Si Yeun Lee, Shahriyar Habib, Don Min, Yi Qiu

Word Count: 2439
Penalty: 0 %

APS360 - Baseball Pitch Prediction Final Report

Introduction

Baseball is a complex sport that requires the consideration of both statistics and psychology. Once a 90-mph baseball leaves a pitcher's hand in Major League Baseball (MLB), it takes only 400 milliseconds to reach home plate. The batter takes the first 100 milliseconds to see the ball, 75 milliseconds to identify the pitch speed, spin, trajectory, and location, and another 50 milliseconds to decide whether or not to swing [1]. This leaves only 150 to 175 milliseconds for the batter to complete the swing if one opts to do so.These tight time constraints challenge the limits of human reaction time, meaning that batters must instinctively predict a pitch. Consequently, a significant part of team training is statistical analysis for pitch prediction. Since the data available is extensive, machine learning is faster and more suitable for predicting a pitch. Hence, a suitable model, trained on historical data of each pitcher and batter, may successfully predict the next pitch.

Given the sheer size of the available data, it was not feasible to learn the unique pitch behaviour of all MLB pitchers. This project seeks to predict the pitch type thrown by a specific pitcher based on statistics that would normally be available to a MLB team. Therefore, the scope of the project focuses on a pitcher considered to be one of the best pitchers in the league - Max Scherzer, an eight-time MLB All-Star with three Cy Young Awards[1][3].

Fig. 0 - Max Scherzer

[1] An award that is given annually to the best pitcher in MLB

Illustration

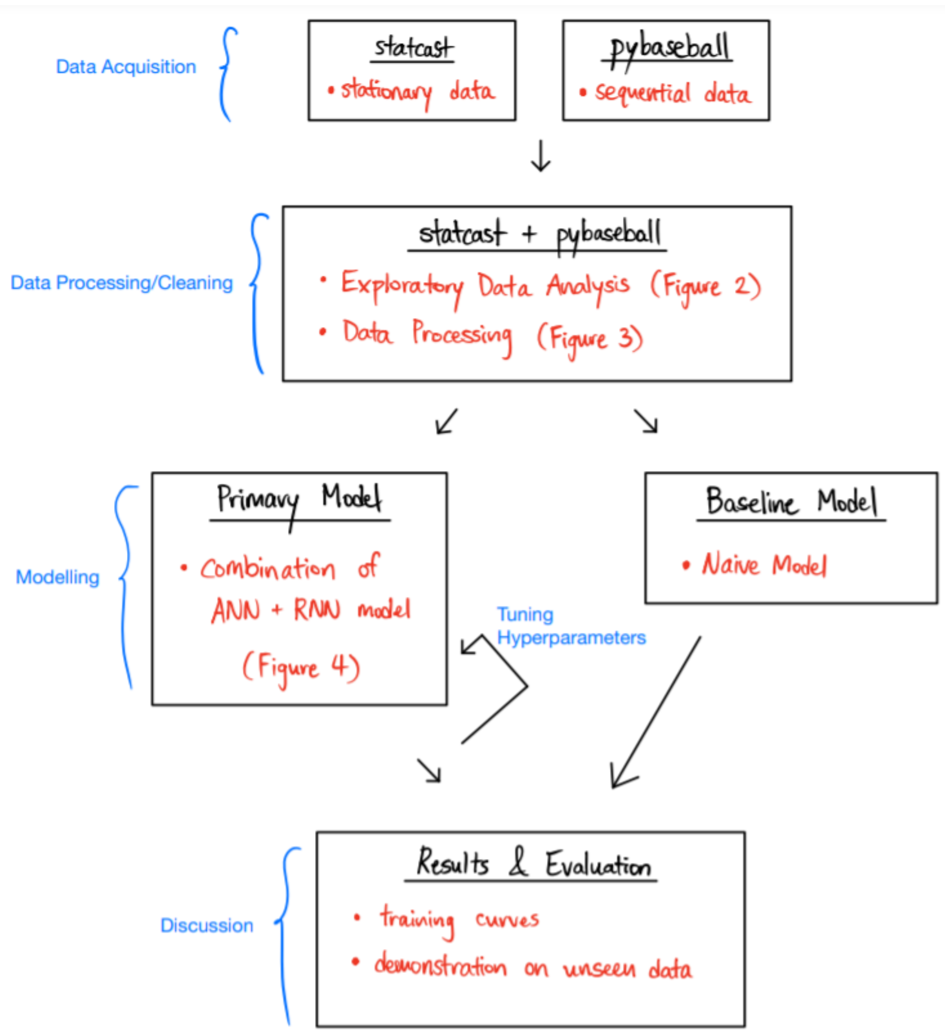


Fig. 1 - Overall Project Workflow

Background & Related Work

A review of current literature reveals that past research generally framed pitch prediction as a binary classification problem. A frequently cited 2012 paper used a set of features consisting of historical performance and game status to create a predictor for each player through linear support vector machines (SVM) [4]. The model was able to ascertain whether the pitch is a fast-ball or not with an accuracy of approximately 70%.

