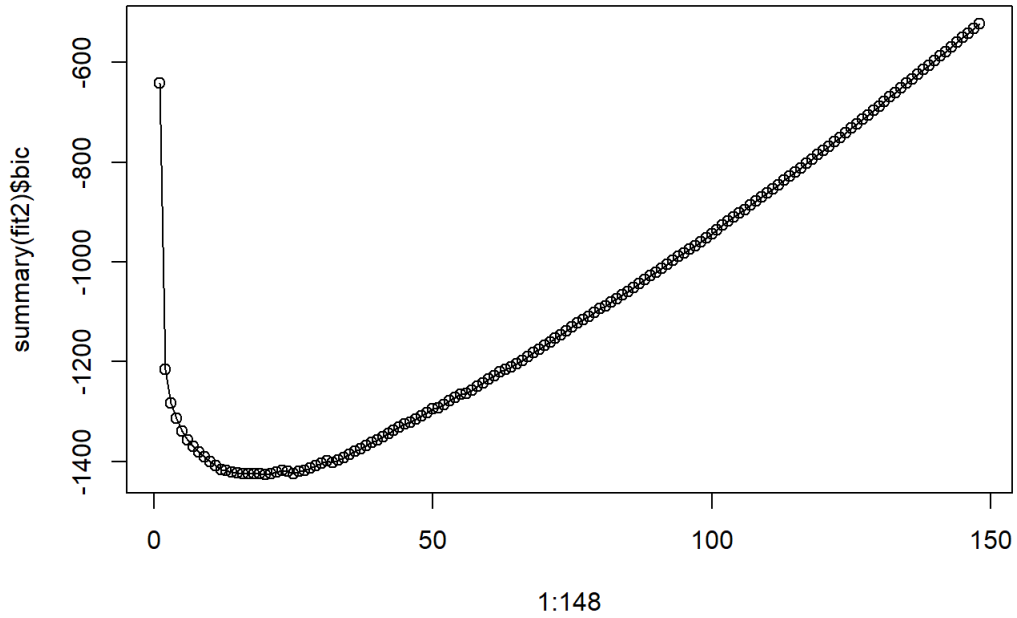
Final Presentation

Gregory Chang, Indoo Park, Seulchan Kim

12/12/2019

Cleaning and Transformation of Training Data

From the start, we chose to omit the categorical variables, “id”, “gameID”, “VT”, “HT”, “VTleague”, and “HTleague”. Since these were predictors with either unique team identifiers or character strings, we decided to remove them from our training dataset to prevent potential errors that may arise from applying a model that does not accept categorical variables. We would later run models with the variables “VTleague” and “HTleague”, however these attempts did not yield better results. From here, our group ran a forward and backwards stepwise to see which predictors we should include in our model. We used the judged each model based on the BIC value and concluded that we should use a 20 predictor model with the following predictors:




```
##  [1] "VT.OTS.fgm"   "VT.OTA.ast"   "VT.S1.pts"    "VT.S5.stl"    "VT.OS2.plmin"
##  [6] "VT.OS3.fgm"   "VT.OS4.dreb"  "HT.S3.pts"    "HT.S5.ast"    "HT.OS1.fgm"
## [11] "HT.TS.fta"    "HT.TS.to"     "HT.TS.pf"     "HT.TA.ast"    "HT.TA.stl"
## [16] "HT.OTS.blk"   "HT.OTA.fga"   "HT.OTA.dreb"  "HT.OTA.ast"   "HT.OTA.to"
```

Description of our Models

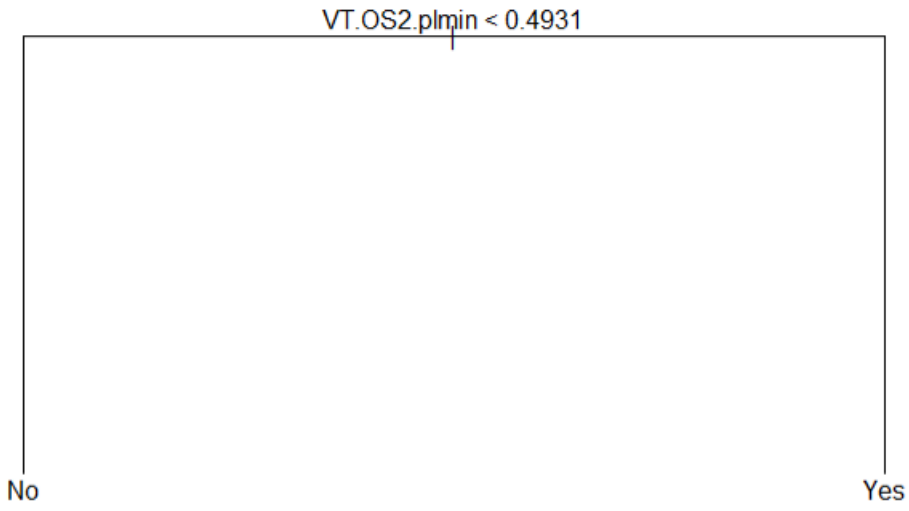
Using the 20 predictors, we then attempted to apply various modeling methods to see if we could keep improving our classification rate. In the end, our best models were a gbm model and a tree model.

