RoofStyle` * `Street` * `SaleType` * `Functional`

* `Neighborhood` * `OpenPorchSF` * `Condition2` * `ExterCond` *`MoSold`

* `Fence` * `LotConfig` *`BsmtFinType2` * `Condition1` *`BsmtCond`

* `RoofMatl` * `YrSold`

Impact of Variables on Target Variable(Correlation)

- The columns ['1stFlrSF', 'Exterior2nd', 'Exterior1st', 'OverallQual', 'TotRmsAbvGrd', 'GarageYrBlt', 'LotArea', 'FullBath', 'YearRemodAdd', 'BsmtUnfSF', 'GarageArea', 'CentralAir', 'BedroomAbvGr', 'Foundation', 'HouseStyle', 'Fireplaces', 'MasVnrArea', 'HalfBath', 'BsmtFullBath', 'LotFrontage', 'GarageQual', 'MasVnrType', 'LandSlope', 'GarageCond', 'PavedDrive', 'WoodDeckSF', 'SaleCondition', 'FireplaceQu', 'LandContour', 'Electrical', 'RoofStyle', 'Street', 'Functional', 'Neighborhood', 'OpenPorchSF', 'Condition2', 'ExterCond', 'MoSold', 'Fence', 'BsmtFinType2', 'Condition1', 'BsmtCond', 'RoofMatl'] are positively correlated to the target variable 'SalePrice'. That means increase in the contribution of these variables for a house will increase the sale price of the house property.
- The columns ['YrSold', 'SaleType', 'LotConfig', 'MSSubClass', 'OverallCond', 'BldgType', 'BsmtFinType1', 'Heating', 'MSZoning', 'LotShape', 'BsmtExposure', 'GarageType', 'HeatingQC', 'GarageFinish', 'KitchenQual', 'ExterQual', 'BsmtQual'] are negatively correlated to the Target Variable 'SalePrice'. That increase in the contribution of these variables for a house will reduce its sale price.

Learning Outcomes of the Study in respect of Data Science

The problem was prominent at present as house properties are important next to basic needs for a human being. This shows the scope of the real estate market, and this is the one of the major reasons there are many companies involved and looking forward to this market. With the help of data science and machine learning we were able to build a model that can predict the sale price of the houses using the several information about the houses.

We have used CatBoost algorithm as it is based on gradient boosting on decision tree algorithm and since this a regression problem, the algorithm can use mean squared error as cost function. Thus, after all the tests, validations and hyper parameter tunings, the model could give an R2 Score of 92.8% which is quite good.

The data collected by Surprise Housing company was well versed and included much information about the houses in Australia. But there was many missing information, and the data was limited to 1460 houses including the Train and Test data. So, the scope of learning of the algorithm is limited and thus there can be errors in the predictions. There were many extreme outliers and high level of skewness present in the data. Since the data was limited to several houses and when it includes extreme outliers and skewed data, this will affect the model in learning as the data is complex and it will not be easy to learn the patterns of independent features in contribution to the target variable.

Limitations of this work and Scope for Future Work

Limitations

- Even though the dataset included many prominent features of the houses for prediction, it missed out many important features which can affect the sale price of a house such as price of houses in nearby places, availability of basic amenities, weather etc.
- The dataset was limited to 1460 houses including the train and test data. So, machine was able to identify the patterns from the limited data. The dataset was noisy as the data was not normally distributed and there were extreme outlier presence and there were highly skewed data in the dataset.
- There were many missing data from the dataset. We tried to avoid variables with more than 90% missing data and imputed mean and mode for missing data in other numerical and categorical data. But this has resulted in slight bias towards this majority categories or central tendencies.
- Even though we were able to built a powerful machine learning algorithm that can predict the price of houses, there are several factors that can influence the house price and there are limitations in including these data for learning purpose as these factors are always fluctuating and its impact on house property sale price changes over time.

Scope

- While real estate and housing have been dominating the market since a long time, it is always volatile and subject to market and economic conditions. But with the integration of machine learning and data science, we can analyze the vast input data that influence the valuation of house properties and draw inferences from this big data.
- Many of the real estate giants and housing companies already started implementing machine learning and early adoption of data science in this sector to maintain their dominancy. This unleashes a vast scope for data science and machine learning in this sector because of the instant impact and huge savings on time and effort in the research.
- Implementation and application of machine learning algorithms in this sector is not limited to providing accurate predictions through various input features, it is further giving option for recalibration of the model as the market is always fluctuating and the features influencing the house pricing and valuation can always change. So, with machine learning and data science our model is forward looking and can cope with changes in features with an option to addition of new features and removal of existing features from the predicting framework.

Thank You