Final Project
FIT3139
Daniel Nguyen
32471033

Table of Contents

Specification Table

Base Model	Tennis game simulated using Markov Chains
Extension Assumptions	In the base model, each state in the markov chain represents a potential score in the game. To create the transition probabilities in the markov chain of the base model, I will model each individual point using a markov chain, and use monte carlo simulation to calculate the approximate probabilities of the transitions. This provides a higher level of detail and differentiation between different players. To simulate each point, the markov chain starts from a serve node, and finishes on a “winner” node. The transition matrix probabilities are based on player serve, return and rally statistics. I can then simulate a full game/set/match using the probabilities calculated from my markov chain.
Techniques Showcased	Markov Chains Monte carlo simulation
Question 1	How well does my model predict match and tournament outcomes?
Question 2	What is the most important part of the game in my model?

Introduction

I would like to go more in depth into simulating a tennis match using markov chains, and analyse the accuracy of the model when simulating real tennis matches and tournaments. To do this, I will take the base model, where we simulate a game of tennis using a markov chain where each state represents the score of the game, and calculate the transition probabilities of the chain by simulating individual points in the match.

In combination with my point markov chain, I will use monte carlo simulation to take advantage of the law of large numbers to provide accurate measurement of outcome probabilities.

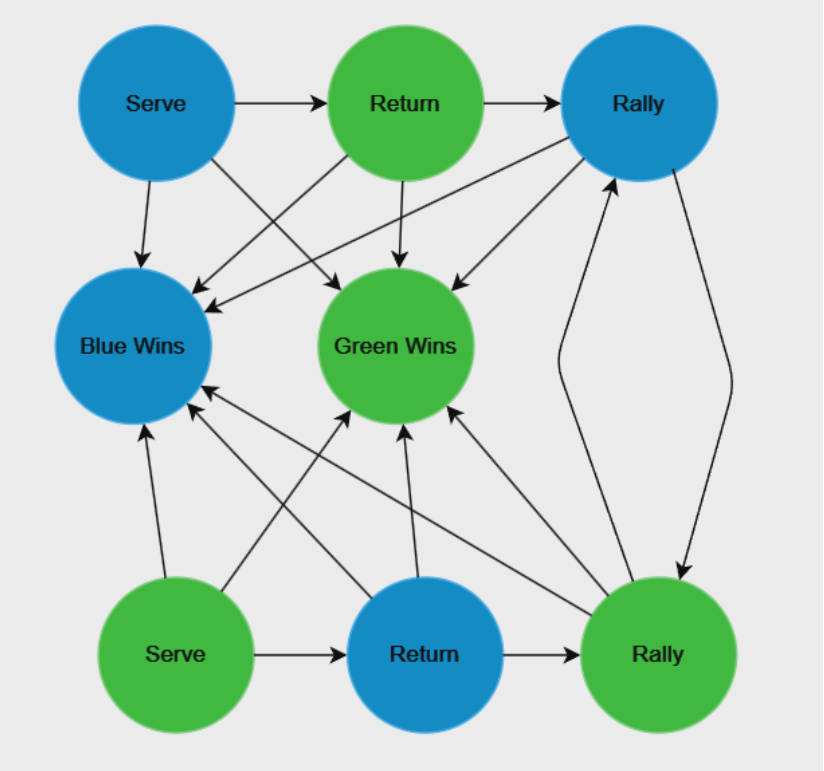
I would like to explore the accuracy of the model in predicting real tennis matches, as it ties my work to a real world scenario..

I would also like to look at how different tennis skills impact simulation in my model. I am using serve, return and rally statistics as the basis of my model creation, and I would like to see if it overstates or understates the importance of these 3 parts of a game of tennis. For example, do strong servers have a disproportionate chance of winning according to my model, or are consistent rally shotmakers not winning enough?

Model Description

Model Explanation

My markov chain models a tennis point, starting from a serve state, and ending on 1 of 2 absorbing states, representing a point win for either player. Following the serve state are states the serve return, and the following rally. A visualisation of the chain could look like:



Where green nodes represent the green player is in control of the ball, and blue nodes represent that the blue player is in control of the ball. This allows me to model individual players and their different strengths and weaknesses when playing tennis.

