HeadMind Partners
AI & Blockchain

Master Thesis in Data and Computer Science

Optimizing Profits in Sports Betting

Author :
M. Julien Delavande

Supervisors :
M. Pierre Louis Perez

From Avril 2024 to September 2024

Abstract

In the dynamic and high-stakes world of sports betting, effective predictive modeling and bankroll management are crucial for maximizing returns while mitigating risks. This report presents the development and implementation of a comprehensive system designed to optimize sports betting strategies, with a focus on football matches. The system is grounded in a theoretical framework that models the interactions between two agents—the bettor and the bookmaker—where, at each time t, the bettor allocates a fraction of their bankroll across possible outcomes, and the bookmaker sets the odds for each outcome. To manage the complexities inherent in dynamic betting environments, we simplify the problem by transitioning from a dynamic optimization framework to a static one, allowing for tractable solutions and efficient computations.

Leveraging predictive models, including logistic regression, and utility-based optimization techniques such as the Kelly Criterion, the system aims to forecast match outcomes probabilites accurately and allocate betting capital efficiently. The architecture integrates data collection from various sources, predictive modeling, optimization algorithms, and deployment on a cloud-based infrastructure using Kubernetes on Azure. Through both Monte Carlo simulations and real-world online testing over a five-week period, the strategies were evaluated for performance and robustness.

The results demonstrate that sophisticated utility-based strategies significantly outperform naive betting approaches, achieving higher returns and better risk management. The transition to a static framework enables the effective application of these strategies in a practical setting. The deployment of the system on Azure Kubernetes Service (AKS) ensures scalability, reliability, and the ability to handle real-time data processing demands. Limitations and future enhancements are discussed, including the incorporation of more complex models and dynamic risk preferences. This work contributes to the field of sports analytics by providing a practical framework for optimized betting strategies in a real-world environment, grounded in a solid theoretical foundation.

Contents

Chapter 1

Introduction

Sports betting has evolved into a sophisticated industry that combines statistical analysis, predictive modeling, and strategic financial management [1]. With the global popularity of football and the abundance of data available, there is a significant opportunity to apply advanced analytical techniques to optimize betting strategies. The challenge lies in accurately predicting match outcomes and effectively managing the allocation of betting capital to maximize returns while minimizing risk.

This report presents the development and implementation of a comprehensive system designed to address these challenges in sports betting. The system integrates predictive modeling to forecast football match outcomes and optimization algorithms to determine the optimal allocation of a bettor's bankroll. The focus is on creating a practical, scalable solution that can operate in real-time, leveraging cloud-based technologies and microservices architecture.

1.1 Background and Motivation

The sports betting market is highly competitive, with bettors seeking any edge to improve their chances of success. Traditional betting strategies often rely on intuition or simplistic models that fail to account for the complexities of sports data and market dynamics. The advancement of machine learning and statistical methods offers the potential to enhance predictive accuracy and optimize betting decisions systematically [7].

Effective bankroll management is equally important, as even accurate predictions can lead to losses if the betting amounts are not strategically allocated. The application of utility theory and optimization techniques, such as the Kelly Criterion, provides a mathematical framework for balancing risk and reward in betting decisions.

1.2 Objectives of the Study

The primary objectives of this study are:

- To establish a rigorous mathematical framework that defines the theoretical foundations and sets the stage for the study.
- To develop a predictive model that accurately estimates the probabilities of football match outcomes using historical and real-time data.
- To design an optimization module that calculates the optimal fraction of the bankroll to wager on each betting opportunity, applying various utility-based strategies.
- To implement a scalable, microservices-based system architecture that integrates data collection, predictive modeling, optimization, and user interface components.
- To deploy the system on a cloud platform using Kubernetes for scalability and reliability.
- To evaluate the performance of different betting strategies through Monte Carlo simulations and real-world online testing.

1.3 Scope of the Report

This report details the theoretical framework underlying the predictive modeling and optimization strategies, the system architecture and implementation, and the results of both simulated and real-world testing. The report is organized as follows:

- Chapter 2 provides a theoretical presentation of the models and preliminary research conducted.
- Chapter 3 describes the design and implementation of the solution, including system architecture and data management.
- Chapter 4 focuses on the development, training, and evaluation of the predictive models for match outcomes.
- Chapter 5 discusses the optimization of bankroll allocation using various strategies.
- Chapter 6 details the deployment of the complete system on Azure Kubernetes Service and the practical considerations involved.
- Chapter 7 presents the conclusions drawn from the study and discusses potential future work.

1.4 Significance of the Study

By integrating advanced predictive modeling with optimized bankroll allocation, this work aims to contribute to the field of sports analytics and betting strategies. The deployment of the system on a scalable cloud infrastructure demonstrates its practical applicability and readiness for real-world use. The findings from this study have implications for bettors seeking to enhance their strategies, as well as for researchers interested in the application of machine learning and optimization techniques in sports betting.

Chapter 2

Theoretical presentation and preliminary research

2.1 Introduction

In this section, we introduce a mathematical framework that describes the sports betting market. We model the interactions between bettors and bookmakers, considering key elements such as match outcomes, associated probabilities, odds, and bankroll dynamics. Bettors aim to maximize their utility by placing bets based on their estimated probabilities of match outcomes, while bookmakers strategically set odds to manage risk and ensure profitability. This agent-based framework enables a systematic analysis of decision-making processes in sports betting, balancing risk and reward. The complete list of notations can be found here A.

2.2 Mathematical Formalization of Sports Betting

2.2.1 Matches and Outcomes

At any given time $t \in \mathbb{R}^+$, we define the set of matches available for betting as:

$$\mathbb{M}(t) = \{m^1, m^2, \ldots, m^{M(t)}\}$$

where $M(t) \in \mathbb{N}$ represents the total number of matches available at time t.
For each match $m^k \in \mathbb{M}(t)$, there is a set of possible outcomes:

$$\Omega^k = \{\omega_1^k, \omega_2^k, \ldots, \omega_{N^k}^k\}$$

where $N^k \in \mathbb{N}$ represents the number of possible outcomes for match m^k.

Example: In a football match, possible outcomes might be chosen as {home team wins, draw, away team wins}, so $N^k = 3\ \forall k$.

2.2.2 Probabilities of Outcomes

We define $\mathbb{P}_Y(\omega_i^k)$ as the probability that outcome ω_i^k occurs for match m^k, given the state of the world Y at time t:

$$r_i^k(t) = \mathbb{P}_Y(\omega_i^k)$$

These probabilities may change over time as new information becomes available.
We introduce the random variable X_i^k associated with outcome ω_i^k:

$$X_i^k = \begin{cases} 1, & \text{if outcome } \omega_i^k \text{ occurs,} \\ 0, & \text{otherwise.} \end{cases}$$

Therefore, $r_i^k(t) = \mathbb{P}_Y(X_i^k = 1)$.

Example: Consider a football match m^k between Team A and Team B. The possible outcomes ω_i^k are:

$$\omega_1^k = \text{Team A wins}, \quad \omega_2^k = \text{Draw}, \quad \omega_3^k = \text{Team B wins}.$$

At time t, based on current information Y (such as form, injuries, and past results), the probabilities of these outcomes are:

$$r_1^k(t) = \mathbb{P}_Y(\text{Team A wins}), \quad r_2^k(t) = \mathbb{P}_Y(\text{Draw}), \quad r_3^k(t) = \mathbb{P}_Y(\text{Team B wins}).$$

For example, if $r_1^k(t) = 0.55$, it means there is a 55% chance that Team A will win.

2.2.3 Bettors and Bookmakers

Let $\mathbb{J}$ be the set of bettors, and $\mathbb{B}$ be the set of bookmakers.

Each bettor $J \in \mathbb{J}$ has a bankroll at time t, denoted by:

$$B_{\text{bettor}}^J(t)$$

Similarly, each bookmaker $B \in \mathbb{B}$ has a bankroll at time t, denoted by:

$$B_{\text{bookmaker}}^B(t)$$

2.2.4 Odds

At time t, bookmaker B offers odds on the outcomes of matches. For match m^k, the odds offered by bookmaker B are:

$$\mathbb{O}^k(B,t) = \{o_1^{k,B}(t), o_2^{k,B}(t), \ldots, o_{N^k}^{k,B}(t)\}$$

where $o_i^{k,B}(t)$ represents the odds offered on outcome ω_i^k of match m^k at time t.

Example: Consider the same football match m^k between Team A and Team B. At time t, bookmaker B offers the following odds:

$$\mathbb{O}^k(B,t) = \{2.00, 3.50, 4.00\}$$

Where $o_1^{k,B}(t) = 2.00$ for Team A to win, $o_2^{k,B}(t) = 3.50$ for a draw, and $o_3^{k,B}(t) = 4.00$ for Team B to win. These odds represent the potential payouts for each outcome.

2.2.5 Bets and Wagers

At time t, bettor J may choose to place bets on various outcomes. We define:

- $f_i^{k,J}(t)$: The fraction of bettor J's bankroll $B_{\text{bettor}}^J(t)$ that is wagered on outcome ω_i^k of match m^k.
- $b_i^{k,J}(t)$: The bookmaker B with whom bettor J places the bet on outcome ω_i^k of match m^k.

Therefore, the amount wagered by bettor J on outcome ω_i^k at time t is:

$$w_i^{k,J}(t) = f_i^{k,J}(t) \times B_{\text{bettor}}^J(t)$$

Example: Consider bettor J with a bankroll of $B^J_{\text{bettor}}(t) = 100$ units at time t. Bettor J decides to wager:

$$f_1^{k,J}(t) = 0.2 \quad \text{(20\% of the bankroll on Team A to win)}$$

Thus, the amount wagered is:

$$w_1^{k,J}(t) = 0.2 \times 100 = 20 \text{ units}$$

Bettor J places the 20-unit bet with bookmaker B.

2.2.6 Bankroll Evolution

The evolution of the bettors' and bookmakers' bankrolls depends on the outcomes of the matches and the settlement of bets.

Bettor's Bankroll Evolution

The bankroll of bettor J at time t is given by:

$$B^J_{\text{bettor}}(t) = B^J_{\text{bettor}}(0) + \int_0^t \sum_{b \in \mathcal{B}^J_{\text{settled}}(\tau)} G^J_{\text{bettor}}(b)\, d\tau$$

where:

- $\mathcal{B}^J_{\text{settled}}(s)$ is the set of bets placed by bettor J that are settled at time s.
- $G^J_{\text{bettor}}(b)$ is the gain or loss from bet b, calculated as:

$$G^J_{\text{bettor}}(b) = w^J(b) \times \left(o^B(b) \times X(b) - 1\right)$$

  - $w^J(b)$ is the amount wagered on bet b.
  - $o^B(b)$ is the odds offered by bookmaker B for bet b.
  - $X(b)$ indicates whether the bet was successful ($X(b) = 1$) or not ($X(b) = 0$).

Example: Consider bettor J starts with a bankroll of $B^J_{\text{bettor}}(0) = 100$ units. At time t_1, the bettor places a bet of $w^J(b) = 20$ units on a match with odds $o^B(b) = 2.50$ offered by bookmaker B.

If the outcome $X(b) = 1$ (the bettor wins the bet), the gain from the bet is:

$$G^J_{\text{bettor}}(b) = 20 \times (2.50 \times 1 - 1) = 30 \text{ units}$$

Thus, the updated bankroll at time t_1 is:

$$B^J_{\text{bettor}}(t_1) = 100 + 30 = 130 \text{ units}$$

If the bettor loses another bet at time t_2 with a wager of 30 units on odds of 3.00, then $X(b) = 0$ and the loss is:

$$G^J_{\text{bettor}}(b) = 30 \times (3.00 \times 0 - 1) = -30 \text{ units}$$

The bankroll at time t_2 becomes:

$$B^J_{\text{bettor}}(t_2) = 130 - 30 = 100 \text{ units}$$

Bookmaker's Bankroll Evolution

Similarly, the bankroll of bookmaker B at time t is given by:

$$B^B_{\text{bookmaker}}(t) = B^B_{\text{bookmaker}}(0) + \int_0^t \sum_{J \in \mathcal{J}} \sum_{b \in \mathcal{B}^{B,J}_{\text{settled}}(\tau)} G^B_{\text{bookmaker}}(b)\, d\tau$$

where:

- $\mathcal{J}$ is the set of all bettors $\{J_1, J_2, \ldots, J_N\}$ placing bets with bookmaker B.
- $\mathcal{B}^{B,J}_{\text{settled}}(s)$ is the set of bets accepted by bookmaker B from bettor J that are settled at time s.
- $G^B_{\text{bookmaker}}(b)$ is the gain or loss from bet b, which now takes into account multiple bettors J, calculated as:

  $$G^B_{\text{bookmaker}}(b) = w^J(b) \times \left(1 - o^B(b) \times X(b)\right)$$

  where:

  - $w^J(b)$ is the amount wagered by bettor J on bet b.
  - $o^B(b)$ is the odds offered by bookmaker B for bet b.
  - $X(b)$ indicates whether the bet was successful ($X(b) = 1$) or not ($X(b) = 0$).

Impact of Multiple Bettors

For each bet b, the gain or loss for bookmaker B depends on which bettor placed the bet. If bettor J wins, bookmaker B pays out, and if bettor J loses, bookmaker B gains:

$$G^B_{\text{bookmaker}}(b) = -G^J_{\text{bettor}}(b)$$

Thus, for each bet placed by a bettor J, the bookmaker's gain is equal to the bettor's loss, and vice versa. With multiple bettors, the bookmaker's bankroll reflects the combined gains and losses from all bets settled across the bettors $J_1, J_2, \ldots, J_N$.

2.2.7 Bankroll Factor

To abstract from the initial bankroll amounts, we can define the *Bankroll Factor* for bettors and bookmakers.

Bettor's Bankroll Factor

The bankroll factor for bettor J at time t is defined as:

$$BF^J_{\text{bettor}}(t) = \frac{B^J_{\text{bettor}}(t)}{B^J_{\text{bettor}}(0)}$$

This represents the growth of the bettor's bankroll relative to their initial bankroll.

Bookmaker's Bankroll Factor

Similarly, the bankroll factor for bookmaker B at time t is:

$$BF^B_{\text{bookmaker}}(t) = \frac{B^B_{\text{bookmaker}}(t)}{B^B_{\text{bookmaker}}(0)}$$

2.2.8 Gain Calculation

The cumulative gain for bettor J up to time t is:

$$G^J_{\text{bettor}}(t) = B^J_{\text{bettor}}(t) - B^J_{\text{bettor}}(0) = B^J_{\text{bettor}}(0)\left(BF^J_{\text{bettor}}(t) - 1\right)$$

Similarly, for bookmaker B:

$$G^B_{\text{bookmaker}}(t) = B^B_{\text{bookmaker}}(t) - B^B_{\text{bookmaker}}(0) = B^B_{\text{bookmaker}}(0)\left(BF^B_{\text{bookmaker}}(t) - 1\right)$$

2.2.9 Utility Function

The utility function U represents the agent's preferences regarding risk and reward, crucial in decision-making under uncertainty [17]. Bettors and bookmakers use this function to optimize their gains over time while minimizing risk. Unlike expected returns, utility functions incorporate risk preferences, allowing agents to balance the trade-off between potential gains and variability [20] [2] [23].

Forms of Utility Functions

Different utility functions capture varying risk attitudes, ranging from risk-neutral to risk-averse behaviors. Below are the common types of utility functions in the betting market:

1. Expected Value Utility (Risk-Neutral) The simplest form, where utility is directly proportional to wealth:

$$U(B) = B$$

Agents using this function are risk-neutral, focusing solely on maximizing expected returns without considering risk.

2. Logarithmic Utility (Moderate Risk Aversion) Logarithmic utility models constant relative risk aversion (CRRA) and is expressed as:

$$U(B) = \ln(B)$$

This function reflects diminishing marginal utility of wealth, balancing risk and reward, commonly used in the Kelly Criterion [18] [25] for long-term growth.

3. Power Utility (CRRA) A generalization of logarithmic utility, with risk aversion controlled by γ:

$$U(B) = \frac{B^{1-\gamma}}{1-\gamma}, \quad \gamma \neq 1$$

Higher γ values indicate greater risk aversion. When $\gamma = 1$, the function becomes logarithmic.

4. Exponential Utility (Constant Absolute Risk Aversion - CARA) The exponential utility models constant absolute risk aversion (CARA):

$$U(B) = -e^{-\alpha B}$$

Here, α controls risk aversion. Agents using this function maintain consistent risk preferences regardless of wealth level.

5. Quadratic Utility Quadratic utility is given by:

$$U(B) = B - \frac{\lambda}{2}B^2$$

Though it captures increasing risk aversion, it has the drawback of implying decreasing utility at higher wealth levels, making it less commonly used.

Implications of Different Utility Functions

Each utility function models specific risk preferences, influencing the agent's decisions:

Risk-Neutral Behavior Agents with linear utility ($U(B) = B$) focus solely on maximizing returns, indifferent to risk. This behavior is rare in practice due to the inherent risks in betting.

Risk-Averse Behavior Utility functions like logarithmic, power, and exponential represent risk-averse behavior:

- **Logarithmic Utility:** Moderate risk aversion, favoring long-term growth.
- **Power Utility (CRRA):** Flexibility in modeling different degrees of risk aversion via γ.
- **Exponential Utility (CARA):** Constant risk aversion regardless of wealth.

Risk-Seeking Behavior Agents may occasionally exhibit risk-seeking behavior, favoring higher variance. This is typically modeled by utility functions with convex regions or negative coefficients but is unsustainable in the long term.

Choosing an Appropriate Utility Function

Selecting the right utility function depends on:

- **Risk Preference:** It should reflect the agent's risk tolerance.
- **Mathematical Tractability:** Functions like logarithmic utility offer simpler analytical solutions.
- **Realism:** The chosen function should realistically model the agent's behavior in the market.

2.3 General Agent-Based Betting Framework

In order to model the decision-making processes of bettors and bookmakers in sports betting, we adopt a general agent-based framework [11]. This framework allows us to formalize the interactions between agents (bettors and bookmakers) and the environment (the sports betting market) in a comprehensive and systematic manner. By defining the state space, action space, and other essential components in the most general terms, we can capture the complexity of sports betting and lay the groundwork for more specific analyses.

2.3.1 Agents in the Betting Market

There are two primary types of agents in the sports betting market:

- **Bettors (Players):** Individuals or entities who place bets on the outcomes of sporting events with the aim of maximizing their returns.
- **Bookmakers:** Organizations or individuals who offer betting opportunities by setting odds on the possible outcomes of sporting events, aiming to maximize their profits.

Each agent operates based on their own objectives, information, and strategies, interacting with the environment and other agents through their actions.

2.3.2 State Space

At any given time $t \in \mathbb{R}^+$, the state of the sports betting environment, denoted by $S(t)$, encompasses all the information relevant to the agents' decision-making processes. The state space $\mathcal{S}$ is the set of all possible states $S(t)$.

The state $S(t)$ can be defined as:

$$S(t) = (\mathbb{M}(t), \Omega(t), \mathbb{O}(t), B_{\text{bettor}}(t), B_{\text{bookmaker}}(t), H(t), \mathcal{I}(t))$$

where:

- $\mathbb{M}(t)$: The set of all matches available at time t.
- $\Omega(t)$: The set of possible outcomes for each match in $\mathbb{M}(t)$.
- $\mathbb{O}(t)$: The set of odds offered by bookmakers for each possible outcome at time t.
- $B_{\text{bettor}}(t)$: The set of bettors' bankrolls at time t.
- $B_{\text{bookmaker}}(t)$: The set of bookmakers' bankrolls at time t.
- $H(t)$: The history of past events up to time t, including past bets, match results, and odds movements.
- $\mathcal{I}(t)$: Any additional information available to the agents at time t, such as team news, player injuries, weather conditions, etc.

The state $S(t)$ encapsulates all the variables that can influence the agents' decisions, making it comprehensive and general.

2.3.3 Action Space

At each time t, agents choose actions from their respective action spaces:

Bettors' Action Space

The action space for a bettor J at time t, denoted by $\mathcal{A}^J_{\text{bettor}}(t)$, consists of all possible betting decisions they can make. An action $A^J_{\text{bettor}}(t) \in \mathcal{A}^J_{\text{bettor}}(t)$ can be defined as:

$$A^J_{\text{bettor}}(t) = \left\{ \left(f_i^k\right) \mid f_i^k \in [0,1], \sum_{i,k} f_i^k <= 1 \right\}$$

where:

- f_i^k: The fraction of the bettor's bankroll $B^J_{\text{bettor}}(t)$ to wager on outcome ω_i^k.

Hence, the bettor chose the outcomes to bet on by assigning 0 (no bet) or more to an outcome at a given time t.

Bookmakers' Action Space

The action space for a bookmaker B at time t, denoted by $\mathcal{A}^B_{\text{bookmaker}}(t)$, can be simplified to the selection of odds for each outcome. An action $A^B_{\text{bookmaker}}(t) \in \mathcal{A}^B_{\text{bookmaker}}(t)$ is defined as:

$$A^B_{\text{bookmaker}}(t) = \left\{ \mathbb{O}^k(B,t) = \{o_i^k \mid o_i^k \in [1, \infty) \right\}$$

where:

- o_i^k: The odds set by the bookmaker B for outcome ω_i^k of match m^k at time t.

If $o_i^k = 1$, the bookmaker does not offer bets on outcome ω_i^k. If all odds $o_i^k = 1$ for a match m^k, the bookmaker does not offer that match for betting.

Example: At time t, bettor J allocates fractions of their 100 unit bankroll across two matches, with three possible outcomes:

$$f = \begin{pmatrix} 0.3 & 0.2 & 0 \\ 0.5 & 0 & 0 \end{pmatrix}$$

The bookmaker sets the following odds for each outcome:

$$o = \begin{pmatrix} 2.50 & 3.00 & 4.00 \\ 1.80 & 2.90 & 3.50 \end{pmatrix}$$

This means bettor J wagers 30 units on ω_1^1 (Team A wins m^1), 20 units on ω_2^1 (draw in m^1), and 50 units on ω_1^2 (Team A wins m^2).

2.3.4 Transition Dynamics

The state transitions $\frac{dS(t)}{dt}$ are governed by the interactions between the agents' actions and the environment. The transition dynamics can be described in general terms:

$$\frac{dS(t)}{dt} = \Phi\left(S(t), A_{\text{bettor}}(t), A_{\text{bookmaker}}(t), \epsilon(t)\right)$$

where:

- Φ is the state transition function.
- $A_{\text{bettor}}(t)$: The set of all bettors' actions at time t.
- $A_{\text{bookmaker}}(t)$: The set of all bookmakers' actions at time t.
- $\epsilon(t)$: Represents the stochastic elements inherent in sports outcomes and market dynamics, modeled as random variables.

The transition function Φ captures how the state evolves due to:

- The resolution of matches (outcomes becoming known), represented by changes in outcome variables over time..
- The settlement of bets (adjustment of bettors' and bookmakers' bankrolls).
- Changes in available matches and odds for the next time period.
- Updates to the history $H(t)$ and information set $\mathcal{I}(t)$, represented by $\frac{dH(t)}{dt}$ and $\frac{d\mathcal{I}(t)}{dt}$.

2.3.5 Policies

Each agent follows a policy that guides their decision-making process:

Bettors' Policy

A bettor's policy π^J_{bettor} is a mapping from states to actions:

$$\pi^J_{\text{bettor}} : \mathcal{S} \to \mathcal{A}^J_{\text{bettor}}$$

The policy determines how the bettor decides on which bets to place and how much to wager, based on the current state $S(t)$.

Bookmakers' Policy

A bookmaker's policy $\pi^B_{\text{bookmaker}}$ is a mapping from states to actions:

$$\pi^B_{\text{bookmaker}} : \mathcal{S} \to \mathcal{A}^B_{\text{bookmaker}}$$

The policy dictates how the bookmaker sets odds and offers betting opportunities, considering factors like market demand, risk management, and competitive positioning.

2.3.6 Objectives and Constraints

Each agent aims to optimize an objective function over time, such as maximizing expected utility or profit, subject to specific constraints that reflect their operational limitations and risk management considerations.

