CITY UNIVERSITY LONDON

The prediction of football results using Naive Bayes Classification and Time Series Analysis

Project Supervisor: Dr. Peter W. H. Smith

Prepared by: Daniel Grewal, abmj097

28 April 2014

Academic Year: 2013-2014

ACKNOWLEDGEMENTS

I would like to thank Dr. Peter W. H. Smith for his valued input and guidance throughout this project. I would also like to thank City University London and its teaching staff, without who, I would not have the skills and knowledge to complete this project.

ABSTRACT

Predicting football results, or any sports results for that matter, has fascinated academics, sports enthusiasts and professional betting consultants for decades. Many different approaches have been made in an attempt to create successful forecasting models which give the investor a better-than-average return on a consistent basis.

This project is focuses on using well-known statistical methods to accurately predict football matches over an entire season for a Premier League football team. The challenge of the project is that the team I have opted to observe, plays their football in what is considered to be the most difficult league in the world, which adds to the challenge of being able to successfully predict football match results.

This report focuses on my attempt to create various models to accurately predict football matches at a better-than-random rate (50%). I report my findings, as well as the application that I created and the algorithms I developed to successfully achieve the project objectives.

1. INTRODUCTION

The main aim and concept of the project, is to build an application which allows users to predict the results of an English Premier League football team over the course of the 2013-2014 Premier League Season. The predictions will be made using the Naive Bayes classification model and a time series analysis model.
Users will be able analyse and compare the predictions against actual results, as well as compare the accuracy of the models presented to the user. The application will also allow users to view the different variables which have been considered when building the models and determine which, if any, are an important factor when determining the result of football matches.
Users will also be able to view the form and "head-to-head" comparisons of each current Premier League football team against Arsenal Football Club during the managerial tenure of Arsene Wenger.
This report details the steps taken to build the application from the initial idea.

1.1 Problem to be Solved

The problem to be solved is to give users an accurate prediction system when determining the outcome of football matches over the course of the season. The result of a football match is determined by many factors, some which are not measurable (such as luck). Therefore, by building multiple prediction models, users will be able to determine which model is most suitable for each match to be predicted.
A project of this magnitude poses many challenges, some of which have been outlined below.

- **Mathematical programming:** The nature of my project requires extensive mathematical programming. The Java programming language is one I am familiar with however I am not sure if it is the best choice for this type of project. I will discuss the alternative choices with my Project Supervisor during our first meeting.

- **Time series analysis:** As a Software Engineering BSc. (Hon) student, I have not been exposed to time series analysis. I will have to research extensively in this area to understand which time series model will be best suited when predicting the result of a football match.
- **Naive Bayes Classification:** I have come across the classification method whilst researching Data Mining in my spare time. I have never applied it and will use the "Introduction to Data Mining" module during my first semester of my final year to learn more about the method. I will also have to spend more time researching how I can apply a binary classification method to a multi-class 'problem' whereby I will be predicting three classes, "Win", "Draw" and "Lose".
- **Extensive research and data collecting:** As I will be concentrating on Arsenal F.C. during the managerial tenure of Arsene Wenger, it will require extensive research. I will have to choose reliable sources to gather the data which will stem from the 1996-1997 Premier League season up until the end of the 2013-2014 Premier League season.

1.2 Project Motivation

The project idea was thought up by my Project Supervisor when I approached him with a similar idea. My initial idea was to use statistical analysis to predict the performance of football players over the course of a football season.

I spent my placement year working at a pharmaceutical research facility for GlaxoSmithKline, where I worked alongside colleagues who worked as Data Scientists. The role of a Data Scientist was something which I had never previously come across before. After learning about the role of a Data Scientist, I discovered a real sense of excitement in being able to solve problems and predict future events using a combination of statistical and technological methods. This made me curious and I began thinking about how I could use my excitement and curiosity to solve a problem for my dissertation.

Although I have never applied data mining methods, time series analysis or any mathematical or statistical programming, I wanted a project which would challenge my ability to interpret data, make use of the data in ways which could benefit other people by producing positive results as

well as extend my programming knowledge. I wanted a project which would test whether or not a career as a Data Scientist is one which I could pursue.

1.3 Project Scope & Beneficiaries

The scope of the project will very much depend on which programming language I choose to build the application with. Currently, Java is my preferred option as it is the language I am most familiar with however I will evaluate all options before making a final decision. If I do choose to use Java, I will design my application so that it an be accessed by users using the Windows platform as well as Android tablet devices. This means that a lot of time will have to be spent on designing the functionality of the application.

Should I decide to use a different programming language, the scope of my project will change depending on the software that I use. This has the potential to narrow the number of users the application will be available to.

In terms of a target audience, the application is not limited to any specific user. Anyone who has an interest in using prediction models and data mining techniques to solve real-world problems will most definitely be interested in using this application. However, should the application be built to an acceptable standard during this project, a list of project beneficiaries can be found below.

- **Academia/Research:** Any persons who study/research the use of data mining techniques and prediction models to solve real-world problems.
- **Professional Betting Consultant:** Any persons who work in the betting trade professionally and may want to use or revise the models in the application to compliment their work.
- **Bookmaker:** Any professional and registered company who are legally allowed to make bets with betting trade customer. A bookmaker may find the application useful when setting odds for future football matches.
- **Betting Trade Customer:** Any persons who have an active interest in placing bets on football matches.

The application will include a user guide which states that the application should be used at the users' risk as predicted results are not guaranteed. The source code for the application and the accompanying user guide can be found in Appendices D and F.

1.4 Project Objectives

The project objectives focused on two criteria, *Design & Build* and *Research*.

1.0 Design and Build Objective: This project shall result in the creation of an application to automatically predict the outcome of a football match.

Sub-Objective	Test
1.1 Design and build an Android application using the Java programing language and Android SDK (Software Developers Kit).	Start the application and test whether its functions work correctly.
1.2 The application shall be suited for mobility.	Use the application on an Android tablet and ensure all functions work correctly.
1.3 The application shall be user friendly and intuitive.	Select 5 people at random to use the application and record their navigation and accessibility of the GUI through a short survey.
1.4 The application shall make automatic predictions for the user.	Use the application prediction functionality to ensure predictions are being made based on the chosen theorem.
1.5 The application shall perform reliably.	Use the application for 25 hours and calculate the MTBF (Mean Time Between Failure) and decide whether or not it is an acceptable time.

2.0 Research Objective: How can we effectively predict results for a football team?

Sub-Objective	Test
2.1 What variables are important when predicting the outcome of a football match?	Identify if any common patterns appear during the research and analysis phase of researching Arsenal FC results.
2.2 Research and choose the best prediction theorem and compare advantages and disadvantages of theorems considered.	Compare advantages and disadvantages of various theorems in a detailed document and outline in conclusion why chosen theorem was chosen with supporting evidence.
2.3 Will the prediction theory predict the correct score line at a 50% accuracy rate or better?	Compare the predicted score line against the actual score line for 60 random matches.
2.4 Will the prediction theory predict a win, draw or loss at a 75% accuracy rate or better, even if the score line isn't correct?	Compare the final outcome of matches against the actual outcome of the matches. Do this for 60 random matches.
2.5 Will the prediction theory correctly predict Arsenal FC final league position at a 60% accuracy rate or better, based on the predicted results?	Compare the predicted final league standings of Arsenal FC for previous seasons using the prediction theory against the actual final league standings of Arsenal FC. Compare the previous English Premier League (EPL) seasons dating back to 30/09/96 (current manager start date).

1.5 Project Assumptions

I am assuming that the project deadline will allow all 38 Premier League football matches (2013-2014) for Arsenal F.C. to be predicted and actual results recorded for this project. I also assume that key information I gather from sources is consistent and reliable. The application will not use more than 20% of the users' CPU (Central Processing Unit) power when running. The application will be available on the Windows platform and minimum system requirements will be sufficient to run the application smoothly.

1.6 Project Outline

During the project, I decided to adopt an Incremental working method. However as I ventured into the second half of my project, I started to deviate away from the standard methodology as my requirements on how the features of the application should work, eventually changed. See Chapter **4.1** for more details.

After discussions with my Project Supervisor, I decided to use the Matlab software to build my application. This alters the original sub-objectives (**1.1** and **1.2**) which was to use the Java programming language and build an application for an Android device. Matlab is specifically useful for mathematical, scientific and engineering programming projects which is much more suited to a project of this nature. This effected the scope of my project, but not the quality.

My original work plan did not take into account the January exam period for which I was not able to focus on the project at all. This meant I had to update my work plan to ensure the deadline for the project could still be met. Christian W. Dawson (2000:104) states that 'All projects have five elements that require managing to some extent as the project progresses: *time, cost, quality, scope* and *resources*.' Cost is an element I had little concern or control over during my academic project, scope and quality were the two elements that were key to completing the project to a suitable and high standard.

According to objective **2.3**, the application should predict the score-line of each football match at an accuracy of 50% or better. I decided during the project, that score-line predictions would effect my ability to deliver the project on the agreed deadline. Initial research and data gathering suggested that for me to deliver on this objective, I would need an extension of approximately two weeks, which is not plausible in this case. Therefore objective **2.3** was not considered during the final project as it would have had negative effects on *time* and *quality*.

1.7 Report Breakdown Structure

A breakdown of the final project report.

1.7.1 Output Summary

A two page summary of the outputs produced from the project. Each output includes an overview description; the output type e.g. software code, evaluation report; its recipient or end user and how the output will be used by its recipient.

