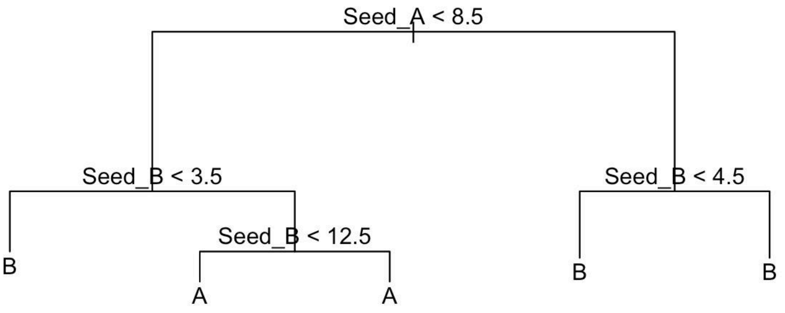
STAT 4340 Project 1 Report

Silas Kluever, Johnny Ferrara, Eliot Saphian, and Garrett Rollett

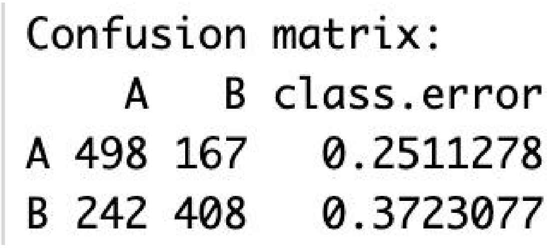
Model Building

To determine a best predicting model we planned to make three separate models and choose the best one based on misclassification rate; the three models being GLM, classification trees, and random forest. For the process of making our Logistic regression model, we had to first determine which predictor variables would be most significant in predicting a win or loss. To do this we used the leaps package to find a best subset for the predictor to determine a win or loss based on the criterion of adjusted r-squared. Through this process we determined the best GLM would be one with the predictor variables of FGM_A, FTA_B, DR_A, DR_B, Ast_A, Stl_B, Blk_A, Seed_A, and Seed_B. This model gives us a misclassification rate of 27.98%.

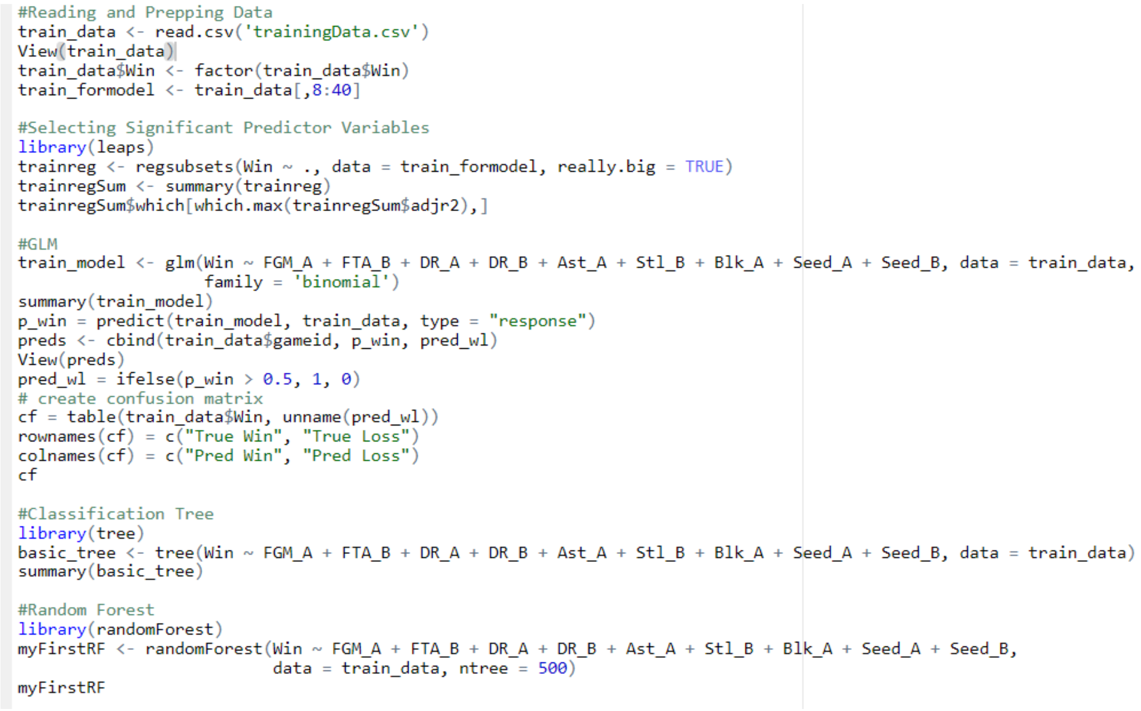
Next, we aimed to make a classification tree model. We started with a tree that contained all the predictor variables we used in our final GLM model and pruned it to find a best classification tree model. This final pruned model had a misclassification rate of 30.87%



