Conor Desmond

I checked for null values and there were only some in a column that I was not going to use, therefore I did not remove any nulls. After that, I made a data frame where I grouped the initial data frame by Pitcher Key and got the mean for every variable based on each pitcher's pitches. I will use some of these columns later for determining the probability the dewpoint affected the pitch. Then, I made a data frame where I sorted the initial data frame by Pitcher Key, so I could add the probability values to the corresponding pitchers in a new column. I then only included the stats I would use for determining the dewpoint affected.

I thought if the dewpoint affected a pitch, it would change values of some variables from their usual values. Therefore, to determine the dewpoint affected, I would subtract the mean of a variable for a pitcher from the value of that variable for a specific pitch thrown by that pitcher. I would do that for all 9 stats involved in my calculation. I would then take the absolute value of those values and divide by 1312 to get probability values between 0 and 1, where all are less than 1. I added the dewpoint affected values to the sorted data frame in a new column named dewpoint affected. I would repeat this process for all the pitchers. The 9 stats I used were INDUCED_VERTICAL_BREAK', 'HORIZONTAL_BREAK', 'SPIN_RATE_ABSOLUTE', 'RELEASE_SPEED', 'RELEASE_SIDE', 'RELEASE_HEIGHT', 'RELEASE_EXTENSION', 'HORIZONTAL_APPROACH_ANGLE', and 'VERTICAL_APPROACH_ANGLE.

Next, I made a graph to check the distribution of dewpoint affected values. The distribution is skewed right meaning most pitches were likely not affected by. After that, I made a scatter plot to see how the dewpoint affected values were distributed for each pitcher. Most pitchers did not have a lot of values above 0.6 in this scatter plot, but there were a lot of points between 0.4 and 0.6 in this scatterplot.

I made some machine learning models to predict dewpoint affected in case the team collects new data they want to use to predict dewpoint affected. I made a Scores function to get the MAE, RMSE, normalized MAE, normalized RMSE, and get the average of the normalized scores. I made a function that would make a ridge, lasso, or elastic net model depending on the input, split the data into a training and test set, and get the scores on the training and test data by calling the scores function. I then made a function that uses autoML to make autoML models. This function prints the models, splits the data into a training and test set, and gets the scores on the training and test data by calling the scores function. The autoML ensemble was the most accurate, with an avg. Normalized Score of 64.6% on the test data, but the difference of avg normalized scores, between the test and training sets, was 30.176%. Not very accurate and it overfit, but that was the best I came up with.