Department of Computer Engineering
Faculty of Computer Science of A Coruña
UNIVERSIDADE DA CORUÑA

MASTER'S THESIS
INTERUNIVERSITY MASTER'S DEGREE IN
HIGH PERFORMANCE COMPUTING

Predicting Professional Tennis Match Outcomes with Data Science and High Performance Computing

Student: Paul Serin
Supervisor(s): Guillermo López Taboada
Juan Angel Lorenzo del Castillo

A Coruña, June 19, 2025.

Dedication

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my advisor, Professor Guillermo López Taboada, for his constant support and guidance throughout this project. His insights, availability, and constructive feedback were instrumental in helping me structure this work and move forward with clarity.

I am also thankful to the friends and fellow students who showed genuine interest in my topic. Their fresh ideas, enthusiastic discussions, and suggestions, brought both creativity and perspective to this work. Your encouragement and curiosity helped shape the project in ways I could not have anticipated alone.

Finally, I'd like to thank everyone who supported me, directly or indirectly, during the course of this journey. Whether through a conversation, a piece of advice, or a simple word of encouragement, you've all contributed to making this experience both intellectually stimulating and personally meaningful.

Abstract

This master's thesis explores the application of data science and High Performance Computing (HPC) techniques to predict outcomes of professional tennis matches. Using historical data from the ATP Tour, we build a comprehensive feature set including ranking metrics, ELO ratings, surface-specific performance, and head-to-head statistics.

Several machine learning models (Decision Tree, Random Forest, and XGBoost) are trained and evaluated using a temporal train/test split. The XGBoost model achieves the best performance, with an accuracy of over 66% and a log loss below 0.61 on unseen data from 2024. This model is then used to simulate entire tournaments, including the Australian Open and Roland-Garros 2025.

In round-by-round simulations, the model successfully predicted over 70% of matches, including both finalists and the eventual champion at the Australian Open. Monte Carlo simulations over 5,000 tournament iterations further confirmed the model's consistency, with dominant players emerging in line with real-world outcomes.

To scale both simulations and hyperparameter tuning, we leverage Dask and SLURM for distributed execution on the Finisterrae III supercomputer. Hyperparameter optimization using Dask's RandomizedSearchCV was accelerated across 128 CPU threads and 4 NVIDIA A100 GPUs, reducing total runtime from hours to minutes. Similarly, parallel Monte Carlo tournament simulations were efficiently scaled across multiple GPUs, achieving high speedup and parallel efficiency. These results highlight not only the predictive potential of machine learning in professional tennis, but also the crucial role of HPC in enabling fast, scalable, and reproducible analysis on large datasets.

Resumen

Este trabajo de fin de máster explora la aplicación de técnicas de ciencia de datos y computación de alto rendimiento (HPC) para predecir los resultados de partidos de tenis profesional. Utilizando datos históricos del circuito ATP, se construye un conjunto de características completo que incluye métricas de ranking, puntuaciones ELO, rendimiento por superficie y estadísticas de enfrentamientos directos (head-to-head).

Se entrenan y evalúan varios modelos de aprendizaje automático (Árbol de Decisión, Bosque Aleatorio y XGBoost) mediante una partición temporal entre los datos de entrenamiento y prueba. El modelo XGBoost alcanza el mejor rendimiento, con una precisión superior al 66% y una pérdida logística inferior a 0.61 sobre datos no vistos del año 2024. Este modelo se utiliza posteriormente para simular torneos completos, incluyendo el Abierto de Australia y Roland-Garros 2025.

En las simulaciones ronda por ronda, el modelo predice correctamente más del 70% de los partidos, identificando incluso a los finalistas y al campeón del Abierto de Australia. Las simulaciones de Monte Carlo, repetidas en 5 000 torneos, confirman la consistencia del modelo, con jugadores dominantes que reflejan con precisión el panorama competitivo real.

Para escalar tanto las simulaciones como la búsqueda de hiperparámetros, se utilizan Dask y SLURM para ejecutar tareas distribuidas en el supercomputador Finisterrae III. La optimización de hiperparámetros mediante RandomizedSearchCV de Dask se aceleró usando 128 hilos de CPU y 4 GPUs NVIDIA A100, reduciendo el tiempo total de ejecución de varias horas a unos pocos minutos. De forma similar, las simulaciones paralelas de torneos se distribuyeron eficazmente entre varias GPUs, logrando una alta aceleración y eficiencia. Estos resultados subrayan no solo el potencial predictivo del aprendizaje automático en el tenis profesional, sino también el papel clave de la HPC para permitir análisis rápidos, escalables y reproducibles sobre grandes volúmenes de datos.

Keywords:

- Data Science
- High Performance Computing
- Machine Learning
- Tennis Match Prediction
- ELO Rating System
- Feature Engineering
- Distributed Training
- Monte Carlo Simulation
- Decision Tree
- Random Forest
- XGBoost

Palabras clave:

- Ciencia de los Datos
- Computación de Altas Prestaciones
- Aprendizaje Automático
- Predicción de Partidos de Tenis
- Sistema de Puntuación ELO
- Ingeniería de Características
- Entrenamiento Distribuido
- Simulación Monte Carlo
- Árbol de Decisión
- Bosque Aleatorio
- XGBoost

Contents

1 Introduction **1**

2 Problem Statement and Objectives **3**

2.1 Motivation for match outcome prediction in tennis . . . 3

2.2 Problem formulation (binary classification) . . . 3

2.3 Objectives of the project . . . 4

3 Data Collection and Preprocessing **5**

3.1 Source datasets . . . 5

3.2 Match history formatting and merging . . . 6

3.3 Player ID harmonization and cleaning . . . 6

3.4 Handling missing data and duplicates . . . 7

4 Feature Engineering **9**

4.1 Rolling statistics . . . 9

4.2 Advanced features . . . 10

4.2.1 Head-to-head metrics . . . 10

4.2.2 Surface-adjusted performance . . . 10

4.2.3 ELO-based features . . . 10

4.3 Feature symmetry and target construction . . . 11

5 Exploratory Data Analysis and Visualization **13**

5.1 Match Overview . . . 14

5.2 Player Overview . . . 15

5.3 Winrate Analysis . . . 17

5.4 ELO Evolution . . . 19

5.5 Serve Performance . . . 20

5.6 Correlation Analysis . . . 22

5.7 Conclusion . . . 23

6 Model Training and Evaluation **25**

6.1 Train/Test Split (Temporal Strategy) . . . 26

6.2 Models Selected . . . 26

6.3 Evaluation Metrics . . . 26

6.4 Decision Tree . . . 27

6.4.1 Training and Evaluation . . . 27
6.4.2 Confusion Matrix Analysis . . . 28
6.4.3 Feature Importance . . . 28
6.4.4 Model Interpretability . . . 29
6.4.5 Conclusion . . . 29
6.5 Random Forest . . . 29
6.5.1 Training and Evaluation . . . 30
6.5.2 Confusion Matrix Analysis . . . 31
6.5.3 Feature Importance . . . 31
6.5.4 Conclusion . . . 32
6.6 XGBoost . . . 32
6.6.1 Training and Evaluation . . . 33
6.6.2 Confusion Matrix Analysis . . . 33
6.6.3 Feature Importance . . . 34
6.6.4 Conclusion . . . 34
6.7 XGBoost Distributed Hyperparameter Tuning with Dask . . . 35
6.7.1 Motivation and Objective . . . 35
6.7.2 Infrastructure Setup with SLURM and Dask . . . 35
6.7.3 Workflow Overview . . . 36
6.7.4 Performance Insights . . . 36
6.7.5 Results Summary . . . 39
6.7.6 Conclusion and Transition . . . 39
6.8 Conclusion . . . 39

7 Match and Tournament Simulation 41
7.1 Predicting Individual Matches . . . 42
7.1.1 Case Study 1: Carlos Alcaraz vs Jannik Sinner . . . 42
7.1.2 Surface Sensitivity and Model Credibility . . . 43
7.2 Simulating Real Tournaments . . . 43
7.2.1 Methodology . . . 43
7.2.2 Australian Open 2025 . . . 43
7.2.3 Roland Garros 2025 . . . 44
7.2.4 Discussion . . . 45
7.3 Monte Carlo Simulations . . . 45
7.3.1 Rationale and Methodology . . . 45
7.3.2 Distributed Simulation with SLURM . . . 47
7.3.3 Scalability Analysis and Performance Insights . . . 47
7.3.4 Insights from 5000 Simulated Tournaments . . . 49
7.3.5 Overall Evaluation of the Monte Carlo Approach . . . 51
7.4 Conclusion . . . 51

8 Limitations and Perspectives 53
8.1 Player Injuries and Physical Condition . . . 53
8.2 Gender-Specific Analysis and Model Generalization . . . 53
8.3 Beyond the Numbers: Human Factors . . . 53

8.4 Environmental Conditions 54
8.5 The Limits of the Elo Feature 54
8.6 Responsible Use and Betting Caution 54
8.7 Other Perspectives and Extensions 54

9 Conclusion **57**

A Glossary of Acronyms **61**

B Glossary of Terms **63**

Bibliography **67**

List of Figures

5.1 Number of matches per surface . . . 14
5.2 Average match duration by surface . . . 15
5.3 Distribution of player birth years . . . 16
5.4 Player height distribution . . . 16
5.5 Player distribution by country and dominant hand . . . 17
5.6 Top 10 all-time winrates (min 200 matches) . . . 18
5.7 Top 5 winrates by surface (min 100 matches per surface) . . . 18
5.8 Global ELO evolution of selected players . . . 19
5.9 Surface-specific ELO evolution . . . 20
5.10 Ace rate by surface . . . 20
5.11 Top 10 players by ace percentage . . . 21
5.12 Correlation between features and match outcome (TARGET) . . . 22
5.13 Feature correlation matrix . . . 23

6.1 Temporal split of the dataset used for training and testing the model. . . . 26
6.2 Confusion matrix on the 2024 test set. . . . 28
6.3 Top 20 feature importances in the Decision Tree model. . . . 29
6.4 Top of the Decision Tree (depth = 3). . . . 30
6.5 Confusion matrix for Random Forest on the 2024 test set. . . . 31
6.6 Top 20 feature importances in the Random Forest model. . . . 32
6.7 Confusion matrix for XGBoost on the 2024 test set. . . . 34
6.8 Top 20 feature importances in the XGBoost model. . . . 35
6.9 Dask Task Stream view. Each horizontal bar represents a scheduled task on one worker. Tasks are densely packed with minimal idle time, confirming efficient parallelism. . . . 37
6.10 Worker-level profiling of task execution and CPU/GPU usage. Workers exhibit balanced loads and synchronized progress across all hyperparameter trials. . . . 38
6.11 Inter-worker bandwidth usage. Low values confirm that Dask spent very little time in communication, consistent with the expected behavior of parallel cross-validation. . . . 38

7.1 Simulated matchup: Alcaraz vs Sinner. . . . 42
7.2 Simulated bracket from the quarter-finals onwards – Australian Open 2025. . . . 44
7.3 Simulated bracket from the quarter-finals onwards – Roland Garros 2025. . . . 45
7.4 Illustration of a Monte Carlo match decision . . . 46
7.5 Simulation time per GPU configuration (in seconds and hours). . . . 48
7.6 Speedup as a function of GPU count. . . . 48

7.7 Parallel efficiency as a function of GPU count. 49

List of Tables

6.1 Hyperparameter configuration used for the Random Forest model. 30
6.2 Hyperparameter configuration used for the XGBoost model. 33

7.1 Predicted outcomes for Alcaraz vs Sinner on different surfaces. 42
7.2 Australian Open 2025 – Monte Carlo results (5000 simulations) 50
7.3 Roland-Garros 2025 – Monte Carlo results (5000 simulations) 50

Chapter 1

Introduction

Tennis is a dynamic and globally followed sport where outcomes are often influenced by a complex interplay of factors: player form, surface preferences, physical condition, and psychological resilience. Accurately predicting the result of a professional tennis match requires not only historical data, but also a thoughtful integration of context-specific insights. As interest in sports analytics continues to grow, leveraging data science and machine learning (ML) techniques to make accurate forecasts has become a promising area of exploration.

In this MSc thesis, we focus on predicting the outcomes of men's singles matches on the ATP (Association of Tennis Professionals) Tour using a wide range of data-driven features. From player rankings and ELO scores to surface-specific statistics and head-to-head history, our approach blends classical indicators with engineered variables to capture performance dynamics. But this work goes beyond static predictions: we aim to simulate entire Grand Slam tournaments by chaining predictions round by round, enabling the estimation of win probabilities across complete tournament trees.

To handle the scale and complexity of such simulations, especially when using the computationally intensive Monte Carlo techniques, we turn to High Performance Computing (HPC). We integrate tools like Dask and SLURM to distribute the computational load across HPC clusters, allowing for large-scale training, hyperparameter tuning, and tournament simulations.

This project not only investigates the predictive power of several ML models (Decision Tree, Random Forest, and XGBoost), but also demonstrates how parallel computing can enhance efficiency and scalability in sports analytics. By combining statistical rigor with computational power, we aim to deliver both methodological insights and practical results for tennis match prediction.

Chapter 2

Problem Statement and Objectives

Predicting the outcome of a professional tennis match is a challenging yet enticing task. It requires the integration of multiple data sources, domain-specific knowledge, and scalable machine learning techniques. Unlike casual estimations based on player rankings or recent headlines, a systematic approach must dig deeper, uncovering the subtle statistical patterns that determine success on court.

This chapter presents the motivation behind this research, formalizes the prediction task as a binary classification problem, and outlines the concrete objectives pursued throughout the project. From raw match data to large-scale simulations, the goal is not only to predict who will win, but to understand why, and to do so at scale, leveraging the power of HPC.

2.1 Motivation for match outcome prediction in tennis

Professional tennis offers a unique landscape for predictive modeling. Unlike team sports, it features clearly defined one-on-one confrontations, standardized scoring systems, and rich individual player histories. These characteristics make it particularly suitable for outcome prediction using machine learning.

In recent years, sports analytics has grown in popularity across disciplines, from football to basketball and beyond. In tennis, betting markets, coaching strategies, and fan engagement all benefit from accurate match predictions. However, most publicly available predictions are based on simple heuristics such as player ranking or head-to-head records, failing to capture the nuanced performance indicators that actually influence match results.

This motivates the development of a more robust, data-driven approach to tennis match outcome prediction—one that integrates historical context, engineered features, and scalable algorithms to forecast not only individual matches but also entire tournaments.

2.2 Problem formulation (binary classification)

The core task addressed in this thesis is framed as a supervised binary classification problem. Given a set of pre-match features derived from both players involved in a match, the goal is to predict the binary outcome:

whether Player A (the ”target player”) will win or lose the match.

Each data instance corresponds to a single historical match, and the features are engineered to be symmetric, i.e., always structured around the same ”target player” (typically the one on the left side of the dataset). This structure allows models to generalize better and avoid bias introduced by dataset formatting.

The problem is inherently imbalanced in certain contexts (e.g., top players winning disproportionately), and performance is not only evaluated on raw accuracy, but also using metrics like F1 score and log loss to assess probabilistic quality and class balance.

2.3 Objectives of the project

This project aims to build a complete pipeline for professional tennis match prediction, from raw data acquisition to tournament simulations. The main objectives are organized into three successive stages:

- **Building a robust dataset:** Starting from historical data retrieved from Jeff Sackmann's ATP repository, we aim to consolidate and preprocess a clean match history. This includes player ID harmonization, handling missing values, and engineering features such as rolling statistics, ELO scores, surface performance indicators, and head-to-head metrics.
- **Training and evaluating predictive models:** We train several machine learning models (Decision Tree, Random Forest, and XGBoost) on the structured dataset to predict the outcome of ATP matches. A temporal train/test split is used to prevent data leakage. Performance is assessed using multiple evaluation metrics, and hyperparameter tuning is conducted to optimize results.
- **Simulating matches and full tournaments:** Using the best performing model, we simulate individual matchups and entire Grand Slam tournaments (Australian Open and Roland-Garros 2025). Win probabilities are estimated and Monte Carlo simulations are run to model full tournament progressions from randomized draws.

To handle the computational demands of large scale model training and simulation, the pipeline incorporates High Performance Computing (HPC) tools such as Dask and SLURM. These tools allow for distributed training and parallel execution of thousands of tournament simulations, enabling both speed and scalability.

Chapter 3

Data Collection and Preprocessing

The foundation of any data-driven project lies in the quality and structure of the data used. For this thesis, we leverage the historical match data provided by Jeff Sackmann's open-source ATP repository [1]. This dataset has been curated over many years and offers a detailed record of professional tennis matches, making it an invaluable resource for modeling and prediction tasks.

3.1 Source datasets

The core data originates from the ATP match dataset maintained by Jeff Sackmann [1]. It includes season-by-season CSV files from 1968 onward, each containing information about ATP singles matches.

Each CSV file (e.g., *atp_matches_2024.csv*) contains over 50 columns covering various dimensions: tournament context, player identity, match outcome, and performance statistics. The most relevant fields can be grouped as follows:

Tournament Information

- tourney_id
- tourney_name
- surface
- draw_size
- tourney_level
- tourney_date
- round
- best_of

Match Outcome and Scores

- score
- match_num
- minutes

Player Metadata

- winner_name / loser_name
- winner_hand / loser_hand
- winner_age / loser_age
- winner_ioc / loser_ioc
- winner_rank / loser_rank
- winner_rank_points / loser_rank_points

In-Match Statistics

- Aces / Double Faults
- Serve Points and Games
- 1st / 2nd Serve
- Break Points Faced / Saved

This rich and granular feature set provides both contextual and performance-based variables essential for accurate prediction. Without this curated and standardized dataset, building a high-quality dataset from scratch would have required enormous manual effort.

3.2 Match history formatting and merging

To build a unified dataset suitable for supervised learning, all yearly match files were concatenated into a single dataframe, spanning more than two decades of ATP-level competition. Standardization of column types (e.g., date parsing, numeric conversion) was performed, and additional derived fields such as match year and surface type were added.

Special attention was paid to ensuring consistent match ordering and filtering out incomplete or exhibition events that do not follow the official ATP scoring structure.

3.3 Player ID harmonization and cleaning

Players may appear under slight name variants (e.g., “Rafael Nadal” vs. “R. Nadal”), and the same ID might not always be used across different matches. To ensure data consistency:

- A unified player ID mapping was constructed, based on the *player_id* column when available, otherwise using a cleaned name matching algorithm.

