Football Predictor & Kelly Betting

A project submitted in partial fulfilment of the

Requirements for the award of the degree of

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Mr. Venkatarami Reddy

Roll.No : 12011041

Supervised by:

Dr. Mukesh Mann

Assistant Professor

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

SONEPAT-131201, HARYANA, INDIA

Github-Link -- ***Chaganti-Reddy/Kelly-Betting***

Acknowledgement:

In the present world of competition there is a race of existence in which those are having will to come forward succeed. Project is like a bridge between theoretical and practical working. With this willing I joined this particular project. First of all, I would like to thank the supreme power the Almighty God who is obviously the one has always guided me to work on the one has always guided me to work on the right path of life. Without his grace this project could not become a reality. Next to him are my parents, whom I am greatly indebted for me brought up with love and encouragement to this stage. I am feeling oblige in taking the opportunity to sincerely thanks to **Dr. M. N. Doja** (Director of IIIT Sonepat) and special thanks to my worthy teacher of Computer Science **Dr. Mukesh Mann**. Moreover, I am highly obliged in taking the opportunity to sincerely thanks to all the staff members of CSE department for their generous attitude and friendly behaviour. At last but not the least I am thankful to all my teachers and friends who have been always helping and encouraging me though out the year. I have no valuable words to express my thanks, but my heart is still full of the favours received from every person

Venkatarami Reddy (12011041)

Self Declaration:

I hereby declare that work contained in the research titled "**Football Predictor & Kelly Betting**" is original. I have followed the standards of project ethics to the best of my abilities. I have acknowledged all sources of information which I have used in the project.

Name: Venkatarami Reddy

Roll No.: 12011041

Department of Computer Science and Engineering,
Indian Institute of Information Technology,
Sonepat-131201, Haryana, India.

Certificate:

This is to certify that **Mr. Venkatarami Reddy** has worked on the research entitled "**Football Predictor & Kelly Betting.**" under my supervision and guidance.

The contents of the project, being submitted to **Department of Computer Science and Engineering, IIIT, Sonepat,** for the award of the degree of **B. Tech in Computer Science and Engineering,** are original and have been carried out by the candidate himself. This project has not been submitted in full or part for the award of any other degree or diploma to this or any other university.

Dr. Mukesh Mann
Supervisor

Department of Computer Science & Engineering

Indian Institute of Information Technology

Sonepat-131201, Haryana, India

Table of Contents

List of Abbreviations

ML	Machine Learning
Scikit Learn	Sklearn
SVM	Super Vector Machine
RFC	Random Forest Classifier
KNN	KNeighbours

Abstract

People love to wager on the outcome of matches, so the betting industry is booming. Cricket, football, horseback riding, and wrestling are just few of the sports that are available. Football, which is a major sport in Europe and North America, brings in a lot of money for the betting sector. People wager on the outcome of the game as well as on the outcome of the play event. Each football league produces a large amount of data. To forecast the outcome of a football match, several strategies and methodologies are used. Many of these strategies are based on past performance and goal totals. English Premier League is arguably the most competitive, so it is the most natural sport to study. There has never been an analytic definition of football because the sport is in a constant state of change, with certain characteristics becoming less significant over time. Here, machine learning could prove to be invaluable.

The purpose of this research project is to explore the application of statistical and machine learning methods in an effort to make predictions regarding the outcomes of matches such as win, lose, draw, goal differentials, and total goals by using historical data pertaining to the teams that participated in the match. The majority of the project consists of constructing a framework that, when given raw match data, generates appropriate features by drawing on historical statistics regarding match outcomes. Following that, we will measure the accuracy of the model utilising a variety of different techniques, all of which are based on machine learning.

1. Introduction

This chapter discusses the background research, which describes the current condition of the to-be-applied air theory. Show the important existing or parallel work done by others in this regard. Evident market developments are the impetus for undertaking this endeavour. This establishes the background of the problem to be tackled and the project's goal in order to demonstrate its universal relevance and significance. In addition, the intended contribution and advantages are addressed in the objectives and goals. The final section of this chapter on the project's ethical considerations has been presented.

1.1. Football and Data Science

A wide variety of people from all walks of life have always been fascinated by sporting events. One of the sports that enjoys the greatest level of popularity and acclaim all over the world is football. Data on each play is currently being compiled and subjected to additional analysis. This information can be utilised for a variety of purposes, including but not limited to the following: analysing player performance, training, opponent playing styles, predicting matches and league tables, player evolution, and computing betting odds. Due to the fact that matches are so unpredictable, it can be challenging to forecast how they will turn out. Forecasting the result of a single match is still largely unstudied despite the fact that sports analytics has received a lot of attention in recent years. This is because there are far too many factors that contribute to the unpredictability of the outcome.

Because of the increased accessibility of applications and websites through the internet, the betting business is growing. Bets placed on sporting events are now more widely accepted than in years past. Bets on sports provide entertainment and amusement for millions of people, who are all looking for ways to pass the time. Live coverage of sports events is also growing in popularity. There is now a sizable betting market for association football, which generates a significant amount of revenue and stimulates the interest of a large number of onlookers who are not particularly invested in the sport.

To win big money on sports betting, One have to use logic and reasoning power. The most important thing is that person have to get enough knowledge about the data science algorithms and start using them as a way of gaining an advantage over other bettors in this field.

Data science is an area of study that focuses on the development of strategies to address various challenges. The application of data science to sports betting involves picking a few numbers and then following those numbers in a predetermined sequence. The use of data science can provide gamblers with improved odds or even give them an advantage over the betting house.

1.2. Motivation

The primary objective of utilising data science in the sport of football is to determine which team will come out on top as well as their overall score. Football games typically go for 90 minutes, but they can go on for an additional 30 minutes if there have been no goals scored in the first 90 minutes.

One point is granted to a team that is successful in scoring a goal by kicking a football into the goal post of the opposing team. The winning squad is determined to be the one that finished with the most points. In league competitions, the team that comes out on top is awarded three points, while the losing team is awarded no points at all. If the game ends in a draw, both teams are awarded one point.

It is not possible to make accurate forecasts for upcoming matches based on the total amount of goals that have been scored because there are too many other variables at play. It's possible that the team is facing an opponent with a lower ranking, that players are in good form, that they've faced each other before, that key players have been injured, and that they're either playing at home or away. It is possible that fans have a substantial impact on the success of a team, and data imply that teams perform far better at home than they do when they are away from their stadium. Manchester United is the only side in the history of the Premier League to have won a greater number of away games compared to the number of games they have drawn or lost.

The current form of each team is the most important factor—though not the only one—to consider when attempting to forecast the outcome of a game between two teams. A team's shape can be understood by looking at their most recent string of results in comparison to those of the other teams. Therefore, the probability of a matchup between two teams can vary depending on their respective schedules.

Examining the in-game analytics that are now readily available on gaming websites is one way to get around this problem. A more in-depth investigation of these types of data sets can be used to forecast the achievement of goals. A more accurate method for forecasting the score and outcome of matches can be developed through the application of a variety of machine learning techniques.

1.3. Aim and Objectives

1.3.1 Aim

The objective of this project is to use a significant quantity of data accessible on the internet to estimate the predicted goal and outcome of a match. On such data, various machine learning models will be evaluated in order to maximise model performance.

1.3.2 Objectives

1. Web scraping website and building clean dataset.
2. Performing Exploratory Data Analysis and building a new dataset for Analysis.
3. To perform Feature Selection
4. To Preform Various Machine learning algorithm to predict match outcome
5. Evaluate and experiment with different methods to find best ML models

1.4. Backgrounds

Football is a sport in which two teams of 11 players compete against each other. The game lasts 90 minutes.

There is a specified rectangular area on either side of the pitch called goalpost where the other team attempts to place the ball.

Each time the team puts the ball in the goalpost, they will receive one point.

The team with the most points is declared the winner. In the event of a tie or no score, the game is considered a draw.

In league tournament:

- In Europe, each country usually has its own league with 20 teams. Each team plays the other team twice, once on their own turf (home game) and the other on the opposing team's turf (away game).
- The winning team receives three points, while the losing team receives one point.

Cards: A player receives a card when the referee discovers a foul during play.

- Yellow card: This card is given to a player when the offence is minor and there are no penalties.
- Red card: When a significant foul is discovered, the referee issues this card to the player with the intention of eliminating them from the field. A player that obtains two yellow cards receives a red card.

There are several additional events that we are not examining for this research.

1.5. Ethical Aspects

When conducting a study with ethical standards, there are many things to take into account. First, there is a widespread desire to study concepts like knowledge, accuracy, and error prevention. For instance, laws against creating, manipulating, or falsifying research data encourage reality and ward off mistakes.

This research include web scraping from internet. The only reason we scraped data was for academic research. Any information obtained through the website will not be sold in any form. We are not going to disclose this information or the findings that we get from using this data to any industry. We have not caused anyone any harm. We scraped the data not for the purpose of duplicating it but rather for the purpose of creating new value from it.

2: Literature Review

In this section, we will discuss previous research reports on football prediction. Numerous research papers and articles exist on football prediction based on a variety of parameters, including useful assumptions about football match outcomes and machine learning approaches, team rating, and predicted goals. Since the middle of the 20th century, one of the most prominent study topics has been the generation of forecasts for football score.

Moroney MJ [1] was the first person to develop a statistical model for soccer forecasts. He demonstrated that the Poisson distribution as well as the negative binomial distribution both gave an appropriate fit to the outcomes of soccer matches. In a subsequent study, Reep et al. [2] referred to this as the negative binomial distribution, and they discovered comparable findings for different types of ball games. They validated their theory by analysing data from the English Football League First Division over the course of four seasons, after which they applied the Negative Binomial distribution to data from other ball games. The conclusion of this discovery is that the same Negative Binomial distribution applies to the number of goals scored by a team, regardless of the quality of either that team or the quality of the opposition.

Maher [3] continued by arguing that the outcomes of matches are not entirely dependent on random occurrences but rather may be predicted based on historical information. Maher brought out the possibility that a negative binomial distribution may result from the accumulation of scores that were poisoned and had a different mean for each of the teams.

In 1974, using a comparison of the final league tables from the 1971-1972 football season, with projections given by Goal before the season began contributed to positive correlation, Hill [4] found a positive correlation between the two sets of data. With this he proved that the outcomes of matches are not entirely determined by random chance, but rather can be modelled and anticipated.

Two enhancements to the fundamental poison model were offered by Mark Dixon and Stuart Coles [5]. First, use interaction to account for an underestimated number of low-scoring matches, and then use the time component to weight matches that were completed lately. They developed a parametric model and fit it to English football league and cup data from 1992 to 1995. The model was inspired by a desire to exploit potential inefficiencies in the association football betting market; this is examined using odds from 1995 to 1996 from bookmakers. The technique is based on a Poisson regression model, but the data structure and dynamic nature of team performance complicate its application. It is demonstrated that maximum likelihood estimates are computationally realisable, and that the model has a positive return when used as the basis for a betting strategy.

Rue and Salvesen [6] used a statistician's approach to the prediction, proposing a Bayesian dynamic generalised linear model to estimate the time-dependent skill of all clubs in the league in order to forecast the outcome of the following week's matches. They performed a prediction with application to betting, a retrospective analysis of the final ranking, the detection of surprising matches, and an examination of how each team's properties change throughout the season.

Some forecast the outcome of a match, such as a win, a loss, or a tie, rather than a team's goal score. Previous research studies have attempted to forecast match results or actual goal scores. It would be interesting to compare the results of classification and regression techniques for the same match. To forecast match results, Forrest and Simmons [7] used classification techniques. They examine whether crowd numbers in team sports is proportional to the degree of uncertainty regarding the outcome of a match and discover that admissions to English soccer matches were also positively correlated with the quality of the teams involved and negatively correlated with a measure of the relative win probabilities of the competing teams. Despite the fact that audiences appear to prefer uncertain outcomes, a greater quality of strength across clubs may still result in a decline in aggregate viewership due to the extent to which home field advantage creates an uneven contest between equally strong teams.

Goddard [8] used an Ordered Probit Regression model to try to forecast the outcome of a football match. Bivariate Poisson regression is used to estimate forecasting models for goals scored and conceded. Both types of models are estimated using the same 25-year data set on English league football match outcomes. The optimal forecasting efficiency is driven with a 'hybrid' specification in which goals-based team performance covariates are used to predict win–draw–lose match outcomes.

Adam [9] created a generalised linear model to forecast the outcome of a European football championship match. This was calculated using the joint probability of a goal distribution (number of goals) and a binomial distribution (goal of one team given total number of goals). They also ensure that no home advantage is considered. The features are based on player and team choices from prior seasons. The prediction consisted of assigning probabilities to each team's elimination in the group stage, round of 16, quarterfinal, semi-final, and final, as well as the likelihood of winning the tournament.

Beginning in the 21st century, researchers began modelling match outcomes (win/draw/loss) directly, as opposed to predicting match scores and using them to calculate match outcome probabilities. Forrest and Simmons (2000) [10] evaluated the predictions of British newspaper tipsters, or journalists predicting the outcomes of upcoming football matches, and found that they performed better than random forecasting methods.

Kuypers [11] developed a model that is capable of predicting the outcomes of future matches by using variables that were derived from the results of previous seasons' matches. In addition to this, he was one of the pioneers in the betting market and was one of the first people to try to generate profitable betting strategies based on the model that he developed.

In the past, researchers have attempted to predict both actual match scores and match outcomes. In this project, it would be interesting to compare the performance of a classification model for match outcome versus a regression model for match scores. We will now examine more recent research conducted on the topic, utilising modern Machine Learning algorithms that will be of interest to us when testing different predictive models.

It is also interesting to investigate the algorithms that are utilised for making predictions in other team sports.

For instance, in 2006, Hamadani [12] compared Logistic Regression and SVM with various kernels in order to forecast the outcomes of National Football League matches (American Football). He utilised the Newton-Raphson method in his own implementation of Logistic Regression in MATLAB. He utilised support vector machines (SVM) with a variety of kernels including linear, polynomial, and tangent kernels. He applied methods to several different sets of attributes and achieved an accuracy of almost 65 percent.

Tavakol [13] utilised a linear model to forecast the outcome of the 2016 Euro Cup. He used data that included both basic information about countries and information about their players. Each country's score is predicted using the features. He also used information such as player appearance vs. goal scored. Players and their clubs are ranked. In a match, players who play for the same club and nation have an edge. He also combines player characteristics such as market worth and age. The practical examination demonstrates that the conceptually basic method is accurate for nations having historical data.

