Score Differential v8

2024-11-23

Table of Contents

Notes before proceeding

- The OLS model takes NO ACCOUNT for Heteroskedasticity (The T vals are very high or low however) or Autocorrelation (wont have a issue with autocorrelation).
- The current data ignores how good the teams are (offensive and defensive). For that reason our prediction power doesn't seem incredible... However the entire point of my thesis is that with the proper data then we can improve our results.
- The relationships are definitely non-linear. In some cases though, i could engineer the data to help OLS... for example the minutes left in the game could be two columns one that is the half and one that is the time from 0-30.
- i added a random (simulated/useless) variable to the data to give us a reference point of what variables are useless in our random forest model.
- Week 1 was removed from the data due to how we created some of the variables.

My Thoughts on Point Differential

The score differential without a doubt has a effect and predicting power on whether or not a team attempts a 4th down.

The question is does it have prediction power or a effect on the actual conversion of the 4th down. I would argue not.

My one fear was originally that score differential signals a "better" team. Then since that team is "better" the 4th down result will receive prediction power from the score differential.

However if we account for how good the team are i believe that argument would not hold up. Even without accounting for how good teams are i found that the score differential was almost useless for predicting the 3rd down conversion result (in this case we used 3rd down plays to stand in for 4th down conversions).

In my head this is my point. Pretend the panthers (bad team) are playing the saints (average team). the panthers are losing by 30 points so the score differential is -30. Obviously here the panthers probably have a below average chance of conversion on a 4th down. However on 4th down the teams swap and the chiefs (good team) take over for the panthers. I don't think that it is fair to say that the 30 point deficit will make it harder for the chiefs to convert on 4th down.

Measures of prediction power Random Forest

MDI (gini/node purity)

As Gini impurity decreases (meaning nodes become more pure), the MDI value increases.

This measures how good a variable is at promoting node purity during the splits. This is for the training of the model. This leads MDI to not always be the best measure.

That is why the random variable did good in this measure. The splits in the trees will sometimes use useless variables. This is because it fits the training data and it is very normal for RF to use bad predictors (that is the entire point of randomforest). However MDI can make it appear that unimportant variables are in fact important.

MDA (Mean Descrease in Accuracy)

This looks are more how the model would do if we removed the variable in question from the model.

A negative MDA means that the model would do better without the variable.

It is more robust and we need to pay more attention to this than MDI.

Literature topics

- hot hand
- momentum
- coaches decisions
- the effect of the hot seat (that college coaches paper)

look for - score differential (coaches decisions) - AUCs of other peoples models. (to prove that the extended information is better)

Libraries

The Data

```
model_3rd3 <- read_csv("model_3rd3.csv")
```

```
## Rows: 7022 Columns: 20
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (20): week, ydstogo, yardline_100, posteam_timeouts_remaining, defteam_t...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
model_4th3 <- read_csv("model_4th3.csv")
```

```
## Rows: 4226 Columns: 19
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (19): week, attempt, ydstogo, yardline_100, minutes_remaining, posteam_t...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

We are gonna select every single column but we will make sure it include

Points about the data

- week 1 was removed
- The data has been prepared for OLS.
- The OLS (variances) we are using ignore Heteroskedasticity Autocorrelation
- the data sets have no NAs (temp and wind where removed to acheive this)

```
model_4th3 <- model_4th3 %>%
  select(
    attempt,              # Target variable: Whether coach elected to go for it on 4th down (1/0) (0 = p
    # Game situation
    down1_pct,            # run % on 1st down
    down2_pct,            # Run % on 2nd down
    down3_pct,            # Run % on 3rd down
    opp_scss = successes,             # Number of successful 4th down attempts the opposing team had las
    opp_fails = failures,             # Number of failed 4th down attempts the opposing team had last ga
    score_diff,           # Point differential (positive = winning)
    min_rem = minutes_remaining,    # Minutes remaining in game

    # Play specifics
    ydstogo,             # Yards needed for first down
    yardline_100,        # Distance from opponent's endzone

    # Game management
    offtimes = posteam_timeouts_remaining,  # Offensive team's timeouts left
    deftimes = defteam_timeouts_remaining,  # Defensive team's timeouts left

    # Game context
```

```
    week,                # Week of the season
    prep_days,           # Days to prepare for game
    home,                # Whether team is home (1/0) (0 = away)
    dome,                # Whether game is in dome (1/0) (0 = outdoor or open)

    # Play type indicators
    KICKOFF,             # If the team received the ball from the opponent via kickoff (1/0)
    PUNT,                # If the team received the ball from the opponent via a punt (1/0)
    #if kickoff and punt are both 0 then the team received the ball from the opponent via a different w

    random_var           # Random variable for analysis
  )
```

```
model_3rd3 <- model_3rd3 %>%
  select(
    converted,            # Target variable: Whether the conversion was successful (1/0)
    # Game situation
    down1_pct,           # run % on 1st down
    down2_pct,           # Run % on 2nd down
    down3_pct,           # Run % on 3rd down
    opp_scss = successes,           # Number of successful 4th down attempts the oposing team had last
    opp_fails = failures,            # Number of failed 4th down attempts the oposing team had last gam
    score_diff,          # Point differential (positive = winning)
    min_rem = minutes_remaining,   # Minutes remaining in game

    # Play specifics
    ydstogo,             # Yards needed for first down
    yardline_100,        # Distance from opponent's endzone
    rush,                # Whether it's a rushing play (1/0) (0 = pass)

    # Game management
    offtimes = posteam_timeouts_remaining,  # Offensive team's timeouts left
    deftimes = defteam_timeouts_remaining,  # Defensive team's timeouts left

    # Game context
    week,                # Week of the season
    prep_days,           # Days to prepare for game
    home,                # Whether team is home (1/0) (0 = away)
    dome,                # Whether game is in dome (1/0) (0 = outdoor or open)

    # Play type indicators
    KICKOFF,             # If the team received the ball from the opponent via kickoff (1/0)
    PUNT,                # If the team received the ball from the opponent via a punt (1/0)
    #if kickoff and punt are both 0 then the team received the ball from the opponent via a different w

    random_var           # Random variable for analysis
  )
```

```
glimpse(model_3rd3)
```

```
## Rows: 7,022
## Columns: 20
## $ converted    <dbl> 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, ~
```