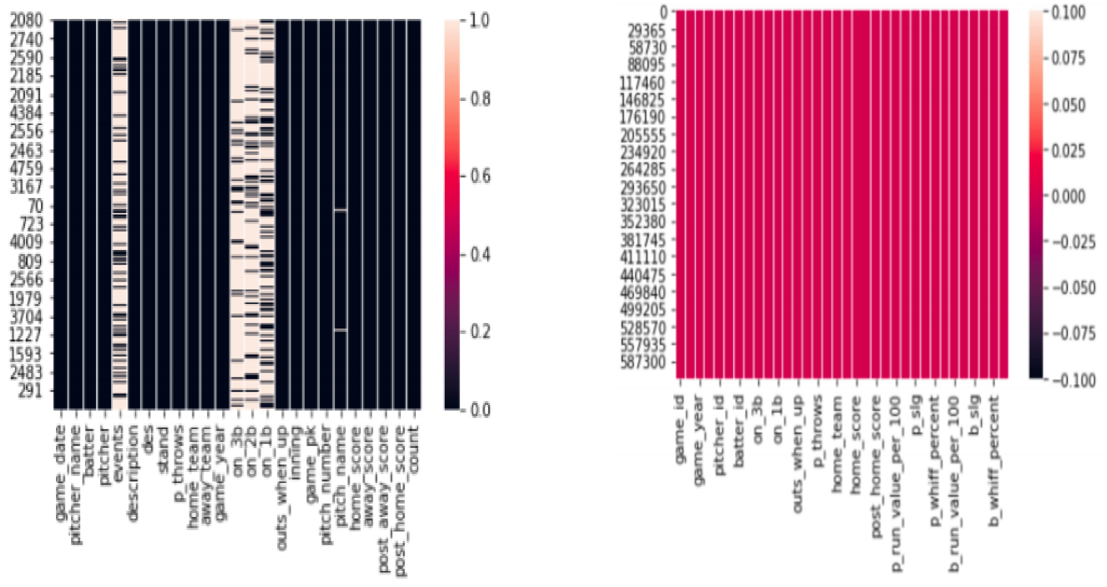
More recent advances attempt to expand on the scope of the pitches, resulting in a multi-class classification problem. One core paper compared three methods - linear discriminant analysis (LDA), multiclass SVM, and classification trees (CT) - to determine the most suitable model [5]. In increasing order, SVM, LDA, and CT all had prediction accuracies that fluctuated around 65%.

Data Processing

Our model requires two forms of data: *Stationary* and *Sequential*. Data was acquired from two sources: Statcast [6], a recent technological innovation in baseball that allows for the collection and analysis of baseball game data; and pybaseball [7], an open source python package for baseball data analysis.

Statcast provides useful pitcher and batter metrics such as: *expected batting average (EBA), run values, pitch type,* and many more. Any metrics related to player performance and averages are referred to as *Stationary Data*. 5100+ pitching/batting statistics from the 2019-2021 seasons were collected. A helper function, *get_statcast_data()*, was created to prepare the .csv files into dataframes which were then concatenated into the pybaseball dataframe.

Sequential Data was gathered from the pybaseball library. It includes features such as *inning, pitch_number, strikes, ball_count,* and many more. The sequential dataframe, however, was unprocessed and had unnecessary features and null values. For example, an important step to cleaning was to combine *ball_count* (e.g. [0,0,1,2,1]) and *strikes* (e.g. [0,1,2,3,1] into a singular mapping: *count* (e.g. ['0-0', 0-1', '1-2', '2-3', 1-1']). This would turn these categorical columns into a sequential data type since the count (and other variables) in a baseball match will tell us what game state the match is currently. Additionally, to balance out the numbers of pitch types, we multiplied the less common pitch types to achieve a balanced distribution. Furthermore, missing values were dropped as they accounted for roughly ~1% of the data. Then, a second helper function, *get_pybaseball_data()*, was created to concatenate the stationary dataframe with the sequential dataframe.



* On the left diagram, white spots represent null/NaN values; right diagram shows no missing values

Fig. 2 - Before & After of Exploratory Data Analysis on Null Values

The final dataframe was then sorted via *game_date, inning, batting_order,* and *pitch_count* to preserve chronological play-by-play states. Furthermore, within each game, data was batched by plate appearance and shuffled. The sequential aspect is the order in which pitches have been thrown. For example, pitch 1, 2, and 3 are grouped by throwing order for that plate appearance[2] (see Fig. 3). This essentially creates 'mini-dataframes' with varying lengths that represent play-by-play interactions. In other words, since the model is based only on Max Scherzer's pitches, each 'mini-dataframe' contains a set of Max Scherzer's plays against a specific batter. Finally, the data was normalized and one-hot encoded.

