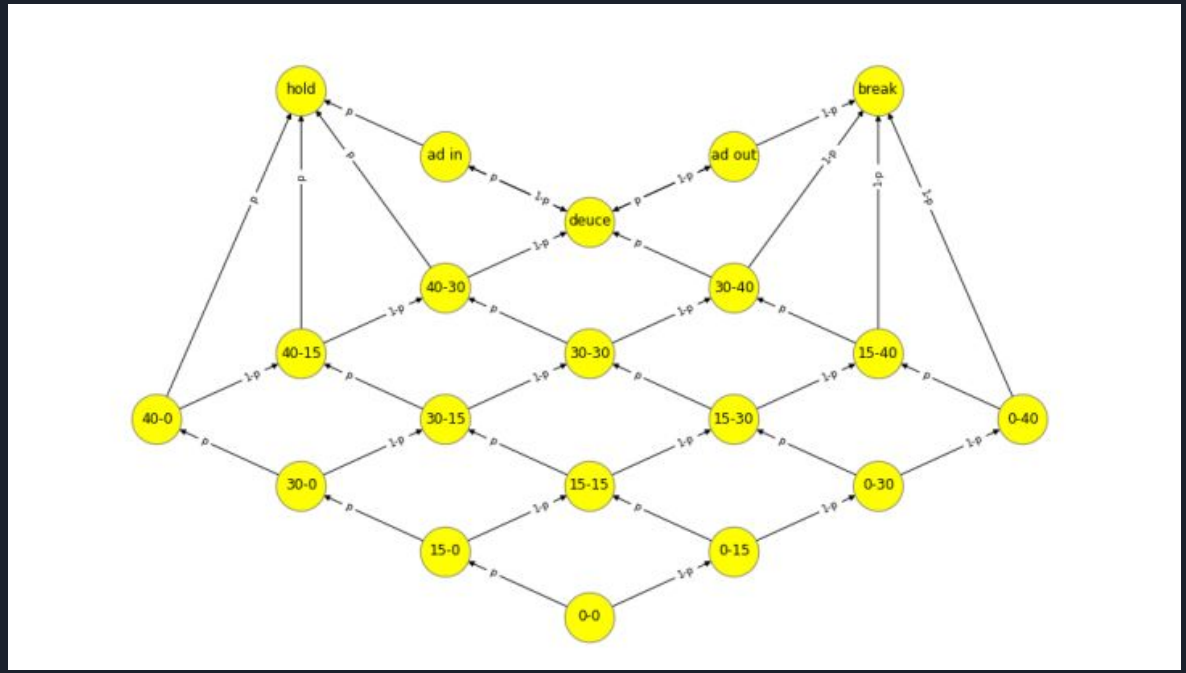
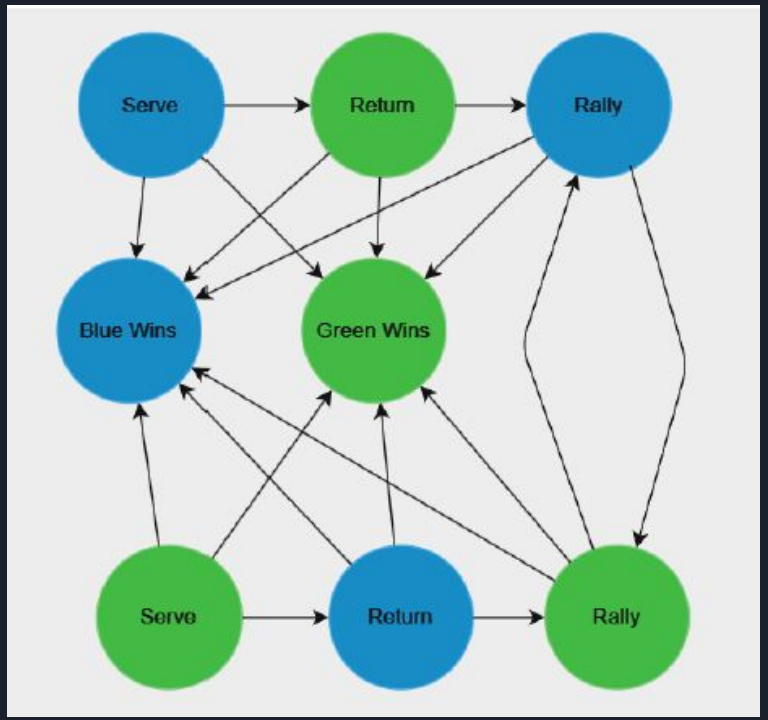
FIT3139 Final Project

Daniel Nguyen
32471033

Base Model:
Tennis Game with Markov Chains



Extension:
How are the transition probabilities calculated?