From our glm model, we get the following summary output:

```
Call:
glm(formula = HTWins ~ VT.TS.fgm + VT.TS.pts + VT.TA.pts + VT.OTS.fgm +
    VT.OTA.ast + VT.S1.pts + VT.S5.stl + VT.OS2.plmin + VT.OS3.fgm +
    VT.OS4.dreb + HT.S3.pts + HT.S5.stl + HT.S5.ast + HT.OS1.fgm +
    HT.TS.fta + HT.TS.to + HT.TS.pf + HT.TA.ast + HT.TA.stl +
    HT.OTS.fgm + HT.OTS.blk + HT.OTA.fga + HT.OTA.dreb + HT.OTA.ast +
    HT.OTA.to, family = "binomial", data = train1)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.393  -1.092   0.614   0.956   2.108

Coefficients: (2 not defined because of singularities)
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.161984   0.910348   0.178 0.858773
VT.TS.fgm    -0.030507   0.012128  -2.516 0.011886 *
VT.TS.pts    -0.093964   0.008877 -10.585  < 2e-16 ***
VT.TA.pts     0.089957   0.007835  11.482  < 2e-16 ***
VT.OTS.fgm    0.074973   0.010241   7.321 2.46e-13 ***
VT.OTA.ast   -0.164143   0.015428 -10.640  < 2e-16 ***
VT.S1.pts    -0.037509   0.008659  -4.332 1.48e-05 ***
VT.S5.stl    -0.314158   0.067129  -4.680 2.87e-06 ***
VT.OS2.plmin  0.116478   0.009470  12.299  < 2e-16 ***
VT.OS3.fgm   -0.104049   0.023810  -4.370 1.24e-05 ***
VT.OS4.dreb   0.059164   0.016059   3.684 0.000229 ***
HT.S3.pts     0.059980   0.010990   5.458 4.82e-08 ***
HT.S5.stl     0.024443   0.075456   0.324 0.745990
HT.S5.ast    -0.014984   0.025419  -0.589 0.555533
HT.OS1.fgm    0.038832   0.020550   1.890 0.058809 .
HT.TS.fta     0.067997   0.012518   5.432 5.57e-08 ***
HT.TS.to     -0.099267   0.035477  -2.798 0.005141 **
HT.TS.pf     -0.004571   0.008950  -0.511 0.609551
HT.TA.ast           NA         NA      NA       NA
HT.TA.stl     0.041755   0.030058   1.389 0.164794
HT.OTS.fgm          NA         NA      NA       NA
HT.OTS.blk   -0.071530   0.039397  -1.816 0.069426 .
HT.OTA.fga    0.006449   0.011312   0.570 0.568619
HT.OTA.dreb   0.011054   0.015196   0.727 0.466953
HT.OTA.ast    0.068860   0.017725   3.885 0.000102 ***
HT.OTA.to     0.011647   0.016479   0.707 0.479716
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 12863  on 9519  degrees of freedom
Residual deviance: 11427  on 9496  degrees of freedom
AIC: 11475

Number of Fisher Scoring iterations: 3
```

We see that the model did not mark all of our 20 predictors as significant with two predictors HT.TA.ast and HT.OTS.fgm having NA values across the board. In an attempt to clean our model further, we removed all non-significant variables, however this actually decreased our classification rate.

For our tree model, our plot looked as follows:



Despite us using 20 predictor variables, our tree fit only uses one predictor, “VT.OS2.plmin”, to draw the split. Despite its simplicity, our tree model matched our previous glm model’s classification rate. With this in mind, our group chose to move forward with the tree model due to

its easy interpretability and simplicity. In the end, however, our group was not able to improve our model further.

Classification Rate with this Model

```
For this model, we got a 0.6711 classification rate on the public leaderboard.
```

Why this Model Worked

Of all the other methods our group tried, tree gave us the best results as well as the most interpretable model due to its simplicity. However, One tree model most likely had too much variability, therefore, we should have used a random forest or a boosting method in order to hopefully increase the accuracy of our model. Allowing more trees in the forest to weigh in and make a group decision would decrease the variability of our model. Boosting could have given us a better prediction for our testing data as it builds a forest of trees with each dependent on the previous tree. This would allow our subsequent trees to learn to predict outcomes.

Other Models We Tried