annand\_final\_project\_qualitative

Joseph Annand

2023-12-14

## Load libraries

library(ISLR2)  
library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:ISLR2':  
##   
## Boston

library(e1071)

## Warning: package 'e1071' was built under R version 4.3.2

library(class)  
library(tree)

## Warning: package 'tree' was built under R version 4.3.2

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.3.2

## randomForest 4.7-1.1

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

library(gbm)

## Warning: package 'gbm' was built under R version 4.3.2

## Loaded gbm 2.1.8.1

## Prepare data set

full.data <- read.table("student\_data.csv", sep=";", header = T)  
View(full.data)  
  
student.data <- na.omit(full.data)  
student.data$Target[student.data$Target=="Enrolled"] <- "Student"  
student.data$Target[student.data$Target=="Graduate"] <- "Student"  
  
length(student.data$Target[student.data$Target == "Dropout"]) # 1421 dropout records

## [1] 1421

student.data$Target <- as.factor(student.data$Target)

The data set includes 4424 records of undergraduate students in higher education who either dropped out of school or did not drop out. Each record has 21 attributes that are evaluated and used as predictor variables in a model to predict if a student drops out or not. Of the 4424 records, 1421 of them dropped out, which is equal to a little less than a third of the total number of records. The original data set includes three possible values of the target variable: dropout, enrolled, and graduate. In this analysis, all records where the target was equal to enrolled and graduate were set to “Student” to indicate that the student associated with that record has not or did not drop out. Before loading the data set, the column names had to be manually adjusted to remove spaces and parentheses in the names. Upon successfully loading data into R Studio program and adjusting the values in the target column to meet criteria for binary classification problem, the target column was converted from character class to factor class.

## Split data set into train and test data

set.seed(1)  
train <- sample(1:nrow(student.data), 0.5 \* nrow(student.data))  
test <- (-train)

The total data set was split in half where one half was assigned to the training data set, which was used to train the various models to predict whether a student will drop out or not. The second half of the data was assigned to the test data and used to measure the accuracy of the model in predicting the response.

# Create data frame to track test errors of each model  
model.errors <- data.frame(model = c("Log Regression", "LDA", "QDA", "NB Classifier",  
 "KNN", "Tree", "Pruned Tree", "Bagging", "RF",  
 "Boosting"),  
 test.error = rep(NA, 10))

## Logistic Regression with All Predictors

log.student <- glm(Target ~ ., data = student.data, family = binomial)  
  
summary(log.student)

