



# HOUSING PRICE PREDICTION & ANALYSIS PROJECT

**Submitted By:**

SHAHLA M

**Fliprobo SME :**

Ms. KHUSHBOO GARG

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

Special thanks to “Data trained” institute who are the reason behind my Internship at Flip Robo Technologies. Last but not least my parents who have been my backbone in every step of my life.

### REFERENCE

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

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. 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. 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.
11. H. Kusan, O. Aytekin, and I. Ozdemir, "The use of fuzzy logic " in predicting house selling price," Expert Systems with Applications, vol. 37, no. 3, pp. 1808–1813, 2010.
12. 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.

13. 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.

# Chapter 1. Introduction

## 1.1 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 investments.

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

## Housing Price Prediction

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.

### 1.2 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.

### 1.3 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)

## Housing Price Prediction

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.

Zhongyun, Jiang, Guoxin, Shen [8], 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.

Zhen Peng, Qiang Huang, Yincheng Han [9], precisely study the cost of recycled houses, and examined 35417 bits of information caught by the Chengdu HOME LINK organization. First and foremost, the caught information was cleaned, and the attributes were chosen. The test results show that the precision of XGboost expectation is the most elevated, and the forecast exactness score arrives at 0.9251. Dissimilarity with linear regression, decision tree model, the XGboost algorithm gives improved speculation capacity and strength in information forecast, and, furthermore, data prediction over-fitting aspect, establishing a strong framework for the ensuing recycled house value expectation.

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 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.



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 were used mainly to compare all the predictions and display the most 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 Attributes” [13], uses visual attributes to predict the sale price of the property.

### **1.4 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, Lasso, Ridge, 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:

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

## Chapter 2

# Analytical Problem Framing

### 2. Mathematical / Analytical 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

# Housing Price Prediction

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.

## 2.1 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.

```
In [2]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')

from sklearn.model_selection import train_test_split
```

```
In [3]: train = pd.read_csv("C:\\Users\\sahal\\Downloads\\Project-Housing splitted\\train.csv")
```

```
In [4]: train.sample(n=6)
```

```
Out[4]:
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	Yi
230	1308	20	RL	60.0	8072	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	5	
1107	1180	20	RL	77.0	8335	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	4	
37	176	20	RL	84.0	12615	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0	6	
272	810	75	RM	90.0	8100	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0	11	
550	852	120	RL	NaN	3196	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	5	
875	1304	20	RL	73.0	8688	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	4	

6 rows x 81 columns

The data types of different features are as shown below:

```
# 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',
'LandSale', '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']}]
```

### 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.

## Data Pre-processing Done:

### 2.3.1 Checking Null:

Null values are present in the total dataset.

### Checking Null:

```
In [9]: null_val= train.isna().sum().any()
null_val

Out[9]: True

In [10]: pd.set_option('display.max_rows',None)
null_val= train.isna().sum().sort_values(ascending = False)
null_val_per =(null_val/train.shape[0])*100
l= [null_val, null_val_per]
ll= pd.concat(l, axis =1, keys =['Null Values', 'Null Values percentage']).sort_values('Null Values')

print("missing value details \n\n", ll)

missing value details
```

Null Values Null Values percentage

Some features contain missing values as shown below:

## Housing Price Prediction

```
# Finding what percentage of data is missing 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
GarageType	64	5.479452
GarageCond	64	5.479452
GarageYrBlt	64	5.479452
GarageFinish	64	5.479452
GarageQual	64	5.479452
BsmtExposure	31	2.654110
BsmtFinType2	31	2.654110
BsmtFinType1	30	2.568493
BsmtCond	30	2.568493
BsmtQual	30	2.568493
MasVnrArea	7	0.599315
MasVnrType	7	0.599315
Exterior2nd	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 &

```
: # Removing columns with high 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])
```

```
: dft.rename(columns = {'Year_Since' : 'Year_Since_Sold'}, inplace = True)
```

There are some columns with same meaning. Drop those columns also for both train and test dataset.

# Housing Price Prediction

```
# Converting years column to age column
df['Year_SinceBuilt'] = df['YearBuilt'].max() - df['YearBuilt']
df['Year_SinceRemodAdd'] = df['YearRemodAdd'].max() - df['YearRemodAdd']
df['Year_Since'] = df['YrSold'].max() - df['YrSold']
df['GarageAge'] = df['GarageYrBlt'].max() - df['GarageYrBlt']

# Dropping old columns in train dataset
df.drop(['YearBuilt', 'YearRemodAdd', 'YrSold', 'GarageYrBlt'], axis=1, inplace = True)
```

another similar feature.

- **BsmtFinSF1:** Type 1 finished square feet
- **BsmtFinSF2:** Type 2 finished square feet
- **BsmtUnfSF:** Unfinished square feet of basement area
- **TotalBsmtSF:** Total square feet of basement area

*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)
```

- **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

*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.



## Housing Price Prediction

```
# 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()
```

□ Standard Scaling:

### 4.1 Standard Scaling for Train Dataset

```
# Splitting data in target and dependent feature
X = df.drop(['SalePrice'], axis =1)
Y = df['SalePrice']
```

```
from sklearn.preprocessing import StandardScaler
scaler= StandardScaler()
X_scale = scaler.fit_transform(X)
```

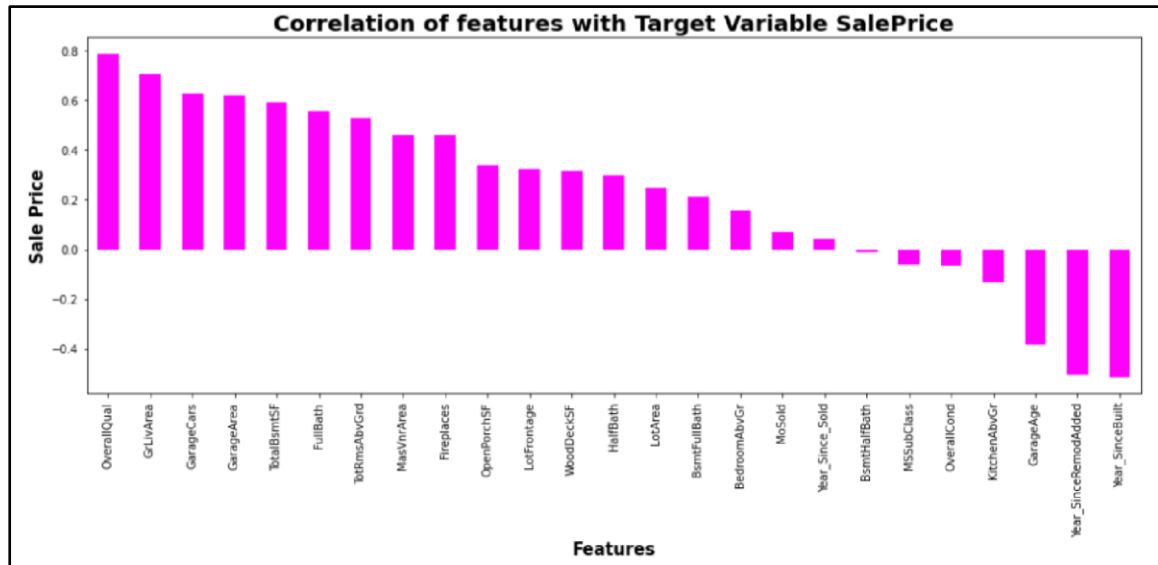
### 4.2 Standard Scaling for Test Dataset

```
from sklearn.preprocessing import StandardScaler
scaler= StandardScaler()
X_scale_test = scaler.fit_transform(dft)
```

## 4. 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.

# Housing Price Prediction



## 5. Hardware & Software Requirements with Tool 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

