MARKOV CHAINS, MACHINE LEARNING, AND TENNIS

ABOUT US

01

Alex Kwok

- Senior in CC
- Majoring in Applied Math
- Grew up watching Tennis with cousins, Best Friend plays pro overseas

02

Vittorio Luzio

- Senior in SEAS
- Majoring in Applied Math
- Huge fan of tennis, used to play in high school

03

Alex Medina

- Senior in CC
- Majoring in Astrophysics, Applied Math
- Sports Enthusiast

04

Eric Shin

- Senior in SEAS
- Majoring in Applied Math
- Loves sports in general
- Tennis team captain back in high school

PRIMARY REFERENCES

Moretti, Christopher. (2015). Monte Carlo Tennis: A Stochastic Markov Chain Model. Analyzing a Tennis Game with Markov Chains

Newton, Paul. (2009). Journal of Quantitative Analysis in Sports. Monte Carlo Tennis: A Stochastic Markov Chain Model

De Seranno, Alexander. (2021). Predicting Tennis Matches Using Machine Learning. Predicting Tennis Matches Using Machine Learning

Praet, Robin. (2017) Predicting Sport Results by using Recommendation Techniques. Robin Praet Techniques Predicting Sport Results by using Recommendation

DATA SET

ATP tennis rankings, results, and stats:
GitHub - JeffSackmann/tennis_atp: ATP Tennis Rankings, Results, and Stats

Ultimate Tennis Statistics
Ultimate Tennis Statistics

TABLE OF CONTENTS | OUTLINE

SPORTS ANALYTICS

- Analytics in sports have been popularized by movies such as Moneyball, where computer-driven data analysis was used to find undervalued baseball players
  - More recently, analytics to determine winning tactics, develop practice regimens, and scout opposing players have come to the forefront of major sports

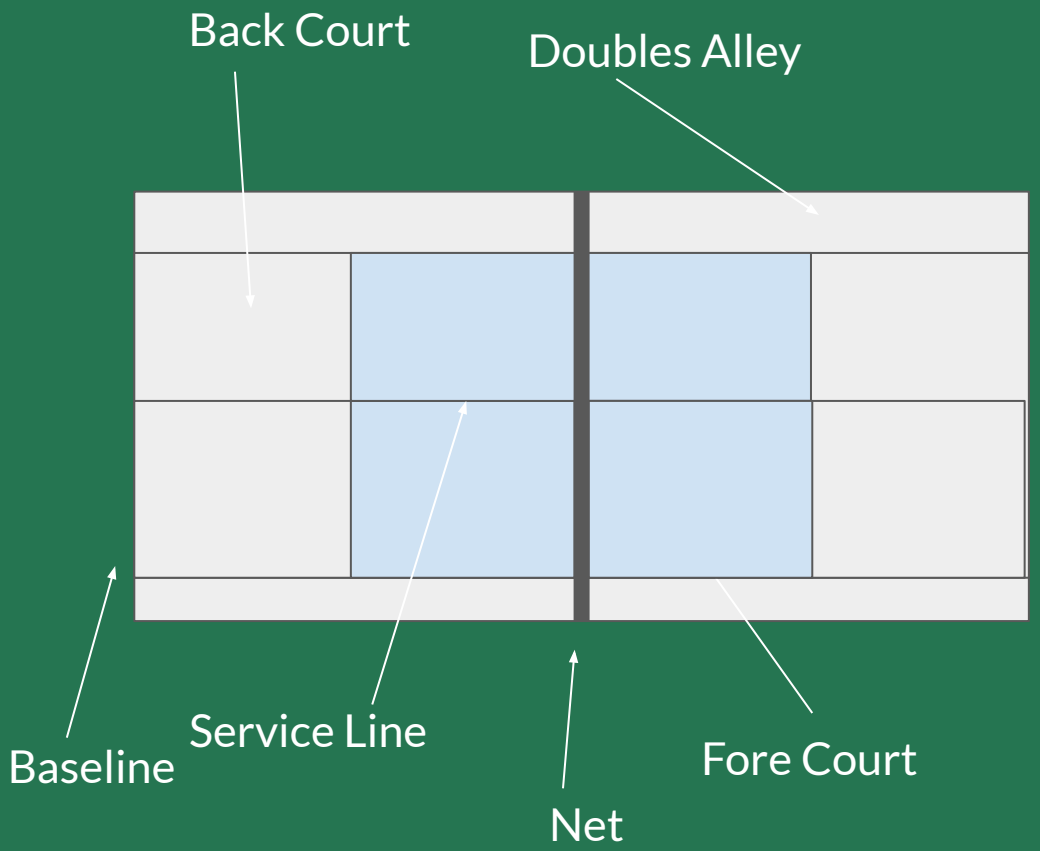
- Sports analytics plays a crucial role in modern sports and it's importance stems from several key aspects
  - Performance Optimization
  - Talent identification and recruitment
  - Injury Prevention and Management

WHY TENNIS?

- General Interest in Sports and Predictive Models
- Tennis is an easily quantifiable sport
  - Binary Outcomes
  - Around 250 points per game
- Extensive Data Available
  - Player Data, Rankings, Historical Performance
- Global Interest
  - International Sport

