

***HOUSING PRICE PREDICTION PROJECT***

**Submitted by:**

**Sudhanshu Kumar**

# ACKNOWLEDGMENT

I would like to thank Flip Robo Technologies for providing me with the opportunity to work on this project from which I have learned a lot. I am also grateful to my SME Mr. Shubham Yadav for his constant guidance and support.

Some of the reference sources are as follows:

* Coding Ninjas
* Medium.com
* StackOverflow
* Kaggle.com
* Github.com
* Stack-Overflow
* Scikit-learn.org
* Learning.datatrained.com

# INTRODUCTION

## BUSINESS PROBLEM FRAMING

This is a real estate problem where a **US based** housing company named **Surprise Housing** has decided to **invest in Australian Market**. Their agenda is to buy houses in Australia at prices below their actual value in the market and sell them at high prices to gain profit. To do this this company uses data analytics to decide in which property they must invest.

Company has collected the data of previously sold houses in Australia and with the help of this data they want to know to the value of prospective properties to decide whether it will suitable to invest in the properties or not.

To know the value of properties company has provided data to us to do data analysis and to extract the information of attributes which are important to predict the price of the houses. They want a machine learning model which can predict the price of houses and also the significance of each important attribute in house prediction i.e, how and to what intensity each variable impacts the price of the house.

## CONCEPTUAL BACKGROUND OF THE DOMAIN PROBLEM

In real estate the value of property usually increases with time as seen in many countries. One of the causes for this is due to rising population.

The value of property also depends on the proximity of the property, its size its neighbourhood and audience for which the property is subjected to be sold. For example if audience is mainly concerned of commercial purpose. Then the property which is located in densely populated area will be sold very fast and at high prices compared to the one located at remote place. Similarly if audience is concerned only on living place then property with less dense area having large area with all services will be sold at higher prices.

The company is looking at prospective properties to buy houses to enter the market. We 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.

## REVIEW OF LITERATURE

Houses are one of the necessary needs of each 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.

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.

We 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.

## MOTIVATION FOR THE PROBLEM UNDERTAKEN

To understand real world problems where Machine Learning and Data Analysis can be applied to help organizations in various domains to make better decisions with the help of which they can gain profit or can be escaped from any loss which otherwise could be possible without the study of data .One of such domain is Real Estate.

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

# ANALYTICAL PROBLEM FRAMING

## MATHEMATICAL/ ANALYTICAL MODELING OF THE PROBLEM

In this project we have performed various mathematical and statistical analysis such a

Graphical user interface, application

Description automatically generated

From this statistical analysis we make some of the interpretations that,

* Maximum standard deviation of 8957.44 is observed in LotArea column.
* Maximum SalePrice of a house observed is 755000 and minimum is 34900.
* In the columns Id, MSSubclass, LotArea, MasVnrArea, BsmtFinSF1, BsmtFinSF2, BsmtUnfsF, TotalBsmtSF, 1stFlrSF, 2ndFlrSF, LowQualFinSF, GrLivArea, BsmtFullBath, HalfBath, TotRmsAbvGrd, WoodDeckSF, OpenPorchSF, EnclosedPorch, 3SsnPorch, ScreenPorch, PoolArea, Miscval, salePrice mean is considerably greater than median so the columns are positively skewed.
* In the columns FullBath, BedroomAbvGr, Fireplaces, Garagecars, GarageArea, YrSold Median is greater than mean so the columns are negatively skewed.
* In the columns Id, MSSubClass, LotFrontage, LotArea, MasVnrArea, BsmtFinSF1, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF, 1stFlrSF, 2ndFlrSF, LowQualFinSF, GrLivArea, BsmtHalfBath, BedroomAbvGr, ToRmsAbvGrd, GarageArea, WoodDeckSF, OpenPorchSF, EnclosedPorch, 3SsnPorch, ScreenPorch, PoolArea, MiscVal, SalePrice there is considerable difference between the 75 percentile and maximum, so outliers are present.

## DATA SOURCES AND THEIR FORMATS

The variable features of this problem statement are as:

MSSubClass: Identifies the type of dwelling involved in the sale

MSZoning: Identifies the general zoning classification of the sale

LotFrontage: Linear feet of street connected to property

LotArea: Lot size in square feet

Street: Type of road access to property

Alley: Type of alley access to property

LotShape: General shape of property

LandContour: Flatness of the property

Utilities: Type of utilities available

LotConfig: Lot configuration

LandSlope: Slope of property

Neighborhood: Physical locations within Ames city limits

Condition1: Proximity to various conditions

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

BldgType: Type of dwelling

HouseStyle: Style of dwelling

OverallQual: Rates the overall material and finish of the house

OverallCond: Rates the overall condition of the house

YearBuilt: Original construction date

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

RoofStyle: Type of roof

RoofMatl: Roof material

Exterior1st: Exterior covering on house

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

MasVnrType: Masonry veneer type

MasVnrArea: Masonry veneer area in square feet

ExterQual: Evaluates the quality of the material on the exterior

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

Foundation: Type of foundation

BsmtQual: Evaluates the height of the basement

BsmtCond: Evaluates the general condition of the basement

BsmtExposure: Refers to walkout or garden level walls

BsmtFinType1: Rating of basement finished area

BsmtFinSF1: Type 1 finished square feet

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

BsmtFinSF2: Type 2 finished square feet

BsmtUnfSF: Unfinished square feet of basement area

TotalBsmtSF: Total square feet of basement area

Heating: Type of heating

HeatingQC: Heating quality and condition

CentralAir: Central air conditioning

Electrical: Electrical system

1stFlrSF: First Floor square feet

2ndFlrSF: Second floor square feet

LowQualFinSF: Low quality finished square feet (all floors)

GrLivArea: Above grade (ground) living area square feet

BsmtFullBath: Basement full bathrooms

BsmtHalfBath: Basement half bathrooms

FullBath: Full bathrooms above grade

HalfBath: Half baths above grade

Bedroom: Bedrooms above grade (does NOT include basement bedrooms)

Kitchen: Kitchens above grade

KitchenQual: Kitchen quality

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

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

Fireplaces: Number of fireplaces

FireplaceQu: Fireplace quality

GarageType: Garage location

GarageYrBlt: Year garage was built

GarageFinish: Interior finish of the garage

GarageCars: Size of garage in car capacity

GarageArea: Size of garage in square feet

GarageQual: Garage quality

GarageCond: Garage condition

PavedDrive: Paved driveway

WoodDeckSF: Wood deck area in square feet

OpenPorchSF: Open porch area in square feet

EnclosedPorch: Enclosed porch area in square feet

3SsnPorch: Three season porch area in square feet

ScreenPorch: Screen porch area in square feet

PoolArea: Pool area in square feet

PoolQC: Pool quality

Fence: Fence quality

MiscFeature: Miscellaneous feature not covered in other categories

MiscVal: $Value of miscellaneous feature

MoSold: Month Sold (MM)

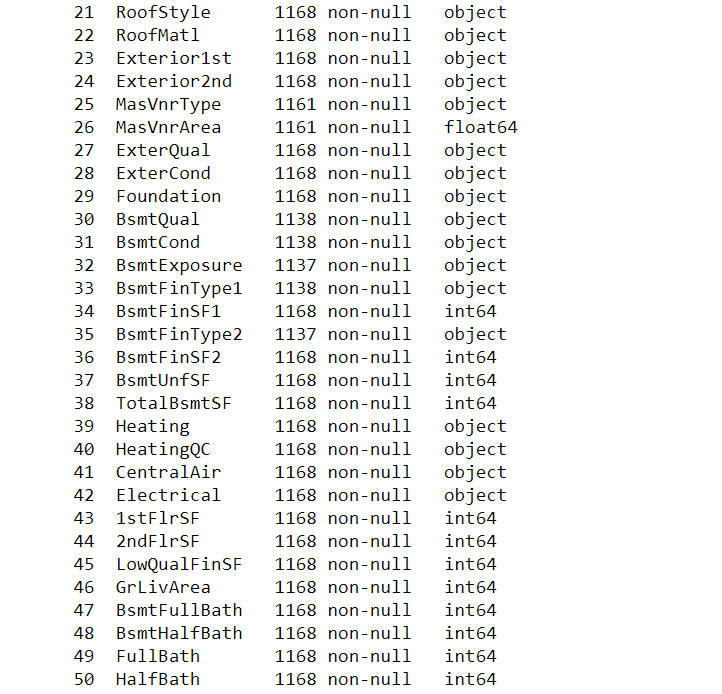
YrSold: Year Sold (YYYY)

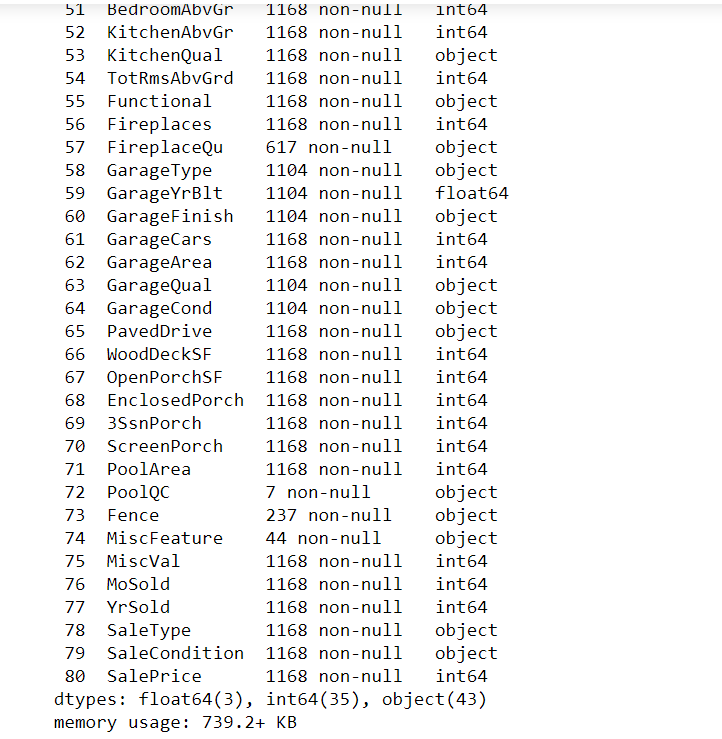
SaleType: Type of sale

SaleCondition: Condition of sale

Graphical user interface, application, Word

Description automatically generated





Graphical user interface, text, application

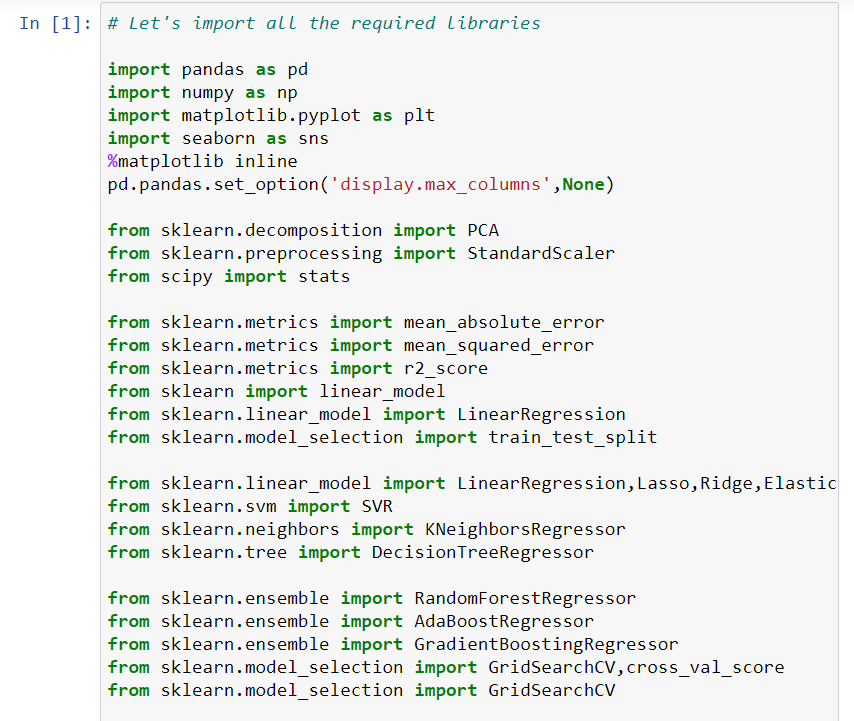
Description automatically generated

## DATA PREPROCESSING DONE

After loading all the required libraries we loaded the data into our jupyter notebook.

Graphical user interface

Description automatically generated



Feature Engineering has been used for cleaning of the data. Some unused columns have been deleted and even some columns have been bifurcated which was used in the prediction. We first done data cleaning. We first looked percentage of values missing in columns then we imputed missing values.

