Major League Baseball Pitch Sequence Neural Network: Using Recurrent Neural Networks to Automate Pitch Sequencing

Quin Yuter

March 30th, 2025

Abstract

This personal research project investigates whether Major League Baseball pitch sequences can be predicted using Recurrent Neural Networks (RNNs). Focusing on 2024 Cy Young winners Chris Sale and Tarik Skubal, pitch level data was gathered, cleaned, explored, and used to build GRU-based RNN models using PyTorch. While model test accuracy hovered around 50%, I developed a new metric called **Predictability Score (PS)** that better reflects how predictable a pitcher is, accounting for the number of pitches in their arsenal. The results suggest that Tarik Skubal is slightly more predictable than Chris Sale, with a PS of 0.375 compared to 0.350. This project not only sharpened my RNN and Pytorch skills but also yielded interesting questions about how data science can affect decision making in baseball. There is still plenty of room for this model to grow, whether it be incorporating batter tendencies, experimenting with different model types, or removing sequence length constraints, but this was a rewarding step towards building a predictive tool that could benefit hitters, pitchers, and analysts.

I. Introduction

Many say that hitting a baseball is the hardest thing to do in professional sports. Pitching has become so advanced that it is hard for any hitter to get an upper hand in each at-bat. But what if there was a way to gain leverage for the batter? Through machine learning techniques, this may be possible. I want to see if there is a way to predict what pitch will come based on the pitches already seen in a given at-bat. For this project, I am focusing on two pitchers who inarguably performed the best in the 2024 Major League Baseball season: National League Cy Young winner Chris Sale and American League Cy Young winner Tarik Skubal. Is one pitcher more predictable than the other? How do they compare? Through my various data science methods (see Section II), these are questions I was able to answer.

Image 1: Chris Sale

Image 2: Tarik Skubal

Do I think I will be successful in creating models that are accurate in their predictions? Probably not. The complexity of pitching, beyond what can be put into data (i.e. pitcher confidence and other emotion), makes it extremely difficult to create well-performing ML models.

II. Methodology

There are three important steps to being successful in completing a full, end to end data science project. These are: Data Cleaning and Preparation, Exploratory Data Analysis, and Modeling. This entire project was done using Python.

a. Data Cleaning and Preparation

PyBaseball is a library in Python that scrapes data from websites like Baseball Reference, Baseball Savant, and Fangraphs. I used a few different commands to get the data for every single pitch thrown by Chris Sale and Tarik Skubal. However, this data is not clean. In other words, there is data missing, there is data that is not necessary, and there is not any particular order to it.

To start, I had to drop “noisy” columns, or columns that did not matter in the context of what I was trying to do. My reasons behind dropping columns include: Unnecessary information about fielders, data unrelated to pitch sequencing, data that describes what happens after the pitch is thrown (data leakage risk), and historical data that has no impact on real-time sequences. The columns I kept define the game state, pitch characteristics, and context necessary for sequencing. For example, pitch_type, release_speed, and the other pitcher metrics describe what the pitcher does (and really who they are as a pitcher). Game context (like balls, strikes, and inning) are important because they affect decision making. The rest of the columns kept are necessary to sequence the data.

I also had to deal with missing data. Collecting data is not easy and it will not always be perfect, which means it is common that there will be important data missing from the data sets. There are two methods to dealing with missing data: removing the entire observation (or in this case, sequence) or imputing the missing value. Tarik Skubal was the only pitcher that had missing data. To deal with his missing data, I deleted the sequences that were missing the pitch types, and imputed the data that weren't. Spin rate and spin axis were two variables that were missing data, so I took the average spin rate and axis for each type of pitch and used those values to fill in the missing observations.

Now that the data is full and focused, I can one-hot encode my categorical data. One-Hot encoding takes data that is categorical (i.e. pitch type) and creates new columns for each category. Then, a 1 takes place in the column whose category represents the observation, and a 0 is filled in for the rest of the columns. Recurrent Neural Networks require one-hot encoding because they expect a numerical input.

While I have experience using RNNs, my knowledge is not full. That is, I am using this project to hone my skills rather than add new ones. Part of my limited RNN knowledge is that I do not know how to model data that has sequences of different length (see Section V). So, I need to manipulate the data such that all the sequences are the same length. After exploring both Chris Sale's and Tarik Skubal's pitch sequences, I found that the longest length of sequence that they reach most often is four pitches. So, I removed any sequences that were not of length four and removed any pitches from sequences that went beyond four pitches (5th pitch, 6th pitch, etc.).

The penultimate step of data cleaning is normalizing the data, which helps with efficiency of modeling. However, to someone who may not have a data science background, this doesn't mean much. Normalizing data puts every variable on the same scale (i.e. 0 to 1). Why does this matter? Think of creating ML models as coaching. Let's say a baseball coach is looking at his/her players' statistics to make decisions regarding playing time. They have running speed, which is single digits with decimals, batting average, which is a decimal, and homeruns, which could be single or double digits, but is always just a discrete integer. The coach may think the higher number statistics, like home runs, are more important because the number is higher. This isn't the case. Normalizing the data puts all the statistics on the same scale so that they can be fairly compared. Back to data science methods, having normalized data helps the RNN focus on each variable equally, leading to more accurate predictions.

