

# Predicting the outcomes of amateur pool matches in North America

## 1 Introduction

Pocket billiards, in particular the games of 8-ball, 9-ball, and snooker, is a hugely popular recreational activity in the Americas, Europe, and the Far East. It is estimated that 46 million people play pool in North America alone. Particularly in Europe and North America, a growing number of leagues cater to the casual player, with handicap systems in place to encourage participation of players of all ability, from beginner level to highly experienced.

A typical handicapped pool league might consist of approximately 10 teams each containing 8 signed-up players, of which 5 play on a given game day. Each player's ability is assessed by a league operator, or prior knowledge, and assigned a quantitative value, referred to as the player's skill level. The combined skill level of each team is generally capped in order to ensure that each team has an even distribution of casual and experienced players, therefore ensuring a fair and balanced competition. To further even the odds, the number of pool games required to win an individual matchup might be skewed proportional to the skill level difference between the two competing players. For example, Player A, with skill level 5, may be required to win 4 games to win a match against player B, who has skill level 3, and only needs to win 2 games to win the match. As players play more games and acquire more experience, their skill level is adjusted according to a pre-determined algorithm, such as the ELO Rating System widely used in chess and other zero-sum outcome games.

The desired result is that every participant, regardless of ability, should average out to a 50/50 win/loss ratio, provided they play sufficient games to achieve an accurate skill rating. However, as an avid pool player myself, and a participant of a handicapped league, I often find myself discussing if extremely lopsided races truly have an even probability of each player winning, since luck and external conditions can affect the performance of even the best players. Is there a metric which has a larger influence on results than skill level? Could this be exploited by team captains in order to pick favourable 1v1 matchups, and boost their teams's chances of a win?

## 2 Data


### 2.1 Source

Although multiple billiards associations keep detailed records of player and match statistics, access to data is often restricted to subscribing members. An exception is the North American Poolshooters Association (NAPA), which has no annual subscription fee, and publishes comprehensive statistical data on its website, [www.napaleagues.com](http://www.napaleagues.com). A homepage exists for each individual league (also referred to as division), and each registered player. From these homepages, further statistics can be accessed via sub-menus, including full match results, player skill levels, and bonus statistics such as number of break-and-runs, etc. A typical player homepage is shown in Figure 1, and a typical division homepage is shown in Figure 2.

Records exist for every player and division that has existed since the leagues conception in 2010. This is a total of >60,000 player profiles, and >8000 division profiles. If each division consisted of 8 teams of 5 players, and ran for a session of 10 weeks, then this would yield a total of 1.6 million individual matchups. In reality, most divisions run for longer than 10 weeks, so this is probably a conservative estimate.

A typical match summary page is shown in Figure 3. Here, the division format is 5-man teams, which means that each match consists of 5 individual player 'matchups' (only two are shown in Figure 3). Each matchup has details of the race, number of game wins, number of break-and-runs (BO), number of 8-on-the-breaks (SNAP), and

Player Stats



**Callum Odonnell**

<b>Shooter's ID:</b>	10061781
<b>Gender:</b>	Male
<b>League:</b>	ATX NAPA
<b>Member Since:</b>	May 24, 2019
<b>Home Base:</b>	Travis, Texas
<b>Last Match:</b>	Jan 26, 2020
<b>Current NRP:</b>	27 Level Green Since Jan 29, 2019
<b>Active Divisions:</b>	8220

**LIFETIME STATS**

<b>Matches Played:</b>	27
<b>Match Record:</b>	23 wins 4 losses
<b>Overall Win %:</b>	85%
<b>AvgPPM:</b>	14.74

[SKILL LEVELS](#) | [BONUS PLAY](#) | [TOUR ELIGIBILITY](#) | [H2H](#) | [GENDER](#) | [SEASONS](#) | [MONTHLY](#) | [YEARLY](#) | [VENUES](#) | [RIVALS](#) | [RACE BIN](#)  
[8-BALL](#) | [9-BALL](#) | [10-BALL](#) | [FAST 8](#)

Figure 1: Typical player homepage.

## League Standings

OVERVIEW

Division ID:	8025
Division Name:	Thursday "The Grand Pool Bah" MOD-330 8-ball League
Location:	Austin, Texas
Completed Weeks:	12 of 16
Number of Teams:	11
Format:	5-man Teams
H2H Bonus:	1 points
Match Time:	7:00 PM
110 Rule:	Yes
Flex Point Standings:	Average Points Per Match Excluding Forfeits
Forfeit Savior:	Yes
Session Schedule:	<a href="#">View</a>
Weekly Scratch:	<a href="#">View</a>
Weekly Scores:	<a href="#">View</a> ▾

STANDINGS | [FLEX POINT](#) | [W-L-T](#) | [ACH](#) | [F HUSTLERS](#)

STANDINGS

#	TEAM	PTS	AVG PPM
1	<a href="#">Hi Pumpkin</a> ▾	676	61.5
2	<a href="#">Vape Lordz</a> ▾	594	54
3	<a href="#">Cory And The Revolution</a> ▾	590	53.6
4	<a href="#">I'm The Captain Now</a> ▾	581	58.1
5	<a href="#">Keep Your Face Tight</a> ▾	568	51.6

W1: Oct. 10, 2019  
W2: Oct. 17, 2019  
W3: Oct. 24, 2019  
W4: Nov. 07, 2019  
W5: Nov. 14, 2019  
W6: Nov. 21, 2019  
W7: Dec. 05, 2019  
W8: Dec. 12, 2019  
W9: Dec. 19, 2019  
W10: Jan. 09, 2020  
W11: Jan. 16, 2020  
W12: Jan. 23, 2020

Figure 2: Typical division homepage.

