## Global Power Plant Database Project

**In association with Data Trained Academy**

****

I am going to write about a complete end-to-end project for Global Power Plant Database.

I have written down all the steps in the form of sub-topics that I will be explaining one by one. And those sub-topics are as follows:  
  
1.      Problem Definition.  
2.      Data Analysis.  
3.      EDA Concludi ng Remark.  
4.      Pre-Processing Pipeline.  
5.      Building Machine Learning Models.  
6.      Concluding Remarks.

**Introduction**:

An affordable, reliable, and environmentally sustainable power sector is central to modern society. Governments, utilities, and companies make decisions that both affect and depend on the power sector. For example, if governments apply a carbon price to electricity generation, it changes how plants run and which plants are built over time.

On the other hand, each new plant affects the electricity generation mix, the reliability of the system, and system emissions. Plants also have significant impact on climate change, through carbon dioxide (CO2 ) emissions; on water stress, through water withdrawal and consumption; and on air quality, through sulfur oxides (SOx ), nitrogen oxides (NOx ), and particulate matter (PM) emissions. 2 | Despite the importance of the power sector, there is no global, open-access database of power plants. Existing databases fail to be either truly comprehensive or fully open. Many countries do not report their power sector data at the plant level, and those that do vary wildly in what they report, how they report it, and how frequently they report. The lack of reporting standards makes data gathering time intensive, as the data are in different formats and must be harmonized. This creates a barrier for conducting global and national research and analysis of the power sector.

Thanks to Data Science and Machine Learning, which has been very useful in many industries that have managed to bring accuracy or detect negative incidents. Here in this blog, I have created a Machine Learning model to predict the power plant data on the label of primary fue; and capacity in MW

Detect if the claim is fraudulent or not. Here various features have been used like insured information, insured persons, personal details and the incident information. In total the dataset has 40 features and 1000 entries rows of data. Using all these previously acquired information and analysis done with the data I have achieved a good model that has 95% accuracy. Let’s see what are the steps involved to attain this accuracy.

**Hardware & Software Requirements & Tools Used:**

### **Hardware used:**

Processor: Core i5 -10300H CPU @ 2.50GHz HP Pavilion Laptop 15-cc1xx

RAM: 8 GB

Operating System: 64-bit

ROM/SSD: 256GB SSD

Graphics: NVIDIA GeForce 940MX

Intel(R) UHD Graphics 620

### **Software requirement:**

Anaconda Navigator - Jupyter Notebook

### **Libraries Used:**

Numpy

Pandas

Matplotlib

Seaborn

Scipy

Date Time

Scikit Learn

1.Problem Definition.

The Global Power Plant Database is a comprehensive, open source database of power plants around the world. It centralizes power plant data to make it easier to navigate, compare and draw insights for one’s own analysis. The database covers approximately 35,000 power plants from 167 countries and includes thermal plants (e.g. coal, gas, oil, nuclear, biomass, waste, geothermal) and renewables (e.g. hydro, wind, solar). Each power plant is geolocated and entries contain information on plant capacity, generation, ownership, and fuel type

Make two prediction for labels​

1.primary\_fuel

2.capacity\_mw

2.Data Analysis.

In order to build a Machine Learning Model, we have a Machine Learning Life Cycle that every Machine Learning Project has to touch upon in the life of the model. Let’s take a look at the model life cycle and then we will look into the actual machine learning model and understand it better along with the lifecycle as we move forward.



Now that we understand the lifecycle of a Machine Learning Model, let’s import the necessary libraries and proceed further.

**Importing the necessary Libraries:**

To analyze the dataset or even to import the dataset, we have imported all the necessary libraries as shows below.

Pandas has been used to import the dataset and also in creating data frames.

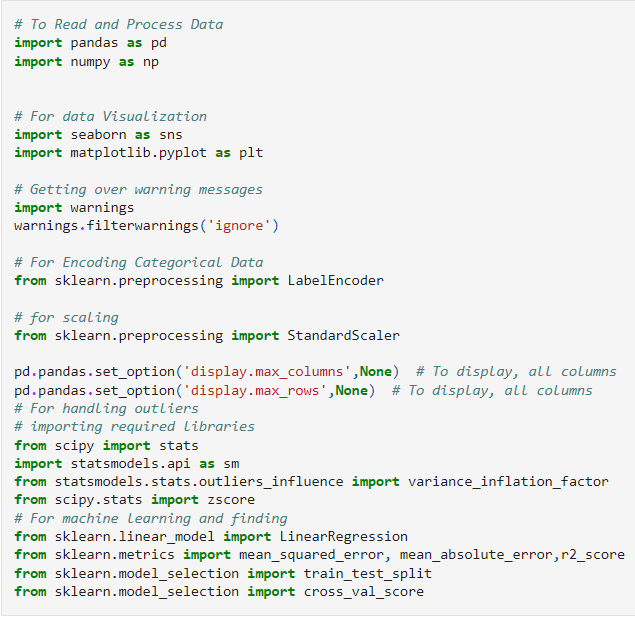
Numpy has been used for numerical tasks.

Seaborn and Matplotlib have been used for Data Visualization.

Date Time has been used to extract day/month/date separately.

Scipy has been used in the Zscore method for removing outliers.

Sklearn has been used in the model building.



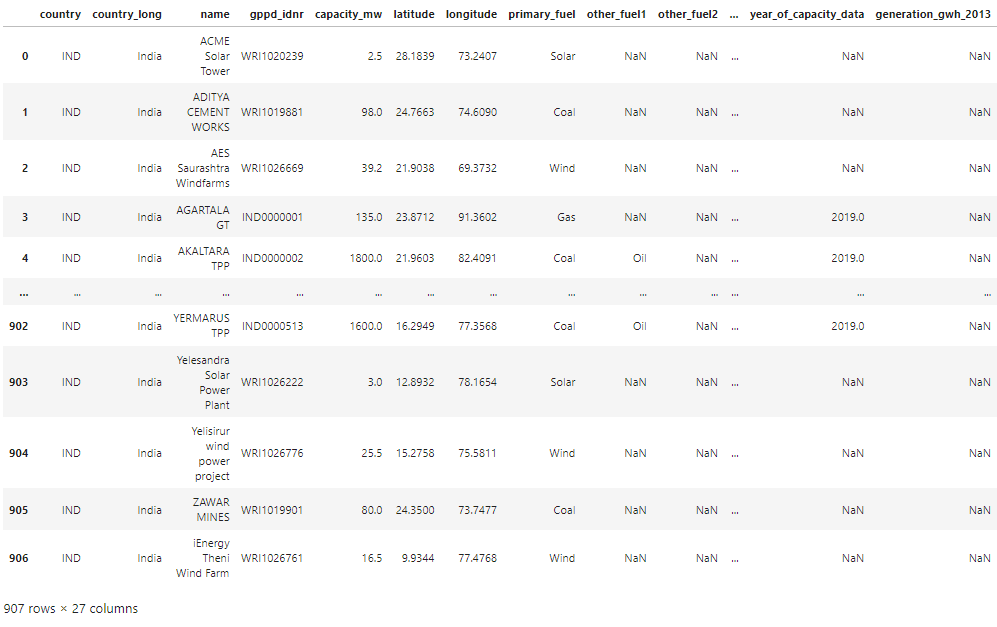
Importing the Dataset

Let’s import the dataset first.

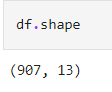


Copied the raw data and saved it as a csv file on my local computer after which I imported the entire dataset on this Jupyter Notebook with the help of pandas.

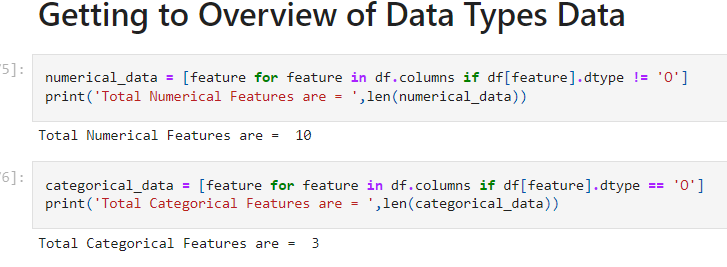
I have imported the dataset which was in “csv” format as “df”. Below is how the dataset looks.

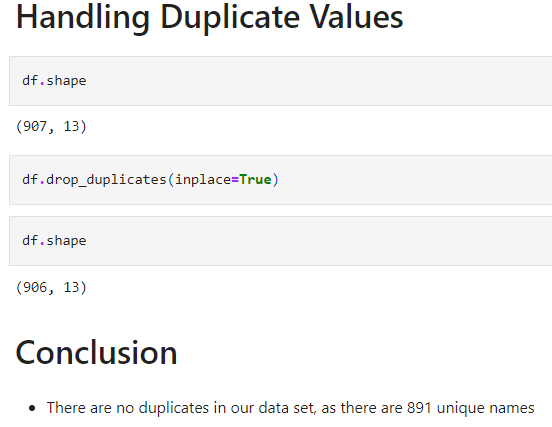


Checking the shape of the data

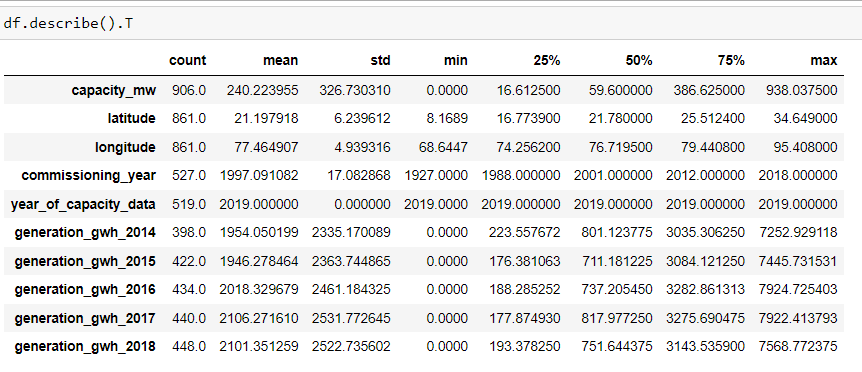


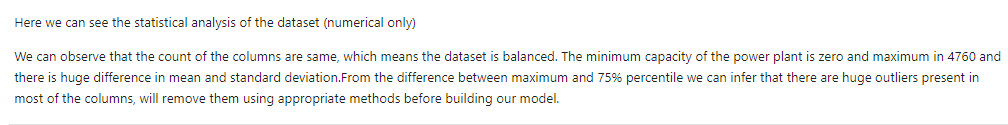
There are 907 rows and 13 columns in our dataframe



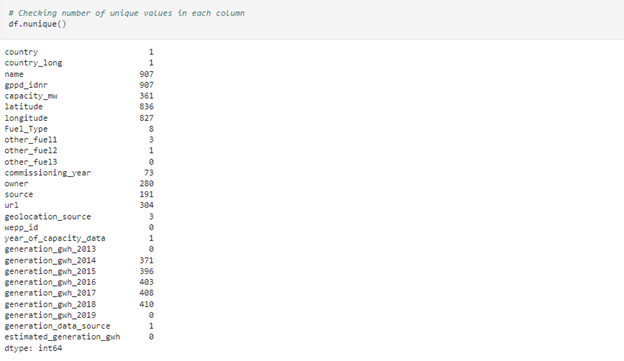


**Statistical Summary of Dataset**





**Unique Value of Dataset**

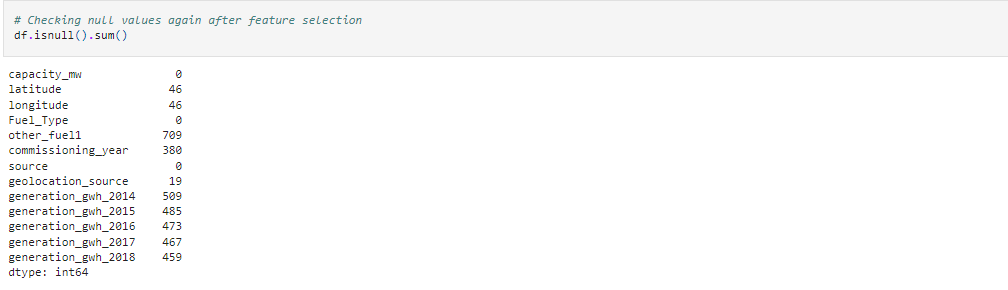


the column with one unique value are country, country\_long, other\_fuel2, year\_of\_capacity\_data and generation\_data\_source

other\_fuel3, wepp\_id,generation\_gwh\_2013, generation\_gwh\_2019, estimated\_generation\_gwh have no unique values which means they are filled with only NAN values

Here the columns have only one unique value.. Since these columns have same entries thoughout the dataset so we can drop these columns.

