National College of Ireland

Configuration Manual

MSc Research Project
Programme Name

Mairead O'Doherty
Student ID: x20172826

School of Computing
National College of Ireland

Supervisor: Mohammed Hasanuzzaman

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Mairead O'Doherty
Student ID:	x20172826
Programme:	Programme Name
Year:	2022
Module:	MSc Research Project
Supervisor:	Mohammed Hasanuzzaman
Submission Due Date:	15/12/2022
Project Title:	Configuration Manual
Word Count:	608
Page Count:	5

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:	
Date:	1st February 2023

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

Mairead O'Doherty
x20172826

1 Data Preparation

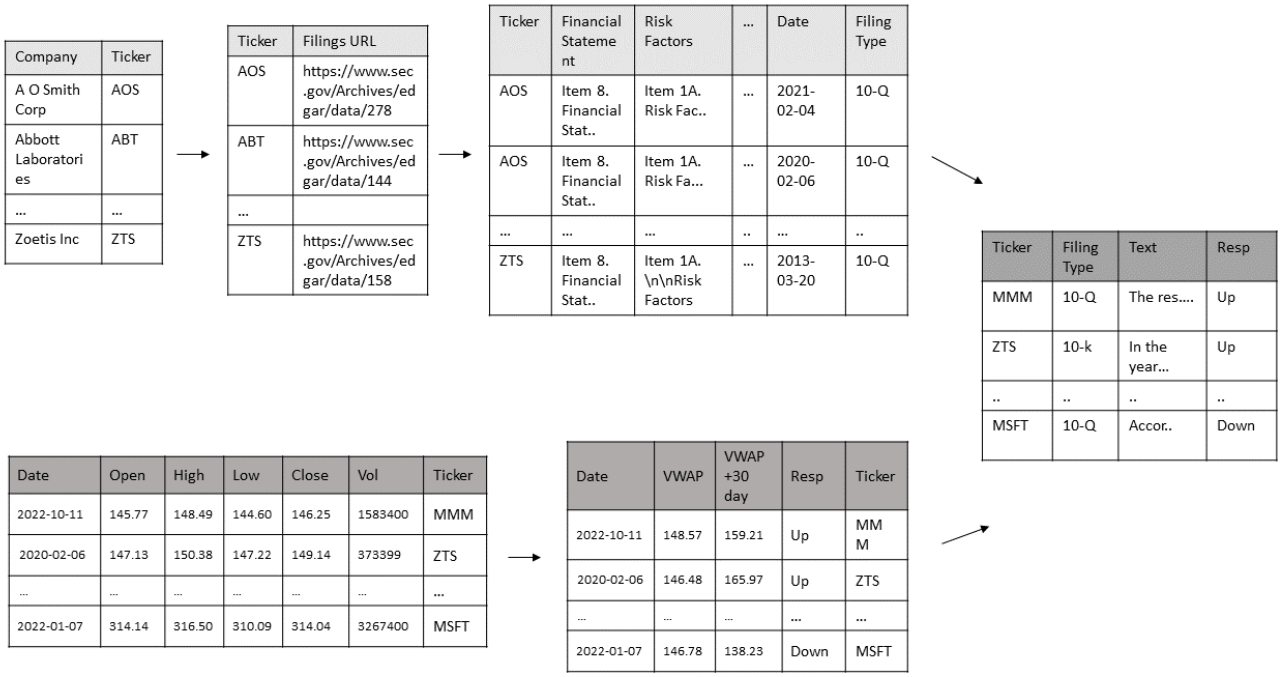

Figure 1: Formulation of the final dataset using SEC filings text and stock price data

1.1 Python Files

Python files for data preparation are in jupyter notebook format for clarity and visualising outputs. The process is split into individual files as there are a number of steps in data preparation as seen in Table 4. Several steps require dataframes for both 10Q and 10K filling types to be complete before moving on to the next step. Total run time to create the dataset is over 24 hours due to constraints on the number of hits per second to APIs at multiple stages in the process. This is another reason the process is split among several files, as it is easier to monitor process and not loose information if the process fails at any point. This process was done on a personal computer, device specifications are below in Figure 2.

