Predicting NFL Play Calls for Optimum Success

Introduction

Project Idea and Reason

In the NFL, a single play-call can be the difference between a win or a loss. A designed run play may work well in a certain situation, while it may work terribly in a different situation. Therefore, it is vital to make use of analytics to know when and where these calls need to be made. The NFL is fairly new to analytics, so you often see new innovations and projects geared towards making insightful decisions. For example, Kaggle hosts an annual NFL data scientist competition called the "Big Data Bowl" (National Football League, n.d.). It is a football analytics competition affording college students and professionals the opportunity to utilize historical data sets of the same player tracking data used by teams and suggest innovations about how football is played and coached. The winners are invited to Scouting combine to present their project to coaches. From this competition you can find many different viewpoints of analyst around the world and some of those projects gave me the inspiration for my own project. Some of my inspiration also came from social media as well. There is a large community of sports analysts who interact and encourage new ideas. Ron Yurko, who's nflscrapr package has been used in many varieties) and Michael Lopez (Director of Analytics for the NFL) are prime examples.

In this project, we will be looking in the lens of an NFL Defensive Coordinator. They are responsible for setting up the defense and ensuring the have they correct personnel on the field at any point the team is on defense. "The responsibilities of defensive coordinators include leading the defense during practices, designing plays and strategy and calling defensive plays during football games." (Rookie Road, n.d.). This project will seek to create a model that can satisfactorily predict the next play type the offense will run. This will allow defensive coordinators to prepare and shift their defense in an effort to stymie any play.

Dataset

Publicly available NFL data sources has been a major obstacle in the creation of modern, reproducible research in football analytics. Maksim Horowitz, Ron Yurko, and Sam Ventura built and released nflscrapR an R package which uses an API maintained by the NFL to scrape, clean, parse, and output clean datasets at the individual play, player, game, and season levels. This dataset has full game play-by -play from the 2009 NFL season to the 2018 NFL season. There are 255 columns and 449371 rows. The dataset is rather large, with many unnecessary columns that will have no impact on the model to be constructed.

The dataset contains information such as:

a) General game information
b) Home/Away team
c) Type of play
d) Player personnel on the field for any play
e) Time of play
f) Result of play

Motivation

The motivation for this project was two-fold. First, being an avid sports fan, I'm constantly looking for ways to apply machine learning techniques to the sports realm. Sports analytics has also

rapidly ingrained itself into major sports organizations. The most successful teams are constantly looking for answers backed by numbers, to give them the best chance at winning. An article on Forbes says that "Analytics are the present and future of professional sports. Any team that does not apply them to the fullest is at a competitive disadvantage." (Steinberg, 2015) The use of analytics by these major sports organizations, has also increased the popularity of the league to the fans! It has allowed to achieve a more hands-on feel to running an organization.

Related Work

As the world of data science has transcended to sports, many organizations have sought out new ideas that can help take their team to the next level. There has been a lot of NFL related projects, such as predicting run play success (2020 Big Data Bowl), pass play success or even concussion related projects. There's been many variations of "predicting the opponents next play", but a lot of the models I've seen have had poor accuracy scores; hovering around 30%. My job and project is to significantly improve on these accuracies and produce a score around 70-80%.

Exploratory Data Analysis

Before starting the construction of any machine learning model, it is wise to perform EDA (Exploratory Data Analysis). This will provide an insight into the data at hand, allowing the designer to tailor his/her model to make even better predictions.

Part 1: The first bit of EDA was used to analyze league-wide statistics. Presently, the NFL is now a more passing oriented league, much different from the old-school "Ground-N-Pound" run game of the past. I looked at the Pass Attempts versus Rush Attempts per quarter [Graph 1].

Part 2: I then looked at the Pass Attempts versus Rush Attempts per down [Graph 2].

