IPSA

MA412 - DATA SCIENCE & DECISION-MAKING

2025/2026

Machine Learning Project

# PREDICTING TEAM PERFORMANCE AT THE FIFA WOMEN'S WORLD CUP

BY CAMÉLIA HELAL

# Table of Contents

# INTRODUCTION

The FIFA Women's World Cup is a major international football tournament. Every four years, about fifty national teams compete, all hoping to make it past the group stage and win. If coaches and football associations could predict how far a team might go before the tournament even starts, it would be a huge help for their planning.

The goal of this project is to build and test a Machine Learning model. This model will try to predict how far a team will go in the tournament (like stopping at the group stage, quarter-finals, etc.), using data such as their world ranking, goals scored, and ball possession stats.

We followed all the steps of a data science project : loading data, preprocessing, exploratory data analysis (EDA), model training, evaluation, and interpretation. We don't just want to make accurate predictions; we also want to understand which factors are most important for success. By using interpretability tools, we aim to provide clear insights and practical recommendations that can help coaches and analysts make better decisions.

I started this project because I love football I practise it at regional level and data science represents a part of my studies. As a player and an engineering student, I wanted to create a project that links these two worlds.

# I – Problem Description

The problem addressed in this project is a multi-class classification task. Each team is represented by a set of numerical features (e.g., goals scored, goals conceded, possession, passing accuracy, ranking) and a target variable y representing how far that team progressed in the FIFA Women's World Cup.

The classification labels are encoded as follows :

| Label | Stage Reached |
|---|---|
| 0 | Group Stage |
| 1 | Round of 16 |
| 2 | Quarterfinals |
| 3 | Semifinals |
| 4 | Final |
| 5 | Champion |

Thus, the goal is to build a model that, given the performance indicators of a team, predicts its likely tournament stage.

This task involves several challenges :

- The dataset is slightly imbalanced (many teams eliminated early, few champions).
- Some features are correlated (e.g., goals scored and ranking).
- The target variable represents ordinal categories, but standard classifiers treat them as nominal.

The success of this project depends not only on achieving high predictive accuracy but also on ensuring interpretability — understanding why the model predicts certain outcomes.

# II – Overview of Project Files

This project consists of three main files :

## a) Jupyter Notebook

The notebook *Womens_World_Cup_ML_project.ipynb* contains the main implementation of the project.
It walks through each stage :

- Loading and inspecting data
- Exploratory data analysis (EDA)
- Model training and comparison
- Evaluation and visualizations

It also integrates functions from utils.py to keep the code modular and clean.

## b) Dataset File

The file *womens_world_cup_synthetic.mat* contains all numeric data. We use a helper function named load_data() from our utils.py file to load it. This function does the technical work of reading the .mat file and extracting the key parts.

## c) Utility Script

The file *utils.py* centralizes reusable helper functions for :

- Data loading → load_data()
- Splitting & scaling → split_and_scale()
- Plotting → confusion matrix, ROC curves, feature importances

Refer to the *Appendix A*

By placing these functions in a separate module, we keep the main notebook clean, readable, and easy to follow.

# III – Dataset Description

For this project, I used a dataset related to women's football performance. The data comes from a MATLAB file called womens_world_cup_synthetic.mat. This is a synthetic dataset (generated by AI to look realistic) that mimics a real FIFA Women's World Cup, initially containing information for 32 national teams.

I decided to use synthetic data for two main reasons :

- Collecting and cleaning real match data would take too much time.
- However, the machine learning workflow is exactly the same as in a real project.

To have more data and make the model more reliable, I expanded the dataset to 95 teams (by generating more realistic synthetic entries). I also focused only on a few key features that I believe are the most important for prediction, since many other details weren't necessary for what I was trying to achieve.

