Problem Chosen	2024 MCM/ICM Summary Sheet	Team Control Number
C		2429173

The Hidden Factor of Momentum Within the 2023 Wimbledon Gentlemen's Singles

Tennis is a constantly growing game, consistently gaining attention globally. Although the game hinges on the simple premise of scoring points by hitting the ball into the opponent's court, there are many factors that impact how a game will play out. Understanding the inner workings of this sport can aid players, coaches, and even those looking to bet on a game.

The two most important factors of a match at any given moment are the current flow of the game and who has the momentum, if either player. In fact, "tennis more than any other sport, is a game of momentum" [3]. Thus, analyzing what specifically impacts momentum would aid players and coaches alike in strengthening their game play and success rates. In order do this, we must first produce a model for flow, which looks at the current score of the match and utilizes previous information in order to see who is "winning" and by how much. We aim to address first how flow and consequently momentum is formed and which variables are most significant in impacting swings. We produced several models to assess the fluctuations throughout a match and how it impacts each player's winning chances.

In order to develop an accurate model to depict the flow of a tennis match, we create **finite state machines (FSM)** in order to enumerate every possible score of the match. From the **FSM**s, we construct recursive relations to determine each player's current probability of winning the match from any given score. Using these equations, we are able to generate models that illustrate the flow of any match from the beginning to the current point being played.

We used a preliminary **lasso regression** to isolate the most relevant predictors of who will win the upcoming point, which we then use in our **multinomial logistic regression model (MLR)**. In addition to this, we analyzed the importance of certain variables through hypothesis tests and analyzed **variance inflation factor (VIF)** for issues with multicollinearity.

We selected several qualitative and quantitative features of play and implemented them to build a multinomial logistic regression model which looks at who scores the next point (point_victor). We completed a cross validation model with training and testing data to analyze the predictability on outside data.

After that, we looked at the **receiver operating characteristic curve (ROC)** curve to measure specificity. For error analysis we also used **root mean squared error (RMSE)**, **Akaike information criterion (AIC)** and **Bayesian information criterion (BIC)** to measure prediction accuracy. We extracted 9 indicators from the data set: lag(point_victor), server, lag(speed_mph), p1_sets, p2_sets, p1_games, p2_games, serve_no, lag(distance_run). With this, we predicted the outcome of points which led us to predictions of swings in momentum.

Finally, we evaluated and refined our model and reported the findings in a letter to tennis coaches. We hope this letter acts as an aid to help players and coaches alike learn the math behind this beloved game and better their play.

KEYWORDS: Finite State Machine, Lasso Regression, Multinomial Logistic Regression, Variance Inflation Factor, ROC, RMSE, AIC, BIC

Contents

1 Introduction

1.1 Background

The sport of tennis is one that has grown in popularity over the years. Specifically Wimbledon, the global tennis competition, is world renowned – getting over 54.3 million streams for the 2023 competition. The goal of this game seems simple: score more points than your opponent. However, it gets more complicated when looking at factors that impact your ability to achieve this such as whether you are serving or if you simply are having an off day regarding game play.

1.2 Restatement of the Problem

- Task 1: Develop a model and visualization that captures the flow of play as points occur and apply it to one or more of the matches.
- Task 2: Determine if momentum plays a role in the match or if swings in play are random through statistical modeling.
- Task 3: Develop a model that determines if there are other indicators to predict momentum. Then, generalize the model to "all matches."

1.3 Our Work

1. Developed an algorithm to compute the current probability of each player winning the match from a given score through the use of finite state machines and recursive equations.
2. Computed lasso regression as well as bootstrap tests to ensure importance of variables towards momentum.
3. Used multinomial logistic regression to predict momentum. For this model, we determined that each point scored adds to the swing of momentum and is impacted by the previous point.
4. Generalized MLR to make it applicable to different types of games (such as women's tennis rules or different levels of play).

2 Assumptions and Notations

2.1 Assumptions and Justification

In order to develop and construct our models, we make the following assumptions:

1. **The crowd has no effect on our data set.** Although the crowd can definitely affect the outcome of any point, we will assume the crowd has no effect on our data set for three reasons: it allows for a simpler model, our data set did not account for this variable, and players in our data set are elite professionals who are used to a crowd.
2. **The court is grass.** All data comes from matches held on grass courts. Thus, our model will assume a grass court.
3. **The game follows Wimbledon men's single rules.** Since we are only looking at the results from the Wimbledon men's competition, it is appropriate to assume that will be the data going into our model.

2.2 Notations

Symbols	Definitions
x_i	Player i's Points in Current Game
x_j	Player j's Points in Current Game
g_i	Player i's Games in Current Set
g_j	Player j's Games in Current Set
s_i	Player i's Sets in Current Match
s_j	Player j's Sets in Current Match
P_G	Probability of Winning the Current Game
P_{TB}	Probability of Winning the Current Tie-Breaker
P_{TB5}	Probability of Winning the Fifth-Set Tie-Breaker
P_S	Probability of Winning the Current Set
P_M	Probability of Winning the Current Match

Table 1: Notations

3 Flow Model and Methods

In order to create our Flow Model, we decided to break down every possible score in a best-of-five tennis match. The score of a match is composed of the overall set score, the game score in the current set, and the point score in the current game. Our Flow Model assumes that points, games, and sets are **independent** of one another. For more details on the overly-complicated intricacies of tennis scoring, we recommend reading this article [4]. In addition, our approach to this problem was greatly aided by Jacob Gollub's Bachelor's Thesis from Harvard College [2].

We will begin by first modeling points and games, then possible tie-breakers, then sets, and finally the whole match.

3.1 Modeling Points and Games

In a given (non-tie-breaker) tennis game, there are only 5 different scores per player: 0, 15, 30, 40, and AD. From these, there are 18 different possible score combinations in a game. However, the scores (40, 30) and (AD, 40) have identical score successors: if the server wins the point, they win, and if the returner wins the point, the new score is (40, 40). The same games for the scores (30, 40) and (40, AD). Therefore, our model will treat these pairs as equivalent scores. For simplicity, we will write the scores 0, 15, 30, and 40 as 0, 1, 2, and 3, respectively, in our diagram. Each state represents the current score of the game, SW means the server has won the game, RW means the returner has won the game, and transitions s and r mean the server has won the point or the returner has won the point, respectively. We would like to note our diagram's similarity to one in [2].

