

**The Truth about Twitter**

Final Year Project Report

*DT228 B.Sc. (Hons) in Computer Science*

**Max MacDonald**

**C15740661**

**Cindy Liu**

*School of Computing*

*TUD*

**11-04-2019**



# Abstract

The rise of social media platforms in the last few decades have led to the surge of numerous types of fake accounts across them. These platforms have struggled to keep up with the massive volume of nefarious fake accounts being created everyday due largely in part to how hard it can be to identify them as many are created to masquerade efficiently as human coupled with said platforms initial disinterest in dealing with such a problem. As pressure from their user base and governments mount though many are now actively pursuing the agenda to cleanse their platforms of this taint.

The purpose of this report is to show how data mining and machine learning algorithms can be used to deal with the issue of correctly identifying whether a Twitter account is real or a form of fake account. Machine learning models are created and used in a simple web application where a user can check any and as many accounts as the wish to see what type of account the models predict it to be. They are then able to share their results on different social media platforms and give feedback on the web application.

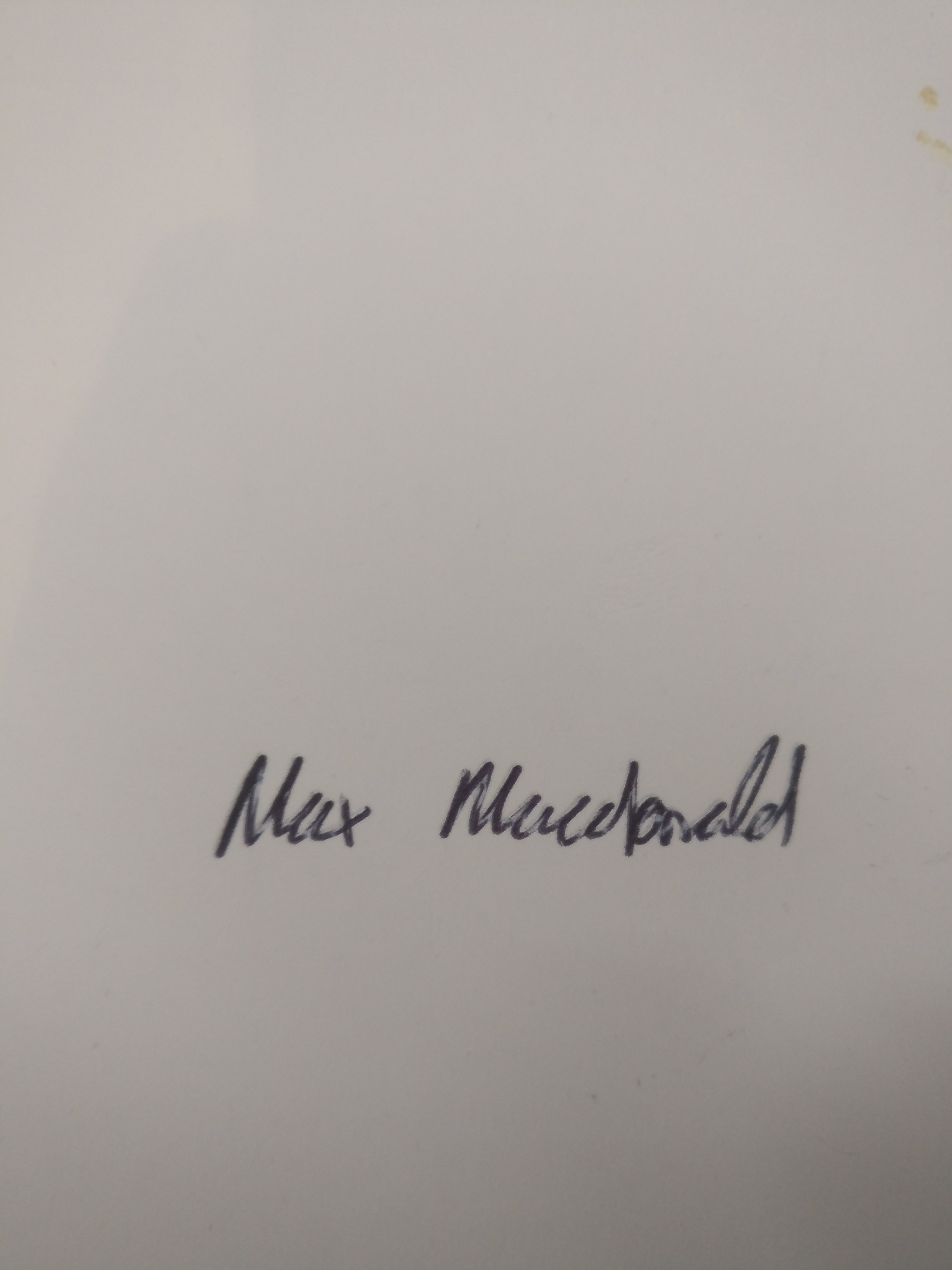
The report starts off by explaining the background of the project including description, objectives and roadmap and all the research that has gone into it. Next, the design and development stages are delved into in detail with the various decisions made during creation of the web application explained. Evaluation of the experiments done on the datasets to find the most accurate classifier, test and train data split technique and parameters and the various forms of testing on the website are discussed. Lastly all conclusions drawn from the report and any future work planned is reviewed.

Keywords: Data mining, Machine learning, Twitter, Fake accounts

# Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

**Signed:**



\_\_\_\_\_\_\_\_\_\_\_

Max MacDonald

11/04/2018

# Acknowledgements

I would like to thank my supervisor, Cindy Liu, who has helped me immeasurably throughout the project, always having time to meet me and offer feedback at every stage.

To my parents, I cannot possibly express my gratitude to you for always being there for me no matter what else is going on and especially for these last four years while undertaking this undergraduate degree.

