UC3M

Data Integration and Visualization

SOCCER WORLD CUP ANALYSIS

ARIADNA GARCIA CORTIZO |
JOSHUA NASS |
RAGHAV THONDIKULAM |
NAHUM EPHREM |
JAY CHANG |

INDEX

1. Executive summary

This project aims to create a data integration system focused on predicting football match outcomes, particularly for the FIFA World Cup, and updating yearly with new data from international matches. The system utilizes historical match data (from 1872 to 2024), including team scores, penalties, goals scored, country's political situation, and more, to predict which countries will qualify for future World Cups and how they will perform during the tournament.

The primary data sources consist of three datasets: match results, penalty shootouts, and goalscorers. These datasets are integrated to provide a comprehensive view of team performances, including full-time results, penalty outcomes, and goal statistics. Using machine learning models, specifically classification and regression techniques, the system aims to predict match outcomes such as the winner, goals scored, and potential penalties. The system also accounts for factors such as FIFA rankings and recent team performance.

The proposed solution is designed to handle yearly updates automatically, integrating new match data and retraining models to ensure the most accurate predictions. This involves creating an ETL (Extract, Transform, Load) pipeline that processes new data and a machine learning model that retrains with updated inputs. The platform will be scalable, capable of handling increasing volumes of data as more international matches are played each year.

With predictive analytics as the core functionality, the project will also provide insights into key questions such as which teams are more likely to succeed in future tournaments, how penalties influence outcomes, and how player performance affects team success.

2. Analysis and design of the proposed system

2.1. Overall system architecture

Our overall system architecture involves several key components working together to create an efficient data flow system:

1. **Data Collection:** Data such as historical World Cup statistics (match results, penalty shootouts, goalscorers, winners, etc.) and contextual factors (host country, attendance, etc.) are collected.
2. **Data Storage:** How our data is organized in databases (Excel files, SQL, CSV, etc.) for efficient data storage and retrieval.
3. **Data Processing:** Cleaning, preprocessing, and transformation of our raw data into a suitable format to load into our databases or directly into our pipelines
    a. Cleaning missing/redundant data, normalizing datasets, and utilizing ETL pipelines to extract raw data and automate the cleaning process.
4. **Visualization and Reporting:** Method of presenting our raw data and insights from our predictive models/analysis through dashboards and reports.

    a. **Dashboards:** Visualizing raw data, team predictions, tournament/match simulations, and key statistics (win probability, expected goals, player performance, etc.)
    b. **User Interface and Interaction:** Allowing users to interact with the system; simulating different outcomes based on matchups, historical patterns, team status, querying predictions, condensed reports, and use of APIs for third-party use (media outlets, sports betting, etc.)
5. **Feedback Loop:** As new match data is continuously collected, the system updates and retrains models to improve predictions.

2.2. Data Integration Description

The success of our predictive analytics system hinges on effectively integrating diverse data sources to create a unified view of football match outcomes. Our project utilizes a variety of formats, including CSV, Excel (.xlsx), and SQL databases, to gather historical match data and relevant metrics that inform our predictions for the FIFA World Cup.

Data Sources

1. **p**

Data Cleaning and Preparation

Before integration, we prioritize data cleaning to ensure accuracy and consistency across datasets. This process includes:

- **Handling Missing Values**: Identifying and addressing any gaps in the data, either through imputation or removal, to maintain the integrity of the analysis.
- **Standardizing Formats**: Ensuring that data is in a consistent format (e.g., date formats, team names) across all sources to facilitate smooth integration.
- **Removing Duplicates**: Eliminating any redundant records to avoid skewing results during analysis.

Data Integration Process

Once the data is cleaned, we proceed with integrating the datasets into a consolidated file. This combined dataset will provide a holistic view of match results and team performance metrics. The integration process involves:

- **ETL Pipeline**: We will establish an Extract, Transform, Load (ETL) pipeline to automate data processing. This pipeline will extract data from the various sources, transform it into a unified format, and load it into our analysis environment. We plan to use Apache NiFi during this process.
- **Database Management**: Utilizing a relational database to store integrated data allows for efficient querying and analysis. SQL queries will enable us to extract relevant information based on specific criteria, facilitating our predictive modeling efforts.