**Null Value in Dataset**

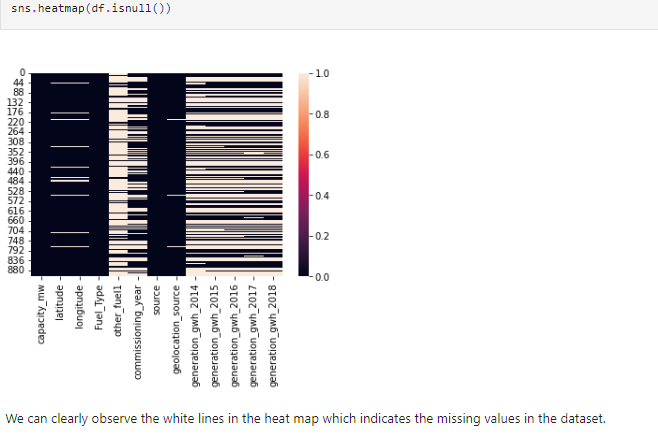


Observations:

We can see that there are null values in the dataset.

The dataset contains 3 different types of data namely integer data type, float data type and object data type.

**Plotting the graph of null values**

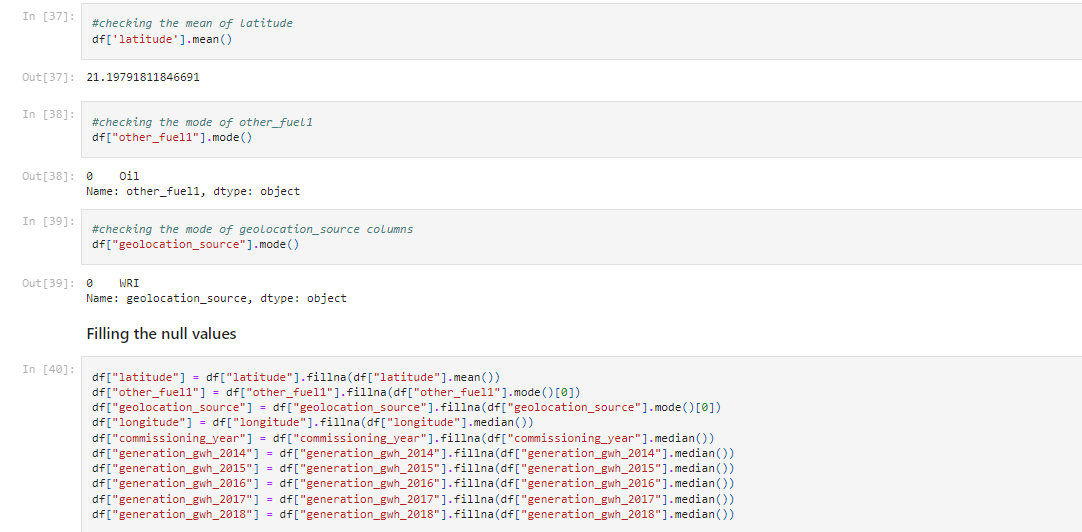


**Imputation:**

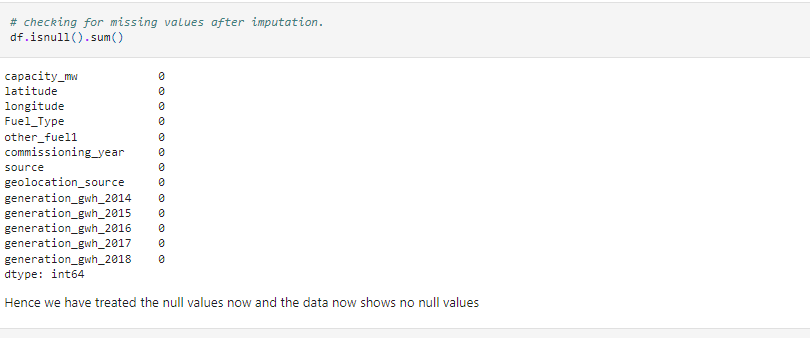
Dealing with missing value by using imputation technique

For missing value can be filled by using median in numerical variable and

For categorical variable we use mode



**Again checking for missing value**



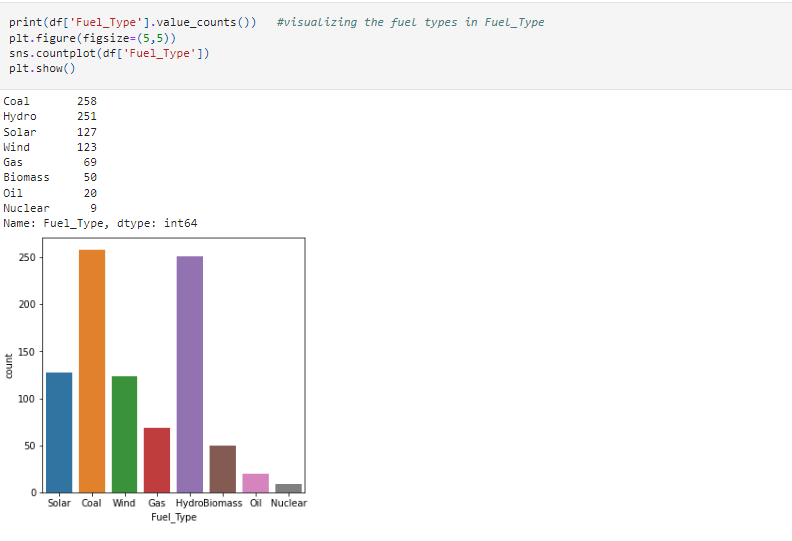
Clearly there is no null values

**3.Exploratory Data Analysis (EDA)**

For exploratory data analysis we go through three stage of analysis

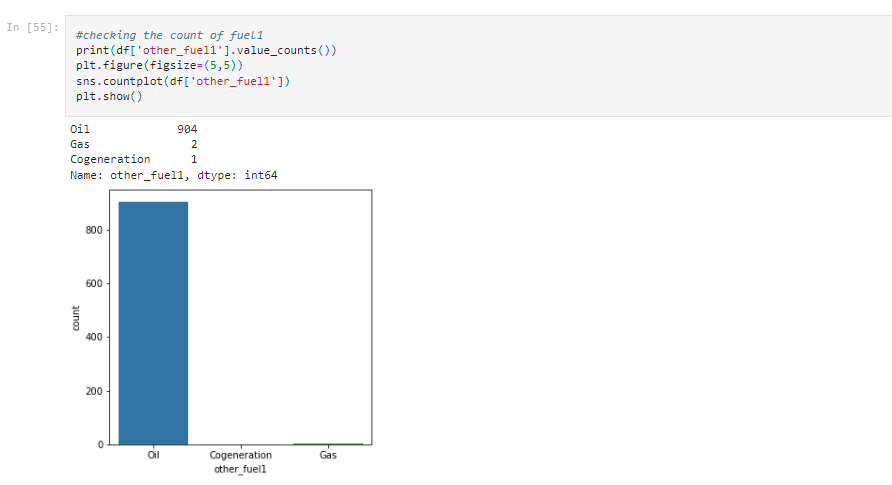
**3.1 Univariate Analysis**

Categorical column visualization

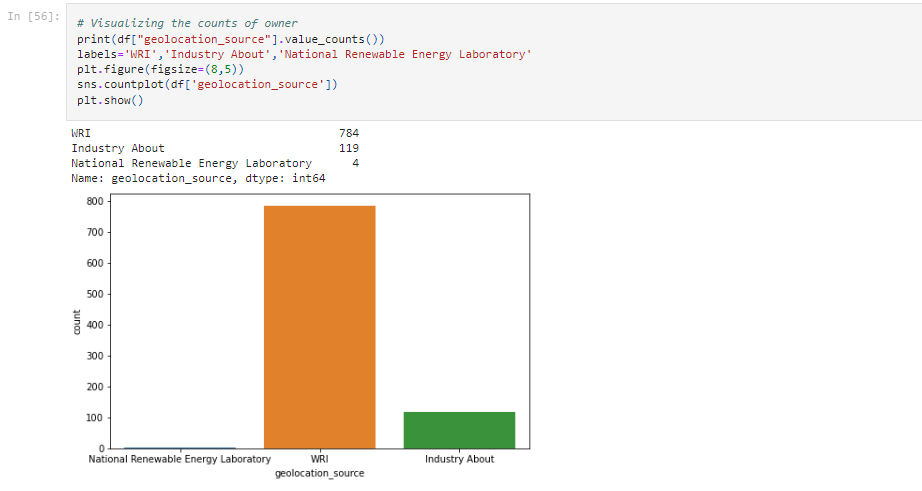


In the above count plot for "primary\_fuel" column we can see that the highest number of values have been covered by coal and hydro fuel types then comes solar and wind. Finally we see that gas, biomass, oil and nuclear have very low data counts.

However when we will be considering "primary\_fuel" as our target label then this is impose a class imbalance issue while trying to create a classification model and therefore will need to be treated accordingly

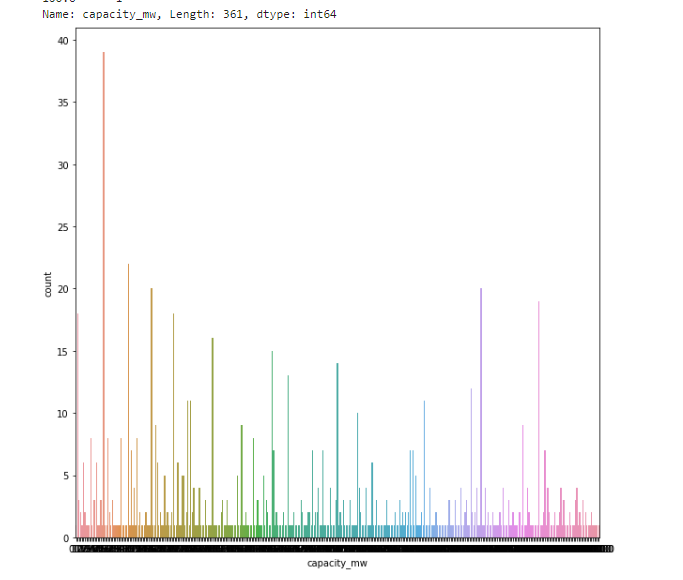


It can be observed that 'other\_fuel1' type has 3 unique types namely: 'Oil', 'Cogeneration other fuel', 'Gas'. And it is clearly seen that oil is the max used fuel type.



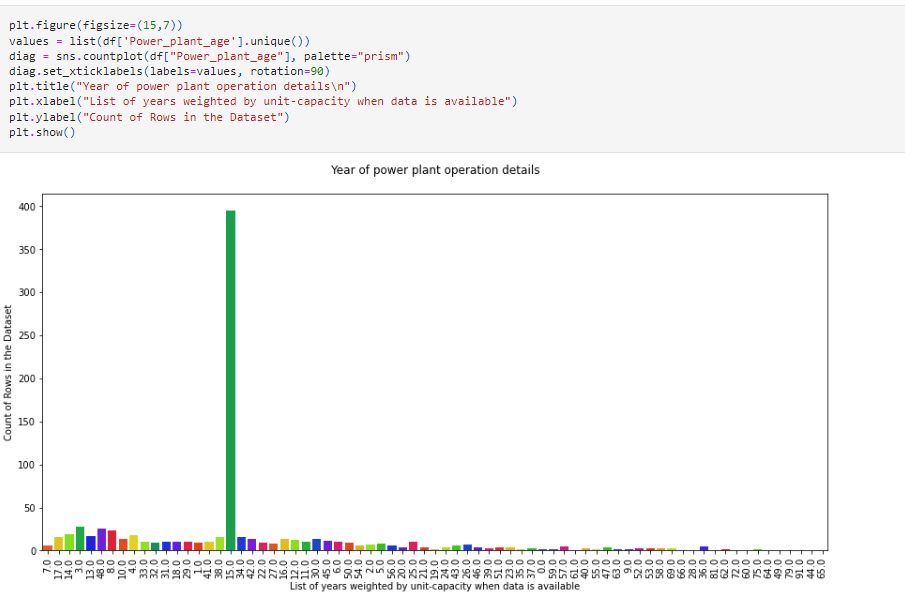
Here it can be seen that the count of WRI is the max, which means that the max information is shared by this source.





Observation:

Here it can be seen the counts with respect to capacity mw.