To my thanks to my friends and family, you all made it possible for me to step away from college work when it got tough, recharge and unwind and come back to my work with renewed focus and vigour.

Lastly, I would like to express my thanks to all those who took part in the testing of my application and provided such great feedback.

**Table of Contents**

1. Project Statement 5

2. Research 5

2.1. Background Research 5

2.1.1. What is a Twitter Bot Account? 6

2.1.2. Types of Bots Accounts 6

2.1.3. Importance of Identifying Bots Accounts 6

2.1.4. Important Characteristics of Bot Accounts 7

2.2. Alternative Existing Solutions to Your Problem 8

2.2.1. Twitter 8

2.2.2. Botometer 8

2.3. Technologies Researched 9

2.3.1. Technologies for Data Mining & Machine Learning Models 10

2.3.2. Technologies for Web Application 12

2.3.3. Technologies for Web Server 13

2.3.4. Technologies for Version Control 14

2.3.5. Technologies for Data Storage 14

2.4. Other Relevant Research Done 15

2.4.1. Big Data 15

2.4.2. Data Mining 16

2.4.3. Machine Learning 17

2.5. Resultant Findings and Requirements 22

2.5.1. Chosen Technologies 22

2.5.2. Chosen DataSets 22

2.5.3. Challenges 23

3. Approach and Methodology 24

3.1. Agile & Kanban 24

3.1.1. Agile 24

3.1.2. Kanban 25

3.1.3. Project Use 25

3.2. Data Mining Project Management Models 26

3.2.1. CRISP-DM 26

3.2.2. SEMMA 28

3.2.3. Differences in Models 30

3.2.4. Conclusion 30

4. Design 31

4.1. Technical Architectures 31

4.1.1. Model View Controller 31

4.1.2. Model View Template 31

4.2. Technical Architecture Diagram 32

4.3. Other Design Documents 33

4.3.1. Use Case Diagram 33

4.3.2. Entity Relationship Diagram 33

5. Prototyping and Development 34

5.1. Vertical Prototype 34

5.1.1. Web Front-End 34

5.1.2. Basic Machine-Learning Model 35

5.2. Development 35

5.2.1. Web Front-End 35

5.2.2. Basic Machine-Learning Model 38

6. Testing 40

6.1. Data Mining & Machine Learning section 40

6.2. Web Front-End 41

6.3. Web-based Data Mining Application 41

7. Issues and Risks 41

8. Plan and Future Work 42

9. Bibliography 43

**Table of Figures**

[Figure 1 Botometer GUI [13] 9](#_Toc531620765)

[Figure 2 R code example 10](#_Toc531620766)

[Figure 3 Python code example 10](#_Toc531620767)

[Figure 4 PyCharm IDE 11](#_Toc531620768)

[Figure 5 Scikit-learn code example 11](#_Toc531620769)

[Figure 6 Pandas code example 12](#_Toc531620770)

[Figure 7 NumPy code example 12](#_Toc531620771)

[Figure 8 12 Agile Principles 24](#_Toc531620772)

[Figure 9 Sample Trello Board [40] 26](#_Toc531620773)

[Figure 10 CRISP-DM Model [41] 27](#_Toc531620774)

[Figure 11 SEMMA Model [43] 29](#_Toc531620775)

[Figure 12 Model View Controller 31](#_Toc531620776)

[Figure 13 Model View Template 32](#_Toc531620777)

[Figure 14 Application Technical Architecture 32](#_Toc531620778)

[Figure 15 User Use Case Diagram 33](#_Toc531620779)

[Figure 16 Entity Relationship Diagram 33](#_Toc531620780)

[Figure 17 Web Front End 34](#_Toc531620781)

[Figure 18 Web Front End with returned tweets 34](#_Toc531620782)

[Figure 19 Results of Model being run 35](#_Toc531620783)

[Figure 20 Saving Twitter credentials to file 36](#_Toc531620784)

[Figure 21 Basic form 36](#_Toc531620785)

[Figure 22 Creating or reading in form 36](#_Toc531620786)

[Figure 23 Twitter Authentication and tweet retrieval 37](#_Toc531620787)

[Figure 24 Render Index.html with variables passed 37](#_Toc531620788)

[Figure 25 Template directory added 37](#_Toc531620789)

[Figure 26 Index.html 37](#_Toc531620790)

[Figure 27 Load web\_style.css 37](#_Toc531620791)

[Figure 28 Enabling CSS file load part 1 38](#_Toc531620792)

[Figure 29 Enabling CSS file load part 2 38](#_Toc531620793)

[Figure 30 Enabling Django file use 38](#_Toc531620794)

[Figure 31 Read in all data 39](#_Toc531620795)

[Figure 32 Read in single dataset from CSV 39](#_Toc531620796)

[Figure 33 Data check example 39](#_Toc531620797)

[Figure 34 Adding data to database 39](#_Toc531620798)

[Figure 35 Get 2000 accounts 39](#_Toc531620799)

[Figure 36 Return random sub-set of accounts from database 40](#_Toc531620800)

[Figure 37 Convert arrays and initialise model 40](#_Toc531620801)

[Figure 38 Run model and output results to screen 40](#_Toc531620802)

[Figure 39 Project Gantt chart 42](#_Toc531620803)

# Introduction

## Project Background

Twitter, a free social networking microblogging service, is one of today’s leading digital platforms with 326 million active users worldwide in the third quarter of 2018. [1] Registered users can broadcast short posts called tweets which can be liked, retweeted and responded to by other users. Just like other social media platforms such as Facebook and Instagram, Twitter has and still is facing a massive problem with fake or bot accounts. Estimates place the percentage of bot accounts on Twitter anywhere from 9 to 15% of the total user count. [2]

### What is a Twitter Bot Account?

A Twitter bot account is an account that is controlled by a software application, via the Twitter API, which will automatically generate and publish new tweets, follow specific users, retweet other tweets and liking specific sets of tweets all based on content or hashtags included, depending on the settings of the controlling application. [2]

These bots can perform tasks at a much higher rate than a human user can and as such push out more content or tweets in the same timeframe, some even working around the clock. Bot accounts on other platforms are similar with any differences being based on the difference in platforms.