TENNIS GAME SCORING

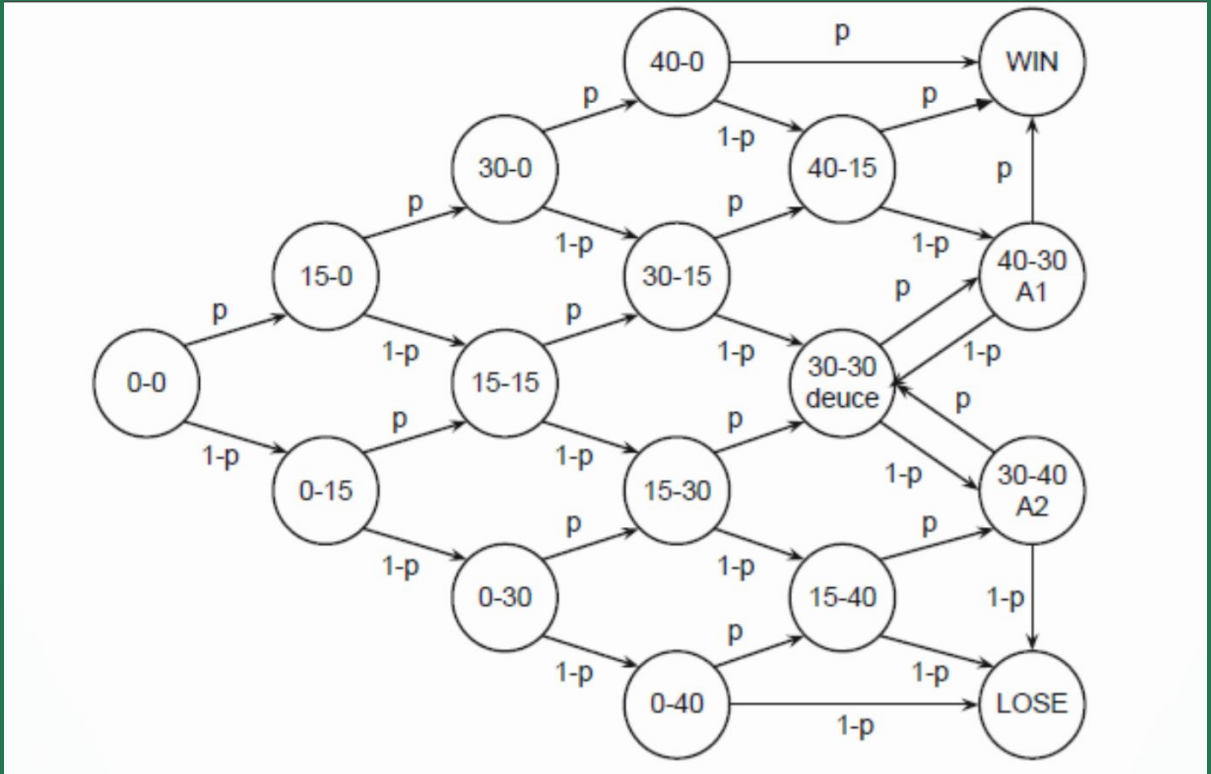
- Point System in a Game
  - Scoring terms: 0 (love), 15, 30, 40, game point
  - Requires at least a two-point lead to win
- Deuce
  - Occurs at a 40-40 score
  - Indicates an equal score requiring a two-point lead to win
- Advantage
  - After deuce, the next point leads to “advantage”
  - Winning at advantage wins the game; losing returns to deuce



TENNIS SET SCORING

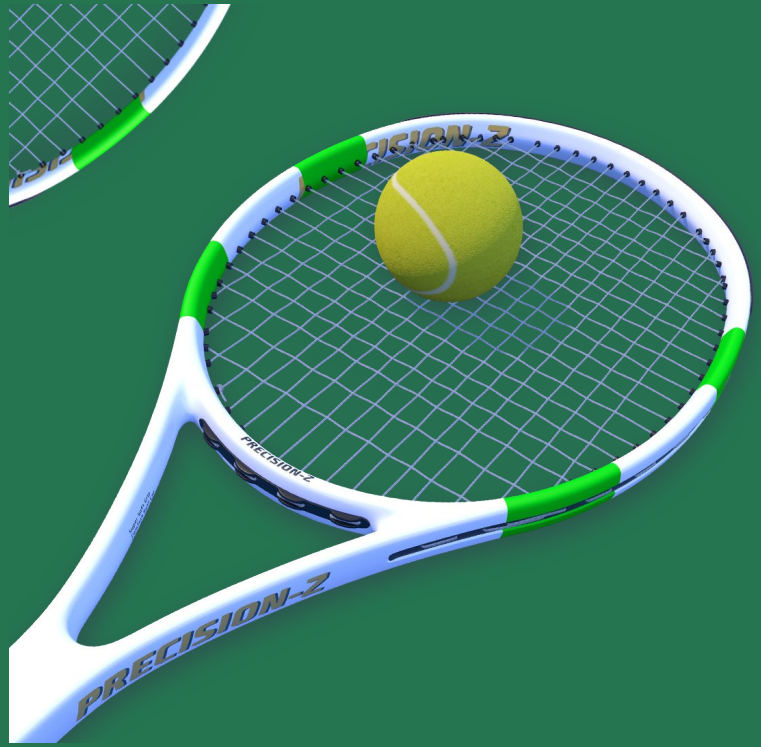
- Set Play
  - Consists of several games
  - First to 6 games wins the set
  - Requires at least a two-game lead to win
- Tie-break
  - Used at 6-6 in games
  - First to 7 points wins the set
  - Requires at least a two-point lead to win

GRAPHICAL REPRESENTATION



MATCH FORMAT

- Best-of-Three
  - Typically played in this format
  - Winning two sets wins the match
- Grand Slam Variations
  - Becomes a Best-of-Five format
  - Final sets continue until one wins two sets in a row if it is tied 6-6



SERVING ADVANTAGE

- A Strategic Edge
  - Server has the initial control of the point
  - Can set the pace and style of play
- Why advantageous?
  - Only shot not dictated by an opponent
  - Can score quick points (aces, service winners)
  - Psychological advantage by starting on the front foot
- Statistics
  - Higher win rates on serve
  - Breaking serve considered a significant event

MARKOV CHAINS

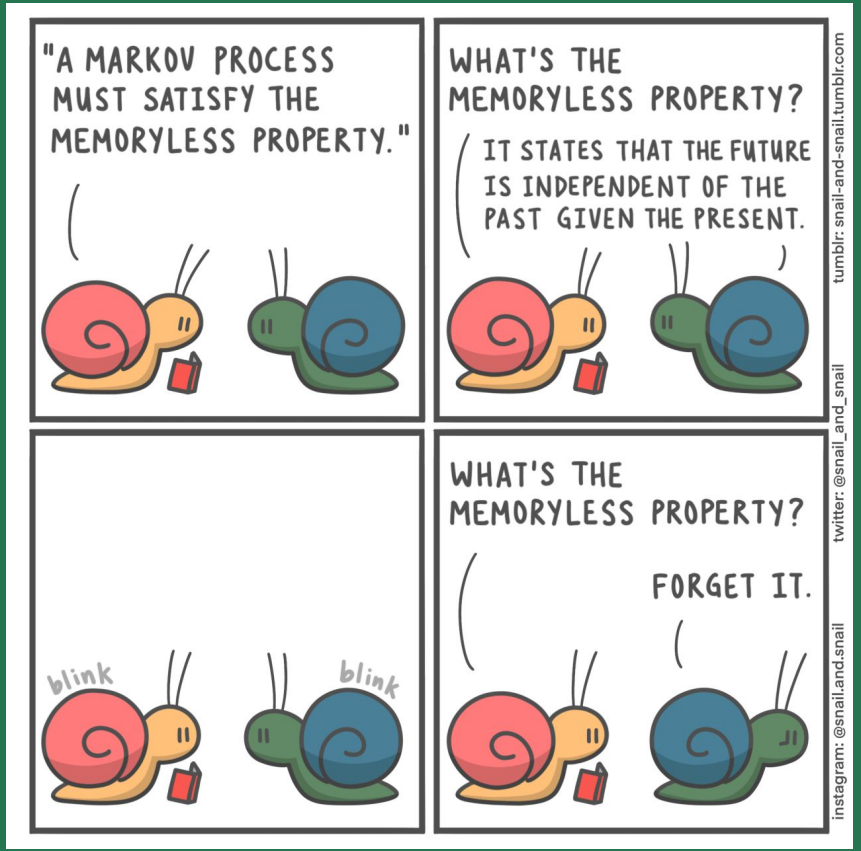


Image Credit : Snail and Snail Blog

WHAT ARE MARKOV CHAINS?