1.7.2 Literature Review

A detailed summary of the literature I read which enabled me to do the project as reported in the rest of the Project Report Document.

1.7.3 Method

An objective description of the work undertaken during the project including the analysis, design, implementation and evaluation processes of the project described in detail.

1.7.4 Results

A presentation of **all of the** results produced during the project. This includes software code, specific algorithms, coding decisions and data analysis.

1.7.5 Conclusion and Discussion

This chapter concludes the project and re-visits the project objectives to demonstrate the success of the project by measuring the objectives that have been met.

1.7.6 Glossary

Any specialist terms are defined in this chapter.

1.7.7 Appendices

A complete collection of all data used in producing the results defined in this project.

2. OUTPUT SUMMARY

2.1 Project Definition Document

Before work on the final project could begin, a **Project Definition Document** (PDD) was created to outline the project aims and objectives and to specify what the project hoped to achieve. The PDD also includes the project plan as well as a work breakdown structure of how the project was to be conducted and completed before the deadline. The project beneficiaries and risks are also identified within this document whose intended recipient is any persons interested in understanding how my project was formed. The complete Project Definition Document can be found in Appendix A.

2.2 Project Plan

The project plan includes the revised work breakdown structure as well as accompanying Gantt chart to show the scheduling of the project and any deviations from the scheduling stated in the Project Definition Document. The plan is intended to be viewed by the Project Supervisor and any other person(s) who will be grading my project. The project plan can be found in Appendix B.

2.3 Requirements, Analysis & Design & Test Plans

The requirements specify what I wanted the application to achieve in terms of functional and non-functional states. Models and diagrams depict how these requirements were to be met. The User Acceptance Tests demonstrate that functional requirements have been met. The tests, analysis and design plans and requirements can be found in Appendix C.

2.4 Project Report Document

This is a detailed account of the project itself. It identifies the work done on the project, the methods used as well as the results of the project upon completion. It includes sections on the methods used throughout the project, the results of the project as well as a literature review which contributed to the development of the project.

2.5 Source Code

The source code can be opened with Matlab or a basic text editor such as *notepad* on the Windows platform. The source code is the code created as part of the project to build the application. The source code can be found in Appendix D.

2.6 Table of Attributes

The tables of attributes are two tables. The first of which is the potential attributes that were considered for the Naive Bayes Classifiers, the second is a table showing the actual attributes that were used. These can be found in Appendix E.

2.6 Application Package Installation Instructions & User Guide

The application was packaged as a Matlab application therefore can be used on any platform running the Matlab software. The user guide is a short document which details how the application should be installed and used for its intended purposes by the *project beneficiaries*. The application package and User Guide can be found in Appendix F.

3. LITERATURE REVIEW

3.1 Naive Bayes Classification

Although the Naive Bayes Classifier (NBC) is a relatively simple method, it competes very well with more sophisticated and complex classification methods (I. Rish 2001:41). The Naive Bayes Classifier is actually based on Bayes theorem, a probability theory named after Thomas Bayes (1701 -1767) who first introduced the theorem. Tan, Steinbach and Kumar (2006:231) describe how a NBC estimates the probability of a class by assuming that the attributes are conditionally independent (naive). To be put in more simple terms, a NBC assumes that the presence (or absence) of a particular feature of a class, is independent of the presence (or absence) of any other feature of that class. With the conditional independence assumption, Naive Bayes Classification can be defined as:

$$P(X|\mathbf{Y}) = \frac{P(X)\,P(\mathbf{Y}|X)}{P(\mathbf{Y})}$$

Tan, Steinbach and Kumar (2006:232) explain that instead of computing every class-conditional probability for every combination of **Y**, NBC works by estimating the conditional probability of $\mathbf{Y}_i$ (feature set of the class), given X. The NBC would classify a new test record (instance) by calculating the posterior probability for each class **Y**. This unrealistic assumption that each attribute is independent of the other is in fact remarkably successful and proven effective in many applications including text classification and medical diagnosis (I. Rish 2001:41).

3.2 Time Series Analysis

'A time series is a collection of observations of well-defined data items obtained through repeated measurements over time' (Australian Bureau of Statistics 2008). Observations of experimental data made over time leads to unique problems in statistical analysis and inference (Shumway & Stoffer 2014:1). One of the many benefits of *time series analysis* is the ability to make intelligent predictions from observing data at different points over a time period. A component of *time series*

analysis is *trend*. *Trend* is defined as 'long-term change in the mean level' (Chatfield 2003:12). Chris Chatfield (2003:12) also notes the difficulty in defining 'long-term change' as understanding what is considered 'long-term' is a difficult matter. Chris Chatfield (2003:12) recommends taking into account the number of observations available and making a subjective assessment on what can be considered 'long-term' according to the data. The unique statistical analysis problem concerning this project is that the observation of football match results over a long period of time should show patterns of form throughout a season. This is where the focus of *time series analysis* lies within the project. Univariate time series refers to 'single (scalar) observations recorded sequentially over equal time increments' (NIST/SEMATECH 2012). My focus lies solely on the past results of football matches for the observed football team over a time series between the start of managerial reign for Arsene Wenger at Arsenal F.C., up until the end of the 2012-2013 Premier League season. A univariate time series model will enable me to make the predictions of future football matches based off 'single (scalar) observations', in this case, past results of football matches.

3.3 AutoRegressive Process

Once I discovered a univariate time series model would be the best option for the problem I wanted to solve, I began researching different types of univariate models. The AutoRegressive (AR) process is a stochastic univariate model and the most commonly used model for time series analysis (Horvàth and Johnston, page 3).

$$Y_t = pY_{t-1} + pY_{t-2} + \ldots + pY_{t-n} + E_t$$

The equation above represents an AR(*p*) process. $\boldsymbol{Y_t}$ is the observation which is being predicted using the observations made at lags $\boldsymbol{Y_{t-1}}$ to $\boldsymbol{Y_{t-n}}$. $\boldsymbol{E_t}$ is a random variable often referred to as the error term. This error term makes up the variability that is part of the system when it moves from one time period to the next (Horvàth and Johnston, page 4). The error term $(0, \sigma^2)$ represents the difference between the estimated value and the expected value; the closer to 0, the more accurate the estimated value. Chris Chatfield (2003:59) points out that for an AR process to be implemented successfully, it is important to consider parameters and the order of the process. The

parameters ***p*** can be estimated using least squares estimation however this method will not be discussed or used for the scope of this project. Instead I have used my own parameter estimates which act as weights at the lags in the process. Chris Chatfield (2003:62) acknowledges that it is difficult to determine the order of an AR process. One way to aid the determining of the order is to use the autocorrelation function (ac.f) and partial autocorrelation function however these will not be discussed or tested during the scope of this project as I will determine my own order, discussed further in the Methods chapter.

3.4 Matlab

Matlab is well-suited to the type of project that I pursued as it is renowned for being a powerful mathematical programming tool (Hanselman & Littlefield 2001:1). Matlab is a unique programming tool in the sense that all data is stored as an array making it perfect for manipulating large data sets. Matlab also has in-built functions for supporting mathematical algorithms in various fields of science, maths and engineering in the form of *Toolboxes*. The *toolboxes* swayed my decision to use Matlab as my programming tool, knowing that I could download the *Econometrics Toolbox* which supports many time series analysis models. In the end, the graphical user interface (GUI) tools, programming features and combination of array data structures (Hanselman & Littleman 2001:1) offered by Matlab would enable me to best meet the objectives of providing an intuitive application for predicting football matches using various statistical analysis models.

3.5 Forecasting Sports Results

Dr. Ian Hale (2010) explains the different methods used in association with forecasting sports results and its rapid growth of interest. ‘Statisticians showcase their proficiency with advanced statistical techniques by modelling the intricacies of football data’ (Dr. Ian Hale 2010). He also explains the Efficient Market Hypothesis as a cornerstone of financial theory, which states that an investor should not consistently returns above the average. I hope to use this project to generate predictions which would provide a better-than-average return and contribute to the developments of sports forecasting.

4. METHOD

4.1 Software Methodology

The Incremental methodology is one which combines elements of the Waterfall model and applies them in an iterative manner (Rahul Tillo 2013). When considering the project and it's main requirement, *'The User shall be able to make predictions for Arsenal F.C. Premier League football matches',* I decided that I needed to adopt a methodology which would allow me to focus on developing the core of the project in phases, whereby I could implement regression testing and then add more functionality as I move into the next increment. This idea of this would mean the project could be conducted in phases and when one phase is complete, I could move onto the next phase. My only concern however was that this type of methodology is very rigid and any deviation would result in phases overlapping each other, hence breaking away from the standards of the methodology itself.

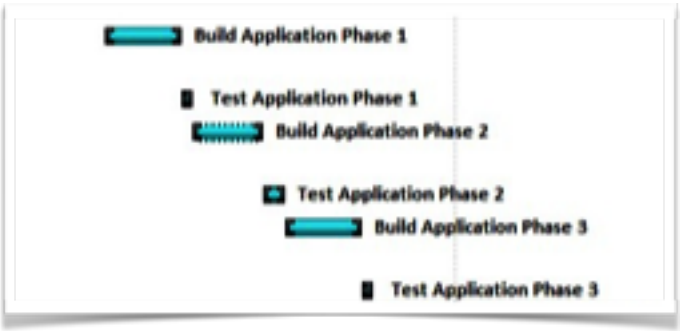

Figure 1 - Gantt Chart

As can be seen in Figure 1, the main core of the project would be conducted in three phases. After each build phase, there would be a testing phase to ensure the functionality was working as specified and that the application at that point was error free.

