CS310: COMPUTER SCIENCE PROJECT

ANALYSING HISTORIC DATA AND TWITTER FEEDS TO DETERMINE THE OUTCOME OF FOOTBALL MATCHES

November 26, 2017

Student ID: u1515140
Rushil Gala-Shah
University of Warwick

Contents

INTRODUCTION

The aim of this project is to predict the outcome of football matches in the English Premier League[1]. Predicting results is becoming more and more common with varying levels of success and as such there are multiple theories that try to use various variables to determine a certain outcome[2]. The focus of this project is based on the English Football League due to the competitiveness as well as the surprising outcomes that may occur every week. Chelsea player Conte has even commented that "Football is full of surprises"[4] showing that no prediction model will ever be fool proof. Another example of this is when Leicester won the English Premier League during the 2015-16 season against 5000-to-1 odds[5]. Having established the difficulty in using the English Premier League, the focus of the project is going to determine whether certain factors influence the outcome of the match, where outcome is the winner of the match rather than the score.

There are already many resources and models available which use statistics to determine the outcome as well as predictions on how many goals will be scored. However, these use their own set of variables and tend to range from simple to complex. Due to the wide variety of choices, the focus of the project has been narrowed to analysing various games and in particular of one team to determine if certain statistics happen most of the time. Although each team plays differently, the simplification will help to generalise whether certain variables such as possession or passing are valuable when taking into account of a team's success and is a good starting point.

BACKGROUND RESEARCH

Understanding the various models already out there has been the core part of the research so far. In most cases the research has also revolved around betting, yet it has still been useful to find out the variables other people consider important to winning a game. In particular the home advantage model has been a valuable resource. The model is based on a Poisson Distribution and takes into account how many goals the home and away teams score depending on whether they are playing home or away[6]. The model itself is only based on historic data and does not take into account any other factors. This is further illustrated in another model which states that the model does not work since there would be an even number of goals scored in each half[7]. The article suggests that more success may be achieved with bivariate Poisson distribution or Weibull distribution. However the article does highlight

some key features that would be required such as home advantage, the varying strengths and the quality of the opposition.

The other side of the research is selecting the statistical data. This data needs to be accurate and contain as much information as possible so that the factors that are taken into account can correctly calculate the outcome. Originally the API used was supposed to be Optasports[8]. However, due to the lack of availability for students and too high of a cost to make it viable for this project, other options involving free APIs had to be used. The first is football-data.org[10] which contains data for upcoming fixtures as well as past results. The past results are validated through various social media platforms such as Twitter. The bots collect mass amounts of data from Twitter streams to and are validated if the majority of Tweets for an event are parsed and meet a certain criteria. The other APIs used collect more information about particular fixtures giving in-depth analysis on particular players and whether they are likely to play well[11]. All this data will be considered when providing the outcome determined by statistical analysis.

DESIGN

The design of the solution is in two parts. The first is based on historical data and the second is based on real-time data. The user interface should contain the upcoming games for that week in the English Premier League. This is then linked to a python script that will work out a model to give a percentage of how likely each is likely to win. Any further information is found for each particular fixture. The scripts use will use the API data by calling various APIs and parsing the JSON format. If there are many requests then it may be more suitable for cron tasks to collect data once a week and add it to a database so that repetitive data may be used repeatedly without additional overhead. The real time data will need to scrape Twitter and will need to happen when live games are. Once again this should be accessed for each fixture when giving in-depth detail.

System Overview

The overall design will be similar to fig. 1 where the user will see the upcoming fixtures initially. The python-based web server will interpret the query and pass along the parameters to the a script that will either access the API directly or look it up in the database depending on further evaluation. The query may be just the initial page or it may be about a certain

