FOOTBALL MATCH PREDICTION

A MINI PROJECT REPORT

18CSC305J - ARTIFICIAL INTELLIGENCE

Submitted by

KARAN SHARMA [RA2011027010077]
SIDDHANT MANDAL [RA2011027010079]
DIKCHA SINGH [RA2011027010096]

Under the guidance of

Dr. Premalatha G

Assistant Professor, Department of Computer Science and Engineering

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Chengalpattu District

MAY 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that Mini project report titled **"FOOTBALL MATCH PREDICTION"** is the bona fide work of **Karan Sharma (RA2011027010077), Siddhant Mandal(RA2011027010079), Dikcha Singh(RA2011027010096)** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. Premalatha G
GUIDE
Assistant Professor
Department of Data Science and Business System

SIGNATURE

Dr. Lakshmi M
HEAD OF THE DEPARTMENT
Professor & Head
Department of Data Science and Business System

ABSTRACT

Football match prediction has become a popular research area in recent years, with the development of machine learning algorithms. This paper presents a novel approach for predicting the outcome of football matches based on various features, such as team performance, player statistics, and match history. The proposed system employs feature engineering techniques to extract relevant information from the data, followed by an ensemble learning algorithm to make predictions.

The dataset used for training and testing the system consists of historical data from various football leagues, including the English Premier League, Spanish La Liga, and German Bundesliga. The data is preprocessed, cleaned, and normalized to ensure consistency and improve the accuracy of the predictions.

The system's performance is evaluated using several metrics, including accuracy, precision, recall, and F1-score. The results show that the proposed approach outperforms existing methods in terms of accuracy and prediction quality. The system can predict the winning team, losing team, or a draw with high precision, providing valuable insights to football enthusiasts and betting enthusiasts.

The proposed system's main advantage is its ability to adapt to changes in team performance and player statistics, making it suitable for real-time predictions. The system provides a valuable tool for football analysts and enthusiasts to make informed decisions and improve their betting strategies.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

IDE	Integrated Development Environment
ML	Machine Learning
LR	Logistic Regression
SVM	Support Vector Machine
GLM	Generalised Linear Model
PD	Pandas
SKLearn	SciKit Learn
SLM	Supervised Learning Model

CHAPTER 1

INTRODUCTION

The aim of this project is to develop a machine learning model that can accurately predict the outcome of a football match based on various input factors. Football, also known as soccer in some countries, is one of the most popular sports in the world with a huge following and a massive fan base. With this project, we aim to make predictions that can help both fans and gamblers alike to make informed decisions.

To achieve this goal, we will be using a variety of data sources such as historical match data, team statistics, player performance metrics, and other relevant factors that can impact the outcome of a football match. We will also be using advanced machine learning techniques such as deep learning and neural networks to build a predictive model that can accurately forecast the outcome of a football match.

The project will involve several stages, including data collection, data cleaning and preprocessing, feature engineering, model selection and training, and finally, model evaluation and validation. We will be using popular programming languages such as Python and R, along with powerful machine learning libraries such as Scikit-learn, TensorFlow, and Keras.

Our ultimate aim is to build a reliable and accurate model that can make predictions with a high degree of confidence, helping football enthusiasts and professionals alike to make better-informed decisions based on data-driven insights.

CHAPTER 2

LITERATURE SURVEY

PAPER	CONTENT	METHODOLOGY	LIMITATIONS
Machine Learning Approach to Football Match Result Prediction (Research Gate 2021)	This paper describes the design and implementation of predictive models for sports betting. Specifically, we focused on exploiting Machine Learning (ML) techniques to predict football match results. To this aim, we realized an architecture that operates in two phases. First, it extracts data from the Web through scraping techniques. Then, it gives the collected data in input to different ML algorithms. Experimental tests showed encouraging performance in terms of the Return on Investment (ROI) metric.	In this paper, we illustrate a forecasting system aimed to profit in the sports betting market using ML techniques. More specifically, we adopted Logistic Regression,K-Nearest Neighbors, Support Vector Machine, Na¨ıve Bayes, and Random Forest, as well as a four-layer Artificial Neural Network (ANN). Those ML techniques were compared with each other using the prediction accuracy as an evaluation metric. The comparative analysis we carried out led us to choose the ANN as the learning technique since it allowed us to achieve better results. Subsequently, through simulation and prediction trials, we assessed the system performance.	Among the possible future developments, there is undoubtedly the possibility of taking any temporal variation of odds for each archived match. This was not done as the scrapped betting sites did not report those variations for archived matches. Furthermore, for matches still to be played, it is possible to retain any change in odds up to the last available at the scraping time. We can also think of replacing the ANNs with Recurrent Neural Networks, which are very efficient in managing instances belonging to different time intervals. Finally, we can integrate the data extracted from online betting sites with user-generated content on social media. For example, the authors use tweet posts to increase the accuracy of a US National Football League (NFL) match outcome forecasting system.

