Development of a Model Used to Forecast the Outcome of a Rugby Game Using Machine Learning Algorithms

The End Goal

Conor Grant
Data Analytics
X13522167

Declaration Cover Sheet for Project Submission

SECTION 1 *Student to complete*

Name: Conor Grant
Student ID: x13522167
Supervisor: Michael Bradford

SECTION 2 Confirmation of Authorship

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature: **Conor Grant** Date: **11/5/2018**

Table of Contents

Executive Summary

This report documents the approach taken by the researcher to complete the software project as part of the BSc (Honours) Computing program in the National College of Ireland. As part of the project, data will be gathered on teams competing in the Guinness Pro 14 Rugby competition and analysed in order to attempt to determine the winner.

The system for the analysis consists of a database for storing and analysing real data on statistics from every rugby match in the league, over the past two seasons. Machine learning algorithms such as K Nearest Neighbour, Naïve Bayes, Decision Trees, Random Forest and Neural Networks have been applied to the datasets to predict the outcome of the games, along with some statistical methods like time series analysis and multiple linear regression. These methods are applied to the data in RStudio and the results are visualised through the use of an interactive web application.

This document has been created in order to assist the reader in comprehending what will happen over the course of the year and also the approach taken to complete different milestones.

The 4th of December marked the mid-point of the project and many of the techniques that were needed in order to make predictions on the dataset would only be taught in semester two. The 13th of May is when we have to upload the project in its entirety.

1 Introduction

1.1 Background

The purpose of this study is to see if one can predict the outcome of a rugby game using machine learning algorithms with a degree of certainty. Data mining has grown over the last few years and has been successfully implemented in areas such as fraud detection, market basket analysis and customer segmentation. As well as implementing it with these useful applications its use is becoming increasingly more popular in sport to attempt to predict the outcome and score of games. However, it has consistently been a research problem due to the number of factors that can affect a game, which is out of the algorithm's scope, such as management decisions, weather and the luck of both teams.

Rugby in Ireland has always been a hot topic to talk about socially. Its popularity has only grown further after the national team beat the All Blacks, denied England the grand slam in 2017 and went on to win the grand slam in 2018. Leinster, Munster, Ulster and Connacht also all regularly compete with some of the best teams across Europe. After Ireland won the grand slam in March many people wondered whether the sport has over taken Gaelic football and hurling to become the peoples game in Ireland. The big teams in Ireland are the national team and the provincial teams Leinster, Munster, Ulster and Connacht who compete in the Guinness Pro 14 against teams from Scotland, Wales and South Africa and in the Champions Cup were they tackle the best teams in Europe. What has significantly helped the rise of rugby in Ireland over the last number of years is that nearly all of the games are televised, which gives the fans the opportunity to watch their team in action and also that the teams have been very successful. With the rise of interest more people are talking about rugby and making assumptions on who will win the upcoming games.

Is there a way to change the way fans make assumptions about the game? Instead of them guessing or making an informed decision, they could use a model to determine the outcome and understand the patterns which could swing the game in their team's favour.

1.2 Aims

The main objective is to analyse the games from the Guinness Pro 14, and in doing so identify patterns which will determine the result of the match. The following is a list of aims which will assist the researcher to achieve this.

Aim 1: The first aim is to find a dataset with a lot of stats on every game in the Pro14 over the last two seasons. In order to acquire a comprehensive dataset with this information the researcher will contact companies who analyse performance in sports and also search the web to see if it's possible to scrape the data.

Aim 2: The second aim will be to clean and transform the data. The researcher may have to deal with values or transform the data so that it can analysed at a later stage.

Aim 3: The third aim is to create a database to store the data and from there the researcher will be able to query it whenever the data is needed.

Aim 4: The fourth aim is to analyse the data. The researcher intends on using machine learning algorithms and based on the results, determine if there is a trend in correctly predicting the outcome of the game.

Aim 5: Using the results from the analysis, the sixth aim will be to visually represent the results through the use of an interactive web application.

Aim 6: To complete all the documentation associated with this project. This aim will be ongoing throughout the lifecycle of the project.

1.3 Technologies

RStudio

The project will use RStudio to construct this project, which is an open source integrated development environment for R, which itself is a programming language used for statistical computing and creating graphs. RStudio will also be used to retrieve the relevant data from the datasets.

R

R will be used to build this project and visualise the results. The packages used are listed below in the implementation section.

Excel

Excel is a spreadsheet tool developed by Microsoft with built in statistical and graphing commands that allow a user to manipulate the data.

This project will use Excel to hold the data in a structured format before it is cleansed and stored in an SQL server.

MySQL

This project will make use of MySQL servers which is an open source relational database management system owned by Oracle Corporation that supports SQL queries. It will hold the data on this server, making it easy to gain access to the data. Doing this ensures attainability of the data is assured allowing for use on multiple machines.

Tableau

Tableau is a software which allows for instantaneous insight by converting data into visually appealing, interactive visualisations called dashboards.

This project will use Tableau to display all results and interpretations of final figures that are acquired through the datasets, so the end user will easily be able to comprehend the results which are being shown.

These are the technologies that have been implemented to help the researcher complete the project.

1.4 Structure

Throughout the life cycle of this project we will be following the KDD methodology which is very important in data analytics projects. Following this process is key for the project to be completed in full. Below is an example of the process which has been modified to visualise the steps involved in the project and a small explanation on each step involved.

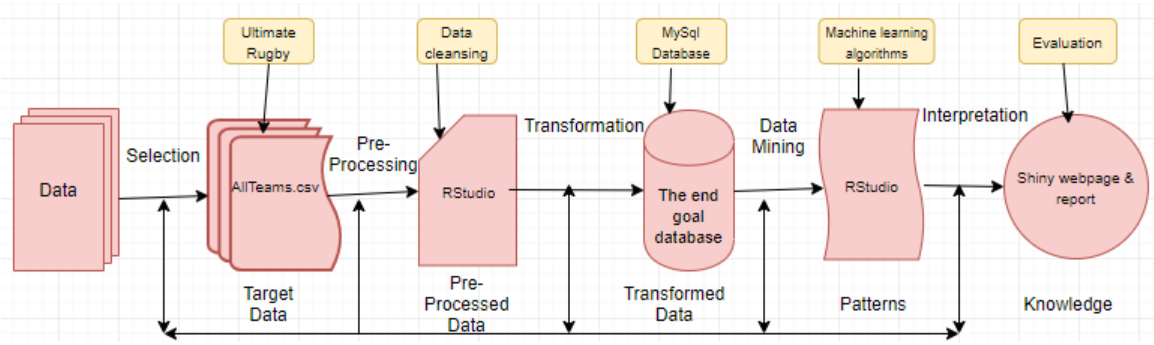


Figure 1: Knowledge discovery in databases diagram

Selection: The selection stage focuses on obtaining the datasets relevant to the project which will then be analysed to discover important information. The data collected was from an app and has to be manually imported to a csv file called "allTeams.csv".

Pre-processing: The pre-processing stage involves the cleansing of data. Data cleansing is the process of noticing and correcting inaccurate, corrupt or obsolete data from a table. The data was examined and there were no missing values to deal with however, for the classification algorithms we do have to declare the "Result" variable as a factor, remove the team name and score difference variables and randomise the data to ensure we obtain reliable results.

Transformation: The transformation stage involves taking the clean dataset and generating better data. The methods includes dimension reduction and record sampling. The transformation process implemented for this project included preparing the data for K nearest neighbour (KNN)and neural network algorithms. For KNN the data will be normalised, so the ranges of dimension do not affect the distance function. For the neural network the transformed data will be scaled as the different ranges in value will negatively affect the algorithm. After this step the data is then stored in a database.

Data Mining: The data mining stage of the KDD process searches for patterns in the data which may be of interest. It involves applying machine learning algorithms to find trends and relationships in the data and allow for predictive analysis to be performed. In relation to this project KNN, Naïve Bayes, Decision Tree, Random Forest and Neural network were

all implemented to discover if it was possible to use machine learning models to predict the outcome of a rugby game.

Interpretation: Interpretation is the final stage in the KDD process and it involves evaluating the discovered knowledge. To complete this final step, the results from the data mining stage are used and applied with visualisations to get a better understanding of the results recorded. To do this the package "Shiny" in R was used to create interactive web pages.

1.5 Research

Before we could continue with the project, research was required in order to determine whether similar studies had been conducted and if this was the case, how this project would differ. After searching on websites such as Google Scholar, UCI and Kaggle, the researcher only found one similar project, however other sport analytic projects have been performed on different sports such as golf and football, as those sports are more popular around the world. One obstacle from no projects such as this being performed in rugby was that there were no datasets currently available on the web.

The aim of the similar project found on the web was also to predict rugby scores [1]. In that project the researchers tested their model on international rugby games, using the teams rank from the world rankings which is updated and maintained by World Rugby and the teams' previous results. The data was gathered from ESPN's website and this data is not relevant for this project. They only applied linear regression to attempt to predict the score difference between teams whereas in this project we will be using multiple data mining techniques to predict the outcome of each game.

The researcher also looked at how bookmakers determine who will be the favourite for each match. In team sports such as rugby, the odds are determined by taking into account who the home team is (as they would have an edge), team form and team stats. Using these factors, they apply regression techniques to produce their odds.

We also studied the Guinness Pro 14 at great length. The "pro14rugby" website provided a good starting point in understanding more information about each team, such as how they

performed last year, and it also provided plenty of statistics on each team. There was also plenty of newspaper articles about each team regarding their goals for the year, player recruitment and an overall overview of each team.

1.6 Definitions, Acronyms and Abbreviations

KDD: Knowledge discovery in databases is a methodology that is used in data analytics projects. It is the process of finding useful information from a collection of data. This technique uses data preparation and selection of data, data cleansing, data mining and interpreting accurate solutions from the results of the project.

Database application: A database is a graphical user interface were a user can setup a database to store data and retrieve the data whenever they wish to do so.

Visualisation application: A visual application allows the user to visually display and present their results based on the data. It is used at the end of the project when the user has something to show and it can display anything, such as a simple X and Y graph.

Programming application: A programming application is an API that has subroutine definitions, tools for building applications and protocols. It would enable the user to manipulate datasets.

Algorithm: In computer science and mathematics an algorithm is a set of rules or process to be adhered to in calculations or other problem-solving operations. Algorithms can perform tasks like data processing and calculations.

Machine learning: Machine learning is a scientific method used in data analysis which automates an analytical model. They use algorithms which learn from the data and it allows the computer to find hidden relationships and insights without exactly being programmed where to look.

Pro 14: The Pro 14 is a professional rugby union league tournament in which teams from Ireland, Wales, Scotland, Italy and South Africa compete. The researcher will do the project on matches in this league.

GitHub: GitHub is an online Git repository service. It offers source code management functionality and version control. The user can upload their programming code here via the GitBash command line.

Google Drive: Google Drive is a cloud service for sharing and storing files. The user can upload their documents here to back them up.

ETL: Extract Transform Load is the methodology used to extract, transform and load data to the database and also used for data storage. It also includes the processes of retrieving data.

Microsoft Excel: Microsoft Excel is a spreadsheet program in Microsoft Office. Excel presents data in a table view with rows and columns that can be manipulated by the operations and functions that are built in.

SPSS: Statistical Package for Social Science is used for logical batched and non-batched statistical analysis.

Data cleansing: Data cleansing is the process of noticing and correcting inaccurate, corrupt or obsolete data from a table. The user can either remove the columns or input them if they're missing.

Data source: Data source is where the user will get the data from. This could be downloading files that are already available online or scraping the web for information.

MySQL: MySQL is an open source relational database management system. It stores information in a MySQL database in the form of related tables.

SQL: Standardised query language is used to retrieve data from the database. This is the standard language for querying relational database management systems.

Data mining: Data mining, also known as analysing data, is the process of discovering patterns, relationships, knowledge and insights from datasets and involves using methods of statistics and machine learning.

RStudio: RStudio is a programming application that is free and an open source development environment for R, a programming language for statistics.

R: R is a programming language for statistical computing supported by the R foundation.

Python: Python is another example of a programming language that will be utilised. It's used for general purpose programming. It will used to scrape data from web pages using a loop feature.

Beautiful Soup: Beautiful Soup is a Python package that is used for parsing HTML and XML documents, it can then be used to extract data from HTML.

Tableau: Tableau software helps people use data to solve problems by allowing them to visualise it. It makes analysing data quick and easy.

Ultimate Rugby app: This is an app which stores information about rugby games.

