NFL Play Predictor - In-depth Analysis

Viraj P. Modak

We started with a problem statement - "can we predict the type of play (pass/run) in NFL games based on the game situation?" We represented the raw play-by-play data as features and response variable. Exploratory data analysis helped us identify some key trends and the statistical analysis helped us estimate their statistical significance. In this report, we present the various predictive models which were trained using our data as well as their performance.

Ours is a binary classification problem. It can be turned into a multi-class classification exercise by adding plays like short/long/right/left pass, QB scramble and right/left run and so on. However, it would be appropriate to build such an enhanced model only ***after*** establishing a proof-of-concept for a much simpler pass/play predictor, which essentially is this project.

Approach:

To train our data, we tried the following Machine Learning (ML) Techniques*:

1. K-Nearest Neighbors (KNN)
2. Logistic Regression (LogReg)
3. Decision Trees (DT)
4. Random Forests (RF)

**SVMs were tried but ultimately not included in the report as they were noticeably slower compared to all other models and did not perform better than any of the techniques*

These are standard ML techniques and no further explanation is provided on how the techniques work. With each technique, we first built a training model using our "as is" features. We calculate standard performance metrics such as net accuracy score, precision, recall, confusion matrix and the ROC curve. After using our "as is" features we tried to attempt feature engineering to improve model performance. Toward the end we further analyze the performance of the best predictive model.

Preliminary model training:

Data is split using 80/20 Train/Test ratio. For each model, hyperparameters are tuned using a grid search over 5-fold cross validation. The final hyperparameters and the performance metrics are presented in Table 1. The confusion matrices and ROC curves are presented in Figure 1 and Figure 2 respectively. **For building the model 0/1 were used for Run/Pass plays respectively.**

Table 1: Performance parameters of preliminary model training

	KNN	LogReg	DT	RF

Best hyperparameter grid/value	n_neighbors = 39	C = 100	max_depth = 13	max_depth = 20
				min_samples_leaf = 4
				min_samples_split = 2

Training set accuracy	67%	72%	76%	82%
Test set accuracy	65%	72%	73%	75%
Avg Precision	64%	72%	73%	75%
Avg Recall	65%	72%	73%	75%

In all cases, expectedly, the training set performs better than the test set. However, it is not a big difference which means we have not over-fit or under-fit the model. KNN is the poorest perfroming model of all as it relies only on proximity in the feature space. LogReg and DT perform better because they add more complexity to the model. Logreg incorporates a linear dependence on the features and DT relies on a complex hierarchy of if/else questions to reach a final outcome. The weights of the LogReg model are reported in Table 2. We can see that shotgun formation, down and previous pass performance are more influential than others. RF performs even better because of multiple iterations over the training data which incorporates a "wisdom of the crowd" aspect in training.

Table 2: LogReg coefficient by training feature

Feature	**Coefficient**
qtr	-0.123501
down	0.554158
ydstogo	0.089554
shotgun	1.468002
no_huddle	-0.009586
posteam_timeouts_remaining	-0.152588
score_differential	-0.024096
play_clock	-0.009307
yards_to_goal	0.001771
cum_pass_comp%	0.551482
cum_pass_yards	0.002881
cum_run_yards	-0.007682
teamQB_dual_threat_rating	-0.45137

