Predicting Football Plays

Ghizlaine Bennani, Matthew Leach and Tate Campbell

San Francisco, CA, United States

Abstract

All American Football plays can be categorized into passing plays and running plays. Using data from the 2002-2012 seasons we attempted to build classifiers which would be able to categorize these plays. Using over 359,000 plays in our data set we used features such as down, amount of time left in the game and current score in our models.

Using cross validation we tested various tree based and regression models. We achieved a best test accuracy of 67.89% using an Adaboost model, significantly better than random chance.

1. Introduction

Since the early 2000s the use of data analytics within sports has been growing at a fast pace. Arguably the revolution was started in baseball, documented in the book and film Moneyball. However in recent years this has spread over to all other sports, including football.

Football is perhaps a harder sport to analyze than baseball due to all the interlocking parts.[1] It can be readily observed that moving the ball down the field is the heart of nearly all football strategy. In fact, teams devise entire playbooks in which they scheme seemingly endless ways to move the ball down the field against all types of defensive formations. Despite the multitude of unique football plays and their apparent complexity, there is a fundamental dichotomy between football plays and their design, namely every football play is either designed as a passing play or a rushing play. Over the years various machine learning methods have been used to predict the type of a football play given the game conditions at that time. Strange & Sharmir [1] used classification algorithms to predict whether a football play would be a pass or a run. They obtained an average accuracy (by team) of 59-60% using Support Vector Machine and Weighted Nearest Distance techniques.

We therefore set out to see if we could replicate and potentially improve the prediction methods used.

[1]The game is played on a grass field that is 100 yards in length, with additional 10 yard regions on both sides of the field representing "the end zones" (which are clearly marked from the rest of the field, and contain the quintessential yellow goal posts). Teams are made up of 11 men and teams take turns on offense and defense, each offense attempting to take the ball into the end zone and each defense attempting to stop the opposing offense from doing so. When the offensive team starts with the ball (typically on the 20-yard line) that team has four "downs" (opportunities to run plays) to advance the ball a total of 10 yards. If the offense is able to do this, they receive a new set of four downs, and if not, they must surrender the ball to the other team. Teams receive points by either taking the ball to the opposite end zone (which is called a touchdown - worth 6 points) or by kicking the ball through the opposite goal posts from the field (called a field goal - worth 3 points).

2. Data Cleaning

We used data from source www.advancedfootballanalytics.com containing data from the 2002-2012 NFL seasons in csv format.

The advanced football analytics play-by-play data contains variables game_id, quarter, minute & second, offensive and defensive team, down, distance to a first down, yard line (distance to the goal), description of the play, offensive and defensive scores. Note in particular that we do not have what type of play it was which is what we want to predict. Instead the description contains text from which we attempt to determine whether the play was a pass or run. This was the major cleaning task that had to be done with this data.

Luckily the description column contains many repeated phrases so it is not too difficult a text processing challenge. For this initial investigation we did not use a very complicated method for analyzing the text but in the future a tf-idf method could potentially be used. We simply searched each description for a few set strings that corresponded to passes or runs (e.g. 'left tackle', 'rushed', 'right end' for run plays and 'pass' for pass plays). This had advantages that there would not be any plays mis-categorized.

Once the play type was done the other variables were easy to create. We had the following variables as in the paper: time_to_half (seconds), time_left_in_game (seconds), down (between 1 and 4), dist_to_first (meters), quarter (1 to 4), score_diff (offensive score - defensive score), yard_line (meters), off_score, def_score, is_pass (boolean).

3. Exploratory Data Analysis

Using Python we created plays.csv and plays.json files containing the variables noted above. We then used Apache Drill to run some SQL queries on the raw data files as this allows easy aggregation to get some initial insights.

Firstly the number of plays we have is 359,101 of which 201,096 are passes. This is 56% which corresponds fairly well with NFL statistics on the pass rate [2] where team pass rates range from 49% to 65%.

Using aggregation queries we found that the pass rate by downs is as outlined in Table 1.

Down	Pass Rate
1	0.47
2	0.55
3	0.76
4	0.61

Table 1: Pass rates by down

We see that the pass rate is significantly higher in the third down than any other and much lower in the first. This makes intuitive sense, as teams can be more flexible in the early downs but must take more risks in the 3^{rd} down in order to convert the play into a first down. We also run a query to group by the score difference. The results of this query are plotted in Figure A.3.

Here we see the pass rate drops as the score difference increases. This is again unsurprising since as a team gets a larger lead they are more likely to try and protect the lead. It suggests in our models the score difference ought to be fairly significant in predicting the pass rate.

