NFL Sabermetrics
project-050

[National Football League Prediction and Analysis using Python]

Sanket Ninawe
Indiana University
sninawe@iu.edu
gitlab: sninawe

Peter Zale
Indiana University
pzale@indiana.edu
gitlab: pzale

ABSTRACT
In this project, we analyze the 2015 NFL season statistics. Using the nfldb data set along with Python and Tableau several models and visualizations were produced in order better understand the outcome of the season and to be able to forecast and predict results. We focused on a few statistical methods for analyzing this data.

General Terms
Statistics, Analytics

Keywords
NFL, python, nfldb

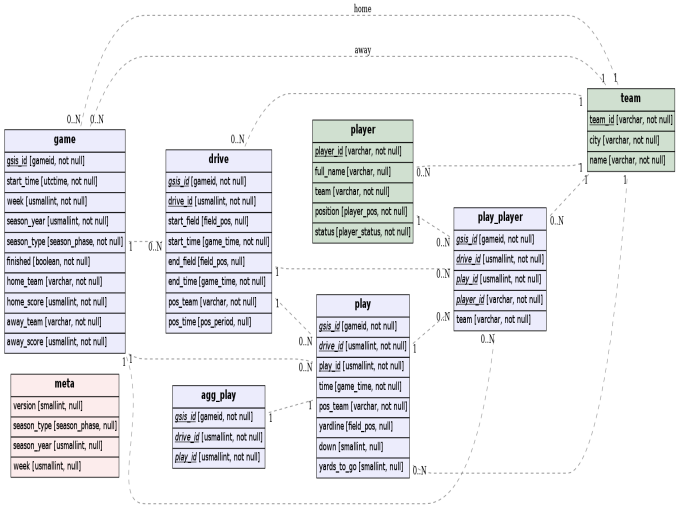
1. INTRODUCTION
In football player position, player run time, scoring, player data, game data is documented at a very granular level which is one of the reasons various statistics can be developed and tracked for NFL games. It is heavily a team sport and each play relies on eleven guys on offense coming together to advance the ball against eleven guys on defense. In this project, we are going to run analysis using play-by-play data to gain insight on what makes NFL teams successful. We are building NFL statistics for enthusiasts which will help them derive insights from the historical team and player data. To analyze the NFL data there are various data sources that are available in python like nfldb, nflwin etc. The process of calculating statistics, statistical correlations and prediction models involve using SQL, plots, machine learning algorithms, and Tableau. We have used various python libraries, supervised and unsupervised learning methods to predict probabilities.

Statistical analysis of NFL data has been popular for many years but with the rise of fantasy sports there is more of a demand for it due to participants looking for these charts and visualizations to help them set their teams each week. These visualizations create an easy way to see the players who are leading the NFL in categories such as reception yards, rushing yards, or touchdowns which are big scoring areas in fantasy football.

Another reason for this rise of analytics in sports is the organizations themselves beginning to leverage them. The most famous example is from the 2003 book turned movie "Moneyball," by Michael Lewis, which showcased the Oakland A's use of analytics to field a team based on player evaluation that was specially formulated. The A's general manager, Billy Beane, came up with this method and after their success it skyrocketted in popularity [6].

We have used nfldb database which is a library to manage and update NFL data. nfldb is a relational database bundled with a Python module to quickly and conveniently query and update the database with data from active games. Data is imported from nflgame, which in turn gets its data from a JSON feed on NFL.com's live GameCenter pages. This data includes, but is not limited to, game schedules, scores, rosters and play-by-play data for every preseason, regular season and postseason game dating back to 2009. [3]

nfldb Entity-Relationship diagram (condensed)