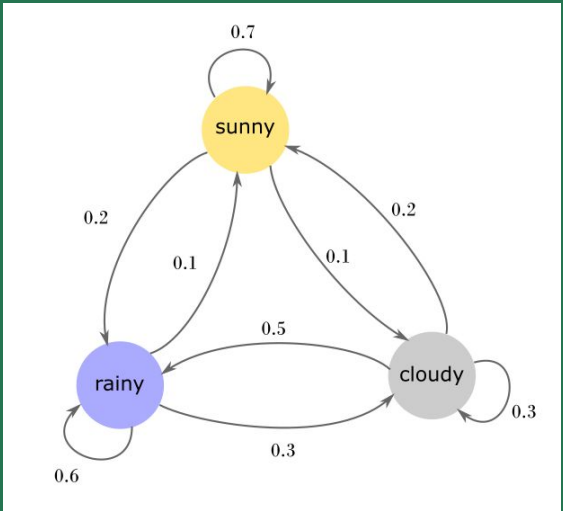
- “A markov chain is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event.” (Markov chain - Wikipedia)

- Developed by Andrey Markov in the early 20th century

- Key Feature: Memoryless

WHAT ARE MARKOV CHAINS?

- States: Distinct conditions or positions in which the system can exist
- Transition probability: Probability of moving from one state to another
- Transition matrix: A matrix representing the transition probabilities



$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}$$

Image Credit: Markov Chain Diagram

MATH BEHIND THE MARKOV CHAIN

Definition: A discrete-time stochastic process is a Markov chain if, for t = 0,1,2… and all states,

$$P(X_{n+1} = x_{n+1} \mid X_1 = x_1, \ldots, X_n = x_n) = P(X_{n+1} = x_{n+1} \mid X_n = x_n)$$

CLASSIFICATION OF STATES

- Transient States: Not guaranteed to return to once left. (Deuce)

$$\sum_{n=1}^{\infty} P(X_n = i \mid X_0 = i) < \infty$$

- Recurrent States: Guaranteed to be revisited eventually. (0-0)

$$\sum_{n=1}^{\infty} P(X_n = i \mid X_0 = i) = \infty$$

- Absorbing States: Once entered, you cannot leave. (End of the game)

$$P(X_{n+1} = i \mid X_n = i) = 1$$

STEADY-STATE DISTRIBUTION

- Definition: A probability distribution that remains unchanged as the system evolves
- Long-term probability that the system will be in each state

$$\pi P = \pi \qquad \sum_i \pi_i = 1$$

$$\begin{aligned}
\pi_1 p_{11} + \pi_2 p_{21} + \cdots + \pi_n p_{n1} &= \pi_1 \\
\pi_1 p_{12} + \pi_2 p_{22} + \cdots + \pi_n p_{n2} &= \pi_2 \\
&\vdots \\
\pi_1 p_{1n} + \pi_2 p_{2n} + \cdots + \pi_n p_{nn} &= \pi_n \\
\pi_1 + \pi_2 + \cdots + \pi_n &= 1
\end{aligned}$$

STEADY-STATE DISTRIBUTION

- Eigenvalue Approach

$$P^T \mathbf{v} = \lambda \mathbf{v}$$

- Iterative Methods: Power method
  - Useful when state space is large or transition matrix is not easily diagonalizable
  - Our Markov Model

ASSUMPTIONS OF MARKOV CHAINS

- The probabilities of moving from a state to all others sum to one.
- The probabilities apply to all system participants.
- The probabilities are constant over time.

WHY MARKOV CHAINS?

- State-based Predictions
- Memoryless Property
- Sequential Nature
- Model Simplicity and Efficiency

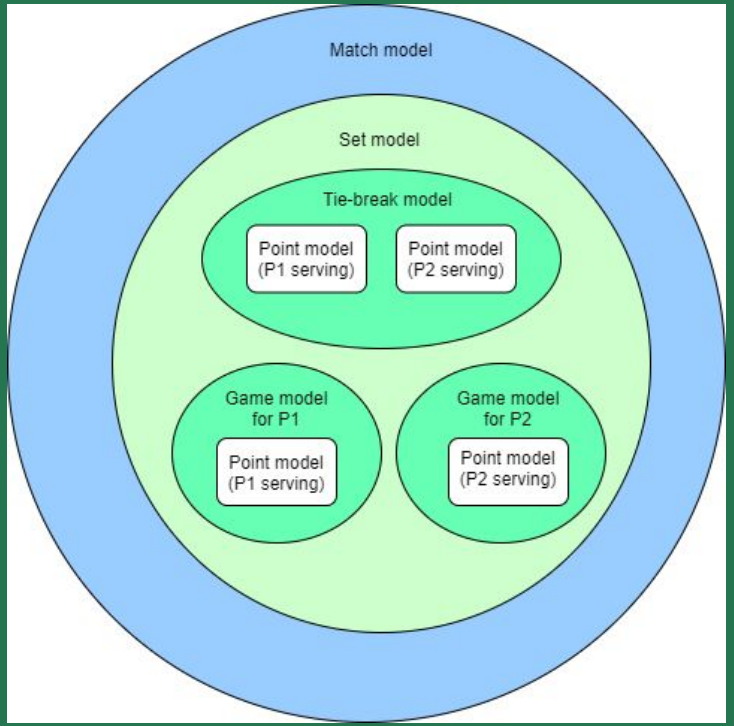


Image Credit : Tennis Markov Model

LIMITATIONS OF MARKOV CHAINS

- Oversimplification
- Assumptions
  - Memorylessness
  - Homogeneity over time
- Data Limitations
- Neglect of External Factors
- Code takes really long to execute - 1h 30min

DATA COLLECTION

- Data Scraping from Ultimate Tennis Statistics
  - Service Point Winning Percentage for each player *on hard court*
  - Tournament bracket for each year's US Open
  - Actual winner of each year's US Open