### Types of Bot Accounts

There are many different types of bot accounts from helpful and informative ones tweeting spiritual wellbeing tips to ones which retweet tweets that push extreme ideologies to advertising accounts which are set up to tweet content about specific brands, products or services at certain times of the day.

Some accounts are even used to boost a person’s fame or influence on Twitter by following that person’s account and can be bought in packages. This is a massive industry in and of itself reportedly being a $40 to $360 million-dollar business annually. [3] Major celebrities such as 50 Cent and brands like Mercedes-Benz have come under scrutiny for possibly engaging in this practice. [4]

Then there is the complexity of the software applications behind these accounts. Older and more traditional bot accounts tend to be easier to detect as they follow much simpler patterns in their activities while newer social bots need far more complex algorithms to detect as they are set up to masquerade effectively as human accounts by mimicking human behaviour better.

### Importance of Identifying Bot Accounts

The main reason why it is so important to be able to identify, unless the account states so itself, whether an account is real or not is the erosion of trust that can occur due to the account’s activities. If the account is followed by one million other accounts, even if most of them are fake themselves, and posts something that, while untrue, pushes a narrative that certain groups would be inclined to believe then that post can gain a lot of traction and spread quickly all over Twitter and beyond to other social media platforms causing untold damage.

Many individuals or groups wish to affect the perception of specific events or entities through Twitter and this ranges from boosting their own profiles through fake followers as mentioned above to trying to influence public campaigns such as the 2016 US Presidential elections. Studies have estimated that in the lead up to this election, a fifth of all Twitter traffic related to the election came from a legion of bots. [5] That much traffic would have had a massive influence on people’s views and how they voted and in turn the outcome of the election.

If a bot account is masquerading as a real human then, since it is inherently trying to deceive us, it is highly unlikely much good can come of its sustained existence and as such the sooner it is detected and shutdown the better.

### Important Characteristics of Bot accounts

When trying to identify if an account is a bot or not there are some key characteristics that can help [6,7]:

* How often per day an account tweets and if there is a regular interval between tweets can both lead to suspicions as this is a hallmark of automation.
* How anonymous that account is trying to be, does it have a profile picture and if so is it of a person? Same for the background picture. Does their bio information help identify them or add to their anonymity? Is the account handle just an alphanumeric scramble?
* Generic bio or lack of one as the programs which create these bots are not set up to make completely unique bios.
* Ratio of how many other accounts an account follows to how many follow it.

## Project Description

The focus of this project is to be able to differentiate between real and bot Twitter accounts and allow the public to make use of this ability. This will require a deep understanding into what constitutes each of the various types of bot accounts, research into the areas of data mining and machine learning, learning how to work towards the required insights from a procured dataset and lastly to create and deploy a web application. There are a lot of new subjects to learn about and interesting challenges to overcome that are detailed later.

## Project Objectives

The main goal of this project is first and foremost to create accurate mathematical models, employing various data mining and machine learning techniques and classifiers, that can tell a real Twitter account from a fake one. The secondary goal is to create a simple web application that makes use of these models to allow users to get a prediction for any Twitter account they wish.

To be able to achieve both goals, the following list of objectives must be completed:

1. Research into what makes a fake Twitter account and alternative solutions.
2. Research data mining, machine learning, the various technologies involved and Project Management and Data Mining methodologies.
3. Gather historical Twitter user account data that has both real and fake accounts.
4. Understand, clean and prepare gathered data.
5. Design the overall project architecture.
6. Design a database to store data and read the prepared data into it.
7. Experiment with the data and various classifiers until an acceptable level of accuracy is achieved.
8. Develop the web application to make use of the results from the previous part.
9. Deploy the web application onto the server.
10. Perform usability and \* testing to gain user feedback and ensure everything works as intended.

## Project Challenges

With all the main objectives set out for project set out, there will invariably be many challenges along the way in completion of these. Here are the main challenges that could be encountered:

### Time

Time management for this entire project will be a major challenge as the year’s modules, their workloads and exams must all be juggled with this project. While everything has been planned out into a project plan and tweaked as progress is made, unexpected situations can arise within college and in outside life which can throw specific pieces of work’s timelines off.

### Data Preparation and Feature Selection

One of the longest and most important parts to any data mining project is preparing the data and then selecting the right features for use in training models. Any errors at either of these stages will lead to inaccurate predictions thereby losing users trust and interest in using the web application. The data must be explored carefully, and time invested in understanding how to prepare the data correctly and what features are important, and which are not.

### Accuracy of models

To be able to create accurate models requires that the previous challenge be overcome and then being to understand the various machine learning classifiers and how they arrive at their conclusions. This will require a significant portion of time dedicated to research and discerning between the different classifiers.

This challenge is the most important aspect that the success of the project relies on as without an accurate model the web application holds little value to the user as it cannot make accurate predictions.

### Deployment of Web Application

The deployment of the web application onto a server is potentially a complicated challenge due to a lack of experience in this area. Each provider of deployment servers has their own steps needed to be taken and settings altered to ensure a web application can be deployed smoothly and correctly. To overcome this challenge will very much be a process of trail and error and reading through tutorials.

## Dissertation Structure

***Chapter 1: Introduction***

An Introduction to the project background, a description of what it entails, a list of all major objectives requiring completion and what are the main set of challenges that will be faced to achieve completion.

***Chapter 2: Research***

Covers all research done before and throughout the design and development of the project. Alternative solutions to the core issue and similar past projects are investigated and detailed. Various relevant technologies are analysed, and an explanation given on why specific ones were chosen. Lastly the areas of data mining and machine learning are delved into, explaining the various techniques used.

