Capstone Project 2 Milestone Report

As summarized in my proposal write-up, there have been numerous studies on the need to cut back on a pitcher's pitch count to reduce stress on the arm and avoid injuries. The combination of pitch count limitations, frequency of playtime, and the fact that pitchers get paid just as much, or sometimes more than, as everyday players, the need to optimize how effective they are to get outs could be beneficial to how general managers manage their teams payroll and get, so-called, "as much bang for their buck". Therefore, the aim of the project is to uncover batting tendencies and use that information to the pitcher's advantage to get outs.

The data used for this project comes from Major League Baseball Gameday as it offers pitch-by-pitch data dating back years and is extremely detailed in JSON format making it easy to pull into Python. The downside is that each game is its own separate webpage with a unique game ID number. Luckily after some research, the game ID's were in a sequential order and all that was needed was the first and last game ID of all the seasons of interest. For this project, the last three regular seasons (2015-2017) were chosen to provide ample amounts of data for various situations and matchups. Along with this data though was the MLB All-Star Game which is played halfway through each season featuring the best players at the time. I didn't want the stats of these games to affect the analysis or model so it was a matter of finding what the game ID's were for these games and removing it from the dataset. I also didn't want to include playoff games because situations are played differently and more preparation is put into these games.

A Python script was coded to run through each web page containing the JSON pitch-by-pitch data by game ID and save it to a larger JSON file indexed by game ID. Each season was saved into its own JSON file to reduce file size and time when running it through another Python script for further wrangling.

Once the data was collected, it was a matter of extracting the necessary information embedded in the JSON files and removing plays that didn't involve a pitch such as pick-off plays. To automate the process of extracting all the necessary information, the function **get_pitches_data.py** was created. It takes the raw JSON files and creates a row of observations to be later converted into a pandas dataframe and exported to a CSV file.

The function works by opening the JSON file that houses all the game data for an entire season. An empty array is then created to house information about each pitch where each element is its own separate pitch.

Within each game of the JSON file, a summary of the matchup is stored where it's further broken down to pitch sequence. It was noticed early on that some of the data, like count, was recorded as a result of the pitch instead of preceding it. Therefore, the count had to be initialized, updated, and reset as matchups played out. It will only be updated if the next play was labeled as type "pitch" and if it was, multiple fields needed to be extracted. Data such as inning number, which half inning, who was batting, who was pitching, their handiness (which side their using to bat or pitch), their names, and player ID numbers. Then details about the pitch thrown was also extracted like pitch type, pitch speed, coordinates of the pitch location, and the result of the pitch. For some instances, the data wasn't consistent for most cases so it was a matter of finding where there were embedded or disregard them and insert a null entry. Again, this was done for every pitch in every game for the 2015-2017 MLB regular seasons and exported into a CSV file.

Once the data was converted into CSV format, it made it easier to visualize and manipulate in pandas. After importing it into pandas, it immediately became obvious that many outcomes of the pitch, or calls, were redundant for purposes of this project. For example, a normal ball was labeled "B" whereas a pitch thrown into the dirt was labeled as "*B" which is

also technically a ball so it became a priority to simplify the calls and their codes to make filtering by an outcome easier and less complicated. What remained were six possible outcomes of a pitch: called ball, called strike, foul ball, hit, swinging strike, and an out.

With the outcomes simplified, missing and null values need to be handled which only appeared in pitch type and pitch speed columns. It's possible to average out the speed of certain pitches and fill in the missing values with the averages or based on the speed of the pitch but some observations could have both values missing making it difficult to hypothesize which pitch was thrown. Fortunately roughly 4,600 values were missing out of over 2 million so removing those data points wouldn't impact the data or statistics to heavily. The hope for this project is to eventually pass this data through a deep neural network to model so all predictor variables need to be numerical.Therefore variables like the outcome call and pitch type needed to be converted to dummy variables to satisfy that requirement.

The last part of data cleaning that was necessary before modeling was dealing with pitch location. The data provided precise locations as to where and how high the pitch crossed the strike zone. If we were to pass in the data with precise pitch coordinates, it may become difficult or over complicated to use in practice. In reality, pitchers aren't typically precise enough with their pitches to pinpoint where they want to throw their next pitch. Instead, it may be better to generalize locations of the strike zone into zones and it started with finding, or determining, the edges of the strike zone.

There are rules for determining where the strike zone is for a specific batter. The width of it is the width of home plate but the height is based on the batter. It varies, but the general rule is from the height of the batter's knees to midway up the torso but ultimately, it's up to the umpire to call balls and strikes. So instead of getting player heights and averaging them out to get the height of the strike zone, I plotted the distribution of called strikes from all three seasons.

The resulting plot had some unexplainable outliers but the plot was more or less grouped together. After confirming the distribution of pitches by axes for normal, two standard deviations from both the height and width distributions were taken to be the edges of the strike zone. That's because two standard deviations represents 95% of the data within a normal distribution so of pitches that were called strikes, the height was determined by the height of 95% of called strikes and same for the width. The bounds for the strike zone were then used to split the strike zone into 9 equal parts and 8 areas outside it to be considered balls. Just like calls and pitch types, these zones would then be converted into dummy variables.

As stated earlier, the goals for this project is to find patterns in batters for pitchers to utilize and possibly get outs quicker. There are many idioms and theories on how to pitch to disrupt a batter's timing and rhythm. One phrase is to pitch "hard stuff in, soft stuff away". This means to pitch hard, fast pitches inside the zone towards the batter while throwing softer pitches with more movement away from batters. It's said that this causes disruption in the batter's timing and creates deception in the pitches. Another theory is that breaking balls up are often a mistake which result in a hit, or at least contact with the ball. Regardless, are there patterns or tendencies that batters abide by?

The first thing to investigate was if breaking balls up lead to hits. Breaking balls are pitches like curveballs and sliders that bend in flight. A couple functions were created to help visualize hot spots and determine their significance. When plotting slower, breaking pitches, there didn't seem to be any outstanding patterns between left or right-handed batters and those pitches. The only highlight was the middle of the strike zone that showed the highest percentage of hits of pitches thrown in a location at just above 0.125 for both left and right-handed batters. When taking the upper two thirds of the strike zone (since it's considered up) and comparing hit rates from breaking balls to non-breaking balls, there was a statistical difference. That difference

was about 0.007 for both types of batters, so although there was a statistically significant difference in means, the practical significance seems minimal. The hit rates for pitches up was greater than all pitches thrown in the strike zone so there is some promising results but further exploration should be done. Something to investigate is to factor in severity of the hit. For example if all the hits that came off high breaking pitches were home runs and non-breaking pitches produced singles, the result of high breaking pitches would have a lot more significance.

The second thing that was investigated was the expression often told to pitchers to pitch “hard stuff in, soft stuff away”. Hard stuff refers to pitches that are typically faster while soft stuff are pitches that are slower and tend to have more movement. Looking at hard stuff in, the visualizations show highlighted zones on the inner third of the strike zone to batters. When comparing the average rate at which hard pitches get outs, there was a statistical difference of about 0.03 for right-handed batters and 0.01 for left-handed batters. That’s not much of a difference practically that it may not make a difference when pitching inside to a hitter. However when you combine it with soft pitches away, that average could go up. When looking at a heatmap of soft stuff away, there’s an obvious pattern of outs for pitches thrown down and away from a batter. Statistically, soft pitches down and away produced an out 0.08 and 0.09 more than non-soft stuff to right and left-handed batters respectively. Not only is that statistically significant, but in a game where a 0.05 difference in your batting average can make you great or average, I would say that’s practically significant as well.