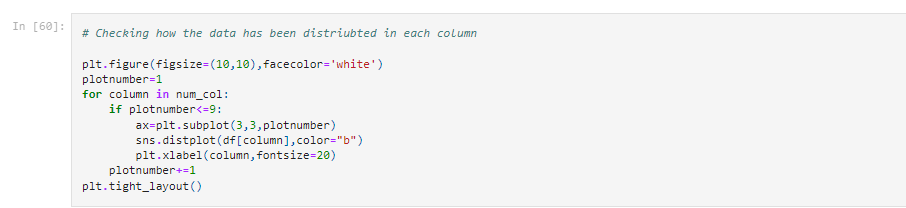


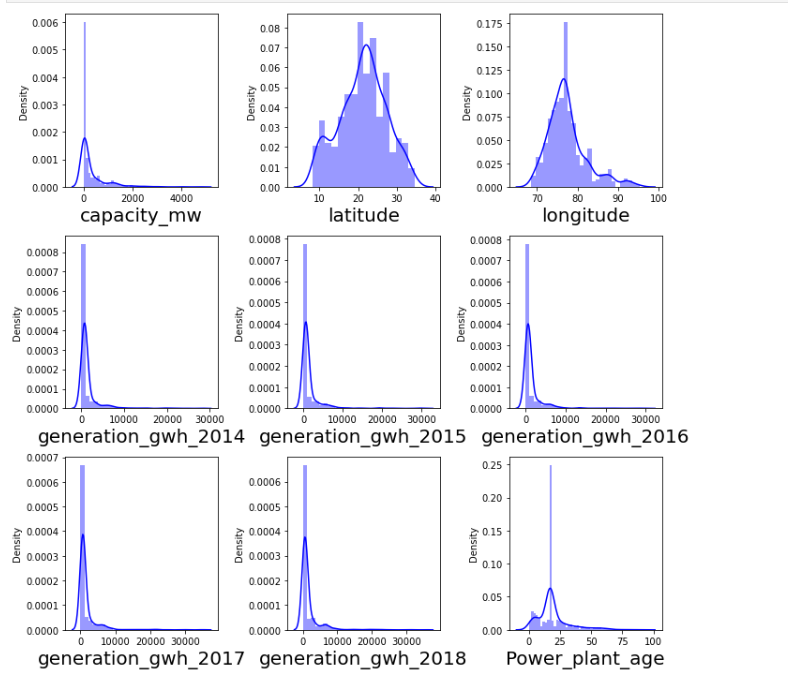
Observation:

In the above count plot we can see the list of years as to when the power plant data was made available. Since we had missing values in the "commissioning\_year" column we replaced it with the median wherein the year "15" covered the most rows in our dataset compared to all the other years.

]

**Checking the Distribution of the Dataset, if it is normal**



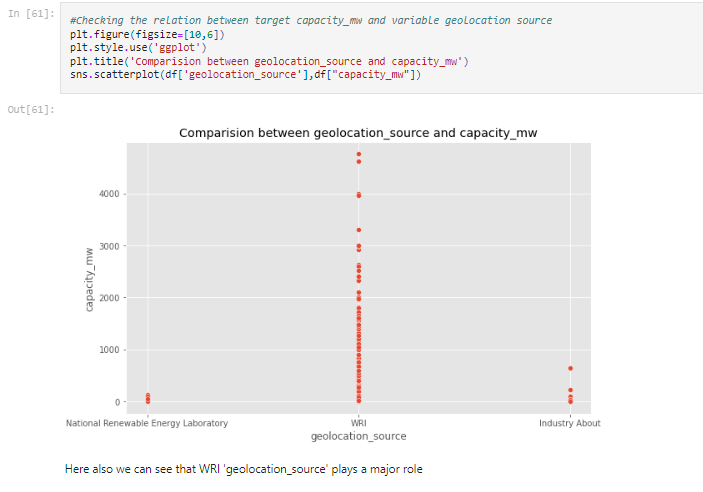
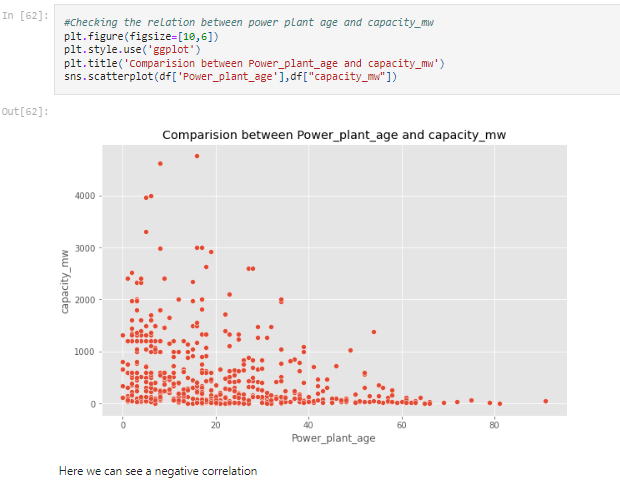


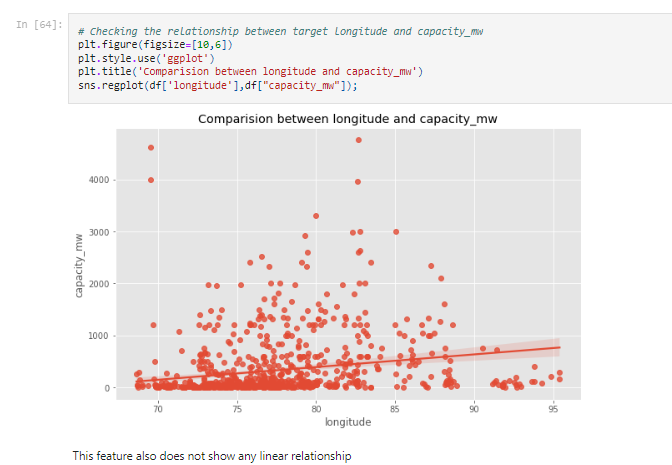
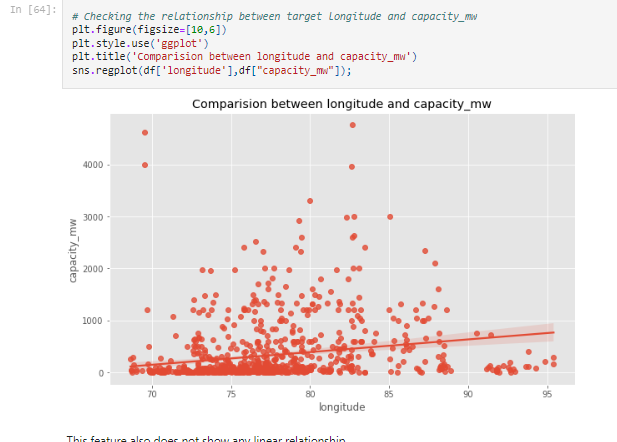
Observation:

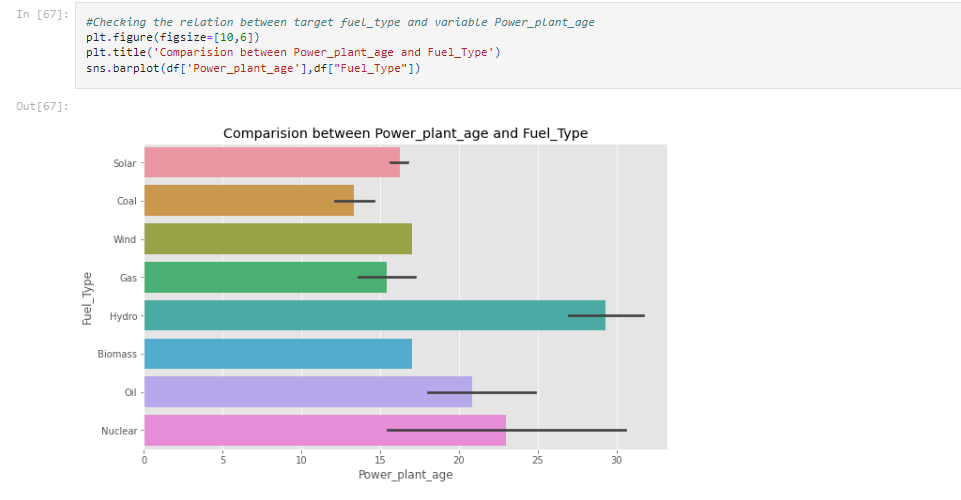
Here in the plots we can see that the data is not normally distributed. Outliers and skewness is present, which needs to be treated.

**3.2 Bivariate Analysis**

Correlation between features and target 'Capacity\_mw'

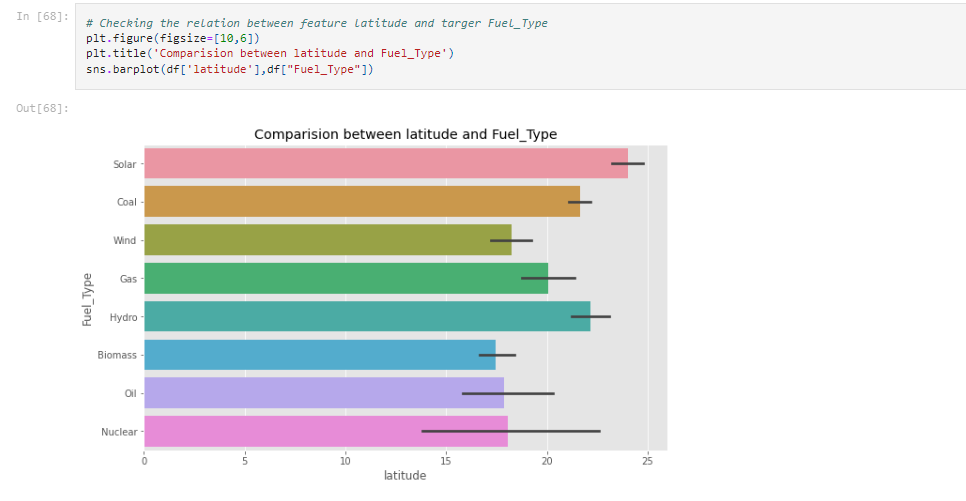






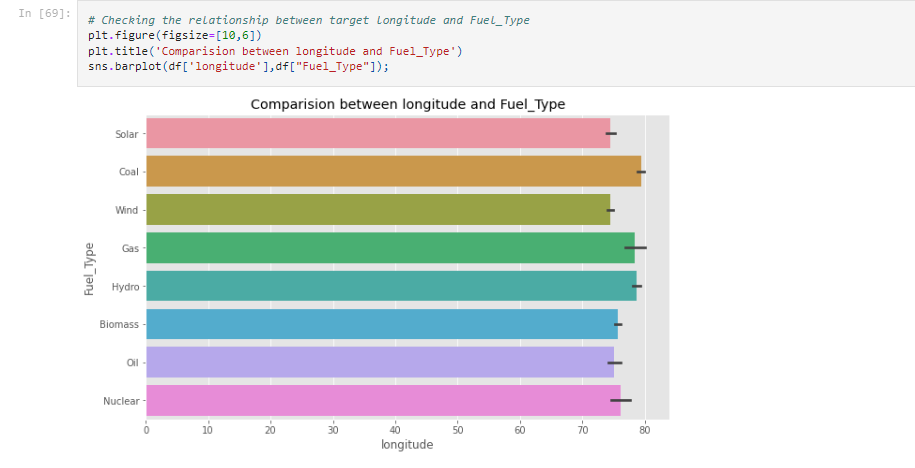
Observation:

Here we can see that older power plants uses Hydro as energy source, followed by oil. The newer power plants are using more of Coal, Solar and Gas



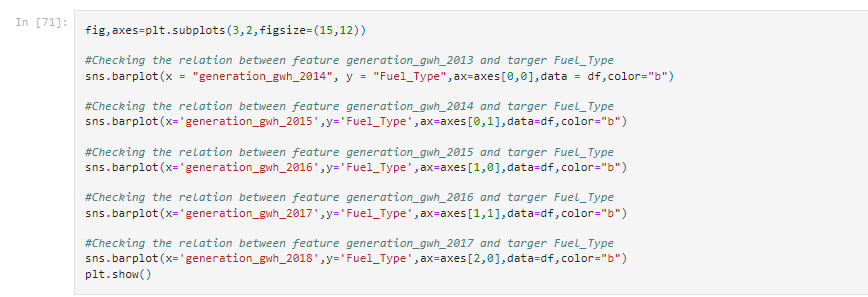
Observation:

Solar has the highest latitude



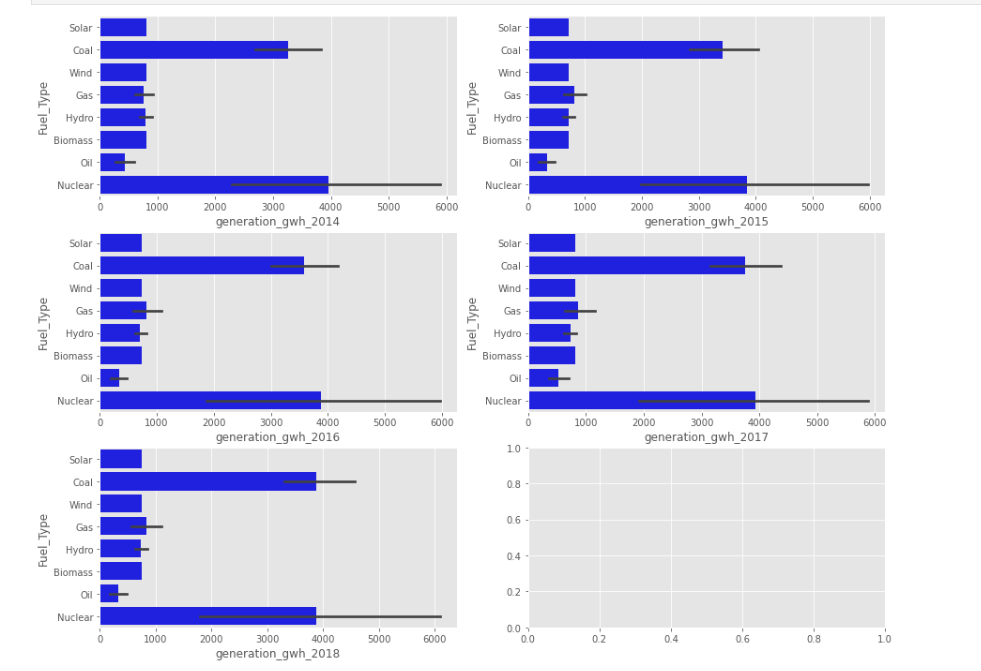
Observation:

Here Gas shows the highest longitude



Observation:

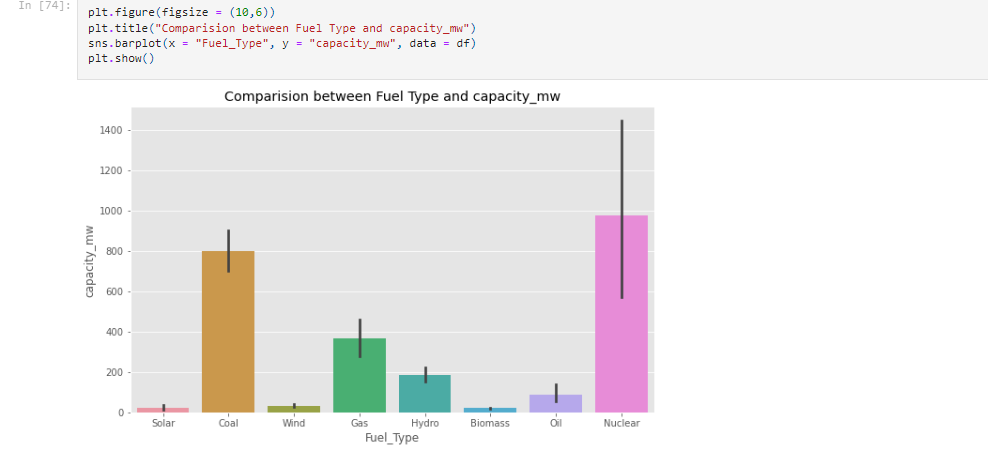
Here we can see that the most used energy source



Observation:

all the years is nuclear followed by coal

**Checking the relationship between both the targets**

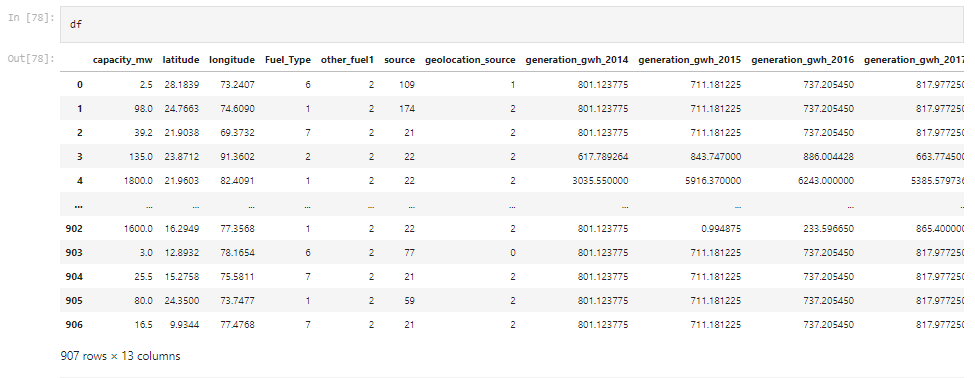


Observation:

Here also it shows that energy source Nuclear has the major contribution

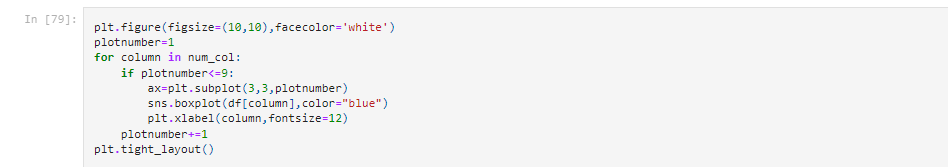
## Label Encoding

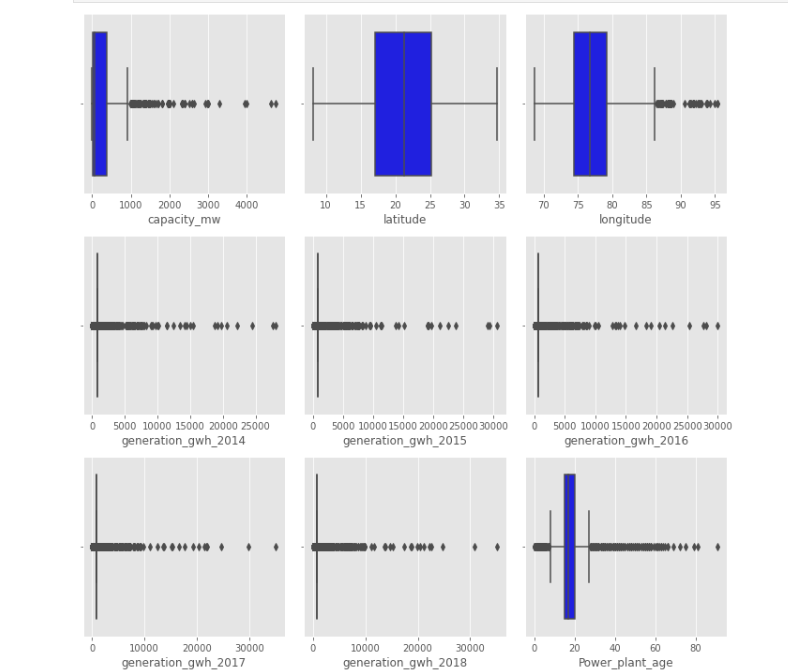
## 



**4.Pre-Processing Pipeline**

Identifying the outliers





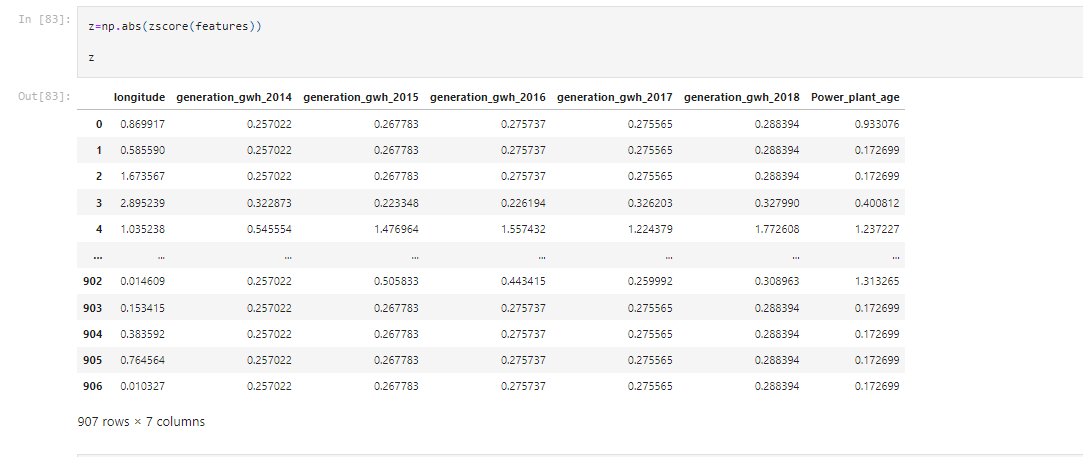
Observation:

In the boxplot we can notice the outliers present in all the columns except latitude. Even target column has outliers but no need to remove it. Let's remove outliers using Zscore method.

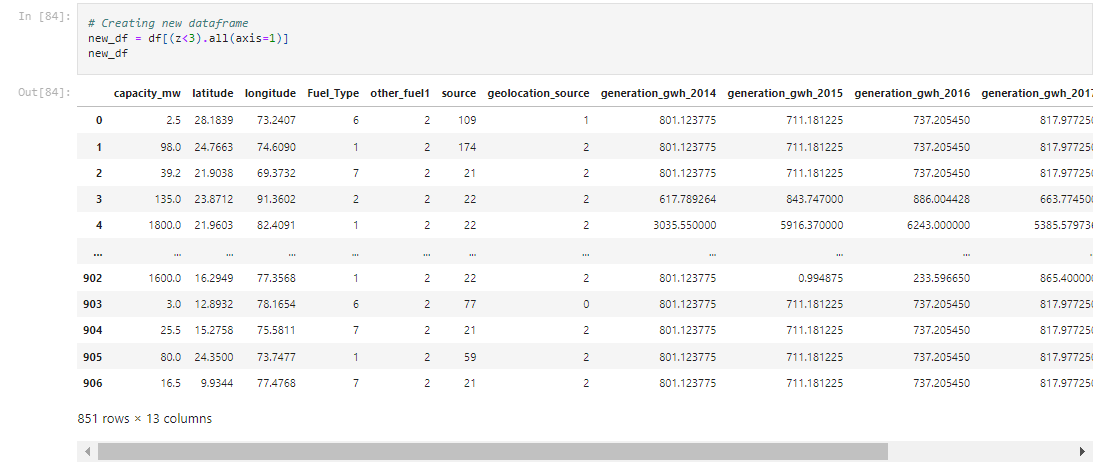
Feauture column with outlier



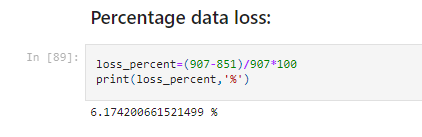
ZScore



Creating new dataframe



Percentage Data Loss

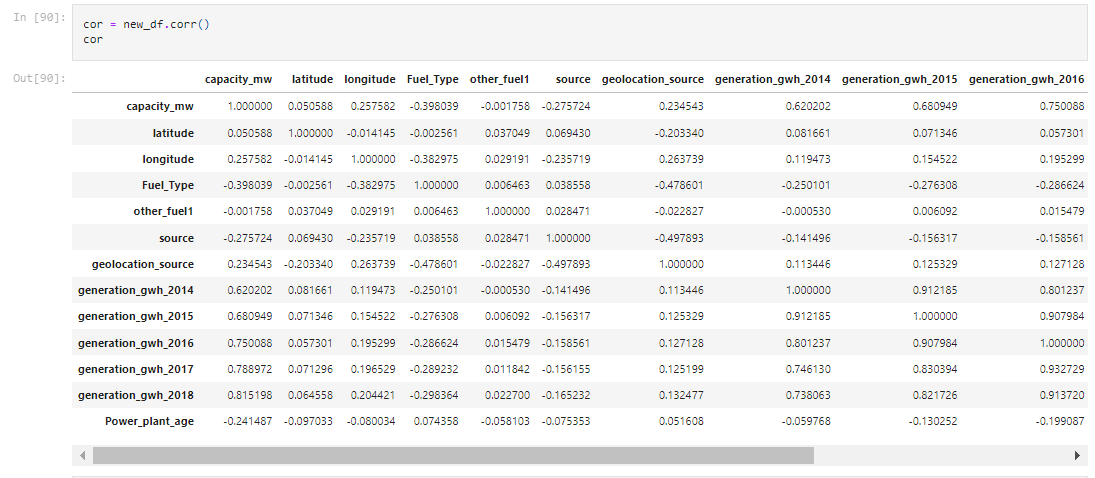


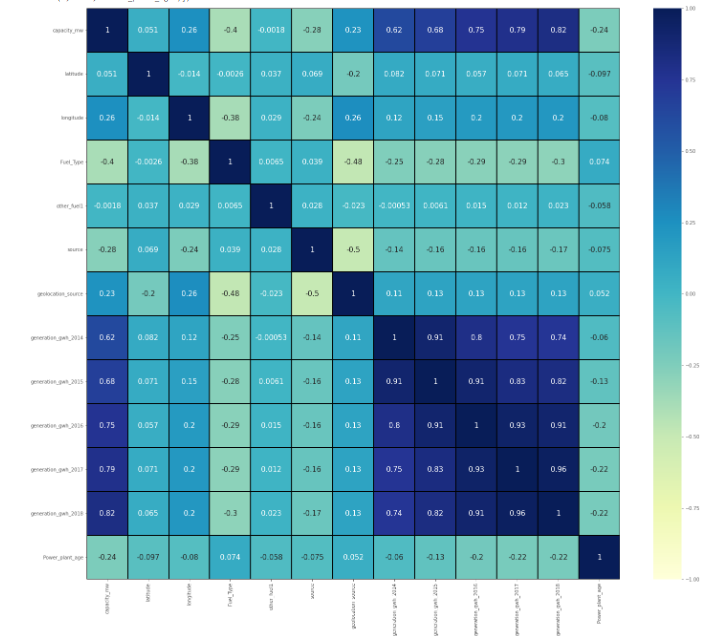
Observation:

checking the data loss percentage by comparing the rows in our original data set and the new data set after removal of the outliers. usually less than 10% data loss is acceptable

Correlation between the target variable and features

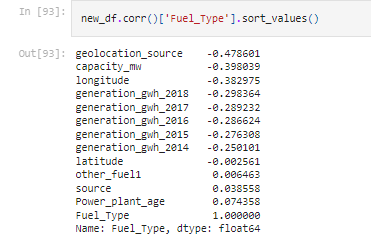
Checking The correlation



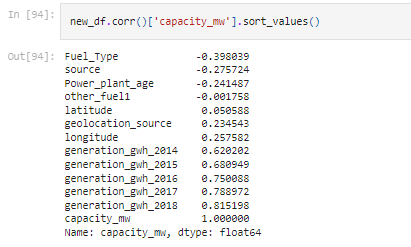


Observation:

From the heat map we can notice most of the features are highly correlated with each other which leads to multicollinearity problem. So will try to solve this problem by Checking VIF value before building our models.