##   
## Call:  
## glm(formula = Target ~ ., family = binomial, data = student.data)  
##   
## Coefficients:  
## Estimate Std. Error z value  
## (Intercept) 4.513e-01 7.488e-01 0.603  
## Marital.status 1.267e-01 9.782e-02 1.295  
## Application.mode -6.871e-04 3.713e-03 -0.185  
## Application.order -8.428e-02 4.302e-02 -1.959  
## Course -9.950e-05 3.603e-05 -2.762  
## Daytime.evening.attendance -3.113e-02 1.797e-01 -0.173  
## Previous.qualification 1.107e-02 5.395e-03 2.052  
## Previous.qualification.grade -3.115e-03 4.590e-03 -0.679  
## Nacionality -3.550e-02 1.069e-02 -3.321  
## Mother.qualification -1.117e-02 3.990e-03 -2.800  
## Father.qualification 4.386e-03 3.909e-03 1.122  
## Mother.occupation 1.237e-02 4.854e-03 2.548  
## Father.occupation -2.069e-03 5.058e-03 -0.409  
## Admission.grade 4.493e-03 4.271e-03 1.052  
## Displaced -3.412e-01 1.161e-01 -2.938  
## Educational.special.needs -2.619e-01 4.193e-01 -0.624  
## Debtor -4.585e-01 1.663e-01 -2.756  
## Tuition.fees.up.to.date 2.417e+00 1.815e-01 13.319  
## Gender -2.905e-01 1.059e-01 -2.743  
## Scholarship.holder 5.687e-01 1.386e-01 4.104  
## Age.at.enrollment -4.785e-02 9.435e-03 -5.071  
## International 1.975e+00 5.752e-01 3.434  
## Curricular.units.1st.sem.credited -1.429e-01 7.894e-02 -1.810  
## Curricular.units.1st.sem.enrolled 1.649e-02 1.015e-01 0.162  
## Curricular.units.1st.sem.evaluations 7.084e-03 2.398e-02 0.295  
## Curricular.units.1st.sem.approved 2.964e-01 5.169e-02 5.735  
## Curricular.units.1st.sem.grade -5.742e-02 2.372e-02 -2.421  
## Curricular.units.1st.sem.without.evaluations 1.365e-01 9.474e-02 1.441  
## Curricular.units.2nd.sem.credited -2.274e-01 8.441e-02 -2.694  
## Curricular.units.2nd.sem.enrolled -4.842e-01 9.774e-02 -4.954  
## Curricular.units.2nd.sem.evaluations 3.862e-02 2.272e-02 1.700  
## Curricular.units.2nd.sem.approved 6.136e-01 4.765e-02 12.877  
## Curricular.units.2nd.sem.grade 6.915e-02 2.236e-02 3.093  
## Curricular.units.2nd.sem.without.evaluations 1.120e-01 7.833e-02 1.430  
## Unemployment.rate -8.048e-02 2.099e-02 -3.834  
## Inflation.rate -2.374e-02 3.614e-02 -0.657  
## GDP -8.831e-03 2.501e-02 -0.353  
## Pr(>|z|)   
## (Intercept) 0.546677   
## Marital.status 0.195344   
## Application.mode 0.853204   
## Application.order 0.050125 .   
## Course 0.005749 \*\*   
## Daytime.evening.attendance 0.862479   
## Previous.qualification 0.040126 \*   
## Previous.qualification.grade 0.497404   
## Nacionality 0.000898 \*\*\*  
## Mother.qualification 0.005117 \*\*   
## Father.qualification 0.261874   
## Mother.occupation 0.010839 \*   
## Father.occupation 0.682497   
## Admission.grade 0.292814   
## Displaced 0.003304 \*\*   
## Educational.special.needs 0.532325   
## Debtor 0.005848 \*\*   
## Tuition.fees.up.to.date < 2e-16 \*\*\*  
## Gender 0.006091 \*\*   
## Scholarship.holder 4.07e-05 \*\*\*  
## Age.at.enrollment 3.95e-07 \*\*\*  
## International 0.000594 \*\*\*  
## Curricular.units.1st.sem.credited 0.070322 .   
## Curricular.units.1st.sem.enrolled 0.870944   
## Curricular.units.1st.sem.evaluations 0.767641   
## Curricular.units.1st.sem.approved 9.75e-09 \*\*\*  
## Curricular.units.1st.sem.grade 0.015479 \*   
## Curricular.units.1st.sem.without.evaluations 0.149581   
## Curricular.units.2nd.sem.credited 0.007053 \*\*   
## Curricular.units.2nd.sem.enrolled 7.27e-07 \*\*\*  
## Curricular.units.2nd.sem.evaluations 0.089180 .   
## Curricular.units.2nd.sem.approved < 2e-16 \*\*\*  
## Curricular.units.2nd.sem.grade 0.001985 \*\*   
## Curricular.units.2nd.sem.without.evaluations 0.152778   
## Unemployment.rate 0.000126 \*\*\*  
## Inflation.rate 0.511307   
## GDP 0.724043   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 5554.5 on 4423 degrees of freedom  
## Residual deviance: 2739.7 on 4387 degrees of freedom  
## AIC: 2813.7  
##   
## Number of Fisher Scoring iterations: 6

sig\_predictors <- c("Application.order", "Course", "Previous.qualification",  
 "Nacionality", "Mother.qualification", "Mother.occupation",  
 "Displaced", "Debtor", "Tuition.fees.up.to.date",  
 "Gender", "Scholarship.holder", "Age.at.enrollment", "International",  
 "Curricular.units.1st.sem.approved", "Curricular.units.1st.sem.grade",  
 "Curricular.units.2nd.sem.credited", "Curricular.units.2nd.sem.enrolled",  
 "Curricular.units.2nd.sem.approved", "Curricular.units.2nd.sem.grade",  
 "Unemployment.rate", "Target")  
  
student.data <- student.data[, sig\_predictors]

To begin, the total data set is fit using a logistic regression model. The summary of the model reveals information about each predictor. The p-value of each predictor variable is calculated, and those with a p-value less than 0.05 are statistically significant. This information is used to subset our full data set to include only the statistically significant predictors.