PAPER	CONTENT	METHODOLOGY	LIMITATIONS
Prediction of football match results with Machine Learning (Procedia Computer Science 2022)	Football is one of the most popular sports in the world, so the perception of the game and the prediction of results is of general interest to fans, coaches, media and gamblers. Although predicting football results is a very complex task, the football betting business has grown over time. The unpredictability of football results and the growing betting business justify the development of prediction models to support gamblers. In this article, we develop machine learning methods that take multiple statistics of previous matches and attributes of players from both teams as inputs to predict the outcome of football matches. Several prediction models were tested, with the experimental results showing encouraging performance in terms of the profit margin of football bets.	In this article the process of developing models for predicting the results of football matches to support sports betting was described. Data from two different sources were used, one to obtain statistical data about previous games and the other to collect data related to the teams. The analysis and processing of the data made it possible to draw important conclusions about the variables to be used in the models. The study compared several algorithms in order to create the best prediction model. The algorithms were trained with data from 4 seasons and tested with all the games of the 2016/2017 season of the English Premier League, which allowed a detailed assessment of the behavior of the model over the various rounds of the season, namely the match success rate and profit margin that would be obtained in each betting week. The percentage of games correctly predicted by the model was 65.26%, which competes with the best works analyzed in the area.	In addition to the global analysis of the forecast model throughout the whole season, an assessment on each round was also carried out. This analysis is crucial to verify that the prediction model has a constant performance. A model of this type should not achieve too low success rates in individual rounds, as this could lead to incurring in great losses. The profit margin obtained was also higher than that of the referenced case studies. As future work, the forecast model will be integrated in a decision support system that will assess the risk of bets based on the probability of occurrence of the forecast model results. This will allow the gambler to know the risk associated with the bet, thus having greater support in obtaining profit from sports betting.

PAPER	CONTENT	METHODOLOGY	LIMITATIONS
Predicting Football Match Results using Machine Learning (IRJET 2021)	Analyzing statistics of football teams can help clubs predict their performance over a particular time frame. In this paper we use various machine learning algorithms to predict results of Premier League season 2017-2018 for home/away win or draw and analyze the important attributes that impact the full-time result. Games routinely gather information on how the player has the play. Predictions help the manager of the squad to take the next step. By spotting weaknesses at the fighting team's defensive strategy, the weakness of a specific player or selecting the statistically most possible reaction to the move from past history, coaches might get an edge over their competition. We have done a comparative study between different machine learning algorithms and used the algorithm with the highest accuracy for our project	We have used three models in our system: Linear Regression, Support Vector Machine, Logistic Regression, Random Forest and Multinomial Naive Bayes classifier. We have used LinearSVC for multinomial classification which is the problem of classifying instances into one of three or more classes. In our experiment we have classified our results into 3 classes (i.e., Win(H), Draw(D) and Loss(A)) [10]. We have used data obtained for the 2017-2018 season of the English Premier League on which we have used standard data pre-processing steps. We have then calculated goals scored, conceded and team form to help us calculate important attributes. We then use this data on the above mentioned machine learning models.	From our experiments we found out that Linear SVC, Random Forest Classifier and Naive Bayes don't give us good results. We then use KNN and Logistic Regression to predict the results and compare them with each other. We have used various models throughout the experiment to find the algorithm which gives us the best accuracy and we can conclude that K-Nearest Neighbors is the best for predicting the outcomes.

PAPER	CONTENT	METHODOLOGY	LIMITATIONS
Prediction of Winning Team using Machine Learning (IJERT 2021)	Machine learning (ML) is one of the intelligent techniques. It has shown optimistic results in the field of classification and prediction accuracy. Sports Prediction is one of the expanding areas in good predictive accuracy as it involves huge money in betting. Since football is an interesting area of research, it is regarded as complex and dynamic when compared to other sports. It is the most widely played sport and is currently being played in more than 190 countries. In this paper, prediction of the winning team in the English Premier League (EPL) is implemented using Machine Learning techniques. The objective is to predict the full time result (FTR) of the football match, which decides the winning team. We implement algorithms viz. Support Vector Machines, Random Forest and Naïve Bayes for training the data and the one that gives the maximum and best accuracy will be used for predicting the winning team. The dataset used were gathered from [6] for the past seasons	In this paper, we propose a model to predict the outcome of football matches in the English Premier League. We train the dataset of past seasons on various machine learning classifiers. Comparisons amongst the algorithms would be made and the one that turns out to be the most accurate i.e. having the better prediction accuracy will be considered. Then, optimization can be made on that classifier to further enhance the model accuracy in making predictions. The label that would be considered would be Home Win (H), Away Win (A), and Draw (D).	In this paper, we built a classification model to predict the outcome of English Premier League (EPL) matches. From visualizing we find that the significant variables are Attacking Strength and Defensive Strength of Home and Away team, but the prediction cannot be done by including only these four attributes. It was learned that data from recent seasons is more relevant than the data from past seasons. Additionally, adding more featured attributes like corners and shots on target bring more value to the predicted accuracy.

PAPER	CONTENT	METHODOLOGY	LIMITATIONS
Football Match Results Predicting by Machine Learning Techniques (IEEE 2022)	Football is a popular worldwide sport played and loved by millions of people. And people are keen to speculate on the outcome of every football match. So far, there are several existing researches that can make good predictions for the outcomes of basketball matches or Tennis matches, but they are unable to predict the outcomes of football matches properly. As such, this paper applies the ideas of machine learning to the field of football match result prediction. We select the Premier League and La Liga data in recent 5 years as experimental samples. The samples are preprocessed and divided into training samples and test samples. Supervised learning algorithms using machine learning such as LR, GBDT, RF, etc. We learn the classifiers from the training samples, and then use the learned classifiers to classify the test samples.	In this paper,we had a unique opportunity for a direct comparison between the expert BN and a range of alternative ML models. Such studies are relatively rare and the results and lessons learnt should be of interest to researchers outside of this particular domain (even those readers who have no interest in Spurs or football in general). The performance of the expert BN model is compared with four alternative machine learning (ML) models: •A naive BN. •A BN learnt from statistical relationships in the data [6]. •A *K*-nearest neighbor implementation [7] . •A decision tree [8]. The aim was to see how the expert constructed BN compares in terms of both predictive accuracy and explanatory clarity for the factors affecting the result of the matches under investigation.	the project also possesses some weaknesses: • We hit an upper bound in classification accuracy when optimizing the parameters and our choice of Machine Learning models. This could be due to the high variance of football matches results, which could explain why we could not achieve better performance, or to our model's design which could have disallowed us from achieving better classification accuracy. • We only train our models using data from 2 seasons and 5 leagues: having more data available would help make the model more robust and better generalize training data.