Part 3: As I am an avid Washington Redskins, I thought it would be cool to view statistics for their division rivals. The Redskins' division is the NFC East and the other teams in it are the Philadelphia Eagles, New York Giants & Dallas Cowboys. I looked at their Passing Attempts versus Rush Attempts per quarter and down. [Graph 3, Graph 4, Graph 5]

Part 4: To further explore play types, I created a function that combines the play type and the location of the play to form a more detailed description of the play. This was then used to create a bar chart representation of the plays ran by the Redskins' NFC East opponents. [Graph 6, Graph 7, Graph 8]

Model Construction

To begin my model construction, I had to carefully choose my features. Now, a lot of the columns in this data where not very useful and could potentially lower the performance of any model drastically. Out of 102 columns, I selected 12 that I believed would provide valuable information to help our model make predictions for the next play. I then filtered out records that I would not be using to predict a run or pass. For example, this included removing rows where the Play type was a kickoff, punt, no play, field goal, etc. I added a new column called "ScoreDiff" which would replace two other columns: "PosTeamScore" & "DefTeamScore". (Used to signify the scores of the current Offense and Defense teams). Then I proceeded to fill out missing values with 0 for the model to work. Possibly in future attempts, I could fill out these values with a quarterly average value instead. Finally, I separated the datasets into two; my features table and my target values (PlayType). After preparing the data for the models, I decided on 4 algorithms that I would be comparing accuracies and area under the curves. I

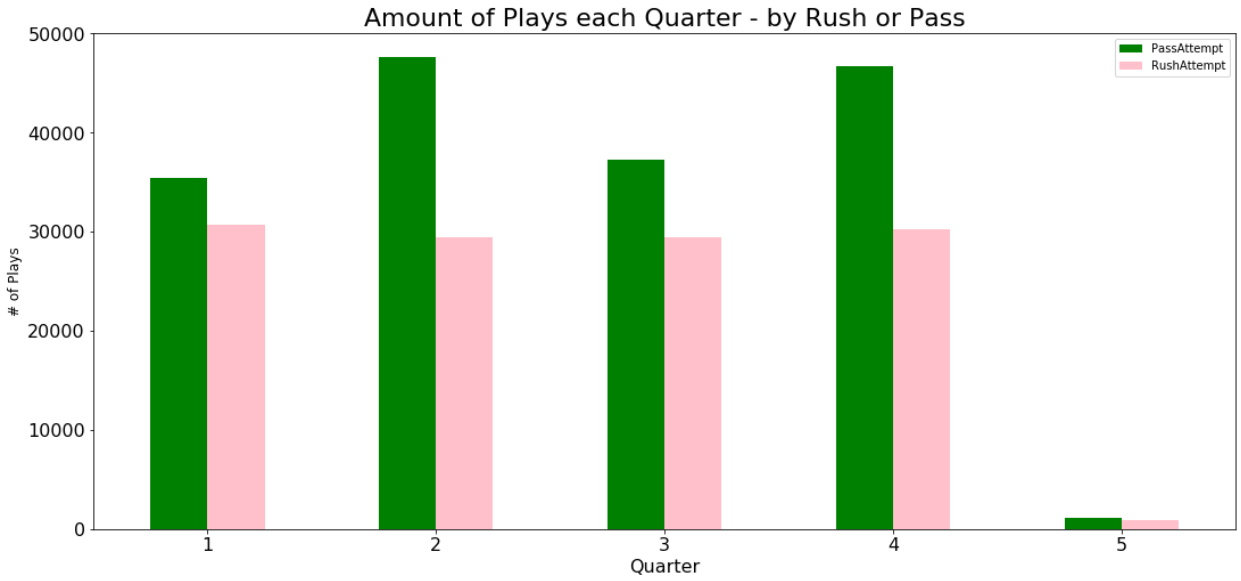
choose Decision Tree Classifier, Random Forest Classifier, Logistic Regression and K-Nearest Neighbors Classifier. (Scikit-Learn, n.d.) The results can be seen below. [Graph 9]

I then examined the importance of each of my random forest features to see which ones are playing the vital roles in making predictions. Unsurprisingly, EPA and WPA were the top two. (They are statistically calculated values already). [Graph 10] After this, I tried a Gradient Boosting Classifier in hopes of improving upon the 4 previous algorithms. Using inspiration from a function used by another programmer, I tested different parameters on the Gradient Boosting Classifier to see their effect on the accuracy. (Donelan, 2018) [Graph 11, Graph 12]