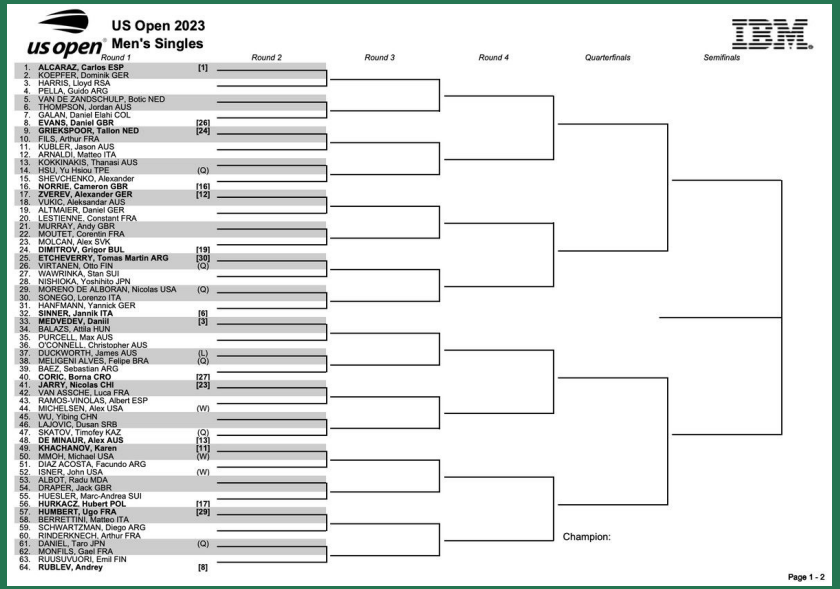


Image Credit: US Open 2023 Bracket

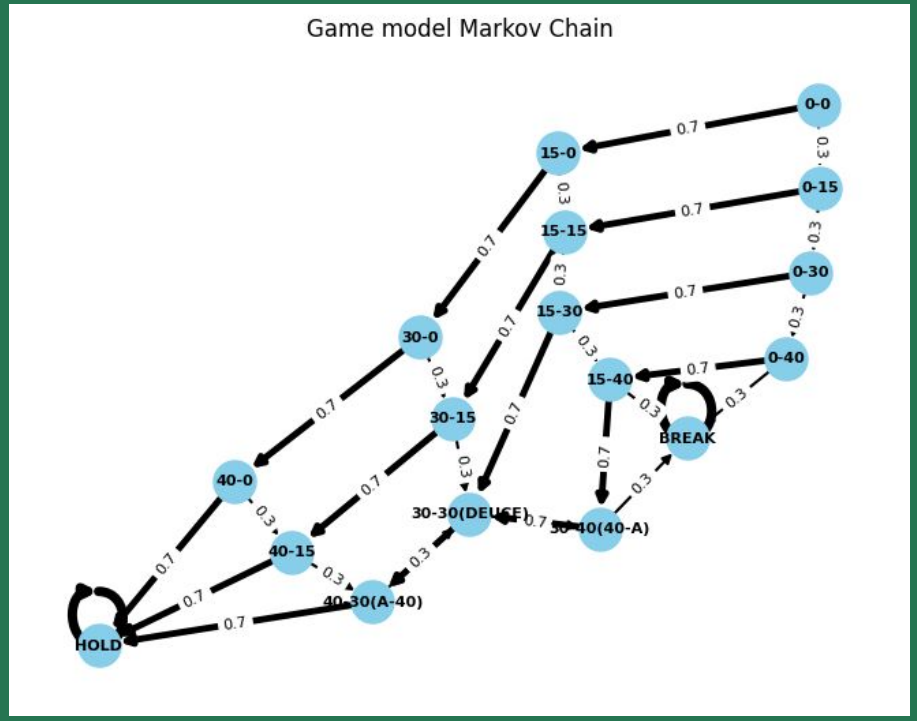
PSEUDOCODE

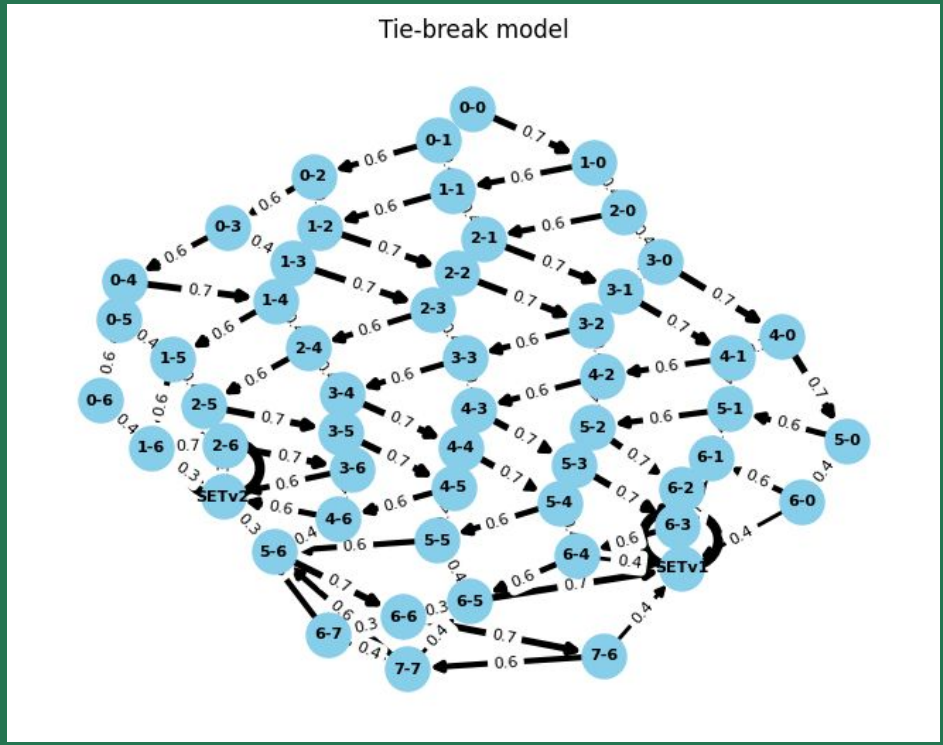
```
Initialize game_states, tiebreak_states, set_states,
match_states

function createTransitionMatrix for each of 4 stages

function computeProbabilities(tMat, init):

    multiply init with tMat raised to a high power

    return computed probabilities
```

