US Research University Prediction Model

Philip Gabriel Andrada

November 02, 2016

# Preparation

# loading necessary libraries  
library(rpart)  
library(randomForest)

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

library(tree)  
library(party)

## Loading required package: grid

## Loading required package: mvtnorm

## Loading required package: modeltools

## Loading required package: stats4

## Loading required package: strucchange

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

## Loading required package: sandwich

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

##   
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':  
##   
## margin

library(Boruta)

## Loading required package: ranger

##   
## Attaching package: 'ranger'

## The following object is masked from 'package:randomForest':  
##   
## importance

library(e1071)  
library(ROCR)

## Loading required package: gplots

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

library(corrplot)  
library(ggplot2)

#Reading Data Files  
usuniv2010 <- read.csv("C:\\Users\\Philip\\Desktop\\Capstone\\MERGED2010\_11\_PP.csv")  
usuniv2011 <- read.csv("C:\\Users\\Philip\\Desktop\\Capstone\\MERGED2011\_12\_PP.csv")  
usuniv2012 <- read.csv("C:\\Users\\Philip\\Desktop\\Capstone\\MERGED2012\_13\_PP.csv")  
usuniv2013 <- read.csv("C:\\Users\\Philip\\Desktop\\Capstone\\MERGED2013\_14\_PP.csv")  
usuniv2014 <- read.csv("C:\\Users\\Philip\\Desktop\\Capstone\\MERGED2014\_15\_PP.csv")  
  
#Binding All Data Files into One Data Frame  
usuniv <- rbind(usuniv2010,usuniv2011,usuniv2012,usuniv2013,usuniv2014)

## Warning in `[<-.factor`(`\*tmp\*`, ri, value = c(100200L, 105200L,  
## 2503400L, : invalid factor level, NA generated  
  
## Warning in `[<-.factor`(`\*tmp\*`, ri, value = c(100200L, 105200L,  
## 2503400L, : invalid factor level, NA generated  
  
## Warning in `[<-.factor`(`\*tmp\*`, ri, value = c(100200L, 105200L,  
## 2503400L, : invalid factor level, NA generated  
  
## Warning in `[<-.factor`(`\*tmp\*`, ri, value = c(100200L, 105200L,  
## 2503400L, : invalid factor level, NA generated  
  
## Warning in `[<-.factor`(`\*tmp\*`, ri, value = c(100200L, 105200L,  
## 2503400L, : invalid factor level, NA generated  
  
## Warning in `[<-.factor`(`\*tmp\*`, ri, value = c(100200L, 105200L,  
## 2503400L, : invalid factor level, NA generated

#Since there are some incomplete Carnegie Classifications, we use usuniv2014 as basis for the classification for the rest  
usuniv$CCBASIC2 <- usuniv2014$CCBASIC[match(usuniv$OPEID6,usuniv2014$OPEID6)]  
  
#added the ACCEPTED column for those that are research universities (CCBASIC2 is equal to 15 or 16), as our focus will be on these  
usuniv$ACCEPTED <- ifelse(usuniv$CCBASIC2 %in% c(15,16), 1, 0)  
  
#Create a vector with the columns that is needed from the study  
# 19 - institution region (1-New England, 2-Mid East, 3-Great Lakes, 4-Plains, 5-Southeast, 6-Southwest, 7-Rocky Mountains, 8-Far West, 9-Outlying Areas)  
# 37-38 - admission rate  
# 39-61 - SAT and ACT Scores  
# 62-99 - percentage of degrees awarded for each field of study  
# 293-299 - total share of enrollment for different ethnicities  
# 300 - total share of enrollment that are non-resident aliens (i.e. international students)  
# 301 - total share of enrollment that have unknown race  
# 314 - share of undergraduate, degree-/certificate-seeking students who are part-time  
# 377 - average cost of attendance in an academic year institution  
# 379 - in-state tuition and fees  
# 380 - out-of-state tuition and fees  
# 387 - completion rate of first-time, full-time students at four-year institutions with 150% of expected time to completion)  
# 397-403 - completion rate for first-time, full-time students for different ethnicities  
# 404 - completion rate for first-time, full-time students for non-resident aliens  
# 405 - completion rate for first-time, full-time students that have unknown race  
# 429 - retention rate for first-time, full time studnets at four-year institutions  
# 438 - percent of all federal undergraduate students receiving a federal student loan  
# 1412 - percentage of first-generation students  
# 1740-1741 - total share of enrollment per gender  
# 1745 - acceptance flag  
col\_select <- c(19,37:38,61:99,293:301,314,377,379:380,387,397:405,429,438,1412,1740:1741, 1744, 1745)  
  
# Create a new data frame with the columns that will be filtered out  
usunivfilter <- usuniv[,col\_select]  
  
# Change the factor columns to numeric for faster processing  
for (i in 1:ncol(usunivfilter)){  
 usunivfilter[,i] <- as.numeric(as.character(usunivfilter[,i]))  
}

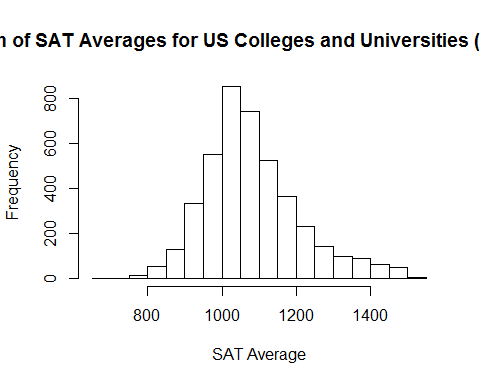
## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion