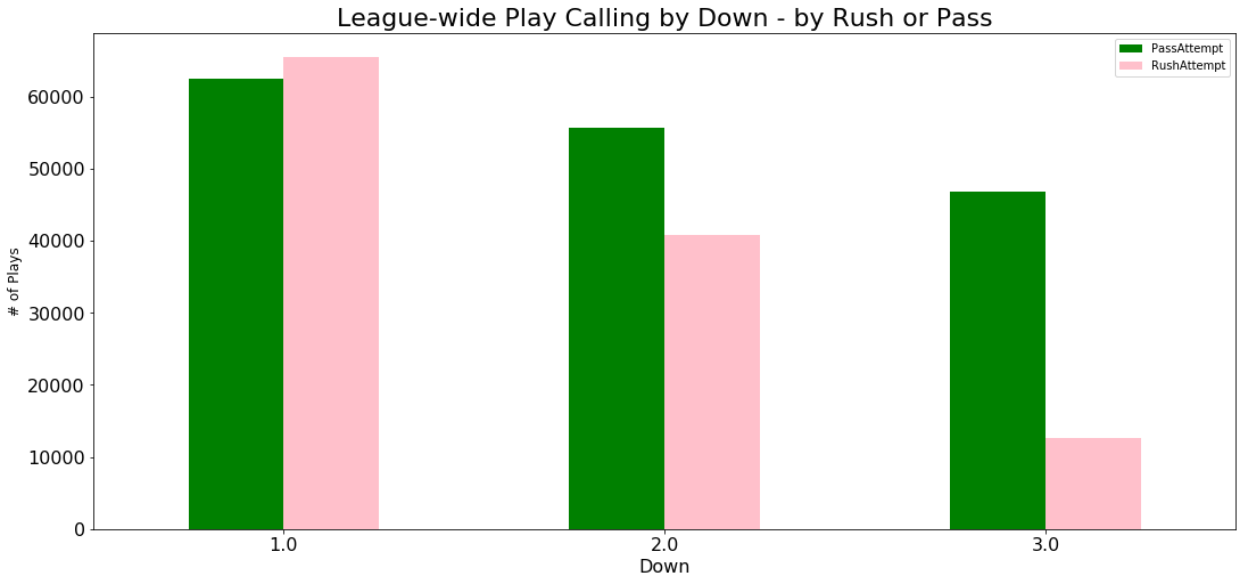
Results/Graphs

Here are all the graphs and results associated with this project.

