Cogs 260, Spring 2021: Mini-project #2
Due date: May 23th, 2021
Authors: Quirine van Engen, Davis Lee, Monica Van & Yueying Dong

Initial Speed and Spin Direction are the Most Relevant Features for Determining Baseball Pitch Types

1. Introduction

Baseball is one of the most popular sports played in the United States. In this adversarial game, the Pitcher is pitted against a Batter whose goal is to hit the ball as far as possible. The Pitcher utilizes a variety of ball pitches in order to increase the chances of a Batter missing the ball for a 'Strike', three strikes leads to a 'Strike out' for the Batter. There are four main types of pitches: fastball (FA), slider (SL), curveball (CU), and changeup (CH). Other less commonly-thrown pitch types include the sinker (SI), fast cutter (FC), fast splitter (FS), knuckleball (KN), screwball (SC), forkball (FO), and eephus (EP). Through the course of a baseball game, sports commentators would determine and announce the type of pitch that was thrown. Commentators were trained through exposure to different pitches over time and would often determine the type based on attributes such as the speed and arc of the pitch. New advances in technology and computer vision can now precisely capture the different aspects of any given pitch, for instance: the release point of the ball, the acceleration at different times, the arc path, and the end position of the ball relative to the batter's batting zone. Using these metrics, machine learning algorithms can be trained on this data to accurately predict what type of pitch a pitcher throws. The dataset we will be using contains information about the properties of each pitch, the pitcher, and the classification by a sports commentator.

Our project focuses on applying the K-nearest neighbors (KNN) classifier to examine how well it can label different pitch types given different k-values and for different subsets of features on which the model is trained. We are interested in identifying which features are most important in classifying pitch type and hypothesize that the initial velocity of the pitch will be the most predictive feature

Humans use pitch speed and arc trajectory to determine pitch type, while a KNN based Sequential Feature Selector model identifies the ball's spin direction and its speed as the most predictive features. This result is interesting in that while it is hard to identify spin direction with the naked eye, its effects can be seen in the form of the ball's arc trajectory. Due to the advanced and precise nature of computer vision, machine learning models such as the one implemented here have identified the causal feature of the arc trajectory as the top determinate feature.

2. Methods

2.1 Datasets

The dataset contains 6920 observations in rows, and consists of the pitcher, the pitch type, and 19 different features: `mlbid` (pitcher ID), `ab_id` (ID of player at bat), `pitch_id`, `start_speed` (of the pitch), `x0` (the pitch's initial right-left position), `z0` (initial height of pitch), `px` (right-left position of pitch over home plate), `pz` (height of pitch over home plate), `pfx_x` (displacement between initial L/R position and home plate L/R position), `pfx_z` (displacement between initial height and home plate height), `stand` (position of the batter), `inning`, `height` (height of the ball release), `spinrateND` (the spin rate of the ball), `spindirND` (the direction of spin of the ball), `vxf` (final pitch velocity on the horizontal axis), `vzf` (final pitch velocity on the vertical axis), `xangle` (angle of ball's horizontal trajectory), and `zangle` (angle of the ball's vertical trajectory).

Table 1 lists the number of observations for each type of pitch. Fastballs (FA) make up more than a third of the pitch frequencies with some pitches (i.e. the eephus (EP)) appearing only once.

Table 1: The 11 pitch types and corresponding observations in our dataset. There is a heavy bias towards fastballs, whereas screwballs, forkballs, and eephus have very limited observations.

Pitch type	**Observations in dataset**
Fastball (FA)	2481
Sinker (SI)	1430
Slider (SL)	1006
Curveball (CU)	731
Changeup (CH)	640
Fast Cutter (FC)	425
Fast Splitter (FS)	149
Knuckleball (KN)	49
Screwball (SC)	6
Forkball (FO)	2
Eephus (EP)	1
Total Pitches	6920

2.2 Data Pre-processing

Pitcher names and identifiers (pitch_id, mldid, ab_id) were excluded from the feature array. Though some pitchers throw a limited amount of pitch types and could therefore be predictive for the pitch type, our scope of analysis is focused on the specific features of the ball itself (velocity, acceleration, and curves), rather than the pitchers. Thus we excluded the pitcher information from our dataset.

