Department of Computer Science

Analysing and Presenting the General Public Opinions of Feature Films through Data Mining from Social Media Feeds and a ChatBot

Student Name: Thomas Braund

Student Number: 23003330

Supervisor: Dr Varun Ojha

Date of Submission: 29/04/2019

Abstract – Data mining is used to discover patterns and trends in big data to predict outcomes. The current problem faced in the modern world is how can that data be extracted and used to help improve the standard of living. This project focuses on the use of social media mining, which is a form of data mining, to get the publics opinions of films from social media. This data that is mined will then be analysed and presented to a user. A ChatBot will be created to be the programs interface and will present the data. Multiple data visualization methods are used to make the data more presentable. To create this system the Python programming language along with its various libraries and tools will be used. The data will be stored in a locally hosted MySQL database. This report explains the methodology and implementation done to create the social media mining system. The report will contain sections explaining; the problem, solution approach, design, implementation, testing, discussion and conclusion. The goal of the report is to explain the process of creating the system and to present the results that were gained through implementation and testing.

Acknowledgements

I would like to thank Dr Varun Ojha for the guidance and supervision he provided throughout the project process.

Table of Contents

Glossary of Terms and Abbreviations

AI – Artificial Intelligence

UI – User Interface

API – Application Programming Interface

PID – Project Initiation Document

CNN – Convolutional Neural Network

AIML - Artificial Intelligence Mark-up Language

NLP - Natural Language Processing

ASCII – American Standard Code for Information Interchange

DBMS – Database Management System

SQL – Structured Query Language

NLTK - Natural Language Toolkit

TF-IDF - Term Frequency-Inverse Document Frequency

IDE – Integrated Development Environment

1. Introduction

In the modern world of ever improving technology and increasing internet popularity, data mining has become increasingly influential. It can be used for a variety of reasons and by multiple different sectors. Examples of this could be; from service providers such as mobile phone companies to predict customer churn. Also from E-commerce companies like Amazon which uses it to find what kind of products customers will be interested in so they can target certain customer types (Thelwell, 2015). These are just a couple of examples of the vast number of ways that data mining can be used in the modern world to be able to collect useful data for almost any company or business.

This project aims to use data mining techniques in order to mine social media feeds to gather data on what the general public's opinions of feature films are. Social media is a form of web application which is dedicated to community based input and interactions (Rouse, 2019). Hence it is seen as a great source of big data for a variety of organizations to gather data from. This is due to the fact that every day millions of people post opinions on their social media feeds about anything and everything. Once a person has posted this opinion, unless specified it becomes public data and is free for companies to mine and use to help them gather information and research trends.

ChatBots are AI programs that can communicate with a human user through text or speech (Nagar, 2019). A ChatBot will be used to present the feature films data to users of the program due to its ability to allow for the user to make specific requests for pieces of information. The ChatBot will be used as the UI for the program and will be the layer in-between the user and the data mining application. Through using the ChatBot the user will be able to request information on certain films and will be presented with the processed data that has been found by the data mining program.

Current similar systems that present users are sites such as IMDb and Rotten Tomatoes. These are sites that have a collection of movies in a database with facts, reviews and ratings given to those movies. Whilst these sites contain a variety of information about a huge amount of various movies, most of the reviews and ratings given are from avid movie watchers and goers. Through taking social media data about feature films and presenting that to users, it appeals to the more average movie goer who may not share the same sort of opinions about movies as more enthusiastic movie goers. Also the system created for this project will give a new way of showing the movie data by using the ChatBot and social media data mining. This will allow for better data visualization and popularity statistics to be presented to the users.

The following aims and objectives have been set for this project:

- Set up a database that is able to handle and quickly store the data that is collected from social media.
- Have the database hosted locally so to be easily accessible by the program.
- Use data mining techniques to gather people's opinions about feature films from social media sites.

- Pre-process any text data that is taken from social media feeds to make it readable for the machine and the user.
- Use the Python programming language to implement the data mining and database creation.
- Create a Chabot that is able to respond to a user's questions with relevant answers by using the data that has been mined from the social media.
- Create a system that is able to answer user's queries about any movie they are seeking information on.
- Do analysis of the data that is gathered so to understand what it means and how it can be used in the Chabot to create good relevant responses.
- Have output data be visualized so to be able to be easily read and understood for users seeking information.
- Create clear documentation of all stages of creation of the program that clearly shows what the program is and what was done in order to create it.

The following report will contain a problem articulation which will provide a detailed description of the problem being addressed. This will include; the problem statement, problem context, stakeholder and the technical specification. The next section will be the literature review, which is essential part of any project report. It will look into other pieces of work that have similar goals and objectives as the project being done. The literature review will be split into subsections so to look at other pieces of work to do with different parts of the program that is being made. After this section is the solution approach which looks at the chosen methods taken in order to complete the system that was wanted in the problem articulation. In this section will be an explanation of the solutions chosen and why each solution approach was taken. The next section is the design section which explains the design of the system and why the certain design choices were made. In relation to this section is the implementation section which explains how the program was made and implemented. These are the two key sections of the following report due to the fact they provide the most details about the system that has been created. After this is the testing section which explains how tests were performed on the system created in order to get rid of errors and improve the overall standard of the system. Linked to the testing section is the discussion which looks at the results that have been gained from the system and discusses the implications of the results and analyses the overall success of what has been done when compared to initial goals for the system. The penultimate section is the social, legal, health &safety and ethical issues sections where any of these kinds of issues that occur are discussed. The final section of the report is the conclusion and future improvements which summarises the report and discusses what could be done in further work to improve the overall quality of the report.

2. Problem Articulation

2.1 Problem Statement

The purpose of developing this software was to provide a tool that would allow movie fans to find out the general public's up to date opinion on feature films.

Currently there are databases available online such as IMDB and Rotten Tomatoes that provide movie ratings. However these predominantly feature reviews by critics and are based on the first opinions made when the film was first brought out. By taking Twitter data about feature films this provides recent opinions about the films and the opinions gathered may be more relatable to the average movie watcher.

Using a ChatBot coded in Python the gathered Twitter data about feature films will be presented in an easy to understand and well visualized way.

2.2 Description of Problem Context

The problem area being addressed within this project is how data is mined from social media platforms such as Twitter and then how to present that data to a user in a well visualized way. In this instance a ChatBot is the predominant method used to present the data that is collected. This problem was solved by creating a system that was able to respond to most user queries about any feature film that they are interested in. Below are the constraints and assumptions that are to be considered when assessing if the project managed to achieve this goal.

2.2.1 Constraints

The following constraints are to be considered in attempts of meeting the overall goal of this project:

- Can't have a server to consistently collect data from Twitter about the movies.
- The database won't be online and just hosted locally.
- Database will have a limited data size due to there being not enough time to build it up; this will affect accuracy of results.
- ChatBot will only be able to be run in the program console as not enough time to develop a proper UI for it.
- Every Tweet search can only give back a maximum of 100 Tweets due to the way Twitter API is designed.

2.2.2 Assumptions

The following assumptions are to be considered in attempts of meeting the overall goal of this project:

- Only 1 developer will work on the system.
- Users will be able to follow instructions provided by the ChatBot.

- Users will only input actual movies and not abuse the system.
- There will be Tweets about any film a user is seeking information on.

2.3 Key Stakeholders

Key stakeholders for this project were people looking to find out information about feature films, anyone who have an interest in ChatBots and the people who made the project happen.

The most important concerns for those looking to find out information about feature films would be, firstly getting accurate information about the film they are looking for displayed clearly and quickly. This means that the program needs to be able to look up the information required fast and then visualise that data in an easy to understand way. A second key concern for these types of stakeholders will be having an easy to understand interface with the ChatBot. This means that the ChatBot program will have to provide clear instructions about what needs to be input and have a way of error checking in case an incorrect response is given. A final key concern will be to be able to find out information about any particular feature film they are interested in. This will mean that the program should be able to find information on any film that is queried and at least display some kind of results even if the data is minimal.

For the stakeholders interested in the workings of the ChatBot the key concern will be the functionality of the ChatBot. This will include such things as how it responds to certain queries, its error handling and how it processes different kinds of texts. Therefore the ChatBot needs to run smoothly and be thoroughly tested to make sure it runs as expected. A further key concern for these stakeholders will be how the ChatBot gets the information that it displays to the user of the program. Hence a database was created to store the Tweet data in an organised and efficient manner.

Further key stakeholders in this project will be the developer (Thomas Braund), project supervisor (Varun Ojha) and the work assigners (University of Reading). The developer is responsible for the management of the whole project from beginning to end and develops all software used for the program. The project supervisor is responsible for checking the work and guiding the developer to help create the final program. The work assigner is who asked for the program to be created and is responsible for assessing the overall project that has been done by the developer. Hence all of these stakeholders are key to the project's success and have a vested interest on the final outcome of the program.

2.4 Technical Specification

When creating the technical specification for this project certain aspects of the project previously done and looked into were taken into consideration to be able to create a realistic and achievable specification. Firstly the PID was looked at, in particular the original project objectives and specification. Due to changes made from the PID to actually starting the project some of the original objectives and specifications have had changes and this was taken into consideration. A further piece of prior work that was looked at to help create the technical specification was research into previous systems. This was used as a kind of basis to

what sort of tools should be in the final product and what the industry standard for this kind of product is. Using information from looking at these prior pieces of information the following technical specification was created:

- The system will be able to search Twitter for any movie the user wants.
- A database will be created that is hosted locally and can be accessed at any point in the program.
- The database will contain no duplicate Tweets in it.
- The database will allow for easy and fast searches for relevant data to a user's query.
- All Tweets stored in the database will have clean and easy to read text.
- All Tweets in the database will have usernames and hyperlinks removed to allow for user privacy.
- Only Tweets that are in written in English will be stored in the database.
- The system will be able to understand the sentiment that the Tweet is being written in.
- The ChatBot will be able to respond to every user query that it is given with a relevant response.
- The ChatBot will provide clear instructions for the user at all points in the program.
- The ChatBot will be able to call on relevant functions in the program to deal with the user query.
- The ChatBot will be able to understand the sentiment the user query is being asked in.
- The ChatBot will be able to present the data the user seeks in a presentable and easy to understand way.
- Data visualization methods will be used to help with the clarity of the data that is presented.

3. Literature Review

In this section a literature review of topics relevant to the project was completed. This was done by splitting the project into its key sections and finding relevant work about the section that has been completed within the past 5 years. Each section will look at relevant papers that explain what they have used the technique for and how they have implemented the technique to achieve their goal.

3.1 Data Mining from Social Media Feeds

The first section of literature that was looked into focuses on the use of data mining techniques to gather information from social media feeds. Due to this project focusing on extracting data from Facebook and Twitter, it was a goal to find literature that has also used these two sites when doing data mining, thus making the information more relevant to the project. Data mining from social media has been very much a fast growing industry in recent years due to being able to get large amounts of data about people's opinions on a variety of different subjects relatively easily whilst being cost effective. Therefore it has been very much used as a commercial tool by a majority of companies for finding out there target audiences opinions and trends on the market and their own products.

3.1.1 Mining Twitter Data

Firstly when looking into examples of how to mine Twitter data a paper looking to understand students learning experiences was found which used data mining from Twitter as their primary source of data gathering (Chen, 2014). When attempting to mine the data from Twitter their method was to take an exploratory approach using Boolean combinations of keywords. An exploratory approach is an initial step when doing data analysis and is a way to discover patterns and find anomalies. To improve their data search in this project, it involved expanding and refining the keywords used along with combining the Boolean logic iteratively. Another interesting note from the article is the use of the hashtag on Twitter to improve the data gathering. Twitter users can put a hashtag symbol in front of a tweet as a keyword to describe the topic of the user's tweet (Nations, 2018). Thanks to finding a hashtag pattern among certain tweets it allowed for the project to bring up more results when performing the data mining. This shows that the use of hashtags when mining Twitter data is a key part and should be considered to be implemented into any project looking to gather data from the site.

In another paper that focused on mining Twitter to gather data on the unemployment crisis a very similar methodology was used (Nirmala, 2015). A Twitter API was authenticated by the Twitter system to allow them to access the Twitter servers and mine the data. For this project they solely used the hashtag system to find the relevant data they were looking for. Due to the larger nature of the subject that they were studying compared to (Chen, 2014) it meant that they were still getting a large enough sample just from using the hashtag system. Considering the popularity of feature films it can be considered that just using hashtags may also be a viable way to mine data in this project.

3.1.2 Mining Facebook Data

When looking into the mining of Facebook data it was found that it was harder to gather relevant data because a lot of peoples posts are private and their isn't a way to really search for a collection of public posts. However it was found that the easiest way to gather data on Facebook was to target groups relevant to the target subject. Facebook has a variety of types of groups from fan pages to businesses pages. Therefore for a project about mining data of people with rare diseases, a search of groups was created where certain parameters were needed to be met (Reguera, 2017). Parameters such as certain key words to do with the diseases and a minimum amount of people were used to do this. Once the correct groups had been found the data was then taken from them that were needed for the project.

A different method to find a reasonable data sample would be to select or own a group with enough user activity to be able to collect the data for the project. For example when mining student opinions on the educational system a Facebook academic group was used which had a lot of students creating posts and comments (Tanwani, 2017). This method will remove a step when trying to mine the data and as long as a there is a group with enough relevant data, would be the preferred method of the two.

In reference to the actual gathering of the data from the Facebook groups the methodology is very similar to how it is done on Twitter. The process is started by creating an App and signing in through Facebooks developer website so to be allowed to access and use the public data (Reguera, 2017). In this process the developer is given an access token *ak* which will be used to send a request to a specific server depending on where they are looking to collect the data from (Chen , 2017). From there key words can be used in a search query to collect the data from posts and comments.

The unique feature Facebook contains is that users are able to react to things posted on the site. 'Reactions are Facebook's line-up of emoji that allow you to react to posts with six different animated emotions: Love, Haha, Wow, Sad, Angry, and, the classic Like' (Russell, 2017). This allows for another way to analyse the data where it can be seen what opinions and sentiment are by looking at how people react to a post.

3.2 Machine Learning

Machine learning can be defined as a type of AI where machines are able to perform processes that they have not been specifically programmed to do. The program learns from receiving large amounts of data to improve its algorithm and produce improved outputs overtime. There are a few different ways that machine learning can be used in Computer Science but this section will just focus on techniques that are based around data mining.

Deep learning is a subfield of machine learning and is the practice of training large neural networks to handle large amounts of data with increased performance. It is called deep learning due to the networks having many layers so it creates a deep network such as the CNN. Using deep learning allows networks to continuously increase performance as the

amount of data fed to the network is increased (Ng, 2015). This makes the practice suited to data mining projects that deal with large amounts of data.

3.2.1 Convolutional Neural Networks

CNN is a sub class of neural networks that are used to find patterns within data. Key features of CNN are that; they require fewer parameters than connected networks, they automatically learn from the input domain and contain many different layers (Brownlee, 2016). These layers are split into 3 types; the convolutional layer which is comprised of filters (neurons) & feature maps, the pooling layer which reduces the feature map & takes the most important information and the fully connected layer which uses a softmax activation function in the output layer (Kam, 2016).To train the CNN to get the output that is wanted a large input of data is put in and the network will use backpropagation to learn from the data to continue to improve (Deshpande, 2016).

The main way that CNNs are used in data mining projects is by using them for sentiment analysis. For example in a project which predicts broadband internet customer churn rate, it is used to find whether a positive, negative or neutral sentiment is given by the person who is tweeting (Napitu, 2017). This is done through pre-processing the data taken from Twitter and then feeding that processed data through the CNN to get a desired output.