## Logistic Regression

glm.student <- glm(Target ~ ., data = student.data, subset = train,  
 family = binomial)  
  
summary(glm.student)

##   
## Call:  
## glm(formula = Target ~ ., family = binomial, data = student.data,   
## subset = train)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 6.547e-01 5.571e-01 1.175 0.239881   
## Application.order -9.465e-02 6.105e-02 -1.550 0.121039   
## Course -1.164e-04 4.725e-05 -2.463 0.013775 \*   
## Previous.qualification 9.697e-03 7.080e-03 1.370 0.170816   
## Nacionality -4.010e-02 1.172e-02 -3.421 0.000625 \*\*\*  
## Mother.qualification -4.052e-03 4.843e-03 -0.837 0.402774   
## Mother.occupation 1.013e-02 3.149e-03 3.217 0.001297 \*\*   
## Displaced -4.217e-01 1.600e-01 -2.636 0.008382 \*\*   
## Debtor -6.741e-01 2.310e-01 -2.918 0.003519 \*\*   
## Tuition.fees.up.to.date 2.400e+00 2.608e-01 9.205 < 2e-16 \*\*\*  
## Gender -4.124e-01 1.461e-01 -2.823 0.004763 \*\*   
## Scholarship.holder 4.595e-01 1.901e-01 2.418 0.015622 \*   
## Age.at.enrollment -4.895e-02 1.088e-02 -4.497 6.88e-06 \*\*\*  
## International 1.970e+00 6.608e-01 2.981 0.002870 \*\*   
## Curricular.units.1st.sem.approved 2.546e-01 5.879e-02 4.330 1.49e-05 \*\*\*  
## Curricular.units.1st.sem.grade -4.446e-02 3.029e-02 -1.468 0.142157   
## Curricular.units.2nd.sem.credited -3.589e-01 6.538e-02 -5.490 4.01e-08 \*\*\*  
## Curricular.units.2nd.sem.enrolled -3.526e-01 6.793e-02 -5.191 2.09e-07 \*\*\*  
## Curricular.units.2nd.sem.approved 5.850e-01 5.829e-02 10.035 < 2e-16 \*\*\*  
## Curricular.units.2nd.sem.grade 6.859e-02 2.893e-02 2.371 0.017748 \*   
## Unemployment.rate -5.821e-02 2.681e-02 -2.172 0.029891 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 2770.5 on 2211 degrees of freedom  
## Residual deviance: 1387.6 on 2191 degrees of freedom  
## AIC: 1429.6  
##   
## Number of Fisher Scoring iterations: 6

glm.probs <- predict(glm.student, student.data[test, ], type = "response")  
glm.pred <- rep("Student", nrow(student.data[test, ]))  
  
contrasts(student.data$Target)

## Student  
## Dropout 0  
## Student 1

glm.pred[glm.probs < 0.5] <- "Dropout"  
table(glm.pred, student.data$Target[test])

##   
## glm.pred Dropout Student  
## Dropout 516 75  
## Student 199 1422

glm.error <- mean(glm.pred != student.data$Target[test])  
glm.error

## [1] 0.1238698

model.errors[model.errors$model == "Log Regression", "test.error"] <- glm.error

Using only the significant predictors from the first logistic regression model, a new logistic regression models trained using the training data set. The contrasts() function reveals that R has assigned 0 to “Dropout”and 1 to the “Student” response. Therefore, if the prediction value is less than 0.5, the response is equal to “Dropout” while all other predicted responses are set equal to “Student.” The test error rate for the trained logistic regression model is 12.4%.

## Linear Discriminant Analysis

lda.student <- lda(Target ~ ., data = student.data, subset = train)  
lda.student

## Call:  
## lda(Target ~ ., data = student.data, subset = train)  
##   
## Prior probabilities of groups:  
## Dropout Student   
## 0.3191682 0.6808318   
##   
## Group means:  
## Application.order Course Previous.qualification Nacionality  
## Dropout 1.533994 8756.80 5.432011 2.260623  
## Student 1.788181 8930.02 3.899070 1.899734  
## Mother.qualification Mother.occupation Displaced Debtor  
## Dropout 20.63314 10.30312 0.4645892 0.23229462  
## Student 18.48938 11.63878 0.5989376 0.05909695  
## Tuition.fees.up.to.date Gender Scholarship.holder Age.at.enrollment  
## Dropout 0.6756374 0.5113314 0.1062323 25.97592  
## Student 0.9760956 0.2795485 0.3266932 21.78486  
## International Curricular.units.1st.sem.approved  
## Dropout 0.02974504 2.594901  
## Student 0.02589641 5.747012  
## Curricular.units.1st.sem.grade Curricular.units.2nd.sem.credited  
## Dropout 7.406225 0.4518414  
## Student 12.192280 0.5624170  
## Curricular.units.2nd.sem.enrolled Curricular.units.2nd.sem.approved  
## Dropout 5.752125 1.947592  
## Student 6.474768 5.615538  
## Curricular.units.2nd.sem.grade Unemployment.rate  
## Dropout 5.935216 11.60949  
## Student 12.231881 11.55252  
##   
## Coefficients of linear discriminants:  
## LD1  
## Application.order -3.086898e-02  
## Course -3.918505e-05  
## Previous.qualification 4.689505e-03  
## Nacionality -1.775997e-02  
## Mother.qualification -1.387344e-03  
## Mother.occupation 3.678917e-03  
## Displaced -1.233246e-01  
## Debtor -3.779885e-01  
## Tuition.fees.up.to.date 1.188291e+00  
## Gender -1.833625e-01  
## Scholarship.holder 1.608477e-01  
## Age.at.enrollment -2.021151e-02  
## International 8.167896e-01  
## Curricular.units.1st.sem.approved 1.341369e-01  
## Curricular.units.1st.sem.grade -2.361303e-02  
## Curricular.units.2nd.sem.credited -1.787996e-01  
## Curricular.units.2nd.sem.enrolled -2.784260e-01  
## Curricular.units.2nd.sem.approved 3.746244e-01  
## Curricular.units.2nd.sem.grade 4.295753e-02  
## Unemployment.rate -1.789627e-02