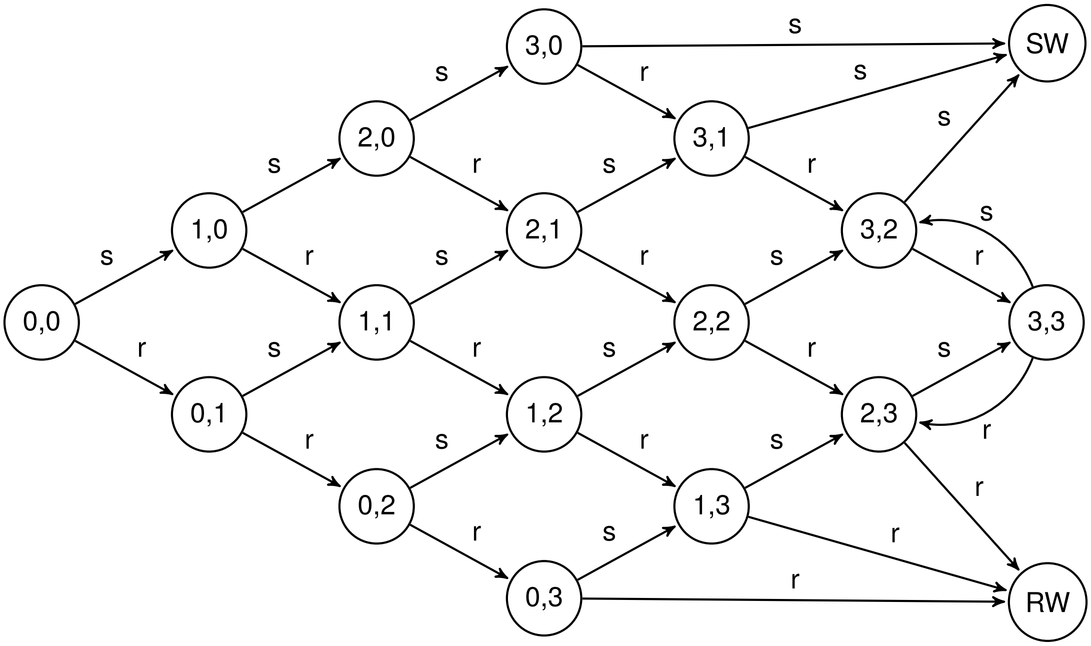


Figure 1: Finite State Diagram for Points in a Game

In order to quantify the probability of a player winning from a given score, we ran calculations on our data set and discovered a very important fact. Given a player serves, their probability of winning the point is 67.31%. Conversely, given a player returns, their probability of winning the point is 32.69%. There is a substantial advantage to serving rather than returning, which is to be expected in tennis (but was quite shocking to us non-tennis players). For simplicity, we will give the server a probability, p, of $\frac{2}{3}$ to win the point and the returner a probability, $1-p$, of $\frac{1}{3}$. Quantifying the probability for a given score would be tough if not for a key insight. From a score of (3,3), we can measure the probability of where the game will be in exactly 2 points. The server will win if they win the next two points, which has probability of p^2. The returner will win if they win the next two points, which has probability of $(1-p)^2$. Finally, the game will be back at a score of (3,3) if each player wins one point, which has a probability of $2(p)(1-p)$. Therefore, from the server's perspective of the game, their total probability of winning is:

$$p^2+p^2[2(p)(1-p)]^1+p^2[2(p)(1-p)]^2+\ldots=\sum_{k=0}^{\infty}p^2[2(p)(1-p)]^k=\frac{p^2}{1-[2(p)(1-p)]}=\frac{p^2}{1-2p+2p^2}.$$

Now that we can quantify the probability of the server winning the game from a score of (3,3), we can create a recursive model in order to calculate the probability of the server winning from any possible score.

$$P_G(x_i,x_j)=\begin{cases}1, & \text{if } x_i=4, x_j\leq 2\\ 0, & \text{if } x_j=4, x_i\leq 2\\ \frac{p^2}{1-2p+2p^2}, & \text{if } x_i=x_j=3\\ (p)(P_G(x_i+1,x_j))+(1-p)(P_G(x_i,x_j+1)), & \text{otherwise}\end{cases}$$

Figure 2: Recursive Model for Server Winning the Game

3.2 Modeling Tie-Break Games

Tie-break games occur when a set score of (6,6) is reached and are decided by the first to 7 points (must win by 2 points) except in the 5th set of a match when it is first to 10 points (must win by 2 points). The returner of the previous game serves the first point, and then each player alternates serving 2 points at a time. Because of the constant switching of the servers, calculating the base case for recursion was more difficult than in a regular game. However, we discovered that from a score of (7,7), the next four points will either end the tie-breaker or place the score at (9,9), which is equivalent to (7,7). Thus, we calculated the probability of winning of the player serving at (7,7) to be

$$(p)(1-p)+(p)(p)(1-p)(p)+(1-p)(1-p)(1-p)(p)=\frac{28}{81}$$

when $p=\frac{2}{3}$. The probability is the same for the returning player at (7,7), so the probability of returning to (7,7) is $\frac{25}{81}$. By computing the geometric series, $\sum_{k=0}^{\infty}\frac{28}{81}(\frac{25}{81})^k$, we discovered that both players have a $\frac{1}{2}$ chance to win the tie-breaker from a score of (7,7). The same holds for the fifth set tie-breaker too.

The recursion becomes slightly more complicated because we have to account for servers switching. Ultimately, the recursive models which track the probability of player i winning are as follows:

$$P_{TB}(x_i,x_j)=\begin{cases}1, & \text{if } x_i\geq 7, x_i-x_j\geq 2\\ 0, & \text{if } x_j\geq 7, x_j-x_i\geq 2\\ \frac{1}{2}, & \text{if } x_i\geq 7, x_i=x_j\\ (p)(P_{TB}(x_i+1,x_j))+(1-p)(P_{TB}(x_i,x_j+1)), & \text{if } (x_i+x_j) \mod 2=1\\ (p)(1-P_{TB}(x_j,x_i+1))+(1-p)(1-P_{TB}(x_j+1,x_i)), & \text{if } (x_i+x_j) \mod 2=0\end{cases} \tag{3.1}$$

Figure 3: Recursive Model for Player i Winning a Non-Fifth-Set Tie-Breaker

$$P_{TB5}(x_i,x_j)=\begin{cases}1, & \text{if } x_i\geq 10, x_i-x_j\geq 2\\ 0, & \text{if } x_j\geq 10, x_j-x_i\geq 2\\ \frac{1}{2}, & \text{if } x_i\geq 10, x_i=x_j\\ (p)(P_{TB5}(x_i+1,x_j))+(1-p)(P_{TB5}(x_i,x_j+1)), & \text{if } (x_i+x_j) \mod 2=1\\ (p)(1-P_{TB5}(x_j,x_i+1))+(1-p)(1-P_{TB5}(x_j+1,x_i)), & \text{if } (x_i+x_j) \mod 2=0\end{cases}$$

Figure 4: Recursive Model for Player i Winning the Final Tie-Breaker

3.3 Modeling Sets

In a given set, there are 7 different scores per player: 0, 1, 2, 3, 4, 5, 6. From these, there are 39 different possible score combinations in a set. In order to win a set, one player must reach 6 games and be up by 2 games against their opponent. If (6,6) is reached, they play the tiebreaker as described before. Each state represents the current score of the set, BW means player i has won the set, CW means player j has won the set, transitions b and c mean player i has won the game or player j has won the game, respectively, and transitions b' and c' mean player i has won the tie-breaker or the player j has won the tie-breaker, respectively.

