**United States Air Force Academy**

**Predicting Play Calls of the National Football League**

**Final Report**

**C1C Patrick Mackintosh, C1C Jacob Lindell**

**Econometrics 465**

**Lieutenant Colonel Martin Gaupp**

**December 4th, 2015**

Table of Contents

[Executive Summary 3](#_Toc397375186)

[Motivation 4](#_Toc397375186)

[Dataset Overview 4](#_Toc397375187)

[Data Processing 5](#_Toc397375188)

[Analysis 6](#_Toc397375189)

Results [11](#_Toc397375190)

Conclusions [13](#_Toc397375190)

[References 14](#_Toc397375190)

[Appendix 15](#_Toc397375191)

[Documentation 25](#_Toc397375191)

**Executive Summary**

One of the biggest battles in a football game is not between any of the players on the field, but between the Offensive and Defensive Coordinators. The defense is trying to run the most optimal play to stop the offense, while the offense is trying to run the most optimal play in order to make the desired yardage, and ultimately a touchdown. Although Defensive Coordinators are usually seasoned veterans that are very good at looking at the current situation and inferring what the offensive team may call, it is almost an educated guess every time the defense calls a play.

We have found a way to take the idea of play calling and enhance it from an educated guess an individual makes to a model that can predict the offensive play call with approximately 70% accuracy, which takes the guessing game out of football. Our model takes different play metrics; down, yards to go, side of field, etc. to predict what the Offensive Coordinator is going to call. In order to solve the problem, it was important to determine which factors were most influential in predicting the next play that would be called. We determined that factors such as team, down, yards to go for a first down, and both teams’ scores were the most significant variables in predicting the play call. After using a random forest model that takes in all of the important factors, we crafted one model that tells us whether or not the team will run or pass, while an additional model takes in that information and tells us exactly where a play will go; whether that be run left, right, or middle, or a pass short or a pass deep. The first model was able to take data from both 2013 and 2014, and correctly predict the next play going through the both year’s data with a 99.6% prediction average. Our second model, which was used to predict the more specific type of play, was able to predict the specific play call 45.3% of the time.

To truly determine the worth of our model, we will have to predict using games from the 2015 season and determine the accuracy of the model this year. As long as our model has the ability to correctly predict the next play called greater than 70% of the time, it could be very useful to a Defensive Coordinator who is trying to call the best defensive play to stop the offense from gaining yards, but more importantly keep them from getting a first down, and ultimately a touchdown.

**Motivation**

We think that football analytics is coming along and going to be much more prevalent in the near future, and this will be one of the ways it will be used. We have heard that Gary Kubiak, the head coach of the Denver Broncos, is trying to replace one of the coordinators on his headset with an analyst that will be running regressions and using statistics to determine the next play that should be run (Wagner-McGough). One of the very interesting analyses online is from the 2014 Green Bay Packers vs. Seattle Seahawks NFC Championship Game, in which Brian Burke breaks down multiple decisions throughout the game and statistically shows how each decision changed the outcome of the game (Burke).

The purpose of the project is to be able to input the data points of the current play; the yards to go, the down, etc. and predict what the offense is going to call. If the Defensive Coordinator was able to use analytics in order to determine what the offense is going to call, this would take all of the guessing out of play calling and get rid of any bias the Defensive Coordinator has; if the defensive team is most likely to blitz on 3rd down, the offense could then look at that and call a play that would combat that.

**Dataset Overview**

The data was built by Bryan Povlinski using the Pro-Football-Reference.com’s Game Play Finder for Spreadsheet Sports, in which he collects play by play data from every game, all 256 of them, and collects over 58 variables for every play. The data is given in Excel format. We have data from the 2013 season and 2014 season, but so far the analysis has focused on the latter year’s data. However, exploring our data we have come to the conclusion that about 22 of the given variables will most likely be explanatory in our model, which are listed in Table 1. The many variables can be cut down further with more exploration. Another 4 variables are related to the dependent variable, as they have to do with what the team’s play call is and where the ball is going.

The type of data we have acquired is cross-sectional data. While there is a time variable present in our data, we do not see it significant to test how variables differ over time. We are looking at many individual subjects, in this case offensive plays from the NFL regular season, therefore not making time-series or panel data applicable.

Our data is very complete, especially for its size. Of the almost 2 million data points only 7,693 are not filled in, which is less than 1% of the data. However, it appears the missing data is not absent, but represent an action not happening. For example, if there isn’t a fumble on the play there is no data point in the fumble column. This is relevant for other variables like interceptions, receiver, or pass location, which would not have values if the play wasn’t a pass. To deal with this problem we have created dummy variables with a zero when an action didn’t happen and a 1 when it did.

**Data Processing**

In order to generate a model that will predict a play outcome before each play is run, we had to be vigilant as to the variables included in our analysis. Variables like yards gained or the intended receiver aren’t applicable as they happen once the play has already occurred, meaning they are dependent. Our model is going to predict the play call based on what the defense sees before the play, making those variables irrelevant. Of the 58 variables included in the data, they were cut down to 19 relevant explanatory variables as well as 2 independent variables which are listed in Table 1.

The data present is cross-sectional data, but there are some aspects of time-series analysis that were beneficial, specifically the lagging of variables. The structure of the data didn’t lend itself to time series analysis, as no time variable appears viable to use, but it is reasonable to believe that prior events play a role on future play calls. If a team is constantly throwing interceptions, it is reasonable to believe that they are going to run the ball more. Conversely, if a running back is having a bad game and constantly fumbling the ball, then the team is more likely to throw use the talents of the quarterback and throw the ball. To account for this we created three new variables; Interception Count, Fumble Count, and Sack Count. If a team was intercepted on the prior play then Interception Count was given a value of 1. The same was done for the fumble and sack variables as well.

Between the 2013 and 2014 data there are 65,341 observations. However, we only chose to focus our model on the 2014 data in hopes to capture the most recent sample. We decided the 31,943 plays present was more than enough to grant our analysis enough power and make our results valid. Furthermore, every season has intricacies that our model won’t be able to account for. The dynamic of each NFL team changes drastically from year to year. Players get traded, benched, and injured, while coaching staffs change and teams’ playbooks change. All of these factors make up the error term, and to best decrease the error we have on predictions during this season, it is best to use the most recent data.

Finally, there are two outcomes we would like to predict. The first being whether the offensive team is going to run or pass the ball. Using the variable Play Type, we crafted a variable called Run. This variable was given a value of 1 if a team ran the ball and a 0 if the team passed the ball. The third outcome of Play Type is a sack, which we assumed to be a pass.

**Analysis**

The first step in our analysis was to create a model that could predict whether a team is going to run or pass with a high accuracy. While there are other play types, such as punt, kick, or kneel, those were originally excluded from our data set in order to better predict more significant plays. Those other plays are present and important during an NFL game, but they rarely surprise coaching staffs and wouldn’t be necessary to predict in our analysis. Therefore our first model had a binary outcome.

Before starting our analysis, it was important to get a context for our problem statement. Looking at the 2014 play by play data, NFL teams passed the ball 56% of the time and 58% of the time in 2013. So the worst a model could do at predicting the 2014 season is 56%, because you could merely predict a pass play every time. Also, through our background research we found that a statistical model with an identical problem statement coming out of North Carolina State University was able to predict 20 randomly selected games from the 2011-2014 NFL seasons with an accuracy of 75%. For our model to be relevant and make a difference within the eyes of a defensive coordinator, that is the realm of accuracy our model has to reach.