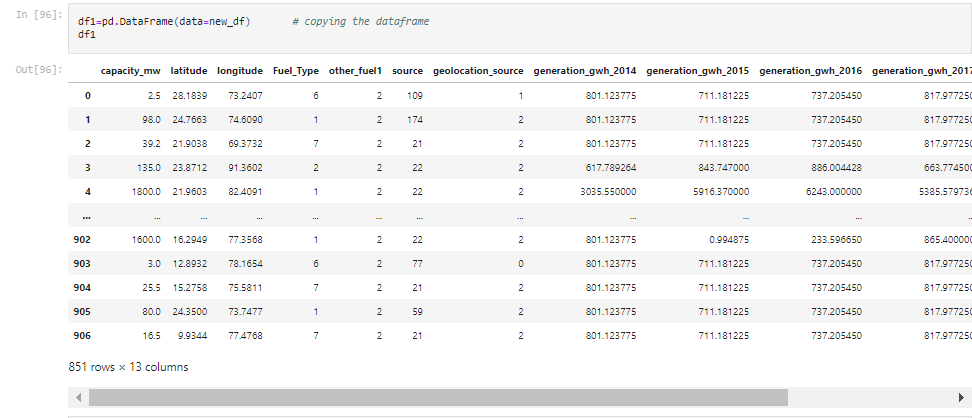
The label Fuel\_Type is less correlated with Power\_plant\_age and source. The label is negatively correlated with geolocation\_source, longitude, capacity\_mw, and all generation\_gwh years.

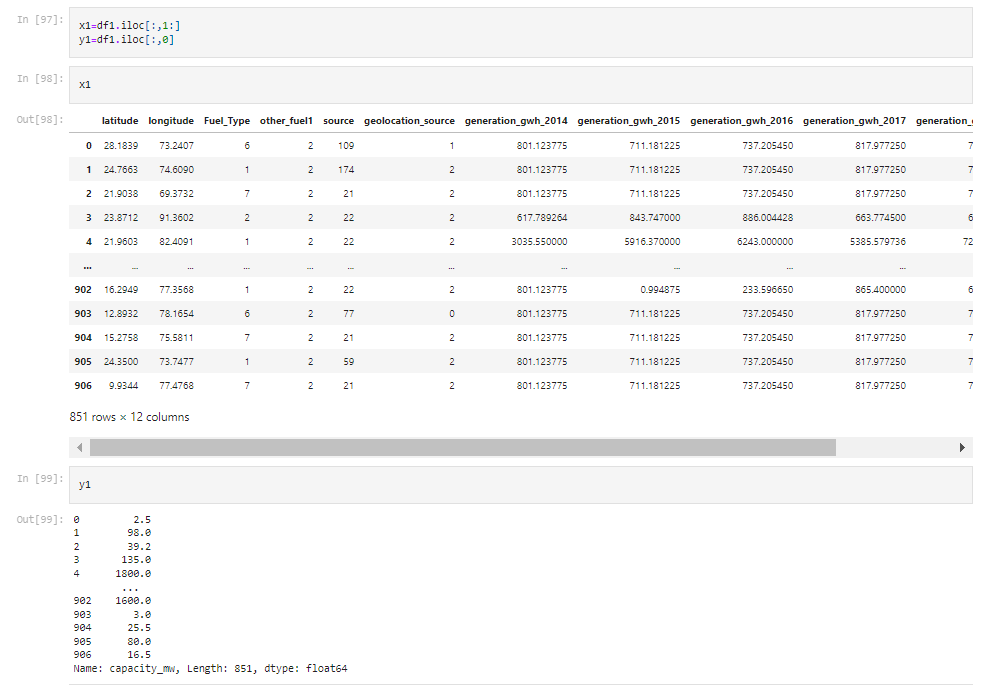


Here we can see the co-relation between all the features and the features and targets

The label capacity\_mw is highly positively correlated with the features generation\_gwh\_2017, generation\_gwh\_2016, generation\_gwh\_2015, generation\_gwh\_2014, generation\_gwh\_2013. And the label is negatively correlated with the features Fuel\_Type, source and Power\_plant\_age. The columns other\_fuel1 and latitude have no relation with the label, so we can drop them.

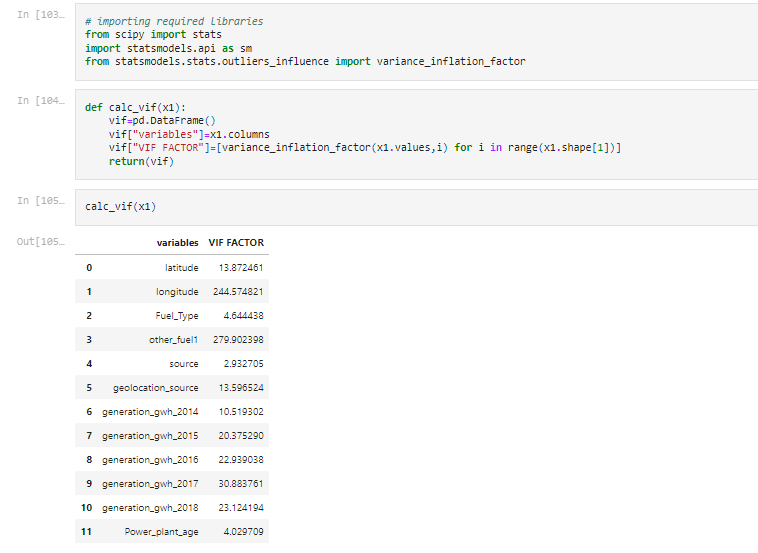
**MultiCollinearity with Variance Inflation Factor**