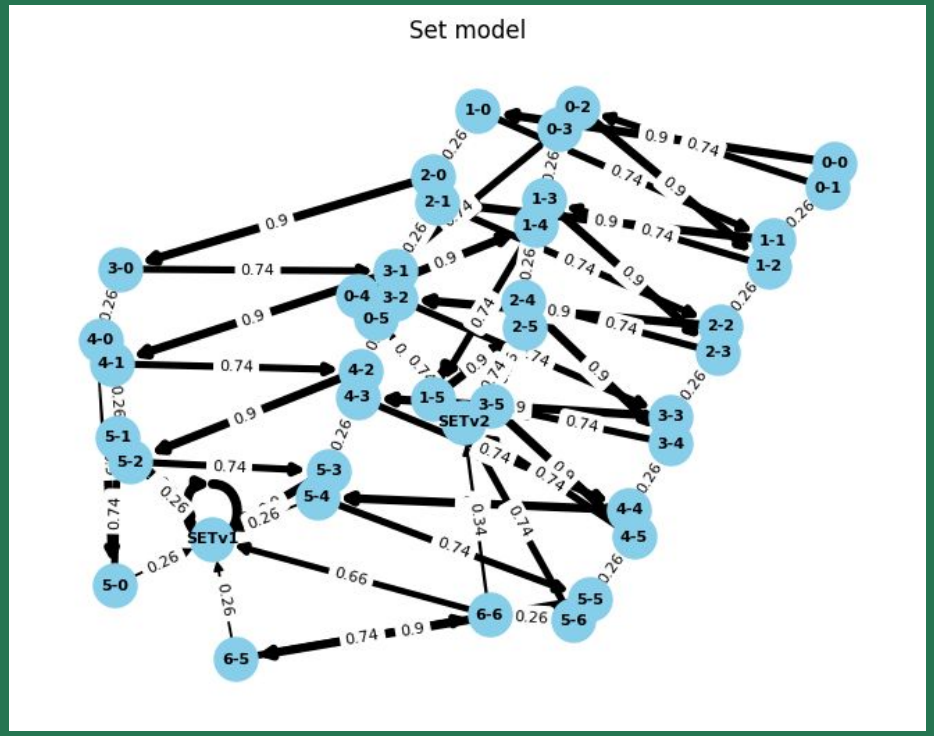
PSEUDOCODE

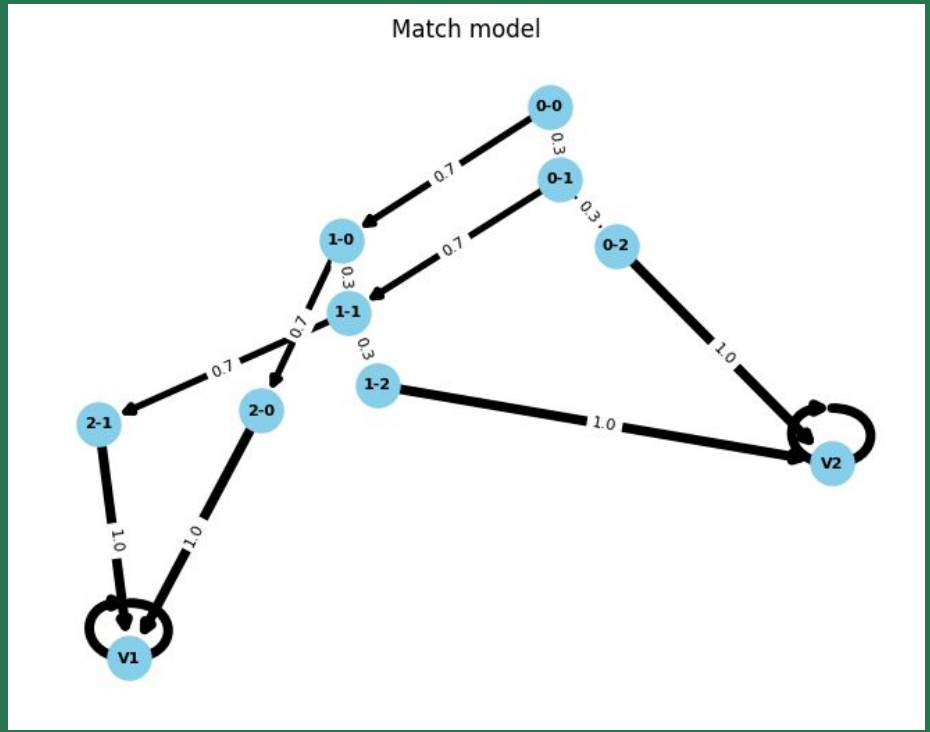
```
function predictMatch for each of 3 states (game, set, match)
    update init based on current state
    calculate probabilities for match outcomes
    return probabilities
function simulateMatch(p1, p2, stats)
function simulateTournament(matches, stats)
function main()
```

DEMO


Game model Markov Chain
0-0
15-0
0-15
15-15
0-30
30-0
15-30
0-40
15-40
30-15
BREAK
40-0
30-30(DEUCE)
40-40(40-A)
40-15
40-30(A-40)
HOLD


Tie-break model
SETv2
SETv1


Set model
1-0
0-2
0-3
0-0
0-1
2-0
2-1
1-3
1-4
1-1
1-2
3-0
3-1
0-4
3-2
0-5
2-4
2-5
2-2
2-3
4-0
4-1
4-2
4-3
1-5
3-5
SETv2
3-3
3-4
5-1
5-2
5-3
5-4
4-4
4-5
SETv1
5-0
6-6
5-5
5-6
6-5
0.9
0.74
0.26
0.66
0.34


Match model
0-0
0-1
0-2
1-0
1-1
1-2
2-0
2-1
V1
V2
0.7
0.3
1.0

Predicting a game

```
Match Setup:
Player A Service Point Winning Percentage: 0.7
Probability of each player winning a game:
    0    1    2    3    4    5    6    7    8    9    10   11   12   13   14  \
0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0

          15        16
0  0.900789  0.099211
```

Predicting a set

```
Match Setup:
Player A Service Point Winning Percentage: 0.7
Player B Service Point Winning Percentage: 0.6
Probability of each player winning a set:
    0    1    2    3    4    5    6    7    8    9    10   11   12   13   14  \
0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0

    15   16   17   18   19   20   21   22   23   24   25   26   27   28   29  \
0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0

    30   31   32   33   34   35   36   37   38        39        40
0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.562058  0.437942
```

Predicting a tiebreak

```
Match Setup:
Player A Service Point Winning Percentage: 0.7
Player B Service Point Winning Percentage: 0.6
Probability of each player winning a tiebreak:
    0    1    2    3    4    5    6    7    8    9    10   11   12   13   14  \
0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0

    15   16   17   18   19   20   21   22   23   24   25   26   27   28   29  \
0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0

    30   31   32   33   34   35   36   37            38   39   40   41   42  \
0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  1.412584e-133  0.0  0.0  0.0  0.0

    43   44   45   46   47   48   49   50   51        52        53
0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.662973  0.337027
```