Aside from factors like the current down and score difference, another likely component of pass rate if the team in possession of the ball. In line with passing vs. rushing dichotomy

within football plays, there are (in general) two playing styles that teams focus on when planning their plays. Some teams are known for taking an aggressive style to the the game by taking lots of chances and passing the ball a lot in the hopes of scoring more. Other teams are defensive strongholds who prefer not taking excessive risks with the ball and in turn feature a run-heavy offense. To that end, pass rates were aggregated by team and are plotted in Figure A.4.

In order to further illuminate the relationships between down and pass rate, we plot a heat map to compare pass rates across a grid of down and dist_to_first values, which can be seen in Figure 1. The bottom half portion of this heat map isn't particularly interesting, as it's widely known that in these situations (3^{rd} & long or 4^{th} & long, in football parlance) the optimal and default strategy is to pass. However in the case of the 2^{nd} down, we get a nice linear gradient, suggesting that the pass rate can be accurately predicted based on dist_to_first when the play is on the 2^{nd} down. This is certainly of interest to us, and in future work we are considering training multiple models across downs, teams, quarters, or some combination thereof.

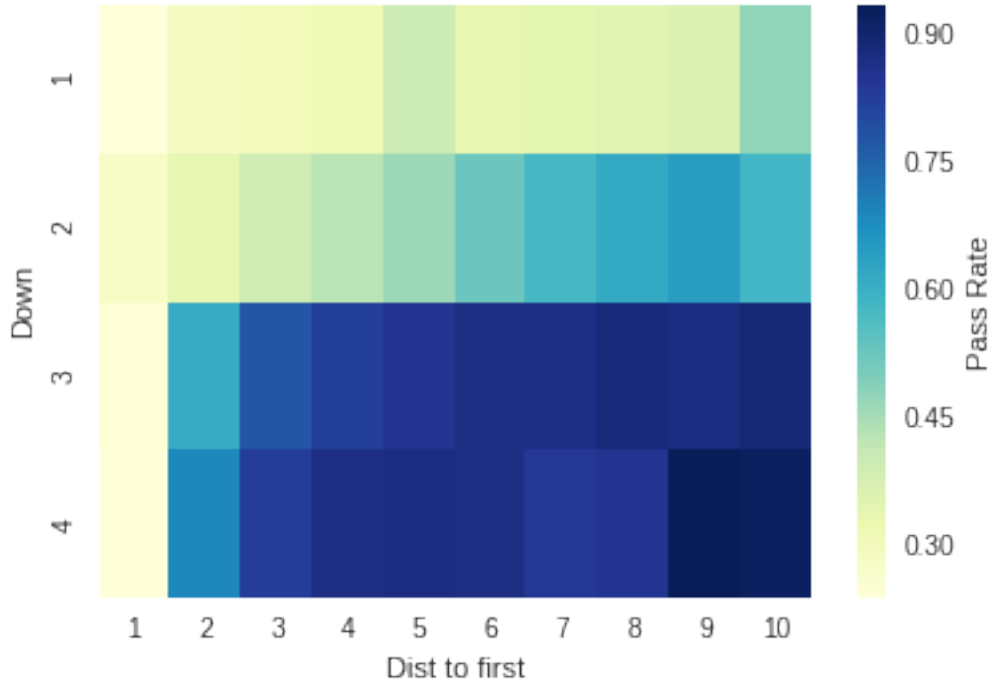

Figure 1: Pass Rates by Down and distance to go

4. Classification Models

Eight machine learning classification algorithms were trained using plays.csv in an 80/20 train to test ratio via cross validation. Of the classification algorithms used, three were tree based method (Decision tree, Random forest and Extra tree classifier) and the rest of the implemented classifiers covered logistic regression, linear SVM and K nearest neighbors. The optimal shrinkage parameters for both logistic regression and SVM were found through a grid search via 10 fold cross validation. To define C, we have used L2 regularization penalty to reduce the weights of less important features.

For KNN, choosing the optimal level of flexibility is critical to properly fit our model and avoid overfitting. Choosing the optimal K will depends on the bias-variance tradeoff and the resulting U-shape in the test error rate that we can observe in Figure 2.