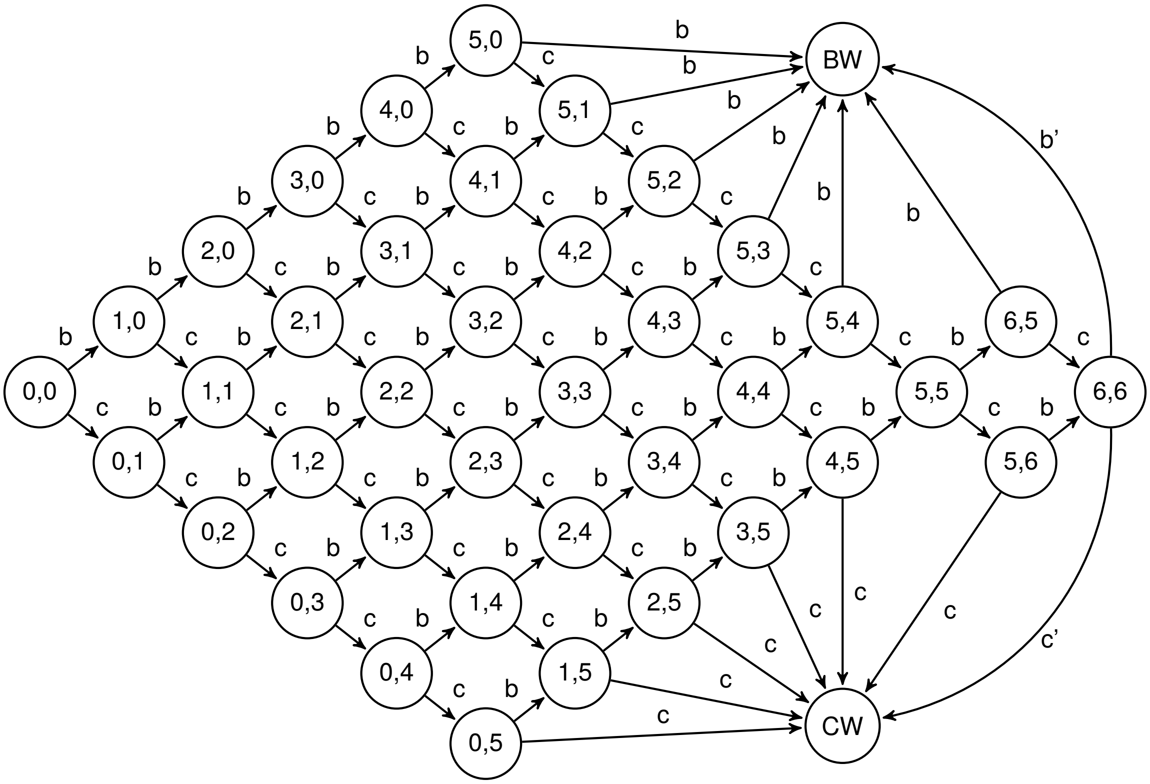


Figure 5: Finite State Diagram for Games in a Set

For the same reason as in the tie-breakers, the recursion becomes trickier as we have to account for switching in servers after a game has finished, which is why we changed the variables above from s and r to avoid confusion.

$$P_S(g_i, g_j) = \begin{cases} 1, & \text{if } g_i \geq 6, g_i - g_j \geq 2 \\ 0, & \text{if } g_j \geq 6, g_j - g_i \geq 2 \\ P_{TB}(s_i, s_j), & \text{if } g_i = g_j = 6 \\ P_G(0,0)(1 - P_S(g_j, g_i + 1)) + (1 - P_G(0,0))(1 - P_S(g_j + 1, g_i)), & \text{otherwise} \end{cases}$$

Figure 6: Recursive Model for Player i Winning the Set

3.4 Modeling a Match

In a given match, there are 3 different set scores per player: 0, 1, and 2. From these, there are 9 different possible score combinations in a match. In order to win a match, one player must reach 3 sets. Each state represents the current set score of the match, BW means player i has won the match, CW means player j has won the match, transitions b and c mean player i has won the match or player j has won the match, respectively.

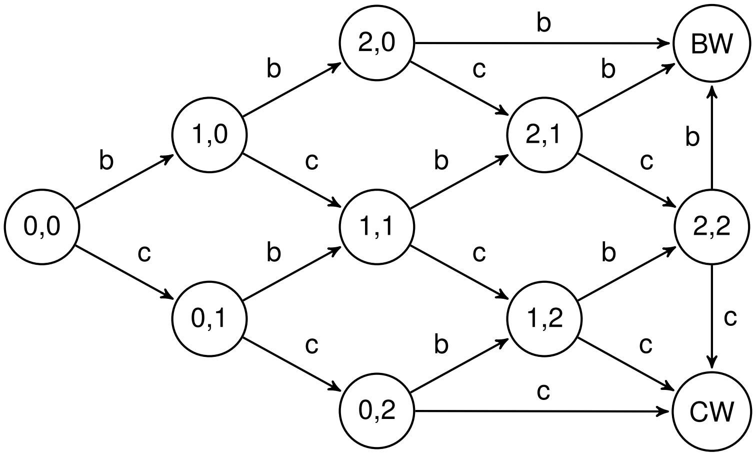


Figure 7: Finite State Diagram for Sets in a Game

The recursive model below evaluates the probability of winning the match from the perspective of the player with s_i sets won.

$$P_M(s_i, s_j) = \begin{cases} 1, & \text{if } s_i \geq 3 \\ 0, & \text{if } s_j \geq 3 \\ P_S(0,0)(P_M(s_i+1, s_j)) + (1 - P_S(0,0))(P_M(s_i, s_j+1)), & \text{otherwise} \end{cases}$$

Figure 8: Recursive Model for Player i Winning the Match

3.5 Final Flow Equation

Now, we can recursively calculate the probability of the serving player i defeating the returning player j from any given score of $(s_i, s_j, g_i, g_j, x_i, x_j)$ where (s_k, g_k, x_k) is player k's amount of sets won, games in the current set won, and points in the current game won. The recursive relation is as follows:

$$P_M(s_i, s_j, g_i, g_j, x_i, x_j) = \begin{cases} 1, & \text{if } s_i \geq 3 \\ 0, & \text{if } s_j \geq 3 \\ P_{TB5}(x_i, x_j), & \text{if } g_i = g_j = 6, s_i = s_j = 2 \\ P_{TB}(x_i, x_j), & \text{if } g_i = g_j = 6, s_i \neq 2 \text{ or } s_j \neq 2 \\ 1 - P_M(s_j, s_i + 1, 0, 0, 0, 0), & \text{if } g_i \geq 6, g_i - g_j \geq 2 \\ 1 - P_M(s_j + 1, s_i, 0, 0, 0, 0), & \text{if } g_j \geq 6, g_j - g_i \geq 2 \\ 1 - P_M(s_j, s_i, g_j, g_i + 1, 0, 0), & \text{if } x_i = 4, x_j \leq 2 \\ 1 - P_M(s_j, s_i, g_j + 1, g_i, 0, 0), & \text{if } x_j = 4, x_i \leq 2 \\ (\frac{p^2}{1-2p+2p^2})(1 - P_M(s_j, s_i, g_j, g_i + 1, 0, 0)), & \text{if } x_i = x_j = 3 \\ p(P_M(s_i, s_j, g_i, g_j, x_i + 1, x_j)) + & \\ (1-p)(P_M(s_i, s_j, g_i, g_j, x_i, x_j + 1)), & \text{otherwise} \end{cases}$$

Figure 9: Complete Recursive Model for Player i Winning the Match

4 Flow Model Results and Evaluation

4.1 Results

When testing our Flow Model on a match, we will do it on arguably one of the "most remarkable battles" in tennis history: the 2023 Wimbledon Gentlemen's Final. Our model breaks the flow into each set with the use of vertical lines – thus we will analyze set by set. We will compare how the game was described in the MCM problem to the flow graph that we produced, and then check with the actual game footage and points.