- Special cases such as juniors, qualifiers without full metadata, or corrupted entries were either corrected or discarded based on validation rules.

This harmonization was essential for computing historical player statistics, head-to-head features, and rolling performance metrics later in the pipeline.

3.4 Handling missing data and duplicates

Despite the dataset's quality, some matches lack complete information. Typical missing data issues included:

- Missing match statistics (especially in older seasons).
- Unknown player attributes (height, handedness, etc.).
- Absent rankings or scores in early rounds of Challenger-level tournaments.

The following strategies were applied:

- Matches with missing target variables (e.g., no winner) were dropped.
- Player metadata was imputed when possible from external snapshots.
- Duplicate entries were eliminated using `tourney_id`, `match_num`, and a hash of key fields to identify repeated rows.

In summary, this preprocessing stage was essential to ensure the reliability of both the training data and the tournament simulation logic developed later in the project. The curated dataset ultimately contains over 100,000 ATP matches with cleaned and enriched features, ready for advanced machine learning applications.

Chapter 4

Feature Engineering

Once the historical match dataset is cleaned and structured, the next step is to engineer meaningful features that enhance the predictive power of machine learning models. In tennis, the richness of player interactions and the sequential nature of the sport offer opportunities to design features that go beyond raw statistics.

The goal of this stage is to translate domain knowledge into data: we attempt to capture momentum, surface preferences, player dominance in past encounters, and empirical strength via rating systems. A comprehensive overview of all variables generated in the pipeline is available at the project's public repository: https://github.com/PaulSerin/TFM/blob/main/Dictionnary.md.

4.1 Rolling statistics

In racket sports, a player's recent performance, also known as *momentum*,is often more informative than their long-term average. Using the chronological structure of the dataset, we compute rolling win rates and service statistics over various window sizes.

The main idea is to quantify how well a player has been performing in the lead-up to a given match. For example:

- PLAYER1_LAST_25_WINRATE: fraction of wins over the last 25 matches
- PLAYER2_LAST_10_WINRATE: recent short-term form

In parallel, we compute rolling service efficiency metrics, such as:

- PLAYER1_P_1STIN_LAST_50: percentage of first serves in
- PLAYER2_P_BPSAVED_LAST_100: ability to save break points

These statistics are crucial to approximate a player's current shape and consistency under pressure.

4.2 Advanced features

4.2.1 Head-to-head metrics

The psychological and tactical dynamics between two players can impact match outcomes. We compute:

- H2H_TOTAL_DIFF: net number of wins for PLAYER1 over PLAYER2 in all previous encounters
- H2H_SURFACE_DIFF: same difference but restricted to the current surface (e.g., clay)

These variables help capture rivalries or matchups historically favorable to one player.

4.2.2 Surface-adjusted performance

Players often perform differently depending on the surface. To reflect this, we compute:

- PLAYER1_SURFACE_MATCHES, PLAYER2_SURFACE_MATCHES
- PLAYER1_LAST_50_WINRATE_CLAY, PLAYER2_LAST_25_WINRATE_GRASS

This accounts for surface specialization (e.g., Rafael Nadal's dominance on clay).

4.2.3 ELO-based features

The Elo rating system, originally designed for chess, has become a cornerstone in various competitive environments, including esports, football, and tennis. It provides a dynamic and continuous way to estimate player skill, evolving match after match based not only on the outcome but also on the opponent's strength.

This system is particularly attractive in tennis analytics, where official ATP rankings can be delayed or inflated by tournament structure and protected points. Elo, by contrast, updates in real time and can be tailored to different contexts—such as global skill or surface-specific expertise.

The Elo score is updated after each match using the following formula:

$$\text{Elo}_{\text{new}} = \text{Elo}_{\text{old}} + K \cdot (S - E)$$

Where:

- Elo_{old} is the player's rating before the match
- K is a sensitivity parameter that controls how fast the Elo adapts

- $S \in \{0, 1\}$ is the actual match result (1 for win, 0 for loss)
- E is the expected result, computed as:

$$E = \frac{1}{1 + 10^{(\text{Elo}_{\text{opponent}} - \text{Elo}_{\text{player}})/400}}$$

The parameter K plays a critical role: higher values make the system more reactive (useful for modeling rookies or early-season shifts), whereas lower values yield smoother trajectories and historical stability.

In this project, we compute both:

- PLAYER1_ELO_BEFORE and PLAYER2_ELO_BEFORE – the players' Elo ratings before the match
- PLAYER1_ELO_SURFACE_BEFORE and PLAYER2_ELO_SURFACE_BEFORE – surface-specific ratings (e.g., clay Elo)
- ELO_DIFF and ELO_SURFACE_DIFF – useful for direct match comparison

These features have shown strong predictive power in our models, offering a robust summary of historical performance and contextual strength.

A visual analysis of how Elo evolves over time, both globally and per surface, is provided in Section 5.4, illustrating its value for distinguishing eras and surface dominance. For a detailed discussion on tennis-specific Elo modeling, we refer to the article [2].

4.3 Feature symmetry and target construction

In its original form, the dataset includes separate columns for the `winner` and the `loser` of each match (e.g., `winner_rank`, `loser_age`, `winner_id`, `loser_id`, etc.). If used directly, this structure introduces a strong data leakage: the identity of the winner is hardcoded in the input features. As a result, the model would trivially learn to predict the target with near-perfect accuracy, but only because the label is embedded in the column naming, not because of meaningful patterns.

To address this, we reformulate each match in a **symmetrical binary format** by applying the following strategy:

- For every match, we create two rows: one where the original winner is assigned as PLAYER1, and one where the winner is assigned as PLAYER2.
- All features are re-indexed accordingly to represent PLAYER1 and PLAYER2 instead of `winner` and `loser`.
- The target column TARGET is set to 1 if PLAYER1 wins the match, and 0 otherwise.

This process ensures that:

- The model has no positional bias: it cannot assume PLAYER1 is stronger just because of the order.
- The dataset becomes balanced by construction: each match appears once with TARGET = 1 and once with TARGET = 0.
- Any asymmetry in feature values must stem from the underlying player characteristics—not the structure of the data.

By enforcing this symmetry, we preserve the integrity of the binary classification task and allow the model to learn purely from meaningful patterns and relative player features.

Chapter 5

Exploratory Data Analysis and Visualization

Before diving into model training and simulations, it is essential to gain a thorough understanding of the underlying data. Exploratory Data Analysis (EDA) serves as a fundamental step in any data-driven project, as it allows us to uncover patterns, detect anomalies, verify assumptions, and generate hypotheses.

In this chapter, we employ a variety of visualizations to explore the historical tennis dataset we constructed. These plots help us better understand the distribution of matches across surfaces, player demographics, win rates, ELO progression, and serve-related metrics. We also perform correlation analyses to identify which pre-match features are most strongly linked to match outcomes.

By visualizing these dimensions, we not only gain insights into the structure and trends of professional tennis but also ensure that the modeling process is grounded in data that is well understood and statistically sound.

5.1 Match Overview

Understanding how matches are distributed across surfaces is a fundamental first step in tennis analytics, as surface type significantly affects playing styles, rally lengths, and player performance.

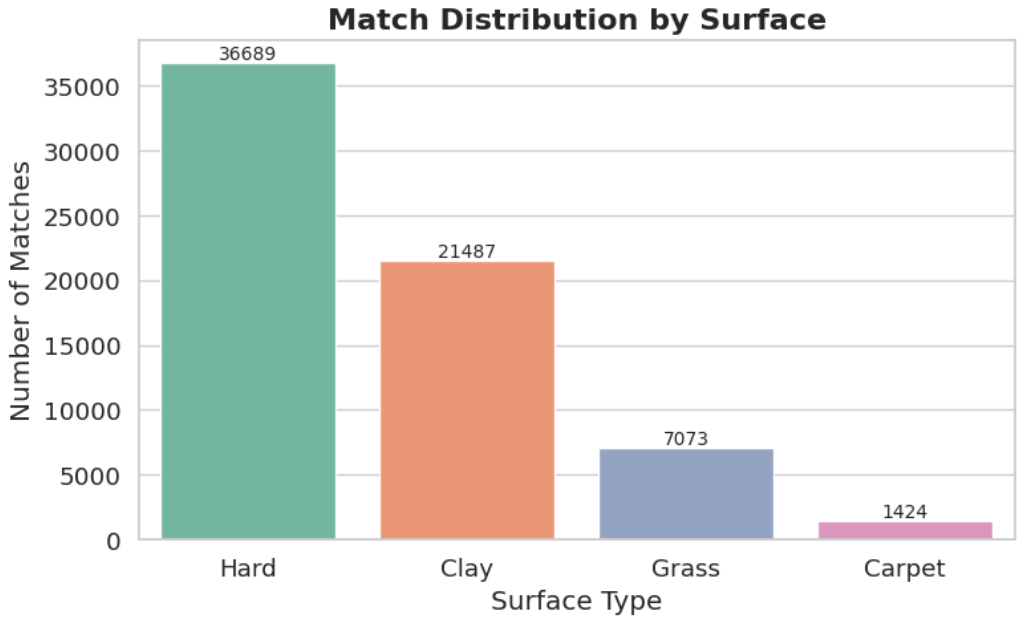

Figure 5.1: Number of matches per surface

Figure 5.1 shows the total number of ATP matches recorded per surface. We observe a clear dominance of `Hard` courts, which account for over 36,000 matches, more than half of the total. This is followed by `Clay` with around 21,500 matches, while `Grass` and `Carpet` are far less represented. Carpet courts, in particular, are nearly obsolete in the modern ATP Tour and only appear in historical datasets or indoor events from previous decades.

This unbalanced distribution has important modeling implications: models must learn to generalize across surface types despite the smaller sample sizes for grass and carpet.

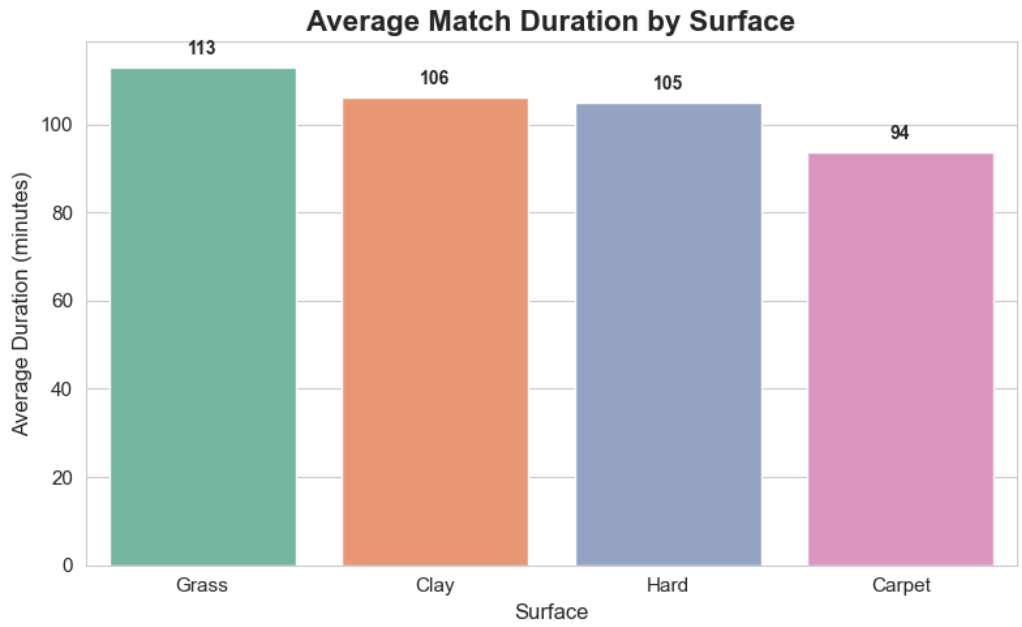

Figure 5.2: Average match duration by surface

Figure 5.2 provides complementary insight by analyzing the average match duration per surface. Interestingly, `Grass`, despite being the least frequent, hosts the longest matches on average (113 minutes), followed closely by `Clay` (106 minutes) and `Hard` (105 minutes). The shortest matches occur on `Carpet` (94 minutes), reflecting its historically fast conditions that often favored short rallies and dominant serving.

Together, these figures confirm that both match frequency and match duration vary meaningfully with surface type. These factors should be taken into account when designing surface-aware features or stratifying datasets during model training.

5.2 Player Overview

A deeper understanding of player demographics provides context on the population we are modeling. This section explores the age distribution, physical characteristics, and geographical origins of ATP players, helping us better frame the variability in performance and playing style.

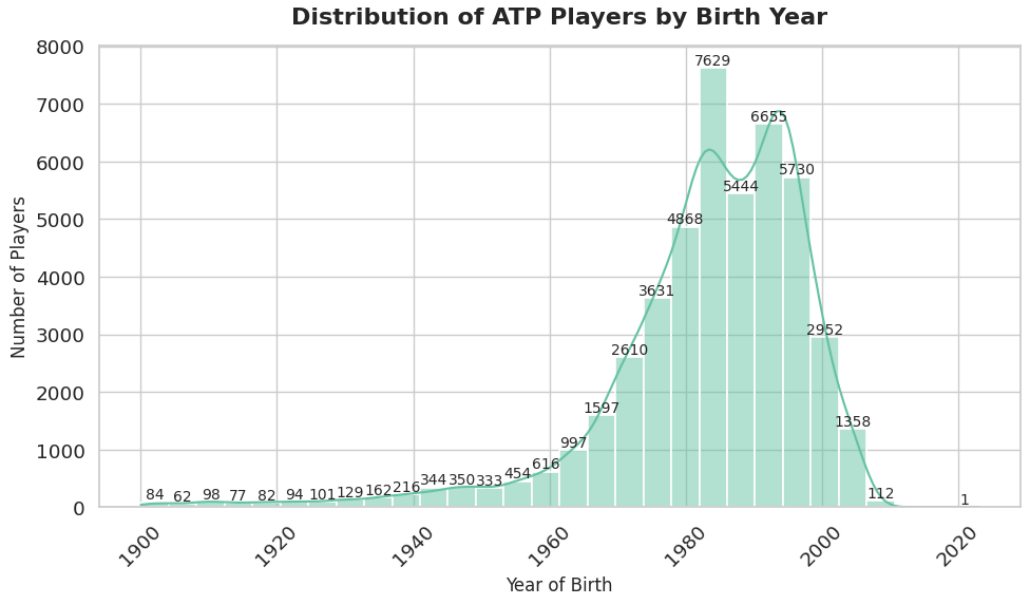

Figure 5.3: Distribution of player birth years

Figure 5.3 displays the distribution of ATP players by birth year. The dataset covers players born as early as 1900, but the bulk of observations corresponds to generations born between 1975 and 2000, with a notable peak around 1985. This reflects both the growth of the sport in the 1980s and the availability of digital records. The recent drop in players born after 2005 is due to the natural lag in professional entry age, not a decline in participation.

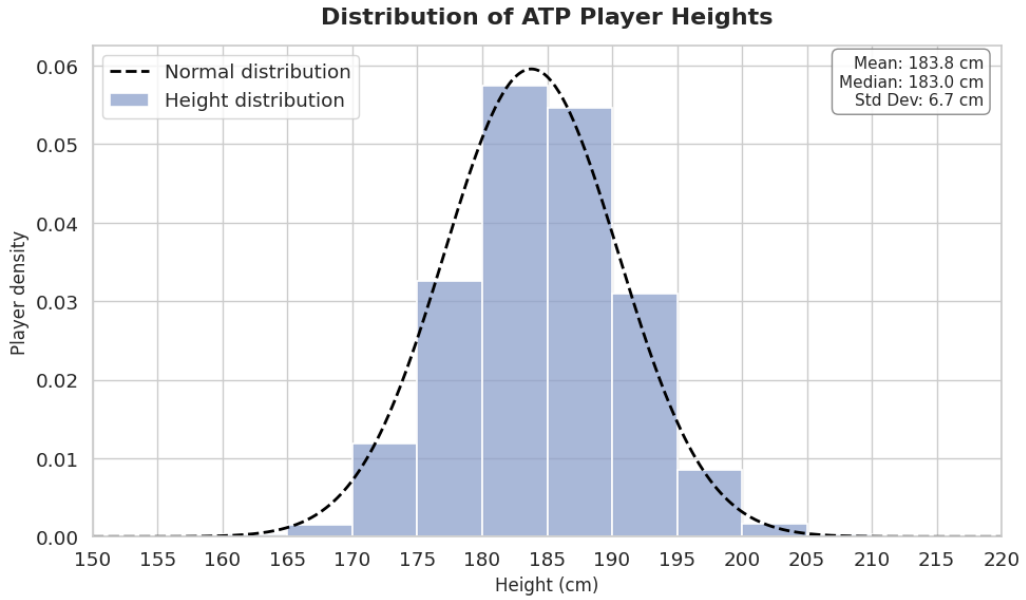

Figure 5.4: Player height distribution

Figure 5.4 shows the height distribution of players. The data is well approximated by a normal distribution centered around a mean and median of 183 cm, with a standard deviation of 6.7 cm. This suggests that top-level tennis favors players with above-average height, likely due to the physical demands of serve and reach. However, the relatively narrow spread also reflects a degree of homogeneity in physical attributes at the elite level.

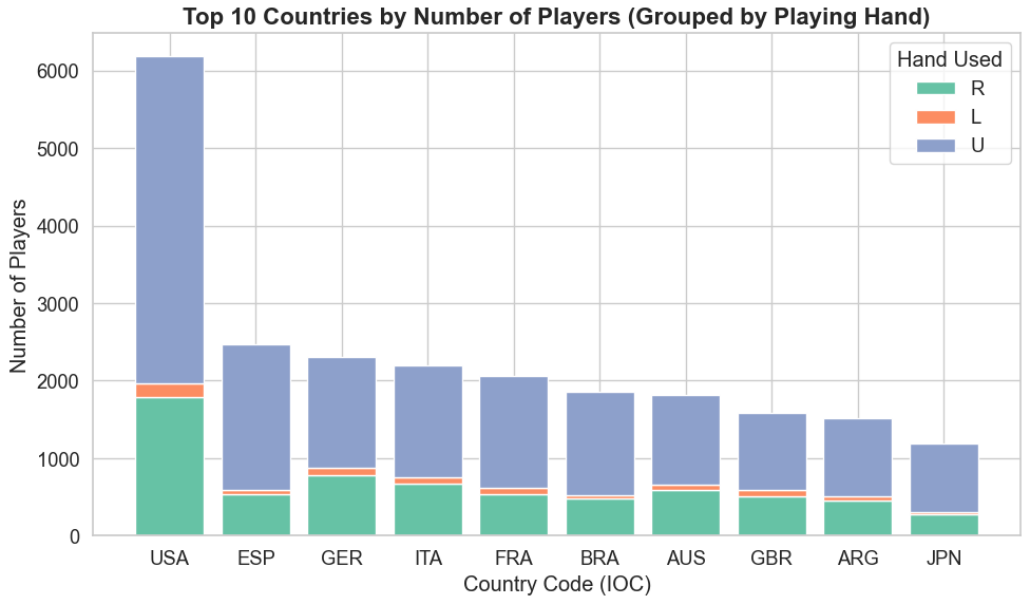

Figure 5.5: Player distribution by country and dominant hand

Figure 5.5 ranks the top 10 countries by number of ATP players, while also segmenting by dominant hand. The United States leads by a wide margin, followed by Spain, Germany, and Italy. The overwhelming majority of players are right-handed, with left-handed players forming a small but non-negligible minority. Interestingly, the data also includes a significant portion of players with unknown handedness, likely due to incomplete historical records.

