**FIRS DELIVERABLE OF EMISSION TRACKING EXPLORATIVE PROJECT**

This project aims to test and prototype a dashboard to track carbon footprint of code. The final output will be a dashboard prototype and the knowledge to understand if we can proceed in the future to create a real project to track carbon footprint of all of our data science projects.

This first deliverable contains information about the setup and data used to create the protoype deliverable.

The project will use the CodeCarbon python library <https://codecarbon.io/#howitwork> that let track emission of python code.

We know that at the moment we have deployed projects that uses Oracle Cloud data science pipeline to run machine learning models. This pipeline can be made of 1 or more computing steps. Therefore to create the sample data and then the prototype dashabord we decided to recreate a simulated environment like this:

* We have 3 project
* Project 1 has 1 step, project 2 has 2 steps, project 3 has 3 steps
* We want to simulate that all the 3 projects run daily for a year
* We want to be able to aggregate data related to the different steps of a pipeline but about the same daily run.

Producing this data resulted in a table with XX rows and the following columns resulted as emission log of the code carbon library:

| **Field** | **Description** |
| --- | --- |
| timestamp | Time of the experiment in %Y-%m-%dT%H:%M:%S format |
| project\_name | Name of the project, defaults to codecarbon |
| run-id | id of the run |
| duration | Duration of the compute, in seconds |
| emissions | Emissions as CO₂-equivalents [CO₂eq], in kg |
| emissions\_rate | emissions divided per duration, in Kg/s |
| cpu\_power | CPU power (W) |
| gpu\_power | GPU power (W) |
| ram\_power | RAM power (W) |
| cpu\_energy | Energy used per CPU (kWh) |
| gpu\_energy | Energy used per GPU (kWh) |
| ram\_energy | Energy used per RAM (kWh) |
| energy\_consumed | sum of cpu\_energy, gpu\_energy and ram\_energy (kWh) |
| country\_name | Name of the country where the infrastructure is hosted |
| country\_iso\_code | 3-letter alphabet ISO Code of the respective country |
| region | Province/State/City where the compute infrastructure is hosted |
| on\_cloud | Y if the infrastructure is on cloud, N in case of private infrastructure |
| cloud\_provider | One of the 3 major cloud providers, aws/azure/gcp |
| cloud\_region | Geographical Region for respective cloud provider,  examples us-east-2 for aws, brazilsouth for azure, asia-east1 for gcp |
| os | os on the device  example Windows-10-10.0.19044-SP0 |
| python\_version | example 3.8.10 |
| cpu\_count: | number of CPU |
| cpu\_model | example Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz |
| gpu\_count | number of GPU |
| gpu\_model | example 1 x NVIDIA GeForce GTX 1080 Ti |
| longitude | Longitude, with reduced precision to a range of 11.1 km / 123 km².  This is done for privacy protection. |
| latitude | Latitude, with reduced precision to a range of 11.1 km / 123 km².  This is done for privacy protection. |
| ram\_total\_size | total RAM available (Go) |
| Tracking\_mode: | machine or process``(default to ``machine) |

The codecarbon library keeps track of the hardware usage during the run of the project and appends the logs in real time. It is aware of different hardware power usage so there's no need for any manual input when tracking a project. The tool also provides a build in dashboard, however due to the easily accessible nature of the result file, it is also possible to create a custom dashboard which can be tailor made to fit our own requirements.

The prototype simulates a real-life model training scenario. In the meantime, the same source code is capable of running a real project, however for this first phase toy datasets are used in order to iterate quickly. As a result of the previous choice, but also thanks to the efficency of modern processors, the energy consumption in our simulation is equal to watt/hours instead of the usual kilowatt metric(1kw = 1000w)