Referee Bias and Racial Discrimination in the English Premier League: A Machine Learning Analysis

Prepared by:

Erwin Medina
December 2025

Submitted To:

Dr. Rong Jin
Department of Computer Science
College of Engineering and Computer Science
California State University, Fullerton

CALIFORNIA STATE UNIVERSITY
FULLERTON™

Department of Computer Science

CPSC 597 / 598 PROJECT / THESIS DEFINITION

To The Graduate Student:

1. Complete a project proposal, following the department guidelines
2. Have this form signed by your advisor and reviewer / committee
3. Submit it with the proposal attached to the Department of Computer Science.

[X] Project
[ ] Thesis

Please print or type.

Student Name: **Erwin Medina** Student ID: **888233475**

Address: **1812 Anacapa, Irvine, CA, 92602**

Home Phone: **909-503-2834** Work Phone: _____

Email: **erwinmedina@csu.fullerton.edu** Units: **24** Semester: **Spring**

Are you a Classified Graduated Student?

[X] Yes
[ ] No

Is this a group project?

[ ] Yes
[X] No

Proposal Date: 05/2025

Tentative Date for Demonstration/
Presentation/Oral Defense: _____

Completion Date: _____

Tentative Title: Referee Bias and Racial Discrimination in the English Premier League: A Machine Learning Analysis

We recommend that this proposal be approved:

Faculty Advisor: Kenneth Kung _____ 5/31/2025
Printed Name Signature Date

Faculty Reviewer: _____
Printed Name Signature Date

Faculty Reviewer: _____
Printed Name Signature Date

Table of Contents

1. Introduction

Referee bias in the sport of soccer has been a subject of debate for some time. Bias, in this context, can be defined as having favoritism for a specific team, such as by carding one team more than another throughout multiple games over time, allowing additional time atop of the extra time after the 90 minutes, or simply not calling and overlooking deliberate fouls. During the 2019-2020 Premier League season, a Video Assistant Referee (VAR) was introduced. A VAR is a referee sitting offsite, in front of multiple screens, with the ability to replay fouls, offside calls, penalties, and goals. They have direct communication with the on-field referee and can alert the referee in real time. VAR was designed and implemented to help on-field referees make accurate decisions and maintain the integrity of the game.

Despite this technological advancement, there are limitations placed on this referee to avoid VAR from dictating the flow of the game. VAR is only allowed to make decisions on direct red cards such as denying a goal scoring opportunity (DOGSO), altercations, or dangerous fouls. They also have power on whether a goal is valid or not, along with determining if a player was offside. Despite the sport having five referees (1 on-field, 2 linesman, 1 fourth official, 1 VAR), mistakes continue to occur, and fouls are continuously missed [11]. Fans, analysts, and pundits agree that referees are inconsistent across games when calling fouls or handing out cards. This inconsistency breeds doubt and confusion as to whether the referees fully under the rules of the game or are biased and avoiding making calls due to favoritism.

Favoritism is only half the battle. Within the sport, there exists a tremendous amount of racism by the fanatics [12, 13]. Football players are often criticized, ridiculed, and sent racist comments over social media when they miss crucial penalties, after losing a game, or if they gave a lackluster performance. Players will either report these messages to the authorities, and or post them on their social media by adding a small caption of, “this needs to stop” or “disgusting!”. Due to the amount of racism and animosity, the English Premier League (EPL) began a movement called, “No Room for Racism”. Both players and referees wear a badge with the quote on their shoulder, and the quote can be seen throughout the game as well. A few actions that the EPL have taken against racism are, investigating over 2500 cases of online discriminatory abuse targeted at players since 2020, providing mental and emotional support for players, and providing an online reporting system on club websites and the EPL website [1]. Even with all the support, advertisements, and repeated messages, racism, unfortunately, continues to plague the sport. But does this racism transcend into the referees as well? Are there any patterns as to which referees are handing out cards to the players of a certain nationality more than others?

2. Problem Domain

Many soccer match predictor tools exist throughout the internet, with some ranging from amateur models that are not well known and hard to understand, to higher end, refined, and well-known models such as FiveThirtyEight's prediction tool. Oftentimes, these predictor tools are hidden behind paywalls because they're valuable monetary tools that can lead to successful bets for those that gamble. Some tools fail to mention how they draw their conclusions, while others like FiveThirtyEight outline exactly how they draw their predictions and the parameters they use [2]. Many of these tools use the same parameters such as how often a team is scoring, shots on target, their form (winning/losing streaks), whether they're away or at home, using previous seasons' data, and much more. All these parameters are crucial for determining how well a team is doing and whether if they can produce a win in their following game. But a parameter that seems to be ignored or unaccounted for is the referee.

Given how vital a referee is in each match, it is difficult to understand the reasoning on why the referee parameter is ignored. A couple of potential conclusions that can be drawn are that it could be possible that a referee makes little to no impact on the outcome of the game relative to all the other parameters, which most would disagree with or have a hard time believing. Alternatively, they do have an impact, but to reduce harassment, threats, or even physical violence against the referees by fans, predictor tools simply fail to mention that as a parameter. Along the same lines, there are no tools that mention whether a referee has a racial bias, presumably for the same reasons. Determining whether a referee has a bias in the sport is crucial for viewers, analysts, betting prediction sites, and most importantly for ethics and impartiality of the game.

Within Europe, there are many rivalries among cities and teams. For example, in England, Liverpool Football Club (FC) and Manchester United FC have a rivalry dating back several decades if not over a hundred years due to their cities' histories. With some referees originating or having grown up in Manchester and refereeing a game of Liverpool vs Manchester United, many fans began to feel unsettled that this would generate a conflict of interest. Are there patterns with less fouls for the Manchester team with a Manchester referee or is it irrelevant?

2.1. Key Issues

2.1.1. Data Gathering

An important issue with determining whether a referee has a bias is gathering key data points. In Table 1A in the appendix, I've outlined which data points I aim to collect, what type of data point they are, and what my reasoning is behind that data point. This

clarity allows readers to determine if my proposed analysis is on the right track or if it requires refining. Machine learning models are only as good as the data they are given, and training a model can be challenging without the correct data. The majority, if not all, of this information can be collected manually, or web scraped, through websites like the Premier League official website, Google, Wikipedia, LiveScore, and much more.

2.1.2. Safety Concerns

Soccer has a tendency of riling individuals up and unfortunately experiences hooliganism and ultra fans. Hooliganism are individuals who cause disruptive, unlawful behavior such as rioting, bullying, and vandalism. While Ultras are fans who are known for their extreme support for their team but have been linked to extremist movements [3]. Although soccer attracts all types of individuals, it should be noted that not all individuals are violent or extreme. However, as mentioned before, when players miss crucial penalties, lose a game, or perform poorly, fans tend to have a negative reaction and send hateful messages through social media. Referees are not exempt from this. An example of this is Anthony Taylor, an English referee, who partook as a referee in the Europa League final, a notable European cup. After questionable on-field decisions, he was heckled and harassed later that day at the airport with his family, along with chairs being thrown at him [4]. Another notable example occurred recently in Spain when a player, Jude Bellingham from Real Madrid, was sent off with a red card for foul language that the referee thought was directed at him but had misunderstood. Jose Munuera, the referee for the game received an extreme level of harassment, death threats, and had his family harassed as well, leading to him shutting down his social media [5]. Thus, we can see that violence and harassment are not solely targeted at players.

Although using a referee as a parameter is important to understand whether there is a bias, racial or not, it is crucial to note that analyzing referee bias is not intended to cause violence, hatred, or anger towards specific referees. Instead, it is intended to be used as a tool to help understand if such a bias exists, and if so, are there ways to prevent, reduce, or eliminate this bias through additional training.

2.1.3. Subjective Decision-Making

Soccer is a contact sport where fouls and injuries can happen within seconds and decisions need to be made on whether that foul was a warning, yellow, or a red. Split second decision making will always be prone to error, and it can be challenging to determine whether the call was biased or not. The referee also experiences social or external pressures from the home crowd leading to home team bias. External factors such as crowd noise, proximity of the crowd, and instant decision making are all challenges

that are incorporated into whether a decision was correct or not. What also makes this topic difficult is that not all referees apply the rules equally. One might consider something being a handball while another might see the same play and determine it is not a handball. However, this paper and project are not determined to find comparisons between referees but instead patterns between referees and teams.

2.1.4. Legal Barriers and Public Perception

Referee bias studies also run the risk of legal pushback if studies single out specific referees or leagues. Referees could sue for defamation if they believe that their reputation is damaged due to a study that suggests they card more Nigerian players than English players, for example. And despite the research being sound and valid, the legal costs could be detrimental to publishing findings. Fédération Internationale de Football Association (FIFA) has historically been reluctant to acknowledge bias in referees or officiating. Although they have been accused of tolerating racial discrimination among referees in the past, they rarely release any disciplinary records. Due to most investigations occurring internally, leagues have little to no incentive to acknowledge that there is any bias in their league. Thus, damaging not only the referee's credibility but also the league's could lead to further legal pushback or defamation lawsuits.

Public perception of referee bias is often dismissed as "cherry picked" data or being "fan-driven narratives". Fans are highly invested in the sport and their team and will accuse the referee of bias anytime a controversial call goes against them. Thus, it is difficult to separate research findings from emotional reactions, which can undermine credibility of the project. Another issue is confirmation bias. If the research finds that there is a significant bias towards Latino players, for example, those who already believe this will support the project, but those that do not or are skeptics would dismiss the findings. And of course, if it shows that there is no evidence of bias, then some may argue that the study or research was flawed and did not use the proper measurements. Public perception of bias, along with legal barriers are plausible reasons for the lack of further research into the topic.

