Using Python and Machine learning to Predict Football Match/Tournament Winners

Minor Project Report

For

B.E. [Computer Science & Engineering]

7th Semester [CS-757]

By: **Bibhuti Singha** [UE213024]

Gitansh Sehgal [UE213037]

Under the guidance of

Dr. Ravreet Kaur

Assistant Professor, UIET

Department of Computer Science and Engineering

University Institute of Engineering & Technology

Panjab University, Chandigarh

CERTIFICATE

I hereby certify that the work which is being submitted in this project work titled " Using Python and Machine learning to Predict Football Match/Tournament Winners " in partial fulfilment of the requirement for the award of the degree of "Bachelor of Engineering in Computer Science and Engineering” submitted in UIET, Panjab University, Chandigarh, is an authentic record of my work carried out under the supervision of Dr. Ravreet Kaur and refers to other researchers work which is duly listed in the reference section. The matter presented in this project work has not been submitted for the award of any other degree of this or any other university.

Bibhuti Singha

UE213024

This is to certify that the statements made above by the candidate are correct and true to the best of my knowledge.

Gitansh Sehgal

UE213037

This is to certify that the statements made above by the candidate are correct and true to the best of my knowledge.

Dr. Ravreet Kaur

Assistant Professor, CSE, UIET,

Panjab University,

Chandigarh – 160014

VISION:

To be recognized as an eminent department in Computer Science and Engineering education and research for the benefit of society globally.

MISSION:

● To sustain world-class computing infrastructure for the enhancement of technical knowledge in the field of Computer Science and Engineering.

● To excel in research and innovation for the discovery of new knowledge and technologies.

● To produce technocrats, entrepreneurs, and business leaders of the future.

● To foster human values for national growth and life-long learning amongst all the stakeholders.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs):

I. Graduates will work as software professional in industry of repute.

II. Graduates will pursue higher studies and research in engineering and management disciplines.

III. Graduates will work as entrepreneurs by establishing startups to take up projects for societal and environmental cause.

PROGRAMME SPECIFIC OBJECTIVES (PSOs):

I. The ability to use software engineering techniques to design and develop software solutions.

II. The ability to employ data science principles to extract insights and knowledge from data.

PROGRAMME OUTCOMES:

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solution: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigation of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear infrastructure.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognition of the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

COURSE OUTCOMES (CO):

On completion of this course, a student will be able to

1. Apply the knowledge from previous semesters to undertake and solve a real-life problem

2. Illustrate the solution after identifying various objectives of the problem undertaken

3. Devise an organised action plan along with all the team members

4. Develop a solution using appropriate methodology and tools available

5. Communicate and demonstrate the work through structured reports and oral Presentation.

Abstract

Our project focuses on predicting football match outcomes using machine learning, specifically Logistic Regression. The project starts with preparing and cleaning the data to ensure it's ready for analysis. Then, we explore the data using charts and graphs to understand patterns and relationships. These patterns help us choose the most important features for the prediction model. Logistic Regression is used as the main method for prediction, and we improve its performance by testing different settings (hyperparameters) to find the best combination. The model's accuracy and ability to make good predictions are evaluated using metrics like ROC curves and AUC scores. We also look at which features (like team rankings or scores) have the most impact on the predictions, making the results easier to understand.

The results are visualized through clear charts, and tools like Streamlit are used to make the project interactive. The project uses Python libraries like NumPy, Pandas, Matplotlib, Seaborn, and Scikit-learn.

We created this project to build a reliable tool for predicting football match outcomes. This can be useful for fans, analysts, and organizations to understand factors that affect game results and make informed decisions, such as predicting match winners or evaluating team performance. It also demonstrates how machine learning can be applied to solve real-world problems in sports analytics.

Table of Contents

List of Figures

List of Tables

Introduction

Chapter 1: Project Research

1.1 Inspiration about the project

EA (Electronic Arts), the company behind the EA FC football game series, is known for using advanced simulations and algorithms to predict the outcomes of major football tournaments. In its game, EA FC, simulations are run to replicate real-world football scenarios, predicting outcomes based on team and player data. EA uses historical performance, player statistics, and team dynamics to simulate matches, providing an engaging and realistic experience for players.

The company has successfully predicted the winners of the last four World Cups by running these simulations, which consider various factors such as team strength, player form, and match conditions.

Inspired by EA's approach, this project aims to replicate similar prediction models for real-world football match outcomes. By analysing historical data spanning decades, the project applies machine learning techniques to predict match results, focusing on key factors like offense, defense, and goalkeeper performance. The goal is to offer insights for better decision-making in football and sports analytics, similar to how EA FC uses simulations to enhance its game experience and predict tournament winners.

1.2 Research on Project requirements

We conducted comprehensive research to determine the most effective methods, algorithms, and tools for building a reliable football match prediction system. After evaluating various approaches, we identified Monte Carlo simulation as a key technique due to its ability to model uncertainty and simulate numerous match scenarios. By incorporating variables such as team performance metrics, historical match data, and FIFA rankings, Monte Carlo simulations allowed us to analyze a wide range of outcomes and predict match results effectively.

Additionally, we explored machine learning algorithms like logistic regression, Random Forest, and SGD SVM, assessing their strengths in handling classification tasks and identifying patterns within complex datasets. These algorithms were chosen for their ability to improve prediction accuracy and uncover the relationships between key factors influencing match outcomes.

For the development environment, Python was selected for its robust ecosystem of libraries, including pandas, Scikit-learn, and Matplotlib, which facilitated efficient data analysis, machine learning implementation, and visualization. We also chose VS Code for coding and utilized Jupyter notebooks to structure and execute our code interactively, as it allows for seamless integration of data analysis, visualization, and machine learning tasks. To present the insights dynamically, we used Streamlit to build interactive dashboards, making it easier for users to explore and interpret the results. This combination of simulations, machine learning, and Python's versatility provided a strong foundation for our project.

Language: Python

Libraries :

Data Manipulation

- numpy (np): For numerical computations and array manipulations.
- pandas (pd): For handling tabular data and performing data analysis.

Visualization

- matplotlib.pyplot (plt): For creating static, interactive, and animated visualizations.
- seaborn (sns): For high-level, aesthetically pleasing statistical data visualization.

Machine Learning (scikit-learn)

- sklearn (sl): open source ML library.
- PCA: Dimensionality reduction technique to transform features.
- GradientBoostingClassifier: Ensemble learning algorithm for classification tasks.
- RandomForestClassifier: Ensemble-based classifier using decision trees.
- LogisticRegression: Logistic regression model for binary classification.

- SGDClassifier: Stochastic gradient descent-based classifier for large-scale learning.
- accuracy_score: Metric to evaluate the accuracy of a model.

