Name Of The Project

Surprise Housing - Housing Price Prediction & Analysis Project

Submitted By

Mrs. Swati Amit Motugade

FlipRobo SME

Gulshana Chaudhari

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 analysis skills. Also, I want to express my huge gratitude to Ms. Gulshana Chaudhari Mam (SME, Flip Robo), she is the person who has helped me to get out of all the difficulties I faced while doing the project.

A huge thanks to “Data trained” who are the reason behind Internship at Fliprobo. Last but not least my parents who are there to support me at every step of my life.

References used 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.
6. Sifei Lu, Zengxiang Li, Zheng Qin, Xulei Yang, Rick Siow Mong Goh,” A Hybrid Regression Technique for House Prices Prediction”, @2017 IEEE, 2017 IEEE International Conference on Industrial Engineering & Engineering Management, Singapore DOI:10.1109/IEEM.2017.8289904
7. CH. Raga Madhuri, Anuradha G, M. Vani Pujitha “House Price Prediction Using Regression Techniques: A Comparative Study”, IEEE 6th International Conference on smart structures and systems ICSSS 2019

8. J.-G. Liu, X.-L. Zhang, and W.-P. Wu, “Application of fuzzy neural network for real estate prediction,” Advances in Neural Networks -ISNN 2006, vol. 3973, pp. 1187–1191, 2006.

9. H. Kusan, O. Aytekin, and I. Ozdemir, “*e use of fuzzy logic ¨ in predicting house selling price,” Expert Systems with Applications, vol. 37, no. 3, pp. 1808–1813, 2010.

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.

Chapter 1

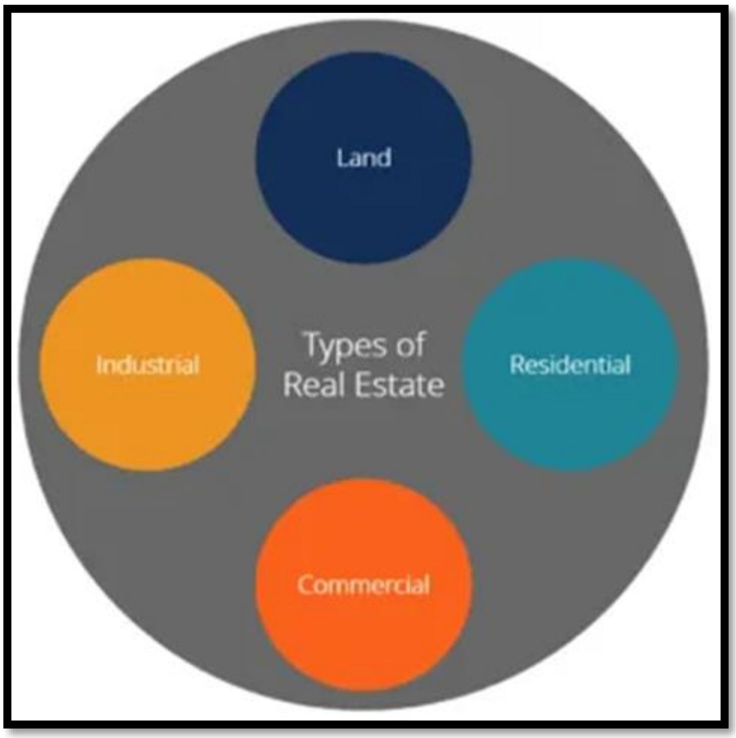
1.1 Introduction to Real Estate

Real estate is real property that consists of land and improvements, which include buildings, fixtures, roads, structures, and utility systems. Property rights give a title of ownership to the land, improvements, and natural resources such as minerals, plants, animals, water, etc.

Types of Real Estate

There are several types of real estate, each with a unique purpose and utility. The main categories are:

1. Land
2. Residential
3. Commercial
4. Industrial



Residential:

Residential real estate consists of housing for individuals, families, or groups of people. This is the most common type of estate and is the asset class that most people are familiar with. Within residential, there are single-family homes, apartments, condominiums, townhouses, and other types of living arrangements.

Here we've outlined the four main categories, let's explore some specific examples of different types of Residential real property.

- **Single-family dwelling** – Any home designed for only one family
- **Multi-family dwelling** – Any group of homes designed for more than one family
- **Attached** – Any unit that's connected to another (not freestanding)
- **Apartment** – An individual unit in a multi-unit building. The boundaries of the apartment are generally defined by a perimeter of locked or lockable doors. Often seen in multi-story apartment buildings.
- **Multi-family house** – Often seen in multi-story detached buildings, where each floor is a separate apartment or unit.
- **Multi-family house** – Often seen in multi-story detached buildings, where each floor is a separate apartment or unit.
- **Condominium (Condo)** – A building with individual units owned by individual people.
- **Detached house** – A free-standing building not connecting to anything else (a stereotypical "home")
- **Portable house** – Houses that can be moved on a flatbed truck
- **Mobile home** – A vehicle on wheels that has a permanent residence attached to it
- **Villa** – A building with only one room and typically a steep pointy roof
- **Hut** – A dwelling typically made of raw materials such as bamboo, mud, and clay

1.2 Business Problem Framing

Real Estate Property is not only the basic need of a man but today it also represents the riches and prestige of a person. Investment in real estate generally seems to be profitable because their property values do not decline rapidly. The market demand for housing is always increasing every year due to increase in population and migrating to other cities for their financial purpose. Changes in the real estate price can affect various household investors, bankers, policy makers and many. Investment in Housing seems to be an attractive choice for the Investors.

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. 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.

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:

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.

1.3 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 . The company is looking at prospective properties to buy houses to enter the market. We are required to build a Regression model using Machine Learning in order to predict the actual value of the prospective properties and decide whether to invest in them or not.

The Company wants to know

1. Which variables are important to predict the price of variable?
2. 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.

1.4 Review of Literature

Related Work or Literature survey is the most important step in any kind of research. 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.

Sifei Lu, Zengxiang Li, Zheng Qin, Xulei Yang, Rick Siow Mong Goh [6] had proposed an advanced house prediction system using linear regression. This system's aim was to make a model that can give us a good house price prediction based on other variables. This paper proposed on Hybrid Regression technique for housing Prices Prediction focused on the use of creative feature engineering to find the optimal features and their correlation with Sales Prices. Feature engineering improved the data normality and linearity of data. Their system showed that working on the Ames Housing dataset was convenient and showed that the use of Hybrid algorithms (65% Lasso and 35% Gradient Boost) provided results in predicting the house prices rather than using one from lasso, ridge or gradient boost.

CH.Raga Madhuri, Anuradha G et.al [7], 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.

Liu et al. [10] 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 the paper proposed by Ayush Varma et. al. [12] suggested that the use of neural networks along with linear and boosted algorithms improved prediction accuracy. Three algorithms were used namely Linear Regression, Forest Regression and Boosted Regression. The dataset was tested on all three and the results of all the above algorithms were fed as an input to the neural network. Neural networks algorithms were fed as an input to the neural network. Neural networks accurate result. A neural network along with Boosted Regression was used to increase the accuracy of the result.

Another research study showed that there exist relationships between the visual appearance and non-visual attributes such as crime statistics, housing prices, population density, etc. of a city. For instance, “City Forensics: Using Visual Elements to Predict Non-Visual City “City Forensics: Using Visual Elements to Predict Non-Visual City Attributes” [13], uses visual attributes to predict the sale price of the property.

1.5 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, XGBoost, Random Forest regression and Extra Tree Regressor 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. These features may or may not be shared with all houses, which means they do not have the same influence on the house pricing resulting in inaccurate output.

The study answers the following research questions:

1. Research question 1: Which machine learning algorithm performs better and has the most accurate result in house price prediction? And why?
2. Research question 2: What are the factors that have affected house prices in Australia over the years?

Chapter 2

Analytical Problem Framing

Mathematical Modelling of the Problem

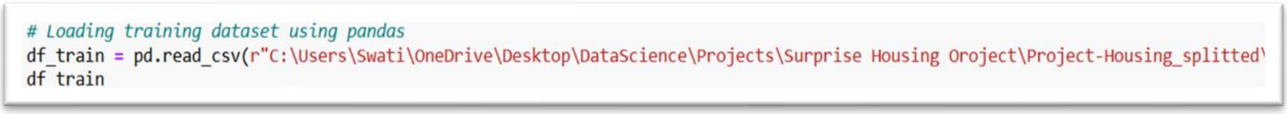
Our objective is to predict House price which can be resolve by use of regression-based algorithm. In this project we are going to use different types of algorithms which uses their own mathematical equation on background. This project comes with two separate data set for training & testing model. Initially 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.

2.2 Data Sources and their formats

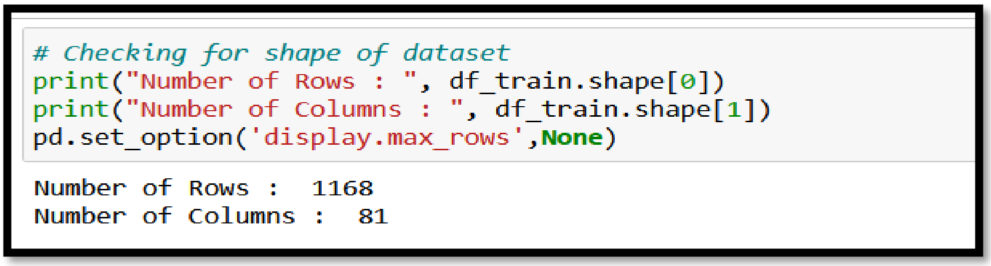
Data set provided by Flip Robo was in the format of CSV (Comma Separated Values).

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.

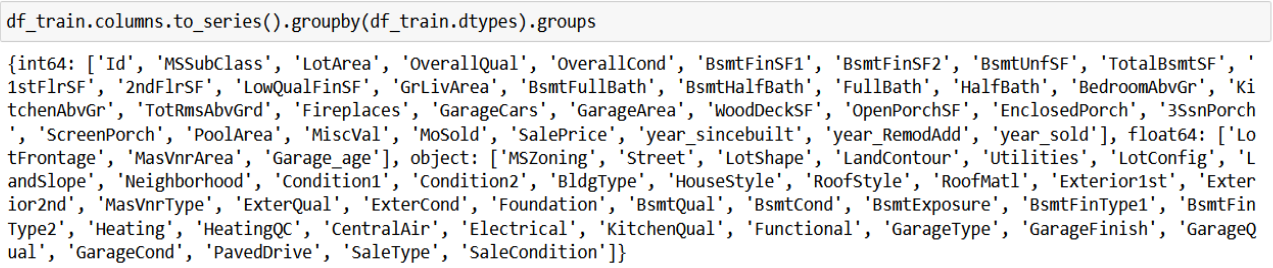


```
# Loading training dataset using pandas
df_train = pd.read_csv(r"C:\Users\Swati\OneDrive\Desktop\DataScience\Projects\Surprise Housing Oroject\Project-Housing_splitted\
df_train
```



```
# Checking for shape of dataset
print("Number of Rows : ", df_train.shape[0])
print("Number of Columns : ", df_train.shape[1])
pd.set_option('display.max_rows',None)
```

```
Number of Rows :  1168
Number of Columns :  81
```



