Predicting Football Match Outcomes

*Capstone Project 1 Research Paper Writeup

Connor Leyden†
Harvard University
Washington D.C.
United States of America
cleyden@college.harvard.edu

ChatGPT
OpenAI
Palo Alto, CA
United States of America
GPT-4o

Sigma
No Affiliation
No Location
Respect Please

***Abstract*—This paper presents a comprehensive study on predicting soccer match outcomes using machine learning techniques. The project encompasses various stages, including data collection from reliable sources, data cleaning and preprocessing to handle missing values and normalize data, and exploratory data analysis (EDA) using visualization libraries such as pandas, matplotlib, and seaborn. Feature engineering was performed to create meaningful features from raw data. Multiple supervised learning models, including logistic regression, random forest, and XGBoost, were selected for analysis. The models were trained and evaluated using metrics such as accuracy, precision, recall, and F1 score. Hyperparameter tuning techniques, such as grid search and random search, were employed to optimize model performance. The findings and insights were documented in a structured format using Jupyter Notebooks and Overleaf, adhering to the standard IEEE format. The results demonstrate significant improvements in prediction accuracy, highlighting the potential of data-driven approaches in sports analytics**

***Index Terms*—Soccer Match Prediction, Machine Learning, Data Preprocessing, Feature Engineering, Model Evaluation, Hyperparameter Tuning, Sports Analytics**

Fig. 1. Match between Manchester Teams

I. Introduction

Predicting the outcome of soccer matches has long been a subject of interest for researchers, sports analysts, and enthusiasts. Soccer, being one of the most popular sports worldwide, attracts a significant amount of attention, making accurate match predictions highly valuable for various stakeholders, including bettors, coaches, and fans. Traditionally, match predictions have relied on expert knowledge and simple statistical methods. However, the advent of machine learning has opened new avenues for improving prediction accuracy by leveraging large datasets and complex algorithms. Despite advancements in machine learning, accurately predicting soccer match outcomes remains a challenging task due to the inherent unpredictability and complexity of the sport. Variability in player performance, team dynamics, and external factors such as weather conditions contribute to the difficulty in making precise predictions.

The current state of soccer match prediction combines traditional statistical methods with modern machine learning techniques. Traditional approaches rely on expert analysis and simple models, but often miss the sport's complexity. Recent advancements in machine learning, such as logistic regression, decision trees, and neural networks, offer improved accuracy by processing large datasets and identifying complex patterns. Despite this progress, there's still room for improvement, particularly in integrating real-time data and optimizing model performance. This research aims to advance the field by leveraging these modern techniques to provide more accurate and insightful predictions.

The motivation behind this research is to explore the potential of machine learning techniques in enhancing the accuracy of soccer match predictions. By utilizing historical match data and sophisticated machine learning models, this study aims to uncover patterns and insights that traditional methods may overlook. Improved prediction accuracy can have significant implications for sports betting, team strategy development, and fan engagement. Previous studies have applied various machine learning algorithms, such as logistic regression, decision trees, and neural networks, to predict sports outcomes. These studies have demonstrated that machine learning can outperform traditional statistical methods in certain scenarios. However, there remains a need for further research to optimize model performance and explore the use of advanced techniques like ensemble learning and hyperparameter tuning.

In this research, I will develop two distinct models to predict soccer match outcomes. The first model will use only data

available before the match, such as recent form, betting odds, team standings, and head-to-head records. This model aims to provide predictions based on pre-match information that is typically available to analysts and bettors before the game begins. The second model will utilize exclusively in-match data, including statistics such as possession, shots on target, and other real-time performance metrics. This approach allows for a dynamic assessment of the match as it unfolds, providing insights that can be used during the game to predict the final outcome. By comparing these two models, I aim to evaluate the effectiveness of pre-match data versus in-match data in predicting soccer match results.

The primary objectives of this research are to collect and preprocess historical soccer match data from reliable sources, perform exploratory data analysis (EDA) and visualize key patterns in the data, engineer meaningful features from raw data that can enhance model performance, evaluate the performance of various supervised learning models in predicting match outcomes, optimize model performance through hyperparameter tuning, and document findings and insights in a structured format. This study follows a systematic approach, beginning with data collection and preprocessing to handle missing values and normalize data. Exploratory data analysis (EDA) is conducted to identify key patterns and trends. Feature engineering is employed to create meaningful features from raw data. Multiple supervised learning models, including logistic regression, random forest, and XGBoost, are trained and evaluated using metrics such as accuracy, precision, recall, and F1 score. Hyperparameter tuning techniques, such as grid search and random search, are utilized to optimize model performance. The findings are documented using Jupyter Notebooks and Overleaf, adhering to the standard IEEE format.