The final step of data cleaning and preparation is adding a unique identifier to each sequence. This helps with modeling, as it divides the data into multiple sequences instead of having the model think the data is just one long sequence.

b. Exploratory Data Analysis

Exploratory Data Analysis, informally known as EDA, is extremely important. Not only does it help you learn more about the data (like trends and past performance), but it also can help you decide the best methods to use. That being said, I already knew I was going to use RNNs, but the EDA helped me eventually with interpreting my models. I split my EDA into three sections: Pitch Types, Pitch Sequences, and Situational Pitching.

In the pitch types section, I learned more about the nature of each pitcher. I first wanted to learn more about their usage and distribution.

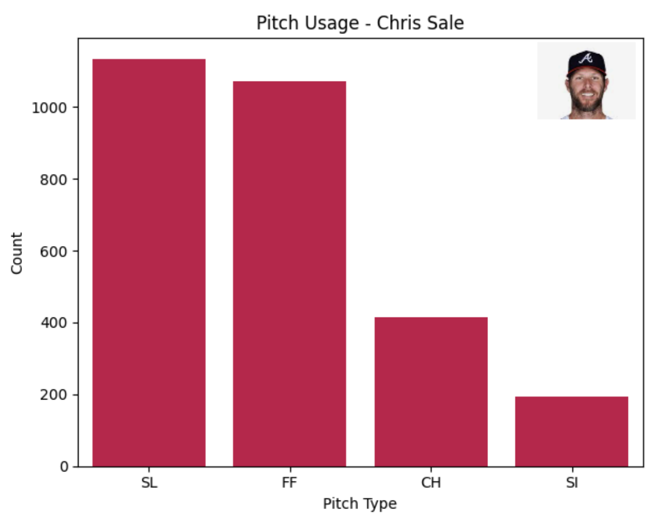

Figure 1.1: Chris Sale Pitch Usage

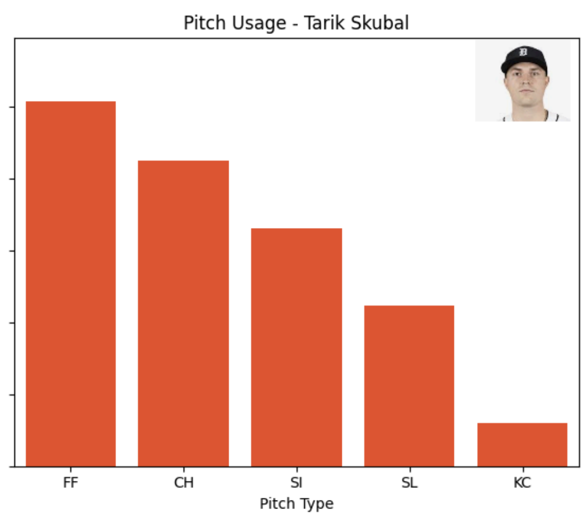

Figure 1.2: Tarik Skubal Pitch Usage

Figures 1.1 and 1.2 show us the pitch distribution for each pitcher. The first thing I notice is the similarity in their arsenals. They have all of the same pitches: Four-Seam Fastball, Slider, Changeup, Sinker. Skubal just adds one more pitch to his arsenal, a Knuckle-Curve, but it is rarely used. The cool difference between the two is the different distribution of usage of these pitches. Chris Sale uses his slider more often than his fastball, which I find to be extremely rare. He uses the slider the most, his four seamer the second most, and then we see an extreme drop in his usage in his changeup and his slider. Tarik Skubal's distribution looks to be a bit more linear. He, as expected, uses his four-seam fastball the most. His second most used pitch is a changeup, which sees just a bit less usage than his fastball. The rest of his pitches, in order from most to least used, are sinker, slider, and knuckle-curve. It is cool that these pitchers have such similar arsenals, use them so differently, but were still both the best pitchers in their respective leagues.

Let's dig a little deeper into these pitches. I want to see the speed of them. Are they similar in speed? Super different?

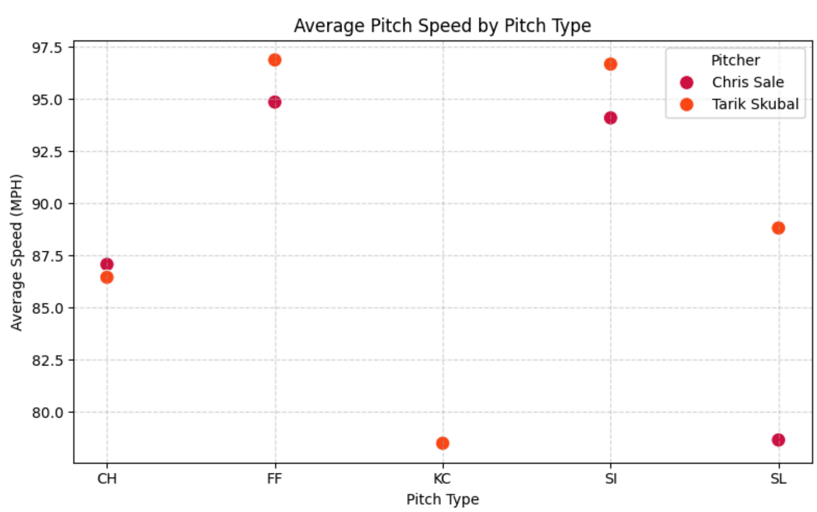

Figure 2: Average Pitch Speed by Pitch Type per Pitcher

In almost all of his pitches, Skubal is a faster pitcher. His average four seam fastball speed is almost 97.5, while Chris Sale sits around 95. Skubal's average sinker reaches about 97 mph while Sale's looks to be at about 94 mph. The most eye opening difference on this graph is that Skubal's slider is about 10 mph faster on average than Sale's. Sale's slider averages about 79 mph, which is on the slower end of pitches in Major League Baseball. However, Sale's changeup is just faster on average than Skubal's, but this difference doesn't look to have much meaning. The final pitch, which is Skubal's Knuckle-Curve, averages about 78 mph. This speed is much more similar to Sale's slider than his own slider.

