VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering
VIT Chennai
Vandalur - Kelambakkam Road, Chennai - 600 127

PROJECT REPORT

Programme: CSE

Course Code: CSE3506

Course Name: Essentials of Data Analytics

Slot: G2

Faculty: Dr. Asnath Phamila

Title: Football Match Prediction

Arnav Singh 20BCE1731
Archit Agarwal 20BCE1773
Gautam Adlakha 20BCE1227
Yogesh Harlalka 20BCE1696

ABSTRACT

The goal of this project is to develop a prediction system that can accurately predict whether a home team will win or not in a football match. We will use different machine learning algorithms—SVM, Logistic Regression, Random Forest, XGBoost, Naïve Bayes, LDA, QDA, and Neural Network—to develop our prediction system. We will use a dataset of past football matches with associated outcomes to train and test our models. The dataset contains various features such as the team names, the date of the match, the venue, and various statistics related to the teams and their players. We will preprocess the data and perform feature engineering to extract relevant features that can help us predict the outcome of a match. We will then train our models using the pre-processed data and evaluate their performance using various metrics such as accuracy, precision, recall, F1 score, and area under the curve (AUC). We will also perform a statistical analysis to determine the significance of the results. The model with the best performance on the test dataset will be selected as the final model for our prediction system. The prediction system can be used by sports enthusiasts, fans, and gamblers to make informed decisions about placing bets on future football matches. Our prediction system can also be extended to other sports and used for various applications such as player selection, team composition, and strategy development.

Keywords: Prediction System, Machine Learning, SVM, Logistic Regression, Random Forest, XGBoost, Naïve Bayes, LDA, QDA, Neural Network.

INTRODUCTION

Sports betting has always been a popular activity for fans, enthusiasts, and gamblers alike. The thrill of predicting the outcome of a match and the possibility of winning a considerable amount of money make sports betting an exciting and enjoyable pastime for many. However, predicting the outcome of a match is not always straightforward, as there are many factors that can influence the final result. Therefore, the development of prediction systems that can accurately forecast the outcome of a match has gained significant attention in recent years. In this project, we will focus on developing a prediction system that can determine whether a home team will win or not in a football match.

Football is the most popular sport worldwide, and predicting the outcome of a match is a challenging task due to various factors such as team composition, player injuries, home advantage, etc. Therefore, we will use these eight popular machine learning algorithms to develop a prediction system that can predict whether a home team will win its match or not. We will use the following algorithms for this purpose: Support vector machines (SVM), logistic regression, random forest, XGBoost, Naïve Bayes, LDA, QDA, and neural networks. These algorithms are commonly used in machine learning and have been shown to provide accurate predictions for various tasks.

PROBLEM IDENTIFICATION

The prediction of football match outcomes has significant practical applications, such as betting, sports analytics, and team management. However, accurate predictions require the analysis of various factors, including team strength, player form, past performance, and match conditions. Traditional statistical methods and expert opinions can only provide limited accuracy in predicting the outcomes of football matches. Therefore, there is a need for more advanced and accurate prediction models that can leverage the power of machine learning algorithms.

In this project, we aim to develop a prediction system that uses different machine learning algorithms to predict the outcomes of football matches. Our goal is to achieve high accuracy and performance while considering various factors that can affect the outcome of the match. This project will contribute to the field of sports analytics by providing a more advanced and accurate prediction system for football match outcomes.

BACKGROUND STUDY

Football is one of the most popular sports in the world, with millions of fans and enthusiasts following the game. Predicting the outcome of a football match has always been a challenge due to the complexity of the game, the various factors that can influence the result, and the unpredictability of the players' performances. Over the years, many prediction systems have been developed to forecast the results of upcoming matches, and machine learning algorithms have played a vital role in this field.

Support Vector Machine (SVM), Logistic Regression, Random Forest, XGBoost, Naïve Bayes, LDA, QDA, and Neural Network are widely used machine learning algorithms that have been successfully applied to various prediction tasks. SVM is a binary classification algorithm that uses a hyperplane to separate the data points into different classes. Logistic regression is another binary classification algorithm that models the probability of a binary outcome based on the input variables. Random Forest is an ensemble learning algorithm that builds multiple decision trees and combines their predictions to improve accuracy. XGBoost is a gradient boosting algorithm that uses decision trees as base models and optimises them through gradient descent. Naïve Bayes is a probabilistic algorithm that calculates the probability of a hypothesis being true based on input features using Bayes' theorem. LDA is a generative statistical model that learns the underlying structure of the input data by assuming a distribution for the classes. QDA is another generative statistical model that learns the underlying structure of the input data by assuming different distributions for each class. A Network is a machine learning model inspired by the structure and function of biological neural networks that can learn complex patterns in data.

