Make sense of DiMaggio's hitting streak (and win a fortune along the way)

July 6, 2021

Joshua Silver

Eighty years ago this month, Joe DiMaggio was in the midst of a hitting streak that would captivate the nation. By the time his streak ended on July 17th, 1941, he had hit in 56 straight games and established one of the most extraordinary achievements in the history of athletics.

In 2001, Major League Baseball challenged fans to break DiMaggio's streak through an online contest called Beat the Streak in which each fan selects one player per day who they think will get a hit. If the player gets a hit, the fan receives credit for the day, and continues on to the next day with another selection. If a fan correctly choses batters for 57 consecutive games, they win $5.6 million. While there have been more than 100,000 fans playing each day, not one has made it past 51 straight games.

The fact that this contest is so difficult to win proves the staggering improbability of DiMaggio's streak, and begs many questions: Exactly how improbable was DiMaggio's streak? Could any of today's players break his record? How difficult is it for a fan to win the MLB contest? These are just some of the questions we'll answer. And we'll show you how to use machine learning to bring home that prize money for yourself.

Why DiMaggio's Streak Was Virtually Impossible

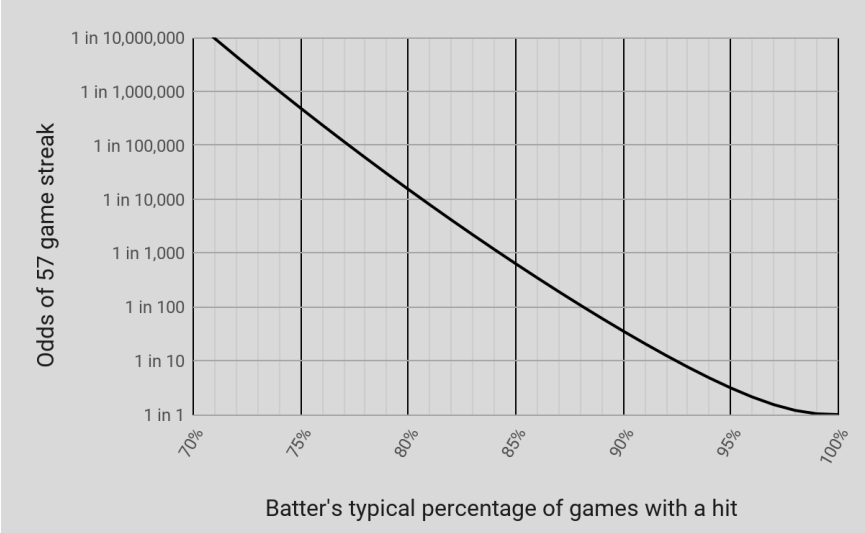
During DiMaggio's 1941 season, he batted .357 and hit safely in 84% of his games. Statistically, a player who hits in 84% of games has a 1 in 20,000 chance of hitting for 57 straight games $(0.84)^{57}$.

Of course, the likelihood of a streak occurring some time in the season grows because the opportunity to begin a new streak occurs many times throughout the season. Therefore, the calculation for odds of a 57 game streak occuring at **any** point during a season can be generated using de Moivre's formula from 1738.[1] The probabilities plotted in Figure 1 below show that even if a player averages a hit in 84% of his games, the chances that he would hit 57 times in a row at some point during a 162-game season is only 1 in 1,200[2].

[1] Malinovsky, Yaakov. "A note on the closed-form solution for the longest head run problem of Abraham de Moivre." 2021, https://arxiv.org/pdf/2009.07765.pdf.

[2] In 1941, the season was only 156 games long and DiMaggio only played in 139 games which made his odds slightly more difficult than 1 in 1,200.

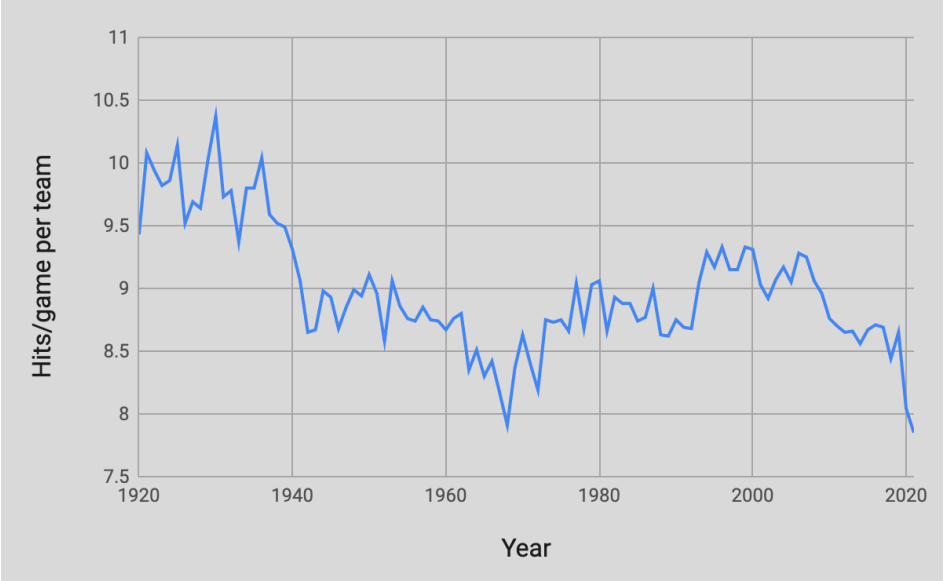
Figure 1: Odds of a 57 game streak in a 162 game season