2.3 Model comparison and protocol

Because the dataset is heavily skewed with a bias towards fastballs, we will use permutation shuffling of the pitch types to calculate a baseline accuracy that reflects a normalized chance level.

We initially train naive KNN models with k-neighbors ranging from 1 to 200 on the full set of features to establish an optimal k-neighbors value. Using the optimal k-neighbors value, we then perform a forward sequential feature selection to determine a subset of features best predictive of our pitching data. Lastly, we will train a features-KNN model with the optimal number of predictive features in order to predict the pitch type of a test data subset. For each fold in a 5-fold cross-validation, we split the data using a 80:20 train-test split paradigm. Then, for each different number of neighbors (ranging from 0 to 200), we fit a KNN model to the training data and test how well this model performs on the testing set. The metric we chose to evaluate model performance is accuracy (ACC), calculated using the following formula:

$$ACC = \frac{True\ Positives\ +\ True\ Negatives}{Positives\ +\ Negatives}$$

Software & packages

From the Sklearn package, we used Kfold and KNeighborsClassifier, and SequentialFeatureSelector from the mlxtend package. All default parameters were kept for KNeighborsClassifier, with the exception of neighbors. Furthermore, Seaborn and matplotlib were used for plotting.

2.4 Model Selection + Inference

Using the amount of neighbors that returned the highest accuracy score during model comparison, we obtain the best model for KNN to fit to the entire data set. After fitting the optimal KNN model, we again report the final ACC score, including a confusion matrix to identify model false identifications. The matrix allows us to interpret failure areas of the model, such as when the model is returning Fastball more frequently for Pitch types with lower observation points.

To obtain the features that are most predictive of pitch type, we implemented a sequential feature selection model using the training set. We found that inclusion of >2 top predictive features produced rapidly diminishing returns on accuracy. The top feature was identified as '`spindirND`' (the

direction of ball spin) which accounted for the highest baseline accuracy, and 'start_speed' (the initial speed of the ball) as the second best additive predictive feature (figure 3).

3. Results

3.1 Model comparison

The baseline accuracy for the KNN when true pitch labels are shuffled is 0.3567; this is interpreted as at-chance, the model predicts around 0.36 of the labels correctly.

The naive KNN model, trained without feature selection, achieves a max accuracy of 0.7404, which is much higher than the at-chance accuracy of 0.3567. The naive model's highest accuracy was found at k = 9 (figure 1).

We then use forward sequential feature selection to curate the most predictive features with a KNN model using 9 neighbors. Using just two features, `'start_speed'` and `'spindirND'`, the KNN is able to achieve an accuracy of 0.7334 (table 2). Adding an additional third feature only increases the accuracy by 0.01 (figure 3), suggesting diminishing returns of further addition of features.

Table 2: Result of test set accuracy of KNN models (k = 9) with and without best subset of features. The accuracy is approximately the same.

Test accuracy without feature selection	0. 7404
Test accuracy with best feature subset ('start_speed', 'spindirND')	0. 7334