## Score Sheet

Nothing Goes Here		
Thursday Oct. 10, 2019		
8-ball	<b>Clayton Bullock</b> (Nothing Goes Here)	<b>Mikey Carillo</b> (Team Fun)
RACE	5	3
# WINS	5	1
BO	-	-
SNAP	-	-
RL	-	-
<b>SCORE</b>	<b>14</b>	<b>3</b>
	Report Incorrect Score	

8-ball	<b>Andy Erdmann</b> (Team Fun)	<b>Krush Gandhi</b> (Nothing Goes Here)
RACE	6	5
# WINS	6	3
BO	-	-
SNAP	-	-
RL	-	-
<b>SCORE</b>	<b>14</b>	<b>3</b>
	Report Incorrect Score	

Figure 3: Typical match summary page.

finally a field to show if the match was rackless or not (RL). See the appendix for an explanation of these bonus fields and the scoring system. Note that in divisions where the format is a singles league, each team consists only of a single player.

## 2.2 Web scraping

### 2.2.1 Player information

The URL for a player homepage is conveniently given by [www.napaleagues.com/stats.php?playerSelected=Y&playerID=xxxxxxxx](http://www.napaleagues.com/stats.php?playerSelected=Y&playerID=xxxxxxxx), where the final 8 characters are the player's unique ID. Similarly, the URL for a division homepage is <https://www.napaleagues.com/states.php?did=xxxx>, where the final 4 characters are the unique division ID. This is highly convenient when running a web scraping script, as each webpage can be opened in turn by looping through the respective IDs. Firstly, the basic information was collected from the homepage of each player using the script in `get_player_info.ipynb`, and exported to a csv file. The relevant

	Name	ID	Gender	League	Join Date	County	State	Last Game	Active Divisions	Total Matches	Won	Lost	AvgPPM	Win %
5	Jason Fayman	10062005	Male	NAPA of Montgomery	2019-06-09 00:00:00	Montgomery	Alabama	2019-07-01 00:00:00	[]	1	0	1	7	0
6	Ricky Ledbetter	10062006	Male	NAPA of Montgomery	2019-06-09 00:00:00	Montgomery	Alabama	2019-08-26 00:00:00	[]	9	7	2	12.22	77.7778
7	Greg Turner	10062007	Male	NAPA of Corpus Christi	2019-06-09 00:00:00	Aransas	Texas	2019-12-04 00:00:00	[7689, 8002, 8030]	21	6	15	6.24	28.5714
8	Jerome Lawrence	10062008	Male	NAPA of Franklin	2019-06-09 00:00:00	Franklin	Missouri	2019-08-26 00:00:00	[]	8	2	6	6.75	25
9	Gary Tang	10062009	Male	NAPA of San Diego	2019-06-09 00:00:00	San Diego	California	2019-09-09 00:00:00	[]	8	1	7	5.88	12.5

Figure 4: Dataframe of player details, scraped from each player homepage.

ID	8 Games	8 Skill	9 Games	9 Skill	10 Games	10 Skill	F8 Games	F8 Skill
10000006	38	78	35	79	TBD	TBD	0	TBD
10000007	53	34	44	21	TBD	TBD	0	TBD
10000008	9	42	0	TBD	TBD	TBD	0	TBD
10000010	9	44	0	TBD	TBD	TBD	0	TBD
10000011	6	56	0	TBD	TBD	TBD	0	TBD

Figure 5: Player skills dataframe.

information is contained within the 'td' html tags, where in most cases the webpage table format remains the same, enabling the data indices to be hard-coded. Notable exceptions are for VIP players, who have an additional title headings, and for empty records (IDs which have no associated player). Due to the abundance of players, the code was modified to return empty values for these pages. An example dataframe returned by `get_player_info.ipynb` is shown in Figure 4.

Unfortunately, player skill level, and number of games played of each discipline (8 ball/9 ball/10 ball/Fast 8), are crucial details which are missing from the homepage, so the script in `get_player_skills.ipynb` was used to read data from the "Skill Levels" tab. This returns the dataframe shown in Figure 5. It can be seen that the 9 ball, 10 ball, and Fast 8 columns are often empty, as 8 ball is the most popular billiard discipline in the USA and Canada among amateurs. Therefore, for the remainder of this study, only statistics referring to 8 ball matches were considered. For the remainder of this study, all results and analysis are based on 8 ball matches, unless otherwise specified. The skills dataframe was merged (on player ID) with the dataframe in Figure 3 to create the `players_with_skills.csv` file.

### 2.2.2 Match information

Collecting match result data is a slightly more complicated process, due to the required URL consisting of 3 elements: Division ID, Team ID, and the match date. Fortunately, all the required information can be found on the respective division homepage. The division ID is a 4 digit integer which runs from 0001 to ~8100 as of Dec 2019. Team IDs can be found in the team hyperlinks in the results table near the bottom of the division homepage. Finally, all match dates are contained within a drop down menu next to "Weekly Scores" (see Figure 2).

Firstly, two separate csv files were created for each division: one containing information about the division such as Name, Location, Length in weeks, Format, etc., and the second containing information from the results table, including Team Name, Team ID, and points scored. The web scrape was performed using the code in cell 5 of `get_match_data.ipynb`. The two resultant dataframes are shown in Figures 6 and 7.

	Division Name	Division ID	Location	Weeks	Num Competitors	Format	H2H Bonus	110 Rule	Forfeit Saviour	Dates
5	Tuesday "Hampden County" Standard Limit 8-ball...	3145	Hampden County, Massachusetts	14	7	5-man Teams	0 points	Yes	No	[2015-01-20 00:00:00, 2015-01-27 00:00:00, 201...
6	Thursday "RAB" Lagger's Choice League	3146	Conway, Arkansas	20	6	5-man Teams	0 points	Yes	No	[2015-01-22 00:00:00, 2015-01-29 00:00:00, 201...
7	Tuesday Standard Limit 8-ball League	3147	LaSalle, Illinois	15	8	4-man Teams	0 points	Yes	No	[2015-01-20 00:00:00, 2015-01-27 00:00:00, 201...
8	Monday No Limit 8-ball League	3148	Folsom, Louisiana	15	5	3-man Teams	0 points	Yes	No	[2015-01-19 00:00:00, 2015-01-26 00:00:00, 201...
9	Monday "Bankshots In-House" Lagger's Choice Le...	3149	Westerville, Ohio	15	5	4-man Teams	0 points	Yes	No	[2015-01-05 00:00:00, 2015-01-12 00:00:00, 201...