LITERATURE REVIEW

1. **Utilizing machine learning techniques in Football prediction**

   Link:https://research.thea.ie/bitstream/handle/20.500.12065/4059/Ryan%20Duarte%20MSc.pdf?sequence=1&isAllowed=y

   Summary: The paper discusses the importance of predicting the winner of football matches and how it can benefit football teams and fans. With the help of machine learning techniques and by considering base statistics, psychological, and non-psychological factors, the paper aims to provide an accurate prediction for which team would win and why they won. The author utilized two seasons of Premier League football data to compile a quantitative data set and used eight models to learn and predict the results. The author successfully predicted the results of 72.37% matches, with the best performing model returning 85% accuracy when trained on base statistics and 75% accuracy when trained on additional factors. The predicted final league table closely resembled the real final league table.

2. **Machine Learning Models Reveal Key Performance Metrics of Football Players to Win Matches in Qatar Stars League**

   Link:https://ieeexplore.ieee.org/document/9261335

   Summary: Jassim Almulla and Tanvir Alam proposed a system that analyzed football players' performance in a total of 864 football matches of the Qatar Stars League (QSL) between the years 2012 and 2019. They formulated a classification framework in the machine learning (ML) context to distinguish the winning team from the losing team in a match. Different ML models were considered for this classification task, and the logistic regression-based model was considered the best performing model, with more than 80% accuracy.

3. **Prediction of Winning Team using Machine Learning**

   Link: https://www.ijert.org/research/prediction-of-winning-team-using-machine-learning-IJERTCONV9IS03096.pdf

   Summary: The paper discusses the implementation of machine learning techniques to predict the full-time result of football matches in the English Premier League (EPL) for the purpose of sports betting. The algorithms used were Support Vector Machines, Random Forest, and Naive Bayes, and the dataset was gathered from past seasons. The paper concludes that the Support Vector Machine algorithm had the highest accuracy at 63%, followed by Naive Bayes and Random Forest at 57% and 55%, respectively. The significant variables were found to be the attacking and defensive strengths of the home and away teams, but the accuracy improved when more features such as corners and shots on target were added. The paper emphasizes the relevance of recent season data for accurate predictions.

4. **Machine Learning Enabled Team Performance Analysis in the Dynamical Environment of Soccer**

   Link: https://ieeexplore.ieee.org/document/9085411

   Summary: Shitanshu Kusmakar, Sergiy Shelyag, Ye Zhu, Dan Dwyer, Paul Gastin and Maia Angelova performed analysis of the pattern-forming dynamics of player interactions can uncover the clues to underlying tactical behaviour. Their study aims to propose quantitative measures of a team's performance derived only using player interactions. The used Coarse-Grain Player Interaction Model including SVM and Learning classification models. The algorithm showed an accuracy of 0.84 in the classification and correctly predicted the match outcome in 81.9% of matches that ended in a result.

5. **Exploiting sports-betting market using machine learning**

   Link:https://www.researchgate.net/publication/331218530

   Summary: The authors of this paper present a system that uses machine learning for sports-betting market forecasting, with a focus on NBA data from seasons 2007-2014. They introduce three novel contributions to the field: 1) optimizing the model's predictive accuracy while also reducing correlation with bookmaker's predictions, allowing for better profit generation, 2) using convolutional neural networks to leverage a large number of player-related statistics for match outcome prediction, and 3) applying elements of modern portfolio theory to design a betting strategy that optimizes profit expectation and variance. The results of experiments show that their method yields positive cumulative profits, making it a successful approach to exploit the bookmaker.

6. **A machine learning framework for sport result prediction**

   Link: https://www.sciencedirect.com/science/article/pii/S2210832717301485

   Summary: The article discusses the application of machine learning (ML) in sport result prediction, focusing on the use of artificial neural networks (ANNs) as the key approach used in earlier research papers. The aim of sport result prediction is to predict the outcome of a match, which can be treated as a classification problem with one of three classes (win, lose, or draw). Large volumes of data, including historical performance of the teams, match results, and player data, are used to help stakeholders, such as bookmakers, fans, and club managers, understand the odds of winning or losing forthcoming matches. The article provides a critical review of the literature on ML for sport result prediction and proposes a novel sport prediction framework, SRP-CRISP-DM, based on the six steps of the standard CRISP-DM framework. The article also discusses the challenges facing the sport prediction application and provides recommendations for future research. The proposed framework and the critical analysis

presented in this article may be useful for researchers, sport fans, club managers, bookmakers, academics, and students interested in intelligent solutions based on ANN for the challenging problem of sport result prediction.

7. **Machine learning in football betting: Prediction of match results based on player characteristics**

   Link:https://www.mdpi.com/2076-3417/10/1/46

   Summary: The article discusses the increasing popularity of soccer and the expansion of bookmakers offering betting on soccer games. The paper utilizes machine learning to predict the outcomes of soccer games based on various match and player attributes. The study includes all matches of the five greatest European soccer leagues and their corresponding second leagues between 2006 and 2018. The results show that an ensemble strategy combining different machine learning algorithms achieves statistically and economically significant returns of 1.58% per match, outperforming traditional approaches such as linear regression or naive betting strategies. The authors present a machine learning framework for forecasting soccer matches and achieving excess returns through appropriate betting. The study provides a comparison of different approaches and challenges traditional approaches in this area. The article concludes with suggestions for future research, including the possibility of using the statistical arbitrage framework to forecast the results of other sporting events and incorporating time as an important feature to increase the accuracy of the model.

