PREDICTING TENNIS MATCH OUTCOME: A MACHINE LEARNING APPROACH USING THE SRP-CRISP-DM FRAMEWORK

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY
BY

TOYAN ÜNAL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

DECEMBER 2023

PREDICTING TENNIS MATCH OUTCOME: A MACHINE LEARNING APPROACH USING THE SRP-CRISP-DM FRAMEWORK

submitted by **TOYAN ÜNAL** in partial fulfillment of the requirements for the degree of **Master of Science in Information Systems Department, Middle East Technical University** by,

Prof. Dr. Banu Günel Kılıç
Dean, **Graduate School of Informatics**

Prof. Dr. Altan Koçyiğit
Head of Department, **Information Systems**

Prof. Dr. Sevgi Özkan Yıldırım
Supervisor, **Information Systems, METU**

Examining Committee Members:

Prof. Dr. İbrahim Soner Yıldırım
Computer Education and Instructional Technology, METU

Prof. Dr. Sevgi Özkan Yıldırım
Information Systems, METU

Assist. Prof. Dr. Banu Yüksel Özkaya
Industrial Engineering, Hacettepe University

Date: 07.12.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Toyan Ünal

Signature :

ABSTRACT

PREDICTING TENNIS MATCH OUTCOME: A MACHINE LEARNING APPROACH USING THE SRP-CRISP-DM FRAMEWORK

Ünal, Toyan

M.S., Department of Information Systems

Supervisor: Prof. Dr. Sevgi Özkan Yıldırım

December 2023, 85 pages

Machine learning methods have demonstrated effectiveness in forecasting tennis match results. However, due to their empirical nature, decisions regarding the choice of specific datasets, models, feature sets, or hyperparameters significantly impact outcomes. In this thesis, we employed the Sports Result Prediction Cross-Industry Standard Process for Data Mining experimental framework to address this uncertainty. This approach ensures that results are both replicable and reproducible across diverse datasets and sports types. Our study encompasses 14 years of men's singles tennis match data, from 2009 to 2022, with data from 2021 and 2022 designated as the hold-out test set. We applied six advanced feature extraction techniques, alongside three machine learning models and two feature selection methods. A 10-fold time-based cross-validation approach, coupled with hyperparameter tuning, was adopted. The Extreme Gradient Boosting model, after training and tuning, emerged as the most effective, achieving the lowest Brier score of 0.1913 and an accuracy of 70.5% on the test set. The feature with the highest predictive power was identified as the average win ratios implied by the betting odds of the bookmakers, which played a pivotal role in forecasting match outcomes.

Keywords: Sports analytics, Tennis match outcome prediction, SRP-CRISP-DM, Machine learning, Feature extraction

ÖZ

TENİS MAÇ SONUCU TAHMİNLEME: SRP-CRISP-DM ÇERÇEVESİNİ KULLANAN BİR MAKİNE ÖĞRENMESİ YAKLAŞIMI

Ünal, Toyan

Yüksek Lisans, Bilişim Sistemleri Bölümü

Tez Yöneticisi: Prof. Dr. Sevgi Özkan Yıldırım

Aralık 2023, 85 sayfa

Makine öğrenimi yöntemleri tenis maçı sonuçlarının tahmin edilmesinde etkilidir. Ancak, veri kümelerinin, modellerin, özellik kümelerinin veya hiper parametrelerin seçimine ilişkin kararlar, bu yöntemlerin ampirik doğaları nedeniyle sonuçları önemli ölçüde etkilemektedir. Bu tez çalışmasında, bu belirsizliği gidermek için deneysel SRP-CRISP-DM çerçevesi kullanılmıştır. Bu yaklaşım, sonuçların çeşitli veri kümeleri ve spor türleri genelinde hem tekrarlanabilir hem de tekrarlanabilir olmasını sağlar. Çalışmamız, 2009 ile 2022 yılları arasındaki 14 yıllık tek erkekler tenis maçı verilerini kapsamaktadır. 2021 ve 2022 yıllarına ait veriler, test seti olarak ayrılmıştır. Üç makine öğrenimi modeli ve iki özellik seçme yönteminin yanı sıra altı gelişmiş özellik çıkarımı tekniği uygulanmıştır. Hiper parametre kestirimiyle birlikte 10 katlı zamana dayalı çapraz doğrulama yaklaşımı benimsenmiştir. Aşırı Gradyan Artırma modeli, eğitim ve ayarlamalardan sonra en etkili model olarak ortaya çıkmış olup test setinde 0,1913 ile en düşük Brier skoruna ve %70,5 doğruluğa ulaşmıştır. Bahis şirketlerinin bahis oranlarının ima ettiği ortalama kazanma oranlarının en yüksek tahmin gücüne sahip özellik olduğu belirlenmiştir ve bu özellik maç sonuçlarının tahmin edilmesinde önemli rol oynamıştır.

Anahtar Kelimeler: Spor analitiği, Tenis maç sonucu tahminleme, SRP-CRISP-DM, Makine öğrenmesi, Özellik çıkarımı

To My Beloved Wife

ACKNOWLEDGMENTS

First and foremost, my deepest gratitude goes to my esteemed supervisor, Prof. Dr. Sevgi Özkan Yıldırım. Her invaluable guidance, profound wisdom, and steadfast encouragement have been the cornerstone of this research endeavor.

A special word of thanks is due to my colleagues at the METU Informatics Institute. In particular, I want to acknowledge Alp Bayar, Arif Ozan Kızıldağ, and Burcu Koç for their camaraderie and constructive feedback, which have been invaluable. I am also indebted to the academic and administrative staff of the institute. My sincere appreciation goes to Ayşe Nur Özdere Yuksel, Çetin İnci, Hakan Güler, Nilüfer Şeker, and Sibel Gülnar for their unwavering assistance.

I extend my gratitude to the Scientific and Technological Research Council of Türkiye (TÜBİTAK) for their financial support through the 2210-A MSc scholarship during my studies.

Last but certainly not least, my heartfelt gratitude goes to my family and my wife, whose constant encouragement and unconditional support have sustained me throughout this academic journey. Their love and faith in my abilities have been my greatest strength.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
ATP	Association of Tennis Professionals
BOI	Betting Odds-Implied
BS	Brier Score
CV	Coefficient of Variation
DT	Decision Tree
FS	Forward Selection
GB	Gradient Boosting
IID	Independently and Identically Distributed
ITF	International Tennis Federation
KNN	K-Nearest Neighbors
LR	Logistic Regression
ML	Machine Learning
MOV	Margin of Victory
RF	Random Forest
RPI	Rank Points-Implied
MI	Mutual Information
MSE	Mean Squared Error
SRP-CRISP-DM	Sports Result Prediction Cross-Industry Standard Process for Data Mining
SVM	Support Vector Machine
WTA	Women's Tennis Association
XGB	Extreme Gradient Boosting
χ^2	Chi-square

CHAPTER 1

INTRODUCTION

Tennis holds a significant place in the world of sports, being the fourth most popular sport globally with approximately 1 billion fans [1]. The sport is governed by various organizations: the Association of Tennis Professionals (ATP) oversees men's singles and doubles, the Women's Tennis Association (WTA) is responsible for women's singles and doubles, and the International Tennis Federation (ITF) governs both. Each of these organizations plays a unique role in shaping the sport, from setting rules to organizing tournaments. With a professional circuit spanning numerous countries, coupled with significant media coverage and a large global fanbase, tennis offers a rich and diverse landscape that appeals to fans and professionals, including analysts.

The field of sports analytics has seen growing interest in recent years, fueled by advancements in data collection technology and increased computational power. Tennis, in particular, presents a rich area for data-driven research, as it blends individual performance metrics with a variety of other factors, such as player characteristics and tournament settings.

1.1 Motivation and Problem Definition

Sports analytics offers far-reaching benefits for various actors in the sports industry. For athletes, analytics provide a comprehensive way to improve various aspects of their game, from refining strategies and skills to injury prevention, ultimately contributing to longer, more successful careers. Coaches can leverage data not only to evaluate the efficacy of training methods but also to gain tactical insights and identify high-potential emerging talents. Teams can utilize analytics for smarter recruitment choices and team synergy, thereby maximizing success on the field while optimizing costs. In the betting ecosystem, analytics can refine odds for companies, minimizing their financial risks and maximizing profits while equipping bettors with more capable tools for strategic planning and enhanced risk assessment, increasing their chances of winning. Media outlets can deploy data-driven insights to boost credibility, personalize content, and better target ads, thereby enhancing audience engagement. Tournament organizers can employ analytics to streamline logistics, optimize player matchups for viewer interest, and demonstrate sponsorship value, creating more compelling and efficient events.

In the rapidly growing field of sports analytics, choosing the right sport to analyze can have a big impact on the complexity and accuracy of predictive models. Men's singles tennis stands out as a particularly good starting point for several reasons, especially when compared to team sports and those with subjective scoring systems and rules. Unlike team sports, which require analyzing a wide range of variables and interactions, such as team dynamics and diverse strategies, men's singles tennis focuses

on just two players, simplifying the modeling process while maintaining an analytical challenge. This simplicity is further improved by the sport's structured scoring system; points, sets, and matches are counted in a deterministic manner, leaving little room for subjective interpretation. Lastly, the unpredictability caused by subjective rulings, such as fouls, penalties, or even timekeeping, is not often seen in tennis. Therefore, men's singles tennis offers a balanced yet challenging analytical landscape. Its structured, individualistic nature serves as an excellent steppingstone for analysts and researchers who aim to later tackle more complex predictive models in team sports or those with more subjective rules.

Predicting tennis match outcomes offers significant financial benefits, particularly in sports betting. Moreover, gaining a deeper understanding of the factors and conditions leading to victory can open new horizons for all stakeholders in the world of tennis. There are several approaches for forecasting match results in tennis, including regression-, points-, and paired comparison-based methods [2].

Regression-based methods, recently increasingly categorized under the broader umbrella of Machine Learning (ML) techniques, are trained on a specific number of matches and then tested on a previously unseen test set to confirm their generalizability. These methods often utilize probit or logit estimators, along with more complex ML algorithms [3] [4] [5] [6] [7]. For instance, Klaassen and Magnus [5] proposed a logit regression based on ATP rankings to predict match winners. Del Corral and Prieto-Rodriguez [6] used a probit model to show that recent player performances are the most important factors of match outcomes.

Points-based methods assume each tennis point is Independently and Identically Distributed (IID) [5] and focus on predicting the outcome of a single point. With tennis's hierarchical scoring system—matches consisting of sets, sets of games, and games of points—the probability of winning a game, set, or match can be estimated using probabilities of players winning a point on their serve [8] [9] [10].

Lastly, paired comparison methods estimate winning probabilities based on evaluating the latent abilities of competing players. Examples include the Bradley-Terry model [11], its time-varying version [12], dynamic paired comparisons [13] [14], the Elo rating system [15], and hybrid approaches integrating these models with neural networks [16].

Each of these methods offers unique advantages but also comes with its own set of limitations. ML models are highly effective in analyzing complex patterns within large datasets for predictions. However, they often struggle to incorporate changes in player performance or external conditions that occur in real-time. Point-based methods provide a detailed analysis of the immediate dynamics within a match, focusing on specific game moments. However, they may sometimes overlook broader aspects, such as the psychological factors and momentum shifts that can occur during a game or across a player's career. Dynamic paired comparison models are efficient in adjusting player ratings based on recent match outcomes, reflecting their current form, but they tend to neglect external factors such as the type of court or specific match conditions, which can crucially influence match results.

Recognizing these limitations, our study adopts a comprehensive ML approach, addressing its inherent complexity and need for better alignment to recency. By implementing the Sports Result Prediction CRoss-Industry Standard Process for Data Mining (SRP-CRISP-DM) framework [17], we aim to streamline the complexity and experimental aspects of ML. In addition, we introduce six advanced feature extraction techniques designed to elevate the sensitivity of ML models to real-time player form changes and external match conditions. This approach is designed to overcome the usual constraints

encountered in predicting tennis match outcomes. This is especially relevant given the focus on men's singles tennis, a sport that, as argued, offers an ideal starting point for advancing the field of sports analytics.

1.2 Research Questions

This study aims to answer the following main research questions:

1. What is the effectiveness of the SRP-CRISP-DM framework when applied to the men's singles tennis domain?
2. How do the proposed six distinct feature extraction techniques from sports analytics literature contribute to the predictive modeling for men's singles tennis?
3. Can ML models provide more accurate forecasts than those based solely on players' official rankings or betting odds?
4. Which features are most impactful in predicting the outcomes of tennis matches?

1.3 Contributions of the Study

The contributions of the study are as follows:

- Application of the SRP-CRISP-DM Framework: This research is among the first to apply the SRP-CRISP-DM framework specifically to the men's singles professional tennis domain.
- Methodical Feature Extraction: The study proposes a comprehensive and systematic feature extraction methodology, utilizing six distinct techniques from sports analytics literature.
- Predictive Performance Assessment: This study critically assesses and demonstrates the superior predictive capabilities of ML models over traditional forecasting methods that rely solely on players' official rankings or betting odds.
- Feature Importance Analysis: The research also identifies and ranks the most critical features for predicting tennis match outcomes, providing valuable insights for further research and practical applications.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows: Chapter 2 provides background information, reviews prior studies, and explains how the thesis fits the literature. Chapter 3 presents the selected data sources, data collection and integration methods, and analysis techniques and provides the results obtained. Chapter 4 summarizes the experiments conducted and provides results. Chapter 5 discusses the results of the study, along with its limitations and future works.

CHAPTER 2

BACKGROUND AND RELATED WORK

In this chapter, an overview of foundational concepts and related studies in the domain of tennis analytics and predictive modeling is provided. To assemble this comprehensive review, an initial keyword-based search of articles, dissertations, and other scholarly works was undertaken. Following this, Scite [18], an AI-powered tool, was employed to conduct advanced semantic and network-based searches to pinpoint papers directly relevant to the subject of this thesis. The studies and frameworks uncovered during this meticulous search process served as both a foundation and a point of comparison for the current research. The background and related work section can be categorized into four main areas: tennis as a game, tennis datasets, the SRP-CRISP-DM framework, and predictive models. Each category is integral to understanding the complexities of predicting tennis match outcomes and contributes to the broader landscape of sports analytics. These categories, along with their subcategories, will be discussed in detail in the following sections.

2.1 Tennis as a Game

Tennis is a racquet sport that can be played in various formats, most commonly as singles or doubles. For the purposes of this study, we will focus exclusively on men's singles tennis. In a standard singles match, one player serves while the other receives, with roles alternating as the match progresses. Players stand on opposite ends of a rectangular court, separated by a net. The gameplay follows a specific set of rules. A player has two chances to serve the ball legally into the opponent's court. Faults occur if the serve either hits the net or goes out of bounds. Once served, the ball must land within the opponent's court and can only bounce once before being returned. Players are also restricted from touching the net. Points are awarded based on successful plays, with a game requiring a player to earn at least four points (scored as 15-30-40-Game) and be two points ahead to win. A set is won by the first player to reach six games while being at least two games ahead of the opponent. In the event of a 6-6 tie in games, most tournaments employ a tiebreaker. Finally, a match is won by securing the majority of sets, which can be either best of 3 or best of 5, depending on the tournament.

Men's professional tennis is largely overseen by the ATP, which was formed in 1972 to protect the interests of professional tennis players and is responsible for organizing a range of tournaments. The type of tournament—whether it is a Grand Slam, ATP Finals, or ATP Masters—can greatly influence players' strategies and performance. Each of these tournament types comes with its own set of rules, levels of prestige, and implications for player rankings, making them unique arenas for competition.

Additionally, the type of court surface has a significant impact on the pace and style of play. Clay courts, such as those used in the French Open, are known for their slower ball speed and higher bounce, requiring players to adopt a more strategic, endurance-based game. Grass courts, like those at Wimbledon, offer a faster, lower bounce, favoring players with strong serve-and-volley skills. Hard courts, used in tournaments like the US Open and the Australian Open, provide a consistent, medium-paced bounce and are less demanding on player movement compared to clay courts.

2.2 Tennis Datasets

The advancement of predictive modeling in tennis is fundamentally linked to the availability and quality of datasets. This section aims to provide a brief overview of the most commonly used datasets and online data sources in tennis analytics.

The ATP Tour website (www.atptour.com) serves as the official source of information for the men's professional tennis circuit. It offers live scores, detailed match statistics, current player rankings, head-to-head records between players, information on individual players, and details about various tournaments.

OnCourt (www.oncourt.info) is a paid software package that offers extensive data on over 1.6 million tennis matches for both men's and women's tennis since 1990. The program provides a range of information from player statistics, tournament results, and head-to-head comparisons to match predictions and bookmaker odds.

Flashscore (www.flashscore.com) is a sports statistics website offering real-time scores and results across a variety of sports, including tennis. The platform provides detailed statistics—including point-by-point data—on individual tennis matches, current player standings, head-to-head statistics, and event-specific data.

Tennis-Data.co.uk (www.tennis-data.co.uk/alldata.php) is a portal specialized in tennis betting and historical match results. It offers weekly updated data in CSV and Excel formats for both the men's ATP and women's WTA tours. The site provides match results and statuses as well as ATP rank points and rankings of players from the year 2000 onward and closing betting odds of different professional bookmakers from the year 2001 onward.

Jeff Sackmann's tennis_atp repository (github.com/jeffsackmann/tennis_atp) is a public GitHub repository offering an extensive collection of ATP tennis data. The repository features data on individual ATP matches dating from 1968 to present, doubles matches from 2000 to 2020, futures matches from 1991 to present, and qualifier and challenger matches from 1978 to present. It also includes player-specific columns such as player ID, name, hand preference, birth date, country code, and height, as well as ATP ranking dating back to the 1970s. This repository is updated weekly.

In the course of conducting research for this thesis, 46 academic works—including research papers, theses, and dissertations—explicitly identified the datasets they utilized for predicting tennis match outcomes. Among these, the ATP Tour's official website was the most frequently cited, being referenced 17 times. It was closely followed by Jeff Sackmann's repository, cited 16 times, and the Tennis-Data website, cited 13 times. The OnCourt software package and the Flashscore website were each mentioned in three studies.

For the purposes of this study, two specific data sources were chosen: *atp_matches* dataset from the Jeff Sackmann's *tennis_atp* repository and the *ATP Men's Tour* dataset provided by Tennis-Data.co.uk. The rationale for this selection is multifaceted. First, Jeff Sackmann's repository includes ATP rankings, which are widely recognized as a robust predictor and a valuable baseline for tennis match outcome prediction [19] [20] [21]. Second, Tennis-Data.co.uk allows for the calculation of the consensus among bookmakers, which is another highly influential predictor [2] [20] [21]. Beyond their predictive capabilities, both datasets are freely accessible, regularly updated, and held in high esteem within the academic community. Utilizing these datasets not only enhances the study's reliability but also allows for meaningful comparisons with previous research, thereby enriching the academic discourse.

2.3 The SRP-CRISP-DM Framework

The original Cross-Industry Standard Process for Data Mining (CRISP-DM) framework was introduced by Wirth and Hipp [22] in a conference paper. While it does not specifically address the CRISP-DM framework, it provides a foundational reference for the CRISP-DM methodology. The paper outlines the six steps of CRISP-DM and discusses its application in the field of data mining.

The paper by Bunker and Thabtah [17] introduces the Sports Result Prediction CRISP-DM framework, which extends the original CRISP-DM framework for the specific issues faced when forecasting match results in sports. It provides a comprehensive overview of the framework and its six steps, along with a discussion of its application in sports analytics. The paper also includes a case study to demonstrate the practical implementation of the SRP-CRISP-DM framework in the context of match result prediction.

2.4 Predictive Models

The primary objectives of a forecasting model designed to predict the winning probabilities of players prior to a match are to surpass existing methods both statistically and economically.

The most widely used methods for predicting tennis match outcomes can be broadly categorized into three groups: mathematical models, as described by Newton and Keller [23]; paired comparison models, explored in depth by McHale and Morton [11]; and ML models that incorporate player rankings and other predictors, initially referred to as regression models by Boulier and Stekler [3] but now commonly recognized under the broader umbrella of ML [24].

2.4.1 Mathematical Models

Point-based models rely on mathematical assumptions and calculations. Consequently, they are examined under a broader category which is Mathematical Models. These models focus on estimating the likelihood of winning individual points, which then informs overall match predictions. Under an IID assumption for point outcomes, Klaassen and Magnus [5] demonstrate this approach, detailing how match-winning probabilities can be deduced from serving and returning statistics. Newton and Keller [23] further elaborated on this method, providing formulas to predict match outcomes based on a player's point-winning probabilities, accounting for factors like tiebreakers. Their work, however, does not offer a specific model for estimating serve and return probabilities, suggesting only to adjust

for the opponent's skill level. Building on this, Barnett and Clarke [8] introduced a method adjusting tournament averages for player serve advantage and opponent return advantage, though its broader performance was not assessed. Spanias and Knottenbelt [25] developed a state model for serve-win probabilities, combining various on-serve events. Their model averages event probabilities over recent play, with adjustments for opponent strength, mirroring Barnett and Clarke's approach. They found that using data from the latest 12 months yielded better predictions for the 2011 ATP season compared to using 6 or 18 months of data. Knottenbelt et al. [9] proposed a unique angle, employing a *common opponent model* to balance serve and return win probabilities. This model averages performance against shared opponents to mitigate bias from varying opponent quality. They found that both their model and a modified version of Barnett and Clarke's method produced profitable betting returns for the 2011 Grand Slams, especially when stratified by playing surface. Ingram [26] made a case for point-based models by using a Bayesian hierarchical approach for match prediction. Taking surface, tournament, and match date into account, he reported results that are comparable to those of the other model classes.

2.4.2 Paired Comparison Models

Paired comparison methods analyze past matches among players to deduce their strength rankings and forecast future game outcomes. McHale and Morton [11] propose a paired comparison probability model, fine-tuned using players' historical performance data and the playing surface. Their findings suggest that this approach outperforms logistic regression models in both prediction accuracy and potential betting gains. Conversely, Lyocsa and Vyrost [27] applied a similar model to examine various betting strategies based on odds and player rankings. Still, they found only marginal evidence of profitability, challenging McHale and Morton's conclusions about market inefficiencies. Gorgi et al. [12] introduced a dynamic model that adapts to players' changing skill levels on different surfaces, claiming it surpasses models relying solely on rankings.

The Elo rating system, which was originally developed for chess player ratings [15], is a special case of paired comparison approaches. It has been commonly used to estimate ratings and predict outcomes in different sports, such as football [28] [29], rugby [30], and tennis [2] [10]. Williams et al. [20] extended this to a surface-adjusted Elo rating, considering varied tennis court surfaces like grass, hard, and clay. Kovalchik [24] further evolved the Elo system to factor in the Margin of Victory (MOV) in tennis, experimenting with four MOV integration methods: linear, joint additive, multiplicative, and logistic regression. Among these, the joint additive model emerged as the most reliable, showing consistent variance and minimal bias in a simulation of player ratings. Highlighting a limitation of the standard Elo in reflecting a player's recent form, Angelini et al. [31] proposed a weighted Elo approach, which gives more importance to the player's latest match score to capture their current momentum.

2.4.3 Machine Learning Models

ML models, trained on a set number of matches and validated on unseen test sets for generalizability, are primarily divided into regression and classification types. The former, as illustrated by Kovalchik [24], might predict numeric outcomes like the MOV in sets, while the latter focuses on discrete outcomes such as win/loss, as we explore in our study.

Early examples of these models, like those by Clarke and Dyte [4] and Klaassen and Magnus [5], directly model match outcome probabilities. Clarke and Dyte [4] used data from the men's Grand Slam tournaments and fitted a logistic regression model to official ATP rankings to estimate a player's chance of winning as a function of the difference in rating points. Klaassen and Magnus [5] proposed a logistic regression-based method applied at the match and point-level data, where the estimated win probabilities are again functions of the difference in player ranking, to forecast the winner of a tennis match at the beginning of and during the Wimbledon Grand Slam matches. Among the more comprehensive studies, Del Corral and Prieto-Rodriguez [6] employed probit models using past performance, player features, and match characteristics, identifying rankings as key for prediction accuracy. For women, being a former top-ten player was significant, and age differences had a significant impact for both genders, albeit with different patterns. Lisi [32] used logistic regression with ATP points/rankings, player ages, home advantage, and bookmaker odds to predict 501 matches, reporting about 16% returns using a specific betting strategy. Gu and Saaty [33] combined data with expert judgments using an analytical network process, achieving 85.1% accuracy, albeit on a very small dataset of 63 ATP men's and 31 WTA women's matches played in the 2015 US Open.

Ghosh et al. [34] examined different ML classifiers, including Decision Tree (DT) and Support Vector Machine (SVM) for singles tennis match predictions, with DT achieving the highest accuracy, which is 99.14% for a 70%-30% train-test split and 98.45% for 10-fold cross-validation. However, their use of cross-validation might have inflated accuracy by incorporating future match data, which may partly explain the overly high accuracy reported. Candila and Palazzo [16] applied an Artificial Neural Network (ANN) to various features, outperforming four out of five models in out-of-sample predictions. Wilkens [21] applied a range of ML models to professional tennis matches, finding that average prediction accuracy levels off at about 70%. Wilkens [21] also found that regardless of the model applied, most relevant information was contained in betting odds, and adding other match- or player-specific information did not improve results. The set of models used was: SVM [35], ANN, Logistic Regression (LR), Random Forest (RF) [36], and Gradient Boosting (GB). The author also included baselines based on the betting odds-implied probabilities. Grinsztajn et al. [37] showed that tree-based ML models, such as RF and eXtreme Gradient Boosting (XGB), remain state-of-the-art on medium-sized tabular datasets ($\sim$10K samples) even without accounting for their superior speed. Their contribution includes comparing extensive benchmarks of deep learning methods as well as tree-based models across a large number of datasets and hyperparameter combinations.

The use of ML techniques is more of a novel area in sports prediction. While ML in sports prediction is relatively new, especially in tennis, studies report prediction accuracies around 65-70% (allegedly up to 99% in some cases), generally agreeing that models do not usually outperform bookmaker odds. These studies, particularly in betting analysis, often use data from no more than a year.

CHAPTER 3

METHODOLOGY

Machine learning is an empirical process, and it is difficult to predict how choosing a specific dataset, model, feature set, or hyperparameters will impact the results. The best approach is to test these elements, evaluate the outcome, and then decide on how to proceed. In this uncertain landscape, the need for a structured experimental approach becomes evident. The primary objective of employing a structured experimental methodology in sports prediction is to achieve results that are both replicable and reproducible across diverse datasets and types of sports.

The methodology of this thesis is primarily structured around the SRP-CRISP-DM framework, a standardized six-step approach employed for making predictions in sports analytics [17]. The framework was originally developed for predictive modeling in team sports and has mainly seen applications in that specific area. Our study extends its application to individual sports, with a specific focus on men's singles tennis, demonstrating its versatility and potential for broader application in sports analytics.

Each phase of the SRP-CRISP-DM framework [17], as illustrated in Figure 1, including Domain Understanding, Data Understanding, Data Preparation and Feature Extraction, Modeling, Model Evaluation, and Model Deployment, is elaborated in the subsequent sections, providing a comprehensive understanding of the tools, techniques, and procedures employed in this research. This structured approach ensures replicable and reproducible results across diverse datasets and types of sports.

To further support the reproducibility of this study, all code and scripts used in the data collection, data preprocessing, model training, and evaluation are made publicly available. The code is written in Python and utilizes libraries such as Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, SciPy, XGBoost, and MLxtend for the various stages of the study. The code repository also includes Jupyter Notebooks that walk through the data exploration and modeling steps and Excel spreadsheets with intermediatory outputs of these steps. The entire codebase is hosted on GitHub and can be accessed at https://github.com/toyanunal/Tennis-Outcome-Prediction-Thesis/. Interested readers are encouraged to download and explore the code to gain a deeper understanding of the methodologies employed in this research.

In this thesis, the Methodology section outlines the general procedures and presents results that are replicable and reproducible without going into the specifics of experimental outcomes and iterative processes. Conversely, the Experiments section delves into empirical aspects, including experimentation processes, extensive testing, iterative refinement, and detailed empirical results associated with the Feature Extraction, Modeling, and Model Evaluation phases of the SRP-CRIPS-DM framework (see Chapter 4 for details). This structure was chosen to prevent redundancy and enhance clarity. It circumvents the need for excessive technical reiterations and avoids constant cross-referencing between

the Methodology and Experiments sections, which could otherwise complicate the understanding of technically intensive content.

