UNIVERSITY OF CALIFORNIA
Los Angeles

Forecasting the Outcomes of Professional Tennis Matches

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Applied Statistics & Data Science

by

Noah Clair Read

2024

ABSTRACT OF THE THESIS

Forecasting the Outcomes of
Professional Tennis Matches

by

Noah Clair Read
Master of Applied Statistics & Data Science
University of California, Los Angeles, 2024
Professor Frederic R. Paik Schoenberg, Chair

Data analytics and machine learning have become vital applications in the world of sports, providing glimpses of statistically probable, mathematically formulated future states while helping teams and individual athletes make smarter decisions. The sport of tennis, featuring a relatively low number of moving parts, a bevy of readily available data and a global interest, is ripe for advanced forecasting analysis. In the following, the method of adaptive least squares is leveraged in conjunction with Kalman filtering to create time-variant match statistics for professional tennis players. Once the adaptive model is constructed, the sigmoid function is utilized to transform forecasted delta set values into forecasted probabilities of winning for any given matchup. The most successful model constructed from the web-scraped pool of data is continuously improving and correctly forecasts tennis match outcomes 63.54% of the time, exceeding the prediction accuracy if outcomes were chosen solely based on professional rankings.

The thesis of Noah Clair Read is approved.

David Anthony Zes

Yingnian Wu

Michael Tsiang

Frederic R. Paik Schoenberg, Committee Chair

University of California, Los Angeles

2024

To my mom and dad . . .

thanks for everything

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost I would like to thank Dave Zes for all of the time and expertise he provided throughout the entire life cycle of this project. His guidance was paramount, and I am incredibly grateful for his mentorship and imparted knowledge.

I'd like to acknowledge the primary data sources for this project, Rotowire, Tennis Abstract, and the ATP & WTA. The consistently-updated information made this project possible.

Finally, I would like to express an appreciation for the time change between California and Melbourne, Australia. The numerous late, late nights finishing up this project coincided nicely with 2024 Australian Open coverage.

CHAPTER 1

Introduction

Having the ability to forecast future events, sentiments or outcomes is notoriously valuable in nearly all facets of daily life. The implementation of said forecasting can happen on any scale, whether it be on an individual level, a global level, or beyond. The process is frequently leveraged to peer into a possible future state of some discipline such as meteorology, economics, consumer activity, sports, or even climatology. Regardless of the slice of life in which forecasting may be implemented, the process is completely reliant upon historical data, underlying reasoning, and logic. Looking into the future through a data-bolstered lens is powerful and can yield shockingly accurate results, but the degree of accuracy is at the mercy of the quality of the data.

I have adored sports my entire life, and while I grew up with tennis serving as my primary sport, my love of this specific sport did not blossom until years later. In recent years past I have attended Women's Tennis Association (WTA) events in San Jose, CA, Association of Tennis Professionals (ATP) Challenger events in Newport Beach, CA, and larger ATP/WTA tournaments in Indian Wells, CA and Flushing Meadows, NY. Figure 1.1 below shows a snapshot from the US Open, a Grand Slam tournament played in New York each fall that I was fortunate enough to attend in 2023.

Figure 1.1: Louis Armstrong Stadium, US Open 2023

With the explosion of data science, machine learning and analytics within the world of sports and sports betting, an opportunity was identified to bridge one of my biggest hobbies with some of my deepest educational interests. Sports such as American football, basketball and baseball have already become heavily reliant upon analytics, and for good reason. While there will always be some elements of stochasticity in sports, the incredibly vast amount of data that can be aggregated can lead to the discovery of underlying patterns that assist in forecasting the next play, the next shot, or even a competition's eventual outcome. If a sport like football, featuring twenty-two moving parts on the field, can leverage principles of data science for analytical purposes such as forecasting, narrowing down that number of confounding variables to two within a sport like tennis theoretically has a good chance to yield even more meaningful analytical patterns and accurate predictive models.

The following project will leverage the method of Kalman filtering to transform time-

variant data into an appropriate input to an adaptive least square regression model which ultimately outputs forecasted delta set values. "Delta set" refers to the change in sets in any given match for each opposing player, given the outcome. If player A wins a best-of-3 set match in straight sets, player A's corresponding delta set value is +2. On the other hand, the delta set value for player A's opponent is -2. This forecasted value is subsequently mapped to a probability of winning, which allows for a forecasted match winner to be identified. While it is powerful to be equipped with forecasts of match outcomes, the resulting logistic cost and forecast accuracies with respect to the true match outcomes must be measured to quantify each model's performance.

Tennis is a grueling, physical sport, yet the mental aspect of the game is equally as vital to the outcome, if not more. Just when it appears that a player is clearly superior to his/her opponent and is going to win the match, it is not uncommon for the mental nature of the sport to creep in and overwhelm the level of skill. The unpredictability of tennis helps to give the sport its awe-inspiring feel. Take the recently-completed ATP 250 Los Cabos event in February 2024 as an example: number eight seed Jordan Thompson of Australia found himself serving down 0-6, 1-4, 15-40 (5 points away from defeat) to unseeded American teenager Alex Michelsen in the quarterfinals. Any live match forecasting model would overwhelmingly favor Michelsen to close the match out, and the odds of Thompson winning the tournament at that juncture were basically zero. Thompson ended up beating Michelsen, taking out the top-seeded German Alexander Zverev in the semifinals, and beating four-seed Norwegian Casped Ruud in the finals. What changed for Thompson all of a sudden? How could any model forecast such a turn of events?

For all of the historical outcomes and statistics available, the computing power that our machines harness and even the raw knowledge of the game that we collectively possess, match outcomes can not be forecasted with 100% accuracy, but what fun would the sport be if that were possible? While it is unclear how best to capture the mental portion of tennis, or the levels of momentum which appear to switch directions on a dime, this project strives to

construct algorithms which process every collected piece of statistical information to forecast match outcomes as accurately as possible, as well as reveal what statistical variables appear to be the most influential.

CHAPTER 2

Methodology

2.1 Overview

A considerable percentage of this project involved web scraping and subsequent data preparation, with the output of those processes serving as the input to the ultimate adaptive least squares regression model and logistic mapping. A reliable data source needed to be unearthed, one that allowed for the scores and statistics of every professional tennis match to be scraped on a daily basis. Tournament matches from the ATP, WTA, and ATP Challenger Tour were included. The consistent frequency of data scraping was primarily due to tennis matches being played every day, meaning that new information from freshly-completed matches was available and ready to be leveraged. Information of interest was not solely found within one website; multiple data sources underwent daily scraping and the information was retained to ensure that it would remain available at all times, regardless of its public availability.

Three unique sets of model features were chosen to ultimately construct three different models. One model included every feature that had been scraped, the next model included only a few vital features to reduce complexity and the third model included a set of features chosen at the author's discretion. After extensive preparation, the data was filtered so that average, time-variant values of each feature were calculated over time for each individual player and the respective opponents. This process quadrupled the amount of data present; a matrix was constructed consisting of time-variant average values for four metrics pertaining to player and player opponent. This matrix was used in conjunction with other variables that

were known before the match started, known as *a priori* variables, and fed into a Kalman filter and adaptive least squares model. Once the forecasted values for the delta set variable were calculated, a logistic filter was initialized and the forecasted values were mapped to probabilities of winning. A flowchart displaying each step of the model-creation process is shown below in figure 2.1. Each method will be explored in more depth in the following sections.

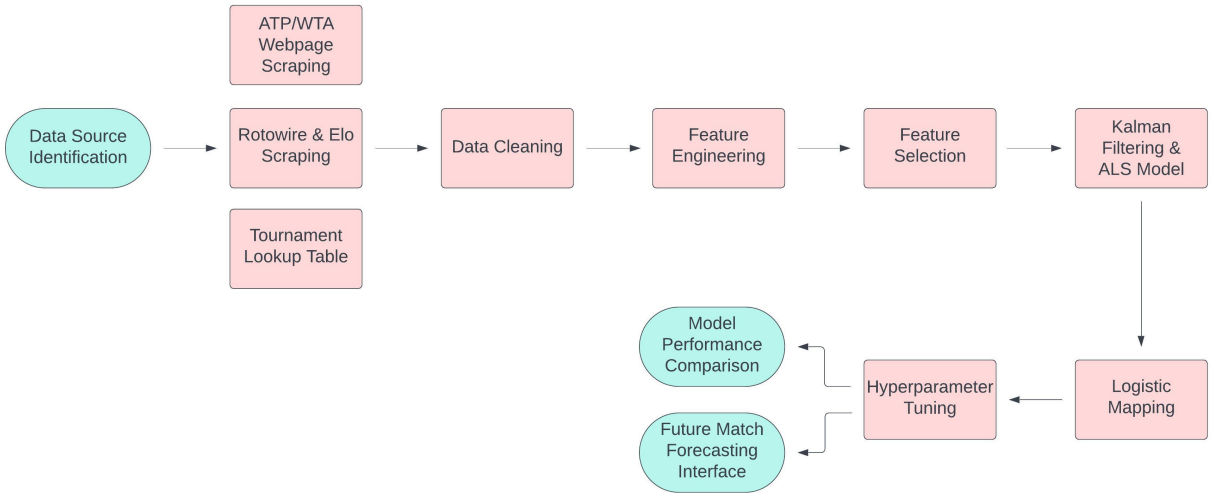

Figure 2.1: Project Process Overview

2.2 Data Collection

2.2.1 Rotowire

Given the parameters for the desired data source outlined above, Rotowire [Rot] fit the bill and became the predominant means of data acquisition. Rotowire primarily serves as a fantasy sports and sports betting website, covering numerous professional leagues (including some college sports). Game scores and individual player stats are available along with news, articles, and a deep reservoir of additional content. Narrowing the focus to the sport of tennis, match scores and corresponding player statistics from ATP, WTA and ATP Challenger

Tour [1] matches are featured on the site.

The statistics corresponding to each match were extracted from different Rotowire webpages by digging into the HTML (HyperText Markup Language) code of each webpage and pulling out elements of interest, whether they be text, numbers, or even full tables. A noteworthy downside of choosing Rotowire as a data source is that the tennis match scores and statistics featured on the website are overwritten after seven days, so for any given day there only exists one full week of information to scrape. To ensure that all data was gathered, match scraping was (and will continue to be) performed on a daily basis. The Rotowire data-scraping process was performed using the R programming language, with the *XML* package serving as the primary HTML-parsing tool. Each webpage housing the statistics for some unique match had its respective HTML source code analyzed, and every element of interest was extracted and saved into an R dataframe object.

2.2.2 ATP & WTA Rankings

As comprehensive a data source as Rotowire appeared to be, there remained to be variables of interest that were not present on the website; therefore, additional data sources needed to be unearthed. While the "player rank" variable was indeed available on most of the match statistic webpages on Rotowire, the number was frequently missing, especially for the smaller WTA tournaments and ATP Challenger tournaments [2]. To combat this infrequency, a script was written in the python programming language that scrapes the ATP live rankings [ATPa] and the WTA live rankings [WTA] from the associations' respective websites. The *BeautifulSoup4* module was leveraged to scrape the ATP website. Scraping the WTA

[1] There exists an equivalent to the Challenger Tour featured in women's tennis (WTA 125 tournaments), yet Rotowire does not feature match statistics for these tournaments. WTA 125 tournaments are far less frequent than ATP Challenger tournaments, as there were 196 Challenger tournaments in 2023 and just 31 WTA 125 tournaments. This infrequency paired with the "lower" level may play into the data exclusion.

[2] I wanted the rankings to be as accurate as possible; if a player won his/her previous match, that result is recorded in the "live" rankings even though the "official" rankings would not have reflected that match result quite yet, since those are updated weekly.

website proved to be slightly more challenging. A PDF version of the WTA ranking data was unearthed within the HTML code of the website, and the contents of the PDF were scraped and ordered into a dataframe. This unique process was performed in python by leveraging the *BeautifulSoup4*, *PyPDF2*, and *tabula* modules. These scripts were (and once again, will continue to be) executed on a daily basis to capture the most updated live ranking for each player. This ensures that all units of observation include an accurate-to-the-day live rank at the time of each match. Along with live rank, the age of each player was scraped from the ATP website and the nationality of each player was scraped from the WTA website[3].

The ATP website underwent a complete overhaul at one point while this project was active. With a completely novel HTML structure supporting the site, the elements of interest on the rankings webpage were no longer found in the same exact location, nor were they labeled the same within the code. The python script that scraped ranking information was rewritten to gel with the new HTML webpage structure.

2.2.3 Tennis Abstract

The elo rating system was originally designed for chess, yet its application has expanded greatly over time. For a zero-sum game such as tennis where a win for one player and a loss for another happen in tandem, elo ratings are helpful representations of each individual player's strength. Elo ratings for a player and any changes are contingent on the elo ratings of the opponents. Beating a player with a low elo rating will not result in as large of a rating increase as beating a high elo player. When gauging player performance, it is natural to consider historical data, that is, how well the player has performed in the past. Another worthy consideration involves the breadth of that historical data. It is certainly easier to estimate how good a player is when they have played a lot of matches. Elo ratings experience greater alterations when a player has not played many matches in the past, as the rating

[3]Age and nationality didn't play a part in this project's models, but they were easily available to scrape and may play a part in future modeling.

confidence is not as high.

