

PROJECT - I

Report on

AUGUR

Submitted in partial fulfillment of the requirements

of the degree of

**Bachelor of Engineering
(Electronics and Telecommunication Engineering)**

by

Naik Ameya Mahendra (18ET7030)

Gurav Chinmayee Kiran (18ET7010)

Mondal Saumen Sunil Kumar (18ET7004)

Jha Shivam Anand Mohan (18ET7043)

Supervisor

Mr. Vijay Dahake



D Y PATIL
— RAMRAO ADIK —
INSTITUTE OF
TECHNOLOGY
NAVI MUMBAI

Department of Electronics and Telecommunication Engineering
Ramrao Adik Institute of Technology,
Sector 7, Nerul , Navi Mumbai
(Affiliated to University of Mumbai)
November 13,2021



D Y PATIL
— RAMRAO ADIK —
INSTITUTE OF
TECHNOLOGY
NAVI MUMBAI

Ramrao Adik Education Society's
Ramrao Adik Institute of Technology
(Affiliated to the University of Mumbai)
Dr. D. Y. Patil Vidyanagar, Sector 7, Nerul, Navi Mumbai 400 706.

Certificate of Approval

This is to certify that, the Project - I report entitled

“AUGUR ”

is a bonafide work done by

Naik Ameya Mahendra (18ET7030)
Gurav Chinmayee Kiran (18ET7010)
Mondal Saumen Sunil Kumar (18ET7004)
Jha Shivam Anand Mohan (18ET7043)

and is submitted in the partial fulfillment of the requirement for the
degree of

Bachelor of Engineering
(Electronics and Telecommunication Engineering)
to the
University of Mumbai.



Examiner

Supervisor

Project Coordinator

Head of Department

Principal



D Y PATIL
— RAMRAO ADIK —
INSTITUTE OF
TECHNOLOGY
NAVI MUMBAI

Ramrao Adik Education Society's

Ramrao Adik Institute of Technology

(Affiliated to the University of Mumbai)

Dr. D. Y. Patil Vidyanagar, Sector 7, Nerul, Navi Mumbai 400 706.

Declaration

We wish to state that work embodied in this dissertation entitled “**AUGUR**” has been carried out under the guidance of “MR. Vijay Dahake” at Department of Electronics and Telecommunication Engineering, Ramrao Adik Institute of Technology during 2020-2021.

We declare that the work being presented forms our own contribution and has not been submitted for any other Degree or Diploma of any University/Institute. Wherever references have been made to previous works of others, it has been clearly indicated. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Naik Ameya Mahendra (18ET7030)
Gurav Chinmayee Kiran (18ET7010)
Mondal Saumen Sunil Kumar (18ET7004)
Jha Shivam Anand Mohan (18ET7043)

Acknowledgments

We take this opportunity to express my profound gratitude and deep regards to our guide **Mr. Vijay Dahake** for his/her exemplary guidance, monitoring and constant encouragement throughout the completion of this report. We are truly grateful to his/her efforts to improve our understanding towards various concepts and technical skills required in our project. The blessing, help and guidance given by him/her from time to time shall carry us a long way in the journey of life on which we are about to embark. We take this privilege to express our sincere thanks to **Dr. Mukesh D. Patil**, Principal, RAIT for providing the much necessary facilities. We are also thankful to **Dr. Chandrakant Gaikwad**, Head of Department of Electronics and Telecommunication Engineering, Project Coordinator **Mr. Vijay Dahake** and Project Co-coordinator **Mr. Manoj Dongre** , Department of **Electronics & Telecommunication Engineering**, RAIT, Nerul, Navi Mumbai for their generous support. Last but not the least we would also like to thank all those who have directly or indirectly helped us in completion of this project.

Naik Ameya Mahendra (18ET7030)

Gurav Chinmayee Kiran (18ET7010)

Mondal Saumen Sunil Kumar (18ET7004)

Jha Shivam Anand Mohan (18ET7043)

Abstract

The model uses machine learning algorithm as its learning curve and gets accustomed to the data provided . The algorithm bisects information according to its requirement on which team is better and is more accurate than than majority of the football pundits . The data feeded into the algorithm is from the football data repository and the model is trained using specific machine learning models to get the precise result. While predicting factors such as home -away advantage , player form , past records ,head to head and several other prospects are also taken into consideration. By amalgamating unified data the accurate result can be achieved . These prophecy can be used by people of various age groups and genres who are not familiar with the game and can get detailed analysis of the outcome

Contents

Abstract	v
List of Figures	viii
1 Introduction	1
1.1 Introduction to football	1
1.2 Introduction to Machine Learning for Football Results	1
1.3 Literature Overview	2
1.4 Objective	3
1.5 Challenges and Limitations	3
2 Project Description	4
2.1 Aim	4
2.2 Softwares Used	4
2.3 Steps	4
3 Dataset Description	5
3.1 Dataset Origin	5
3.2 Dataset Preprocessing	5
3.3 Dataset Features	5
4 Models Description	8
4.1 Introduction	8
4.2 Logistic Regression	8
4.3 Support Vector Machine	8
4.4 K-Nearest Neighbour	10
4.5 XG Boost	10
5 Implementation	12
5.1 Introduction	12
5.2 Approach	12
5.3 Scrapping and Cleaning	13
5.4 Exploratory Data Analysis and Visualisation	14
6 Model Fitting and Results	17
6.1 Data Preparation	17
6.2 Training and Model evaluation	17
6.3 Expected Outcomes	18
6.4 Future Aspects and Improvements	18