2 System

2.1 Requirements

2.1.1 Functional requirements

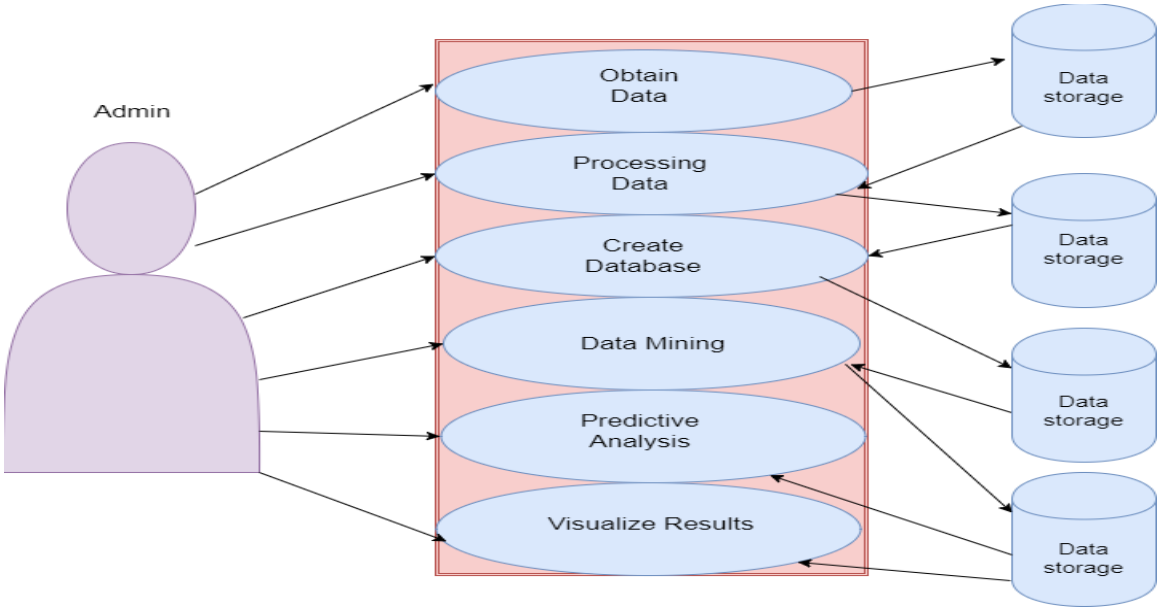
The functional requirements in this project will be completed by applying the KDD methodology. The important stages in a data analytics project are data selection, processing data, data transformation, data mining and then finally the evaluations.

The functional requirements are all about how the researcher interacts with the system. For example, how the researcher will go about obtaining data, cleansing the dataset by removing columns that aren't required and analysing the data.

Each functional requirement will either be of priority 1 or 2. Priority 1 meaning it is a must have for the project and vital if we wish to achieve the goals of the project and 2 meaning it is also a must have but there are alternatives.

2.1.2 Use Case Diagram

The Use Case Diagram provides an overview of all functional requirements.



2.1.3 Requirement 1 Obtaining the data

2.1.3.1 Description & Priority

This requirement is essential as the user can't analyse anything without the data, so it would be considered a priority 1. Once the data is obtained it can start to be explored.

2.1.3.2 Use Case

Obtaining the data.

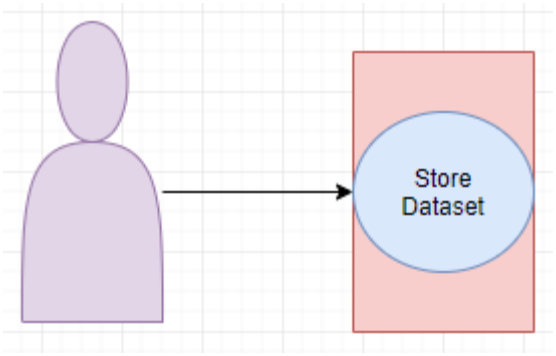
Scope

The scope of this use case is to manually obtain raw data.

Description

This use case describes the process in manually obtaining data. Subsequently the relevant files are stored in a secure location.

Use Case Diagram



Flow Description

Precondition

There must be valid data on the internet.

Activation

This use case starts when the Administrator starts searching the web for data.

Main flow

1. The Administrator logs into "Ultimate Rugby" app.
2. Selects the statistics for every game.

3. The Administrator manually inputs the data into a CSV file.
4. CSV file is stored in a secure location and in the correct format.

Alternate flow

A1: The data is corrupt

1. The app containing the data doesn't seem to be accurate.
2. Check another website to verify the information.
3. The use case continues at part three of the main flow.

Exceptional flow

1. The relevant app is down.
2. Microsoft Office is not accessible.

Termination

Collect all the relevant data and it is now stored in a file on the computer and saved.

Post condition

Data is obtained, and the selection process of the KDD methodology is now complete.

2.1.4 Requirement 2 Processing Data

2.1.4.1 Description & Priority

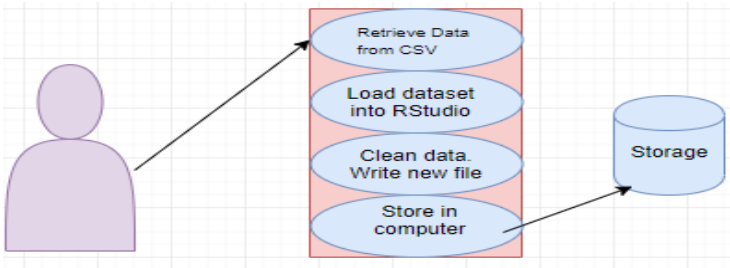
Processing data use case entails data cleansing to obtain the most accurate figures and statistics. This would be considered a priority 2 requirement.

2.1.4.2 Use Case

Scope

Processing the data improves the administrator's ability to conduct the appropriate analysis.

Use Case Diagram



Flow Description

Precondition

The data has been collected and is in a wait state.

Activation

This use case starts when the Administrator retrieves the data.

Main flow

1. The Administrator retrieves the data from the file on the computer.
2. The Administrator loads the dataset into RStudio.
3. The Administrator checks the data for missing values.
4. The Administrator removes any variables which will not be needed.
5. The Administrator randomises the data.
6. The Administrator writes a new file with the clean data using R.
7. The Administrator exits all applications.

Exceptional flow

E1:

1. The Administrator is unable to retrieve the dataset from the system.
2. The Administrator checks that all path files leading to the dataset are correct.
3. The Administrator fixes the errors.
4. The use case continues at point 3 of the main flow.

Termination

Data cleansing has now been performed on the dataset and the use case is complete.

Post condition

The new dataset is now stored on the computer, waiting to be retrieved.

2.1.5 Requirement 3 Creating a Database

2.1.5.1 Description & Priority

This requirement would be considered very important towards completing the project. It's so important because it allows the data to be stored in a safe reliable place, where it can be easily retrieved.

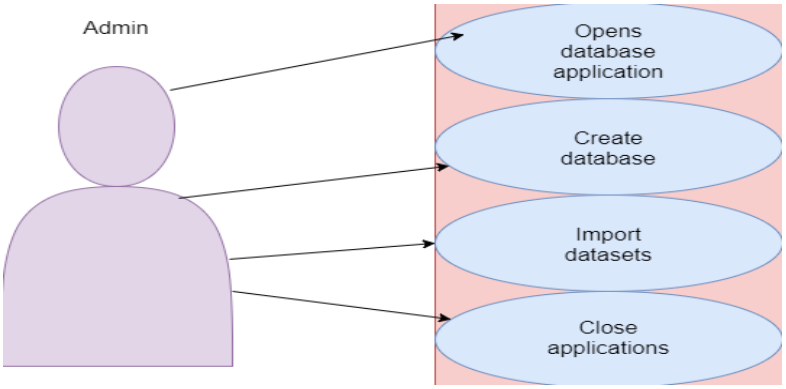
2.1.5.2 Use Case

The Administrator creates the database which acts as storage for all the relevant datasets for the duration of the project.

Scope

The scope of this use case is to create a place where we can store all the data using a database application.

Use Case Diagram



Flow Description

Precondition

The database must be accessible at all times after it is created.

Activation

This use case starts when the Administrator accesses MySQL, opening the application in order to create the storage.

Main flow

1. The Administrator opens MySql.
2. The Administrator creates a new database and calls it “the_end_goal”.
3. The Administrator imports the dataset from computer to the storage using R.
4. The Administrator exits the database application.
5. The Administrator exits the programming application.

Alternate Flow

A1: The user creates a schema for the database

1. The Administrator uploads the CSV file to SQLite.
2. The website creates a schema for the database with all the information from the CSV file.
3. The Administrator enters the Schema into MySql in the query box.
4. The database is created.
5. The Administrator exits off the applications

Exceptional flow

E1: Encountered error with the dataset

1. The system states that an error has occurred and that the dataset is corrupt in some way.
2. The Administrator checks the dataset saved on the computer, finds the error and corrects the problem.
3. The use case continues at point 3 of the main flow.

Termination

The database is created with the dataset imported. The use case is now terminated.

Post condition

The database is setup with datasets imported waiting to be used.

2.1.6 Requirement 4 Initial analysis

2.1.6.1 Description & Priority

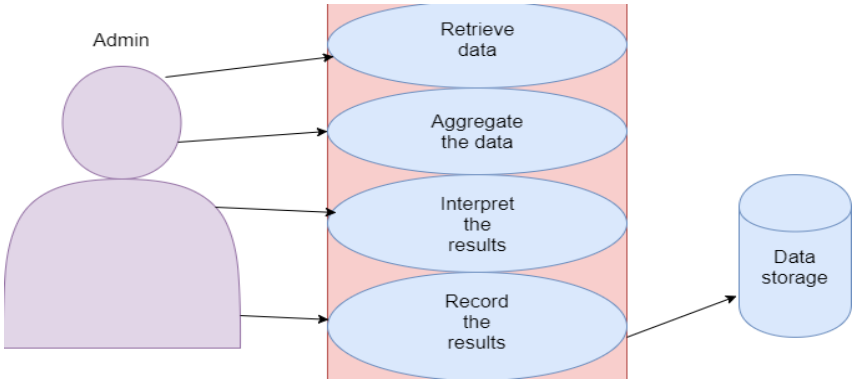
In order for the project to succeed the data will have to be analysed. Here the user applies some basic statistical methods to the dataset. This would be considered a priority 2 requirement.

2.1.6.2 Use Case

Scope

Data will be retrieved from the database, brought into a programming application were scripts will be applied to analyse the data.

Use Case Diagram



Flow Description

Precondition

The data is in the database waiting for the admin to retrieve it.

Activation

This use case starts when the Administrator retrieves the data from the database.

Main flow

1. The Administrator gets the data from the database.
2. The data is analysed using R.
3. The Administrator interprets the results.
4. The Administrator record the results.
5. The Administrator closes all applications.

Exceptional flow

E1: Unable to retrieve data

1. The Administrator is unable to retrieve the dataset from the system.
2. The Administrator checks that all path files leading to the dataset are correct.
3. The Administrator fixes the errors
4. The use case continues at point 2 of the main flow.

E2: Programming application error

1. Error in the code and the programming language won't display the results.
2. The Administrator investigates the error online.
3. The Administrator fixes the error.
4. The Administrator tests the code to make sure it runs properly.
5. The use case continues at point 2 of the main flow.

Termination

The data has been analysed, results have been recorded and the use case is now closed.

Post condition

The dataset is back in the database again and is waiting for the admin to retrieve it.

2.1.7 Requirement 5 Predictive analysis

2.1.7.1 Description & Priority

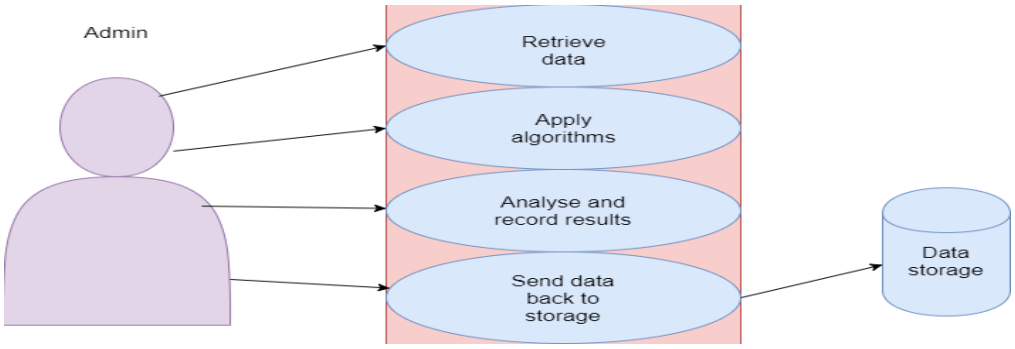
The predictive analysis requirement is a level 1 requirement, it's an exploratory requirement. It involves implementing machine learning algorithms to forecast the outcome of a rugby match.

2.1.7.2 Use Case

Scope

The admin will retrieve the datasets and apply classification algorithms to them. The scope is to predict the correct outcome of the rugby match.

Use Case Diagram



Flow Description

Precondition

The data is in the database waiting for the admin to retrieve it.

Activation

This use case starts when the Administrator retrieves the data from the database and loads the dataset into a development environment.

