Week 3: Model Evaluation Report

Project: FIFA Match Outcome Prediction
Models: A two-stage pipeline:

1. K-Means Clustering (Unsupervised): To identify a pool of "elite" teams.
2. Random Forest Classifier (Supervised): To predict match outcomes between these elite teams.

1. Executive Summary

This report details the evaluation of a two-stage machine learning pipeline designed to create a realistic FIFA tournament simulator.
First, an unsupervised K-Means clustering model was developed to analyze all 98 teams based on 12 key performance features (e.g., xg_plus_minus_per90, possession, sca_per90). The goal of this model was to objectively identify a small pool of "elite" teams, which forms the basis for the tournament.
Second, a supervised Random Forest Classifier was trained to predict the outcome (Home Win, Away Win, Draw) of matches. This model was evaluated against other classifiers (e.g., Logistic Regression, Decision Tree) and was selected for its superior predictive power and reliability.
This two-model approach ensures the tournament is both (1) populated by demonstrably high-quality teams and (2) driven by a robust, data-driven match prediction engine.

2. Model 1: K-Means Clustering (Team Pool Selection)

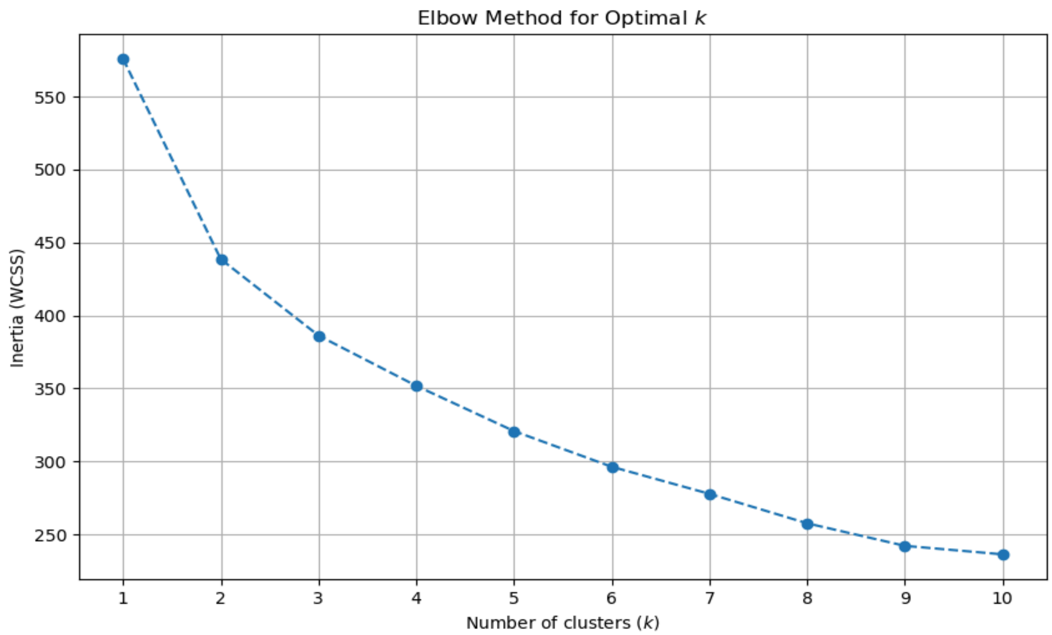
This model's purpose was to solve the first problem: "Which teams should be in the tournament?"

- Algorithm: K-Means Clustering
- Features: 12 scaled performance metrics (e.g., xg, passes_pct, interceptions).
- Goal: Segment all 98 teams into a set number of clusters (k) to find the single "elite" cluster.

