Anticipating Your Opponent's Next Move

Colleen Kenney

Abstract – In this paper, I used NFL play-by-play data to build a predictive model for play type. The model considers run and pass plays. After collecting, cleaning, exploring, and engineering the data, I trained 6 different machine learning models. I chose to optimize for accuracy since the cost of false positives and false negatives are similar in my mind. XGBoost resulted in the highest accuracy – 70.6%. The next step was evaluating the model's performance in certain scenarios to determine what types of circumstances were the hardest to predict.

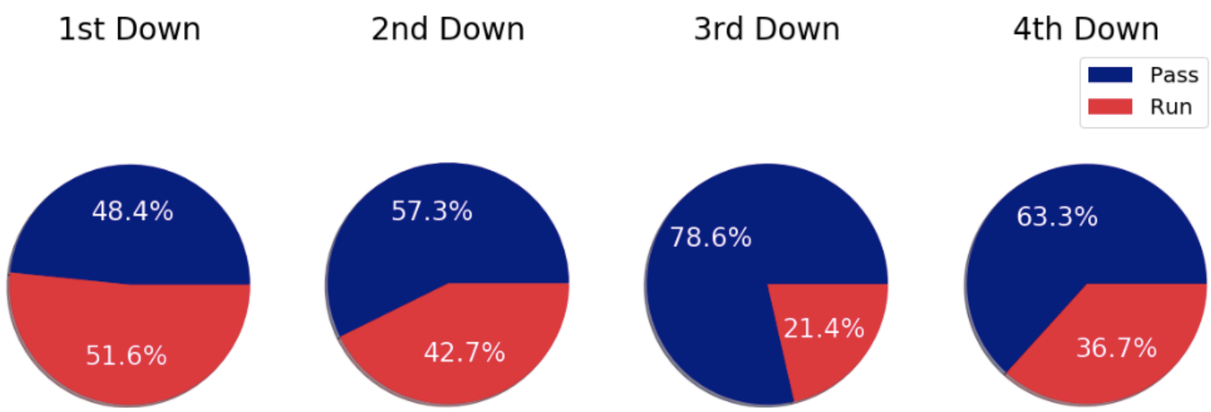
I. Introduction

NFL teams spend hours and hours each week reviewing film in order to understand their opponent's tendencies. This project takes a data-driven approach to understanding team tendencies. Over the past 10 seasons NFL teams have called pass plays 57.8% of the time versus run 42.2% of the time. On any given play, a Defensive Coordinator can guess pass and get it right 57.8% of the time. The goal of this project is to use play-by-play data from 2009 to 2018 to predict play type and achieve accuracy higher than 57.8%.

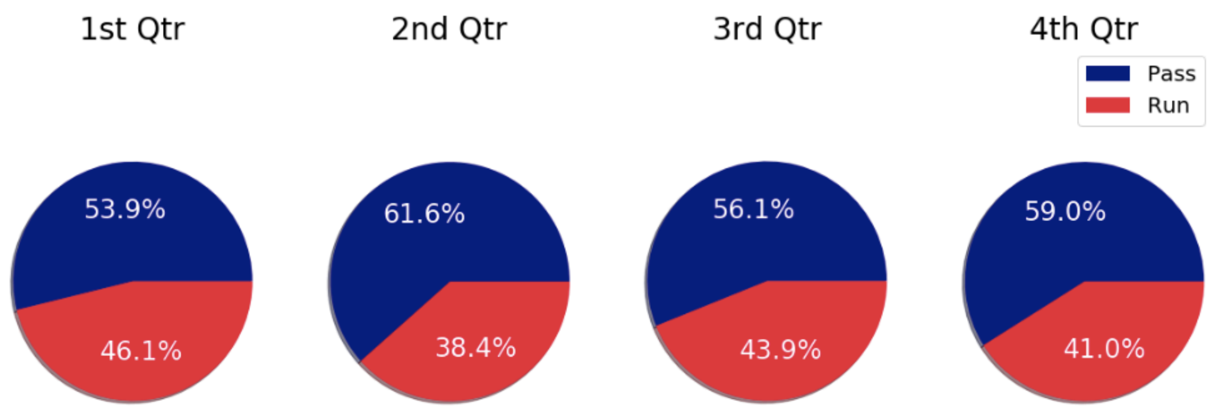
The bulk of the data used was downloaded from Max Horowitz's Kaggle dataset. Additional weather data was scraped from NFLWeather.com, and team specific data was provided by Github user cnizzardini.