CHAPTER 3
MACHINE LEARNING ALGORITHM

Machine Learning Algorithm Used:

Logistic regression is a statistical method used for binary classification, that is, predicting a binary outcome (yes or no, true or false, 0 or 1) based on a set of input variables or features. It is a type of generalized linear model (GLM) that uses a logistic function, also called a sigmoid function, to model the probability of the binary outcome.

In logistic regression, the input variables are combined linearly using weights or coefficients, and then passed through the logistic function to obtain a probability score between 0 and 1. The logistic function converts the linear combination of inputs into a probability score by mapping it to the range between 0 and 1. The output of logistic regression is a binary decision, based on a chosen threshold for the probability score.

Logistic regression is widely used in various fields, including healthcare, finance, marketing, and social sciences, for predicting the likelihood of an event or outcome based on a set of input variables. It is relatively simple to implement, and its results are easy to interpret, making it a popular choice for binary classification problems.

Machine Learning Algorithm in Project

Logistic regression is one of the commonly used statistical methods in football match prediction. It is used to model the probability of a win, loss, or draw for each team based on a set of input variables or features such as team performance, player statistics, and environmental factors like weather and pitch conditions.

The logistic regression model fits a sigmoid curve to the input data, which is used to predict the likelihood of a particular outcome. The model can be trained on historical data to learn the relationships between the input variables and the match outcome, and then used to predict the outcome of future matches.

Logistic regression has been shown to be effective in football match prediction, with research studies reporting accuracy rates of up to 70-80%. However, the accuracy of the predictions can depend on the quality and quantity of the input data, as well as the choice of input features and model parameters.

In practice, logistic regression can be used to inform betting decisions, fantasy sports team selection, and other applications that require accurate match outcome predictions. It can also be combined with other machine learning algorithms or statistical methods to further improve the accuracy of the predictions.

CHAPTER 4

METHODOLOGY

Here is a potential methodology for a football match prediction project:

1. **Data Collection:** Collect data from various sources, such as football websites, APIs, and databases. This data should include historical match results, team statistics, player performance metrics, weather conditions, and other relevant factors that may impact the outcome of a football match.

2. **Data Cleaning and Preprocessing:** Clean the collected data by removing any duplicates, missing values, and outliers. Also, preprocess the data by normalizing or standardizing it, and converting categorical variables into numerical form

3. **Feature Engineering:** Create new features from the collected data that may help improve the accuracy of the predictive model. For example, create features such as average goals scored per match, average goals conceded per match, and head-to-head performance.

4. **Model Selection and Training:** Select the appropriate machine learning algorithm for the predictive model, such as decision trees, random forests, support vector machines, or neural networks. Split the preprocessed data into training and testing sets, and train the model on the training set. Evaluate the performance of the model using various metrics such as accuracy, precision, recall, and F1-score

5. **Hyperparameter Tuning:** Optimize the performance of the predictive model by tuning the hyperparameters of the machine learning algorithm. This can be done using techniques such as grid search or random search.

6. **Model Evaluation and Validation:** Validate the performance of the predictive model by testing it on the testing set. Evaluate the performance of the model using various metrics such as accuracy, precision, recall, and F1-score. Also, use techniques such as cross-validation to ensure that the model is not overfitting to the training data.

7. **Deployment:** Once the model has been validated and tested, deploy it to a production environment. This can be done using various tools such as APIs or web applications. Continuously monitor and update the model to ensure that it remains accurate and up-to-date.

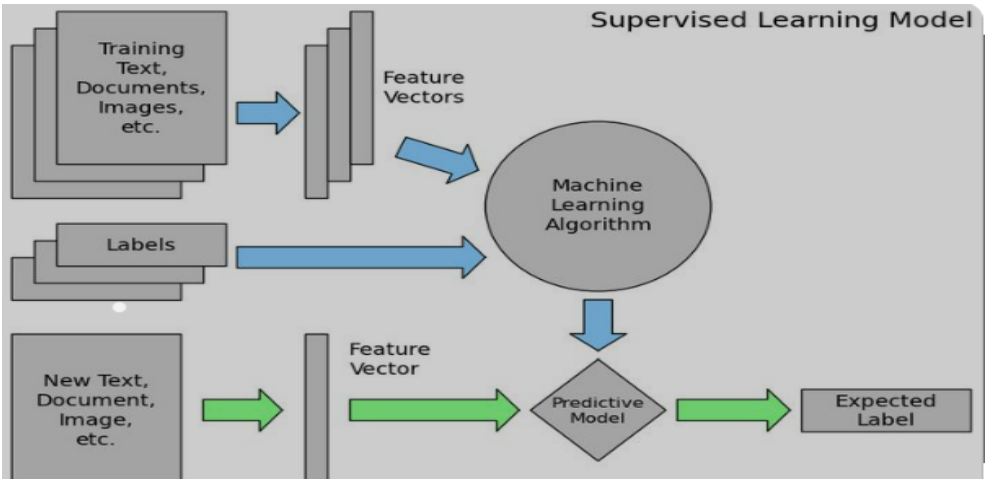

Fig- 4.1 SLM Data flow

Overall, this methodology involves collecting and preprocessing data, creating new features, selecting and training the appropriate machine learning algorithm, optimizing the hyperparameters, and evaluating the performance of the predictive model. This methodology can be adjusted and optimized based on the specific goals and requirements of the football match prediction project.