However I must note that the project did deviate away from the Incremental methodology during the build phases of my project. What I failed to realise was that the other modules I had chosen in my Second Semester, would require me to spend more time on them than first anticipated during what would have been my three build phases. Another reason I deviated away from the Incremental methodology is that, I hadn't anticipated the difficulty of the programming. Problems were encountered when I had to implement the Naive Bayes algorithm and AutoRegressive

process. I made a conscious decision to adopt a *rolling wave* approach to the rest of the project (Dawson 2000:55). The *rolling wave* approach is one where my project planning is performed 'on the fly'. I made decisions as to where my project was actually heading and what work I needed to perform in the following stage of my project as I completed the previous stages. This is evident in Figure 2.

Program Naive Bayes	21 days?	2/3/14 8:00 AM	3/3/14 5:00 PM
Program AutoRegression	45 days?	2/12/14 8:00 AM	4/15/14 5:00 PM
Program GUI	58 days?	2/10/14 8:00 AM	4/30/14 5:00 PM
Design Look and Feel	58 days?	2/10/14 8:00 AM	4/30/14 5:00 PM
User Acceptance Testing	2 days?	4/21/14 8:00 AM	4/22/14 5:00 PM

Figure 2 - Revised WBS

The figure shows that in the revised work plan, main elements were done in parallel to each other rather than in iterative stages as first planned. The advantages of this method was that I could make amendments to the functionality and the code, then add the functionality to the GUI as I progressed. Although many will argue that a *rolling wave* approach is risky and not an effective methodology to follow, I discovered that it complimented my ways of working very well. It allowed me to focus on building the application to a standard that I was happy with. I knew that from the beginning, it would be a challenging project as I was not an experienced programmer, I had little to no experience of prediction algorithms and time series analysis and no experience of using the Matlab software tool. Rather than spend hours of valuable time meticulously planning the project, I focused on doing the project itself. If I had decided to keep with the Incremental methodology, I think I would have failed to meet the deadline and achieve the quality of work I was aiming for.

4.2 Data Gathering

The process of gathering the required data was a long and exhaustive one. I first had to consider how much data I required to provide a good statistical analysis of Arsenal F.C. football matches. As this was a project focused on predicting future football matches, I decided to consider what are the key variables that decide the result of a football match. Figure 3 shows part of a table that I

created for considering what important events affect football match results in the Premier League.

By assessing the table, I decided upon which variables I could consider within the scope of this project. Some of the variables would have taken me too much time to analyse and research within the project scope. For instance, in the table found in Appendix E, the question, *'Is the Arsenal F.C. manager ranked better than the opposition manager?'* could quite easily have made up an entirely new hypothesis and project on its own.

Question	Variable	Importance (High/ Medium/ Low)	Reason
Was the match played at the Arsenal F.C. home venue, or the opponent venue?	Home/ Away	High	Playing in front of their own fans and in a more familiar surrounding will have a better psychological effect on players.
Did Arsenal F.C. play a domestic or European Cup game prior to the Premier League match being played	Cup Game Played (Yes/No)	Medium	Playing an important fixture prior to the league match can have a fatigue bearing effect on the players.

Figure 3 - Potential Variables

After analysing the table, I opted to choose no more than 6 variables which would make up the relative feature sets for the Naive Bayes Classifiers that I would be creating. The final variables that were used for the NBC's, can be found in Appendix. There were many websites which allowed me to gather the data that was required to satisfy the project. I used the website www.statto.com as I could make a detailed search such as, searching for Arsenal F.C. "head-to-head" results against a particular opponent. These searches would allow me to gather information such as the previous league fixtures results against opponents dating back to the start of

	1	2	3	4	5	6	7
1	'Everton'	'Home'	'2-1'	'W'	3	'Away'	'L'
2	'Middlesb...	'Away'	'4-0'	'W'	6	' '	' '
3	'Aston Vil...	'Home'	'2-0'	'W'	9	' '	' '
4	'Manches...	'Away'	'2-1'	'W'	12	' '	' '
5	'Portsmo...	'Home'	'1-1'	'D'	13	' '	' '
6	'Manches...	'Away'	'0-0'	'D'	14	'Home'	'L'
7	'Newcastl...	'Home'	'3-2'	'W'	17	' '	' '
8	'Liverpool'	'Away'	'2-1'	'W'	20	'Away'	'D'
9	'Chelsea'	'Home'	'2-1'	'W'	23	' '	' '

Figure 4 - Array of League Fixtures

Arsene Wenger's managerial reign. I could also search for Arsenal F.C. results over an entire season, these searches produced the dates of league fixtures as well as dates of domestic and European cup fixtures. I used this data to figure out when Arsenal had played a domestic or European cup fixture directly before a Premier League fixture, as this was one of the variables I had chosen to adopt for the NBC's. I then proceeded to place this data in a Matlab array, which could be viewed as a table, like the one found in Figure 4.

The figure shows a some of the results for an entire season as well as the venue of the fixture, the score, whether or not Arsenal F.C. won, drew or lost the fixture, points gained, whether or not a cup game was played prior to the league fixture and if that cup game was won, drawn or lost. The Matlab data produced in table arrays similar to the one above, would prove pivotal when it came to programming the Naive Bayes Classification algorithm and AutoRegressive process.

4.3 Naive Bayes Classification

To program the Naive Bayes Classification (NBC) algorithm, I planned to use Matlab's *Statistics Toolbox* found at www.mathworks.co.uk/products/statistics. The *Toolbox* included a function for classifying data with the NBC method. However I quickly encountered a problem which meant I had to program the NBC myself, without the aid of a Matlab *Toolbox*.

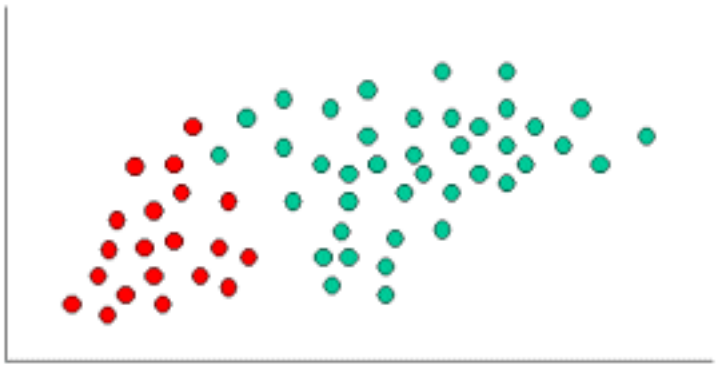

Figure 5 - www.stasoft.com

To be able to predict the result of a football match, I had to consider three class variables to predict, Win; Draw; Lose. NBC is a binary classification method. By binary, what I mean is that it is a method used to predict two class variables, either the feature set belongs to the class variable X or it belongs to the class variable Y. Before any prediction could be made, I had to program the prior probabilities.

The prior probability is information that we can already assume about the classes based on previous experiences.

For example, consider the example found on www.statsoft.com. Figure 5 shows 60 objects in total. Here, we can assume that there are two class variables. An object can either be classified as RED or as GREEN. Since there are twice as many GREEN objects as there are RED objects, we can assume that any new object, is twice as likely to be classified as GREEN, than as RED. Therefore, the prior probabilities for classes GREEN and RED can be written as follows:

$$\text{Prior probability for GREEN} = \frac{\text{Number of GREEN objects}}{\text{Number of TOTAL objects}}$$

$$\text{Prior probability for RED} = \frac{\text{Number of RED objects}}{\text{Number of TOTAL objects}}$$

Therefore I had to calculate the prior probability of Win; Draw and Lose, just as the example calculated the prior probabilities for GREEN and RED objects.

```
%P(Y):
pW = sum (SouthamptonW) / size (SouthamptonW,1); %get probability of a win
pNW = sum (1-SouthamptonW) / size (SouthamptonW,1); %get probability of other results
```

Figure 6 - Prior Probability of Win

Figure 6 shows the code I programmed to get the prior probabilities for an Arsenal F.C. win against Southampton F.C. and the other results (an Arsenal F.C. draw and a loss) suffered against Southampton F.C.. I repeated this for Arsenal F.C. draws and Arsenal F.C. losses as part of the multi-class problem discussed below. The feature set, which is made up of the attributes from the table shown in Figure 3 are programmed as values ranging from 0 to 2. For more information on the feature set, please use the source code provided in Appendix D.

```
Southampton_NBC = [0 0 1 1 1;1 0 0 1 2;0 1 1 1 1;1 1 1 1 1;1 0 1 1 2;0 0 1 1 2;
```

Figure 7 - Vector Array of Feature Sets

The multi-class problem I had encountered meant that I had to program the NBC so that it was still performing binary classification. To do this, I used the one-against-rest approach (1-r) as discussed by Tan, Steinbach and Kumar (2003:307).

```
SouthamptonW = SouthamptonY; %new array to begin first binary classification
SouthamptonW(SouthamptonW==1)=0; %change all draws to a 0
SouthamptonW(SouthamptonW==2)=1; %change all wins to a 1
```

Figure 8 - Multi-Class Programming

The class 'Win' is represented as the value 2, the class 'Draw' as value 1 and the class 'Lose' as the value 0. By following the 1-r approach, I manipulated the data dependent on the class the I wanted to predict for the test record. By looking at Figure 8 it is clear to see that I used simple mathematical programming to manipulate the class 'Draw' to the same value as class 'Lose'. This allowed me to predict the likelihood of the new test record being classified as class 'Win' against the other classes. I would repeat this step for the other classes I was trying to predict. To find out which class had the highest posterior probability of the test record being classified as, I would use a voting system to tell Matlab to display the class which had the highest *posterior probability*. For more information on how I programmed the NBC algorithm, please see Appendix D.