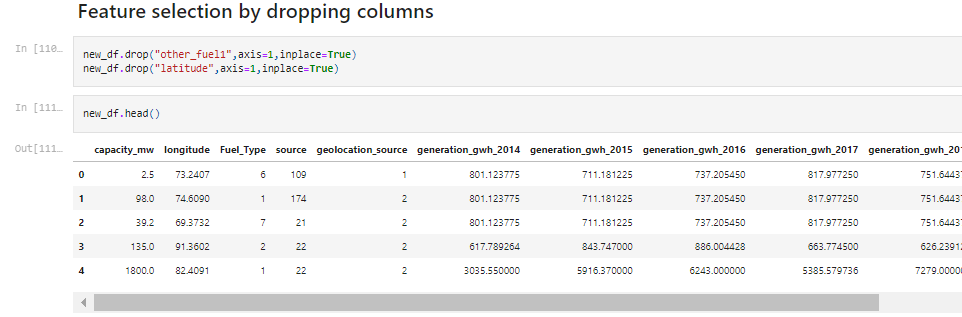




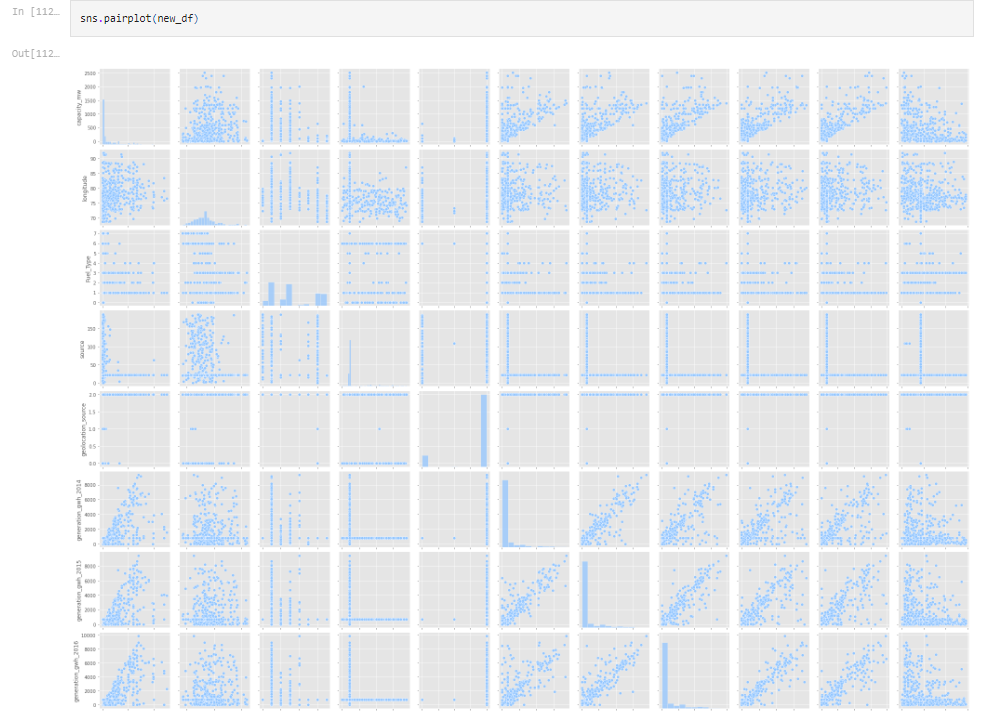
**Variable Inflation Factor**







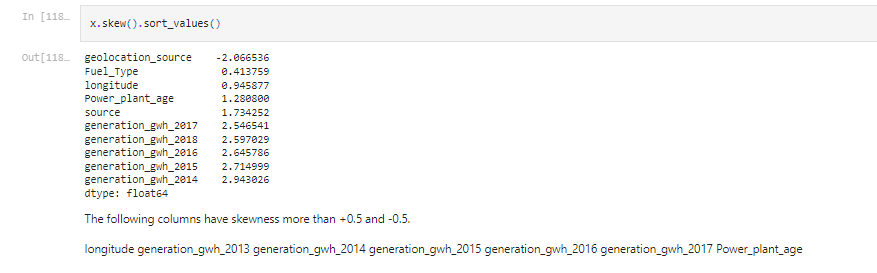
**3.3 Multivariate Analysis:**

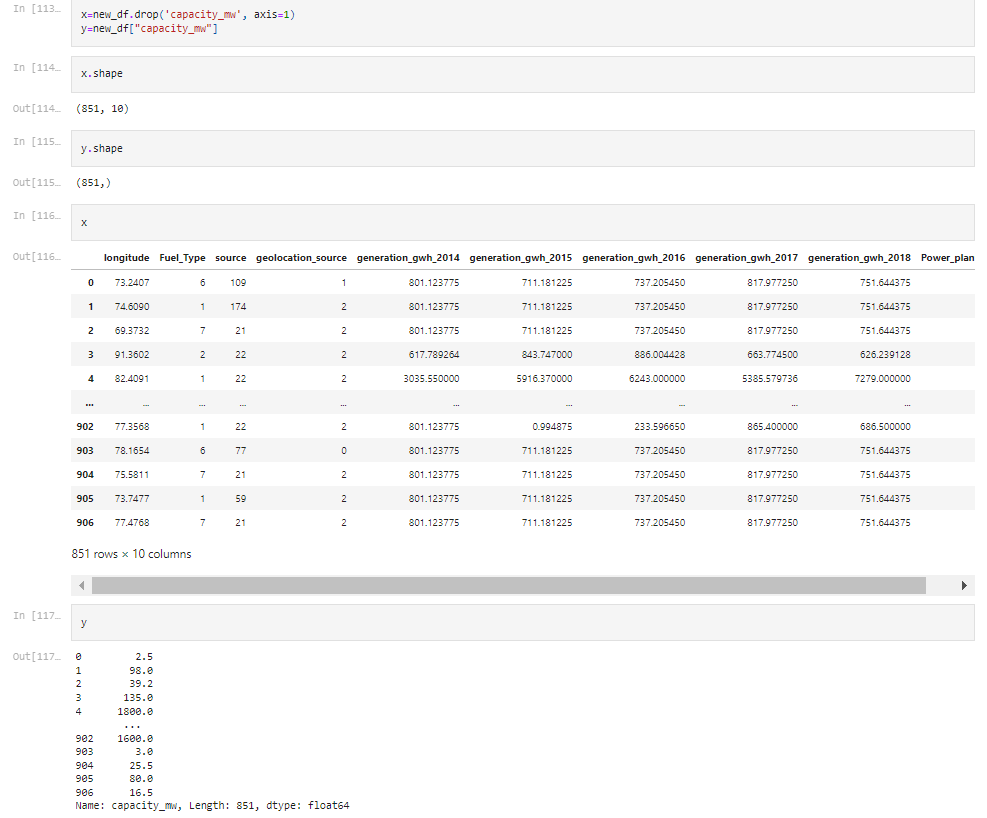


**5. Machine Learning Algorithm**

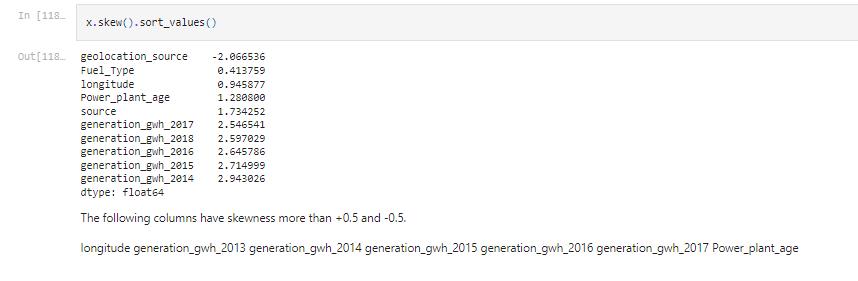
5.1 Predicting "Capacity\_mw" Target

Splitting the dataset into Features and Target

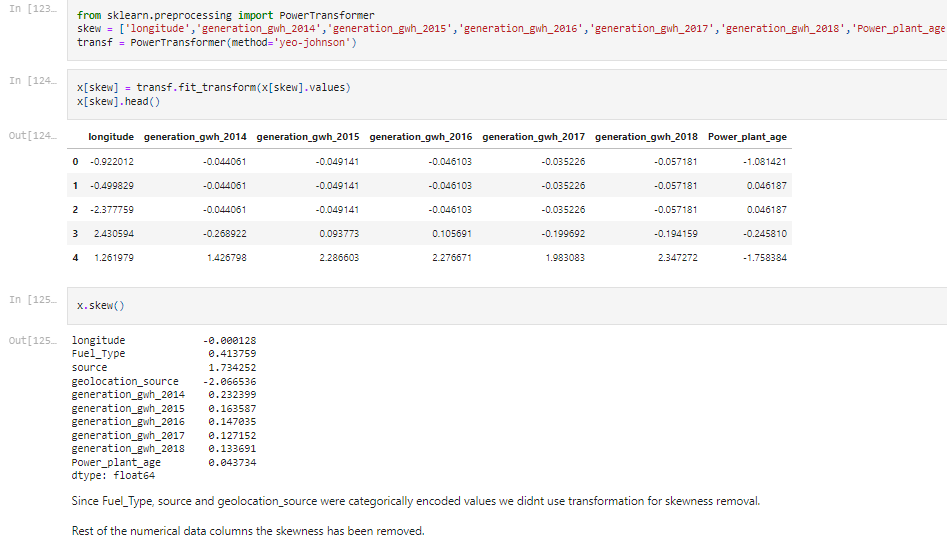




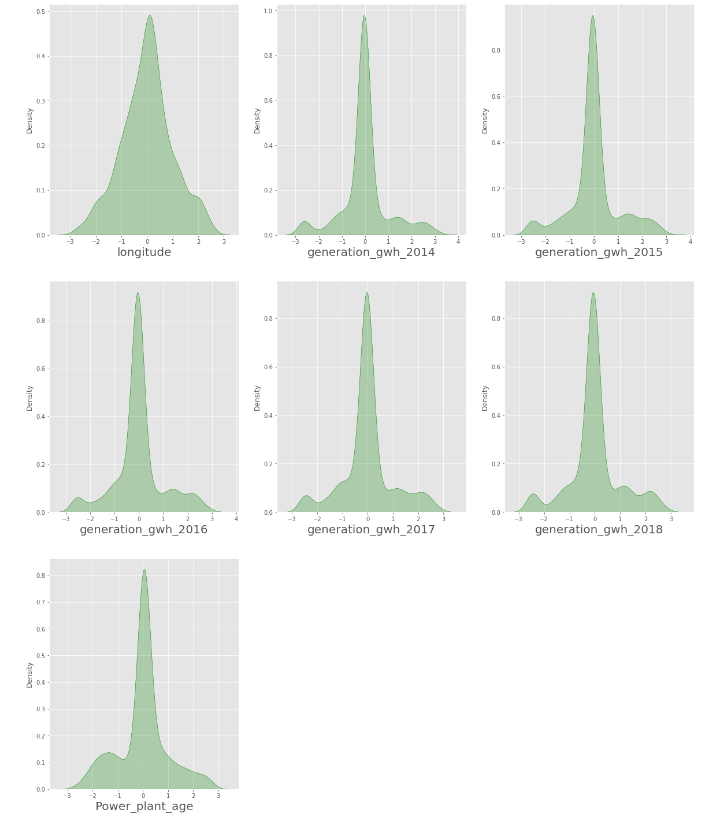
Checking for skewness



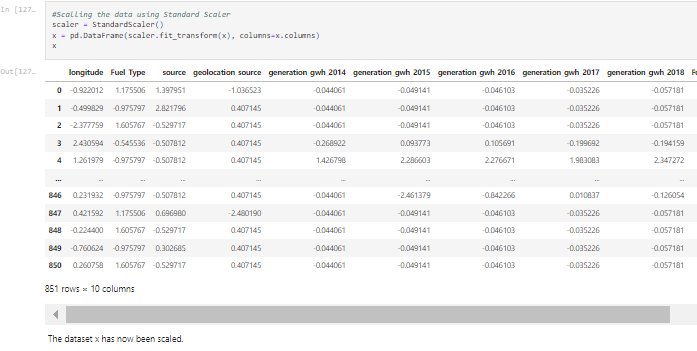
Removing skewness using yeo-johnson method



Checking the Distribution of the dataset



**Feature Scalling**



**MultiCollinearity with Variance Inflation Factor**

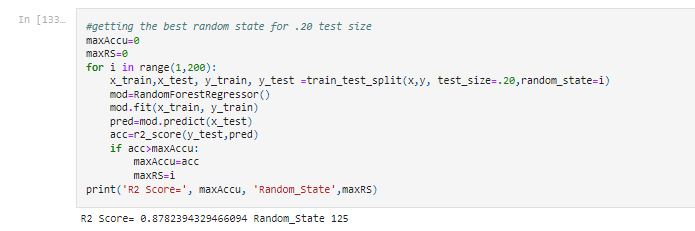


**Finding best random state**

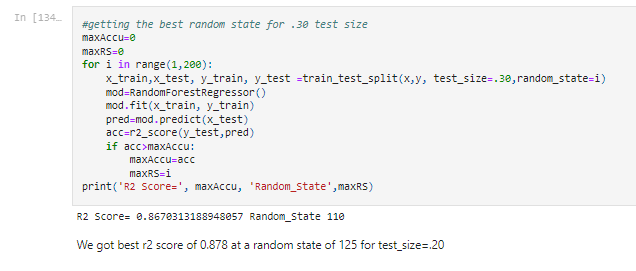
let’s find the best random state in which we can build the model*.*

(Random state ensures that the splits that you generate are reproducible. Scikit-learn use random permutations to generate the splits. The random state that you provide is used as a seed to the random number generator. This ensures that the random numbers are generated in the same order.)

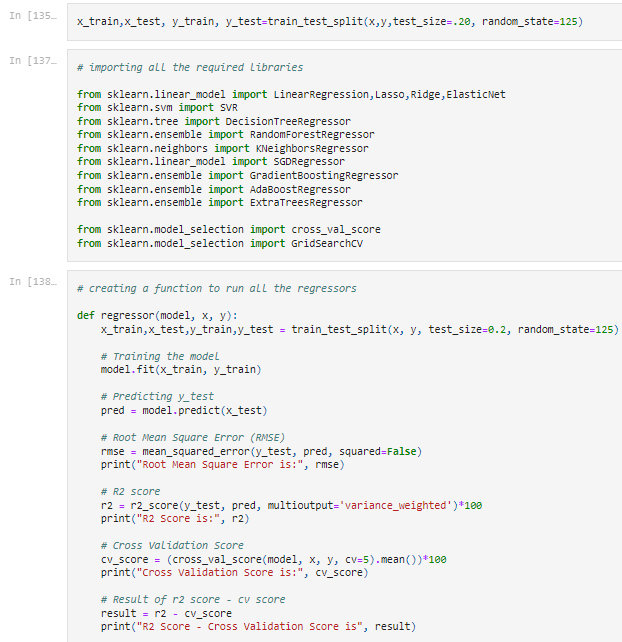
For Test size of 0.20



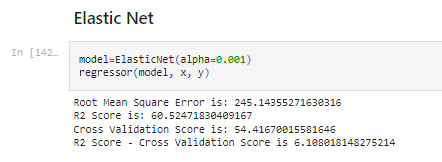
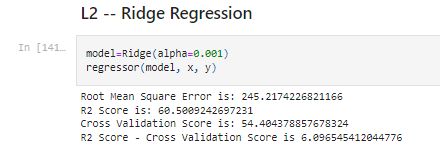
For Test size of 0.20

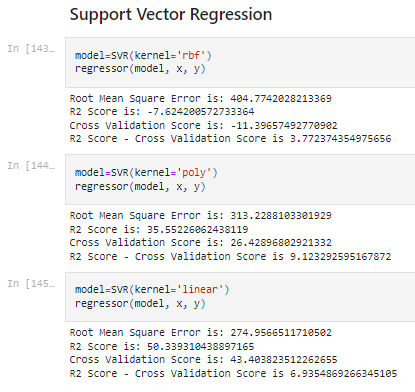
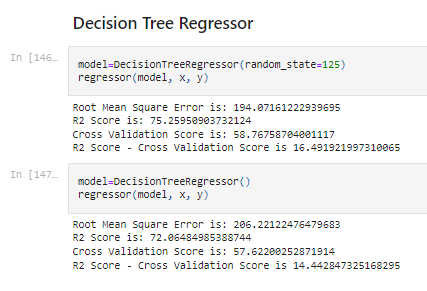


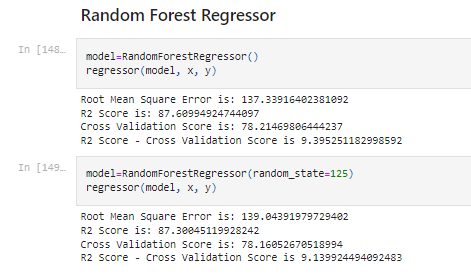
**Splitting the dataset into Features and Target**