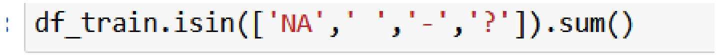
```
df_train.columns.to_series().groupby(df_train.dtypes).groups
```

```
{int64: ['Id', 'MSSubClass', 'LotArea', 'OverallQual', 'OverallCond', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', '
1stFlrSF', '2ndFlrSF', 'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'HalfBath', 'BedroomAbvGr', 'Ki
tchenAbvGr', 'TotRmsAbvGrd', 'Fireplaces', 'GarageCars', 'GarageArea', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch
', 'ScreenPorch', 'PoolArea', 'MiscVal', 'MoSold', 'SalePrice', 'year_sincebuilt', 'year_RemodAdd', 'year_sold'], float64: ['Lo
tFrontage', 'MasVnrArea', 'Garage_age'], object: ['MSZoning', 'Street', 'LotShape', 'LandContour', 'Utilities', 'LotConfig', 'L
andSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType', 'HouseStyle', 'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exter
ior2nd', 'MasVnrType', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFin
Type2', 'Heating', 'HeatingQC', 'CentralAir', 'Electrical', 'KitchenQual', 'Functional', 'GarageType', 'GarageFinish', 'GarageQ
ual', 'GarageCond', 'PavedDrive', 'SaleType', 'SaleCondition']}
```

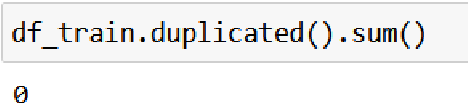
2.3.Data Pre-processing

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.



```
df_train.isin(['NA',' ','-','?']).sum()
```

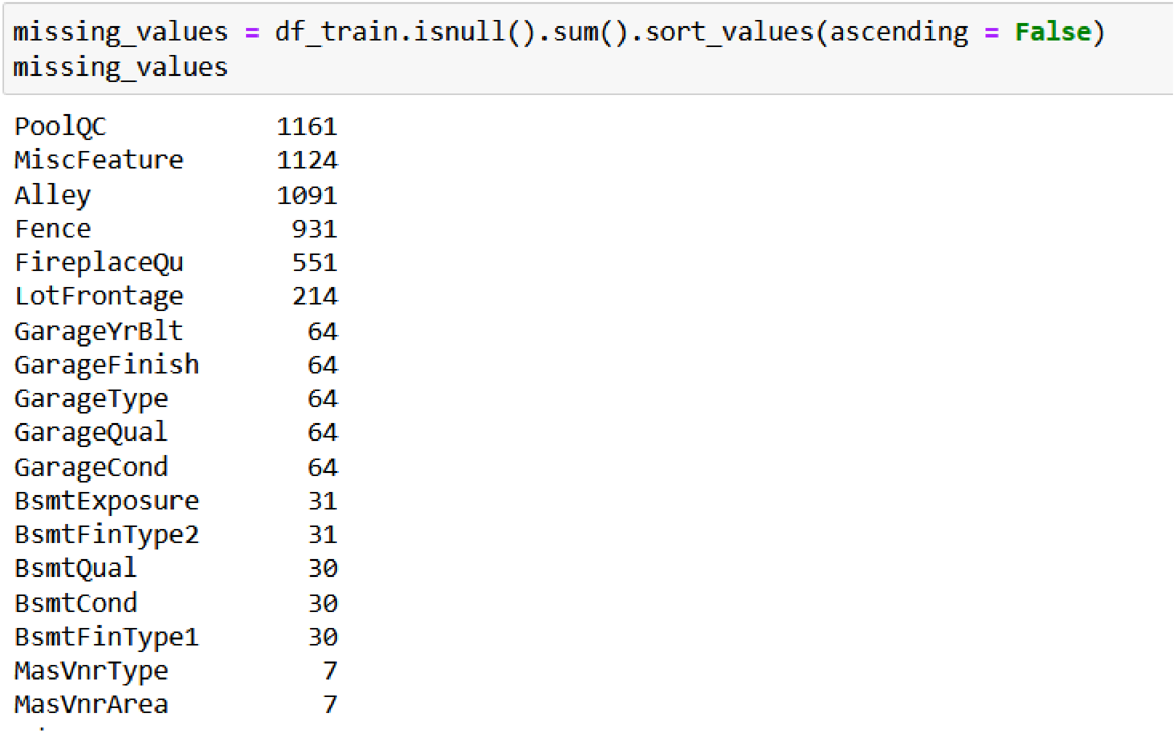


```
df_train.duplicated().sum()

0
```

No duplicate entries present in dataset.

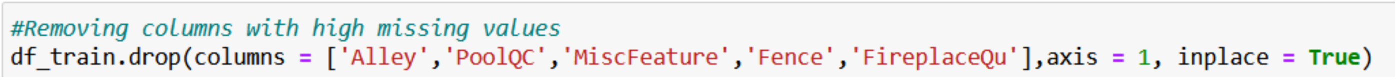
Some features contain missing values as shown below:



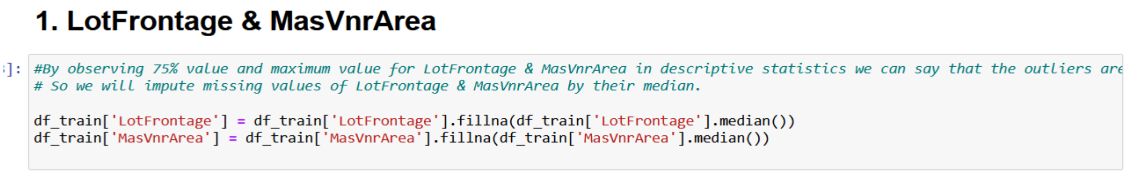
```
missing_values = df_train.isnull().sum().sort_values(ascending = False)
missing_values

PoolQC          1161
MiscFeature     1124
Alley           1091
Fence            931
FireplaceQu      551
LotFrontage      214
GarageYrBlt       64
GarageFinish      64
GarageType        64
GarageQual        64
GarageCond        64
BsmtExposure      31
BsmtFinType2      31
BsmtQual          30
BsmtCond          30
BsmtFinType1      30
MasVnrType         7
MasVnrArea         7
```

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.



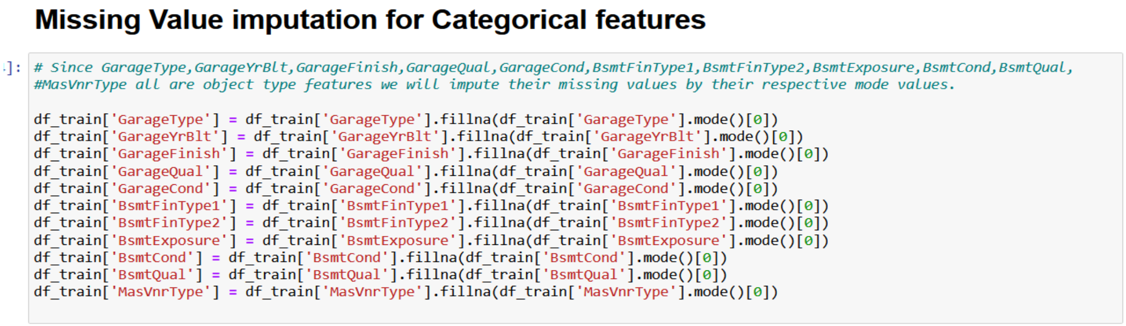
```
#Removing columns with high missing values
df_train.drop(columns = ['Alley','PoolQC','MiscFeature','Fence','FireplaceQu'],axis = 1, inplace = True)
```



1. LotFrontage & MasVnrArea

```python
#By observing 75% value and maximum value for LotFrontage & MasVnrArea in descriptive statistics we can say that the outliers are
# So we will impute missing values of LotFrontage & MasVnrArea by their median.

df_train['LotFrontage'] = df_train['LotFrontage'].fillna(df_train['LotFrontage'].median())
df_train['MasVnrArea'] = df_train['MasVnrArea'].fillna(df_train['MasVnrArea'].median())
```

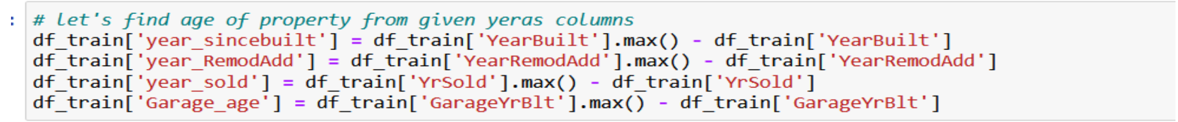


Missing Value imputation for Categorical features

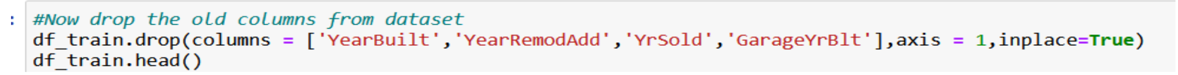
```python
# Since GarageType,GarageYrBlt,GarageFinish,GarageQual,GarageCond,BsmtFinType1,BsmtFinType2,BsmtExposure,BsmtCond,BsmtQual,
#MasVnrType all are object type features we will impute their missing values by their respective mode values.

df_train['GarageType'] = df_train['GarageType'].fillna(df_train['GarageType'].mode()[0])
df_train['GarageYrBlt'] = df_train['GarageYrBlt'].fillna(df_train['GarageYrBlt'].mode()[0])
df_train['GarageFinish'] = df_train['GarageFinish'].fillna(df_train['GarageFinish'].mode()[0])
df_train['GarageQual'] = df_train['GarageQual'].fillna(df_train['GarageQual'].mode()[0])
df_train['GarageCond'] = df_train['GarageCond'].fillna(df_train['GarageCond'].mode()[0])
df_train['BsmtFinType1'] = df_train['BsmtFinType1'].fillna(df_train['BsmtFinType1'].mode()[0])
df_train['BsmtFinType2'] = df_train['BsmtFinType2'].fillna(df_train['BsmtFinType2'].mode()[0])
df_train['BsmtExposure'] = df_train['BsmtExposure'].fillna(df_train['BsmtExposure'].mode()[0])
df_train['BsmtCond'] = df_train['BsmtCond'].fillna(df_train['BsmtCond'].mode()[0])
df_train['BsmtQual'] = df_train['BsmtQual'].fillna(df_train['BsmtQual'].mode()[0])
df_train['MasVnrType'] = df_train['MasVnrType'].fillna(df_train['MasVnrType'].mode()[0])
```

- Feature extraction for age related feature.