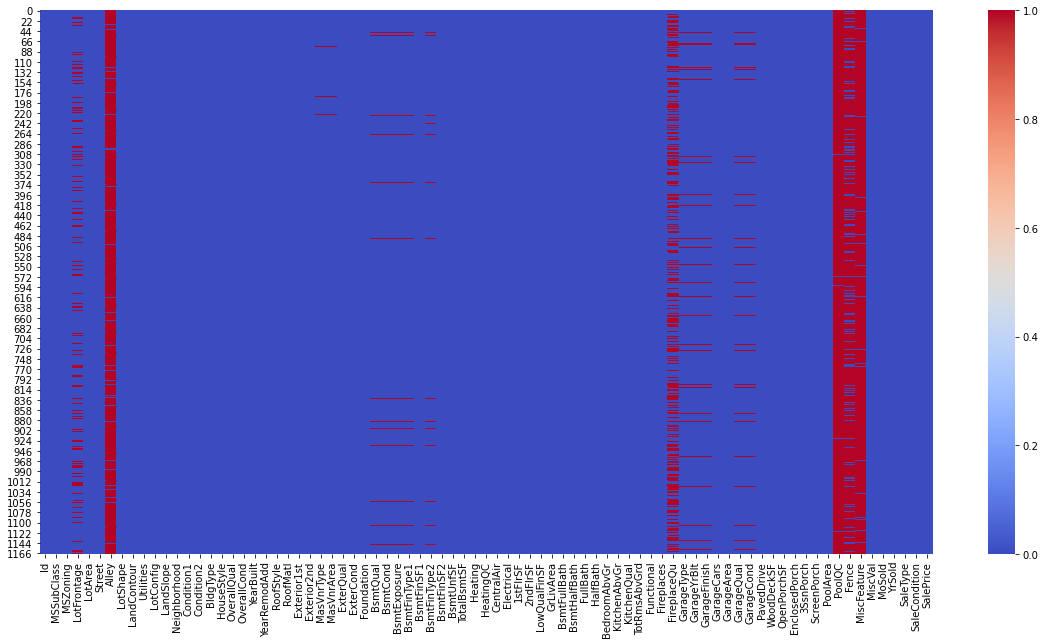
Graphical user interface, text

Description automatically generated

**Heat Map for missing Value**

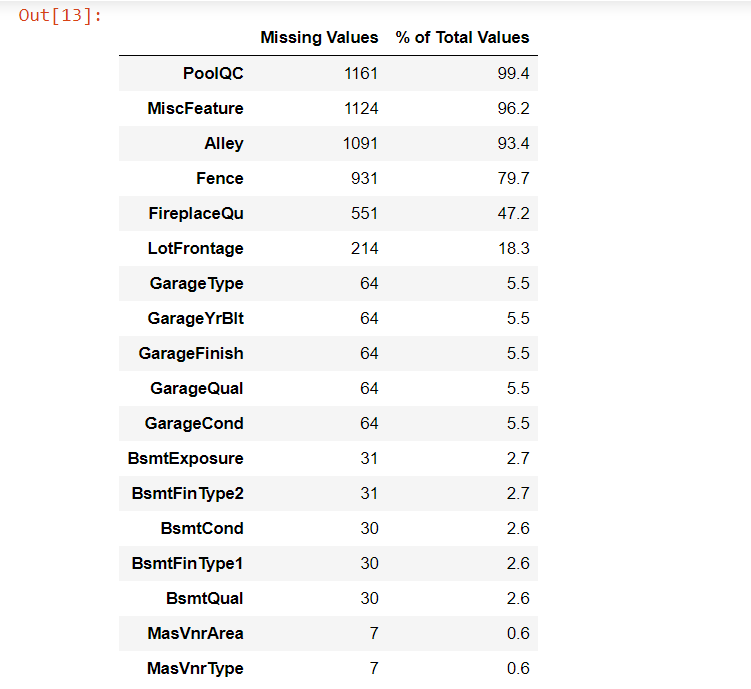
plt.figure(figsize=(20,10))

sns.heatmap(df.isnull(),xticklabels=True, cmap="coolwarm")



Graphical user interface, text, application

Description automatically generated



Graphical user interface, application

Description automatically generatedHere we can see that the columns PoolQC, MiscFeature, Alley and Fence has the maximum number of the null values/ missing values

Graphical user interface, text, application, email

Description automatically generated

We observed that there is only one unique value present in Utilities so will be dropping this column. Then we encoded all the categorical columns into numerical columns using dummy variables.

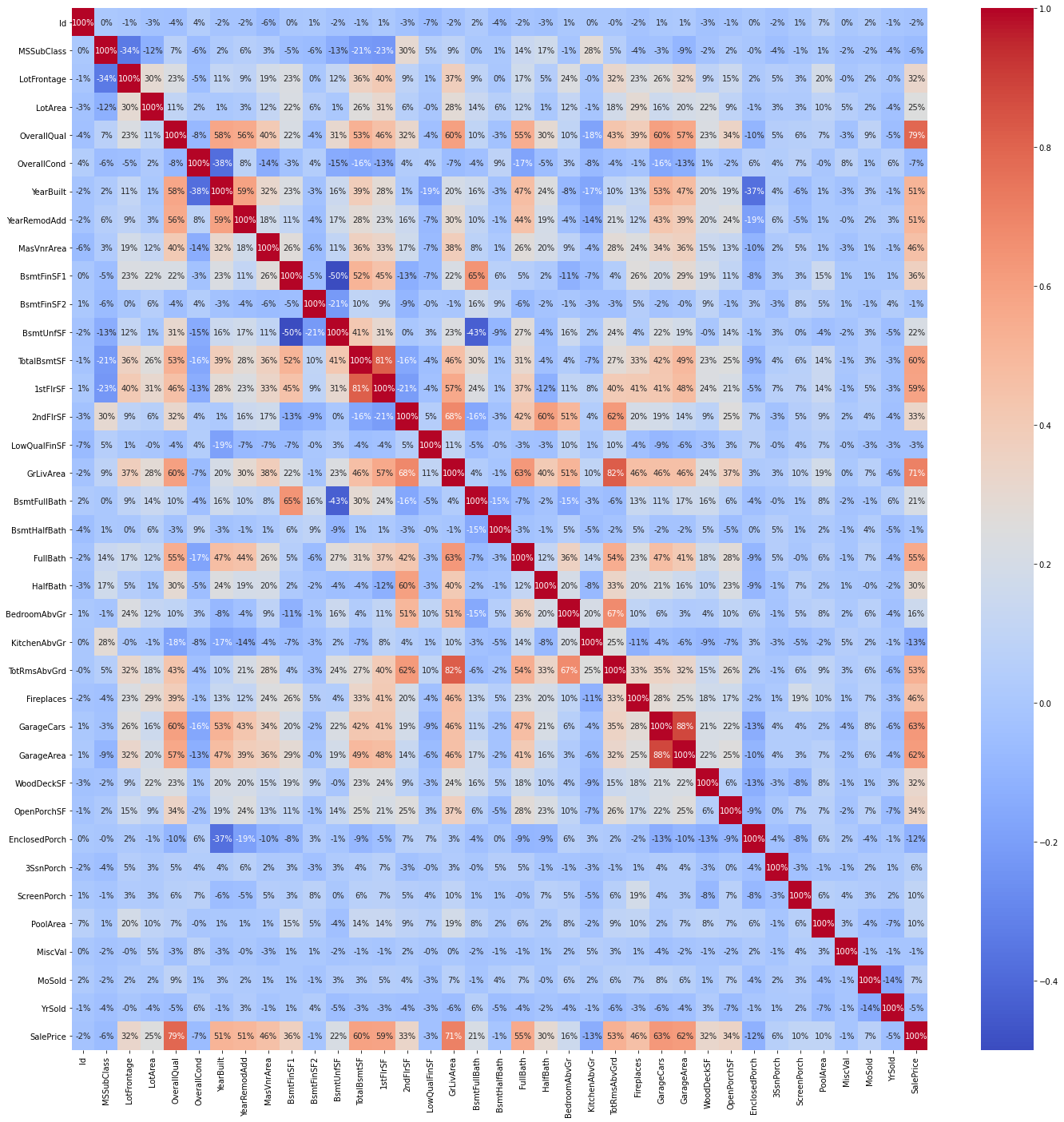
Then we checked the correlation with the help of heatmap.

# Let's plot the heat map

plt.figure(figsize=(24,24))

sns.heatmap(corr,annot=True,fmt='.0%',cmap='coolwarm')

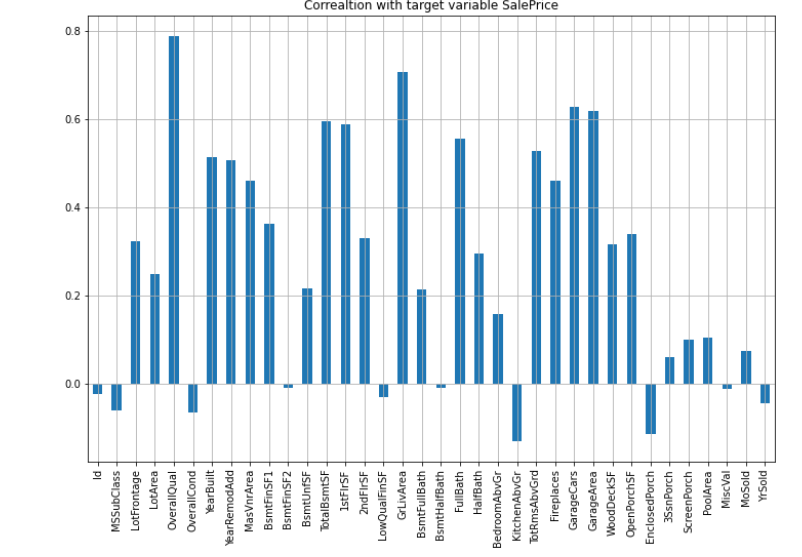
plt.show()



While checking the heatmap of correlation we observed that:

* SalePrice is highly positively correlated with the columns OverallQual, YearBuilt, YearRemodAdd, TotalBsmtSF, 1stFlrSF, GrLivArea, FullBath, TotRmsAbvGrd, GarageCars, GarageArea.
* SalePrice is negatively correlated with OverallCond, KitchenAbvGr, Encloseporch, YrSold.
* We observe multicollinearity in between columns so we will be using Principal Component Analysis(PCA).
* No correlation has been observed between the column Id and other columns so we will be dropping this column.

## DATA INPUTS- LOGIC- OUTPUT RELATIONSHIPS

Here we check the correlation between all our feature variables with target variable label

1. The column OverallQual is most positively correlated with SalePrice.

2. The column KitchenAbvGrd is most negatively correlated with SalePrice.

Set of assumptions related to the problem under consideration

By looking into the target variable label we assumed that it was a Regression type of problem.

We observed multicollinearity in between columns so we assumed that we will be using Principal Component Analysis (PCA).

We also observed that only one single unique value was present in Utilities column so we assumed that we will be dropping these columns.

## HARDWARE AND SOFTWARE REQUIREMENTS AND TOOLS USED

***HARDWARE:***

Graphical user interface, text, application

Description automatically generated

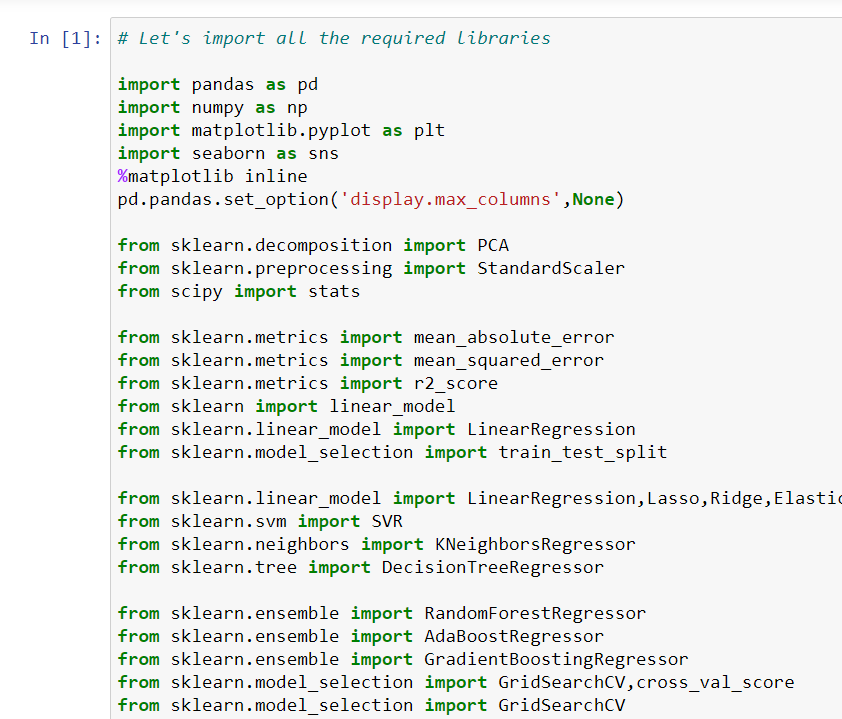
***SOFTWARE:***

Jupyter Notebook (Anaconda 3) – Python 3.7.6

Microsoft Excel 2016

***LIBRARIES:***

The tools, libraries, and packages we used for accomplishing this project are pandas, numpy, matplotlib, seaborn, scipy stats, sklearn.decomposition pca, sklearn standardscaler, GridSearchCV, joblib.



***From sklearn.preprocessing import StandardScaler***

As these columns are different in scale, they are standardized to have common scale while building machine learning model. This is useful when you want to compare data that correspond to different units.

***from sklearn.preprocessing import Label Encoder***

Label Encoder  and One Hot Encoder. These two encoders are parts of the SciKit Learn library in Python, and they are used to convert categorical data, or text data, into numbers, which our predictive models can better understand.

***from sklearn.model\_selection import train\_test\_split,cross\_val\_score***

Train\_test\_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, Sklearn train\_test\_split will make random partitions for the two subsets.

Through pandas library we loaded our csv file ‘Data file’ into dataframe and performed data manipulation and analysis.

