Data Science

REPORT TITLE

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.

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

Data Analysis

DATA COLLECTION CRITERIA AND SOURCES

The database is built entirely from open sources, which are publicly available on the Internet, including data from national government agencies, reports from companies that build power plants or provide their components, data from public utilities, and information from multinational organizations. One goal of the data collection process is to identify reliable sources that are regularly updated and sources of data that are unavailable elsewhere. Data collection occurs on a country-by-country basis, with each country requiring different strategies to integrate data. Ideally, we use sources that are comprehensive and can be integrated automatically (through application programming interfaces, or APIs), but this is not usually possible. For many countries, data collection requires manually gathering data from various sources. This section describes the data collection criteria (reliability and ease of updating) and which data sources were used. 3.1 Data Reliability The reliability of this database depends on the reliability of the data sources we identify and aggregate. Sources that are directly linked to power plant operations or have the legal authority to gather power plant statistics are considered the most reliable.

The quality of data sources is broadly ranked as follows:

1. Primary sources or entities with legal authority Definition

▪ Data provider is a primary source (i.e., not aggregated from other sources)

▪ Source has the authority to collect statistics directly from plant-operating entities Criteria ▪ Government agency reporting on a power plant within its boundaries;

company reporting on a power plant it owns or operates;

construction or manufacturing company reporting on a power plant it has serviced Examples

▪ National governments

▪ Utilities ▪ Transmission or system operators

▪ Power plant construction companies

▪ Power plant operators

▪ Intergovernmental organizations

2. Secondary sources without legal authority but with quality assurance processes Definition

- **Data provider is a secondary or indirect source (e.g., aggregated data source)**

Criteria

- **Data provider has a system or process for crosschecking/verifying data claims**
- **Data are traceable to a source, although connecting a specific data point to a specific source may be difficult**

For primary and secondary sources, data must meet the definitions and criteria listed above. The same rules apply to crowdsourced data, although the verification stage makes the data collection process for crowdsourced data more time-consuming. The database uses the most reliable data available for each power plant observation. Although official data is preferred because it is most authoritative, we cannot guarantee it is completely accurate. In general, we do not have access to alternative and equally credible sources of data and cannot verify the accuracy of official data. However, we are able to verify plants' geolocation through satellite imagery and have conducted random sampling, which lets us estimate how reliably our data is geolocated (see Section 5.2). The data sources can vary for each plant characteristic or indicator (shown in Table 3). For installed capacity, the database uses type 1 sources (primary sources or entities with legal authorities) for 67 percent of the data and type 2 sources (secondary sources) for 33 percent of the data. We use crowdsourcing to expand the number of characteristics that are covered for each plant. At this point we have not developed a formal process for contributing data or submitting corrections through crowdsourcing, although we have accepted some corrections from researchers who have used the beta version; we have also granted KTH access to our data for some manual additions and edits. At a later stage we will revisit a more robust crowdsourcing feature.

Ease of Updating the Data

The second criterion used to evaluate a data source is how easy it is to update the data. The database favors sources that provide automatic or mandated updates at set intervals to ensure it reflects the most recent information. Static or non-updating sources are avoided if an alternative exists. Preferred sources ▪ Update information in a regular, timely fashion ▪ Data format is easy to read and load into the database Examples ▪ Easiest to update: an API with machine-readable data maintained by a national government that updates annually ▪ Easy to update: a transmission operator that produces an annual spreadsheet of data ▪ Difficult to update: a utility website with information about a power plant in paragraph form

Data Sources

Although much of the data comes from information sources that report on a large number of power plants (such as the U.S. Energy Information Agency, Arab Union of Electricity, and European Network of Transmission System Operators), the majority of sources contained in the database provide only a small number of observations. More than 600 unique sources are used for the database that range in coverage from a single power plant to several thousand. Different characteristics of a power plant can be linked to different sources. Due to space limitations, we are not listing all the original sources in this technical note. Each power plant entry contains a direct link to where the data was obtained so users can check the original data source. We have documented the details of all sources of power plant characteristics by country in separate source documentation, which is available upon request. It is easy to track the data source within the database because every power plant entry is directly connected to its source(s). We also include the year associated with the capacity data and the electricity generation information.The only information that is not linked to a source is geolocation (latitude and longitude), which users can verify independently through satellite imagery.2 In some cases, power plant location is collected from web map providers such as Google Maps, so it is not always attributable to a unique data source. Geolocation information in the database is determined by a few methods, including parsing datasets produced by standard sources, matching plants to other global datasets containing geolocation data (detailed in the next section), using manual verification via satellite or aerial imagery.