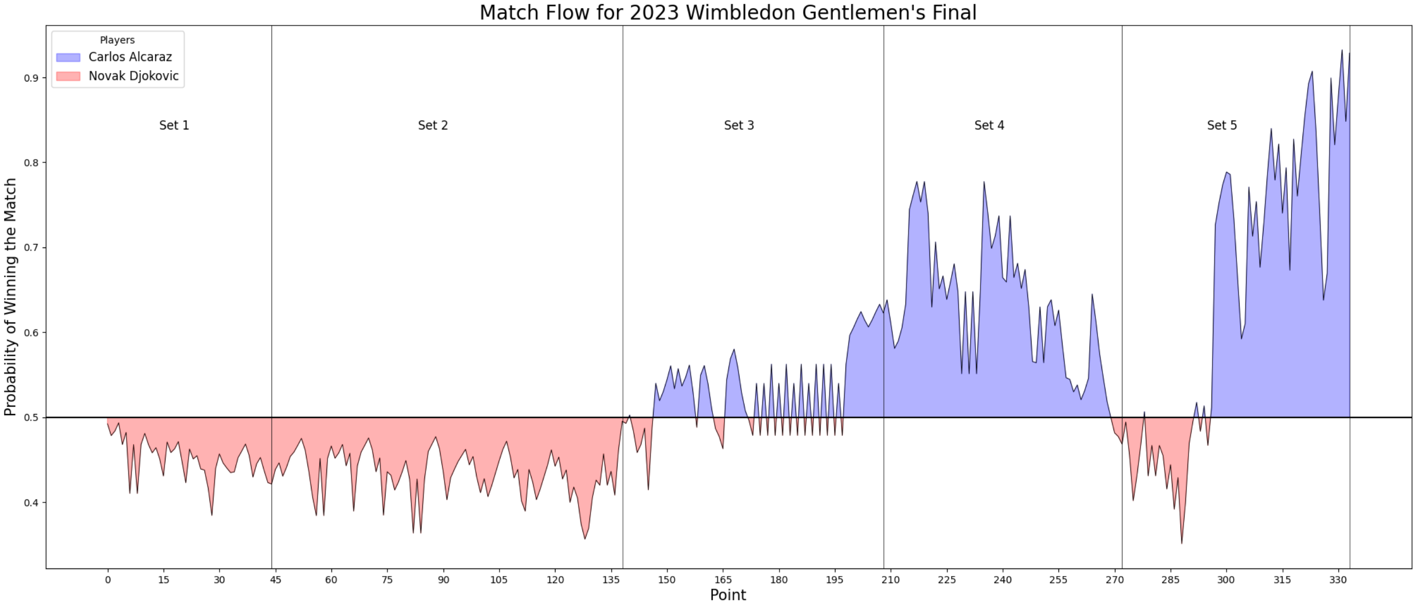


Figure 10: Example of Flow Model

Set 1: "Djokovic dominated the first set 6 – 1 (winning 6 of 7 games)"

- The flow graph clearly goes, seemingly uncontested, into Djokovic's side.

Set 2: "Tense and finally won by Alcaraz in a tie-breaker 7 – 6"

- The graph shows the continuation of the data on Djokovic's side, which is logical as he is up one set. However, we see fluctuations in the graph more than during Set 1. At the conclusion of the set, the line is at 0.5, indicating they are tied in sets.

Set 3: "Alcaraz won handily 6 – 1"

- We can see the jump to Alcaraz's side, indicating he is ahead. However, there is one concern when looking at the obvious fluctuations of the graph from point 168 to point 199. It seemed like there was a lot of back and forth, so we looked into the data. What was occurring was a very intense game jumping from Deuce to AD over and over. The game lasted over 28 minutes and encapsulated 32 points, 7 of which were breakpoints! [5]
- Despite this back and forth, the end of Set 3 shows the flow towards Alcaraz's side, indicating he won the set.

Set 4: "Alcaraz seemed in control at first, but Djokovic took complete control to win the set 6 – 3"

- The graph begins by increasing in favor of Alcaraz. However, noticeable dips in the graph indicate the loss of certain games. The graph then begins to dip even more and ends around the 0.5 line, only slightly in favor of Djokovic.

Set 5: "Djokovic started with the edge, but Alcaraz gained control and the victory 6 – 4"

- These words once again translate directly to what is visualized in the graph. We can see the dip into Djokovic's favor as the final set begins. However, as Djokovic loses control, the set becomes even at 0.5 and finally swings into the favor of Alcaraz. The end of the model grows dramatically as the odds of winning are all but secured that close to the end of the game.

4.2 Sensitivity Analysis

As demonstrated above our model picks up on very small factors (such as a 32 point back and forth game between players). The sensitivity on our model is quite high. However there are still many improvements we would like to make. In regards to robustness, our model has a very specific level of talent and category. This model is made for elite level men's players. With more time, we would like to expand our model and available data by beginning with the women's Wimbledon tournament, and then branching to other levels of tennis too.

Our Flow Model would definitely have to be tweaked for women given the difference of the sets and a possible difference in probability of the server winning the point, but by using the same steps, we could reproduce a model for women's tennis. Another interesting test for our model would be to look at doubles tennis.

When discussing error analysis possibilities, we could begin by testing edge cases such as a game with an abnormal amount of lead changes or a game with an extremely unlikely comeback. If we had more time, we would have liked to do integration regression testing, which would validate the flow of the data and would note if any additional variables would disrupt our model. This would greatly help us to better understand our model because if we are able to simply add to the model in order to accommodate women's play, that would be something that we should utilize and expand upon in the future.

4.3 Model Assessment

Strengths

1. The finite state machines offer a compact representation of complex behaviors and allow us to fully enumerate every possible score in a tennis match.
2. Our recursive models provide definitive equations that can provide the current probability of each player winning the match (assuming independence of points).
3. The Flow Model allows for simple visual readability, allowing both statisticians and coaches alike to understand and gain information from our model.

Weaknesses

1. Since this model assumes independence of points, the probability of each player winning the match is calculated solely on score and current server. It does not consider or predict flow based on possible hot-streaks in play or player rating before the match begins.
2. Furthermore, when running a hypothesis test later in our analyses, we concluded that one point does have a significant impact on the next point.
3. We assumed that the server has a $\frac{2}{3}$ chance of winning the point regardless of server number. However, later in our analysis we concluded this probability decreases by over 10% given they fault on their first serve (and are on serve number two).

5 Momentum Model and Methods

$$\log \frac{P}{1-P} = \beta_0 + \beta_1(x_{i1}) + \beta_2(x_{i2}) + \ldots + \beta_k(x_{ik})$$

Figure 11: Multinomial Logistic Regression Formula

We used a multinomial logistic regression to calculate momentum. This structure allows us to consider multiple features of declared importance through the explanatory data analysis (EDA). Through the EDA, we concluded that the server, as well as what serve number it is, has a critical impact on who will win the point. Another thing we noted was that a person is more likely to hit an ace given they have a faster serve, so we also wanted to look at speed_mph when computing the regression.

We decided to test multiple formulas and then compute the AIC and BIC as well as AUC to see which formula performs the best. We compared three models to see which categories are most indicative of who will score the next point (and therefore in whose direction the momentum will shift).

5.1 Variables

We began our variable test by conducting a bootstrap hypothesis test to see if one point impacts the outcome of the next.