- Model a point with markov chains
- Run a monte carlo simulation

Questions

- Is my model accurate?
- What makes a good tennis player?

Model Creation: Data Collection

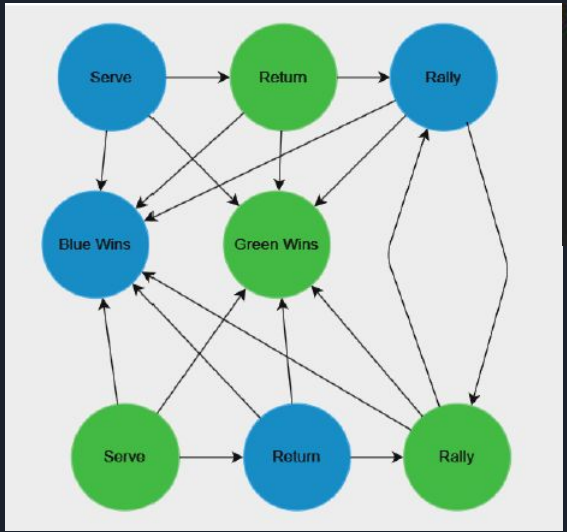
https://www.tennisabstract.com/charting/AlexanderZverev.html

SERVE BASICS	Pts	Won---%	Aces---%	Unret---%	FcdE---%	<=3W---%	Wide---%	Body---%	T---%	DF
All Serves	12490	8166 (65%)	1220 (10%)	337 (3%)	2197 (18%)	4671 (37%)	4871 (39%)	2390 (19%)	5226 (42%)	263 (4%)
First Serves	8279	6059 (73%)	1193 (14%)	312 (4%)	1746 (21%)	3893 (47%)	3765 (45%)	643 (8%)	3869 (47%)	251 (4%)
Second Serves	4211	2107 (50%)	27 (1%)	25 (1%)	451 (11%)	778 (18%)	1106 (26%)	1747 (41%)	1357 (32%)	

RETURN OUTCOMES	Pts	PtsW----%	Returnable	RtbleW--%	inPlay--%	inPlayW-%	Wnr-----%	AvgRally
Total	12951	4904 (38%)	9251 (71%)	4478 (48%)	9095 (98%)	4478 (49%)	134 (1%)	4.5

SHOT TYPES	Total	PtEnd---%	Winner--%	IndFcd--%	UnfErr--%	SvReturn	inPtsW--%	inPtsL--%
Total	45642	8392 (18%)	2685 (6%)	1955 (4%)	3751 (8%)	9147 (20%)	23169 (51%)	22473 (49%)

Model Creation:
A point in tennis and monte carlo




```
while currentState not in [6, 7]:
    transitionProb = self.pointMatrix[currentState]
    nextState = np.random.choice(numStates, p = transitionProb)

    currentState = nextState
```