Figure 6: Division details dataframe.

	Bye Points	Div ID	Match Points	Team	Team ID
13	0.0	4738.0	108.0	Devon Moreno	32101
14	0.0	4738.0	62.0	Jose Hernandez	32105
15	0.0	4738.0	60.0	Amanda Tebay	32103
16	0.0	4738.0	46.0	Larry Milam jr	32108
17	0.0	4738.0	44.0	Jake Oro	32114

Figure 7: Team table dataframe.

Once the division and constituent team information was collected, match results were scraped from a subset of divisions using the code in cell 7 of `get_match_data.ipynb`. Care was taken to only select divisions where 8-ball is the sole discipline that is played. These results were then exported to a csv file, where each row represents a single matchup. A typical output dataframe is shown in Figure 8.

## 2.3 Player data

Firstly, I looked at the structure of the player data in the `players_with_skills.csv` file, with boxplots of the Number of 8 Ball Games, 8 Ball Skill Level, Win % and Average Points-Per-Match (AvgPPM) columns plotted in Figure 9. Summary statistics are shown in Table 1. Data for a total of 56,559 players was collected. The number of games played by each player is highly skewed, with a majority of players (57.4%) playing less than 20 games total, compared to a median of 9 and a mean of 22.3. This implies that the churn rate is relatively high, and a lot of casual players move on after just one session (typically 10-15 matches).

The skill level is assigned to a player upon joining a league, and is set by default to 50 for a male, and 40 for a female. Exceptions to the starting skill level are made if the player is personally known to the league operator,

	8 Games	8 Skill	Win %	Avg PPM
Maximum	576	167	100	23.0
Mean	22.3	53.2	44.4	8.58
Median	9	51	48.5	9.02
Std. Dev.	36.6	24.1	23.1	3.24

Table 1: Summary of player statistics from the `players_with_skills.csv` file.

	A BO	A ID	A Name	A RL	A Race	A Score	A Snap	A Team	A Wins	B BO	...	B Name	B RL	B Race	B Score	B Snap	B Team	B Wins	Date	Discipline	Division
225	-	10021704	Jennifer Parker	-	3.0	1.0	-	Alter Egos	0.0	-	...	Joanna Caceres	-	3.0	20.0	-	We're Solids Right	3.0	2016-04-08	8-ball	4613
226	-	10045424	Jesus Guevara	-	3.0	1.0	-	We're Solids Right	0.0	-	...	Matt Walsh	-	3.0	20.0	-	Alter Egos	3.0	2016-04-08	8-ball	4613
227	-	10020617	Scott Saucier	-	8.0	14.0	-	Alter Egos	8.0	-	...	Dagoberto Perez	-	3.0	3.0	-	We're Solids Right	2.0	2016-04-08	8-ball	4613
228	-	10021703	Matt Parker	-	4.0	15.0	1	Alter Egos	4.0	-	...	Angel Castillo	-	4.0	3.0	-	We're Solids Right	1.0	2016-04-08	8-ball	4613
229	-	10020621	Stephen Codina	-	3.0	3.0	-	Alter Egos	2.0	-	...	Bertoni Mencia	-	5.0	14.0	-	We're Solids Right	5.0	2016-04-08	8-ball	4613

Figure 8: Match results dataframe.

or has played NAPA pool in the past. It is interesting to see that the skill level follows an approximate normal distribution centered close to 50. The distribution is slightly skewed towards larger values, likely influenced by a handful of highly dedicated and talented players.

Win % and AvgPPM are both symmetrically distributed, with the few outliers being players who have played very few matches. These two features are expected to be very closely related, although the amount of points awarded for a win can vary significantly under NAPA rules (see appendix for full details of points scoring system).

It is of interest to study how these parameters vary over time, as players log more games in the system. Skill level is a dynamic value, which self-adjusts after each result according to a proprietary algorithm. The magnitude of change is influenced by factors including opponent skill level, margin of victory, and bonus achievements (e.g. 8-on-the-break, break and run). The skill levels of a random selection of 100 players were tracked over a period of 100 games and the rolling standard deviation and rate of change plotted in Figure 10. The black lines represent the rolling averages for each individual player after a given number of games. It can be seen that after an initial period of rapid change, both plots flatten out and are stable after a period of about 20 games. This is therefore a safe estimate for the average number of games required for a player to reach their true skill level.

If a player is initially underrated, as their skill level increases, the number of games required to win a match will also increase, leading to progressively more difficult matches. Equilibrium is reached when the player is winning on average 50% of their games. This can be visualized by plotting number of games versus match win %, as displayed in Figure 11. The effectiveness of the algorithm can be seen in the rapid convergence to a 50% win percentage.

Having seen that player statistics can fluctuate wildly during the first few matches, the question arises of whether results involving new players can be predicted more or less accurately than those of experienced players. Do any obvious patterns arise when match data is analyzed?

## 2.4 Match data

Before looking at patterns in the results data, player statistics (skill level, number of games played, win %) were added by use of a join between the dataframe in Figure 8 and the players\_with\_skills.csv file. This was then reduced to single columns describing the mismatch between the two competing players, which is the stat for player B subtracted from the respective stat for player A. For this study, in-game bonus statistics were not included, as this would have necessitated scraping an additional webpage per player to collect their aggregate bonus stats. The new results dataframe is shown in Figure 12.

