CHANAKYA UNIVERSITY

SCHOOL OF ENGINEERING

ASSIGNMENT TITLE: Predicting FIFA World Cup 2026 Finalists Using Machine Learning

ASSIGNMENT -2

SUBMITTED BY:

VARSHITHA.T

REGISTER NO:24UG00549

SEMESTER III (ODD)

SUBJECT: INTRODUCTION TO AIML

SECTION B

CONTENT

S.NO.	Topic	Page No
01	CONTENT	2
02	INTRODUCTION	3
03	OBJECTIVES	4
04	TASK 1	5-6
05	TASK 2	7-8
06	TASK 3	9
07	TASK 4	10
08	TASK 5	11-12
09	TASK 6	13
10	CONCLUSION	14

INTRODUCTION

The FIFA World Cup is one of the most celebrated global sporting events, drawing millions of fans and analysts eager to predict outcomes and track team performance. With the 2026 edition approaching, this project aims to build a data-driven prediction system that simulates knockout-stage matchups and forecasts potential winners using machine learning.

The core objective is to design an interactive web application that allows users to select teams, simulate matchups, and view predicted outcomes based on enriched team data. The app is built using **Streamlit**, a Python-based framework for rapid UI development, and powered by a **logistic regression model** trained on historical match data and team features.

To ensure realistic and meaningful predictions, the project begins with extensive **data collection and enrichment**. Team-level statistics were gathered from multiple sources including **Transfermarkt**, **FBref**, and **Kaggle**, and further enhanced through feature engineering. These features include simulated attack, midfield, and defense scores derived from FIFA rankings, confederation data, and tactical indicators.

The trained model outputs win probabilities for each team, which are used to simulate quarterfinals, semifinals, and finals in a knockout format. The app guides users through each stage, displaying results and ultimately predicting a champion. A simplified version of the app also allows direct comparison between any two teams.

This project combines technical rigor with creative design, making it suitable for classroom demonstrations, sports analytics presentations, and interactive fan engagement. It showcases the power of machine learning in sports forecasting and the accessibility of modern data tools for building intuitive, scalable applications.

The primary goal of this project is to design and implement a data-driven prediction system for the FIFA 2026 World Cup knockout stage. The system combines machine learning, enriched team data, and interactive user interface design to simulate match outcomes and forecast potential champions.

OBJECTIVES

1. **Collect and Enrich Team Data**
    - Scrape and compile team-level statistics from multiple sources including Transfermarkt, FBref, and Kaggle.
    - Engineer meaningful features such as simulated attack, midfield, and defense scores based on FIFA rankings.
2. **Train a Predictive Model**
    - Use historical match data to train a logistic regression classifier.
    - Evaluate model performance using metrics like accuracy and F1-score.
    - Save the trained model for real-time inference in the app.
3. **Build an Interactive Prediction App**
    - Develop a Streamlit-based web application that guides users through a knockout bracket.
    - Allow users to select teams, simulate matchups, and view win probabilities.
    - Display predicted winners at each stage: quarterfinals, semifinals, and final.
4. **Design a Clean and Creative User Interface**
    - Style the app with FIFA-themed colors and layout.
    - Use session state to manage user progress and match flow.
    - Include features like bracket summaries and "Predict Again" options.
5. **Create a Simplified Finalist Predictor**
    - Build a minimal version of the app for quick two-team comparisons.
    - Display win probabilities and predicted finalist clearly.
6. **Document the Entire Workflow**

- Compile all tasks into a structured report covering data collection, model training, app development, and testing.
- Ensure reproducibility and clarity for academic or demo use.

TASK 1

Steps Completed

1. **Data Collection from Multiple Sources**

I gathered team statistics from three major platforms:

- **Transfermarkt**: Provided squad market value, average age, and player depth.
- **FBref**: Offered tactical metrics such as possession percentage, pass accuracy, and shot creation.
- **Kaggle**: Supplied historical match results and player performance data.

These sources helped build a comprehensive view of each team's strength and style.