- confusion_matrix: To analyze model prediction results.
- roc_curve: To compute and plot the ROC curve.
- roc_auc_score: To calculate the Area Under the ROC Curve (AUC).
- GridSearchCV: Hyperparameter tuning using exhaustive grid search.
- RandomizedSearchCV: Hyperparameter tuning with randomized search.
- train_test_split: To split data into training and testing sets.
- Pipeline: To chain multiple preprocessing steps and a model into one pipeline.
- PolynomialFeatures: To generate polynomial and interaction features for models.

Utilities

- Counter: To count occurrences of elements in a collection.
- tqdm: To display progress bars for loops and processes.
- tabulate: To format and display tabular data in a readable format.

Additional Libraries

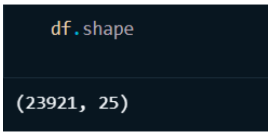
- streamlit (st): For building interactive web-based data science applications.
- datetime: To work with dates and times.
- warnings: To manage and suppress warnings during code execution.

User Interface: Streamlit

Chapter 2: Data Collection & Data cleaning

2.1 Data collection

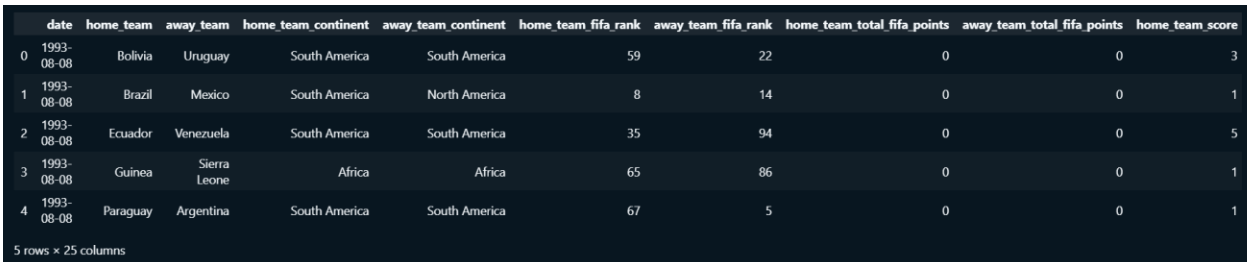
We used 3 datasets from Kaggle site and compiled them to form a single dataset. The dataset has a dimension of 23,921 x 25. It comprises of all the international matches played from 1993-2022 showing the home and away team, the goals they scored, the goals they conceded, in which city/continent it was played, the tournament type like was it a friendly or an official tournament. It also displayed their FIFA rank, FIFA points and various parameters like gk, midfield, defense and attack score which helped in predicting the match outcomes. You could refer to **Table 1** to view the attributes and dimensions of the dataset used.



```
df.shape

(23921, 25)
```

Figure 1: Dimension of the dataset



	date	home_team	away_team	home_team_continent	away_team_continent	home_team_fifa_rank	away_team_fifa_rank	home_team_total_fifa_points	away_team_total_fifa_points	home_team_score
0	1993-08-08	Bolivia	Uruguay	South America	South America	59	22	0	0	3
1	1993-08-08	Brazil	Mexico	South America	North America	8	14	0	0	1
2	1993-08-08	Ecuador	Venezuela	South America	South America	35	94	0	0	5
3	1993-08-08	Guinea	Sierra Leone	Africa	Africa	65	86	0	0	1
4	1993-08-08	Paraguay	Argentina	South America	South America	67	5	0	0	1

5 rows × 25 columns

Table 1: The above two tables display the international matches that has been played from 1993-2022. It also displays the different attributes like the match date, home and away team's rank , their scores , total points and many more.

2.2 Data cleaning

We cleaned the data by handling missing values through imputation or removal, eliminating duplicates, and standardizing formats for consistency. Outliers were addressed by removal or capping, and irrelevant columns were dropped to reduce noise. Finally, numerical features were scaled to ensure compatibility with machine learning algorithms, resulting in a clean and reliable dataset for analysis.

2.3 Challenges faced

Before acquiring the structured dataset used in this project, we attempted web scraping to collect match-related data, which presented several significant challenges.

1. **Scattered Data Sources:** The data was not consolidated in a single location but was scattered across multiple websites, making it difficult to gather a complete and consistent dataset. Many links provided partial or incomplete information; for instance, one site might list the teams and scores but omit crucial details like match type or location, necessitating cross-referencing multiple sources to fill in the gaps.

2. **Inconsistent Website Structure:** Many websites lacked clear or consistent HTML structures with proper div or class identifiers, making it harder to parse data accurately. This added complexity to the web scraping scripts, increasing the likelihood of errors.

3. **Time-Consuming Process:** Scraping the required data was not only resource-intensive but also time-consuming. Writing scripts to extract data from various sources, cleaning and standardizing it, and then merging it into a usable format involved numerous steps and lines of code, many of which felt redundant.

4. **Technical Limitations:** Using tools like Selenium for scraping presented additional hurdles. While Selenium is powerful for dynamic content, its bot-like behavior often led to websites blocking access, requiring us to repeatedly manage sessions and use proxies. Furthermore, Selenium's high CPU usage added strain to our systems, slowing down the process. BeautifulSoup, on the other hand, was slower and consumed significant memory, limiting its effectiveness for large-scale scraping tasks.

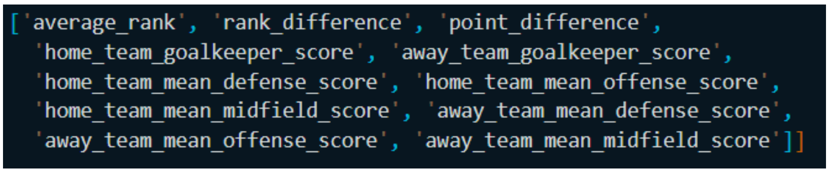
5. **Manual Interventions:** Due to the scattered and incomplete nature of the data, we had to manually handle several aspects of the scraping process, such as making HTTP requests, managing sessions, and reformatting the data to make it usable. This added extra overhead to the already tedious process.

Chapter 3: Selection of algorithms

3.1 Algorithms we used

→Logistic regression

we used **Logistic Regression** as one of the foundational models. Logistic regression is effective for predicting binary outcomes, in this case, whether a team will win a match or not. The model uses input features such as average rank, rank difference, point difference, home team goalkeeper score, away team goalkeeper score, and various performance scores (defense, offense, midfield) for both the home and away teams. These features directly impact the target variable, is_won, representing match outcomes. We fitted the logistic regression model on the training data using a **Pipeline**, which allowed us to manage the entire process of transforming features and training the model in an organized and systematic manner.



```
['average_rank', 'rank_difference', 'point_difference',
  'home_team_goalkeeper_score', 'away_team_goalkeeper_score',
  'home_team_mean_defense_score', 'home_team_mean_offense_score',
  'home_team_mean_midfield_score', 'away_team_mean_defense_score',
  'away_team_mean_offense_score', 'away_team_mean_midfield_score']]
```

Figure 2: All the input features that is responsible for influencing the outcome of a football match.

