**Trump Tweets vs The Markets**

Final Report for CS39440 Major Project

*Author*: Mateusz Stankiewicz ([mas15@aber.ac.uk](mailto:mas15@aber.ac.uk))

*Supervisor*: Dr. Neil Mac Parthaláin (ncm@aber.ac.uk)

26th April 2018

Version 1.0 (Release)

This report is submitted as partial fulfilment of a BSc degree in  
Computer Science (G401)

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Wales, UK

**Declaration of originality**

I confirm that:

* This submission is my own work, except where clearly indicated.
* I understand that there are severe penalties for Unacceptable Academic Practice, which can lead to loss of marks or even the withholding of a degree.
* I have read the regulations on Unacceptable Academic Practice from the University’s Academic Quality and Records Office (AQRO) and the relevant sections of the current Student Handbook of the Department of Computer Science.
* In submitting this work I understand and agree to abide by the University’s regulations governing these issues.

Name …………………………………………

Date ……………………………………………

**Consent to share this work**

By including my name below, I hereby agree to this dissertation being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Name …………………………………………

Date ……………………………………………

**Acknowledgements**

I am grateful to…

I’d like to thank StackOverflow for my Degree.

**Abstract**

Include an abstract for your project. This should be no more than 300 words.

“World events often have a great influence over international markets. Political uncertainty can often drive commodities up or down in value depending on where it occurs in the world. “[1]. Politicians of countries with the biggest markets have got a strong impact on the value of currencies and trading commodities. A simple message such as "With Mexico being one of the highest crime Nations in the world, we must have THE WALL. Mexico will pay for it…"[2] sent from the USA president's account can drop down Mexican Peso value. Trump’s infamous Twitter account, which is followed by almost 50 million people, can be an effective tool to influence the markets.

The goal of this project is to develop a system which considers the sentiment of tweets and can predict whether a stock index will increase or decrease depending on the current index, words, phrases and the sentiment of the tweet.

Python, the Scikit-learn and NLTK (Natural Language Toolkit) are used to process the data and Python Flask framework is used to create a web interface.

The results show that when using a Naïve Bayes classifier, the accuracy of predicting the USD Index change is 53.6%, whereas the base rate of the three-class problem (up, down, no change) is 41%.

**Contents**

1. Background, Analysis & Process 3

1.1. Background 3

1.2. Analysis 3

1.3. Process 3

2. Design 3

2.1. Overall Architecture 3

2.2. Detailed Design 3

2.2.1. Even More Detail 3

2.3. User Interface Design 3

2.4. Other Relevant Sections 3

3. Implementation 3

4. Testing 3

4.1. Overall Approach to Testing 3

4.2. Automated Testing 3

4.2.1. Unit Tests 3

4.2.2. User Interface Testing 3

4.2.3. Stress Testing 3

4.2.4. Other Types of Testing 3

4.3. Integration Testing 3

4.4. User Testing 3

5. Critical Evaluation 3

6. Appendices 3

A. Third-Party Code and Libraries 3

B. Ethics Submission 3

C. Code Samples 3

Annotated Bibliography 3

# Background, Analysis & Process

This section should discuss your preparation for the project, including background reading, your analysis of the problem and the process or method you have followed to help structure your work. It is likely that you will reuse part of your outline project specification, but at this point in the project you should have more to talk about.

**Notes:**

* All of the sections and text in this example are for illustration purposes. The main Chapters are a good starting point, but the content and actual sections that you include are likely to be different.
* Look at the document on the MMP: Final Report and Technical Work [7] for additional guidance.

## Background

What was your background preparation for the project? What similar systems did you assess? What was your motivation and interest in this project?

## Analysis

Taking into account the problem and what you learned from the background work, what was your analysis of the problem? How did your analysis help to decompose the problem into the main tasks that you would undertake? Were there alternative approaches? Why did you choose one approach compared to the alternatives?

There should be a clear statement of the objectives of the work, which you will evaluate at the end of the work.

In most cases, the agreed objectives or requirements will be the result of a compromise between what would ideally have been produced and what was felt to be possible in the time available. A discussion of the process of arriving at the final list is usually appropriate.

As mentioned in the lectures, think about possible security issues for the project topic. Whilst these might not be relevant for all projects, do consider if there are relevant for your project. Where there are relevant security issues, discuss how they will this affect the work that you are doing. Carry forward this discussion into relevant areas for design, implementation and testing.

## Process

You need to describe briefly the life cycle model or research method that you used. You do not need to write about all of the different process models that you are aware of. Focus on the process model that you have used. It is possible that you needed to adapt an existing process model to suit your project; clearly identify what you used and how you adapted it for your needs.