There are numerous methods for reducing the number of characteristics required to train a Machine Learning model. For instance, Kampakis's [14] models were initially optimised with only team characteristics, and afterwards with both team and individual characteristics. A basic prediction method paired with complicated hierarchical characteristics comprised the best model, which was proven to significantly beat a gambling industry benchmark.

To analyse a Dutch football competition, Tax et al. [15] coupled dimensionality reduction techniques with Machine Learning algorithms. They concluded that the PCA dimensionality reduction approach coupled with a Naive Bayes or Multilayer Perceptron classifier yielded the best results. The training of the model was conducted using thirteen seasons of Dutch Eredivisie match data pulled from public sources. On the public data training set, multiple combinations of dimensionality reduction techniques and classification algorithms have been systematically evaluated. Combining PCA with either a Naive Bayes or Multilayer Perceptron classifier yielded the greatest prediction accuracy on the public data feature set.

In a number of recent research publications, Bayesian networks have been used to forecast football results. Using Bayesian nets and other machine learning techniques, Joseph [16] predicted the outcome of matches played by Tottenham Hotspur Football Club during the Premier League campaigns of 1995–1996, as well as 1996–1997. Expert knowledge may be represented and made accessible using Bayesian networks. He compares the performance of an expert-built Bayesian networks to machine learning techniques like Decision Tree, Naive Bayes, and K-nearest. The results demonstrate that the expert Bayesian networks approach is more accurate than the others. Experts found Bayesian networks to be rather easy to construct, and its structure might be reused in similar problems in the future.

Artificial neural networks were employed by R.K. Balla [17] to forecast football game results. In his study, he makes match predictions for every team competing in the Premier League in the 2006–2007 season. He contrasts the outcome with the result attained using the Naive Bayesian, K-NN method. The number of matches played, points earned at a particular time, the result of the match played on the home and away grounds, and the current form used as set of features.

Artificial Neural Networks have also been employed in previous studies for other sports. For forecasting NFL (National Football League) games, Kahn [18] utilised ANN. Classification problems are well-suited to neural networks. It is capable of non-linear mapping. He deployed ANN because back propagation multi layer perception allows a lot of freedom.

Hucaljuk [19] used features such as the team's current form based on results from the previous six games, previous head-to-head results, current ranking position, number of injured first-team players, average number of goal scored verses received per game, and nearly other 15 features in his ML techniques. He used them in Naive Bayes, Bayesian Networks, LogitBoost, K-nearest, Random Forest, and Artificial Neural Networks method. Among this ANN has higher accuracy. This experiment is of particular interest to us because we will be testing multiple algorithms in a similar fashion to determine which works best with our data and features.

Football game final scores are currently predicted using methods that model the total number of goals scored by each team, with parameter estimation typically based on the maximum likelihood approach. A weighted likelihood approach that Dimitris and Ioannis [20] proposed enables the model to underweight a particular football score. Even if unexpected scores are observed, this procedure may still provide reliable estimates. For this study, the Champions League was utilised.

Darwin, P & Dra, H (2016). Predicting Football Match Results with Logistic Regression. 2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA). Darwin, P & Dra, H (2016). Predicting Football Match Results with Logistic Regression. 2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA). Darwin, P & Dra, H (2016). Predicting Football Match Results with Logistic Regression. 2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA).

Darwin and Dra. [21] use logistic regression to forecast match results. They foresee how the 2015–16 English Premier League season will turn out. Their work differs from others' because they don't create their own significant variables; instead, they only use significant variables found in research in the same field. Variations of training data from 2010 to 2015 were used to build the model. They made use of functions like Home Offence, Home Defence, and Away Offence.

By looking at the historical research done on the subject of creating models to forecast football match results, we have been able to obtain valuable information on the different strategies we should attempt utilising as well as the potential problems we should avoid.

In view of the goal of our project, we will now examine how team ranks or points have previously been applied to forecasting. Any rating system's obvious goal is to give the subject of the rating a ranking list. A ranking list based on a single event is not always reliable in a sports competition like chess, football because individual performances fluctuate from time to time Additionally, it might be necessary to evaluate the performances of individuals or teams who have never engaged in direct competition.

In 1978 to rank chess players, Elo [22] created the most well-known rating system, still in use today. Many betting house provides Elo states for football teams.

Lasek [23] utilised a variety of scoring systems to forecast the outcome of the 2016 European Championships. To begin, he evaluated the rating systems to calculate the odds of match results amongst all participating nations. Then he used Monte Carlo simulations to calculate the chances of making it through a particular stage of the competition.

For his work, Hvattum [24] employed the ELO grading system to produce covariates, which were subsequently used in ordered logit regression models. Covariates are derived from the

ELO rating system used in ordered logit regression models. He compared ELO-based prediction to other techniques of forecasting. Using the ELO rating, he achieves a better performance.

Fenton [25] propose the pi-rating, a new rating system. This rating system is applied to other sports where relative performance and score are regarded as reliable indicators for forecasting results. He evaluated the rating using the Elo score and market odds, and result shows that Pi ratings outrank Elo ratings.

Estimating the probabilities of scoring for each goal scoring opportunities determines the projected outcome of a football match. This strategy was used by Eggles [26] to predict the outcome of a football match. His research demonstrates that the probability of obtaining goal scoring opportunities accurately reflect reality.

We will be inspired by these approaches for our project, but we will enhance the expected objectives model by experimenting with different Machine Learning algorithms and adding characteristics in order to get the best possible predictions.

3: Techniques and Methods

3.1 Technology

We will clean the dataset after we have collected the data. Pre-processing data in order to move it into useable form for Model design and testing which is a vital stage in any Data science project.

We will use following technology in this project.

Python : Python is a high-level, general-purpose programming language that is used primarily for data science, automation, web development, and artificial intelligence. It is also very helpful in making complex scientific and mathematical applications. And with that, Python has become one of the programming languages with the most rapid expansion.

Python is an ideal programming language for developers who want to write scripts for applications and websites because it is simple to learn. Python is an open-source programming language, which means that anyone can use it for free and that its development follows a community-driven model. Python has a sizable community of users and is widely implemented in a variety of fields, including academia and industry. More than seventy percent of machine learning developers and data scientists use it, making it the most popular language in its field.

In addition, there is an abundance of helpful analytics libraries available, such as NumPy, Pandas, and Matplotlib, which are some of the most frequently used libraries in Data Science projects.

Other Most popular language is R. Python, on the other hand, offers superior scalability, performance, and integration. Python is going to be used because it is the best choice for this research. Since Python has structured ways to deal with large amounts of data and enough libraries to deal with machine learning and deep learning, it is a good choice for those who need a programming language that can handle these tasks.

Some of the python libraries we are using for this research are NumPy, pandas, matplotlib, and seaborn.

Pandas - Pandas is the most widely used open-source Python tool for data analytics and machine learning applications. It's based on a second NumPy module that allows us to work with multidimensional arrays. Pandas, along with NumPy, are probably the most important libraries for Python data science activities. The main Pandas object is a _dataframe_, which is used to store data into rows and columns. The best way to think of Pandas is to consider it as a very powerful version of Excel, and a dataframe is like a spreadsheet. Each column of a dataframe object is a Pandas _series_ object. A Pandas series object is built from a NumPy array (series is like an array but with labelled index). We import using the following syntax by convention:

```
import pandas as pd
```

NumPy - also known as NumPy or number Python is an open-source Python module that simplifies and complicates numerical tasks. Working with machine learning and deep learning apps necessitates complex math and large datasets. NumPy makes the implementation of these operations more easier and more effective than its pure Python counterpart. NumPy allows performing various mathematical operations on arrays easily. NumPy provides different ways to slice the arrays to get the data we need.

We can install NumPy really easily via `pip` or `conda`

```
pip install numpy
```

```
conda install numpy
```

Once installed, the canonical way to import it in Python is giving it the alias np, so it will look like: `import numpy as np`

Beautiful Soup - Beautiful Soup is a Python library that allows you to parse HTML and XML files. It builds a parse tree for parsed pages, which may be used to extract data from HTML and is important for web scraping.

Sklearn : In Python, Scikit-learn (Sklearn) is the most usable and robust machine learning package. Scikit-learn is a high-level machine learning library containing machine learning algorithms, example datasets, data pre-processing & pipelines. It uses a Python consistency interface to give a set of fast tools for machine learning and statistical modelling, such as classification, regression, clustering, and dimensionality reduction. It was around for more than 10 years and is used throughout the industry.

It is used in fast prototyping and testing ideas, part of more complicated pipelines, and often as part of Machine Learning research.

And will use other library as we progress further in this study.

Sklearn has several different classification algorithms.

3.2 Machine Learning Techniques

In this sections, we will provide an overview of supervised machine learning techniques. The process of learning a function that maps input data to output data based on example combinations of input and output data is referred to as supervised learning. The process of classification takes place when the output is in the form of a category, whereas the process of regression takes place when the output is in the form of a continuous number. For our purposes, we are only interested in the supervised learning landscape of Machine Learning because we want to predict the outcome category (home win, draw, or away win) or the number of goals scored by a team (continuous number).

3.2.1 K-nearest neighbours

The K-nearest neighbours algorithm makes predictions by taking the average of the labels of the K-nearest neighbours in feature space.

The number of neighbours whose labels will be averaged out by the algorithm is denoted by the variable K. It is a hyperparameter, to be specific.

- for classification: the most common class among the k neighbors
- for regression: the average of the values for the k neighbors

Finding a predetermined number of training samples that are the closest in distance to a new sample that needs to be classified is the core concept that underpins the nearest neighbour classification method. This number is denoted by the letter k. These adjacent samples will be used to define the label for the newly generated sample. Classifiers based on the k-nearest neighbour have a user-defined constant that is always the same value for the number of neighbours that need to be determined. Radius-based neighbour learning algorithms are another type. These algorithms have a variable number of neighbours based on the local density of points, which includes all of the samples that are contained within a predetermined radius.

`neighbors` is a package that is part of the `sklearn` module.

3.2.2 Decision Trees

Decision trees are a type of supervised learning algorithm that can be applied to classification and regression tasks respectively.

Decision trees are assigned to the information based learning algorithms which use different measures of information gain for learning.

The primary purpose of decision trees is to identify those descriptive features that offer the most "information" concerning the target feature. Once this has been accomplished, the dataset is partitioned along the values of these descriptive features in such a way that the values of the target feature in the sub-datasets that are produced are as untainted as is practically possible.

One of the criteria for determining which of two descriptive features is the most informative is which one leaves the target feature in the purest form. This process of determining which feature is the "most informative" is repeated until we meet a stopping criterion, at which point we arrive at the final destination, which are referred to as leaf nodes.

The primary components of a decision tree are the root node, the interior nodes, and the leaf nodes, all of which are connected by branches.

Sklearn contains a highly effective decision tree classification model : `sklearn.tree.DecisionTreeClassifier`

3.2.3 Naive Bayes

In machine learning, a Bayes classifier is a simple probabilistic classifier, which is based on applying Bayes' theorem. The feature model used by a naive Bayes classifier makes strong independence assumptions. This means that the existence of a particular feature of a class is independent or unrelated to the existence of every other feature.

The benefit of using Naive Bayes classifiers is that they are highly scalable when presented with large amounts of data.

3.2.4 Support Vector Machines

Support vector machines (SVM) is a supervised machine learning algorithm which can be used for both classification and regression. However, it is mostly used in classification problems. In this algorithm, It plot each data item as a point in n – dimensional space(where n is number of features) with the value of each feature being the value of a particular coordinate. It perform classification by finding the hyperplane that differentiates the classes very well. In other words SVM outputs an optimal hyperplane which categorise new examples.

3.2.5 Random Forest Classifier

Random Forests are supervised ensemble learning models used for classification and regression. Random Forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. Ensemble models are those which combine the predictions of multiple models to make a prediction. Random forests ensemble lots of weak models to form a stronger model with greater representational capacity. Ensembling weak (low capacity) models can reduce the variance in their predictions, hence ensembling is a form of regularisation. In

The randomness of random forests come from:

- each tree being trained on a bootstrapped dataset
- each tree being trained on a subset of features of each example

3.2.6 AdaBoost

Boosting is an ensemble method that combines a sequence of weak classifiers which are fit on successively modified versions of the data, which increasingly prioritise examples misclassified by the previous model.

AdaBoost is a classification algorithm. Boosting algorithms vary in how they adjust the weights of the examples that are samples into each successive bootstrapped dataset, and in how they weight the contribution of each hypothesis to the final prediction. AdaBoost is the most popular boosting algorithm. It is used for classification problems. The name AdaBoost is short for **adaptive boosting**.

3.2.7 Logistic Regression

Logistic Regression is used for classification problems. Here the aim is to predict the group to which the current object under observation belongs to. Logistic Regression measures the relationship between the dependent variable and the one or more independent variable by estimating probabilities using its underlying logistic function. By default, logistic regression cannot be used for classification tasks that have more than two class labels, so-called multi-class classification.

Instead, it needs to be modified in order to support problems with multiple classes of classification. Splitting the multi-class classification problem into multiple binary classification problems and then fitting a standard logistic regression model on each of the subproblems is a common strategy for adapting logistic regression to multi-class classification problems. This is one of the more popular approaches. Alternately, one could modify the logistic regression model so that it directly supports the prediction of multiple class labels. This would be an alternative approach. To be more specific, the goal is to forecast the likelihood that an input example will be found in each known class label.

The probability distribution that is used to define the probabilities of multiple classes is referred to as a multinomial probability distribution. The term "multinomial logistic regression" refers to a type of logistic regression model that is adapted to learn and predict a multinomial probability distribution. We will use this approach in this research.

3.3 Prediction Models

In this research project we will implement 3 prediction models.

1. Outcome
   In this experiment, we will attempt to forecast the outcome of a football match as either a win for the home team, a draw, or a loss for the home team.
2. Goal Difference
   With the help of this model, we will make an attempt to forecast the difference in the number of goals scored by the Home Team and the Away Team.
   In order to accomplish this, we will introduce a new feature that we will refer to as the Goal Difference between the Home Team Goal and the Away Team Goal.
3. Total Goals
   In this mode, we will attempt to forecast the total goals scored in the match.

