Grading Contact Quality

A Description of Data Exploration, Data Cleaning, Overall Methodology & Design Decisions, Model Building & Model Selection, and Validation

Goal

Design an algorithm to grade quality of contact on a Trackman dataset of batted balls.

Overview

Although not a difficulty worded research objective, designing an algorithm to grade quality of contact without the specific contact "labels" provided in the dataset proved to be the essential task. Moreover, the algorithm in question must be able do the following two things: 1) identify where each observation (batted ball) falls in a specific grade, and 2) validate the effectiveness of the contact grade.

With this objective in mind, I implemented unsupervised clustering algorithms — k-means, agglomerative, and gaussian — to grade groups of contact using the three most important variables found in initial data analysis and reinforced through multiple iterations of clustering: exit velocity, launch angle, and distance. When clustering with all the variables including the categorical data, the clustering was not precise. As the number of variables decreased, the important numerical variables stood out. Essentially, clustering was able to classify exit velocity, distance, and launch angle into groups which could be statistically analyzed, plotted, and visualized. Once each cluster's characteristics were analyzed, it became possible to label the cluster with a quality contact grade.

Data Exploration/Data Analysis

The first step in any project with data is to explore, visualize, and analyze the data. In this step of the project, I felt that I had a leg up in understanding the data and variables because I have used Trackman before and am a baseball player who actively seeks out advanced statistical information. I am very familiar with Trackman and the variables which were provided in the dataset as I have analyzed my own data as a pitcher.

Thus, because of my previous experience with Trackman data, my understanding of batted ball variables which Trackman produces, and my deep familiarity with baseball analysis I found the exploration and initial analysis relatively straight forward. Moreover, I did not have to put much effort into using statistical analysis and visualization to understand the data, the key variables, and the relationships between the variables.

However, I still went through the process of doing some standard data exploration. For each of the numerical variables, I plotted them and described them with basic statistics to examine outliers, distribution, and their characteristics. In addition, I found it informative to count the frequency of occurrence for each categorical variable. I was able to find some "undefined" observations, noting that there must have been a data entry error or Trackman malfunction. Next, I wanted to explore what it meant for the key variables to slice the data by certain thresholds. I then used basic descriptive statistics to comprehend the significance of

positive play outcomes (hit type, play result) with specific thresholds for distance, exit velocity, and angle.

Next, I wanted to understand the difference between direction and bearing. I was able to do so through the Trackman website and by looking at some individual observations in the data. I then ran some simple correlations between numerical variables such as distance vs. exit velocity and exit velocity vs. launch angle. These values turned out to be essentially what I intuitively understood them to be as a consumer of baseball statistics. But I wanted to visualize these correlations, so I ended up plotting some of them to gather some insight to the plots behind the numbers.

Finally, I did some deeper data exploration and analysis by finding the optimal "band" of launch angles for balls travelling at 95 or greater mph that are home runs. Essentially, the question here was to find the optimal range of launch angles for balls that are hit really hard and end up being home runs. I sliced the data accordingly and was able to describe the launch angle column.

In summary, I felt that I was successful in data exploration for two specific reasons. I am comfortable with the baseball specific nature of this dataset and I was able to reinforce that understanding by executing sound data exploration. There was undoubtedly more room for data exploration, but I found it unnecessary because I understand this data well.

Data Cleaning

While often the single most time-consuming aspect of any data analysis project, the nature of this dataset was that it did not require much cleaning. However, because of the cluster analysis that I am going to run, I need to do some specific cleaning tasks. As a note, I made the decision of keeping bunts in the dataset because I felt that using bunts might help to gather five different clusters of data. Moreover, the bunt/weak contact cluster would be significantly different from other clusters. Additionally, the dataset is robust (large number of observations), so I did not have many reservations about removing nulls, outliers, etc. because they did not occur very often.