Finally, abstract Dataset and DataLoader classes were inherited from PyTorch. These custom classes enable the conversion of these 'mini-dataframes' into 3 tensors - stationary tensor, sequential data tensor, and label tensor (the pitch types). The data was then split into the following train:test:validation ratio — 70:15:15.

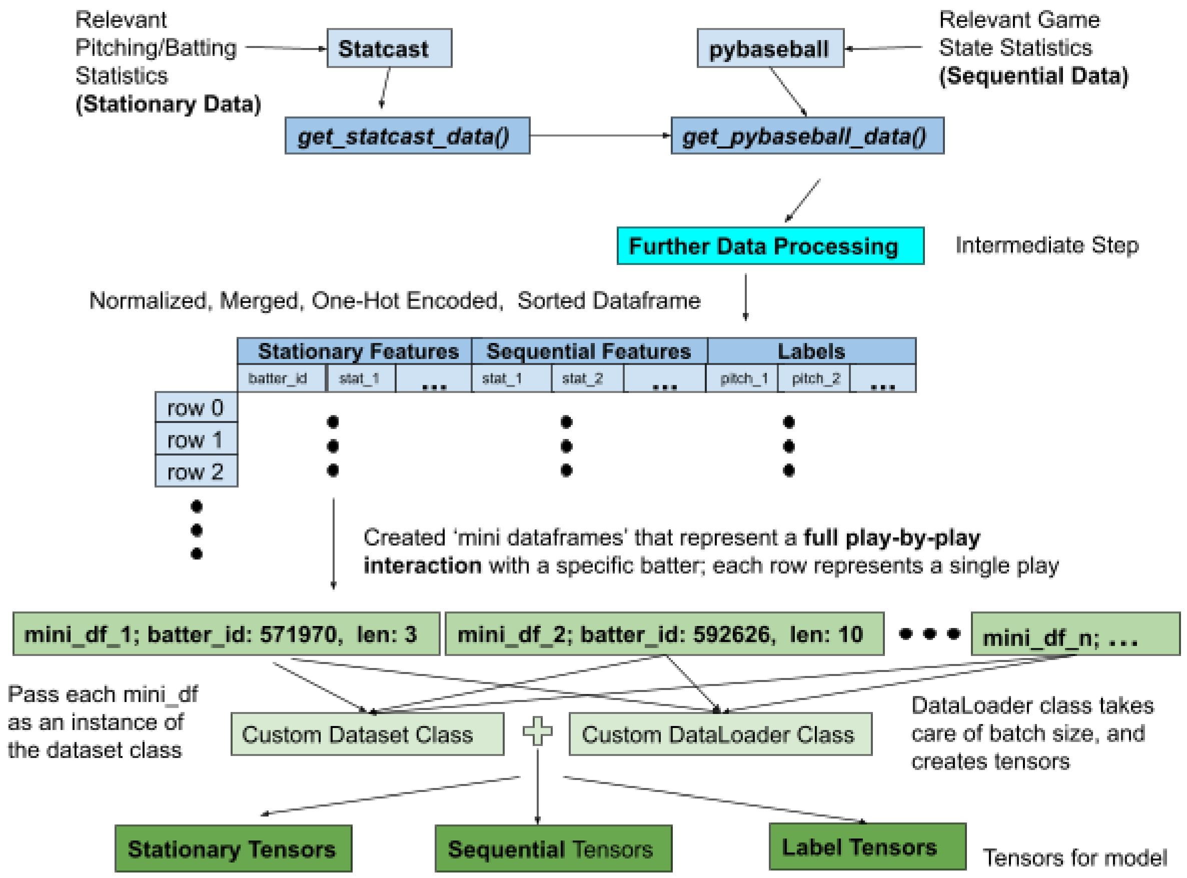

Relevant Pitching/Batting Statistics (Stationary Data)
Statcast
pybaseball
Relevant Game State Statistics (Sequential Data)
get_statcast_data()
get_pybaseball_data()
Further Data Processing
Intermediate Step
Normalized, Merged, One-Hot Encoded, Sorted Dataframe
Stationary Features
Sequential Features
Labels
batter_id
stat_1
stat_1
stat_2
pitch_1
pitch_2
row 0
row 1
row 2
Created 'mini dataframes' that represent a full play-by-play interaction with a specific batter; each row represents a single play
mini_df_1; batter_id: 571970, len: 3
mini_df_2; batter_id: 592626, len: 10
mini_df_n; ...
Pass each mini_df as an instance of the dataset class
Custom Dataset Class
Custom DataLoader Class
DataLoader class takes care of batch size, and creates tensors
Stationary Tensors
Sequential Tensors
Label Tensors
Tensors for model

Fig. 3 - Data Processing Flowchart

[2] Here, plate appearance is equivalent to the batter_id in Fig. 3

Architecture

The final chosen neural network architecture (Fig. 4) links multiple neural networks in order to process the two types of data. An artificial neural network (ANN) with 2 fully connected layers intakes the 13 stationary features, decreasing the dimensionality to 7 features.