2.3. Functional Architecture

The goal of our data integration process is to consolidate and analyze historical match data spanning over a century to extract valuable insights. The datasets include thousands of football matches from different periods and regions, capturing key match statistics such as team performance, match outcomes, goals, and penalties. These statistics are crucial for analyzing trends and predicting future winners, particularly in major tournaments like the FIFA World Cup.

Data Sources
We work with multiple data formats, primarily:

- CSV files (results_clean.csv, summary_by_country_full.csv)
- SQL databases (winners.sql)
- Excel files (summary_combined.xlsx)

These files contain detailed match results, country-level performance summaries, penalty shootout outcomes, and historical tournament winners.

Data Extraction and Transformation
The raw data from these sources is first extracted. The CSV files are ingested directly into the system, while SQL databases are queried to retrieve the relevant data. All data is cleaned and standardized, addressing issues such as missing values, inconsistent formats (e.g. date formats), and discrepancies in country names or match identifiers.

Transformation Process:

- Normalization: Ensuring consistency across datasets, such as aligning team names and standardizing performance metrics.
- Aggregation: Combining match-level data with country-level summaries (e.g., combining results from summary_results.csv with shootout outcomes from summary_shootouts.csv) to generate holistic insights into a team's overall performance across tournaments and years.

Data Consolidation
Once the data is transformed, it is consolidated into a single comprehensive file that merges all relevant match statistics. This enhanced dataset provides a complete view of each country's historical performance, from match results to penalty shootouts and goals scored.

Result
This comprehensive file serves as the basis for further analysis, including visualizations that highlight key trends, such as which countries have consistently performed well in international competitions. These visual insights, combined with machine learning models, are used to predict future winners based on historical performance and emerging trends.

2.4. Description of the proposed system in execution (include how the system should work, how it would take data, update data, how the client interacts, etc.).

The proposed system is designed to automate the integration, processing, and analysis of football match data across various competitions, with a focus on matches, shootouts, and World Cup results. This description explains how the system functions, from acquiring data to updating, processing, and enabling client interaction.

1. System Workflow:

The system performs the following tasks:
- Data Acquisition: The system takes in raw data from multiple sources, including CSV files (`results.csv` for match results, `shootouts.csv` for penalty shootouts) and an SQL file (`winners.sql`) that stores historical World Cup winners.
- Data Cleaning and Preprocessing: The data is cleaned to remove any null values or duplicates, this is done with python functions. Each dataset is stored as a cleaned version (e.g., `results_clean.csv`, `shootouts_clean.csv`, `winners_clean.csv`).
- Data Integration and Analysis:
  - Match Results Analysis: For each country, the system calculates the number of matches played, wins, losses, and home victories. This is processed from `results_clean.csv`.
  - Shootout Analysis: The system also tracks shootout results by country, counting the number of shootout matches played, won, and lost. This information comes from `shootouts_clean.csv`.
  - World Cup Data Integration: Historical World Cup wins and the locations where each country won the tournament are appended to the country's overall match results summary. The data is pulled from the `winners_clean.csv` file, which stores the World Cup history.
- Data Merging: The results from match analysis, shootout analysis, and World Cup data are merged into a single comprehensive dataset (`summary_combined.csv`), providing an overall view of a country's football performance.

2. Data Flow and Updates:

- Data Input:
  - Results (`results.csv`): New football match data, including tournament name, teams, scores, and location, can be added periodically (e.g., after each major football season).
  - Shootouts (`shootouts.csv`): Data about shootout results is updated regularly as penalty shootouts occur.
  - Winners (`winners.sql`): World Cup results are updated once every four years, following each World Cup tournament. This includes the winner, runner-up, and the host country of the event.

