Predicting MLB Pitcher Injuries With Statcast Data

Drew Bennison
The Pennsylvania State University
Email: dhb5140@psu.edu

Mallet James
The Pennsylvania State University
Email: mrj5250@psu.edu

***Abstract*—At any level of baseball, including the major leagues, upper body pitcher injuries can derail a team's season. A major league team's pitching staff may be headlined by a 5+ wins above replacement rated ace pitcher, but if he is out for an extended period of time, this hurts not only his chances at competing at a high level but also the performance of his team as a whole. Predicting pitcher injuries is a complicated task, but with the vast amount of data that is collected and used currently to analyze MLB pitcher performance there has to be a better way to combat upper body and arm injuries and keep major league pitchers out of potential harm. Past research has focused on injury prediction using pitcher fastball velocity, age, and specific pitch repertoires. With the wide range of data points that MLB Statcast data tracking has collected since 2015, injury prediction can be conducted at a much finer level. In addition to tracking the speed of pitches over time that could signal injury we have the opportunity to look at many other features that relate to the result of a pitch. These features include horizontal and vertical movement, release point of a pitch, pitch location in and out of the strike zone and many more that have not been available in previous research endeavors. With this data we have been able to train a machine learning model that is able to help identify which pitchers are at a higher risk of developing an injury within the weeks following a start. Managers and team analysts alike can use this model to monitor their pitchers and better evaluate their players' injury risk to prevent injuries, prolong player careers, and win more games.**

I. Introduction

The advancement of data tracking and the widespread use of Statcast [1] information across Major League Baseball (MLB) has changed the game in many ways. Three to five terabytes of data are collected in every MLB game, and that number is on the rise. Two of the most notable ways Statcast has influenced the game include how front offices decide who to acquire and maintain as an organization and also how players themselves decide to structure their own development. The number of data analysts teams employ in their front office only continues to grow [8], and though their work isn't public, it is assumed that they utilize Statcast data in their analyses because of the level of detail it provides them. For every pitch thrown, detailed information is recorded about the release point, velocity of the ball, spin rates, as well as the location of every player on the field at the time. In the decision making process of the front office staff and at the player level, injuries play a major role in discussion of player acquisition and the potential impact that a player can have. Injuries can dissuade teams from signing players and severely hurt a player's future value. The focus of our study will be to try to determine if with Statcast data we can accurately predict pitcher injuries and try to spot the signs of a physical ailment before an injury occurs and time on the Injured List is required.

II. Purpose

Every year in Major League Baseball there are examples of pitcher injuries hurting an MLB team's chances at playoff contention or simply being a competitive baseball team. This in turn can cost a club millions of dollars, cause the loss of jobs within the organization, and most importantly negatively impact the player who has to spend extended time on the injured list. The purpose of our project is to develop a model that effectively classifies at risk injury behavior based on trends in Statcast measurements. The combination of a low release point with less movement on a pitcher's slider could possibly be cause for concern for an elbow injury, or there might be specific pitches that when thrown at a high velocity put a pitcher at risk for finger blisters. We hope to be able to spot these high injury-risk events with machine learning techniques to help improve pitcher longevity and performance. Successful prediction of injury events could help teams gain a competitive edge over their opponents and maintain a healthy roster throughout the long, 162 game season of a typical baseball year. Additionally, discovering what variables are important for injury prediction could be used to prevent certain types of injury from occurring in the first place, furthering the advantage teams would have if they chose to implement this sort of model. This is an area of the game that we feel has been under researched at the public level, yet could have huge implications for MLB baseball teams and pitching staffs

III. Stakeholders

The groups of people who will benefit from our project include players, managers, and medical professionals. MLB players will benefit as staying healthy is of utmost importance when it comes to performing at a high level. If they can more accurately train, recover, and identify when they are likely to be injured, they can prolong their careers and proactively seek help for injuries. Sports science is an increasingly important part of player development, and this would fit into that category of tools available to players. Managers will also receive many of the same benefits that the players do. With detailed injury predictions for their pitchers, they can intervene when necessary to protect their players and field the healthiest team possible. Managers are largely not responsible for players getting injured, and a model like the one we have developed would provide valuable information for them. Since baseball is not a contact sport for the most part, there are not as many visual clues that players should be monitored more closely for possible injury. In other sports, broken bones or concussions are things outsiders can see or players can identify, but in baseball injuries are almost always muscle related and thus tough to spot visually. Evaluating the data then becomes one of the main sources managers have to go by to find injury, particularly if the injury is forthcoming and has not started to physically trouble the player.

San Diego Padre Mike Clevinger leaving a game late in the 2020 season with an elbow injury, preventing him from playoff competition.