EDA CONCLUDING REMARKS

EDA SUMMARY

country	country_long	name	gppd_idnr	capacity_mw	primary_fuel	source	
0	0	0	0	657	2.5	6	109
1	0	0	1	519	98.0	1	174
2	0	0	2	853	39.2	7	21
3	0	0	3	0	135.0	2	22
4	0	0	4	1	1800.0	1	22
...	...	...	...	...	...	...	...
902	0	0	902	491	1600.0	1	22
903	0	0	903	822	3.0	6	77
904	0	0	904	891	25.5	7	21
905	0	0	905	539	80.0	1	59
906	0	0	906	876	16.5	7	21

907 rows × 7 columns
capacity_mw: we can see capacity is mostly dependent on Coal.

primary_fuel: Mostly Coal is used as primary fuel.

MAINTAINING THE DATABASE

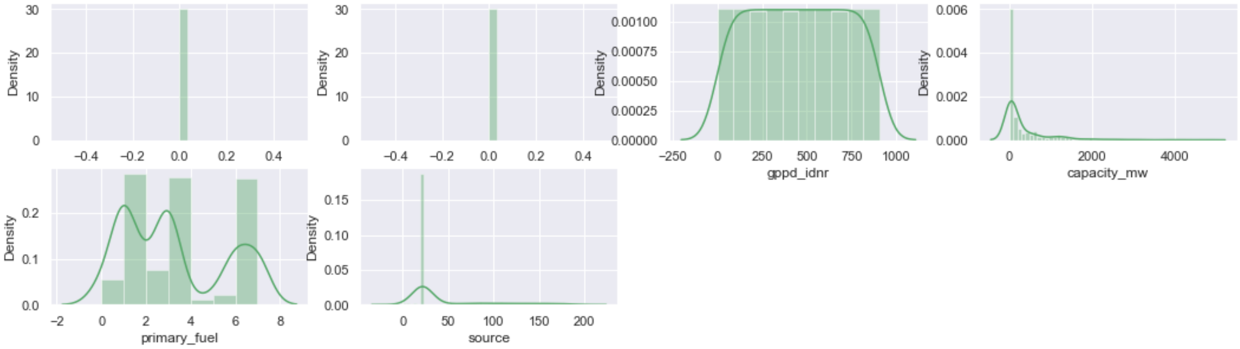
Ease of updating is a major concern. There are two basic paths to updating the database that are not mutually exclusive. When new or updated data are identified for a specific country, they are either incorporated manually into the relevant Fusion Table, or the relevant automated script is run to incorporate them. When machine-readable data are identified for a country for the first time and replace manually collected data, a new automated script is developed to incorporate them. This replaces the manual Fusion Table (which is archived for later reference). Much of the data, especially for low-income countries, has been obtained manually. This means that any updating is likely to also be done manually. At this stage, additional data are likely to come from smaller sources that provide the data occasionally and in formats that are not machine readable. Collecting more data manually expands the database's coverage but makes it more expensive to maintain over time. To address the update and maintenance challenges, WRI has written guidelines for data updates for each country where the data were collected manually.16 Through a partnership with KTH, the data for Africa and Latin America will be manually updated periodically.

