



### **7th Inning Stretch**

*There is no luck in a winning strategy*

#### **Team 11**

April 29, 2022

Advisor: Santosh Venkatesh  
[venkates@seas.upenn.edu](mailto:venkates@seas.upenn.edu)

Team Members:

Adit Arora

[adita@seas.upenn.edu](mailto:adita@seas.upenn.edu)

Systems Engineering and Finance

Parth Daga

[padaga@seas.upenn.edu](mailto:padaga@seas.upenn.edu)

Systems Engineering and Operations

Cal Rothkrug

[chroth@seas.upenn.edu](mailto:chroth@seas.upenn.edu)

Systems Engineering and Math

Durga Srivatsan

[dsri@seas.upenn.edu](mailto:dsri@seas.upenn.edu)

Networked and Social Systems Engineering

**Table of Contents:**

<b>Executive Summary</b>	<b>3</b>
<b>Overview and motivation of the project</b>	<b>3</b>
<b>Technical Description</b>	<b>4</b>
<b>Self-learning</b>	<b>5</b>
<b>Ethical and Professional Responsibilities</b>	<b>5</b>
<b>Meetings</b>	<b>6</b>
<b>Schedule with milestones</b>	<b>6</b>
<b>Discussion of teamwork</b>	<b>7</b>
<b>Budget and justification</b>	<b>7</b>
<b>Standards and Compliance</b>	<b>7</b>
<b>Work done since last semester</b>	<b>8</b>
<b>Discussion and Conclusion</b>	<b>8</b>
<b>Appendices</b>	<b>10</b>

## **Executive Summary**

Our project, 7th Inning Stretch, is an online web application that allows users to access predictions for baseball games throughout the MLB season. Inspired by the trials and tribulations of recreational sports bettors, we aim to inform subscribers on how to make more profitable and calculated bets. The backend platform uses statistical analysis, machine learning, and confidence testing to predict the final score of a baseball game given both teams' starting lineups and pitchers. The final frontend website includes an interface that allows users to select both game predictions as well as expected at-bat outcomes between any given batter and pitcher.

Our model creates a simulation of a full nine-inning baseball game based on the at-bat model of given batters and pitchers. Our model is able to predict correctly 54% of the time based on the testing dataset of the 2021 season, with predictions that have a win probability greater than 70%, the model was correct 57% of the time. The platform, which is free for all, is designed as an educational website that helps casual sports bettors to develop a greater understanding of how certain matchups play out and thereby make more informed decisions. The statistical model can also be utilized by team management for lineup selection and strategy purposes.

## **Overview and motivation of the project**

Today, sports betting is a \$3B industry with a 20% CAGR growth. Upon non-trivial investigation, it can be deduced that sports betting platforms are not inherently sophisticated at sports predictive capabilities, and rather rely on a "vig" or a cutoff where sports betting spreads are set in order to minimize net payout to bettors, rather than accurately representing the predicted outcome of the game. Therefore, when amateur bettors follow sports betting books and odds, they do not have all the information to make an informed decision about where to place their money. In an industry that preys on people's insecurities and lackluster knowledge, we are hoping to provide a solution that will give power back to the individual. We believe our product has a strong capability to aid consumers as they navigate the sports betting world.

7th Inning Stretch is an interactive platform created to predict the outcome of baseball games. We are able to predict individual batter vs. pitcher at-bats as well as simulate a game. Our solution helps sports bettors understand all aspects of the game and be able to select both individual and team stats in order to better understand the predictions. Bettors would no longer have to rely on misinformation provided by various platforms and can instead come to their own conclusions based on real facts and data. Beyond just bettors, we provide key insights to baseball franchises who will benefit from advanced analytics that will help them create better teams that can be even more successful. Coaches will be able to understand how to have a better functioning team, while scouts can recruit players that will help strengthen the overall team. Sports entertainment companies can utilize our product to give more concrete evidence to various analyses that they conduct on air.

## Technical Description

For the product itself, one of our top priorities was ease of use for the user, which is why we decided to construct a user interface in which the user could simply select the teams that would be playing in an anticipated game or players that would be facing off against one another. We wanted the model itself to output a simple and understandable result that would cater to what the user was interested in learning, such as the score in a game, the accuracy at which it predicted said score, and a confidence interval to alert the user to the limited reliability of the result.