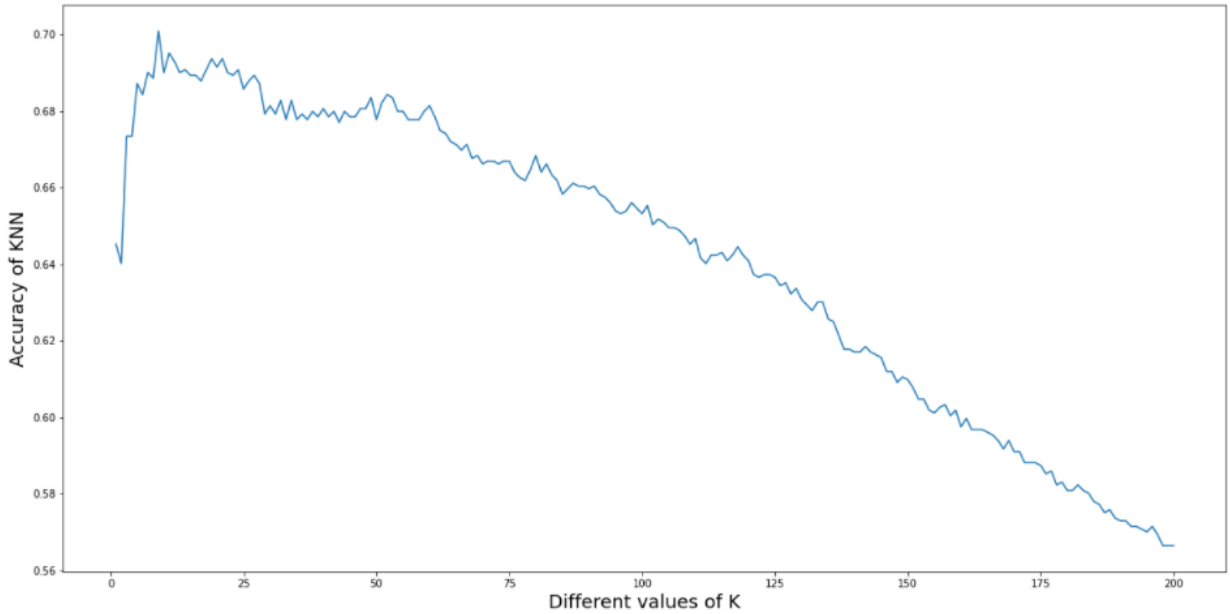

Figure 1: Performance of our KNN model using all features for different values of k on the normal data. Here we see the performance of our model in accuracy, as a function of k. Note that the pitch types labels were shuffled. We see an increase of performance when increasing k for the first 9 values of k. After that, there is a downward trend of performance. The maximum accuracy is reached at 74% for k = 9.