***Chapter 3: Approach & Methodology***

The Agile and Kanban project management methodologies are explained and shown how they are used within this project. The CRISP-DM and SEMMA data mining methodologies are gone into in detail and compared with a conclusion on why one was chosen over the other.

***Chapter 4: Design & Architecture***

Details the various design elements of the project including the general technical architecture and how it was adapted for use, how the data mining and web application link together and discussions on Use Cases, Entity Relationship and Class Diagrams.

***Chapter 5: Prototyping & Development***

Shows what prototyping was done and goes into depth on all the development done for both parts of the project, describing how accurate machine learning models were created through the various stages of the CRISP-DM methodology and the construction of a website to make use of them such that accurate predictions are returned to the user.

***Chapter 6:*** ***Project Plan***

Gives the overall project plan, how this is broken down into various Kanban boards and the fine tuning done to it through engagement with my supervisor.

***Chapter 7: Testing and Evaluation***

Explains how the separate parts of the system were tested and evaluated. Usability and \* testing for the web application and what the results of these were. For the data mining side of the project, this revolved around the evaluation of various machine learning classifiers across multiple different output combinations. Various accuracy metrics are computed and explored leading to the final model/s selection used in the web application.

***Chapter 8: Conclusion and Future Work***

Wraps up the project, elaborating on the key points learned while undertaking this project, the parts that did and didn’t live up to expectation and what additional work could be done going forward.

***Chapter 9: Bibliography***

Details all references used throughout the document. Websites, books, academic papers and studies are all included.

# Research

This section will cover all research done for this project including applications or solutions akin this project, other final year projects like this, all technologies researched for this project, research into data science and its sub topics big data, data mining and machine learning and finally the results of all this research such as chosen technologies and areas that will prove challenging.

## Alternative Existing Solutions to the Problem

This section explains how Twitter deals with bot accounts and looks at a similar application to this project, Botometer.

### Twitter

It has only been in the last few years that Twitter has taken the detection and suspension of bot accounts seriously. Brexit and the US Presidential elections were the deciding factors as the activities of bot accounts in the lead up to these proved to be a liability for the company. After an internal investigation, Twitter announced it would not be selling any more advertising to Russia media outlets Russia Today and Sputnik as these organisations were found to have interfered with the Presidential election on behalf of their government. [8]

Twitter has also been quiet active this year in detecting and shutting down bot accounts, between May and July around 70 million fake and suspicious accounts were shut down, same in October to a bot network of a few hundred accounts, that were involved in a coordinated campaign to defend Saudi Arabia’s Government’s role in the disappearance of Jamal Khashoggi, and most recently in November around 10 thousand more, that were all aimed at discouraging Americans to vote in the midterm elections. [9,10,11]

While the company has been trying, it is not an easy fight as they will always be on a reactive footing rather than a proactive one since the creation and running of bots, which are constantly evolving, can be automated but their large-scale detection relies on human intervention. This combined with the sheer volume of users and content through the site makes it a daunting and never-ending task. [12]

### Botometer

Botometer is a joint project between Indiana University Network Science Institute (IUNI) and the Center for Complex Networks and Systems Research (CNeTS). It employs a machine learning algorithm trained to classify an account as real or bot based on a labelled dataset comprised of over 10 thousand. It uses the Twitter REST API to gather public data on an account and then passed to the Botometer API which “*extracts about 1,200 features to characterize the account's profile, friends, social network structure, temporal activity patterns, language, and sentiment*”. [13] These are passed onto its models to compute the various scores which in turn go towards the overall score.

Its web front allows a user to check the activity of a Twitter account, after having the user’s Twitter account authorised, and gives it a score, out of 5, based on how likely the account is to be a bot with the closer the number is to 5 the more likely it is. There is also an option to check that accounts followers and the accounts it follows as well. It is simple, easy on the eye and informative

I used my own Twitter account to test it and the results are shown below. As you can see it rates my account with a bot score of 4.6/5 and a Complete Automation Probability (CAP) of 83% which is the probability that this account is fully automated. I set my Twitter account up a few years ago, followed some people, sent out one tweet and then completely ignored it so it is not surprising that Botometer’s models gave back these results even if they are wrong.



Figure 1 Botometer GUI [13]

## Existing FYPs

This section looks at existing final year projects related to this one.

### Fantasy Premier League Predictive Analytics

### CrimAnalytics – A Crime Prediction Web Application

### Anti-Bullying with Machine Learning

## Technologies Researched

This section deals with all research into the various possible technologies that could be used in this project and their benefits and limitations.

### Technologies for Data Mining & Machine Learning

#### R

R, a GNU project, is a programming language and environment for statistical computing and graphics. It is a variation on the S language and can run code from other languages such as C, C++ and Foltran. It has a wide and enthusiastic community worldwide ensuring there is plenty of support for beginners and its functionality can be extended through numerous packages found online. It has a wide, coherent and well-developed suite of facilities for data handling, storage, data analysis and graphical displays. [17]

Even with all this it does have its limitations such as memory management, R can consume all available memory, since some packages are created by normal users they might not always be up to industry standard and a basic knowledge of statistical vocabulary is needed as it was written by statisticians for statisticians.



Figure 2 R code example

#### Python

Python is an interpreted, high level programming language that places a lot of emphasis on code readability. It is Open Source, friendly and easy to learn with one of the largest communities in the programming world. [18] It also has a wide variety of packages covering nearly any topic a user might need or need, entire frameworks that can be used to get a project up and running quickly and simply and is supported across multiple platforms and systems.

It does have its downsides though, due to the fact it is compiled at run time it can be quiet slow running, it is also not a good choice if mobile development is at the core of your work or if your project is a game with high-end graphics.