This begs the question: what do their pitches look like in terms of H-break and V-break? It is obvious that speed isn't the reason behind being successful as a major league pitcher, as Sale looks to be on the slower end. So, do they fool batters with the movement on the ball?

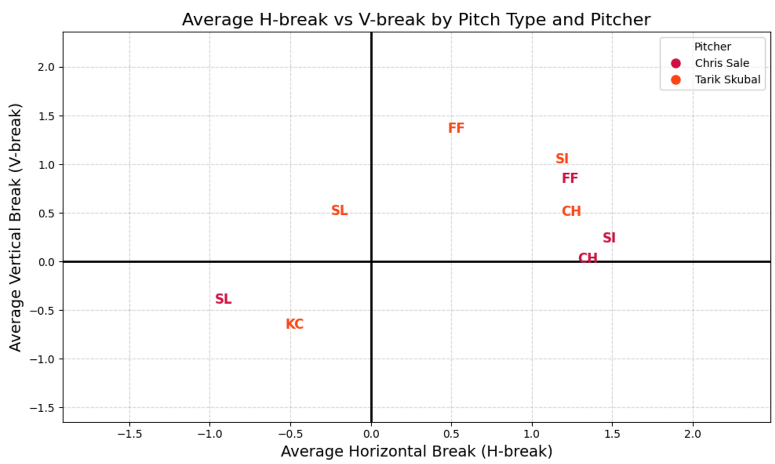

Figure 3: Average Break by Pitch Type per Pitcher

Let's first look at both of their four seam fastballs. Both of their fastballs move upwards and toward the left, as they are both lefty pitchers. However, Skubal's has more V-break, so his moves up more, while Sale's has more H-break, so his moves more left. As for Changeups, their H-breaks and V-breaks are similar as well. They see a similar H-break, with Skubal's V-break just being a bit more aggressive. Their sinkers both see positive movement in both directions as well. Skubal's has significantly more V-break, while Sale's has a small bit more H-break. What is most interesting about this is seeing how different their sliders are. Skubal's slider has upward movement, which I feel is somewhat rare. Sale's slider has downward movement. Overall, the H-break on their sliders differs by about an inch, which is quite significant based on the similarities of their other pitches. Skubal's Knuckle Curve moves more similar to Chris Sale's slider than his slider does.

I also want to learn about what outcomes are most common with each pitch type. First, what pitches yield more strikes than balls and vice versa? What pitches yield ground balls, fly balls, etc.? I am choosing to show the most common outcomes per pitch that is not a ball. Ball was the highest outcome for each pitch for both pitchers, which makes sense. Any pitch outside of the zone (that isn't swung at) is considered a ball, but any pitch inside the zone has 6 options that share strike zone frequency (not including hit by pitch).

Most Common Outcomes per Pitch per Pitcher (Not Ball)

Pitch Type	**Chris Sale**	**Tarik Skubal**
Change-Up	Hit Into Play	Swinging Strike
Four-Seam Fastball	Foul Ball	Foul Ball
Slider	Called Strike	Called Strike
Sinker	Called Strike	Called Strike
Knuckle-Curve	NA	Called Strike

Table 1: Outcomes Per Pitch per Pitcher

It is interesting that each pitcher has the same outcomes for almost every pitch. The changeup is the only pitch that they both throw that has a different outcome. The outcome of Chris Sale’s changeup that we see the most, other than ball, is Hit Into Play, while Tarik Skubal’s most common outcome for changeup is a Swinging Strike. This tells me that Tarik Skubal’s changeup is either harder to hit or harder to see. For the final part of this section of EDA, I want to see the most common at-bat ending outcome for each pitcher for each pitch.

Most Common At-Bat-Ending Events per Pitch per Pitcher

Pitch Type	Chris Sale	Tarik Skubal
Change-Up	Field Out	Strikeout
Four-Seam Fastball	Strikeout	Field Out
Slider	Strikeout	Field Out
Sinker	Field Out	Field Out
Knuckle-Curve	NA	Field Out

Table 2: At-Bat-Ending Events Per Pitch per Pitcher

Let's first look at Chris Sale. His change up and sinker yield field outs most often while his four seam fastball and slider yield strikeouts most often. This is interesting, as his two strikeout pitches are the pitches he throws the most. This makes sense, as most of the time, a pitcher would probably prefer a strikeout over the ball being put in play. As for Tarik Skubal, the only one of his pitches that has strikeout as the most common event is his changeup. The rest of his pitches most commonly are field outs. With much more equal usage (in relation to Sale), it doesn't surprise me that most of his pitches yield the same output.

As this project focuses on pitch sequencing, I wanted to learn more about the order of their pitches. First, what do the pitchers start each at-bat with the most? Does it differ heavily with righty and lefty hitters?