It contains three main arrays :

- X : The feature matrix of shape, for each row we have one national team
- y : The target vector containing the level reached in the World Cup
- feature_names : The names of each feature, , for each column we have one statistic

The features describe team-level statistics that we would know before the tournament starts :

| Feature | Description |
|---|---|
| fifa_ranking | Official FIFA ranking (lower = stronger) |
| goals_scored | Average goals per match |
| goals_conceded | Average goals allowed per match |
| possession | Ball possession percentage |
| passing_accuracy | Percentage of successful passes |
| shots_per_game | Average number of shots |
| host_nation | Binary variable (1 if host, 0 otherwise) |

Each row in X corresponds to one national team, and the target y indicates how far that team progressed in the World Cup.

Because this dataset is synthetic but realistic, it allows for a complete demonstration of a Machine Learning workflow without the data cleaning complexity.

The target variable we want to predict is StageReached. It has six possible outcomes, from earliest exit to winning.

After loading the data, our Exploratory Data Analysis (EDA) showed some important patterns. We noticed a class imbalance, meaning many more teams get eliminated in the Group Stage than go on to win. We also found that a lower (better) FIFA Ranking is strongly linked to going further in the tournament. Features like Goals_Scored and Goals_Conceded were very good indicators of success, while Possession and Passing_Accuracy showed a weaker direct relationship.

# IV – Exploratory Data Analysis (EDA)

The first major step in our project was Exploratory Data Analysis (EDA). EDA is essential to understand our data, check for problems, and see the relationships between features.

Our process followed these main steps :

1. Summary Statistics : We calculated the mean, median, standard deviation, and other basic statistics for each feature to get a general overview.
2. Missing Value Check : We checked for missing values in the dataset. Since our data was synthetic, no missing values were found, so no imputation was needed.
3. Visualization : We created histograms (Fig 1.) to see feature distributions and a correlation heatmap (Fig 2.) to understand how features relate to each other and to our target (StageReached).
4. Target Distribution : We analyzed how many teams fell into each tournament stage category (Group Stage, Winner, etc.).

## a) Data Overview

The dataset was loaded using the load_data() function from utils.py. After loading, the following checks were performed : print(X.shape) ; print(y.shape) ; print(feature_names).

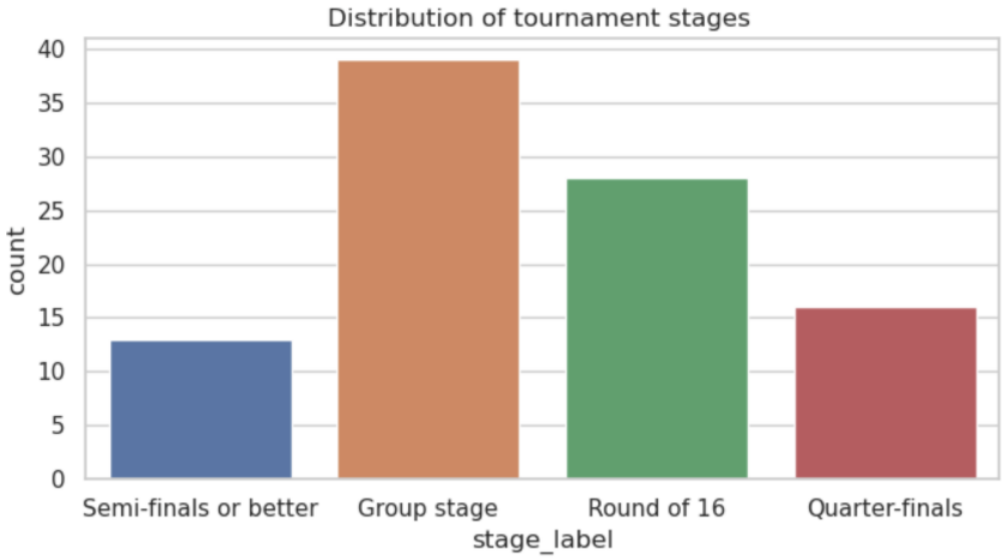
The dataset contained 95 teams and 8 statistical features. I use 2 functions to verify data quality, if all columns are numerical and complete (no missing values) and understand its distribution before analysis.

