**Title**: Analyzing the Global Power Plant Dataset

* **Problem Definition:**

The Global Power Plant Database is an open-source resource that provides comprehensive data on power plants worldwide. It includes information on approximately 35,000 power plants from 167 countries, covering thermal plants like coal, gas, oil, nuclear, biomass, waste, geothermal and renewables like hydro, wind, solar. Each entry contains data on plant capacity, generation, ownership, and fuel type.

The goal of this project is to use this dataset to predict the total power capacity of power plants. Power plant capacity, the maximum electrical output a plant can produce, is a critical factor in energy planning and management, infrastructure development, and environmental policy.

By building a predictive model based on the features in the database, we aim to estimate a power plant’s total power capacity accurately. This prediction can help balance energy supply and demand, plan for peak times, ensure a stable energy supply, and make informed decisions about energy production and infrastructure development.

This project also contributes to the field of data science by demonstrating how machine learning techniques can be applied to real-world datasets to extract meaningful insights. It serves as a case study for data scientists interested in energy data or predictive modelling.

In conclusion, this project aims to predict the total power capacity of power plants using the Global Power Plant Database, contributing to efficient and sustainable management of global energy resources.

* **Data Analysis:**

Dataset includes data on approximately 35,000 power plants from 167 countries, covering a wide range of power generation types, including thermal plants like coal, gas, oil, nuclear, biomass, waste, geothermal etc and renewables like hydro, wind, solar etc.

The first step in our data analysis process is to understand the structure and content of the dataset. Each entry in the database represents a power plant and contains information on its capacity, generation, ownership, and fuel type. The capacity refers to the maximum electrical output the plant can produce. The generation refers to the amount of electricity the plant produces over a certain period. The ownership indicates who owns the plant, and the fuel type specifies the type of fuel the plant uses to generate electricity.

Next, we perform exploratory data analysis to uncover patterns and relationships in the data. We might look at the distribution of power plant capacities, the correlation between capacity and generation, or the prevalence of different fuel types.

In addition to descriptive statistics, we can use data visualization techniques to better understand our data. For example, we might create a histogram of power plant capacities, a scatter plot of capacity versus generation, or Boxplots for better understanding the range of the data or heatmap to show the relationship in better manner.

Finally, we prepare our data for predictive modelling. This might involve cleaning the data, handling missing values, encoding categorical variables, and normalizing numerical variables. We might also create new features that could improve the performance of our predictive models.

In conclusion, our data analysis process involves understanding our dataset, exploring patterns and relationships, visualizing our data, and preparing our data for predictive modelling. Through this process, we aim to gain insights that will help us predict the total power capacity of power plants.

* **EDA Concluding Remarks:**

Exploratory Data Analysis (EDA) is a critical step in any data science project as it allows us to understand the underlying patterns, spot anomalies, test hypotheses, and check assumptions related to our dataset. In the context of the Global Power Plant Dataset, our EDA has yielded several key insights.

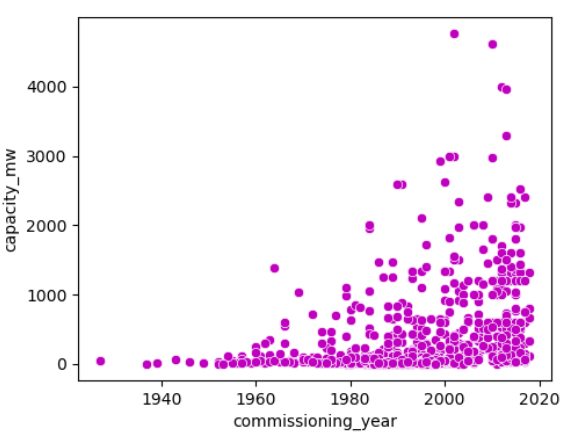
So if I talk about our dataset, We have total 907 entries and 27 columns. So that’s mean we have huge no of columns. Our main purpose of EDA is to build an efficient model by using less no of columns because more no of columns can lead to overfitting. So out of 27 columns we will check the relationship of the dependent variable with the independent variables. For that we have lots of techniques that we can use to find out the relationships between the columns, We can use various types of plots like Scatter plots, Reg plots, Lm plots and so many plots we have. After checking the relationship if we see that some of the columns are not giving that much importance to the target variable then we can simply remove these columns. Second process is to use statistical analysis like we can check the correlation between the columns by using corr() method as well as if there are some columns those are highly correlated with each other then we can remove these columns because it can create multi collinearity issue. We can check this by using variance inflation factor method. We can also check the distribution of the data, like how the data’s are distributed over every columns. Is it normally distributed, skewed distributed or any other types of distributions is there. Data follows a normal distribution often simplifies analyses and makes the data easier to work with. This assumption is prevalent in many statistical techniques and machine learning models for several reasons:

1. **Central Limit Theorem:** This theorem states that the sum of a large number of independent and identically distributed variables, regardless of their shape, approaches a normal distribution. This makes the normal distribution a natural choice for many analyses.
2. **Simplicity:** The normal distribution is defined by two parameters: the mean and the standard deviation. This simplicity makes it computationally efficient and easy to work with.
3. **Statistical Tests:** Many statistical tests (like t-tests, ANOVA, etc.) assume normality because their theoretical derivations are based on this assumption.
4. **Predictive Modelling:** Some predictive models (like linear regression) assume that the errors, or residuals, follow a normal distribution, which leads to the best linear unbiased estimates.

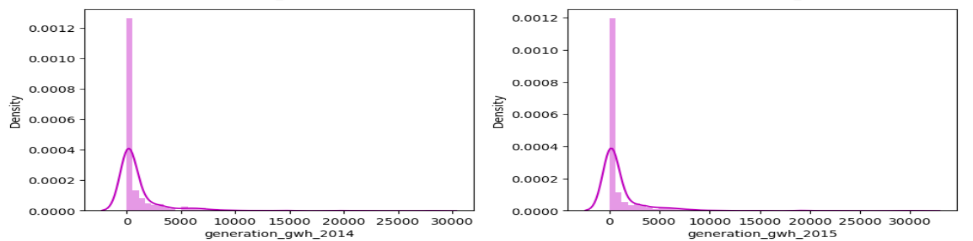
Let’s come to the analysis part of our dataset.

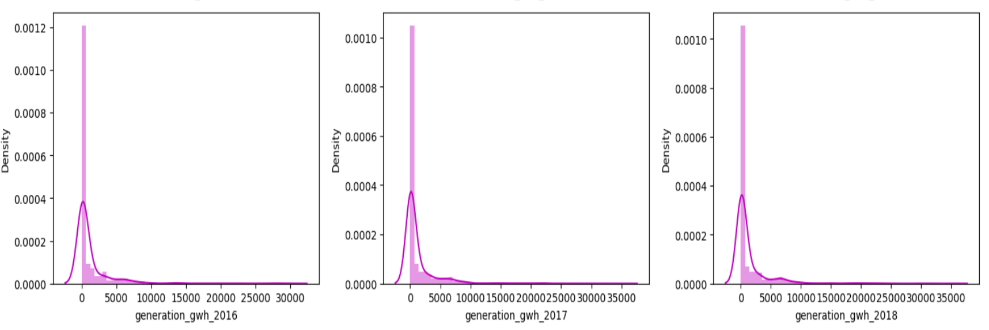
By the analysis, we can see that every powerplant uses different type of fuels like coal, oil etc. But when the primary fuel is coal then the capacity is maximum and for the other fuel1 when it is oil then also the capacity is maximum.

In the place of Geolocation source when it is in WRI then the capacity is maximum and capacity is linearly related with generation\_gwh(2014-2018). We can also see that for the latest commissioning year, capacity is continuously increasing.

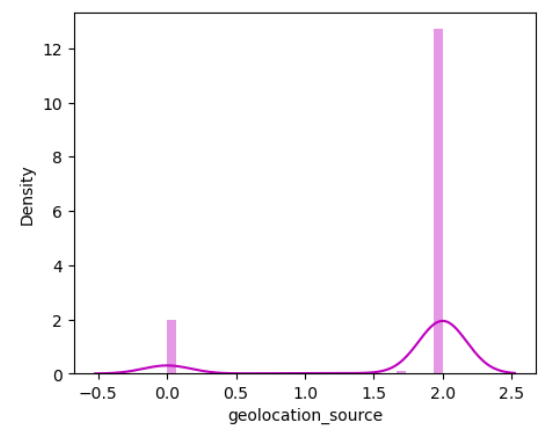


From the statistical summary we can see that ['capacity\_mw', 'longitude', 'other\_fuel1', 'commissioning\_year', 'generation\_gwh\_2014', 'generation\_gwh\_2015', 'generation\_gwh\_2016', 'generation\_gwh\_2017', 'generation\_gwh\_2018'] these columns has positive skewness because mean is greater than median.

