## 2.3. Existing FYPs

For each project:

* Author
* Web address
* Screengrab (options)
* 100-200 word description of proejct

100-200 word description of how it inspires your project.

## Technologies Researched

### Technologies for Version Control

#### Git

#### Mercurial

### Technologies for Data Storage

#### MySQL

#### PostgreSQL

#### MongoDB

## Other Relevant Research Done

### Big Data

### Data Mining

#### 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.[32]

#### 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 help 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.[32] 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.[32] Therefore, feature selection is a key component that must be carefully deliberated and decided on.

### 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.[33] 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
  + Examples of Unsupervised learning algorithms:
    - Clustering
    - Artificial Neural Networks
* ***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

##### 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.[32]

###### Artificial Neural Networks

Artificial Neural Networks are “*biologically inspired computer programs designed to simulate the way in which the human brain processes information*”.[34] 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.[34]

##### 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.[33] 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 Machine

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.[33]

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.

###### Linear Regression

Linear Regression is an algorithm used to model the relationship between two or more continuous variables, or features in this case, by fitting a linear equation to their data.[33] Understanding these relationships can help with fine tuning accuracy as well as using the fitted equation to make predictions. Unlike Naïve Bayes it is used to compute a numerical value rather than predict a class type.

Two Types of Linear Regression:

* Simple: Gives us the relationship between one explanatory variable and one dependent variable.
* Multiple: Gives us the relationship between multiple explanatory variable and one dependent variable.

###### 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.[32] While it makes no assumptions about data and is versatile it can be computationally expensive and sensitive to irrelevant data.

#### 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.[32]

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 takes a 2:1 ratio. The main problem with this method is that as more training data is used there is less testing data to be used. Ideally you want both the training and testing sets to be as large as possible.[33]

##### 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.[32]

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.[33] 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.[33] 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.[33] 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 I have chosen to use in my project and why, what datasets 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 DataSets

For the data that my machine-learning models will use, I have selected cresci-2017 dataset.[35] 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 and Methodology

This project has been performed iteratively, using the Agile and Kanban methodology for the overall scope of the project, including the web application, while for the data mining aspect of the project the CRISP-DM methodology was chosen from a few data mining project management models.

Both the Agile and Kanban methodologies and their use in this project as well as the CRISP-DM methodology and a similar methodology: SEMMA, will be explained in this section. A comparison will be made between them and reasons given why I chose the former over the latter.

## Agile & Kanban

### Agile

The Agile concept is an approach to project management within the domain of software development and includes various principles such as Feature Driven Design, Scrums, XP and Kanban, the last of which I have incorporated into my project and discuss later.

Agile and its sub-principles revolve around being adaptable, collaborative and versatile and with the focus on iterative and incremental development.[36] Projects that adopt Agile need an approach that facilitates rapid and flexible responses to change as well as continuous improvement.

The 12 Agile principles outlined in the Agile manifesto are [37] :

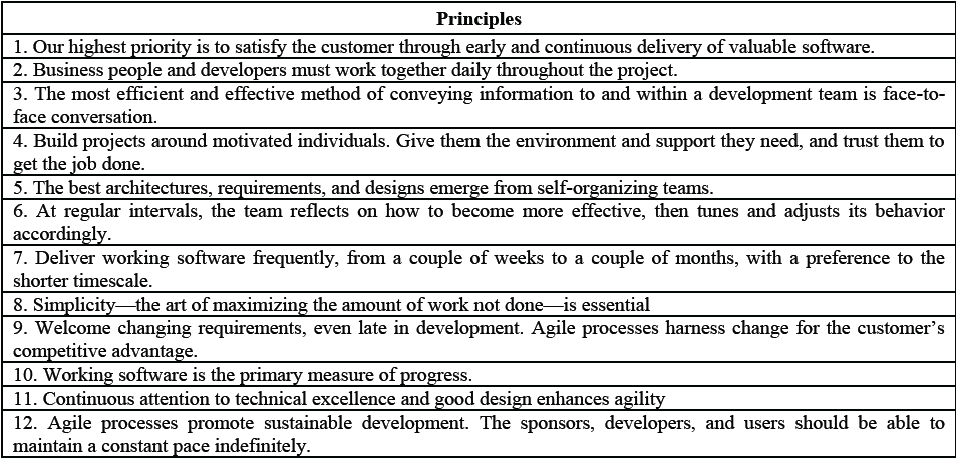


Figure 8 12 Agile Principles

Some of the key differences between Agile and traditional methodologies are that the management style of a project is to lead and collaborate in the former and command and control in the later, that communication throughout the project is informal vs formal, that the developmental model is an evolutionary-delivery model vs a life-cycle model and that implementation is the focus vs spending large quantities of time on design.[36]

### Kanban

Kanban is a part of the Agile family with a heavy focus on continuous delivery while at the same time ensuring that the development team do not become overburdened.[38]

The Kanban methodology was named in 2007 after several presentations given by David Anderson of his management approach at various companies. The word “*kanban*” is the Japanese for sign or signal card and the methodology’s roots date back to the middle of the 20th century in Japan where Taiichi Ohno, while working for Toyota, employed the first Kanban system to regulate the workflow in the company. [39]

Kanban has three main principles behind it [38] :

* Visualise the flow of work: Set up an environment, either through post its on a board or digitally to visualise all the work items to give them context.
* Limit the work in progress: Limit each team member to at most 3 pieces of work at any given time to ensure the team does not start and commit to too much work.
* Enhance the flow: Once a team member has finished a piece of work, they take the highest priority work piece in the backlog.