```
## $ down1_pct    <dbl> 72.0, 72.0, 63.2, 72.0, 63.2, 72.0, 72.0, 72.0, 63.2, 63.~
## $ down2_pct    <dbl> 61.1, 61.1, 23.5, 61.1, 23.5, 61.1, 61.1, 61.1, 23.5, 23.~
## $ down3_pct    <dbl> 20.0, 20.0, 13.3, 20.0, 13.3, 20.0, 20.0, 20.0, 13.3, 13.~
## $ opp_scss     <dbl> 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, ~
## $ opp_fails    <dbl> 3, 3, 5, 3, 5, 3, 3, 3, 5, 5, 5, 3, 5, 3, 5, 5, 5, 5, 3, ~
## $ score_diff   <dbl> 0, 0, -7, 7, -7, 7, 0, 0, -3, -3, -3, 0, -3, 3, -10, -10,~
## $ min_rem      <dbl> 57.450000, 53.400000, 50.450000, 48.450000, 46.900000, 43~
## $ ydstogo      <dbl> 2, 4, 8, 10, 9, 5, 1, 3, 2, 1, 5, 29, 2, 3, 3, 4, 3, 3, 3~
## $ yardline_100 <dbl> 56, 10, 73, 41, 50, 73, 55, 27, 36, 16, 9, 32, 23, 3, 68,~
## $ rush         <dbl> 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, ~
## $ offtimes     <dbl> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 2, 3, 2, 2, 2, 2, 3, ~
## $ deftimes     <dbl> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 3, 2, 3, 3, 3, 3, 2, ~
## $ week         <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ~
## $ prep_days    <dbl> 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, ~
## $ home         <dbl> 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, ~
## $ dome         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ KICKOFF      <dbl> 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, ~
## $ PUNT         <dbl> 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ random_var   <dbl> 1.137847564, 0.370376882, -2.109306283, 1.475054090, 0.61~
```

```
glimpse(model_4th3)
```

```
## Rows: 4,226
## Columns: 19
## $ attempt      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, ~
## $ down1_pct    <dbl> 63.2, 72.0, 63.2, 72.0, 72.0, 63.2, 72.0, 63.2, 72.0, 63.~
## $ down2_pct    <dbl> 23.5, 61.1, 23.5, 61.1, 61.1, 23.5, 61.1, 23.5, 61.1, 23.~
## $ down3_pct    <dbl> 13.3, 20.0, 13.3, 20.0, 20.0, 13.3, 20.0, 13.3, 20.0, 13.~
## $ opp_scss     <dbl> 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, ~
## $ opp_fails    <dbl> 5, 3, 5, 3, 3, 5, 3, 5, 3, 5, 3, 3, 1, 1, 3, 1, 1, 3, 1, ~
## $ score_diff   <dbl> -7, 7, -7, 7, 0, -3, 0, -10, 10, -10, 0, -4, 4, -3, 3, -3~
## $ min_rem      <dbl> 50.366667, 48.383333, 46.300000, 43.516667, 39.316667, 33~
## $ ydstogo      <dbl> 8, 10, 8, 5, 2, 5, 19, 5, 4, 4, 5, 6, 24, 8, 1, 20, 15, 1~
## $ yardline_100 <dbl> 73, 41, 49, 73, 26, 9, 22, 70, 63, 45, 8, 22, 64, 73, 72,~
## $ offtimes     <dbl> 3, 3, 3, 3, 3, 3, 0, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, ~
## $ deftimes     <dbl> 3, 3, 3, 3, 3, 3, 1, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, ~
## $ week         <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ~
## $ prep_days    <dbl> 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, ~
## $ home         <dbl> 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, ~
## $ dome         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ KICKOFF      <dbl> 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, ~
## $ PUNT         <dbl> 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, ~
## $ random_var   <dbl> -0.42146137, -0.77635780, 1.75755225, 1.26955117, 0.50321~
```

```
# Print unique columns in each dataset
cat("Unique to 3rd down:", setdiff(names(model_3rd3), names(model_4th3)), "\n")
```

