Housing Market Prediction

Kishan Teli
Dept of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, USA
kteli@stevens.edu

Mansi Lokhande
Dept of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, USA
mlokhan1@stevens.edu

Abstract – In this project, a machine learning model is proposed to predict the price of house based on the data related to house. This project can help buyer to predict the actual price of the house according to the area, facilities and utilities that include in the house. We are using some regression techniques to reach the goal of the project and including every single step and output to state the project progress. We are using Python programming language with some Python libraries like numpy, sklearn, seaborne, etc. To build machine learning models we need to clean data and able to predict the price of house based on house features in order to analyze and compare model performance to choose the best model that can give accurate result.

Keywords – House price prediction, Regression, Machine Learning

I. INTRODUCTION

Everyday thousands of houses are sold, and every buyer have questions like : What is the actual price of the house that I am gone buy? Am I paying a good price? Price of house is depending on size of the house, the year it was built in, location, etc. For the prediction we need to look some statistical data and try to figure out which model will give us an accurate result to predict price. For that we need to study some models and select one of the models for our project to get the perfect result to reach the goal of this project.

To, accomplish, we have pulled the dataset from the Kaggle.

II. TOOLS USED

- Sklearn
- Numpy
- Matplotlib
- Pandas
- Jupyter Notebook
- Kaggle
- Seaborn

III. Git Repository

https://github.com/KishanTeli/Housing-Market-Prediction

IV. Literature Review

In this section, we look some recent studies that are related to this project topic and see how models were built and how results achieved in these studies. Ticknor' study, he used Bayesian regularized article neural network to predict the future operation of financial market and he developed a model to predict future stock prices. The model input is previous stock statistic and output of the model is the next day closing price of the stock that he used as an input.

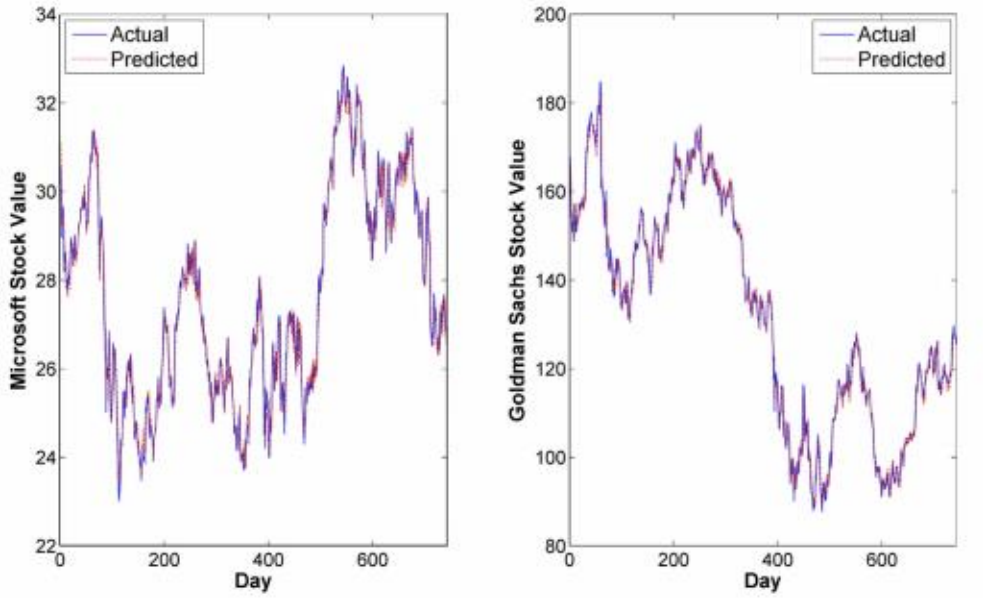


Figure 1. predicted vs actual price

A different study was done by Feng and Jones to predict house prices and they built a multilevel model (MLM) and an artificial neural network model (ANN). These models were compared to each other and with hedonic price model (HPM). The three models were tested using three scenarios and it shows that MLM model has better performance than other models with the accuracy of 0.75.

Data Collection

The data for this project is collected from the Kaggle.
https://www.kaggle.com/prevek18/ames-housing-dataset

The data consisted of the following:

- Ameshousing.csv – This contains the housing data with different features that related to house (the year built in, area code, street, condition, etc.)

Related Work

This project is divided into two sections, Data preparation and analysis from the data set, Mansi Lokhande has worked on Data preparation and Kishan Teli has worked on Model preparation for prediction.

a. – Data Preparation

In this project we used Ames housing dataset to predict the price of the house. In this figure 2., all columns of the datasets are shown below.

```
import pandas as pd
import numpy as np

dataset = pd.read_csv("AmesHousing.csv")
```

```
# Configuring float numbers format
pd.options.display.float_format = '{:20.2f}'.format
dataset.head(n=5)
```

	Order	PID	MS SubClass	MS Zoning	Lot Frontage	Lot Area	Street	Alley	Lot Shape	Land Contour	...	Pool Area	Pool QC	Fence	Misc Feature	Misc Val	Mo Sold	Yr Sold	Sale Type	Conc
0	1	526301100	20	RL	141.00	31770	Pave	NaN	IR1	Lvl	...	0	NaN	NaN	NaN	0	5	2010	WD	N
1	2	526350040	20	RH	80.00	11622	Pave	NaN	Reg	Lvl	...	0	NaN	MnPrv	NaN	0	6	2010	WD	N
2	3	526351010	20	RL	81.00	14267	Pave	NaN	IR1	Lvl	...	0	NaN	NaN	Gar2	12500	6	2010	WD	N
3	4	526353030	20	RL	93.00	11160	Pave	NaN	Reg	Lvl	...	0	NaN	NaN	NaN	0	4	2010	WD	N
4	5	527105010	60	RL	74.00	13830	Pave	NaN	IR1	Lvl	...	0	NaN	MnPrv	NaN	0	3	2010	WD	N

