**Global Power Plant Database**

**Problem Statement:**

**Description**

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. It will be continuously updated as data becomes available.

**Key attributes of the database**

The database includes the following indicators:

* `country` (text): 3 character country code corresponding to the ISO 3166-1 alpha-3 specification [5]
* `country\_long` (text): longer form of the country designation
* `name` (text): name or title of the power plant, generally in Romanized form
* `gppd\_idnr` (text): 10 or 12 character identifier for the power plant
* `capacity\_mw` (number): electrical generating capacity in megawatts
* `latitude` (number): geolocation in decimal degrees; WGS84 (EPSG:4326)
* `longitude` (number): geolocation in decimal degrees; WGS84 (EPSG:4326)
* `primary\_fuel` (text): energy source used in primary electricity generation or export
* `other\_fuel1` (text): energy source used in electricity generation or export
* `other\_fuel2` (text): energy source used in electricity generation or export
* `other\_fuel3` (text): energy source used in electricity generation or export
* `commissioning\_year` (number): year of plant operation, weighted by unit-capacity when data is available
* `owner` (text): majority shareholder of the power plant, generally in Romanized form
* `source` (text): entity reporting the data; could be an organization, report, or document, generally in Romanized form
* `url` (text): web document corresponding to the `source` field
* `geolocation\_source` (text): attribution for geolocation information
* `wepp\_id` (text): a reference to a unique plant identifier in the widely-used PLATTS-WEPP database.
* `year\_of\_capacity\_data` (number): year the capacity information was reported
* `generation\_gwh\_2013` (number): electricity generation in gigawatt-hours reported for the year 2013
* `generation\_gwh\_2014` (number): electricity generation in gigawatt-hours reported for the year 2014
* `generation\_gwh\_2015` (number): electricity generation in gigawatt-hours reported for the year 2015
* `generation\_gwh\_2016` (number): electricity generation in gigawatt-hours reported for the year 2016
* `generation\_gwh\_2017` (number): electricity generation in gigawatt-hours reported for the year 2017
* `generation\_gwh\_2018` (number): electricity generation in gigawatt-hours reported for the year 2018
* `generation\_gwh\_2019` (number): electricity generation in gigawatt-hours reported for the year 2019
* `generation\_data\_source` (text): attribution for the reported generation information
* `estimated\_generation\_gwh\_2013` (number): estimated electricity generation in gigawatt-hours for the year 2013
* `estimated\_generation\_gwh\_2014` (number): estimated electricity generation in gigawatt-hours for the year 2014
* `estimated\_generation\_gwh\_2015` (number): estimated electricity generation in gigawatt-hours for the year 2015
* `estimated\_generation\_gwh\_2016` (number): estimated electricity generation in gigawatt-hours for the year 2016
* `estimated\_generation\_gwh\_2017` (number): estimated electricity generation in gigawatt-hours for the year 2017
* 'estimated\_generation\_note\_2013` (text): label of the model/method used to estimate generation for the year 2013
* `estimated\_generation\_note\_2014` (text): label of the model/method used to estimate generation for the year 2014
* `estimated\_generation\_note\_2015` (text): label of the model/method used to estimate generation for the year 2015
* `estimated\_generation\_note\_2016` (text): label of the model/method used to estimate generation for the year 2016
* `estimated\_generation\_note\_2017` (text): label of the model/method used to estimate generation for the year 2017

**Fuel Type Aggregation**

We define the "Fuel Type" attribute of our database based on common fuel categories.

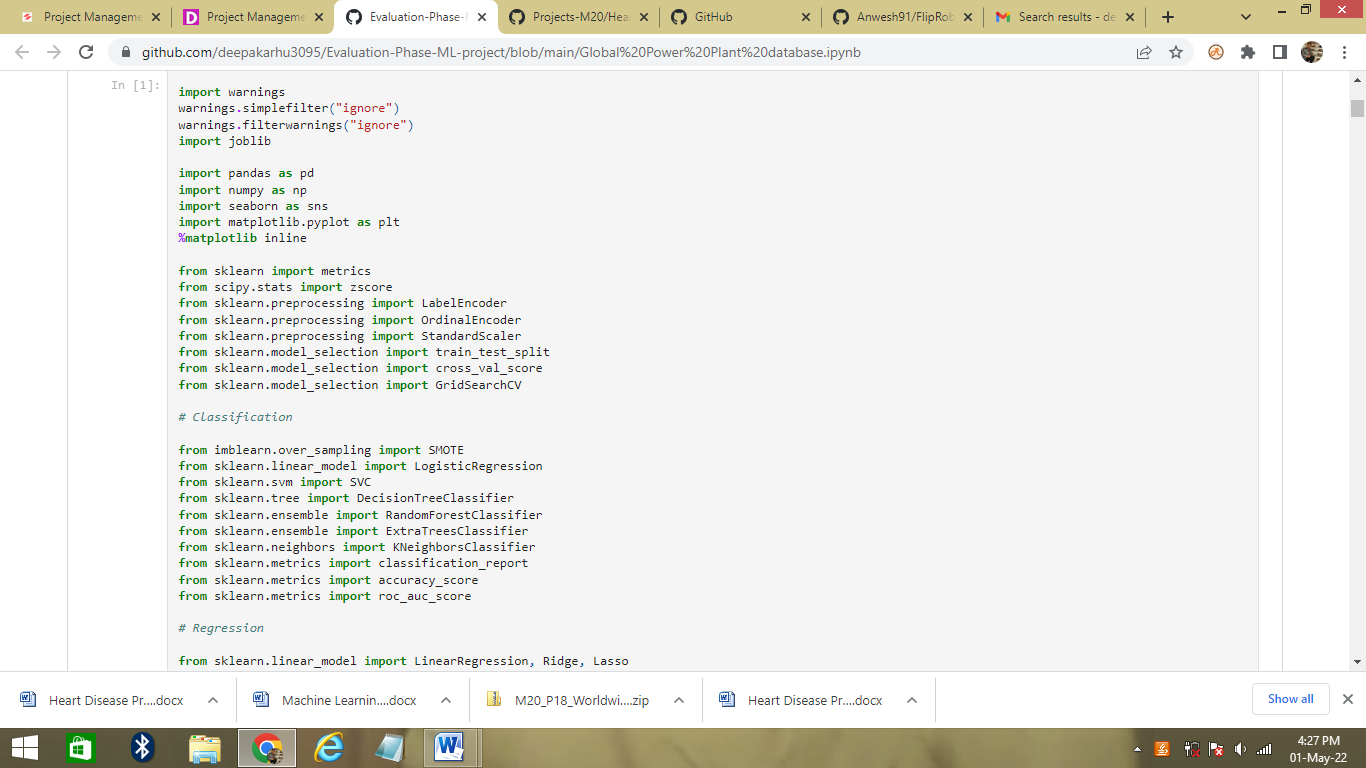
**Prediction :**   Make two prediction  1) **Primary** **Fuel**   2)**capacity\_mw**

**Find the dataset link below.**

Downlaod Files:

* <https://github.com/wri/global-power-plant-database/blob/master/source_databases_csv/database_IND.csv>