Simultaneously, 27 sequential features are delivered to a recurrent neural network (RNN). The chosen RNN model used was a stacked gated recurrent unit (GRU) with 5 layers. This was linked to an ANN consisting of 3 fully connected layers which ultimately outputs 8 features.

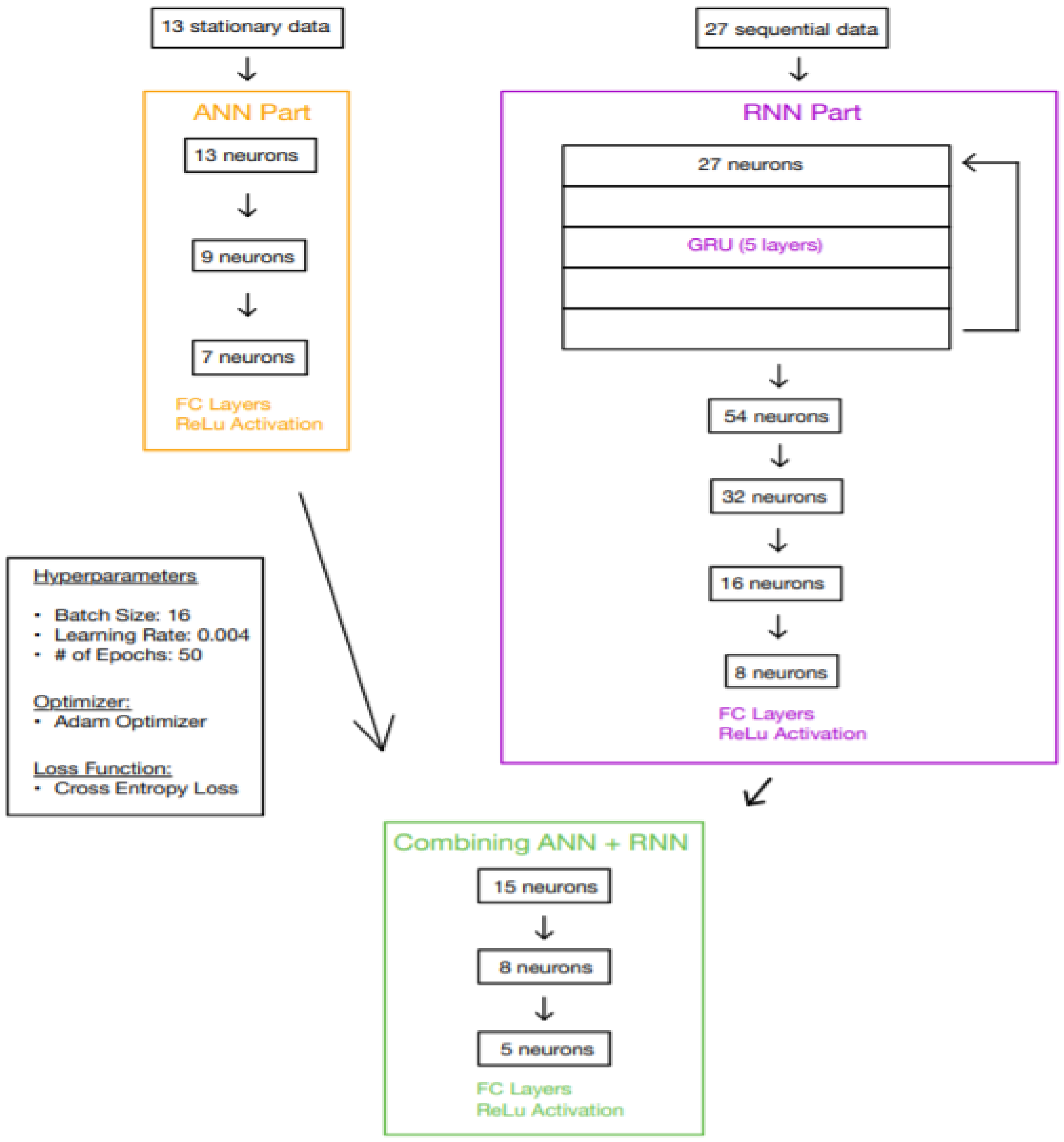


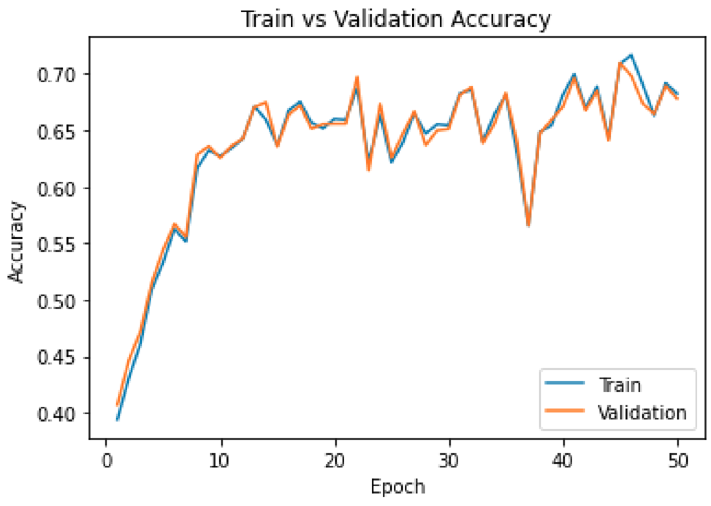
Fig. 4 - Primary Model Architecture Illustration