Our initial instinct is to use a logit/probit model to predict whether a team will run or not. Both of these models are designed to predict a binary outcome which lends itself to our problem. Also, with the types of independent variables we have, it is likely that the relationship with dependent variables is non-linear. For example, a team probably has a much larger probability of running on the one yard line, but most likely about an equal chance of run or pass when on the fifty yard line. This suggests a nonlinear relationship that the model can account for without having to transform the variables ourselves. Furthermore, with the many categorical variables present in the model, such as team or side of field, this creates a large amount of dummy variables. A linear regression does not fit dichotomous independent variables very well, and a logit/probit model performs much better under these circumstances.

After inputting the 2014 data into Stata, we ran our initial Logit regression on all variables we had prior to the snap of the ball excluding the categorical variables. Other variables excluded were factors that didn’t logically have a significant effect on whether a team will run or not which were: Date, Quarter, and Time. A full list of the variables can be examined in the Table 1. The first regression gave us a lot of insight on our data moving forward. Stata was able to pick up multicollinearity that we had not found in our initial examination of the data and the model omitted the variable Score Differential. The effect of this variable was already taken into account because of the inclusion of **TmScore** and **OppScore**. The logit output can be found below in **Table 2**.

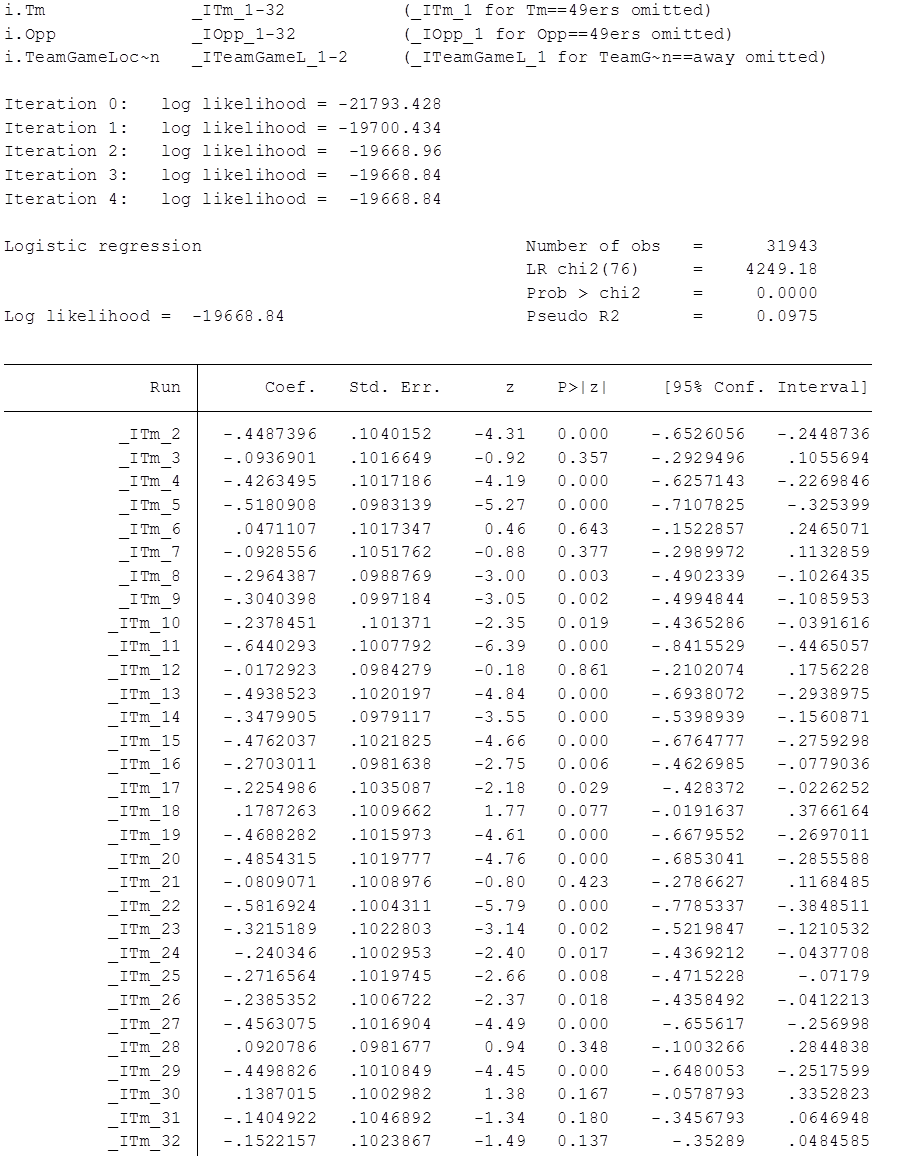
**Table 2. 1st Iteration Logit Output**

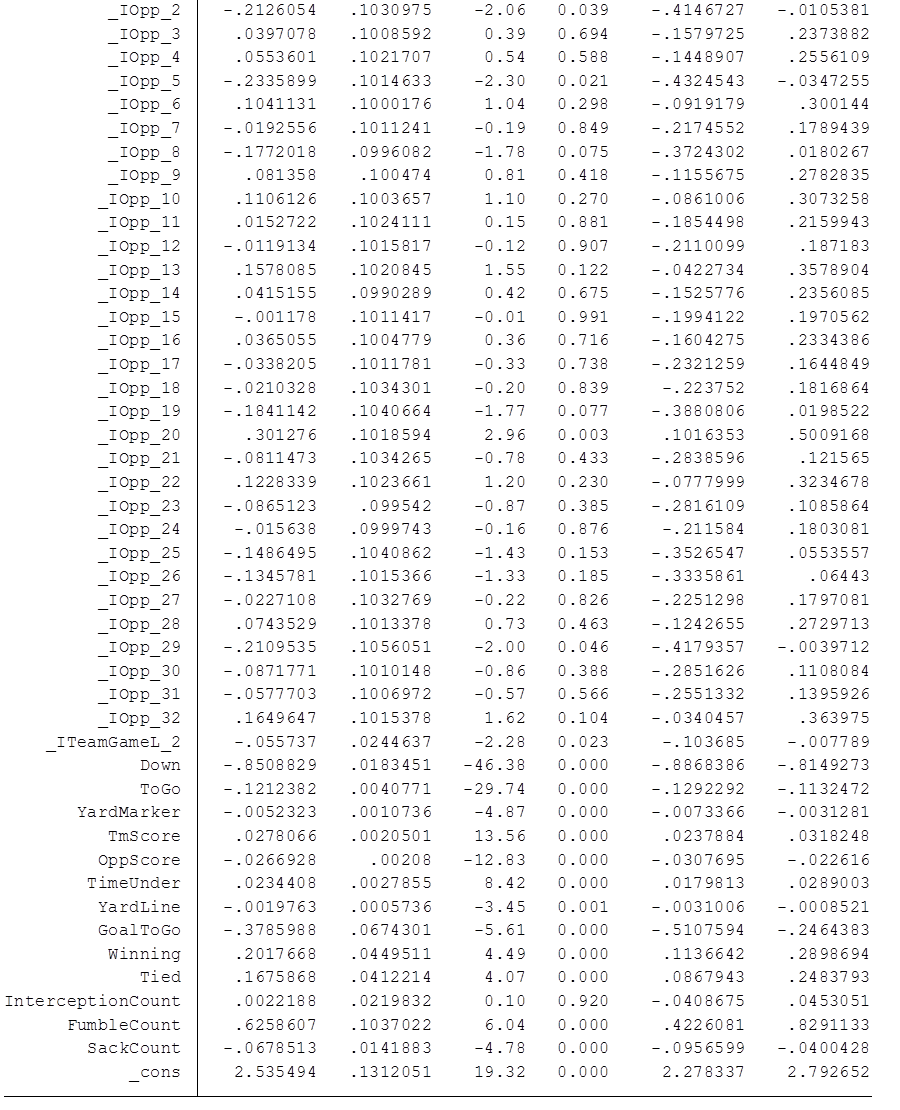
Further inspection of our variables and results led us to much more insight into our problem. The first being that the **Absolute Score Differential** along with **Game Week** were not significant in predicting run or pass on a .05 significance level. It is reasonable to believe that the week of the game has little to do with play calls. Also, a difference in scores is very variable and couldn’t be a very effective predictor, a 3 could mean 53-50 or 3-0. Additionally, score differential is extremely correlated with the scores of the two teams.

