IE 582: Statistical Learning for Data Mining

Final Project Report

Predictive Live Betting Strategy for Soccer Matches

Advisor: Mustafa Gökçe Baydoğan

Fatih Mehmet Yılmaz	2024702054
Yusuf Sina Öztürk	2023702075

Department of Industrial Engineering

13 January 2025

1 Introduction

1.1 Background

The growth of live sports betting has introduced a dynamic and complex avenue for leveraging real-time analytics and predictive modeling. Among the various sports, soccer provides a particularly intricate context due to the low scoring nature of the sport and the significant impact of discrete game-changing events, such as goals, red cards, and substitutions. These events drive fluctuations in betting odds, which serve as a real-time proxy for market beliefs about match outcomes. This dynamic environment presents a unique opportunity for the application of predictive analytics to develop strategies that not only forecast match outcomes but also optimize the timing and nature of betting decisions.

1.2 Problem Description

The purpose of this study is to formulate a live betting strategy that identifies the optimal moment during a soccer match to predict its final outcome, such as home win, draw or away win, and decide on a specific betting action: 'bet home win', 'bet draw', 'bet away win' or 'no action'. In this project, the optimal betting amount strategy was not considered; instead, only the optimal timing and outcome were analyzed.

1.3 Proposed Approach

The methodology for this project consists of three main stages. The first stage focuses on predicting match outcomes using statistical models. Accurate prediction forms the basis for subsequent analyses. The second stage involves determining the optimal timing for decision making using simulations that leverage betting odds predictions and match outcome probabilities for each minute of the match. In the final stage, a comprehensive strategy is developed to minimize the negative outcomes caused by the stochastic nature of soccer, such as unexpected goals, red cards, or other significant game-changing events.

2 Related Literature

The use of statistical models and betting odds to predict sports outcomes has been a key area of research. Several studies have shaped the understanding of this domain.

Mirza and Fejes (2016) analyzed how bookmakers set odds for football matches, particularly in the English Premier League. They showed that bookmakers use a combination of statistical models and expert opinions to create odds, while also incorporating a margin to ensure profitability. However, this margin reduces the accuracy of the odds as true probabilities of match outcomes.

Štrumbelj (2014) explored the conversion of betting odds into probabilities to predict match results. The study highlighted that odds are often distorted by bookmakers' margins, making them unreliable as pure probability estimates. To address this, methods for normalizing odds to remove the margin were introduced, enabling a closer approximation to actual probabilities.

Shin (1993) examined the influence of insider information on betting markets. The research revealed that when certain individuals possess privileged knowledge, the odds do not fully reflect the true chances of outcomes. Although the study primarily focused on financial markets, its findings are applicable to sports betting, where market inefficiencies can arise due to similar dynamics.

Together, these studies provide insights into the strengths and limitations of using betting odds to predict sports outcomes, shedding light on how statistical models, bookmaker strategies, and market inefficiencies interact.

3 Approach

The methodology proposed for this project integrates data preprocessing, feature engineering, predictive modeling, simulation, and strategy optimization.

3.1 Data Preparation and Preprocessing

The analysis begins with loading the imputed dataset, `match_data_imputed.csv`, and performing necessary transformations to facilitate further analysis. Probabilities are calculated using bookmaker odds for each match outcome (home win, draw, away win), normalized to account for the bookmaker's profit margin. Additional features, such as the difference between home and away win probabilities, are engineered to capture relative strengths. Match outcomes are converted into a numerical target variable to support predictive modeling, and datasets are segmented into training, validation, and testing sets based on match dates.

With Exploratory Data Analysis (EDA), the following observations and preprocessing steps were applied:

- **High Percentage of Zero Warnings and Missing Values:** EDA revealed a high percentage of zero and missing values in the dataset. To address these:
  - Missing values were imputed using linear interpolation.
  - For zero values, we verified whether they originated at the start of the match or were misinformation:
    - After analyzing distributions and specific instances, no misinformation was found in zero-valued instances.