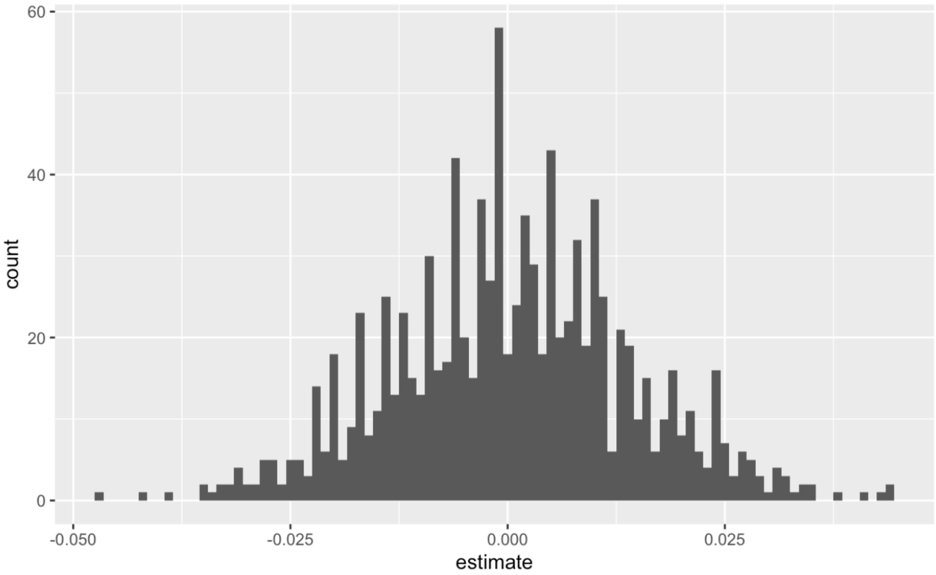
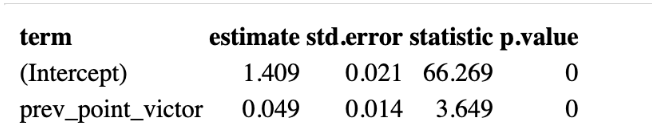


Figure 12: Bootstrap distribution



term	estimate	std.error	statistic	p.value
(Intercept)	1.409	0.021	66.269	0
prev_point_victor	0.049	0.014	3.649	0

Figure 13: Results and P-Values

Through conducting a bootstrap hypothesis test we noted that one point has a large impact on the next point. The p-value for prev_point_victor (the player who won the previous point) indicates that it is statistically significant at the 0.05 level. This suggests that winning the previous point has a substantial impact on winning the current point in the tennis match. This contradicts the coach's theory of randomness as a player is more likely to win the next point given they have won the previous point. From here, we conducted lasso regression to aid in variable selection.

$$\min_{\beta_0,\beta} \frac{1}{N} \sum_{i=1}^{N} w_i l(y_i, \beta_0 + \beta^T x_i) + \lambda \left[(1-\alpha)\|\beta\|_2^2/2 + \alpha\|\beta\|_1\right]$$

Figure 14: Lasso Regression Formula

Lasso regression is a method that helps with variable selection and enhances the prediction accuracy of the statistical model.