4.4 The AutoRegressive Process

To program the AR process, I used the *Econometrics Toolbox* found at www.mathworks.co.uk/products/econometrics.

As discussed in Chapter 3.3, I was going to use my own parameters and order to determine the process. I began by evaluating the data I had gathered and used Matlab to analyse patterns in results for Arsenal F.C. over a season by representing data in Figure 4, as a graph. I determined that on average, the previous four league fixtures had an effect on the next league fixture. I determined this by analysing points gained throughout a season so that I could look for 'dips' in form. Horvàth and

Time Lag	Parameter Value
Y	0.5
Y	0.25
Y	0.15
Y	0.09

Table 1 - Parameter Values

```
[Y YMSE] = forecast(estmdl,1,'Y0',data); %forecast 1st prediction
[YF YMSE] = forecast(estmdl2,1,'Y0',data2); %forecast 2nd prediction
```

Figure 10 - Forecasting Next Result

Johnston (page 5) state that for a AR process to be stable, the parameter *p* coefficients should be less than or equal to 1. I used a logical approach to satisfy this condition and considered that the most previous result would have a greater effect on the next result with the fourth result having the least effect. Lets again revisit the AR process equation.

$$Y_t = pY_{t-1} + pY_{t-2} + pY_{t-3} + pY_{t-4} + E_t$$

The parameter values I assigned to each time lag can be found in Table 1. If we consider the data of previous results against Norwich City in Figure, Matlab's function for the AR process is known as 'arima'. The Matlab *Help* documentation on this function states that the most recent result needed to be entered into the array, last, so that it could be processed as a Last-In-First-Out (LIFO) method used in similar data structures. Therefore, according to our parameters, the last value (which in this case represents an Arsenal F.C. win) will have the parameter *p* value of 0.5. $\mathbf{Y}_{t-2}$, which in this case, was an Arsenal loss as shown by the value 0, has the parameter *p* value of 0.25 and so on.

```
data = [2 2 2 1 0 2]'; %data
```

Figure 11 - Array of Previous Results

I then used the 'arima' function to fit the AR (4) process (the number 4 represents the order of the process), to the data. This is where Matlab would compute the error term E_t (which is explained in Chapter 3.3) so that the model was ready to predict the next outcome based on the previous results, using the data in Figure 11. The next step was to predict the next fixture against the opposition which is shown in Figure 10.

The code demonstrates that Matlab is to use the AR (4) process "estmdl" to predict 1 point into the the future, using the data in the array called "data" (the second line of code predicts the second fixture against the respective opponent). The result will output a value between 0 and 2. This result wouldn't mean much to the User so I then implemented an 'IF' statement which asks Matlab that if the value is less than 0.5, print "Arsenal will lose". If the value is greater than or equal to 0.5 but less than 1.5, print "Arsenal will draw". If the value is greater than or equal to 1.5, print "Arsenal will win". I completed these steps twice for each opposition. The reason it was performed twice

was that I was predicting the first league fixture against the opposition and then the second league fixture. To predict the second, I had to add the actual result of the first league fixture into a new array which I called “data2”. I then concatenated each predicted result into a cell array, which I would use to display in the application GUI, as both a table and a graphical representation. For more information on the source code, please use Appendix D.

4.5 The Graphical User Interface

The GUI was programmed using Matlab’s GUIDE tool. GUIDE aids the user to create basic interfaces on which to view figures such as axes or tables. Unfortunately, Matlab GUIDE was not developed for advanced GUI’s. It did not allow advanced GUI design without integrating with Java which, for this project, I would not have time to implement within the scope. I used the Matlab GUIDE tool to create all the interfaces for the application. It did speed the process up somewhat in the sense that I did not have to program manually, where I wanted objects to be placed on the screen as I could just adopt a ‘click and drag’ approach. All of the buttons, axes and tables in the application, were added using the GUIDE tool. I programmed manually how I wanted the objects to work and interact with each other, as well as the style of the application including background colours and font-sizes. For more information, please see the source code in Appendix D.

5. RESULTS

5.1 Meeting Requirements & Objectives

Before project work could begin, I created a **Project Definition Document** (Appendix A). This document stated my intentions for the project, what I wanted to achieve as well as how I planned to achieve it. As mentioned in Chapter 4.1, my work plan was revised on more than one occasion as I deviated away from the Incremental software methodology towards a 'rolling wave' approach.

Actor Type	The User…	The User…	The User…	The User…
Result Type	…shall be able to predict…	…shall be able to analyse…	…shall be able to analyse…	…shall be able to predict…
Object	…Arsenal F.C. Premier League football matches…	…statistical information about previous Arsenal F.C. league standings…	…statistical information about Arsenal F.C. opponents…	…Arsenal F.C. football matches using different statistical methods…
Qualifier	…easily	…confidently	…confidently	…reliably

Figure 12 - Functional Requirements

The objectives of the project were, in simple terms, to create an application which could predict Arsenal F.C. football matches using different statistical approaches, achieving a better than random accuracy (greater than 50%). These objectives led to the creation of functional and non-functional requirements. The full set of requirements can be found in Appendix C but Figure 12 shows the functional requirements as stated before work on the project began. The table demonstrates how I chose to write the requirements as simply as possible. I used the standard set by Alexander and Hall (2002). I did not want to over-complicate the project by creating a long set of requirements, as my experience whilst on placement at GlaxoSmithKline, taught me that the more requirements a project must satisfy, the easier it is to lose sight of the projects' reason for

existence. The next few chapters will define how I met these requirements to successfully create an application to predict the outcome of football matches with a better than random accuracy.

5.2 Research & Data Analysis

Before I could begin designing and building the application, I first had to research and consider the data that would be crucial to the development of the prediction algorithms. I'd then have to gather this data and organise it so that it could be easily analysed before being used as part of a prediction theory.

Chapter 4.2 explains the method I used to research the data needed for the project. I created two tables, one containing a list of potential data that could be used to create the prediction algorithms, the other table consists of the actual data I decided to use as part of the algorithms. I knew that in the sport of football, more often than not, the better team wins. What makes football so difficult to predict, is that it is perhaps the only sport where on any given day, a team from the lower echelons of professional football can beat a team from the top tier of professional football. A feat that we so often see in the English FA Cup competition. As time was a major factor in doing this project, I used my personal knowledge and experience to determine the data I believed was crucial to the outcome of football matches. I then considered whether or not I would have sufficient time to collect this data and analyse it as part of this project. The table from Appendix E displays the data I would eventually determine as important when considering the outcome of football matches, as well as data which is able to be gathered and analysed in a realistic time frame.

I used the website www.statto.com to gather the data necessary for this project. I manually input the data into a cell array using Matlab. The reason for this was that it would be a good chance for me to become accustomed with the Matlab programming language and tool features. There were four types of arrays that I constructed.

The first type of array was a cell array that was typically a 38x7 array, of which there were seventeen in total. A cell array is one which can contain non-numerical values as well as numerical

values. This is particularly useful when you want to display the data as table as well. A typical cell array contained information such as:

- League fixture opponent
- Venue fixture was played at
- Result of the league fixture
- Whether Arsenal F.C. won, drew or lost the fixture
- Points gained from the fixture
- Whether Arsenal F.C. played a cup game prior to the league fixture
- Whether Arsenal F.C. won, drew or lost the cup game (if one was played)

These cell arrays contained data on league fixtures played in every season under Arsene Wenger. Although in the end I decided not to use the last point as part of the prediction algorithms, I included it when creating the cell arrays as I was still undecided at that point in time. It is also worth noting that the cell array, 'Table9697' is a 30x7 array. This is because Arsene Wenger did not take managerial control of Arsenal F.C. until late September 1996. Therefore the first eight fixtures were not included in this project as I was only concerned with results under Arsene Wenger.

```
Points_0001 = [0;3;6;7;8;11;12;15;18;21;24;27;28;28;28;31;
```

Figure 13 - Points Array

The next type of array I constructed from the data I gathered from www.statto.com was a vector array, containing the points Arsenal F.C. gained throughout a league season. Again, there were seventeen of these, one for each season ranging from 1996-1997, to 2012-2013. A vector array is one which contains only numerical data and one which is of the same 'n-x-n dimensions'. These arrays were 1x38 dimension (1 row x 38 columns) arrays as shown in the figure above.

	Home/Away	Result	W/D/L
1	Away	2-1	L
2	Home	5-2	W
3	Home	5-2	W
4	Away	2-1	L
5	Away	3-3	D
6	Home	3-2	L
7	Away	2-1	L
8	Home	3-0	W
9	Away	0-0	D
10	Home	4-4	D

Figure 14 - Head-2-Head Table

The next two types of array I constructed were both cell arrays. These arrays contained information about past results against each opponent Arsenal F.C. would face in the 2013-2014 league campaign.