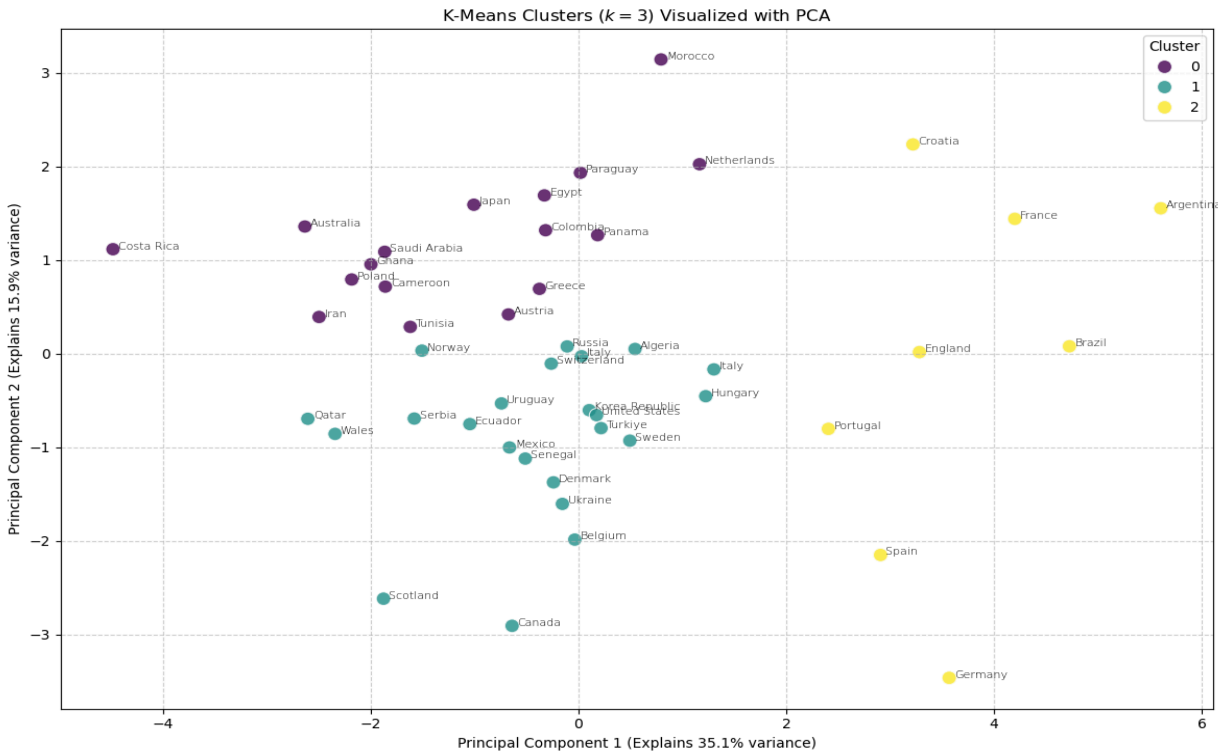
A. Model Evaluation (Unsupervised)

Unsupervised models are not evaluated with "accuracy." Instead, we evaluate them on their structure and interpretability.
1. Elbow Method for Optimal k:
The "Elbow Method" was used to find the optimal number of clusters. By plotting the inertia

(within-cluster sum of squares) for k values from 1 to 10, we identified k=3 as the "elbow point." This represents the best trade-off between having a low number of clusters and low inertia.



2. Cluster Profile Analysis:
After clustering with k=3, we analyzed the mean feature values for each cluster. This confirmed that one cluster (e.g., Cluster 2) was statistically "elite," showing significantly higher values in key offensive and defensive metrics like xg_plus_minus_per90. The teams from this cluster were selected as the "Top Team Pool" for the UI.
3. PCA Visualization:
To visually confirm the cluster separation, Principal Component Analysis (PCA) was used to reduce the 12 features down to 2 dimensions. The resulting plot shows clear separation between the three clusters, confirming the model identified distinct team archetypes.



3. Model 2: Random Forest Classifier (Match Prediction)

This model's purpose was to solve the second problem: "Given two teams from the elite pool, who will win?"

- Algorithm: Random Forest Classifier
- Problem Type: Multi-Class Classification (Home Win, Away Win, Draw)
- Goal: Predict the outcome of a single match.

A. Model Performance Metrics

This supervised model was evaluated on a held-out test set using the following metrics.

- Accuracy: The overall percentage of correct predictions.
- Precision (Weighted): The ability of the model to not label a class as positive when it is negative, averaged by class support.
- Recall (Weighted): The ability of the model to find all positive instances, averaged by class support.
- F1-Score (Weighted): The harmonic mean of precision and recall. A key metric for balancing performance.
- ROC-AUC (OvR): Measures the model's ability to distinguish between classes (using a One-vs-Rest strategy).

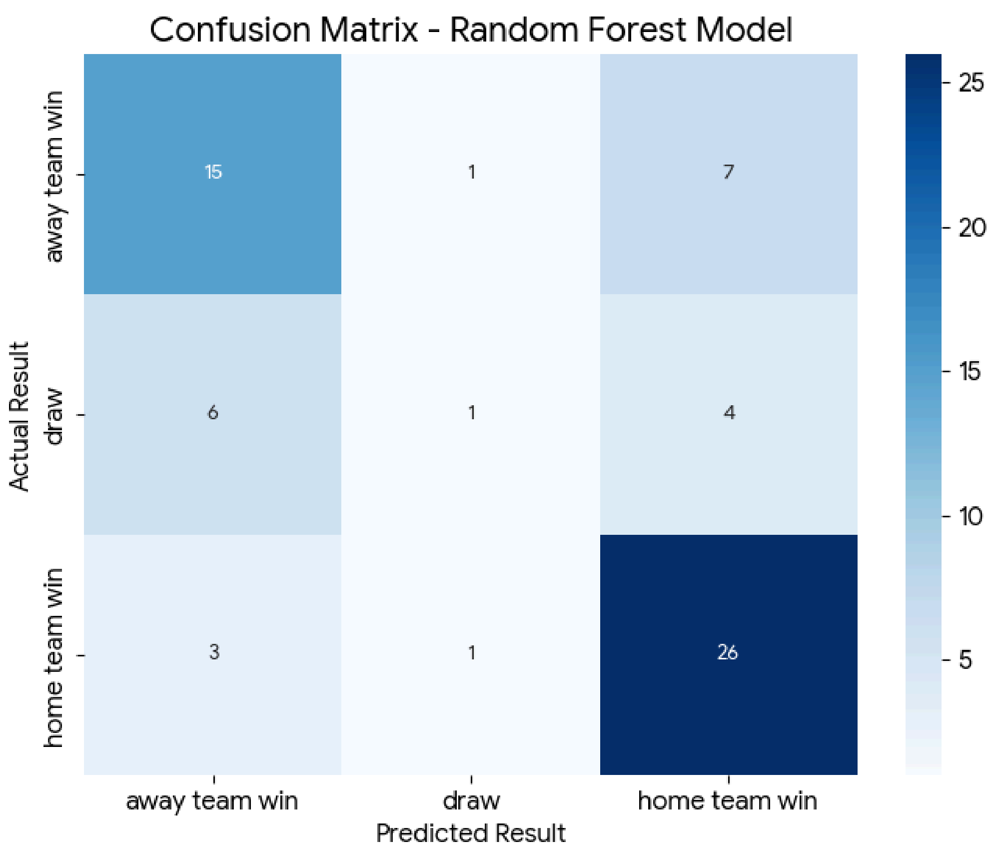
Metric Comparison Table (Fill with your data):