Another way that CNNs can be used in data mining is by having it sort information into certain specified categories. For example taking tweets about 2016 electoral candidates and sorting them into which candidate the tweet suggests a vote for (Heredia, 2017). This is similar to the previous example and shows that CNNs are good for taking the processed data and finding out what is really meant from that piece of data.

3.3 ChatBot

ChatBot is a virtual person that is able to interact with human users via an instant messaging platform (Rahman, 2017). The process that the ChatBot uses to be able to properly communicate with a person is; firstly it the ChatBot always needs to receive an initial message from the user, then it will use entity recognition to find out what information is wanted from the message. Alongside entity recognition it will use intent classification to find out the sentiment being used and what context the message is being used in. Finally with all the information gathered it will select a response to reply with to the human user (Rahman, 2017).

To be able to properly communicate with a person the ChatBot implements a variety of computing techniques like; machine learning, AI and NLP.

NLP is a vital part to creating a ChatBot; it is the practice of getting the ChatBot to understand what a human is typing to it. One way NLP has been achieved is by firstly understanding what the sentence entails, then classifying what the domain and utterance intention of the sentence is, then finding what the spatial-temporal context is before finally working out what the emotional context being used is (Oh, 2017). Machine learning is a key part to understanding the sentiment analysis and what the emotional context is. Due to the

complicated nature humans express their emotions it is probably the most difficult part of the human language to understand. Thus when the ChatBot is being fed the data it is important for it to continuous log and learn from the emotional context that is being used at it (Oh, 2017).

In terms of the actual creation of the ChatBot the most common way it is done is by using AIML which has a set of patterns/ queries and their possible responses (Ravi, 2018). The three types of possible AIML classes are; atomic, default and recursive. Atomic is a category where exact matches to a query are given. Default is where wild card symbols are used to coordinate information. Recursive is where exceptional labels are utilized to allude to a recursion to persuade the user to be more particular (Ranoliya, 2017).

4. The Solution Approach

This section looks into the approach that was used in order to meet the requirements made in the problem specification and the technical specification. To help decide how to approach these problems the literature review was used to look at examples of how other similar projects created solutions to the problems. This solution approach was then implemented and used throughout the creation of the program as a foundation as to what needs to be done and guided each step of the program implementation.

4.1 Social Media Data Mining Approach

The first problem to decide was what the approach to gathering the movie data from social media sites was going to be. This included deciding such things as; what social media sites was going to be used to gather the data from, the software that was going to be used to gather and process the data and the database that was going to be used for storing and managing the data.

4.1.1 Social Media Sites to Use

When doing the literature review and looking at similar systems that had been created there was only really two main social media sites that were used to mine data from; Facebook and Twitter. Whilst there are other social media sites available such as Instagram and Bebo these sites either don't have as many active users or has a focus on using images as its main source of sharing information. Therefore this left just Facebook and Twitter as the best available options to use for doing the data mining about movies.

From the literature review it can be seen that whilst Facebook does have many users who actively post to their page the privacy rules of Facebook means that it is difficult to access this data unless you are on the user's friends list. A way of getting round this would be to create a public group that has members actively sharing opinions about films to its page. However this may not be a viable option for completing this project due to a few factors. Firstly due to the limited time frame of the project it would probably not be possible to set up and gain a large enough following of active posters to collect the amount of data needed for this project. Also if were such a page exist it would be more likely to attract people who have a passion for movies and maybe some critics therefore meaning this wouldn't necessarily have opinions that appeal to the average movie follower.

Hence this leaves Twitter as the only viable social media site to use as the site to gather movie data from. The benefits to using Twitter for this are that firstly Twitter has a huge number of active users. According to the site internet live stats (internetlivestats, 2011) around 6000 Tweets are sent every second which corresponds to 500 million Tweets per day. Even when filtering out the non-English Tweets this still leaves a vast amount of data available and with a high chance of there being information on most feature films made. A further benefit of using Twitter as the social media site to get data from is that when a user of Twitter does a Tweet it becomes public data unless the user has set certain privacy options.

This means that most Tweets made are available publicly and are easy to access. From this information Twitter was decided as the best social media site to use for this project.

4.1.2 Software to Use

When looking into what software to use to do the data mining and run the program from, 3 main choices were looked at; KNIME, R and Python.

KNIME is an open-source data analytics software that incorporates data mining and machine learning through a modular pipelining concept. The benefits of using KNIME for this project are that firstly there is a specific KNIME node that allows the developers to get Tweets straight from Twitter. Another benefit of using KNIME is that it provides an easy to use pipelining approach to data management which could take the data from Twitter and then process it so it is presentable to the user. Further benefits of KNIME are; simple to use and numerous data visualization options, can handle very large data and allows for the use of multiple languages. Disadvantages for using KNIME in this project is; would have to be implemented alongside a different source of code to allow for the ChatBot creation and it doesn't have its own programming language.

R is a programming language that was made for statistical computing and graphics. It is largely used for developing statistical software and doing data analysis. Benefits of using R to for doing this project would be; R has multiple libraries available to developers for doing Twitter data mining and doing data analytics. R contains libraries such as 'twitteR' which can be used to mine data from Twitter and 'tm' which can be used to clean text. Another benefit is that it is a popular programming language and therefore has lots of examples of similar projects available online to analyse and use. Disadvantages to using R for this project is that it is mainly a programming language used for purely statistical purposes so doesn't contain many libraries and functions to create a ChatBot. When using R it doesn't have very good memory management and it is possible that R could use up a lot of memory so not good for this type of system which can't afford high memory loss.

Python is a high-level general-purpose programming language that gives the developer numerous options to create multiple different types of programs. Python can be object-oriented and structured. It is simple to use & understand and has a huge amount of libraries available for developers to use. The first benefit of using Python for this task is that it has a number of different environments that it can be ran in, such as anaconda which is a distribution of Python that focuses on doing data science and machine learning methods. Another benefit to using Python would be that because it is a general purpose language it will mean that a ChatBot can be easily implemented into the program and ran alongside the data mining program. This will make the project as a whole easier and simpler to develop. Further benefits to using Python would be; dedicated Twitter data mining libraries, numerous data analytic libraries, easy to use methods for data visualization and the fact that it contains memory management functions. Disadvantages of using Python for this project would be that it doesn't have as good pipelining features as KNIME and there are better languages for making a ChatBot.

From looking at the three possible software's for doing the project and comparing their features to what is wanted from the technical specification it was decided that Python was the best choice for the project. This was due to its flexibility, scalability, memory management and ease of use. Using Python would mean that it is possible to mine the Twitter data, store it in a database and also create a ChatBot all in the same program. This gives Python an advantage over the other two choices and whilst they were all viable options Python was the software that was best suited for this program.

4.1.3 Database to Use

With Python being the selected software used for doing the programming in the project the next step was to pick what DBMS that would be used to store the data. This choice was only really between two different database systems, either PostgreSQL or MySQL. These are the two most common open-source database software's used with Python and are highly compatible with the Python programming language.

PostgreSQL and MySQL are for the most part very similar software's with most of the key differences between them being about higher level functions. They have more or less the same feature set as each other and don't have much differences in terms of speed, scalability and reliability when handling the amount of data that is being used for this project. Therefore when deciding on the database it came down to more the personal preference of the developer and due to having past experience with MySQL it was chosen as the database for the project.

4.2 Program Development Approach

With the social media site and software that was going to be used for the project decided the next set of problems to decide a solution for was about how the program was going to be developed and ordered. This stage of planning the solution approach included deciding; what approach to use for mining the data from Twitter, how the ChatBot will respond to user queries and how the data will be displayed to the user.

4.2.1 Data Mining Approach

When deciding on the approach to get the optimal solution for the data mining system many factors had to be considered. This included such things as; the Python library to use, whether to stream Tweets or do searches when the program ran and the parameters of the search when mining the data from Twitter.

Looking into the available libraries for Python developers showed that there are lots of different libraries available to use when wanting to access the Twitter API from Python. Libraries included in this list are; Tweepy, Birdy, twython and python-twitter. All of these libraries have positives and negatives about them and would probably be able to help achieve the technical specification. However when looking into the different libraries and at other similar projects to the one being done, it was decided that Tweepy was the best to use for this project. This was decided due to the number of functions that the Tweepy library provides, the ease of use of the library and its popularity in other pieces of work meaning there is a lot of resources available to help improve this project.

For deciding whether to stream Tweets from Twitter or to do searches when the program was ran it was decided that it was best to do searches when the program was run. This was because as mentioned in the constraints of the problem articulation there isn't the resources to have the program streaming for long enough to get the required amount of data. Also by using the search function it allowed for more variety in the parameters used for getting the data off Twitter.

The final decision that needed to be made for data mining in the program was how to implement it and the parameters that would be used in the implementation. Firstly it needed to be decided whether to use a limited set of films which the user could choose from where the data had already been mined from Twitter or to have the user type in a movie which would then be searched for by the program. The benefit of using a predetermined list of films would be that the database would be able to be built up with lots of data before the user used the ChatBot therefore meaning a lot of information available to the user. This would mean that the user would be given more accurate results when asking about one of these movies and the program would not need to run the data mining program when running the ChatBot which would speed up the process. The benefits of doing the search from a user input of a film would mean that the user could get information on any film they wanted and that all information would be kept up to date. This would also mean that for each use of the program the database would be expanding and improving. The chosen approach for this was to do the search of what the user input as the wanted movie. This was because it allowed for the user to search for any movie they would like and having a database that is up to date and improving. This matched up well to what was wanted from the technical specification.

4.2.2 ChatBot Implementation

The next part of the approach to decide was how the ChatBot was going to be implemented into the program and how it was going to work. When researching ChatBots available online that have been made in similar projects, it can be seen that there are a variety of ways to implement a ChatBot into programs such as this. Methods such as creating lists of keywords for which will get specific responses or searching for similar terms in text data are commonly used by programs with a ChatBot.

The solution approach decided for this project was to use a variety of techniques when programming the ChatBot to provide a realistic AI experience for the user of the program. This includes having the ChatBot ask for specific inputs, looking for specific keywords in a user's input and having the input compared against Twitter data for similar key terms. Through implementing a variety of these techniques it was hoped that the ChatBot would be able to meet the requirements set in the technical specification.

4.2.3 Data Visualization Implementation

The final part of the program to decide for the approach was how to present and visualize the data that had been mined from Twitter. When looking at how to present the data the most important factor to consider is how the data can be presented in a clear and understandable way. Methods that could be considered for doing this are doing such things as; using graphs

and charts, presenting the numerical statistics, showing the Tweets that have been made or using one of the various visualization functions available in the Python libraries.

The solution approach that was thought to be best for this project was to use a variety of these methods to give the user a variety of visualized data. This would mean that the user would be able to gain a good understanding of what the data that has been analysed means and it also gives flexibility to how each user query is answered. Depending on what kind of query the user gives, the ChatBot will be able to use methods like producing charts or showing Tweets that'll help the ChatBot give the user a satisfactory response to their query.

4.3 Solution Definition

The final solution approach that was decided for this project was to mine Twitter data using Python, store the data in a MySQL database and then present that data in a variety of ways using a ChatBot. Twitter will be the only social media site used to mine the movie data from due to its ease of use and privacy policies. All programming will be done using the Python programming language. This is due to the fact that it has a variety of libraries available for the ChatBot and data analytics, hence it is possible to create all aspects of the project using Python and it is an easy to use programming language capable of producing complicated systems. Then MySQL was decided as the DBMS to use due to its compatibility with Python and the fact the developer had prior experience with it.

For the program itself it was decided to use the search Twitter Tweepy method every time the program was ran so the user could find information on any movie they wish. This leads to an up to date and ever expanding database. Then for the ChatBot it was decided to use a variety of the available ChatBot implementations available due to it giving the ChatBot flexibility and meaning it has a more realistic AI. Finally for presenting the data to the user it was decided to have the data visualized in a multitude of ways that depended on what the user was querying. This would mean that the user would get a better understanding from the output of the ChatBot and could lead to more accurate and easier to understand responses.

On completion of this program it will be tested against the objectives stated, the technical specification and the solution approach that has been defined in this section. The created system will be deemed a success if it meets these requirements and does everything that is specified in a way that is considered understandable and efficient.

5. Design

5.1 MySQL Database

The MySQL database was hosted on the local host using the XAMPP software. XAMPP is an open-source cross-platform web server solution stack. Using XAMPP it is possible to run Apache and MySQL modules. This means that a database can be hosted on the localhost on the phpMyAdmin component. phpMyAdmin is an open-source administration tool for MySQL and MariaDB. It allows users to create and host MySQL databases that can be accessed from programs such as Python.

With phpMyAdmin being used as the host a MySQL database was created on the localhost called ‘projectdb’. In the ‘projectdb’ database a table named ‘movietweets’ was created which would be the table that stored all the Twitter data that is mined and be used and edited by the Python program. This table contains 4 columns of different types of Twitter data; ‘tweets’, ‘retweet_count, ‘favourite_count’ and ‘movies’ as shown in Figure 1. The ‘tweets’ column contains the cleaned Tweet text data that has been mined from Twitter. Every value in this column is unique so not to have repeating Tweets that will skew the data analysis later in the program, therefore giving accurate data results to the user. This column has a character limit of 280 due to the fact that any Tweet made on Twitter also has a character limit of 280. The next column in the table is ‘retweet_count’ this column contains a numerical character of how many times each Tweet has been retweeted. This helps with analysis of how popular each Tweet and movie are. Next to that column is ‘favourite_count’, which shows how many times that tweet has been favorited. The final column presented in the table is the ‘movie’ column. The purpose of this column is to show what the associated movie with each Tweet is. This value is taken from what the user inputs as their choice of movie to look up and has a character limit of 255 which is the default value MySQL tables set it to. This column is useful when doing lookups of particular films for data analysis.