Some of the benefits of the Kanban methodology are shorter cycle times ensuring that new features can be delivered quicker, easy adaption to frequent changes in work piece priority and requires less oversight ensuring team and project leads have more time to focus on other activities.

### Project Use

While Agile and Kanban methodologies revolve around a team of multiple members working together it isn’t hard to adapt them to my case of a team of one for this project. I take on the roles of not only all members of the team and the team lead but also that of the shareholders that commission the project.

* Shareholders: All the requirements of this project were laid out by me at its outset as well as the one who has been and will continue to assign priorities to all the deliverables and smaller pieces of work.
* Team lead & members: Every piece of work on this project has been and will be worked on and completed by me. I am in charge of ensuring I do get the work done in a timely manner and to a sufficient quality.

I will be using Trello, a free to use web-based project management application, to keep track of and manage my work load through the Kanban method. This was a piece of software I was introduced to during my work placement and is extremely simple and easy to use.[40]

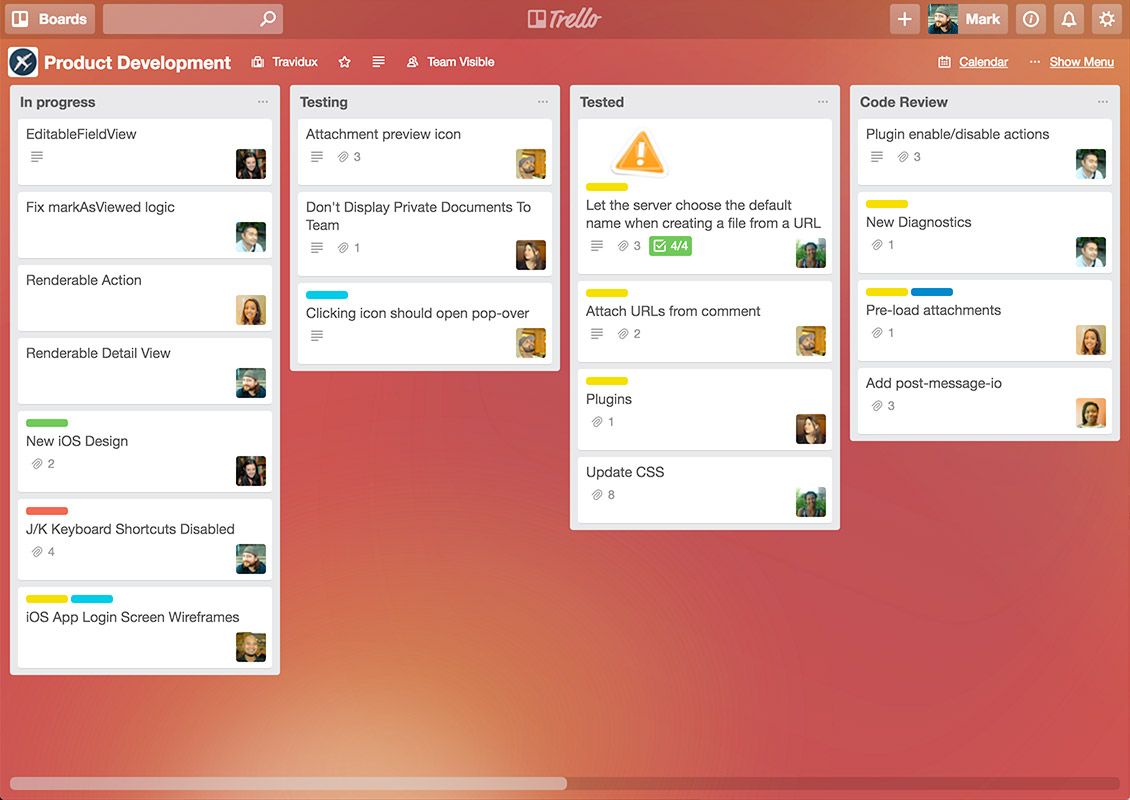


Figure 9 Sample Trello Board [40]

## Data Mining Project Management Models

### CRISP-DM

CRISP-DM model is an acronym that stands for Cross-Industry Standard Process for Data Modelling and is a cycle which consists of 6 stages. The sequence of these stages is not strict which allows for movement between any stage if so required.[41]