It is very important to note here, that not all opponents could be included when making predictions. The reason for this is that I required a minimum amount of data to fairly predict the outcome of a football match. I decided that six matches would provide a sufficient amount of data to achieve this. The reason I decided on six, is because it accounts for three seasons of football against an opponent, three games played at a home venue and three games played at an away venue. This number balances out nicely in a mathematical sense without giving an advantage to Arsenal F.C. or the opponent. There were two football teams who I could not predict football matches for using the Naive Bayesian Classifier or the AutoRegressive process. These teams were Crystal Palace and Cardiff City. For more information on the data that was collected for this project, see the source code in Appendix D.

5.3 Multi-Class Naive Bayesian Classifier Algorithm

The idea of building a prediction algorithm using the available add-ons from the Matlab tool creators, MathWorks, seemed very simple to begin with. It was only after I purchased the required add-ons, that I realised I would have to create the algorithms myself.

The problem, was that a Naive Bayesian Classifier, as mentioned in Chapter 4.3, is a binary classification method. I had to innovate how I was going to achieve my project goal of predicting football results as Matlab did not have a function class for manipulating binary classification methods for multi-class problems. I referred to Tan, Steinbach and Kumar's (2006:306) description of the 'multi-class' problem to overcome my issue of being able to use a binary classification method, to assign a new test record to one of three classes. I mentioned in Chapter 4.3, I adopted a method known as, one-against-rest (1-r).

Attribute/Class	Numerical Representation 0	Numerical Representation 1	Numerical Representation 2
Home/Away	Away	Home	
Cup Game Played	No	Yes	
More Goals Scored	No	Yes	
More Goals Conceded	Yes	No	
Last League Fixture Result	Lost	Drawn	Won
Last League Opponent Rank	Lower	Higher	
League Result	Lost	Drawn	Won

Table 2 - Model Feature Sets

To begin with, I considered the feature set I wanted to use which included 6 attributes and input this data into an array along with the class variable belonging to the feature set. This array would serve as the data that would allow me to predict the prior probabilities. I built a total of 5 Naive Bayesian Classifiers (NBCs) using a variation of the attributes. Table 2 shows the different attributes that were included for each NBC.

Model	Attributes Included in Feature Set					
Naive Bayes 1	Home /Away	Cup Game Played Prior To League Fixture	More Goals Scored	Less Goals Conceded	Last League Fixture Result	Last League Fixture Against Higher/Lower Ranked Team
Naive Bayes 2		Cup Game Played Prior To League Fixture	More Goals Scored	Less Goals Conceded	Last League Fixture Result	Last League Fixture Against Higher/Lower Ranked Team
Naive Bayes 3	Home /Away		More Goals Scored	Less Goals Conceded	Last League Fixture Result	Last League Fixture Against Higher/Lower Ranked Team
Naive Bayes 4	Home /Away	Cup Game Played Prior To League Fixture			Last League Fixture Result	Last League Fixture Against Higher/Lower Ranked Team
Naive Bayes 5	Home /Away	Cup Game Played Prior To League Fixture	More Goals Scored	Less Goals Conceded		

Table 3 - Numerical Representation of Attributes

Next, I created a new function class in Matlab to code the algorithm into. The reason I chose a function class, was that it would make it much easier to add or change the algorithms in the future. I could also simply refer to the function in any other script file or GUI file that I created where I wanted to call the algorithm. There were 17 teams that I had to code the algorithm for (excluding the two teams I couldn't predict which would have made it 19 teams), so I decided to code each team alphabetically as they appear in the league table before the first fixture is played.

I input the feature set and class variable as a vector array. Using Table 3, I manipulated the data so that it was numerical, representing values from 0 to 2. This was so that the algorithm could classify more easily, the likelihood of each class occurring given an attribute, or attributes grouped together, which is what is meant by the term, feature set. The class result is always the 'League Result', this is the class that I wanted to predict for future football matches. This is where the 1-r approach is used. There are three different class variables I wanted to predict as shown by the Table 3. To predict the outcome of a win in the next game, I would manipulate the 'Drawn' class variable (numerical representation 1) to be represented by the number 0. Then I'd change the 'Won' class variable (numerical representation 2) to be represented by the number 1. This is how 1-r solves the multi-class problem, it decomposes the problem into K binary problems (Tan, Steinbach and Kumar 2006:306). In simple terms, it manipulated the data so that the class variable I wanted to predict, is now represented by the number 1, and the other two classes are represented by the number 0, thus enabling one-against-rest. For more information on how I manipulated the class variables, see Chapter 4.3.

```
NB_Results_Array = {Aston_Villa_str;Fulham_str;Tottenham_str;Sunderland_str;
```

Figure 15 - Naive Bayes Results Array

I then broke down the algorithm so that I could work out the probability of X when it is equal to 1 and X when it is equal to 0. I then worked out the probability of X being equal to 1 when each attribute in the feature set is 1 and vice versa for 0. I carried on this process until I had all the prior

probabilities that I needed. I then introduced the feature set of the next football match (test record) to predict against the opponent and programmed it into the algorithm so that I would have the posterior probabilities of X being equal to 1 and X being equal to 0, given the feature set of the new instance. I repeated this process to predict the probability of a win, draw and a loss using the multi-class approach described earlier.

```
'Arsenal vs Aston Villa: Arsenal will Win'
'Arsenal vs Fulham: Arsenal will Win'
'Arsenal vs Tottenham: Arsenal will Win'
'Arsenal vs Sunderland: Arsenal will Win'
'Arsenal vs Stoke: Arsenal will Win'
'Arsenal vs Swansea: Arsenal will Lose'
'Arsenal vs West Brom: Arsenal will Win'
'Arsenal vs Norwich: Arsenal will Win'
'Arsenal vs Crystal Palace could not be predicted'
'Arsenal vs Liverpool: Arsenal will Draw'
'Arsenal vs Man United: Arsenal will Lose'
'Arsenal vs Southampton: Arsenal will Win'
'Arsenal vs Cardiff City could not be predicted'
```

Figure 16 - Matlab Command Output of Array

I then ended the algorithm by implementing a voting system so that it would automatically compute the variable with the highest probability. As you can see from the figure below, I assigned variable names to each probability. So for instance, “pAston_VillaXWF” is the variable containing the probability of a win for Arsenal F.C. in their next league encounter. I then assigned this to a new variable called “Win” which was placed into an array with the other possible class variables. Code is then used to print to the screen, the variable with the highest probability out of the three. The same approach was used twice for each team, as Arsenal F.C. faced each opponent twice during the season.

```
%voting system chooses the probability with highest value
Win = pAston_VillaXWF;
Draw = pAston_VillaXDF;
Lose = pAston_VillaXLF;
Aston_Villa_Probs = [Win, Draw, Lose];
Aston_Villa_Probs_name = {'Win','Draw','Lose'};
```

Figure 17 - Voting System for Multi-Class Problem

Once the algorithm has been programmed twice for all teams, I created a cell array to hold the variables representing the highest probability of each result for each league fixture. The variables were entered into the array so that Matlab would print to the screen, the results in the exact same order as fixtures were played in the 2013-2014 season. Figure 16 shows a snippet of how the information would be shown on the Command Output of Matlab.

I repeated this process for all 5 variations of the Naive Bayes Classification models that I created.

5.4 AutoRegressive (4) Process

The AutoRegressive (4) process was programmed using the Econometrics toolbox add-on found at www.mathworks.co.uk/products/econometrics. Programming the AutoRegressive (AR) process was fairly simple once I gained an understanding of how it works. I used the Matlab Help document, www.mathworks.co.uk/help/econ/arima-class.html so that I could understand how Matlab interprets the AR process which I wanted to build.

I used the AR process to predict the next result based on previous results, as explained in Chapter 4.4. To do this, I created a Matlab function, called 'AutoRegressive.m'. All Matlab script and function files have the extension, '.m'. I used the same approach as I did with the Naive Bayesian Classifier algorithms, whereby I programmed each AR process for each team twice, in alphabetical order. I began by creating a vector array of the past results for Arsenal F.C. against each opponent. The figure below shows how past results against an opponent during Arsene Wenger's reign, were coded. The number '1', represents a draw, the number '0' represents a loss and the number '2' represents a win. Those who are familiar with the sport of football my ask why a win is represented by the number '2' when in real life, a win earns 3 points. The reason being is that from a mathematical stand point, it would be much more difficult to calculate when the AR process calculates a draw or a win. The AR process works by predicting a number between '0' and '2'. If that number is greater than 1.5, I could confidently say that Arsenal F.C. are more likely to

```
data = [1;1;0;0;2;2;1;2;1;2;2;2;1;2;2;2;2;1;2;1;2;2;1;0;1;2;1;2;0;2;2;1;2;]; %d
data2 = [1;1;0;0;2;2;1;2;1;2;2;2;1;2;2;2;2;1;2;1;2;2;1;0;1;2;1;2;0;2;2;1;2;0;];
```

Figure 18 - AR Process Array

win than draw or lose. However, if a win was represented by the number '3', rather than the number '2', there would be greater bias in the results. The AR process would then predict a number between 0 and 3. Imagine if the predicted number was less than 0.5, it would be logical to assume Arsenal F.C. are more likely to lose than draw. If that number was greater than 0.5, a decision has to be made as to where the boundary between a draw and a win is. Would it be logical assume a draw is between the values of 0.5 and 2? Or 0.5 and 1.5? As you can see, a win

represented by the value of '3' poses mathematical problems. However a win represented by the value '2' is much easier to work with. For my predictions, I decided that any predicted value between 0 and 0.5 would represent a loss, values greater than or equal to 0,5, but less than 1.5 would represent a draw and values greater than or equal to 1.5 represent a win.

