

Submited to DERBEW.F

Machine Learning

Name Abdelah Nesredin

Id 1400751

Individual Assignemnt

Course Machine Learning

Title : English Primer League Football match Prediction based on historical data

Name

Abdelah Nesredin

Idnumber 1400751

Submitted to Derbew.F

**1,Problem Definition**

The goal of this project is to predict the outcome of football matches (win, lose, or draw) based on historical match data and team performance metrics. This is a multi-class classification problem, where the target variable is the match outcome (e.g., Home Win, Away Win, Draw). The model will help football analysts, fans, and bettors make informed predictions about future matches.

Why This Problem?

Football is one of the most popular sports globally, and predicting match outcomes has significant applications in sports analytics, betting, and team strategy.

By leveraging historical data, we can build a model that provides insights into factors influencing match results.**Problem Type**: This is a **classification problem** as we are trying to classify the match result into one of three categories.

· **Goal**: Predict the outcome of an English Premier League match (Home Win, Away Win, or Draw).

· **Type**: **Classification Problem** (categorizing the match result).

**2. Data Acquisition**

Dataset Source

Dataset Name: Engilsh Premier leauge Football Results from 1872 to 2017

Source: github link ;- [Premier-League/dataset/PremierLeague.csv at master · IvanRamosDataTech/Premier-League](https://github.com/IvanRamosDataTech/Premier-League/blob/master/dataset/PremierLeague.csv)

License: CC0 1.0 Universal (Public repository)

Description: The dataset contains Priemer leuage football match results from 1993 to 2017, including features such as date, home team, away team, home score, away score, tournament, and location.

Dataset Structure

Format: CSV

Size: ~11,978rows, 49 columns

**Features:**

date: Date of the match.

home\_team: Name of the home team.

away\_team: Name of the away team.

home\_score: Goals scored by the home team.

away\_score: Goals scored by the away team.

Head to head match :between each clubs

city: City where the match was played.

**Target Variable**

The target variable is derived from home\_score and away\_score:

**Data Quality**

The dataset is clean, with no missing values. I have cleaned it all and save it in cleaned\_data.csv.

### Data Understanding and Exploration:I have load the dataset into a pandas DataFrame and perform basic exploration (EDA) to better understand the data.

#### Steps:

1. **Load the CSV file into a pandas DataFrame.**
2. **Display the first few rows to inspect the data.**
3. **Examine the columns and data types.**

predicting football match outcomes in the English Premier League), identifying outliers could help you detect unusual data points that don't follow the expected patterns. Outliers in your dataset could negatively impact the performance of your machine learning models by distorting relationships between variables or skewing statistical analyses. Here's how outliers could affect your project and how you might handle them:

### 1. ****Outliers in the Numerical Columns:****

Some of My columns are numeric (e.g., FullTimeHomeTeamGoals, FullTimeAwayTeamGoals, HomeTeamShots, etc.). Outliers in these columns could represent unusual matches where the score or other statistics are far higher or lower than typical matches.

* For example, a match with an extremely high score like 9-0 may be an outlier, as most football matches have scores between 1-5 goals.
* Market-related columns (like MarketMaxHomeTeam, MarketAvgAwayTeam, etc.) could have outliers because of extreme market odds based on unusual events like a star player injury or team suspension.

### 2. ****Effect of Outliers:****

* **Distortion of Trends:** Outliers can skew averages or trends. For instance, if most teams score 1-3 goals per match, a match where a team scores 7 could distort the average or correlation calculations.
* **Modeling Impact:** Machine learning models could be influenced by outliers, as they might incorrectly assume that extreme values are the norm, which can affect the model's predictive accuracy.