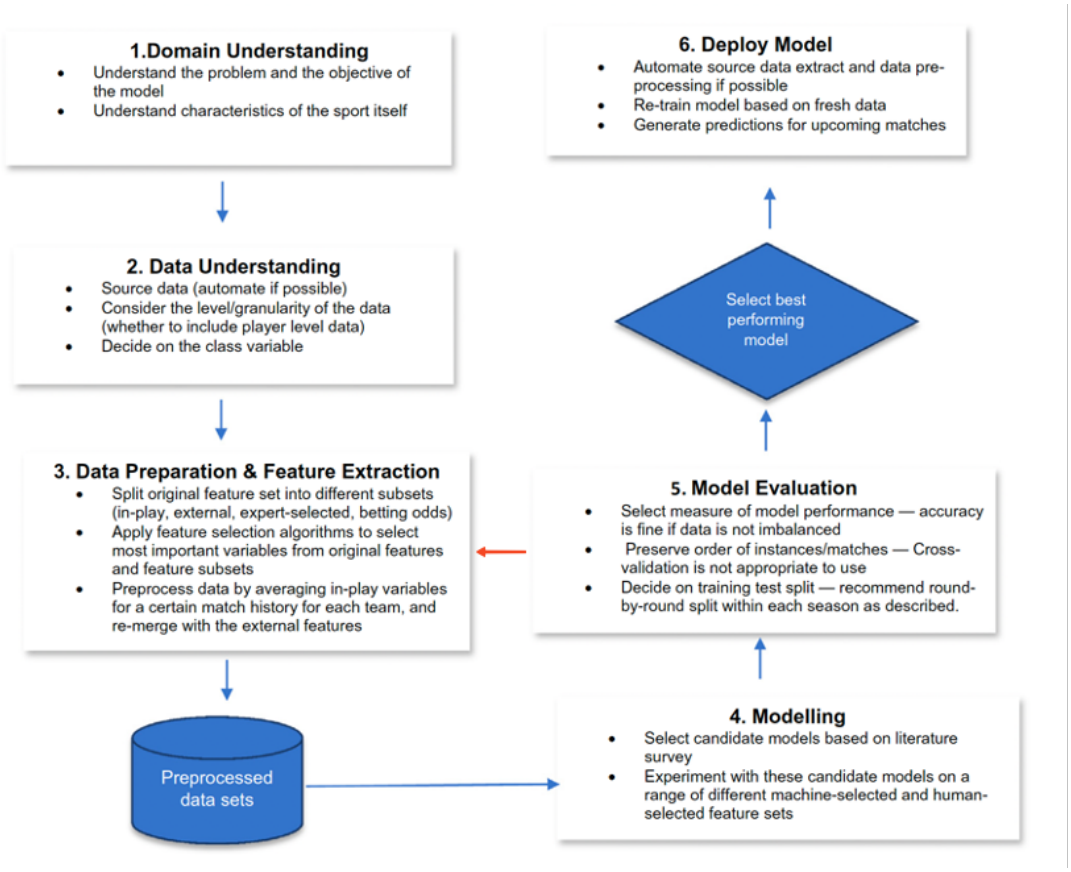

Figure 1: The six consecutive phases of the SRP-CRISP-DM framework. The red arrow, which is not present in the original paper by Bunker and Thabtah [17], has been added to highlight the iterative nature of the model-building process, emphasizing the necessity of revisiting the model development stage upon the evaluation of performance metrics.

3.1 Domain Understanding

The first step of this framework entails a thorough understanding of the problem, the goal of the modeling, and the unique characteristics of the sport itself, such as the rules, the potential factors affecting match outcomes, and player characteristics [17].

Men's singles tennis is an individual sport, making each match's outcome dependent solely on the participating players' performance. Players bring a unique blend of skills and styles to the court, with some excelling in serves and others in returns. Tennis matches can last several hours, requiring not only physical endurance but also mental resilience from the players.

The structural elements of the sport also add layers of complexity to any predictive model. A match is segmented into sets and games. To win a match, a player typically needs to win two out of three

sets—or three out of five in some premier tournaments like the Grand Slams—to claim the match. The serve, which alternates between players after each game, is another key element; it often provides a slight advantage to the server, thereby influencing match outcomes. Additionally, the type of court surface—grass, clay, or hard court—can alter the ball's speed and bounce, thereby favoring different styles of play.

The comprehensive goal of this study is to develop an effective predictive model for men's singles tennis and to do so, all these tennis-specific factors should be taken into account.

3.2 Data Understanding

The Data Understanding section is organized into three key objectives, each crucial for achieving a clear understanding of the dataset and its potential for predictive modeling. These objectives include Dataset Assembly, Data Granularity Considerations, and Class Variable Selection, as guided by the SRP-CRISP-DM framework [17].

3.2.1 Dataset Assembly

The first aim of the Data Understanding phase is to assemble a comprehensive dataset of historical data on the relevant sporting event. This dataset should include a range of variables that may influence the outcome, from player performance metrics to match conditions and statistics. To achieve a dataset with high predictive potential, we combined two reputable datasets that were previously described in Section 2.2: Tennis-Data.co.uk's "ATP Men's Tour" dataset and Jeff Sackmann's "atp_matches" dataset from his *tennis_atp* repository. Throughout this thesis, we will refer to these datasets as "UK-Data" and "ATP-Data", respectively. To ensure these source datasets are well-integrated and prepared for the consequent stages of our analytical framework, we carried out the following mandatory steps:

1. Since both datasets are structured on a year-by-year basis, we combined both datasets into two unified structures. This allows for a more streamlined time-series analysis and offers a more organized approach for future data manipulation.

2. We decided to exclude match data ranging from the years 1968 to 2008. This was due to significant changes in technology, play style, betting companies, and tournament formats that could introduce confounding variables into our analysis. Additionally, data for the year 2023 was omitted as it was not fully available at the time this thesis was written, preventing a complete annual analysis.

3. The `Comment` field in UK-Data provides match status, categorizing matches as completed (i.e., "Sched"), unfinished (i.e., "Retired", "Disqualified", or "Awarded"), or not played (i.e., "Walkover"). Based on these categories, we chose to eliminate all non-completed matches from the datasets. The rationale behind this decision is twofold: Firstly, these matches often involve unique circumstances like walkovers or disqualifications that deviate from standard match progression, potentially introducing noise into the data. Secondly, as these matches lack a conventional conclusion, they do not contribute to the primary target variable of our ML models, which is a definitive match outcome.

4. While both UK-Data and ATP-Data offer records for men's and women's tennis matches, we decided to focus exclusively on men's matches. This decision was based on the predictive models developed for men's and women's tennis in the literature. For instance, in men's tennis, variables such as player ranking, serve statistics, and specific performance indicators are often pivotal for predicting match outcomes [5] [38]. Additionally, the concept of "home advantage" has been shown to impact male players significantly but not their female counterparts [39]. Conversely, in women's tennis, factors like player consistency and pressure-handling abilities, as well as being a former top-ten player, have been deemed more critical [6]. Focusing solely on men's tennis allows us to utilize a more homogeneous dataset, which is advantageous for our ML models as it reduces variability and increases the reliability of our predictions. This narrow focus also permits a more in-depth, specialized analysis of similar game dynamics, such as the level of competition, playing conditions, and tournament structures.

5. Doubles matches were also deliberately removed from our dataset. Similar to the gender focus, this decision was made to ensure that the data remains as homogeneous as possible, allowing us to concentrate on similar game dynamics.

6. UK-Data contains the matches of only top-tier tournaments, namely Grand Slams, ATP Finals, ATP Masters 1000, ATP 500, and ATP 250. In contrast, ATP-Data offers a more diverse range of match records, extending beyond top-tier tournaments to include the Olympics, ATP Cup, Davis Cup, ATP Challenger Tour (e.g., Challenger 175, 125, 100, 75, 50), and ITF Men's World Tennis Tour (e.g., M25 and M15). We excluded tournaments from ATP-Data that are not present in UK-Data so that we could integrate these two datasets. This decision serves two main purposes: First, it narrows our analytical focus to top-tier competitions and the world's top 2000 players, ensuring more consistent playing conditions and the involvement of high-caliber athletes. Second, it enhances dataset homogeneity, reducing possible unrelated variables that could introduce noise into our predictive models.

7. Upon applying these filters, both datasets were distilled down to exactly $34,121$ matches. While merging these two datasets, we faced a unique challenge: the `Date` field in UK-Data specifies the exact match date, whereas ATP-Data's `tourney_date` only notes the Monday of the tournament week. Given that each top-tier tennis tournament occurs annually, we used the year and tournament name as identifiers for each unique tournament. Additionally, the names of the competing players were essential for correctly pairing each match across the datasets. However, these identifiers were not sufficient on their own, especially in cases of repeated matchups within the same tournament due to the round-robin stage in the ATP Finals or the introduction of "lucky losers" to replace injured or withdrawn players. To resolve this, we determined that a unique composite key for accurate match identification would require five distinct pieces of information: 1) Year, 2) Tournament name, 3) Round, 4) Winner name, and 5) Loser name.

8. At this point, our aim was to merge every single match from both datasets, if possible. The process began by extracting the years from match dates and creating a `year` field in both datasets.

9. Unlike the name of the field suggests, instead of the `tournament` field in UK-Data, `location` field better matches with the tournaments in the `tourney_name` field of ATP-Data. However, there were numerous inconsistencies between these two fields of the datasets, some of which were that one dataset used the same name such as "Adelaide", while the other used "Adelaide", "Adelaide 1", and "Adelaide 2" to separate different years of the tournament; one dataset used different names "Oeiras" and "Estoril" while the other used "Estoril" for both; one dataset used

"Nur-Sultan" as the tournament name when referring the capital city of Kazakhstan while the other used "Astana". Eventually, a dictionary that could differentiate unique tournaments in each dataset and successfully match them was formed.

10. The `round` fields in two datasets also have different naming conventions. UK-Data named rounds based on the order of the round (i.e., 1st round, 2nd round, 3rd round, and 4th round), while ATP-Data named rounds based on the number of contestants in that stage (i.e., R128, R64, R32, and R16). We converted UK-Data's round convention into ATP-Data's for consistency, which also provided more informative data regarding the specific round's number of contestants.

11. The two datasets employ different naming conventions for players. Specifically, UK-Data adopts a format of the last name followed by the initial letter of the first name, whereas ATP-Data includes the full first and last names of players. Simple text processing tools were utilized to align these naming conventions. Manual checks were conducted for cases with issues like hyphen usage, multiple last names, or spelling variations across (and within) data sources. Through this process, a dictionary was constructed to successfully match 946 unique male tennis players' names across both datasets.

12. In the final step, clerical errors within the datasets were addressed. The strategy was straightforward: if both datasets agreed on a piece of information, it was deemed accurate; if there was a discrepancy, web-based resources were utilized to verify and correct the information. This process helped in identifying and correcting winner-loser mix-ups in three matches and round misclassifications in two matches. Additional adjustments were made to ensure a smooth merging process, resulting in a unified and coherent dataset ready for further analytical exploration.

Consequently, our merged dataset comprises $34,121$ men's professional singles tennis matches played between 2009 and 2022. It encapsulates the following yearly tournaments (with slight changes over the years): 4 Grand Slams, 1 ATP Finals, 9 ATP Masters 1000s, 13 ATP 500s, and 39 ATP 250s.

3.2.2 Data Granularity Considerations

The second aim in the Data understanding phase, as outlined in the SRP-CRISP-DM framework, is to consider the granularity of the data. Granularity refers to the level of detail the data provides, which, in this case, revolves around whether to include both player-level and match-level data.

Given that our research is focused on men's singles tennis, a type of individual sport, the data at both the match level and player level are closely related and often used interchangeably. Each match in tennis is a standalone event but also tied to the individual players involved in it. Therefore, the performance metrics and characteristics of the players are closely linked with the outcomes and statistics of the matches, making player-level and match-level data almost synonymous in the context of our analysis.

When examining the granularity, a key consideration was whether to split or combine certain metrics. For example, a single match contains a variety of data points ranging from the overall match outcomes to more detailed player statistics like serve performance (including aces, double faults, and serve points) and breakpoint statistics (including break points saved and faced). These player-specific metrics, even when looked at the match level, provide a detailed view of the match dynamics and, by extension, the tournament and player career progressions.

Furthermore, the granularity of data also involves looking at the time and place aspects of the sport, covering the sequential order of matches, tournaments, and seasons, as well as the differences in surface, court type, and tournament settings across various tournament locations.

By carefully looking at and adjusting the granularity of the data, a strong basis is created for the upcoming analytical and predictive modeling tasks. The blending of player-level and match-level data provides a rich set of information ready to support insightful analyses and improve the predictive capability of the ML models to be used in later stages of this research.

3.2.3 Class Variable Selection

The third objective in the Data Understanding phase is choosing the class (target) variable. This step is pivotal as it lays the foundation for the modeling tasks ahead. Most historical efforts in tennis outcome prediction have treated the problem as a binary classification task, typically distinguishing between a win for either of the two competing players. However, some studies like that of Kovalchik [24] and Angelini et al. [31] explored a numeric prediction approach, employing regression techniques to estimate the MOV, which then aided in making a win-loss prediction.

In our specific case, the way our combined dataset is set up makes it suitable for a binary classification task. Unlike conventional representations where match-related information is allocated to "player 1" and "player 2" with a binary `result` column indicating the winner, our dataset utilizes `winner` and `loser` columns to depict match outcomes. For a more coherent representation, we relabeled the data, assigning "player 1" and "player 2" to either the winner or loser randomly for each match while also indicating whether player 1 emerged victorious. This label, `p1_won`, is what our model aims to predict. This re-labeling was done once during data pre-processing and kept the same throughout the study.

We could also look at the problem as a numeric prediction, using ML techniques to initially predict the MOV based on points, games, and sets and then make a win-loss prediction based on the predicted margin. This method is especially viable in betting situations, as accurately forecasting the expected return on investment can be more important than just predicting the winner of a match.

Research conducted by Delen et al. [40], Valero [41], and Elfrink [42] provide evidence that models based on classification are more accurate than those based on regression for predicting match outcomes. Classification models' ability to capture uncertainty, handle non-linear relationships, and consider sport-specific dynamics has been reported to contribute to their superior performance.

In this study, our rigorous experimentation aligned with the literature's findings, showing that treating the problem as a classification yielded superior results for our dataset and the men's singles tennis domain. However, this finding is not a one-size-fits-all; the effectiveness of classification over regression may vary across different sports, datasets, or use cases.

Lastly, it is worth noting that in our transformed dataset, the target variable had a nearly 50:50 split, having $17,107$ (50.1%) matches won by player 1 and $17,014$ (49.9%) matches won by player 2. This even split in class distribution is beneficial for training robust ML models, as it mitigates the risk of bias towards a particular outcome, thus promoting a more accurate and generalized predictive performance.

3.3 Data Preparation and Feature Extraction

The Data Preparation and Feature Extraction section is structured around four key objectives, each serving a vital role in readying the dataset for efficient and accurate predictive modeling. In accordance with the SRP-CRISP-DM framework, these objectives involve categorizing features into meaningful subsets for better interpretability and feature selection, handling outlier and missing data to enhance the model's robustness, extracting new features to capture complex relationships in the data, and preparing the dataset through validation checks, train-test splitting, and feature encoding and scaling to make it compatible with various ML models [17].

3.3.1 Feature Subsets of the Dataset

The first objective in the Data Preparation and Feature Extraction phase is to categorize features into meaningful groups. Our dataset can be divided into four distinct groups: Match Characteristics, Player Characteristics, Betting Odds, and In-play Statistics. Table 1 provides a summary of the most relevant data available in our combined dataset. It is worth noting that, for simplicity in the table, we represented any feature that exists for both player 1 and player 2 from the perspective of player 1, using the prefix "p1". Any feature with the "p1" prefix in its naming is also available for player 2.

Table 1: Features of the Combined Dataset

Match Characteristics	**Description**
tourney_id	A unique identifier for each tournament (e.g., "2022-9410", where the first four characters represent the year, and the remaining characters are tournament-specific)
match_num	A match-specific identifier, starting from "1" and counting up for each tournament
match_date	Date of the match (e.g., "26-03-2022")
tourney_name	Name of the tournament (e.g., "US Open")
tourney_level	Level abbreviation for different ATP tennis series (i.e., "G" for Grand Slams, "F" for ATP Finals, "M" for ATP Masters 1000, and "A" for ATP 250 and 500)
court	Type of court (i.e., "outdoor" where players have to consider the effects of court surface and environmental conditions, or "indoor" where conditions are more controlled)
surface	Type of surface (i.e., "clay", "hard", or "grass")
draw_size	Number of players in the draw, often rounded up to the nearest power of 2 (e.g., a tournament with 56 players has a 'draw_size' of "64")
round	Round of the match (i.e., "RR" for a round-robin stage where each player plays against all the other players in their group; "R128", "R64", "R32", and "R16" for an elimination round with at most 128, 64, 32, 16 players, respectively; "QF" for quarterfinals, "SF" for semifinals, and "F" for the final)

Continued on next page

Table 1 – continued from previous page

Match Characteristics	Description
best_of	Maximum number of sets playable in a match (i.e., "3" or "5")
Player Characteristics	
p1_id	Male tennis player identifier of player 1 (e.g., "104925" for Novak Djokovic)
p1_name	First and last name of player 1
p1_rank	ATP ranking of player 1 as of the tournament date
p1_rankpt	ATP rank points of player 1 as of the tournament date
p1_hand	Dominant hand of player 1 (i.e., "L" for left, "R" for right; serving hand for ambidextrous players)
p1_ht	Height of player 1
p1_ioc	Three-character country code of player 1
p1_age	Age, in years, of player 1 as of the tournament date
p1_seed	Player 1's seeding (if present), which is used to separate top players in a draw so that they do not meet in the early rounds of a tournament
p1_entry	Player 1's entry status (if present), indicating why he was admitted into a tournament; "WC" for Wild Card, "Q" for Qualifier, "LL" for Lucky Loser, "PR" for Protected Ranking, "ITF" for ITF Entry
Betting Odds	
p1_B365	Closing betting odds of Bet365 in favor of player 1
p1_EX	Closing betting odds of Expekt in favor of player 1
p1_LB	Closing betting odds of Ladbrokes in favor of player 1
p1_PS	Closing betting odds of Pinnacle in favor of player 1
p1_SJ	Closing betting odds of Stan James in favor of player 1
p1_UB	Closing betting odds of Unibet in favor of player 1
In-play Statistics	
minutes	Match duration, in minutes
score	Score breakdown of the match in terms of games (including tiebreak games) and sets (e.g., "7-6(4) 6-4")
p1_ace	Player 1's number of aces, a legal serve that is not touched by the receiver
p1_df	Player 1's number of doubles faults, occurring when both serve attempts in a point fail to land in the service box
p1_svpt	Player 1's number of serve points, a legal serve that is not returned to the server's court by the receiver
p1_1stIn	Number of first serves that landed in the service box by player 1
p1_1stWon	Number of points won by player 1 on the first serve
p1_2ndWon	Number of points won by player 1 on the second serve, only possible when there has already been one fault on the point
p1_SvGms	Number of serve games played by player 1
p1_bpSaved	Number of break points defended successfully by player 1
p1_bpFaced	Number of break points faced by player 1

Each category of features encapsulates unique aspects of the dataset. Match Characteristics encompass details related to the tournament and the match itself, including but not limited to the date of the match, level of the tournament, surface, and the round in which the match occurs. Player Characteristics provide information about both competing players, such as their ATP ranking, dominant hand, height, country of origin, and age at the time of the tournament. Betting Odds feature the closing betting odds from six distinguished bookmakers, providing insight into the betting landscape surrounding each match. In-play Statistics capture the dynamics of the match, including its duration, score, and player-specific statistics like the number of aces, double faults, and service points, among others.

It is important to clarify that the features in the subsets—Match Characteristics, Player Characteristics, and Betting Odds—are known prior to each upcoming match. This includes information such as the match's location, court type, and surface, along with player-specific details like age, height, and dominant hand, as well as the closing betting odds. In-play Statistics, however, are only determined after the match has concluded. Thus, we only know an average of these features for a certain number of past matches for the players participating in the upcoming match.

3.3.2 Outlier and Missing Data Handling

Handling outlier and missing data is another aim of the Data Preparation and Feature Extraction phase. Statistical analyses with outlier and missing data are important topics in statistics. It involves developing methods for analyzing data sets with outlier and missing values. The goal is to make valid inferences and draw accurate conclusions despite the presence of outlier and missing data [43].

In this study, we employed a comprehensive strategy for outlier and missing data handling, which was crucial for enhancing the robustness of our subsequent analyses. Data points that appeared as outliers within their respective columns were not immediately discarded. Instead, these potential outliers were subjected to modeling using other predictor variables. Any data points that significantly deviated from the predictions of these models were then considered actual outliers and removed [44].

When it came to addressing missing data, a structured decision-making process was followed, as outlined in Figure 2. Initially, we assessed if the missing data could be retrieved from reliable online sources. If available, these values were filled in. In the absence of reliable online sources, we proceeded to assess the significance of the column in question. For columns deemed less critical and constituting less than 1% of the total data, simple imputation techniques were employed. For more important columns, we resorted to model-based imputation techniques. The choice of model-based technique was then determined by the structure of the non-missing data points. If these data points could be fitted into a regression line, regression imputation was utilized. Otherwise, K-Nearest Neighbors (KNN) imputation was employed.

This section is detailed under four categories, each corresponding to previously discussed feature subsets: Match Characteristics, Player Characteristics, Betting Odds, and In-play Statistics. Within the section, feature representation for both players is uniformly simplified, using a "p1" prefix to denote features that are also available for player 2.

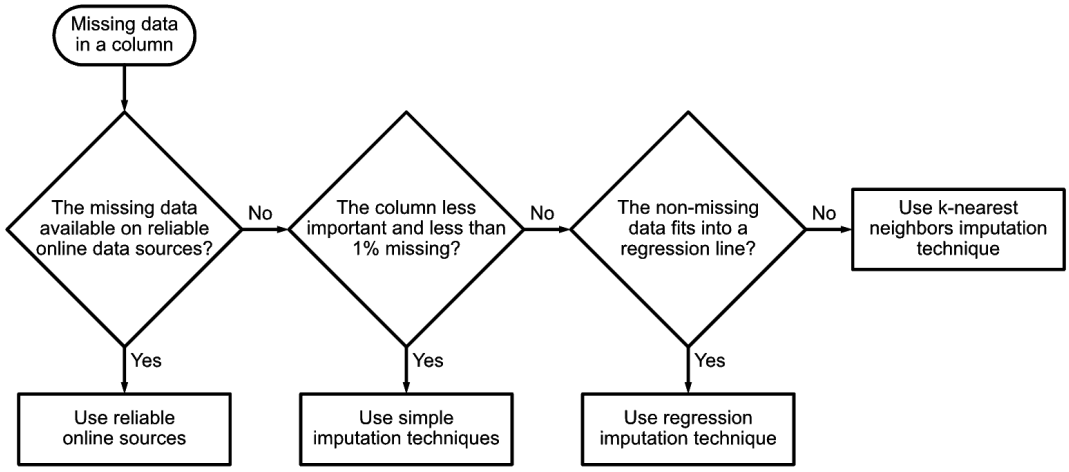

Figure 2: Data imputation decision-making flowchart

Match Characteristics

No outlier or missing data were identified in this category.

Player Characteristics

- The column that indicates the age of players at the time of each match, `p1_age`, contained no outlier as the age range for professional tennis players in our dataset spans from 15.4 to 44.0. The youngest player in the dataset is Stefan Kozlov, born in 1998, who competed in an ATP match in 2013; while the oldest player is Tomas Muster, born in 1967, who competed in an ATP match in 2011. A single instance was identified where the age was missing and was subsequently imputed using reliable online sources, such as the ATP Tour's official website.

- The column that represents the height of a player, `p1_ht`, contained no outlier as the height for professional tennis players in our dataset ranged from 158 to 211 cm. The shortest player in the dataset is Mehdi Ziadi, while the longest player is Reilly Opelka. There were 253 match instances where 129 distinct players were missing their heights. The missing data points were imputed using reliable online sources. For 27 players whose heights could not be identified, median imputation was applied, setting their height to 185 cm. Median imputation is a technique that replaces missing values with the median value of the available data [45]. The median was chosen over the mean or mode because it is less sensitive to outliers and skewed data.

- There were 133 match instances where the dominant hand, `p1_hand`, was missing for 65 players. The majority of these were filled out using reliable online sources. For the remaining four players, mode imputation was employed, assigning them as right-handed based on the predominant orientation in the dataset. Mode imputation is a technique used to replace missing values in a dataset with the mode, which is the most frequently occurring value [46]. This method was chosen because approximately 86% of all players in the dataset are right-handed.

- In 87 match instances, both ATP rank points (`p1_rankpt`) and rankings (`p1_rank`) were missing for 58 players. To address this absence and maintain consistency in our data representation, we assigned these players a rank point of 0 and an arbitrary ranking of 2200. This choice was

based on the observation that the player with the lowest defined ranking, 2159^{th}, possesses a rank point of 1. Therefore, by assigning a rank point of 0 and a ranking of 2200 to players without rank points and rankings, we ensured a coherent ordinal representation while distinguishing them from players who have earned rank points.

- $30,243$ matches were missing seed, `p1_seed`, for at least one of the competing players. To address this, we grouped the data by `year` and `tourney_name` to calculate the maximum seed (`max_seed`) assigned in each tournament. Later, a calculated midpoint between the total number of players (`draw_size`) and the maximum seed (`max_seed`) in each tournament was assigned to these instances. By using this imputation strategy, we aimed to assign a reasonable and consistent seed value for players lacking this information.

Betting Odds

- A total of 51 matches were missing all six bookmakers' betting odds. To address this, we were guided by Kovalchik's study [2], which found a consensus model based on various bookmakers' betting odds to be most accurate for tennis win prediction, as well as by Shah et al.'s study [47] that highlighted the efficacy of KNN imputation in high-dimensional datasets. Given these insights, we employed KNN imputation to impute the average of the missing betting odds using the newly constructed features `diff_rankpt`, `diff_rank`, and `p1_points_ratio` as predictors for the imputation process.

The exact steps taken during the handling of the Betting Odds are explained as follows:

Our initial step involved investigating the underlying cause of the missing data for the Betting Odds. We discovered that in all 51 instances where betting odds were missing, one player had to withdraw from the match for various reasons, such as injury. Consequently, a "lucky loser" took their place. Due to these unique circumstances, bookmakers often abstain from offering odds, explaining why these data points were intentionally left blank in the dataset. The distribution of these missing values appeared to be random and proportional.

Instead of trying to predict six different betting odds for six bookmakers, for simplicity in calculations, a new feature, `p1_Avg` (see Section 3.3.3.3), was formed to hold the average betting odds in favor of player 1 for each match.

To boost the precision of our predictions, we introduced several experimental features such as `diff_rankpt` and `diff_rank`, which represent the difference between the rank points and the rankings of the two opposing players, and `p1_points_ratio`, `p1_games_ratio`, and `p1_sets_ratio` which measure MOV based on points, games, and sets won in a match by player 1. The calculation of these new features is explained in detail in Section 3.3.3.3. We chose these particular features as they were expected to show a strong relationship with the average betting odds.

During this phase, it became imperative to assess how closely and in what direction our target variable, `p1_Avg`, was related to various other variables. These included differences in rank points and rankings, as well as metrics for MOV based on points, games, and sets. We also considered player-specific In-play Statistics, such as the number of aces, serve points, and serve games for each player. To conduct this assessment, Spearman's correlation test was chosen over Pearson's due to the visibly non-normal distribution of `p1_Avg`, as confirmed by the Kolmogorov-Smirnov test.

Spearman's correlation is a non-parametric test that measures the strength of a monotonic relationship, whether linear or non-linear, without making assumptions about the distribution of the data. This method is particularly useful in scenarios involving non-linear relationships or non-normally distributed data, making it more suitable for our analysis than Pearson's correlation [48]. This approach enabled us to accurately capture the nature of the relationships between `p1_Avg` and other variables in our dataset.

Reflecting these insights, the feature `diff_rankpt` exhibited the strongest negative correlation, registering a coefficient of -0.81. This negativity indicated an inverse relationship, logically suggesting that smaller betting odds for a player correlate with higher potential dominance by that same player. Higher rank point differences, as well as outperforming the opponent in terms of points, games, and sets, further underscore this dominance. Other features like `diff_rank`, `p1_points_ratio`, `p1_games_ratio`, and `p1_sets_ratio` followed with coefficients of 0.73, -0.55, -0.53, and -0.49, respectively (Table 2).

Table 2: Correlation Between 'p1_Avg' and its Predictors

Target	**Variable**	**Correlation**
	diff_rankpt	-0.81
	diff_rank	$+0.73$
p1_Avg	p1_points_ratio	-0.55
	p1_games_ratio	-0.53
	p1_sets_ratio	-0.49

Our preliminary objective was to discern straightforward relationships between the available variables and average odds (`p1_Avg`). We conducted first- and second-order regression analyses, yielding R^2 scores of 0.383 and 0.434, respectively. These analyses neither pinpointed outliers nor exhibited a strong fit with our target variable (Figure 3).

Finally, to impute the 51 missing data points in the Betting Odds, we used KNN imputation, guided by the three most highly correlated features. In this technique, missing values are substituted with those from the k data points that are the closest in terms of Euclidean distance. Leveraging the positive correlations between rows, the method assumes that the k closest data points offer the most reliable information for filling in the missing values [49]. The optimal value of k was determined through the elbow method, which balanced minimizing the Mean Squared Error (MSE) with computational efficiency. Our analysis revealed that a k value of 5 was the most effective in achieving this balance (Figure 4).