lda.pred <- predict(lda.student, student.data[test, ])  
lda.class <- lda.pred$class  
table(lda.class, student.data$Target[test])

##   
## lda.class Dropout Student  
## Dropout 499 60  
## Student 216 1437

lda.error <- (216 + 60) / (499 + 60 + 216 + 1437)  
lda.error

## [1] 0.124774

model.errors[model.errors$model == "LDA", "test.error"] <- lda.error

A linear discriminant analysis (LDA) model is trained and used to predict the response. The LDA model yields a similar test error rate as the logistic regression model with a misclassification rate of 12.5%. Unsurprisingly this is not much different from logistic regression since there are only two classes of the response.

## Quadratic Discriminant Analysis

qda.student <- qda(Target ~ ., data = student.data, subset = train)  
qda.student

## Call:  
## qda(Target ~ ., data = student.data, subset = train)  
##   
## Prior probabilities of groups:  
## Dropout Student   
## 0.3191682 0.6808318   
##   
## Group means:  
## Application.order Course Previous.qualification Nacionality  
## Dropout 1.533994 8756.80 5.432011 2.260623  
## Student 1.788181 8930.02 3.899070 1.899734  
## Mother.qualification Mother.occupation Displaced Debtor  
## Dropout 20.63314 10.30312 0.4645892 0.23229462  
## Student 18.48938 11.63878 0.5989376 0.05909695  
## Tuition.fees.up.to.date Gender Scholarship.holder Age.at.enrollment  
## Dropout 0.6756374 0.5113314 0.1062323 25.97592  
## Student 0.9760956 0.2795485 0.3266932 21.78486  
## International Curricular.units.1st.sem.approved  
## Dropout 0.02974504 2.594901  
## Student 0.02589641 5.747012  
## Curricular.units.1st.sem.grade Curricular.units.2nd.sem.credited  
## Dropout 7.406225 0.4518414  
## Student 12.192280 0.5624170  
## Curricular.units.2nd.sem.enrolled Curricular.units.2nd.sem.approved  
## Dropout 5.752125 1.947592  
## Student 6.474768 5.615538  
## Curricular.units.2nd.sem.grade Unemployment.rate  
## Dropout 5.935216 11.60949  
## Student 12.231881 11.55252

qda.class <- predict(qda.student, student.data[test, ])$class  
table(qda.class, student.data$Target[test])

##   
## qda.class Dropout Student  
## Dropout 508 132  
## Student 207 1365

qda.error <- (207 + 132) / (508 + 132 + 207 + 1365)  
qda.error

## [1] 0.153255

model.errors[model.errors$model == "QDA", "test.error"] <- qda.error

The training data is fit to a quadratic linear discriminant analysis (QDA) model and used to predict the responses in the test data. The QDA model yields a test error rate of 15.3%, slightly worse than logistic regression and LDA. The increase in test error rate suggests that the increased flexibility of the QDA model leads to a worse prediction error rate. This may lead us to believe that the assumption that there is a common covariance among the two response classes is better suited for this problem rather than assuming each class has a its own covariance matrix.