7 Conclusion	19
7.1 Summary:	19
7.2 Finding Data :	19
7.3 Model and Parameter Choices	19
Bibliography	20

List of Figures

1.1	Prediction Graph	2
3.1	Dataset View	7
4.1	Logistic Regression	9
4.2	Support Vector Machine	9
4.3	KNN Classifier	10
4.4	XG Boost 1	11
4.5	XG Boost 2	11
5.1	Stages Of Machine Learning	13
5.2	Importing Snippet	14
5.3	Data processing	14
5.4	Columns Rearranging	15
5.5	Aggregating the points	15
5.6	final dataset	15
5.7	Data Visualization	15
5.8	scatter Plot Matrix	16
6.1	Training and test data visualisation	17

Chapter 1

Introduction

1.1 Introduction to football

Association football more commonly known as football or soccer, a sport played between 2 sides of 11 players with a spherical ball. Football is the world's most popular sport. Outfield players move the ball with any part of the body except their hands or arms, whilst the ball is in play. Only the goalkeeper can use their hands and this is only in their penalty area. When the ball goes out of play at either side of the pitch an opposing player from the side which did not put the ball out of play can throw the ball back into play using their hands. Both feet must remain on the floor behind the throw line. Both hands must remain on the ball until it is released from behind the thrower's head. The object of football is to outscore your opponents. A goal is scored when the entire ball crosses the goal line between the goal posts. A draw is when both teams score the same amount of goals during the allotted time. A match consists of two 45 minute periods known as the first and second half. In some instances, extra time can be played, two 15 minute periods. If the side remains level after extra time a replay or penalty kicks can occur. Each team will take 5 penalties each, if scores remain level after both teams take their allotted 5 penalties then sudden death occurs. The first team to miss their penalty will lose if the opponents have scored their sudden death penalty kick. Extra time and penalties tend to only occur in tournaments and cup competitions. League games will result in a draw if both teams score the same amount of goals during the 90 minutes.

1.2 Introduction to Machine Learning for Football Results

As one of the most popular sports on the planet, football has always been followed very closely by a large number of people. In recent years, new types of data have been collected for many games in various countries, such as play-by-play data including information on each shot or pass made in a match. The collection of this data has placed Data Science on the forefront of the football industry with many possible uses and applications:

Match strategy, tactics, and analysis

- Identifying player's playing styles
- Player acquisition, player valuation, and team spending

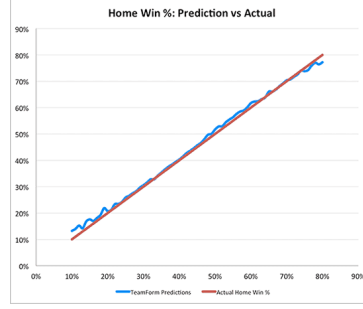


Figure 1.1: Prediction Graph

- Training regimens and focus
- Injury prediction and prevention using test results and workloads
- Performance management and prediction
- Match outcome and league table prediction
- Tournament design and scheduling
- Betting odds calculation

In particular, the betting market has grown very rapidly in the last decade, thanks to increased coverage of live football matches as well as higher accessibility to betting websites thanks to the development of mobile and tablet devices. Indeed, the foot- ball betting industry is today estimated around 350millions to 450 million pounds.

1.3 Literature Overview

Most of the work on this task has been done by gambling organizations for the benefit of odds makers. However, because our data source is public, several other groups have taken to predicting games as well. One example of the many that we examined comes from a CS229 Final Project from autumn 2013 by (Timmeraju et al. 2013). Their work focused on building a highly accurate system to be trained with one season and tested with one season of data. Because they were able to use features such as corner kicks and shots in previous games, as well as because their parameters were affected by such small datasets, their accuracies rose to 60% with an RBF-SVM. We chose to focus on a much larger training set with the focus on building a more broadly applicable classifier for the EPL. Betting in sports is a global business with lots of billion dollars invested in it, popularly the betting market of the United Kingdom which is formatted by the fixed odds that means all odds are determined by the bookmakers a number of days before every match to be played. They do not update the odds by the number of bets or the player status in this case they can make good predictions of the matches. The authors in (Bunker, R. P., and Thabtah, F., 2017) creates a model which records a team’s attacking and defensive abilities. On the matches of the 2013/2014 and 2014/2015 English Premier League seasons, their model outperformed the model by Dixon and Coles (1997) based on number of predicted goals (Dixon, M. J., and Coles, S. G., 1997). To predict outcomes of the 2002 FIFA World Cup, the authors in (O’Donoghue, P. G., et al, 2004) used different methods consisting of

probabilistic neural networks, linear and logistic regression, bookmakers' odds, computer simulations, and expert forecasts. There also have been investigations on rating systems to predict the result of football games. Which the rating system in (Elo, A. E1978) is the best known method that was proposed by Arpad Elo on prediction of chess games and later deployed to football (Hvattum, L. M., and Arntzen, H., 2010). Multilayer perceptron (MLP) with back-propagation learning rule is deployed in (Huang, K. Y., and Chen, K. J., 2011) to predict the winning rates of two teams according to their official statistical data of 2006 World Cup Football Game at the earlier stages. The training samples are used from three classes: win, draw, and loss. At the new stage, new training samples are selected from the previous stages and are added to the training samples, then they construct the neural network. In this model they finally achieved the accuracy of 75% by excluding the games which finished as a draw.