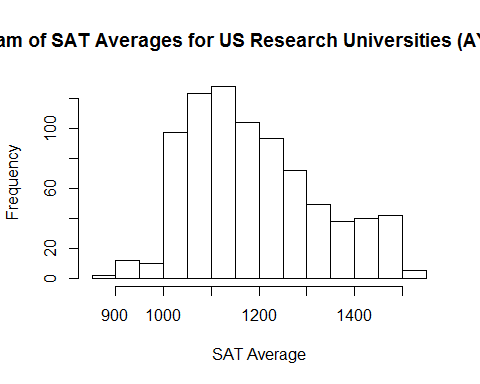
# Clean the results to have all complete   
usunivfilter <- usunivfilter[!is.na(usunivfilter$C150\_4),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$C150\_4\_ASIAN),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$C150\_4\_WHITE),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$C150\_4\_BLACK),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$C150\_4\_NRA),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$ADM\_RATE\_ALL),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$SAT\_AVG\_ALL),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$UGDS\_ASIAN),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$UGDS\_WHITE),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$UGDS\_BLACK),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$UGDS\_NRA),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$UGDS\_WOMEN),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$UGDS\_MEN),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$COSTT4\_A),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP11),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP12),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP14),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP15),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP24),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP26),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP27),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP40),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP45),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP51),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCIP52),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$PCTFLOAN),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$PPTUG\_EF),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$RET\_FT4),]  
usunivfilter <- usunivfilter[!is.na(usunivfilter$PAR\_ED\_PCT\_1STGEN),]  
  
#We will create another data frame for the research universities only  
usresearchuniv <- usunivfilter[usunivfilter$CCBASIC2 %in% c(15,16),]

# Distributions and Box and Whisker Plots

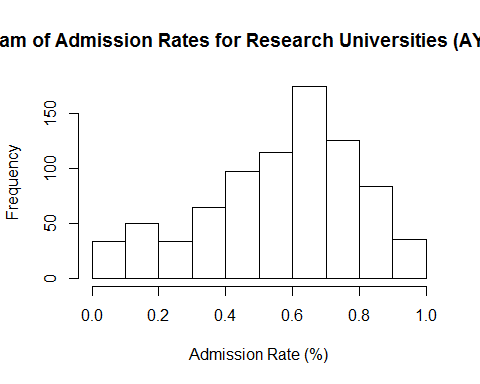
# Histogram of SAT Averages for US Colleges and Universities  
hist(usunivfilter$SAT\_AVG\_ALL, main = "Histogram of SAT Averages for US Colleges and Universities (AY2010-2015)", xlab="SAT Average")



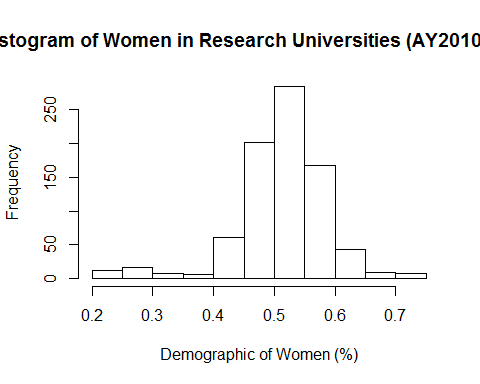
# Histogram of SAT Averages for US Research Universities  
hist(usresearchuniv$SAT\_AVG\_ALL, main = "Histogram of SAT Averages for US Research Universities (AY2010-2015)", xlab="SAT Average")



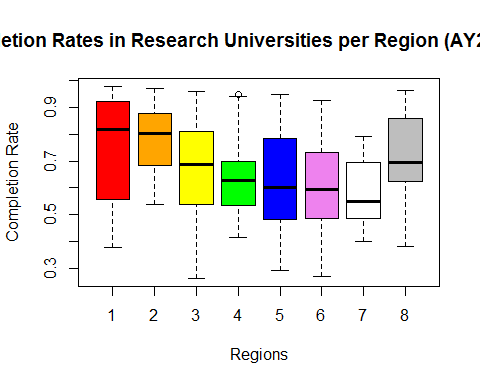
# Histogram of Admission Rates for US Research Universities  
hist(usresearchuniv$ADM\_RATE\_ALL, main = "Histogram of Admission Rates for Research Universities (AY2010-2015)", xlab = "Admission Rate (%)")



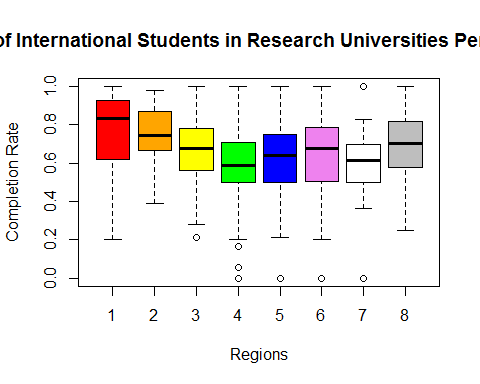
# Histogram of Women in US Research Universitie  
hist(usresearchuniv$UGDS\_WOMEN, main = "Histogram of Women in Research Universities (AY2010-2015)", xlab = "Demographic of Women (%)")



# Boxplot of Completion Rates per Region in US Research Universities  
boxplot(C150\_4 ~ REGION, usresearchuniv, main = "Completion Rates in Research Universities per Region (AY2010-2015)", col=c("red", "orange", "yellow", "green", "blue", "violet", "white", "gray", "magenta"), ylab = "Completion Rate", xlab = "Regions")

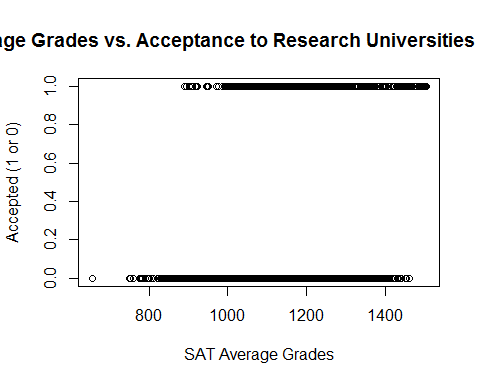


# Boxplot of COmpletion Rates of International Students per Region in US Research Universities  
boxplot(C150\_4\_NRA ~ REGION, usresearchuniv, main = "Completion Rates of International Students in Research Universities Per Region (AY2010-2015)", col=c("red", "orange", "yellow", "green", "blue", "violet", "white", "gray", "magenta"), ylab = "Completion Rate", xlab = "Regions")