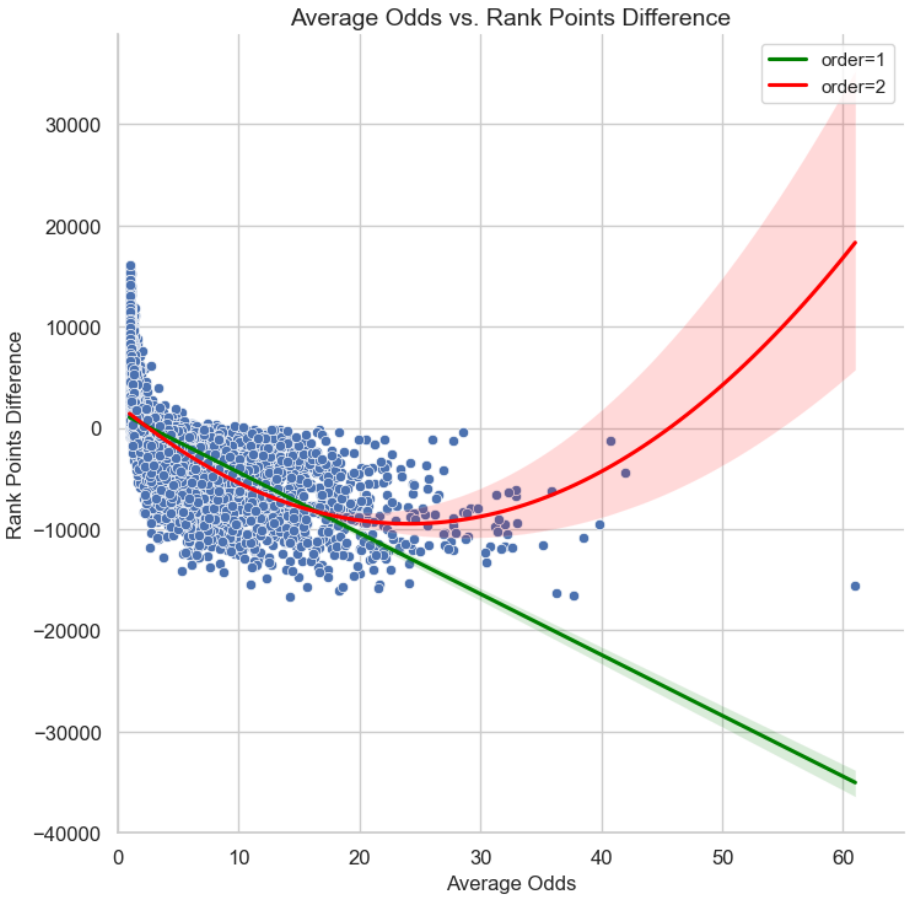

Figure 3: First and second-order regression analyses on average odds and rank points difference

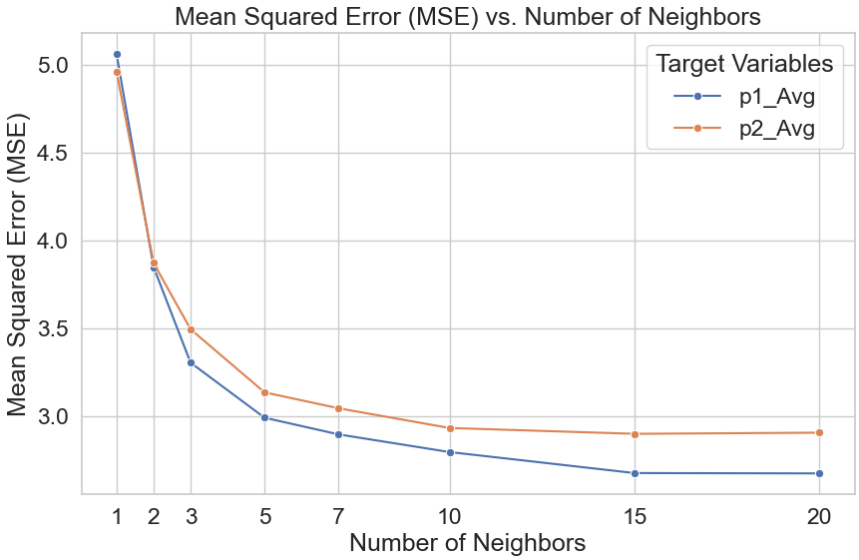

Figure 4: Optimal k-value determination for KNN imputation of average odds

In-play Statistics

- A total of 34 matches were missing In-play Statistics, excluding the `minutes` and `score` columns. Of these, 18 matches originally lacked this data, and an additional 16 matches had their In-play Statistics removed during the Data Validation Checks detailed in Section 3.3.4.1. Imputation for these missing data points was conducted using reliable online sources.
- The match duration column, `minutes`, had $1,390$ missing data points out of a total of $34,121$ matches, accounting for approximately 4% missingness. Since it is higher than 1%, model-based imputation was applied. Specifically, the missing values were imputed via linear regression, with the newly constructed feature `total_points` serving as the predictor.

The exact steps taken during the handling of the match durations are explained as follows:

Firstly, we initiated an investigation to determine if there was a hidden mechanism that could explain the missing data in the `minutes` column. Our exploratory data analysis revealed that $1,212$ of these missing values originated from matches played in the year 2015 (Figure 5). However, the missingness did not show any discernable pattern related to other variables. More specifically, the missing values in the `minutes` column were distributed in a manner that appeared random and proportional, with the exception of a noticeable pattern for the year 2015.

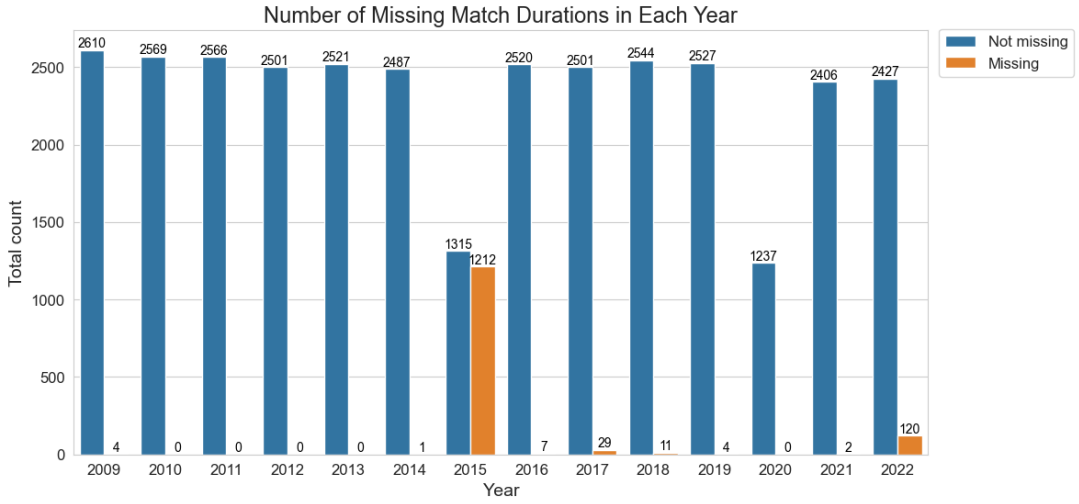

Figure 5: Annual distribution of missing match durations

We further probed whether there were significant variations in match durations across different years. A box plot visualization showed that the first and third quartiles, as well as the median of match durations, were mostly consistent across years, implying that there were no major differences that could potentially skew our imputation strategy (Figure 6).

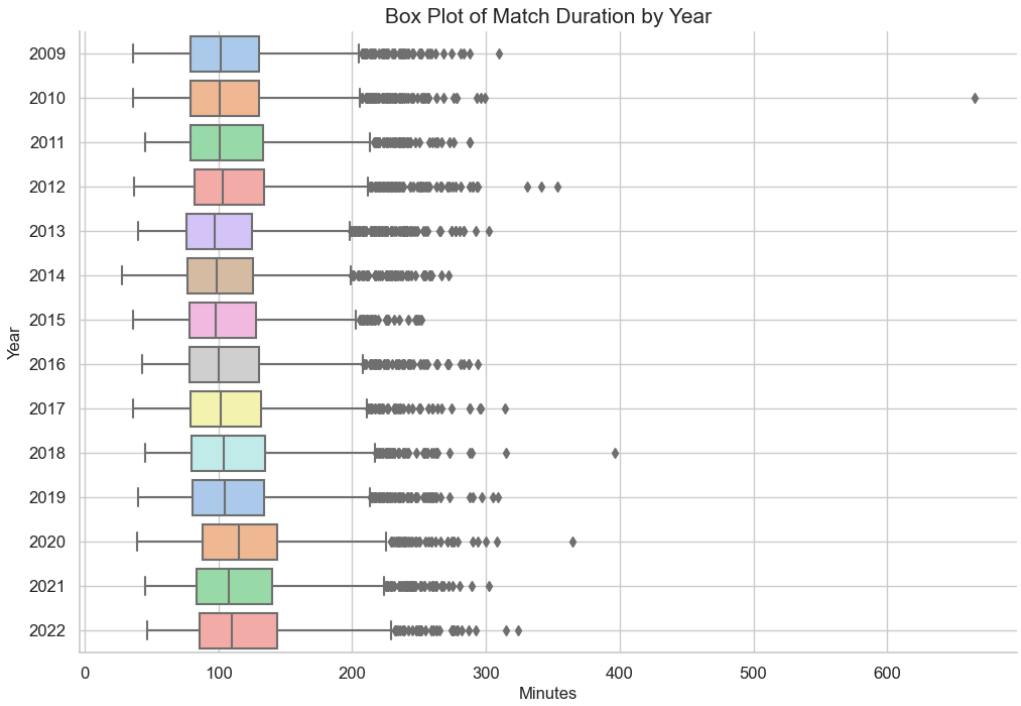

Figure 6: Box plot analysis of match duration by year

When performing missing data imputation, our goal is to predict the missing values as accurately as possible using all the available data points. Therefore, there is no need to worry about any data leakage related to information obtained during and after a match (i.e., the In-play Statistics). To enhance predictive accuracy, we introduced new experimental features such as `total_points` and `total_games`, which represent the total number of points and games played in a match, respectively. The calculation of these new features was based on the In-play Statistics, as to be explained in detail in Section 3.3.3.3. These features were specifically selected because they were anticipated to exhibit linear relationships with match duration.

At this stage, it was crucial to quantify both the strength and direction of the relationship between our target variable, `minutes`, and various predictors. These predictors included total points, total games, as well as the total number of aces, serve points, and serve games for each player. Given that the distribution of `minutes` was visibly right-skewed and confirmed as non-normal by the Kolmogorov-Smirnov normality test, we opted for Spearman's correlation test over Pearson's. According to the test results, newly created `total_points` feature showed the highest correlation with a coefficient of 0.95, closely followed by `p1_svpt`, `total_games`, `p1_SvGms`, and `p1_points` with coefficients of 0.92, 0.92, 0.91, and 0.90, respectively (Table 3).

We initially aimed to identify simpler relationships between our non-missing variables and `minutes` column. To this end, we employed a linear regression analysis, which provided an initial R^2 score of 0.878. This analysis highlighted the presence of several outlier match durations that skewed the regression line.

Table 3: Correlation Between 'minutes' and its Predictors

Target	Variable	Correlation
minutes	total_points	+0.95
	p1_svpt	+0.92
	total_games	+0.92
	p1_SvGms	+0.92
	p1_points	+0.90

To systematically identify the number of outliers that should be removed, we employed the elbow method, visualized in Figure 7. The technique involved plotting the R^2 scores against different threshold levels for marking outliers. Using this approach, we found that a threshold of 0.4 offered an optimal balance between maximizing the R^2 score and minimizing the number of data points removed.

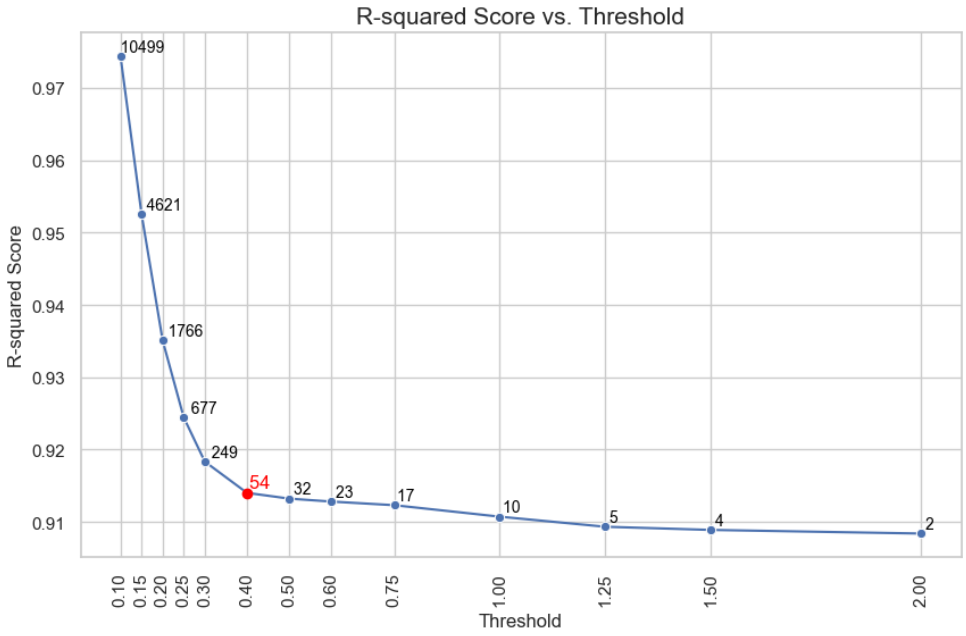

Figure 7: Elbow plot for outlier removal threshold determination

The effect of the outlier removal process is illustrated in Figure 8, where we displayed the regression lines and data points before and after the removal of outliers. With the chosen threshold of 0.4, a total of 54 match duration values were marked as outliers and removed from the dataset, resulting in an improved R^2 score of 0.914. It is worth noting that if we had simply used the `minutes` variable alone to identify outliers, we would have mistakenly removed matches with durations of 650 and 395 minutes, along with those lasting 1160 and 990 minutes. However, our regression analysis indicated that the matches with durations of 650 and 395 minutes were not outliers, as they closely aligned with the total points, thereby validating our model-based outlier detection approach.

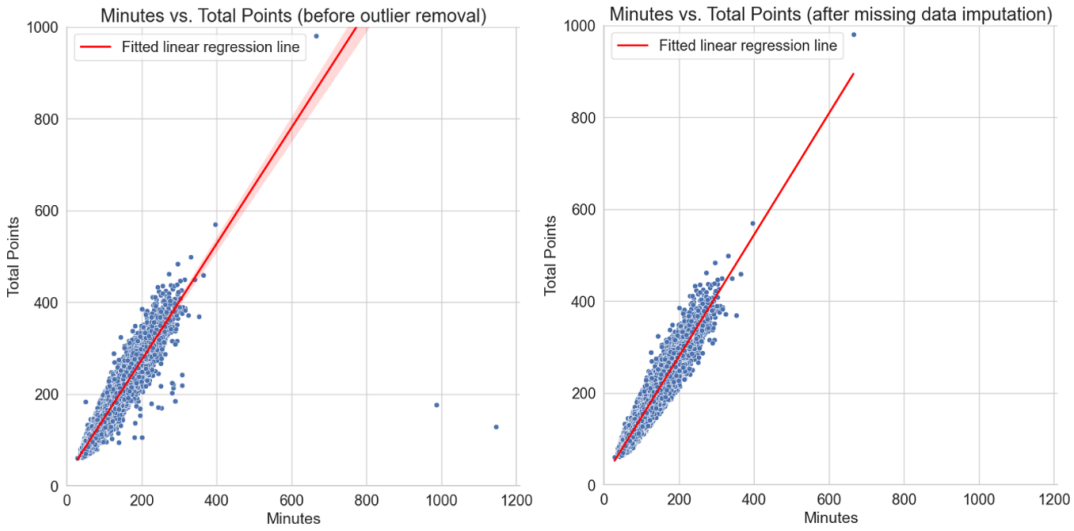

Figure 8: Pre- and post-outlier removal regression analysis

As a final step, we employed regression imputation to address the $1,440$ missing data points in the `minutes` column ($1,390$ initially missing and 54 identified as outliers). This method was chosen because it aligned well with our regression model. This method substitutes missing values based on other related variables in the dataset. Caution is advised when selecting predictor variables since poor choices can lead to inaccurate estimates. Ideally, predictors should be selected based on their ability to account for the highest variance, R^2 score, in the target variable [50].

In this study, model-based outlier detection, as well as mode, median, regression, and KNN imputation techniques, were put into use. This unified approach not only rigorously identified and removed outliers but also adaptively managed missing data. As a result, we maintained both the integrity and robustness of the dataset, setting a strong foundation for the subsequent ML analyses.

3.3.3 Feature Extraction

Feature extraction is a critical component within the Data Preparation and Feature Extraction phase in building an effective predictive model, particularly when dealing with complex systems like tennis matches. In tennis outcome prediction, each training instance represents a historical match and consists of the following:

1. A vector of input features (X), representing the characteristics of the players and the match.
2. A target value (y), corresponding to the outcome of the match.

The trained model can then be applied to predict the outcome of a future tennis match as long as the required set of input features can be assembled for that match. Upon training, the model leverages this structure to forecast outcomes of future tennis matches, provided that it has access to a similar set of input features for the upcoming matches.

Our feature extraction process employed a range of techniques aimed at enhancing predictive accuracy of the models. These techniques were categorized as essential, which are fundamental to our model, and optional, which are applied to further boost the predictive relevance of individual features.

3.3.3.1 Essential Techniques

Symmetric Representation

In the construction of a predictive model for tennis match outcomes, it is essential to include all relevant data concerning both players. This requires capturing two metrics for every variable, one for each player.

A straightforward approach is to include each player's individual data as separate features, which might preserve more detail about the match. However, the literature suggests that synthesizing a single feature by calculating the difference between these paired features yields more informative outputs than utilizing these features individually [51] [52]. Take ATP rankings as an example: We formed a feature by subtracting the rank of one player from the other, which has been shown to be an effective predictor on its own, as used by Clarke and Dyte in their LR analysis [4].

The key advantage of using the differences in variables is that it leads to a model that is symmetric. A symmetric model would predict the same outcome even if we switched the players' positions. Without this, models can unintentionally give different importance to the same feature for each player, leading to inconsistent predictions. Furthermore, using differences between variables reduces the number of features, which simplifies the model and decreases its susceptibility to minor variations in the training data, consequently preventing overfitting [52].

Historical Averaging

In our dataset, Match Characteristics, Player Characteristics, and Betting Odds contain only pre-match information and thus do not cause data leakage for an ML model. Such features reliably reflect the pre-game conditions and player attributes. Conversely, In-play Statistics, such as final score and player performance metrics, if used as predictors, could inadvertently lead to data leakage and give a false impression of achieving high model performance because these statistics inherently contain information about the outcome [17]. To mitigate this and still harness historical data effectively, we employed a historical averaging process. This involves aggregating past data to estimate a player's current form and capabilities.

Furthermore, ATP rank points and ranking features under the Player Characteristics category also benefit from historical averaging. While they do not disclose match outcomes, analyzing their changes over time offers a valuable perspective on a player's form and trajectory. For example, examining the change in a player's ATP ranking over the past six months can provide insights into their performance momentum. The resulting averages are then synthesized with other features to form a comprehensive pre-processed dataset, which is then fed into the subsequent stages of analysis within the SRP-CRISP-DM framework.

This method, while providing a comprehensive view of a player's historical performance, does come with its limitations. It may not adjust for the quality of the opponents faced historically, potentially leading to skewed performance estimates. A player who has consistently faced top-tier opponents

may have a different performance trajectory compared to one facing lower-ranked opponents [9]. We explored methods to mitigate this bias through the use of common opponents in the following section.

Additionally, historical averaging treats all past matches as equally relevant, which may not be the case for predicting current performance. Therefore, we also apply a time decay weighting to give more significance to recent matches, as detailed in the next section.

3.3.3.2 Optional Techniques

Common Opponents Adjustment

In addition to essential feature extraction techniques, our model refines the historical data by considering the variability in the skill levels of past opponents. When aggregating player performance data, it is crucial to normalize for the quality of the opponents faced. A naive average could skew a player's estimated ability if there has been a significant disparity in the skill level of their past opponents. To adjust for this, we have incorporated an analysis of common opponents, inspired by the methodology proposed by Knottenbelt [9], which allows for a fairer comparison of players.

This adjustment process begins by identifying a set of common opponents that both players have faced and selecting a specific feature, such as win rate percentage (WRP). The common opponents are labeled as C_1 to C_n. For player i, $WRP_i(C_j)$ is their average win rate percentage in all matches against common opponent C_j, as outlined in Figure 9.

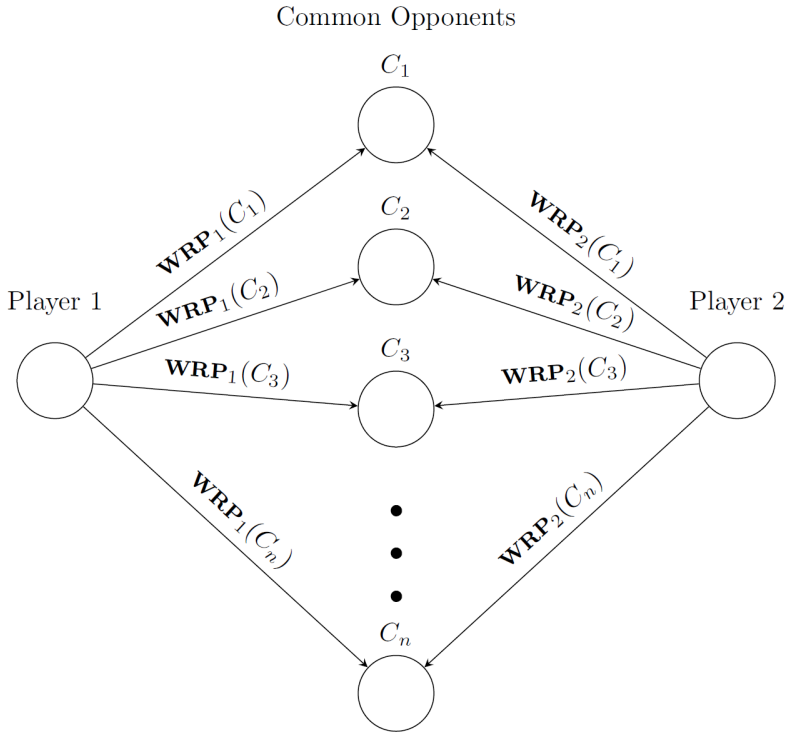

Figure 9: Diagram of common opponents win rate comparison (taken from Sipko [52])

We take the average of these values to obtain an estimate for each player:

$$WRP_i = \frac{\sum_{j=0}^{n} WRP_i(C_j)}{n}$$

Finally, we calculate the difference by subtracting the estimates for the two players, as discussed earlier regarding symmetrical representation:

$$WRP = WRP_1 - WRP_2$$

While this approach creates a consistent basis for comparison between the two players involved in an upcoming match, its accuracy depends on having a sufficient number of common opponents. To ensure reliability in cases where the number of common opponents is limited, we incorporate an uncertainty measure, which is elaborated upon in the coming sections. This measure adjusts the confidence in our predictions, taking into account the volume and significance of the data informing each feature.

Time Discounting

In professional tennis, a player's form is dynamic, with improvements and declines influenced by a variety of factors such as experience, age, injury, and private life events. For instance, a player's performance typically peaks during their prime years and may be affected by life events, with studies indicating noticeable impacts such as ranking point declines following personal milestones [53].

In response to this dynamic nature of player performance, our model places a greater emphasis on recent matches when estimating performance-related features, thus ensuring a more accurate reflection of a player's current form. We achieve this through a technique known as time discounting, where matches closer in time to the prediction date are given more weight than those further in the past. This temporal weighting is governed by the following exponential function:

$$W(t) = min(f^t, f)$$

Here, t represents the time elapsed since the match occurred (in days), and f is the discount factor. The discount factor, a value between 0 and 1, dictates the degree of impact that time discounting has on the historical data. A smaller f means that older matches contribute less significantly to the performance estimation. As illustrated in Figure 10, when we use a discount factor of 0.8, all matches within the past t days receive equal weight to prevent disproportionately large weights on the most recent matches. It is important to note that both t and f are hyperparameters that require optimization to fine-tune the model's sensitivity to the time relevance of data (detailed in Section 4.1.1).

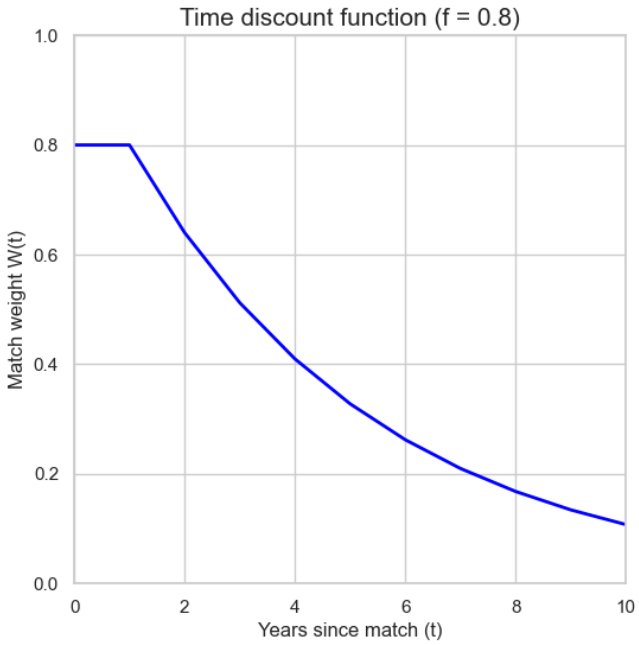

Figure 10: Time discount function for historical match weighting for f = 0.8

Time Frame Selection

Our dataset spans an extensive period of 14 years. Identifying the optimal historical data range for feature construction is another important hyperparameter of our feature extraction process.

Through this process, we explore multiple hyperparameter settings of the optional techniques, such as 1) applying common opponent adjustment, 2) with a time decay of 0.8 each 30 days, 3) using the most recent 365 days as the time frame, and evaluate their impact on the model's predictive performance. The optimal time period for each feature is thus selected based on a thorough grid search and empirical evidence. The intricacies of the grid search approach and the specific time frames adopted for each feature are elaborated in Section 3.5.3 and Section 4.1.1, respectively.

Uncertainty Assessment

Time discounting, as outlined earlier, is a technique that assigns weights to players' past matches to compute performance estimates for upcoming matches. Our methodology integrates these weights to assess uncertainty, which is a useful metric in removing noise prior to training and obtaining a level of confidence regarding the input features [52].

The computation of uncertainty begins with the accumulation of weights from a player's past matches, following the time-discounting approach. The sum of these weights for player i is represented by S_i:

$$S_i = \sum_{m \in P_i} W(m)$$

Here, W(m) stands for the weight assigned to match m, and P_i denotes the set of past matches for player i. This summation provides a weighted total that reflects the relevance and recency of the data being considered.

The overall uncertainty for a match's features, denoted by U, is then calculated as the inverse of the product of total weights for both players involved in the match:

$$U = \frac{1}{S_1 \cdot S_2}$$

A high value of U indicates greater uncertainty, implying that the data is either sparse or not recent enough to be deemed reliable. Conversely, a low value of U suggests that the match features are based on substantial and relevant data history, thus providing confidence in the predictive model's estimates. We set a parameter termed "uncertainty threshold". If this U value surpasses this threshold, it indicates a deficiency in the historical match data, and such values are disregarded. Uncertainty threshold serves as an additional hyperparameter within our feature extraction process and needs to be fine-tuned, as previously detailed.

The integration of uncertainty assessment into our feature extraction process complements other optional techniques such as common opponent adjustment, time discounting, and time frame selection. Each of these techniques refines the historical data, adjusting for opponent quality, temporal relevance, and data recency to provide a nuanced and accurate reflection of a player's current form. This ensemble of techniques ensures that our model is not only informed by comprehensive historical data but also calibrated for the inherent variability and unpredictability of player performances.

3.3.3.3 New Feature Construction

We developed several new features by applying the feature extraction techniques previously discussed. Our approach to feature extraction typically involves three steps:

1. For features containing values for both players, we apply symmetric representation.
2. For features not known by the time the match starts, such as all In-play Statistics listed in Table 1, we use historical averaging. This approach is also used for the changes in the ATP rankings of players over time.
3. For all features, we systematically explore various settings and combinations of optional techniques, where applicable, and evaluate their performance using the Chi-square (χ^2) score.

   The χ^2 score is a statistical measure used to determine the relationship between two variables in a contingency table. A χ^2 score quantifies how much the observed frequencies differ from the frequencies that would be expected if there were no relationship between the variables [54]. In the context of feature construction, the χ^2 test allows us to quantify the strength and significance of the relationship between newly created categorical features and the target variable. A higher χ^2 score indicates a stronger relationship and suggests that the feature may be predictive of the outcome, while a low score suggests a weak relationship.

The section is divided into five subsections, focusing on different data subsets: the Class variable, Match Characteristics, Player Characteristics, Betting Odds, and In-play Statistics. To provide a comprehensive background for each feature, we have included concise references to studies where similar features have been implemented, aiding in understanding their relevance and validation in the context of tennis match prediction.

The Class Variable

- **p1_won:** A binary class variable, set to 1 if player 1 wins the match and 0 if player 2 wins. In tennis, there are only two possible outcomes for a match: a win or a loss. Our dataset reflects this binary outcome structure.