Since the stationary and sequential data are being analyzed separately by neural networks running in parallel, a final ANN is used to join these two streams of data. The results of stationary and sequential streams are concatenated to form a total of 15 features before being inputted into the 2 fully connected layers.The output of this complex system is the respective probabilities of each pitch type. The model then predicts the pitch with the highest probability.

Baseline

The selected baseline model to compare the designed neural network will be the naive model, chosen for its frequent use as a baseline in a multitude of works [4,5] concerning binary and multi-class classification of the pitch prediction problem. Moreover, it is an intuitive solution that relies on a single quantity, the historic frequency of each pitch thrown, making it the most basic informed solution. This hand-coded heuristic model is implemented by calculating the probability of the pitch types for a selected pitcher, in our case Max Scherzer, based on the aforementioned statistic. The predicted pitch will have the highest probability which functions as the prediction accuracy.

Quantitative Results



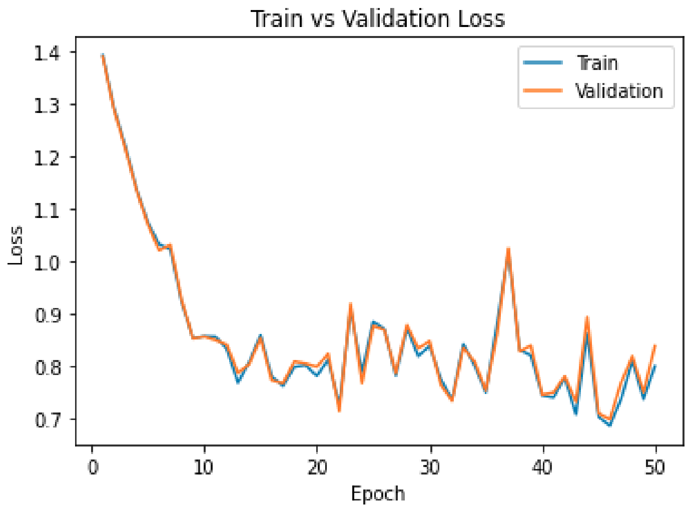


Fig. 5 - Accuracy Training Curve

Fig. 6 - Loss Training Curve

The final model underwent a rigorous process of tuning, after which the architecture and hyperparameters shown in Fig. 4 were finalized while being vigilant of under/overfitting. The training curves for the model are displayed in Fig. 5 and 6. While running through the epochs, the model's loss and accuracy reached a plateau rapidly. Although the training curve is smooth near the first 20 epochs, there is some instability thereafter. However, the train and validation curves closely follow each other and the fluctuations are centered around a plateau. The quantitative results are summarised as follows:

Table 1: Summary of loss/accuracy at the end of 50 epochs for train/validation data

	Train	Validation
Loss	0.8005	0.8389
Accuracy	68.26%	67.81%

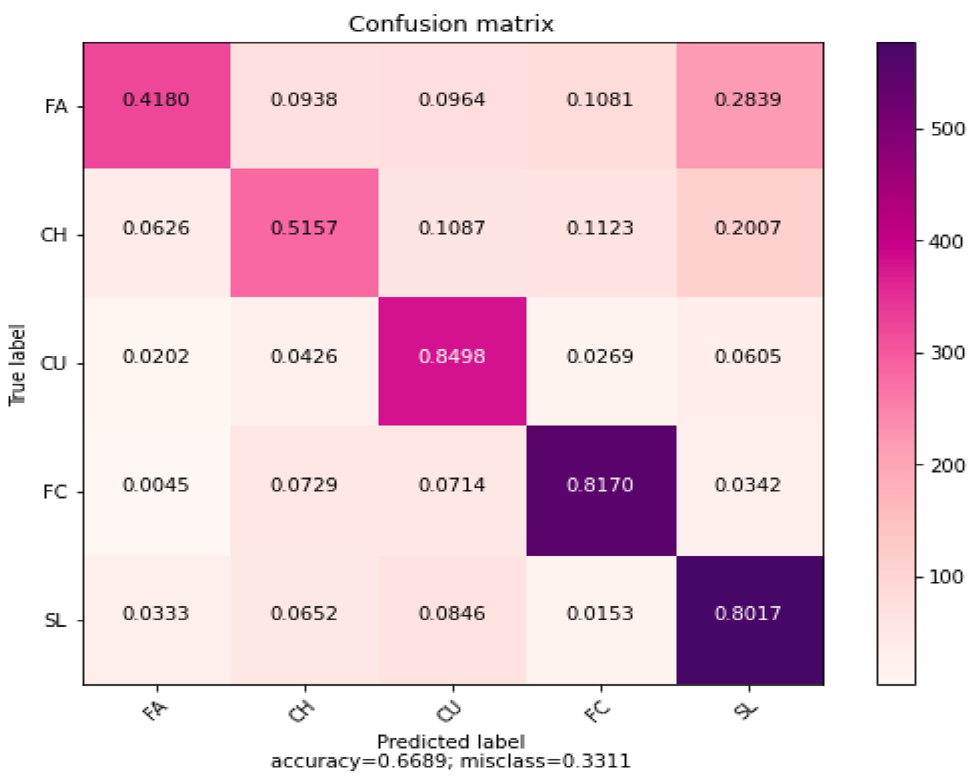
The accuracy was calculated by summing the number of correct predictions and comparing its proportion against the total number of pitches in the provided data. This accuracy code was executed using the test data for both the naive and primary models in order to acquire the test accuracies. The final values are presented in Table 2 — its significance is discussed in the Discussion section of the report.