# Design

You should concentrate on the more important aspects of the design. It is essential that an overview is presented before going into detail. As well as describing the design adopted it must also explain what other designs were considered and why they were rejected.

The design should describe what you expected to do, and might also explain areas that you had to revise after some investigation.

Typically, for an object-oriented design, the discussion will focus on the choice of objects and classes and the allocation of methods to classes. The use made of reusable components should be described and their source referenced. Particularly important decisions concerning data structures usually affect the architecture of a system and so should be described here.

How much material you include on detailed design and implementation will depend very much on the nature of the project. It should not be padded out. Think about the significant aspects of your system. For example, describe the design of the user interface if it is a critical aspect of your system, or provide detail about methods and data structures that are not trivial. Do not spend time on long lists of trivial items and repetitive descriptions. If in doubt about what is appropriate, speak to your supervisor.

You should also identify any support tools that you used. You should discuss your choice of implementation tools - programming language, compilers, database management system, program development environment, etc.

Some example sub-sections may be as follows, but the specific sections are for you to define.

## Overall Architecture

### 2.1.1 Tools and Technologies

Very important part of staring the project is to select programming language and libraries that will be used. More about blabla

2.1.1.1 Programming Language

When choosing a programming language, I focused on its usability for the project, its libraries, ease of use and experience. The language should provide:

* Machine learning and natural language processing libraries
* Simple Web framework
* Ease of experimenting
* Simple tools to gather the data from web APIs
* Simple tools to do data manipulations and analysis

I think that Python is the perfect choice in this case. It provides very good ML languages (NLTK, Scikit-learn, Textblob), web frameworks (Flask, Django, Pyramid) and Interpreter which is useful to do quick, ad hoc experimenting. It also provides a Pandas library that allows to form data into DataFrames what is very handy in data analysis and provides many build-in data processing functions. I was also programming in this language during my Industrial Year and gained a lot of experience. Python is also very popular and has got documentation and SOCIETY?

Other languages that I took into account are:

* **R** - has good ML and data processing libraries but has not got any libraries providing a way to create a web interface.
* **Ruby** – provides good web framework (Ruby on Rails) but does not provide good libraries/gems to do ML. There is Weka for JRuby or other ways to use Java Weka library in Ruby.
* **Java** – Provides many web framworks such as Spring, JSF or Vaadin. XX also can use Weka (Waikato Environment for Knowledge Analysis) that is a very popular software suite.

2.1.1.2 Libraries

To do a language processing I used firslty a Textblob that is build on top of NLTK but it very poorly documented and NIE DAWAL TKAIEJ swobody so I had to move into NLTK.

To build a classificator which predicts market changes I used Scikit-learn. I find this library very easy to use. It has got good documentation with many examples and there is also many topics about it on sites such as StackOverflow so BLBLALB

Other possibility would be to use Weka but it would require to use either Jython(implementation of the Python language for the Java platform[3]) or run weka library using wrappers around JNI calls such as javabridge(package that allows Python to interact with JVM[4]) but both ways are too complicated.

The web interface of the program is very simple so I chosen Flask. It is a micro web framework using Jinja2 template engine. It allows to create simple pages in very easy way. I did not use Django because it is rather better for more complex webpages. Pyramid is also good for creating simple webpages but is much less popular, what makes it harder to find some solutions ETC in the Internet.

I also used Mlxtend library to sue Apriori algorithm, Nose to do unittests and Flask-SQLAlchemy to quickly create a simple database in Flask.

To do data processing I also used Pandas and NumPy which are the most popular Python libraries providing …BLLBLB

To scrap tweets from Twitter API I used a Tweepy library which is the most popular and very simple.

## Detailed Design

The project was designed in Object-oriented way??? And most of the functionalities ZXX in classes. Because the project can be split into two parts: market analisys and web interface I split the project into modules:

* Markets module
* Webpage Module
* Tests Module

2.2.1 Markets module

TweeterScrapper – a class used to scrap tweets from Tweeter.

FeatureExtractor – a class responsible for extracting features from tweets. Features are phrases and words that are important for the analisys. To extract features, an object of this class must have set features by the user or has to build the vocabulary from a dataset.

SentimentAnalyser – a class responsible for tweets sentiment analysis. It wraps NaiveBayesClassifier from NLTK library and uses a FeatureExtractor to extract features from tweets which are then used to train a model or predict a value of the particular tweet.

FeatureSelector – a small class used to do feature selection using MultinomialNB and RFECV from Scikit-learn. It ranks features and selects user-defined k number of features.

CurrencyAnalyser – a class that wraps wraps XXx functionality. It analyses a CSV file of stock prices and provides results of this analys such as assosiation rules, model to predict markets or the most coefficient features.It holds the functionality of reading files and saving the results.

