

**HOUSE PRICE PREDICTION**

**PROJECT – REPORT**

**SUBMITTED BY – RAHUL JANGRA**

**INTRODUCTION**

***BUSINESS PROBLEM:***

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 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. 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. With its great weather, cosmopolitan cities, diverse natural landscapes and relaxed lifestyle, it’s no wonder that Australia remains a top pick for expats. Living cost in Australia for one person: $2,835 per month. Average living expenses for a couple: $4,118 per month. Average monthly living expenses for a family of 4: $5,378. Australia currently has the 16th highest cost of living in the world, with the USA and UK well behind at 21st and 33rd place respectively. Sydney and Melbourne are popular choices for expats moving to Australia. House pricing in some of the top Australian cities:-

Sydney - median house price A$1,142,212

Adelaide- median house price A$542,947.

Hobbart (smaller city)- median house price A$530,570.

**DATA SOURCES AND THEIR FORMATS**

Data (Housing price dataset) was provided by flip robo technologies pvt ltd. , Which contains the various features and label (Sale price) mentioned below:

Data file format : CSV ( Comma separated values) .

We have provided 2 datasets here : Train dataset , Test dataset .

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

**Language and IDE used:**

Python 3 and Jupyter Notebook.

So, now we’ll start working on this data in jupyter notebook . and taken following steps:

**Importing the Required liabraries :**

****

**Liabraries and their use:**

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

**Numpy** we worked with arrays.

**Matplotlib and seaborn** we did plot various graphs and figures and done data visualization.

**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.

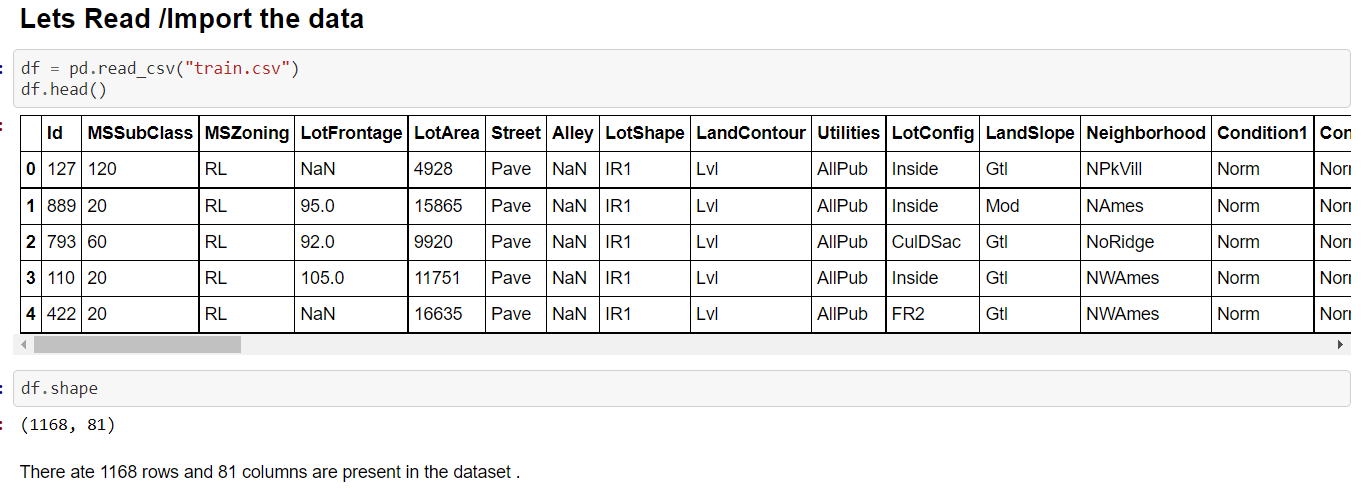
**from sklearn.model\_selection import train\_test\_split,cross\_val\_score** Train\_test\_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default,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

**scipy stats**

we treated outliers through winsorization technique.

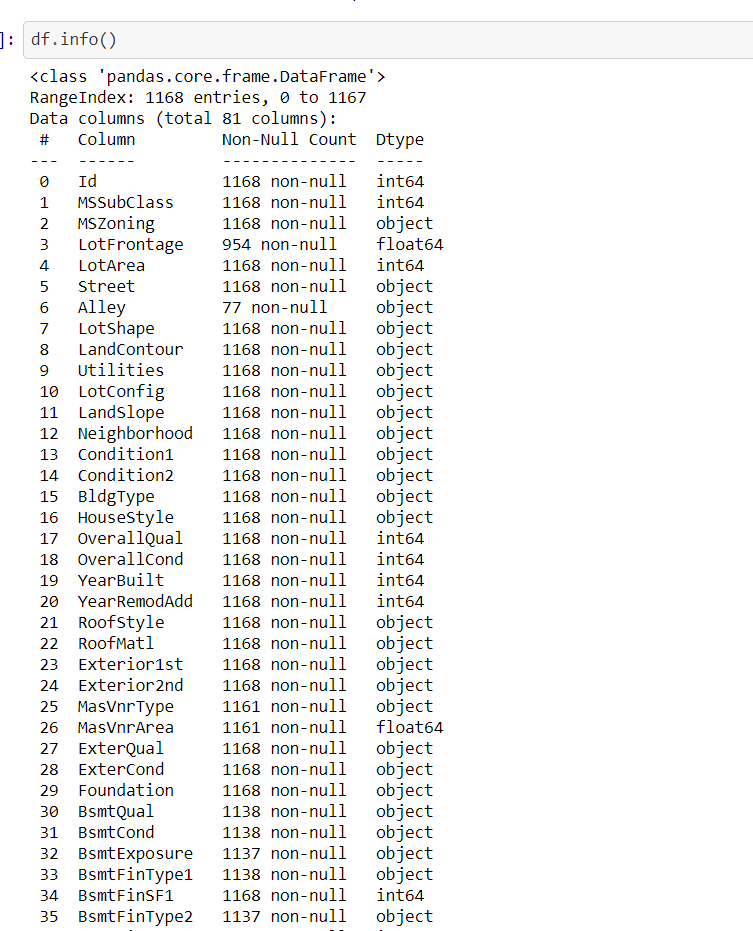
**sklearn’s standardscaler** package will–+ scaled all the feature variables onto single scale.

**Importing / Reading the Dataset:**

****

Imported the training dataset which contains 1168 rows and 81 columns.

**Let’s Explore the data:**

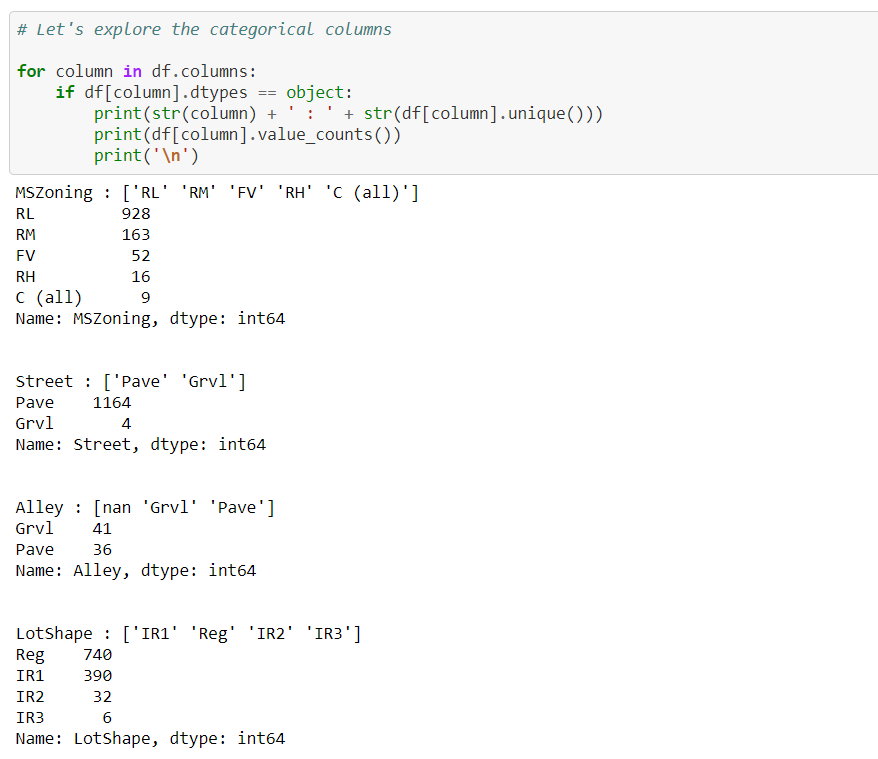
****

So, we have categorical and numerical type data in the dataset.