Table 2: Comparison of predictive accuracies for the baseline and primary model on testing data

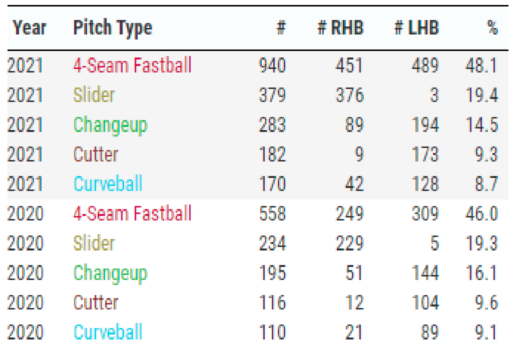
Naive Model	Primary Model	Improvement
22.83%	66.89%	44.06%

Qualitative Results

The confusion matrix below (Fig. 7) demonstrates that the model predicts Max Scherzer's next pitch being a curveball (CU) with a high accuracy of 84.98 %, and 4-seam fastball (FA) with a relatively low accuracy of 41.8%. Incidentally, Max Scherzer's most and least common pitch is the FA and CU with 48.1% and 8.7% usage, respectively (Fig. 8). Thus, our model predicts Max Scherzer's least common pitch (i.e. CU) most accurately and his most common pitch (i.e. FA) least accurately. This demonstrates that our model's accuracy of nearly 70% can be misleading because it fails to capture the true pattern of Max Scherzer's pitches. However, it still performs significantly better than the naive model which predicts the next pitch to be a FA with a mere 22.83% accuracy, barely better than a 1/5 random guess of all the pitch types. Conclusively, our model is superior in predicting overall accuracy across all pitch types. It achieves high accuracies (> 80 %) for the Slider and Cutter pitches; these two pitches being Max Scherzer's second and fourth most common pitch in the 2020/2021 season.



* FA : 4-Seam Fastball, CH: Changeup, CU: Curveball, FC: Cutter, SL: Slider
Fig. 7 - Test Set Confusion Matrix



Year	Pitch Type	#	# RHB	# LHB	%
2021	4-Seam Fastball	940	451	489	48.1
2021	Slider	379	376	3	19.4
2021	Changeup	283	89	194	14.5
2021	Cutter	182	9	173	9.3
2021	Curveball	170	42	128	8.7
2020	4-Seam Fastball	558	249	309	46.0
2020	Slider	234	229	5	19.3
2020	Changeup	195	51	144	16.1
2020	Cutter	116	12	104	9.6
2020	Curveball	110	21	89	9.1

Fig. 8 - Max Scherzer’s Most Common Pitch Types in 2020/2020

<u>Evaluation on New Data</u>

To test on unseen data, we gathered Max Scherzer’s pitches from July 16th to August 4th 2021, and replicated the data to be fed into the model. This time, however, the model was assessed on an unbalanced set of pitches as this is a more realistic representation of Max Scherzer’s pitching style. This way, the model can hopefully detect patterns easily for his most common pitch types: FA, and SL. The distribution of his pitches can be seen below in Fig. 9:

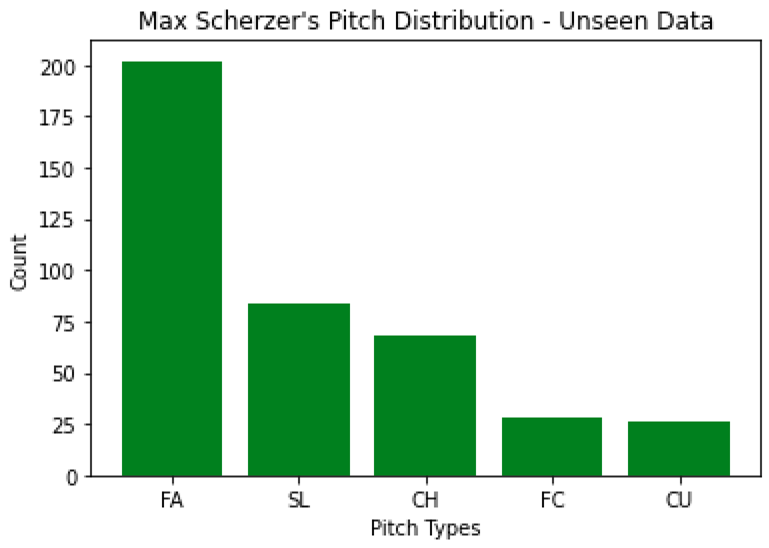
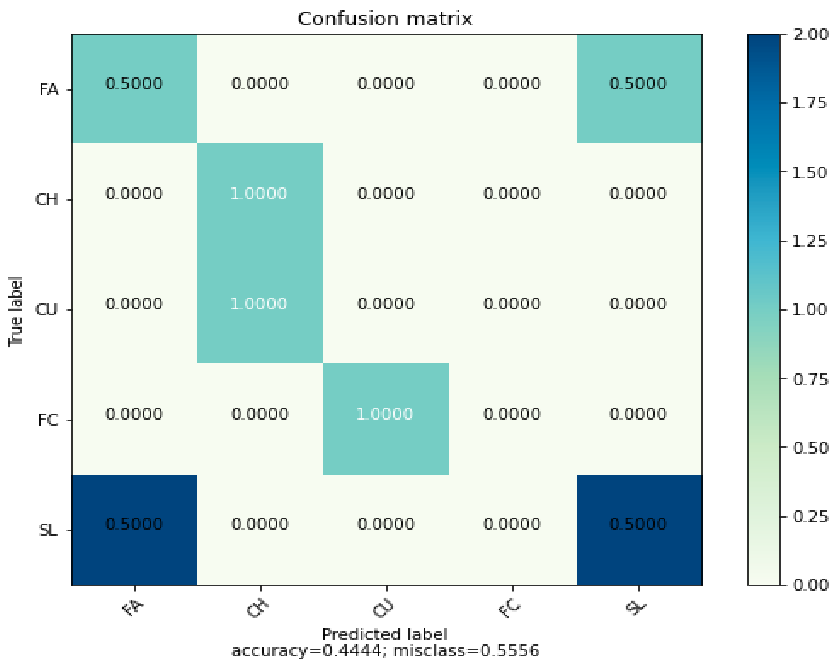


Fig. 9 - Distribution of Max Scherzer's Pitch Types from July - August

The results show that the model only achieved an overall accuracy of 44.4% and the confusion matrix can be seen below in Fig. 10. As expected, the model did not perform as well compared to the test set because the model was trained on balanced data labels (i.e. pitch types). Despite this, the confusion matrix depicts a better accuracy on predicting the next pitch to be a FA (50%) compared to the test set confusion matrix (41.80%). This could be due to the differences in data distribution, and the fact that the unseen data was left unbalanced. Though, the confusion matrix also shows that the model misclassified 55.56% of the time, further suggesting that the model overall performs poorly on imbalanced data.



	FA	CH	CU	FC	SL
FA	0.5000	0.0000	0.0000	0.0000	0.5000
CH	0.0000	1.0000	0.0000	0.0000	0.0000
CU	0.0000	1.0000	0.0000	0.0000	0.0000
FC	0.0000	0.0000	1.0000	0.0000	0.0000
SL	0.5000	0.0000	0.0000	0.0000	0.5000

Fig.10 - Confusion Matrix for Unseen Data

When passing the new data through the naive model, it achieved an accuracy of 33.34%, which is significantly better than its previous result of 22.83%. Again, we believe this is due to the imbalance of pitch types that resulted in improved performance. Though our model only predicts ~10% better than the naive model on unseen data, an overall accuracy of 44.4% is still acceptable considering the randomness of pitches, and the fact that Max Scherzer pitches 5 different pitch types. To determine the full capacity of our model, we would need to run it on future seasons, and different pitchers to see the variability of results.

Table 3: Comparison of model performances before and after balancing data

	Naive Model Accuracy	Primary Model Accuracy	Confusion Matrix: ‘FA’ Accuracy	Confusion Matrix: ‘SL’ Accuracy
Test Dataset (balanced)	22.83 %	66.89%	41.80%	80.17%
Unseen Dataset (unbalanced)	33.34 %	44.44%	50.00%	50.00%

Discussion

An accurate prediction of an incoming pitch provides the batter with a significant offensive advantage. Considering both the random and psychological factors between the batter and pitcher, an accuracy of ~66.89% is adequate. Given that past literature dealing with multi-class pitch prediction also produced models with accuracies around 65% [5], it is concluded that our model is passable.