```
## Unique to 3rd down: converted rush
```

```
cat("Unique to 4th down:", setdiff(names(model_4th3), names(model_3rd3)), "\n")
```

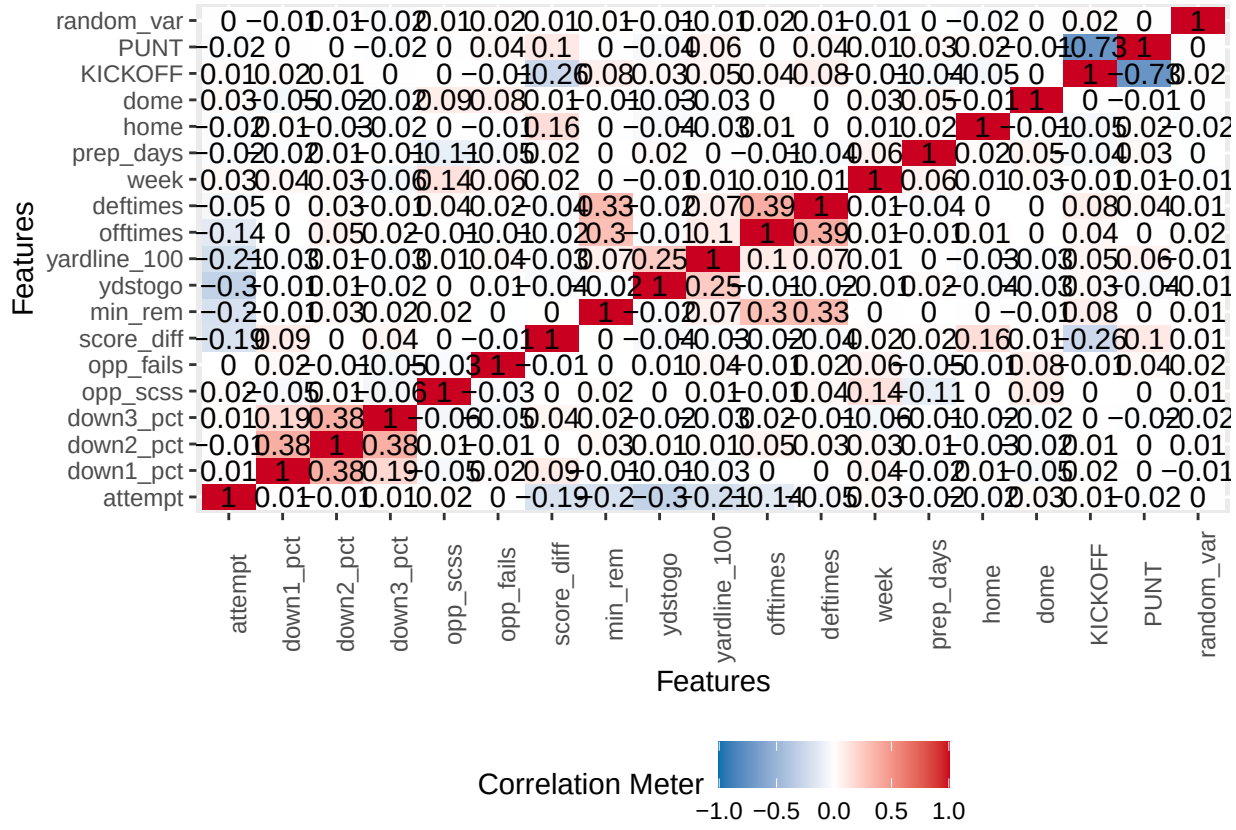
```
## Unique to 4th down: attempt
```

Correlation analysis

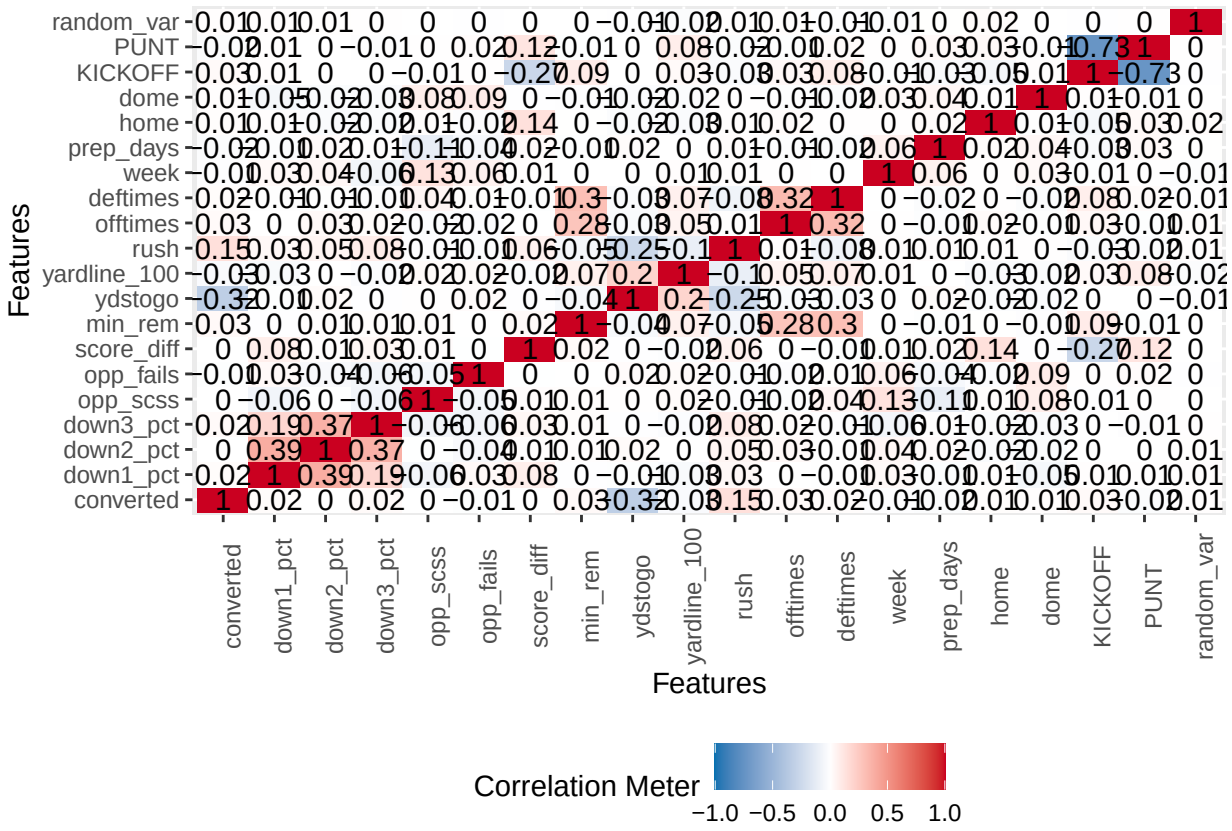
scale the none binary data without changing names

```
# Scale non-binary variables while preserving column names
model_3rd3 <- model_3rd3 %>%
  mutate(across(c(down1_pct, down2_pct, down3_pct, opp_scss, opp_fails, score_diff,
                  min_rem, ydstogo, yardline_100, prep_days, random_var), scale))

model_4th3 <- model_4th3 %>%
  mutate(across(c(down1_pct, down2_pct, down3_pct, opp_scss, opp_fails, score_diff,
                  min_rem, ydstogo, yardline_100, prep_days, random_var), scale))
```

```
plot_correlation(model_4th3)
```




```
plot_correlation(model_3rd3)
```



OLS

4th down "will the coach attempt to go for it?"