Main flow

1. The Administrator retrieves the data from the database and loads it into a programming application.
2. The Administrator runs the data through a machine learning algorithm.
3. The Administrator records the results.
4. The Administrator sends the data back to storage.
5. The Administrator saves the work and closes down all the applications.

Exceptional flow

E1: Algorithm error

1. The programming applications can't run the algorithm.
2. The Administrator investigates the error online.
3. The Administrator reviews the code and fixes any errors.
4. The use case continues at point 2 of the main flow.

Termination

The machine learning algorithm has been successfully applied to the dataset and the use case is over once the results have been stored.

Post condition

The dataset is back in the database again and is waiting for the admin to retrieve it.

2.1.8 Requirement 6 Visualise results

2.1.8.1 Description & Priority

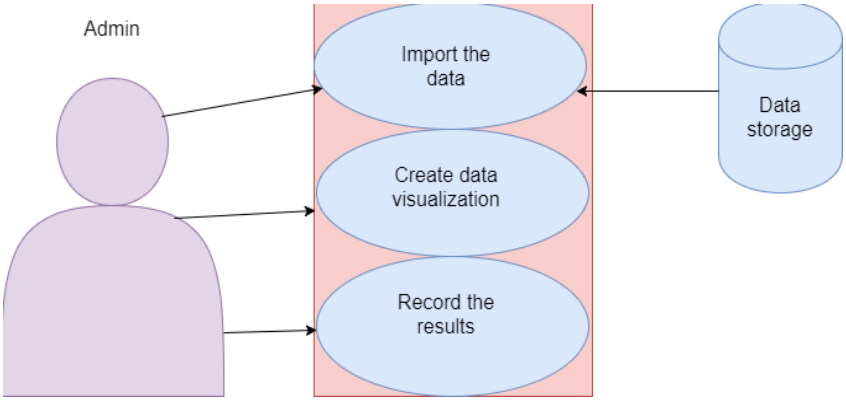
Visualising the results is the last use case within the project. This use case is ranked as a priority 2 requirement. It is important to be able to present the findings however the more important work is in the other use cases.

2.1.8.2 Use Case

Scope

The scope of this use case is to import the analysed data to a visualisation application. The tool will then help us interpret and present the results of the project.

Use Case Diagram



Flow Description

Precondition

It's the last step in my project so all data should be analysed and complete in order to show the final results in the project.

Activation

This use case starts when the Administrator accesses the data visualisation tool.

Main flow

1. All the analysed data is imported to a data is in RStudio.
2. The Administrator creates the user interface for the web application.
3. The Administrator connects with the server.
4. The Administrator uploads the results of the classification models.
5. The Administrator saves all the work.

Exceptional flow

E1: Unable to upload data

1. The Administrator is unable to load the data to the data visualisation tool.
2. The Administrator checks the problem and fixes it.
3. The use case continues at point 1 of the main flow.

Termination

This use case is complete when all visualisations and representations are complete, and the information is saved.

Post condition

The admin is now able to report on its project and be able to present its findings.

2.2 Non-Functional Requirements

2.2.1 Performance/Response time requirement

High performance and response times are seen as a high priority in most system architectures and it would be useful to have it implemented, however it is not a priority in this project. The project will allow the user to analyse the information in their own time.

2.2.2 Availability requirement

The data shall always remain available to the system throughout the project scope.

2.2.3 Recover requirement

Recoverability is a high priority for this project. It must be ensured that all data and code is fully recoverable in the case of server errors or hardware failures. Cloud storage such as Google Drive and GitHub will be utilised in order to back up all data and documents relating to the project.

2.2.4 Security requirement

Security is a fundamental aspect to the system, however the raw data that is obtained which comes in unstructured form will be from open source websites and therefore does not have security protection. The work will be conducted using a personal laptop that is fully password protected and the data will be hosted on a secure server with a password needed to gain access. The data administrator will have full access to everything and strong passwords will be used.

2.2.5 Reliability requirement

The datasets are composed by developers who own the Ultimate Rugby application and are updated and maintained by them. This requirement is of high priority, in order to get accurate results we need the data to be reliable. The datasets shall contain values and attributes in the right order, for the correct analysis to be performed.

2.2.6 Extendibility requirement

The project could very well be extended depending on the answer to the question. Phase two of the project will involve predicting the results of the game before it has started so that it can be incorporated into betting.

2.2.7 Resource utilisation requirement

Hardware such as a personal laptop, college computers along with internet access, backup storage devices, programming and visualisation tools will all be used.

2.3 Data requirements

There was one dataset required to fulfil the project objectives. The dataset was on the team statistics for every game within the Pro 14 league over the 2016/2017 and 2017/2018 seasons. Finding a data source proved difficult, however open source data was acquired from the "Ultimate Rugby" app which is the only place the data is available. The app can be downloaded from Google Play Store. The data consists of 546 rows with 18 variables and is as follows:

Table 1: All Teams dataset

Column Name	Type	Description
Team	Nominal	Name of the opposition
Result	Binary	1 = Win, O = Loss
Possession	Numeric	Control of the ball by one team
Territory	Numeric	Time spent in the other teams half
Tries	Numeric	Way of scoring 5 points
Conversions	Numeric	Way of scoring 2 points
Penalties	Numeric	Way of scoring 3 points
Metres	Numeric	Distance covered with the ball
Defenders beaten	Numeric	Number of defenders beaten
Clean breaks	Numeric	Breach of the line of defenders with the ball

Gain line carries	Numeric	Moving forward with the ball in possession
Passes	Numeric	Giving the ball to teammate
Offloads	Numeric	Passing the ball while being tackled
Turnovers won	Numeric	Dispossessing the opposition team
Kicks from hand	Numeric	Kicking the ball while in possession
Rucks won	Numeric	Clearing out the breakdown
Score Difference	Numeric	Score difference at the end of the game
Venue	Binary	1 = Home, 0 = Away

There are very few draws in professional rugby games. This is due to the fact that there are numerous ways to score with a variety of points attached, meaning the chances of each team scoring the same number of points are naturally low. A team coming from behind usually won't aim for a draw but instead choose a different points method (taking a chance to score a try instead scoring a penalty) which makes a win or a loss the more likely result. To get around this problem all draws have been removed from the dataset. There was only seven draws across both seasons, so this won't affect the result.

The data has all been entered into the database, team after team (i.e. all statistics from Leinster's games followed by all statistics from Munster's games followed by Connacht's and so forth). This would mean when we're selecting our train and test datasets for the machine learning analysis part of the project the data would not be stratified (representative of the data set as a whole) and we couldn't ensure reliable results. To get around this the data must always be randomised using the "Runiff" function and by setting the seed to the same number we can make that dataset unique so that we know when comparing algorithms, they're using the same test dataset.

Before building our machine learning models we must also make the columns "Team" and "Score difference" null. We do not need to know the team name for making predictions so that is irrelevant and score difference will only inform the model of the outcome of the game, which we do not want. Score difference is used for the time series analysis.

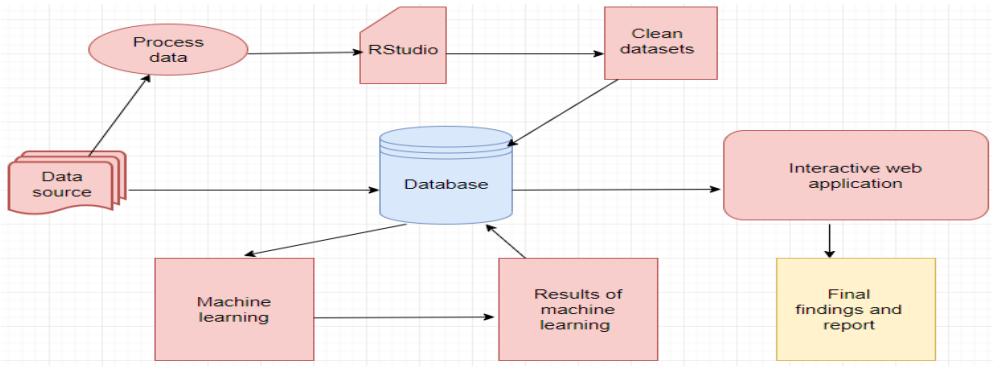
The dataset above was meant to be acquired from the website “sportradar” using an API key which would then allow the user to have all the data required in an XML file [2]. A script was implemented using Python which could extract the relevant data from the excel file. However, the website’s data feed for this file stopped working and was not available, attempts were made to contact the company in the hope that they would fix the bug however they proved to be unsuccessful.

Please note we tried to obtain the data from other websites who sell the relevant sports data. They were contacted, asking if we could use their data in order to complete the project however no response was received until we contacted them again and were quoted €500 for a basic data feed and €3000 for a more complex data feed, unfortunately this was slightly out of the price range to complete this project.

2.4 Design and Architecture

The following diagram shows the architecture at a high-level view which will be utilised in the project. The database component is central to the system and it will interact with all the different components. Once the raw data is obtained and stored we can then start processing the data to remove empty columns or input values. Once we’ve processed the data we can load it back into the database and then the analysis can begin. The data will run through machine learning algorithms to find relationships, patterns and insights within the data that will determine the outcome of a rugby match before and during the game. The system will then provide the results of the analysis through visualisations and representations for the end user.

Design and Architecture



3 Implementation

This section outlines all significant analysis conducted throughout the life cycle of this project. All of the R code used in the analysis is included with the submission. Tableau was used along with R for exploring the data.

3.1 Important R packages

- Shiny
- Rpart
- Rattle
- Rpart.plot
- RColorBrewer
- Party
- Partykit
- Tree
- Neuralnet
- Xtable
- E1071
- randomForest
- gmodels
- caret
- class
- ggplot2
- plyr
- RMySql
- dbConnect
- DBI

3.2 Data Exploration

Some exploratory analysis was conducted at the beginning of this project to explore the idea that machine learning models can be used to forecast the results of rugby games, have a better underatnding of the project and see what could be learned by visualising the data. The data exploration part of the project was completed in RStudio and Tableau.

Here we are trying to visually understand why some matches end in a win or loss through the use of the "ggplot2" function. We are combining the result factor with the venue variable. All the results that end in a loss (0) are respresented in red, the wins (1) are respresnted in blue, away games are marked on the x-axis as 0 and home games are 1 on the x-axis. There are more home wins than away wins, so from this plot we cannot tell if a team will win based on whether the venue for the match is home or away but we can say that teams that have home advantage are more likely to win the match. This could be down to many factors such as the crowd having an influence or the team are more comfortable playing on their pitch.

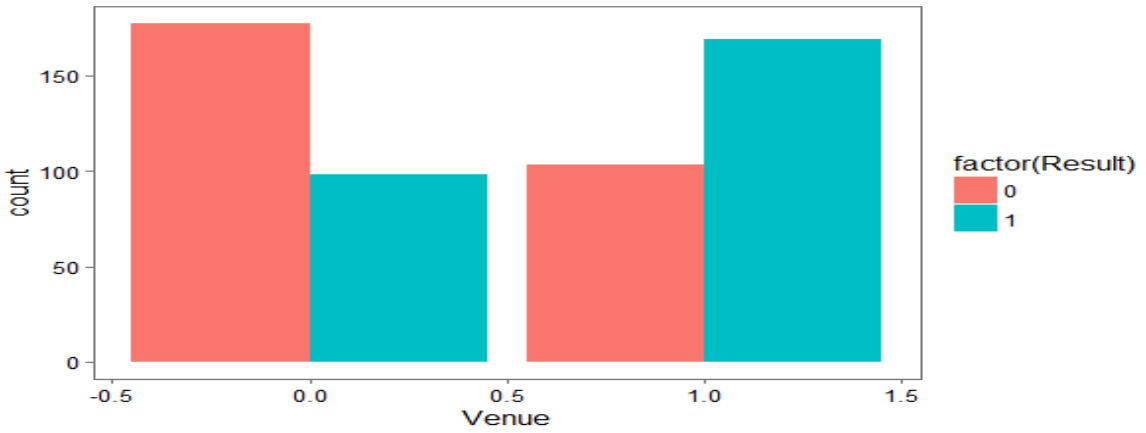


Figure 2: Ggplot of distribution of Venue and Result

We now take a look at the amount of tries scored and metres gained however we split the dataset in two, blue graphs represent the results which ended in a win and the red represent the results that ended in defeat. Figure 3 and 4 represent a packed bubble chart which was created in Tableau. Tries are important in rugby as they offer a higher score than other score types, such as a drop goal, and usually the team who scores the most tries in a game will win. In figure 3, most of the results which ended in a win, the team scored three or more tries. In figure 4 we can see that teams who lose are not scoring as many tries. The main bubbles in the second graph are 0-3 whereas in the first graph the most occurring bubbles were 2-5. Within both graphs there are outliers for example in figure 4 there are a few occasions where a team has scored 5 tries and still lost the game so we can't say with confidence that from the amount of tries scored we will be able to predict the outcome however it gives us a good indication.