As exhibited by Table 2 in the Quantitative Results section, the primary model outperforms the baseline by ~44.06%. For unseen unbalanced data, the baseline model shows higher accuracy by about ~11.1%. Conversely, the primary model underperforms for the most common pitches thrown by Max Scherzer- possessing a relatively low accuracy of ~41.8% for his most frequent FA pitch. The primary model achieves greater than 80% accuracy on other less common pitch types. This indicates that the primary model is unable to extrapolate general trends from re-balancing data before input. One justification for balancing data via pitch type was for equal sequence lengths and removing pitch type bias in the dataset.

Once again referring to alternative literature, research [4] reveals that a binary classifier model for pitch prediction performs poorly when pitchers change their preferred pitch type between seasons. For instance, a 51% change in fastball throw frequency by the pitcher reduced the binary classifier model accuracy to ~32%. Our primary model's ~66.89% accuracy is shy of the binary classifier model; however, it performs ~41.8% accuracy in the worst case. Re-balancing game data allows stable outputs when players change preferred pitch types.

After analyzing the results, re-balancing became the primary source of error for future directions. Without rebalancing, models become static and often underperform in changes of pitch type frequency, less common pitch types, and changes in seasonal pitch style. With re-balancing, the model obtains shortfalls in identifying general trends, maximizing accuracy for frequent pitches. For further analysis, if the test model fails to capture the pattern of pitchers' most common pitch type, separate models (with the exact same parameters) should be created for various pitchers. If inconsistencies due to data processing persists, then the model design is at fault, rather than it being an anomaly for Max Scherzer. Future testing requires multiple models for each data format, which is beyond the scope of this project. Future studies should focus on testing formats of balanced, weighted balance, and unbalanced datasets for baseball prediction models.

Ethical Considerations

MLB provides public baseball data for constructing a machine learning model. Terms of use for MLB websites/datasets, such as Statcast, give exceptions to personal and non-commercial use [6,8]. Hence, no ethical issues will arise from data collection. In addition, there is minimal bias since the model trains on a particular pitcher evaluated by the same game performance metrics as their peers. Disparate impact falls on baseball pitchers. Reducing the surprise element of throwing different pitch types reduces the complexity of the game for batters and gives an advantage. However, the final judgment is reliant on batter observation. Reliance on the predictive model does not eliminate the element of surprise.

Project Difficulty/Quality

There exists unexpected difficulties in the pitch prediction problem. The main issue is that the two types of data must be identified: stationary and sequential. Starting from the data processing stage, they must be selected and organised correctly so that the model will intake useful data and be able to effectively analyze the patterns. Afterwards, the model architecture must be chosen to accommodate these two data types then relate them together. There is also an imbalance in the data - each pitcher will favour a specific set of pitches, meaning that there is no equal distribution of data for the pitches because their total appearances over the baseball seasons are inconsistent. Moreover, individual pitchers will have their own patterns, meaning that attempting to make a singular model across all pitchers will result in a low predictive accuracy. As a result, the solution to the problem is more useful when it is reframed so that the model is attuned to predict the pitch for a designated pitcher.

References

[1] D. Coburn, “Baseball Physics: Anatomy of a Home Run,” Popular Mechanics, 14-Nov-2017. [Online]. Available: https://www.popularmechanics.com/adventure/sports/a4569/4216783/#:~:text=A%2090%2Dmph%20fastball%20can,%2C%22%20Zimmer%2Dman%20says.

[2] J. Bogage, “What is sign stealing? Making sense of Major League Baseball's latest scandal.,” The Washington Post, 14-Feb-2020. [Online]. Available: https://www.washingtonpost.com/sports/2020/01/14/what-is-sign-stealing-baseball/.

[3] “Max Scherzer,” Encyclopædia Britannica. [Online]. Available: https://www.britannica.com/facts/Max-Scherzer. [Accessed: 13-Aug-2021].

[4] G. Ganeshapillai and J. Guttag, “Predicting the Next Pitch,” *MIT Sloan Sports Analytics Conference 2012*, Mar. 2012.

[5] G. Sidle and H. Tran, “Using multi-class classification methods to predict baseball pitch types,” Journal of Sports Analytics, vol. 4, no. 1, pp. 85–93, 2018.

[6] “Statcast Pitch Arsenal Stats,” baseballsavant.com. [Online]. Available: https://baseballsavant.mlb.com/leaderboard/pitch-arsenal-stats.

[7]“Pybaseball: Pull current and historical baseball statistics using Python,” GitHub. [Online]. Available: https://github.com/jldbc/pybaseball. [Accessed: 13-Aug-2021].

[8] “MLB.com Terms Of Use,” *MLB.com*, 04-Feb-2020. [Online]. Available: https://www.mlb.com/official-information/terms-of-use.