```
# Fit OLS model
OLS_4th <- lm(attempt ~ ., data = model_4th3)
summary(OLS_4th)
```

```
##
## Call:
## lm(formula = attempt ~ ., data = model_4th3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8257 -0.2769 -0.1271  0.2473  1.2600
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.3160314  0.0322914   9.787  < 2e-16 ***
## down1_pct     0.0077406  0.0065025   1.190  0.23396
## down2_pct    -0.0032155  0.0068676  -0.468  0.63966
## down3_pct     0.0096273  0.0064719   1.488  0.13695
## opp_scss      0.0073908  0.0060942   1.213  0.22530
```

```
## opp_fails     0.0004991  0.0060073   0.083  0.93380
## score_diff   -0.0903594  0.0063111 -14.318  < 2e-16 ***
## min_rem      -0.0763195  0.0064332 -11.863  < 2e-16 ***
## ydstogo      -0.1209819  0.0061534 -19.661  < 2e-16 ***
## yardline_100 -0.0555800  0.0062399  -8.907  < 2e-16 ***
## offtimes     -0.0559352  0.0087679  -6.380 1.97e-10 ***
## deftimes      0.0247389  0.0085647   2.888  0.00389 **
## week          0.0024273  0.0011569   2.098  0.03596 *
## prep_days    -0.0030699  0.0060230  -0.510  0.61030
## home         -0.0001392  0.0120401  -0.012  0.99078
## dome          0.0183271  0.0155110   1.182  0.23745
## KICKOFF      -0.0233333  0.0183838  -1.269  0.20443
## PUNT         -0.0217328  0.0182463  -1.191  0.23369
## random_var   -0.0010197  0.0059464  -0.171  0.86386
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3857 on 4207 degrees of freedom
## Multiple R-squared:  0.2011, Adjusted R-squared:  0.1977
## F-statistic: 58.83 on 18 and 4207 DF,  p-value: < 2.2e-16
```

```
vif(OLS_4th)
```

```
##    down1_pct    down2_pct    down3_pct     opp_scss    opp_fails   score_diff
##     1.200585     1.339217     1.189334     1.054568     1.024679     1.130962
##      min_rem      ydstogo yardline_100     offtimes     deftimes         week
##     1.175143     1.075126     1.105578     1.233714     1.273330     1.038651
##    prep_days         home         dome      KICKOFF         PUNT   random_var
##     1.030074     1.028967     1.023308     2.380244     2.242410     1.004033
```

3rd down "will the play convert to a 1st down?"

```
# Fit OLS model
OLS_3rd <- lm(converted ~ ., data = model_3rd3)
summary(OLS_3rd)
```

```
##
## Call:
## lm(formula = converted ~ ., data = model_3rd3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7210 -0.4001 -0.2135  0.4839  1.2515
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.3622769  0.0334614  10.827  < 2e-16 ***
## down1_pct     0.0062776  0.0060435   1.039  0.29896
## down2_pct    -0.0052631  0.0063519  -0.829  0.40737
## down3_pct     0.0084082  0.0059920   1.403  0.16059
## opp_scss      0.0003513  0.0056698   0.062  0.95060
```

```
## opp_fails    -0.0018679  0.0055901  -0.334  0.73828
## score_diff   -0.0006671  0.0058512  -0.114  0.90923
## min_rem       0.0069052  0.0059443   1.162  0.24542
## ydstogo      -0.1516121  0.0058256 -26.025  < 2e-16 ***
## yardline_100  0.0185814  0.0057252   3.246  0.00118 **
## rush          0.0810665  0.0136165   5.954 2.75e-09 ***
## offtimes      0.0064112  0.0085715   0.748  0.45450
## deftimes      0.0010178  0.0088982   0.114  0.90894
## week         -0.0010464  0.0010644  -0.983  0.32560
## prep_days    -0.0074347  0.0055880  -1.330  0.18340
## home          0.0059439  0.0111604   0.533  0.59434
## dome          0.0068584  0.0143262   0.479  0.63214
## KICKOFF       0.0156793  0.0170950   0.917  0.35908
## PUNT         -0.0103443  0.0170304  -0.607  0.54360
## random_var    0.0024869  0.0055171   0.451  0.65218
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.462 on 7002 degrees of freedom
## Multiple R-squared:  0.1113, Adjusted R-squared:  0.1089
## F-statistic: 46.15 on 19 and 7002 DF,  p-value: < 2.2e-16
```

```
vif(OLS_3rd)
```

```
##    down1_pct    down2_pct    down3_pct     opp_scss    opp_fails   score_diff
##     1.201427     1.327195     1.181037     1.057431     1.027906     1.126175
##      min_rem      ydstogo yardline_100         rush     offtimes     deftimes
##     1.162294     1.116364     1.078211     1.096974     1.169818     1.205572
##         week    prep_days         home         dome      KICKOFF         PUNT
##     1.037229     1.027151     1.024274     1.021051     2.391598     2.247289
##   random_var
##     1.001233
```

Random Forest (i will do some type of boosting later as there are definitly non linearities in the data)

4th down "will the coach attempt to go for it?"

```
# Train Random Forest
rf4 <- randomForest(as.factor(attempt) ~ .,
                    data = model_4th3,
                    importance = TRUE,  # Calculate both MDI and MDA
                    ntree = 1500)

# Get variable importance measures
importance_df <- importance(rf4) %>%
  as.data.frame() %>%
  rownames_to_column("Variable") %>%
  arrange(desc(MeanDecreaseAccuracy))
```

```r
# Calculate OOB AUC
oob_pred <- predict(rf4, type = "prob")[,2]
oob_auc <- auc(model_4th3$attempt, oob_pred)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
# Print results
print("Variable Importance (sorted by Mean Decrease Accuracy):")
```

```
## [1] "Variable Importance (sorted by Mean Decrease Accuracy):"
```

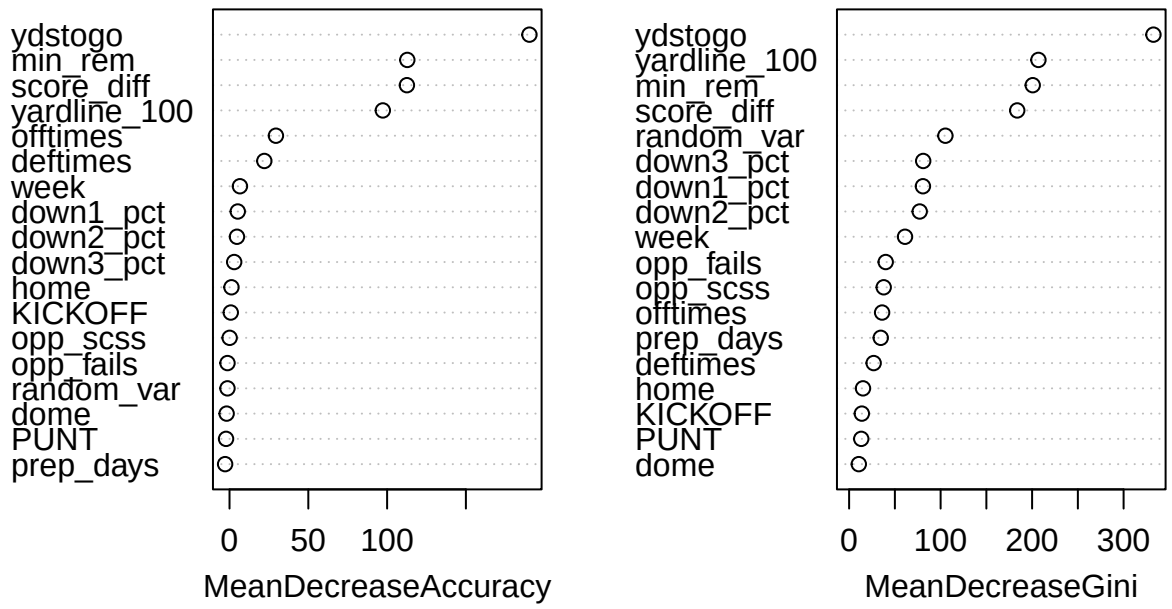
```r
print(importance_df)
```