Match Characteristics

- **match_id:** A unique key identifier created for each match in the dataset. It is a combination of `tourney_id` and `match_num` (e.g., 2022-9410-001), ensuring each match is distinctly identified.
- **is_grandSlam:** A binary indicator, set to 1 for Grand Slam matches and 0 for others. This feature was included following our exploratory data analysis, which revealed that the predictability of outcomes in Grand Slam matches appears higher, with the favorite winning 73.35% of the time, compared to 64.17% for non-Grand Slam matches (Figure 11). These percentages align with the findings that Grand Slam matches, typically played in a best-of-5 set format, tend to be more predictable and often favor higher-ranked players [6] [16].

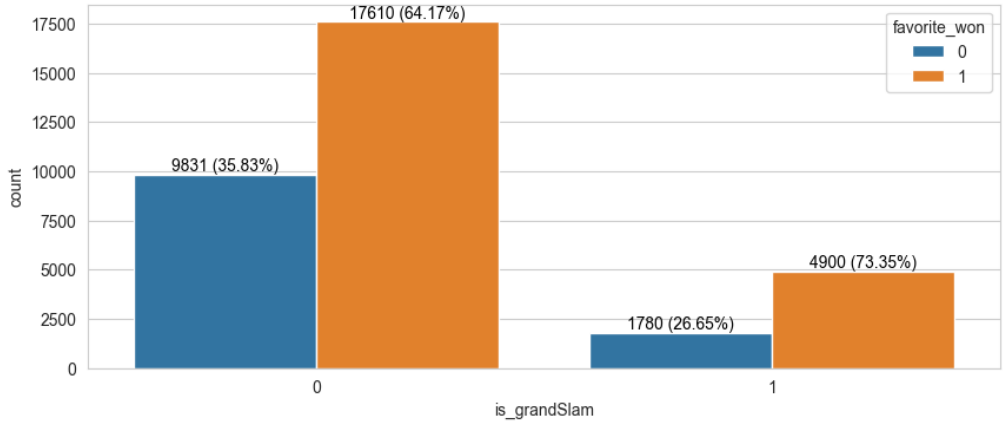

Figure 11: Grand Slam impact on favorite players' win rate

- **diff_surfaceAdvantage:** A categorical variable quantifying each player's advantage based on the match surface. It is derived from comparing a player's historical performance on the specific surface of the upcoming match using the win ratios. This feature is set to 1 if the surface of the match aligns with player 1's historically preferred surface, -1 if it aligns with player 2's, and 0 if there is no clear surface preference for either player. The implementation of this feature is inspired by the understanding that certain players perform better on specific surfaces [55] [21].

Player Characteristics

- **is_formerTop10:** A binary feature, assigned the value 1 if either player in a match has been in the ATP top 10 during a specified period, and 0 otherwise. The inclusion of this feature was inspired by Del Corral and Prieto-Rodriguez's study, which suggested that match outcomes are more predictable for players who have been ranked in the top 10 [6].

- **diff_rank, diff_rank_log, diff_rankpt, diff_rankpt_log:** Normal and log-scaled differences in rankings and rank points between two players. Incorporating insights from Kovalchik [2] and Wilkens [21], we recognized the predictive power of player rankings in tennis match outcomes and thus introduced features based on rankings and rank points. The underlying idea behind the use of a logarithmic scale is that the skill difference is larger among high-ranked players than among medium-ranked and even larger among low-ranked players [5] [6] [21]. The logarithmic transformation aims to help capture the non-linear nature of skill differences across various ranking tiers.

- **diff_rankMomentum_log, diff_rankptMomentum_log:** Log-scaled difference in player's average ranking or rank points over the previous given period minus his current ranking or rank points [21]. A positive value for the rankings-based feature (or a negative value for the rank points-based feature) suggests that the player has achieved a winning streak. This "momentum" is presumed to influence the outcome of the upcoming match, capturing the dynamic changes in a player's form over time.

- **favorite_won:** A binary feature indicating the victory of the higher ATP-ranked player. It is set to 1 when the favorite (the player with the better ranking) wins and 0 when the underdog (the player with the lower ranking) wins. This feature encapsulates the general expectation in tennis that players with superior rankings are more likely to win. However, it is important to note that when combined with rank and rank points features, `favorite_won` could lead models to directly infer the match winner, fully skewing the predictive process. Therefore, we primarily utilized this feature for exploratory data analysis and not as a direct input for predictive modeling.

- **diff_age, diff_ht, diff_seed:** Difference in age, height, tournament seeding of the two players [6] [32] [16] [21]. We included these features to observe whether age, height, and seeding have a significant effect on tennis match outcome.

- **diff_agePersonalPeak:** Reflects the age difference between each player's current age and their personal peak performance age. This feature is crafted based on historical data showing the age at which each player achieved their highest ATP rank points. A significant age difference from their personal peak could indicate a player's progression or decline phase, potentially affecting their performance in the match.

- **diff_ageGeneralPeak:** The absolute difference between each player's current age and the general peak performance age across all players. This measure considers the median peak performance age of all players (including only those with at least 10 matches in our dataset), offering a benchmark to evaluate how a player's age aligns with the typical peak performance age in professional tennis. As demonstrated in Figure 12, the median peak performance age, based on ATP rank points, is 26.75 years. A lower value of `diff_ageGeneralPeak` indicates proximity to

this median peak age, potentially signifying a player's prime performance period, which may offer a competitive advantage in upcoming matches.

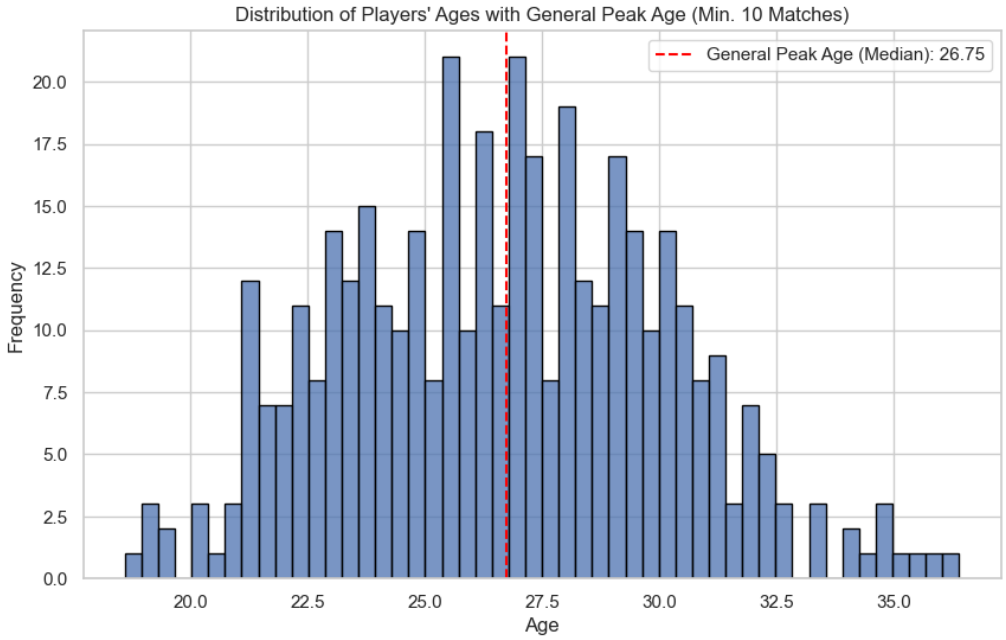

Figure 12: Histogram of players' ages at peak performance

- **diff_homeAdvantage:** A categorical variable reflecting the player's competitive edge when playing in their home country. It is set to 1 if player 1 is playing in his home country, -1 for player 2, and 0 if the match location is neutral to both players or if both are playing at home. This feature is founded on the concept of "home advantage", observed across various sports, where players often exhibit enhanced performance on familiar ground, possibly due to local support, acclimatization to conditions, or reduced travel fatigue. While the existence and extent of home advantage in tennis are debated, several studies have examined its effects in sports [39] [32] [16].

- **handedness:** A categorical variable that categorizes the dominant playing hand of each player, with four possible outcomes: both right, both left, favorite right and longshot left, favorite left and longshot right [6] [16] [21].

Betting Odds

- **p1_Avg, p2_Avg:** Average closing betting odds of six distinct bookmakers for player 1 and player 2, respectively. These odds are crucial as they represent the aggregated perspective of multiple bookmakers, translating into implied probabilities of each player winning the match.

- **diff_Avg_implied:** Difference between the normalized implied probabilities of the two players [32] [16], calculated using Shin's method [56] [57]. Bookmakers usually do not offer fair odds, meaning that the sum of the two odds exceeds 1. This practice is known as overround, and it inflates the odds. In order to use inverse odds as accurate probability estimates, a normalization is required. Here, we normalize these inverse odds according to Shin's method, which corrects

for the overround and ensures the probabilities sum to 1. This normalization process converts the inverse odds into more realistic probability estimates. This feature is particularly insightful, as it highlights the relative perceived strengths of the players according to the betting market.

In-play Statistics

- **diff_gamesFatigue:** Difference in games-based fatigue between the two players [52] [16]. This feature is calculated by assessing the total games played by each player in a specified period before the match, weighted by how recent each game was to the match date. The resulting difference in fatigue levels offers insight into the potential physical strain on the players leading up to the match, with higher values suggesting a greater difference in recent playing workload. This feature is crucial in understanding player readiness and recovery, which can significantly influence match outcomes, particularly in tournaments with dense schedules.
- **diff_avgDuration:** Difference in the average match duration between the two players. This measure is derived from the `minutes` column, representing the length of past matches for each player. To avoid data leakage—since the duration of a match is available post-completion—the feature uses historical data up to a specified period before the current match. The difference in average durations highlights variations in players' typical match durations, with a higher difference suggesting a notable disparity in their current stamina and playing style.
- **diff_surfaceWinRatio:** Difference in players' win ratios on the specific surface type for the upcoming match [16]. This metric is calculated using historical data on each player's past performance on the same surface type, providing insight into how well each player adapts to different surfaces.
- **diff_overallWinRatio:** Difference in overall win ratios for each player, determined from their past match outcomes [16]. This feature reflects the general performance level of each player by comparing their winning track record, offering a broad perspective on their competitive effectiveness.
- **diff_h2hWinRatio:** Head-to-head win ratio difference between the two players [52] [16] [21]. This is calculated by analyzing the outcomes of past matches where both players have faced each other, offering a direct comparison of their performance in similar match-up scenarios.
- **diff_winLossStreak:** Difference in recent win-loss streaks for each player [16]. This feature considers the consecutive wins or losses each player has experienced leading up to the match, providing insight into their current form and momentum.

To streamline the explanations in the remaining of this section, we represented any feature that exists for both player 1 and player 2 from the perspective of player 1, using the prefix "p1". Any feature and any calculation with the "p1" prefix in its naming is also available for player 2. In the new feature construction process, we prioritize clarity and precision, particularly when translating the In-play Statistics into historical statistical features. These In-play Statistics include aces (`p1_ace`), double faults (`p1_df`), serve points (`p1_svpt`), first serves in (`p1_1stIn`), first serve wins (`p1_1stWon`), and second serve wins (`p1_2ndWon`).

To comprehensively understand the In-play Statistics in our dataset, it is essential to dissect how a typical point in a tennis match unfolds, excluding "let" scenarios. A typical tennis point starts with a first serve (`p1_svpt`). If it lands in the service box (`p1_1stIn`), it may result in an ace (`p1_ace`) or a rally, leading to the server (`p1_1stWon`) or receiver winning the point. A failed first serve leads to a second serve attempt, which can also end in an ace, a rally (`p1_2ndWon`, if the server wins), or, if unsuccessful, a double fault (`p1_df`), conceding the point to the receiver. Figure 13 illustrates this progression, clearly associating each in-play event with its respective dataset column.

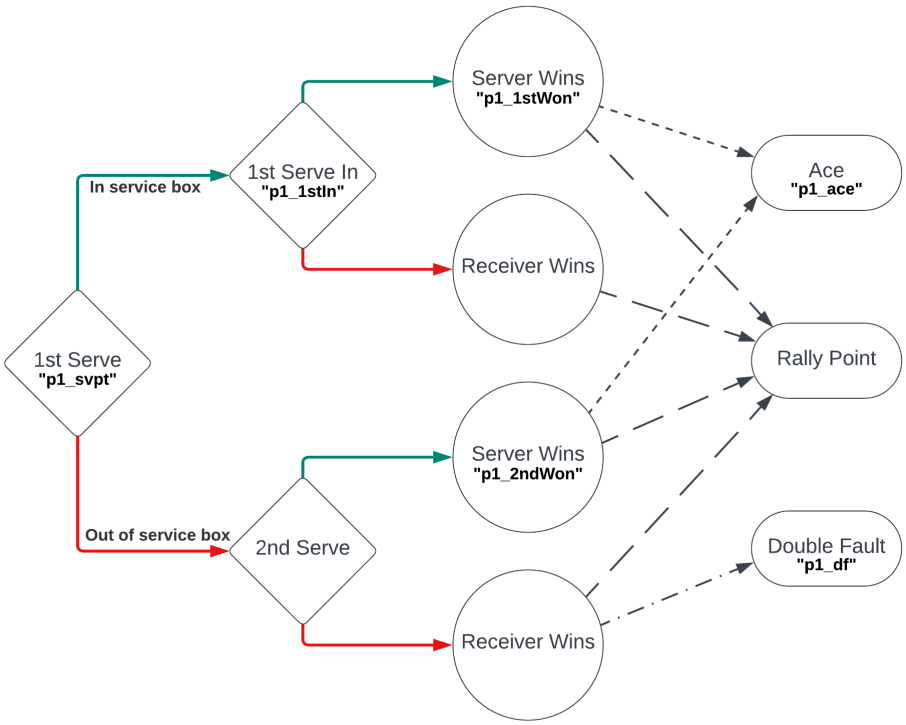

Figure 13: Flowchart of a typical tennis point

This flow from the serve attempts to point outcomes sets the ground for the calculations of historical in-play statistics:

- **p1_points, total_points, p1_games, total_games, p1_sets, total_sets:** Representation of the points, games, and sets won by player 1 and the total accumulated in a match. `p1_points` reflects the number of points by player 1, while `total_points` is the sum for two competing players, indicating the total points played in the match. The games/sets features follow the same structure, detailing the games/sets won by player 1 and the total games/sets played. Games and sets won by each player are derived from the `score` column. On the other hand, `p1_points` feature is derived using the following calculation:

$$\begin{aligned} p1\_points = {} & p1\_1stWon + p1\_2ndWon \\ & + (p2\_1stIn - p2\_1stWon) \\ & + (p2\_2ndIn - p2\_2ndWon) + p2\_df \end{aligned}$$

The calculation of `p1_points` combines player 1's successful first (`p1_1stWon`) and second (`p1_2ndWon`) serve points with points won on player 2's serves, accounting for double faults (`p2_df`) and returns.

- **p1_points_ratio, p1_games_ratio, p1_sets_ratio:** MOV measurements based on the points, games, and sets won by each player in a match [52]. These features are calculated as a proportion of the total, providing a normalized view of a player's performance relative to the overall match. They offer insight into the dominance or competitiveness of a match, with higher ratios indicating a more dominant performance by a player.

- **diff_firstServeInRatio:** Difference in first serve in ratios between two players [52] [16]. This feature highlights each player's accuracy and consistency in landing their first serves, an essential aspect of serving strategy. The ratio is calculated by dividing the number of successful first serves (`p1_1stIn`) by the total serve attempts (`p1_svpt`) for each player, and their differences are compared to illustrate relative serving precision.

- **diff_firstServeWinRatio:** Difference in first serve win ratios of competing players [52] [16]. This ratio assesses the effectiveness of players in winning points on their first serve, a key indicator of serving dominance. It is computed by dividing the number of points won on first serves (`p1_1stWon`) by the total number of first serves in (`p1_1stIn`) for each player, with the difference between these ratios indicating comparative serve effectiveness.

- **diff_secondServeWinRatio:** Difference in second serve win ratios between players [52] [16]. This feature evaluates the players' ability to win points on their second serves, a critical part of maintaining serve under pressure. The ratio is derived by dividing the number of points won on second serves (`p1_2ndWon`) by the number of second serves in, and their differences depict the players' relative strength in second serve situations. Note that `p1_2ndIn` is not readily available in the dataset and derived by subtracting the sum of first serves in (`p1_1stIn`) and double faults (`p1_df`) from total serve points (`p1_svpt`).

- **diff_overallServeWinRatio:** Difference in overall serve win ratios of the two players [52] [16]. This ratio provides a comprehensive view of players' overall effectiveness in serving. It is calculated by dividing the total serve points won—including first (`p1_1stWon`) and second serves (`p1_2ndWon`) combined—by the total serve attempts (`p1_svpt`).

- **diff_overallReturnWinRatio:** Difference in overall return win ratios between players [52] [16]. This metric reflects players' proficiency at winning points on their opponents' serves. The ratio is calculated by dividing the total points won on the opponent's serve by the total serve points faced. Specifically, the formula is expressed as:

$$p1\_overallReturnWinRatio = \frac{1}{p2\_svpt}\Big((p2\_1stIn - p2\_1stWon) + (p2\_2ndIn - p2\_2ndWon) + p2\_df\Big)$$

- **diff_completeness:** Difference in the "completeness" metric between two players [52] [16]. This feature reflects a player's all-round performance, combining their serving and returning effectiveness. Completeness is calculated by multiplying a player's overall serve win ratio (`p1_overallServeWinRatio`) with his return win ratio (`p1_overallReturnWinRatio`).

- **diff_acePerServePoint:** Difference in the ratio of aces per serve point between two players [52] [16]. This feature aims to quantify each player's efficiency in scoring aces relative to their total serve attempts, highlighting their serving competence. The ratio is calculated by dividing the number of aces (`p1_ace`) by the total serve points (`p1_svpt`), and two competing players' difference is taken to construct this feature.

- **diff_dfPerServePoint:** Difference in the ratio of double faults per serve point between two players. This metric aims to assess players' serving accuracy, with a lower ratio indicating better precision and control. It is derived by dividing the number of double faults (`p1_df`) by the total serve points (`p1_svpt`), and calculating the difference between two competing players' ratios.

- **diff_acePerServeGame:** Difference in the number of aces per serve game between the two players [52] [16]. This feature offers insight into a player's ability to score aces over the course of serve games. It is calculated by dividing the total aces (`p1_ace`) by the number of serve games (`p1_SvGms`) for each player, and comparing these ratios.

- **diff_dfPerServeGame:** Difference in the number of double faults per serve game between two players. This feature evaluates players' serving reliability, with fewer double faults per game indicating better serving consistency. The ratio is computed by dividing the total double faults (`p1_df`) by the number of serve games (`p1_SvGms`) for each player, and assessing the difference.

- **diff_bpSaveRatio:** Difference in the break point save ratio between two players [52] [16]. This statistic is an indicator of a player's resilience under pressure, reflecting their ability to save break points during crucial moments in a match. It is computed by dividing the number of break points saved (`p1_bpSaved`) by the total break points faced (`p1_bpFaced`) for each player, with the difference highlighting their comparative skill in defending break points.

3.3.4 Data Preparation

Preparation of the data serves as the final but critical stage of the Data Preparation and Feature Extraction phase, ensuring that the dataset is not only clean but also optimized for predictive modeling. This phase is divided into four pivotal tasks: Data Validation Checks, Train-Test Split, Feature Encoding, and Feature Scaling.

3.3.4.1 Data Validation Checks

The integrity of our dataset is fundamental to the accuracy and reliability of the predictive model we aim to develop. To this end, we have carried out a comprehensive series of validation checks designed to identify and rectify any inconsistencies or anomalies that could compromise the model's performance. These validation checks are organized into four categories, each corresponding to previously discussed feature subsets: Match Characteristics, Player Characteristics, Betting Odds, and In-play Statistics. Below, we highlight some of the key checks conducted in each group to illustrate our validation process:

Match Characteristics

- No duplicate rows representing the same match should exist within the dataset.
- Each match should be uniquely identified by a combination of `tourney_id` and `match_num`, with no repetitions.
- Game, set, and match scores should be within the bounds set by the official rules of tennis.
- The declared winner of the match should align with the recorded match score.

Upon careful review, we found that our dataset adhered to all the validation criteria for the Match Characteristics.

Player Characteristics

- A player should not be recorded as participating in two different tournaments within the same time frame.
- Static characteristics of a player, like id, name, and dominant hand, should remain consistent across different records within the dataset.
- Player attributes such as rankings, ages, and country codes should be within reasonable and officially recognized limits.

Our dataset showed full compliance with these validation criteria for the Player Characteristics.

Betting Odds

- Betting odds from any of the six bookmakers should not be less than 1.
- Betting odds from different bookmakers should not exhibit significant deviations from the mean of these six betting odds.

In our dataset, we encountered eight instances where betting odds from various bookmakers were slightly less than 1 (e.g., 0.972). Given that it is illogical for a bookmaker to offer odds less than 1—since such odds would never attract bets—these odds were rounded up to 1.

To further ensure the robustness of our betting odds data, we employed a Coefficient of Variation (CV) test. This involved calculating the mean, standard deviation, and subsequently the CV, for the odds associated with each player in every match. The CV serves as a normalized measure of the dispersion of betting odds across different bookmakers. If the CV exceeded a predetermined threshold of 0.4—a value determined through experimentation— the specific odd causing this high CV was deemed inconsistent. As a result, 209 such instances were identified and removed from the dataset. It is worth noting that while variations in betting odds are generally expected due to different calculation methodologies and market dynamics among bookmakers, such high CV values were considered red flags that warranted immediate removal from our dataset.

In-play Statistics

- The total number of games each player serves must correspond to the final score indicated in the `score` column. This ensures consistency between individual player statistics and the overall match outcome.
- The aggregate number of points played in a match should be no less than four times the completed games. This is in line with the fundamental scoring structure of tennis, where winning a single game requires a player to secure at least four points, scored as 15, 30, 40, and then "$game$".
- The count of first and second serves "in" (successfully within bounds) should at least match the number of serves won in the categories.
- The sum of each player's serve points should be equal to the total match points.

Our dataset mostly aligned with these expectations, with inconsistencies identified across 16 matches. These inconsistencies were flagged as data errors and subsequently removed from the dataset. These 16 missing data points were imputed using the imputation methods discussed in Section 3.3.2.

3.3.4.2 Train-Test Split

In sports prediction, it is critical to preserve the chronological order of data to ensure that future outcomes are predicted solely based on past matches. In this study, we opted for a time-based split for dividing the dataset into train and test sets. Our dataset spans 14 years, covering men's singles tennis matches from 2009 to 2022. The last two years, 2021 and 2022, were split as the held-out test set, while the years 2009 to 2020 served as the train set. This structure ensures that the model is trained on a broad and comprehensive set of match data while being tested on the most recent and, thus, most relevant matches. However, it should be noted that the effectiveness of this approach assumes that past trends somewhat influence future outcomes, an assumption that holds reasonably well in the domain of professional tennis [58] [59].

3.3.4.3 Feature Encoding

Feature encoding is a pivotal step in the data preparation pipeline for ML models. It involves converting categorical data into a numerical format. While some modern tree-based algorithms, such as XGB, can handle categorical data directly [60], encoding is often required for many ML algorithms, particularly linear models like LR. Additionally, encoding can make it easier to compare different types of algorithms on the same dataset.

Our dataset contains two types of categorical features: ordinal and nominal. Ordinal features consist of a finite set of discrete values with an inherent order or level of preference. In contrast, nominal features consist of a finite set of discrete values without any inherent relationship [61]. We employed two feature encoding techniques to suit the nature of each feature type:

1. Ordinal Encoding: For features with a clear ordinal relationship, categories are encoded in a manner that preserves this relationship. For instance, the `round` feature represents the progression of rounds in a tennis tournament, with categories such as "R128", "R64", "R32", etc. These were converted into numerical values, preserving their inherent order ("R128" as 1, "R64" as 2, and so on, up to "F" which was encoded as 7). `tourney_level` and `draw_size` are the other two ordinal features that were encoded in this way.

2. One-Hot Encoding: Each category of a nominal feature is transformed into a new binary feature. For example, the `surface` variable, which includes surface types of "Clay", "Grass", and "Hard", was converted into three separate binary features: `surface_Clay`, `surface_Grass`, and `surface_Hard`. Each of these new features takes a binary value of 0 or 1, indicating the absence or presence of the corresponding surface type for a given match. `court` and `handedness` were also one-hot encoded in a similar manner.

It is important to note that the train-test split was carried out prior to this step to prevent data leakage, which occurs when information from the test set inadvertently influences the train set [62]. Encoding and scaling the features using the test set could expose the model to future data, thereby compromising the study's validity. Therefore, feature encoding and scaling were applied separately to the train and test sets to maintain the validity of our model evaluation.

3.3.4.4 Feature Scaling

Feature scaling, the final step in data preparation, is indispensable for algorithms sensitive to the magnitude and range of features. It involves normalizing the magnitude of features in a dataset. Our dataset includes features with diverse magnitudes and ranges. For instance, `diff_rankpt` spans from $-16,000$ to $+16,000$, `diff_age` varies between -25 and $+25$, while most difference-based and binary features range from -1 to 1 and 0 to 1, respectively. To ensure ML models interpret these features consistently, feature scaling is essential. The importance of feature scaling is contingent upon the chosen ML models.

Gradient descent, an iterative optimization algorithm, is used by models like linear regression, logistic regression, neural networks, and principal component analysis to minimize loss functions, effectively reducing the error between predicted and actual values. Features with varying magnitudes and ranges can cause inconsistent step sizes during optimization. Hence, scaling ensures a smoother and faster convergence of gradient descent.

Distance-based models such as KNN, SVM, and K-means clustering are particularly sensitive to unscaled data as they determine similarity based on the distance between data points. Features with larger scales can dominate these models, necessitating feature scaling to ensure equal contribution from all features.

Tree-based models, DT, RF, and GB, are invariant to feature scaling. Each node in these models represents a single feature, and the split at each node aims to increase homogeneity unaffected by the scale of other features.

Since we plan to explore at least one model from each group, all features underwent scaling as a result of sheer necessity as well as to achieve consistency across models, enhance the interpretability

of feature importance, and facilitate a smoother and more efficient optimization process for models. Upon inspecting the types and distributions of our features, we categorized them into six distinct groups and applied scaling techniques based on the characteristics of each group:

1. **Nominal Categorical Features:** The features `surface_Clay`, `surface_Grass`, `surface_Hard`, `handedness_L`, `handedness_LR`, `handedness_R`, and `handedness_RL` were already one-hot encoded. Since these features take on values of 0 or 1, they were left unscaled, thereby preserving their comparable scale.

2. **Binary Categorical Features:** The features, `is_grandSlam`, `is_formerTop10`, and `is_outdoor` were also left unscaled. Similar to nominal categorical features, these binary features already exist on a comparable scale of 0 and 1.

3. **Ordinal Categorical Features:** The features `ord_tourneyLevel`, `ord_drawSize`, and `ord_round` were normalized between 0 and 1. This scaling was necessary as these features ranged between 1 and the number of distinct values. Normalization was performed using the formula:
$$X_{\text{normalized}} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

4. **Bimodally-distributed Numerical Features:** The features including `diff_Avg_implied`, `diff_rank_log`, and `diff_rankpt_log` exhibit a bimodal distribution with two distinct peaks. These features are symmetrical due to the calculation of differences between two players, and the peaks are not very distant from each other. To ensure uniform contribution to model training, these features were standardized using the formula:
$$X_{\text{standardized}} = \frac{X - \text{mean}(X)}{\text{std}(X)}$$

5. **Spike-distributed Numerical Features:** Features including `diff_rank`, `diff_rankpt`, and `diff_seed`, exhibit distributions with a pronounced spike. Although these features do not conform to a normal distribution, their standard deviations provide suitable scaling factors. Standardizing these features, using the same standardization formula as above, ensures that the spikes do not disproportionately influence the models.

6. **Normally-distributed Numerical Features:** All the remaining features, including differences of the Player Characteristics like `diff_ht` and `diff_age` and differences of historically averaged In-play Statistics like `diff_firstServeInRatio`, `diff_completeness`, are approximately normally distributed. To ensure their uniform contribution to model training, these features were also standardized using the standardization formula mentioned above.

Figure 14 illustrates the distributions of a representative categorical feature and a numerical feature from each described group prior to scaling.

By systematically executing these four tasks, this Data Preparation section ensures that our dataset is not only free from inconsistencies but also structured in a way that is optimized for ML modeling. Through this rigorous preparation, we aim to build a predictive model that is both accurate and reliable.