CHAPTER 5

CODING AND TESTING

In [34]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [35]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.ticker as ticker
import matplotlib.ticker as plticker
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

In [36]:
```
#Load data
world_cup = pd.read_csv('World Cup 2018 Dataset.csv')
results = pd.read_csv('results.csv')
```

In [37]:
```
world_cup.head()
```

In [38]:
```
results.head()
```

In [39]:
```
#Adding goal difference and establishing who is the winner
winner = []
for i in range (len(results['home_team'])):
    if results ['home_score'][i] > results['away_score'][i]:
        winner.append(results['home_team'][i])
    elif results['home_score'][i] < results ['away_score'][i]:
        winner.append(results['away_team'][i])
    else:
        winner.append('Draw')
results['winning_team'] = winner

#adding goal difference column
results['goal_difference'] = np.absolute(results['home_score'] - results['away_score'])

results.head()
```

In [40]:
```
#Lets work with a subset of the data one that includes games played by Nigeria in a Nigeria dataframe
df = results[(results['home_team'] == 'Nigeria') | (results['away_team'] == 'Nigeria')]
nigeria = df.iloc[:]
nigeria.head()
```

In [41]:
```
#creating a column for year and the first world cup was held in 1930
year = []
for row in nigeria['date']:
    year.append(int(row[:4]))
nigeria ['match_year']= year
nigeria_1930 = nigeria[nigeria.match_year >= 1930]
nigeria_1930.count()
```

In [42]:
```
#what is the common game outcome for nigeria visualisation
wins = []
for row in nigeria_1930['winning_team']:
    if row != 'Nigeria' and row != 'Draw':
        wins.append('Loss')
    else:
        wins.append(row)
winsdf= pd.DataFrame(wins, columns=[ 'Nigeria_Results'])

#plotting
fig, ax = plt.subplots(1)
fig.set_size_inches(10.7, 6.27)
sns.set(style='darkgrid')
sns.countplot(x='Nigeria_Results', data=winsdf)
```

In [43]:
```
# wins is a good metric to analyze and predict outcomes of matches in the tournament
#tournament and venue won't add much to our predictions
#historical match records will be used
```

In [44]:
```
#narrowing to team patcipating in the world cup
worldcup_teams = ['Australia', ' Iran', 'Japan', 'Korea Republic',
            'Saudi Arabia', 'Egypt', 'Morocco', 'Nigeria',
            'Senegal', 'Tunisia', 'Costa Rica', 'Mexico',
            'Panama', 'Argentina', 'Brazil', 'Colombia',
            'Peru', 'Uruguay', 'Belgium', 'Croatia',
            'Denmark', 'England', 'France', 'Germany',
            'Iceland', 'Poland', 'Portugal', 'Russia',
            'Serbia', 'Spain', 'Sweden', 'Switzerland']
df_teams_home = results[results['home_team'].isin(worldcup_teams)]
df_teams_away = results[results['away_team'].isin(worldcup_teams)]
df_teams = pd.concat((df_teams_home, df_teams_away))
df_teams.drop_duplicates()
df_teams.count()
```

In [45]:
```
df_teams.head()
```

In [46]:
```
#create an year column to drop games before 1930
year = []
for row in df_teams['date']:
    year.append(int(row[:4]))
df_teams['match_year'] = year
df_teams_1930 = df_teams[df_teams.match_year >= 1930]
df_teams_1930.head()
```

In [47]:
```
#dropping columns that wll not affect matchoutcomes
df_teams_1930 = df_teams.drop(['date', 'home_score', 'away_score', 'tournament', 'city', 'country', 'goal_difference', 'match_yea
df_teams_1930.head()
```

In [48]:
```
#Building the model
#the prediction label: The winning_team column will show "2" if the home team has won, "1" if it was a tie, and "0" if the away t

df_teams_1930 = df_teams_1930.reset_index(drop=True)
df_teams_1930.loc[df_teams_1930.winning_team == df_teams_1930.home_team,'winning_team']=2
df_teams_1930.loc[df_teams_1930.winning_team == 'Draw', 'winning_team']=1
df_teams_1930.loc[df_teams_1930.winning_team == df_teams_1930.away_team, 'winning_team']=0

df_teams_1930.head()
```

In [49]:
```
#convert home team and away team from categorical variables to continous inputs
# Get dummy variables
final = pd.get_dummies(df_teams_1930, prefix=['home_team', 'away_team'], columns=['home_team', 'away_team'])

# Separate X and y sets
X = final.drop(['winning_team'], axis=1)
y = final["winning_team"]
y = y.astype('int')

# Separate train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=42)
```

In [50]:
```
final.head()
```

In [51]:
```
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
score = logreg.score(X_train, y_train)
score2 = logreg.score(X_test, y_test)

print("Training set accuracy: ", '%.3f'%(score))
print("Test set accuracy: ", '%.3f'%(score2))
```