```
##        Variable            0           1 MeanDecreaseAccuracy MeanDecreaseGini
## 1       ydstogo 126.17631625 180.9269565          190.3083195        332.60430
## 2       min_rem  65.94363477 106.5847706          112.8129772        200.68140
## 3    score_diff  60.51933141 111.0943481          112.5648033        183.72028
## 4  yardline_100  50.16170221 104.3865726           97.2745206        206.90040
## 5      offtimes  19.88262754  22.9882154           29.4434448         36.05632
## 6      deftimes  22.79042316   2.6332014           22.0888674         26.81589
## 7          week   0.03098718  11.3460291            6.6156613         61.07585
## 8     down1_pct   0.42812868   8.6462555            5.2355853         80.65640
## 9     down2_pct   2.33487550   4.7864976            4.7187556         77.14416
## 10    down3_pct  -2.45809956   8.9714101            2.9547354         80.95604
## 11         home  -2.59465522   5.7055116            1.2423519         15.15534
## 12      KICKOFF   2.05504900  -1.6499915            0.8015327         13.89925
## 13     opp_scss  -3.34722226   5.3058397            0.0800720         37.78732
## 14    opp_fails  -5.56149343   6.1242880           -1.2828882         39.96522
## 15   random_var  -2.40776905   1.1594118           -1.2958161        105.20990
## 16         dome  -1.64456856  -0.8195412           -1.8305850         10.55843
## 17         PUNT  -3.68464118   1.5831185           -2.1976977         13.24278
## 18    prep_days  -3.78860584   0.7236162           -2.7828429         34.54754
```

```r
print(paste("OOB AUC:", round(oob_auc, 3)))
```

```
## [1] "OOB AUC: 0.88"
```

```r
# Plot variable importance
varImpPlot(rf4,
           sort = TRUE,
           main = "Variable Importance Plot",
           n.var = min(20, ncol(model_4th3)-1))
```




```
# Calculate ROC object from existing predictions
roc_4th <- roc(model_4th3$attempt, oob_pred)
```

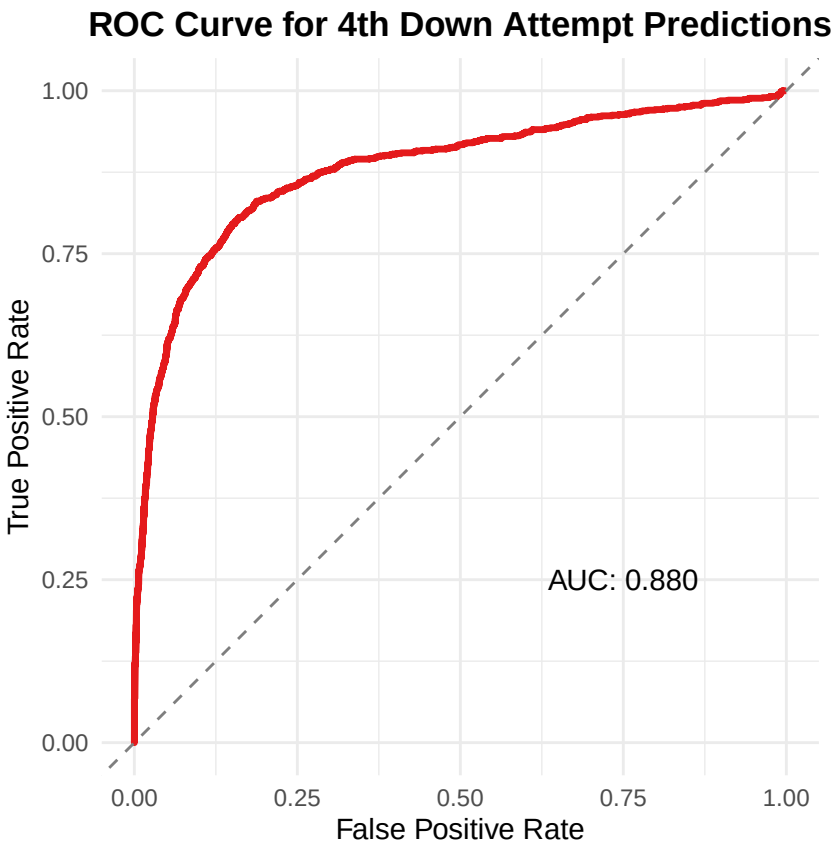
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
# Create data frame for plotting
roc_df_4th <- data.frame(
  FPR = 1 - roc_4th$specificities,
  TPR = roc_4th$sensitivities
)

# Create the plot
ggplot(roc_df_4th, aes(x = FPR, y = TPR)) +
  geom_line(size = 1.2, color = "#E41A1C") +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "gray50") +
  annotate("text", x = 0.75, y = 0.25,
           label = sprintf("AUC: %.3f", oob_auc)) +
  labs(title = "ROC Curve for 4th Down Attempt Predictions",
       x = "False Positive Rate",
       y = "True Positive Rate") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold")
```

```
  ) +
  coord_equal()
```