Predicting a match

```
Match Setup:
Player A Service Point Winning Percentage: 0.7
Player B Service Point Winning Percentage: 0.6
Probability of each player winning a match:
     0    1    2    3    4    5    6    7         8         9
0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.592608  0.407392
```

LOGISTIC REGRESSION

Image Credit : Logistic Regression Meme

WHAT IS LOGISTIC REGRESSION

- Logistic regression is a statistical modeling technique used for classification and predictive analytics.
- The resulting model helps us predict whether a given match will fall into the 'winning' or 'losing' category.
- Classification by setting a cutoff value
- Ideal for the tennis — whether it's a win or a loss
- S-Curve

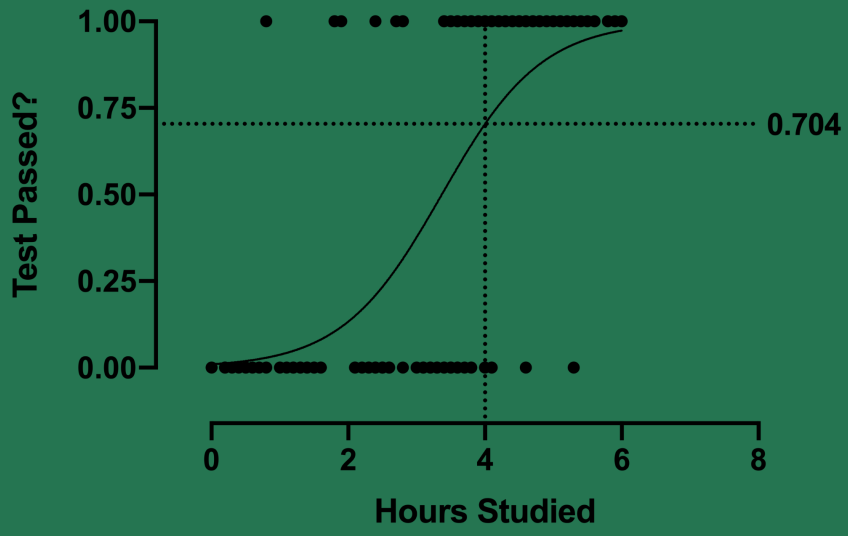


Image credit

KEY FEATURES OF LOGISTIC REGRESSION

- Information about the thing we're trying to classify using coefficients
- Sigmoid function to turn the result of the calculation into a number between 0 and 1.

$$h_w(x) = \phi(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)$$

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

- Binary classification

$$P(y = 1 \mid x; w) = h_w(x), \quad P(y = 0 \mid x; w) = 1 - h_w(x)$$

KEY FEATURES OF LOGISTIC REGRESSION (CONT)

- Our goal is to find the best model parameters, w, p is a functions of the sigmoid

$$L(w) = \prod_{i=1}^{N} p(y_i \mid x_i; w)$$

$$\max_w L(w) = \max_w \prod_{i=1}^{N} p\left(y^{(i)} \mid x^{(i)}; w\right)$$

- We can convert the maximization problem into minimization so that we can write the loss function. This is the logistic regression loss function used in Scikit-Learn.

$$L_{\log}(w) = -\log L(w) = -\sum_{i=1}^{N} \log p\left(y^{(i)} \mid x^{(i)}; w\right)$$

KEY FEATURES OF LOGISTIC REGRESSION (CONT)

- The negative log-likelihood quantifies how well the model's predicted probabilities align with the actual labels

$$P(y = 1 \mid x; w) = h_w(x), \quad P(y = 0 \mid x; w) = 1 - h_w(x)$$

$$L_{\log}(w) = -\log L(w) = -\sum_{i=1}^{N} \log p\left(y^{(i)} \mid x^{(i)}; w\right)$$

$$= -\sum_{i=1}^{N} y^{(i)} \log(p(y = 1|x^{(i)}, w)) - \sum_{i=1}^{N} (1 - y^{(i)}) \log(p(y = 0|x^{(i)}, w))$$

LOGISTIC REGRESSION (EXTRA INFO)

- Hyperparameter tuning
  - Techniques like RandomizedSearchCV help find the best hyperparameters for optimal model performance
- Evaluation metrics
  - Accuracy, precision, recall, and area under the ROC curve

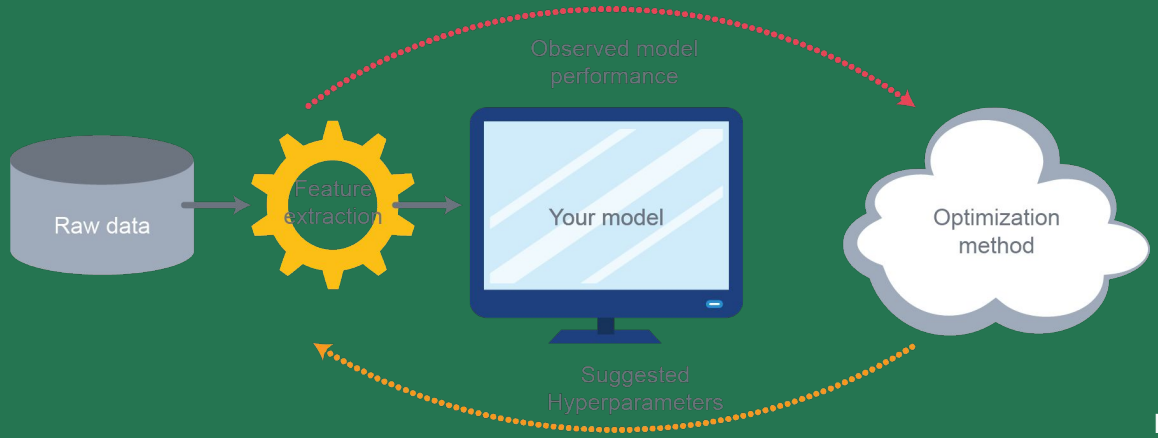


Image Credit: Hyperparameter

WHY LOGISTIC REGRESSION?