A particularly important group that will benefit from this research is medical professionals. Sports provide an environment for doctors to assess the reasons certain injuries occur in a relatively controlled environment – they have the ability to track every workout done by a player and every pitch thrown. The results of our work could be used to study how these injuries come to be in the first place, and hopefully that knowledge could be applied outside of sports. Although the average person is not throwing fastballs repeatedly, they might be performing actions in their hobbies or at work that use a similar motion and can cause injury. An environment like baseball could be a great testing ground for injury prevention and muscle longevity techniques. Our work should also make their job easier as the intention is that more at risk pitchers will be identified before serious rehab or injury remedy is required.

IV. Previous Research

Most of the past research that has been conducted on the topic of predicting pitcher injury has been focused around fastball velocity and the role that specific pitches play in pitcher injuries. Statcast data is relatively new within the past few years, so the current body of literature has yet to fully implement an injury detection model using all of the available measurements that Statcast now tracks and provides. Our approach will be to use fastball velocity as it has been predictive of injury in past research but to also use newer data measurements like spin, vertical and horizontal movement of a pitch, and pitch release position. To be able to most accurately identify what it is that causes a pitcher's arm to fatigue and ends up causing injury, we think that there will be lots of key indicators in Statcast data that have not been identified in previous research endeavors.

One study by Chalmers et al. [2] looked to identify the factors that led to ulnar collateral ligament reconstruction surgery in pitchers. The common name for this surgery is Tommy John surgery which any baseball fan will remember hearing at some point or another on a broadcast. It is very common and a product of the unnatural arm motion that throwing a baseball requires. Their research found that a higher pre-injury fastball velocity was the best predictor of the future need of Tommy John surgery, suggesting that pitchers who throw harder put themselves at a greater risk for future injury. For our project, the takeaway is that pitchers who increase their fastball velocity from multiple previous games average velocity might be putting themselves at a higher risk for injury and should possibly be flagged for further evaluation. This is an interesting conclusion because a naive manager might see that a pitcher is throwing faster than usual and think he is getting better and stronger, when in reality he might be showing the early signs of a serious injury. One of the features of our model is a variable that tracks the change between a pitcher's current game fastball velocity and their previous five game rolling average of fastball velocity. We hope that this variable detects the presence of a possible early ulnar collateral ligament injury.

Using fastball velocity along with other variables such as age and rate, Chalmers' group was only able to explain 7% of the variance seen in ulnar collateral ligament reconstruction surgery rates in pitchers. This points to two additional conclusions we can take from this journal article. First, predicting injuries is a difficult task, so when evaluating our model we will have to be cognisant of what the current state of injury prediction is and what a successful model really looks like. Second, this is definitely still a novel problem that needs a lot of work to be solved, so it is a worthwhile project to take on in the hopes that we can advance the understanding of injury prediction in pitchers.

A review paper by Erickson et al. [3] summarized the results of a body of work about pitcher injuries. The main takeaway from this paper was that statistics that measured season-long stats, such as the cumulative number of pitches thrown, was not predictive in determining whether or not a pitcher would have ulnar collateral ligament reconstruction surgery at the Major League Baseball level. It was predictive at the younger levels of baseball, suggesting that younger players might be more prone to overuse. This was surprising given that the classic argument for why pitchers get injuries at every level is because of overuse. Because we are looking at predicting any upper body injury, not just injuries that require ulnar collateral ligament reconstruction surgery, we are choosing to include cumulative pitches thrown in our data set to see if it has predictive power in other cases. The other takeaway from this paper is the risk factors they did identify as being predictive.

These included a lack of rotation both at the pitcher and arm level. Due to the detail of the Statcast data, we have a large amount of data on the position of a pitcher's body when they release the ball. We plan to use this release data to recreate some of the rotational features that were assessed as high risk in this article.

Another recent study done by Davis [6] focused on predicting pitcher injuries but did so by operating mainly on the notion that pitcher injuries are mostly influenced by the pitches that any given pitcher throws. For example, the author found that in four very successful major league pitchers, two of them being Gerrit Cole and Corey Kluber, when their sinker or cutter usage significantly increased they had resulting upper body or arm injuries in the same season or in the year following the season when the usage rate of those pitches spiked. This is interesting as a general concept as we are hoping to spot injuries on a more detailed basis using the advanced measurements of these pitches such as horizontal break, pitch positioning, etc. in order to not only predict that an injury could occur in the next year but hopefully in the next few weeks or days.

The paper also covered injury prediction based on pitcher age and performance. When it comes to performance the study found that the subset of injured pitchers in the data won 43 percent of their games while the healthy pitchers won only 36 percent of their games. We think this idea essentially speaks to the idea of usage. Pitchers who are performing well will be used at a higher rate which leads to higher risk of injury. Pitchers who are performing poorly, especially the bottom 25 percent of poor performers, will not be in games long enough in most cases to tire their arm to the point of possible injury. Age did not end up being much of a factor in the analysis; the average injury age was 27 and the average age of an MLB player in 2018 was 28.1 so there was not much of a difference in that regard. As we get into our feature building and model testing it will be good to keep the features that have been tested in previous analysis in mind in order to select features that are most predictive and not waste too much time looking into features that may not carry any weight in the model.