3. Literature Survey

3.1. Survey Paper 1

Referee Bias by T. Dohmen and J. Sauermann [6]

Dohmen and Sauermann go into depth about their findings of referee bias. First, they outline where in the sport they found this bias, and later dive deeper about potential causes on why referee's might have acted that way. They take note on referees having a bias in allowance for time lost after the 45 minutes, also known as stoppage time. This allowance of time is determined by time lost during substitutions, injuries, time wasting, or other causes. And although the amount allotted is at the discretion of the referee, Dohmen and Sauermann mention that more time is often provided to the home team when they are behind. Papers they analyzed mention that stoppage time was approximately 113 seconds longer when the home team was behind one goal, compared to when the home team was ahead by one goal. This evidence suggesting that a referee weighs the social reward from the crowd, thus allowing for more or less time.

Other crucial points that they factor in are biases in other decisions such as goals, penalties, and yellow/red cards. As these decisions impact the game more than providing extra time, referees are known to be more cautious to avoid making the wrong call. Within goals, they determined that there was no evidence that referees award more towards the home team than the visiting teams. In their analysis of penalty kicks, they found that referees were biased in awarding penalty kicks to home teams. With yellow and red cards, they found research that supported that referees tend to favor the home team by punishing their players less and those of the visiting team more strongly. Dohmen and Sauermann mention that the crowd and spectators affect the referee's decisions through social pressure but struggled to find conclusive evidence on how.

A notable item that I found in this paper was that the data analyzed came from the 1990's and early 2000's. Although the sport has not changed, technological advancements in the sport have, such as goal-line technology (checking whether the ball crossed the goal line), VAR, and semi-automated offside cameras. These advancements aid the referees in making more precise and accurate decisions. The laws of the game have also changed such as adjustments to the handball rule, goalkeeper rules for penalties, changes to the number of substitutions, and more. Along the laws of the game, there have also been a new era of referees who might have a different view or opinion on how to manage the game versus 20 years ago. Thus, a revisit of the data and conclusions drawn are important in ensuring the integrity of the game.

3.2. Survey Paper 2

Do Soccer Referees Display Home Team Favoritism? by B. Lucey and D. Power [7]

Lucey and Power delve into determining whether referees provide a home team advantage over the visiting team. They touch on topics and previous research that Dohmen and Sauermann referenced in their paper, such as social pressures from the

crowd, additional time after the 90 minutes, but also how the biases increased by 40% from the beginning to the end of the season. The influence of the crowd tends to have a heavy sway in determining the referee's decision making. A study was done on a sample of referees, half were given audio and video of tackles, while the other half were only given video. What was noticed was that referees with audio were more reluctant to give fouls and were uncertain in their decisions because of the crowd noise. In contrast, the referees with just video were able to classify fouls and make more confident decisions. Although we can see there are external factors that play into the decision making of the referee, it is important to note that my focus is to determine whether there are patterns against certain teams, not simply a home team. In their conclusion, they state that Italian and American referees are not influenced by social environments, but that referees do tend to favor home teams but only in close games. They also outline that their study was only partially done due to examining only one potential area where biases exist, added time.

Although their study is tailored and focused on more of a home team advantage, it is clear and obvious that referees experience some type of bias either through internal or external pressures.

3.3. Survey Paper 3

Favoritism and Referee Bias in European Soccer: Evidence from the Spanish League and the UEFA Champions League by B. Buraimo, R. Simmons, and M. Maciaszczyk [8]

In this paper, they analyze and find evidence of referee bias in favor of home teams within the Union European Football Association (UEFA) and Spain's first division league. Although many papers cover similar topics of home teams getting an advantage based on the proximity of the crowd, or if the home team is trailing by a single goal, this paper is slightly different from the rest. Within their testing, they utilize minute-by-minute analysis to control for within-game events. They also discovered that within the UEFA Champions League, a competition that includes some of the greatest club teams in Europe, the organizers appoint neutral or non-biased referees from countries outside of the represented teams in any of the matches.

They conclude their paper by saying that they hypothesized the Champions League competition would have less bias due to the neutral referees versus Spain's first division league. However, what they discovered is that whether the stadium has a running field track dividing the soccer pitch and the fans plays a significant part in the bias. They mention that seeing a decrease in away team yellow cards and more cards for home teams when there is a track present implies that the referees do carry a certain bias despite having neutral referees. Unfortunately, this fails to touch on the topic of whether

a referee has favoritism or biases for certain teams. Despite this, the paper does include a critical statistical analysis of referee bias in the sport and reinforces the idea of crowd proximity influencing referees.

3.4. Survey Paper 4

An Integrated Conceptual Framework of Decision-Making in Soccer Refereeing by R. Samuel, G. Tenenbaum & Y. Galily [9]

Within this paper, they focus on the decision making of the referee such as any influences, external factors, distance from the foul, positioning of the referee, fatigue, stress, and more. This paper deviates from the over saturated fan proximity research and instead analyzes the complexity of decision making. During the game, referees make numerous repetitive decisions, where many are dependent on the field location of the referee and effective communication of the assistant referees. Although it is evident that the proximity of the referee for a foul impacts the decision making, it is important to note that there exists an optimal distance. Samuel, Tenenbaum, and Galily determined that decision to be approximately 11-15 meters (35-50ft). Any closer and the referee can potentially miss important details. Likewise, any further produces the same result.

Another analysis they focused on was the height of the referee. Their findings concluded that shorter referees tended to produce more yellow cards. But for red cards, it depended on the division. Lower division and short referees had more red cards, but higher divisions and taller referees had the same result. A factor not yet touched on is the decision making of the referee on where to run and position him or herself. A poorly placed referee increases their chances of making a decision error as they are not able to see the important details. But this factor was noted to be attributed to fitness level and physiological fatigue. If positioned correctly, they then must anticipate, through experience and prior knowledge, where to move to next based on how the ball and players are moving. And should the call they make be unpopular, the referee must remain calm and in control to continue with their effective decision making.

They finalize the paper by concluding that there are many critical factors that determine a referee's decision making, and not every factor can be analyzed in one paper. However, they outline a few steps to shift the focus from referee biases to improving successful referee performances. Unfortunately, there is not a mention of whether a referee has a particular team or racial bias. Nonetheless, it sheds light on external factors that were not mentioned before in other research, such as referee height, proximity (optimal distance) to the foul, and mental fatigue. All of which are important factors that can be detrimental on the reasoning behind why a card or foul is given.

3.5. Survey Paper 5

Referee Bias Contributes to Home Advantage in English Premiership Football by R. Boyko, A. Boyko, & M. Boyko [10]

Boyko et al. focus their paper on not only the home advantage bias but whether individual referees vary in their home bias or whether biased decisions contribute to overall home advantage. They examine over 5200 English Premier League matches with over 50 referees and found that home bias differs between referees. They outline that there are different measures of home advantage, such as goal differentials between home and away teams, yellow cards, and penalty differentials. Boyko et al. determined the variability between referees implies that referees are responsible for the observed home advantage but also suggest that the home advantage is dependent on subjective decisions. They then suggest that further research needs to be done on crowd noise, and the referees psychological and behavioral responses to biased crowds.

The unique approach of expanding on the home advantage bias is key for my research topic. Boyko et al. not only approached the problem as an existing issue but expanded on the research already done and produced results of variability amongst referees. Yet, the question lingers on whether these referees have a particular bias for certain teams, or if any racial bias was determined by analyzing the players that were carded. Although their research serves as a good indicator for a bias existing, and varying among referees, can we delve deeper into finding results for team biases and racial discrimination in the sport.

4. Project Objectives and Significance

This project aims to determine whether referee bias exists toward specific teams or players in the English Premier League (EPL), with a particular emphasis on potential racial bias. To achieve this, the first objective is to collect and clean approximately 10 seasons worth of match data, which includes referee information, player nationalities, fouls, yellow/red cards, match outcomes, added time, and much more. The data points to be collected can be found in Table 1A in the appendix section of this paper. The sources of where the data will come from vary from the Premier League's official website, Wikipedia, and LiveScore.

The second objective is to design and train a machine learning model that can analyze this data and identify statistically significant patterns in referee decision-making. This includes detecting whether specific referees consistently penalize certain nationalities or favor certain clubs. Once validated, the third objective is to develop a simple and interactive user interface where users can explore historical referee behavior,

simulate hypothetical matchups, and view predicted outcomes such as likely winners or number of cards issued based on the model's learned patterns.

Identifying and evaluating referee bias is essential to ensuring fairness in the sport. When rules are consistently upheld, it provides confidence to fans, analysts, and betting platforms that the outcomes are legitimate and unbiased. Transparent officiating builds trust in the integrity of the game. Most importantly, this project also addresses the ethical concern of potential racial profiling, reinforcing the need for equality and accountability. Beyond its impact on sports, the project highlights the power of artificial intelligence and machine learning to uncover insights that traditional analysis might overlook or have not captured yet.

5. Project Activities

The project will be broken up into six distinct phases, each designed to build on the previous step to achieve the final objective: identifying patterns of referee bias through ML and presenting the findings in an interactive user-facing tool. The phases cover the full pipeline from data gathering to model training and final deployment. Each phase is described in detail below.

5.1. Phase 1 – Data Collection

Within this phase, data will be collected and compiled using Python's BeautifulSoup web scraping tool or manually depending on the website's design and readability. The data that will be collected can be found in Table A1 in the appendix and ranges from team names, referees, yellow/red cards, player names, and more. Manually acquired data will be saved into a CSV Excel document, while data acquired by BeautifulSoup will utilize Pandas to convert tables of information into a CSV. This approach is done to consolidate and track everything in one place, avoid scattered files, and reduce inconsistent formatting. With all the data centralized in one location, it is easier to jump into Phase 2.

5.2. Phase 2 – Data Cleaning and Processing

This phase is designed to clean up any inconsistencies, anomalies, absent data, but also normalize information such as player names, club names, and encode categorical features such as nationalities. A few examples of normalizing data can be club names such as "Man United" vs "Manchester United", converting timestamps of yellow cards or extra time from "89.5" to 89th minute + 30 seconds, or converting player nationalities into regional codes for encoding categorical values. The overarching goal of this phase is to prepare the dataset for the machine learning model and visualization that will occur in the following phases. By ensuring that all the data is consistent and normalized, I can move toward the next phase.

