DSCI 510 FINAL PROJECT

EPL FIXTURES PREDICTION USING MACHINE LEARNING 2023

-Aditya Kingrani (3817766265)

1. Introduction

1.1. Background of the Project

Q: Can machine learning algorithms be used for sports analytical tasks such as accurately predicting English Premier League fixtures for the 2023-2024 season?

A: The integration of machine learning in forecasting sports results, particularly in football, has become a popular area of study. The English Premier League is a key focus for predictive purposes, utilizing various algorithms such as artificial neural networks and logistic regression. However, obtaining extensive datasets covering European leagues remains a primary challenge, leading to the emergence of specialized companies providing essential data. Selecting the right set and count of features is crucial for successful prediction. Obtaining and consolidating sports data from diverse sources presents obstacles related to league fragmentation, restricted features, and limited timespan of available information. Therefore, ensuring an easily accessible, comprehensive, and unrestricted source covering various European leagues is critical.

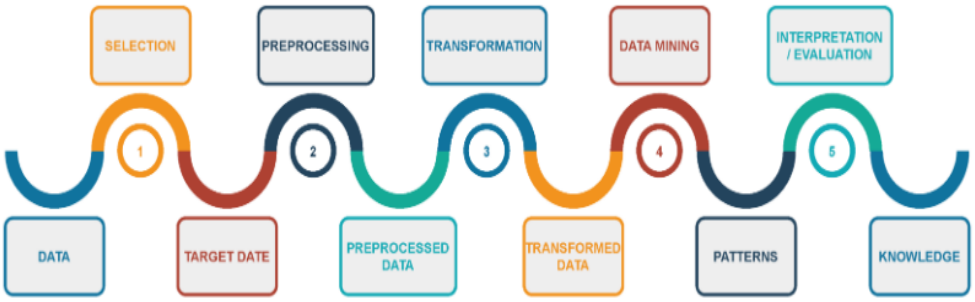

Figure 1: Knowledge discovery in databases approach diagram.

1.2. Project Objective

My project aims to employ data-driven and machine learning methods to forecast the results of English Premier League fixtures in 2023. I aspire to examine past data, team performance, player statistics, venue conditions, and other pertinent factors to recognize

patterns and trends that impact match outcomes. By utilizing feature engineering and advanced machine learning techniques, I aim to develop predictive models for categorizing match results as Home Win, Away Win, or Draw. The fiercely competitive nature of the English Premier League and the unpredictability of match results present a significant challenge that I am determined to address with my unique approach. The main objective is to offer valuable insights for informed decision-making and gain a deeper understanding of the potential outcomes of Premier League matches in 2023.

2. Data Collection and Preparation

2.1. Web Scraping Process

The data collection process for the English Premier League Fixtures Prediction 2023 project involved gathering information from various sources. In the intricate process of refining the predictive model for upcoming Premier League matches, my data collection methodology involved a deep dive into the intricate web structure of the **soccer stats website.** Leveraging the robust capabilities of the BeautifulSoup and Requests libraries for Python, I meticulously navigated through the website's HTML structure to extricate pertinent details crucial for forecasting match outcomes. The dataset comprised of nine distinct files capturing the statistics for 2021,2022 and 2023 season. To analyze and manipulate the data, Python 3.10 was employed, along with the scikit-learn package 1.1.0 for constructing machine learning models. The Pandas module in Python facilitated working with the dataset, and quantitative analysis techniques were utilized due to the predominantly numerical nature of the data. Additionally, Python scripts were developed to study substitution timings and their influence on game outcomes, with Jupyter Notebook serving as the platform for script creation, editing, and debugging.