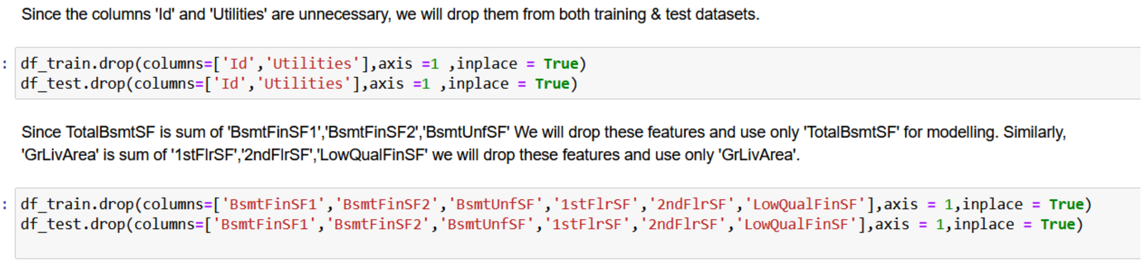


```python
# Let's find age of property from given yeras columns
df_train['year_sincebuilt'] = df_train['YearBuilt'].max() - df_train['YearBuilt']
df_train['year_RemodAdd'] = df_train['YearRemodAdd'].max() - df_train['YearRemodAdd']
df_train['year_sold'] = df_train['YrSold'].max() - df_train['YrSold']
df_train['Garage_age'] = df_train['GarageYrBlt'].max() - df_train['GarageYrBlt']
```



```python
#Now drop the old columns from dataset
df_train.drop(columns = ['YearBuilt','YearRemodAdd','YrSold','GarageYrBlt'],axis = 1,inplace=True)
df_train.head()
```

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

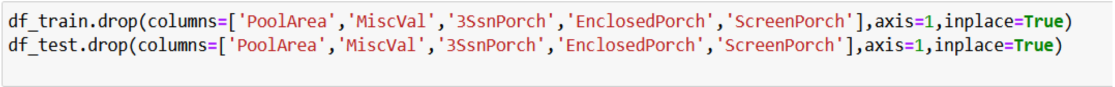


Since the columns 'Id' and 'Utilities' are unnecessary, we will drop them from both training & test datasets.

```python
df_train.drop(columns=['Id','Utilities'],axis =1 ,inplace = True)
df_test.drop(columns=['Id','Utilities'],axis =1 ,inplace = True)
```

Since TotalBsmtSF is sum of 'BsmtFinSF1','BsmtFinSF2','BsmtUnfSF' We will drop these features and use only 'TotalBsmtSF' for modelling. Similarly, 'GrLivArea' is sum of '1stFlrSF','2ndFlrSF','LowQualFinSF' we will drop these features and use only 'GrLivArea'.

```python
df_train.drop(columns=['BsmtFinSF1','BsmtFinSF2','BsmtUnfSF','1stFlrSF','2ndFlrSF','LowQualFinSF'],axis = 1,inplace = True)
df_test.drop(columns=['BsmtFinSF1','BsmtFinSF2','BsmtUnfSF','1stFlrSF','2ndFlrSF','LowQualFinSF'],axis = 1,inplace = True)
```

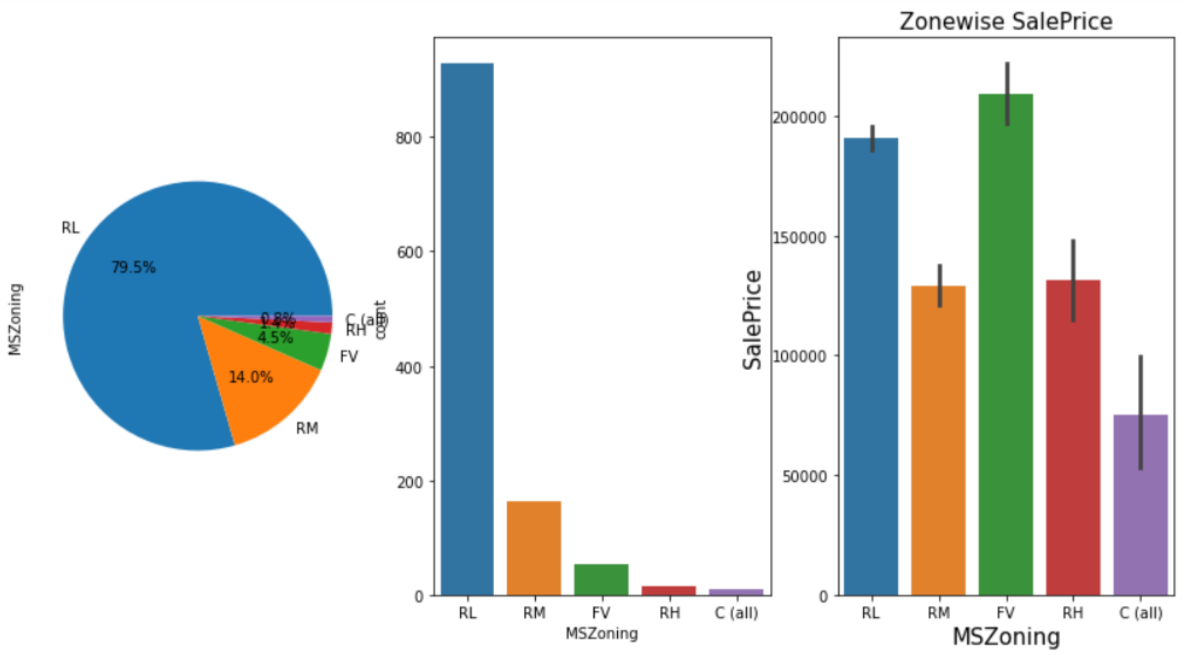


```python
df_train.drop(columns=['PoolArea','MiscVal','3SsnPorch','EnclosedPorch','ScreenPorch'],axis=1,inplace=True)
df_test.drop(columns=['PoolArea','MiscVal','3SsnPorch','EnclosedPorch','ScreenPorch'],axis=1,inplace=True)
```

Chapter 3

VISUALIZATIONS

1. Zonewise Property Distribution



Observations :

1. Majority of properties belongs to Low Density Residential area(RL) which is 79.5%.

2. 14% properties belongs to Medium density Residential area(RM).

3. 4.5% properties belongs to Floating Village Residential area.

4. 1.4 % properties belongs to High density Residential area(RH).

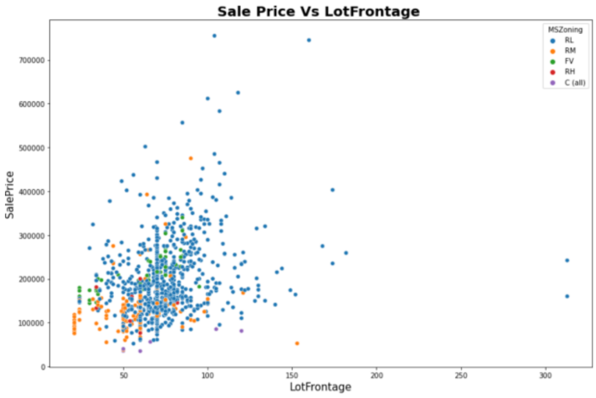
5. Only 0.8% properties belongs to Commercial zone.

6. Highest SalePrice is for Floating Village Residential zone followed by Low Density Residential zone.

7. SalePrices for High density Residential zone and Medium density Residential zone are nearly same.

8. The Lowest Sale Price is for Commercial zone.

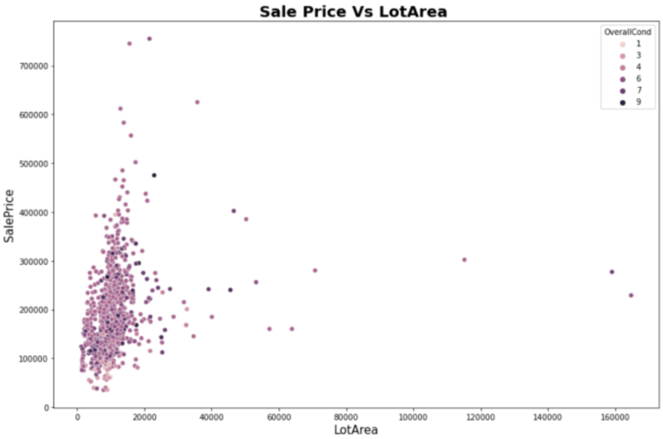
2.LotFrontage



Observations:

Sale price increases with increase in LotFrontage area except for Commercial zone

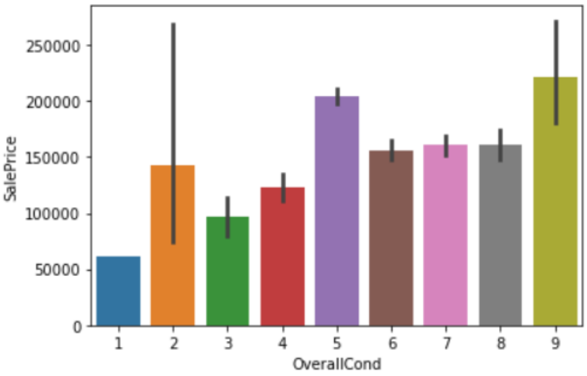
3.LotArea



Observations :

No significant relationship seen between LotArea and SalePrice

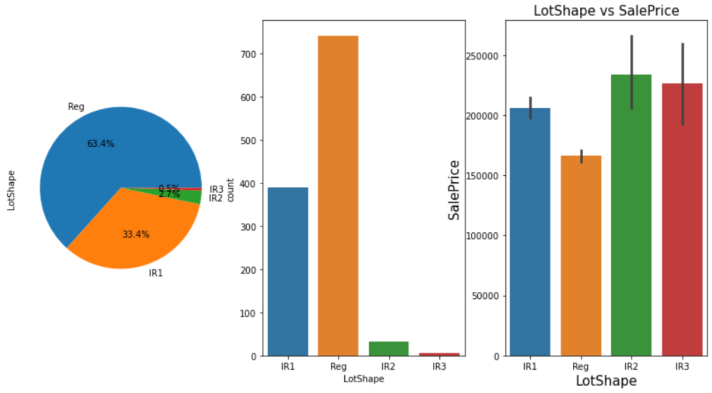
3. Overall Condition vs SalePrice



Observations :

1. The highest saleprice is for overall cindition rating 9.
2. Secondly highest saleprice is for overall cindition rating 5.
3. Lowest saleprice is for overall cindition rating 1.
4. Here we can see that the saleprice is less for overall condition rating 6,7 & 8 as compared with overall condition rating 5 which means that Overall condition rating is not that much affecting feature for SalePrice.

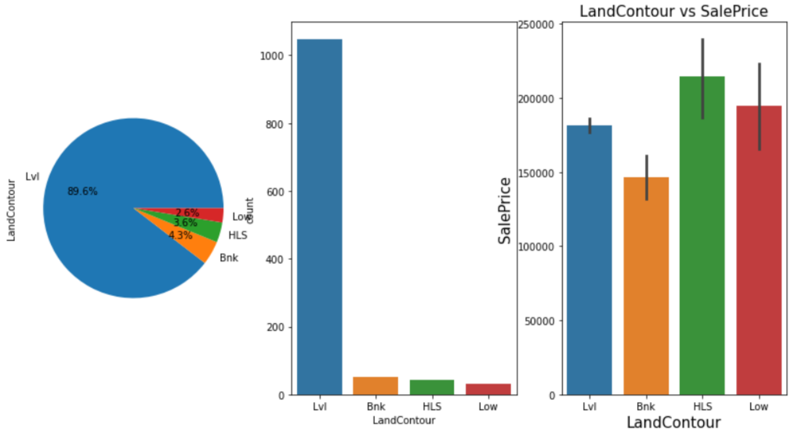
5. LotShape



Observations :

1. Maximum number of the properties are of Regular property shape.
2. Second highest number of properties are of Slightly Irregular shape.
3. Very less number of properties are of Irregular shape.
4. the highest saleprice is for properties with Moderately irregular shape folloed by properties with Irregular shape.
3. Lowest sale price is for properties with Regular shape.

6. LandContour



Observations :

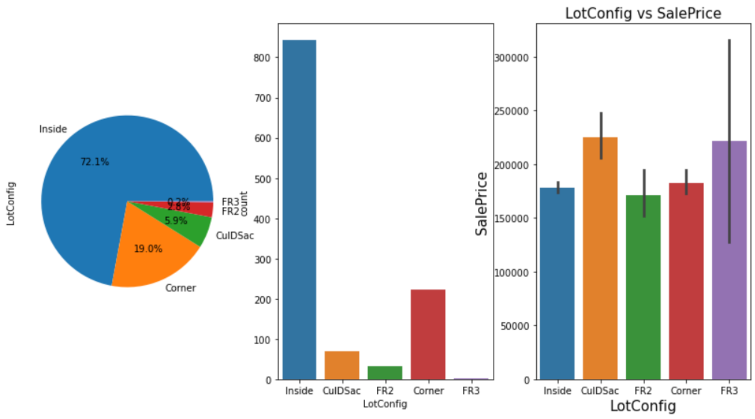
1. Near about 90% properties are with Level or flat LandContour type.