5 rows × 82 columns

Figure 2. dataset with all columns

We want to know the mean, the standard deviation, the minimum and the maximum for each numeric column in the dataset:

```
dataset.describe(include=[np.number], percentiles=[.5]) \
    .transpose().drop("count", axis=1)
```

	mean	std	min	50%	max
Order	1465.50	845.96	1.00	1465.50	2930.00
PID	714464496.99	188730844.65	526301100.00	535453620.00	1007100110.00
MS SubClass	57.39	42.64	20.00	50.00	190.00
Lot Frontage	69.22	23.37	21.00	68.00	313.00
Lot Area	10147.92	7880.02	1300.00	9436.50	215245.00
Overall Qual	6.09	1.41	1.00	6.00	10.00
Overall Cond	5.56	1.11	1.00	5.00	9.00

Data cleaning:

Machine learning models do not accept data with missing values for we need to consider the missing value and fill it with the mean value to use this data in our model to get the accurate results. In data cleaning we focus on different aspects of the data to make clean and useful data for our prediction and we attach every single steps and outputs.

Data cleaning is done using the below code and all missing values are replaced by mean value of respective columns.

```
# Getting the number of missing values in each column
num_missing = dataset.isna().sum()
# Excluding columns that contains 0 missing values
num_missing = num_missing[num_missing > 0]
# Getting the percentages of missing values
percent_missing = num_missing * 100 / dataset.shape[0]
# Concatenating the number and perecentage of missing values
# into one dataframe and sorting it
pd.concat([num_missing, percent_missing], axis=1,
          keys=['Missing Values', 'Percentage']).\
          sort_values(by="Missing Values", ascending=False)
```

	Missing Values	Percentage
Pool QC	2917	99.56
Misc Feature	2824	96.38
Alley	2732	93.24
Fence	2358	80.48
Fireplace Qu	1422	48.53

Pool QC:

Now consider Pool QC row, we can see that missing value percentage of Pool QC is 99.56 which is very high. We may think that 2917 houses do not have pool. To verify this and consider Pool QC column separately.

```
dataset["Pool Area"].value_counts()
```

```
0      2917
561       1
555       1
519       1
800       1
738       1
648       1
576       1
512       1
480       1
444       1
368       1
228       1
144       1
```

We can see that there are 2917 sections in Pool Area column that have an estimation of 0. This verifies our theory that each house without a pool has a missing value in Pool QC column of value 0 in Pool Area section. Therefore, lets fill missing values with “No pool” in Pool QC column by below code.

```
dataset["Pool QC"].fillna("No Pool", inplace=True)
```

Misc. Feature :

Now considering Misc. Feature row, we can see that missing value percentage of Pool QC is 96.38 which is very high. Misc. Feature column.

```
dataset["Misc Val"].value_counts()
```

```
0        2827
400        18
500        13
450         9
600         8
700         7
2000        7
650         3
1200        3
1500        3
4500        2
2500        2
480         2
3000        2
12500       1
300         1
350         1
8300        1
420         1
80          1
54          1
460         1
```

We can see that there are 2827 sections in Misc. Feature column that have an estimation of 0. This verifies our theory that each house without any miscellaneous feature has a missing value in Misc. feature column of value 0. Therefore, lets fill missing values with “No feature” Misc. Feature column by below code.

```
dataset['Misc Feature'].fillna('No feature', inplace=True)
```

Similar method is applied to Alley, Fence, and Fireplace Qu, Garage Cond, Garage Qual, Garage Finish, Garage Yr Blt, Garage Type, Garage Cars, and Garage Area, Bsmt Exposure, BsmtFin Type 2, BsmtFin Type 1, Bsmt Qual, Bsmt Cond, Bsmt Half Bath, Bsmt Full Bath, Total Bsmt SF, Bsmt Unf SF, BsmtFin SF 2, and BsmtFin SF 1, Mas Vnr Area and Mas Vnr Type.

```
dataset['Garage Cars'].fillna(0, inplace=True)
dataset['Garage Area'].fillna(0, inplace=True)

dataset.loc[~pd.isna(dataset['Garage Type']) &
            pd.isna(dataset['Garage Qual']), "Garage Type"] = "No Garage"

for col in ['Garage Type', 'Garage Finish', 'Garage Qual', 'Garage Cond']:
    dataset[col].fillna('No Garage', inplace=True)

dataset['Garage Yr Blt'].fillna(0, inplace=True)
```

Electrical:

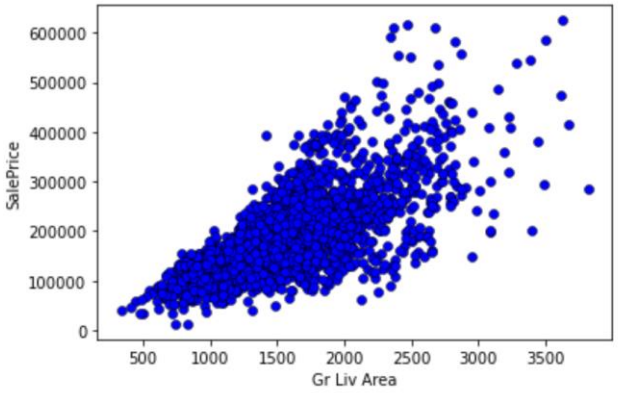
Electrical column has one missing value. Now we will fill this value with the mode of this column. Next, we will check if there are any missing values in the column. If we get value=0 that means, there are no missing values.