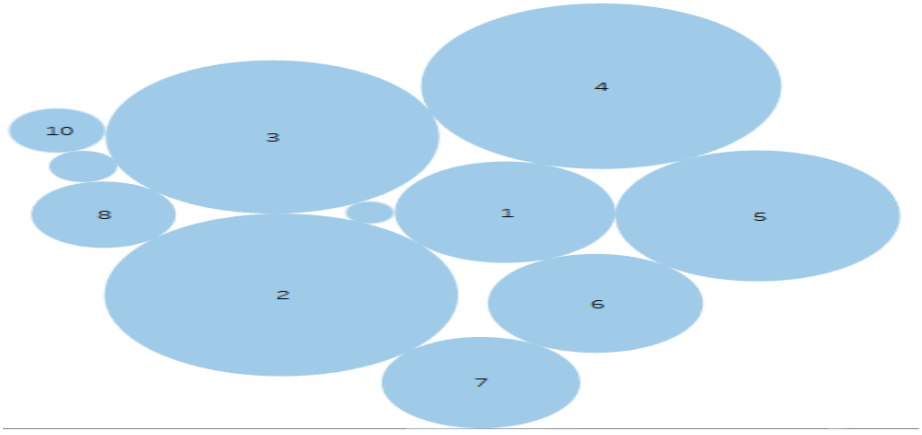


Figure 3: Tries scored by teams who won

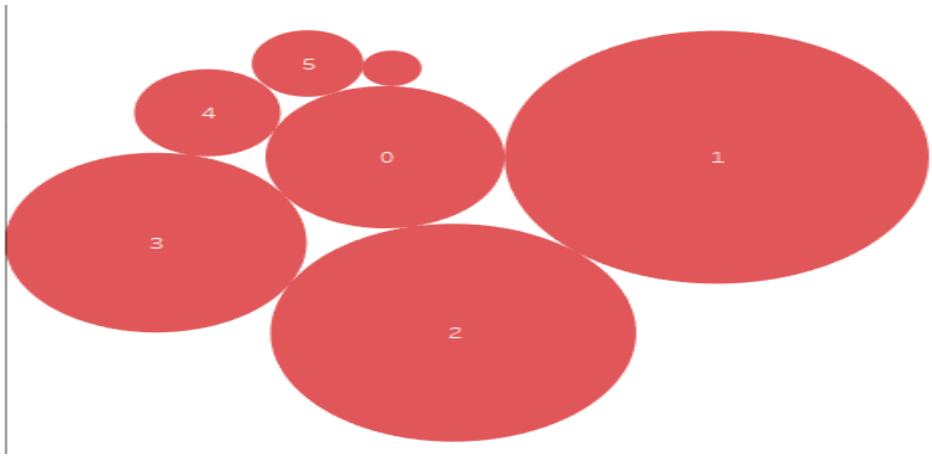


Figure 4: Tries scored by the losing team

Figures 5 and 6 represent the amount of metres gained while in possession, through the use of a histogram which again was created in Tableau. Both graphs are in a bell shape which indicates a normal distribution. As shown in figures 5 and 6 the average metres gained is slightly higher for teams who ended up winning the match which you would expect. In figure 5 the histogram peaks at 450 and in figure 6 the histogram peaks at 300. Based on the variable metres gained a user would not be able to make an accurate prediction on the result of the game.

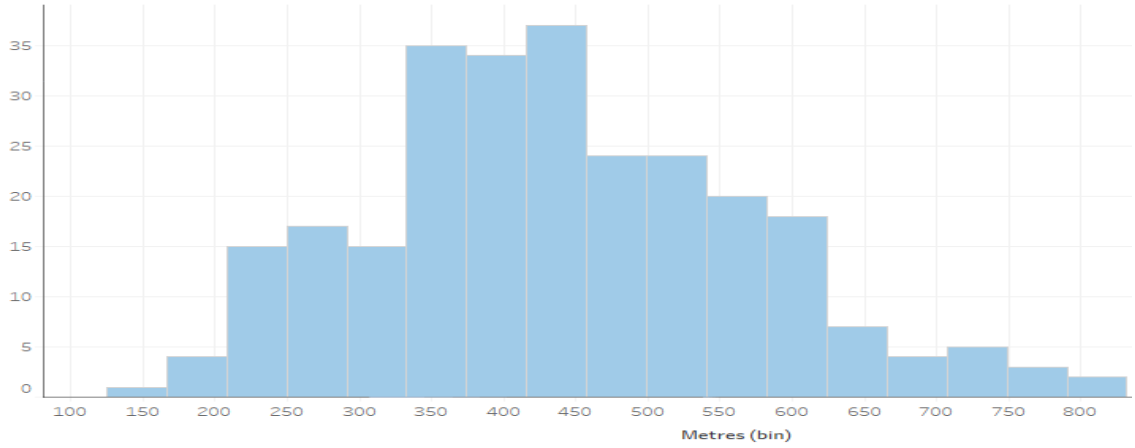


Figure 5: Metres gained by the winning team

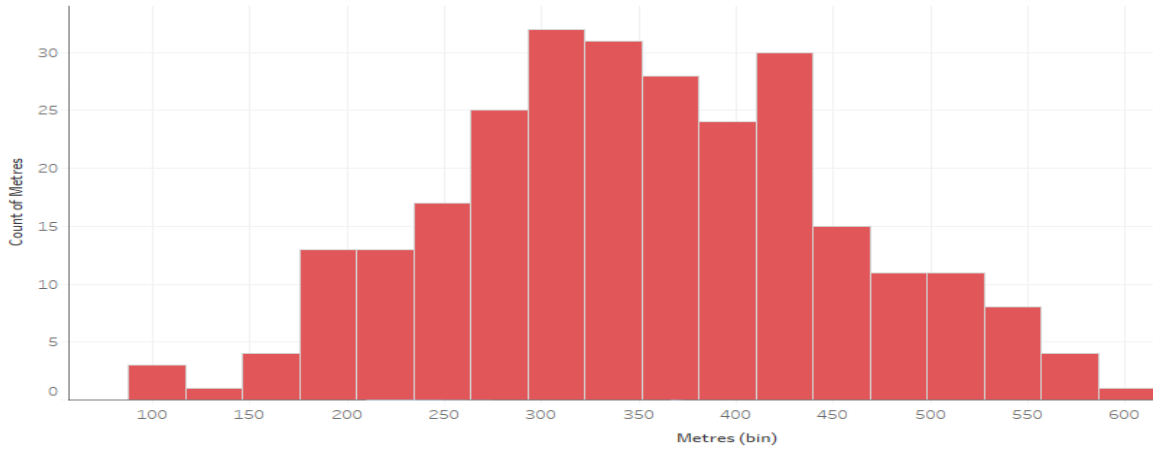


Figure 6: Metres gained by the losing team

3.3 Initial Analysis

Here we carry out some simple statistical analysis to find about more about our data. We will be doing some time series analysis, multiple linear regression and factor analysis. Time series analysis is a method of forecasting which examines the pattern of time series data. A weakness of the method is it only looks at past behaviour to predict the future. The method of time series analysis we'll be using is exponential smoothing which forecasts the score by taking into account the actual score in the current period and the forecast which was made previously for the current period, to do this we'll use the "holtwinters" function.

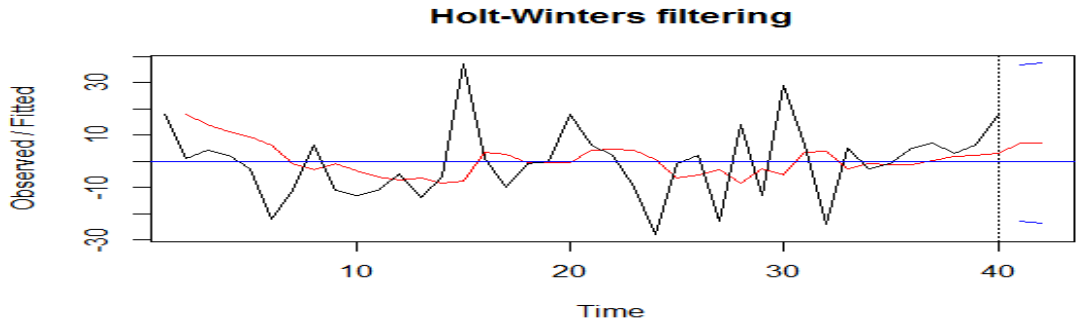


Figure 7: Time series analysis

For the time series analysis, we are only using the "score difference" variable and trying to predict that. The abline in blue runs along the x axis, anything above this is counted as win and anything below is a loss. The red line represents the time series analysis prediction and

the black line represents the actual results. The team's results are very mixed with plenty of defeats and wins. By stating that the forecast which was made previously for the current period is "4" the time series model predicts that the team will win their next two matches.

Linear regression uses one independent variable to predict one dependent variable. The principle being if one variable can predict an outcome with some degree of accuracy, then why couldn't two or more do a better job which is where multiple linear regression comes into it. The equation for the regression model is as follows: $y = a + bx_1 + bx_2$.

X_1 is the first independent variable, X_2 is the second independent variable, b is the slope of the line, a is the y intercept of the regression line and y is the dependent variable we are trying to predict. The hypothesis for this test is as follows: H_0: p=0 (No relationship between the independent variables and dependent variable), H_1: ≠0 (There is a relationship between independent variables and the dependent variable). In R we use "lm" function which will build our regression model.

```
Coefficients:
          (Intercept)           teams$Tries        teams$Penalties  teams$Kicks.from.hand
              -0.41566               0.20123               0.09244               0.01123
        teams$Venue
              0.05802
```

The dependent variable that we are attempting to predict is "Result" and the independent variables are tries, penalties, kicks from hand and venue. If the equation adds up to greater or equal to one the team won the match and if it is less than one they lost. R generates the y intercept for us and when we add the numbers shown in the image above we get an answer of (y = -.41566 (+ 0.20123*4) + (0.09244*3) + (0.01123*18) + (0.05802*1)) = 0.92674 meaning the team lost this match according to our model.

The multiple R-squared value provides a measure of how well our model as a whole explains the values of the dependent variable. The closer the value is to 1.0 the better the model explains the data. Since the R squared value is 0.3494, we know that nearly 35% of the variation in the dependent variable is explained by our model which is not sufficient to rely on this model to provide accurate results in the future. We can use multicollinearity to look at the relationships between the x variables and check for redundancy.

Multicollinearity can have a negative effect on the model fitting process and return answers that are inconsistent.

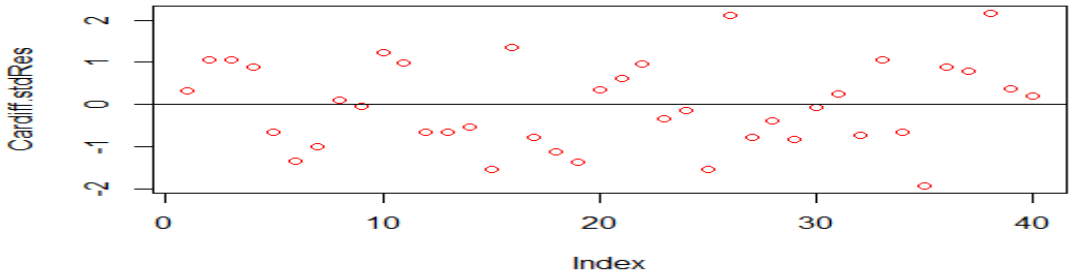


Figure 8: Residual distribution

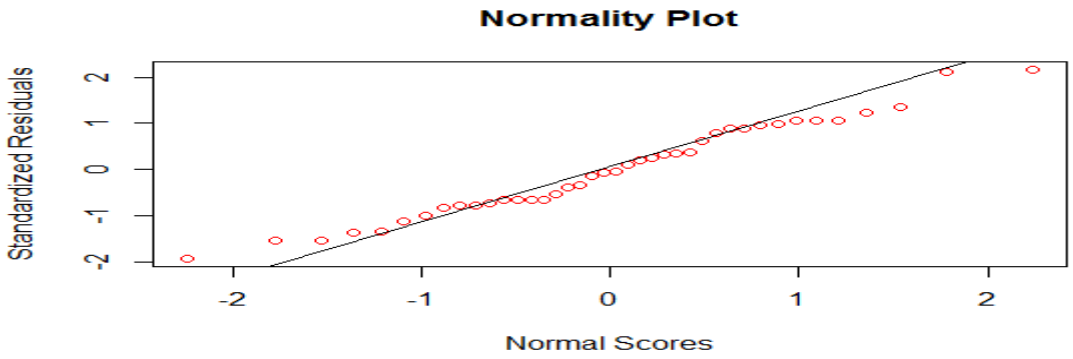


Figure 9: Normality plot

In the first image above, we can see that most of the standardised residuals fall within two standard deviations of the mean. Ideally, we want to see more residuals hovering around zero and they shouldn't be spread out as much.