The distribution of each of these columns is visualized in Figure 12. In total, 20,024 individual results were

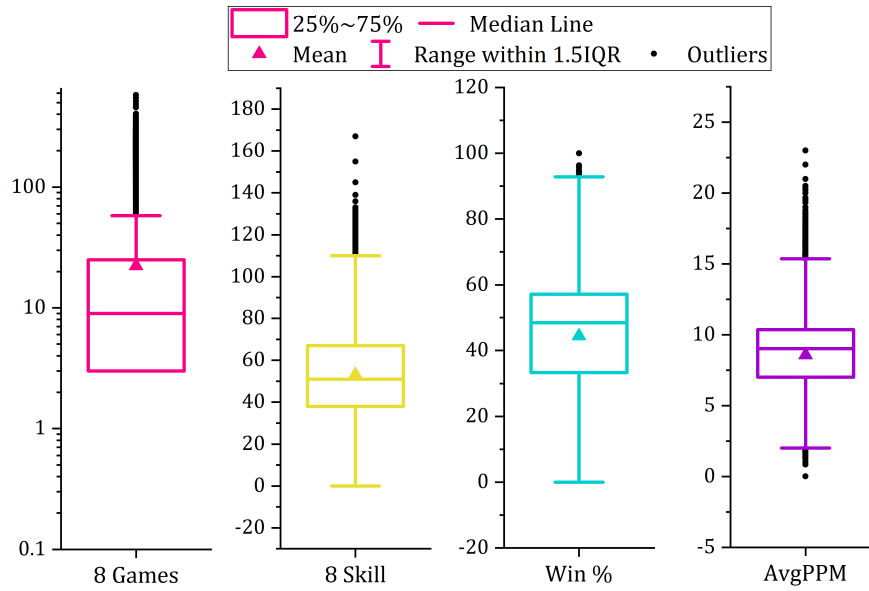


Figure 9: Boxplots of Number of 8 Ball Games, 8 Ball Skill Level, Win %, and AvgPPM, taken from players\_with\_skills.csv.

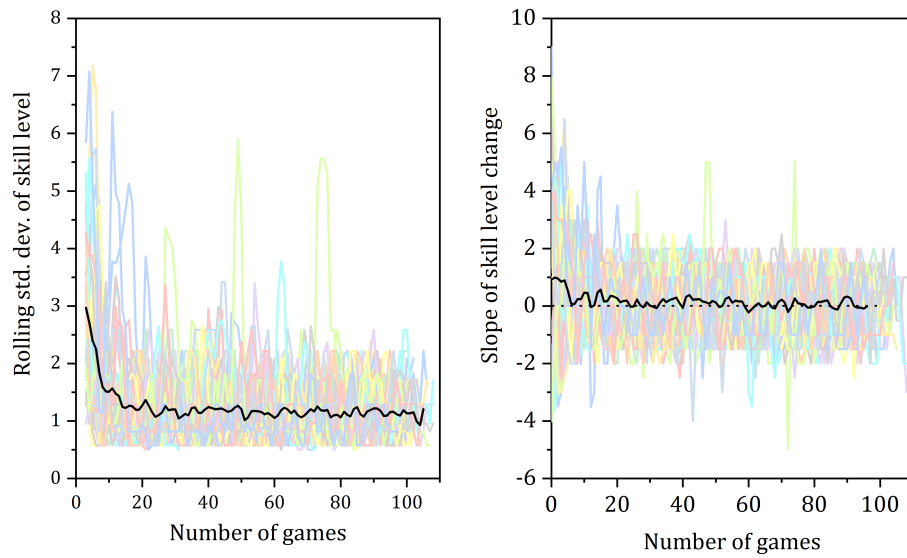


Figure 10: Standard deviation and rate of change of skill level of 100 players over 100 games.



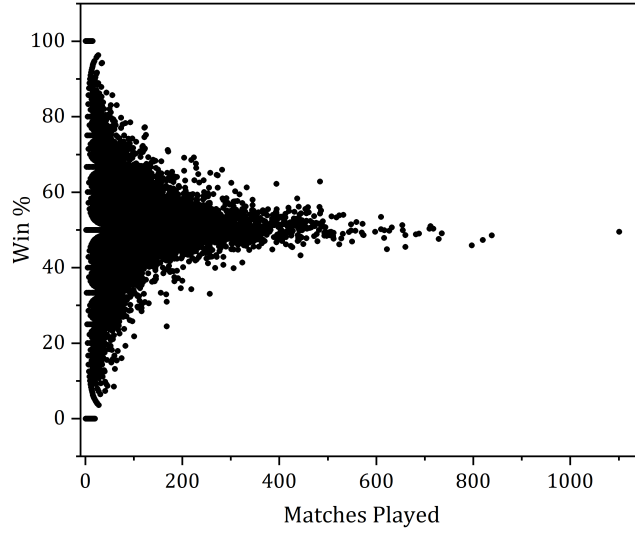


Figure 11: Number of matches played versus match win %.

	A ID	B ID	Race Margin	Win % Margin	Skill Margin	Game Margin	AvgPPM Margin	Win Margin
0	10021704	10015795	0.0	5.769231	-3	-64	-0.02	-3.0
1	10021704	10015795	-1.0	5.769231	-3	-64	-0.02	2.0
2	10020621	10015795	0.0	-1.388889	6	-69	-0.73	4.0
3	10020621	10015795	0.0	-1.388889	6	-69	-0.73	-1.0
4	10020616	10015795	0.0	2.040816	1	51	-0.42	-2.0

Figure 12: Results dataframe, with added player statistics and expressed in terms of their mismatches

logged. With the exception of the winning margin, which for obvious reasons cannot be zero, each feature follows a symmetric single-peaked distribution.