1.4 Objective

This project aims to extend the state of the art by combining two popular and modern prediction methods, namely an expected goals model as well as attacking and defensive team ratings. This has become possible thanks to the large amount of data that is now being recorded in football matches. Different Machine Learning models will be tested and different model designs and hypotheses will be explored in order to maximise the predictive performance of the model. In order to generate predictions. There are some objectives that we need to fulfill: Firstly, we need to find good-quality data and sanitize it to be used in our models. In order to do so, we will need to find suitable data sources. This will allow us to have access to a high number of various statistics to use, compared to most of the past research that has been done on the subject where only the final result of each match is taken into account. The main approach we will take is to build a model for expected goal statistics in order to better understand a team's performance and thus to generate better predictions for the future. To build this model, we will test different Machine Learning techniques and algorithms in order to obtain the best possible performance. We will be able to use data for shots taken and goals scored such as the location on the pitch or the angle to goal to estimate how many goals a team would have been expected to score during the game, and reduce the impact of luck on the final match result.

1.5 Challenges and Limitations

The data set contains the English premier league matches results from season 2013/2014 till 2018/2019 [<http://www.football-data.co.uk/englandm.php>] as public data. Dataset in CSV format containing 64 columns (features) and 380 rows (matches) for each season. Here we use the public data so the accuracy will be not as much as the odds and betting houses. And the data for all teams which are relegated from the English premier league to championship would not be on hand therefore it's considered to be predicted from the available data and the data which is not on hand there would be two zeroes it means the teams have made a draw.

Chapter 2

Project Description

2.1 Aim

The **aim** of our Project is to take into consideration all the different features of a team and implement various machine learning models to predict the target, which in our case will be the FTR(Full time result) which will predict if the winning team is the Home team, the Away team or will it will result in a draw.

2.2 Softwares Used

- Jupyternotebook :

Project Jupyter is a project and community whose goal is to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.

- Microsoft Excel :

Microsoft Excel is a spreadsheet developed by Microsoft for Windows, macOS, Android and iOS. It features calculation, graphing tools, pivot tables, and a macro programming language called Visual Basic for Applications.

- Google Colaboratory :

Colaboratory, or "Colab" for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

2.3 Steps

- We will clean our dataset.
- Split it into training and testing data (12 features & 1 target (winning team (Home/Away/Draw))
- Train 4 different classifiers on the data - Logistic Regression, Support Vector Machine, K-Nearest Neighbour and XGBoost.
- Use the best Classifier to predict who will win given an away team and a home team

Chapter 3

Dataset Description

This chapter describes the data that is being used in the project.

3.1 Dataset Origin

We have obtained a dataset from [Football-Data.co.uk](http://football-data.co.uk/data.php) (<http://football-data.co.uk/data.php>). In addition to the Livescore, Tables and Statistics service Football-Data continues to provide the football punter with computer-ready football results, match statistics and betting odds data for use with spreadsheet applications, to help with the development and analysis of football betting systems. In doing so Football-Data takes the time out of recompiling pages and pages of results data and past betting odds found on a number of football results and odds comparison websites.

In this we will be using the data of the English Premier League.

It includes the following:

- Datasets of English Premier League seasons of around 10 years (2005-06 to 2015-16).
- Data of 300+ matches per season (each dataset carries data for one season).
- Data features ranging from Dates and referees of matches to fouls and red/yellow cards of the teams.

3.2 Dataset Preprocessing

- Our objective here was to collect all the data from the matches from 2005-06 to 2015-16 and compile them into a single final dataset. This will be done by scraping and cleaning.
- Out of all the features in the dataset, we will preprocess the data to use only the useful features. Then we will rearrange the data.
- Finally the data is ready to work on.

3.3 Dataset Features

- Div = League Division

- Date = Match Date (dd/mm/yy)
- HomeTeam = Home Team
- AwayTeam = Away Team
- FTHG = Full Time Home Team Goals
- FTAG = Full Time Away Team Goals
- FTR = Full Time Result (H=Home Win, D=Draw, A=Away Win)
- HTHG = Half Time Home Team Goals
- HTAG = Half Time Away Team Goals
- HTR = Half Time Result (H=Home Win, D=Draw, A=Away Win)
- HS = Home Team Shots
- AS = Away Team Shots
- HST = Home Team Shots on Target
- AST = Away Team Shots on Target
- HHW = Home Team Hit Woodwork
- AHW = Away Team Hit Woodwork
- HC = Home Team Corners
- AC = Away Team Corners
- HF = Home Team Fouls Committed
- AF = Away Team Fouls Committed
- HO = Home Team Offsides
- AO = Away Team Offsides
- HY = Home Team Yellow Cards
- AY = Away Team Yellow Cards
- HR = Home Team Red Cards
- AR = Away Team Red Cards

