BAS 320 - Assignment 10 - Model Building

Jack Guptill

## Part 1: Descriptive model for if the Home Team will Win a Football Game

I’m using logistic regression to predict whether or not a football team that is home will win a football game. For this logistic regression I am predicting that they will win the football game (Yes level) from prior games data. This data includes predictors such as the year, what week in the season it is, the points of the winning team, the points lost by the winning team, the amount of yards gained by the winning team, the turnovers by the winning team, the yards lost by the home team, the turnovers lost by the home team, the total points for the home team in a season, the total points against the home team in a season, the strength of the home teams schedule, their offensive ranking, their defensive ranking, if they have been in the playoffs, or if they are a superbowl winner.

I’m making this model primarily out of curiosity and to see how the predictor variables interact to make a prediction for the home team winning.

The best model included the amount of points against the home team, the offensive ranking of the home team,the yards lost by the home team, if the team made the playoffs, how many turnovers the team had, the interaction between offensive ranking and yards lost, the interaction of offensive ranking and turnovers lost, and finally the interaction between yards lost and turnovers lost.

It is important to note that this model doesn’t hold much practical significance due to having to need to know some variables information that you would only know after a game concluded. This model is purely build to see from a descriptive perspective how the predictor variables interact and which predictor variables are best to explain the home team winning.

The data I’m using comes from kaggle, however a lot of data transformations and joins were done personally with python. The data has a total of 5104 rows with 18 total variables and 17 predictor variables.

Kaggle Link: <https://www.kaggle.com/datasets/sujaykapadnis/nfl-stadium-attendance-dataset> Github: <https://github.com/jguptil1/Portfolio-Projects/tree/main/FootballStandings>

S <- step(naive,scope=list(lower=naive,upper=full),direction="both",trace=0)  
S

##   
## Call: glm(formula = home\_team\_winner ~ offensive\_ranking + points\_against +   
## X + yds\_loss + turnovers\_loss + turnovers\_win + offensive\_ranking:yds\_loss +   
## points\_against:yds\_loss + points\_against:turnovers\_loss +   
## offensive\_ranking:turnovers\_win + points\_against:turnovers\_win,   
## family = binomial, data = SmallLog)  
##   
## Coefficients:  
## (Intercept) offensive\_ranking   
## 7.416e+00 4.768e-01   
## points\_against X   
## -1.957e-02 1.382e-04   
## yds\_loss turnovers\_loss   
## -1.886e-02 8.848e-01   
## turnovers\_win offensive\_ranking:yds\_loss   
## -5.361e-01 -9.527e-04   
## points\_against:yds\_loss points\_against:turnovers\_loss   
## 4.854e-05 -2.765e-03   
## offensive\_ranking:turnovers\_win points\_against:turnovers\_win   
## -3.780e-02 1.837e-03   
##   
## Degrees of Freedom: 999 Total (i.e. Null); 988 Residual  
## Null Deviance: 1375   
## Residual Deviance: 1180 AIC: 1204

## Part 2: Predictive model for the Points of the Winning Team

I’m using linear regression to predict what the points will be for the winning team using the year, what week in the season it is, the points of the winning team, the points lost by the winning team, the amount of yards gained by the winning team, the turnovers by the winning team, the yards lost by the home team, the turnovers lost by the home team, the total points for the home team in a season, the total points against the home team in a season, the strength of the home teams schedule, their offensive ranking, their defensive ranking, if they have been in the playoffs, or if they are a Superbowl winner.

I’m making this model because I am genuinely curious to how you would predict the points of the winning team by knowing the other variables. This could have applications for sports betting or knowing how well your team will do in a given game.

The data I’m using comes from kaggle, however a lot of data transformations and joins were done personally with python. The data has a total of 5104 rows with 18 total variables and 17 predictor variables.

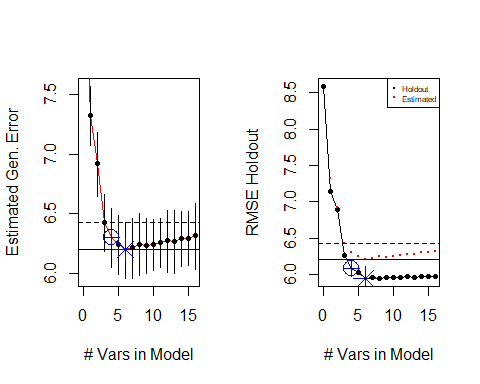
Kaggle Link: <https://www.kaggle.com/datasets/sujaykapadnis/nfl-stadium-attendance-dataset>

Github: <https://github.com/jguptil1/Portfolio-Projects/tree/main/FootballStandings>

The best predictors that ended up in the model is the home team winning the game, the year of the game, the points lost by the team, yards won by the winning team, the turnovers by the winning team, the yards lost, hte turnovers lost, the total points for the team in a season, the total points against the team in a season, and if the team made it to the Superbowl in a given season. With these ten predictor variables the estimated generalized error of the model from K-fold cross validation is 6.01. The RMSE of the holdout was ~6.65 meaning that there is a 10.52% increase in error in the holdout. This tells us that the model is not great at predicting new data as the percentage of error is greater than 10% for hold out data.

BM <- build\_model(pts\_win~.,data=TRAIN2,type="predictive",seed=320,holdout=HOLDOUT2, prompt=FALSE)

##   
## Model with lowest estimated generalization error has:  
## pts\_loss yds\_win turnovers\_win yds\_loss turnovers\_loss points\_for  
## Closest Formula: pts\_win ~ pts\_loss+yds\_win+turnovers\_win+yds\_loss+turnovers\_loss+points\_for  
##   
## Model selected with one standard deviation rule has:  
## pts\_loss yds\_win turnovers\_win turnovers\_loss  
## Closest Formula: pts\_win ~ pts\_loss+yds\_win+turnovers\_win+turnovers\_loss



BM$CVtable

## k SquaredEstGenErr SD EstGenErr RMSEholdout  
## 0 79.49042 4.520046 8.915740 8.587163  
## 1 53.61107 3.616472 7.321958 7.136993  
## 2 47.89533 3.716477 6.920646 6.895649  
## 3 41.37286 3.053173 6.432174 6.254665  
## + 4 39.70707 3.088712 6.301354 6.084114  
## 5 39.04657 3.075248 6.248726 6.026866  
## \* 6 38.44730 2.902038 6.200589 5.949464  
## 7 38.64196 3.098193 6.216265 5.952416  
## 8 39.04236 3.243911 6.248388 5.945944  
## 9 38.87946 2.862055 6.235340 5.958231  
## 10 38.98085 2.642075 6.243465 5.964842  
## 11 39.17855 2.552450 6.259277 5.957827  
## 12 39.38775 3.257095 6.275966 5.966342  
## 13 39.35297 3.321485 6.273195 5.963157  
## 14 39.65375 2.915440 6.297122 5.969338  
## 15 39.67310 2.854574 6.298659 5.966110  
## 16 39.93964 3.463646 6.319782 5.967076

6.645071/6.012168 #ratio of RMSE Holdout ot Estimated Generalized Error

## [1] 1.10527