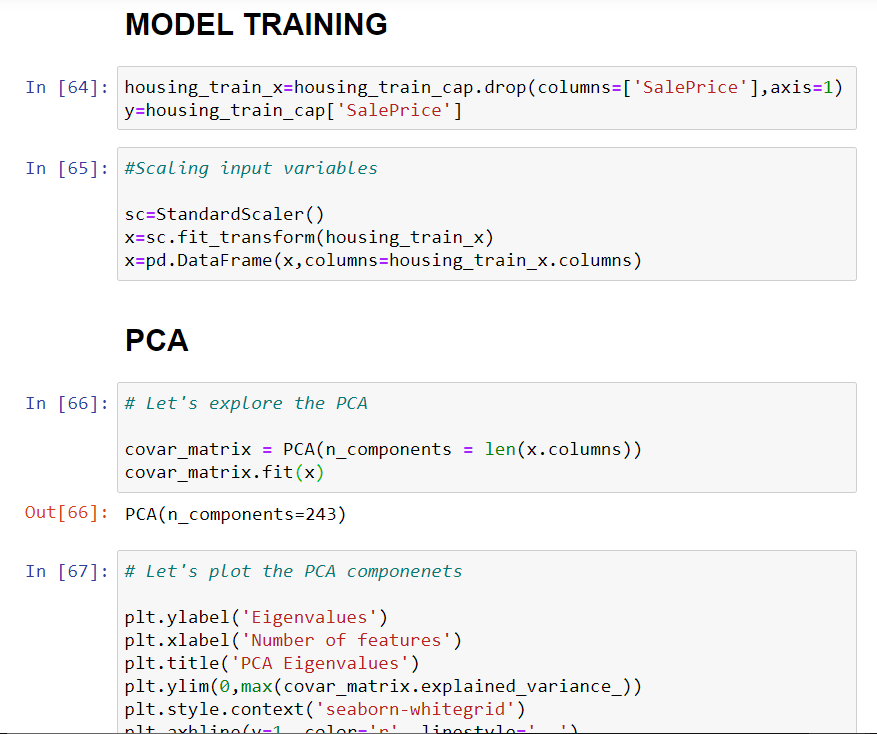
With the help of numpy we worked with arrays.

With the help of matplotlib and seaborn we did plot various graphs and figures and done data visualization.

With scipy stats we treated outliers through winsorization technique.

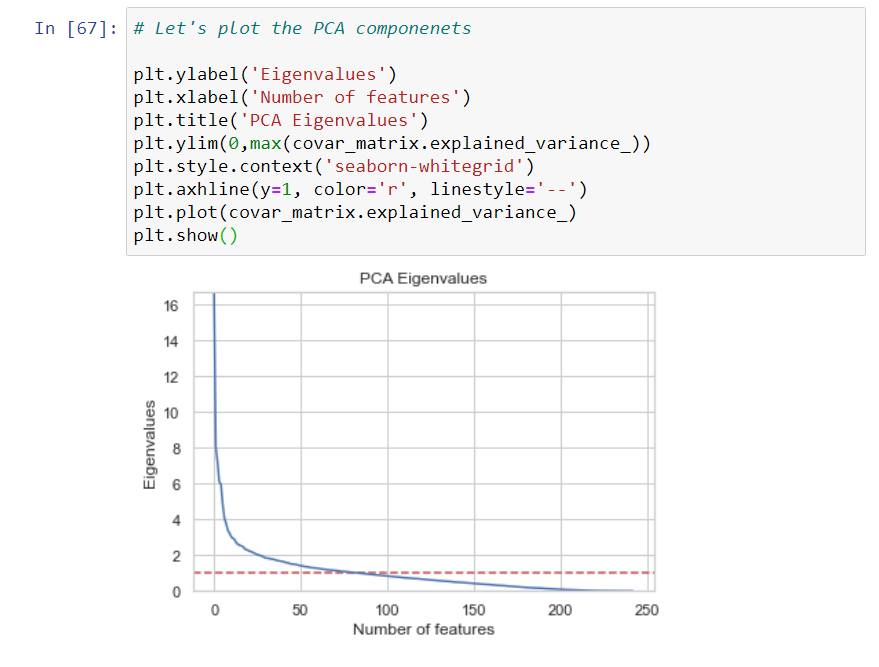
With sklearn.decomposition’s pca package we reduced the number of feature variables from 256 to 100 by plotting scrre plot with their Eigenvalues and chose the number of columns on the basis of their nodes.

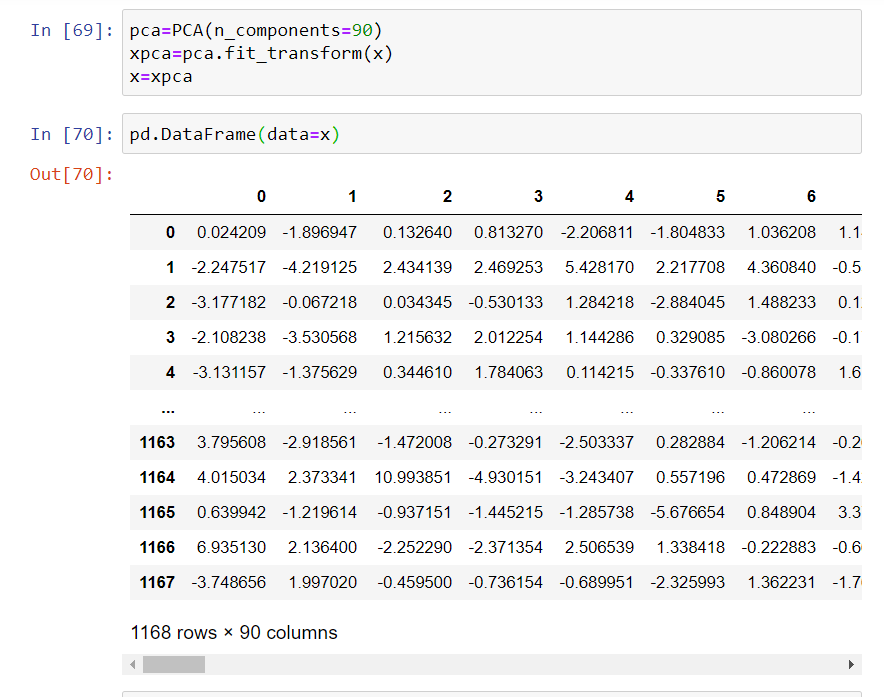
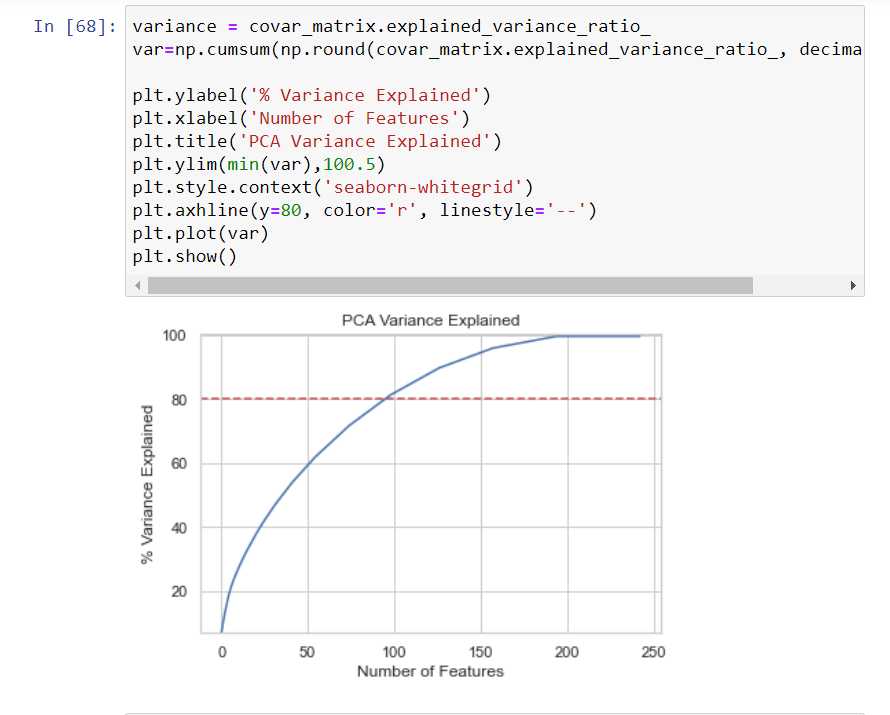
With sklearn’s standardscaler package we scaled all the feature variables onto single scale.



Graphical user interface, text, application, email

Description automatically generated





Through GridSearchCV we were able to find the right parameters for hyperparameter tuning. Through joblib we saved our model in csv format.

# MODEL/S DEVELOPMENT AND EVALUATION

## IDENTIFICATION OF POSSIBLE PROBLEM-SOLVING APPROACHES (METHODS)

We first converted all our categorical variables to numeric variables with the help of dummy variables to checkout and dropped the columns which we felt were unnecessary.

We observed skewness in data so we tried to remove the skewness through treating outliers with winsorization technique.

The data was improper scaled so we scaled the feature variables on a single scale using sklearn’s StandardScaler package.

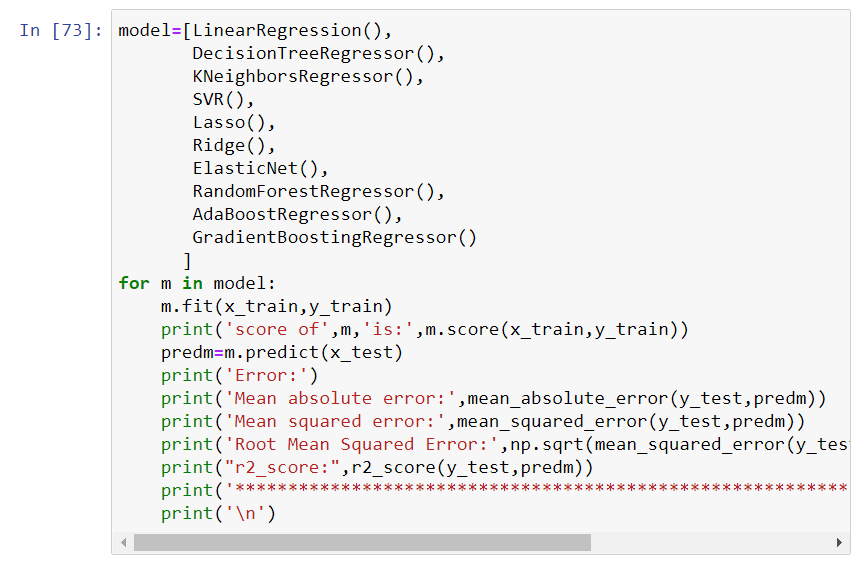
There were too many (256) feature variables in the data so we reduced it to 100 with the help of Principal Component Analysis(PCA) by plotting Eigenvalues and taking the number of nodes as our number of feature variables.

## TESTING OF IDENTIFIED APPROACHES (ALGORITHMS)

The algorithms we used for the training and testing are as follows:-

* Linear Regression
* Lasso
* Ridge
* Elastic Net
* SVR
* KNeighbors Regressor
* Decision Tree Regressor
* Random Forest Regressor
* Ada Boost Regressor
* Gradient Boosting Regressor

## RUN AND EVALUATE SELECTED MODELS



Text

Description automatically generated

Graphical user interface, text

Description automatically generated with medium confidence

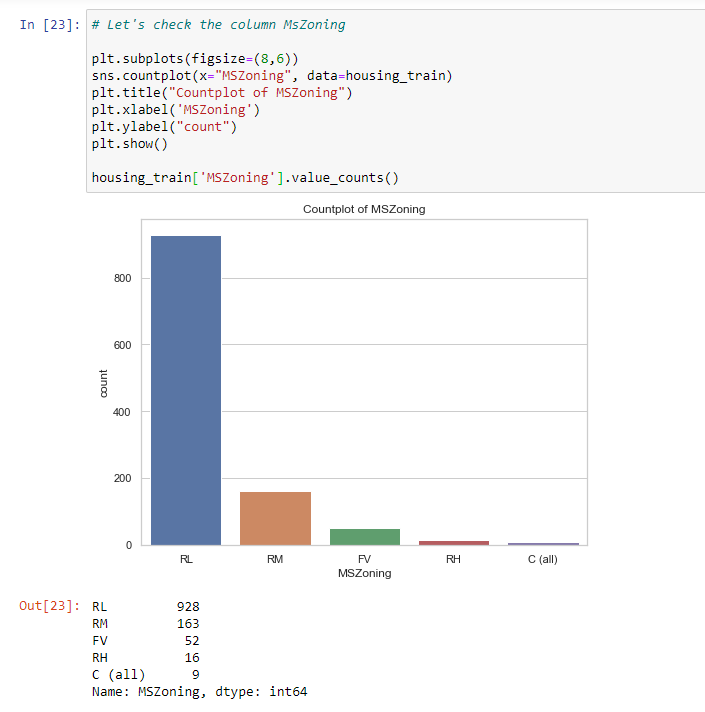
## KEY METRICS FOR SUCCESS IN SOLVING PROBLEM UNDER CONSIDERATION

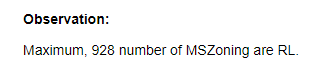
We used the metric Root Mean Squared Error by selecting the Gradient Boosting Regressor model which was giving us best(minimum) RMSE score.