2. Remaining 10% properties are of Banked, Hillside and Low type LandContour.

3. SalePrice for Hillside Properties are much greater than other LandContour.

4. Lowest SalePrice is for Banked LandContour properties.

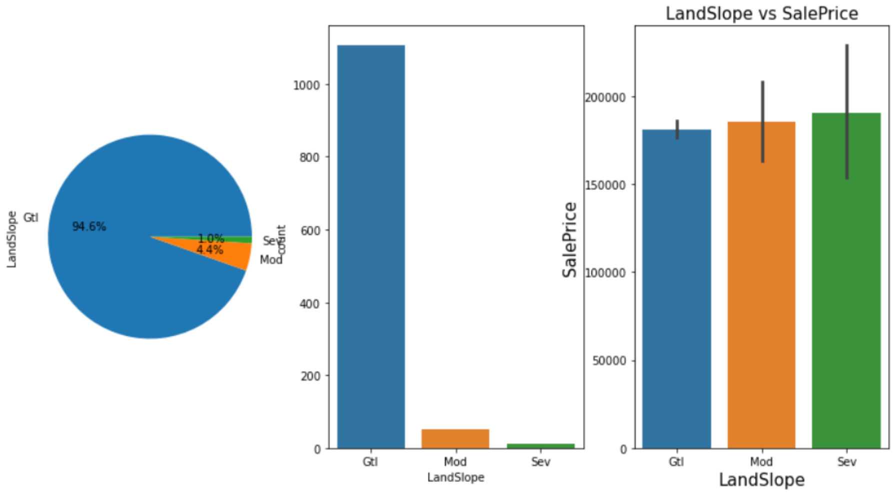
7. LotConfig



Observations :

1. Near about 72% properties are with Inside Lot configuration.
2. Very few properties are with FR3 configuration.
3. Properties with CulDSac configuration has maximum selling price followed by properties with FR3 configuration.
4. FR2 configuration properties has less sale price.

8. LandSlope

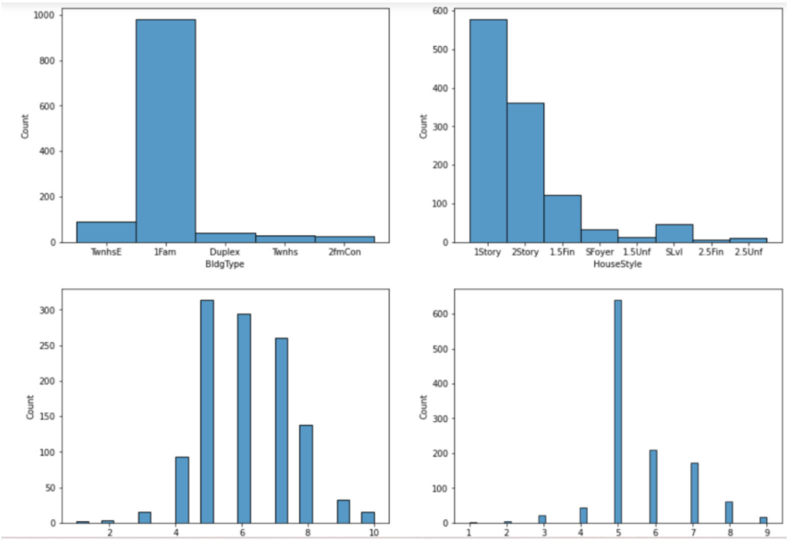


Observations :

1. Near about 95% properties has Gentle slope.
2. 4.4% properties are with moderate slope.
3. Only 1% properties has Severe slope.
4. Sale price is maximum for properties of Severe type slope.
5. Sale price is minimum for properties of Gentle type slope.

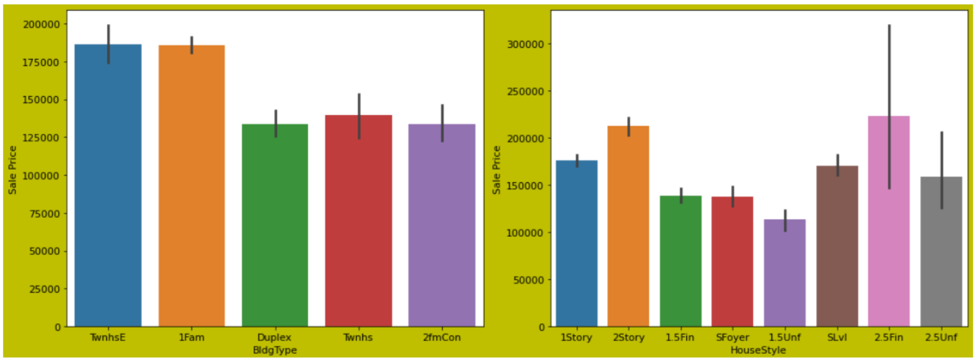
9. Effect of Building Style

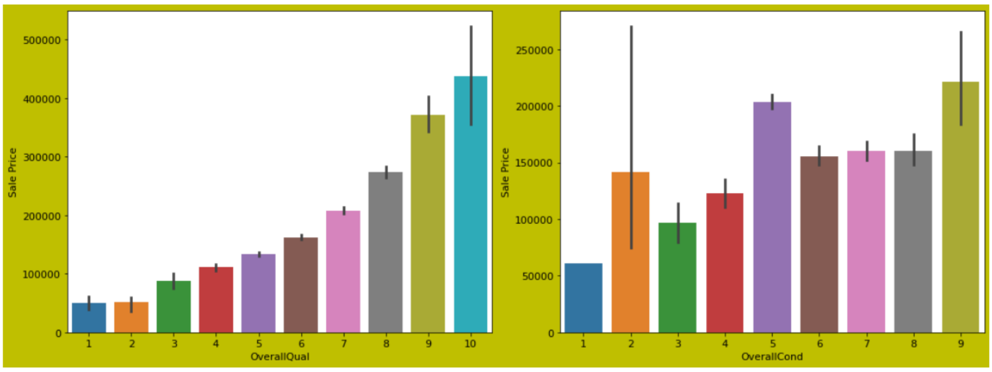
The features "BldgType", "HouseStyle", "OverallQual", "OverallCond" are of Same category, so we will combine them into one group.



Observations :

1. 90% of the properties are of Building type Single-family Detached.
2. There are very less properties with building type Two-family Conversion and Townhouse End Unit.
3. More than 50% of house properties comes with Overall Condition Rating of 5.
4. More than 75% of house properties come with overall Quality Rating varies between 5 to 6.



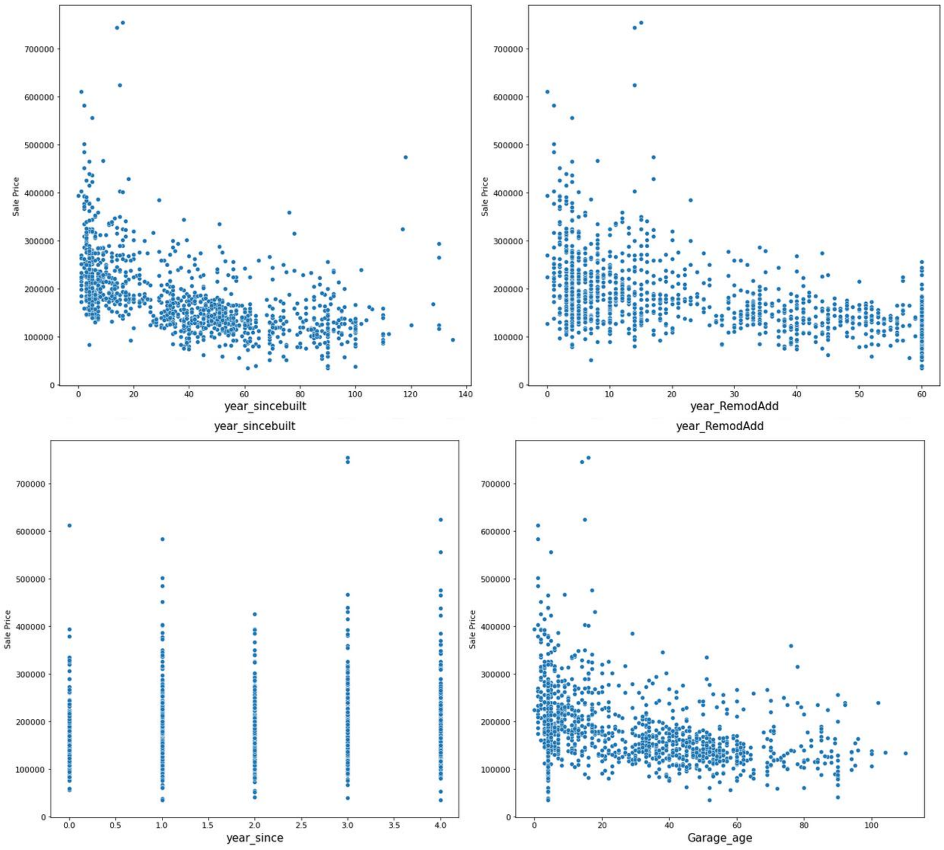


Observations:

As Overall Quality increases the sale price also increase with it

10.Age_features

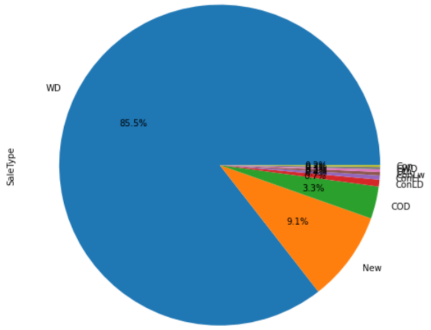
Age_features = ["year_sincebuilt", "year_RemodAdd", 'year_since', "Garage_age"]

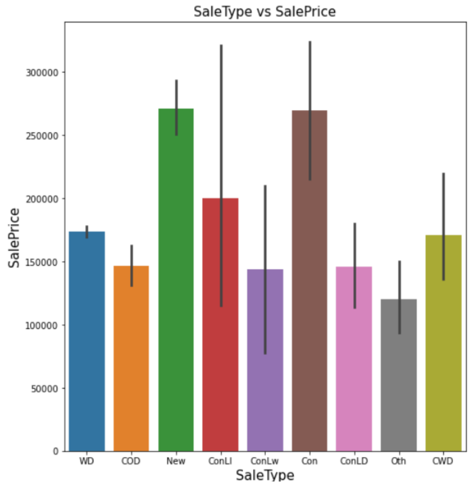


Observations :

1. We can see that as Property get older with time its sale Price get depricates.
2. 20 years after Remodelling Price of properties start decreases.
3. Older the garage age less the price of Property.

11.SaleType

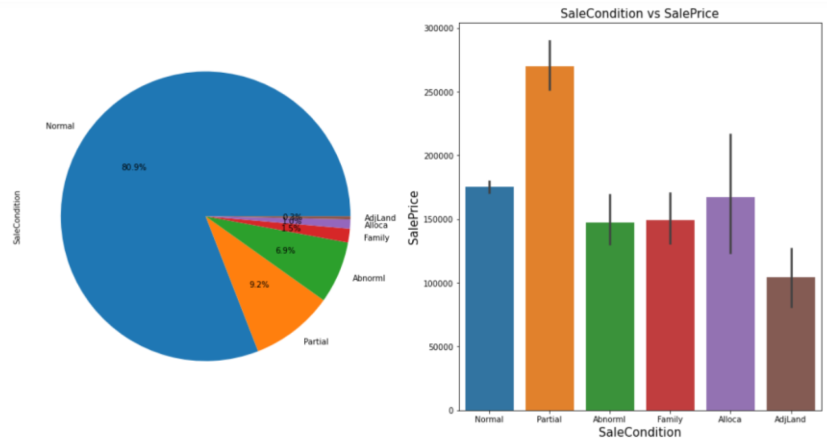




Observations :

1. About 85% properties are sold by Conventional Warranty Deed.
2. There are very less number of properties with sale type Contract 15% Down payment regular terms.
3. The sale prices are more for Homes just constructed & Sold and for Contract 15% Down payment regular terms.
4. The sale price is minimum for Other sale type properties.