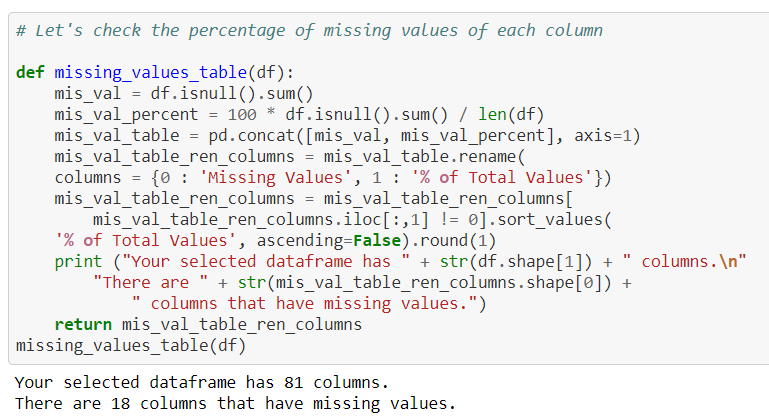
**Columns available in the dataset:**

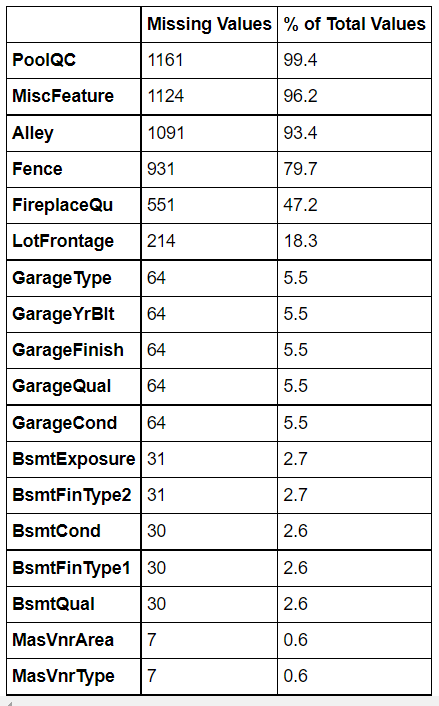


Through the below mentioned code we can identify the unique values available in each Categorical columns:



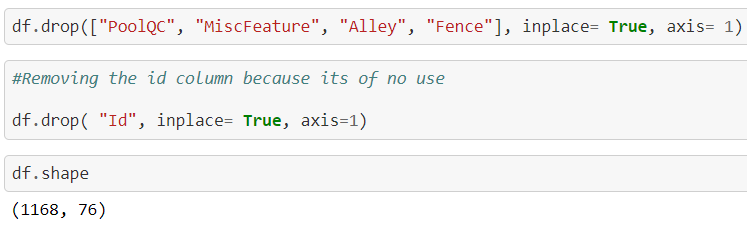
**Now , we’ll Check the null values in the dataset :**





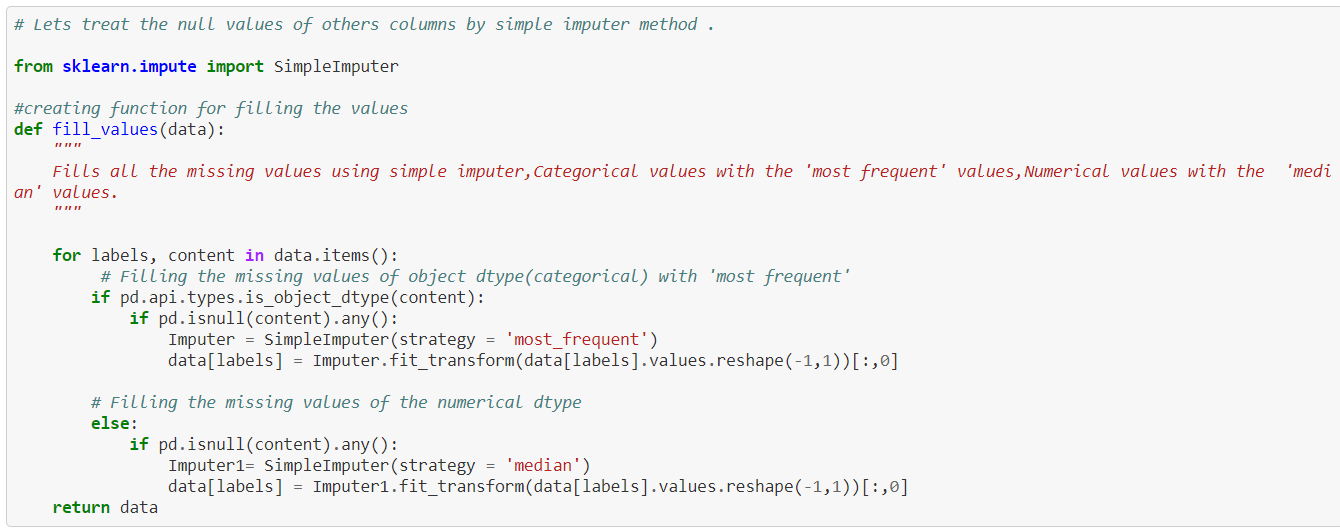
**Observation :** Out of 81 columns there are 18 columns having null values , So we can remove the columns having more than and equal to approx 80% of null values , and treat the rest null values.

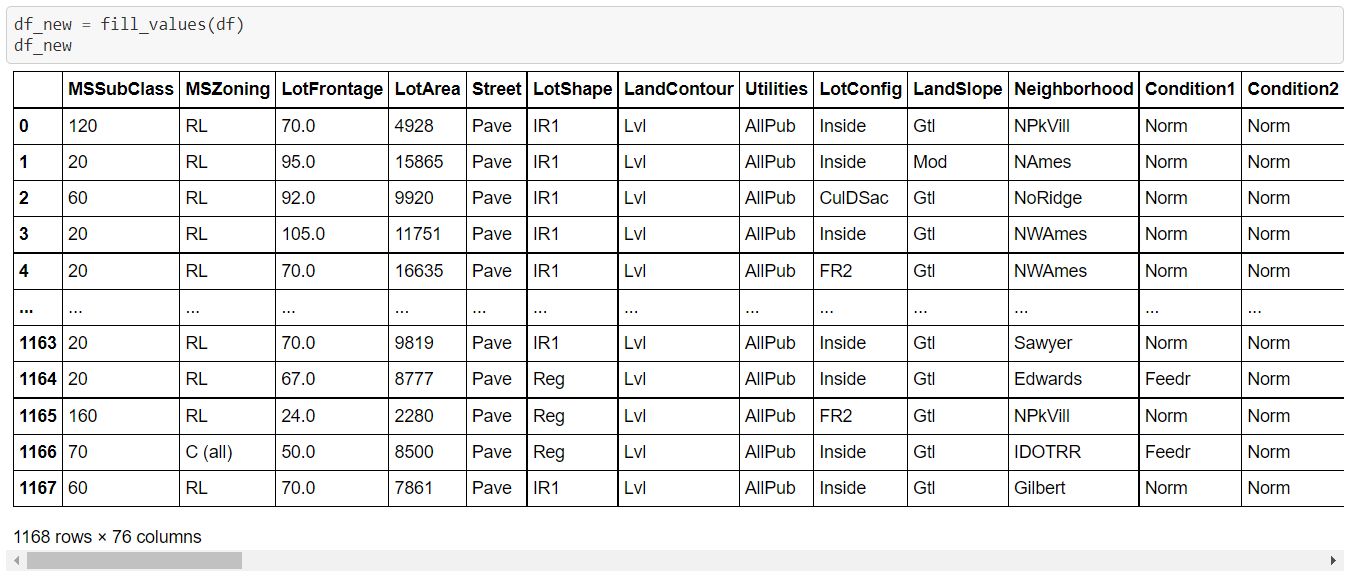
Dropping the columns :