8. **Sports prediction and betting models in the machine learning age: The case of tennis**

   Link: https://content.iospress.com/articles/journal-of-sports-analytics/jsa200463

   Summary: This paper uses machine learning to predict professional tennis matches and exploit betting market inefficiencies. The study uses one of the largest datasets, covering 39,000 professional men's and women's tennis matches from 2010 to 2019. The study examines two main questions: (1) whether machine learning techniques outperform simple model-free forecasts based on players' official rankings or betting odds, and (2) whether any model can consistently generate positive returns for bettors. All models outperform player rankings alone, but none outperform betting odds-implied forecasts. Bookmaker odds predict match outcomes better than historical match and player data. The study also finds that model-based betting strategies have high volatility, low liquidity, and negative long-term returns. Model ensembles that combine predictions from multiple approaches are the best option. Due to low returns and high volatility, machine learning in professional tennis betting may not be a good investment. However, the results should encourage sports prediction and betting

research using recurrent neural networks and other advanced machine learning methods.

9. **Predicting Results of Indian Premier League T-20 Matches using Machine Learning**

   Link : https://ieeexplore.ieee.org/document/8820235

   Summary: The article discusses the popularity of cricket as a game among people of all ages and how it has become a billion-dollar industry for many who invest in it. The article highlights the concerns surrounding spot fixing, which has contributed to the gambling market being on the rise. The paper presents a study on predicting the winner of upcoming IPL matches using a model based on machine learning algorithms. The model takes into account factors such as the individual competency of each player, coordination and teamwork of the whole team, and the techniques used by each team in each match. The paper applies three machine learning algorithms, namely Support Vector Machine, CTree, and Naïve Bayes, achieving an accuracy of 95.96%, 97.98%, and 98.99%, respectively. This study could be valuable for cricket enthusiasts and those involved in the cricket industry who are interested in predicting match outcomes.

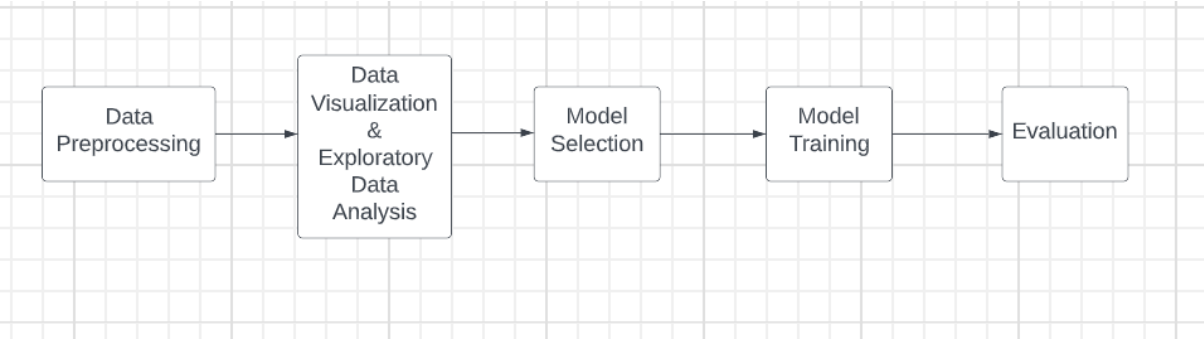
10. **Artificial Intelligence and Machine Learning in Sport Research: An Introduction for Non-data Scientists**

    Link : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8692708/

    Summary: The article discusses how artificial intelligence (AI) has revolutionized sports analysis in the past two decades, enhancing decision-making, forecasting, and other aspects of sports. Despite this, the connection between AI and sports is not well understood by many in the industry and academic sector who are not experts in AI. This paper provides a non-technical overview of the machine learning (ML) paradigm, highlighting its potential to enhance sports performance and business analytics. The article summarizes relevant research literature on the application of AI and ML in the sports industry and sport research. The authors also provide hypothetical scenarios of how AI and ML could shape the future of sports. Overall, the article aims to improve understanding of the role of AI in sports and its potential to drive innovation in the field.

PROPOSED METHODOLOGY

1. Data Collection: Collect data on previous football matches, including information on the teams, players, match conditions, and match outcomes.
2. Data Pre-processing: Pre-process the data by converting categorical variables into numerical values and normalizing the data.
3. Feature Selection: Select the relevant features that can affect the outcome of the match, such as team strength, player form, past performance, and match conditions.
4. Model Training: Train the SVM, Random Forest, Logistic Regression, XGBoost, Naïve Bayes, LDA, QDA, and Neural Network models using the selected features and the training data.
5. Hyperparameter Tuning: Use cross-validation to tune the hyperparameters of the models and improve their performance.
6. Model Evaluation: Test the models using the testing data and evaluate their performance using evaluation metrics such as accuracy, precision, recall, and F1-score.
7. Ensemble Learning: Combine the predictions of the four models using an ensemble learning approach to improve the overall prediction accuracy.
8. Deployment: Deploy the prediction system to predict the outcomes of new football matches and provide the results to the users.