### 3 Predicting match outcomes

#### 3.1 Correlations

When determining the factors influencing the likelihood of a player winning a match, a good place to start is to look at the effectiveness of the handicapping system. One way to measure this is to record the win % rate of a higher skilled player for each possible race length (the number of games required for each player to win the match). Possible races ranges from 2-2 for two lower skilled players, to 6-6 for two higher skilled players. Where a lower skill meets a hgher skill, the race may be as lopsided as 10-2 (higher skill requires 10). Figure 14 shows the winning % for the higher skilled player in each race bracket. Also plotted on the right axis is the number of results in that particular race bin. The success rate for the higher skilled player is 59% on average, suggesting that it not quite enough to truly balance the odds, but still offers a strong equalizing force. Interestingly, the highest win % is for the 10-2 race, despite it being the most unbalanced scenario. However, this is likely due to most matches in this bin being won by few truly exceptional players for whom the handicap is not sufficient to overcome their talent.

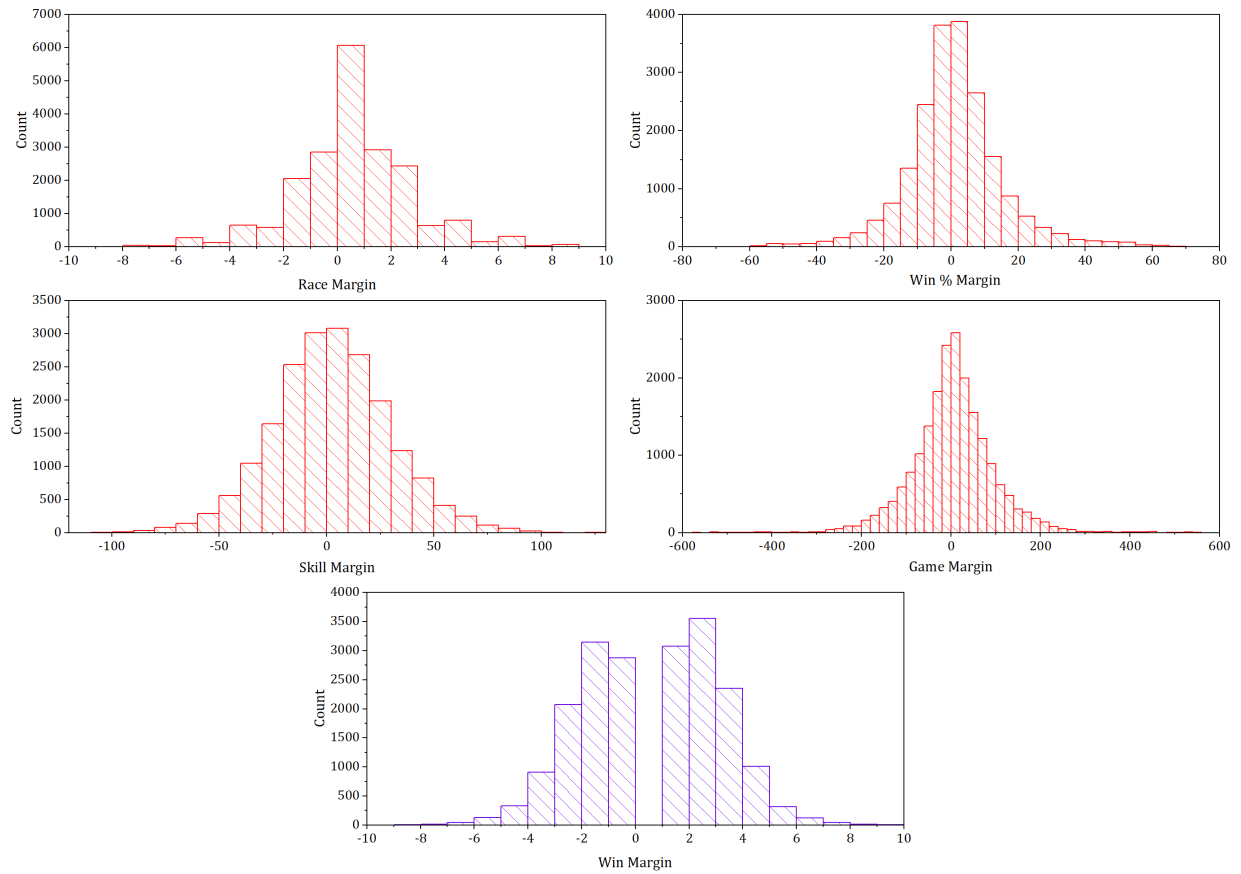


Figure 13: Histograms of each feature in the results dataframe.

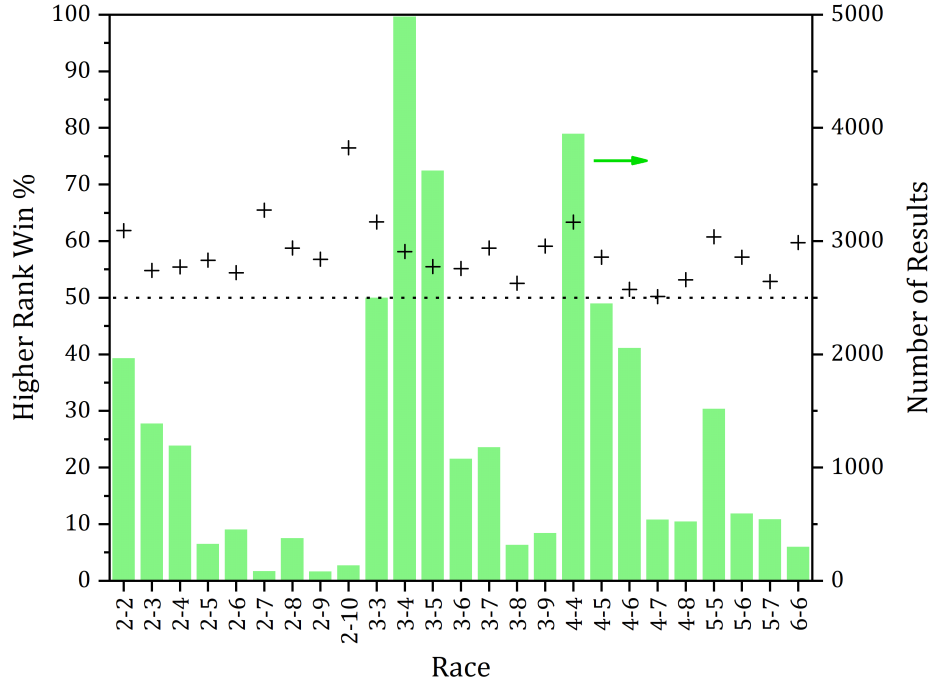


Figure 14: Left axis: winning % for the higher skilled player for each possible race length, right axis: number of results with that race.

