

Evaluation Projects

PHASE -4

**Global Power Plant Database**

SAYAN MONDAL | DS 2309 | 06/03/2024

# 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. It will be continuously updated as data becomes available.
* Fuel Type Aggregation
* We define the "Fuel Type" attribute of our database based on common fuel categories.
* Prediction: Make two predictions

1) Primary Fuel

2) capacity mw

* 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

# Data Analysis

As we can see by the project description, we need two types of predictions amongst which one of them (**primary fuel**) needs to be done using **Classification models** and the other asked (**capacity mw**) must be done through **Predictions models**.

For the data analysis at first necessary libraries are imported. Then the data is loaded by using pandas’ function.

Cleaning the data involves removing any inconsistencies or errors that may exist in the dataset, such as duplicate entries, incorrect values, or outliers. This step is crucial to ensure the accuracy and reliability of the analysis.

Handling missing values is another important aspect of data analysis, as missing data can skew results and lead to erroneous conclusions. This may involve imputing missing values using statistical techniques or dropping rows or columns with a significant number of missing values.

Exploring the various features available in the dataset allows us to better understand the relationships between different variables and identify patterns or trends within the data. This may involve creating visualizations such as histograms, scatter plots, or correlation matrices to better visualize the data.

By understanding the distribution of different variables, we can gain insights into the underlying structure of the data and make more informed decisions about the next steps in the analysis. This includes selecting appropriate statistical methods, building predictive models, or generating actionable insights from the data.

Overall, thorough data analysis is essential in order to extract meaningful insights from the Global Power Plant Database and guide the project towards successful outcomes.

# EDA Concluding Remarks

* The data provide contains **907 rows** and **27 columns**.
* The data is definitely widely consisting of two data types that are **object** and **float64.** The columns in the data are ['country', 'country\_long', 'name', 'gppd\_idnr', 'capacity\_mw',

'latitude', 'longitude', 'primary\_fuel', 'other\_fuel1', 'other\_fuel2',

'other\_fuel3', 'commissioning\_year', 'owner', 'source', 'url',

'geolocation\_source', 'wepp\_id', 'year\_of\_capacity\_data',

'generation\_gwh\_2013', 'generation\_gwh\_2014', 'generation\_gwh\_2015',

'generation\_gwh\_2016', 'generation\_gwh\_2017', 'generation\_gwh\_2018',

'generation\_gwh\_2019', 'generation\_data\_source',

'estimated\_generation\_gwh'].

* The complete data is free of any duplicates as such checked through **duplicated().sum()** function
* But still it consists of various null values in different column which is close to **10,000.**
* Several columns have missing values, notably [**latitude**, **longitude**, **other\_fuel1**, **other\_fuel2**, **other\_fuel3**, **commissioning\_year**, **owner**, **geolocation\_source**, **year\_of\_capacity\_data**, **generation\_gwh\_2013** to **generation\_gwh\_2019**, **generation\_data\_source**, and **estimated\_generation\_gwh]**.
* As through primary data visualization we can find that the **primary fuel- coal and hydro is the most used.** Followed by solar, wind, gas, biomass, oil, nuclear.
* In the **year 2015** it had the highest unit of usage in the column **commission\_year.**
* In India coal and hydro are the most used primary\_fuel as found in histogram comparison with primary\_fuel and country.
* **Solar pow**er accounts to **28.4%** and **coal** to **27.7%** which are the top two highest primary\_fuel in respect to proportion in the data frame

# Pre-processing Pipeline

* The **data** has quite a number of categorical columns which have object type data which are need to transformed accordingly to numeric using **Label Encoder.**
* The data also has Outliers found by at first data visualization in box plot. These columns are ['capacity\_mw','longitude', 'commissioning\_year',

'generation\_gwh\_2014', 'generation\_gwh\_2015', 'generation\_gwh\_2016',

'generation\_gwh\_2017', 'generation\_gwh\_2018']. Which are then removed using **Z- Score method**. After removal of the outliers the data at present contains **864 rows and 13 columns**.

* Using **Correlation function Corr ()** we find that the columns {'generation\_gwh\_2015', 'generation\_gwh\_2017', 'generation\_gwh\_2018'} are correlated with a value over 0.85 or 85%. Thus, they were removed from the dataset and proceeded.
* The column “other\_fuel1” was also removed from the data due to its lack of importance in our project aim topic.
* Thus, after all these removals of column the data now has 846 **rows and 9 columns.**
* **We also find various Correlation of desired column such as ‘capacity mw’ with other features and found that the column generation\_gwh\_2016 is highly correlated with it followed by generation\_ gwh\_2014 and the least correlated feature with our target variable capacity mw is primary fuel.**
* **Similarly, source column is highly correlated with our other target variable Primary fuel and geolocation\_source is the least correlated.**
* **Now we split the data into various data frames X and Y for our target variable “capacity\_mw” where X contains all the feature variables and Y is the target variable. Similarly, also X1 consists of the feature variable for the prediction of primary fuel and Y1 contains the target variable Primary\_fuel.**
* Followed by which Skewness is checked and the data is transformed to make it properly Skewed neither too much left skewed or too much right skewed.
* Later the feature variables for both the target variables are Standardised using **StandardScaler function** to make the data more usable for our project work.