****Handling Outliers:****

* **Remove Outliers:** I have find extreme outliers that don’t add value or are simply noise (e.g., extreme scores), you can remove those rows from the dataset. For example, a match with an unusually high score (e.g., 10-0) might be considered an anomaly and removed from the dataset.
* **Capping/Clipping:** In some cases, you might decide to cap the outliers at a specific threshold (e.g., scores above 5 are capped at 5).
* **Transformation:** You can also apply transformations like log or square root to reduce the effect of large outliers.

visualization

I have tried ti visualize the data accorndigly

### 1. ****Boxplot: FullTimeHomeTeamGoals by FullTimeResult****

* **Observation:** The boxplot shows the distribution of home team goals for different match outcomes (Home Win, Away Win, and Draw). Home wins (FullTimeResult = H) tend to have higher goal values, while away wins (FullTimeResult = A) generally have a wider range of goals.

### 2. ****Boxplot: FullTimeAwayTeamGoals by FullTimeResult****

* **Observation:** Away team goals are typically higher for away wins (FullTimeResult = A), and the distribution is generally more spread out. Home wins tend to show a lower range of goals scored by away teams.

### 3. ****Violin Plot: FullTimeHomeTeamGoals by FullTimeResult****

* **Observation:** The violin plot further confirms that home wins have a peak near higher goal values for home teams, while away wins have a broader distribution. Draws have a narrower distribution of home team goals.

### 4. ****Violin Plot: FullTimeAwayTeamGoals by FullTimeResult****

* **Observation:** For away team goals, away wins have a wider and higher range of values. Home wins have relatively lower and tighter distributions for away team goals, while draws show moderate goal values.

### 5. ****Histogram: FullTimeHomeTeamGoals****

* **Observation:** The histogram shows the distribution of home team goals. Most home teams scored between 1 and 3 goals, with a few extreme cases showing higher scores.

### 6. ****Histogram: FullTimeAwayTeamGoals****

* **Observation:** Away teams tend to score fewer goals overall, with most goals falling in the range of 0 to 2. There’s a noticeable peak around 1 goal, indicating that many matches feature low-scoring away teams.

### 7. ****Bar Plot: FullTimeResult by HomeTeam****

* **Observation:** This bar plot illustrates the match results based on the home team. Certain home teams (e.g., Arsenal, Manchester United) show more frequent home wins, while other teams are more likely to have away wins or draws.

### 8. ****Bar Plot: FullTimeResult by AwayTeam****

* **Observation:** Away team match results are shown here. Similar to the home team analysis, the away teams have varying success rates, with some teams (e.g., Liverpool) having a higher chance of winning away matches.

### 9. ****Count Plot: Distribution of FullTimeResult (Match Outcome)****

* **Observation:** This plot shows the overall distribution of match results. It likely reveals a balance between home wins, away wins, and draws, though home wins might be slightly more frequent.

### 10. ****Scatter Plot: FullTimeHomeTeamGoals vs HomeTeamPoints****

* **Observation:** The scatter plot shows a positive correlation between the number of goals scored by home teams and the points earned by them. Matches with higher goals scored generally lead to higher points (home wins).

### 11. ****Scatter Plot: FullTimeAwayTeamGoals vs AwayTeamPoints****

* **Observation:** A similar trend is observed for away teams, where higher away team goals correlate with more points earned (away wins).

### 12. ****Correlation Heatmap: Key Features****

* **Observation:** The heatmap reveals correlations between features. There’s likely a strong positive correlation between home goals and home points, as well as away goals and away points. However, the correlation between home and away goals might be weaker, showing that these are independent features.

### 13. ****Line Plot: Goals Over Time (Home Team vs Away Team)****

* **Observation:** The line plot helps identify trends over time. It could show that home and away team goals fluctuate seasonally or based on match weeks. For example, some teams might perform better during certain parts of the season.

### 14. ****Pair Plot: Goals and Points by Match Result****

* **Observation:** The pair plot visually highlights how the goals scored by home and away teams relate to the points earned and how these relationships vary based on the match result. Home and away team goals have a clear relationship with match outcomes, with home team goals showing stronger ties to home wins.