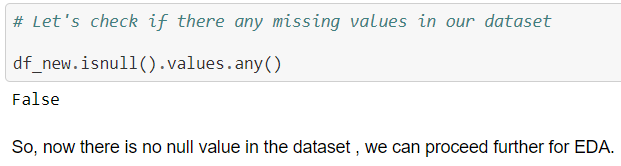


**Filling the null values using simple Imputer :**

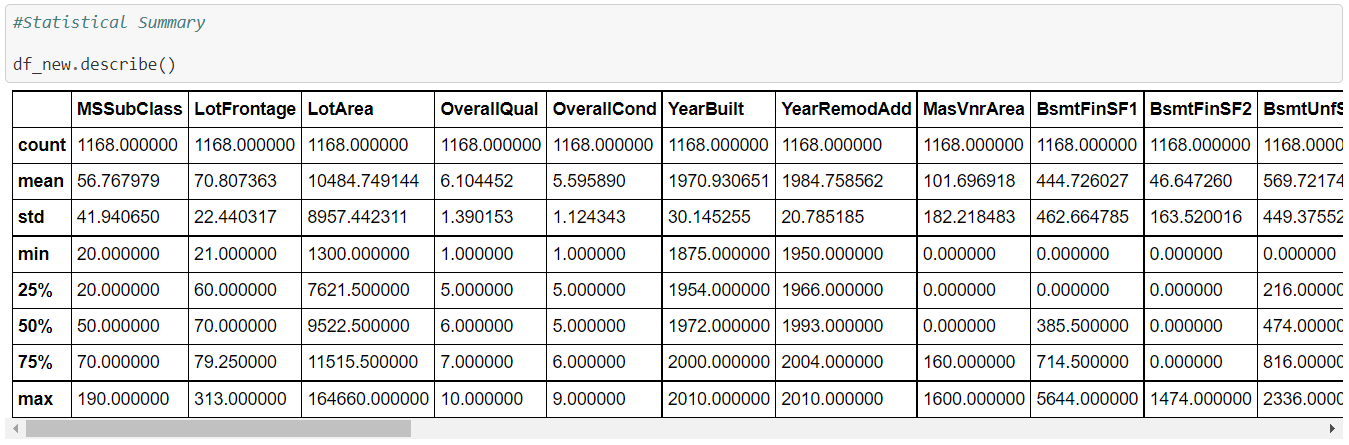
**Simple Imputer:** The SimpleImputer class provides basic strategies for imputing missing values. Missing values can be imputed with a provided constant value, or using the statistics (mean, median or most frequent) of each column in which the missing values are located.







**EDA/ Data Analysis:**



**Obseravtaions :**

Maximum standard deviation of 8957.44 is observed in LotArea column.

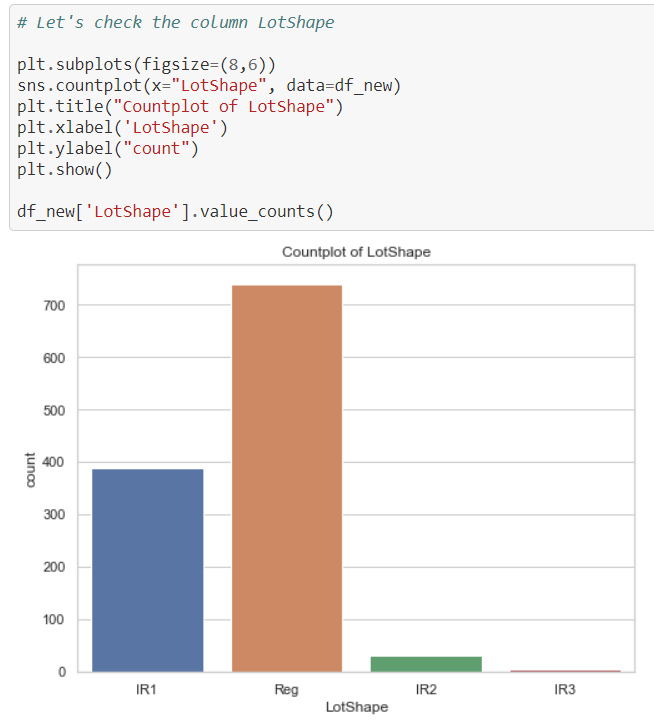
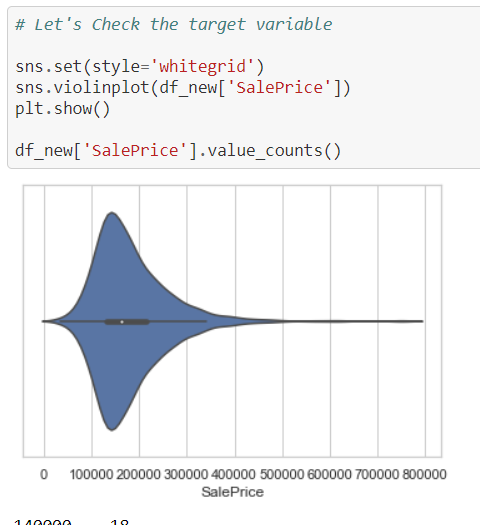
Maximum SalePrice of a house observed is 755000 and minimum is 34900.

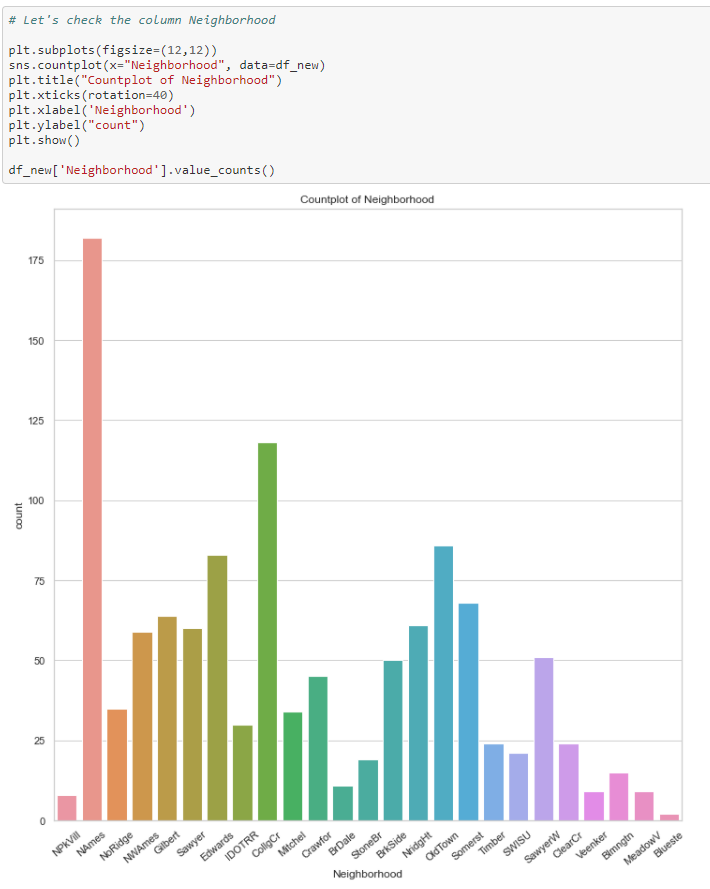
In the columns Id, MSSubclass, LotArea, MasVnrArea, BsmtFinSF1, BsmtFinSF2, BsmtUnfsF, TotalBsmtSF, 1stFlrSF, 2ndFlrSF, LowQualFinSF, GrLivArea, BsmtFullBath, HalfBath, TotRmsAbvGrd, WoodDeckSF, OpenPorchSF, EnclosedPorch, 3SsnPorch, ScreenPorch, PoolArea, Miscval, salePrice mean is considerably greater than median so the columns are positively skewed.

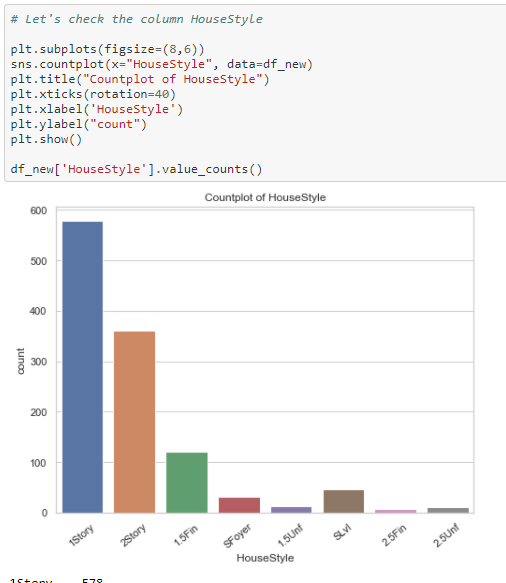
In the columns FullBath, BedroomAbvGr, Fireplaces, Garagecars, GarageArea, YrSold Median is greater than mean so the columns are negatively skewed.

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.

**Univirate Analysis :** When we analyse one variable .







Above are some Univirate Analysis Visualization .

**Observations :**

Maximum number of SalePrice lies between 140000 and 230000.

Maximum, 928 number of MSZoning are RL.

Maximum, 1164 number of Street are Pave where as only 4 are Grvl.

Maximum, 1046 number of LandContour are Lvl.

Maximum, 182 number of Neighborhood are Names.