Bettors' Objective

The bettor seeks to maximize a chosen utility over a time horizon T:

$$\max_{\pi^J_{\text{bettor}}} \quad \mathbb{E}\left[U^J\left(BF^J_{\text{bettor}}(T)\right)\right]$$

Constraints for the Bettor

The bettor's optimization problem is subject to the following mathematical constraints:

- 1. Budget Constraint at Each Time t:

  The total fraction of the bankroll wagered on all outcomes cannot exceed 1 at any time t:

  $$\sum_{k=1}^{M(t)} \sum_{i=1}^{N^k} f_i^{k,J}(t) \leq 1 \quad \forall t$$

  where:

  - $f_i^{k,J}(t)$ is the fraction of the bettor J's bankroll $BF^J_{\text{bettor}}(t)$ wagered on outcome i of match k at time t.
  - $M(t)$ is the total number of matches available at time t.
  - N^k is the number of possible outcomes for each match k.

- 2. Non-Negativity of Wager Fractions:

  The bettor cannot wager negative fractions of the bankroll:

  $$f_i^{k,J}(t) \geq 0 \quad \forall i, k, t$$

Bookmakers' Objective

The bookmaker aims to maximize a chosen utility over a time horizon T:

$$\max_{\pi^B_{\text{bookmaker}}} \quad \mathbb{E}\left[U^B\left(BF^B_{\text{bookmaker}}(T)\right)\right]$$

Constraints for the Bookmaker

The bookmaker's optimization problem is subject to the following mathematical constraints:

- 1. Liquidity Constraint:

  The bookmaker must ensure sufficient funds to cover potential payouts:

  $$BF^B_{\text{bookmaker}}(t) \geq \text{Maximum Potential Liability at } t$$

  This ensures that the bookmaker's bankroll at time t is greater than or equal to the maximum possible payout based on the accepted bets.

- 2. Odds Setting Constraints:

  The odds must be set to ensure profitability and competitiveness:

– Overround Constraint (Bookmaker's Margin):
  For each match k, the sum of the implied probabilities must exceed 1:

$$\sum_{i=1}^{N^k} \frac{1}{o_i^k(t)} = 1 + \epsilon^k(t) \quad \forall k, t$$

  Here, $\epsilon^k(t) > 0$ represents the bookmaker's margin for match k at time t.
– Margin Bound:
  To balance profitability and competitiveness, we impose the following bound on $\epsilon^k(t)$:

$$\epsilon_{\min} \leq \epsilon^k(t) \leq \epsilon_{\max} \quad \forall k, t$$

  This ensures that the margin $\epsilon^k(t)$ stays within a specified range, keeping the odds competitive enough to attract bettors while securing a minimum margin for profitability.
– Competitive Odds Constraint:
  The odds $o_i^k(t)$ must remain competitive, influenced by market averages or competitors' odds. Therefore, the bookmaker may aim to keep $\epsilon^k(t)$ as low as possible while maintaining profitability and covering risk.

2.4 Reduction to the Studied Case

Building upon the general agent-based betting framework, we aim to simplify the agent-based betting framework and reduce computational complexity. We transition from a dynamic to a static optimization model by introducing key assumptions. By assuming immediate resolution of bets and the absence of intertemporal dependencies—where current decisions do not influence future opportunities—we make the static and dynamic problems effectively equivalent for our purposes. This simplification allows us to optimize agents' decisions at each time step independently, facilitating the derivation of optimal solutions without the need for complex dynamic programming. However, this reduction comes at a cost, notably in terms of long-term interpretability, as the model no longer accounts for cumulative effects and evolving dynamics over time.

2.4.1 Hypotheses for the Constrained Problem

1. **No Intertemporal Dependencies (Additive Utility Function):** Utility is additive over time, meaning decisions at time t do not affect future periods. The agent maximizes utility independently at each step, simplifying the problem into sequential sub-problems.

   Reason: This eliminates the need to account for future wealth in current decisions, reducing complexity.

2. **Discrete Time Steps:** Time is divided into discrete intervals where decisions are made periodically. Bets are resolved by the end of each period before moving to the next. $t = 0, 1, 2, \ldots, T$

   Reason: Discrete time steps reduce the dynamic problem to a series of static decisions, simplifying optimization.

3. **Non-Overlapping Bets:** Bets are settled within the same period, ensuring that wealth at the end of each period is fully available for the next, avoiding unresolved wagers impacting future decisions.

   Reason: This ensures no carryover of unresolved bets, keeping each period's wealth independent.

4. **Independence of Match Outcomes:** Match outcomes are independent random events, meaning there is no correlation between the results of different matches.

   Reason: This simplifies probability calculations by eliminating the need to model inter-match dependencies.

5. **Static Information Environment:** Information is fixed within each period. No new data arrives mid-period, and updates are considered only in the next time step.

   Reason: A static environment avoids real-time strategy adjustments, making the problem more manageable.

These assumptions significantly simplify the model by reducing the complexity inherent in a dynamic optimization problem, but they also modify or limit certain long-term interpretations, such as how future wealth or intertemporal risk is managed across multiple betting periods.

2.4.2 Simplification of the Utility Maximization Problem

With no overlapping bets and a static information environment, agents do not need to consider how current actions might affect future opportunities or states. This myopic decision-making approach allows agents to focus solely on the current time period, simplifying their optimization problem to a static one.

Hence, the agents' objective functions depend only on the current wealth and the outcomes of bets placed in the current period. The expected utility maximization problem at each time t becomes:

For bettors:

$$\max_{\{f_i^{k,J}(t)\}} \quad \mathbb{E}\left[U\left(B^J_{\text{bettor}}(t+1)\right) \mid S(t)\right]$$

For bookmakers:

$$\max_{\{o_i^k(t)\}} \quad \mathbb{E}\left[U\left(B_{\text{bookmaker}}(t+1)\right) \mid S(t)\right]$$

where $S(t)$ is the state at time t, which includes the available matches, odds, and the agents' current bankrolls.

2.4.3 Dynamic and Total Utility under Assumptions

In our framework, under the assumption of discrete time steps and no intertemporal dependencies, the total utility across all periods T is given by the sum of the static utilities at each time step:

$$U_{\text{total}} = \sum_{t=1}^{T} U(B(t)).$$

This assumes that decisions are made independently at each t, with the utility depending solely on the wealth $B(t)$ at that moment. Additive utility functions, such as $U(B) = B$, respect this assumption directly, meaning maximizing the utility at each step also maximizes total utility.

However, logarithmic and exponential utilities do not preserve a simple additive structure due to risk preferences that influence future decisions. While linear utility maintains additivity, $U(B) = \ln(B)$ and $U(B) = -e^{-\alpha B}$ do not.

Utility Functions Respecting Additivity

- Linear utility: $U(B) = B$

Utility Functions Not Respecting Additivity

- Logarithmic utility: $U(B) = \ln(B)$
- Exponential utility: $U(B) = -e^{-\alpha B}$
- CRRA: $U(B) = \frac{B^{1-\gamma}}{1-\gamma}, \quad \gamma \neq 1$
- Quadratic utility: $U(B) = B - \frac{\lambda}{2}B^2$

Approximation with Additive Properties

By using a first-order Taylor expansion for $\ln(B)$ or $-e^{-\alpha B}$, these utilities can become approximately additive. For small deviations around B, we approximate:

$$\ln(B) \approx \ln(B_0) + \frac{B - B_0}{B_0}, \quad -e^{-\alpha B} \approx -e^{-\alpha B_0} + \alpha e^{-\alpha B_0}(B - B_0)$$

These approximations are linear in B, making the utility functions additive for small changes in wealth. Under these assumptions, the complexity of the problem is reduced, allowing the use of simpler optimization techniques without fully abandoning the original utility structure.

Non-Additive Utility Maximization and Long-Term Interpretation

When maximizing non-additive utility functions (such as logarithmic or exponential) at each step t, the interpretation of utility over the entire period T changes. Unlike additive functions, where the total utility is simply the sum of the utilities at each time step, non-additive functions induce a more complex relationship between short-term and long-term behavior.

For non-additive utilities, maximizing utility at each step does not guarantee maximization of the utility across the entire period. The decisions made at each step can interact non-linearly across time, meaning that the long-term growth or risk profile may differ significantly from the one-step behavior. This highlights the difference between local (step-by-step) optimization and the global impact over the entire period.

Interpretation of Log Utility in Terms of Long-Term Geometric Growth

Maximizing the logarithmic utility at each time step involves maximizing the expected utility:

$$\max_{f(t)} \mathbb{E}\left[\ln B_{\text{agent}}(t+1) \,\middle|\, \mathcal{F}_t\right],$$

where $B_{\text{agent}}(t+1)$ is the wealth at time $t+1$, $f(t)$ represents the decision variables at time t, and $\mathcal{F}_t$ denotes the information available at time t.

The total utility over T periods is given by:

$$U_{\text{total}} = \sum_{t=1}^{T} \ln B_{\text{agent}}(t) = \ln\left(\prod_{t=1}^{T} B_{\text{agent}}(t)\right).$$

Taking the expectation of the total utility, we have:

$$\mathbb{E}[U_{\text{total}}] = \mathbb{E}\left[\ln\left(\prod_{t=1}^{T} B_{\text{agent}}(t)\right)\right].$$

However, due to the concavity of the logarithm and the properties of expectations, we cannot simplify this expression to $\ln\left(\prod_{t=1}^{T} \mathbb{E}[B_{\text{agent}}(t)]\right)$ unless the $B_{\text{agent}}(t)$ are deterministic. The expected value of the logarithm of a product of random variables is not equal to the logarithm of the product of their expectations.

To interpret $\mathbb{E}[U_{\text{total}}]$ in terms of expected wealth and variance, we can use a second-order Taylor expansion of the logarithm around $\mathbb{E}[B_{\text{agent}}(t)]$:

$$\mathbb{E}[\ln B_{\text{agent}}(t)] \approx \ln \mathbb{E}[B_{\text{agent}}(t)] - \frac{1}{2} \frac{\mathbb{V}\text{ar}[B_{\text{agent}}(t)]}{\left(\mathbb{E}[B_{\text{agent}}(t)]\right)^2}.$$

Summing over T periods, we obtain:

$$\mathbb{E}[U_{\text{total}}] \approx \sum_{t=1}^{T} \left(\ln \mathbb{E}[B_{\text{agent}}(t)] - \frac{1}{2} \frac{\mathbb{V}\text{ar}[B_{\text{agent}}(t)]}{\left(\mathbb{E}[B_{\text{agent}}(t)]\right)^2}\right).$$

This approximation shows that the expected total utility depends on both the expected wealth and the variance at each time step. The logarithmic utility function captures the trade-off between expected wealth growth and risk (variance), penalizing volatility and favoring steady growth.

Over the long term, maximizing the expected logarithmic utility leads to maximizing the **expected logarithm of cumulative wealth**, which corresponds to maximizing the **geometric mean return**. This strategy ensures that wealth grows at the highest possible geometric rate, accounting for both returns and risks.

Long-Term Interpretation of Exponential Utility

For the exponential utility function $U(B) = -e^{-\alpha B}$, where $\alpha > 0$ is the coefficient of absolute risk aversion, the total utility over T periods is:

$$U_{\text{total}} = \sum_{t=1}^{T} U(B(t)) = -\sum_{t=1}^{T} e^{-\alpha B(t)}.$$

Taking the expectation, we have:

$$\mathbb{E}[U_{\text{total}}] = -\sum_{t=1}^{T} \mathbb{E}\left[e^{-\alpha B(t)}\right].$$

We cannot simplify $\mathbb{E}\left[e^{-\alpha B(t)}\right]$ without specifying the distribution of $B(t)$. However, using a second-order Taylor expansion around $\mathbb{E}[B(t)]$:

$$\mathbb{E}\left[e^{-\alpha B(t)}\right] \approx e^{-\alpha \mathbb{E}[B(t)]}\left(1 + \frac{\alpha^2}{2}\mathbb{V}\text{ar}[B(t)]\right).$$

Therefore, the expected total utility becomes:

$$\mathbb{E}[U_{\text{total}}] \approx -\sum_{t=1}^{T} e^{-\alpha \mathbb{E}[B(t)]}\left(1 + \frac{\alpha^2}{2}\mathbb{V}\text{ar}[B(t)]\right).$$

This expression highlights that the expected utility depends heavily on both the expected wealth and the variance. As α increases, the variance term becomes more significant, reinforcing the agent's aversion to risk. The exponential utility function thus focuses on **risk minimization** and **capital preservation** over wealth maximization.

Long-Term Interpretation of Mean-Variance Utility

For the mean-variance utility, which can be associated with a quadratic utility function $U(B) = B - \frac{\lambda}{2}B^2$ for small variations in B, the expected utility at each time step is:

$$\mathbb{E}[U(B(t))] = \mathbb{E}[B(t)] - \frac{\lambda}{2}\mathbb{E}[B(t)^2].$$

Assuming that $\mathbb{E}[B(t)^2] = (\mathbb{E}[B(t)])^2 + \mathbb{V}\text{ar}[B(t)]$, we have:

$$\mathbb{E}[U(B(t))] = \mathbb{E}[B(t)] - \frac{\lambda}{2}\left((\mathbb{E}[B(t)])^2 + \mathbb{V}\text{ar}[B(t)]\right).$$

Over T periods, the expected total utility is:

$$\mathbb{E}[U_{\text{total}}] = \sum_{t=1}^{T} \mathbb{E}[U(B(t))].$$

Simplifying, we obtain:

$$\mathbb{E}[U_{\text{total}}] = \sum_{t=1}^{T}\left(\mathbb{E}[B(t)] - \frac{\lambda}{2}\left((\mathbb{E}[B(t)])^2 + \mathbb{V}\text{ar}[B(t)]\right)\right).$$

This expression demonstrates that the agent considers both the expected wealth and the variance, with the parameter λ controlling the trade-off between maximizing returns and minimizing risk.

2.4.4 Simplification of State Transitions

The agents' state variables, particularly their bankrolls, evolve in a straightforward manner without considering future uncertainties or pending bets. The bankroll update equations become:

$$B_{\text{bettor}}(t+1) = B_{\text{bettor}}(t) + G_{\text{bettor}}(t)$$

$$B_{\text{bookmaker}}(t+1) = B_{\text{bookmaker}}(t) + G_{\text{bookmaker}}(t)$$

where $G_{\text{bettor}}(t)$ and $G_{\text{bookmaker}}(t)$ represent the gains or losses realized from bets placed and settled within time t.

2.4.5 Detailed Simplification of the Bookmaker's Problem

Similarly, the bookmaker's optimization problem simplifies under the assumptions:

Objective Function:

$$\max_{\{o_i^k(t)\}} \quad U_{\text{bookmaker}}(t) = \mathbb{E}\left[U\left(B_{\text{bookmaker}}(t) + G_{\text{bookmaker}}(t)\right) \mid S(t)\right]$$

Constraints:

$$BF^B_{\text{bookmaker}}(t) \geq \text{Maximum Potential Liability at } t$$

$$\sum_{i=1}^{I} \frac{1}{o_i^k(t)} = 1 + \epsilon^k(t) \quad \text{for all } k, t$$

Variables:

- $o_i^k(t)$: Odds set for outcome i of match k at time t.
- $G_{\text{bookmaker}}(t)$: Gain or loss from bets, calculated based on the total bets received and payouts made in the current period.
- $\epsilon^k(t)$: Margin for each match at every time step that the bookmaker set to maximise attractiveness, minimize risque and maximize pay off.

2.4.6 Reasons for the Simplifications

We introduce these simplifications for several important reasons:

Reducing Computational Complexity

Dynamic optimization problems, especially those involving stochastic elements and intertemporal dependencies, can be highly complex and computationally intensive. By simplifying the problem to a static one, we make it more tractable and amenable to analytical or numerical solutions.

Simplifying the Use of Historical Odds

Solving the general dynamic optimization problem requires a sufficiently large history of odds at each time step t to ensure convergence towards an optimal solution. This includes tracking all relevant historical data for each time step and state S. By reducing the problem to a static case, the need for such an extensive history is eliminated, as the model only relies on current odds. This simplification significantly reduces computational complexity while maintaining the core of the decision-making process.

Facilitating Analytical Derivations

With the assumptions of immediate bet resolution and independence, we can derive closed-form solutions or straightforward algorithms for optimal betting strategies, such as the Kelly Criterion for bettors using logarithmic utility functions.

Focusing on Core Decision-Making Principles

The simplifications allow us to isolate and analyze the fundamental principles of optimal betting and odds setting without the confounding effects of dynamic interactions. This clarity helps in understanding the key factors that influence agents' decisions in the sports betting market.

2.4.7 Limitations of the Simplified Model

While the simplifications make the model more manageable, they also introduce limitations that should be acknowledged:

1. **Hypothesis: Additive Utility Function with No Intertemporal Dependencies**
   - **Domain of Validity:** Valid when agents focus solely on immediate wealth without concern for future utility.
   - **Limitation with Reality:** Agents usually consider future wealth and utility; this assumption ignores long-term planning and risk preferences.
   - **Risk:** Ignoring intertemporal effects may result in strategies that maximize short-term gains at the expense of long-term wealth, increasing the risk of ruin or failing to achieve overall financial objectives. Among the utility functions described, only $U(B) = B$ is additive with time.
2. **Hypothesis: Discrete Time Steps**
   - **Domain of Validity:** Applicable when betting decisions are made at fixed, regular intervals.
   - **Limitation with Reality:** Real betting markets operate continuously; opportunities and information arise at any time, making this assumption somewhat unrealistic.
   - **Risk:** By assuming discrete time steps, we risk missing profitable opportunities that occur between intervals and fail to capture the continuous dynamics of the market, leading to suboptimal strategies.
3. **Hypothesis: Non-Overlapping Time Steps**
   - **Domain of Validity:** Valid when all bets are short-term and resolved within the same period.
   - **Limitation with Reality:** In practice, many bets span multiple periods, and unresolved bets can impact future wealth and decisions; this assumption is restrictive.
   - **Risk:** Ignoring overlapping bets may lead to underestimating risk exposure and mismanaging bankrolls, potentially resulting in unexpected losses or liquidity issues.
4. **Hypothesis: Independence of Match Outcomes**
   - **Domain of Validity:** Appropriate when matches are truly independent events without any influence on each other.
   - **Limitation with Reality:** In reality, match outcomes can be correlated due to common factors; this simplification overlooks potential dependencies.
   - **Risk:** Assuming independence when correlations exist can lead to inaccurate probability assessments and risk underestimation, possibly causing overbetting on correlated outcomes and increasing the chance of significant losses.
5. **Hypothesis: Static Information Environment**
   - **Domain of Validity:** Suitable for very short periods where no new information is expected to arrive.
   - **Limitation with Reality:** Information flows continuously in real markets; ignoring new information is unrealistic and limits strategic adjustments.
   - **Risk:** By not accounting for new information, we risk making decisions based on outdated data, leading to poor betting choices and missed opportunities to adjust strategies in response to market changes.

2.4.8 Conclusion

By adhering to the constraints imposed by these hypotheses, we effectively narrow the search space, making it easier to find an optimal solution for our simplified problem. However, it's important to note that the first hypothesis —assuming an additive utility function with no intertemporal dependencies— will not be applied (in every case) in our model. As a result, the optimal solution we derive will differ -if using a non additive utility- from the true optimal solution for the general (using the same utility function), constrained problem under the four next assumptions.

2.5 Incorporating Estimated Probabilities in Betting Strategies

In the real-world sports betting environment, both bettors and bookmakers do not have access to the true probabilities of match outcomes. Instead, they rely on their own estimations based on available information, statistical models, expert opinions, and other predictive tools. These estimated probabilities often differ from the true underlying probabilities and can vary between bettors and bookmakers due to differences in information, analysis techniques, and biases.

This section introduces the concept of estimated probabilities for match outcomes as perceived by bettors and bookmakers, explains the necessity of considering these estimates in modeling betting strategies, and provides analytical derivations for expected gain and variance incorporating these estimated probabilities. We also explore how these differences influence optimal betting strategies, particularly through the application of the Kelly Criterion.

2.5.1 Estimated Probabilities

Let:

- r_i^k: The true probability of outcome ω_i^k occurring in match m^k.
- $p_i^{k,J}$: The probability estimate of outcome ω_i^k as perceived by bettor J.
- $p_i^{k,B}$: The probability estimate of outcome ω_i^k as perceived by bookmaker B.

Due to the inherent uncertainty and complexity of predicting sports outcomes, the estimated probabilities $p_i^{k,J}$ and $p_i^{k,B}$ generally differ from the true probabilities r_i^k and from each other. These discrepancies are critical in the betting market because they create opportunities for bettors to find value bets (situations where they believe the bookmaker's odds underestimate the true likelihood of an outcome) and for bookmakers to manage their risk and profit margins.

2.5.2 Utility Maximization and the Role of Estimated Probabilities

The bettor aims to maximize their expected utility, which is influenced by both the expected value and the variance of the bankroll factor. The utility function U encapsulates the bettor's risk preferences.

Expected Utility in Terms of Bankroll Factor

The expected utility at time $t+1$ is given by:

$$\mathbb{E}_{p^J}\left[U\left(BF_{\text{bettor}}(t+1)\right)\right] = \sum_{\text{all outcomes}} U\left(BF_{\text{bettor}}(t+1)\right) \times \text{Probability of outcomes}$$

To compute this expectation, the bettor must consider all possible combinations of match outcomes, weighted by their estimated probabilities $p_{i_k}^{k,J}$. This requires:

- Knowledge of $p_{i_k}^{k,J}$ for each outcome i_k in match k.
- Calculation of $BF_{\text{bettor}}(t+1)$ for each possible combination of outcomes.

Assuming there are M matches at time t to bet on, each with $N(k)$ possible outcomes, the expected utility expands to:

$$\mathbb{E}_{p^J}\left[U\left(BF_{\text{bettor}}(t+1)\right)\right] = \sum_{i_1=1}^{N(1)} \sum_{i_2=1}^{N(2)} \cdots \sum_{i_M=1}^{N(M)} U\left(BF_{\text{bettor}}^{(i_1,i_2,\ldots,i_M)}(t+1)\right) \times \prod_{k=1}^{M} p_{i_k}^{k,J}$$

Where:

- i_k indexes the outcome of match k.
- $BF_{\text{bettor}}^{(i_1,i_2,\ldots,i_M)}(t+1)$ is the bankroll factor after all matches, given the outcomes $i_1, i_2, \ldots, i_M$.

- $p_{i_k}^{k,J}$ is the estimated probability of outcome i_k for match k.
- The product $\prod_{k=1}^{M} p_{i_k}^{k,J}$ represents the joint probability of the specific combination of outcomes, assuming independence between matches.

For each outcome combination $(i_1, i_2, \ldots, i_M)$, the bankroll factor is calculated as:

$$BF_{\text{bettor}}^{(i_1,i_2,\ldots,i_M)}(t+1) = BF_{\text{bettor}}(t) \times \left(1 + \sum_{k=1}^{M} f_{k,o_{i_k}} \left(o_{k,o_{i_k}} - 1\right)\right)$$

Where:

- $f_{k,o_{i_k}}$ is the fraction of the bankroll wagered on outcome o_{i_k} in match k.
- $o_{k,o_{i_k}}$ is the odds offered by the bookmaker for outcome o_{i_k} in match k.

An analytic simplification demonstration for the Kelly criteria, $u = ln$, can be found at the end of this work B.

Importance of Accurate Probability Estimates

The bettor's decisions hinge on their estimated probabilities. Inaccurate estimates can lead to sub-optimal betting strategies:

- **Overestimation** of probabilities may cause the bettor to wager too much, increasing the risk of significant losses.
- **Underestimation** may result in conservative wagers, leading to missed opportunities for profit.

By accurately estimating the probabilities, the bettor can better align their strategy with their utility function, optimizing the trade-off between expected return and risk.

2.5.3 Expected Bankroll Factor

The expected value $\mathbb{E}$ of the bankroll factor BF corresponds to a simple utility function $U(B) = B$, representing a risk-neutral perspective. This expected value is crucial in understanding the growth of wealth without considering risk preferences. An analytic form for this expectation can be derived straightforwardly.

The expected bankroll factor at time $t+1$ incorporates the bettor's actions and the estimated probabilities of outcomes. The evolution of the bankroll factor from time t to $t+1$ is given by:

$$BF_{\text{bettor}}^{J}(t+1) = BF_{\text{bettor}}^{J}(t) \left[1 + \sum_{k=1}^{M} \sum_{i=1}^{N^k} f_i^{k,J}(t) \left(o_i^{k,B}(t) X_i^k - 1\right)\right]$$

Here, X_i^k is an indicator variable that equals 1 if outcome ω_i^k occurs and 0 otherwise. The term inside the square brackets represents the return on the bettor's wagers during time t.