2. K-MEANS CLUSTERING FOR PLAYER POSITION

Background -

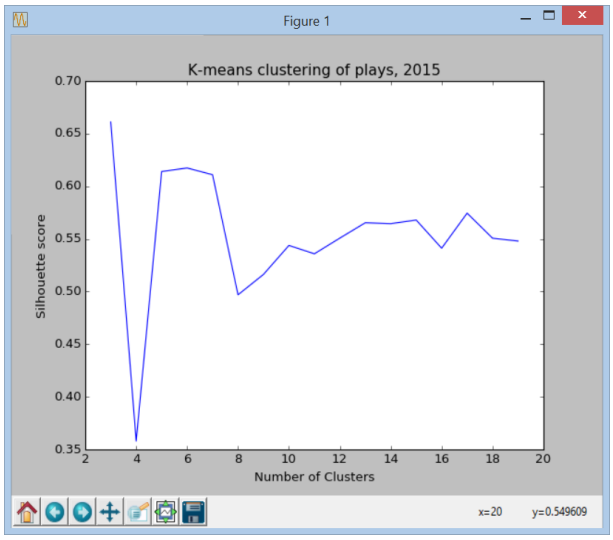
In this model, we trying to use unsupervised learning method to predict which players will be best suited for playing at a particular position. We have used unsupervised learning as we did not had any classifier label for creating training dataset. So the best possible algorithm to predict is unsupervised k-means clustering.

In this particular type of machine learning algorithm, clusters/groups with the give features are created.

This algorithm creates the centroid of the cluster assuming datapoints are closest to the centroid. This is then done multiple times until a correct centroid is created which groups the datapoints in correct group. The distance from this centroid is used to calculate the prediction of the algorithm. In unsupervised learning the prediction outcomes are generated by the algorithm itself for the training set. These classifiers are then used to predict outcomes on the test set.

k-means clustering procedure -

So here for predicting the best players for a particular position we have used SQL query to pull the data from nfldb database. We used python SQLAlchemy library to write SQL query on the 3 tables (*play_player, player and game*). We have used 2015 data from 'Regular' and 'Postseason' seasons. We then group by all the selected player categories like passing_yds, receiving_yds, defense_ffum etc for each player(full_name). Then, we have used silhouette_score library to find out what is the best number of cluster we should define for our purpose. This function helps us give highest efficiency in predictions when we define number of clusters with highest silhouette_score. We then fit our prediction features into k-means clusters creating number of clusters derived from above process. This creates the cluster centroids which can be used for predicting the best players for a particular position. We can choose the centroid values and the player distance from this centroid to see which players are best suited for a particular position. We can also need to add player position along with the centroid distance of the player to find out the best player.



Results and validation -

After running the cluster analysis and selecting the cluster with centroid values and player position, the model predicted the players which are best suited for that position. This can be verified by actually looking at the 2016 statistics of the player position. We tried this with several examples which produced correct results most of the times.

3. WIN PROBABILITY

Background -

In this model we have used supervised learning algorithm to predict win probability of the teams based on some of the calculated statistics. We split the dataset randomly into training set(70%) and test set(30%). Logistic Regression algorithm is used to predict the win probability. The binary classifier from the training set is used to estimate the probability of a binary response based on one or more predictor features. So our case we have defined winning team as 1 and losing team as 0 based on the team_score. We have calculated 5 statistics based on industry standards and have used them as predictors in the logistic model. Logistic model works on coming up with weights for each of the predictor features and then maximizing the summation of the weight and feature values, so that feature that is most significantly contributing to classifier distribution is given more weight and predictions are more accurate. Results of the logit model can only be binary, so we can predict win or loss using this model. The accuracy of the model depends on selecting the right number of training set. We have to make sure the training set is not skewed. An accuracy between 70-80% is considered good typically.

There are other ways and methods or statistical categories on which win probabilities can be predicted. This method used in our analysis is developed based on one the approaches selected of the various methods.

Logistic regression procedure -

Here we have again used SQLAlchemy to pull data from nfldb database required for our prediction. The tables which are used for this purpose are (*play_player*, *play*, *drive*, *player and game*). We have limited the number of rows from the SQL query to be 99999. After a lot of testing, it was found out that these number of records provide highest accuracy of the model. We have used 2015 data from 'Regular' and 'Postseason' seasons.

