HOUSING: PRICE PREDICTION

Submitted by:

Shweta Mishra

ACKNOWLEDGMENT

I would like to express my special gratitude to “Flip Robo” team, who has given me this opportunity to deal with a beautiful dataset and it has helped me to improve my analyzation skills. A huge thanks to “Data trained” who are the reason behind my Internship at Fliprobo

References use in this project:

1. SCIKIT Learn Library Documentation

2. Blogs from towardsdatascience, Analytics Vidya, Medium

3. Andrew Ng Notes on Machine Learning (GitHub)

4. Data Science Projects with Python Second Edition by Packt

5. Hands on Machine learning with scikit learn and tensor flow by Aurelien Geron

8. Zhongyun, Jiang, Guoxin, Shen, “Prediction of House Price Based on The Back Propagation Neural Network in The Keras Deep Learning Framework”, 2019 6th International Conference on Systems and Informatics (ICSA I2019).

9. Zhen Peng, Qiang Huang, Yincheng Han, “Model Research on Forecast of Second-Hand House Price in Chengdu Based on XGboost Algorithm”, 2019 IEEE 11th International Conference on Advanced Infocomm Technology

10.Ayush Varma, Abhijit Sarma, Rohini Nair and Sagar Doshi, ”House Price Prediction Using Machine Learning And Neural Networks”, @2018 IEEE, 2018 Second International Conference on Inventive Communication and Computational Technologies(ICICCT), Coimbatore, India, DOI:10.1109/ICICCT.2018.8473231.

11. Arietta, Sean M., et al. "City forensics: Using visual elements to predict non-visual city attributes." IEEE transactions on visualization and computer graphics 20.12 (2014): 2624-2633

INTRODUCTION

- ## Business Problem Framing

Houses are one of the necessary need of each and every person around the globe and therefore housing and real estate market is one of the markets which is one of the major contributors in the world's economy. It is a very large market and there are various companies working in the domain. Data science comes as a very important tool to solve problems in the domain to help the companies increase their overall revenue, profits, improving their marketing strategies and focusing on changing trends in house sales and purchases. Predictive modelling, Market mix modelling, recommendation systems are some of the machine learning techniques used for achieving the business goals for housing companies. Our problem is related to one such housing company

In general, purchasing and investing in any real estate project will involve various transactions between different parties. Thus, it could be a vital decision for both households and enterprises. How to construct a realistic model to precisely predict the price of real estate has been a challenging topic with great potential for further research.

There are many factors that have an impact on house prices, such as the number of bedrooms and bathrooms. House price depends upon its location as well. A house with great accessibility to highways, schools, malls, employment opportunities, would have a greater price as compared to a house with no such accessibility.

Regression is a supervised learning algorithm in machine learning which is used for prediction by learning and forming a relationship between present statistical data and target value i.e., Sale Price in this case. Different factors are taken into consideration while predicting the worth of the house like location, neighbourhood and various amenities like garage space etc. if learning is applied to above parameters with target values for a certain geographical region as different areas differ in price like land price, housing style, material used, availability of public utilities.

- ## Conceptual Background of the Domain Problem

A US-based housing company named Surprise Housing has decided to enter the Australian market. The company uses data analytics to purchase houses at a price below their actual values and flip them at a higher price. For the same purpose, the company has collected a data set from the sale of houses in Australia. The data is provided in the CSV file below. The company is looking at prospective properties to buy houses to enter the market. You are required to build a model using Machine Learning in order to predict the actual value of the prospective properties and decide whether to invest in them or not.

For this company wants to know:

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

It is required 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

Before start developing Machine Learning Model, we need to study the previous papers of our domain which we are working and on the basis of study we can predict or generate the drawback and start working with the reference of previous papers. In this section, we briefly review the related work on house price prediction and the techniques used. Predicting house prices manually is a difficult task and generally not very accurate, hence there are many systems developed for house price prediction

CH.Raga Madhuri, Anuradha G et.al , estimated house price by the analysis of fare ranges, foregoing merchandise and forewarns of developments. The author discussed diverse regression techniques such as Gradient boosting and AdaBoost Regression, Ridge, Elastic Net, Multiple linear, LASSO to locate the most excellent. The performance measures used are [MSE] Mean Square Error and [RMSE] Root Mean Square Error.

Zhongyun, Jiang, Guoxin, Shen, develops 6-layer BP neural network with Kera's deep learning. For backend Tens flow or Theano is used. The test results give the predict real value of house with accuracy of 95.59%. The Gaussian noise model is favourable for the house price forecast.

Liu et al. have constructed a statistical model based on the fuzzy neural network prediction model, which incorporates the hedonic theory and a great database with relevant characteristics affecting the price of properties based on recently sold projects. The experimental outcome and analysis have shown that the fuzzy neural network prediction model has a promising ability for real estate price prediction given reliable input data with high quality.

According to Kusan et al. [11], factors affecting housing sale price can be classified into three types: house factors, environmental factors, and transportation factors. The most influential type is residential factors, including residence, usability, and number of rooms. When people consider purchasing a house for living purposes, the factors above are the main determinants for the living quality. Buyers with family members would typically attach more importance to the essential feature of the house, like the living area and number of rooms, which have a significant impact on the overall living quality and experience in the house. Besides, the intangible features, like the view of residence and usability, also have a rather considerable influence on the housing price, through affecting buyers' experience on the house and willingness to pay. The other influential types are the main factors related to building properties and floor factors. Building properties are mainly about hardware and basic facilities

in the building, such as the elevator, generator, and garage. The rising trend of numbers of vehicles per household possessing generates a necessary demand for the quality and capacity of a garage in a house. Other affiliated facilities to the house like the swimming pool and backyard have also played an essential role in determining the housing price, as the demand for leisure and relaxation has been arising with the economic progress. On the other hand, floor factors, like the number of floors, have also impacted the housing price significantly. A family with children and elders tends to prefer a house with multi floor construction, which offers different family members separate living areas with appropriate privacy while living together. Environmental factors mainly consist of two parts: regional environment and nearby pollution. Regional Scientific Pro environment refers to the overall living conditions in the surrounding community. Sanitation, as a significant indicator of the living quality, has been given more importance in the recent decades. A community with comprehensive sanitation services tends to attract buyers to pay a higher price.

- # Motivation for the Problem Undertaken

The project is provided to me by Flip Robo Technologies as a part of the internship programme. The exposure to real world data and the opportunity to deploy my skillset in solving a real time problem has been the primary motivation.

Our main objective of doing this project is to build a model to predict the house prices with the help of other supporting features. In order to improve the selection of customers, the client wants some predictions that could help them in further investment and improvement in selection of customers.

The No Free Lunch Theorem state that algorithms perform differently when they are used under the same circumstances. This study aims to analyse & predicting house prices when using multiple linear, Ridge and Random Forest regression algorithms. Thus, the purpose of this study is to deepen the knowledge in regression methods in machine learning.

In addition, the given datasets should be processed to enhance performance, which is accomplished by identifying the necessary features by applying one of the selection methods to eliminate the unwanted variables since each house has its unique features that help to estimate its price.