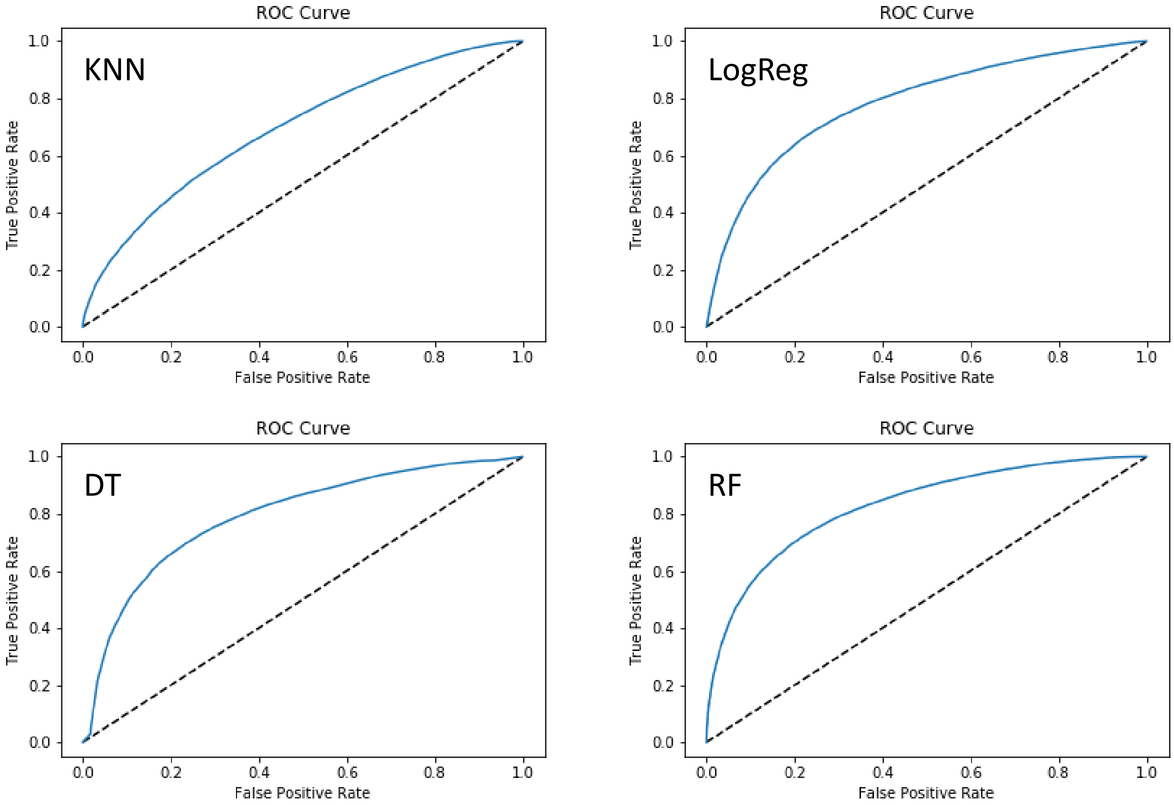

Figure 1: ROC curves for preliminary model training. Inner textboxes correspond to the particular model. KNN is the worst performer while RF is the best performer