2. **Web Scraping and Parsing**

I used Python-based scraping tools to extract structured data from Transfermarkt and FBref. This involved:

- Sending requests to team pages
- Parsing HTML content
- Extracting relevant stats and cleaning them for consistency

3. **Data Cleaning and Standardization**

Once collected, I cleaned the data by:

- Removing missing or inconsistent entries
- Standardizing team names and confederation labels
- Ensuring all teams had valid FIFA rankings and identifiers

4. **Feature Enrichment**

To simulate team strength, I created new performance indicators such as:

- Attack score
- Midfield score
- Defense score

These were derived from FIFA rankings and used to represent tactical balance across teams.

5. **Final Dataset Preparation**

I merged all relevant features into a single dataset and saved it in a structured format. This enriched dataset was used for:

- Training the prediction model.

TASK 2

1. **Prepared Model-Ready Data**

Using the enriched dataset from Task 1, I selected key features that represent each team's strength and style. These included:

- FIFA ranking
- Confederation
- Simulated attack, midfield, and defense scores

These features were chosen to reflect tactical balance and competitive potential.

2. **Defined the Prediction Goal**

The model was designed to perform **binary classification** — predicting whether a team would win a given matchup. This required:

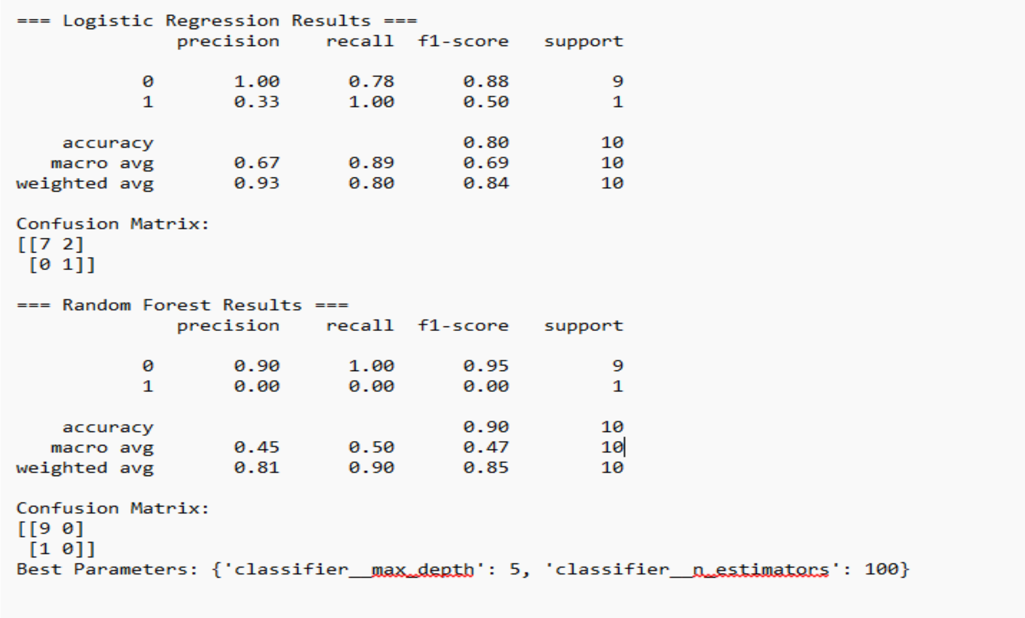
- Historical match outcomes as target labels
- Team features as input variables