The second image is the normality plot. We want all the residuals to fall as close as possible to the line however they do not, and they form an S shape which indicates that there's multicollinearity. Based on the graphs and the R squared value we can reject the null hypothesis in favour of the alternative hypothesis, there is a relationship between independent variables and the dependent variable. Moving forward we now know we can't use this model.

Factor analysis is a statistical procedure that reduces a large number of variables into a smaller set of variables and it can establish underlying dimensions between measured variables and latent constructs, thereby allowing the formation and refinement theory. We run a principal component test to see if there are many similarities among are variables. We will use all the variables except for “Result”, “Team” and “Score difference”. First, we check the correlation matrix to see if there is high or low correlations between the variables. There is high correlations between possession and territory, tries and conversions, metres and clean breaks, passes and rucks won and low correlations between all the other variables. On the first attempt using the “fa” function as well as setting the rotate to the “oblimin” it fails to run. We then attempt it again using the “factanal” function and setting the nfactors as 11 based on the results from the correlation matrix. It reduces our variables from 15 to 11 by grouping the variables with high correlations together. On completion of the analysis we are not satisfied the results and when possible will continue to use all the variables available when applying it to machine learning algorithms.

3.4 Machine learning algorithms

Implementing machine learning algorithms are the most important requirement for the project. Here we hope to find patterns and relationships among the data to be able to predict a rugby game with a degree of certainty. The data is bi-class, trying to predict a win or a loss, so it is a classification problem. Classification is a supervised learning approach in which the computer identifies hidden relationships and patterns from the input data to be able to classify the new observation. The types of classification algorithms that have been selected for this project are: K Nearest Neighbour, Naïve Bayes (linear classifier), Decision Tree, Random Forest and Neural Network [3] .

3.4.1 K Nearest Neighbour

Things that are alike are likely to have properties that are similar. KNN uses this principle to categorise data by placing it in the category with its nearest neighbour [4]. It forms a majority vote between the k most occurring incidences to a given number of unseen observations which in this project will be the “teams_test” data. A distance metric among

the two points governs the similarity and the metric used, the Euclidean distance formula is represented below:

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

The algorithm is used with numeric data, which all the variables are, however the distance function can be affected by the potential ranges of the dimensions within the data. If there was one dimension with binary values and another with values ranging between 0 and 1000 it would ruin the algorithm. To get around this we must transform the data. A prevalent method for rescaling KNN is the min-max normalisation. The transformed data will have values between [0,1]. Xnew = X−min(X) /max(X)−min(X), is the method for normalising the data. Subtract the minimum of feature X from each value and divide it by the range of X. Could be taken as a percentage of its original min/max value. By using the "lapply" function within R we can automate the process instead of individually applying it to each feature.

Before running the algorithm, we must also decide on what K should be. Deciding on how many neighbours to use for KNN outlines how well the model will generalise upcoming instances. Trying to find a balance between a small and large K so you don't under or over fit the data. A small K allows outliers to affect the outcome, and a large K limits the effect caused by noisy data but can bias the learner so that it affects small but significant patterns. Best K value is somewhere in between the two and a common practice is to use the square root of the training instances and not to select K as an equal number. The square root of the training data was 20.90 which was then rounded up to 21 and this was used to begin with. After testing the model and trying it with different K values it was soon discovered the most optimal K value was 17.

To run the KNN function we use the package "Class" and to evaluate the model we use "caret" to avail of the confusion matrix and "gmodels" for the Crosstable function. A training dataset is used to produce the KNN model and a test dataset will be used to estimate the predictive accuracy of the classifier. The training data is set as 80% and the remainder 20% will be used as testing data. The function "prop.table" is used to ensure that two

datasets are stratified to make sure we get reliable results. The code takes the "Result" factor in column one of the data frame and creates the vectors.

The function uses four parameters for training and classification: data frames with training and testing models, vector with the class for every row and K which specifies the number of nearest neighbours. It returns a factor vector which we have called "KNN_model". We then evaluate how well the predicted classes did in comparison with the "teams_test_labels". We use the confusion matrix to determine the quality and performance of the method.

```
          teams_test_labels
KNN_model  0  1
        0 45 19
        1  5 41

               Accuracy : 0.7818
                 95% CI : (0.693, 0.8549)
    No Information Rate : 0.5455
    P-Value [Acc > NIR] : 2.17e-07

                  Kappa : 0.57
 Mcnemar's Test P-Value : 0.007963

            Sensitivity : 0.9000
            Specificity : 0.6833
         Pos Pred Value : 0.7031
         Neg Pred Value : 0.8913
             Prevalence : 0.4545
         Detection Rate : 0.4091
   Detection Prevalence : 0.5818
      Balanced Accuracy : 0.7917

       'Positive' Class : 0
```

Figure 10: Confusion Matrix

At the top of the diagram we can see that the True negative rate is 41% and the true positive rate is 37%, the model is better at identifying negatives than positives. True negatives represent the model predicting a loss and in reality, the team did lose whereas a true positive represents a win when the team did win. Overall accuracy of the model is 78%. Numbers on the diagonal from top left to bottom right are correct decisions and everything else is an error. Other pieces of important information from the matrix are 95% confidence interval, 90% sensitivity which indicates the proportion of positive examples correctly classified, 68% specificity which is negative examples correctly classified and importantly kappa value of 0.57. Kappa statistic adjusts the notion of accuracy by also accounting for the possibility that a correction prediction is chance. Values are in the range [0,1], 1 is very

rare. 0.57 would be considered a moderate agreement when it comes to predicting a sports event.

3.4.2 Naïve Bayes

Classifier founded on Bayesian methods which utilises training data to calculate an observed probability of each class based on feature values. When the algorithm is used on unlabelled data, it uses the observed probabilities to forecast the most likely class for the new instances. It's a simple idea but it results in a method that often has outcomes on par with more sophisticated algorithms.

The algorithm uses Bayes theorem to compute a posterior probability that measures how likely the team are to win or lose the match. The notation P(A|B) means the probability of A given that event B happened. The formula for Bayes theorem is as follows [5]:

$$p(B \mid A) = \frac{p(A \mid B)\ p(B)}{p(A)}$$

To run the naïve Bayes function, we use the caret and e1071 and gmodels for the Crosstable function. We specify that we're trying to predict the "Result" factor in the training data and then run the code. We make predictions by using our naïve Bayes model with the test data and use the Crosstable to check the accuracy of the model.

```
Total Observations in Table:  110

                    | prediction
teams_test$Result   |         0 |         1 | Row Total |
--------------------|-----------|-----------|-----------|
                  0 |        46 |         4 |        50 |
                    |     6.321 |    11.062 |           |
                    |     0.920 |     0.080 |     0.455 |
                    |     0.657 |     0.100 |           |
                    |     0.418 |     0.036 |           |
--------------------|-----------|-----------|-----------|
                  1 |        24 |        36 |        60 |
                    |     5.268 |     9.218 |           |
                    |     0.400 |     0.600 |     0.545 |
                    |     0.343 |     0.900 |           |
                    |     0.218 |     0.327 |           |
--------------------|-----------|-----------|-----------|
       Column Total |        70 |        40 |       110 |
                    |     0.636 |     0.364 |           |
--------------------|-----------|-----------|-----------|
```

Figure 11: Cross-table function

The results of the model are as follows: True negative rate 41%, true positive rate 33%, false positives 3%, false negatives represent 23% and the total accuracy is 74%. Based on the results this model is predicting more losses than wins, 70 losses compared to 40

wins which could mean the model is struggling to identify patterns associated with wins. The kappa value for this model is 0.5 which falls in the moderately acceptable category.

3.4.3 Decision Tree

A machine learning technique that applies an approach of dividing data into smaller and smaller portions to identify patterns that can be used for prediction. Knowledge is represented as logical structures and decision trees have a high explanatory power. The tree is terminated by terminal nodes, also known as leaf nodes. Terminal nodes deliver the classification based on combinations of decisions and data travels through the tree from root to leaf nodes.

The model is built using the c5.0 package but first we must split the data into training and test datasets. The ratio used is 80/20 meaning there's 437 rows of data in the training dataset. Using the c5.0 function we train the model by declaring that we're trying to predict the Result factor and will use all the variables within the train dataset. On completion of the model, the summary function will give us a good indication of how the algorithm is working and what the important variables are.

```
Attribute usage:

100.00% Tries
 82.15% Penalties
 72.54% Conversions
 56.75% Kicks.from.hand
 42.56% Venue
 31.35% Gain.line.carries
 23.34% Metres
 21.05% Offloads
 14.19% Territory
  6.64% Passes
  3.89% Defenders.Beaten
  3.43% Possession
  3.20% Rucks.won
  1.83% Clean.breaks
```

Figure 12: Attribute usage

As we can see the most important variables are "Tries", "Penalties", "Conversions", "Kicks from hand", "Venue" and "Gain line carries". We then test the model with the test dataset using the predict function and evaluate it. The tree size was 35 and it was 75% accurate.

The next step involved was to try and boost the model to hopefully find a better result, we would be training another model but this time a series of models. When training the model,

we use the same information as the first time however we had “trials = 15”. When we assess the new model, it does in fact improve the accuracy. It correctly predicts 88 of the results which equates to 80% accuracy. The kappa value is 0.602, falling into the good category meaning it’s a reliable prediction. The tree however is too big to plot, would pruning the tree also improve the accuracy?

Pruning a tree involves reducing its size, so that it generalises better to unseen data. There’s two methods, post and pre-pruning. We decide to post prune the tree as it uses pruning criteria to reduce the tree, often more effective because it’s difficult to determine the optimal tree size, it ensures all important decisions are included and overall the most commonly used strategy is to post-prune the tree [6]. Tested various splits ensued by examining the cross-validation error through “printcp” and “plotcp” functions. The first diagram below shows the lowest XError of all the splits and displays the important variables. The “plotcp” function provides a visual representation of the cross-validated error summary. It shows the result of the CP values plotted against the mean to show the deviation until the minimum value is reached. The graph displays that the tree is most optimal at seven terminal nodes.

```
Variables actually used in tree construction:
 [1] Clean.breaks       Defenders.Beaten  Gain.line.carries Kicks.from.hand
 [5] Metres             Offloads          Penalties         Possession
 [9] Territory          Tries             Venue

Root node error: 207/437 = 0.47368

n= 437

         CP nsplit rel error  xerror     xstd
1 0.4154589      0   1.00000 1.00000 0.050424
2 0.0531401      1   0.58454 0.58454 0.045188
3 0.0241546      3   0.47826 0.48792 0.042572
4 0.0144928      5   0.42995 0.48792 0.042572
5 0.0096618      6   0.41546 0.54106 0.044090
6 0.0048309     12   0.34783 0.62319 0.046064
7 0.0000000     15   0.33333 0.64251 0.046468
```

Figure 13: Complexity parameter table

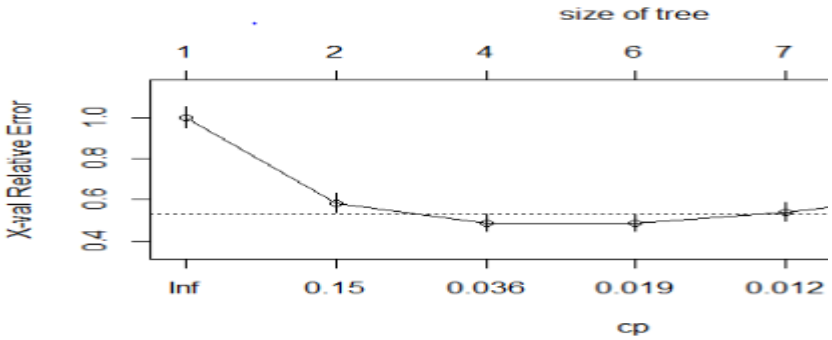


Figure 14: Complexity parameter table

We are now ready to train the new model. The idea here is to let the decision tree grow fully and observe the CP value. We cut the tree with optimal CP value which in our case is 0.012. When we compare the new pruned tree model with our test dataset by computing the accuracy of the pruned tree we discover the result is 82.72%. The true negative rate is 42% and the true positive rate is 40%. The pruned tree is more accurate than the boosted c5.0 model.