The unique contributions of this research include a comprehensive analysis of historical soccer match data using machine learning techniques, the development of a robust prediction model that outperforms traditional methods, and insights into the key factors influencing match outcomes. The rest of the paper is organized as follows:

- Section II discusses the related work and literature review.
- Section III describes the data collection and preprocessing steps.
- Section IV presents the exploratory data analysis and feature engineering process.
- Section V details the model selection, training, and evaluation for the pre-match dataset.
- Section VI covers the model selection, training, and evaluation for the match dataset.
- Section VII discusses the results and findings.
- Section VIII concludes the paper and suggests directions for future research.

By following this structured approach, the paper aims to provide a comprehensive understanding of the potential of machine learning in predicting soccer match outcomes and contribute to the field of sports analytics.

The flowchart below explains the process of this paper:

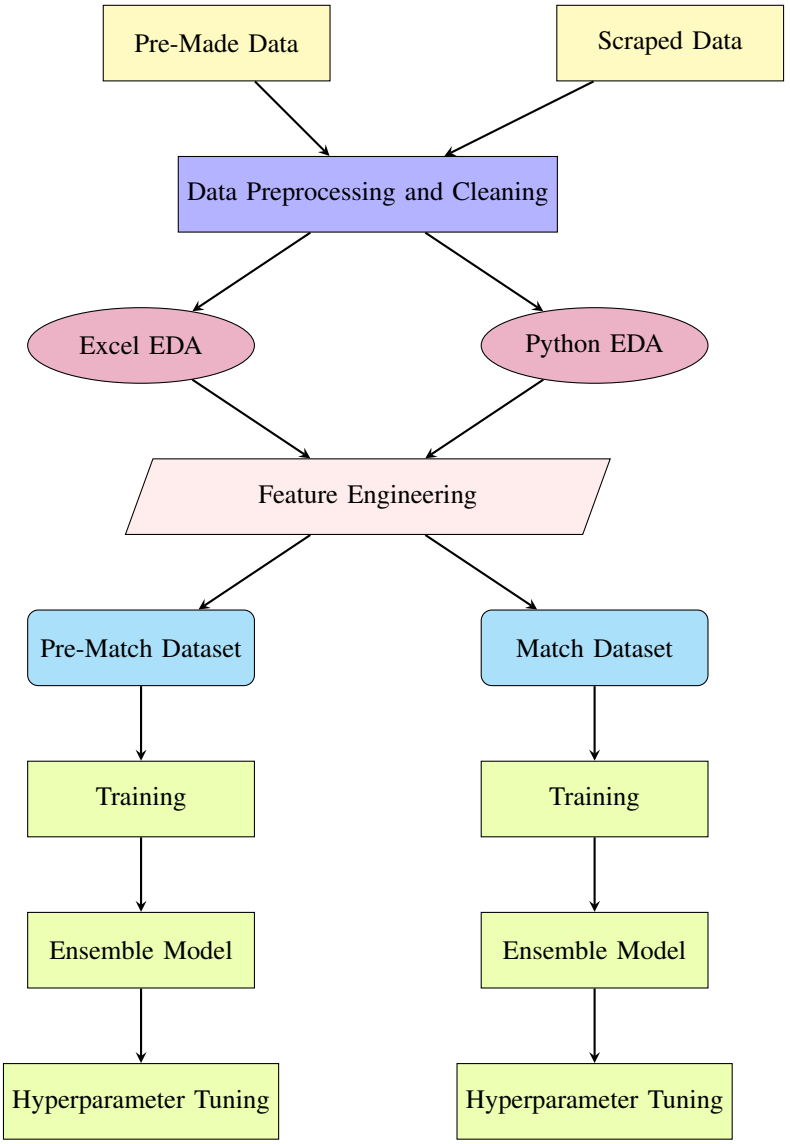

Fig. 2. Flowchart of the research process

II. Data Collection and Preprocessing

For this research, I aimed to collect comprehensive data from the Big 5 Leagues (Premier League, La Liga, Serie A, Bundesliga, Ligue 1) covering the past four seasons. This section outlines the methods and sources used to gather the necessary data, including pre-made datasets, web scraping techniques, and the process of merging and cleaning the collected data.

A. Pre-Made Data

To begin with, I explored available pre-made datasets on platforms like Kaggle and other internet resources. One of the primary sources was football-data.co.uk, which provided a rich dataset containing in-game match data and live betting data. This dataset included information such as match results, team statistics, and betting odds, which are crucial for developing predictive models. Additionally, I sourced Expected Goals

(xG) data from a different website, adding another layer of depth to the analysis by incorporating advanced metrics that reflect the quality of scoring opportunities.