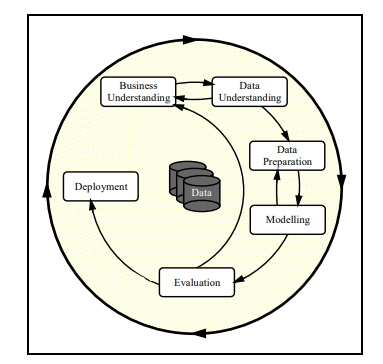


Figure 10 CRISP-DM Model [41]

#### Business Understanding

This stage revolves around understanding the business side of the data mining project:

* What is the domain of the business?
* Assessing the situation in terms of hardware, software, data sources and knowledge bases.
* Transforming business goals into data mining objectives.
* Producing a project plan.

#### Data Understanding

This stage is where data is acquired, and time spent exploring it, trying to understand any correlations between features as well as finding any data quality issues present in the data through a data quality report and data visualisations.

#### Data Preparation

This is the most important stage of the model and requires an excellent understanding of the previous two stages. This is where numerous tasks are performed on the data such as the pre-processing of data, cleaning and reformatting data to remove missing or fix corrupt values and feature selection. Aggregation or merging of data may also occur here.[41]

If the data is not prepared properly then the accuracy of any models it is passed to will suffer greatly and can take multiple iterations to get right.

#### Modelling

This stage revolves around the selection of modelling technique or machine-learning algorithms, decide how to measure the model’s validity or accuracy, the building of the model and its assessment to fix any mistakes that can occur in the building process.[41]

These mistakes arise due to *noise*: outliers or missing values in the data that have managed to slip through the previous stage. This then leads to *overfitting*, where the model is too complex for the data and therefore is highly sensitive to noise, or *underfitting*, where the model is too simplistic and unable to detect patterns within the data. Either of these have a negative impact on the accuracy.[32]

#### Evaluation

Evaluation of results from the models, once data has been passed through, is done here. This allows us to get an understanding of the suitability of the models as well as any errors we may have made at an earlier stage. The entire process is reviewed and the next iteration, if needed, is planned.

#### Deployment

Once a model has been created that hits a satisfactory accuracy level it can be deployed. This can have a variety of different meanings, dependent on the project itself, from being used in a work report or a scientific paper or in a data mining application. The deployment must be planned with any monitoring and maintenance considered.

### SEMMA

SEMMA is another model, developed by the SAS Institute, which is used to manage a data mining application and is an acronym for the 5 stages that comprise the model [42] :

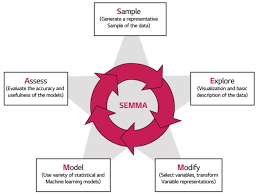


Figure 11 SEMMA Model [43]

#### Sample

This stage focuses on taking a sample of the dataset for use in the model. The data must be of sufficient size such that accurate patterns can be drawn from it while small enough that it can still be used efficiently. Data partition of the sampled dataset into training and testing sets is also done here.

#### Explore

Exploration of the data is done here to detect any unexpected patterns, anomalies such as missing or corrupted data, or instances in the data that prove to be unnecessary while also gaining a better understanding of the data. This is done through visual representations and statistical techniques.[42]

#### Modify

The data is modified by creating, deleting, selecting and/or transforming variables within it to ensure the data being passed into the model is of high quality and considers any issues that arose in the exploration stage. Not every issue will be perfectly solvable, and it will be a case of applying the best fit solution.[42]

#### Model

The creation of the model is dealt with here, where selected modelling techniques are employed to build a model that will accurately make predictions based upon the data that is passed to it.

Again, like in CRISP-DM, the problems of *overfitting* and *underfitting* due to *noise* must potentially be dealt with.

#### Assess

The outputs of the model, after inputting the training set, that was set aside during the sampling stage, are compared to the actual outcomes of that dataset with the model’s accuracy and usefulness being evaluated.

### Differences in Models

Looking at both models it is easy to see that they are quite similar, in fact SEMMA appears to be akin to a slimmed down version of CRISP-DM with the first, Business Understanding, and last stage, Deployment, omitted such that all the focus is on data modelling aspect of the application, though realistically some knowledge of the business domain must be known or else the final product will be lacking in various areas.

This table clearly demonstrates this comparison:

|  |  |
| --- | --- |
| CRISP-DM | SEMMA |
| Business Understanding | ----------- |
| Data Understanding (Part 1) | Sample |
| Data Understanding (Part 2) | Explore |
| Data Preparation | Modify |
| Modelling | Model |
| Evaluation | Assess |
| Deployment | ----------- |

### Conclusion

The CRISP-DM model was chosen for use within this project as it is an industry standard model that is used widely and as such is well proven, is robust and versatile allowing for movement between any stage if needed and finally inclusion of the deployment stage, which SEMMA is missing, as the data mining aspect of the project will need to be integrated and deployed within the overall web application.