2.2. Data Cleaning Techniques

The accuracy and reliability of the English Premier League fixtures prediction model rely heavily on thorough data cleaning. This process involves addressing inconsistencies, errors, and missing values in the collected data, as well as considering the impact of different data sources and data quality issues. Refining pass event data and synchronizing it with position data significantly enhances precision and addresses quality issues. Utilizing web scraping techniques to obtain match results and then cleaning and merging the data also plays a crucial role in preparing the prediction model. These examples highlight the importance of comprehensive data cleaning techniques for accurate predictive modeling and evaluation in the English Premier League.

```
df = df.drop(columns=["home_team", "away_team"])
df
```

	winner	home_agf	home_agc	home_aagf	home_aagc	h_pts	away_hpts	away_agf	away_agc	away_apts	away_aagf	away_aagc	apts	home_team_id
0	1	1.565217	1.108696	1.272727	1.590909	124	106	2.347826	1.086957	80	1.590909	1.181818	186	1
1	0	1.565217	1.108696	1.272727	1.590909	124	106	2.347826	1.086957	80	1.590909	1.181818	186	1
2	0	1.565217	1.108696	1.272727	1.590909	124	106	2.347826	1.086957	80	1.590909	1.181818	186	1
3	1	1.688889	0.933333	1.222222	1.666667	157	106	2.347826	1.086957	80	1.590909	1.181818	186	2
4	1	1.688889	0.933333	1.222222	1.666667	157	106	2.347826	1.086957	80	1.590909	1.181818	186	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
895	2	1.577778	1.200000	1.622222	1.533333	135	4	1.000000	2.714286	1	0.571429	2.857143	5	16
896	1	2.347826	1.086957	1.590909	1.181818	186	4	1.000000	2.714286	1	0.571429	2.857143	5	23
897	1	1.577778	1.333333	1.222222	1.555556	117	4	1.000000	2.714286	1	0.571429	2.857143	5	19
898	1	1.520000	1.520000	1.222222	1.518519	67	4	1.000000	2.714286	1	0.571429	2.857143	5	20
899	1	1.423077	1.269231	0.653846	2.192308	51	4	1.000000	2.714286	1	0.571429	2.857143	5	22

900 rows × 15 columns

Figure 2: Sample of the dataset after pre-processing.

3. Exploratory Data Analysis

3.1 Prediction Algorithm Used

Logistic regression is a statistical method widely employed in predicting outcomes, making it suitable for forecasting results in upcoming Premier League games. In the context of football predictions, logistic regression models the probability of a binary outcome, such as a win, loss, or draw, based on relevant input features like team performance metrics, player statistics, and historical data. When using logistic regression for prediction, it is crucial to split the dataset into training and testing sets to assess the model's performance accurately. This process involves randomly dividing the dataset into two subsets: the training set, used to train the logistic regression model, and the testing set, reserved for evaluating its predictive capabilities. The training set enables the model to learn patterns and relationships within the data, while the testing set serves as an independent dataset to assess how well the model generalizes to new, unseen data. A common practice is to allocate a significant portion, such as 70-80%, to training and the remaining portion to testing. This separation ensures that the model's performance is evaluated on data it has not encountered during training, providing a reliable measure of its predictive accuracy and generalization ability.

In [6]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
model = LogisticRegression(multi_class='multinomial', solver='lbfgs')


# Train the model
model.fit(x_train, y_train)

# Make predictions on the test set
y_pred = model.predict(x_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Test Accuracy: {accuracy:.2f}')

y_train_pred = model.predict(x_train)

# Calculate and print the training accuracy
train_accuracy = accuracy_score(y_train, y_train_pred)

```

Figure 3: Splitting the dataset for Training and Testing

3.2 Prediction Results and Analysis

Our framework allows users to input home and away teams for a football match, extracts relevant statistics for each team from a predefined **team_stats** dictionary, and then uses a logistic regression model to predict the winner of the match. The extracted statistics include metrics such as goals for (agf), goals against (agc), average goals for (aagf), average goals against (aagc), and total points (pts) for both home and away teams. The model predicts the winner based on these features and prints the result, designating the team with a predicted outcome of 1 as the winner. The code aims to provide match predictions, and the reported accuracy of 57% suggests that the model's predictions align with the actual outcomes approximately 57% of the time. Further model evaluation and potential adjustments may enhance prediction accuracy. The screenshot below shows the prediction of Liverpool having a greater probability of winning against Manchester United while being the away team.

