Description of Files and Notebooks Included in this Folder

Problem that we are addressing:

- Binary classification problem of whether each punt in an NFL game is a fair catch or not.

CSV Files

- We downloaded the tracking2018, tracking2019, tracking2020, plays, games, and PFFscouting .csv files from Kaggle initially
    - We then manipulated these files in the steps listed below in our Jupyter Notebooks

Notebooks

The dataimport notebook contains the process of importing all of the kaggle datasets and merging the data frames based on game and play. Then we kept only the punt plays and got rid of any punts that were out of bounds and blocked because punts that go out of bounds or are blocked are incapable of being caught by the receiving team so they don't have the opportunity to call a fair catch or not.

The dataorganization notebook took the resulting data frame from the previous notebook punt_df.csv. We start by adding a column containing a unique ID for each individual play using the play's game ID and the play ID. Then we made a copy of the data frame excluding the tracking data and called it punt_df2 that contains only the first row for each of the unique play IDs. Then we added 1,078 columns to represent the tracking data for all 22 players at 7 time points for each punt. Then using a nested loop we stored the corresponding values from punt_df into punt_df2. You must have punt_df to run this notebook.

The modelBuilding_1 notebook takes punt_df2 and does one hot encoding for our categorical data. We then perform a test-train split and scale to prepare for model building. Next, we build numerous different models and look at the results of each for comparison. To run this notebook you must have punt_df2.

The dataScaling&ModelBalanced notebook takes punt_df2 and does one hot encoding for our categorical data. We then balance the data to have an equal number of fair catches and not fair catches. We then perform a test-train split and scale to prepare for model building. Next, we build numerous different models and look at the results of each for comparison. To run this notebook you must have punt_df2.

Featureselecton takes the modified dataset from the modelbuilding_1 named dfupdated and is just used as an exploration of our feature importance and correlations.

Footballvisual uses tracking data from kaggle either tracking2018, tracking2020 or tracking2020, based on the input parameter of our animation function. This notebook takes the tracking data and runs it through a function to create an animation and then exports the animation into an

mp4 video. This notebook was referenced from someone else's code in the 2021 kaggle data bowl.

PlayerMapping was made in order to see if there is a pattern in the order the players are listed in the tracking data in hopes to find insight in the pattern our model was picking up. This uses tracking2018.

Further exploration notebooks

DSC500_returns uses the same data punt_df2 as model_building1 to prepare the data for linear regression to predict kick return yardage and a logistic regression binary classification of whether or not a punt will result in a return or not.

Multi_class notebook uses the same data punt_df2 and is then prepared for a multi class random forest classifier.