```
#Importing warning library to avoid any warnings
import pandas as pd # for data wrangling purpose
import numpy as np # Basic computation library
import seaborn as sns # For Visualization
import matplotlib.pyplot as plt # plotting package
%matplotlib inline
import warnings # Filtering warnings
warnings.filterwarnings('ignore')
```



### 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
from sklearn.ensemble import ExtraTreesRegressor
from xgboost import XGBRegressor
```

## Chap. 3 Models Development & Evaluation

1. Identification Of Possible Problem-Solving Approaches (Methods)
2. Testing of Identified Approaches (Algorithms)

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

- ✚ Linear Regression
- ✚ Decision Tree Regressor
- ✚ KNN Regressor
- ✚ Random Forest Regressor
- ✚ Support Vector Regressor

✚ Gradient Boost Regressor

### 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. 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

## 4. RUN AND EVALUATE SELECTED MODELS

### 1. Linear Regression:

Using LinearRegression()

```
In [146]: x_train,x_test,y_train,y_test = train_test_split(x_scale,y,test_size = 0.25, random_state=654)

lin_reg= LinearRegression()

lin_reg.fit(x_train, y_train)

y_pred = lin_reg.predict(x_test)

print('R2 Score:', r2_score(y_test, y_pred))
print('Mean absolute error:', mean_absolute_error(y_test, y_pred))
print('Mean square error:', mean_squared_error(y_test, y_pred))
print('Root mean square error:',np.sqrt( mean_squared_error(y_test, y_pred)))

R2 Score: 0.8816206614832437
Mean absolute error: 19209.648878261378
Mean square error: 653721476.9612094
Root mean square error: 25567.977568849856
```

### 2. Decision Tree Regressor

# Housing Price Prediction

Using DecisionTreeRegressor():

```
In [149]: from sklearn.tree import DecisionTreeRegressor
dt = DecisionTreeRegressor()
dt.fit(x_train, y_train)
y_pred = dt.predict(x_test)
print('R2 Score:', r2_score(y_test, y_pred))
print('Mean absolute error:', mean_absolute_error(y_test, y_pred))
print('Mean square error:', mean_squared_error(y_test, y_pred))
print('Root mean square error:', np.sqrt(mean_squared_error(y_test, y_pred)))

R2 Score: 0.7524415137483609
Mean absolute error: 26604.39383561644
Mean square error: 1367082307.5582192
Root mean square error: 36974.076155574454
```

Using DecisionTreeRegressor():

```
In [149]: from sklearn.tree import DecisionTreeRegressor
dt = DecisionTreeRegressor()
dt.fit(x_train, y_train)
y_pred = dt.predict(x_test)
print('R2 Score:', r2_score(y_test, y_pred))
print('Mean absolute error:', mean_absolute_error(y_test, y_pred))
print('Mean square error:', mean_squared_error(y_test, y_pred))
print('Root mean square error:', np.sqrt(mean_squared_error(y_test, y_pred)))

R2 Score: 0.7524415137483609
Mean absolute error: 26604.39383561644
Mean square error: 1367082307.5582192
Root mean square error: 36974.076155574454
```

Using DecisionTreeRegressor():

```
In [149]: from sklearn.tree import DecisionTreeRegressor
dt = DecisionTreeRegressor()
dt.fit(x_train, y_train)
y_pred = dt.predict(x_test)
print('R2 Score:', r2_score(y_test, y_pred))
print('Mean absolute error:', mean_absolute_error(y_test, y_pred))
print('Mean square error:', mean_squared_error(y_test, y_pred))
print('Root mean square error:', np.sqrt(mean_squared_error(y_test, y_pred)))

R2 Score: 0.7524415137483609
Mean absolute error: 26604.39383561644
Mean square error: 1367082307.5582192
Root mean square error: 36974.076155574454
```

## 4. Extra Trees Regressor:

```
X_train, X_test, Y_train, Y_test = train_test_split(X_scale, Y, random_state= 135, test_size=0.33)
etc = ExtraTreesRegressor()
etc.fit(X_train, Y_train)
y_pred = etc.predict(X_test)
print('\033[1m+ 'Error of Extra 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 Extra Tree Regressor : '+' '\033[0m')
print(r2_score(Y_test,y_pred)*100)
```

```
Error of Extra Tree Regressor:
Mean absolute error : 17501.656424870467
Mean squared error : 672482129.7249866
Root Mean squared error : 25932.260405236302
R2 Score of Extra Tree Regressor :
90.3380632075058
```

## 5. XGB Regressor:

```
X_train, X_test, Y_train, Y_test = train_test_split(X_scale, Y, random_state= 135, test_size=0.33)
xgb = XGBRegressor()
xgb.fit(X_train, Y_train)
y_pred = xgb.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 XGB Regressor :'+ '\033[0m')
print(r2_score(Y_test, y_pred)*100)
```

```
Error of XGB Regressor:
Mean absolute error : 19112.426550356216
Mean squared error : 927210436.1906452
Root Mean squared error : 30450.130314838476
R2 Score of XGB Regressor :
86.67823540310482
```

## 6. Ridge Regressor:

```
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 XGB Regressor :'+ '\033[0m')
print(r2_score(Y_test, y_pred)*100)
```

```
Error of XGB Regressor:
Mean absolute error : 20090.342142178335
Mean squared error : 868587960.7881687
Root Mean squared error : 29471.81638087766
R2 Score of XGB Regressor :
87.52049815912765
```

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

# Housing Price Prediction

parameter tuning over this model to enhance its R2 Score.

```
from sklearn.model_selection import GridSearchCV

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': 'auto', 'max_leaf_nodes': None, 'max_samples': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 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': 'sqrt', 'min_samples_leaf': 1, 'n_estimators': 100}

Final Model

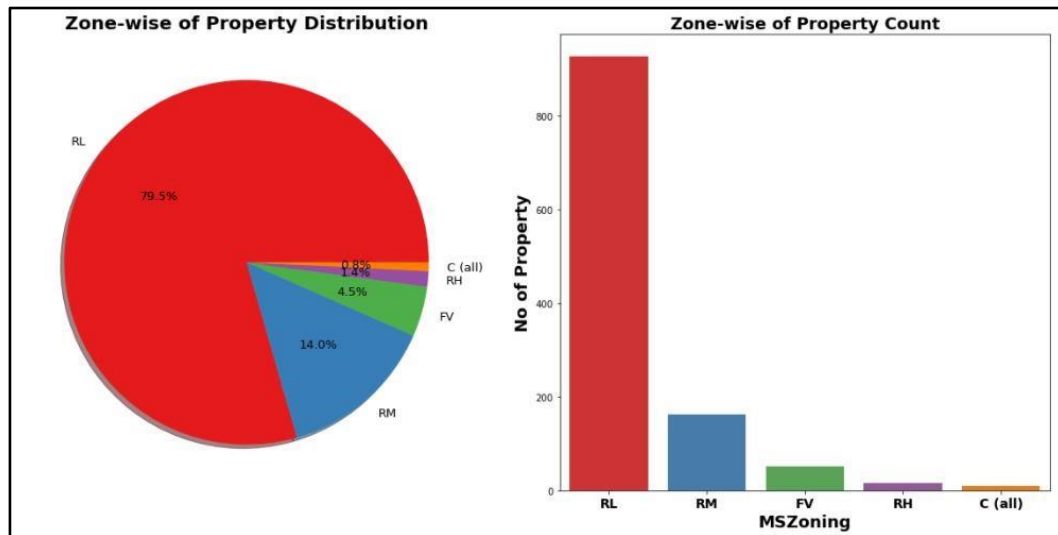
Final_mod=RandomForestRegressor(max_depth = None, bootstrap = True, max_features='sqrt', min_samples_leaf = 1,
                                n_estimators = 100 )
Final_mod.fit(X_train, Y_train)
pred=Final_mod.predict(X_test)
print('R2_Score:', r2_score(Y_test, pred)*100)
print('mean_squared_error:', 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: 90.39311489727983
mean_squared_error: 668650467.566633
mean_absolute_error: 17782.643231865284
RMSE value: 25858.276577657547
```