Figure 3 Python code example

#### PyCharm

PyCharm is a Python IDE developed by JetBrains that includes intelligent code compilations, error checking and quick fixes and easy project navigation. [19] The professional version being made available to students, allowing access to many great other features such as starting a project with a framework already in place, database and SQL support and a Python Profiler.



Figure 4 PyCharm IDE

Below are a few integral Python libraries for this project if it is to be used.

##### Scikit-learn

Easily the most important and fundamental library to this project, Scikit-learn facilitates machine-learning for users of all levels by supporting various classification, regression and clustering algorithms including gradient boosting, random forests, support vector machines and k-fold cross validation. [20] It allows a user to easily create models, run them and compare their accuracy scores.

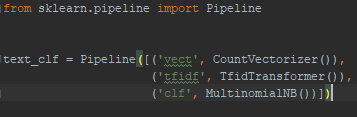


Figure 5 Scikit-learn code example

##### Pandas

The Pandas library provides high-performance, accessible data structures, such as Series and Data Frames, and data analysis tools. Data Frames are two-dimensional arrays while Series are only one-dimensional, and both offer huge array of features across them such as easily sorting them, iterating through them, searching across them for a count of specific entries or gaining stats on each column like mean or mode. [21]



Figure 6 Pandas code example

##### NumPy

The NumPy library offers numerous features for scientific computation and works well with the Scikit-learn library. It offers a powerful N-dimensional array object which can be used as an efficient container of generic data, as well as several sophisticated functions and tools. [22]

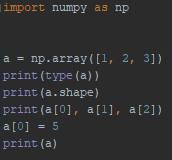


Figure 7 NumPy code example

### Technologies for Web Application

#### Flask

Flask is a web micro-framework for Python that provides users with a simple and effective core of tools, libraries and technologies to build a web application while also allowing it to be easily extended. [23] This has its benefits, as it is light with little need to keep an eye out for security bugs, but also has its limitations as the user will still have to do a lot of work themselves or increase the list of dependencies within the project.



Figure 8 Flask Logo [23]

#### Django

Django is a full web framework for Python that enables rapid deployment and elegant, practical design. It was built by skilful developers that abstracted much of the work required to get a web application of the ground such as managing views and templates, URL endpoints and security features, allowing users to focus on the nuts and bolts of their application instead. [24]



Figure 9 Django Logo [24]

#### Python to Twitter API

There are numerous Python libraries or wrappers that can connect to and gather data from the Twitter API such as Tweepy, Twython or Python Twitter. As Tweepy provides some great documentation and examples and is brilliantly supported I will be starting with that library. [25] If it does everything I need of it I will not need to use any of the other libraries.

### Technologies for Web Server

As Django and Flask web hosting only truly fulfil the role of development servers another web hosting service must be chosen for deployment onto a production server.

#### Apache HTTP Server

Apache HTTP Server is a free and open-source HTTP server built to operate on numerous different operating systems such as UNIX, Windows or Mac. [26] It is developed and maintained by an open community of developers, has a strong community of users willing to help first timers and is the most widely used web server in the market today. It does have its restrictions though: a strict updating policy must be put in place and the ability to modify its configuration can potentially cause a serious threat to the security of the web application.



Figure 10 Apache HTTP Server Logo [26]

#### Heroku

Heroku is a cloud platform as a service (PaaS) that allows users to deploy applications onto its servers and supports several programming languages including Python. It “*makes the processes of deploying, configuring, scaling, tuning, and managing apps as simple and straightforward as possible*” enabling developers to focus on building their app. [27] With this though comes a lack of control as the exact configuration of an application is set by them and if there is a high volume of data traffic then there is a premium charged.



Figure 11 Heroku Logo [27]

#### Amazon Web Services

Amazon Web Services (AWS) provides on-demand cloud computing platforms to users on a paid subscription basis. [28] This means that for many users it eliminates capacity constraints while mitigating the costs involved as well as adding in global reach and scalability. It is a high-tier grade service, but you are also paying for it unlike many others. It does offer a first-year free tier of all its services for first time customers and for students and educators there is an AWS Educate account that gives credits enabling hands on experience with their services.



Figure 12 AWS Logo [28]

### Technologies for Version Control

#### Git

Git is an open source distributed version control system that has seen a massive surge in popularity in recent years. It is free and easy to use and learn and can be run from the Git Bash client or from its integration in PyCharm, making it even easier to use and track changes in the process. Using either of these ways, it is simple to connect to GitHub, a web-based hosting service for Git repositories and ensure that a project is backed up with required access given to specific team members as well as giving public access to view the project and its code base. [29]



Figure 13 Git Logo

#### Mercurial

Mercurial is a free, distributed source control management tool that prides itself on how fast and powerful it is, it claims it can handle any project no matter the size or type. It is easy to learn and offers an instinctive interface. It is platform independent and extensible. For Mercurial, “history is permanent and sacred.” It only allows the rollback of the last pull or commit although there are extensions if more is needed. [30]



Figure 14 Mercurial Logo [30]

### Technologies for Data Storage

#### MySQL

MySQL is an open source relational database management system and one of the most popular systems in the world due to how easy it is to use, its nature as a relational database and how much investment and innovation has gone into it. It allows for powerful joins as well as standard features such as triggers, stored procedures and cursors. [31]

Due to its acquisition by Oracle though there have been some negatives: It is no longer completely open-source as some modules for it are now closed-source and it is no longer community driven.



Figure 15 MySQL Logo [31]

#### PostgreSQL

PostgreSQL is an open source object-relational database management system with a big emphasis on extensibility and creating features which safely store and scale the most complicated data workloads. [32] It essentially is a combination of relational and NoSQL databases, giving the best of both worlds through its extensions. It is highly scalable and supports JSON

Even with this it has some drawbacks: Its documentation has been known to be spotty and its configuration can be confusing to an inexperienced eye.