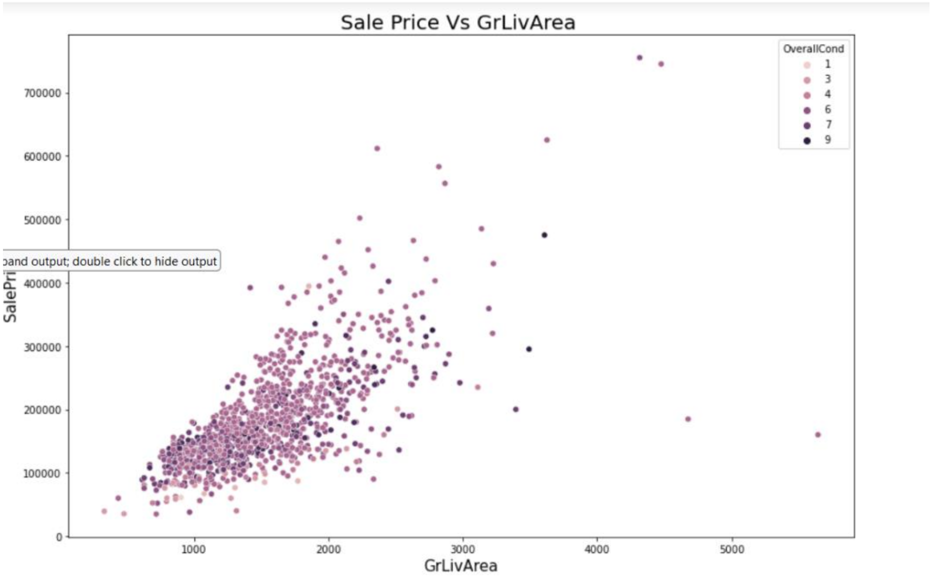
12.Condition of sale



Observations :

1. More than 80% properties are with Normal sale condition.
2. 9.2 properties are with Partial sale condition.
3. Nearly 7% properties are with Abnormal sale condition.
4. 1.5% properties are with Family sale condition.
5. 1% properties are with Alloca sale condition.
6. Only 0.3% properties are with AdjLand sale condition.
7. Highest SalePrice is for Partial salecondition.
8. Lowest saleprice is for Adjoining Land Purchase sale condition.

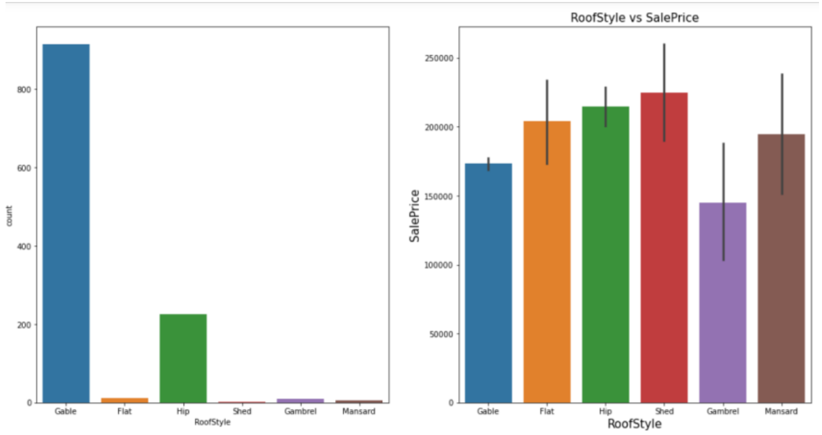
12. GrLivArea



Observations :

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

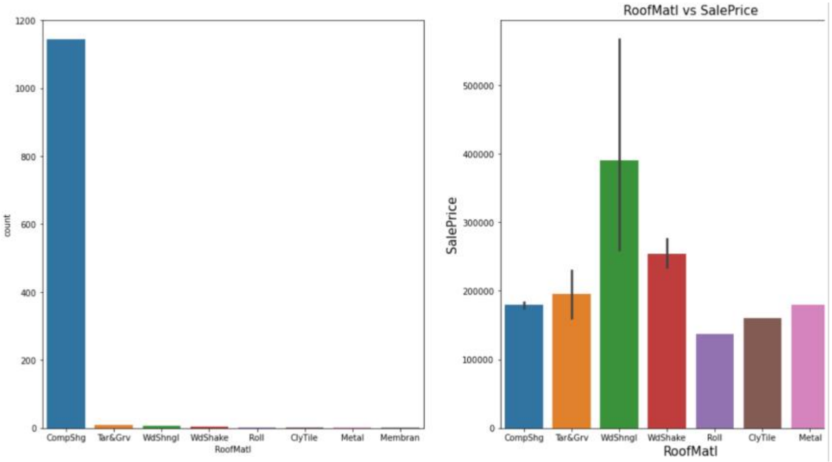
13.RoofStyle



Observations

1. 78% properties are of Gable roof style.
2. 19% properties are of Hip roof style.
3. 1% properties are of Flat roof style.
4. 0.77% properties are of Gambrel roof style.
5. 0.42% properties are of Mansard roof style.
6. Only 0.17% properties are of Shed roof style.

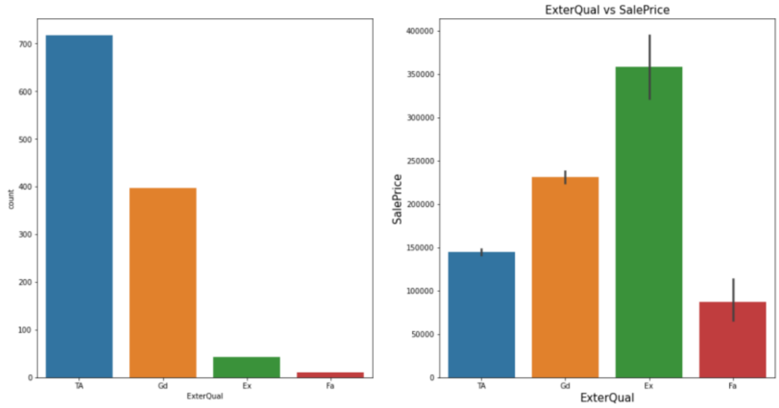
14. RoofMatl



Observations :

1. About 98% of properties are with Standard (Composite) Shingle floor material.
2. Very less number of properties has roof material Roll,Wood Shakes,Wood Shingles and Clay or Tile.
3. Maximum sale price is for properties with Wood Shingles roof material.
4. Minimum sale price is for properties with Roll roof material.

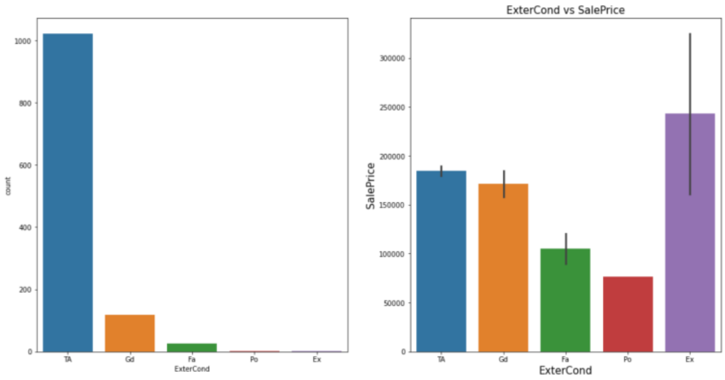
15. ExterQual



Observations :

1. Near about 61% properties are of Average/Typical Exterior quality.
2. 33% properties are of Good Exterior quality.
3. 3.6% properties are of Excellent Exterior quality.
4. Only 1% properties are of Fair Exterior quality.
5. Sale price is High for Excellent Exterior quality.
6. Minimum sale price is for Fair Exterior quality.
7. There are no properties with Poor Exterior quality.

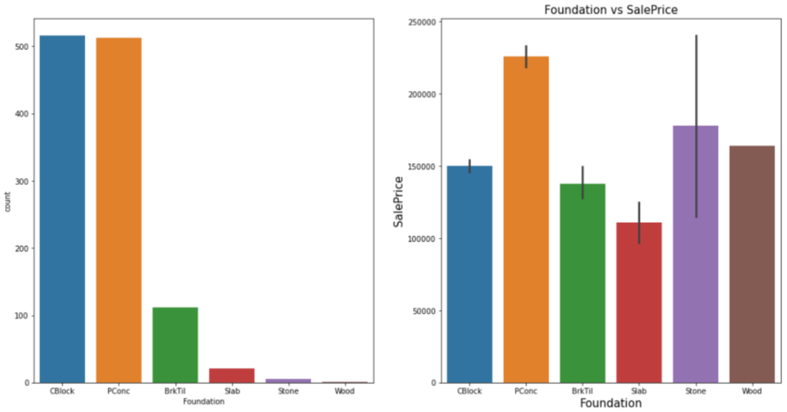
16. ExterCond



Observations :

1. Near about 87% properties are of Average/Typical Exterior Condition.
2. 10% properties are of Good Exterior Condition.
3. 2.2% properties are of Fair Exterior Condition.
4. 0.17% properties are of Excellent Exterior Condition.
5. 0.08% properties are of Poor Exterior Condition.
6. Sale price is High for Excellent Exterior Condition properties.
7. Minimum sale price is for Poor Exterior Condition properties.

17. Foundation

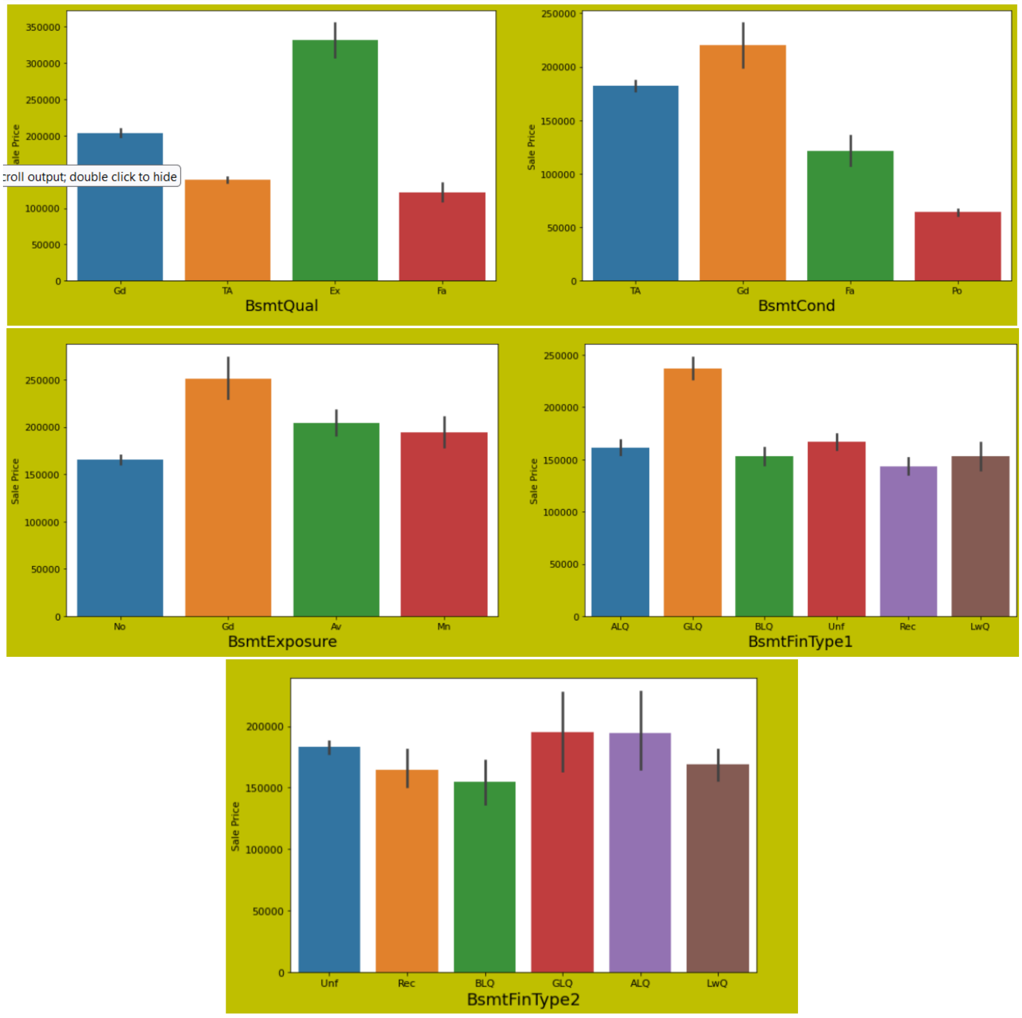


Observations :

1. There are 44% properties of Cinder Block and Poured Contrete foundation respectively.
2. 9.5% of properties are of Brick & Tile foundation.
3. 1.79% properties are of Slab foundation.
4. 0.42% properties are of Stone foundation.
5. Only 0.08% properties are of Wood foundation.
6. Maximum sale price is for Poured Contrete foundation properties.
7. Minimum sale price is for Slab foundation properties.

