

|  |  |  |
| --- | --- | --- |
| The Truth about Twitter | | |
| **Student Name:**  **Max MacDonald** | **Student ID:**  **C15740661** | **Supervisor Name:**  **Cindy Liu** |
| **Link to Software Repository:** | **https://github.com/TheBeardBeatsAll/The-Truth-About-Twitter** | |

**Table of Contents**

1. Project Statement 3

2. Research 3

Background Research 3

Alternative Existing Solutions to Your Problem 3

Technologies Researched 3

Other Relevant Research Done 3

Resultant Findings and Requirements 3

Bibliography 3

3. Approach and Methodology 3

4. Design 4

a. Technical Architecture Diagram 4

b. Other Design Documents 4

5. Prototyping and Development 4

6. Testing 4

7. Issues and Risks 4

8. Plan and Future Work 4

# Project Statement

Social media sites, such as Facebook, Instagram and Twitter, have been flooded with numerous types of fake accounts in the past few years and each of these sites have an urgent need to be able to tell the difference between a real user and a bot user.

The goal of this project is to create a web application that users can use to find out the likelihood that a Twitter account is real or a type of bot account. Users of this application will be able to input a Twitter handle that will be used to hit the Twitter API, retrieving specific information about the account. This data will be passed into a machine-learning model which has been trained on a dataset of real and bot accounts. The model will then decide whether the account is real or not, displaying it’s answer to the web front with relevant statistics.

# Research

This section will cover all research done for this project including background research on Twitter and bot accounts, applications or solutions similar to this project, 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.

## Background Research

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. Registered users can broadcast short posts called tweets which can be liked, reposted or 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.

### 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, all depending on the settings of the controlling application.

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 to this with any differences being based on the platform differences.

### Types of Bots 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. Major celebrities such as 50 Cent and brands like Mercedes-Benz have come under scrutiny for possibly engaging in this practice.

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 Bots 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 accounts 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. 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, due to the fact that 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:

* How often per day an account tweets can 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 bio help identify them or add to their anonymity? Is the account handle just an alphanumeric scramble?
* Links in the bio as some bots’ purpose is to redirect people to certain websites or have them download malicious software without them knowing although not complete indicative as some people do put links in their bio for example a link to their company’s home page.
* Abnormal posting hours as for example an account that identifies as a British man living in London but is posting 9-5pm Moscow time.
* Generic bio or lack of one as the programs which create these bots are not set up to make completely unique bios.
* Lack of followers as some bot accounts will have few yet still be retweeted thousands of times
* Ratio of how many other accounts an account follows to how many follow it. Most bot accounts won’t be followed by that many.
* Screen name and account handle completely mismatching.

## Alternative Existing Solutions to Your Problem

This section explains how Twitter deals with bot accounts, looks at an application, Botometer, similar to this project as well as academic studies done into detecting bot accounts with increasing accuracy.

### 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.

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 Suadi 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.

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.

### 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 data set 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*”. These are passed onto its models to compute the various scores which in turn go towards the overall score.

It’s web front allows a user to check the activity of a Twitter account, after giving permissions using the user’s account, 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.



### Academic Studies

#### Supervised Machine Learning Bot Detection Techniques to Identify Social Twitter Bots

sds

#### The Paradigm-Shift of Social Spambots: Evidence, Theories, and Tools for the Arms Race

asd

## 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 Models

#### 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.

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.



#### 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. 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.



#### PyCharm

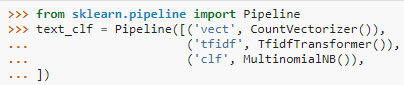
PyCharm is a Python IDE developed by JetBrains that includes intelligent code compilations, error checking and quick fixes and easy project navigation. The professional version being made available to students, allowing access to many great other features such as starting a project with a framework already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



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. It allows a user to easily create models, run them and compare their accuracy scores.



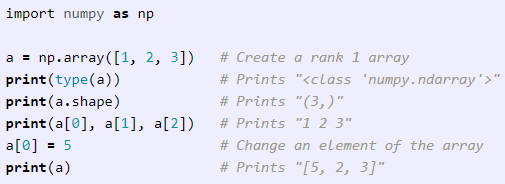
##### Pandas

The Pandas library provides high-performance, accesible 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.



##### 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 a number of sophisticated functions and tools.



### 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. This has its benefits, as it is light and 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.

#### 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.

#### Python to Twitter

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. As long as 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 or Windows. 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.

#### 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. 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.

#### Amazon Web Services

Amazon Web Services (AWS) provides on-demand cloud computing platforms to users on a paid subscription basis. 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.

### Technologies for Version Control

#### Git

Git is an open source distributed version control system that I am already quiet familiar with from use in previous college 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 .

#### 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.

### 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.

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.

#### 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. 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: It’s documentation has been known to be spotty and its configuration can be confusing to an inexperienced eye.

#### 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. Allows for powerful ways to access and analyse data through the use of Ad-hoc queries, indexing and real-time aggregation. 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).

## Other Relevant Research Done

Approaches, Methodolgies and Technical architectures dealt with in their own section.

### Data Science

#### Big Data

#### Data Mining

#### Machine Learning

## Resultant Findings and Requirements

This section will deal with what technologies I have chosen to use in my project and why, what data sets I will be using to create my machine-learning models and any challenges I foresee leading from research into development.