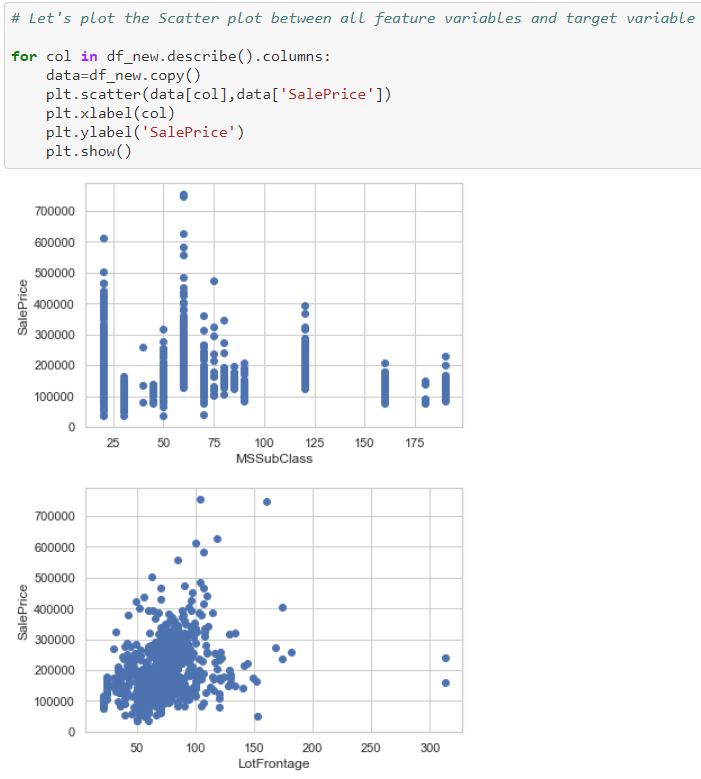
Maximum, 1005 number of Condition1 is Norm.

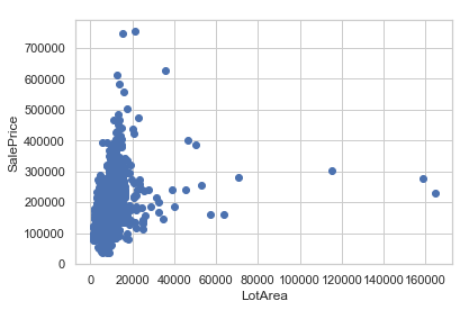
Maximum, 981 number of BldgType are 1Fam.

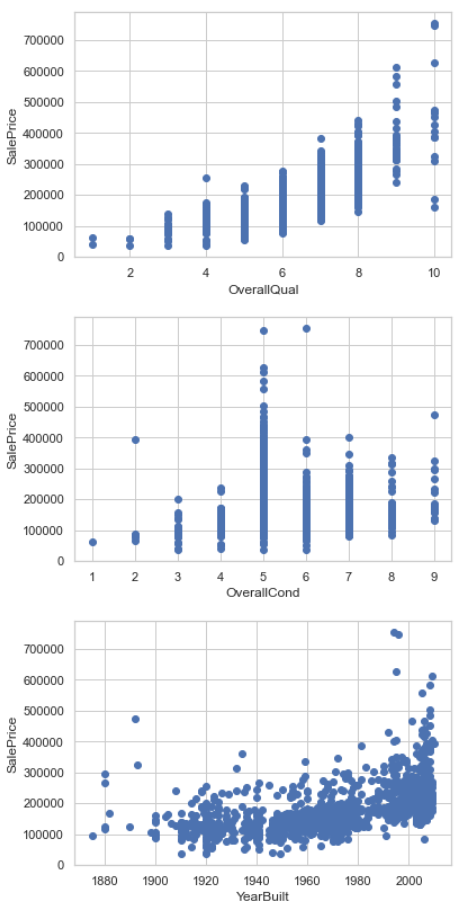
1 Story has highest number of count followed by 2Story, 1.5Fin, SlvL etc

Maximum, 516 number of Foundation are CBlock.