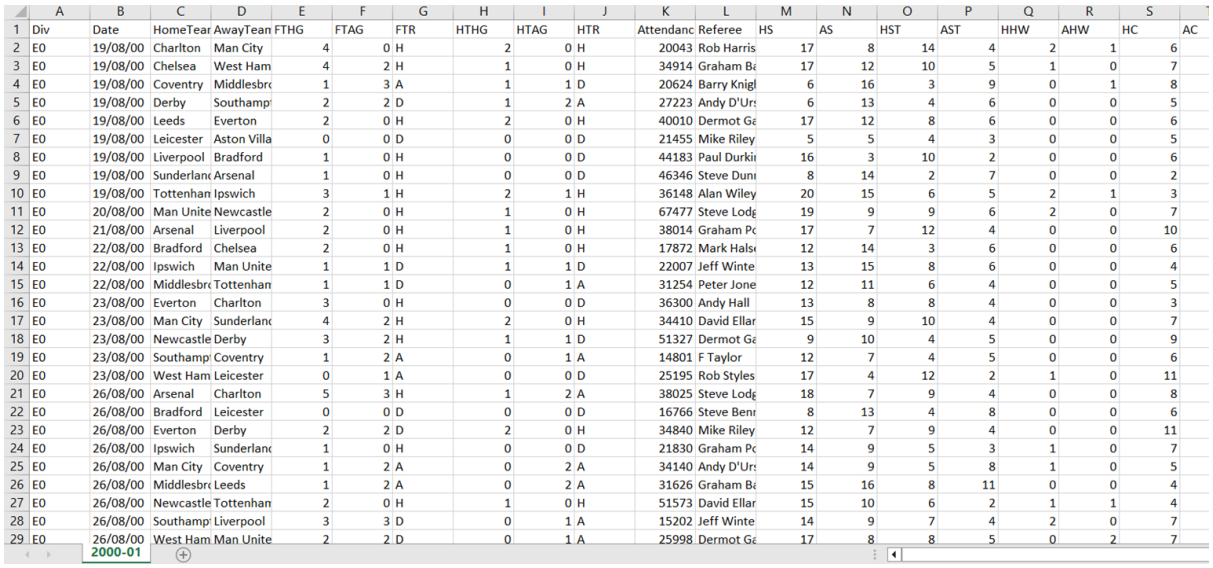
DATASET

Basic Information: It has the data of all the matches of the English Premier League (top-division football tournament in England) season wise from 2000-01 to 2017-18.

The Major Attributes in the dataset:

- Div - League Division
- Date - Date of the Match
- Time - Time of start of the match
- HomeTeam - Name of the home team
- Away Team - Name of the away team

- HTR - Score of the game till half-time
- HTAG - Goals scored by home team till half-time
- HTHG - Goals scored by home team till half-time
- FTAG - Goals scored by away team
- FTHG - Goals scored by home team



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
1	Div	Date	HomeTear	AwayTean	FTHG	FTAG	FTR	HTHG	HTAG	HTR	Attendanc	Referee	HS	AS	HST	AST	HHW	AHW	HC	AC
2	E0	19/08/00	Charlton	Man City	4	0	H	2	0	H	20043	Rob Harris	17	8	14	4	2	1	6	
3	E0	19/08/00	Chelsea	West Ham	4	2	H	1	0	H	34914	Graham Ba	17	12	10	5	1	0	7	
4	E0	19/08/00	Coventry	Middlesbr	1	3	A	1	1	D	20624	Barry Knigl	6	16	3	9	0	1	8	
5	E0	19/08/00	Derby	Southamp	2	2	D	1	2	A	27223	Andy D'Urs	6	13	4	6	0	0	5	
6	E0	19/08/00	Leeds	Everton	2	0	H	2	0	H	40010	Dermot Ga	17	12	8	6	0	0	6	
7	E0	19/08/00	Leicester	Aston Villa	0	0	D	0	0	D	21455	Mike Riley	5	5	4	3	0	0	5	
8	E0	19/08/00	Liverpool	Bradford	1	0	H	0	0	D	44183	Paul Durkii	16	3	10	2	0	0	6	
9	E0	19/08/00	Sunderlan	Arsenal	1	0	H	0	0	D	46346	Steve Duni	8	14	2	7	0	0	2	
10	E0	19/08/00	Tottenhan	Ipswich	3	1	H	2	1	H	36148	Alan Wiley	20	15	6	5	2	1	3	
11	E0	20/08/00	Man Unite	Newcastle	2	0	H	1	0	H	67477	Steve Lodg	19	9	9	6	2	0	7	
12	E0	21/08/00	Arsenal	Liverpool	2	0	H	1	0	H	38014	Graham Pc	17	7	12	4	0	0	10	
13	E0	22/08/00	Bradford	Chelsea	2	0	H	1	0	H	17872	Mark Halsl	12	14	3	6	0	0	6	
14	E0	22/08/00	Ipswich	Man Unite	1	1	D	1	1	D	22007	Jeff Winte	13	15	8	6	0	0	4	
15	E0	22/08/00	Middlesbr	Tottenhan	1	1	D	0	1	A	31254	Peter Jone	12	11	6	4	0	0	5	
16	E0	23/08/00	Everton	Charlton	3	0	H	0	0	D	36300	Andy Hall	13	8	8	4	0	0	3	
17	E0	23/08/00	Man City	Sunderlan	4	2	H	2	0	H	34410	David Ellar	15	9	10	4	0	0	7	
18	E0	23/08/00	Newcastle	Derby	3	2	H	1	1	D	51327	Dermot Ga	9	10	4	5	0	0	9	
19	E0	23/08/00	Southamp	Coventry	1	2	A	0	1	A	14801	F Taylor	12	7	4	5	0	0	6	
20	E0	23/08/00	West Ham	Leicester	0	1	A	0	0	D	25195	Rob Styles	17	4	12	2	1	0	11	
21	E0	26/08/00	Arsenal	Charlton	5	3	H	1	2	A	38025	Steve Lodg	18	7	9	4	0	0	8	
22	E0	26/08/00	Bradford	Leicester	0	0	D	0	0	D	16766	Steve Benr	8	13	4	8	0	0	6	
23	E0	26/08/00	Everton	Derby	2	2	D	2	0	H	34840	Mike Riley	12	7	9	4	0	0	11	
24	E0	26/08/00	Ipswich	Sunderlan	1	0	H	0	0	D	21830	Graham Pc	14	9	5	3	1	0	7	
25	E0	26/08/00	Man City	Coventry	1	2	A	0	2	A	34140	Andy D'Urs	14	9	5	8	1	0	5	
26	E0	26/08/00	Middlesbr	Leeds	1	2	A	0	2	A	31626	Graham Ba	15	16	8	11	0	0	4	
27	E0	26/08/00	Newcastle	Tottenhan	2	0	H	1	0	H	51573	David Ellar	15	10	6	2	1	1	4	
28	E0	26/08/00	Southamp	Liverpool	3	3	D	0	1	A	15202	Jeff Winte	14	9	7	4	2	0	7	
29	E0	26/08/00	West Ham	Man Unite	2	2	D	0	1	A	25998	Dermot Ga	17	8	8	5	0	2	7	

