Assignment 7

knitr::opts\_chunk$set(echo = TRUE)  
  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.0 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.1 ✔ tibble 3.1.8  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to become errors

library(randomForest)

## randomForest 4.7-1.1  
## Type rfNews() to see new features/changes/bug fixes.  
##   
## Attaching package: 'randomForest'  
##   
## The following object is masked from 'package:dplyr':  
##   
## combine  
##   
## The following object is masked from 'package:ggplot2':  
##   
## margin

library(caret)

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

library(gbm)

## Loaded gbm 2.1.8.1

library(pROC)

## Type 'citation("pROC")' for a citation.  
##   
## Attaching package: 'pROC'  
##   
## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

library(rpart.plot)

## Loading required package: rpart

library(rpart)

# Part 1

set.seed(123)  
hw7\_data = read\_csv("./mi.data.csv") %>%   
 janitor::clean\_names()

## Rows: 1700 Columns: 16  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## dbl (16): ID, Age, Sex, sodium, ALT, WBC, ESR, SBP, DBP, Pulm.adema, FC, Arr...  
##   
## 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.

#stripped id variable  
hw7\_data<-hw7\_data[,2:16]  
  
#changing variable types to appropriate ones  
hw7\_data$sex<-as.factor(hw7\_data$sex)  
hw7\_data$pulm\_adema<-as.factor(hw7\_data$pulm\_adema)  
hw7\_data$fc<-as.factor(hw7\_data$fc)  
hw7\_data$arr<-as.factor(hw7\_data$arr)  
hw7\_data$diab<-as.factor(hw7\_data$diab)  
hw7\_data$obesity<-as.factor(hw7\_data$obesity)  
hw7\_data$asthma<-as.factor(hw7\_data$asthma)  
hw7\_data$readmission<-as.factor(hw7\_data$readmission)  
  
#re-leveling outcome variable: readmission (readmitted to hospital within 30 days) so "0=No" is reference group  
hw7\_data$readmission<-relevel(hw7\_data$readmission, ref="0")  
  
#omitting any missing data  
hw7\_data<-na.omit(hw7\_data)

# Part 2 : Partition data into training and testing sets to 70/30 split

set.seed(123)  
  
training.data<-hw7\_data$readmission %>%   
 createDataPartition(p=0.7, list=F)  
  
#Training Data  
train.data<-hw7\_data[training.data, ]  
  
#Testing Data  
test.data<-hw7\_data[-training.data, ]

# Comparing Two Different Algorithms: Elastic Net vs. Random Forest

Elastic Net Model Model that chooses alpha and lambda via cross-validation using all of the features

set.seed(123)  
  
train.control<-trainControl(method="cv", number=10, sampling="down")  
  
elastic.model<- train(  
 readmission ~., data = train.data, method = "glmnet",  
 trControl = train.control, preProc=c("center", "scale"),  
 tuneLength=10  
 )  
  
elastic.model$bestTune

## alpha lambda  
## 56 0.7 0.02444849

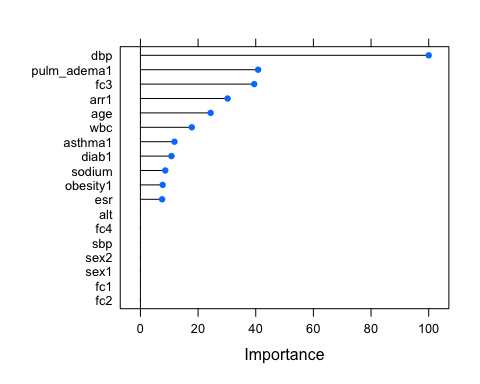
confusionMatrix(elastic.model)

## Cross-Validated (10 fold) Confusion Matrix   
##   
## (entries are percentual average cell counts across resamples)  
##   
## Reference  
## Prediction 0 1  
## 0 53.1 3.9  
## 1 37.5 5.5  
##   
## Accuracy (average) : 0.5861

varImp(elastic.model)

## glmnet variable importance  
##   
## Overall  
## dbp 100.000  
## pulm\_adema1 40.827  
## fc3 39.466  
## arr1 30.253  
## age 24.366  
## wbc 17.823  
## asthma1 11.815  
## diab1 10.777  
## sodium 8.618  
## obesity1 7.723  
## esr 7.518  
## sex2 0.000  
## fc2 0.000  
## fc4 0.000  
## sex1 0.000  
## alt 0.000  
## fc1 0.000  
## sbp 0.000

plot(varImp(elastic.model))



Random Forest Model Setting hyperparameters mtry (number of features eligible for each split)

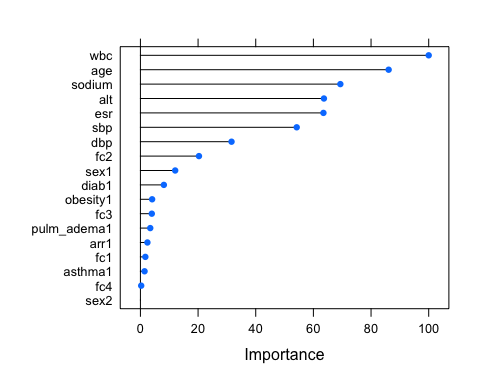
set.seed(123)  
  
control.settings<-trainControl(method="cv", number=10, sampling="down")  
  
#Producing various values of mtry  
mtry.vals<-c(ncol(train.data)-1, sqrt(ncol(train.data)-1), 0.5\*ncol(train.data)-1)  
mtry.grid<-expand.grid(.mtry=round(mtry.vals))  
  
rf\_model<-train(readmission ~., data=train.data, method="rf", metric="Accuracy", tuneGrid=mtry.grid, trControl=control.settings, ntree=200)  
  
confusionMatrix(rf\_model)

## Cross-Validated (10 fold) Confusion Matrix   
##   
## (entries are percentual average cell counts across resamples)  
##   
## Reference  
## Prediction 0 1  
## 0 50.9 4.1  
## 1 39.7 5.3  
##   
## Accuracy (average) : 0.5617

varImp(rf\_model)

## rf variable importance  
##   
## Overall  
## wbc 100.0000  
## age 86.1062  
## sodium 69.3524  
## alt 63.6537  
## esr 63.4856  
## sbp 54.2176  
## dbp 31.6060  
## fc2 20.3082  
## sex1 12.0616  
## diab1 8.1315  
## obesity1 4.0620  
## fc3 3.9470  
## pulm\_adema1 3.4119  
## arr1 2.4281  
## fc1 1.7131  
## asthma1 1.4360  
## fc4 0.2712  
## sex2 0.0000

plot(varImp(rf\_model))



# Evaluation Metrics with Final Model - Elastic Model

set.seed(123)  
  
#Make predictions in test set  
test.outcome<-predict(rf\_model, newdata=test.data)  
  
#Model Prediction Performance  
confusionMatrix(test.outcome, test.data$readmission, positive="1")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 243 20  
## 1 219 27  
##   
## Accuracy : 0.5305   
## 95% CI : (0.486, 0.5745)  
## No Information Rate : 0.9077   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.0346   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.57447   
## Specificity : 0.52597   
## Pos Pred Value : 0.10976   
## Neg Pred Value : 0.92395   
## Prevalence : 0.09234   
## Detection Rate : 0.05305   
## Detection Prevalence : 0.48330   
## Balanced Accuracy : 0.55022   
##   
## 'Positive' Class : 1   
##