Figure 18 also shows 2 sets of data, which are identical except for the last value in the array. The second array, "data2" is the set of data to be used for predicting the second fixture in the league season between Arsenal F.C. and the opponent this dataset belongs too. The last value in this dataset represents the actual result of the first league fixture between Arsenal F.C. and their opponent during the season of 2013-2014. You will notice, this value is the value we are trying to predict using the "data" array.

In respect to the AR process, each value represents a time lag. Using the equation in Chapter 4.4, I wanted to predict the result at Y_t, which is the next time lag in the series, the time lag which is yet to occur. I did this by placing parameter values at previous observations (time lags) in the series, from Y_{t-1}, which was the most recent observation before Y_t, up until Y_{t-4}, which is 4 observations from the time lag, Y_t. As mentioned in Chapter 4.4, the reason why there are 4 parameters is because I decided that the previous 4 results against an opponent would have some kind of effect on the result of the next result against that opponent.

```
model = arima('AR',[0.5,0.25,0.15,0.09]); %autoregression model with known coefficient
estmdl = estimate(model,data); %fit ar model to the data specified
estmdl2 = estimate(model,data2); % fit ar model to the data specified
[Y YMSE] = forecast(estmdl,1,'Y0',data); %forecast 1st prediction
[YF YMSE] = forecast(estmdl2,1,'Y0',data2); %forecast 2nd prediction
```

Figure 19 - Matlab Code for Predicting using AR Process

The figure above shows how I created an AR process model called "model", using the 'arima' class from the *Econometrics Toolbox*. The values within the class function, represent the parameter values at each time lag, which in this case, there are 4. I then asked Matlab to 'fit' the model, to the data which I entered as shown in the figure above. When Matlab 'fits' the model to the data, it automatically estimates the error term which I described in Chapter 4.4. Once the model has been 'fitted', the AR (4) process model has its known parameter values as well as its error term, this

```
WestHamAR_str = Y;
WestHamAR_str2 = YF;

%if statement to display predicted result of 1st meeting
if WestHamAR_str < 0.5
    WestHamR = ('Arsenal will lose');

elseif (0.5 <= WestHamAR_str) && (WestHamAR_str < 1.5)
    WestHamR = ('Arsenal will draw');

elseif WestHamAR_str >= 1.5
    WestHamR = ('Arsenal will win');
```

Figure 20 - IF Statement To Determine Correct Prediction

new model is then saved into a variable called "estmdl" or "estmdl2", dependent on which fixture I was predicting. I then predicted the next result at Yt, using the forecast class. "[Y YMSE]" are two variables with which I wanted to store the prediction values. "Y" ("YF" is the variable that holds the predicted value of the second fixture to be predicted between the two teams, from here on out it will be referred to as "Y") would be the predicted value of the next result, i.e. a value between 0 and 2. "YMSE" is the mean-squared-error of "Y" . This value would represent the error term, or 'variance' of Y. The error term, as explained in Chapter 4.4, estimates how far from the predicted value, "Y", the result could fall. For example, imagine the code in Figure 19 produced a "Y" value of '1.2' (value represents a draw) and a "YMSE" value of '0.5'.

1.2 + 0.5 = 1.7, which indicates the predicted result could be a win as the value is greater than or equal to 1.5.

1.2 - 0.5 = 0.7, which indicates the predicted result could also be a draw but not a loss as the value is greater than or equal to 0.5.

Once I had the "Y" value, I then stored it into a new variable. I then used this variable within an 'IF' statement so that Matlab would print to the screen, a sting dependent on the value that was predicted in "Y". The 'IF' statement also demonstrates what values would have satisfied a loss, a draw and a win. I used the same "IF" statement for the second fixture as well.

Once all fixtures for all opponents had been predicted using the AR (4) process, I stored the predicted results into a cell array. This cell array would output strings for each fixture which

would be one of the three strings as seen in Figure 20, as well as the actual "Y" value that was predicted for each fixture.

The output of the array on the Matlab Command Output would look like Figure 21. As you can see, each fixture is predicted showing the string value associated with the "Y" value. Also, each fixture was predicted in the same order that they were played in real life. Some fixtures could not be predicted as there was not sufficient data to predict with. As it was an AR (4) process, it required at least 5 previous fixtures to be able to make predictions. The fixtures which could not be predicted, produce the string, "Arsenal vs … could not be predicted".

```
'1'     'Arsenal will win'              [1.7874]
'2'     'Arsenal will draw'             [1.3227]
'3'     'Arsenal will draw'             [0.7838]
'4'     'Arsenal will win'              [1.8094]
'5'     'Arsenal will win'              [1.6504]
'6'     'Arsenal vs Swansea cou...'     ''
'7'     'Arsenal will win'              [2.0109]
```

Figure 21 - Matlab Command Output of Predicted AR Process Results

5.5 The Application GUI

Figure 22 - Application Main Menu

I wanted to make sure that when I created a GUI which was simple to use and so that users could quickly get to the area of the application they were looking for, without getting lost or placing too many clicks. I created create analysis and design documentation (Appendix C) to model how I expected the application to behave as well as how the user could interact with the interface.

The GUI for the application was created using Matlab's GUIDE. GUIDE is an easy-to-use tool which aided me to quickly create a basic interface, where you can add axes, push buttons, tables etc. using a 'click-and-drag' method. This saved time in terms of having to program where I wanted to place objects on the GUI, as well as having to program the size of the GUI window. I did however have to program the functions of the objects myself. For example, programming a push button to go back to the main menu when it is clicked by the user. I also used the GUIDE tool to set the colours of text, push buttons and backgrounds. The exception is the home page as shown in the figure above, I instead opted to use a graphic that I downloaded from a Google search to give the application a more visual appeal. Another exception are the graphs that are used throughout the application, I programmed all graph features manually, including the colours, tick labels, x-axis and y-axis limits and the gridlines used on each graph. The reason for this, is that the GUIDE tool is very basic. For example, it allows you to plot points on a graph and change colours etc. but it doesn't control advanced features such as setting the line width of a plot line, or how many tick labels should show on the axes.

	Home/Away	Result	W/D/L	
1	Away	1-0	W	
2	Home	7-3	W	
3	Home	2-1	W	
4	Away	0-0	D	
5	Away	4-4	D	
6	Home	1-0	L	
7	Away	3-1	W	
8	Home	3-0	W	
9	Home	3-0	W	
10	Away	1-1	D	
11	Away	0-0	D	

Figure 23 - Example of GUIDE Error

GUIDE did have its limitations. As I mentioned, it was a basic tool which makes creating visuals easy for inexperienced programmers. It also had a few bugs which I noticed when creating tables to show some of my data. Figure 23 shows a table representing 'head-to-head' statistics against an opponent. You can see that the last column is completely blank. This was because when creating the table, the GUIDE tool would for some

unknown reason, add an extra column after my data had been inserted. This was something I could not control or repair therefore I decided to leave it as it was, seeing as it did not really effect the presentation of the data.

Another limitation of GUIDE is that it did not allow you to change the style or colours of column and row headings. As can be seen from the figure above, the column and row headings appear quite dull and basic. This was an annoying feature of Matlab which I had no control over. One other limitation was the control over text alignment in cells. Matlab did not allow me to align the data in table cells, which again was an annoying feature. I tried to compensate this by making sure each column's width was set so that it showed the data without too much space being left in the cells.

Each GUI file was saved with a '.fig" extension rather than the ."m" extension. The figure below shows that the 'main menu' interface of the application, is the parent of all other interfaces which I created. I envisioned that I did not want the intended user to have to navigate through more than 3 different interfaces to end up at the interface they were looking for. I achieved this by creating sub-menus which would appear when the user clicked a menu option shown in the "MainMenu" analysis class.