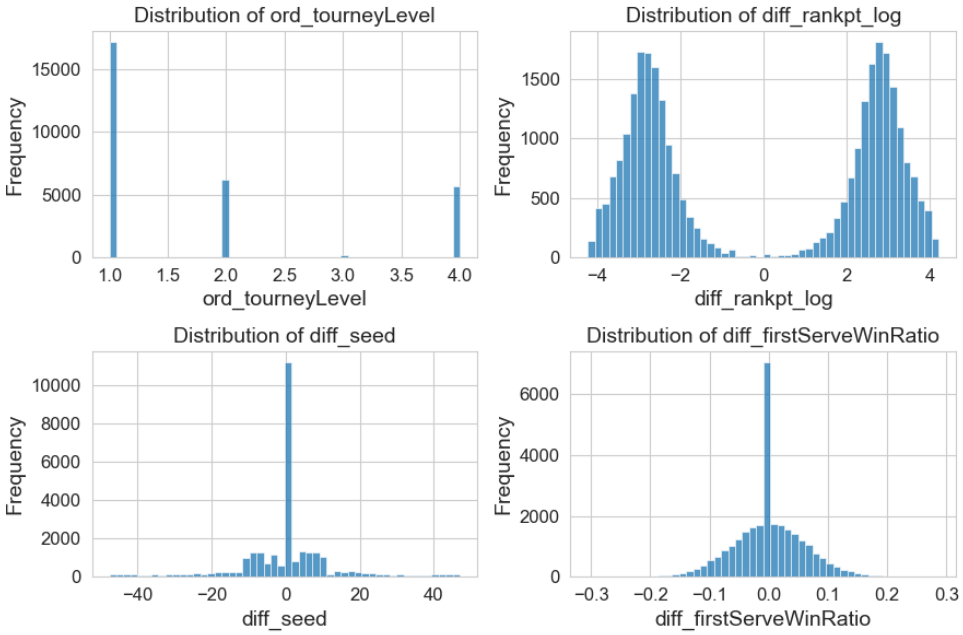

Figure 14: Pre-scaling distributions of the four different groups

3.4 Modeling

The Modeling section provides a detailed roadmap for applying ML techniques to the preprocessed dataset of tennis matches. It is methodically segmented into key phases: selecting suitable predictive models, undertaking the intricate process of feature selection, and compiling a concise summary of crucial features [17].

3.4.1 Model Selection

The Model Selection step is based on a comprehensive review of existing literature, encompassing a range of studies that have addressed the challenge of predicting outcomes in professional tennis matches. Our model selection was informed by a targeted literature review, identifying algorithms proven effective in sports outcome prediction. Key studies by Sipko [52], Cornman et al. [7], Koseler and Stephan [63], Gao and Kowalczyk [64], Wilkens [21], and Grinsztajn et al. [37] guided our choice of models that balance computational simplicity with predictive accuracy. Consequently, the following three models were selected for their distinct characteristics and proven track record in similar predictive tasks:

1. **Logistic Regression:** Selected for its robustness and efficiency in binary classification tasks, LR serves as our gradient descent-based algorithm, offering a strong baseline for performance comparison.

   LR is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous vari-

able—where there are only two possible outcomes. It estimates the probability of a binary response based on one or more predictor variables, in other words, features. The odds of the outcome are modeled as a linear combination of predictor variables using a logistic function. The LR function is defined as:

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n$$

Where:

- p represents the probability of the presence of the characteristic of interest,
- $X_1, X_2, \ldots, X_n$ are the predictor variables,
- β_0 is the intercept term,
- n is the number of predictors,
- $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients that represent the amount of change in the log odds of the outcome for a one-unit change in the predictor.

2. **Support Vector Machine:** Chosen for its capability to handle non-linear boundaries on high-dimensional data due to its use of kernel functions, SVM is utilized as our distance-based algorithm, adept at managing the complex patterns in the dataset.

   First proposed in the paper by Drucker et al. [65], SVM operates on the principle of finding the hyperplane that best divides a dataset into classes. The best hyperplane for an SVM is the one with the largest margin between the two classes. The margin is defined as the distance between the nearest data point of each class and the hyperplane. This concept is best visualized in a two-dimensional space where the data points are separated by a clear gap, as illustrated in Figure 15 [66]. For non-linear boundaries, SVM employs the kernel trick to transform the data into a higher dimension where a linear separation is possible. Common kernels used in this process include linear, polynomial, radial basis function, and sigmoid. The formula for an SVM utilizing the kernel trick is generally expressed as follows:

$$f(x) = \beta_0 + \sum_{i=1}^{n} (\alpha_i \cdot y_i \cdot K(x_i, x))$$

   Where:

   - $f(x)$ is the predicted output for the input x,
   - β_0 is the bias term,
   - n is the number of support vectors,
   - α_i are Lagrange multipliers (which are non-zero for support vectors),
   - y_i are the class labels,
   - $K(x_i, x)$ is the kernel function applied to the input data x_i and new input x. The kernel function transforms the input data into a higher-dimensional space where it becomes easier to separate the classes linearly.

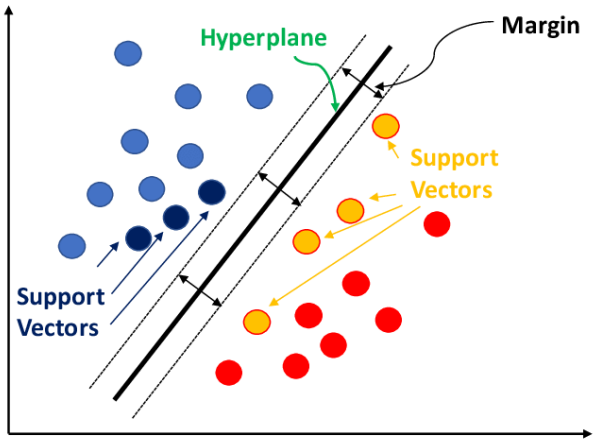

Figure 15: Schematic representation of SVM hyperplane of margins (taken from Lei et al. [67])

3. **Extreme Gradient Boosting:** Selected for its exceptional performance in a variety of ML competitions, XGB represents the tree-based algorithmic group, known for its scalability and high predictive accuracy.

   First proposed in the paper by Chen and Guestrin [60], XGB improves upon the concept of gradient boosting, as illustrated in Figure 16, by focusing on computational speed and model performance. It incorporates regularized boosting techniques to prevent overfitting and can process a vast amount of data relatively quickly. The algorithm constructs sequential trees where each tree aims to correct the errors of its predecessor, effectively combining multiple weak learners into a strong one. The base learner in XGB can be either a classification and regression tree or a linear classifier, depending on the choice made during the model configuration. The learning process involves minimizing an objective function that consists of a loss term measuring how well the model fits the data and a regularization term controlling the model's complexity. This makes the learning robust against overfitting and gives XGB a significant advantage in terms of predictive power. Furthermore, XGB has built-in routines to handle missing values, ensuring robustness to incomplete datasets. Its flexibility allows for custom optimization objectives and evaluation criteria, which is a significant advantage when adapting to various data science problems.

   The general form of the XGB model can be given by:

$$\hat{y}^{(t)} = \sum_{k=1}^{t} f_k(x_i)$$

   Where:

   - $\hat{y}^{(t)}$ is the predicted value for the i-th instance at iteration t,
   - f_k represents the k-th tree's contribution to the model,
   - x_i are the feature values of the i-th instance.

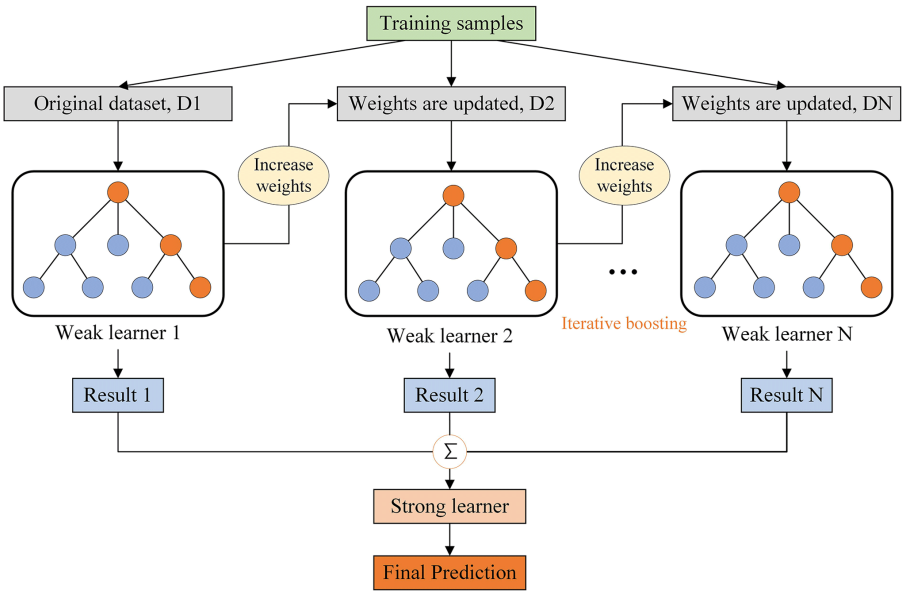

Figure 16: Illustration of the gradient boosting algorithm process (taken from Lei et al. [67])

3.4.2 Feature Selection

Feature selection is the process of selecting a subset of all available features to use within a model. Upon establishing the candidate models, our next step is to narrow down the dataset to the most informative features. This involves removing unnecessary or undesired features, mitigating multicollinearity, and experimenting with various combinations of features that have been selected by several feature selection algorithms. A model with fewer features has a lower variance, which prevents overfitting to the train set. In addition, feature selection will allow us to gain insight into the relative importance of different features in predicting match outcomes.

Preliminary Feature Removal

As a preliminary step to feature selection, we removed certain columns on our dataset in order to address four specific concerns: eliminating redundant features, maintaining a dataset of purely numerical features, ensuring feature symmetry, and preventing any potential data leakage. The following measures were taken to address these concerns:

- Columns that include pre-encoded Match Characteristics were removed as useful information they contain was extracted into new features during the Data Preparation and Feature Extraction phase of the framework.
- Columns that contain Player Characteristics about a single player were removed so as to provide symmetry to the feature set.
- Columns that include pre-historical averaging In-play Statistics were removed so as to prevent data leakage.

- The individual and average Betting Odds from bookmakers were rendered unnecessary, as we derived a more representative feature in the form of average implied win probabilities.

Mitigating Multicollinearity

Multicollinearity arises when two or more predictor variables in a regression model exhibit high correlation, undermining the statistical assumption of independent variables. This correlation can bias the model's results, obscuring the individual effect of each variable. In essence, the presence of multicollinearity can lead to misinterpretations of the data, as it becomes challenging to distinguish the separate influence of correlated predictors on the outcome variable.

The primary issue with multicollinearity is its concealment of the unique contribution of each independent variable. Ideally, a change in one predictor should not be associated with changes in another; however, multicollinearity violates this condition, making it difficult to isolate the effect of individual predictors on the response variable.

The significance of multicollinearity lies in its impact on regression models. It affects the precision of the estimated coefficients, which can become unstable and sensitive to minor changes in the model. This instability compromises the reliability of the model and its predictions, making it crucial to identify and address multicollinearity. In the presence of multicollinearity, gradient descent-based algorithms can experience slower convergence. Distance-based models can have their distances skewed by highly correlated features. Tree-based models are generally robust to multicollinearity, but it can still affect the interpretation and importance of feature variables.

Our approach to handling multicollinearity involved two steps:

1. **Correlation Matrix Analysis:** We started by constructing a Pearson correlation matrix to visualize the relationships between all features, setting a threshold of 0.85 to flag significant correlations (Figure 17).
2. **Feature Removal:** When faced with pairs of highly correlated features, we made the following strategic choices about which features to retain:
    - We decided to keep `diff_overallServeWinRatio` over `diff_firstServeWinRatio` as it contains more comprehensive information.
    - Features `diff_acePerServeGame` and `diff_dfPerServeGame` were removed due to near-perfect correlation with `diff_acePerServePoints` and `diff_dfPerServePoints`, indicating redundancy. To illustrate the concept of multicollinearity, Figure 18 shows scatter plots of two pairs of highly correlated features. The near-linear relationship between `diff_acePerServePoint` and `diff_acePerServeGame` supports the decision to exclude one to avoid redundancy in the model.
    - The ATP ranking, even when transformed to a logarithmic scale, does not adequately capture the subtle distinctions in the skill levels of top-tier players. In contrast, ATP rank points are inherently more granular and provide a more accurate and discriminating measure of the relative competencies and recent skill level of the players. Consequently, we eliminated features related to ATP ranking (i.e., `diff_rankMomentum_log` and `diff_rank_log`) in favor of those derived from ATP rank points.

- The feature `ord_tourneyLevel` was excluded due to its high correlation with the features `ord_drawSize` and `is_grandSlam`, whereas the latter two demonstrated negligible correlation with each other.

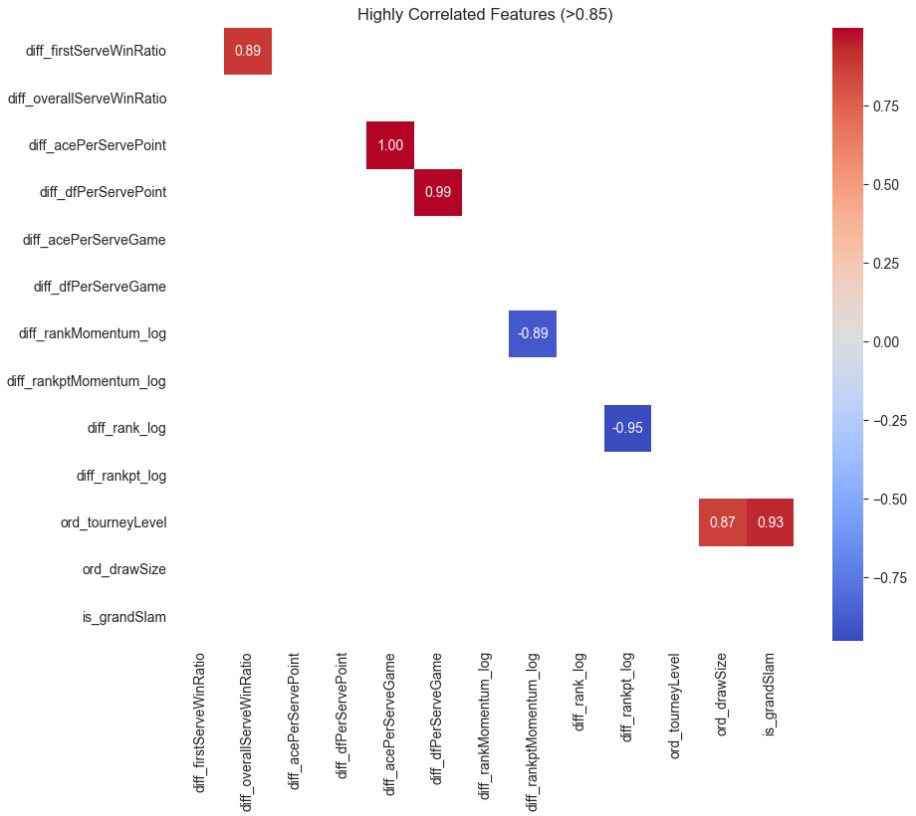

Figure 17: Feature correlation heatmap of highly correlated features (>0.85)

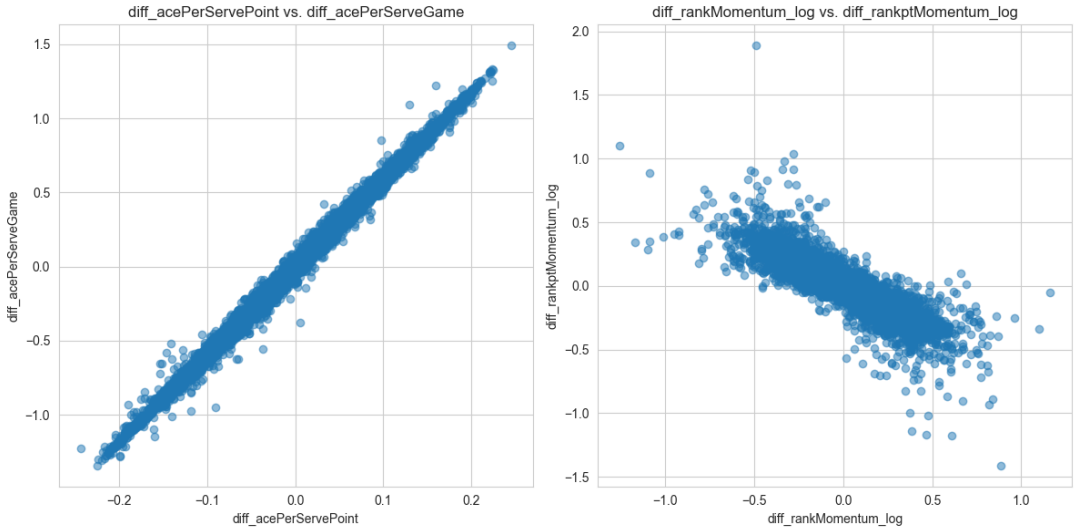

Figure 18: Scatter plot demonstration of two pairs of highly correlated features

Implementing Feature Selection Algorithms

After reducing feature redundancy and addressing multicollinearity, our feature space was narrowed to 39 candidates. With each feature having the option to be included or excluded, we faced over half a trillion (2^{39}) possible subsets, making exhaustive search computationally infeasible. We thus employed several heuristic-based methods to explore this space efficiently. Feature selection methods are typically classified into three main approaches and hybrids that combine these approaches: wrappers, which rely on the performance of the modeling algorithm; filters, which select features solely based on performance measures; embedded, which incorporate feature selection within the model's execution; and hybrid approaches [68].

In line with the SRP-CRISP-DM framework, which advocates for assessing various feature selection techniques alongside candidate classification models to find the most accurate combination, we specifically utilized Forward Selection (FS) from the wrapper approach and Mutual Information (MI) from the filter approach to identify our feature subsets:

1. **Forward Selection:** A wrapper method that begins with an empty model and sequentially adds features, each selected for its contribution to the model's performance until no improvement is gained by adding additional features.

   FS is a type of wrapper method used in feature selection. As a wrapper method, it involves a search algorithm that begins with an empty model and iteratively adds a feature. Each candidate subset of features is used to train a model, and its performance is then assessed on a validation set, using model performance as the evaluation criterion. The advantage of wrapper methods, such as FS, lies in their ability to find the best-performing feature set for a specific model, optimizing the feature space to enhance the model's predictive accuracy. However, one of the main drawbacks of wrapper methods is their tendency to overfit the model type, potentially leading to feature subsets that do not generalize well across different models. Furthermore, wrapper methods are computationally intensive due to the necessity to train and evaluate numerous models, each with a different subset of features [68] [69] [70].

2. **Mutual Information:** A filter method that quantifies the dependency between variables. It calculates the MI value for each independent variable in relation to the dependent variable, selecting those with the highest information gain. In MI, a higher score indicates a stronger dependency between the feature and the target.

   Filter methods play a critical role in feature selection. These methods evaluate the relevance of each feature by analyzing its statistical relationship with the target variable. They operate independently of any ML model, making them faster and more computationally efficient compared to other methods like wrappers. By using statistical measures like MI score, filter methods provide a preliminary yet powerful way to screen for features that have a meaningful connection with the target variable. This model-agnostic approach ensures that the selected features are not tailored to any specific algorithm, thereby reducing the risk of overfitting, enhancing the generalizability of the model, and making the method suitable for high-dimensional space. However, a limitation of filter methods is their tendency to evaluate features in isolation, which might lead to the omission of features that, while not strongly correlated with the target individually, could be significant predictors in combination with others [71] [72].

3.5 Model Evaluation

Model Evaluation is the conclusive phase where the performance of the models is rigorously assessed using a range of measures and techniques, validating the effectiveness of the predictive analysis [17].

3.5.1 Performance Measure Selection

The selection of appropriate performance measures is critical for a fair and comprehensive evaluation of model efficacy. Our overall goal is to maximize the proportion of correctly predicted match outcomes. Consequently, the following performance metrics were utilized to effectively measure and compare the performance of the predictive models:

1. **Classification Accuracy:** To evaluate model performance, match results are classified into player 1 wins and player 2 wins. Classification accuracy is then determined by the number of matches that the model correctly identified using a standard classification matrix. We utilized classification accuracy as one of our performance metrics, as our classes are balanced, having $17,107$ (50.1%) matches won by player 1 and $17,014$ (49.9%) matches won by player 2.

   Classification accuracy is defined as the ratio of correctly predicted observations to the total observations. It is expressed as a percentage and calculated using the formula:

   $$\textit{Classification Accuracy} = \frac{\textit{Number of Correct Predictions}}{\textit{Total Number of Predictions}} \times 100$$

   While classification accuracy is useful for comparing models, it does not measure how far the predicted probabilities are from the actual class labels. For that purpose, we need a measure that penalizes confident and wrong predictions more than less confident ones. Recognizing this, we expanded our evaluation to include the Brier Score (BS) metric, which provides insights into the probabilistic accuracy of predictions.

2. **Brier Score:** The BS, also known as the MSE, measures the mean squared difference between predicted probabilities and the observed outcomes. BS is useful when the interest is on well-calibrated probability estimates. It is a proper scoring rule, meaning that it is optimized when the predicted probabilities are close to the true probabilities of the events [73]. BS always takes on a value between 0 (best value) and 1 (worst value) since this is the largest possible difference between a predicted probability (which must be between 0 and 1) and the actual outcome (which can take on values of only 0 and 1). BS is similar to logistic loss because it uses probabilities to measure the confidence of the models. Contrarily, BS is easier to interpret since its formula is easier and it ranges between 0 and 1, while logistic loss does not have an upper bound.

3.5.2 Cross Validation

Cross-validation is a technique used to assess the generalizability and consistency of a model's performance across different subsets of data. In the context of sports prediction, where models are trained on historical matches to predict future outcomes, maintaining the chronological order of data is crucial. Traditional cross-validation methods, which often involve random shuffling of data, are inappropriate

for this domain as they could inadvertently leak future information into the training set. Instead, time-based cross-validation is employed, ensuring that predictions for upcoming matches are based solely on past data. This approach aligns with the SRP-CRISP-DM framework as the framework strongly advises preserving the order of the training data for the sports prediction problem [17].

In our study, we utilized time-based cross-validation by incrementally training our models on early years and validating on subsequent years. Specifically, our initial training set comprised the years 2009 and 2010, with subsequent years added incrementally to the training set, and each following year serving as the validation set. For instance, models were trained on data from 2009-2010 and validated on 2011, then trained on 2009-2011 and validated on 2012, and so forth, up to training on 2009-2019 and validating on 2020. This methodology resulted in 10 unique train-validation splits, as illustrated in Figure 19. For each split, we calculated classification accuracy and BS, and then averaged these metrics across all splits to derive an overall measure of model performance.

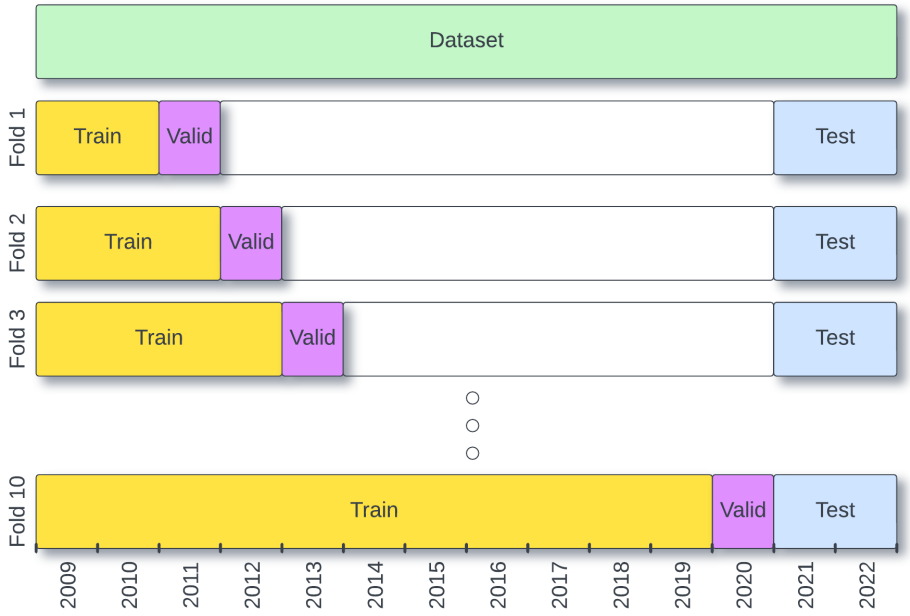

Figure 19: Time-based cross-validation splits for model training, validation, and testing

3.5.3 Hyperparameter Tuning

Hyperparameter tuning is essential in optimizing an ML model's performance. Unlike model parameters learned during training, hyperparameters are predefined by the user and play a crucial role in determining the model's architecture and behavior. Effective tuning of these parameters can significantly enhance model accuracy and generalization capabilities.

One common approach to hyperparameter tuning is the Grid Search method, which systematically explores a range of specified hyperparameter values to find the combination that yields the best per-

formance on a validation set. This exhaustive search, although computationally demanding, ensures a thorough examination of the parameter space [74].

In our study, we employed Grid Search during the hyperparameter tuning phase, particularly in conjunction with the FS step of the feature selection. The hyperparameters optimized for each algorithm were as follows:

- **LR:** *penalty* (type of regularization, e.g., *l1*, *l2*), *C* (regularization strength, controlling the degree of penalization), *solver* (algorithm used in the optimization problem), *max_iter* (maximum number of iterations taken for the solvers to converge).
- **SVM:** *C* (regularization parameter, influencing the trade-off between smooth decision boundary and classifying training points correctly), *kernel* (type of kernel, e.g., linear and poly), *degree* (degree of the polynomial kernel function), *gamma* (kernel coefficient).
- **XGB:** *learning_rate* (step size shrinkage used to prevent overfitting), *max_depth* (maximum depth of the trees), *min_child_weight* (minimum sum of instance weight needed in a child), *subsample* (ratio of the training instances to sample), *colsample_bytree* (subsample ratio of columns when constructing each tree), *alpha* (*l1* regularization on weights), *lambda* (*l2* regularization on weights), and *n_estimators* (number of gradient boosted trees).

3.6 Model Deployment

The Model Deployment phase, typically characterized by the implementation of a predictive model within an operational environment [17], is not applicable in the context of our study due to the nature of our dataset assembly process. Our dataset, derived from two distinct sources with different feature conventions, requires extensive manual intervention to merge and prepare for analysis. This requirement for manual supervision rules out the possibility of automating the data preparation process, which is a fundamental prerequisite for the deployment of a model.

Moreover, the scope of our research does not extend to real-time prediction or continuous model retraining with new data. The models developed contain parameters that are products of rigorous experimentation, and they are constructed to provide insights and validate hypotheses on the historical data available rather than being applied for ongoing predictive tasks. Therefore, the deployment in a traditional sense, where a model is continuously updated and used for decision-making, is outside the purview of this study.

CHAPTER 4

EXPERIMENTS

Building upon the structured experimental methodology outlined in Chapter 3, this section delves into the specific experiments conducted as part of this thesis. Here, we present the empirical processes and detailed results derived from extensive testing and iterative refinement within the Feature Extraction, Modeling, and Model Evaluation phases of the SRP-CRISP-DM framework [17]. This section explores the experimental outcomes in-depth, highlighting the iterative processes underpinning the predictive models developed for tennis match outcome prediction. Our focus is to present a comprehensive account of the experimental setups, data manipulations, model comparisons, and in-depth analysis of results, thereby offering a transparent and detailed view of the empirical work that substantiates the findings of this thesis.

4.1 Experimental Setup

This section outlines the experimental setup of the thesis, grounded in the SRP-CRISP-DM framework, adapted for the unique demands of predictive modeling in individual sports, particularly men's singles tennis. The core objective is to develop replicable and reproducible predictive models that are robust across diverse datasets and sports types. The methodology is structured methodically, encompassing various phases: Domain Understanding, Data Understanding, Data Preparation and Feature Extraction, Modeling, Model Evaluation, and Model Deployment [17]. Figure 20 illustrates the general structure of our experimental setup.

The Domain Understanding phase involves a thorough analysis of tennis as a sport, considering elements like match rules, player characteristics, and critical factors influencing match outcomes.

Data Understanding focuses on compiling a comprehensive dataset, considering aspects like dataset assembly, granularity, and class variable selection. This phase includes intricate steps like combining two datasets with different data recording conventions, excluding certain years' data for consistency and targeting specific match types to ensure data homogeneity.

The subsequent Data Preparation and Feature Extraction phase involves categorizing the dataset features, addressing outlier and missing data, extracting new features to enhance the model's predictive capacity, and undertaking essential data preparation steps, including data validation checks, train-test splitting, feature encoding, and feature scaling.

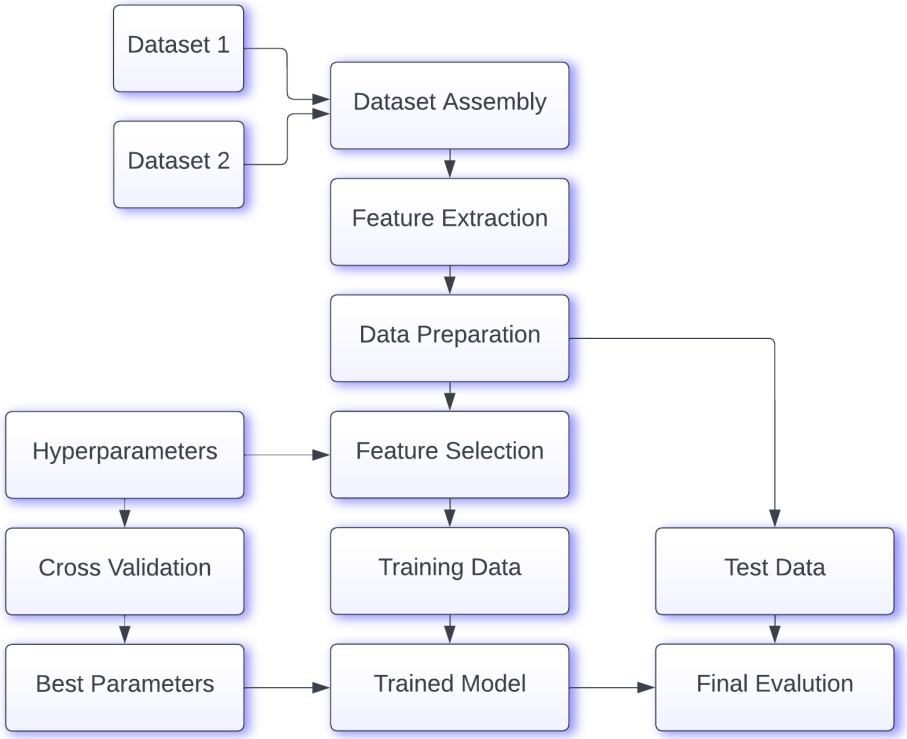

Figure 20: Overview of the thesis experimental framework

The Modeling phase includes careful selection and application of ML models based on a thorough literature review and a rigorous process of feature selection to refine the dataset by choosing the most impactful features, thereby ensuring the model's accuracy and efficiency.