# Design

This section details the technical architecture chosen for this project, a diagram of it and all other design documents including Use Case and Class diagrams and an Entity Relationship Diagram.

## Technical Architectures

### Model View Controller

The Model View Controller (MVC) architecture is used across a wide range of applications where there is a need to provide a User Interface through a desktop or web front-end. It is a three-tier architecture which uses the Controller, comprised of several classes such as a Command Factory class and Command, Service, and DAO classes, to pass information between the View, i.e. the front end, and the Model, i.e. the backend.[44]

This ensures the separation of roles between the different sections of code in a project. This makes it easier to divide up the work in a project as team members can focus on specific sections without worrying too much about the other parts enabling better development and testing.

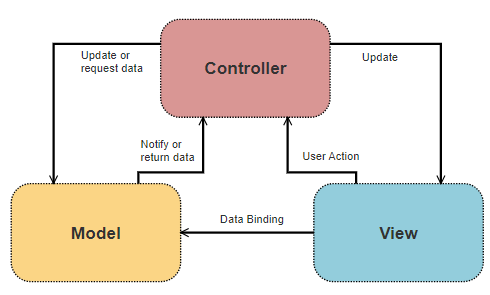


Figure 12 Model View Controller

### Model View Template

For this project I have chosen to use the Django framework which uses its own modified version of the MVC called the Model View Template (MVT). In this adaption Django takes care of the Controller role and replaces it with the Template section, which takes the role of the presentation layer by containing all the HTML, CSS and Forms files while the View section deals with all business logic and handles all requests from and responses to the User. The Model section stays the same and deals with everything to do with the database.[45]

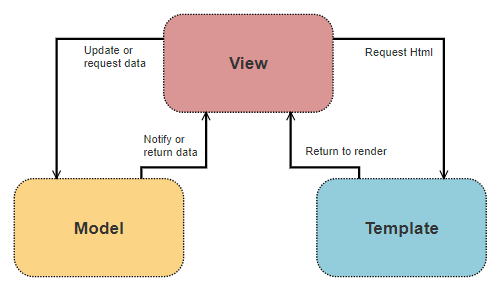


Figure 13 Model View Template

## Technical Architecture Diagram

As this is a web-based data mining application I have adapted the MVT architecture to suit this project by adding another layer between View and Model, called Mining, to consider the data mining and machine learning aspects of this project. This layer deals with everything from data pre-processing to the creation and evaluation of the machine-learning models to analysis of new data passed to it.

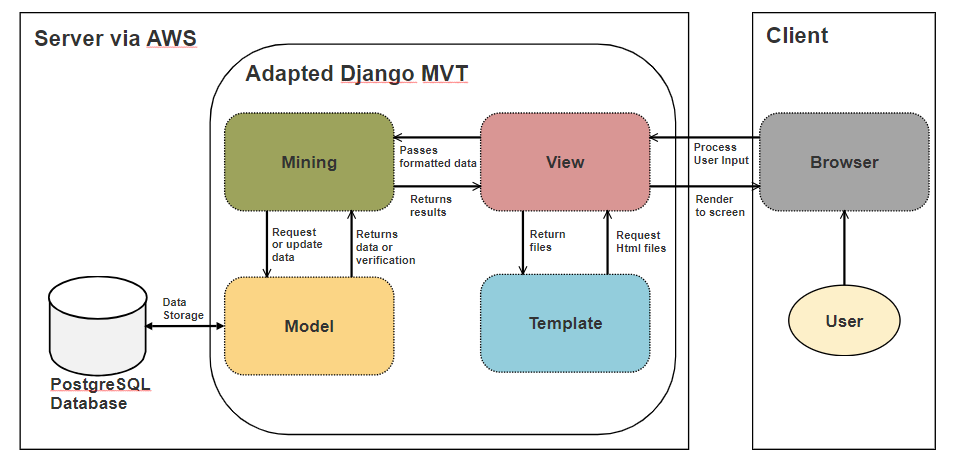


Figure 14 Application Technical Architecture

## Other Design Documents

### Use Case Diagram

The use case below details how a user will interact with the system. The user can enter in their own Twitter username or any other one they wish, be it a celebrity’s, one of their friends or any other account they know off. They will then be able to view the results about how likely that account is a bot and be able to either share those results on social media or download them into a file.

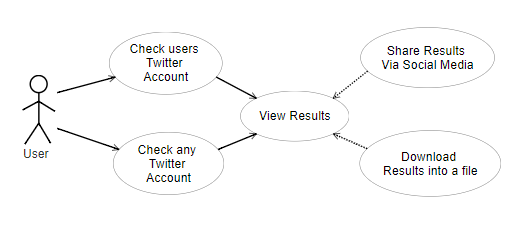


Figure 15 User Use Case Diagram

### Entity Relationship Diagram

This diagram is subject to change due to trying to achieve a more accurate result in further iterations of the development cycle.