Calculating Expected Bankroll Factor

To find the expected bankroll factor at time $t+1$, we take the expectation with respect to the bettor's estimated probabilities $p_i^{k,J}$:

$$\mathbb{E}_{p^J}\left[BF_{\text{bettor}}^{J}(t+1)\right] = BF_{\text{bettor}}^{J}(t) \left[1 + \sum_{k=1}^{M} \sum_{i=1}^{N^k} f_i^{k,J}(t) \left(o_i^{k,B}(t) p_i^{k,J} - 1\right)\right]$$

This expression shows that the expected growth of the bettor's bankroll factor depends on:

- The fraction of the bankroll wagered $f_i^{k,J}(t)$.
- The odds offered $o_i^{k,B}(t)$.
- The bettor's estimated probabilities $p_i^{k,J}$ of the outcomes.

2.5.4 Variance of the Bankroll Factor

The variance of the bankroll factor provides insight into the risk or uncertainty associated with the bettor's strategy. A higher variance indicates greater risk, which may or may not be acceptable depending on the bettor's utility function.

Calculating Variance for a Single Match

For a single match m^k, the variance of the bankroll factor component due to that match is:

$$\mathrm{Var}_{p^J}\left[BF^{J,k}_{\text{bettor}}(t+1)\right] = \left(BF^{J}_{\text{bettor}}(t)\right)^2 \mathrm{Var}\left[\sum_{i=1}^{N^k} f_i^{k,J}(t)\left(o_i^{k,B}(t)X_i^k - 1\right)\right]$$

Within match m^k, the outcomes are mutually exclusive and collectively exhaustive, so we account for the covariance between different outcomes.

The variance expands to:

$$\mathrm{Var}_{p^J}\left[BF^{J,k}_{\text{bettor}}(t+1)\right] = \left(BF^{J}_{\text{bettor}}(t)\right)^2 \left[\sum_{i=1}^{N^k}\left(f_i^{k,J}(t)o_i^{k,B}(t)\right)^2 \mathrm{Var}[X_i^k] \quad -2\sum_{i<j} f_i^{k,J}(t)o_i^{k,B}(t)f_j^{k,J}(t)o_j^{k,B}(t)\mathrm{Cov}[X_i^k, X_j^k]\right]$$

Given that X_i^k is a Bernoulli random variable with success probability r_i^k, the true probability of outcome ω_i^k:

$$\mathrm{Var}[X_i^k] = r_i^k(1 - r_i^k)$$

However, the bettor does not know r_i^k and may use their estimated probability $p_i^{k,J}$ in their calculations. Despite this, the true variance depends on r_i^k, reflecting the inherent risk in the actual outcomes.

For the covariance between different outcomes:

$$\mathrm{Cov}[X_i^k, X_j^k] = -r_i^k r_j^k$$

This negative covariance arises because only one outcome can occur in a match.

Total Variance Across All Matches

Assuming independence between different matches, the total variance of the bankroll factor is the sum over all matches:

$$\mathrm{Var}_{p^J}\left[BF^{J}_{\text{bettor}}(t+1)\right] = \left(BF^{J}_{\text{bettor}}(t)\right)^2 \sum_{k=1}^{M}\left[\sum_{i=1}^{N^k}\left(f_i^{k,J}(t)o_i^{k,B}(t)\right)^2 r_i^k(1-r_i^k) - 2\sum_{i<j} f_i^{k,J}(t)o_i^{k,B}(t)f_j^{k,J}(t)o_j^{k,B}(t)r_i^k r_j^k\right]$$

Implications for Risk Management Understanding the variance of the bankroll factor helps the bettor manage risk. A higher variance indicates that the bankroll factor is more sensitive to the outcomes of the bets, which could lead to larger fluctuations in wealth.

2.5.5 Comparison of Objectives: Bettor vs. Bookmaker

Bettor's Perspective

The bettor observes the odds $o_i^{k,B}(t)$ offered by the bookmaker and decides on the fractions $f_i^{k,J}(t)$ of their bankroll to wager on each outcome ω_i^k. The bettor's optimization problem is to choose $f_i^{k,J}(t)$ to maximize their expected utility, given their estimated probabilities $p_i^{k,J}$.

Bookmaker's Perspective

The bookmaker sets the odds $o_i^{k,B}(t)$ before knowing the exact fractions $f_i^{k,J}(t)$ that bettors will wager. The bookmaker faces uncertainty regarding the bettors' actions and must estimate the aggregate fractions:

$$F_i^k(t) = \sum_J f_i^{k,J}(t)$$

across all bettors.

The bookmaker's optimization problem involves setting the odds $o_i^{k,B}(t)$ to maximize their expected utility, considering their own estimated probabilities $p_i^{k,B}$ and their expectations about bettors' wagering behavior.

Asymmetry and Strategic Interaction

This asymmetry creates a strategic interaction:

- **Bettor's Advantage:** The bettor acts after observing the odds, optimizing their bets based on their own estimated probabilities and the offered odds.
- **Bookmaker's Challenge:** The bookmaker sets the odds without knowing the exact betting fractions but must anticipate bettors' reactions. They need to estimate $F_i^k(t)$ to manage risk and ensure profitability.

If the aggregate fractions wagered by bettors are biased relative to the true probabilities, the bookmaker's optimization may lead to odds that create opportunities for bettors. This can happen if bettors do not optimize their bets uniformly or have varying probability estimates, giving an advantage to informed bettors even when their estimated probabilities are closer to the bookmaker's than to the true probabilities.

2.6 Conclusion

While this framework provides a solid structure for understanding the dynamics of the betting market, it comes with several limitations. First, the assumptions of independent match outcomes and a static information environment simplify the complexity of real-world dynamics, where outcomes may be correlated, and new information arrives continuously. Additionally, we do not optimize the timing of bets, which is a critical factor in real betting markets where odds fluctuate over time.

Moreover, the non-additivity of certain utility functions, such as logarithmic and exponential utilities, limits the general interpretation of long-term gain and risk. While maximizing utility in each time period offers insights into short-term decision-making, it does not fully capture the long-term wealth dynamics, especially under more realistic non-additive frameworks. This can affect the risk management strategies of both bettors and bookmakers, particularly when considering future opportunities and evolving market conditions.

In the next section, we will focus on the implementation of a system to apply this framework and evaluate it in practice, through both simulation and the integration of real-world data. This will allow us to test the framework's assumptions and explore the effects of relaxing some of these limitations.

Chapter 3

Design and implementation of the solution

3.1 Introduction

In the realm of sports betting, football stands out as an ideal focus for developing predictive and optimization models due to its global popularity and the abundance of available data. The rich historical datasets, comprehensive statistics, and extensive coverage make football a fertile ground for data-driven analysis. By concentrating on football, we can leverage vast amounts of information to build robust models that capture the nuances of the game, ultimately enhancing the accuracy of predictions and the effectiveness of betting strategies.

This chapter provides a comprehensive overview of the system architecture designed to implement the theoretical framework outlined earlier. We present the various components of the system, describe how they interact, and explain the workflows involved in data collection, storage, processing, and presentation. The goal is to give the reader a clear understanding of how the theoretical concepts are translated into a practical, working solution before delving into the specifics of the inference and optimization modules in subsequent chapters.

3.2 General System Architecture

The system is designed with modularity and scalability in mind, adhering to a microservices architecture [22] that allows individual components to operate independently and communicate through well-defined interfaces. This approach facilitates maintenance, testing, and future enhancements.

3.2.1 Components Overview

The system comprises the following primary components:

- **Data Collection Module**: Responsible for gathering historical and real-time data on football matches and betting odds from various sources.
- **Database**: Centralized storage for all collected data, predictions, and optimization results.
- **Prediction Module**: Utilizes machine learning models to estimate the probabilities of different match outcomes.
- **Optimization Module**: Computes optimal betting strategies based on the selected utility function and model predictions.
- **Model Monitoring Module**: Monitors training of inference models.
- **User Interface (UI) and Backend**: Provides users with access to data, predictions, and betting recommendations through a web-based platform.

- **Scheduler**: Automates the execution of tasks such as data collection, model retraining, and optimization at predefined intervals.
- **APIs**: Facilitate communication between components, ensuring seamless data flow and integration.

3.2.2 Interactions Between Components

The interactions between the components are orchestrated to ensure efficient data processing and timely updates:

1. The **Data Collection Module** retrieves data from external sources and stores it in the **Database**.
2. The **Prediction Module** trains models and infers probabilities on asked outcomes using data from the **Database** and storing the results in the **Database**. The training of the models is monitored using the **Model Monitoring Module** which stores the models metrics into the **Database**.
3. The **Optimization Module** calculates optimal betting fractions based on the predictions and current odds stored in the **Database** using a given strategy.
4. The **User Interface** fetches data from the **Database** via the **Backend** and presents it to the user.
5. The **Scheduler** triggers data collections, training, inference and optimisation using the APIs from **Data Collection Module**, **Prediction** and **Optimization Module** at specified times scheduled.

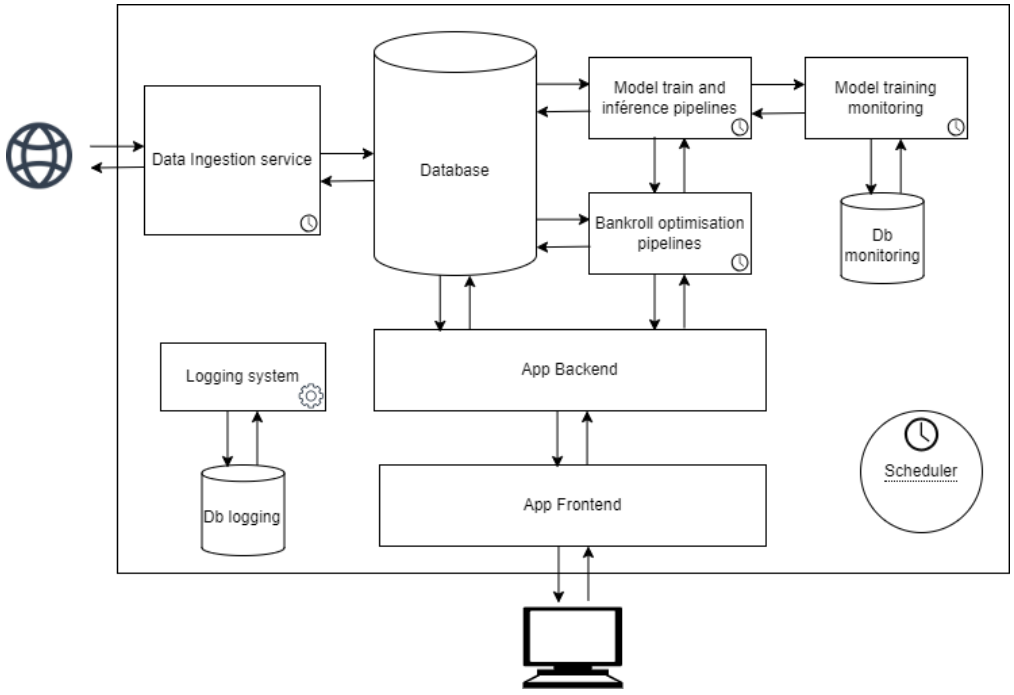

Figure 3.1: Architecture of the system

3.3 Data Collection

Accurate and comprehensive data collection is vital for building reliable predictive models and effective betting strategies. The goal is to build an historical database which continues to build with real time relevant data.

3.3.1 Data Sources Used

We utilize a variety of reputable sources to gather data:

- **Football Match Data**: Historical match results, match schedule, team statistics, player performance metrics, and other relevant information are sourced using scrapping on two websites:
  - FBref: For historical match results and coming match schedule.
  - SoFifa: For teams and players past and current ratings and statistics.
- **Odds Data**: Betting odds are collected from multiple bookmakers through one API.
  - The Odds API: The free tier credits allows to perform 500 requests per month on various sports, bookmakers and leagues to retrieve the current odds. Historical odds data are not included.

3.3.2 Collection Methods

Data is collected using a combination of methods:

Web Scraping A fork of the Soccerdata python library, has been adapted to scrape data from websites that do not provide APIs (FBref, SoFifa).

APIs For sources that offer APIs (The Odds API), we integrate with them using HTTP requests to fetch structured data efficiently.

Data Pre-processing Collected data undergoes a very simple pre-processing to ensure consistency and usability:

- **Data type conversion**: Adapting the type of the data to the most adapted type.
- **Unity**: Only inserting new data in the database, or filling None values of existing data (for instance, the score of a match is only available after the match is played
- **Integration**: Aligning data from different sources for seamless storage and analysis.

3.4 Data Storage

A robust data storage solution is essential for managing the diverse datasets involved.

3.4.1 Database Choice

We opted for a relational database management system (RDBMS), specifically *PostgreSQL*, due to its reliability, scalability, and support for complex queries.

3.4.2 Data Model

The database schema is designed to reflect the relationships between different types of data:

Tables

- '**fbref_results**': Each row corresponds to a match (historic and coming), with league, date and time of the match, both team and score if match is finished and the result is available and fetched from FBref website.
- '**sofifa_teams_stats**': Each row corresponds to a a team and a date of update with metrics and categorical values that represent at best the team at the moment of the update (overall score, attack, build_up_speed ...).

- '**soccer_odds**': Each row corresponds to a match, a bookmaker, an outcome with its odd at a given update time. There is also information about the commence time of the match, the league, the home and away team names, the type of odd...
- '**models_results**': Each row corresponds to a match, the inference results of the model, the date-time of inference and the model used, with additional information such as the date and tile of the match and home and away team.
- '**optim_results**': Each row corresponds to a game, a date time of optimisation, the model used for inference, the best odds for each outcome found across a pool of bookmaker as well as the bookmakers names of each odds chose and the fraction of the bankroll to invest given utility function. There is additional information such as the probability inferred and used by the optimiser, the date-time of inference of these probabilities, the date and time of the match...

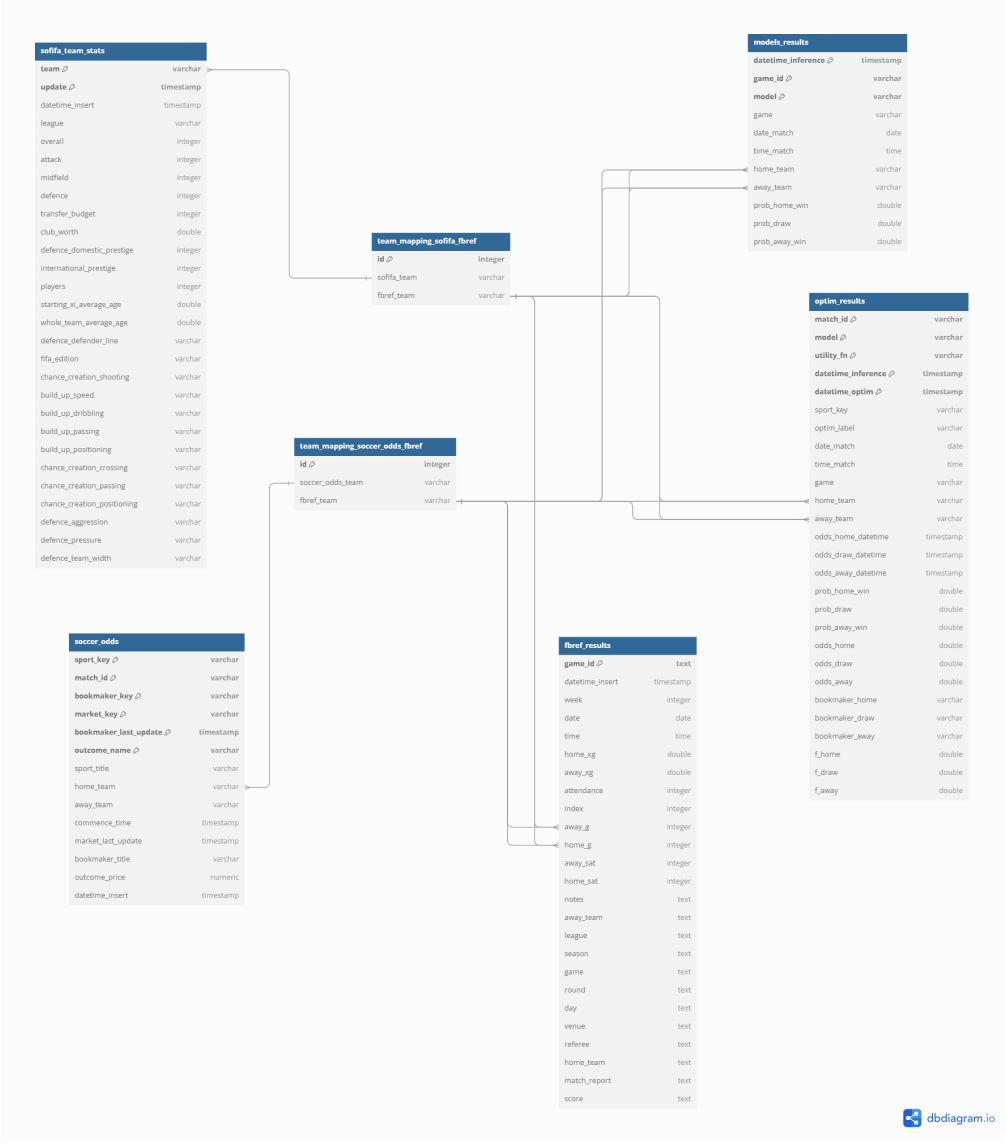

Figure 3.2: UML diagram of the database

3.5 Module Overview

The system incorporates several modules, each performing specific functions within the overall architecture.

3.5.1 Data Collection Module

As described earlier, this module is responsible for fetching and pre-processing data from various sources.

3.5.2 Prediction Module

Although detailed discussion is deferred to a later chapter, this module uses machine learning algorithms to predict the probabilities of different match outcomes based on historical data.

3.5.3 Optimization Module

This module calculates optimal betting strategies by applying mathematical optimization techniques to the predictions and odds data. The specifics of the optimization algorithms and utility functions will be explored in a subsequent chapter.

3.5.4 Scheduler

The scheduler automates the execution of tasks such as data collection, model retraining, inference, and optimization. It ensures that the system remains up-to-date with the latest data and predictions.

3.5.5 User Interface and Backend

The user interface provides a platform for users to access data, view predictions, and interact with the system. The backend handles user requests, processes data, and communicates with other modules via APIs.

3.6 User Interface and Monitoring

3.6.1 User Interface Design

The UI is designed to be intuitive and user-friendly, providing clear visualizations and easy navigation.

Features

- **Dashboard**: Displays key metrics, including upcoming matches, predicted probabilities, and recommended betting strategies.
- **Historical Data Access**: Allows users to explore past matches, predictions, and outcomes.
- **Customization**: Users can select preferred bookmakers according to their interests.

3.6.2 Monitoring

System health and performance are monitored continuously:

- **Logging**: Activity logs are maintained for debugging and audit purposes.
- **Alerts**: Notifications are sent in case of errors or significant events.

3.7 Conclusion

This chapter provided an overview of the system architecture implemented to realize the theoretical framework developed earlier. By focusing on football, we leverage abundant data to build predictive models and optimize betting strategies. The modular design, utilizing microservices and APIs,ensures scalability and flexibility. The database serves as the central repository, integrating data from various sources and supporting the different modules. The user interface offers a gateway for users to access the system's functionalities. In the subsequent chapters, we will delve into the specifics of the prediction and optimization modules, as well as the deployment strategy using Kubernetes and containerization technologies.

Chapter 4

Predictive Modeling of Match Outcomes

4.1 Introduction

This chapter presents the development, training, and evaluation of a predictive model aimed at forecasting the outcomes of football matches. The primary objective is to construct a robust model that can accurately predict match results, thereby optimizing gains in sports betting [3] [9]. The significance of predictive modeling in the context of sports betting lies in its potential to provide bettors with a strategic advantage by identifying value bets and minimizing risks.

4.2 Performance Metrics and Selection Criteria

Evaluating the performance of a predictive model in a multi-class classification setting, especially with imbalanced classes, requires a comprehensive set of metrics. This section delineates both classic and advanced metrics employed in this study, incorporating mathematical formulations and addressing class imbalance. Given the three-class problem—home win, draw, and away win—with home wins constituting 47% of the data, it is crucial to select metrics that provide a nuanced understanding of model performance across all classes.

4.2.1 Metrics

A list of all the metrics considered with their used definition can be found in Appendix F.

4.2.2 Selection Criteria

Accurate evaluation of the predictive model requires appropriate performance metrics, particularly in a multi-class classification context with class imbalance. The primary goal of this study is to ensure that the predicted probabilities of football match outcomes (home win, draw, away win) closely align with the true probabilities, emphasizing well-calibrated probability estimates.

Given the class distribution—47% home wins, 26% draws, and 25% away wins—we have selected the **Mean Squared Error (MSE)** as the primary metric for assessing calibration. MSE directly measures the average squared difference between predicted probabilities and actual outcomes, making it suitable for evaluating how well the model's probabilities reflect the true frequencies.

In addition to MSE, we will consider the following metrics to provide a comprehensive evaluation:

- **Log Loss**: To assess the quality of the predicted probability distributions by penalizing incorrect and overconfident predictions, thus encouraging well-calibrated estimates.
- **Classwise Expected Calibration Error (ECE)**: To evaluate the calibration of predicted probabilities for each class individually, offering insights into how closely these probabilities match the observed outcomes across different categories.

- **Accuracy for Home Win, Draw, and Away Win**: To examine the model's performance on each outcome separately, taking into account the class imbalance.

By focusing on MSE for calibration and incorporating Log Loss, Classwise ECE, and class-specific accuracy, we aim to ensure that the model not only produces accurate probability estimates but also maintains reliability across all outcome categories. This concise set of metrics aligns with our objective of accurately predicting football match outcomes while ensuring the predicted probabilities are well-calibrated and trustworthy.

4.3 Exploration and Choice of Features

Selecting appropriate features is pivotal for building an effective predictive model. This section delineates the various types of features utilized in this study, the methodology employed for feature selection, the engineering of new features to enhance predictive power, and the handling of categorical variables to ensure they are appropriately represented in the model.

4.3.1 Types of Features Utilized

The feature set comprises a combination of ranking-based, simple statistical, and domain-specific features. Each feature is defined mathematically where applicable and accompanied by an explanation of its relevance and computation.

Ranking Features

Ranking features provide a quantitative measure of team strength based on historical performance. These metrics are crucial as they encapsulate the overall ability and consistency of teams over time. All ranking features detailed formula are described in G.

- **Elo Score**

  The **Elo score** [10] [16] is a rating system originally developed for chess but widely adapted to various sports to rate players or teams. It reflects the relative skill levels of the teams based on game outcomes.

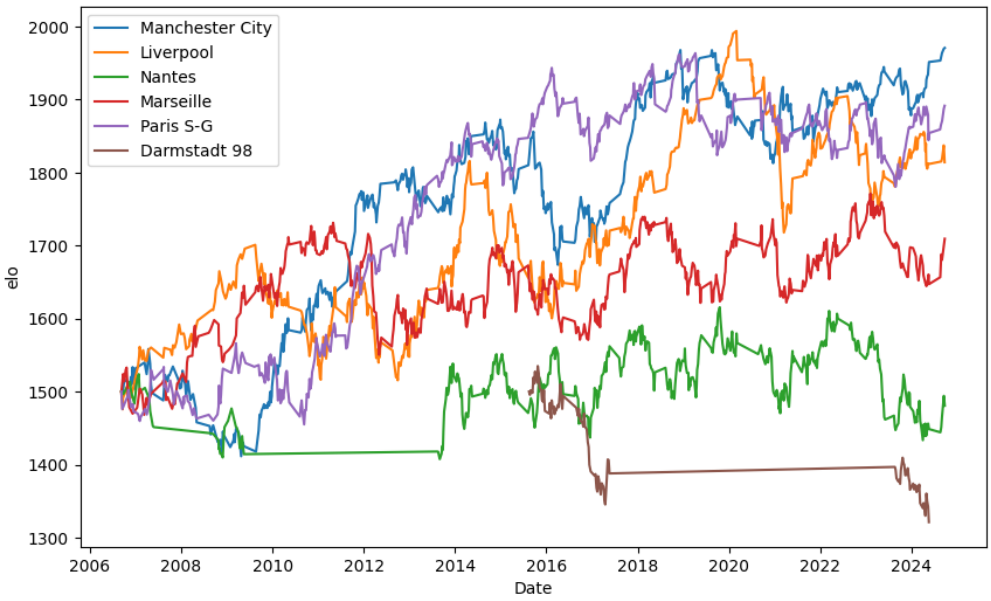

Figure 4.1: Elo score of 5 football teams evolving during time

- **Glicko-2 Score**

The **Glicko-2 score** [12] is an advanced rating system developed by Mark Glickman, which enhances the Elo rating system by incorporating not only the skill levels of teams or players (R) but also the reliability of these ratings through Rating Deviation (RD) and volatility. This system provides a more dynamic and accurate reflection of performance by accounting for the uncertainty and variability in teams' ratings.