Furthermore, 10-2 races only constitute a tiny fraction ( $\sim 1\%$ ) of the total number of matches, so although these results appear to be more predictable, it is not a very helpful result for building a general model.

In section 2.3 it was seen that players have volatile skill levels over the course of their first few games, on average taking about 20 games to reach a stable value. Since players are often dramatically underrated in this period, it can be hypothesised that more 'upsets' (low rank beating a high rank) will occur between in players. To explore this, match outcomes were tallied for four types of player matchup: Inexperienced v inexperienced, experienced v inexperienced, and experienced v experienced. The second type has two categories: where the experienced player is higher ranked, and vice versa. The results are shown in Figure 15. It is immediately obvious that the vast majority of games are played between players in the 'experienced' category, which has the lowest average success rate for the higher skilled player (close to 56%). Additionally the hypothesis of more upsets occurring between inexperienced players proves to be incorrect. In summary, while interesting, this figure also tells us little about how to predict the outcome of the majority of matches.

In Table 2, we can see the winning rate for the player who has the largest value of each feature in turn. It turns out that the best indicator is the player historical win %, where the player with the higher win % prevails 61.7% of the time. A full picture of the correlations between each feature is given by the Pearson's  $r$  correlation table in Figure 16. It can be seen that all features have relatively weak correlations with the Win Margin column, which is perhaps the best evidence so far that the handicapping system does a pretty good job of evening the winning odds.

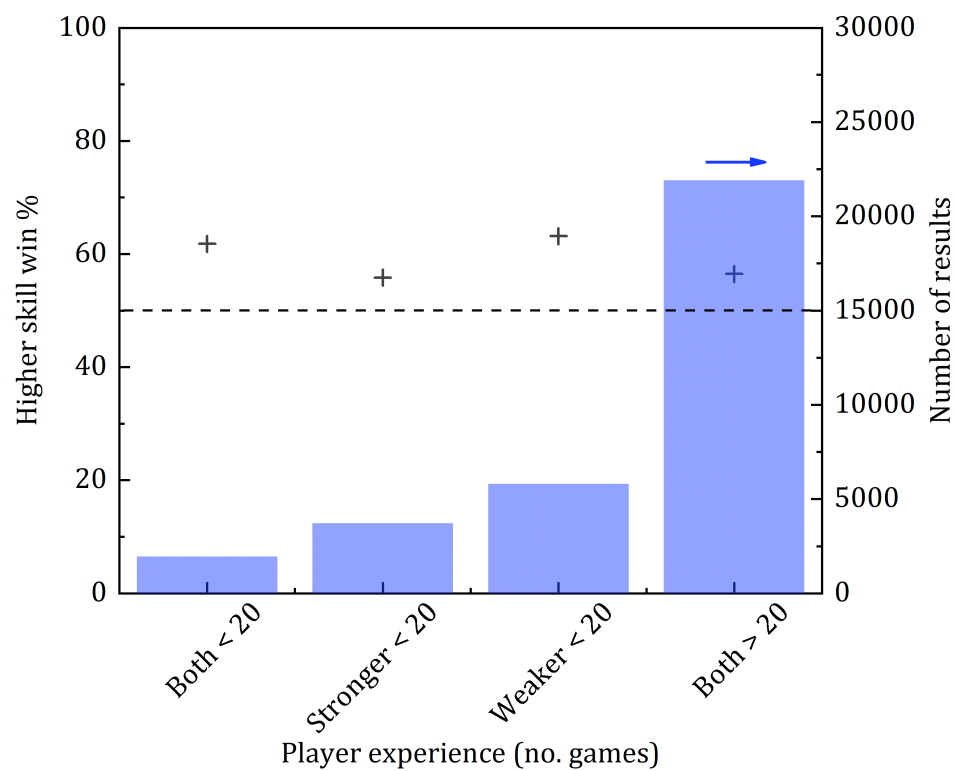


Figure 15: Left axis: winning % for the higher skilled player for each type of matchup, right axis: number of matches in each category.

Player with:	Longer Race	Higher Win %	Higher Skill	Higher Avg PPM	More Games
% Wins	56.7	61.7	59.4	60.6	52.2

Table 2: Player winning percent for each criterion.

	Race Margin	Win % Margin	Skill Margin	Game Margin	AvgPPM Margin	Win Margin
Race Margin	1	0.321	0.811	0.136	0.347	-0.12
Win % Margin	0.321	1	0.522	0.079	0.953	0.223
Skill Margin	0.811	0.522	1	0.187	0.531	0.045
Game Margin	0.136	0.079	0.186	1	0.104	-0.003
AvgPPM Margin	0.347	0.953	0.531	0.104	1	0.211
Win Margin	-0.12	0.223	0.045	-0.003	0.211	1

Figure 16: Pearson’s r coefficient correlation table.

However, it is promising that a modest value of 0.223 is observed for the Win % Margin, which hopefully means that some matchups can be reliably predicted. Note that although the AvgPPM column also displays a similar coefficient value (0.211), the inter-correlation between the two features is extremely high (0.953), indicating that the two features are almost completely dependent.

The correlations of the Win Margin with the Win % Margin, Skill Margin, Game Margin, and AvgPPM Margin are visualized in Figure 17. The corresponding regression lines are shown in red.