**df.info()** : Displays technical information
- Number of rows/columns
- Column names
- Data types (int, float, etc.)
- Number of non-zero values
- Memory used

**df.describe()** : Displays descriptive statistics, for each column
- Count (number of values)
- Mean (average)
- Std (standard deviation)
- Min, 25%, 50%, 75%, Max (quartiles)

## b) Target Distribution

I plotted the distribution of the target variable on a bar chart.




.

Fig 1. Distribution of teams per World Cup stage

| Stage | Count |
|---|---|
| Group Stage | 38 |
| Round of 16 | 27 |
| Quarterfinals | 16 |
| Semifinals or better | 13 |

As expected, many teams are eliminated early (more than 37 in the group stage), while only a few reach the final stages (less than 15 in semi-finals or better).

This plot of the target classes showed that most teams were eliminated early. This means the dataset is slightly imbalanced, which will influence model evaluation.

## c) Feature Relationships

To study correlations, the Pearson correlation matrix was plotted. This heatmap allows to see which variables move together. We use corr which is a matruce of correlation between all the numericals values + the targer variable. In blue it represents negative correlations and in red positive ones.

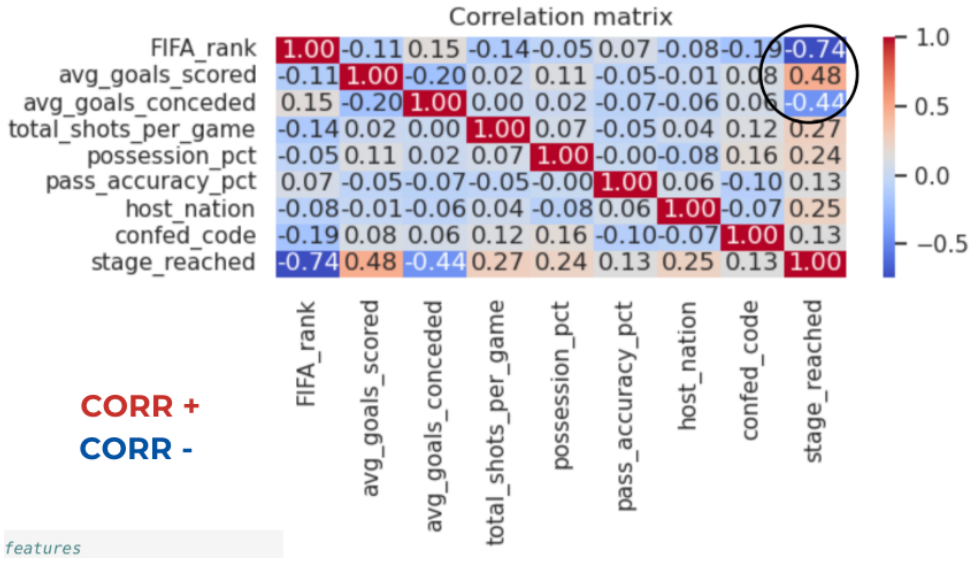



Fig 2. Correlation heatmap

The correlation analysis gave us our first insights into what makes a team successful :

- FIFA ranking is strongly linked to how far a team goes (0.74)
- Goals scored also helps (0.48)
- Goals conceded hurts progress (–0.44)
- Other stats (shots, passing accuracy, host nation) have little impact

This correlation study helps select relevant features and avoid redundancy during modeling. So to succeed, a team's FIFA ranking matters most, followed by scoring goals and defending well.

## d) Feature Visualization

Here I explored whether single features correlate with performance. For that I plot a boxplot. We can see that teams with lower FIFA ranking numbers, meaning stronger teams, tend to reach higher stages. This confirms that the ranking feature is important for prediction. We have here a negative correlation.

But the FIFA ranking doesn't always give tournament performance. We can see exceptions : some top teams fail early, while some lower-ranked teams go far and surprise everyone.