We are calculating the team which has won the game based on the home_score and away_score. We are assigning 1 to winning team and 0 to losing team. Some of the statistical values calculated in this model are - [2]

Offensive Passing Rate = (offensive pass yds - sack yds) / pass plays

Offensive Running Rate = offensive run yds / run plays

Offensive Interception Rate = offensive interceptions / pass attempts

Offensive Fumble Rate = fumbles / offensive plays

Team Penalty Rate = team penalty yds / total plays

Some of the denominators of these statistics can be 0, which will result in Null values. So we are replacing all the null values in these fields with zeros. Next we are fitting the statistical categories as the predictor features in the logistic regression python library. The classifier is the team_won variable which has binary values. We are using ac-

curacy_score python library to calculate the accuracy score on the test set.

Results and validation -

After fitting the training set, logistic regression produces an accuracy of 74%. We verified the accuracy on the test set and also validated the test set results with the actual results and found accuracy to be 74%. Thus using the above mentioned team statistics we can predict if a team will win or lose the game. Team Penalty Rate is the highest contributor to predicting win probability based on the coefficient values. This program takes longer to run because of the SQL joins. We have tried to optimize the query, but further optimization needs to be done for faster processing. [4]

4. PLAYER CONTRIBUTION TO WIN

Background -

In this statistic we are trying to find out which player in the team has contributed more toward the win. This will help coaches to select the team rooster for next upcoming games. Here we have tried to create offensive and defense play points based on the player categories and points they earn during play time. These points are indicative of how well or bad a player has performed in the current game if that team has won the game. We have ranked each player who has played in the winning team according to the points scored in various activities each player has performed. The highest the rank, higher is the contribution of the player in the team win. These ranking will help coaches and selectors to choose the player available for selection.

Player Ranking procedure -

Here we have again used SQLAlchemy to pull data from nfldb database required for our prediction. The tables which are used for this purpose are (*play_player*, *play*, *player and game*). We have used 2015 data from 'Regular' seasons. We have separated the results in 2 categories when away team has won and when home team has won based on hone_score and away_score variables. Also, in each of these categories we have selected player only belonging to the winning team based on player team and winning team. We have done this so that we get players from the winning team where home team has won and another set of players where away team has won the games.

Offensive play points and defensive play points are calculated based on categories in which offensive or defensive player can earn points or defend points. Example: defense interception before a touchdown can save 6 points for a team where as a player from the offensive team does a touchdown while receiving the ball, which can earn him 6 points. Similarly we have assigned points for each of the significant categories and summed them together which can create play points in that game for each of the player. Our aim is to create player points whether he is offensive or defensive per game, so that we can rank him if he is on the winning team. [5]

We have done group by on full_name, home_team and away_team so that we have play points aggregated per player for each game. Here we are assuming that a combination of home_team and away_team is unique in the season.

So the 2 separate categories of away team winner and home team winner are created using the SQLAlchemy queries. Next is we are adding the offensive and defensive based on the assumption that a player can be defensive or offensive and can be both as total points. This will help us when we are ranking each player. We then rank each player as the offensive/defensive points the player has earned when grouped by each game.i.e. combination of home_team and away_team. We have used python method = min which is used to rank observations which start from minimum and we are using another option ascending = False, which ranks the maximum score as 1 and minimum as n ranks in the group.

Results and validation -

We have done validation on each player point calculation and uniqueness of the home and away teams. We have also done validation on the ranking based on the points earned in the game by each player. We checked if the teams had as many as 24 players playing in the game which can give higher player ranks. We have tried to optimize the queries as much as possible though there is some scope of optimization.

5. STATISTICAL CORRELATION

Background -

When calculating the statistical correlation, it is preferred to use the Pearson correlation coefficient. According to [8], the correlation coefficient is a "measure of the linear dependence between two variables X and Y, giving a value between +1 and -1 inclusive, where 1 is total positive linear correlation, 0 is no linear correlation, and -1 is total negative linear correlation."