- **TrueSkill**

  The **TrueSkill** [14] is a Bayesian ranking system developed by Microsoft, primarily used in gaming but adaptable to sports analytics. It models each team's skill as a Gaussian distribution, updating beliefs about team strengths based on match outcomes.

Simple Statistical Features

Simple statistical features offer basic quantitative measures of team performance, providing foundational data for the predictive model.

- **Average Goals Scored per Season**: Total goals scored by a team divided by the number of matches played so far.
- **Average Goals Conceded per Season**: Total goals conceded by a team divided by the number of matches played so far.

Sofifa Performance Metrics

SoFIFA provides detailed metrics for both individual players and teams, based on data from the FIFA video game by EA Sports. This document outlines the primary metrics and explains how team ratings are calculated using individual player attributes.

- **Player Metrics** The primary player metrics on SoFIFA are based on individual attributes that are weighted differently depending on the player's position. Below are the key metrics:
  - **Overall Rating (OVR)**: This is the weighted average of various player attributes, with different weights depending on the position. For example, an attacker (*Forward*) will have more emphasis on *Shooting* and *Pace*, while a defender (*Centre Back*) will weigh attributes like *Defending* and *Physicality* more heavily.
  - **Pace (PAC)**: Calculated as a combination of the *Acceleration* and *Sprint Speed* attributes.
  - **Shooting (SHO)**: Includes *Finishing*, *Shot Power*, *Long Shots*, and *Positioning*.
  - **Passing (PAS)**: Comprised of *Vision*, *Short Passing*, and *Long Passing*.
  - **Dribbling (DRI)**: Includes *Ball Control*, *Dribbling*, *Agility*, and *Balance*.
  - **Defending (DEF)**: Based on *Tackling*, *Marking*, *Interceptions*, and *Defensive Awareness*.
  - **Physicality (PHY)**: Includes *Strength*, *Stamina*, and *Aggression*.
  - **Potential**: Indicates the maximum possible rating the player can achieve over time.

  The formula for the Overall Rating (OVR) is generally unknown, but it can be expressed as a weighted sum of key attributes, depending on the player's position. A simplified formula for a forward might look like:

$$\text{OVR}_{\text{Forward}} = w_1 \cdot \text{PAC} + w_2 \cdot \text{SHO} + w_3 \cdot \text{DRI} + w_4 \cdot \text{PAS}$$

  where w_1, w_2, w_3, w_4 are position-specific weights.

- **Team Metrics** Team metrics on SoFIFA are calculated by aggregating individual player ratings, focusing on key areas like attack, midfield, and defense. The following are the primary team metrics:
  - **Overall Team Rating**: A weighted average of the starting 11 players' Overall Ratings, considering the importance of each player's position.

– **Attack Rating**: The average Overall Rating of forwards and attacking midfielders, weighted based on the formation.
– **Midfield Rating**: The average Overall Rating of central and wide midfielders, weighted based on their roles in the formation.
– **Defense Rating**: The average Overall Rating of defenders and goalkeepers.

A simplified version of the team rating could be expressed as:

$$\text{Team OVR} = \frac{1}{11} \sum_{i=1}^{11} \text{OVR}_i$$

where OVR_i represents the Overall Rating of the i-th player in the starting lineup.

Sofifa metrics are comprehensive team-specific performance indicators sourced from the Sofifa database, widely used in sports analytics and fantasy football contexts.

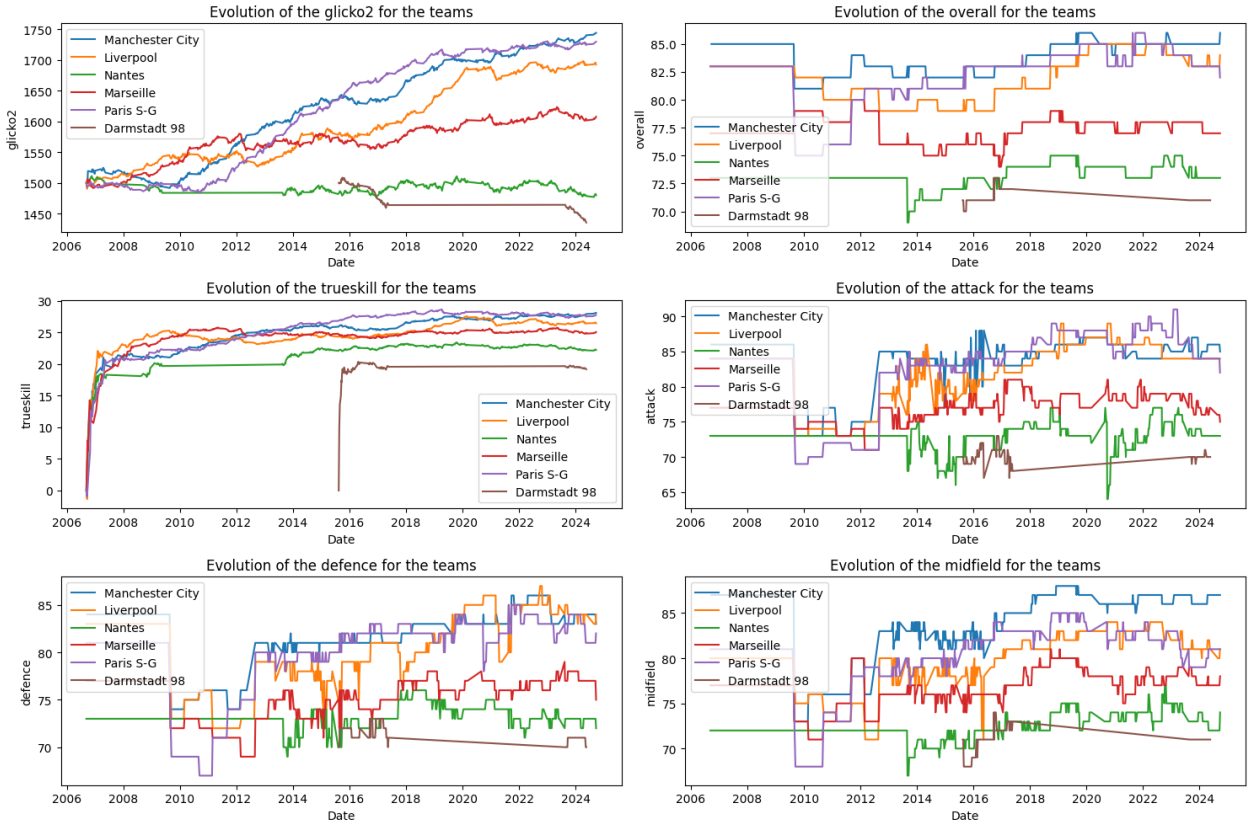

Figure 4.2: Different scores of 5 football teams evolving during time

A detailed description of each of the 90 features used can be found her H.

4.3.2 Feature Selection Methodology

Feature selection was performed using a forward selection approach applied to a logistic regression model. This method iteratively adds the most significant features, enhancing predictive performance while maintaining model simplicity.

Forward Selection with Logistic Regression

Procedure: Starting with no features, at each iteration, the feature that most improves the model's fit is added. The selection criterion is based on the mse (mean squared error).

Explanation: By incorporating features that significantly contribute to the model, forward selection optimizes predictive accuracy and ensures interpretability by excluding irrelevant variables.

4.4 Data Preparation

We trained our model on matches from 2006 to the present, focusing on games from the top 5 European leagues, European championships, and World Cups during this period. The limiting factor in our data came from SoFIFA, which does not provide data prior to 2006, while FBref offers data extending far into the past. We merged the two datasets based on team names and computed the ranking and statistical features described earlier, initializing the metrics at the first entry of a team in a tournament. For categorical features, we applied one-hot encoding. We removed matches with any missing values in the columns, then applied a standard scaler. This left us with 28,850 completed matches and a 90-feature vector for each match to train our model.

Metric	Value
Total matches	28,850
Matches in Top 5 Leagues	28,481
Matches in European Championships	185
Matches in World Cup	184
Home win ratio	45.0 %
Draw ratio	25.4 %
Away win ratio	29.5 %
Average home team goals	1.54
Average away team goals	1.19
Average Elo rating	1558
Number of unique teams	242
Number of features per match	90
First match date	2006-09-09
Last match date	2024-09-24

Table 4.1: Summary Metrics for the Dataset

4.5 Cross-Validation on Temporal Data

In predictive modeling with football match data, which is inherently temporal, it's essential to use cross-validation techniques that respect the chronological order of observations. Standard cross-validation methods, such as random shuffling or traditional k-fold cross-validation, are unsuitable because they can lead to data leakage by training models on future data to predict past events.

Traditional cross-validation assumes that data samples are independent and identically distributed (i.i.d.) and that the order of observations does not matter. In temporal data, however, observations are time-dependent, and future events should not influence model training aimed at predicting past events. Using standard methods can result in:

- **Data Leakage**: Incorporating future information into the training set leads to overly optimistic performance estimates.
- **Violation of Temporal Order**: Disrupting the sequence of events undermines the model's ability to generalize to real-world forecasting scenarios.

To address these issues, we employ cross-validation methods designed for temporal data [4].

4.5.1 Sliding Window Cross-Validation

This technique involves moving a fixed-size window across the data timeline. In each iteration, the model is trained on a training window and tested on the immediately following testing window.

- Choose a training window size W_{train} and a testing window size W_{test}.
- For each iteration:
  - Train the model on data from time t to $t + W_{\text{train}} - 1$.
  - Test the model on data from $t + W_{\text{train}}$ to $t + W_{\text{train}} + W_{\text{test}} - 1$.
  - Slide the window forward by W_{test} units.

4.5.2 Expanding Window Cross-Validation

Also known as growing window, this method expands the training set with each iteration by including more historical data.

- Start with an initial training window of size W_{initial}.
- For each iteration:
  - Train the model on data from time t to $t + W_{\text{train}} - 1$, where W_{train} increases with each iteration.
  - Test on the subsequent data from $t + W_{\text{train}}$ to $t + W_{\text{train}} + W_{\text{test}} - 1$.
  - Expand the training window to include the latest testing window.

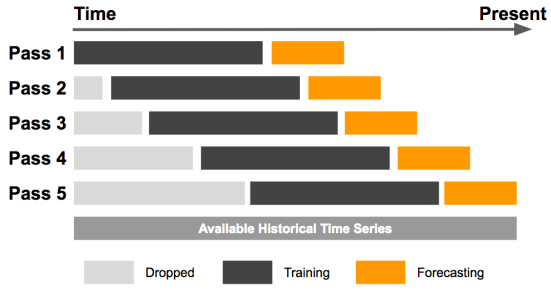

Figure 4.3: Sliding window graphic

Figure 4.4: Expanding window graphic

The results presented below were obtained using an expanding window technique with 5 folds and a test set ratio of 0.2. Notably, the results were very similar when applying the sliding window method.

4.6 Choice and Justification of the Prediction Model

In this section, we present the results of the feature selection and model selection processes [13] [8], followed by interpretations of the selected model's performance. Feature selection was conducted using forward selection, and multiple classification models were evaluated to determine the most suitable model for predicting football match outcomes.

4.6.1 Feature Selection Using Forward Selection

Feature selection was performed using forward selection with logistic regression, iteratively adding the most significant feature at each step based on mean squared error (MSE) improvement, using an expanding window validation with 5 splits and a test size of 20% of the training data.

Table 4.2: Feature Selection Process Summary

Method	**Details**
Feature Selection	Forward Selection
Model Used	Logistic Regression
Validation Method	Expanding Window (5 splits)
Performance Metric	Mean Squared Error (MSE)
Test Size	20% of training data

We selected 35 features, which corresponding to the features resulting in the lowest MSE, using this feature selection strategy.

Table 4.3: Feature Selection with Corresponding MSE and their adding number

Order	**Feature Added**	**MSE**	**Order**	**Feature Added**	**MSE**
1	Elo Away	0.20613	19	Home Passing Risky	0.19438
2	Elo Home	0.19661	20	Away Positioning Org.	0.19436
3	Glicko Vol Away	0.19619	21	Away Defense Pressure Med	0.19435
4	Away Overall	0.19594	22	Away Domestic Prestige	0.19434
5	Home Overall	0.19540	23	Away Shooting Lots	0.19433
6	Away Build Speed Slow	0.19518	24	Home Defense Line Offside	0.19432
7	Away Avg Age	0.19501	25	Away Team Width	0.19431
8	Home League INT	0.19487	26	Home Defense Pressure Med	0.19431
9	Home Avg Goals	0.19476	27	Home Build Speed Slow	0.19430
10	Home Positioning Org.	0.19467	28	Away Defense Aggression	0.19430
11	Home Build Speed Fast	0.19461	29	TrueSkill Home	0.19430
12	Away Defense Pressure High	0.19457	30	Away Build Positioning Org.	0.19430
13	Away Defense Offside Trap	0.19453	31	Away Defense	0.19430
14	Home League ITA	0.19449	32	Home Attack	0.19427
15	Glicko RD Home	0.19447	33	Home Defense Prestige	0.19427
16	Home Shooting Normal	0.19444	34	Away Attack	0.19427
17	Away Passing Mixed	0.19442	35	Away League INT	**0.19427**
18	Away Avg Goals	0.19440			

The table above summarizes the features added during the selection process and their corresponding MSE values, highlighting the importance of each feature as it contributes to minimizing the error. As we can see, features such as Elo ratings and overall team metrics play a significant role [19]. Now, let's examine how the number of features impacts the performance metrics more broadly, as shown in the following feature selection graph.

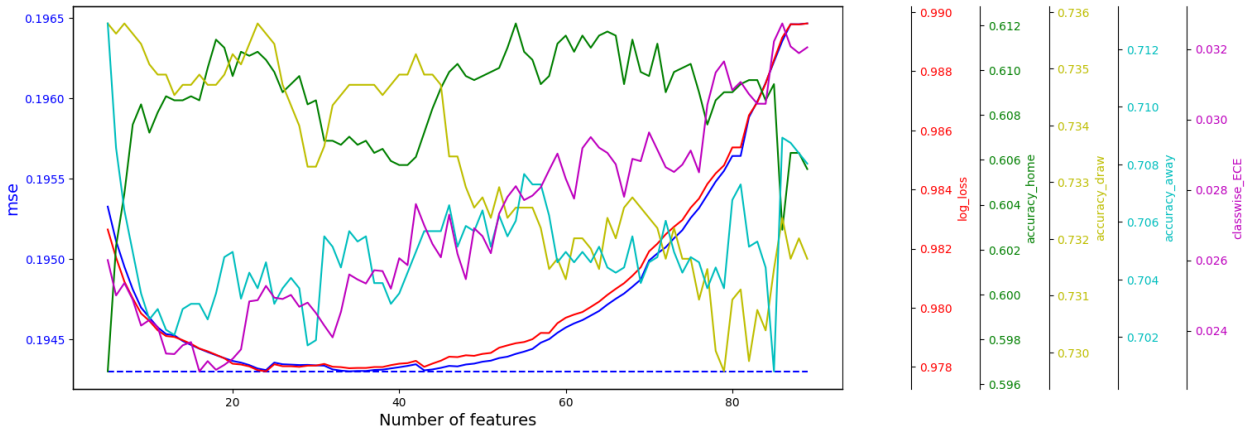

Figure 4.5: Metrics of interrest function of the number of features added

This graph shows the performance of various metrics (MSE, log loss, accuracy for home, draw, and away predictions, and classwise ECE) as a function of the number of selected features. The MSE (in blue) decreases as more features are added, stabilizing around the optimal point before increasing again, which suggests that selecting too many features can lead to overfitting. Similarly, log loss follows a similar trend (in red), indicating better model calibration with fewer features. The accuracy metrics (home, draw, away) fluctuate, but accuracy seems to peak at a certain range of features, with performance diminishing as more features are added. Classwise ECE (in pink) decreases and then increases, a little bit before MSE and log loss, indicating better calibration for class predictions with fewer features. Overall, the graph highlights the balance between feature selection and model performance, suggesting that an optimal subset of features yields the best generalization.

4.6.2 Model Selection

The following table summarizes the performance of various classification models [5], comparing metrics such as mean squared error (MSE), log loss, classwise ECE, and accuracy for home, draw, and away predictions to identify the best-performing model.

Table 4.4: Model Performance Comparison

Model	MSE	Log Loss	C. ECE	A. Home	A. Draw	A. Away
Logistic Regression	**0.195**	**0.983**	0.029	0.605	0.733	0.702
Logistic Regression CV	0.196	0.983	**0.028**	0.602	**0.735**	0.703
Gradient Boosting Classifier	0.199	1.002	0.037	0.604	0.709	**0.706**
Random Forest Classifier	0.202	1.022	0.038	0.595	0.705	0.693
Extra Trees Classifier	0.204	1.026	0.043	0.597	0.683	0.686
AdaBoost Classifier	0.221	1.092	0.069	0.599	0.721	0.695
Bagging Classifier	0.224	2.471	0.093	0.602	0.646	0.661
MLP Classifier	0.224	1.187	0.108	0.585	0.665	0.684
K Neighbors Classifier	0.238	5.404	0.096	0.599	0.643	0.631
Gaussian NB	0.332	7.570	0.302	**0.615**	0.584	0.625
Quadratic Discriminant Analysis	0.353	10.831	0.316	0.582	0.561	0.613
Decision Tree Classifier	0.390	20.219	0.195	0.578	0.614	0.638
Extra Tree Classifier	0.399	20.686	0.200	0.559	0.615	0.628

4.6.3 Interpretation of Results

The selection of the logistic regression model allows for straightforward interpretation of feature effects on the predicted probabilities of match outcomes.

Feature Importance

Feature importance was assessed based on the magnitude of the coefficients in the logistic regression model. Below sits the feature importance of the Home win class. Draw and Away win classes analysis can be found in I.

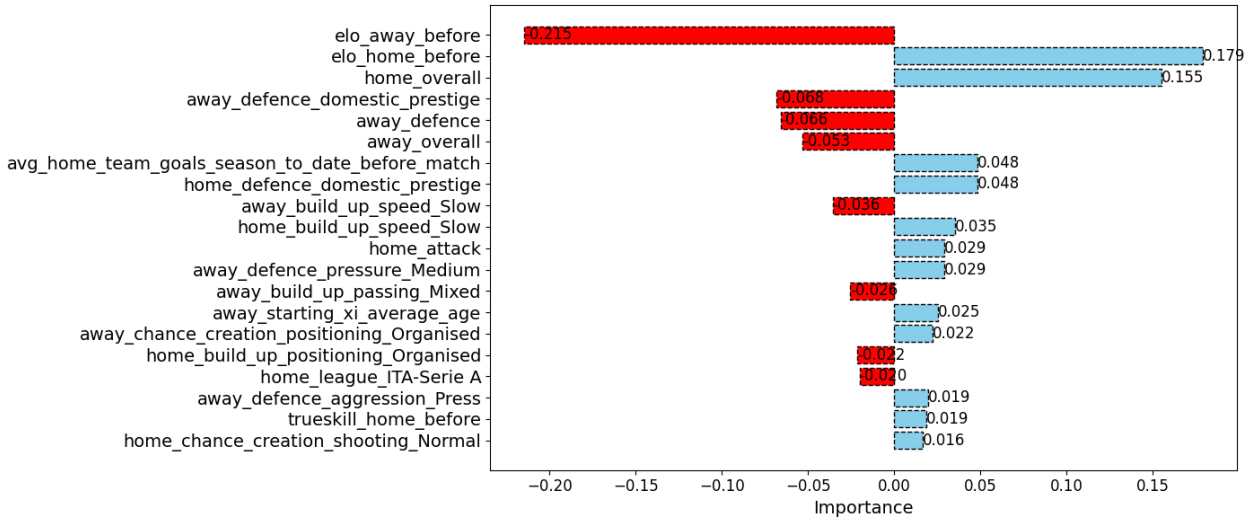

Figure 4.6: Coefficients of the Logistic Regression Model for Home win class

For the home class, the most important features, such as Elo ratings for the away and home teams, suggest that pre-match team strength is the most significant predictor of match outcomes. Both overall team quality and specific defensive attributes, like pressure and aggression, also play a key role. Features related to player average characteristics, such as average age and tactical elements like build-up speed, indicate that team composition and playstyle are also relevant, though their impact is less pronounced. Defensive strategies, particularly pressure and team width, add further predictive value, showing the importance of tactical decisions in determining match results. The feature importance analysis graphs for draw and away class can be found in the annex section.

Why Logistic Regression Outperforms Other Models

Logistic regression may outperform other models due to its simplicity and interpretability, especially when feature selection is based on it. By using logistic regression for feature selection, the model is specifically tuned to highlight the most important predictors of the outcome, leading to better generalization. Additionally, logistic regression handles multicollinearity well when regularization is applied, preventing overfitting. The linear relationship between the features and the log-odds of the outcomes makes it easier to capture important patterns in the data, particularly in problems like sports prediction where relationships between variables are often linear. Other models, such as random forests or gradient boosting, may add unnecessary complexity and are more prone to overfitting when features are already well-selected.

4.7 Training and Retraining of the Model

Figure 4.7 illustrates the Mean Squared Error (MSE) of two models over time, where the blue line represents Model A with no retraining, and the orange line represents Model B, which is retrained daily. Both models are initialy trained from 2006-01-01 up to 2010-01-01 data and are evaluated using a 120-day rolling average.

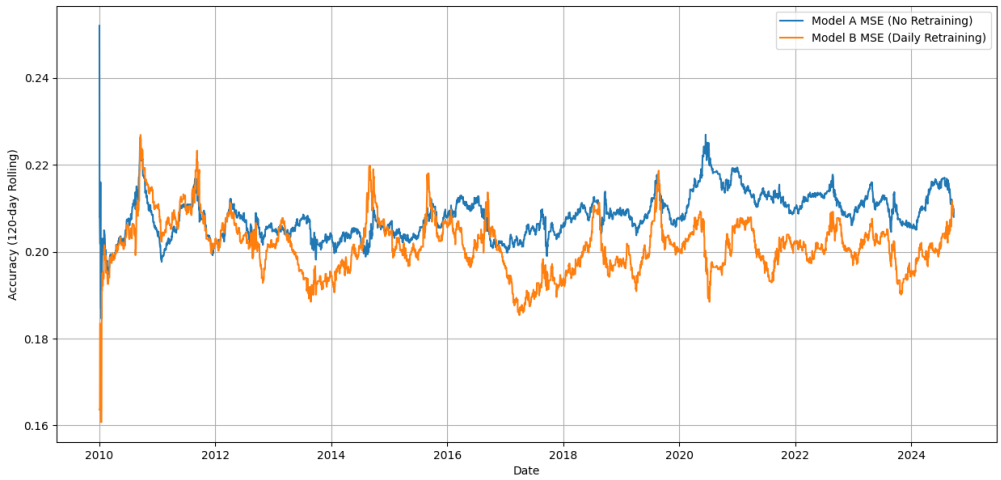

Figure 4.7: Model MSE rolling average over time

From the figure, we observe that Model B, which is frequently retrained, exhibits lower MSE compared to Model A throughout most of the time period. Retraining appears to allow Model B to adapt more effectively to evolving patterns, leading to consistently better performance in terms of accuracy. Moreover, as time goes by, we can observe a mse drift from the not retrained model as well as a slight improvement from the retrained model.

There are very few periods where Model A outperforms Model B. It appends especially during phases of sudden changes. Despite these fluctuations, retraining offers a more stable and improved long-term performance.

The results highlight the importance of regular retraining for maintaining model accuracy, particularly in dynamic environments where data patterns change over time.

