A Tennis Match Performance Rating System using Logistic Regression of 32 Statistics

Aidan Sims

The Rating System

- Seek to use match statistics to quantify performance from 0-10
- Similar to passer rating and WhoScored Rating
- Use logistic regression
- Especially useful in high school and college, and doubles

%	Serve	%	Serve
8.7%	Ace %	5.3%	Ace %
1.0%	Double Fault %	1.3%	Double Fault %
74.8%	1st Serve %	69.7%	1st Serve %
83.1%	1st Serve Won %	75.5%	1st Serve Won %
53.8%	2nd Serve Won %	60.9%	2nd Serve Won %
0.0%	Break Points Saved %	0.0%	Break Points Saved %
75.7%	Service Points Won %	71.1%	Service Points Won %
%	Return	%	Return
14.9%	1st Srv. Return Won %	23.6%	1st Srv. Return Won %
52.9%	2nd Srv. Return Won %	65.4%	2nd Srv. Return Won %
66.7%	Break Points Won %	100.0%	Break Points Won %
27.7%	Return Points Won %	37.0%	Return Points Won %
%	Total	%	Total
1.14	Points Dominance	1.28	Points Dominance
52.0%	Total Points Won %	53.5%	Total Points Won %
	Match Time		Match Time

Idea behind Model

- Want to rank a large corpus of data, figure out what leads to a high or low rank
  - Assign top to 10, bottom to 0.
- But that's not very simple
  - Considered RSVM, merge sort, other ranking techniques
- Key Idea: A higher ranked/rated performance should have a higher probability of winning
- Now just have to predict winning
  - Higher certainty is higher rate/rank

Data

- Data from https://www.ultimatetennisstatistics.com/
- Training data from 2022 Montreal and Cincinnati ATP Masters
  - 110 matches so 220 data points
- Test data from 2022 ATP Finals
  - 15 matches so 30 data points
  - Plus fake matches to test corner cases

Data

- Store these statistics in a Python class

- Whether the player won (as a binary indicator)
- Ace Percentage (Aces per Service Point)
- Double Fault Percentage
- First Serve Percentage
- First Serve Win Percentage
- Second Serve Win Percentage
- Break Points Saved Percentage
- Service Points Won Percentage
- First Serve Return Points Won Percentage
- Second Serve Return Points Won Percentage
- Break Points Won Percentage
- Return Points Won Percentage
- Points Won Percentage (All points)
- Aces per Service Game
- Double Faults per Second Serve
- Double Faults per Service Game
- Service In-Play Points Won
- Points per Service Game
- Points Lost per Service Game
- Break Points per Service Game
- Service Games Won Percentage
- Service Games Lost per Set
- Return In-Play Points Won
- Points per Return Game
- Points Won per Return Game
- Return Games Won Percentage
- Winner Percentage (Winners per Point Won)
- Unforced Error Percentage (Unforced Errors per Point Lost)
- Forced Error Percentage
- Points Played at the Net Percentage
- Net Points Won Percentage
- Points Won at the Net Percentage (Net Points Won per Point Won
- Average Second Serve Speed to First Serve Speed Ratio
- Average Serve Speed to Maximum Serve Speed Ratio
- Games Won Percentage

$$g(z) = \frac{1}{1 + e^{(-z)}}$$

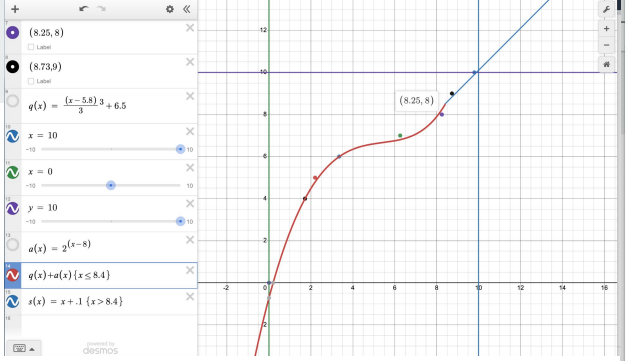
Logistic Regression

- In a regression, we look at how each individual input (statistic) affects the output (% chance of winning)
  - “Logistic” constrains our output
- Have a set of weights which are initially 0
- For 900,000 times:
  - See how far off we are with prediction and error functions
  - Use calculus to see how we can update our model

$$h = g(X\theta)$$

$$J(\theta) = \frac{1}{m} \cdot \left(-y^T \log(h) - (1 - y)^T \log(1 - h)\right)$$

$$\theta_{t+1} = \theta_t - \eta \nabla L(f(x; \theta), y)$$



Results

After running:

- 100 % accuracy on training data, 91% on test

Need to compress the sigmoid function by a factor of 500,000

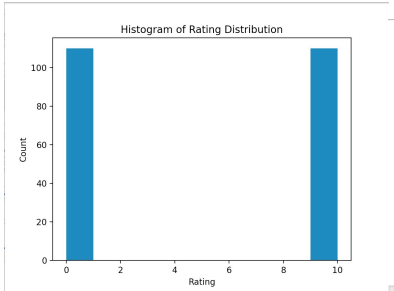
- 86% on training data, 84% on test
- Average rating of 6.6

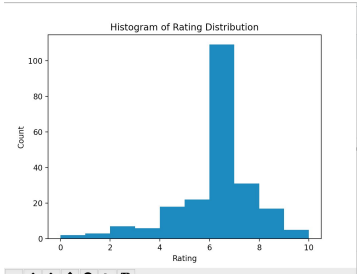
To get rating, multiply by 10.

If output > 8.4, add .1

Otherwise, rating =((x-5.8)/3)^3+2^(x-8)+6.5

```
Points Won per Return Game: 251228.26044414777
Return Games Won %: 73296.4579668816
Games Won %: 69936.87120564694
Break Points Won %: 64839.01104321236
Aces per Service Game: 64654.8298387351
Points per Return Game: 62381.04717280838
Service Games Won %: 60917.39687420993
Break Points Saved %: 57111.80109493688
Return Points Won %: 39844.029056681706
1st Serve Return Points Won %: 36208.54698458697
Service Points Won %: 33408.02659834564
Total Points Won %: 31887.166917162504
2nd Serve Return Points Won %: 31558.480548716754
Return in Play Points Won %: 31112.79776881266
1st Serve Won %: 29596.397874260252
2nd Serve Won %: 26273.148454328733
Serve in Play Points Won: 25440.26904668242
Net Points Won %: 21526.757584653264
Forced Error %: 13959.012280759684
Ace %: 11360.772800111796
1st Serve %: 4502.6193957814985
Return to Service Points Ratio: -366.73736830684794
Net Points Played %: -1233.4547197224801
Winner %: -1862.0304604789148
DF %: -5670.169724754847
Points Won at Net %: -6317.107819530779
Double Faults per Second Serve %: -12328.529493552753
Unforced Error %: -13627.35702253192
DFs per Service Game %: -40540.391441273365
Points per Service Game: -80998.84293565416
Break Points faced per Service Game: -89646.83896531095
Points Lost per Service Game: -209633.195151262
Service Games Lost per Set: -283257.88424522063
```

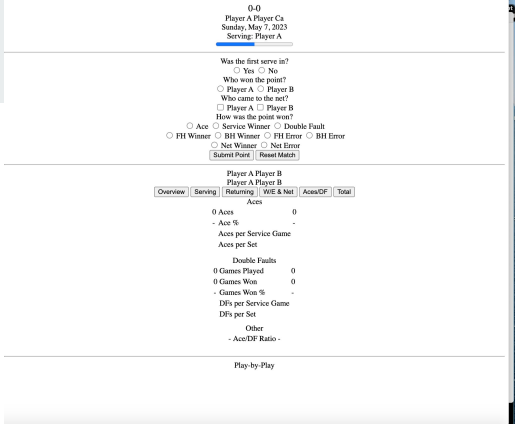




Testing

2023 Madrid Final (May 7th)
Alcaraz d. Struff 6-4, 3-6, 6-3
Model:
Alcaraz - 6.81
Struff 6.21

- For a “perfect match”, a rating of 9.85 is given
  - For the opposite, a 0
- Model seems to reward good play well
- But it is hard to test the model exactly because ratings are subjective
- Need to test it for matches in progress



Next Steps

- Further testing is needed
- Either on data or on live matches
- Add as a feature into another project

Thank you!