18. Basement features

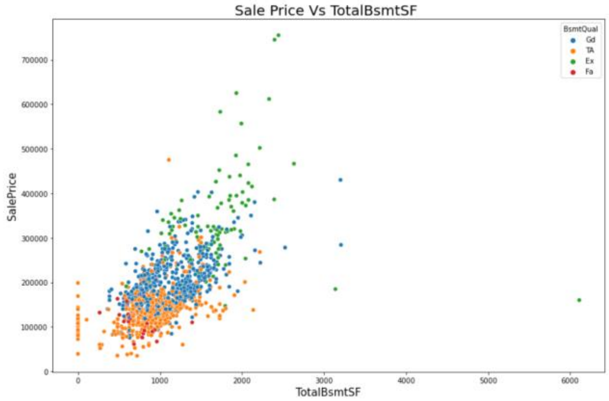
basement_features = ['BsmtQual','BsmtCond','BsmtExposure','BsmtFinType1','BsmtFinType2']



Observations :

1. More number of properties with Excellent Basement quality.
2. More number of properties with Good Basement Condition.
3. More number of properties with Good Basement Exposure.
4. More number of properties with Good Living Quarters Rating of basement finished area.
5. More number of properties with Good Living Quarters and Average Living Quarters Rating of basement finished area.

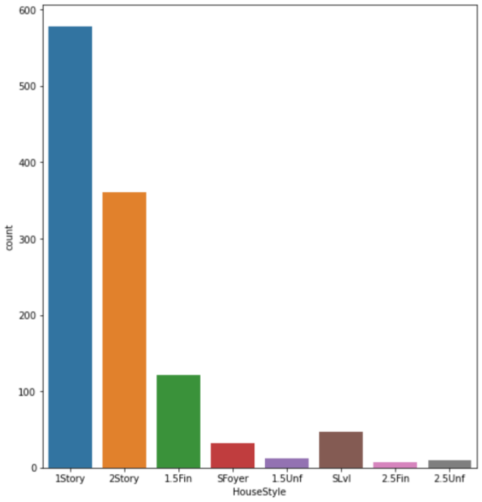
19. TotalBsmtSF

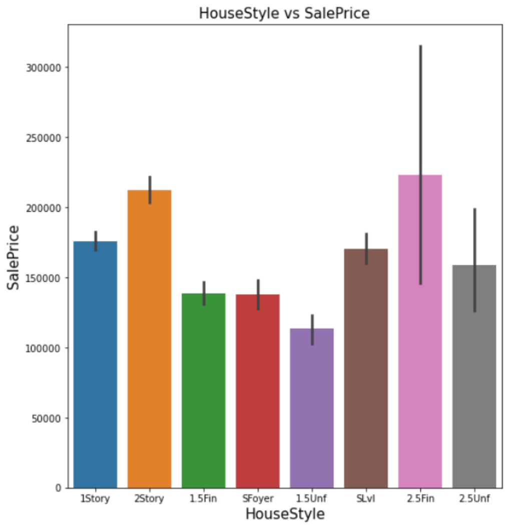


Observations :

As basement SF increases in relation to basement quality the sale price also increases.

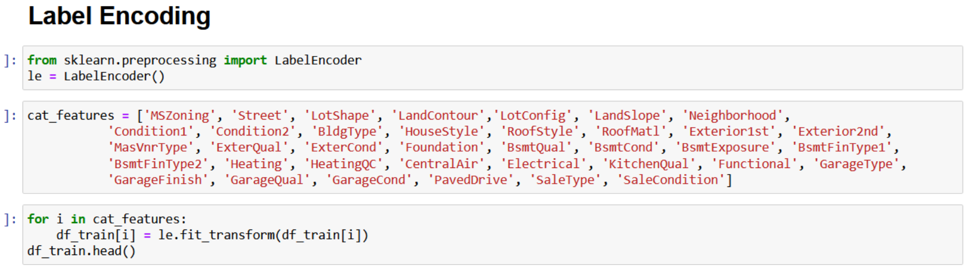
20. HouseStyle





▪ Label Encoding of Categorical features:

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



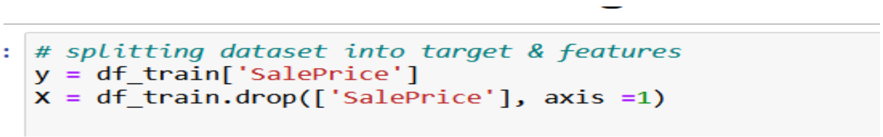
Label Encoding

```
]: from sklearn.preprocessing import LabelEncoder
   le = LabelEncoder()
```

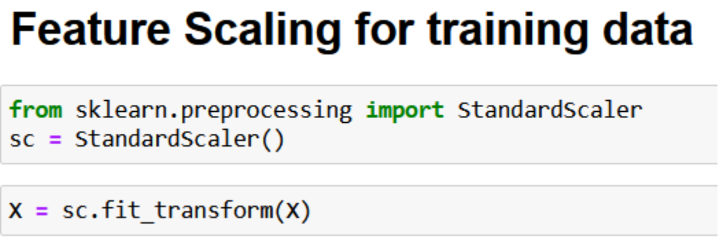
```
]: cat_features = ['MSZoning', 'Street', 'LotShape', 'LandContour','LotConfig', 'LandSlope', 'Neighborhood',
                'Condition1', 'Condition2', 'BldgType', 'HouseStyle', 'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd',
                'MasVnrType', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1',
                'BsmtFinType2', 'Heating', 'HeatingQC', 'CentralAir', 'Electrical', 'KitchenQual', 'Functional', 'GarageType',
                'GarageFinish', 'GarageQual', 'GarageCond', 'PavedDrive', 'SaleType', 'SaleCondition']
```

```
]: for i in cat_features:
       df_train[i] = le.fit_transform(df_train[i])
   df_train.head()
```

▪ Standard Scaling:



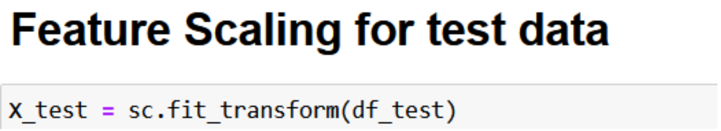
```
: # splitting dataset into target & features
  y = df_train['SalePrice']
  X = df_train.drop(['SalePrice'], axis =1)
```



Feature Scaling for training data

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
```

```
X = sc.fit_transform(X)
```



Feature Scaling for test data

```
X_test = sc.fit_transform(df_test)
```

Hardware & Software Requirements with Tool Used

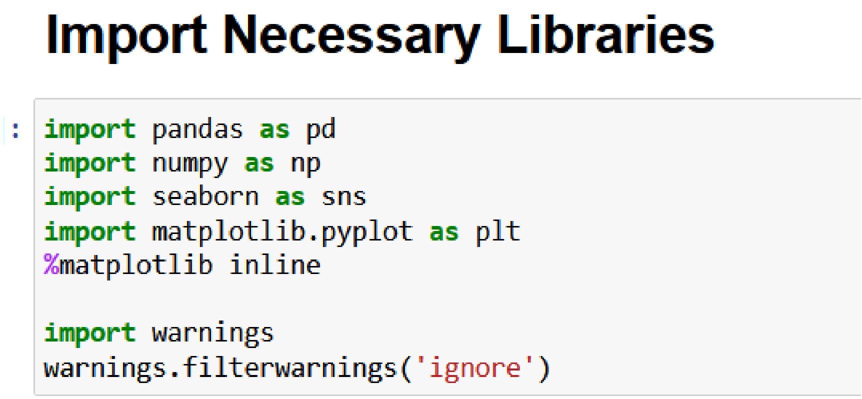
Hardware Used –

1. Processor — Intel i5 processor with 2.4GHZ
2. RAM — 8 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

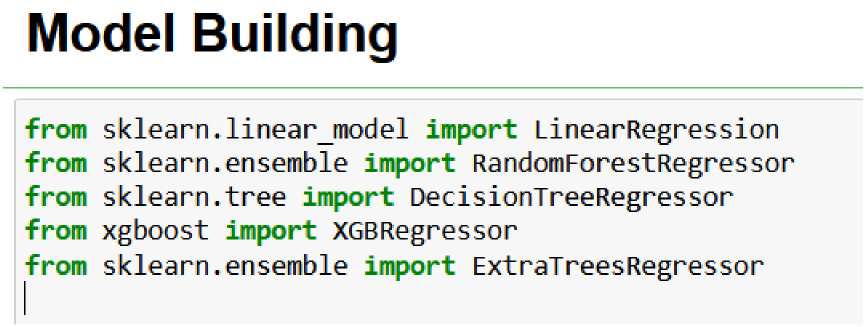


Import Necessary Libraries

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

import warnings
warnings.filterwarnings('ignore')
```

Libraries used for machine learning model building



Model Building

```
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from xgboost import XGBRegressor
from sklearn.ensemble import ExtraTreesRegressor
```

Chapter 4

Models Development & Evaluation

3.1. 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.