Our last task for the model building process was to make a random forest model; Once again, we used the same variables from our GLM function. This model gave us an OOB error rate of 32.24%



```
Confusion matrix:
    A   B class.error
A 498 167   0.2511278
B 242 408   0.3723077
```



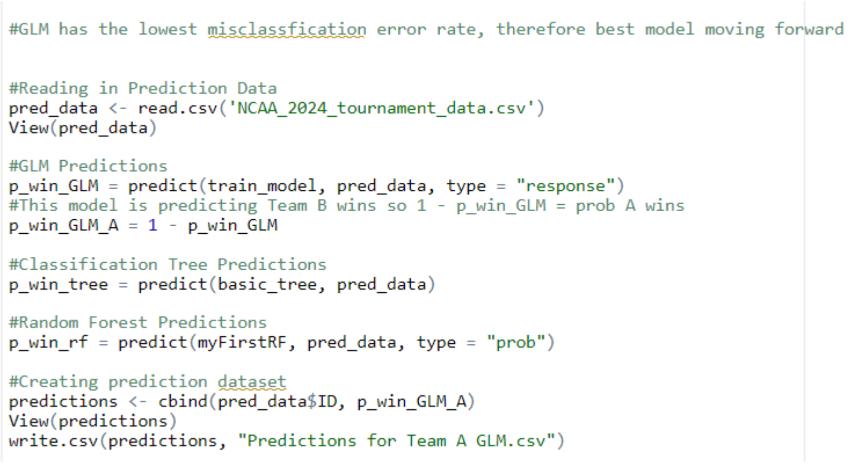
```r
#Reading and Prepping Data
train_data <- read.csv('trainingData.csv')
View(train_data)
train_data$Win <- factor(train_data$Win)
train_formodel <- train_data[,8:40]

#Selecting Significant Predictor Variables
library(leaps)
trainreg <- regsubsets(Win ~ ., data = train_formodel, really.big = TRUE)
trainregSum <- summary(trainreg)
trainregSum$which[which.max(trainregSum$adjr2),]

#GLM
train_model <- glm(Win ~ FGM_A + FTA_B + DR_A + DR_B + Ast_A + Stl_B + Blk_A + Seed_A + Seed_B, data = train_data,
                   family = 'binomial')
summary(train_model)
p_win = predict(train_model, train_data, type = "response")
preds <- cbind(train_data$gameid, p_win, pred_wl)
View(preds)
pred_wl = ifelse(p_win > 0.5, 1, 0)
# create confusion matrix
cf = table(train_data$Win, unname(pred_wl))
rownames(cf) = c("True Win", "True Loss")
colnames(cf) = c("Pred Win", "Pred Loss")
cf

#Classification Tree
library(tree)
basic_tree <- tree(Win ~ FGM_A + FTA_B + DR_A + DR_B + Ast_A + Stl_B + Blk_A + Seed_A + Seed_B, data = train_data)
summary(basic_tree)

#Random Forest
library(randomForest)
myFirstRF <- randomForest(Win ~ FGM_A + FTA_B + DR_A + DR_B + Ast_A + Stl_B + Blk_A + Seed_A + Seed_B,
                          data = train_data, ntree = 500)
myFirstRF
```



```r
#GLM has the lowest misclassfication error rate, therefore best model moving forward


#Reading in Prediction Data
pred_data <- read.csv('NCAA_2024_tournament_data.csv')
View(pred_data)

#GLM Predictions
p_win_GLM = predict(train_model, pred_data, type = "response")
#This model is predicting Team B wins so 1 - p_win_GLM = prob A wins
p_win_GLM_A = 1 - p_win_GLM

#Classification Tree Predictions
p_win_tree = predict(basic_tree, pred_data)

#Random Forest Predictions
p_win_rf = predict(myFirstRF, pred_data, type = "prob")

#Creating prediction dataset
predictions <- cbind(pred_data$ID, p_win_GLM_A)
View(predictions)
write.csv(predictions, "Predictions for Team A GLM.csv")
```

Comparative Results

Generalized Linear Model (GLM)

Model Fit: The GLM was fitted using the binomial family to predict the binary outcome of win/loss. Significant predictors include FGM_A, FTA_B, DR_A, DR_B, Ast_A, Stl_B, Blk_A, Seed_A, and Seed_B. Highlighting the importance of both in-game performance and pre-tournament seeding in determining game outcomes.