→polynomial features

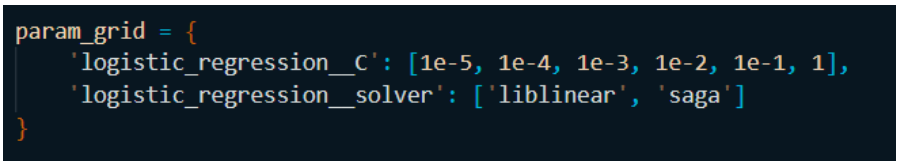
To enhance the logistic regression model's ability to learn more complex relationships between features, we introduced Polynomial Features with a degree of 2. This transformation creates new features that capture interaction terms and higher-order terms, allowing the model to account for non-linear relationships in the data. This feature transformation significantly improved the model's predictive power, especially when dealing with patterns that were not linear.

→pipeline

The Pipeline we used in the workflow combined both polynomial feature generation and logistic regression into a seamless process, ensuring all steps happened in the correct order. This prevented the model from being trained on the entire dataset, which would otherwise lead to data leakage. It also ensured that only the training set was used during the fitting phase, leading to a more reliable model evaluation.

→logistic regression with hyperparameter tunning

We also performed Hyperparameter Tuning to find the optimal settings for the logistic regression model. Hyperparameters such as the regularization strength (C) and the solver type (either loglinear or saga) were adjusted to improve model performance. The loglinear solver is well-suited for small datasets and binary classification tasks, while saga is better for handling large datasets and more complex problems. By tuning these parameters, we ensured the model avoided underfitting (too simple) or overfitting (too complex), leading to better generalization on unseen data.



```
param_grid = {
    'logistic_regression__C': [1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1],
    'logistic_regression__solver': ['liblinear', 'saga']
}
```

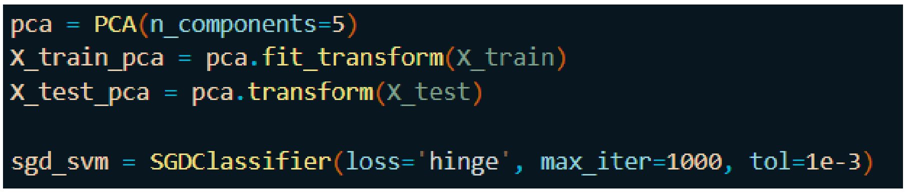
Figure 3: we adjusted the regularization strength and the algorithm solver to improve the model's performance.

→random forest classifier

The Random Forest Classifier is an ensemble learning technique that builds multiple decision trees during training and outputs the mode (most common) class of the individual trees. Each tree in the forest is trained on a random subset of the data with bootstrapping, and features are randomly selected for each split. This randomness helps reduce the variance of the model, making it more robust and less prone to overfitting compared to a single decision tree. We used the Random Forest model because it can handle both classification and regression tasks efficiently and works well with complex, high-dimensional datasets. It's particularly useful for datasets with multiple features and interactions between them. Random forests also provide feature importance scores, which help us identify the most influential factors in predicting the outcome of a football match.

→ Stochastic Gradient Descent – support vector machine

The Stochastic Gradient Descent (SGD) combined with Support Vector Machine (SVM) is a powerful machine learning algorithm used for classification tasks. SVM aims to find the hyperplane that best separates the classes in the feature space, with the largest margin between them. SGD is an optimization method used to update the model's parameters iteratively, making it highly efficient for large datasets. We chose the SGD SVM because it is suitable for high-dimensional data, which is often the case when dealing with multiple features, such as team performance scores and rankings. SVMs are known for their ability to find complex boundaries in data, making them effective for identifying patterns in match outcomes that may not be linearly separable. The model can be further optimized with the choice of kernel functions (e.g., linear, polynomial) and hyperparameters.



```
pca = PCA(n_components=5)
X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.transform(X_test)

sgd_svm = SGDClassifier(loss='hinge', max_iter=1000, tol=1e-3)
```

Figure 4: use of PCA for dimension reduction (5 components/features)

→ Gradient Boosting Classifier

The Gradient Boosting Classifier (GBC) is another ensemble learning method, but unlike Random Forest, it builds trees sequentially, where each tree corrects the errors of the previous one. It uses gradient descent to minimize the loss function, iteratively fitting new trees to the residual errors of the current model. The Gradient Boosting model is highly effective for complex datasets where other models may struggle to capture non-linear relationships. We used Gradient Boosting because it often produces highly accurate results by combining weak learners into a strong model. This technique is particularly useful in predicting match outcomes where various factors interact in non-linear ways. GBC can also be fine-tuned for performance by adjusting parameters such as the number of estimators, learning rate, and maximum depth of trees. It is known for its high predictive accuracy and can handle both classification and regression problems efficiently.

We split the dataset into **training** and **test** sets using a **Train-Test Split**, where 40% of the data was reserved for testing. This division allowed us to evaluate the model’s performance on unseen data, ensuring its ability to generalize beyond the training set.

Finally, we used the **ROC Curve** and **AUC Score** to assess the model's performance. The ROC curve plots the true positive rate against the false positive rate at various threshold settings, while the AUC score provides a quantitative measure of the model's ability to discriminate between positive and negative outcomes. A score of 0.5 suggests no discrimination, a score between 0.7 and 0.8 is considered acceptable, and a score between 0.8 and 0.9 is excellent. These metrics were critical in understanding the effectiveness of the logistic regression model in predicting match outcomes.

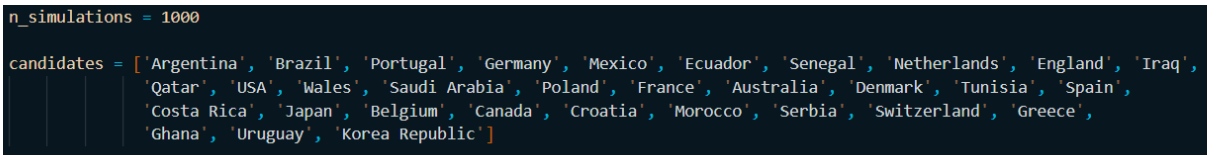
3.2 Challenges faced

- The dataset contained many features, leading to potential overfitting in models like Random Forest and Gradient Boosting.
- Reducing dimensionality without losing important information proved difficult.
- Models like Random Forest and Gradient Boosting were prone to overfitting, particularly with a large number of trees or deep trees.
- Achieving the right balance between underfitting and overfitting was challenging.
- Hyperparameter tuning for algorithms like Gradient Boosting and Logistic Regression was computationally expensive and time-consuming.
- Training models like Random Forest and Gradient Boosting took a long time, especially with larger datasets.
- The need for faster training times led to challenges in optimization.
- The dataset was imbalanced, leading to biased predictions in models.
- Handling class imbalance effectively while maintaining performance was a significant challenge.
- Models like Random Forest and Gradient Boosting were difficult to interpret and explain, especially for feature importance and prediction results.
- Determining the right evaluation metrics was crucial, and balancing between accuracy and model robustness was challenging.
- Ensuring models could generalize well across different datasets and thresholds was difficult.