4.8 Conclusion

This chapter presented the development, training, and evaluation of a predictive model for football match outcomes, with a focus on sports betting. Feature selection via forward selection with logistic regression helped identify key predictors, and regular retraining improved model performance over time.

However, several limitations remain:

- **Hyperparameters and Features**: Ranking feature hyperparameters were left at default, and additional domain-specific or external data sources could further improve predictions.
- **Feature Selection**: Feature selection was arbitrarily based on logistic regression, and no hyperparameter optimization was performed for any models.
- **Retraining**: The timing and method of retraining (e.g., sliding window) were not explored, potentially missing optimal strategies as well as posing computational challenges that could be optimized.
- **Model Complexity**: Incorporating deep learning models could enhance predictive performance, particularly for capturing complex patterns in the data.
- **Bookmaker Odds Decorrelation**: Adding a metric to assess the decorrelation between model predictions and bookmaker odds could help identify more value bets and further optimize betting strategies.

In the next chapter, we address the optimization problem of determining the bankroll share to invest on each outcome, building on the predictions of this model to develop actionable betting strategies.

Chapter 5

Optimization of bankroll allocation

5.1 Introduction

Effective bankroll management is a critical component of successful sports betting. It involves determining the optimal amount to wager on each bet to maximize growth while minimizing the risk of ruin. This section details the development and evaluation of an optimization module designed to calculate the optimal fraction of the bankroll to invest in each betting opportunity. The module leverages probabilistic forecasts from the predictive models discussed in the previous section and applies various investment strategies, including the Kelly criterion, expected value maximization, and naive betting approaches.

5.2 Methodology

5.2.1 Investment Strategies

This section provides an overview of the different bankroll allocation strategies implemented, ranging from naive methods to more advanced optimization techniques. The focus is on the principles guiding each strategy, with a detailed formula provided only for the naive strategy.

List of Strategies

- **Kelly Criterion Strategy**: [18] [24] This strategy maximizes the logarithmic utility of wealth, aiming for long-term bankroll growth while managing risk. The bankroll fractions are derived from the analytical solution using the approximations B, $\mathbb{E}(U(B)) = \mathbb{E}(B) - \frac{1}{2} \cdot \mathbb{V}\text{ar}(B)$ which comes down to a Linear utility strategy using $\lambda = \frac{1}{2}$.
- **Log Utility Strategy**: Similar to the Kelly criterion, this strategy focuses on maximizing the expected logarithmic utility $U(B) = \ln(B)$ but using no approximations C.
- **Exponential Utility Strategy**: This strategy uses an exponential utility function $U(B) = -e^{-\alpha B}$ to take into account the bettor's risk aversion, balancing between expected returns and risk tolerance D.
- **Linear Utility Strategy**: In this strategy, the objective is to maximize the trade-off between expected returns and risk, represented by the function $\mathbb{E}(U(B)) = \mathbb{E}(B) - \lambda \cdot \mathbb{V}\text{ar}(B)$. For the simulations, we set $\lambda = 10$, reflecting a high level of risk aversion. This approach seeks to maximize returns while penalizing high variance, aiming to balance growth and stability E.
- **Expected Value Maximization Strategy**: This strategy optimizes bankroll allocation based purely on maximizing expected value, $U(B) = B$, without considering risk or variance.
- **Naïve Strategy: Bet on the Most Likely Outcome**: In this straightforward approach, the bettor places the entire bet on the outcome with the highest implied probability, as per the bookmaker's odds.

The formula for this strategy is:

$$f_{k,i} = \begin{cases} \frac{1}{M}, & \text{if } i = \arg\max(o_k) \\ 0, & \text{otherwise} \end{cases}$$

where:

- $f_{k,i}$ is the fraction of the bankroll wagered on outcome i of match k,
- o_k are the odds for match k.
- M is the number of matches available.

This strategy is simple and allocates all the available funds to the outcome with the highest bookmaker odds.

These strategies were benchmarked against each other in the Monte Carlo simulations and then Online testing to assess their effectiveness in managing risk and maximizing bankroll growth.

For each strategy, a factor of $\gamma = \frac{1}{2}$ was applied to the bet fractions to ensure that not the entire bankroll was wagered at any given time, thereby providing a margin of safety, such as: $f_{strategy\_final} = \gamma \times f_{strategy}$.

5.2.2 Optimization Algorithms

Two optimization algorithms were employed to solve the bankroll allocation problem [6]:

- **Sequential Least Squares Programming (SLSQP):** An iterative method for constrained nonlinear optimization that is efficient for problems with a moderate number of variables.
- **Trust-Region Constrained Algorithm (trust-constr):** Suitable for large-scale optimization problems, it handles large numbers of variables and constraints effectively.

The choice between SLSQP and trust-constr depends on the number of betting opportunities (matches) considered at once. For a large number of matches, trust-constr is preferred due to its scalability.

5.3 Monte Carlo Simulations

To assess the performance of different investment strategies under simulated sports betting conditions, we conducted Monte Carlo simulations modeling the inherent uncertainties. The goal was to evaluate how various bankroll allocation strategies perform over numerous trials.

5.3.1 Simulation Setup

We simulated true match outcome probabilities r using a Dirichlet distribution appropriate for mutually exclusive and collectively exhaustive events:

$$r_i^k = \text{Dirichlet}(\alpha), \quad \alpha = (1, 1, 1)$$

To introduce discrepancies between true probabilities and those estimated by bookmakers (b) and players (t), we added biases and normally distributed noise:

$$b_i^k = \text{clip}(r_i^k + \text{bias}_{\text{bookmaker}} + \epsilon_{\text{bookmaker}},\ \text{min\_prob},\ \text{max\_prob})$$
$$t_i^k = \text{clip}(r_i^k + \text{bias}_{\text{player}} + \epsilon_{\text{player}},\ \text{min\_prob},\ \text{max\_prob})$$

where $\epsilon_{\text{bookmaker}}, \epsilon_{\text{player}} \sim \mathcal{N}(0, \sigma^2)$. Probabilities were normalized to sum to one, and bookmaker probabilities included a margin $\text{margin}_{\text{bookmaker}}$, clipping between min_prob and max_prob. Bookmaker odds were calculated as:

$$o_i^k = \frac{b_i^k}{(\sum_{i=0}^{N} b_i^k) - \text{margin}_{\text{bookmaker}}}$$

5.3.2 Simulation Procedure

The simulation followed a structured approach to evaluate the performance of different betting strategies, using predefined constants and a series of steps to simulate match outcomes and bankroll updates.

Table 5.1: Simulation constants

Constant	Value
H	30
T	50
N	3
min_prob	0.05
max_prob	0.95

Constant	Bettor	Bookmaker
bias	0	0
σ (for noise ϵ)	0.1	0.1
margin		0.1

1. Generated true probabilities r using bias $= 0$ and $\epsilon = 0.1$ for both bettor and bookmaker.
2. Computed bookmaker and player estimates b and t.
3. Calculated bookmaker odds o.
4. For each strategy:
   - Determined bet sizes using t and o by performing optimisation using `truct_constr` algorithm.
   - Simulated match outcomes based on r.
   - Updated bankrolls accordingly.

5.3.3 Evaluation Metrics

Strategies were evaluated using:

- **Final Bankroll Statistics**: Mean, standard deviation, median, minimum, and maximum.
- **Average Growth Rate**: Geometric mean per time step.

$$GGR = \left(\frac{B(n)}{B(0)}\right)^{\frac{1}{n}} - 1$$

- **Sharpe Ratio**: Risk-adjusted return.

$$\text{Sharpe Ratio} = \frac{\frac{1}{n}\sum_{t=1}^{n} R(t)}{\mathbb{V}ar(R(t))}) \text{ with } R(t) = \frac{B(t+1) - B(t)}{B(t)}$$

- **Probability of Ruin**: Frequency of bankroll falling below a threshold.

5.3.4 Results

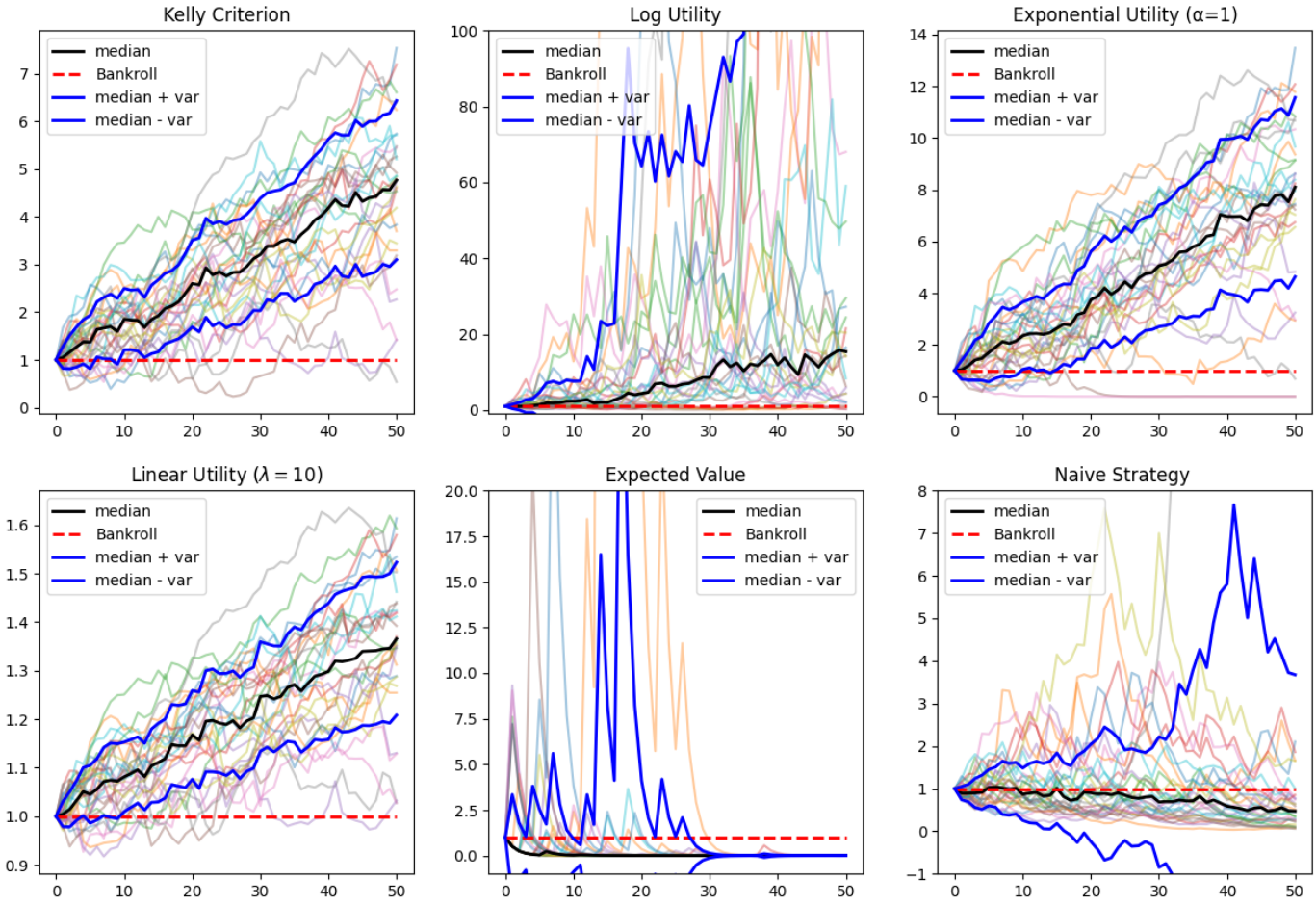

Figure 5.1: Monte carlo simulations for each strategy

Table 5.2: Final Bankroll Statistics

Strategy	**Mean**	**Std Dev**	**Median**	**Min**	**Max**
Kelly Criterion	4.48	1.67	4.77	0.54	7.54
Log Utility	270.85	983.14	15.39	0.19	5414.01
Exponential Utility	7.46	3.46	8.10	0.00	13.48
Linear Utility	1.35	0.16	1.37	1.03	1.61
Expected Value	0.00	0.00	0.00	0.00	0.01
Naïve Strategy	1.20	3.20	0.47	0.06	18.19

The **Log Utility** strategy achieved the highest mean final bankroll but with significant variability, indicating high risk. The **Kelly Criterion** and **Exponential Utility** strategies demonstrated moderate returns with lower variability, suggesting consistent performance.

Table 5.3: Average Growth Rate Per Step

Strategy	**Growth Rate**
Kelly Criterion	2.82%
Log Utility	5.54%
Exponential Utility	2.55%
Linear Utility	0.59%
Expected Value	−29.83%
Naïve Strategy	−1.55%

While the **Log Utility** strategy had the highest growth rate, it came with increased volatility. The **Kelly Criterion** and **Exponential Utility** strategies offered positive growth with better risk control.

Table 5.4: Sharpe Ratio

Strategy	**Sharpe Ratio**
Kelly Criterion	0.30
Log Utility	0.29
Exponential Utility	0.25
Linear Utility	0.30
Expected Value	0.14
Naïve Strategy	0.01

The highest Sharpe Ratios were achieved by the **Kelly Criterion** and **Linear Utility** strategies, indicating superior risk-adjusted returns.

Table 5.5: Probability of Ruin

Strategy	**Probability**
Kelly Criterion	0.00%
Log Utility	3.33%
Exponential Utility	6.67%
Linear Utility	0.00%
Expected Value	100.00%
Naïve Strategy	20.00%

Zero probability of ruin for the **Kelly Criterion** and **Linear Utility** strategies underscores their robustness.

An ANOVA test (performed to assess whether the differences in final bankrolls among the strategies), (F-statistic: 2.16, p-value: 0.0612) suggested that differences among strategies were not statistically significant at the 5% level. However, the p-value is close to the threshold, suggesting that with a larger sample size, the differences might become statistically significant.

5.3.5 Conclusion

The simulations indicate that strategies like the **Kelly Criterion** and **Exponential Utility**, which balance growth and risk through utility maximization, offer favorable outcomes. The **Log Utility** strategy provides high growth potential but with greater volatility. Ignoring risk, as in the **Expected Value** strategy, leads to poor performance.

Limitations include the limited number of simulations, simplified assumptions, and exclusion of real-world factors like transaction costs.

Recommendations for future work involve increasing simulation runs, incorporating realistic market conditions, and exploring additional strategies.

5.4 Online Testing

To assess the strategies in a real-world environment, an online testing phase was conducted over five weeks, from 2024 August 24th to 2024 September 30th, focusing on matches from the five major European football leagues. This real-world testing evaluated the practical applicability and performance of the strategies under actual market conditions. Odds were scraped each day at 12pm from the Odd Api website.

5.4.1 Static Problem Reduction and Parameter Settings

To simplify the dynamic nature of sports betting, we reduced the problem to a series of static optimizations at discrete time intervals. At each decision point t, bankroll allocation was optimized based on the current available information. This approach allowed us to manage the complexity of real-time betting while ensuring the practical applicability of the strategies.

Temporal Parameters Key temporal parameters were defined as follows:

- **Betting Interval (Δt)**: The interval between placing bets, set to 24 hours to capture daily betting opportunities.
- **Bet Placement Timing**: Bets were placed at a fixed time each day (12:00 PM) to ensure up-to-date information was used while accommodating market dynamics.

These settings ensured a balance between information accuracy and practical market conditions.

Match Selection The matches selected for each optimization were determined based on:

- **Number of Matches (M)**: Matches occurring within the next 24 hours were selected, balancing opportunity and reliability of information as well as having all results while perfomring next step.
- **Selection Criteria**: Focus was given to matches from top European leagues where the bettor had a higher perceived edge.

This careful match selection helped reduce computational complexity while enhancing potential returns.

Re-Betting Policy The re-betting policy was defined by the following criteria:

- **Not allowing Re-Bets**: Additional bets on previously considered matches were not allowed. As we only bet on matches on the same day and only once a day, this was an implication of the previous choices.

This policy helped manage risk and adapt to evolving market conditions.

5.4.2 Practical Implementation Settings

The practical implementation settings for the online testing phase are summarized in Table 5.6. The testing period ran from August 24, 2024, to September 30, 2024. The `trust-constr` algorithm was used for optimization, with a multiplier of $\gamma = \frac{1}{2}$ applied to the matrix f. The best odds from a pool of bookmakers (detailed in the appendix) were selected for each match.

Table 5.6: Practical Implementation Settings

Setting	Value
Betting Interval (Δt)	24 hours
Bet Placement Time	12:00 PM daily
Look-Ahead Horizon	Matches within the next 24 hours
Re-Betting Policy	Not allowed
Testing Period	August 24, 2024 – September 30, 2024
Optimization Algorithm	`trust-constr`
Strategy factor mult. γ	0.5
Odds Selection	Biggest odds from a pool of bookmakers
Markets	5 biggest European leagues (Big 5)

5.4.3 Results and Interpretation

Figure 5.2 illustrates the capital evolution for each strategy during the testing period. The Kelly and Exponential Utility strategies exhibited the strongest performance, both ending with approximately twice the initial capital. These results highlight their ability to optimally balance risk and reward, consistently outperforming the more conservative Log and Naive strategies. However, they also demonstrated higher volatility compared to the more stable Linear strategy.

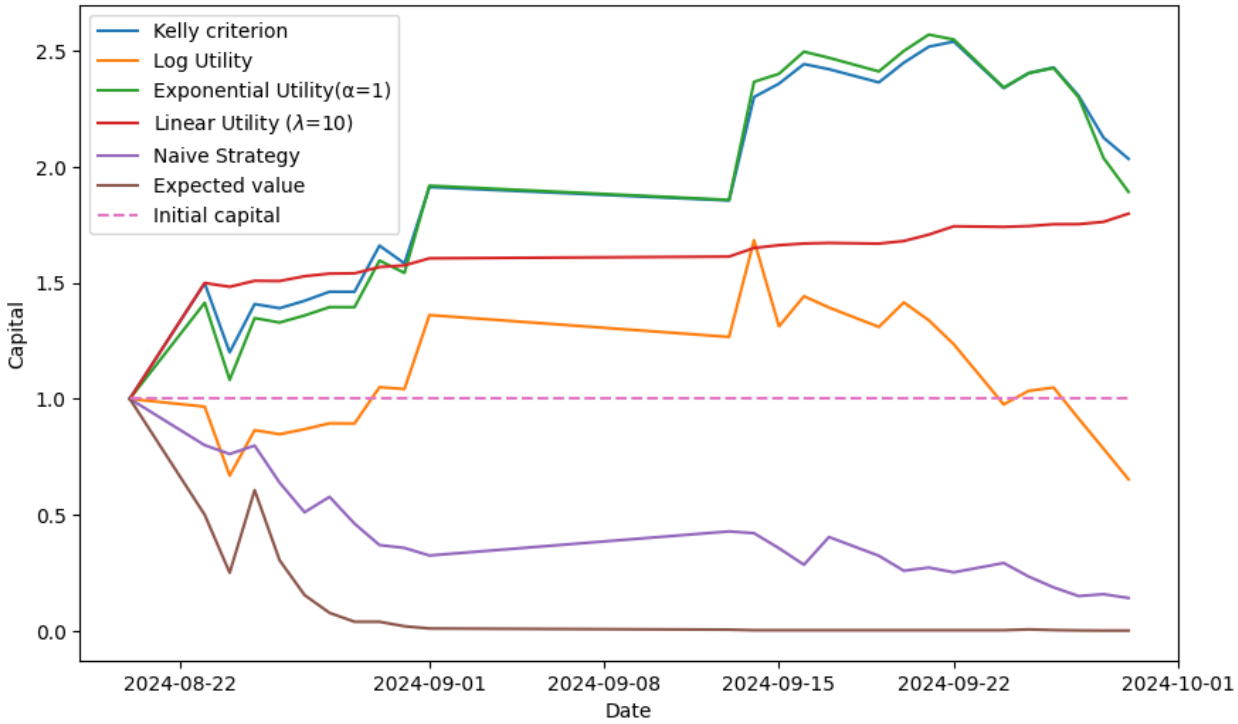

Figure 5.2: Capital Evolution for Each Strategy During the Online Testing Period

The Log Utility strategy underperformed, particularly at the start and after the midpoint of the test period, failing to capitalize on high-return opportunities. Its conservative nature, aimed at minimizing risk, resulted in modest growth but ultimately led to a negative outcome.

Both the Naive and Expected Value strategies experienced sharp declines in capital. The Naive strategy approached near-zero capital by the end of the test, while the Expected Value strategy exhibited extreme volatility, leading to rapid capital depletion. These simpler strategies, which lack sophisticated optimization, were highly vulnerable to adverse market conditions and failed to adapt effectively to fluctuations in odds or match outcomes.

In contrast, the Linear Utility strategy showed steady and consistent growth, with minimal volatility throughout the testing period, ultimately achieving a final growth of 1.8 times the initial capital. This highlights its effectiveness in maintaining a stable growth trajectory while avoiding the excessive risk associated with other strategies.

Overall, the results underscore the superiority of more advanced utility-based strategies such as Kelly and Linear. These approaches consistently outperformed simpler methods by balancing risk and reward more effectively under real-world betting conditions.

5.4.4 Performance Metrics

To further quantify the performance of each strategy, we computed key metrics, including final bankroll $B(T)$, mean growth per step, and standard deviation of growth per step, both in absolute terms and as a percentage of the final bankroll.

- The mean growth per step is defined as:

$$\mu = \frac{1}{T-1} \sum_{t=1}^{T-1} \Delta B_t$$

  where $\Delta B_t = B_{t+1} - B_t$,

- the standard deviation of growth per step is given by:

$$\sigma = \sqrt{\frac{1}{T-1} \sum_{t=1}^{T-1} (\Delta B_t - \mu)^2}$$

Table 5.7 summarizes the results for each strategy.

Table 5.7: Strategy Performance Metrics

Strategy	**Final Bankroll $B(T)$**	**Mean Growth (%)**	**Std Growth (%)**
Kelly Criterion	2.034	2.034	8.923
Log Utility	0.653	-2.129	26.516
Exponential Utility ($\alpha = 1$)	1.892	1.886	10.309
Linear Utility ($\lambda = 10$)	1.798	1.776	5.360
Naive Strategy	0.141	-24.299	52.419
Expected Value	0.001	-5032.448	18175.649

5.4.5 Interpretation of Metrics

The results demonstrate the effectiveness of the Kelly Criterion and Exponential Utility strategies, both of which ended with a final bankroll close to 2.0. These strategies also displayed reasonable volatility, with standard deviations of growth per step under 10%. The Linear Utility strategy performed consistently, achieving steady growth with the lowest volatility among all strategies.

On the other hand, the Log Utility strategy suffered from negative growth, highlighting its inability to capitalize on high-return opportunities. The Naive and Expected Value strategies performed poorly, with significant capital depletion and extreme volatility, particularly for the Expected Value approach, indicating their inadequacy in real-world betting scenarios.

5.5 Conclusion

The results from both the Monte Carlo simulations and the real-world online testing phase demonstrate the clear advantages of sophisticated bankroll management strategies such as the Kelly Criterion and Exponential Utility methods. These strategies consistently provided strong returns while managing risk effectively, leading

to a final bankroll close to twice the initial amount. In contrast, simpler strategies like the Naive and Expected Value approaches underperformed, suffering from capital depletion and high volatility, emphasizing the importance of balancing risk and return in real-world betting scenarios.

The Linear Utility strategy offered a steady, reliable growth trajectory with minimal volatility, making it an appealing option for risk-averse bettors. The Log Utility strategy, though conservative, failed to capture sufficient growth opportunities, resulting in a negative final outcome. Overall, the Kelly and Exponential Utility strategies are best suited for bettors seeking long-term growth with manageable risk.

5.5.1 Limitations and Future Improvements

Despite the promising results, several limitations were identified in the current approach:

- **Simulation Assumptions**: The Monte Carlo simulations relied on several simplifying assumptions that limit the realism of the results. Firstly, the simulation of probabilities was based on the true clipped probabilities plus a bias and Gaussian noise, which does not fully capture the actual flaws in the predictive models, and the constants used were chosen arbitrarily without strong justification. Secondly, the bookmaker margin was fixed, and the odds provided by the bookmaker did not account for the influence of large bets from the players, which in reality could cause deviations in the bookmaker's odds and probabilities. Lastly, the simulations used a fixed number of matches and time steps. Both the number of simulations and the range of strategies could be expanded to provide a more thorough and diverse analysis of performance over a wider variety of scenarios.
- **Limited Testing Period**: The online testing phase covered only a five-week period, which may not fully capture the long-term performance and robustness of each strategy. Extending the testing period or repeating it across multiple seasons would provide a more comprehensive assessment.
- **Risk Preferences**: While the utility-based strategies successfully managed risk, the models relied on fixed parameters for risk aversion (e.g., λ in Linear Utility). Introducing dynamic adjustments to these parameters based on market conditions or bettor preferences could further enhance performance.