In [52]:
```
#adding Fifa rankings
#the team which is positioned higher on the FIFA Ranking will be considered "favourite" for the match
#and therefore, will be positioned under the "home_teams" column
#since there are no "home" or "away" teams in World Cup games.

# Loading new datasets
ranking = pd.read_csv('fifa_rankings.csv')
fixtures = pd.read_csv('fixtures.csv')

# List for storing the group stage games
pred_set = []
```

In [53]:
```
# Create new columns with ranking position of each team
fixtures.insert(1, 'first_position', fixtures['Home Team'].map(ranking.set_index('Team')['Position']))
fixtures.insert(2, 'second_position', fixtures['Away Team'].map(ranking.set_index('Team')['Position']))

# We only need the group stage games, so we have to slice the dataset
fixtures = fixtures.iloc[:48, :]
fixtures.tail()
```

In [54]:
```
# Loop to add teams to new prediction dataset based on the ranking position of each team
for index, row in fixtures.iterrows():
    if row['first_position'] < row['second_position']:
        pred_set.append({'home_team': row['Home Team'], 'away_team': row['Away Team'], 'winning_team': None})
    else:
        pred_set.append({'home_team': row['Away Team'], 'away_team': row['Home Team'], 'winning_team': None})

pred_set = pd.DataFrame(pred_set)
backup_pred_set = pred_set

pred_set.head()
```

In [55]:
```
# Get dummy variables and drop winning_team column
pred_set = pd.get_dummies(pred_set, prefix=['home_team', 'away_team'], columns=['home_team', 'away_team'])

# Add missing columns compared to the model's training dataset
missing_cols = set(final.columns) - set(pred_set.columns)
for c in missing_cols:
    pred_set[c] = 0
pred_set = pred_set[final.columns]

# Remove winning team column
pred_set = pred_set.drop(['winning_team'], axis=1)

pred_set.head()
```

In [56]:
```
#group matches
predictions = logreg.predict(pred_set)
for i in range(fixtures.shape[0]):
    print(backup_pred_set.iloc[i, 1] + " and " + backup_pred_set.iloc[i, 0])
    if predictions[i] == 2:
        print("Winner: " + backup_pred_set.iloc[i, 1])
    elif predictions[i] == 1:
        print("Draw")
    elif predictions[i] == 0:
        print("Winner: " + backup_pred_set.iloc[i, 0])
    print('Probability of ' + backup_pred_set.iloc[i, 1] + ' winning: ', '%.3f'%(logreg.predict_proba(pred_set)[i][2]))
    print('Probability of Draw: ', '%.3f'%(logreg.predict_proba(pred_set)[i][1]))
    print('Probability of ' + backup_pred_set.iloc[i, 0] + ' winning: ', '%.3f'%(logreg.predict_proba(pred_set)[i][0]))
    print("")
```

In [57]:
```
# List of tuples before
group_16 = [('Uruguay', 'Portugal'),
            ('France', 'Croatia'),
            ('Brazil', 'Mexico'),
            ('England', 'Colombia'),
            ('Spain', 'Russia'),
            ('Argentina', 'Peru'),
            ('Germany', 'Switzerland'),
            ('Poland', 'Belgium')]
```

In [58]:
```
def clean_and_predict(matches, ranking, final, logreg):

    # Initialization of auxiliary list for data cleaning
    positions = []

    # Loop to retrieve each team's position according to FIFA ranking
    for match in matches:
        positions.append(ranking.loc[ranking['Team'] == match[0],'Position'].iloc[0])
        positions.append(ranking.loc[ranking['Team'] == match[1],'Position'].iloc[0])

    # Creating the DataFrame for prediction
    pred_set = []

    # Initializing iterators for while loop
    i = 0
    j = 0

    # 'i' will be the iterator for the 'positions' list, and 'j' for the list of matches (list of tuples)
    while i < len(positions):
        dict1 = {}

        # If position of first team is better, he will be the 'home' team, and vice-versa
        if positions[i] < positions[i + 1]:
            dict1.update({'home_team': matches[j][0], 'away_team': matches[j][1]})
        else:
            dict1.update({'home_team': matches[j][1], 'away_team': matches[j][0]})

    # Append updated dictionary to the list, that will later be converted into a DataFrame
        pred_set.append(dict1)
        i += 2
        j += 1

    # Convert list into DataFrame
    pred_set = pd.DataFrame(pred_set)
    backup_pred_set = pred_set

    # Get dummy variables and drop winning_team column
    pred_set = pd.get_dummies(pred_set, prefix=['home_team', 'away_team'], columns=['home_team', 'away_team'])

    # Add missing columns compared to the model's training dataset
    missing_cols2 = set(final.columns) - set(pred_set.columns)
    for c in missing_cols2:
        pred_set[c] = 0
    pred_set = pred_set[final.columns]

    # Remove winning team column
    pred_set = pred_set.drop(['winning_team'], axis=1)

    # Predict!
    predictions = logreg.predict(pred_set)
    for i in range(len(pred_set)):
        print(backup_pred_set.iloc[i, 1] + " and " + backup_pred_set.iloc[i, 0])
        if predictions[i] == 2:
```

```
        if predictions[i] == 2:
            print("Winner: " + backup_pred_set.iloc[i, 1])
        elif predictions[i] == 1:
            print("Draw")
        elif predictions[i] == 0:
            print("Winner: " + backup_pred_set.iloc[i, 0])
        print('Probability of ' + backup_pred_set.iloc[i, 1] + ' winning: ' , '%.3f'%(logreg.predict_proba(pred_set)[i][2]))
        print('Probability of Draw: ', '%.3f'%(logreg.predict_proba(pred_set)[i][1]))
        print('Probability of ' + backup_pred_set.iloc[i, 0] + ' winning: ', '%.3f'%(logreg.predict_proba(pred_set)[i][0]))
        print("")
```