In summary, previous approaches have frequently tried to predict the rates that pitchers will have injuries or surgeries as opposed to predicting whether or not an injury will develop on an individual level. If predictions were made on an individual level, they were usually made on the season long level and not on a daily basis as we are proposing to do. One reason for this is that predicting rates is in some sense a more straightforward task. You are able to predict on an aggregated group level and try to explain why that group gets injured at a higher or lower level than other groups. Predicting individual players' injuries is not written about as often, and we are aware of the challenges that we will face in developing a model to do this. Even though injury prediction of pitchers at an individual level might prove to be difficult, we believe there is a lot of value in attempting to solve this problem and developing a base model that can be built off of.

V. Data Science Solution Framework

The following framework outlines the steps taken to formulate our data science solution to this problem.

A. Data Collection

To start our analysis of MLB pitcher injuries we focused on collecting Statcast data to help predict injuries and see if we can spot any trends in a pitcher's movement, release, velocity etc. that might signal injury. We started collecting data by defining the pitchers that we wanted to analyze from the 2015-2020 MLB seasons. First we looked at the historical injured list data that Spotrac [4] provides. We narrowed our selection down to only starting pitcher upper body injuries and scraped all data from 2015-2020. We chose not to include relief and closer pitchers because they are often in the game for a maximum of two or three innings each game and thus do not generate sufficient data to be predictive. Starting pitchers are in the game the longest and in most instances therefore provide the most value to their team because they will be the ones tasked with preventing the most runs being scored, and often are higher paid players as well. Limiting the analysis to upper body injuries is useful because upper body injuries are more likely to be caused by the actual act of pitching and not external factors. For example, pitchers can be injured due to a ball being batted back at them and getting hit, but that type of injury prediction is outside the scope of this study.

The Spotrac [4] data was the least clean data set that we dealt with so there was a fair amount of data cleaning required in order to merge it to our more descriptive data set which is the Statcast data. We found a total of 324 instances of starting pitchers with upper body injuries from 2015-2020 from Spotrac; some of those 324 had repeat injuries and a second unrelated injury which contributed to the large amount of data cleaning needed. Next we knew that we needed to find a sizeable amount of pitchers with significant time played that were not injured in order to hopefully spot differences in the Statcast data between injured pitchers and healthy pitchers. In order to do this we searched FanGraphs [5] leader board data to find all pitchers that had pitched more than 100 innings in each respective year. We found 770 instances of starting pitchers that threw more than 100 innings in the time frame between 2015-2020 and then made sure that we had no duplicates between the injured pitchers and pitchers that had thrown more than 100 pitches. This left us with 611 healthy pitchers and 324 pitchers with upper body injuries to examine at a game by game level. From here we will focus our initial analysis on these 935 starting pitcher instances and their Statcast data from 2015-2020.

After we had injury data and our healthy and injured players defined, we needed to add a Statcast lookup ID to all players in order to merge the injury data on each of the pitchers' respective pitch by pitch Statcast statistics (release point, velocity, break, etc.). We wrote some code to allow us to do this and then had a data set with both healthy and injured players from 2015-2020 along with their Statcast data.

Figure 1 shows a visualization of the type of information that is available in the Spotrac injury data. The bar graph displays the number of upper body pitcher injuries on each team in from 2017-2020, along with whether the pitcher was on the shorter 10-day injured list (gray) or the longer, 60-day injured list (blue).

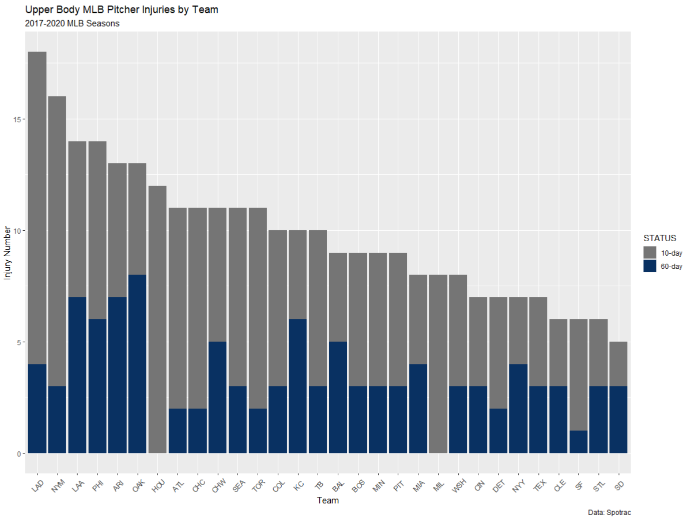

Fig. 1. A breakdown of upper body pitcher injuries by MLB team. The Los Angeles Dodgers, New York Mets, and Los Angeles Angels were most impacted by the injuries that we have attempted to predict and prevent in MLB seasons from 2017-2020.

