The Global power plant study - Darshan Dhirai

The Global Power Plant Database is a comprehensive, open source database of power plants in India that was provided to us. It centralizes power plant data to make it easier to navigate, compare and draw insights for one’s own analysis. The database covers approximately 908 power plants and includes thermal plants (e.g. coal, gas, oil, nuclear, biomass, waste, geothermal) and renewables (e.g. hydro, wind, solar). Each power plant is located through latitude and longitude and entries contain information on plant capacity, generation, ownership, and fuel type.

**Problem Definition** -

Building a model that predicts the capacity and fuel type from the information about the plant provided (country, identifier, latitude, longitude, primary fuel, other fuels, commissioning year, owner, source, url, geolocation source, wepp id, year of capacity data, generation in gwh from the years 2013 to 2019, estimated generation from the year 2013 to 2017 and so on). Owing to the nature of the target variables, both, classification and regression models were required.

**EDA concluding remarks:**

1. The shape of dataset was 908 X 25. Fuel type and capacity were target variables - 1 being categorical and the other being continuous.
2. The dataset contained information factors, namely:

* `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

1. There were many missing values that needed to be addressed. Also, there was vivid skew in many contuinuous factors which was understood through the analysis of the skew and description of the dataset.
2. There was varying amounts of data missing data in factors namely:
   1. Latitude
   2. Longitude
   3. Other\_fuel 1
   4. Commissioning\_year
   5. Owner
   6. geolocation\_source
   7. generation\_gwh\_2013
   8. generation\_gwh\_2014
   9. generation\_gwh\_2015
   10. generation\_gwh\_2016
   11. generation\_gwh\_2017

**Pre-processing**: The process mainly consisted of cleaning the data, identifying the data that is useful and not useful and also, filling missing values through correlation with other factors in the data set.

1. Dealing with null data is important. Almost all data was missing in wepp\_id, estimated\_generation\_gwh, other\_fuel2, other\_fuel3 which was then removed all together as it did not add anything in the information front. Country\_Long was the same as country - dropped. Latitude and longitude were to be imputed with reference to the country they are in. Other fuel to be removed as it is mostly missing data.
2. All names were unique, garnering no value for the data prediction. The gdpp identifier, upon splitting the numerical and string code, gave an insight that the string part had only 2 variations which was utilized and coded later.
3. In terms of other fuels, All Other fuel1s were from IND.
4. All object type columns were to be encoded - done through the labelencoder as they did not have any numerical significance whatsoever.
5. All data was from India itself - removed the country name as it didn’t give any insight.
6. Generation data source too was also same for all energy plants - it didn’t give any data richness - so, it was removed immediately.
7. There was a correlation matrix created to understand how correlated the different data points were to each other - generation\_gwh\_2013, generation\_gwh\_2014, generation\_gwh\_2015, generation\_gwh\_2016 and generation\_gwh\_2017 were very closely related to their previous year. And, upon creating the correlation graph, there was an understanding that the gwh generation from 2013 to 2017 depend on each other and also, heavily on the capacity. So, we used the KNNImputer to input null data from capacity and subsequent years of gwh generation. Latitude and longitude were correlated with gppd\_idhr. Itirative Imputation was used to get that data too. Commissioning year can be filled in using the mode. The reason behind using itirative imputation was that it maintained the correlation amongst 2 continuous variables that was present. A mean or KNN Imputation would have created anomalies in the data. The correlation amongst these laddered information factors was all above 0.85 (which is very high). It was exploited as follows:
   1. df['generation\_gwh\_2017'] = itim.fit\_transform(df[['capacity\_mw', 'generation\_gwh\_2017']])[:,1]
   2. df['generation\_gwh\_2016'] = itim.fit\_transform(df[['generation\_gwh\_2017', 'generation\_gwh\_2016']])[:,1]
   3. df['generation\_gwh\_2015'] = itim.fit\_transform(df[['generation\_gwh\_2016', 'generation\_gwh\_2015']])[:,1]
   4. df['generation\_gwh\_2014'] = itim.fit\_transform(df[['generation\_gwh\_2015', 'generation\_gwh\_2014']])[:,1]
   5. df['generation\_gwh\_2013'] = itim.fit\_transform(df[['generation\_gwh\_2014', 'generation\_gwh\_2013']])[:,1]
8. The commissioning year was mostly 2013 and didn’t vary all that much. Hence the null values in that column were filled with 2013. The data was also converted to age as 2013 was the least of the years that the plant was commissioned in.
9. There was a major skew seen in many continuous factors, namely:
   1. Longitude
   2. Commissioning\_year
   3. generation\_gwh\_2013
   4. generation\_gwh\_2014
   5. generation\_gwh\_2015
   6. generation\_gwh\_2016
   7. generation\_gwh\_2017