II. Exploratory Data Analysis

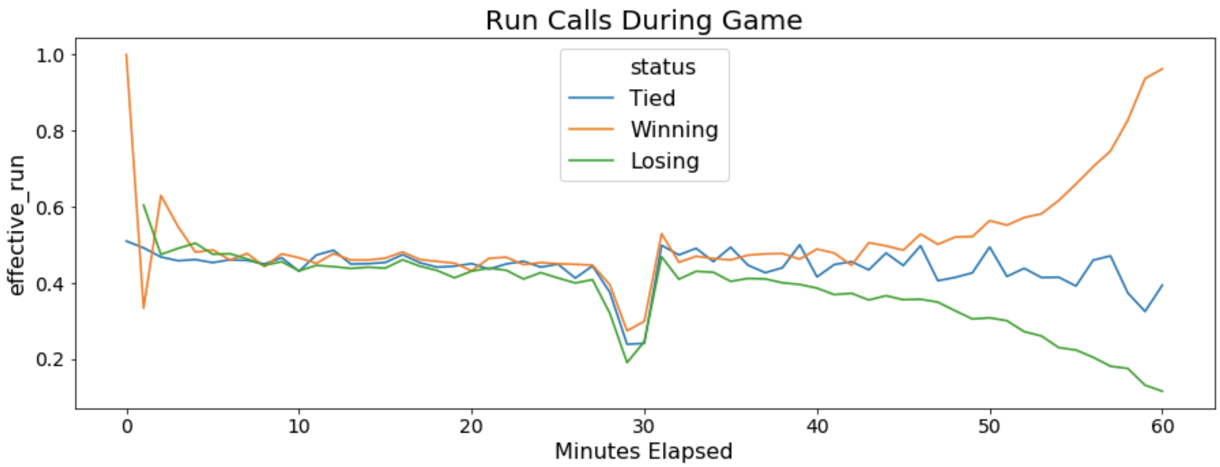
Before training models on the data, I wanted to identify patterns in the data that will be relevant when predicting play type. The first relationship to explore is how play type differs by down. The figure below shows passing is more popular as downs progress. Third down results in



the highest % of pass plays with 78.6%. The second relationship to evaluate is how play call differs by quarter. The figure below shows interestingly that the second quarter actually has had



the highest % pass plays. The first quarter seems to be the most conservative quarter with runs on 46.1% of plays. The last big relationship to understand is how play calling changes over the game given the score differential. The figure below shows how run calls increase for winning



teams in the last 10 minutes and decrease for tied and losing teams. All of these relationships make logical sense, but they will be important features when constructing our model.

III. Feature Selection & Feature Engineering

The model included 101 features, but 68[1] of those are binary indicators for who is on offense and who is on defense for a given play. The features that I used straight from the dataset include: week, yardline out of 100, quarter seconds remaining, half seconds remaining, game seconds remaining, drive, quarter, down, goal to go, yards to go, possession team timeouts, defensive team timeouts, score differential, posteam home, and win probability.

The rest of the features were engineered using existing data and weather data. Twelve features are interaction terms that I created. These features are binary for down and distance. If a play is first and 10, the 1_Long variable will equal one. The next set of engineered features were the run tendency and pass tendency. These were calculated as the % of runs/pass for the past 5 plays to try to capture what the team has been calling in that game. The last set of engineered features used weather data. Dome is a binary indicator for whether or not a game was played in a dome. Temperature range was divided into 45°F or lower (cold), between 46°F and 70°F

[1] 68 because SD moved to LAC, STL moved to LA, and JAC now goes by JAX, and we dropped the first team (ARI)

(moderate), and above 70°F. Games played in domes were assigned moderate. The last weather-related feature is a binary indicator for precipitation.

IV. Modeling

After exploring the data and creating select features, I trained these model types: Random Forest Classifier, Extra Trees Classifier, Logistic Regression, Gaussian Naïve Bayes, Extreme

Model	Test Score	Train Score
XGBoost	0.696494	0.701464
Neural Network	0.694324	0.712636
<class 'sklearn.ensemble.forest.RandomForestCl...	0.676250	0.984213
<class 'sklearn.ensemble.forest.ExtraTreesClas...	0.669215	0.999992
<class 'sklearn.linear_model.logistic.Logistic...	0.659412	0.664003
<class 'sklearn.naive_bayes.GaussianNB'>	0.625447	0.625419

Gradient Boosted Trees Classifier, and a simple neural network. The accuracy results are below. Although the Neural Network and XGBoost models have similar test scores, I chose to moved forward with the XGBoost model because it was slightly less overfit and easier to interpret. After optimizing the hyperparameters for XGBoost, my model was able to achieve 70.6%. Medium has a great explanation for the mathematics behind XGBoost.