For the development of the model portion, our goal was to identify the optimal method to predict the outcome of any given MLB game based on readily accessible open-source data. Our first step was to acquire said data, which we did via web scraping using the BeautifulSoup library in Python. We obtained individual statistics for all of the pitchers and batters of each team since 2014. Because our code already took a while to communicate with our data on Google Cloud, we knew the computational efficiency of our feature selection method would not have too much bearing on our current process. We also knew that we weren't interested in any of the intrinsic properties of any of the data columns we were inputting to our code, hence this led us to use a recursive feature elimination process (RFE) to select the relevant columns for our model. Using said selected data from the respective batter-pitcher pairings, we were then able to simulate an entire baseball game, with the outputs being either our prediction on the outcome of a batter vs. pitcher matchup or the outcome of a game with a confidence interval on our prediction. Our model is able to predict correctly 54% of the time based on the testing dataset of the 2021 season, with predictions that have a win probability greater than 70, the model was correct 57% of the time.

For the website of the project which was the final output, we used React as the frontend, Node.js as the backend, and Google Cloud Platform as the database (see Appendix Figure 1 for a chart of the workflow). We created 4 different pages (see Appendix Figures 2-5) for the website that included the homepage, teams, individual, and the disclaimer. The homepage and the disclaimer both are just information of the website but the teams and individual pages display the data generated from the model. The teams page takes 2 teams and generates the final score for each team, a win probability, and an accuracy score of the model. The individual page takes 2 players and generates a histogram of how the at-bat between this batter and pitcher.

Overall, our project was able to create a simulation of a baseball game with both confidence intervals and accuracy in order to have a product that casual sports bettors would be able to use with ease. We were able to meet our goals of having a project that displayed how confident we were about the scores that were generated and providing bettors information in order to make better decisions.

## **Self-learning**

The model itself was coded in Python, which we were already well acquainted with, as well as several libraries such as Pandas, Numpy, sci-kit learn, etc. The front-end was designed in React, with which most members of our team had little to no experience with. Additionally, learning React was very tedious despite abundant resources, as the framework itself has many nuances and we had to write our code in terms of React components, which was a new concept to grasp. Furthermore, interfacing our front-end and back-end was another task we originally did not know how to undertake, hence we utilized and built upon our existing knowledge of JavaScript to implement our Node.js backend.

Classes such as CIS 110 (Intro to Programming) and 120 (Programming Languages) gave us a solid foundation on which we could learn other languages more easily than we would have whereas classes such as ESE 402 (Engineering Statistics) and ESE 305 (Foundations of Data Science) gave us both the opportunity to refine our Python programming skills through hands-on projects as well as an understanding of how to go about developing a pipeline to use so that we would know how to draw conclusions from existing datasets. CIS 557 (Programming for the Web) was also extremely helpful in creating the website.

## **Ethical and Professional Responsibilities**

The main ethical issue of our project is the misconception that our project promotes sports betting in and of itself; however, it is clearly presented in our product that we hold no association with the practice of betting on MLB games. Our resolution to this is the disclaimer on our website reiterating said dissociation from betting activities.

Our project promotes informed decision-making while at the same time supporting any inclination of our users' to engage in meticulously-calculated sports betting techniques by augmenting their knowledge and thereby making them more confident in the decisions they do make. This directly promotes the expansion of the betting market itself by offering users an informative entry, but it also heightens the users' tendencies to think twice about their betting practices and do their own research. The disclaimer behind our accuracy measurements regarding the expected unpredictability of MLB games hints that users should not simply blindly follow our product's recommendations, but they would do well to consider the probability that our measurements yield the correct directions in which to bet.

## **Meetings**

We met with our advisor, Professor Venkatesh, 3-4 times this semester in person where he gave us insights on how to progress with our project. We also met with Steven Xing who is the

Director of Engineering at TempusEx, a company built around data productization in partnership with sports leagues such as the NFL, and planning to expand into the NBA as well. Our meeting at the beginning of the semester helped us pivot our project from basketball (our sport of choice last semester) to baseball. This was due to the fact that basketball requires much more player interaction than baseball which can be boiled down to a batter vs a pitcher and therefore makes the game a lot easier to simulate based on the data.