The box plot too signified this skew. This skew was eventually removed using z score (the data loss was less than 10% at z<3). However, there still was a requirement of yeo - johnson power transformation to get the skew to 0.5

1. URL, latitude, gppd\_idnr and source had very little impact on the target. They were removed.

**Overall, the pipeline of pre-processing included:**

1. Evaluation of data relevance through unique value counts - removing data that does not add value.
2. Encoding categorical data to make it ready for modelling.
3. Filling in the NA values or removing the columns altogether depending on the amount of data available
4. Evaluating the correlation amongst factors to be able to impute missing data through reference of a relevant column
5. Checking for multicollinearity, skew to be able to remove outliers and make the data normal (skew <= 0.5) and useful to be able to process.
6. Checking for the balance of the categorical target variable, in the absence of which we use smote to balance the data - in this case, it was not a problem as there were multiple categories of fuel type almost equally distributed

**Analysis of the data**:

1. Upon applying analyses such as SelectKBest, df.corr(), there was a realization that there is very high relation between both target variables and:
   1. Longitude
   2. Commissioning\_year
   3. generation\_gwh\_2013
   4. generation\_gwh\_2014
   5. generation\_gwh\_2015
   6. generation\_gwh\_2016
   7. generation\_gwh\_2017
2. The correlation of target variables with latitude, owner and data source was very minimal as these factors didn’t change much or changed too much to garner any insights. However, the longitude did have an impact on the type of fuel - owing to the geographical constraints to the fuel availability in that area.
3. The data was to be analyzed through the train - test - split method (test was 25% of the data) and mainly was put through 4 model types for each:

**For capacity\_mw**

* 1. knn = KNeighborsRegressor() - k-NN (k-Nearest Neighbor), one of the simplest machine learning algorithms, is non-parametric and lazy in nature. Non-parametric means that there is no assumption for the underlying data distribution i.e. the model structure is determined from the dataset. Lazy or instance-based learning means that for the purpose of model generation, it does not require any training data points and whole training data is used in the testing phase.
  2. lreg = LinearRegression() - 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.
  3. rf = RandomForestRegressor() - A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.
  4. dt = DecisionTreeRegressor() - the decision tree model is the model of computation in which an algorithm is considered to be basically a decision tree, i.e., a sequence of queries or tests that are done adaptively, so the outcome of the previous tests can influence the test is performed next.

**For fuel\_type**

* 1. dtc = DecisionTreeClassifier() - - Similar to regressor above. Only difference is that I gives categorical output.
  2. lor = LogisticRegression() - Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
  3. rfc = RandomForestClassifier() - Similar to regressor above. Only difference is that I gives categorical output.
  4. knc = KNeighborsClassifier() - - Similar to regressor above. Only difference is that I gives categorical output.

The reason behind using all the above models is that, apart from a slight similarity between Decision tree and Random forest, they apply completely different ways of prediction, giving us a holistic coverage of different ways of analyzing the data and coming out with the accurate predictions.

1. Upon analyzing the data through all the above models, Random Forest came out to be the clear winner in both formats because of its capacity to do multiple iterations with varied sample sets. It gave an accuracy of 83% in capacity and 95% in fuel type prediction.
2. Following the model building, there was a cross validation carried out on it to ensure there was no overfitting. The method employed was Grid Search CV with final params (after multiple combinations of params that were checked) as follows:
   1. params = {'bootstrap': [True, False],

'n\_estimators': [150, 200, 250, 300],

'max\_depth': [9, 11, 13, 15],

'min\_samples\_split': [5, 10, 25],

'min\_samples\_leaf': [5, 10, 25]}

1. The best params were found to be:
   1. **Capacity**: {'bootstrap': True, 'max\_depth': 13, 'min\_samples\_leaf': 5, 'min\_samples\_split': 10, 'n\_estimators': 250}
   2. **Fuel type**: {'bootstrap': False, 'max\_depth': 11, 'min\_samples\_leaf': 5, 'min\_samples\_split': 25, 'n\_estimators': 200}
   3. The lack of overfitting was understood because of the minimal difference between output of crossvalidated accuracy and the former.
2. Hence, because of a final accuracy of almost 85 and 92%, we saved the random forest modes as the final one.

**Conclusion**:

It was possible to get such high accuracy scores with the Global power plant data set because of mainly 3 factors:

1. Removing unnecessary data
2. Back tracking the correlation from the target to the factors to be able to feed in the right information in NA areas
3. Having a plethora of varied modelling methodologies to get the best way forward to fit the data - in this case, Random Forest model.