Outlier Removal:

The author De Cock of this dataset states that there are five unusual values and outliers in the dataset and recommend for removal of these outliers. This can be done by plotting Sale Price against Gr Liv Area to spot the outliers.

```
dataset = dataset[dataset["Gr Liv Area"] < 4000]
```

```
plt.scatter(x=dataset['Gr Liv Area'], y=dataset['SalePrice'],
            color="blue", edgecolors="#000000", linewidths=0.5);
plt.xlabel("Gr Liv Area"); plt.ylabel("SalePrice");
```



To reduce problems in modeling later, we will reset our dataset index by removing the outlier rows, so that no gaps remain in our dataset.

```
dataset.reset_index(drop=True, inplace=True)
```

Deleting some unimportant columns :

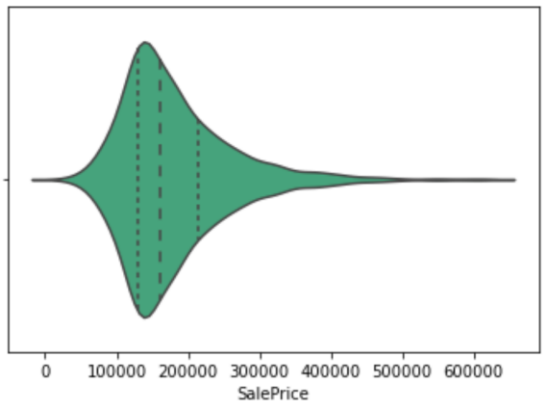
We will delete some unnecessary columns which are not required for further analysis. The column to be deleted are Order and PID

```
dataset.drop(['Order', 'PID'], axis=1, inplace=True)
```

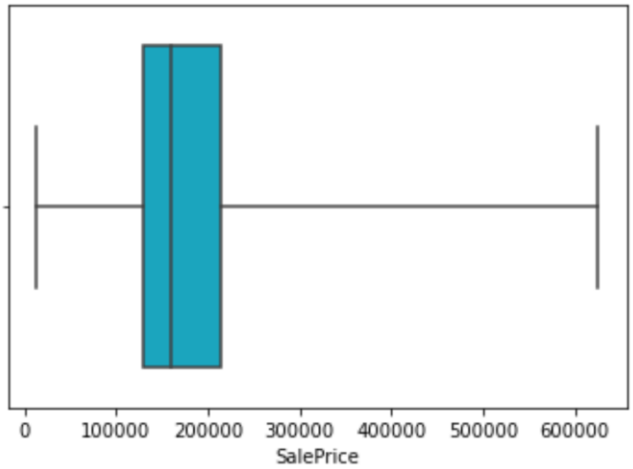
Target Variable Distribution:

Our dataset contains lot of variable, but we need to study the most important variable i.e. target variables for understanding the distribution. To begin with, we start by plotting the violin plot for the target variable. The width of the violin depicts the frequency. This means that if a violin is the widest between 300 and 400, then the area between 300 and 400 contains more data points than other areas:

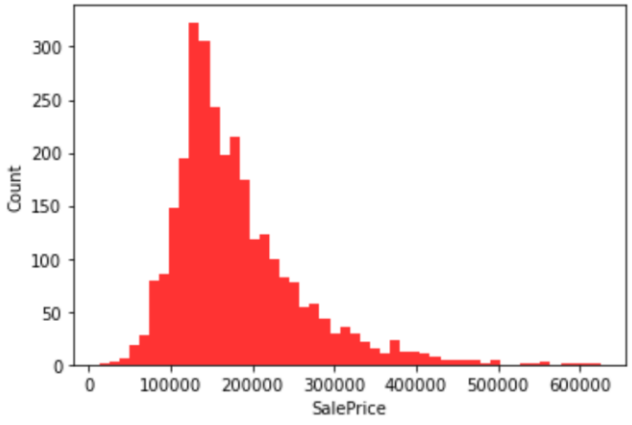
```
sns.violinplot(x=dataset['SalePrice'], inner="quartile", color="#36B37E");
```



We can see from the plot that most house costs fall somewhere in the range of 100,000 and 250,000. The dashed lines speak to the areas of the three quartiles Q1, Q2 (the middle), and Q3. Presently we should see the box plot of Sale Price:

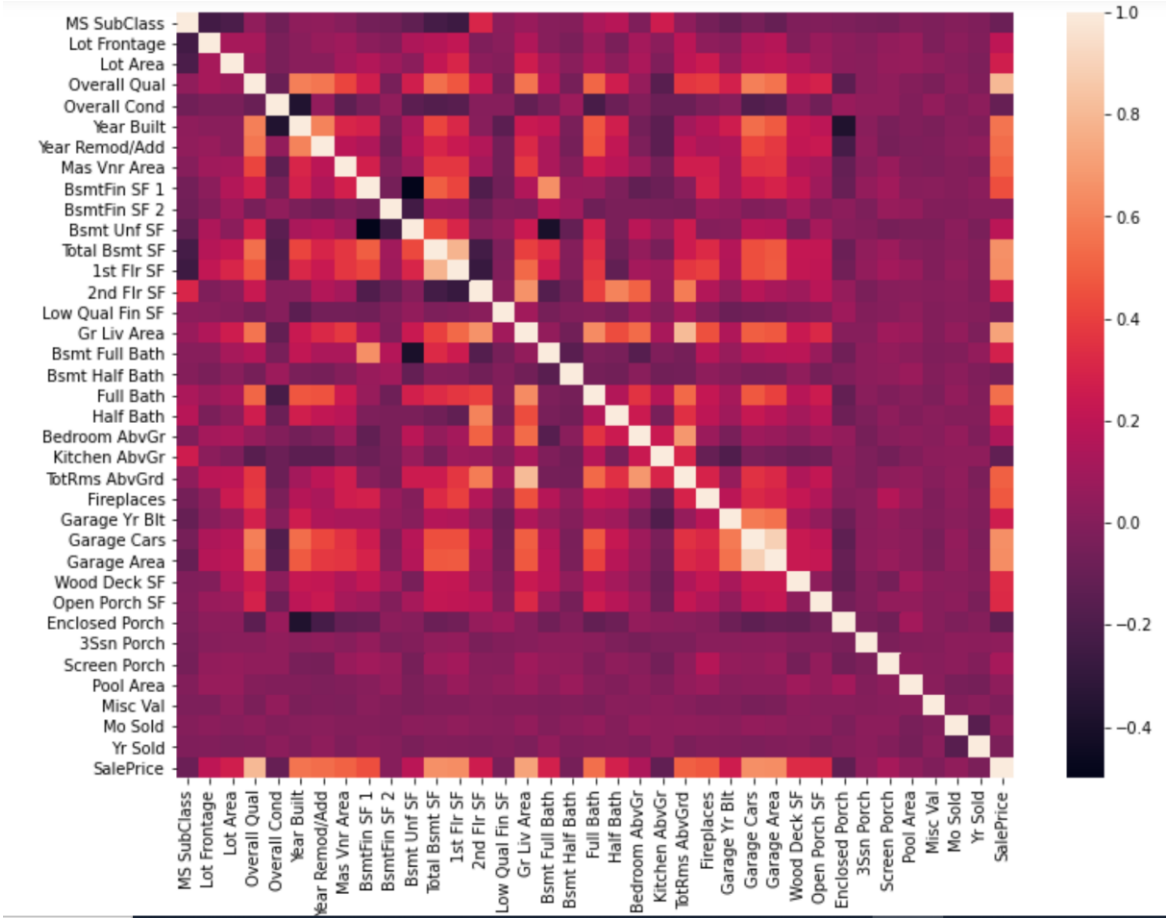


This shows us the minimum and maximum estimations of Sale Price. It shows us additionally the three quartiles spoke to by the box and the vertical line within it. Finally, we plot the histogram of the variable to see a more itemized perspective on the distribution:



Correlation between variables:

We need to see how the dataset variables are correlated with each other and how predictor variables are correlated with the target variable. Correlation is represented between -1 and +1 values. Where +1 depicts the highest positive correlation, -1 depicts the highest negative correlation, and 0 depicts that there is no correlation.



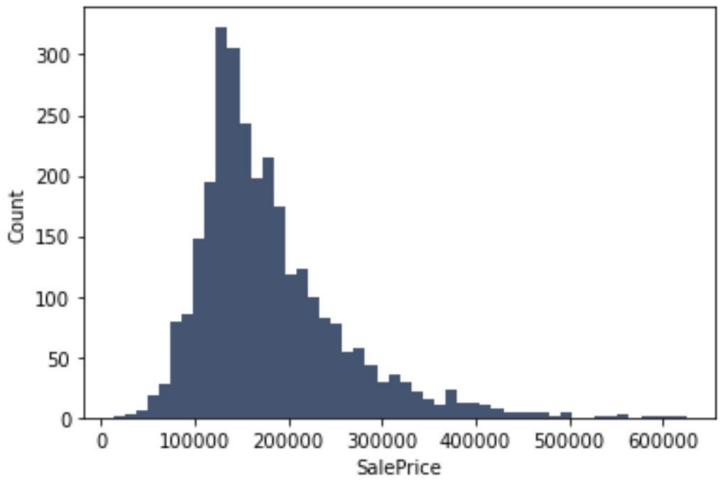
We can see that Garage Cars and Garage Area have high positive correlation, which is sensible because when the garage area increases, its car capacity increases too.

With respect to negative correlation, we can see that Bsmt Unf SF is negatively correlated with BsmtFin SF 1, and that makes sense because when we have more unfinished area, this means that we have less finished area.

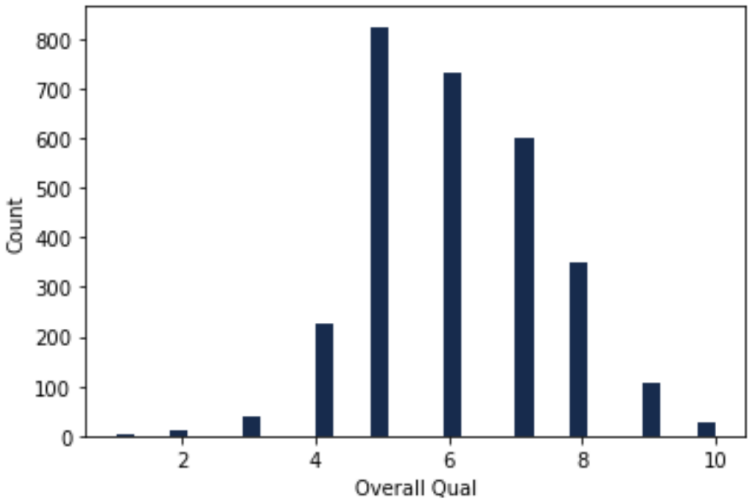
By observing at the last row of the heatmap, we can see that the target variable is highly positively correlated with Overall Qual and Gr Liv Area. We see also that the target variable is positively correlated with Year Built, Year Remod/Add, Mas Vnr Area, Total Bsmt SF, 1st Flr SF, Full Bath, Garage Cars, and Garage Area.