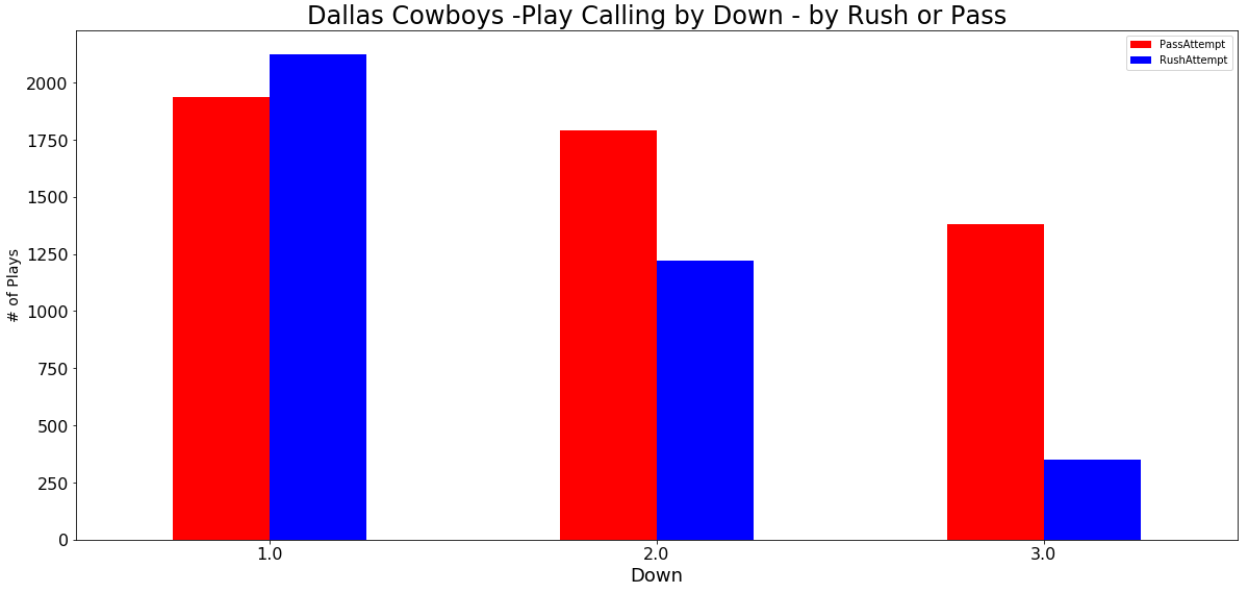
Graph 1



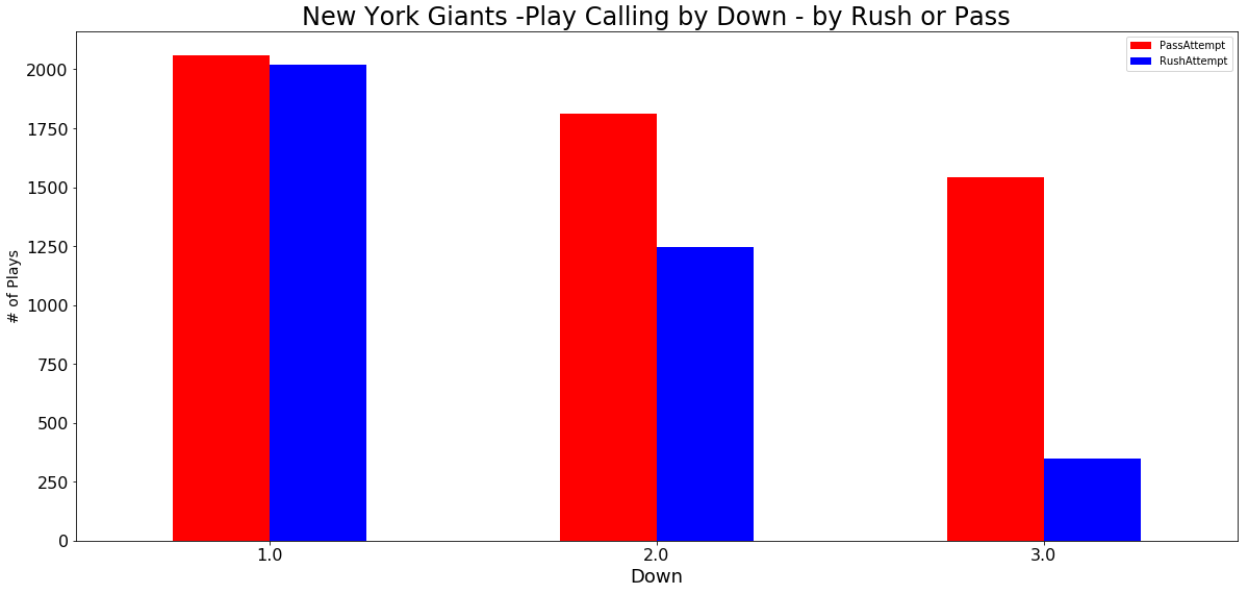
Graph 2



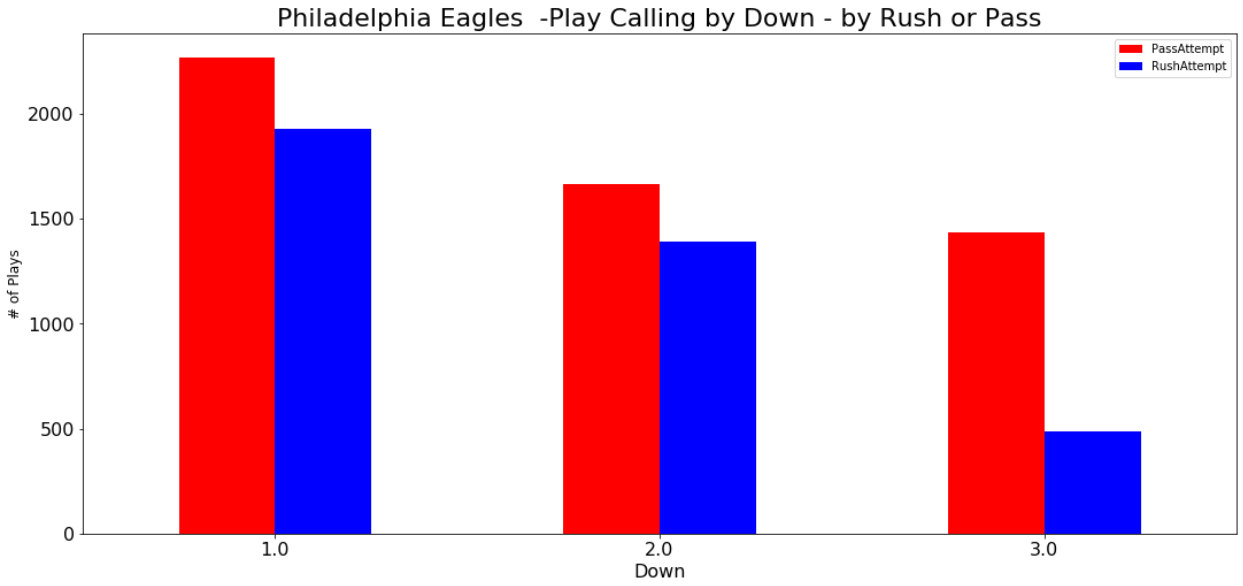
Graph 3



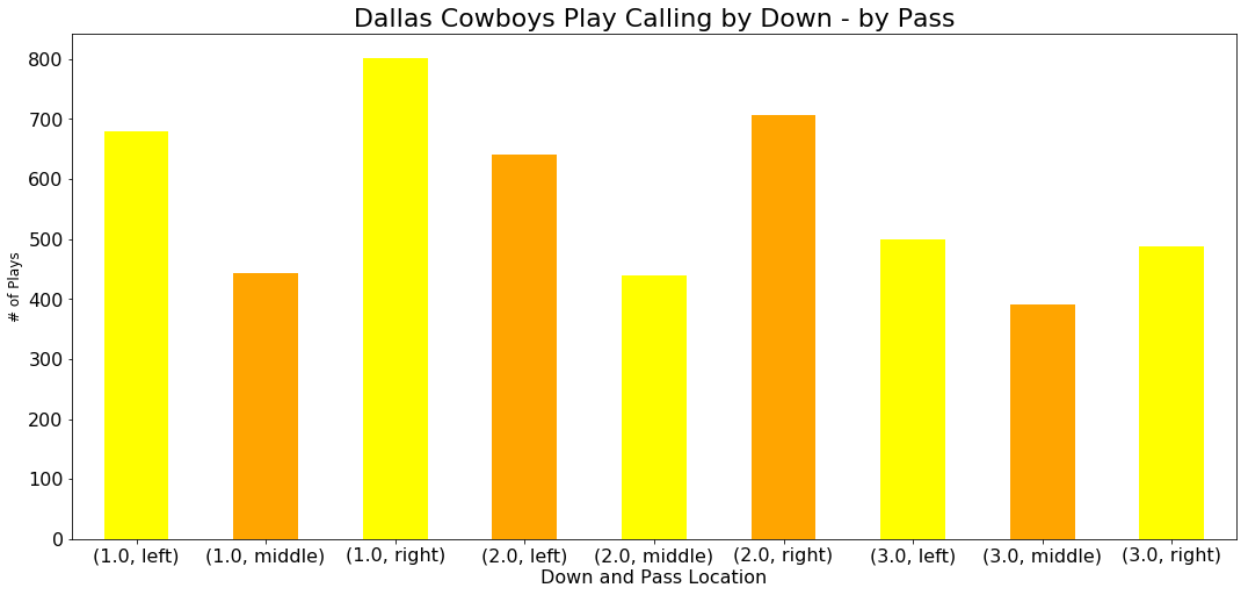
Graph 4



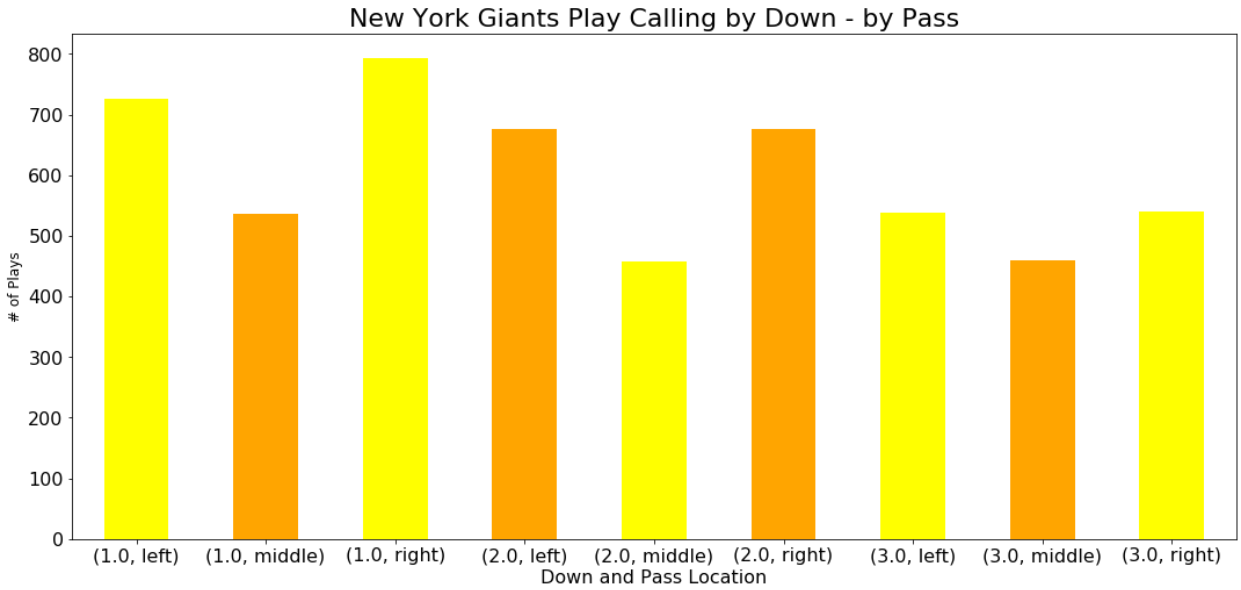
Graph 5



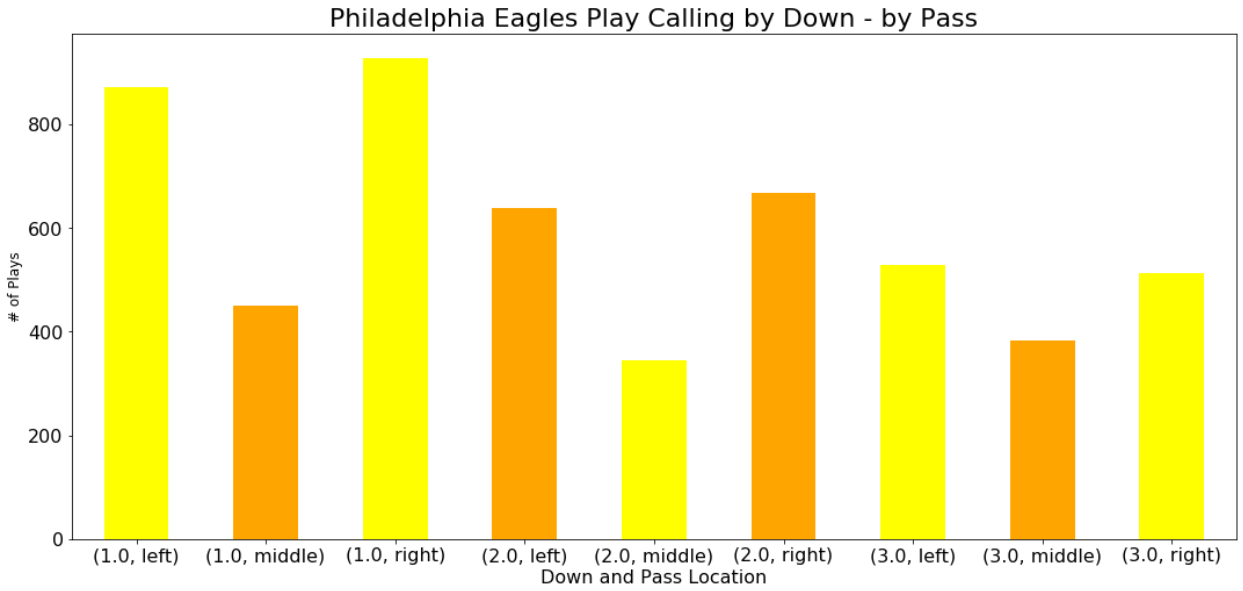
Graph 6



Graph 7



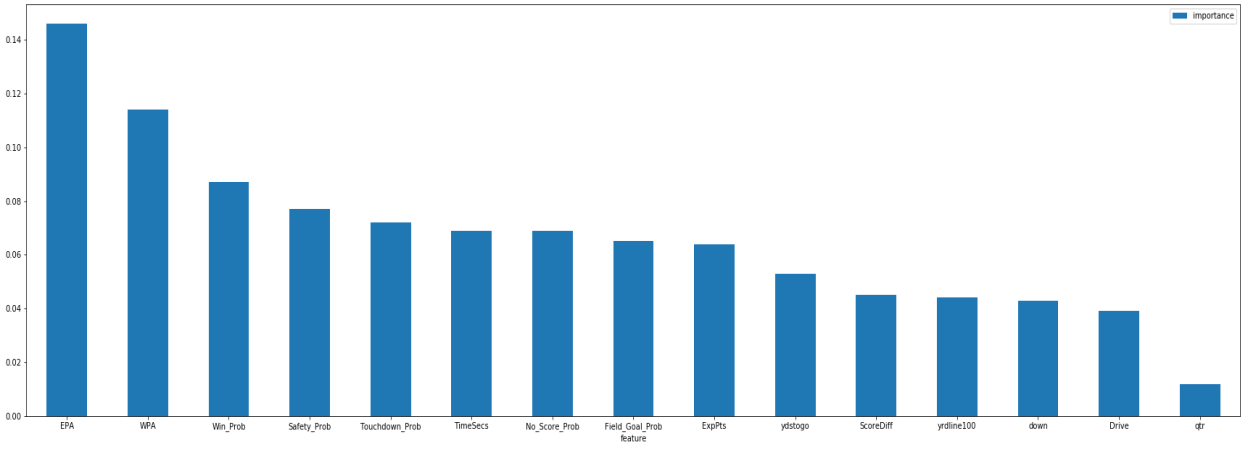
Graph 8



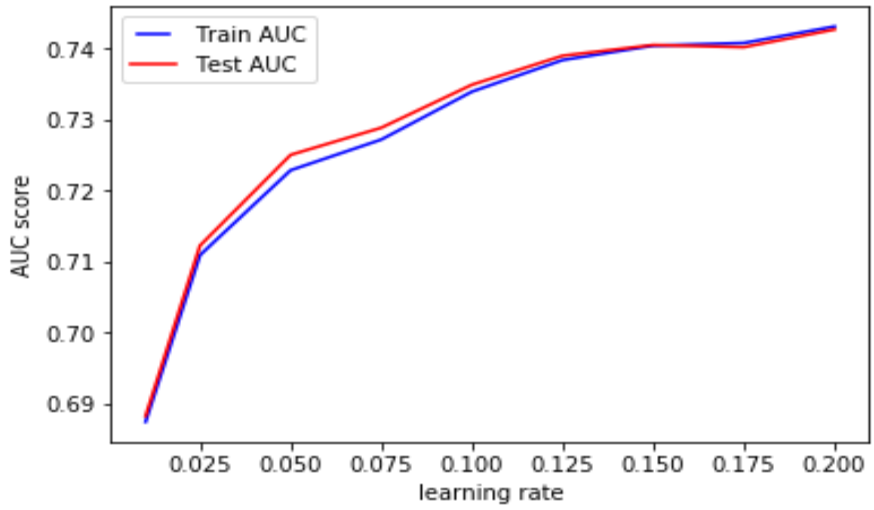
Graph 9

Model	Area-Under-Curve /Accuracy
Random Forest	0.751466
Gradient Boosting	0.734932