Elo ratings for both male and female tennis players were scraped from the Tennis Abstract [4][Sac] website using python. On this site ratings exist for all professional players that have played at least ten professional tour matches in the previous fifty-two weeks. The elo ratings are not inherently court surface-specific, but Tennis Abstract does provide court-specific ratings as well as blended ratings that are combinations of the two aforementioned webpage offerings. Both the general ratings and the court-specific ratings were scraped for potential future model inclusion.

2.2.4 Tournament Lookup Table

While the outcome of a tennis match and the resulting player statistics are unknown prior to the match actually being played, there is some information that is indeed known ahead of time. There were multiple *a priori* variables pertaining to each tennis match that were of interest: the maximum number of sets that the match could consist of (bestOf), the number of games featured in each set (gamesPerSet), the gender of the players, the court surface that the match would be played on and the level of the tournament. While all women's tennis matches and the majority of men's matches are played as best-of-three sets, men's Grand Slam tournaments (Australian Open, French Open, Wimbledon, and US Open) are played as best-of-five sets. In terms of scoring within each set, nearly all professional tournament sets are scored as the first player to six games, win by two, with a seven-point tiebreaker played at 6-6. The one tournament during the calendar year that does not abide by these rules is the Next Gen ATP Finals, which is played each November. This tournament incorporates rule tweaks basically as a trial run for potential future application to the entire ATP Tour. The scoring system is a best-of-five set format, yet each set is first to just four games rather than six, with a tie-breaker at 3-3. The inclusion of these matches in the model requires the inclusion of this variable.

[4]If you enjoy following tennis, there is a *vast* collection of compelling content over on the Tennis Abstract website.

Throughout the calendar year, professional matches are played on multiple different surfaces: hard courts, clay courts, grass courts and even carpet courts. A dummy variable was created for surface to capture these differences. Finally, each tournament is not exactly equal in terms of prize money and allocated points. The winner of a Grand Slam, for example, receives 2,000 ranking points while the winner of a Challenger 75 tournament receives 75 points. The hope for the inclusion of tournament level as a model feature was to encapsulate the typical level of competition at which each player competes, as the players who participate in higher-level tournaments face stronger opponents.

Each of the *a priori* variables described above (bestOf, gamesPerSet, gender, surface, and tournament level) were manually documented directly from the ATP and WTA websites and inserted into a lookup table to be joined later with the existing data. Since there is no absolute guarantee that the listed tournament name from the Rotowire match data will exactly match the tournament name from the ATP website, the lookup table updating is performed every day after the Rotowire scraping is completed.

2.3 Feature Engineering

2.3.1 Miscellaneous Transformations

At this juncture, data had been identified and scraped from all sources of interest. The majority was formatted within the scraping scripts, which performed data transformation (letter casing, character replacing, etc.), yet a few variables needed to undergo further engineering. A few of the more noteworthy alterations are described below.

Tiebreaks and supertiebreaks are prevalent in professional matches, as it is often difficult to break a player's serve[5]. Scraping scores from Rotowire included both the set's game score

[5]Two of the best servers of all time, Ivo Karlovic and John Isner, held their service games 92% of the time over their careers. [Ivo] [Isn]

and the tiebreaker score in tiebreaker sets. The exact tiebreaker score was not deemed necessary to capture, so some programming logic was implemented to strictly notate the number of games won in tiebreak/supertiebreak sets. For example, a set that ended in a tiebreaker would be displayed in the source HTML as 7 games won for one player and $6x^6$ for the other. This correction allowed for the creation of a new metric, total games won, which in turn led to the delta games feature.

Information of interest on Rotowire was often represented by a tuple of conversions and attempts, and it was scraped as such. Rather than include these raw numbers of conversions and attempts in a model, percentages were calculated and stored as decimals. Even though match durations are fixed to a degree (there is a preset maximum number of sets), the number of points in a match greatly varies, therefore metrics such as the number of first serves converted/attempted also greatly vary. Those metrics ultimately don't matter in this context, but the *percentage* of first serves converted does matter.

Another vital step in the data preparation process included manipulating the structure of the match statistic dataframe. The information scraped from Rotowire was structured so that the unit of observation for the dataframe was *date match*, meaning that each row of the dataframe corresponded to a certain match on a certain date, encapsulating match statistics for two separate players. A function was written to split each row into two so that the unit of observation would instead be *player match*, where each row corresponds to an individual player on some date. This process was necessary because indexing is a vital component of forecasting, and adjusting the dataframe's unit of observation to *player match* made indexing far simpler.

[6]If player A wins a tiebreaker 7-4 over player B, the scraped numbers would yield 7 for player A and 64 for player B. This obviously needed to be corrected in all instances.

2.3.2 Dataframe Joins

Once any necessary transformations were applied to the data, each source's respective scraping yield was joined together to form one comprehensive dataframe. Data was scraped or notated from four sources and the subsequent step involved consolidating it. Figure 2.2 below displays noteworthy variables from each dataframe, as well as any inter-dataframe relationships that were utilized for joining purposes.

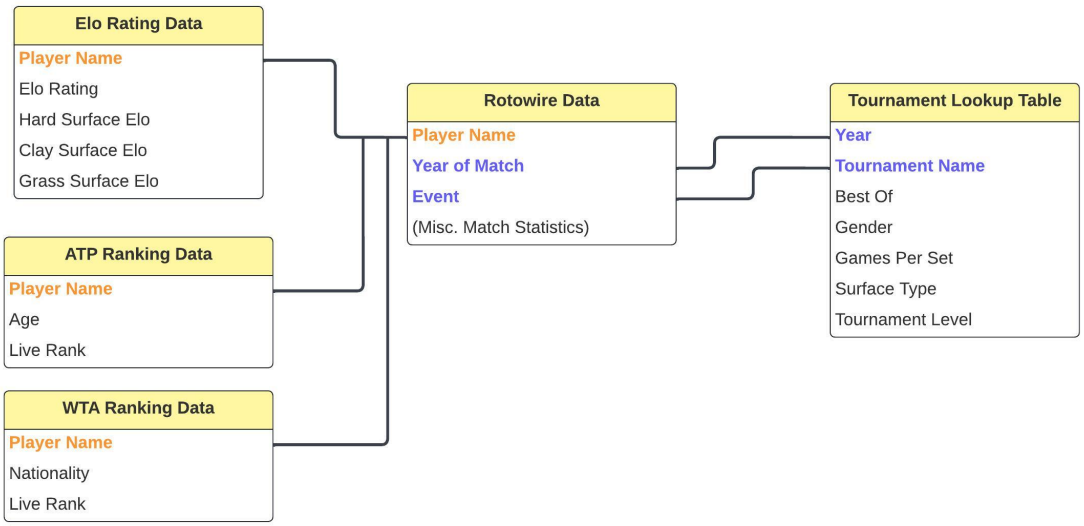

Figure 2.2: Dataframe Joining Diagram

The dataframe consisting of the match statistics from Rotowire contained the bulk of the information, and each of the other information dataframes were joined with it. As figure 2.2 reveals, the elo rating data, ATP ranking data, and the WTA ranking data were all joined with the Rotowire match data on "player name." The tournament lookup table and the match data were joined on both "year" and "tournament name." It is infrequent but possible for a tournament's level to change year over year; therefore, including "year" in the join ensured that any potential level changes would be captured, as the lookup table was manually updated to include every tournament each year. These joins were only performed on the newly-scraped matches each day, as it would not make sense to pair current player rankings or elo ratings with matches from the past. After these joins were performed on

only the new matches, the result was bound with the dataframe consisting of the rest of the historical match data.

2.3.3 Variable Mapping

Any type of head-to-head, winner-take-all competition will oftentimes utilize a draw format. In this format, individuals or teams compete against some opponent and the winner moves onto the next round. This process continues and the number of remaining competitors gets cut in half each round until a champion is crowned[7]. Professional tennis tournaments operate in such a manner, and each tournament has a preset number of spots for competitors; for example, it was known ahead of time that the 2024 Australian Open would have 128 main draw spots available. Tennis is unique in that there is also a *qualifying* draw that is played out prior to the main draw matches commencing. A select few players who perform well in qualifying will be awarded a spot in the main draw. Imagine that every professional player were healthy and a Grand Slam tournament was coming up. Grand Slam tournaments have 128 main draw spots to fill, but they also award 16 qualifying spots (and 8 wild cards[8]). In this theoretical situation, the top 104 ranked players will directly enter the main draw alongside 8 wild cards. The qualifying draw would feature the next 119 players in the rankings, as well as 9 *qualifying wild cards*. Players need to win 3 straight matches to be awarded one of the final 16 main draw spots [Had] [9].

Any qualification matches scraped from Rotowire were labeled as such in the tournament lookup table, either as "qualification" or "challenger qualification." Most professional tournaments are categorized by way of a numeric descriptor, such as "Challenger 125" or "WTA

[7]For Grand Slam tournaments which feature 128 players in the main draw, the eventual champion needs to win 7 main draw matches.

[8]Players who receive a wild card into a tournament are chosen at the discretion of the tournament organizers, but they are typically awarded to local players, up-and-coming young players, or highly ranked players who suffered an injury in the past and fell in the rankings. To provide an example, each year the winner of the Boys' 18s National Championship (played in Kalamazoo, MI) receives an automatic wildcard into the US Open main draw.

[9]As improbable as it was, Emma Raducanu won the 2021 US Open after winning 3 straight qualifying matches to receive a main draw spot, and then winning 7 main draw matches to win the tournament.

500," which specifies the number of ATP or WTA points that are awarded to the tournament's winner. Higher level tournaments award a higher number of points and more prize money, both of which attract higher level players. For this reason, higher level tournaments are mapped to higher numerical values. The majority of the tournament levels already include a numeric representation, but the "qualification" and "challenger qualification" labels needed to be mapped to numeric values. Table 2.1 below displays the manner in which the levels were mapped, which was necessary prior to the variable's inclusion in the regression models.

Initial Tournament Level	**Mapped Value**
ATP Challenger Qualification	0.5
ATP Challenger 50	0.5
ATP Challenger 75	0.75
ATP Challenger 80	0.8
ATP Challenger 90	0.9
ATP Challenger 100	1.0
ATP Challenger 125	1.25
ATP Challenger 175	1.75
ATP/WTA Qualification	2.5
ATP/WTA 250	2.5
ATP/WTA 500	5
ATP/WTA 1000	10
ATP/WTA Finals	15
Grand Slam	20

Table 2.1: Tournament Level Mapping

The pattern that may be deduced in the table above is not by accident. The mapping process utilized the number of points each tournament awarded to its respective champion

and simply divided it by 100. The ATP/WTA Finals award each winner 1500 points[10] so that level was mapped accordingly to 15. Any matches that were labeled as "challenger qualification" were given an equal value to that of the lowest-level Challenger event: Challenger 50 tournaments. ATP/WTA qualification matches were marked equivalent to ATP/WTA 250 events, once again the lowest ATP/WTA level events. Since larger tournaments have a higher number of main draw spots than smaller tournaments[11] and the rankings of the players who enter smaller tournaments aren't as sequentially constant[12], the caliber of each ATP/WTA qualification match was deemed equal regardless of the tournament.

In the ultimate model, tournament level served as both an *a priori* variable, because the tournament level is known prior to the match occuring, as well as an input to the adaptive least squares model. The latter inclusion provided filtered player ranks and opponent ranks, which provided insight into the filtered average ranking of the opponents that each player faces. Further detailed explanation in regard to variable selection will be provided in a later section.

2.4 Final Variable Pool

The process of scraping information from the numerous sources outlined above yielded a slew of variables, with the *a priori* variables shown below in table 2.2.

[10]The ATP and WTA Finals are unique in that the top 8 men and the top 8 women play round-robin style tournaments. The scoring is such that an *undefeated champion* will receive 1500 points[ATPb].

[11]Grand Slams have 128 main draw spots up for grabs, whereas 1000/500/250-level tournaments typically have 24-32 spots.

[12]Grand Slams can count on all of the highest ranked players to enter unless injury strikes, but that's not the case for other tournaments. When main draw entry rankings shift lower, qualifying player rankings also shift lower.

***A Priori* Variable**	**Description**
Gender	Dummy variable, denotes male or female match
Surface Type	Dummy variable, denotes whether match was played on hard, clay, grass, or carpet surface
Best Of	Maximum number of sets the match could go
Games Per Set	Number of games to which each set is played in the match
Player Rank	Live rank of the player
Elo Rating	Current overall elo rating of the player (not surface specific)
Tournament Level	Tournament level of the match

Table 2.2: *A Priori* Variables

In addition to the variables shown above, *a posteriori* variables were also included and are listed below in table 2.3.