**Importing Warning and important all Library:**

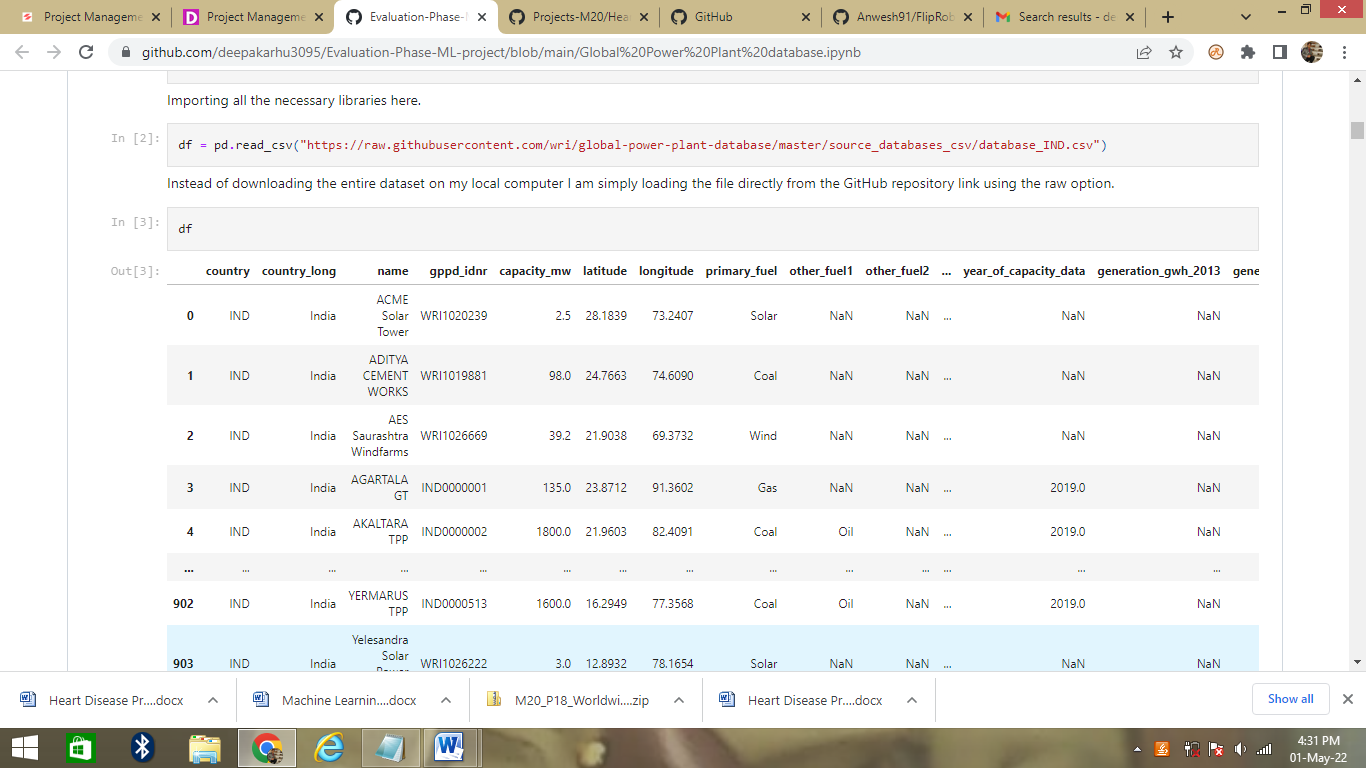


Instead of downloading the entire dataset on my local computer I am simply loading the file directly from the GitHub repository link using the raw option. And Checking the first 5 and last 5 rows of our entire dataset. We can see that our dataset comprises of total 907 rows and 25 columns.

In our problem statement we have been asked to predict 2 labels primary\_fuel and capacity\_mw. When we take a look at the values present in the column primary\_fuel we see that there are categorical data in that column so when we consider it as our label it will be termed as a Classification problem!

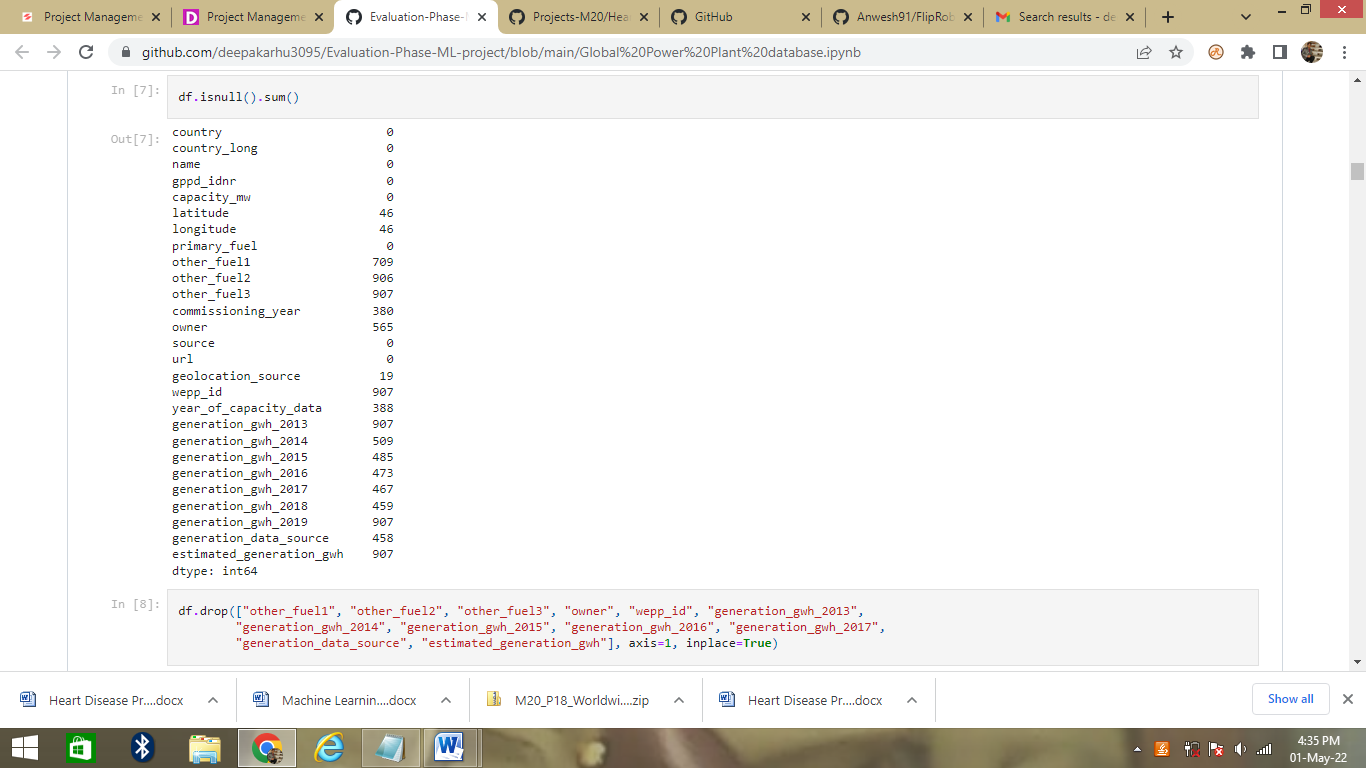
Similarly, if we take a look at the values present in the column capacity\_mw we see that there are continous data in that column so when we consider it as our label it will be termed as a Regression problem!

I will choose to perform the analysis on our entire data set first then will process the information accordingly to bifurcate the inputs for a Classification model and a Regression model.