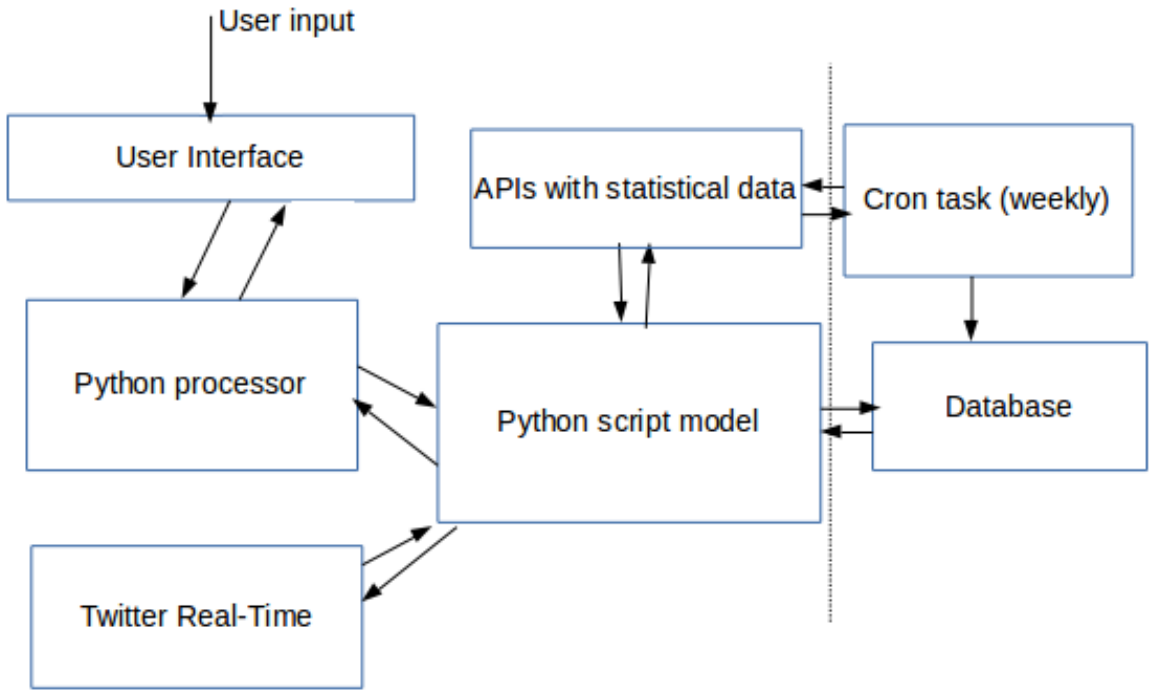


Figure 1: The system overview

match which will require additional data. The real-time system will involve parsing Twitter streams after the script determines whether the game is live or not. If the game is live then a model will be built on various statistics available on Twitter and verification will take place by quantifying the number of similar results. Furthermore this can be compounded on the normal model to determine if there are any mistakes. For example, if a top tier team is playing a team in the relegation and the team in the relegation is currently winning, however the Twitter stream is saying that the top tier team will make a come back and win the game, then a prediction can be made in favour of the top tier team based on the Twitter stream and backed up by statistical data.

Figure 2 shows the initial user interface that will be shown to users. The figure shows the upcoming games and who will win based on statistical data. This has already been pre-determined by the model and will provide percentages. A link for each match will prove more analysis, including reasoning behind the choosing that outcome.