B. Feature Engineering

With the combined Statcast and injured list data, the next step was to create and select features that were important and predictive of player injuries. The Statcast data returns one row for every pitch thrown in a game by the pitcher, so the size of the entire data set grows large very quickly from the data collection phase. For example, we had 21,413 player-games in our data set, so the original size of the Statcast data, assuming each pitcher throws an average of 80 pitches per game, was around 1,700,000 rows. The first step in the feature engineering phase was to aggregate all pitching stats to a per-game level bringing it back down to the 21,413 rows. Individual pitch variability is so high that a single pitch will not be predictive of an incoming injury, but game level trends are much more likely to contain valuable information about how a pitcher is performing. The following is a sampling of the features that were created by the game level.

- Mean Four-seam Fastball velocity
- Game pitch count
- Horizontal pitch movement
- Vertical pitch movement
- Pitch zone
- Cumulative season pitch count
- Mean release position along the x-axis
- Mean Four-seam Fastball release spin-rate

With these features created and more, the next data set that was created was a five-game rolling average of these statistics. The idea behind this data set was to get a baseline value for these statistics that an individual game performance could be compared to. For example, if a pitcher's Four-seam Fastball velocity is 93 miles-per-hour in a game, but their previous five game average Four-seam Fastball velocity was 95 miles-per-hour, that is potentially valuable information for the model when predicting whether the pitcher has picked up an injury and their velocity is suffering because of it. Without a reference point, we anticipated it would be difficult for the model to discern changes in the pitcher's performance, especially because different pitchers might throw at different natural velocities.

The game level aggregation data set and the rolling average data set were then merged together, and the final features for the model were created. For most of the main statistics, a percentage change variable was created for the difference between that game's stat value and that stat's average value for the previous five games. In this way, the data set is a collection of variables that show how the selected stats in Game X compare to the value of those stats in Games x-1, x-2, x-3, x-4, and x-5. Percentage change was chosen as the medium for most of these stats because pitchers might naturally throw at different velocities and different stats are not all measured in the same unit. We also included some of the original raw numbers, such as mean fastball velocity, in the final data set in order to test how predictive they are on their own.

Due to the coding of the feature engineering function, the five game rolling average and the stats selected could easily be changed if it was found that other values are more predictive in the process of working on this project. Five games was initially set as the value because it roughly equates to a month of starts for a standard pitcher, and this turned out to be a good value for our predictions to use.

There were many key elements to consider when building features for our model and we were aware that selecting the predictors with the most impact would have a large influence on our model. The Statcast function that we used to pull data returns 78 different features that all describe what transpired in an at-bat on a pitch by pitch level. We had a pretty good idea from testing the model and from general baseball knowledge which features should carry the most weight in injury prediction, but there could always be a key element that might be missed in feature prediction and engineering. We found that some of the features we thought would be very predictive were not and some that we didn't think would be too important turned out to greatly impact the model. We discuss the results in more detail in the Model Assessment section.

Another element of injury prediction that we spent time thinking about is how many days into the future we should be classifying a pitcher as at-risk for injury. From a useful standpoint, the farthest possible amount of time is best, because that means we were able to predict the injury well before time on the Injured List was required. This early intervention would be best for the health of a player. But from a practical standpoint, there is a limit to how far out injuries can reasonably be expected to be spotted with a machine learning model. We initially decided on trying to predict whether an injury would happen within five days or not, but moved that window back to fourteen days after working on our model for a bit. We thought two weeks would be a balance between predicting something that is about to happen and something that will happen after a few more pitching starts have taken place.

We created a function that took the player's ID as an input and then return the aggregated data set for a specific season. We looped through every player's season and ID and then created a full data set with all of the information we needed and features created.

C. Model Building

The next step of this project was to begin building the machine learning model. We split out merged data into training and testing sets using an 80 percent training data and 20 percent testing data split and then used a supervised machine learning approach to classify the pitcher behaviors that were predictive of pitcher injuries. We used many different types of models throughout this process such as logistic regression, Random Forest classifiers, Support Vector Machine Classifiers, and XGBoost classifiers.

Throughout the model building process we used 5-Fold Cross Validation to evaluate the performance of our injury prediction model, select features, and tune hyper-parameters in the models that allowed it. The output we were predicting was whether or not a pitcher would be injured within fourteen days of a game or not. This was a binary (1 or 0) prediction problem based on their Injured Lists status – 1 if they were on the Injured List and 0 if they were not. The predicted probability, not just the final class, was also important to us because we wanted to examine the relative risk players were facing when it was predicted that they would be injured within fourteen days of a game.