Model Creation:
Games -> Sets -> Matches -> Tournaments

```
p1gameMatrix = np.array([
    [0, p, 0, 0, 1-p, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], # 0 - 0
    [0, 0, p, 0, 0, 0, 0, 1-p, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], # 15 - 0
    [0, 0, 0, p, 0, 0, 0, 0, 1-p, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], # 30 - 0
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 1-p, 0, 0, 0, 0, 0, 0, 0, 0, p, 0], # 40 - 0
    [0, 0, 0, 0, 0, 1-p, 0, p, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], # 0 - 15
    [0, 0, 0, 0, 0, 0, 1-p, 0, p, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], # 0 - 30
    [0, 0, 0, 0, 0, 0, 0, 0, 0, p, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1-p], # 0 - 40
    [0, 0, 0, 0, 0, 0, 0, 0, p, 0, 1-p, 0, 0, 0, 0, 0, 0, 0, 0, 0], # 15 - 15
    [0, 0, 0, 0, 0, 0, 0, 0, 0, p, 0, 0, 1-p, 0, 0, 0, 0, 0, 0, 0], # 30 - 15
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1-p, 0, 0, 0, 0, p, 0], # 40 - 15
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, p, 0, 1-p, 0, 0, 0, 0, 0], # 15 - 30
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, p, 0, 0, 0, 0, 1-p], # 15 - 40
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, p, 1-p, 0, 0, 0, 0, 0], # 30 - 30
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1-p, 0, 0, p, 0], # 40 - 30
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, p, 0, 0, 0, 1-p], # 30 - 40
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, p, 1-p, 0, 0], # Deuce
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1-p, 0, 0, p, 0], # Advantage Server
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, p, 0, 0, 0, 1-p], # Advantage Receiver
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0], # Server Wins
    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1] # Receiver Wins
])
```