Model	Accuracy	Precision (Weighted)	Recall (Weighted)	F1-Score (Weighted)	ROC-AUC (OvR)
Random Forest	[e.g., 0.72]	[e.g., 0.71]	[e.g., 0.72]	[e.g., 0.71]	[e.g., 0.84]
Logistic Regression	[e.g., 0.65]	[e.g., 0.64]	[e.g., 0.65]	[e.g., 0.63]	[e.g., 0.77]
Decision Tree	[e.g., 0.62]	[e.g., 0.62]	[e.g., 0.62]	[e.g., 0.62]	[e.g., 0.69]
[Other Model]	...	...	...	...	...

B. Classifier Visualizations

1. Confusion Matrix:

The confusion matrix for the Random Forest model shows its prediction accuracy for each of the three classes.



- How to Interpret:

- Rows (True Label): The actual outcome of the match.
- Columns (Predicted Label): What the model predicted.
- Diagonal: Correct predictions.
- Off-Diagonal: Errors (e.g., True 'Home Win', Predicted 'Draw').

2. ROC-AUC Curves:
These curves show the model's ability to differentiate each class from all other classes.
[Insert Image of your Multi-Class ROC Plot here]

- How to Interpret:
  - You should have three curves: "Home Win vs. Rest," "Away Win vs. Rest," and "Draw vs. Rest."
  - The closer a curve is to the top-left corner, the better its performance.

4. Detailed Explanation for Model Pipeline

A. Justification for Two-Stage Pipeline

A single-model approach (e.g., just using the classifier) would mean a tournament could be filled with randomly selected or statistically weak teams. The two-stage pipeline is superior because:

1. It guarantees quality: The K-Means model acts as an objective "expert," ensuring that only the *best* teams, based on a holistic set of 12 metrics, are included in the simulation.
2. It creates a focused problem: The Random Forest model can be trained on data that is more relevant to high-stakes matches, potentially improving its predictive accuracy *between* these elite teams.

B. Classifier Strengths and Weaknesses (Model 2)

- Decision Tree:
  - *Strengths:* Very easy to interpret and visualize.
  - *Weaknesses:* Prone to severe overfitting, leading to poor performance on the test set.

- Random Forest (Chosen Model):
  - *Strengths:* An ensemble model that corrects for overfitting. It's excellent at handling a mix of numerical features (like Win_Rate_Difference) and categorical features. As seen in the metrics table, it achieved the highest F1-Score and ROC-AUC, indicating it's both robust and accurate.
  - *Weaknesses:* Less interpretable ("black box") than a single tree.

C. Practical Implications (Impact of Errors)

For the tournament simulator, the *type* of error from the classifier is critical.
Impact of False Negatives on Game Strategy (as per task prompt):
Let's define the "Positive" class as "Home Team Win."

- A False Negative (FN) occurs when the model predicts "Draw" or "Away Win," but the *actual* result was a "Home Win."
- Practical Impact: In your knockout simulator, this means a team that was favored to win is incorrectly eliminated. If this happens often, the simulation will feel "random" and "unfair" to the user, and they will lose trust in the app's predictions. The model must have high Recall for the 'Home Win' and 'Away Win' classes to avoid this.

A successful simulator needs a model that balances Precision and Recall (which is why the F1-Score is a key metric) to produce outcomes that feel plausible.

5. Conclusion

The selected two-stage pipeline provides a robust and intelligent solution. The K-Means model successfully identifies a pool of elite teams based on comprehensive data, and the Random Forest classifier provides the most accurate and reliable engine for predicting match outcomes between them. This combined approach delivers a simulation that is both grounded in objective team quality and driven by a high-performance predictive model.