5.5.2 Future Work and Deployment on Azure Kubernetes

The next stage of this project involves deploying the entire system on a cloud infrastructure using Kubernetes on Azure (AKS). This deployment will enable scalable, real-time processing of betting opportunities, continuous model updates, and the handling of multiple simultaneous users and markets. By leveraging Azure's powerful compute and orchestration capabilities, the system will be capable of efficiently managing the computational load and data flows needed for real-time sports betting optimization.

Chapter 6

Development of the Complete System and Production Deployment

6.1 Microservices Architecture and Frameworks

To build a scalable and maintainable system for sports betting optimization, we adopted a microservices architecture. This approach allows independent development, deployment, and scaling of individual components, facilitating modularity and flexibility.

The system comprises several microservices:

- **Data Ingestion Service**: Collects real-time match data and odds from external APIs and web scraping. We use the Python library *soccerdata* to conveniently scrape historical and real-time data from various websites. *SQLAlchemy* is employed to communicate with the PostgreSQL database, allowing interaction using a mix of Python syntax and SQL queries. An API is provided for other services to trigger this service's logic, created using the *FastAPI* framework. The Python library `logging` is used for proper logging of all operations, and `pandas` is utilized for data manipulation and transformation.
- **Prediction and Optimization Service**: Processes data to train models and generate probability estimates for the prediction component. Calculates optimal bet allocations based on the probabilities and selected strategies for the optimization component. *Scikit-learn* is used for model training and inference, while *SciPy.optimize* handles the optimization processes. Similar to the Data Ingestion Service, an API is deployed using *FastAPI*, with communication to the database via *SQLAlchemy*, and `logging` and `pandas` for logging and data handling.
- **User Interface Service**: Provides a web-based dashboard for monitoring and interaction, developed using the Python web framework *Streamlit*.
- **Backend Service**: Manages communication and logic between the frontend User Interface and the database, as well as other services, using *FastAPI*, `pandas`, and `logging`.
- **Database Service**: Stores historical data, odds, inferred probabilities, optimization results, and transaction logs. We chose PostgreSQL as the database due to its robustness, scalability, and compatibility with *SQLAlchemy*. PostgreSQL's advanced features support complex queries and transactions essential for our application's needs.
- **MLflow Service**: Monitors the training metrics of the models. MLflow provides a convenient way to track experiments, record model parameters, metrics, and artifacts, facilitating reproducibility and model versioning.
- **Airflow Service**: Acts as a scheduler, providing a convenient way to orchestrate and monitor complex workflows using Directed Acyclic Graphs (DAGs). Apache Airflow allows us to define data pipelines, schedule tasks, and manage dependencies between them, ensuring timely execution of data ingestion, model training, and optimization processes.

Services communicate over HTTP/HTTPS protocols, with well-defined API endpoints ensuring loose coupling and ease of integration.

6.2 Docker and Kubernetes

To ensure consistency across development, testing, and production environments, all microservices are containerized using Docker [21]. Docker allows us to package each service with all its dependencies into isolated containers, ensuring consistent behavior across different environments.

6.2.1 Dockerization

Each microservice is encapsulated in a Docker container, defined by its own `Dockerfile`, which specifies the base image, dependencies, and entry points. In local development, we used containers for services such as MLflow, PostgreSQL, and Airflow, facilitating a consistent and reproducible environment.

6.2.2 Kubernetes Deployment

For orchestration and management of the containerized applications, we utilized Kubernetes [15]. Kubernetes automates deployment, scaling, and management of containerized applications. We packaged our system into a Helm chart, which simplifies the deployment of the entire application, including dependencies like MLflow, PostgreSQL, and Airflow.

Helm Chart Packaging

By encapsulating our services and their dependencies into a Helm chart, we streamlined the deployment process. Helm charts define, install, and upgrade complex Kubernetes applications, allowing us to manage configurations and versioning efficiently.

Database Migrations

Database schema changes are managed using migration files and scripts. Changes are first applied locally for testing and validation. Once validated, migrations are executed on the production database using scripts designed to apply changes incrementally. This process ensures that the database schema remains synchronized with the application code without disrupting ongoing operations.

6.3 Deployment on Azure AKS

6.3.1 Azure Kubernetes Service (AKS)

We deployed our Kubernetes cluster on Microsoft Azure using Azure Kubernetes Service (AKS), a managed Kubernetes service that simplifies cluster management by handling critical tasks like health monitoring and maintenance. AKS reduces the operational overhead and provides features like automated scaling and updates.

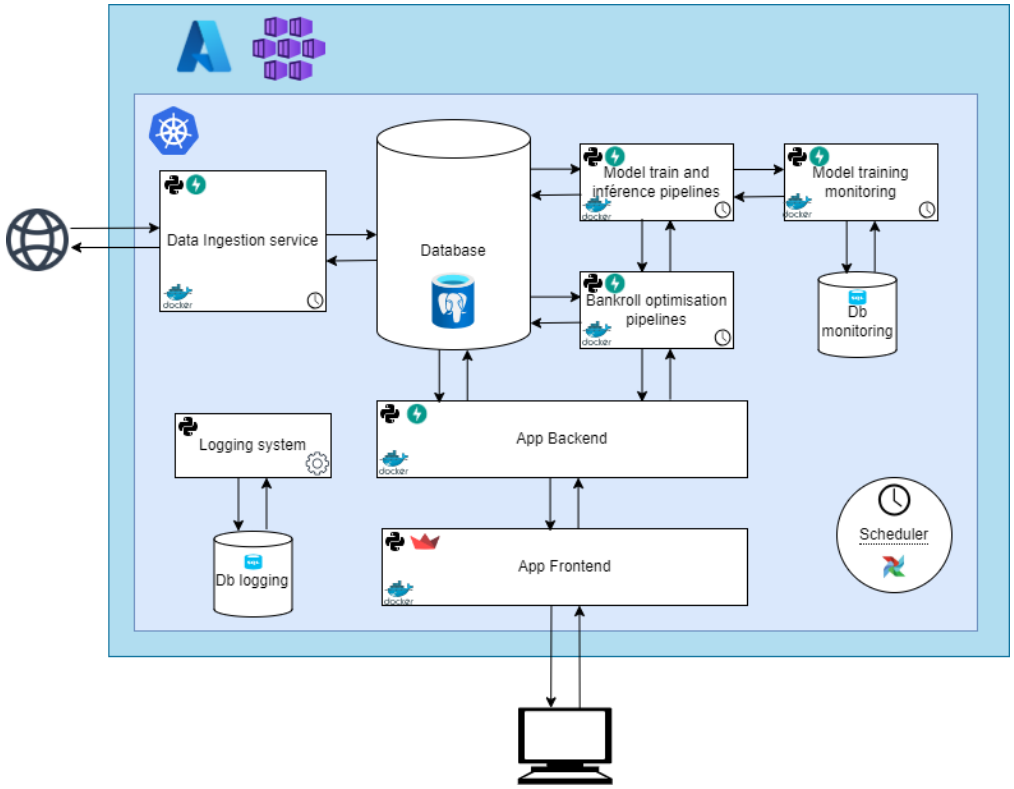

Figure 6.1: Architecture of the system deployed on AKS

6.3.2 Infrastructure Details

Our AKS deployment utilizes two virtual machines to ensure high availability and load balancing across the cluster. While Azure offers its own virtual machines, in this context, we refer to the compute resources allocated to our Kubernetes nodes. The integration with AKS allows for efficient resource utilization and scalability.

6.3.3 Azure Services Integration

Using Azure's cloud infrastructure offers several benefits:

- **Azure Container Registry (ACR)**: Stores our Docker images securely, facilitating seamless deployment to AKS.
- **Azure DevOps Repo**: Provides a storage for our code.

6.3.4 Pricing Considerations

Azure's pricing model charges for compute resources used by virtual machines, storage, and network bandwidth. Managed services like AKS can reduce operational overhead but require careful monitoring to manage costs effectively. We optimized resource allocation by:

6.4 Conclusion

By adopting a microservices architecture, containerization with Docker, orchestration with Kubernetes, and deploying on Azure AKS, we built a scalable, reliable, and maintainable system for sports betting optimization. This architecture allows for independent development and deployment of components, ensuring the

system can adapt to changing requirements and handle real-time data processing demands efficiently. Leveraging cloud infrastructure and managed services enhances our ability to focus on core application development while ensuring high availability and performance.

Chapter 7

Discussion and Conclusion

7.1 Summary of Findings

This study embarked on the ambitious task of developing a comprehensive system for optimizing sports betting strategies, focusing on football matches. Through the integration of predictive modeling, utility-based optimization, and scalable system architecture, we have addressed the critical components necessary for successful sports betting.

The predictive model, developed using logistic regression and advanced feature selection techniques, demonstrated significant accuracy in forecasting match outcomes. Regular retraining of the model proved essential in maintaining performance over time, highlighting the dynamic nature of sports data.

The optimization module applied various bankroll allocation strategies, including the Kelly Criterion, logarithmic, exponential, and linear utility functions. Both Monte Carlo simulations and real-world online testing over a five-week period indicated that sophisticated utility-based strategies substantially outperform naive betting approaches. Strategies like the Kelly Criterion and Exponential Utility provided favorable returns while effectively managing risk.

The system's deployment on Azure Kubernetes Service (AKS) showcased its scalability and readiness for real-time application. By leveraging a microservices architecture and containerization technologies like Docker and Kubernetes, the system can handle the computational demands of real-time data processing and optimization.

7.2 Contributions to the Field

This work contributes to the field of sports analytics and betting strategies in several ways:

- **Integration of Predictive Modeling and Optimization**: By combining accurate probability estimations with utility-based optimization strategies, the system provides a robust framework for sports betting.
- **Scalable System Architecture**: The implementation of a microservices architecture and deployment on cloud infrastructure ensures that the system is scalable, maintainable, and adaptable to real-world conditions.
- **Empirical Evaluation**: The use of both simulations and real-world testing provides empirical evidence of the effectiveness of advanced betting strategies over simpler methods.

7.3 Limitations

Despite the positive results, several limitations were identified:

- **Predictive Model Enhancements**: While the current model performs adequately within the constraints of a static framework, it could be significantly improved by incorporating additional features, conducting hyperparameter optimization, and exploring more complex models such as deep learning

architectures. These enhancements would allow the model to capture dynamic patterns and temporal dependencies inherent in football matches, which are not fully addressed due to the static nature of the current framework.

- **Static Framework Limitations and Long-Term Gain-Variance Interpretation**: The reduction of the betting problem to a static framework simplifies the optimization process but introduces limitations in interpreting gains and variance over the long term. Since the model does not account for intertemporal dependencies and the evolving nature of the bankroll, the strategies derived may not fully capture the risks associated with long-term betting. This static approach may lead to strategies that optimize short-term gains without adequately considering the cumulative effect on wealth over time. Future work should focus on extending the framework to a dynamic setting, allowing for a more accurate interpretation of long-term gain and variance, and better aligning the strategies with the bettor's long-term financial goals.

- **Risk Preferences and Dynamic Adaptation**: The optimization strategies employed fixed parameters for risk aversion, which do not adjust to changes in the bettor's wealth or market conditions over time. This static treatment of risk preferences limits the adaptability of the betting strategies, especially in a long-term context where the bettor's financial situation and the market dynamics can vary significantly. Introducing dynamic risk preferences that evolve with the bettor's bankroll and external factors would enhance the strategies' responsiveness and effectiveness, leading to better management of gain and variance over the long term.

- **Testing Period and Scope**: The real-world testing was confined to a five-week period focusing on the top five European leagues. Due to the static framework and the short testing duration, the evaluation may not fully reflect the strategies' performance over extended periods or in different market conditions. A longer testing period encompassing a broader range of leagues and varying competitive environments would provide more comprehensive insights into the strategies' long-term viability and their ability to manage gains and risks effectively within a dynamic setting.

7.4 Future Work

Building upon the findings of this study, several promising avenues can be explored to enhance the system's performance and address the challenges identified.

Firstly, integrating real-time data streams and developing adaptive predictive models could significantly improve forecasting accuracy. By incorporating techniques from time-series analysis and machine learning, the model can capture temporal dependencies and evolving patterns inherent in football matches. This dynamic approach would allow the model to adjust to new information promptly, potentially leading to more accurate probability estimates and better alignment with the actual match outcomes.

Secondly, advancing the optimization strategies to include stochastic elements and multi-period planning could address the complexities associated with long-term gain and variance interpretation. Developing a dynamic framework that accounts for intertemporal dependencies and the evolving nature of the bankroll would enable more effective risk management. Strategies that adapt risk preferences in response to changes in the bettor's financial status or market conditions could lead to more sustainable betting practices and improved long-term financial outcomes.

Thirdly, conducting extensive real-world testing over longer periods and across a broader range of leagues and competitions would provide deeper insights into the robustness and generalizability of the betting strategies. Such testing would help to evaluate the performance of the models under varying market conditions and competitive environments, ensuring that the strategies remain effective over time and are not limited to specific contexts or short-term scenarios.

Finally, enhancing the user interface to offer more advanced analytics and personalized insights could empower users to make more informed decisions. Features that allow users to visualize performance trends, adjust parameters interactively, and receive tailored recommendations would improve the overall user experience. Providing tools for long-term performance monitoring and strategic adjustments would enable users to better understand the implications of their betting decisions and manage their bankrolls more effectively.

These potential developments represent initial steps toward refining the system's capabilities. By focusing on dynamic modeling, adaptive optimization, comprehensive testing, and user-centric design, future work can

contribute to more robust predictive performance, effective risk management, and ultimately, more successful sports betting strategies.

7.5 Final Remarks

The integration of predictive modeling and utility-based optimization represents a significant step forward in developing effective sports betting strategies. This work demonstrates that with accurate predictions and strategic bankroll management, it is possible to achieve superior returns while managing risk effectively. The deployment on cloud infrastructure ensures that the system is ready for practical application, paving the way for future advancements in the field.

Appendix A

List of Notations

A.1 General Notations

$t \in \mathbb{R}^+$: Time.

$\mathbb{M}(t) = \{m^1, m^2, \ldots, m^{M(t)}\}$: Set of matches available for betting at time t.

$M(t) \in \mathbb{N}$: Total number of matches available at time t.

$m^k \in \mathbb{M}(t)$: A specific match k.

$\Omega^k = \{\omega_1^k, \omega_2^k, \ldots, \omega_{N^k}^k\}$: Set of possible outcomes for match m^k.

$N^k \in \mathbb{N}$: Number of possible outcomes for match m^k.

ω_i^k: Outcome i of match m^k.

A.2 Probabilities of Outcomes

$\mathbb{P}_Y(\omega_i^k)$: Probability that outcome ω_i^k occurs for match m^k at time t.

$r_i^k(t) = \mathbb{P}_Y(\omega_i^k)$: Probability of outcome ω_i^k occurring at time t.

X_i^k: Random variable indicating whether outcome ω_i^k occurs:

$$X_i^k = \begin{cases} 1, & \text{if outcome } \omega_i^k \text{ occurs,} \\ 0, & \text{otherwise.} \end{cases}$$

A.3 Bettors and Bookmakers

$\mathbb{J}$: Set of bettors.

$\mathbb{B}$: Set of bookmakers.

$B^J_{\text{bettor}}(t)$: Bankroll of bettor J at time t.

$B^B_{\text{bookmaker}}(t)$: Bankroll of bookmaker B at time t.

A.4 Odds

$\mathbb{O}^k(B,t) = \{o_1^{k,B}(t), o_2^{k,B}(t), \ldots, o_{N^k}^{k,B}(t)\}$: Set of odds offered by bookmaker B for match m^k at time t.

$o_i^{k,B}(t)$: Odds offered by bookmaker B for outcome ω_i^k of match m^k.

A.5 Bets and Wagers

$f_i^{k,J}(t)$: Fraction of bettor J's bankroll wagered on outcome ω_i^k at time t.

$b_i^{k,J}(t)$: Bookmaker with whom bettor J places the bet on outcome ω_i^k at time t.

$w_i^{k,J}(t) = f_i^{k,J}(t) \times B^J_{\text{bettor}}(t)$: Amount wagered by bettor J on outcome ω_i^k at time t.

A.6 Bankroll Evolution

$\mathcal{B}^J_{\text{settled}}(\tau)$: Set of bets settled for bettor J at time τ.

$G^J_{\text{bettor}}(b)$: Gain or loss from bet b for bettor J:

$$G^J_{\text{bettor}}(b) = w^J(b) \times \big(o^B(b) \times X(b) - 1\big)$$

$w^J(b)$: Amount wagered on bet b.

$o^B(b)$: Odds offered by bookmaker B for bet b.

$X(b)$: Indicator variable for whether bet b wins.

A.7 Bookmaker's Gain

$\mathcal{J}$: Set of bettors placing bets with bookmaker B.

$G^B_{\text{bookmaker}}(b)$: Gain or loss for bookmaker B from bet b:

$$G^B_{\text{bookmaker}}(b) = w^J(b) \times \big(1 - o^B(b) \times X(b)\big)$$

A.8 Bankroll Factors

$BF^J_{\text{bettor}}(t) = \frac{B^J_{\text{bettor}}(t)}{B^J_{\text{bettor}}(0)}$: Bankroll factor for bettor J.

$BF^B_{\text{bookmaker}}(t) = \frac{B^B_{\text{bookmaker}}(t)}{B^B_{\text{bookmaker}}(0)}$: Bankroll factor for bookmaker B.

A.9 Gain Calculation

$G^J_{\text{bettor}}(t) = B^J_{\text{bettor}}(t) - B^J_{\text{bettor}}(0) = B^J_{\text{bettor}}(0)\,\big(BF^J_{\text{bettor}}(t) - 1\big)$: Gain for bettor J.

$G^B_{\text{bookmaker}}(t) = B^B_{\text{bookmaker}}(t) - B^B_{\text{bookmaker}}(0) = B^B_{\text{bookmaker}}(0)\,\big(BF^B_{\text{bookmaker}}(t) - 1\big)$: Gain for bookmaker B.

A.10 Utility Functions

$U(B)$: Utility function for wealth B.

Expected value utility: $U(B) = B$.

Logarithmic utility: $U(B) = \ln(B)$.

Power utility: $U(B) = \frac{B^{1-\gamma}}{1-\gamma}, \quad \gamma \neq 1$.

Exponential utility: $U(B) = -e^{-\alpha B}$.

Quadratic utility: $U(B) = B - \frac{\lambda}{2}B^2$.

A.11 Agent's State Space

$S(t)$: State of the betting market at time t:

$$S(t) = (\mathbb{M}(t), \Omega(t), \mathbb{O}(t), B_{\text{bettor}}(t), B_{\text{bookmaker}}(t), H(t), \mathcal{I}(t))$$

$H(t)$: History of past events.

$\mathcal{I}(t)$: Additional information available to agents at time t.

A.12 Action Space

$A^J_{\text{bettor}}(t)$: Action of bettor J at time t, choosing f_i^k values.

$A^B_{\text{bookmaker}}(t)$: Action of bookmaker B at time t, setting odds $o_i^k(t)$.

A.13 Transition Dynamics

$\frac{dS(t)}{dt} = \Phi\left(S(t), A_{\text{bettor}}(t), A_{\text{bookmaker}}(t), \epsilon(t)\right)$: Transition function.

A.14 Policies

π^J_{bettor}: Policy for bettor J, mapping states to actions.

$\pi^B_{\text{bookmaker}}$: Policy for bookmaker B, mapping states to actions.

Appendix B

Analytical Solution Using the Kelly Criterion

B.1 Derivation of the Optimal Betting Fraction

The bettor seeks to maximize:

$$\max_{\{f_i^{k,J}(t)\}} \mathbb{E}_{p^J}\left[\ln\left(B^J_{\text{bettor}}(t+1)\right)\right]$$

Since:

$$B^J_{\text{bettor}}(t+1) = B^J_{\text{bettor}}(t) + G^J_{\text{bettor}}(t)$$

and the gain is:

$$G^J_{\text{bettor}}(t) = B^J_{\text{bettor}}(t)\sum_{k=1}^{M}\sum_{i=1}^{N^k} f_i^{k,J}(t)\left(o_i^{k,B}(t)X_i^k - 1\right)$$

we can write:

$$\ln\left(B^J_{\text{bettor}}(t+1)\right) = \ln\left(B^J_{\text{bettor}}(t)\right) + \ln\left(1 + \sum_{k=1}^{M}\sum_{i=1}^{N^k} f_i^{k,J}(t)\left(o_i^{k,B}(t)X_i^k - 1\right)\right)$$

Assuming that the fractions $f_i^{k,J}(t)$ are small, we can approximate the logarithm using a Taylor expansion around 0:

$$\ln(1+\delta) \approx \delta - \frac{\delta^2}{2}$$

Applying this approximation:

$$\ln\left(B^J_{\text{bettor}}(t+1)\right) \approx \ln\left(B^J_{\text{bettor}}(t)\right) + \Delta - \frac{\Delta^2}{2}$$

where:

$$\Delta = \sum_{k=1}^{M}\sum_{i=1}^{N^k} f_i^{k,J}(t)\left(o_i^{k,B}(t)X_i^k - 1\right)$$

Taking the expectation with respect to the bettor's estimated probabilities $p_i^{k,J}$:

$$\mathbb{E}_{p^J}\left[\ln\left(B^J_{\text{bettor}}(t+1)\right)\right] \approx \ln\left(B^J_{\text{bettor}}(t)\right) + \mathbb{E}_{p^J}\left[\Delta\right] - \frac{1}{2}\mathbb{E}_{p^J}\left[\Delta^2\right]$$

Computing the expected value and variance:

$$\mathbb{E}_{p^J}[\Delta] = \sum_{k=1}^{M}\sum_{i=1}^{N^k} f_i^{k,J}(t)\left(o_i^{k,B}(t)p_i^{k,J} - 1\right)$$

$$\mathbb{E}_{p^J}\left[\Delta^2\right] = \left(\mathbb{E}_{p^J}[\Delta]\right)^2 + \mathrm{Var}_{p^J}(\Delta)$$

Thus:

$$\mathbb{E}_{p^J}\left[\ln\left(B^J_{\text{bettor}}(t+1)\right)\right] \approx \ln\left(B^J_{\text{bettor}}(t)\right) + \mathbb{E}_{p^J}[\Delta] - \frac{1}{2}\left(\left(\mathbb{E}_{p^J}[\Delta]\right)^2 + \mathrm{Var}_{p^J}(\Delta)\right)$$

Since $\left(\mathbb{E}_{p^J}[\Delta]\right)^2$ is typically small compared to the variance term, we can focus on maximizing:

$$\mathbb{E}_{p^J}[\Delta] - \frac{1}{2}\mathrm{Var}_{p^J}(\Delta)$$

This shows that the bettor's optimization involves a trade-off between expected return and risk, both of which depend on the estimated probabilities $p_i^{k,J}$.

B.2 Optimal Betting Fraction

By taking the derivative with respect to $f_i^{k,J}(t)$ and setting it to zero:

$$\frac{\partial}{\partial f_i^{k,J}(t)}\left(\mathbb{E}_{p^J}[\Delta] - \frac{1}{2}\mathrm{Var}_{p^J}(\Delta)\right) = 0$$

After calculation, the optimal fraction is:

$$f_i^{k,J*}(t) = \frac{\left(o_i^{k,B}(t)p_i^{k,J} - 1\right)}{o_i^{k,B}(t) - 1}$$

This is the classical Kelly formula, showing that the optimal betting fraction depends on:

- The bettor's estimated probability $p_i^{k,J}$.
- The odds offered $o_i^{k,B}(t)$.