***A Posteriori* Variable**	**Description**
1st Serve Percentage (%)	Percentage of successful first serves
1st Serve Win %	Percentage of first serve points won
2nd Serve Win %	Percentage of second serve points won
Service Points Won %	Percentage of all service points won
Service Games Won %	Percentage of service games won
Aces	Number of aces (unreturned serves)
1st Serve Return Points Won %	Percentage of first serve return points won
2nd Serve Return Points Won %	Percentage of second serve return points won
Return Points Won %	Percentage of all return points won
Return Games Won %	Percentage of return games won
Break Point Conversion %	Percentage of break points converted (as returner)
Break Point Save %	Percentage of break points saved (as server)
Winners	Number of winners (service aces included)
Unforced Errors	Number of unforced errors
Total Points Won %	Percentage of total points won
Total Games Won	Number of games won
Total Sets Won	Number of sets won
Match Won	Binary variable (won: 1, lost: 0)
Delta Game	Difference in total games with respect to opponent
Delta Set	Difference in total sets with respect to opponent

Table 2.3: *A Posteriori* Variables

The descriptions of the variables above are (for the most part) self-explanatory. They are considered to be *a posteriori* as they are only known after a given match has ended. The information is not known pre-match. Further down the project's road, time-variant averages of the *a posteriori* variables are calculated and paired with the *a priori* variables to yield the design matrix, $\mathbf{X}$. This design matrix is multiplied with a calculated vector of slopes, $\hat{\boldsymbol{\beta}}$,

yielding a forecast for the outcome variable, delta set.

The data transformation process yielded a design matrix comprised of an ever-changing number of rows with seventy features. After roughly seventeen full months of match scraping (Nov 2022–Mar 2024), the constructed matrix holds over forty-two thousand unique player matches. This number will continue to rise as professional matches continue to be played and the corrresponding match data accumulates.

2.5 Data Management

As previously mentioned, the information of interest on Rotowire was only available for a limited number of days. In the event of some computer memory setback, each webpage of match statistics was saved to disk as an HTML file. This ensured that the data source remained accessible in case questions of validity arose, or if the constructed dataframe was mistakenly altered. Another implementation of sound data management involved the use of Amazon Web Services Simple Storage Service (AWS S3) for cloud storage. The tens of thousands of match stat webpage files, along with the numerous programming script files and data files, were all uploaded regularly to an AWS S3 bucket to further avoid a potential loss of information and model results.

2.6 Adaptive Least Squares & Kalman Filtering

Once properly formatted and transformed, the data could be utilized within the model-creation and improvement process. The explanation of this project's modeling methods centers around an adapted version of the multiple linear regression model, known as adaptive least squares, which incorporates a concept called Kalman filtering. These methods are commonly applied to time series data, and are described in further detail below. An explanation of the least squares linear regression method is described first to set the table for the implemented ideas of adaptive least squares and Kalman filtering.

2.6.1 Least Squares Regression Theory

Simple linear regression involves some numerical output variable, y_i, and some predictor, x_i. There is also a "slope" parameter paired with x_i, represented by β_i. An error term, or residual, ϵ_i and an intercept term β_0 also exist. Combining each of the above elements together into a formula yields the following, where $i = 1, ..., n$, with n representing the number of model predictors, or features.

$$y_i = \beta_0 + x_i\beta_i + \epsilon_i \tag{2.1}$$

The equation above may also be written in matrix form, which is the exact same equation written in a more concise form. For multiple linear regression which involves numerous predictor variables, the following matrix form is preferred:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{2.2}$$

Here, $\boldsymbol{Y}$ and $\boldsymbol{\epsilon}$ are n x 1 column vectors. The design matrix, $\boldsymbol{X}$, is an n x k matrix, where k is equal the number of predictors (including the intercept). $\boldsymbol{\beta}$ must therefore be a k x 1 column vector so that the matrix multiplication works out. To figure out the estimated values for the vector of slopes, $\hat{\boldsymbol{\beta}}$, for multiple linear regression, the loss function is introduced. The loss function is utilized to minimize $\hat{\boldsymbol{\beta}}$ by finding a global minimum via setting the derivative of the loss function equal to zero.

As was defined above, $\hat{\boldsymbol{\beta}}$ serves as an estimate of $\boldsymbol{\beta}$. The estimate for the forecasted values of $\boldsymbol{Y}$ ($\hat{\boldsymbol{Y}}$) is therefore defined as such:

$$\hat{\boldsymbol{Y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} \tag{2.3}$$

The corresponding vector of residuals can be written as follows:

$$\boldsymbol{\epsilon} = \boldsymbol{Y} - \hat{\boldsymbol{Y}} \tag{2.4}$$

$$\boldsymbol{\epsilon} = \boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}} \tag{2.5}$$

The method of least squares involves minimizing the sum of the squared error, which is represented as $\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}$. This can be rewritten as such:

$$\min(\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}) = \min\left((\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^T(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})\right) \tag{2.6}$$

To minimize the right side of equation 2.6 above, the derivative is taken with respect to $\hat{\boldsymbol{\beta}}$ and set equal to zero:

$$\frac{d}{d\hat{\boldsymbol{\beta}}}\left((\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^T(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})\right) = 0 \tag{2.7}$$

Performing some linear algebra yields the following equation:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}(\boldsymbol{X}^T\boldsymbol{Y}) \tag{2.8}$$

Equation 2.8 above reveals the method of calculating $\hat{\boldsymbol{\beta}}$ in multiple linear regression. In this instance, $\hat{\boldsymbol{\beta}}$ does not change; it is time-invariant[13]. The process of calculating $\hat{\boldsymbol{\beta}}$ also involves a static, time-invariant design matrix, $\boldsymbol{X}$. As the time-invariant nature depicts, the values for $\hat{\boldsymbol{\beta}}$ are calculated using all of the data from the design matrix without index consideration.

The equations and their ultimate result shown above are incredibly powerful, but disciplines such as sports forecasting or financial market forecasting require a slightly different approach. As this project exists to forecast the outcomes of professional tennis matches, the constructed models implement additional wrinkles stemming from the method of adaptive least squares and the Kalman filter. The underlying mathematics is comparable, but there are some major differences that will be introduced below.

2.6.2 Adaptive Least Squares and Kalman Filtering Theory

From a bird's eye view, one can imagine that a vital component of sports forecasting *must* be time. The context that time provides matters greatly: past performance will obviously

[13]To provide a preview for the paragraphs to come, another way to say "time-invariant" is "index-invariant."

be taken into consideration when forecasting, yet should more recent matches influence the forecast to some higher degree? Encapsulating this type of variable uncovers a glimpse into historical performance on a deeper level, as signal-to-noise ratios can be leveraged to "weight" past matches differently based on how deep in the past they were played (more on this later). Since tennis matches are completed nearly every day, the new metrics from said matches will ultimately cause the values within $\hat{\boldsymbol{\beta}}$ to undergo slight adjustments[14]. Performing these calculations on a daily basis over such a large dataset is inefficient and can get computationally expensive. Therein lies one of the extremely valuable extensions of adaptive least squares: time-variant slopes.

Not only are the values for $\hat{\boldsymbol{\beta}}$ updated whenever new tennis match statistics are processed, their calculations are solely dependent on the previous day's best approximation for the values of $\hat{\boldsymbol{\beta}}$. This method is valid because time-variant modifications are also made to the design matrix when making delta set forecasts rather than simply calculating slopes from a static matrix. This is where the Kalman filter is applied. Just as the values for $\hat{\boldsymbol{\beta}}$ are calculated in a time-variant manner, the contents of the design matrix are also time-variant. The Kalman filter computes a rolling average[15] of each metric for each individual player, introducing the notion of *state*. At any given state, there exists some measure of all inputs that were applied to the filter up to that state [Hay96]. Any given delta set forecast is constructed from far more than just a $\hat{\boldsymbol{\beta}}$ vector and a static design matrix.

Consider a tennis match in which Jannik Sinner is opposing Carlos Alcaraz[16]. The $\hat{\boldsymbol{\beta}}$ vector resulting from the adaptive least squares regression model will have been updated based on information stretching all the way up to time $t-1$. This same vector of slopes will be applied to every match on the following day, time t, to generate delta set forecasts. If

[14]Adjustments are slight because scraping information from ± 50 new matches one day won't drastically affect the already-calculated $\hat{\boldsymbol{\beta}}$ from the tens of thousands of matches already scraped from the past.

[15]Moving forward, "rolling average" is synonymous with "filtered."

[16]At the time of writing, Sinner sits at #4 in the world and Alcaraz at #2.

the design matrix were static, the resulting output would only encapsulate data from each player's one previous match. Instead, rolling averages for every metric (see metrics in table 2.3 are calculated for Jannik and rolling averages are calculated for Carlos based on *all* of their respective past matches. The model leverages indices of the constructed dataframe to seek out historical matches for each player and features the filtered values of each metric for each respective player within X_i. Therefore even though the $\hat{\boldsymbol{\beta}}$ values are player-agnostic, the design matrix $\boldsymbol{X}$, full of filtered metrics for each individual player, is most definitely not. Not only are rolling averages of the metrics calculated, but averages for metrics given up, metrics of the opponent, and metrics that the opponent have given up are generated and paired together in X_i. The number of features that the model consists of quadruples[17] with the inclusion of these rolling metrics.

Returning to the specific Sinner vs. Alcaraz example, the process of forecasting the winner of a match between Jannik and Carlos would have the example information below at its disposal to make as accurate a forecast as possible. Table 2.4 concisely displays how each of the metrics is "expanded" into four different components by listing what variables come from the aces statistic. The actual model includes filtered values for every *a priori* and *a posteriori* variable, as well as the same *a priori* variables in an unfiltered manner. This allowed the model to leverage current player rankings at the time of the match in addition to filtered averages for player ranking, meaning that in the above example, the filtered rankings for Jannik revealed a rolling average of his ranking, Alcaraz's ranking, the average rank of his historical opponents, and that of Alcaraz's opponents. Quite a bit of information was captured through the filtering process beyond raw variable inclusion. In table 2.4 below, filtered *a posteriori* values are displayed in bold, while *a priori* variables are in plain text[18].

[17]Each feature is transformed from being a standalone value to being represented in four separate ways: player scored, opponent scored, player allowed and opponent allowed. For example, instead of including a raw number of aces for a player, the model calculates and incorporates the average number of aces the player hits, average aces the opponent hits, plus the average number of aces the player allows and the average aces the opponent allows.

[18]As a reminder, the (plain text) *a priori* variables are included both as filtered values and as unfiltered values.

	Jannik Sinner	**Carlos Alcaraz**
Player Rank	4	2
Elo Rating	2.32	2.29
Surface Type	hard	hard
BestOf	5	5
GamesPerSet	6	6
Gender	1	1
Tournament Level	20	20
Player AVG Aces	**10.3**	**9.9**
Opponent AVG Aces	**9.9**	**10.3**
Player AVG Aces Yielded	**5.2**	**4.5**
Opponent AVG Aces Yielded	**4.5**	**5.2**

Table 2.4: Abbreviated Set of Model Features

It is necessary to stress once again that table 2.4 is an abbreviated display showing how each metric gets "expanded" into four through the filtering process. A model containing twenty features will ultimately contain eighty features after filtering. This project featured a model that included filtered values for every variable in tables 2.2 and 2.3, as well as unfiltered variables in table 2.2 (bestOf, gamesPerSet, player rank, tournament level, and elo rating). It is apparent that some of these filtered pieces of information represent the same metric if Sinner and Alcaraz are competing against one another: the filtered number of aces that Sinner hits are the same as the filtered number of aces of Alcaraz's opponent. Similarly, the filtered number of aces that Sinner gives up is the same as the filtered number of aces that Alcaraz's opopnent gives up, and so on. Every metric for Sinner will reside as a unit of observation in one row, and every metric for Alcaraz will reside as another unit of observation in a row directly beneath. This process was carried out for every match that was

scraped from the aforementioned data sources, yielding a sizeable matrix of filtered features that was utilized to formulate accurate delta set forecasts at any given point in time.