## 3.2 Machine Learning Models

### 3.2.1 Logistic Regression

Identifying the outcome of each match can be treated as a binary classification problem, with the possible outcomes being a win or a loss. To simplify the results data, a new column was added to the results dataframe with the heading 'Result,' where a value of 1 was assigned for a positive Win Margin (a win for player A), and 0 assigned for a negative Win Margin (a win for player B). A good starting model for this problem is a simple logistic regression model with a sigmoid activation function. The hypothesis takes the form

$$h_{\theta}(x) = g(\theta_0 + \theta_{RM}x_{RM} + \theta_{WPM}x_{WPM} + \theta_{SM}x_{SM} + \theta_{GM}x_{GM} + \theta_{PPM}x_{PPM}) = \Phi^T \mathbf{X} \quad (1)$$

where  $\Phi^T$  is the transposed vector of weights, and  $\mathbf{X}$  is the matrix of data from the match results dataframe, with an additional column of ones inserted to the left, referred to as bias units.  $\theta_0$  is the bias weight, and the subscripts RM, WPM, SM, GM, and PPM refer to Race Margin, Win % Margin, Skill Margin, Game Margin, and AvgPPM Margin, respectively. The sigmoid activation function is written as

$$g(z) = \frac{1}{1 + e^{-z}} \quad (2)$$

The optimal value of  $\Phi$  is calculated by minimizing the cost function, given by

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \left[ y^{(i)} \ln(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \ln(1 - h_{\theta}(x^{(i)})) \right] \quad (3)$$

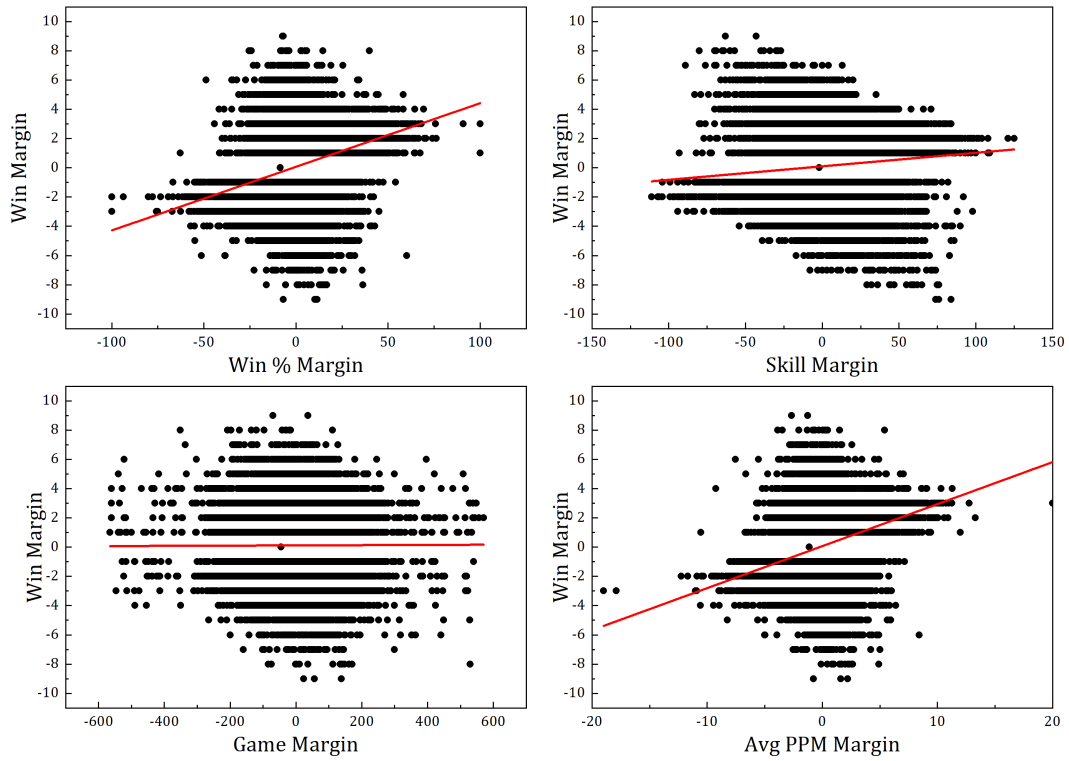


Figure 17: Correlations of the Win % Margin, Skill Margin, Game Margin, and AvgPPM Margin with the Win Margin for all results in the results dataframe.

$\theta_0$	$\theta_{RM}$	$\theta_{WPM}$	$\theta_{SM}$	$\theta_{GM}$	$\theta_{PPM}$
0.045	-0.031	0.047	0.031	-0.0019	-0.067

Table 3: Optimized hypothesis parameters.

Parameter	Value
<b>Accuracy</b>	63.6%
<b>Precision</b>	0.648
<b>Recall</b>	0.676
<b>F1</b>	0.662

Table 4: Accuracy, precision, recall, F1 score for logistic regression.

where  $m$  is the number of results,  $y^{(i)}$  is the  $i$ th value in the Result column, and  $x^{(i)}$  is the  $i$ th row in the feature matrix. The data is randomly split into training and validation datasets in a ratio 80:20 using scikit-learn’s build-in `train_test_split` function. Then, by using the code in **logistic\_regression\_model.ipynb**, the cost function  $J(\theta)$  is minimized with a Constrained Truncated Newton method (TNC) optimization algorithm, which yields the hypothesis parameters shown in Table 3.

The optimized hypothesis parameters in table 3 are then applied to the validation data, which enables the effectiveness of the model to be quantified. Useful metrics are the accuracy (% of results correctly predicted), the precision (true positives/(true positives + false positives)), recall (true positives/(true positives + false negatives)), and F1 value ((2\*precision\*recall)/(precision + recall)).

Note that the training accuracy was 64%, very close to the validation accuracy. This suggests that the data is biased, and requires a higher number of features to improve the model.