### **Schedule with milestones**

When we switched to a baseball predictive model from our initial plan of doing basketball, we redesigned our spring milestones to be the following six milestones.

1. Gather relevant data for the past 5 years including individual player data, game by game data, player vs player data, and in-depth statistics. We achieved this objective and surpassed the amount of data we thought we could gather, which eventually led to a more refined model.
2. Process the data and use statistical learning to gather relevant features from the vast amount of data gathered. We were able to achieve this objective but had to scale back on the number of features we wanted in the model.
3. Create the base layer of our model and output probability distribution for each batter vs pitcher combination. Unfortunately, due to limited availability of data on sacrifice flies, bunts, double plays, etc. we were unable to create a probability distribution that spanned every outcome. Rather, we hit 90% of the outcomes (walk, SO, 1B, 2B, 3B, HR) and used league average probabilities to approximate the other 10%.
4. Code the game model using a simulated inning approach and the probability distributions achieved from the player vs player model.
5. Achieve an accuracy of 55% on our testing data set using the created model, reiterate over the feature selection process to achieve better results while keeping the data sets separate. At first, we achieved an accuracy of 54%, however, we went through multiple feature selection processes and were able to improve our result to 59%.
6. Present confidence intervals for each simulated game, aim for a confidence level of 70% or greater in most games. Since COVID impacted the 2020-2021 season, we had limited recent data and due to a heavy recency bias in our model, we were not able to achieve the required confidence levels in our outputs for the most recent games. On average, our confidence levels were between 60-65%, however, we expect this to improve as we gather more recent data and the season progresses with better data on newer players and rookies.

### **Discussion of teamwork**

Amongst the team, we were able to split the work in a way to play to each of our strengths. Parth and Adit focused on generating the simulation and logic behind the model, Cal focused on

actually coding the model, and Durga focused on creating the front end and connecting the model to the website. We had weekly meetings with each other to check in on the progress of the work so we would stay on track. Every member of the team contributed to the development of the presentations and reports.

### **Budget and justification**

Regarding our spending, we initially forecasted money to be allocated towards an AWS service to host our project, as well as a subscription to a sports predictive platform to cross-check our results. In the end, these expenses were not necessary. We ended up using Google Colab instead of AWS; however, Colab tends to have more bugs and is slower, so if we were to take this project further and even add automated scraping, transitioning it to AWS would likely be preferred. We were able to mitigate the cost of a third-party sports prediction platform because when we tested our results, we reached a high enough level of accuracy from our testing set that cross-checking was not a necessary step. We also learned from Steven Xing that many of these sports prediction platforms are not reliable so he cautioned us from subscribing to the service.

### **Standards and Compliance**

We realize that the goals of our project lie in improving the common sports bettor's quantitative perspective, and by doing so, threatening bookmakers' chances of profiting off of incorrect betting assumptions. Our first priority is to convey that our product does not incentivize any sort of betting practices, rather that it offers an opportunity to improve upon preexisting betting practices for those who have already committed to placing future bets. In fact, we hope that our product improves the quality of decisions sports bettors make, essentially encouraging bettors to be more responsible by using additional tools at their disposal. Additionally, we want to clarify to market participants that our product does not in any way attempt to undermine bookmakers' price discovery methods; the application itself merely offers additional quantitative tools to our users based on publicly available data scraping methods. Furthermore, the data we use is completely open-source and available to the general public, and our software does not violate any data privacy policies from any institution. Lastly, our project follows IEEE standards 1448a and 2675.

### **Work done since last semester**

Since last semester, we have made several key milestones. A big turning point came early in the semester when we were able to speak with Stephen Xing, a Penn alum who works for a sports prediction startup. At this point, our model aimed to predict basketball not baseball. Stephen was able to guide us through some challenges with basketball prediction - namely the continuous nature of the game, high variances in-game scores, and the idiosyncratic nature of

the game caused by a more individual focus. For this reason, he suggested we switch to baseball, which is discrete in nature and is much more intuitive to model, compounded by the fact that there is a higher volume of relevant statistics available for baseball that provide a cause-effect relationship that can be modeled. Given that we had already made substantial progress in scraping and analyzing basketball data, we were able to easily translate this methodology to reach that same point with baseball data.