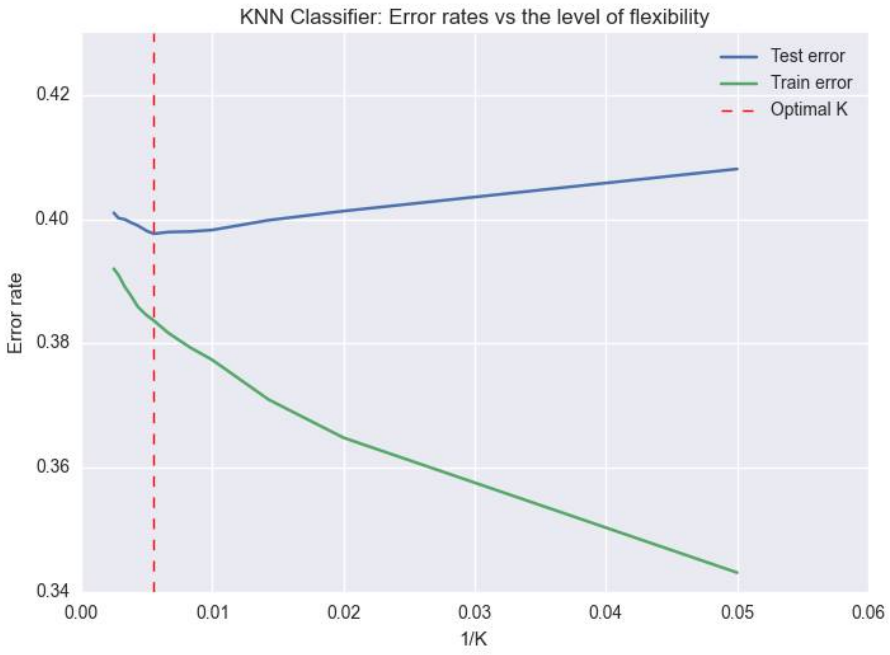

Figure 2: Choosing an optimal K avoiding overfitting

The level of flexibility is assesed via 1/K. A small value of K provides the most flexible fit, which will have low bias but high variance as in this case the classification in a given region depends on small number of observations. Larger value of K provides a smoother and less variable fit but causes a big bias. The optimal level of flexibility and thus the optimal K that gives us the best tradeoff between bias and variance is computed via 10 fold cross validation and correspond to red dashed line.

Regarding the three based method, the optimal scores are summarized for each method after bagging and pruning. In order to improve the accuracy we have used Adaboost on top of the optimal weak learner classifier that we have found (random forest). The optimal tradeoff between the number of estimators and the learning rate is defined through a grid search via 10 fold cross validation. On top of that, We have also tuned the max depth tree for a better performance for both Gradient boosting and our base learners. D, n, eta and m respectively correspond to the optimal values of the max depth of the tree, the number of estimators, the learning rate and the number of iterations. Out of all the classifiers and after tunning several parameters we have achieved the best accuracy via Adaboost with Random forest as a base learner.

The corresponding train and test scores are summarized in Table 2 and 3 . The coefficients for the regression models are listed in Table 4.

Table 2: Tree based Model Accuracies

Model	Decision Tree	Random Forest	Gradient Boosted Trees
Training accuracy (%)	91.8	69.2	68.7
Test accuracy (%)	61.32	69.1	68.7
Optimal Tuning Parameter	D = 50	n = 10, D = 50	eta = 0.08

Table 3: Regression Models Accuracies

Model	Logistic	KNN	SVM
Training accuracy (%)	64.5	61.6	48.0
Test accuracy (%)	64.9	60.2	43.1
Optimal Tunning Parameter	C = 8.9	K = 180	C = 0.8

Table 4: Regression Model Weights

Variable Name	intercept β_0	quarter β_1	time_left_in_game β_2	down β_3	dist_to_first β_4	yard_line β_5	score_diff β_6
Logistic weight	-0.0918	-0.406	-0.0004	0.6867	0.1079	0.0010	-0.0228
SVM weight	0.0	-2.2342	-0.2473	3.0421	0.5	-0.0998	-0.3181

5. Conclusions and Further Work

Overall we found that predicting football plays is something that can be done at better than random chance with a best model success of 68%. Therefore NFL teams should be using analysis of this sort to help prepare their defences. One of the main challenges is creating a model that can be run quickly as results are needed in real time. This project has required us to complete all stages in a data science project from data collection to model validation.

In addition we have created a Flask application which allows anyone to use the logistic regression classifier to predict what the play will be. This is located at `http://52.35.109.13:5000/toolbox` and an example screenshot is at A.5.

[1] Prediction of American Football Plays Using Pattern Recognition, Robert Strange & Lior Shamir

