Data Science project for CAPGEMINI:

Lucas Benyamin

Elena Moreac

Arthus Rouhi

1) Subject Explications and Objectives

The purpose of this project is to industrialize a **Datascience model** through **containerization**.

In order to achieve this goal, we first convert a Jupyter notebook into a python script. The main goal is to containerize this python code containing our Deep Learning model using **Docker container**.

This Docker container must meet the following criteria:

- must contain Python, python scripts and required packages
- a volume mapping on an input directory containing images to predict
- a volume mapping on an output directory in the local filesystem

2) Description of the conception and implementation

To begin, we use and test the notebook "weather-classification-TP.ipynb" who include imports libraries, the predictions and All of libraries use in must be call in the "requirements" if we would like a run of the app.py To create the requirements by using "PyCharm", from Tools Menu, select sync Python Requirements. We have a opened dialog, specify the name of requirements file and use the compatible version.

We also have:

- An app.py containing the python code for loading and preprocessing of images, for loading of our classification model and to make predictions on the images and export them
- A dockerfile containng all the instructions required to build our functional image

3) Compilation process

a) Way to start - Initialization

First, you need Docker Desktop installed on your local machine

In your machine, create a local folder "ds-capgemini" containing 3 subfolders:

- input : contains the .jpg images to predict
- output : where the .csv output will be printed after we run the docker image
- model : contains the classification model ResNet152V2-Weather-Classification-03.h5

Do not lose sight of the path of your folder that you have created for the occasion in order to use it in the deployment.

We have for predictions "test_images" which you can use some images but we decided to use more in our case

b) Run the project

weather-classification-app using docker for deployment

First, build the image "image" containing all the requirements needed and our project, Docker command is:

```
docker build -t image .
```

Then, run the image "image" to launch the container with volume mapping on the local folder, Docker command is : On Linux:

```
docker run -v /Users/<name_of_user>/Desktop/ds-capgemini/output:/weather-classification-app/output
-v /Users/<name_of_user>/Desktop/ds-capgemini/input:/weather-classification-app/test_images
-v /Users/<name_of_user>/Desktop/ds-capgemini/model:/weather-classification-app/model image
```

On Windows:

```
docker run -v C:\Users<name_of_user>\Desktop\DS_Capgemini\input:/weather-classification-app/test_images -v C:\Users<name_of_user>\Desktop\DS_Capgemini\output:/weather-classification-app/output -v C:\Users<name_of_user>\Desktop\DS_Capgemini\model:/weather-classification-app/model image
```

These commands depend on where you have created your project folder "ds-capgemini".

Troubleshoot : if mkdir error, simply restart the docker desktop and try again

Finally you should get in the "output" folder of "ds-capgemini" (<name_of_the_folder_created_in_init>) folder:

- A csv file (with timestamp in filename) with two columns : "image_name" for the name of the image and "prediction_label" for the predicted class of this image
- A visual plot of the predictions to get a first visual idea of the performance of the Deep Learning model