- Updating Data:
  - The system is designed to take updated versions of each dataset. For example, once a new football season is completed or after a World Cup, new data can be appended to the `results.csv`, `shootouts.csv`, and `winners.sql` files.

- The system reprocesses the data whenever updates are made. Each time the script is executed, the new data is cleaned, analyzed, and integrated, ensuring the system's output is always up to date.

- Client Interaction:
  - Output Files: The main output of the system is `summary_combined.csv`, which contains a comprehensive summary of each country's football performance, including wins, losses, home victories, shootout results, and World Cup wins.
  - Client Interface: This output can be fed into a client-facing dashboard for real-time data visualization, allowing users (e.g., analysts or fans) to interact with the data and explore specific countries' performance over time. The CSV files can also be opened using common spreadsheet software for further analysis.

3. How the System Works:

- Automated Processing: Once data files are placed in the input directory, the system can be executed through a Python script. This script will automatically clean the data, integrate various sources, and generate the final output files.

- Data Structure:
  - Results Data: Captures all match information, including date, home and away teams, scores, tournament name, city, and whether the match was on neutral ground.
  - Shootouts Data: Captures date, home and away teams, and the winner of penalty shootouts.
  - World Cup Data: Captures the winner, runner-up, and the country where the tournament was held for each World Cup.

4. System Output:

The system's primary output is the merged dataset in `summary_combined.csv`, which offers insights into:
- Matches played, wins, losses, and home victories by country.
- Shootout matches played, won, and lost by country.
- World Cups won and the locations where each country won.

3. Proof of Concept

3.1. Scope.

This project focuses on the development of a comprehensive data integration system aimed to predict football match outcomes, but specifically, targeting the FIFA World Cup. Rather than relying on complex machine learning models to make these predictions, our system will simply leverage historical trends to make educated guesses about potential future winners.The scope encompasses the following key components: data integration, data sources, predictive analytics, automated updates, and scalability. More specifically, we have included an ETL (Extract, Transform, Load) pipeline to ensure seamless data ingestion, cleaning, and standardization from different sources, like CSV, Excel, and SQL files. This brings all of our datasets together to give us a wider view of match statistics and team performance. The system will also incorporate visualization tools to present these insights. We integrated historical match data spanning from 1872 to 2024, incorporating various factors such as team scores, penalties, goals scored, and socio-political influences on team performance. Our primary datasets to be utilized include, match results, penalty shootouts, and goalscorers. Based on these data inputs, our system's predictions will include the likelihood of a team winning, number of goals scored by each team, and the probability of penalties affecting match results. The system is also designed for yearly updates, automatically integrating newly collected match data and retraining the models to maintain prediction accuracy. The architecture will be built to accommodate increasing data volumes as more international matches are played annually.

3.2. Objectives.

1. Create Visualization for Historical Match Outcomes and Predictions

- **Objective**: Develop a visual representation of past football match outcomes and leverage that data to predict future winners.
- **Tools**: Use Python libraries such as Matplotlib or Seaborn, or platforms like Tableau and Power BI for creating interactive dashboards.
- **Steps**:
  - Collect and clean historical match data (e.g., wins, losses, draws, goals scored, penalties, and individual performance).
  - Plot team performance trends over time (e.g., wins/losses, goal differentials).
  - Create a heatmap or scatter plot to show team dominance by region or tournament year.
  - Include a forecasting model (e.g., time series analysis or machine learning models like logistic regression) to predict future winners based on historical performance data.

2. Enhance User Insights for Future Team Success

- **Objective**: Provide actionable insights into team success probabilities in future tournaments.
- **Steps**:
  - Create a predictive model that uses historical match data and individual player stats to predict match outcomes.
  - Visualize the probabilities of team success using gauge charts or radial charts.
  - Factor in dynamic metrics like win ratio, match statistics, and penalties.
  - Create a dashboard that allows users to adjust variables (e.g., recent performance vs. historical performance) and see updated predictions.