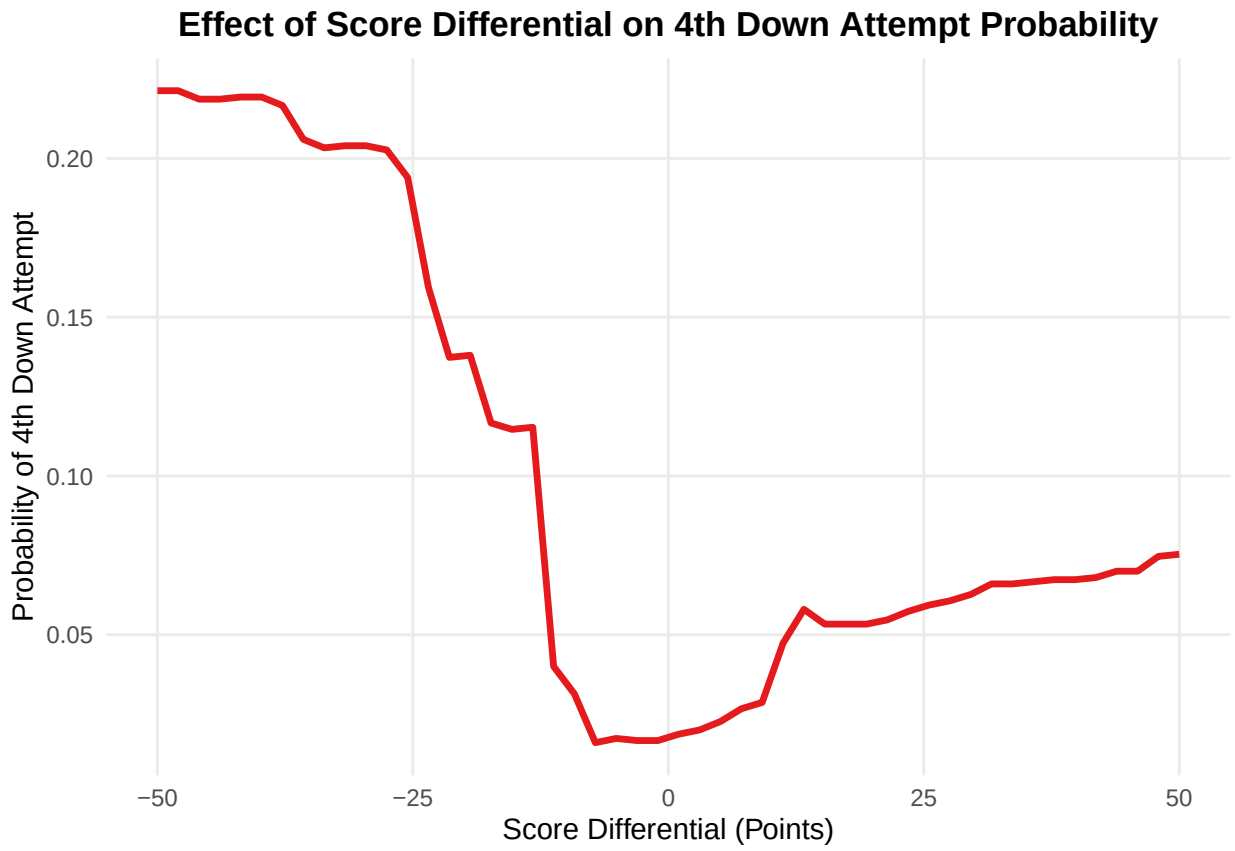
```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```



Note that these partial dependency plots tell a story more about how the RF model uses the variable. It could still be used wrong/given too much weight by the model.

```
# Get scaling attributes for score_diff from the original data FIRST
score_diff_center <- attr(model_4th3$score_diff, "scaled:center")
score_diff_scale <- attr(model_4th3$score_diff, "scaled:scale")

# Create a grid of score_diff values
grid_points <- seq(min(model_4th3$score_diff), max(model_4th3$score_diff), length.out = 50)

# Create prediction data frame
pred_data <- model_4th3[rep(1, length(grid_points)), ]
pred_data$score_diff <- grid_points

# Get predictions
predictions <- predict(rf4, pred_data, type = "prob")[,2]
```

```r
# Create plot data
plot_data <- data.frame(
  score_diff = grid_points * score_diff_scale + score_diff_center,  # Convert back to original scale
  probability = predictions
)

# Create plot
ggplot(plot_data, aes(x = score_diff, y = probability)) +
  geom_line(color = "#E41A1C", size = 1.2) +
  labs(
    title = "Effect of Score Differential on 4th Down Attempt Probability",
    x = "Score Differential (Points)",
    y = "Probability of 4th Down Attempt"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    panel.grid.minor = element_blank()
  )
```



Note that these partial dependency plots tell a story more about how the RF model uses the variable. It could still be used wrong/given to much weight by the model.

This is a very good AUC dispite missing how good the teams are. This makes sense though since NFL teams are trying to make the right decision.

3rd down "will the play convert to a 1st down?"

```
# Train Random Forest
rf3 <- randomForest(as.factor(converted) ~ .,
                    data = model_3rd3,
                    importance = TRUE,  # Calculate both MDI and MDA
                    ntree = 1500)

# Get variable importance measures
importance_df <- importance(rf3) %>%
  as.data.frame() %>%
  rownames_to_column("Variable") %>%
  arrange(desc(MeanDecreaseAccuracy))

# Calculate OOB AUC
oob_pred <- predict(rf3, type = "prob")[,2]
oob_auc <- auc(model_3rd3$converted, oob_pred)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
# Print results
print("Variable Importance (sorted by Mean Decrease Accuracy):")
```