The states labelled “Blue Wins” and “Green Wins” are examples of absorbing states, where, once reached, they cannot be left. Logically, this makes sense, as once a point has been won by either player, it cannot be changed.

In terms of the base model, where each state represents a potential score of the game, instead, each state can instead be represented by an instance of the above markov chain. The resulting markov chain is extremely complex, so to increase efficiency, my point model will be used to create an outcome distribution that is then used to define the transition probabilities of the base model. Over a large number of iterations, these two models are equivalent.

Model Assumptions and Disadvantages

For my model, I am assuming that each player plays at a constant level of skill, and things such as current player form and player cardio do not impact their play. I was not able to find statistics regarding players first serve fault rate, so I have combined the first and second serve into a single state, and used the double fault rate as the transition probability from serve to the receiver win state. Additionally, some player stylistic differences are not captured by the model. For example, a player that has a high preference for a “serve and volley” play, or a player who excels at “grinding out” a long match (a famous example of this is Novak Djokovic) would not be totally accurately represented by the different probabilities in the transition matrix, which models their overall abilities. I have also not differentiated between playing surface in my model. This means that players who are a “specialist” on one surface will have an understated chance of winning on their preferred surface, while they would have an overstated chance of winning on a different surface.

As I am using existing player statistics, player performance will be represented across all matches in which their statistics were recorded. Thus, players will be represented as the average of all of their performances where the statistics were recorded, which means that players who have played a top level in the past, but have now fallen off (the big 4 for example), or players who have recently had a rise or fall in their playing ability may not be accurately represented as the level in which they are playing now.

Additionally, some players have more matches included in their statistics than others. Players who are popular with the fans, or who have played for a long time will naturally have many more matches included in their statistics. Players who are popular may have more victories transcribed, while players who are disliked by many tennis fans may have more of their defeats transcribed. I have no way to prove or disprove this, but it is something to keep in mind.

Initially, I attempted to model as accurately as possible the differences between players, specifically by modelling shot depth, shot placement, net points, forehand/backhand preference and rally length in the markov chain structure, however, there is not not enough data regarding shot direction and depth that is differentiated by shot selection (a deep shot would be recorded, but not how it was hit or where it was hit from). Additionally, the resulting model is much too complex. The model I have created does show a meaningful difference between players however, as differences in the serve and return are shown in the model, as well as differences in consistency during the rally.

When simulating thousands of tournaments, hundreds of thousands (or even millions) of points must be simulated. This results in painfully slow progress when running a monte carlo simulation. I managed to remedy some of these issues by re-using already generated tennis game markov chain transition probabilities, however with each extra round that is included in the tournament, the possible player matchups scales very quickly. Thus I have chosen to simulate tournaments from the quarter finals onwards.

Model Creation

For my data collection, I made use of the Match Charting Project (https://www.tennisabstract.com/charting/) in which tennis fans catalogue tennis matches and collect player statistics. It contains data on player serves, returns and shot preferences, among many other things. I used player data from this resource to calculate the transition probabilities for my point markov chain.

Using data about 2 different players, I can create a tennis point markov chain, and run a monte carlo simulation to find the probability that the server wins the point, and use this extracted probability in the tennis game markov chain. I can then use this markov chain to simulate tennis games, sets, matches and partial tournaments

.

Algorithmic and Mathematical Methods Used

In the creation of my model, I have used markov chains and monte carlo simulation.

Markov chains are a stochastic model that models future evolution based on a current state, in which there is a probability of moving from the current state the next. I have used this to model a tennis point, where at each stage of the point there is a probability of moving to one of multiple following states. For example, when a player is serving, there is a chance of going from the serving state to the return state, the receiver wins state (in the case of a double fault), or the server wins state (in the case of an ace or a forced error).

Monte carlo simulation is the use of random sampling of input to produce a range of outputs many times, and then modelling a probabilistic distribution based on these outputs. I have used this to randomly walk the markov chain based on its transition probabilities, to calculate the probability of a player winning a point/game/set/match of tennis.