The Model Evaluation phase employs a range of performance metrics and validation techniques, further strengthened by cross-validation and hyperparameter tuning, to underpin the models' accuracy and generalizability.

Finally, Model Deployment is not pursued in this thesis due to the nature of the study's focus on manual dataset assembly and analysis rather than real-time or continuous prediction models.

The experiments were conducted using Python 3.10. Python libraries such as Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn, SciPy, XGBoost, and MLxtend were utilized for the various stages of the study, including data processing, model development, and visualization. The computational experiments were primarily executed on a Macbook Pro M2 with 8GB RAM. Additionally, Google Colab's TPU environment was utilized for more resource-intensive tasks.

This experimental setup, while comprehensive in its approach to predictive modeling, differentiates between general procedures and detailed empirical outcomes. The detailed testing, iterative refinements, and empirical results associated with Feature Extraction and Feature Selection are given in the following sections.

4.1.1 Feature Extraction Application

The feature extraction step required experimentation and rigorous iteration through hyperparameters to identify the set that yields the highest performance score. Due to the vast computational resources needed to iterate over feature extraction hyperparameters, along with adjusting feature subsets and ML model hyperparameters, such an endeavor was practically unattainable. Therefore, our focus shifted to tuning the hyperparameters of optional feature extraction techniques and evaluating their effectiveness via χ^2 scores. Table 4 presents the optimal hyperparameter settings of the optional applications with their corresponding χ^2 scores and p-values. A detailed listing of the top 10 performing hyperparameter combinations for each extracted feature is provided in Appendix A.

Table 4: Summary of the Techniques Utilized for each Newly Constructed Feature

Feature	**SR**	**HA**	**CO**	**TD(t)**	**TD(f)**	**TF**	**UA**	**χ^2 Score**	**p-value**
diff_rank/rankpt	+	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
diff_age/ht/seed	+	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
diff_agePersonalPeak	+	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
diff_ageGeneralPeak	+	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
diff_homeAdvantage	+	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
diff_Avg_implied	+	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
is_formerTop10	n/a	n/a	n/a	n/a	n/a	15	n/a	1.534	0.22
diff_winLossStreak	+	n/a	n/a	n/a	n/a	365*20	n/a	1.363	0.24
diff_gamesFatigue	+	+	−	1	0.99	5	10.0	1.763	0.18
diff_surfaceAdvantage	+	+	−	30	0.60	365	1.0	15.950	<0.01
diff_rankMomentum_log	+	+	−	90	0.99	365*2	1.0	2.162	0.14
diff_rankptMomentum_log	+	+	−	30	0.80	365	1.0	0.233	0.63
diff_avgDuration	+	+	−	90	0.80	365*20	1.0	0.550	0.46
diff_surfaceWinRatio	+	+	+	365	1.00	365*3	5.0	92.293	<0.01
diff_overallWinRatio	+	+	+	365	0.95	365	5.0	103.352	<0.01
diff_h2hWinRatio	+	+	−	180	0.90	365*20	100.0	138.311	<0.01
diff_firstServeInRatio	+	+	−	90	0.80	365*20	0.2	3.212	0.07
diff_firstServeWinRatio	+	+	+	90	0.90	365*3	0.2	28.447	<0.01
diff_secondServeWinRatio	+	+	+	365	0.90	365*3	0.2	21.742	<0.01
diff_overallServeWinRatio	+	+	+	90	0.80	365*20	0.2	44.552	<0.01
diff_overallReturnWinRatio	+	+	+	90	0.80	365*20	0.2	3.698	0.05
diff_completeness	+	+	+	365	0.90	365*20	0.2	4.333	<0.05
diff_acePerServePoint	+	+	−	90	0.80	365*3	0.2	14.006	<0.01
diff_dfPerServePoint	+	+	+	365	0.95	365*20	0.2	3.864	<0.05
diff_acePerServeGame	+	+	−	30	0.95	365*3	5.0	11.402	<0.01
diff_dfPerServeGame	+	+	+	365	0.95	365*3	0.2	4.686	<0.05
diff_bpSaveRatio	+	+	+	365	0.80	365*3	0.2	1.066	0.30

n/a: Not applicable
SR: Symmetric Representation (+: applied, −: not applied)
HA: Historical Averaging (+: applied, −: not applied)
CO: Common Opponents (+: applied, −: not applied)
TD: Time Discount (t: discount period, in days, f: discount factor, between 0 and 1)
TF: Time Frame Selection (in days)
UA: Uncertainty Assessment (uncertainty threshold, between 0 and $+\infty$)

In our analysis, while the χ^2 test is typically employed for categorical data, we adapted it for numerical features to evaluate their association with the match outcomes. This was accomplished by discretizing the numerical values into categorical bins, thus transforming continuous variables into ordinal categories. Specifically, we divided each of the newly constructed features into six bins. These bins were chosen based on meaningful thresholds that preserved the integrity of the original numerical

data, reflecting the natural distribution and potential breakpoints within each feature. This binning process allowed us to apply the χ^2 test to assess whether the distribution of data across these bins was independent of the target variable.

Newly constructed features such as `match_id`, `is_grandSlam`, `favorite_won`, `handedness`, `p1_Avg`, `p1_points`, `p1_games`, and `p1_sets` cannot utilize any of the feature extraction techniques by design. Consequently, they were removed from the analysis. Similarly, `diff_age`, `diff_ht`, `diff_seed`, `diff_agePersonalPeak`, `diff_ageGeneralPeak`, `diff_homeAdvantage`, and `diff_Avg_implied` do not integrate historical data and are therefore only subject to Symmetrical Representation, with no hyperparameter tuning involved, hence no χ^2 scores were computed.

The feature `is_formerTop10` is exclusively associated with Time Frame Selection, while `diff_winLossStreak` combines Symmetric Representation and Time Frame Selection by design. A time frame described as "$365 * 20$" encompasses our entire dataset, a strategic choice to sidestep complications from leap years and other calendar variations.

An f-value of 1 indicates the removal of the Time Discount technique, reflecting its negligible impact on feature enhancement. For variables such as `diff_gamesFatigue`, which reflect recent player performance, our grid search focused on a short-term span of 1 to 14 days to emphasize immediate impacts. In contrast, the time frame for other features extended up to the full dataset range of $365 * 20$ days, ensuring comprehensive temporal coverage.

An increase in the uncertainty threshold that corresponds with better χ^2 scores implies that more data points enhance predictive power. For instance, `diff_h2hWinRatio` recorded optimal hyperparameters at an uncertainty threshold of 100, reflecting the premise that a more extensive collection of head-to-head matches can enhance prediction reliability. Conversely, most features peaked in χ^2 scores at an uncertainty threshold of 0.2. This indicates that a tighter selection of matches—ignoring statistics from less frequent matchups—tends to strengthen the association between the feature and the match outcome.

The strategic implementation of the Common Opponents Adjustment technique has generally led to a noticeable improvement in the χ^2 scores of most historical features derived from in-play statistics. This suggests that normalizing player performance against common opponents provides a more accurate reflection of their abilities, as it accounts for the quality of their opposition. Notably, there are exceptions such as `diff_firstServeInRatio`, and the ace-related features `diff_acePerServePoint` and `diff_acePerServeGame`, which demonstrated higher χ^2 scores when the adjustment was not applied. This indicates that for metrics reflecting personal competence, such as scoring aces or successfully landing the first serve, direct individual performance indicators may offer a more immediate correlation with match outcomes than metrics adjusted for opponent quality, suggesting that these particular elements of play may be less impacted by the caliber of the opposition.

Higher χ^2 values and lower p-values are preferred, as they suggest a strong and statistically significant association between the feature and the outcome variable, particularly when p-values fall below standard thresholds (e.g., 0.05). Features `diff_surfaceWinRatio`, `diff_overallWinRatio`, `diff_h2hWinRatio`, and `diff_overallServeWinRatio` stand out with exceptionally high χ^2 scores, whereas others vary between 0 to 40. Features like `diff_surfaceAdvantage` and `diff_acePerServeGame`, despite having p-values under 0.05, suggest that their influence on the outcome may be significant even if not highly impactful. Some other features, such as `diff_surfaceAdvan-`

`tage` and `diff_acePerServeGame` have p-values under 0.05, suggesting that even though their effect on the outcome variable is limited, their association is statistically significant.

In our analysis, the hyperparameter configurations yielding the highest χ^2 scores consistently corresponded to the lowest p-values. Therefore, these optimal settings were employed for feature computation, even in cases where the p-value exceeded the 0.05 threshold, relying on their enhanced predictive strength.

4.1.2 Feature Selection Application

This section provides a detailed overview of the features selected for the final models, including their statistical significance and overall contribution to the model's predictive capabilities. To determine the most predictive features, we employed MI as a filter method and FS as a wrapper method.

The objective was not to exhaustively search for the best combination of feature subsets, ML models, and ML hyperparameters due to the prohibitive computational demands. Instead, our FS approach omits cross-validation and operates within a limited hyperparameter space. In order to assess model performance, we adopted a train-validation split, designating data from 2009-2018 as the training set and 2019-2020 as the validation set. It is vital to note that the train-test split previously reserved the matches from the years 2021 and 2022 as a hold-out test set. This set is utilized only at the very end of the Modeling phase to compare performances across different ML models.

During the FS process, we incrementally added features to each of the three pre-selected ML models under a limited range of hyperparameters, tracking the BS of the validation set. As depicted in Figure 21, each model reached its lowest BS with a differing number of features, with these optimal counts highlighted in the figure. For instance, LR performed best with 13 features, SVM with 9, and XGB with 8.

In instances where BSs were similar across feature subsets, accuracy was the deciding factor. For example, in the LR model, BSs were comparable when including 13 and 16 features, but the accuracy substantially declined with the 14^{th} feature. Therefore, 13 was chosen as the optimal feature count for LR.

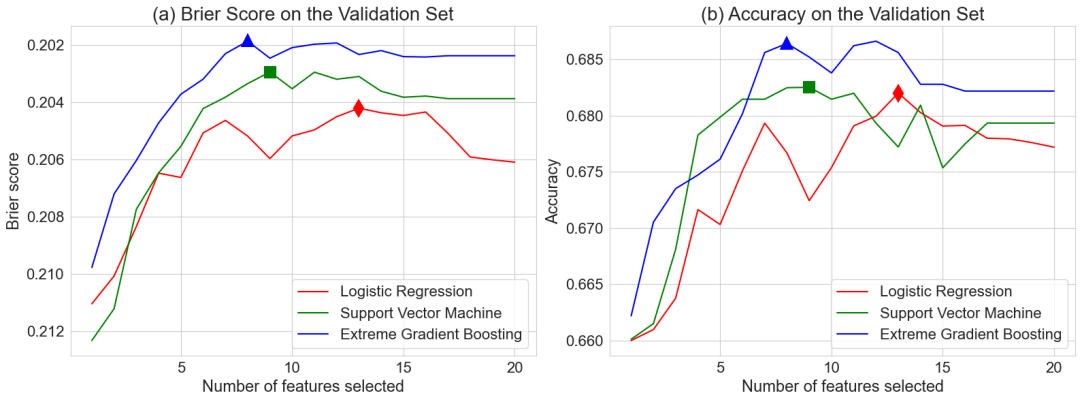

Figure 21: Model validation scores by number of features

It is important to note that while XGB consistently outperformed LR and SVM across all feature numbers, drawing definitive conclusions about overall model performance would be premature since our approach did not extend to cross-validation and was limited to a specific hyperparameter space. This heuristic and the resulting visualizations are intended solely for comparing validation set performances within the same ML model, not across different models.

It should be emphasized that while XGB consistently outperformed LR and SVM across all feature numbers, drawing definitive conclusions about overall model performance would be premature since our approach did not extend to cross-validation and was limited to a specific hyperparameter space. This heuristic and the resulting visualizations are intended solely for comparing validation set performances within the same ML model, not across different models.

In our application of the filter approach, a threshold of 0.010 for the MI score was chosen. Consequently, this methodical selection process identified 12 features that exceeded the set MI score threshold, indicating a strong link with the match outcomes and potential to improve model accuracy (see Table 5).

Table 5: Selected Feature Set of the Mutual Information Method

Feature	**MI Score**
diff_Avg_implied	0.135
diff_rankpt_log	0.093
diff_rankpt	0.091
diff_seed	0.074
diff_rank	0.072
diff_overallWinRatio	0.047
diff_surfaceWinRatio	0.046
diff_overallServeWinRatio	0.046
diff_secondServeWinRatio	0.031
diff_h2hWinRatio	0.016
diff_winLossStreak	0.013
diff_completeness	0.010

The final set of features was chosen based on their MI scores and BS across the selection methods, as summarized in Table 6. Notably, the average implied win probabilities, `diff_Avg_implied` emerged as the most significant across all methods. Other features like `diff_rankpt_log`, `diff_overallServeWinRatio`, `diff_surfaceWinRatio`, and `diff_winLossStreak` were consistently selected, though with varying significance levels. Interestingly, `diff_secondServeWinRatio`, `diff_completeness`, `is_grandSlam`, `diff_homeAdvantage`, and `diff_surfaceAdvantage` were identified by only one method each.

Table 6: Feature Ranks in Terms of Mutual Information and Forward Selection with Logistic Regression, Support Vector Machine, and Extreme Gradient Boosting Models

Features	MI	FS-LR	FS-SVM	FS-XGB
diff_Avg_implied	1	1	1	1
diff_rankpt	2	2		
diff_rankpt_log	3	9	3	2
diff_rank	4	3		5
diff_seed	5	5		
diff_overallWinRatio	6		7	
diff_overallServeWinRatio	7	4	4	3
diff_surfaceWinRatio	8	7	2	4
diff_secondServeWinRatio	9			
diff_winLossStreak	10	6	5	7
diff_h2hWinRatio	11	11	6	
diff_completeness	12			
diff_age		10		6
diff_gamesFatigue			8	
diff_agePersonalPeak		8		8
is_grandSlam			9	
diff_homeAdvantage		12		
diff_surfaceAdvantage		13		

4.2 Final Dataset

In an effort to capitalize on the distinct features for predicting tennis match outcomes, we merged two robust datasets detailed in Section 3.2.1. Subsequently, as outlined in Section 3.3.3.3, we engineered additional features to extract further predictive value from the data. The feature selection process, introduced in Section 3.4.2, was then applied to identify the most significant features. The composition of the finalized dataset, categorized by Match Characteristics, Player Characteristics, Betting Odds, Historical Statistics, and the Target variable, is outlined in Table 7.

Table 7: Summary of the Finalized Dataset by Category

Feature	Description	Category
year*	Year of the match	Match Characteristics
is_grandSlam	Indicator whether the match is a Grand Slam	Match Characteristics
diff_rankpt	Difference in players' ATP rank points	Player Characteristics
diff_rankpt_log	Logarithmic difference in players' rank points	Player Characteristics
diff_rank	Difference in players' ATP rankings	Player Characteristics
diff_seed	Difference in players' tournament seedings	Player Characteristics
diff_age	Difference in players' ages	Player Characteristics
diff_agePersonalPeak	Difference in players' personal peak performance ages	Player Characteristics
diff_homeAdvantage	Indicator if players are competing in their home country	Player Characteristics

Continued on next page

Table 7 continued from previous page

Feature	Description	Category
diff_Avg_implied	Difference in the average implied win probabilities	Betting Odds
diff_overallWinRatio	Difference in players' overall win ratios	Historical Statistics
diff_surfaceWinRatio	Difference in players' win ratios for the specific surface	Historical Statistics
diff_surfaceAdvantage	Players' comparative performance on current surface	Historical Statistics
diff_overallServeWinRatio	Difference in players' overall serve win ratios	Historical Statistics
diff_secondServeWinRatio	Difference in players' second serve win ratios	Historical Statistics
diff_winLossStreak	Difference in players' current win-loss streaks	Historical Statistics
diff_h2hWinRatio	Difference in players' head-to-head win ratio	Historical Statistics
diff_gamesFatigue	Difference in players' fatigue based on recent games	Historical Statistics
p1_won	Indicator whether player 1 won the match	Target variable

*Note: `year` is solely used for train-test split and cross-validation procedures.

4.3 Compared Models

Model comparison is an essential aspect of predictive analytics, allowing us to assess and compare the performance of different predictive models. This process is crucial for identifying the model that most accurately predicts tennis match outcomes based on BS and classification accuracy. In this study, we evaluated a total of nine models to ascertain their predictive efficacy on the hold-out test set.

4.3.1 Baseline Models

To provide benchmarks for our analysis, we established two baseline models using key features identified during feature selection:

- **Betting odds-implied (BOI) model:** Utilizing the `diff_Avg_implied` feature, this model assesses the predictive power of bookmakers' average implied probabilities. It serves as a comparison point for the more complex ML models. The probability of player 1 winning is calculated as:

$$\text{win probability of player 1} = \frac{1 + diff\_Avg\_implied}{2}$$

- **Rank points-implied (RPI) model:** Reflecting the significance of ATP rankings and rank points in our dataset, this model uses the `diff_rankpt` feature to predict outcomes. The higher-ranked player is favored to win, with equal rank points indicating an even chance of winning. The probability of player 1 winning is determined by:

$$\text{win probability of player 1} = \begin{cases} 1, & \text{if } diff\_rankpt > 0 \\ 0.5, & \text{if } diff\_rankpt = 0 \\ 0, & \text{if } diff\_rankpt < 0 \end{cases}$$

4.3.2 Machine Learning Models

In addition to the baseline models, our study also compared three distinct ML models: LR, SVM, and XGB. These models were selected, and their feature sets were obtained through MI and FS methods during the Modeling phase of the framework [17]. The lineup of ML models are as follows:

- **LR-MI model:** LR with features selected based on the MI method.
- **SVM-MI model:** SVM using MI for feature selection.
- **XGB-MI model:** XGB model incorporating features chosen via MI.
- **LR-FS model:** LR employing features identified through FS.
- **SVM-FS model:** SVM with feature selection driven by FS.
- **XGB-FS model:** XGB utilizing a feature set derived from FS.

4.4 Modeling

In this section, we compare the performance of the baseline and ML models using the BS and accuracy metrics. Each of the three pre-selected models was trained using two different feature sets and 10 time-based cross-validation folds, with hyperparameters tuned accordingly. Performance was then assessed on an independent hold-out test set.

4.4.1 Model Comparison

As detailed in Section 3.5.2, our validation folds were determined by using years as the cut-off points. This approach resulted in an inadvertent weighting discrepancy due to the uneven distribution of matches across years, most notably in 2020. The COVID-19 pandemic significantly reduced the number of matches to only 1237, in stark contrast to the approximately 2500 matches in other years. Consequently, the 2020 matches are effectively given double the importance within our cross-validation process. To ensure comparability, we applied the same methodology when calculating performance metrics for the baseline models. While not ideal, this approach was necessary to maintain consistency across the evaluation of all models.

The BOI model achieves a BS of 0.1957 and an accuracy of 0.6940 in the validation set by merely using the difference in average implied win probabilities of six bookmakers of two competing players as the predictor. The RPI model achieves a BS of 0.3396 and an accuracy of 0.6604 in the validation set by merely using the difference in ATP rank points of two competing players as the predictor. These baseline models provide competitive benchmarks for the ML models, in line with literature that acknowledges the strong predictive power of betting odds, which incorporate bookmakers' comprehensive pre-match evaluations. As Table 8 depicts, the BOI model severely outperforms the RPI model, in line with literature that acknowledges the strong predictive power of betting odds, which incorporate bookmakers' judgment on a tennis match right before the match starts [2] [21]. These baseline models provide competitive benchmarks for the ML models.

As elaborated in detail in the earlier sections of this thesis, three selected ML models (i.e., LR, SVM, and XGB) were trained using numerous hyperparameter settings and two separate feature sets derived by utilizing two feature selection algorithms (i.e., MI and FS). Later, these six distinct models were cross-validated on 10 time-based folds. Finally, one of each of these six models was chosen as the best performer based on the average BS on 10 validation folds. To illustrate the issue on the XGB model, we trained models on 10 cross-validation segments, using 2 different feature sets, and 4 *learning_rate*, 3 *max_depth*, 3 *min_child_weight*, 3 *subsample*, 3 *colsample_bytree*, 3 *alpha*, 3 *lambda*, and 3 *n_estimators* settings. In total, we trained $174,960$ (10 x 2 x 4 x 3 x 3 x 3 x 3 x 3 x 3 x 3) models to merely find the optimally performing XGB model on our dataset. This exhaustive approach was mirrored for the LR and SVM models, with the top 10 performers listed in Appendix B.

As a result, the best-performing ML models—LR-MI, SVM-MI, XGB-MI, LR-FS, SVM-FS, and XGB-FS—displayed validation BS of 0.1905, 0.1916, 0.1905, 0.1906, 0.1904, and 0.1902, respectively, with corresponding accuracies of 0.7080, 0.7075, 0.7028, 0.7076, 0.7078, and 0.7077 (see Table 8). Notably, XGB-FS achieved the best (smallest) cross-validation BS with 0.1902. LR-MI achieved the best (highest) cross-validation accuracy with 0.7080. On the other hand, SVM-FS became the second in both BS and accuracy with 0.7078 and 0.1904, respectively. All ML models showed superior performance compared to the baseline models, indicating the effectiveness of more complex predictive algorithms in capturing the nuances of tennis match outcomes.

Table 8: Train and Validation Set Performances of the Compared Models

Models	**Train**		**Validation**	
	Brier score	**Accuracy**	**Brier score**	**Accuracy**
BOI	-	-	0.1957	0.6940
PRI	-	-	0.3396	0.6604
ERS	-	-	0.2203	0.6732
LR-MI	0.1867	0.7139	0.1905	**0.7080**
SVM-MI	0.1883	0.7119	0.1916	0.7075
XGB-MI	0.1828	0.7241	0.1905	0.7028
LR-FS	0.1863	0.7156	0.1906	0.7076
SVM-FS	0.1874	0.7135	**0.1904**	**0.7078**
XGB-FS	0.1829	0.7208	**0.1902**	0.7077

When it came to feature selection techniques, MI proved to be remarkably time-efficient compared to FS, with no significant performance loss, making it the preferred method in time-sensitive or computationally restricted scenarios. However, in cases where computational resources are abundant or slight improvements in model performance are critical, the FS method could be of choice.

A noteworthy disparity was observed between the training and validation metrics of the XGB model. Overfitting occurs when a model learns the training data too well, including its noise and outliers, leading to poor generalization on unseen data, and XGB's capacity to capture complex relationships between features demands careful hyperparameter tuning to prevent overfitting. This was mitigated by constraining parameters such as tree depth and minimum child weight. On the contrary, the LR model exhibited close proximity between its training and validation metrics, highlighting its limitations in deciphering complex relationships. Therefore, hyperparameter tuning, while beneficial, is less critical for LR than for XGB.

All in all, we selected XGB-FS as our model of choice, prioritizing a lower BS over accuracy. Selecting XGB-FS as the best-performing model based on its cross-validation BS of 0.1902, despite its slightly lower accuracy (0.7077) compared to that of LR-MI (0.7080), underscores the importance of generalizability in model selection. The expectation is that XGB-FS, with its superior BS, would demonstrate better performance in real-world scenarios, aligning with our focus on men's singles tennis.

4.4.2 Final Evaluation

In the final evaluation stage, we examined the generalizability and real-world efficacy of the top three ML models—LR-MI, SVM-FS, and XGB-FS—using a hold-out test set that was not exposed to the models during the training phase. The test set performances are presented in Table 9, where a modest decrement of approximately 0.3% in accuracy is observed when compared with the validation set performance. This minor deviation in performance is likely attributable to temporal changes in player fitness and play styles or could simply be a matter of random variation.

Notably, the LR-MI and XGB-FS models demonstrated remarkable consistency, with minimal variation between validation and test performance metrics. The consistency across both BS and accuracy metrics across validation and test sets reinforces the robustness of the selected model. The implications of these findings are significant, especially when applied to domains like sports betting, where even a slight edge in prediction accuracy can translate to substantial financial outcomes. As such, these results validate that our carefully adjusted models can reliably predict tennis match outcomes, even though these outcomes inherently involve lots of randomness.

Table 9: Test Set Performances of the Top Performing ML Models

Models	**Validation**		**Test**	
	Brier score	**Accuracy**	**Brier score**	**Accuracy**
LR-MI	0.1905	**0.7080**	**0.1913**	**0.7055**
SVM-FS	0.1904	0.7078	0.1920	0.7039
XGB-FS	**0.1902**	0.7077	**0.1913**	0.7050

4.4.3 Feature Importance on the Selected Model

In the selected XGB-FS model, a set of eight features was determined as significant through the FS method. Figure 22 presents a visual representation of the relative importance of these features as utilized by the model for predictive inferences. Predominantly, the feature representing the average implied betting odds, labeled as `diff_Avg_implied`, stands out with an extraordinary importance score of 0.743, signifying its substantial influence on the model's decisions. This is a compelling insight as it underscores the predictive power of betting odds in tennis match outcomes, which is consistent with the BOI baseline model's performance, being less accurate by only 1.4% compared to other ML models. The remaining features exhibit relatively similar importance scores, ranging from 0.042 to 0.026. Notably, the age difference between players, `diff_age`, emerges as the next significant feature.

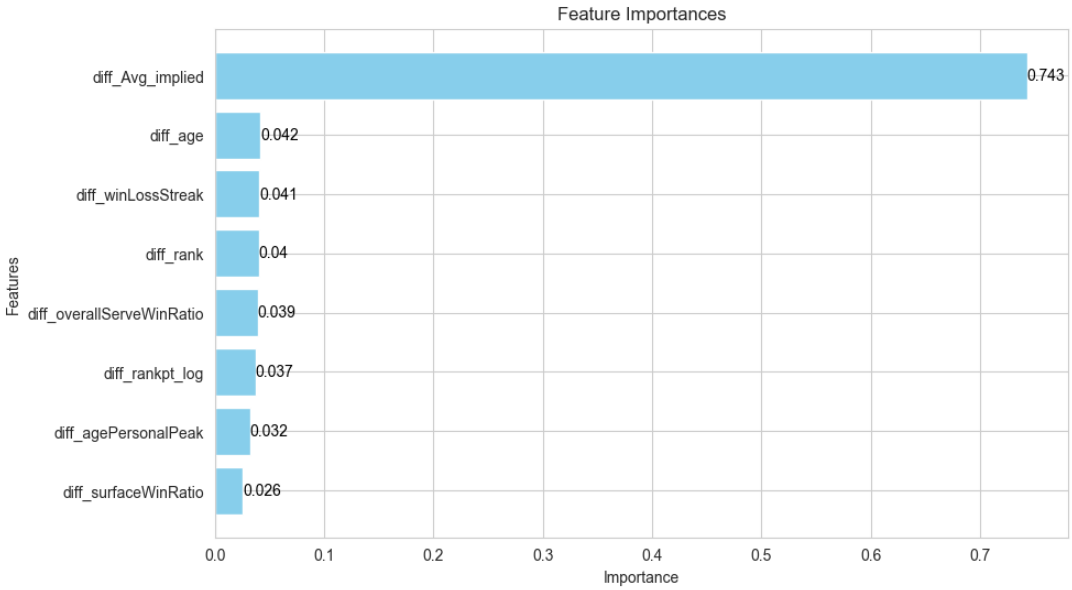

Figure 22: Feature importances of the XGB-FS model

4.4.4 Hyperparameter setting of the Selected Model

In the context of our XGB-FS model, the hyperparameter values selected through Grid Search, as detailed in Table 10, were chosen based on their impact on model performance and efficiency. The learning rate of 0.05 ensures gradual and more robust learning, minimizing the risk of overfitting. A *max_depth* of 3 helps in keeping the model sufficiently complex while avoiding excessive depth that could lead to overfitting. The *min_child_weight* of 4 provides a balance between underfitting and overfitting. The subsample and *colsample_bytree* values of 0.5 and 1, respectively, are set to optimize the number of samples and features used for training each tree, ensuring diversity in the model while maintaining the full feature set. Regularization terms *alpha* and *lambda* were set to 0, reflecting a decision to minimize the regularization effect in this specific context. Lastly, the *n_estimators* value of 50 was found to be optimal for balancing model complexity. The rationale behind these settings, derived from the Grid Search strategy, involved balancing the trade-off between model accuracy and overfitting while also considering computational constraints.