Analytical Problem Framing

- ## Mathematical/ Analytical Modeling of the Problem

  Our objective is to predict House price which can be resolve by use of regression-based algorithm. In this project we are going to use different types of algorithms which uses their own mathematical equation on background. This project comes with two separate data set for training & testing model. Initially data cleaning & pre-processing perform over data. Feature engineering is performed to remove unnecessary feature & for dimensionality reduction. In model building Final model is select based on evaluation benchmark among different models with different algorithms. Further Hyperparameter tuning performed to build more accurate model out of best model.

- ## Data Sources and their formats

  Data set provided by Flip Robo was in the format of CSV (Comma Separated Values). There are 2 data sets that are given. One is training data and one is testing data.

  1) Train file will be used for training the model, i.e., the model will learn from this file. It contains all the independent variables and the target variable. The dimension of data is 1168 rows and 81 columns.
  2) Test file contains all the independent variables, but not the target variable. We will apply the model to predict the target variable for the test data. The dimension of data is 292 rows and 80 columns.

```
# importing dataset Train CSV file using Pandas libarary
df = pd.read_csv('train.csv')
```

```
print('No. of Rows :',df.shape[0])
print('No. of Columns :', df.shape[1])
```

```
No. of Rows : 1168
No. of Columns : 81
```

The data types of different features

```
# As we have too many columns let sort columns by their datatype
df.columns.to_series().groupby(df.dtypes).groups
```

```
{int64: ['Id', 'MSSubClass', 'LotArea', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd', 'BsmtFinSF1', 'BsmtFinSF2',
'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'H
alfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'TotRmsAbvGrd', 'Fireplaces', 'GarageCars', 'GarageArea', 'WoodDeckSF', 'OpenPorchS
F', 'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'MiscVal', 'MoSold', 'YrSold', 'SalePrice'], float64: ['LotFrontag
e', 'MasVnrArea', 'GarageYrBlt'], object: ['MSZoning', 'Street', 'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType', 'HouseStyle', 'RoofStyle', 'RoofMatl', 'Exterior1st', 'Ext
erior2nd', 'MasVnrType', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtF
inType2', 'Heating', 'HeatingQC', 'CentralAir', 'Electrical', 'KitchenQual', 'Functional', 'FireplaceQu', 'GarageType', 'Garage
Finish', 'GarageQual', 'GarageCond', 'PavedDrive', 'PoolQC', 'Fence', 'MiscFeature', 'SaleType', 'SaleCondition']}
```

- # Data Preprocessing Done

The dataset is large and it may contain some data error. In order to reach clean, error free data some data cleaning & data pre-processing performed data.

• Data Integrity check – No duplicate entries present in dataset.

Data Integrity check

```
df.duplicated().sum()
```

```
0
```

Check for presense of any whitespaces, '?', 'NA', '-' in dataset

```
df.isin(['NA','N/A','-',' ','?',' ?']).sum().any()
```

```
False
```

Some features contain missing values as shown below:

```
# missing data percentage from dataset
pd.set_option('display.max_rows',None)
missing_values = df.isnull().sum().sort_values(ascending = False)
percentage_missing_values = (missing_values/len(df))*100
print(pd.concat([missing_values, percentage_missing_values], axis = 1, keys =['Missing Values', '% Missing data']))
```

	Missing Values	% Missing data
PoolQC	1161	99.400685
MiscFeature	1124	96.232877
Alley	1091	93.407534
Fence	931	79.708904
FireplaceQu	551	47.174658
LotFrontage	214	18.321918
GarageYrBlt	64	5.479452
GarageFinish	64	5.479452
GarageType	64	5.479452
GarageQual	64	5.479452
GarageCond	64	5.479452
BsmtExposure	31	2.654110
BsmtFinType2	31	2.654110
BsmtQual	30	2.568493
BsmtCond	30	2.568493
BsmtFinType1	30	2.568493
MasVnrType	7	0.599315
MasVnrArea	7	0.599315
Id	0	0.000000
Functional	0	0.000000
Fireplaces	0	0.000000
KitchenQual	0	0.000000
KitchenAbvGr	0	0.000000
BedroomAbvGr	0	0.000000
HalfBath	0	0.000000

We have removed feature which contain high amount missing values e.g., Top 5 features with missing value in above list. Rest of feature are handle based on mean, median or mode imputation depending on outliers & distribution of feature.

```
# Removing columns with max missing value percentage
dft.drop(['PoolQC','MiscFeature','Alley','Fence','FireplaceQu'],axis =1, inplace = True)
```

```
dft['LotFrontage'] = dft['LotFrontage'].fillna(dft['LotFrontage'].median())
dft['GarageType'] = dft['GarageType'].fillna(dft['GarageType'].mode()[0])
dft['GarageYrBlt']= dft['GarageYrBlt'].fillna(dft['GarageYrBlt'].mode()[0])
dft['GarageFinish']= dft['GarageFinish'].fillna(dft['GarageFinish'].mode()[0])
dft['GarageQual'] = dft['GarageQual'].fillna(dft['GarageQual'].mode()[0])
dft['GarageCond'] = dft['GarageCond'].fillna(dft['GarageCond'].mode()[0])
dft['BsmtFinType2'] = dft['BsmtFinType2'].fillna(dft['BsmtFinType2'].mode()[0])
dft['BsmtExposure'] = dft['BsmtExposure'].fillna(dft['BsmtExposure'].mode()[0])
dft['BsmtFinType1'] = dft['BsmtFinType1'].fillna(dft['BsmtFinType1'].mode()[0])
dft['BsmtCond'] = dft['BsmtCond'].fillna(dft['BsmtCond'].mode()[0])
dft['BsmtQual'] = dft['BsmtQual'].fillna(dft['BsmtQual'].mode()[0])
dft['MasVnrType'] = dft['MasVnrType'].fillna(dft['MasVnrType'].mode()[0])
dft['MasVnrArea'] = dft['MasVnrArea'].fillna(dft['MasVnrArea'].median())
dft['Electrical'] = dft['Electrical'].fillna(dft['Electrical'].mode()[0])
```

Some of the features are remove as they are representing in another similar feature.

```
TotalBsmtSF is sum of above remaining features. We will drop other three features for modelling.
```

```
df.drop(['BsmtFinSF1','BsmtFinSF2','BsmtUnfSF'],axis=1,inplace=True)
dft.drop(['BsmtFinSF1','BsmtFinSF2','BsmtUnfSF'],axis=1,inplace=True)
```

```
GrLivArea is sum of above remaining features. We will drop other three features for modelling.
```

```
df.drop(['1stFlrSF','2ndFlrSF','LowQualFinSF'],axis=1,inplace=True)
dft.drop(['1stFlrSF','2ndFlrSF','LowQualFinSF'],axis=1,inplace=True)
```

```
Some of the features contain lot of zeros (above 90 %), it will meaningless to keep them while model building.
```

```
PoolArea - 1161 zeros out of 1168 entries
MiscVal - 1126 zeros out of 1168 entries
3SsnPorch - 1146 zeros out of 1168 entries
EnclosedPorch - 999 zeros out of 1168 entries
ScreenPorch - 1073 zeros out of 1168 entries
```

```
#We will drop these columns from dataset.
df.drop(['EnclosedPorch','3SsnPorch','ScreenPorch','PoolArea','MiscVal'],axis=1,inplace=True)
dft.drop(['EnclosedPorch','3SsnPorch','ScreenPorch','PoolArea','MiscVal'],axis=1,inplace=True)
```