**Bivariate Analysis :** When we Analyse two variables at one time , with each other .



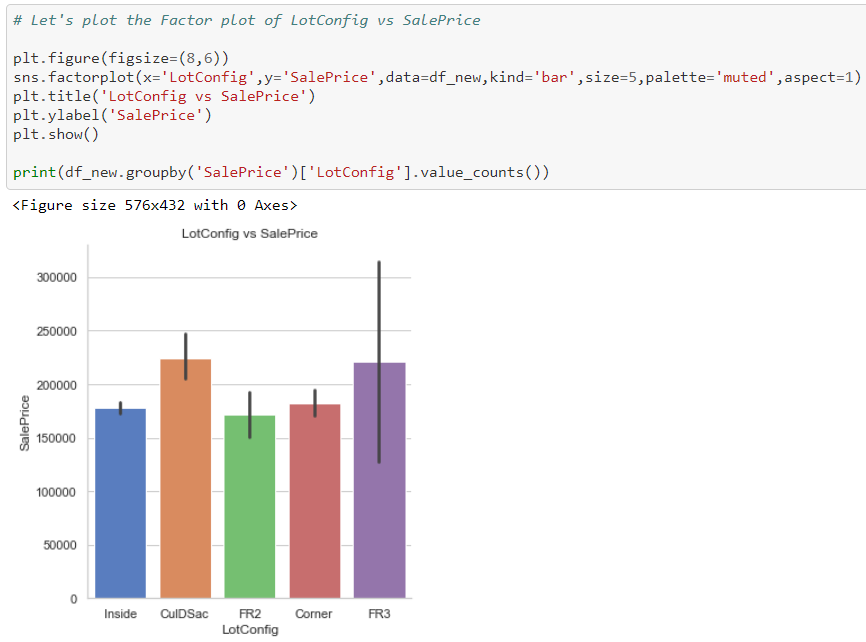


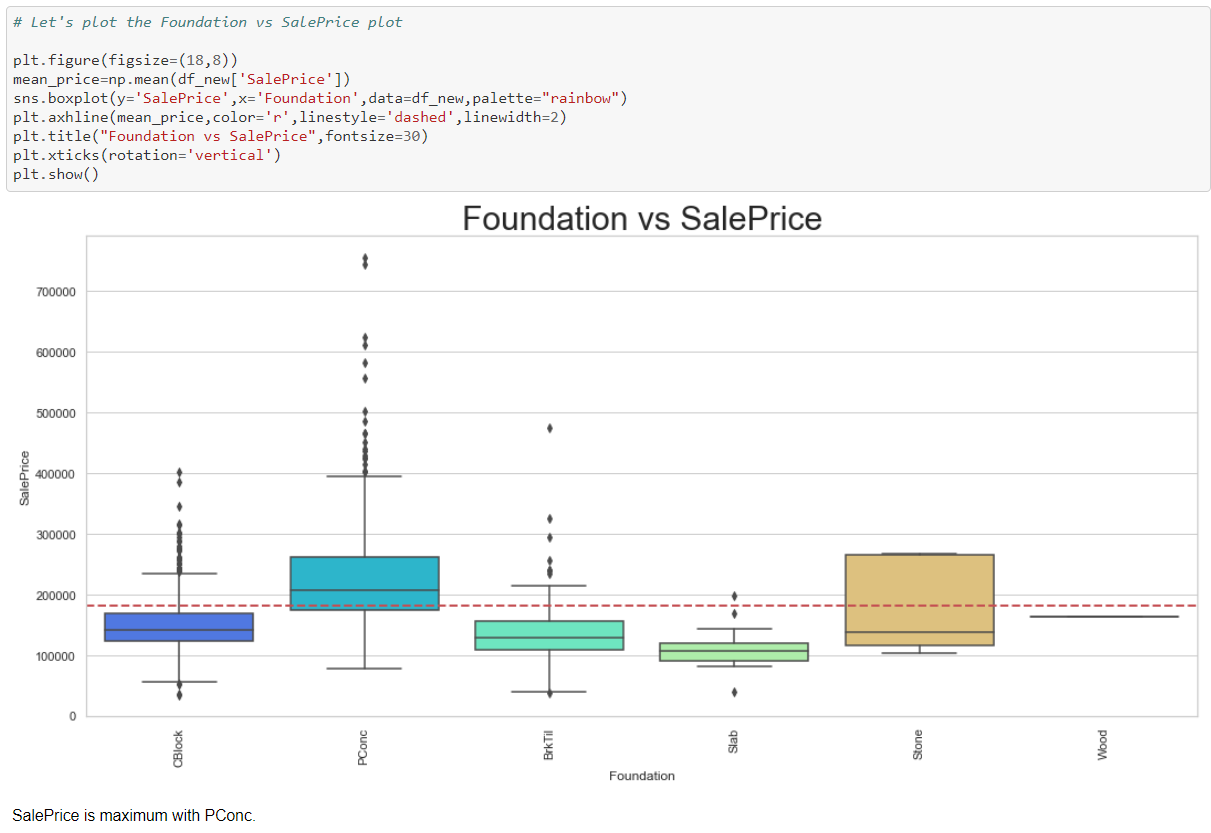




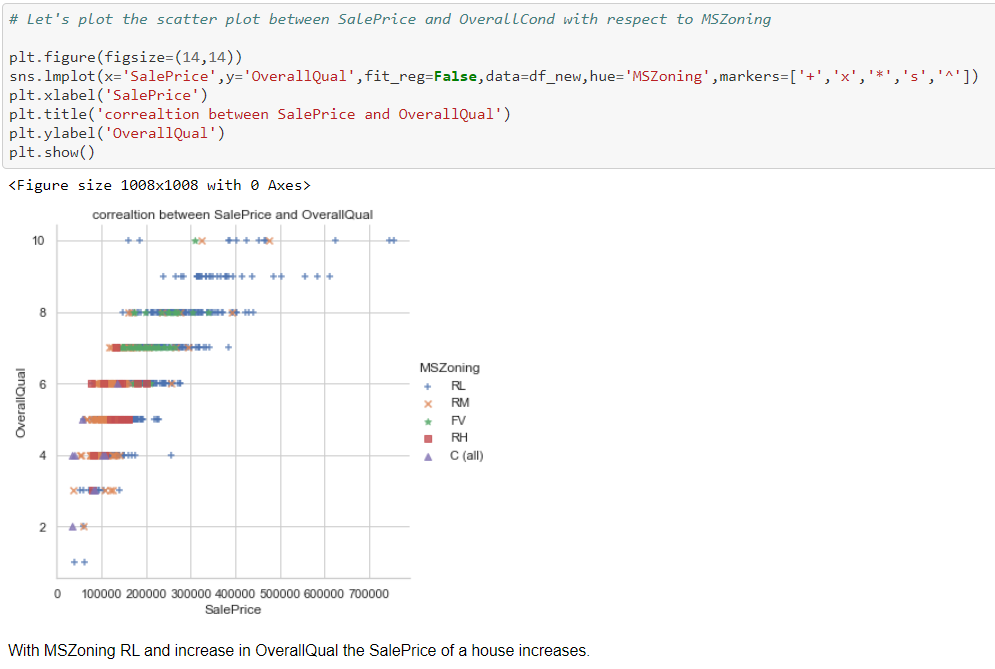


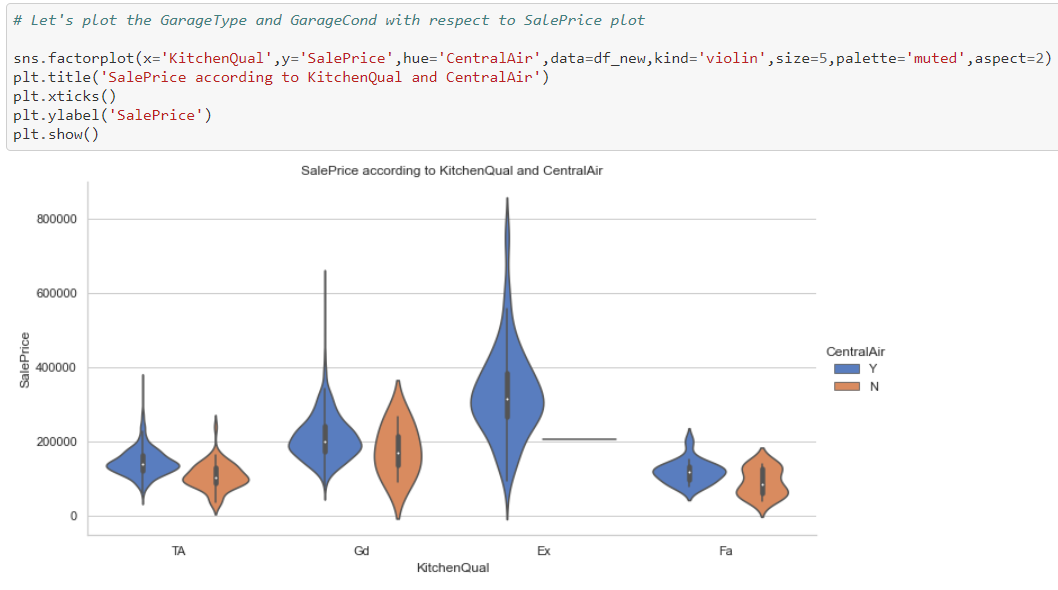






**Multivirate Analysis :** When we analyse more than 2 variables on single graph.



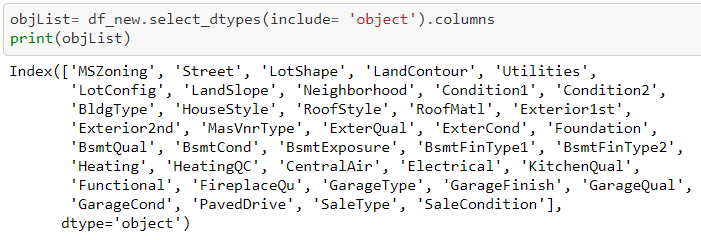




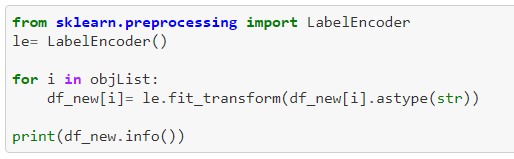
SalePrice is highly positively correlated with GrLivArea and OverallQual.

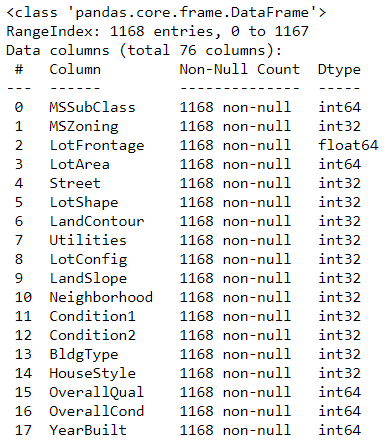
**Data PreProcessing :**

**Label Encoding :** Label Encoding helps us **to converting the labels into a numeric form so as to convert them into the machine-readable form**. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.



From the above code , we are detecting the object data type columns , and then will pass them in lablel encoding fit as mentioned below :



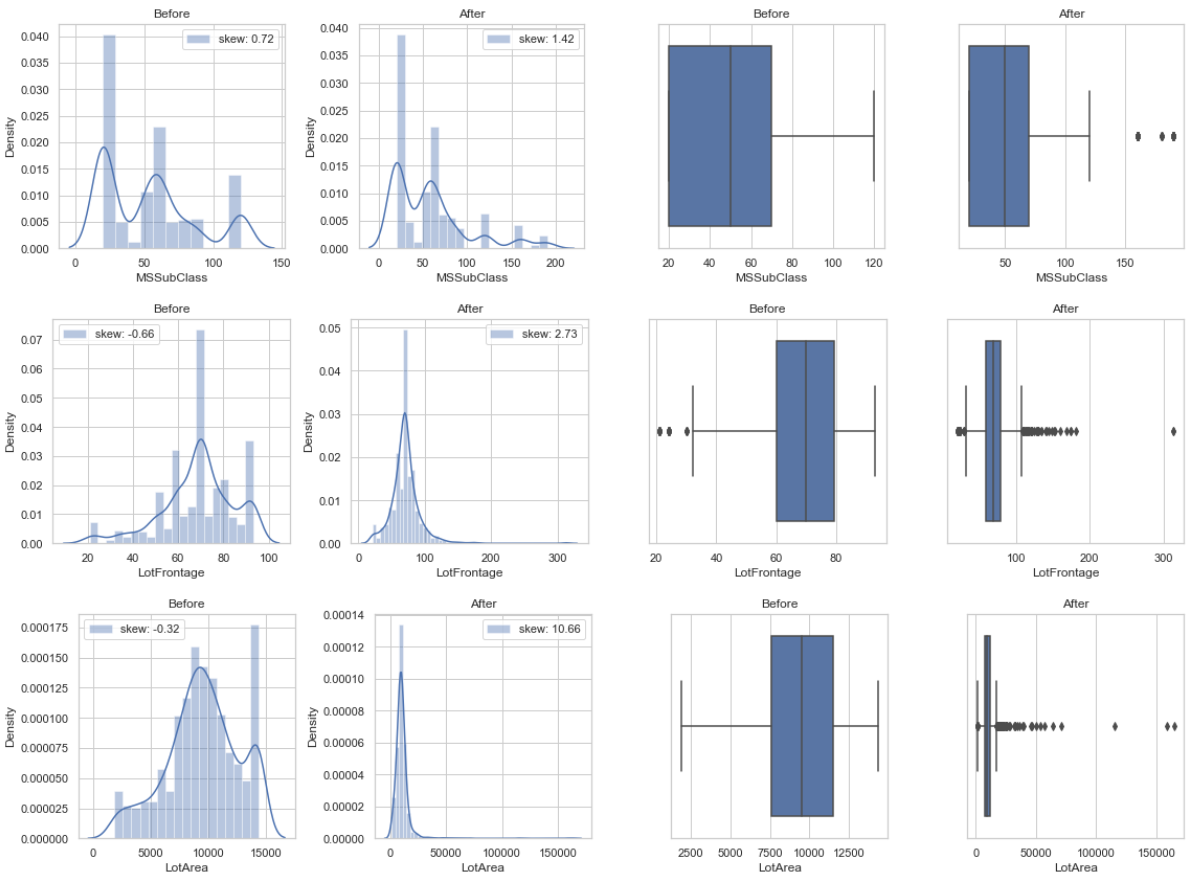
………….. and more ..