Decision Tree	0.702152
Logistic Regression	0.639241
KNN	0.583055

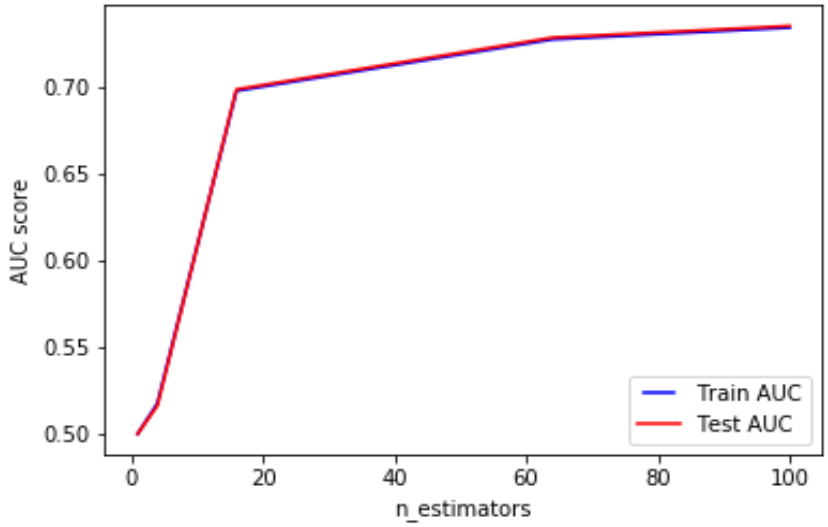
Graph 10



Graph 11



Graph 12



Discussion

In all this was a satisfactory and interesting start to a machine learning project. The dataset itself was tough to navigate through. It was so large that even before loading into my Jupyter Notebook, I had to manually go through the csv document and delete columns I was certain of not using in the model. The first lesson of note for future