When comparing the results of my monte carlo simulation and the results of real tennis matches, it is impossible to see what the “real” winning chances of a player are. Thus, when looking at singular matches, I use the ELO system to calculate winning chances to compare my model to. I have collected the ELO ratings of players from (https://tennisabstract.com/reports/atp_elo_ratings.html), and used the following formula to calculate winning probability:

$$\frac{1}{1 + 10^{elo\ difference\ /\ 400}}$$

For entire tournaments, it is more difficult to determine winning probabilities, so I have opted to use betting website odds, and compare the rankings from most likely to least likely winners with the predictions of my model.

Model Classification

The use of monte carlo simulation indicates a numerical model, as we use randomness to simulate a large number of points, and obtain numerical results.

As the probability of moving from one state to another does not change over time, the model is linear in nature. However, if I were to add complexity to the model by changing the rally transition probabilities (to model a players ability in shorter/longer rallies), the model may have been non-linear. I have not done this however, to allow for easier modelling.

My model takes a set of discrete states and transitions between these states, and simulates them a set number of times to provide a distribution of a binary outcome. This means that the model is discrete. With larger numbers of iterations, we can approach continuity in our results, as we approximate a probability density function, however, by nature it is still discrete, as there are a finite number of results.

Markov chains and monte carlo simulation are by definition both stochastic techniques, and this does not change for my model, which uses a combination of both. This is because markov chains depend on probabilistic state changes, and monte carlo simulation uses random sampling to provide an estimation of outcome probability.

Results

#disclaimer#
When you run the code in my .py file, the generated graphs may look slightly different as the monte carlo simulation technique uses randomness to generate a path through the markov chain. I have tried to set the number of iterations as high as reasonably possible given code runtime constraints, so the difference between graphs in this report and your generated graphs using my code should be minimal. Thanks :)

French Open Prediction Results

Lets look at a match that happened recently, and test our model against the results. I am using the French Open (also known as Roland Garros) finals, where Carlos Alcaraz defeated Alexander Zverev in 5 sets. After simulating the match 5000 times, according to my model, we arrive at the following win distribution:

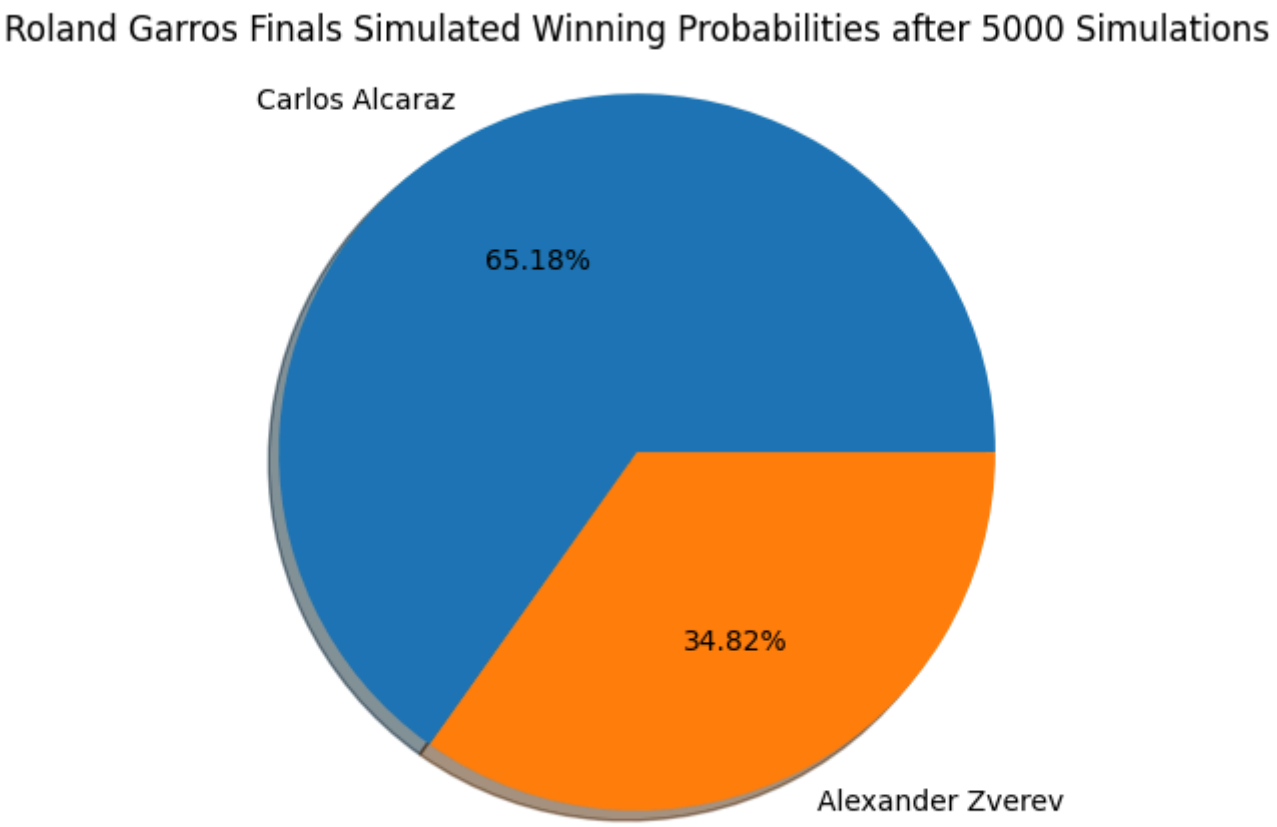
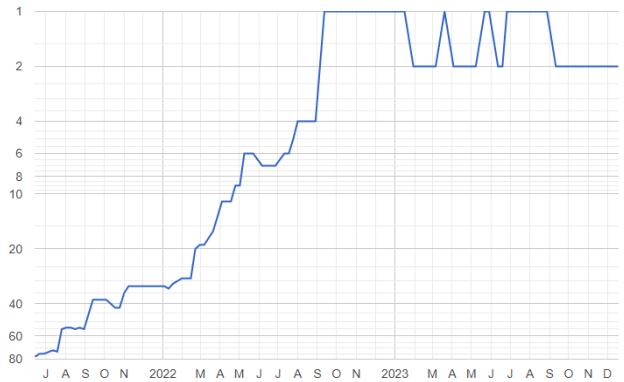