	date	round	day	venue	result	GF	GA	opponent	Poss	Touches
6081	2019-08-25	Matchweek 1	Sun	Away	W	3	2	SPAL	59.0	636
6082	2019-09-01	Matchweek 2	Sun	Home	L	2	3	Torino	58.0	574
6083	2019-09-15	Matchweek 3	Sun	Away	W	2	1	Genoa	48.0	494
6084	2019-09-22	Matchweek 4	Sun	Home	D	2	2	Fiorentina	64.0	721
6085	2019-09-25	Matchweek 5	Wed	Away	W	2	0	Roma	53.0	679
...	...	...	...	...	...	...	...	...	...	...
9118	2023-05-08	Matchweek 34	Mon	Home	W	2	0	Sampdoria	46.0	585
9119	2023-05-14	Matchweek 35	Sun	Away	L	0	2	Fiorentina	55.0	557
9120	2023-05-21	Matchweek 36	Sun	Home	L	0	1	Lazio	40.0	563
9121	2023-05-27	Matchweek 37	Sat	Away	L	2	3	Salernitana	52.0	568
9122	2023-06-04	Matchweek 38	Sun	Home	L	0	1	Juventus	43.0	493

Fig. 3. Pre-Made Dataset Example

B. Scraping Data

Beyond pre-made datasets, I employed web scraping techniques to collect additional match data. Using Beautiful Soup, a Python library for parsing HTML and XML documents, I scraped data from the fbref website. This included detailed match data and possession statistics, which were not fully covered by the pre-made datasets. The scraping process involved identifying relevant HTML elements, extracting the data, and storing it in a structured format for further analysis. The use of Beautiful Soup allowed me to automate the data collection process, ensuring that I could gather up-to-date and comprehensive information efficiently.

```
# Step 2: Set up WebDriver for Selenium
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.chrome.options import Options
from webdriver_manager.chrome import ChromeDriverManager
!chmod +x /content/chromedriver

def create_webdriver():
    chrome_options = Options()
    chrome_options.add_argument("--headless")
    chrome_options.add_argument("--no-sandbox")
    chrome_options.add_argument("--disable-dev-shm-usage")
    chrome_driver_path = "/content/chromedriver"  # Path where you uploaded chromedriver
    driver = webdriver.Chrome(service=Service(chrome_driver_path), options=chrome_options)
    return driver

# Create a WebDriver instance
driver = create_webdriver()

# Step 3: Define the scraper functions from scrape.py
import requests
from bs4 import BeautifulSoup
import pandas as pd
import time
import os

def get_season_urls(base_url, league_path):
    season_urls = []
    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36'
    }
    response = requests.get(f"{base_url}/{league_path}", headers=headers)
    if response.status_code != 200:
        print(f"Failed to retrieve {base_url}/{league_path}")
        return []

    soup = BeautifulSoup(response.content, 'html.parser')

    for link in soup.find_all('a', href=True):
        if 'results' in link['href']:
            season_urls.append(base_url + link['href'])

    print(f"Found {len(season_urls)} seasons")
    return season_urls
```

Fig. 4. Selenium Web Scraper on Chrome

Merging Datasets After collecting data from multiple sources, the next step was to merge and clean these datasets to ensure they were aligned and ready for analysis. This was a meticulous and time-consuming process, as it involved handling discrepancies in data formats, addressing missing values, and ensuring consistency across different datasets. I used various data preprocessing techniques to normalize the data, resolve conflicts, and create a unified dataset that combined match results, team statistics, betting data, and possession statistics. The resulting dataset provided a robust foundation for building and evaluating machine learning models to predict soccer match outcomes.

```
# Function to resolve specific conflicts
def resolve_conflict(row):
    if row['league_id'] == 4:  # La Liga
        if row['home_team_id'] == 231:  # Valladolid
            row['home_team_code'] = 'VLL'
        elif row['home_team_id'] == 146:  # Valencia
            row['home_team_code'] = 'VAL'
        if row['away_team_id'] == 231:  # Valladolid
            row['away_team_code'] = 'VLL'
        elif row['away_team_id'] == 146:  # Valencia
            row['away_team_code'] = 'VAL'
    return row

# Apply the conflict resolution
normal_data = normal_data.apply(resolve_conflict, axis=1)


# Function to append the league number to the team code
def append_league_id(row):
    league_id = row['league_id']
    row['home_team_code'] = f"{row['home_team_code']}{league_id}"
    row['away_team_code'] = f"{row['away_team_code']}{league_id}"
    return row
```

Fig. 5. Valencia and Valladolid having the same team code caused headaches

By combining pre-made datasets with scraped data and thoroughly cleaning and merging the information, I ensured that the dataset used in this study was both comprehensive and reliable. This multi-faceted approach to data collection was crucial for capturing the complexity of soccer matches and enhancing the accuracy of the predictive models developed in subsequent sections of the paper.

III. Excel EDA

In the initial phase of the exploratory data analysis (EDA), I utilized Microsoft Excel to create a comprehensive and user-friendly database for the dataset. By incorporating buttons and applying visual formatting, I ensured that the dataset was easily navigable, allowing for efficient exploration and analysis. The visual enhancements included color coding, conditional formatting, and the use of slicers to filter data dynamically.

The Excel database is shown below.