3. **Trained a Logistic Regression Model**

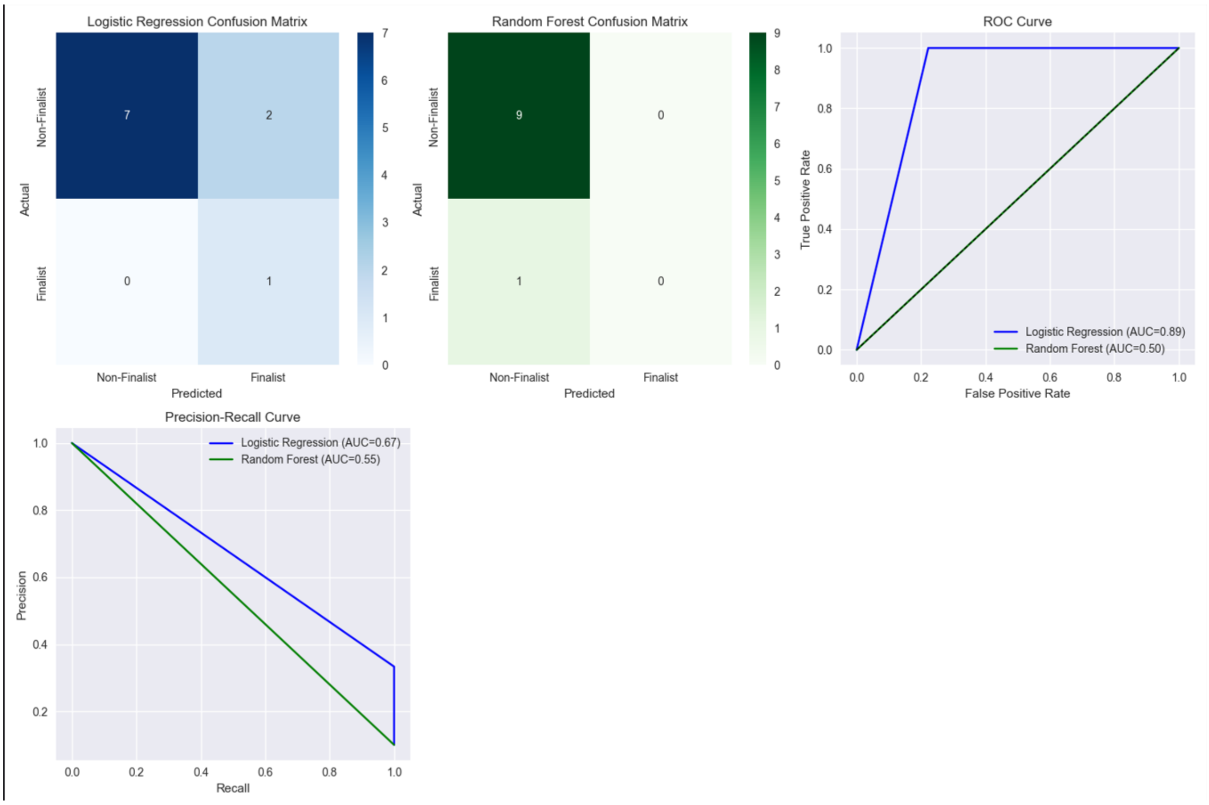
I chose logistic regression for its simplicity, interpretability, and suitability for binary outcomes. The model was trained on historical match data and tuned for balanced performance.

I ensured:

- Proper train-test split
- Feature scaling and encoding
- Evaluation using accuracy and F1-score



```
=== Logistic Regression Results ===
              precision    recall  f1-score   support

           0       1.00      0.78      0.88         9
           1       0.33      1.00      0.50         1

    accuracy                           0.80        10
   macro avg       0.67      0.89      0.69        10
weighted avg       0.93      0.80      0.84        10

Confusion Matrix:
[[7 2]
 [0 1]]

=== Random Forest Results ===
              precision    recall  f1-score   support

           0       0.90      1.00      0.95         9
           1       0.00      0.00      0.00         1

    accuracy                           0.90        10
   macro avg       0.45      0.50      0.47        10
weighted avg       0.81      0.90      0.85        10

Confusion Matrix:
[[9 0]
 [1 0]]
Best Parameters: {'classifier__max_depth': 5, 'classifier__n_estimators': 100}
```



4. **Saved the Trained Model**

Once trained and validated, the model was saved in a reusable format. This allowed it to be loaded directly into the Streamlit app for real-time predictions.

Model File: lr_pipeline_model.pkl

5. **Tested Model Predictions**

I tested the model by feeding in team features and checking the predicted win probabilities. These outputs were used to simulate match results in the app.