```
## [1] "Variable Importance (sorted by Mean Decrease Accuracy):"
```

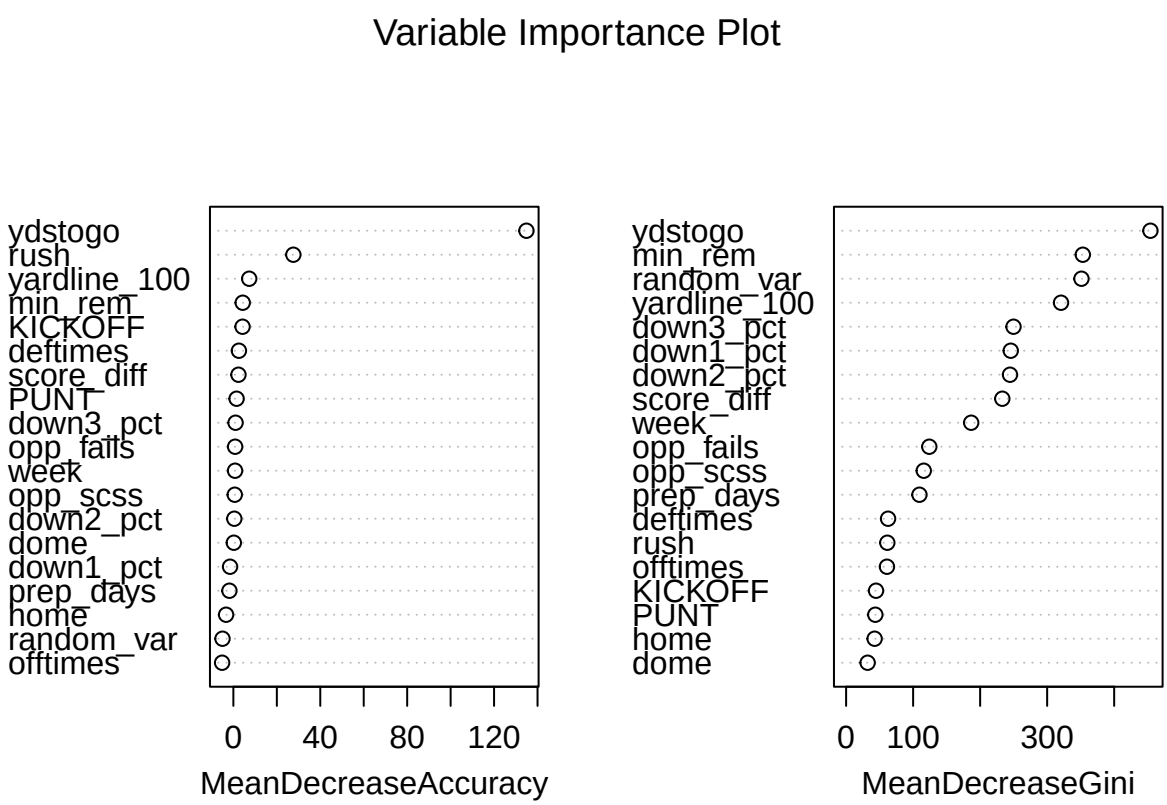
```
print(importance_df)
```

```
##        Variable          0           1 MeanDecreaseAccuracy MeanDecreaseGini
## 1       ydstogo 83.8317157 113.3953462          134.8941181        454.12430
## 2          rush  6.0331399  29.1659960           27.6009120         61.48460
## 3  yardline_100 13.4272333  -5.4120204            7.2849001        320.70320
## 4       min_rem  4.1929055   1.4608068            4.2965789        353.20318
## 5       KICKOFF -1.3613237   7.3698471            4.2205141         44.62363
## 6      deftimes  2.8137233   0.4715451            2.5519175         62.58283
## 7    score_diff  1.6180197   1.5418142            2.3357744        233.08673
## 8          PUNT -0.8853559   3.2450684            1.4851308         43.54321
## 9     down3_pct  3.2618783  -2.4079385            1.0078558        249.78640
## 10    opp_fails -1.4481913   2.9407760            0.8012161        124.09046
## 11         week -0.1777957   1.3953460            0.7732938        186.64045
## 12     opp_scss -1.3835820   2.5356622            0.6327623        115.84639
## 13    down2_pct -0.9685220   1.7745029            0.4199519        244.59013
## 14         dome -1.4001511   2.0311682            0.2020687         32.09727
## 15    down1_pct -4.9484584   3.7109929           -1.5320837        245.67208
## 16    prep_days -0.9468553  -1.6394586           -1.8155464        109.40753
## 17         home -2.7670202  -1.8286736           -3.3569124         42.51158
## 18   random_var -1.9955416  -5.3715894           -5.0160631        351.19421
## 19     offtimes -3.6036979  -3.4318202           -5.1787098         60.91589
```

```r
print(paste("OOB AUC:", round(oob_auc, 3)))
```

```
## [1] "OOB AUC: 0.668"
```

```r
# Plot variable importance
varImpPlot(rf3,
           sort = TRUE,
           main = "Variable Importance Plot",
           n.var = min(20, ncol(model_3rd3)-1))
```



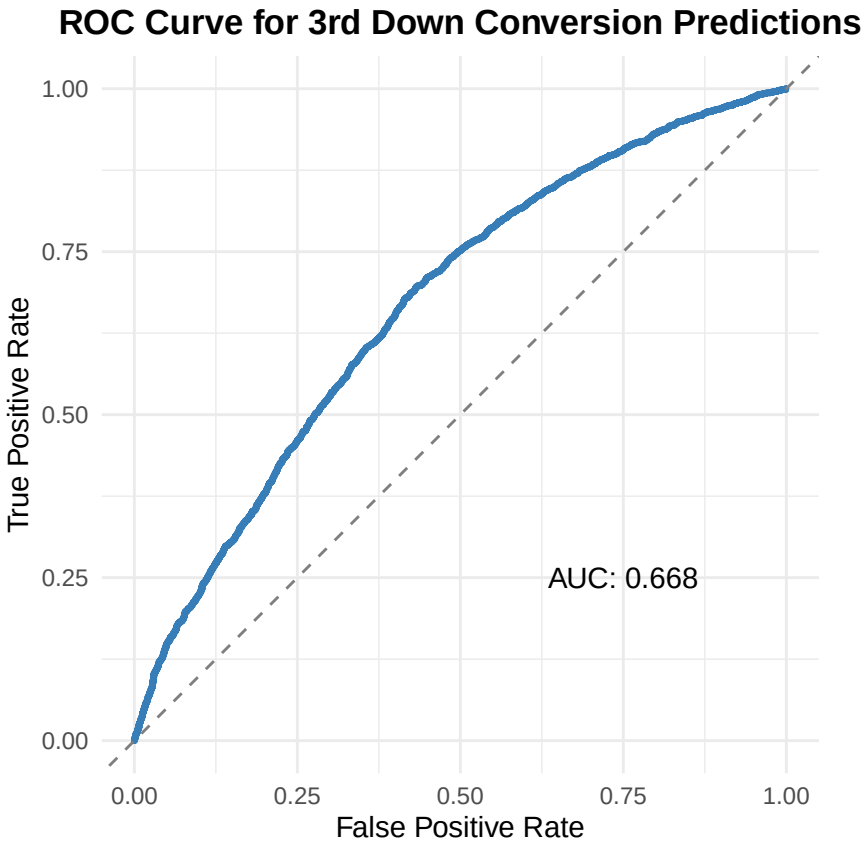