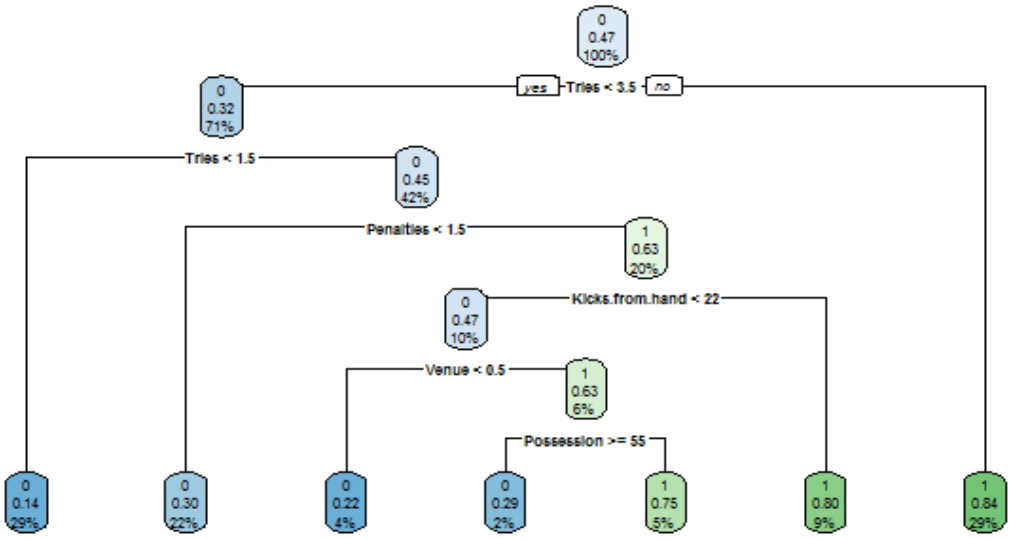


Figure 15: Decision tree

Above is our new decision tree model which has seven terminal nodes. As we can see there are four rules for the "Result" = 0 and three rules for the "Result" = 1.

3.4.4 Random Forest

This is a collaborative learning method for classification that operates by creating a host of decision trees at training time and outputting the class that is the mode of the classes. Decision trees have a habit of over fitting their training set and random forests amends this. The random forest seen in this section was built using the "party" and "randomforest" packages. The purpose of the algorithm is to predict the outcome of the results. The model was built using all the variables.

To start with the training size for the model was 80% and we would then test the accuracy on the other 20%. When using a big sample as the training size you would expect good results. We run the random forest function through the train dataset, within it we specify that we're trying to predict the "Result" factor and call it "teams_forest_predict". When that is complete we use the predict function to run the model with the test dataset, use the table function to check the accuracy and confusion matrix to return the kappa value.

```
  teams_forest_predict
    0  1
0 47  3
1 16 44                    Kappa : 0.6591
```

As we can see this model is 82.73% accurate which is so far the best result and returns a kappa value of 0.65 which falls in the good category meaning the model is good and a correct prediction is not made by chance. Overall the model is predicting more negatives than positives which is similar to the other models. In the model we are using the default parameters within the algorithm set by the rule of thumb. Could we improve the performance of the model by tuning the parameters?

We tune the algorithm parameters through R using tools that come with the algorithm, the caret package and making our own parameter search. When creating the train and test datasets we continue to use the 80/20% split as that is what we have used above. The two parameters that will be tuned are "mtry" and "ntree" which are the ones that are likely to have the most effect on the accuracy. "Mtry" represents the quantity of variables at each split and "ntree" is the amount of trees to grow. Start off by using the recommend defaults for each parameter, "ntree" =500 and "mtry" =7. The caret package offers an excellent facility to tune the parameters, however only "mtry" parameter can be tuned in caret. "Ntree" is different as it can be as big as you want, and the accuracy continues to grow up to a certain point. This could be limited by compute time.

Random search is a strategy that we can use to try random values within a range and is a good way to overcome any biases. We use a ten-fold cross validation and repeat it three times to slow it down but also to limit overfitting. Using caret, we try a random search for mtry.

```
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   3    0.7629316  0.5231407
   9    0.7469168  0.4918745
  10    0.7522727  0.5025132
```

As we can see from the results of the random search the mtry value which is most optimal is 3, accuracy was used to select the most optimal. It also has the highest kappa value.

Another strategy looked at is grid search which states the grid of algorithm parameters to try. Every axis of the grid is a parameter and points in the grid are specific groupings of parameters. It is a linear search through a vector of values because we are only modifying one parameter.

```
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 393, 393, 393, 393, 393, 393, .
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   1    0.7520613  0.4996541
   2    0.7482030  0.4933285
   3    0.7566949  0.5110775
   4    0.7528189  0.5033745
   5    0.7542812  0.5063796
   6    0.7467054  0.4909849
   7    0.7467935  0.4914036
   8    0.7482734  0.4944606
   9    0.7374912  0.4725927
  10    0.7352713  0.4677589
  11    0.7398344  0.4774546
  12    0.7238020  0.4446930
  13    0.7223221  0.4413191
  14    0.7368041  0.4705968
```

Figure 16: Mtry cross-validation

As we can see from the results of the cross validation the mtry with the highest accuracy is again three with 75.66% however it doesn't beat the default algorithms accuracy.

The next option for tuning the algorithm is by creating your own parameter search. There is two ways of doing that either by tune manually, or create extension to caret which adds in additional parameters. The method chosen to use is to extend caret. It is the same as the random forest algorithm, however it allows multiple tuning of several parameters. One problem with using this method is that it takes a long time for it to run and uses a significant compute effort. By defining a list that contains several custom named elements like how to predict, we can define our own algorithm. When we have our custom list, we can use the train function to tune different values for mtry and ntree [7].

```
mtry  ntree  Accuracy   Kappa
 1    1000   0.7412438  0.4777795
 1    1500   0.7488020  0.4926453
 1    2000   0.7488020  0.4924089
 1    2500   0.7472868  0.4898269
 2    1000   0.7495948  0.4957377
 2    1500   0.7503700  0.4973608
 2    2000   0.7473397  0.4912353
 2    2500   0.7503347  0.4970208
 3    1000   0.7534355  0.5045170
 3    1500   0.7549331  0.5072756
 3    2000   0.7489429  0.4953605
 3    2500   0.7542459  0.5059060
 4    1000   0.7511628  0.4998512
 4    1500   0.7549859  0.5078420
 4    2000   0.7588443  0.5157435
 4    2500   0.7580691  0.5142964
 5    1000   0.7542636  0.5065902
 5    1500   0.7542107  0.5065072
 5    2000   0.7535060  0.5052659
 5    2500   0.7542459  0.5066295
 6    1000   0.7549683  0.5082825

custom_predicted  0  1
              0 47 18
              1  3 42
```

Figure 17: Cross-validation

Above, is a sample screenshot from the console in R, the list continues to mtry 15. The most accurate values are for mtry and ntree are 4 and 2000 with an accuracy of 75.88%. Using these values in the model we then run the random forest function with our test dataset. The results are above in the table, true negative rate is 42%, the true positive rate is 38% and total accuracy is 80%. This means that after tuning the algorithm we now know the most accurate random forest model is the one which uses the default parameters.

3.4.5 Neural Network

The last machine learning network implemented was the neural network model. As usual we are trying to predict the result based on features such as possession, territory, tries scored, gain line carries and many more. Neural network is a supervised black box method typically used for classification problems among other things. It's composed of layers of perceptrons, when networked together networks are very powerful. Neural networks model the relationship between a set of input and output signals.

The data for this algorithm is loaded in through a csv file called "Book1" and the only difference between it and the other file which is loaded in through the database is in this file the variables are separated from each other with a semicolon. When the data is loaded in, you must make sure that you set the semicolon as the delimiter. When you don't include this step, it will affect the next step which is scaling the data down. The formula for scaling the data down is "center = min, scale = maxs-min", we then use the "lappy" function to automate this process throughout the data instead of applying it individually. After we've

scaled the data, we must set up our training and test datasets, as usual we use the 80/20 split.

We are now ready to build the neural network, using the "neuralnet" package. "XO" represents the result so we specify that we are trying to predict that, include all the other variables, set hidden nodes to 16, stepmax is "1e6" and the "linear.output" is T[7]. It won't run without the stepmax set and all it does is allow more time for the algorithm to converge. Having built the neural network, we're now able to plot it too visualise what it looks like, below is the plot:

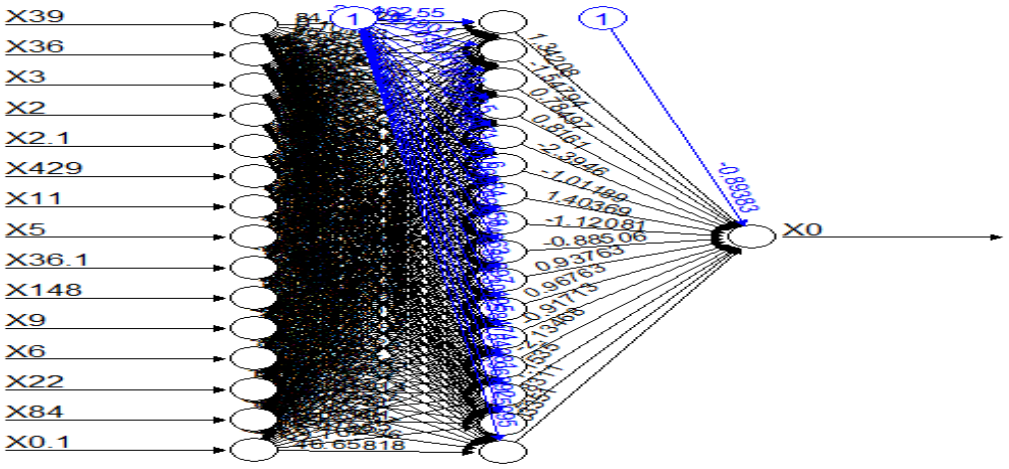


Figure 18: Neural network

As we can see from the plot, the neural network is working to try and find all possible patterns within the data which will determine the result. From the above diagram the model is difficult and even confusing to understand as there is a lot of variables. Maybe using less variables would improve the model but for now we continue with this.

Finally, we're able to use our trained model for the prediction. We need to scale the prediction and the actual back up to the original range before we can compute any error metric. We use the compute function to predict the scaled rating and after this is complete we can check the accuracy of the model. The model works out to be 70% accurate which is not performing as well as the other algorithms. One reason for this could be that the neural network as seen from the plot above is quite complex and could be taking things

into account that it should not be. The true negative rate is 39% and the true positive rate is 30% for this algorithm.

After completing the model, we decide to build a new one however this time we will only be using 6 variables. We keep the target column and other columns useful for predicting and put them in a data frame together. The columns we have decided to keep are those with the highest "variable importance" from the summary of the decision tree, these variables are tries, conversions, penalties, kicks from hand and gain line carries. We use all the same steps as the last time but when we go to build the network we set the hidden nodes as 6 and can remove the stepmax as it is not required this time and will just take longer for it to converge. When we plot the tree, as seen below we note that it is a lot easier to understand with less variables and we hope it will improve the results. The accuracy of the model turned out to be 65% accurate so removing the variables negatively affected the performance of the algorithm.

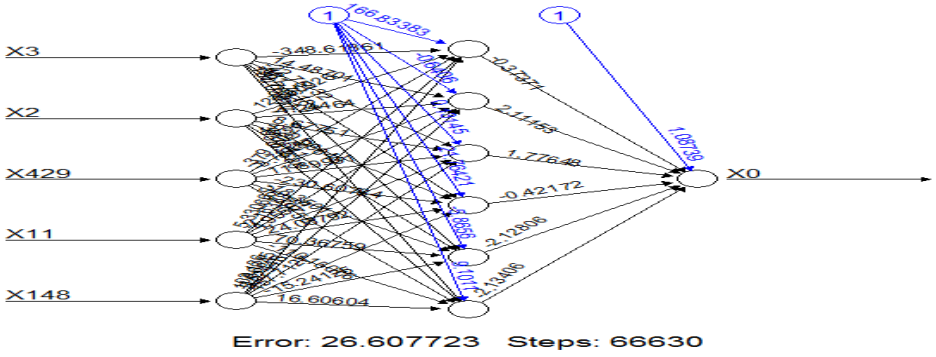


Figure 19: Neural network

3.5 Shiny

This is an R package that makes it very easy to build interactive web sites straight from RStudio. Shiny combines the computational effect of R with the interactivity of the web. No web development skills are needed, and it uses its own servers to host the dashboards [8]. It allows us to integrate our machine learning algorithms and to demonstrate them on Shiny.

The first step in implementing this package is to install it [9]. Next, we must recognise the template code by connecting with the server and setting up our user interaction function.

We then try and include our first machine learning algorithm which was the decision tree. In the user interface section, we outline how many different tabs we want and how many pages, our shiny app will have five different pages for each machine learning model and five tabs on each page for “first 5 rows of the data”, “Model result”, “Model plot”, “Model summary” and “evaluation” which will be a confusion matrix. The user will be able to select whatever page they want on the left-hand side of the web page, at the top they can flick in between tabs and on the side the user will be able to decide on the training data %, the min value is set at 0.5, max value at 0.9 and the default value is 0.75, if the user changes the training data the results will automatically change.