The most important stat in any sport is wins. Every player on each team and every member of each organization is going to do all that they can to get the most wins. Statistical analysis has helped many sports organizations when they do decide to leverage it. In this project we looked through the NFL statistical categories in order to uncover some of the key reasons that help a team produce wins. We chose to stick to offensive performance statistics to see if we could find correlation. In our analysis, we chose total team passing yards and total team rushing yards from the 2015 NFL season and compared each one to team winning percentage and then compared passing yards to rushing yards.

For this step of our analysis, we wanted to visualize what factors help teams end with winning seasons. The categories we looked at are directly related to the team as a whole instead of focusing on the statistical categories that are related to individual players since we are focusing on team wins throughout the season. Scatter plots were chosen as the visualization for these figures since they easily show

relationships between many points. Also, the scipy stats module in Python was used to calculate the correlation coefficient between the two categories on each plot.

Statistical Correlation Analysis -

Figure 1

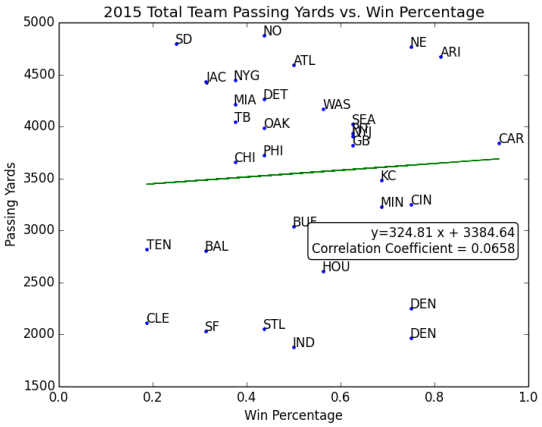
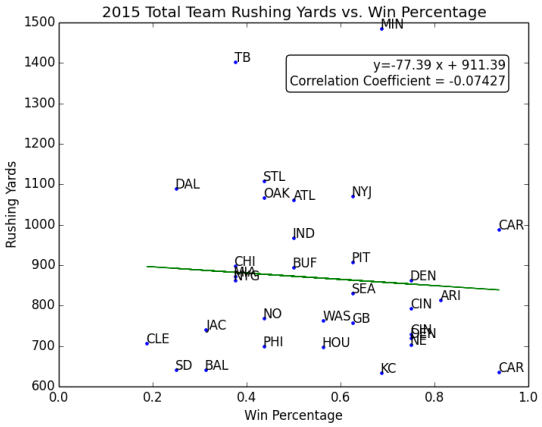


Figure 1 shows the relationship between total team passing yards and win percentage for each team from the 2015 NFL season. As you can see from the trend line one the plot, this relationship is positive. Also, the correlation coefficient labeled on the plot is 0.0658 which, although it's a small number, it is positive so that leads us to conclude that there is a correlation between these two statistical categories.

Looking at figure 1 we can see that the Carolina Panthers are in the the top right quadrant of this plot and they went on to the Super Bowl after the 2015 season.

Figure 2



Our next plot, figure 2, shows total team rushing yards versus team win percentage. The trend line one in figure 2 is downward sloping and the correlation coefficient is -0.07427 which conclude that there is a negative relationship between these two categories. There were several errors in this data set that resulted in two data points for several teams which we were not able to resolve. Even with this error the majority of the teams appear correctly which allows us to draw this conclusion.

Figure 3

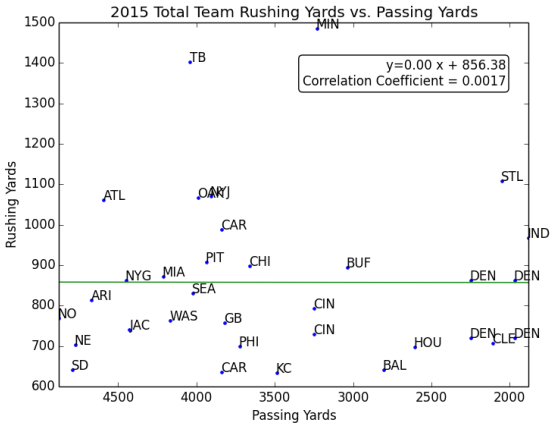


Figure 3 was included out of our own curiosity to see if a team's rushing yards were correlated to their passing yards. As you can see in figure 3, the trend line is pretty much perfectly horizontal and the correlation coefficient is very close to zero. This query also gave us trouble and produced multiple results for a few of the teams. Taking the error into account, we still conclude that there is no relationship between passing yards and rushing yards.