To perform the general EDA, I extensively used pivot tables and pivot charts. These tools allowed me to summarize and visualize the data effectively, providing insights into various aspects of the dataset. One of the primary focuses of this analysis was to explore non-match-specific information, which

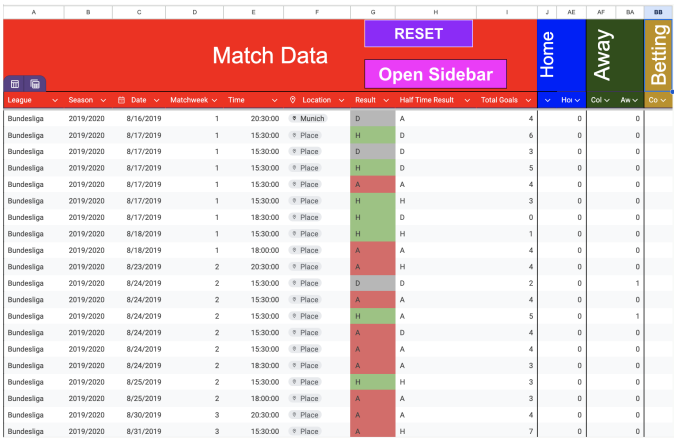

Fig. 6. Organized Excel Database

included examining the overall trends and patterns across different leagues and seasons.

One key aspect of the EDA was analyzing the distribution of match outcomes (win, draw, loss) for each league in each season. By setting up pivot tables, I could aggregate the match results and generate pivot charts that displayed the proportions of wins, draws, and losses. This analysis highlighted the competitive nature of different leagues and showed how these distributions varied from season to season.

Result Analysis For Leagues and Season

Result Table		Result			
League	Season	A	D	H	Grand Total
Bundesliga	2019/2020	115	68	123	306
	2020/2021	96	81	129	306
	2021/2022	90	73	143	306
	2022/2023	86	75	145	306
Bundesliga Total		387	297	540	1224
La-Liga	2019/2020	101	105	174	380
	2020/2021	113	109	158	380
	2021/2022	104	111	165	380
	2022/2023	109	89	182	380
La-Liga Total		427	414	679	1520
Ligue-1	2019/2020	75	70	134	279
	2020/2021	143	95	142	380
	2021/2022	116	102	162	380
	2022/2023	125	92	163	380
Ligue-1 Total		459	359	601	1419
Premier-Leag	2019/2020	116	92	172	380
	2020/2021	153	83	144	380
	2021/2022	129	88	163	380
	2022/2023	109	87	184	380
Premier-League Total		507	350	663	1520
Serie-A	2019/2020	137	85	158	380
	2020/2021	128	97	155	380
	2021/2022	134	98	148	380
	2022/2023	119	100	161	380
Serie-A Total		518	380	622	1520
Grand Total		**2298**	**1800**	**3105**	**7203**

Fig. 7. Pivot Table showcasing Results across Season/League

Additionally, I investigated the amount of goals scored across all leagues and matchweeks. Using pivot tables, I aggregated the total number of goals and created pivot charts to visualize the distribution of goals scored. This analysis provided insights into the scoring trends and helped identify periods with higher or lower goal frequencies. It also allowed for comparisons between different leagues, revealing differences in offensive performance.

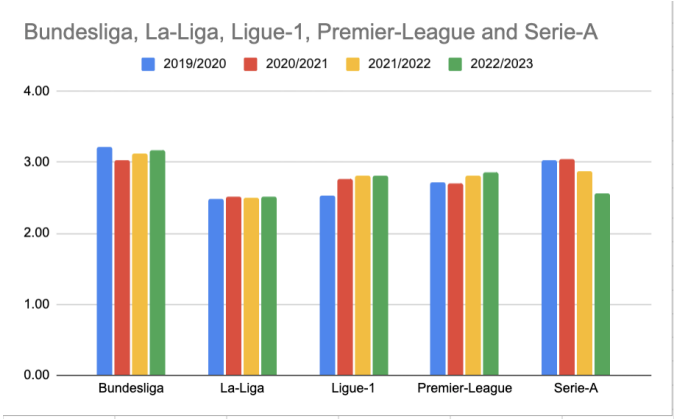

Fig. 8. Pivot Chart showcasing goals across League/Season

Overall, the use of Excel for EDA enabled a thorough examination of the dataset, focusing on general trends and patterns that are not specific to individual matches. The combination of a well-organized database, interactive navigation, and powerful pivot tables and charts provided a solid foundation for understanding the dataset and guided subsequent, more detailed analyses.

IV. Python EDA

The Python Exploratory Data Analysis (EDA) phase involved a more detailed and technical examination of the dataset using various Python libraries. This analysis was divided into two subsections: Relationships and EDA of results.

A. Relationships

In the Relationships subsection, I explored various relationships within the data that intrigued me, though they were not directly related to the primary goal of predicting match outcomes. These analyses included examining Expected Goals (xG) versus actual goals, Expected Points (xP) versus actual points, possession versus PPDA (Passes Per Defensive Action), and possession versus shots.