2000-01

IMPLEMENTATION

Support Vector Machine:

```
> library("e1071")
> library(caret)
> options(repos = c(CRAN = "http://cran.rstudio.com"))
> set.seed(319)
> ind <- sample(2, nrow(dataset), replace = TRUE, prob = c(0.90,0.10))
> training = dataset[ind==1,]
> testing = dataset[ind==2,]
> dim(training)
[1] 5449   43
> dim(testing)
[1] 631  43
> svm_model1 <- svm(FTR ~ .,kernel = "radial", data=training)
> summary(svm_model1)

Call:
svm(formula = FTR ~ ., data = training, kernel = "radial")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.0004103406
```

```
Number of Support Vectors:  3265

 ( 1632 1633 )


Number of Classes:  2

Levels:
 H NH



> pred1 <- predict(svm_model1, newdata = testing)
> confusionMatrix(pred1, testing$FTR )
Confusion Matrix and Statistics

          Reference
Prediction   H  NH
        H  269   8
        NH  25 329

               Accuracy : 0.9477
                 95% CI : (0.9273, 0.9637)
    No Information Rate : 0.5341
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8945
 Mcnemar's Test P-Value : 0.005349
```

Logistic Regression:

```
> library(clusterSim)
> dataset = data.frame(lapply(dataset, function(x) as.numeric(x)))
> dataset = data.Normalization (dataset,type="n4",normalization="column")
> test = data.frame(lapply(test, function(x) as.numeric(x)))
> test = data.Normalization (test,type="n4",normalization="column")
> glm.fits=glm(FTR ~ .,data=dataset,family=binomial)
Warning messages:
1: glm.fit: algorithm did not converge
2: glm.fit: fitted probabilities numerically 0 or 1 occurred
> #glm.fits=glm(FTR ~ HomeTeam + AwayTeam + DiffPts+DiffLP+ DiffFormPts,da
ta=dataset,family=binomial)
> summary(glm.fits)

Call:
glm(formula = FTR ~ ., family = binomial, data = dataset)

Deviance Residuals:
             Min                1Q            Median                3Q
Max
-0.0000134258884  -0.0000000210734   0.0000000210734   0.0000074003043   0
.0000114591770
```

```
Coefficients: (2 not defined because of singularities)
                    Estimate       Std. Error  z value Pr(>|z|)
(Intercept)       24.777835096  41140.470790097  0.00060  0.99952
X                  2.832266444 583824.709939697  0.00000  1.00000
Date              -2.740236949 581339.323508215  0.00000  1.00000
HomeTeam          -0.113207924   5940.976712012 -0.00002  0.99998
AwayTeam          -0.101499952   5954.867029370 -0.00002  0.99999
FTHG            -430.091202898  32670.733741953 -0.01316  0.98950
FTAG             286.241820871  22348.509378651  0.01281  0.98978
HTGS              -0.134851137  39014.916733706  0.00000  1.00000
ATGS               0.142451016  37900.521012846  0.00000  1.00000
HTGC               0.255795467  34719.176369211  0.00001  0.99999
ATGC               0.246261645  33631.291961249  0.00001  0.99999
HTP                0.350616531  29109.485458441  0.00001  0.99999
ATP                0.187161381  28679.114968246  0.00001  0.99999
HM1                0.336088210  29845.014706247  0.00001  0.99999
HM2                0.292520241  12078.595036077  0.00002  0.99998
HM3                0.139292378   7096.816970097  0.00002  0.99998
HM4                0.028816661   5963.409513349  0.00000  1.00000
HM5               -0.040035893   5686.760236158 -0.00001  0.99999
AM1                1.077214689  29490.873608620  0.00004  0.99997
AM2                0.378240118  11943.243082862  0.00003  0.99997
AM3                0.098414596   6892.930782557  0.00001  0.99999
AM4                0.095207671   5984.354733471  0.00002  0.99999
AM5               -0.007577885   5772.348839886  0.00000  1.00000
HomeTeamLP         0.100898393   7202.061698739  0.00001  0.99999
AwayTeamLP        -0.017877180   7308.578551302  0.00000  1.00000