3. Ensure Data Freshness

- **Objective**: Implement an automated system for the yearly update of match data.
- **Steps**:
  - Set up a data pipeline to scrape or ingest updated match data annually (or in real-time during the season) from reliable sources like football databases
  - Use tools like SQL for database management and Python's Pandas library to handle the data flow.

3.3. Implemented Architecture

3.3.1 General architecture of the proof of concept (include connections, APIS, etc.).

The architecture of the proof of concept is designed to efficiently integrate, clean, and analyze soccer data from various sources. The system processes match results, shootout outcomes, and World Cup performance data to provide a comprehensive view of international soccer statistics. The components of the architecture include:

- **Data Sources**: The system incorporates multiple data files. These include a CSV file for match results, another for shootout results, and an SQL database for World Cup winners. This data is regularly updated and cleaned to ensure accuracy.
- **Data Cleaning and Integration Module**: The system includes a module to clean the data by removing duplicates and handling missing values. This module processes all sources before merging them into a unified dataset. The result is an integrated summary table containing match statistics, shootout outcomes, and World Cup victories, organized by country.
- **Database Management**: A local SQLite database is used to load and query World Cup data from the winners.sql file. The data is combined with results and shootouts, adding additional insights into each country's World Cup performance and hosting history.
- **Output and Analysis**: Once the data is cleaned and combined, the system generates CSV files summarizing the performance of each country. This includes total matches played, wins, losses, home victories, shootout results, and World Cup victories. The system is prepared for further analysis and the eventual training of machine learning models.

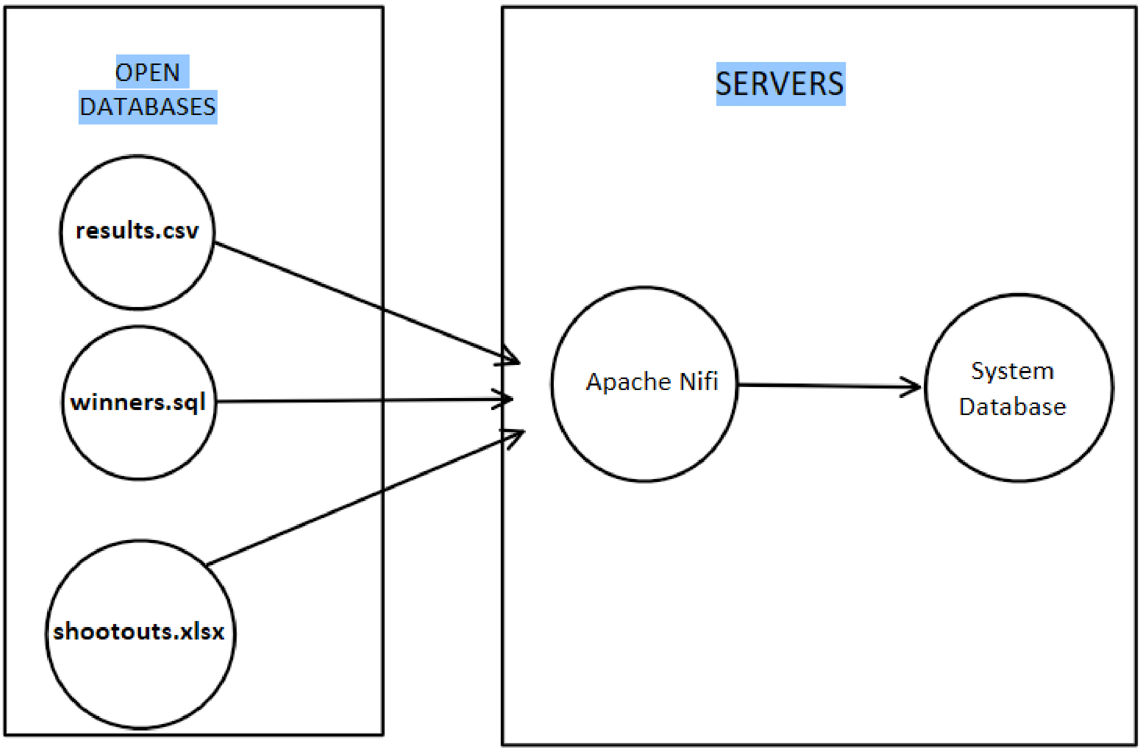
3.3.2 Description of the data integration