To investigate these relationships, I created various charts using libraries such as Matplotlib and Seaborn. For each relationship, I plotted scatter plots to visualize the data distribution and calculated correlation coefficients to quantify the strength of the relationships. Additionally, I performed linear regression analysis to understand the linear relationships between these variables.

For example, in the xG versus goals analysis, I plotted xG against the actual goals scored in matches and found a strong positive correlation, indicating that xG is a good predictor of goals. Similarly, in the possession versus shots analysis, I observed that teams with higher possession tended to have more shots on target, demonstrating a clear link between ball possession and offensive opportunities.

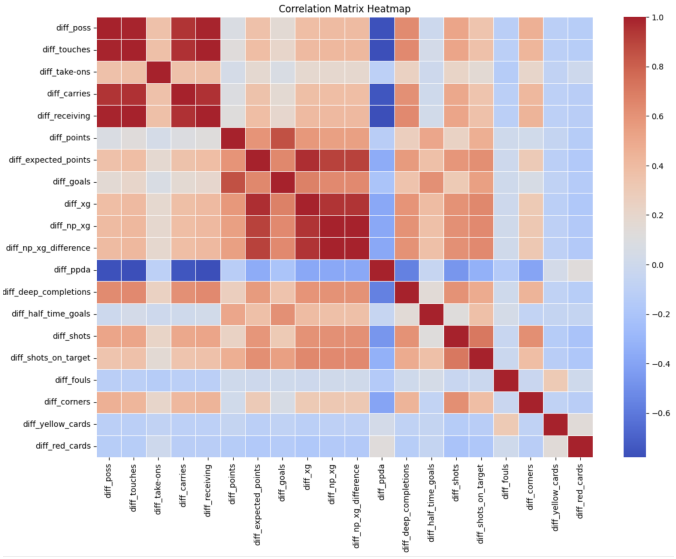

Fig. 9. Heatmap of Correlations between key variables

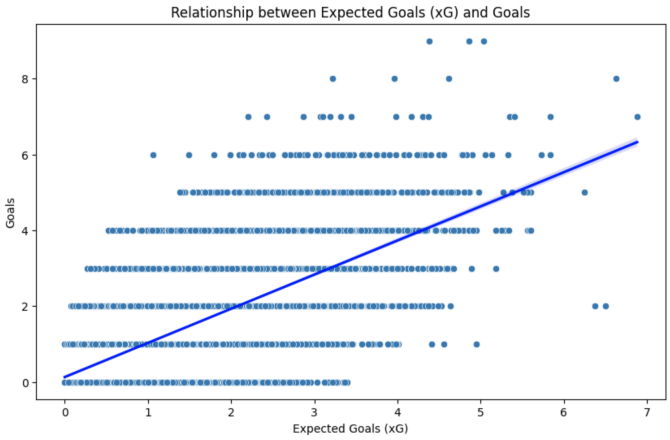

Fig. 10. XG vs. Goals

For the relationship between XG and Goals, I calculated the Least Squares Regression line as:

$$y = 0.9003x + 0.1307$$

I calculated the Pearson Regression Coefficient $r = 0.6363$.

These analyses provided valuable information on the underlying dynamics of soccer matches, although they were not directly related to the project's predictive modeling goal.

B. Result EDA

The Result EDA subsection focused on examining how various variables influenced the match outcomes. This analysis was crucial for identifying the most impactful variables that would be used in the predictive models.

One of the key findings from this analysis was the observation of more Home wins than Away wins. By visualizing the distribution of match outcomes, I noted a significant home advantage across different leagues and seasons. This insight was crucial for model development, as it highlighted the importance of home-field advantage in predicting match results.