Div	Date	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG	HTR	Referee	HS	AS	HST	AST	HF	AF	HC	AC	HY	AY	HR	AR	B3
E0	#####	Aston Villa	West Ham	3		0 H		2	0 H	M Dean	23	12	11	2	15	15	16	7	1	2	0	0	
E0	#####	Blackburn	Everton	1		0 H		1	0 H	P Dowd	7	17	2	12	19	14	1	3	2	1	0	0	
E0	#####	Bolton	Fulham	0		0 D		0	0 D	S Attwell	13	12	9	7	12	13	4	8	1	3	0	0	
E0	#####	Chelsea	West Bron	6		0 H		2	0 H	M Clattenl	18	10	13	4	10	10	3	1	1	0	0	0	
E0	#####	Sunderland	Birmingham	2		2 D		1	0 H	A Taylor	6	13	2	7	13	10	3	6	3	3	1	0	
E0	#####	Tottenham	Man City	0		0 D		0	0 D	A Marriner	22	11	18	7	13	16	10	3	0	2	0	0	
E0	#####	Wigan	Blackpool	0		4 A		0	3 A	M Halsey	11	9	6	7	8	11	6	4	1	1	0	0	
E0	#####	Wolves	Stoke	2		1 H		2	0 H	L Probert	13	10	7	6	17	13	5	5	0	2	0	0	
E0	#####	Liverpool	Arsenal	1		1 D		0	0 D	M Atkinson	7	14	4	7	13	15	9	11	1	3	1	1	
E0	#####	Man Unite	Newcastle	3		0 H		2	0 H	C Foy	18	7	10	3	9	5	5	3	2	2	0	0	
E0	#####	Arsenal	Blackpool	6		0 H		3	0 H	M Jones	26	3	16	1	9	3	8	2	0	0	0	1	
E0	#####	Birmingham	Blackburn	2		1 H		0	0 D	M Oliver	10	7	7	2	13	14	4	12	2	3	0	0	
E0	#####	Everton	Wolves	1		1 D		1	0 H	L Mason	14	9	6	4	18	15	3	3	1	2	0	0	
E0	#####	Stoke	Tottenham	1		2 A		1	2 A	C Foy	15	7	10	6	15	4	6	2	3	1	0	0	
E0	#####	West Bron	Sunderland	1		0 H		0	0 D	K Friend	11	10	5	4	16	12	4	6	2	2	0	0	
E0	#####	West Ham	Bolton	1		3 A		0	0 D	A Marriner	17	12	10	8	11	18	8	5	2	4	0	0	
E0	#####	Wigan	Chelsea	0		6 A		0	1 A	M Dean	17	10	12	8	12	7	0	0	1	2	0	0	
E0	#####	Fulham	Man Unite	2		2 D		0	1 A	P Walton	16	14	8	7	7	6	5	11	2	2	0	0	
E0	#####	Newcastle	Aston Villa	6		0 H		3	0 H	M Atkinson	20	3	14	1	15	12	10	2	3	1	0	0	
E0	#####	Man City	Liverpool	3		0 H		1	0 H	P Dowd	8	13	3	7	13	12	5	4	1	1	0	0	
E0	#####	Blackburn	Arsenal	1		2 A		1	1 D	C Foy	15	14	7	9	9	7	4	8	1	0	0	0	
E0	#####	Blackpool	Fulham	2		2 D		0	1 A	M Oliver	14	12	5	7	13	16	4	5	1	0	0	0	
E0	#####	Chelsea	Stoke	2		0 H		1	0 H	M Atkinson	15	9	6	3	5	10	6	1	0	2	0	0	
E0	#####	Man Unite	West Ham	3		0 H		1	0 H	M Clattenl	18	7	11	2	6	11	9	3	1	2	0	0	
E0	#####	Tottenham	Wigan	0		1 A		0	0 D	P Dowd	12	13	7	5	10	11	16	7	3	2	0	0	
E0	#####	Wolves	Newcastle	1		1 D		1	0 H	S Attwell	5	11	2	6	20	17	5	3	7	4	0	0	
E0	#####	Aston Villa	Everton	1		0 H		1	0 H	M Jones	12	15	6	9	19	10	4	16	4	0	0	0	
E0	#####	Bolton	Birmingham	2		2 D		0	1 A	K Friend	11	9	8	5	19	17	6	1	2	3	1	0	
E0	#####	Liverpool	West Bron	1		0 H		0	0 D	L Probert	11	14	7	5	9	8	7	6	0	0	0	1	
E0	#####	Sunderland	Man City	1		0 H		0	0 D	M Dean	12	9	4	7	5	9	3	7	0	3	0	0	
E0	#####	Arsenal	Bolton	4		1 H		1	1 D	S Attwell	22	7	14	6	11	11	9	6	2	2	0	1	

Figure 3.1: Dataset View

Chapter 4

Models Description

4.1 Introduction

Machine learning has a variety of models to be used for prediction and classification problems. Out of the following available models we have selected are :

- Logistic Regression
- Support Vector Machine,
- XG boost
- KNN

4.2 Logistic Regression

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

4.3 Support Vector Machine

- Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

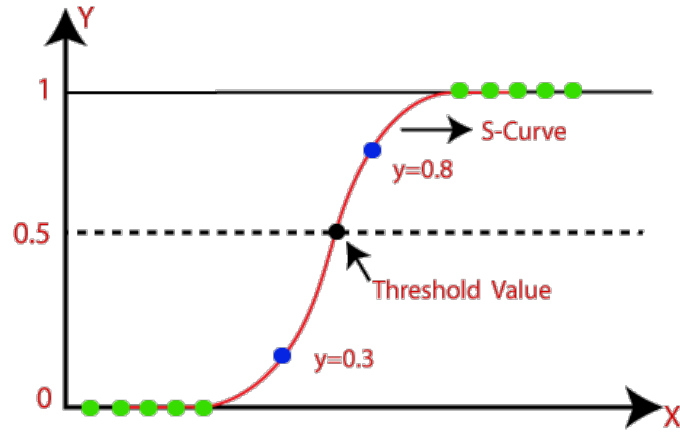


Figure 4.1: Logistic Regression

- The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane
- SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