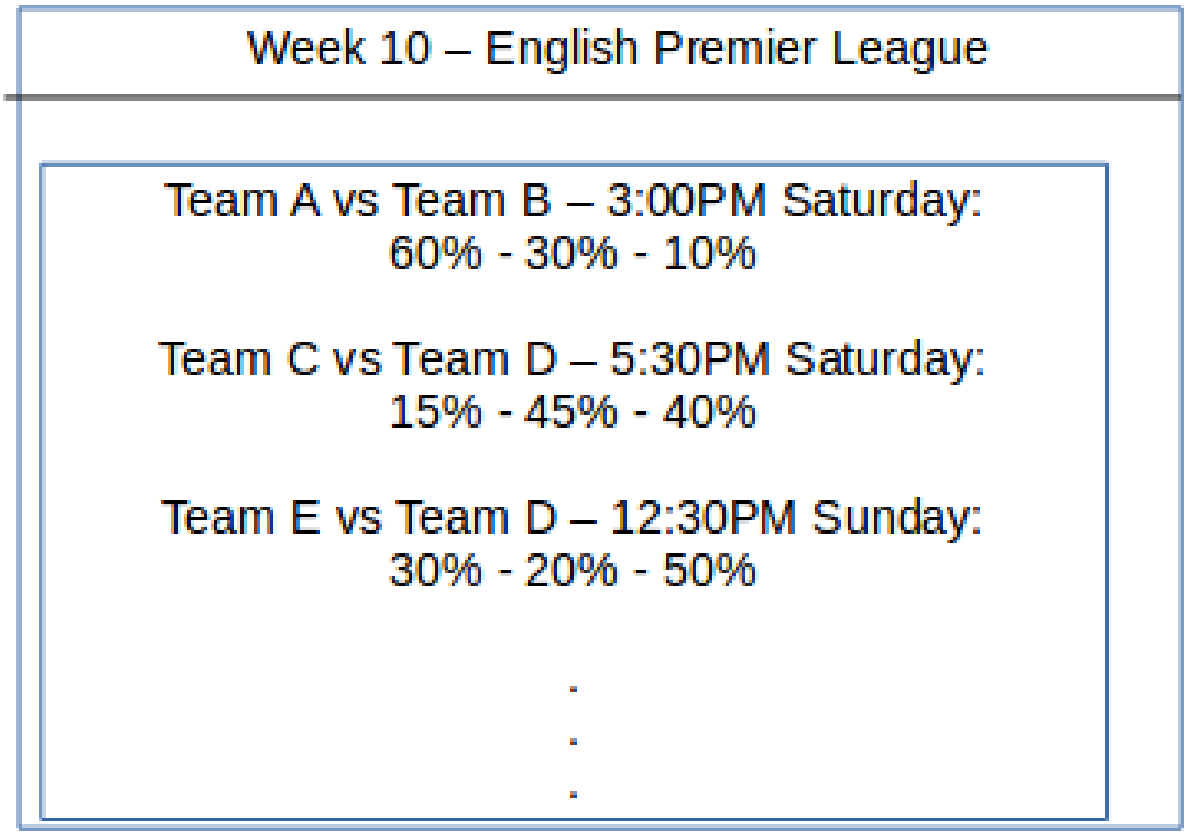


Figure 2: The initial user interface

Model

The model has already been mentioned several times and in this project, there are certain key features that will be prioritised. In particular, the following relationships will be taken into account:

1. Whether the team playing is home or away (home advantage)
2. The number of goals a team scores at home/away
3. The home/away form for the last 5 games
4. The head-to-head between the two games for the last 5 games
5. An average of the shots on target for the last 5 games
6. The number of goals conceded on average for the last 5 games

7. Whether more possession has resulted in a win

The choosing of these fixtures is due to the analysis of some of the games already played this season and in particular by Arsenal. An example of this is when Arsenal beat Tottenham Hotspur 2-0 despite being lower in the league and not in form at all. Yet, the fact that Arsenal were playing at home and were undefeated at home prevailed leading credence to the home advantage theory. Despite the fact that Tottenham Hotspur had more possession, Arsenal had more shots on goal showing contradiction therefore those variables were factored into the model.

PROJECT STATE

The progress made so far has mainly been research based and as such implementation has been slow. Furthermore, several factors including the time to learn Python and setting up the environment were underestimated initially and therefore delayed progress further.

Implementation

The implementation of the model has yet to be finalised, however the web interface and underlying technology has been set up. Flask is a microframework[3] which is going to be used in this project. The reason it is chosen over Django is due to the small nature of the project and therefore there is no reason in having a full stack of which Django implements. Having set up Flask, the user interface has been boosted with the help of bootstrap and all that is needed is to inject the python code. Furthermore, the use of a database is something that was not expected initially but due to the limitation on the request for APIs being limited, it is essential due to the number of calls made in one page. The choice in the type of database is going to be MongoDB due to the simplicity of modifying tables and querying them in Python with a wrapper. The APIs used will be football-data.org, crowdscores.com and football-api.com. The combination of all three APIs will provide the information necessary to calculate the outcome.

The Twitter API will be used for evaluating real-time scores, which will be parsed by the python script. This will use statistics such as possession and shots on target to make a more informed decision on the outcome.

Testing

Initially the testing has been made by determining the outcome through analysing the past behaviour and results of Arsenal. This resulted in testing out different models to determine which best fit Arsenal and then generalising it further to fit the premier league. Once a model has been decided, a simulation will be run using the previous season's results and data against the results this season to determine whether it is accurate. The testing can be done multiple times and can continue to extend every week matches are played.

The testing is done in two parts. The first part is checking to see whether all API calls work and whether connections can be made to the database so that information can be received. This is to ensure that the system is working. The second sets of tests deals with the accuracy of the model. This ensures that the logic of the system is working and that most predictions made occur. The baseline for the tests to pass is that they must ensure 70 percent of outcomes are correct.