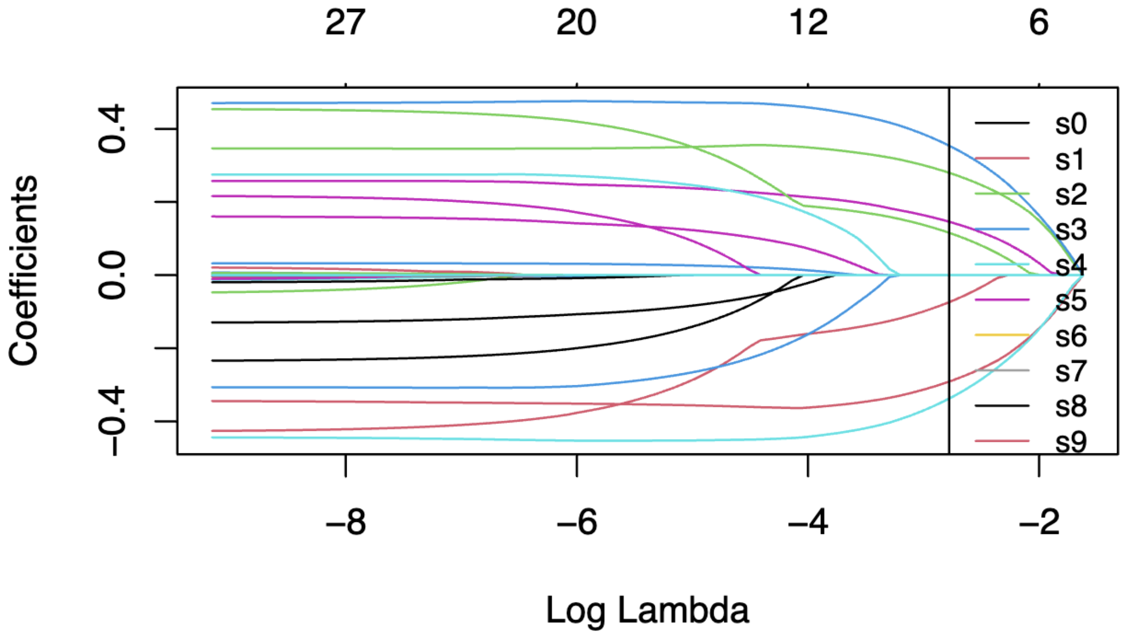


Figure 15: Lasso Model

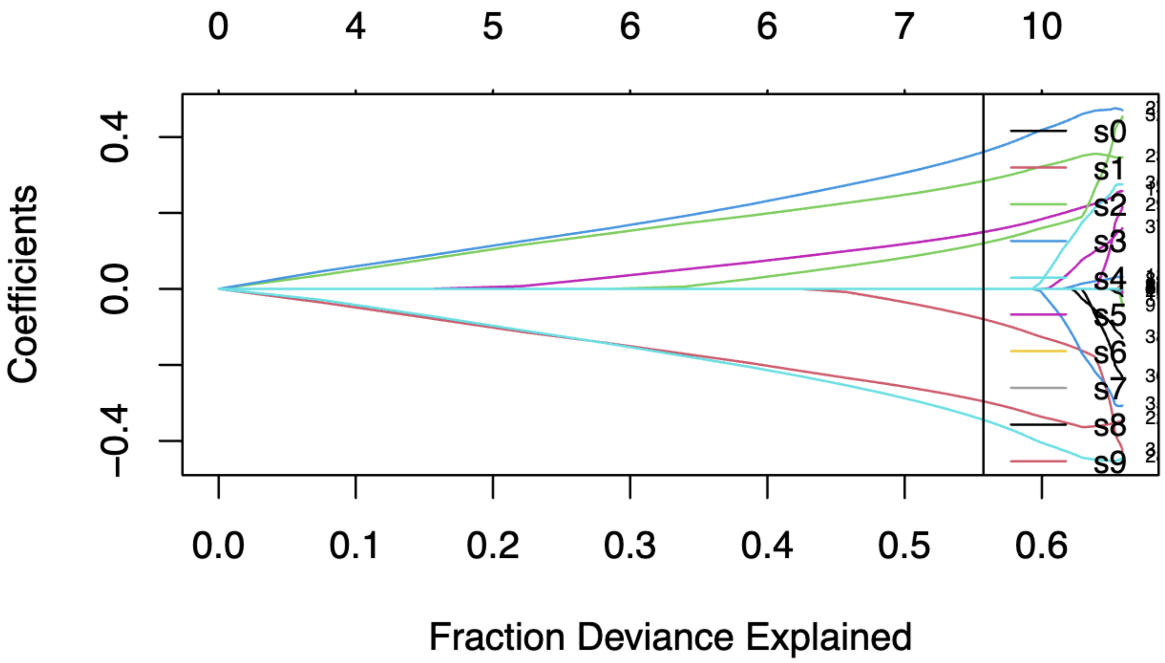


Figure 16: Lasso Deviance Model

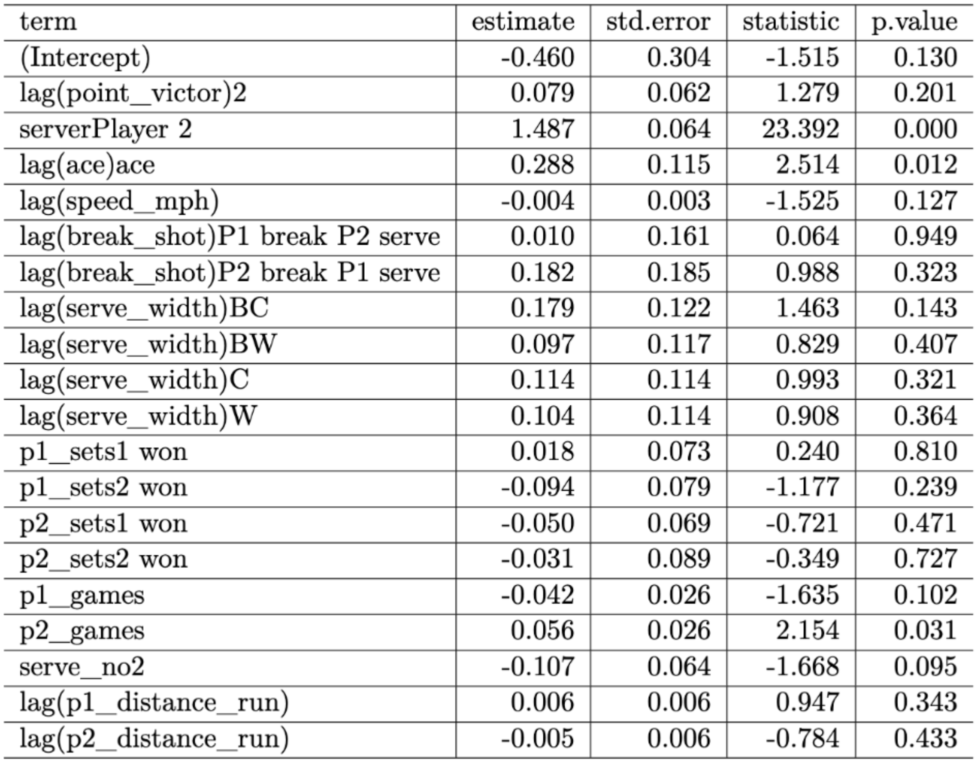
Through this regression as well as hypothesis tests conducted, we selected several variables of interest and put them into the first model. Our first model only looked at server, lag(ace), and lag(speed).

term	estimate	std.error	statistic	p.value
(Intercept)	-0.431	0.274	-1.572	0.116
serverPlayer 2	1.454	0.061	23.846	0.000
lag(ace)ace	0.277	0.107	2.584	0.010
lag(speed_mph)	-0.003	0.002	-1.262	0.207

Figure 17: MLR 1

It is important to note the use of lag as it looks at the previous value. When predicting the current point victor, we cannot know anything about the play so we look at the events during the previous rally for data. We continued to add all of the variables that were not aliased into the model.

5.2 Model Development and Analysis

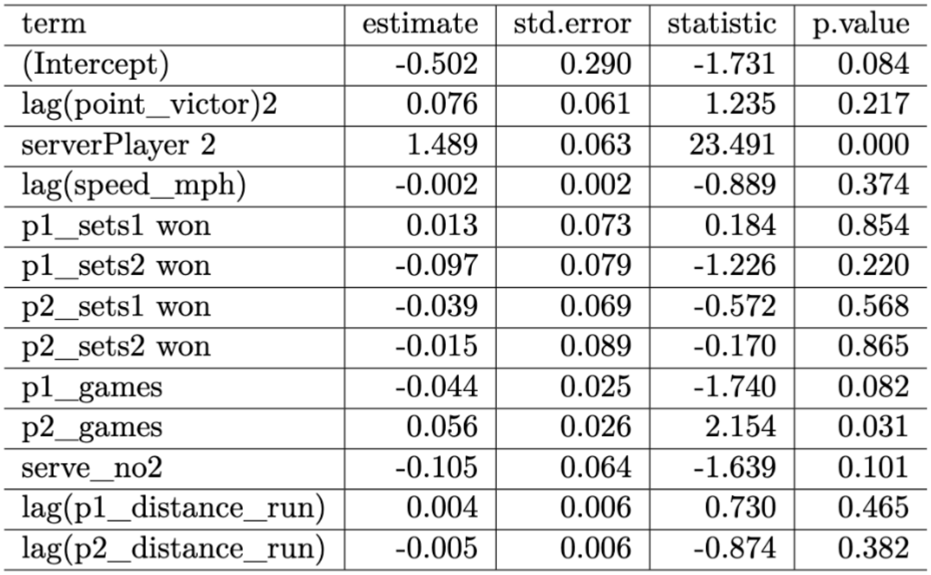


term	estimate	std.error	statistic	p.value
(Intercept)	-0.460	0.304	-1.515	0.130
lag(point_victor)2	0.079	0.062	1.279	0.201
serverPlayer 2	1.487	0.064	23.392	0.000
lag(ace)ace	0.288	0.115	2.514	0.012
lag(speed_mph)	-0.004	0.003	-1.525	0.127
lag(break_shot)P1 break P2 serve	0.010	0.161	0.064	0.949
lag(break_shot)P2 break P1 serve	0.182	0.185	0.988	0.323
lag(serve_width)BC	0.179	0.122	1.463	0.143
lag(serve_width)BW	0.097	0.117	0.829	0.407
lag(serve_width)C	0.114	0.114	0.993	0.321
lag(serve_width)W	0.104	0.114	0.908	0.364
p1_sets1 won	0.018	0.073	0.240	0.810
p1_sets2 won	-0.094	0.079	-1.177	0.239
p2_sets1 won	-0.050	0.069	-0.721	0.471
p2_sets2 won	-0.031	0.089	-0.349	0.727
p1_games	-0.042	0.026	-1.635	0.102
p2_games	0.056	0.026	2.154	0.031
serve_no2	-0.107	0.064	-1.668	0.095
lag(p1_distance_run)	0.006	0.006	0.947	0.343
lag(p2_distance_run)	-0.005	0.006	-0.784	0.433

Figure 18: MLR 2

We computed the ROC, AUC, AIC, and BIC of this model, which we note has an AUC of 0.688. We then began to remove variables with high p-values one by one – tracking their impact on AUC. We had expected that lag(distance) should be removed because of the high p-value and a VIF of approximately 6. However, removing it decreased the AUC by several percent, so we will leave it in the final model but keep note of a possible multicollinearity issue. We were left with these variables as ones that influence the point_victor and therefore the momentum of the game. Our final model passed the test for linearity, randomness, and independence so we can continue with the model.

5.3 Results and Interpretations



term	estimate	std.error	statistic	p.value
(Intercept)	-0.502	0.290	-1.731	0.084
lag(point_victor)2	0.076	0.061	1.235	0.217
serverPlayer 2	1.489	0.063	23.491	0.000
lag(speed_mph)	-0.002	0.002	-0.889	0.374
p1_sets1 won	0.013	0.073	0.184	0.854
p1_sets2 won	-0.097	0.079	-1.226	0.220
p2_sets1 won	-0.039	0.069	-0.572	0.568
p2_sets2 won	-0.015	0.089	-0.170	0.865
p1_games	-0.044	0.025	-1.740	0.082
p2_games	0.056	0.026	2.154	0.031
serve_no2	-0.105	0.064	-1.639	0.101
lag(p1_distance_run)	0.004	0.006	0.730	0.465
lag(p2_distance_run)	-0.005	0.006	-0.874	0.382

Figure 19: MLR 3

This leads us to our final model:

$$
\begin{aligned}
log\left(\frac{P}{1-P}\right) = {} & -0.502 + 0.076 * lag(point\_victor)2 + 1.489 * (serverPlayer2) \\
& - 0.002 * lag(speed\_mph) + 0.012 * p1\_sets1won - 0.097 * p1\_sets2won \\
& - 0.039 * p2\_sets1won - 0.015 * p2\_sets2won - 0.044 * p1\_games + 0.056 * p2\_games \\
& - 0.105 * serve\_no2 + 0.004 * lag(p1\_distance\_run) - 0.005 * lag(p2\_distance\_run)
\end{aligned}
$$

Quantitative Predictors:

- **lag(speed_mph)**: For a one-unit increase in the previous speed in miles per hour, the log-odds of player 2 winning the point decrease by 0.002.
- **lag(p1_distance_run)**: For a one-unit increase in the previous distance run by player 1, the log-odds of player 2 winning the point increase by 0.004.
- **lag(p2_distance_run)**: For a one-unit increase in the previous distance run by player 2, the log-odds of player 2 winning the point decrease by 0.005.
- **p1_games**: For every one-unit increase in the number of games won by player 1, the log-odds of player 2 winning the point decrease by 0.044.