Coefficients: The signs of the coefficients provide insights into their relationship with the probability of Team A winning. For example, a positive coefficient for DR_A suggests that higher defensive rebounds for Team A increase their chances of winning.
Model Evaluation: The model's performance was evaluated using a confusion matrix, resulting in 485 true wins and 462 true losses correctly identified, with misclassifications of 180 false losses and 188 false wins. Based on this model our misclassification rate was 27.98%. Resulting in this model correctly predicting the outcome 72.02% of the time.

Classification Tree

Model Summary: The tree model primarily used the seeds of teams Seed_A and Seed_B to make predictions, reflecting the significant impact of seeding on game outcomes. The number of terminal nodes was 5 and the residual deviance was 1.16.
Performance: The misclassification error rate was 30.87%, slightly lower than the random forest but not as good as might be hoped for predictive accuracy. This means our model makes the correct prediction in 69.13% of the games.

Random Forest

Model Overview: With 500 trees, the random forest model was used to improve prediction accuracy by averaging multiple decision trees, each built on a subset of the data and features.
Out-of-Bag (OOB) Error: The OOB error rate was 32.24%, with the confusion matrix indicating 481 true wins and 409 true losses correctly identified, with misclassifications of 241 false losses and 184 false wins. The error for Seed_A is 0.2766 and for Seed_B is 0.3708.

Final Model Choice

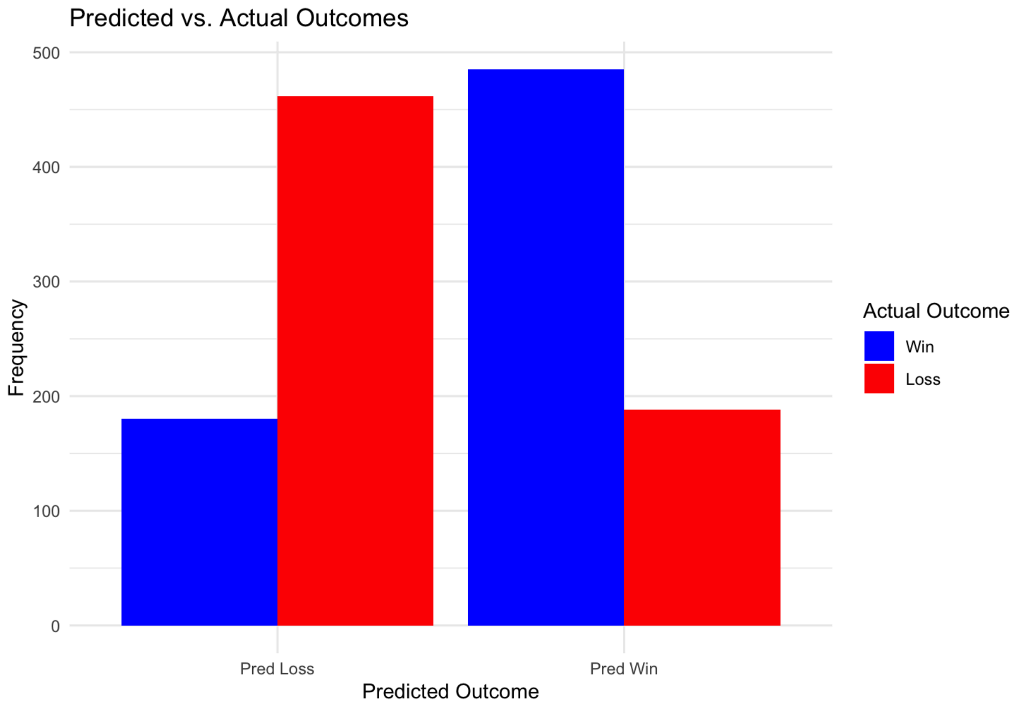
Based on the lowest misclassification error rate, we can conclude the GLM is the best predictive model going forward.

Final Model Evaluation (GLM)

- Confusion matrix:

	Pred Win	Pred Loss
True Win	485	180
True Loss	188	462

- Misclassification error rate: (188 + 180)/(485+180+188+462) = 0.2798479
- Confusion Matrix Visualization:



As shown above, the misclassification error rate of our predictive model is calculated to be approximately 0.2798 or 27.98%. This signifies the proportion of incorrect predictions made by the model across both classes (wins and losses). From the provided visual and confusion matrix, we can see the model falsely predicted 188 instances of true losses as wins and 180 instances of true wins as losses. While it correctly predicted 485 wins and 462 losses. Although a lower misclassification error rate indicates better predictive performance, due to the intricate and volatile nature of college basketball we can conclude this model serves as a fairly accurate predictor of March Madness tournament games.