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

### VISUALIZATIONS

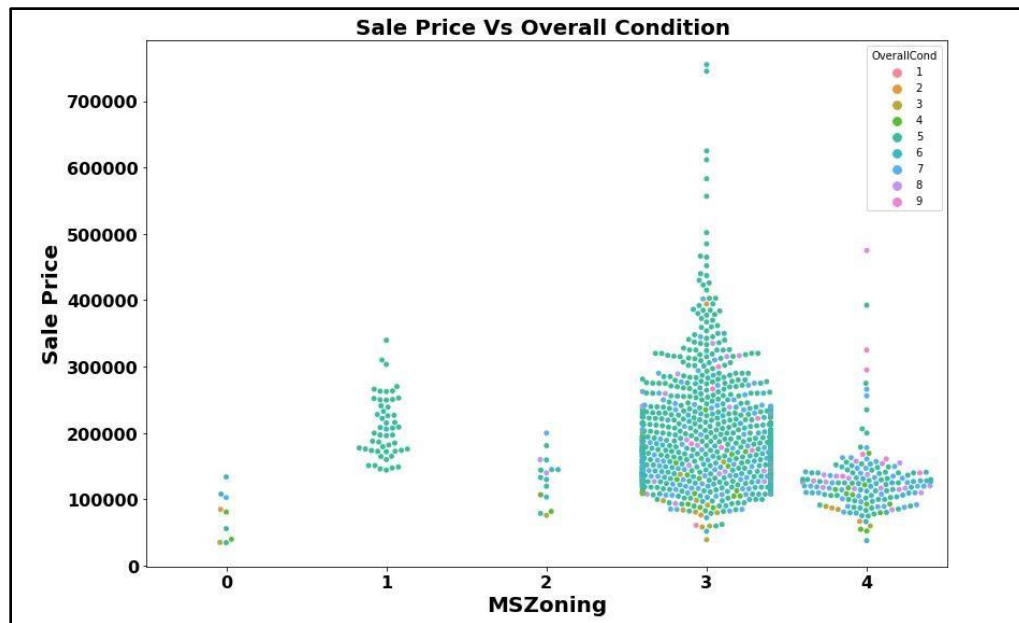
Let see key result from EDA, start with zone wise distribution of property



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 properties (0.8%) belong to Commercial zone.

## Housing Price Prediction



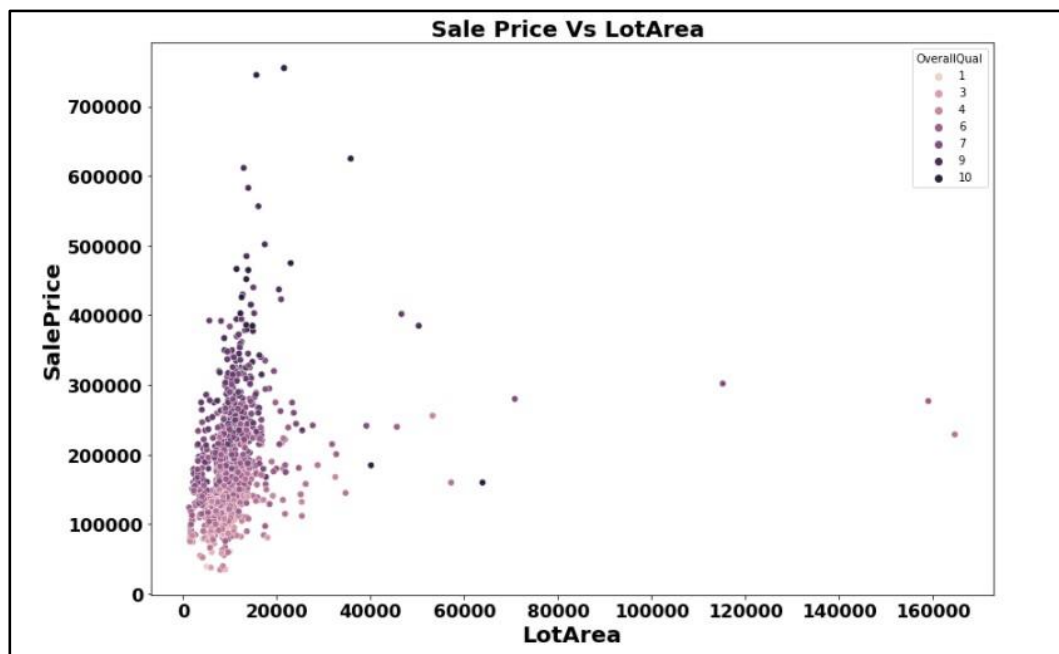
### Observation:

- Most of property for sale have overall condition rating of either 5 or 6.
- We already know of 80% of housing data belongs to Low density Residential Area and Now we can see in Swarm Plot that Sale Price inside RL Zone is much higher than another remaining zone.
- Cheapest properties are available in Commercial zone.
- Another interesting observation we get here is for 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. Overall Condition Rating may helpful to buyer in taking decision of Buying property but not in determination of House Price.

## Housing Price Prediction



With Exception of Commercial zone, As Lot Frontage area increase (which indicate Size of street connected to property) the Sale Price increases.



Observation:

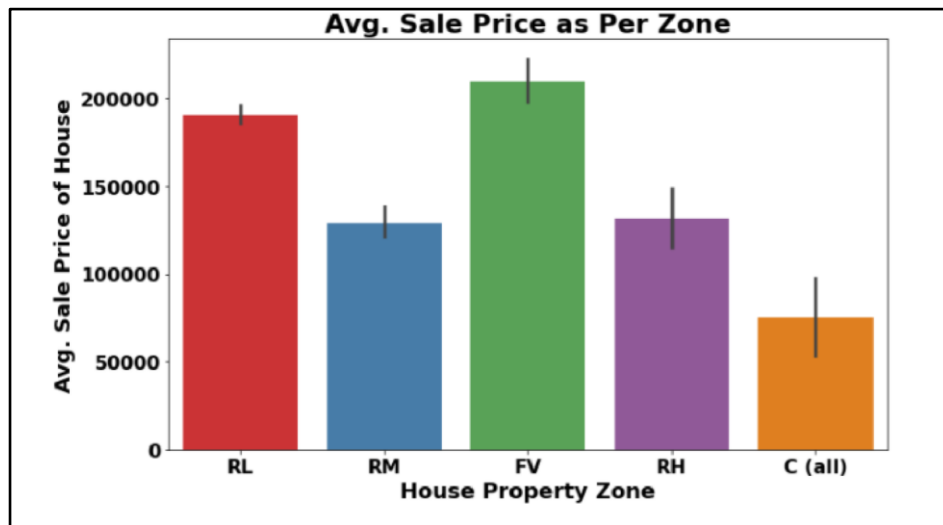
- There is No Significant relationship found between Sale price & Lot area.

Here we get Important Observation that -

As Overall Quality of House Increase the Sale Price of House also Increases.