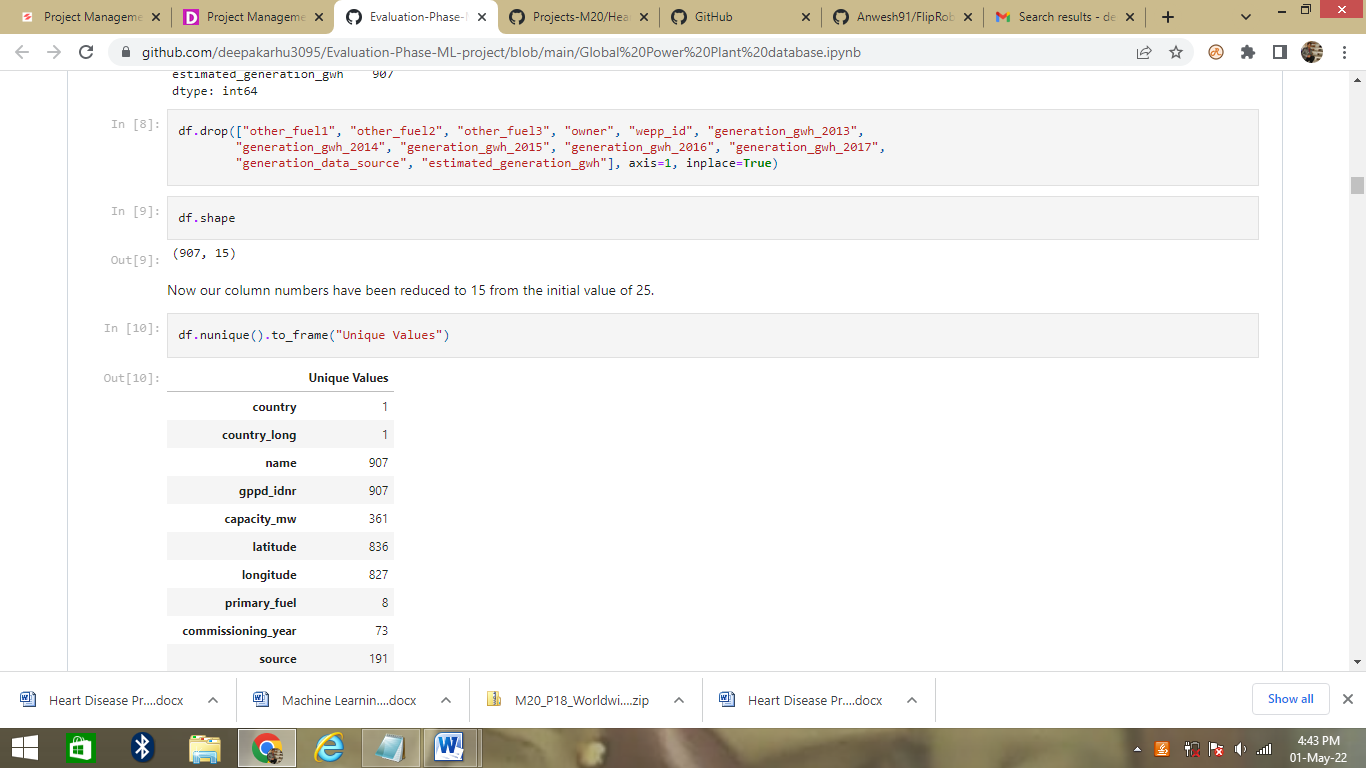


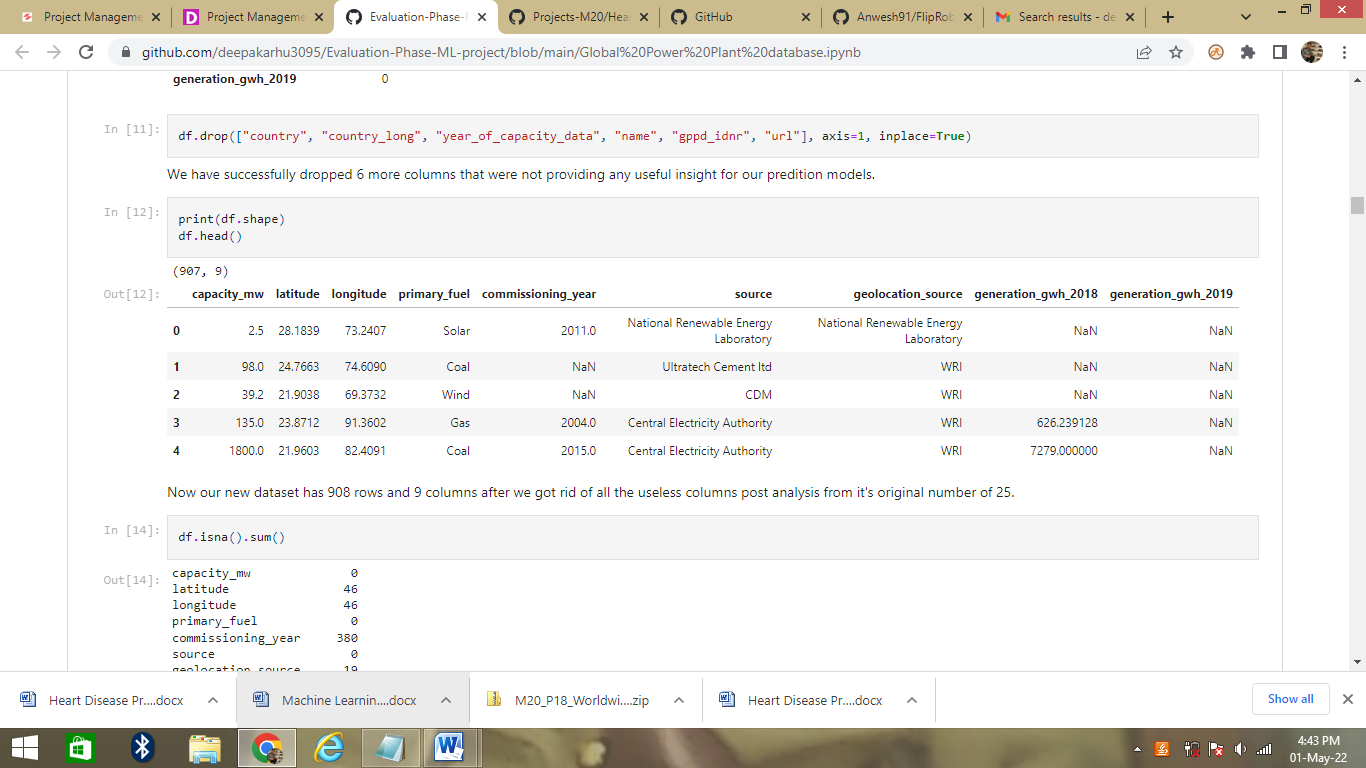
**Checking the columns and null values of the data**



