Predicting Types of Pitches in Baseball

Gabriel Levy and **Alexander Donald**
{galevy,aldonald}@davidson.edu
Davidson College
Davidson, NC 28035
U.S.A.

Abstract

In this paper, we aim to predict baseball pitch types based on the pitch's speed, spin profile, and movement. We implement a K-nearest neighbors model, a support vector machine model, and a decision tree model. The KNN model performed with an average 0.855 F1 score and the SVM model had a 0.849 F1 score average. The decision tree model performed a bit worse with an 0.820 F1 score. Splitters and cutters performed the worst out of all pitch types in both the SVM model and the KNN model. 4-Seam fastballs performed the best in both models.

1 Introduction

In this paper, we attempt to predict the type of pitch in a baseball game based on several factors: the spin profile, the speed, and the movement of the baseball. The spin profile of the baseball involves the spin rate, the spin axis, and the spin efficiency. These variables will be explained more in the background section. There are seven different types of pitches that we will try to predict: 4-Seam fastballs (FF), sinkers (SI), cutters (FC), sliders (SL), curveballs (CU), changeups (CH), and splitters (FS). There are other classifications of pitches that exist, but almost all pitches can fit into these labels. In our experiments, we use 3 models: K-nearest neighbors, support vector machines, and decision trees. We will utilize the Scikit-Learn library for all of our models.

Another experiment with the same goal was performed by Frank Bruni and posted on the BaseballDataPros page (Bruni 2020). Bruni uses a decision tree to predict pitch types based on speed, spin rate, and movement data. He achieves an accuracy and F1 score of 87%. We credit Bruni for a portion of the code for our decision tree. An additional paper that further analyzes the same question used support vector machines, decision trees, logistic regression, and other methods. This paper was written by Ryan Plunkett for his bachelor's thesis at Harvard University (Plunkett 2019). For most of his methods, he achieved success in the $60\% - 62\%$ range.

Ethical Concerns

Since this is baseball and just a game, there are no real ethical concerns. However, some pitchers and pitches are underrepresented in baseball. There are pitchers that throw with drastically different mechanics. Additionally, some pitches like splitters are not thrown nearly as often as pitches like fastballs and sliders.

2 Background

About the Data

We collected all of our data from Baseball Savant (Savant 2022). Our outcome variable is pitch type, and our features are spin rate, spin axis, spin efficiency, speed, horizontal movement, and vertical movement. The following definitions come from our own personal knowledge. Spin rate, measured in rotations per minute, is simply how fast the ball is spinning. The spin axis is given as the hour and minute the ball is spinning on, pictured as the hour hand on a clock. This data is also given as the hours and minutes deviated from 12 o'clock. The spin efficiency is defined as the percentage of the spin that contributes towards the movement of the baseball. Speed is just the miles per hour of the ball at release. Horizontal movement and vertical movement are how much the ball moves in those respective directions.

Data Preprocessing

To put this data into a format that we could use, we did a few different preprocessing methods. First, our data was not all from the same CSV. Our horizontal and vertical movement stats came from a separate dataset. We had to average the data from 2020 to 2022 in the movement dataset, and then combine that dataset with the spin and speed data. There were also 4 players that seemed to be missing horizontal break data, so we chose not to include those rows.

Once all of our data was in the same dataset, we had to adjust the spin axis data into a usable format. It started in hours and minutes, but we changed it to a decimal where the number to the left of the decimal was the hours and to the right was the fraction of an hour (minutes/60). After this, we had all of our data in decimal form, and we could normalize the features to each have a mean of 0 and standard deviation of 1.

Hyperparameters

There are a few hyperparameters that we search through in our models. In our K-nearest-neighbors model, we search to find the best k value. In our support vector machine model,

we search through regularization penalty strength and tolerance levels. For our decision tree method, we manually input the maximum depth of the tree and the minimum samples to split parameters. For the KNN model, we implemented our own search method in which we used our own validation set of data. For the SVM model, we utilized the Scikit-Learn library's random search method, which has cross-validation built in to the method.

F1 Score

Since some pitch types are thrown much more often than others, some pitch types in our dataset are under and over represented. Thus, we decided to score our models based on F1 score, which accounts for discrepancies in the amount of each class.

3 Experiments

Using the normalized pitch data for each pitcher, we created both K-Nearest Neighbors and Support Vector Machine models to generate predictions of pitch type. For our K-Nearest Neighbor model, we used a manual approach for finding the optimal k value using a validation set. We then took the average of the best k values over runs to determine the best value. For our Support Vector Machine, we used sklearn's RandomizedSearchCV to perform a cross-validated search for optimal parameters. In the case of the support vector machines, we varied the regularization strength and the tolerance value used for stopping. The success of each model created with a given set of hyperparameters was evaluated using the average accuracy achieved over trials. More details of the experimental setup can be seen in Figure 2.

Experimental Setup	
General	
Train/Test Split	80/20
Validation Method for Searches	Cross validation
SVM Random Search	
Number of Models Created	10
C (Inverse Regularization Strength)	$.5 - 2.5$
Tolerance Distribution	$1e-5 - 1.01e-3$
K-Nearest Neighbors	
Number of Models Created	100
Weighted/Unweighted	Weighted

Figure 1: The methods and values used to set up our experiments . When a range of values are shown in the random search sections, each test selected hyperparameter values from a uniform distribution of the provided values.

4 Results

In general, all the models that we created gave solid predictions for pitch type given data from MLB pitchers. We present the specific results of each model type below.