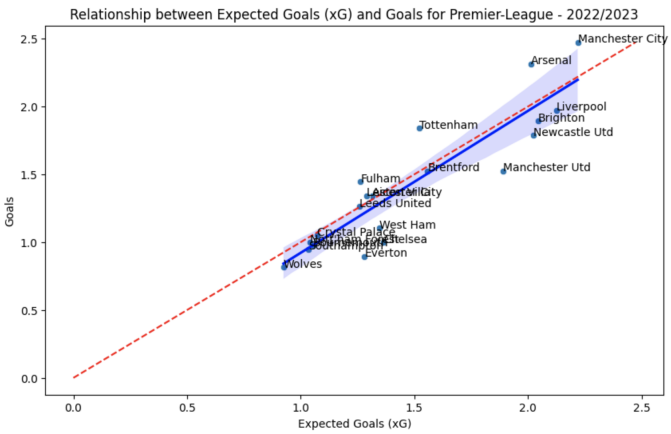

Fig. 11. XG vs. Goals for 2023 Premier League Season

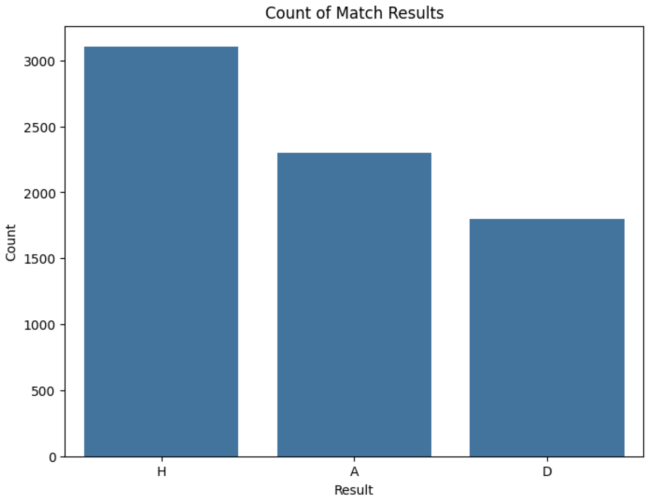

Fig. 12. Distribution of Results

Another interesting discovery was the unexpected relationship between match result and possession. While it was initially assumed that higher possession would highly correlate with a higher likelihood of winning, the analysis revealed that this was not always the case. In some instances, teams with lower possession percentages managed to secure victories, suggesting that other factors, such as counter-attacking efficiency or defensive solidity, played a significant role. In the end, the relationship is not as strong as modern doctrine seems to suggest.

Additionally, I explored other variables such as shots on target, PPDA, and xG in relation to match outcomes. By creating scatter plots, heatmaps, and conducting regression analyses, I identified which variables had the strongest associations with winning, drawing, or losing matches. This comprehensive examination of the data provided a deeper understanding of the

key factors influencing match results and informed the feature engineering process for the predictive models.

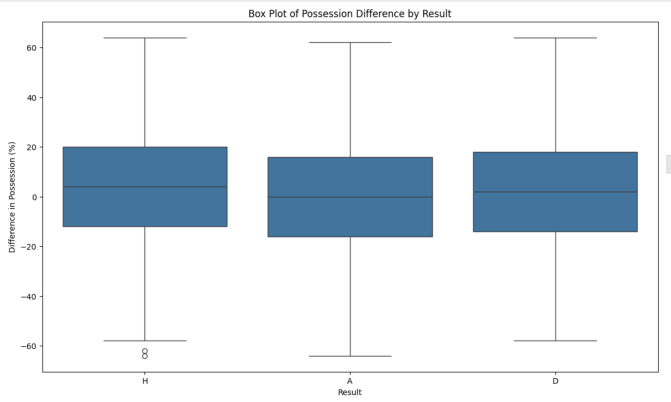

Fig. 13. Results vs. Possession Differential

Through this detailed Python EDA, I gained valuable insights into the dataset, identified critical relationships, and laid the groundwork for developing accurate and robust predictive models.

V. Method

A. Feature Engineering

1) Match Data: For the match data, feature engineering involved creating variables that captured the differences between teams in various match-specific metrics. The columns include:

- **diff_poss**: The difference in possession percentage between the two teams.
- **diff_take_ons**: The difference in successful take-ons (dribbles) between the teams.
- **diff_ppda**: The difference in Passes Per Defensive Action, a measure of defensive pressure.
- **diff_deep_completions**: The difference in deep completions, passes completed within a certain distance to the opponent's goal.
- **diff_shots**: The difference in total shots taken by the teams.
- **diff_shots_on_target**: The difference in shots on target.
- **diff_fouls**: The difference in the number of fouls committed.
- **diff_corners**: The difference in the number of corner kicks awarded.
- **diff_yellow_cards**: The difference in yellow cards received.
- **diff_red_cards**: The difference in red cards received.
- **result**: The match outcome, with possible values indicating Home win (H), Draw (D), or Away win (A).
- **result_encoded**: The encoded result used for machine learning, typically with values 2 (Home win), 1 (Draw), and 0 (Away win).

These engineered features aimed to capture the relative strengths and weaknesses of the teams during the match, providing the model with rich, in-game data.

2) Pre-Match Data: For the pre-match data, feature engineering focused on variables that are available before the match begins. The columns include:

- **result**: The match outcome, with possible values indicating Home win (H), Draw (D), or Away win (A).
- **diff_avg_closing_home_win**: The difference in average closing odds for a home win between the teams.
- **diff_avg_closing_draw**: The difference in average closing odds for a draw.
- **diff_avg_closing_away_win**: The difference in average closing odds for an away win.
- **diff_avg_closing_asian_handicap_home**: The difference in average closing odds for the Asian handicap favoring the home team.
- **diff_avg_closing_home_away_win**: The difference in average closing odds for a home or away win.
- **normalized_avg_draw**: The normalized average odds for a draw.
- **rolling_5_diff_deep_completions_diff**: The rolling 5-match difference in deep completions.
- **rolling_5_diff_expected_points_diff**: The rolling 5-match difference in expected points.
- **rolling_5_diff_points_diff**: The rolling 5-match difference in actual points.
- **rolling_5_diff_ppda_diff**: The rolling 5-match difference in Passes Per Defensive Action.
- **rolling_5_diff_shots_diff**: The rolling 5-match difference in total shots.
- **rolling_5_diff_shots_on_target_diff**: The rolling 5-match difference in shots on target.
- **rolling_5_diff_take_ons_diff**: The rolling 5-match difference in successful take-ons.
- **diff_form**: The difference in recent form, typically based on points gained in recent matches.
- **diff_h2h_exp_points_diff**: The difference in expected points from head-to-head matches.

