SCHackaton 2022

Work Analysis

*Code further explained in main.ipynb

Team 4: Pol Gràcia Espelt - Gerard Jover Pujol

Data Retrieval

3 data sources to store in pandas DataFrame format:

- **CSV:** Stored via pandas read_csv.
- **API:** Retrieval using requests module and stored via pandas normalize_json.
- **PDF:** Read via pdfplumber module and manually cleaned and stored in a DataFrame.

Final data frame with 65710 rows (yet to be processed).

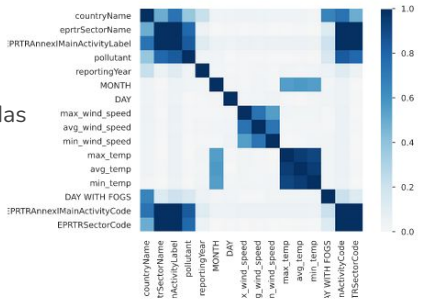
We managed to fill the 'EPRTRAnnexIMainActivityCode', 'EPRTRSectorCode' (which are important fields) and where empty in the csv files joining them with the 'EPRTRAnnexIMainActivityLabel' column on the api retrieved data.

Exploratory Data Analysis

pandas_profiling is the tool that has been used for EDA. The report has been included in the delivered solution despite not asking for it. We considered its analysis has been the most important part in the EDA process and has lead us to choose which variables to use with the models, understand the data, clean it...

Columns used for training are the ones with highest correlation with the target class. From this columns the ones with very high correlation have been depreciated to avoid overfitting.

Correlation matrix Phik (pandas profiler)



Phik (φk)

Phik (φk) is a new and practical correlation coefficient that works consistently between categorical, ordinal and interval variables, captures non-linear dependency and reverts to the Pearson correlation coefficient in case of a bivariate normal input distribution. There is extensive documentation available here.

Model

We have used two approaches for the classification task:

- **Machine learning algorithms:** We have worked with several algorithms: SVM, KNN, Decision Trees and XGBClassifier; mostly focusing on the last two. We have developed a ModelHandler class which performs a grid search for these models and returns the best performing hyparameters according to a confidence interval. We have had trouble mostly with the XGBClassifier as its grid search executed for 4 hours straight but it has been our best performing model.

- **Deep Learning algorithms:** We could not explore this option as far as we would have wanted due the lack of time and the lack of computing resources. It is time consuming to create a good dataset and dataloader class and create the functions using pytorch. We have just tested one small MLP network which has not performed better than our best ML models. We strongly believe that with more time we could have improved the results with this approach.

We have used several oversampling, undersampling and normalization methods to try to improve our score. In the notebook we have not showed how we trained every model with the best performing oversampling and undersampling methods.

*Note: Some of the algorithms tested have not been included in the notebook for their poor performance or due to the difficulty of merging results between the group members.

Results & conclusions

The models trained can accurately predict the class Methane (CH4) but are not able to distinguish between CO2 and NOX. We have found this problem across all the models we have trained and tested (ML and DL), with grid searches for each model and trying oversampling and undersampling techniques.

The main problem we have had with the models is the overfitting. We have tried to solve it via undersampling and reducing the depth of the training but resulting in poor classification score.

We conclude that there is very little correlation between the data and the pollutant target class and with the time and computation resources available we could not manage to get any results better than the shown.

With more time we could have tried to explore further into deep learning algorithms, with more complex architectures and hyperparameter search as there is a considerable amount of data to use for training.