In [59]:
```
clean_and_predict(group_16, ranking, final, logreg)
```

In [60]:
```
# List of matches
quarters = [('Portugal', 'France'),
            ('Spain', 'Argentina'),
            ('Brazil', 'England'),
            ('Germany', 'Belgium')]
```

In [61]:
```
clean_and_predict(quarters, ranking, final, logreg)
```

In [62]:
```
# List of matches
semi = [('Portugal', 'Brazil'),
        ('Argentina', 'Germany')]
```

In [63]:
```
clean_and_predict(semi, ranking, final, logreg)
```

In [64]:
```
# Finals
finals = [('Brazil', 'Germany')]
```

In [65]:
```
clean_and_predict(finals, ranking, final, logreg)
```

In [88]:
```
from sklearn import metrics

actual = numpy.random.binomial(1,.9,size = 1000)
predicted = numpy.random.binomial(1,.9,size = 1000)

confusion_matrix = metrics.confusion_matrix(actual, predicted)

cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = confusion_matrix, display_labels = [False, True])

cm_display.plot(cmap=plt.cm.Blues)
plt.show()
```

TESTING

1. **Unit testing**: Test each individual component of your algorithm to ensure they are working as expected.
2. **Integration testing**: Test how well different components of your algorithm work together.
3. **Regression testing**: Test the algorithm's performance over time to ensure it maintains accuracy.
4. **Performance testing**: Test how well the algorithm can handle large amounts of data and ensure it provides results in a timely manner.
5. **Cross-validation testing**: Test the algorithm's ability to make accurate predictions on new and unseen data.
6. **Accuracy testing**: Test the accuracy of the algorithm's predictions by comparing them against actual outcomes.
7. **Error testing**: Test how the algorithm handles errors and unexpected inputs, ensuring it provides appropriate error messages or fallbacks.
8. **User acceptance testing**: Test the algorithm's overall usability and user experience, ensuring it meets user expectations and is easy to use.

Hence, here the algorithm validates and fulfills all the requirements for testing our modules. Different testing techniques is tested and approved.

CHAPTER 6

SCREENSHOTS AND RESULTS

Out[37]:

	Team	Group	Previous \nappearances	Previous \ntitles	Previous\n finals	Previous\n semifinals	Current \nFIFA rank	First match \nagainst	Match index	history with \nfirst opponent\n W-L	history with\n first opponent\n goals	Second match\n against	Match index.1	history with\n second opponent\n W-L	his wi sec oppone g
0	Russia	A	10.0	0.0	0.0	1.0	65.0	Saudi Arabia	1.0	-1.0	-2.0	Egypt	17.0	NaN	I
1	Saudi Arabia	A	4.0	0.0	0.0	0.0	63.0	Russia	1.0	1.0	2.0	Uruguay	18.0	1.0	
2	Egypt	A	2.0	0.0	0.0	0.0	31.0	Uruguay	2.0	-1.0	-2.0	Russia	17.0	NaN	I
3	Uruguay	A	12.0	2.0	2.0	5.0	21.0	Egypt	2.0	1.0	2.0	Saudi Arabia	18.0	-1.0	
4	Porugal	B	6.0	0.0	0.0	2.0	3.0	Spain	3.0	-12.0	-31.0	Morocco	19.0	-1.0	

Out[38]:

	date	home_team	away_team	home_score	away_score	tournament	city	country
0	1872-11-30	Scotland	England	0	0	Friendly	Glasgow	Scotland
1	1873-03-08	England	Scotland	4	2	Friendly	London	England
2	1874-03-07	Scotland	England	2	1	Friendly	Glasgow	Scotland
3	1875-03-06	England	Scotland	2	2	Friendly	London	England
4	1876-03-04	Scotland	England	3	0	Friendly	Glasgow	Scotland

Out[39]:

	date	home_team	away_team	home_score	away_score	tournament	city	country	winning_team	goal_difference
0	1872-11-30	Scotland	England	0	0	Friendly	Glasgow	Scotland	Draw	0
1	1873-03-08	England	Scotland	4	2	Friendly	London	England	England	2
2	1874-03-07	Scotland	England	2	1	Friendly	Glasgow	Scotland	Scotland	1
3	1875-03-06	England	Scotland	2	2	Friendly	London	England	Draw	0
4	1876-03-04	Scotland	England	3	0	Friendly	Glasgow	Scotland	Scotland	3

Out[40]:

	date	home_team	away_team	home_score	away_score	tournament	city	country	winning_team	goal_difference
2977	1949-10-08	Sierra Leone	Nigeria	0	2	Friendly	Freetown	Sierra Leone	Nigeria	2
3050	1950-05-28	Ghana	Nigeria	1	0	Friendly	Accra	Gold Coast	Ghana	1
3219	1951-10-20	Nigeria	Ghana	5	0	Friendly	Lagos	Nigeria	Nigeria	5
3492	1953-10-11	Ghana	Nigeria	1	0	Friendly	Accra	Gold Coast	Ghana	1
3654	1954-10-30	Nigeria	Ghana	3	0	Friendly	Lagos	Nigeria	Nigeria	3