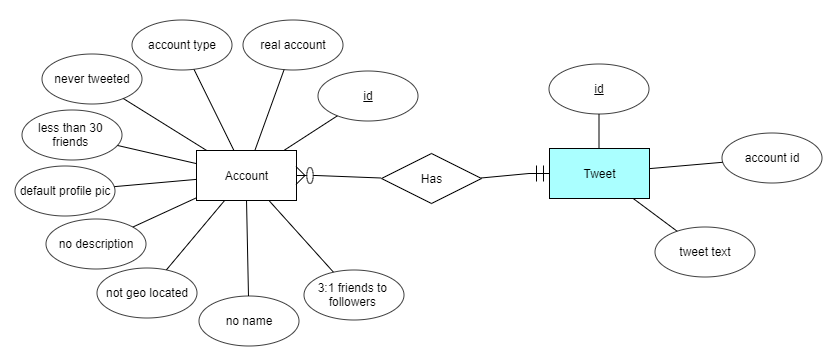


Figure 16 Entity Relationship Diagram

## 3.1 Introduction

Following on from the previous chapter, where some of the key background research was presented, these themes will be continued in this chapter, where the design of the system will be presented. The first section will look at the software methodology employed in this project which describes … The next section outlines the technical architecture of the system …

## 3.2 Software Methodology

Consider using personas, user stories, and storyboards.

## 3.3. Technical Architecture

This section discusses the technical architecture of the system, it first discusses the front-end design of the system, and presents some proto

### 3.3.1. Front-End Design

Screen prototypes – paper and computer-based

### 3.3.2. Middle-Tier Design

The connection type, security considerations

### 3.3.3. Back-End Design

ERDs and description of data

## 3.4. Software Test plan

## 3.5. Conclusions

# 6. Project plan

# Prototyping and Development

This section explains what prototyping and development has been completed to date, giving details on building the web front-end and the creation of a basic machine-learning model.

## Vertical Prototype

The prototyping for this project revolves around creating a vertical prototype, which shows the basic structure and functionality of both sub-sections of the project. This will then be reviewed over the December break and if found satisfactory, built upon heavily to create the final application.

### Web Front-End

For this section of the vertical prototype, the goal was to have a working web-app that could connect to the Twitter API and retrieve data from it, in this case the twenty most recent tweets from the account linked to the username chosen by the web-app user.

Below are two screenshots of the working web page which asks for the user to input a Twitter username then retrieves and outputs the data received back from the Twitter API to screen.