4: Dataset

4.1 Web Scraping

There are numerous data available over the internet. Python and the Beautiful Soup library were utilised in order to harvest data regarding the English Premier League from the internet for the purpose of this project. Initially, We have gathered data from season 2012 to 2021. And for the second phase we have scrape data of season 1990 to 2012.

Web scraping is quickly evolving into one of the most widely used strategies for gathering data from websites and other online sources. The format of the content that is displayed on the website is defined by HTML. On our end, the data was obtained through the use of Requests as well as the BeautifulSoup library. When it comes to sending HTTP requests to a particular URL, one of the most essential aspects of Python is the Requests library. XML and HTML files can both have data extracted from them using this libraries. It accomplishes this by generating a parse tree from the source code of the page. This parse tree can then be used to extract data in a hierarchical structure that is simpler to read. BeautifulSoup is both incredibly quick and incredibly relaxed in terms of its requirements.

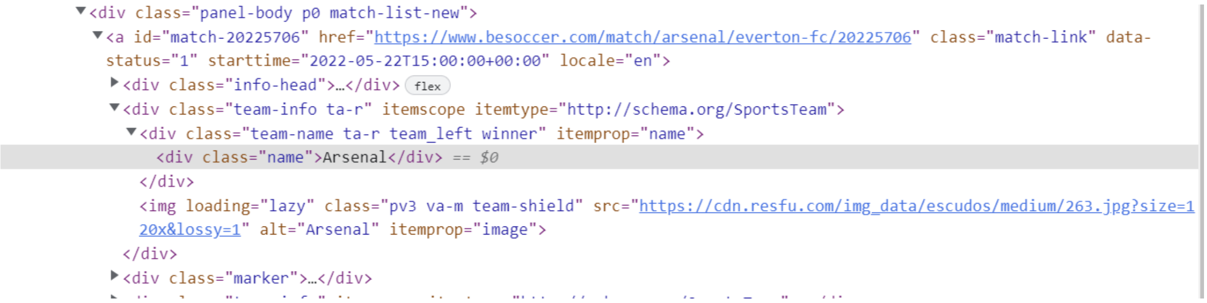
In order to successfully install Beautiful Soup and requests, we will need a framework that was written in Python. In addition, some other frameworks that are supported or additional can be installed using the PIP command given below:

```
pip install beautifulsoup4
```

```
pip install requests
```

For the purpose of sending a GET request to a particular URL, the requests module of Python includes a built-in method known as get(). With request.get(URL)we can have access to website same as we visit website. Here, we have passed “https://www.besoccer.com/competition” , And we can pass this html page to BeautifulSoup.

For Excluding unwanted data and scrap reliable information only, we have to inspect the webpage. Following image shows the inspect view of website.



```
▼<div class="panel-body p0 match-list-new">
  ▼<a id="match-20225706" href="https://www.besoccer.com/match/arsenal/everton-fc/20225706" class="match-link" data-
   status="1" starttime="2022-05-22T15:00:00+00:00" locale="en">
    ▶<div class="info-head">…</div> flex
    ▼<div class="team-info ta-r" itemscope itemtype="http://schema.org/SportsTeam">
      ▼<div class="team-name ta-r team_left winner" itemprop="name">
          <div class="name">Arsenal</div> == $0
        </div>
        <img loading="lazy" class="pv3 va-m team-shield" src="https://cdn.resfu.com/img_data/escudos/medium/263.jpg?size=1
        20x&lossy=1" alt="Arsenal" itemprop="image">
      </div>
    ▶<div class="marker">…</div>
```

Fig. 1: Inspect view of Website

By just hovering through the webpage, we can select the elements; and corresponding code will be available like shown in the above image.

List of matches is inside class = “panel-body p0 match-list-new”. And inside that we have home team name in class = “match-link”.

Using find method of BeautifulSoup we can travers inside that tag.

```
matches_box = soup.find('div', {'class': 'panel-body p0 match-list-new'})
matches = matches_box.find_all('a', {'class': 'match-link'})
```

We were able to find data for the Premier League tournament using this method, including the names of the home team and the away team, the goal score, the date, and the yellow and red card totals. The data that was scraped has been saved in a different directory for each season.

Following structure shows file and data saved in csv files.

Result_{year}_premier_league.csv

- Home_Team = Name of Home Team
- Away_Team = Name of Away Team
- Result = Outcome of Match
- Link = URL of the match
- Season = Year
- Round = Week of the Season
- League = Name of the league

Match_elo_{year}_premier_league.csv

- Link = URL of the match
- Home_ELO = ELO rating of Home Team
- Away_ELO = ELO rating of Away Team

Match_card_{year}_premier_league.csv

- Date = Date of Match played
- Home_Yellow = No. of Yellow card given to the Home Team
- Home_Red = No. of Red card given to the Home Team
- Link = URL of the match
- Away_Yellow = No. of Yellow card given to the Away Team
- Away_Red = No. of Red card given to the Away Team

4.2 Data Cleansing and Preparation

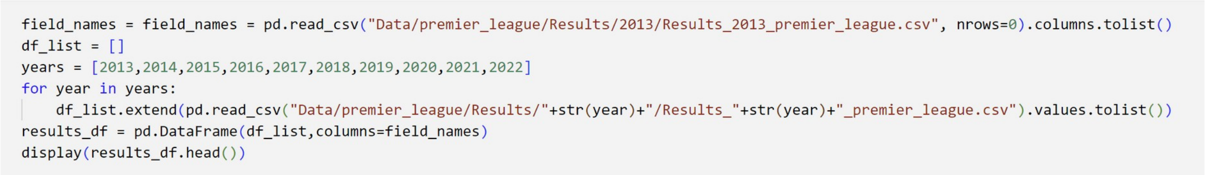
4.2.1 Data Cleaning and Merging

“Data cleaning” refers to the process of eliminating inaccurate, corrupted, incorrectly formatted, duplicate, or incomplete data from within a dataset. This can be accomplished by either manually correcting or removing the offending data. When multiple sources of data are combined, there is a greater chance that some of the data will be duplicated or incorrectly labelled. This is because there are more data sources to combine.

Analysing and pre-processing the data in order to guarantee that it will be in a format that is usable for us to use when training and testing various models is an essential step that must be completed before we can build our model.

In this research we have initially combine data from season 2012 to 2021. Sole purpose is to find select features are more suit for implementation of Machine Learning technique in later stage. Which contains 3700 records.

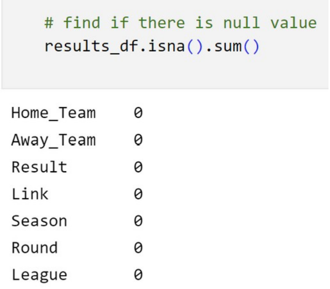
We were able to successfully combine all of the 'Result_{year}_premier_league.csv' files into a single data frame by utilising the code that is presented in the following figure.



```
field_names = field_names = pd.read_csv("Data/premier_league/Results/2013/Results_2013_premier_league.csv", nrows=0).columns.tolist()
df_list = []
years = [2013,2014,2015,2016,2017,2018,2019,2020,2021,2022]
for year in years:
    df_list.extend(pd.read_csv("Data/premier_league/Results/"+str(year)+"/Results_"+str(year)+"_premier_league.csv").values.tolist())
results_df = pd.DataFrame(df_list,columns=field_names)
display(results_df.head())
```

Fig. 2: Snippet of Combining Results csv files

Pandas has built-in functions called isnull() and isna() that can identify missing values. Both of them are responsible for the same activity. And result_df.isna().sum() returns the total number of missing values in from each column.



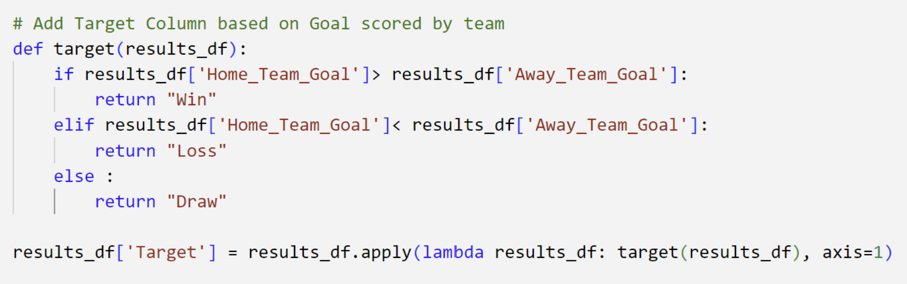
```
# find if there is null value
results_df.isna().sum()
```

```
Home_Team    0
Away_Team    0
Result       0
Link         0
Season       0
Round        0
League       0
```

Fig. 3: Output of isna() method

So, this data frame do not have any null or value. Through the splitting the ‘Result’ column of the data frame, we were able to retrieve the ‘Home_Team_Goal’ and ‘Away_Team_Goal’ columns.

From the columns labelled ‘Home_Team Goal’ and ‘Away_Team_Goal’, we were able to derive the ‘Target’ column, which labels the Match result in Win, Loss, Draw for Home Team.

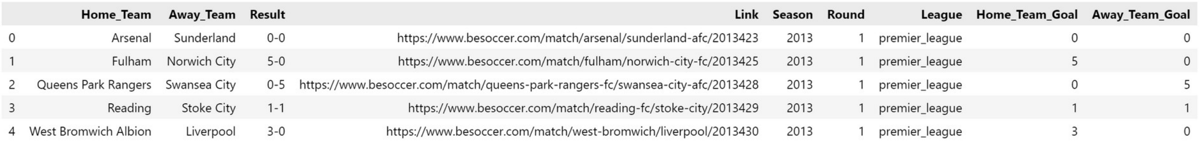


```
# Add Target Column based on Goal scored by team
def target(results_df):
    if results_df['Home_Team_Goal']> results_df['Away_Team_Goal']:
        return "Win"
    elif results_df['Home_Team_Goal']< results_df['Away_Team_Goal']:
        return "Loss"
    else :
        return "Draw"

results_df['Target'] = results_df.apply(lambda results_df: target(results_df), axis=1)
```

Fig. 4: Snippet of Creating Target column

The image below depicts the first 5 rows of the Data frame.



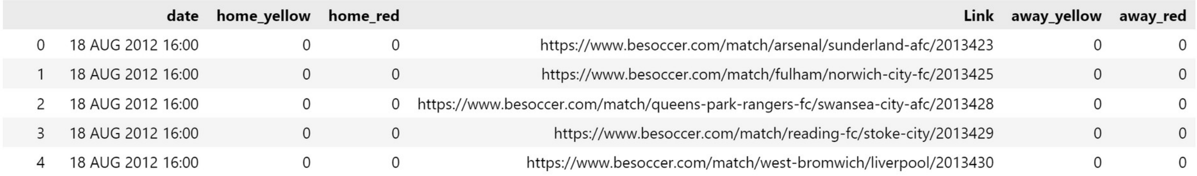
	Home_Team	Away_Team	Result	Link	Season	Round	League	Home_Team_Goal	Away_Team_Goal
0	Arsenal	Sunderland	0-0	https://www.besoccer.com/match/arsenal/sunderland-afc/2013423	2013	1	premier_league	0	0
1	Fulham	Norwich City	5-0	https://www.besoccer.com/match/fulham/norwich-city-fc/2013425	2013	1	premier_league	5	0
2	Queens Park Rangers	Swansea City	0-5	https://www.besoccer.com/match/queens-park-rangers-fc/swansea-city-afc/2013428	2013	1	premier_league	0	5
3	Reading	Stoke City	1-1	https://www.besoccer.com/match/reading-fc/stoke-city/2013429	2013	1	premier_league	1	1
4	West Bromwich Albion	Liverpool	3-0	https://www.besoccer.com/match/west-bromwich/liverpool/2013430	2013	1	premier_league	3	0

Fig. 5: First 5 Rows of Result DataFrame

There are 12164 Match records in this Data frame which contains information about Name of Home Team and Away Team, Final Goal score line, Link, Season, Round and League name, Goal score by Home Team, Away Team, and Match result as Home Team Win, Loss or Draw.

In the same manner we were able to successfully combine and load all of the csv files containing match card data into a single data frame(match_card_df).

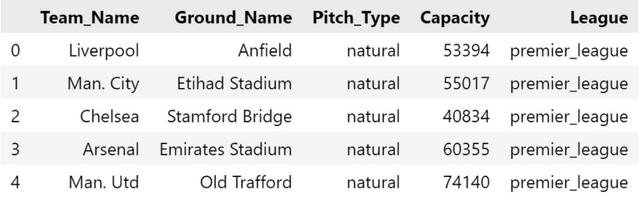
Following is image shows first 5 rows of match_card_df.



	date	home_yellow	home_red	Link	away_yellow	away_red
0	18 AUG 2012 16:00	0	0	https://www.besoccer.com/match/arsenal/sunderland-afc/2013423	0	0
1	18 AUG 2012 16:00	0	0	https://www.besoccer.com/match/fulham/norwich-city-fc/2013425	0	0
2	18 AUG 2012 16:00	0	0	https://www.besoccer.com/match/queens-park-rangers-fc/swansea-city-afc/2013428	0	0
3	18 AUG 2012 16:00	0	0	https://www.besoccer.com/match/reading-fc/stoke-city/2013429	0	0
4	18 AUG 2012 16:00	0	0	https://www.besoccer.com/match/west-bromwich/liverpool/2013430	0	0

Fig. 6: First 5 Rows of Match card DataFrame

Also we have stored 'Stadium_premier_league.csv' into another data frame. Which contains information like Pitch type, capacity of stadium, and Ground name.

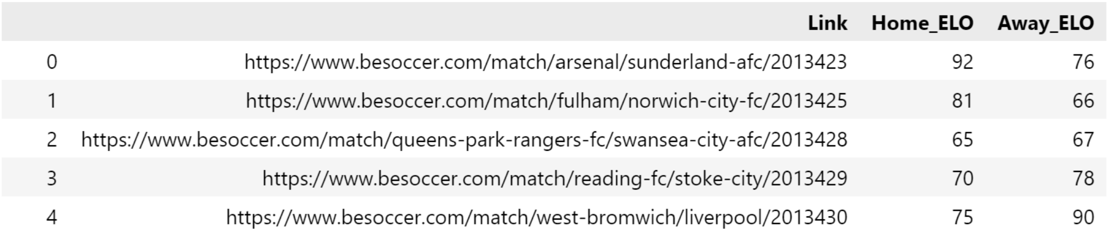


	Team_Name	Ground_Name	Pitch_Type	Capacity	League
0	Liverpool	Anfield	natural	53394	premier_league
1	Man. City	Etihad Stadium	natural	55017	premier_league
2	Chelsea	Stamford Bridge	natural	40834	premier_league
3	Arsenal	Emirates Stadium	natural	60355	premier_league
4	Man. Utd	Old Trafford	natural	74140	premier_league

Fig. 7: First 5 Rows of Team info DataFrame

There were 3 records were Pitch_Type have null values. We have filled that value with most number of Pitch Type which is natural.