- **Label Encoding of Categorical features:**

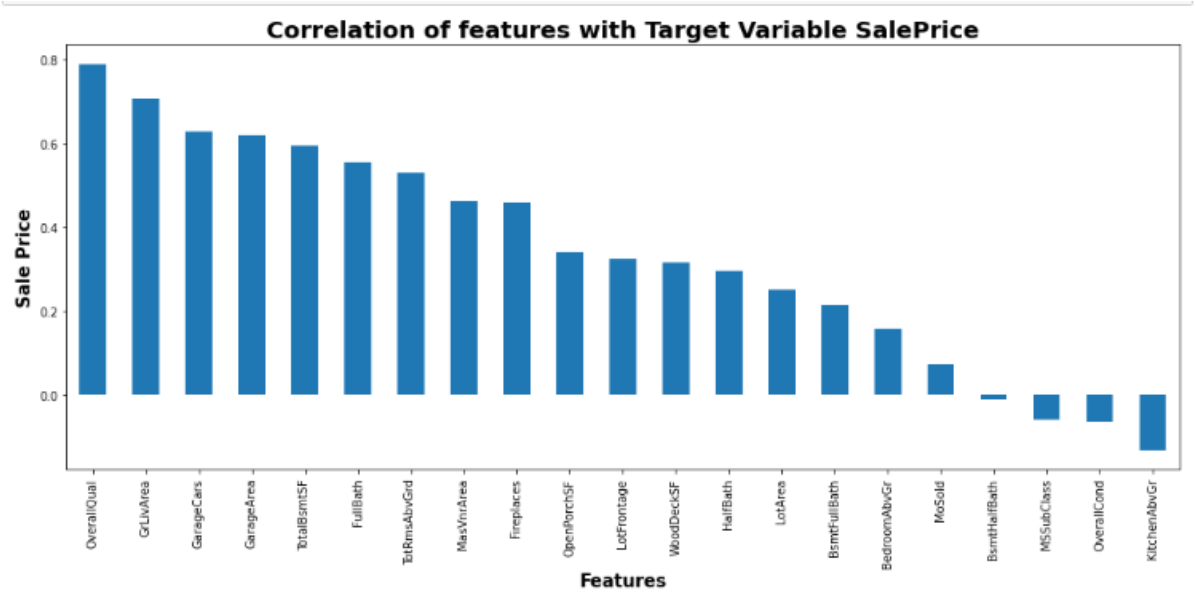
The categorical Variable in training & testing dataset are converted into numerical datatype using label encoder from scikit library.

Encoding for Training data

```
# Using Label encoder for transforming Categorical data
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
for i in Categorical_features:
    df[i] = le.fit_transform(df[i])
df.head()
```

- Data Inputs- Logic- Output Relationships

Correlation heatmap is plotted to gain understanding of relationship between target features & independent features. We can see that lot of features are highly correlated with target variable Sale Price. To gain insights about relationship between Input & output different types of visualization are plotted which we will see in EDA section of this report.



- # Hardware and Software Requirements and Tools Used

Hardware Used –

1. Processor — Intel i3 processor with 2.4GHZ

2. RAM — 4 GB

3. GPU — 2GB AMD Radeon Graphics card

Software utilised –

1. Anaconda – Jupyter Notebook

2. Google Colab – for Hyper parameter tuning

Libraries Used – General libraries used for data wrangling

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

Libraries used for machine learning model building

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
```

Model/s Development and Evaluation

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

Our objective is to predict house price and analyse feature impacting Sale price. This problem can be solve using regression-based machine learning algorithm like linear regression. For that purpose, first task is to convert categorical variable into numerical features.

Once data encoding is done then data is scaled using standard scalar. Final model is built over this scaled data. For building ML model before implementing regression algorithm, data is split in training & test data using train_test_split from model_selection module of sklearn library.

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model. After that model is train with various regression algorithm and 5-fold cross validation is performed. Further Hyperparameter tuning performed to build more accurate model out of best model.

- ## Testing of Identified Approaches (Algorithms)

The different regression algorithm used in this project to build ML model are as below:

- Linear Regression
- Random Forest Regressor
- Decision Tree Regressor
- Ridge Regression

Key Metrics for success in solving problem under consideration

Following metrics used for evaluation:

1. Mean absolute error which gives magnitude of difference between the prediction of an observation and the true value of that observation.

2. Root mean square error is one of the most commonly used measures for evaluating the quality of predictions.

3. R2 score which tells us how accurate our model predict result, is going to important evaluation criteria along with Cross validation score.

4. Cross Validation Score

- Run and Evaluate selected models

  - ***Linear Regression***

```
x_train, x_test, y_train, y_test = train_test_split(x_scale, y, random_state= 135, test_size=0.33)
lin_reg= LinearRegression()
lin_reg.fit( x_train,y_train)
y_pred = lin_reg.predict(x_test)
print('\033[1m'+ 'Error :'+ '\033[0m')
print('Mean absolute error :', mean_absolute_error(y_test,y_pred))
print('Mean squared error :', mean_squared_error(y_test, y_pred))
print('Root Mean squared error :', np.sqrt(mean_squared_error(y_test, y_pred)))
print('\033[1m'+' R2 Score :'+'\033[0m')
print(r2_score(y_test,y_pred)*100)
```

```
Error :
Mean absolute error : 20185.765166015673
Mean squared error : 866180861.134785
Root Mean squared error : 29430.95073446974
 R2 Score :
87.55508234163038
```

```
from sklearn.model_selection import cross_val_score
score = cross_val_score(lin_reg, x_scale, y, cv=5)
print('\033[1m'+'Cross Validation Score :',lin_reg,":"+'\033[0m\n')
print("Mean CV Score :",score.mean())
print('Difference in R2 & CV Score:',(r2_score(y_test,y_pred)*100)-(score.mean()*100))
```

```
Cross Validation Score : LinearRegression() :

Mean CV Score : 0.7689632941334926
Difference in R2 & CV Score: 10.658752928281118
```

  - ***Random Forest Regressor***

```
x_train, x_test, y_train, y_test = train_test_split(x_scale, y, random_state= 135, test_size=0.33)
rfc = RandomForestRegressor()
rfc.fit( x_train,y_train)
y_pred = rfc.predict(x_test)
print('\033[1m'+ 'Error of Random Forest Regressor:'+ '\033[0m')
print('Mean absolute error :', mean_absolute_error(y_test,y_pred))
print('Mean squared error :', mean_squared_error(y_test, y_pred))
print('Root Mean squared error :', np.sqrt(mean_squared_error(y_test, y_pred)))
print('\033[1m'+'R2 Score of Random Forest Regressor :'+'\033[0m')
print(r2_score(y_test,y_pred)*100)
```

```
Error of Random Forest Regressor:
Mean absolute error : 18447.565647668394
Mean squared error : 739607439.4685146
Root Mean squared error : 27195.72465422671
R2 Score of Random Forest Regressor :
89.37363534949893
```