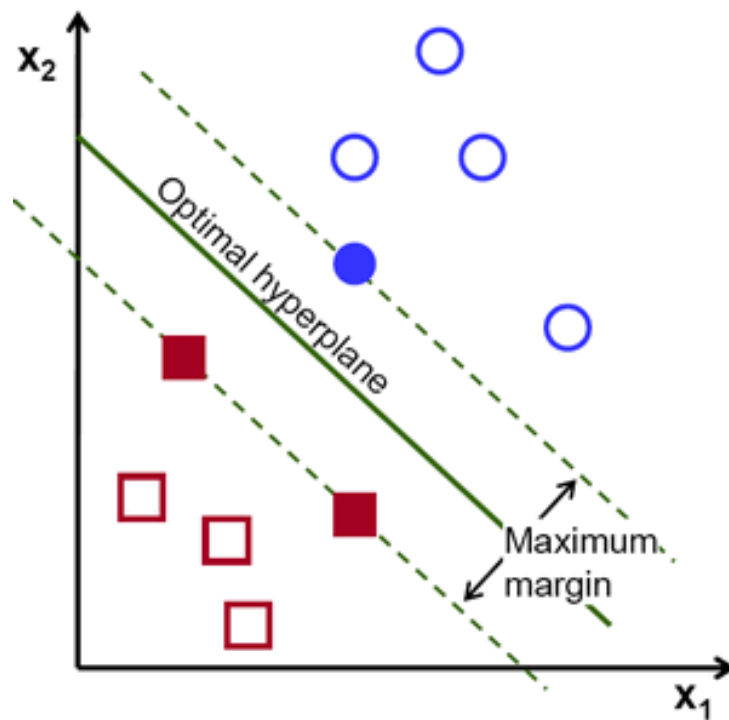


Figure 4.2: Support Vector Machine

4.4 K-Nearest Neighbour

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- The K-NN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suited category by using K- NN algorithm.
- The K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
- It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- The KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.



Figure 4.3: KNN Classifier

4.5 XG Boost

XGBoost stands for “Extreme Gradient Boosting”. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements Machine Learning algorithms under the Gradient Boosting framework. It provides a parallel tree boosting to solve many data science problems in a fast and accurate way.

XGBoost explained in 2 pics (1/2)

Classification And Regression Tree (CART)

Decision tree is about learning a set of rules:

if($X_1 \leq t_1$) & if($X_2 \leq t_2$) then R_1

if($X_1 \leq t_1$) & if($X_2 > t_2$) then R_2

...

Advantages:

- Interpretable
- Robust
- Non linear link

Drawbacks:

- Weak Learner ☹
- High variance

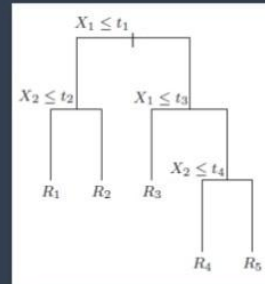
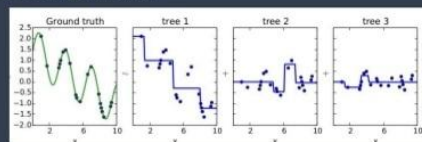


Figure 4.4: XG Boost 1

XGBoost explained in 2 pics (2/2)

Gradient boosting on CART



- One more tree = loss mean decreases = more data explained
- Each tree captures some parts of the model
- Original data points in tree 1 are replaced by the loss points for tree 2 and 3

Figure 4.5: XG Boost 2

Chapter 5

Implementation

5.1 Introduction

In this section we will display the actual implementation of the project .

5.2 Approach

- We begin by gathering our data from Football-Data.co.uk which consists of datasets of around 10 years (2005-06 to 2015-16) with one season per dataset
- After getting our datasets together we scrap and clean using various libraries like numpy, pandas etc.
- At the end of cleaning the datasets,we compile them together into a final dataset,which is ready for our utilization.
- Now we begin our data exploration and model making based on the data.
- Here we first try to display the data and briefly visualise it to make a rough estimate of the features.
- By using numpy we calculate important results such as win rate for the home team,total number of matches,number of features which will be used in the end etc.
- For a much more detailed visualisation of the data we will be using a scatter plot,available in the matplotlib library.This helps to analyse the following features:
 1. HTGD - Home team goal difference
 2. ATGD - away team goal difference
 3. HTP - Home team points
 4. ATP - Away team points
 5. DiffFormPts Diff in points
 6. DiffLP - Difference in last year's prediction
- Now we will make extensive use of scikit-learn to seperate the feature set and target variables and then separate the dataset into training and test sets.

- Now we begin training and evaluating our model.
- By fitting a classifier to the training data, we will make predictions based on the `f1_score`.
- We fit the above models and evaluate the `f1score` for each model and in the end we tune the parameters using `XGBoost`.
- We also perform some miscellaneous tasks using `GridSearchCV` and `make_scorer` to compile the results of our model.