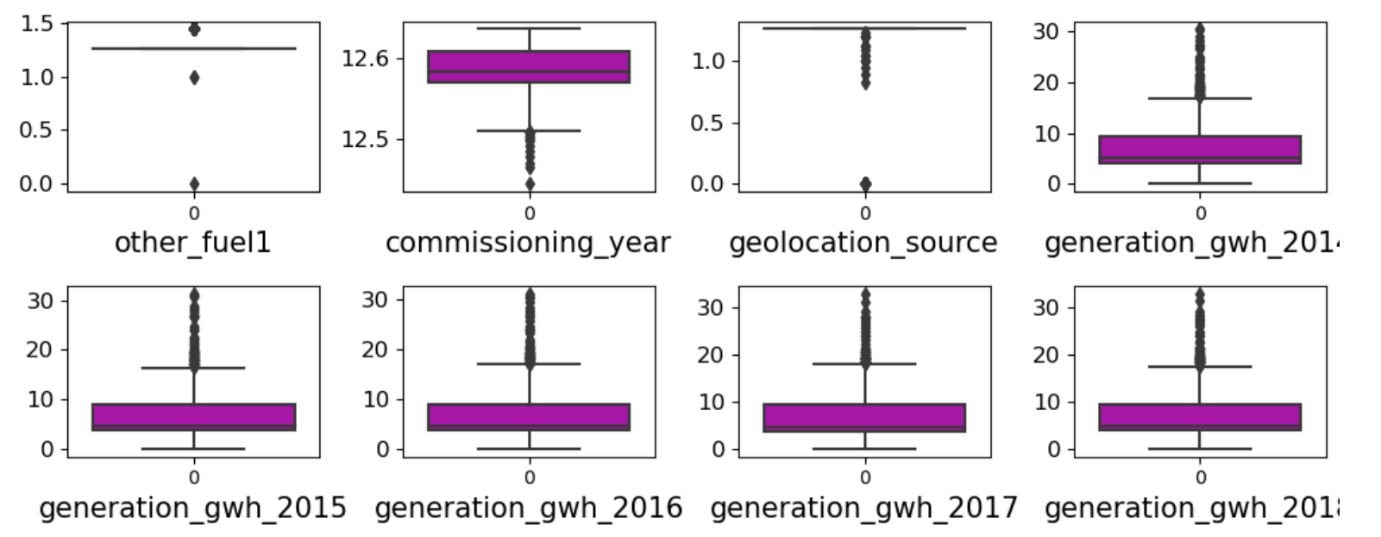




And geolocation\_source column has negative skewness because here mean is less than median.



By observing the box plot we can see that

  
all the generation\_gwh columns from the year 2014-2018 has outliers present and for the column geolocation\_source and commissioning\_year has very less outliers present.

So basically, outliers are data points that are significantly differs from other observations. An outlier may be due to variability in the data or it could indicate experimental error; the latter are sometimes excluded from the data set.

After checking the correlation by using corr() method and then creating heatmap from these correlation values we can observe that primary fuel is comparatively high negatively correlated with all the columns

And capacity\_mw column is highly positive correlated with all the columns

capacity\_mw 1.000000

generation\_gwh\_2017 0.844914

generation\_gwh\_2016 0.843928

generation\_gwh\_2018 0.831472

generation\_gwh\_2015 0.810897

generation\_gwh\_2014 0.796319

other\_fuel1 0.672030

commissioning\_year 0.476280

So from the above data we can see that capacity\_mw column has very high positive correlation with all the generation\_gwh columns. And also capacity\_mw column has good positive correlation with other\_fuel1 and commissioning\_year column.

Latitude column has very less positive correlation with the target variable (latitude - 0.05849).

If we check the multicollinearity with all the columns by using variance inflation method

We can see



longitude, other\_fuel1 and commissioning\_year column has very high VIF values.

So having very high VIF values, columns should be dropped but here these columns are very important for this dataset this we can say from our domain knowledge.

I will talk about the dropped columns in the next step I mean in the preprocessing pipeline.

So, after doing train\_test\_spilt we have now 11 columns in our x\_train dataset. When I use RandomForestRegressor we get the r2 score of 94.98% for the random\_state value 175.