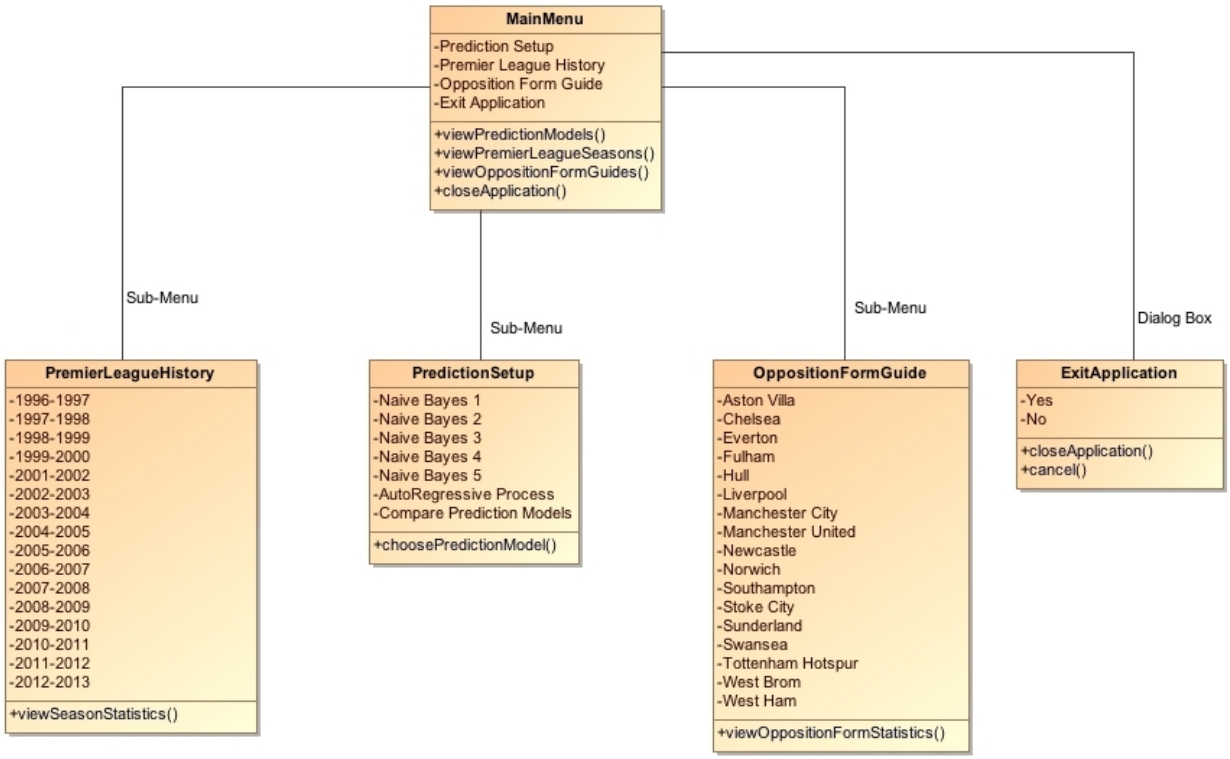

Figure 24 - Analysis Class Diagram of GUI

Figure 25 - Analysis Class Diagram of Route to Opposition Form Interface

For example, if the user wished to view statistical information on an opponent of Arsenal, they would have to activate the sub-menu options, "OppositionFormGuide" and then choose the opponent it wished to view. The figure above shows how the user would get to the interface showing information about an opponent.

5.6 Accuracy of Predictions

Upon completing the algorithmic coding, I ran each Naive Bayes model and the AutoRegressive model to test how accurately it could predict the results for Arsenal F.C. over the 2013-2014

Premier League season. I was delighted with the results of the predictions, as they surpassed my objective to predict a better than random accuracy (greater than 50%). I have produced Table 4, showing the accuracy of each model that I created.

Model	Accuracy	Comments
Naive Bayes 1	20 out of 34 = 58.82%	4 fixtures could not be predicted
Naive Bayes 2	21 out of 34 = 61.76%	4 fixtures could not be predicted
Naive Bayes 3	11 out of 34 = 32.35%	4 fixtures could not be predicted
Naive Bayes 4	14 out of 34 = 41.18%	4 fixtures could not be predicted
Naive Bayes 5	16 out of 34 = 47.07%	4 fixtures could not be predicted
AutoRegressive Process	11 out of 30 = 36.66%	8 fixtures could not be predicted

Table 4 - Accuracy of Models

In total, there were 30 out of 34 results predicted correctly using the 6 models shown above. That is an accuracy of **88.24%**. I also realised after completing my application, that I had unfairly predicted results for 4 fixtures, using the Naive Bayesian Classifier models. I stated earlier that I would only consider predicting results against opponents of which had a minimum of 6 league encounters with Arsenal up until the end of the 2012-2013 season. However the predictions against Swansea City and Hull City, although predicted correctly, did not have enough data to fairly predict with. This was a mistake which I made during the programming of the algorithms and a mistake which I missed.

I also allowed users to compare the predictions against actual results in the form of a graph. The figure below shows all the prediction models, as well as the actual results plotted on the same graph. Each win is represented by a 3 point increase, a draw is represented by a 1 point increase and a loss is represented by 0 points. Also note, that results which could not be predicted were replaced with the actual points Arsenal F.C. were awarded for that fixture. The figure also shows that the cyan-coloured line, ends at fixture 37, this is because the application was completed before the last fixture was played. For more information about the application, please see the User Guide in Appendix F.

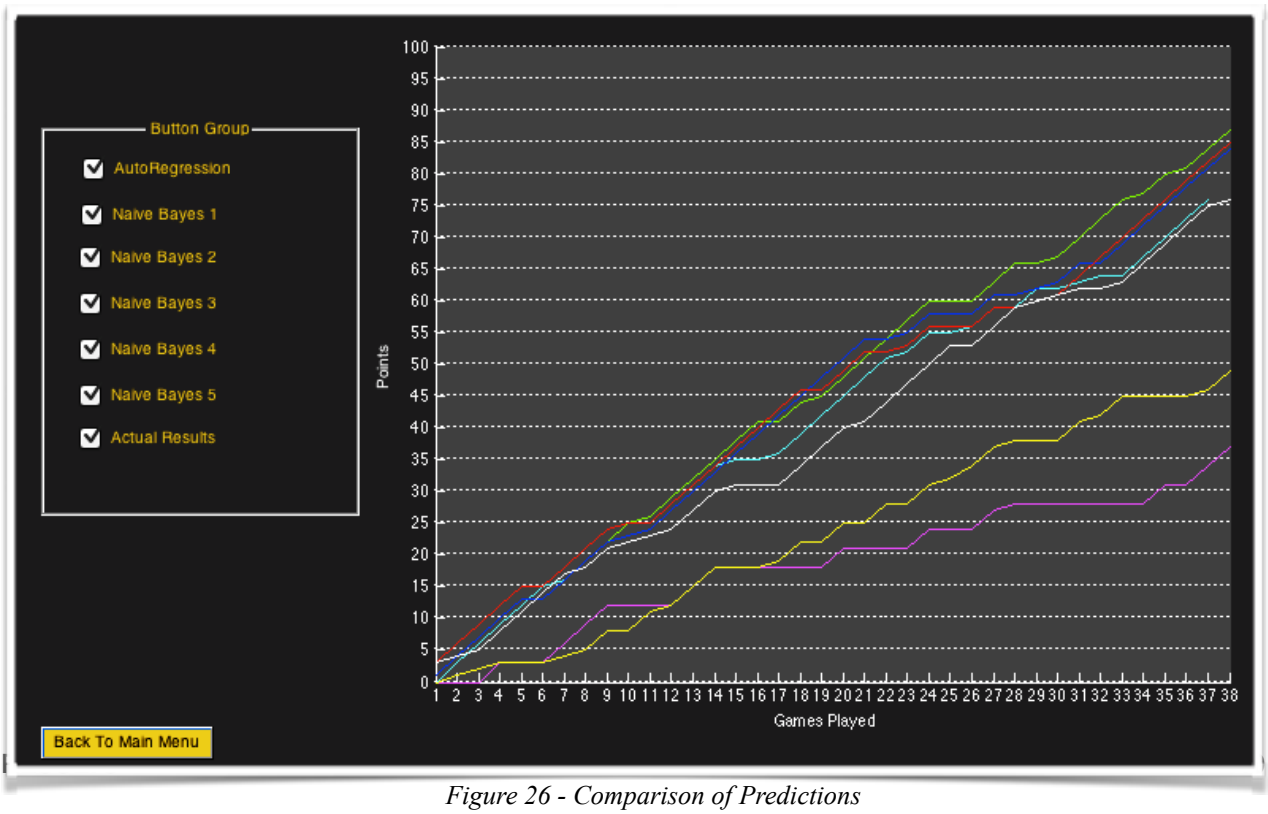

Figure 26 - Comparison of Predictions

5.7 User Acceptance Testing

Once the application was complete, I created testing documents to ensure that all functions of the application worked as expected. I then asked 5 people at random, to carry out these tests for me and report back if they were successful or not. All tests were successful and without any error. The User Acceptance Test Plans can be found in Appendix C.

5.8 Making The Arsenal F.C. Application Publicly Available

Once the application was complete, it was packaged as a Matlab application and uploaded to the MathWorks File Exchange at www.mathworks.co.uk/matlabcentral/fileexchange/46505-arsenal-f-c-premier-league-predictions. Any users with an internet connection, the Matlab software and the *Econometrics Toolbox* can download the application onto their own computer. The instructions for which, are in the User Guide in Appendix F.

I also uploaded the multi-class Naive Bayes Classifier algorithm to the Matlab File Exchange, as well as the AutoRegressive Process algorithm. Both files can be found at:

- www.mathworks.co.uk/matlabcentral/fileexchange/46512-nb-all-variables-m
- www.mathworks.co.uk/matlabcentral/fileexchange/46511-autoregressive-m

6. CONCLUSION & DISCUSSION

6.1 Project Objectives Evaluation

At the beginning of this project, I outlined objectives which I wanted to satisfy upon completion of the project. Overall, I was extremely please with what I'd been able to accomplish. I managed to create an application which managed to predict the outcome of football matches which would give better-than-average returns to a sports better. I also achieved an overall accuracy 88.24%. There were objectives that I was not able to achieve for a variety of reasons, of which I have outlined below.

Objectives 1.1 and 1.2 were not met as part of this project. Although I originally intended to create the application using Java for an Android device, on further research and on the advice of my Project Supervisor, I opted to build the application using Matlab. I discovered that Matlab was much more suited to solving problems in computational mathematic fields due to its 'matrix' programming capabilities, vast array of add-ons and the ability to work with large data sets (Hanselman and Littlefield 2001:1). Although Matlab could not help me achieve the objective of creating a mobile application, I decided to sacrifice mobility for an application of a higher quality build, which is what I felt I could I achieve with Matlab. This was not a decision that was taken lightly as I had no prior experience of using Matlab. This meant I had to learn a new programming language in a short space of time.