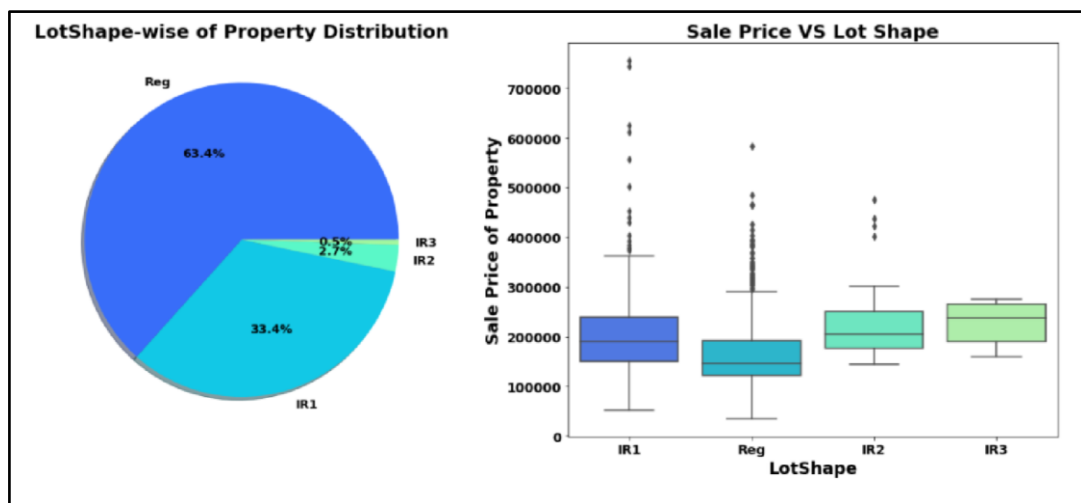


## Housing Price Prediction



Observation:

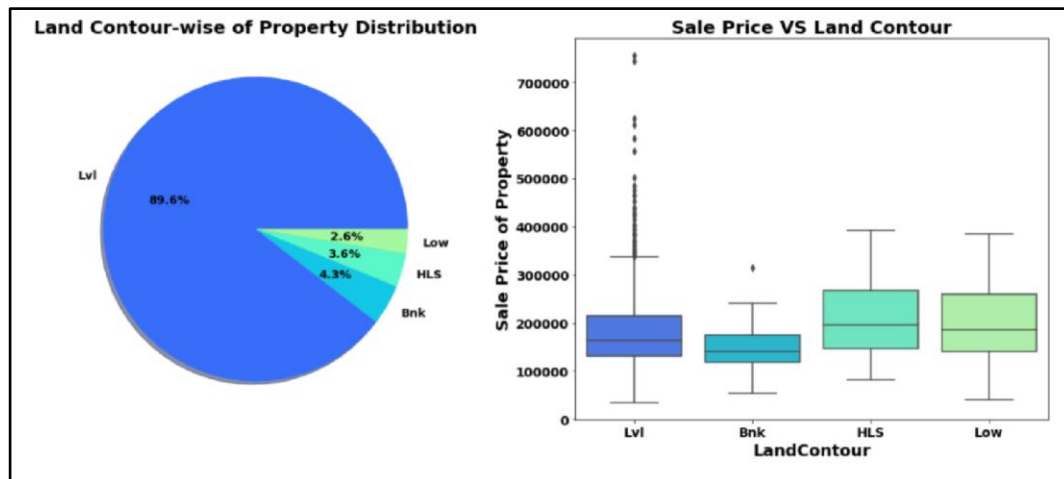
- In terms of Average Sale price house properties belonging to Floating Village Residential Zone are costlier than rest.



Observation:

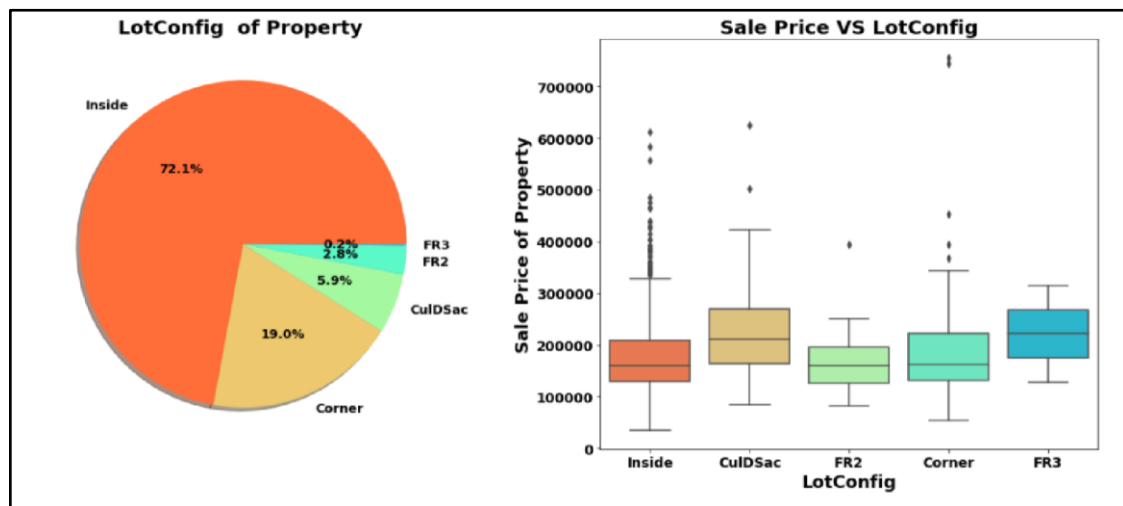
- 63.4% house properties are regular in shape.
- Sale Price of property with slight irregular shape is higher than regular shape.

## Housing Price Prediction



Observation:

- 89.6% of House properties are near flat level surface.
- Also, price for Flat level surface house is much higher than other land contour.

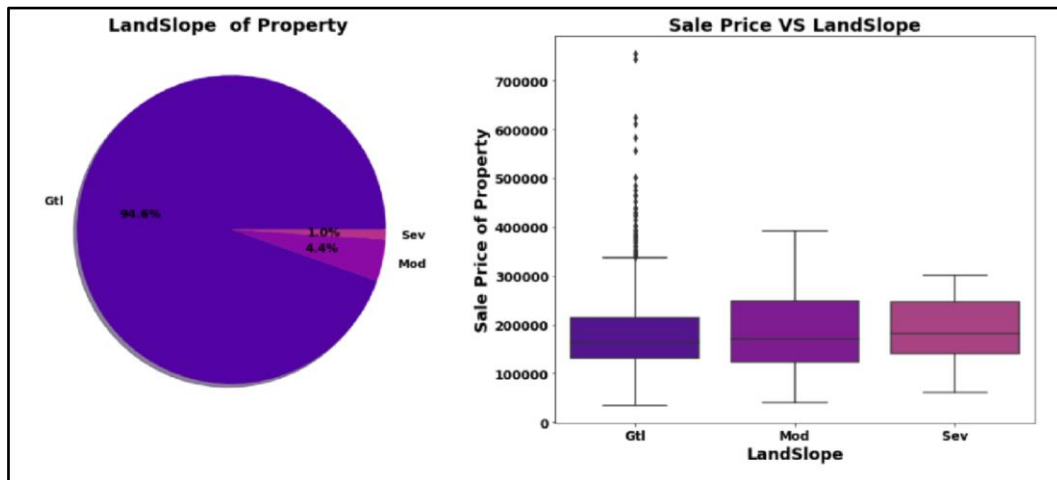


Observation: -

- Around 72 % of house comes with inside Lot configuration.
- Cul-de-sac has maximum Mean Sale Price among all lot configuration.