When we were training our models, we were evaluating the F1-Score of our predictions in order to select the best model. We chose F1-Score because we were dealing with such unbalanced data in this classification problem. Around 96 percent of our data was of the 0 class, a pitcher not being injured within fourteen days of a game date, meaning that we could have achieved about 96 percent accuracy just by predicting that every pitcher would not be injured within fourteen days. Of course, this would not be helpful to any team, so accuracy was not a good metric for us. The formula for F1-Score is:

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{1}$$

where

$$Precision = \frac{\sum_{(} TruePositive)}{\sum_{(} PredictedConditionPositive)} \tag{2}$$

and

$$Recall = \frac{\sum_{(} TruePositive)}{\sum_{(} ConditionPositive)} \tag{3}$$

The F1-Score produces a balance between precision and recall which makes it a good choice for our evaluation metric with unbalanced data.

As previously mentioned, the data was overwhelming in the "Not Injured" category, so we had to address this issue in the beginning. We thought of several methods for dealing with this issue, including ignoring it entirely and hoping our features alone were strong enough to separate the data and also oversampling the injured group in order to create a more balanced data set. After further research [10], we came across a technique that changes the threshold for categorizing an instance into the positive or negative category. The default threshold for every model we were evaluating was 0.5, meaning that the model would only classify a player into the injured group if they had a 50 percent chance or higher of becoming injured within the next two weeks. Due to the nature of our problem, this was an unrealistic threshold. If a player really did have a 50 percent or greater chance of being injured, it's likely they are showing extreme signs and they are probably already on the Injured List or at least not pitching anymore. Injuries are a rare event, so we would not expect to see any model have that much confidence that an injury was forthcoming. We evaluated different threshold levels throughout our model building process and ultimately selected a much lower threshold in our final model that resulted in the best possible F1-Score on our testing data. The original threshold of 0.5 would have produced poor results so this was a necessary modification to make for this type of classification task.

With five years of Statcast data, we had a good amount of pitching data to build our model on. Figure 2 shows a visual of our solution framework for this project.

As shown in Figure 2, we went through an iterative process in building our models after combining the Statcast and Spotrac data and creating new features. We trained a model, evaluated its performance, and then compared it to other models. We would also add features to models and remove features to see what would result in the best performance. Ultimately, using XGBoost resulted in the best performance by far, so we chose that to use for building our final model. Both Logistic Regression and Random Forest models performed similarly to each other, and Support Vector Machine performed slightly worse. We spent the rest of our time focusing exclusively on XGBoost after ruling out the other model types.

D. Model Assessment

After the XGBoost model was selected, another round of hyper-parameter tuning was done and the best parameters were selected. We then split our model once more into training data (80 percent of our data) and testing data (20 percent of our data) to get final results for its performance on unseen data. Table I shows the confusion matrix of the predicted and

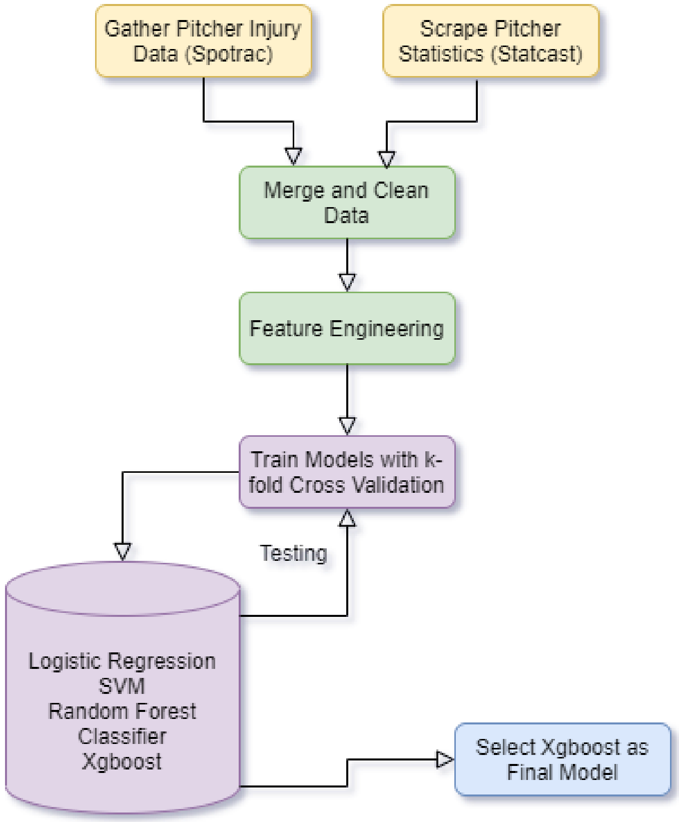
Gather Pitcher Injury Data (Spotrac)
Scrape Pitcher Statistics (Statcast)
Merge and Clean Data
Feature Engineering
Train Models with k-fold Cross Validation
Testing
Logistic Regression
SVM
Random Forest Classifier
Xgboost
Select Xgboost as Final Model

Fig. 2. Our data science solution framework.

TABLE I
XGBOOST CONFUSION MATRIX.