Figure 16 PostgreSQL Logo [32]

#### MongoDB

MongoDB is a free and open-source distributed NoSQL, or document, database that is scalable and flexible. It stores data in JSON- like documents which can be of any desired structure, removing the need for schemas, as in relational databases and allows for powerful ways to access and analyse data using Ad-hoc queries, indexing and real-time aggregation. [33] What is given up for this is the lack of functions or stored procedures as well as loss of strength in terms of ACID (Atomic, Consistency, Isolations, Durability).



Figure 17 MongoDB Logo [33]

## Other Relevant Research Done

This section covers all other relevant research done for this project. Research into approaches and methodologies will be dealt with in their own section later.

### Big Data

Over the last few decades the amount of data we have gone from using has increased from kilobytes to megabytes to gigabytes and has now hit terabytes in the everyday home. Many businesses have gone beyond this and are now dealing in petabytes or even exabytes of data.

These large datasets, both structured and unstructured, that are beyond the scope of traditional techniques to process due to their size or complexity and are used heavily within the domain of data science are what have been termed: Big Data. [34]

Organisations the world over have been investing into this area in the last number of years as the results that can arise from proper storage and use, through data mining and data analysis projects, of Big data can lead to massive returns or scientific breakthroughs.

### Data Mining

Data mining is the process of detecting anomalies, correlations and patterns within Big data to make predictions using a wide range of methods including various machine-learning algorithms. [35] There are various project models that can be used, although all of them are built upon the same foundation of stages

#### Data Acquisition

At the beginning of every data mining or analytics project, the first stage of the project is to ensure there is data to use. Where this data comes from varies from project to project as it may come from an inhouse databases or from surveys carried out with a business’s customer base.

The data used is referred to as a dataset with each row in the dataset being an instance of the data and each column being a descriptive feature of the data. [35]

#### Data Understanding

Understanding the business logic and context behind the acquired data and having a base knowledge of the project’s domain are integral parts of any data mining project as these helps to make sense of the relationships between features and enable easier selection of machine learning algorithms and improving their accuracy.

#### Data Preparation

This stage deals with the pre-processing of the data to ensure it’s in the correct state to be used for various business purposes such as in a machine-learning algorithm or data analysis. It includes the sub-stages of data cleaning and feature selection.

#### Data Cleaning

Data cleaning is the process of finding and removing entries in the data that has been either entered or formatted incorrectly. [35] Without proper data cleaning, when passing the data into a model various errors can arise, leading to program failure or completely inaccurate results.

#### Feature Selection

In any data mining project, the aim is to produce accurate predictions as efficiently as possible. To do this, we want to minimise the number of features without affecting the accuracy. [35] Therefore, feature selection is a key component that must be carefully deliberated and decided on.

#### Training & Testing

The data is used to train and test the models using various splitting techniques and the results are saved to be used in the following stage.

#### Evaluation

Where the results from the previous stage are evaluated based on various accuracy metrics to see how certain models perform. If they have reached the required level of accuracy then they can start to be used in an application or to help make business decisions, otherwise a return to a previous stage is needed.

### Machine Learning

Acquiring and preparing large sets of data is only part of the battle, the next major stage is to be able to detect patterns within this data and then make predictions from this. This is the core of Machine Learning, enabling us to extract significant insight from Big Data through complex, mathematical algorithms with minimal human intervention. [36] These algorithms are trained on sub-sets of the data to grow more accurate in their predictions.

#### Classifiers

Machine learning classifiers are divided into two sections

* ***Unsupervised Learning***
  + A classifier is given a set of inputs without any outputs known. It learns by itself, through specific methods, what outputs it should prescribe to each input.
  + Uses of Unsupervised learning algorithms
    - Find hidden patterns within data
    - Face recognition software
  + Example of Unsupervised learning algorithms
    - Clustering
* ***Supervised Learning***
  + A classifier is given a set of inputs with all outputs known. Using these it learns what outputs to prescribe to any future inputs.
  + Uses of Unsupervised learning algorithms
    - Predicting Football scores based on previous years data
    - Selection of advertising to be displayed to specific users
  + Examples of Unsupervised learning algorithms:
    - Support Vector Machines
    - K-nearest Neighbour
    - Naïve Bayes
    - Decision Trees
* **Example of Both**
  + Artificial Neural Networks
  + Anomaly Detection

##### Unsupervised Learning Algorithms

###### Clustering

There are multiple different types of clustering algorithms such as K-Means Clustering and Hierarchal Clustering. All of them revolve around grouping the data based on each input’s feature similarity.

The difference between algorithms between, for example, K-Means Clustering and Hierarchal clustering is that the former separates the data points iteratively into K clusters based on the features of the data while the latter considers each data point a cluster then identifies the clusters that are closest to each other and merging them, while taking note of the hierarchal relationship between them, and so on until only one cluster remains with one large hierarchy.[35]

##### Supervised Learning Algorithms

###### Naïve Bayes

Naïve Bayes classifiers belong to the family of probability-based classifiers and are based on Bayes’ theorem with the added assumption of conditional independence between all the features in the data. [36] This added assumption allows for the model to drastically reduce the amount of probabilities it must compute.

While this is quite a leap of faith to make, it still results in a robust model that delivers strong results and, when coupled with its scalability, efficiency and simplicity, is the reason it is normally the starting point for most data mining projects.

###### Support Vector Machines

A support vector machine classifier belongs to the family of error-based classifiers. It maps all the data as points in an N-dimension space, N being the number of features, and then tries to find a hyperplane, or decision boundary, that distinctly classifies the data points. [36]

It tries to maximise the distance between the hyperplane and data points from both classes. Those points closest to the hyperplane are called support vectors and have a significant impact on its placement.

New points are mapped to this space and classified depending which side of the hyperplane they belong to. It has a high degree of accuracy, takes up less computation power than other algorithms and can be used for both regression and classification task.