## VISUALIZATIONS

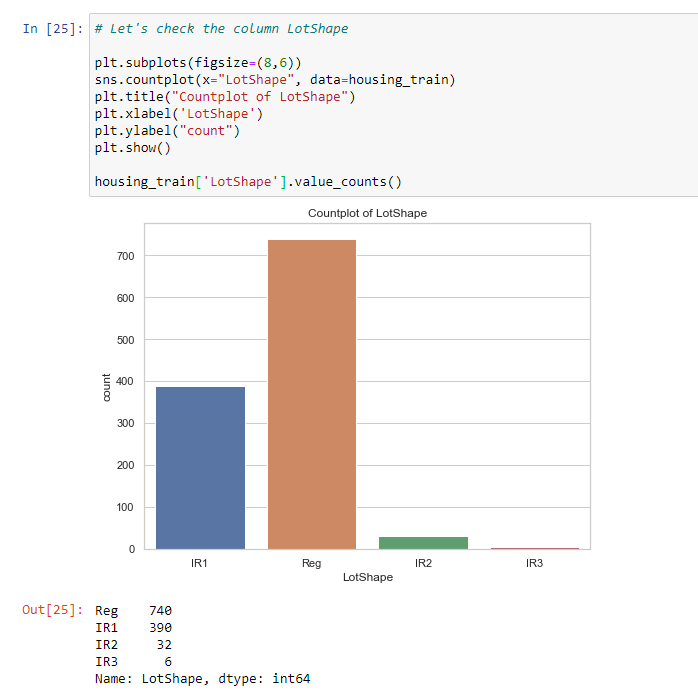
Chart

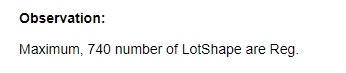
Description automatically generated



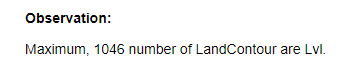


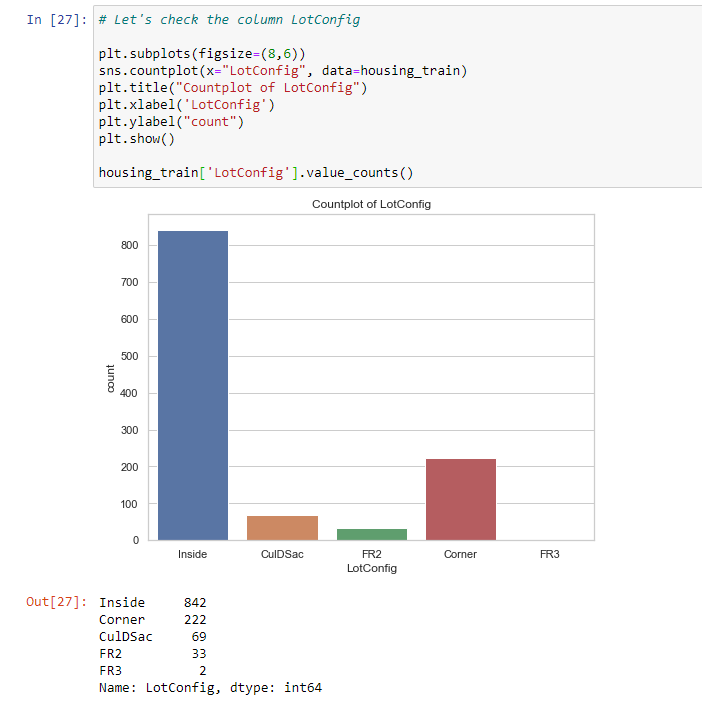


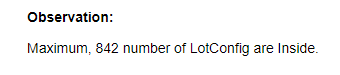






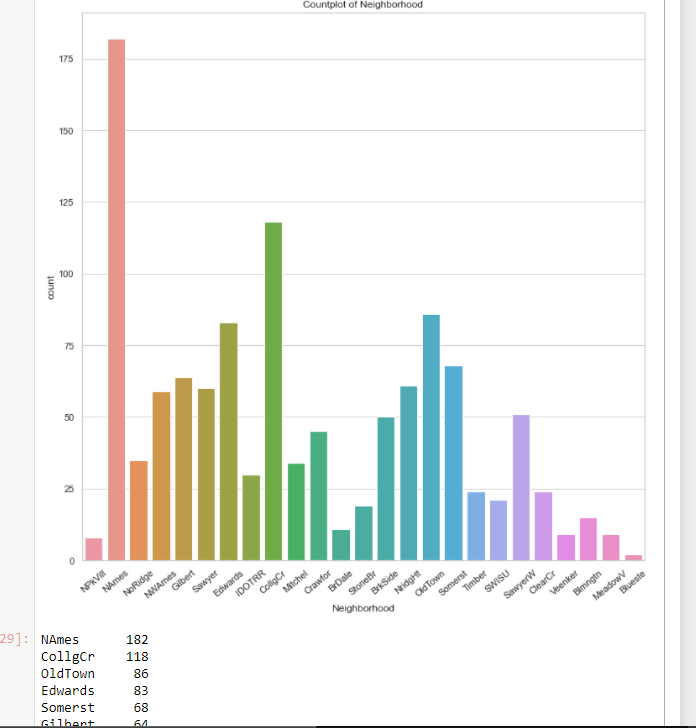


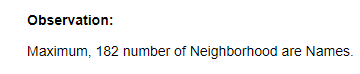


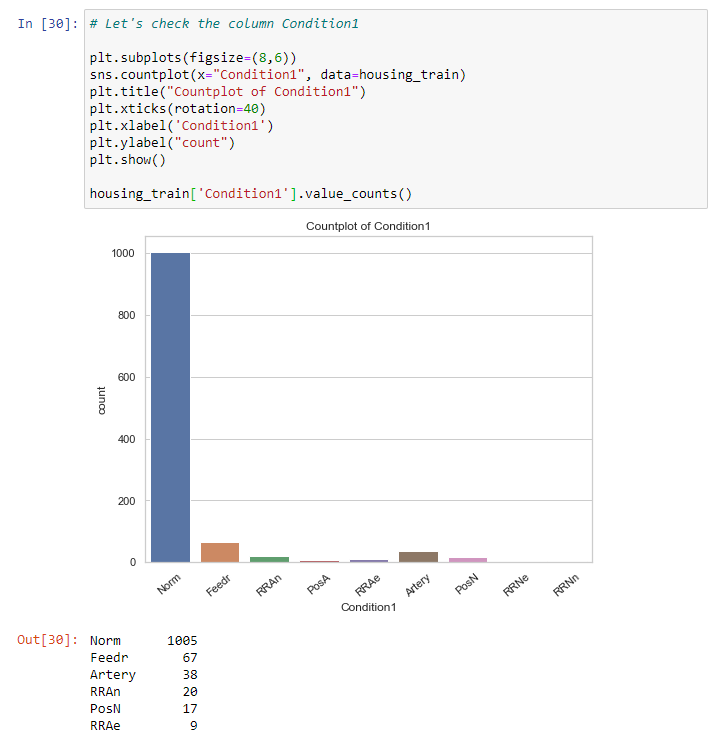


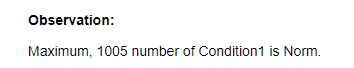




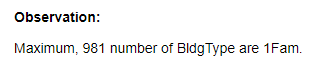


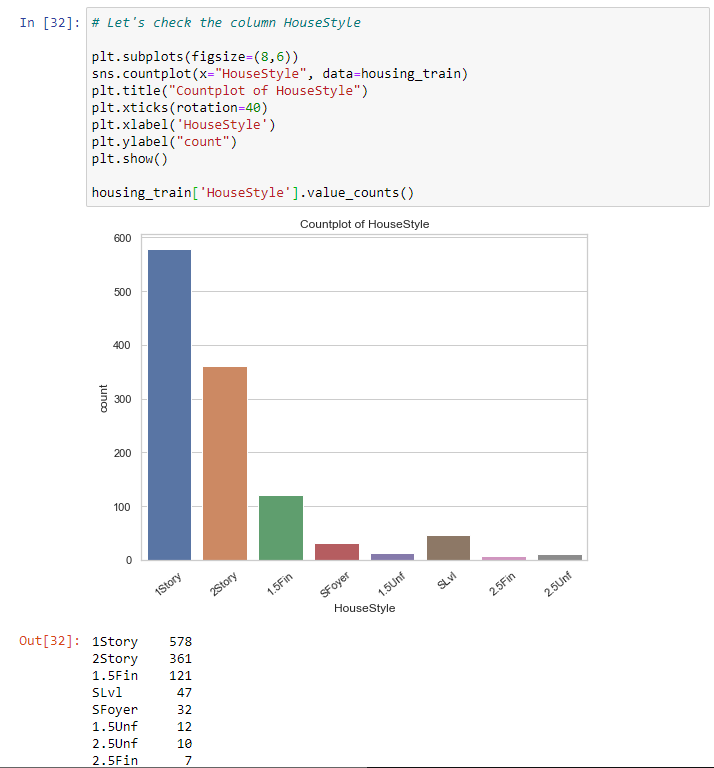


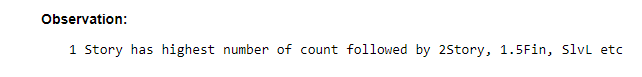


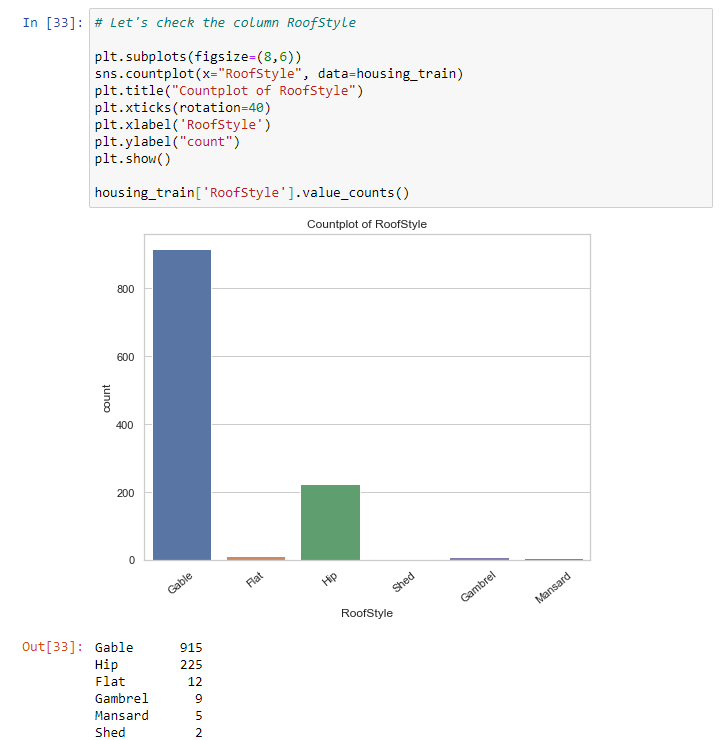


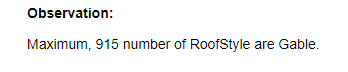


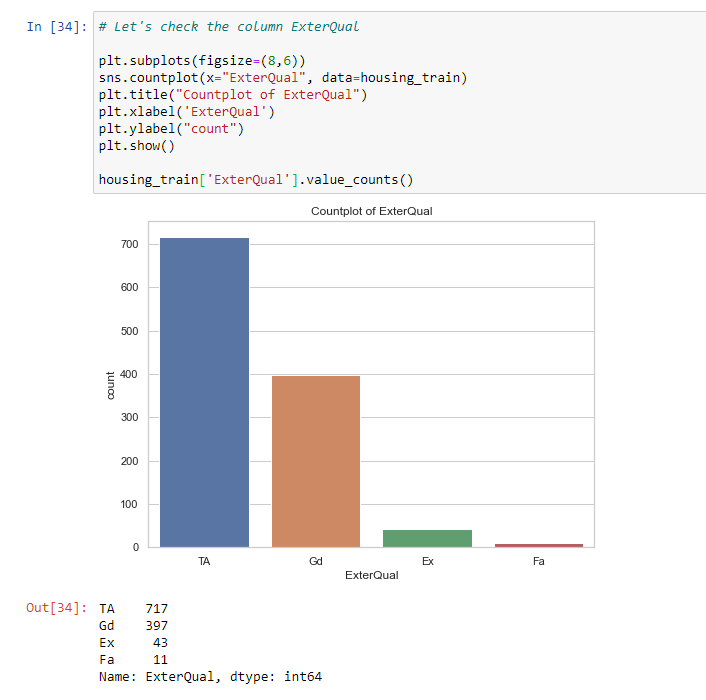


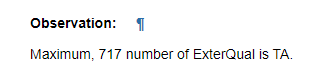


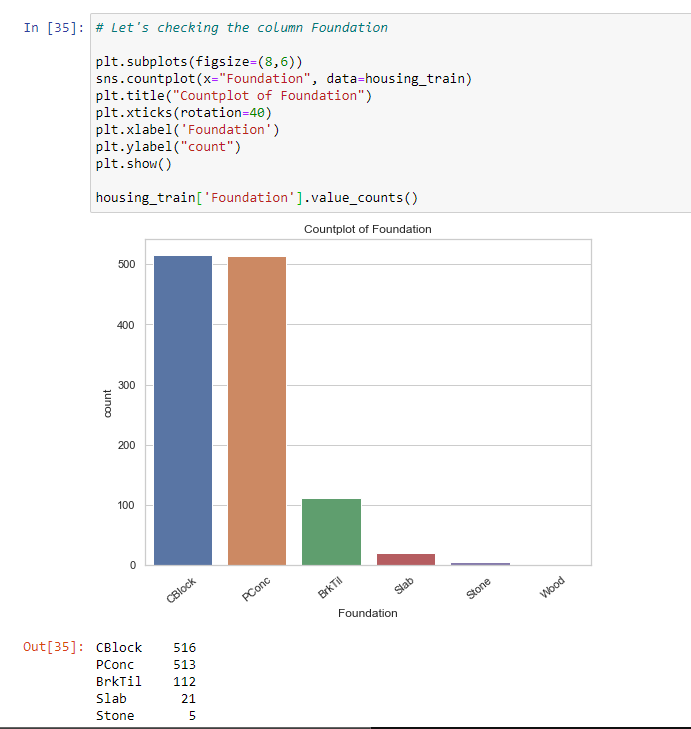


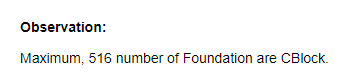




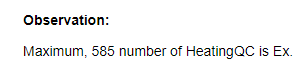


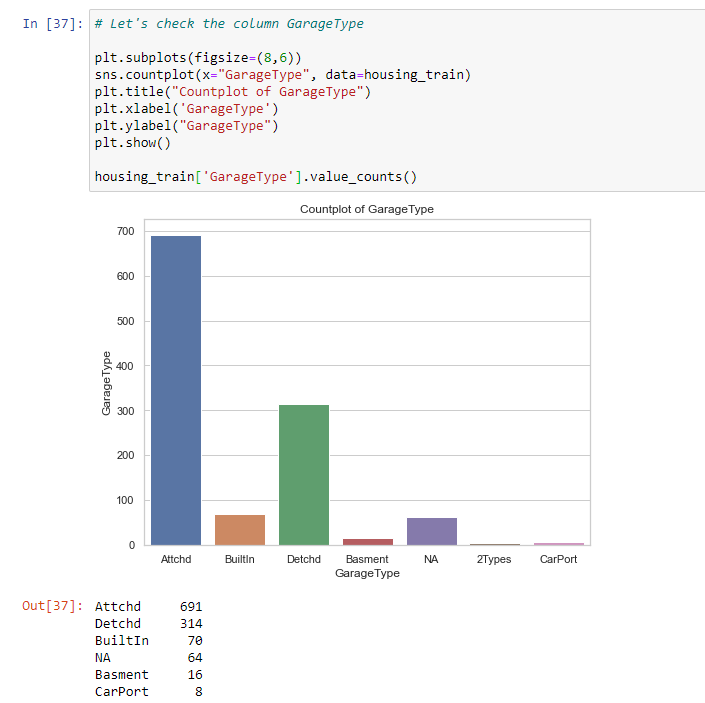


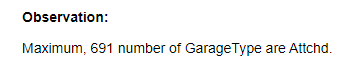


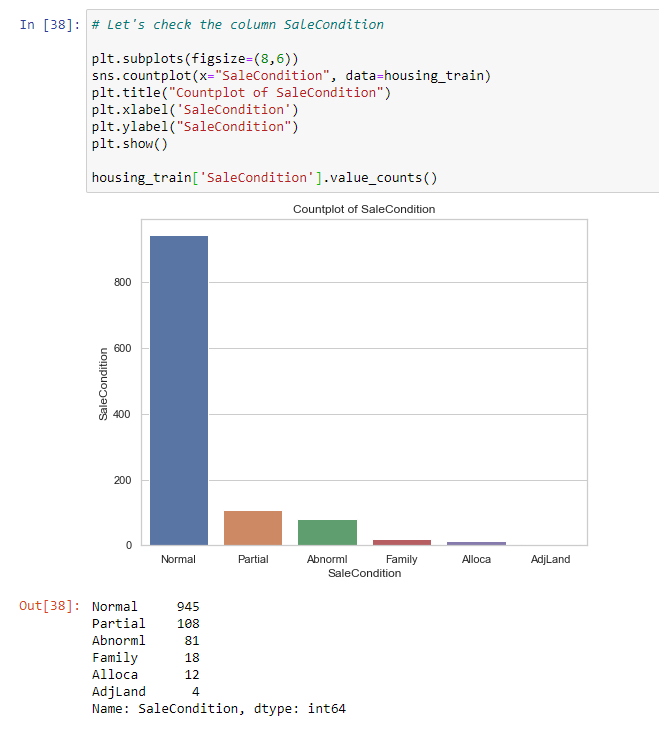


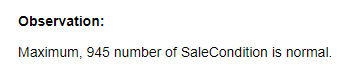




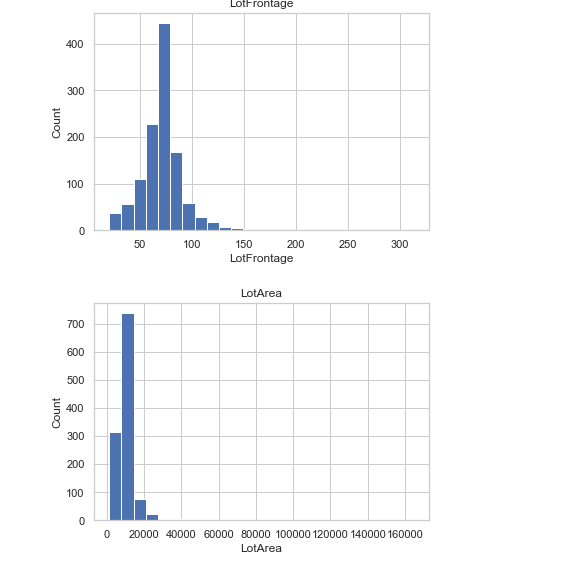


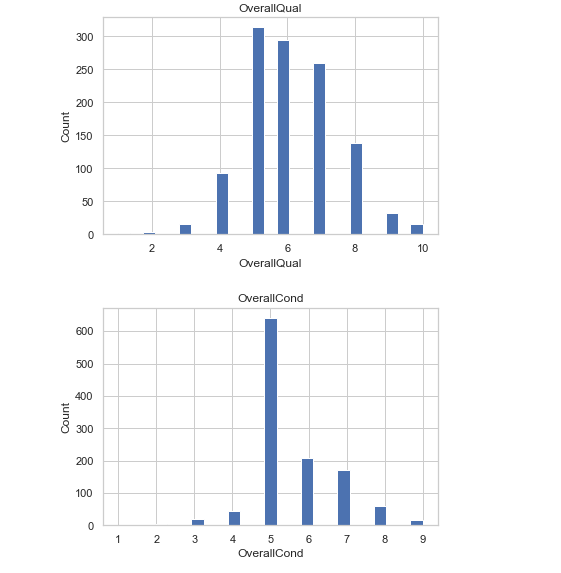


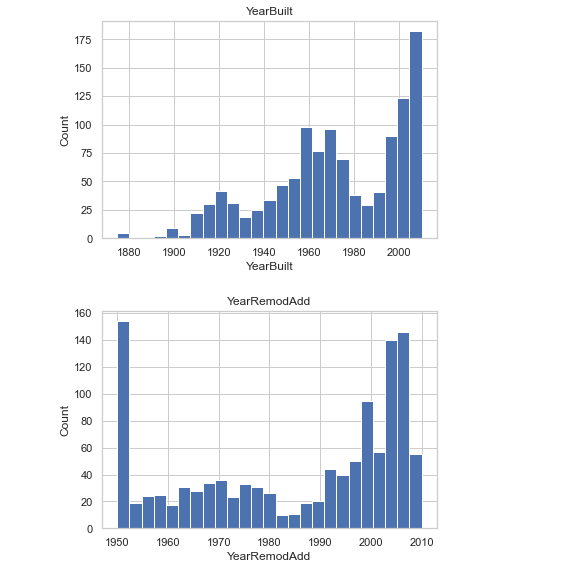


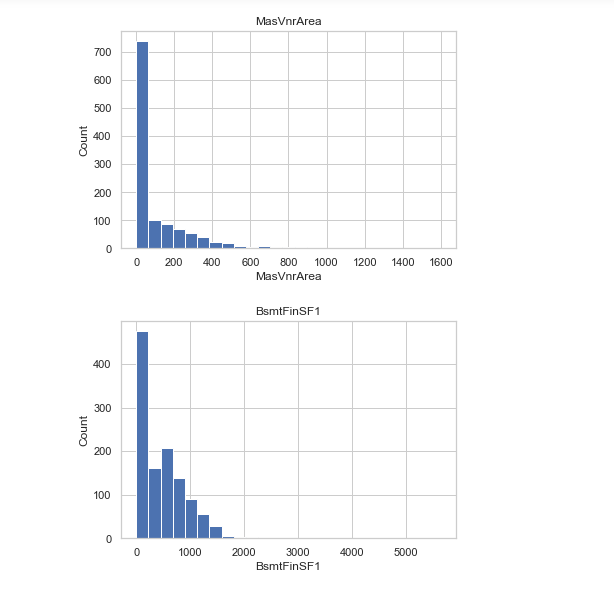


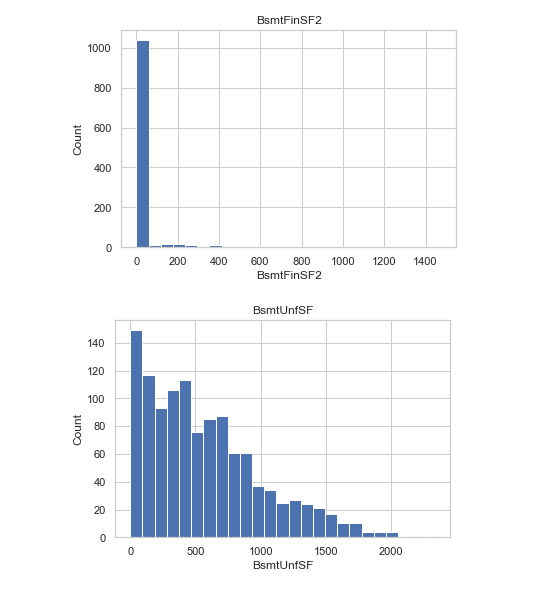