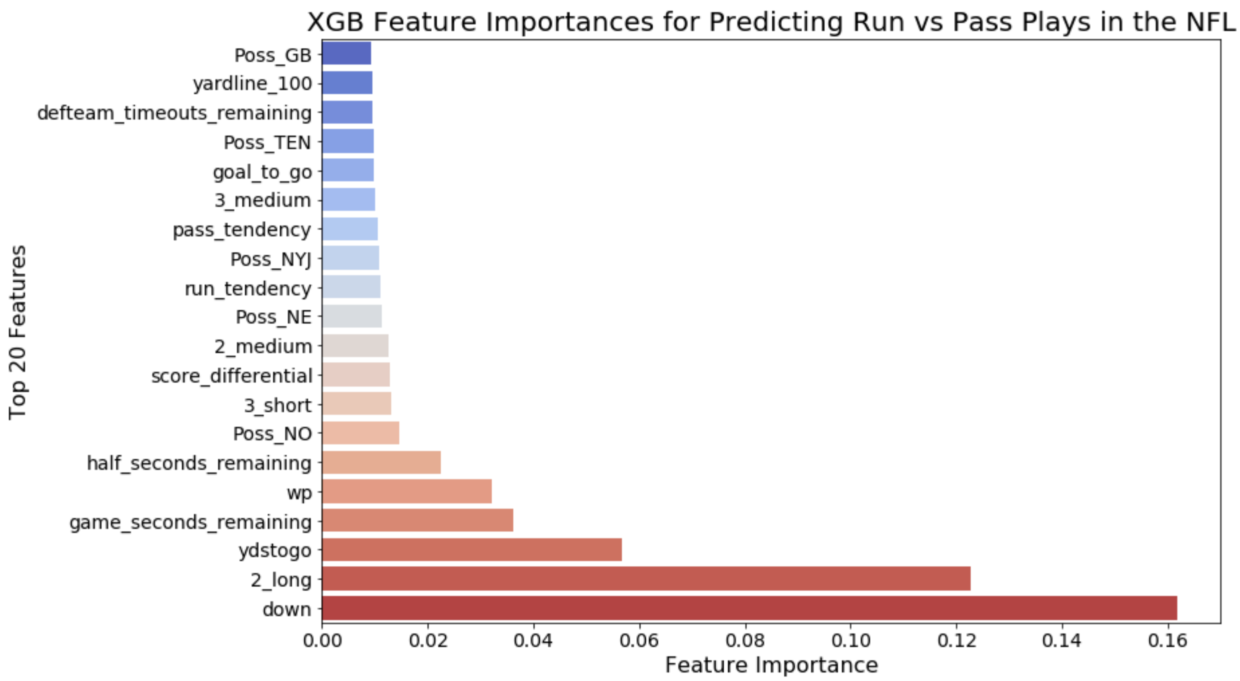
V. Interpreting Results

The model generated a confusion matrix that can be seen below. The specificity of the model

	Pred Pass	Pred Run
True Pass	33760	12630
True Run	10923	22860

(ability to predict the negative case) is 72.77%, while the sensitivity of the model (ability to predict positive case is 67.67%. Therefore, the model was better at predicting pass (negative) than it was at predicting run (positive).

Below you can see the most important features for predicting pass or run. Down was very important, but yards to go, time, and score differential were also important features.



The next step is to evaluate the performance of the model in a given scenario. Let's first look at down. The model had the highest accuracy when predicting plays that were third down. However, third down also had the highest baseline score. First down was surprisingly the best marginal improvement in accuracy (excluding 4th down).

DOWN	MODEL ACCURACY	BASELINE ACCURACY
1	64.84%	51.61%
2	68.55%	57.31%
3	85.61%	78.56%
4	81.41%	63.29%

To get more granular, we can look at the baseline accuracy scores for various down and distance scenarios. The chart on the next page shows that the model improves upon the basline accuracy in every case except for fourth and medium which has only resulted in a pass or run 251 times in the past 10 years. The most exciting result to me is predicting third and short correctly 75% of the time versus a baseline accuracy of 55%.

		correct		baseline
		mean	count	
down	distance			
1.0	Long	0.646693	33928	0.510238
	Medium	0.661692	1005	0.588264
	Short	0.714080	696	0.695621
2.0	Long	0.700048	14479	0.648864
	Medium	0.633553	8184	0.537981
	Short	0.737597	4112	0.630200
3.0	Long	0.892279	6981	0.863919
	Medium	0.895158	5246	0.878845
	Short	0.751375	4364	0.551924
4.0	Long	0.918288	257	0.886386
	Medium	0.860558	251	0.862450
	Short	0.756716	670	0.548088

The next scenario to consider is the quarter. The chart below shows that the model is the most accurate in the fourth quarter despite the second quarter having the highest baseline.

	correct	baseline
qtr		
1.0	0.644448	0.538603
2.0	0.703489	0.615784
3.0	0.676504	0.561167
4.0	0.785227	0.589670

The final scenario to consider is who is possessing the ball. The LA Chargers were the easiest to predict, while the Kansas City Chiefs where the hardest the predict.

	correct
posteam	
LAC	0.742489
KC	0.670532

VI. Conclusions & Next Steps

The final model was able to achieve 70.6% accuracy versus a baseline of 57.8%. Overall, this is a big improvement, but there is probably room for improvement. Next steps include collecting new features such as a team's tendencies over the season, what players are on the field and perhaps predicting the direction of the play (left, right, middle).