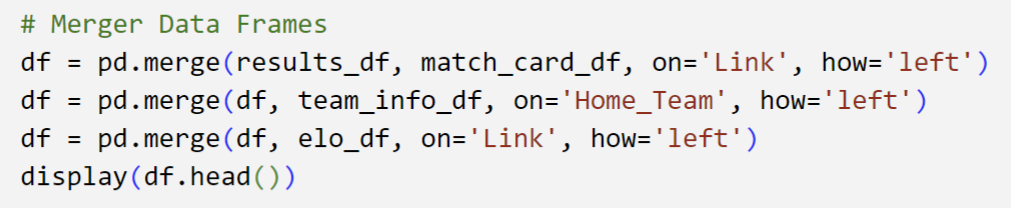
Using the same way we have combine csv files containing data about ELO have stored in Data frame. Following image shows first 5 rows of that data.



	Link	Home_ELO	Away_ELO
0	https://www.besoccer.com/match/arsenal/sunderland-afc/2013423	92	76
1	https://www.besoccer.com/match/fulham/norwich-city-fc/2013425	81	66
2	https://www.besoccer.com/match/queens-park-rangers-fc/swansea-city-afc/2013428	65	67
3	https://www.besoccer.com/match/reading-fc/stoke-city/2013429	70	78
4	https://www.besoccer.com/match/west-bromwich/liverpool/2013430	75	90

Fig. 8 First 5 Rows of Elo Data Frame

We were able to combine all of these data frames into a single one by making use of the merge method in the Pandas library. The relevant code is presented below.



```
# Merger Data Frames
df = pd.merge(results_df, match_card_df, on='Link', how='left')
df = pd.merge(df, team_info_df, on='Home_Team', how='left')
df = pd.merge(df, elo_df, on='Link', how='left')
display(df.head())
```

Fig. 9: Snippet of Merging Data Frame

We have used this data for build base model and select feature for further tuning of model.

4.3.2 Data Insights

Following is some insights about on full data set.

- We have total 12164 match record from season 1990 to 2021.
- There were 52 teams played in this duration.
- Total 32110 Goals have been scored by Teams in this duration.
- Total 18245 Goals scored by Home Team while 13865 Goals scored by Visiting Team
- Average 2.63 Goals registered per Match.
- 3182 Matches have been Draw.

The following graph displays the total number of matches played by each team.

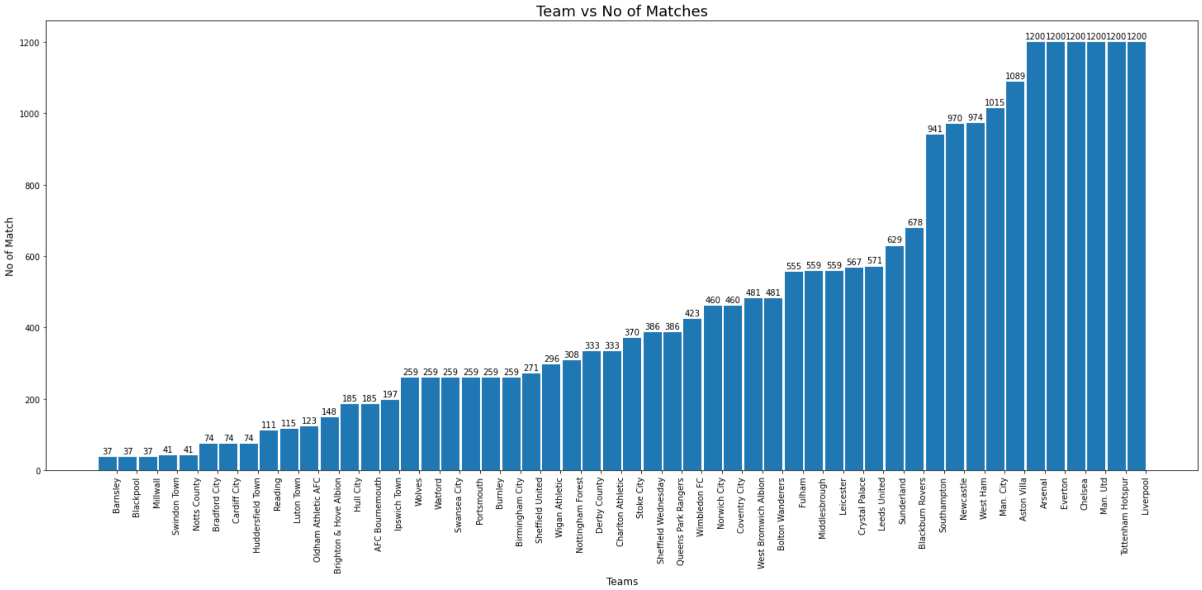


Fig. 10: No of Matches played by Team

- Everton, Chelsea, Liverpool, and Manchester United have all played the most games out of all of the clubs in this league. Over the course of their history, from 1990 to 2021, each member of this team competed in 1200 games.
- When compared to the other teams, Barnsley, Blackpool, and Millwall have participated in the fewest number of games.

The following bar graph depicts the number of wins achieved by a Team when playing at either Home or Away.

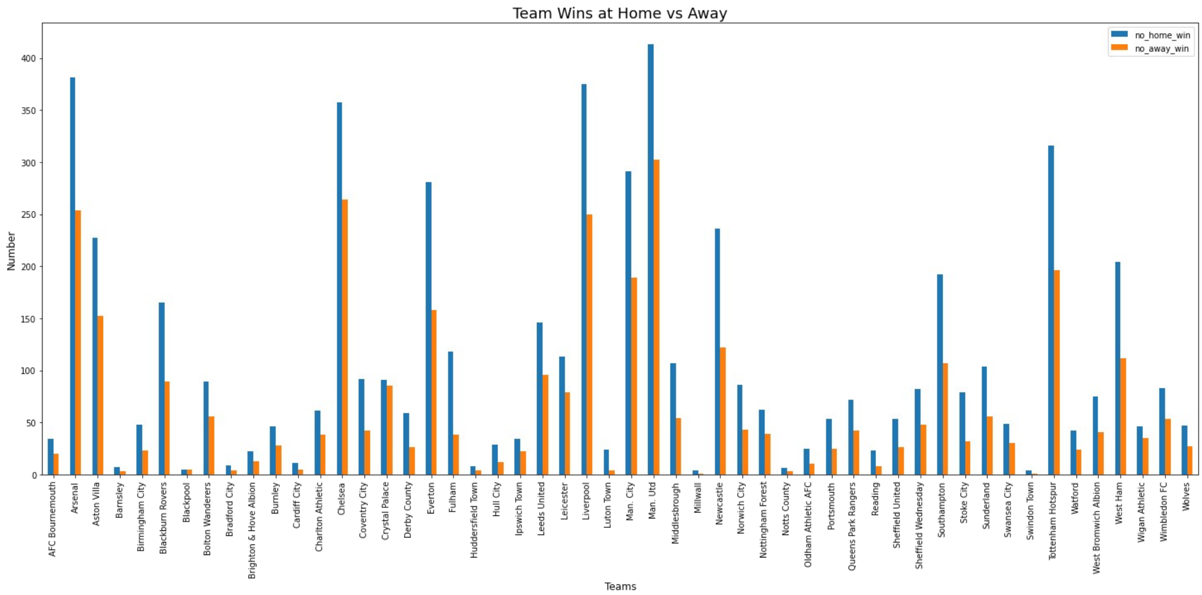


Fig. 11: Team Wins at Home vs Away

- Taking a look at this graph makes it abundantly evident that every single team puts up impressive numbers whenever they are competing in their very home stadium. When they are not participating in their home arena, they have a lower percentage of victories.

The following bar graph depicts the number of victories and losses incurred by a team while they were away from home.

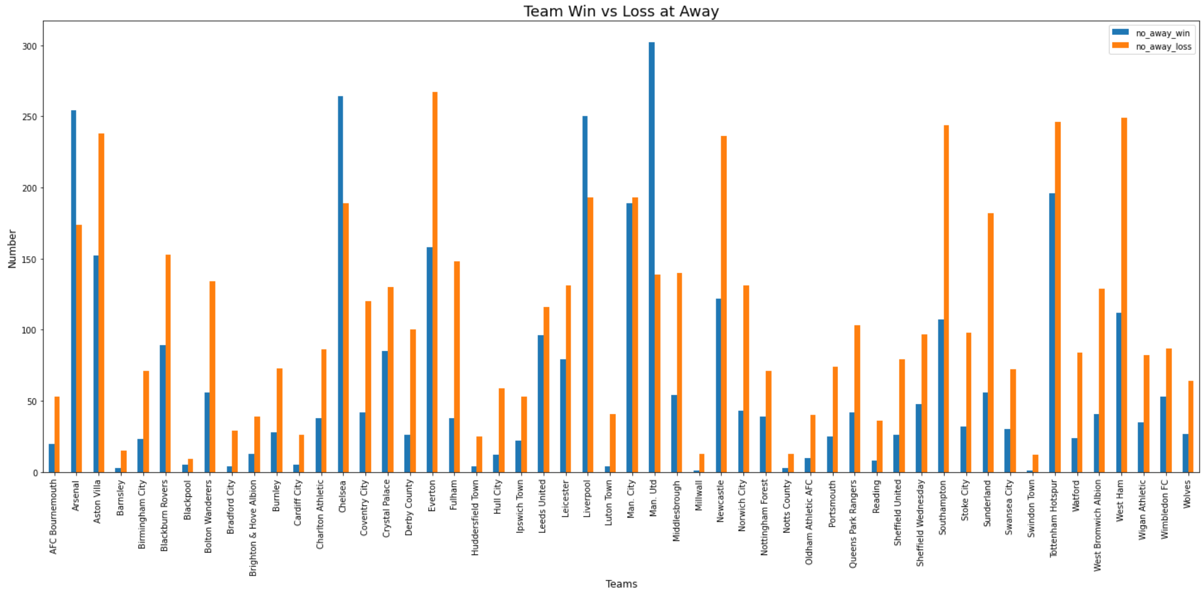


Fig. 12: Team's Win and Loss at Away

- When they play on the opponent's turf, some of the Teams are showing strong performance. Teams such as Arenal, Chelsea, Liverpool, and Manchester United have a higher win percentage when playing away from home compared to their loss percentage.

The following bar graph depicts the number of victories and losses incurred by a team while they were playing at home turf.

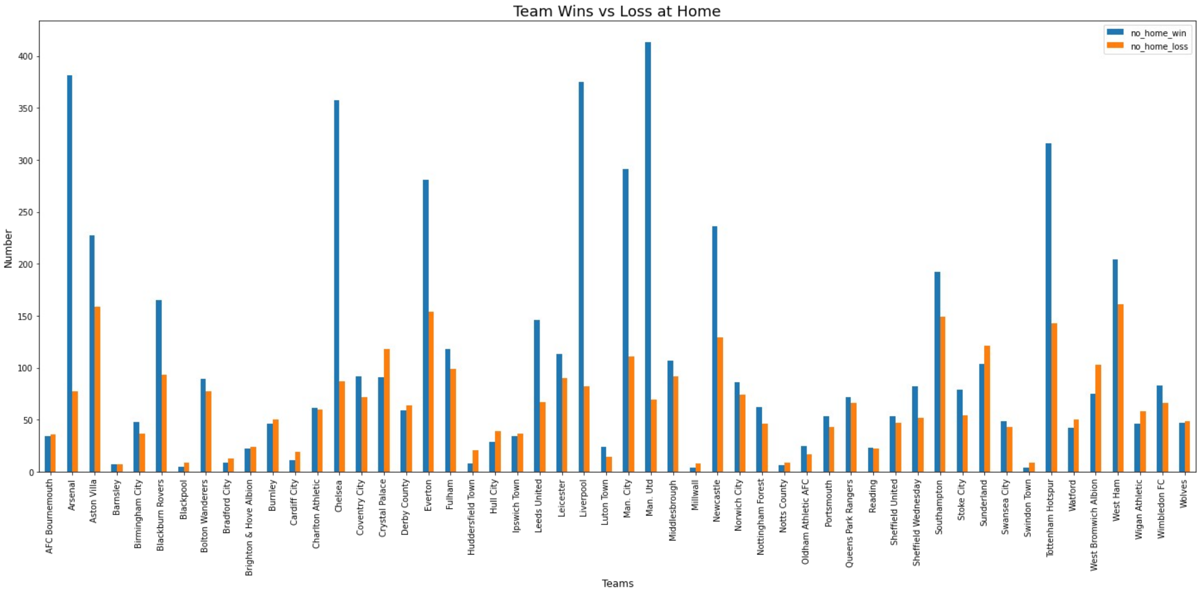


Fig. 13: Team's Win and Loss at Home

- When they are competing on their own turf, the vast majority of teams have a high percentage of victories. Teams with a winning percentage that is significantly higher than their losing percentage include Manchester City, Manchester United, Liverpool, Chelsea, and Tottenham Hotspur.

The following graph compares the number of wins, draws, and losses accumulated by each Team on whether they played at home or away.

