Schneider Electric - Team 28 - Data Science

Alvaro García Bamala

Daniel Häggström Pérez-flecha

Javier Hernández de Lorenzo

Data extraction

Extract csv data

We use pandas read_csv method to read both train1.csv and train2.csv. The only difference between these two files is that we need to use a semicolon delimiter to read the train2.csv data.

Extract json data:

We use an iterative process, using python's library requests in order to query the json data.
It is then loaded into a Pandas Dataframe and some small modifications are performed.

Extract pdf data

We use PDFQuery library to convert pdf into xml format, once in xml format we can get the different areas from the pdf. Once we have the areas, we scrap each different area and save the result into a dataframe. Some values needed some parsing to get the actual values.

Data preparation

Check that the column names from the different sources of data match

We find some discrepancies and fix them.

Get sector code

Of note, we find that CSV data doesn't have the column EPRTRAnnexIMainActivityCode, but it can be obtained from EPRTRAnnexIMainActivityLabel, and this mapping is obtained from JSON data. With the exception of code 4(a)(vii), which we determined from this source https://iir.umweltbundesamt.de/2021/general/point_sources/start

Concatenate different sources of data

Once we have gathered all three intermediate datasets, we join them together. Some adjustments are needed, like column name changes and dropping unnecessary features.

Fix values of wind speed and temperature from max, min, and average columns:

- Defective pdf values:
  We noticed that these values from the pdf source had non realistic values. We also noticed that all of these values belonged to the UK. We computed the average of these

variables from the csv and the json sources for the rows that belonged to the UK and replaced the defective values of the pdf with the average we obtained.

- Swap values:
  We also noticed that some of the values from the min columns were greater than the values from the max columns. We swapped them in the scenarios that this problem occurred.

Encoding:

The classifiers we used for training required numerical data. We encoded all the non-numerical values to numerical values to be able to train the classifiers.

Split train test:

We perform a train test split of 80% training and 20% test.

Data modeling

Check validation metrics from several classifiers:

The algorithms we trained have been:

- Knn
- Naive Bayes
- Random forest

We achieved the best performance using Random Forest.
We also noticed that accuracy, precision, recall and f1 score had the exact same value. This happens because the sum of false positives is the same as the amount of false negatives.

Cross Validation using Random Forest:

We performed 10 splits and repeated this process 3 times to validate our Random Forest classifier.

Predict on test data:

Finally we used the final classifier to obtain predictions on the test data and stored the results on csv and json formats.