PROJECT MANAGEMENT

Progress against original schedule

The project is behind schedule to underestimating some parts of the process. This is due to the length of time it took to research various models. The time taken to develop a model and to perform analysis on various premier league games took more time than expected. Furthermore, since different leagues play different styles of football, it took longer to determine how the premier league differs from various leagues around the world. For example, the premier league is known world wide for its pace which means that passes completed will be high and possession will be even[12] whereas the Italian league is known for its defensiveness and efficiency in attacks[12] so there would not be that many shots on target. This means that the implementation, testing and user interface is behind schedule, with only parts of it completed such as the framework and initial testing.

Unexpected Developments

There were several unexpected developments that impacted on the progress of the project. The original plan was to use Opta Sports data due to its comprehensiveness. However, the package was not within budget and not viable for a project of this nature. Fortunately, there

were several APIs that were more viable and they worked well with Python. Another issue was underestimating how much research would need and the time needed to learn and set up the development environment. The former took much longer than expected due to the extensiveness of the league and the multitude of factors, whereas the latter required time to set up the framework due to my lack of familiarity with Python. I ended up settling on Flask[3] as the framework and had to take extra time to learn how to set up the folder structure properly as well as learn how the data from the API can be displayed as a web page.

Updated Schedule

Term 2 will have more emphasis on the implementation side of the project now that the initial set up has been done. Figure 3 shows how the project will be split into manageable sections. The next few weeks will be spent developing the model further so that it can deliver a basic outcome

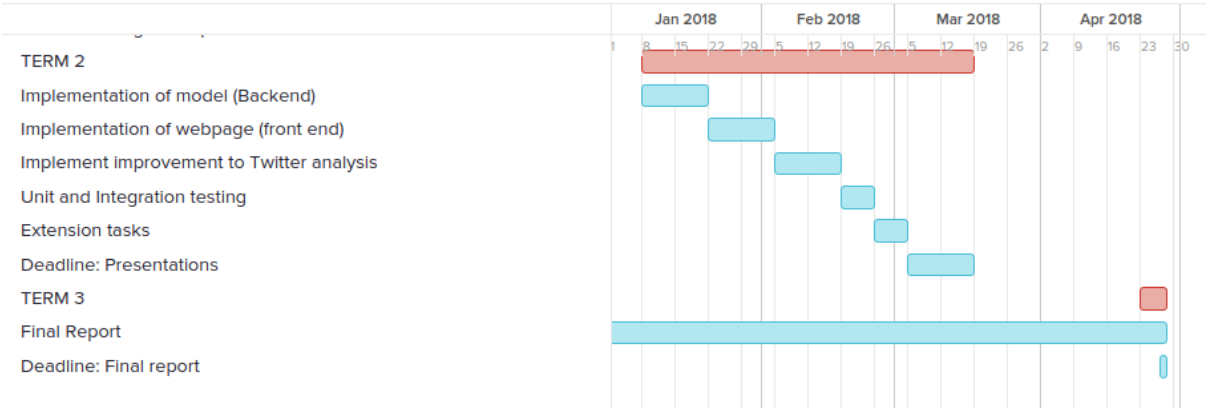


Figure 3: Term 2 Gnantt Chart

Legal, Social, Ethical and Professional Issues

As the people who are testing to see if the user interface is user-friendly and can be easily navigated are colleagues, there are no need to have ethical consent. Furthermore the software used in building the application is open-source. The APIs are available for free to use as long as the requests are limited to a certain number per day. This is easily remedied by caching the data and following any terms set out by the developers of these APIs. All the data used is available to the public and the system does not collect any personal data.

Bibliography

[1] Premier League, *Premier League Football News, Fixtures, Scores & Results*, https://www.premierleague.com/home

[2] Engineering and Technology, *The science of predicting football results*, https://eandt.theiet.org/content/articles/2010/06/the-science-of-predicting-football-results

[3] Flask, *A Python Microframework*, http://flask.pocoo.org/

[4] Goal, *'Football is full of surprises'-Conte downplays Chelsea's 10 point lead*, http://www.goal.com/en-ie/news/football-is-full-of-surprises-conte-downplays-chelseas-10/

[5] Telegraph, *Leicester City Win Premier League and Cost Bookies Biggest Ever*, http://www.telegraph.co.uk/news/2016/05/02/leicester-city-win-premier-league-and-cost-bookies-biggest-ever/