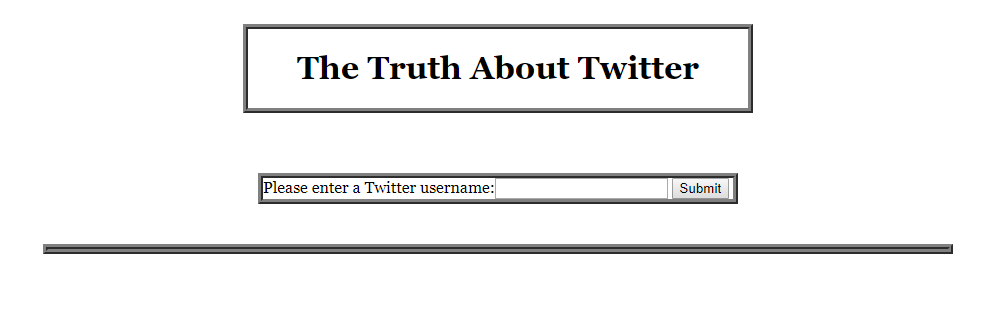


Figure 17 Web Front End

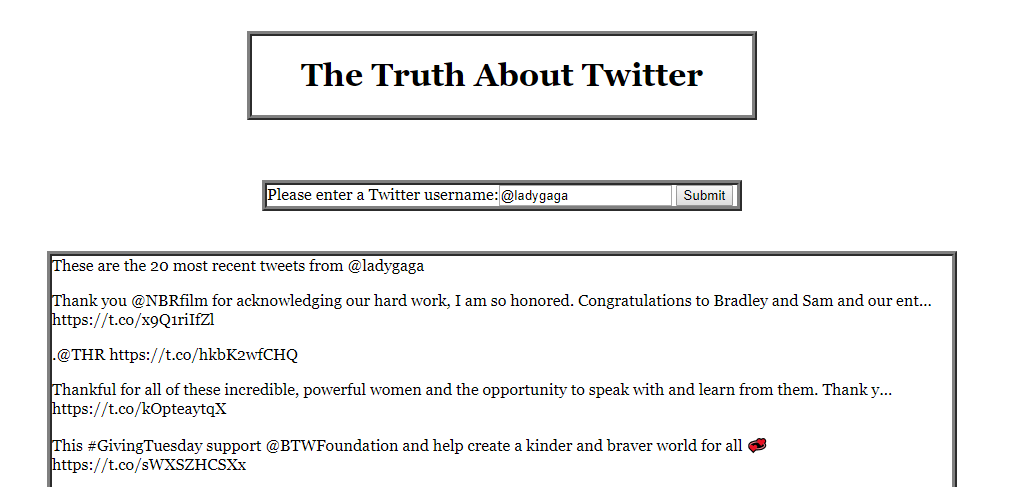


Figure 18 Web Front End with returned tweets

### Basic Machine-Learning Model

For this section of the vertical prototype, the goal was to have a basic machine-learning model that uses data from the acquired datasets and using k-fold cross validation, uses the entirety of the selected data as training and testing sets, outputting, to console, the average accuracy result across the partitions, repeated five times.

Seven features were chosen for this initial model and will be re-evaluated and changed further into development: Whether the account has the default profile picture, has a screen name, has a description, has less than 30 friends, has more than 1000 friends, has never tweeted, the account is geo located and the ratio of friends to followers is 3:1.

A Naïve Bayes classifier with Bernoulli distribution was chosen for this as it a good classifier to start with any data mining project and the inputs are of a binary format, 0 and 1’s with 2000 accounts chosen, 1000 random genuine accounts and 1000 random traditional bot accounts. Below is the output from this completed section the accuracy sitting at around 60% depending on which accounts are selected at the start.

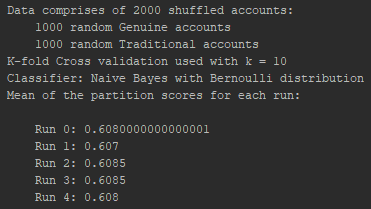


Figure 19 Results of Model being run

## Development

This section deals with all the development done to date in the creation of the vertical prototype using the Django framework and various python libraries such as Tweepy and Scikit-learn.

### Web Front-End

There were several steps to creating the web front-end shown above in the previous section and each will be explained with code snippets where needed.

A new Django project was created within PyCharm Professional, allowing a lot of the tedious groundwork for a web application to be taken care. This meant a bare-bones skeleton app was ready for use and to be built upon.

Next a Twitter developer account was created using my own Twitter account and a Twitter app created, noting its Consumer Key and Secret Key. An Access Token and Access Token Secret were created and noted next. These were outputted to a json file: twitter\_credentials.json, using twitter\_credentials.py, for use later.

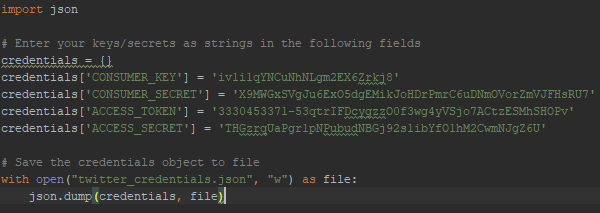


Figure 20 Saving Twitter credentials to file

Next are all changes made to the Django skeleton code:

* Creating a basic form called UsernameForm, in forms.py, to take in user input.



Figure 21 Basic form

* This form was added to views.py within the index method. If the request method was POST, the existing form instance was read in and the user input read into username for use later otherwise a new form instance was created.

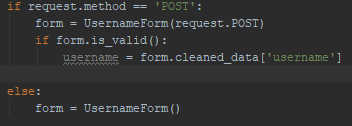


Figure 22 Creating or reading in form

* The values from twitter\_credentials.json are read in and are used in combination with the username inputted in the form to access the Twitter API to return the most recent twenty tweets from that username.[22]

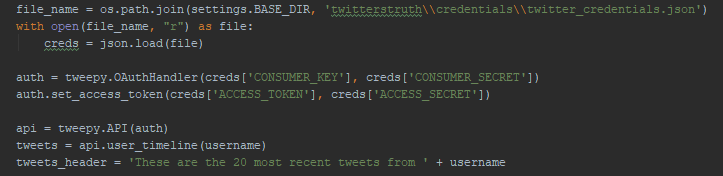


Figure 23 Twitter Authentication and tweet retrieval

* Then a response is returned, asking to render index.html with passed variables: username, tweets and tweets\_header for use in the Html file.