Chapter 4: Simulation

We performed a Monte Carlo simulation to predict the outcomes of a football tournament, simulating each stage from the Round of 16 to the final. The simulation runs for 1000 iterations, generating different potential outcomes for each match based on a series of factors influencing the match results.

The tournament involves 32 teams, with each stage progressively narrowing the field. The simulation begins with the Round of 16, where 16 pairs of teams compete against each other. In each match, the winner is determined based on a probability model (best_model), which uses several input features such as rank difference, point difference, and team-specific scores like goalkeeper score, defense score, offense score, and midfield score. These features help predict the likelihood of a home team winning against an away team. A random binomial distribution is used to simulate the actual outcome based on the predicted probability.



```
n_simulations = 1000

candidates = ['Argentina', 'Brazil', 'Portugal', 'Germany', 'Mexico', 'Ecuador', 'Senegal', 'Netherlands', 'England', 'Iraq',
              'Qatar', 'USA', 'Wales', 'Saudi Arabia', 'Poland', 'France', 'Australia', 'Denmark', 'Tunisia', 'Spain',
              'Costa Rica', 'Japan', 'Belgium', 'Canada', 'Croatia', 'Morocco', 'Serbia', 'Switzerland', 'Greece',
              'Ghana', 'Uruguay', 'Korea Republic']
```

Figure 5: we ran the simulation for 1000 times and also we can see all the 32 participating teams

As teams win and advance through the tournament, the Quarterfinal, Semifinal, and Final stages follow. For each of these stages, the winning teams from the previous round play against each other, with the process repeating similarly. Each match's outcome is simulated using the same model, ensuring consistency throughout the tournament simulation.

The results from each stage, including the winners and their corresponding probabilities, are collected and stored. This data is captured in DataFrames, which are created for each stage of the tournament (Round of 16, Quarterfinal, Semifinal, and Final). The data includes columns for the stage, the winning team, and the probability of that team winning, providing a detailed record of each simulated match outcome.

After completing all 1000 simulations, the results are consolidated into four final DataFrames, representing the outcomes at each stage of the tournament. These results can be analyzed to determine which teams are most likely to progress through each round and potentially win the entire tournament.

The key challenge of this simulation lies in the accuracy of the input data. The predictions are highly dependent on the quality of the team rankings and performance metrics used to train the model. If the data is not accurate or up-to-date, the simulation results may not be reliable. Additionally, the performance of the model itself can affect the outcomes; if the model is overfitted or underfitted, it may lead to inaccurate predictions.

Chapter 5: UI

The **Tournament Simulation UI** enables users to simulate a football tournament by uploading a CSV file containing team data. The file should have specific columns like 'home_team_fifa_rank', 'away_team_fifa_rank', 'home_team', and 'away_team'. The app checks for these required columns before proceeding. Once the dataset is validated, users can select 32 teams from the available teams in the dataset to participate in the tournament. The app then calculates each team's win probability based on their FIFA rankings and normalizes these probabilities for home and away games.

After the teams are selected and win probabilities are calculated, the app runs the tournament simulation. The simulation involves grouping the teams into four groups, followed by knockout stages (Round of 16, Quarter-finals, Semi-finals, and Final). For each match, the winner is determined by a random process weighted by the calculated win probabilities. After running the specified number of simulations, the results are displayed, including the team that won the most simulations. The app also provides a summary of the selected teams with updated win probabilities.

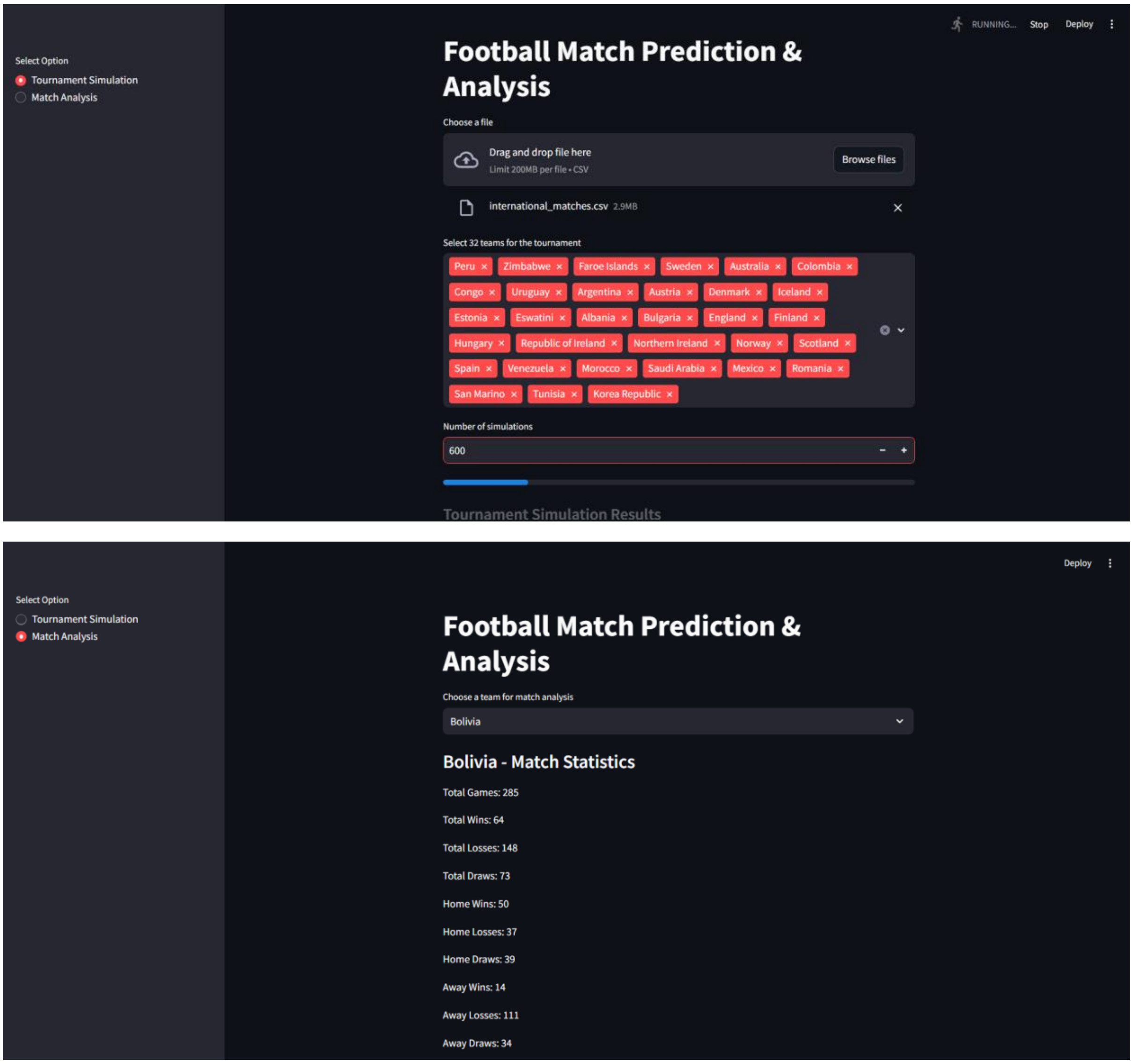
Match Analysis UI

The **Match Analysis UI** allows users to analyze the performance statistics of any team based on historical match data. The data is loaded from a CSV file containing information about various football matches, including home team results, away team results, and performance metrics like midfield, defense, and offense scores. Users can select a team from a dropdown menu, and the app will calculate and display several statistics related to the team's performance, such as the total number of games played, wins, losses, draws, home/away performance, FIFA ranking, and specific scores for midfield, defense, and offense.