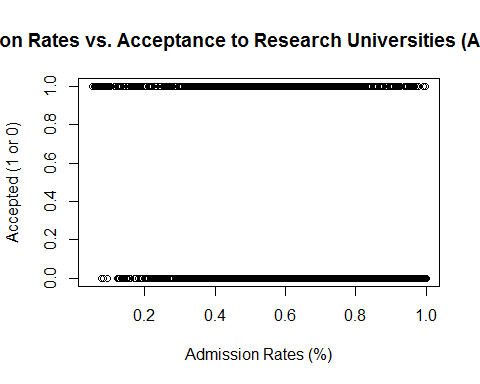


# Correlations

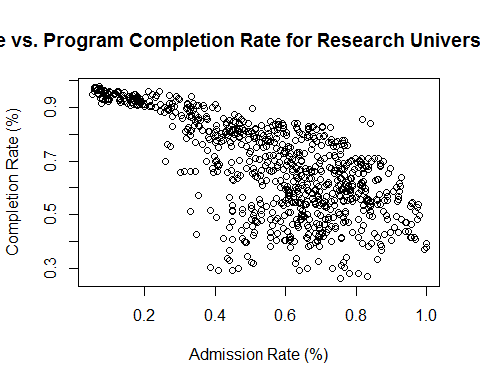
#Correlation between the SAT grades and the acceptance for the research universities  
plot(usunivfilter$SAT\_AVG\_ALL, usunivfilter$ACCEPTED, main="SAT Average Grades vs. Acceptance to Research Universities (AY2010-2015)", xlab="SAT Average Grades", ylab="Accepted (1 or 0)")



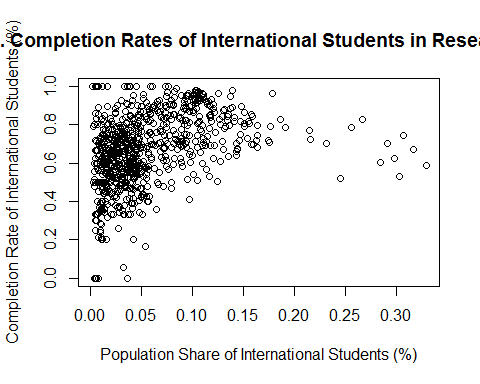
#Correlation between the admission rates and the acceptance for the research universities  
plot(usunivfilter$ADM\_RATE\_ALL, usunivfilter$ACCEPTED, main="Admission Rates vs. Acceptance to Research Universities (AY2010-2015)", xlab="Admission Rates (%)", ylab="Accepted (1 or 0)")



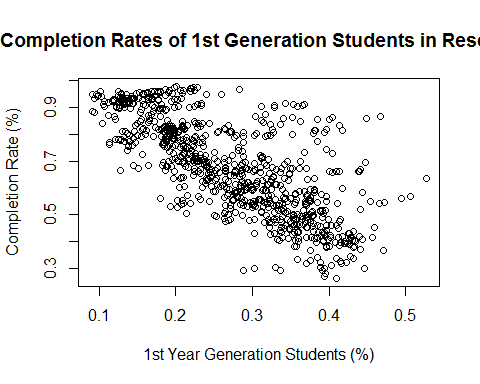
#Correlation between admission rate for research universities and program completion rate  
plot(usresearchuniv$ADM\_RATE\_ALL, usresearchuniv$C150\_4, main="Admission Rate vs. Program Completion Rate for Research Universities (AY2010-2015)", xlab="Admission Rate (%)", ylab="Completion Rate (%)")



#Correlation between attendees and completion rate of non-resident aliens (International Students)  
plot(usresearchuniv$UGDS\_NRA, usresearchuniv$C150\_4\_NRA, main="Percentage of Attendees vs. Completion Rates of International Students in Research Universities (AY2010-2015)", xlab="Population Share of International Students (%)", ylab="Completion Rate of International Students (%)")



#Correlation between attendees and completion rate of 1st Generation students in Research Universities  
plot(usresearchuniv$PAR\_ED\_PCT\_1STGEN, usresearchuniv$C150\_4, main="Percentage of Attendees vs. Completion Rates of 1st Generation Students in Research Universities (AY2010-2015)", xlab="1st Year Generation Students (%)", ylab="Completion Rate (%)")



# U.S. Research University Acceptance Model

In this report section, we are going to create a formula on getting an acceptance to a US Research University based on the College Scorecard statistics. We will try different methods of regression, and find the best regression technique from the following sources.

We will also consider another formula based on an international student taking up science degree/major.

# create a training and test model using a 75%/25% from the data set   
rm\_train <- sample(nrow(usunivfilter), floor(nrow(usunivfilter)\*0.75))  
univ\_train <- usunivfilter[rm\_train,]  
univ\_test <- usunivfilter[-rm\_train,]  
  
# create a generic formula for the US research university acceptance model for International Students based on SAT, average cost, loans, and gender  
formula\_ISAcceptance <- formula(ACCEPTED ~ REGION + ADM\_RATE\_ALL + SAT\_AVG\_ALL + UGDS\_NRA + COSTT4\_A + PCTFLOAN + UGDS\_WOMEN)  
  
# do a logistic regression model based on this  
glm\_ISAcceptance <- glm(formula\_ISAcceptance, data = univ\_train, family = binomial())  
summary(glm\_ISAcceptance)