- **p2_games**: For every one-unit increase in the number of games won by player 2, the log-odds of player 2 winning the point increase by 0.046.

Categorical Predictors:

- **lag(point_victor)2**: Given that the previous point victor is player 2, the log-odds of player 2 winning the point increase by 0.076.
- **serverPlayer 2**: If the server is player 2 (compared to player 1, the reference category), the log-odds of player 2 winning the point increase by 1.489.
- **serve_no2**: If the serve number is 2 (compared to 1, the reference category), the log-odds of player 2 winning the point decrease by 0.105.
- **p1_sets1 won**: If player 1 has won 1 set, the log-odds of player 2 winning the point increase by 0.012
- **p1_sets2 won**: If player 1 has won 2 sets, the log-odds of player 2 winning the point decrease by 0.097.
- **p2_sets1 won**: If player 2 has won 1 set, the log-odds of player 2 winning the point decrease by 0.039.
- **p2_sets2 won**: If player 2 has won 2 sets, the log-odds of player 2 winning the point decrease by 0.015.

6 Momentum Model Results and Evaluation

6.1 Sensitivity Analysis

When analyzing sensitivity we applied an ROC curve to the model in order to look at both sensitivity, the ability of the model to correctly identify true positives, and specificity, the ability of the model to correctly identify true negatives.

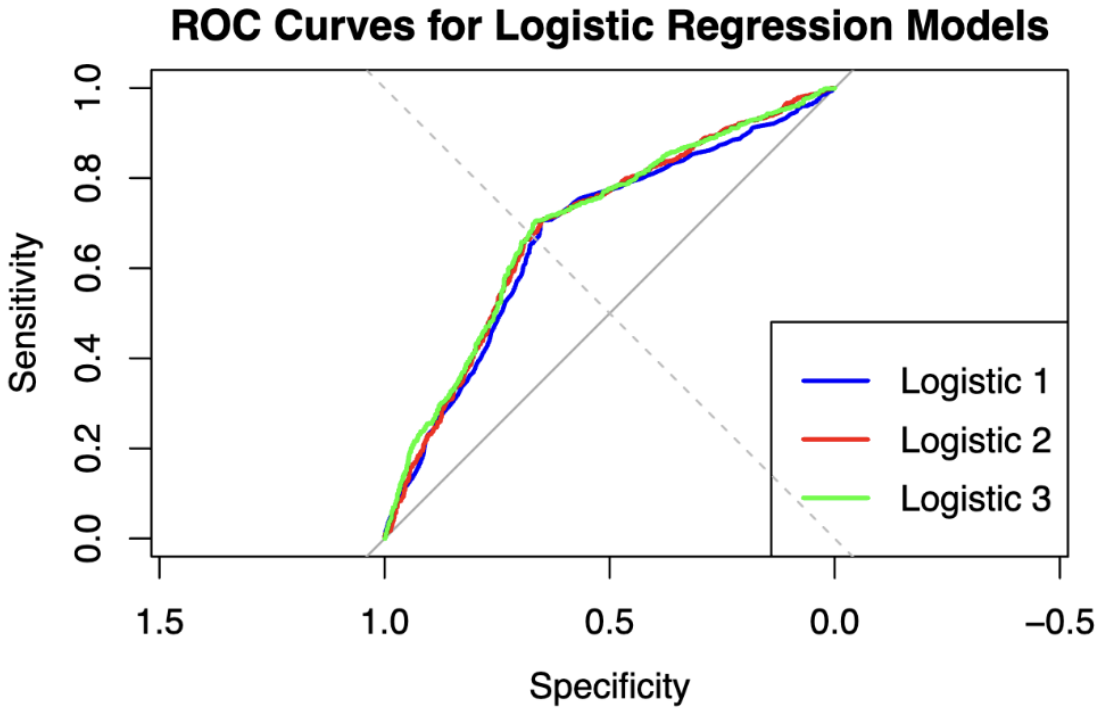
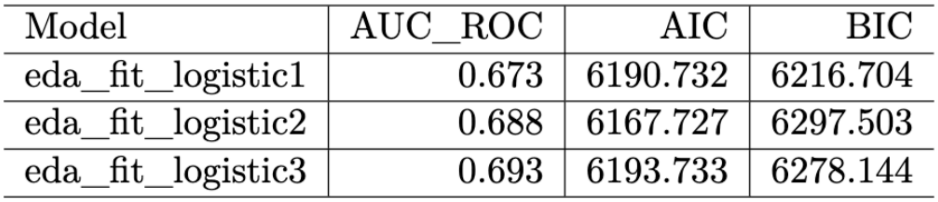


Figure 20: ROC Graph



Model	AUC_ROC	AIC	BIC
eda_fit_logistic1	0.673	6190.732	6216.704
eda_fit_logistic2	0.688	6167.727	6297.503
eda_fit_logistic3	0.693	6193.733	6278.144

Figure 21: Accuracy Data

With an area under the curve (AUC) of 0.693 for the logistic model predicting point_victor, it means that the model correctly ranks a randomly chosen instance where "point_victor" is 1 (positive instance of P2 scoring) higher than a randomly chosen instance where "point_victor" is 0 (negative instance of P1 scoring) approximately 69.3% of the time. The AUC reflects the model's ability to distinguish between the two classes based on their predicted probabilities.

6.2 Model Assessment

Strengths

1. The use of lasso regression is a great strength as it allows us to hone in on specific variables so we have a starting point for the MLR.

2. MLR allows us to put in several variables and get a binary outcome regarding which level of a variable is the most important (i.e. if p1_sets has two wins, it is considered more significant than if there is only one win).

Weaknesses

1. Some of the variables in the final model have a high VIF, so there may be a multicollinearity issue.

2. We did not account for some points, such as break points, having more importance than others. The lack of weight towards variables could have a negative impact on our model.

7 Discussion and Future Work

In regards to our models in general, a closer look at more variables would allow us to better ascertain which factors are the most important. In particular, studying variables such as court type, previous wins, player rankings, and crowd support could aid in making a more accurate model.

In addition to this, analyzing different levels of play would be beneficial. Our data set of only men's Wimbledon matches limits our analysis to that of male pro tennis players. Using data from a larger range of players would possibly allow us to note how different levels of players are impacted by different factors.

Another important addition to our work would be the exploration of entropy. In information theory, entropy is something that quantifies the amount of information conveyed by a given event. In the context of a tennis game, we can consider an event to be a single point and the information conveyed includes but is not limited to a specific server, a specific way of scoring a point, and a specific rally amount with a result of who scored. We can gain information about who is in the "lead" from this, but it is important to look at how much information we can gain from each observation. In the future, we would like to look at positional entropy, which explains that order and specific points matter more than others. This was backed up in several papers we researched in which, aside from game points, (15, 30) is determined to have the highest range of swing [1]. However, we did not get the opportunity to analyze this ourselves, so we note our curiosity.