Together, these visualizations highlight important demographic characteristics that may influence playing style, longevity, and surface preferences, factors that can all contribute to match outcomes and model generalization.

5.3 Winrate Analysis

Winrate is a fundamental performance metric in tennis analytics. It measures a player's ability to consistently win matches, and is particularly useful when contextualized by surface or compared across eras. To avoid distortion due to small sample sizes, we only consider players with a minimum number of matches (typically 100 or 200) in the rankings presented here.

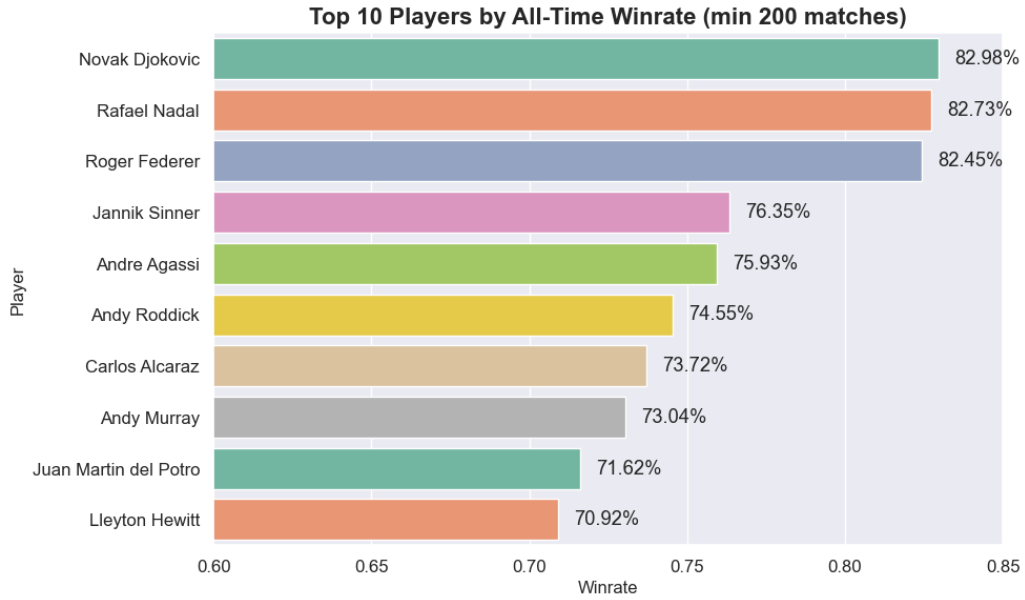

Figure 5.6: Top 10 all-time winrates (min 200 matches)

Figure 5.6 presents the top 10 ATP players in terms of career win percentage. Novak Djokovic leads the list with a winrate of 82.98%, followed closely by Rafael Nadal (82.73%) and Roger Federer (82.45%). These three players, often referred to as the "Big Three," have dominated men's tennis for nearly two decades. Their high winrates are not only a reflection of their talent but also of their longevity and consistency across surfaces and eras.

Several younger or more recent players also appear, including Jannik Sinner and Carlos Alcaraz. While their winrates are already impressive, their long-term positions will depend on how they sustain this performance across more matches and surfaces.

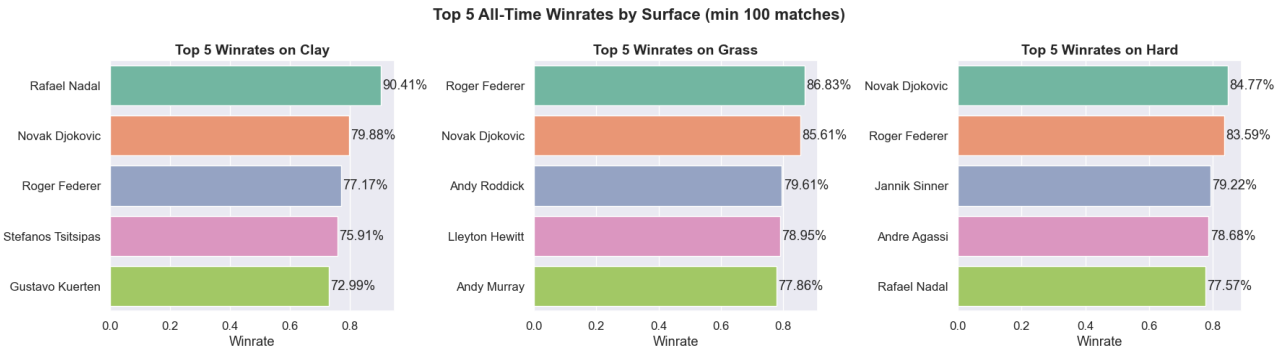

Figure 5.7: Top 5 winrates by surface (min 100 matches per surface)

Figure 5.7 dives deeper by splitting winrates by surface type. On `Clay`, Rafael Nadal is in a class of his own, with a remarkable 90.41% winrate, far ahead of Djokovic (79.88%) and Federer (77.17%). Nadal's dominance on clay is historically unparalleled, especially at Roland-Garros.

On `Grass`, Roger Federer (86.83%) and Novak Djokovic (85.61%) again top the list, reflecting their multiple Wimbledon titles. On `Hard`, Djokovic leads (84.77%) ahead of Federer and Sinner, underscoring his adaptability and strength on the most common surface on the tour.

These analyses validate that while overall winrate is informative, surface-specific performance provides a richer understanding of player strengths and specialization.

5.4 ELO Evolution

Building on the surface-independent and surface-specific Elo features described in Section 4.2.3, this section offers a visual exploration of how players' Elo ratings evolve over time. The Elo system provides a continuous, match-by-match estimate of player strength, which reflects not only outcomes but also the quality of opposition.

Unlike static metrics such as ATP rankings, Elo captures short-term momentum and long-term dominance, making it particularly well-suited for evaluating trajectories and simulating match probabilities.

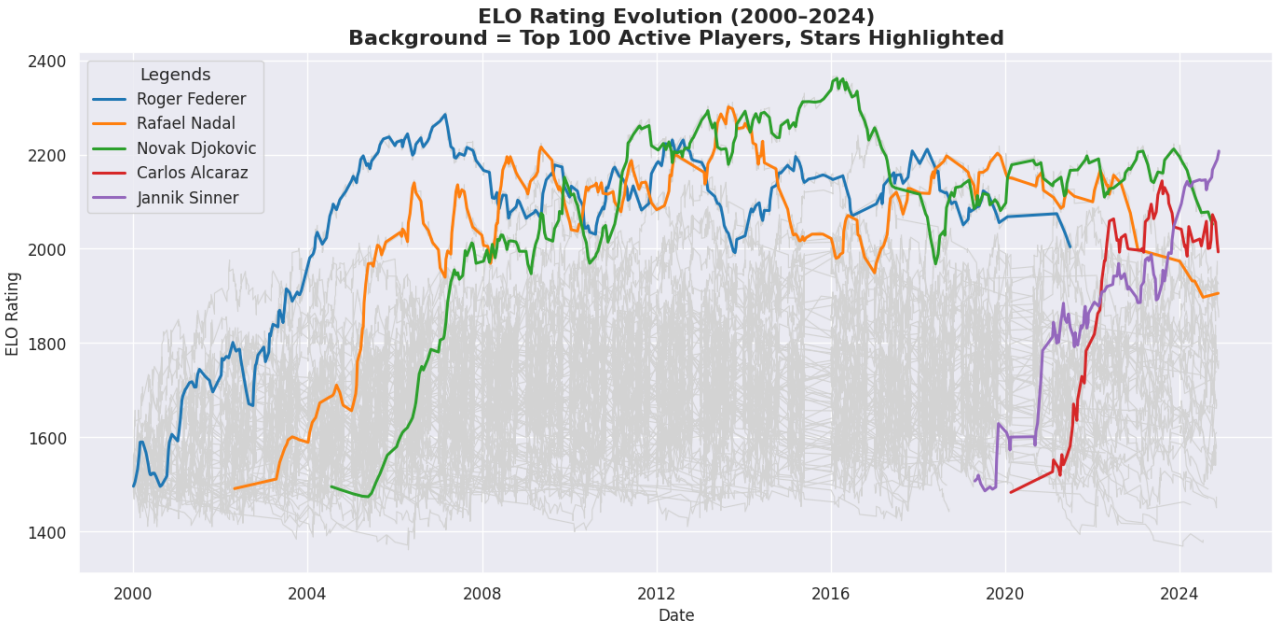

Figure 5.8: Global ELO evolution of selected players

Figure 5.8 displays the global Elo ratings from 2000 to 2024 for five standout players: Roger Federer, Rafael Nadal, Novak Djokovic, Carlos Alcaraz, and Jannik Sinner. The bold lines represent these stars, while lighter gray traces in the background correspond to other top 100 active players. This layered view emphasizes the separation between the elite tier and the broader field.

We observe distinct phases of dominance: Federer's peak around 2006–2009, Nadal's climb between 2008–2013, and Djokovic's prolonged excellence starting in 2011. The recent emergence of Alcaraz and Sinner, rising sharply from 2021 onward, suggests a new generational shift.

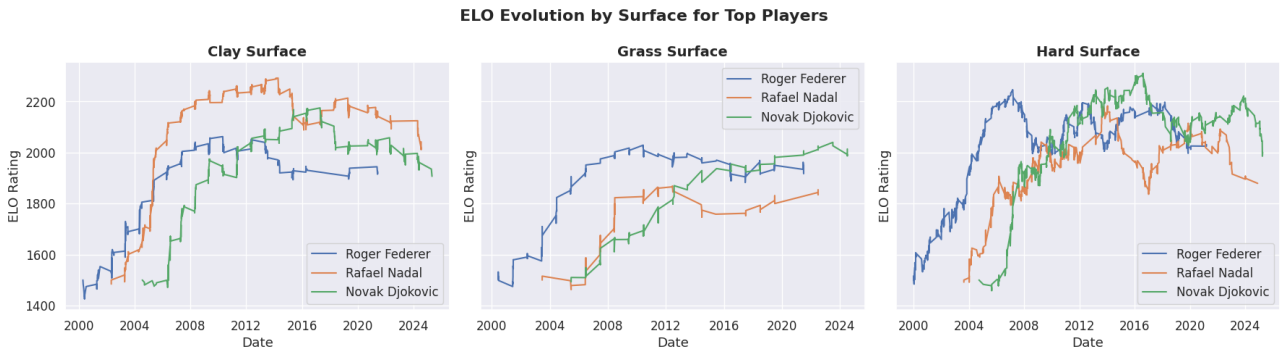

Figure 5.9: Surface-specific ELO evolution

To further analyze player strengths, Figure 5.9 breaks down Elo ratings by surface. Rafael Nadal stands out with unmatched clay Elo peaks above 2200, consistent with his reputation as the "King of Clay." Federer shows strong dominance on grass, while Djokovic exhibits more balanced ratings across all three surfaces, underscoring his versatility.

This surface-aware Elo representation is particularly valuable for downstream tournament simulations, where court type plays a decisive role. For example, a matchup between Nadal and Djokovic would yield very different win probabilities on clay versus hard court.

5.5 Serve Performance

Serving is one of the most critical aspects of a tennis match, particularly on faster surfaces where short rallies are more common. In this section, we explore two complementary perspectives: how ace rates vary across surfaces, and which players have historically excelled at producing aces.

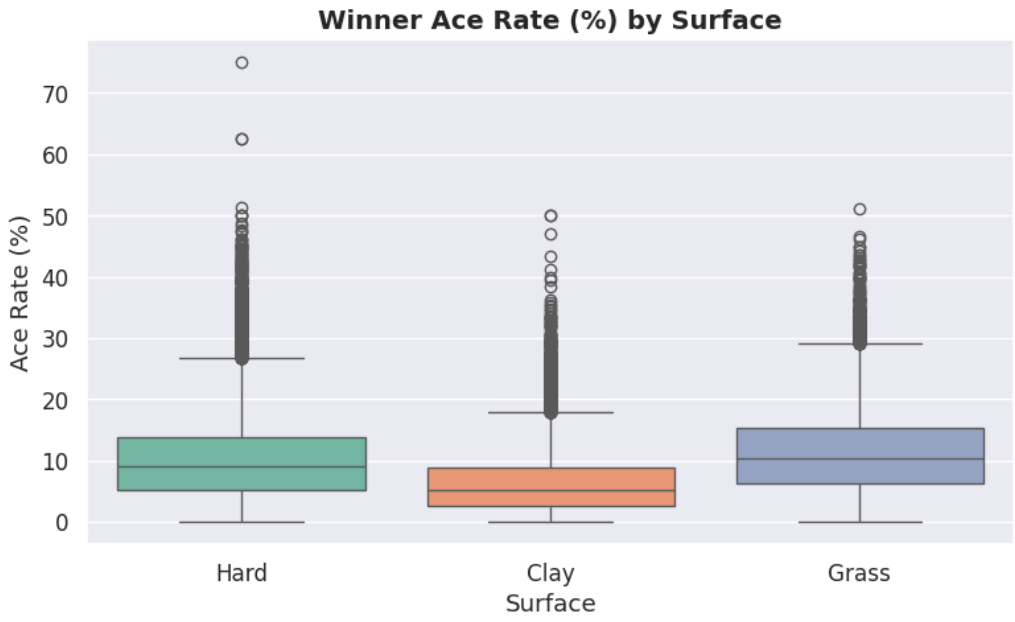

Figure 5.10: Ace rate by surface

Figure 5.10 shows the distribution of *ace rates* across different surfaces for match winners. Unsurprisingly, **Grass courts** exhibit the highest median ace percentage, followed by **Hard** and then **Clay**. This confirms the long-standing intuition that faster surfaces like grass reward strong servers, allowing for more free points on serve. On the other hand, clay courts, with their slower bounce, make it harder to hit unreturnable serves, resulting in lower ace rates on average.

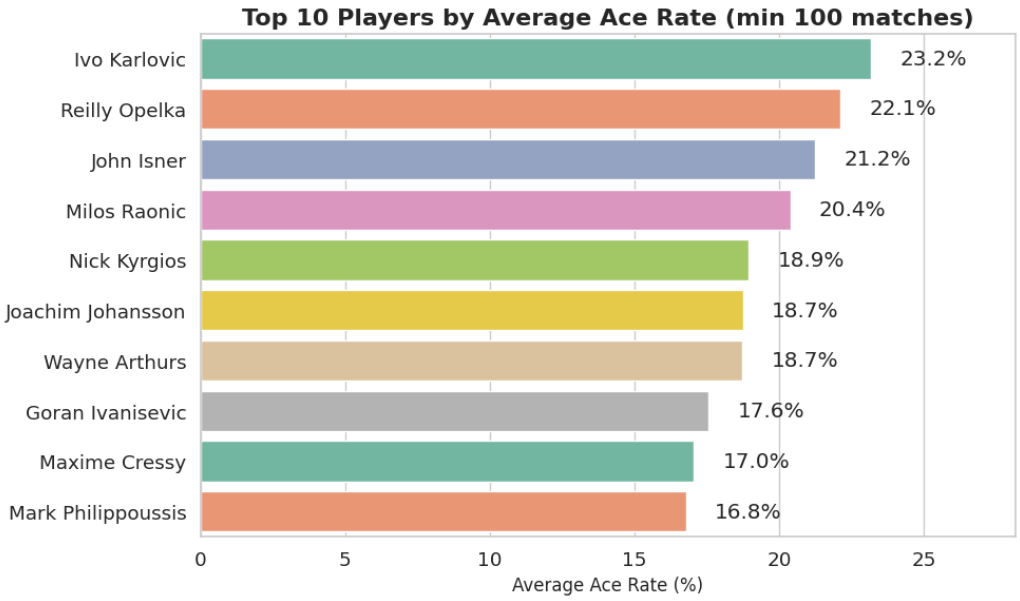

Figure 5.11: Top 10 players by ace percentage

Figure 5.11 ranks the top 10 ATP players by their average ace percentage (minimum 100 matches played). Unsurprisingly, the list is dominated by tall, powerful servers such as **Ivo Karlovic**, **Reilly Opelka**, and **John Isner**, each averaging above 21% ace rate. These players have built their game around the serve, frequently ending points without rallies and maintaining dominance in their service games.

This analysis is not only descriptive but also helps guide feature construction in predictive models. For example, surface-specific rolling ace percentages are key features in our dataset, helping capture context-aware serving ability that generalizes across matches.