The core of the data integration process involves merging various datasets to provide an in-depth view of each country's soccer performance. Specifically:

- **Match Results**: Data on total matches played, wins, losses, and home victories are extracted and summarized for each country.

- **Shootouts**: Shootout outcomes, including victories and losses in penalty shootouts, are incorporated.
- **World Cup Data**: World Cup performance is added, including the number of tournaments won and the locations of those victories.

This integration enables the system to offer a comprehensive dataset (summary_combined.csv) that allows for deep analysis of soccer performance. Additionally, the system is flexible, making it possible to expand its data sources in the future.



Future Enhancements

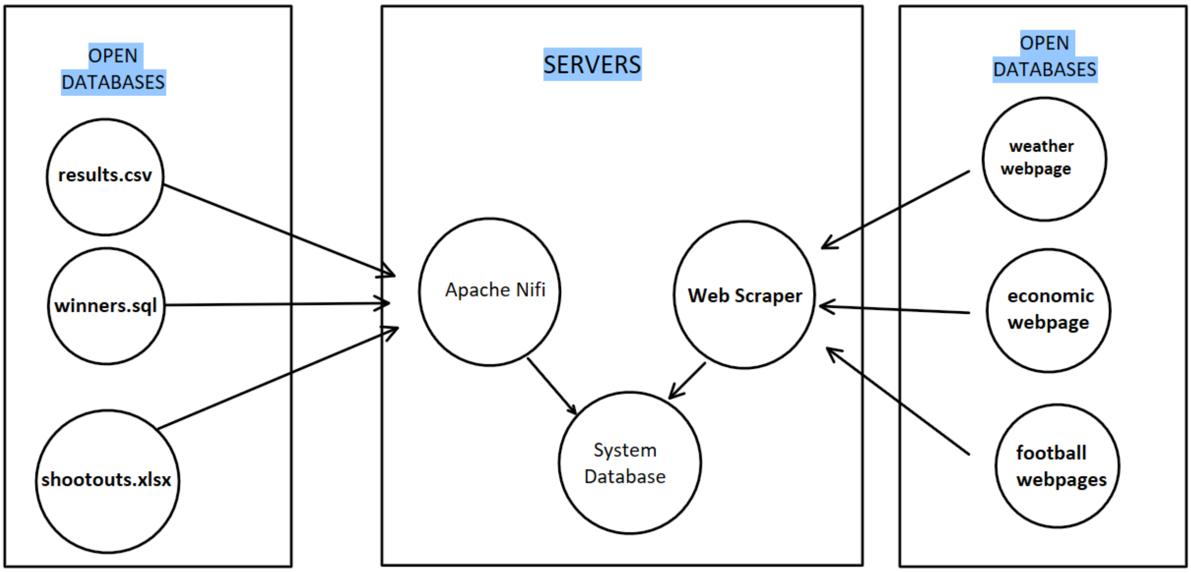
The current architecture provides a strong foundation, but there are several planned enhancements to further increase the system's capabilities and scalability:

- **Automated Web Integration for Soccer Results**: In future updates, the system could integrate with APIs or scrape live sports websites to fetch up-to-date soccer results from leagues and international tournaments. This would automate the process of updating the database and ensure that fans always have access to the latest statistics without manual intervention.
- **Weather Data Integration**: Another future enhancement is incorporating weather data from meteorological APIs. This would allow the system to analyze the impact of weather conditions—such as temperature, humidity, and precipitation—on match

outcomes, providing a deeper level of insight into soccer performance for each country.