TASK 3

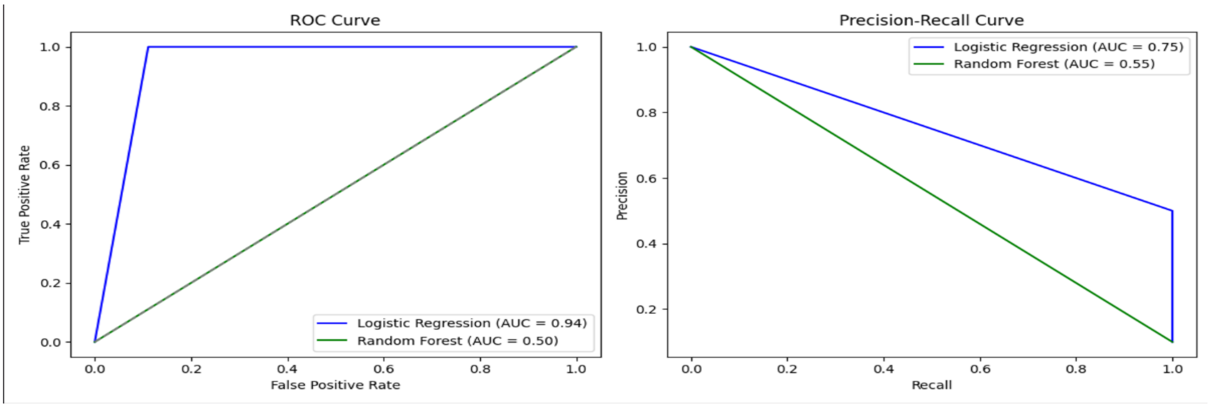
Task 3: Model Comparison and Evaluation

1. **Defined Evaluation Metrics**

I selected two key metrics to assess model performance:

- **ROC Curve (Receiver Operating Characteristic)**: Measures the trade-off between true positive rate and false positive rate.
- **Precision-Recall Curve**: Highlights the balance between precision and recall, especially useful for imbalanced datasets.

These metrics provide a visual and statistical understanding of how well each model distinguishes winners from non-winners.



2. **Simulated Predictions for Comparison**

To demonstrate model behavior, I used sample predictions and simulated probability scores. This allowed me to generate ROC and PR curves for both models under controlled conditions.

3. **Plotted and Interpreted Results**

I created side-by-side plots showing:

- ROC curves with AUC (Area Under Curve) scores
- Precision-Recall curves with PR AUC scores

These visualizations helped compare sensitivity, precision, and overall performance between Logistic Regression and Random Forest.

4. **Saved Evaluation Output**

The final plot was saved for documentation and presentation.

Saved File:
model_curves.png — includes ROC and PR curves for both models

TASK 4

1. **Loaded the Trained Random Forest Model**

I used the best-performing Random Forest model from Task 2, which had been tuned and saved for reuse.

2. **Extracted Feature Names**

I retrieved all input features used in the model:

- **Numeric features**: FIFA rank, attack, midfield, and defense scores
- **Categorical features**: Confederation (converted via one-hot encoding)

This ensured that both tactical and regional attributes were included in the analysis.