Which of today's players could challenge DiMaggio's streak?

Of course in the last 20 years, hits are markedly down (See Figure 2), and we are currently flirting with the lowest league-average batting average in the history of the game.

Figure 2: Hits/game over the last 100 years



And, less than half way through the season, there have already been six no-hitters.

Nonetheless, there are batters who consistently end each game with at least one hit. Table 1 lists the players year-by-year who were the most consistent at getting a hit each game. Table 2 lists the 10 players over the last 5 years with the highest percentage of games with at least one hit.

Table 1: Players with the highest percentage of games with a hit each year

Year	Top Player	% of Games with a Hit(min 100 games/yr)
2011	Jacoby Ellsbury	82.2%
2012	Derek Jeter	81.1%
2013	Michael Cuddyer	80.8%
2014	Jose Altuve	80.4%
2015	Dee Gordon	80.0%
2016	Mookie Betts	81.0%
2017	Jose Altuve	77.6%
2018	Jose Altuve	78.1%
2019	Whit Merrifield	79.5%

Table 2: Players with the highest percentage of games with a hit over the last 5 years

Batter	% of Games with a Hit(min 300 games)
Whit Merrifield	75.9%
Charlie Blackmon	75.2%
DJ LeMahieu	74.9%
Trea Turner	74.8%
Michael Brantley	74.6%
Jose Altuve	74.6%
Mookie Betts	74.2%
Xander Bogaerts	73.5%
Jean Segura	72.4%
Alex Bregman	72.1%

Beat the Streak

Despite the small number of hitters seemingly capable of challenging DiMaggio's actual record, it seems as though at least one of the 100,000 contestants consistently playing Beat the Streak would have won by now.

How can we pick the player most likely to get a hit each day? Should we pick based on the best batter or based on the worst pitcher? How do we incorporate the effects of the batting order, ballpark, lefty/righty platoon matchups, head-to-head history, hot/cold streaks, and the weather. Is there any way to use advanced stats such as batted-ball exit velocity or batted-ball expected batting average? Finally, when we do choose a player most likely to get a hit, is there a way to "explain" our pick, so that we can understand why a batter was picked and can validate that the pick was made for the right reasons?

To blend all of these factors together, we created a model called Singlearity-BTS. Singlearity-BTS uses an AI computing technique called a neural network, and our approach is similar to how we built a model to predict the outcome of a Batter vs. Pitcher plate appearance.

Neural networks are a branch of machine learning that mimic the computation done in the human brain. Unlike more simplistic models, neural networks are able to discover nuanced and complex relationships. The goal of a neural network is to predict an output value based on input values. Singlearity-BTS predicts the number of hits that a batter in the starting lineup is likely to get in a given game. We then assume that the players with the highest predicted number of hits in a game are also the players most likely to get at least one hit in a game.

Feature Creation

The process of determining which inputs to use to predict an output is called feature creation. We defined each appearance of a batter in the starting lineup as a batter-game-instance (BGI). The inputs to the BGI consisted of 127 floating-point numbers representing information that we speculated was important to predict how many hits the batter would get. The output of the BGI was the number of hits the batter got in the game.

Table 3: Singlearity-BTS inputs[3][4]