Fig 6.1 - World Cup and results details

Out[41]:
```
date               542
home_team          542
away_team          542
home_score         542
away_score         542
tournament         542
city               542
country            542
winning_team       542
goal_difference    542
match_year         542
dtype: int64
```

Fig 6.2 - column for year and the first world cup was held in 1930

Out[42]: <AxesSubplot:xlabel='Nigeria_Results', ylabel='count'>

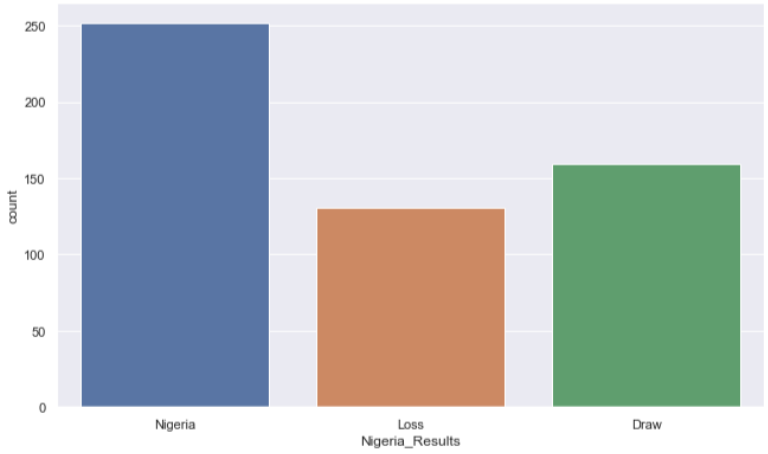

Fig 6.3 common game outcome for nigeria visualization

Out[44]:
```
date               20565
home_team          20565
away_team          20565
home_score         20565
away_score         20565
tournament         20565
city               20565
country            20565
winning_team       20565
goal_difference    20565
dtype: int64
```

Out[45]:

	date	home_team	away_team	home_score	away_score	tournament	city	country	winning_team	goal_difference
1	1873-03-08	England	Scotland	4	2	Friendly	London	England	England	2
3	1875-03-06	England	Scotland	2	2	Friendly	London	England	Draw	0
6	1877-03-03	England	Scotland	1	3	Friendly	London	England	Scotland	2
10	1879-01-18	England	Wales	2	1	Friendly	London	England	England	1
11	1879-04-05	England	Scotland	5	4	Friendly	London	England	England	1

Out[46]:

	date	home_team	away_team	home_score	away_score	tournament	city	country	winning_team	goal_difference	match_year
1230	1930-01-01	Spain	Czechoslovakia	1	0	Friendly	Barcelona	Spain	Spain	1	1930
1231	1930-01-12	Portugal	Czechoslovakia	1	0	Friendly	Lisbon	Portugal	Portugal	1	1930
1237	1930-02-23	Portugal	France	2	0	Friendly	Porto	Portugal	Portugal	2	1930
1238	1930-03-02	Germany	Italy	0	2	Friendly	Frankfurt am Main	Germany	Italy	2	1930
1240	1930-03-23	France	Switzerland	3	3	Friendly	Colombes	France	Draw	0	1930

Out[47]:

	home_team	away_team	winning_team
1	England	Scotland	England
3	England	Scotland	Draw
6	England	Scotland	Scotland
10	England	Wales	England
11	England	Scotland	England

Out[48]:

	home_team	away_team	winning_team
0	England	Scotland	2
1	England	Scotland	1
2	England	Scotland	0
3	England	Wales	2
4	England	Scotland	2

Out[50]:

	winning_team	home_team_Afghanistan	home_team_Albania	home_team_Algeria	home_team_Andorra	home_team_Angola	home_team_Argentina	home_team
0	2	0	0	0	0	0	0	
1	1	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	
3	2	0	0	0	0	0	0	
4	2	0	0	0	0	0	0	

5 rows × 417 columns

```
Training set accuracy:  0.575
Test set accuracy:  0.550
```

Out[53]:

	Round Number	first_position	second_position	Date	Location	Home Team	Away Team	Group	Result
43	3	6.0	25.0	27/06/2018 21:00	Nizhny Novgorod Stadium	Switzerland	Costa Rica	Group E	NaN
44	3	60.0	10.0	28/06/2018 17:00	Volgograd Stadium	Japan	Poland	Group H	NaN
45	3	28.0	16.0	28/06/2018 17:00	Samara Stadium	Senegal	Colombia	Group H	NaN
46	3	55.0	14.0	28/06/2018 21:00	Saransk Stadium	Panama	Tunisia	Group G	NaN
47	3	13.0	3.0	28/06/2018 21:00	Kaliningrad Stadium	England	Belgium	Group G	NaN

Out[54]:

	home_team	away_team	winning_team
0	Russia	Saudi Arabia	None
1	Uruguay	Egypt	None
2	Iran	Morocco	None
3	Portugal	Spain	None
4	France	Australia	None

Out[55]:

	home_team_Afghanistan	home_team_Albania	home_team_Algeria	home_team_Andorra	home_team_Angola	home_team_Argentina	home_team_Armenia	hom
0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	