Figure 24 Render Index.html with variables passed

* The templates section of settings.py was altered so to know where to look for template files suck as index.html

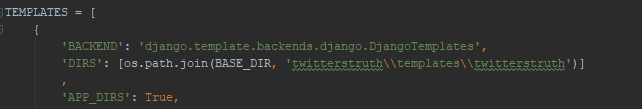


Figure 25 Template directory added

* Index.html was been altered to show the form and tweets using the variables passed to it via views.py

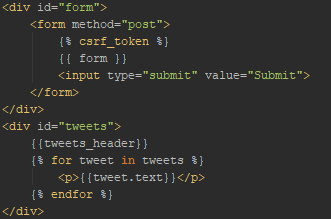


Figure 26 Index.html

* Index.py uses a CSS file web\_style.css, which is located within the twitterstruth/staticfiles/css sub-directory. [46]



Figure 27 Load web\_style.css

* To enable this CSS file, and others within the twitterstruth/staticfiles sub-directory, to be found urls.py was altered [46] :

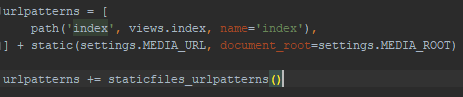


Figure 28 Enabling CSS file load part 1

* As was settings.py [46]:

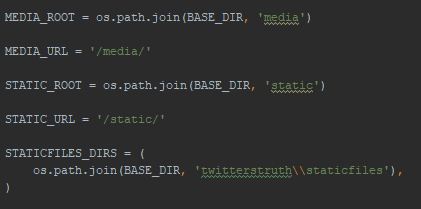


Figure 29 Enabling CSS file load part 2

### Basic Machine-Learning Model

There were several steps to creating the basic machine-learning model shown above in the previous section and each will be explained with code snippets where needed.

* As the files for this section are being run separately but still need access to certain files within the Django framework, the environmental variable, DJANGO\_SETTINGS\_MODULE, must be set and Django setup within each file:

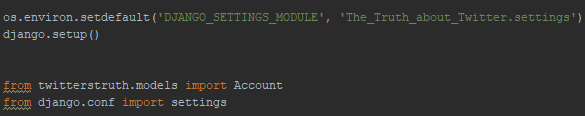


Figure 30 Enabling Django file use

* The file read\_store.py deals with all reading in and storing of the datasets.
* Each dataset is read in from their CSV files one at a time using the pandas library [18], with the tweets files being ignored until the next phase of development.

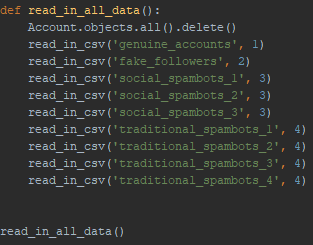


Figure 31 Read in all data

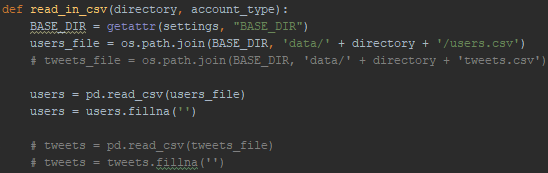


Figure 32 Read in single dataset from CSV

* Checks are done on certain columns in the dataset and binary outputs given depending on the result to form the data going into the database:

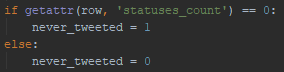


Figure 33 Data check example

* The data is then read into the database:

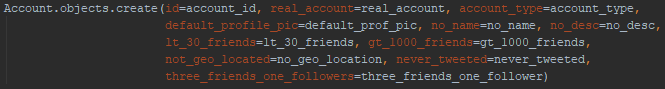


Figure 34 Adding data to database

* In machine\_learning.py, data is read out from the database and passed to through a model, giving output to the console of the model’s accuracy with each run.
* All the randomly chosen, accounts are read out from the database, being split into features and corresponding targets lists:



Figure 35 Get 2000 accounts

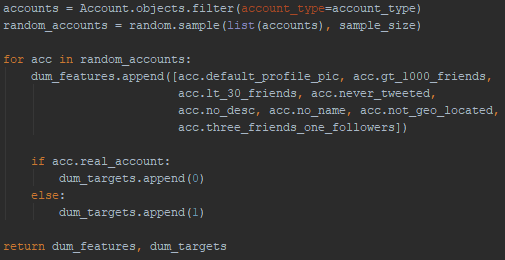


Figure 36 Return random sub-set of accounts from database

* The lists were converted into NumPy arrays [19] and the sklearn library used for k-fold cross validation and classifiers initialisation [17]:

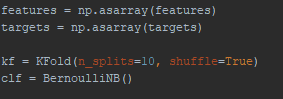


Figure 37 Convert arrays and initialise model

* This model was run five times, with the mean accuracy score across the partitions outputted each time:



Figure 38 Run model and output results to screen

# Testing

This section will explain how testing for the various parts of this project will be done. The testing is split up into 3 parts: The data mining and machine learning section, the web front-end and lastly the fully integrated combination of these two parts.

## Data Mining & Machine Learning section

This part will be employing the K-fold cross validation procedure, talked about during the research stage of this document, to perform my testing on all the models that I build or tweak in this project.

This method involves partitioning up a dataset into K partitions of equal size and for each one, taking that as the testing set with the remaining partitions as the training set.