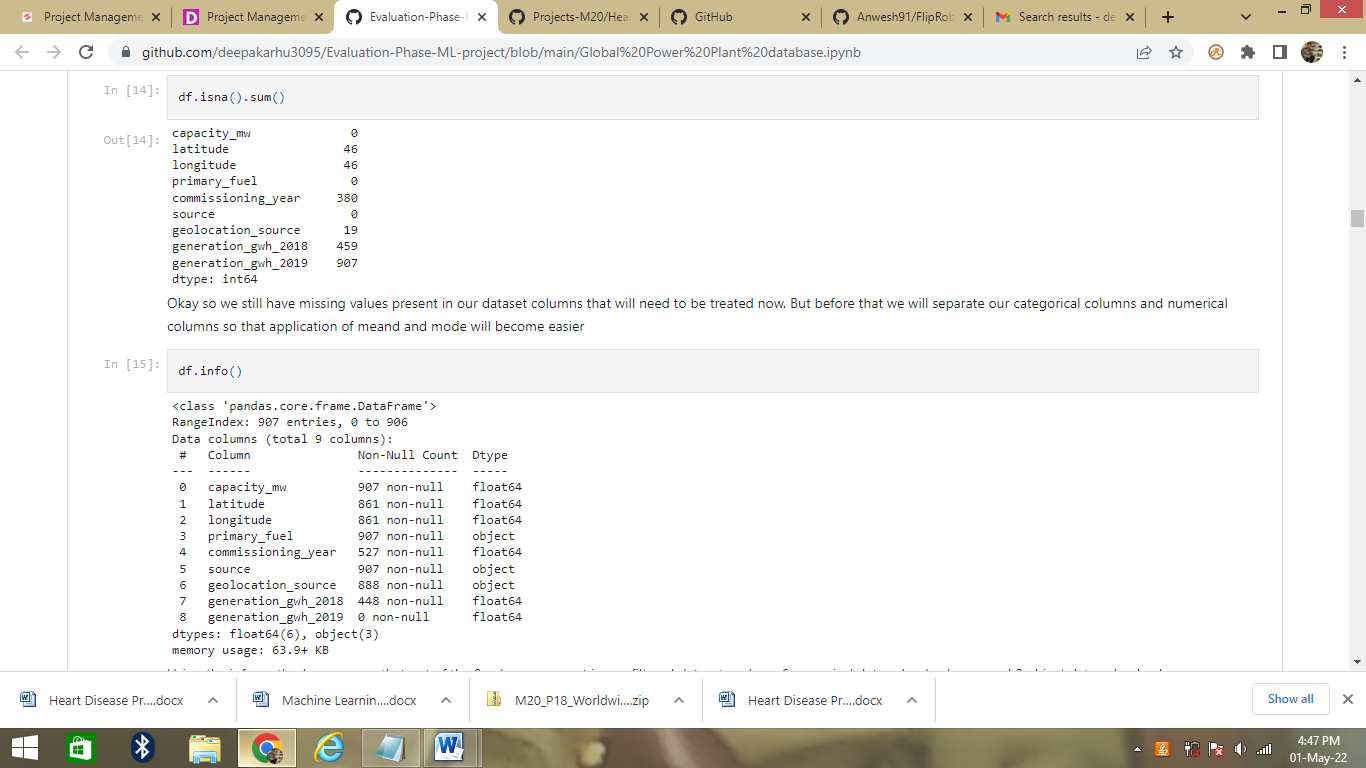


We can see that there are so any null values the data set. We need to fill the data and remove some unwanted data by drop method.

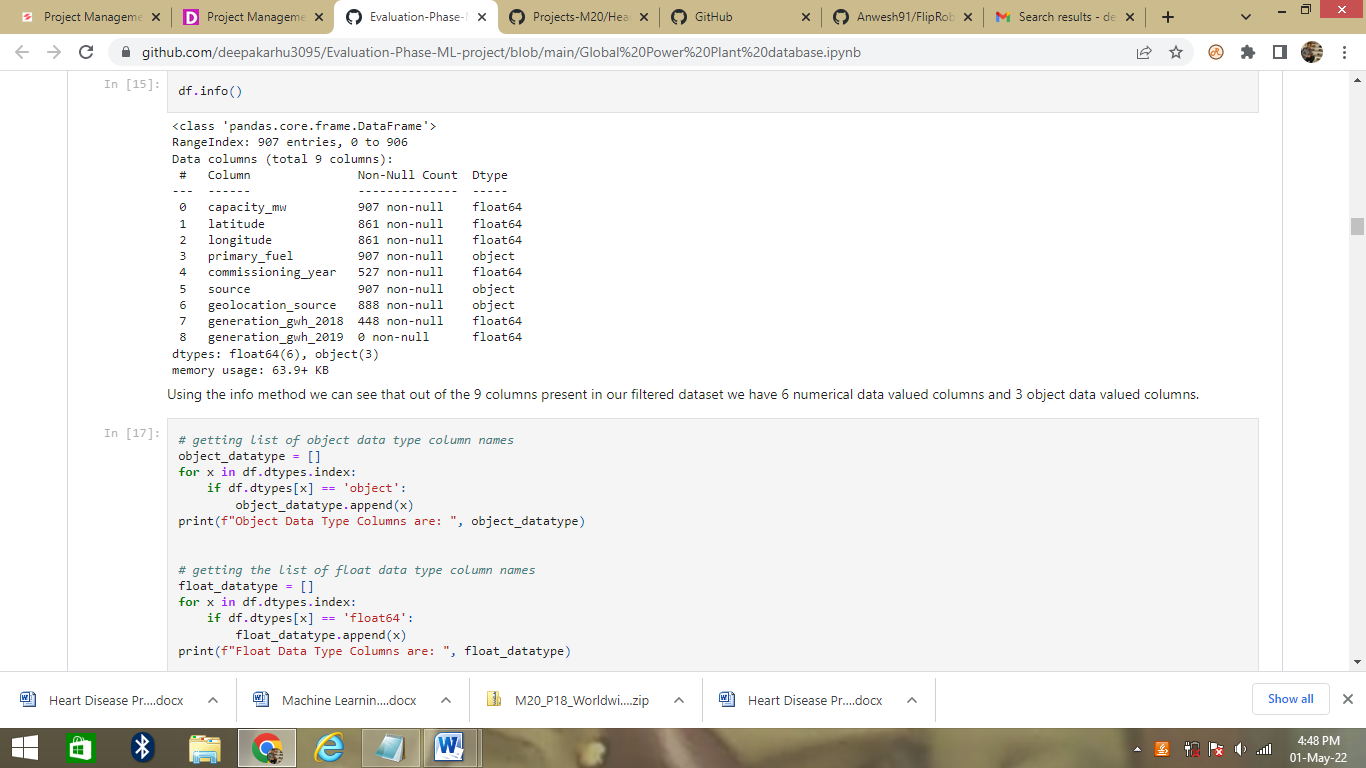




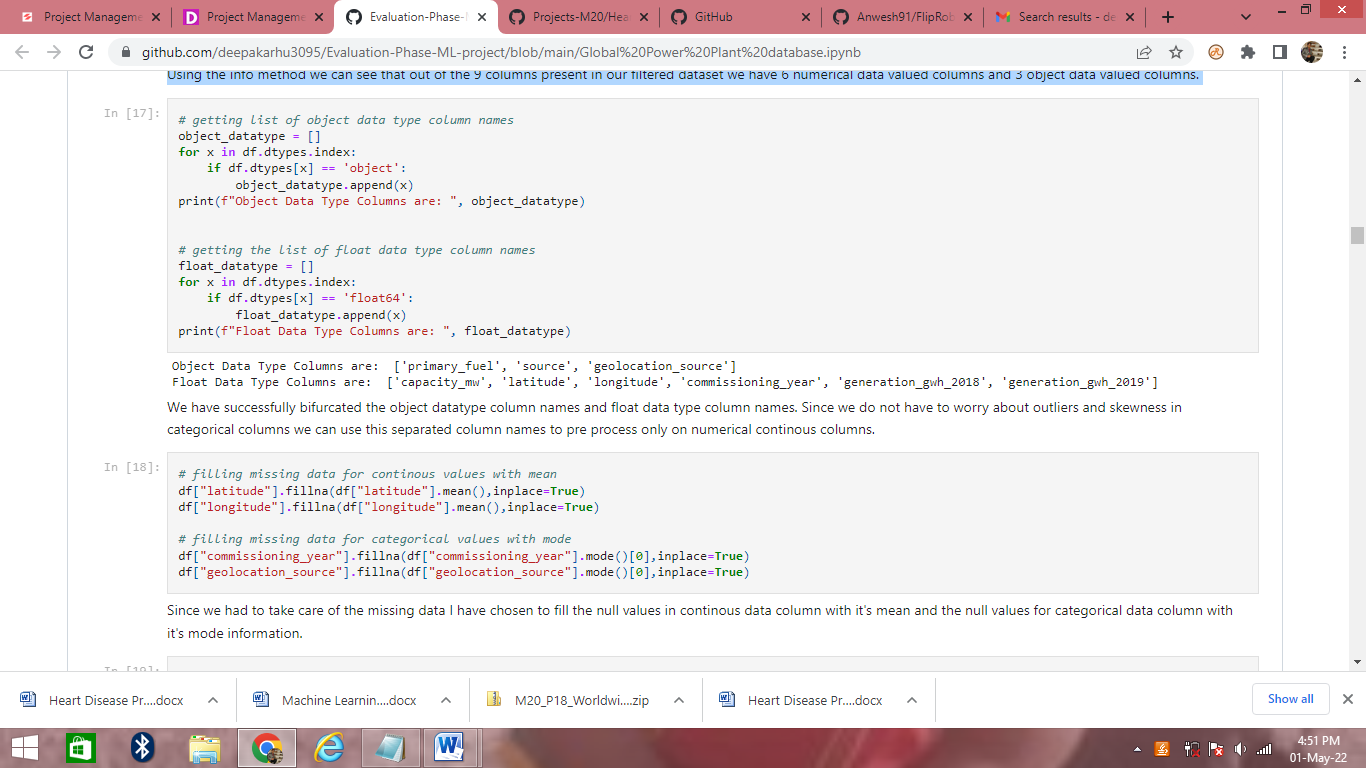
Now we drop the some columns. Now our new dataset has 908 rows and 9 columns after we got rid of all the useless columns post analysis from it's original number of 25.