###### K-Nearest Neighbour

The K-Nearest Neighbour (KNN) is a simple, non- parametric classifier and belongs to the family of instance-based classifiers and as such keeps has either no or a very small training phase to it.

To classify new data points, the feature similarity of its k-nearest neighbours is used, with the new data point going to the class with the majority count. [35] While it makes no assumptions about data and is versatile it can be computationally expensive and sensitive to irrelevant data.

###### Decision Trees

##### Supervised & Unsupervised Learning Algorithms

###### Artificial Neural Networks

Artificial Neural Networks are “*biologically inspired computer programs designed to simulate the way in which the human brain processes information*”. [37] They detect patterns and relationships in data and from this infer knowledge and grow from their experience, learning to better classify data or perform tasks.

Using artificial neurons, the computerized version of a brain cell, a network is formed by connecting the output of specific neurons to the input of other neurons, forming a directed, weighted graph. A neurons weights and activation functions can be tuned over the learning process to increase the networks accuracy. [37]

###### Anomaly Detection

#### Training & Testing

Once a dataset has been fully prepared for use in a machine learning algorithm it must be divided up into training and testing datasets. The training dataset is a sub set of the original dataset used to train the model while the testing dataset is what is left. [35]

The training dataset is passed through a working model, with the results compared against the actual outcomes enabling the accuracy of the model to be measured. There are various ways that the base dataset can be divided up into training and testing datasets.

##### Holdout

This is the most basic division of the original set into training and testing sets with the partitioning of the original into two mutually exclusive sets. The split is usually taking a 2:1 ratio. The main problem with this method is that as more training data is used there is fewer testing data to be used. Ideally you want both the training and testing sets to be as large as possible. [36]

##### K-fold Cross Validation

Using this method, the original set is divided up into K partitions of equal size. Then for each partition, that partition acts as the testing set with the remaining partitions becoming the training set. A model is fitted using this training set and evaluated using the testing set. The model is discarded with the results being held onto before moving onto the next partition. [35]

This method deals with the main issue of the Holdout method, ensuring the entire dataset is used for both training, each partition being used K-1 times, and testing, each partition used once, with the results being of significant use at the end for evaluation.

#### Evaluation

After a model has been created and data run through it, results will have been produced. The accuracy of these results must be measured carefully. Only by truly understanding the accuracy of the results and what influenced it will someone be able to improve the model and its accuracy.

##### Classification Accuracy

Simply put, the accuracy of a model is the amount of predictions it got right. Put into formulaic terms: Accuracy = No. of correct predictions / Total no. of predictions. [36] This in and of itself is not enough in terms of detail for a proper model evaluation and as such other methods must also be employed.

##### Confusion Matrix

A confusion matrix is a table layout for the visualisation of the performance of a model. The totals of correct and incorrect predictions are calculated and broken down by class. These values are placed into a matrix with predicted across the top and expected down the side.

When looking at a two-class instance or one class against all the others this matrix will then hold the values for True Positives, False positives in the 1st row and False Negatives and True Negatives in the 2nd row. [36] This data holds much more meaning then the previous method of evaluation and can help in knowing what part of the model needs to be tuned to gain a better accuracy level.

##### F1 Score

The F1 score is another method of measuring a model’s accuracy. It is obtained by computing the weighted average of the Recall and Precision. [36] The closer this score is to 1 the more accurate the model is.

F1 = 2\*(Recall \* Precision) / (Recall + Precision).

Recall is found by dividing the total correct predictions by the sum of the total correct predictions and false negatives.

Recall = TP/ (TP + FN)

Precision is found by dividing the total correct predictions by the sum of the total correct predictions and false positives.

Precision = TP/ (TP + FP)

## Resultant Findings and Requirements

This section will deal with what technologies were chosen to use in this project and why and what datasets will be used to create my machine-learning models.

### Chosen Technologies

Python will be used over R for this project due to its familiarity, design tropes and high extensibility with a wide variety of libraries supporting every aspect of what is need in this project. PyCharm Professional IDE was chosen due to its wide variety of features including integrated Git support connecting to a GitHub repository with Git being the preferred VCS due to it being well supported, easily operated and widely recognised.

Django will be the framework of choice as it just provides more structure out of the box then Flask does. Amazon Web Services will be used for hosting the production server as the first-year free tier completely nullifies the costs and offers a great range of services, although if this project was to be extended beyond final year there would be more consideration to swapping to Apache HTTP Server due to fiscal reasons. For data storage PostgreSQL was chosen as it combines the best of both a relational and NoSQL databases.

### Chosen Datasets

For the data that my machine-learning models will use, the cresci-2017 dataset was selected. [39] This dataset has been used in academic studies in the field of Twitter bot detection, [7] is part of the datasets used by the Botometer application and covers an excellent range of different accounts.

It is split further into several smaller datasets. First there is a dataset of genuine account, then there are three groups of traditional spambots, the first group are general spambots without any focus, second group are spambots attempting to promote a web URL to try and get users to click it and lastly a group of spambots attempting to push job offers on users as well as getting them to click a specific URL.

Next is a group of fake follower bots which exist purely to make a user appear more popular or influential on the platform. Lastly are three groups of social spambots, the first group are spambots that retweeted a specific political candidate in Italy, second one group spambots attempting users to download a specific mobile application and lastly a group of spambots trying to sell products on Amazon.com.