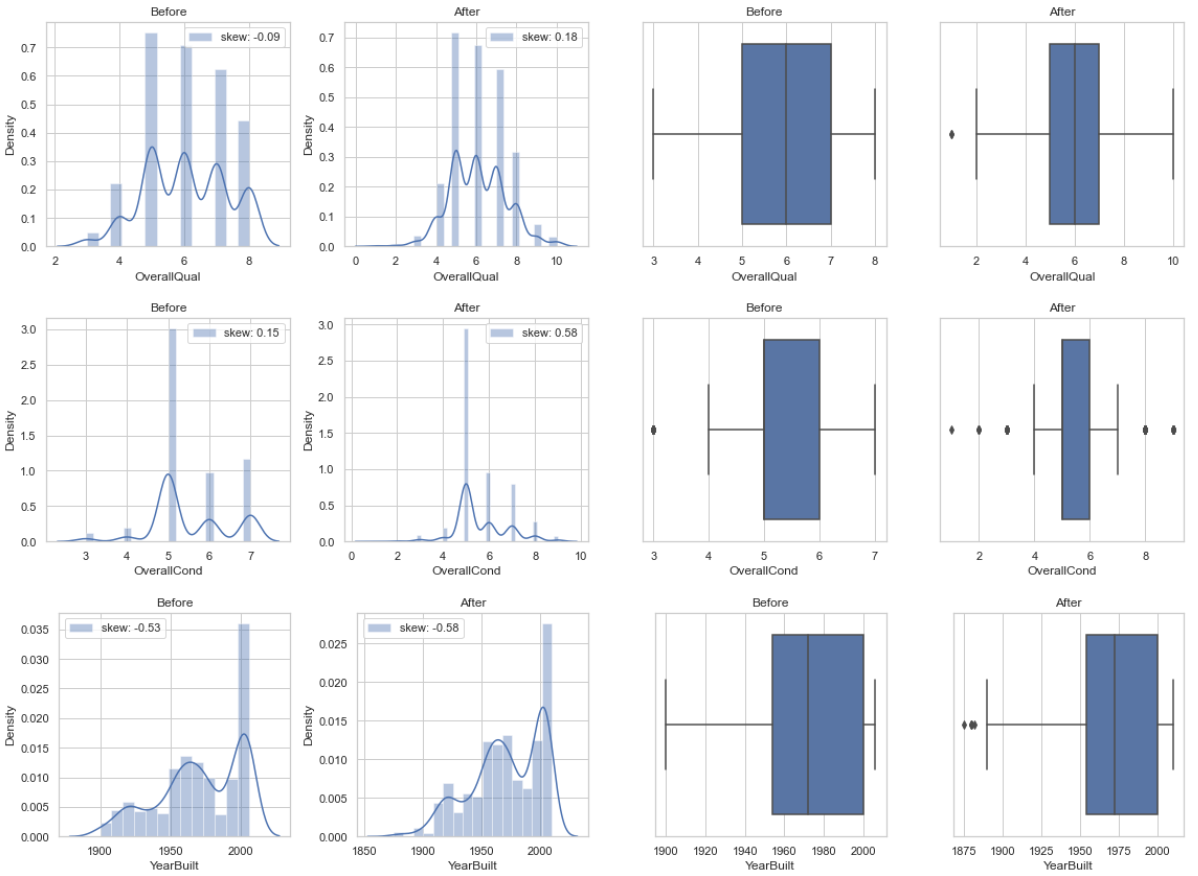
So , now there is no object type data is there in our dataset .

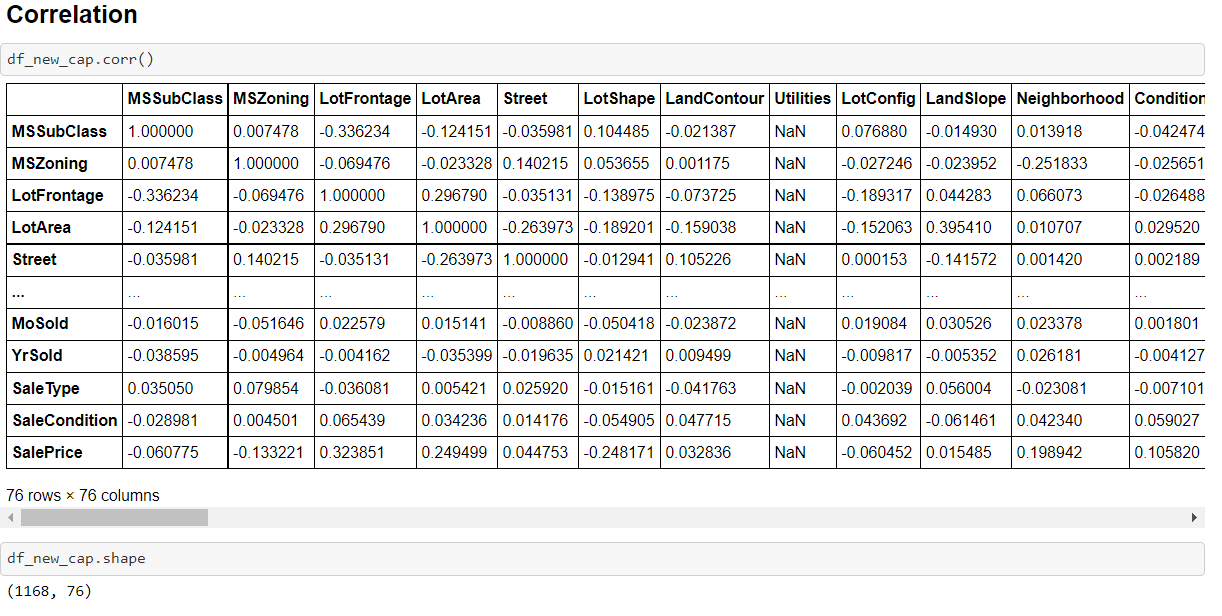
**Outliers and Skewness with winsorize Method :**

**Winsorizing** is another technique to deal with outliers and is named after Charles Winsor. In effect, Winsorization clips outliers to given percentiles in a symmetric fashion. For instance, we can clip to the 5th and 95th percentile. SciPy has a winsorize() function, which performs this procedure. The data for this recipe is the same as that for the Clipping and filtering outliers recipe.



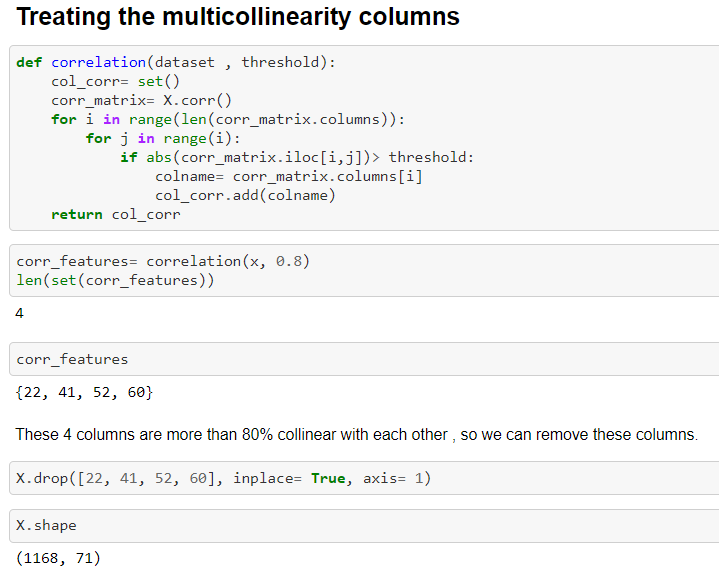
Output : 



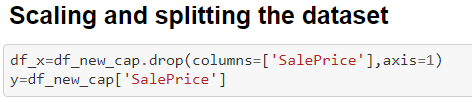


Now we,ll treat the multicollinearity columns :

**Multicollinearity** is the occurrence of high intercorrelations among two or more independent variables in a multiple regression model.