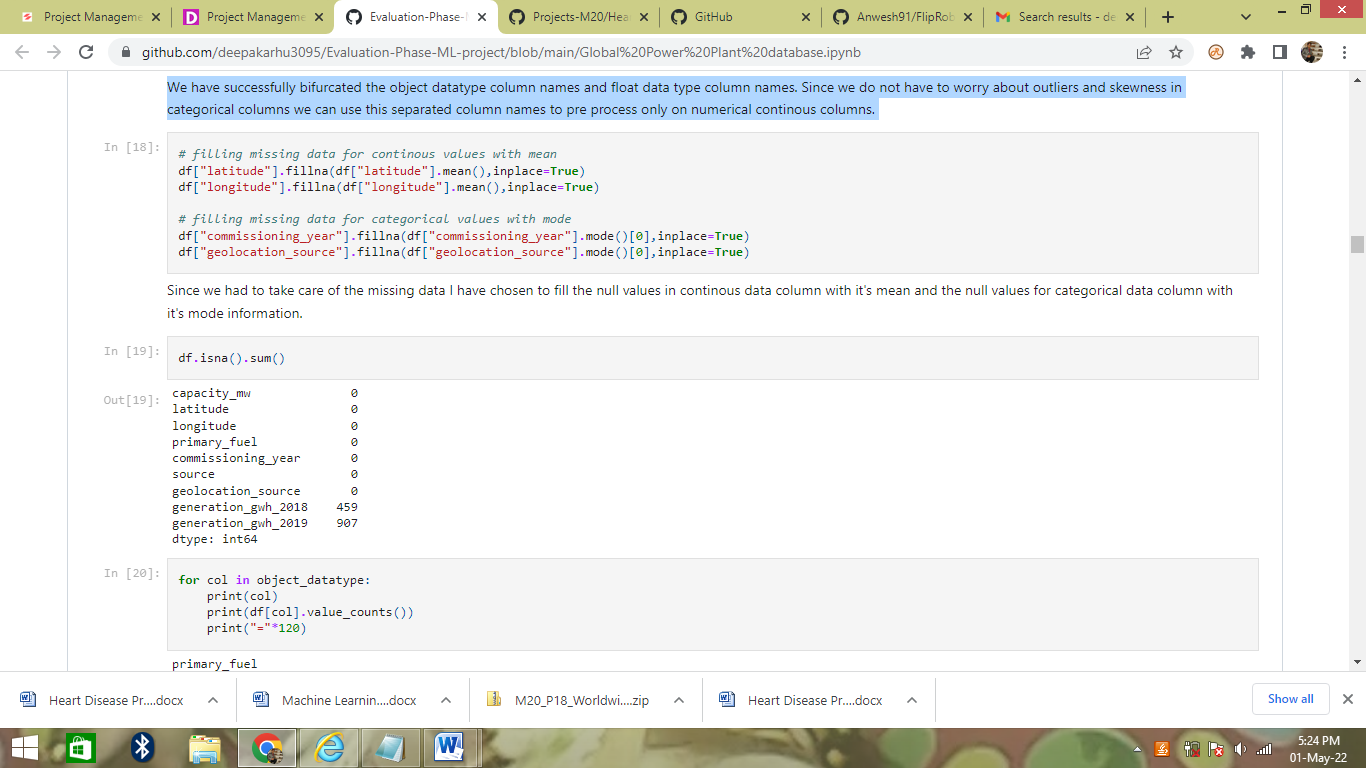
Okay so we still have missing values present in our dataset columns that will need to be treated now. But before that we will separate our categorical columns and numerical columns so that application of mean and mode will become easier



Using the info method we can see that out of the 9 columns present in our filtered dataset we have 6 numerical data valued columns and 3 object data valued columns.

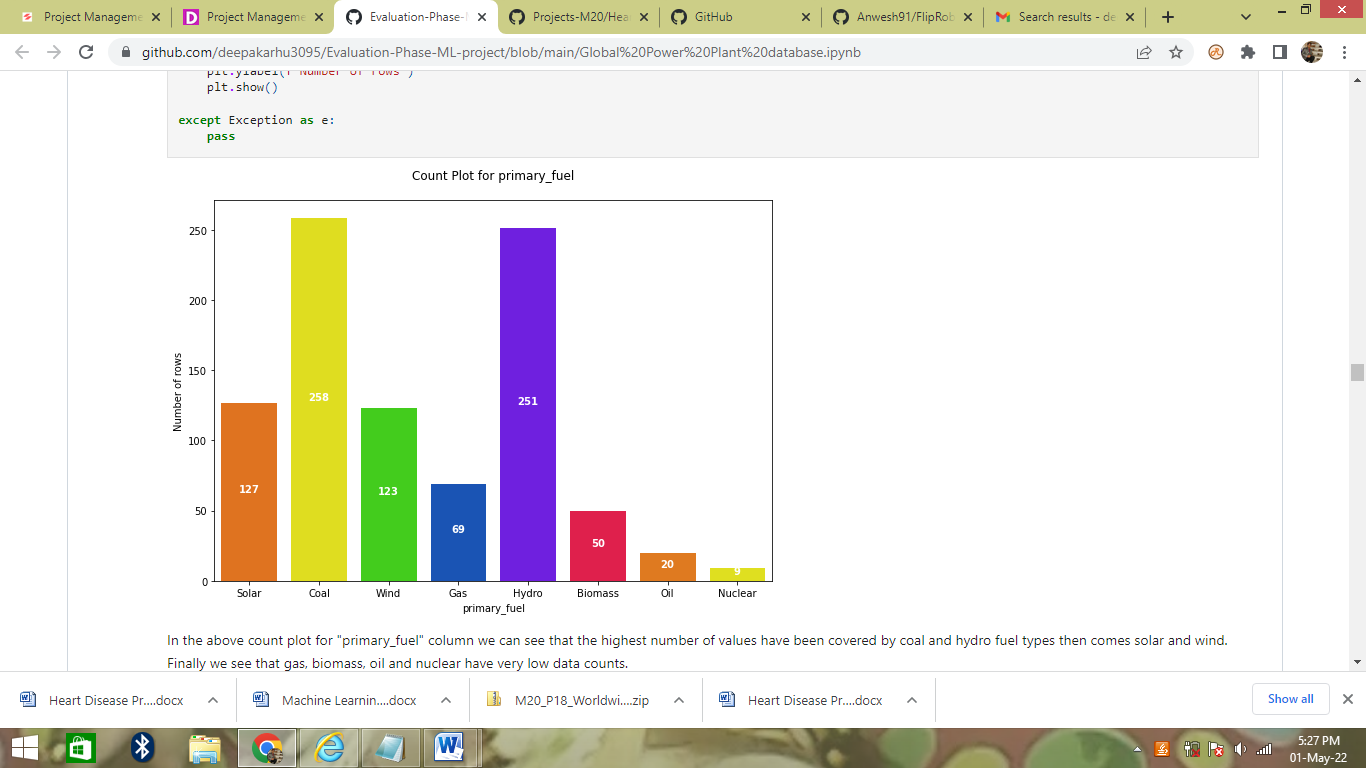


We have successfully bifurcated the object data type column names and float data type column names. Since we do not have to worry about outliers and skewness in categorical columns we can use this separated column names to pre process only on numerical continuous columns.