Distribution of First Pitches per Pitcher

Batter Stands	Pitch Type	Chris Sale	Tarik Skubal
L	Changeup	0.027	0.068
L	Four-Seam Fastball	0.491	0.041
L	Knuckle-Curve	NA	0.007
L	Sinker	0.188	0.728

L	Slider	0.295	0.156
R	Changeup	0.078	0.263
R	Four-Seam Fastball	0.405	0.402
R	Knuckle-Curve	NA	0.092
R	Sinker	0.103	0.061
R	Slider	0.414	0.182

Table 3: Distribution of First Pitch Pitches per Pitcher

Let's first look at how Chris Sale and Skubal deal with Left Handed hitters. We see that they approach these hitters very differently. Chris Sale does not have a pitch that he predominantly uses when dealing a lefty the first pitch of the at bat. He uses his fastball 49% of the time, but we still see no majority pitch taking over. Skubal, on the other hand, throws his sinker to lefties about 73% on the first pitch of the bat. Lefty hitters can expect a sinker from Skubal on the first pitch. As for righties, Sale again does not have a pitch that he relies on. He starts with his fastball and slider both 41% of the time. With righties, Skubal approaches at bats similarly to Sale, as there is no pitch he uses more than 50% of the time. He throws his four seam fastball 40% on the first pitch to righties, but that is not frequent enough for righties to assume that this is the first pitch they will see from him.

Alternatively, let's see which pitch they strikeout batters the most with.

Distribution of Strikeout Pitches per Pitcher

Batter Stands	**Pitch Type**	**Chris Sale**	**Tarik Skubal**
L	Changeup	0.026	0.297
L	Four-Seam Fastball	0.231	0.162
L	Knuckle-Curve	NA	0.027
L	Sinker	0.051	0.216

L	Slider	0.692	0.297
R	Changeup	0.103	0.385
R	Four-Seam Fastball	0.368	0.361
R	Sinker	0.012	0.173
R	Slider	0.519	0.082

Table 4: Distribution of Strikeout Pitches per Pitcher

Before getting into the super nitty-gritty of the cool stuff we learn here, I want to point out my initial observation, which is that there is no row signifying a righty Knuckle-Curve combination. I do not think this is a data issue, rather Tarik Skubal did not strikeout any righty hitters using his Knuckle-Curve.

Let's first look at Chris Sale. He used his slider almost 70% of the time to strike out lefties. In context, this makes sense. A lefty slider to a lefty goes away from a batter, making it harder to hit. Further, if Sale is up in the count, he can throw this out of the zone and maybe get some chase. For righties, it's a little less straightforward. He uses his slider about 52% of the time to strike out righties, which means that can expect a slider more than anything else, but this is nearly 50/50, decreasing its meaning and predictive ability from the batter's end. He uses his fastball about 37% of the time.

Tarik Skubal doesn't seem to have any "strikeout pitch". His pitches are all across the board. With lefties, the most used pitch to strikeout batters is changeup, but that is only 30% of the time. We see a bit more frequency pattern with righties, where changeup and fastball are used about 75% of the time. Technically, batters can expect either of these two pitches when they have two strikes. However, they are very different pitches, which can be seen in both the velocity graphs and movement graphs.

This would make me think that for Chris Sale, at least, we would see sliders more in the end of the at-bat, and other pitches in the beginning. Let's explore usage along at-bats.

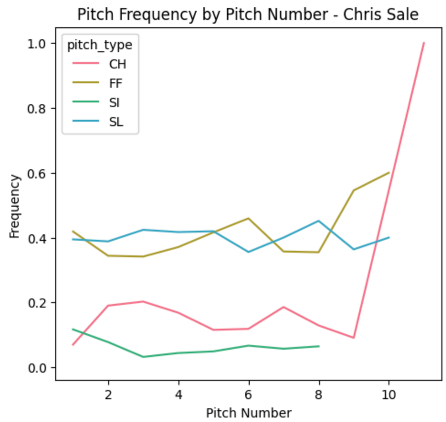

Figure 4.1: Chris Sale Pitch Frequency By pitch number

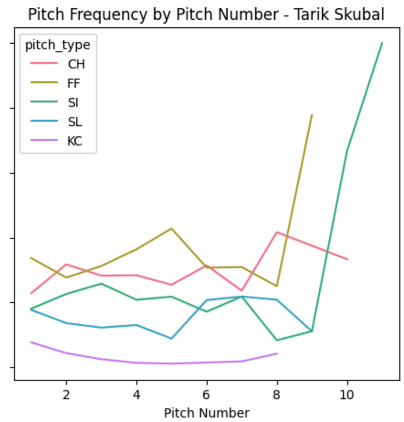

Figure 4.2: Tarik Skubal Pitch Frequency By pitch number

For this interpretation, I am really only going to focus on pitches 1-10. The big jump we see in frequency past pitch 10 tells me that they really only reached a pitch number past 10 once, in which they both threw a changeup so the frequency is at 1.0.

Looking at Chris Sale's line graph of pitch frequency by pitch number, we don't see many trends in terms of changing direction (i.e. low usage early in counts, high usage later in counts, and vice versa). We see that fastball and slider are consistently frequently used across all pitch numbers, while change up and sinker stay at a lower frequency. This was already seen in the pitch usage plots earlier. One thing that is super interesting about Chris Sale's graph is that his sinker was not used past pitch 8 in the count. Late in the count, he probably wants to use his more comfortable pitches, which are slider and four seam fastball, since he wants to just win the at-bat.

Tarik Skubal's line graph also does not show much change in direction, but his usage is much more "erratic". We still see that the four-seamer is the most used, mostly, with his changeup being second most used. We also see that Knuckle-Curve is consistently not used much across the board, which is what we saw in the pitch usage plot. Sinker and slider are more interesting pitches to me in this situation. Even though their change is quite minimal in frequency across pitch numbers, we do see somewhat of a trend. As he gets later in the at-bat, he increases his usage of his slider. Inversely, he decreases his use of his sinker.