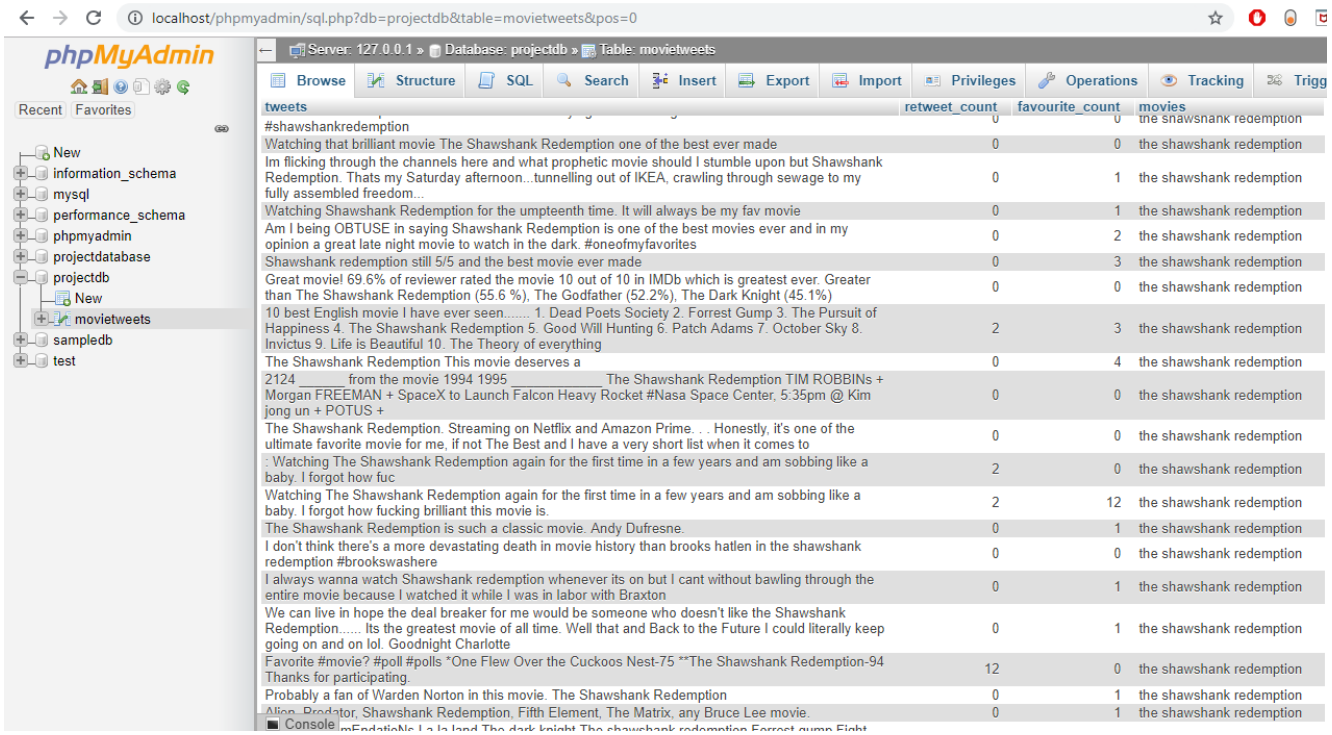


Figure 1: Database table being hosted locally on the phpMyAdmin component

5.2 ChatBot

In the program that has been created the ChatBot is used as the only source of interaction between the user and the program. Thus being the only form of an UI in the system the ChatBot needs to provide clear instructions to the user and output the information that the user is looking for. Hence it means that it was important for this part of the program to have a clear and simplistic design that can easily understood and comprehended by the user of the program.

To achieve this, the first important part was to clearly define the difference between what the ChatBot is outputting and other information that is shown in the program. This was done by labelling every statement output by the ChatBot program with 'ChatBot :'. By doing this it clearly showed to the user whenever an output was being done by the ChatBot program so would not lead to confusion when the program was running.

```
Chatbot: Hello, I will answer your queries about movies. If you want to exit, type bye
Chatbot: What is the name of the film you'd like to know about?
pulp fiction
Chatbot: What about pulp fiction would you like to know?
how good is the movie?
```

Figure 2: ChatBot introducing itself to the user and explaining its functions

Another important part to the design of the ChatBot was to make sure it gave clear instructions and error messages depending on the user input. This needed to be done throughout the program run time so at every stage the user knew exactly what the required input is. This design concept is achieved by having the ChatBot output clear instruction

messages such as ‘Chatbot: What is the name of the film you'd like to know about?’. In this instance the user can clearly see that the required input for this part of the program is the name of the movie that they are seeking information on. Therefore if all instructions made by the ChatBot are clear like in the instance shown here then the user will know what is happening in the program at all times. For when an unexpected input is entered then an error message will need to be displayed to inform the user that they have made an error in their input. For example if a query is asked that doesn’t make sense or doesn’t have any data related to it in the Tweet data then the ChatBot will respond ‘ChatBot: I am sorry! I don't understand you’. This clearly shows that the ChatBot was not able to find a response to what the user has asked. After this output the user will then be given another question to keep the program moving along.

```
Chatbot: Would you be interested to see other statistics for this film? Please enter 'yes' or 'no'
no
Chatbot: Would you like to know anything else about this film? Please enter 'yes' or 'no'
no
Chatbot: Would you like to learn about another film? Please enter 'yes' or 'no'
yes
Chatbot: What is the name of the film you'd like to know about?
paddington
Chatbot: What about paddington would you like to know?
how popular is paddington?
```

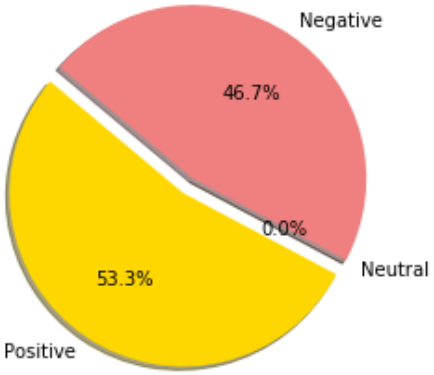
Figure 3: Dialogue between the user and the ChatBot. Shows the ChatBot giving clear instructions to the user at all times

By having these features in the ChatBot a clear UI is made that the user of the program can use and clearly understand. This will keep the program moving forward and improve the overall quality of the program. The ChatBot is displayed in the console of Jupyter Notebook which is the software that the Python programming has been written in. Jupyter Notebook is a web-based interactive computational environment for coding the Python programming language.

5.3 Data Visualization

This program uses data visualization to present the data that has been collected from Twitter to the user in a clear, interesting and easy to understand way. Data visualization is considered a crucial part to any data analytics program due to the way it is able to present potentially complex big data in a relatively simplistic way. Models such as statistical graphics, plots, information graphics, charts and numerous other tools are used to do data visualization. When performing data visualization it is important to choose the right method with relevant data to the point that is trying to be put across.

In this project data visualization is used to get across multiple different pieces of information about the data. The first way this is done is to show the results of performing sentiment analysis on Tweets about the movie the user is interested in. For this instance a pie chart is used which is split into 3 different segments, positive, neutral and negative as shown in Figure 4. Each segments value and slice size is decided by the percentage of Tweets found to have that kind of sentiment.




```
Chatbot: The percentage of positive Tweets found about pulp fiction = 53.333333333333336%
```

Figure 4: Pie chart showing the sentiment analysis results for the film Pulp Fiction

The next part of data visualization used in this program is done to show the related statistics to the movie the user is querying about. This starts with having the most popular Tweet in the database about the film be output to the user. The most popular Tweet in the database is considered to be the one with the most retweets for it. Once this Tweet is found the cleaned Tweet text data with no special characters or usernames is printed to the user.

```
Chatbot: The most popular Tweet about this film is:
Just watched David Lynch's Wild at Heart (1990) for the first time in years. Amazing movie. Reignites the spirit of 50s rock
n roll within a pulp-ish film noir story but feels very Grunge era. Did Tarantino see this before he made Pulp Fiction? Oh ye
s.
```

Figure 5: Most popular Tweet about Pulp Fiction displayed via the ChatBot

A further method of data visualization used is the creating and displaying of a word cloud. A word cloud is a visual representation of text data that highlights frequently used words in text by making them appear larger than other words. This program uses a word cloud to show the most frequent words that occur in Tweets to do with the movie the user is inquiring about. The word cloud can be used further by the user to find out more about the movie by searching the keywords that the word cloud finds. The word cloud also is a good way of conveying the general opinions of the movie by showing key sentiment words like good or bad to the user.

Figure 6: Word Cloud showing the most popular words in the Tweet text data about the movie Pulp Fiction

The final data visualization method used in this program is a bar chart to show the popularity of the movie the user is inquiring about in the database. Once again this is done by retweet count and how many unique Tweets there are in the database about the movie in question. This data is then put in a bar chart of all the different movies in the database against the number of mentions in the database that a particular movie has, as shown in Figure 7. Another part to showing the popularity is that the ChatBot outputs a message saying how many mentions the specific movie the user is inquiring about has. Through doing this it shows the comparative popularity of that specific film whilst also showing how many mentions there is of the movie in the database.

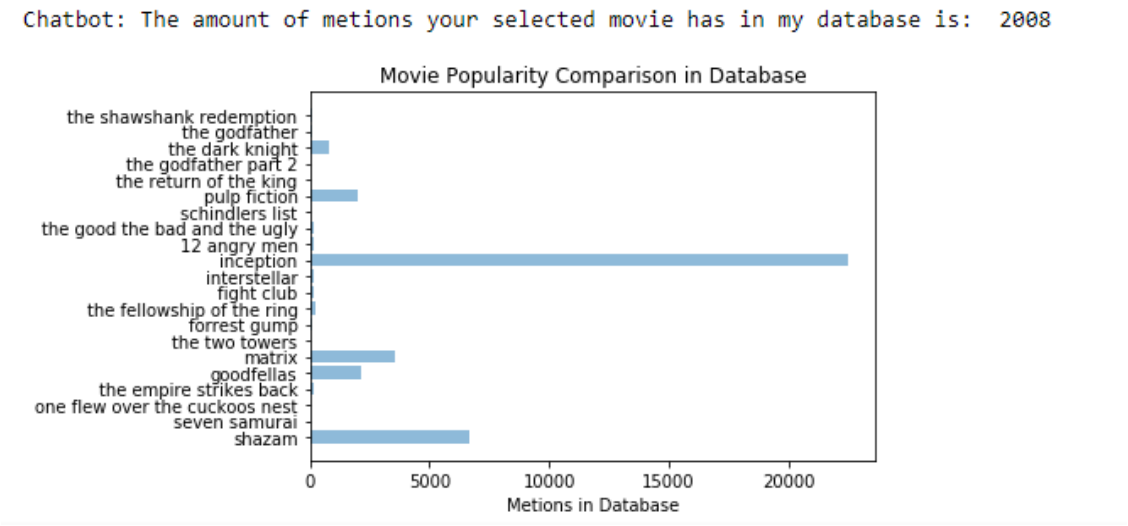


Figure 7: Bar chart and text output to show the popularity of the movie Pulp Fiction in the current database.

Through applying these different data visualization methods the hope is that the program displays the data that is has gathered and analysed in a clear, interesting and easy to

understand way. Various different methods were used to get across different points about the data to the user. By using these different methods it means that the user can get a good understanding of what data has been gathered from Twitter without having to present and explain the database.

6. Implementation

6.1 Twitter Data Mining

This section will look into how the Tweets were gathered from Twitter and organised so to be usable for the ChatBot. Data mining Tweets from Twitter was the first part of completing this project due to the fact that all other aspects of the project needed the data to be able to run.

The first part to doing the data mining from Twitter was to gain access to the Twitter API so to be able to manipulate the public data on Twitter. Hence a developer's account was set up on Twitter that gives the user the ability to create an app. Included in setting up the app for the developers account Twitter required information about the app being created, things such as in what way was the data going to be used and what kind of information will be collected. This was required due to the fact Twitter needed to know what permissions could be granted depending on the project that was being made. Once the app was created and given the required permissions personal keys and tokens were given to the user to be used to access the Twitter API. The developers account made can be seen in Figure 8.

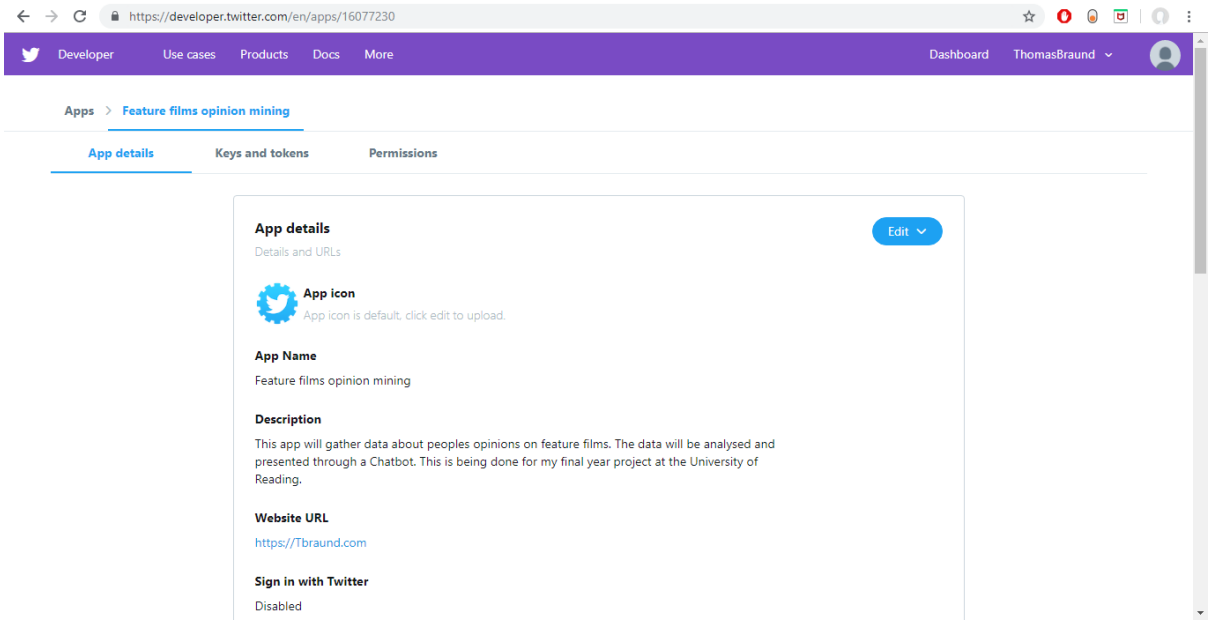


Figure 8: Screenshot of the developers account that has been made to access the Twitter API in the program

With the developers app created and the keys and tokens given it was then possible to access the Twitter API from the Python program. To be able to access the Twitter API from python the Tweepy library was used. Tweepy is an open-source python library that is hosted on GitHub and lets users access the Twitter API. Once Tweepy is installed and imported into the program it can be used to access and manipulate the Tweets. To gain access in the program an authentication check of the given tokens and keys is done by the Tweepy library. If this is successful the program can now use the Twitter API via Tweepy.

The next step for the Twitter data mining process was to use the Tweepy library to search Twitter for the relevant tweets as shown in Figure 9. Tweepy provides an easy way to do this with a method called 'search'; this is a helpful method that gathers Tweets based on certain parameters set by the developer. The parameters that were set in this program was the query which was set to the movie the user was looking up, the 'search' method would then only gather Tweets which had the query within it. Then the language which was set to English so only English tweets were gathered to make the program more simple. The 'tweet_mode' was set to extended so that the whole Tweet text was collected rather than a part of it. Finally 'count' which was set to 100 which is the maximum value, 'count' is used to define how many Tweets were to be returned per page.

```
def get_movie_twitter(movie, tweetdb):
    query = movie, "movie"
    language="en"

    result = api.search(q=query, lang=language, count=100, tweet_mode='extended')
    for tweet in result:
        tweet.full_text = preprocess_tweets(tweet.full_text)
        add_tweet_to_db(tweet.full_text, tweet.retweet_count, movie, tweet.favorite_count, tweetdb)
```

Figure 9: get_movie_twitter function that gets the data from Twitter

With the search method set up the program was able to do the Twitter data mining; hence the next program stage could be created which was pre-processing the data gathered.

6.2 Data Pre-Processing

With the data gathered the next step in the program is to process the data so to make it useable for the further operations in the program. Data pre-processing is an important part of any data mining program due to the raw data gathered being incomplete, error strewn and generally hard to understand. By doing the pre-processing it means that it is less likely they'll be errors at runtime and it allows for the data to be easier to read via the program and make sense when output to the user..

To do the pre-processing the regular expressions re library was used. The re library has a sub method within it, which when given a regular expression will replace designated characters with the characters specified. This can be used to remove unwanted characters from strings amongst other things. To remove the unwanted characters in all instances of 'sub', the designated characters will be swapped with a blank character. The code to do the pre-processing can be seen in Figure 10, which shows the preprocess_tweets function.