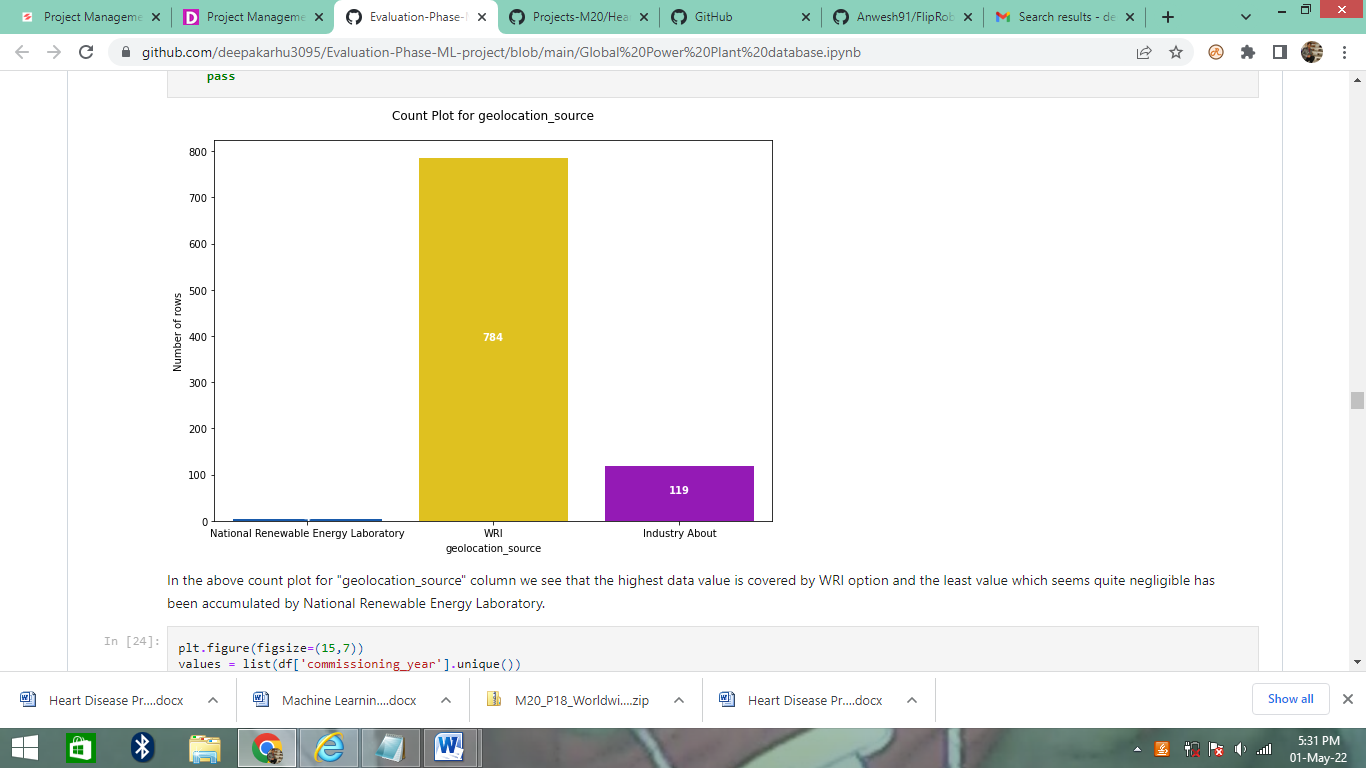
Since we had to take care of the missing data I have chosen to fill the null values in continous data column with it's mean and the null values for categorical data column with it's mode information.

**Now visualize the data**

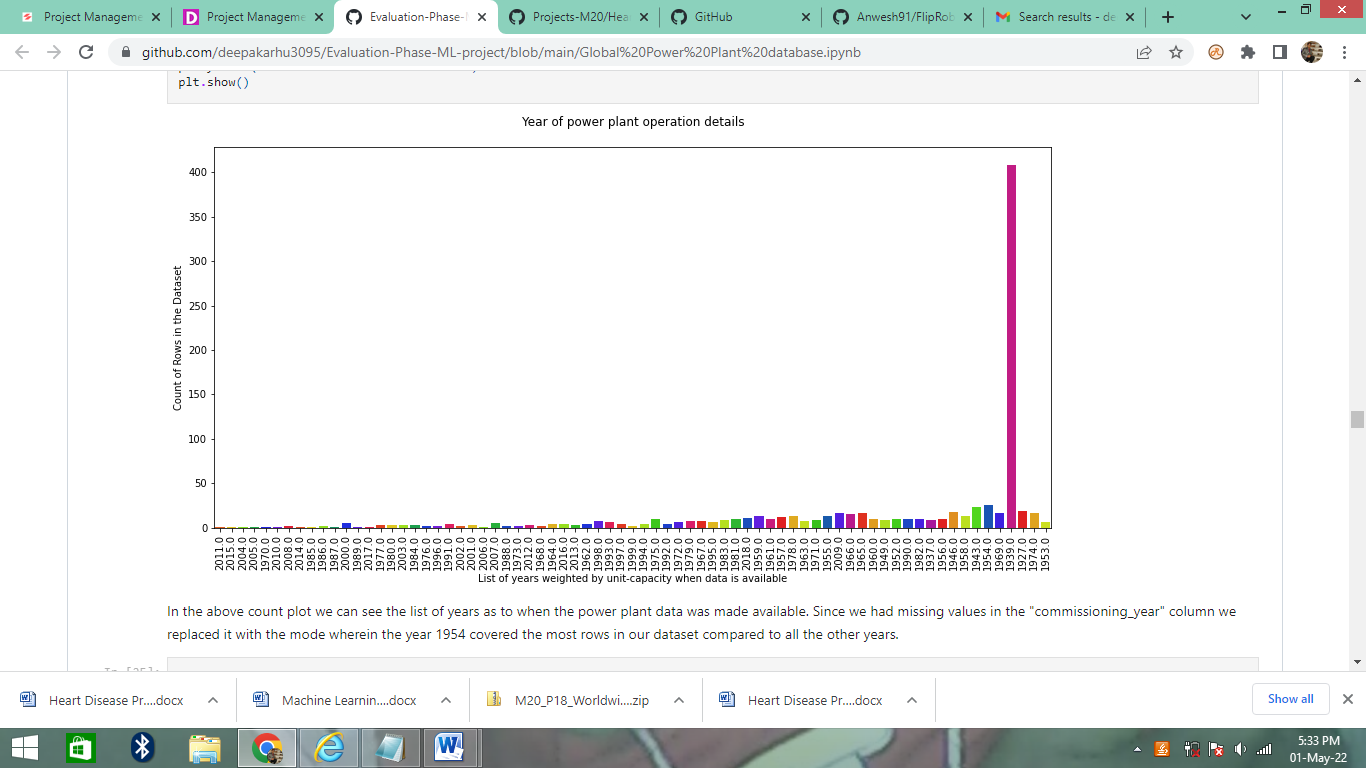


In the above count plot for "primary\_fuel" column we can see that the highest number of values have been covered by coal and hydro fuel types then comes solar and wind. Finally we see that gas, biomass, oil and nuclear have very low data counts.