- **Economic Data Integration**: To provide a broader context of soccer performance, the system could also pull economic data (such as GDP or sports investments) from public APIs. This would help analyze how financial factors influence the performance of national teams over time.

By integrating these additional data sources, the system would not only track soccer performance but also factor in external variables such as weather and economic conditions, making it a powerful tool for fans, analysts, and professionals looking for comprehensive insights into international soccer.



(The webpages names are written in references)

3.3.3 Functional Architecture of the Proof of Concept

The functional architecture of our predictive analytics system outlines the key components and their interactions, ensuring efficient data integration, processing, and analysis. This architecture encompasses the following layers:

Data Sources Layer

- This layer consists of various data sources in different formats (CSV, Excel, SQL) that feed into the system.
- Components:
  - Historical match data (winners.sql, shootouts_clean.csv, results_clean.csv)
  - External data sources (if applicable), such as FIFA rankings or news articles that may impact predictions.

ETL Pipeline Layer

- The ETL pipeline serves as the backbone for data extraction, transformation, and loading, ensuring that data is prepared for analysis.
- Components:
    - Extract: Automated scripts using Apache NiFi to gather data from multiple sources.
    - Transform: Data cleaning, formatting, and enrichment processes applied to the extracted data.
    - Load: The cleaned and transformed data is loaded into a relational database for further analysis.

Database Management Layer

- This layer manages the integrated dataset, providing a structured environment for data storage and retrieval.
- Components:
    - Relational Database: Utilizes a relational database management system (RDBMS) to store the integrated data, allowing for efficient querying.
    - SQL Queries: Customized SQL queries are executed to extract relevant data for predictive modeling and analysis.

Analytics Layer

- This layer encompasses the analytical tools and machine learning models used to generate predictions based on the integrated dataset.
- Components:
    - Predictive Models: Implementation of classification and regression algorithms to predict match outcomes, including winners and goals scored.
    - Data Visualization Tools: Integration with visualization tools (e.g., Tableau, Power BI, or Python libraries) to create dashboards and visual reports that convey insights.

User Interface Layer

- This layer provides an interface for users to interact with the system, input data, and view analysis results.
- Components:
    - Dashboard: A user-friendly dashboard that displays key metrics, predictions, and visualizations of match outcomes and team performances.
    - Reporting Tools: Functionality for generating reports based on specific queries or analyses.

3.4. Implementation

The implementation is structured into three main sections: data cleaning, data analysis, and merging results. The Python script handles various files (**results.csv, shootouts.csv**, and **winners.sql**) and follows a systematic approach for cleaning the data, analyzing matches, and generating summaries. Below are key steps:

1. **Data Cleaning:**
   - The cleaning process is uniform across all data sources. Missing values are dropped using dropna(), and duplicates are handled based on specific columns like match dates and teams.
   - Cleaned data is saved as *_clean.csv files.
2. **Data Analysis:**
   - **For Shootouts (**summary_shootouts.csv**):** The script calculates the total matches played by each country, the number of wins, and losses in penalty shootouts.
   - **For Results (**summary_results.csv**):** The total matches, wins, losses, and home wins for each country are computed by analyzing the results data.
   - **For Winners (**summary_by_country_full.csv**):** The script counts how many World Cups each country has won and where these tournaments were held.
3. **Data Merging:**
   - The final output combines the summary of results and shootouts by country, including columns for shootout wins and world cup achievements (**summary_combined.csv**).