As the design matrix is comprised of rolling averages, slopes for time t are calculated from the slopes for time $t-1$, those for $t-1$ are calculated from $t-2$, and so on. The Kalman filter allows for the calculation of a match forecast at any point in time, leveraging only the past and current history of the data at any point of interest [McC05]. The values for $\hat{\boldsymbol{\beta}}$ are not re-calculated over the entire dataset every time new match metrics are available, rather the slight effects of the new information on the previous information are concisely captured. This process is incredibly efficient and embraces the consideration of time in forecasting as best it can. The underlying math behind adaptive least squares [Zes09] is shown below:

$$\hat{\boldsymbol{\beta}}_{t-1} = \widehat{\boldsymbol{L}_{\boldsymbol{xx}}}_{t-1}^{-1} \; \widehat{\boldsymbol{l}_{\boldsymbol{xy}}}_{t-1} \tag{2.9}$$

$$\widehat{\boldsymbol{L}_{\boldsymbol{xx}}}_{t} = \widehat{\boldsymbol{L}_{\boldsymbol{xx}}}_{t-1} + \; \mathbf{K}_t \; \cdot \; \left(\boldsymbol{x}_t^T \boldsymbol{x}_t - \widehat{\boldsymbol{L}_{\boldsymbol{xx}}}_{t-1} \right) \tag{2.10}$$

$$\widehat{\boldsymbol{l}_{\boldsymbol{xy}}}_{t} = \widehat{\boldsymbol{l}_{\boldsymbol{xy}}}_{t-1} + \; \mathbf{K}_t \; \cdot \; \left(\boldsymbol{x}_t^T \boldsymbol{y}_t - \widehat{\boldsymbol{l}_{\boldsymbol{xy}}}_{t-1} \right) \tag{2.11}$$

If you squint hard enough you'll see that the structure of adaptive least squares doesn't stray too terribly far from the "vanilla" version of least squares, but there are some additional properties, namely the consideration of time and the Kalman gain (K_t). The covariance objects ($\widehat{\boldsymbol{L}_{\boldsymbol{xx}}}_{t}$ & $\widehat{\boldsymbol{l}_{\boldsymbol{xy}}}_{t}$) calculated for today, time t, are calculated using information aggregated through time t, meaning that these calculations leverage time t results. For forecasting purposes, it may be helpful to consider that a set of forecasts for matches that will be played today (t) will be calculated using the most updated information from *before today*. Both the values within $\hat{\boldsymbol{\beta}}$ and the design matrix $\boldsymbol{X}$ that were updated at time $t-1$ will be used for the forecasts made today. This is because the results for today (time t) are not yet known, therefore the most recent information available refers to the values calculated for matches before today (time $t-1$).

Another novel inclusion in the adaptive least squares calculation is the K_t variable, known as the Kalman gain. This value serves as a model hyperparameter that controls the degree to which the existing design matrix features are adjusted when new information is introduced. The Kalman gain K_t is influenced according to the values set for different signal-to-noise ratios, oftentimes referred to as "forgetting filters," which also serve as model hyperparameters. These signal-to-noise ratios were implemented to discern between the natural seasons that the ATP and WTA professional tours instill. Unfortunately for the players, there is not much of an offseason in the sport of tennis. Unlike other professional sports, tennis professionals choose what tournaments they compete in[19], so the "season" extends for nearly the entire year. Depending on what tournaments players compete in, there is only a short 1-2 month break spanning most of November and December[20]. As such, matches completed in more recent "seasons," or years, were treated with higher signal-to-noise ratios than matches from previous seasons. Hence, the further in the past some match metrics may be from, the less impact they have on the model, and the more they are "forgotten."

The values for the Kalman gain and the signal-to-noise ratios are somewhat intertwined; if the signal-to-noise ratios were set to zero, the Kalman gain would trace the following pattern, known as the "harmonic sequence": 1, $\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$, etc. Every subsequent set of match statistics would have a sequentially smaller Kalman gain as the amount of already-present data would continue to accumulate. It turns out that the signal-to-noise ratios were not set to zero, rather they were optimally chosen to yield the best modeling results. If the existing data does not include a large number of matches for some individual player, the match stats from a recent match will influence the design matrix to a greater degree, and the value for K_t will be higher. On the other hand, one newly-completed match for a player who has a large number of matches already scraped and processed will not affect the design matrix as much, and the value for K_t will be lower. In this sense, the contents of the design matrix are

[19]Player schedules are entirely made at their own discretion (assuming the presence of a sufficient ranking to have unrestricted options).

[20]A top eight player will compete in the ATP /WTA Finals which are played in November, enjoy the short offseason, and return to action in beginning-to-mid January in Australia.

not simply player-specific rolling averages of match statistics, rather they are time-variant, player-specific rolling averages that are ultimately influenced by the value of the Kalman gain, K_t. As players compete in more and more matches, the higher volume of gathered information stabilizes the model and the resulting forecasts improve. The manner in which the values for the Kalman gain and the signal-to-noise ratios were chosen will be explored further in the hyperparameter tuning section.

2.6.3 Logistic Mapping

The three adaptive least squares models each utilized different sets of features, yet all models were constructed to output forecasted values for the outcome variable of interest, delta set. At this point in the process, the adaptive least squares model successfully yielded delta set forecasts for every match that had been scraped. The delta set forecasts were formatted as a vector that shared the same index as the design matrix[21]. The next step involved mapping these forecasted values, which included both positive and negative numbers, to probabilities between 0 and 1. To do so, a time-variant sigmoid function was utilized, with the associated formula shown below [Dan24]:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2.12}$$

The resulting value of $\sigma(z)$ above is a function of z, the output of the adaptive least squares regression model which represents forecasted delta sets. $\sigma(z)$ is therefore the *forecasted* probability of some given player winning their match. For a match that is being played today, the adaptive least squares model uses the most recent information "before today" to generate a delta set forecast. This forecast is then fed into the sigmoid function shown above (as variable z), which maps the delta set value into a probability between 0 and 1. Because the adaptive least squares algorithm does not know what two rows are actually

[21]The common index made it quick and easy to pair the model results with the existing design matrix. The 42,000th element of the delta set vector corresponded to the results of the player-match unit of observation residing within the 42,000th row of the design matrix (Fabio Fognini's results from a match on Feb 14th against Abedallah Shelbayh at the Manama Challenger).

related, the probabilities of winning for two competitors more often than not will not sum exactly to 1, but they will be close. Once the hyperparameters are tuned and more match data is processed in the model, the sum of two competitors' probabilities should get very close to 1. This is a testament to the power of the algorithm: even though the model does not know that the two rows corresponding to match opponents are related, the sum of the two forecasted probabilities will get closer and closer to 1 in an independent manner.

2.6.4 Hyperparameter Tuning

The existence of hyperparameters was briefly alluded to in earlier sections. The value for the Kalman gain and the numerous signal-to-noise ratios served as hyperparameters to the adaptive least squares model. The logistic mapping process also involved a few hyperparameters covering initial slope and intercept values along with associated "forgetting" factors. To ensure that these parameters are defined so that the model generates as accurate of forecasts as possible, a metaheuristic algorithm was created in R with the ultimate goal of minimizing a performance metric of interest, the logistic cost. The logistic cost is a measurement of how far the forecasted outcome falls with respect to the actual outcome [Dan24], which in this case is the match outcome {0,1} with 0 signifying that the player lost the match and 1 signifying that the player won. To minimize the logistic cost, the vector of hyperparameters[22] had random noise applied to each element and the model would be executed with the new values. If the logistic cost was lower, the new vector of parameters was kept and the process continued. If the logistic cost was the same or higher, the new vector values were thrown out and the initial parameter vector once again had random noise applied to each element and the model was executed.

In addition to the signal-to-noise and logistic hyperparameters, a regularization parameter was also included within the adaptive least squares model. There is an appeal to having

[22]There were 12 total hyperparameters in this model: 8 for the adaptive least squares algorithm and 4 for the logistic regression algorithm.

all of the model's variables on similar scales, which is why player rankings and elo ratings were both scaled down. As the variables included in the model have differing variances, the process of scaled regularization penalizes the predictors evenly by adding a constant to the diagonal terms of the $\widehat{\boldsymbol{L}_{\boldsymbol{xx}}}$ matrix. This addition helps to avoid oversuppressing the results from specific predictor variables based on their scale. The presence of this scaled regularizer afforded the option to not scale variables, but rank and elo rating were still scaled down for good measure.

2.6.5 Modeling Application & Performance Comparison

It was of interest to compare multiple models with differing features in an attempt to uncover those that were most influential in forecasting the delta set metric (and subsequent outcome) of a tennis match. Three sets of features were constructed to serve as the inputs to three separate models, with the resulting logistic costs and prediction accuracies serving as comparison metrics. The first, "all-encompassing" model (model 1) included *all* of the variables listed in tables 2.2 and 2.3. This model involved throwing every scraped feature into the design matrix, and its results served as a baseline for the other two models. Model 2 analyzed the results of model 1 and isolates the top features that possess the highest t-statistic and lowest standard error with respect to the coefficients from the adaptive least squares model. Oftentimes simpler models perform better, so model 2 was created to quantify any differences. The final model, model 3, includes hand-picked features that the avid-tennis-fan author expects to be the most influential, mostly based off of prior tennis knowledge acquired from playing and watching the sport. The more-defining characteristic of model 3 is that it does not feature player ranking and elo rating variables, as these two variables are widely used when simple predictions of matches are made. Whoever is ranked higher is typically the favorite to win, which is not a sentiment solely present in the sport of tennis. Excluding rank and elo rating isolated the pure performance statistics and reduced the collective power of the *a priori* variables, which would lead to interesting comparisons. Such comparisons may assist in determining whether or not a model strictly leveraging filtered match statistics

performs nearly as well as a model incorporating player rank and elo rating.

The table below displays the features implemented within model 2. To once again explore model 1's features, please refer back to the contents within tables 2.2 and 2.3.

a priori	***a posteriori (filtered)***
Player Rank	Player Rank
Elo Rating	Elo Rating
Surface Type	Winners
BestOf	Unforced Errors
GamesPerSet	Aces
Gender	Double Faults
Tournament Level	Tournament Level
	Delta Game
	Delta Set
	Total Sets Won
	Total Games Won

Table 2.5: Model 2 Variables

Compared to model 1, model 2 includes far fewer variables, keeping only the most "influential." Model 3 is a subset of model 1, including hand-picked features but notably excluding player rank and elo rating from both the *a priori* and filtered variable pools. Variables featured in model 3 are shown in table 2.6 below.

a priori	*a posteriori (filtered)*	
BestOf	First Serve Pct	Aces
GamesPerSet	First Serve Win Pct	Double Faults
Tournament Level	Service Pts Won Pct	First Serve Return Pct
Surface Type	Service Games Won Pct	Return Games Won Pct
Gender	Break Pt Conversion Pct	Return Pts Won Pct
	Break Pt Save Pct	Winners
	Total Games Won	Unforced Errors
	Total Sets Won	Delta Set
	Total Pts Won Pct	Delta Game

Table 2.6: Model 3 Variables

The performances of the three respective models were quantified by analyzing logistic costs, which measure how far the probability predictions stray from the actual match outcomes {0,1}, and the accuracies of the forecasts with respect to actual match outcomes. Variable importance was also documented for each model by observing the resulting $\hat{\boldsymbol{\beta}}$ magnitudes and t-statistics.

2.7 Probability Output Interface

With the best model storming into the forefront due to its superior performance, its forecasting abilities were applied to tennis matches that had not yet occurred. An interface was created that transmits three prompts to the user: requests for the names of two opposing players and the corresponding event at which they are competing. A program was written to recognize each of the inputted player names and join the corresponding live ranking and elo rating with each player as *a priori* information. The inputted event is joined with the tournament lookup table, yielding more *a priori* information about the tournament.

Once the inputted information is automatically prepped behind the scenes, it serves as input to the model. The user does not experience any of these intermediate steps; the only process component requiring any user activity is the initial insertion of player and event names. The model outputs two rows of information, each containing the following: player name, player rank, event name, probability of winning, forecasted delta sets, and the forecasted winner[23].

[23]This can be deduced from the winning probabilities. The player who has the higher probability of winning is denoted as the model's forecasted winner.