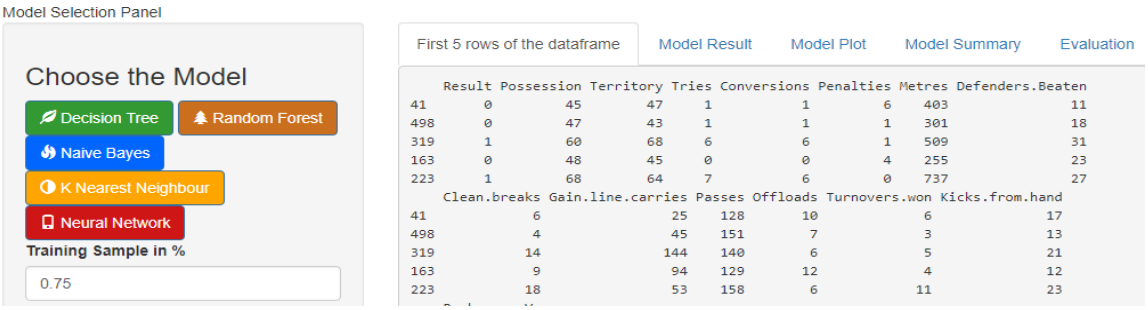


Figure 20: Shiny web application

Before we enter the machine learning code into the server function we must first ensure that the data and packages needed to run the algorithms are installed. Here we can make any adjustments to the data, for example we can normalise the data for the K nearest neighbour algorithm.

Now we enter the machine learning code into the server. Each model will only run when you select that page. The neural network takes a minute or two to run so patience is required. Less variables have been used to increase the speed but this will affect the accuracy.

Shiny is very useful for visualising the models and it allows the user to see which models perform the best, how they work and allows them to play around with the training size to determine what is most optimal for a good model.

4 Testing

The importance of software testing cannot be overstated. Especially when using a programming language where not even a compiler checks the basic consistency of a file. Unit testing is vital throughout the life cycle of this project and each use case is tested to ensure all the components perform as intended. The first unit test is conducted in MySql and all the other tests are conducted in RStudio. Each unit test outlines the purpose of the function, the expected and actual outcomes. The expected outcome details the necessary output for the function to work as intended. Testing will be done on the different use cases.

4.1 Information Schema

Table 2: Information Schema testing

Function	Purpose	Expected Outcome	Actual Outcome	Solution
Information schema	To calulate the amount of rows in database to ensure there's no missing data	The end goal database containing all the data has the right number of rows	As expected	N/A

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE	ENGINE	VERSION	ROW_FORMAT	TABLE_ROWS	AVG_ROW_LENGTH	DATA_LENGTH
def	the_end_goal	allteams	BASE TABLE	InnoDB	10	Dynamic	546	180	98304

The Information Schema unit test conducted in MySql and is displayed on the table above. The purpose of the test was to ensure that all the data transferred correctly to the database and that we weren't missing any rows. Now, that we know all the data is stored in the database we can continue with the analysis.

4.2 Integrity of the data

Table 3: Integrity of the data testing

Function	Purpose	Expected outcome	Actual Outcome	Solution
Gplot and prop.table	To check the integrity of the data and to ensure we have balanced classes	The expected result for this function is the number of 0s and 1s (Wins and losses) will be the same	The observed result were the ones as expected	N/A

The “gplot” and “prop.table” function were used in R to test the integrity of the data. As we’re using data from matches with all the teams over the past two seasons the amount of wins and losses should be the same which would also mean the “Result” variable is a balanced class. This is important as some machine learning classifiers like Random forests fail to cope with imbalanced datasets and as a result they tend to favour the class with the largest proporation of observations (http://amsantac.co/blog/en/2016/09/20/balanced-image-classification-r.html) . However this is not a problem for the dataset we are using, as expected we have the same amount of wins as losses therfore meaning we have balanced classes

4.3 Transformation of data

Table 4: Normalization function testing

Function	Purpose	Expected Outcome	Actual Outcome	Solution
Normalization	To transform all numbers so that the values are between [0,1]	The column selected features will be transformed so that all values are between [0,1]	As expected	N/A

```
test_that("Variables are normalised",{
  test <- normalize(teams_rand$Metres)
  check <- mean(test)
  expect_true(check < 1)
})
```

```
Error: Test failed: 'Variables are normalised'
* check > 1 isn't true.
```

For some alogorithms the data has to be transformed so that the model will work as expected. An example of this type of algorithm is KNN were the data has to normalised so the different ranges of dimension don’t affect the distance formula. We create a nomalisation function which subtracts the minimum of feature X from each value and divide by the range of X. The table above outlines the purpose of the function along with the expected and actual outcomes. Using the “testthat” package in R we were able to test the results. When the code above is run the function passes the test however, if we change

the "<" to ">" sign the test will fail as the values are less than 1. This shows that all the features within that column have been normalised and that the function works.

4.4 Shiny test

Table 5: Web application testing

Function	Purpose	Expected Outcome	Actual Outcome	Solution
Shinytest	To test the performance of the web application itself	The web application is running fine in the browser	As expected	N/A

The Shiny test function unit test was conducted in RStudio using the "shinytest" package and the results are displayed on the table above. As we can see from the table above the web application passed the test and is operating as expected. It's important to test the web application after any changes as modifying the application code or changing data source could potentially break the application.

4.5 Machine learning testing

Unit testing machine learning algorithms is difficult. Some companies are currently researching how you could apply metamorphic testing to machine learning applications but there have not been many breakthroughs to date. One of the best ways to test the algorithms is to check its behaviour when we know the outcome, we will be trying to make the algorithm bomb out/fail in order to learn more about it. Would the model work with data that is N/A? Would algorithm work with a very small training size? We tested these questions using the KNN and the decision tree algorithms.

Table 6: KNN testing

Function	Purpose	Expected Outcome	Actual Outcome	Solution
KNN	Understand the functionality of the algorithm, test how it deals with missing values.	Cannot train the model with any missing values.	As expected.	Before building model ensure train and test datasets are not missing any values.

```
Error in knn(train = teams_train, test = teams_test, cl = teams_train_labels,  :
  no missing values are allowed
```

As we can see from the table above the outcome is as expected, the algorithm cannot deal with missing values. It's important before implementing this algorithm to quickly check the training and test datasets to ensure the data is correct and therefore the algorithm should run as expected.

Table 7: Decision tree testing

Function	Purpose	Expected Outcome	Actual Outcome	Solution
Decision Tree	Use a small training size to see if to see if the algorithm will fail	The decision tree will not pick up on as many patterns within the data, reducing the accuracy	As expected, the tree size is 9 and when using a larger training set its 35.	Ensure the training size is big enough to recognise as many patterns as possible within the data. Let the tree grow as big as it can and prune it if needed.

```
Classification Tree
Number of samples: 100
Number of predictors: 15

Tree size: 9
```

When implementing a decision tree, we test it using a small training dataset to examine the results. The outcome is as expected, the model will build however it will miss out on patterns which may be important and affect the accuracy of the model. In order to prevent this from happening we should build the model with a large training dataset, let it grow as big as possible and prune the tree afterwards if needed which will also ensure all important patterns and decisions are included.

5 Evaluation

5.1 Why some classifiers return better results?

At the start of this project we didn't know which supervised learning classification technique to use and which one would be the most accurate. The algorithms that were used belong to different "classifier families", for example decision trees come from symbolic artificial intelligence and data mining whereas neural networks come from connectionist approaches. The approach taken was to implement five different classifiers using the trial and error method to discover which method was the most accurate, using the confusion matrix and the cross-table function as evaluation techniques.

The classifiers accuracy is affected by the amount of data. Here we are using a relatively small dataset however decision trees and random forest are able to pick up on the key factors which help determine the outcome of the game. The lack of data affects the neural network as there is insufficient data to train the model properly. KNN works well with a relatively small dataset as it means it won't overfit the result. Naive Bayes can work well with a small dataset however it performs better with nominal data compared to numeric data.

There has been a lot of research into comparing the performance of algorithms such as Janez Demsar "Statistical comparisons of classifiers over multiple data sets" however as stated in most of these reports, the best practice is to test multiple algorithms and parameter settings yourself, which has been done here.

5.2 Conclusion

Table 8: Machine learning results

	KNN		Naïve Bayes		Decision Tree		Random forest		Neural Network	
Classification Accuracy	78%		74%		82%		82%		70%	
Crosstable	Predicted									
Actual	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes
No	45	19	46	4	47	16	47	3	43	18
Yes	5	41	24	36	3	44	16	44	15	34

Throughout the project lifecycle, the 80/20 split for the training and test data was used to ensure we can properly evaluate our results along with setting the same seed before randomising the data so the dataset is unique. Table * above shows the overall results that were obtained for each machine learning model that was implemented. As we can see Decision trees and Random forests performed the best, followed by KNN, Naïve Bayes and then the Neural network.

To answer our classification problem, we use the accuracy measure to decide on which model is the best with the most number of correct predictions. However, Decision trees and Random forest return the same classification accuracy, so we must use another performance measure to determine which one is the most optimal and can be used to solve the problem.

In the context of our problem we believe that false positives are probably worse than false negatives. As we do not want our model to predict a win only for the team to go on and lose the match. Taking this into account, we must use the precision measure to evaluate our classification problem.

Precision represents the number of true positives divided by the number of true positives and false positives (TP/TP+FP). Using the figures in table 8 we can work this out. For Decision tree (44/44+16) it's equal to 73% and for Random forest (44/44+3) it's equal to 93%. Based on the precision measure Random Forest is our most accurate model. To be fully satisfied that this is correct we also take into consideration the kappa statistic for each model. The kappa statistic is 0.6491 for the Decision tree model and for the random forest model its 0.6591. This means there's less of a chance that the correct prediction for random forest is by chance.

Our final model using the Random Forest algorithm returned an accuracy of 82% which is very good when it comes to predicting sports games because there are a lot of factors that we cannot control.

5.3 Further development or research

On reflection, if the project had to be done over or if given more time there would be some small changes. Changes to the data may significantly improve the accuracy of the models. Variables which we believe would enhance the models are teams form (take into consideration how the teams have done over the last five games), take their league position into account and possibly including some data around the players involved which would be difficult to rate, but one way of rating could be by taking the amount of international caps the player has, usually the best players will have more caps.

There is scope for applying the same models while the game is in play. To see how the events of the match affect the models and how many times the models predicted outcome will change during the game. Would the decision tree and random forest algorithms still be the most accurate models?

If the project was extended phase two of the project would entail the model predicting the result of the game before it started. The user will be informed of the predictions and also of the confidence interval which is a range of values that we are fairly sure the prediction lies in. The model will predict all the results for the upcoming round of fixtures.

This project touched on only one sport, however, the environment could easily be adapted to find results in other sports such as football or cricket.

6 References

[1] G. Paterson, “Predicting international rugby scores,” *Graham Paterson*, 09-Sep-2017. .

[2] “The Sportradar API | Sportradar US API Portal.” [Online]. Available: https://developer.sportradar.com/. [Accessed: 08-May-2018].

[3] Sifium, “Types of classification algorithms in Machine Learning,” *Medium*, 28-Feb-2017. .

[4] “NCI_DWM2015SS_kNN.pdf.” .

[5] “Bayes’ Theorem: Deceptively Simple.” [Online]. Available: https://freethoughtblogs.com/reprobate/2017/12/25/bayes-theorem-deceptively-simple/. [Accessed: 05-May-2018].

[6] D. R. Bierig, “Decision Trees - H8DWM Data and Web Mining Semester 2,” p. 42.

