Machine Learning Project: Predicting Football Match Outcomes

Why I Chose This Project

Football has been a passion of mine since childhood. I grew up watching matches and idolizing FC Barcelona, immersing myself in the beautiful game and its endless intricacies. Analyzing teams, debating strategies, and predicting outcomes became second nature to me.

When I began studying machine learning, I saw it as a perfect opportunity to combine my lifelong love for football with my growing interest in data science. Initially, my choice for this project was focused on La Liga, as it featured my favorite club, FC Barcelona. I later considered the Bundesliga due to its tactical diversity. However, I realized that these leagues didn't exhibit the same level of unpredictability I was looking for.

This led me to the English Premier League, particularly the mid-2010s seasons, renowned for their surprises and competitive balance. These years encapsulated the essence of unpredictability in football, making them an ideal dataset for a machine learning model. The idea of applying data science to predict outcomes in such a dynamic environment felt like an engaging way to learn and create something meaningful.

Why Random Forest Was Chosen

Random Forest was selected as the model for this project due to its versatility and robustness in handling structured data like football match statistics. It's an ensemble learning method that builds multiple decision trees during training and combines their outputs to make predictions. Here are the reasons for choosing Random Forest:

- **Feature Importance**: It provides insights into the importance of different predictors, helping identify the key factors influencing match outcomes.
- **Handles Missing Data**: Random Forest can handle datasets with missing values and remains effective even when some predictors are less informative.
- **Non-linear Relationships**: Football matches often have complex, non-linear relationships between features, and Random Forest is well-suited for capturing these patterns.
- **Avoids Overfitting**: By averaging the results of multiple decision trees, it reduces the risk of overfitting, making it ideal for predictive tasks with high variance.

Code Structure and Flow

1. **Data Scraping**
    - The script begins with importing required libraries such as requests and BeautifulSoup to fetch and parse data from the web.
    - The raw HTML data is cleaned and structured into a pandas DataFrame.
2. **Data Cleaning and Preprocessing**
    - Irrelevant columns are dropped, and missing values are handled.
    - The cleaned data is split into features (X) and target (y), where y represents match outcomes (e.g., home win, away win, or draw).
3. **Initial Predictor Selection**
    - Predictors such as venue, opposition, and the time of day a game is played are crucial factors in determining the outcome of a match and are readily available before the game begins.
    - These served as the starting point for our predictor selection. However, using these parameters directly is not effective for modeling purposes, so we transformed them into encoded formats to enhance their utility.
    - As a result, features like Venue_code, Opp_code, Hour, and Day_code were created. These encoded features were selected for their strong intuitive connection to match outcomes, effectively capturing contextual information such as venue advantage and the difficulty posed by the opponent.
4. **Testing Initial Predictors**
    - The test data was evaluated as a single block to assess the baseline performance of the model.
    - Metrics such as accuracy and precision were computed to gauge the effectiveness of the initial predictors.
5. **Yearly Splits for Variation Analysis**
    - To analyze temporal variations, the test data was divided into distinct years.
    - This helped identify how the model's performance fluctuated across different seasons and highlighted potential inconsistencies in prediction accuracy.

6. **Introducing Rolling Scores**
    - Rolling averages were introduced for features like goals scored (GF), goals conceded (GA), shots (Sh), and shots on target (SoT).
    - These rolling features capture short-term team performance trends, offering a dynamic perspective that complements static predictors. For example, a high rolling average for SoT indicates a team's current offensive strength.
    - Rolling averages significantly improved predictions by incorporating recent performance dynamics, which are often crucial in sports analytics.
7. **Current Model Status**
    - While the inclusion of rolling features enhanced the model, it remains a work in progress.
    - The final model is yet to be optimized fully, and further refinements are planned to improve accuracy and precision.