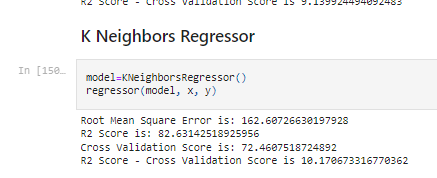


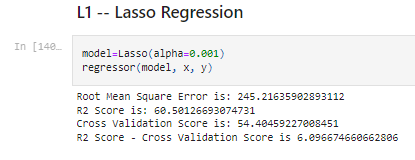
**Machine Learning Algorithm**

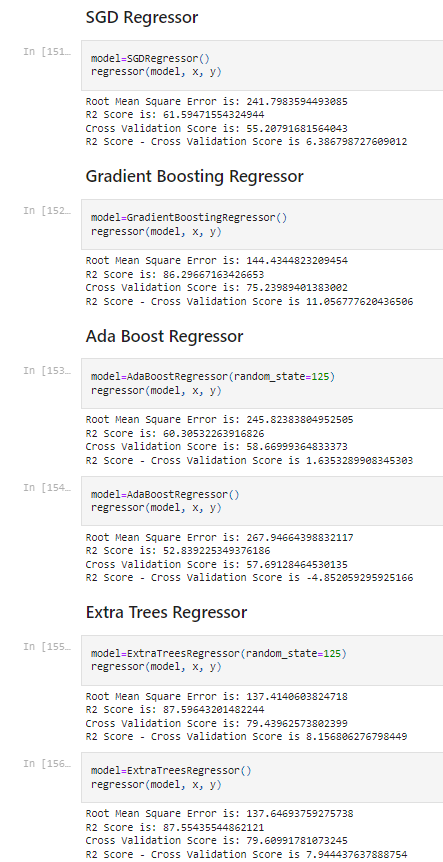




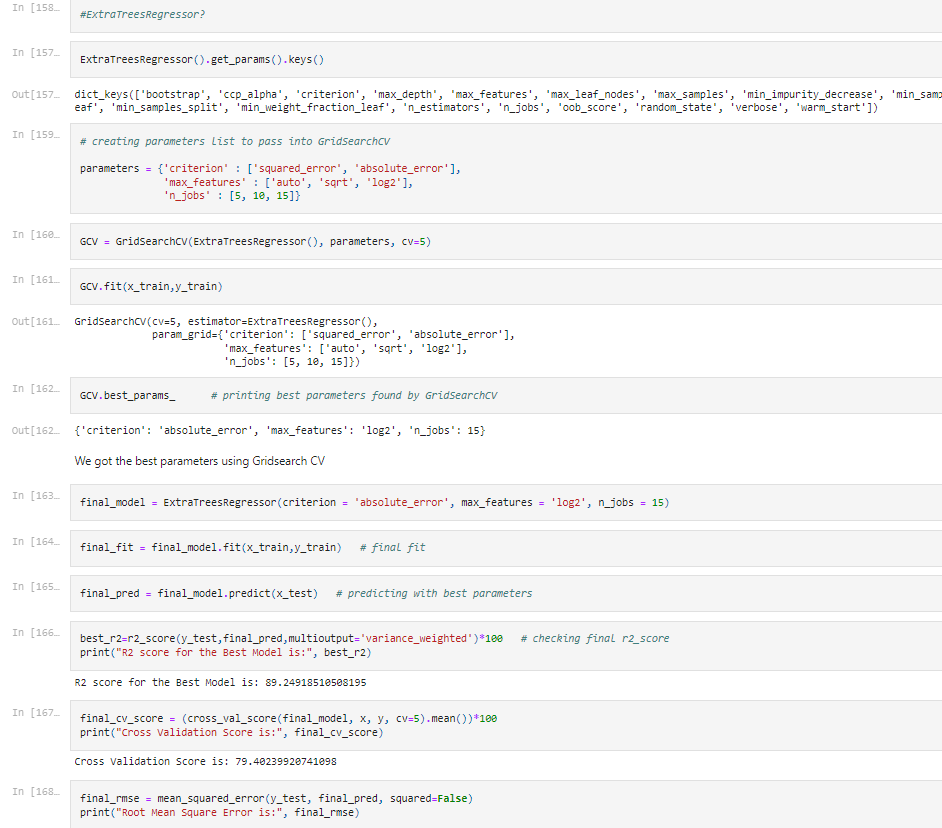


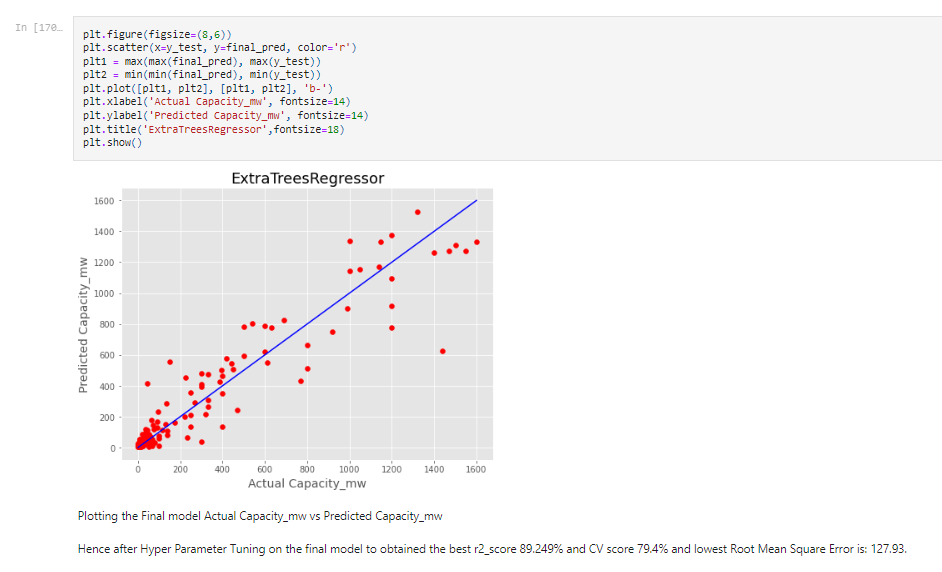




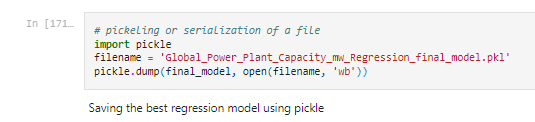


**Hyper parameter tuning**

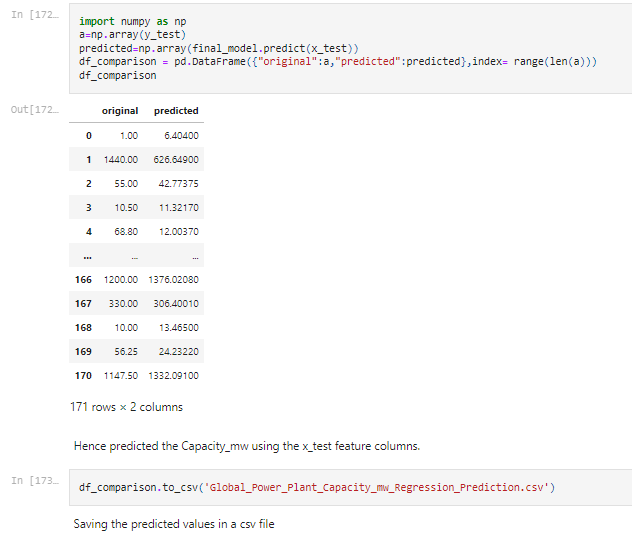




**Saving the model in pickle Format**

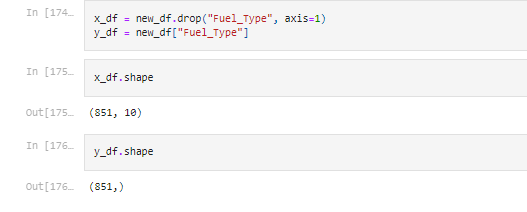


**Prediction Conclusion:**

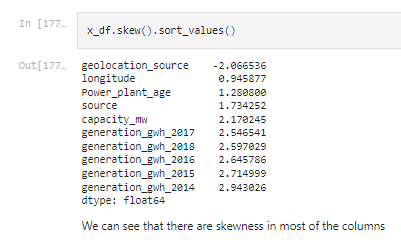


**5.2. Predicting "Fuel\_Type" Target**

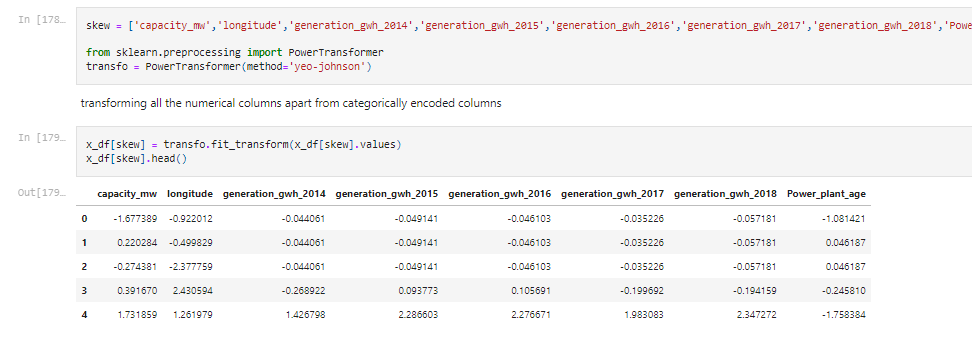
**Seperating the Dataset into Features and Label(Fuel\_Type)**

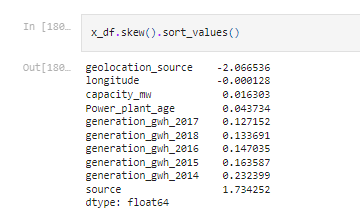


**Checking the skewness**



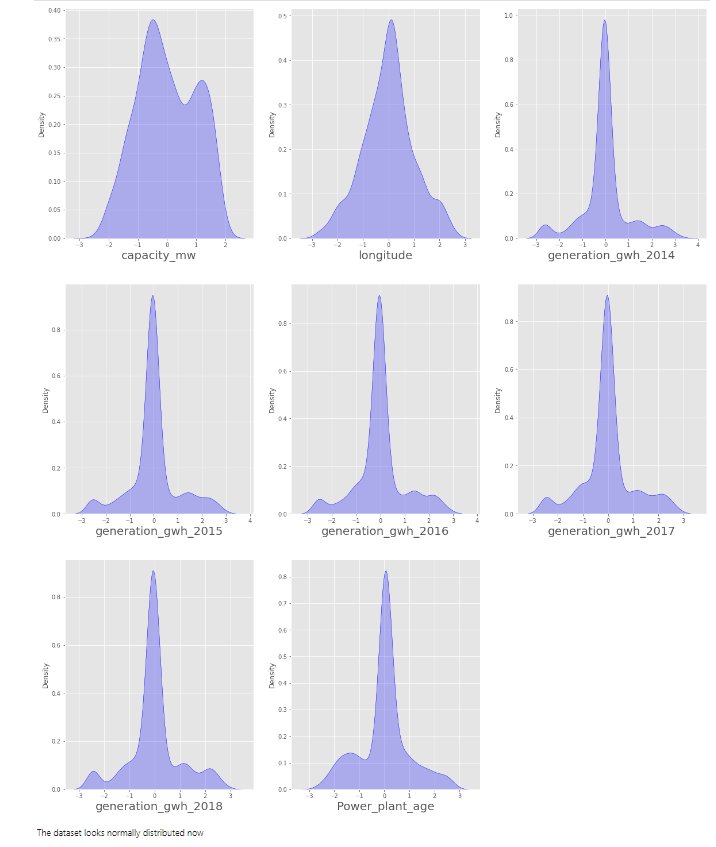
**Removing the skewness**



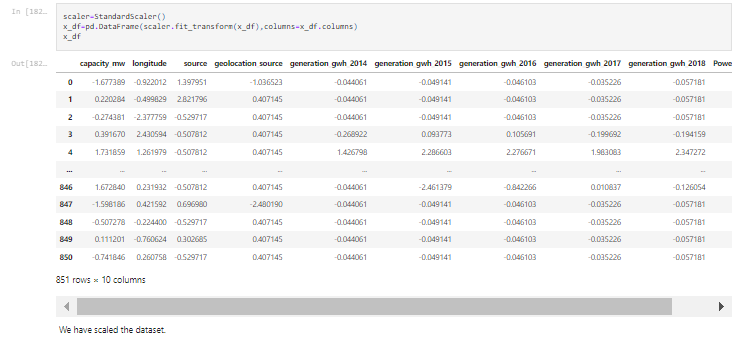


**Distribustion of all the feature**





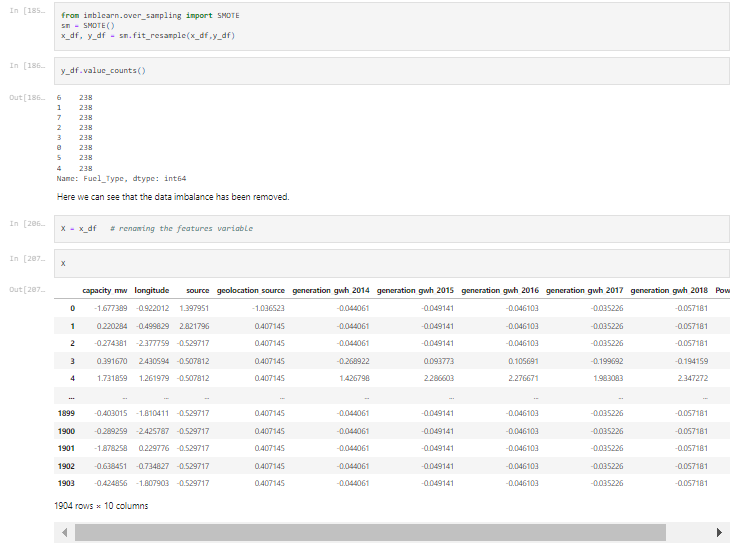
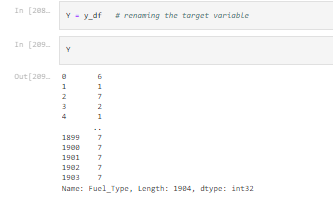
**Feature Scaling**