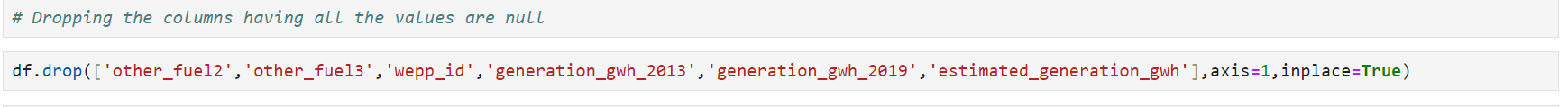
After that for the further reduction of the columns I use sequential feature selection method where I get 8 best columns, these columns are ('latitude', 'longitude', 'other\_fuel1', 'commissioning\_year', 'generation\_gwh\_2014', 'generation\_gwh\_2015', 'generation\_gwh\_2016', 'generation\_gwh\_2018') and when I apply RandomForestRegressor algorithm on this I get r2 score of around 89% that is lesser than the previous model. So I will not use that feature selection method.

* **Pre-processing Pipeline:**

Data pre-processing is a crucial step in any data science project. It involves preparing the raw data to make it suitable for a machine learning model. This process can significantly influence the outcome of the model as the quality of data and the useful information that can be derived from it directly affects the ability of the model to learn.

The first step in the pre-processing pipeline is **data cleaning**. This involves handling missing values and removing duplicates. Missing values can be filled with a central tendency measure like mean, median, or mode. Alternatively, if the dataset is large and the missing data is not significant, those rows can be dropped. Duplicates can skew the model’s understanding, so they are removed.

So after analysing our dataset we can see that we have lots of null values in almost every columns. Estimated\_generation\_gwh, wepp\_id, generation\_gwh\_2013, generation\_gwh\_2019, other\_fuel2 and other\_fuel\_3, These columns has all the values are null. So it will be better to drop these columns.



Apart from this, generation\_gwh(2014-2018) col has almost 50% null values. other\_fuel1 col has 709 null values out of 907 values.

We will use the Imputation techniques to fill the null values.

URL column has of no use because it has all the values are unique and if I use the encoding methods to convert the URL column then due to the high cardinality it may lead to overfitting.

So I will drop the URL column too.



Country\_long and country columns has all the values are same India and IND respectively so I can say that these columns are constant columns. In machine learning constant columns has of no use because it will not give any importance to the target variable. It will be better to drop these columns.



gppd\_idnr column has all the values are unique codes. So this particular column has of no importance to predict the capacity because machines will not learn anything from this column. Same for the name column. So it will be better to drop these columns .



Owner and source columns are individual columns basically these columns have no relation with any columns. So, I can say that these columns are useless so it is better to drop these columns.



year\_of\_capacity\_data column has all the values are 2019 and rest are the null values. So, if I impute this column then it will be a constant column that has of no use. So, I am dropping this column



generation\_data\_source column has all the values are 0. That’s mean it’s a constant column. Dropping this column

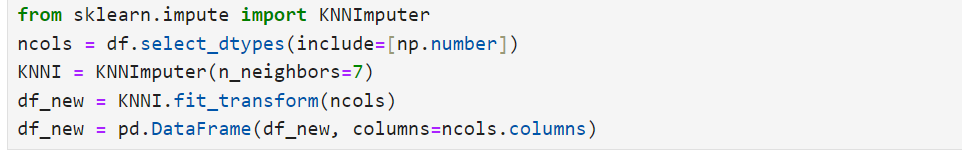


So, after dropping most of the columns we have 12 columns left.

**Encoding categorical variables** is another important step. Machine learning models require numerical input, so categorical variables are converted into numerical form. Techniques like one-hot encoding, label encoding or ordinal encoding are used based on the type of categorical variable.

Here I use Ordinal Encoding to convert the categorical values into numerical values. I used this technique because here the categories have some specific order or hierarchy. After using Ordinal Encoder our all-categorical variables are converted into numeric values.

**Imputation:**  Replacing Null values with mean, median and mode can be a good approach instead of removing all the Null values. Depending on the datatype whether it is categorical or numerical based on that we have to use the imputation methods like for categorical we have to use mode and for numerical we have to use mean and median. Another approach is using KNN Imputer, It basically look for the n\_neighbors values that I have mentioned and then do the mean of these numbers and fill the Null values.



**Skewness:** In our dataset ['capacity\_mw', 'longitude', 'other\_fuel1', 'commissioning\_year', 'geolocation\_source', 'generation\_gwh\_2014', 'generation\_gwh\_2015', 'generation\_gwh\_2016', 'generation\_gwh\_2017', 'generation\_gwh\_2018'] these columns has the skewness. Removing skewness from the dataset will give us better accuracy because tail region contains most of the values then our model will train specifically on these data range which is maximum this will create biasness in our model.

For removing skewness here, I used cube root method that comes under NumPy library.