In addition to the statistics, the app visualizes the team's performance through a pie chart that shows the percentage distribution of wins, losses, and draws. This feature helps users gain insights into a team's overall performance and their success rates in different types of matches.

Main Streamlit App Structure

The **main function** of the app offers two main options through a sidebar: **Tournament Simulation** and **Match Analysis**. Depending on the user's choice, the app either directs them to the tournament simulation feature or the match analysis feature. This streamlined navigation ensures that users can easily access the functionality they need without being overwhelmed by multiple options on the main screen.



Select Option
● Tournament Simulation
○ Match Analysis

Football Match Prediction & Analysis

Choose a file

Drag and drop file here
Limit 200MB per file • CSV

international_matches.csv 2.9MB

Select 32 teams for the tournament

Peru × Zimbabwe × Faroe Islands × Sweden × Australia × Colombia × Congo × Uruguay × Argentina × Austria × Denmark × Iceland × Estonia × Eswatini × Albania × Bulgaria × England × Finland × Hungary × Republic of Ireland × Northern Ireland × Norway × Scotland × Spain × Venezuela × Morocco × Saudi Arabia × Mexico × Romania × San Marino × Tunisia × Korea Republic ×

Number of simulations
600

Tournament Simulation Results

Select Option
○ Tournament Simulation
● Match Analysis

Football Match Prediction & Analysis

Choose a team for match analysis
Bolivia

Bolivia - Match Statistics

Total Games: 285

Total Wins: 64

Total Losses: 148

Total Draws: 73

Home Wins: 50

Home Losses: 37

Home Draws: 39

Away Wins: 14

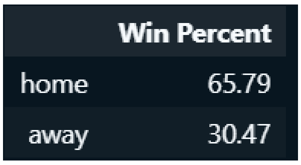
Away Losses: 111

Away Draws: 34

Fig 6: Displaying the UI

Chapter 6: Results and Discussions

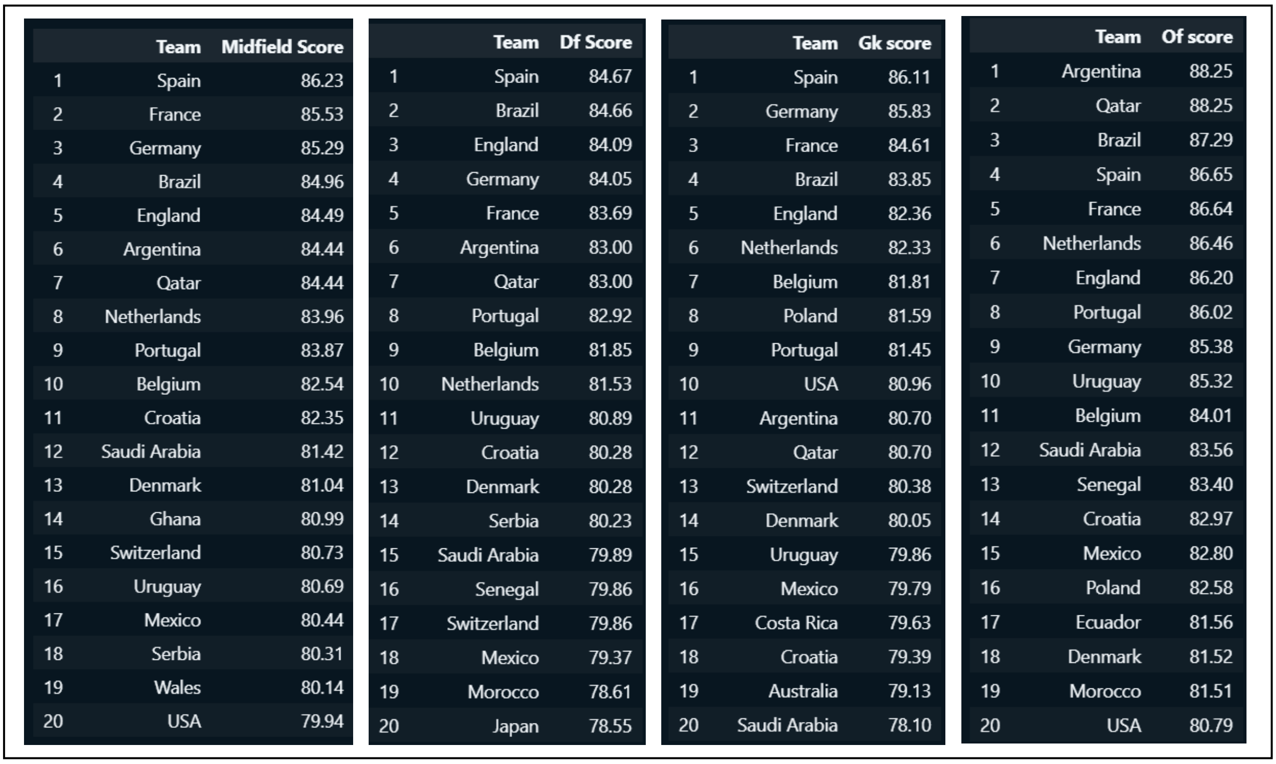
As we made our analysis, we came to know that the home team has the most chances of winning when the match is being played at their home ground with a win percentage of 65.79%, the away team won only 30.47 % of the time.



	Win Percent
home	65.79
away	30.47

Fig 7: Displaying the win %age of home and away teams

We came to know that Spain is one of the best teams as they have the highest midfield, defense and gk scores. Argentina with the offense score of 88.25 is the best attacking team.