Table 10: Hyperparameter Setting of the XGB-FS Model

Hyperparameter	**Value**
learning_rate	0.05
max_depth	3
min_child_weight	4
subsample	0.5
colsample_bytree	1
alpha	0
lambda	0
n_estimators	50

CHAPTER 5

DISCUSSION AND FUTURE WORK

5.1 Discussion

In this study, we focused on enhancing the predictive accuracy of models in the men's singles tennis domain, utilizing a structured and methodical approach to ML supported by the SRP-CRISP-DM framework. This framework was specifically chosen to navigate the complexities of sports prediction, emphasizing the need for replicable and reproducible results across diverse datasets and sports types. Additionally, to address the rapid changes in player form and varying external match conditions, we incorporated six sophisticated feature extraction techniques. This strategy was specifically designed to tackle the challenges commonly faced in tennis match outcome predictions, making it highly relevant for men's singles tennis, a field ripe for advancements in sports analytics.

Our research spanned a comprehensive 14-year dataset of men's singles tennis matches from 2009 to 2022. We split data from the last two years, 2021 and 2022, as our hold-out test set. To refine our predictive models, we applied an array of six advanced feature extraction techniques complemented by three ML models and two distinct feature selection methods. We trained and validated our models using a 10-fold time-based cross-validation scheme while also further enhancing their predictive capabilities by hyperparameter tuning. After extensive training and tuning, the XGB model achieved the most favorable results with a BS of 19.13% and an accuracy rate of 70.5% on the test set. A key finding was the identification of the average win ratios implied by bookmakers' betting odds as the feature with the most significant predictive power.

To contextualize our research within the broader scope of the tennis match outcome prediction field, it is essential to consider the findings of other pivotal studies in this field:

- Somboonphokkaphan et al. [75] trained an ANN using match surface and multiple player-specific features as training parameters, claiming an accuracy ranging between 67% and 81% for Grand Slam matches in 2007 and 2008.
- Del Corral and Prieto-Rodriguez [6] developed an LR model that achieved a Brier Score of 17.5% for the Australian Open matches in 2009.
- Lisi and Zanella [32] trained an LR model, attaining a Brier Score of 16.5% across the four Grand Slam Championships in 2013.
- Cornman et al. [7] explored various models, including LR, SVM, ANN, and RF, and reported an approximate accuracy of 70% for professional tennis matches in 2016 and 2017.

- De Araujo Fernandes [76] utilized multiple ANNs, combining them with a majority vote model along with two other approaches, achieving around 70% accuracy for ATP matches and 75 - 80% for Grand Slam matches in 2015. Notably, forecasts based on betting odds were found to be more accurate than those using ranking information alone.
- Chavda et al. [77] applied LR, DT, and GB models, with the GB model achieving roughly 75% accuracy using men's US Open matches from 2000 to 2015 as the training set and 2016 as the test set.
- Gu and Saaty [33] combined data and expert judgments using an analytical network process model, reporting a prediction accuracy of about 85%, though for a limited sample of fewer than 100 matches.
- Gao and Kowalczyk [34] implemented LR, SVM, and RF models on ATP matches from 2000 to 2016, observing an accuracy of about 83% using RF, compared to 69% when relying solely on betting odds.
- Ghosh et al. [34] trained DT and SVM models on men's and women's Grand Slam matches in 2013, claiming a remarkable prediction accuracy of 92% to 99%, with DT showing superior performance.
- Kovalchik [2] examined the predictive efficacy of 11 published forecasting models for 2395 singles matches during the 2014 ATP season, finding that FiveThirtyEight's Elo model was the most successful, reported to achieve 75% accuracy for high-ranked players, while 59% to 64% for lower-ranked players.
- In a more recent study, which stands out as one of the most comprehensive in the field of tennis analytics, Wilkens [21] applied a range of ML models to both male and female professional singles matches. The paper shows that the average prediction accuracy cannot be increased to more than about 70%. Irrespective of the used model, most of the relevant information is embedded in the betting markets, and adding other match- and player-specific data does not lead to any significant improvement.

Comparing these findings with existing literature, our study aligns with the broader consensus regarding the efficacy of ML in sports outcome prediction. The advanced feature extraction techniques and the implementation of SRP-CRISP-DM provided a structured framework that proved effective in handling the complexities and dynamic nature of tennis matches.

5.2 Research Questions Addressed

In this section, we revisit and address the research questions identified in Section 1.2, providing a concise summary of our findings:

RQ1: What is the effectiveness of the Sports Result Prediction Cross-Industry Standard Process for Data Mining (SRP-CRISP-DM) framework when applied to the men's singles tennis domain?

The SRP-CRISP-DM framework proved effective in the men's singles tennis domain by enabling a streamlined integration of complex data, which includes real-time match conditions and player performance, into predictive models that are both replicable and reproducible. While traditionally used in

team sports, our methodology adapted and extended this framework to individual sports with a focus on men's singles tennis. This adaptation involved a thorough analysis of tennis specifics, including the sport's rules, data assembly, and granularity considerations.

In applying the SRP-CRISP-DM framework, we made three significant modifications to better suit our domain-specific needs. First, we treated both match-level and player-level data with equal granularity, acknowledging that individual player metrics are as critical as match-level outcomes in our analyses. Second, to prevent data leakage, we strategically positioned the train-test split before the Feature Encoding and Feature Scaling steps within the Data Preparation and Feature Extraction phase, diverging from the framework's usual sequence where it is situated during the Model Evaluation phase. Lastly, we introduced an iterative loop from the Model Evaluation back to the Data Preparation and Feature Extraction phase. This iteration underscores the importance of revisiting and refining data preparation and feature extraction processes based on the insights gained from performance metrics evaluation.

These tailored adjustments to the SRP-CRISP-DM framework were instrumental in our approach, allowing for the categorization of features, management of outliers, and dataset preparation for ML model application. The modeling phase was pivotal in applying these refined techniques to our dataset, leading to the selection of predictive models and feature sets. The evaluation phase enabled us to methodically assess the models' performance. These enhancements to the SRP-CRISP-DM framework have provided a comprehensive understanding of its application in predictive modeling for tennis match outcomes, proving its adaptability and effectiveness in individual sports analytics.

RQ2: How do the proposed six distinct feature extraction techniques from sports analytics literature contribute to the predictive modeling for men's singles tennis?

The six feature extraction techniques were systematically employed to deepen the model's understanding of each match's dynamics and player profiles. *Symmetric Representation* was vital in capturing data regarding both players in a match. By calculating differences in paired player data, this technique created a more informative and symmetric model, ensuring consistent predictions regardless of player positions. For example, subtracting the ATP rankings of one player from another provided a powerful predictor for match outcomes. *Historical Averaging* helped in leveraging historical data without causing data leakage. By aggregating past performance data, we were able to estimate a player's current form more accurately, which was crucial in predicting match outcomes. This approach was particularly beneficial for features like ATP rank points and player performance metrics. *Common Opponents Adjustment* refined our historical data by accounting for the variability in the skill levels of past opponents. This normalization process, crucial for fair player comparison, adjusted our model for the disparity in opponent skill levels. *Time Discounting* addressed the dynamic nature of a player's form by placing greater emphasis on recent matches. This technique weighted matches closer in time more heavily, ensuring that our model's predictions reflected current player form accurately. *Time Frame Selection* involved identifying the optimal historical data range for feature construction. By determining the best time frame for each feature, we ensured that our model was fed the most relevant and impactful data. *Uncertainty Assessment* was integrated into our methodology to measure the confidence in our predictions, considering the volume and significance of the data informing each feature. This helped in fine-tuning the model's sensitivity to the inherent variability and unpredictability of player performances.

Together, these feature extraction techniques provided a comprehensive and generalizable view of each player's capabilities and match conditions. By systematically applying and fine-tuning these

techniques, we were able to build a capable and robust model that could accurately predict outcomes in men's singles tennis.

RQ3: Can ML models provide more accurate forecasts than those based solely on players' official rankings or betting odds?

Based on our findings, ML models indeed provide more accurate forecasts than baseline models. The comparative analysis indicates that ML models, with their ability to integrate and analyze complex data patterns, can provide more generalizable and accurate forecasts in men's singles tennis matches compared to models relying solely on players' official rankings or betting odds.

RQ4: Which features are most impactful in predicting the outcomes of tennis matches?

The analysis of our ML models indicates that the feature representing the difference in average implied win probabilities by bookmakers is the most impactful for predicting tennis match outcomes. This implies that the collective judgment of bookmakers, which encapsulates a variety of factors, including past performance, player conditions, and public sentiment, is highly indicative of match results. Other relevant features include differences in players' ages, win-loss streaks, ATP rankings and rank points, and overall serve win ratio, all contributing to the model's predictive power, albeit to a lesser extent.

5.3 Key Contributions of the Study

This thesis makes the following notable contributions to the domain of men's singles tennis analytics, despite not achieving a higher predictive performance compared to existing models:

- **Comprehensive Dataset Utilization:** Utilizing two freely-accessible, regularly-updated, and reliable datasets spanning an extensive time frame of 14 years, this thesis stands out for its comprehensive approach to data analysis. This broad dataset not only provided a rich foundation for the study but also increased the generalizability of the findings.
- **Thorough Analysis of Tennis Specifics:** The study delved deep into the nuances of tennis, considering aspects like player performance, match conditions, and game dynamics. This thorough analysis forms the backbone of the predictive models, ensuring they are well-informed and contextual.
- **Sophisticated Feature Extraction Techniques:** A significant achievement of this study is the development and application of six advanced feature extraction techniques. These techniques enhanced the predictive models' ability to capture the complex nature of tennis matches, providing a more nuanced and detailed analysis of player performance and match outcomes.
- **Enhancing Replicability and Reproducibility:** A key focus of this research was to enhance the replicability and reproducibility of predictive models in sports analytics. By adopting and extending a structured and transparent approach (via utilizing the SRP-CRISP-DM framework), this study sets a precedent for future research in the field, promoting a more rigorous and methodical application of machine learning techniques in sports prediction.
- **Insightful Comparative Analysis:** The study's comparative analysis of various machine learning models against traditional prediction methods provided valuable insights into the efficacy of

different approaches in sports prediction. This analysis is crucial for guiding future research and practice in tennis analytics.

In summary, while the existing predictive performance benchmarks were not surpassed, this thesis significantly contributes to the field by providing a robust, detailed, and methodical approach to sports analytics, particularly in the context of men's singles tennis. These contributions are instrumental in paving the way for future research that can build upon this foundation to achieve greater predictive accuracy and deeper analytical insights.

5.4 Limitations and Future Work

The primary limitations encountered in our study are as follows:

- **Computational Resource Constraints:** The extensive computational requirements for ML processes, which include iterations and experimentation in dataset selection, feature extraction, model and feature set selection, and hyperparameter tuning, were beyond our limited resources. Hence, we established certain thresholds and heuristic methods to confine the space for experimentation.
- **Data Source Constraints:** Our study was limited to two public datasets, precluding the incorporation of proprietary or restricted data sources, such as OnCourt[1] or HawkEye[2].

To address these limitations and expand the scope of research in tennis match outcome prediction, we suggest the following directions for future investigation:

- **Model Interpretability:** Future studies could focus on models with greater interpretability or employ techniques like SHAP values for post-hoc explanations to enhance transparency in predictions.
- **Error Analysis:** A comprehensive error analysis could uncover model biases towards specific player types or conditions, offering insights for corrective measures.
- **Granular Data Utilization:** Incorporating point-by-point in-play statistics, such as serve speed and stroke types, could provide a deeper understanding of match play and influence prediction models positively.
- **Technology Integration:** Leveraging data from advanced tennis technologies, like sensors and wearables, could enrich datasets and improve the precision of predictive models by capturing detailed aspects of match dynamics and player strategies.
- **Adoption of Alternative Ranking Metrics:** Integrating rankings from advanced paired comparison models may serve as a substitute for traditional ATP rankings, potentially elevating the precision of new predictive models.

[1] OnCourt is a software package providing extensive tennis data for various analysis purposes.
[2] HawkEye is a sophisticated analytics tool widely used for tracking ball trajectories and other match data.

- **Longitudinal Player Studies:** Investigating individual players over time can shed light on factors influencing career paths, such as performance peaks, aging, and strategic changes. These new features can be extracted/added into the feature set to increase predictive accuracy.
- **Incorporation of Qualitative Insights:** Leveraging Natural Language Processing to examine textual data from social media, interviews, and expert analyses could enrich the dataset with qualitative insights, adding another layer to the predictive framework.
- **Real-Time Predictive Systems:** Developing a production-ready predictive system with real-time data would test the best model's effectiveness and uncover the challenges of deploying such systems in practice.

REFERENCES

[1] WorldAtlas, "What are the most popular sports in the world?," 9 2023.

[2] S. A. Kovalchik, "Searching for the GOAT of tennis win prediction," *Journal of Quantitative Analysis in Sports*, vol. 12, no. 3, pp. 127–138, 2016.

[3] B. L. Boulier and H. O. Stekler, "Are sports seedings good predictors?: an evaluation," *International Journal of Forecasting*, vol. 15, no. 1, pp. 83–91, 1999.

[4] S. R. Clarke and D. Dyte, "Using official ratings to simulate major tennis tournaments," *International transactions in operational research*, vol. 7, no. 6, pp. 585–594, 2000.

[5] F. J. G. M. Klaassen and J. R. Magnus, "Forecasting the winner of a tennis match," *European Journal of Operational Research*, vol. 148, no. 2, pp. 257–267, 2003.

[6] J. del Corral and J. Prieto-Rodríguez, "Are differences in ranks good predictors for Grand Slam tennis matches?," *International Journal of Forecasting*, vol. 26, no. 3, pp. 551–563, 2010.

[7] A. Cornman, G. Spellman, and D. Wright, "Machine learning for professional tennis match prediction and betting," *Working Paper, Stanford University*, 2017.

[8] T. Barnett and S. R. Clarke, "Combining player statistics to predict outcomes of tennis matches," *IMA Journal of Management Mathematics*, vol. 16, no. 2, pp. 113–120, 2005.

[9] W. J. Knottenbelt, D. Spanias, and A. M. Madurska, "A common-opponent stochastic model for predicting the outcome of professional tennis matches," *Computers & Mathematics with Applications*, vol. 64, no. 12, pp. 3820–3827, 2012.

[10] S. Kovalchik and M. Reid, "A calibration method with dynamic updates for within-match forecasting of wins in tennis," *International Journal of Forecasting*, vol. 35, no. 2, pp. 756–766, 2019.

[11] I. McHale and A. Morton, "A Bradley-Terry type model for forecasting tennis match results," *International Journal of Forecasting*, vol. 27, no. 2, pp. 619–630, 2011.

[12] P. Gorgi, S. J. Koopman, and R. Lit, "The analysis and forecasting of tennis matches by using a high dimensional dynamic model," *Journal of the Royal Statistical Society Series A: Statistics in Society*, vol. 182, no. 4, pp. 1393–1409, 2019.

[13] R. D. Baker and I. G. McHale, "A dynamic paired comparisons model: Who is the greatest tennis player?," *European Journal of Operational Research*, vol. 236, no. 2, pp. 677–684, 2014.

[14] R. D. Baker and I. G. McHale, "An empirical Bayes model for time-varying paired comparisons ratings: Who is the greatest women's tennis player?," *European Journal of Operational Research*, vol. 258, no. 1, pp. 328–333, 2017.

[15] A. E. Elo and S. Sloan, *The rating of chessplayers: Past and present*. 1978.

[16] V. Candila and L. Palazzo, "Neural networks and betting strategies for tennis," *Risks*, vol. 8, no. 3, p. 68, 2020.

[17] R. P. Bunker and F. Thabtah, "A machine learning framework for sport result prediction," *Applied computing and informatics*, vol. 15, no. 1, pp. 27–33, 2019.

[18] J. M. Nicholson, M. Mordaunt, P. Lopez, A. Uppala, D. Rosati, N. P. Rodrigues, P. Grabitz, and S. C. Rife, "Scite: A smart citation index that displays the context of citations and classifies their intent using deep learning," *Quantitative Science Studies*, vol. 2, no. 3, pp. 882–898, 2021.

[19] B. Scheibehenne and A. Bröder, "Predicting Wimbledon 2005 tennis results by mere player name recognition," *International Journal of Forecasting*, vol. 23, no. 3, pp. 415–426, 2007.

[20] L. Vaughan Williams, C. Liu, L. Dixon, and H. Gerrard, "How well do Elo-based ratings predict professional tennis matches?," *Journal of Quantitative Analysis in Sports*, vol. 17, no. 2, pp. 91–105, 2021.

[21] S. Wilkens, "Sports prediction and betting models in the machine learning age: The case of tennis," *Journal of Sports Analytics*, vol. 7, no. 2, pp. 99–117, 2021.

[22] R. Wirth and J. Hipp, "CRISP-DM: Towards a standard process model for data mining," in *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, vol. 1, pp. 29–39, 2000.

[23] P. K. Newton and J. B. Keller, "Probability of winning at tennis I. Theory and data," *Studies in applied Mathematics*, vol. 114, no. 3, pp. 241–269, 2005.

[24] S. Kovalchik, "Extension of the Elo rating system to margin of victory," *International Journal of Forecasting*, vol. 36, no. 4, pp. 1329–1341, 2020.

[25] D. Spanias and W. J. Knottenbelt, "Predicting the outcomes of tennis matches using a low-level point model," *IMA Journal of Management Mathematics*, vol. 24, no. 3, pp. 311–320, 2013.

[26] M. Ingram, "A point-based Bayesian hierarchical model to predict the outcome of tennis matches," *Journal of Quantitative Analysis in Sports*, vol. 15, no. 4, pp. 313–325, 2019.

[27] Lyócsa and T. V\'yrost, "To bet or not to bet: a reality check for tennis betting market efficiency," *Applied Economics*, vol. 50, no. 20, pp. 2251–2272, 2018.

[28] C. Leitner, A. Zeileis, and K. Hornik, "Forecasting sports tournaments by ratings of (prob) abilities: A comparison for the EURO 2008," *International Journal of Forecasting*, vol. 26, no. 3, pp. 471–481, 2010.

[29] P. Robberechts and J. Davis, "Forecasting the FIFA World Cup–Combining result-and goal-based team ability parameters," in *Machine Learning and Data Mining for Sports Analytics: 5th International Workshop, MLSA 2018, Co-located with ECML/PKDD 2018, Dublin, Ireland, September 10, 2018, Proceedings 5*, pp. 16–30, 2019.

[30] J. Carbone, T. Corke, and F. Moisiadis, "The rugby league prediction model: Using an Elo-based approach to predict the outcome of National Rugby League (NRL) matches," *International Educational Scientific Research Journal*, vol. 2, no. 5, pp. 26–30, 2016.

[31] G. Angelini, V. Candila, and L. De Angelis, "Weighted Elo rating for tennis match predictions," *European Journal of Operational Research*, vol. 297, no. 1, pp. 120–132, 2022.

[32] F. Lisi, "Tennis betting: can statistics beat bookmakers?," *Electronic Journal of Applied Statistical Analysis*, vol. 10, no. 3, pp. 790–808, 2017.

[33] W. Gu and T. L. Saaty, "Predicting the outcome of a tennis tournament: Based on both data and judgments," *Journal of Systems Science and Systems Engineering*, vol. 28, pp. 317–343, 2019.

[34] S. Ghosh, S. Sadhu, S. Biswas, D. Sarkar, and P. P. Sarkar, "A comparison between different classifiers for tennis match result prediction," *Malaysian Journal of Computer Science*, vol. 32, no. 2, pp. 97–111, 2019.

[35] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, pp. 273–297, 1995.

[36] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[37] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?," *Advances in Neural Information Processing Systems*, vol. 35, pp. 507–520, 2022.

[38] M. Reid, M. Crespo, L. Santilli, D. Miley, and J. Dimmock, "The importance of the International Tennis Federation's junior boys' circuit in the development of professional tennis players," *Journal of sports sciences*, vol. 25, no. 6, pp. 667–672, 2007.

[39] R. H. Koning, "Home advantage in professional tennis," *Journal of Sports Sciences*, vol. 29, no. 1, pp. 19–27, 2011.

[40] D. Delen, D. Cogdell, and N. Kasap, "A comparative analysis of data mining methods in predicting NCAA bowl outcomes," *International Journal of Forecasting*, vol. 28, no. 2, pp. 543–552, 2012.

[41] C. S. Valero, "Predicting Win-Loss outcomes in MLB regular season games–A comparative study using data mining methods," *International Journal of Computer Science in Sport*, vol. 15, no. 2, pp. 91–112, 2016.

[42] T. Elfrink, "Predicting the outcomes of MLB games with a machine learning approach," *Vrije Universiteit Amsterdam*, 2018.

[43] R. J. A. Little and D. B. Rubin, *Statistical analysis with missing data*, vol. 793. John Wiley & Sons, 2019.

[44] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John wiley & sons, 2005.

[45] A. K. Waljee, A. Mukherjee, A. G. Singal, Y. Zhang, J. Warren, U. Balis, J. Marrero, J. Zhu, and P. D. R. Higgins, "Comparison of imputation methods for missing laboratory data in medicine," *BMJ Open*, vol. 3, no. 8, p. e002847, 2013.

[46] P. Royston, "Multiple imputation of missing values: update of ice," *The Stata Journal*, vol. 5, no. 4, pp. 527–536, 2005.

[47] J. S. Shah, S. N. Rai, A. P. DeFilippis, B. G. Hill, A. Bhatnagar, and G. N. Brock, "Distribution based nearest neighbor imputation for truncated high dimensional data with applications to pre-clinical and clinical metabolomics studies," *BMC Bioinformatics*, vol. 18, no. 1, pp. 1–13, 2017.

[48] J. Hauke and T. Kossowski, "Comparison of values of Pearson's and Spearman's correlation coefficients on the same sets of data," *Quaestiones geographicae*, vol. 30, no. 2, pp. 87–93, 2011.

[49] C. L. Parr, A. Hjartåker, I. Scheel, E. Lund, P. Laake, and M. B. Veierød, "Comparing methods for handling missing values in food-frequency questionnaires and proposing k nearest neighbours imputation: effects on dietary intake in the Norwegian Women and Cancer study (NOWAC)," *Public health nutrition*, vol. 11, no. 4, pp. 361–370, 2008.

[50] S. M. Fox-Wasylyshyn and M. M. El-Masri, "Handling missing data in self-report measures," *Research in nursing & health*, vol. 28, no. 6, pp. 488–495, 2005.

[51] A. J. O'Malley, "Probability formulas and statistical analysis in tennis," *Journal of Quantitative Analysis in Sports*, vol. 4, no. 2, 2008.

[52] M. Sipko and W. Knottenbelt, "Machine learning for the prediction of professional tennis matches," *MEng computing-final year project, Imperial College London*, vol. 2, 2015.

[53] D. Farrelly and D. Nettle, "Marriage affects competitive performance in male tennis players," *Journal of Evolutionary Psychology*, vol. 5, no. 1, pp. 141–148, 2007.

[54] R. L. Plackett, "Karl Pearson and the chi-squared test," *International statistical review/revue internationale de statistique*, vol. 51, no. 1, pp. 59–72, 1983.

[55] C. Martin and J. Prioux, "Tennis playing surfaces: The effects on performance and injuries," *Journal of Medicine and Science in Tennis*, vol. 21, no. 1, pp. 11–19, 2016.

[56] H. S. Shin, "Optimal betting odds against insider traders," *The Economic Journal*, vol. 101, no. 408, pp. 1179–1185, 1991.

[57] H. S. Shin, "Measuring the incidence of insider trading in a market for state-contingent claims," *The Economic Journal*, vol. 103, no. 420, pp. 1141–1153, 1993.

[58] K. Morita, T. Mizuno, and H. Kusuhara, "Investigation of a Data Split Strategy Involving the Time Axis in Adverse Event Prediction Using Machine Learning," 2022.

[59] A. Vakayil and V. R. Joseph, "Data Twinning," 2022.

[60] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[61] A. Agresti, *Categorical data analysis*, vol. 792. John Wiley & Sons, 2012.

[62] H. Weytjens and J. De Weerdt, "Creating unbiased public benchmark datasets with data leakage prevention for predictive process monitoring," in *International Conference on Business Process Management*, pp. 18–29, 2021.

[63] K. Koseler and M. Stephan, "Machine learning applications in baseball: A systematic literature review," *Applied Artificial Intelligence*, vol. 31, no. 9-10, pp. 745–763, 2017.

[64] Z. Gao and A. Kowalczyk, "Random forest model identifies serve strength as a key predictor of tennis match outcome," *Journal of Sports Analytics*, vol. 7, no. 4, pp. 255–262, 2021.

[65] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Transactions on Neural networks*, vol. 10, no. 5, pp. 1048–1054, 1999.

[66] O. N. Manjrekar and M. P. Dudukovic, "Identification of flow regime in a bubble column reactor with a combination of optical probe data and machine learning technique," *Chemical Engineering Science: X*, vol. 2, p. 100023, 2019.

[67] H. Lei, H. Zhao, and T. Ao, "A two-step merging strategy for incorporating multi-source precipitation products and gauge observations using machine learning classification and regression over China," *Hydrology and Earth System Sciences*, vol. 26, no. 11, pp. 2969–2995, 2022.

[68] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*, pp. 1200–1205, 2015.

[69] A. B. Gumelar, A. Yogatama, D. P. Adi, F. Frismanda, and I. Sugiarto, "Forward feature selection for toxic speech classification using support vector machine and random forest," *IAES International Journal of Artificial Intelligence*, vol. 11, no. 2, p. 717, 2022.

[70] J. Wieczorek and J. Lei, "Model selection properties of forward selection and sequential cross-validation for high-dimensional regression," *Canadian Journal of Statistics*, vol. 50, no. 2, pp. 454–470, 2022.

[71] J. Yang, Z. Qu, Z. Liu, and others, "Improved feature-selection method considering the imbalance problem in text categorization," *The Scientific World Journal*, vol. 2014, pp. 1–17, 2014.

[72] O. Almomani, "A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms," *Symmetry*, vol. 12, no. 6, p. 1046, 2020.

[73] J. E. Bickel, "Some comparisons among quadratic, spherical, and logarithmic scoring rules," *Decision Analysis*, vol. 4, no. 2, pp. 49–65, 2007.

[74] R. Ghawi and J. Pfeffer, "Efficient hyperparameter tuning with grid search for text categorization using kNN approach with BM25 similarity," *Open Computer Science*, vol. 9, no. 1, pp. 160–180, 2019.

[75] A. Somboonphokkaphan, S. Phimoltares, and C. Lursinsap, "Tennis winner prediction based on time-series history with neural modeling," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, pp. 18–20, 2009.

[76] M. De Araujo Fernandes, "Using soft computing techniques for prediction of winners in tennis matches," *Machine Learning Research*, vol. 2, no. 3, pp. 86–98, 2017.

[77] J. Chavda, N. Patel, and P. Vishwakarma, "Predicting tennis match winner and comparing bookmakers odds using machine learning techniques," *National College of Ireland, Dublin*, 2019.