5.3. Phase 3 – Exploratory Data Analysis (EDA)

Although I will be building our machine learning model with the data I collected along with creating a full stack user interface, it's always best to pinpoint areas or trends of significant importance prior to the machine learning phase. In this phase, I can create visualizations and possibly build or adjust a hypothesis to ensure that time is not being wasted and also highlighting standout points in my data. For example, we can analyze the data by creating bar charts of cards per nationality / team / referee, card distribution by game state (whether a team is losing vs winning), or box plots of added time vs match outcomes. These examples provide further insight on what to look out for in the further stages but also identifies any bias candidates. This helps us guide the model design, the feature engineering, and later how we interpret the results. By the end of this stage, I should have some evidence to suggest that there are specific data points worth looking at and others that show little to no trends. It would also allow me to refine or strengthen my hypotheses, such as "Referee X gives more red cards to African players" or "Referee Y adds more extra time when Home Team A is losing". After we refine our hypothesis, we're ready for Phase 4.

5.4. Phase 4 – Machine Learning Modeling

In this stage, we will apply supervised machine learning techniques to draw meaningful conclusions, such as using regression or classification models. Our objective here is to build and train predictive models that can detect patterns of favoritism, bias, or possibly racial disparity in referee decisions by using historical match data. We want to identify statistically significant features associated with referee's in-game decisions regarding yellow/red cards and added time. The model will ideally help quantify if our hypothesis of referee behaviors matches or aligns with consistent trends, and whether those trends are linked to team affiliation or player nationality. The model that most aligns with our objective would be a classification approach, such as if a player will receive a card in a particular match, or if the likelihood of the card being yellow or red.

Another important aspect of the machine learning model phase is the feature engineering. Building meaningful input features from our data is critical to the output of our model. Examples of input features could be player nationalities, team names, match score at card time, whether a team is home or away, referee name / city / nationality, and much more. Table A1 in the appendix has the input features that will be utilized for the machine learning model.

After we've defined our problem, outlined our objective, built our input features, then we're ready for selecting the model that we want to utilize for our approach. Sticking strictly with one model might not be a wise choice as different models can produce slightly different results. Examples of different models can be a logistic regression, a random

forest, or naive-bayes model. Thus, trying different models can help us determine which one is more accurate or effective for the data that was provided.

Once we've tried a few different models, we can aim to train and validate / evaluate the results. Ideally, we want to provide the model with 70-75% of the data for training, while using 25-30% for testing purposes. Within this portion of the stage, we should be determining the accuracy, precision, recall, and confusion matrix for our results to ensure the model is behaving as expected. Our goal after this is to understand which features are influencing the model's predictions. In doing so, we can determine whether a referee is biased (or not) or if some teams are being penalized more than others.

5.5. Phase 5 – Results and Analysis

The metrics outputted from phase 4 allow us to draw conclusions and meaningful insights. Here we can highlight important trends such as certain referees being statistical outliers, disproportionate yellow/red cards towards certain nationalities, favoritism during close games, and more. Although this phase blends in with the previous, it does allow us to have a period of reflection or retrospection on the outcome of the project. An example of this can be whether the project experienced limitations such as too small of a sample size, or if there was a model bias, and whether we should be revising the data and redoing the training. Tackling these questions with ample time can potentially produce a stronger model in the future, or a quick rework depending on time constraints.

5.6. Phase 6 – Interactive Tool

This section depends on whether Phase 5 detects a pattern of referee bias, racial or otherwise. With the assumption that it does find trends, it would be best to illustrate these trends to a user. By combining machine learning, data science, and full stack engineering, we can create a tool that can be used by viewers, analysts, and anyone interested simply by using a standard laptop environment. The interactive web-based tool will ideally have a simple design that is easy to understand for those inexperienced with the sport. Using Streamlit or Flask (connected with Python) will allow for a seamless transfer of data from the machine learning model to the user interface. In the user interface, a user will be able to select from a dropdown to select referees and match up two different teams. Once done, it'll display the referee profile along with their card history. After the user is ready to move forward, it will then predict a winner along with the expected card counts. This will finalize the project and allow users to see whether the referee chosen has a bias towards the teams selected.

For example, say the user selects "Arsenal" vs "Liverpool" and then selects a referee of "Howard Webb". If this referee has shown bias towards Arsenal in the past and the data suggests that Arsenal wins their games under this referee often, then the predicted output would suggest that Arsenal will beat Liverpool, and we should expect 7

yellow cards. There are many additional parameters that go into winning games such as player fitness, team cohesion, player statistics, etc, which is not what this tool is aiming to achieve. However, it would allow users to interactively see whether having a specific referee assigned to their game would impact their team and thus make an educated guess on which team would win.

The racial bias portion of the project will most likely be omitted from the interactive tool as it might not as accurate or possibly deliver the wrong message. The interactive tool will not predict which players receive yellow or red cards as that would require a continuously updated player database to track team rosters, thus making the racial bias a bit obsolete for the interactive tool.

5.7. Software Requirements Specification

5.7.1. Functional Requirements

- The system shall scrape and preprocess English Premier League (EPL) match data during Phase 1.
- The system shall export match data as CSV files for use in Phase 2.
- The system shall allow users to:
    - Select referees and teams from dropdown menus (Phase 6).
    - Display historical data related to the selected referee and team.
    - Run a predictive model to estimate the number of yellow/red cards and match outcome.

5.7.2. Non-Functional Requirements

- The system shall return predictive results within 5–10 seconds of user input.
- The interface shall be designed for readability and accessibility for non-technical users.
- The system shall be mobile-friendly and responsive to different screen sizes.
- The system assumes that users have basic familiarity with soccer terminology (e.g., yellow/red cards, match outcome).

5.7.3. Constraints

- The system depends on publicly available or manually gathered match and referee data.
- Proprietary data sources, such as internal referee reports or official league datasets, are not accessible.
- Web scraping may be limited by:
    - The HTML structure of the target websites.
    - Anti-scraping measures or rate limits implemented by website owners.

- The accuracy of the machine learning model is constrained by the quality and completeness of the available data.

5.7.4. Assumptions

- All collected data is assumed to be accurate and reliable.
- Users are expected to understand basic soccer-related concepts.
- The interactive tool is intended solely for demonstration and educational purposes, not for production deployment or real-world decision-making.

6. Environment

The development environment for this project will mainly consist of open-source tools and commonly available hardware and software resources. The objective of this section is to ensure that the project can be run and executed on any standard personal machine without requiring specialized resources or hardware (e.g. high-end graphics processing units (GPU) or central processing units (CPU)). Outlined below will be the overall structure of the environment.

- **Operating System:** macOS, Windows
- **Programming Language:** Python 3.10+
- **Integrated Development Environment (IDE) / Editor**: Visual Studio Code
- **Primary Libraries:**
    - Pandas and Numpy for manipulating and processing data.
    - BeautifulSoup4 for web scraping
    - Matplotlib, plotly for visualization and EDA purposes.
    - Scikit-learn, xgboost for the machine learning portion.
    - Streamlit or Flask for building the interactive front-end interface.
- **Data Format:** CSV files will be used for input and output, allowing for easy editing, version control, and readability.
- **Hardware Requirements**: Project will be developed and tested on a MacBook Air with approximately 8GB of Random Access Memory (RAM) and a standard CPU.
- **Version Control:** GitHub will be primarily used for source code tracking and backup.
- **Hosting:** Streamlit

With the above environment, we can ensure that all components of the project, from web scraping to model training, visualization, and deployment, can be run efficiently on a local machine without specialized hardware or external dependencies.

7. Project Deliverables

This project will produce several deliverables which will demonstrate the full lifecycle from data gathering to deployment. The first deliverable will be a clean, well-structured dataset of EPL match data spanning across 10 seasons. Once completed, the next deliverable will be a trained machine learning model that can identify bias patterns in referee decisions (or lack thereof). An example of this would be the machine learning model outputting ("referee name", "is_bias", ["supported teams"]) → "Howard Webb – True – Arsenal". This is the primary deliverable that I aim to produce through this project and hope to deliver to an advisor and the Computer Science (CS) department. However, to add more value to the project, I aim to create a web-based interactive tool where users can select referees and teams, view historical data, simulate a matchup, and see a predicted outcome. The submission of my final presentation and the interactive tool will signal the completion of this project.

8. Project Schedule

This project is expected to span approximately the entire duration of the fall semester of 2025. Below is a table of the timeline.

Phase	Task	# of Hours	Week(s)
Phase 1 – Data Collection	Manual + automated scraping, data exports	25	Weeks 1-2
Phase 2 – Data Cleaning	Normalizing, encoding, missing value checks	15	Week 3
Phase 3 – EDA	Visualizations, adjusting hypotheses	20	Weeks 4-5
Phase 4 – Modeling	Feature engineering, training, evaluation	30	Weeks 6-9
Phase 5 – Results Analysis	Analyze trends, interpret ML results	20	Weeks 10-11
Phase 6 – UI Development	Build and test web tool with Streamlit/Flask	30	Weeks 12-14
Documentation + Final Presentation	Code comments, github polishing, slide prep, functionality check	15	Week 15

Total		**155**	

This timeline allows for overlap between the phases. For example, insights from the EDA phase may directly help with the feature engineering, and the model results may influence how the UI is structured or presented. However, it is expected that there will be minor delays in the modeling and UI development, depending on their success. Thus, I've provided additional time during those weeks to ensure the project is on track.

