Configuration Manual

MSc Research Project
Masters in Data Analysis

Rory O’Loughlin
Student ID: 17132835

School of Computing
National College of Ireland

Supervisor: Jorge Basilio

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Rory O’Loughlin		
Student ID:	17132835		
Programme:	Masters in Data Analytics	**Year:**	2
Module:	Masters Project		
Lecturer:	Jorge Basilio		
Submission Due Date:	16/08/2021		
Project Title:	Investigation Into Parameterisation of the FIMOTS Algorithm for Computational Efficiencies on FPM on Streams of Data		
Word Count:	661	**Page Count: 3**	

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Rory O’Loughlin

Date: 01/08/2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Rory O'Loughlin
Student ID: 17132835

1 Pre-Requisites

This project is completed within Python and uses PySpark. Therefore, a pre-requisite for running the enclosed python scripts is to be able to run on a UNIX machine with access to both of:

- Python 3
- PySpark

The PySpark implementation is required to have a version of MLlib[1] which includes the implementation of FP-Growth[2] which is used for benchmark results. The Python files assume that PySpark may be retrieved using the findspark[3] utility which may also need to be installed on the chosen machine.

Aside from these dependencies the graphs are generated using matplotlib[4] which is not standard within all python environments.

2 Installation

The zip file *projectFiles* is supplied. Unzip it on a UNIX machine within which PySpark is already installed. Following the unzipping of this file there should be the following structure:

- projectFiles
  - myLibrary
    - __init__.py
    - FiMOTS.py
    - FiMOTSFunctions.py
    - FPGrowth.py
    - prepareData.py
    - treeCreation.py
  - tests
    - OnlineRetail.py
    - OnlineRetailResults.py
    - T40.py
    - T40Results.py

[1] https://spark.apache.org/docs/latest/ml-guide.html
[2] https://spark.apache.org/docs/latest/ml-frequent-pattern-mining.html
[3] https://github.com/minrk/findspark
[4] https://matplotlib.org/

- Twitter.py
- TwitterResults.py

In the above, the files with myLibrary are functions used indirectly. Only the scripts within the "tests" directory are run directly.
A new folder, data, should be created within the projectFiles directory and data for the use case to be tested should be downloaded and placed here. The necessary data may be taken from:

- Twitter : https://archive.org/details/archiveteam-twitter-stream-2021-01
- T40 : http://fimi.uantwerpen.be/data/T40I10D100K.dat
- Online Retail : https://archive.ics.uci.edu/ml/datasets/online+retail

3 Updating File Paths

There are certain file paths which must be updated before running the tests. Depending on the test there is Python script which runs and generates the data and another which processes those results to generates graphs and metrics (e.g. OnlineRetail.py and OnlineRetailResults.py). Both files must be updated before running the full test.

In the test Python scripts:

- Line 3 contains a reference to the myLibrary directory. The path should be updated to the directory it has been placed in on the machine running the tests
- The "path" variable which points to the data should be updated in accordance with where the informed is stored after downloading in Section 2.
- The "outputResults", "outputStats" and "outputResultsFP" variables should all be updated to the location with which these outputs should be stored after the test completed

In the results Python scripts:

- At the start of the file all the input and output file paths must be updated to the location where the results were written to earlier, and where it is desired the graphs should be stored.

4 Running Tests

Provided python can access PySpark, the process for running the tests should be straightforward. Just call the script you want to run (either OnlineRetail.py, T40.py or Twitter.py) directly:

```
python3 tests/<filename>.py
```

As the time taken to cycle through the different multipliers can be lengthy, it is advisable to use nohup, e.g.:

```
nohup python3 tests/T40.py > T40_Random.log
```

This command allows the UNIX session being used for the test to be shut down without stopping the test from running and will output the standard output to a log file for review.

At the end of each iteration of the loop data will be written to standard output on the number of frequent items found for the sliding window.

Note that when running the T40 dataset, a parameter exists to vary whether to use a uniform or randomized time structure. This may be altered within the T40.py script by changing the Boolean *useRandom* parameter.

5 Processing Results

Once the tests on the dataset have completed a secondary step is required to compare the results against the baseline and generate graphs and metrics. As per Section 3, the locations specified in the results scripts should match those from the tests scripts otherwise they will be unable to find the data.

This script can be run using:

```
python3 tests/<filename>.py
```

For example:

```
Python3 tests/T40Results.py
```

It will result in graphs created in the folder specified and output to the screen of a table indicating the percentage reduction in time taken and number of checks across different values of the multiplier.