- **Distribution of Percentage Features:** Percentage features exhibited a close-to-normal distribution. However, certain values at 0% and 100% did not fit well into the histograms:
  - These values corresponded to the first seconds of the match and were deemed valid, requiring no further modification.
- **Encoding of Text Features:** Text features were transformed using one-hot encoding to enhance the learning capability of machine learning models:
  - The `name` feature, representing opponent team names, was not encoded as it closely aligned with the `fixture_id`.
  - For advanced analyses, such as league-based models or adding a league dimension to the dataset, encoding these features could be considered.
- **Dropping Uninformative Features:** The `ticking` feature was found to be constant across the dataset and was removed due to its lack of usefulness.

3.2 Match Outcome Prediction

Tree-based boosting methods, specifically LightGBM, are employed to predict match outcomes using the engineered features. Model evaluation metrics, including accuracy, log loss, and ROC - AUC, are used to guide model selection and tuning. Sequential feature selection is applied iteratively to identify the most informative predictors, enhancing the model's predictive performance.

3.2.1 Accuracy Metrics

The following accuracy metrics were observed during the project:

- **Train Accuracy:** 0.74
- **Validation Accuracy:** 0.67
- **Test Accuracy:** 0.62
- **Simulation Accuracy:** 0.68

3.2.2 AUC Metrics

The AUC (Area Under the Curve) metrics provide insights into the model's performance for match outcome prediction for each dataset and class:

- **Overall AUC Scores:**
  - Train Set AUC: 0.926

  - Validation Set AUC: 0.820
  - Test Set AUC: 0.768
  - Simulation Set AUC: 0.838

- **Class-wise AUC Scores:**
  - *Train AUC:*
    - Class 0 (Draw): 0.917
    - Class 1 (Home Win): 0.926
    - Class 2 (Away Win): 0.936
  - *Validation AUC:*
    - Class 0 (Draw): 0.709
    - Class 1 (Home Win): 0.842
    - Class 2 (Away Win): 0.909
  - *Test AUC:*
    - Class 0 (Draw): 0.643
    - Class 1 (Home Win): 0.845
    - Class 2 (Away Win): 0.818
  - *Simulation AUC:*
    - Class 0 (Draw): 0.788
    - Class 1 (Home Win): 0.857
    - Class 2 (Away Win): 0.869

3.2.3 Feature Importance

The importance of features in the predictive model was determined using feature importance scores. The most influential features are summarized in **Table 1**.

Table 1: Feature Importance

Feature	**Importance Score**
p_tie_norm	421
Assists - home	377
Counter Attacks - away	361
Counter Attacks - home	297
Assists - away	286
p_away_norm	281
p_home_norm	280
1	270
Goal Attempts - away	238
X	238

3.3 KNN-Based Time-Series Forecasting

The K-Nearest Neighbors (KNN) algorithm is utilized for time-series forecasting of future betting odds and probabilities at specific match minutes. Observed sequences of match data are compared to historical sequences using Euclidean distances, and predictions are generated by aggregating the nearest neighbors' future sequences. Two feature sets are employed: betting odds and model-predicted probabilities, ensuring accurate KNN-based forecasting. Forecasts are integrated into simulations to provide minute-by-minute decision-making insights.

Algorithm 1 KNN-Based Forecasting

Input:
$train\_sequences$: Historical sequences of match data
$observed\_sequence$: Observed sequence up to time t
k: Number of nearest neighbors
$observed\_sequence\_length$: Length of the observed sequence
$features$: Features for distance computation
Output:
$forecast$: Predicted future values
Step 1: Prepare the Observed Sequence
Format the observed sequence for comparison.
Step 2: Compare with Training Sequences
Compute the similarity between the observed sequence and each sequence in $train\_sequences$.
Step 3: Identify Nearest Neighbors
Select the k most similar sequences.
Step 4: Aggregate Future Data
Extract and combine the future data from the k nearest neighbors.
Step 5: Generate Forecast
Compute an aggregate (mean) of the future data.
Step 6: Return the Forecast
$forecast$

3.4 Optimal Timing Determination via Simulation

Simulations are conducted using match data and predictions to evaluate betting actions at each match minute. These decisions are informed by both betting odds and predicted match outcomes, optimizing the timing for placing bets. The simulations assess the interplay of real-time odds, predictions, and match dynamics, identifying the most favorable betting opportunities.