- Outcome is very easy to interpret
- Can handle changes over time
  - Like player performance as a tennis match evolves
- Markov Chain model showed limitations in predicting outcomes
- Flexibility in choosing relevant features
  - MC only had 2 probabilities, where LR had 36

LIMITATIONS OF LOGISTIC REGRESSION

- Large sample size vs. small sample size
  - Overfitting
- Outliers can significantly influence the estimated coefficients
- Assumes that each observation in your dataset is independent of the others

Regression

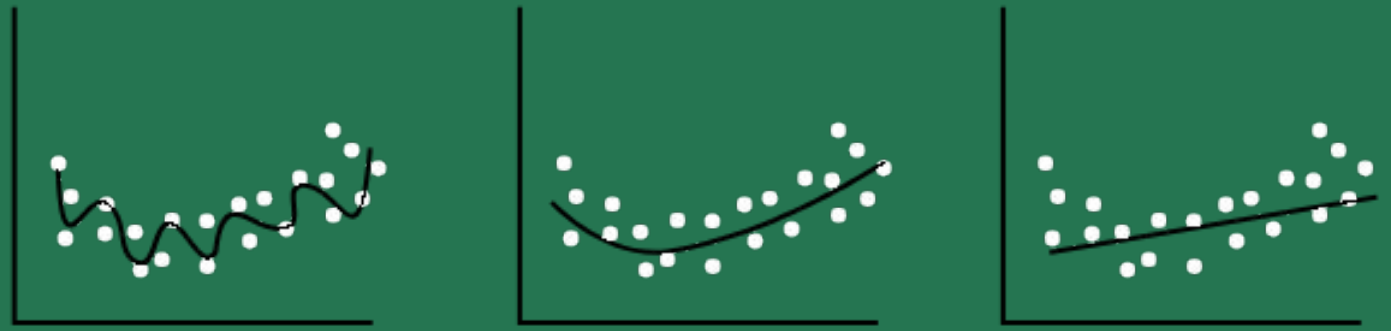


Image Credit: Overfitting

DATA COLLECTION

- ATP Tennis data: https://github.com/JeffSackmann/tennis_atp
- Extract all the matches we wanted, getting only matches that related to the US Open
- Filling in missing data
- Normalizing the data

- At the end, our final dataset contain a total of 3302 entries (matches)

Image Credit: US Open

PSEUDOCODE

- Concatenate US Open games into dataframe
  - us_open_dataframe = “tourney_name” contains “US OPEN” for dataset
- Randomize order of winner (ATP always has winners first)
  - for i in range(0 to length/2)
    - y[i] = 0
  - for i in range(length/2 to df_length)
    - y[i] = 1
  - us_open_dataframe[‘y’] = NewColumn(y)
  - if us_open_dataframe[‘y’] = 0
    - Swap(player 1 data columns, player 2 data columns)
- Shuffle Data set
  - shuffle(us_open_dataframe)

PSEUDOCODE

- Separate Categorical and Numerical data, and Drop other data
  - Categorical_cols = [player_nationality, player_hand, etc…]
  - Numerical_cols = [player1_seed, player2_age, player1_rank, etc…]
  - To_Drop = [tourney_id, tourney_date, etc…]
- Convert Categorical data to Numerical and Normalize the Data
  - dummies(us_open_dataframe, categorical columns)
  - for feature in us_open_dataframe
    - normal_feature = (current_feature - min)/(max - min)

PSEUDOCODE

- Split Data for training/test (67/33 split)
  - train, test = train_test_split(us_open_dataframe, test_size=0.33)
- Logistic Regression Model
  - Model = LogisticRegression()
  - Result = search.fit(train)

DEMO

MARKOV CHAIN RESULTS

- Accurate computation of probability of each player winning
  - Based on Player's "Winning Serve Percentage"
- Using real data, output provides 1,000 simulated tournaments
- Comparing to the actual winner of each tournament, the accuracy varied from 1% to 7%

DEMO

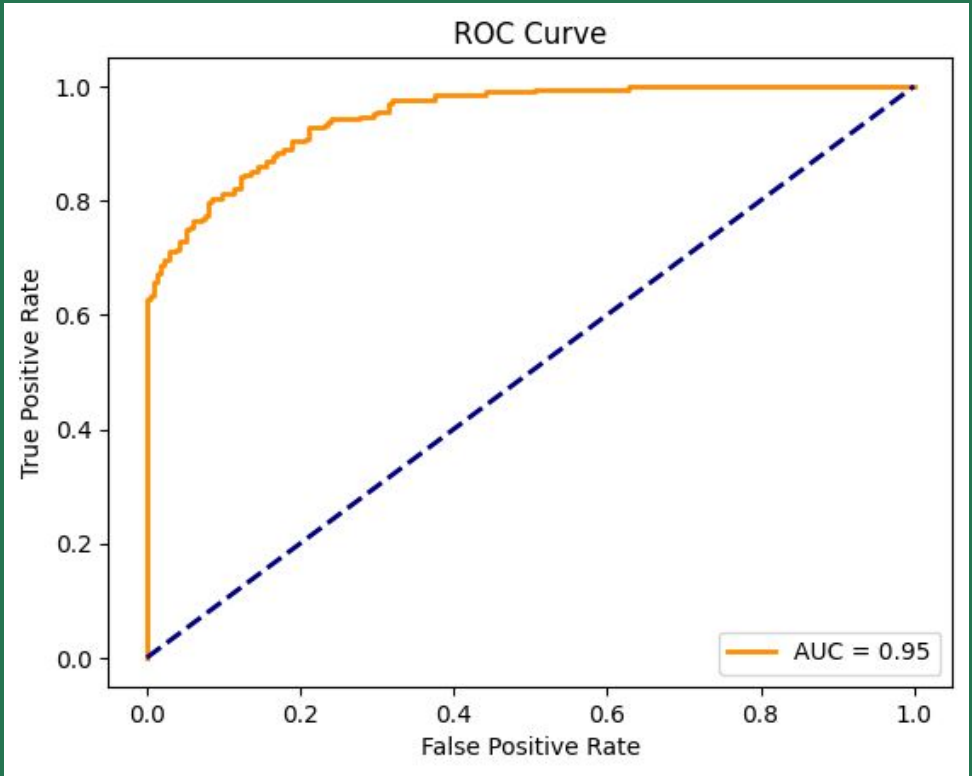
LOGISTIC REGRESSION RESULTS

- The logistic regression model achieved an accuracy of 84%
- Probability Equation:

$$P(\text{Win}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n)}}$$

- Each feature is associated with a coefficient (β) indicating its impact on the log-odds of winning; Positive coefficients increase the odds of winning, while negative coefficients decrease the odds.
- The model's interpretability is enhanced through the examination of coefficients, providing insights into the influence of each feature