Chart, bar chart

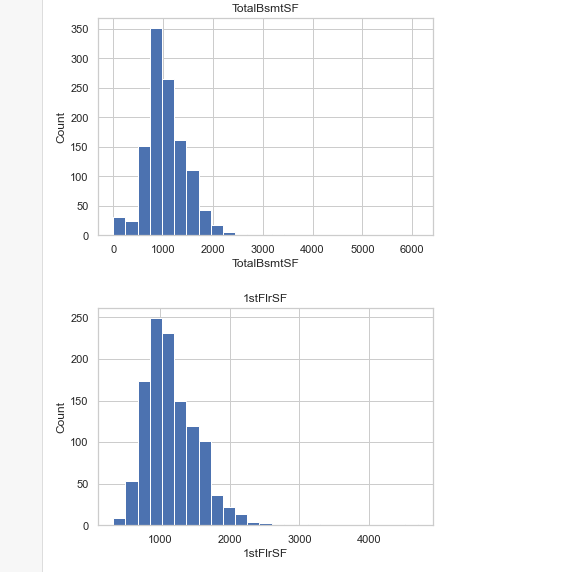
Description automatically generated

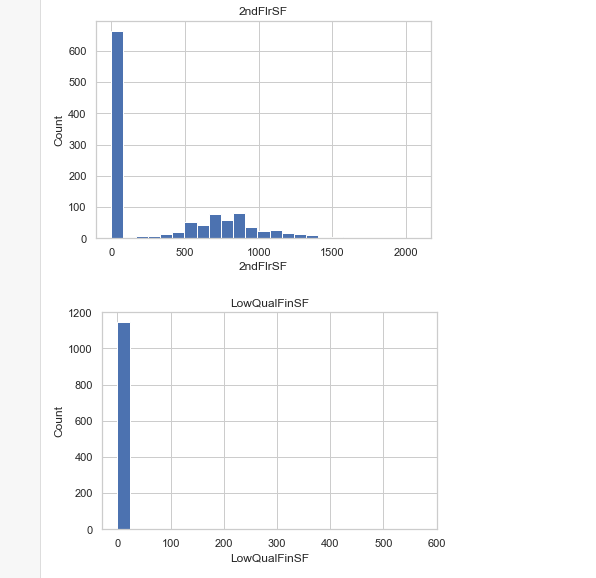


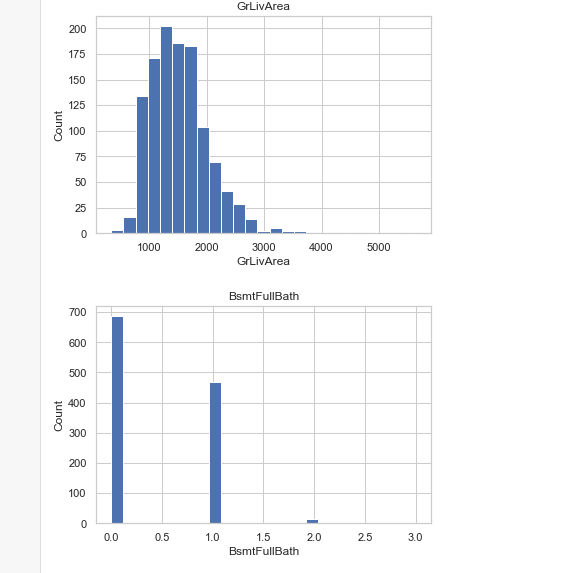


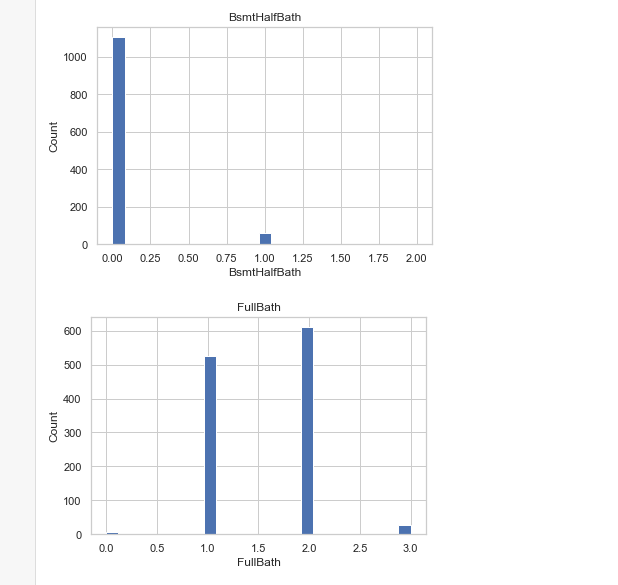


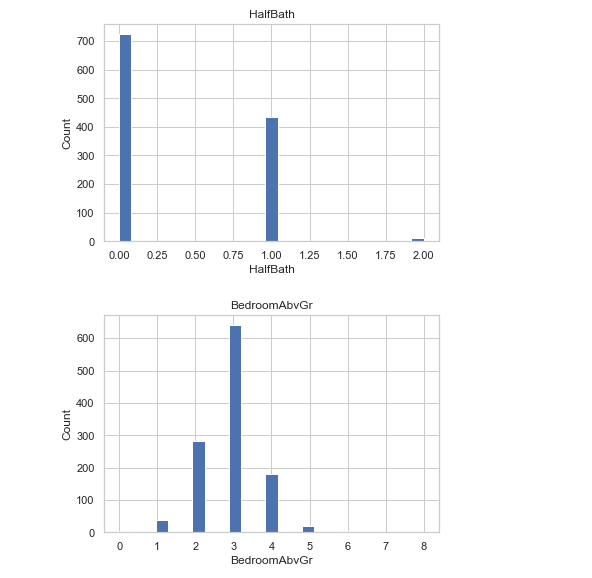


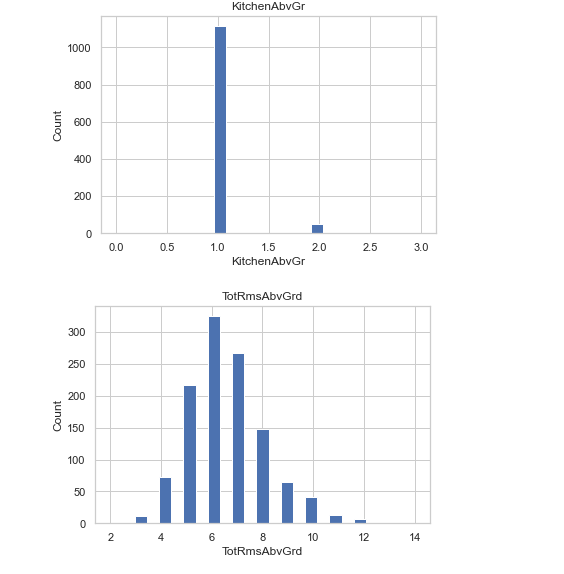


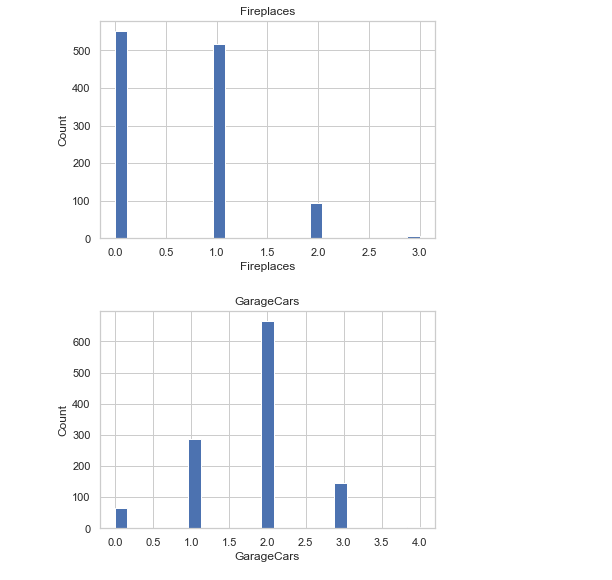


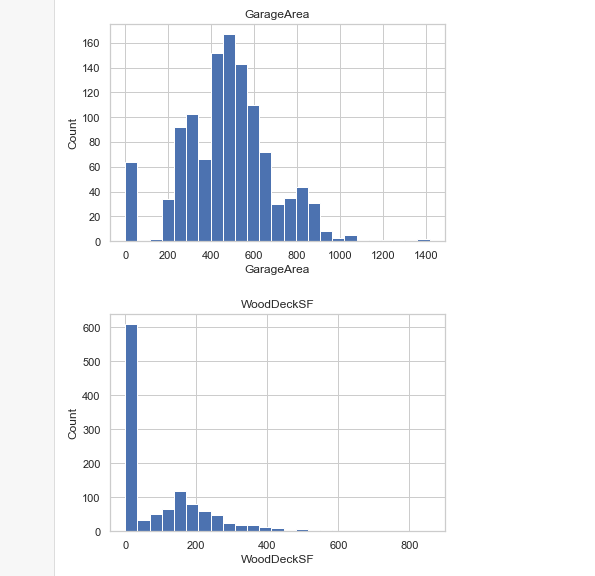


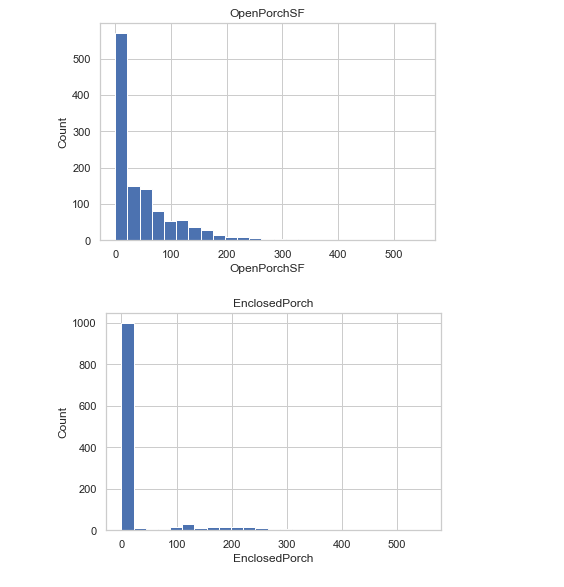


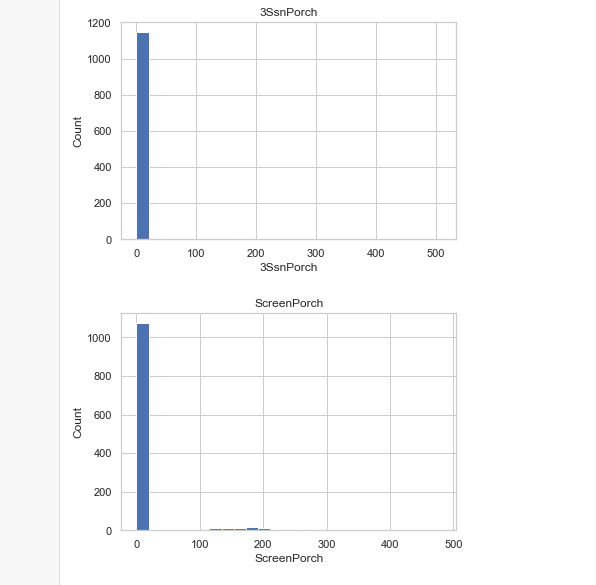


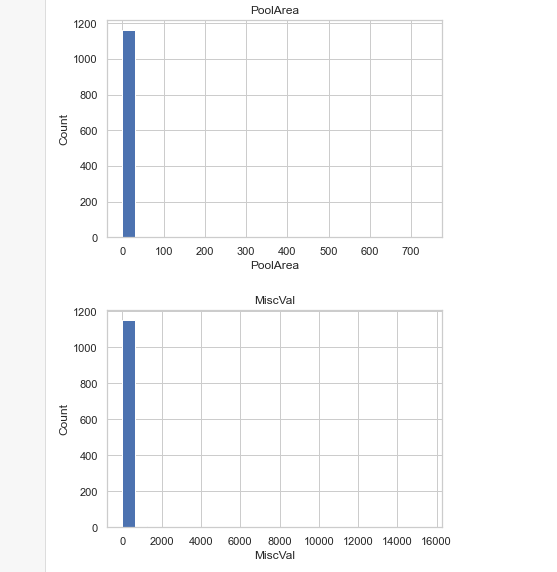


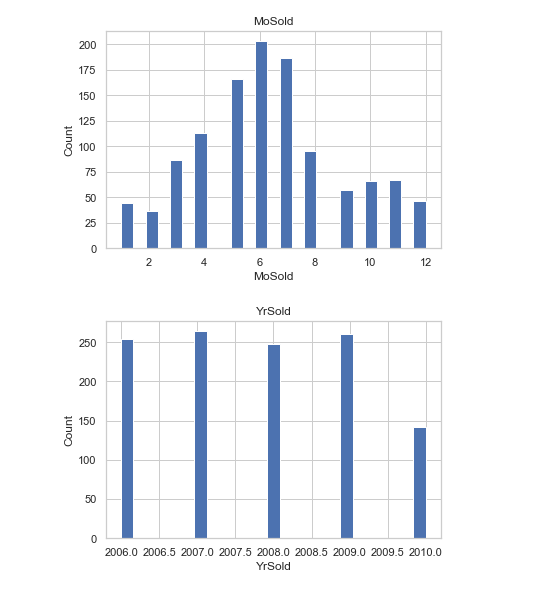


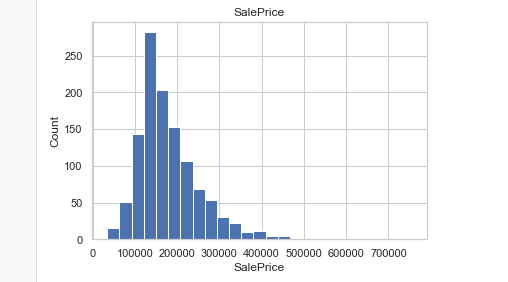






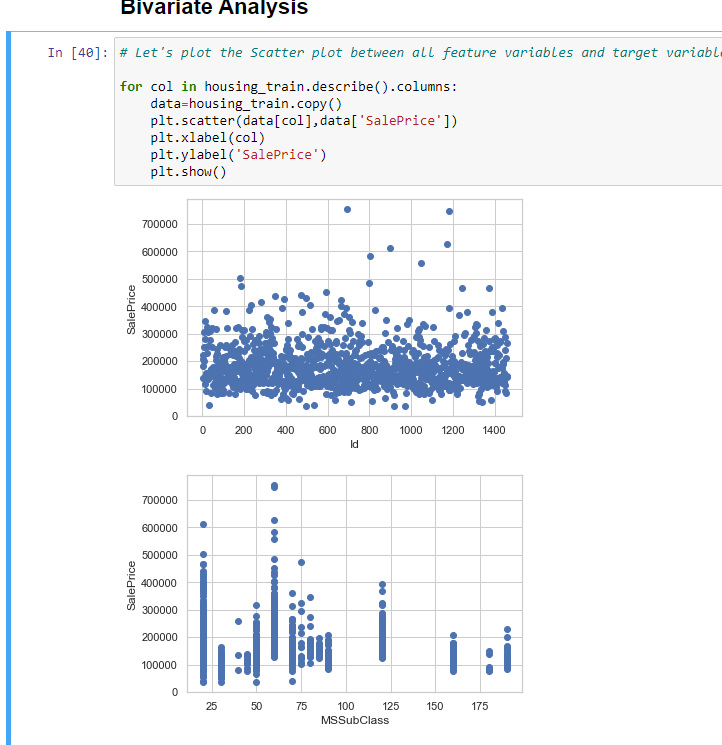


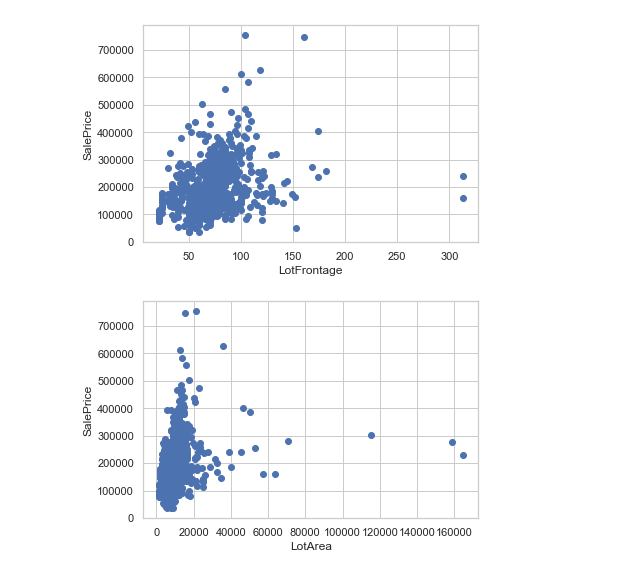


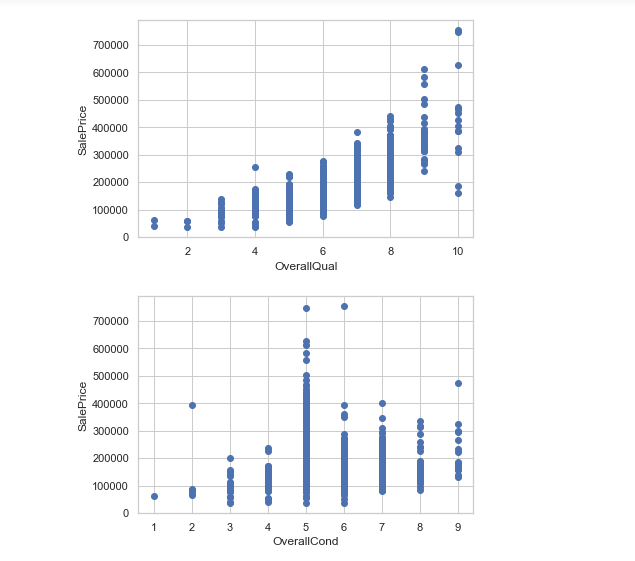


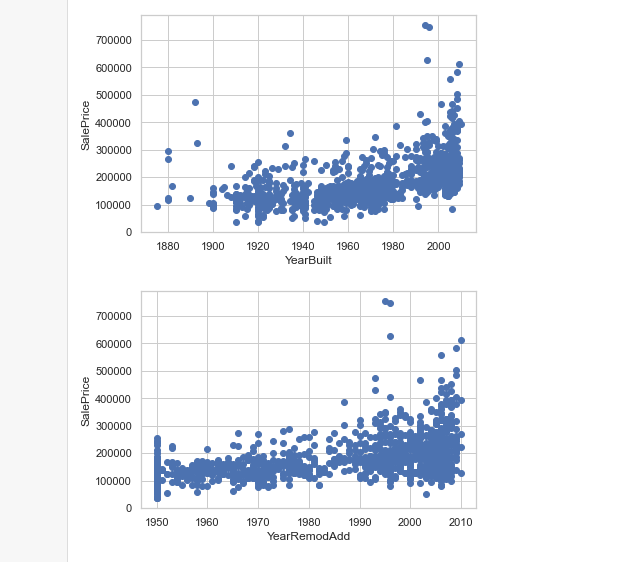
Graphical user interface, application

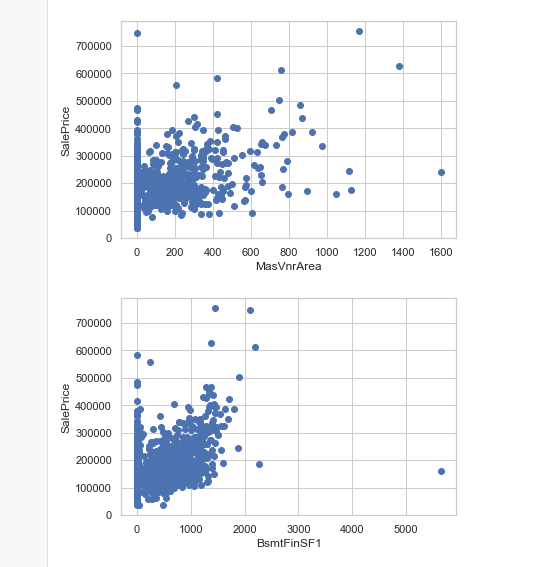
Description automatically generated

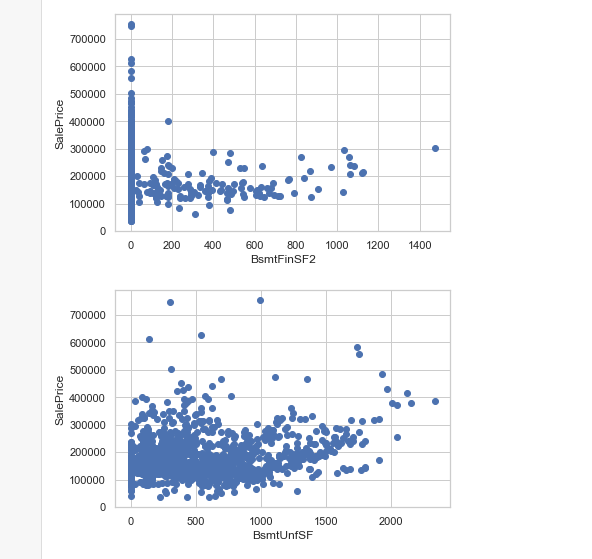


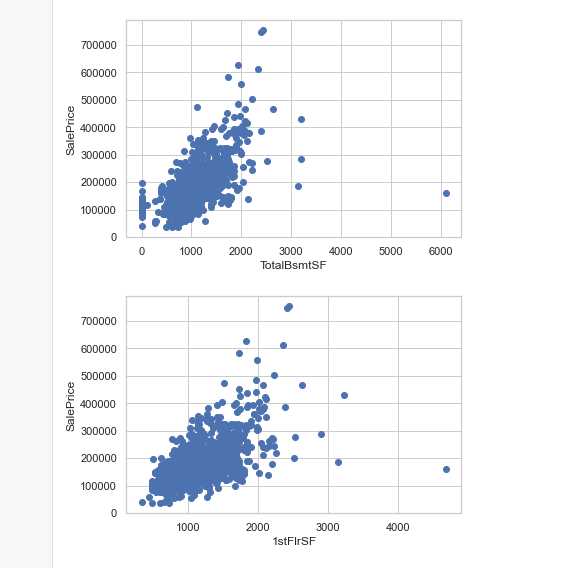


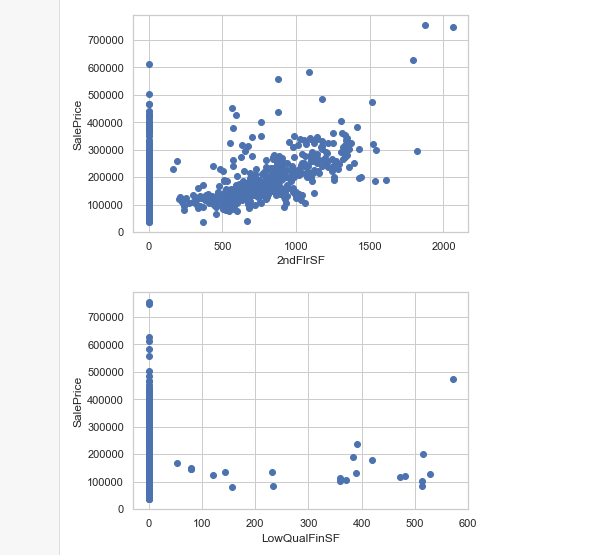


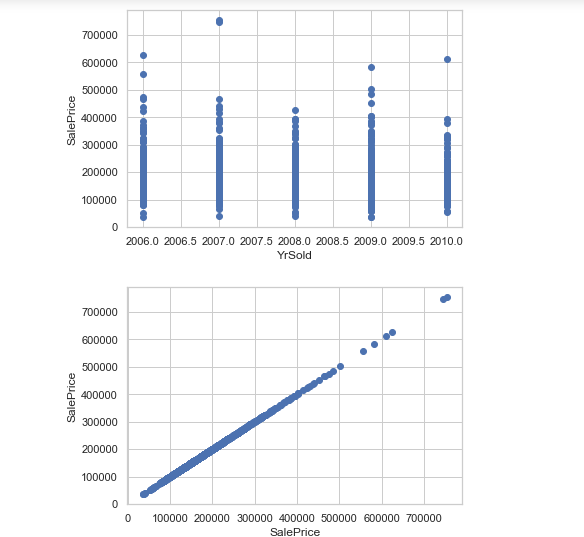




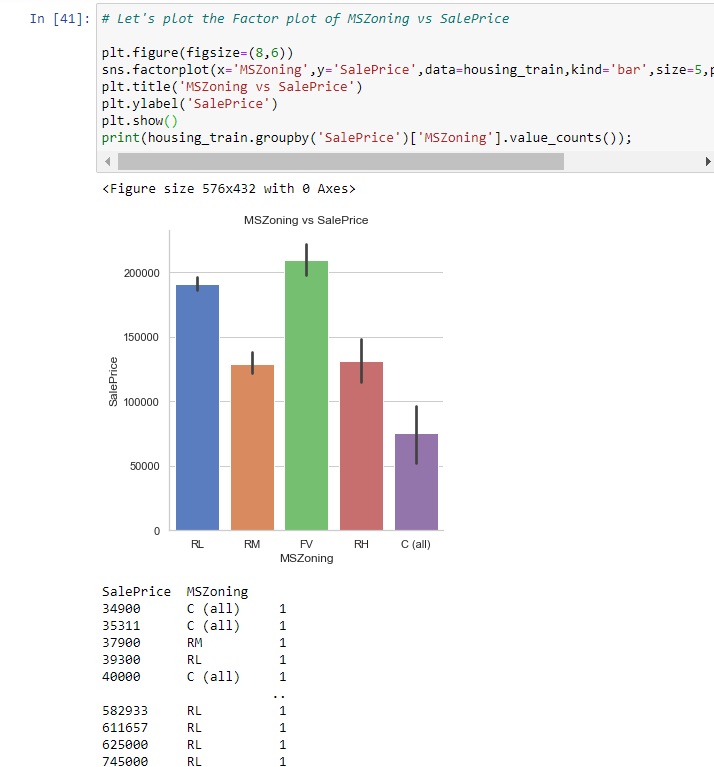


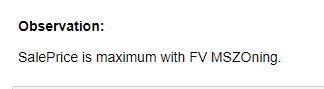




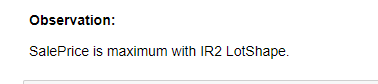