##   
## Call:  
## glm(formula = formula\_ISAcceptance, family = binomial(), data = univ\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.1280 -0.5405 -0.2929 -0.1168 2.7949   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.454e+01 1.203e+00 -12.091 < 2e-16 \*\*\*  
## REGION 1.115e-01 2.907e-02 3.835 0.000125 \*\*\*  
## ADM\_RATE\_ALL 7.853e-01 3.764e-01 2.086 0.036966 \*   
## SAT\_AVG\_ALL 1.466e-02 8.492e-04 17.263 < 2e-16 \*\*\*  
## UGDS\_NRA 6.343e+00 1.382e+00 4.591 4.41e-06 \*\*\*  
## COSTT4\_A -9.294e-05 6.264e-06 -14.836 < 2e-16 \*\*\*  
## PCTFLOAN -6.410e-01 4.903e-01 -1.307 0.191071   
## UGDS\_WOMEN -2.497e+00 5.584e-01 -4.471 7.78e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 3125.6 on 3184 degrees of freedom  
## Residual deviance: 2128.7 on 3177 degrees of freedom  
## AIC: 2144.7  
##   
## Number of Fisher Scoring iterations: 6

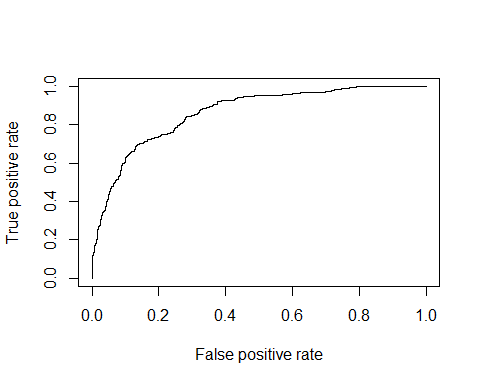
# do the first testing with the prediction model  
accepted\_ind <- predict(glm\_ISAcceptance, type="response", newdata = univ\_test)  
pred1 <- prediction(accepted\_ind, univ\_test$ACCEPTED)  
  
# create the confusion matrix and accuracy for this prediction model  
c1 <- confusionMatrix(as.integer(accepted\_ind > 0.5), univ\_test$ACCEPTED)  
c1$table

## Reference  
## Prediction 0 1  
## 0 819 115  
## 1 43 85

c1$overall['Accuracy']

## Accuracy   
## 0.8512241

# show the curve on the performance  
perf1 <- performance(pred1, "tpr", "fpr")  
plot(perf1, lty = 1)



# Now we check on what acceptable ways we could do for regression  
# doing single decision tree  
model\_dtree1 <- rpart(formula\_ISAcceptance, method="anova",data = univ\_train)  
pred\_dtree1 <- predict(model\_dtree1, newdata = univ\_test)  
accu1 = abs(pred\_dtree1 - univ\_test$ACCEPTED) < 0.5  
frac1 = sum(accu1)/length(accu1)  
print(frac1)

## [1] 0.86629

# doing random forest  
model\_forest1 <- randomForest(formula\_ISAcceptance, data = univ\_train)

## Warning in randomForest.default(m, y, ...): The response has five or fewer  
## unique values. Are you sure you want to do regression?

pred\_forest1 <- predict(model\_forest1, newdata = univ\_test)  
accu2 <- abs(pred\_forest1 - univ\_test$ACCEPTED) < 0.5  
frac2 <- sum(accu2)/length(accu2)  
print(frac2)

## [1] 0.9350282

# doing support vector machine  
model\_svm1 <- svm(formula\_ISAcceptance, data = univ\_train)  
pred\_svm1 <- predict(model\_svm1, newdata = univ\_test)  
accu3 <- abs(pred\_svm1 - univ\_test$ACCEPTED) < 0.5  
frac3 <- sum(accu3)/length(accu3)  
print(frac3)

## [1] 0.8898305

# doing simple tree  
model\_tree1 <- tree(formula\_ISAcceptance, data = univ\_train)  
pred\_tree1 <- predict(model\_tree1, newdata = univ\_test)  
accu4 <- abs(pred\_tree1 - univ\_test$ACCEPTED) < 0.5  
frac4 <- sum(accu4)/length(accu4)  
print(frac4)

## [1] 0.86629

# doing conditional inference tree  
model\_party1 <- ctree(formula\_ISAcceptance, data = univ\_train)  
pred\_party1 <- predict(model\_party1, newdata = univ\_test)  
accu5 <- abs(pred\_party1 - univ\_test$ACCEPTED) < 0.5  
frac5 <- sum(accu5)/length(accu5)  
print(frac5)

## [1] 0.8653484

Based on the run, random forest is the best regression method to use in this model.

Next, another formula is created. This is an acceptance model for an international student that wants to take up Science degree/major

# create a formula for the US research university acceptance model for International Students taking up Science degrees/majors  
formula\_ISSciAcceptance <- formula(ACCEPTED ~ REGION + ADM\_RATE\_ALL + SAT\_AVG\_ALL + PCIP11 + PCIP12 + PCIP14 + PCIP15 + PCIP24 + PCIP26 + PCIP27 + PCIP40 + PCIP45 + PCIP51 + PCIP52 + UGDS\_NRA + UGDS\_UNKN + COSTT4\_A + PCTFLOAN + UGDS\_WOMEN)  
  
# do a logistic regression model based on the formula created  
glm\_ISSciAcceptance <- glm(formula\_ISSciAcceptance, data=univ\_train,family=binomial())  
summary(glm\_ISSciAcceptance)