Tuned Hyperparameters and f1 Scores	
K-Nearest Neighbors	
k Value	11
Weighted/Unweighted	Weighted
F1 Score	0.855
Support Vector Machine	
C	2.387
Tolerance	0.00069
F1 Score	0.849

Figure 2: The optimal hyperparameters and the corresponding model's accuracy are listed

Support Vector Machine

After training our SVM model, we found optimal parameter as listed in Figure 2 with an average F1 score of .849 over 10 runs. This is a solid score that matches up closely with our K-Nearest Neighbors model. With that being said, nearly 15% of pitches were missed. Figure 3 breaks down what percent of the missed pitches came from each pitch type and the percent wrong from each pitch type.

Pitch Type	Percent of Missed	Percent Wrong of Test Set
CH	11.1%	12.5%
CU	15.6%	14.7%
FC	21.5%	41.8%
FF	6.7%	2.3%
FS	5.9%	66.7%
SI	17.1%	17.5%
SL	22.2%	13.8%

Figure 3: Percentage of missed predictions for each pitch type in test set and percentage of missed predictions per each pitch in the test set.

As seen in Figure 3, sliders and cutters are the two most missed pitch types by our SVM model. However, splitters and cutters were the worst performing pitches in the model. One reason that sliders may be missed more than other pitches is because sliders are close to halfway between a cutter and a curveball in terms of metrics, making it more likely for this pitch type to be confused with others. Cutters are halfway between a fastball and a slider. Splitters (FS) may have a poor score because of their under-representation in the dataset. Fastballs are the most represented in the dataset and they have the best score.

Based on the performance of our model on unseen data, we do not see any evidence of overfitting. In fact, our model performed even better on the testing data than its average accuracy on the cross-validation data, which strongly suggests that our model was not overfit to the training data.

K-Nearest Neighbors

After training our K-Nearest Neighbors model, we found an optimal k value of 11 as listed in Figure 2. This means that our final model predicts based off of the 11 most similar data examples. With an average F1 score of .855 over 100

runs, we felt confident that this was an accurate representation of our model. This score is slightly higher than that of the SVM. Figure 4 breaks down what percent of the missed pitches came from each pitch type and the percent wrong from each pitch type.

Pitch Type	Percent of Missed	Percent Wrong of Test Set
CH	12.0%	16.8%
CU	12.7%	21.4%
FC	28.2%	41.4%
FF	5.6%	3.6%
FS	9.2%	60.0%
SI	17.6%	15.3%
SL	14.8%	15.1%

Figure 4: Percentage of missed predictions for each pitch type and percentage of missed predictions per each pitch in the test set

As seen in Figure 4, cutters and sinkers are the two most missed pitch types by our SVM model. However, splitters and cutters are once again the worst performing pitches in the model. To reiterate, cutters are metrically close to both sliders and fastballs. Splitters may simply be underrepresented since metrically they are not very similar to many pitches, with a low spin rate. Sinkers may be performing poorly since they have similar metrics to a fastball including speed and spin efficiency. Just like the SVM model, fastballs performed the best of all the pitch types.

Comparison to previously used model

As mentioned in the introduction, Frank Bruni did a similar experiment. He used a Decision Tree model. We implemented his model on our own data which returned a score around 0.82. This model seems to do slightly worse than the two models we built and tuned the hyperparameters for. Using previous work helps to give a baseline of the quality of our models. Based off this Decision Tree model, it is apparent that our K Neighbors and SVM models are quality models and are an improvement over Bruni's Decision Tree model.

5 Conclusions

During our study of MLB pitch data, we used data on different pitch types such as spin rate, spin axis, spin efficiency, speed, and horizontal and vertical movement in order to create a predictive model. We generated both a Support Vector Machine (SVM) and K-Nearest Neighbors models. After tuning both models and comparing them to a previously created model for the same experiment, we concluded that the K-Nearest Neighbors model was the most accurate with a F1 score of 0.855. Although this is not perfect and has room for improvement, it is a solid model based off of completely real world data. One potential area for improvement could be adjusting the data for pitchers who throw from different arm angles. For example, if someone throws more sidearm than over the top, their axis of rotation is going to be different. A possible solution could be grouping pitchers based off of arm angle and using each group to predict pitches by other pitchers in that group. Another interesting project would be to analyze who is pitching at the time and train the model to them. Assuming these models would be used for either analytics or to enhance fan experience at a game, the model could be trained when a new pitcher enters the game and specifically for that pitcher. This should yield much better results.

6 Contributions

G.L. and A.D partner programmed the data loading and cleaning, and performed data exploration together. G.L. and A.D. did the programming of the models together.. G.L. wrote the abstract, introduction, and background sections of the paper. A.D. wrote the experiments, results, and conclusion sections and created the figures therein.

References

Bruni, F. 2020. Classifying pitch type using machine learning. https://www.baseballdatapros.com/posts/2. Retrieved on May 7, 2022.

Plunkett, R. 2019. Pitch type prediction in major league baseball. https://dash.harvard.edu/bitstream/handle/1/37364634/PLUNKETT-SENIORTHESIS-2019.pdf?sequence=1&isAllowed=y. Retrieved on May 7, 2022.

Savant, B. 2022. Spin direction - pitches leaderboard. https://baseballsavant.mlb.com/leaderboard/spin-direction-pitches?year=ALL&min=1&sort=9&sortDir=asc&pitch_type=ALL&throws=&playerName=&team=&pov=Bat. Retrieved on May 7, 2022.