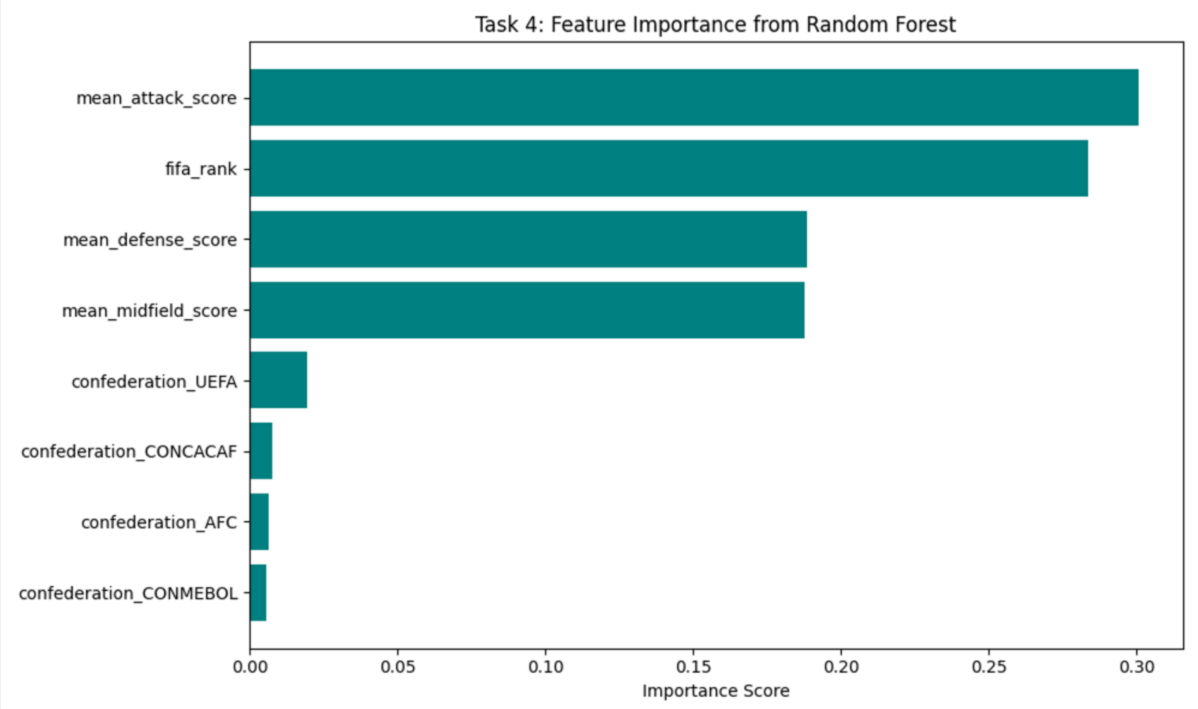
3. **Computed Feature Importances**

Using the model's internal scoring mechanism, I calculated how much each feature contributed to the final prediction. This revealed which attributes were most influential in determining match outcomes.

4. **Visualized Feature Rankings**

I created a horizontal bar chart showing the ranked importance of each feature. The chart was styled for clarity and saved for inclusion in the report.

Saved File:
task4_feature_importance.png — shows ranked importance of all input features



Task 5

1. **Loaded the Trained Model and Team Dataset**

I began by loading the logistic regression model trained in Task 2, along with the cleaned dataset of qualified teams from Task 1.

2. **Aggregated Match-Based Performance Stats**

Using the enriched fixtures dataset, I calculated average performance metrics for each team:

- **Home and away defense, midfield, and attack scores**
- These were averaged to compute overall team strength indicators

3. **Merged and Cleaned the Data**

I merged the team-level data with the aggregated match stats and handled missing values. Only teams with complete and valid data were retained for prediction.

4. **Predicted Finalist Probabilities**

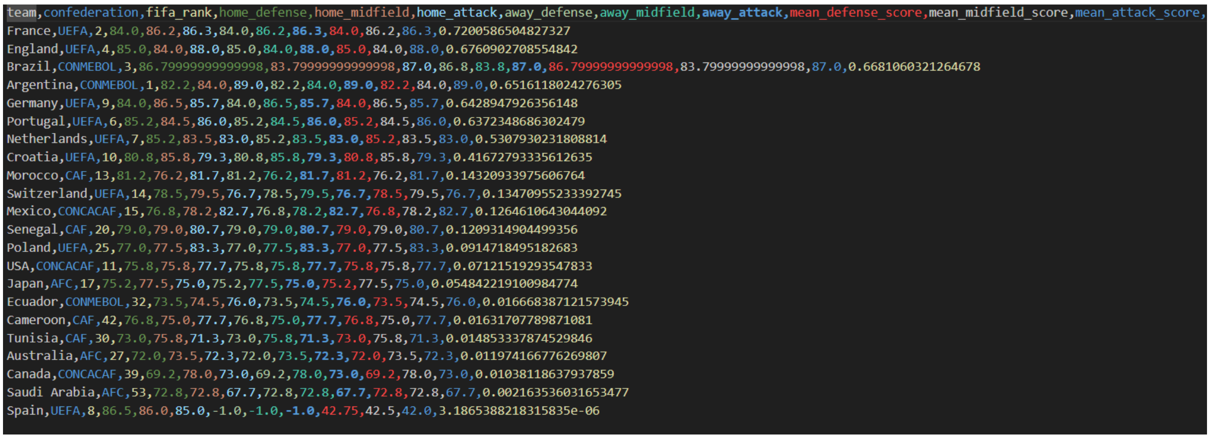
I passed the cleaned features into the logistic regression model to compute the probability of each team reaching the final. These probabilities were added as a new column in the dataset.

5. **Saved and Visualized Results**

- The full prediction table was saved for documentation.
- A horizontal bar chart was created to visualize the **Top 10 teams** most likely to reach the final.