Actual Class	Precision	Recall	F1-Score	Support
Not-Injured (0)	0.97	0.90	0.93	4103
Injured (1)	0.12	0.31	0.17	180

actual results. The best model we created resulted in an F1-Score of 0.17 which we were very happy with. Predicting pitcher injuries in this way has not been done before and we knew it would be a difficult task, but the F1-Score showed that the model was able to identify features that separated future pitchers who would be injured from those who would remain healthy.

Putting the numbers in context, if our model predicted a pitcher to be injured within the next two weeks, they wound up on the injured list 12 percent of the time. Conversely, when our model predicted a player would not be injured within the next two weeks, they only did end up becoming injured about 3 percent of the time. That means that pitchers our model identified as likely to be injured were over three times more likely to be injured than pitchers who were predicted to not be injured. Out of all pitchers that did get injured within two weeks, our model correctly identified 31 percent of them as injured. These were very promising results for a first attempt at building such a model and they showed that pitcher injuries could be predicted on a per-game basis during the season. From a practical standpoint, seeing a higher recall than precision was better for the use case of this model. We would rather identify more pitchers who are going to be injured than have an extremely high precision on the players actually getting injured. Expert doctors and team personnel would ultimately be making the final evaluation and decision on a player's status once they were identified by the model, so we wanted to return as many high probability candidates for injury as possible.

XGBoost also allowed us to examine which features were the most important for predicting whether or not a pitcher would be injured within the next two weeks. Figure 3 shows the plot of feature importance generated by the XGBoost model.

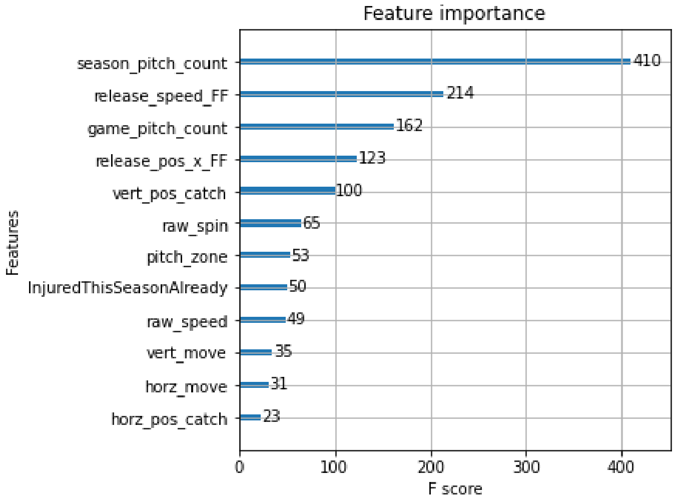

Fig. 3. Feature importance plot.

The most important feature by far was the cumulative season pitch count. This made intuitive sense to us but it was interesting in that it went against previous research [3] that had identified cumulative stats such as pitch count to not be predictive at the Major League Baseball level. It was found to be predictive at lower levels of baseball, however. We think it was predictive in our model for two reasons. First, the previous work only looked at injuries that required ulnar collateral ligament reconstruction surgery in pitchers. We looked at all upper body injuries in this study, so that could be one point of difference between the two works. The second reason is that we were predicting injury probability after each game, while the previous research was summarizing the results from a body of previous work, not implementing a prediction model themselves. There might be cases where cumulative season stats are not predictive, but in our case it was highly predictive.

The second most important feature was the percentage change in a player's average Fourseam Fastball velocity in the game compared to their five game rolling average. This was one of the features we created so it was nice to see that it held predictive power and was the second most important feature in the model. Release position percentage change, another feature we created, was the fourth most useful feature for the model. Players who suddenly start releasing the ball from a different position along the x-axis than they usually do are a good candidate for injury. This might be a sign that the arm is getting weaker and thus the player cannot use the same motion they are use to, or that their original motion is causing them discomfort. Without looking at the data, it would be hard to evaluate the release position of the ball visually because the changes are so small. This is a great example of how data can help evaluate pitcher injuries better than the eye can alone.

Game pitch count, or the number of pitches a player threw in a given game, was the third most important feature of the model. This provided some evidence that players can be overused in a single game and not just over the course of the season. It might not be worth it to leave your starting pitcher in a game longer if they are doing well, but that is exactly what managers are prone to doing. This is particularly true if a player is throwing a "No-Hitter", where no opposing batter has hit the ball into play. It is considered a great achievement if a pitcher can stay in the game the entire nine innings without letting the opposing team get a hit, so managers are unlikely to take a pitcher out of the game and deprive them of that opportunity. After all, if a player is doing well, why take them out? One reason could be to prevent them from overusing their arms and putting the rest of their season in jeopardy for a single highly productive outing.