**General Observations:**

* **Home vs Away Performance:** Home teams tend to score more goals than away teams and have a higher chance of winning at home.
* **Goal Distribution:** Most goals scored by both home and away teams tend to be low, with a few outliers.
* **Match Outcome Trends:** Home wins seem to dominate, but away wins and draws still represent a significant portion of the data.
* **Correlation between Goals and Points:** There’s a strong correlation between goals scored by both home and away teams and the points earned (win/loss/draw).

Conclusion to all summary of EDA

### 1. ****Data Distribution:****

* **Description:** The data distribution refers to how the values in your dataset are spread out across the different features. Understanding this distribution is important as it helps you understand the range and commonality of different values.
* **Observation:** You may find that the goals scored by home and away teams tend to follow a normal distribution (bell curve) for most matches, with fewer occurrences of very high or very low goal scores. However, there may be some skewness in the data due to unusually high-scoring matches or more frequent low-scoring matches.
  + For example, home teams might score more often than away teams, leading to a distribution that’s skewed to the right for home team goals.
  + Market odds may also show patterns, with certain teams consistently favored for wins based on betting tendencies. For example, a high MarketMaxHomeTeam value could reflect that a certain home team is frequently favored by bookmakers due to its form or historical performance.

### 2. ****Missing Values:****

* **Description:** Missing values are entries in your dataset where data is not available. These missing values need to be addressed as they can hinder data analysis and affect model performance.
* **Observation:** Some columns may have missing values. For example, columns such as Time, Referee, or various market-related columns like MarketMaxHomeTeam might have significant missing percentages. You should specify which columns have missing data and the exact percentage of missing entries.
  + For example, you could notice that around **82%** of the Time column is missing, indicating a serious gap in match timing data, while others like Referee may have around **22%** missing, which might be less critical.

### 3. ****Outliers:****

* **Description:** Outliers are data points that differ significantly from the rest of the dataset. Outliers can be the result of errors or can represent unusual but legitimate events.
* **Observation:** Outliers may exist in certain columns like FullTimeHomeTeamGoals and FullTimeAwayTeamGoals, where matches with unusually high scores (e.g., 9-0 or 8-1) may skew the data. Market odds might also show extreme values when a match has unusual betting conditions, such as a star player being absent.
  + For example, a match with an extreme score (e.g., 6-0) could be considered an outlier. Similarly, market odds showing extreme favorability for one team, such as odds of 1.01 for the home team, could indicate an outlier based on the betting trend.

### 4. ****Relationships:****

* **Description:** Identifying relationships between features and the target variable (match outcomes) is key in predictive modeling. Correlations or associations between variables help us understand what influences the target variable and can guide feature selection.
* **Observation:** Strong correlations may be observed between features like market odds and match outcomes. For instance, a high MarketAvgHomeTeam could indicate a higher probability of a home team win, especially when combined with a positive correlation between home team goals and match wins.
  + You may also observe relationships where certain features like HomeTeamGoals or AwayTeamGoals have a strong impact on match results. For example, home team goals might correlate positively with home team victories, and away team goals might correlate with away team victories.
  + Market odds can also reveal interesting patterns. For example, if the odds for an away team are extremely high (indicating a high probability of losing), it might suggest the home team is favored, and this could correlate with a higher probability of home team victories.

### Overall Summary of the Task:

* You’ll explore the distribution of various features, identify any missing data and outliers, and document relationships between key variables. By performing this exploratory data analysis (EDA), you will gather valuable insights into how the data behaves and what influences match outcomes, which will inform the subsequent steps of preprocessing, feature engineering, and model building.

**Data Preprocessing**

**1. Clean Column Names**:Goal: Standardize column names for consistency.Action:Remove extra spaces using .str.strip().Replace spaces with underscores using .str.replace(' ', '\_').Convert all column names to lowercase using .str.lower().