CHAPTER 3

Results

3.1 Modeling

As new match statistics continued to be available on a daily basis, only matches played between November 12th, 2022 and February 22nd, 2024 (inclusive) were included when analyzing model performance. These date parameters resulted in 42,872 unique player-match units of observation, further reduced to 34,034 after applying a mask to the dataframe. The mask excluded the first 5,000 indexed player-matches and excluded players who had not yet played more than three matches. This allowed for a period of "burn-in" which ensured that forecasts were being made only on units of observation that were supported by a substantial amount of data. Generating a forecast for a player who had previously only appeared once or twice in the dataframe would not yield an accurate nor well-supported result. Once the algorithms were completed, the distributions of the forecasted delta set outcomes were visualized below in figure 3.1:

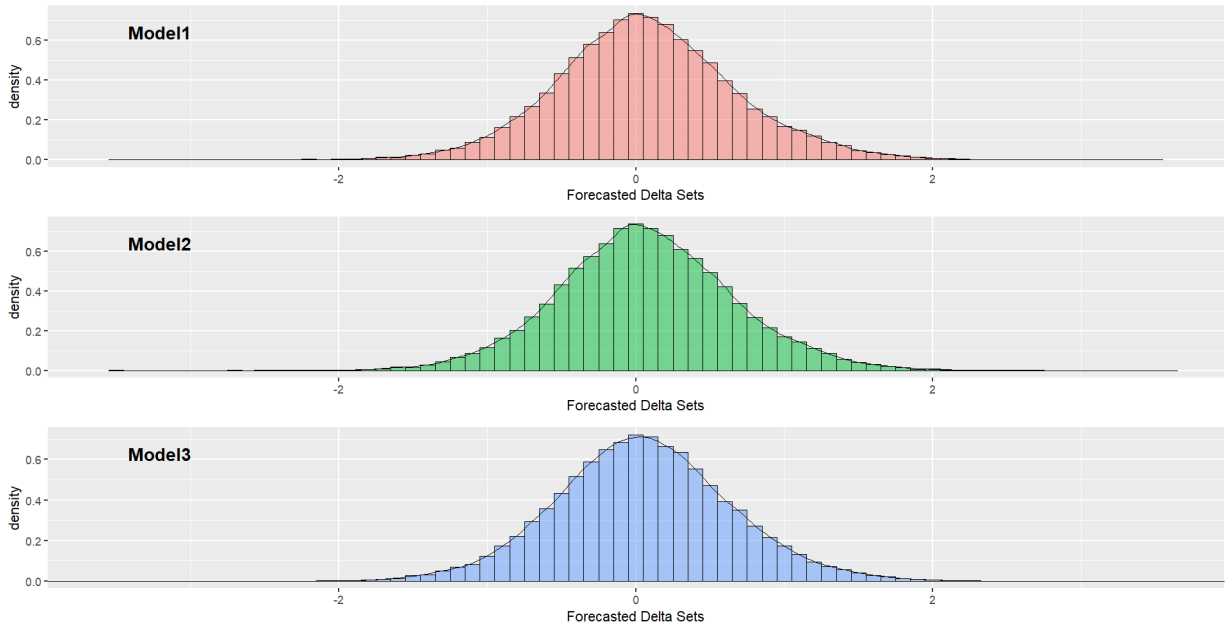

Figure 3.1: Distribution of Forecasted Delta Set Values

The differences in the three distributions are slight, yet each is centered around zero. This should be the case, as half of the units of observation should feature positive delta set forecasts and the other half negative. Once the delta set values were mapped to probabilities via the sigmoid function, the manner in which this process occurred could be analyzed. The true match outcome for each unit of observation is also shown to reveal any meaningful patterns. Figures 3.2, 3.3 and 3.4 once again display results from models 1, 2 and 3, respectively.

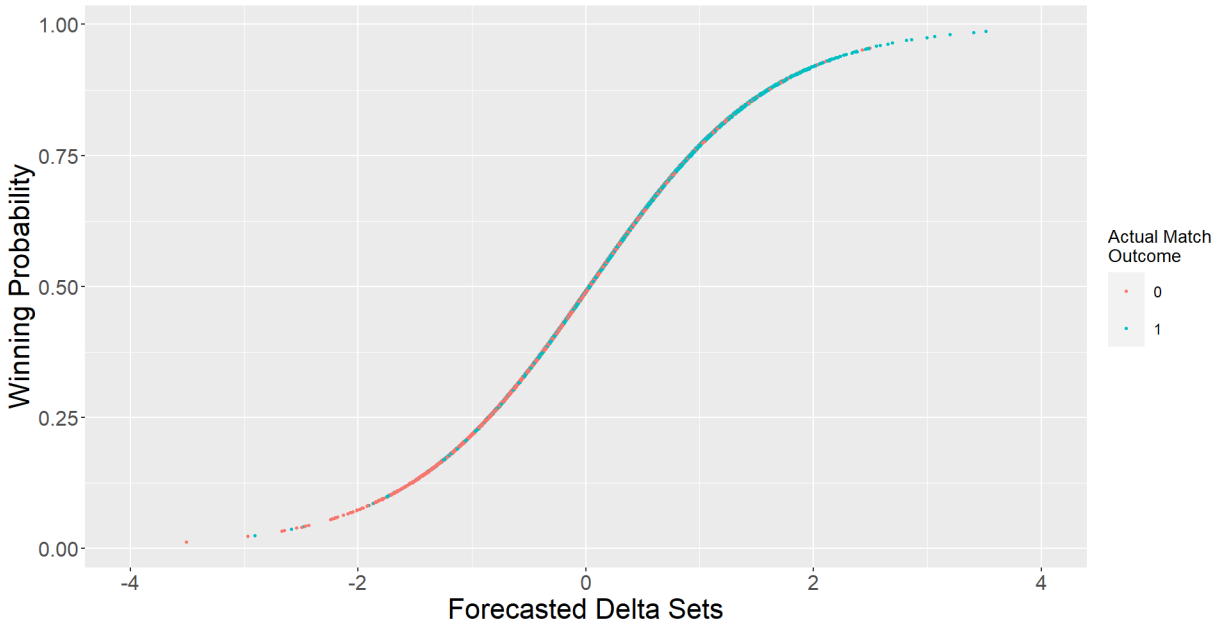

Figure 3.2: Model 1 Sigmoid Curve

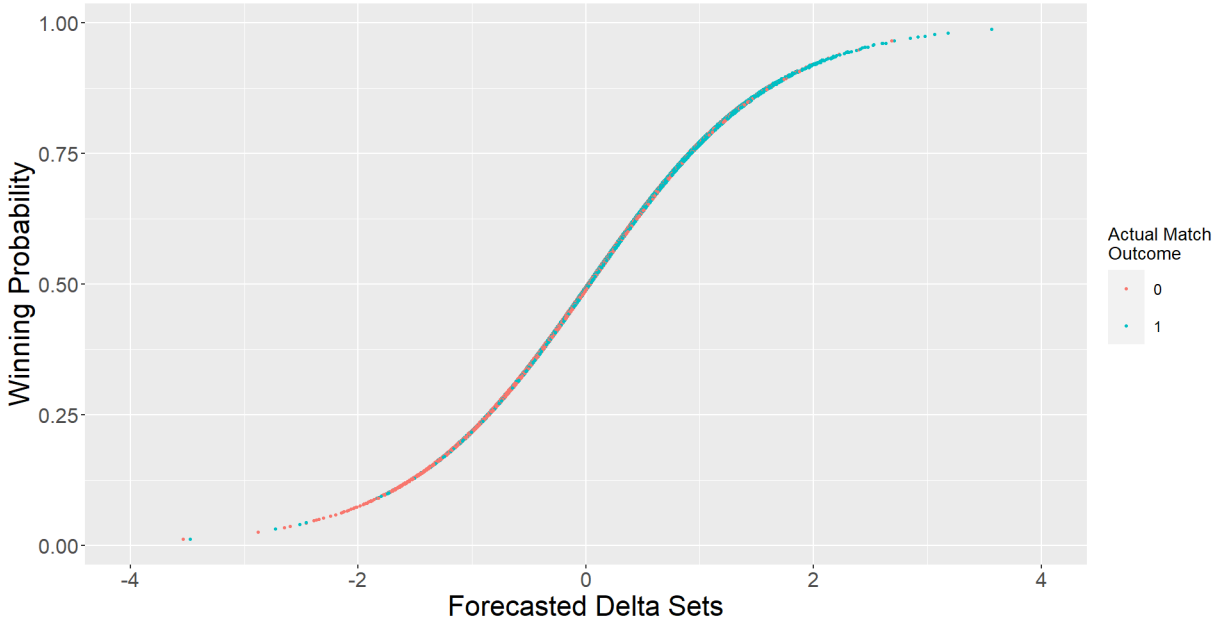

Figure 3.3: Model 2 Sigmoid Curve

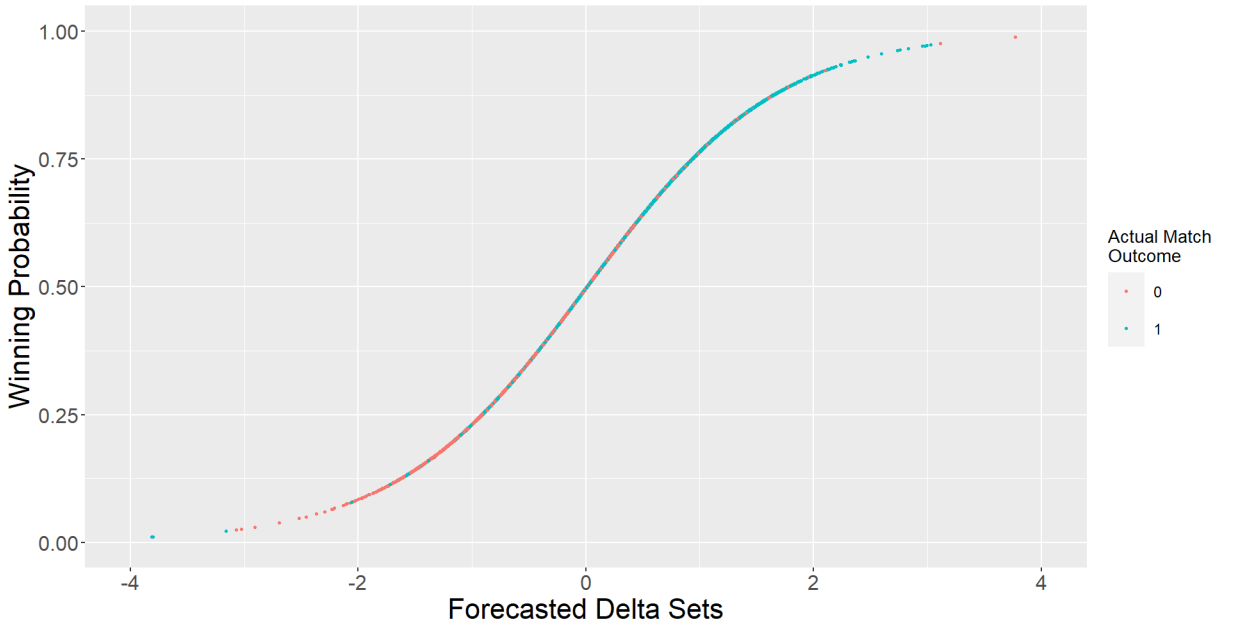

Figure 3.4: Model 3 Sigmoid Curve

It is once again difficult to visually isolate differences between the three sigmoid transformation visualizations above. All three sigmoid curves are asymptotically bounded at 0 and 1 and each curve's "S" shape is nearly identical to the others. The noticeable differences are seen in the actual model forecasts, and are most obvious when looking at the tails of the curve, which show that the resulting forecasts are indeed different with respect to input features. Each model appears to have performed fairly well, as there is a distinct difference in each plot in which forecasted winning probabilities above 0.50 oftentimes culminated in a win, and probabilities below 0.50 oftentimes culminated in a loss. As well as the models can perform, when delta set forecasts hover around 0 it becomes much more difficult to delineate between a win and a loss. Matches yielding such forecasts could be considered "coin flips," where the two opponents are so close in terms of historical performance that there is no strong favorite. This explains why wins and losses appear to be more inconsistent when delta set forecasts hover close to 0. This aligns with how the model "should" work, because more extreme forecasts originate from substantial differences in the data, which is the raw representation of how each player performs on average. Forecasted probabilities closer to 0.50 are more or less coin flips, and the match could realistically culminate in either player

winning.

Further visualization of the results includes analyzing the degree to which player rankings affect the model forecasts. Since player rank serves as the standard method of comparing two players, its inclusion (or exclusion) in the models was singled out. Figures 3.5, 3.6 and 3.7 below display the delta set forecasts with respect to the player rankings at the time of the match. The true outcome is also shown.