5.6 Correlation Analysis

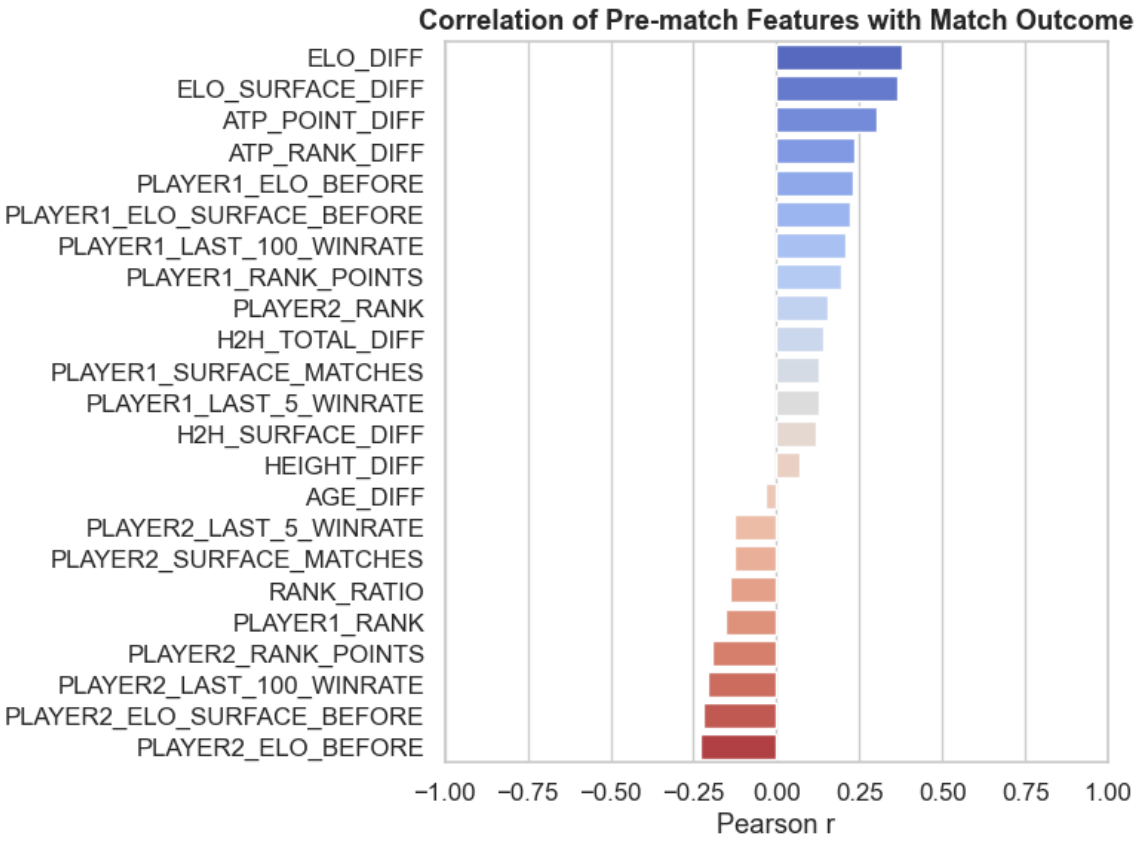

Figure 5.12: Correlation between features and match outcome (TARGET)

Understanding how features relate to each other, and to the match outcome, is a critical step before model training. Figure 5.12 shows the Pearson correlation between each pre-match feature and the binary target variable TARGET, indicating whether PLAYER1 won.

As expected, features such as ELO_DIFF, ELO_SURFACE_DIFF, and ATP_POINT_DIFF exhibit the strongest positive correlations with the match result. These variables capture relative strength indicators and validate the relevance of ELO and ranking metrics. Similarly, head-to-head statistics and recent win rates also display moderate correlation, suggesting they provide useful context.

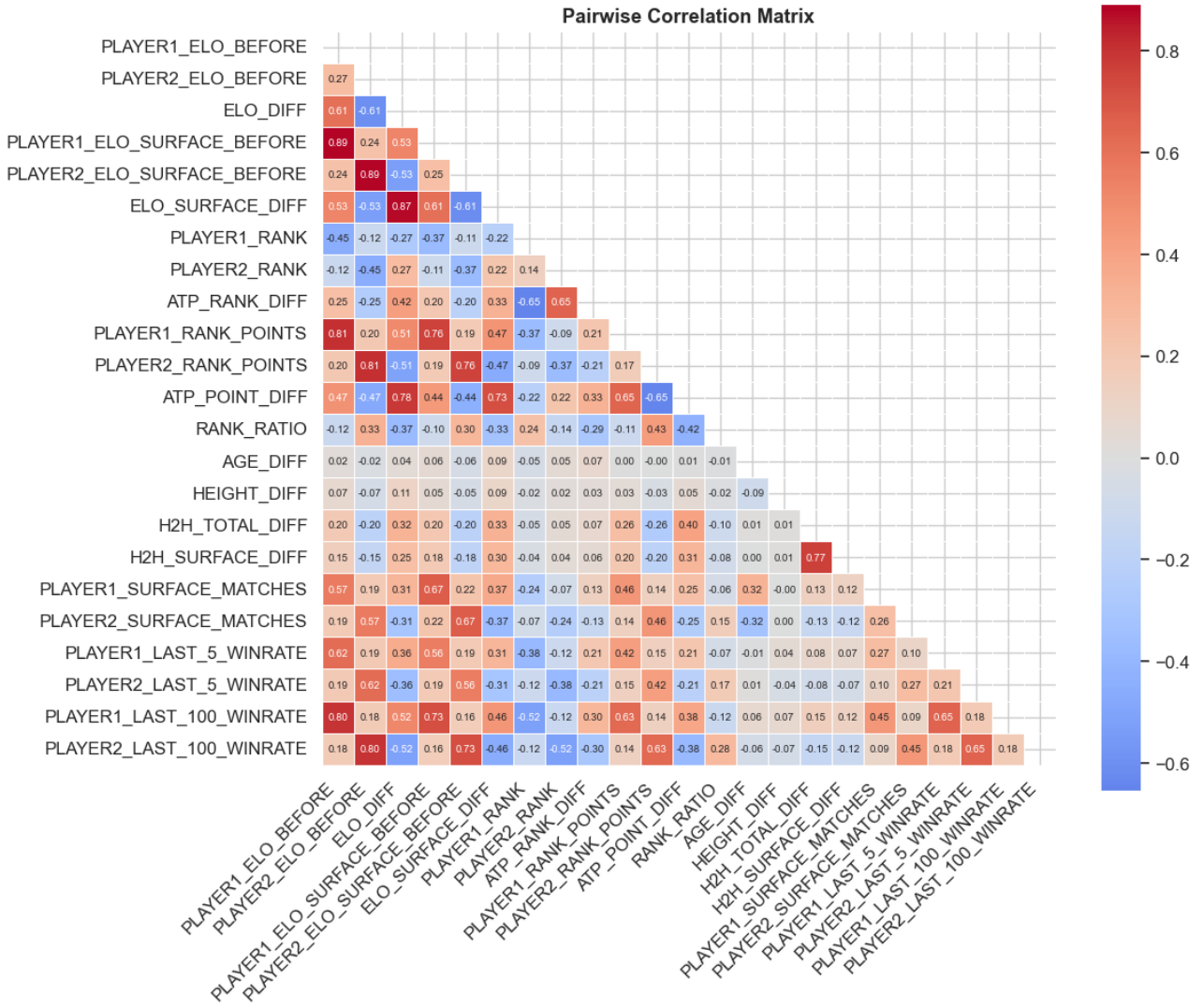

Figure 5.13: Feature correlation matrix

Figure 5.13 presents the full pairwise correlation matrix. Strong linear relationships can be observed between some engineered metrics, for instance, between ATP_POINT_DIFF and RANK_RATIO, or between surface-specific and global ELO scores. These redundancies highlight the importance of careful feature selection or dimensionality reduction during modeling, to prevent multicollinearity and overfitting.

5.7 Conclusion

This exploratory analysis confirms the richness and predictive value of the engineered dataset. By visualizing the key dimensions, match context, player demographics, historical performance, and advanced metrics, we gain intuition into what drives success on the ATP Tour. The consistency of top players, the impact of surface types, and the discriminative power of features like ELO and recent win rate all suggest that machine learning models have strong potential in this domain.

The next chapter will now explore how we train and evaluate several classification models using this dataset.

Chapter 6

Model Training and Evaluation

The training and evaluation process is a critical step in the development of any machine learning pipeline. In this thesis, particular emphasis was placed not only on ensuring reliable model selection, but also on leveraging high-performance computing (HPC) resources to scale and accelerate the training process.

After building a rich and symmetric feature set from historical ATP data, several machine learning algorithms were trained to predict match outcomes: a Decision Tree, a Random Forest, and an XGBoost model. To ensure the validity of results, we adopted a temporal split strategy , training on matches from 2000 to 2023, and evaluating on the 2024 season. This approach avoids data leakage and mirrors realistic deployment scenarios.

Among the models tested, XGBoost demonstrated the best overall performance. However, its effectiveness depends heavily on hyperparameter tuning , a task that can be computationally expensive due to the large search space and the need for repeated cross-validation.

To address this, we deployed a distributed hyperparameter optimization strategy using **Dask** combined with **SLURM** on the **Finisterrae III** supercomputer. We requested four compute nodes, each equipped with a **NVIDIA A100 GPU**, 32 CPU cores, and 40 GB of memory. Dask dynamically launched parallel training workers across the cluster, each running GPU-accelerated training sessions using XGBoost's CUDA backend.

This setup enabled the execution of over 200 parallel training tasks in under 2 minutes of wall time, while consuming more than 88 minutes of aggregated compute time , showcasing a high level of parallelism and resource utilization. Profiling reports generated during this process confirmed the scalability and efficiency of the system, with minimal communication overhead between workers.

Ultimately, this HPC-enhanced training strategy allowed us to select an optimal XGBoost configuration that reached **68.61% cross-validation accuracy** and **68.96% test accuracy on the 2024 season**. This model was then used as the backbone for match and tournament simulations.

In the following sections, we present a detailed comparison of the models, their performance metrics, feature importance analysis, and justification for selecting XGBoost as the final model.

6.1 Train/Test Split (Temporal Strategy)

Given the chronological nature of sports data and the risk of information leakage, we employed a temporal split to evaluate model generalization. Matches from 2000 to 2023 were used as the training set, while the 2024 season was reserved for final evaluation.

Older matches prior to 2000 were excluded due to data quality issues and the fact that tennis dynamics have evolved significantly over time (equipment, playstyles, physicality). This choice ensures that models are trained on relevant and representative data.

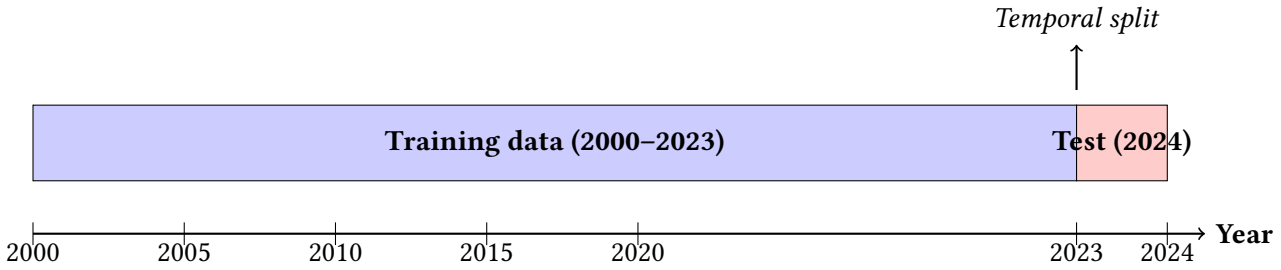

Figure 6.1: Temporal split of the dataset used for training and testing the model.

6.2 Models Selected

To balance interpretability, computational efficiency, and predictive power, we selected the following supervised learning models:

- **Decision Tree** : A simple and interpretable model that recursively splits the data to classify samples. Often used as a baseline despite its tendency to overfit.
- **Random Forest** : An ensemble of decision trees trained on different data and feature subsets. It improves generalization and provides robust feature importance scores.
- **XGBoost** : A high-performance gradient boosting algorithm that builds trees sequentially. Well-suited for structured data and able to capture complex, nonlinear patterns.

6.3 Evaluation Metrics

To evaluate the predictive performance of our classification models, we used a combination of widely adopted metrics. Each provides complementary insights into different aspects of model behavior:

- **Accuracy:** Measures the overall proportion of correct predictions. While intuitive, it can be misleading in the presence of class imbalance.

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

where:

- TP = True Positives (correctly predicted wins),
- TN = True Negatives (correctly predicted losses),
- FP = False Positives (predicted win but actually a loss),
- FN = False Negatives (predicted loss but actually a win).

- **F1 Score:** The harmonic mean of precision and recall. It is especially informative when the dataset is imbalanced, as it balances the trade-off between false positives and false negatives.

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

with:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}$$

- **Log Loss (Cross-Entropy):** A probabilistic metric that penalizes confident but incorrect predictions. Particularly relevant for models that output class probabilities.

$$\text{LogLoss} = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i)]$$

where:

- N = total number of observations,
- $y_i \in \{0, 1\}$ is the true label for instance i (e.g., 1 if player A wins, 0 otherwise),
- $\hat{p}_i$ is the predicted probability that $y_i = 1$.

In addition to these metrics, we include **confusion matrices** to visualize the distribution of predictions. This allows us to detect asymmetric errors, for instance, a tendency to overpredict wins for higher-ranked players, which may not be fully captured by scalar metrics alone.

6.4 Decision Tree

As a baseline model, we first trained a Decision Tree classifier to evaluate the potential of simple interpretable rules in predicting ATP match outcomes. This model is particularly useful for understanding which features contribute most to prediction without requiring computationally intensive training or hyperparameter tuning.

6.4.1 Training and Evaluation

The model was trained on historical ATP data from 2000 to 2023 and evaluated on the 2024 test set. As expected, training was computationally inexpensive and did not require HPC resources. The tree was limited to a depth of 3 to avoid overfitting and maintain interpretability.

The main results obtained on the test set are:

- **Accuracy:** 0.6405
- **F1 Score:** 0.6418
- **Log Loss:** 0.6232

6.4.2 Confusion Matrix Analysis

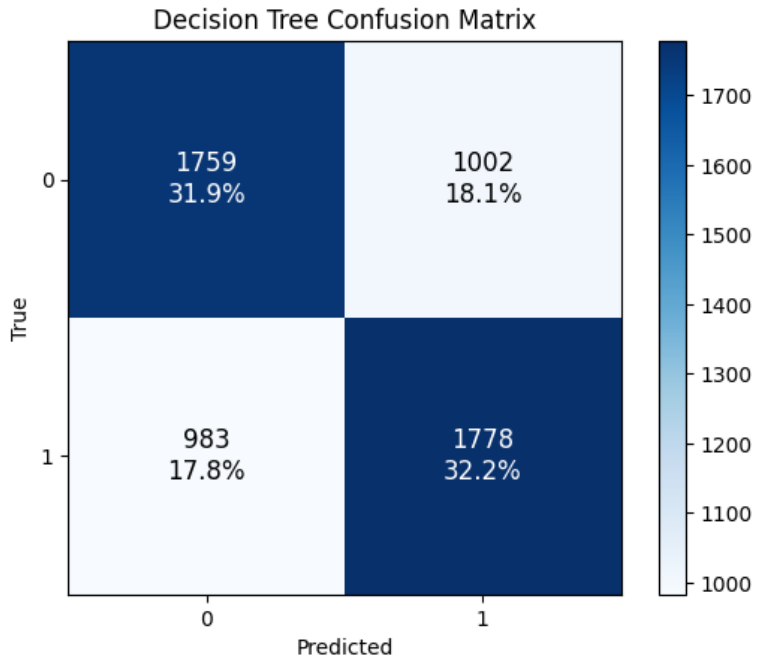

Figure 6.2: Confusion matrix on the 2024 test set.

The confusion matrix (Figure 6.2) shows a fairly balanced prediction behavior, with both classes (win/loss) having similar precision and recall (around 64%). This indicates that the model does not exhibit a significant bias toward predicting the higher-ranked player or the favorite. Given the balanced support (2761 matches per class), this is encouraging and suggests that the features used were effective in representing match competitiveness.

6.4.3 Feature Importance

Figure 6.3 highlights the dominant role played by the ELO_DIFF variable, which captures the difference in ELO rating between the two players. This is consistent with the tennis literature, where ELO has proven to be a more robust performance indicator than ATP ranking. Other relevant features include ELO_SURFACE_DIFF and surface-specific performance metrics, indicating that the model learned to account for players' surface preferences (e.g., clay vs hard court). Interestingly, variables related to recent form and serve performance (e.g., DF_LAST_5, P1STWON_LAST_5) had very limited influence in the decision path.

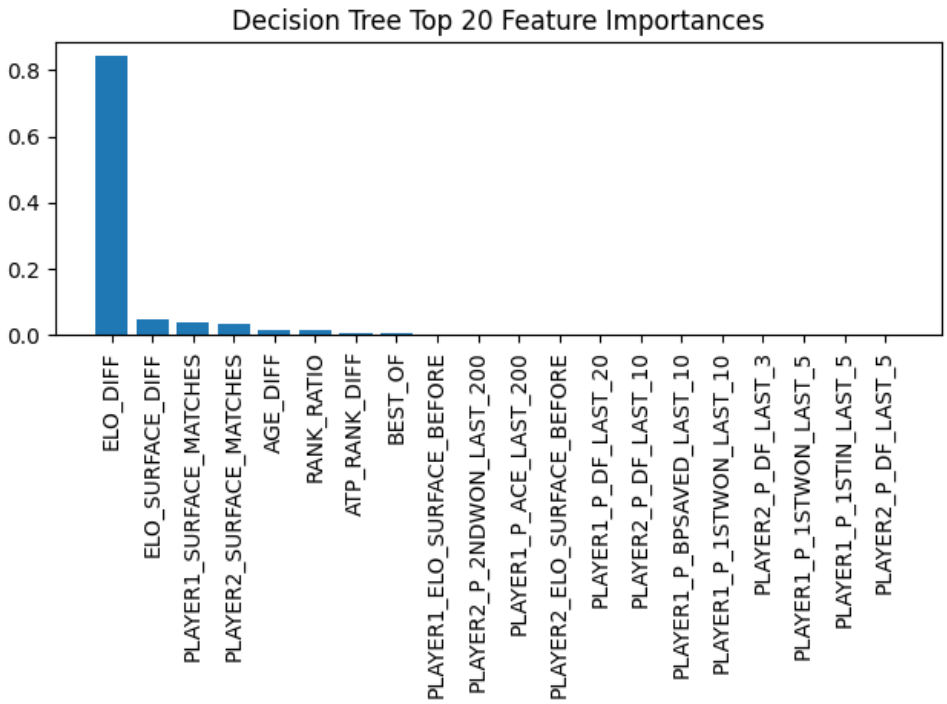

Figure 6.3: Top 20 feature importances in the Decision Tree model.

6.4.4 Model Interpretability

Figure 6.4 shows the upper part of the decision tree structure. The first split is based on ELO_DIFF, clearly separating matchups where one player is a strong favorite. Subsequent splits involve surface-specific differences and match counts, which reflect typical tennis dynamics: players perform differently on various surfaces, and experience can play a key role in outcome prediction.