[6] Sports Betting Online, *Football Prediction Model: Poisson Model*, https://www.sbo.net/strategy/football-prediction-model-poisson-distribution/

[7] Python Football, *Predicting Football Results With Statistical Modelling*, https://dashee87.github.io/football/python/predicting-football-results-with-statistical-modelling/

[8] Opta, *Opta Classic Data feed*, http://www.optasports.com/services/media/data-feeds/classic-data-feeds.aspx

[9] 90mins.com, *6 Very Different Styles of Football*, http://www.90min.com/posts/2990047-6-very-different-styles-of-football-from-across-the-world

[10] football-data.org, *football-data.org API documentation*, http://football-data.org/documentation

[11] football-api.com, *Football-API v2.0 Documentation*, https://football-api.com/documentation2

[12] Quora, *What is the difference between different styles of football*, https://www.quora.com/What-is-the-difference-between-Spanish-British-German-Italian-and-French-styl

APPENDIX

A - Specification

INTRODUCTION

Football is a sport that is watched, analysed and talked all over the world by billions of people [3]. There is a huge industry built around the sport and an entire culture dedicated to it. However, one thing remains the same... predicting the final outcome of the match can never be determined with absolute confidence.

There are a multitude of factors that play into determining the final outcome such as a team's recent form, line up, whether they are home or away to name a few. Many people take these into account without realising when they attempt to predict the outcome of a football match.

The aim of this project is to combine statistical confirmed data and process such data whilst taking multiple opinions through Twitter feeds, analysing them and comparing it against the data that is already there to produce a more accurate prediction. As many opinions will be analysed, the statistical analysis will avoid bias and ensure that no opinion holds any weight.

OBJECTIVES

This project will attempt to determine the outcome of various football matches using different sources. It will make good use of the data collected from Twitter as well as data collected from the statistical history of the various football teams. The goal is to have as many accurate predictions as possible. To achieve this goal, there are several functional and non-functional objectives.

Functional

1. The system should be able to obtain historical data using a service such as Opta [5] or similar about a football team as well as through Twitter in real time.

2. The system must be able to identify keywords through multiple opinions.

3. The system must alert the user if the original prediction is different from the new prediction.
4. The system must be able to analyse multiple teams and leagues.
5. The system must learn to adapt to unexpected developments in a football match.
6. The system should ensure that erroneous data designed to cause confusion to the system is discarded.

Non-functional

1. The system must be able to eventually respond to every request.
2. The system should ensure that the data feed is accurate and the display of such data is accurate at all times.
3. The system should be compatible with mobile, tablet, laptop & desktop devices.

METHODOLOGY

I will take an agile approach to this project. The reason behind this is because it allows more flexibility and allows priority to the features that are required rather than bonus ones. However, a waterfall plan-driven approach may be used when documenting as that would allow the project to be documented thoroughly and as such easier to write throughout the project lifetime.

Using the 3rd party service Trello [1] will allow an easy way to organise the tasks in such a way that the timeline will be kept intact and that no objective is left forgotten. It will also serve another purpose in the sense of keeping track of the completed features at each stage of the project and measuring progress throughout the process.

The project itself will be kept in a git repository so that version control is easy to implement and as such allows features to be developed independently whilst keeping a working clean version of the code.

Development

Development will be split into two sections. The first will be the the data processing side which will be the core logic behind this project. The second will be the display which the end user sees using data that has been processed. Data processing will be written primarily in python due to being reasonably fast when processing data. This is especially important as data will need to be processed in real-time. The front-end will be using web technologies such as HTML, CSS, JavaScript due to the ease in connecting and displaying data from a database.

As development of the project matures, it is essential that there is a solid core to work and rely upon. This means that all data collected and processed even though that data may be of no use straight away, allowing future iterations or features to call upon this data if needed. With each iteration, I plan to address each objective in a sequential manner whilst using previous test data to improve previous implementations of past objectives.