9. Project Results

9.1. Data Collection

9.1.1. Dataset 1 – Incident Reports + Match Statistics

Existing data for game statistics is widely available on the internet, and one of the more reliable sources I came across happened to contain almost everything that I initially set out to capture. Kaggle, which is an online platform where data scientists, analysts, and machine learning engineers share datasets and collaborate on projects, provided a Premier League dataset with a date range from 2002 to 2022. This range was far beyond the 10-year window I originally planned to scrape, and the completeness of the dataset made it a strong foundation. It included detailed incident reports for each match, the final scores, a broad set of match statistics, team names, and additional data. One particularly helpful feature was that each match record included a link back to FlashScore.com, allowing me to cross-check the information directly against the official match report. That extra layer of verification was important, especially since my results would depend heavily on accuracy and consistency.

Even with such a broad dataset, I still felt that the newer matches would add value. Soccer evolves, refereeing styles shift, and match reporting itself becomes more detailed over time. For those reasons, I wanted to include the most recent three seasons, covering 2022 to 2025. To make this possible, I wrote a series of scripts for web scraping, each one designed to extract one part of the needed data and then merge it with the existing dataset. Breaking the scraping into separate components helped reduce confusion and lowered the risk of accidentally pulling incorrect or mismatched information – something that becomes a real concern when working with thousands of individual match records.

The first script focused exclusively on retrieving the incident reports for each match. FlashScore's incident timeline contains substitutions, yellow and red cards, penalties, goals, own goals, and even VAR-disallowed goals. The incident data in the original dataset did not include substitutions, but FlashScore did. In my first iteration of

the scraper, it felt safer to collect every event available and clean it later through Python and Pandas rather than attempt to filter it while collecting. Because these events are sometimes subjective, inconsistently labeled, or not easily verified at a glance, gathering everything first provided a better chance at maintaining data integrity. Cleaning and sorting the data later allowed me to be far more deliberate in deciding what should stay, what should be removed, and what needed additional verification.

The second script handled the match statistics. FlashScore reports a wide array of stats – ball possession, total shots, shots on target, passes completed, corner kicks, offsides, yellow cards, fouls, and many others. My goal wasn't to capture every statistic, only the ones that lined up with the structure of the original Kaggle dataset and were relevant to the broader purpose of measuring possible referee bias. I focused on possession, total shots, shots on target, corner kicks, offsides, free kicks, and fouls for each team. These metrics felt like the core indicators of how a match unfolded and would provide enough context to identify irregularities or patterns when comparing referees across multiple seasons.

The final script collected the core match summary information: the home and away teams, the matchday, kickoff time, final scoreline, the assigned referee, and the attendance numbers. This information formed the backbone of each record and acted as the reference point for merging the incidents and match statistics together. Once all three scripts were functioning reliably, I merged their results into a single structure that mirrored the layout and content of the Kaggle dataset while extending it to include the newer matches.

Developing these scripts took significantly longer than writing them. The challenge wasn't simply coding, it was the constant cycle of testing, correcting, and verifying. Web pages change layouts, labels shift, and sometimes a single inconsistent field can lead to a confusing output. I spent weeks refining each script, checking random matches manually, and piecing the results together until they produced a clean, uniform dataset. That long, repetitive process ended up being the bulk of the work, but it was necessary to ensure that the final dataset was reliable enough to use for analysis later. The result was a reconstructed and expanded version of the Kaggle dataset that I trusted as a foundation for testing the presence of referee bias.

9.1.2. Dataset 2 – Foreign Player Names + Nationalities

I originally planned to collect the full list of player names for every team along with each player's nationality. The idea behind this was simple: if I wanted to test whether the model showed any racial or nationality-based patterns in how cards were distributed, I needed to know exactly who the players were and where they came from. But after a brief

look into the referees themselves, specifically their backgrounds and places of origin, it became clear that almost all Premier League referees are British citizens. With that in mind, I realized that I didn't need an exhaustive list of every player's nationality. Instead, for the purpose of identifying bias, it made more sense to focus on a much simpler distinction: whether the players who received cards were foreigners or not.

That shift in scope made the data collection for this part of the project much more manageable. Rather than scraping multiple sources and assembling a full database of players across dozens of clubs and seasons, I relied on a comprehensive resource already available on Wikipedia [14]. There is a page dedicated to listing every foreign player who has played in the Premier League, both retired and currently active. The list includes the player's country of origin, football continental confederation, full name, the teams they played for, and the years they participated in the league. Altogether, this gave me a pool of approximately 3,000 players who qualified as "foreign" under Premier League rules.

No scripting was required for this dataset, but it did involve a surprising amount of manual work. I had to copy, paste, clean, and format all the information into something readable and functional. While tedious, this process made the dataset straightforward to verify, since every player listed had a clear reference point and a defined history in the league. Once completed, the dataset gave me what I needed: a reliable way to identify when a carded player was foreign without overcomplicating the problem.

9.1.3. Dataset 3 – Referee Information

The central objective of this project is to determine whether refereeing bias exists and whether any meaningful patterns appear once the data is assembled and analyzed. Because of that, having reliable referee information was necessary. The Kaggle dataset, though rich in match statistics and incident details, did not include referee names. Without that key piece, I couldn't move forward with any kind of bias analysis, which meant I had to generate the referee dataset myself.

To start, I wrote a script that iterated through each match report link in the Kaggle dataset. These links pointed to FlashScore, and each one contained the referee's name somewhere within the match summary. The script's job was simple: visit the FlashScore match report, extract the referee's name, and return it while keeping the match link as the primary key. Once the script completed, I finally had a basic list of referees tied to specific matches, something that the original dataset was missing.

However, to explore deeper forms of bias, I wanted more than just names. I needed background information about the referees, particularly their place of birth. Knowing where referees come from opens the door to analyzing possible geographical biases, like

whether a referee born in Liverpool might unconsciously officiate differently when overseeing matches involving Manchester clubs. This part of the process, unlike the earlier one, required no scripting at all. Instead, I manually searched each referee on Transfermarkt, a global football database known for its comprehensive player and referee profiles. TransferMarkt was one of the few platforms that consistently listed birthplace information of referees, making it the most reliable source for what I needed.

The manual effort here was significant. After pulling the referee names with the script, I spent time searching each one individually, confirming the details, and then recording their birthplace and any other relevant biographical information. While time-consuming, this manual verification ensured that the referee dataset would be clean, accurate, and aligned with the same standards as the others I created.

By the end of this process, I had a complete referee dataset containing names, match associations, and birthplace information. Combined with the incident data, match statistics, and foreign player identification from the earlier datasets, this final dataset filled the last major gap needed to move into analyzing whether Premier League refereeing shows measurable signs of bias.

9.2. Data Cleaning and Normalization

9.2.1. Dataset 1 – Incident Reports + Match Statistics

Once the initial data collection phase was complete, the next challenge was making sure the information I had was correct. This started with something as simple as verifying yellow-card and red-card counts per match in Excel. I expected this to be a quick task. Instead, it immediately exposed inconsistencies between my manual calculations and the counts listed in the Kaggle dataset. A similar issue appeared with half-time scores; some games were off by a goal or two even though the final score matched. These discrepancies were small in scale but large in consequence, because I ended up with roughly 200 games that needed to be double checked manually. I removed the Kaggle counts and kept my own, since mine had been validated through the official Premier League website and FlashScore.

Even after correcting those fields, something still felt off. When I reviewed the incident reports more closely, I noticed that some entries were written in Polish (likely a leftover from the original dataset creator) and several player names were either missing, incomplete, or spelled differently than the official match report. At that point I realized I couldn’t move forward while relying on patched data. So, I made the decision to redo the entire incident report collection from all seasons from 2002-2022 using the same scraping scripts I had written for the 2022-2025 range. This essentially meant repeating weeks of work, but it ensured that the final dataset rested on information I gathered myself, rather

than a mix of my work and someone else's inconsistencies. After collecting everything again, I compared the results of the original Kaggle dataset and found that most values, especially the match statistics, lined up well, which was reassuring. The difference was that now I trusted the data.

To make the dataset easier to reference, I created my own primary key for each match in the format *year-PL-matchNumber*, where the match number ranged from 1 to 380 for that season. In practice, this identifier didn't play a major analytical role later, but it did make navigation easier. When I needed to jump to a particular match or double check a specific incident, having that clean index helped.

The Kaggle dataset also included dozens of extra columns that had nothing to do with my research goals, betting odds, missing players, "dangerous attacks", "blocked shots", shot breakdowns, and several league tables from competitions outside the Premier League. Since my work focused strictly on the Premier League and referee behavior within it, I removed everything that was unnecessary. Cleaning out the noise helped the dataset settle into something manageable and specific to the bias-related questions I wanted to explore.

9.2.2. Dataset 2 – Foreign Player Names + Nationalities

Since my incident report dataset told me which players received cards and when, the next step seemed simple: match those names to the foreign-player list and determine which players were non-British. At first glance, this sounded like an easy one-to-one join. In reality, it became one of the most tedious and frustrating parts of the entire project.

The core problem was name formatting. My incident reports, because I had scraped them from FlashScore, displayed names in the format "LastName FirstInitial". To keep consistency, I treated the incident report as the "source of truth", meaning the foreign-player dataset needed to conform to that format. I added a new column and wrote formulas to convert full names into this abbreviated structure. That part worked smoothly for players with simple two-word names. The trouble began with everyone else.

Many players do not follow a neat First-Last pattern. Some go by one name ("Hulk", "Pedri"). Others have chains of three, four, or even five words: "Sepp van den Berg", "Jean-Clair Todibo", "Andreas Pereira da Silva", and so on. I added a new column that counted how many words were in each full name, and while that helped with sorting the easy cases, it didn't do much for the messy ones. The incident reports themselves were inconsistent, sometimes spelling out "Van de Berg S." and other times shortening it to "V.D. Berg S." or even reversing the emphasis. Trying to match every one of these was exhausting. It felt that no matter how many formulas I wrote, there was always another

exception, another abbreviation pattern, another inconsistency. I never fully felt that every name in the foreign-player dataset successfully mapped to the incident reports.