```
from sklearn.model_selection import cross_val_score
score = cross_val_score(rfc, x_scale, y, cv=5)
print('\033[1m'+'Cross Validation Score :',rfc,":"+'\033[0m\n')
print("Mean CV Score :",score.mean())
print('Difference in R2 & CV Score:',(r2_score(y_test,y_pred)*100)-(score.mean()*100))
```

```
Cross Validation Score : RandomForestRegressor() :

Mean CV Score : 0.8278249761219596
Difference in R2 & CV Score: 6.591137737302972
```

  - ***Decision Tree Regressor***

```
x_train, x_test, y_train, y_test = train_test_split(x_scale, y, random_state= 135, test_size=0.33)
dtc = DecisionTreeRegressor()
dtc.fit(x_train, y_train)
y_pred = dtc.predict(x_test)
print('\033[1m'+ 'Error of Decision Tree Regressor:'+ '\033[0m')
print('Mean absolute error :', mean_absolute_error(y_test,y_pred))
print('Mean squared error :', mean_squared_error(y_test, y_pred))
print('Root Mean squared error :', np.sqrt(mean_squared_error(y_test, y_pred)))
print('\033[1m'+'R2 Score of Decision Tree Regressor :'+'\033[0m')
print(r2_score(y_test,y_pred)*100)
```

```
Error of Decision Tree Regressor:
Mean absolute error : 31468.354922279792
Mean squared error : 2794542151.6088085
Root Mean squared error : 52863.42924564021
R2 Score of Decision Tree Regressor :
59.849208716003986
```

```
from sklearn.model_selection import cross_val_score
score = cross_val_score(dtc, x_scale, y, cv=5)
print('\033[1m'+'Cross Validation Score :',dtc,":"+'\033[0m\n')
print("Mean CV Score :",score.mean())
print('Difference in R2 & CV Score:',(r2_score(y_test,y_pred)*100)-(score.mean()*100))
```

```
Cross Validation Score : DecisionTreeRegressor() :

Mean CV Score : 0.6831697280968356
Difference in R2 & CV Score: -8.467764093679577
```

- ***Ridge Regression***

```
from sklearn.linear_model import Ridge
x_train, x_test, y_train, y_test = train_test_split(x_scale, y, random_state= 135, test_size=0.33)
rd = Ridge()
rd.fit(x_train, y_train)
y_pred = rd.predict(x_test)
print('\033[1m'+ 'Error of XGB Regressor:'+ '\033[0m')
print('Mean absolute error :', mean_absolute_error(y_test,y_pred))
print('Mean squared error :', mean_squared_error(y_test, y_pred))
print('Root Mean squared error :', np.sqrt(mean_squared_error(y_test, y_pred)))
print('\033[1m'+'R2 Score of  Ridge Regressor :'+'\033[0m')
print(r2_score(y_test,y_pred)*100)
```

```
Error of XGB Regressor:
Mean absolute error : 20180.71457038398
Mean squared error : 866147967.3154476
Root Mean squared error : 29430.39189877443
R2 Score of  Ridge Regressor :
87.55555494601532
```

```
from sklearn.model_selection import cross_val_score
score = cross_val_score(rd, x_scale, y, cv=5)
print('\033[1m'+'Cross Validation Score :',rd,":"+'\033[0m\n')
print("Mean CV Score :",score.mean())
print('Difference in R2 & CV Score:',(r2_score(y_test,y_pred)*100)-(score.mean()*100))
```

```
Cross Validation Score : Ridge() :

Mean CV Score : 0.7693107095046043
Difference in R2 & CV Score: 10.624483995554883
```

Fold cross validation performed over all models. We can see that Random Forest Regressor gives maximum R2 score of 90.50 and with cross validation score of 83.30 %. Among all model we will select Random Forest Regressor as final model and we will perform hyper parameter tuning over this model to enhance its R2 Score

```
x_train, x_test, y_train, y_test = train_test_split(x_scale, y, random_state= 135, test_size=0.33)
```

```
print(rfc.get_params())
```

```
{'bootstrap': True, 'ccp_alpha': 0.0, 'criterion': 'squared_error', 'max_depth': None, 'max_features': 1.0, 'max_leaf_nodes': N
one, 'max_samples': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_lea
f': 0.0, 'n_estimators': 100, 'n_jobs': None, 'oob_score': False, 'random_state': None, 'verbose': 0, 'warm_start': False}
```

```
parameter = {
 'bootstrap': [True, False],
 'max_features': ['auto'],
 'min_samples_leaf': [1, 2, 4 ],
 'n_estimators': [100, 500,1000,1500,2000]}
```

```
GCV = GridSearchCV(RandomForestRegressor(),parameter,verbose =10)
```

```
GCV.fit(x_train,y_train)
```

```
Fitting 5 folds for each of 30 candidates, totalling 150 fits
```

Final model is built with best params got in hyper parameter tuning

```
GCV.best_params_
```

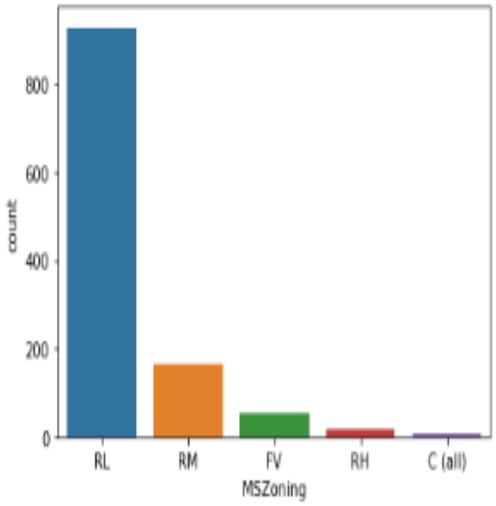
```
{'bootstrap': True,
 'max_features': 'auto',
 'min_samples_leaf': 4,
 'n_estimators': 100}
```

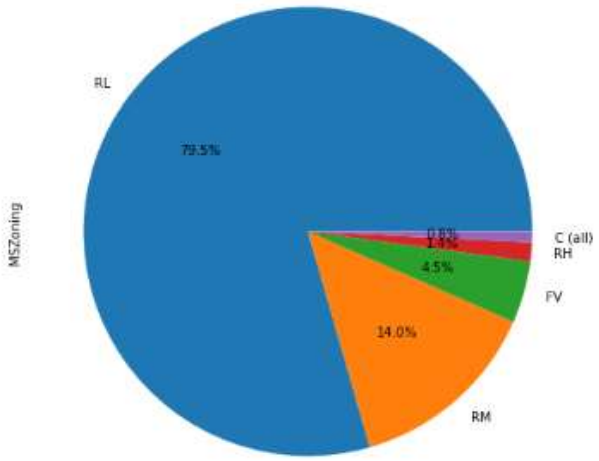
```
#Final Model
Final_mod=RandomForestRegressor(max_depth = None, bootstrap = True, max_features='auto', min_samples_leaf = 4,
                                n_estimators = 100 )
Final_mod.fit(x_train,y_train)
pred=Final_mod.predict(x_test)
print('R2_Score:',r2_score(y_test,pred)*100)
print('mean_squared_error:',mean_squared_error(y_test,pred))
print('mean_absolute_error:',mean_absolute_error(y_test,pred))
print("RMSE value:",np.sqrt(mean_squared_error(y_test, pred)))
```