The main disadvantage of Matlab, is that at the time of writing, it is not an open-source tool. This greatly affected the project beneficiaries I had stated in the **Project Definition Document**. A typical betting trade customer for example, is unlikely to have used or even heard of Matlab. This changed the scope of the project slightly as it was no longer a project I could make available to the general public, only those with a license to use the Matlab tool would be able to use the application I was to build. This was now a project aimed more at a target audience of fellow students and academia, as Matlab is a tool used widely in academic institutions.

Objective 2.3 could not be met as it would have meant sacrificing the quality of the application as well as some of what I considered to be, more important objectives. This objective was focused on producing an application which could predict a correct score-line of a football match. In order for me to have achieved this, I would have had to have spent more time researching and gathering data from past football matches of Arsenal F.C.. This would primarily involve collecting the goals scored and goals conceded in every Premier League match Arsenal F.C. have played in, dating back to the 1996-1997 season. I considered my project deadline at the beginning of the project and came to the conclusion that it would have been unrealistic to try and achieve this objective within the given time frame. I instead opted to focus on predicting the correct outcome of either a win, loss or draw for Arsenal F.C..

Objective 2.5 was another objective I decided not to adopt during the project. To predict the final league position of Arsenal F.C., I would have had to collect data on all Premier League teams and their average league finishes over the course of the season. I felt that by adopting this objective, I would be taking the focus away the main objective of the project, which was to predict football matches of Arsenal F.C.. I instead decided to predict the amount of points Arsenal F.C. would gain over the course of the 2013-2014 Premier League season.

6.2 Literature Review Evaluation

The literature review was instrumental in not only helping me achieve the project objectives, but also allowing me to appreciate the developments in the area of applied probability theory and statistics. The literature I reviewed as part of the project was excellent as it allowed me to understand the concepts and identify the boundaries of what the project could potentially achieve and providing me with a better understanding of the current development in the area of professional sports betting. The literature is referenced throughout the report of this project, highlighting the importance of the understanding of key concepts I garnered from reviewing literature and the contributions it had on the project. I believe that this project investigates the potential developments to the knowledge of sports betting and that the project fits into the wider context of the literature reviewed as part of the project.

6.3 Method Evaluation

The Method section of the project report summarises how I progressed through the project. It details the problems I came across and what methods I applied to overcome them. In the Methods section of this report, the reader will be able to see the different obstacles that I was faced with along the way, including the problem of having to deviate away from the software methodology that I had planned to use to carry out this project. It also details how I created an algorithm to combat the issue of using a binary classification method to solve a multi-class problem.

6.4 Results Evaluation

The Results section identifies and measures the success of the project. It is a clear indication of whether or not the applied probability and statistics techniques used within the project, were implemented successfully. Seeing as this was a project which involved a large of degree of technical and mathematical work, I tried to write my results as clearly and concise as possible so that any interested reader could understand and interpret what had been achieved with this project, how I achieved it and if the project contributes to the field of computational statistics. The Results section outlines the important areas of focus throughout the project, from creating algorithms to building a GUI, and obviously predicting the outcome of a professional sports event, it is explicitly detailed which hopefully allows the reader to understand the importance of the results achieved.

6.5 The Future of The Project

I will continue to work on the project and expand its capabilities over the next couple of years. I would like to spend my spare time learning new data mining techniques and time series analysis methods so that I can experiment with the data that I used for this project. I am particularly interested in using unsupervised learning to predict the outcome of football matches as well as solve other problems. I think Artificial Neural Networks is one area I would like to explore and to see how I could use this type of machine learning and apply it to the project. The application itself is already available on the MathWorks File Exchange where I shall continue to monitor the

number of downloads that it receives. I will also spend time cleaning up some of the code and files that are not needed but still included in the application when it was first packaged together.

As it stands, only users with the Matlab software and relevant add-ons can actually use the application, I would like to look into the possibility of re-creating the application using the R or Python programming language and making the application available as an open-source project.

6.6 Personal Progress Evaluation

Overall, I am extremely proud of what I have been able to achieve with this project. As an undergraduate, I could have chosen a simple project to complete, one which would offer me very little challenge, which is the direction many students take each year. I decided to take the risk, challenge myself on a technical level and also challenge my mathematical intelligence. There was always the risk that if I could not achieve this project either because of technical boundaries or mathematical boundaries, that I could possibly fail my dissertation and not achieve a 2:1 degree, which is what my aim was before starting university. Before undertaking the project, I had doubts as to whether I could really achieve a project of this magnitude, or if I was aiming to achieve something which was beyond my level of capability. Completing the project has given the confidence to pursue my interests further in the areas of Statistics and Machine Learning.

Perhaps one of the most important things I have learnt from this project, is that it takes a curious mind to be able to analyse and interpret large sets of data, which seems ambiguous and meaningless to a person not familiar with the fields of statistics or data mining. The encouragement I had received from my Project Supervisor throughout the lifecycle of the project helped push me to breaking the boundaries and achieving a project which I can honestly say is my proudest academic achievement.

I hope to use the knowledge and experience gained from this project and follow up my Software Engineering BSc. (Hon) degree with a Master's Degree in the field of Statistics or Data Science.

7. GLOSSARY

Term	Definition
Array	Mathematical term for a type of list structure.
Attribute	An important term referring to important features used to predict an instance belonging to a class variable. Attribute and Variable have the same definition in context of this report.
AutoRegressive Process	A time series analysis model to forecast future events at a certain time period, dependent on the analysis of events of previous observations.
Class Variable	The Class variable is the outcome which is being predicted. In this project, there are three class variables, Win; Draw and Lose.
Naive Bayesian Classifier (NBC)	A probabilistic method based on the Bayes theorem, first introduced by Thomas Bayes.
Parameter	An assigned weighting to a time lag used in the AR process.
Posterior Probability	A term associated with Naive Bayes Classifiers for calculating the probability of a new instance belonging to a class, using the NBC algorithm.
Prior Probability	A term associated with Naive Bayes Classifiers for calculating the probability of an instance belonging to a class based on prior observations.
Test Record (Instance)	A new object containing a feature set which is being classified using NBC.
Univariate	Refers to a model which is based on one variable, as opposed to multi-variate which is based on multiple variables.
Variable	An important term which is used to describe a feature of importance belonging to a prediction model, e.g. Home/Away variable.

8. REFERENCES

Australian Bureau of Statistics (2008), '*Time Series Analysis: The Basics*', available from Internet (http://www.abs.gov.au/websitedbs/D3310114.nsf/home/Time+Series+Analysis:+The+Basics) (24 March 2014).

Chatfield, C. (2003) *The Analysis of Time Series An Introduction, 6th edition,* CRC, Florida.

Dawson, C.W. (2000) THE ESSENCE OF COMPUTING PROJECTS, Pearson Education, Essex.

Hale, Dr. I. (2010) '*The science of predicting football matches*', available from Internet (http://eandt.theiet.org/magazine/2010/08/predicting-football.cfm) (24 March 2014).

Hanselman, D. & Littlefield, B. (2001) *Mastering MATLAB 6 A Comprehensive Tutorial and Reference*, Prentice Hall, New Jersey.

Horvàth, Z. and Johnston, R. '*AR(1) TIME SERIES PROCESS Econometrics 7590*', available from Internet (http://www.math.utah.edu/~zhorvath/ar1.pdf) (24 March 2014).

NIST/SEMATECH e-Handbook of Statistical Methods, '*Univariate Time Series Model*', available from Internet (http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc44.htm) (26 March 2014).

Rish, I. (2001) '*An empirical study of the naive Bayes classifier*', *IJCAI 2001 workshop on empirical methods in artificial intelligence*, Vol. 3(22), 41-46 .

Shumway, R.H. & Stoffer, D.S. (2014), '*Time Series Analysis and Its Applications With R Examples*', available from Internet (http://www.stat.pitt.edu/stoffer/tsa3/tsa3ez.pdf) (6 May 2014).

Tan, P.N., Steinbach, M., Kumar, V. (2006) *Introduction to Data Mining*, Pearson Education, Essex.

Tillo, R. (2013), '*What is Incremental Model In Software Engineering? It's Advantages & Disadvantages*', available from Internet (http://www.technotrice.com/incremental-model-in-software-engineering/) (20 November 2013)

9. BIBLIOGRAPHY

Arlow, J. & Neustadt, I. (2005), *UML 2 AND THE UNIFIED PROCESS*, *2nd edition*, Addison-Wesley, New Jersey.

Attaway, S. (2013), *Matlab A Practical Introduction to Programming and Problem Solving, 3rd edition*, Elsevier Inc, Oxford.

Granville, V. (2014), *Developing Analytical Talent*, Wiley, Indianapolis.

Hamilton, J.D. (1994), *Time Series Analysis*, Princeton University Press, New Jersey.

Lin-Du, K & Swamy, M.N.S (2013), *Neural Networks and Statistical Learning*, Springer, New York.

Provost, F. & Fawcett, T. (2013), *Data Science for Business*, O'Reilly Media, California.

'Time Series Analysis', available from Internet (*http://www.statslab.cam.ac.uk/~rrw1/timeseries/t.pdf*) (3 May 2014).

10. APPENDICES

The Appendices have been attached as a separate document.