AssociationDataProcessor – a class used to proces a dataframe. It removes XXXX

MarketPredictingModel – a class that wraps a classifier model (MultinomialNB by default). It has to be trained or loaded from file to analyse a tweet and predict the market change.

ModelTrainer – a class used to train MarketPredictingModel with different features to obtain the best accuracy. It trains model many times with different seeds that are used by random number generator to produce different folds in cross validation.

ModelTrainingResult - a class that holds a training result such as accuracy, base rate, used dataframe, features and number of tweets. It provides comparision operators so that results can be compared easily in XXX way. For example If two results have got the same accuracy then the number of features is compared and the difference by baseline BLBLBL

The markets module has got also a “data” and “pickled\_models” direcotries. “data” folder stores all data used to do analysis such as a list of stopwords, scrapped tweets and CSV files generated by the application. “pickled\_models” stores saved sentiment anlisys and market predicting models.

2.2.2 Webpage module

Webpage module has got a typical structure for Flask projects.

It is split into:

* Static folder – holding static files such as images or css styles
* Templates folder – holding Jinja2 templates that are filled with content by views.
* Views.py – this is where routes are defined. It has got routes for each curriency BLBLA
* models.py – holds Currency model. This model stores information about currencies such as its name and accuracy of the model in database.
* \_\_init\_\_.py – Initializes the application, sets up its configuration and database.
* Currencies.db – stores Currencies models that can be loaded when application runs.

## User Interface Design

## Tools used to develop the project

For my Python IDE I chose PyCharm made by JetBrains company. I used it during my Industiral year and I really liked it. It has got all code assistace features such as syntax and error highlighting. It supports Flask projects and many file extensions such as html, css, js, csv which I used in a project. It also has got integrated debugger which I used a lot.

Github and GitKraken

To z meila

Do you have backups of your technical work and your reports? More importantly, do you have backups of your technical work and your reports on a different machine from the one you normally develop on? Better still, are those backups on machines in different buildings?

# Implementation

The implementation should look at any issues you encountered as you tried to implement your design. During the work, you might have found that elements of your design were unnecessary or overly complex; perhaps third party libraries were available that simplified some of the functions that you intended to implement. If things were easier in some areas, then how did you adapt your project to take account of your findings?

It is more likely that things were more complex than you first thought. In particular, were there any problems or difficulties that you found during implementation that you had to address? Did such problems simply delay you or were they more significant?

You can conclude this section by reviewing the end of the implementation stage against the planned requirements.

# Testing

Detailed descriptions of every test case are definitely not what is required here. What is important is to show that you adopted a sensible strategy that was, in principle, capable of testing the system adequately even if you did not have the time to test the system fully.

Provide information in the body of your report and the appendix to explain the testing that has been performed. How does this testing address the requirements and design for the project?

How comprehensive is the testing within the constraints of the project? Are you testing the normal working behaviour? Are you testing the exceptional behaviour, e.g. error conditions? Are you testing security issues if they are relevant for your project?

Have you tested your system on “real users”? For example, if your system is supposed to solve a problem for a business, then it would be appropriate to present your approach to involve the users in the testing process and to record the results that you obtained. Depending on the level of detail, it is likely that you would put any detailed results in an appendix.

The following sections indicate some areas you might include. Other sections may be more appropriate to your project.

## Overall Approach to Testing

## Automated Testing

### Unit Tests

### User Interface Testing

### Stress Testing

### Other Types of Testing

## Integration Testing

## User Testing

# Critical Evaluation

Examiners expect to find in your dissertation a section addressing such questions as:

* Were the requirements correctly identified?
* Were the design decisions correct?
* Could a more suitable set of tools have been chosen?
* How well did the software meet the needs of those who were expecting to use it?
* How well were any other project aims achieved?
* If you were starting again, what would you do differently?

Other questions can be addressed as appropriate for a project.

Such material is regarded as an important part of the dissertation; it should demonstrate that you are capable not only of carrying out a piece of work but also of thinking critically about how you did it and how you might have done it better. This is seen as an important part of an honours degree.

There will be good things and room for improvement with any project. As you write this section, identify and discuss the parts of the work that went well and also consider ways in which the work could be improved.

In the latter stages of the module, we will discuss the evaluation. That will probably be around week 9, although that differs each year.

# Appendices

The appendices are for additional content that is useful to support the discussion in the report. It is material that is not necessarily needed in the body of the report, but its inclusion in the appendices makes it easy to access.

For example, if you have developed a Design Specification document as part of a plan-driven approach for the project, then it would be appropriate to include that document as an appendix. In the body of your report you would highlight the most interesting aspects of the design, referring your reader to the full specification for further detail.