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.



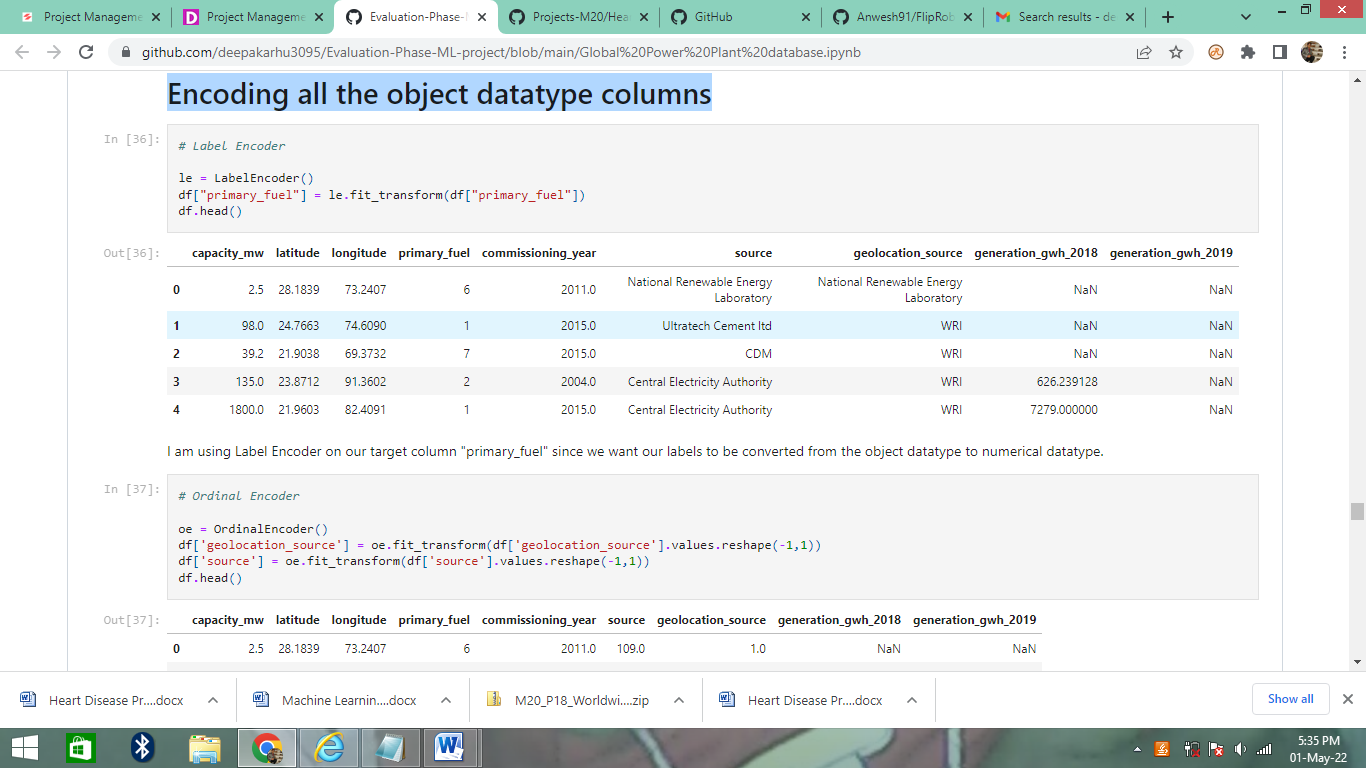
In the above count plot for "geolocation\_source" column we see that the highest data value is covered by WRI option and the least value which seems quite negligible has been accumulated by National Renewable Energy Laboratory.



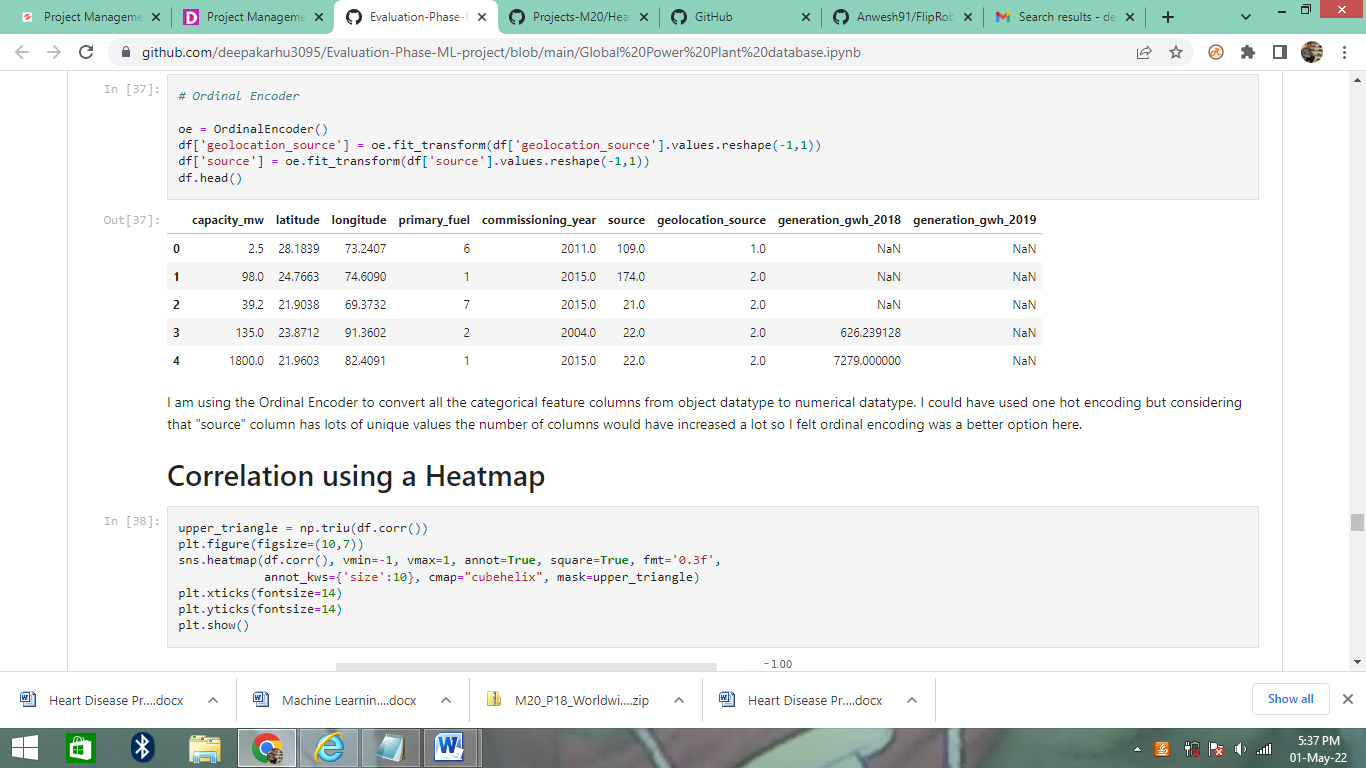
In the above count plot we can see the list of years as to when the power plant data was made available. Since we had missing values in the "commissioning\_year" column we replaced it with the mode wherein the year 1954 covered the most rows in our dataset compared to all the other years.