```
R2_Score: 88.99634446898294
mean_squared_error: 765867327.1395156
mean_absolute_error: 18543.616472519083
RMSE value: 27674.308069751547
```

• Visualizations

Let see key result from EDA, start with zone wise distribution of property.



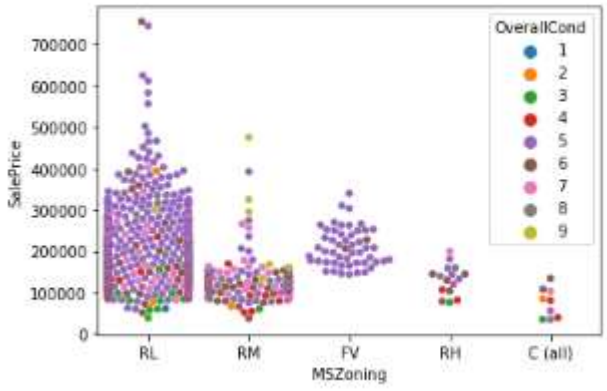


Observation:

79.5% of House properties belongs to Low Density Residential Area followed by 14 % of properties belong to Medium Density Residential Area.

Very Few property (0.8%) belongs to Commerical zone.

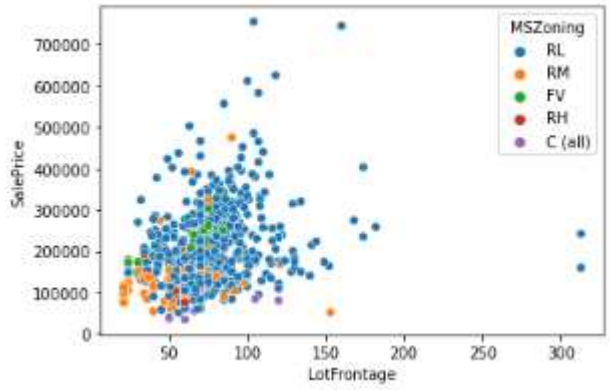
Zone relation with respect to Sale Price



Observation:
Most of property for sale have overall condition rating of either 5 or 6.
Sale Price inside RL Zone is much higher than other remaining zone.
Cheapest properties are available in Commerical zone.
Some house properties having Overall condition Rating of 8 & 9 have low price compare to others.

This indicate that Overall Condition Rating is Not significant factor in determination of Sale price.

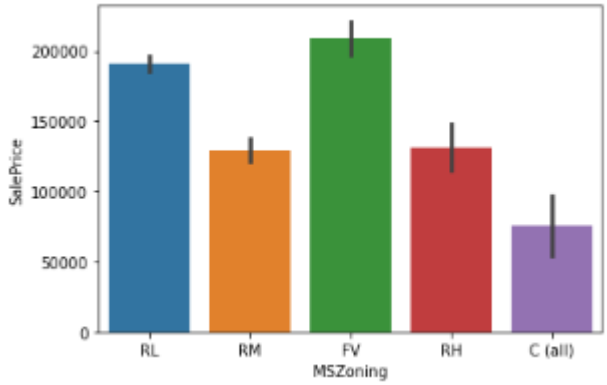
Sale Price Vs LotFrontage



Observation:

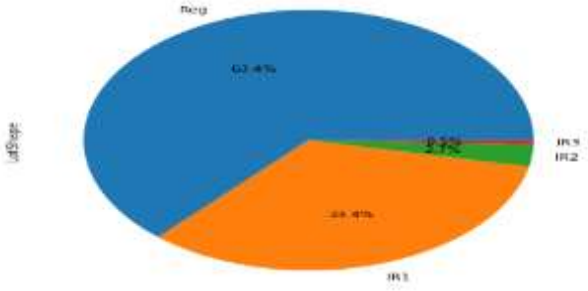
Lot Frontage area increase (which indicate Size of street connected to property) the Sale Price increases.

Avg. Sale Price as Per Zone



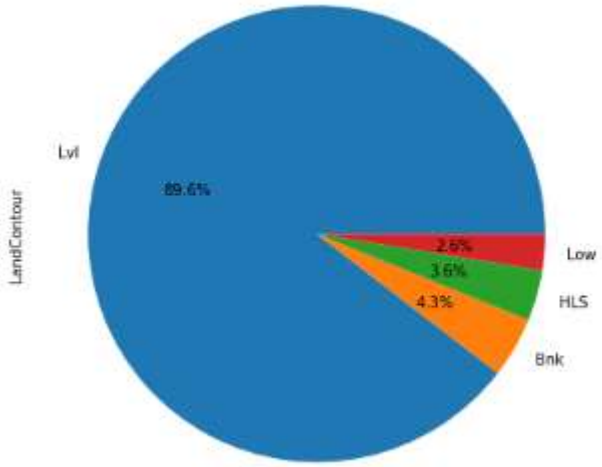
Observation:
Average Sale price house properties belonging to Floating Village Residential Zone are costiler than Others.