The next issue was nicknames. Some players are universally known by a single name, even though their legal full name is much longer. “Lionel Messi” appears formally as “Lionel Messi”, but in match incidents he shows up as “Messi”. Meanwhile, someone like, “Givanildo Vieira de Sousa” appears only as “Hulk”. My automated formulas would turn “Lionel Messi” into “Messi L.”, which was not how he appeared in the reports. So, the join would fail. And this wasn’t a rare situation, this happened dozens of times. I had to decide whether to force the incident report to match the foreign-player dataset, or to adjust the foreign player dataset to match the incident reports. I chose the latter, even though it meant hours of manual corrections.

My process looked something like this: I sorted players by the years they were active in the Premier League. Anyone before the 2002+ window was removed. I added a column that checked whether each formatted name appeared anywhere in my incident-report dataset. If it did, I marked it as “YES”, and if not, it was marked with “NO”. And all the “NO”s became my problem pile. The “YES” list needed no further work. The “NO” list required endless manual checking on TransferMarkt, verifying that the player appeared in the Premier League during the relevant years, cross-checking clubs, and confirming whether they ever received a card, scored, or appeared in match incidents at all.

Some players never appeared simply because they were goalkeepers who never picked up a yellow card. Others played fewer than 30 minutes total in the entire season. Others had name-formatting issues I hadn’t accounted for. And because many players share common surnames, Martinez, Silva, Gomes, Williams, false positives from fuzzy matching sent me down rabbit holes that never led anywhere. A search for “Martinez A.” would return “Martinez B.” as a match, which was clearly wrong. I preferred excluding borderline cases rather than mislabeling them and injecting incorrect data into my analysis.

Even after several passes, I was still left with “NO” players who were legitimate foreign players but never appeared in any incident. After reviewing a large portion of these manually, I noticed a pattern: many were single-season players with extremely low minutes played, enough to be on the roster, but not enough to make it onto the incident report. I made the choice to exclude these players from deeper validation, accepting the small inaccuracy for the sake of moving the project forward. The number was very small compared to the full dataset, and I felt the tradeoff was reasonable.

And just when I thought I was done, one last hurdle appeared: accented characters and non-English alphabets. Many European players use characters like “ø”, “æ”, or “č”, which FlashScore translates into their closest English equivalents. “Møller Dæhli” becomes “Moeller Daehli”. They are technically the same person, but string-matching

does not treat them as such, and thus I wrote another cleanup script to remove accents and normalize letters across datasets. Despite this, I still suspect a tiny number of players slipped through the cracks. Matching thousands of names across two inconsistent systems is delicate work, and perfection is almost impossible, but the dataset was cleaned enough to be analytically reliable.

9.2.3. Dataset 3 – Referee Information

Compared to the first two datasets, the referee information required almost no heavy cleaning at all. With only around fifty referees, and with each one appearing under a consistent name format, there were no major formatting issues or spelling variations to resolve. Their names matched the incident reports cleanly, and the table joined on a simple index-match without any complications. Since nationality and birthplace were manually verified earlier, and because referees almost never change their professional names, the dataset was already in a naturally normalized state. In contrast to the chaos of player-name matching, this part was straightforward.

The only issue that arose was referee names didn't appear on the match summary before 2009. This led me to return to the incident report dataset and remove all the matches from 2002-2009 since my hypotheses relied heavily on the referee's name existing for each game. This unfortunately forced me to remove approximately 2660 rows of match data.

9.3. Hypotheses, Models, and First Passes

9.3.1. Hypotheses and the Master Dataset

I began this project with three guiding research questions.

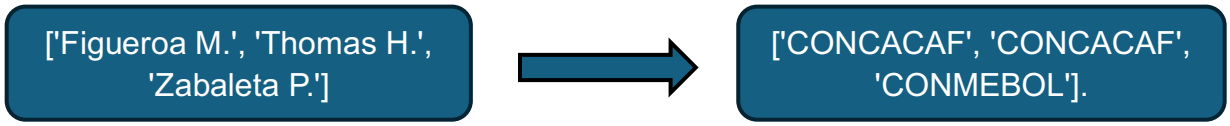
1. Could a model identify the referee of a match solely from the game statistics and stadium information provided?
2. Using that same information, would the model treat the presence of foreign players receiving cards as a meaningful feature?
3. And could the model determine the match outcome [win, loss, draw] when given the combined inputs of game statistics, referee identity, and stadium data?

These questions shaped how I designed and structured the dataset that would ultimately be fed into the models.

To work toward any answers, I needed a finalized and comprehensive master dataset. As discussed earlier, the incident report dataset functioned as the project's

“source of truth” but it required augmentation. I needed to layer in referee information as well as the nationality-based player data. Integrating the referee data was straightforward: after running a script that captured the referee listed on each FlashScore match report, I joined those results to the Kaggle dataset through an index-match process in Excel. The more complex issue involved linking the player dataset to the incident report dataset. Because I could not reliably map every individual player to every individual match, I needed a more flexible solution that still preserved the integrity of the information.

This led to the creation of an additional script, which I divided into two parts. The first part extracted every player who received a card in each match, reduced the list to unique names, and then mapped each name to its confederation. A processed incident report, for example, might look like:



This yielded an array of confederation labels for each match. Given that there are only six continental confederations (AFC, CAF, CONMEBOL, CONCACAF, OFC, UEFA), it was logical to present them directly as six columns in the master dataset.

The second script handled this transformation. It took the confederation array, generated the six corresponding columns, counted the occurrences of each confederation, and assigned the totals to their respective columns. Using the example above, the output would appear as:

Confederations	AFC	CAF	CONMEBOL	CONCACAF	OFC	UEFA
['CONCACAF', 'CONCACAF', 'CONMEBOL']	0	0	1	2	0	0

The output file also retained the incident report for each match, which allowed me to merge these new confederation counts into the master dataset using simple index-match logic. At this point, the dataset finally felt complete: it contained match statistics, referee identities and origins, and player confederation data consolidated into a single, coherent document.

Even with a comprehensive dataset, I still needed to refine the feature space to what was most relevant to the project’s hypotheses. The objective was not to overload the model with peripheral information but to isolate the most meaningful variables. As a result, certain columns were excluded from the final master dataset: matchday

timestamps, the matchday number, timing-specific scoring data (such as goals scored within the first 15 minutes), offsides, shots on target, the original match-report URLs, and other values that added noise rather than insight. The resulting CSV served as a focused, cleaned, and more analytically useful version of the original incident report dataset.

9.3.2. Model Implementation and Process

The project initially began in Visual Studio Code because it provided a convenient environment for developing Python scripts and handling the early stages of data manipulation. As the project shifted toward the machine learning and modeling stage, it became clear that Jupyter Notebook offered a more effective workflow. The ability to execute code in modular cells, validate intermediate outputs, and visualize transformations step-by-step made the iterative modeling process substantially more transparent. While Visual Studio Code can mimic this behavior, Jupyter Notebook proved far more intuitive for analytical work that required continual reflection and adjustment.

Because the bulk of the time had already been invested in data collection, normalization, and cleaning, the modeling stage was conceptually straightforward, at least in structure. I outlined the workflow as follows:

1. Load the data
2. Define targets and features
3. Label encode the target
4. One-Hot encode key categorical columns
5. Impute and Scale
6. Split Train/Test
7. Train Random Forest
8. Evaluate the performance
9. Feature Importance
10. Visualizations
11. Multiple Models
12. More Visualizations

This sequence provided a backbone for the modeling process and allowed me to iterate and make corrections without losing the larger sense of direction. Because my research questions required predicting multiple referee identities and determining whether a match resulted in a win, loss, or draw, the task was inherently a multi-class classification problem. Scikit-Learn was the most suitable choice for this work as it offered mature implementations of the necessary preprocessing tools and classifiers, required minimal boilerplate code, and allowed me to focus on experimentation rather than software engineering.

I began by loading the master dataset in Pandas. During this step, I noticed that several referees had officiated very few Premier League matches. To avoid giving the model classes with extremely sparse representation, which would affect the bias of both learning and evaluation, I counted match appearances for each referee and removed those with fewer than seventy games. Many of these referees were also the ones whose birth city or demographic was difficult to obtain and excluding them ultimately produced a cleaner and more stable dataset.

Next came defining the targets and features. I created a Boolean switch, TargetReferee, to indicate whether the model's target variable should be "Referee" or "WIN". The "Referee" target corresponded to the individual referee for the match, while "WIN" represented the match outcome (home victory, loss, or draw). This Boolean mechanism allowed me to reuse the same pipeline structure for two separate hypotheses without rewriting the preprocessing logic.

As I developed the model further, I revised the feature definitions and removed additional columns. When predicting "WIN", the model achieved near-perfect accuracy in early iterations, not because of any deep pattern recognition, but because I had mistakenly allowed it to see the final score. The model simply learned the trivial rule that a higher home score meant a win and so on. To avoid this leakage, I introduced a conditional preprocessing step that dropped the score columns entirely whenever "WIN" was the selected target.

Label encoding became the next essential step. Before the model could interpret the referee names or match outcomes, these targets had to be expressed numerically, for example, mapping "Howard Webb" → 0 and "Mike Dean" → 1. This transformation preserved the categorical meaning while converting the values into a form suitable for learning algorithms. I applied the same encoding to the logic to the "WIN" target when that was selected.

The following phase involved one-hot encoding the remaining categorical columns. Although many match statistics (such as shots, fouls, or possession) were already numeric, numerous core attributes were not, such as stadium names, team names, birthplaces, and the target column itself when not encoded. Machine learning models cannot directly interpret text-based categories, so each category was expanded into its own binary feature, for instance, Stadium_Anfield = 0/1; Stadium_Emirates = 0/1. This allowed the model to recognize categorical distinctions without introducing artificial numeric relationships.