##   
## Call:  
## glm(formula = formula\_ISSciAcceptance, family = binomial(), data = univ\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.55082 -0.47745 -0.23826 -0.08003 3.13892   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.717e+01 1.430e+00 -12.005 < 2e-16 \*\*\*  
## REGION 1.135e-01 3.138e-02 3.618 0.000296 \*\*\*  
## ADM\_RATE\_ALL 1.105e+00 4.153e-01 2.660 0.007808 \*\*   
## SAT\_AVG\_ALL 1.566e-02 1.025e-03 15.283 < 2e-16 \*\*\*  
## PCIP11 5.339e-01 1.978e+00 0.270 0.787201   
## PCIP12 -6.242e-02 1.879e+01 -0.003 0.997349   
## PCIP14 5.667e+00 8.200e-01 6.911 4.80e-12 \*\*\*  
## PCIP15 -2.161e+00 2.289e+00 -0.944 0.345101   
## PCIP24 -6.001e+00 1.272e+00 -4.716 2.41e-06 \*\*\*  
## PCIP26 7.766e+00 1.788e+00 4.344 1.40e-05 \*\*\*  
## PCIP27 -3.430e+01 7.158e+00 -4.791 1.66e-06 \*\*\*  
## PCIP40 -3.305e+01 4.893e+00 -6.755 1.43e-11 \*\*\*  
## PCIP45 8.505e+00 1.232e+00 6.904 5.05e-12 \*\*\*  
## PCIP51 1.963e+00 6.116e-01 3.209 0.001330 \*\*   
## PCIP52 5.772e-01 6.610e-01 0.873 0.382490   
## UGDS\_NRA 8.793e+00 1.586e+00 5.545 2.94e-08 \*\*\*  
## UGDS\_UNKN -1.971e+00 1.637e+00 -1.204 0.228572   
## COSTT4\_A -1.112e-04 7.384e-06 -15.058 < 2e-16 \*\*\*  
## PCTFLOAN -4.712e-01 5.701e-01 -0.827 0.408495   
## UGDS\_WOMEN -3.476e-01 8.357e-01 -0.416 0.677395   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 3125.6 on 3184 degrees of freedom  
## Residual deviance: 1878.0 on 3165 degrees of freedom  
## AIC: 1918  
##   
## Number of Fisher Scoring iterations: 6

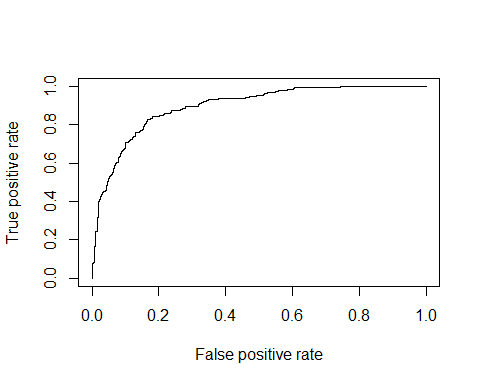
# do the testing with the prediction model  
accepted\_ind2 <- predict(glm\_ISSciAcceptance, type="response", newdata = univ\_test)  
pred2 <- prediction(accepted\_ind2, univ\_test$ACCEPTED)  
  
# prepare confusion matrix and accuracy to see the scores  
c2 <- confusionMatrix(as.integer(accepted\_ind2 > 0.5), univ\_test$ACCEPTED)  
c2$table

## Reference  
## Prediction 0 1  
## 0 817 96  
## 1 45 104

c2$overall['Accuracy']

## Accuracy   
## 0.8672316

# show the curve on the performance  
perf2 <- performance(pred2,"tpr","fpr")  
plot(perf2, lty = 1)



# Now we check on what acceptable ways we could do for regression  
# doing single decision tree  
model\_dtree2 <- rpart(formula\_ISSciAcceptance, method="anova",data = univ\_train)  
pred\_dtree2 <- predict(model\_dtree2, newdata = univ\_test)  
accu6 <- abs(pred\_dtree2 - univ\_test$ACCEPTED) < 0.5  
frac6 <- sum(accu6)/length(accu6)  
print(frac6)

## [1] 0.9048964

# doing random forest  
model\_forest2 <- randomForest(formula\_ISSciAcceptance, data = univ\_train)

## Warning in randomForest.default(m, y, ...): The response has five or fewer  
## unique values. Are you sure you want to do regression?

pred\_forest2 <- predict(model\_forest2, newdata = univ\_test)  
accu7 <- abs(pred\_forest2 - univ\_test$ACCEPTED) < 0.5  
frac7 <- sum(accu7)/length(accu7)  
print(frac7)

## [1] 0.9623352

# doing support vector machine  
model\_svm2 <- svm(formula\_ISSciAcceptance, data = univ\_train)  
pred\_svm2 <- predict(model\_svm2, newdata = univ\_test)  
accu8 <- abs(pred\_svm2 - univ\_test$ACCEPTED) < 0.5  
frac8 <- sum(accu8)/length(accu8)  
print(frac8)

## [1] 0.9161959

# doing simple tree  
model\_tree2 <- tree(formula\_ISSciAcceptance, data = univ\_train)  
pred\_tree2 <- predict(model\_tree2, newdata = univ\_test)  
accu9 <- abs(pred\_tree2 - univ\_test$ACCEPTED) < 0.5  
frac9 <- sum(accu9)/length(accu9)  
print(frac9)

## [1] 0.9048964

# doing conditional inference tree  
model\_party2 <- ctree(formula\_ISSciAcceptance, data = univ\_train)  
pred\_party2 <- predict(model\_party2, newdata = univ\_test)  
accu10 <- abs(pred\_party2 - univ\_test$ACCEPTED) < 0.5  
frac10 <- sum(accu10)/length(accu10)  
print(frac10)

## [1] 0.8898305

Based on this, random forest is the best regression method to use.

In this portion, we will consider all variables, and use Boruta and RFE to use what variables we could use for doing a better outcome of the moded

# First, we will create another copy of the dataset  
usunivnoccbasic <- usunivfilter  
  
# Next, we will change those that have "NA" to 0, since there is no data in it  
usunivnoccbasic[usunivnoccbasic == "NA"] <- 0  
  
# Next, we will choose rows that have complete cases  
usunivnoccbasic <- usunivnoccbasic[complete.cases(usunivnoccbasic),]  
  
# Now that we have the cleansed dataset, we will implement Boruta  
boruta.train <- Boruta(ACCEPTED ~ .-CCBASIC2, data=usunivnoccbasic)  
print(boruta.train)

## Boruta performed 99 iterations in 6.637883 mins.  
## 61 attributes confirmed important: ADM\_RATE, ADM\_RATE\_ALL,  
## C150\_4, C150\_4\_2MOR, C150\_4\_AIAN and 56 more.  
## 7 attributes confirmed unimportant: C150\_4\_NHPI, PCIP12, PCIP25,  
## PCIP29, PCIP46 and 2 more.  
## 2 tentative attributes left: PCIP10, PCIP22.