The first stage of creating the pre-processing program was to define a method that would take in each tweet one at a time to be pre-processed. The first part removed from the tweet is any username in the tweet. This is done by removing any character set that begins with the @ symbol, which is used at the start of a username on Twitter. Next any web addresses included in the Tweet were removed. Most web addresses linked in Tweets begin with the character set 'https://'. Therefore when removing web addresses, any character set that began with 'https://' was removed. Another regular piece of grammar to be removed from Tweets was 'RT' which is given to any Tweet that is a Retweet. However Retweet grammar comes in

three different forms so to get rid of all versions of this grammar three different types of regular expression substitution needed to be used. Firstly any expression in the form ‘RT :’, then any in the form ‘RTusername’ where ‘username’ would be the Twitter handle of the user and finally ‘RT username’. The different combinations used to do this can be seen in Figure 10. This is the final case of using re in the Tweet pre-processing stage.

The final stage of pre-processing the data is to remove any special characters in the Tweets and to get rid of any extra unwanted spaces in the Tweet. To get rid of any non-normal characters the join function was used along with the use of the ASCII character set. The join method is used to take all variables in an iterable and join then all into one string. ASCII is a standard for setting out what normal characters are and contains 128 normal characters including English letters, numbers and some special characters. By using the join method along with the ASCII character set all non ASCII characters are removed from the string. Finally the join and split methods are used to remove any unnecessary whitespace from the Tweets that have been gathered. The split method is used to split the string into a list and then join can be used to put the string back together without their being any items that have 2 or more characters of space.

```
def preprocess_tweets(tweet):
    tweet=re.sub(r'@[A-Za-z0-9]+','',str(tweet))
    tweet=re.sub('https://[A-Za-z0-9./]+','',str(tweet))
    tweet=re.sub('RT :','',str(tweet))
    tweet=re.sub('RT[A-Za-z0-9./]+','',str(tweet))
    tweet=re.sub('RT [A-Za-z0-9./_]+','',str(tweet))
    tweet= ''.join([c for c in tweet if ord(c) < 128])
    tweet=" ".join(tweet.split())
    return str(tweet)
```

Figure 10: Screenshot of the function used to pre-process the Tweet text data

By doing this the fully cleaned Tweet can be returned in the form of a string and be ready to be stored in the Tweet database.

6.3 MySQL Server

For this project the selected database to store the Tweets data in was MySQL. MySQL is an open source relational DBMS. Reasons for choosing MySQL as the database server for this project was firstly that MySQL is designed and optimized to work with web applications. Due to this project being focused on taking information from Twitter in the program and then storing that data it is useful to have a server that can operate well with web applications. Another reason for using MySQL is that it works well with the Python programming language and has many different functions and methods to make it easy to use via the Python code. This means that accessing, storing and manipulating the data can be easily done in the programming language. A final reason for wanting to use MySQL in this project is that it is known to be one of the best database servers available and that is shown through its popularity. Reasons that make MySQL so popular when compared to other database servers are its; performance, scalability, robustness and security features.

To be able to set up and connect to the MySQL database a server was created and ran using the XAMPP software. XAMPP is an open source cross-platform web server solution package. XAMPP allows the user to set up and run a local test server with multiple different modules, for this program only Apache and MySQL will be used. The XAMPP console used to run the server can be seen in Figure 11. Using XAMPP, a local test server is created and a database for the project is made and named 'projectdb'. A table can be then created in this table; this is given the name 'movietweets'. The table contains 4 columns 'tweets' which contains the cleaned tweets gathered from Twitter, 'retweet_count' which shows how many times the tweets has been retweeted, 'favourite_count' which shows how many favourites a Tweet has and 'movies' which shows which movie the Tweet is about.

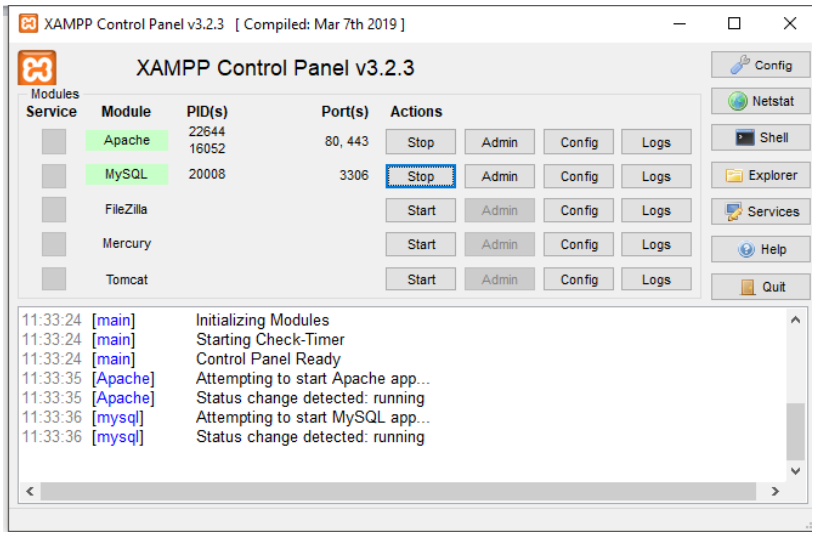


Figure 11: XAMPP control panel used to create the localhost server for the MySQL database

With the MySQL server created and containing the required database and table it can now be used to store the data gathered. The first step done in order to do this is to connect to the database in the Python program as shown in Figure 12. To do this the connector function in the MySQL library in Python is imported into the program. Using the connect function in this library the database details can be entered in and the program can attempt to establish a connection to the MySQL database that has been created. An exception handler is put into this function in case connection cannot be established for some reason, in which case the exception error is printed out. When the database connection is established the program could then access the table and be able to manipulate the data in it.

```
def connect_to_database():
    try:
        tweetdb = mysql.connector.connect(host='localhost', user='root', password='', charset = 'utf8', database = 'projectdb')
        if tweetdb.is_connected():
            return tweetdb
    except Error as e:
            print(e)
```

Figure 12: Screenshot of the function used to connect the database to the program

To add the pre-processed data to the database table a function 'add_tweet_to_db' is made that takes in a single tweet, it's retweet count, favourite count, the movie and the connected database cursor as seen in Figure 13. The cursor is what the connected database has been

connected to in the program. The cursor contains a method called execute which will execute the input MySQL statement to the database. This is used to input the tweet, retweet count, favourite count and movie that have been given to the function into the database. Before the data can be input into the table the Tweet text data is firstly checked against the existing Tweets text data in the database to make sure it is not a repeat. This is so that when presenting data later in the program, it is not skewed by having repeating values. To check whether a Tweet is already contained in the database firstly the data is filtered down to just Tweets that are about the same movie. The Tweet being input is then compared against the current Tweets within the database and if it is not matching to any existing Tweet it can be inserted into the database. This is done by writing in the correct SQL statement into the cursors execute method before using the commit method which commits the current transaction to the selected database.

```
def add_tweet_to_db(tweet, retweet_count, movie, favourite_count, tweetdb):
    cursor = tweetdb.cursor(buffered=True)
    cursor.execute("SELECT * FROM movietweets")
    cursor.execute("SELECT tweets, COUNT(*) FROM movietweets WHERE tweets = %s GROUP BY tweets",(tweet,))
    row_count = cursor.rowcount
    if row_count == 0:
        query = "INSERT INTO movietweets (tweets, retweet_count, favourite_count, movies) VALUES (%s, %s, %s, %s)"
        cursor.execute(query, (tweet, retweet_count, favourite_count, movie))
        tweetdb.commit()

    cursor.close()
```

Figure 13: The add_tweet_to_db function used to insert Tweet data into the MySQL database

Each time the 'add_tweet_to_db' function in Figure 13 is called it will check and add the tweet data given as the argument. Therefore the more the program is used it will create a larger database of Tweets about the films; hence this continuous use will improve the accuracy of the results given over time. With the data gathered and stored in the MySQL database it can now be accessed and used by other functions in the program.

6.4 Sentiment Analysis

Sentiment analysis is the name given to the NLP method where a machine attempts to understand the opinion given to it in the form of text or speech. For this program sentiment analysis is used to attempt to understand what is the opinion being conveyed from the Tweet being analysed. For this to be achieved in the system a Python library called TextBlob is used. TextBlob is a library that is used for processing textual data; the library contains a number of features including parsing, classification and tokenization. For this program only the sentiment analysis part of the library will be used.

To do sentiment analysis using TextBlob the sentiment and polarity functions need to be called. By using these functions a number will be returned, if the value is higher than 0 the sentiment is judged to be positive, if it is equal to 0 it is neutral and if it is less than 0 the sentiment is negative. Through this the system Tweets text data about the chosen movie can be analysed and the program can make a judgement on what sentiment each Tweet says about the film. This will create a list of whether the Tweets are 'positive', 'neutral' or 'negative'. The name of this function is 'tweet_sentiment' and can be seen in Figure 14. This can then be

returned to another function that checks the overall polarity of the Tweets about the movie and outputs the results. This function is named ‘sentiment_check’ and is shown in Figure 15.

```
def tweet_sentiment(movie, tweetdb):
    tweet_sent = []

    cursor = tweetdb.cursor(buffered=True)
    sql = "SELECT tweets FROM movietweets WHERE movies = %s"
    cursor.execute(sql, (movie,))
    tweets = cursor.fetchall()

    for tweet in tweets:
        analysis = TextBlob(str(tweet))
        if analysis.sentiment.polarity > 0:
            tweet_sent.append('positive')
        elif analysis.sentiment == 0:
            tweet_sent.append('neutral')
        else:
            tweet_sent.append('negative')

    cursor.close()
    return tweet_sent
```

Figure 14: Screenshot of the function used to assign the sentiment of Tweets in database

The way that the ‘sentiment_check’ function works is that firstly it calls the previously described function and is parsed a list of Tweet polarities about the movie being tested. The function then takes the user input it has been given and matches it against lists of keywords to work out what kind of sentiment the user is looking for. For example if keywords such as ‘good’, ‘great’ or ‘like’ are used it is presumed that the user is looking for positive sentiment information. With the user sentiment decided the program will then use the created list of Tweet sentiments to get the overall percentage of Tweets that have the sentiment that is being looking for. This is done by using the following equation: ((Number of tweets with sentiment looked for) / (Total number of tweets analysed)) * 100. This creates the percentage of Tweets with the required sentiment and can be output to the user.

```
POSITIVE_INPUTS = ("good", "great", "positive", "superb", "worth", "excellent", "wonderful", "like", "rating")
NEGATIVE_INPUTS = ("bad", "awful", "terrible", "poor", "lousy", "dreadful", "rough", "atrocious")
NEUTRAL_INPUTS = ("okay", "alright", "average", "acceptable", "fine", "tolerable", "ok", "passable")

def sentiment_check(user_response, movie, tweetdb):

    tweets_sent = []
    tweets_sent = tweet_sentiment(movie, tweetdb)
    chatbot_response=''
    ppercent=(tweets_sent.count('positive')/len(tweets_sent))*100
    npercent=(tweets_sent.count('negative')/len(tweets_sent))*100
    neupercent=(tweets_sent.count('neutral')/len(tweets_sent))*100
    user_response = re.sub(r"[^A-Za-z]+", ' ', user_response)
    for word in user_response.split():
        if word.lower() in POSITIVE_INPUTS:
            chatbot_response=chatbot_response + "The percentage of positive Tweets found about " + str(movie) + " = " + str(pper
        elif word.lower() in NEGATIVE_INPUTS:
            chatbot_response=chatbot_response + "The percentage of negative Tweets found about " + str(movie) + " = " + str(nper
        elif word.lower() in NEUTRAL_INPUTS:
            chatbot_response=chatbot_response + "The percentage of neutral Tweets found about " + str(movie) + " = " + str(neuper
    if(chatbot_response == ''):
        return None
    else:
        visualize_sentiment(ppercent, npercent, neupercent)
        return chatbot_response
```

Figure 15: Program used to work out movies sentiment percentages and output the results

6.5 Query Handler

This section looks into how queries that aren't about the sentiment of the Tweets were handled by the program. The key library used to do this is the NLTK. NLTK is the most popular Python library that works on handling human language data. It contains many sub libraries which lets the programmer do such things as classification, stemming, parsing and tokenization. From NLTK the program uses the tokenization function. The tokenizer will do such things as; removing noise, remove stop words, stemming and lemmatization. Noise is anything that isn't a standard number or letter, stop words are common words of little value, stemming is the process of putting the words into their base form and lemmatization is similar to stemming but all words created are actual words. The tokenization function is shown in Figure 16.

```
def tokenize_tweets(tweetdb, movie):
    df = pd.read_sql('SELECT * FROM movietweets', con=tweetdb)
    tweets = df.loc[df['movies'] == movie, 'tweets']
    raw=str(tweets).lower()
    sent_tokens = nltk.sent_tokenize(raw)
    word_tokens = nltk.word_tokenize(raw)
    return sent_tokens
```

Figure 16: Screenshot of tokenize_tweets function which puts Tweet text data into set of tokens

A further library used in this part of the program is the scikit-learn library. This is a simple and efficient Python library that is used for doing machine learning in Python. It can be used to do things like regression, clustering and classification. The functions used from this library were 'TfidfVectorizer' and 'cosine_similarity'. These functions will be used to compare the similarity between what the user has input to the data that has been gathered from Twitter in the 'response' function that is shown in Figure 17.

When given a user input the first thing that is done is for this section is that the user input along with the related movie data are sent to be tokenized by the 'LemTokens' and 'LemNormalize' functions. Firstly the 'LemNormalize function removes punctuation and tokenizes the words, then sends the tokenized words to 'LemTokens'. The 'LemTokens' will then lemmatize the given tokens and return it back to the original function.

With the text now filtered and put into tokenized words the scikit-learn library functions can now be used to look for similarity between the user input and the text. Firstly the tokens are put into the 'TfidfVectorizer' which puts the raw text into a matrix of TF-IDF features. Term frequency is the frequency of a word in a document and has the formula: TF = (Number of times word in document) / (Total number of words in document). Inverse document frequency is how rare a word is across documents and has the formula: IDF = 1 + log(N/n) where N is the number of documents and n is the number of documents term t appears in. Then TF-IDF is the weight if these two things so would have the formula: TF-IDF = TF * IDF.

With the TF-IDF matrix obtained the 'cosine_similarity' function can now be applied along with this to find the cosine similarity between any pair of vectors. The 'cosine_similarity'

function works by applying the following formula: Cosine Similarity (d1, d2) = (Dot product (d1, d2)) / (||d1|| * ||d2||). This gives the similarity between any two vectors provided where d1 and d2 are the non-zero vectors.