If you have taken an agile approach to developing the project, then you may be less likely to have developed a full requirements specification. Perhaps you use stories to keep track of the functionality and the ’future conversations’. It might not be relevant to include all of those in the body of your report. Instead, you might include those in an appendix.

There is a balance to be struck between what is relevant to include in the body of your report and whether additional supporting evidence is appropriate in the appendices. Speak to your supervisor or the module coordinator if you have questions about this.

* 1. Third-Party Code and Libraries

If you have made use of any third party code or software libraries, i.e. any code that you have not designed and written yourself, then you must include this appendix.

As has been said in lectures, it is acceptable and likely that you will make use of third-party code and software libraries. If third party code or libraries are used, your work will build on that to produce notable new work. The key requirement is that we understand what is your original work and what work is based on that of other people.

Therefore, you need to clearly state what you have used and where the original material can be found. Also, if you have made any changes to the original versions, you must explain what you have changed.

As an example, you might include a definition such as:

**Apache POI library** – The project has been used to read and write Microsoft Excel files (XLS) as part of the interaction with the client’s existing system for processing data. Version 3.10-FINAL was used. The library is open source and it is available from the Apache Software Foundation [5]. The library is released using the Apache License [6]. This library was used without modification.

* 1. Ethics Submission

This appendix includes a copy of the ethics submission for the project. After you have completed your Ethics submission, you will receive a PDF with a summary of the comments. That document should be embedded in this report, either as images, an embedded PDF or as copied text. The content should also include the Ethics Application Number that you receive.

* 1. Code Samples

This is an example appendix. Include as many appendices as you need. The appendices do not count towards the overall word count for the report.

For some projects, it might be relevant to include some code extracts in an appendix. You are not expected to put all of your code here - the correct place for all of your code is in the technical submission that is made in addition to the Final Report. However, if there are some notable aspects of the code that you discuss, including that in an appendix might be useful to make it easier for your readers to access.

As a general guide, if you are discussing short extracts of code then you are advised to include such code in the body of the report. If there is a longer extract that is relevant, then you might include it as shown in the following section.

Only include code in the appendix if that code is discussed and referred to in the body of the report.

Random Number Generator

The Bayes Durham Shuffle ensures that the pseudo random numbers used in the simulation are further shuffled, ensuring minimal correlation between subsequent random outputs.

// Some example code here…

# Annotated Bibliography

This final section should list all relevant resources that you have consulted in researching your project. Each reference should also include a brief annotation.

1. Neil Mac Parthaláin, “MMP: Project descriptions”, 2018 (Online) Available at: https://teaching.dcs.aber.ac.uk/mmp Accessed April 2018.

2. <https://twitter.com/realdonaldtrump/status/901802524981817344>, 27th Febuary 2017. Accessed April 2018.

This is of Donald Trumps’ Tweets.

3. “What is Jython?”, Jython webpage <http://www.jython.org/archive/21/docs/whatis.html> Accessed April 2018POCZ

4. https://pypi.python.org/pypi/javabridge/1.0.17

1. Sylvia Duckworth. A picture of a kitten at Hellifield Peel. <http://www.geograph.org.uk/photo/640959>, 2007. Copyright Sylvia Duckworth and licensed for reuse under a Creative Commons Attribution-Share Alike 2.0 Generic Licence. Accessed August 2011.  
     
   This is my annotation. I should add in a description here.
2. Mark Neal, Jan Feyereisl, Rosario Rascunà, and Xiaolei Wang. Don’t touch me, I’m fine: Robot autonomy using an artificial innate immune system. In *Proceedings of the 5th International Conference on Artificial Immune Systems*, pages 349–361. Springer, 2006.   
     
   This paper…
3. W.H. Press et al. *Numerical recipes in C*. Cambridge University Press Cambridge, 1992.  
     
   This is my annotation. I can add in comments that are in **bold** and *italics*and then further content.
4. Various. Fail blog. <http://www.failblog.org/>, August 2011. Accessed August 2011.  
     
   This is my annotation. I should add in a description here.
5. Apache Software Foundation (2014) “*Apache POI - the Java API for Microsoft Documents*” (Online) Available at: <http://poi.apache.org> Accessed: 14th March 2014.
6. Apache Software Foundation (2004) “Apache License, Version 2.0” (Online) Available at: <http://www.apache.org/licenses/LICENSE-2.0> Accessed: 14th March 2014.
7. Neil Taylor, “MMP: Final Report and Technical Work”, 2017 (Online) Available at: <http://blackboard.aber.ac.uk/> Accessed 26th April 2017.

A document that outlines information about the marking guide for the Final Report and Technical Work. This document was referred to as Structure of the Final Report before academic year 2016-2017. This is published in the Assignments folder. If you are logged in to Blackboard, you can access the folder using <http://jump.aber.ac.uk/?sxxpt>.