From this point onward, we made our initial framework modeling pitcher-batter outcomes - which was the base layer of our model. This continued to be refined and iterated as the semester went on. Furthermore, we developed the program that models a given game, as well as the entire frontend over the course of the semester.

## **Discussion and Conclusion**

We were able to predict baseball games on our testing data set with an accuracy of 59% (for games with a confidence level of 70%+) and overall accuracy of 54% for all games. We expect this to get better over time as we gather more data on rookies and newer players to the game who had fewer data points in our original master dataset as well as data from games during the season.

We set up a front end to display our results, both for the base layer of our model (player vs player probability distributions) and for the game simulations (with an accuracy/confidence level), which was done to maintain full transparency with consumers on how our model was built. This website was free for all casual sports bettors.

Some challenges we faced were the inclusion of more sophisticated statistics and plays in a baseball game, such as balks, double plays, bunts, manager switching DHs, sacrifice flies, and others. In addition, we were not able to incorporate the direction a ball is hit, for example to right field or left field - and how this would impact the runners on base. Obviously, our biggest challenge was switching projects midway from basketball to baseball; however, we were able to apply the lessons we learned from the first sport to make a fully functional and accurate model for baseball.

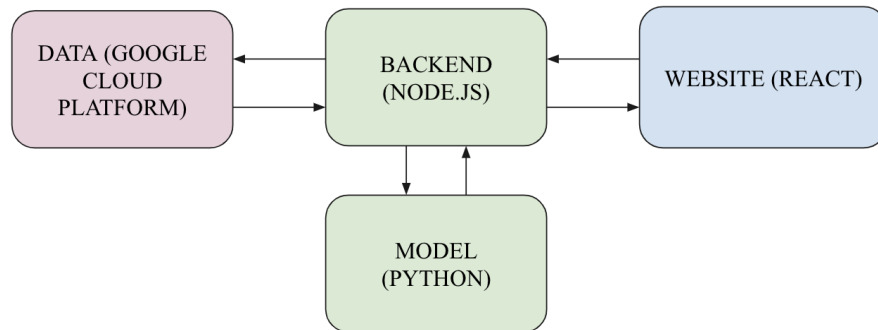
If we were to continue with the project, some of the key changes we would make are to automate the data scraping for both the lineups and per game statistics for players so that our model on our front end is automatically updated every day there are games. We would also have relied less on a single base layer for our model (player vs player), and instead simulated the game in different ways to get a more sophisticated model.

This was a very interesting project for us to work on, especially as avid sports bettors and watchers and we learned a lot about how betting websites create their odds and understood the models behind them and how they can be beaten. Making a model from scratch using publicly