	Team	Midfield Score
1	Spain	86.23
2	France	85.53
3	Germany	85.29
4	Brazil	84.96
5	England	84.49
6	Argentina	84.44
7	Qatar	84.44
8	Netherlands	83.96
9	Portugal	83.87
10	Belgium	82.54
11	Croatia	82.35
12	Saudi Arabia	81.42
13	Denmark	81.04
14	Ghana	80.99
15	Switzerland	80.73
16	Uruguay	80.69
17	Mexico	80.44
18	Serbia	80.31
19	Wales	80.14
20	USA	79.94

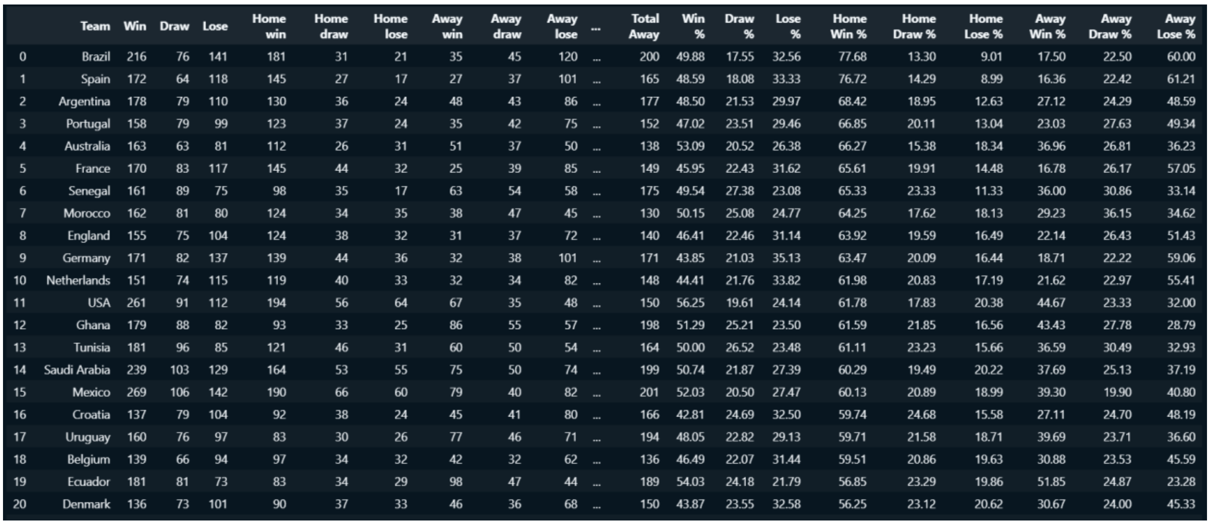
	Team	Df Score
1	Spain	84.67
2	Brazil	84.66
3	England	84.09
4	Germany	84.05
5	France	83.69
6	Argentina	83.00
7	Qatar	83.00
8	Portugal	82.92
9	Belgium	81.85
10	Netherlands	81.53
11	Uruguay	80.89
12	Croatia	80.28
13	Denmark	80.28
14	Serbia	80.23
15	Saudi Arabia	79.89
16	Senegal	79.86
17	Switzerland	79.86
18	Mexico	79.37
19	Morocco	78.61
20	Japan	78.55

	Team	Gk score
1	Spain	86.11
2	Germany	85.83
3	France	84.61
4	Brazil	83.85
5	England	82.36
6	Netherlands	82.33
7	Belgium	81.81
8	Poland	81.59
9	Portugal	81.45
10	USA	80.96
11	Argentina	80.70
12	Qatar	80.70
13	Switzerland	80.38
14	Denmark	80.05
15	Uruguay	79.86
16	Mexico	79.79
17	Costa Rica	79.63
18	Croatia	79.39
19	Australia	79.13
20	Saudi Arabia	78.10

	Team	Of score
1	Argentina	88.25
2	Qatar	88.25
3	Brazil	87.29
4	Spain	86.65
5	France	86.64
6	Netherlands	86.46
7	England	86.20
8	Portugal	86.02
9	Germany	85.38
10	Uruguay	85.32
11	Belgium	84.01
12	Saudi Arabia	83.56
13	Senegal	83.40
14	Croatia	82.97
15	Mexico	82.80
16	Poland	82.58
17	Ecuador	81.56
18	Denmark	81.52
19	Morocco	81.51
20	USA	80.79

Table 2: Displaying top 20 teams with their midfield, defense, gk and attacking score respectively.

We came to know that Brazil, Spain, Argentina, Portugal, France are the teams with the most home win rates. Teams like Brazil, Spain, and Argentina demonstrate strong overall win percentages, with Brazil having a notable home win percentage of 77.68%, reflecting a significant home advantage. Meanwhile, the data also reveals how teams like Tunisia and Saudi Arabia perform under different conditions, showing variations between home and away matches. This information is valuable for analyzing team strengths, understanding home versus away dynamics, and making predictions.



	Team	Win	Draw	Lose	Home win	Home draw	Home lose	Away win	Away draw	Away lose	...	Total Away	Win %	Draw %	Lose %	Home Win %	Home Draw %	Home Lose %	Away Win %	Away Draw %	Away Lose %
0	Brazil	216	76	141	181	31	21	35	45	120	...	200	49.88	17.55	32.56	77.68	13.30	9.01	17.50	22.50	60.00
1	Spain	172	64	118	145	27	17	27	37	101	...	165	48.59	18.08	33.33	76.72	14.29	8.99	16.36	22.42	61.21
2	Argentina	178	79	110	130	36	24	48	43	86	...	177	48.50	21.53	29.97	68.42	18.95	12.63	27.12	24.29	48.59
3	Portugal	158	79	99	123	37	24	35	42	75	...	152	47.02	23.51	29.46	66.85	20.11	13.04	23.03	27.63	49.34
4	Australia	163	63	81	112	26	31	51	37	50	...	138	53.09	20.52	26.38	66.27	15.38	18.34	36.96	26.81	36.23
5	France	170	83	117	145	44	32	25	39	85	...	149	45.95	22.43	31.62	65.61	19.91	14.48	16.78	26.17	57.05
6	Senegal	161	89	75	98	35	17	63	54	58	...	175	49.54	27.38	23.08	65.33	23.33	11.33	36.00	30.86	33.14
7	Morocco	162	81	80	124	34	35	38	47	45	...	130	50.15	25.08	24.77	64.25	17.62	18.13	29.23	36.15	34.62
8	England	155	75	104	124	38	32	31	37	72	...	140	46.41	22.46	31.14	63.92	19.59	16.49	22.14	26.43	51.43
9	Germany	171	82	137	139	44	36	32	38	101	...	171	43.85	21.03	35.13	63.47	20.09	16.44	18.71	22.22	59.06
10	Netherlands	151	74	115	119	40	33	32	34	82	...	148	44.41	21.76	33.82	61.98	20.83	17.19	21.62	22.97	55.41
11	USA	261	91	112	194	56	64	67	35	48	...	150	56.25	19.61	24.14	61.78	17.83	20.38	44.67	23.33	32.00
12	Ghana	179	88	82	93	33	25	86	55	57	...	198	51.29	25.21	23.50	61.59	21.85	16.56	43.43	27.78	28.79
13	Tunisia	181	96	85	121	46	31	60	50	54	...	164	50.00	26.52	23.48	61.11	23.23	15.66	36.59	30.49	32.93
14	Saudi Arabia	239	103	129	164	53	55	75	50	74	...	199	50.74	21.87	27.39	60.29	19.49	20.22	37.69	25.13	37.19
15	Mexico	269	106	142	190	66	60	79	40	82	...	201	52.03	20.50	27.47	60.13	20.89	18.99	39.30	19.90	40.80
16	Croatia	137	79	104	92	38	24	45	41	80	...	166	42.81	24.69	32.50	59.74	24.68	15.58	27.11	24.70	48.19
17	Uruguay	160	76	97	83	30	26	77	46	71	...	194	48.05	22.82	29.13	59.71	21.58	18.71	39.69	23.71	36.60
18	Belgium	139	66	94	97	34	32	42	32	62	...	136	46.49	22.07	31.44	59.51	20.86	19.63	30.88	23.53	45.59
19	Ecuador	181	81	73	83	34	29	98	47	44	...	189	54.03	24.18	21.79	56.85	23.29	19.86	51.85	24.87	23.28
20	Denmark	136	73	101	90	37	33	46	36	68	...	150	43.87	23.55	32.58	56.25	23.12	20.62	30.67	24.00	45.33