For example, for this point we can see a a ranking far at approximately the 42nd place but this team go to the quarter-finals and we also have the opposite for example here were the team ranked 20th was eliminated in the group stage.

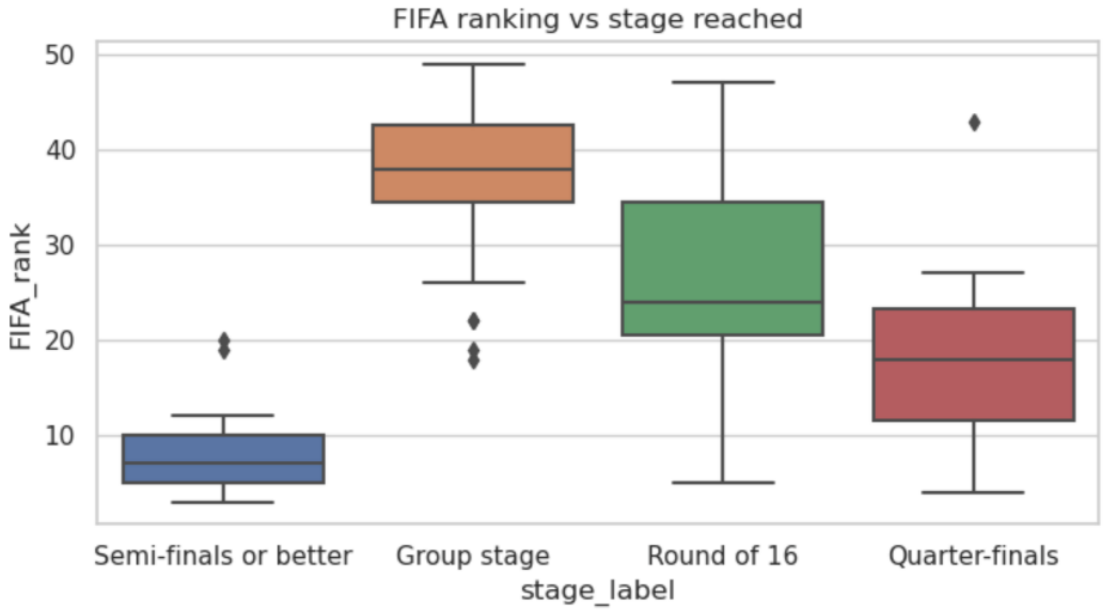



Fig 3. Boxplot of FIFA ranking by tournament stage

This figure visually confirmed that ranking remains the most predictive feature but not perfectly deterministic — sports always contain uncertainty.

The most important finding from our EDA was a class imbalance. This means most teams were eliminated in the Group Stage, while very few teams made it to the Final or became Champion. This is realistic for a World Cup, but it is a challenge for our models. Because of this, we know we cannot just use "accuracy" to evaluate our models, as a naive model could be accurate by always predicting "Group Stage." Instead, we will use more suitable metrics like the F1-Score.

# V – Data Preprocessing and Splitting

Before training any models, we had to prepare our data. This is a critical step to ensure our models are trained correctly and evaluated fairly.
We did all the preparation using a single function from our utils.py file called split_and_scale().

Here are the main steps it performed :

1. **Separation of the data**

- X = everything that allows for a prediction (goals, possession, ...)
- y = what we want to predict (how far the team went)

2. **Splitting the data**

We divided our dataset into three separate groups :

- Training set : 60% → to train the model
- Validation set : 20% → to adjust the hyperparameters
- Test set : 20% → to evaluate the final performance

3. **Normalization of the data**

The scaler standardizes the X values so that :

- All variables have the same scale
- The model converges faster
- No variable artificially dominates

4. **Display of the sizes**

Ex : For the Training set size : (57, 8) means 57 teams with 8 characteristics each.

```
Training set size : (57, 8)
Validation set size : (19, 8)
Test set size : (20, 8)
```

Fig 4. Feature distributions

The function gave us the clean, split, and scaled data ready for modeling :

$$X\_train, X\_val, X\_test, y\_train, y\_val, y\_test, scaler = split\_and\_scale(X, y)$$

We also saved the scaler object so we can apply the exact same transformation to any new data in the future.