FUTURE STEPS

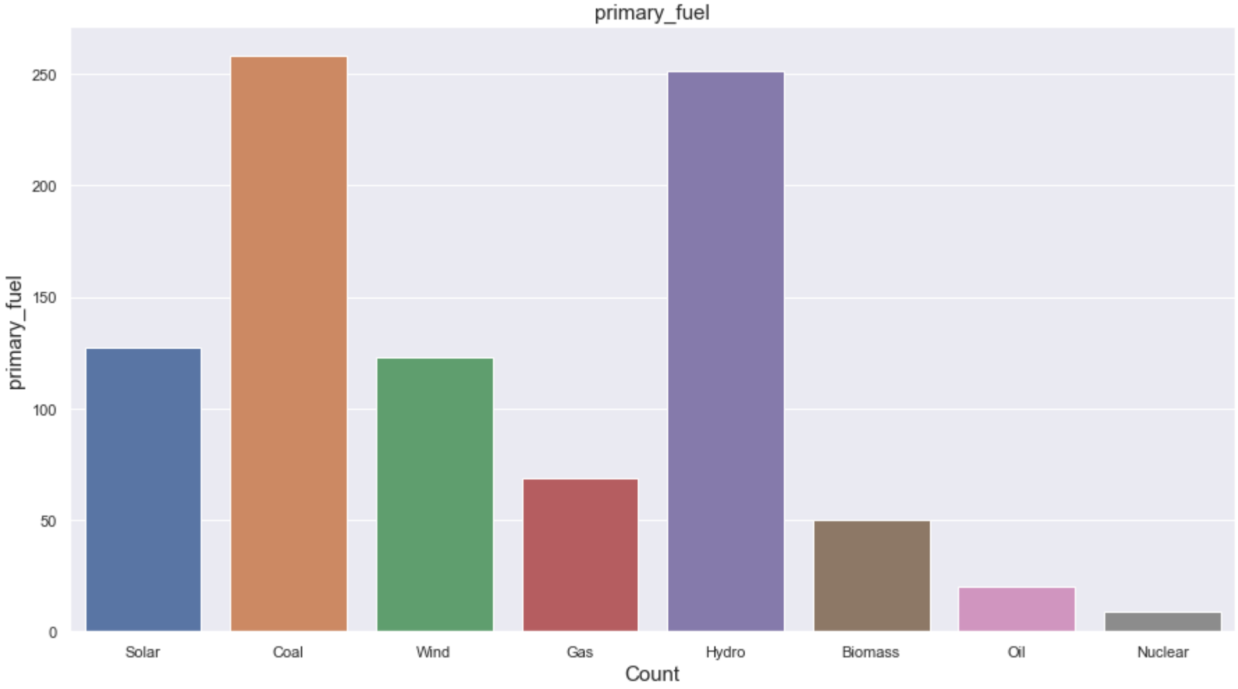
Partners and power plant data experts provided feedback on the database during its development in 2016 and 2017. Moving forward, the focus will be on improving the core database and launching new research based on the core dataset. Future steps include ▪ improving generation estimates by plant; ▪ expanding data coverage using artificial intelligence, remote sensing, and crowdsourcing; and ▪ expanding the number of indicators included by plant, specifically greenhouse gas emissions, water use, and particulate matter emissions. The estimation of generation by plant will take place in close consultation with partner organizations (Carbon Tracker, GE, IBM). We aim to both identify new data that can improve the accuracy of generation estimation through the existing machine-learning model (see Appendix A) and explore alternative ways to estimate plant-level generation. This may include using country-level unit commitment models. We plan to use advanced data collection methods to expand coverage of power plants in the database. In partnership with IBM, we will test whether AI programs can query text-based web sources, identify power plant commissioning and closure dates, and translate them into a format that is easy to read into the database.We are testing new data collection methods, including text analysis from web sources and remote sensing using satellite imagery.

Remote sensing can be especially useful for identifying power plant characteristics or locating new power plants. Using remote sensing effectively requires using a large amount of data as training data for the machine-learning algorithm, as well as recent and highresolution imagery. Using crowdsourcing to confirm plant geolocation has been tested with KTH master students, to some success. Allowing the broader public to have access to the database via an online platform may let the project expand geolocated coverage of plants.

Relative Influence of Each Independent Variable



Individual Fuel Types



Explanatory Power and Accuracy We measure the explanatory power and accuracy of the GBM projections by comparing how much of the in-sample variation is explained by the projections and by showing the size of the error, both in sample and out of sample. The out-of-sample measure refers to jurisdictions where we have the information on generation capacity and capacity factors by plant. This should be interpreted as a lower bound of the projection error for jurisdictions where we do not have access to generation information by plant. We cannot independently verify the validity of the projections in these jurisdictions.

the set of tuning parameters for the GBM process that we use in this application. Number of iterations (n_estimators). This is the number of boosting stages (or, equivalently, the number of weak learners) that the model performs. This can generally be set fairly high, as gradient boosting techniques are not particularly vulnerable to overfitting. We determined empirically that the optimum R2 value occurs for a value of 1,500. Learning rate (learning_rate). This is the fraction by which the contribution to the loss function is reduced for each tree. A small value leads to a larger number of estimators being required to achieve a similar loss, although this also reduces the chance of the model getting stuck in a local rather than global optimum. We empirically find that 0.003 leads to good performance. Depth of each tree (max_depth). This value specifies the number of splits at each node, or the degree of variable interactions. A value at 1 means there is no interaction between variables and an additive model will be applied. A value greater than 1 will add nonlinearities to the model and improve performance.