Table 3: Displaying top 20 teams with the number of wins, draw and loss

The performance comparison of various machine learning models shows some nuanced differences in their ability to classify accurately, as measured by accuracy and AUC (Area Under the ROC Curve). Gradient Boosting stands out slightly, achieving the highest accuracy at 68.62% and an AUC of 0.75, indicating it performs marginally better in correctly classifying data points. The Logistic Regression models, both standard and tuned, follow closely with accuracies of 68.19% and 68.06%, respectively, and both achieve an AUC of 0.75. This suggests that while these models are relatively straightforward, they provide robust classification capabilities comparable to Gradient Boosting.

The Random Forest model, however, lags a bit behind with an accuracy of 65.16% and an AUC of 0.70, indicating it might be less effective at distinguishing between classes in this dataset. The SGD SVM with PCA transformation achieves a slightly better accuracy than Random Forest at 67.15% and an AUC of 0.72, showing that while it benefits from dimensionality reduction, its performance still falls short of the Logistic Regression and Gradient Boosting models.

Overall, the differences in performance are minor, with most models reaching similar accuracy and AUC values, particularly the top three models (Gradient Boosting and the two Logistic Regression models). This similarity in results might imply that the dataset or feature set has a certain degree of complexity that these models are approaching similarly. To achieve a more substantial improvement, further feature engineering, trying other model architectures, or exploring ensemble techniques may be necessary to capture additional patterns within the data.

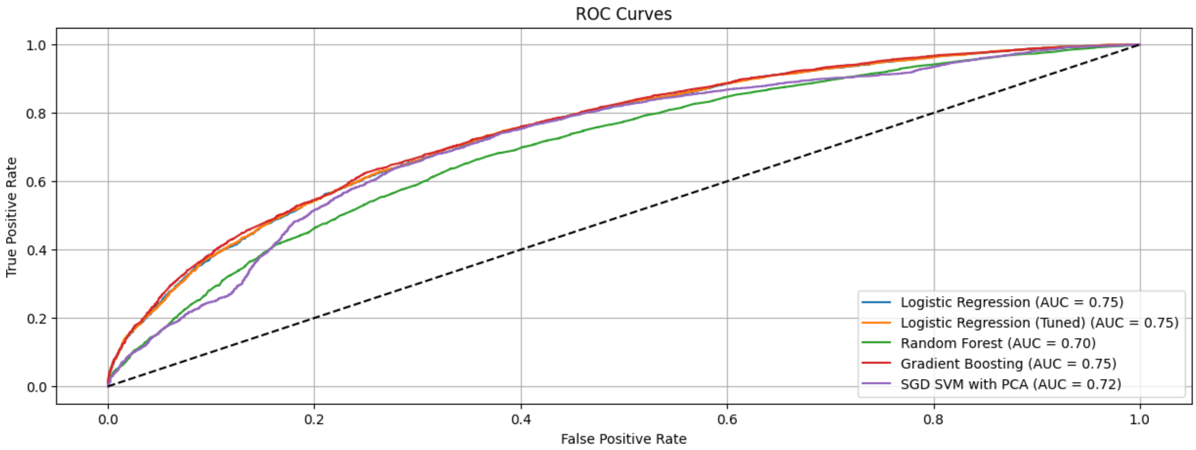
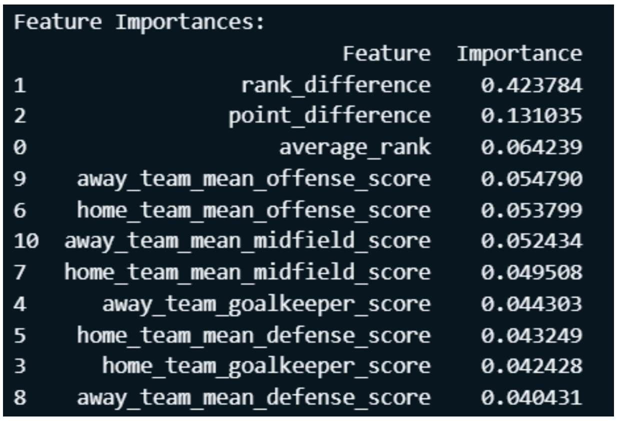


Fig 8: displaying the AUC of all the different models that we trained



```
Feature Importances:
                        Feature  Importance
1               rank_difference    0.423784
2              point_difference    0.131035
0                  average_rank    0.064239
9     away_team_mean_offense_score  0.054790
6     home_team_mean_offense_score  0.053799
10   away_team_mean_midfield_score  0.052434
7    home_team_mean_midfield_score  0.049508
4         away_team_goalkeeper_score  0.044303
5     home_team_mean_defense_score  0.043249
3         home_team_goalkeeper_score  0.042428
8     away_team_mean_defense_score  0.040431
```

Fig 9: displaying the important features that help in making contribution to the model's prediction.

Higher importance means the feature has a stronger influence on the model's decision-making process.

The feature importance values indicate which factors most significantly influence the model's predictions. The top three features are rank_difference (0.4238), point_difference (0.1310), and average_rank (0.0642), making them the primary drivers of the model's performance. These features reflect the competitive gap between teams, with rank_difference being the most crucial. Secondary features, such as offense scores, midfield scores, and goalkeeper scores for both home and away teams, have lower but meaningful importance. This analysis highlights that team rankings and point differences are far more impactful than individual player performance metrics in determining match outcomes.