Feature Group	# of Inputs	Details
Batter 3 Year Stats	13	Batter last 3 year offensive stats
Batter 1 Year Stats	13	Batter last 1 year offensive stats
Batter 21 Day Stats	13	Batter last 21 day offensive stats
Batter 1 Year Usage	5	1 Year stats about rate at which batter is subbed out
Batter 21 Day Usage	5	21 day stats about rate at which batter is subbed out
Pitcher 3 Year Stats	13	Pitcher last 3 year pitching stats
Pitcher 1 Year Stats	13	Pitcher last 1 year pitching stats
Pitcher 21 Day Stats	13	Pitcher last 21 day pitching stats
Batter/Pitcher Head-to-Head	10	Batter-pitcher 3-year head-to-head stats
Park Factor	5	Relative offense/game at this venue
Batting Order	1	Batter's 1-9 position in the Batting Order
Batter Position in Field	10	One-hot encode valued of batter's position(P,1B,...,DH)
Platoon Stats	7	Batter's and pitcher's 3-year platoon statistics
Home or Away	1	Home or Away
Weather	1	Game time temperature at the start of the game
Batter GB/FB Ratio	1	Batter's 3-year ratio of ground balls to fly balls
Pitcher GB/FB Ratio	1	Pitcher's 3-year ratio of ground balls to fly balls
Game Year	1	Game year
Game Day	1	Day of the season
Total	**127**	

Training

To train our neural network, we used publicly available MLB data from 2011 through June 1, 2021, representing 23,404 regular season games played on 1,747 different dates, for a total of 421,205 BGIs. We split the total of 1,747 game dates such that 60% were used for training data, 20% for validation (e.g. architecture and parameter tuning) and the remaining 20% for testing results on our selected trained network.

The goal of the neural network was to predict how many hits the batter was likely to get. The network was optimized to minimize the mean-squared-error (MSE) between the predicted hits and the actual hits[5].

Results

The trained neural network predicted how many hits each starting batter was likely to get in a game. When compared to actual hits/game, the error rates were 0.7637 MSE on the training data, 0.7662 MSE on the validation data, and 0.7664 MSE on the test data. This indicates that the neural network was not overfitting on the training data.

---

[3] Batter/Pitcher x 21day/1year/3year stats are: G, PA, PA/G, H/PA, HR/PA, wOBA, xWOBA, xH/PA, MaxLS, AvgLS, ParkFactor, G-Imputed, PA-Imputed.

[4] Details for the remaining 127 inputs can be found at dash.singlearity.com

[5] We experimented with both the input feature selection and the architecture of the neural network and ended up with a network with 127 inputs, two internal layers of 80 nodes each, and a single final output representing the expected number of hits for the player. We used TensorFlow for the neural network software and did our experiments using virtual machines on Amazon Web Services (AWS). We used a hidden layer dropout value of 0.2 in order to prevent the network from overfitting the training data.

When the trained neural network was asked to produce the best pick each day from the test data, the batter it picked got at least one hit on 79.3% of days, and the picked batters averaged 1.354 hits/game. For reference, a batter who averages 1.354 hits/game would get 219 hits per 162 game season.

Revisiting Figure 1, we can see that if we are able to make a correct pick 79.3% of the time, the odds of a 57-game hit streak during the season are 1 in 24,000[6]. This is still very, very, unlikely but with up to 100,000 people playing each day, it seems as though someone could finally claim the $5.6 million prize.

How Singlearity-BTS stacks up

We compared Singlearity-BTS to other strategies for picking players:

Random - arbitrarily choosing a batter each day
Random Leadoff - randomly choosing a batter in the leadoff spot each day
Best Last Year - Choosing the batter that led the league in hits the prior season

The table below shows the average number of hits/game and the success rates of the various strategies used to pick the best player each day.

Table 4: Singlearity-BTS results

Strategy	Average # of hits/game	% of Games successfully hit
Random	0.913	61.8%
Random Leadoff	1.090	69.6%
Best Last Year	1.227	74.9%
Singlearity-BTS	1.354	79.3%

Explainability

While we now have an AI-based model that can accurately predict the number of hits a batter will get in a game, the formula for computing this number is contained in thousands of edge weights embedded in the neural network and is not understandable by mere mortals. So we also seek to make the model's predictions explainable to human beings. This marrying of human understanding with AI is part of a burgeoning field known as explainable AI, of which there are two main parts: local and global explainability. To create explainable models, we use Shapley values which are a game theory concept to attribute the impact of different inputs to the ultimate prediction.

Local Explainability

Local explainability describes the importance of different inputs to the outcome of a single prediction.

Let's look at Table 5 which shows the predictions the neural network made for games on a randomly chosen date, June 3, 2021. We show 3 different batters, along with the opposing starting pitcher, the

[6] The Beat the Streak contest actually has two variations which make things easier: 1) the *Mulligan* and 2) the ability to choose two players per day. If utilized correctly, these odds can increase your chances of winning by 2-3x.