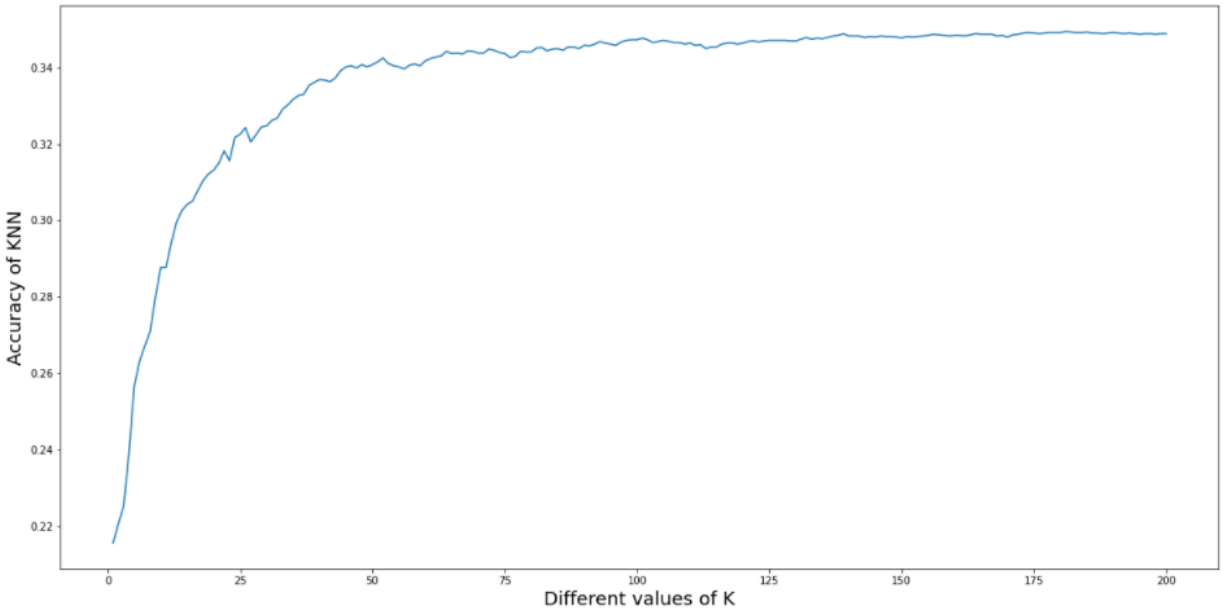

Figure 2: Performance of our KNN model using all features for different values of k on the shuffled data. Here we see the performance of our model in accuracy, as a function of k. Note that the pitch types labels were shuffled. We see an increase of performance when increasing k. But the maximum accuracy is still only 35.7%.

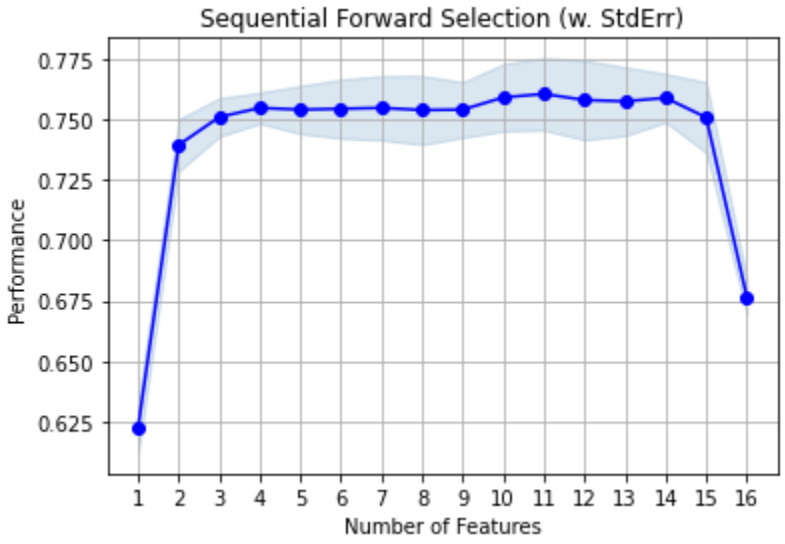

Figure 3: Sequential forward selection of features. Here we see the performance, measured in accuracy, as a function of features. We see the biggest increase up to using two features. Adding more than two features still increases our model's performance, but only in small increments.

3.2 Visualization

Our analysis demonstrated that the top two features were sufficient to reach an acceptable performance. We are able to show you a beautiful scatter plot of the true labels (figure 4 bottom panel), and the ones predicted by our model (figure 4 top panel). One observation is that the true labels contain clusters with quite some overlap. Whereas the clusters predicted by our model show less overlap.