With all categorical data numerically encoded, I moved to imputation and scaling. The order here is critical. Encoding must happen first so that all features exist in numeric form; only then can they be imputed or standardized. In my first approach, I used Scikit-Learn's SimpleImputer to fill missing values with the column mean. This ensured that no

null values would disrupt model training. After imputation, I applied StandardScaler to bring all features to comparable magnitudes, preventing large-scale variables from dominating the model's internal calculations.

Because the first approach produced lackluster results, I returned to this stage and implemented a second imputation strategy using the K-Nearest Neighbors (KNN) imputer. KNN imputation identifies similar rows based on distance and uses them to estimate missing values. For this reason, I scaled the data before applying the KNN imputer, otherwise, features with larger numerical ranges such as attendance would overwhelm distance calculations and distort the imputation. After scaling, KNN imputation produced more coherent and context-aware estimates.

Steps six and seven, splitting the data and training the Random Forest, were comparatively straightforward. Given the size of the dataset, an 80/20 train-test split provided a balance between maximizing training information and preserving a meaningful evaluation set. I trained the model using Scikit-Learn's RandomForestClassifier with balanced class weights and a fixed random seed of 42. This approach was applied to both target configurations. When testing the "WIN" target, however, the model struggled to identify draws due to their relative rarity. I addressed this by adjusting the class weights to amplify the importance of the draw class, improving the model's sensitivity to that outcome.

Evaluating performance and analyzing feature importance formed the next pair of steps. I used Scikit-Learns' classification_report to produce precision, recall, F1-scores, and support values. These metrics revealed whether the model was learning the intended patterns or simply defaulting to high-frequency classes. Feature importance analysis was central to the broader question of potential discrimination. By creating a dataframe of features and their associated importances, sorting them, and examining the top thirty contributors, I could assess whether confederations or player origins disproportionally influenced predictions. I also aggregated the importance of the confederation features and compared them against match statistics and referee birthplaces to identify any concerning patterns.

Finally, I visualized the most important features using bar charts. These visualizations offered a direct and interpretable summary of what the model considered influential. Although the classification report conveyed similar information numerically, the charts supported a more intuitive understanding.

As an extension of the analysis in the final two steps, I also compared multiple models – Logistic Regression, Random Forest, Decision Tree, Gradient Boosting, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). Scikit-Learn's unified API made this process simple, I imported each classifier, fit the data, and produced the same bar-chart visualizations to identify which algorithms performed the best and which

struggled. This provided a richer context for evaluating the Random Forest's behavior and helped validate the reliability of the final conclusions.

9.4. Results and Revisions

9.4.1. Results

The early outputs from the models looked encouraging on the surface. The average f1-score hovered around 0.68, roughly a 70% accuracy rate, which for a first pass felt surprisingly strong. The number was almost too good and that made me uneasy. In several instances, the model guessed the referee correctly every time. That kind of consistency raised all the usual red flags: had I accidentally given the model too much information? Too little? Was it exploiting some hidden relationship I had not accounted for? The results were good enough to tempt me into accepting them but unsettling enough to push me to investigate further.

Accuracy: 0.6793478260869565
Classification Report:

	precision	recall	f1-score	support
0	0.79	0.64	0.71	72
1	1.00	1.00	1.00	43
2	0.63	0.63	0.63	19
3	1.00	1.00	1.00	41
4	1.00	1.00	1.00	22
5	0.76	0.47	0.58	74
6	1.00	1.00	1.00	32
7	1.00	1.00	1.00	18
8	0.42	0.33	0.37	15
9	0.47	0.74	0.57	27
10	1.00	1.00	1.00	54
11	0.65	0.68	0.67	25
12	1.00	1.00	1.00	38
13	0.30	0.50	0.38	18
14	0.17	0.27	0.21	33
15	0.34	0.52	0.41	23
16	0.19	0.21	0.20	19
17	1.00	1.00	1.00	67
18	0.60	0.54	0.57	50
19	0.48	0.36	0.41	55
20	0.62	0.72	0.67	80
21	0.72	0.85	0.78	54
22	1.00	1.00	1.00	31
23	0.60	0.57	0.59	21
24	0.74	0.74	0.74	27
25	0.41	0.24	0.30	80
26	0.21	0.32	0.26	37
27	0.95	0.72	0.82	29
accuracy			0.68	1104
macro avg	0.68	0.68	0.67	1104
weighted avg	0.70	0.68	0.68	1104

Before circling back to diagnose the "too-good" performance, I tried to refine the model using Scikit-Learn's parameter grid search. This tool let me explore a wide range of hyperparameters systematically and identify a seemingly optimal configuration. After running the grid, the recommended settings were: 200 estimators, a minimum split size of 2, a minimum leaf size of 4, and a maximum depth of 20. When I plugged these values in, the effects were mixed. The overall average f1-score dropped from about .74 to .68, but the macro average improved from .62 to .67. At that moment, I took the improvement to the macro average as the more meaningful gain, especially since macro metrics weigh minority classes more fairly and kept the selected parameter combination.

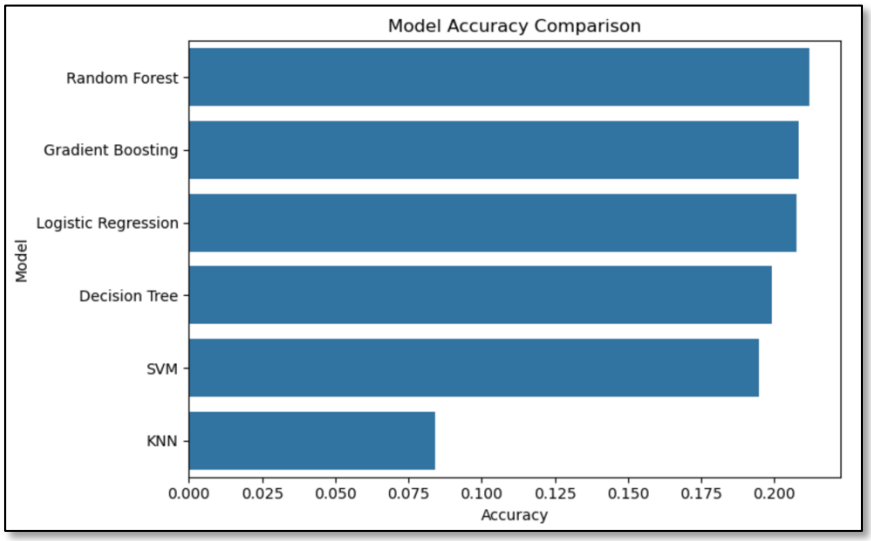
Once satisfied that hyperparameter tuning wasn't the core issue, I turned back to the original puzzle, why were the initial scores so high? A quick look at the feature importances offered the answer. The model was relying heavily on the "NearestMajorCity" field to determine the referee. This column represented the closest major city to each

Top 15 Most Influential Features:

	Feature	Importance
111	NearestMajorCity_Newcastle	0.054755
109	NearestMajorCity_Manchester	0.048867
107	NearestMajorCity_Leeds	0.046737
108	NearestMajorCity_Liverpool	0.045774
116	NearestMajorCity_Sheffield	0.041834
103	NearestMajorCity_Bristol	0.041224
105	NearestMajorCity_Coventry	0.032167
106	NearestMajorCity_Durham	0.031829
104	NearestMajorCity_Chester	0.029163
8	H_Ball_Possession	0.027288
9	A_Ball_Possession	0.025839
117	NearestMajorCity_Stoke	0.024922
15	A_Fouls	0.023887
12	H_Corner_Kicks	0.023707
11	A_Free_Kicks	0.023602

referee's birthplace, a perfect shortcut. If the data showed that a referee like "Oliver M." had "Newcastle" listed as his nearest major city, and no other referee shared that label, then the model needed nothing else. The presence of "Newcastle" in the encoded data became a direct pointer to "Oliver M.". It functioned as a lookup table disguising as a feature.

Realizing this was frustrating because the birthplace data had taken substantial time and manual effort to collect. That said, the feature amounted to data leakage, so it had to be reworked. I revised the dataset by replacing the specific major city with a broader regional classification: North, Midlands, or South. This change forced the model to abandon the one-to-one mapping and instead work with fuzzier geographic categories. The effect was immediate and dramatic. The Random Forest's average f1-score plummeted to around .21. The other models followed suit, all landing in roughly the same range. Some performed marginally better, but none came close to the artificially inflated numbers from before. As expected, K-Nearest Neighbors struggled the most, and even the KNN-based imputation I experimented with later did not produce meaningful gains.



The models, without their shortcut, now had to rely on far more subtle statistical patterns. Patterns, as it turns out, do not strongly distinguish one Premier League referee from another. Further exploration continued from that more honest baseline.

9.4.2. Revisions in the Dataset

After a few modest attempts to boost the model's performance, I returned to the dataset itself and began testing ways to enrich it. The hope was that better inputs, whether new features or refined ones, might push the average accuracy upward. My first idea was to incorporate stadium capacity and attendance. I suspected that crowd size might exert some influence on a referee's decision-making, at least indirectly. The obstacle, however,

was practical: collecting accurate attendance figures for every match would have required a substantial time investment. To work around this, I approximated the attendance as 80% of stadium capacity. Only later did it become clear that this shortcut had effectively normalized the column into a constant multiplier. The result was a pair of features that looked meaningful but behaved like noise; unsurprisingly, they offered no improvement to the model.

Continuing along this path, I introduced a “Season Start Year” feature, imagining that the context might shrink the model’s search space. If a referee only worked in certain seasons, perhaps the model could use that information to weigh its guesses more effectively. This also turned out to be wishful thinking. The new feature produced almost identical results to the previous attempt – statistically flat, practically useless.