## Housing Price Prediction

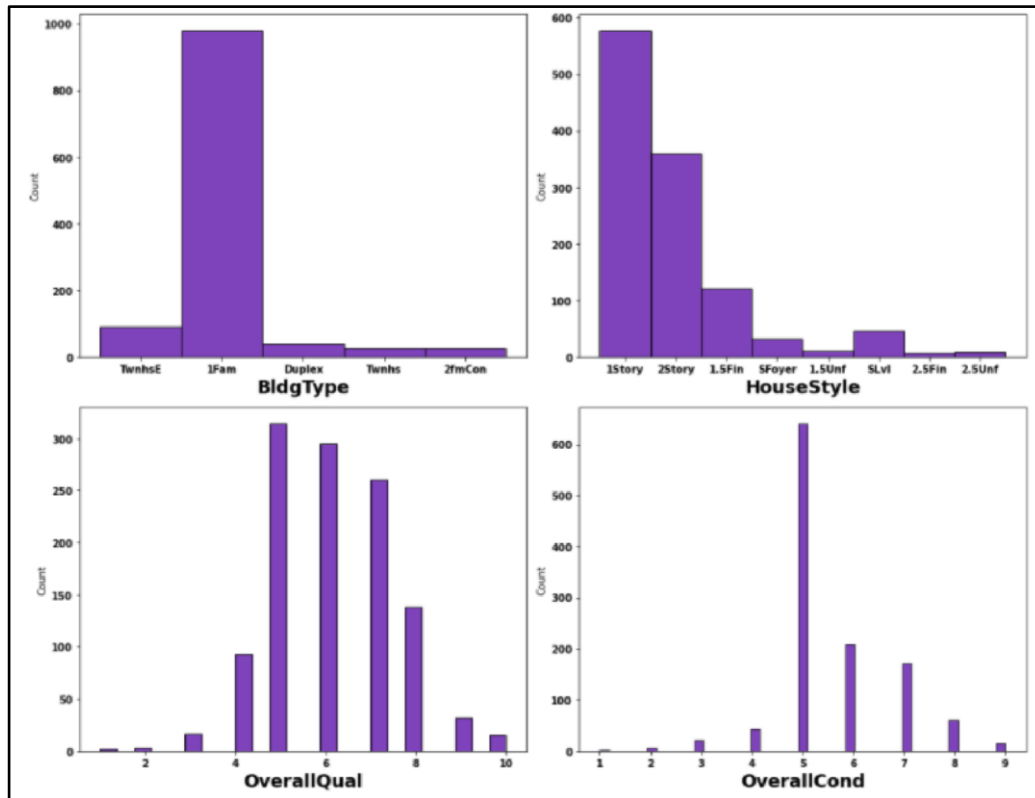
- Cheapest Houses belong to Inside lot configuration while Costlier houses belongs to Corner Lot Configuration.



### Observation:

- Clearly, 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.

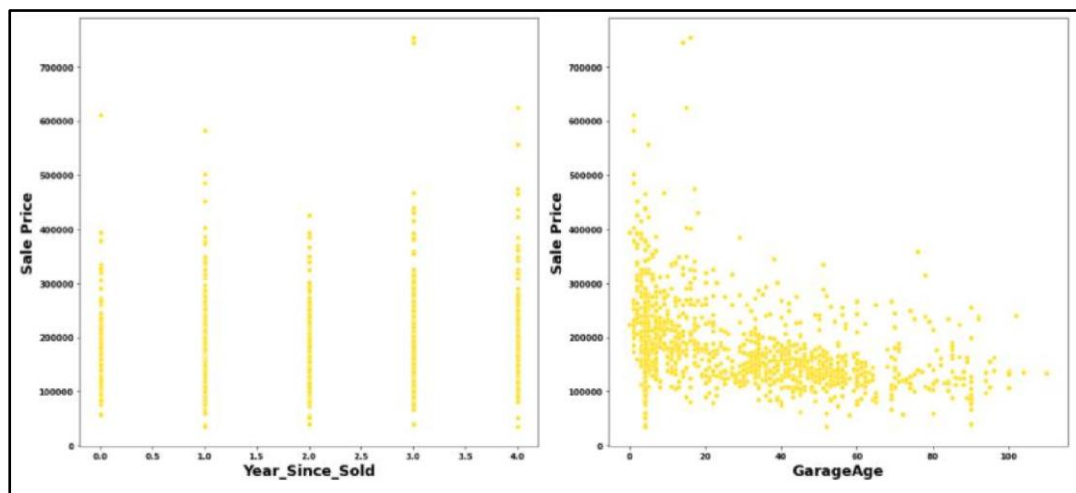
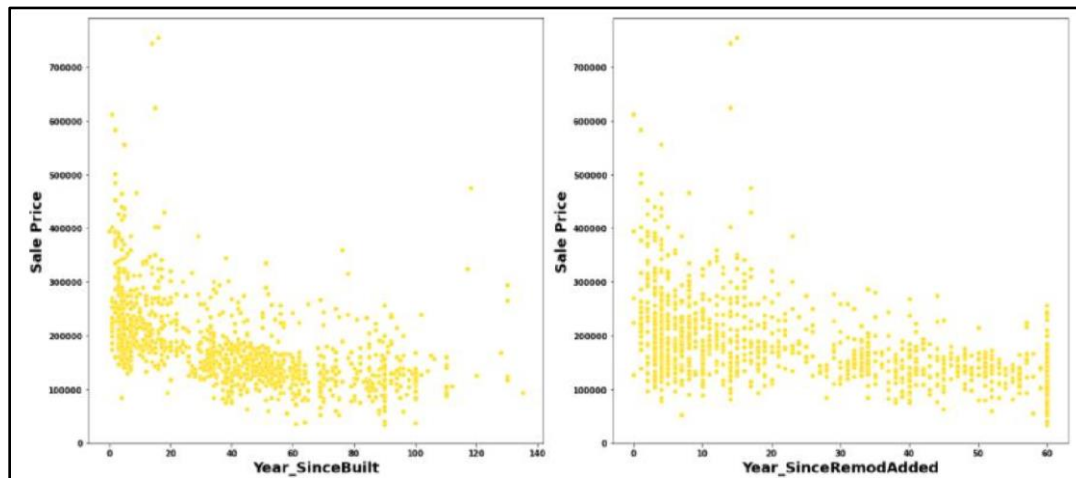
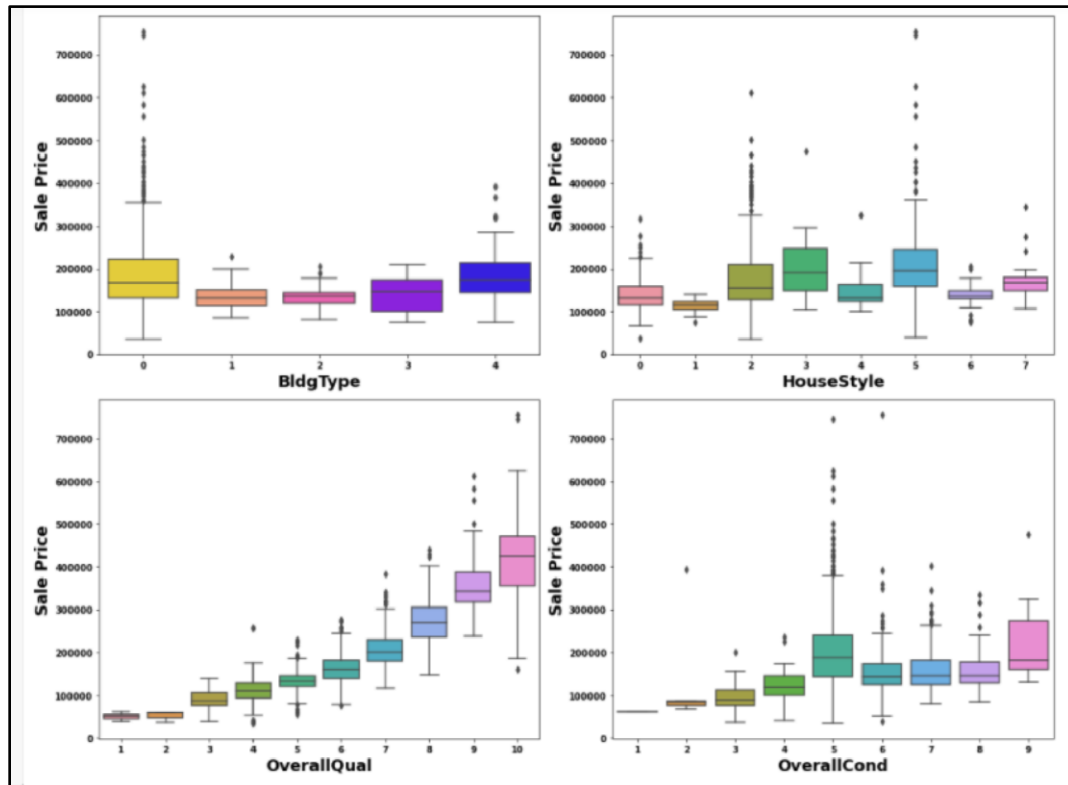
## Housing Price Prediction



### Observation:

- More than 950 house properties are with building type Singlefamily 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.