## Naive Bayes Classifier

nb.student <- naiveBayes(Target ~ ., data = student.data, subset = train)  
nb.student

##   
## Naive Bayes Classifier for Discrete Predictors  
##   
## Call:  
## naiveBayes.default(x = X, y = Y, laplace = laplace)  
##   
## A-priori probabilities:  
## Y  
## Dropout Student   
## 0.3191682 0.6808318   
##   
## Conditional probabilities:  
## Application.order  
## Y [,1] [,2]  
## Dropout 1.533994 1.134678  
## Student 1.788181 1.331806  
##   
## Course  
## Y [,1] [,2]  
## Dropout 8756.80 2261.171  
## Student 8930.02 1915.336  
##   
## Previous.qualification  
## Y [,1] [,2]  
## Dropout 5.432011 10.436790  
## Student 3.899070 9.663187  
##   
## Nacionality  
## Y [,1] [,2]  
## Dropout 2.260623 9.032176  
## Student 1.899734 6.968837  
##   
## Mother.qualification  
## Y [,1] [,2]  
## Dropout 20.63314 15.64445  
## Student 18.48938 15.63806  
##   
## Mother.occupation  
## Y [,1] [,2]  
## Dropout 10.30312 20.82031  
## Student 11.63878 29.21718  
##   
## Displaced  
## Y [,1] [,2]  
## Dropout 0.4645892 0.4990981  
## Student 0.5989376 0.4902764  
##   
## Debtor  
## Y [,1] [,2]  
## Dropout 0.23229462 0.4225953  
## Student 0.05909695 0.2358844  
##   
## Tuition.fees.up.to.date  
## Y [,1] [,2]  
## Dropout 0.6756374 0.4684681  
## Student 0.9760956 0.1528021  
##   
## Gender  
## Y [,1] [,2]  
## Dropout 0.5113314 0.5002260  
## Student 0.2795485 0.4489264  
##   
## Scholarship.holder  
## Y [,1] [,2]  
## Dropout 0.1062323 0.3083532  
## Student 0.3266932 0.4691598  
##   
## Age.at.enrollment  
## Y [,1] [,2]  
## Dropout 25.97592 8.437555  
## Student 21.78486 6.425049  
##   
## International  
## Y [,1] [,2]  
## Dropout 0.02974504 0.1700036  
## Student 0.02589641 0.1588790  
##   
## Curricular.units.1st.sem.approved  
## Y [,1] [,2]  
## Dropout 2.594901 2.884082  
## Student 5.747012 2.649718  
##   
## Curricular.units.1st.sem.grade  
## Y [,1] [,2]  
## Dropout 7.406225 5.991239  
## Student 12.192280 3.114367  
##   
## Curricular.units.2nd.sem.credited  
## Y [,1] [,2]  
## Dropout 0.4518414 1.721315  
## Student 0.5624170 1.998443  
##   
## Curricular.units.2nd.sem.enrolled  
## Y [,1] [,2]  
## Dropout 5.752125 2.090038  
## Student 6.474768 2.195365  
##   
## Curricular.units.2nd.sem.approved  
## Y [,1] [,2]  
## Dropout 1.947592 2.575680  
## Student 5.615538 2.430208  
##   
## Curricular.units.2nd.sem.grade  
## Y [,1] [,2]  
## Dropout 5.935216 6.105464  
## Student 12.231881 3.112023  
##   
## Unemployment.rate  
## Y [,1] [,2]  
## Dropout 11.60949 2.766519  
## Student 11.55252 2.638518

nb.class <- predict(nb.student, student.data[test, ])  
table(nb.class, student.data$Target[test])

##   
## nb.class Dropout Student  
## Dropout 514 159  
## Student 201 1338

nb.error <- (201 + 159) / (514 + 159 + 201 + 1338)  
nb.error

## [1] 0.1627486

model.errors[model.errors$model == "NB Classifier", "test.error"] <- nb.error

A naive Bayes classifier model was applied to the training data set. The model yields a test error rate of 16.3%, which is considerably worse than those of the logistic regression and LDA models.

## K-Nearest Neighbor

length(sig\_predictors)

## [1] 21

train.X <- as.matrix(student.data[train, sig\_predictors[-21]])  
test.X <- as.matrix(student.data[test, sig\_predictors[-21]])  
train.target <- student.data$Target[train]  
  
k.values <- c(1, 3, 5, 10)  
knn.errors <- data.frame(k.value = k.values,  
 pred.error = rep(NA, length(k.values)))  
for (x in 1:length(k.values)) {  
 set.seed(2)  
 knn.pred <- knn(train.X, test.X, train.target, k = k.values[x])  
 knn.errors[x, "pred.error"] <- mean(knn.pred != student.data$Target[test])  
}  
  
knn.errors # k = 5 lowest test error

## k.value pred.error  
## 1 1 0.2373418  
## 2 3 0.2255877  
## 3 5 0.2124774  
## 4 10 0.2188065

model.errors[model.errors$model == "KNN", "test.error"] <- min(knn.errors$pred.error)