3.4.1 Algorithm for Betting Strategy

This section explains the three main algorithms used in the betting strategy.

Firstly, we define the dataset splitting logic, which describes the process of splitting the dataset into different subsets required for training, validation, and testing. The test data is selected based on fixture IDs after a specific date (2024-11-01), while the remaining data is proportionally divided into training and validation sets.

Additional subsets are prepared for KNN-based forecasting, enabling the integration of historical data for odds and predictions.

Algorithm 2 Dataset Splitting Logic

Input: Dataset with fixture IDs and match outcomes
Output: Train, validation, and test splits; KNN datasets
Step 1: Dataset Splits
Split dataset as follows:
Test Data: Fixture IDs after '2024-11-01'
Model Training Data: 70% of leftover fixture IDs
Model Validation Data: 15% of leftover fixture IDs
Model Test Data: 15% of leftover fixture IDs
Odd KNN Data: All leftover fixture IDs
Model Prediction KNN Data: Model Test Data

After that, we developed the simulation algorithm. This algorithm simulates the betting process for each match minute. It calculates potential profits for each class (Home Win, Draw, Away Win) based on model predictions and odds, forecasts future outcomes using KNN, and identifies the optimal time to place bets.

A bet is placed only if the current profit exceeds future profits and is greater than 1, ensuring informed decision-making.

Algorithm 3 Simulation Process Algorithm

Input: Test Data, odds, and model predictions
Output: Betting decisions per match minute
for each match in Test Data **do**
  Initialize simulation at minute $t = 0$
  while $t \leq$ match duration **do**
    Calculate Profit per Class:
    for each class (Home Win, Draw, Away Win) **do**
      $\text{Profit}_c = \text{Model Prediction}_c \times \text{Odd}_c$
    end for
    Forecast Future Minutes:
    Use Odd KNN and Model Prediction KNN to forecast future odds and predictions
    Determine Maximum Profit:
    $\text{max\_profit}(t) = \max(\text{Profit}_{\text{Home}}, \text{Profit}_{\text{Draw}}, \text{Profit}_{\text{Away}})$
    Identify overall maximum profit across all future minutes
    Betting Decision:
    if $\text{max\_profit}(t) >$ future max_profit **and** $\text{max\_profit}(t) > 1$ **then**
      Place a bet on the class with maximum profit at minute t
    else
      Do nothing
    end if
  end while
end for

3.5 Strategy Refinement

To mitigate risks, **no-bet strategies** are implemented to avoid low-confidence or high-risk scenarios. This ensures that bets are placed only when there is sufficient confidence in the predicted outcome, minimizing potential losses. The strategy specifically addresses draw outcomes and goal differences to refine betting decisions.

3.5.1 No-Bet Strategy for Draws

Draw outcomes (`class_0_pred`) are inherently less predictable compared to home or away wins, as highlighted by their lower AUC values. To reduce risks, a **no-bet strategy** is applied whenever the model predicts a draw outcome. This ensures that bets are avoided in scenarios where predictions lack confidence.

3.5.2 No-Bet Strategy for Goal Differences

The betting strategy incorporates logic to handle scenarios involving goal differences. Bets are placed only when:

- The predicted outcome is a home win (`class_1_pred`) and the **goal difference is greater than 1**.
- The predicted outcome is an away win (`class_2_pred`) and the **goal difference is less than -1**.

This ensures that bets are made only in favorable conditions where a strong lead or deficit supports the predicted outcome, reducing the risk of false betting.

3.5.3 Algorithm for Strategy Refinement

Algorithm 4 Betting Logic Algorithm

Input: Match data with predictions (`model_decision`) and goal differences (`goal_diff`)
Output: Betting decision (1 for bet, 0 for no-bet)
for each match row **do**
  if `model_decision` is `class_0_pred` **then**
    Return: 0 *(no bet for draw predictions)*
  else if `model_decision` is `class_1_pred` **and** `goal_diff` >1 **then**
    Return: 1 *(bet on home win)*
  else if `model_decision` is `class_2_pred` **and** `goal_diff` <-1 **then**
    Return: 1 *(bet on away win)*
  else
    Return: 0 *(no bet)*
  end if
end for

3.6 Betting Success Check Logic

The success of each placed bet is evaluated based on the model's decision and the actual match outcome (`target`). The process involves the following:

- `successful_bet_check` **Function:**
  - Compares the model's predicted decision (`model_decision`) to the actual match result (`target`).
    - If the predicted decision matches the actual result (e.g., predicted home win and the result is home win), the bet is marked as successful (`1`).
    - Otherwise, the bet is marked as unsuccessful (`0`).