Fig 1.
According to the model, Carlos Alcaraz has around a 65% chance of winning the match. The model predicted the outcome of the match successfully, however, the head to head score is actually 5 - 5 since 2021. Looking deeper into this rivalry, we can see the following ATP rankings for each player:



Carlos Alcaraz Ranking over Time

Alexander Zverev Ranking over Time

Charts taken from https://www.ultimatetennisstatistics.com

In 2021 and 2022, the head to head was 3 wins to Zverev and 1 to Alcaraz, while in 2023 and 2024, the head to head was 1 win to Zverev and 4 to Alcaraz. This seems to suggest, along with the ranking over time, that Carlos Alcaraz has significantly improved his game over the past 2 years, and the head to head statistics may not be indicative of current winning probabilities.

Furthermore, the current ELO rating of each player as of 6/10/2024 are as shown below:

	Alexander Zverev	Carlos Alcaraz
ELO rating	2074.5	2197.4

Applying our formula as detailed in the previous section, Alexander Zverev has a 33.01% chance of winning the match. Comparing this to the outcome of our model, it is actually quite accurate!

We can also see the match score distribution of our model:

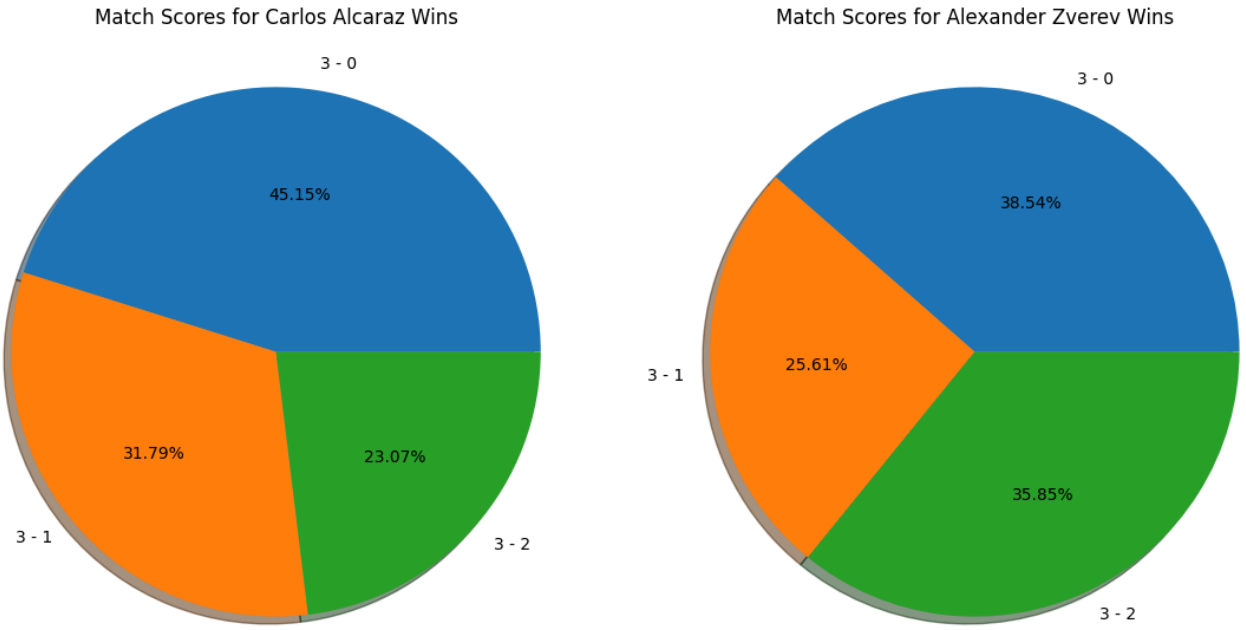


Fig2.

The distributions suggest, again, that Carlos Alcaraz is more likely to win, as Alcaraz wins are more likely to be 3 or 4 matches, while Zverev wins are comparatively more likely to be 4 or 5 set matches according to my model.

Lets look at the quarterfinals onwards of the French open this year, and see who our model thinks has the highest chance of winning.

As each matchup is played for the first time, the model generates the tennis game markov chain by performing a monte carlo simulation of our point markov chain, and if the matchup has already been played, it reuses the already generated markov chain. In this way, we can dramatically speed up the tournament simulation, as generating the tennis game matrix is the biggest bottleneck for our process.