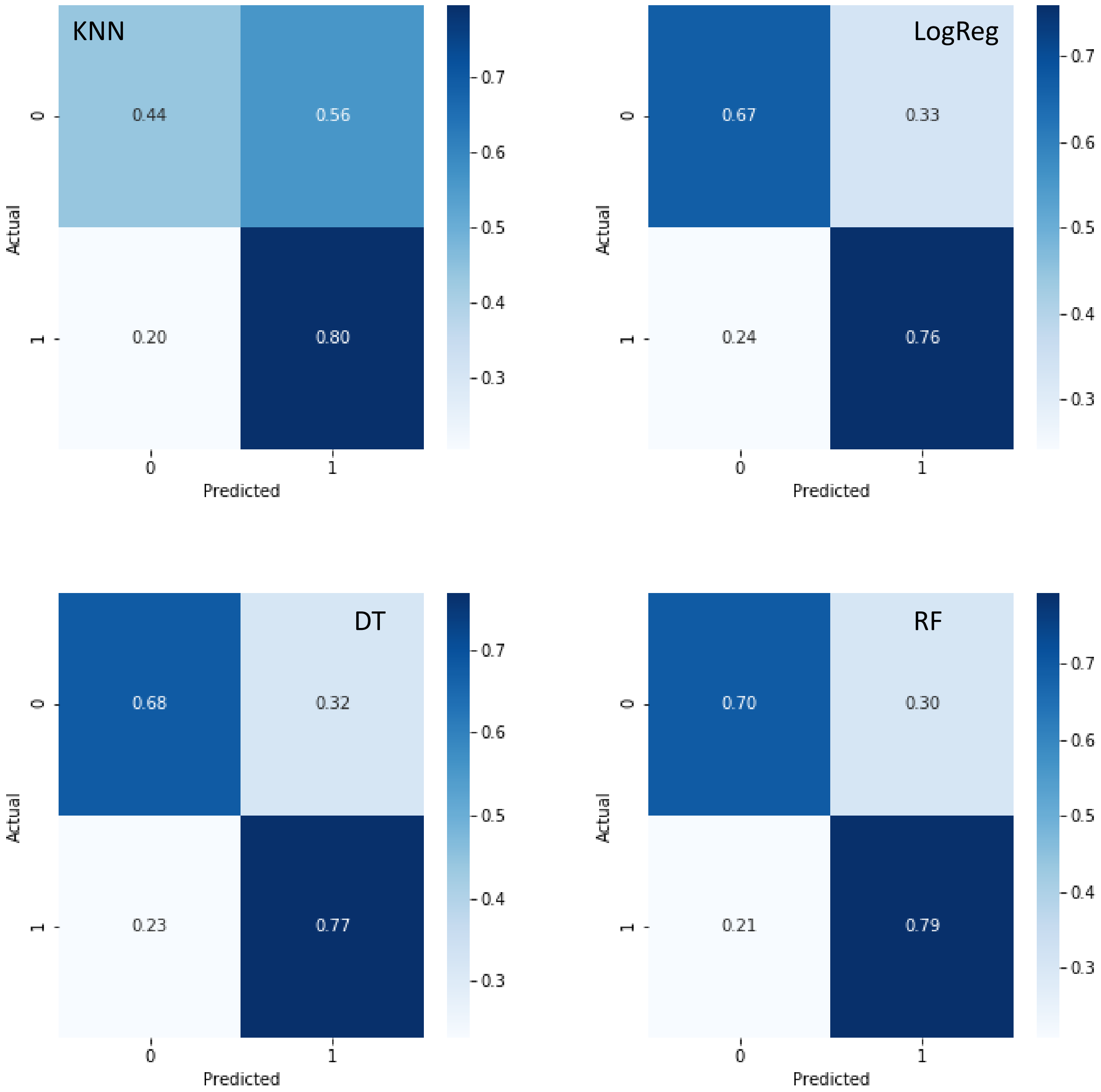

Figure 2: Confusion matrices for preliminary model training. Inner textboxes correspond to the particular model. There is a general tendency to predict Pass plays over Run plays. KNN, in fact performs the best on Pass plays but is poorest on Run plays, which impacts its average accuracy score reported in Table 1

Feature Engineering:

Feature engineering was incorporated to improve performance of the model as follows:

1. Categorical feature for game quarter was represented using "one-hot encoding". There are other categorical features including play down and time-outs remaining were not converted to one-hot encoded features as they are more fluid than game quarter
2. Previous play (pass/run) was added as a feature
3. Continuous variables such as yardage, play_clock, score_differential, previous pass/run performance were normalized

Feature Engineering led to only marginal improvement of model performance compared to using raw features. This be due to (1) existing features have enough complexity that adding more using one-hot encoding does not enhance the feature space and (2) features don't differ by orders of magnitude which leaves feature scaling redundant. Model performance parameters are reported in Table 3. Given that KNN is the worst performer among the models tested, it was excluded from this analysis. ROC curves and confusion matrices are also largely similar and not shown here to avoid redundancy.

Table 3: Performance metrics for model training post Feature Engineering

	LogReg	**DT**	**RF**

Best hyperparameter grid/value	C = 10	max_depth = 11	max_depth = 20
			min_samples_leaf = 4
			min_samples_split = 2

Training set accuracy	72%	75%	82%
Test set accuracy	72%	74%	76%

Discussion and Analysis:

From the data (confusion matrices) we can see that across all models, the general tendency is for the model to predict pass plays more - resulting in a better performance for pass plays. This may be expected because it was seen during EDA that pass plays occur more frequently overall, than run plays do. However, it would be interesting to identify the game situations where the model performs the best and where the model performs the worst. Given that Random Forests with

Feature Engineering has been the best model so far, only this model has been considered for the analysis performed in this section.

The accuracy score as a function of qtr and downs is shown in Figure 3. We can see that the plays are most predictable in qtr 4, which is in a way expected, because (1) teams are less likely to take risks with "trick-plays" toward the end of the game and (2) if the teams are not going to make a play and not punt on 4th down, they will most likely play according to the game situation. Teams are most predictable on 3rd and 4th down - highest accuracy seen so far. This can be explained in the last two of the available 4 downs, teams are likely to stick to a script and make plays exactly according to what the situation demands. This short analysis offers a key insight into where teams might want to add a bit of unpredictability in their play-calling.