Our next iteration of the Logit model included categorical variables as dummy variables. These variables were Team, Opponent, Side of Field, and whether the team is home or away. On top of that we included the manufactured lagged variables that consider whether the last play resulted in an interception, fumble, or sack.

A huge benefit of including the playmaking team as a variable is the huge diversity of NFL teams. Many of the factors that go into a play call can be attributed to a specific team and the dummy variables for team can act as a proxy for all those factors our data doesn’t include. This is beneficial because it can explain for more of the data, rather than those factors being tied to the error term. For example, the Patriots have one of the best quarterbacks in the league, Tom Brady, so it might make sense that the dummy variable for the Patriots, \_ITm\_22, to be negative in comparison to the dummy variable that is omitted, the 49ers. In a logit model it is required to have values for all variables to calculate the exact change in the probability of running the ball, but given that the Patriots coefficient is significant to a 0.001 level with a value of -0.58, it is reasonable to believe that they are generally much more likely to pass.

The inclusion of the dummy variables did impact our model. Primarily, our R2 value went up from 0.0865 to 0.0975, meaning the new model explains more of the variance in the data. Also, we see a decrease in some of the coefficient estimates, such as **Down**, **GoalToGo,** and **Winning**. This can be attributed to the new variables, likely the dummy variables for team, accounting for the some of the variation that was previously being explained by the variables in the 1st model. One significant aspect of the new model is just because some of the dummy team coefficients are not significant, that does not mean they don’t give valuable insight and aren’t useful in the model. The model omits \_ITm\_1, the 49ers, and uses them as the control. So the coefficients of the other teams compare how differently that team calls plays in comparison to the 49ers. So if a coefficient isn’t significant, that just means the outcomes of their plays is not statistically significant from the 49ers. So \_ITM\_12, the Cowboys, calls plays similarly to the 49ers because the p-value is very large at 0.861. On the other hand the Colts, \_ITM\_11, have a p-value that is approximately 0, meaning they call plays significantly differently from the 49ers. This is the same for opponents as well.

**Table 3. 2nd Iteration Logit Output.** 



To see our Logit model’s effectiveness we determined their prediction accuracy. By predicting off data, classifying the predictions greater than 0.5 as a run and less than 0.5 as a pass, and then determining how many were classified correctly, we could calculate how accurate our models predicted. Both models’ accuracies can be found in the **Results** section.

When determining if a logit regression is actually even appropriate, we had to look at some of the assumptions required for our model to be BLUE(Best-Linear-Unbiased-Estimator). Fortunately for logit models, homoscedasticity is not a requirement, so being concerned with omitted variable bias wasn’t as much of an issue. Heteroscedasticity is also much more difficult to detect in these models.

Logistic models differ slightly from the assumptions of other models, as they don’t perfectly follow Gauss-Markov assumptions. Error terms don’t need to be normally distributed and it doesn’t need linear relationships. There are some new assumptions however, and most of them are easily satisfied. The first being logit models require a binary dependent variable, run or pass, where a 1 represents a run and a 0 means a pass. A large sample size is also required, which our dataset covers well. Furthermore, a logit model assumes a linear relationship between the independent variables and log odds. This is the contemporary assumption to assuming linearity, and because of this some relationships can be over and underestimated.

However one important assumption is that the error term needs to be independent. This means that each observation is independent of each other. Unfortunately, in a football game, it is reasonable to assume past plays will impact future ones. The data was not treated as a time series, but it has elements of one. Looking at the residuals of each consecutive play of a specific game could help us test for independence of errors. By examining these graphs we can determine if autocorrelation is present, meaning if a run or pass on the past play could affect the current play call. Normally, a quantitative test like a Durbin-Watson would be more appropriate, but the logistic model is not in a suitable format to run such a test.

As shown in **Figure 1** and **Figure 2** the errors have been plotted against **seqnum**, which corresponds to the number of the offensive play being run. Both of these are single games from the 2014 season. Inspecting the curves there is no noticeable correlation between the error terms, which tells us that the errors are likely independent.

**Figure 1. Residuals Versus Play Number Figure 2. Residuals Versus Play Number**

 **(49ers versus Bears 2014) (Broncos versus Patriots 2014)**



With so many games, it is unrealistic to observe every game’s error terms. However, after testing many random games we found it unlikely that any autocorrelation could pose an issue. Also, it was determined that a play would only in the rarest of cases effect anything but the current game, so testing for independence of errors across separate games wasn’t seen as necessary. Furthermore, we saw no other reasonable independent variables that would make sense to relate to the error term.

Another major concern we had with further model assumptions is collinearity. Slight multicollinearity is expected but if it’s too severe that could bias our estimates. As the factors of each NFL game are related, severe multicollinearity could easily be present. For instance, a team that is winning is likely to have a high team score, or having a lot of 3rd downs could be connected with losing because the opponent’s defense is stopping you. A consequence could mean inflated standard errors and possibly biased coefficients.

**Figure 3. ToGo Versus TmScore Figure 4. TmScore Versus Winning**

Looking at **Figure 3** and **Figure 4**, two correlations can be demonstrated. As a team scores more points, the value of **ToGo** tends to go down as well. This makes sense, as a team scores more points, they are likely more successful with their plays. This leads to 2nd, 3rd and 4th downs being closer to a first down, and the **ToGo** variable consistently being less.

Also, a more obvious relationship, if a team is currently winning; they tend to have a higher score. However, it makes sense that winning the game and having a high score could not have the same effect on play calls. A team that is winning late in the game likely has less aggressive play calls, keeping the ball on the ground. This is shown in **Table 2** because winning has a positive coefficient. Concurrently, that team could have a high score, which is shown by the positive coefficient tied to **TmScore.** However, a team that is losing, even if they have a high score would most likely pass more than run the ball. This is why both variables are needed, but, as discussed earlier, multicollinearity could be a problem.

To truly diagnose if this is bad enough to cause a problem with our model, a more quantitative approach was taken. In **Table 3** the collinearity of each variable, excluding the dummy variables, is quantified.

**Table 3. Collinearity Diagnostics**



Tolerance and VIF are your best indicators of multicollinearity. The closer they are to 1, the more orthogonal the variables are. The tolerance is 1-R2 if a regression was done on that variable. Unfortunately, it seems our intuition was correct and approximately 100% of both teams’ scores could be explained by the other variables.

