Q1. is_strike

To determine which pitches were balls and which were strikes in the test set, a model was created with the training data set and applied to the test data.

Data Cleaning
The "is_strike" variable needed to be set. Values of "pitch_call" that were "HitByPitch", "BallCalled", and "BallIntentional" were set to 0. All other values were set to 1.

Running a summary on the numerical values in the training data set showed there were no outliers. The columns "pitch_id" and "pitch_call" were removed as they are not in the test data set. The "y55" column was removed, as all of its values were the same for each row, and the "tilt" column was removed as it provided the same information as the "spin_axis" column and was more precise. Another summary showed some columns had more NaN values than others. The rows with any NaN's were removed, which was less than 2% of all data.

Modeling
All categorical data was one-hot encoded, then all data was standardized by removing the mean and scaling to unit variance. The data was split into a training data set and testing data set. Ensembles of decision tree methods can automatically provide estimates of feature importance. The data was imbalanced as the proportion of strikes was greater than the proportion of balls (364884 strikes: 207752 balls). Thus, the f1 score was a more important metric than accuracy. After running a Random Forest algorithm with all features, only the numeric features were shown to be important. Additional features did not increase the f1 score or accuracy score much so they were omitted.
With the features selected, Logistic Regression, Random Forest, and Gradient Boosting algorithms were run. Gradient Boosting provided the best f1, accuracy, and AUC scores.

Prediction
When applying the model to the 2020-test data, any "nan" values were replaced with the mean of the column.

Improving Model with More Resources
With more resources, Random Forest could be optimized by testing different estimator (boosting stage) values and different maximum depth values. Gradient Boosting could also be optimized with different estimator values. Extreme Gradient Boosting uses more approximations to find the best tree model and it could be run to improve the model.