MW                -0.391787353  31448.781734635 -0.00001  0.99999
HTFormPtsStr      -0.890941115  42546.813305330 -0.00002  0.99998
ATFormPtsStr      -1.813896947  42384.900918111 -0.00004  0.99997
HTFormPts         -0.022660524  22486.617562129  0.00000  1.00000
ATFormPts         -0.209716234  21733.339220800 -0.00001  0.99999
HTWinStreak3      -0.016772026  10169.383971325  0.00000  1.00000
HTWinStreak5       0.130261856  16493.923311396  0.00001  0.99999
HTLossStreak3      0.009183538   9406.991368763  0.00000  1.00000
HTLossStreak5      0.216805324  19411.604241708  0.00001  0.99999
ATWinStreak3       0.098532600  10023.389944058  0.00001  0.99999
ATWinStreak5      -0.293865560  17866.714842300 -0.00002  0.99999
ATLossStreak3     -0.067678888   9624.037581689 -0.00001  0.99999
ATLossStreak5      0.095418803  18946.938573252  0.00001  1.00000
HTGD              -0.020878120  53909.889815444  0.00000  1.00000
ATGD              -0.546292650  52997.739892298 -0.00001  0.99999
DiffPts                   NA             NA       NA       NA
DiffFormPts       -0.569780309  54118.336747709 -0.00001  0.99999
DiffLP                    NA             NA       NA       NA

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8395.6292532105308  on 6079  degrees of freedom
Residual deviance:    0.0000002440788  on 6039  degrees of freedom
AIC: 82

Number of Fisher Scoring iterations: 25
```

```
Number of Fisher Scoring iterations: 25

> glm.probs=predict(glm.fits,test,type="response")
Warning message:
In predict.lm(object, newdata, se.fit, scale = 1, type = ifelse(type ==  :
  prediction from a rank-deficient fit may be misleading
> glm.probs = ifelse(glm.probs>0.5,1,0)
> table(glm.probs,test[,7])

glm.probs   0   1
        0 157  49
        1   0 174
> mean(glm.probs==test[,7])
[1] 0.8710526316
```

Random Forest:

```python
# Train a Random Forest Regression model
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Make predictions on the testing set
y_pred = rf_model.predict(X_test)
y_pred_binary = (y_pred > 0.5).astype(int)

# Convert the actual labels to binary labels
y_test_binary = y_test.astype(int)

# Calculate accuracy based on binary predictions
accuracy = accuracy_score(y_test_binary, y_pred_binary)

# Evaluate the model based on mean squared error
mse = mean_squared_error(y_test, y_pred)
confusion_mat = confusion_matrix(y_test_binary, y_pred_binary)
print(f"Accuracy: {accuracy}")
print(f"Mean Squared Error: {mse}")
print("Confusion matrix:\n", confusion_mat)
```

```
print(classification_report(y_test, Y_pred))
```

	precision	recall	f1-score	support
H	0.63	0.55	0.58	953
NH	0.65	0.72	0.68	1099
accuracy			0.64	2052
macro avg	0.64	0.63	0.63	2052
weighted avg	0.64	0.64	0.64	2052

XGBoost:

```
# Make predictions on the testing set
y_pred = model.predict(dtest)
y_pred_binary = (y_pred > 0.5).astype(int)

# Convert the actual labels to binary labels
y_test_binary = y_test.astype(int)

# Calculate accuracy based on binary predictions
accuracy = accuracy_score(y_test_binary, y_pred_binary)

# Evaluate the model based on mean squared error
mse = mean_squared_error(y_test, y_pred)
confusion_mat = confusion_matrix(y_test_binary, y_pred_binary)
print(f"Accuracy: {accuracy}")
print(f"Mean Squared Error: {mse}")
print("Confusion matrix:\n", confusion_mat)
```

```
              0       0.60      0.56      0.58       953
              1       0.64      0.68      0.66      1099

    micro avg       0.62      0.62      0.62      2052
    macro avg       0.62      0.62      0.62      2052
 weighted avg       0.62      0.62      0.62      2052
  samples avg       0.62      0.62      0.62      2052
```