The direct effect of factors on an NFL field is apparent. Unlike predictors in other regressions, instrumenting or dropping correlated variables isn’t as feasible as we would like. Also, a logistic regression assumes linearity of independent variables and log odds. The dynamic of how variables effect running the ball is likely not as continuous as a logit regression would require. The down of the play can vastly changes the outcome, but it’s hard to generalize that the higher the down, the higher the chance of a run or pass, especially when accounting for how factors like ToGo and the team can affect outcomes. It then became apparent that a method like a classification tree could make for the best model.

RandomForest is one of the most accurate models in classification used today, even rivaling machine learning methods. While it takes a large amount of processing power, it can handle many variables, and generates an unbiased estimate. Unfortunately the one major drawback of using RandomForest is that it is near impossible to determine the exact effect of variables. It is referred to as the “BlackBox” model as it is difficult to see what is going on internally. The most we can do is determined variable importance, but not their exact effect on the classification of run or not. The best way to show the effectiveness of this algorithm is to run on data and determine its accuracy.

To run RandomForest, we used the statistical software R. We used the same variables as we did in our 2nd Logit model, but included an interaction between the down and yards to go for a first down. We ran two models, one trained off the 2014 data, and another trained off all data. To test the power of each model we predicted off subsets of data, such as the 2013 or 2014 season and determined its accuracy. This is done in the same fashion as in the logistic regression, classifying outputs greater than 0.5 as a run (equal to 1) and less than 0.5 as a pass (equal to 0).These were then compared with our logit regressions to determine which models were more powerful.

To better refine the accuracy of the RandomForest models certain parameters can be sharpened. Parameters like *mtry*, the number of variables per split, and *ntrees*, the number of trees to create, can be tuned to optimize the accuracy. Along with that, passes are more prevalent than runs. By tuning the cutoff value where we classify a run versus a pass to favor a pass, meaning less than 0.5, the predictive power of our model can be increased.

RandomForest is not a probabilistic model, but a bootstrapped aggregation of binary trees. Therefore the only assumption required is that the sample it is trained off of is representative of population. This makes the model very stable, and limits the effect of outliers, but means the model would have difficultly predicting future plays if drastic changes in the NFL took place. Multicollinearity is still present and may make a small impact, but not to the same degree as in the logistic model.

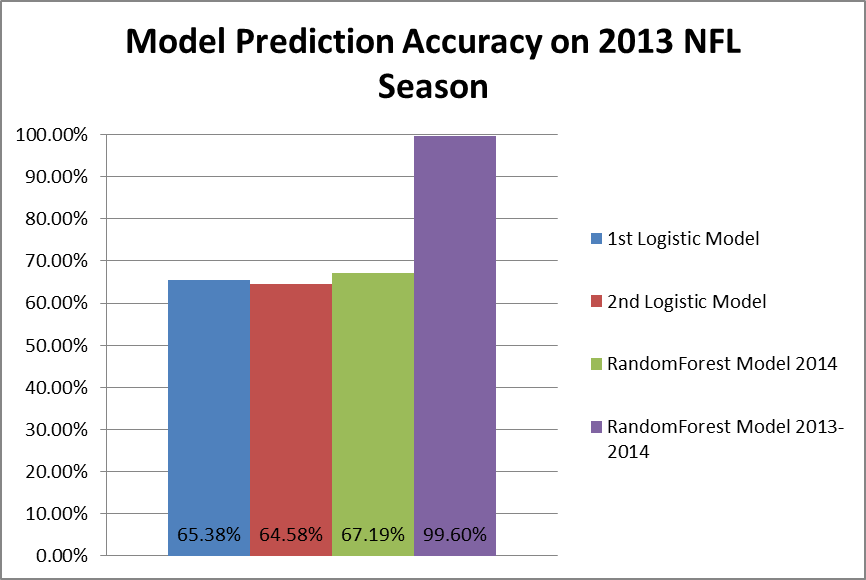
**Results**

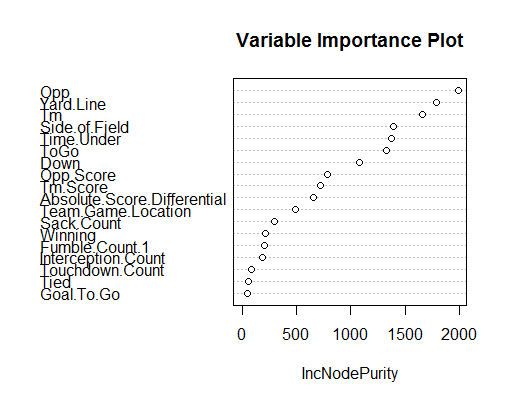
The predictive power of each model on a different year’s data, known as test data, is comparable, in the high 60 percent. However, when looking at the power of the predictive power on each models training data, RandomForest is near perfect. It should be noted, however, that RandomForest models are able to predict off their own training data extremely efficiently, and might not be as effective with predicting off new observations, meaning the model created at North Caroline State University might not have the same predictive power in the upcoming season. Looking at **Figure 5** we see that each model can predict whether a team will run or pass to a high 60% accuracy, 10% better than if you were to always assume a pass.

RandomForest, when predicting on the 2013 data, performed 2% better than the logit models. However, under certain conditions the model can be vastly more accurate. If you subset predictions from the 2013 data to only include 3rd down, we find the 2nd Logit model to predict at a 75% accuracy while the RandomForest model trained off the 2014 data predicts correctly 83% of the time. But when predicting under 1st down, the RandomForest, at 60%, actually does worse than the Logit model predicting at 62%.

As discussed in the **Analysis,** one drawback of RandomForest is we don’t know the exact effect of certain variables as they aren’t given a continuous effect, but used as classifiers. So we don’t get as much insight as the Logit can grant us. The best we can do is determining variables importance within the model which is shown in **Figure 6.** Multicollinearity is known to have a reasonable effect on importance measures, but the plot can give us some valuable insight.

**Figure 5. Model Prediction Accuracy**

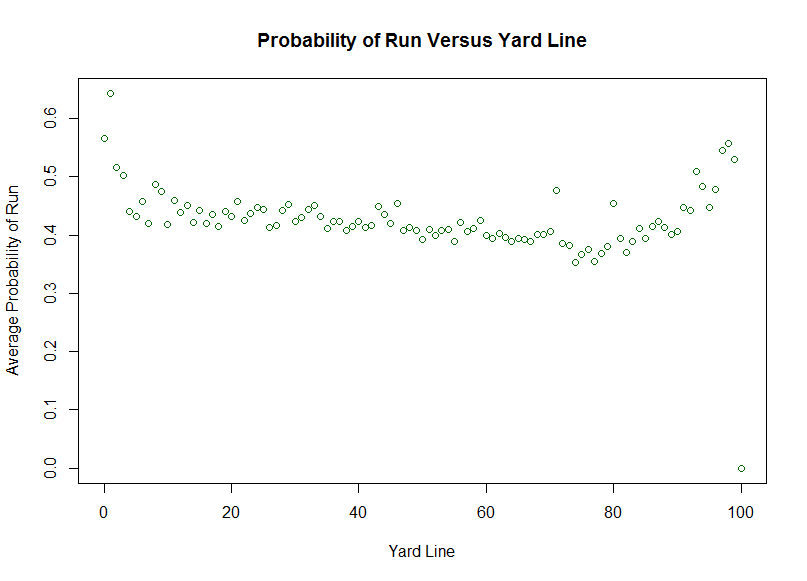


**Figure 6. RandomForest Model 2013-2014 Variable Importance Plot**

While some multicollinearity could be biasing the coefficients of our Stata models, valuable insight that isn’t always intuitive can be found. Looking at Table 1 we can see the coefficient values, and their significance. One important value we found is that interceptions actually don’t deter a team from passing like most would think, but fumbles and sacks do. Also if you look to TimeUnder we can see that the chance of running decreases as time in the quarter goes down. This means that teams are passing more at the end of the quarter then the beginning.