getSelectedAttributes(boruta.train)

## [1] "REGION" "ADM\_RATE" "ADM\_RATE\_ALL"   
## [4] "SAT\_AVG\_ALL" "PCIP01" "PCIP03"   
## [7] "PCIP04" "PCIP05" "PCIP09"   
## [10] "PCIP11" "PCIP13" "PCIP14"   
## [13] "PCIP15" "PCIP16" "PCIP19"   
## [16] "PCIP23" "PCIP24" "PCIP26"   
## [19] "PCIP27" "PCIP30" "PCIP31"   
## [22] "PCIP38" "PCIP39" "PCIP40"   
## [25] "PCIP41" "PCIP42" "PCIP43"   
## [28] "PCIP44" "PCIP45" "PCIP49"   
## [31] "PCIP50" "PCIP51" "PCIP52"   
## [34] "PCIP54" "UGDS\_WHITE" "UGDS\_BLACK"   
## [37] "UGDS\_HISP" "UGDS\_ASIAN" "UGDS\_AIAN"   
## [40] "UGDS\_NHPI" "UGDS\_2MOR" "UGDS\_NRA"   
## [43] "UGDS\_UNKN" "PPTUG\_EF" "COSTT4\_A"   
## [46] "TUITIONFEE\_IN" "TUITIONFEE\_OUT" "C150\_4"   
## [49] "C150\_4\_WHITE" "C150\_4\_BLACK" "C150\_4\_HISP"   
## [52] "C150\_4\_ASIAN" "C150\_4\_AIAN" "C150\_4\_2MOR"   
## [55] "C150\_4\_NRA" "C150\_4\_UNKN" "RET\_FT4"   
## [58] "PCTFLOAN" "PAR\_ED\_PCT\_1STGEN" "UGDS\_MEN"   
## [61] "UGDS\_WOMEN"

# We will print the stats of the variables that would be accepted or not  
stats <- attStats(boruta.train)  
print(stats)