These plots tell me that when it comes to using the RNN to predict the next pitch, we will likely see four seam fastball or slider for Chris Sale and either four seam fastball or changeup for Tarik Skubal.

Another super important part of pitching is situational pitching. How does their arsenal change based on the situation at hand? Do they pitch differently early in the game vs. late? How do they pitch with no outs vs. with 2 outs? What about when they are ahead in the game versus behind?

Let's first look at how they pitch differently as the game progresses.

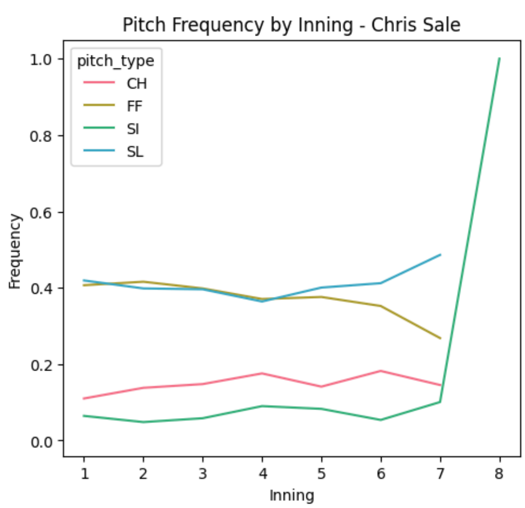

Figure 5.1: Chris Sale Pitch Frequency By pitch number

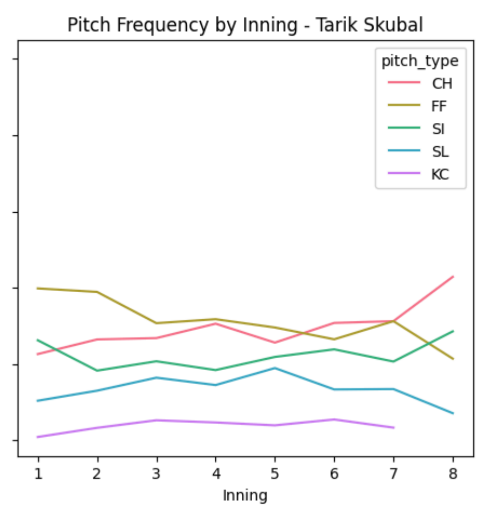

Figure 5.2: Tarik Skubal Pitch Frequency By pitch number

The outputs to these two plots are pretty much expected based on the various plots prior to this. However, I was expecting to see a bit more of a pattern or trend. Based on the usage plots, we know that Chris Sale sticks mainly to his slider and four seam fastball while Tarik

Skubal sticks mainly to his four seam fastball and changeup. I conclude that these are probably the pitches they are more comfortable throwing, whether that be they are more confident in their outcomes or are easier on their arms. That being said, I would think that as the game goes on, these pitches would increase in frequency while the lesser-used pitches decrease in frequency. Looking at the plots above, that is not the case. They are pretty consistent across innings with what pitches they use.

The only trend I can kind of see is a decrease in fastball usage for both of them as the game goes on. Could this be because it is harder on the arm? We also see a bit of an increase in Skubal's Changeup usage as the game goes on. However, with the minimal change, I don't think we can jump to the conclusion that certain pitches increase in usage as the game goes on while others decrease.

What is super interesting is just the sheer difference in usage in Chris Sale's arsenal. He uses his slider and four seam fastball about 30% more than his other pitches. On the other hand, Skubal definitely has a hierarchy of most used to lesser used pitches, but it isn't as spaced out. Will this make Chris Sale more predictable? Will my RNN for him work better?

Let's now look at if we can learn anything regarding pitch usage and outs.

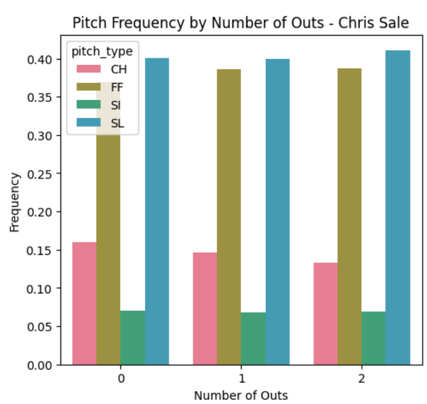

Figure 6.1: Chris Sale Pitch Frequency By Number of Outs

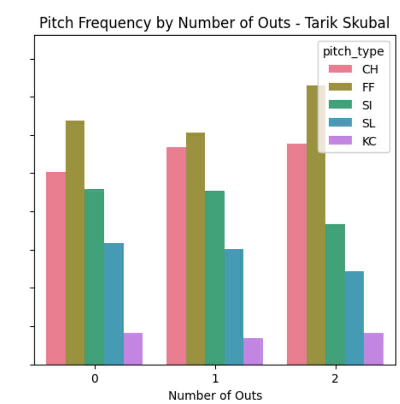

Figure 6.2: Tarik Skubal Pitch Frequency By Number of Outs

Again, this looks exactly as expected. There is not much we really learn here. We see, in Chris Sale's plot, that he predominantly sticks with his four-seamer and slider, and has much rarer usage of his other two pitches. There is really no difference across outs in the frequency of these pitches.

With Tarik Skubal's frequency across outs, there is a bit more of a pattern, but nothing super noticeable. His four-seam fastball and Changeup usage increases with 2 outs, while the other main two (not including Knuckle Curve), decrease. However, again, this difference is not big enough to reach strong conclusions.