One method to introduce new features without recording more data is to account for interaction between the features. This could be included in the model by adding polynomial features, or by exploring a neural network model as described in the following section.

### 3.2.2 Neural Network

Briefly, an artificial neural network provides a method with which to include interaction between various features in a dataset by means of layers of interconnected ‘neurons.’ The number of input neurons is equal to the number of training examples, and the number of output neurons is equal to the number of ‘classes’ (e.g. when classifying handwritten numbers from 1 to 10, the number of classes is 10). Propagation between the layers is achieved by matrix multiplication, where the propagation matrices contain the weights for each neuron in the form of  $\Phi_{ij}$ . An arbitrary number of ‘hidden’ layers can exist between the input and output layers, with more layers enabling more complex interactions to be modelled. However, including several layers leads to longer computation times and a higher risk of overfitting the data.

For this study, the code in **neural\_network\_model.ipynb** was used to simulate a neural network with a single hidden layer. The cost function was minimized using a backpropagation algorithm and a TNC optimization function. In order to determine the optimal number of neurons for the hidden layer, the accuracy was measured as a function of increasing number of neurons. For each data point, the result was averaged over three measurements to account for randomness. The results are plotted in Figure 17. Although not a great deal of variation was observed, the highest value of 64.9% was measured for 40 neurons. This is an improvement of 2.3% over the logistic regression model. Furthermore, the precision, recall, and F1 values are all higher than their counterparts for logistic regression.

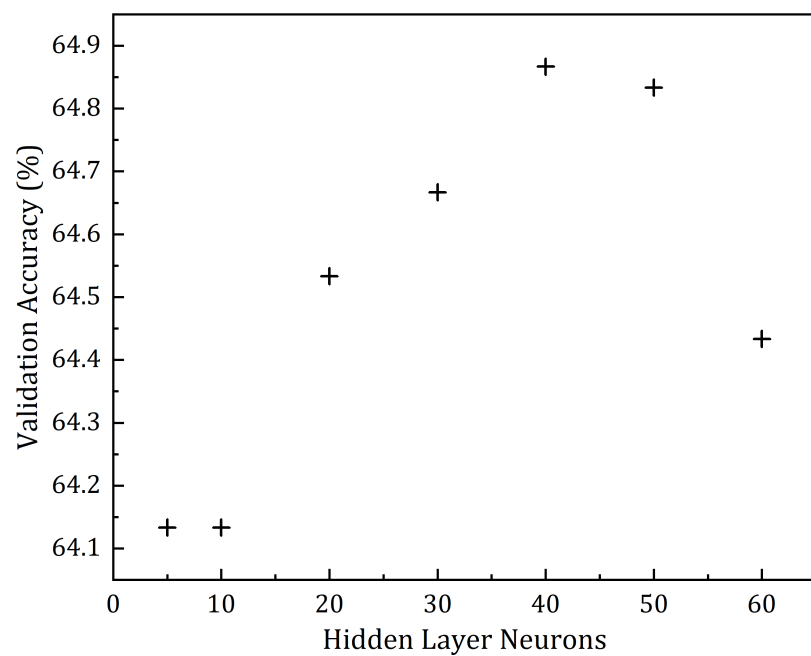


Figure 18: Neural network result prediction accuracy for different sized hidden layers.

Parameter	Value
Accuracy	64.9%
Precision	0.664
Recall	0.684
F1	0.673

Table 5: Accuracy, precision, recall, F1 score for the neural network.



BEST PLAYER MATCHUP FOR TEAM A			
	Player A	Player B	% Win Prob
0	Brugamyer	Carillo	61.6
1	Jones	Bilberry	86.5
2	Andres	Erdmann	13.5
3	Czys	Warrington	75.0
4	Myers	Miller	50.0
-----			
Match win probability for team A: 57.3%			
Average match win probability for team A: 50.6%			

Figure 19: Optimal lineup for team A, calculated using the neural network model.

### 3.3 Application to real match scenarios

While as an individual player it might be interesting to know the approximate odds of winning your matchup, there's not really that much benefit to be gained from it, as you can't change your opponent. On the other hand, team captains would find it extremely useful to be able to choose the individual player matchups which yield the best overall chance of winning the whole match. In a team division, typically one team is designated the 'home team' and the opponent designated the 'away team.' The away team writes down their player list first, enabling the home team to decide who is matched up with who. Experienced team captains will have done their research, and have a general idea of which players on the opposing team are vulnerable or on a hot streak. However, no captain would shirk the opportunity to obtain statistical proof that their choice is best, or highlight risks that they may have missed.

The code in **neural\_network\_team\_predictor.ipynb** uses the neural network model from section 3.2.2 to calculate the matchup combination yielding the highest probability of winning for the home team (team A). It is demonstrated in Figure 19 for a contest between two very evenly matched teams. Of all possible permutations, the average probability in % of team A winning is found to be 50.6%. The optimal set of matchups yields a winning probability of 57.3%, a gain of 6.7%. While not a huge amount, it could be sufficient to tip the balance in favour of team A.

## 4 Summary

In conclusion, we have seen that the outcomes of amateur pool matches can, to a certain extent, be predicted based on records of previous results. The most important factor in determining the victor was found to be the player's historical win %, rather than their skill level. Despite grumblings from certain players, the handicapping system was by and large found to function equally effectively across all skill levels. On average, highly skilled players were found to prevail in 59% of matches.

Despite the randomizing effect of the handicap system, matches could still be predicted with an accuracy of 65% by use of an artificial neural network. The neural network was found to yield the highest prediction accuracies, ahead of alternative methods such as logistic regression and random forests. The accuracy is limited by a lack of in-game statistics, which are not recorded by NAPA. A model trained on a dataset including these features, such as number of innings per game, or number of defensive shots, is likely to provide a better metric for measuring the strength of a player. Furthermore, inclusion of polynomial terms in logistic regression, or deep layered neural networks, are two possible avenues to explore in order to improve the model performance.