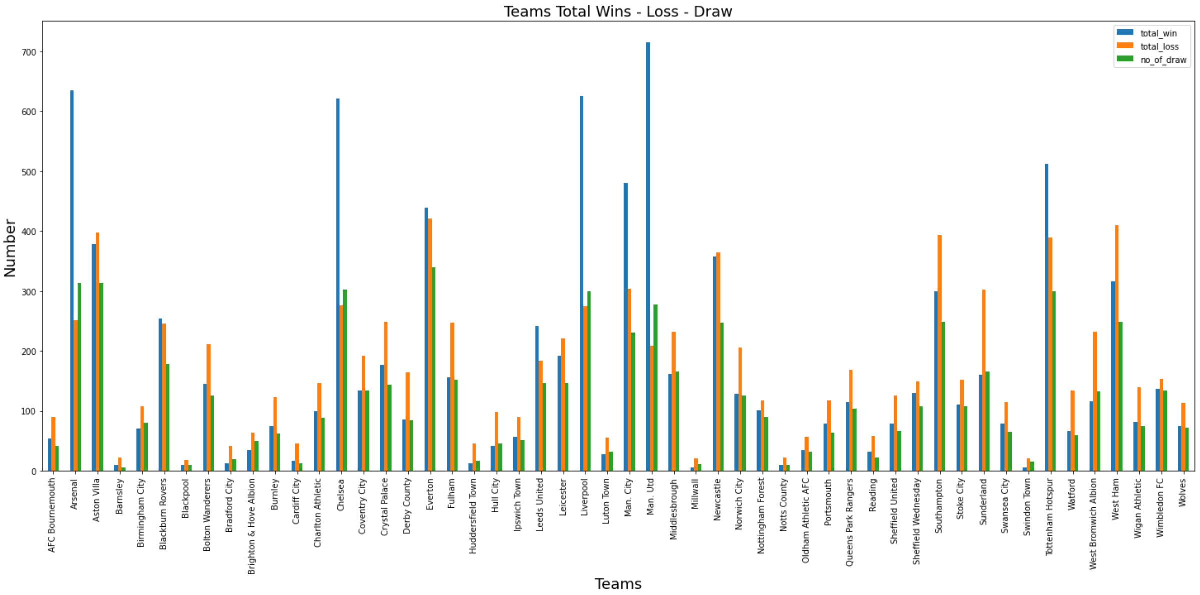


Fig. 14: Team's Overall Performance

The histogram that follows displays the frequency of the number of goals scored by both the home team and the away team.

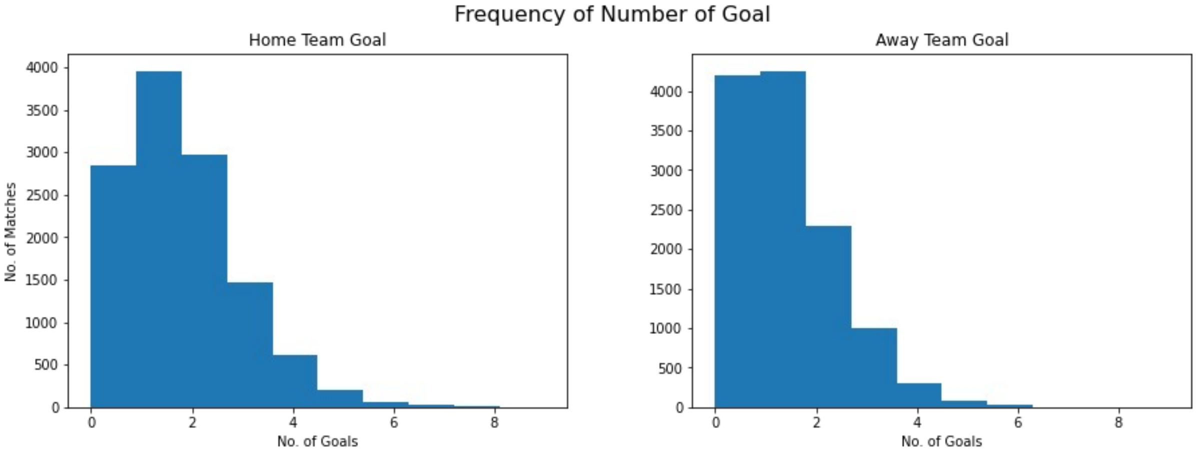


Fig. 15: Histogram for Number of Goals

- It is easy to see that in the majority of matches, the home team scored between 0 and 2 goals, with 1 being the maximum number of goals scored. The majority of the time, the Visitor Team scores either 0 or 1.

5: Baseline Model

In this chapter we will discuss about feature and model implementation.

5.1 Features

Initially we had collected data from 2012 to 2021 and perform EDA on that data. We want to find out which feature is more related and will help in increase accuracy of prediction model.

We have cleaned data set and this data set contains following features :

- 'Home_Team' : Home Team Name
- 'Away_Team' : Away Team Name
- 'Result' : Match Result
- 'Link' : Link of Match
- 'Season' : Year of League
- 'Round' : Week number
- 'League' : Name of League
- 'Home_Team_Goal' : Goal scored by Home Team
- 'Away_Team_Goal' : Goal scored by Away Team
- 'Target' : Match Result (Win, Draw, Loss)
- 'date' : Date of Match
- 'home_yellow' : Number of Yellow Card issued to Home Team
- 'home_red' : Number of Red Card issued to Home Team
- 'away_yellow' : Number of Yellow Card issued to Away Team
- 'away_red' : Number of Red Card issued to Away Team
- 'Ground_Name' : Name of Ground
- 'Pitch_Type' : Pitch Type
- 'Capacity' : Capacity of Stadium
- 'Home_ELO' : ELO rating of Home Team
- 'Away_ELO' : ELO rating of Away Team
- 'time' : Time of Match
- 'day_code' : Week day

From above features some of not usable or duplicating the information. So, we have dropped those column using drop() method of pandas.

```
data = data.drop(['League', 'Link', 'Season', 'Result', 'Round', 'Target',
'date', 'day_code', 'Ground_Name', 'time'],axis= 1)
```

Also we have calculated Target Column ‘Outcome’ based on Goal scored by Home Team and Away Team. ‘Outcome’ columns will be used as Target column for baseline model.

Following is pairplot with respect to Target column ‘Outcome’.

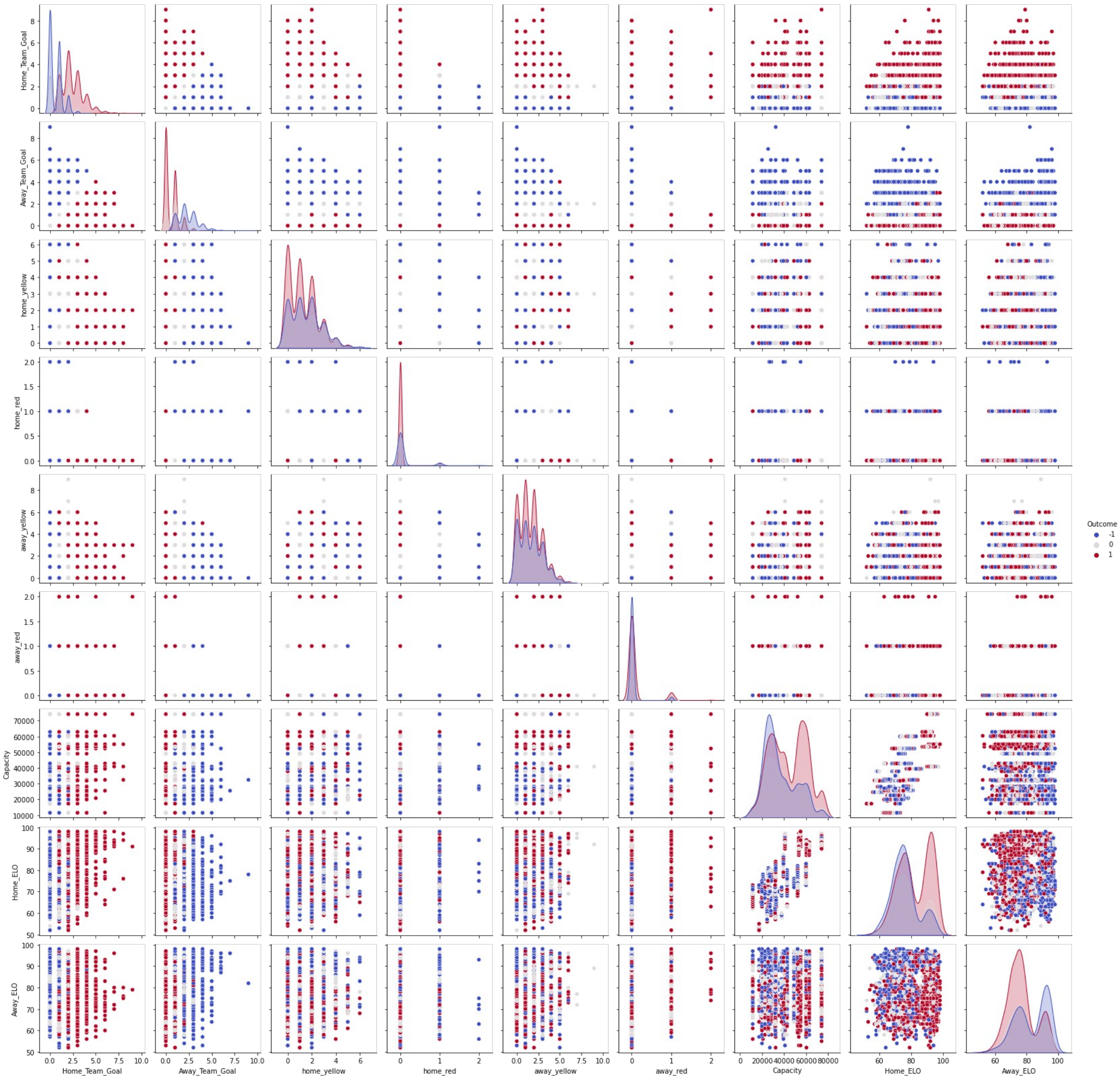


Fig. 16: Pair plot of data

And following graph shows correlation among variable.

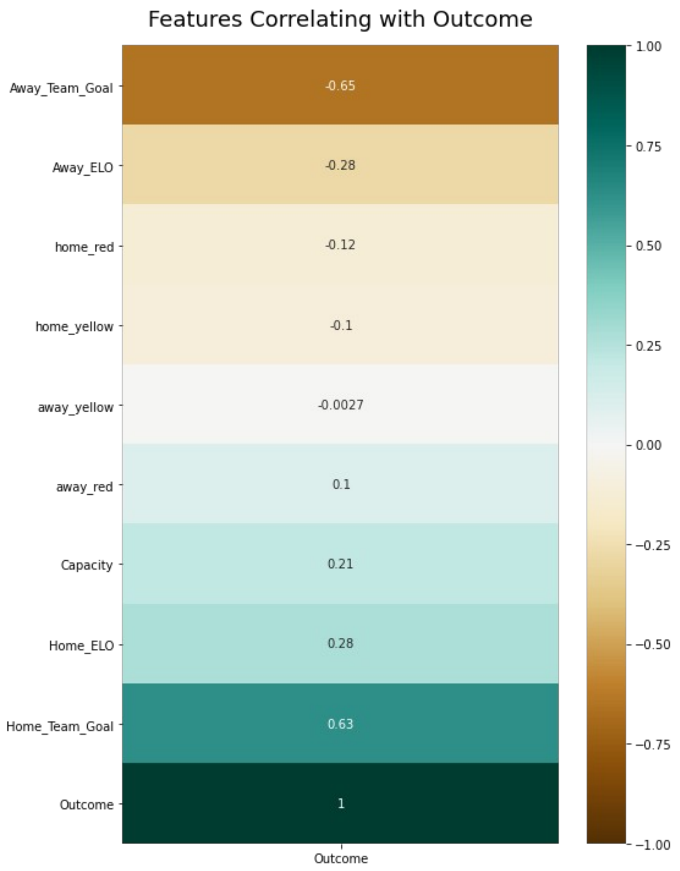


Fig. 17: Correlation Graph

Based on this graph, we are able to infer that the information regarding yellow and red cards has a weaker correlation with the final score of the match. Even our team will not be knowledgeable on yellow and red cards before to the game. In addition to that, we computed some features depending on the data that we had.

- **Last_5_Home_Team_avgGoal** : Average Goal of Home Team from last 5 match
- **Last_5_Away_Team_avgGoal** : Average Goal of Away Team from last 5 match
- **Last_5_Home_Team_Home_avgGoal** : Average Goal of Home Team when they played at Home in Last 5 match
- **Last_5_Away_Team_Away_avgGoal** : Average Goal of Away Team when they played at Away in Last 5 match
- **Last_5_Home_Team_All_Streak** : Home Team's Last 5 Result(sum).
- **Last_5_Away_Team_All_Streak** : Away Team's Last 5 Result(sum).
- **Last_5_Home_Team_Home_Streak** : Home Team's Last 5 match result when played at home
- **Last_5_Away_Team_Away_Streak** : Away Team's Last 5 match result when played at away

We have removed row containing NULL values after adding features.

And using LableEncoder module of sklearn.preprocessing, we have encoded categorical features.

5.2 Base line Model

We have created to set of Data set.

1. X1 data set comprises all 20 features
2. The X2 data set contains information that does not include the columns labelled home_yellow, home_red, away_yellow, away_red, Pitch_Type and Capacity.

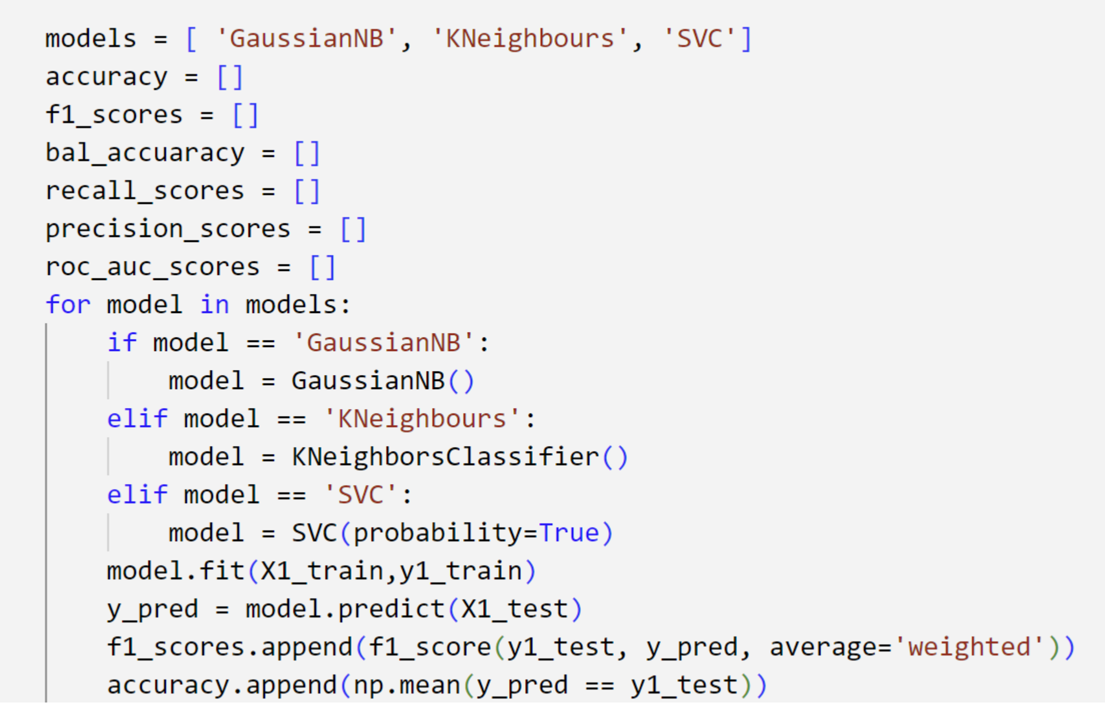
Using train_test_split, we have separated the two sets of data into a training set and a testing set.