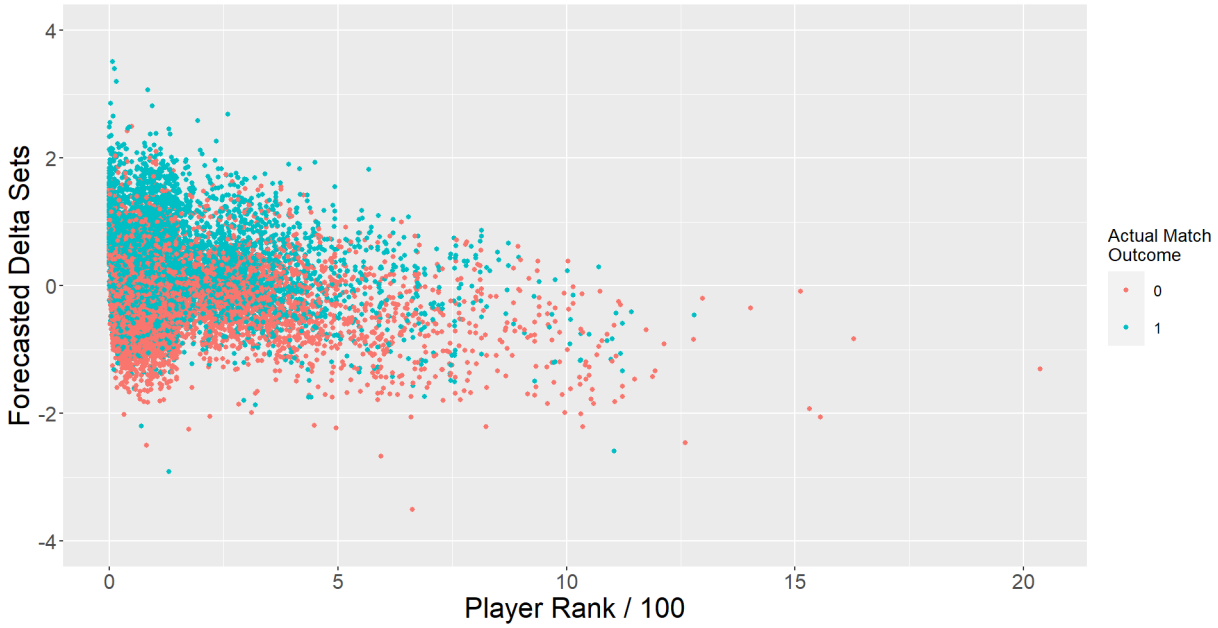

Figure 3.5: Model 1 Forecasts and Outcomes by Player Rank

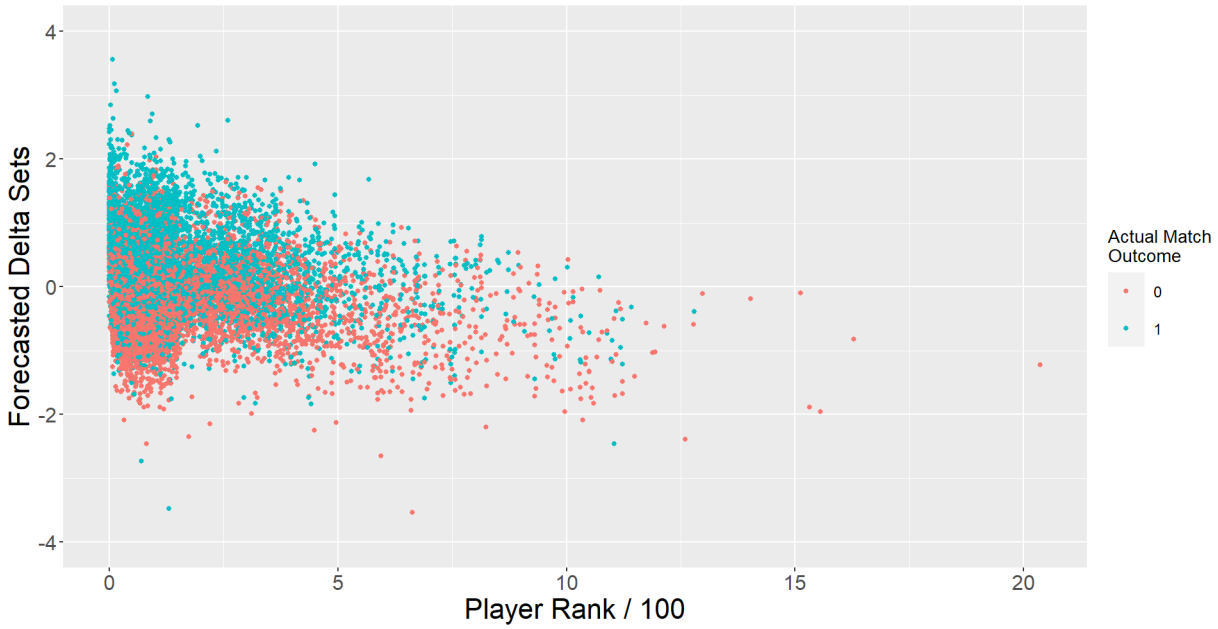

Figure 3.6: Model 2 Forecasts and Outcomes by Player Rank

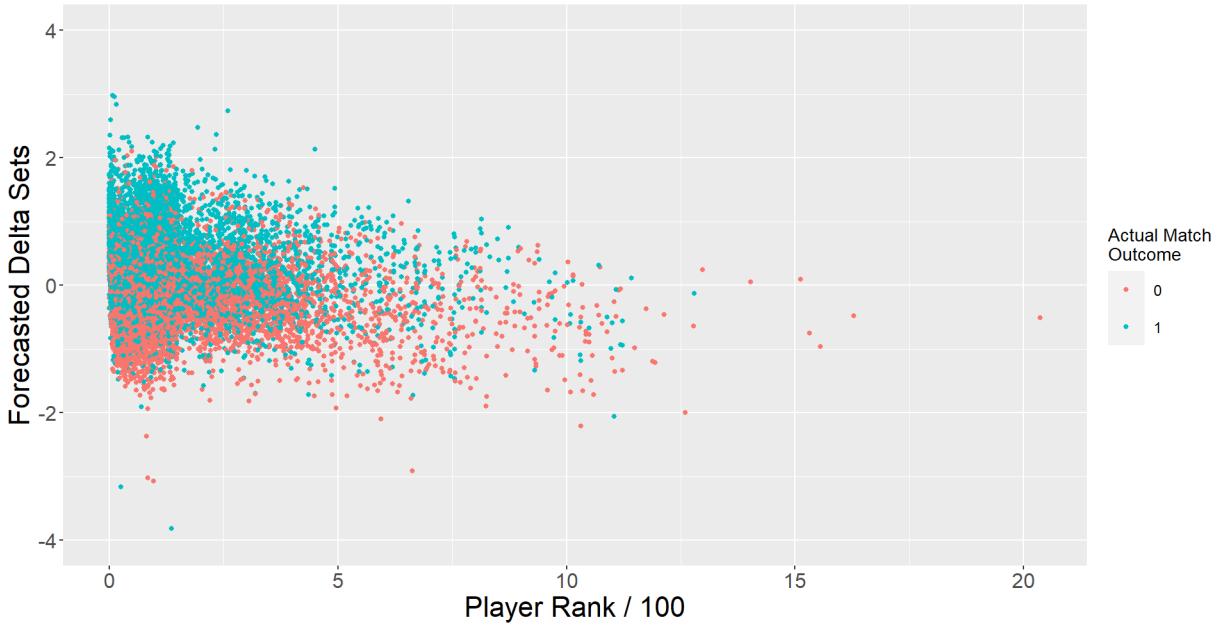

Figure 3.7: Model 3 Forecasts and Outcomes by Player Rank

All three visualizations look fairly similar, revealing the delta set forecasts getting slightly smaller or more negative when the player ranking is lower (the larger the ranking number). An interesting element is that model 3, generating the results shown in figure 3.7, does not

include player ranking or elo ratings as featured variables when calculating forecasts. While it is tough to discern any exact patterns from the three scatterplots, the fact that model 3's pattern of forecasts looks very similar to the others reveals that the strength of the models is not solely due to the presence of rankings nor elo ratings. The raw match statistics tell enough of a story on their own to construct a model that behaves in a comparable manner.

The visualizations thus far within this section have introduced general behaviors of the three models, but do not reveal specific performance numbers. Upon construction of the three models and completion of the appropriate hyperparameter tuning, the following performance metrics in table 3.1 allowed for deeper comparison.

	Logistic Cost	**Match Forecast Accuracy**
Model 1	1.27377	63.54%
Model 2	1.27815	63.07%
Model 3	1.28629	62.48%

Table 3.1: Model Performance

Model 1, which served as the "all-encompassing" model featuring everything in tables 2.2 and 2.3, resulted in the best performance of the three models, correctly forecasting 63.54% of the match outcomes while boasting the lowest logistic cost value at 1.27377. This process once again called on the model to make two independent[1] forecasts per match, one for each player. Model 2 had reduced dimensionality by including just the best-performing features from model 1. Ultimately model 2 performed *slightly* worse, possessing a logistic cost of 1.27815 and a forecasting accuracy of 63.07%. The third and final model included hand-picked features that the author deemed influential, notably excluding player ranking and elo rating as both *a priori* and filtered values. The purpose of this exclusion was to see to what

[1]While the forecasts were indeed independent, some information utilized in each set of two forecasts was the same: namely, player stats for player A would be the same as opponent stats for player B. Still, the model had no knowledge that two sequential player-match units of observation were related.

degree knowledge about rankings and elo ratings affected model performance. Surprisingly, the exclusion of these two variables did not drastically affect performance. Model 3 did have the highest logistic cost and the lowest forecasting accuracy of the three models, 1.28629 and 62.48%, respectively, but these numbers still well outperformed expectations.

It was of interest to compare the model results with match prediction accuracies when the higher-ranked player is predicted to win. If a player ranked number 9 in the world were opposing a player ranked 62, the former player would be the predicted winner. If the predictions based solely on rank were compared to the actual match outcomes, this method of predicting would result in an accuracy of 61.72%. The inclusion of the filtered match statistics ultimately culminated in a nearly 2% improvement in match forecasting accuracy compared to simply choosing a winner based on ranking. This may appear to be a marginal difference, but even the slightest of differences can be incredibly valuable when it comes to the accuracy of future forecasting. Future extensions to this project that have the potential to bolster model forecasting accuracy will be described in detail in later sections.

When it comes to analyzing feature importance within the models, the regression slopes, t-statistics and standard error values are of interest. Regression results for model 1 are displayed in table 3.2 on the following pages. There are four variable prefixes in the first column: PS, OPS, PA and OPA. PS refers to "player scored" or the player's own metrics. OPS refers to "opposing player scored" or the player's opponent's metrics. PA refers to "player against" or what metrics the player "gives up." Finally, OPA refers to "opposing player against" or the metrics that the opponent "gives up."

Table 3.2: Model 1 Regression Results

Variable	$\hat{\boldsymbol{\beta}}$	**T-Statistic**	**Standard Error**
PS GamesPerSet	0.0008601818	0.048785668	0.017631854
PS BestOf	0.0038955141	0.115364905	0.033766890
PS Player Rank	-0.0328108885	-1.297297501	0.025291723
PS 1st Serve %	0.0397011077	0.248920581	0.159493070
PS Total Games Won	0.0031332959	0.361743060	0.008661661
PS Total Sets Won	0.0241245186	0.300535613	0.080271747
PS Match Won	-0.0057780851	-0.032002278	0.180552307
PS 2nd Serve Win %	0.1448315951	0.707888077	0.204596743
PS Tournament Level	0.0404251380	2.884943094	0.014012456
PS Elo Rating	0.0361227163	0.333992750	0.108154193
PS Break Point Save %	-0.0028517821	-0.016844176	0.169303740
PS 2nd Serve Return %	0.1316077029	0.656717864	0.200402197
PS Total Points Won %	0.1176586817	0.562149788	0.209301301
PS 1st Serve Win %	0.0976900138	0.647181159	0.150946937
PS Service Games Won %	0.0744886148	0.536098241	0.138945829
PS Aces	0.0052599981	0.499632968	0.010527724
PS Double Faults	0.0040422626	0.255944152	0.015793534
PS 1st Serve Return %	0.2103627062	0.673000456	0.312574389
PS Break Point Conversion %	-0.1683020729	-0.811356155	0.207433039
PS Return Games Won %	0.2053492751	0.636798407	0.322471402
PS Winners	0.0065043938	1.434960967	0.004532802
PS Unforced Errors	-0.0029576110	-0.653777707	0.004523879
PS Service Points Won %	0.1096440752	0.643746974	0.170321694
PS Return Points Won %	0.1698042259	0.645249028	0.263160762
PS Delta Set	0.2004948521	1.360913014	0.147323782

Continued on next page

Table 3.2 – continued from previous page

Variable	$\hat{\beta}$	T-Statistic	Standard Error
PS Delta Game	0.1022104003	2.347488642	0.043540317
PS Gender Female	-0.1955252658	-1.012845549	0.193045491
PS Gender Male	0.0581407118	0.508700738	0.114292564
PS Surface Carpet	3.7651481181	0.989196236	3.806270161
PS Surface Clay	-0.0498788786	-0.281367922	0.177272798
PS Surface Grass	0.3377566032	0.583843964	0.578504915
PS Surface Hard	-0.0017928593	-0.012650219	0.141725549
OPS GamesPerSet	-0.0022262275	-0.123663266	0.018002334
OPS BestOf	-0.0068574361	-0.199167332	0.034430526
OPS Player Rank	0.0553874670	2.249075504	0.024626771
OPS 1st Serve %	-0.0414790781	-0.254772174	0.162808510
OPS Total Games Won	-0.0040896978	-0.463682834	0.008820033
OPS Total Sets Won	-0.0299787588	-0.367305832	0.081617976
OPS Match Won	-0.0238825783	-0.130105178	0.183563626
OPS 2nd Serve Win %	-0.1500558333	-0.718609501	0.208814152
OPS Tournament Level	-0.0395605175	-2.787256429	0.014193354
OPS Elo Rating	0.0004157821	0.005471814	0.075986157
OPS Break Point Save %	0.0185465394	0.107459151	0.172591531
OPS 2nd Serve Return %	-0.1094100553	-0.534595832	0.204659387
OPS Total Points Won %	-0.1086399178	-0.508442060	0.213672169
OPS 1st Serve Win %	-0.0889502132	-0.577624205	0.153993223
OPS Service Games Won %	-0.0742277602	-0.523826147	0.141703045
OPS Aces	-0.0074452072	-0.695619454	0.010702989
OPS Double Faults	-0.0037941390	-0.234756210	0.016162039
OPS 1st Serve Return %	-0.1969273041	-0.616602626	0.319374741