Relationship between Target variables and other variables:

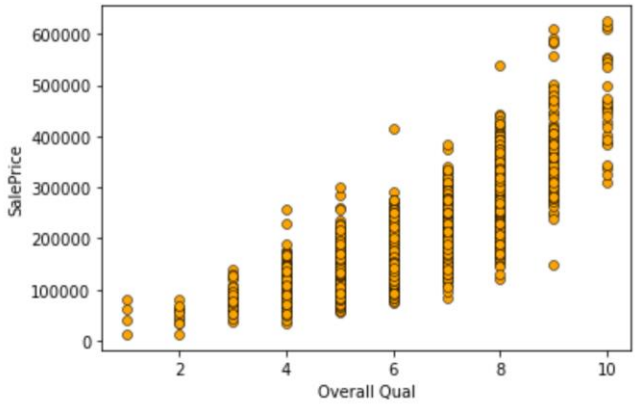
Let us begin with the relationship between the target variable and Overall Qual, but before that, let's see the distribution of each of them. Target variable Sale Price:



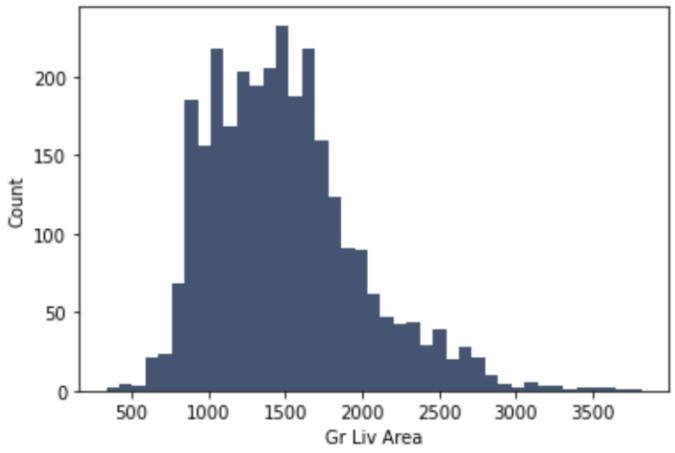
We can observe that most house prices fall between the range of 100,000 and 200,000.There are number of expensive houses to the right of the plot. Now, let us see the distribution of Overall Qual variable:



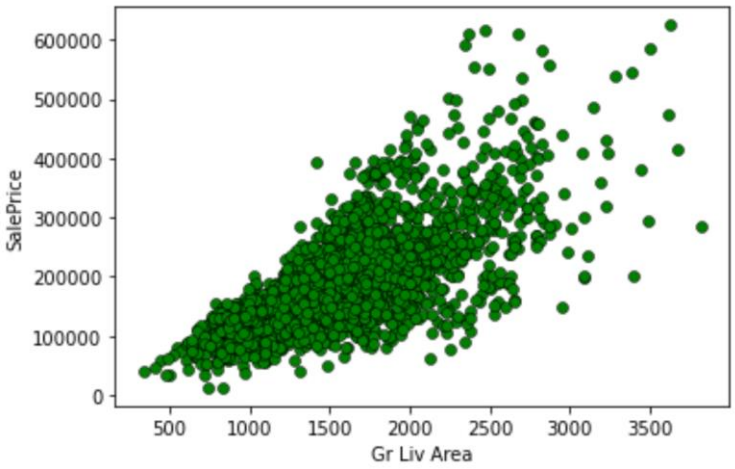
We see that Overall Qual takes a whole number value somewhere in the range of 1 and 10, and that most houses have a general value somewhere in the range of 5 and 7. Presently, let us we plot the disperse plot of SalePrice and Overall Qual to see the connection between them:



We can see that they are truly positively correlated; generally, as the overall quality increases, the sale price increases too.



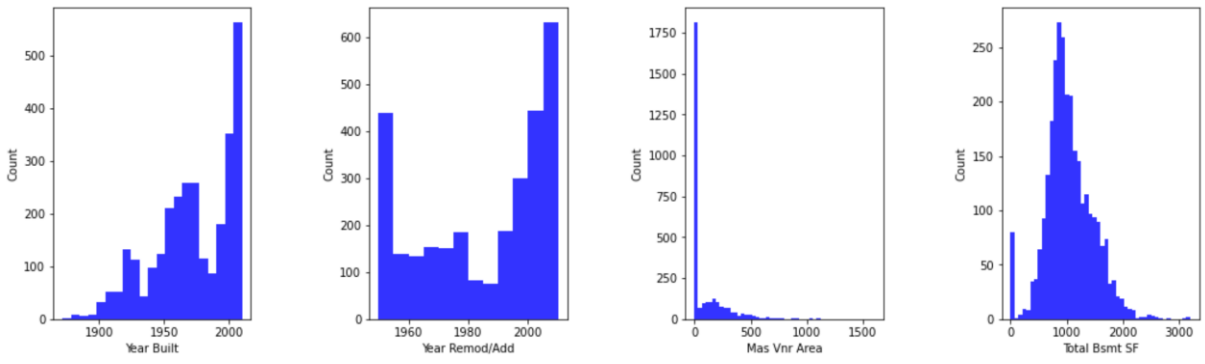
We can see that the above-ground living area falls approximately between 800 and 1800.