Naïve Bayes:

```
> dataset = read.csv("final_dataset.csv")
> test = read.csv("test_set.csv")
> library('varhandle')
> library('e1071',warn.conflicts=FALSE)
> naive_bayes_model<-naiveBayes(FTR ~ ., data = dataset)
> naive_bayes_predictions<-predict(naive_bayes_model, newdata=test)
> head(naive_bayes_predictions)
[1] NH NH NH H  NH NH
Levels: H NH
> table(naive_bayes_predictions,test[,7])
```

```
naive_bayes_predictions   H  NH
                      H  398 144
                      NH 119 479
> mean(naive_bayes_predictions==test[,7])
[1] 0.7692982
>
```

LDA:

```
> dataset = read.csv("final_dataset.csv")
> test = read.csv("test_set.csv")
> library(MASS)
> #lda.fit = lda(FTR ~ .,data=dataset)
> lda.fit = lda(FTR ~ HomeTeam + AwayTeam + DiffPts+ DiffFormPts,data=dataset)
> summary(lda.fit)
        Length Class  Mode
prior     2    -none- numeric
counts    2    -none- numeric
means   176    -none- numeric
scaling  88    -none- numeric
lev       2    -none- character
svd       1    -none- numeric
N         1    -none- numeric
call      3    -none- call
terms     3    terms  call
xlevels   2    -none- list
> lda.pred=predict(lda.fit, test[-7,])
> names(lda.pred)
[1] "class"     "posterior" "x"
> lda.class=predict(lda.fit,test)$class
> head(test[,7])
[1] "NH" "NH" "NH" "H"  "H"  "NH"
> table(lda.class,test[,7])

lda.class   H  NH
       H  284 164
       NH 233 459
> mean(lda.class==test[,7])
[1] 0.6517544
>
```

QDA:

```
> dataset = read.csv("final_dataset.csv")
> test = read.csv("test_set.csv")
> library(MASS)
> qda.fit = qda(FTR ~ HomeTeam + AwayTeam + DiffPts+ DiffFormPts,data=dataset)
> summary(qda.fit)
```

```
        Length Class  Mode
prior        2  -none- numeric
counts       2  -none- numeric
means      176  -none- numeric
scaling  15488  -none- numeric
ldet         2  -none- numeric
lev          2  -none- character
N            1  -none- numeric
call         3  -none- call
terms        3  terms  call
xlevels      2  -none- list
> qda.class=predict(qda.fit,test)$class
> table(qda.class,test[,7])

qda.class   H  NH
       H  306 189
       NH 211 434
> mean(qda.class==test[,7])
[1] 0.6491228
> |
```

Neural Network:

```
> library(clusterSim)
> dataset = data.frame(lapply(dataset, function(x) as.numeric(x)))
> dataset = data.Normalization (dataset,type="n4",normalization="column")
> test = data.frame(lapply(test, function(x) as.numeric(x)))
> test = data.Normalization (test,type="n4",normalization="column")
> # Neural Networks
> library(neuralnet)
> set.seed(319)
> # 5 neurons hidden layer
> n = neuralnet(FTR ~ X + Date + HomeTeam + AwayTeam + FTHG + FTAG + HTGS
+ ATGS +
+                 HTGC + ATGC + HTP + ATP + HM1 + HM2 + HM3 + HM4 + HM5 +
AM1 +
+                 AM2 + AM3 + AM4 + AM5 + HomeTeamLP + AwayTeamLP + MW + H
TFormPtsStr +
+                 ATFormPtsStr + HTFormPts + ATFormPts + HTWinStreak3 + HT
WinStreak5 +
+                 HTLossStreak3 + HTLossStreak5 + ATWinStreak3 + ATWinStre
ak5 +
+                 ATLossStreak3 + ATLossStreak5 + HTGD + ATGD + DiffPts +
DiffFormPts +
+                 DiffLP,data = dataset,hidden = 5,err.fct = "ce",linear.o
utput = FALSE)
> # Prediction
> output <- compute(n, dataset[,-7])
> head(output$net.result)
                      [,1]
1 0.00000000005496020571
2 0.00000000094983300103
3 1.00000000000000000000
4 0.99999843402808674675
5 0.00000000065638662296
6 0.99999795743205188714
> head(dataset[1,])
```

```
  X Date     HomeTeam      AwayTeam          FTHG FTAG FTR HTGS ATGS HTGC ATG
C HTP ATP           HM1           HM2           HM3
1 0    0 0.243902439 0.5365853659 0.4444444444    0   0    0    0    0
0   0   0 0.6666666667 0.6666666667 0.6666666667
           HM4           HM5           AM1           AM2           AM3
AM4           AM5 HomeTeamLP AwayTeamLP MW
1 0.6666666667 0.6666666667 0.6666666667 0.6666666667 0.6666666667 0.66666
66667 0.6666666667           1           1  0
  HTFormPtsStr ATFormPtsStr HTFormPts ATFormPts HTWinStreak3 HTWinStreak5
HTLossStreak3 HTLossStreak5 ATWinStreak3
1  0.658045977 0.6703910615         0         0            0            0
0             0             0
  ATWinStreak5 ATLossStreak3 ATLossStreak5         HTGD         ATGD       D
iffPts DiffFormPts DiffLP
1            0             0             0 0.4285714286 0.487804878 0.5123
152709         0.5    0.5
> # Confusion Matrix & Misclassification Error - training data
> output <- compute(n, dataset[,-7])
> p1 <- output$net.result
> pred1 <- ifelse(p1>0.5, 1, 0)
```

```
> tab1 <- table(pred1, dataset$FTR)
> tab1

pred1    0    1
    0 2816    0
    1    0 3264
> sum(diag(tab1))/sum(tab1)
[1] 1
> # Confusion Matrix & Misclassification Error - testing data
> output <- compute(n, test[,-7])
> p2 <- output$net.result
> pred2 <- ifelse(p2>0.5, 1, 0)
> tab2 <- table(pred2, test$FTR)
> tab2

pred2   0   1
    0 157  51
    1   0 172
> sum(diag(tab2))/sum(tab2)
[1] 0.8657894737
```