**Encoding all the object datatype columns**

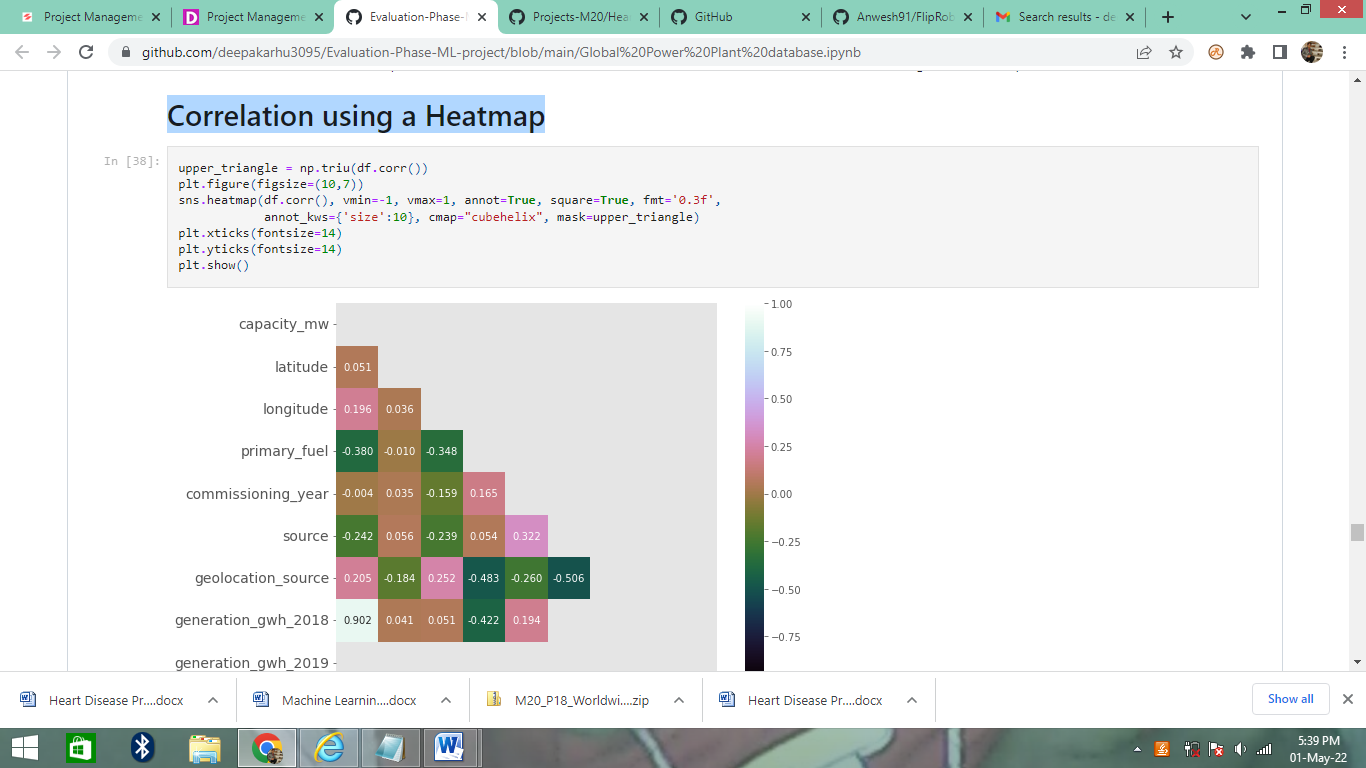
I am using Label Encoder on our target column "primary\_fuel" since we want our labels to be converted from the object datatype to numerical datatype.

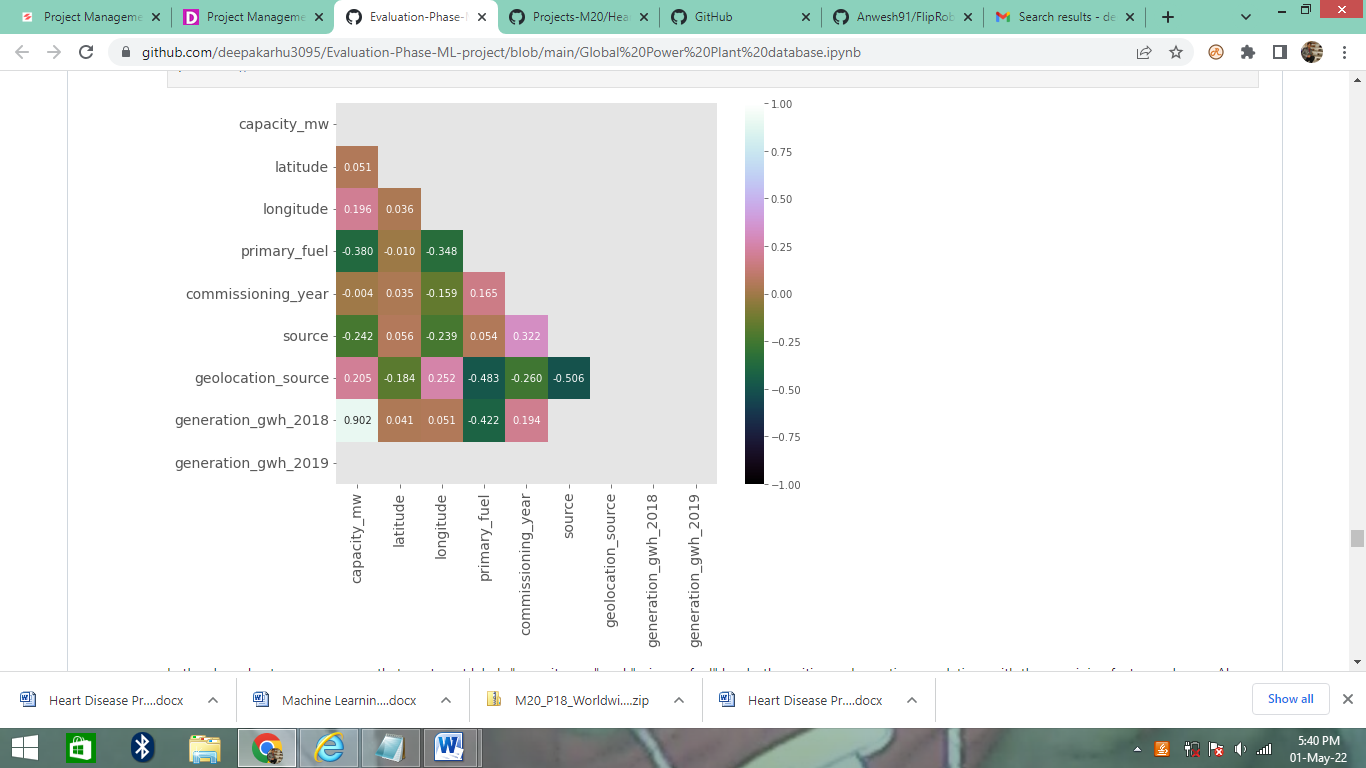


I am also using the Ordinal Encoder to convert all the categorical feature columns from object datatype to numerical datatype. I could have used one hot encoding but considering that "source" column has lots of unique values the number of columns would have increased a lot so I felt ordinal encoding was a better option here.



**Correlation using a Heatmap**





In the above heatmap we can see that our target labels "capacity\_mw" and "primary\_fuel" has both positive and negative correlations with the remaining feature columns. Also we see very less or negligible amount of multi colinearity so we will not have to worry about it. Since the one's which are reflecting the value are inter dependent on those feature columns and I intend to retain and keep them.

# Correlation Bar Plot comparing features with our labels