Before the above step we did the splitting if the data as well mentioned below:



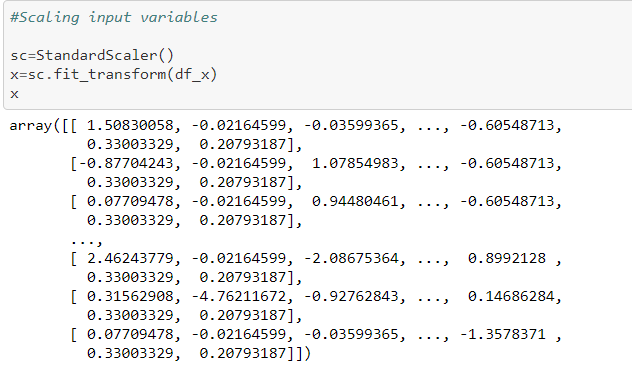
Through above code we splitted the data into two categories ,one : Feature columns and 2. Target columns , and stored them in different variables i.e. x and y .

**Scaling of Data :** When your data has different values, and even different measurement units, it can be difficult to compare them.Data scaling is a recommended pre-processing step when working with many machine learning algorithms.

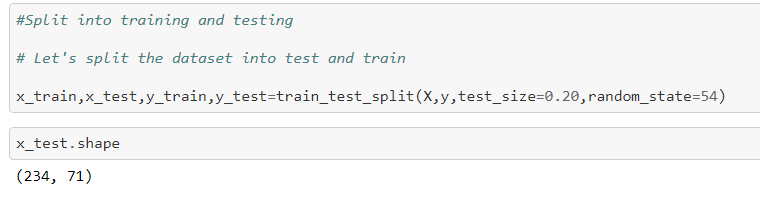
Data scaling can be achieved by normalizing or standardizing real-valued input and output variables. We can apply the StandardScaler to the dataset directly to standardize the input variables.

We will use the default configuration and scale values to subtract the mean to center them on 0.0 and divide by the standard deviation to give the standard deviation of 1.0. First, a StandardScaler instance is defined with default hyperparameters.

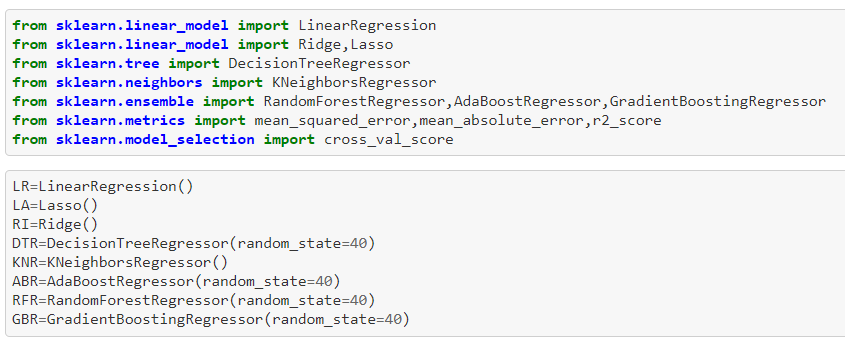
Once defined, we can call the fit\_transform() function and pass it to our dataset to create a transformed version of our dataset .

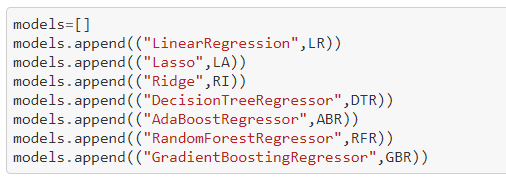


Now , our data is ready to go for training and testing , we’ll split the data in train and test . to fit the train data to models to learn , and test data to test the accuracy of the model .



**ML Alogrirhms :**



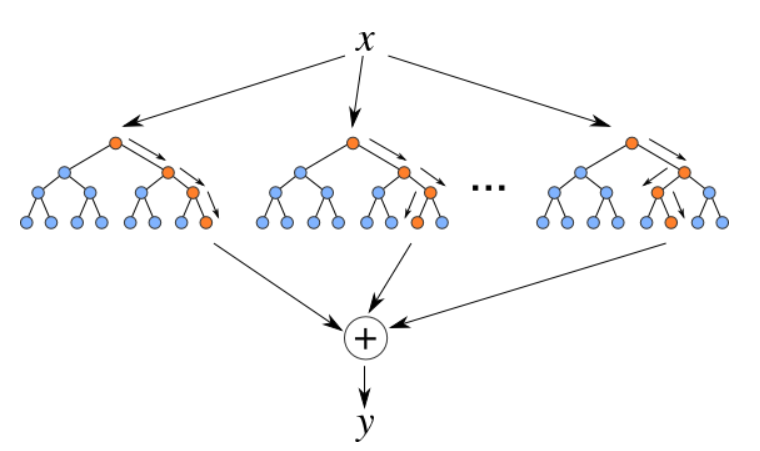


**Various Regression models we used :**

**Linear Regression : Linear Regression** is a machine learning algorithm based on **supervised learning**. It performs a **regression task**. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used.

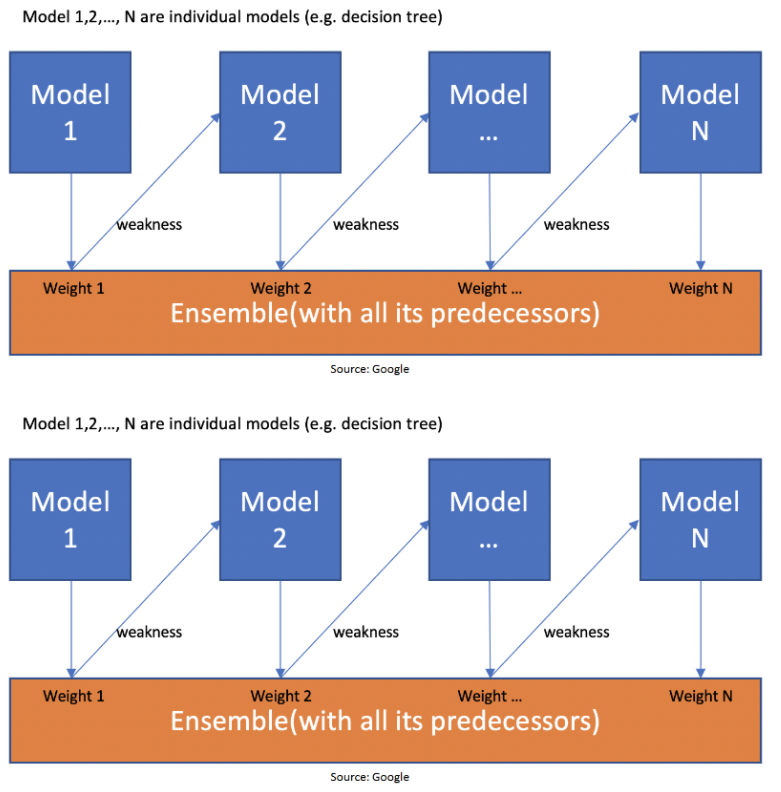
**Lasso and Ridge :** Ridge and lasso regression **allow you to regularize ("shrink") coefficients**. This means that the estimated coefficients are pushed towards 0, to make them work better on new data-sets ("optimized for prediction"). This allows you to use complex models and avoid over-fitting at the same time.

**Decision Tree Regressor : Decision Tree** is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs, and utility.  
Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.

**Random Forest Regressor :** 

**Random Forest Regression** is a supervised learning algorithm that uses **ensemble learning** method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

**Adaboost Regressor** : AdaBoost algorithm, short for Adaptive Boosting, is a Boosting technique used as an Ensemble Method in [Machine Learning](https://www.mygreatlearning.com/blog/what-is-machine-learning/). It is called Adaptive Boosting as the weights are re-assigned to each instance, with higher weights assigned to incorrectly classified instances. Boosting is used to reduce bias as well as variance for supervised learning. It works on the principle of learners growing sequentially. Except for the first, each subsequent learner is grown from previously grown learners. In simple words, weak learners are converted into strong ones. The AdaBoost algorithm works on the same principle as boosting with a slight difference.