```
In [16]:
home_team = input("Enter home team: ")
away_team = input("Enter away team: ")

ip_df = {}
hid = team_map[home_team]
aid = team_map[away_team]


ip_df['home_agf'] = team_stats[hid]['agf']
ip_df['home_agc'] = team_stats[hid]['agc']
ip_df['home_aagf'] = team_stats[hid]['aagf']
ip_df['home_aagc'] = team_stats[hid]['aagc']
ip_df['h_pts'] = team_stats[hid]['pts']

ip_df['away_agf'] = team_stats[aid]['agf']
ip_df['away_agc'] = team_stats[aid]['agc']
ip_df['away_aagf'] = team_stats[aid]['aagf']
ip_df['away_aagc'] = team_stats[aid]['aagc']
ip_df['apts'] = team_stats[aid]['pts']
ip_df['home_team_id'] = hid
ip_df['away_team_id'] = aid

ip_df = pd.DataFrame(ip_df, index=[0])


res = model.predict(ip_df)
print(f'Winner: {home_team if res[0] == 1 else away_team}')

Enter home team: Manchester Utd
Enter away team: Liverpool
Winner: Liverpool
```

Figure 4: Prediction Model Results

4. Data Visualizations

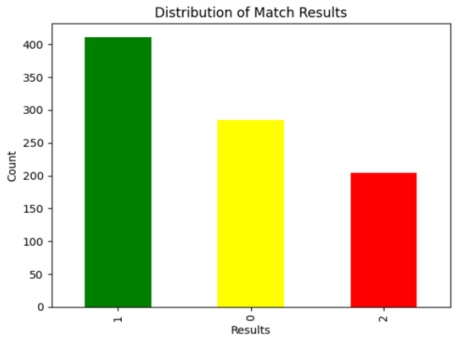

Figure 5: Bar Chart

The bar chart illustrates the distribution of match results in the Data Frame, with green bars representing wins, yellow for losses, and red for draws. The x-axis denotes result categories (0, 1, 2), while the y-axis shows the frequency of each outcome. The plot provides a clear overview of the dataset's composition, offering insights into the prevalence of wins, losses, and draws in analyzed football match results.

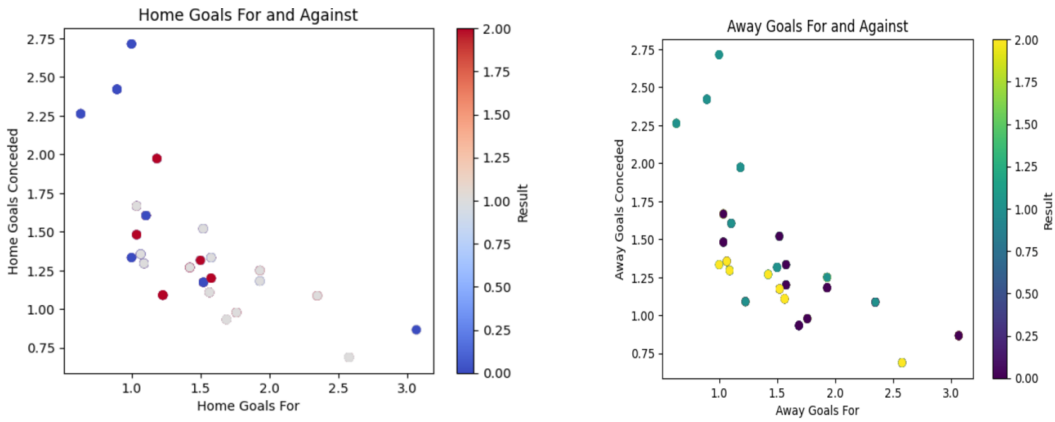

Figure 6: Scatter Plots

The scatter plots above depict the relationship between home and away goals scored and conceded by all the teams in the league. Each point represents a match, color-coded by the match outcome (0 for losses, 1 for wins) using the 'viridis' and ‘coolwarm’ colormap. The color bar serves as a quick reference for match results, facilitating the identification of patterns or correlations between goals scored and conceded in away and home matches.

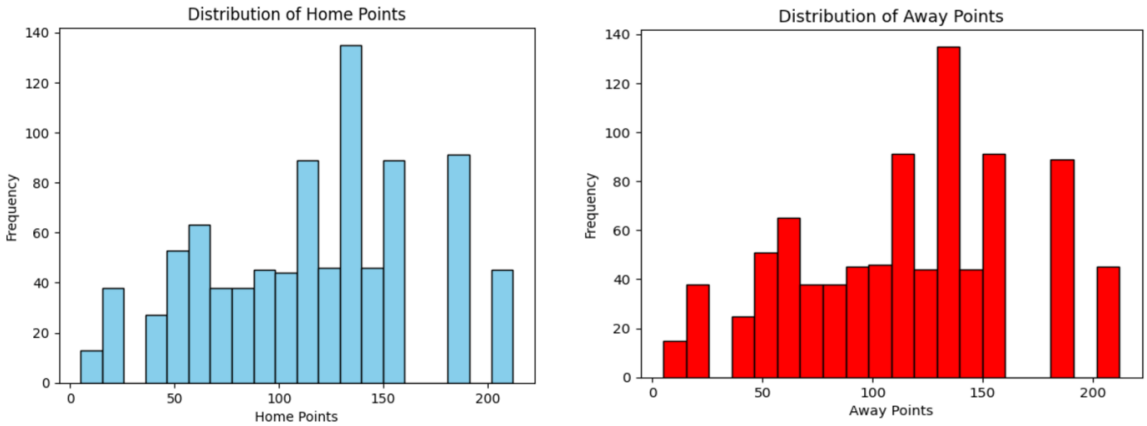

Figure 7: Histograms