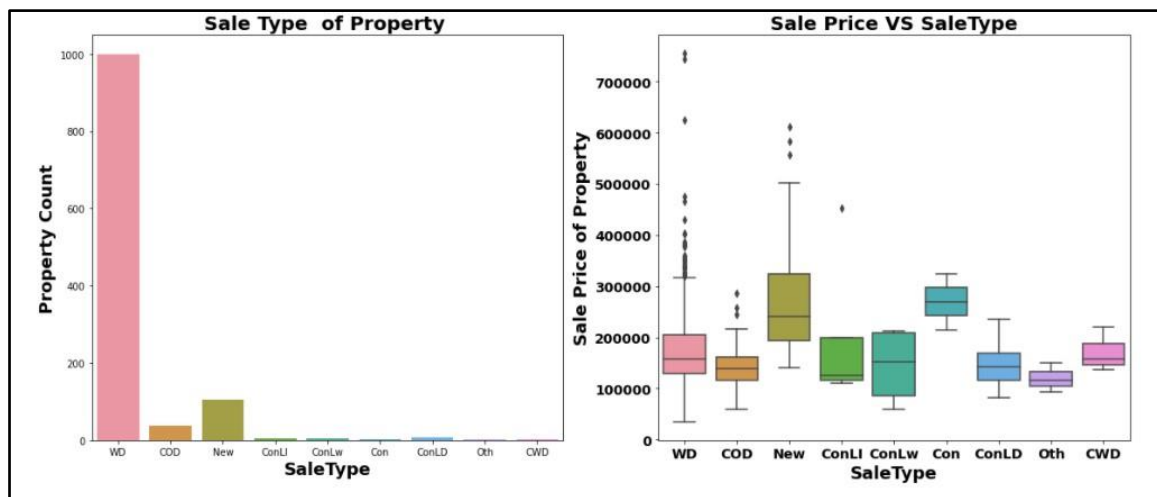
# Housing Price Prediction



-

Observation:

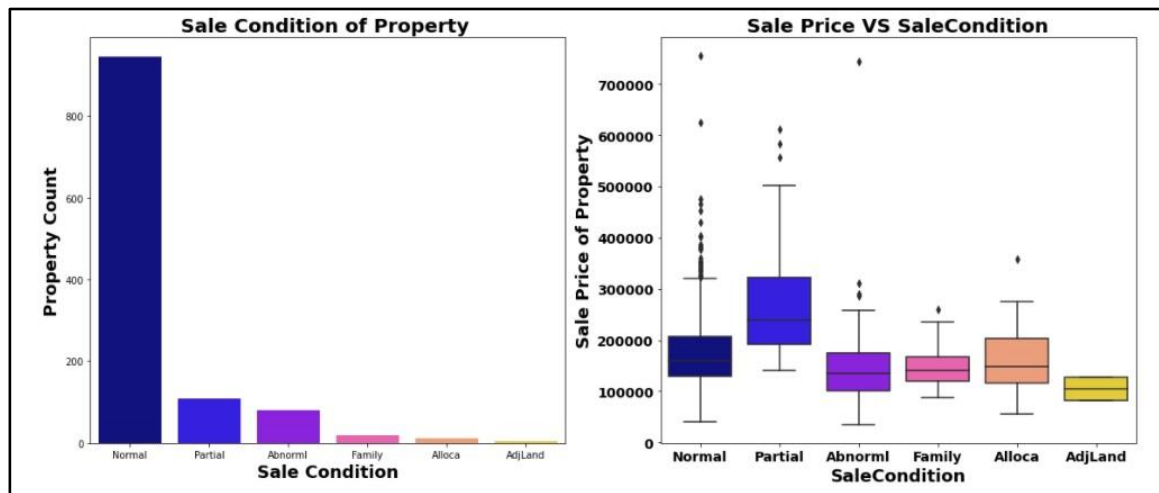
- We can see that as Property get older with time its sale Price get depreciates.
- 20 years after Remodelling Price of properties start decreases.
- Older the age of garage lesser the price of Property.



Observation: -

- Around 1000 sales happen by Conventional Warranty Deed.
- Home just constructed and sold category are exceptionally much costlier than anyone else.
- All loan-based sale is below 300000.

## Surprise Housing - Housing Price Predication & Analysis Project



Observation:

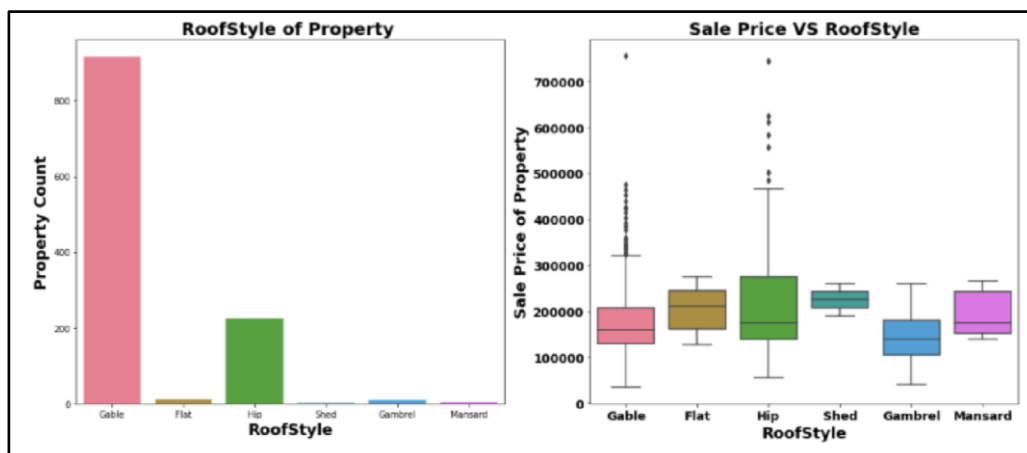
- We can see that Sale with condition like Abnorml, Family, Alloca, AdjLand are below the price of 300000.
- Maximum Base Price for House comes from Partial category- Home was not completed when last assessed (associated with New Homes) is higher than rest.
- Minimum base price comes from Normal condition sale and also highest sale price comes from this category.

## Surprise Housing - Housing Price Predication & Analysis Project



Observation: -

In above plot we can clearly see relation between all three feature very clearly. As total floor area increases the sale price also get increases corresponding the overall quality of House.

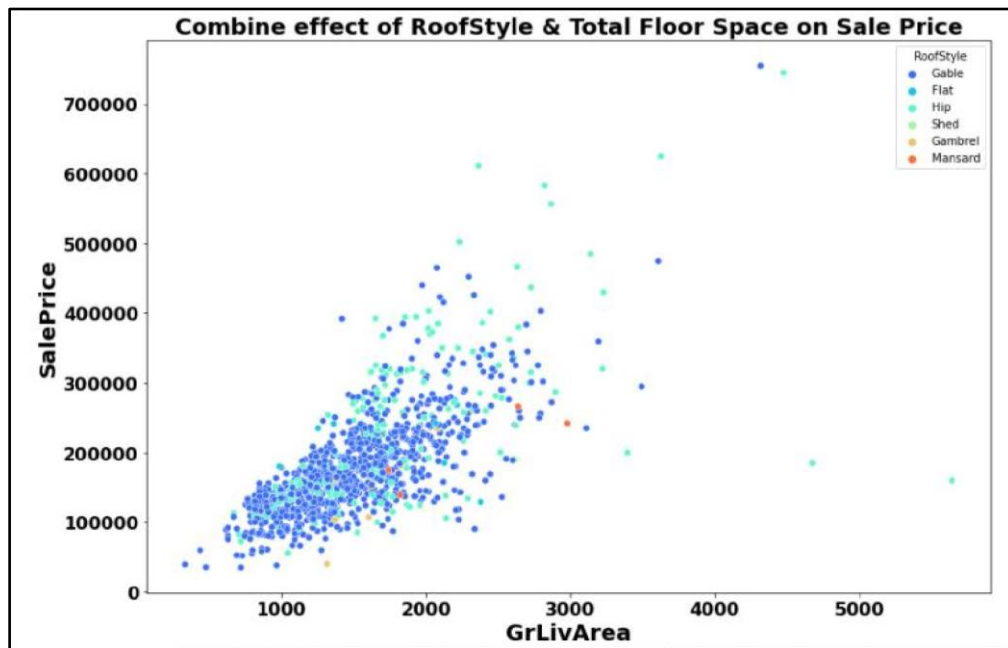


Observation:

- More than 75% House properties come with Gable Roof  
Style followed by around 15 % house properties with Hip Style.
- From Boxplot we can see that Hip style Roof are much costlier than remaining roof style.