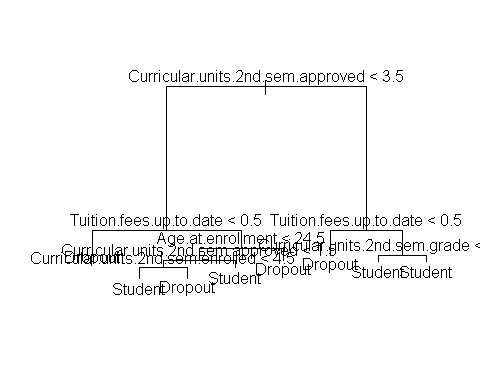
The non-parametric K-nearest neighbor (KNN) model is evaluated with four different values of k: 1, 3, 5, and 10. The model was trained using the same training data set as the previous models that were evaluated. The lowest test error amongst the four k values is 21.2%. Interestingly, similar to the observations in the test error rate for QDA, the results of the KNN modeling suggests that less flexibility leads to better accuracy: the test error rate is higher for k = 1 or 3 compared to k = 5 and 10.

## Classification tree

# Classification Tree Model  
  
tree.student <- tree(Target ~ ., data = student.data, subset = train)  
summary(tree.student)

##   
## Classification tree:  
## tree(formula = Target ~ ., data = student.data, subset = train)  
## Variables actually used in tree construction:  
## [1] "Curricular.units.2nd.sem.approved" "Tuition.fees.up.to.date"   
## [3] "Age.at.enrollment" "Curricular.units.2nd.sem.enrolled"  
## [5] "Curricular.units.2nd.sem.grade"   
## Number of terminal nodes: 8   
## Residual mean deviance: 0.6947 = 1531 / 2204   
## Misclassification error rate: 0.1293 = 286 / 2212

plot(tree.student)  
text(tree.student, pretty = 0)



tree.pred <- predict(tree.student, student.data[test, ], type = "class")  
table(tree.pred, student.data$Target[test])

##   
## tree.pred Dropout Student  
## Dropout 497 86  
## Student 218 1411

tree.error <- mean(tree.pred != student.data$Target[test])  
tree.error

## [1] 0.1374322

model.errors[model.errors$model == "Tree", "test.error"] <- tree.error

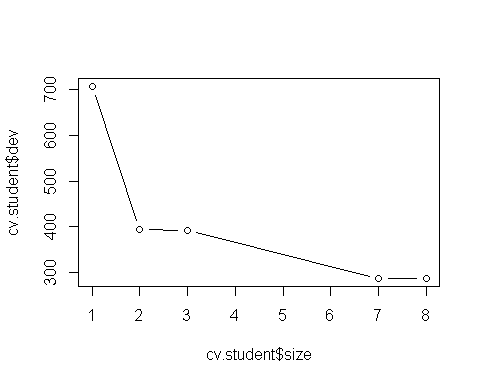
Using all predictors, a classification tree model is trained using the training data set. The tree has 8 terminal nodes, and the variables actually used in the tree construction are second semester curricular units approved, tuition fees up-to-date, age at enrollment, second semester curricular units enrolled, and second semester curricular units grade. The training error rate is 12.9% while the test error rate is 13.7%.

## Pruned Tree

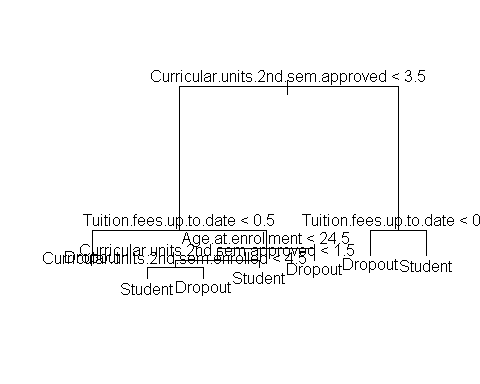
set.seed(4)  
cv.student <- cv.tree(tree.student, FUN = prune.misclass)  
cv.student

## $size  
## [1] 8 7 3 2 1  
##   
## $dev  
## [1] 288 288 392 395 706  
##   
## $k  
## [1] -Inf 0.00 20.25 23.00 316.00  
##   
## $method  
## [1] "misclass"  
##   
## attr(,"class")  
## [1] "prune" "tree.sequence"

plot(cv.student$size, cv.student$dev, type = "b")



pruned.student <- prune.misclass(tree.student, best = 7)  
plot(pruned.student)  
text(pruned.student, pretty = 0)



summary(pruned.student)