Appendix C

Analytical Reduction of $\mathbb{E}[\ln(B)]$

C.1 Problem Setup

The bettor aims to maximize the expected logarithmic utility of their bankroll:

$$\max_{\{f_i^{k,J}(t)\}} \mathbb{E}_{p^J}\left[\ln\left(B^J_{\text{bettor}}(t+1)\right)\right]$$

where:

$$B^J_{\text{bettor}}(t+1) = B^J_{\text{bettor}}(t) \times \text{BF}(t+1)$$

The **bankroll factor** $\text{BF}(t+1)$ is given by:

$$\text{BF}(t+1) = 1 + \sum_{k=1}^{M}\sum_{i=1}^{N^k} f_i^{k,J}(t)\left(o_i^{k,B}(t)X_i^k - 1\right)$$

However, this can be reformulated as:

$$\text{BF}(t+1) = 1 - F(t) + \sum_{k=1}^{M}\sum_{i=1}^{N^k} f_i^{k,J}(t)o_i^{k,B}(t)X_i^k$$

where:

- $F(t) = \sum_{k=1}^{M}\sum_{i=1}^{N^k} f_i^{k,J}(t)$ is the total fraction of the bankroll bet.

In this context, the expected value of the logarithmic utility of the future bankroll is:

$$\mathbb{E}_{p^J}\left[\ln\left(B^J_{\text{bettor}}(t+1)\right)\right] = \ln\left(B^J_{\text{bettor}}(t)\right) + \mathbb{E}_{p^J}\left[\ln\left(\text{BF}(t+1)\right)\right]$$

Our goal is to compute $\mathbb{E}_{p^J}\left[\ln\left(\text{BF}(t+1)\right)\right]$ without any approximations.

C.2 Expected Value of the Logarithm of the Bankroll Factor

Given that the matches are independent and the outcomes within each match are mutually exclusive, we can consider the bankroll factor as the product of the individual match factors.

For each match k: The bankroll factor for match k is:

$$\text{BF}_k = 1 - F_k + \sum_{i=1}^{N^k} f_i^{k,J}(t)o_i^{k,B}(t)X_i^k$$

where:

- $F_k = \sum_{i=1}^{N^k} f_i^{k,J}(t)$ is the fraction of the bankroll wagered on match k.
- X_i^k is a random variable that indicates whether outcome i of match k occurs ($X_i^k = 1$) or not ($X_i^k = 0$).

Since only one outcome occurs for each match, we can write BF_k for match k as:

$$\mathrm{BF}_k = 1 - F_k + f_{i^*}^{k,J}(t) o_{i^*}^{k,B}(t)$$

where i^* is the realized outcome of match k.

Total Bankroll Factor: Since matches are independent, the total bankroll factor is the product of the factors of each match:

$$\mathrm{BF}(t+1) = \prod_{k=1}^{M} \mathrm{BF}_k$$

C.3 Expected Logarithm of the Bankroll Factor

The expected value of the logarithm of the total bankroll factor is:

$$\mathbb{E}_{p^J}\left[\ln\left(\mathrm{BF}(t+1)\right)\right] = \sum_{k=1}^{M} \mathbb{E}_{p^J}\left[\ln\left(\mathrm{BF}_k\right)\right]$$

For a single match k: For each match k, the expectation is:

$$\mathbb{E}_{p^J}\left[\ln\left(\mathrm{BF}_k\right)\right] = \sum_{i=1}^{N^k} p_i^{k,J} \ln\left(1 - F_k + f_i^{k,J}(t) o_i^{k,B}(t)\right)$$

where $p_i^{k,J}$ is the bettor's estimated probability for outcome i of match k.

C.4 Final Expression for the Expected Logarithm of the Future Bankroll

Combining the expressions for all matches, we obtain:

$$\mathbb{E}_{p^J}\left[\ln\left(B^J_{\text{bettor}}(t+1)\right)\right] = \ln\left(B^J_{\text{bettor}}(t)\right) + \sum_{k=1}^{M}\left[\sum_{i=1}^{N^k} p_i^{k,J} \ln\left(1 - F_k + f_i^{k,J}(t) o_i^{k,B}(t)\right)\right]$$

This expression makes no assumption about the smallness of the betting fractions $f_i^{k,J}(t)$ or the return factor, and is therefore exact.

C.5 Optimization Without Approximation

The bettor must solve the following optimization problem:

$$\max_{\{f_i^{k,J}(t)\}} \sum_{k=1}^{M}\left[\sum_{i=1}^{N^k} p_i^{k,J} \ln\left(1 - F_k + f_i^{k,J}(t) o_i^{k,B}(t)\right)\right]$$

subject to the constraints:

- $f_i^{k,J}(t) \in [0,1]$ for all i, k.
- $F_k = \sum_{i=1}^{N^k} f_i^{k,J}(t) \leq 1$ for all k
- $\sum_{k=1}^{M} F_k \leq 1$

C.6 Example of Simplification

To illustrate this method, consider a match k with two possible outcomes (e.g., win or loss):

- Outcomes: $i = 1, 2$
- Estimated probabilities: $p_1^{k,J}$, $p_2^{k,J}$
- Betting fractions: $f_1^{k,J}(t)$, $f_2^{k,J}(t)$
- Odds: $o_1^{k,B}(t)$, $o_2^{k,B}(t)$
- $F_k = f_1^{k,J}(t) + f_2^{k,J}(t)$

The expected logarithm of the return factor for this match is:

$$\mathbb{E}_{p^J}\left[\ln\left(\mathrm{BF}_k\right)\right] = p_1^{k,J} \ln\left(1 - F_k + f_1^{k,J}(t) o_1^{k,B}(t)\right) + p_2^{k,J} \ln\left(1 - F_k + f_2^{k,J}(t) o_2^{k,B}(t)\right)$$

The bettor must choose $f_1^{k,J}(t)$ and $f_2^{k,J}(t)$ to maximize this expression, subject to the constraints:

- $f_1^{k,J}(t) \geq 0$
- $f_2^{k,J}(t) \geq 0$
- $f_1^{k,J}(t) + f_2^{k,J}(t) \leq 1$

C.7 Numerical Optimization

This optimization can be solved analytically in some simple cases, or more generally using numerical methods such as nonlinear optimization algorithms (e.g., Newton-Raphson, gradient-based methods).

C.8 The Role of Estimated Probabilities

The estimated probabilities $p_i^{k,J}$ directly influence the expected logarithm of the return factor. A higher estimated probability for a particular outcome increases the weight of the logarithmic return factor associated with that outcome in the overall expectation. Thus, the bettor is incentivized to bet more on outcomes they believe are more likely to occur, while considering the offered odds.

C.9 Conclusion

In conclusion, it is entirely possible to analytically compute $\mathbb{E}_{p^J}\left[\ln\left(B^J_{\text{bettor}}(t+1)\right)\right]$ without assuming that the return factor is small. This allows the bettor to optimize their betting strategy by fully considering the impact of each wager on their future bankroll, without approximation.

Appendix D

Analytical Reduction Using the Exponential Utility Function

To further illustrate how estimated probabilities influence the optimization problem, consider the case where the bettor uses an exponential utility function:

$$U(B) = -e^{-\alpha B}$$

where $\alpha > 0$ is the coefficient of absolute risk aversion (CARA). This utility function represents a bettor whose absolute risk aversion remains constant regardless of wealth level.

D.1 Derivation of the Optimal Betting Fraction

The bettor seeks to maximize the expected utility:

$$\max_{\{f_i^{k,J}(t)\}} \mathbb{E}_{p^J}\left[U\left(B^J_{\text{bettor}}(t+1)\right)\right]$$

Substituting $B^J_{\text{bettor}}(t+1) = B^J_{\text{bettor}}(t) + G^J_{\text{bettor}}(t)$, we have:

$$\mathbb{E}_{p^J}\left[U\left(B^J_{\text{bettor}}(t+1)\right)\right] = \mathbb{E}_{p^J}\left[-e^{-\alpha\left(B^J_{\text{bettor}}(t)+G^J_{\text{bettor}}(t)\right)}\right]$$

Since $B^J_{\text{bettor}}(t)$ is constant with respect to the expectation, we can factor out $e^{-\alpha B^J_{\text{bettor}}(t)}$:

$$\mathbb{E}_{p^J}\left[U\left(B^J_{\text{bettor}}(t+1)\right)\right] = -e^{-\alpha B^J_{\text{bettor}}(t)}\mathbb{E}_{p^J}\left[e^{-\alpha G^J_{\text{bettor}}(t)}\right]$$

The gain $G^J_{\text{bettor}}(t)$ is given by:

$$G^J_{\text{bettor}}(t) = B^J_{\text{bettor}}(t)\sum_{k=1}^{M}\sum_{i=1}^{N^k} f_i^{k,J}(t)\left(o_i^{k,B}(t)X_i^k - 1\right)$$

Substituting this expression into the expected utility:

$$\mathbb{E}_{p^J}\left[U\left(B^J_{\text{bettor}}(t+1)\right)\right] = -e^{-\alpha B^J_{\text{bettor}}(t)}\mathbb{E}_{p^J}\left[e^{-\alpha B^J_{\text{bettor}}(t)\sum_{k=1}^{M}\sum_{i=1}^{N^k} f_i^{k,J}(t)\left(o_i^{k,B}(t)X_i^k-1\right)}\right]$$

Simplify the exponent:

$$-\alpha B^J_{\text{bettor}}(t)\sum_{k=1}^{M}\sum_{i=1}^{N^k} f_i^{k,J}(t)\left(o_i^{k,B}(t)X_i^k - 1\right) = -\alpha B^J_{\text{bettor}}(t)\left(\sum_{k=1}^{M}\left(\sum_{i=1}^{N^k} f_i^{k,J}(t)o_i^{k,B}(t)X_i^k\right) - F(t)\right)$$

where $F(t) = \sum_{k=1}^{M}\sum_{i=1}^{N^k} f_i^{k,J}(t)$ is the total fraction of the bankroll wagered.

We can rewrite the exponent as:

$$-\alpha B^J_{\text{bettor}}(t)\left(\sum_{k=1}^{M} R_k - F(t)\right)$$

where $R_k = \sum_{i=1}^{N^k} f_i^{k,J}(t) o_i^{k,B}(t) X_i^k$ represents the return from match k.
Thus, the expected utility becomes:

$$\mathbb{E}_{p^J}\left[U\left(B^J_{\text{bettor}}(t+1)\right)\right] = -e^{-\alpha B^J_{\text{bettor}}(t)} e^{\alpha B^J_{\text{bettor}}(t) F(t)} \mathbb{E}_{p^J}\left[e^{-\alpha B^J_{\text{bettor}}(t)\sum_{k=1}^{M} R_k}\right]$$

Since the matches are independent and the outcomes within each match are mutually exclusive, we can factor the expectation:

$$\mathbb{E}_{p^J}\left[e^{-\alpha B^J_{\text{bettor}}(t)\sum_{k=1}^{M} R_k}\right] = \prod_{k=1}^{M} \mathbb{E}_{p^J}\left[e^{-\alpha B^J_{\text{bettor}}(t) R_k}\right]$$

For each match k, the expectation over the outcomes is:

$$\mathbb{E}_{p^J}\left[e^{-\alpha B^J_{\text{bettor}}(t) R_k}\right] = \sum_{i=1}^{N^k} p_i^{k,J} e^{-\alpha B^J_{\text{bettor}}(t) f_i^{k,J}(t) o_i^{k,B}(t)}$$

Combining these results, the expected utility simplifies to:

$$\mathbb{E}_{p^J}\left[U\left(B^J_{\text{bettor}}(t+1)\right)\right] = -e^{-\alpha B^J_{\text{bettor}}(t)(1-F(t))} \prod_{k=1}^{M}\left(\sum_{i=1}^{N^k} p_i^{k,J} e^{-\alpha B^J_{\text{bettor}}(t) f_i^{k,J}(t) o_i^{k,B}(t)}\right)$$

The bettor's objective is to choose the fractions $f_i^{k,J}(t)$ that maximize this expected utility. However, since the utility function is negative, maximizing the expected utility is equivalent to minimizing:

$$\mathcal{L} = e^{-\alpha B^J_{\text{bettor}}(t)(1-F(t))} \prod_{k=1}^{M}\left(\sum_{i=1}^{N^k} p_i^{k,J} e^{-\alpha B^J_{\text{bettor}}(t) f_i^{k,J}(t) o_i^{k,B}(t)}\right)$$

To simplify the optimization, we can take the natural logarithm and consider the negative of the expected utility (since the exponential utility function is negative):

$$\min_{\{f_i^{k,J}(t)\}} -\ln\left(-\mathbb{E}_{p^J}\left[U\left(B^J_{\text{bettor}}(t+1)\right)\right]\right)$$

Computing the logarithm:

$$-\ln\left(-\mathbb{E}_{p^J}\left[U\left(B^J_{\text{bettor}}(t+1)\right)\right]\right) = \alpha B^J_{\text{bettor}}(t)(1-F(t)) - \sum_{k=1}^{M} \ln\left(\sum_{i=1}^{N^k} p_i^{k,J} e^{-\alpha B^J_{\text{bettor}}(t) f_i^{k,J}(t) o_i^{k,B}(t)}\right)$$

The bettor's optimization problem reduces to minimizing this expression with respect to $\{f_i^{k,J}(t)\}$.

D.2 Certainty Equivalent Interpretation

Alternatively, we can interpret the optimization in terms of the *certainty equivalent* CE, which satisfies:

$$U(CE) = \mathbb{E}_{p^J}\left[U\left(B^J_{\text{bettor}}(t+1)\right)\right]$$

Using the exponential utility function:

$$-e^{-\alpha CE} = -e^{-\alpha B^J_{\text{bettor}}(t)} \mathbb{E}_{p^J}\left[e^{-\alpha G^J_{\text{bettor}}(t)}\right]$$

Simplifying:

$$e^{-\alpha CE} = e^{-\alpha B^J_{\text{bettor}}(t)} \mathbb{E}_{p^J}\left[e^{-\alpha G^J_{\text{bettor}}(t)}\right]$$

Therefore:

$$CE = B^J_{\text{bettor}}(t) - \frac{1}{\alpha} \ln\left(\mathbb{E}_{p^J}\left[e^{-\alpha G^J_{\text{bettor}}(t)}\right]\right)$$

Using the previous results, the certainty equivalent becomes:

$$CE = B^J_{\text{bettor}}(t) - \frac{1}{\alpha} \ln\left(e^{\alpha B^J_{\text{bettor}}(t) F(t)} \prod_{k=1}^{M} \left(\sum_{i=1}^{N^k} p_i^{k,J} e^{-\alpha B^J_{\text{bettor}}(t) f_i^{k,J}(t) o_i^{k,B}(t)}\right)\right)$$

Simplifying the logarithm:

$$CE = B^J_{\text{bettor}}(t) - B^J_{\text{bettor}}(t)F(t) - \frac{1}{\alpha} \sum_{k=1}^{M} \ln\left(\sum_{i=1}^{N^k} p_i^{k,J} e^{-\alpha B^J_{\text{bettor}}(t) f_i^{k,J}(t) o_i^{k,B}(t)}\right)$$

This expression shows that the certainty equivalent depends on:

- The initial bankroll $B^J_{\text{bettor}}(t)$.
- The total fraction wagered $F(t)$.
- The estimated probabilities $p_i^{k,J}$.
- The odds $o_i^{k,B}(t)$.
- The betting fractions $f_i^{k,J}(t)$.
- The risk aversion parameter α.

D.3 Optimization Problem

The bettor's optimization problem is to choose $\{f_i^{k,J}(t)\}$ to maximize the certainty equivalent CE:

$$\max_{\{f_i^{k,J}(t)\}} \left\{ CE = B^J_{\text{bettor}}(t)(1 - F(t)) - \frac{1}{\alpha} \sum_{k=1}^{M} \ln\left(\sum_{i=1}^{N^k} p_i^{k,J} e^{-\alpha B^J_{\text{bettor}}(t) f_i^{k,J}(t) o_i^{k,B}(t)}\right)\right\}$$

Subject to the constraints:

- Non-negativity: $f_i^{k,J}(t) \geq 0$ for all i, k.
- Budget constraint: $F(t) = \sum_{k=1}^{M} \sum_{i=1}^{N^k} f_i^{k,J}(t) \leq 1$.

D.4 Role of Estimated Probabilities

The estimated probabilities $p_i^{k,J}$ enter the optimization problem explicitly in the logarithmic terms of the certainty equivalent. They affect the expected utility by weighting the potential outcomes according to the bettor's beliefs.

A higher estimated probability $p_i^{k,J}$ for a particular outcome increases the weight of the term $e^{-\alpha B^J_{\text{bettor}}(t) f_i^{k,J}(t) o_i^{k,B}(t)}$ in the logarithm. This, in turn, influences the optimal betting fraction $f_i^{k,J}(t)$ assigned to that outcome.

D.5 Interpretation

The exponential utility function leads to an optimization that balances the expected returns against the risk, adjusted for the bettor's absolute risk aversion α. The bettor allocates their bets to maximize the certainty equivalent, effectively trading off potential gains against the disutility of risk.

The presence of α in the exponentials and logarithms quantifies the bettor's sensitivity to risk. A higher α implies greater risk aversion, leading the bettor to wager smaller fractions $f_i^{k,J}(t)$.

D.6 Conclusion

Using the exponential utility function demonstrates how estimated probabilities $p_i^{k,J}$ influence the bettor's optimal strategy. The optimization problem incorporates these probabilities directly, affecting the allocation of bets across different outcomes and matches. The bettor must consider both their beliefs about the likelihood of outcomes and their risk preferences to determine the optimal betting fractions.

This analytical solution provides insight into the interplay between estimated probabilities, risk aversion, and optimal betting strategies under constant absolute risk aversion.

Appendix E

Derivation of the linear Objective Function

This appendix provides a formal derivation of the objective function $E(B) - \lambda \cdot \text{Var}(B)$ from a quadratic utility function.

E.1 Quadratic Utility Function

Consider a quadratic utility function of the form:

$$U(B) = B - \frac{\lambda}{2}B^2$$

where B is the wealth (or bankroll), and λ is a constant representing the individual's risk aversion. The function is concave, capturing the notion of diminishing marginal utility and aversion to risk.

E.2 Expected Utility

The expected utility is given by:

$$E[U(B)] = E\left[B - \frac{\lambda}{2}B^2\right] = E[B] - \frac{\lambda}{2}E[B^2]$$

The term $E[B^2]$ can be expressed as:

$$E[B^2] = \text{Var}(B) + E(B)^2$$

Substituting this into the expression for expected utility:

$$E[U(B)] = E[B] - \frac{\lambda}{2}\left(\text{Var}(B) + E(B)^2\right)$$

E.3 Simplification of the Objective Function

Expanding the expression:

$$E[U(B)] = E[B] - \frac{\lambda}{2}\text{Var}(B) - \frac{\lambda}{2}E(B)^2$$

To simplify, we assume that the term $\frac{\lambda}{2}E(B)^2$ is small enough to be negligible, or we focus on cases where the effect of the variance is more significant. Therefore, the expected utility becomes:

$$E[U(B)] \approx E(B) - \frac{\lambda}{2}\text{Var}(B)$$

Multiplying the entire expression by 2 (to match the typical form of the objective function):

$$E[U(B)] \approx E(B) - \lambda \cdot \text{Var}(B)$$

E.4 Conclusion

Thus, the objective function $E(B) - \lambda \cdot \mathrm{Var}(B)$ can be derived from a quadratic utility function, where λ represents the individual's degree of risk aversion. This form is widely used in decision theory and finance to model the trade-off between expected return and risk (variance).

Appendix F

Predictive model Metrics

Classic classification metrics offer foundational insights into model performance. However, their effectiveness can be limited in the presence of class imbalance. The following metrics are defined mathematically to facilitate precise evaluation:

• F.1 Accuracy

Accuracy measures the proportion of correctly predicted instances out of the total instances. It is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

where:

- TP: True Positives (correctly predicted positive instances),
- TN: True Negatives (correctly predicted negative instances),
- FP: False Positives (incorrectly predicted positive instances),
- FN: False Negatives (incorrectly predicted negative instances).

This overall accuracy is simple and intuitive, making it widely used. However, it may not provide a full picture in cases of class imbalance, as the performance on majority classes can dominate the overall score.

Accuracy by Class can be defined to better understand how the model performs on each class individually. For class i, the accuracy is:

$$\text{Accuracy}_i = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}$$

This metric is particularly useful when dealing with imbalanced datasets or when certain classes are more important than others. It allows for an in-depth evaluation of how well the model performs across different categories, helping to identify underperforming classes.

• F.2 Precision

Precision for a given class is the ratio of correctly predicted positive observations to the total predicted positives. For class i, it is defined as:

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i}$$

In multi-class classification problems, precision can be computed in different ways depending on how the classes are aggregated:

- **Micro-averaged Precision**: This metric calculates the precision by aggregating the true positives and false positives across all classes, treating them as a single combined class. It is particularly useful when the dataset is balanced or in multi-label problems. The micro-averaged precision is defined as:
$$\text{Precision}_{micro} = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} (TP_i + FP_i)}$$
- **Macro-averaged Precision**: In this approach, the precision is calculated for each class individually, and the unweighted average of these precisions is taken. It is useful when you want to treat each class equally, regardless of class size, and is often applied in imbalanced datasets to ensure fair evaluation. The macro-averaged precision is given by:
$$\text{Precision}_{macro} = \frac{1}{N} \sum_{i=1}^{N} \text{Precision}_i$$
- **Weighted Precision**: This is the weighted average of the precision scores for each class, with the weight being proportional to the number of instances in each class. It provides a more balanced view when the class distribution is highly imbalanced, as it gives more importance to the performance on the majority classes. The weighted precision is defined as:
$$\text{Precision}_{weighted} = \sum_{i=1}^{N} w_i \cdot \text{Precision}_i$$
where $w_i = \frac{\text{Number of examples in class } i}{\text{Total number of examples}}$.

• F.3 Recall

Recall (also known as Sensitivity) for a given class is the ratio of correctly predicted positive observations to all actual positives. For class i, it is defined as:

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i}$$

Similar to precision, recall can be aggregated in multiple ways for multi-class classification:

- **Micro-averaged Recall**: Aggregates the true positives and false negatives across all classes and calculates the recall as if the problem were binary. This approach is useful for optimizing the overall recall across all classes. The micro-averaged recall is defined as:
$$\text{Recall}_{micro} = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} (TP_i + FN_i)}$$
- **Macro-averaged Recall**: Computes the recall for each class and then takes the unweighted average across all classes. It is helpful when you want to treat each class equally, regardless of the class distribution. This is particularly useful in imbalanced datasets where recall for minority classes is critical. The macro-averaged recall is given by:
$$\text{Recall}_{macro} = \frac{1}{N} \sum_{i=1}^{N} \text{Recall}_i$$

– **Weighted Recall**: This metric calculates the recall for each class and then computes the weighted average, with the weights proportional to the size of each class. It is effective in imbalanced datasets where the goal is to ensure better recall for major classes. The weighted recall is defined as:

$$\text{Recall}_{weighted} = \sum_{i=1}^{N} w_i \cdot \text{Recall}_i$$

where $w_i = \frac{\text{Number of examples in class } i}{\text{Total number of examples}}$.

• F.4 F1-Score

The **F1-score** is a metric that combines both **precision** and **recall** into a single measure. It is particularly useful in scenarios where a balance between precision and recall is needed, especially when there is an uneven class distribution. The F1-score is defined as the harmonic mean of precision and recall:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

In multi-class or multi-label classification, different versions of the F1-score can be computed depending on how the performance across classes is aggregated:

– **Per-class F1-score**: The F1-score can be computed for each class individually without averaging. This gives insight into how the model performs on specific classes.

$$F1_i = 2 \cdot \frac{\text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}$$

– **Micro-averaged F1-score**: The micro F1-score is calculated by aggregating the true positives, false positives, and false negatives across all classes and then computing the F1-score from these totals. This treats all classes as a single combined class, ignoring class distributions.

$$F1_{\text{micro}} = 2 \cdot \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} (2 \cdot TP_i + FP_i + FN_i)}$$

– **Macro-averaged F1-score**: The macro F1-score computes the F1-score for each class individually and then takes the unweighted average of these scores. Each class is treated equally, regardless of its size.

$$F1_{\text{macro}} = \frac{1}{N} \sum_{i=1}^{N} F1_i$$

– **Weighted F1-score**: The weighted F1-score calculates the F1-score for each class and then takes the weighted average, where the weight is the proportion of instances of each class. This metric gives more importance to majority classes.

$$F1_{\text{weighted}} = \sum_{i=1}^{N} w_i \cdot F1_i$$

where w_i is the proportion of examples in class i:

$$w_i = \frac{\text{Number of examples in class } i}{\text{Total number of examples}}$$

• F.5 Classwise Expected Calibration Error (ECE)

Classwise Expected Calibration Error (ECE) is a metric that evaluates the calibration of predicted probabilities for each class individually in a multi-class classification setting. Calibration refers to the alignment between the predicted probabilities and the actual outcome frequencies. A well-calibrated model ensures that, for each class, the predicted probability reflects the true likelihood of that class being the correct outcome.