6.4.5 Conclusion

Despite its simplicity, the Decision Tree provides competitive results with interpretable structure and fast training. Its performance offers a strong reference point before moving to more advanced ensemble models.

6.5 Random Forest

Random Forest is an ensemble learning method that constructs multiple decision trees from different subsets of data and features. By aggregating their predictions, it reduces variance and improves generalization, making it a natural progression from single decision trees. In the context of ATP match prediction, this allows the model to capture more subtle patterns in player form, ranking dynamics, and surface-specific performance.

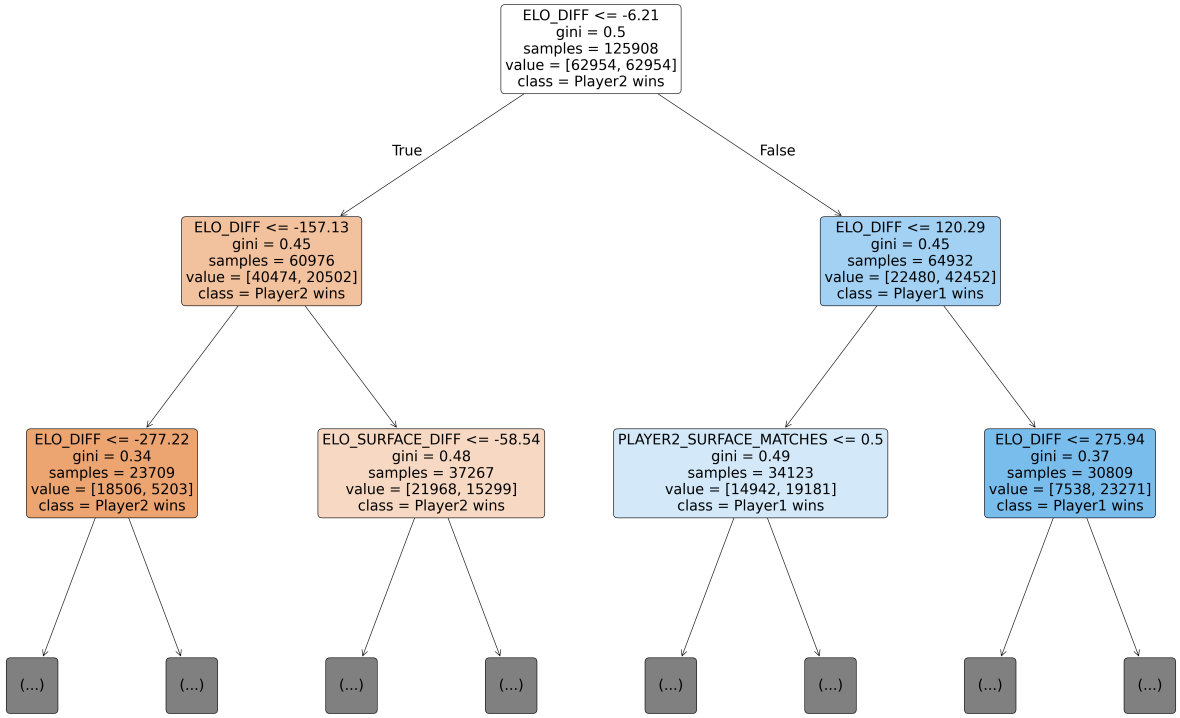

Figure 6.4: Top of the Decision Tree (depth = 3).

6.5.1 Training and Evaluation

The model was trained on historical data from 2000 to 2023 and evaluated on the 2024 test set. We adopted a simple yet effective configuration, described in Table 6.1.

Parameter	Value	Description
`n_estimators`	100	Number of trees in the forest. More trees generally improve performance but increase training time.
`max_depth`	12	Maximum depth of each tree to prevent overfitting on training data.
`min_samples_split`	30	Minimum number of samples required to split an internal node. Avoids splits on small subsets.
`class_weight`	balanced	Automatically adjusts weights inversely proportional to class frequencies to handle imbalance.
`n_jobs`	-1	Enables parallel training across all available CPU cores.
`random_state`	42	Ensures reproducibility of results.

Table 6.1: Hyperparameter configuration used for the Random Forest model.

This configuration yielded the following performance metrics on the 2024 test set:

- **Accuracy:** 0.6490
- **F1 Score:** 0.6487
- **Log Loss:** 0.6172

6.5.2 Confusion Matrix Analysis

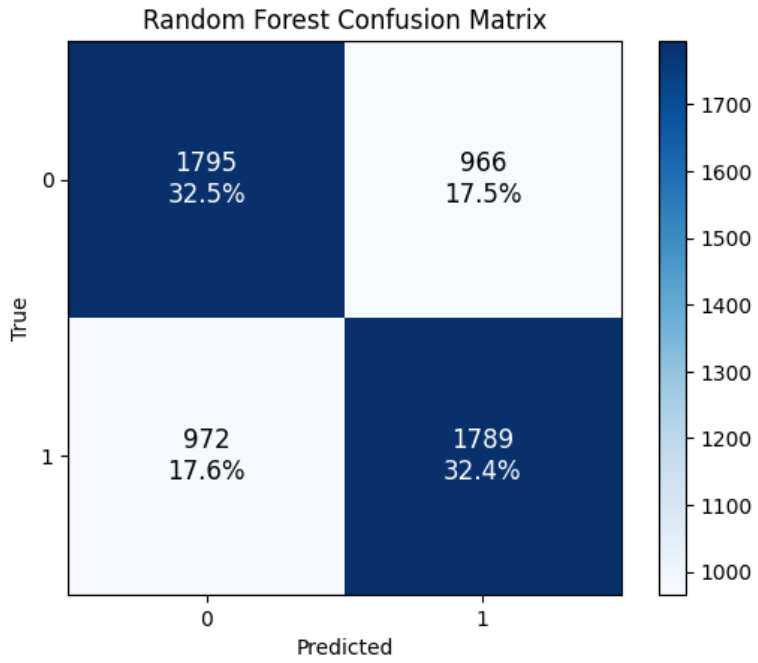

Figure 6.5: Confusion matrix for Random Forest on the 2024 test set.

The confusion matrix in Figure 6.5 shows a well-balanced classification outcome. The model correctly predicted 1795 out of 2761 losses (class 0) and 1789 out of 2761 wins (class 1), achieving almost symmetric precision and recall for both classes. The misclassification rates, 17.5% for false positives and 17.6% for false negatives, remain low, confirming that the ensemble learning strategy improves reliability compared to a single tree.

6.5.3 Feature Importance

Figure 6.6 highlights the variables most used by the Random Forest in its split decisions. The difference in ELO rating between the two players (`ELO_DIFF`) remains the most dominant feature, reaffirming its role as a robust indicator of player strength. Other key contributors include:

- `ELO_SURFACE_DIFF`: captures relative advantage on the surface of the match,
- `RANK_RATIO` and `ATP_POINT_DIFF`: quantify performance disparities from the ATP ranking perspective,
- `SURFACE_FORM` and `RECENT_WINRATE` indicators: incorporate short-term momentum and surface-specific trends.

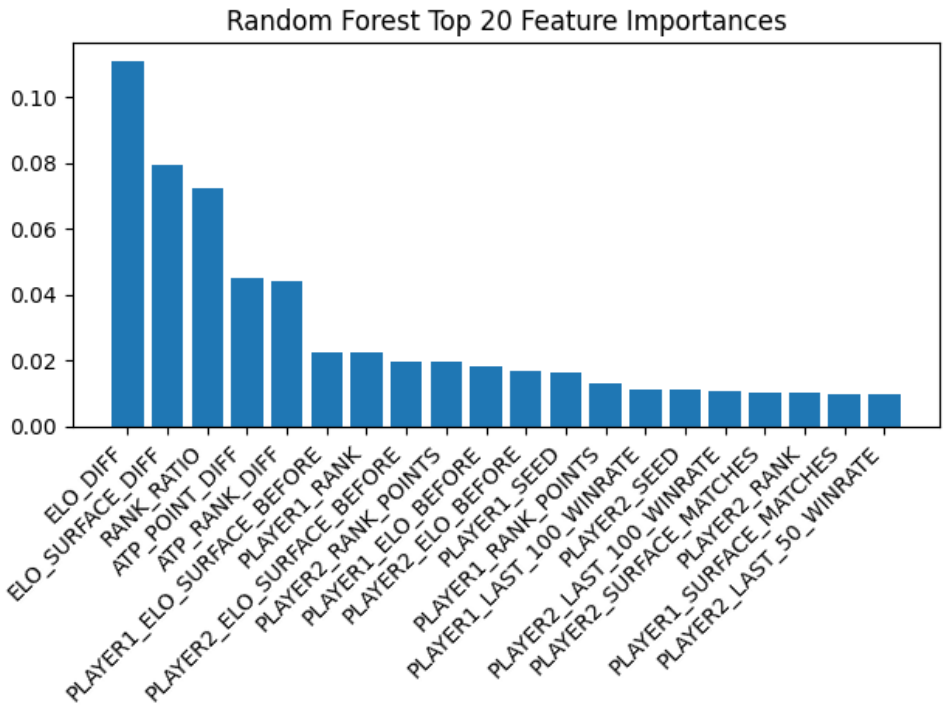

Figure 6.6: Top 20 feature importances in the Random Forest model.

This richer hierarchy of features, compared to the single decision tree, shows how the ensemble was able to capture complex interactions, balancing historical strength with recent and contextual performance.

6.5.4 Conclusion

The Random Forest model delivers an improvement in predictive performance over the standalone Decision Tree, while maintaining interpretability through feature importance analysis. Its ability to handle feature interactions and reduce variance makes it a strong candidate for large-scale simulations and tuning under HPC settings, which we explore in the next section.

6.6 XGBoost

XGBoost (eXtreme Gradient Boosting) is a high-performance implementation of gradient boosting that builds decision trees sequentially, each new tree correcting the residual errors of the previous ensemble. Known for its efficiency and scalability, it has become a go-to algorithm for structured/tabular data competitions, including applications in sports analytics. Its capacity to capture subtle interactions between features makes it especially well-suited for modeling tennis match dynamics, which often involve complex interdependencies between surface, form, ranking, and head-to-head statistics.

6.6.1 Training and Evaluation

The XGBoost model was trained on historical ATP data from 2000 to 2023 and evaluated on the 2024 test set. The configuration adopted here aimed to balance expressive power and generalization ability, and is detailed in Table 6.2.

Parameter	Value	Description
`objective`	binary:logistic	Specifies that the task is binary classification with logistic loss.
`eval_metric`	logloss	Evaluation metric used during training; lower is better.
`n_estimators`	100	Number of boosting rounds (trees) in the ensemble.
`max_depth`	6	Maximum depth of each tree; controls model complexity.
`learning_rate`	0.1	Step size shrinkage used to prevent overfitting.
`subsample`	0.8	Fraction of training samples used per tree; adds randomness and reduces variance.
`colsample_bytree`	0.8	Fraction of features used per tree; helps generalization.
`n_jobs`	-1	Trains in parallel across all available CPU cores.
`random_state`	42	Reproducibility of results.

Table 6.2: Hyperparameter configuration used for the XGBoost model.

This configuration achieved the strongest performance among the baseline models:

- **Accuracy:** 0.6603
- **F1 Score:** 0.6615
- **Log Loss:** 0.6057

6.6.2 Confusion Matrix Analysis

As shown in Figure 6.7, the XGBoost classifier correctly predicted 1813 out of 2761 losses (class 0) and 1833 out of 2761 wins (class 1), leading to a very symmetric and balanced matrix. The model's false positive and false negative rates are both under 17.5%, marking a slight improvement over both the Decision Tree and Random Forest models.

This improved performance suggests that XGBoost successfully captures finer-grained patterns within the data, such as recent form, nuanced surface tendencies, or latent interactions between performance indicators, that simpler models might overlook.

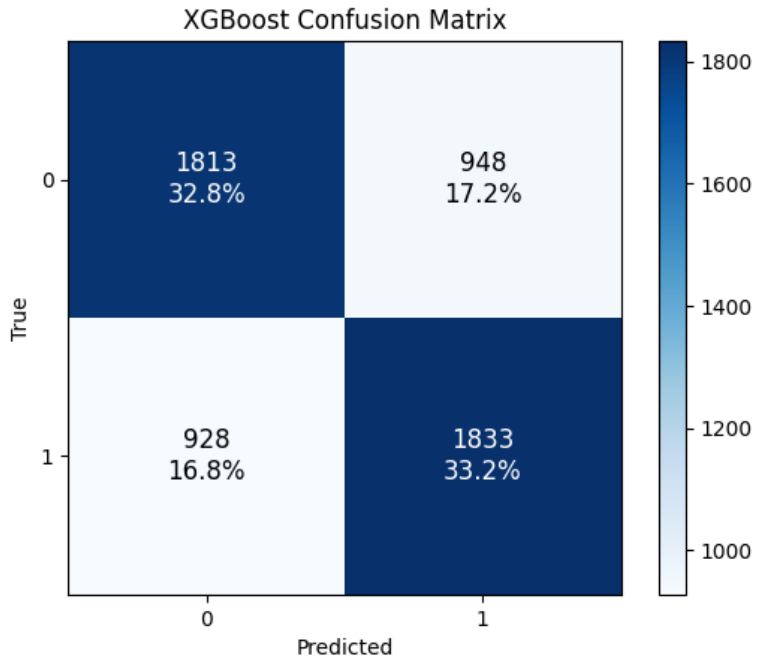

Figure 6.7: Confusion matrix for XGBoost on the 2024 test set.

6.6.3 Feature Importance

Figure 6.8 illustrates the top 20 most important features for XGBoost. Once again, ELO_DIFF dominates the model's decision-making, reaffirming its role as the most informative feature in ATP match prediction.

Notably, the model also places strong emphasis on:

- ATP_POINT_DIFF and ELO_SURFACE_DIFF, which capture performance gaps on general and surface-specific levels,
- RANK_RATIO and AGE_DIFF, adding demographic and ranking context,
- Recent winrate indicators (LAST_3_WINRATE, LAST_5_WINRATE), which help model short-term form and momentum.

Compared to the Random Forest, the XGBoost model appears to make more efficient use of features lower in the importance ranking, suggesting that the boosting framework benefits from both shallow and deep feature interactions.

6.6.4 Conclusion

XGBoost outperformed the other baseline models in all evaluation metrics, confirming its strength in handling structured datasets like historical tennis results. Its ability to learn from both strong predictors and subtle signals, while maintaining regularization and generalization, makes it the most promising candidate for tournament simulation and parallel optimization workflows. In the following section, we explore how XGBoost was scaled using HPC resources for hyperparameter tuning and simulation batching.

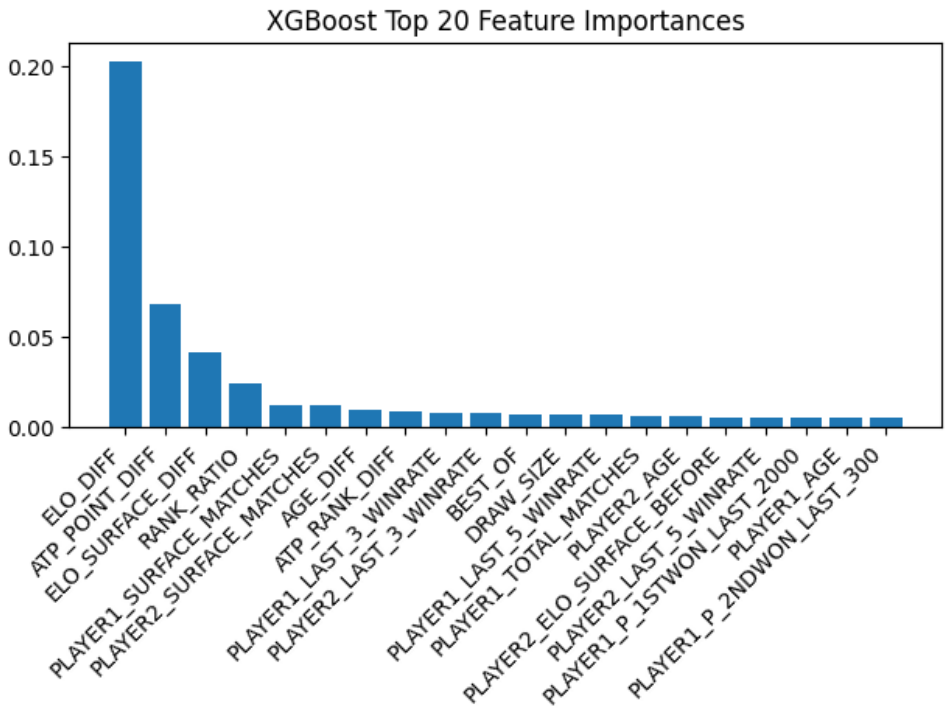

Figure 6.8: Top 20 feature importances in the XGBoost model.

6.7 XGBoost Distributed Hyperparameter Tuning with Dask

To enhance model performance and harness the power of HPC infrastructure, we implemented a distributed hyperparameter tuning strategy for XGBoost using **Dask** in conjunction with **SLURM** on the Finisterrae III supercomputer.

6.7.1 Motivation and Objective

Gradient boosting models such as XGBoost provide high predictive power but require careful tuning of many hyperparameters. Manual tuning is inefficient, and grid search is computationally expensive. To overcome these limitations, we used `RandomizedSearchCV` from `dask-ml`, which distributes the hyperparameter search across multiple parallel workers.

6.7.2 Infrastructure Setup with SLURM and Dask

We executed the tuning process on 4 SLURM nodes, each equipped with:

- 1 NVIDIA A100 GPU,
- 32 CPU cores and 40 GB of RAM,
- A wall-time limit of 4 hours.