## meanImp medianImp minImp maxImp normHits  
## REGION 5.48986547 5.5764846 4.0238390 6.85127533 1.00000000  
## ADM\_RATE 7.12670847 7.1978557 5.5369565 8.39711474 1.00000000  
## ADM\_RATE\_ALL 7.33220384 7.3199171 6.0754068 8.55828806 1.00000000  
## SAT\_AVG\_ALL 12.62223310 12.6252231 11.4550385 14.26705032 1.00000000  
## PCIP01 6.22981477 6.2892032 4.7122646 7.34605482 1.00000000  
## PCIP03 6.58262771 6.5903045 4.6782346 7.82100345 1.00000000  
## PCIP04 11.68587879 11.6509294 10.4192048 13.07258352 1.00000000  
## PCIP05 8.35059958 8.3191540 7.2892642 9.42417759 1.00000000  
## PCIP09 4.82012263 4.8658259 2.9334424 6.84822451 0.98989899  
## PCIP10 2.67664850 2.8196417 0.3901718 4.39250864 0.52525253  
## PCIP11 6.50391297 6.5010782 4.8364039 8.44887724 1.00000000  
## PCIP12 0.56083461 0.6921771 -2.2524392 2.44835096 0.00000000  
## PCIP13 6.00083087 5.9733416 3.5511062 7.53443212 1.00000000  
## PCIP14 18.67574850 18.6782216 17.2032189 20.15532281 1.00000000  
## PCIP15 4.90165658 4.9116398 1.9031639 6.38126173 0.98989899  
## PCIP16 7.57967370 7.5707296 6.3331594 9.15580477 1.00000000  
## PCIP19 7.65581252 7.6951995 6.3608447 9.14104583 1.00000000  
## PCIP22 2.46579176 2.4282897 0.2405642 5.03723582 0.39393939  
## PCIP23 8.46951814 8.4175560 6.8864974 10.13838016 1.00000000  
## PCIP24 5.90350112 5.8935974 4.4907830 7.66078486 1.00000000  
## PCIP25 -1.05121667 -1.0010015 -2.2448923 0.07023359 0.00000000  
## PCIP26 5.91390475 5.8963983 4.4238578 7.76542776 1.00000000  
## PCIP27 5.31332109 5.3903971 3.4118245 7.30330396 1.00000000  
## PCIP29 0.00000000 0.0000000 0.0000000 0.00000000 0.00000000  
## PCIP30 4.23325711 4.1565571 2.3659130 6.35235407 0.92929293  
## PCIP31 4.94096320 4.9843890 2.7718797 6.80612509 0.96969697  
## PCIP38 4.27262707 4.3868517 2.5145756 5.91183434 0.90909091  
## PCIP39 5.47495206 5.4369568 4.2511005 6.69168265 1.00000000  
## PCIP40 5.72665082 5.7441088 4.1180288 7.40934313 1.00000000  
## PCIP41 3.15052162 3.2328166 0.8257030 4.94183403 0.68686869  
## PCIP42 4.94094124 4.9266680 2.0058214 6.66059264 0.97979798  
## PCIP43 7.07854580 6.9774467 5.2857271 8.95261169 1.00000000  
## PCIP44 4.54674620 4.5702151 2.1849952 6.02742232 0.94949495  
## PCIP45 7.45839673 7.4120724 5.3791210 8.80125578 1.00000000  
## PCIP46 -0.05591078 0.0000000 -1.0673758 1.73060709 0.00000000  
## PCIP47 -0.02908648 0.0000000 -1.2767639 1.41684848 0.00000000  
## PCIP48 0.34096613 0.4370075 -1.0010015 1.41702560 0.00000000  
## PCIP49 3.29115851 3.3254198 0.7901509 4.86980657 0.72727273  
## PCIP50 5.78913807 5.7649679 3.8929017 7.81470729 1.00000000  
## PCIP51 3.96159421 3.9056897 1.9332097 6.87337769 0.90909091  
## PCIP52 9.78878724 9.8347709 7.6595935 11.49379381 1.00000000  
## PCIP54 3.79103360 3.8012300 1.6221753 5.57906072 0.85858586  
## UGDS\_WHITE 8.13586420 8.1173994 6.8984922 9.89789936 1.00000000  
## UGDS\_BLACK 10.62121844 10.6292004 9.1423191 12.48788797 1.00000000  
## UGDS\_HISP 6.42427181 6.5440472 4.4727396 8.09029836 1.00000000  
## UGDS\_ASIAN 9.19767034 9.2423281 7.9823483 10.33819647 1.00000000  
## UGDS\_AIAN 4.41753205 4.4399501 1.9408106 6.35331614 0.93939394  
## UGDS\_NHPI 3.77681648 3.7502886 1.8098096 5.34367323 0.89898990  
## UGDS\_2MOR 4.58716503 4.6915727 1.6544745 6.55532030 0.93939394  
## UGDS\_NRA 7.21277417 7.1464674 5.8327604 8.76775076 1.00000000  
## UGDS\_UNKN 5.98875605 6.0334501 3.7559610 7.25722267 0.98989899  
## PPTUG\_EF 6.67871907 6.6263012 4.7823285 8.01194600 1.00000000  
## COSTT4\_A 9.73829598 9.7768062 7.9104143 11.18116420 1.00000000  
## TUITIONFEE\_IN 9.45082241 9.4423950 8.1571952 11.02433016 1.00000000  
## TUITIONFEE\_OUT 5.55753227 5.5160817 3.8926064 7.13392403 1.00000000  
## C150\_4 8.00594128 7.9456296 6.4955825 9.48030283 1.00000000  
## C150\_4\_WHITE 6.73988998 6.7493521 5.2459747 8.42162235 1.00000000  
## C150\_4\_BLACK 7.07021641 7.0575032 5.3881356 8.34178704 1.00000000  
## C150\_4\_HISP 5.73283769 5.7817271 4.5856167 7.17850704 1.00000000  
## C150\_4\_ASIAN 5.98440269 5.9319885 4.3418902 7.28189079 1.00000000  
## C150\_4\_AIAN 7.30060003 7.2319092 5.7381917 9.16639566 1.00000000  
## C150\_4\_NHPI 0.66262311 0.7096564 -1.5177765 2.68956664 0.01010101  
## C150\_4\_2MOR 3.15839657 3.2200209 1.4520739 4.65769133 0.68686869  
## C150\_4\_NRA 4.51795234 4.5436051 2.6272728 6.03906475 1.00000000  
## C150\_4\_UNKN 7.20448668 7.1886349 5.7464313 8.72530511 1.00000000  
## RET\_FT4 10.62552805 10.7363144 9.2301743 11.74140876 1.00000000  
## PCTFLOAN 14.18921158 14.2138947 12.7339226 15.63276570 1.00000000  
## PAR\_ED\_PCT\_1STGEN 5.96350550 5.9926178 4.3686494 7.34009325 1.00000000  
## UGDS\_MEN 12.51516640 12.5654472 11.1199518 13.72805584 1.00000000  
## UGDS\_WOMEN 12.37656343 12.4100971 11.0568982 14.12074161 1.00000000  
## decision  
## REGION Confirmed  
## ADM\_RATE Confirmed  
## ADM\_RATE\_ALL Confirmed  
## SAT\_AVG\_ALL Confirmed  
## PCIP01 Confirmed  
## PCIP03 Confirmed  
## PCIP04 Confirmed  
## PCIP05 Confirmed  
## PCIP09 Confirmed  
## PCIP10 Tentative  
## PCIP11 Confirmed  
## PCIP12 Rejected  
## PCIP13 Confirmed  
## PCIP14 Confirmed  
## PCIP15 Confirmed  
## PCIP16 Confirmed  
## PCIP19 Confirmed  
## PCIP22 Tentative  
## PCIP23 Confirmed  
## PCIP24 Confirmed  
## PCIP25 Rejected  
## PCIP26 Confirmed  
## PCIP27 Confirmed  
## PCIP29 Rejected  
## PCIP30 Confirmed  
## PCIP31 Confirmed  
## PCIP38 Confirmed  
## PCIP39 Confirmed  
## PCIP40 Confirmed  
## PCIP41 Confirmed  
## PCIP42 Confirmed  
## PCIP43 Confirmed  
## PCIP44 Confirmed  
## PCIP45 Confirmed  
## PCIP46 Rejected  
## PCIP47 Rejected  
## PCIP48 Rejected  
## PCIP49 Confirmed  
## PCIP50 Confirmed  
## PCIP51 Confirmed  
## PCIP52 Confirmed  
## PCIP54 Confirmed  
## UGDS\_WHITE Confirmed  
## UGDS\_BLACK Confirmed  
## UGDS\_HISP Confirmed  
## UGDS\_ASIAN Confirmed  
## UGDS\_AIAN Confirmed  
## UGDS\_NHPI Confirmed  
## UGDS\_2MOR Confirmed  
## UGDS\_NRA Confirmed  
## UGDS\_UNKN Confirmed  
## PPTUG\_EF Confirmed  
## COSTT4\_A Confirmed  
## TUITIONFEE\_IN Confirmed  
## TUITIONFEE\_OUT Confirmed  
## C150\_4 Confirmed  
## C150\_4\_WHITE Confirmed  
## C150\_4\_BLACK Confirmed  
## C150\_4\_HISP Confirmed  
## C150\_4\_ASIAN Confirmed  
## C150\_4\_AIAN Confirmed  
## C150\_4\_NHPI Rejected  
## C150\_4\_2MOR Confirmed  
## C150\_4\_NRA Confirmed  
## C150\_4\_UNKN Confirmed  
## RET\_FT4 Confirmed  
## PCTFLOAN Confirmed  
## PAR\_ED\_PCT\_1STGEN Confirmed  
## UGDS\_MEN Confirmed  
## UGDS\_WOMEN Confirmed