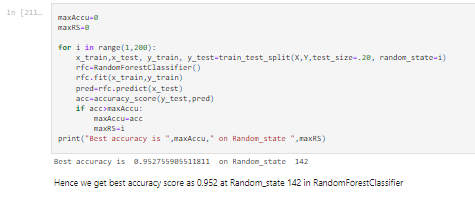
**Checking Multicolinearity**



**SMOTE OverSampling**

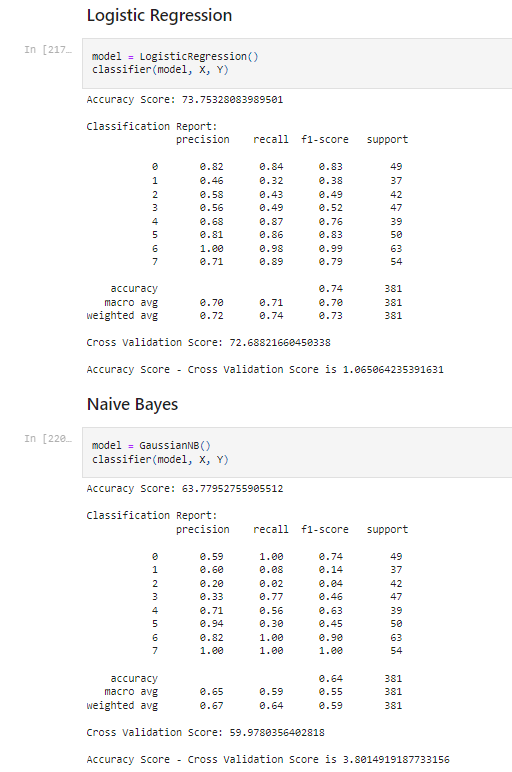
Finding best random state

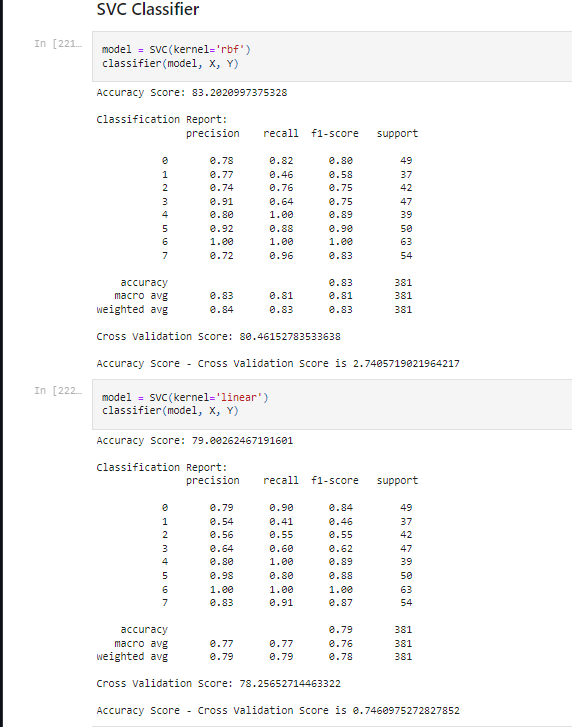


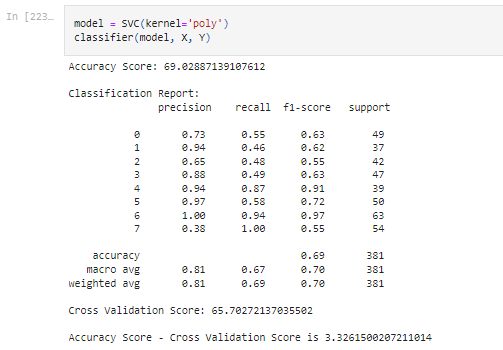
**train\_test\_split**

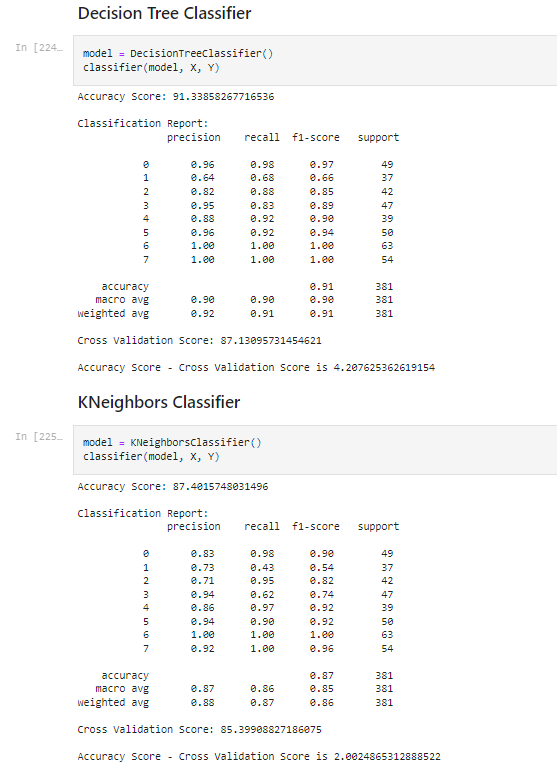


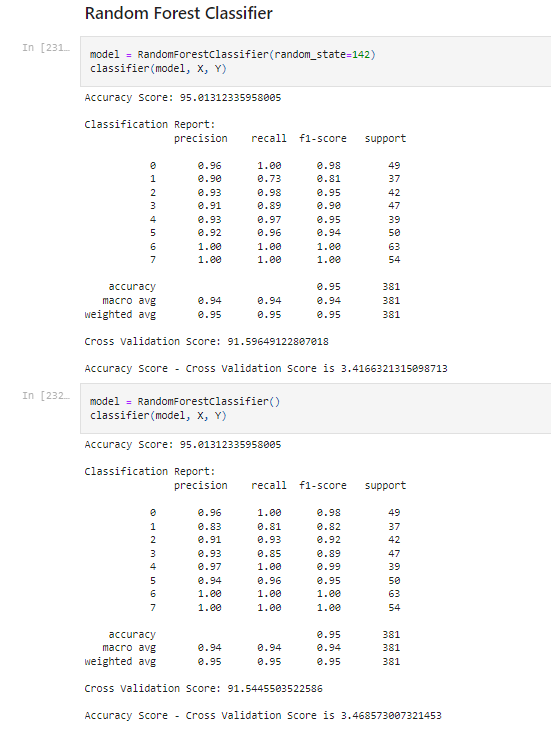
Machine Learning Algorithm

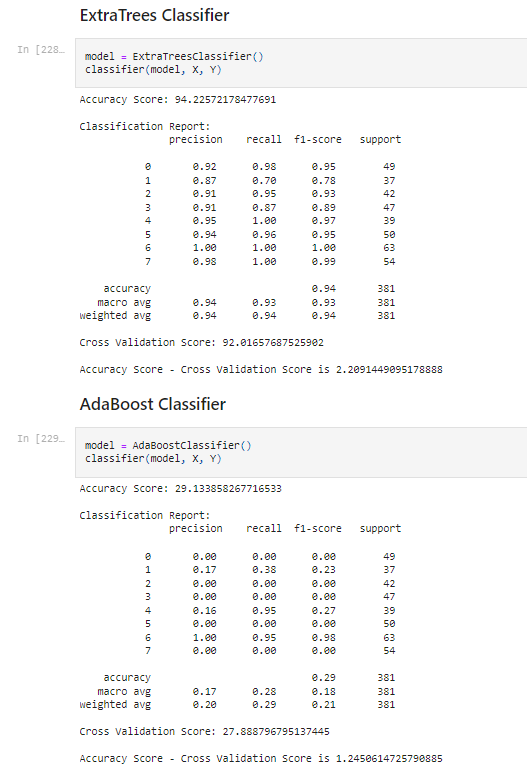


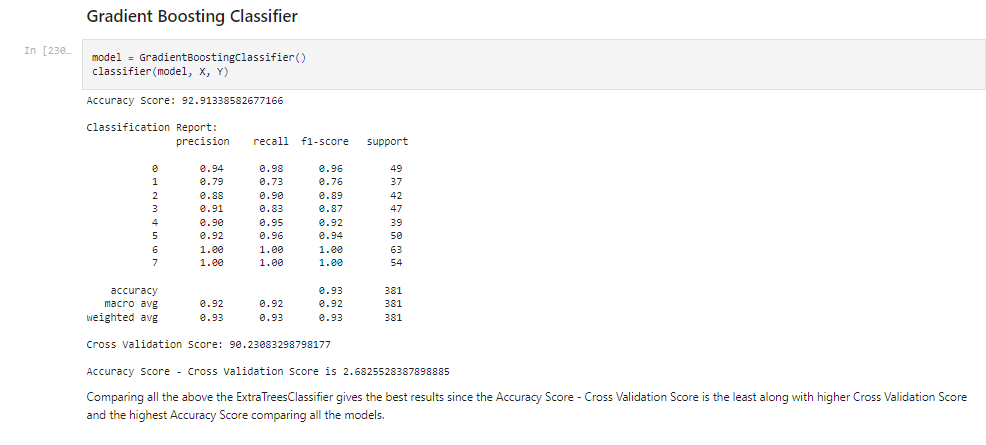




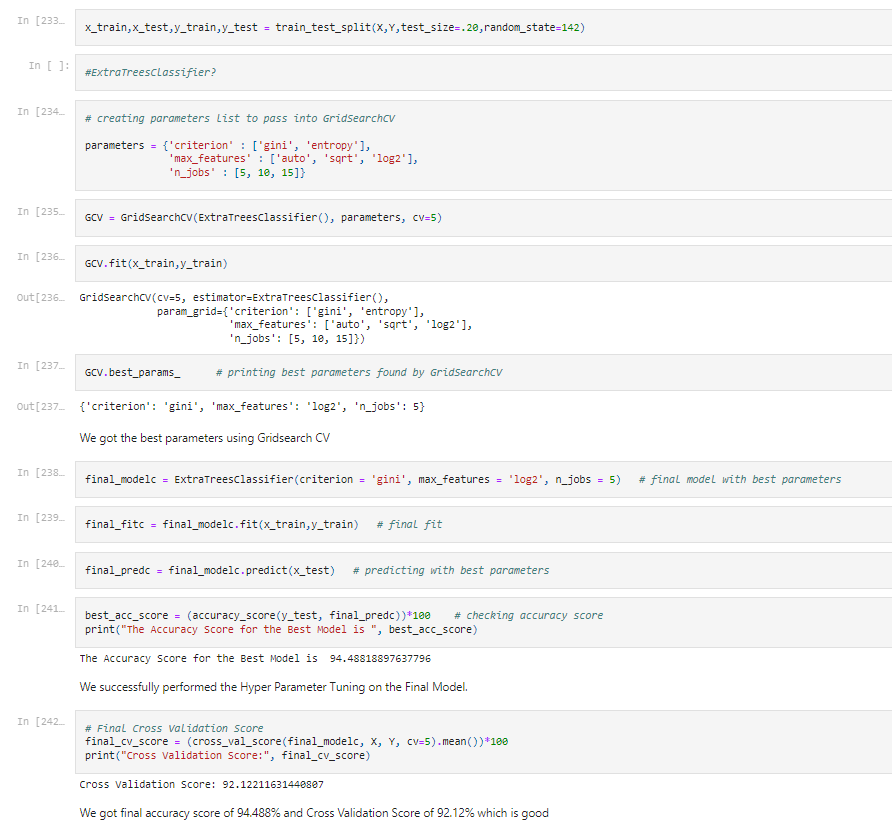






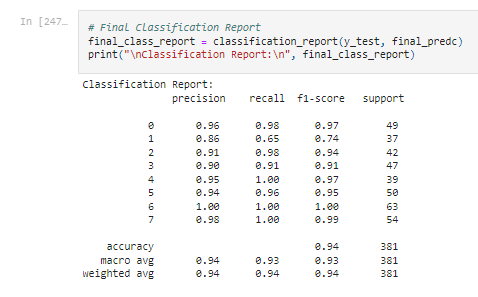


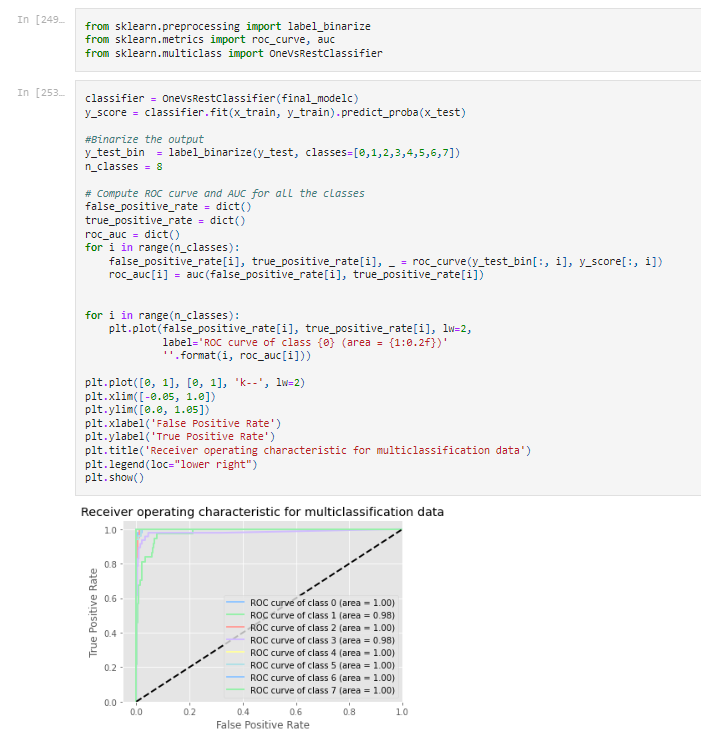
Hyper Parameter Tuning



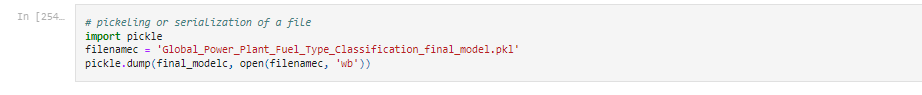


Cross validation Score





Saving the model in pickle Format



Prediction Conclusion:



------------------------- Thank You------------------------