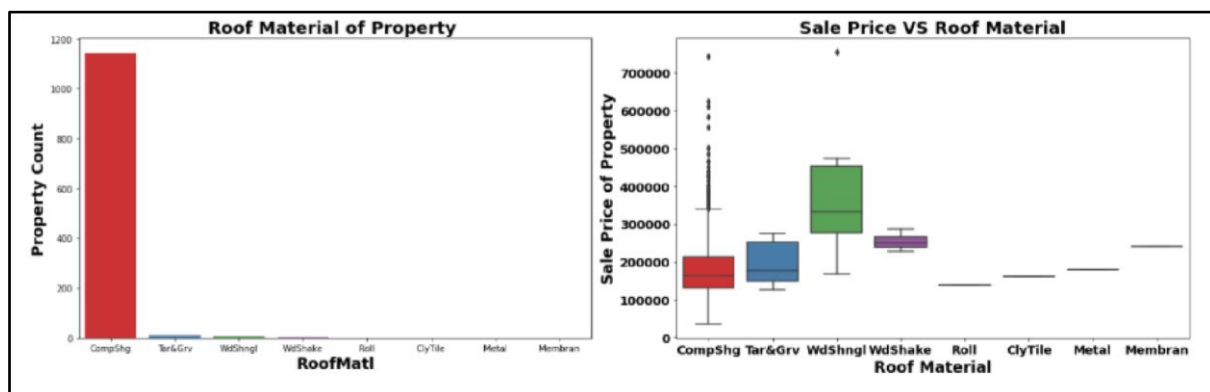


## Surprise Housing - Housing Price Predication & Analysis Project



Observation: -

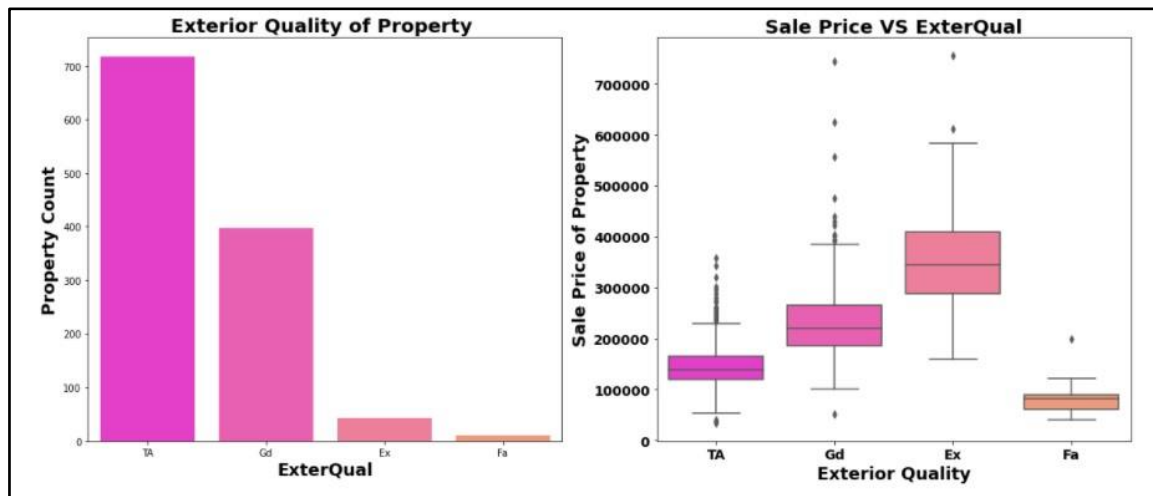
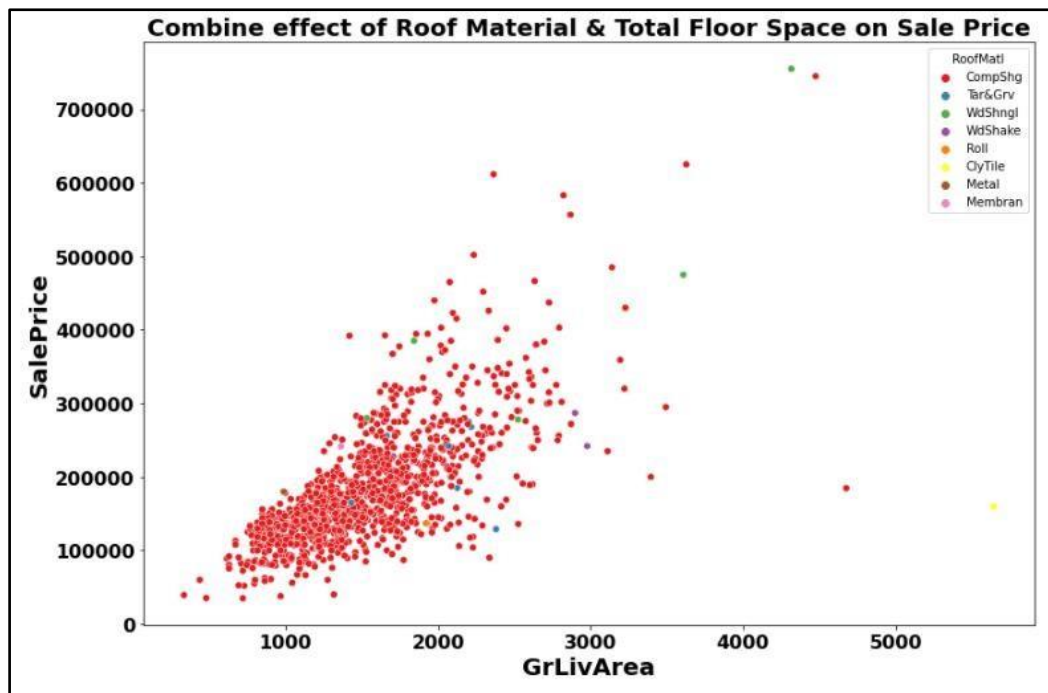
For High floor area construction mainly Hip style Roof is used and invariably high-cost properties mostly comes up with Hip Style Roof.



Observation: -

- More than 90% Properties in Data set made with roof material of Standard (Composite) Shingle.
- Wood Shingles is Costlier Material compare to rest.