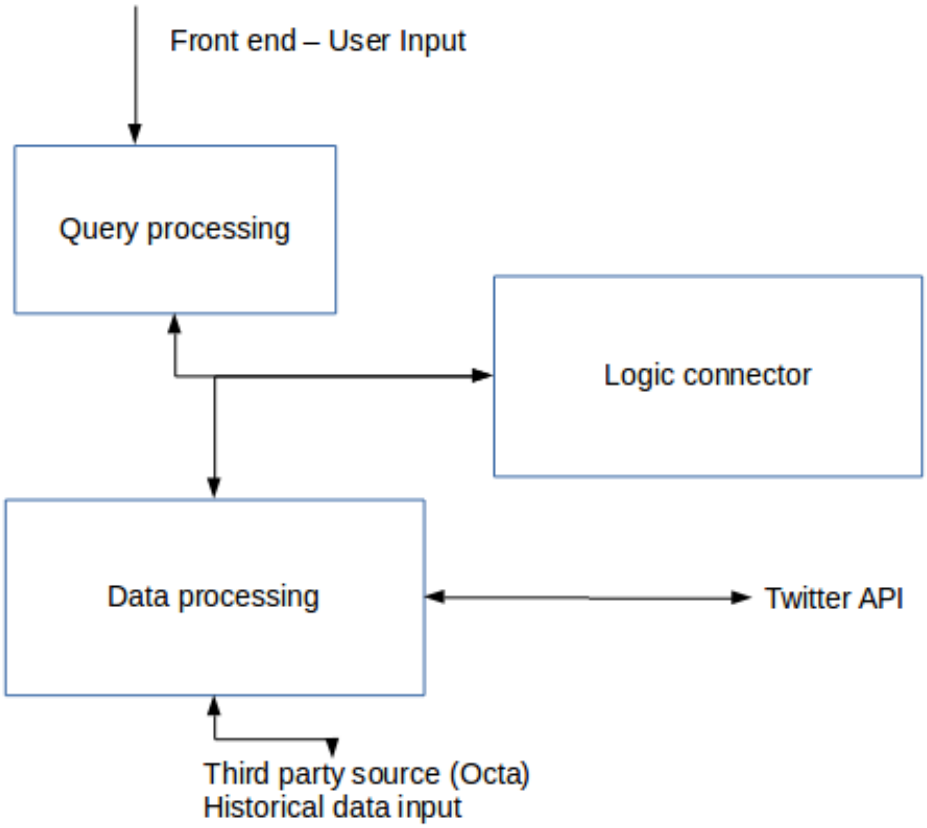
The front end will allow a user to enter a query such as the winner between two football teams. This query is processed by the logic which connects to the data processing side to calculate the outcome from the various sources. The result is passed back to the logic core and that is passed to the front end to display the result. A simple diagram containing the initial infrastructure is displayed below.

Testing

The actual events of a match will be compared against the predicted events. There will be three factors to determine the scale of correctness.

1. The correct outcome of whether it was a draw or had an outright winner.
2. If any prediction contained any additional events such as the next goal scorer.
3. The confidence in the prediction of the outcome compared to the outcome. In this case, the higher the confidence, the more certain the outcome is likely to be.

After determining the correctness of the match, an evaluation of the data that produced those results will be taken. This will review whether there were any outside factors interfering with the logical conclusion that needed to be taken into account. Following the review, a discussion could identify whether there are potential extensions or factors that need to be



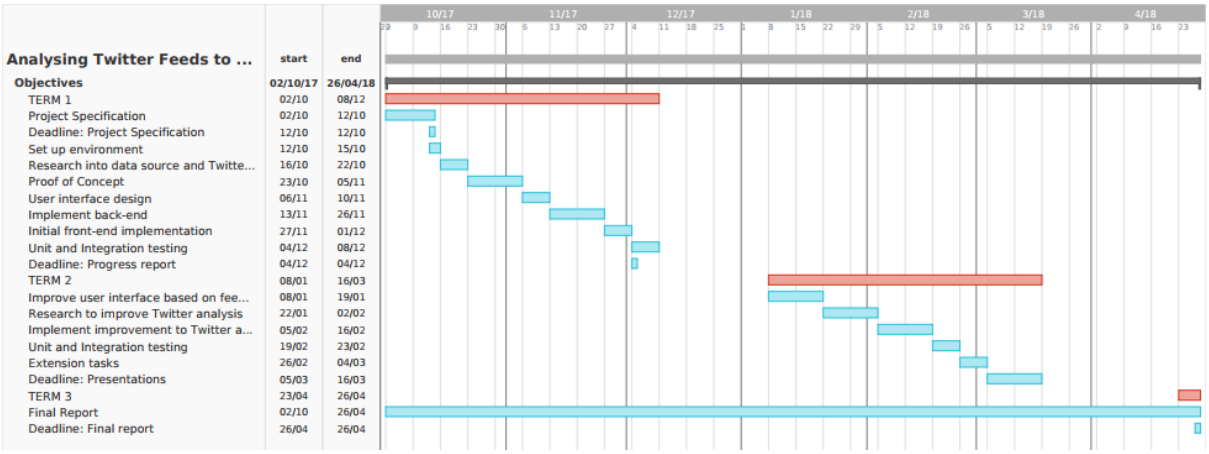
included in further iterations of the project. Any factor that influences the final outcome but can not be calculated will be identified and explained as well as having solutions produced to reduce the impact of such instances.