3.5. Execution of the Proof of Concept (include here the instructions so that a proof of all can be executed)

To execute this proof of concept, follow these steps:

1. **Prerequisites:**
   a. Python must be installed (Python 3.11 or later)
   b. Install the following libraries: pandas and sqlite3. They can be installed using **pip**
2. **Data Setup:**
   a. Place the following files in the same directory as the Python script:
      i. **results.csv**
      ii. **shootouts.csv**
      iii. **winners.sql**
3. **Run the script:**
   a. To run the script in the terminal use “**python lab1.py”**
   b. The script will generate the following output files:
      i. **results_clean.csv**: Cleaned results data.
      ii. **shootouts_clean.csv:** Cleaned shootouts data.
      iii. **winners_clean.csv:** Cleaned winners data.
      iv. **summary_results.csv:** Summary of matches played, wins, losses, and home wins by country.
      v. **summary_shootouts.csv:** Summary of matches played and wins/losses in shootouts by country.
      vi. **summary_by_country_full.csv:** Extended summary including World Cups won and locations.
      vii. **summary_combined.csv:** The final combined summary with shootout wins and results.

3.6. Scalability of the system (explain how the proof of concept could scale).

This proof of concept can be scaled in several ways:

1. **Handling Larger Datasets:**
   - **Database Optimization:** Instead of using an in-memory SQLite database for handling the winners' data, a persistent database like MySQL or PostgreSQL could be used to store and manage data as it grows. This would allow for better performance and scalability when dealing with millions of records.
   - **Parallel Processing:** Libraries like **Dask** or **Modin** could replace pandas to handle larger datasets in parallel, improving the processing time as the data size increases.
2. **Automated Updates:**
   - The system can be updated yearly with new match results, shootouts, and World Cup winners. This can be automated through scheduled scripts that download new data from an API (such as a football stats provider) and update the CSV or database tables accordingly.
3. **Cloud-Based Solution:**
   - Moving the system to a cloud platform such as AWS or Google Cloud can ensure better scalability. Cloud-based databases, storage, and computational power can be used to handle the increased data volume and processing needs.
4. **Web Interface for Access:**
   - A web-based dashboard could be built to allow users to interact with and visualize the results in real-time. This interface could use the output CSVs to dynamically present data or integrate directly with the database for live queries.

By implementing these scaling strategies, the current proof of concept can evolve into a robust system capable of handling vast datasets, automating updates, and serving multiple users and functions with fast query responses.

4. Conclusions.

Our program is designed for the most passionate soccer enthusiasts, offering a comprehensive platform to track and analyze match results from countries around the globe. With our tool, users can easily see which nations are dominating the world of soccer, identify trends in World Cup qualifications based on historical data, and assess how factors such as home advantage or tournament location impact a team's performance.

Powered by machine learning, the program continuously improves, aiming to predict future outcomes such as World Cup qualifiers with greater accuracy. Fans could potentially leverage these insights for various purposes, including betting strategies or even detecting anomalies that could hint at match-fixing.

In the short term, our goal is to refine predictions around key metrics like yellow and red cards, penalties, corners, and other in-game statistics. With these enhanced capabilities, the platform will offer deeper insights into the sport, making it a must-have tool for any true soccer fan.

5. References.

Banerjee, S. (2021). *FIFA Football World Cup dataset*. Retrieved October 22, 2023, from https://www.kaggle.com/datasets/iamsouravbanerjee/fifa-football-world-cup-dataset

Ogakulov, M. (2023). *World Cup penalties awarded* [Dataset]. Kaggle. Retrieved October 22, 2023, from https://www.kaggle.com/datasets/ogakulov/world-cup-penalties-awarded

Statista. (2023). Goals scored per game at the FIFA World Cup from 1930 to 2022. Retrieved October 22, 2024, from https://www.statista.com/statistics/269031/goals-scored-per-game-at-the-fifa-world-cup-since-1930/

Statista. (2023). *Number of World Cup titles won by country from 1930 to 2022*. Retrieved October 22, 2023, from https://www.statista.com/statistics/266464/number-of-world-cup-titles-won-by-country-since-1930/

Trading Economics. (n.d.). *Countries*. Retrieved October 22, 2023, from https://tradingeconomics.com/countries

World Weather Online. (n.d.). *Football weather history*. Retrieved October 22, 2023, from https://www.worldweatheronline.com/football.aspx#google_vignette