# Disruptive Data Summer School

I decided to develop the project in Python even if I’m not an expert of this programming language for mainly two reasons: firstly for increasing my knowledge and secondly because Python is commonly consider one of the best choices in the field of big data manipulation and post processing.

I structured the project into a main file which is the entry point of the project. From there you can execute the single projects changing the id number. I’ve also written an utility function for io process.

Project1:

I haven’t worked on it yet, but in my idea is to retrieve recursively with a wget the html pages and map them into a csv/db

Project2:

Once data is retrieved trough google sheets or local csv, the io function divides the dataframe into two list. In order to avoid inconsistent data or errors, i filtered the nan/None data coming from dataframe and introduced an exception to avoid out of index errors.

The idea is to tokenize every uri of list1 acording to “/” or “-“ separators and check if each token of list1 is contained or contains tokens in each uri of list2. Every match increase a matching counter, and the first uri of list2 with the maximum matching counter is associated to the uri of list1. This is cycled for every uri of list1. I simply filtered this check on word length > 2 for avoiding connective words like “a”, “da”, “ai”…

Possible upgrades could be:

* A filter over a prior defined db of connective words to exclude
* The matching logic with a matching weight, such that some words are more important than others
* Similarity check on each token in order to spot plurals (eg. sale/sali)

Project3:

In my idea this project could have been accomplished calling a search engine API for each word, where we could specify geolocation (e.g. US). As far as I discovered, google disabled these api last year, and the best alternative <https://keywordtool.io/api> is not free.

Project4:

I completed this project simply ordering by name the list of words with sort function natively exposed in python, retrieved the number of words according to size list and the average word size simply adding chars of each word divided by words size.

# Biography

I’m a computer science and automation PhD engineer currently employed in ENEA research center. I’ve worked in the past in some ICT companies (IBM, Terrasystem, Content Interface) and I covered different business areas: mainly energy efficiency, digital signage and precision agriculture.

My skills cover both scientific/academic background (mainly regarding multiobjective optimization, forecasting models, soft computing, computer vision) and industry background (mainly regarding android app, web app and backend development, database management).

I recently began to work in the fields of big data and blockchain, so i found really interesting this summer school. It looks really useful for increasing both theoretical and technical aspects of big data world, although SEO is not my main business area.