BUILD OUR DATASETS

1. We had 3 types of data formats: csv, json and pdf. For the csv we decided to export them directly with a read_csv instruction using the pandas library to get a dataframe object. To the second one we have to take into account the split character.
2. To get the data from the jsons, we first extracted the json from the url using the requests library, and then used the read_json to get the dataframe.
3. Lastly to get the data from the PDFs, we created different functions to parse it, but it can be the idea behind it was to get the text from the pages and get all of the attributes by comparing the words with the columns of the dataframe we want to obtain. Later on we had to change some of the column names in order to correctly merge all of the dataframes.
4. The last step was to merge all of the dataframes into one, which we were able to do using the command concat.

CLEAN OUR DATASETS

1. We decide to drop some columns because they are corrupted/not consistent since for instance min_temp has values greater than avg_temp
    - max_wind_speed
    - avg_wind_speed
    - min_wind_speed
    - max_temp
    - avg_temp
    - min_temp
2. We remove also some columns that are useless cause it can be seen clearly that they don't share any relation with the target pollutant or because we have a name and Id for the same feature(so we decide to drop one of them)
    - CONTINENT
    - targetRelease
    - REPORTER NAME
    - FacilityInspireID
    - facilityName
    - EPRTRAnnexIMainActivityLabel
    - (empty variable)
3. From the remain ones, we take the DAY WITH FOGS varible to discard those samples that are outliers (greater value than 5·IQR) because can affect negatively to our predictions.
4. Finally we construct a correlation matrix with those variables that we don't know exactly how are related with the output (variables City, EPRTRAnnexIMainActivityCode and countryName are mapped to int with a dictionary) . Finally we decide to build a final dataset with the first three features with more correlation, that are:
    - City -> corr: 0,55
    - EPRTRAnnexIMainActivityCode -> corr: 0,51
    - EPRTRSectorCode -> corr: 0,34

TRAINING

1. First we split our final dataset with train set with the 80% of the samples and test set with the remainder 20%. In that way, we are able to see results before testing the desired set.
2. We use several classification algorithms:
   - SVC
   - DT
   - Random Forest
   - Gradient Boosting
   - Voting Ensemble that combines 2 of the previous models
3. We fit the model, predict the output of the test set build from the final dataset and compute some statistics such as the accuracy of the model using the test set mentioned.
4. After that, we decide to equilibrate the number of samples of each class in order to find a better accuracy using the same algorithms. So we undersample NOX and CO2 and we repeat the computations.
5. For our case, we get a best accuracy using Gradient Boosting: 0,66

PREDICTIONS

1. Finally, we take the classification algorithm with best accuracy, in our case Gradient Boosting, and we use the predict function of that model with the desired test set to get our outputs.
2. To generate the predictions files we used the to_csv and to_json functions from the pandas library.