30

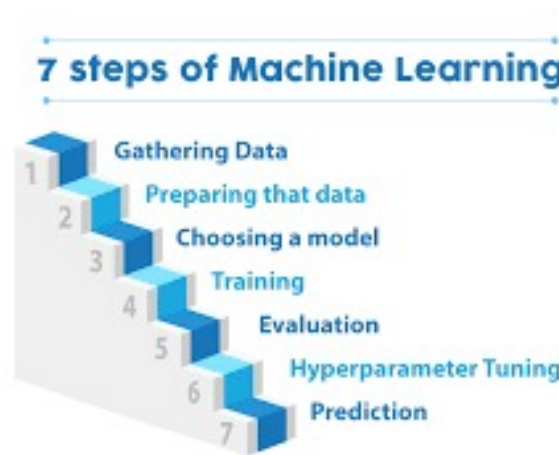


Figure 5.1: Stages Of Machine Learning

5.3 Scrapping and Cleaning

Steps: 1.Importing necessary libraries

- Numpy :- NumPy is a Python library used for working with arrays. It also has functions for working in the domain of linear algebra, fourier transform, and matrices.
 - Pandas :- Pandas is an open-source library that allows you to perform data manipulation and analysis in Python. Pandas Python library offers data manipulation and data operations for numerical tables and time series.
 - Datetime :- The datetime module supplies classes for manipulating dates and times. While date and time arithmetic is supported, the focus of the implementation is on efficient attribute extraction for output formatting & manipulation.
- 2.Parsing date as time and aggregating goals scored by the teams, followed by removing unnecessary features. 3.Rearranging columns. 4.Getting aggregate points of respective teams, followed by finding the current form of the team. 5. Getting last year's team positions as an independent variable. 6. Extracting the goal differences and compiling the final dataset.

```

[ ] # import all the necessary libraries

import numpy as np
import pandas as pd
import datetime
import time
import itertools

%matplotlib inline

[ ] # Read data from the csv into a dataframe
raw_data_1 = pd.read_csv('2005-06.csv')
raw_data_2 = pd.read_csv('2006-07.csv')
raw_data_3 = pd.read_csv('2007-08.csv')
raw_data_4 = pd.read_csv('2008-09.csv')
raw_data_5 = pd.read_csv('2009-10.csv')
raw_data_6 = pd.read_csv('2010-11.csv')
raw_data_7 = pd.read_csv('2011-12.csv')
raw_data_8 = pd.read_csv('2012-13.csv')
raw_data_9 = pd.read_csv('2013-14.csv')
raw_data_10 = pd.read_csv('2014-15.csv')
raw_data_11 = pd.read_csv('2015-16.csv')

```

Figure 5.2: Importing Snippet

```

[ ] # Parse date as time

def parse_date(date):
    if date == '':
        return None
    else:
        return time.strptime(date, '%d/%m/%y').date()

raw_data_1.Date = raw_data_1.Date.apply(parse_date)
raw_data_2.Date = raw_data_2.Date.apply(parse_date)
raw_data_3.Date = raw_data_3.Date.apply(parse_date)
raw_data_4.Date = raw_data_4.Date.apply(parse_date)
raw_data_5.Date = raw_data_5.Date.apply(parse_date)
raw_data_6.Date = raw_data_6.Date.apply(parse_date)
raw_data_7.Date = raw_data_7.Date.apply(parse_date)
raw_data_8.Date = raw_data_8.Date.apply(parse_date)
raw_data_9.Date = raw_data_9.Date.apply(parse_date)
raw_data_10.Date = raw_data_10.Date.apply(parse_date)
raw_data_11.Date = raw_data_11.Date.apply(parse_date)

#Gets all the statistics related to gameplay
columns_req = ['date', 'homeTeam', 'awayTeam', 'FTHG', 'FTAG', 'FTR']

playing_statistics_1 = raw_data_1[columns_req]
playing_statistics_2 = raw_data_2[columns_req]
playing_statistics_3 = raw_data_3[columns_req]
playing_statistics_4 = raw_data_4[columns_req]
playing_statistics_5 = raw_data_5[columns_req]
playing_statistics_6 = raw_data_6[columns_req]
playing_statistics_7 = raw_data_7[columns_req]
playing_statistics_8 = raw_data_8[columns_req]
playing_statistics_9 = raw_data_9[columns_req]
playing_statistics_10 = raw_data_10[columns_req]
playing_statistics_11 = raw_data_11[columns_req]

```

Figure 5.3: Data processing

5.4 Exploratory Data Analysis and Visualisation

1. Importing required libraries:

- **Scikit-Learn :-** Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python.
- **Matplotlib :-** Matplotlib is a useful cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy.
- 2. **Vital Data exploration:**
- We explore important data such as win rates of home teams, amount of features , number of matches played etc.
- 3. **Data Visualisation** using multiple scatter plots and making a scatter plot matrix.
- Scatter plot is a graph of two sets of data along the two axes. We used it to find dependencies among two features for better understanding of the data.


```
# Rearranging columns
cols = ['date', 'homeTeam', 'awayTeam', 'FTHS', 'FTAG', 'FTR', 'HTHS', 'ATGS', 'HTGC', 'ATGC', 'HTP', 'ATP', 'HHS', 'HQS', 'HPS',
        'HMA', 'HQS', 'APS', 'AQ2', 'APS', 'AMA', 'APS']

playing_statistics_1 = playing_statistics_1[cols]
playing_statistics_2 = playing_statistics_2[cols]
playing_statistics_3 = playing_statistics_3[cols]
playing_statistics_4 = playing_statistics_4[cols]
playing_statistics_5 = playing_statistics_5[cols]
playing_statistics_6 = playing_statistics_6[cols]
playing_statistics_7 = playing_statistics_7[cols]
playing_statistics_8 = playing_statistics_8[cols]
playing_statistics_9 = playing_statistics_9[cols]
playing_statistics_10 = playing_statistics_10[cols]
playing_statistics_11 = playing_statistics_11[cols]
```