**Factor Plot**



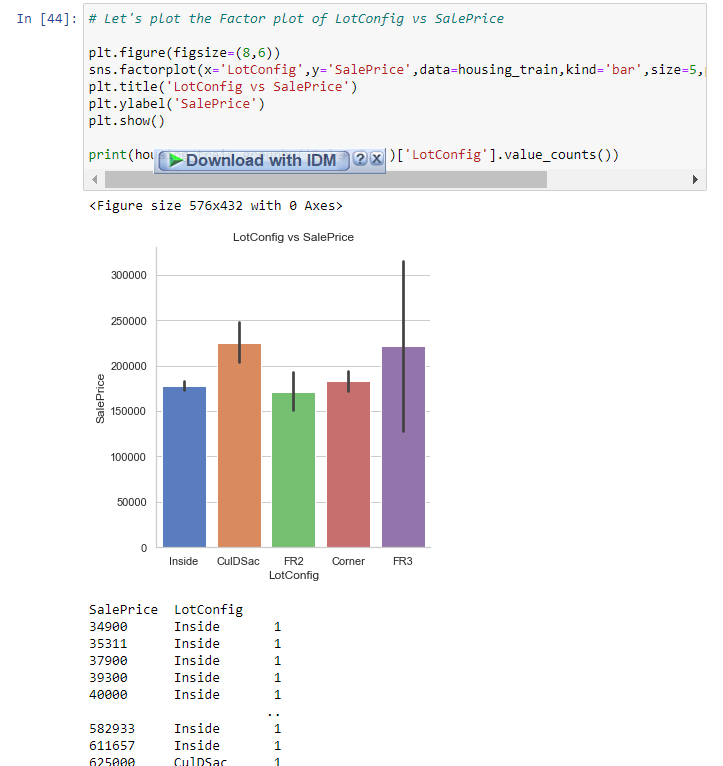




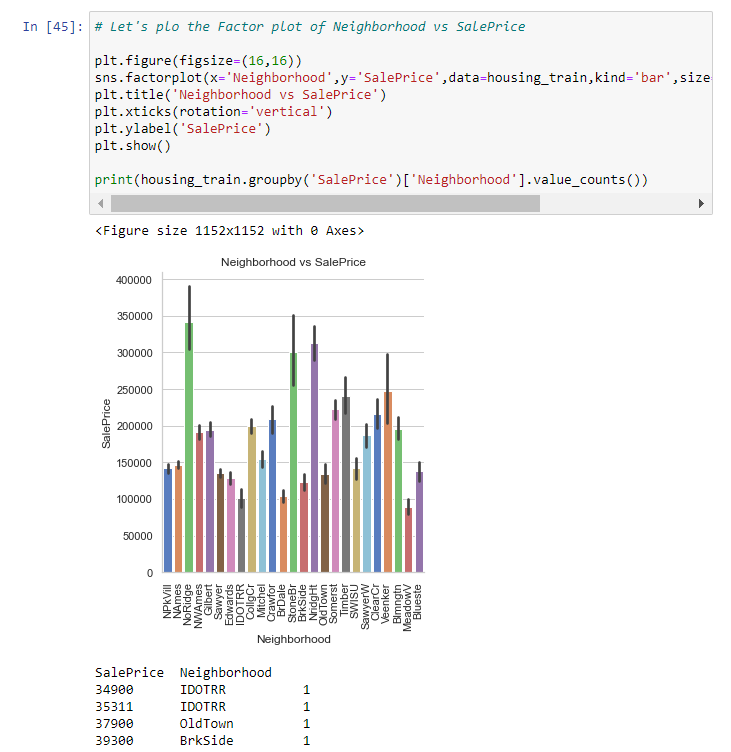


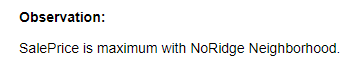


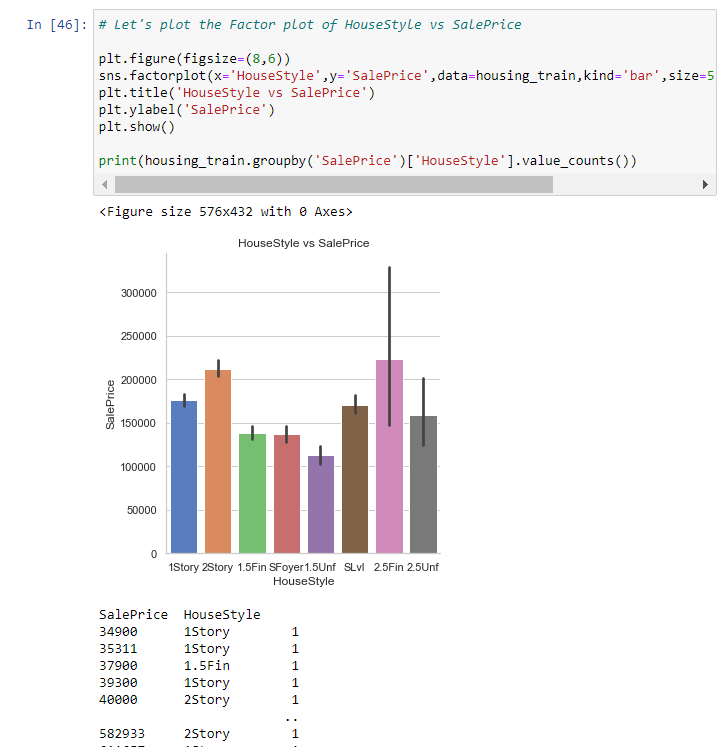


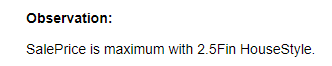


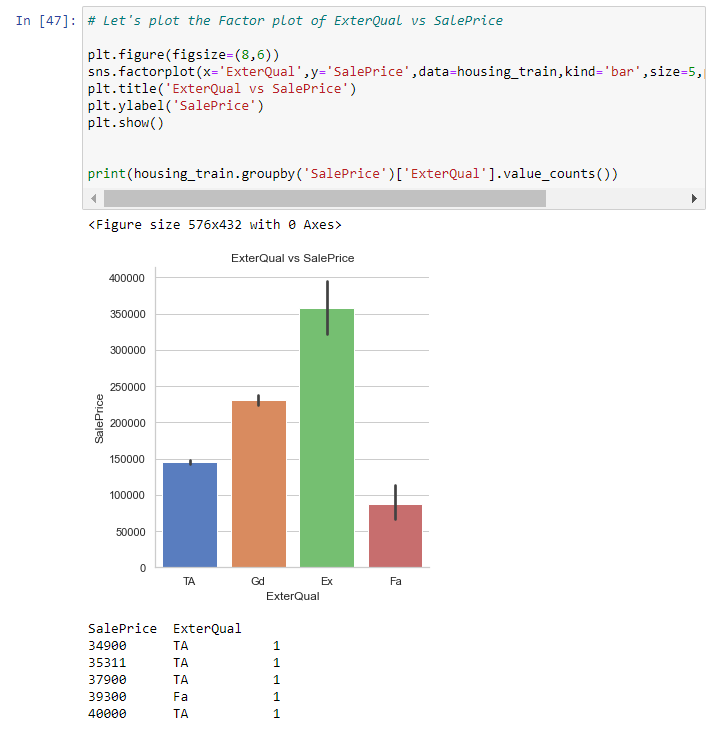


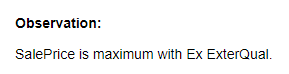


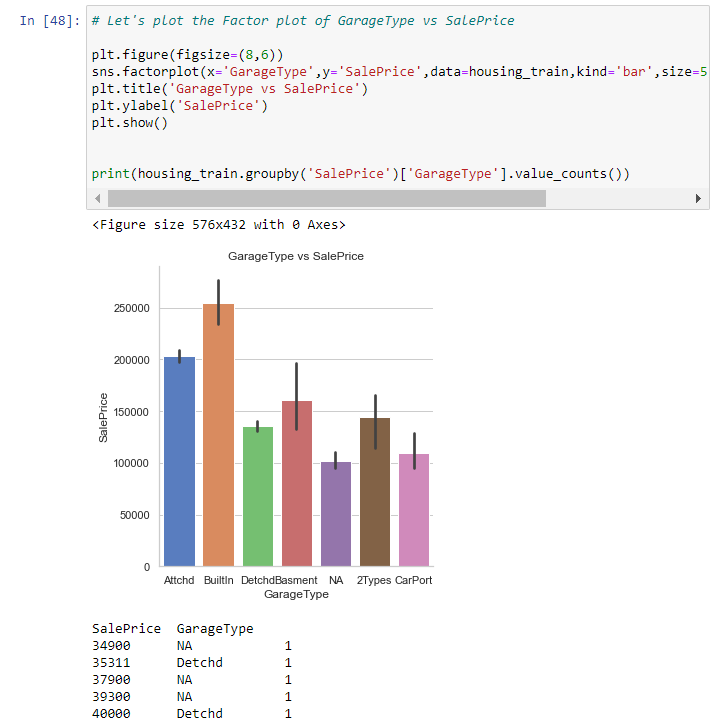


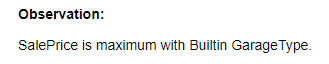


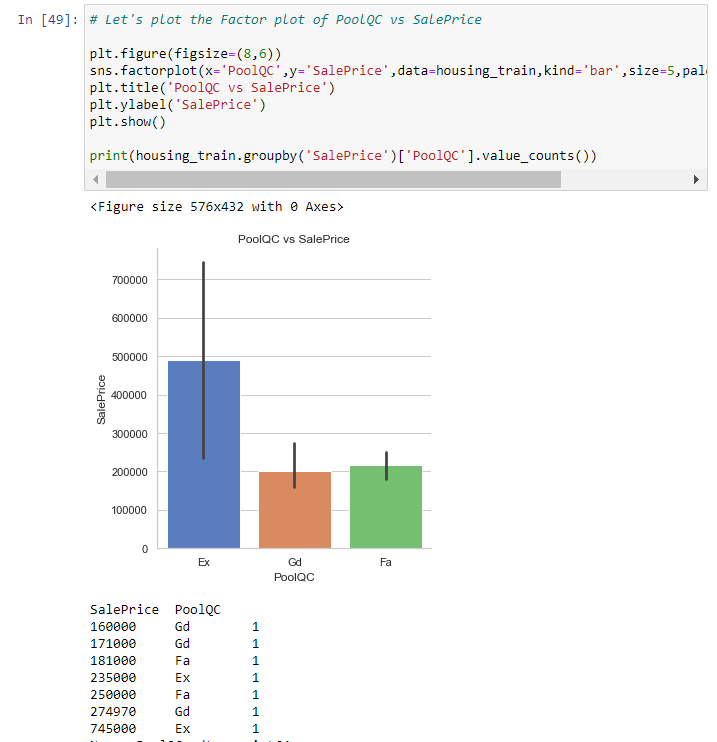


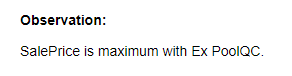


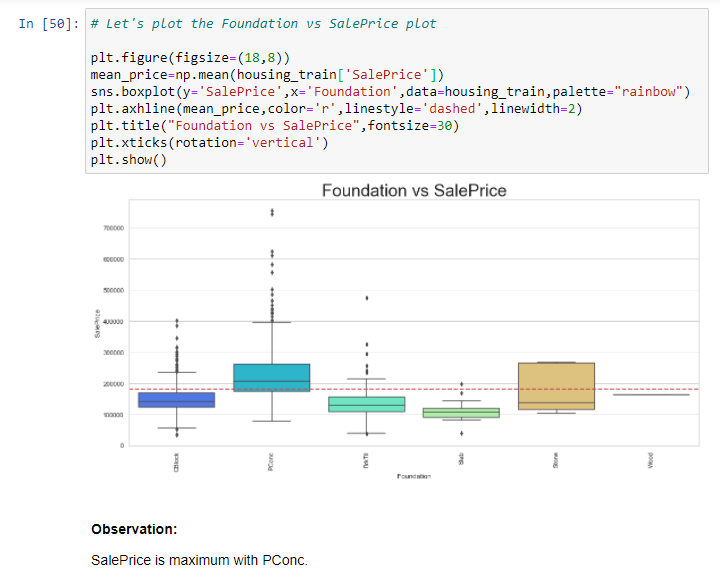


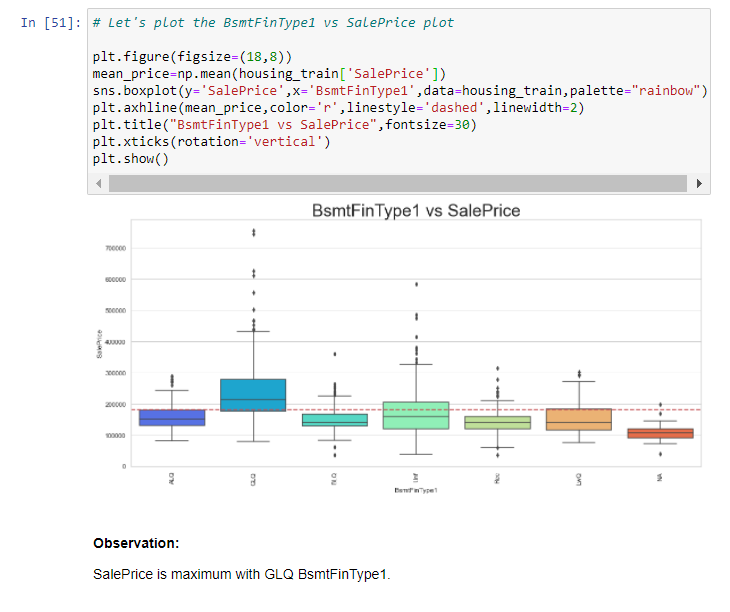












**Graphical user interface

Description automatically generated with low confidenceObservation**:

With MSZoning RL and increase in OverallQual the SalePrice of a house increase.

Graphical user interface

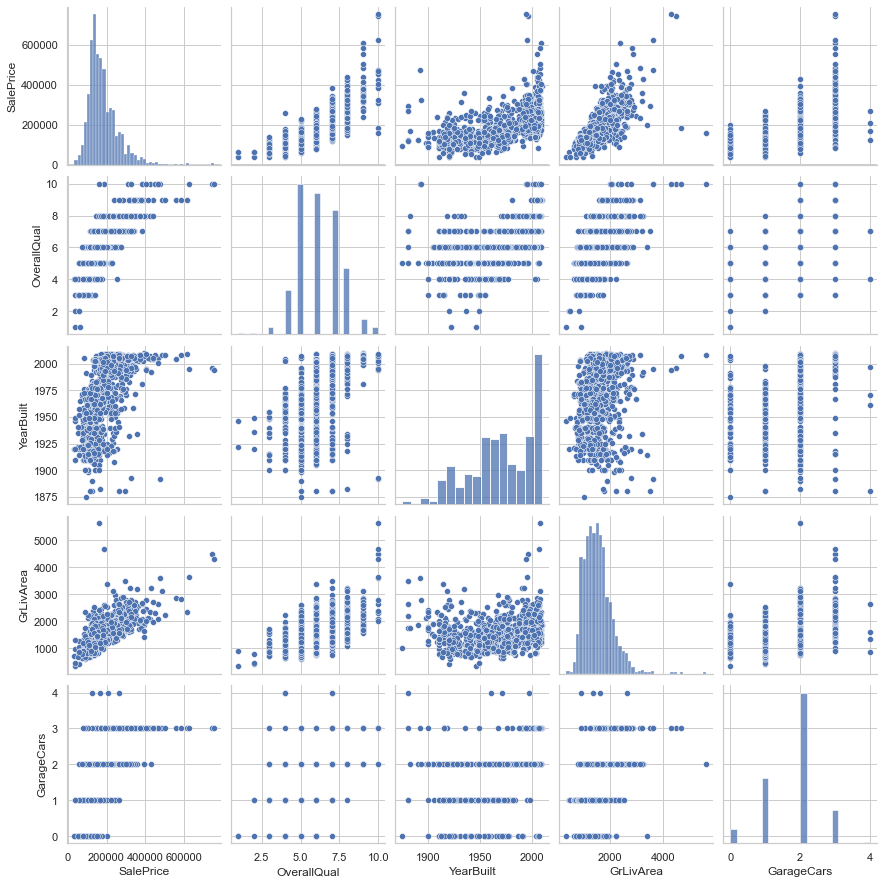
Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

**# Let's plot the pairplot**

**sns.pairplot(df, vars=['SalePrice','OverallQual','YearBuilt','GrLivArea','GarageCars']);**



**Observation:**

SalePrice is highly positively correlated with GrLivArea and OverallQual.

## INTERPRETATION OF THE RESULTS

From the visualization we interpreted that the target variable SalePrice was highly positively correlated with the columns GrLivArea, YearBuilt, OverallQual, GarageCars, GarageArea.