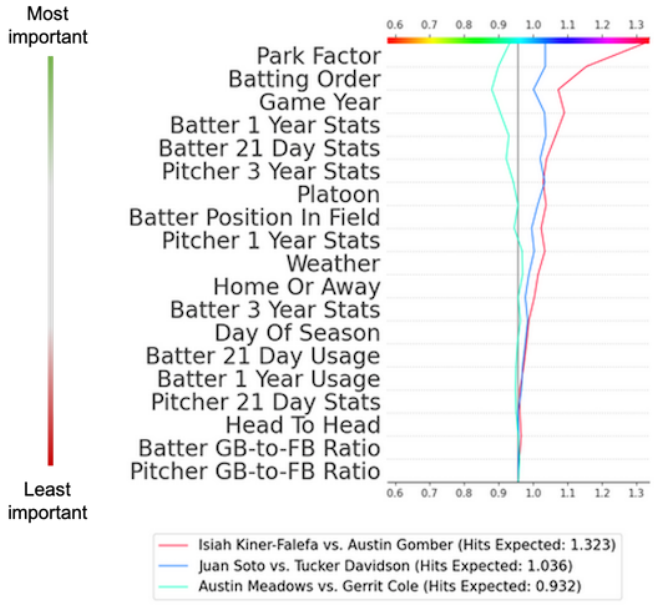
batter's position in the order, and the temperature for each game. Note that the actual prediction that Singlearity-BTS makes will be based on 127 different inputs for each of these rows.

Table 5: Sample predictions on June 3

Batter	Starting Pitcher	Location	Batting Position	Temperature	Prediction (# Hits)
Isiah Kiner-Falefa	Austin Gomber	Coors Field	1	83°	**1.323**
Juan Soto	Tucker Davidson	Truist Park	3	80°	**1.036**
Austin Meadows	Gerrit Cole	Yankee Stadium	4	69°	**0.932**

We can quantify and visualize the rationale for the predictions using Figure 3. This graph, called a *decision plot,* uses Shapley values to explain how the neural network made its predictions[7]. Reading the graph from the bottom up, we see how the predictions vary from the league average of 0.913 hits/game for starters during the last 10 years. For instance, for the prediction for Isiah Kiner-Falefa, we see that the Shapley value for *Park Factor* was approximately 0.20, driving his predicted hits/game from approximately 1.12 to 1.32. Thus, we can now visualize and quantify the impact that Coors Field had on this prediction.

Figure 3: Local explainability

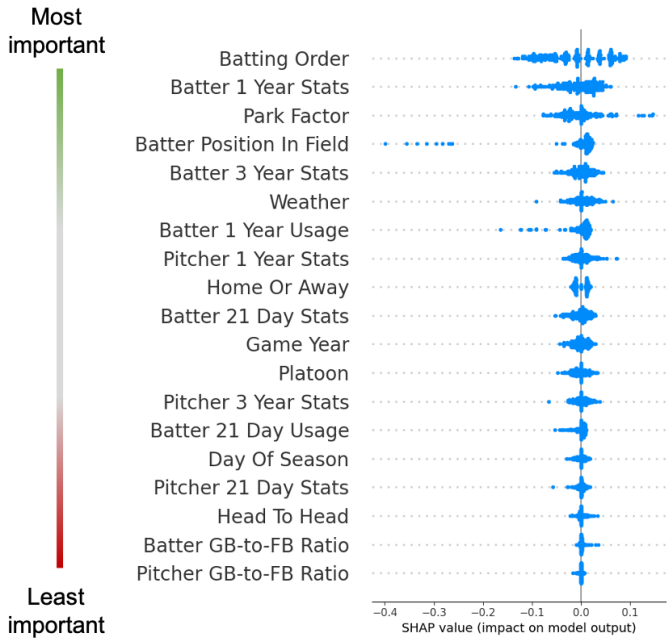


[7] We used the Python-based SHAP implementation by Lundberg and Lee for deriving Shapley values and creating visualizations.

Global Explainability

Global explainability measures the importance (on average) of a feature for making an accurate prediction. We can visualize global feature importance by plotting the Shapley values for a sample of predictions. Figure 4 ranks the most important features impacting the prediction of hits/game for batters. A single blue point represents the Shapley value for a single batter's game due to that feature. From this graph, we see that the three most important features for predicting a batter's hits/game are **Batting Order**, **Batter 1 Year Stats**, and **Park Factor**. **Batter Position in Field** is the fourth most important feature, but this is due almost entirely to Singlearity learning that pitcher's produce few hits/game.

Figure 4: Global explainability



Wrap-Up

AI-based models have the ability to use massive amounts of data to help with predictions and decision making. We've built the model Singlearity-BTS that can predict how many hits a starting batter is likely to get in a game.

You can view daily predictions of Singlearity-BTS and the Shapley explanations of the predictions here.

With tens of thousands of people playing this contest daily and the odds of winning (assuming good picks) at better than 1 in 24,000, it's high time for someone (maybe you!) to lay claim to the $5.6 million prize.