Results and validation -

Although, we found that total team passing yards in a season directly correlate to win percentage we have to remember the saying in statistical analysis that correlation does not imply causation. Just as [7] points out there is often another underlying variable which explains the correlation. Solely relying on the numbers to predict sports isn't ever going to be absolutely perfect because in the end, there is a human element to every aspect of the game that could change the result in an instant. A player injury or an athlete having more or less "drive" than the other are both things that can't be quantified so we can't take those aspects into account when performing statistical analysis.

Brian Burke runs a website names "Advanced Football Analytics" which analyzes all aspects of the NFL. Brian has an article about these correlations and his results were different from ours in that he found a positive correlation from both rushing and passing during a season with relation to number of wins [1]. He doesn't include the variables and exact numbers that he was using to produce these results but judging by his resume we are led to believe that he has a more robust model than ours.

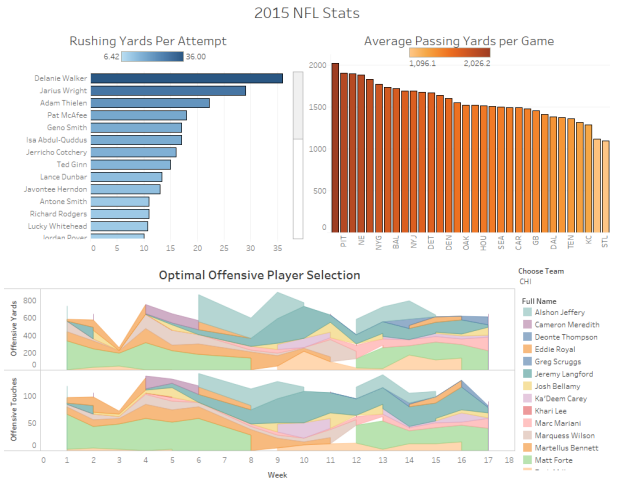
6. TABLEAU VISUALIZATION

Background -

Tableau, founded in 2003, is a relatively new way of conducting data analysis and producing visualizations.

We decided to leverage it's robust interface and analyzing tools to help us view offensive player performance. Tableau allows you to create interactive visualizations so you can

Figure 4



To see the full interactive interactive version of this dashboard please click here.

Figure 4 is made up of three different plots that all relate to different offensive statistical categories; rushing yards per attempt for offensive players, average passing yards per game by team, and the bottom section allows you to select a team and see who lead that team in offensive possessions each week of the 2015 season. This dashboard could be used by fantasy football enthusiast and NFL team managers to do scouting on an opponent or to look at the the players on their own team.

7. CONCLUSION

Statistical analysis in the NFL has come a long way since the days of using pen and paper to keep track of statistics. We now have access to powerful analytical tools such as Python and Tableau and the tools will continue to become more powerful as computing becomes more powerful. Leveraging this information can truly give you an advantage over the opponent by being able to predict and forecast results of games and being able to pick position of the game to focus on such as rushing, passing, and other aspects of the NFL game.

8. REFERENCES

[1] B. Burke. What makes teams win? website, 2007. Accessed 12/1/16.

[2] B. Burke. Game model coefficients, 2016.

[3] A. Gallant. Burntsushi. *github*, 2016.

[4] J. Markham. Game model coefficients. *github*, 2016.

[5] A. Schechtman-Rook. Nflwin. *github*, 2016.

[6] L. Steinberg. Changing the game: The rise of sports analytics. website, 2015. Accessed 11/30/16.

[7] C. Stuart. Correlating passing stats with wins. website, 2012. Accessed 11/30/2016.

[8] Wikipedia. Pearson product-moment correlation coefficient — wikipedia, the free encyclopedia, 2016. [Online; accessed 24-November-2016].