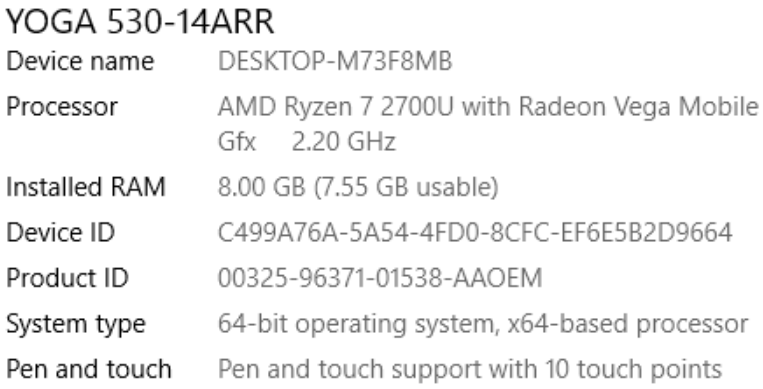

Figure 2: Device Specifications

Table 1: Dataset Preparation Python Files

File Name	**Description**
config.py	Configuration file for dataset, includes timeframes and API keys
get_links_for_filing.ipynb	Uses API to get list of filings within timeframe and corresponding URLS for each ticker
read_in_10k/10Q.ipynb	Takes URLs to filings and selects appropriate text segments, returning a dataframe with each each segment listed in a column. Separate files to deal with slight formatting differences between 10K and 10Q filling types
sec_pull_functions.py	Functions for use in reading in text segments
vwap_calculation.ipynb	Calculates VWAP value from stock information for each date.
add_response_10Q/10K.ipynb	Creates response variable by matching VWAP and VWAP value 30 days later. Calculates difference between values and determines response variable.
merge_final_dfs.ipynb	Merges 10Q and 10K data and cleans up dataset.

1.2 Python Packages

Python 3.8.3 was used in dataset creation. Python package versions not native to the python version are listed in table Table 3.

1.3 Data Sources and APIs

The Securities and Exchange Commision (SEC) operate an API where filings can be accessed. In Figure 3 there is a manual version of the API accessed by the sec api python package. The API can be accessed without using the specialised package however there is a higher limit on the number of hits per second using the sec-api package which sped up the process. Two API querys are accessed as to download the text, one to determine all the filings in the timeframe and their corresponding URL locations. Then these URLs are used as part of a query to the second API for the text output.

Table 2: Python Packages and Versions used in Data Preparations

Packages	Version	Site
Pandas	1.0.5	https://pypi.org/project/pandas/
Numpy	1.22.3	https://pypi.org/project/numpy/
Sec_api	1.0.12	https://pypi.org/project/sec-api/
Yfinance	0.1.77	https://pypi.org/project/yfinance/
ta	0.10.2	https://pypi.org/project/ta/

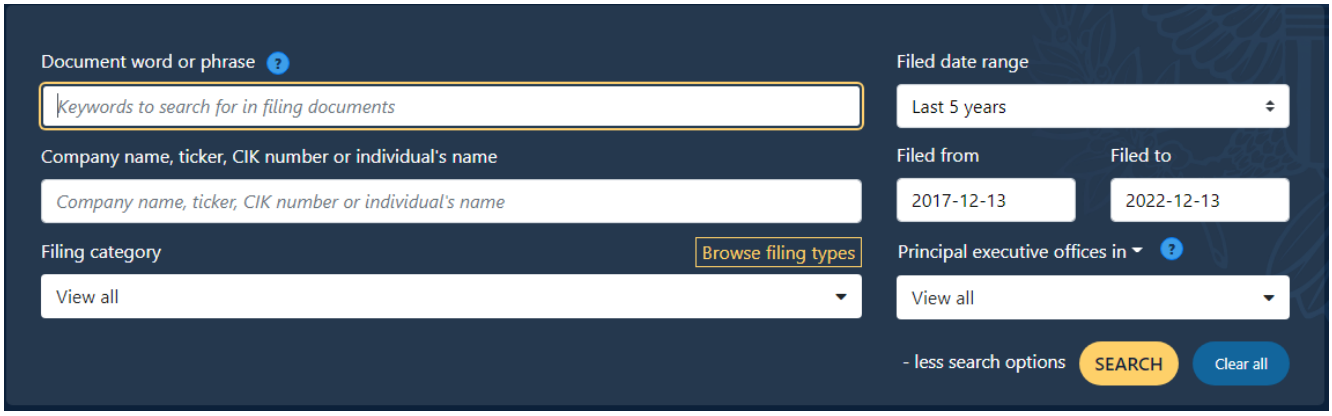

Figure 3: Manual EDGAR Search Portal

The Yahoo Fiance API is accessed through the yfiance python package which facilitates access to finance information made available by Yahoo.