Through using the 'TfidfVectorizer' and 'cosine_similarity' functions with the user input and relevant movie data, a list of tokens that have a similarity are created. Hence a string can be output of relevant information to what the user has asked taken from the Tweet text data in the database. If there are no relevant tokens found by using the algorithm then an output of 'Sorry! I don't understand you' is output to tell the user that there is no relevant data for their query in the current database. To be able to get a user query and to output the relevant responses a ChatBot is created which was used as the interface to interact with the user.

```
def response(user_response, st):
    robo_response=''
    sent_tokens=st
    sent_tokens.append(user_response)
    TfidfVec = TfidfVectorizer(tokenizer=LemNormalize, stop_words=['english', 'ha', 'le', 'u', 'wa'] )
    tfidf = TfidfVec.fit_transform(sent_tokens)
    vals = cosine_similarity(tfidf[-1], tfidf)
    idx=vals.argsort()[0][-2]
    flat = vals.flatten()
    flat.sort()
    req_tfidf = flat[-2]
    if(req_tfidf==0):
        robo_response=robo_response+"I am sorry! I don't understand you"
        return robo_response
    else:
        robo_response = robo_response+sent_tokens[idx]
        return robo_response
```

Figure 17: Function used to respond to a user's query that isn't associated with sentiment analysis

6.6 ChatBot

The ChatBot is the interface of the program, its main roles for the program is to interact with the user and call the right functions that accurately deals with what the user would like to know. A ChatBot is generally defined as an AI computer program that interacts with a human user through text or speech. In this program it will only interact via using text. The overall goal for the ChatBot in this program was to be able to handle every query that it is given in some form and to provide the user with information about the movie that the user is asking about. The main ChatBot function is shown in Figure 18

```
flag=True
tweetdb = connect_to_database()
print("Chatbot: Hello, I will answer your queries about movies. If you want to exit, type bye")
movie = get_movie()
if(movie != 'bye'):
    movie = movie.lower()
    get_movie_twitter(movie, tweetdb)
    sent_tokens=tokenize_tweets(tweetdb, movie)
else:
    flag=False
    print("Chatbot: Bye! take care..")
    tweetdb.close()
while(flag==True):
    user_response = input()
    user_response=user_response.lower()
    if(user_response!='bye'):
        if(user_response=='thanks' or user_response=='thank you' ):
            flag=False
            print("Chatbot: You are welcome..")
        else:
            if(greeting(user_response)!=None):
                print("Chatbot: "+greeting(user_response))
            else:
                if(sentiment_check(user_response, movie, tweetdb)!=None):
                    print("Chatbot: " + sentiment_check(user_response, movie, tweetdb))
                    movie = anymore_questions(tweetdb, movie)
                    if(movie == 'exit program'):
                        flag=False
                else:
                    print("Chatbot: ",end="")
                    print(response(user_response, sent_tokens))
                    sent_tokens.remove(user_response)
                    movie = anymore_questions(tweetdb, movie)
                    if(movie == 'exit program'):
                        flag=False
    else:
        flag=False
        print("Chatbot: Bye! take care..")
        tweetdb.close()
```

Figure 18: The main ChatBot program that is responsible for running the program and interacting with the user

The first thing that is done by the ChatBot when running the program is to output a welcome message to the user as shown at the start of the code in Figure 18. This shows the user that the program is successfully running and introduces that the ChatBot is ready to receive an input. This message also provides the information to the user that the ChatBot can be exited at any time by typing the word ‘bye’. After the introduction message the ChatBot then calls the ‘getmovie’ function shown in Figure 19. The purpose of this function was to ask the user to input the movie they are seeking to find information on, which can then be used in further functions in the program. This function will then ask the user what information they are seeking to find out about the film they are inquiring about and will return the movie to the ChatBot that the user is inquiring about.

```
def get_movie():
    print("Chatbot: What is the name of the film you'd like to know about?")
    movie = input()
    if(movie != 'bye'):
        print("Chatbot: What about", movie,"would you like to know?")
    return movie
```

Figure 19: The get_movie function that takes in a movie from the user and returns it to the main ChatBot program

The next part of the ChatBot is to send the movie that the user has inquired about to the Twitter data mining function in the program to get the relevant data from Twitter and add that data to the database. The program will then gather the Tweet data in the database and tokenize all the current data available to be used for other functions.

With the initial greeting and information gathering done the ChatBot then goes into a while loop which is the main part of the ChatBot. This while loop will keep looping until the flag it is given goes from True to False. This is done by a few different actions that the user can perform to signify that the user wants to exit the program such as typing ‘bye’ or ‘thanks’. Doing this will change the flag to False and the loop will end.

Whilst the program is looping it goes through a few different if statements to work out what the user is trying to ask from the input. Firstly the ChatBot checks if the input matches ‘bye’ or ‘thanks’ to make sure that the user isn’t trying to exit the program. If this is not the case the program will then check if the user is putting in their own greeting message such as saying ‘hi’. To do this a greeting function is called that contains a list of normal greetings that are used by people in daily lives. If the input given matches one of these usual greeting messages then the ChatBot will respond with its own greeting message that is chosen at random from a list of greeting messages that the functions contains.

If the user input is found not to be a greeting input the next type of input that is checked for is whether the user is asking for some kind of sentiment recognition about the movie being queried about. This is done by calling the ‘sentiment_check’ function that is shown in Figure 15. If the user input matches one of the keywords in the ‘sentiment_check’ function then the relevant sentiment analysis will be done by the function to answer the user query. The ChatBot will then respond to the user what the function has found from the sentiment analysis that has been done.

If a sentiment analysis query is answered by the ChatBot, it will then ask the user if they would like to see further statistics about the film. This is a yes or no question. If the user answers ‘yes’ the data_visualization function in Figure 22 will be called, which is used to display various visualizations of the data. If the user answers no the ChatBot will ask if they would like to know any more information about the film they are inquiring about. This is a yes or no question. If the user answers yes then the ChatBot the will ask what the user would like to know and the main ChatBot program loop will restart with the new user input. However if the user answers no the ChatBot will then query if the user would like to learn about another film. Once again this is a yes or no question. If the user answers yes the ‘get_movie’ function is called to get the new movie and find the relevant information. This will then restart the main ChatBot program loop with the user being asked what they would like to know. If the user was to answer no then the ChatBot would assume the user has no further queries to ask it and will output a goodbye message and exit the main ChatBot program loop. This is done in a separate function named anymore_questions that is shown in Figure 20.

```
def anymore_questions(tweetdb, movie):
    while(flag==True):
        print("Chatbot: Would you be interested to see other statistics for this film? Please enter 'yes' or 'no'")
        user_response = input()
        if(user_response=='bye'):
            print("Chatbot: Thanks for visiting!")
            movie = 'exit program'
            return movie
        elif(user_response=='yes'):
            data_visualization(tweetdb, movie)
        elif(user_response=='no'):
            print("Chatbot: Would you like to know anything else about this film? Please enter 'yes' or 'no'")
            user_response = input()
            if(user_response=='bye'):
                print("Chatbot: Thanks for visiting!")
                movie = 'exit program'
                return movie
            elif(user_response=='yes'):
                print("Chatbot: What would you like to know?")
                return movie
            elif(user_response=='no'):
                print("Chatbot: Would you like to learn about another film? Please enter 'yes' or 'no'")
                user_response = input()
                if(user_response=='bye'):
                    print("Chatbot: Thanks for visiting!")
                    movie = 'exit program'
                    return movie
                elif(user_response == 'yes'):
                    movie = get_movie()
                    get_movie_twitter(movie, tweetdb)
                    sent_tokens=tokenize_tweets(tweetdb, movie)
                    return movie
                elif(user_response == 'no'):
                    print("Chatbot: Okay, enjoy your day...")
                    movie = 'exit program'
                    return movie
                else:
                    print("Chatbot: Please answer yes or no")
            else:
                print("Chatbot: Please answer yes or no")
        else:
            print("Chatbot: Please answer yes or no")
```

Figure 20: Function which is called to see if the user has any more questions to ask the ChatBot

If the user query does not fill the parameters for any of the previous if statement conditions then the input is sent to the query handler that was described before. The ChatBot will send the user input along with the tokenized data to the response function which will use the NLTK and scikit libraries to try and understand what the user is asking and find the relevant data to output. If the relevant data is found it is returned to the ChatBot which will output the response to the user. If the relevant data cannot be found or the function fails to understand what the user input means then an output of 'sorry I don't understand you' will be returned to the ChatBot and will be output to the user.

Once the query has been dealt with the ChatBot will do the same outputs that were done after the sentiment analysis function was called using the 'anymore_questions' function. The ChatBot will ask if the user would like to know any more about the current movie or would like to query about a different movie. Once again depending on how the user responds the ChatBot will either exit the loop and the program will end or it will restart the loop.

6.7 Data Visualization

Data visualization is used in this program to present the data that has been gathered in an interesting and easy to understand way. This allows for the data to be simplified into clear charts and graph that can be displayed to the user. Multiple data visualization methods are used in the program to present different parts of the data analysis that has been carried out to the user. An important note about the data visualization done in this program is that all charts are created using the Matplotlib Python library. Matplotlib is a library which provides an

interface similar to MATLAB and allows developers to embed various plots into their Python programs.

The first way that this is done in the program is by using the 'visualize_sentiment' function, shown in Figure 21, to display the sentiment analysis that has been done, in the form of a pie chart. The 'visualize_sentiment' function is called from 'sentiment_check' and is given the positive, neutral and negative sentiment percentages that have been worked out in that function. These values are each given a section and relevant labels in the pie chart. The pie chart is then made with each section having different colours and one section being exploded out to make it visually appealing.

```
: def visualize_sentiment(p, n, neu):

    labels = 'Positive', 'Neutral', 'Negative'
    sizes = [p, neu, n]
    colors = ['gold', 'yellowgreen', 'lightcoral']
    explode = (0.1, 0, 0)
    plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%', shadow=True, startangle=140)
    plt.axis('equal')
    plt.show()
```

Figure 21: Function to visualize the sentiment analysis in the form of a pie chart

The next part of data visualization that was implemented was methods to do with the statistics found in the data. For this section if the user says yes to see the related statistics about the movie they are inquiring about, the data_visualization function shown in Figure 22 is called which in turn will call the various data visualization functions that are used in the program. All of these functions require input parameters of the connected database and the movie that has been input by the user when they are called.

```
def data_visualization(tweetdb, movie):
    pop_tweet(tweetdb, movie)
    word_cloud(tweetdb, movie)
    popularity_chart(tweetdb, movie)
```

Figure 22: The data_visualization function that calls all the visualization methods used for the program

The first function called by data_visualization is the pop_tweet function that is shown in Figure 23. The purpose of this function is to output the most popular Tweet, based on the retweet count, in the database. This is done by firstly putting the entire database into a pandas DataFrame. Pandas is a software library in Python that is used to do data manipulation and analysis. It allows for developers to manipulate the data contained within database tables. Once the table is loaded into the DataFrame the data is then sorted by putting the retweet count into descending order. Hence the Tweet data with the highest retweet count will be at the top of the table. A command is then passed using pandas to take the first value in the Tweet column in the data which has the movie name matching the users input. This value can be output through the ChatBot and it will show the user the single Tweet text data which has been retweeted the most.

```
def pop_tweet(tweetdb, movie):
    df = pd.read_sql('SELECT * FROM movietweets', con=tweetdb)
    df.sort_values(by='retweet_count', ascending=False)
    toptweet = df.loc[df['movies'] == movie, 'tweets'].iloc[0]
    print("Chatbot: The most popular Tweet about this film is: ")
    print(toptweet)
```

Figure 23: Screenshot of function used to display the most popular Tweet about the users movie in current database

The next function that is called by data_visualization is the word_cloud function shown in Figure 24. The purpose of this function is to produce a word cloud from the text data gathered about the users given movie. A word cloud is produced to display to the user the keywords that are used in Tweets about the movie and hence gives the user an idea of details about the movie. The first thing that is done within this function is that the Tweets text data that has the movie value that the user input will be loaded into a cursor using an SQL statement. A cursor is a class in the mysql.connector library that is bound to the connected database and can be used to execute commands. The Tweet text data gathered is then put into an array and the text is cleaned to get rid of unwanted characters. This is done so that these unwanted characters don't end up being displayed in the word cloud. Next the stop words are set; these are words to not be included in the word cloud. In this instance the chosen stop words are; 'movie', 'film' and the movie the user has entered. This is done because these words are likely to occur in most Tweets and don't provide any real useful information to the user. Finally the word cloud is created by using the wordcloud library and output via the ChatBot. The wordcloud library has been made to specifically produce a wordcloud from any string of text data that it is given.

```
def word_cloud(tweetdb, movie):
    cursor = tweetdb.cursor(buffered=True)
    sql = "SELECT tweets FROM movietweets WHERE movies = %s"
    cursor.execute(sql, (movie,))
    tweets = cursor.fetchall()
    tweets=re.sub(r'bytearray','',str(tweets))
    tweets=re.sub(r"b'",'',str(tweets))
    stopwords = set(STOPWORDS)
    stopwords.update(["movie", movie, "film"])
    wordcloud = WordCloud(stopwords=stopwords, background_color="white").generate(str(tweets))
    print("Chatbot: Below is a word cloud showing the most popular words used in tweets about the movie: ")
    plt.figure()
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis("off")
    plt.show()
```

Figure 24: The function used to produce a word cloud from the Tweet text data about the movie that the user is inquiring about

The final function to be called by the data_visualization function is popularity_chart shown in Figure 25. This functions purpose is to display to the user the popularity of their chosen film in the database and how that popularity compares to other films within the database. The function does that by creating a bar chart and by outputting the amount of times the movie is mentioned in the database. To achieve this once again the table is first put into a pandas DataFrame. The DataFrame is then used to create a list of the unique movies that is currently contained within the database. A loop of these unique movies is then done where for each unique movie the total number of Tweets and Retweets about it in the database is worked out.

If the current movie being worked on is the movie that was input by the user then the number that is worked out is printed out to the user. The unique movies were then put in a bar chart against their related mention count that has been worked out from the database. The movies were put on the y axis so to not have the text overlapping to make it visually clear what bar relates to what movie. Finally the axes are given labels and the chart is given a title so the user knows what the chart is about and the chart is then output.

```
def popularity_chart(tweetdb, movie):
    mmcount = []
    df = pd.read_sql('SELECT * FROM movietweets', con=tweetdb)
    uniqmovies = df['movies'].unique()
    for m in uniqmovies:
        mmcount.append((df.groupby('movies')['retweet_count'].sum()[m])+(df.groupby('movies')['tweets'].count()[m]))
        if(m==movie):
            print("Chatbot: The amount of metions your selected movie has in my database is: ", mmcount[-1])
    fig, ax = plt.subplots()
    y_pos = np.arange(len(uniqmovies))
    ax.barh(y_pos, mmcount, align='center', alpha=0.5)
    ax.set_yticks(y_pos)
    ax.set_yticklabels(uniqmovies)
    ax.invert_yaxis()
    ax.set_xlabel('Metions in Database')
    ax.set_title('Movie Popularity Comparison in Database')
    plt.figure(figsize=(20,40))
    plt.show()
```

Figure 25: The popularity_chart function used to output the popularity of the users chosen movie in the form of a bar chart and a text output

7. Testing: Verification and Validation

Testing was done throughout the creation of this project, through doing the testing errors in the program were found and the program was improved. Thus testing is an important factor that is required in order to create a system that will be able to match what was laid out in the technical specification. For this project both static and dynamic testing was done to check the program met the original program objectives and that there were no faults or errors in the final software.

7.1 Static Testing

Static testing is a form of testing that is used to detect faults in software development without actually running the software. It is mainly used to discover faults early in development by applying it before running dynamic tests. Another reason that static testing is performed is that it means reduced time doing dynamic testing of the software due to there being a better development process and less defects in the software.