However, we can still determine the importance of variables in RandomForest. The opponent and team seem to have the highest weight when it comes to determining a run or pass, as well as the yard line and which side of the field a team is on. Looking at **Figure 7** the probability of a run is plotted against yard line. We can observe that, whenever a team is near the end zone, there are huge spikes in the chance of a run. Much further analysis can be done with the various factors taken into account in the model, and a more comprehensive understanding when a team will likely run can be determined.

**Figure 7. Probability of Run Versus YardLine**

****

**Conclusions**

The goal of the project was to predict the next play the Offensive Coordinator will call given the current play parameters. Our model predicts the outcome correctly 68% of the time in terms of run versus pass, but only 45% correct prediction when trying to predict the specific play (run right, pass short, etc.). Although our model did not have a prediction accuracy percentage that seems astounding, our model was still able to predict the next play. The model is covering such a wide range of circumstances, such as quarter and leading or losing, so 68% accuracy for run versus pass with all of these specific circumstances not accounted for is good; especially considering individuals that split up the data by quarter and team have only gotten an average of 75% correct (ScienceDaily).

What we have seen from our RandomForest model is that it predicts off its own training data extremely well, but new data can pose a problem. This is why the 75% seen by the model created by North Carolina is not as impressive as initially found. However, it does give some important information. If we were able to train a RandomForest on more current data, and retrain after each game week, it is likely that our model could be extremely accurate and more useful. This would take much more current data and better machines.

Our analysis has also shown us that, while the RandomForest was more accurate, there is value in using more traditional models to grant improved intuition as to what really effects play calls. The coefficients within the logit model provide us instant feedback as to which variables factor into play calls, and how significant they. That is why we have deemed it vital that both models be implemented. Trusting the “BlackBox” of a RandomForest isn’t enough for a proper analysis.

Furthermore, to best implement our model, it is best to use it conditionally. To take the model to the next step, we could look at specific teams, under certain conditions and see its predictive power. For example, under 3rd downs our model was 85% accurate at predicting off new data. It’s possible some teams and easier to predict than others, which could make our model a proficient tool in many cases.

As we talked about earlier, Gary Kubiak of the Denver Broncos is very interested in play analytics. He would like to replace one of his coordinators on the headset with an analyst in order to help Kubiak make real-time decisions. Our model solidifies the idea that analytics in sports is increasing in popularity, and could become very prevalent in the near future.

We reached out to the Air Force Football Defensive Coordinator, Steve Russ, because we were hoping to receive some statistics from him that he may know as a football coach, and once we explained our project, he was very interested in what we were studying. He wants us to reach out to him once the football season has ended so we can talk about our model with the hopes of him being able to apply it to college football. If we had more time to work on the project, we would split up the model into certain specific situations; quarters, teams, and leading or losing. Splitting the model up may allow us to be more accurate because football coaches may be more willing to run earlier in the game, or when they are leading, and pass later in the game, or when they are losing. Specific teams also play a big role in the willingness to run or pass because a team that has a dominant quarterback, like the New England Patriots, are going to be much more willing to throw the ball than a team with a strong run game. There would be many ways to make the model more accurate because of the expansive ideas and play options teams have, but our model does a very good job at predicting the next play a team will call.