APPENDIX A

FEATURE EXTRACTION HYPERPARAMETER TUNING

Table 11: Top 10 Grid Search Hyperparameter Settings for each Newly Constructed Feature

Feature	SR	HA	CO	TD(t)	TD(f)	TF	UA	χ^2 Score	p-value
is_formerTop10	n/a	n/a	n/a	n/a	n/a	15	n/a	**1.534**	**0.215**
is_formerTop10	n/a	n/a	n/a	n/a	n/a	30	n/a	0.980	0.322
is_formerTop10	n/a	n/a	n/a	n/a	n/a	365	n/a	0.747	0.388
is_formerTop10	n/a	n/a	n/a	n/a	n/a	180	n/a	0.685	0.408
is_formerTop10	n/a	n/a	n/a	n/a	n/a	60	n/a	0.569	0.451
is_formerTop10	n/a	n/a	n/a	n/a	n/a	730	n/a	0.487	0.485
is_formerTop10	n/a	n/a	n/a	n/a	n/a	90	n/a	0.464	0.496
is_formerTop10	n/a	n/a	n/a	n/a	n/a	1095	n/a	0.129	0.720
is_formerTop10	n/a	n/a	n/a	n/a	n/a	7300	n/a	0.003	0.954
is_formerTop10	n/a	n/a	n/a	n/a	n/a	1825	n/a	0.002	0.968
diff_winLossStreak	+	n/a	n/a	n/a	n/a	7300	n/a	**1.363**	**0.243**
diff_winLossStreak	+	n/a	n/a	n/a	n/a	1825	n/a	1.346	0.246
diff_winLossStreak	+	n/a	n/a	n/a	n/a	1095	n/a	1.286	0.257
diff_winLossStreak	+	n/a	n/a	n/a	n/a	730	n/a	1.182	0.277
diff_winLossStreak	+	n/a	n/a	n/a	n/a	365	n/a	0.995	0.318
diff_winLossStreak	+	n/a	n/a	n/a	n/a	90	n/a	0.729	0.393
diff_winLossStreak	+	n/a	n/a	n/a	n/a	180	n/a	0.698	0.403
diff_winLossStreak	+	n/a	n/a	n/a	n/a	60	n/a	0.609	0.435
diff_winLossStreak	+	n/a	n/a	n/a	n/a	30	n/a	0.230	0.632
diff_winLossStreak	+	n/a	n/a	n/a	n/a	15	n/a	0.013	0.909
diff_gamesFatigue	+	+	−	1	0.99	5	10.0	**1.763**	**0.184**
diff_gamesFatigue	+	+	−	1	0.99	5	4.0	1.763	0.184
diff_gamesFatigue	+	+	−	1	0.99	5	5.0	1.763	0.184
diff_gamesFatigue	+	+	−	1	0.99	5	3.0	1.763	0.184
diff_gamesFatigue	+	+	−	1	0.99	5	20.0	1.763	0.184
diff_gamesFatigue	+	+	−	1	0.99	5	2.0	1.763	0.184
diff_gamesFatigue	+	+	−	1	0.99	5	1.0	1.763	0.184
diff_gamesFatigue	+	+	−	1	0.90	5	1.0	1.693	0.193
diff_gamesFatigue	+	+	−	1	0.95	5	5.0	1.624	0.203
diff_gamesFatigue	+	+	−	1	0.95	5	20.0	1.569	0.210
diff_surfaceAdvantage	+	+	−	30	0.60	365	1.0	**15.950**	**0.000**
diff_surfaceAdvantage	+	+	−	30	0.60	365	0.5	15.864	0.000
diff_surfaceAdvantage	+	+	−	30	0.60	365	5.0	15.082	0.000
diff_surfaceAdvantage	+	+	−	30	0.60	365	20.0	15.047	0.000
diff_surfaceAdvantage	+	+	−	30	0.80	365	20.0	13.220	0.000
diff_surfaceAdvantage	+	+	−	30	0.80	365	5.0	13.163	0.000
diff_surfaceAdvantage	+	+	−	30	0.80	365	1.0	12.943	0.000
diff_surfaceAdvantage	+	+	−	30	0.80	365	0.5	12.375	0.000
diff_surfaceAdvantage	+	+	−	30	0.90	365	20.0	11.627	0.001
diff_surfaceAdvantage	+	+	−	30	0.90	365	5.0	11.574	0.001
diff_rankMomentum_log	+	+	−	90	0.99	730	1.0	**2.162**	**0.141**
diff_rankMomentum_log	+	+	−	90	0.95	730	1.0	2.161	0.142
diff_rankMomentum_log	+	+	−	90	1.00	730	1.0	2.160	0.142

Continued on next page

Table 11 continued from previous page

Feature	SR	HA	CO	TD(t)	TD(f)	TF	UA	χ^2 Score	p-value
diff_rankMomentum_log	+	+	−	180	0.99	730	1.0	2.160	0.142
diff_rankMomentum_log	+	+	−	180	0.95	730	1.0	2.157	0.142
diff_rankMomentum_log	+	+	−	180	0.90	730	1.0	2.149	0.143
diff_rankMomentum_log	+	+	−	90	0.90	730	1.0	2.145	0.143
diff_rankMomentum_log	+	+	−	180	0.99	365	1.0	0.934	0.334
diff_rankMomentum_log	+	+	−	90	0.99	365	1.0	0.932	0.334
diff_rankMomentum_log	+	+	−	180	0.95	365	1.0	0.928	0.335
diff_rankptMomentum_log	+	+	−	30	0.80	365	1.0	**0.233**	**0.629**
diff_rankptMomentum_log	+	+	−	180	0.80	180	5.0	0.214	0.644
diff_rankptMomentum_log	+	+	−	180	0.90	180	20.0	0.214	0.644
diff_rankptMomentum_log	+	+	−	180	0.80	180	20.0	0.214	0.644
diff_rankptMomentum_log	+	+	−	180	0.99	180	20.0	0.214	0.644
diff_rankptMomentum_log	+	+	−	180	0.99	180	5.0	0.214	0.644
diff_rankptMomentum_log	+	+	−	180	0.90	180	5.0	0.214	0.644
diff_rankptMomentum_log	+	+	−	180	0.95	180	20.0	0.214	0.644
diff_rankptMomentum_log	+	+	−	180	0.95	180	5.0	0.214	0.644
diff_rankptMomentum_log	+	+	−	90	0.99	180	5.0	0.214	0.644
diff_avgDuration	+	+	−	90	0.80	7300	1.0	**0.550**	**0.458**
diff_avgDuration	+	+	−	90	0.80	1095	1.0	0.540	0.462
diff_avgDuration	+	+	−	365	0.95	7300	1.0	0.525	0.469
diff_avgDuration	+	+	−	365	0.90	7300	1.0	0.516	0.473
diff_avgDuration	+	+	−	365	0.95	7300	5.0	0.505	0.477
diff_avgDuration	+	+	−	365	0.95	7300	20.0	0.505	0.477
diff_avgDuration	+	+	−	365	0.80	7300	1.0	0.501	0.479
diff_avgDuration	+	+	−	90	0.95	7300	1.0	0.498	0.480
diff_avgDuration	+	+	−	365	0.90	7300	5.0	0.496	0.481
diff_avgDuration	+	+	−	365	0.90	7300	20.0	0.496	0.481
diff_surfaceWinRatio	+	+	+	365	1.00	1095	5.0	**92.293**	**0.000**
diff_surfaceWinRatio	+	+	+	365	0.99	1095	2.0	92.291	0.000
diff_surfaceWinRatio	+	+	+	365	0.99	1095	5.0	92.291	0.000
diff_surfaceWinRatio	+	+	+	365	0.95	1095	5.0	92.285	0.000
diff_surfaceWinRatio	+	+	+	365	0.95	1095	2.0	92.285	0.000
diff_surfaceWinRatio	+	+	+	365	0.95	1095	20.0	92.285	0.000
diff_surfaceWinRatio	+	+	+	365	0.90	1095	5.0	92.278	0.000
diff_surfaceWinRatio	+	+	+	365	0.90	1095	20.0	92.278	0.000
diff_surfaceWinRatio	+	+	+	365	0.90	1095	2.0	92.278	0.000
diff_surfaceWinRatio	+	+	+	365	0.80	1095	5.0	92.271	0.000
diff_overallWinRatio	+	+	+	365	0.95	365	5.0	**103.352**	**0.000**
diff_overallWinRatio	+	+	+	365	0.90	365	20.0	103.352	0.000
diff_overallWinRatio	+	+	+	365	0.99	365	20.0	103.352	0.000
diff_overallWinRatio	+	+	+	365	0.99	365	5.0	103.352	0.000
diff_overallWinRatio	+	+	+	365	0.95	365	20.0	103.352	0.000
diff_overallWinRatio	+	+	+	365	0.80	365	5.0	103.352	0.000
diff_overallWinRatio	+	+	+	365	0.80	365	20.0	103.352	0.000
diff_overallWinRatio	+	+	+	365	0.90	365	5.0	103.352	0.000
diff_overallWinRatio	+	+	+	180	0.99	365	5.0	103.342	0.000
diff_overallWinRatio	+	+	+	180	0.99	365	20.0	103.342	0.000
diff_h2hWinRatio	+	+	−	180	0.90	7300	100.0	**138.311**	**0.000**
diff_h2hWinRatio	+	+	−	90	0.95	7300	100.0	138.284	0.000
diff_h2hWinRatio	+	+	−	90	0.95	7300	20.0	137.754	0.000
diff_h2hWinRatio	+	+	−	900	0.95	7300	20.0	137.754	0.000
diff_h2hWinRatio	+	+	−	180	0.90	7300	20.0	136.943	0.000
diff_h2hWinRatio	+	+	−	180	0.95	7300	100.0	136.560	0.000
diff_h2hWinRatio	+	+	−	180	0.95	7300	20.0	136.560	0.000
diff_h2hWinRatio	+	+	−	365	0.90	7300	20.0	136.473	0.000
diff_h2hWinRatio	+	+	−	365	0.90	7300	100.0	136.473	0.000
diff_h2hWinRatio	+	+	−	365	0.90	7300	20.0	136.473	0.000
diff_firstServeInRatio	+	+	−	90	0.80	7300	0.2	**3.212**	**0.073**
diff_firstServeInRatio	+	+	−	90	0.80	1095	0.2	3.193	0.074
diff_firstServeInRatio	+	+	−	30	0.95	1095	0.2	3.173	0.075
diff_firstServeInRatio	+	+	−	90	0.90	1095	0.2	3.139	0.076
diff_firstServeInRatio	+	+	−	30	0.95	7300	0.2	3.133	0.077

Continued on next page

Table 11 continued from previous page

Feature	SR	HA	CO	TD(t)	TD(f)	TF	UA	χ^2 Score	p-value
diff_firstServeInRatio	+	+	−	90	0.80	7300	1.0	3.082	0.079
diff_firstServeInRatio	+	+	−	365	0.80	1095	0.2	3.060	0.080
diff_firstServeInRatio	+	+	−	90	0.80	1095	1.0	3.056	0.081
diff_firstServeInRatio	+	+	−	90	0.90	7300	0.2	3.054	0.081
diff_firstServeInRatio	+	+	−	90	0.95	1095	0.2	3.050	0.081
diff_firstServeWinRatio	+	+	+	90	0.90	1095	0.2	**28.447**	**0.000**
diff_firstServeWinRatio	+	+	+	180	0.80	1095	0.2	28.176	0.000
diff_firstServeWinRatio	+	+	+	90	0.90	1460	0.2	27.885	0.000
diff_firstServeWinRatio	+	+	+	90	0.90	1825	0.2	27.786	0.000
diff_firstServeWinRatio	+	+	+	180	0.80	1460	0.2	27.653	0.000
diff_firstServeWinRatio	+	+	+	180	0.80	1825	0.2	27.552	0.000
diff_firstServeWinRatio	+	+	+	90	0.80	1825	0.2	27.530	0.000
diff_firstServeWinRatio	+	+	+	365	0.80	1825	0.2	26.259	0.000
diff_firstServeWinRatio	+	+	+	365	0.80	1095	0.2	25.767	0.000
diff_firstServeWinRatio	+	+	+	365	0.80	1460	0.2	25.227	0.000
diff_secondServeWinRatio	+	+	+	365	0.90	1095	0.2	**21.742**	**0.000**
diff_secondServeWinRatio	+	+	+	90	0.95	1095	0.2	21.617	0.000
diff_secondServeWinRatio	+	+	+	365	0.80	1095	0.2	21.144	0.000
diff_secondServeWinRatio	+	+	+	90	0.95	730	0.2	20.895	0.000
diff_secondServeWinRatio	+	+	+	365	0.95	1095	0.2	20.624	0.000
diff_secondServeWinRatio	+	+	+	90	0.90	1095	0.2	20.517	0.000
diff_secondServeWinRatio	+	+	+	365	0.90	730	0.2	20.361	0.000
diff_secondServeWinRatio	+	+	+	365	0.80	730	0.2	20.341	0.000
diff_secondServeWinRatio	+	+	+	90	0.90	730	0.2	20.279	0.000
diff_secondServeWinRatio	+	+	+	30	0.95	1095	0.2	20.246	0.000
diff_overallServeWinRatio	+	+	+	90	0.80	7300	0.2	**44.552**	**0.000**
diff_overallServeWinRatio	+	+	+	90	0.80	7300	0.1	39.741	0.000
diff_overallServeWinRatio	+	+	+	30	0.95	1095	0.2	38.637	0.000
diff_overallServeWinRatio	+	+	+	90	0.80	1095	0.2	38.018	0.000
diff_overallServeWinRatio	+	+	+	365	0.80	1095	0.2	37.403	0.000
diff_overallServeWinRatio	+	+	+	365	0.90	1095	0.2	37.051	0.000
diff_overallServeWinRatio	+	+	+	30	0.95	7300	0.2	36.997	0.000
diff_overallServeWinRatio	+	+	+	90	0.90	7300	0.2	36.881	0.000
diff_overallServeWinRatio	+	+	+	30	0.90	7300	0.2	36.423	0.000
diff_overallServeWinRatio	+	+	+	90	0.95	1095	0.2	36.386	0.000
diff_overallReturnWinRatio	+	+	+	90	0.80	7300	0.2	**3.698**	**0.054**
diff_overallReturnWinRatio	+	+	+	30	0.95	7300	0.2	3.532	0.060
diff_overallReturnWinRatio	+	+	+	365	0.80	7300	0.2	3.293	0.070
diff_overallReturnWinRatio	+	+	+	365	0.90	7300	0.2	3.093	0.079
diff_overallReturnWinRatio	+	+	+	365	0.80	1095	0.2	3.052	0.081
diff_overallReturnWinRatio	+	+	+	365	0.90	1095	0.2	2.978	0.084
diff_overallReturnWinRatio	+	+	+	90	0.90	7300	0.2	2.759	0.097
diff_overallReturnWinRatio	+	+	+	90	0.80	1095	0.2	2.718	0.099
diff_overallReturnWinRatio	+	+	+	30	0.95	1095	0.2	2.462	0.117
diff_overallReturnWinRatio	+	+	+	90	0.95	7300	0.2	2.437	0.118
diff_completeness	+	+	+	365	0.90	7300	0.2	**4.333**	**0.037**
diff_completeness	+	+	+	365	0.80	7300	0.2	4.329	0.037
diff_completeness	+	+	+	90	0.90	7300	0.2	4.158	0.041
diff_completeness	+	+	+	90	0.90	1095	0.2	4.096	0.043
diff_completeness	+	+	+	30	0.95	7300	0.2	3.946	0.047
diff_completeness	+	+	+	30	0.95	1095	0.2	3.730	0.053
diff_completeness	+	+	+	30	0.90	1095	0.2	3.721	0.054
diff_completeness	+	+	+	90	0.80	7300	0.2	3.469	0.063
diff_completeness	+	+	+	365	0.90	1095	0.2	3.447	0.063
diff_completeness	+	+	+	90	0.80	1095	0.2	3.238	0.072
diff_acePerServePoint	+	+	−	90	0.80	1095	0.2	**14.006**	**0.000**
diff_acePerServePoint	+	+	−	30	0.95	7300	0.2	13.958	0.000
diff_acePerServePoint	+	+	−	90	0.80	7300	0.2	13.945	0.000
diff_acePerServePoint	+	+	−	30	0.95	1095	0.2	13.919	0.000
diff_acePerServePoint	+	+	−	90	0.90	1095	0.2	13.861	0.000
diff_acePerServePoint	+	+	−	90	0.90	7300	0.2	13.721	0.000
diff_acePerServePoint	+	+	−	90	0.95	1095	0.2	13.660	0.000

Continued on next page

Table 11 continued from previous page

Feature	SR	HA	CO	TD(t)	TD(f)	TF	UA	χ^2 Score	p-value
diff_acePerServePoint	+	+	−	365	0.80	1095	0.2	13.651	0.000
diff_acePerServePoint	+	+	−	365	0.90	1095	0.2	13.547	0.000
diff_acePerServePoint	+	+	−	365	0.95	1095	0.2	13.519	0.000
diff_dfPerServePoint	+	+	+	365	0.95	7300	0.2	**3.864**	**0.049**
diff_dfPerServePoint	+	+	+	90	0.95	7300	0.2	3.707	0.054
diff_dfPerServePoint	+	+	+	365	0.90	7300	0.2	3.699	0.054
diff_dfPerServePoint	+	+	+	365	0.80	7300	0.2	3.656	0.056
diff_dfPerServePoint	+	+	+	90	0.90	7300	0.2	3.526	0.060
diff_dfPerServePoint	+	+	+	365	0.95	1095	0.2	3.515	0.061
diff_dfPerServePoint	+	+	+	365	0.90	1095	0.2	3.349	0.067
diff_dfPerServePoint	+	+	+	90	0.95	1095	0.2	3.345	0.067
diff_dfPerServePoint	+	+	+	90	0.80	7300	0.2	3.344	0.067
diff_dfPerServePoint	+	+	+	30	0.95	7300	0.2	3.329	0.068
diff_acePerServeGame	+	+	−	30	0.95	1095	5.0	**11.402**	**0.001**
diff_acePerServeGame	+	+	−	30	0.95	7300	5.0	11.307	0.001
diff_acePerServeGame	+	+	−	30	0.95	1095	1.0	11.282	0.001
diff_acePerServeGame	+	+	−	30	0.95	7300	1.0	11.202	0.001
diff_acePerServeGame	+	+	−	90	0.95	1095	5.0	11.145	0.001
diff_acePerServeGame	+	+	−	30	0.95	1095	0.2	11.086	0.001
diff_acePerServeGame	+	+	−	90	0.80	1095	0.2	11.046	0.001
diff_acePerServeGame	+	+	−	90	0.95	1095	1.0	11.045	0.001
diff_acePerServeGame	+	+	−	90	0.90	1095	0.2	11.004	0.001
diff_acePerServeGame	+	+	−	30	0.95	7300	0.2	10.996	0.001
diff_dfPerServeGame	+	+	+	365	0.95	1095	0.2	**4.686**	**0.030**
diff_dfPerServeGame	+	+	+	90	0.80	1095	0.2	4.658	0.031
diff_dfPerServeGame	+	+	+	90	0.95	1095	0.2	4.497	0.034
diff_dfPerServeGame	+	+	+	365	0.90	1095	0.2	4.483	0.034
diff_dfPerServeGame	+	+	+	365	0.80	1095	0.2	4.427	0.035
diff_dfPerServeGame	+	+	+	90	0.90	1095	0.2	4.328	0.037
diff_dfPerServeGame	+	+	+	30	0.95	1095	0.2	4.260	0.039
diff_dfPerServeGame	+	+	+	90	0.80	365	0.2	4.035	0.045
diff_dfPerServeGame	+	+	+	30	0.90	365	0.2	3.706	0.054
diff_dfPerServeGame	+	+	+	365	0.95	365	0.2	3.493	0.062
diff_bpSaveRatio	+	+	+	365	0.80	1095	0.2	**1.066**	**0.302**
diff_bpSaveRatio	+	+	+	365	0.90	1095	0.2	1.041	0.308
diff_bpSaveRatio	+	+	+	90	0.90	1095	0.2	1.028	0.311
diff_bpSaveRatio	+	+	+	90	0.95	1095	0.2	1.017	0.313
diff_bpSaveRatio	+	+	+	365	0.95	1095	0.2	0.811	0.368
diff_bpSaveRatio	+	+	+	30	0.95	1095	0.2	0.769	0.380
diff_bpSaveRatio	+	+	+	90	0.80	1095	0.2	0.668	0.414
diff_bpSaveRatio	+	+	−	365	0.80	1095	0.2	0.507	0.476
diff_bpSaveRatio	+	+	−	90	0.95	1095	0.2	0.494	0.482
diff_bpSaveRatio	+	+	+	90	0.90	365	0.2	0.494	0.482

APPENDIX B

MODELING HYPERPARAMETER TUNING

Table 12: Top 10 Grid Search Hyperparameter Settings for each Machine Learning Model

Model	Hyperparameter Setting	Train		Validation	
		Brier Score	Accuracy	Brier Score	Accuracy
XGB-FS	alpha: 0, colsample_bytree: 1, lambda: 0, learning_rate: 0.05, max_depth: 3, min_child_weight: 4, n_estimators: 50, subsample: 0.5	0.1829	0.7208	0.1902	0.7077
XGB-FS	alpha: 0, colsample_bytree: 1, lambda: 0.5, learning_rate: 0.05, max_depth: 3, min_child_weight: 4, n_estimators: 50, subsample: 0.5	0.1830	0.7212	0.1903	0.7041
XGB-FS	alpha: 0, colsample_bytree: 1, lambda: 0, learning_rate: 0.05, max_depth: 3, min_child_weight: 1, n_estimators: 50, subsample: 0.5	0.1825	0.7229	0.1904	0.7048
XGB-FS	alpha: 0.5, colsample_bytree: 1, lambda: 0, learning_rate: 0.05, max_depth: 3, min_child_weight: 4, n_estimators: 50, subsample: 0.5	0.1830	0.7210	0.1904	0.7044
XGB-FS	alpha: 0, colsample_bytree: 1, lambda: 0.5, learning_rate: 0.05, max_depth: 3, min_child_weight: 2, n_estimators: 50, subsample: 0.5	0.1828	0.7206	0.1904	0.7041
XGB-FS	alpha: 0, colsample_bytree: 1, lambda: 1, learning_rate: 0.05, max_depth: 3, min_child_weight: 1, n_estimators: 50, subsample: 0.5	0.1829	0.7200	0.1904	0.7052
XGB-FS	alpha: 0, colsample_bytree: 1, lambda: 0, learning_rate: 0.05, max_depth: 3, min_child_weight: 2, n_estimators: 50, subsample: 0.5	0.1826	0.7224	0.1904	0.7043
XGB-FS	alpha: 0, colsample_bytree: 1, lambda: 1, learning_rate: 0.05, max_depth: 3, min_child_weight: 2, n_estimators: 50, subsample: 0.5	0.1830	0.7216	0.1904	0.7039
XGB-FS	alpha: 0.5, colsample_bytree: 1, lambda: 1, learning_rate: 0.05, max_depth: 3, min_child_weight: 1, n_estimators: 50, subsample: 0.5	0.1830	0.7191	0.1904	0.7042
XGB-FS	alpha: 0, colsample_bytree: 1, lambda: 0.5, learning_rate: 0.05, max_depth: 3, min_child_weight: 1, n_estimators: 50, subsample: 0.5	0.1827	0.7208	0.1904	0.7033
SVM-FS	C: 100, degree: 2, gamma: 'auto', kernel: 'linear'	0.1874	0.7135	0.1904	0.7078

Continued on next page

Table 12 continued from previous page

Model	Hyperparameter Setting	Train		Validation	
		Brier Score	Accuracy	Brier Score	Accuracy
SVM-FS	C: 100, degree: 4, gamma: 'scale', kernel: 'linear'	0.1874	0.7135	0.1904	0.7078
SVM-FS	C: 100, degree: 1, gamma: 'scale', kernel: 'linear'	0.1874	0.7135	0.1904	0.7078
SVM-FS	C: 100, degree: 1, gamma: 'auto', kernel: 'linear'	0.1874	0.7135	0.1904	0.7078
SVM-FS	C: 100, degree: 2, gamma: 'scale', kernel: 'linear'	0.1874	0.7135	0.1904	0.7078
SVM-FS	C: 100, degree: 3, gamma: 'scale', kernel: 'linear'	0.1874	0.7135	0.1904	0.7078
SVM-FS	C: 100, degree: 4, gamma: 'auto', kernel: 'linear'	0.1874	0.7135	0.1904	0.7078
SVM-FS	C: 100, degree: 3, gamma: 'auto', kernel: 'linear'	0.1874	0.7135	0.1904	0.7078
SVM-FS	C: 1, degree: 4, gamma: 'scale', kernel: 'linear'	0.1874	0.7135	0.1904	0.7077
SVM-FS	C: 1, degree: 2, gamma: 'scale', kernel: 'linear'	0.1874	0.7135	0.1904	0.7077
XGB-MI	alpha: 0, colsample_bytree: 1, lambda: 0.5, learning_rate: 0.05, max_depth: 3, min_child_weight: 1, n_estimators: 50, subsample: 0.8	0.1828	0.7241	0.1905	0.7028
XGB-MI	alpha: 0, colsample_bytree: 1, lambda: 0, learning_rate: 0.05, max_depth: 3, min_child_weight: 1, n_estimators: 50, subsample: 0.5	0.1825	0.7175	0.1905	0.7045
LR-MI	C: 0.1, max_iter: 500, penalty: 'l1', solver: 'saga'	0.1867	0.7139	0.1905	0.7080
LR-MI	C: 0.1, max_iter: 100, penalty: 'l1', solver: 'saga'	0.1867	0.7139	0.1905	0.7080
LR-MI	C: 0.1, max_iter: 2000, penalty: 'l1', solver: 'saga'	0.1867	0.7139	0.1905	0.7080
XGB-MI	alpha: 0.5, colsample_bytree: 1, lambda: 1, learning_rate: 0.05, max_depth: 3, min_child_weight: 4, n_estimators: 50, subsample: 0.5	0.1831	0.7216	0.1905	0.7044
XGB-MI	alpha: 0, colsample_bytree: 1, lambda: 1, learning_rate: 0.05, max_depth: 3, min_child_weight: 1, n_estimators: 50, subsample: 0.8	0.1829	0.7218	0.1905	0.7041
XGB-MI	alpha: 1, colsample_bytree: 1, lambda: 0.5, learning_rate: 0.05, max_depth: 3, min_child_weight: 4, n_estimators: 50, subsample: 0.5	0.1832	0.7210	0.1905	0.7039
XGB-MI	alpha: 1, colsample_bytree: 1, lambda: 1, learning_rate: 0.05, max_depth: 3, min_child_weight: 1, n_estimators: 50, subsample: 0.5	0.1831	0.7233	0.1905	0.7044
XGB-MI	alpha: 1, colsample_bytree: 1, lambda: 0.5, learning_rate: 0.05, max_depth: 3, min_child_weight: 1, n_estimators: 50, subsample: 0.5	0.1831	0.7204	0.1905	0.7047
XGB-MI	alpha: 0.5, colsample_bytree: 1, lambda: 0.5, learning_rate: 0.05, max_depth: 3, min_child_weight: 4, n_estimators: 50, subsample: 0.5	0.1830	0.7224	0.1905	0.7046
XGB-MI	alpha: 1, colsample_bytree: 1, lambda: 0, learning_rate: 0.05, max_depth: 3, min_child_weight: 1, n_estimators: 50, subsample: 0.5	0.1829	0.7220	0.1905	0.7043
XGB-MI	alpha: 0.5, colsample_bytree: 1, lambda: 0, learning_rate: 0.05, max_depth: 3, min_child_weight: 1, n_estimators: 50, subsample: 1	0.1835	0.7185	0.1905	0.7043
LR-MI	C: 0.1, max_iter: 500, penalty: 'elasticnet', solver: 'saga'	0.1866	0.7141	0.1906	0.7084
LR-MI	C: 0.1, max_iter: 100, penalty: 'elasticnet', solver: 'saga'	0.1866	0.7141	0.1906	0.7084
LR-MI	C: 0.1, max_iter: 2000, penalty: 'elasticnet', solver: 'saga'	0.1866	0.7141	0.1906	0.7084
LR-FS	C: 0.1, max_iter: 100, penalty: 'l1', solver: 'saga'	0.1863	0.7156	0.1906	0.7076

Continued on next page

Table 12 continued from previous page

Model	Hyperparameter Setting	Train		Validation	
		Brier Score	Accuracy	Brier Score	Accuracy
LR-FS	C: 0.1, max_iter: 2000, penalty: 'l1', solver: 'saga'	0.1863	0.7156	0.1906	0.7076
LR-FS	C: 0.1, max_iter: 500, penalty: 'l1', solver: 'saga'	0.1863	0.7156	0.1906	0.7076
LR-MI	C: 0.1, max_iter: 2000, penalty: 'l2', solver: 'saga'	0.1866	0.7125	0.1908	0.7080
LR-MI	C: 0.1, max_iter: 500, penalty: 'l2', solver: 'saga'	0.1866	0.7125	0.1908	0.7080
LR-MI	C: 0.1, max_iter: 100, penalty: 'l2', solver: 'saga'	0.1866	0.7125	0.1908	0.7080
LR-FS	C: 0.1, max_iter: 100, penalty: 'elasticnet', solver: 'saga'	0.1863	0.7154	0.1908	0.7071
LR-FS	C: 0.1, max_iter: 2000, penalty: 'elasticnet', solver: 'saga'	0.1863	0.7154	0.1908	0.7071
LR-FS	C: 0.1, max_iter: 500, penalty: 'elasticnet', solver: 'saga'	0.1863	0.7154	0.1908	0.7071
LR-MI	C: 10, max_iter: 2000, penalty: 'l1', solver: 'saga'	0.1866	0.7139	0.1909	0.7078
LR-FS	C: 0.1, max_iter: 500, penalty: 'l2', solver: 'saga'	0.1863	0.7145	0.1910	0.7069
LR-FS	C: 0.1, max_iter: 100, penalty: 'l2', solver: 'saga'	0.1863	0.7145	0.1910	0.7069
LR-FS	C: 0.1, max_iter: 2000, penalty: 'l2', solver: 'saga'	0.1863	0.7145	0.1910	0.7069
LR-FS	C: 10, max_iter: 2000, penalty: 'l1', solver: 'saga'	0.1863	0.7135	0.1911	0.7070
SVM-MI	C: 1, degree: 4, gamma: 'scale', kernel: 'linear'	0.1883	0.7119	0.1916	0.7075
SVM-MI	C: 1, degree: 2, gamma: 'scale', kernel: 'linear'	0.1883	0.7119	0.1916	0.7075
SVM-MI	C: 1, degree: 2, gamma: 'auto', kernel: 'linear'	0.1883	0.7119	0.1916	0.7075
SVM-MI	C: 1, degree: 1, gamma: 'auto', kernel: 'linear'	0.1883	0.7119	0.1916	0.7075
SVM-MI	C: 1, degree: 1, gamma: 'scale', kernel: 'linear'	0.1883	0.7119	0.1916	0.7075
SVM-MI	C: 1, degree: 4, gamma: 'auto', kernel: 'linear'	0.1883	0.7119	0.1916	0.7075
SVM-MI	C: 1, degree: 3, gamma: 'scale', kernel: 'linear'	0.1883	0.7119	0.1916	0.7075
SVM-MI	C: 1, degree: 3, gamma: 'auto', kernel: 'linear'	0.1883	0.7119	0.1916	0.7075
SVM-MI	C: 100, degree: 3, gamma: 'auto', kernel: 'linear'	0.1883	0.7116	0.1917	0.7076
SVM-MI	C: 100, degree: 2, gamma: 'auto', kernel: 'linear'	0.1883	0.7116	0.1917	0.7076