The first way that static testing was done in this project was by performing reviews of the initial documentation and objectives that were created in the project. This was to find any of the following; unrealistic objectives for the program, faulty designs, missing program requirements and errors in technical specification. For doing these reviews the first document to be looked at was the PID. The PID was the first document to be set up for this project and set out the initial goals, objectives and aims. Hence when reviewing the PID the key things looked into were whether the goals set were realistically achievable and whether any new or different objectives needed to be set. The first thing noticed when reviewing the initial plan was that the program was going to be limited to only Marvel and DC films. From researching the available data mining tools it was found that by setting this restriction it was limiting the possibilities of the software whilst also actually making the software needlessly complex. The next fault found in the original plan was how it was planned that both Python and KNIME were to be used to develop the software. Once again this seemed to be making the program needlessly complex as Python is able to achieve the proposed data mining objectives whilst still being able to create the ChatBot. Thus it was decided to only use Python in development to provide a singular platform and not having to worry about cross implementation. A final required change found when reviewing the technical specification in the PID was that it originally wanted machine learning to be applied to the data mining. Whilst this would have been an interesting feature improving the data mining it would have not really been possible to add with the time restrictions in place. A further objective added that would improve the program was to use sentiment analysis to improve how the software interprets the data gathered.

A further type of static testing that was done was to have other people review the plan and objectives to get feedback on the proposed program and advice on how to implement the wanted system. This was achieved by sharing the PID and initial documentation with the project supervisor and other peers who are working on similar projects. Through doing this type of testing throughout the whole project it meant that the quality of software development

and documentation would be overall improved. In terms of software development by doing peer reviews it allowed for suggestions of functions to add to the program such as types of charts and graphs for the data visualisation. It also allowed for different libraries to be used that others had tried and found success with. For the documentation it meant that there was a reduction in grammar errors and improvements in the overall explanations of the program.

The final kind of static testing that was done for the project was to perform static analysis on the code whilst it was being written. This is done to check for errors in the coding without having to run it first and also to check the optimal method has been used to write the code. This is done by going through every function and method that has been written in the code checking for errors and then looking for potential options to make that function run better or quicker. This testing will improve the overall quality of the final software created.

7.2 Unit Testing

Unit testing is the first form of dynamic testing done in the project. It is also known as component testing and is used to test the individual components of the software being developed. By doing the unit testing it allows for the developers to check that every individual part of the overall program works and doesn't contain errors or faults. It is testing the lowest-level of independently testable software.

Unit testing was carried out throughout the development process of this program and was done before any piece of code was committed so that an added unit was acceptable to be put into the main system. Unit testing was carried out by using the Python print function to output any methods that produce results and also by using exception handlers that would catch any errors that have been made in the program. By doing this everything coded and added to the program would have already been tested and would therefore be less likely to cause errors on integration.

7.3 Function Testing

Function testing was used to test all the different functions contained in the program. This was done by running the function and checking to see if that function produces the expected outcome from running it. Table 1 is a test case of the function testing that has been done.

Table 1: Test cases of the function testing

Test Case ID	Function	Description	Desired Outcome	Actual Outcome
1	Access Twitter API	Use given keys and tokens to connect to the API	Get authentication for accessing API	Authentication granted
2	Connect to the database	Establish a connection to the MySQL database	Able to access the MySQL database	Access granted
3	Get data from Twitter	Mine the movie data from Twitter using Tweepy	Relevant movie Tweets retrieved from Twitter	Tweet data collected

4	Pre-process text data collected	Clean the Tweet text data to make it readable and get rid of unwanted characters	String of the given Tweet without unwanted text	String of Tweet but a few unneeded characters
5	Add Tweet data to database	Add wanted Twitter data to the MySQL database	Database displays added data in relevant columns	Tweets data available in database
6	Tokenize Tweet text data	Tokenize the strings of text data	Collection of tokens of the Tweet text data	Tokens of Tweet text data
7	Perform sentiment analysis	Do sentiment analysis on every Tweets text data to find sentiment	Each Tweet text data has a given sentiment; positive, negative or neutral	Every Tweet text data has given sentiment however neutral has 0
8	Percentage of movie sentiment	The percentage of each sentiment Tweet text each movie has	A percentage of each type of sentiment for a given movie	Sentiment percentages of a movie, however neutral is 0%
9	Respond to a query	Generate a response to a user query through analysing a movies Tweet text data	Relevant response to query output to the user	Response generated
10	Visualize movie Tweet sentiment	Use sentiment analysis that has been done to create a pie chart	Pie chart showing the sentiment analysis of Tweets text data about a movie	Pie chart with sentiment analysis, neutral is 0%
11	Visualize movie popularity	Create a bar chart and text output showing the popularity of a movie from database data	Bar chart and text data showing the popularity of user selected film	Bar chart and text data with correct data
12	Generate wordcloud	Use the Tweet text data to create a wordcloud showing most popular words for a movie	A wordcloud for Tweet text data of user selected film	Wordcloud with variety of relevant words
13	Most popular Tweet	Use Tweet retweet count data to work out the most popular Tweet about users movie	The most popular Tweet text data output in string form	String of Tweet text data
14	ChatBot main program	Have ChatBot that is able to interact with user and tell the user the required input	ChatBot UI outputting information and responses	ChatBot with relevant responses and information
15	Get movie	Retrieve the movie that the user wants to get information on	Output question asking for movie and use input as the movie to use for system run	User interaction resulting in user inputting movie for program
16	Further questions	Series of yes or no questions asking the user	Output of yes or no questions	Output of yes or no questions

		if they want to perform further actions		

7.4 Integration Testing

For integration testing in this project the functions are split into 3 types and then tested together to see if all functions of a type integrate together well. The 3 types used to group the functions together is data mining, ChatBot and data analysis. These are the chosen types due to them being the 3 main features of the program, therefore each of the functions tested are assigned to one of the features. Table 2 shows which functions are associated with which feature and whether the feature passed or failed the integration testing that is described. The type of integration testing being done is white-box testing due to practicality and understanding of the program.

The first feature to be tested was data mining. This feature was tested for the integration between all the functions that is used to get the data from Twitter and store the data within the database. All of the functions interact with each other in some kind of way so it is important for the integration test to pass and there not be any errors stopping smooth integration between the functions. Firstly it was tested that the 'get data from twitter' function was able to use the 'access twitter API' function to be able to connect to the Twitter API. This is shown to work because the function is able to get the data from Twitter which wouldn't be able to happen if access hadn't been granted and shared. Next the integration between 'get data from Twitter' and 'pre-process text data collected' was tested. The integration was shown to pass due to the fact that the pre-process text function could use the data that had been gathered. Finally whether the data collected could be put into the database was tested. This required integrating 'connect to database', 'add Tweet to database', 'pre-process text data' and 'get data from Twitter'. The required result for this integration to be successful was to have the required Tweet data be put into the live MySQL database.

The next feature to be tested was the ChatBot feature. The ChatBot feature contained the main program function in the form of 'ChatBot main program'. This function was used to call all other functions for this feature and hence was the key to having successful integration within the program. Firstly it was tested that the 'ChatBot main program' function could call all the other functions for the features whilst also providing the required parameters to these other functions. The 'further questions' and 'tokenize Tweet text data' functions both required the database connection and input movie as parameters. Hence to test the integration between these functions and the ChatBot main program function, whether the required parameters were parsed needed to be tested along with checking they produce the expected output with the parameters. Next the 'get movie' function is tested to see if integrates with the 'ChatBot main program'. This function is only connected to the 'ChatBot main program' function and does not require any parameters to be input. Therefore the integration is considered successful if the function returns the users input movie to the 'ChatBot main program' function. The last function to be tested is 'respond to a query'. This function requires the created tokens from 'tokenize Tweet data' function and the user input. Hence

integration is successful if the function is able to use these parameters to return a relevant response to the query that has been input by the user.

Finally the data analysis feature integration is tested. Firstly the integration between the sentiment analysis functions was tested. This meant checking 'perform sentiment analysis' function passed on the Tweets sentiment to the 'percentage of movie sentiment' function to create the required percentages. In turn these percentages were then passed to the 'visualize movie Tweet sentiment' function then as long as a pie chart was successfully output with accurate sentiment analysis of the movie then integration was successful. Next the 'visualize movie popularity', 'generate wordcloud' and 'most popular tweet' functions were tested for successful integration. To test for integration between these functions another function called 'data visualization' was tested. It was decided to have these tested together for successful integration due to the fact that they all take in the same parameters of the user's movie and the database connection. To test for successful integration between these functions it was decided to use whether or not the required output was created accurately from the data being that was parsed to them.

Table 2: Test cases of the integration testing

Feature	Functions	Pass/ Fail
Data Mining	Access Twitter API, Connect to the database, Get data from Twitter, Pre-process text data collected, Add Tweet data to database	Pass
ChatBot	Tokenize Tweet text data, Respond to a query, ChatBot main program, Get movie, Further questions	Pass
Data Analysis	Perform sentiment analysis, Percentage of movie sentiment, Visualize movie Tweet sentiment, Visualize movie popularity, Generate wordcloud, Most popular Tweet	Pass

7.5 System and Acceptance Testing

This is the final stage of testing to be done on the system and it tests the system as a whole against whether it matches up to specified requirements, the technical specification and the original aims & objectives. System testing is done by running the whole program and going through various simulations with different inputs to see if the system does what is required from it.

To achieve the system testing the results the system gives is tested against the technical specification made in section 2.4. Table 3 shows the results of system testing against the technical specification. This table shows what result the system output for each technical specification wanted and whether this result is considered a pass, fail or a partially done result.

Table 3: Test cases of the technical specification acceptance testing

Technical Specification	System Result	Pass/ Fail / Partially

		done
The system will be able to search Twitter for any movie the user wants	User can enter movie into ChatBot and program will search for that movie	Pass
A database will be created that is hosted locally and can be accessed at any point in the program.	XAMPP used to create localhost server and MySQL database created in server. Program able to access database.	Pass
The database will contain no duplicate Tweets in it.	Every Tweet text data checked against database to avoid duplicates.	Pass
The database will allow for easy and fast searches for relevant data to a user's query.	The system does find relevant data to queries but sometimes partially complete and not understandable.	Fail
All Tweets stored in the database will have clean and easy to read text.	Tweets are pre-processed before being put into database but do contain some unwanted characters.	Partially done
All Tweets in the database will have usernames and hyperlinks removed to allow for user privacy.	Most usernames are removed but some unexpected ways of writing usernames not removed. All hyperlinks removed.	Pass
Only Tweets that are in written in English will be stored in the database.	All Tweets taken from Twitter are in English.	Pass
The system will be able to understand the sentiment that the Tweet is being written in.	System does understand Tweet sentiment but doesn't recognise neutral sentiment Tweets.	Partially done
The ChatBot will be able to respond to every user query that it is given with a relevant response.	ChatBot can respond to every user query with a response that is relevant.	Pass
The ChatBot will provide clear instructions for the user at all points in the program.	Instructions provided at every step of execution, done in various ways.	Pass
The ChatBot will be able to call on relevant functions in the program to deal with the user query.	Multiple functions used to deal with the various queries presented.	Pass
The ChatBot will be able to understand the sentiment the user query is being asked in.	ChatBot uses keyword recogniser to check for sentiment in user query. Only has list of certain keywords.	Partially done
The ChatBot will be able to present the data the user seeks in a presentable and easy to understand way.	ChatBot able to call functions that will analyse the data and present the data to the user in various ways.	Pass
Data visualization methods will be used to help with the clarity of the data that is presented.	Various data visualization methods used to present the data to the user.	Pass

From testing the system against the technical specification it can be seen that the system pretty much meets all the original criteria that was set out at the beginning of the project. Through doing this it shows the system has passed the acceptance criteria and overall does what was wanted and required to be considered a successful piece of software.

8. Discussion: Contribution and Reflection

Through looking at and analysing the results that were attained through doing the testing the overall successes and failures of the system can be analysed and discussed. The first noticeable details when looking at the testing results are where the program has failed to match up to some of the original technical specifications and not passed the wanted criteria for them.

8.1 Limitations

The first instance of this type of limitation is shown is how the system is not able to fully understand some queries and use the data to output a relevant response to the user. This is where in some cases in testing it was found that either the system would output the wrong movie data to the user or would only partially output some pieces of data. These errors only occur when using the user response function and tend to occur when multiple films are inquired about. Whilst these errors do occur the ChatBot does still produce some kind of output and still continues to run the program so it is not a serious error in the program.

The next limitation if the program is that it can't completely clean the Tweet text data to get rid of all unwanted characters and usernames. This is because there's no specific function to get rid of usernames in the Python programming language and it has to be done manually through the regular expression library to substitute out certain character patterns. However this does not account for if a username is spelt in a different way that could be assumed as a normal English word so will not seem to be required to be filtered out. This could cause some ethical issues and may be considered a vital error that would need to be addressed.

Another limitation found in testing was that in the sentiment analysis no neutral Tweets were found. This could be due either to the fact no Tweet is of neutral opinion which is highly unlikely or that there is an error within the Text Blob library that means it cannot identify neutral text properly. Overall whilst this error is present it does not affect the overall program that much as it still allows for a comparison between positive and negative sentiment which is what the user would be mainly interested in anyway.

A further limitation found in the overall system was that there was no function made to truly understand what the user's sentiment was when testing a query. Once again whilst being considered an error in the program it is not something that affects the overall quality of the system too much due to that sentiment is still recognised through looking at keywords used.

A final few limitations that can be seen from looking at the final results from testing is that not all data is collected in a fast & simple manner, data visualization may not look as good if database size increases and ChatBot is very basic with a reliance on yes & no questions. However these errors are minor and don't affect the program too much.

8.2 Testing Summary and Discussion

Various conclusions can be drawn from the testing that has been done for this system. It is important to analyse the testing results because it allows for the overall success to be decided and lets the developer draw conclusions and think about how the project and similar projects can be improved in the future.

Through analysing the testing results it can be seen that all the different aspects of the system pretty much fully worked. The data mining was able to collect the relevant data to a user's movie input from the Twitter API and store it on a locally hosted MySQL database. Testing showed that the data mining was done in relatively fast way on each occasion a user runs the program. Thus it means that the database will be constantly growing as users continue to use the database hence making any analysis done more accurate. This shows that the database is always expanding hence the system will be on a whole consistently improving. A further conclusion that can be made from the results of the data mining is that whilst Twitter is a great platform to gather information from, the current API makes it difficult for developers to extract big data for a big subject such as movies. This is due to that every search will only give back 100 Tweets maximum and a lot of these could potentially be retweets which will be filtered out. Whilst streaming is an option to use in other projects to collect data this wasn't really an option for this project due to the fact that there were limited resources and an expansive subject matter. A final conclusion to be made from the data mining performed in the system is that through testing it is shown that doing the data mining never takes too long to gather the data. This is important considering the data is being collected whilst the user is running the program therefore the user would want the program to run quick and smoothly.

Looking at the results from testing the ChatBot the first conclusion that can be drawn is that the ChatBot is able to perform the key functionality of being the interface to the user. This is one of the most important functions in the system as a whole due to the fact that if the user can't use and understand the program then it would be considered a failure. Therefore the fact that the ChatBot is able to provide clear instructions throughout the use of the program and will respond to every user query is vital to the software's overall success. However the ChatBot is required to use some yes & no questions at points in the program and has some seemingly unnecessary loops that can put the user off. Another important factor about the ChatBot that can be seen from testing is that it a lot of its functions relies on the user inputting queries that have been expected by the developer. For example to access the sentiment analysis function a certain keyword out of a set of keywords is needed to be entered by the user. If the user is looking to find a function similar to sentiment analysis but doesn't type in one of the expected words they will instead be taken to the query function which will search the text data for similarity to the users query. This was shown when a tester tried using the word 'popular' when looking for sentiment analysis and was instead taken to the query response function. An overall conclusion made from the results of testing the ChatBot is that whilst it does allow for the program to run smoothly and without errors, it is limited in what it can do and there is definitely room for improvement.