Result: Clean, consistent column names (e.g., Home Team becomes home\_team)

1. **Handle Missing ValueGoal:** Remove columns with too many missing values.Action:Drop columns where more than 50% (threshold = 0.5) of the values are missing.df.isnull().mean() calculates the percentage of missing values in each column.Result: Dataset retains only columns with sufficient data.
2. **Handle OutliersGoal**: Remove extreme values (outliers) from numerical columns.Action:Use the Interquartile Range (IQR) method:Calculate Q1 (25th percentile) and Q3 (75th percentile).Define outliers as values outside [Q1 - 1.5 \* IQR, Q3 + 1.5 \* IQR].Remove rows containing outliers in any numerical column.Result: Dataset is free of extreme outliers in numerical features.

**4. Encode Categorical FeaturesGoal:** Convert categorical columns (e.g., team names) into numerical format for machine learning.Action:

Check if awayteam and hometeam columns exist.Use one-hot encoding (pd.get\_dummies) to convert categorical values into binary columns (e.g., awayteam\_TeamA, awayteam\_TeamB).drop\_first=True avoids multicollinearity by dropping the first category.Result: Categorical data is transformed into a format suitable for ML models.

**5. Scale/Normalize Numerical FeaturesGoal:** Standardize numerical features to have a mean of 0 and a standard deviation of 1.**Action:**Use StandardScaler from sklearn.preprocessing to scale numerical columns.Fit the scaler on the data and transform the numerical columns.Result: Numerical features are scaled, ensuring no single feature dominates due to its magnitude.6. Output the Final DatasetGoal: Save and inspect the cleaned, preprocessed dataset.Action:Save the final dataset to a CSV file (final\_dataset.csv).Print the first few rows of the dataset to the console for inspection.Result: The dataset is ready for model training.

**Summary**

This section performs data cleaning and preprocessing:

Cleans column names.

**Handles missing values.Removes outliers**.

**Encodes categorical features.**

**Scales numerical features.**

**Outputs the final dataset for training.**

**The final dataset is saved as final\_dataset.csv and displayed for inspection.**

**Model Training and Evaluation**

**A RandomForestClassifier:** is selected for the classification task based on its robustness and ability to handle complex relationships in data. ➔ The dataset is split into training (80%) and testing (20%) sets to evaluate model performance. ➔ The model is trained using the preprocessed training data, which includes engineered features such as head-to-head statistics, team form, and seasonal rankings. ➔ Hyperparameter tuning is performed using cross-validation techniques to optimize model performance. ➔ Key hyperparameters tuned include the number of trees (n\_estimators), maximum tree depth, and minimum samples per leaf. ➔ Challenges encountered include data imbalance and missing values, which were handled by appropriate preprocessing techniques.

**Its not the optimize method according my dataset I have to use xgboost algorithm**

Predictions are made on the testing dataset using the trained model, and results are analyzed for accuracy and reliability. ➔ The model's performance is evaluated using multiple metrics to ensure a comprehensive understanding:

**Accuracy:** Measures the percentage of correctly predicted outcomes.

**Precision and Recall:** Evaluate the correctness of positive predictions and sensitivity to different classes.

**F1-score:** Balances precision and recall to provide a holistic view of model performance.

**Confusion Matrix:** Visualizes true positive, false positive, and false negative rates. ➔ Various visualization techniques, including classification reports and confusion matrices, are used to interpret results and identify areas for improvement. ➔ Model performance is compared against a baseline dummy classifier to quantify its effectiveness in predictive analytics. ➔ A thorough analysis is conducted on the impact of individual features, examining their contribution to prediction accuracy and potential refinements for future iterations.

You can see all the anaysis in the notbook

**Model Deployment :-**

**Flask-based API** is implemented to facilitate real-time predictions, enhancing accessibility for users.

The API loads the trained model and processes POST requests containing match data (home and away teams, recent performance metrics, and rankings).

Predictions are returned in JSON format, mapping numerical outputs to intuitive match outcomes (Home Win, Draw, Away Win).

Cross-Origin Resource Sharing (CORS) is enabled to allow integration with external platforms and applications.

Error handling mechanisms are incorporated to ensure robustness and reliability during API usage.

I have Sucessfully Deployed it using render third party deployemnet you can test my api

Thank you