Continued on next page

Table 3.2 – continued from previous page

Variable	$\hat{\beta}$	T-Statistic	Standard Error
OPS Break Point Conversion %	0.1609324624	0.760348635	0.211656147
OPS Return Games Won %	-0.1517801889	-0.460310811	0.329734139
OPS Winners	-0.0047083434	-1.028991436	0.004575688
OPS Unforced Errors	0.0026604459	0.579214903	0.004593193
OPS Service Points Won %	-0.1051175144	-0.604956112	0.173760563
OPS Return Points Won %	-0.1524606937	-0.567041568	0.268870401
OPS Delta Set	-0.2067340260	-1.388130993	0.148929767
OPS Delta Game	-0.1076487869	-2.445553018	0.044018178
OPS Gender Female	0.1826940822	0.924401255	0.197635043
OPS Gender Male	-0.0648269911	-0.556210942	0.116551089
OPS Surface Carpet	-2.6164447193	-0.671558082	3.896081052
OPS Surface Clay	0.0077221370	0.042553396	0.181469345
OPS Surface Grass	-0.3990896879	-0.682397919	0.584834268
OPS Surface Hard	0.0017426974	0.012099790	0.144027083
PA GamesPerSet	0.0009835349	0.053485892	0.018388680
PA BestOf	0.0054593907	0.154233942	0.035396817
PA Player Rank	-0.0594243496	-2.116297827	0.028079389
PA 1st Serve %	0.0746809158	0.423968774	0.176147208
PA Total Games Won	0.0021862987	0.229858002	0.009511519
PA Total Sets Won	0.0230132456	0.243320446	0.094579991
PA Match Won	0.1927947474	0.879462689	0.219218791
PA 2nd Serve Win %	0.0480992569	0.217804724	0.220836610
PA Tournament Level	0.0509104984	3.350309945	0.015195758
PA Elo Rating	0.2605274018	2.982772465	0.087344041
PA Break Point Save %	0.1880715222	1.028148118	0.182922595

Continued on next page

Table 3.2 – continued from previous page

Variable	$\hat{\beta}$	T-Statistic	Standard Error
PA 2nd Serve Return %	0.0402302153	0.187877049	0.214130548
PA Total Points Won %	0.0402119989	0.179417649	0.224125102
PA 1st Serve Win %	0.0358498445	0.221808485	0.161625217
PA Service Games Won %	0.0410724597	0.272172916	0.150905756
PA Aces	-0.0070121228	-0.516042090	0.013588277
PA Double Faults	-0.0302850110	-1.150578058	0.026321561
PA 1st Serve Return %	-0.0823247253	-0.246804942	0.333561899
PA Break Point Conversion %	0.1326984884	0.588125244	0.225629642
PA Return Games Won %	-0.0253272385	-0.069503063	0.364404640
PA Winners	0.0032025829	0.741520919	0.004318938
PA Unforced Errors	-0.0012032168	-0.254671004	0.004724593
PA Service Points Won %	0.0385837378	0.211934598	0.182054927
PA Return Points Won %	-0.0009563398	-0.003372403	0.283578131
PA Delta Set	0.0372189286	0.226642642	0.164218561
PA Delta Game	-0.0365911361	-0.759259877	0.048193164
PA Gender Female	-0.2053378252	-1.017671313	0.201772245
PA Gender Male	0.0619568591	0.519875116	0.119176427
PA Surface Carpet	-7.3642841971	-1.178400511	6.249389856
PA Surface Clay	-0.0502470096	-0.272740570	0.184230053
PA Surface Grass	-0.2439309019	-0.382788450	0.637247289
PA Surface Hard	0.0198863835	0.128806585	0.154389495
OPA GamesPerSet	-0.0038222647	-0.203590775	0.018774253
OPA BestOf	-0.0107260548	-0.297292682	0.036079108
OPA Player Rank	0.0710193802	2.451638883	0.028968124
OPA 1st Serve %	-0.0829994547	-0.461149847	0.179983698

Continued on next page

Table 3.2 – continued from previous page

Variable	$\hat{\beta}$	T-Statistic	Standard Error
OPA Total Games Won	-0.0040065478	-0.412939595	0.009702503
OPA Total Sets Won	-0.0385823028	-0.399165982	0.096657292
OPA Match Won	-0.2485077656	-1.108744727	0.224134338
OPA 2nd Serve Win %	-0.0770101231	-0.341449894	0.225538577
OPA Tournament Level	-0.0491830422	-3.197565075	0.015381405
OPA Elo Rating	-0.2394587125	-2.685051805	0.089182157
OPA Break Point Save %	-0.1933028620	-1.036450939	0.186504594
OPA 2nd Serve Return %	-0.0427641319	-0.195309358	0.218955878
OPA Total Points Won %	-0.0516496850	-0.225469112	0.229076545
OPA 1st Serve Win %	-0.0422017140	-0.255650584	0.165075758
OPA Service Games Won %	-0.0531349568	-0.344729686	0.154135135
OPA Aces	0.0038453375	0.277111454	0.013876502
OPA Double Faults	0.0321199770	1.195281521	0.026872311
OPA 1st Serve Return %	0.0451927362	0.132400310	0.341334068
OPA Break Point Conversion %	-0.1505315849	-0.653593314	0.230313838
OPA Return Games Won %	0.0361232436	0.096759571	0.373329928
OPA Winners	-0.0029261157	-0.670237953	0.004365786
OPA Unforced Errors	0.0008787020	0.183379951	0.004791702
OPA Service Points Won %	-0.0494847918	-0.266121199	0.185948328
OPA Return Points Won %	-0.0092569039	-0.031903746	0.290151004
OPA Delta Set	-0.0258590245	-0.155881634	0.165888846
OPA Delta Game	0.0442349721	0.907285581	0.048755291
OPA Gender Female	0.1751544973	0.846428391	0.206933627
OPA Gender Male	-0.0752224182	-0.618221553	0.121675503
OPA Surface Carpet	5.9623914866	0.925842360	6.439964018

Continued on next page

Table 3.2 – continued from previous page

Variable	$\hat{\boldsymbol{\beta}}$	T-Statistic	Standard Error
OPA Surface Clay	0.0476734415	0.253885193	0.187775588
OPA Surface Grass	0.2085253537	0.324275290	0.643050396
OPA Surface Hard	-0.0627696732	-0.398356033	0.157571790

By analyzing the columns corresponding to the $\hat{\boldsymbol{\beta}}$ vector values, t-statistics and standard errors, the influence of each variable on the delta set forecast can be quantified. Multiplying the number of features by four yields a lot of numbers to sift through, but there are interesting results to glean. Player rank, tournament level, winners, delta set and delta game appear to be the most influential variables, possessing t-statistic values greater than 1 or less than -1 paired with relatively low standard errors (calculated via dividing the regression coefficient by the t-statistic). Simply looking at the regression coefficients can be misleading, with the resulting $\hat{\beta}$ values for the carpet surface dummy variable serving as the perfect example. These coefficients are noticeably larger than the rest because the dataset only contains one tournament that was played on carpet. The standard error for the variable is extremely high, so that result is not considered.

It can be nebulous when interpreting the coefficients for PA and OPA-prefixed variables, as their effects on delta set forecasts can be difficult to definitively conclude. To provide an example of coefficient interpretation, the focus will narrow to the "winners" variable, as the regression results indicate that it is one of the most influential variables that was included and it is fairly simple to interpret. The sign of the regression coefficient is positive for both PS Winners and PA Winners, while the sign is negative for OPS Winners and OPA Winners. American Coco Gauff will serve as an example player to assist in providing a deeper explanation. Coco's forecasted performance should be better (higher forecasted delta set value) if she hits more winners (PS prefix) and worse if her opponent hits more winners (OPS).

This is consistent with the regression coefficient signs. One may also imagine that Coco's performance would be better if she allowed fewer winners (PA), but the coefficient for PA Winners is slightly positive. It would also make sense for Coco's performance to improve if her opponent allowed more winners (OPA), but the coefficient for OPA Winners is slightly negative. PA Winners and OPA Winners do possess t-statistics closer to zero, so they do not appear to be quite as influential as their PS and OPS counterparts.

It is not necessarily surprising for the interpretations of the PS and OPS-prefixed variable coefficients to be sharper, as these variables directly encompass player and opponent performance. PA and OPA also fulfill the same goal, but in a much noisier and more circuitous fashion. None of the raw tennis match statistics have any absolute relationship with the delta set outcome. As much as the winners variable may seem to align with ultimate match outcomes, a high number of winners may accompany a risky playing style and a high number of unforced errors. Similarly, a high first serve percentage may indicate that the player is not going for a lot on their first serve, reducing its effectiveness. As clear cut as some of these relationships may seem, the underlying connections are much more noisy. This fortifies the impressive nature of the model 1 accuracy.

Further analysis in regard to the inner-workings of the models allowed for the fluctuating nature of the Kalman gain to be visualized. The unique Kalman gain values for the one female and one male player who participated in the highest number of matches between Nov 12, 2022 and Feb 22, 2024 were isolated. Time series visualizations in figures 3.8 and 3.9 below show the manner in which the Kalman gain changed match-by-match for the two most active players in the specified time range, Iga Swiatek and Flavio Cobolli.