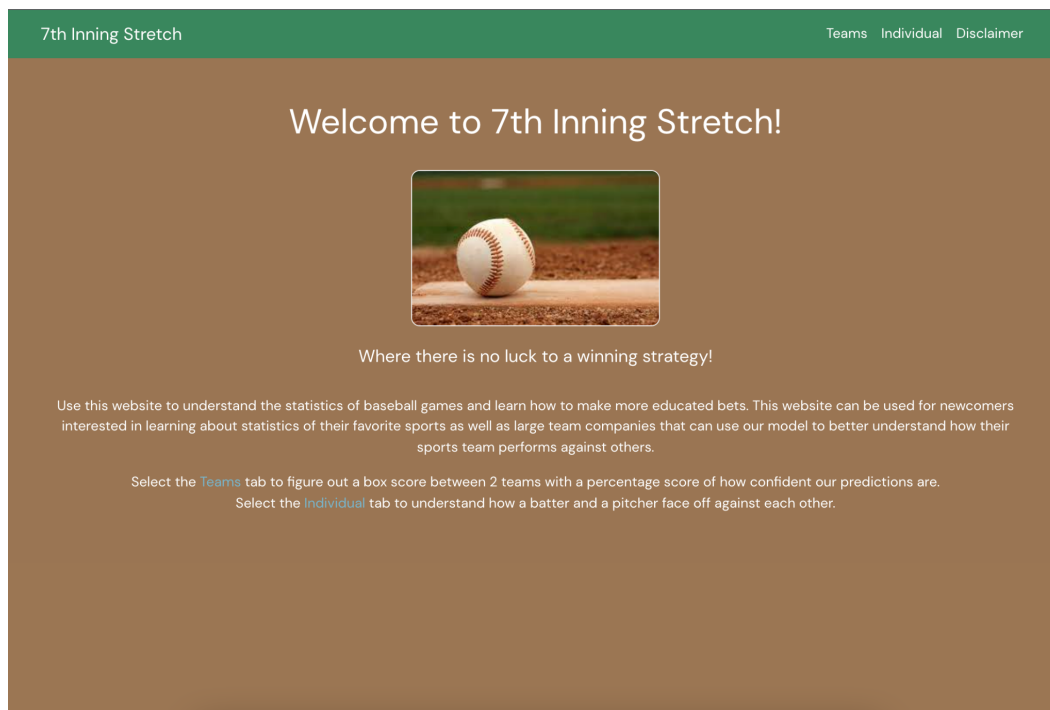


available data was a great experience to learn more about statistics, machine learning, and modeling.

## Appendices



*Figure 1: Workflow chart*



*Figure 2: Home Page*



7th Inning Stretch Teams Individual Disclaimer

### Select Teams for Box Scores

Los Angeles Angels against Pittsburgh Pirates

	Total	Win Probability
Los Angeles Angels	11	0.575
Pittsburgh Pirates	8	0.425

Accuracy of Model: 56.942%

Team 1:  
Los Angeles Angels

Team 2:  
Pittsburgh Pirates

Submit

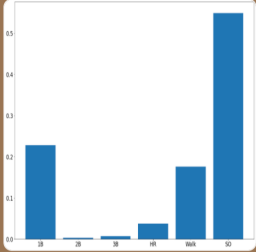
Model last updated with games on or before 11/21/2021



Figure 3: Teams Page

7th Inning Stretch Teams Individual Disclaimer

### Select Baseball Players for Stats

Blake Snell against Cionel Pérez



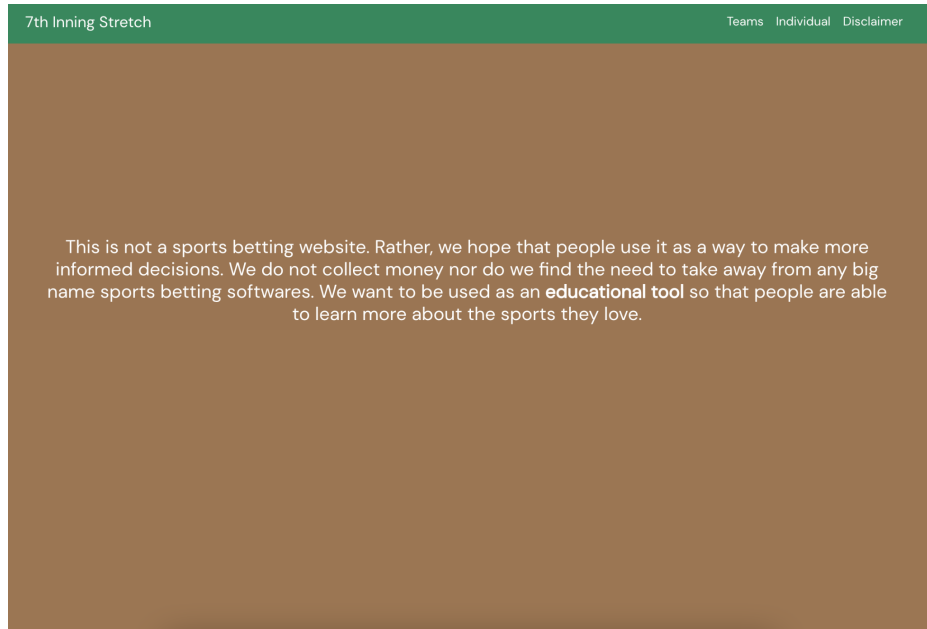
Batter:  
Blake Snell

Pitcher:  
Cionel Pérez

Submit

Model last updated with games on or before 11/21/2021

Figure 4: Individual page



*Figure 5: Disclaimer Page*