The training set has 80 percent of the rows, while the testing set only has 20 percent of the data.

```
X1_train, X1_test, y1_train, y1_test = train_test_split(X1, y, test_size =
0.20,random_state=42)
```

```
X2_train, X2_test, y2_train, y2_test = train_test_split(X2, y, test_size =
0.20,random_state=42)
```

We have implemented GaussianNB, KNeighbours and Support Vector Classifier(SVC) model with default parameter using Sklearn module on both training data sets And also calculate its evaluation metrics.



```
models = [ 'GaussianNB', 'KNeighbours', 'SVC']
accuracy = []
f1_scores = []
bal_accuaracy = []
recall_scores = []
precision_scores = []
roc_auc_scores = []
for model in models:
    if model == 'GaussianNB':
        model = GaussianNB()
    elif model == 'KNeighbours':
        model = KNeighborsClassifier()
    elif model == 'SVC':
        model = SVC(probability=True)
    model.fit(X1_train,y1_train)
    y_pred = model.predict(X1_test)
    f1_scores.append(f1_score(y1_test, y_pred, average='weighted'))
    accuracy.append(np.mean(y_pred == y1_test))
```

Fig. 18: Implementation of Base Models

The following is a table that compares several metrics of models based on their performance on testing data.

Model	Accuracy	F1_score
GaussianNB	0.712012	0.709676
KNeighbours	0.497829	0.481069
SVC	0.454414	0.367682

Table 1: Base model Accuracy(X1 dataset)

In the same manner we get following accuracy on X2 data set.

Model	Accuracy	F1_score
GaussianNB	0.746744	0.748221
KNeighbours	0.535456	0.515414
SVC	0.604920	0.512565

Table 2: Base model Accuracy(X2 dataset)

It is clear from looking at both tables that the accuracy of the models improved when they were trained using the X2 data set, which omitted the information about the Yellow and Red cards. As a result, going forward in this project, we won't be making use of such features for optimizing models.

6: Testing & Optimization techniques

In this chapter, we will discuss the experiments that we carried out in order to improve the overall performance of our model and achieve the best results possible.

Machine learning models are only useful if they can **generalise** to make good predictions on unseen examples. To estimate how well a model will perform on unseen data, we split our initial dataset into two different sets. One is for training on, and the other is for testing on. The testing set is used for evaluating whether a model meets our requirements and estimating real world performance. That is all. It is not for making choices about our model.

Sklearn provides a method `train_test_split()` in it's `model_selection` module to split our data.

6.1 Testing

In this section, we will examine a variety of testing methodologies and metrics that have served as a foundation for us in developing and optimizing our model.

6.1.1 Accuracy

In any classification problem, we aim to predict the category of a given observation based on the general properties of a training data set.

In this context, the simplest way to measure performance, whether with binary or multiclass classification, is to measure the number of correct predictions out of all the whole dataset.

Accuracy - percentage of how many predictions were accurate out of all predictions

6.1.2 Confusion Matrix

A confusion matrix is a table that is used to define the performance of a classification algorithm. The rows correspond to the true classes and the columns correspond to the predicted classes. Each entry counts how often a sample that belongs to the class corresponding to the row was classified as the class corresponding to the column. Entries on the main diagonal of the confusion matrix correspond to correct classifications, while other entries tell us how many samples of one class got mistakenly classified as another class.

One can define accuracy using confusion matrix as well.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

True positive(TP): Where the model predicts the label to be 'Positive' and the true label is 'Positive'

True negative(TN): Where the model predicts the label to be 'Negative' and the true label is 'Negative'

False positive(FP): Where the model predicts the label to be 'Positive' and the true label is 'Negative'

False negative(FN): Where the model predicts the label to be 'Negative' and the true label is 'Positive'

In other words, accuracy is the number of correct predictions (TP and TN) divided by the number of all samples (all entries of the confusion matrix summed up).

6.1.3 Precision

Precision is another commonly used evaluation metric. Precision is a ratio of correctly predicted positives to the total number of predicted positives.

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{TP}}$$

Perfect recall is when we have no false positives. So it is a useful metric to consider when false positives are costly. It is an important measure to have to ensure we evaluate the performance of our model appropriately.

6.1.4 Recall

Another useful evaluation metric for classification models is the recall, also known as **sensitivity**. Recall is the ratio of the correctly predicted positives to the total number of positives in the dataset. It is given by the following equation:

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}$$

Perfect recall is when we have no false negatives, so it is a useful metric to consider when false negatives are costly.

While precision measures how well our model deals with false positives, recall measures how well it deals with false negatives

Together, accuracy, recall and precision are robust metrics for performance evaluation.

6.1.5 F1 Score

The F1 score, also known as the F-score or F-measure, is a metric that takes what is known as the harmonic mean of our precision and recall.

Therefore by maximising the F1-score, we are accounting for both precision AND recall simultaneously. It is given as follows:

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{\text{TP}}{\text{TP}+\frac{1}{2}(\text{FP}+\text{FN})}$$

6.2 Kelly Betting :

In this part of the article, we are going to discuss Kelly betting methods to compare our model.

The Kelly criterion is a mathematical formula that was developed by John L. Kelly Jr. [27] while he was working at AT&T's Bell Laboratories. The formula is related to the growth of capital over a long period of time. It is used to determine how much money should be invested in a particular asset in order to achieve the greatest possible increase in one's wealth over a period of time. It decides how much money should be wagered on each round. In order to achieve an optimal level of expected value, it makes use of the fundamental concepts underlying probability theory.

Odd : Probability of an event occurring, divided by the probability that the event will not take place, is one way to define odds.

To begin, we have determined the likelihood of a particular outcome based on the test data by applying the `predic_proba` method of classification technique. With the help of the following formula, we were able to convert this probability into odds.

odd = $\dfrac{probabilty}{1 - proababilty}$

The Kelly Criterion determines the amount of money that must be betted on each individual bet.

bet_amount = $\dfrac{(probabilty\ *\ odd) - (1 - probabilty)}{odd}$

- probability = the probability of winning
- odd = odd of that match

This formula will tell you how much of your total money you should put on each bet based on the probability of you winning that particular bet.

7: Implementation & Evaluation

In this chapter, we will cover how we have implemented several Classification and Regression techniques for the Outcome, Goal Difference, and Final Goal Score models. In addition to that, we analyse and contrast the evaluation of several machine learning techniques.

Following the selection of the basis feature, we gathered and cleaned the data for the seasons ranging from 1990 to 2021.

7.1 Outcome Model

In this model we have tried to predict match outcome as Win for Home Team, Draw and Loss for Home Team. We didn't gave weight on Home Team advantage in this research.

7.1.1 Features and Model

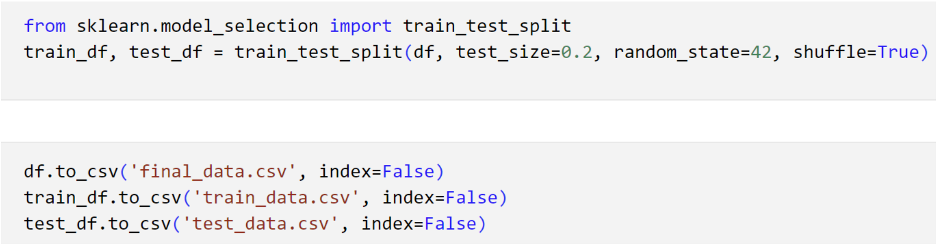
For Outcome prediction model we have added following features apart from Home_Team_Goal, Away_Team_Goal, Home_ELO, Away_ELO.

- Last_5_Home_Team_avgGoal – Average Goal scored in last 5 Match by Home Team
- Last_5_Away_Team_avgGoal - Average Goal scored in last 5 Match by Away Team
- Last_5_Home_Team_Home_avgGoal - Average Goal scored in last 5 Match by Home Team in Home Ground
- Last_5_Away_Team_Away_avgGoal - Average Goal scored in last 5 Match by Away Team when playing away
- Last_5_Home_Team_All_Streak – Home Team's performance in last 5 Match
- Last_5_Away_Team_All_Streak – Away Team's performance in last 5 Match
- Last_5_Home_Team_Home_Streak – Home Team's performance in last 5 Match at Home
- Last_5_Away_Team_Away_Streak – Away Team's performance in last 5 Match at Away
- Last_3_same_team_home_goal – Average Goal scored by Home Team when playing against same opponent in last 3 Match
- Last_3_same_team_away_goal – Average Goal scored by Away Team when playing against same opponent in last 3 Match
- Last_3_same_team_outcome – Home Team's performance in last 3 Match against same Team
- Home_Team_Points – Home Team's overall points in that Season
- Away_Team_Points – Away Team's overall points int that Season

Home_Team_Points and Away_Team_Points shows the how team performing that Season.

This are important features as it shows Teams how teams are perform at home or away and when playing with same opponent also. Also we have encoded Categorical data and created labelencoder.pkl which will be used in modelling part.

After that we have split the data into training and testing data set and saved in separate csv files for modelling. We split our data into training and test sets is that we want to measure how well our model generalize to new, previously unseen data.



```
from sklearn.model_selection import train_test_split
train_df, test_df = train_test_split(df, test_size=0.2, random_state=42, shuffle=True)
```

```
df.to_csv('final_data.csv', index=False)
train_df.to_csv('train_data.csv', index=False)
test_df.to_csv('test_data.csv', index=False)
```

Fig. 19: Train and Test data

For implement our model we used StratifiedKFold validation.

StratifiedKFold : Some classification problems can exhibit a large imbalance in the distribution of the target classes. In such cases it is recommended to use stratified sampling as implemented in StratifiedKFold and StratifiedShuffleSplit to ensure that relative class frequencies is approximately preserved in each train and validation fold.

StratifiedKFold is a variation of k-fold which returns stratified folds: each set contains approximately the same percentage of samples of each target class as the complete set.

GridSearchCV : Grid Search consists of grid of possible hyperparameters, and each combination is used to learn algorithm of choice and validate the results.

Sklearn provide `GridSearchCv` in `model_selection` module.
We have trained model on following algorithms:

1. GaussianNB

2. Decision Tree Classifier

3. KNeighborsClassifier :

4. AdaBoost Classifier

5. Logistic Regression

6. SVC(support vector classifier)

7. Random Forest Classifier

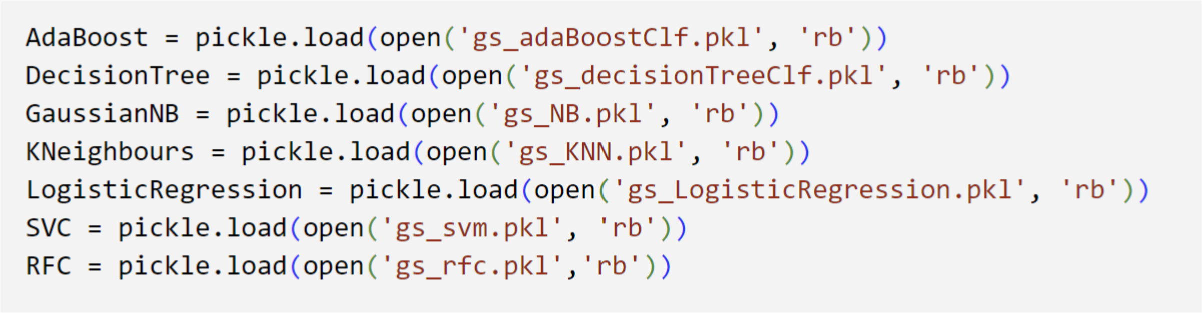
Initially we have build basemodel with default parameter and after using GridSearchCV we have tuned model. Also we have saved those models as pkl files in local system for testing experiment.

7.1.2 Model Comparision

Metric Accuracy :

In this part of the discussion, we will evaluate our final models based on various of performance metrics.

Using the Pickle library, we have loaded all previously saved models for comparison.

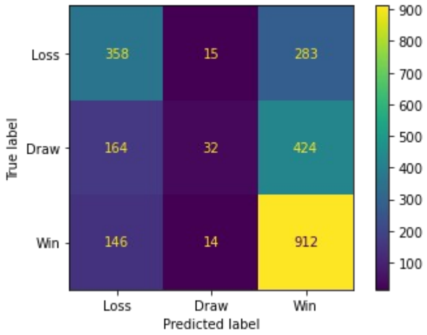


```
AdaBoost = pickle.load(open('gs_adaBoostClf.pkl', 'rb'))
DecisionTree = pickle.load(open('gs_decisionTreeClf.pkl', 'rb'))
GaussianNB = pickle.load(open('gs_NB.pkl', 'rb'))
KNeighbours = pickle.load(open('gs_KNN.pkl', 'rb'))
LogisticRegression = pickle.load(open('gs_LogisticRegression.pkl', 'rb'))
SVC = pickle.load(open('gs_svm.pkl', 'rb'))
RFC = pickle.load(open('gs_rfc.pkl','rb'))
```

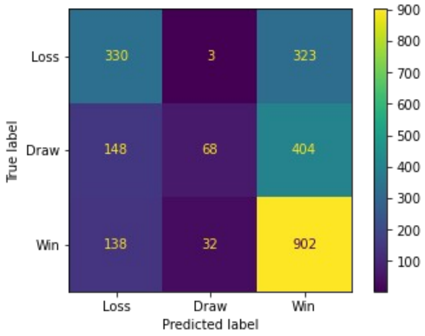
Fig. 20: Snippet for Model Loading

We have test this model on Testing data. We get confusion metrics as shown below.