**Appendix**

**Table 1: Primary Variables Used**

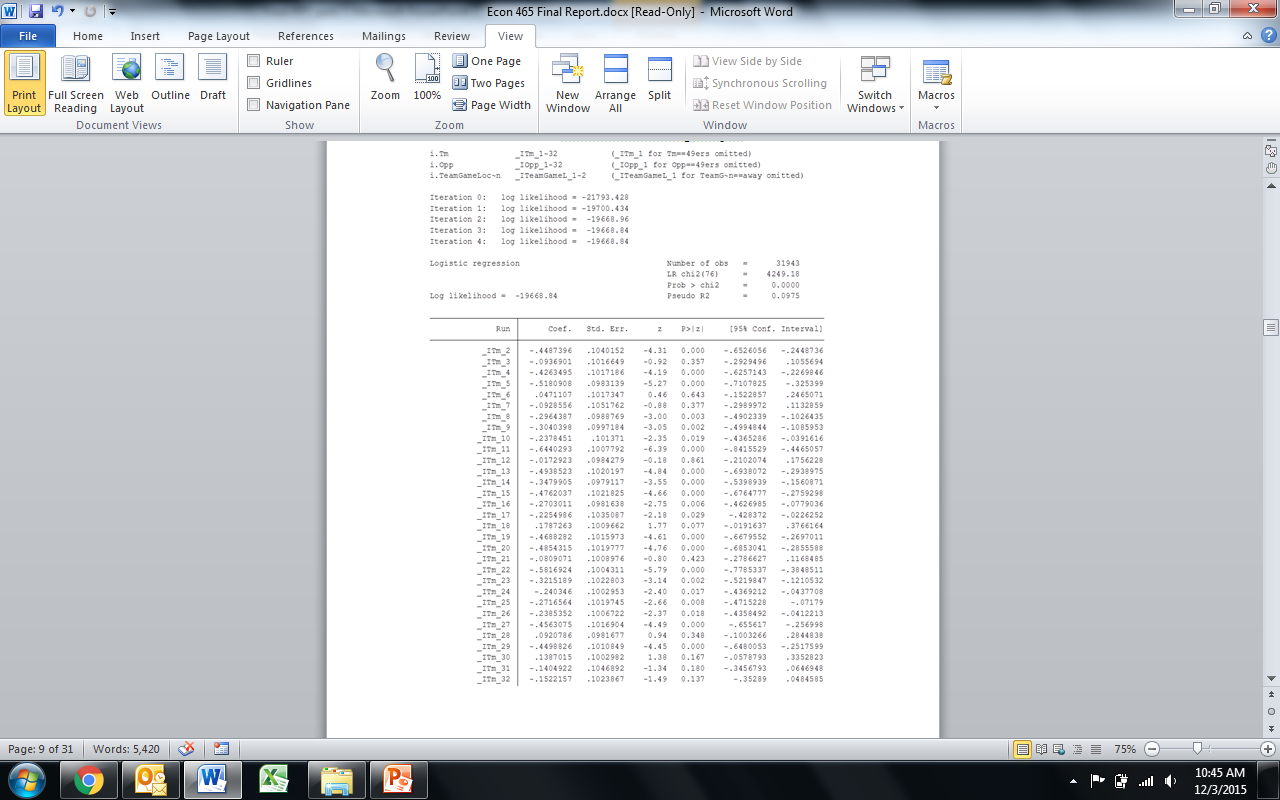


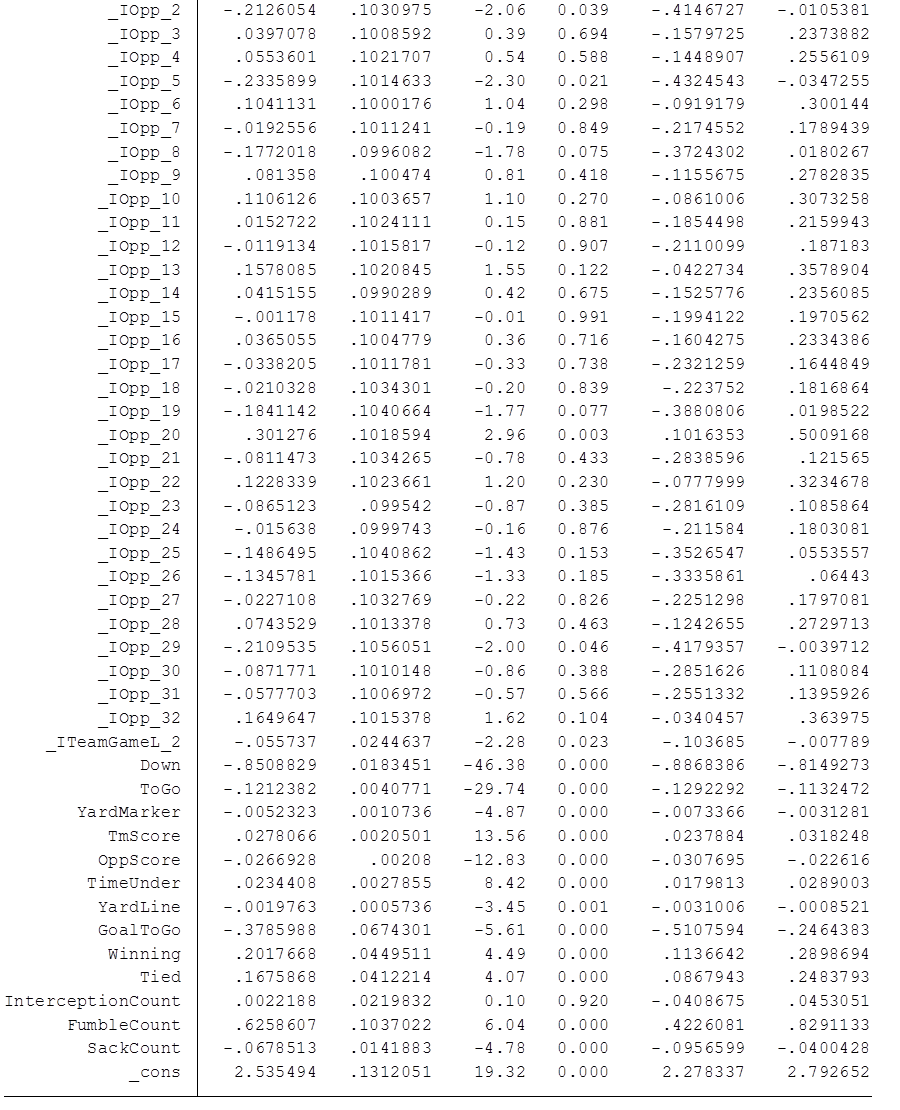


**Table 2. 2nd Iteration Logit Output.**



**Table 3. 2nd Iteration Logit Output.**



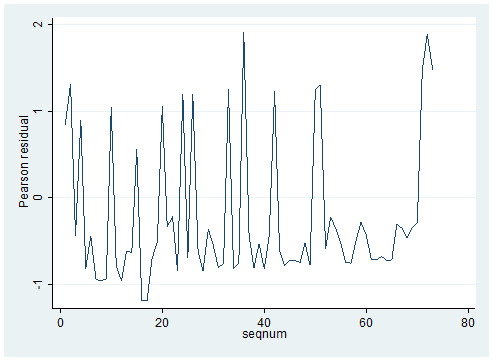


**Figure 1. Residuals Versus Play Number**

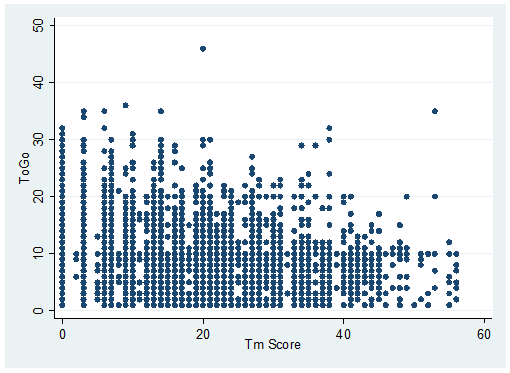
**(49ers versus Bears 2014)**

**Figure 2. Residuals Versus Play Number**

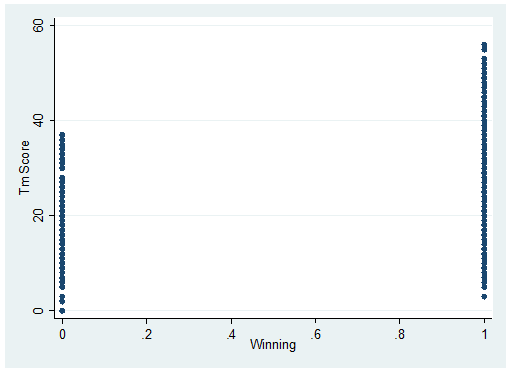
**(Broncos vs. Patriots 2014)**

****

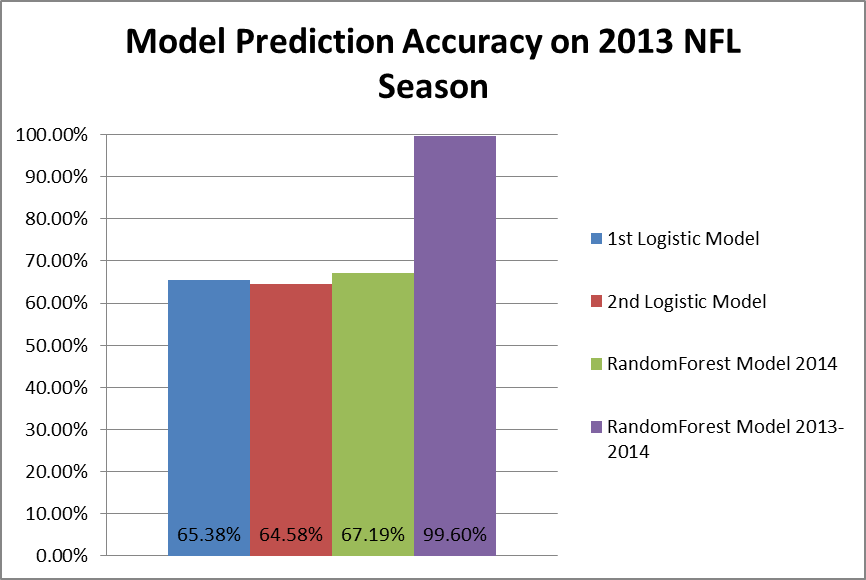
**Figure 3. ToGo Versus TmScore**

****

**Figure 4. TmScore Versus Winning**

****

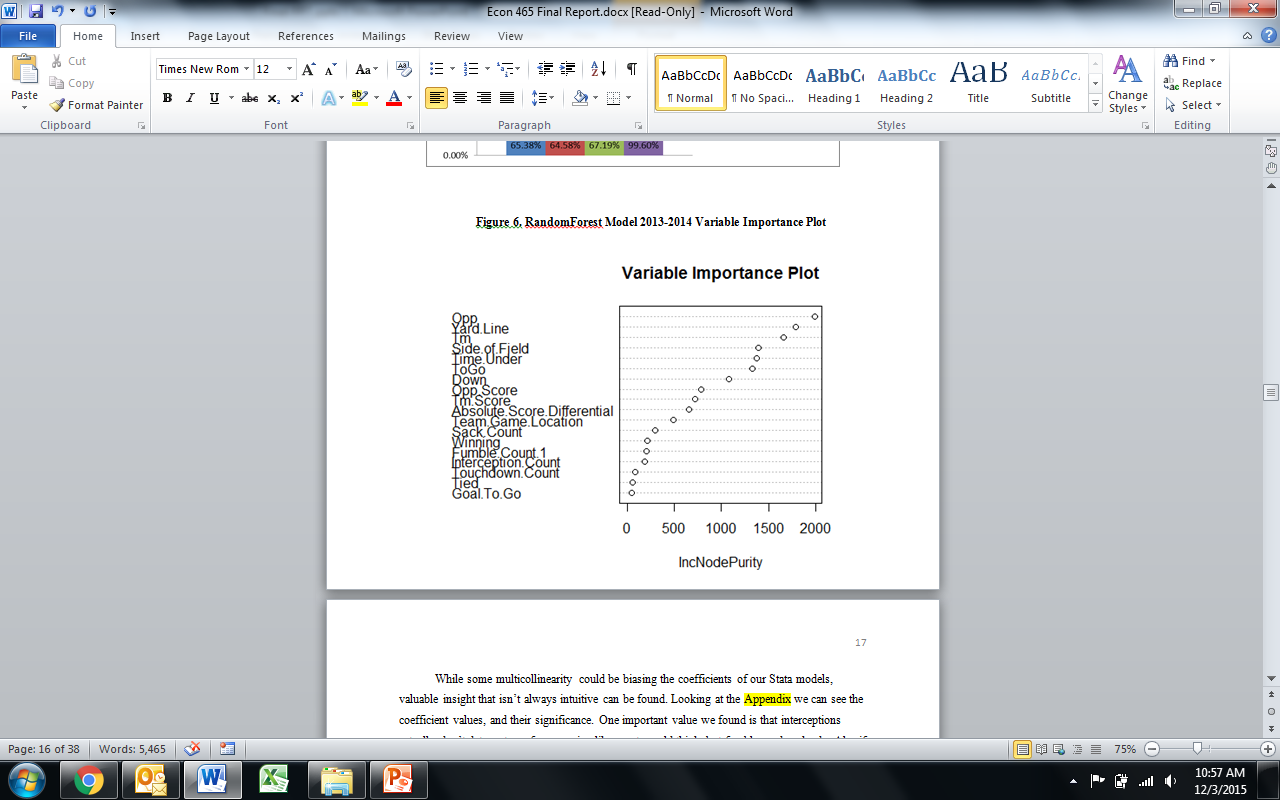
**Figure 5. Model Prediction Accuracy**



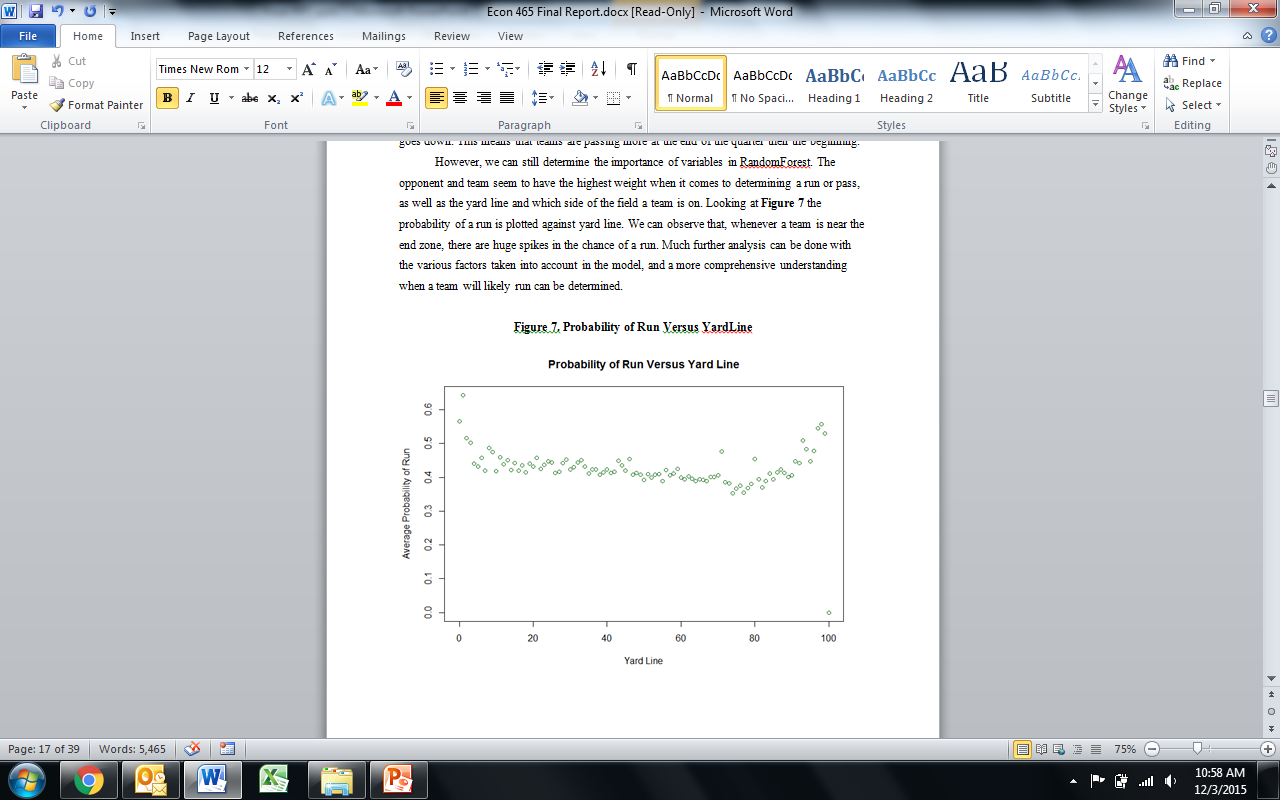
**Table 3. Collinearity Diagnostics**



**Figure 6. RandomForest Model 2013-2014 Variable Importance Plot**



**Figure 7. Probability of Run Versus YardLine**



**Stata Code:**

**2014 data**

*logit Run Down ToGo TmScore OppScore TimeUnder ScoreDifferential AbsoluteScoreDifferential GameWeek YardLine GoalToGo Winning Tied*

*predict predrun*

*replace predrun= 1 if predrun>.5*

*replace predrun= 0 if predrun<=.5*

*table Run predrun*

*xi: logit Run i.Tm i.Opp i.TeamGameLocation Down ToGo TmScore OppScore TimeUnder YardLine GoalToGo Winning Tied InterceptionCount FumbleCount SackCount*

*predict predrun2*

*replace predrun2=1 if predrun2>.5*

*replace predrun2=0 if predrun2<=.5*

*table Run predrun2*

**2013 data**

**Stata Same for both:**

*xi i.Tm i.Opp i.TeamGameLocation i.SideofField*

*predict run2*

*replace run2=1 if run2>.5*

*replace run2=0 if run2<=.5*

*table Run run2*

*collin Down ToGo TmScore OppScore TimeUnder ScoreDifferential AbsoluteScoreDifferential GameWeek YardLine GoalToGo Winning Tied*