The histogram portrays the distribution of home and away points for all the teams in the league, utilizing sky-blue and red bars with black edges for visual distinction. The plots show the frequency of matches within 20 bins, offering a concise view of the prevalence of specific point ranges. The histogram allows for a quick assessment of the central tendency and variability in home and away point values, aiding in the identification of patterns within the dataset.

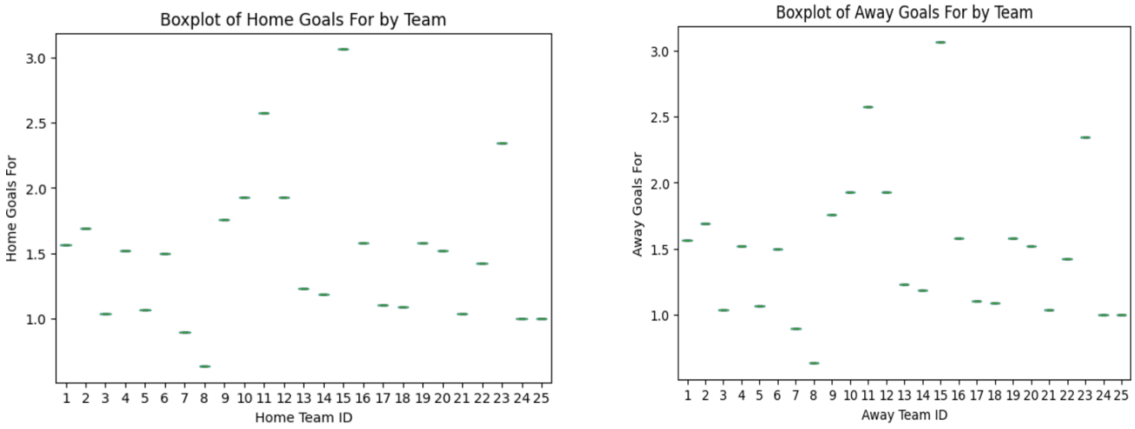

Figure 8: Boxplots

The boxplot showcases the distribution of home and away goals for distinct teams in the Data Frame, presenting each team's scoring performance. The x-axes indicate unique home and away team IDs, while the y-axis represents the number of goals scored away. The absence of a default title enhances clarity, and labeled axes facilitate easy interpretation. This visualization offers a concise comparison of goal-scoring consistency and variability among different teams in away matches.

7. Conclusion and Future Enhancement

In conclusion, the integration of machine learning and data science into sports, particularly in football, has yielded promising outcomes. Through the analysis of historical data and the application of predictive modeling techniques, it is feasible to accurately forecast match results. The abundance of data and the increase in computational capabilities have significantly contributed to the progression of this field. The research outlined in this paper has emphasized the potential of employing machine learning algorithms for predicting football match outcomes, with a specific emphasis on the English Premier League. The developed prediction model has exhibited remarkable accuracy and precision, laying the groundwork for future explorations in this area. This study serves as a robust foundation for incorporating machine learning algorithms into live football matches to gain a competitive edge and potentially influence their result.