projects, will be to find datasets that have better usability. The models, aside from K-Nearest Neighbors performed well. While they can be improved, I believe 75%-85% is the optimum range for a prediction model. This is because it doesn't have overfitting or underfitting.

Further Work

There are many different lanes that one can go in an effort to make analytic-backed football decisions. Predicting the next play is the obvious one. Let's see some of the other aspects machine learning can provide in football analytics. We can do things such as:

- Predict field goal probability: This will help in high pressure situations where a field goal is the difference between a win or loss!
- Predict catch probability: This will depend on the quarterback, potential receiver and a variety of other factors. Even the weather will play a role in this predictor.

Works Cited

Donelan, B. (2018). *Kaggle*. Retrieved from NFL Analysis: Predicting Play Type: https://www.kaggle.com/gnarlyinsights/nfl-analysis-predicting-play-type

Rookie Road. (n.d.). *Rookie Road*. Retrieved from Defensive Coordinator: https://www.rookieroad.com/football/team-staff/defensive-coordinator/

Scikit-Learn. (n.d.). *Scikit-Learn*. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

Steinberg, L. (2015, August 18). *Forbes*. Retrieved from CHANGING THE GAME: The Rise of Sports Analytics: https://www.forbes.com/sites/leighsteinberg/2015/08/18/changing-the-game-the-rise-of-sports-analytics/#45668d5b4c1f

Williams Anosike