3.2. 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
- ❖ XGB Regressor
- ❖ Extra Tree Regressor

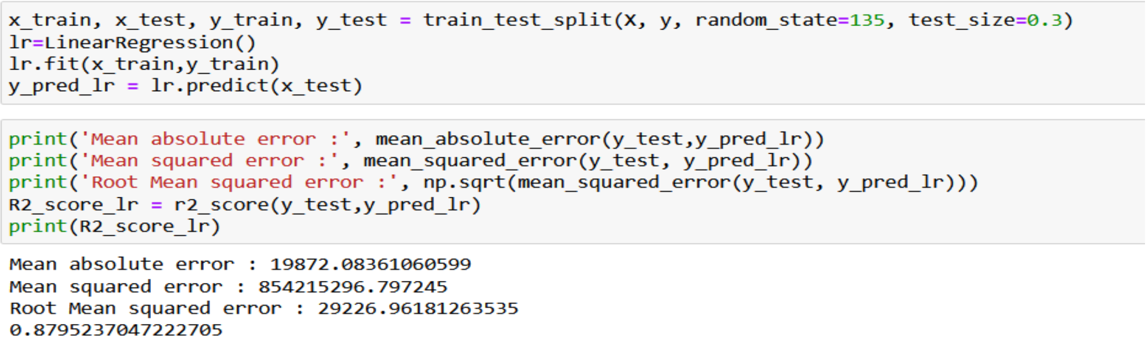
3.3. KEY METRICS FOR SUCCESS IN SOLVING PROBLEM UNDER CONSIDERATION

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

4.RUN AND EVALUATE SELECTED MODELS

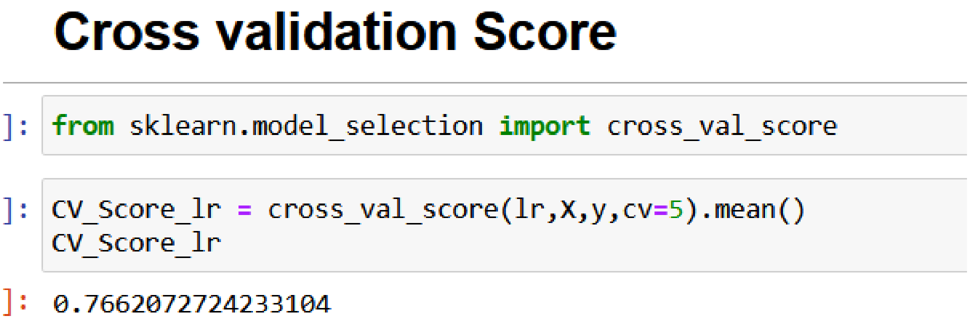
1. Linear Regression:



```
x_train, x_test, y_train, y_test = train_test_split(X, y, random_state=135, test_size=0.3)
lr=LinearRegression()
lr.fit(x_train,y_train)
y_pred_lr = lr.predict(x_test)
```

```
print('Mean absolute error :', mean_absolute_error(y_test,y_pred_lr))
print('Mean squared error :', mean_squared_error(y_test, y_pred_lr))
print('Root Mean squared error :', np.sqrt(mean_squared_error(y_test, y_pred_lr)))
R2_score_lr = r2_score(y_test,y_pred_lr)
print(R2_score_lr)
```

```
Mean absolute error : 19872.08361060599
Mean squared error : 854215296.797245
Root Mean squared error : 29226.96181263535
0.8795237047222705
```



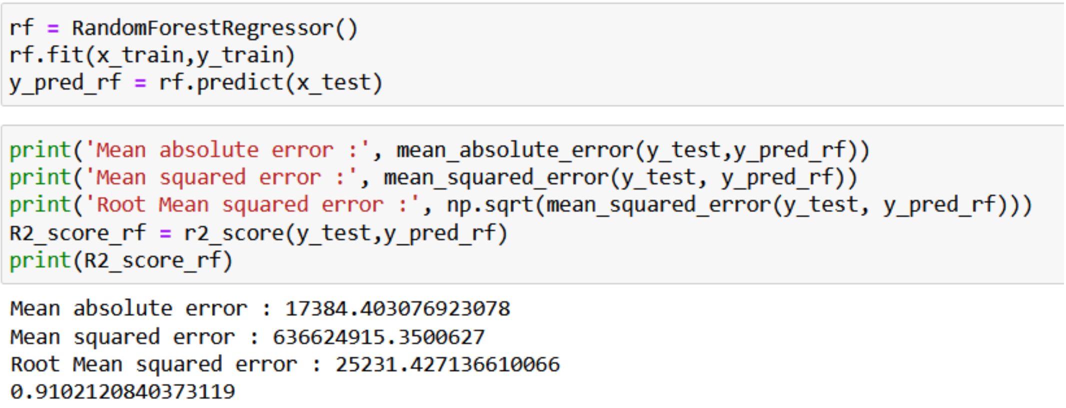
Cross validation Score

```
]: from sklearn.model_selection import cross_val_score
```

```
]: CV_Score_lr = cross_val_score(lr,X,y,cv=5).mean()
   CV_Score_lr
```

```
]: 0.7662072724233104
```

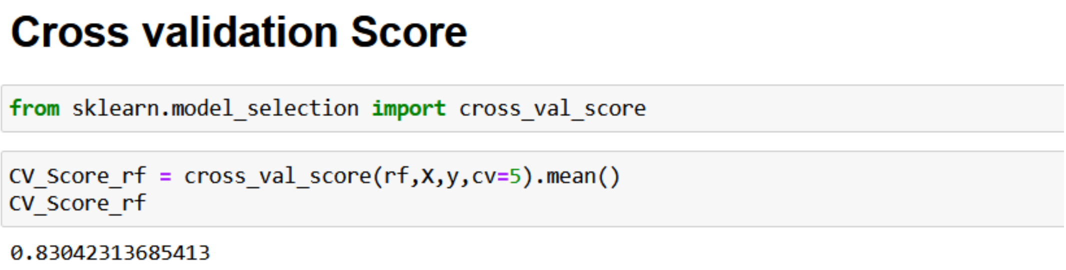
2. Random Forest Regressor:



```
rf = RandomForestRegressor()
rf.fit(x_train,y_train)
y_pred_rf = rf.predict(x_test)
```

```
print('Mean absolute error :', mean_absolute_error(y_test,y_pred_rf))
print('Mean squared error :', mean_squared_error(y_test, y_pred_rf))
print('Root Mean squared error :', np.sqrt(mean_squared_error(y_test, y_pred_rf)))
R2_score_rf = r2_score(y_test,y_pred_rf)
print(R2_score_rf)
```

```
Mean absolute error : 17384.403076923078
Mean squared error : 636624915.3500627
Root Mean squared error : 25231.427136610066
0.9102120840373119
```



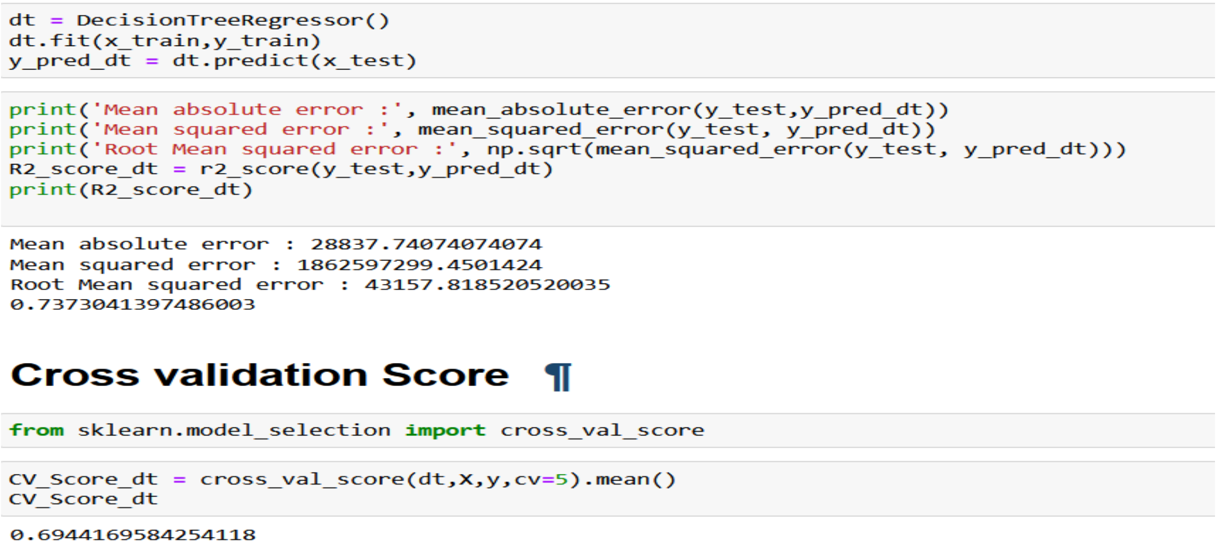
Cross validation Score

```
from sklearn.model_selection import cross_val_score
```

```
CV_Score_rf = cross_val_score(rf,X,y,cv=5).mean()
CV_Score_rf
```

```
0.83042313685413
```

3. Decision Tree Regressor:



```
dt = DecisionTreeRegressor()
dt.fit(x_train,y_train)
y_pred_dt = dt.predict(x_test)
```

```
print('Mean absolute error :', mean_absolute_error(y_test,y_pred_dt))
print('Mean squared error :', mean_squared_error(y_test, y_pred_dt))
print('Root Mean squared error :', np.sqrt(mean_squared_error(y_test, y_pred_dt)))
R2_score_dt = r2_score(y_test,y_pred_dt)
print(R2_score_dt)
```

```
Mean absolute error : 28837.74074074074
Mean squared error : 1862597299.4501424
Root Mean squared error : 43157.818520520035
0.7373041397486003
```

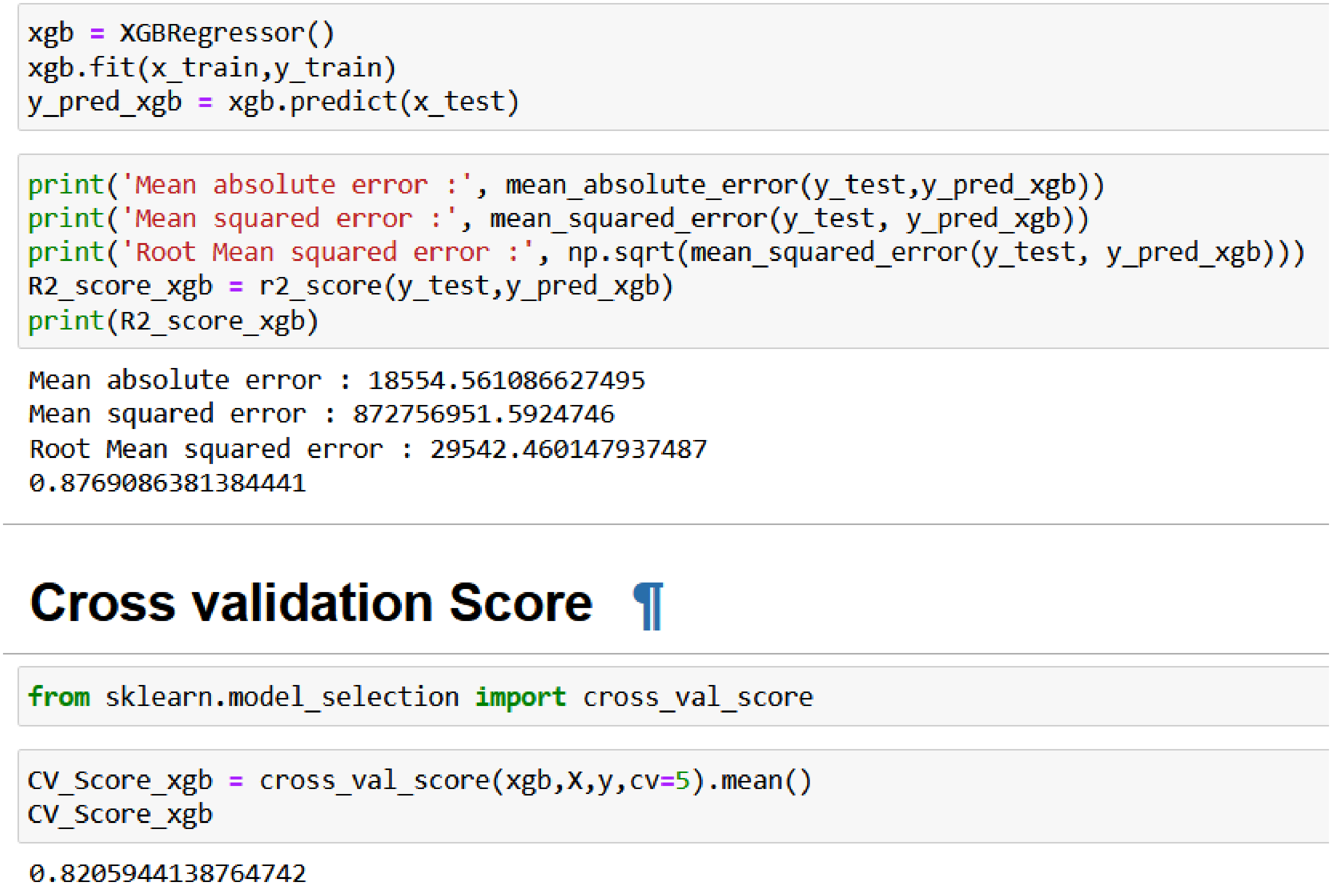
Cross validation Score ¶

```
from sklearn.model_selection import cross_val_score
```

```
CV_Score_dt = cross_val_score(dt,X,y,cv=5).mean()
CV_Score_dt
```

```
0.6944169584254118
```

