NFL Play Predictor - Data Wrangling

Viraj P. Modak

To recap, the objective of this project is to design an NFL offense play-predictor which can anticipate whether a play will be a Run or a Pass given the game situation. For this purpose, the **"nflscrap-R play-by-play"** data will be used to train an ML model. The data is available freely on GitHub at the following address:

https://github.com/ryurko/nflscrapR-data/tree/master/games_data/regular_season

The available data had to be cleaned-up and relevant features needed to be extracted to a format which can then be used to train an ML model. A brief summary is presented here:

1. CSV files were read into a single data frame. These include the player roster and the actual play by play data

2. The dataframe was then cleaned up to remove entries which are not plays. These include Timeouts, Two-minute warnings, End of game/QTR, game suspensions and resumptions. The str.startswith function was used for this purpose This also includes plays which do NOT have a time clock associated with them. The data is read as string, which caused the missing values to be read as empty strings. These values were replaced by np.nan and then removed using the dropna method of a data frame.

Additional features such as play clock, run/pass performance and the dual threat rating were calculated using individual functions

3. The play_clock was represented as either MM:SS or MM:SS:00 and was read as string in the original data frame. This was then split into minutes and seconds using the str.split function and the minutes and the seconds were then converted to float using astype. The final play clock in minutes was then calculated using a simple arithmetic operation over the floats: Minutes = Minutes+Seconds/60

4. Whether a play is a pass/run can also depend on how the teams pass/run performance has been until that point in the game. To calculate the pass yardage, the string value was first converted to numeric using pd.numeric. Following this, groupby and apply functions were used to

calculate the cumulative pass completion % and the cumulative pass yardage - grouped by team and game. There were stray non-numeric or 'NA' values for cases such as incomplete passes. These were addressed using errors='coerce' wherever applicable. This operation was repeated to calculate the cumulative run yards grouped by team and game.

5. To calculate the dual threat rating of a team, the dual threat performance of individual quarterbacks was first calculated. For this, total yardage and run attempts were calculated for QBs over the entire season using the groupby and agg functions. Now the dual threat factor was then calculated using a simple arithmetic calculation as a product of yardage and run attempts, each normalized by the games played. The final dual threat rating was a percentile count for all QBs going from 0 through 1.

6. The dual threat rating calculation for a team in a game proved quite tricky because of the possibility of multiple QBs playing in the game. The QB-specific dual threat ratings were first converted to a default dictionary using zip and defaultdict which was then called using the map function. The contribution of each QB for a team in the game was estimated using the pass count, once again using groupby and agg. The team's dual threat rating was then calculated as a weighted average of the QBs' dual threat rating over the contribution. A separate local function was written for this purpose and then called using apply.

The final data frame was then saved as a CSV file. This will now be used to perform data analysis and for training the model. No outliers were noted as such in the database which could have caused problems in the data wrangling process. More will be known once exploratory data analysis is done on the final cleaned up dataset.