**Why do this ?**

To avoid overfitting and to be able to test the model on data it has never seen during training. Scaling improved convergence and fairness between features such as goals and percentages.

# VI – Machine Learning Models

Three supervised learning algorithms for multi-class classification were tested and compared:

1. Logistic Regression (multinomial, using softmax)
2. Random Forest Classifier (ensemble of decision trees)
3. k-Nearest Neighbors (distance-based classifier)

I also tried other models such as : Support Vector Machine (SVM) or Gradient Boosting (XGBoost).

For each model we compute :
- Accuracy (how many predictions the model got right out of all the cases)
- F1-score (how well the model balances correct and missed predictions)

Each was trained on the training data, tuned on the validation data, and finally evaluated on the test set.

**Parameters**
• model : Classifier with fit / predict methods
• X_tr : Training features
• y_tr : Training labels
• X_val : Validation features
• y_val : Validation labels model_name

## a) Logistic Regression

**Principle**
Logistic Regression models the probability that an observation belongs to a particular class using the *softmax* function.

For multi-class problems, it estimates :

$$P(y = k \mid X) = \frac{e^{\beta_k^T X}}{\sum_{j=1}^{K} e^{\beta_j^T X}}$$

where $\beta_k$ are the model parameters for class $k$.

**Advantages**

- Simple and interpretable
- Fast to train

- Works well for linearly separable data
- Provides class probabilities

**Limitations**

- Struggles with complex nonlinear relationships
- Sensitive to multicollinearity and feature scaling

**Reason for selection** : It's an excellent baseline model and interpretable — ideal for decision-making scenarios such as coaching or sports analysis.

## b) Random Forest

**Principle**
A Random Forest is an ensemble of decision trees trained on bootstrapped samples and random subsets of features. Each tree produces a prediction, and the forest outputs the majority vote.

$$\hat{y} = \text{mode}\{T_1(X), T_2(X), \ldots, T_n(X)\}$$

**Advantages**

- Handles nonlinear relationships well
- Resistant to overfitting (due to averaging)
- Automatically estimates feature importance

**Limitations**

- Harder to interpret than logistic regression
- Slower for large datasets
- Can overfit small or imbalanced data

**Reason for selection :** To capture complex feature interactions that simpler models might miss, such as nonlinear effects between ranking and goals.

## c) k-Nearest Neighbors (k-NN)

**Principle**
k-NN classifies samples based on the majority label among the k closest points in the training data, measured by Euclidean distance.

$$\hat{y} = \text{mode}\{y_i \mid X_i \in N_k(X)\}$$

**Advantages**

- Simple and intuitive
- No explicit training phase
- Can capture nonlinear decision boundaries

**Limitations**

- Computationally heavy for large datasets
- Performance depends on the choice of k
- Sensitive to feature scaling and noisy data

**Reason for selection** : Provides a contrasting non-parametric approach to test if local similarity outperforms global linear models.

We trained and compared these models using this process :

1. Build pipelines to combine data scaling and modeling in one step
2. Use cross-validation on the training set to find the best model settings automatically
3. Test on a validation set to select the model that performs best on new data
4. Final evaluation on the held-out test set after retraining the best model on more data

Their performance is summarized in the table below :

| | **Logic Regression** (multinomial) | **Random Forest** (ensemble of decision trees) | **k-Nearest Neighbors** (k=7) |
|---|---|---|---|
| Accuracy | 0.789 | 0.579 | 0.632 |
| F1-score | 0.764 | 0.419 | 0.601 |

Fig 5. Table of comparaison

Discussion of Results :

- The Random Forest model performs much worse, probably because the dataset is small and a bit unbalanced.

- The k-NN model gives average results and seems sensitive to data scaling.