I will now look to see if there is a difference in how they pitch ahead or behind in the count. I am considering an even count as the pitcher being ahead because there are more balls allowed than strikes

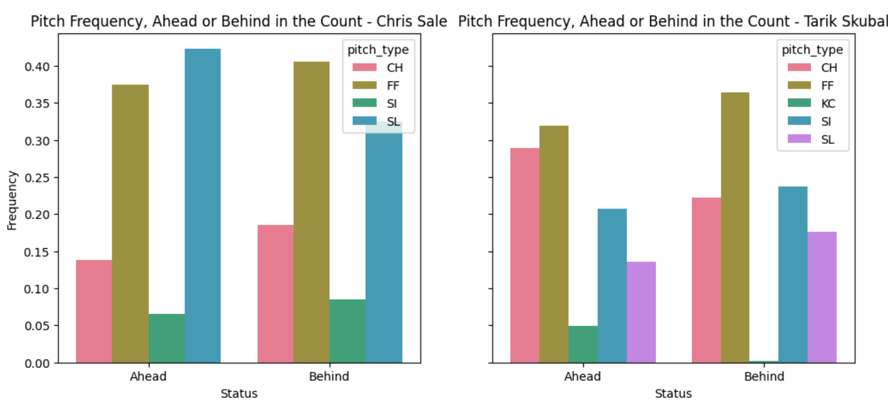

Figure 7.1: Chris Sale Pitch Frequency Ahead or Behind in the Count

Figure 7.2: Tarik Skubal Pitch Frequency Ahead or Behind in the Count

I'd say this is the most interesting plot we have seen in the 'Situational Pitching' section. There are actual differences in what we see here. First, looking at Chris Sale, we see that he uses his slider when he is ahead in the count more but uses his fastball more when he is behind. As for his changeup and sinker, usage goes up when he falls behind in the count. However, focusing more on his two most used pitches, I feel like the difference between the two statuses is enough to come to the conclusion that batters should expect more sliders when they are behind in the count and expect more four seam fastballs when they are ahead in the count.

We also see some cool changes in usage when Tarik Skubal is ahead or behind in the count. His four seam fastball usage seems to increase by about 5% when he is behind in the count. Is he more confident in this pitch than others? On the other hand, his changeup usage decreases when he is behind in the count from when he is ahead. This tells me that he probably struggles with the control of this pitch a bit more. Again, I feel like these differences are enough to come to the conclusion that if the batter is ahead in the count, they can expect way more four seam fastballs. When they are behind, they can probably expect some more changeups.

Now that we have learned a lot of cool information about the two pitchers through this EDA, we can discuss the main part of this project - modeling!

c. Modeling

While I have experience with Recurrent Neural Networks, I have only performed them using the TensorFlow library in Python. Another popular library to perform Neural Networks in Python is PyTorch. Therefore, this project doubles as a learning experience for me, giving me practice to get more comfortable not just with the new library but with Recurrent Neural Networks as well.

You may ask, why use Recurrent Neural Networks? Let's first break this into two parts: Recurrent and Neural Networks. Oxford Dictionary's definition for the word recurrent is "occurring often or repeatedly". In the terms of pitch sequencing, we see pitch sequences occurring one after the next. Neural Networks are a machine learning algorithm that models the way the human brain thinks. Putting these ideas together, Recurrent Neural Networks work really well with sequential and time series data because it maintains the memory of inputs as they repeat, applying said memory to future predictions. Pitch sequencing is sequential data, which is why RNNs work best for this project.

Before modeling can truly begin, there are a few things that need to be done. First, I split the data, 80% of the sequences going to training the model and 20% going to validating. This is important in knowing truly how well a model fits the data. Once this was done, I could start with hyperparameter tuning and eventual modeling. For this, I performed these two steps separately for each pitcher due to the fact that their arsenals are different. See Section V to know how I would change this method in the future.

I hyperparameter tuned my models using Optuna, which is the library that works best with PyTorch. Hyperparameter tuning is the process of finding the best hyperparameters of the model. Think of hyperparameter tuning like adjusting an oven before baking a cake. Hyperparameters of baking could be the temperature to set the oven to, the length in which you bake the cake, and the setting (conventional bake, etc.) in which you bake. You want to choose these settings based on what would give you the best cake as a result. These "settings" are called hyperparameters in machine learning.

To perform hyperparameter tuning, I first had to create the baseline model, in which I used Gated Recurrent Units (GRU) instead of Long Short-Term Memory (LSTM). LSTMs work better when the sequences are long, and with only length 3 sequences, GRUs are the better choice. They work better with simplicity and are more computationally efficient because of it. The different hyperparameters I tested were Batch Size, Epochs, Hidden Size, Learning Rate, Number of Layers, and Drop Out Rate. See the appendix for definitions of these hyperparameters. The final model, or the combination of hyperparameters that were chosen were based on the best balance between test loss and test accuracy. Using the testing metrics to evaluate performance is important because they represent the model being used on new data, not data it was trained on. This most mimics how well it would perform if it is deployed to real-time data.

Test accuracy is great, and is necessary in model selection, but it is kind of misleading in evaluating the predictability of each pitcher. Let’s take a step back. Chris Sale has 4 pitches. Tarik Skubal has 5. Thinking simply, that every pitch is equally as likely to throw, any test accuracy above 25% would suggest that the Chris Sale model is better than random. Similarly, any test accuracy above 20% would suggest that the Tarik Skubal model is better than random. This is because 1/4 or 1/5 is the baseline accuracy, so anything higher is better. I want to create a predictability score to more accurately reflect the power of the model. An untrained eye will see 50% (See Section III. Results) is pretty bad, which isn't necessarily the case. Normalizing the outcomes better helps with comparison as well as interpretation.