For the testing results of the data analytics it was a lot simpler due to most of the data analytics expected results being some kind of data visualization output. Firstly when looking at the data analytics results from testing is that there isn't much variety in the kind of analytics done. Due to only having the Tweet text data, retweet count, favourite count and associated movie there isn't many ways available to do different kinds of data visualization. So whilst three different types of visualization results were produced it didn't really allow for a proper look into the vast amounts of data visualization methods there are available in the Python programing libraries. From looking at the data visualization results it can be concluded that the overall system requires large amounts of data to be accurate. This can be seen because when looking at sentiment analysis for example it produces results that some of the highest rated movies by the general populous have mainly negative opinions about them output by the program. Hence it can be seen when using data analysis with only a small batch of results of less than 1000 it will not produce results that are true or accurate when looking at real world statistics. Further conclusions from the data analysis testing results are that it is important to use as many data visualisation methods as possible due to the ability to make the data interesting and provide key data to the user. Also it is important to identify outliers within the data so to not skew data visualization models which make the models not as useful to the user.

Analysing the test results for the overall system shows that firstly the system created meets the overall goal of being able to find and display data from Twitter about any movie a user inquiries about. Meeting this goal in some manor was considered to be the very basic goal at the start of the project and was to be built on to make a high quality system. The system testing results shows a system that has numerous different functions available to the user to get information on a movie that they are interested in. When comparing the results to what was wanted in the initial aims and objectives of the project shown in the introduction it can be seen that the system at least achieves all of them in some kind of way. From the final analysis of the system testing it can be decided that the project overall was a success and achieved in the most part what was wanted.

8.3 Self Reflection

Looking at the final system that has been made and the results produced from the testing I can say that I am happy with the final outcome of this project. Having started this project with relatively little knowledge about the Python programming language and general data science practices I am happy with what I was able to achieve. My main goal was to create a system that I could use to find Twitter information on any movie I was interested in and I believe the system that has been created achieves this.

When looking at the learning process I think that it was important at the start of the project to get as much advice and information about the best methods to use to create the system. I believe doing this allowed me to have a good foundation of knowledge about how this system was going to be created and what each phase of implementation should be. When it came to the actual implementation of the system I found that due to the vast amount of resources and libraries available, it wasn't too difficult to create a program that was able to produce the

wanted output of the project. Finally testing was done throughout the program to allow for error testing during implementation and was an interesting way to view the progress of the project.

In summary I believe I have learnt a lot through the creation of this project. I came in with an interest in the data science side of computer science and by doing this project it has taught me a lot about it and has made me want to pursue opportunities in the data science field in the future. I also have learnt a lot about the practice of creating a project for a large system. From early ideas through planning, implementation and testing I have learned a lot about all stages of creating a system such as this.

9. Social, Legal, Health & Safety and Ethical Issues

This project has a focus on taking peoples social media data to find out their opinions on feature films without their knowledge of doing so. Therefore this raises ethical, legal and social issues about the use of data. In the case of legal use of the data being gathered, due to having to sign up for a Twitter developers account to be able to use the Twitter API means that Twitter has given the developer legal access to the public data on Twitter. Hence any Tweet made by a user to their feed where they have not chosen to make it private, is fair to use for this project. For meeting ethical standards for the use of this data, firstly at the start of this project an application to do the report was sent to the universities ethics board. With the approval from the ethics board it meant that the type of data mining being done in this project was ethically fine. A further precaution taken when dealing with the Twitter data was to remove any usernames and hyperlinks. This would allow for more privacy of the Twitter users and would address most of the social and ethical issues that would arise from using the Tweet text data.

A further legal issue when creating this project would be the use of different types of algorithms that have been previously used in other similar systems. Due to the nature of the modern programming world a lot of code is shared between programs and are seen as standard procedure for producing parts of a system. However to avoid any legal issues to do with this any similar code used has been edited to uniquely fit into the system that is being made and the source of the code has been referenced.

10. Conclusion and Future Improvements

The objectives for this project was to create a system that would take data from Twitter, store the cleaned data in the database, interact with users through a ChatBot, answer all user queries about any films, answer the queries using the data and use data visualization to make the data clear & interesting.

Looking at the final system that was produced it can be seen that the system can answer all user queries about movies in some ways whilst gathering data from Twitter. Through using techniques like data mining, AI and data analytics a system has been made that meets the objectives. By using these various techniques a complex system has been created that runs smoothly and without producing any errors.

Using the Tweepy library meant that the program was able to access the Twitter API and do data mining of people opinions on films. This data can then be organised and stored in the MySQL database to be able to be used in later functions within the program. By having the user put in a selected movie and having that movie as the query to search for with the data mining method meant that the database was expanding and improving with every use of the program.

With the data stored in the database it meant as long as there was a connection established to the database, any function in the program could access and use the data that had been gathered. This allowed for data analytics to be done at any point in the program relatively quickly and easy. Various methods were used to access and manipulate the data with using techniques like the pandas library and SQL statements in Python.

Through using these data analytics techniques various visualization methods were available to be used to present the data to the user in interesting and simple ways. Various Python libraries were used to do this and create different kinds of visualizations like bar charts and word clouds. By doing this it provided the user with a variety of information about the movie that they were interested in.

Sentiment analysis was performed in different ways in the program to understand what the sentiment of the Tweet text data was and what the sentiment of the user input was. This meant that the program was able to understand what was meant by a Tweet and use the information to tell the user what the general sentiment that was used in Tweets about the movie.

By using NLP techniques the program was able to compare what the user was inputting, to the data that was available in the database. Operations such as TF-IDF and cosine similarity were used to see if relevant data was in the Tweet text data and could be output to the user to answer a query that has been presented.

The ChatBot was used to access all these functions whilst also being the interface of the program to interact with the user. Clear instructions were provided by the ChatBot to the users at all stages of the program to allow for easy operation and simple understanding.

Depending on the user input the ChatBot will call the various available functions and present the information to the user. Error handling was performed by the ChatBot to make sure that at no point would the program crash and the user would always be given information even if a wrong response was input.

If future work was done for the project, the system created could be improved in many different ways. Firstly a proper UI could be implemented to interact with the user. Currently the ChatBot is only presented in the console of the IDE but if a mobile app or website was created it would provide a more user friendly environment for the system to be hosted on.

A further way to improve the program would be to have more variety in the data gathered and thus be able to use more data visualization methods. If data such as Tweet location and the time that the Tweet was made was also gathered when doing the Twitter data mining it would lead to a better understanding of the trends of how Twitter users shape their opinions of movies. This kind of data could be visualized using methods such as map hotspots, line graphs and graphical mapping to give the user a clear understanding of the data.

Another way to improve the program would be to improve the sentiment analysis that is done. This can be done by including emojis in the sentiment analysis. Emojis are special characters that display small images to show emotions, such as a winky face. By implementing these into the sentiment analysis it would give a better understanding to the true emotion given to the Tweet and help identify emotions such as sarcasm.

A final way to improve the program in the future would be to address the limitations found in the testing. By improving query response it would mean that the user would receive useful information to every question that they ask. Also to improve the current sentiment analysis system in place, this would mean have the system truly understand when a Tweet is displaying neutral sentiment and also applying better sentiment analysis to the user input to better understand what they are trying to ask. Finally by increasing the amount of data contained within the database to get more accurate results would also improve the system. However this would also mean that changes would have to be made to the movie popularity bar chart data visualization to make sure that it still provides clear information to the user without having clustered labels.

11. References

Chen, X., Vorvoreanu, M., and Madhavan, K. (2014). Mining Social Media Data for Understanding Students' Learning Experiences. *IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES,* [online] Volume 7, p. 246 – 259. Available at: https://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=6901341&punumber=4620076 [Accessed 8 Mar. 2019].

Nations, D. (2018) *What Is a Hashtag on Twitter?.* [online] Lifewire. Available at: https://www.lifewire.com/what-is-a-hashtag-on-twitter-3486592. [Accessed 9 March 2019].

Nirmala, C., Roopa, G., and Kumar, K. (2015). Twitter Data Analysis for Unemployment Crisis. In: *International Conference on Applied and Theoretical Computing and Communication Technology.* [online] Davangere: IEEE, p. 420-423. Available at: https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7449733&punumber=7449733&filter=issueId%20EQ%20%227456843%22&pageNumber=3&pageNumber=4 [Accessed 8 March 2019].

Reguera, N., Subirats, L., and Armayones, M. (2017) Mining Facebook data of people with rare diseases. In: *IEEE 30th International Symposium on Computer-Based Medical Systems.* [online] Thessaloniki: IEEE, p. 588-593. Available at: https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8100282&punumber=8100282&filter=issueId%20EQ%20%228104134%22&pageNumber=5&pageNumber=6 [Accessed 8 March 2019].

Tanwani, N., Kumar, S., Jalbani, A., Soomro, S., Channa, M., and Nizamani, Z. (2017) Student Opinion Mining regarding Educational System using Facebook group. In: *First International Conference on Latest trends in Electrical Engineering and Computing Technologies.* [online] Karachi: IEEE, p. 1-5. Available at: https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8267171 [Accessed 8 March 2019].

Chen, J., Hsu, C., Yang, C., We, C., and Ciang, H. (2017) A Data Mining Method for Facebook Social Network: Take "New Row Mian (Beef Noodle)" in Taiwan for Example. In: *IEEE 8th International Conference on Awareness Science and Technology.* [online] Taichung: IEEE, p 165-169. Available at: https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?filter=issueId%20EQ%20%228256413%22&rowsPerPage=100&pageNumber=1&resultAction=REFINE&resultAction=ROWS_PER_PAGE [Accessed 8 March 2019].

Russell, J. (2017) *Facebook Reactions: What They Are and How They Impact the Feed.* [Blog] Hootsuite. Available at: https://blog.hootsuite.com/how-facebook-reactions-impact-the-feed/ [Accessed 9 March 2019]

Brownlee, J. (2016) *Crash Course in Convolutional Neural Networks for Machine Learning*. [Blog] Machine Learning Mastery. Available at: https://machinelearningmastery.com/crash-course-convolutional-neural-networks/ [Accessed 9 March 2019]

Karn, U. (2016) *An Intuitive Explanation of Convolutional Neural Networks*. [Blog] the data science blog. Available at: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/ [Accessed 9 March 2019]

Deshpande, A. (2016) *A Beginner's Guide To Understanding Convolutional Neural Networks*. [Blog] Adeshpande3. Available at: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/ [Accessed 15 March 2019]

Napitu, F., Bijaksana, A., Trisetyarso, A., and Heryadi, Y. (2017) Twitter Opinion Mining Predicts Broadband Internet's Customer Churn Rate. In: *IEEE International Conference on Cybernetics and Computational Intelligence*. [online] Phuket: IEEE, p 141-146. Available at: https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8304916 [Accessed 8 March 2019]

Heredia, B., Prusa, J., and Khoshgoftaar, T. (2017) Exploring the Effectiveness of Twitter at Polling the United States 2016 Presidential Election. In: *IEEE 3rd International Conference on Collaboration and Internet Computing*. [online] San Jose: IEEE, p 283-290. Available at: https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8170182&filter=issueId%20EQ%20%228181461%22&pageNumber=2 [Accessed 9 March 2019]

Ng, A. (2015) What data scientists should know about deep learning. In: *Extract Data Conference*. [online] San Francisco: Extract. Available at: https://www.slideshare.net/ExtractConf/andrew-ng-chief-scientist-at-baidu [Accessed 15 March 2019]

Rahman, A., Mamun, A., and Islam, A. (2017) Programming challenges of Chatbot: Current and Future Prospective. In: *IEEE Region 10 Humanitarian Technology Conference*. [online] Dhaka: IEEE, p 75-78. Available at: https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?filter=issueId%20EQ%20%228288891%22&rowsPerPage=100&pageNumber=1&resultAction=REFINE&resultAction=ROWS_PER_PAGE [Accessed 9 March 2019]

Oh, K., Lee, D., Ko, B., and Choi, H. (2017) A Chatbot for Psychiatric Counseling in Mental Healthcare Service Based on Emotional Dialogue Analysis and Sentence Generation. In: *18th IEEE International Conference on Mobile Data Management*. [online] Daejeon: IEEE, p. 371-375. Available at: https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7960738&punumber=7960738&filter=issueId%20EQ%20%227962417%22&pageNumber=2&pageNumber=3 [Accessed 9 March 2019]

Ravi, R. (2018) Intelligent Chatbot for Easy Web-Analytics Insights. In: *International Conference on Advances in Computing, Communications and Informatics*. [online]

Bangalore: IEEE, p 2193-2195. Available at: https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8536361 [Accessed 9 March 2019]

Ranoliya, B., Raghuwanshi, N., and Singh, S. (2017) Chatbot for University Related FAQs. In: *International Conference on Advances in Computing, Communications and Informatics*. [online] Udupi: IEEE, p. 1525-1530. Available at: https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8119306&punumber=8119306&filter=issueId%20EQ%20%228125802%22&pageNumber=2&rowsPerPage=100&pageNumber=3&rowsPerPage=100 [Accessed 9 March 2019]

R, Thelwell. (2015) 5 real applications of data mining and business intelligence. [Blog] Matillion. Available at: https://www.matillion.com/insights/5-real-life-applications-of-data-mining-and-business-intelligence/ [Accessed 26 April 2019]

Internetlivestats, (2011). *Twitter Usage Statistics*. [online] Available at: https://www.internetlivestats.com/twitter-statistics/ [Accessed 26 April 2019]

Rouse, M. (2019) *What is social media?* [Blog] What makes enterprise unified communications work. Available at: https://whatis.techtarget.com/definition/social-media [Accessed 26 April 2019]

Nagar, S. (2019) *How to Make a Chatbot With AI* [Blog] DZone. Available at: https://dzone.com/articles/how-to-make-a-chatbot-with-artificial-intelligence [Accessed 26 April 2019]

12. Appendices

Appendix 1: Project Initiation Document (PID)

Individual Project (CS3IP16)

Department of Computer Science
University of Reading

Project Initiation Document

PID Sign-Off

Student No.	**23003330**
Student Name	**Thomas Braund**
Email	**t.braund@student.reading.ac.uk**
Degree programme (BSc CS/BSc IT)	**BSc CS**
Supervisor Name	**Dr Varun Ojha**
Supervisor Signature	
Date	

SECTION 1 – General Information

Project Identification

1.1	**Project ID** (as in handbook)
	370
1.2	**Project Title**
	Big data analytics using machine learning and data mining techniques.
1.3	**Briefly describe the main purpose of the project in no more than 25 words**
	To analyse social media feeds to find out whether people prefer the DC or Marvel franchise.

Student Identification

1.4	**Student Name(s), Course, Email address(s)** e.g. Anne Other, BSc CS, a.other@student.reading.ac.uk
	Thomas Braund, BSc CS, t.braund@student.reading.ac.uk

Supervisor Identification

1.5	**Primary Supervisor Name, Email address** e.g. Prof Anne Other, a.other@reading.ac.uk
	vk.ojha@reading.ac.uk
1.6	**Secondary Supervisor Name, Email address** Only fill in this section if a secondary supervisor has been assigned to your project

Company Partner (only complete if there is a company involved)

1.7	**Company Name**
1.8	**Company Address**
1.9	**Name, email and phone number of Company Supervisor or Primary Contact**

SECTION 2 – Project Description

2.1 **Summarise the background research for the project in about 400 words. You must include references in this section but don't count them in the word count.**

Problem:
The purpose of this project is to use social media feeds such as the Twitter feed, to be able to gather data on whether the Marvel or DC franchise is more popular in regards to the general public. A social media feed such as Twitter or Facebook are websites on the internet which allow anyone to create an account and post things such as images and opinions to their own personal feeds to be viewed by anyone who is allowed access to the profile. The reasoning behind using social media feeds such as these ones to solve the problem is that they are places which gather huge amounts of data every day from a wide variety of people. This means that there is a big collection of unbiased opinions on a subject like the Marvel or DC franchise that can be easily accessed and analysed using a tool like data mining to find out peoples personal preferences to a subject matter such as this one.

