Jack Pierce
INFO 4390
Project Part 2
3/17/2020

The goal was to create a model that could predict if the favorited NFL team would cover the spread, but I fell short of that goal with models that did not have the accuracy rate to reliably make decisions off of. The data I chose to build models was NFL betting data in order to build a model that could accurately predict if the favored team would cover the spread or not. The data I had originally thought I would have consisted of only 18 variables and that I would have to create a couple variables in order to have a response variable that show if the favored team covered the betting spread established prior to the start of the game. This, however, was not the case as I realized that the data came with an R script that created over 100 variables. The data included weather, team, game result among other data on NFL games dating back to the 1960's, but the betting data on the games only reached as far back as the 1980 season. So, the amount of games I was able to use in my models fell from 12,400 to 9,666 due to the lack of betting data from the games in the 60's and 70's. Most of the 170 variables created from the R script provided were different variations of the same data or variables used to connect different tables or connect various columns with others, such as team data with the favorited team for a specific game. For the models, only about 11 variables were used in the end and those variables can be seen in the code which is in the index. Also, the data was split into training and testing. Subsets by splitting by time, so the training data consisted of games from 1980 to 2012 seasons, and the testing consisted of the data from the 2013 to 2018 seasons. Classification models were used to predict if the favorited team covered the spread or not since a regression model would just not make sense for a question like this. I used 4 different types of classification models to try and predict if the favorited team covered the spread: general linear model, linear discriminant analysis model, random forest model, and a gradient boosted model. All of these 4 models produced an accuracy rate within a couple percent of each other, but the GBM model had the highest of 53.36%. Overall, I was not satisfied with how my models tested, they barely beat the flip of a coin, but no matter how much I tuned or altered the variables considered in each model, the best accuracy rate I could achieve was 53.36%. While this data was pretty extensive, I believe that incorporating more data into the models would help to improve its accuracy rate to something more rewarding. The data I would suggest incorporating would be like stats on previous matchups in recent years between the two teams, as well as player data incorporating their stats as well if they're injured or not. With the incredible amount of data that is now being recorded on NFL games with the power of AWS, there has to be variables that give better insight into which team will cover the spread. All this data would make models much more complex than the ones here and would require more computing power. I hope this can be done so that someone can shake up Las Vegas and the bookmakers there.

**APPENDIX:**

R Code for modeling:

```
library(corpcor)
library(ggplot2)
library(lattice)
library(caret)
library(fastDummies)
library(dplyr)
library(tidyverse)
library(ranger)
library(randomForest)
library(broom)
library(ISLR)
library(plotROC)
library(MASS)
library(gbm)

#Data from 1979 and before has little to no spread data
spread_data <- nfl[nfl$schedule_season>1979,]

nfl$spread_favorite_result <- if_else(
  nfl$spread_favorite_cover_result == "Cover", 1, 0)

#Splitting the data into training and testing data sets
training <- nfl[nfl$schedule_season>1979 & nfl$schedule_season<=2012,]
testing <- nfl[nfl$schedule_season>2012,]

#First Logistic model GLM
spread_glm <- glm(spread_favorite_result ~
                    division_matchup +
                    schedule_sunday + schedule_playoff +
                    spread_outlier + spread_favorite +
                    score_avg_pts_for_roll_lag.x + score_avg_pts_against_roll_lag.x +
                    score_avg_pts_for_roll_lag.y + score_avg_pts_against_roll_lag.y +
                    weather_rain + weather_snow,
                  family = "binomial",
                  data = training)
summary(spread_glm)


spread_lda <- lda(spread_favorite_result ~
                    division_matchup +
```

```
              schedule_sunday + schedule_playoff +
              spread_outlier + spread_favorite +
              score_avg_pts_for_roll_lag.x + score_avg_pts_against_roll_lag.x +
              score_avg_pts_for_roll_lag.y + score_avg_pts_against_roll_lag.y +
              weather_rain + weather_snow,
            data = training)
fits_lda <- predict(spread_lda, newdata = testing)
confusionMatrix(table(fits_lda$class, testing$spread_favorite_result))

set.seed(1982)
training$spread_favorite_result <- as.factor(training$spread_favorite_result)
testing$spread_favorite_result <- as.factor(testing$spread_favorite_result)
tune_grid_rf <- expand.grid(mtry = 2:11,
                    splitrule = "gini",
                    min.node.size = 10)
train_control_rf <- trainControl(method = "cv",
                        number = 10,
                        search = "grid")
spread_rf <- train(spread_favorite_result ~
              division_matchup +
              schedule_sunday + schedule_playoff +
              spread_outlier + spread_favorite +
              score_avg_pts_for_roll_lag.x + score_avg_pts_against_roll_lag.x +
              score_avg_pts_for_roll_lag.y + score_avg_pts_against_roll_lag.y +
              weather_rain + weather_snow,
            data = training,
            method = "ranger",
            num.trees = 500,
            importance ="impurity",
            trControl = train_control_rf,
            tuneGrid = tune_grid_rf)

spread_rf
plot(spread_rf)

test_preds_rf <- predict(spread_rf, newdata = testing)
confusionMatrix(table(test_preds_rf, testing$spread_favorite_result))

set.seed(99)
grid <- expand.grid(interaction.depth = c(1, 3, 5),
              n.trees = seq(100, 500, by = 100),
              shrinkage = c(.01, 0.001),
              n.minobsinnode = 10)
```

```
trainControl <- trainControl(method = "cv", number = 5)

gbm_boston <- train(spread_favorite_result ~
                    division_matchup +
                    schedule_sunday + schedule_playoff +
                    spread_outlier + spread_favorite +
                    score_avg_pts_for_roll_lag.x + score_avg_pts_against_roll_lag.x +
                    score_avg_pts_for_roll_lag.y + score_avg_pts_against_roll_lag.y +
                    weather_rain + weather_snow,
                  data = training,
                  distribution = "bernoulli",
                  method = "gbm",
                  trControl = trainControl,
                  tuneGrid = grid,
                  verbose = FALSE)
gbm_boston

test_pred_gbm <- predict(gbm_boston, newdata = testing)
confusionMatrix(test_pred_gbm, testing$spread_favorite_result)
```

Data and R code provided:

https://www.kaggle.com/tobycrabtree/nfl-scores-and-betting-data#spreadspoke_scores.csv