RESULTS

Model Used	Dataset Used	Model Performance
		Accuracy in %
SVM	Football Dataset	94.77
Logistic Model		87.10
Random Forest		64
XGBoost		68
Naïve Bayes		76.92
LDA		65.17
QDA		64.91
Neural Network		86.57

***Note:** KNN Model was rejected due to overfitting

CONCLUSION

In conclusion, this project aimed to predict whether a home team would win its match or not using four different machine learning algorithms. We found that all methods—Support Vector Machine (SVM), Logistic Regression, Random Forest, XGBoost, Naïve Bayes, LDA, QDA, and Neural Network —achieved high accuracy rates in predicting match outcomes. However, Support Vector Machine (SVM) performed the best among the eight, indicating its potential to be used in sports betting to increase the chances of winning.

Our results suggest that the use of machine learning algorithms could be a powerful tool for predicting the outcomes of soccer matches. It could also provide valuable insights for coaches and teams, helping them to develop better strategies and improve their performance. Overall, this project highlights the importance of machine learning in sports analytics and its potential to revolutionize the way we analyse and predict sports outcomes.

REFERENCES

T. K. Das and K. Roy, "A Comparative Study of Machine Learning Techniques for Sports Outcome Prediction," in Proceedings of the 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 2019, pp. 103-107.

M. V. Vishwanath and S. S. Manvi, "Sports Outcome Prediction Using Machine Learning," in Proceedings of the 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), 2019, pp. 1-5.

X. Huang, J. Zhang, and K. Wang, "A Deep Learning Framework for Sports Result Prediction," in Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 2168-2173.

Y. Li, J. Li, and J. Li, "A Machine Learning Approach for Sports Result Prediction Based on a Decision Tree Algorithm," in Proceedings of the 2019 International Conference on Computer Science, Engineering and Information Technology (CSEIT), 2019, pp. 221-226.

Z. Zou, X. Hu, and Y. Zhang, "Sports Result Prediction Based on Deep Learning Model," in Proceedings of the 2020 IEEE International Conference on Computer and Communications (ICCC), 2020, pp. 189-193.

Y. Cui, C. Liu, and J. Tao, "A Deep Learning Approach for Sports Result Prediction," in Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 1647-1652.

K. J. Keogh, P. E. Pfeiffer, and D. M. D. Paterson, "Sports Forecasting: A Comparison of the Forecast Accuracy of Prediction Markets, Betting Odds and Tipsters," Journal of Prediction Markets, vol. 6, no. 3, pp. 1-18, 2012.

M. C. Kumar and S. U. Kumar, "Sports Outcome Prediction Using Machine Learning Techniques," in Proceedings of the 2016 International Conference on Advanced Computing Technologies and Applications (ICACTA), 2016, pp. 1-6.

R. F. Nascimento, T. F. R. Filho, and L. A. M. Pereira, "A Machine Learning Approach for Sports Result Prediction," in Proceedings of the 2017 IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE), 2017, pp. 174-179.

S. Saha, S. Bandyopadhyay, and S. Maulik, "Sports Result Prediction Using Machine Learning Techniques," in Proceedings of the 2017 International Conference on Computing, Communication and Automation (ICCCA), 2017, pp. 644-648.

Bandyopadhyay and S. Saha, "A Comprehensive Study on Sports Outcome Prediction Using Machine Learning Techniques," in Proceedings of the 2018 International Conference on Information Technology (ICIT), 2018, pp. 167-172.

C. Zhang, T. Chen, Y. Lu, and L. Zhang, "Sports Betting Based on Machine Learning Algorithms," in Proceedings of the 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), 2019, pp. 169-172.

T. Chen, C. Zhang, Y. Lu, and L. Zhang, "Sports Result Prediction Based on Machine Learning," in Proceedings of the International Conference on Internet of Things and Machine Learning (IML'19), 2019, pp. 1-5.