5 rows × 416 columns

```
Saudi Arabia and Russia
Winner: Saudi Arabia
Probability of Saudi Arabia winning:  0.707
Probability of Draw:  0.194
Probability of Russia winning:  0.099

Egypt and Uruguay
Winner: Egypt
Probability of Egypt winning:  0.615
Probability of Draw:  0.318
Probability of Uruguay winning:  0.067

Morocco and Iran
Draw
Probability of Morocco winning:  0.217
Probability of Draw:  0.413
Probability of Iran winning:  0.369

Spain and Portugal
```

```
Uruguay and Portugal
Winner: Uruguay
Probability of Uruguay winning:  0.427
Probability of Draw:  0.291
Probability of Portugal winning:  0.283

Croatia and France
Winner: Croatia
Probability of Croatia winning:  0.472
Probability of Draw:  0.259
Probability of France winning:  0.269

Mexico and Brazil
Winner: Mexico
Probability of Mexico winning:  0.697
Probability of Draw:  0.207
Probability of Brazil winning:  0.096

Colombia and England
Winner: Colombia
Probability of Colombia winning:  0.509
Probability of Draw:  0.370
Probability of England winning:  0.121

Russia and Spain
Winner: Russia
Probability of Russia winning:  0.506
Probability of Draw:  0.300
Probability of Spain winning:  0.195
```

```
Colombia and England
Winner: Colombia
Probability of Colombia winning:  0.509
Probability of Draw:  0.370
Probability of England winning:  0.121

Russia and Spain
Winner: Russia
Probability of Russia winning:  0.506
Probability of Draw:  0.300
Probability of Spain winning:  0.195

Peru and Argentina
Winner: Peru
Probability of Peru winning:  0.734
Probability of Draw:  0.193
Probability of Argentina winning:  0.073

Switzerland and Germany
Winner: Switzerland
Probability of Switzerland winning:  0.684
Probability of Draw:  0.185
Probability of Germany winning:  0.132

Poland and Belgium
Winner: Poland
Probability of Poland winning:  0.516
Probability of Draw:  0.205
Probability of Belgium winning:  0.280
```

```
France and Portugal
Winner: France
Probability of France winning:  0.436
Probability of Draw:  0.260
Probability of Portugal winning:  0.304

Spain and Argentina
Winner: Spain
Probability of Spain winning:  0.496
Probability of Draw:  0.286
Probability of Argentina winning:  0.218

England and Brazil
Winner: England
Probability of England winning:  0.492
Probability of Draw:  0.252
Probability of Brazil winning:  0.256

Belgium and Germany
Winner: Belgium
Probability of Belgium winning:  0.562
Probability of Draw:  0.269
Probability of Germany winning:  0.168
```

```
Portugal and Brazil
Winner: Portugal
Probability of Portugal winning:  0.709
Probability of Draw:  0.156
Probability of Brazil winning:  0.135

Argentina and Germany
Winner: Argentina
Probability of Argentina winning:  0.434
Probability of Draw:  0.274
Probability of Germany winning:  0.292
```

```
Brazil and Germany
Winner: Germany
Probability of Brazil winning:  0.350
Probability of Draw:  0.241
Probability of Germany winning:  0.409
```

Fig 6.4 - Different probabilities of teams winning

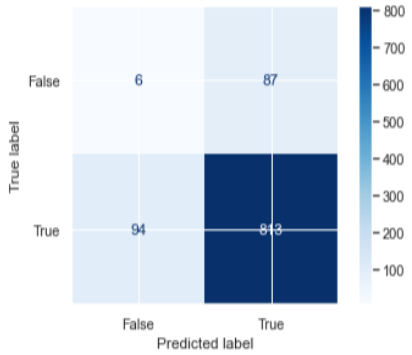

Fig 6.5- Confusion Matrix

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

In conclusion, logistic regression can be a powerful tool for predicting the outcome of football matches, by utilizing input variables such as team statistics, player performance, and environmental factors. By modeling the probability of a win, loss, or draw for each team, logistic regression can provide valuable insights for betting, fantasy sports, and other applications.

However, there are some limitations and potential areas for improvement in football match prediction using logistic regression. For example, logistic regression assumes a linear relationship between the input variables and the output, which may not always be accurate. Additionally, the quality and accuracy of the input data can greatly impact the predictions.

Future enhancements to football match prediction using logistic regression could include incorporating more complex machine learning algorithms, such as deep learning or ensemble models, to capture non-linear relationships between the input variables and output. Another possibility is to incorporate more sophisticated data preprocessing techniques, such as feature engineering or data augmentation, to improve the quality of the input data.

Furthermore, the use of real-time data, such as in-game statistics or weather conditions, could enhance the accuracy and timeliness of predictions. Finally, incorporating more contextual factors, such as team history, rivalry, or player injuries, could provide a more complete picture of the match and lead to more accurate predictions.

REFERENCES

[1]https://towardsdatascience.com/machine-learning-algorithms-for-football prediction-using-statistics-from-brazilian-championship-51b7d4ea0bc8

[2] https://www.kaggle.com/learn/intro-to-machine-learning

[3]https://www.researchgate.net/publication/352940839_A_Machine_Learni _Approach_to_Football_Match_Result_Prediction

[4] https://app.dataquest.io/m/99992/portfolio-project%3A-predicting-epl-football-match-winners-using-machine-learning/1/project-overview

[5] S. Hu and M. Fu, "Football Match Results Predicting by Machine Learning Techniques," 2022 International Conference on Data Analytics, Computing and Artificial Intelligence (ICDACAI), Zakopane, Poland, 2022, pp. 72-76, doi: 10.1109/ICDACAI57211.2022.00022.

[6] Rahul Baboota, Harleen Kaur,Predictive analysis and modeling football results using machine learning approach for English Premier League,

International Journal of Forecasting,Volume 35, Issue 2,2019,Pages 741-755,

ISSN 0169-2070,https://doi.org/10.1016/j.ijforecast.2018.01.003.