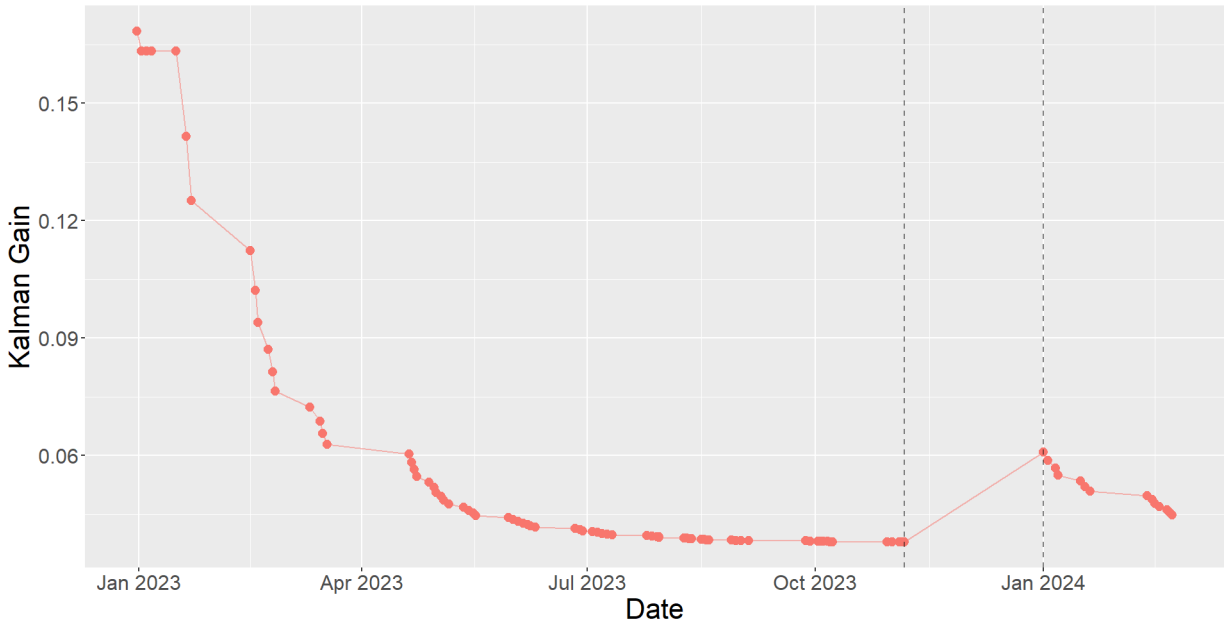

Figure 3.8: Kalman Gain Time Series: Iga Swiatek

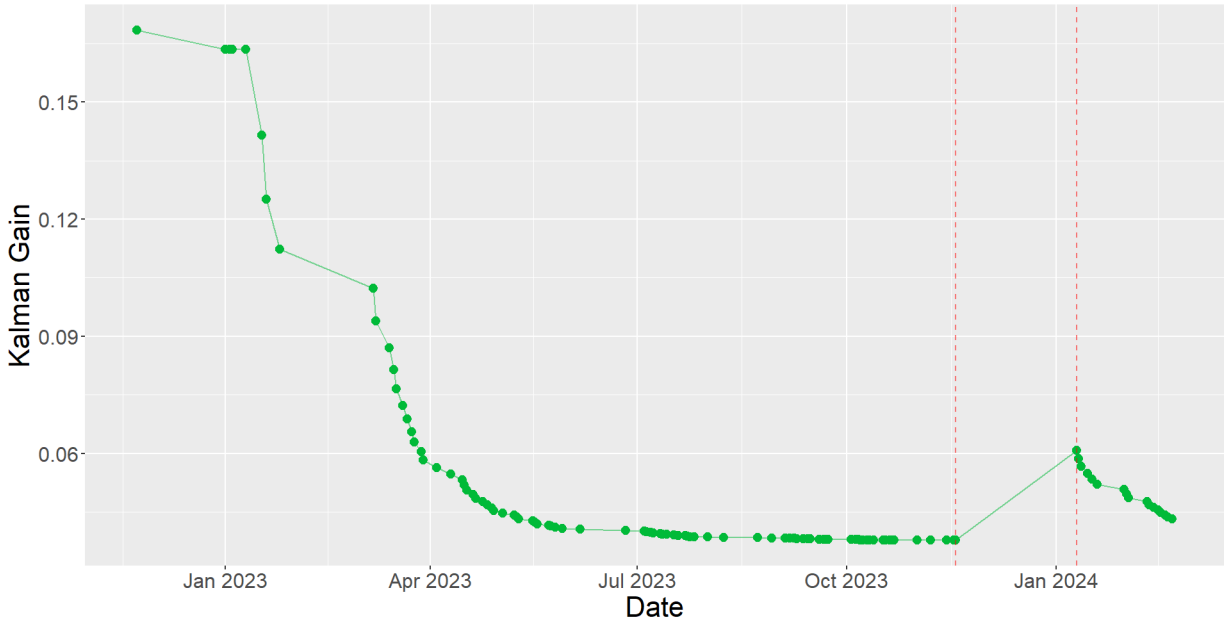

Figure 3.9: Kalman Gain Time Series: Flavio Cobolli

As was described in the methodology section, more data results in a smaller Kalman gain value. As Iga and Flavio played more and more matches, the dataset captured the resulting match statistics and in turn the filtered feature values encapsulated a higher vol-

ume of historical data. Every new match resulted in a decreased value for the Kalman gain; one new data point is much more influential when there are only two data points present rather than fifteen. It reached a point for both players, around April 2023 after roughly 20 or so matches, that new match statistics no longer affected the filtered values or the gain as much because the data was already plentiful. The decreasing rate of the Kalman gain, K_t from equations 2.10 and 2.11, slowed and the differences between the new data and the existing data were not as emphasized. The Kalman gain values for both players ultimately settled around 0.03-0.04, that is until the calendar turned over and a new tennis season began.

When the new year (new season)[2]of 2024 began, the Kalman gain values for both Iga and Flavio just about doubled. The degree to which the gains were affected by the new year was controlled by tuned signal-to-noise hyperparameters. Matches played in the new year were weighted a little greater due to recency, as different years were controlled by different hyperparameters. It is intuitive that forecasts for matches played in January and February 2024 should be affected more by results from other matches played in 2024 rather than results from the previous year. As Iga and Flavio continued to complete matches in 2024, the Kalman gain once again continued its slow decline.

3.2 Forecasting Interface

Model 1 was used to generate forecasts for any user-inputted matchup once the program was provided with two player names and the name of an event. These three pieces of information culminate in two forecasted probabilities of winning, one for each opposing player. As mentioned previously, the generated winning probabilities for two opposing players oftentimes do not exactly add up to 1. Each row of information (unit of observation being player match) will feature the four filtered values for each variable: player metric, opponent metric, metric allowed by player, and metric allowed by opponent. These values are used in the model, for

[2] "Year" and "season" are interchangeable throughout this paragraph. It was determined that the calendar turning over to a new year best exhibits the tennis season starting anew.

each row, not the information from the row of the opposing player. An interaction with the interface script is shown below in 3.10:

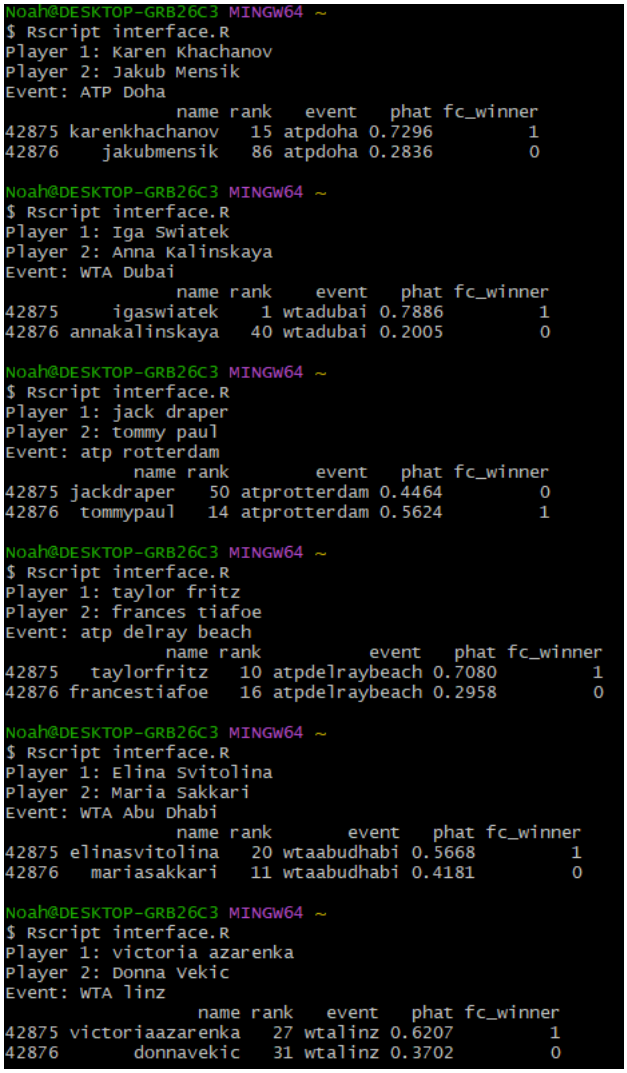

Figure 3.10: Match Forecasting Interface

For the first match that was run through the model shown above, the forecast for Karen Khachanov will incorporate Jakub Mensik's metrics for "opponent metrics," and vice-versa. Even with this data overlap the two forecasts are still made separately. In each match above the sums of the winning probabilities are very close to 1, which is testament to the power of the forecasting model. The Elina Svitolina *vs.* Maria Sakkari mock match at WTA Abu Dhabi features forecasted probabilities summing to about 0.98, which is most likely due to a small volume of metrics for one of the two players. Still, the model made a firm forecast

and chose the lower-ranked Svitolina as the forecasted winner.

CHAPTER 4

Conclusion

4.1 Overview

While the presence of player rank and elo rating resulted in a model featuring a slightly better match forecasting accuracy, the model strictly including past match statistics performed well above expectation. Since metrics such as player rank and elo rating exist as comprehensive benchmarks to quantify overall performance, the strong model results in the absence of such variables gives credence to both the historical match statistics as well as the method of Kalman filtering as effective model inclusions. The filtering process allowed for a "rolling average" of the statistical metrics for every professional player in the dataset. Time series visualizations for the Kalman gain revealed the behaviors of some model hyperparameters, which affected how the forecasts were generated and how the effect of time was quantified. The time-filtered information performed strikingly well in terms of forecasting accuracy even without the inclusion of the aforementioned rank and elo variables. Models 1 and 2 featured a slew of filtered match statistics while also including variables such as player rank and elo rating. The models' forecasting accuracies were 63.54% and 63.07%, respectively. Model 3 featured a lot of the same filtered match statistics but excluded rank and elo rating, and possessed a forecasting accurary of 62.48%. All three constructed models performed better than an algorithm that simply chose the player with the better ranking as the winner. This algorithm had a prediction accuracy of 61.72%, which is nearly 2% below the forecasting benchmark achieved by model 1.

Far more information in regard to individual players, tournament settings, situational variables and past occurrences could be implemented in the future to improve tennis match

forecasting accuracy. As more data is collected, forecasts generated by the current models should continue to improve. The limitations of this project will be described in detail along with ideas for future extensions that will be developed within this project in the future.

4.2 Limitations

A notable limitation of this project involved allocating time to deal with incorrect source data. After the scraping process was completed, it was uncovered on a few occassions that the information online was incorrect. This led to double and triple-checks to ensure that the scraped information was correct, but it was difficult to isolate the infrequent instances[1], especially as the number of scraped matches skyrocketed into the tens of thousands. These situations were only brought to light after visual inspections of the data and were dealt with by replacing the entire row of match statistics with NAs, resulting in the unit of observation not getting used in modeling. It was unfortunate to lose units of observation in this manner, but the instances were thankfully not common. An additional hurdle was alluded to previously and pertains to the HTML structure of the ATP website. This website was overhauled a couple of times throughout the duration of this project, making it necessary to restructure the scraping algorithm.

It is an imperfect process to instill arbitrary caps on some variables when determining whether or not they should be included in the data. For example, it is not uncommon for a player to retire during a match due to injury. The statistics associated with such a curtailed match are skewed in a sense, as the *a priori* variables may claim that the match was best of 3 sets, with 6 games per set. To try to combat this inconsistency, it was arbitrarily chosen that matches must go at least ten total games for the corresponding statistics to be used in the model. It is hard to know how best to deal with rare-but-harmful units of observation like this, but the safest method involves either instilling some sort of threshold or replacing

[1]Such inaccuracies included percentages well over 100%, the number of first serves attempted in a 3-set match being less than 10, etc.

the data with NAs altogether. This project required daily attention, whether it be to scrape match statistics, ensure that the data integrity was upheld, or manually update lookup tables. This does not yet exist as a self-sufficient, fully automated project, yet it can absolutely get to that point one day and require minimal oversight. As tedious as some steps of this project could be, the start-to-finish process is efficient. There still remain opportunities for further optimization and automation, especially once the extensions described below are added.

4.3 Future Work

With a model capable of forecasting the future with a fairly strong degree of accuracy, a logical next step would be to apply the model to upcoming tennis matches and assess how well it can perform on a large scale. The outputted probabilities could be manipulated to create a variable such as "forecast strength" which could be a ratio of some player's winning probability divided by that of their opponent. This value could possibly be useful if the model were ever leveraged to bet on upcoming matches. If the forecast strength exceeded some amount, it could be compared to the opposing players' rankings to uncover betting value and an automated process could alert the user of a high-value bet, or even execute the wager automatically. To potentially improve performance, the delta games variable could be implemented alongside delta sets as a second output. Forecasting delta games may be better suited given the match statistics that were scraped, as strong serving metrics paired with poor return metrics could lead to a flat delta games forecast. Big servers who struggle with returning serve to a degree (John Isner, Hubert Hurkacz, etc.) typically play a high number of tiebreakers, and a tiebreaker set results in a delta games outcome of just +1 or -1. Numerous additional variables could be scraped or created from existing variables, such as how active each player has been over a recent period of time[2], travel time getting to the tournament, distance from home countries, head-to-head records, career statistics and more. Regardless of the additional model inputs that may be chosen in the future, the current array of features is sufficient in creating a successful model. The ultimate performance will

[2]This would (imperfectly) quantify energy level/exhaustion, which is hard to do.

continue to improve as the model is fed with more and more match statistics, the production of which will never cease.

One final future inclusion to this project involves the ability to forecast entire tournaments once the draw matchups from professional tournaments are made public. The makeup of the draw could be scraped and a program could run each of the possible matchups through the forecasting model, generating probabilities of reaching each round of the tournament for each player. These improvements could be applied to the forecasting interface, which could be available for public use online in a more user-friendly environment.

REFERENCES

[ATPa] "ATP Tour." https://www.atptour.com/en/.

[ATPb] "Points and Prize Money." https://www.nittoatpfinals.com/en/event/points-and-prize-money. Accessed: 2024-01-23.

[Dan24] James H. Martin Dan Jurafsky. "Speech and Language Processing.", 2024.

[Had] Gui Hadlich. "How Do Tennis Players Qualify for Grand Slams?" https://mytennishq.com/how-do-tennis-players-qualify-for-grand-slams/.

[Hay96] Simon Haykin. *Adaptive Filter Theory*. Prentice Hall, 3rd edition, 1996.

[Isn] "John Isner Overview." https://www.atptour.com/en/players/john-isner/i186/overview.

[Ivo] "Ivo Karlovic Overview." https://www.atptour.com/en/players/ivo-karlovic/k336/player-stats.

[McC05] J. Huston McCulloch. "The Kalman Foundations of Adaptive Least Squares, With Application to U.S. Inflation." *Unpublished*, 2005.

[Rot] "Rotowire." https://www.rotowire.com/.

[Sac] Jeff Sackmann. "Tennis Abstract." https://www.tennisabstract.com/.

[WTA] "WTA Tour." https://www.wtatennis.com/.

[Zes09] Dave Zes. "Predicting Daily Smog by State Space Estimation.", 2009.