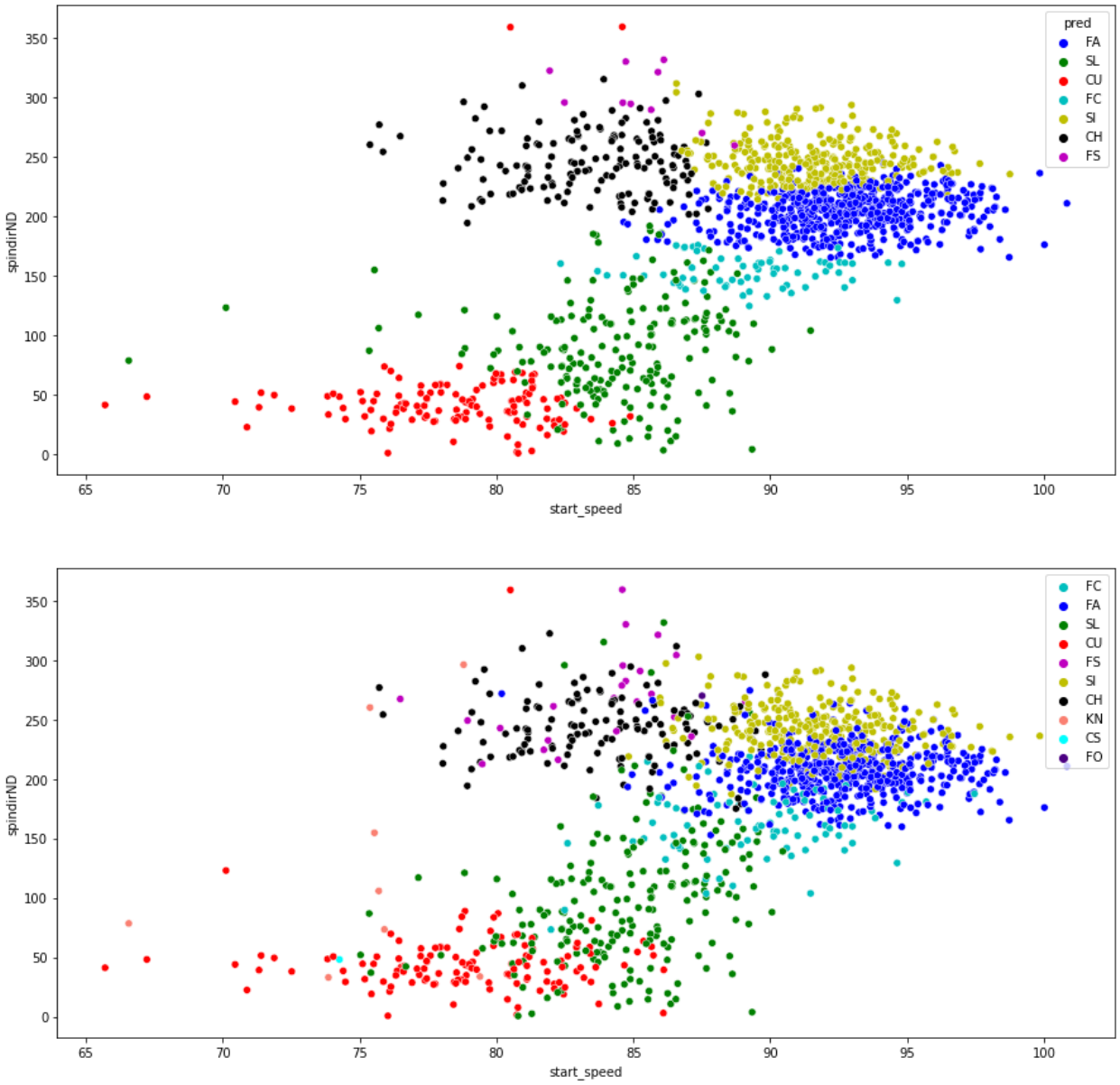

Figure 4: Scatterplots of the predicted pitch types by our KNN model (top panel), and the true pitch types (bottom panel). The colors between the top and bottom panel correspond to the same pitch type. We can see that the true labels are clusters, but with some overlap between them. Our model however has less overlap between the different clusters of pitch types. Also, notice that the model never predicted a KN, CS, or FO ball.

Figure 5 shows a confusion matrix between the true labels and the predicted labels. There are a few key observations that stand out. First, we can see a darker diagonal, which is a key feature of a confusion matrix made from a model with moderate to high accuracy. Second, the pitches with the lowest observation in our dataset have never been predicted by the model. Third, there are a few pitch types that are easily confused. The model has difficulties distinguishing a change-up (CH) and a fast cutter (FS) ball. It incorrectly predicted a fast cutter as a changeup ball. Additional difficulties involve distinguishing

between slider (SL) and knuckle (KN) balls.

