Me and one of my intern team members Bennett Stice worked on and created this code together

Pitch_Seperated_Voting_Class_Model_Hyperparametertuned.py file:

This was a file of code that made the pitch grader model the coaches wanted. We used tons of data over multiple years from the MLB via Baseball Savant: Statcast (Baseball Savant: Statcast, Trending MLB Players and Visualizations | baseballsavant.com) to create the model off of. We first created and stored data from the Savant website into a data frame and then reorganized the data and cleaned it up to what we needed for our model. We also sorted data from the data frame into numerical and categorical columns. Next, we further sorted those two columns of all the variables based on certain pitch types they were that we were looking for. After all of that we trained-test-split the data from the two columns and scaled the data. We then applied various machine learning techniques to the data to create different pitch graders for each type of pitch and validated the scores giving us our training accuracy for each pitch type for specific graders. After we had training scores for each machine learning technique for each pitch we further improved our model by taking those models and applying voting classifiers to each pitch to get essentially the best possible model of all worlds by combining all of the machine learning model's scores together to get a score and model created through what either was the most common result among all the machine learning model's or an average of the probable score(s) the models generated. Once the final model (Prediction Model) was created, it would then be sent into a pickle file to be stored for later use.

predictor_it_3_Hyperparametertuned.py file:

This file of code first uploaded the final prediction model created from the file of code above. Second, it opens the json file that contained the line of pitch data from the pitch wanted graded and loaded the data into an array which then the correct prediction model (depended on type of pitch thrown) was applied. A score was then generated and transferred into a number score from a probability. If the score was a 50 or above, the pitch was graded as good. If the score was below 50, the pitch was graded as a bad pitch.

Overall:
We created a model that could grade past and possible future pitches thrown.