### Chosen Technologies

I will be using Python over R for this project as it is a language I am familiar with and like and it is highly extensible with a wide variety of libraries supporting every aspect of what is need in this project. I will be using the PyCharm Professional IDE with integrated Git support connecting to a Github repository as this IDE provides such a wide variety of features and Git is just so easy to use 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 I was to extend this project beyond final year I would consider starting instead with or switching too Apache HTTP Server. For data storage I will be using PostgreSQL as it combines the best of both a relational and NoSQL databases.

### Chosen Data Sets

For the data that my machine-learning models will use, I have selected cresci-2017 dataset. This dataset has been used in academic studies in the field of Twitter bot detection, 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.

### Challenges

#### Time

Time management for this entire project will be a major challenge as I will also be juggling my modules, their workloads and exams. While I have planned everything out using a Gantt chart, unexpected situations can arise within college and in my outside life which can throw work timelines off.

#### Storing the Data

The data must be read in correctly from csv files into a database in the correct format with the database held in a safe and secure way that preserves its integrity. If data is corrupted in any way or duplicated this can heavily influence the accuracy of any models when the data is passed through a classifier.

#### Creating an accurate Model

Picking the correct classifier and in turn creating and tuning an accurate model will be quite difficult and take up a significant period as this is an area that I am only becoming familiar since the start of the college year. It is the most fundamental aspect that the success of the project relies on as without an accurate model the web application holds little value apart from experience gained.

#### Integration

Once each part of the project, the web app and the machine-learning models, are complete then the models need to be integrated into the web app resulting in the finished product. This could potentially be quite tricky and time consuming as it will involve connecting two separate code bases. Allowing for this though ensures each part can be worked on independently which overall should work out for the better.

## Bibliography

Your researched resources. Use Vancouver citation style.

<https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/>

<https://www.mongodb.com/what-is-mongodb>

<http://flask.pocoo.org/>

<https://www.heroku.com/>

<https://tweepy.readthedocs.io/en/v3.5.0/getting_started.html>

<https://www.djangoproject.com/>

<http://www.numpy.org/>

<https://www.postgresql.org/>

<https://botometer.iuni.iu.edu/bot-repository/datasets.html>

<https://blog.f-secure.com/4-reasons-so-hard-for-twitter-to-shut-down-bots/>

<https://medium.com/dfrlab/botspot-twelve-ways-to-spot-a-bot-aedc7d9c110c>

<https://scholar.smu.edu/cgi/viewcontent.cgi?article=1019&context=datasciencereview>

<https://www.mysql.com/>

<https://www.atlassian.com/blog/software-teams/mercurial-vs-git-why-mercurial>

<https://git-scm.com/>

<https://www.jetbrains.com/pycharm/>

<https://scikit-learn.org/stable/>

<https://botometer.iuni.iu.edu/#!/>

<http://www.netimperative.com/2018/11/us-elections-twitter-shuts-down-10000-bot-accounts-discouraging-voting/>

<https://www.iit.cnr.it/sites/default/files/main-newtemplate.pdf>

<https://www.bbc.com/news/technology-44682354>

<http://uk.businessinsider.com/twitter-shuts-down-pro-saudi-bots-missing-columnist-2018-10?r=US&IR=T>

<https://www.r-project.org/about.html>

# Approach and Methodology

What is your approach to this project? Are you using any particular software methodology? Eg. Are you delivering design/ code in phases, or are you completing all design up front, followed by all coding? Have you some sections lower priority if time runs short?

Talk about all theses in terms of the project then sections explaining them

Cross Industry Standard Process for Data Mining (CRISP-DM), SEMMA, KDD

Agile/ Scrums / Sprints initially Waterfall

## Agile & Scrum/Sprint

## Data Mining Project Management Models

### KDD

### SEMMA

### CRISP-DM

### Differences in Models

### Conclusion



# Design

## Technical Architectures

Model view template which is a Django’s variation on the MVC

## Technical Architecture Diagram

Insert the architecture for your solution

## Other Design Documents

Insert other design artefacts that explain your system: e.g. Use cases/ ERDs/ Class diagrams

# Prototyping and Development

Explain exactly what prototyping and development you have completed.

## Vertical Prototype

Web app which can hit the twitter API and basic model

## Development

# Testing

Explain your planned testing approach: For example: who will be involved, what test scripts are planned, how will the testing be executed.

Random samples from datasets as well as researching and selecting several accounts to test against.

# Issues and Risks

Explain the main issues / challenges that are unresolved on your project. – and your suggested approach to solving them. This is a critical part of your report to show that you understand what is required to complete the project.

Proper understanding of the data and connections between them.

Creating an accurate model, choosing classifiers and algorithms.

# Plan and Future Work

After the submission of this report and subsequent presentation and demo of my vertical prototype the two key deliverables left will be finishing my dissertation and the full web application by the end of April.

As shown in the Gantt chart below, most of the development time will go to continued building and accuracy tweaking of the basic models from the vertical prototype and then integrating these, once finished, into the web application. The rest of the time will go into more background reading and morphing this report into the first part of my dissertation and then writing the rest of it. Time has also been set aside to focus on my Winter exams when they are due.