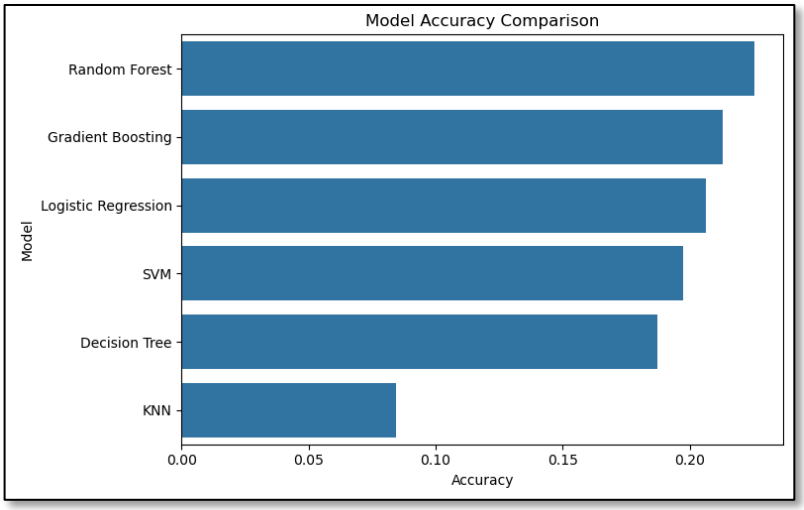
I tried a similar experiment with a “Penalties Awarded” feature, extracted directly from the incident reports. The intuition was simple enough: penalties are high-impact decisions, and perhaps they vary enough by referee to supply a meaningful signal. Once again, the model disagreed. The feature made no measurable difference in accuracy or in the distribution of feature importances.

My final attempt at enriching the dataset involved a bit of feature engineering. Instead of keeping six separate columns for the confederation counts, I condensed them into a single metric: the proportion of foreign players carded. This involved summing all carded foreign players and dividing by the total number of cards shown. The idea was that a normalized ratio might reduce noise and reveal subtler patterns the raw counts could be hiding. In theory, it was elegant; in practice, it did nothing. The model’s performance remained unchanged, and the new feature did not meaningfully elevate its importance.

These experiments made the limitations of the dataset increasingly clear. Even carefully reasoned features were not strong enough to shift the model in a significant direction, and the earlier results, impressive as they seemed, had been inflated by the very data leakage I later removed.

9.4.3. Final Results – Hypotheses Reflection

Despite my best effort and multiple iterations of data cleaning and model tuning, the results ultimately fell short of what I had hoped to uncover. My first hypothesis asked whether a model could identify the referee of a match using only game statistics and stadium information. I cannot say the answer is a firm “no”, because a different dataset or a more refined modeling strategy might reveal patterns that mine did not. What I can say is that with the features I collected, the model failed to make accurate determinations. The accuracy remained unconvincing even after I simplified the dataset, removed noise,



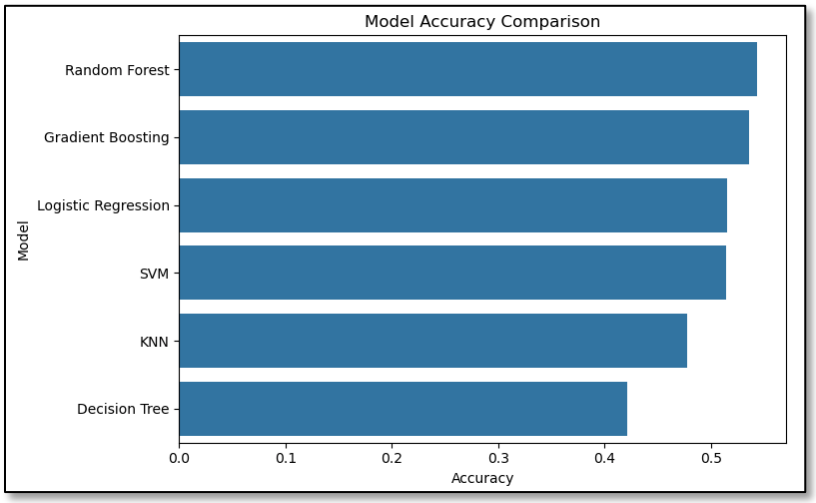
and performed several rounds of preprocessing. The model accuracy comparison chart displayed this point clearly: no matter how I shaped the inputs, the predictions simply refused to be anything meaningful.

My second hypothesis followed a similar trajectory. I wanted to know whether the model treated foreign players receiving cards as a meaningful feature. I broke apart the federation data, isolated the relevant features, and evaluated their importance rankings separately. The outcome was consistent and consistently underwhelming. None of these features surfaced in the top 15 or even top 30 most influential predictors, and their importance values were roughly statistically irrelevant. As before, this does not entirely disprove the hypothesis, it only shows that the data available to me did not capture whatever underlying signals might exist. A richer dataset, different encoding strategies, or more expressive models may reveal patterns that mine could not.

```
Federation Feature Importance
              Feature  Importance
6      Region – UEFA    0.022264
2       Region – CAF    0.012895
3  Region – CONMEBOL    0.012786
4  Region – CONCACAF    0.004748
1       Region – AFC    0.001824
5       Region – OFC    0.000493

Grouped Feature Importances:
Feature
Federation          0.055010
MatchStats          0.741613
RefereeBirthCity    0.203377
```

The third hypothesis asked whether the model could predict match outcomes [win, loss, draw] when given a combined set of features: game statistics, referee identity, and stadium context. Early attempts produced weak results. Adjusting model weights helped modestly but the improvement was far from transformative. Using a Random Forest classifier, the overall accuracy settled around 0.5. When broken down by class, the model was approximately 0.6 for predicting wins, 0.5 for losses, and .25 for draws. These results are barely



distinguishable from what a random person might guess with no data at all. In practice, the model was trying, but the dataset signals were possibly too weak or had too much noise to provide reliable predictions. As with the previous hypotheses, this led to an inconclusive outcome.

Together, these results do not rule out the existence of meaningful patterns; they simply show that the patterns did not emerge from the features and scale of data I used. The underlying questions remain valid, but their answers require deeper, richer, and more granular inputs than this project could provide.

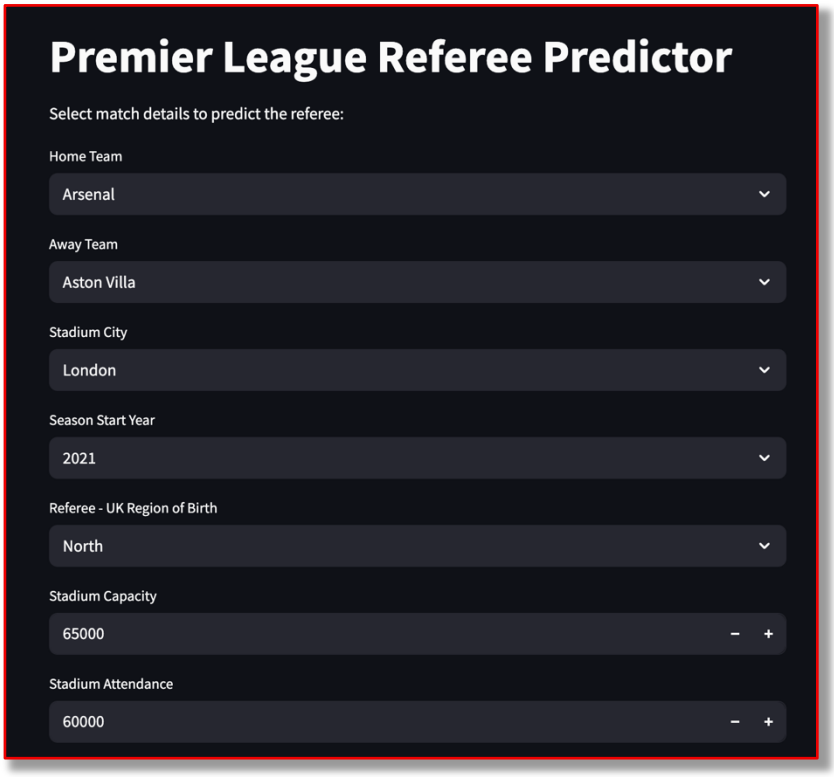
9.5. Full Stack Implementation

Despite the model's underwhelming performance, I shifted my attention to the front-end implementation. The model already existed as the project's brain, trained, exported, and ready to receive inputs, so the remaining task was to build a simple interface that could feed it data. I needed a tool that could speak Python fluently, handle preprocessing quietly behind the scenes, and present everything in a clean, usable way. That narrowed the options to Flask and Streamlit. Flask offered full control but demanded more time: routes, HTML templates, deployment structure, and the usual issues that comes with a traditional web framework. Streamlit, by contrast, let me circumvent most of that complexity. It is essentially a friendly window into Python itself, perfect for demonstrations and quick iteration. Given the experimental nature of the project, Streamlit was the clear choice.

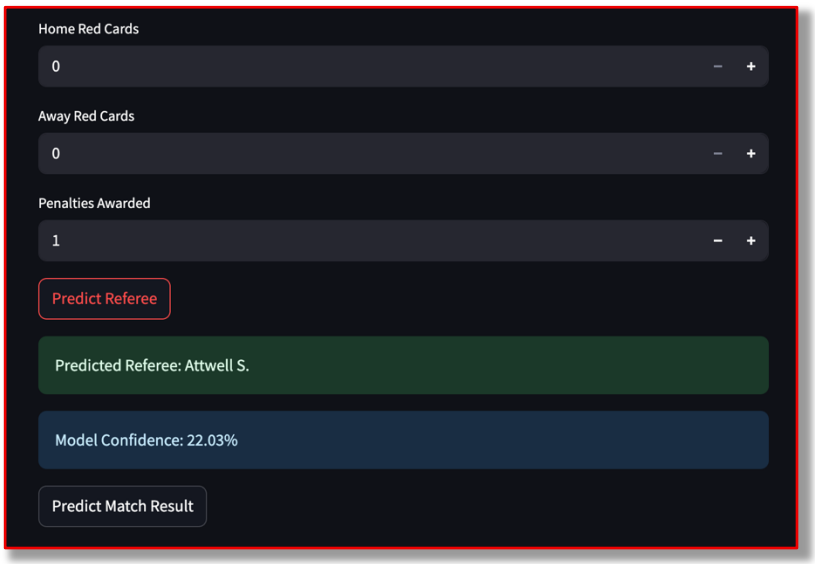
The front end's purpose was straightforward: gather match information, send it to the model, and present a prediction along with confidence levels. With a more reliable model, this kind of interface could someday act as a playful forecasting tool, letting people guess how a match might unfold based on who is officiating. Even with my model's modest accuracy, the structure was worth building.