For each class i, the Classwise ECE is defined as:

$$\text{ECE}_{\text{classwise}} = \frac{1}{k} \sum_{i=1}^{k} \sum_{j=1}^{m} \frac{|B_{ij}|}{n_i} |y_{ij} - \bar{p}_{ij}|$$

where:

- k: **Number of classes**. Represents the total distinct categories the model can predict.
- m: **Number of bins**. The range of predicted probabilities is divided into m equal-width intervals (bins) to aggregate predictions.
- B_{ij}: **Set of instances for class i in bin j**. This is the subset of samples belonging to class i whose predicted probability for that class falls into bin j.
- n_i: **Total number of instances for class i**. It is the total count of samples belonging to class i across all bins.
- y_{ij}: **True frequency** of class i in bin j. Calculated as the ratio of correctly predicted instances of class i in bin j to the total number of instances of class i in that bin.
- $\bar{p}_{ij}$: **Average predicted probability** for class i in bin j. It is the mean of the predicted probabilities for class i for all instances in bin j.

The Classwise ECE provides a granular view of the model's calibration performance for each class. By computing the ECE separately for each class and then averaging, it accounts for potential class imbalances and ensures that the calibration assessment is not dominated by the majority classes. A lower Classwise ECE indicates better calibration, meaning the predicted probabilities closely match the observed frequencies. Conversely, a higher ECE suggests discrepancies between predictions and actual outcomes, signaling potential overconfidence or underconfidence in the model's predictions for specific classes.

• F.6 Log Loss

Log Loss evaluates the accuracy of probability estimates by penalizing false classifications. It is defined as:

$$\text{Log Loss} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log(p_{i,c})$$

where:

- N: Number of samples
- C: Number of classes
- $y_{i,c}$: Binary indicator (0 or 1) if class label c is the correct classification for sample i
- $p_{i,c}$: Predicted probability for class c of sample i

• F.7 Mean Squared Error (MSE)

Mean Squared Error (MSE) measures the average squared difference between predicted probabilities and actual outcomes. For class i, it is defined as:

$$\text{MSE}_i = \frac{1}{N} \sum_{j=1}^{N} (p_{j,i} - y_{j,i})^2$$

Appendix G

Ranking Features

- **Elo Score**

$$R'_A = R_A + K \times (S_A - E_A)$$
$$E_A = \frac{1}{1 + 10^{(R_B - R_A - H_{advantage})/400 - C}}$$

  Where:

  - R_A: Current Elo rating of Team A. The initial value for R used is 1500.
  - R'_A: Updated Elo rating of Team A after a match.
  - K: Development coefficient. It determines the sensitivity of Elo rating adjustments based on match outcomes, with a value of 25 chosen to balance responsiveness and stability, ensuring ratings accurately reflect team performance without excessive volatility.
  - S_A: Actual score for Team A in the match (1 for win, 0.5 for draw, 0 for loss).
  - E_A: Expected score for Team A.
  - R_B: Current Elo rating of Team B.
  - $H_{advantage}$: Home Advantage constant. It accounts for the inherent benefits of playing at home, such as supporter support and familiarity with the venue, with a value of 100 selected to represent a significant yet realistic home-field advantage based on historical performance data.
  - C: Corrective constant. It fine-tunes the expected score calculation by slightly adjusting the probability distribution, using a value of 0 to keep things simple.

- **Glicko-2 Score**

$$\mu = \frac{R - 1500}{173.7178}, \quad \phi = \frac{RD}{173.7178}$$
$$g(\phi_j) = \frac{1}{\sqrt{1 + \frac{3\phi_j^2}{\pi^2}}}$$
$$E(\mu, \mu_j, \phi_j) = \frac{1}{1 + e^{-g(\phi_j)(\mu - \mu_j)}}$$
$$v = \left(\sum_{j=1}^{k} g(\phi_j)^2 E(\mu, \mu_j, \phi_j)(1 - E(\mu, \mu_j, \phi_j)) \right)^{-1}$$
$$\delta = v \cdot \sum_{j=1}^{k} g(\phi_j)(s_j - E(\mu, \mu_j, \phi_j))$$

$$\text{Solve} \quad f(x) = \frac{e^x(\delta^2 - \phi^2 - v - e^x)}{2(\phi^2 + v + e^x)^2} - \frac{x - \ln(\sigma^2)}{\tau^2} = 0 \quad \text{pour} \quad \sigma'$$

$$\phi^* = \sqrt{\phi^2 + \sigma'^2}$$

$$\phi' = \left(\frac{1}{\phi^{*2}} + \frac{1}{v} \right)^{-1/2}$$

$$\mu' = \mu + \phi'^2 \cdot \sum_{j=1}^{k} g(\phi_j)(s_j - E(\mu, \mu_j, \phi_j))$$

$$R' = 173.7178 \cdot \mu' + 1500$$

$$RD' = 173.7178 \cdot \phi'$$

Where:

- R: Current rating.
- RD: Current rating deviation.
- μ: Transformed rating.
- ϕ: Transformed rating deviation.
- k: Number of opponents.
- ϕ_j: Rating deviation of opponent j.
- $E(\mu, \mu_j, \phi_j)$: Expected score against opponent j.
- s_j: Actual score against opponent j (1 for win, 0.5 for draw, 0 for loss).
- σ: Current volatility before update.
- τ: System constant that constrains the change in volatility. It affects how quickly the volatility can change. The default is 0.5.
- σ': Updated volatility after solving the volatility update equation.
- ϕ^*: Intermediate rating deviation.
- μ': Updated rating.
- R': Updated rating.
- RD': Updated rating deviation.

Implementation Notes

- **Numerical Solution for Volatility:** The equation for updating volatility σ' does not have a closed-form solution and is typically solved using iterative numerical methods such as the Newton-Raphson method.
- **Initialization:** All players or teams start with an initial rating (e.g., 1500), an initial rating deviation (e.g., 350), and an initial volatility (e.g., 0.06).
- **Handling Multiple Opponents:** The summations in the formulas account for multiple opponents in a rating period.

- **TrueSkill**

$$c^2 = 2\beta^2 + \sigma_{\text{winner}}^2 + \sigma_{\text{loser}}^2$$

$$x = \frac{\mu_{\text{winner}} - \mu_{\text{loser}}}{c}$$

$$v(x, \epsilon) = \frac{\phi(x - \epsilon)}{\Phi(x - \epsilon)}$$

$$w(x, \epsilon) = v(x, \epsilon)\left(v(x, \epsilon) + x - \epsilon\right)$$

$$\mu_{\text{winner}} \leftarrow \mu_{\text{winner}} + \frac{\sigma^2_{\text{winner}}}{c} \cdot v\left(\frac{\mu_{\text{winner}} - \mu_{\text{loser}}}{c}, \epsilon\right)$$

$$\mu_{\text{loser}} \leftarrow \mu_{\text{loser}} - \frac{\sigma^2_{\text{loser}}}{c} \cdot v\left(\frac{\mu_{\text{winner}} - \mu_{\text{loser}}}{c}, \epsilon\right)$$

$$\sigma^2_{\text{winner}} \leftarrow \sigma^2_{\text{winner}} \cdot \left[1 - \frac{\sigma^2_{\text{winner}}}{c^2} \cdot w\left(\frac{\mu_{\text{winner}} - \mu_{\text{loser}}}{c}, \epsilon\right)\right]$$

$$\sigma^2_{\text{loser}} \leftarrow \sigma^2_{\text{loser}} \cdot \left[1 - \frac{\sigma^2_{\text{loser}}}{c^2} \cdot w\left(\frac{\mu_{\text{winner}} - \mu_{\text{loser}}}{c}, \epsilon\right)\right]$$

Where:

- $\mu_{\text{winner}}, \mu_{\text{loser}}$: Mean (skill) values of the winner and loser, respectively.
- $\sigma_{\text{winner}}, \sigma_{\text{loser}}$: Standard deviations (uncertainty) of the winner and loser, respectively.
- β: Dynamic factor, (default: 25.0/6).
- $\phi(x)$: The probability density function (PDF) of the standard normal distribution.
- $\Phi(x)$: The cumulative distribution function (CDF) of the standard normal distribution.
- ϵ: Draw margin, often calculated based on the draw probability (default: $\epsilon = 0.1$).

TrueSkill updates both the mean and uncertainty of a team's skill after each match. This dual update mechanism allows TrueSkill to not only track the estimated skill level but also the confidence in that estimate, providing a more dynamic and responsive ranking system. The initial value for $\mu = 25.0$ and $\sigma = \frac{25.0}{3}$ are used.

- **Average Goals Scored per Season**

$$\text{AvgGoals}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} G_{ij}$$

Where:

- N_i: Number of matches played by Team i in the season.
- G_{ij}: Number of goals scored by Team i in match j.

Appendix H

All Feature Descriptions

H.1 Team Ratings and Statistics

- **home_attack**: Attack rating of the home team.
- **away_attack**: Attack rating of the away team.
- **home_club_worth**: Total club worth of the home team.
- **away_club_worth**: Total club worth of the away team.
- **home_defence**: Defence rating of the home team.
- **away_defence**: Defence rating of the away team.
- **home_defence_domestic_prestige**: Domestic prestige rating for the home team's defence.
- **away_defence_domestic_prestige**: Domestic prestige rating for the away team's defence.
- **home_midfield**: Midfield rating of the home team.
- **away_midfield**: Midfield rating of the away team.
- **home_overall**: Overall rating of the home team.
- **away_overall**: Overall rating of the away team.
- **home_players**: Number of players in the home team.
- **away_players**: Number of players in the away team.
- **home_starting_xi_average_age**: Average age of the starting eleven for the home team.
- **away_starting_xi_average_age**: Average age of the starting eleven for the away team.
- **home_team_goals_season_to_date_before_match**: Total goals scored by the home team up to this point in the season.
- **away_team_goals_season_to_date_before_match**: Total goals scored by the away team up to this point in the season.
- **home_team_number_of_match_played**: Number of matches played by the home team in the season.
- **away_team_number_of_match_played**: Number of matches played by the away team in the season.
- **home_transfer_budget**: Transfer budget of the home team.
- **away_transfer_budget**: Transfer budget of the away team.
- **home_whole_team_average_age**: Average age of the entire home team.
- **away_whole_team_average_age**: Average age of the entire away team.

H.2 League Information

- **home_league_ESP-La Liga**: Indicator if the home team plays in La Liga.
- **away_league_ESP-La Liga**: Indicator if the away team plays in La Liga.
- **home_league_ENG-Premier League**: Indicator if the home team plays in the Premier League.
- **away_league_ENG-Premier League**: Indicator if the away team plays in the Premier League.
- **home_league_ITA-Serie A**: Indicator if the home team plays in Serie A.
- **away_league_ITA-Serie A**: Indicator if the away team plays in Serie A.
- **home_league_GER-Bundesliga**: Indicator if the home team plays in the Bundesliga.
- **away_league_GER-Bundesliga**: Indicator if the away team plays in the Bundesliga.
- **home_league_FRA-Ligue 1**: Indicator if the home team plays in Ligue 1.
- **away_league_FRA-Ligue 1**: Indicator if the away team plays in Ligue 1.
- **home_league_INT**: Indicator if the home team plays in an international league.
- **away_league_INT**: Indicator if the away team plays in an international league.

H.3 Prestige Ratings

- **home_international_prestige**: International prestige rating for the home team.
- **away_international_prestige**: International prestige rating for the away team.

H.4 Elo, Glicko-2 and Trueskill Ratings

- **elo_home_before**: Elo rating of the home team before the match.
- **elo_away_before**: Elo rating of the away team before the match.
- **glicko2_home_before**: Glicko-2 rating for the home team before the match.
- **glicko2_away_before**: Glicko-2 rating for the away team before the match.
- **glicko2_rd_home_before**: Glicko-2 rating deviation for the home team before the match.
- **glicko2_rd_away_before**: Glicko-2 rating deviation for the away team before the match.
- **glicko2_vol_home_before**: Glicko-2 volatility for the home team before the match.
- **glicko2_vol_away_before**: Glicko-2 volatility for the away team before the match.
- **trueskill_home_before**: TrueSkill rating for the home team before the match.
- **trueskill_away_before**: TrueSkill rating for the away team before the match.

H.5 Build-up and Passing Styles

- **home_build_up_passing_Mixed**: Mixed passing buildup style for the home team.
- **away_build_up_passing_Mixed**: Mixed passing buildup style for the away team.
- **home_build_up_passing_Short**: Short passing buildup style for the home team.
- **away_build_up_passing_Short**: Short passing buildup style for the away team.
- **home_build_up_positioning_Organised**: Organized buildup play for the home team.
- **away_build_up_positioning_Organised**: Organized buildup play for the away team.
- **home_build_up_speed_Fast**: Fast buildup play for the home team.
- **away_build_up_speed_Fast**: Fast buildup play for the away team.
- **home_build_up_speed_Slow**: Slow buildup play for the home team.
- **away_build_up_speed_Slow**: Slow buildup play for the away team.

H.6 Chance Creation and Shooting Styles

- **home_chance_creation_crossing_Lots**: High volume crossing chance creation for the home team.
- **away_chance_creation_crossing_Lots**: High volume crossing chance creation for the away team.
- **home_chance_creation_crossing_Normal**: Normal crossing chance creation for the home team.
- **away_chance_creation_crossing_Normal**: Normal crossing chance creation for the away team.
- **home_chance_creation_passing_Risky**: Risky passing for chance creation by the home team.
- **away_chance_creation_passing_Risky**: Risky passing for chance creation by the away team.
- **home_chance_creation_passing_Safe**: Safe passing for chance creation by the home team.
- **away_chance_creation_passing_Safe**: Safe passing for chance creation by the away team.
- **home_chance_creation_positioning_Organised**: Organized chance creation for the home team.
- **away_chance_creation_positioning_Organised**: Organized chance creation for the away team.
- **home_chance_creation_shooting_Lots**: High volume shooting chance creation for the home team.
- **away_chance_creation_shooting_Lots**: High volume shooting chance creation for the away team.
- **home_chance_creation_shooting_Normal**: Normal shooting chance creation for the home team.
- **away_chance_creation_shooting_Normal**: Normal shooting chance creation for the away team.

H.7 Defensive Strategies

- **home_defence_aggression_Double**: Double marking defensive aggression for the home team.
- **away_defence_aggression_Double**: Double marking defensive aggression for the away team.
- **home_defence_aggression_Press**: Pressing defensive aggression for the home team.
- **away_defence_aggression_Press**: Pressing defensive aggression for the away team.
- **home_defence_defender_line_Offside_trap**: Use of offside trap in the home team's defensive line.

- **away_defence_defender_line_Offside_trap**: Use of offside trap in the away team's defensive line.
- **home_defence_pressure_High**: High pressure defensive strategy for the home team.
- **away_defence_pressure_High**: High pressure defensive strategy for the away team.
- **home_defence_pressure_Medium**: Medium pressure defensive strategy for the home team.
- **away_defence_pressure_Medium**: Medium pressure defensive strategy for the away team.
- **home_defence_team_width_Normal**: Normal defensive width for the home team.
- **away_defence_team_width_Normal**: Normal defensive width for the away team.
- **home_defence_team_width_Wide**: Wide defensive width for the home team.
- **away_defence_team_width_Wide**: Wide defensive width for the away team.

Appendix I

Feature Importance

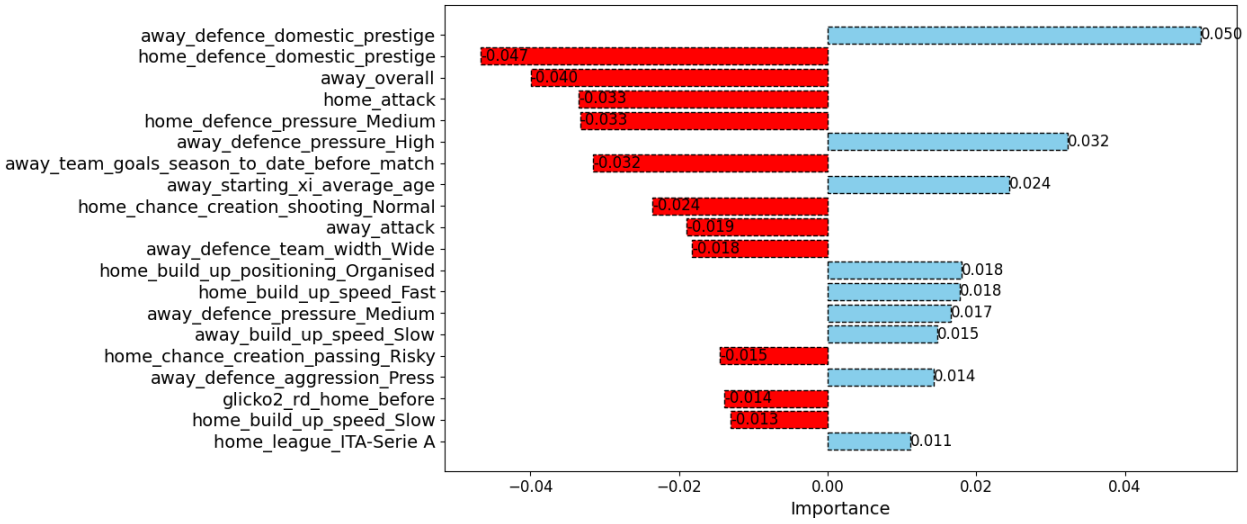

Figure I.1: Coefficients of the Logistic Regression Model for Draw class

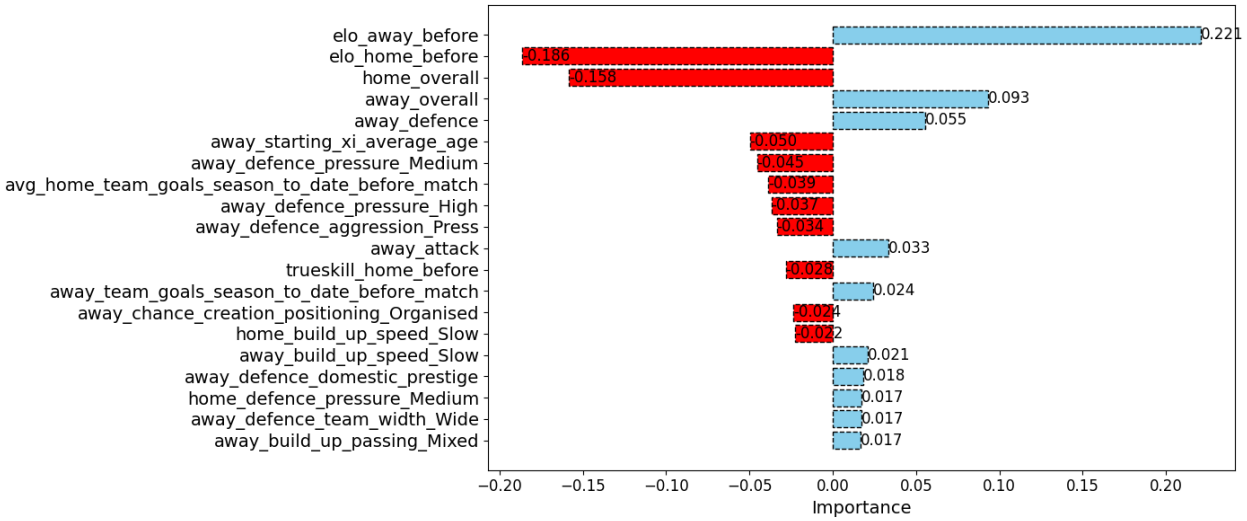

Figure I.2: Coefficients of the Logistic Regression Model for Away win class

Appendix J

Appendix: Bookmakers Used

During the online testing phase, odds were retrieved from a diverse pool of 21 bookmakers, ensuring a wide range of market conditions and betting options. The best odds for each match were selected from this pool to optimize returns. Table J.1 provides the list of bookmakers used in the study.

Table J.1: List of Bookmakers Used

Bookmaker	**Bookmaker**
1xBet	NordicBet
888Sport	Pinnacle
BetClic	Suprabets
BetAnySports	Tipico
Betfair Exchange EU	Unibet EU
BetOnline.ag	William Hill
Betsson	BetVictor
Coolbet	GTBets
Everygame	LiveScoreBet EU
Marathonbet	Matchbook
MyBookie.ag	

References

[1] Chris Anderson and David Sally. *The Numbers Game: Why Everything You Know About Soccer Is Wrong*. Penguin Books, 2013.

[2] Kenneth J. Arrow. *Essays in the Theory of Risk-Bearing*. North-Holland Publishing Company, 1971.

[3] Gianluca Baio and Marta Blangiardo. "Bayesian hierarchical model for the prediction of football results". In: *Journal of Applied Statistics* 37.2 (2010), pp. 253–264.

[4] Christoph Bergmeir, Rob J. Hyndman, and Ben Koo. "A note on the validity of cross-validation for evaluating autoregressive time series prediction". In: *Computational Statistics & Data Analysis* 120 (2018), pp. 70–83.

[5] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[6] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[7] Mike Cain, David Law, and David Peel. "The favourite-longshot bias and market efficiency in UK football betting". In: *Scottish Journal of Political Economy* 47.1 (2000), pp. 25–36.

[8] Girish Chandrashekar and Ferat Sahin. "A survey on feature selection methods". In: *Computers & Electrical Engineering* 40.1 (2014), pp. 16–28.

[9] Mark J. Dixon and Stuart G. Coles. "Modelling association football scores and inefficiencies in the football betting market". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 46.2 (1997), pp. 265–280.

[10] Arpad E. Elo. *The Rating of Chess Players, Past and Present*. Arco Publishing, 1978.

[11] Thomas S. Ferguson. *Mathematical Statistics: A Decision Theoretic Approach*. Academic Press, 1967.

[12] Mark E. Glickman. "Parameter estimation in large dynamic paired comparison experiments". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 48.3 (1999), pp. 377–394.

[13] Isabelle Guyon and André Elisseeff. "An introduction to variable and feature selection". In: *Journal of Machine Learning Research* 3 (2003), pp. 1157–1182.

[14] Ralf Herbrich, Tom Minka, and Thore Graepel. "TrueSkill™: A Bayesian skill rating system". In: *Advances in Neural Information Processing Systems*. 2007, pp. 569–576.

[15] Kelsey Hightower, Brendan Burns, and Joe Beda. *Kubernetes: Up and Running: Dive into the Future of Infrastructure*. O'Reilly Media, Inc., 2017.

[16] Lars Morten Hvattum and Halvard Arntzen. "Using ELO ratings for match result prediction in association football". In: *International Journal of Forecasting* 26.3 (2010), pp. 460–470.

[17] Daniel Kahneman and Amos Tversky. "Prospect theory: An analysis of decision under risk". In: *Econometrica* 47.2 (1979), pp. 263–292.

[18] John L. Kelly. "A new interpretation of information rate". In: *Bell System Technical Journal* 35.4 (1956), pp. 917–926.

[19] Jakub Lasek, Zoltán Szlávik, and Sandjai Bhulai. "The predictive power of ranking systems in association football". In: *International Journal of Applied Pattern Recognition* 1.1 (2013), pp. 27–46.

[20] Harry Markowitz. "Portfolio selection". In: *The Journal of Finance* 7.1 (1952), pp. 77–91.

[21] Dirk Merkel. "Docker: lightweight Linux containers for consistent development and deployment". In: *Linux Journal* 2014.239 (2014), p. 2.

[22] Sam Newman. *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media, Inc., 2015.

[23] John W. Pratt. "Risk aversion in the small and in the large". In: *Econometrica* 32.1/2 (1964), pp. 122–136.

[24] Edward O. Thorp. "Optimal gambling systems for favorable games". In: *Review of the International Statistical Institute* 37.3 (1969), pp. 273–293.

[25] Edward O. Thorp. "Portfolio choice and the Kelly criterion". In: *Business and Economics Statistics Section, Proceedings of the American Statistical Association*. 1975, pp. 215–224.