***R Code:***

*levels(Final2013data$Side.of.Field)->levels(Final2014data$Side.of.Field)*

*test.err=double(15)*

*for (m in 1:15){*

*fp.rffootball2014 <- randomForest(Run~Tm+Opp+Down+ToGo+Side.of.Field+Tm.Score+Opp.Score+Time.Under+Absolute.Score.Differential+Team.Game.Location+Yard.Line+Touchdown.Count+Goal.To.Go+Success.Count+Interception.Count+Fumble.Count+Sack.Count+Down\*ToGo,data=Final2014data,mtry=m)*

*pred.rffootball2014=predict(fp.rffootball2014,Final2013data)*

*class.rffootball2014=ifelse(pred.rffootball2014>.5,1,0)*

*test.football<-table(class.rffootball2014,Final2013data$Run)*

*test.err[m] = (test.football[1,1]+test.football[2,2])/nrow(Final2013data)*

*}*

*fp.rffootball2014 <- randomForest(Run~Tm+Opp+Down+ToGo+Side.of.Field+Tm.Score+Opp.Score+Time.Under+Absolute.Score.Differential+Team.Game.Location+Yard.Line+Touchdown.Count+Goal.To.Go+Winning+Tied+Interception.Count+Fumble.Count+Sack.Count+Down\*ToGo,data=Final2014data,mtry=11)*

*pred.rffootball2014=predict(fp.rffootball2014,Final2013data)*

*class.rffootball2014=ifelse(pred.rffootball2014>.48,1,0)*

*test.football<-table(class.rffootball2014,Final2013data$Run)*

*(test.football[1,1]+test.football[2,2])/nrow(Final2013data)*

*pred.rffootball2014=predict(fp.rffootball2014,Final2014data)*

*class.rffootball2014=ifelse(pred.rffootball2014>.48,1,0)*

*test.football<-table(class.rffootball2014,Final2014data$Run)*

*(test.football[1,1]+test.football[2,2])/nrow(Final2014data)*

*fp.rffootballall <- randomForest(Run~Tm+Opp+Down+ToGo+Side.of.Field+Tm.Score+Opp.Score+Time.Under+Absolute.Score.Differential+Team.Game.Location+Yard.Line+Touchdown.Count+Goal.To.Go+Winning+Tied+Interception.Count+Fumble.Count+Sack.Count+Down\*ToGo,data=NFLdata,mtry=11)*

*pred.rffootballall=predict(fp.rffootballall,Final2014data)*

*class.rffootballall=ifelse(pred.rffootballall>.48,1,0)*

*test.footballall<-table(class.rffootballall,Final2014data$Run)*

*(test.footballall[1,1]+test.footballall[2,2])/nrow(NFLdata)*

*importance(fp.rffootballall)*

*PlayPredictor <- randomForest(Play~Tm+Opp+Down+ToGo+Side.of.Field+Tm.Score+Opp.Score+Time.Under+Absolute.Score.Differential+Team.Game.Location+Yard.Line+Touchdown.Count+First.Down.Count+Goal.To.Go+Success.Count,data=TrainTwo)*

*pred.play=predict(PlayPredictor,ValTwo)*

*Pred<-table(pred.play,ValTwo$Play)*

*(Pred[1,1]+Pred[2,2]+Pred[3,3]+Pred[4,4]+Pred[5,5])/nrow(ValTwo)*

*downval12<-ValTwo[ which(ValTwo$Down==1),]*

*pred.play1=predict(PlayPredictor,downval12)*

*Pred1<-table(pred.play1,downval12$Play)*

*(Pred1[1,1]+Pred1[2,2]+Pred1[3,3]+Pred1[4,4]+Pred1[5,5])/nrow(downval12)*

*#37.9%*

*downval22<-ValTwo[ which(ValTwo$Down==2),]*

*pred.play2=predict(PlayPredictor,downval22)*

*Pred2<-table(pred.play2,downval22$Play)*

*(Pred2[1,1]+Pred2[2,2]+Pred2[3,3]+Pred2[4,4]+Pred2[5,5])/nrow(downval22)*

*#45.8%*

*downval32<-ValTwo[ which(ValTwo$Down==3),]*

*pred.play3=predict(PlayPredictor,downval32)*

*Pred3<-table(pred.play3,downval32$Play)*

*(Pred3[1,1]+Pred3[2,2]+Pred1[3,3]+Pred3[4,4]+Pred3[5,5])/nrow(downval32)*

*#58.4%*

*downval42<-ValTwo[ which(ValTwo$Down==4),]*

*pred.play4=predict(PlayPredictor,downval42)*

*Pred4<-table(pred.play4,downval42$Play)*

*(Pred4[1,1]+Pred4[2,2]+Pred4[3,3]+Pred4[4,4]+Pred4[5,5])/nrow(downval42)*

*#50.2%*

*firstdownval<-Val[ which(Val$Down==1),]*

*pred.firstdown=predict(fosball,firstdownval)*

*class.firstdown=ifelse(pred.firstdown>.5,1,0)*

*percentfirstdown<-table(class.firstdown,firstdownval$Run)*

*(percentfirstdown[1,1]+percentfirstdown[2,2])/nrow(firstdownval)*

*#60.3%*

*seconddownval<-Val[ which(Val$Down==2),]*

*pred.seconddown=predict(fosball,seconddownval)*

*class.seconddown=ifelse(pred.seconddown>.5,1,0)*

*percentseconddown<-table(class.seconddown,seconddownval$Run)*

*(percentseconddown[1,1]+percentseconddown[2,2])/nrow(seconddownval)*

*#65.4%*

*thirddownval<-Val[ which(Val$Down==3),]*

*pred.thirddown=predict(fosball,thirddownval)*

*class.thirddown=ifelse(pred.thirddown>.5,1,0)*

*percentthirddown<-table(class.thirddown,thirddownval$Run)*

*(percentthirddown[1,1]+percentthirddown[2,2])/nrow(thirddownval)*

*#82.5%*

*fourthdownval<-Val[ which(Val$Down==4),]*

*pred.fourthdown=predict(fosball,fourthdownval)*

*class.fourthdown=ifelse(pred.fourthdown>.5,1,0)*

*percentfourthdown<-table(class.fourthdown,fourthdownval$Run)*

*(percentfourthdown[1,1]+percentfourthdown[2,2])/nrow(fourthdownval)*

*#79%*

**References**

American Statistical Association. "Statistical model predicts with high accuracy play-calling

tendency of NFL teams." ScienceDaily. ScienceDaily, 12 August 2015. <www.sciencedaily.com/releases/2015/08/150812103645.htm>.

"Assumptions of Logistic Regression." Statistics Solutions. N.p., n.d. Web.

Breiman, Leo, and Adele Cutler. "Random Forests." Random Forests. Cal Berkeley, n.d. Web.

Burke, Brian. "GB-SEA Analysis." Advanced Football Analytics. N.p., 19 Jan. 2015. Web.

Wagner-McGough, Sean. "Broncos' Director of Analytics Will Help Gary Kubiak with Game

Decisions." CBSSports.com. N.p., n.d. Web. 27 Aug. 2015.

**Documentation**

None