8 Conclusion

In our efforts to track and quantify the flow and momentum of tennis matches, we created two models.

The Flow Model came about through our need to quantify the advantage or disadvantage each player has at any given position of the match. We developed finite state machines to meticulously enumerate and represent possible point match-ups, game match-ups, tie-breaker match-ups, and set match-ups. We found states in the machines that can lead to infinite loops and calculated geometric series in order to assess these probabilities. From these cases, we were able to create recursive relations that can calculate a player's probability of winning a game, a set, and the match from any possible score. We converted the recursive equations into an algorithm to run on our data set and tested it on numerous matches, notably the final, in order to assess its accuracy. The Flow Model is quite accurate and creates a graph that depicts the flow of the match in a readable fashion. Important weaknesses of this model are that it treats every point as an independent event where the server has probability, p, of winning the point and the returner has probability, $1 - p$, of winning the point. Consequently, every game and every set is independent of one another too. However, our intuition told us that this was not the case, so we developed the Momentum Model in order to account for this.

The MLR model allowed us to take the Flow Model and make it predictive. We are able to determine who will score the next point given all of the previous information. We began by determining what factors we suspected would be most indicative of point scored through lasso regression and hypothesis tests. From there, we created several models with the variables selected through the previous analyses. With these three models produced, we tested their error/specificity through ROC curves, AIC, and BIC. From this, we are able to conclude that there are many factors that influence who will win the next point, and we are able to predict this with 69.3% accuracy. As described above, variables such as who is serving, how many previous sets and games were won and by who, and the previous point scored all determine who will win the next point. We can utilize this to determine shifts in the momentum as each point leads to a shift in the game and ultimately the match.

It is clear from our data and research that swings in play and runs of success of a player are not random. Instead, there

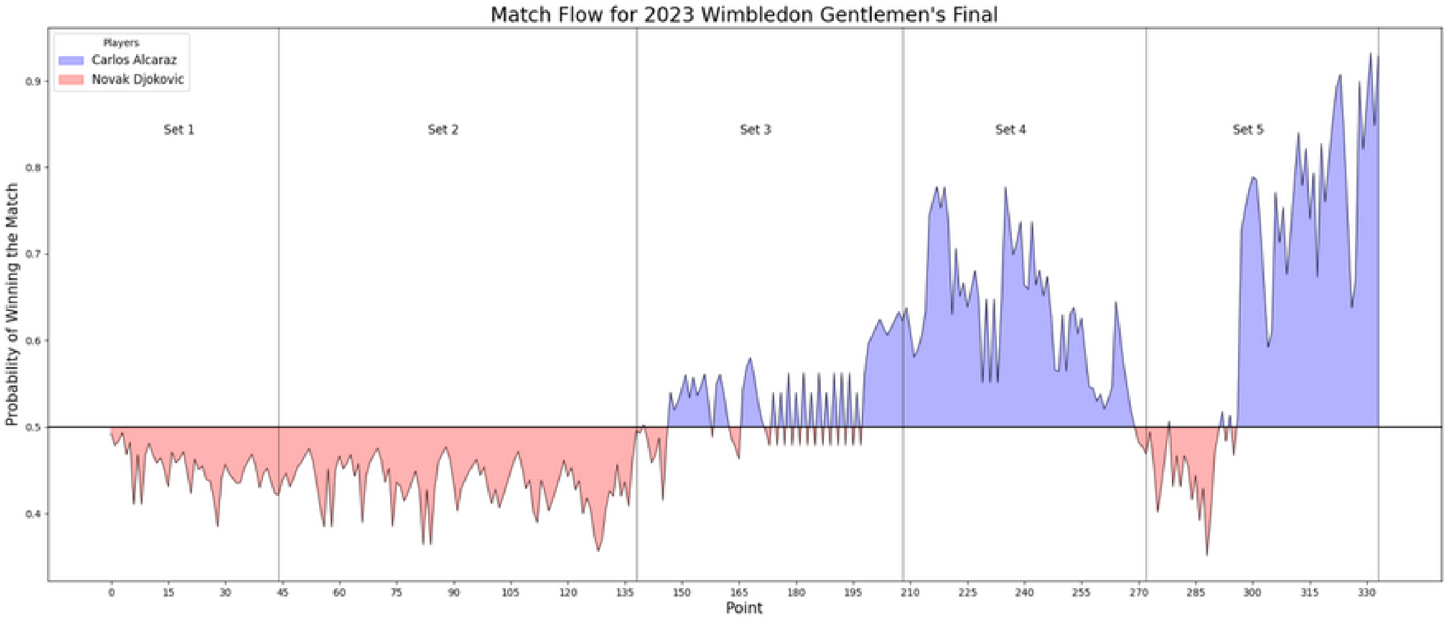
Letter of Advice

Dear Coaches,

As fellow tennis enthusiasts (albeit mediocre players), we wanted to understand the immense variation in game play. There are often instances where the position of a match can show a clear winner, but then somehow the position changes in an instant. It was unclear to us whether the swings in play and runs of success were completely random, or if there was an underlying, possibly quantifiable, component of the game that we were not seeing. In order to assess this question, we first created a Flow Model to track which player is performing better and to what extent, and then created a predictive model to determine who will score the next point.

Creating the Flow Model required us to enumerate the over 5,000 possible tennis scores that a match can find itself in. Then, we were able to calculate a given player's chance of winning from that score depending on which player was serving.

As an example, we applied our model to a particular game as illustrated here.



As you can see, this graph depicts the 2023 Wimbledon Gentlemen’s Final, in which 20-year-old Spanish rising star Carlos Alcaraz defeated 36-year-old Novak Djokovic in a nail-biter of a performance. Our model quite accurately tracks the flow of the game, and it is clear which points and games were the most important throughout the match.

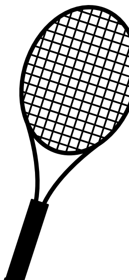
Furthermore, the logistic regression model that we have developed can provide who is more likely to win the next point and even the current game based on the current factors and previous factors of the last rally. From this model, we have identified factors that contribute highly to momentum and would like to share advice based on what we have learned.

To start, one thing to tell your players is to not purely rely on the benefit of serving, but to capitalize on not missing the first serve. It is common knowledge in tennis that the server has an advantage, but our data shows that given a player faults on their first serve, their percentage of winning the point drops over 10%. In addition, a player is significantly more likely to hit an ace given they serve at a higher speed. This is something to work on with your players as our model shows that speed (mph) increases the odds of scoring a point.

Although we did not go into too much detail regarding our mathematical and statistical proofs, we hope that our insight and suggestions will prove beneficial for you and your players' future successes.

All the best,

MCM Team #2429173



References

[1] A. K. Dixit, *Illustration of rollback in a decision problem, and dynamic games of competition*, (2004).

[2] J. Gollub, *Producing win probabilities for professional tennis matches from any score*, (2017).

[3] C. Kriese, *Total Tennis Training*, Masters Press, 1988.

[4] J. Rivera, *Tennis scoring, explained: A guide to understanding the rules terms & point system at wimbledon*, The Sporting News, (2023).

[5] Wimbledon, *A FINAL FOR THE AGES — Carlos Alcaraz vs Novak Djokovic Full Match — Wimbledon 2023*, 2023. YouTube video.

A Appendix

All code is viewable here: `https://github.com/yubrajbhandari923/MCMContest`