PROJECT MANAGEMENT

The project has certain deadlines which need to be followed. To keep on track with the project, I will be meeting my supervisor once a week to ensure that deadlines set are reasonable and continually achievable.

Timeline

The project has been split into manageable sections. This is shown by the following Gantt chart showing not only the sections but the expected deadline for each.



Analysing Twitter Feeds to ...	start	end
Objectives	02/10/17	26/04/18
TERM 1	02/10	08/12
Project Specification	02/10	12/10
Deadline: Project Specification	12/10	12/10
Set up environment	12/10	15/10
Research into data source and Twitte...	16/10	22/10
Proof of Concept	23/10	05/11
User interface design	06/11	10/11
Implement back-end	13/11	26/11
Initial front-end implementation	27/11	01/12
Unit and Integration testing	04/12	08/12
Deadline: Progress report	04/12	04/12
TERM 2	08/01	16/03
Improve user interface based on fee...	08/01	19/01
Research to improve Twitter analysis	22/01	02/02
Implement improvement to Twitter a...	05/02	16/02
Unit and Integration testing	19/02	23/02
Extension tasks	26/02	04/03
Deadline: Presentations	05/03	16/03
TERM 3	23/04	26/04
Final Report	02/10	26/04
Deadline: Final report	26/04	26/04

Risks

This project has a set of risks associated with it that could disrupt the progress of the project and some dependencies that it relies upon.

- The project depends on the availability of the Twitter API [4](3rd party) and therefore if access is no longer available or has changed, then data needs to be collected from a different source or updated accordingly.
- The prediction of outcomes can only run against real results obtained when there are actual matches being played. To remove this dependency for testing, data can be taken from a certain time period and compared against a known outcome.
- There is a chance of illegitimate data being included as the source of the data is primarily from other people. As a solution to data tampering, other pure statistical checks can be made as well as prioritizing data from users whom Twitter has personally already verified.
- The broad scope of the different football matches available (countries, age categories, gender) might make it hard to make a generalised search, especially as some would return insufficient data. A resolution to this is to limit the search to a specific football league and expand that depending on demand.

Resources

The technology used will be based on web technologies as well as data processing and as such the resources used will reflect that. The following list contains the intended resources

that will be used.

1. Git - Used for version control
2. GitHub - Used for its cloud storage, integration with git as well as a back up of the project
3. Twitter API - The bulk of the information will be coming from using this publicly available API
4. MongoDB - Document storage "NoSQL" database
5. Python - Data processing language
6. HTML, CSS, Bootstrap & JavaScript - Front-End languages to display content to the user

LEGAL, SOCIAL, ETHICAL AND PROFESSIONAL ISSUES

Twitter has a developer agreement which defines the way data can be used and the appropriate use of their service. Opinions constructed by other people using the service can be used as long as the Developer Policy is followed [2]. By using data that is publicly available as well as complying with Twitter's Policy, there should not be any legal, social, ethical or professional issues.

REFERENCES

[1] trello.com, *About | What is Trello?,* https://trello.com/about, Accessed 7 October 2017.

[2] twitter.com, *Developer Agreement and Policy,* https://developer.twitter.com/en/developer-terms/agreement-and-policy, Accessed 8 October 2017.

[3] fifa.com, *2014 FIFA World Cup,* http://www.fifa.com/worldcup/news/y=2015/m=12/news=2014-fifa-world-cuptm-reached-3-2-billion-viewers-one-billion-watched--2745519.html, Accessed 5 October 2017.

[4] twitter.com, *Twitter API,* https://developer.twitter.com/en/docs, Accessed 6 October 2017.

[5] optasports.com, *Opta Classic Data feed,* http://www.optasports.com/services/media/data-feeds/classic-data-feeds.aspx, Accessed 10 October 2017.