# Building Machine Learning Models

* At first for predicting our first target variable Capacity mw the split data are used in a model consisting of various **Regressors [LinearRegression(), Ridge(),Lasso()**

**, DecisionTreeRegressor(), SVR(),KNeighborsRegressor(),RandomForestRegressor(),SGDRegressor(),BaggingRegressor(),GradientBoostingRegressor(),XGBRegressor(),AdaBoostRegressor()]** amongst which metrics are calculated [**Best R2 Score, Best Random State, MSE, MAE**].

* Amongst the above Regressor models **XGBRegressor and KNeighbors are the two best performing Regressors on the basis of their Best R2 Score and MAE**. Which are then used in Hyperparameter tuning and later it was found that yet **XGBRegressor was still the best performing amongst the top two with the Best R2 score of above 79% after hyperparameter tuning at a random state of 63**.
* performing hyperparameter tuning and model evaluation for an XGBoost regressor using the RandomizedSearchCV algorithm.
* We start by splitting our data into training and testing sets using the train\_test\_split function.
* We define a hyperparameter grid param\_grid that contains the list of hyperparameters and values we want to search over.
* We create a RandomizedSearchCV object random\_search with the XGBRegressor model, the hyperparameter grid, and other configuration parameters like the number of iterations, scoring metric, cross-validation folds, and number of jobs.
* We fit the RandomizedSearchCV object to the training data to search for the best hyperparameters.
* We extract the best hyperparameters found by the search and create a new XGBRegressor model best\_XGB with those parameters.
* We perform cross-validation to calculate the R-squared score of the best model.
* We fit the best model on the training data and make predictions on the test data.
* We evaluate the model by calculating the R-squared score, Mean Squared Error (MSE), and Mean Absolute Error (MAE) on the test set.
* Finally, we print out the best R-squared score, MSE, MAE, and the cross-validated R-squared scores.
* This process allows us to find the best hyperparameters for the XGBoost model, train it on the data, evaluate its performance, and get a more reliable estimate of how well the model generalizes to unseen data through cross-validation.
* Secondly, for predicting our first target variable Primary Fuel the split data are used in a model consisting of various **Classifiers [Logistic Regression**

**(), Decision Tree Classifier (), Support Vector Classifier (), K-Neighbors Classifier (), Random Forest Classifier (), Gaussian (), Extra Trees Classifier**

**()]** amongst which metrics are calculated

**Best Random State, Accuracy, Precision, Recall, F1Score, Support, Confusion Matrix**].

* Amongst the above Regressor models **Extra Trees Classifier**

**and Random Forest Classifier are the two best performing Regressors on the basis of their Best R2 Score and MAE**. Which are then used in Hyperparameter tuning.

* A RandomizedSearchCV object is then created to search for the best hyperparameters for the model. RandomizedSearchCV searches across the hyperparameter grid defined in param\_grid to find the best combination of parameters that optimize a specified scoring metric, which in this case is accuracy.
* Predictions are made on the test data using the best classifier, and the accuracy of the model is calculated using the accuracy score function by comparing the predicted labels with the actual labels in the test set.
* Later it was found that yet **Extra Trees Classifier was still the best performing amongst the top two with the Best Accuracy score of above 95% after hyperparameter tuning at a random state of 30**.
* THE FINAL MODEL FOR PREDICTING CAPACITY OF THE POWER PLANT WAS SAVED AS globalPlant\_capacity\_model.pkl
* THE FINAL MODEL FOR PREDICTING PRIMARY FUEL OF THE POWER PLANT WAS SAVED AS globalPlant\_fuel\_model.pkl

# Concluding Remarks

The two of the target variables can now be predicted using the best performing models:

* XGB regressor for predicting Capacity mw using the Best random state 63.
* Extra Trees Classifier for predicting Primary \_Fuel using the Best random state 30.

I would greatly like to thank **FLIP ROBO** and my **SME Shwetank Mishra** for giving me such an amazing opportunity to do such real time projects based on varied fields of work which required different analysis and techniques in each case.

Thanking,

Sayan Mondal

DS 2309