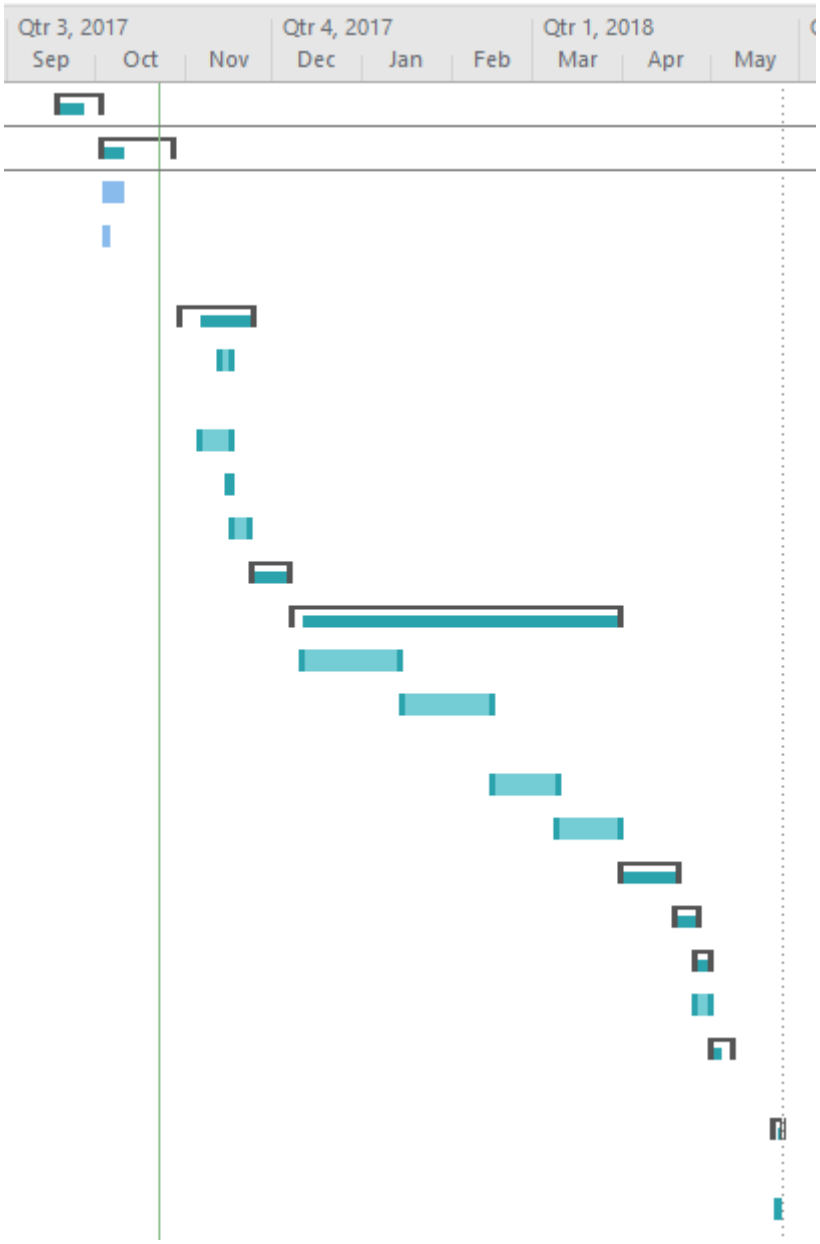
[7] J. Brownlee, “Tune Machine Learning Algorithms in R (random forest case study),” *Machine Learning Mastery*, 05-Feb-2016. .

[8] “Shiny.” [Online]. Available: https://shiny.rstudio.com/. [Accessed: 06-May-2018].

[9] Z. Charpy, “easy 5 steps to build your Shiny app interact with your ML models : decision tree, randomforest, support vector machines, neural network,” 08-Mar-2017. [Online]. Available: https://www.linkedin.com/pulse/easy-5-steps-build-your-shiny-app-interact-ml-models-decision-charpy. [Accessed: 06-May-2018].

7 Appendix

7.1 Project Plan



Project Pitch	**11 days**	**Mon 18/09/1**	**Mon 02/10/17**
Project Proposal	**19 days**	**Tue 03/10/1**	**Fri 27/10/17**
Complete proposal	6 days	Tue 03/10/1	Tue 10/10/17
Upload document	3 days	Tue 03/10/17	Thu 05/10/17
Requirements & Specs	**20 days**	**Mon 30/10/1**	**Fri 24/11/17**
Complete majority of document	4 days	Mon 13/11/17	Thu 16/11/17
Finish document	9 days	Mon 06/11/1	Thu 16/11/17
Send to supervisor	1 day	Thu 16/11/1	Thu 16/11/17
Upload document	4 days	Fri 17/11/17	Wed 22/11/17
Prototype	**9 days**	**Fri 24/11/17**	**Wed 06/12/17**
Development stage	**81 days**	**Fri 08/12/17**	**Fri 30/03/18**
Decision tree	26 days	Mon 11/12/1	Sat 13/01/18
Bayesian Linear regression	23 days	Mon 15/01/18	Wed 14/02/18
Linear regression	17 days	Thu 15/02/18	Fri 09/03/18
Testing	16 days	Fri 09/03/18	Fri 30/03/18
Tableau	**15 days**	**Sat 31/03/18**	**Thu 19/04/18**
Journal	**6 days**	**Thu 19/04/18**	**Thu 26/04/18**
Poster	**3 days**	**Thu 26/04/18**	**Mon 30/04/18**
Draw up the poster	3 days	Thu 26/04/18	Mon 30/04/18
Upload finished project	**6 days**	**Tue 01/05/18**	**Tue 08/05/18**
Project Presentations	**3 days**	**Wed 23/05/18**	**Fri 25/05/18**
Present project	1 day	Thu 24/05/18	Thu 24/05/18

Monday is dedicated solely to working on the project. At the start of every week I will go through everything to see how I'm getting on and what the next steps involved are. This is to ensure that I successfully meet my project milestones and for it to work I have to be honest with myself to make sure I'm going in the right direction. I find it helps to be organised. At the end of each week I also write up what I've done and work on the journal, this is to just make sure that I don't leave the whole journal to the very end of the month and at the end of the project.

In the first semester I am meeting my supervisor every second week to update him on the progress I've made and look for advice. In the second semester I will meet with him every week when I'll be spending more time working on the project.

7.2 Monthly Journals

7.2.1 September

This month, for me was all about focusing on the idea and making sure I was happy with it. My idea was accepted at the pitch and this was down to the preparation work I did for it. My idea is predictive analysis within sport, especially injuries, the sport I'll be focusing on is GAA. I'll be using teams GPS tracker data as my dataset. This is a monitoring device, players wear while training and playing.

My contributions to the project included researching the idea and preparing for the pitch. I researched my idea on the internet by reading relevant websites. I found that no one provides this sort of information for GAA teams in Ireland. A lot of the bigger clubs in all sports hire data scientists who do this work for them and smaller clubs can't afford it.

For the pitch I wrote two pages of notes and memorised them. Then started practicing saying all the information to myself to make sure I wouldn't forget to say anything on the day.

I felt it worked well because my pitch was accepted. I think my idea is a good one and it'll be interesting to work on over the next couple of months as I'm interested in predictive analytics and GAA.

However, I was not successful in obtaining a dataset. It's still early days but I do hope I'll have a dataset by the time I'm writing my next journal. At the pitch the lecturers told me to be careful about the dataset so hopefully I can get it soon.

Next month, I will try to complete the proposal and requirements & specifications. I hope to start working on my project by the end of October so hopefully I can get these documents completed as soon as possible

In order for this project to be as good as I hope it will is I will have to work hard at it so I feel like I need to start working on it as soon as possible.

7.2.2 October

Once my idea was accepted I started working on the proposal that was due on the 27th of October and start looking at gathering the data. My first approach to gathering the data was contacting certain companies which have the data I was looking for, players x, y coordinates on the pitch, heart rate monitor and distance travelled, all this data came from GPS tracking devices. Unfortunately, I had no luck with these companies so instead I contacted GAA teams who collect this sort of data themselves and again was not successful. I knew I had a decision to make with my project and decided that I would be better off to get a new idea.

My new idea is called "Rugby Prophesy" and is all about predicting the end result of a rugby game by applying machine learning algorithms to match stats, team stats and player stats and from there I would produce odds on the outcome of the game and compare them with online sports betting companies' odds for the same matches. I met with Michael and he was happy with my decision because he was a bit concerned as to how I would get the data for my original idea.

Preparing the project proposal was interesting because it gave me the opportunity to sit down and really think about my project, where I could go with it and what new technologies I'd be using to complete the project goals. I created a Gantt chart for my project using Microsoft Project and it was the first time I gave real consideration towards setting goals and deadlines I'd have to meet for my project to be successful.

During the month I also did more research into my new idea and how bookmakers currently decide on the odds they offer. They use match statistics to decide on the favourite and the amount people bet on a certain team also affects the odds.

I am happy with my decision to change my project idea because I was worried about the data and with my new idea the data is on the web so I'll be able to scrape it from there. However, I hope I'll have a dataset soon.

For the month of November, I will continue to do research which will affect my research and specifications document and hope to have gathered all my data for the mid-point presentation.

13/10/2017

In my first supervisor meeting I told Michael how I plan on going about the project and gathering the data. He warned me that gathering the data could be an issue and also advised me to follow the KDD approach towards my project.

27/10/2017

I brought Michael through my new idea and explained to him why I decided to change from my original idea. Presented him with my Project proposal and got him to read through it before I uploaded it.

7.2.3 November

With the mid-point presentation coming up at the start of December I needed to spend half my time working on a prototype and the other half working on the technical report as well as completing projects for other modules such as Web Development and Artificial Intelligence. So far, I've abdicated any responsibility of the prototype as I was hopeful I could get a comprehensive dataset from a company.

After being in contact with a few companies about a dataset and not having any luck, I've decided I must find one on my own. I first searched well known sport websites such as ESPN and sky sports, then dataset websites like Kaggle and UCI but I had no luck. I eventually found a website that would allow me to download the data I needed in either XML or JSON style, through the use of an API key. I signed up for a free trial with the API key which would last 90 days. On their website it said I was able to download team statistics and match events however the team statistics did not work but I decided to continue using the match events.

Michael informed me that I would be able to use beautiful soup which is a Python package that would allow me to extract the data I needed from the XML file. Each file contained about 7000 rows of data. My aim for the presentation was to have this script working along with some small data exploration.

Throughout the month I was working on the technical report. I was referring to previous technical reports which helped me complete this. I brought it to Michael every two weeks to see what he thought of it.

I felt it was important that when I got a chance on a Monday to take 30 minutes to reflect on the previous weeks work and what I needed to get done in order to have something to show for the mid-point presentation.

7.2.4 January

For me the month of January was broken up into two halves I had to sit three exams as part of the course, so the first half of January was all about concentrating on those exams. The second half I was devoting towards my software project. After the midpoint presentation, I wasn't happy with the layout of my data and didn't know how I could go forward with the project using the data I had, it wasn't thorough enough. I had each team's statistics over the course of the year however I wanted more, I wanted the stats for each game, i.e. possession, territory, metres gained, number of turnovers and so on. Using my data source and Python scripts I wouldn't be able to scrape information. The website I used for the data up to this point also had a section containing team statistics which was what I needed however it wasn't working. Attempts were made to contact them to try and solve this bug however they replied and said they couldn't fix it and it has been removed altogether for the time being. I would have to find a new data source and might have to manually input the data. Below is the email I received back from the company.

Sportradar Support
to me
Hello Conor,
The Team Statistics is not yet supported in the Rugby API. It has now been removed from our documentation. Let us know should you have any questions.

On the app "Ultimate Rugby" I found all the data I needed but knew it wouldn't be possible to scrape the data and would involve manually inputting the data. For the current year there were over 275 rows with 17 different variables, it would take quite a while. I eventually finished and had an excel file for all fourteen teams in the Pro 14 all the stats from every match they've played that year. The variables were in the columns while the teams they

were playing against were in the rows. Now I would be at least able to apply classification algorithms to my data.

Michael was happy with the layout of the data however he expressed his concerns with the lack of data. I recommended that it would be possible to add last years data and he thought that was a good idea. He also suggested I investigate generating synthetic data which uses KNN to create similar data to what you already have so I planned on looking into it when I got time. After I added all the previous years data he then recommended I bring it all into one file.

By the end of the month I was happy with the progress I made and the significant improvement with my data. I knew in order for my project to be successful I must meet with Michael as often as possible so that he could make suggestions for my project and I could try and include as many of them as possible.

7.2.5 February

There are two parts to developing a model to predict rugby games, the first would be to try and do so using statistical methods and then see how they compare with classification algorithms. The statistical methods being considered are time series analysis, multiple linear regression and principal component analysis. At the start of this month I was focusing on developing these methods as I knew later in the month I would be busy working on an assignment I had due for Data and Web Mining. The three methods were implemented in RStudio.

Multiple linear regression was done first as I had just done in this topic in my advanced business analysis module. The variables used were tries, penalties, kicks from hand and Venue. R calculated the coefficients for us and all I had to do was multiply them with the data from the match. The results were recorded.

The type of time series analysis conducted was exponential smoothing. This was implemented using the holtwinters function. The weighted value entered was 4 and the results were plotted and recorded.

The principal component analysis was done using the psych and nFactors library. The first step of this was seeing if there were high or low correlations between all the variables and then the PCA analysis was carried out where it was able to group 16 factors into 11.

7.2.6 March

The aim for this month is to start implementing machine learning algorithms into the project. This is the part in which the bulk of the work for the project lies so I want to do as much as possible an early stage, so I won't be stressing out, come the end of April. We are also starting to learn how to use them in our Data and Web Mining class so as we cover each one I'll attempt to implement it into the project if it's relevant and if I have the time to do.

The first algorithm that was used in the project was K Nearest Neighbour. To help me incorporate this into my project I used the notes from the Data and Web Mining class. For this algorithm I set the K value originally as 21 which is not too high or too low. After training the model it was then augmented with the test dataset to make predictions. I checked the results of the algorithm using the cross-table function. I continued testing the model using a different K value every time to figure out which K value was the most optimal and that turned out to be 17.

The next machine learning algorithm which was introduced to the project was Naïve Bayes. Again, we had just covered it in class, so I would be able to use the notes to help me out with it. This model was trained and then used for predictions against the test dataset and its accuracy turned out to be 74%, behind KNN which was 78%.

Important part after running these algorithms is to record the results and document it all as it's fresh in the head and then we aren't left with having to do too much of the write up at the very end.