LotShape-wise of Property Distribution



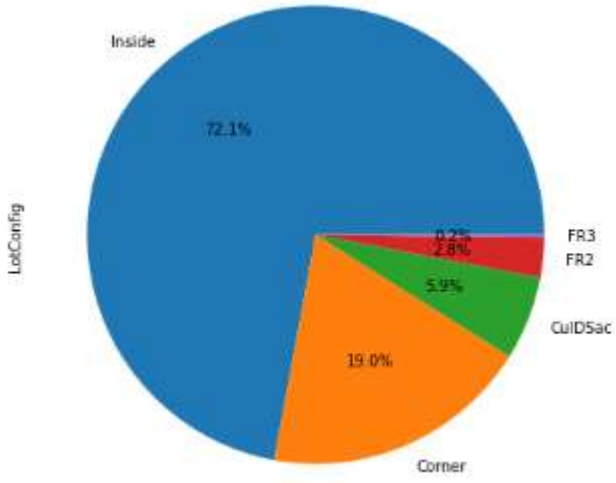
Observation:
63.4% house properties are regular in shape.
Sale Price of property with slight irregular shape is higher than regular shape.

Land Contour-wise of Property Distribution



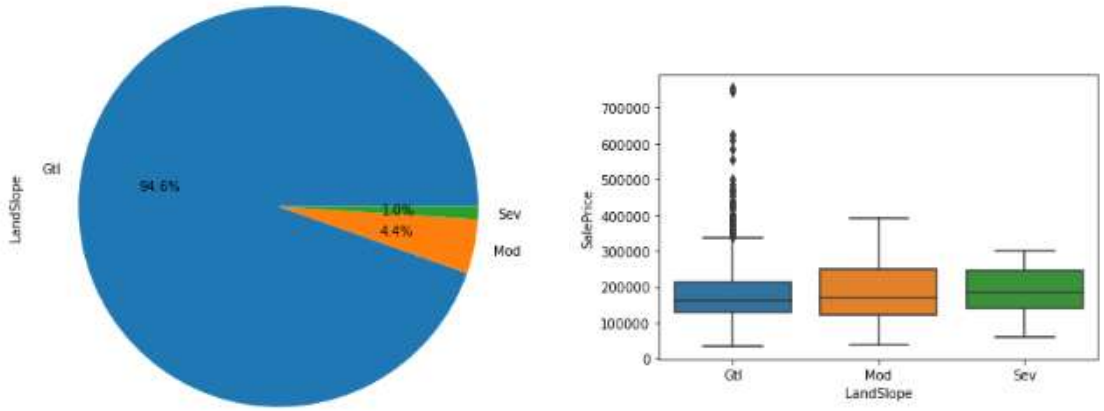
Observation:
89.6% of House properties are near flat level surface
Price for Flat level surface house is much higher than other land contour.

LotConfig of Property



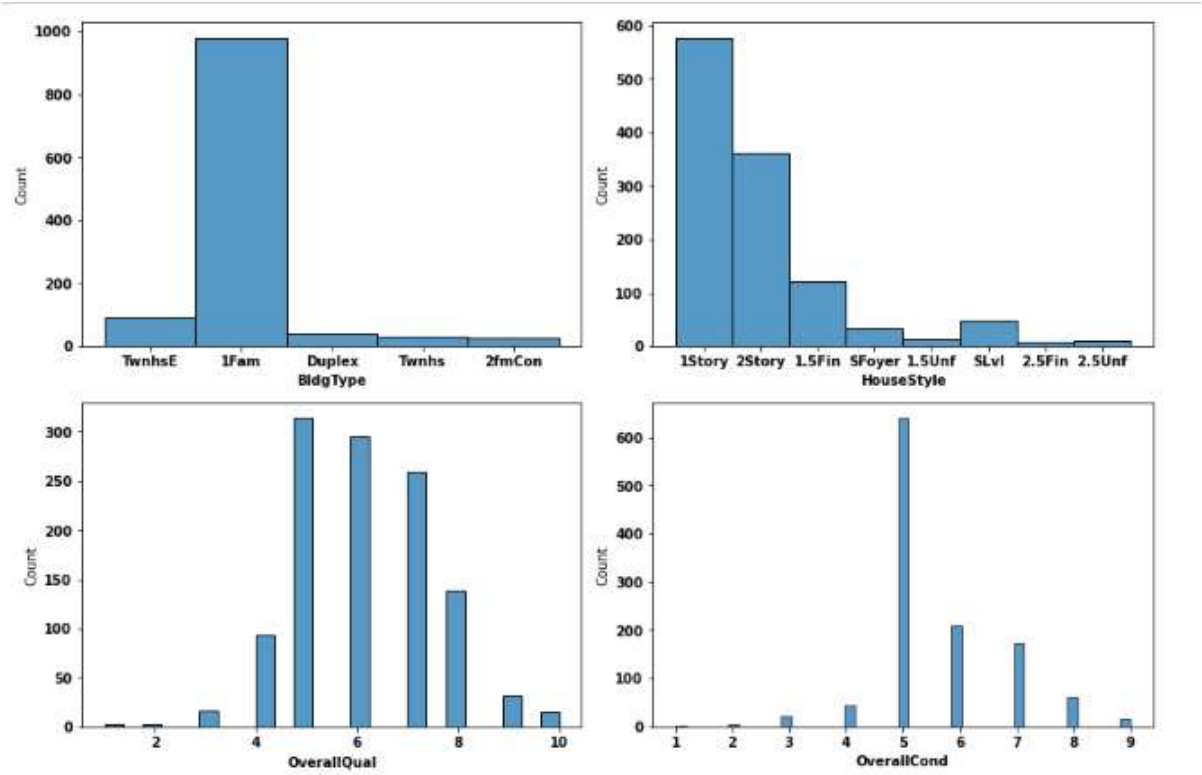
Observation:
Around 72 % of house comes with inside Lot configuration.
Cheapest Houses belong to Inside lot configuration while Costlier houses belongs to Corner Lot Configuration.

LandSlope of Property

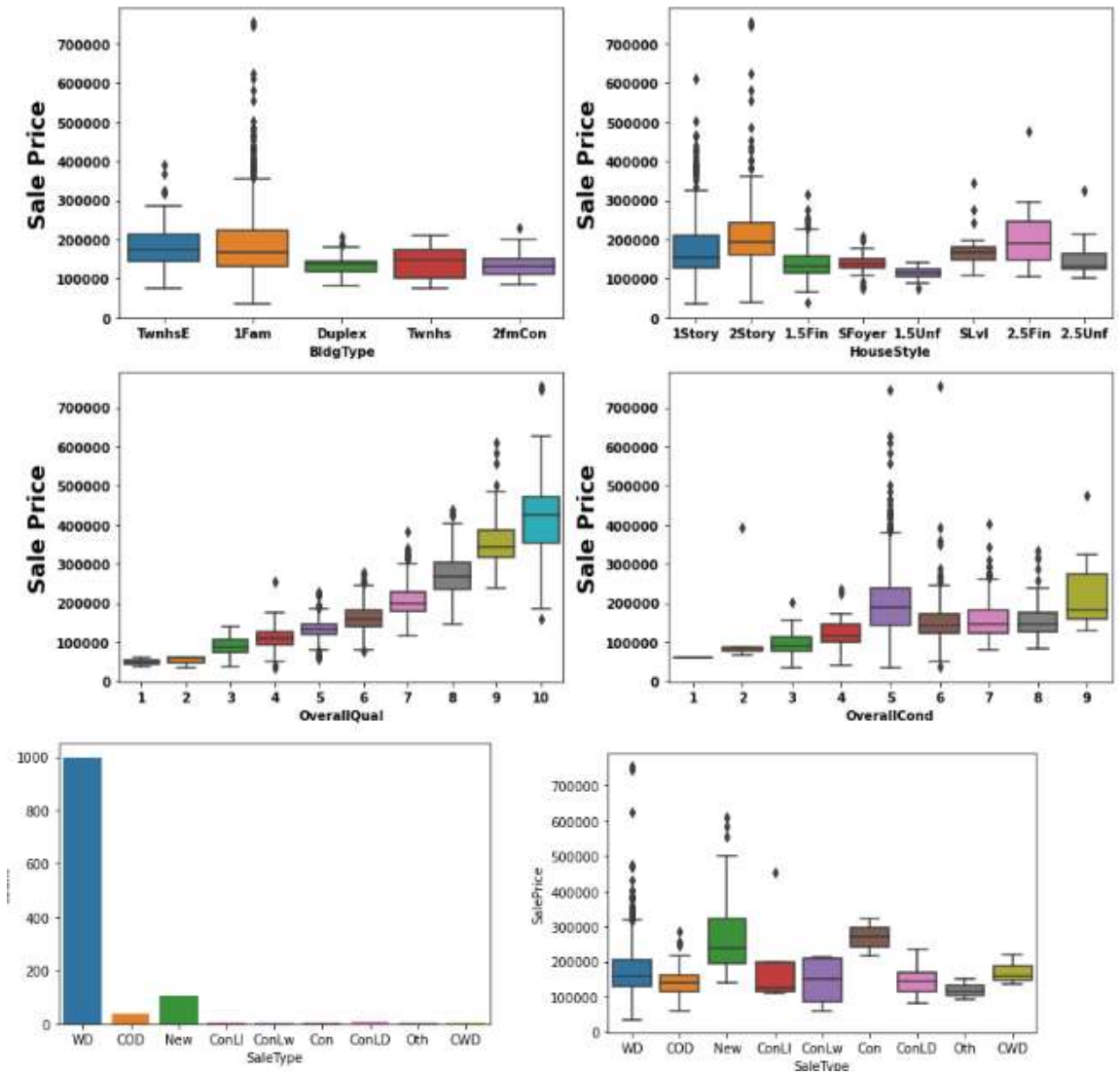


Observation:
we can see in boxplot that as Land slope increases the Sale price of house decreases.
1% properties come with severe slope and they come with low price compare to Gentle Slope properties.

Effect of Building Style on Sale Price



Observation:
More than 950 house properties are with building type Single-family Detached
More than 50% of house properties comes with Overall Condition Rating of 5.
More than 75% of house properties come with overall Quality Rating varies between 5 to 6.
More than 500 House Properties comes with one story dwelling.



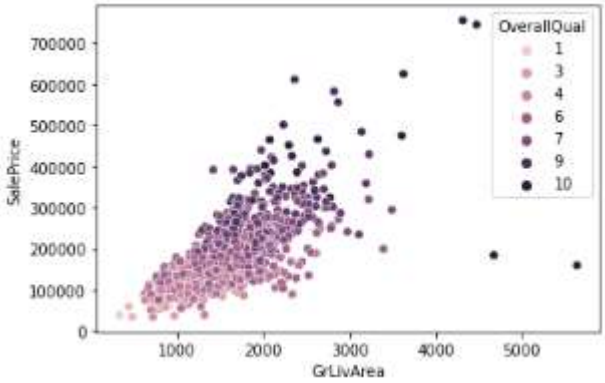