1. Although the dataset was relatively clean (inspected in data exploration), I did find some null/nan values. I removed those observations.
2. Next, I noticed some "undefined" values in data exploration. I figured out where these values were, how many there were for each categorical variable, and then used Boolean indexing to remove them.
3. Next, I wrote code to convert categorical variables into numerical values (manual hot encoding). However, after examining the data and running a few clusters, I came to the realization that this was not necessary. First, I did not want to use these variables to grade quality of contact. They are outcome-based variables and when grading quality of contact, focusing on the controllable of the hitter to grade is the most important. Second, if I wanted to encode this, I could have simply used a vectorized function from Sci Kit Learn which does one hot encoding for categorical variables.
4. I then worked to find outliers. I used information from descriptive statistics in the data exploration in conjunction with boxplots for the numerical variable to first, find outliers, and second, remove them using standard z-scoring methods. I removed outliers which had a z-score of greater or less than 3 (less than 1% of the data) for all of the numerical variables. This helped in removing unproductive launch angles,

distances, exit velocities, directions, and bearings. I then used boxplot and descriptive stats (max and min values) to see that there are still some unreasonable outliers left for distance and exit velocity (538 home run and 120+ exit velocity). While these occurrences are possible (see Luciano 119 home run recently), I want to remove them in an effort to produce more accurate clusters, especially when I use k-means clustering. I wanted to remain in the 99% of data range for z-score values (assuming a semblance of normal distribution) so after some guess and check, I used an absolute z-score value of 2.6 to remove all outlier observations from distance and an absolute z-score value of 2.8 to remove all outlier observations from exit velocity.

In summary, the data presented was already relatively clean (a few minor tasks here and there) and I was not planning to run any sort of regression or use the categorical variables in the clustering, so there was no need for implement one hot encoding and/or dummy variables.

Methodology & Design Decisions

This is probably the best section to talk about some of the design decisions I made in this process. As I alluded to in the overview section, I believe that building a robust and comprehensible contact quality grade algorithm depends heavily on the specific method used to achieve those results. As the dataset did not have a "contact quality grade" variable — which is understandable, since this is the given task — I did not have any y labels (used for classification and regression) or ground truths (used for clustering) to use for predicting or classifying. While I considered implementing KNeighborsClassifier, this is a supervised learning task and is also inherently part of the "classification" family. In addition, I would have needed to build some sort of list for classification, and this would have been opposed to the unsupervised nature of the goal. For example, by defining grades for contact quality myself, I would not be letting the data speak for itself. Machine Learning is so useful in this scenario because I do not need to guess contact quality bands, I can let unsupervised clustering do that task and only tell it how many clusters I want it to find. Thus, I landed on implementing three different unsupervised clustering algorithms, each different in clustering method. I decided to discard the "elbow method" of clustering and principal component analysis decision making because I knew that I wanted five clusters for grading contact quality and that I wanted two principal component analysis variables for plotting the clusters. Because of the small number of variables which I ended up using for clustering, I decided that simply scoring the resulting clustering and being okay with imperfectly defined clusters.

In terms of why clustering is such a useful algorithm for this task: it is able to interpret inherent structure in the data (inclusion of many variables possible) that the human eye has no way of seeing. These characteristics or groupings are essentially what I want to tease out; I want to find inherent structures in key variables of batted balls so that these structures can be exploited to grade quality of contact.

Another important note that fits well in this section: grading contact quality from this dataset requires baseball intuition. When thinking about grading contact quality, I do not care about the play result, the tagged hit type, the pitcher handedness, the hitter handedness, the balls, or the strikes. If I am interested in how each contact quality grade differs in each of these variables, it is something I can do afterward. Contact quality should only be graded on the controllable of each swing. Moreover, using the actual outputs from each batted ball like launch angle, distance, and exit velocity define controllable variables from the hitter's perspective. Also,

if I were to use the hit type or play result, I would be evaluating the quality of contact based on variables which define the results of a hit/a play on a baseball field. This is not an ideal way of assessing contact quality because it uses an uncontrollable from the action of swinging.

Model Building & Model Selection

In my opinion, the most user-friendly and best package for Machine Learning in general is Sci Kit Learn. Not only are the pathways, pipelines, and write-ability of the code simple, the website provides fantastic model analysis and documentation help. In addition, I am most comfortable with Sci Kit Learn.

As I mentioned before, I wanted to run three different types of clustering algorithms: k-means, agglomerative hierarchical, and gaussian mixture models. K-means is the most popular algorithm when it comes to clustering. It essentially works by attempting to pick centroids (central data points) which reduce the sum-of-squares within that cluster. Agglomerative hierarchical clustering works by merging similar clusters until the specified number of clusters is returned; oftentimes, it can be similar to k-means clustering. Gaussian clustering — which is a little bit more sophisticated as it accounts for variance — can deal with multiple distributions of data within the clustering variables.