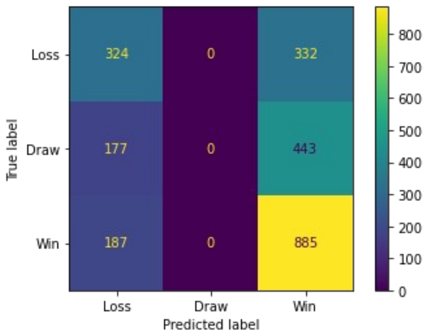
AdaBoost :



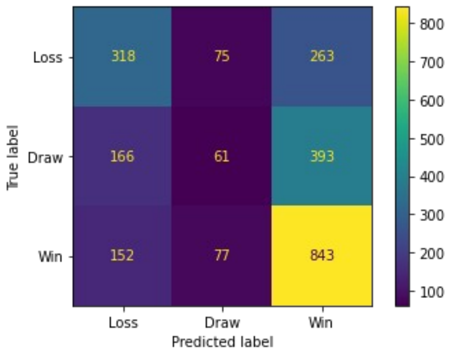
DecisionTreeClassifier:



GaussianNB :

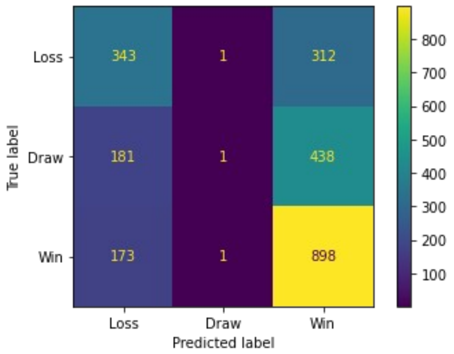


KNeighborsClassifier :



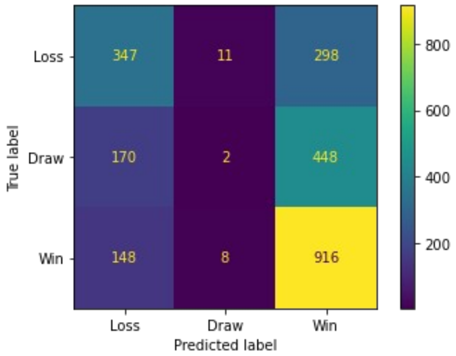
	Loss	Draw	Win
Loss	318	75	263
Draw	166	61	393
Win	152	77	843

Logistic Regression :



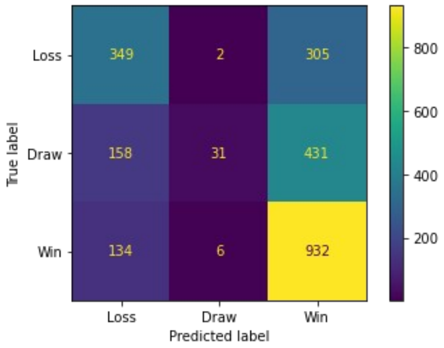
	Loss	Draw	Win
Loss	343	1	312
Draw	181	1	438
Win	173	1	898

SVM :



	Loss	Draw	Win
Loss	347	11	298
Draw	170	2	448
Win	148	8	916

RFC :



	Loss	Draw	Win
Loss	349	2	305
Draw	158	31	431
Win	134	6	932

The following table provides a comparison of the various metrics across the models.

Model	Accuracy	F1_score	Recall_scores	Bal_accuracy	Precision_score	Roc_Auc_score
AdaBoost	0.554514	0.485366	0.482697	0.482697	0.541276	0.674881
DecisionTree	0.553663	0.499571	0.484715	0.484715	0.583207	0.705062
GaussianNB	0.514906	0.430498	0.439821	0.439821	0.334688	0.651341
KNeighbours	0.520443	0.475604	0.456508	0.456508	0.449587	0.654365
LogisticRegression	0.531090	0.444748	0.455383	0.455383	0.347708	0.649180
SVC	0.538756	0.454357	0.462222	0.462222	0.389395	0.653994
RFC	0.558773	0.485792	0.483805	0.483805	0.632696	0.715275

Table 3: Metrics for Outcome Model

This indicates that our model is accurate in predicting the result of more than fifty percent of the games it is applied to.

In Figure, we can see a comparison of the classification accuracy for our final models.

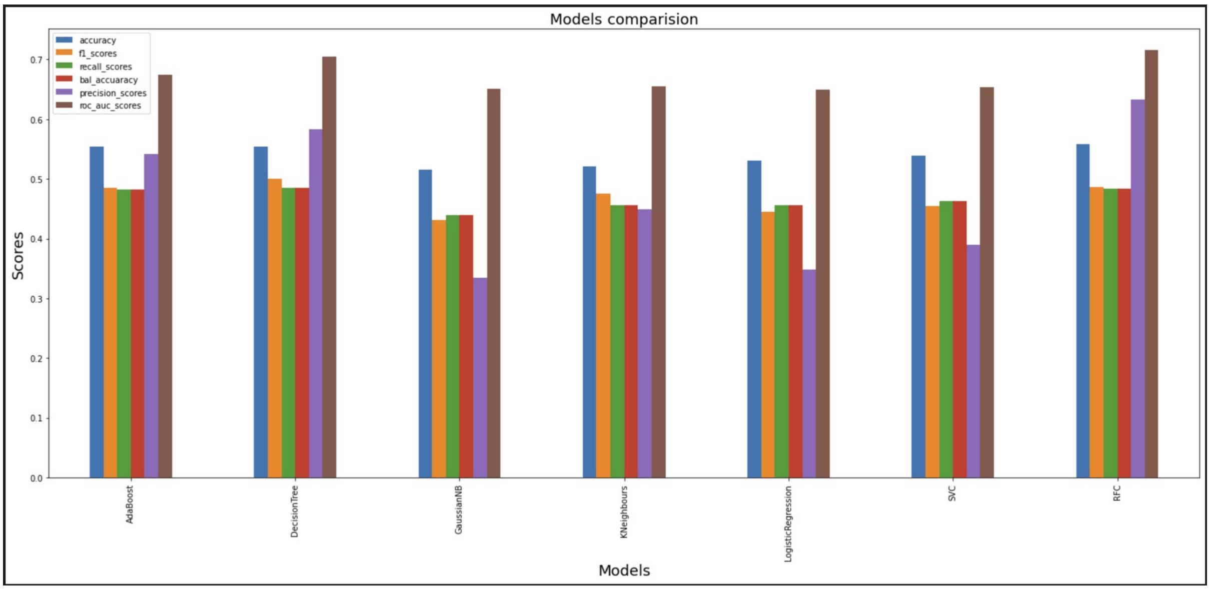
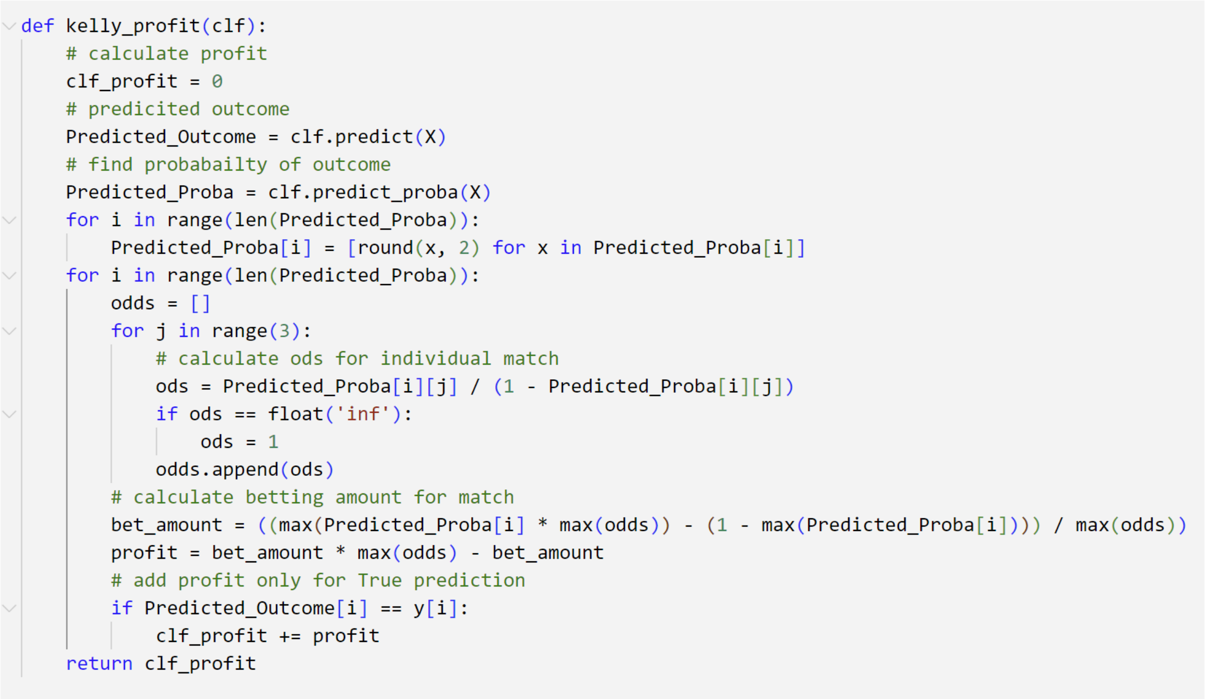


Fig. 21 : Outcome Model Metrics Comparison Graph

We can see that all models have accuracy between 0.52 to 0.56. Random Forest Classifier have highest accuracy as 0.5587. Also, RFC model have high precision score 0.6326. So for Outcome Prediction model RFC is best model among others based on accuracy parameter.

Kelly Betting :

The formula for determining profit based on the machine learning model is displayed in the following figure.



```python
def kelly_profit(clf):
    # calculate profit
    clf_profit = 0
    # predicited outcome
    Predicted_Outcome = clf.predict(X)
    # find probabailty of outcome
    Predicted_Proba = clf.predict_proba(X)
    for i in range(len(Predicted_Proba)):
        Predicted_Proba[i] = [round(x, 2) for x in Predicted_Proba[i]]
    for i in range(len(Predicted_Proba)):
        odds = []
        for j in range(3):
            # calculate ods for individual match
            ods = Predicted_Proba[i][j] / (1 - Predicted_Proba[i][j])
            if ods == float('inf'):
                ods = 1
            odds.append(ods)
        # calculate betting amount for match
        bet_amount = ((max(Predicted_Proba[i] * max(odds)) - (1 - max(Predicted_Proba[i]))) / max(odds))
        profit = bet_amount * max(odds) - bet_amount
        # add profit only for True prediction
        if Predicted_Outcome[i] == y[i]:
            clf_profit += profit
    return clf_profit
```

Fig. 22: Snippet for Kelly Betting

Using the models that were imported before, we were able to determine the profit for each model, and the chart that follows presents a comparison of the various models.

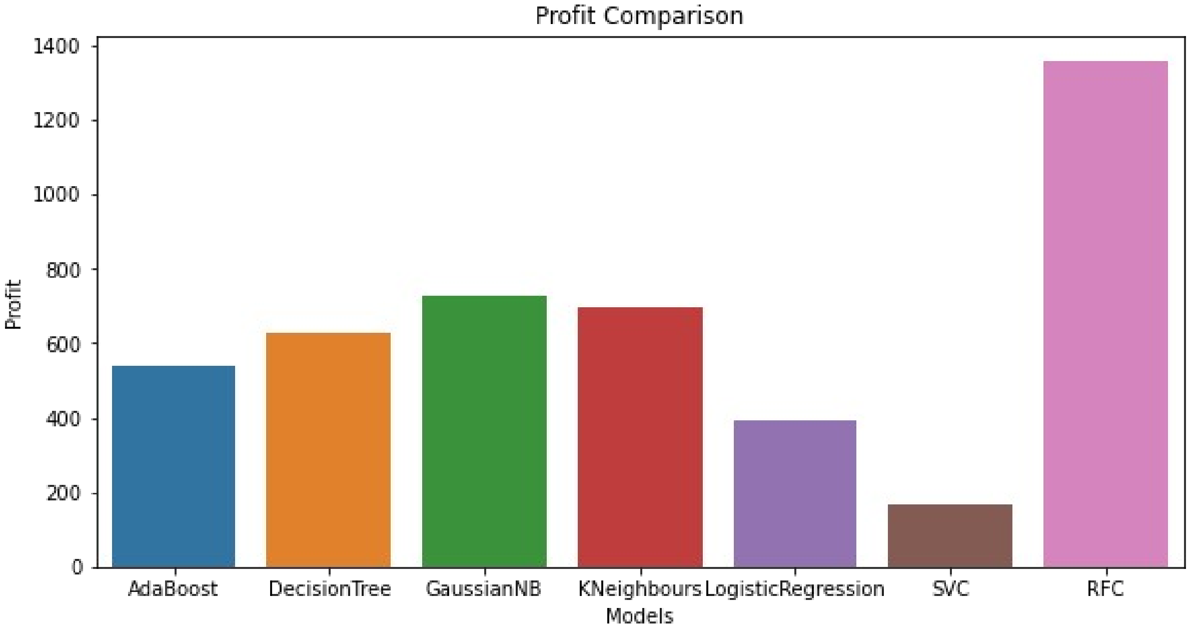


Fig. 23: Profit Comparision based on Kelly Betting

When compared to the other Sklearn models, it is clearly evident that the RFC (Random Forest Classifier) model generates the highest profit.

7.2 Goal Difference

In this section we will discuss about features, model implementation, and testing method for Goal Difference model.

7.2.1 Features and Model

For Goal Difference model we have calculated out target column Goal_Difference from Home and Away Team goal.

Feature :

We have consider following feature for this model.