Instead of launching training jobs directly, we used the `SLURMCluster` interface from `dask-jobqueue`, which dynamically manages Dask workers. The script below shows the SLURM configuration:

```
#!/bin/bash -l
#SBATCH --job-name=xgb-dask
#SBATCH --output=logs/dask/xgb-dask-%j.out
#SBATCH --error=logs/dask/xgb-dask-%j.err
#SBATCH --nodes=4
#SBATCH --ntasks=4
#SBATCH --cpus-per-task=32
#SBATCH --gres=gpu:a100:1
#SBATCH --mem=40G
#SBATCH --time=04:00:00

module load python
source $STORE/mypython/bin/activate

python hyperparam_tune_xgb_dask.py \
  --parquet ../../../Datasets/final_tennis_dataset_symmetric.parquet \
  --cutoff 2025-01-01 \
  --output ./logs/dask/best_xgb_params \
  --n-iter 50 \
  --n-splits 4 \
  --jobs 4
```

6.7.3 Workflow Overview

The script `hyperparam_tune_xgb_dask.py` follows these main steps:

1. **Data loading and filtering**: The parquet file is loaded with Dask and filtered to include only matches before the cutoff date.
2. **Train/test split**: The training set covers 2000–2023; the 2024 data is held out for final evaluation.
3. **Cluster initialization**: A `SLURMCluster` is launched to manage distributed GPU workers.
4. **Hyperparameter search**: A `RandomizedSearchCV` with 50 iterations and 4-fold cross-validation is run.
5. **Profiling and logging**: Dask's `performance_report` captures GPU usage, task distribution, and resource consumption.
6. **Result storage**: All CV results and final test accuracy are saved as CSV and JSON files.

6.7.4 Performance Insights

The tuning process took just **128 seconds** wall-clock time, but used over **88 minutes** of cumulative compute time across the cluster. This gap reflects the efficient distribution of 233 training tasks across 4 GPU nodes

(128 threads total).

Key metrics:

- **Total tasks**: 233
- **Cumulative compute time**: 88 minutes 13 seconds
- **Data transfer time**: 9.66 seconds
- **Cluster size**: 4 GPUs, 128 threads, 149 GiB of memory

The figures below provide a visual breakdown of this performance profile:

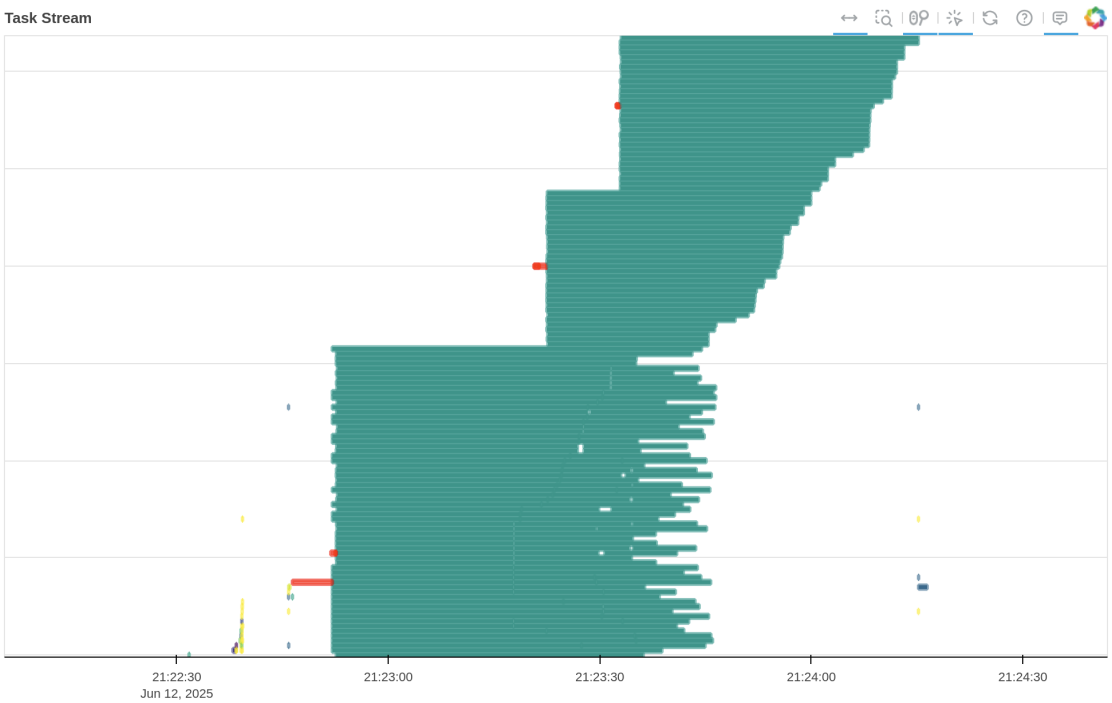

Figure 6.9: Dask Task Stream view. Each horizontal bar represents a scheduled task on one worker. Tasks are densely packed with minimal idle time, confirming efficient parallelism.

Figure 6.9 shows the task distribution timeline. We observe dense and contiguous execution blocks with few gaps, which implies that tasks were distributed and executed efficiently across the 4 workers. The color variation corresponds to different task types (model fitting, data partitioning, scoring, etc.).

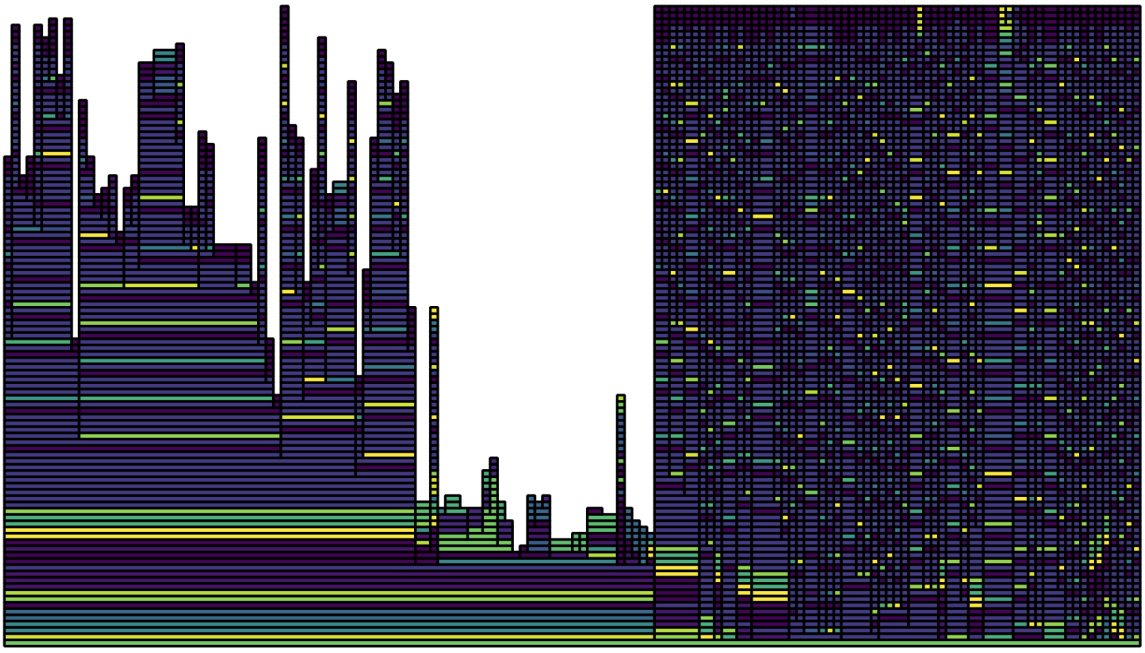

Figure 6.10: Worker-level profiling of task execution and CPU/GPU usage. Workers exhibit balanced loads and synchronized progress across all hyperparameter trials.

In Figure 6.10, each row represents a thread. The high density of colored rectangles suggests sustained activity on all threads, with no significant imbalance or saturation. The synchronization of blocks on the left indicates coordinated task bursts during model training and evaluation.

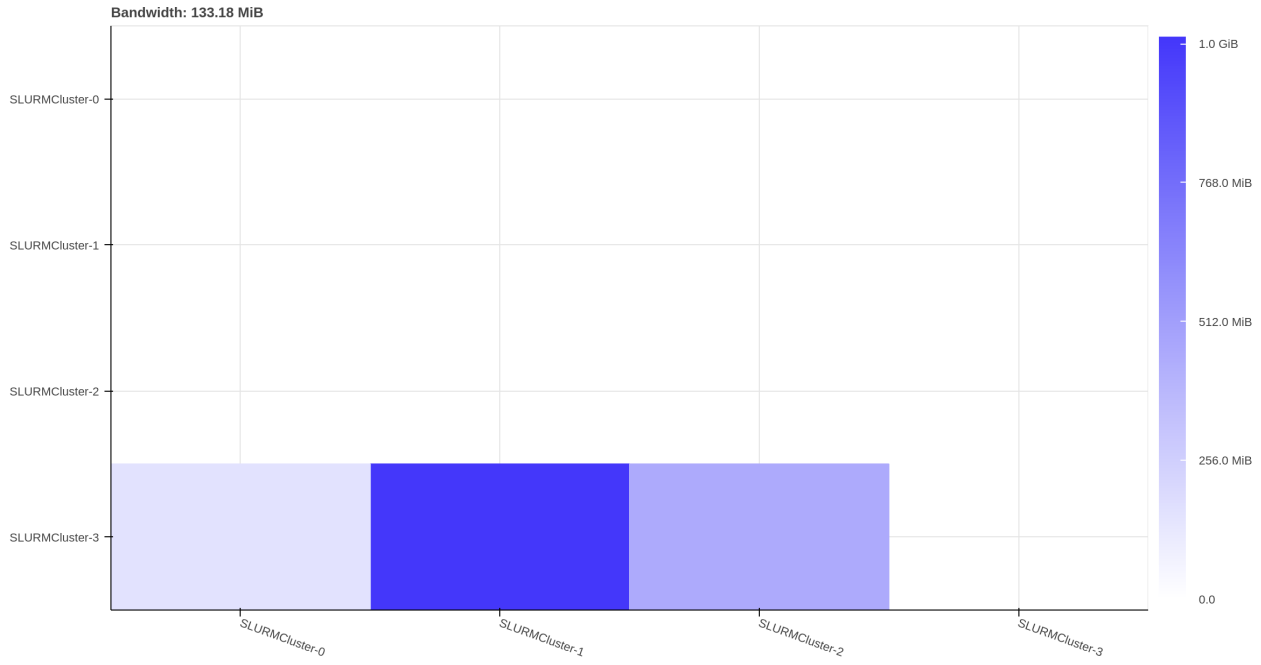

Figure 6.11: Inter-worker bandwidth usage. Low values confirm that Dask spent very little time in communication, consistent with the expected behavior of parallel cross-validation.

Figure 6.11 confirms that the total inter-worker bandwidth was just **133.18 MiB**. This is extremely low, meaning the training workload was embarrassingly parallel , each hyperparameter trial was independent, and there was minimal data shuffling.

6.7.5 Results Summary

The best hyperparameter combination found was:

```
"best_params": {
    "subsample": 0.9,
    "reg_lambda": 5,
    "reg_alpha": 0.5,
    "n_estimators": 300,
    "max_depth": 4,
    "learning_rate": 0.1,
    "colsample_bytree": 0.8
}
```

- **Best CV Accuracy:** 68.61%
- **Test Accuracy on 2024:** 68.96%

6.7.6 Conclusion and Transition

This experiment confirms that Dask and SLURM provide a scalable and efficient framework for large-scale hyperparameter tuning. By distributing the workload across four GPU nodes, we reduced total tuning time from what would have taken over an hour on a single machine to just **128 seconds of wall-clock time**, while completing 233 hyperparameter trials in parallel.

This setup achieved a **41.25 speedup** compared to sequential execution (88 minutes vs. 128 seconds), demonstrating the practical power of distributed GPU computing for scalable hyperparameter optimization.

Such performance is essential in modern data workflows where repeated experimentation and model selection are bottlenecks. The seamless integration of Python, XGBoost, and HPC tools made this approach both powerful and accessible.

The final model obtained here will now serve as the basis for tournament-level simulations presented in the next chapter, where predictive performance and generalization capabilities are further tested at scale.

6.8 Conclusion

This chapter presented the full pipeline used to train, evaluate, and optimize predictive models for ATP match outcomes. We progressively improved performance from a simple Decision Tree to an optimized XGBoost classifier, using temporal validation and relevant evaluation metrics.

Among the tested models, XGBoost emerged as the most accurate and robust option, thanks to its capacity to integrate complex interactions between features like ELO rating, surface specialization, recent form, and head-to-head history. To further enhance its performance, we deployed a distributed hyperparameter tuning

strategy using Dask and SLURM on the Finisterrae III supercomputer. This enabled us to evaluate 233 candidate models in under two minutes of wall-clock time, demonstrating the value of parallel computing for scalable machine learning.

The optimized XGBoost model, with a cross-validation accuracy of **68.61%** and a test accuracy of **68.96%** on the 2024 season, will now serve as the foundation for the next phase of the project.

In the following chapter, we shift our focus from individual predictions to full-scale simulations. We evaluate the model's ability to replicate real tournament outcomes and explore probabilistic scenarios using Monte Carlo simulations. This allows us to test not only accuracy, but also how well the model internalizes the structure and uncertainty of competitive tennis.

Chapter 7

Match and Tournament Simulation

Among all the components of this project, this chapter was undoubtedly my favorite. After months of engineering features, training models, and evaluating metrics, I finally reached the point where I could answer a simple, intuitive question: *Can my model actually predict real tennis matches?*

This part felt particularly rewarding because I could observe the results in real time, especially during Roland Garros 2025, which was happening as I ran the simulations. Watching how the predictions aligned (or didn't) with live outcomes gave me a deeper sense of what the model had truly learned. Was it just good on paper, or could it really simulate what unfolds on court?

To answer that, I structured this chapter in two complementary parts:

- First, a round-by-round simulation of real tournaments, comparing predicted winners to actual results.
- Second, a Monte Carlo simulation engine capable of generating thousands of alternate tournaments from scratch, allowing us to explore uncertainty, repeatability, and player dominance over many hypothetical scenarios.

Both approaches reflect the same underlying question: not just whether the model is accurate, but whether it "understands" the structure of competitive tennis. Let's dive into it.

7.1 Predicting Individual Matches

To evaluate the model's ability to capture tennis-specific dynamics, we simulate head-to-head matchups between top players on different surfaces. This approach helps validate whether the model correctly internalizes the impact of surface type, one of the most critical factors in professional tennis outcomes.

All predictions are generated using the final **XGBoost** model trained on historical ATP data (2000–2023), and fed with the most up-to-date statistics available for each player at the time of the simulation. These input features include both traditional metrics (ranking, age, titles) and engineered variables such as rolling win rates, surface-adjusted ELO, and recent performance indicators. The system pulls relevant pre-match data from the processed dataset, ensuring that the predictions are based on contextually realistic inputs.

7.1.1 Case Study 1: Carlos Alcaraz vs Jannik Sinner

As a first example, we simulate a best-of-three match between **Carlos Alcaraz** and **Jannik Sinner** on clay, hard, and grass. The two players have developed a strong rivalry in recent seasons, marked by tightly contested encounters on all surfaces.

Carlos Alcaraz

VS

Jannik Sinner

Figure 7.1: Simulated matchup: Alcaraz vs Sinner.

Surface	Alcaraz Win Probability	Sinner Win Probability
Clay	49.5%	50.5%
Hard	48.6%	51.4%
Grass	51.3%	48.7%

Table 7.1: Predicted outcomes for Alcaraz vs Sinner on different surfaces.

The results highlight how the model adjusts its predictions based on surface. On clay, where Alcaraz is known for his agility and topspin-heavy game, the match is essentially even. On hard courts, Sinner gains a slight edge, reflecting his explosive baseline play and improving serve. On grass, the balance tilts marginally back to Alcaraz, whose athleticism and net play offer an advantage.

This level of nuance suggests that the model does not simply rely on ranking or form in isolation, but integrates a range of contextual factors when estimating match outcomes.

7.1.2 Surface Sensitivity and Model Credibility

This simple experiment provides qualitative validation of the model's surface sensitivity. The variations in win probability reflect realistic player tendencies, reinforcing the idea that the model internalized key performance indicators, especially those engineered around surface-specific ELO and winrates.

7.2 Simulating Real Tournaments

7.2.1 Methodology

In this approach, we simulate entire Grand Slam tournaments by replaying matches round by round using the trained **XGBoost** model. For each match, we extract the most recent pre-match statistics for both players and predict the outcome based on win probabilities. The winner is determined as the player with the highest predicted probability of victory.

This method allows us to compare simulated results to actual outcomes, providing a concrete measure of how well the model performs under real tournament conditions.

7.2.2 Australian Open 2025

We first applied our simulation to the 2025 Australian Open men's singles draw. A total of 107 matches were evaluated.

Overall accuracy: 70.09% (75 correct predictions out of 107)

Accuracy by round:

- 1st Round: 69.81% (37/53)
- 2nd Round: 62.96% (17/27)
- 3rd Round: 78.57% (11/14)
- 4th Round: 83.33% (5/6)
- Quarterfinals: 75.00% (3/4)
- Semifinals: 50.00% (1/2)
- Final: 100.00% (1/1)

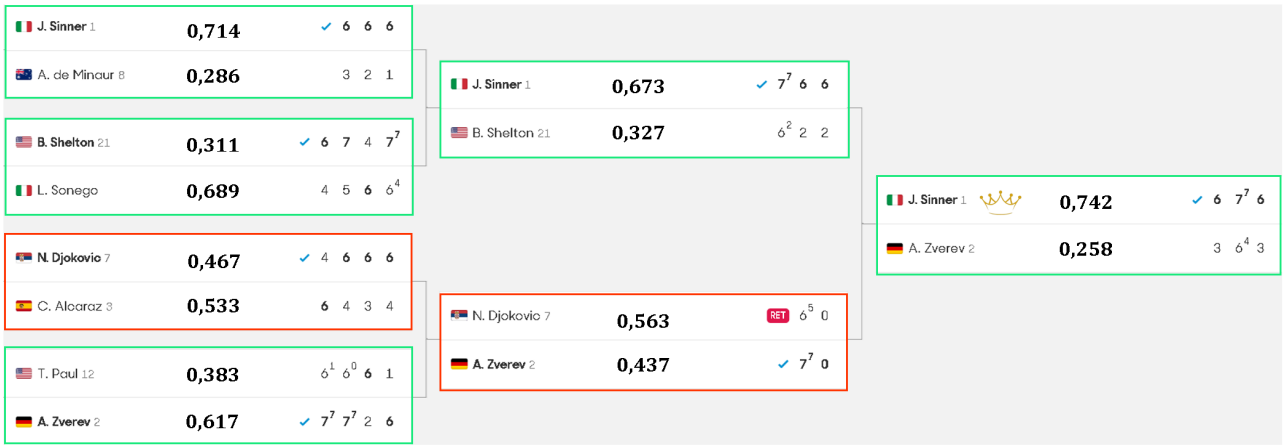

Figure 7.2: Simulated bracket from the quarter-finals onwards – Australian Open 2025.

