# Final Project 2

Python For Data Analysis - Fall 2023

Due Date: December 6th, 2023

Version 1.0

## Goal

The goal of this group project is to write a plagiarism detector that is specialized towards finding similar Jupyter Notebooks.

## Deliverables

Your submission will consist of a single zip-file. The zip file will be subdivided into sections that contain the data, your investigations, your source code, your presentation slides in pdf format, a Jupyter Notebook that generates the results and a requirements.txt file.

More specifically, the structure of the zip file should be organized as follows:

```
{project_name}/
      data/
            0101.ipynb
            0102.ipynb
            ...
            0130.ipynb
      investigations/
            {notebook1}.ipynb
            {notebook2}.ipynb
            ...
      presentation/
            Project - Fall 2023.pdf
            {team_slides}.pdf
      src/
            {module1}.py
            {module2}.py
            ...
      requirements.txt
      results.ipynb
```

The `data` folder should simply contain all of the data provided for this project. The data for this project consists of 30 Jupyter Notebooks.

The `investigations` folder can contain any number of Jupyter Notebooks that you would like to save as part of your project. Examples include exploratory data analysis, figures, etc… Note

that these Jupyter Notebooks can import functions and classes that are defined in the modules in the `src` directory.

The `presentation` folder should contain this PDF file and a copy of the team slides (PDF) that you present in your talk.

The `src` folder should contain all of the code that you call from the `results.ipynb` Jupyter Notebook (as well as any of the notebooks in the `investigations` directory). Your code can be organized in any number of modules. It should be packaged into functions and/or classes.

The `results.ipynb` file will contain minimal code that kicks off your calculations and displays the results in the notebook. All of the underlying functions should be implemented in the modules in the `src` directory.

The `requirements.txt` file should contain all of the packages that are required to run your code.

Note that the {name} syntax indicates that you should replace {name} with a name of your choice. So for instance, your submission could look like this:

```
PlagiarismProject/
      data/
            0101.ipynb
            0102.ipynb
            ...
            0130.ipynb
      investigations/
            exploratory_data_analysis.ipynb
            benchmark_tests.ipynb
            random_stuff.ipynb
      presentation/
            Project - Fall 2023.pdf
            team1_presentation.pdf
      src/
            algorithm1.py
            ...
            algorithm4.py
            report_generator.py
      requirements.txt
      results.ipynb
```

# Data

The data consists of 30 Jupyter Notebooks. These were homework sets from a previous semester.

The files `0101.ipynb` and `0102.ipynb` are exact copies of each other. You can use these as benchmark tests to verify that your algorithms work as expected. The file `0103.ipynb` is very close to `0101.ipynb` so should also have a high similarity score.

# Algorithms

Your presentation should have at least one algorithm per group member. These algorithms should compute the similarity score between two documents. You can tailor them towards Jupyter Notebooks. Your algorithms may output numerical scores, categorical scores, eg (`low, medium, high`) or even boolean scores. You are welcome to use any library to develop these scores, just make sure to include the package in the `requirements.txt` file. You are also welcome to implement your own algorithm!

## `results.ipynb`

The `results.ipynb` Notebook should have minimal code. All of the code should be in the src modules. For example, it could look something like:

```
from src.module1 import Detector

detector = Detector(path="data/")
detector.generate_results()
```