One feature that was not very predictive that we had anticipated being predictive was whether or not a pitcher had already been injured this season before. This feature was called "InjuredThisSeasonAlready" and was a 1-0 flag with a 1 if they had been injured earlier in the season. We created this feature thinking that a player who had been injured already would be more likely to be hurt again, but that wasn't as true as we thought it would be. This does shed some light on another potential discussion point of pitchers being more prone to injury than others. If we were managers of a baseball team, we wouldn't be as worried as before about a star pitcher getting injured for a second time in one season. That previous injury doesn't seem to be extremely predictive of getting another injury.

E. Model Discussion

As mentioned previously, we were very happy with the performance of the model. There were many interesting and good predictions the model made that we looked at, but perhaps the best example of how it could benefit players and managers alike is with 2018 Oakland Athletic, Brett Anderson. On May 7, 2018, still early on in the 162 game baseball season, the model predicted Brett Anderson had a 19 percent chance of being injured within the next two weeks. This was well above the final threshold we set to classify players as injured, and it was among the highest probabilities predicted for players in our data set. Most players, even if they are going to be injured, never get predicted with over a 40 percent chance of being injured because of how rare it is. At 19 percent, there is definite concern and the model classified Anderson into the injured group. Anderson would go on the injured list 12 days later, after playing two more games as starting pitcher. He would not pitch again until July 8, 2020 due to a shoulder injury. If our model had been used by the Athletics at the time, it's likely Anderson would have been evaluated after the May 7 game, and it's possible they would have noticed what the model indicated from a possible injury standpoint and not started him again so soon. There's a chance Anderson would still have had to spend time on the injured list, especially if the injury was already beginning to take form in the shoulder, but we are sure it would have been a better outcome than forcing him to pitch two more times on an injured arm. That could not have helped his recovery time and perhaps he could have spent less time on the injured list.

Overall, we see the model as being used by teams to flag players who need further evaluation. We don't think it could effectively be used to be the final decision maker in whether or not a pitcher should take time off or not. Not only would a training staff never trust a computer alone with this kind of important decision, but in its current state, the model simply isn't good enough to make that decision either. Physicians could evaluate players who are categorized as at-risk by the model and then make a recommendation about what could be done as a medical professional. The fact that almost one-third of pitchers who would be injured were identified by the model is very encouraging and means that almost one-third of players who will be injured could be evaluated by a team physician, begin preventive care, and take steps to ensure they don't have to spend additional time on the injured list. No physician could watch every pitch of every game and make a list of players who need to be evaluated, so this model makes that job easier and shortens the list of candidates to be evaluated.

Despite the good initial results from the model, there were some additional improvements that we would like to see incorporated into additional work on this topic. First, we would like to see some additional features included that might help increase the predictive power of the model. One area we had planned to work on if more time allowed was the feature engineering of off-speed pitch statistics. Not every pitcher throws the same type of pitches outside of the fastball, so it's difficult to include information about other pitch types without having to deal with a lot of null values for each pitcher. We would have liked to include them in some fashion and we feel future work could take up this problem in order to have a more robust model. One possible solution would be to create features about a player's second most used pitch after the fastball and ignore the specific type of pitch it is.

Second, we would like to see relief pitcher data added into the model. We focused only on starting pitchers but relief pitchers are another important part of the game that should be studied. A separate model might have to built for this type of pitcher, however, as their total pitches and velocity are going to be quite different than the average starting pitcher. They throw fewer pitches per game but usually throw a lot faster than starting pitchers because of that. A separate model for starting and relief pitcher injury prediction would probably be the best approach for this situation.

Finally, we would consult with a baseball team physician in the future to better understand pitcher injuries from a medical standpoint and what factors they have seen that cause them. We do not have medical degrees and thus were using our own research to determine what features would be most predictive in predicting upper body pitcher injuries, but someone with more domain and medical experience would be a great asset to the team. Statcast tracking cameras are already in every MLB stadium, so we also thought that they could be used to capture movements and data that physicians would deem important to injury prediction. Essentially this would improve the source data with injury-specific features that we could then use in the model.

There is a lot of exciting work that is still yet to be done and we think great improvements could be made in the future to this work.

VI. Implementation Plan and Milestones

The following is a discussion of the milestones we set in advance for the project as well as how well we achieved them. This is followed by a breakdown of how each member's time was spent working on this project and the tasks each member worked on.

A. Milestone Reports

By Week 6 of working on this project, we had planned for the following milestones to be achieved.

1) Exploratory data analysis completed
2) Data and scripts are working appropriately on the class cluster
3) Framework for our machine learning model is outlined

The first and third points were achieved on schedule. The exploratory data analysis phase was ongoing to some degree throughout the project, but the initial analysis was complete by Week 6 and helped us choose the variables to start with when we were building the model. The framework for our machine learning model was outlined with a clear response variable and an understanding of the data that will be inputted into the model. We narrowed down our model choice to a few that had the potential to work well, all of them being classifiers due to the nature of our predictions problem. We initially started with simple logistic regression before moving on to Support Vector Machine, Decision Trees, and finally XGBoost as our models of choice. At this point in the project, we also noticed that we would likely be able to run our models on our personal computers, so we postponed the second milestone of running scripts on the class cluster. Ultimately we would have liked to use the cluster more, but did not need it to complete this project.