The bar charts illustrate the progression of teams through the stages of the 2022 World Cup, highlighting their performance consistency across rounds. In the "Road to Quarterfinals" chart (top left), USA and England have the highest counts, indicating a strong presence up to this stage, followed by teams like Belgium and Netherlands. Moving to the "Quarterfinal World Cup 2022" chart (top right), Netherlands leads with the highest count, closely followed by

Mexico, Germany, Argentina, and Brazil, showing these teams' strength in reaching the Quarterfinals. In the Semifinals (bottom left), only four teams—Germany, Argentina, Brazil, and Portugal—remain, all with similar counts, demonstrating their competitive edge. Finally, in the "Winner World Cup 2022" chart (bottom right), only Brazil and Argentina appear with equal counts, suggesting they were the leading contenders for the championship. This progression analysis underscores the consistent performance of certain teams across stages, particularly Argentina and Brazil.

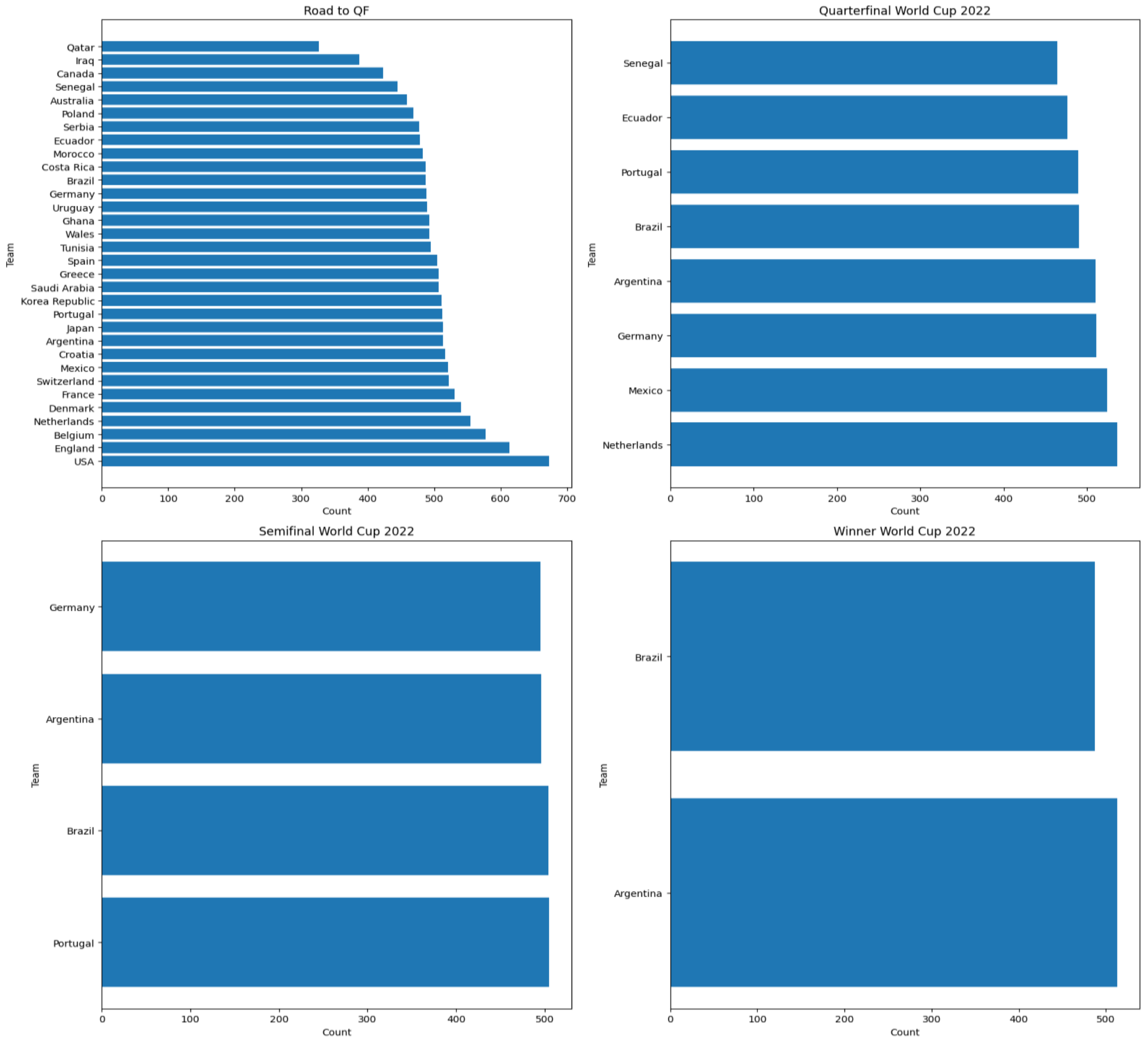


Fig 10: output of the simulation showing the results of various stages of the tournament i.e. Road to QF, QF, SEMI and Final

Conclusion

The analysis of the football match data reveals several key insights. Brazil stands out as one of the strongest teams, ranking highly in both offensive and defensive metrics, with significant home advantage. Teams like Spain, France, and Argentina also demonstrate strong performances, particularly in defense and overall win percentages. Spain excels in goalkeeper scores, reflecting their solid defensive foundation. The data further indicates that home teams generally perform better than away teams, with Brazil and Spain showing high home win percentages. Additionally, Argentina and Qatar excel in offensive capabilities, while teams like Serbia show remarkable defensive efficiency, with a low "conceded goals per goalkeeper score" ratio. The win streaks of top teams like Spain and Brazil reflect their consistency and resilience over time. Overall, the analysis emphasizes the importance of both offensive strength and defensive stability, alongside the significant impact of playing at home in football outcomes.

The simulation results for the 2022 FIFA World Cup indicate that Argentina emerged as the winner, showcasing the strongest performance throughout the tournament. Brazil was a close contender, consistently advancing through the stages. In the Road to Quarterfinals, teams like USA, England, and Netherlands performed notably well, while Senegal, Ecuador, and Portugal led in the Quarterfinals, with Brazil, Argentina, Germany, Mexico, and Netherlands also progressing strongly. In the Semifinals, Germany, Brazil, Argentina, and Portugal were the top contenders, with Germany showing a particularly strong presence. Ultimately, Brazil and Argentina stood out as the most dominant teams, with Argentina clinching the title in the final simulation, reflecting their overall superiority in the competition.

References

1. https://machinelearningmastery.com/
2. Stack Overflow - Where Developers Learn, Share, & Build Careers
3. Machine Learning | Google for Developers
4. What Is Monte Carlo Simulation? | IBM
5. An Introduction and Step-by-Step Guide to Monte Carlo Simulations | by Benjamin Huser-Berta | Medium

For datasets

1. www.kaggle.com/datasets/martj42/international-football-results-from-1872-to-2017/data?select=goalscorers.csv
2. www.kaggle.com/datasets/martj42/international-football-results-from-1872-to-2017/data?select=results.csv
3. www.kaggle.com/datasets/martj42/international-football-results-from-1872-to-2017/data?select=shootouts.csv