**Outliers:** In statistics, an outlier is a data point that significantly differs from other observations. An outlier may be due to variability in the data or it could indicate experimental error, the latter are sometimes excluded from the data set.

Outliers can occur by chance in any distribution, but they often indicate either measurement error or that the population has a heavy-tailed distribution. In the former case one wishes to discard them or use statistics that are robust to outliers, while in the latter case they indicate that the distribution has high skewness and that one should be very cautious in using tools or intuitions that assume a normal distribution.

Here I removed outliers by using zscore method which comes under the principle of central limit theorem, that’s mean if any values are above or below 3 sigmas then I will discard them by considering them as outliers.

**Data transformation:** After analysing all the data I can see that our data are not in proper range or the proper distribution so, I uses here standard scaling method. This method basically converts the distribution in normal distribution where mean = median and standard deviation is (-1 to +1).

**Feature Engineering:**  After encountering with 11 columns, I use sequential feature selection method for further reduction of columns. After using I get 8 best columns but the r2 score I get is considerably low so that I did not use it. I use all the 11 columns.

**Training and Testing sets**: Finally, the data is split into **training and testing sets**. This allows us to train our model on one set of data and then test it on unseen data to evaluate its performance. Her e I uses 80-20 method that’s mean I uses 80% of our data for the training purpose and 20% for the testing purpose.

* **Building Machine Learning Models:**

After doing all the process this is our final process **Model Building**.

Here I basically use most of the regression algorithms and then comparing the metrices for all the algorithms and checks which model is performing the best.

These are some models that I used

LR=LinearRegression()

RD=Ridge()

LS=Lasso()

EN=ElasticNet()

svr=SVR()

DTR = DecisionTreeRegressor()

GB=GradientBoostingRegressor()

RFR=RandomForestRegressor()

ETR=ExtraTreesRegressor()

So apart from all the models **ExtraTreesRegressor** model is performing the best.

Because, the **R2 score** is 0.9566, which indicates that approximately 95.66% of the variance in the dependent variable is predictable from the independent variables. This is a high score, suggesting my model has learned the training data well.

The **Mean Absolute Error (MAE)** is 0.4244. This is the average of the absolute differences between the predicted and actual values. It gives an idea of how wrong the predictions were.

The **Cross-Validation Score** is 0.9031. This is the average of the model’s performance measured over different subsets of the training data. This score is slightly lower than the R2 score, indicating that the model might be slightly overfitting the training data.

The **difference between the mean CV and R2 score** is 0.0535. This difference, while small, suggests that the model might be slightly overfitting the training data, as it performs slightly worse on unseen data compared to the training data.

**The reason why I choose this algorithm is**:

Like other tree-based methods **ExtraTreesRegressor** is less prone to overfitting. The algorithm introduces extra randomness when growing trees; instead of searching for the most optimal threshold for each feature to split on, it searches for the best feature among a random subset of features and this threshold is randomly sampled from the range of the selected feature. This usually allows to reduce the variance of the model a bit more, trading a bit of bias.

This can capture non-linear relationships between features and the target variable, as well as interactions between features.

It is also quite robust to outliers in the dataset.

This can handle missing values in the dataset.

This has fewer hyperparameters to tune compared to other complex models like Gradient Boosting or Neural Networks, making it easier to find a good, robust model.

* **Concluding Remarks:**

In this blog, we embarked on a journey to understand and analyse the Global Power Plant dataset. We started by defining the problem statement, which was to predict the capacity of power plants based on various features. So, this was basically a regression problem.

Through rigorous data analysis and exploratory data analysis (EDA), we gained valuable insights into the dataset. We identified key features, outliers, and relationships between different variables. These insights guided us in the pre-processing stage, where we cleaned and transformed the data to make it suitable for machine learning models.

We then built several machine learning models, each with its strengths and weaknesses. We evaluated these models based on their performance metrics and chose the best one for our problem.

The journey doesn’t end here. The field of machine learning is vast and ever-evolving. There are always new techniques and algorithms to explore. We can further improve our model by tuning hyperparameters, using more complex models, or even using ensemble methods.

In conclusion, this project has shown us the power of machine learning in making predictions and helping us understand complex datasets. It has been a rewarding experience, and we hope that our readers will find it equally enlightening. We look forward to continuing our exploration of machine learning and its applications in future projects.

Remember, the key to mastering machine learning is practice and curiosity. So, keep exploring, keep learning, and most importantly, have fun along the way!

Thank you for joining us on this journey. We hope to see you again in our next blog post. Until then, happy learning!