# Approach & Methodology

# Design

# Prototyping & Development

# Project Plan

# Testing & Evaluation

# Conclusions & Future Work

# Bibliography

1. Statista; (October 2018), Twitter: number of active users 2010-2018, www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/, Date Accessed: November 2018
2. Varol, Onur; Ferrara, Emilio; Davis, Clayton A.; Menczer, Filippo; Flammini, Alessandro; (2018) "*Online Human-Bot Interactions: Detection, Estimation, and Characterization*", ICWSM'17, Page 1-9.
3. Perlroth, Nicole; (April 2013), Fake Twitter Followers Become Multimillion-Dollar Business, bits.blogs.nytimes.com/2013/04/05/fake-twitter-followers-becomes-multimillion-dollar-business, Date Accessed: October 2018
4. Perlroth, Nicole; (April 2013), Researchers Call Out Twitter Celebrities With Suspicious Followings, bits.blogs.nytimes.com/2013/04/25/researchers-call-out-twitter-celebrities-with-suspicious-followings, Date Accessed: October 2018
5. Ferrara, Emilio; (November 2016), How Twitter bots affected the US presidential campaign, theconversation.com/how-twitter-bots-affected-the-us-presidential-campaign-68406, Date Accessed: October 2018
6. Nimmo, Ben; (August 2017), #BotSpot: Twelve Ways to Spot a Bot, medium.com/dfrlab/botspot-twelve-ways-to-spot-a-bot-aedc7d9c110c, Date Accessed: October 2018
7. Efthimion, Phillip George; Payne, Scott; Proferes, Nicholas; (2018) "*Supervised Machine Learning Bot Detection Techniques to Identify Social Twitter Bots*", SMU Data Science Review, 1(2) , Article 5.
8. Rushe, Dominic; (October 2017), Twitter bans ads from RT and Sputnik over election interference, www.theguardian.com/technology/2017/oct/26/twitter-bans-ads-from-russia-today-and-sputnik-over-election-interference, Date Accessed: October 2018
9. BBC news; (July 2018), Twitter ‘shuts down millions of fake accounts’, www.bbc.com/news/technology-44682354, Date Accessed: October 2018
10. Leskin, Paige; (October 2018), Twitter shuts down bots pushing pro-Saudi reports on missing columnist, uk.businessinsider.com/twitter-shuts-down-pro-saudi-bots-missing-columnist-2018-10?r=US&IR=T, Date Accessed: November 2018
11. Netimperative; (November 2018), US elections: Twitter shuts down 10,000 bot accounts ‘discouraging voting’, www.netimperative.com/2018/11/us-elections-twitter-shuts-down-10000-bot-accounts-discouraging-voting/, Date Accessed: November 2018
12. Sattler, Jason; (September 2017), blog.f-secure.com/4-reasons-so-hard-for-twitter-to-shut-down-bots/, Date Accessed: October 2018
13. OSoMe; (-), Botometer by OSoMe, botometer.iuni.iu.edu/#!/faq, Date Accessed: October 2018
14. Brady, Alex, 2018, “Fantasy Premier League Predictive Analytics”, Dublin: D.I.T.
15. Jennings, Sean, 2017, “CrimAnalytics: A Crime Prediction Web Application”, Dublin: D.I.T.
16. O’Neill, Shane, 2017, “Anti-Bullying with Machine Learning”, Dublin: D.I.T.
17. R; (-), R: The R Project for Statistical Computing, www.r-project.org, Date Accessed: October 2018
18. Python; (-), Welcome to Python.org, www.python.org, Date Accessed: October 2018
19. JetBrains; (-), PyCharm: the Python IDE for Professional Developers by JetBrains, www.jetbrains.com/pycharm, Date Accessed: October 2018
20. Scikit-learn; (-), scikit-learn: machine learning in Python, scikit-learn.org/stable/, Date Accessed: October 2018
21. Pandas; (-), Python Data Analysis Library, pandas.pydata.org, Date Accessed: October 2018
22. NumPy; (-), NumPy -NumPy, www.numpy.org, Date Accessed: October 2018
23. Ronacher, Armin; (-), Welcome | Flask (A Python Microframework), flask.pocoo.org, Date Accessed: October 2018
24. The Django Software Foundation; (-), The Web framework for perfectionists with deadlines, www.djangoproject.com, Date Accessed: October 2018
25. Tweepy; (-), Tweepy Documentation, tweepy.readthedocs.io/en/v3.5.0/index.html, Date Accessed: October 2018
26. The Apache Software Foundation; (-), Welcome! – The Apache HTTP Server Project, httpd.apache.org, Date Accessed: November 2018
27. Heroku; (-), Cloud Application Platform, www.heroku.com, Date Accessed: November 2018
28. Amazon; (-), Amazon Web Services (AWS) – Cloud Computing Services, aws.amazon.com, Date Accessed: November 2018
29. Git; (-), Git, git-scm.com, Date Accessed: November 2018
30. Mercurial; (-), Mercurial SCM, www.mercurial-scm.org, Date Accessed: November 2018
31. Oracle; (-), MySQL, www.mysql.com, Date Accessed: October 2018
32. PostgreSQL; (-), PostgreSQL: The world’s most advanced open source database, www.postgresql.org, Date Accessed: October 2018
33. MongoDB; (-), Open Source Document Database, www.mongodb.com, Date Accessed: October 2018
34. Mauro, Andrea De; Greco, Marco; Grimaldi, Michele; “*What is big data? A consensual definition and a review of key research topics*”, AIP conference proceedings, (2015), 1644(1), pp. 97–104.
35. Bramer, Max; (2016) “*Principles of Data Mining*”, Salmon Tower Building New York City: Springer.
36. Kelleher, John; Mac Namee, Brian; D'Arcy, Aoife; (2015) “*Fundamentals of machine learning for predictive data analysis*”, Cambridge Massachusetts: MIT Press.
37. Agatonovic-Kustrin, S; Beresford, R; “*Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research*”, J Pharm Biomed Anal. (2000), 22(5), pp.717-27.
38. S
39. s
40. Botometer; (-), Bot Repository, botometer.iuni.iu.edu/bot-repository/datasets.htm[l](https://botometer.iuni.iu.edu/bot-repository/datasets.html), Date Accessed: October 2018