Profit or loss for each placed bet is calculated based on its outcome and the associated betting odds:

- `calculate_final_profit` **Function:**

- If the bet is successful:
  - Profit is calculated as the **betting odd minus 1**. For example, if the betting odd is 2.5, the profit is `2.5 - 1 = 1.5`.
- If the bet is unsuccessful:
  - A loss of `-1` is assigned.

The overall workflow for evaluating bets and calculating profits is summarized below:

This algorithm evaluates the effectiveness of the betting strategy by calculating profits for each placed bet. Successful bets yield a profit equal to the odd minus 1, while unsuccessful bets result in a loss of 1. It then aggregates results into two output tables: one for minute-wise profits and another summarizing overall betting performance.

1. **Mark Successful Bets:**
   - Use the `successful_bet_check` function to determine whether each bet in the `betted_fixture_minutes` DataFrame was successful (`1`) or unsuccessful (`0`).
2. **Calculate Profits:**
   - Use the `calculate_final_profit` function to compute the profit or loss for each bet based on its outcome and associated odds.
3. **Generate Final Outputs:**
   - Update the `betted_fixture_minutes` DataFrame to include the following columns:
     - `successful_bet`: Indicates whether the bet was successful (`1`) or not (`0`).
     - `final_profit`: Displays the calculated profit or loss for the bet.

Algorithm 5 Result Evaluation and Summary Algorithm

Input: Bets placed during simulation
Output: Profit evaluation and summary tables
for each placed bet **do**
  if bet is successful **then**
    Profit = Betted Odd − 1
  else
    Profit = −1
  end if
end for
Aggregate all profits for performance evaluation
Generate Output DataFrames:
Minute-wise Profit: Includes minute, class, profit, and maximum profit
Betting Summary: Columns include Applied Minute, Betted Class, Betted Odd, Match Result Class, and Profit
Sum up all profits to assess overall strategy effectiveness

4 Results

The final profit from the implemented betting strategy is **0.69 units**, equating to a **2.3% return on investment (ROI)**. This result is based on a **fixed bet of 1 unit per match** and encompasses **30 betted matches out of a total of 111 analyzed matches**. While the return is modest, these findings highlight the potential of predictive analytics in supporting informed betting decisions. However, the results also underscore opportunities for refining the strategy to further **optimize profits** and better **manage risks**.

5 Future Work

Future research can address several limitations of this study.

Firstly, strategies for determining optimal betting amounts, rather than a fixed 1-unit bet, could be developed. This would involve predicted probabilities, risk tolerance, and expected profits.

Second, enhanced feature engineering could improve predictions, particularly for draws, by incorporating momentum indicators, event-based features, and contextual league or team statistics in general.

Finally, dynamic strategy testing with real-time data could validate the approach under actual betting conditions, allowing for the development of adaptive strategies that respond to live odds and significant game events.

6 Code Repository

The implementation code for this project, including the LightGBM model, KNN-based forecasting, and simulation framework, is available at the following link: GitHub Repository for IE 582 Project.

References

1. Jonas Mirza and Niklas Fejes, 2016, "Statistical Football Modeling: A Study of Football Betting and Implementation of Statistical Algorithms in Premier League," available online: `http://www.it.uu.se/edu/course/homepage/projektTDB/ht15/project16/Project16_Report.pdf`
2. Štrumbelj, E., 2014. "On determining probability forecasts from betting odds." *International Journal of Forecasting*, 30(4), pp. 934-943.
3. Shin, H.S., 1993. "Measuring the incidence of insider trading in a market for state-contingent claims." *The Economic Journal*, 103(420), pp. 1141-1153.