```r
# Calculate ROC object from existing predictions
roc_3rd <- roc(model_3rd3$converted, oob_pred)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```r
# Create data frame for plotting
roc_df_3rd <- data.frame(
  FPR = 1 - roc_3rd$specificities,
  TPR = roc_3rd$sensitivities
)
```

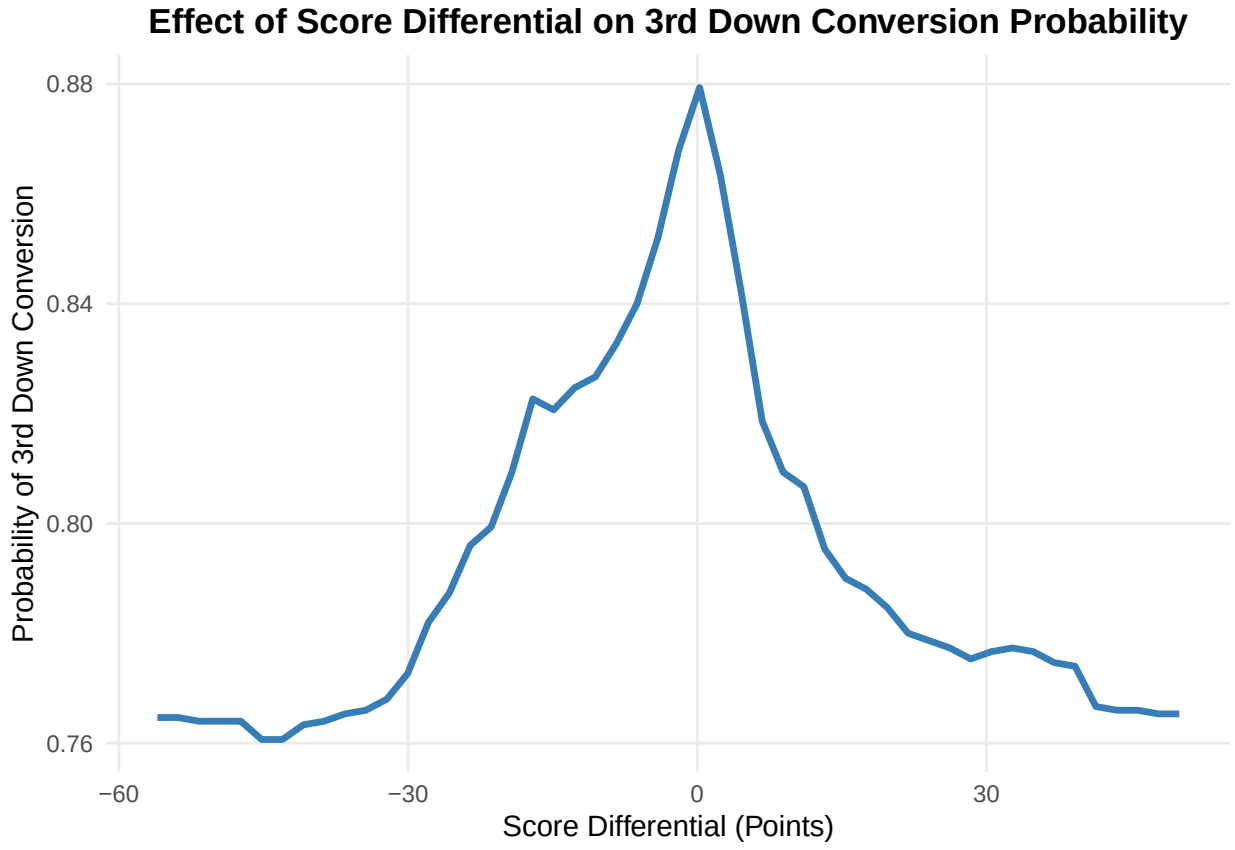
```r
# Create the plot
ggplot(roc_df_3rd, aes(x = FPR, y = TPR)) +
  geom_line(size = 1.2, color = "#377EB8") +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "gray50") +
  annotate("text", x = 0.75, y = 0.25,
           label = sprintf("AUC: %.3f", oob_auc)) +
  labs(title = "ROC Curve for 3rd Down Conversion Predictions",
       x = "False Positive Rate",
       y = "True Positive Rate") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold")
  ) +
  coord_equal()
```



Note that these partial dependency plots tell a story more about how the RF model uses the variable. It could still be used wrong/given to much weight by the model.

```r
# Get scaling attributes for score_diff from the original data FIRST
score_diff_center <- attr(model_3rd3$score_diff, "scaled:center")
score_diff_scale <- attr(model_3rd3$score_diff, "scaled:scale")

# Create a grid of score_diff values
grid_points <- seq(min(model_3rd3$score_diff), max(model_3rd3$score_diff), length.out = 50)

# Create prediction data frame
```

```r
pred_data <- model_3rd3[rep(1, length(grid_points)), ]
pred_data$score_diff <- grid_points

# Get predictions
predictions <- predict(rf3, pred_data, type = "prob")[,2]

# Create plot data
plot_data <- data.frame(
 score_diff = grid_points * score_diff_scale + score_diff_center,  # Convert back to original scale
 probability = predictions
)

# Create plot
ggplot(plot_data, aes(x = score_diff, y = probability)) +
 geom_line(color = "#377EB8", size = 1.2) +
 labs(
   title = "Effect of Score Differential on 3rd Down Conversion Probability",
   x = "Score Differential (Points)",
   y = "Probability of 3rd Down Conversion"
 ) +
 theme_minimal() +
 theme(
   plot.title = element_text(hjust = 0.5, face = "bold"),
   panel.grid.minor = element_blank()
 )
```



This is a bad AUC which makes sense. We don't have how good the teams are here.