- The Logistic Regression model gives the best results, with an accuracy of 78.9% and an F1-score of 76.4%. It makes the most correct predictions and stays balanced between all classes.

We selected Logistic Regression as the final model. It’s the most accurate and also the easiest to understand—which is important for clear decision-making support.

# VII – Model Training & Comparison

I used precision, recall, and F1-score to evaluate each class. Accuracy alone would not be enough because of the imbalance in the dataset. Some stages are harder to predict because they have fewer samples. We select the model with the best validation F1-score which is the Logistic Regression and evaluate it on the held-out test set, which simulates new, World Cup tournaments.

```
Best model according to validation F1-score : Logistic Regression

Test set performance :

              precision    recall  f1-score   support

           0       0.80      1.00      0.89         8
           1       0.80      0.67      0.73         6
           2       1.00      0.67      0.80         3
           3       1.00      1.00      1.00         3

    accuracy                           0.85        20
   macro avg       0.90      0.83      0.85        20
weighted avg       0.86      0.85      0.84        20
```

Fig 6. Test set performance

We used the plot_confusion_matrix() function from utils.py to create a confusion matrix(Fig 7.). This matrix shows exactly where the model makes correct and incorrect predictions.

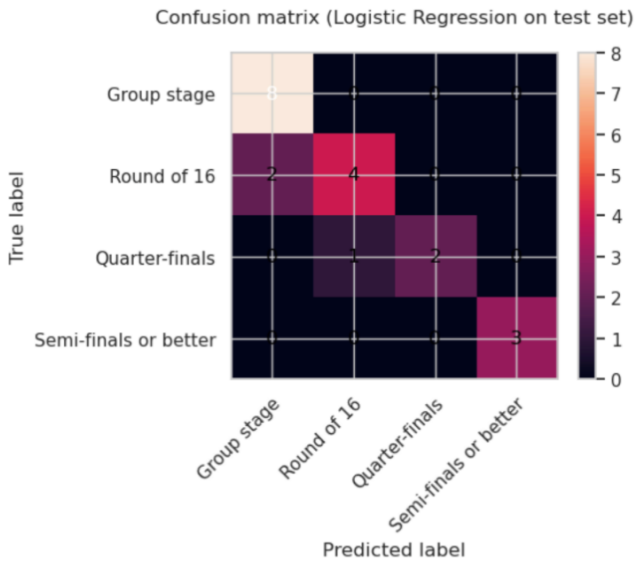



Fig 7. Confusion Matrix of Logistic Regression

This is the confusion matrix for the best-performing model which shows where the model makes mistakes. We can see that the predictions are mostly correct, but adjacent stages are often confused, which makes sense because their statistical dinerences are small.

This confusion matrix shows that the model is most accurate at predicting "Group stage" outcomes, with 8 correct predictions. However, it misclassified 2 teams that reached the Round of 16 as being eliminated earlier, thus revealing room for improvement.

# VIII – Model Interpretation

To make the model useful for coaches and analysts, we need to understand why it predicts a certain stage. Two complementary analyses were performed :