The equation for Predictability Score (PS) is as follows:

$$Predictability\ Score\ =\ \frac{Model\ Accuracy - Baseline\ Accuracy}{1 - Baseline\ Accuracy}$$

Where

$$Baseline\ Accuracy\ =\ \frac{1}{Number\ of\ Pitches}$$

This is an important metric when interpreting the models and their performances. Chris Sale and Tarik Skubal have a different number of pitches. Technically predicting pitches is more difficult for someone who has more potential outputs, or pitches. In this case, Tarik Skubal would be more difficult to predict since he has 5 pitches while Chris Sale only has 4. It allows us to fairly compare the performance of the models and truly decide who is more “predictable”.

III. Results

At initial glance, neither model performed very well. Chris Sale's model has a 0.512 test accuracy while Tarik Skubal's model has a test accuracy of 0.5. It would seem that Chris Sale's model is better, thus making Chris Sale more predictable, but it is important to remember that we cannot compare the two pitchers until their accuracies are normalized using the predictability score equation above.

A great way to understand how the models performed is by looking at their Confusion Matrices.

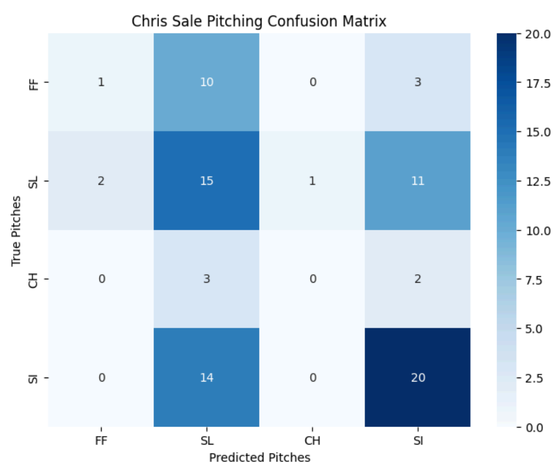

Figure 8.1: Sale Confusion Matrix

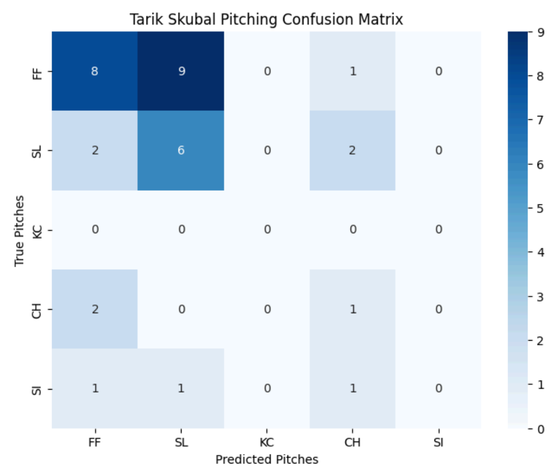

Figure 8.2: Skubal Confusion Matrix

Looking first at Chris Sale's Confusion Matrix, we see that the most common mistake made by the model is confusing Sinker and Slider. We see 14 pitches that were predicted to be sliders that were actually sinkers, and 11 that were predicted to be sinkers that were actually sliders. The second most common confusion by this model is Slider and Fastball. 10 pitches that were predicted to be sliders were actually fastballs. What is quite interesting about this is that there is only 1 pitch that is the inverse. The pitch that is most commonly correctly predicted is sinker, as 20 predicted sinkers were actually sinkers. The second most common correctly predicted is slider, as 15 pitches were classified correctly in this category. Looking back at the pitch usage in the EDA, we see that Chris Sale uses his slider the most. So, it makes sense that most of the pitches were predicted to be sliders. However, he uses his four seam fastball the second most and his sinker the least, which is not reflected above. Based on pitch usage, I would think more would be predicted fastball, yet only 3 pitches were predicted so. On the other hand, sinkers were the least used pitch in 2024 for Chris Sale, yet it is the second most common prediction.

Tarik Skubal's confusion matrix is quite interesting. First, we look at the diagonal, which is where the predictions were correct. We see that this model performed well in predicting fastballs, as 50% of the true fastballs were predicted correctly. Then, we see that 3 sliders were predicted correctly. However, no knuckle-curve, changeup, or sinker was predicted correctly. To be fair, there were no knuckle-curve observations that were the fourth pitch of the at-bat, and there were only 3 each of changeup and sinker. Therefore, there were less chances for this model to perform well. We see that the biggest confusion this model faces was that there were 5 slider observations that were predicted to be fastballs. This is quite interesting because Skubal's slider and fastball are not similar, they do not have similar movement and his fastball is significantly faster. Overall, we see that a majority of the observations were predicted to be his four-seam fastball, which makes sense in accordance to his pitch usage as seen in the EDA.

Based on the confusion matrices, we can conclude that Chris Sale's most predictable pitch is his sinker while Tarik Skubal's most predictable pitch is his slider. To learn a bit about their first few sequences, see Figures a.1 and a.2 in the Appendix

As stated before, it is important to have some normalized score to be able to compare pitchers. Predictability Score not only gives us a way to compare pitchers, but it also could be an added statistic that can help gauge how well a pitcher is performing. The model predicted 51.2% of Chris Sale's pitches correctly, while the model predicted 50% of Tarik Skubal's pitches correctly. Since Tarik Skubal has one more pitch, we can't compare them until we know their Predictability Scores.