The first step in using Sci Kit Learn is to build pipelines through which the algorithms will run. This enables easy usage and readability. I built two different pipelines, one for scaling and pre-processing and the other for the specific cluster algorithm. Then, I built another pipeline which called the pre-processing pipeline and the clustering pipeline. I did this to fit (run the model) all of the aspects of the model in one line of code.

First, I used a standard scaler to ensure that the machine learning algorithm did not behave poorly — this is oftentimes necessary. The standard scaler removes the mean and scales with variance. Next, I used Principal Component Analysis (PCA) to reduce the dimensionality of the dataset by selecting two components. PCA reduces the dimensionality so that the clustering can run better and more accurately while minimizing data loss. By using PCA I could also call each of the components and place them into columns so that they could be plotted in different colors to visualize clusters on a 2-D plane.

The last step is slicing the cleaned dataset for the variables that I want to cluster. While I started off clustering all of the numerical data, I soon realized — after analyzing the descriptive stats of each cluster's key variables and visualizing the clusters — that including non-key numerical variables such as direction and bearing were adding too much noise. In other words, these two variables did not provide insight or structure the clusters in any meaningful way. In fact, as I will discuss later, they brought down the "score" of the clustering. Thus, I simply selected the three key variables which the combined data exploration and baseball knowledge illuminated as meaningful: exit velocity, launch angle, and distance.

After building the model and slicing the dataset, I was finally ready to run the three clustering algorithms, save the results back to the dataset, plot the clusters by utilizing the two PCA components, and score the model. The method which I am using for scoring the clustering algorithms is fairly popular and it scores in an intuitive manner. It is called Silhouette Scoring and it measures how successful the clustering was. Silhouette scoring works by measuring how similar an observation is to its own cluster. In other words, how well do the clusters fit together and, therefore, separately from one another. Silhouette scoring ranges from -1 to +1, where a higher value indicates that the observations are well matched for their own clusters, while a score of -1 indicates the opposite.

After running the models, k-means and hierarchical clustering each had extremely similar scores while the gaussian clustering had a significantly lower score. The scores for each model are as follows:

1. K-Means Clustering Score = 0.4
2. Agglomerative Clustering Score = 0.37
3. Gaussian Clustering Score = 0.25

While these scores are not values of +1 (which is pretty rare for a naturally occurring dataset), they are also not -1. Essentially, the scores indicate that the clustering is indeed successful, but that there is a lack of distinction for some observations on the edges of the cluster. This is largely to be expected from such a large dataset with so many similar data points at every spot along the distribution.

In terms of model selection, I ended up selecting the k-means clustering algorithm for both its score, its plot visualization, and its ability to run all observations through the algorithm. While the scoring enables us to select the best model, I still need to validate the model itself.

Validation & Contact Quality Grade Labeling

Essential to the goal of grading quality of contact is turning the clusters into actual "grades." At the end of the day, I was able to successfully attribute a "grade" to each batted ball. I first went about statistically describing each cluster by the three numerical variables I used to cluster them. Computing the average and median of each cluster's exit velocity, launch angle, and batted balls enabled me to begin the process of understanding each cluster's potential contact quality grade. I then plotted these results as bar charts and tables. This allowed for a statistical comparison through visualization. I also wanted to understand how many home runs, singles, doubles, etc. were in each individual cluster. Essentially, I wanted to get a picture of what the batted ball and play results were for each cluster, even though they were not used in the clustering. I wrote code to extract the value counts of the categorical data and then plot them into a table. Finally, and probably the most informative of the validation steps, I exported the dataset with the k-means clusters into Tableau (a data visualization software) to gather an even deeper understanding of how each cluster could be described. In addition, I created a table with the 1st and 3rd Quartiles of each variable for each cluster to be able to understand the distribution of means.

At the end of the day, I was able to come up with the following contact quality grades based on the plotting, statistical analysis, and visualization:

1. **Cluster 0 = C**
2. **Cluster 1 = B**
3. **Cluster 2 = F**
4. **Cluster 3 = D**
5. **Cluster 4 = A**

The most interesting aspect of labeling the clusters with a contact quality grade was the degree to which I had to make decisions as to which variables to prioritize. I eventually decided that exit velocity and launch angle were the two best variables to describe contact quality because they indicated just how hard the ball was hit and how close a given launch angle was to being optimal. Distance is merely a result of exit velocity and launch angle. Thus, I used baseball intuition and experience to grade out the clusters (lots of charts and tables provided in summary).