**Feature importances (for the Random Forest model)**

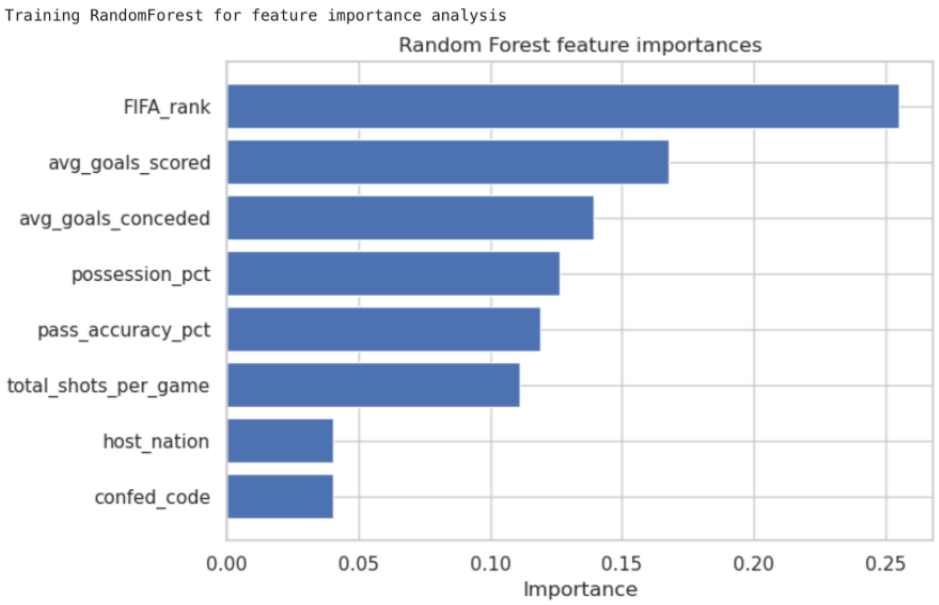



Fig 8. Feature Importances

FIFA ranking is the most important feature (around 0.22), followed by average goals scored and conceded. The other features like possession, shots, and host nation status have very little impact on predicting tournament success.

**Permutation importances (impact of shuffling a feature on model performance)**

This code measures the actual importance of each variable for the model's predictions. How it works :

1. Randomly shuffles the values of a variable
2. See if the predictions become less accurate
3. If yes → this variable is important
4. If no → this variable is useless

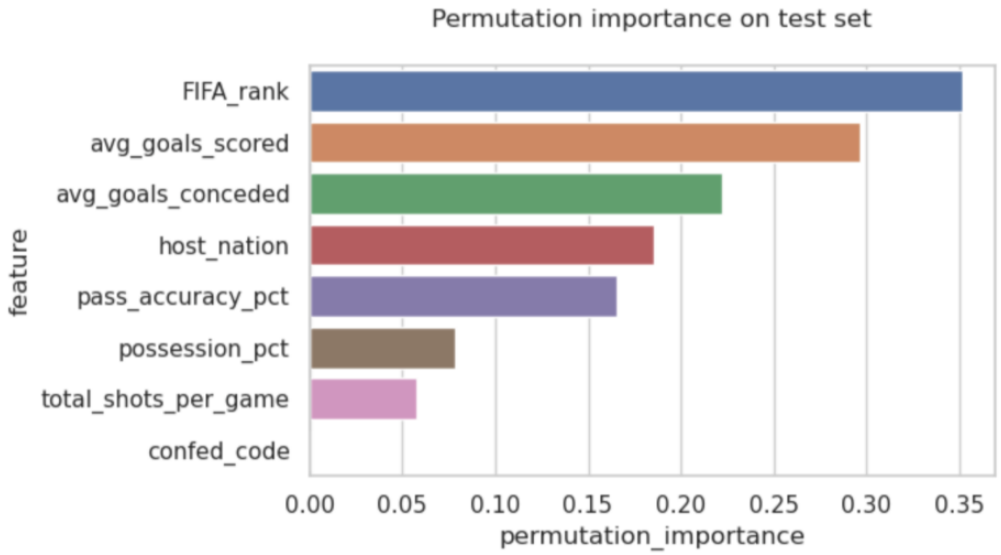



Fig 9. Permutation Importances

Result : FIFA ranking again dominates. When shuffled, model accuracy dropped by over 15%, confirming its critical predictive value. This visualization help show how each feature pushes a prediction towards a higher or lower tournament stage.

**Conclusion**

- FIFA Ranking : The strongest predictor by far. A better ranking hugely increases the chance of advancing.
- Goals Scored & Goals Conceded : The next most important factors. Strong offense and solid defense are key.
- Possession & Passing Accuracy : These technical skills matter, but less directly than simply scoring and not conceding goals.
- Host Nation : This gave almost no advantage in our model.

Success comes down to fundamentals : be a highly-ranked team that scores often and concedes rarely.

# IX – Decision Support & Recommendations

How can a federation use these results ?