Model Creation: Problems

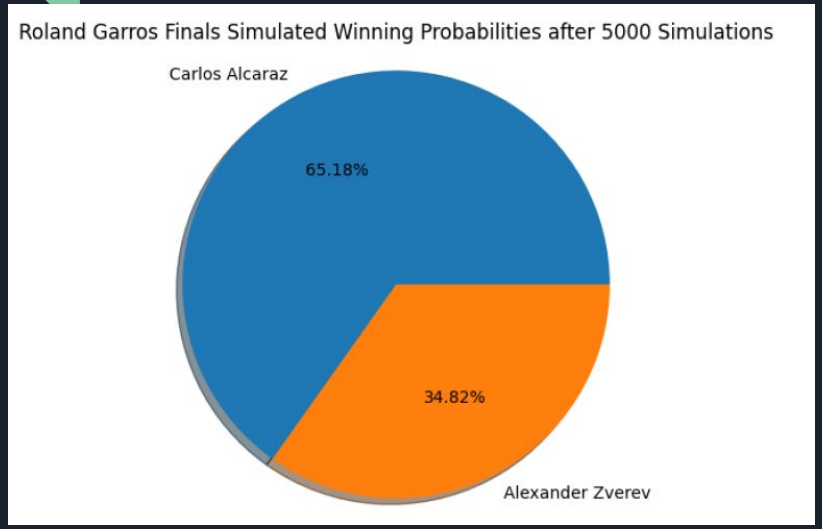
Problem: Doesn’t model everything

Solution: Tough Luck

Problem: Performing point monte carlo for 2000 French Opens is slow

Solution: Dynamic Programming (kind of) and python dictionaries

Results:
French Open Final



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

	Alexander Zverev	Carlos Alcaraz
ELO rating	2074.5	2197.4

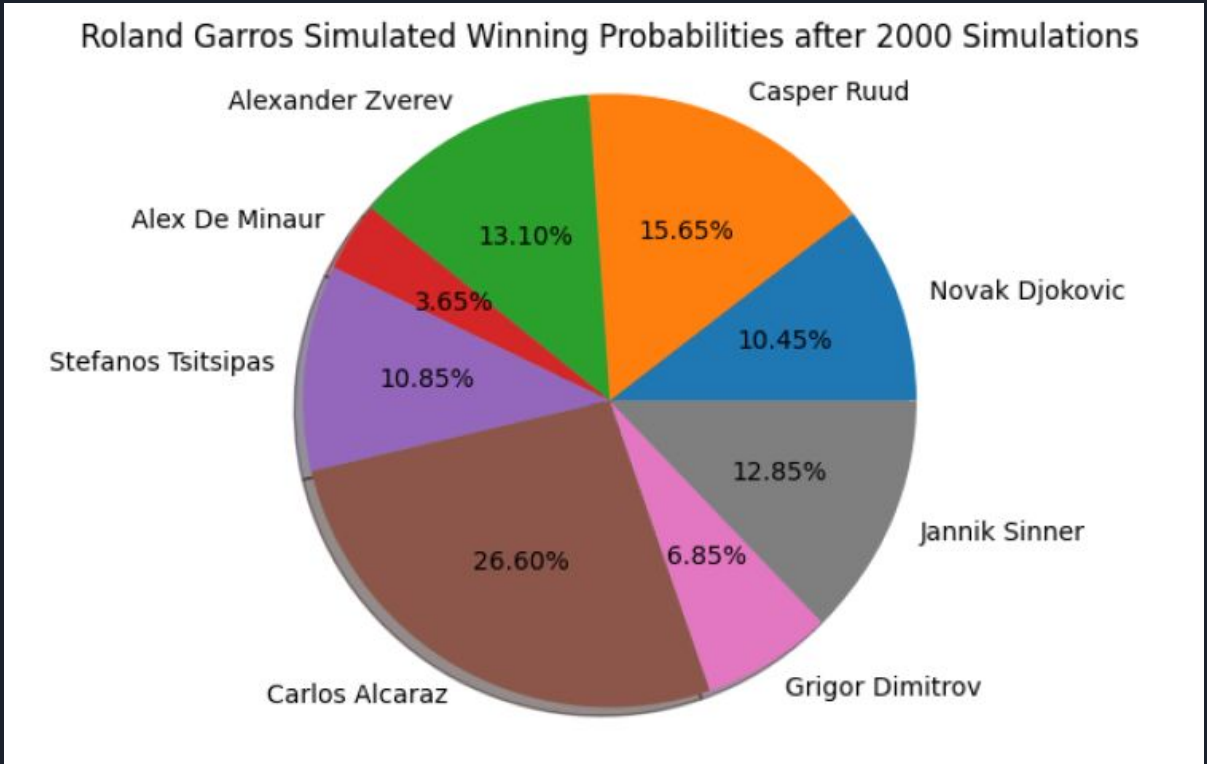
French Open Final: Head to Head

https://www.atptour.com/en/players/atp-head-2-head/carlos-alcaraz-vs-alexander-zverev/a0e2/z355

Carlos Alcaraz Ranking over Time

Alexander Zverev Ranking over Time

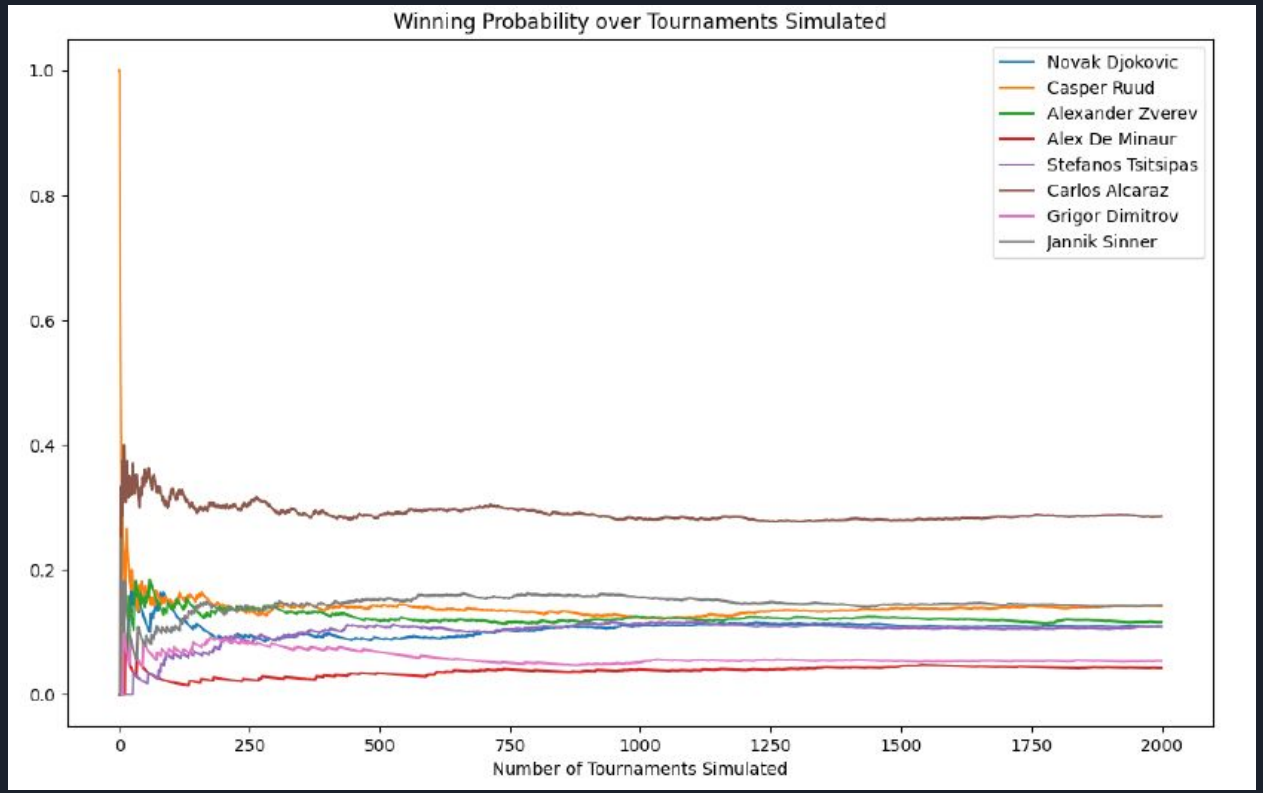
French Open:
Which of the Quarter Finalists Will Win?