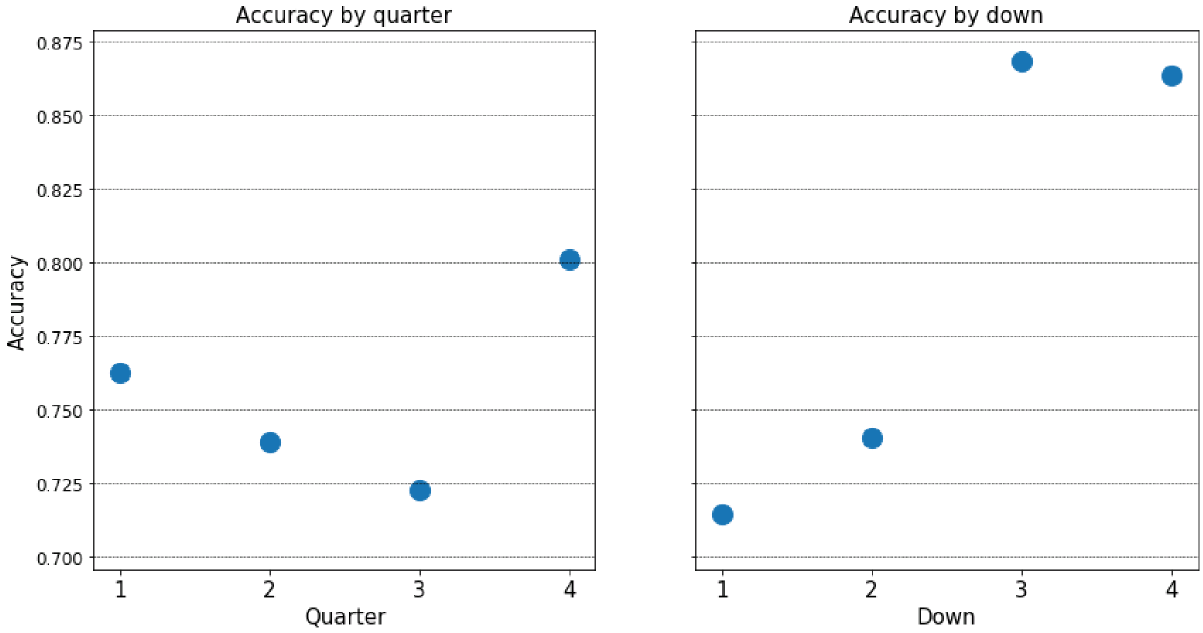

Figure 3: Accuracy score segmented out by qtr and down. Teams are most predictable during qtr 4 and on down 3 and 4

We can also test if certain teams are more predictable than other teams. Let's take the case of the latest 2018-2019 regular season. The RF model was tested for all teams only for the regular season games and the accuracy was plotted as a scatter. This information was correlated to the regular season record. However, no strong trends were seen. But going into the play-offs, the predictability metric will of course be of interest to teams. For example, this model performs better than average for NE - making them more predictable. In spite of that NE qualified for the play-offs. If the oppositions had based their strategy based on this model, would they have been able to break NE's plays? Or was NE's execution flawless - and the opposition could not break their plays even after predicting them. These questions will be important as teams devise their strategies going into the play-offs. And this model will provide keen insights as they analyze the performance of their potential opposition.

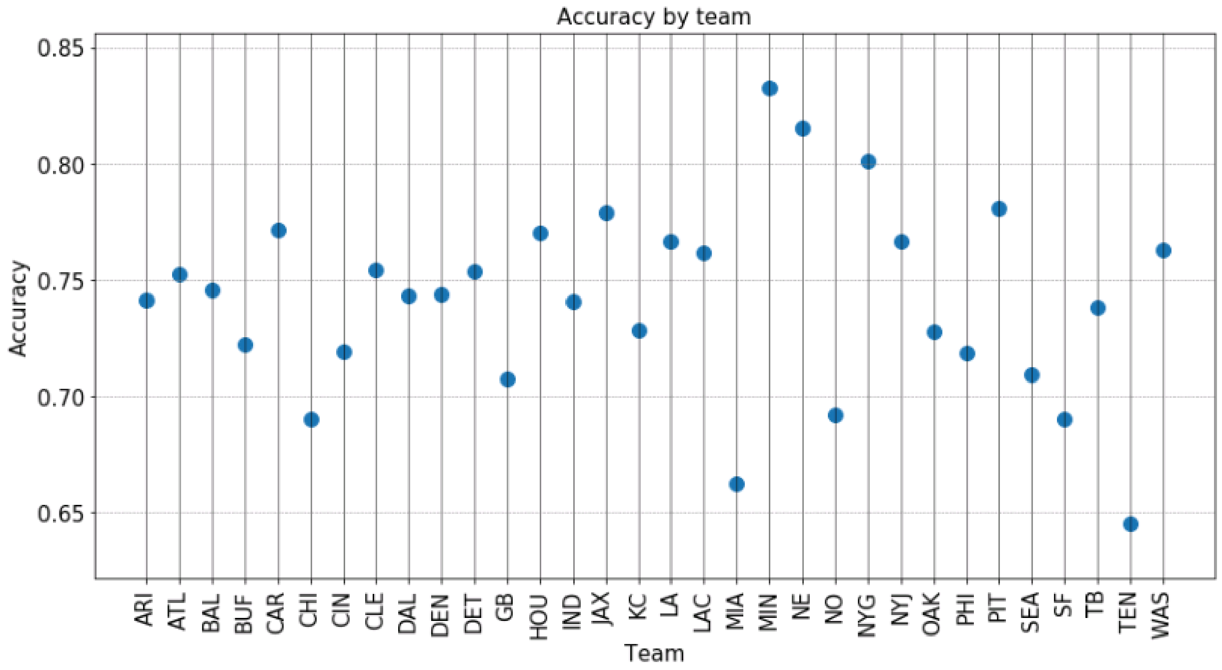

Figure 4: Accuracy over regular season games by Team