From the pre-processing we interpreted that data was improper scaled.

Graphical user interface, text, application, email

Description automatically generated

From the modeling we interpreted that after hyperparameter tuning Gradient Boosting Regressor works best with respect to our model with minimum RMSE of 29516

# CONCLUSION

## KEY FINDINGS AND CONCLUSIONS OF THE STUDY

In this project we have tried to show how the house prices vary and what are the factors related to the changing of house prices. The best(minimum) RMSE score was achieved using the best parameters of Gradient Boosting Regressor through GridSearchCV .

## LEARNING OUTCOMES OF THE STUDY IN RESPECT OF DATA SCIENCE

This project has demonstrated the importance of sampling effectively, modelling and predicting data.

Through different powerful tools of visualization we were able to analyse and interpret different hidden insights about the data.

Through data cleaning we were able to remove unnecessary columns and outliers from our dataset due to which our model would have suffered from overfitting or underfitting.

The data was improper scaled, so we scaled it to a single scale using sklearns’s package StandardScaler.

There were too many features present in the data, so we applied Principal Component Analysis(PCA) and found out the Eigenvalues and on the basis of number of nodes we were able to reduce our features up to 90 columns.

There were lot of missing values present in different columns which we imputed on the basis of our understanding.

The columns were skewed due to presence of outliers which we handled through winsorization technique.

## LIMITATIONS OF THIS WORK AND SCOPE FOR FUTURE WORK

While we couldn’t reach out goal of minimum RMSE in house price prediction without letting the model to overfit, we did end up creating a system that can with enough time and data get very close to that goal. As with any project there is room for improvement here. The very nature of this project allows for multiple algorithms to be integrated together as modules and their results can be combined to increase the accuracy of the final result. This model can further be improved with the addition of more algorithms into it. However, the output of these algorithms needs to be in the same format as the others. Once that condition is satisfied, the modules are easy to add as done in the code. This provides a great degree of modularity and versatility to the project.