##   
## Classification tree:  
## snip.tree(tree = tree.student, nodes = 7L)  
## Variables actually used in tree construction:  
## [1] "Curricular.units.2nd.sem.approved" "Tuition.fees.up.to.date"   
## [3] "Age.at.enrollment" "Curricular.units.2nd.sem.enrolled"  
## Number of terminal nodes: 7   
## Residual mean deviance: 0.7098 = 1565 / 2205   
## Misclassification error rate: 0.1293 = 286 / 2212

pruned.pred <- predict(pruned.student, student.data[test, ], type = "class")  
table(pruned.pred, student.data$Target[test])

##   
## pruned.pred Dropout Student  
## Dropout 497 86  
## Student 218 1411

prune.error <- mean(pruned.pred != student.data$Target[test])  
prune.error

## [1] 0.1374322

model.errors[model.errors$model == "Pruned Tree", "test.error"] <- prune.error

Pruning the classification tree model attempts to reduce the test error rate. Cross-validation is used to determine what tree size yields the lowest training error. Plotting error over the size of the tree shows that the a tree size of 7 yields the lowest error. the tree is pruned so that its size matches that determined from the plot. In the pruned tree, second semester curricular units grade is no longer a predictor used. The pruned tree has 7 terminal nodes. Despite the changes, the pruned tree does not perform any differently than the original classification tree, but it did illustrate a simply tree that cuts down the number of predictors needed to create an equally accurate model.

## Bagging

set.seed(5)  
bag.student <- randomForest(Target ~ ., data = student.data, subset = train,  
 mtry = length(sig\_predictors)-1, importance = T)  
bag.student

##   
## Call:  
## randomForest(formula = Target ~ ., data = student.data, mtry = length(sig\_predictors) - 1, importance = T, subset = train)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 20  
##   
## OOB estimate of error rate: 13.52%  
## Confusion matrix:  
## Dropout Student class.error  
## Dropout 505 201 0.28470255  
## Student 98 1408 0.06507304

bag.pred <- predict(bag.student, student.data[test, ], type = "class")  
table(bag.pred, student.data$Target[test])

##   
## bag.pred Dropout Student  
## Dropout 512 95  
## Student 203 1402

bag.error <- mean(bag.pred != student.data$Target[test])  
bag.error

## [1] 0.1347197

model.errors[model.errors$model == "Bagging", "test.error"] <- bag.error

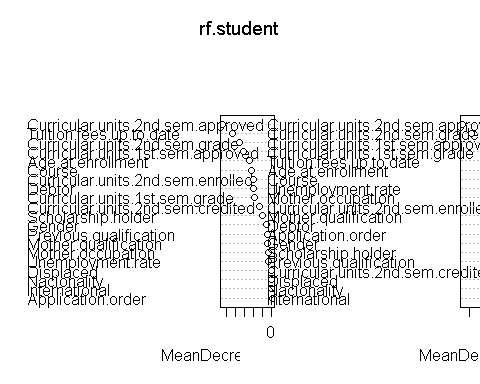
A bagging model is used to predict whether a student will dropout or not. The bagging model is a random forest that uses all the predictors in the data set. The model is trained using the training data set. The out-of-bag observations estimated a test error rate of 13.5%. Despite using bootstrap procedure, the test error rate is equal to that of the original and pruned classification trees.

## Random Forest

# Random Forest using default sqrt(p) predictors  
  
set.seed(5)  
rf.student <- randomForest(Target ~ ., data = student.data, subset = train,  
 importance = T)  
rf.student

##   
## Call:  
## randomForest(formula = Target ~ ., data = student.data, importance = T, subset = train)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 12.79%  
## Confusion matrix:  
## Dropout Student class.error  
## Dropout 511 195 0.27620397  
## Student 88 1418 0.05843293

varImpPlot(rf.student)



rf.pred <- predict(rf.student, newdata = student.data[test, ], type = "class")  
rf.error <- mean(rf.pred != student.data$Target[test])  
rf.error

## [1] 0.1338156

model.errors[model.errors$model == "RF", "test.error"] <- rf.error

Using the training data set, a random forest is used the model the data. Similar to the bagging model, a number of decision trees are built using a bootstrap procedure. The random forest model uses the square root of the total number of predictors at each split rather than all of the predictors. A summary of the trained model shows that four predictors are tried at each split, much less than the 20 used in the bagging model. The forest contains 500 trees and the out-of-bag estimate of the error rate is 12.8%. This differs from the actual test error rate calculated using the test data set, which was 13.4%. The random forest does not perform considerably differently than the bagging model.