2 Model Testing

Model creation and testing took place in the Google Colab Environment using a Pro account. This platform was necessary due to the need for a GPU to train the transformer model. The GPU on the machine used in the dataset preparation portion of this work could not be cuda enabled.

The work was carried out using ipython notebooks on the colab environment. Three algorithms were run on this platform, the TextRank algorithm, the XGBoost Classifier and the Longformer transformer model. Only the Longformer Model required the use of a GPU, specs are listed below in Figure 4 with appropriate CUDA version.

2.1 Python Packages

Python 3.8.3 was used in model testing. Python package versions not native to the python version are listed in table Table 3.

2.2 Python Files

The model testing was carried out in ipython notebook files for visualisation and ease of use. The TextRank algorithm is applied to the dataset in multiple chunks of 1000 rows.

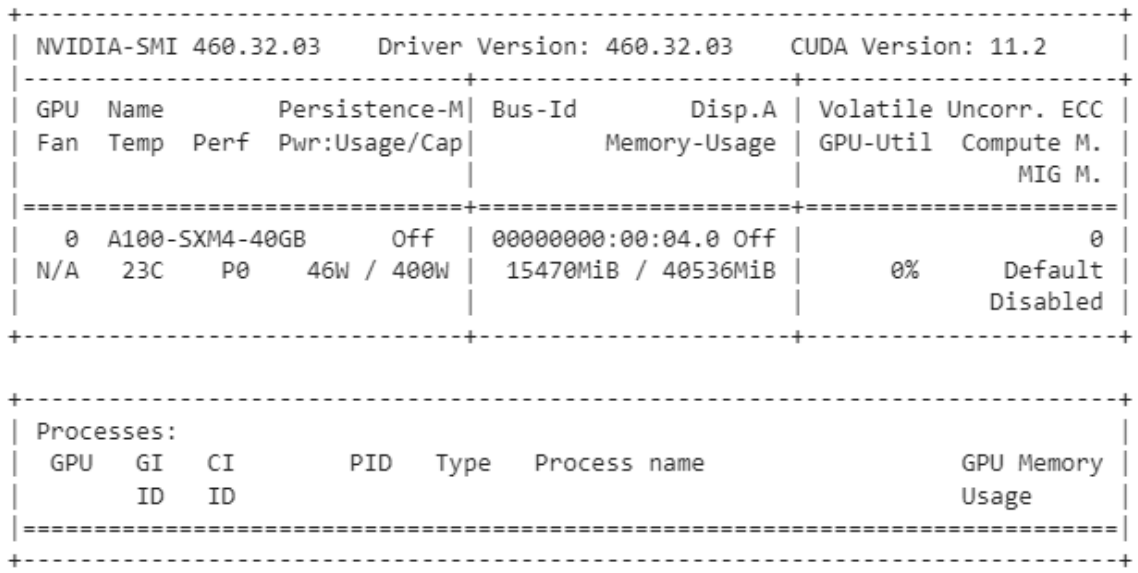

Figure 4: GPU Configuration on Google Colab

Table 3: Python Packages and Versions used in Model Testing

Packages	**Version**	**Site**
torch	1.13.0+cu116	https://pypi.org/project/torch/
transformers	4.25.1	https://pypi.org/project/transformers/
pandas	1.3.5	https://pypi.org/project/pandas/
numpy	1.21.6	https://pypi.org/project/numpy/
wandb	0.13.6	https://pypi.org/project/wandb/
scikitlearn	1.0.2	https://pypi.org/project/scikit-learn/
nltk	3.7	https://pypi.org/project/nltk//
xgboost	1.5.1	https://pypi.org/project/xgboost//

Each chunk takes up to 9 hours to complete and so this was done in a multistep process. The prioritized text is then concatinated from the chunks into a single dataframe. This is then used to create the final two models - the XGBoost Classifier and the Longformer model. The longformer model code was based on a similar multiclassification problem tutorial [1].

[1] https://jesusleal.io/2021/04/21/Longformer-multilabel-classification/

Table 4: Dataset Preparation Python Files

File Name	**Description**
sentence_ranking.ipynb	Apply TextRank Algorithm to Dataset in chunks. Multiday process.
combine_textrank.ipynb	Recombine the summarized text into a single dataframe
Longformer_w_textrank.ipynb	Creates and evaluates Longformer model. Requires CUDA enabled GPU environment.
XGBoost.ipynb	Creates and evaluates XGBoost Classifier model