#Now, let us try RFE  
rfe\_control <- rfeControl(functions=rfFuncs, method="cv", number = 10)  
rfe.train <- rfe(usunivnoccbasic[,1:70], usunivnoccbasic[,72], sizes = 1:70, rfeControl = rfe\_control)

##   
## Attaching package: 'plyr'

## The following object is masked from 'package:modeltools':  
##   
## empty

predictors(rfe.train)

## [1] "PCIP14" "PCTFLOAN" "PCIP04" "SAT\_AVG\_ALL"   
## [5] "UGDS\_BLACK" "PCIP52" "UGDS\_MEN" "UGDS\_WOMEN"   
## [9] "PCIP45" "PCIP43" "COSTT4\_A" "PCIP23"   
## [13] "RET\_FT4" "UGDS\_HISP" "TUITIONFEE\_IN" "C150\_4\_AIAN"   
## [17] "PCIP39" "UGDS\_ASIAN" "PCIP16" "UGDS\_WHITE"   
## [21] "PCIP19" "UGDS\_NRA" "C150\_4" "PCIP05"   
## [25] "PPTUG\_EF" "PCIP03" "PCIP24" "PCIP26"   
## [29] "PCIP09" "PCIP50"

Based on these runs, Boruta has 61 attributes that are confirmed important, and 2 that are tentative. On the other hand, RFE confirms less than 30 variables that are very important.

# US Research University Completion Rate Prediction Model

rm\_train2 <- sample(nrow(usresearchuniv), floor(nrow(usresearchuniv)\*0.75))  
univ\_train2 <- usresearchuniv[rm\_train2,]  
univ\_test2 <- usresearchuniv[-rm\_train2,]  
  
formula\_completionrate <- formula(C150\_4\_NRA ~ REGION + ADM\_RATE\_ALL + UGDS\_NRA + PPTUG\_EF + COSTT4\_A + PCTFLOAN + PAR\_ED\_PCT\_1STGEN)  
  
# using multivariate linear regression to calculate the completion rate for international students  
lm\_NRAcompletion <- lm(formula\_completionrate, data = univ\_train2)  
summary(lm\_NRAcompletion)

##   
## Call:  
## lm(formula = formula\_completionrate, data = univ\_train2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.63579 -0.06718 0.01071 0.07277 0.46776   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 9.268e-01 4.361e-02 21.249 < 2e-16 \*\*\*  
## REGION -2.228e-03 3.223e-03 -0.691 0.4897   
## ADM\_RATE\_ALL -1.503e-01 3.707e-02 -4.055 5.68e-05 \*\*\*  
## UGDS\_NRA 3.182e-01 1.468e-01 2.168 0.0305 \*   
## PPTUG\_EF -3.715e-01 8.267e-02 -4.494 8.39e-06 \*\*\*  
## COSTT4\_A 1.237e-06 5.916e-07 2.091 0.0369 \*   
## PCTFLOAN -3.761e-01 5.868e-02 -6.409 2.95e-10 \*\*\*  
## PAR\_ED\_PCT\_1STGEN -5.515e-03 9.972e-02 -0.055 0.9559   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1375 on 603 degrees of freedom  
## Multiple R-squared: 0.409, Adjusted R-squared: 0.4022   
## F-statistic: 59.62 on 7 and 603 DF, p-value: < 2.2e-16

# do the testing with the prediction model  
accepted\_ind3 <- predict(lm\_NRAcompletion, interval="prediction", newdata = univ\_test2)  
  
# Checking on PRED(25)  
errors <- accepted\_ind3[,"fit"] - univ\_test2$C150\_4\_NRA  
rel\_change <- abs(errors) / univ\_test2$C150\_4\_NRA  
table(rel\_change<0.25)["TRUE"] / nrow(univ\_test2)

## TRUE   
## 0.7647059

# Now we check on what acceptable ways we could do for regression  
# Doing single decision tree  
model\_dtree3 <- rpart(formula\_completionrate, method="anova",data = univ\_train2)  
pred\_dtree3 <- predict(model\_dtree3, newdata = univ\_test2)  
accu11 <- abs(pred\_dtree3 - univ\_test2$C150\_4\_NRA) < 0.25  
frac11 <- sum(accu11)/length(accu11)  
print(frac11)

## [1] 0.8676471

# Doing random forest  
model\_forest3 <- randomForest(formula\_completionrate, data = univ\_train2)  
pred\_forest3 <- predict(model\_forest3, newdata = univ\_test2)  
accu12 <- abs(pred\_forest3 - univ\_test2$C150\_4\_NRA) < 0.25  
frac12 <- sum(accu12)/length(accu12)  
print(frac12)

## [1] 0.9166667

# Doing support vector machine  
model\_svm3 <- svm(formula\_completionrate, data = univ\_train2)  
pred\_svm3 <- predict(model\_svm3, newdata = univ\_test2)  
accu13 <- abs(pred\_svm3 - univ\_test2$C150\_4\_NRA) < 0.25  
frac13 <- sum(accu13)/length(accu13)  
print(frac13)

## [1] 0.9117647

# doing simple tree  
model\_tree3 <- tree(formula\_completionrate, data = univ\_train2)  
pred\_tree3 <- predict(model\_tree3, newdata = univ\_test2)  
accu14 <- abs(pred\_tree3 - univ\_test2$C150\_4\_NRA) < 0.25  
frac14 <- sum(accu14)/length(accu14)  
print(frac14)

## [1] 0.877451

# doing conditional inference tree  
model\_party3 <- ctree(formula\_completionrate, data = univ\_train2)  
pred\_party3 <- predict(model\_party3, newdata = univ\_test2)  
accu15 <- abs(pred\_party3 - univ\_test2$C150\_4\_NRA) < 0.25  
frac15 <- sum(accu15)/length(accu15)  
print(frac15)

## [1] 0.8970588

From the regressions that we have run, the random forest is the best regression model to use for determining completion rates for international students.