French Open:
Comparison with betting odds

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

French Open:
Another Perspective



Recap on our Model: Parts of Tennis

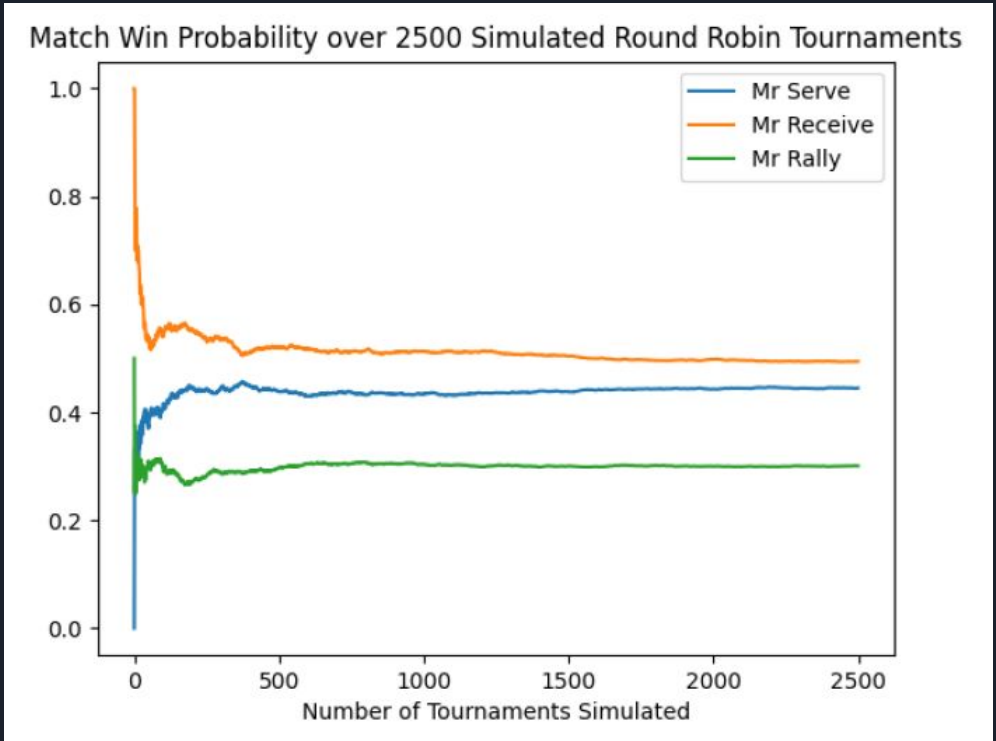
- Serving
- Receiving
- Rally

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

Results:
Serve, Receive and Rally



Results:
Why?

- The serve impacts more than just the receive
- A good receive is needed for a competitive rally
- Players as an average have similar rally statistics
- Model Downfalls:
    - All rally shots are treated the same