Observation:

Abnorml, Family, Alloca, AdjLand are below the price of 300000.

Maximum Base Price for House comes from Partial category- Home was not completed when last assessed (associated with New Homes) is higher than rest.

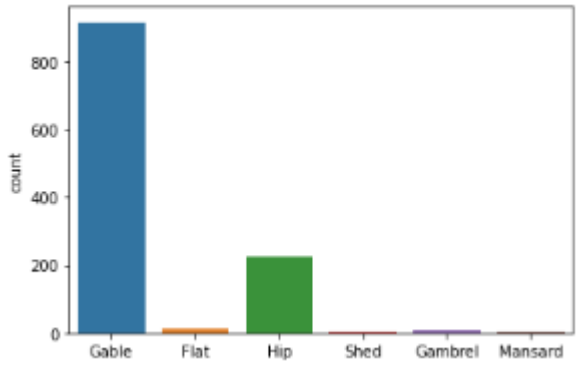
Minimum base price comes from Normal condition sale and also highest sale price comes from this category.

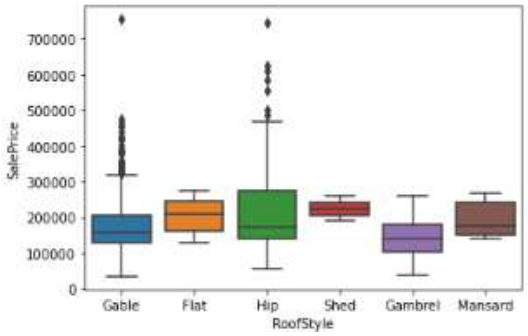
Sale Price Vs GrLivArea



Observation:
As total floor area increases the sale price also get increases corresponding the overall quality of House.

Exploring RoofStyle Type

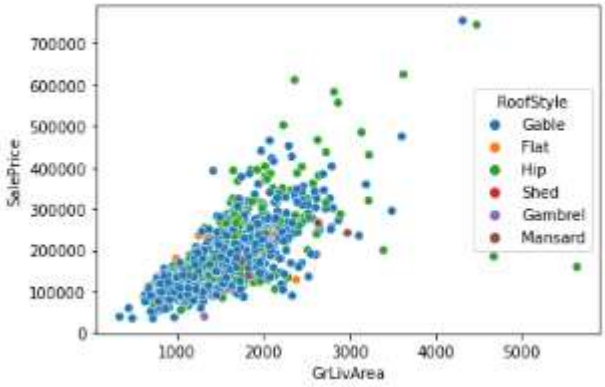




Observation:
More than 75% House properties come with Gable Roof Style followed by around 15 % house properties with Hip Style.

Hip style Roof are much costlier than remaining roof style.

Effect of RoofStyle & Total Floor Space on Sale Price

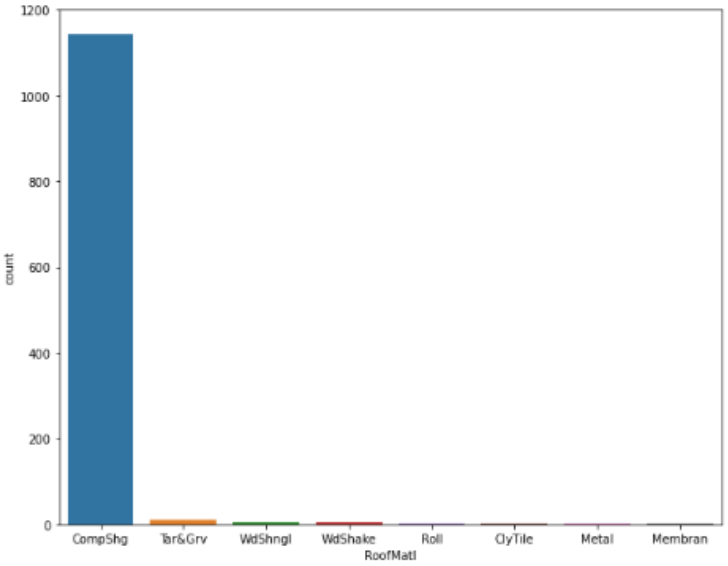


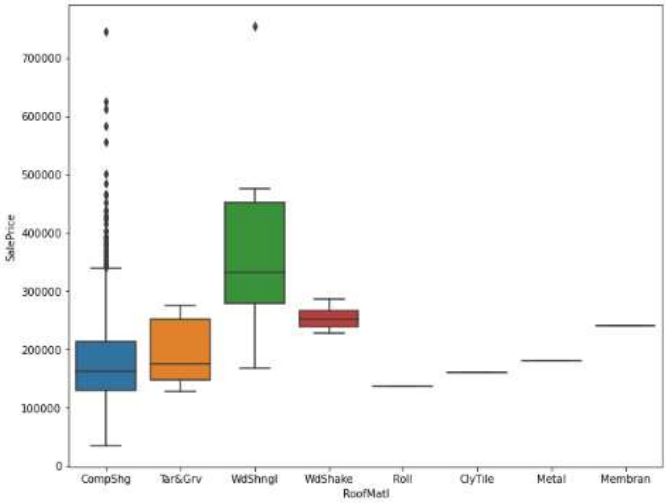
Observation:
High floor area construction mainly Hip style Roof is used and invariably high cost properties mostly comes up with Hip Style Roof.

RoofMatl Type

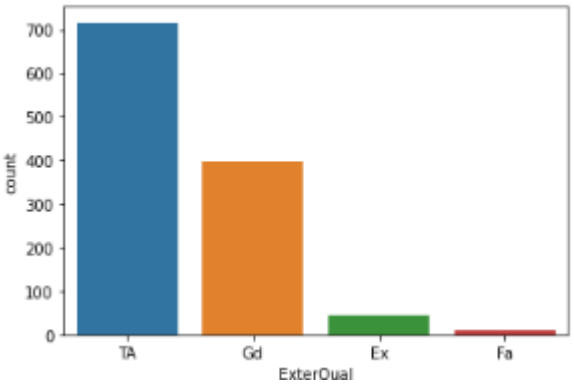
Observation:
90% Properties in Data set made with roof material of Standard (Composite) Shingle.

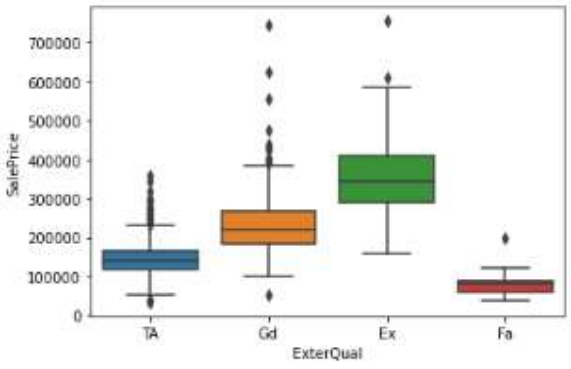
Wood Shingles is Costiler Material compare to rest.





Exterior Quality Description :-