4. XGB Regressor:



```
xgb = XGBRegressor()
xgb.fit(x_train,y_train)
y_pred_xgb = xgb.predict(x_test)
```

```
print('Mean absolute error :', mean_absolute_error(y_test,y_pred_xgb))
print('Mean squared error :', mean_squared_error(y_test, y_pred_xgb))
print('Root Mean squared error :', np.sqrt(mean_squared_error(y_test, y_pred_xgb)))
R2_score_xgb = r2_score(y_test,y_pred_xgb)
print(R2_score_xgb)
```

```
Mean absolute error : 18554.561086627495
Mean squared error : 872756951.5924746
Root Mean squared error : 29542.460147937487
0.8769086381384441
```

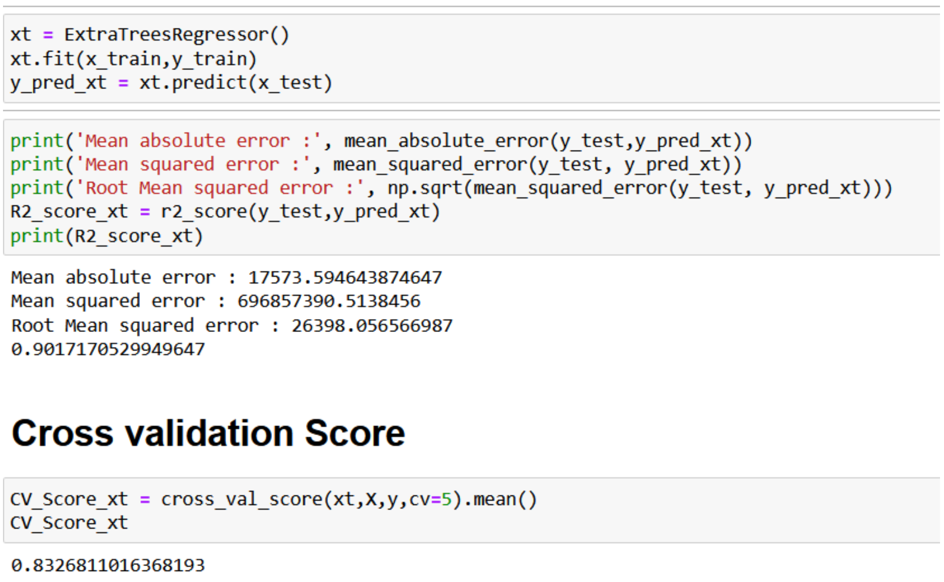
Cross validation Score ¶

```
from sklearn.model_selection import cross_val_score
```

```
CV_Score_xgb = cross_val_score(xgb,X,y,cv=5).mean()
CV_Score_xgb
```

```
0.8205944138764742
```

5. Extra Trees Regressor:



```
xt = ExtraTreesRegressor()
xt.fit(x_train,y_train)
y_pred_xt = xt.predict(x_test)
```

```
print('Mean absolute error :', mean_absolute_error(y_test,y_pred_xt))
print('Mean squared error :', mean_squared_error(y_test, y_pred_xt))
print('Root Mean squared error :', np.sqrt(mean_squared_error(y_test, y_pred_xt)))
R2_score_xt = r2_score(y_test,y_pred_xt)
print(R2_score_xt)
```

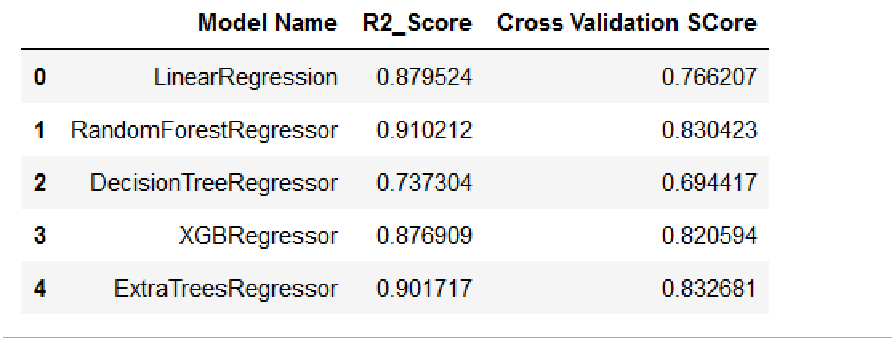
```
Mean absolute error : 17573.594643874647
Mean squared error : 696857390.5138456
Root Mean squared error : 26398.056566987
0.9017170529949647
```

Cross validation Score

```
CV_Score_xt = cross_val_score(xt,X,y,cv=5).mean()
CV_Score_xt
```

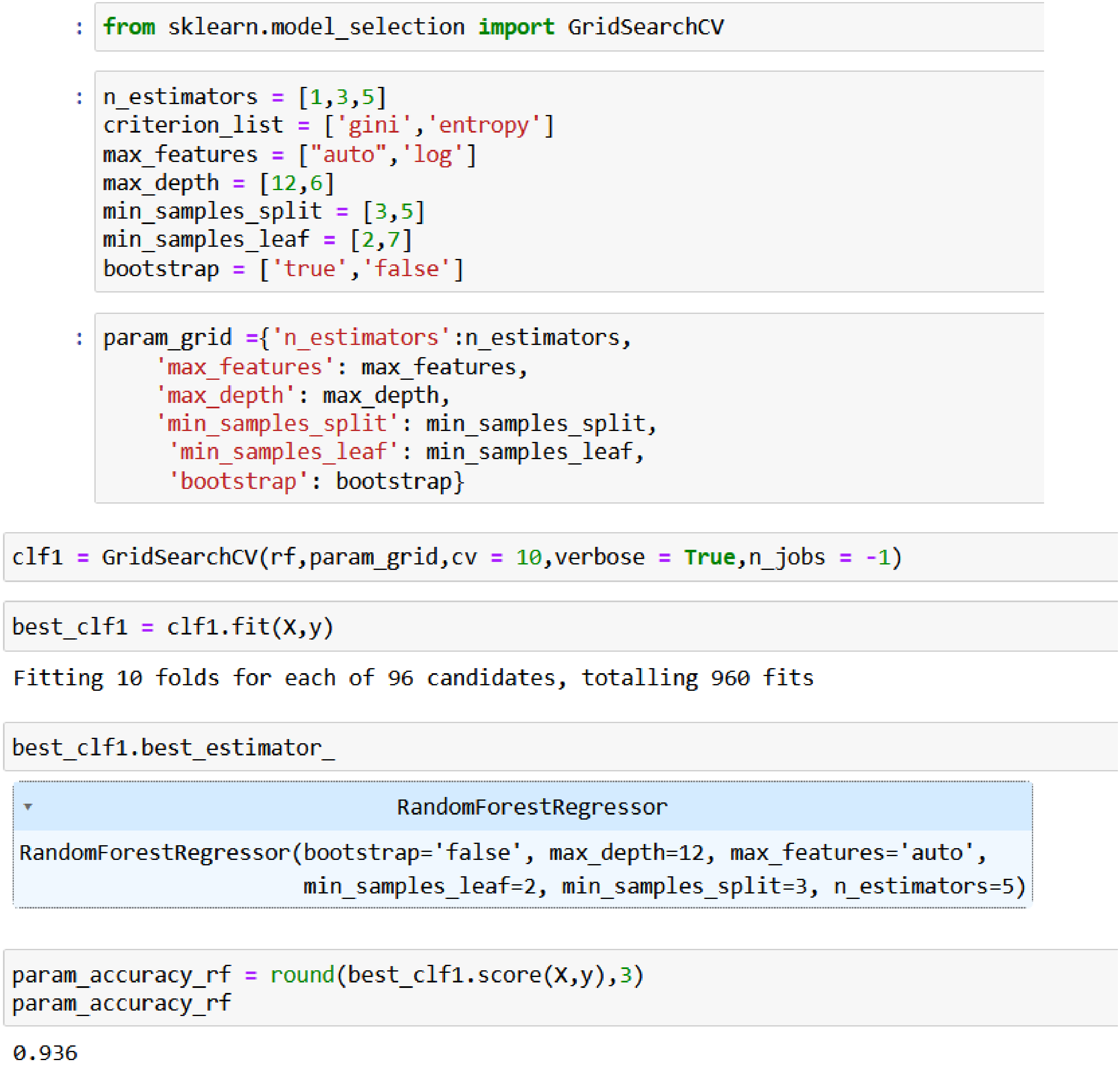
```
0.8326811016368193
```

Result of All models saved in a dataframe :



	Model Name	R2_Score	Cross Validation SCore
0	LinearRegression	0.879524	0.766207
1	RandomForestRegressor	0.910212	0.830423
2	DecisionTreeRegressor	0.737304	0.694417
3	XGBRegressor	0.876909	0.820594
4	ExtraTreesRegressor	0.901717	0.832681

5-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.



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

```
n_estimators = [1,3,5]
criterion_list = ['gini','entropy']
max_features = ["auto",'log']
max_depth = [12,6]
min_samples_split = [3,5]
min_samples_leaf = [2,7]
bootstrap = ['true','false']
```

```
param_grid ={'n_estimators':n_estimators,
    'max_features': max_features,
    'max_depth': max_depth,
    'min_samples_split': min_samples_split,
     'min_samples_leaf': min_samples_leaf,
     'bootstrap': bootstrap}
```

```
clf1 = GridSearchCV(rf,param_grid,cv = 10,verbose = True,n_jobs = -1)
```

```
best_clf1 = clf1.fit(X,y)
```

```
Fitting 10 folds for each of 96 candidates, totalling 960 fits
```

```
best_clf1.best_estimator_
```

```
RandomForestRegressor
RandomForestRegressor(bootstrap='false', max_depth=12, max_features='auto',
                      min_samples_leaf=2, min_samples_split=3, n_estimators=5)
```

```
param_accuracy_rf = round(best_clf1.score(X,y),3)
param_accuracy_rf
```

```
0.936
```

We can see that hyper parameter tuning leading to increase in R2 Score slightly from default model.

Chapter 5

Conclusion

1. Random Forest Regressor giving us maximum R2 Score, so Random Forest Regressor is selected as best model.

2. After hyper parameter tuning Final Model is giving us R2 Score of 0.936.

Important Variables for House price Prediction

1. Overall Quality
2. GrLivArea
3. GarageCars
4. GarageArea
5. TotalbsmtSF
6. FullBath
7. TotRmsAbvGrd
8. MasVnrArea
9. Fireplaces
10. OpenPorchSF
11. LotFrontage
12. WoodDeckSF
13. HalfBath
14. LotArea
15. year_sincebuilt
16. year_RemodAdd
17. Garage_age
18. KitchenAbvGr
19. BsmtFullBath
20. BedroomAbvGr