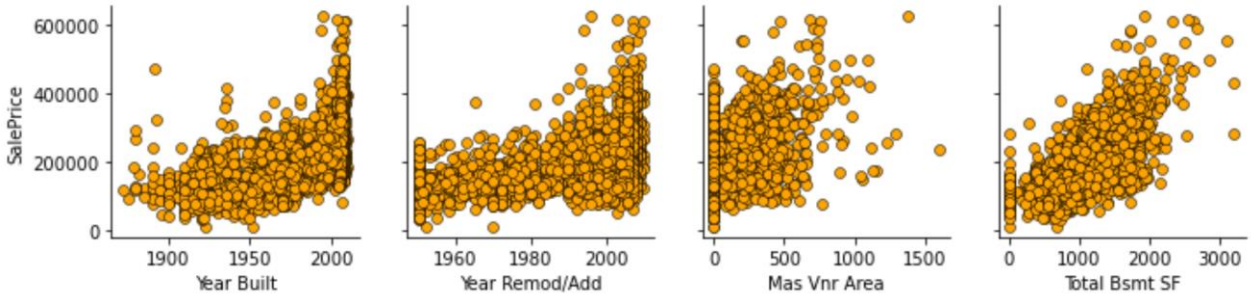
Moderate positive Correlation :

For this we want to imagine the relationship between the target variable and the variables that are positively correlated with it, but the correlation is not very strong.

Let us see the distribution of first four variables:

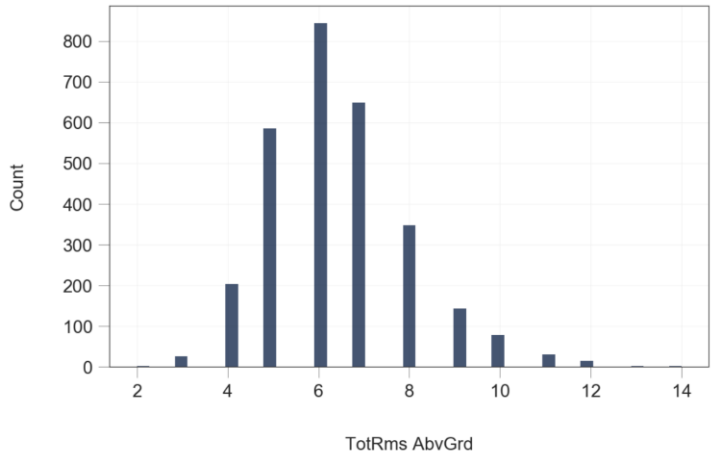


Now let us see the relationships with the target variable using scatter plots:

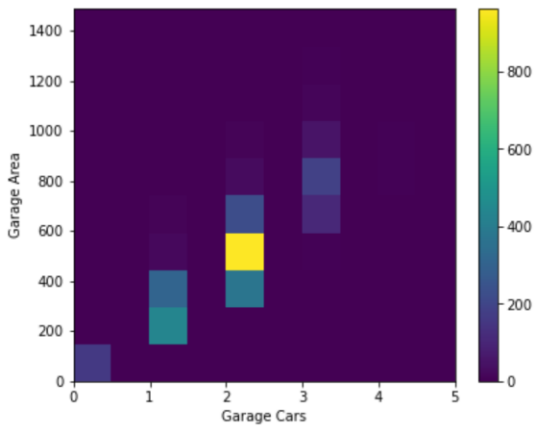


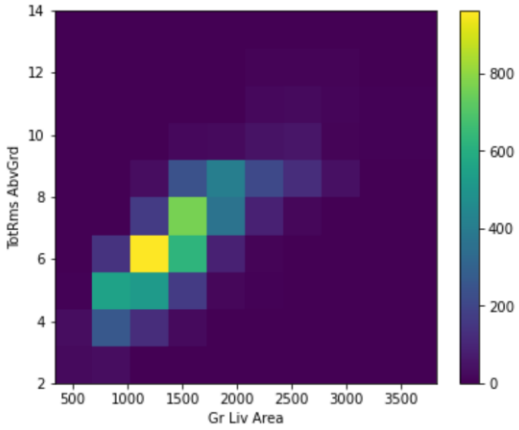
Relationships Between Predictor Variables:

Positive Correlation : Let us see the distribution of TotRms AbvGrd first



Now let us visualize the relationship between Garage Cars and Garage Area and between Gr Liv Area and TotRms AbvGrd:

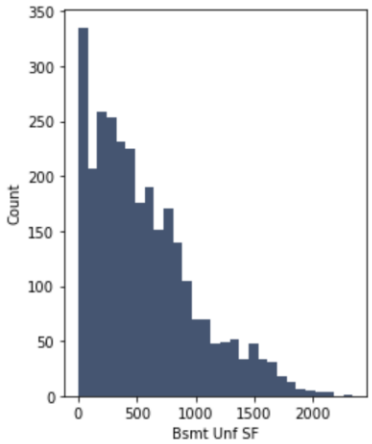


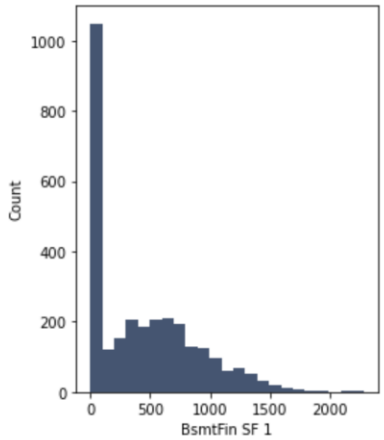


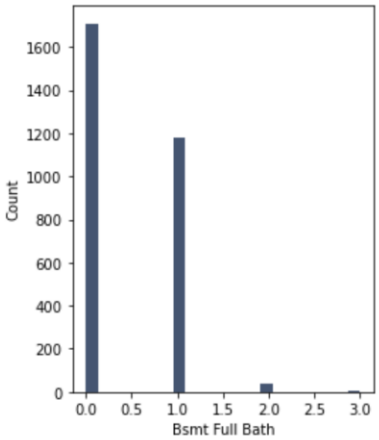
We can see strong correlation between each pair.

Negative Correlation:

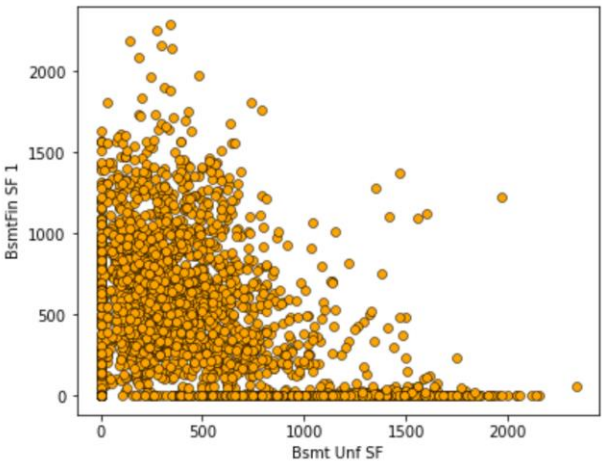
We discovered a significant negative correlation between Bsmt Unf SF and BsmtFin SF 1, and between Bsmt Unf SF and Bsmt Full Bath in the heatmap. We also want to visualize these correlations. Let us see the distribution of these variables first:

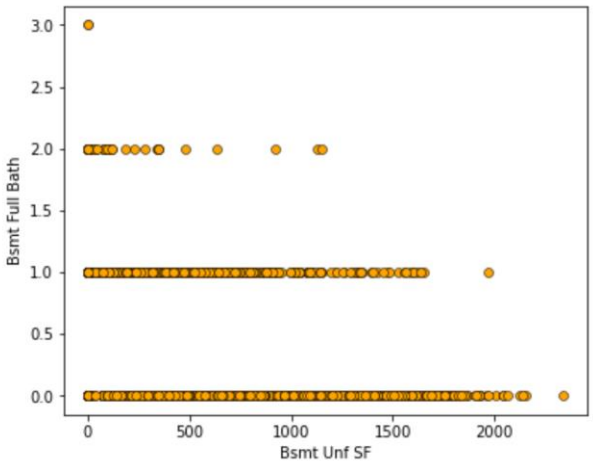






Now, plot in scatter plot





We can see the negative correlation between each pair of these variables, this information got from exploratory data analysis in this section, which will be used in feature engineering in the next section.

Feature Engineering:

We will utilize the bits of knowledge from Exploratory Data Analysis area to design the highlights of our dataset.

Creating New Derived Feature:

We will determine a feature whose values are the squares of original values, and another feature whose values are the cubes of original values. Further, we will create a feature whose values are the product of our two features values:

```
for f in ["Overall Qual", "Gr Liv Area"]:
    dataset[f + "_p2"] = dataset[f] ** 2
    dataset[f + "_p3"] = dataset[f] ** 3
dataset["OverallQual_GrLivArea"] = \
    dataset["Overall Qual"] * dataset["Gr Liv Area"]
```

Likewise, we saw that there are some predictor features that are highly correlated with each other. To remove the Multicollinearity problem, we will delete one feature from each pair of highly correlated predictors.

```
dataset.drop(["Garage Cars", "TotRms AbvGrd"], axis=1, inplace=True)
```

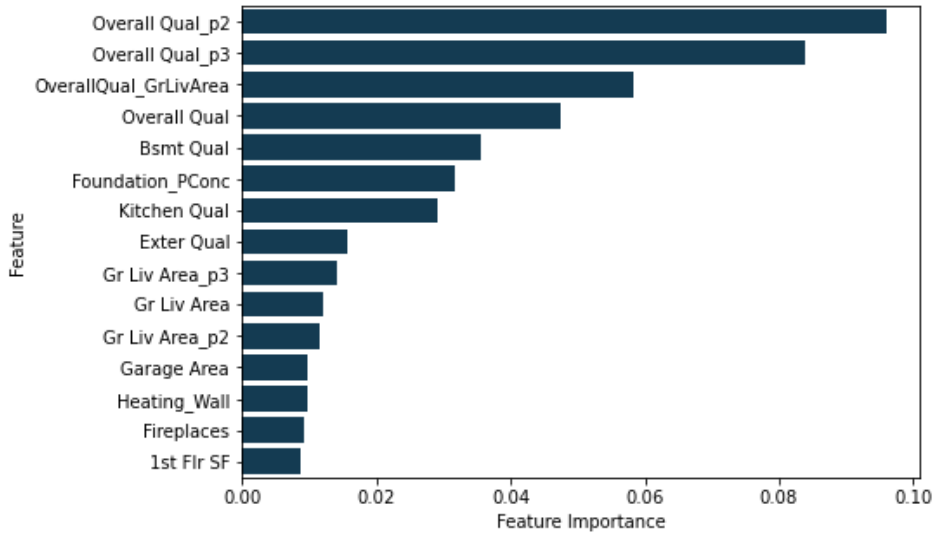
Feature Importance:

Some of the models we used provide the ability to see the importance of each feature in the dataset after fitting the model. We will look at the feature importance provide by both XGBoost and Random Forest models. We have 242 features in our data which is a big number, so we will take a look at the 15 most important features.