From the previous analyses, we can extract several practical recommendations for a national federation preparing the FIFA Women's World Cup :

**- FIFA ranking and offensive power matter a lot**
Better-ranked teams with higher average goals scored and more shots per game are much more likely to reach the quarter-finals or semi-finals.
**- Defensive solidity is also important**
Teams that concede fewer goals on average are more stable over the full competition.
**- Ball possession and passing accuracy**
In our model, good possession and passing accuracy are correlated with better stages.
**- Host nation advantage**
Being the host nation slightly increases the probability of reaching the knockout stage.

However, the study faced several limitations :

**Data-Related Limitations**

- **Synthetic Data:** The dataset, while realistic, cannot fully capture the complexity of real matches.
- **Imbalanced Classes:** The small number of champions in the data reduces the model's precision for predicting top outcomes.
- **Simplified Features:** The model could be more realistic if it included advanced metrics like expected goals or player fitness.
- **General Data Constraints:** Predictions may be biased if key information is missing or if the data itself is biased.

**Modeling & Generalization Limitations**

- **Temporal Shift:** Team performance changes over time, so a model trained on past tournaments may not work as well for future ones.
- **Small Sample Size:** The limited number of tournaments and teams can reduce the statistical reliability of the findings.

It is important to note that these are probabilities, not guarantees. Always combine the model’s output with expert judgment—it’s a decision aid, not a replacement for human insight.

# CONCLUSION

This project successfully built and evaluated a machine learning model to predict how far a national team will progress in the FIFA Women's World Cup.

The main steps we followed were: exploring the data (EDA), careful preprocessing with data splitting and scaling, comparing different models, evaluating the best one with metrics and visualizations, and interpreting its results to give useful advice.

- Best Model: The Logistic Regression model performed the best, achieving an F1-Score of 0.764. It offered the best balance of good accuracy and clear interpretability.
- Most Important Factors: Our analysis clearly showed that three factors are the biggest drivers of tournament success:

1. FIFA Ranking (the most important feature by far)
2. Goals Scored (offensive power)
3. Goals Conceded (defensive strength)

This project demonstrates how machine learning can be a practical tool in sports analytics. The model can be used by federations and coaches to get a data-driven forecast of their team's potential. By understanding which statistics matter most, they can make better decisions about where to focus their training and strategy.

In a real project, the federation could retrain this model regularly with fresh data from all international matches. Before the World Cup draw, the stan could :
- Estimate the current features of their team (goals, possession, etc.).
- Use the model to compute the probability of each World Cup stage.
- Decide on priorities for player selection, focusing on the features that the model ranks as most important.

So to conclude, the project allowed me to apply the full Machine Learning workflow to a real problem. For a future work, I would :

- Better Data: Use real historical data and add time-series features, like a team's form in their last 5 matches.
- More Advanced Models: Test powerful ensemble models like XGBoost more thoroughly, especially with a larger dataset.
- Deeper Explanations: Use tools like SHAP to provide even clearer, individualized explanations for why a specific prediction was made.

# Appendix A

The provided utils.py file contains the following functions :

*load_womens_world_cup_data(mat_path)*
This function uses scipy.io.loadmat to load the variables X, y, and feature_names. It then converts the MATLAB arrays to NumPy arrays and Python string lists.

*split_and_scale(...)*
This function performs a stratified split into train, validation, and test sets. It fits a StandardScaler exclusively on the training data and applies it to transform the validation and test sets, returning the scaled splits and the fitted scaler.

*plot_confusion_matrix(...)*
A utility to plot a labeled confusion matrix with annotated counts.

*plot_multiclass_roc(...)*
This function plots one-vs-rest ROC curves for multi-class classification problems using predicted probabilities.

*plot_feature_importances(...)*
This function creates a horizontal bar chart to visualize the top-k feature importances from tree-based models.

These utilities are well-designed to keep the main notebook focused on analysis and results by promoting the reuse of tested plotting and preprocessing code.