INITIAL RESULTS




```
match_num
Player1_seed
Player1_ht
Player1_age
Player1_rank
Player1_rank_points
Player2_seed
Player2_ht
Player2_age
Player2_rank
Player2_rank_points
minutes
w_ace
w_df
w_svpt
```

```
w_1stIn
w_1stWon
w_2ndWon
w_SvGms
w_bpSaved
w_bpFaced
l_ace
l_df
l_svpt
l_1stIn
l_1stWon
l_2ndWon
l_SvGms
l_bpSaved
l_bpFaced
```

```
Classification Report:
              precision    recall  f1-score   support

         0.0       0.83      0.86      0.85       206
         1.0       0.86      0.83      0.84       214

    accuracy                           0.85       420
   macro avg       0.85      0.85      0.85       420
weighted avg       0.85      0.85      0.85       420
```

FEATURES USED

Categorical Features

Player1_entry
Player1_hand
Player1_ioc
Player2_entry
Player2_hand
Player2_ioc
round

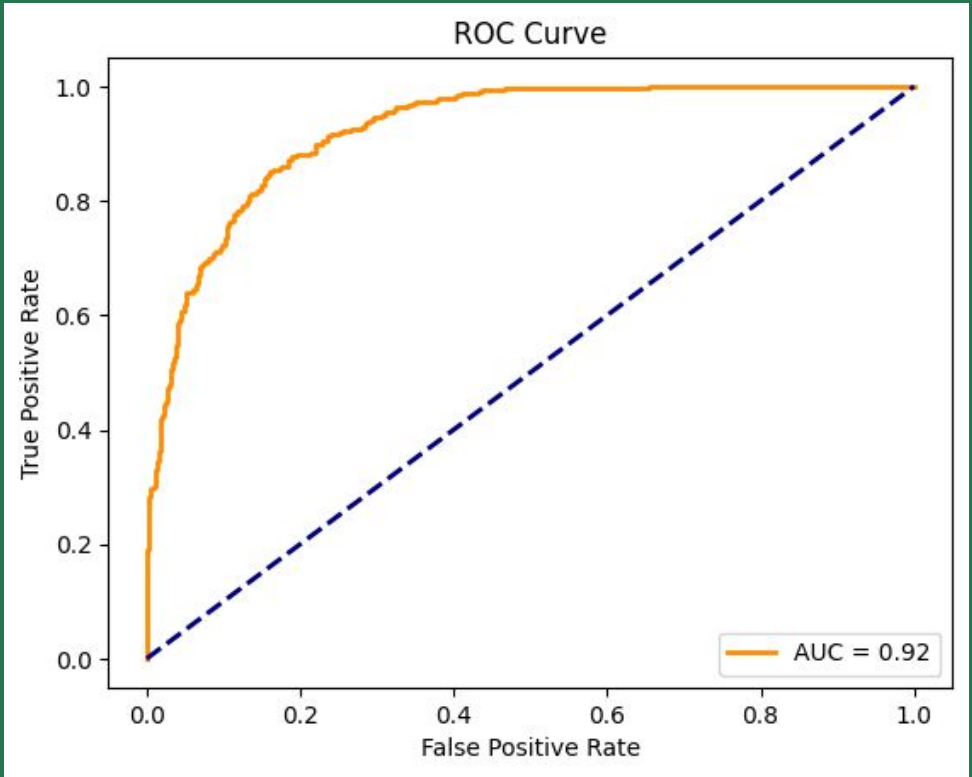
Features Dropped

tourney_id
tourney_date
score
match_num
tourney_name
surface
draw_size
tourney_level
best_of

Numerical Features

Player1_seed
Player1_ht
Player1_age
Player1_rank
Player1_rank_points
Player2_seed
Player2_ht
Player2_age
Player2_rank
Player2_rank_points
minutes
w_ace
w_df
w_svpt
w_1stIn
w_1stWon
w_2ndWon
w_SvGms
w_bpSaved
w_bpFaced
l_ace
l_df
l_svpt
l_1stIn
l_1stWon
l_2ndWon
l_SvGms
l_bpSaved
l_bpFaced

FINAL RESULTS



	precision	recall	f1-score	support
0	0.86	0.81	0.84	550
1	0.82	0.87	0.84	540
accuracy			0.84	1090
macro avg	0.84	0.84	0.84	1090
weighted avg	0.84	0.84	0.84	1090

FEATURE IMPORTANCE

- Important Features
  - Player1_rank_points: 20.0022
  - Player2_rank_points: 18.7944
  - SvGms: 7.2779
- Negative Features
  - 1stWon: -10.1974
  - svpt: -6.7567
  - 2ndWon: -3.4734
- Neutral Features
  - W_ace: 1.4265
  - W_df: 1.0921

MARKOV VS LOG REG

- Accuracy
  - 1-7% vs 84%
- Required data
  - One type of data vs Multiple variables
- Complexity
  - Simplification vs Detail

SIGNIFICANCE OF RESULTS

- Markov chain models can be more useful for real-time predictions
- Logistic regression model is better at predicting the winner of a match, so it would be more useful for betting, for example
- For our original objective, logistic regression model works

```
Match Setup:
Player A Service Point Winning Percentage: 0.7
Player B Service Point Winning Percentage: 0.6
Current Set Score: 1-0
Current Game Score: 0-3
Probability of each player winning a match:
     0    1    2    3    4    5    6    7         8         9
0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.829491  0.170509
```

APPLICATIONS / FUTURE WORK

- Game Outcome Prediction for other sports
  - Predicting match outcomes in various sports such as golf, basketball, soccer, and baseball, using historical data, player statistics, team performance
- Betting
  - Creating a more precise model could be tested to see if it can beat the odds
- Player Strategy and Training
  - Use importance of features to improve players performance through training

OUR FUTURE WORK IN THE FIELD

01

Alex Kwok

- Job offer at Startup
- Possible continuation of model in the future

02

Vittorio Luzio

- Master's in Data Science

03

Alex Medina

- PhD in Astrophysics
- Sports

04

Eric Shin

- Master's in Data Science
- Continue loving sports

CITATIONS - WORK CITED

- CSC 411: Lecture 4 - Logistic regression - Ethan Fetaya, James Lucas and Emad Andrews
- GraphPad Prism 10 Curve Fitting Guide - The goal of simple logistic regression
- Markov4Tennis from "Seb943"
- ITF RULES OF TENNIS
- Tennis Sim from Mark Jamison
- SIMULATING TENNIS MATCHES USING MARKOV MODEL MATH 350 YUVAL CALEV NABIL SALEM DECEMBER 19, 2012
- Modeling a tennis match with Markov Chains | by Sébastien Cararo | Analytics Vidhya | Medium
- Logistic regression - Wikipedia

ANY QUESTIONS?

THANKS FOR LISTENING!