Available tools to solve the problem:
To be able to find a definitive solution to whether the Marvel or DC franchise is preferred a variety of different tools are going to have to be used to be able to gather, structure and visualize the data from the social media feeds. Firstly the tool that is going to be used as a base to bring the project together from data gathering to visualizing the data is the KNIME Analytics Platform. The purpose of using this tool will be to create a data science workflow to eventually present and show the results of the gathered data. To gather and model the data, data mining and machine learning algorithms will be used. The techniques that'll be used when doing data mining and machine learning will mainly be made up of Classification, Association and Regression due to the fact that these types of techniques are most likely to give an accurate result for the problem. When creating algorithms for these tools Python will be used due to the fact it is widely used throughout the data analytics industry and it is relatively easy to use. Python also has the advantage that it allows the use of API's which can be used to get data from remote web servers. Social media sites that will be used in this project such as Facebook and Twitter will offer certain data through API's. Finally a tool that will be used for the project is a Natural Language Processor (NLP) which will help with the consistent improvement of the output results. NLP is a type of ChatBot API which continuously take the data that is given to it, learn from that data to create a output that will match what a user is looking for.

Social media data analytics real use in real world:
In general when it comes to taking data analytics for social media the purpose of gathering information is to find out what perspective target audiences for a product would want and also to create and show the best adverts for specific people. So for example when looking at the analytics used and gathered in Facebook a company such as Musical.ly was able to mine data to find out the best demographics for their product and which new users to target to expand their own product. Another example in Facebook would be Sale Stock who managed to create target ads for different kinds of shoppers so they could bring in customers and increase revenue. These examples show the possibilities of gathering information from social media to help solve solutions.

References:

1. https://www.datasciencecentral.com/profiles/blogs/the-7-most-important-data-mining-techniques
2. https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/
3. https://www.ibm.com/developerworks/library/ba-data-mining-techniques/index.html
4. https://www.knime.com/knime-software/knime-analytics-platform
5. https://analytics.facebook.com/

2.2 **Summarise the project objectives and outputs in about 400 words.**
These objectives and outputs should appear as tasks, milestones and deliverables in your project plan. In general, an objective is something you can do and an output is something you produce – one leads to the other.

The first objective for the project is to do research into what data analysis is and how it can be used to gather information from social media feeds about a pop culture topic. This will require an in depth look into all the programs, techniques and methods that are expected to be used whilst also looking into how to make the final program be as efficient, fast and accurate as possible. The output wanted through doing the research is a document which details all the relevant information about how data can be gathered from social media like twitter feeds to be able to produce the results about Marvel or DC.

A second objective for this project will be to create a general workflow for the project using the KNIME Analytics Platform. This objective will be achieved by using the KNIME software to create the workflow and set up all parts of the project to allow the data to be filtered into the final models. This will include creating a database server, space for the data mining and machine learning algorithms and a web page at the end where the results can be produced and visualized. The required output here is to have a well-structured workflow, ready to be filled with the relevant social media data.

A follow up objective to creating the workflow will be to create the database that will store all the data from the social media feeds about the Marvel or DC topic. For the database it will likely be using a MySQL database due to the fact that it works well with the python programming language that is going to be used for the data mining and machine learning and it will allow for a lot of data to be stored. From doing this objective the output should be a database ready to be filled with the social media data.

A further objective to will be to then set up the data mining and machine learning algorithms to find the required data from social media and move it into the database and workflow. To do this as previously mentioned the Python programming language will be used due to its good usability features, the fact API's can be used with it and that it's already being an industry favourite when doing big data analytics. From doing this it will allow the data to be taken from social media to be stored and structured in the workflow and database for further use in showing the results of the problem.

Another objective will be to create a page where the data can be seen clearly and the results of the project are shown and explained. This will be done by creating visualisation of the data that has been gathered in the form of charts, an analysis of the results that have been gathered and what they mean and a conclusion explaining what the final outcome is.

A final objective will be to create a report that explains the whole process of the project, how it has been done and all findings and conclusion from doing it. The output here will be a documented report of all stages of the project with screenshots, diagrams and code snippets to clearly show what the problem was and how it was solved.

2.3 **Initial project specification - list key features and functions of your finished project.**
Remember that a specification should not usually propose the solution. For example, your project may require open source datasets so add that to the specification but don't state how that data-link will be achieved – that comes later.

- A workflow of the data analysis using KNIME
- A database containing all relevant data about the problem
- Algorithms that will use data mining and machine learning techniques to gather and structure information about Marvel and DC from social media feeds
- Implement an NLP to help improve output of results to the user
- A set of charts and models that visualize the data to show results that have been gathered
- Analysis of data found and how it is relevant to finding a solution of Marvel or DC
- Optimization algorithms to help boost model performance

2.4 **Describe the social, legal and ethical issues that apply to your project. Does your project require ethical approval? (If your project requires a questionnaire/interview for conducting research and/or collecting data, you will need to apply for an ethical approval)**

	The first issue that arises in the process of doing this project is due to the fact that a large set of data required is from social media feeds, some of the data that needs to be gathered may be personal data to people so this may cause an ethical issue for the project due to the nature of taking this data without permission. This part of the project may also cause a social issue due to the nature of just taking a person's information who may not have realized that was out there for everyone for see. When it comes to these issues the fact that they have put this data out in the public domain means that they have consented to allow this data to be viewed and used in data analytics whether the user realizes it or not. Also due to the fact that this is going to be used for educational purposes only then it should mean that it should be okay to use the data without raising serious ethical issues. A legal issue that could occur from doing this project would be that due to the nature of creating algorithms for data mining and machine learning, a lot of the algorithms may be similar to others that have been used previously to do a similar data analytic project. This is because a lot of algorithms used to do these techniques will be standard for the practice so will be similar in a lot of different projects.
2.5	**Identify and lists the items you expect to need to purchase for your project. Specify the cost (include VAT and shipping if known) of each item as well as the supplier.** e.g. item 1 name, supplier, cost
	No purchases are required.
2.6	**State whether you need access to specific resources within the department or the University e.g. special devices and workshop**
	No further resources are needed

SECTION 3 – Project Plan

3.1	**Project Plan** Split your project work into sections/categories/phases and add tasks for each of these sections. It is likely that the high-level objectives you identified in section 2.2 become sections here. The outputs from section 2.2 should appear in the Outputs column here. Remember to include tasks for your project presentation, project demos, producing your poster, and writing up your report.

Task No.	Task description	Effort (weeks)	Outputs
1	**Background Research**	**3**	
1.1	Research into necessary programs	1	Documentation of research
1.2	Research into data mining and machine learning techniques	1	Documentation of research
1.3	Research into complete d real world examples of data analysis	1	Documentation of research
2	**Analysis and design**	**7**	
2.1	Analyse the research gathered	1	Explanation of what has been found and how it'll be used
2.2	Use research gathered to make rough structure of overall program	1	Flow diagram of program structure with explanation
2.3	Design classes to be used in the programs	2	Class, UML and Component diagrams
2.4	Design pseudo code for the possible classes to created	2	Document of the pseudo code created
2.5	Design of final output page	1	Brief explanation of final output with layout diagram
3	**Develop prototype**	**8**	
3.1	Create initial KNIME layout	1	A workflow in the KNIME software
3.2	Code data mining algorithm to get data from social media	3	Data mining code that will be able to take specific data from a big data set
3.3	Code algorithm to create models using data mining and machine learning techniques	2	Models of the data gathered that presents the required analytics
3.4	Structure models and layout of output to make data analysis easy	1	A well-presented page containing all final outputs from the project
3.5	Improve algorithms to allow faster, efficient data analysis	1	Code snippets that makes the overall mining process faster
4	**Testing, evaluation/validation**	**4**	
4.1	Unit testing	1	Documentation of the testing outcome
4.2	Integration testing	1	Documentation of the testing outcome
4.3	System testing	1	Documentation of the testing outcome
4.4	Acceptance testing	1	Documentation of the testing outcome
5	**Assessments**	**3**	

5.1	Write-up project report	2	Project Report
5.2	Produce poster	0.5	Poster
5.3	Prepare for demonstration	0.5	Presentation
TOTAL	**Sum of total effort in weeks**	**25**	

SECTION 4 - Time Plan for the proposed Project work

For each task identified in 3.1, please *shade* the weeks when you'll be working on that task. You should also mark target milestones, outputs and key decision points. To shade a cell in MS Word, move the mouse to the top left of cell until the curser becomes an arrow pointing up, left click to select the cell and then right click and select 'borders and shading'. Under the shading tab pick an appropriate grey colour and click ok.

	START DATE: ../../…. **<enter the project start date here>**												
	Project Weeks												
Project stage	0-3	3-6	6-9	9-12	12-15	15-18	18-21	21-24	24-27	27-30	30-33	33-36	36-39
1 Background Research													
2 Analysis/Design													
3 Develop prototype.													
4 Testing, evaluation/validation													
5 Assessments													

RISK ASSESSMENT FORM

Assessment Reference No.		**Area or activity assessed:**	**Creation of software used for data analytics.**
Assessment date	**4/10/2018**		
Persons who may be affected by the activity (i.e. are at risk)	**Thomas Braund**		

SECTION 1: Identify Hazards - *Consider the activity or work area and identify if any of the hazards listed below are significant (tick the boxes that apply).*

1.	Fall of person (from work at height)		**6.**	Lighting levels		**11.**	Use of portable tools / equipment	x	**16.**	Vehicles / driving at work		**21.**	Hazardous fumes, chemicals, dust		**26.**	Occupational stress	x
2.	Fall of objects		**7.**	Heating & ventilation		**12.**	Fixed machinery or lifting equipment		**17.**	Outdoor work / extreme weather		**22.**	Hazardous biological agent		**27.**	Violence to staff / verbal assault	
3.	Slips, Trips & Housekeeping		**8.**	Layout , storage, space, obstructions		**13.**	Pressure vessels		**18.**	Fieldtrips / field work		**23.**	Confined space / asphyxiation risk		**28.**	Work with animals	
4.	Manual handling operations		**9.**	Welfare facilities		**14.**	Noise or Vibration		**19.**	Radiation sources		**24.**	Condition of Buildings & glazing		**29.**	Lone working / work out of hours	x
5.	Display screen equipment	x	**10.**	Electrical Equipment	x	**15.**	Fire hazards & flammable material		**20.**	Work with lasers		**25.**	Food preparation		**30.**	Other(s) - specify	

SECTION 2: Risk Controls - *For each hazard identified in Section 1, complete Section 2.*

Hazard No.	Hazard Description	Existing controls to reduce risk	**Risk Level** (tick one)			Further action needed to reduce risks *(provide timescales and initials of person responsible)*
			High	Med	Low	
5	Display screen equipment	Taking breaks away from looking at the screen so not looking at it for extenuated amounts of time		x		
10	Electrical equipment	Make sure all equipment is safe and not broken. If any faulty equipment is found replace it as soon as possible		x		
11	Use of portable tools/ equipment	When moving any equipment make sure it is a secure place and be extra careful of trip hazards			x	
Name of Assessor(s)		Thomas Braund	**SIGNED**			
Review date		4/10/2018				

Health and Safety Risk Assessments – continuation sheet

Assessment Reference No	
Continuation sheet number:	

SECTION 2 continued: Risk Controls

Hazard No.	Hazard Description	Existing controls to reduce risk	**Risk Level** (tick one)			Further action needed to reduce risks *(provide timescales and initials of person responsible for action)*
			High	Med	Low	
26	Occupational stress	Make sure to follow an initial plan and keep to the set out deadlines. Avoid having to do long periods of work so not to have to be worrying about finishing quickly.		x		
29	Lone working/ work out of hours	Take breaks from work to meet up and socialize with other people.			x	
Name of Assessor(s)		Thomas Braund	**SIGNED**			

Review date	4/10/2018	

Appendix 2: Logbook

Date	Work Done	Tasks Completed
24/09/2018	- Began researching social media mining techniques and platforms - Started planning the Project Initiation Document (PID)	
28/09/2018	- Researched ChatBots and how they could be implemented into a data mining program. - Completed background research section of the PID	
3/10/2018	- Set initial goals for the project. - Wrote aims and objectives section of PID - Wrote initial project specification section in PID	
5/10/2018	- Decided on project plan - Completed the draft of the PID - Initial meeting with project supervisor - Discussed PID draft with supervisor	
10/10/2018	- Made suggested changes to PID - Completed final PID	
12/10/2018	- Proofread the PID - Met with project supervisor	- PID submitted
15/10/2018	- Initial planning for the development of the code	
19/10/2018	- Met with project supervisor - Discussed what methodology to use for developing program	
23/10/2018	- Planned the solution approach for the project	
2/11/2018	- Attended presentation on research methods	
9/11/2018	- Wrote solution approach section for report	
10/11/2018	- Used solution approach to plan the system structure and implementation	
17/11/2018	- Started data mining section of system implementation	
30/11/2018	- Gave initial plan for introduction and literature review to project supervisor	

7/12/2018	- Created database to store data taken from Twitter	
18/12/2018	- Connected database to data mining program to be able to store the data	
22/12/2018	- Finalised data mining section of the program	
12/01/2018	- Started working on the ChatBot to be the user interface	
13/01/2018	- Continued work on the ChatBot	
18/01/2019	- Filled out feedback form	- Feedback form submitted
24/01/2019	- Updated literature review sent to project supervisor	
1/02/2019	- Finished off literature review - Continued working on ChatBot section of systen	
3/02/2019	- Wrote initial problem articulation section in report	
5/02/2019	- Discussed report with project supervisor	
13/02/2019	- Finalised problem articulation section for report	
17/02/2019	- Made initial project presentation - Practiced presentation skills - Finished ChatBot for system	
18/02/2019	- Demonstration to project supervisor - Attended talk on technical writing skills	
3/03/2019	- Started working on data visualization implementation into the system	
9/03/2019	- Continued working on data visualization	
15/03/2019	- Meeting with project supervisor - Showed initial plan for project poster	
18/03/2019	- Looked through feedback on project poster	
26/03/2019	- Edited project poster - Finished data visualization section for system	
29/03/2019	- Finished and checked project poster	- Project poster submitted
16/04/2019	- Sent second draft of project presentation to supervisor	
18/04/2019	- Wrote report design section - Integrated the different feature of the system - Did integration testing	

19/04/2019	- Began system testing	
20/04/2019	- Finished system testing and making final changes to the final system	
21/04/2019	- Wrote report implementation section	
22/04/2019	- Finalised project presentation - Practiced presentation - Practice system demonstration	
23/04/2019	- Finished testing and discussion sections in report	- Presented project system
24/04/2019	- Wrote conclusion and introduction for report	
25/04/2019	- Started proofreading report - Added in figures and tables where required	
26/04/2019	- Finished proofreading report - Added abstract and contents	
27/04/2019	- Finalised project report	- Project report submitted