The dataset I am using is already divided up into multiple sub sets. One of these represents a collection of real Twitter accounts while the rest represent different types of bot accounts. In turn, each of the bot datasets will be mixed separately with the dataset of real accounts and K-fold cross validation will be applied ensuring that the models are trained and tested using the entire mixed dataset each time.

## Web Front-End

This part will be relatively simple and as such, until the previous part is integrated with this one, taking on the role of tester and trying to break every part of it should suffice.

## Web-based Data Mining Application

Once everything is integrated together I will again take on the role of tester while also asking for several testers from my friends and family to help find any bugs that might arise from the integration stage or missed at an earlier stage.

# 4. System Development

## 4.1. Introduction

This chapter continues the issues explored in the previous chapter, and will outline the development process undertaken in this project …

## 4.2. Software Methodology

## 4.3. Technical Architecture

### 4.3.1. Front-End Development

Screen development, user interface design

### 4.3.2. Middle-Tier Development

JDBC or ODBC

### 4.3.3. Back-End Development

Creating the database, configuring the environment

## 4.4. Software Test plan

## 4.5. Conclusions

This chapter discussed the development process involved in this project, it started by outlining the software methodology, which is .. it then looked at the technical architecture …

# 5. Testing and Evaluation

## 5.1. Introduction

Types: Classifiers + others

This chapter discusses the testing and evaluation of the system. In this chapter we will using the term *Testing* to refer to our own appraisal of the system, and *Evaluation* to refer to the appraisal of the system by other people and other pre-existing metrics. In this chapters the Testing will include … and the Evaluation will include …

## 5.2. System Testing

Not just Black Box and White Box, look at Unit Testing, Module Testing, Subsystem Testing, Integration Testing, and Acceptance Testing.

Automated Testing Tools.

Also the Test Plan again.

## 5.3. System Evaluation

Including some of the following:

* By users
* With automated tools
* Using rubrics
* Using benchmarks

## 5.4. Usability Evaluation

## Demonstation

## 5.5. Conclusions/refelection

This chapter reviewed the testing and evaluation of the system. The testing included … The evaluation included.

# 7. Conclusions and Future Work

## 6.1. Introduction

In this chapter the key lessons learned from this project will be discussed. First a series of conclusions will be presented, starting with … Following this, a discussion of some further areas of research based on the work in this project will be discussing, including …

## 6.2. Conclusions

About 6-10 key conclusions

* 2-3 from Chapter 2
* 2-3 from Chapter 3
* 3-4 from Chapter 4
* 3-4 from Chapter 5

Each one 200-300 words

Conclusions = Summary + Justification (“because”)

Include diagrams in about 33% of them

## 6.3. Future Work

About 6-10 future work ideas

Each one 200-300 words

Include diagrams in about 50% of them

# Bibliography

1. 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.
2. Bramer, Max; (2016) “*Principles of Data Mining*”, Salmon Tower Building New York City: Springer.
3. Kelleher, John; Mac Namee, Brian; D'Arcy, Aoife; (2015) “*Fundamentals of machine learning for predictive data analysis*”, Cambridge Massachusetts: MIT Press.
4. 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.
5. 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
6. Martin, Robert C.; (2013) “*Agile Software Development, Principles, Patterns, and Practices*”, London, Pearson.
7. Manifesto Authors, (2001), Manifesto for Agile Software Development, agilemanifesto.org, Date Accessed: October 2018
8. Anderson, David J.; (2010), “*Kanban: Successful Evolutionary Change for Your Technology Business*”, Seattle, Blue Hole Press
9. Toyota, (2004), Toyota Traditions, www.toyota-global.com/company/toyota\_traditions/quality/mar\_apr\_2004.html, Date Accessed: October 2018
10. Atlassian; (-), Trello, trello.com/en, Date Accessed: October 2018
11. Wirth, Rüdiger; Hipp, Jochen; (2000) "*CRISP-DM: Towards a Standard Process Model for Data Mining*".
12. SAS, (-), Data Mining Using SAS® Enterprise Miner™: A Case Study Approach, Third Edition, support.sas.com/documentation/cdl/en/emcs/66392/HTML/default/viewer.htm, Date Accessed: October 2018
13. Azevedo, A. I. R. L.; Santos, M. F.; "*KDD, SEMMA and CRISP-DM: a parallel overview*", IADS-DM, (2008).
14. Gamma, Erich; Vlissides, John; Johnson, Ralph; Helm, Richard; (1994), “*Design Patterns: Elements of Reusable Object-Oriented Software*”, Boston Massachusetts, Addison-Wesley Professional
15. The Django Book; (2018), The Model-View-Controller Design Pattern, djangobook.com/model-view-controller-design-pattern/, Date Accessed: October 2018
16. Catherine; (2013), Include CSS and Javascript in my django template, stackoverflow.com/questions/15491727/include-css-and-javascript-in-my-django-template, Date Accessed: November 2018