These pre-match features aimed to capture the form, strength, and expectations of the teams before the match starts, providing valuable context for the predictive model.

B. Model Preprocessing

The problem at hand is a 3-way classification problem where the goal is to predict the outcome of a soccer match: Home win, Draw, or Away win. Handling this type of classification involves several preprocessing steps to prepare the data for model training.

First, we addressed the issue of highly correlated columns. Highly correlated features can lead to multicollinearity, which can negatively impact the model's performance by giving redundant information. To mitigate this, we calculated the correlation matrix and dropped one of each pair of features with a correlation coefficient higher than a specified threshold.

Next, we dealt with missing values in the dataset. Rows with NA values were dropped to ensure that the model was trained on complete data without any gaps, which could otherwise lead to inaccuracies.

Scaling the data was an essential step in preprocessing. We used the StandardScaler from the scikit-learn library to standardize the dataset. This process involved transforming the data so that each feature had a mean of zero and a standard deviation of one. Standardizing the data is crucial for models that are sensitive to the scale of input features.

The result variable, which indicates the match outcome (Home win, Draw, or Away win), was encoded into numerical values. This encoding converted the categorical outcome into a format suitable for classification algorithms, with Home win encoded as 2, Draw as 1, and Away win as 0.

Finally, the dataset was split into training and testing sets using an 80-20 split. The training set comprised 80% of the data and was used to train the model, while the testing set comprised the remaining 20% and was used to evaluate the model's performance. This split ensures that the model's performance can be assessed on unseen data, providing a realistic measure of its predictive capabilities.

- **Dropping Highly Correlated Columns**: Calculated correlation matrix and removed features with high correlation to avoid multicollinearity.
- **Drop NA**: Removed rows with missing values to ensure completeness of the dataset.
- **Scaling the Data**: Applied StandardScaler to standardize the features to have a mean of zero and a standard deviation of one.
- **Encoding the Result Variable**: Converted categorical match outcomes into numerical values (Home win: 2, Draw: 1, Away win: 0).
- **Train-Test Split**: Split the data into 80% training and 20% testing sets to evaluate the model's performance on unseen data.

C. Models

In this study, we utilized five different machine learning models: Logistic Regression, Random Forest Classifier, XGBoost, Support Vector Machine (SVM), and Neural Network. Each model was chosen for its unique strengths and the complementary insights it could provide.

1) Logistic Regression: Logistic Regression is a simple yet powerful linear model used for binary and multiclass classification problems. It predicts the probability that a given input belongs to a certain class. This model was chosen for its interpretability and ease of implementation.

The logic behind Logistic Regression involves finding a linear relationship between the input features and the log-odds of the target variable. The mathematical equation for Logistic Regression is:

$$P(y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n)}} \tag{1}$$

Where:

- $P(y = 1|X)$ is the probability of the target variable being 1 given the input features X
- β_0 is the intercept term.
- $\beta_1, \beta_2, ..., \beta_n$ are the coefficients for each feature $X_1, X_2, ..., X_n$.

2) Random Forest Classifier: Random Forest is an ensemble learning method that operates by constructing multiple decision trees during training and outputting the mode of the classes (classification) of the individual trees. It was chosen for its robustness and ability to handle a large number of features.

The logic behind Random Forest involves averaging multiple decision trees to reduce overfitting and improve generalization. The mathematical formulation can be expressed as:

$$f(X) = \frac{1}{N} \sum_{i=1}^{N} h_i(X) \tag{2}$$

- $f(X)$ is the final prediction.
- N is the number of trees in the forest.
- $h_i(X)$ is the prediction of the i-th tree.

3) XGBoost: XGBoost (Extreme Gradient Boosting) is an advanced implementation of gradient boosting designed for speed and performance. It was selected for its efficiency and high predictive power, especially in handling large datasets with complex patterns.

XGBoost works by sequentially adding models to correct the errors made by previous models. The objective function is:

$$Obj(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{3}$$

Where:

- Obj(θ) is the objective function to minimize.
- $l(y_i, \hat{y}_i)$ is the loss function.
- $\Omega(f_k)$ is the regularization term to prevent overfitting.

4) Support Vector Machine (SVM): Support Vector Machine (SVM) is a powerful classification technique that finds the hyperplane that best separates the classes. SVM was chosen for its effectiveness in high-dimensional spaces and its robustness against overfitting.