XGBoost:

```
xgb_feature_importances = xgb_model.feature_importances_
xgb_feature_importances = pd.Series(
    xgb_feature_importances, index=X_train.columns.values
    ).sort_values(ascending=False).head(15)

fig, ax = plt.subplots(figsize=(7, 5))
sns.barplot(x=xgb_feature_importances,
            y=xgb_feature_importances.index,
            color="#003f5c");
plt.xlabel('Feature Importance');
plt.ylabel('Feature');
```

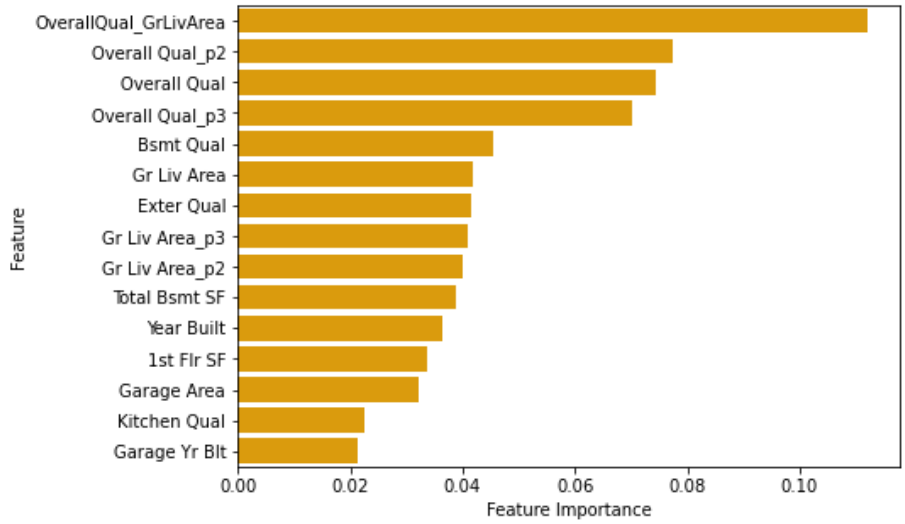


Random Forest:

Important feature as for Random Forest model :

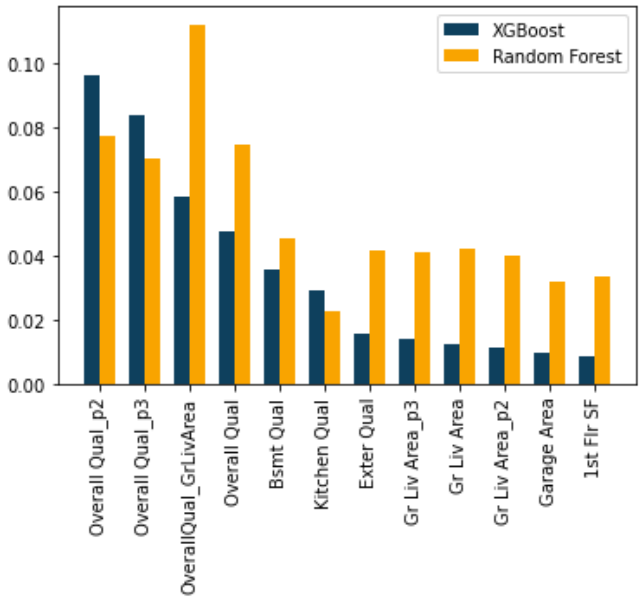
```
rf_feature_importances = rf_model.feature_importances_
rf_feature_importances = pd.Series(
    rf_feature_importances, index=X_train.columns.values
    ).sort_values(ascending=False).head(15)

fig, ax = plt.subplots(figsize=(7,5))
sns.barplot(x=rf_feature_importances,
            y=rf_feature_importances.index,
            color="#ffa600");
plt.xlabel('Feature Importance');
plt.ylabel('Feature');
```



Common important Features:

Which features are among the most important features for both XGBoost and Random Forest models and find out the difference in their importance regarding the two models:



V. Conclusion :

We build several regression models to predict the price of some house given some of the house features. We evaluated and compared each model to determine the on with highest performance. We also looked a how some models rank the features according to their importance.

VI. References :

1. Feng, Y., & Jones, K. (2015, July). Comparing multilevel modelling and artificial neural networks in house price prediction. In Spatial Data Mining and Geographical Knowledge Services (ICSDM), 2015 2nd IEEE International Conference on (pp. 108-114). IEEE.
2. Ticknor, J. L. (2013). A Bayesian regularized artificial neural network for stock market forecasting. Expert Systems with Applications, 40(14), 5501-5506.
3. https://www.kaggle.com/erick5/predicting-house-prices-with-machine-learning
4. https://pbiecek.github.io/xai_stories/story-house-sale-prices.html