The Streamlit setup lives in a single file, app.py. The comments in the code walk through each stage, but the flow of the application unfolded in five small steps: loading the trained model, loading the dataset for reference, constructing the interface, preprocessing the categorical inputs, and generating a final prediction. Exporting the model during the JupyterLab phase made the first step easy enough, though it came with a small twist, the model depended on its label encoder, imputer, and scaler. All these components needed to be reloaded and applied exactly as they were during training, which turned out to be a useful lesson in the importance of reproducibility.

Building the interface was simple: titles, headers, dropdowns, and numeric fields, but the assumptions I carried into the process fell apart quickly. I had imagined a sleek interface where the user selected two teams and clicked “Predict Referee”. However, the model had other ideas. Because it relied on a suite of match statistics like possession, cards, shots on target, attendance, referee region, and more, every one of these features had to be included as an input. That meant gathering user responses, assembling them into a one-row data frame, and transforming them to mimic the shape of a single match from the training set.



To maintain consistency with the training pipeline, I applied the same preprocessing steps. Categorical variables were one-hot encoded, but because a single row does not automatically generate all the category columns seen during training, I had to re-add any missing dummy columns by hand, filling them with zeros. Only then did the structure line up perfectly with the model’s expectations. After encoding, the user inputs were passed through the imputer and scaler, replicating the same sequence of transformations that the model learned from.



By the end of this chain, the interface produced a clean, fully preprocessed row that mirrored the training data. A single button press would then trigger the model to

predict either the referee or the match outcome (win, loss, or draw) and return its confidence in that prediction. The result never being perfect, but it was a faithful version of the model's logic.

9.6. Conclusion

The project ultimately fell short of the outcomes I had hoped for. I expected the model to identify stronger patterns, higher confidence levels, clearer signals, perhaps even subtle indications of bias. When the results failed to show anything conclusive, it felt discouraging at first. Yet the absence of detectable patterns in my dataset does not imply the absence of bias in the Premier League. It only means that the specific features I examined were not sufficient to support or refute that claim. My broader belief remains unchanged: bias and racial discrimination exist in many forms, and they tend to surface wherever humans make rapid, high-pressure decisions. If discrimination is visible among fans, it is difficult to argue that it disappears entirely at the referee level. However, quantifying it is far more difficult than anticipating it.

Looking back, the Premier League may have been an overly ambitious starting point. It is one of the most scrutinized, financially powerful, and professionally regulated leagues in the world, with layers of oversight that likely decrease the probability of bias. A more methodological path would have been to start with smaller leagues, places where disparities in officiating quality, training, or accountability might be more noticeable. Detecting patterns there could have helped me identify which features are genuinely predictive and which are just noise, before moving to larger or more sophisticated leagues.

One more direction I could not explore due to resource constraints was video analysis. Many forms of bias manifest not in the fouls that are called but in the fouls that are ignored, the borderline challenges, the plays where advantage is allowed, the moments that shift a match's momentum. Capturing that kind of referee decision-making would likely require computer vision tools capable of detecting fouls and non-fouls across full-length matches and outputting structured data. Even with the computational tools, access to match footage poses its own hurdle, as broadcast rights and league ownership make these materials difficult to obtain for academic purposes.

Referee bias remains an active area of debate in sports analytics. While my findings were inconclusive for the Premier League, I remain optimistic that future work, whether using this dataset or an improved one, will push the question forward. Bias in officiating is a real and important issue, and with stronger data, more refined features, and advancements in machine learning, it may eventually be possible to demonstrate it in a quantifiable way.

10. References

[1] “How the premier league continues to fight racism,” www.premierleague.com, Apr. 05, 2024. https://www.premierleague.com/news/3951098

[2] ”How Our Club Predictions Work” FiveThirtyEight.com https://fivethirtyeight.com/methodology/how- our-club-soccer-predictions-work/ [Accessed Feb. 24, 2025].

[3] W. Haanstra F. Keijzer, ”Learning from adjacent fields: the relation between extremism and hooli- ganism”, Oct 2018. [Online]. Available: https://home-affairs.ec.europa.eu/document/download/cf539884- 5f8f-4df3-a623-d1434959c622 en. [Accessed: Feb. 24, 2025].

[4] ”Roma fans condemned for harassing Europa League final referee Anthony Taylor at airport” ESPN.com. https://www.espn.com/soccer/story/ /id/37776087/roma-fans-condemned-harassment-europa-league- final-referee-anthony-taylor. [Accessed Feb. 24, 2025].

[5] F. Kallas. ”Spanish referees condemn abuse of official who gave Bellingham red card’. Reuters.com. https://www.reuters.com/sports/soccer/spain-federation-says-referees-sickened-by-abuse- following-bellingham-red-card-2025-02-18/. [Accessed Feb. 24, 2025].

[6] T. Dohmen and J. Sauermann, ”Referee Bias”, Discussion Paper Series, Institute for the Study of Labor (IZA), no. 8857, Feb. 2015. [Online]. Available: https://docs.iza.org/dp8857.pdf [Accessed: Feb. 24, 2025].

[7] B. Lucey and D. Power, ”Do Soccer Referees Display Home Bias?”, SSRN Electronic Journal, Vol. 552223, Jul. 2009.

[8] B. Buraimo, R. Simmons, M. Maciaszczyk, ”Favoritism and Referee Bias in European Soccer: Evidence from the Spanish League and the UEFA Champions League”, Contemporary Economic Policy, Vol. 30, no. 3, pp. 329-343, Dec. 2011.

[9] R. Samuel, G. Tenenbaum, and Y. Galily, ”An Integrated Conceptual Framework of Decision-Making in Soccer Refereeing”, International Journal of Sport and Exercise Psychology, Vol. 19, no. 5, pp. 738-760, Apr. 2020.

[10] R. Boyko, A. Boyko, and M. Boyko, ”Referee Bias Contributes to Home Advantage in English Premier- ship Football”, Journal of Sports Sciences, Vol. 25, no. 11, pp.1185-1194, Jul. 2007.

[11] S. Stone, “Premier League: 13 VAR mistakes in total in season so far, say chiefs,” BBC Sport, Feb. 04, 2025. Available: https://www.bbc.com/sport/football/articles/cd9qvdg57n0o

[12] “Fans sentenced to prison for racist insults directed at soccer star Vinícius Júnior in first-of-its-kind conviction - CBS News,” www.cbsnews.com, Jun. 10, 2024. https://www.cbsnews.com/news/vinicius-junior-soccer-fans-sentenced-to-prison-racist-insults-spain/

[13] F. Müller, L. van Zoonen, and L. de Roode, “We can’t `Just do it’ alone! An analysis of Nike’s (potential) contributions to anti-racism in soccer,” Media, Culture & Society, vol. 30, no. 1, pp. 23–39, Jan. 2008, doi: https://doi.org/10.1177/0163443707084348.

[14] Wikipedia Contributors, “List of foreign Premier League players,” Wikipedia, Nov. 30, 2025.

11. Appendix

11.1. Table A1

Data Points	Data Type	Reason:
Home Team Name	String	Good to know
Away Team Name	String	Good to know
Home Team Score	Int	To determine if bias helped home team
Away Team Score	Int	To determine if bias helped away team
Home Team # of Penalties	Int	To determine if referee gave home team (or favorited team) more penalties on average.
Away Team # of Penalties	Int	To determine if referee gave away team (or favorited team) more penalties on average.
Home Team TimeStamp of Goals	List of Floats/Ints	Are there more yellows/red after scoring late to slow game down.
Away Team TimeStamp of Goals	List of Floats/Ints	Are there more yellows/red after scoring late to slow game down.
Name of Stadium	String	To help determine if there's a bias of a particular team. Maybe this helps with grouping?
Name of City for Stadium	String	To help determine if there's a bias for the city [e.g. Manchester vs Liverpool].
Added Time after 90'	Int	To determine if losing team had extra amount of time compared to normal to equalize or win
Time Game Ended	Int/Float	To determine if game ended when it should [extra time on top of extra time for losing team bias]
# of Home Team Yellow Cards	Int	Good to know
# of Away Team Yellow Cards	Int	Good to know

Yellow Card: - Player Name - Player Nationality - TimeStamp - Foul Type - Explanation	List of Objects - String - String - Int/Float - String - String	Looking for patterns on racial bias. Looking for patterns of when, why [more yellow cards when x team is losing?] to see general bias
# of Home Team Red Cards	Int	Good to know
# of Away Team Red Cards	Int	Good to know
Red Card: - Player Name - Player Nationality - TimeStamp - Foul Type - Explanation - Direct Red - Second Yellow	List of Objects - String - String - Int/Float - String - String - Boolean (T/F) - Boolean (T/F)	Looking for patterns on racial bias. Looking for patterns of when, why [more red cards when x team is losing?] to see general bias Want to know if ref hands out more direct reds vs certain teams, vs second yellows.
HOME - Starting XI + Bench: - {Squad Name, Nationality}	List of Objects - Object with 2 strings	Use it to compute ratio of nationalities vs yellow/red cards given to home team.
AWAY - Starting XI + Bench: - {Squad Name, Nationality}	List of Objects - Object with 2 strings	Use it to compute ratio of nationalities vs yellow/red cards given to home team.
Referee Name	String	Good to know.
Referee Nationality	String	Good to know.
Referee City	String	Good to know.
Date	DateTime	To establish a timeline.
Matchday	Int	To determine if there was more bias towards the end of the season.