Figure 5.4: Columns Rearranging

```
# Apply to each dataset

playing_statistics_1 = get_agg_points(playing_statistics_1)
playing_statistics_2 = get_agg_points(playing_statistics_2)
playing_statistics_3 = get_agg_points(playing_statistics_3)
playing_statistics_4 = get_agg_points(playing_statistics_4)
playing_statistics_5 = get_agg_points(playing_statistics_5)
playing_statistics_6 = get_agg_points(playing_statistics_6)
playing_statistics_7 = get_agg_points(playing_statistics_7)
playing_statistics_8 = get_agg_points(playing_statistics_8)
playing_statistics_9 = get_agg_points(playing_statistics_9)
playing_statistics_10 = get_agg_points(playing_statistics_10)
playing_statistics_11 = get_agg_points(playing_statistics_11)
```

 E:\Installed_Programs\Anaconda2\lib\site-packages\ipykernel_main_.py:54: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame. Try using .loc[row_index,col_indexer] = value instead

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

 E:\Installed_Programs\Anaconda2\lib\site-packages\ipykernel_main_.py:55: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame. Try using .loc[row_index,col_indexer] = value instead

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

Figure 5.5: Aggregating the points

```
[ ] # get Goal Difference
playing_stat['HWD'] = playing_stat['HWS'] - playing_stat['HGC']
playing_stat['ATD'] = playing_stat['ATWS'] - playing_stat['ATGC']

[ ] # Diff in points
playing_stat['DIFPTS'] = playing_stat['HWP'] - playing_stat['AWP']
playing_stat['DIFFORMTS'] = playing_stat['HFORMTS'] - playing_stat['AFORMTS']

[ ] # Diff in last year positions
playing_stat['DIFFLP'] = playing_stat['HomeTeam.P'] - playing_stat['AwayTeam.P']

[ ] # scale DIFPTS, DIFFORMTS, HWD, also by Matchweek
cols = ['HWS', 'ATWS', 'DIFPTS', 'DIFFORMTS', 'HWP', 'AWP']
playing_stat.M = playing_stat.M.astype(float)

for col in cols:
    playing_stat[col] = playing_stat[col] / playing_stat.M

[ ] def only_hw(string):
    if string == 'H':
        return 'H'
    else:
        return 'M'

    playing_stat['FTR'] = playing_stat.FTR.apply(only_hw)

[ ] # testing set (2015-16 season)
playing_stat_test = playing_stat[2005:]

[ ] playing_stat.to_csv("final_dataset.csv")
playing_stat_test.to_csv("test.csv")
```

Figure 5.6: final dataset

```

# Visualizing distribution of data
from pandas.tools.plotting import scatter_matrix

# The scatter matrix is plotting each of the columns specified against each other column.
# you would have observed that the diagonal graph is defined as a histogram, which mean that in the
# section of the plot matrix where the variable is against itself, a histogram is plotted.

# Scatter plots show how much one variable is affected by another.
# the relationship between two variables is called their correlation
# negative vs positive correlation

#HWD - Home team goal difference
#ATD - away team goal difference
#HTP - Home team points
#ATP - Away team points
#HWDiff - Difference in points
#HTDiff - difference in last years prediction

scatter_matrix(data[['HWD','ATD','HTP','ATP','Diffomts','Difflo']], figsize=(10,10))

array([[Outplot1b.axes._subplots.AxesSubplot object at 0x00000000037A7B00,
Outplot1b.axes._subplots.AxesSubplot object at 0x00000000042E7B00,
Outplot1b.axes._subplots.AxesSubplot object at 0x00000000044A7B00,
Outplot1b.axes._subplots.AxesSubplot object at 0x00000000046A7B00,
Outplot1b.axes._subplots.AxesSubplot object at 0x00000000048A7B00,
Outplot1b.axes._subplots.AxesSubplot object at 0x0000000004AA7B00])