## Boosting

tunings <- c(0.001, 0.01, 0.2, 0.5, 0.75, 1.0)  
boost.results <- data.frame(shrinkage = tunings,  
 test.error = rep(NA, length(tunings)))  
student.data$Target <- as.numeric(student.data$Target)  
student.data$Target <- apply(student.data, 1, FUN = function(x) x[1] - 1)  
  
for (x in 1:length(tunings)) {  
 set.seed(6)  
 boost.student <- gbm(Target ~ ., data = student.data[train, ],  
 distribution = "bernoulli", n.trees = 1000,  
 interaction.depth = 7, shrinkage = tunings[x])  
 boost.pred <- predict(boost.student, newdata = student.data[test, ],   
 type = "response", n.trees = 1000)  
 boost.results[x, "test.error"] <- mean(boost.pred != student.data$Target[test])  
}  
  
boost.results

## shrinkage test.error  
## 1 0.001 1.0000000  
## 2 0.010 1.0000000  
## 3 0.200 0.8792948  
## 4 0.500 1.0000000  
## 5 0.750 0.1957505  
## 6 1.000 0.1957505

model.errors[model.errors$model == "Boosting", "test.error"] <- min(boost.results$test.error)

Boosting models with various shrinkage parameters were trained using the student dropout data. Each boosting model was created assuming a Bernoulli distribution of the response to account for the binary classification problem, 1000 trees, and an interaction depth of 7 to match that of the pruned classification tree. Small shrinkage parameters, 0.001, and 0.01, resulted in 100% test error rate. The lowest test error rate observed was from a shrinkage parameter of 0.750. That test error rate is equal to 19.6%.

## Comparison of Models

model.errors

## model test.error  
## 1 Log Regression 0.1238698  
## 2 LDA 0.1247740  
## 3 QDA 0.1532550  
## 4 NB Classifier 0.1627486  
## 5 KNN 0.2124774  
## 6 Tree 0.1374322  
## 7 Pruned Tree 0.1374322  
## 8 Bagging 0.1347197  
## 9 RF 0.1338156  
## 10 Boosting 0.1957505

Logistic regression and LDA yield the lowest test errors of all the models.

## Logistic Regression Using Even Less Predictors

# Reinitialize student data after boosting changes  
student.data <- na.omit(full.data)  
student.data$Target[student.data$Target=="Enrolled"] <- "Student"  
student.data$Target[student.data$Target=="Graduate"] <- "Student"  
student.data$Target <- as.factor(student.data$Target)  
student.data <- student.data[, sig\_predictors]  
  
log.final <- glm(Target ~ Curricular.units.2nd.sem.approved + Tuition.fees.up.to.date  
 + Age.at.enrollment + Curricular.units.2nd.sem.enrolled, data = student.data,  
 subset = train, family = binomial)  
summary(log.final)

##   
## Call:  
## glm(formula = Target ~ Curricular.units.2nd.sem.approved + Tuition.fees.up.to.date +   
## Age.at.enrollment + Curricular.units.2nd.sem.enrolled, family = binomial,   
## data = student.data, subset = train)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.374988 0.332232 -1.129 0.259   
## Curricular.units.2nd.sem.approved 0.782657 0.036030 21.723 < 2e-16 \*\*\*  
## Tuition.fees.up.to.date 2.493032 0.224053 11.127 < 2e-16 \*\*\*  
## Age.at.enrollment -0.053731 0.008827 -6.087 1.15e-09 \*\*\*  
## Curricular.units.2nd.sem.enrolled -0.479326 0.041541 -11.539 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 2770.5 on 2211 degrees of freedom  
## Residual deviance: 1499.4 on 2207 degrees of freedom  
## AIC: 1509.4  
##   
## Number of Fisher Scoring iterations: 5

log.probs <- predict(log.final, student.data[test, ], type = "response")  
log.pred <- rep("Student", nrow(student.data[test, ]))  
  
contrasts(student.data$Target)

## Student  
## Dropout 0  
## Student 1

log.pred[glm.probs < 0.5] <- "Dropout"  
table(log.pred, student.data$Target[test])

##   
## log.pred Dropout Student  
## Dropout 516 75  
## Student 199 1422

log.error <- mean(log.pred != student.data$Target[test])  
log.error

## [1] 0.1238698

Selecting fewer predictors did not change the test error rate for the logistic regression model.