We empirically find a value of 6 to be optimum. Number of cross-validation folds (num_folds). To measure the performance of the model, we use a crossvalidation technique; the training dataset is randomly divided into N equal-size subsets, and the model is repeatedly fit with a different set of N - 1 subsets and tested against the remaining one. The average accuracy is then reported. Following best practices, we choose a value of N = 10. Subsampling fraction (subsample). To improve the model's robustness, we use a subsampling technique such that only a (random) fraction of the training data is used to train each weak learner. Following best practice, we set this fraction at 0.5. Loss function (loss).

The loss function is the quantity the model training process is attempting to optimize. We choose the Huber loss, which is a combination of least squares and least absolute deviation. This tends to be more robust against outliers in training data.

Pre Processing Pipeline

- **Missing values**
- **Polynomial features**
- **Categorical features**
- **Numerical features**
- **Custom transformations**
- **Feature scaling**
- **Normalization**

Missing values

Handling missing values is an essential preprocessing task that can drastically deteriorate your model when not done with sufficient care. A few questions should come up when handling missing values.

Imputing values

For filling up missing values with common strategies, sklearn provides a *SimpleImputer*. The four main strategies are *mean*, *most_frequent*, *median* and *constant* (don't forget to set the *fill_value* parameter). In the example below we impute missing values for our dataframe X with the feature's mean.

Polynomial features

Creating polynomial features is a simple and common way of feature engineering that adds complexity to numeric input data by combining features.

Polynomial features are often created when we want to include the notion that there exists a nonlinear relationship between the features and the target.They are mostly used to add complexity to linear models with little features, or when we suspect the effect of one feature is dependent on another feature.

Categorical features

Munging categorical data is another essential process during data preprocessing. Unfortunately, *sklearn's* machine learning library does not support handling categorical data. Even for tree-based models, it is necessary to convert categorical features to a numerical representation.

Discretization

Discretization, also known as quantization or binning, divides a continuous feature into a pre-specified number of categories (bins), and thus makes the data discrete.One of the main goals of a discretization is to significantly reduce the number of discrete intervals of a continuous attribute. Hence, why this transformation can increase the performance of tree based models.

Binarization

Feature binarization is the process of tresholding numerical features to get boolean values. Or in other words, assign a boolean value (*True* or *False*) to each sample based on a threshold. Note that binarization is an extreme form of two-bin discretization.

Custom transformers

If you want to convert an existing function into a transformer to assist in data cleaning or processing, you can implement a transformer from an arbitrary function with *FunctionTransformer*. This class can be useful if you're working with a *Pipeline* in *sklearn*, but can easily be replaced by applying a lambda function to the feature you want to transform.

Feature scaling

The next logical step in our preprocessing pipeline is to scale our features. Before applying any scaling transformations it is very important to split your data into a train set and a test set. If you start scaling before, your training (and test) data might end up scaled around a mean value (see below) that is not actually the mean of the train or test data, and go past the whole reason why you're scaling in the first place.

Standardization

Standardization is a transformation that centers the data by removing the mean value of each feature and then scale it by dividing (non-constant) features by their standard deviation. After standardizing data the mean will be zero and the standard deviation one.

Normalization

Normalization is the process of scaling individual samples to have unit norm. In basic terms you need to normalize data when the algorithm predicts based on the weighted relationships formed between data points. Scaling inputs to unit norms is a common operation for text classification or clustering.

```
 Building Machine Learning Models

Fitting 4 folds for each of 1 candidates, totalling 4 fits

One-vs-One ROC AUC scores:

0.844087 (macro),

0.868185 (weighted by prevalence)

==========================================

One-vs-Rest ROC AUC scores:


0.855566 (macro),

0.917017 (weighted by prevalence)
```