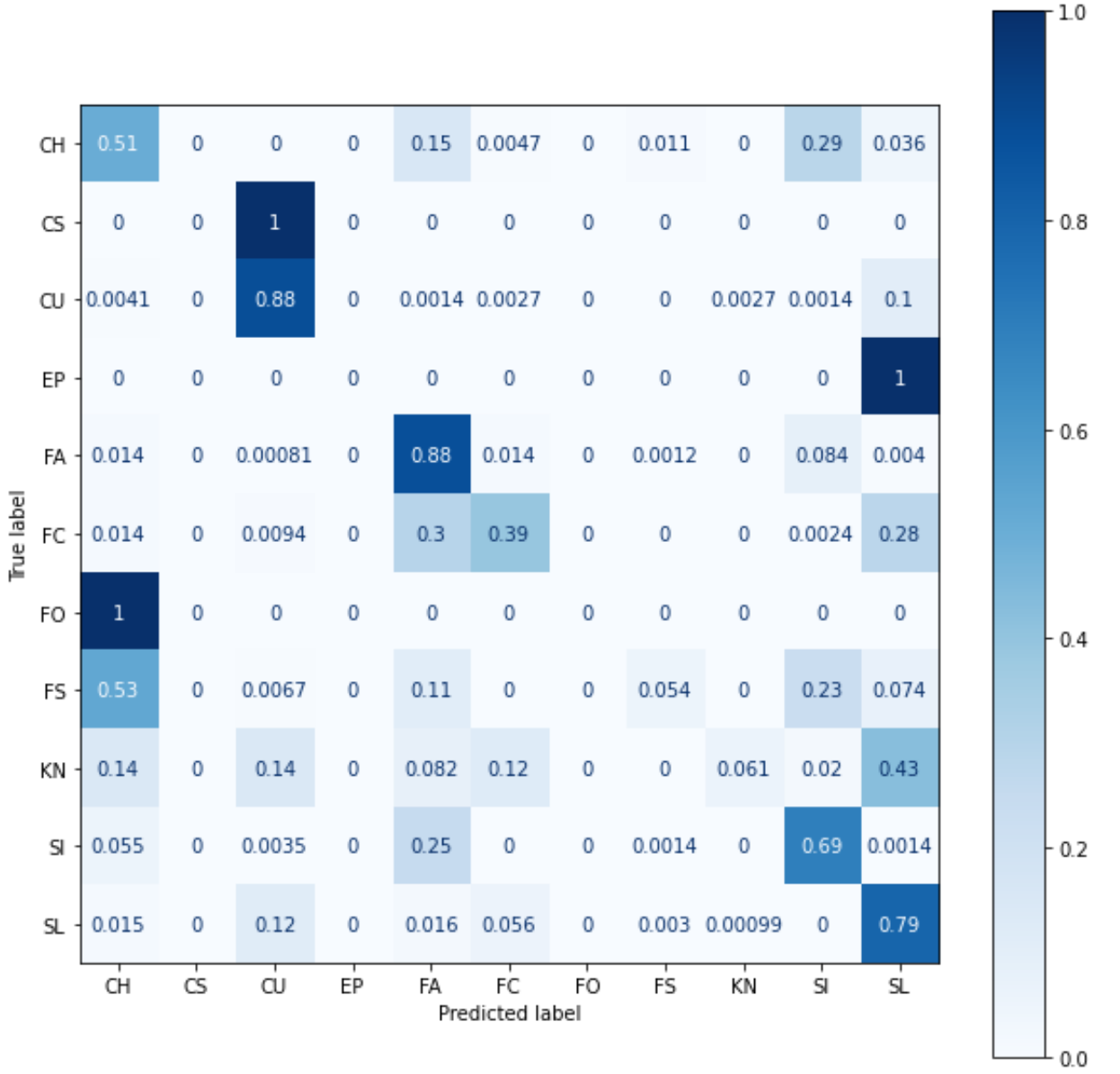

Figure 5: Confusion matrix of the true labels of pitch types and the ones predicted by our model. The numbers are normalized for each pitch type of the true labels. There are a few key observations that stand out. First is the pitch types that were never predicted by our model are guessed to belong to different categories (the 1's in the matrix). Second is that the five pitch types occurring most in the dataset, are well learned and predicted (CH, CU, FA, SI and SL) as you can see as a darker diagonal line. Third, the model seems to have trouble distinguishing between CH and FS, and between SL and KN.