Pitcher	**Predictability Score**
Chris Sale	0.350
Tarik Skubal	0.375

Table 5: Predictability Scores

We see that with a Predictability score of 0.375, Tarik Skubal is a more predictable pitcher. That means that 37.5% of the time, a batter should be able to predict what pitch is coming from Tarik Skubal. It is important to note, however, that neither of these scores are that high. That is, these pitchers have the upper hand since they are more unpredictable than they are predictable (1 - Predictability Score = Unpredictability Score).

IV. Conclusion

I started this project wanting to see if I could build Recurrent Neural Networks to predict pitches for Chris Sale and Tarik Skubal. I was going to just see how well the models performed, interpret their results in context, and call it a day. However, I realized that I was framing it all wrong. I could use these models to see how predictable a pitcher was, instead of just focusing on how well the model worked. I created a predictability score that normalized accuracy for each model, which allowed me to compare the two pitchers. Using this predictability score, we understand that Tarik Skubal's pitches could be predicted about 37% of the time, while we can predict Chris Sale's pitches about 35% of the time. Therefore, with the data and methods used, we can conclude that Tarik Skubal is slightly more predictable than Chris Sale.

The methods used to come to this conclusion were data cleaning, data preparation, exploratory data analysis, and modeling. The data cleaning and preparation was quite extensive, as this data was not perfect. The EDA was also very in depth, looking at pitch types, pitch sequences, and situational pitching. Some of the more important findings from the EDA were that although Chris Sale and Tarik Skubal have very similar arsenals, they use them very differently. Also, their pitches move quite differently as well.

Finally, using Optuna and PyTorch, I created Recurrent Neural Networks, which was my selected model. I further refined my RNN skills while also learning this new library in the process.

V. Limitations and Future Work

While valuable information was learned, this project isn't perfect. With my limited experience with RNNs, I have yet to gain intuition and efficiency in my methods. However, I learned a lot through this project, not only about RNNs but about PyTorch as well. That being said, there are many things I can do in the future to make this project much better.

First, I want to look at the structure of the model. In future iterations of this project, I would not have the 4-pitch at-bat limitation. Can I create this model for any length of pitch sequence? Further, I would create the model such that it can be applicable for any pitcher. This flexibility not only increases the number of training and testing data, but it maximizes its potential application.

While the data was extensive, I also think there are things that can be added to the data to make the models that much more accurate. First, is there a way to include batter tendencies in the data? A batter's favorite zone, best pitch type to slug, and biggest strikeout pitch are just a few things that pitchers will consider when deciding what pitch to throw. However, something to consider is that this would make the predictability score different for each batter. A new pitcher statistic would have to be created from this, called APS (Average Predictability Score), which averages all the pitcher predictability scores per batter. Another cool variable that could be added to the data is a scouting report. Can I apply Natural Language Processing (NLP) methods to this data to increase the accuracy even more?

Lastly, can I apply other data science methods to this data? What would it look like to use XGBoost on this data? Although it doesn't work as well with sequence data, XGBoost is still an incredibly powerful and popular modeling method. Is classification a better method for this type of problem?

I think, in the future, this model should be used by MLB clubs. On the batter side of things, they can learn what to expect from pitchers based on what they have seen. Batter's can gain a competitive advantage in an at-bat by knowing roughly what pitches they might see and in what order they might see them in. On the other side of the ball, if a pitcher is deemed predictable and tends to have patterns in their pitches, they may want to switch it up to confuse the batters. If, after a 3-set pitch sequence, they are predicted to throw a slider, they may throw a fastball instead to throw the batter off.

Overall, while this project goes in depth into how RNNs can be used to create a Predictability Score for pitchers based on predicting sequences, it has its limitations. This problem is an abyss of potential methods to try and data to add - it will never be "done". As this project potentially advances in the future, I think it can be an incredibly powerful tool to be used by Major League clubs.

Sources

Radhakrishnan, P. (2017, August 9). What are Hyperparameters and how to tune the Hyperparameters in a Deep Neural Network. *Medium*. Retrieved April 1, 2025, from https://medium.com/data-science/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a

Appendix

Definitions of Hyperparameters (Radhakrishnan, 2017):

1. Batch Size: Number of sub samples given to the network after parameter updates
2. Epochs: Number of times the network loops through the whole training data
3. Hidden Size (Units): Processing elements located within the hidden layers
4. Learning Rate: defines how quickly a network updates its parameters
5. Number of Layers: Layers between the input layer and output layer
6. Drop Out Rate: A regularization technique that helps to avoid overfitting

Figure a.1: First 5 sequences for Chris Sale

Sequence Number	**Pitch 1**	**Pitch 2**	**Pitch 3**	**Predicted Pitch 4**	**Actual Pitch 4**
1	FF	FF	SI	SI	SL
2	SI	FF	FF	SL	SL
3	SL	FF	SI	SI	SI
4	FF	SI	SI	SL	SI
5	FF	FF	SL	SL	SI

Figure a.2: First 5 sequences for Tarik Skubal

Sequence Number	**Pitch 1**	**Pitch 2**	**Pitch 3**	**Predicted Pitch 4**	**Actual Pitch 4**
1	FF	FF	SL	FF	FF
2	FF	SI	SI	SL	FF
3	SI	FF	SI	SL	SL
4	FF	FF	FF	FF	FF
5	SI	FF	FF	SL	FF