Our Week 9 milestones were:

1) Implemented a machine learning model
2) Evaluated model performance on unseen data
3) Created a plan for final weeks of project

At Week 9, we had completed all three of the steps we had planned for this stage of the project and were on schedule. We had all of our data collected and began to implement machine learning models starting with Logistic Regression. We routinely tested new models that we had chosen earlier with additional features and evaluated their performance using K-Fold Cross Validation. The one issue we ran into around this time was that the data was extremely unbalanced as most people do not get injured, so it was difficult to generate models with a high predictive accuracy. When planning out how we were going to use our time until the end of the project, we knew we would want to dedicate a good chunk of it to experimenting with how to best predict data that is extremely unbalanced. We planned to learn more about the best models for predicting unbalanced data and then implement them to hopefully improve on the performance of logistic regression. We were happy with the progress we made at our Week 9 milestone and accomplished all of the goals we set out to at that point.

B. Team Member Time Breakdown

Both team members worked hard on this project and collaborated well together. Overall, the breakdown of time spent on the project was divided into the following categories:

1) Feature Engineering: 30 hours
2) Data Cleaning: 25 hours
3) Modeling Building and Testing: 20 hours
4) Report Writing: 10 hours
5) Data Collection: 5 hours

This worked out to roughly 50 hours of work per team member over the course of the semester.

By team member, Drew worked mostly on the model building and testing and contributed to feature engineering. Mallet worked more on the data collection and cleaning and also worked on feature engineering. Both team members split the work evenly when writing reports and making presentations.

VII. Conclusion

As shown in this paper, a machine learning model that can predict pitcher injuries before they happen can be successfully designed and created. Using data from 2015-2020 and an XGBoost Classifier model, we were able to achieve an F1-Score of 0.17 and classify pitchers by whether or not they would be injured within two weeks of a game date. This model is beneficial to players, managers, and medical professionals alike. Technologically, we have shown that injury prediction at the game level is possible and that Statcast data can be successfully used in these prediction efforts. Previous research has tried to identify the factors that contribute to injury or predict injury at the season level, so our work at the game level is a new research area that has not been tackled before.

With this work as a starting point, we believe that both of us and other researchers can expand the capability of pitcher injury prediction based on what we have already learned. We outlined future research directions such as incorporating off-speed pitch statistics and consulting with baseball team physicians about key factors that cause injury and how to better use the data at hand.

Pitcher injury prediction is a field that has many areas for more research to be done and we hope this paper has shown that it is an area where better predictions can be made and that time and energy will be well spent. Teams that take advantage of this model and future ones will be at a competitive advantage because they will be able to keep their pitchers healthier and have them spend less time on the injured list. Early intervention for injuries is incredibly important and this paper has shown that that can be successfully done using data provided by MLB Statcast.

REFERENCES

[1] Statcast, *https://baseballsavant.mlb.com/statcast_search*, Accessed: September 13, 2020.

[2] Chalmers PN, Erickson BJ, Ball B, Romeo AA, Verma NN. Fastball Pitch Velocity Helps Predict Ulnar Collateral Ligament Reconstruction in Major League Baseball Pitchers. *The American Journal of Sports Medicine*. 2016;44(8):2130-2135. doi:10.1177/0363546516634305

[3] Erickson, BJ, Chalmers, PN, Bush-Joseph, CA, Romeo, AA (2016). Predicting and preventing injury in Major League Baseball. *Am J Orthop*, 45(3), 152-156.

[4] Spotrac, *https://www.spotrac.com/mlb/disabled-list*, Accessed: October 1, 2020.

[5] FanGraphs, *https://www.fangraphs.com*, Accessed: October 3, 2020.

[6] Davis, K. (2019, March 11). Predicting Injuries in MLB Pitchers. Retrieved from https://towardsdatascience.com/predicting-injuries-in-mlb-pitchers-c2e133deca39

[7] Stephanie. (2017, October 31). Brier Score: Definition, Examples. Retrieved October 26, 2020, from https://www.statisticshowto.com/brier-score/

[8] Lindbergh, B. (2016, April 26). Statheads Are The Best Free Agent Bargains In Baseball. Retrieved October 26, 2020, from https://fivethirtyeight.com/features/statheads-are-the-best-free-agent-bargains-in-baseball/

[9] (Image) Retrieved from https://scnow.com/sports/baseball/professional/clevinger-replaced-on-padres-playoff-roster-with-elbow-issue/

[10] Brownlee, Jason, (2020, August 28). Retrieved December 13, 2020, from https://machinelearningmastery.com/threshold-moving-for-imbalanced-classification/