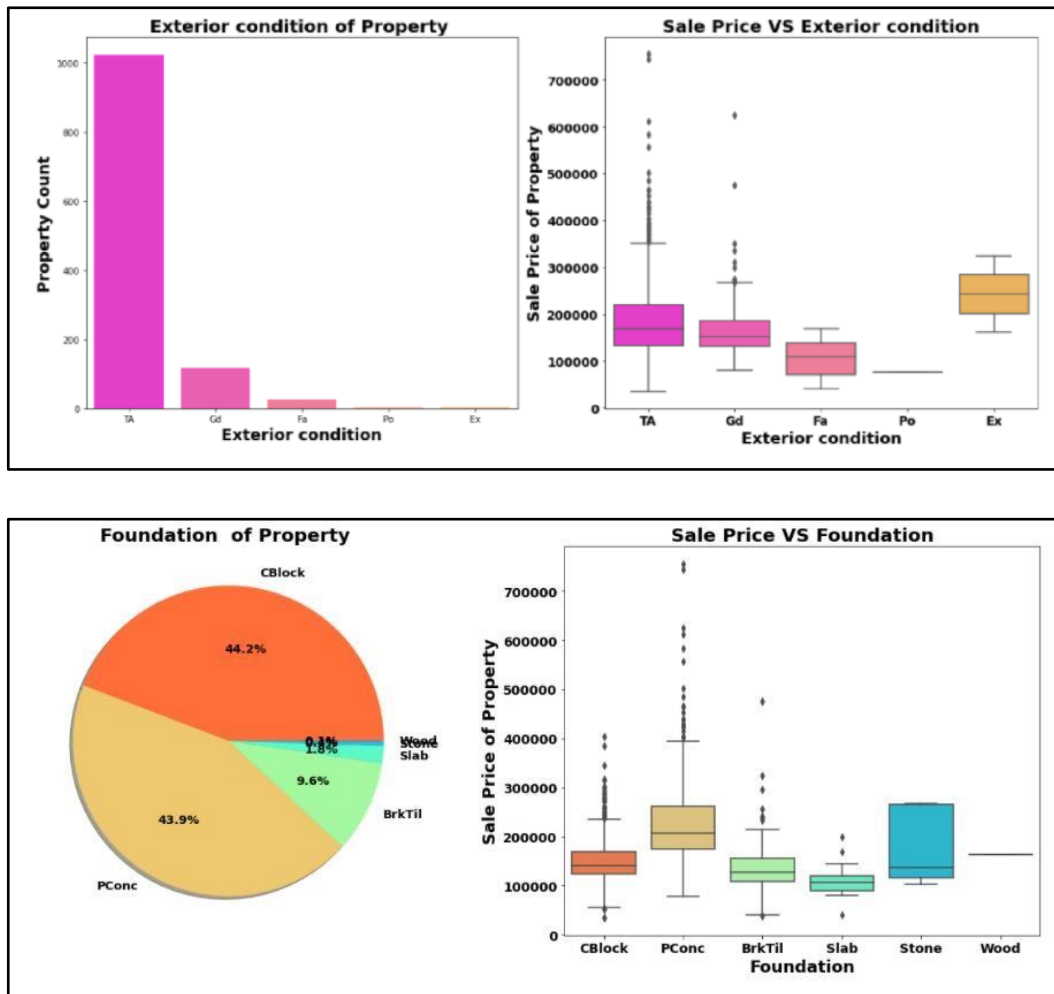
## Surprise Housing - Housing Price Predication & Analysis Project



Observation: -

- Around 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.

## Surprise Housing - Housing Price Predication & Analysis Project



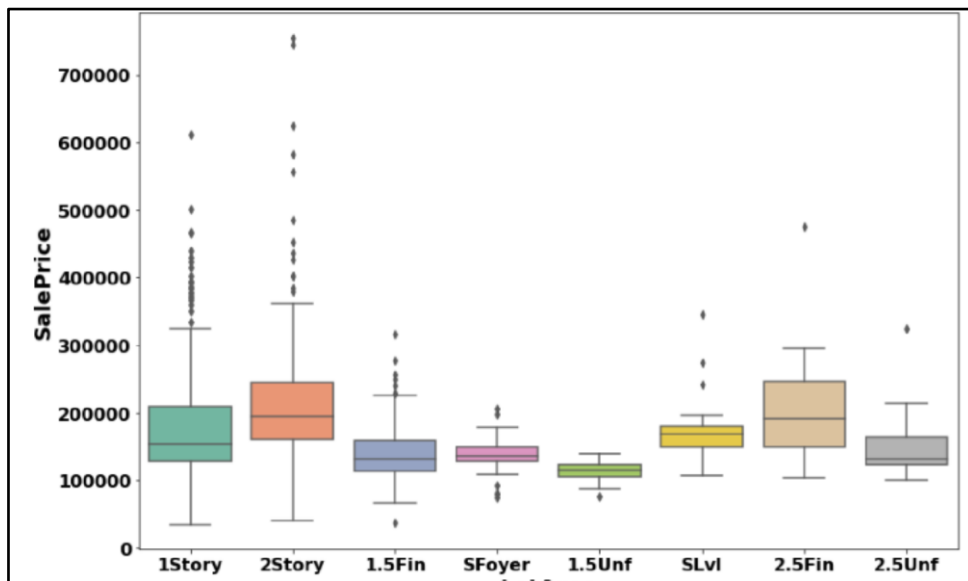
Observation: -

- 44.2% Properties with CBlock Foundation & 43.9% housing property come with PConc Foundation.

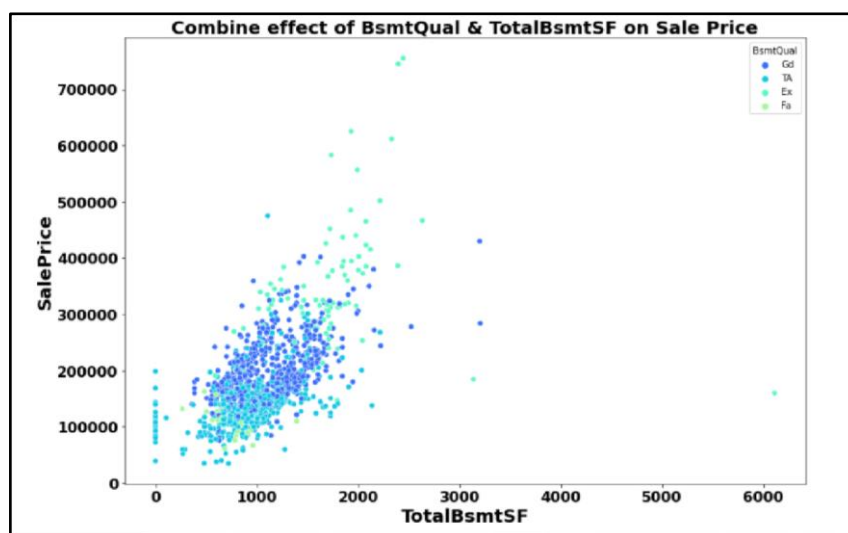
Observation: -

- As Basement Quality increase in relation to it, sale Price also get

increases.



- Two Story Building are costlier than remaining.
- Pconc Foundation are mostly use in costly housing properties.



## Chap 4. Conclusion

### 1. Key Findings and Conclusions of the Study

Algorithm	R2 Score CV Score		RMSE Value	Standard Deviation
Linear Regressor	.882	.789	25567	.052
Decision TreeRegressor	.788	.738	34209	.052
KNN Regression	.862	.790	27563	.030
Random Forest Regressor	.904	.829	22967	.035
SVR	.632	.595	45080	.038
Gradient Boosting Regression	.924	.853	20445	.035
AdaBoost Regressor	.824	.766	31123	.049

Here Gradient Boosting Regressor giving maximum R2 Score, minimum RMSE Value, Maximum CV Score and minimum Standard Deviation. So Gradient Boosting Regressor is selected as best model. The final R2 Score is 0.924 that is 92.4%.

### .2 Learning Outcomes of the Study in respect of Data Science

- Null removal is an important part of any problem.
- Scaling and standardization of data is mandatory.
- Feature selection played an important role in any ML problem. Unnecessary features and the features correlated with another needs to be removed.
- Most of the case accuracy score is improved after applying hyper parameter tuning.
- Overfitting check is an important part, it may effect on actual R2 Score.
- Data needs to be much precise and detailed for much better score.

### 2. Limitations of this work and Scope for Future Work

Some additional features can be added to the dataset. For this addition we can able to perform

some more advance model of ML.

- Artificial Neural Network can be used create more accurate model.
- The model will be more accurate if the dataset (set of observations) is bigger.

## Surprise Housing - Housing Price Predication & Analysis Project

- The model will be more accurate if there is no missing data.