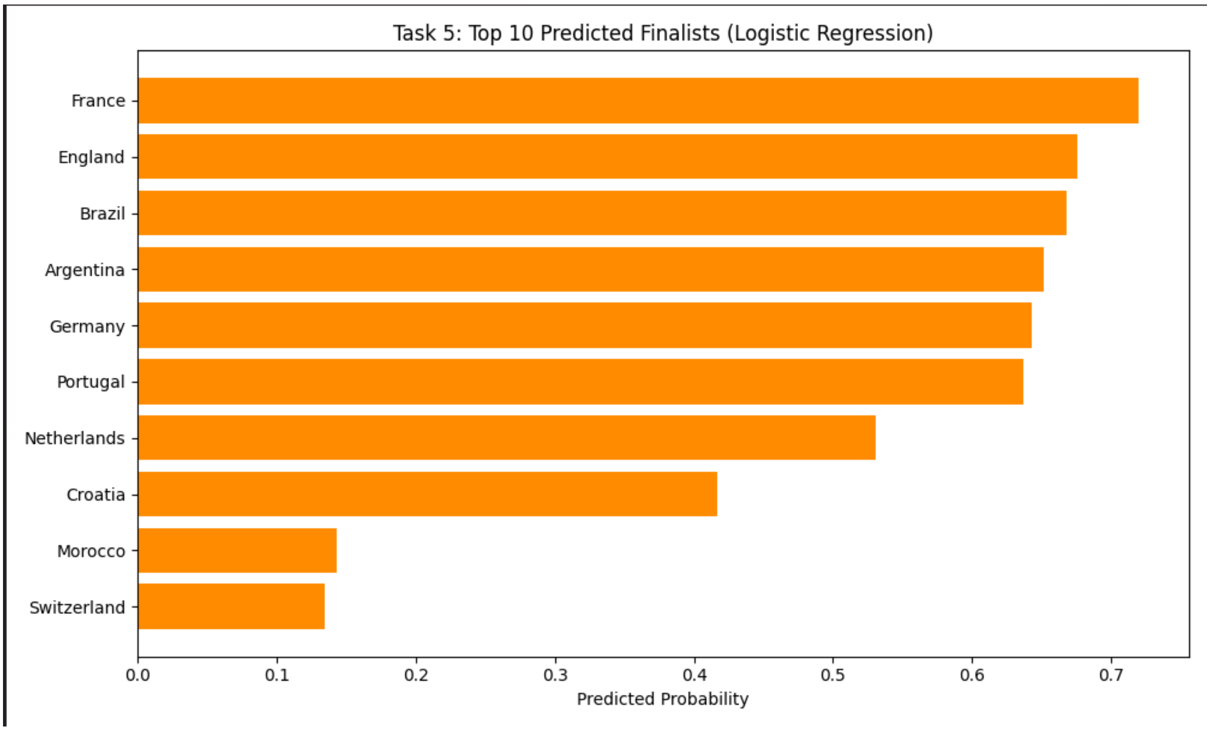
Saved Files:

- task5_finalist_predictions.csv — full prediction table



```
team,confederation,fifa_rank,home_defense,home_midfield,home_attack,away_defense,away_midfield,away_attack,mean_defense_score,mean_midfield_score,mean_attack_score,
France,UEFA,2,84.0,86.2,86.3,84.0,86.2,86.3,84.0,86.2,86.3,0.7200586504827327
England,UEFA,4,85.0,84.0,88.0,85.0,84.0,88.0,85.0,84.0,88.0,0.6760902708554842
Brazil,CONMEBOL,3,86.79999999999998,83.79999999999998,87.0,86.8,83.8,87.0,86.79999999999998,83.79999999999998,87.0,0.6681060321264678
Argentina,CONMEBOL,1,82.2,84.0,89.0,82.2,84.0,89.0,82.2,84.0,89.0,0.6516118024276305
Germany,UEFA,9,84.0,86.5,85.7,84.0,86.5,85.7,84.0,86.5,85.7,0.6428947926356148
Portugal,UEFA,6,85.2,84.5,86.0,85.2,84.5,86.0,85.2,84.5,86.0,0.6372348686302479
Netherlands,UEFA,7,85.2,83.5,83.0,85.2,83.5,83.0,85.2,83.5,83.0,0.5307930231808814
Croatia,UEFA,10,80.8,85.8,79.3,80.8,85.8,79.3,80.8,85.8,79.3,0.41672793335612635
Morocco,CAF,13,81.2,76.2,81.7,81.2,76.2,81.7,81.2,76.2,81.7,0.14320933975606764
Switzerland,UEFA,14,78.5,79.5,76.7,78.5,79.5,76.7,78.5,79.5,76.7,0.13470955233392745
Mexico,CONCACAF,15,76.8,78.2,82.7,76.8,78.2,82.7,76.8,78.2,82.7,0.1264610643044092
Senegal,CAF,20,79.0,79.0,80.7,79.0,79.0,80.7,79.0,79.0,80.7,0.1209314904499356
Poland,UEFA,25,77.0,77.5,83.3,77.0,77.5,83.3,77.0,77.5,83.3,0.0914718495182683
USA,CONCACAF,11,75.8,75.8,77.7,75.8,75.8,77.7,75.8,75.8,77.7,0.07121519293547833
Japan,AFC,17,75.2,77.5,75.0,75.2,77.5,75.0,75.2,77.5,75.0,0.05484221910098477 4
Ecuador,CONMEBOL,32,73.5,74.5,76.0,73.5,74.5,76.0,73.5,74.5,76.0,0.016668387121573945
Cameroon,CAF,42,76.8,75.0,77.7,76.8,75.0,77.7,76.8,75.0,77.7,0.01631707789871081
Tunisia,CAF,30,73.0,75.8,71.3,73.0,75.8,71.3,73.0,75.8,71.3,0.014853337874529846
Australia,AFC,27,72.0,73.5,72.3,72.0,73.5,72.3,72.0,73.5,72.3,0.011974166776269807
Canada,CONCACAF,39,69.2,78.0,73.0,69.2,78.0,73.0,69.2,78.0,73.0,0.01038118637937859
Saudi Arabia,AFC,53,72.8,72.8,67.7,72.8,72.8,67.7,72.8,72.8,67.7,0.002163536031653477
Spain,UEFA,8,86.5,86.0,85.0,-1.0,-1.0,-1.0,42.75,42.5,42.0,3.1865388218315835e-06
```

- task5_top10_predictions.png — bar chart of top 10 finalist probabilities



Outcome

- A ranked list of teams based on predicted finalist probability
- Visual insights into top contenders for the 2026 final
- Ready-to-use outputs for analysis and presentation

TASK 6

1. **Integrated Model Predictions into the App**

I connected the trained model to the Streamlit app, allowing it to:

- Accept user-selected teams
- Predict match outcomes at each knockout stage
- Display win probabilities and final results

2. **Designed the Knockout Flow**

The app was structured to simulate a real knockout tournament:

- **Quarterfinals**: User selects 4 teams, and 2 matches are predicted
- **Semifinal**: Winners face off
- **Final**: User selects the final opponent, and the champion is predicted

3. **Built a Clean and Interactive UI**

I used Streamlit components to create:

- Dropdowns for team selection
- Buttons to trigger predictions
- Styled headers and result boxes
- A "Predict Again" button to restart the flow

4. **Tested and Refined the Experience**

I tested the app for usability, clarity, and responsiveness. The final version is intuitive, visually engaging, and suitable for both classroom demos and public use.

5. **Submitted Demo Video**

As part of this task, I also submitted a **demo video** showcasing the app's functionality, user flow, and prediction results. This video helps explain the app's purpose and usage in under 3 minutes.

CONCLUSION

This project successfully integrates data science, machine learning, and interactive design to simulate and predict FIFA 2026 knockout-stage outcomes. From scraping and enriching team data to training models and building a user-friendly app, each task contributes to a complete, reproducible pipeline. The final Streamlit application is both technically robust and visually engaging, supported by evaluation plots, feature analysis, and demo videos. It demonstrates how predictive modeling can be applied to sports analytics in a way that is accessible, insightful, and ready for real-world use.

Future Work

While the current app successfully predicts knockout-stage outcomes using enriched team data and a trained model, future enhancements could include integrating live match updates, player-level statistics, and dynamic injury reports to improve prediction accuracy. Expanding the model to support group-stage simulations, adding multilingual support for broader accessibility, and deploying the app online for public use are also promising next steps. Additionally, experimenting with ensemble models or deep learning techniques could further refine performance and adaptability across tournaments.