Week 2: Model Building And Training

This document summarizes the approach taken to build, train, and evaluate classification models to predict the outcomes of FIFA World Cup matches.

1. Task Overview

The objective of this task was to implement and evaluate at least two classification models to predict a match's outcome. The randomforest.py script, which predicts results from the fifaworldcup_with_win_rates.csv dataset, was used as the foundation. The target variable is Result, which has three possible classes:

- home team win
- away team win
- draw

Two models were implemented: Logistic Regression and Random Forest Classifier.

2. Data, Features, and Preprocessing

Data and Feature Engineering

- Dataset: fifaworldcup_with_win_rates.csv
- Target Variable: Result
- Selected Features:
    - Knockout Stage (Numeric)
    - Home Team Name (Categorical)
    - Away Team Name (Categorical)
- Engineered Features: Two features were created to add predictive power:
    1. Win_Rate_Difference: The difference between the home team's and away team's win rates (Home Team Win Rate - Away Team Win Rate). This numeric feature directly compares the teams' historical performance.
    2. Is_Home_Advantage: A binary flag (1 or 0) indicating if the "Home Team" was also the host "Country". This captures the well-known home-field advantage.

Preprocessing Steps

A ColumnTransformer pipeline was created to systematically apply the correct preprocessing to each feature type.

1. Target Encoding: The Result column (text) was converted into numeric labels using

LabelEncoder (e.g., away team win: 0, draw: 1, home team win: 2).

2. Numeric Features: The three numeric features (Win_Rate_Difference, Knockout Stage, Is_Home_Advantage) were scaled using StandardScaler. This centers the data around zero and gives it a unit variance, which is important for models like Logistic Regression.
3. Categorical Features: The two text-based features (Home Team Name, Away Team Name) were transformed using OneHotEncoder. This converts each team name into a set of binary columns, allowing the model to understand them numerically without assuming any ordinal relationship.

3. Model Training, Tuning, and Validation

Validation Strategy

To ensure a fair and unbiased evaluation, the dataset was split into two parts:

- Training Set (80%): Used to train the models and perform hyperparameter tuning.
- Test Set (20%): Held back until the very end to evaluate the final, tuned models on unseen data.

stratify=y_encoded was used during the split to ensure that the proportion of home wins, away wins, and draws was the same in both the training and test sets.

Model 1: Logistic Regression

- Description: A linear model that is a good, interpretable baseline for classification tasks.
- Hyperparameter Tuning: GridSearchCV with 5-fold cross-validation was used to find the best combination of parameters.
    - Parameters Tuned: C (regularization strength: [0.1, 1.0, 10.0]) and solver (['liblinear', 'saga']).
    - Best Found Parameters: {'classifier__C': 0.1, 'classifier__solver': 'liblinear'}
- Evaluation on Test Set:
    - Accuracy: 61.11%
    - Performance: The model was strongest at predicting home team win (precision: 0.65) and away team win (precision: 0.61) but struggled significantly with predicting draw (precision: 0.40).

Model 2: Random Forest Classifier

- Description: A powerful ensemble model that builds multiple decision trees and "votes" on the outcome. It can capture complex, non-linear relationships.
- Hyperparameter Tuning: GridSearchCV with 5-fold cross-validation was used.

    - Parameters Tuned: n_estimators (number of trees: [100, 200]) and max_depth (tree depth: [10, 20, None]).
    - Best Found Parameters: {'classifier__max_depth': 10, 'classifier__n_estimators': 100}
- Evaluation on Test Set:
    - Accuracy: 62.96%
    - Performance: The Random Forest performed slightly better overall. Like Logistic Regression, it was best at predicting home team win (precision: 0.65) and away team win (precision: 0.65) and weakest at predicting draw (precision: 0.44).

4. Summary of Results

Both models were successfully implemented, tuned, and evaluated. The Random Forest (62.96% accuracy) marginally outperformed the Logistic Regression (61.11% accuracy) on the test set.

Both models demonstrated a similar pattern: they were reasonably good at distinguishing between a home or away win but found it difficult to predict the less-frequent draw outcome, often misclassifying draws as wins for one of the teams.