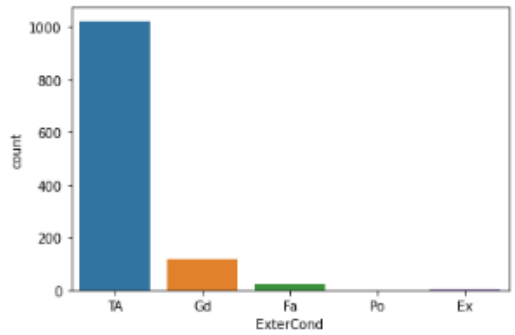


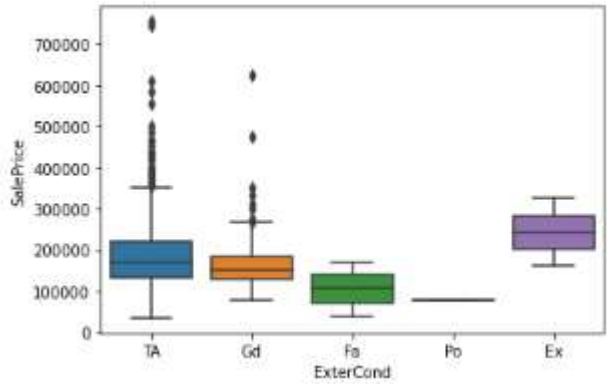
Observation:
Approx. 60% of house properties come with Average Exterior quality and all of them below 400000.

Very few House Properties comes with Excellent Exterior Quality.

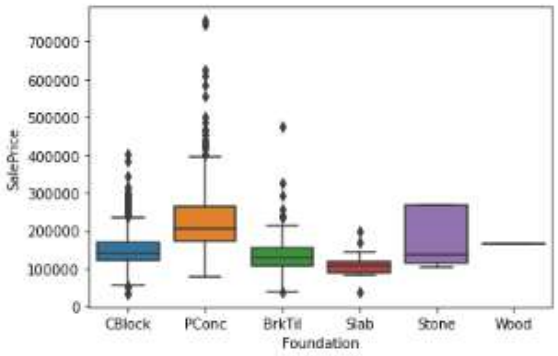
Costlier house properties come with Good & Excellent exterior quality

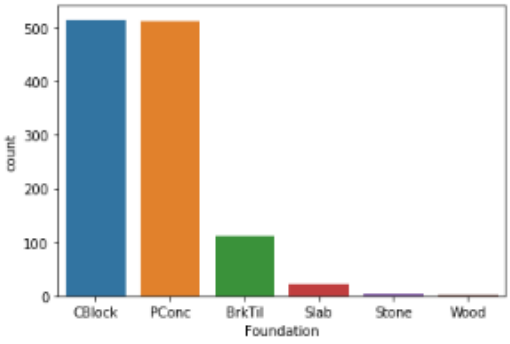
Exterior condition Type





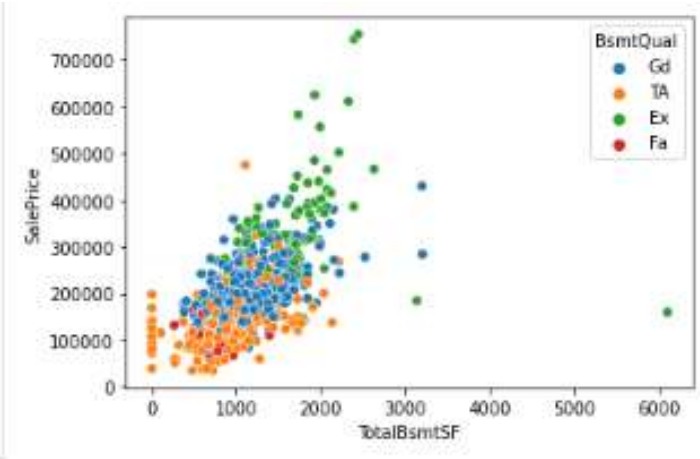
Foundation Vs Sale Price



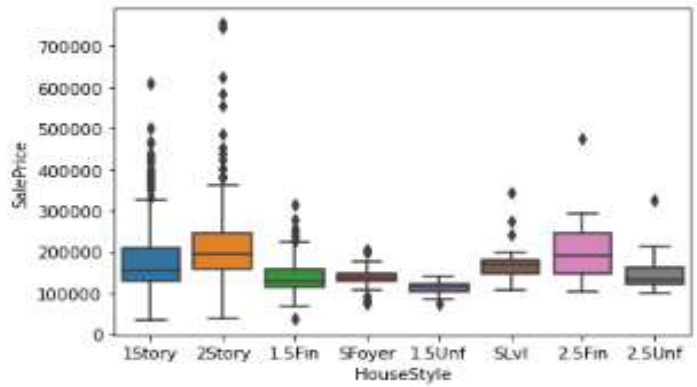


Observation:
Pconc Foundation are mostly use in costily housing properties.

Combine effect of RoofStyle & Total Floor Space on Sale Price



Observation:
As Basement Quality increase in relation to it, sale Price also get increases



Observation:
2 Story Building are costlier than other.

- Interpretation of the Results

Key Findings and Conclusions

Algorithm	R2 Score	CV Score
Random Forest Regressor	89.37	82.78
Linear Regression	87.55	76.89
Decision Tree Regressor	59.84	68.31
Ridge Regression	87.55	76.93
Random Forest Regressor Hyper Parameter Tuned Final Model	88.99	82.78

- Random Forest Regressor giving us maximum R2 Score, so Random Forest Regressor is selected as best model.
- After hyper parameter tuning Final Model is giving us R2 Score of 88.99% which is slightly improved compare to earlier R2 score of 82.78%.