4. Discussion

4.1 Summary of results

Our KNN model, which takes into account 9 nearest neighbors, is doing a moderate job (accuracy = 0. 7404) at classifying pitch type with all features, but when using only two features (spin direction and initial speed), we get a nearly max model performance of 0. 7334 accuracy with only two features (spin direction and initial speed). Our sequential feature selection indicated that spin direction was the feature that accounted for around 0.625 accuracy and adding starting speed increased the accuracy by around 0.10 points. Adding an additional third feature only bumped the accuracy another percentage point, so it appears that two features accounted for much of the KNN's predictions.

According to the confusion matrix, 0.88 of the fastballs are correctly classified, compared to other types of pitches whose performances are all below the accuracy for fastball. This result is unsurprising due to the fact that there are more fastballs in our dataset. Moreover, a potential reason for our model's suboptimal performance on predicting certain types of pitches (e.g. EP) is because there are so few observations (<10) of these pitch types in our dataset, and consequently our training model is biased by this skewed distribution of pitch types.

4.2 Comparison to other group's result

Using a random forest algorithm, the other group found that initial speed and horizontal break are the top two predictors of pitch types (Mehta *et al*., in press). This is partially in alignment with our finding that the starting velocity is one of the two variables in the best feature subset. However, in addition we also found that spin direction of the ball is another significant predictor of pitch type, and the horizontal break isn't included in our best feature subset.

4.3 What we have learned

Initially, we thought PCA would be a good method for dimensionality reduction. However, it makes interpretation of the components difficult and harder to relate to the original features. Furthermore, it's a method that might be further away from the way humans might predict the pitch type based on separate features that are easier to observe.

4.4 Impact of our results

We have demonstrated that not all features are equally important to the prediction of baseball pitch type. Notably, the initial speed is likely to be one of the most important features, as it is also confirmed by the result from another group. Thus, perhaps in the future, baseball commentators will incorporate initial speed of the ball into his/her consideration of pitch type.

4.5 Future Steps

We can improve our project next time by incorporating the pitch thrower as a feature. Since actual sports commentators know from each player which type of pitches they usually throw and which ones they are known for. Therefore, we expect that including this feature in a next analysis might improve the model's accuracy, if n_features = 3.

It would be interesting to compare the predictions from this KNN model to the classifications of pitches from a famous sports commentator, and see how well they overlap or where they differ.

Furthermore, the algorithm's 0.53 confusion of fast sliders (FS) for change-ups (CH) also warrants further investigation. It could be that these two particular types have high overlap in spin direction and initial speed, but there are other features in which they significantly differ that are not included in our analysis.

Lastly, we think the dataset could benefit from introducing another measurement for pitch type, which is reported by the pitcher who threw the ball. This will reduce circular argumentation, since the pitch types in this dataset were originally determined by sport commentators.

4.6 Closing statement

Computers are very good at considering 16 different features to make a prediction. However, humans have limited cognitive processing ability, and usually rely on slow motion replay. Therefore, to closely resemble human performance machines can benefit from data dimensionality reduction. In our case, we found that the top two features (initial speed and spin direction) got 0.74 accuracy. Adding a third feature, only increased the accuracy by 0.01. These features stand out on the field for any sports commentator, if we assume that spin direction is an indirect measurement for the general arc of the ball. Meaning that they might rely mostly on these two features as well to determine the pitch type, like our model did.

References

Mehta, I.,, Huang, Y., Cai, Y. (in press) *Cogs 260 mini project 1*.