[2] https://www.teamrankings.com/nfl/stat/passing-play-pct

Appendix A. Figures

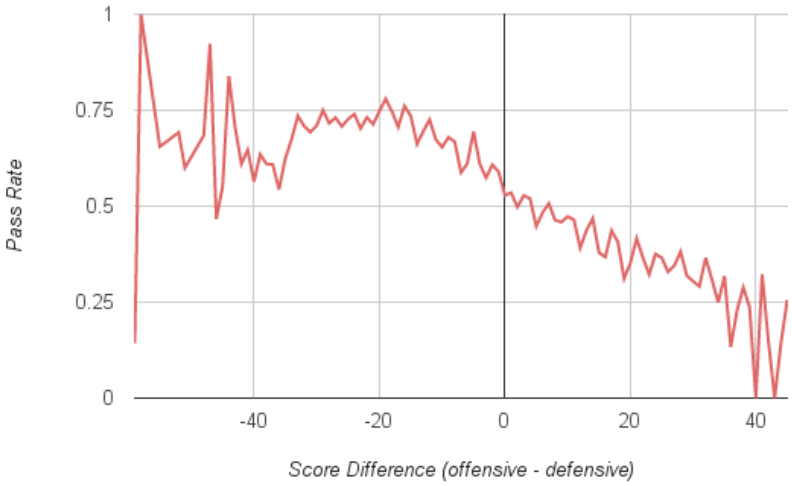

Figure A.3: Pass Rate by Score Difference

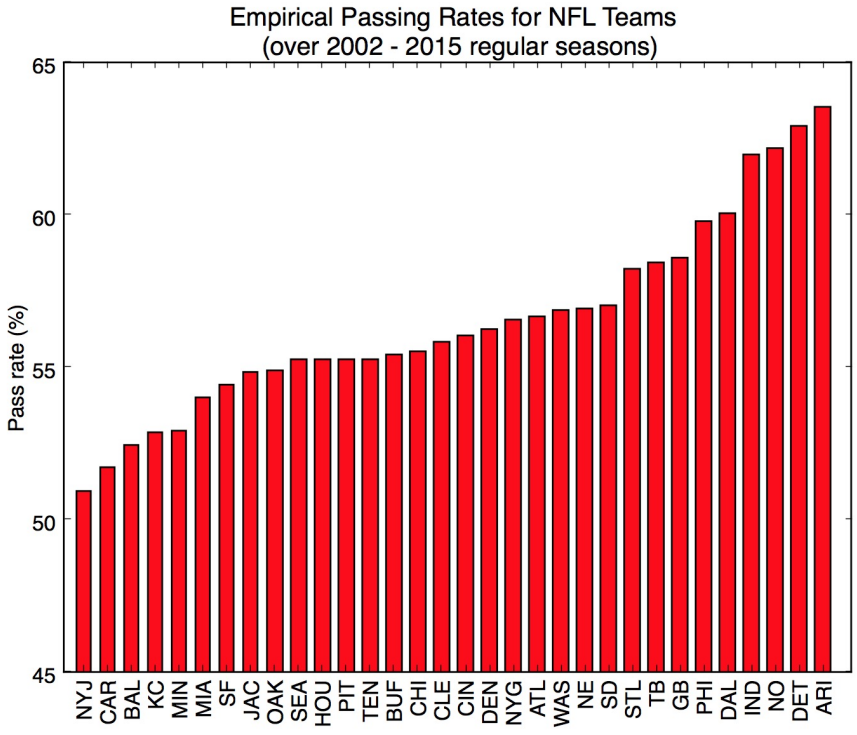

Figure A.4: Pass Rates by Team

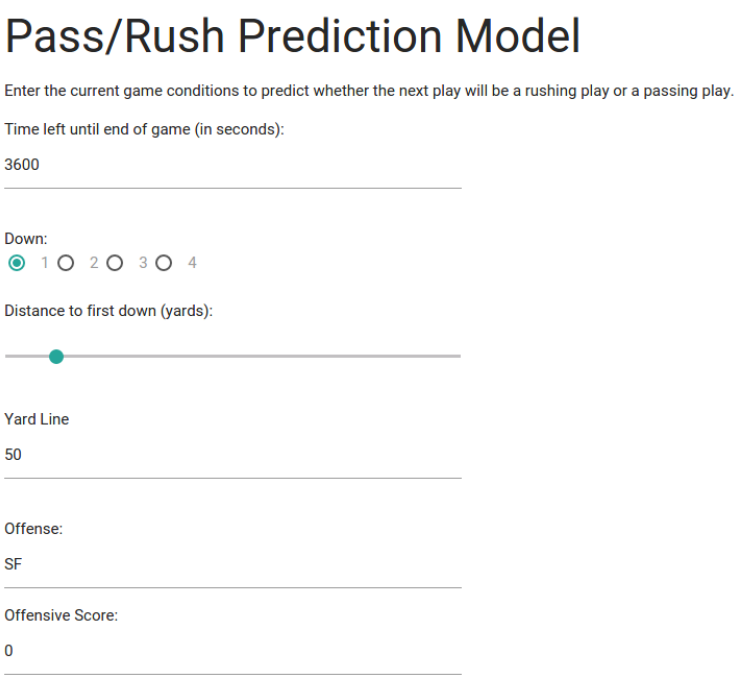

Figure A.5: Screenshot of Flask app