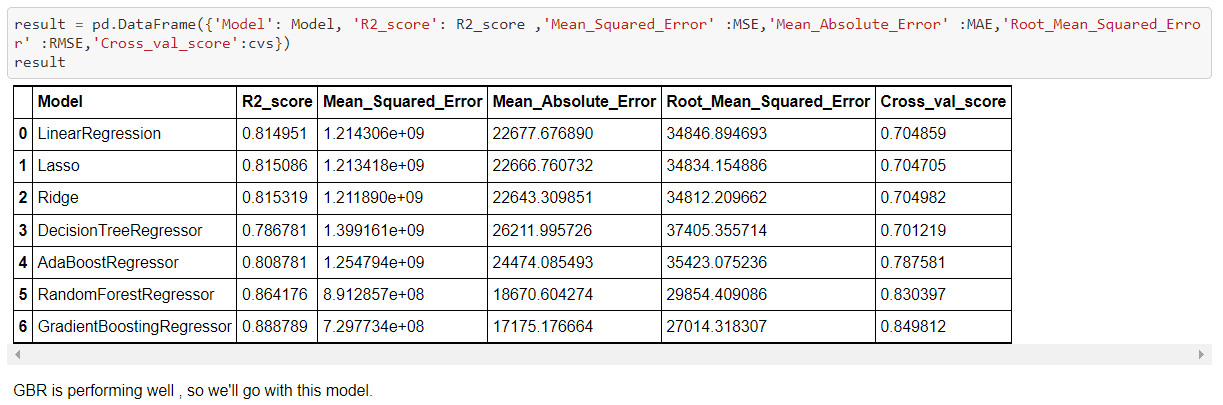
**Gradient Boosting** : Gradient boosting algorithm is one of the most powerful algorithms in the field of machine learning. As we know that the errors in machine learning algorithms are broadly classified into two categories i.e. Bias Error and Variance Error. As gradient boosting is one of the boosting algorithms it is used to minimize bias error of the model.

Unlike, Adaboosting algorithm, the base estimator in the gradient boosting algorithm cannot be mentioned by us. The base estimator for the Gradient Boost algorithm is fixed and i.e. Decision Stump.

Now , lets fit the data in models :

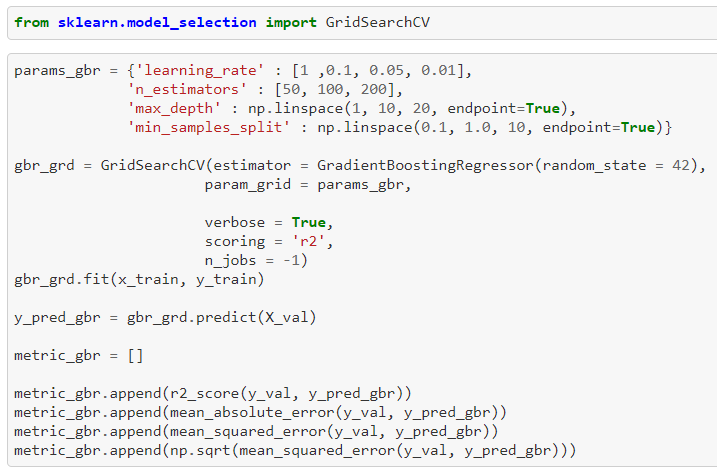


Output :

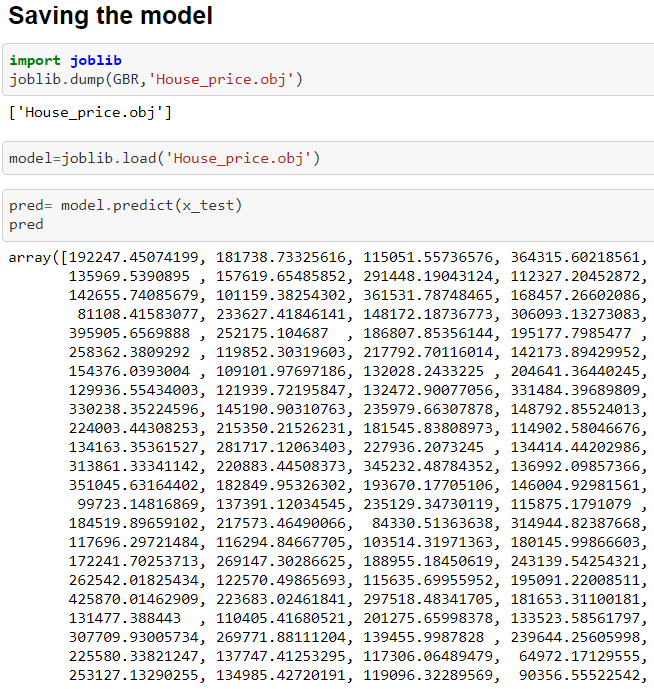


Gradient boosting regressor is performing the best , so we’ll select this .

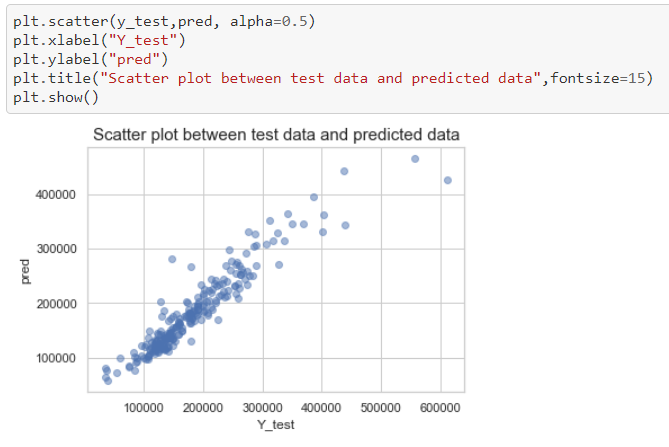
**Hyperparameter Tuning :** In machine learning, hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a **parameter whose value is used to control the learning process.**



**Saving the model and Prediction :**

**……..**

**Graphical representation of Actual vs Predicted data :**

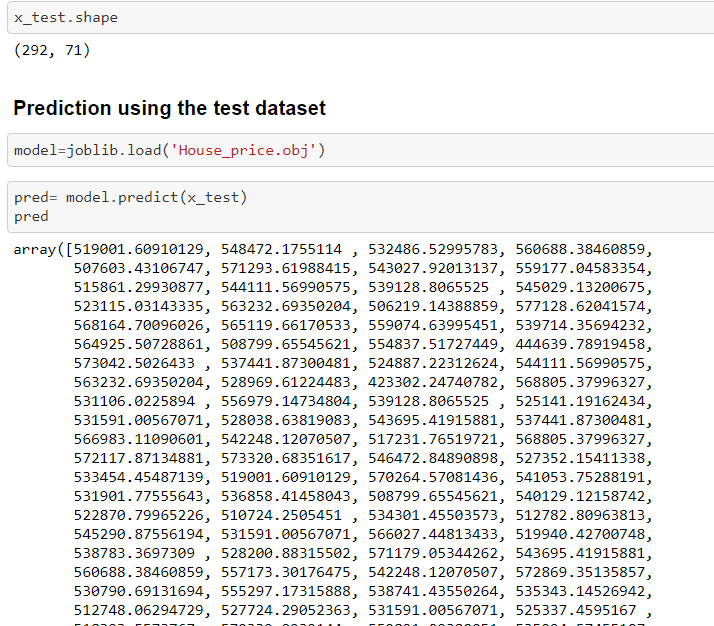


**Let’s Check the test dataset and do the testing or prediction :**

**Importing the test dataset :**



In above code , we imported the test dataset and done the all data cleaning and datapreprocessing things we did with training dataset , and doing the prediction using test dataset as mentioned below :



**We did Exploratory data Analysis on the features of this dataset and saw how each feature is distributed.**

**We did univariate , bivariate and multivariate analysis to see imapct of one another on their features using charts.**

**We analysed each variable to check if data is cleaned and normally distributed.**

**We cleaned the data and removed NA values**

**We treated the skewness and outliers using winsorize method .**

**We calculated correaltion between independent variables and check out the multicollinearity in the data and remove the highly correlated independent variables from the data .**

**Finally, we got a model with highest accuracy.**

**We tested the data and got the accuracy of 88 % with Gredient Boosting regressor .**

**And , predicted the house prices by using test dataset .**