The decision function for SVM is given by:

$$f(x) = sign(\vec{w} \cdot \vec{x} + \vec{b}) \tag{4}$$

Where:

- $\vec{w}$ is the weight vector.
- $\vec{x}$ is the input feature vector.
- $\vec{b}$ is the bias term.
- sign is the sign function that determines the class label.

5) Neural Network: Neural Networks are inspired by the structure and function of the human brain. They consist of layers of interconnected nodes (neurons) and are capable of capturing complex patterns in the data. This model was chosen for its flexibility and ability to model non-linear relationships.

The output of a neural network can be expressed as:

$$y = f_2(W_2 \cdot f_1(W_1 \cdot X + b_1) + b_2) \tag{5}$$

Where:

- y is the output vector.

- W_1 is the weight matrix for the input layer to the hidden layer.
- W_2 is the weight matrix for the hidden layer to the output layer.
- X is the input feature vector.
- b_1 is the bias vector for the hidden layer.
- b_2 is the bias vector for the output layer.
- f_1 is the activation function for the hidden layer, applied element-wise.
- f_2 is the activation function for the output layer, applied element-wise.

VI. Prepare Your Paper Before Styling

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections VI-A–VI-E below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not number text heads—LaTeX will do that for you.

A. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, ac, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

B. Units

- Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as "3.5-inch disk drive".
- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
- Do not mix complete spellings and abbreviations of units: "Wb/m^2" or "webers per square meter", not "webers/m^2". Spell out units when they appear in text: ". . . a few henries", not ". . . a few H".
- Use a zero before decimal points: "0.25", not ".25". Use "cm^3", not "cc".)

C. Equations

Number equations consecutively. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

$$a + b = \gamma \tag{6}$$

Be sure that the symbols in your equation have been defined before or immediately following the equation. Use "(6)", not "Eq. (6)" or "equation (6)", except at the beginning of a sentence: "Equation (6) is . . ."

D. LaTeX-Specific Advice

Please use "soft" (e.g., `\eqref{Eq}`) cross references instead of "hard" references (e.g., `(1)`). That will make it possible to combine sections, add equations, or change the order of figures or citations without having to go through the file line by line.

Please don't use the `{eqnarray}` equation environment. Use `{align}` or `{IEEEeqnarray}` instead. The `{eqnarray}` environment leaves unsightly spaces around relation symbols.

Please note that the `{subequations}` environment in LaTeX will increment the main equation counter even when there are no equation numbers displayed. If you forget that, you might write an article in which the equation numbers skip from (17) to (20), causing the copy editors to wonder if you've discovered a new method of counting.

BibTeX does not work by magic. It doesn't get the bibliographic data from thin air but from .bib files. If you use BibTeX to produce a bibliography you must send the .bib files.

LaTeX can't read your mind. If you assign the same label to a subsubsection and a table, you might find that Table I has been cross referenced as Table IV-B3.

LaTeX does not have precognitive abilities. If you put a `\label` command before the command that updates the counter it's supposed to be using, the label will pick up the last counter to be cross referenced instead. In particular, a `\label` command should not go before the caption of a figure or a table.

Do not use `\nonumber` inside the `{array}` environment. It will not stop equation numbers inside `{array}` (there won't be any anyway) and it might stop a wanted equation number in the surrounding equation.

E. Some Common Mistakes

- The word "data" is plural, not singular.
- The subscript for the permeability of vacuum μ_0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter "o".
- In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
- A graph within a graph is an "inset", not an "insert". The word alternatively is preferred to the word "alternately" (unless you really mean something that alternates).

- Do not use the word “essentially” to mean “approximately” or “effectively”.
- In your paper title, if the words “that uses” can accurately replace the word “using”, capitalize the “u”; if not, keep using lower-cased.
- Be aware of the different meanings of the homophones “affect” and “effect”, “complement” and “compliment”, “discreet” and “discrete”, “principal” and “principle”.
- Do not confuse “imply” and “infer”.
- The prefix “non” is not a word; it should be joined to the word it modifies, usually without a hyphen.
- There is no period after the “et” in the Latin abbreviation “et al.”.
- The abbreviation “i.e.” means “that is”, and the abbreviation “e.g.” means “for example”.

An excellent style manual for science writers is [7].

F. Authors and Affiliations

The class file is designed for, but not limited to, six authors. A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

G. Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is “Heading 5”. Use “figure caption” for your Figure captions, and “table head” for your table title. Run-in heads, such as “Abstract”, will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced.

H. Figures and Tables

a) Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. **??**”, even at the beginning of a sentence.

TABLE I
TABLE TYPE STYLES

Table Head	**Table Column Head**		
	Table column subhead	***Subhead***	***Subhead***
copy	More table copy[a]		

[a]Sample of a Table footnote.

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

REFERENCES

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.
[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.