- Last_5_Home_Team_avgGoal : Average Goal scored in last 5 Match by Home Team
- Last_5_Away_Team_avgGoal : Average Goal scored in last 5 Match by Away Team
- Last_5_Home_Team_Home_avgGoal : Average Goal scored in last 5 Match by Home Team in Home Grounds
- Last_5_Away_Team_Away_avgGoal : Average Goal scored in last 5 Match by Away Team when playing away
- Last_5_Home_Team_Home_avgGD : Average Goal Difference in last 5 Match of Home Team in Home Ground
- Last_5_Away_Team_Away_avgGD : Average Goal Difference in last 5 Match of Away Team in Away Ground
- Last_5_Home_Team_All_avgGD : Average Goal Difference in last 5 Match of Home Team
- Last_5_Away_Team_All_avgGD : Average Goal Difference in last 5 Match of Away Team
- Last_3_same_team_home_avgGD : Average Goal Difference in last 3 Match by Home Team against same opponent playing at Home
- Last_3_same_team_avgGD : Average Goal Difference in last 3 Match of Home Team against same opponent playing at home or away
- Home_Team_Points : Points gained by Home Team in that Season
- Away_Team_Points : Points gained by Away Team in that Season
- Goal_Difference : Goal difference

We have implement following algorithm to predict Goal Difference

1. GaussianNB

2. Decision Tree Classifier

3. KNeighborsClassifier :

4. AdaBoost Classifier

5. Logistic Regression

6. SVC(support vector classifier)

7. Random Forest Classifier

7.2.2 Model Comparision

Metric Accuracy :

Following table shows metrics comparison among Machine learning technique.

Model	Accuracy	F1_score	Recall_scores	Bal_accuracy	Precision_score
AdaBoost	0.317753	0.231901	0.182416	0.182416	0.135595
DecisionTree	0.311910	0.252575	0.192459	0.192459	0.187644
GaussianNB	0.280000	0.215848	0.164454	0.164454	0.159610
KNeighbours	0.265169	0.225922	0.168451	0.168451	0.218052
LogisticRegression	0.280449	0.214074	0.164302	0.164302	0.163389
SVC	0.287640	0.207756	0.164566	0.164566	0.196178
RFC	0.310112	0.232709	0.178867	0.178867	0.173024

Table 4: Metrics for Goal Difference Model

AdaBoost, Decision Tree and RFC algorithm have accuracy around 31% . And, we got highest accuracy in AdaBoost classification, which is 0.31773 . While Precision score is high in KNeighbours classification method around 0.218052.

So, based on Accuracy AdaBoost is best model to predict Goal Difference in this research.

Kelly Betting:

For Kelly betting testing, we have used same method used for Goal Difference prediction.

Following is bar graph to show profit comparison between models.

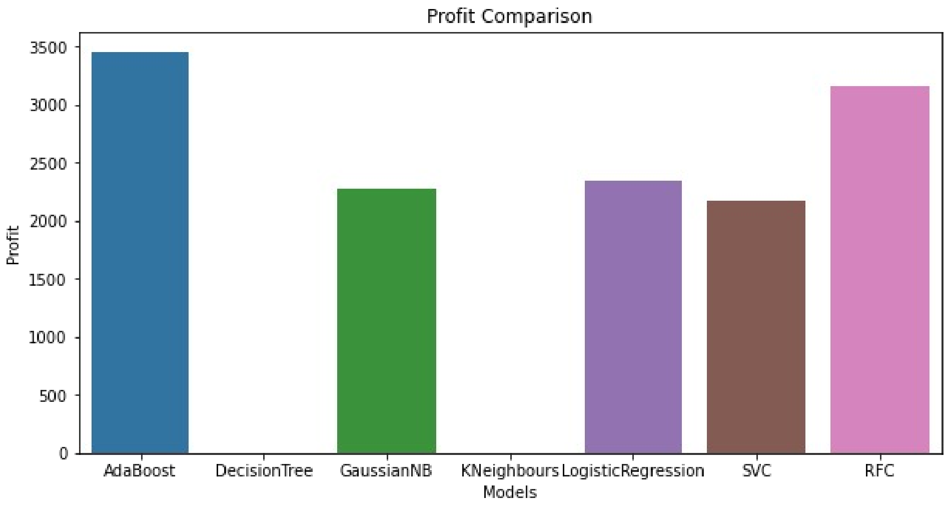


Fig. 24: Profit comparision based on Kelly Betting for Goal Difference Model

We can see the AdaBoost earn highest profit than other models so as per Kelly betting testing AdaBoost is best model.

7.3 Total Goals

In this section we will discuss about features, model implementation, and testing method for Total Goals prediction model.

7.3.1 Features and Models :

We have computed our target column Total_Goals for the Total Goal prediction model by adding up the goals scored by both the Home Team and the Away Team.

Feature :

We have consider following feature for this model.

- Last_5_Home_Team_avgGoal : Average Goal scored in last 5 Match by Home Team
- Last_5_Away_Team_avgGoal : Average Goal scored in last 5 Match by Away Team
- Last_5_Home_Team_Home_avgGoal : Average Goal scored in last 5 Match by Home Team in Home Grounds
- Last_5_Away_Team_Away_avgGoal : Average Goal scored in last 5 Match by Away Team when playing away
- Last_5_Home_Team_Home_avgGT : Average Goal Total in last 5 Match of Home Team in Home Ground
- Last_5_Away_Team_Away_avgGT : Average Goal Total in last 5 Match of Away Team in Away Ground
- Last_5_Home_Team_All_avgGT : Average Goal Total in last 5 Match of Home Team
- Last_5_Away_Team_All_avgGT : Average Goal Total in last 5 Match of Away Team
- Last_3_same_team_home_avgGT : Average Goal Total in last 3 Match by Home Team against same opponent playing at Home
- Last_3_same_team_avgGT : Average Goal Total in last 3 Match of Home Team against same opponent playing at home or away
- Home_Team_Points : Points gained by Home Team in that Season
- Away_Team_Points : Points gained by Away Team in that Season
- Total_Goals: Total Goals scored in the Match

Model :

We have implement following ML technique to predict Total Goals of the match.

1. GaussianNB
2. KNeighbours
3. Super Vector Machine
4. Random Forest
5. Decision Tree
6. AdaBoost
7. Logistic Regression

7.3.2 Model Comparision

Metric Accuracy :

Following table shows metrics comparision among Machine learning technique.

Model	Accuracy	F1_score	Recall_scores	Bal_accuracy	Precision_score
AdaBoost	0.249148	0.117719	0.102366	0.102366	0.048966
DecisionTree	0.254685	0.139604	0.106204	0.106204	0.072326
GaussianNB	0.245315	0.096649	0.100000	0.100000	0.024532
KNeighbours	0.209540	0.175663	0.098803	0.098803	0.092671

LogisticRegression	0.247871	0.127253	0.102462	0.102462	0.047685
SVC	0.245315	0.096649	0.1	0.1	0.024532
RFC	0.240630	0.131242	0.100171	0.100171	0.077103

Table 5: Metrics for Total Goal Predict Model

We were able to forecast about 25 percent of the testing data correctly in the Total Goal Prediction Model.

Based on the various metrics and parameters, the DecisionTree model had the highest accuracy of 0.254685. While the KNeighbours model can only predict with an accuracy of 0.20954, which is the lowest of all the models. So, DecisionTree is the model that performs the best overall in terms of accuracy.

Kelly Betting :

We have calculated the profit based on the Kelly betting strategy.

The subsequent bar graph presents a comparison of earnings obtained from the various business types.

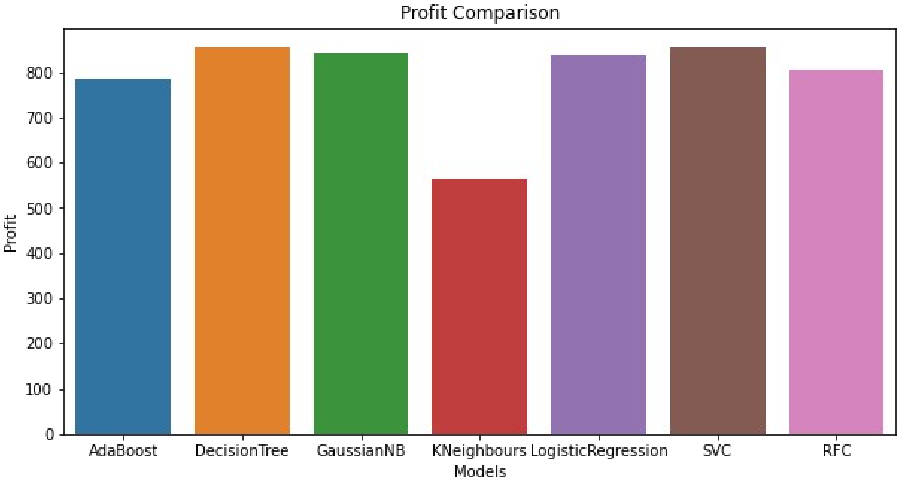


Fig. 25: Profit comparision based on Kelly Betting for Total Goal Model

We can see from the graph that all of the models, with the exception of KNeighbours, produce a profit of approximately 800 units.

Decision Tree and SVC, both models have produced almost the same profit. The Decision Tree model produced 855 units of profit, whereas the SVC model produced 853 units.

According to the results of this testing experiment, the DecisionTree model is the most effective one.

8. Conclusion and Future work

8.1 Summary

Our primary objective of developing a model for predicting output, Goal Difference, and Total Goals by experimenting with various Machine Learning techniques has been attained. In fact, we used contemporary Machine Learning algorithms such as Random Forest, KNeighbors, Decision Tree, GaussianNB, AdaBoost, and Support Vector Machines to forecast match outcomes and scores. We were able to locate and enhance a database with sufficient information to calculate team points and other attributes using scraped data. Using GridSearchCV, a model has completed training and testing, and multiple hypotheses have been tested. We have also evaluated several ways on test data. We found that our model can predict more than 50 percent match outcomes.

8.2 Challenges & Solutions

8.2.1 Finding data

Finding appropriate data to employ in the construction of our model was one of the project's greatest obstacles. A significant amount of time was spent conducting research to identify public databases that allowed us to locate data on a website used for this project. Which provides a vast amount of data from the 1990 Season. After becoming familiar with common Web scraping techniques, we scraped Premier League data using Beautiful Soup.

8.2.2 Testing Experiment

We have used metrics like accuracy, precision, recall, and F1 score to evaluate models. We are also thankful that our supervisor Professor James Orwell for suggesting Kelly betting technique to test data.

8.3 Future extensions

There are many directions in which this project could be taken with more time and resources.

8.3.1 Improved data

We could have gathered additional data about match insights and incorporated it into our algorithm. Additionally, we could have gathered information on other leagues and used it to make predictions. In addition, we were unaware of the geographical location and pass types. We may have also gathered betting odds from additional prominent sources and compared them to our algorithms.

8.3.2 Improve Model

We are able to achieve 55% accuracy for Outcome prediction model. But for Goal Difference prediction model we only able to get 30% accuracy so there is lot can improve in that using depth data set and more records. Same for Total Goal Prediction we achieve 30% accuracy, where there is lot room for improvement. We could implement other classification technique like TensorFlow.

8.3.3 Betting Odds

This experiment could be expanded in the future by examining betting odds to determine whether our model can offer good value bets and effective betting strategies that yield long-term profits.

Bibliography

[1] M. J. Moroney, “Facts from figures, 3rd edn Penguin,” London, 1956.

[2] C. Reep, “ Skill and chance in ball games. Journal of the Royal Statistical Society,” 1971.

[3] M. J. Maher, “Modelling association football scores, Statistica Neerlandica,” 1982.

[4] I. Hill, “Association football and statistical inference, Applied Statistics,” 1974, p. 16.

[5] M.J. Dixon, S.C. Coles, “Modelling association football scores and inefficiencies in the football betting market, Applied Statistics,” 1997, p. 17.

[6] H. Rue, O. Salvesen, “Prediction and retrospective analysis of soccer matches in a league, Statistician,” 2000, p. 18.

[7] Forrest, David, and Robert Simmons, “Outcome Uncertainty and Attendance Demand in Sport: The Case of English Soccer,” 2002.

[8] J. Goddard, “Regression models for forecasting goals and match results in association football,” *International Journal of Forecasting,* 2005.

[9] A. Adam, “Generalised linear model for football matches prediction. KULeuven,” 2016.

[10] D. Forrest, R. Simmons, “Forecasting sport: The behaviour and performance of football tipsters, International Journal of Forecasting,” p. 18, 2000.

[11] T. Kuypers, “Information and efficiency: An empirical study of a fixed odds betting market. Applied Economics,” 2000.

[12] B. Hamadani., “Predicting the outcome of NFL games using machine learning. Stanford University,” 2006.

[13] M. Tavakol, H. Zafartavanaelmi and U. Brefeld, “Feature Extraction and Aggrega tion for Predicting the Euro 2016, Leuphana University of Luneburg,” 2016.

[14] S. Kampakis, W. Thomas, “Using Machine Learning to Predict the Outcome of English County twenty over Cricket Matches. University College London”.

[15] N. Tax, Y. Joustra, “Predicting The Dutch Football Competition Using Public Data: A Machine Learning Approach. Transactions on Knowledge and Data Engineering,” 2015.

[16] A. Joseph, N.E. Fenton, M. Neil., “Predicting football results using Bayesian nets and other machine learning techniques. Knowledge-Based Systems,” 2006.

[17] R. Balla, “Soccer Match Result Prediction using Neural Networks”.

[18] J. Kahn, “Neural network prediction of NFL football games, World Wide Web Electronic Publication,” 2003.

[19] Hucaljuk, J.: Rakipovic,A., “Predicting football scores using machine learning techniques. MIPRO, 2011 Proceedings of the 34th International Convention,” 2011.

[20] Karlis, Dimitris & Ntzoufras, Ioannis, “Robust fitting of football prediction models. IMA Journal of Management Mathematics.,” 2011.

[21] Darwin, P & Dra, H, "Predicting Football Match Results with Logistic Regression. International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA)," 2016.

[22] A. Elo, The rating of chessplayers, past and present, Arco Publishing, 1978.

[23] J. Lasek., "Euro 2016 Predictions Using Team Rating Systems," 2016.

[24] Lars Magnus Hvattuma, Halvard Arntzen, "Using ELO ratings for match result prediction in association football. International Journal of Forecasting," 2010.

[25] Anthony Costa Constantinou, Norman Elliott Fenton, "Determining the level of ability of football teams by dynamic ratings based on the relative discrepanciesin scores between adversaries.," *Journal of Quantitative Analysis in Sport,* p. 21, 2013.

[26] Harm Eggels, Ruud van Elk, Mykola Pechenizkiy, "Explaining soccer match outcomes with goal scoring opportunities predictive analytics. Eindhoven Universityof Technology," 2016.

[27] J. L. J. Kelly, "A New Interpretation of Information Rate, Bell," 1956.