```

Figure 5.7: Data Visualization

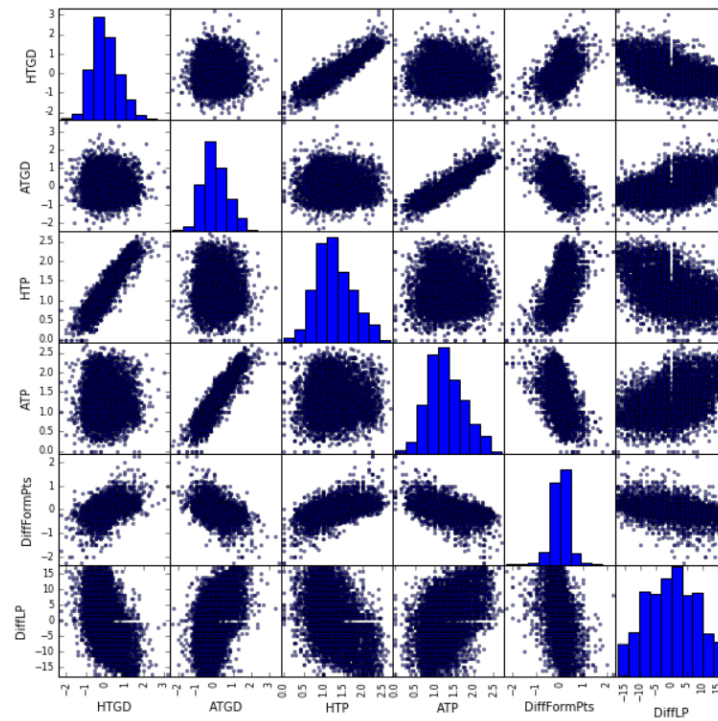


Figure 5.8: scatter Plot Matrix

Chapter 6

Model Fitting and Results

6.1 Data Preparation

- Separating the feature sets and target variables i.e. FTR(Full Time result).
- Shuffle and split the dataset into training and testing set using `train_test_split`.
- Training set is a subset of the data on which we train the model. Whereas test set is the data subset on which we test the model. We have to make sure the
- test set is large enough to yield statistically meaningful results and is representative of the data set as a whole.

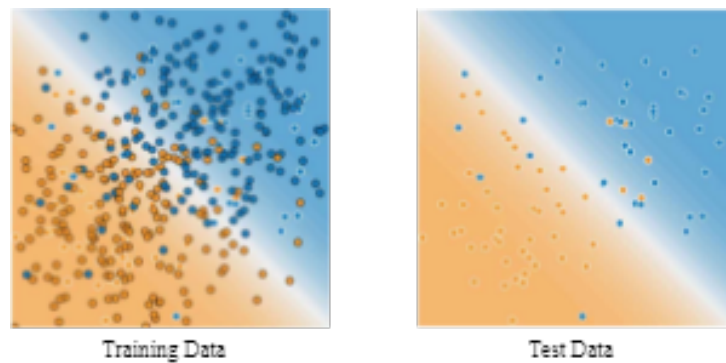


Figure 6.1: Training and test data visualisation

6.2 Training and Model evaluation

- Based on the f1-score we make predictions with respect to the classifier fit on the training data.
- The F1 score can be interpreted as a harmonic mean of precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. Formula is,
- We fit Logistic Regression, Support Vector Machine, KNN models on the training set and calculate their f1-score.

- Then we use XGBoost to tune the parameters. Finally using GridSearchCV and make_scorer we compile the results.

6.3 Expected Outcomes

- Ultimately, we will use our project to predict the winning team in a match, where there will be 3 potential outputs H(Home Team wins), A(Away Team wins) and D(Match Drawn).
- We will also get the f1_score of the given datasets which will verify the accuracy of our models.

6.4 Future Aspects and Improvements

- There are multiple ways of improving our model, and in turn the entire project. One way is having more data. More data results in more accuracy.
- Another way is utilizing more data features, even the ones that might sound unorthodox but ultimately affects the model accuracy e.g) Weather conditions on the day of the match, or individual player statistics against a specific team etc.
- Not only can we predict winning teams but given the right data we can calculate individual player statistics, potential goals that will be made by the team in the entire season etc.

Chapter 7

Conclusion

7.1 Summary:

Our main objective of building an expected result model by exploring different Machine Learning techniques have been accomplished. Indeed, we used modern Machine Learning algorithms such as logical regression, k-nearest neighbours and Support Vector Machines techniques to generate match outcome and match result predictions. We managed to find and improve a database containing enough information to generate expected metrics, through both shots and other in-game statistics, and ELO team ratings. We have also compared our predictions to benchmark methods in order to better understand our models' predictive performance. We have crucially found that our expected result models achieve a similar performance to bookmakers' odds, and that using expected result instead of actual goals in traditional models such as the Bing match predictor , helps achieve better predictive performance.

7.2 Finding Data :

One of the main challenges encountered during the project was to find suitable data to use to build an expected result model. A lot of time was spent doing research to find public databases which enabled me to find the actual database that we have used for this project. However, we believed that we could find more interesting data to build better models so we set out to scrape data from the football data UK website which contains a very large amount of data for each game.

7.3 Model and Parameter Choices

Difficult model choices had to be made to start of the project in terms of model design, which then led to an incremental approach using simple shot expected goals and ELO calculations, then adding elements that made the model more accurate.

Bibliography

- [1] Machine and Deep Learning in Sports Prediction(<https://medium.com/the-sports-scientist/machine-and-deep-learning-in-sports-prediction-9c76cd84b4b3>) Arman Hus-sain Apr 2020
- [2] Dataset: Football data from Uk Library
- [3] Dataset : <https://www.football-data.co.uk/> Last Updated: 2016
- [4] Support Vector Machine Algorithm (<https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm>) Oct 2020
- [5] Logistic Regression in Machine Learning <https://www.javatpoint.com/logistic-regression-in-machine-learning> Nov 2020
- [6] KNN for Machine Learning <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning> July 2019
- [7] Python Tutorial and basics (<https://www.tutorialspoint.com/python/index.htm>)
- [8] Football betting-The global gambling industry worth billions ([ww.bbc.com/sport/football/24354124](http://www.bbc.com/sport/football/24354124)) Joshua Feb 2021
- [9] Data experts are becoming football's best signings (<https://www.bbc.com/news/business-56164159>) John 2021
- [10] Using Machine Learning to Predict the Outcome of English County twenty over Cricket Matches (<https://arxiv.org/ftp/arxiv/papers/1511/1511.05837.pdf>) David 2020
- [11] Python Data Science Tutorial (https://www.tutorialspoint.com/python_data_science/index.htm) 2018