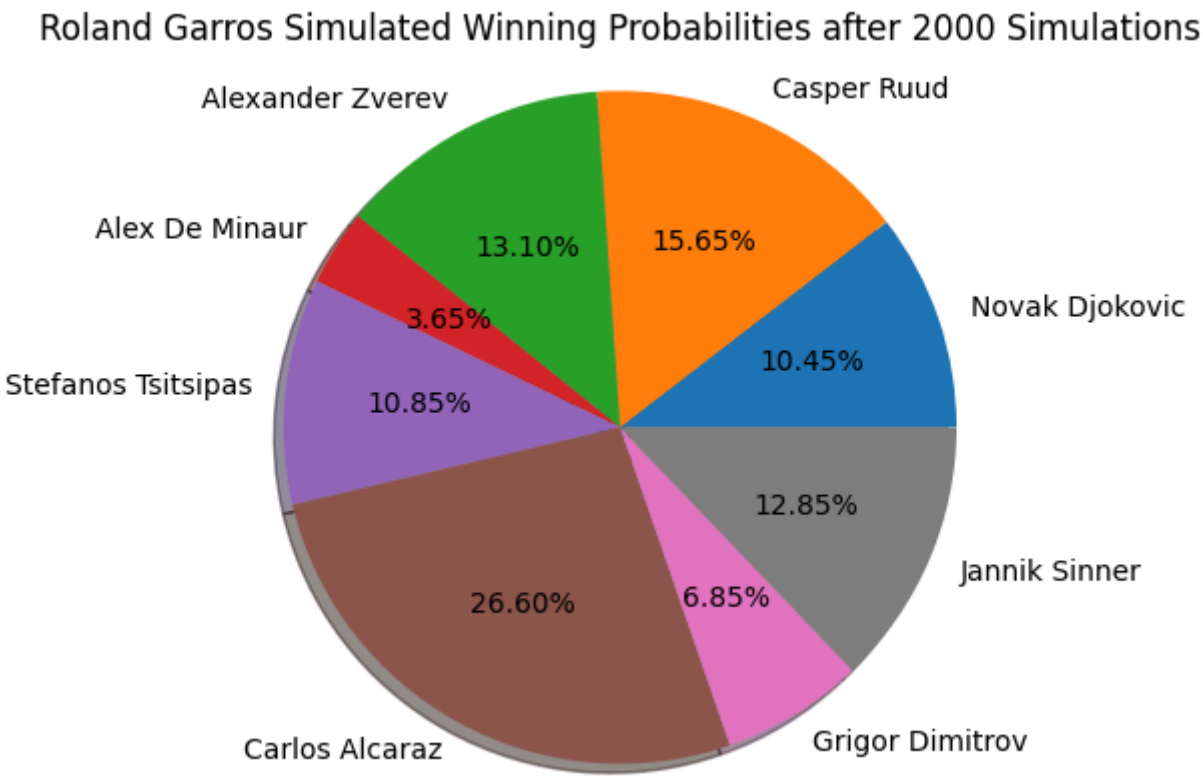


Fig 3.

After simulating the quarterfinals onwards of the French Open 2000 times, we can see that it predicts that the most likely winner is Carlos Alcaraz, while the least likely is Alex De Minaur. When comparing the most likely to least likely winner with betting site “bet365”, we have the following, where green indicates a matching prediction, yellow indicates a prediction that was within 1 place, and red indicates a prediction that was 2 or more places away from bet365’s:

Likelihood to Win Ranking	Our Model	bet365
1	Carlos Alcaraz	Carlos Alcaraz
2	Casper Ruud	Jannik Sinner
3	Alexander Zverev	Novak Djokovic
4	Jannik Sinner	Alexander Zverev
5	Stefanos Tsitsipas	Stefanos Tsitsipas
6	Novak Djokovic	Casper Ruud
7	Grigor Dimitrov	Alex De Minaur
8	Alex De Minaur	Stefanos Tsitsipas

Our mode had 5 / 8 placements that were the same or within 1 place of the predictions by the bet365 website, but relative to the betting site, our model overstated the winning probability of Casper Ruud, and understated the winning probability of Jannik Sinner and Novak Djokovic.

However, when looking at the actual results, our model perfectly predicted the 4 semi finalists as its top 4 players to win the tournament!

We can also look at the French Open winning probabilities over number of tournaments simulated:

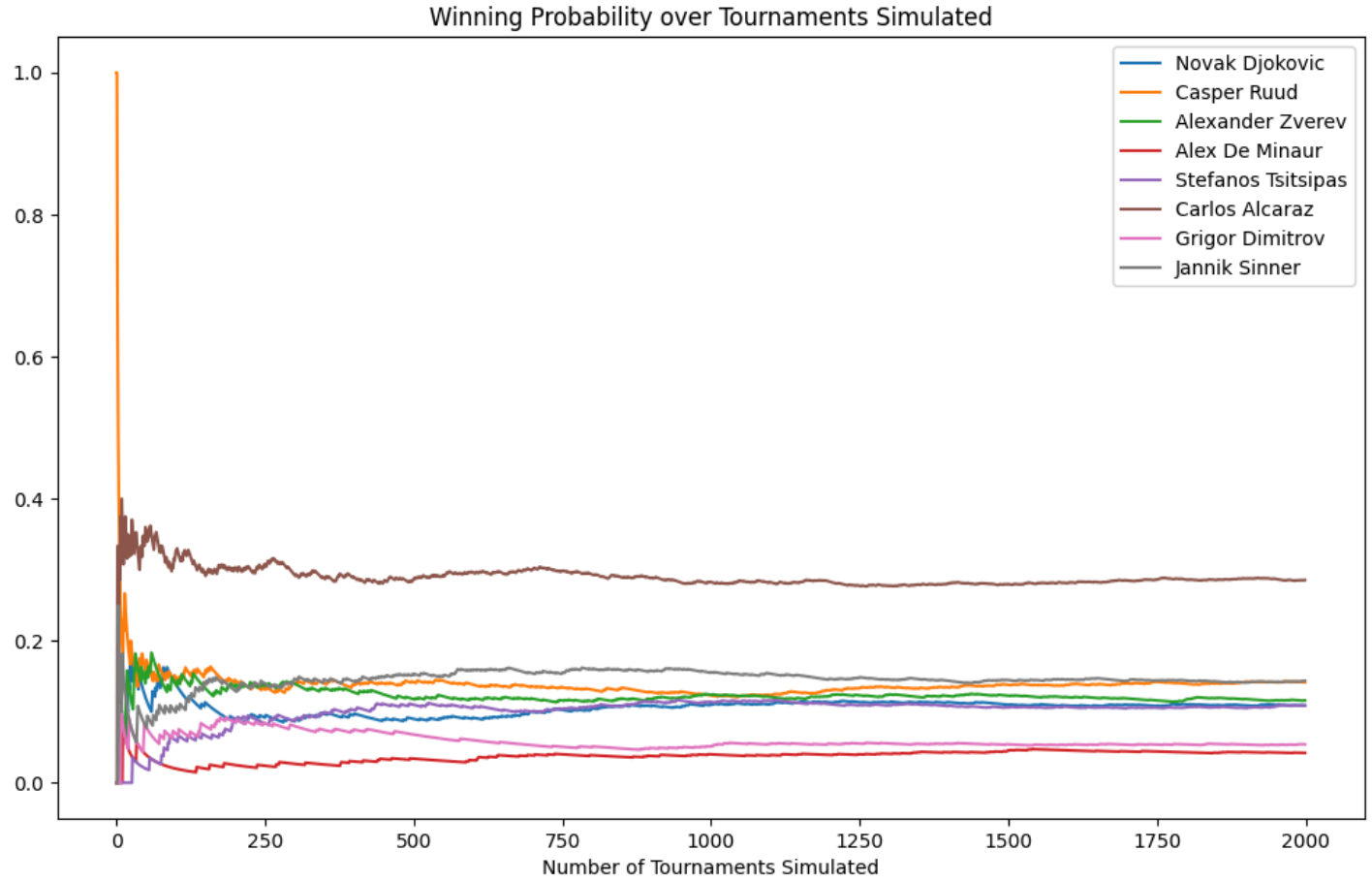


Fig4.

Looking at the winning probabilities over time, we can see a clearer picture rather than looking at the final values. Carlos Alcaraz is still the frontrunner to win, while Djokovic, Ruud, Zverev, Sinner and Tsitsipas are all very similar in their chances to win. These 6 players were the same as the 6 players most likely to win according to betting odds as well.

What is the Most Important Part of Tennis According to My Model?

To see which part of tennis (the serve, return, or rally) is the most important in my model, I made 3 players who are, statistically, the average of all the players on the tennis tour. I then made some changes to create the following 3 players:

Mr Serve:
- Ace proportion is 5% higher
- Double fault proportion is 5% lower

Mr Return:
- All returnable serves are in play (adding 5% goes over 100% so I capped it at 100%)
- Return winner is 5% higher

Mr Rally:
- Rally winner proportion is 5% higher
- Rally unforced error rate is 5% lower

With these 3 players, I simulated a round robin tournament, the calculated probability of winning over 2500 simulations are:

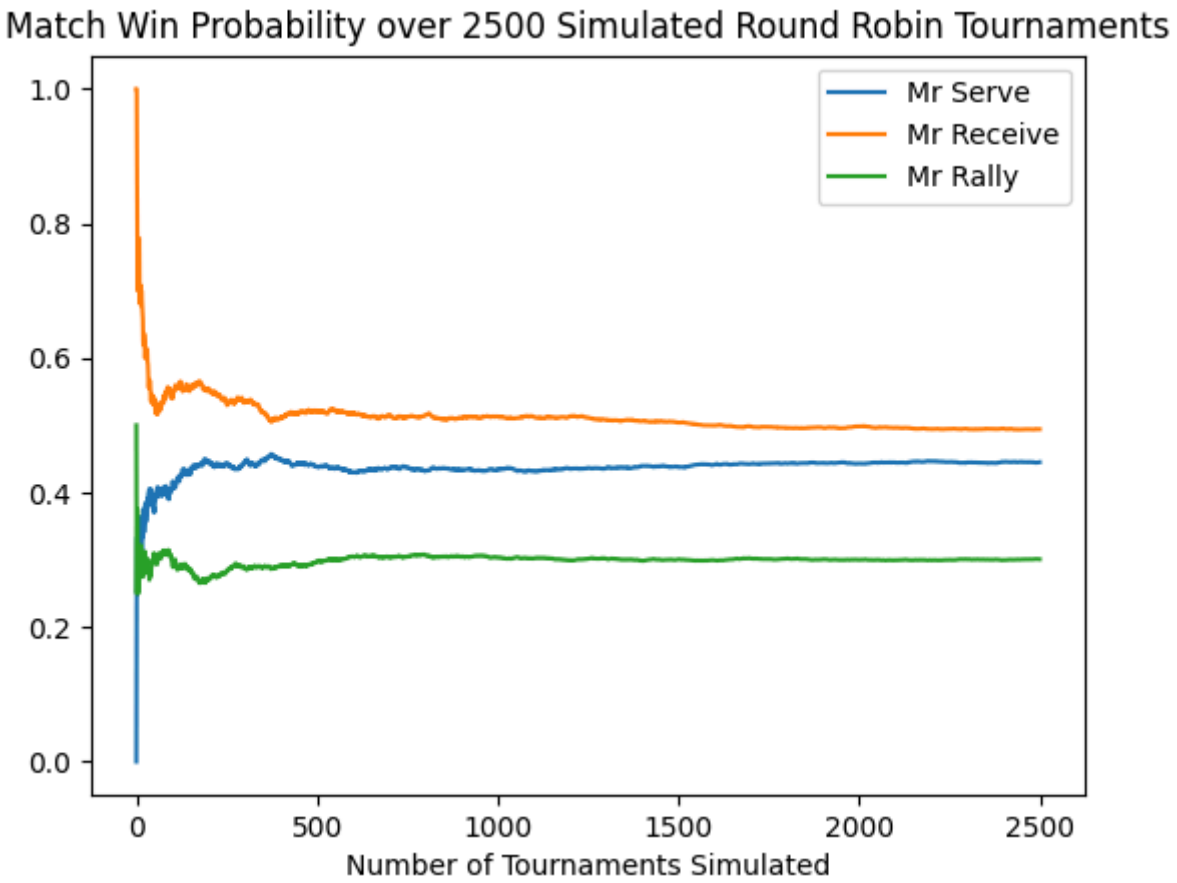


Fig5.

We can see that the player who has above average receiving ends up being the best player out of the 3. However, this may not be totally representative of a real tennis game. The average player on the world tennis tour already has a very high return in play rate (average is 96%), which means that out of all returnable serves, the vast majority are returned in play. So why are “service games” so important in tennis?

What my model fails to capture is the impact that a serve has on the quality of the receive. Although the receiver may be able to return a serve, the server still has a concrete advantage, as the speed and placement of the receive is often not as high quality as a shot in a rally. As my model goes straight from receive to rally, it does not capture the advantage that serving gives you past the possibility of an ace or a forced error.

Furthermore, this understates the performance of a strong server in the model, and overstates the performance of the receiving player in any match. Additionally, a player who excels in the rally but has an average receive will not perform as well as a player who has a strong receive and average rally ability, as to proceed to the rally state, you must first pass the serve or the receive state in the markov chain.

This may be partly why players such as Carlos Alcaraz and Casper Ruud, who excel on and played most of their early career matches on clay courts have such a high chance to win with my model, as clay courts “kick up” the ball, allowing for an easier receive, which would impact the stats of players who play more or less on clay courts than average.

Conclusions

Overall, my model was surprisingly (to me at least) accurate when predicting match outcomes, and the outcome of the French Open past the quarter finals. It correctly predicted the 4 semi finalists, and the tournament winner. This is a small sample size however, and should I run the simulation for the whole tournament, or for other tournaments, the outcomes may be different.

In my opinion, my model overstates the importance of the receive in a game of tennis, and understates the importance of the serve, as it does not capture the impact that a good serve has on the early parts of the rally past the initial receive. It also does not capture the difference in playing surface, or a players ability to employ more sophisticated tactics into their game.

References

Data gathering:
https://www.tennisabstract.com/charting

Player Rankings:
https://www.ultimatetennisstatistics.com

Player ELO ratings:
https://tennisabstract.com/reports/atp_elo_ratings.html

French Open betting odds:
https://www.oddschecker.com/tennis/french-open/mens/mens-french-open/winner