The model achieved excellent performance in early rounds and finals, correctly predicting most quarterfinalists and the eventual champion, **Jannik Sinner**. The only missed prediction occurred in the semifinal between Djokovic and Zverev, where the model favored Djokovic (`prob = 56.3%`), but Zverev won the match. The predicted probability in the final was `Sinner: 74.2%` vs `Zverev: 25.8%`, and Sinner indeed claimed the title.

These results show strong agreement between predicted and actual outcomes, especially in the later stages where form and surface-adjusted features become highly discriminative.

7.2.3 Roland Garros 2025

We also ran a full simulation of the 2025 Roland Garros men's singles tournament. In total, **127 matches** were evaluated, with an overall accuracy of **73.23%**.

Accuracy by round:

- 1st Round: 75.00% (48/64)
- 2nd Round: 59.38% (19/32)
- 3rd Round: 87.50% (14/16)
- 4th Round: 75.00% (6/8)
- Quarterfinals: 100.00% (4/4)
- Semifinals: 50.00% (1/2)
- Final: 100.00% (1/1)

The model was able to make highly accurate predictions in the final rounds. In the semifinals, it correctly predicted **Alcaraz** defeating **Musetti** (probability: 37.7%), but incorrectly predicted **Djokovic** over **Sinner**, assigning only a 25.8% win chance to the actual winner, Sinner.

Figure 7.3: Simulated bracket from the quarter-finals onwards – Roland Garros 2025.

The model performed superbly, almost three out of four matches were predicted correctly, and it was really interesting to follow the results of the matches as the tournament progressed, hoping each time that my model predicted the right result. Obviously, the fact that the favourites won the tournaments was a big advantage for the results.

7.2.4 Discussion

These round-by-round simulations validate the quality of our model under tournament constraints. While early rounds contain a higher degree of unpredictability due to limited context, the model performs remarkably well from the quarterfinals onward. The success in predicting champions (Sinner at AO, Alcaraz at RG) and the strong bracket alignment confirm that the model generalizes well and captures essential tennis-specific patterns (surface specialization, momentum, player strength).

The differences in prediction accuracy by round also reflect tournament dynamics, with earlier rounds subject to greater variability, while deeper rounds reward models that can accurately integrate form, surface familiarity, and historical strength.

7.3 Monte Carlo Simulations

7.3.1 Rationale and Methodology

To estimate the probability distribution of potential tournament outcomes, we implemented a Monte Carlo simulation strategy tailored to the actual participants of each Grand Slam. Rather than sampling random players from a broader database, each simulation begins with the official list of 128 players who entered

the 2025 Australian Open or Roland-Garros. The idea is to reproduce realistic tournament scenarios while allowing for the inherent randomness of bracket placements and match outcomes.

Each simulation proceeds as follows:

- A new single-elimination bracket is randomly generated by shuffling the 128-player list (similar to how actual draws are randomized).
- For every match in the bracket, the win probability of Player A against Player B is predicted using the trained XGBoost model.
- A random number $r \in [0, 1]$ is sampled. If $r < P(\text{win})$, then Player A advances; otherwise, Player B proceeds to the next round.
- This process continues until a champion is declared.

To illustrate this logic, suppose the model predicts a 55% chance that Player A defeats Player B. A random number $r = 0.72$ is drawn. Since $r > 0.55$, Player B unexpectedly wins this match and advances in the bracket. This stochastic process is key to capturing upsets and tournament variability.

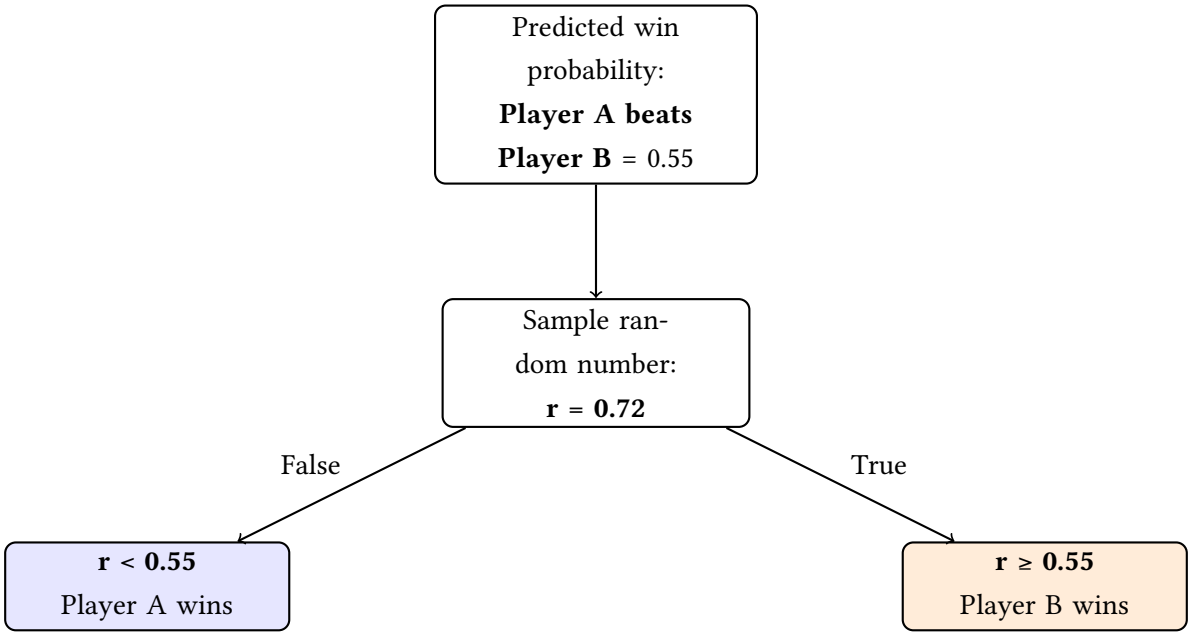

Figure 7.4: Illustration of a Monte Carlo match decision

We repeat this simulation 5000 times for each Grand Slam. By aggregating the results (e.g., number of tournament wins, semifinal appearances, quarterfinal appearances), we obtain probabilistic insights into the performance of each player across multiple alternate tournament histories.

This approach mirrors the real-world uncertainty of sports competitions, where slight changes in the draw or match dynamics can lead to very different outcomes. It also enables us to estimate confidence levels for potential champions and to rank players by their overall likelihood of success.

7.3.2 Distributed Simulation with SLURM

To scale up our Monte Carlo simulations efficiently, we leveraged the **Finisterrae III** supercomputer and its SLURM job scheduler. Instead of simulating 5000 tournaments sequentially, we divided the workload into independent array jobs, where each GPU runs a portion of the simulations in parallel.

For each tournament (Australian Open and Roland-Garros), we ran experiments with **1, 2, 4, 8, and 16 A100 GPUs**. The total number of tournaments per experiment was fixed at 5000. SLURM's array mechanism allowed us to dispatch several jobs concurrently, each simulating, for example, 1250 tournaments when using 4 GPUs (i.e., 4 tasks with `--array=0-3`).

Here is a simplified example of the SLURM configuration used:

```
# SLURM Configuration for 2 GPUs
#SBATCH --job-name=mc-rg-2gpu
#SBATCH --output=logs/2gpu/mc-rg-2gpu-%A_%a.out
#SBATCH --error=logs/2gpu/mc-rg-2gpu-%A_%a.err
#SBATCH --array=0-1
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=32
#SBATCH --gres=gpu:a100:1
#SBATCH --mem=40G
#SBATCH --time=02:00:00
```

Each job activates the Python environment, then launches the simulation script with specific arguments for input data, number of runs, and output location. For example:

```
python monte_carlo_rg.py \
  --json-draw roland_garros_2025_complete_final.json \
  --parquet final_tennis_dataset_symmetric.parquet \
  --model xgb_model.json \
  --cutoff 2025-01-01 \
  --runs-per-job 2500 \
  --job-index $SLURM_ARRAY_TASK_ID \
  --output-dir ./mc_rg_results_2gpu
```

7.3.3 Scalability Analysis and Performance Insights

To evaluate the scalability of our Monte Carlo tournament simulation pipeline, we measured three key metrics under different GPU configurations: execution time, speedup, and parallel efficiency. Each experiment involved **5000 tournament simulations**, with jobs distributed using SLURM across **1, 2, 4, 8, and 16 A100 GPUs**. **Execution Time**

Figure 7.5 shows the total simulation time in seconds (with an additional axis for hours). We observe a sharp drop in execution time when moving from 1 to 2 GPUs, and then a near-linear improvement up to 8 GPUs. With 16 GPUs, the execution time still decreases, but more modestly, reflecting diminishing returns due to increased scheduling overhead and lower workload per GPU.

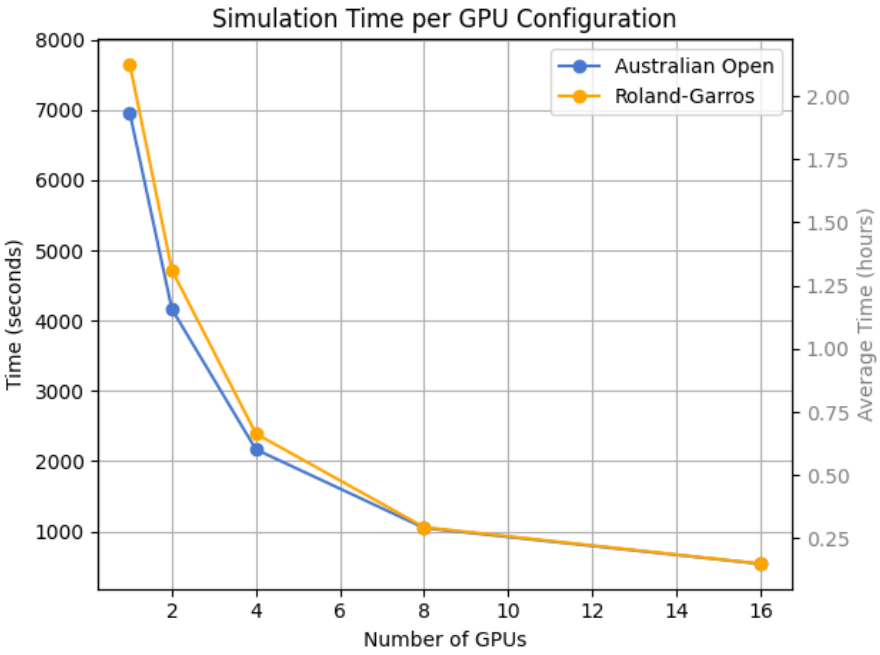

Figure 7.5: Simulation time per GPU configuration (in seconds and hours).

Speedup

Figure 7.6 compares the actual speedup achieved against the ideal linear scaling baseline. Both Australian Open and Roland-Garros exhibit a nearly proportional speedup up to 8 GPUs. At 16 GPUs, the speedup plateaus slightly, due to fixed total workload (5000 simulations), which results in under-utilization of each GPU and limited opportunity for further gains.

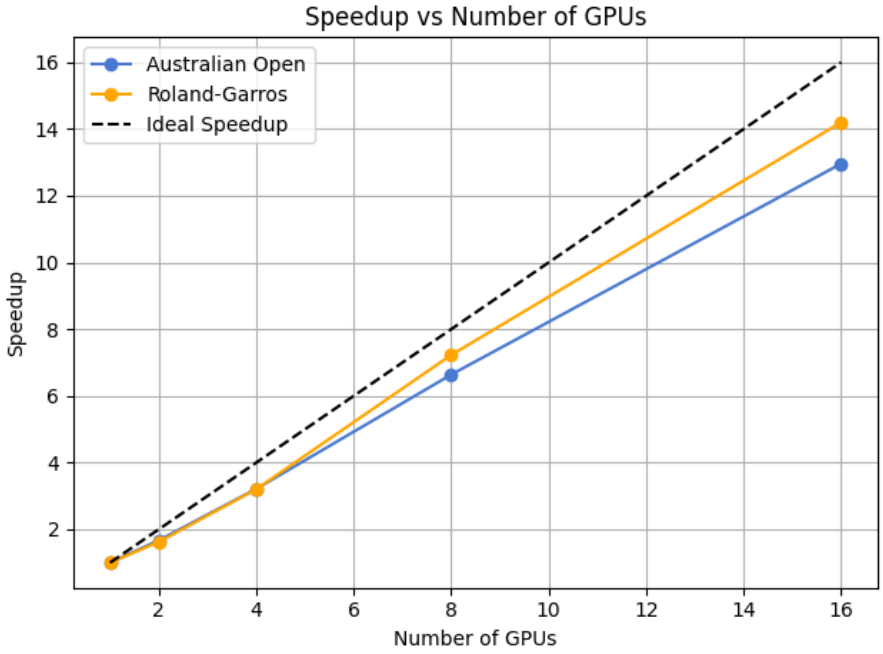

Figure 7.6: Speedup as a function of GPU count.

Parallel Efficiency

Figure 7.7 illustrates the parallel efficiency, computed as $\text{Efficiency} = \frac{\text{Speedup}}{\text{Number of GPUs}}$. Results remain above 80%

for all configurations, which is a strong indicator of good resource usage. The best efficiencies are achieved at 1 and 2 GPUs. Slight drops occur with higher counts due to typical HPC bottlenecks such as job scheduling, data loading, and task coordination.

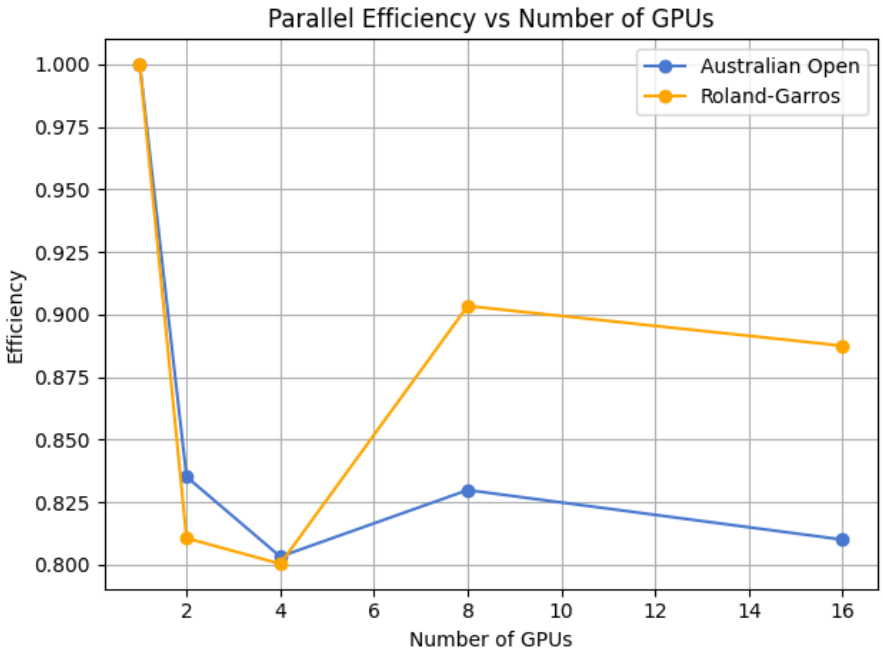

Figure 7.7: Parallel efficiency as a function of GPU count.

Discussion and Conclusion

This scalability study confirms that our simulation pipeline is well-suited to HPC environments. The task is *embarrassingly parallel*, meaning each GPU can simulate tournaments independently, without any inter-process communication. This independence is what allows us to maintain high efficiency even as GPU count increases.

Moreover, the total simulation time dropped from over **2 hours to less than 10 minutes**, depending on the configuration, enabling rapid experimentation and statistical robustness.

SLURM job arrays combined with GPU acceleration proved essential for handling this scale of workload. This experiment illustrates not only the viability but also the **practical power of distributed computing for sports analytics applications**.

Having generated over 5000 realistic tournament simulations for each Grand Slam, we now analyze the aggregated results to identify the most frequent winners, semifinalists, and other top performers.

7.3.4 Insights from 5000 Simulated Tournaments

We simulated **5000 tournaments** for each Grand Slam (Australian Open and Roland Garros), using the same probabilistic rules and feature snapshots as in earlier sections.

Table 7.2: Australian Open 2025 – Monte Carlo results (5000 simulations)

Player	Wins	Runs	Champion %	SF	SF %	QF	QF %
Sinner J.	844	5000	*16.88%*	1480	*29.60%*	2218	*44.36%*
Alcaraz C.	535	5000	*10.70%*	920	*18.40%*	1438	*28.76%*
Rublev A.	284	5000	*5.68%*	1067	*21.34%*	1718	*34.36%*
De Minaur A.	276	5000	*5.52%*	973	*19.46%*	1591	*31.82%*
Korda S.	217	5000	*4.34%*	527	*10.54%*	1037	*20.74%*
Medvedev D.	216	5000	*4.32%*	782	*15.64%*	1184	*23.68%*
Shelton B.	159	5000	*3.18%*	499	*9.98%*	834	*16.68%*
Djokovic N.	151	5000	*3.02%*	397	*7.94%*	637	*12.74%*
Mensik J.	134	5000	*2.68%*	512	*10.24%*	849	*16.98%*
Zverev A.	127	5000	*2.54%*	488	*9.76%*	758	*15.16%*

The Monte Carlo simulation for the Australian Open 2025 identified Jannik Sinner as the most likely tournament winner, with a champion probability of **16.88%**, followed by Carlos Alcaraz (**10.70%**) and Andrey Rublev (**5.68%**). Remarkably, this aligns perfectly with the actual outcome of the tournament, where Sinner claimed the title after defeating Alexander Zverev in straight sets in the final. While Zverev only ranked 10^{th} in simulated win probability (**2.54%**), his advancement to the final is still consistent with the simulation assigning non-negligible chances to lower-ranked contenders. This result demonstrates the model's ability to correctly capture players' surface-specific strengths and recent form, especially on hard courts. Overall, the outcome validates the model's predictive capabilities and shows that probabilistic simulations can produce highly realistic tournament landscapes.

Table 7.3: Roland-Garros 2025 – Monte Carlo results (5000 simulations)

Player	Wins	Runs	Champion %	SF	SF %	QF	QF %
Alcaraz C.	484	5000	*9.68%*	1009	*20.18%*	1411	*28.22%*
Tsitsipas S.	453	5000	*9.06%*	987	*19.74%*	1342	*26.84%*
Sinner J.	413	5000	*8.26%*	1060	*21.20%*	1476	*29.52%*
Djokovic N.	326	5000	*6.52%*	880	*17.60%*	1478	*29.56%*
Zverev A.	286	5000	*5.72%*	813	*16.26%*	1254	*25.08%*
Ruud C.	197	5000	*3.94%*	555	*11.10%*	1054	*21.08%*
Rune H.	197	5000	*3.94%*	680	*13.60%*	1088	*21.76%*
Rublev A.	168	5000	*3.36%*	550	*11.00%*	854	*17.08%*
Musetti L.	113	5000	*2.26%*	470	*9.40%*	783	*15.66%*
Fritz T.	107	5000	*2.14%*	479	*9.58%*	854	*17.08%*

For Roland-Garros 2025, the simulation placed **Carlos Alcaraz** at the top of the list with a **9.68%** win probability, followed closely by **Stefanos Tsitsipas** (**9.06%**) and **Jannik Sinner** (**8.26%**). The model proved to be highly accurate, as Alcaraz ultimately won the tournament after an epic five-set final against Sinner, the longest final in Roland-Garros history. The predicted dominance of these two players closely reflects their actual performance, further confirming the relevance of surface-aware features and pre-match statistics. The

close grouping of win probabilities among top players also illustrates the high level of competitiveness on clay. The model's estimates not only anticipated the real champion but also identified the top-tier contenders with precision, reinforcing the value of machine learning approaches in tennis outcome prediction.

7.3.5 Overall Evaluation of the Monte Carlo Approach

Overall, the Monte Carlo simulations demonstrate that the model internalized key structural biases of the ATP circuit (surface specialization, ELO dynamics, recent form). The alignment with reality validates the robustness of the XGBoost model and the engineered features.

The probabilistic approach also allows us to measure uncertainty, highlighting not only potential winners but also the variability of their performance under different draws.

Monte Carlo simulations offered both predictive strength and computational tractability. Their alignment with real-world outcomes, combined with efficient GPU scaling, makes this approach a powerful tool for sports analytics at scale.

7.4 Conclusion

This chapter marked the culmination of the entire modeling pipeline, where theory met practice. By simulating individual matches, complete tournaments, and thousands of Monte Carlo scenarios, we tested the model's real-world predictive power across multiple scales.

The results speak for themselves. Whether through deterministic bracket simulations or probabilistic tournament runs, the model demonstrated its ability to internalize surface effects, player form, and match dynamics. Not only did it correctly predict both Grand Slam champions of 2025 (Jannik Sinner and Carlos Alcaraz), but it also assigned meaningful win probabilities to other top contenders, closely mirroring the unpredictability and structure of real tournaments.

From a computational standpoint, the parallelized Monte Carlo framework proved highly efficient and scalable. Leveraging SLURM and Finisterrae III's A100 GPUs, we reduced simulation time from hours to minutes without compromising accuracy, a key enabler for future research at larger scales or with extended datasets.

Ultimately, this chapter showcased how machine learning and high-performance computing can come together to simulate the complex, uncertain world of elite sports with both realism and precision. It offered not just predictive answers, but probabilistic insights into what could happen, what might happen, and what often does.

Chapter 8

Limitations and Perspectives

While the results obtained in this project are encouraging, several limitations must be acknowledged, and they naturally open the door to future improvements.

8.1 Player Injuries and Physical Condition

One of the most significant limitations is the absence of explicit information about player injuries or fitness levels. Although some of this information may be indirectly captured by engineered features such as `WINRATE_LAST_5` or other rolling features, the model cannot explicitly distinguish between a temporary dip in form due to injury versus a genuine decline in skill or confidence. Incorporating public injury reports or medical withdrawals could refine the predictive power for high-variance players.

8.2 Gender-Specific Analysis and Model Generalization

This study focused exclusively on the ATP Tour (men's circuit). However, extending the methodology to the WTA (women's tour) would offer a valuable comparative analysis. Women's tennis tends to involve different tactical patterns (e.g., shorter rallies, greater serve-return exchanges), which may influence the importance of different features. Comparing the resulting feature importances and model behavior between ATP and WTA circuits could lead to deeper insights about gender-specific playing styles and prediction challenges.

8.3 Beyond the Numbers: Human Factors

The model operates strictly on quantifiable historical data. However, it overlooks psychological and contextual variables that often influence real match outcomes. Factors such as crowd pressure (e.g., playing at home or facing a hostile audience), emotional momentum, or rivalry history are hard to quantify but play a crucial role in elite sport performance. Including a feature for "home advantage", or proxies like crowd size or tournament country, could help mitigate this gap.

8.4 Environmental Conditions

Environmental variables such as temperature, humidity, altitude, and even the type of tennis balls used (which vary between tournaments) are not taken into account. Yet, these can significantly affect player performance , particularly for those with injury history, stamina issues, or surface-dependent styles. For example, high humidity may reduce serve efficiency and favor baseliners, while altitude (e.g., in Madrid) can amplify serve speed and alter ball bounce.

8.5 The Limits of the Elo Feature

Although ELO_DIFF and its derivatives were among the most predictive features in the model, they rely heavily on historical performance and may underrepresent breakthrough players or one-off surprises. Moreover, Elo ratings update gradually, meaning that players returning from injury or rising rapidly (e.g., young prospects) may be undervalued. A hybrid approach combining Elo with recent short-term spikes (e.g., tournament streaks) could help better capture this dynamism.

8.6 Responsible Use and Betting Caution

With test accuracies above 70%, the model may seem appealing for betting applications. However, it is crucial to emphasize that professional betting markets are highly optimized and reflect real-time information, including insider data (injuries, court conditions, last-minute withdrawals) that this model does not incorporate. Furthermore, bookmakers set odds strategically to ensure profits regardless of match outcomes. Using machine learning models for betting without proper risk assessment can lead to financial losses and overconfidence in probabilistic systems. The aim of this work is to simulate and understand competitive dynamics, not to encourage or guarantee successful betting strategies.

8.7 Other Perspectives and Extensions

Several directions could enhance the current system:

- **Dynamic updating:** Implementing an online learning system to update player features and model weights after each match could improve adaptability and freshness.
- **Ensemble modeling:** Combining the strengths of different classifiers (e.g., blending tree-based models with neural networks) may further improve robustness across different match types.
- **Visualization and interfaces:** Developing interactive dashboards or applications for coaches, journalists, or fans could bring the results of this work to a wider audience.
- **Explainability:** Tools like SHAP or LIME could be used to explain individual predictions in greater detail, increasing trust and transparency in the system.

Overall, this project presents a strong foundation for scalable sports forecasting and offers multiple opportunities for refinement. The limits identified here do not reduce the model's value , rather, they serve as guiding points for deeper exploration, better realism, and more human-aware applications in the future.

Chapter 9

Conclusion

This thesis set out to explore whether machine learning, when combined with high-performance computing (HPC), could effectively predict the outcomes of professional tennis matches. From raw ATP historical data to large-scale tournament simulations, the project covered the full data science pipeline, from preprocessing and feature engineering to model training, evaluation, and deployment in real-world-inspired settings.

The use of advanced supervised learning models, in particular **XGBoost**, demonstrated strong predictive capabilities, achieving over **70% accuracy** on out-of-sample data (season 2024). The success of the model was not limited to individual match prediction: it was validated through full tournament simulations (Australian Open and Roland-Garros 2025), with impressive alignment between predicted outcomes and actual results. In addition, **Monte Carlo simulations** revealed the probabilistic landscape of potential champions, showcasing the model's ability to account for uncertainty and competitive structure.

A key dimension of this work was the **integration of HPC techniques**, particularly through the use of `Dask` and SLURM on the Finisterrae III supercomputer. Distributed hyperparameter tuning and parallelized tournament simulations allowed us to scale efficiently, reducing compute times from hours to minutes while handling large, feature-rich datasets. This aspect not only improved performance but also served as a proof of concept for how HPC can enhance data-driven applications in sports analytics.

Beyond technical performance, this project also highlighted the value of feature engineering in capturing domain-specific knowledge, particularly through surface-aware statistics, rolling winrates, and Elo-based indicators. These features allowed the model to go beyond ATP rankings and understand the nuances that often decide match outcomes in practice.

Nonetheless, several limitations remain. While many psychological or contextual factors, such as confidence, pressure management, or crowd influence, can in theory be partially inferred from data (e.g., recent win streaks, performance under specific conditions, or match location), they are only indirectly captured and not explicitly modeled in this study. Similarly, factors like injury status, fatigue, and environmental conditions (e.g., weather, altitude, court speed) are not included, despite their known impact on performance. These omissions remind us that even with advanced features and high-performing models, tennis, like all sports, retains a degree of unpredictability that goes beyond numbers.

Furthermore, the potential application of such models to betting contexts must be approached with caution.

Even if the model reaches over 70% accuracy, this does not imply systematic profitability. Betting markets are highly optimized, dynamic, and incorporate real-time information that this model does not track. The goal here is to enhance understanding and simulation, not to provide financial guarantees or advice.

In comparison with existing literature, the predictive performance of my XGBoost model (68.96% accuracy on the 2024 test set) is slightly superior to that of Alexander De Seranno [3], who obtained an accuracy of 68.2% using a neural network and 67.4% with logistic regression on a similar task. The results are also comparable to those reported by Chi Pham and Konstantin Bufi [4], whose best model achieved an accuracy close to 69%, though their study does not detail how temporal dependencies or potential data leakage were addressed. Importantly, my methodology strictly prevents data leakage by ensuring that only pre-match features known prior to the match are used for prediction, and by validating the model on entirely unseen tournament data from a later season.

However, my performance remains below the 77.8% accuracy achieved by Nourah Buhamra, Andreas Groll, and Stefan Brunner [5], who modeled Grand Slam matches using various regression-based techniques including spline models and surface-specific player skills. Their approach benefited from richer covariates, notably including betting-related variables such as `B365.1` and `B365.2`, which directly encode bookmaker odds and were shown to be highly predictive. While I chose not to include such variables in my feature set, my work emphasizes generalization and reproducibility without relying on externally priced market information.

Overall, this project offers a robust foundation for future research in predictive sports analytics. Its methodology is generalizable to other circuits (e.g., WTA) and sports, and its HPC scalability opens the door to real-time applications. More broadly, it illustrates how data science and high-performance computing can combine to bring structure, insight, and probabilistic depth to the inherently unpredictable world of elite competition.

All source code, data processing scripts, and simulation experiments are available in my public GitHub repository [6].

Annexes

Appendix A

Glossary of Acronyms

HPC *High Performance Computing*

TFM *Trabajo de Fin de Máster* (Master's Thesis)

ATP *Association of Tennis Professionals*

WTA *Women's Tennis Association*

ML *Machine Learning*

EDA *Exploratory Data Analysis*

CV *Cross-Validation*

SLURM *Simple Linux Utility for Resource Management*

GPU *Graphics Processing Unit*

CPU *Central Processing Unit*

XGBoost *eXtreme Gradient Boosting*

MC *Monte Carlo*

JSON *JavaScript Object Notation*

CSV *Comma-Separated Values*

ROC *Receiver Operating Characteristic*

AUC *Area Under the Curve*

API *Application Programming Interface*

ELO *Elo Rating System* (used for ranking in sports)

EDA *Exploratory Data Analysis*

PARQUET *Apache Parquet* (columnar storage file format)

DASK *Dynamic task scheduling library for parallel computing in Python*

Appendix B

Glossary of Terms

Accuracy Proportion of correct predictions among the total number of predictions made.

Precision Proportion of true positive predictions among all positive predictions made by the model.
$\text{Precision} = \frac{TP}{TP+FP}$

Recall Also known as sensitivity. It is the proportion of true positive predictions among all actual positives.
$\text{Recall} = \frac{TP}{TP+FN}$

F1-score Harmonic mean of precision and recall. It balances the trade-off between these two metrics.
$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$

Log Loss Also called logarithmic loss or cross-entropy loss. Measures the uncertainty of a probabilistic classifier by penalizing incorrect confident predictions. Lower is better.

True Positive (TP) The model correctly predicts a positive class.

True Negative (TN) The model correctly predicts a negative class.

False Positive (FP) The model incorrectly predicts a positive class when the actual class is negative.

False Negative (FN) The model incorrectly predicts a negative class when the actual class is positive.

AUC-ROC Area Under the Receiver Operating Characteristic Curve; evaluates the trade-off between true positive rate and false positive rate across thresholds.

AUC-PR Area Under the Precision-Recall Curve; especially useful for imbalanced datasets to evaluate precision and recall jointly.

Cross-Validation A model evaluation technique where the dataset is split into multiple folds to assess how the model generalizes to unseen data.

Decision Tree A model that predicts by learning decision rules from data features, represented as a tree structure.

Random Forest An ensemble method using multiple decision trees trained on different data subsets, whose predictions are averaged.

XGBoost A gradient boosting framework that builds models iteratively and efficiently, widely used for structured data.

Gradient Boosting A machine learning technique where new models are trained to correct the errors of previous ones, optimizing a loss function.

Hyperparameter Tuning The process of selecting the best configuration parameters that are not learned directly from data (e.g., learning rate, max depth).

Overfitting A model learns the training data too well, including noise, resulting in poor generalization to new data.

Regularization A method used to prevent overfitting by penalizing model complexity.

HPC High Performance Computing; the use of supercomputers and parallel processing for solving complex computational problems.

SLURM A workload manager for HPC systems used to allocate resources and manage parallel jobs.

Speedup Ratio of execution time without parallelism to the execution time with parallelism.
Speedup $= \frac{T_1}{T_p}$

Efficiency Speedup divided by the number of processors or GPUs used. Measures resource utilization.
Efficiency $= \frac{\text{Speedup}}{N}$

Embarrassingly Parallel Describes a problem that can be split into completely independent subproblems with no communication overhead.

GPU Acceleration Use of GPUs to speed up computations, particularly effective for parallel tasks or matrix operations.

Dask A Python library for parallel computing and scalable analytics on clusters or multi-core systems.

Monte Carlo Simulation A method using repeated random sampling to simulate complex systems and estimate outcome distributions.

Probabilistic Simulation A simulation where outcomes are determined by probability distributions rather than deterministic rules.

Random Sampling The selection of a subset of data based on a predefined probability distribution, used in simulation or validation.

Draw In tennis, the process of randomly assigning players to positions in a tournament bracket.

Feature Engineering The process of creating or transforming variables from raw data to improve model learning.

ELO Rating A score representing player skill, updated based on match outcomes. In tennis, it is often adapted by surface and recency.

Rolling Statistics Features computed over a fixed-size moving window (e.g., win rate over last 10 matches) to capture recent form.

Symmetric Dataset A representation where features are defined relative to both players (A vs B), allowing generalizable comparison.

Surface Type of tennis court (clay, hard, grass), which significantly influences match outcomes.

JSON A data format for structured storage and exchange, used for things like tournament brackets or match metadata.

Parquet A columnar storage format optimized for reading subsets of data quickly, widely used in big data pipelines.

Bibliography

[1] J. Sackmann, "Atp match results and statistics," 2024. [Online]. Available: https://github.com/JeffSackmann/tennis_atp

[2] ——, "An introduction to tennis elo," 2019. [Online]. Available: https://www.tennisabstract.com/blog/2019/12/03/an-introduction-to-tennis-elo/

[3] A. D. Seranno, "Predicting tennis match outcomes: An ml approach," Master's thesis, University of Ghent, 2024. [Online]. Available: https://libstore.ugent.be/fulltxt/RUG01/002/945/727/RUG01-002945727_2021_0001_AC.pdf

[4] C. Pham and K. Bufi, "Prediction of atp tennis matches using machine learning," Master's thesis, Lund University, 2023. [Online]. Available: https://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=9121180&fileOId=9121181

[5] N. Buhamra, A. Groll, and S. Brunner, "Modeling and prediction of tennis matches at grand slam tournaments," *Journal of Sports Analytics*, 2024. [Online]. Available: https://doi.org/10.3233/JSA-240670

[6] P. Serin, "Tfm – predicting atp tennis matches with machine learning and hpc," 2025. [Online]. Available: https://github.com/PaulSerin/TFM

[7] CESGA, "Finisterrae iii supercomputer," 2024. [Online]. Available: https://cesga-docs.gitlab.io/ft3-user-guide/index.html

[8] X. Developers, "Xgboost: Scalable and portable gradient boosting," 2025. [Online]. Available: https://xgboost.readthedocs.io/

[9] R.-G. O. Website, "Roland-garros 2025 results," 2025. [Online]. Available: https://www.rolandgarros.com/en-us/results

[10] A. O. O. Website, "Australian open 2025 results," 2025. [Online]. Available: https://ausopen.com/results

[11] J. D. e. a. Hunter, "Matplotlib: Visualization with python," 2025. [Online]. Available: https://matplotlib.org/stable/users/explain/quick_start.html

[12] D. D. Team, "Dask: Library for dynamic task scheduling," 2025. [Online]. Available: https://www.dask.org/

[13] Z. Gao and A. Kowalczyk, "Random forest model identifies serve strength as a key predictor of tennis match outcome," *arXiv preprint arXiv:1910.03203*, 2019. [Online]. Available: https://arxiv.org/abs/1910.03203

[14] C. V. ROOIJ, "Who wins? predicting tennis match outcomes using machine learning," Master's thesis, Tilburg University, 2021. [Online]. Available: https://arno.uvt.nl/show.cgi?fid=158548

[15] J. Sklenička, "Data-driven prediction of tennis matches," Master's thesis, Charles University, Prague, 2024. [Online]. Available: https://dspace.cuni.cz/bitstream/handle/20.500.11956/190596/130386401.pdf?sequence=1&isAllowed=y

[16] P. Gorgi, S. J. Koopman, and R. Lit, "The analysis and forecasting of tennis matches by using a high-dimensional dynamic model," *Statistical Modelling*, 2019. [Online]. Available: https://academic.oup.com/jrsssa/article/182/4/1393/7068322

[17] M. Illum, H. C. B. Mikkelsen, and E. Hovad, "A point-based bayesian hierarchical model to predict the outcome of tennis matches," *Journal of Quantitative Analysis in Sports*, 2019. [Online]. Available: https://martiningram.github.io/papers/bayes_point_based.pdf

[18] A. Goyal and J. S. Simonoff, "Hot racquet or not? an exploration of momentum in grand slam tennis matches," *arXiv preprint arXiv:2009.05830*, 2020. [Online]. Available: https://arxiv.org/abs/2009.05830