Data Mining HW2 Final Code

Blue Team 16

Contents

Data Reading

```r
# Reading in data
trainingBin <- read.csv("insurance_t_bin.csv")
training <- read.csv("insurance_t.csv")

validationBin <- read.csv("insurance_v_bin.csv")
validation <- read.csv("insurance_v.csv")

# Fixing Separations and NAs
trainingBin <- trainingBin %>% mutate(across(everything(), ~ as.character(.x))) %>%
  mutate(across(everything(), ~ replace_na(.x,"M"))) %>%
  mutate(across(everything(), ~ as.factor(.x)))

validationBin <- validationBin %>% mutate(across(everything(), ~ as.character(.x))) %>%
  mutate(across(everything(), ~ replace_na(.x,"M"))) %>%
  mutate(across(everything(), ~ as.factor(.x)))
```

Old Logistic Regression Model

```r
finalModel <- glm(INS ~ NSF + MTG + INV + ILSBAL_BIN + IRA + DDA + TELLER_BIN + CC + ATMAMT_BIN + CHECKS
```

Decision Tree Models

```r
# Making a large tree to prune later. The values I selected are what I came to after playing around wit

# Only LR variables
lrTree <- rpart(INS ~ NSF + MTG + INV + ILSBAL + IRA + DDA + TELLER + CC + ATMAMT + CHECKS + MMBAL + CDB
                    control = rpart.control(minsplit = 30, cp = .001, maxdepth = 6))
# All variables
bigTree <- rpart(INS ~ ., data=training, method='class',parms = list(split="gini"),
                       control = rpart.control(minsplit = 30, cp = .001, maxdepth = 6))
```

Pruning

Subset Variable Model

```r
printcp(lrTree)
```

```
##
## Classification tree:
## rpart(formula = INS ~ NSF + MTG + INV + ILSBAL + IRA + DDA +
##     TELLER + CC + ATMAMT + CHECKS + MMBAL + CDBAL + DDABAL +
##     SAVBAL, data = training, method = "class", parms = list(split = "gini"),
##     control = rpart.control(minsplit = 30, cp = 0.001, maxdepth = 6))
##
## Variables actually used in tree construction:
##  [1] ATMAMT CDBAL  CHECKS DDA    DDABAL INV    IRA    MMBAL  MTG    SAVBAL
## [11] TELLER
##
## Root node error: 2918/8495 = 0.3435
##
## n= 8495
##
##           CP nsplit rel error  xerror     xstd
## 1  0.1329678      0   1.00000 1.00000 0.014999
## 2  0.0277587      1   0.86703 0.87286 0.014472
## 3  0.0099383      2   0.83927 0.83893 0.014306
## 4  0.0056546      5   0.80946 0.83208 0.014271
## 5  0.0054832     10   0.78033 0.81905 0.014203
## 6  0.0049692     11   0.77485 0.81905 0.014203
## 7  0.0035984     13   0.76491 0.81494 0.014181
## 8  0.0032557     15   0.75771 0.80089 0.014105
## 9  0.0027416     17   0.75120 0.79164 0.014054
## 10 0.0023989     18   0.74846 0.80158 0.014109
## 11 0.0020562     19   0.74606 0.80089 0.014105
## 12 0.0017135     23   0.73783 0.79609 0.014079
## 13 0.0010281     24   0.73612 0.79678 0.014083
## 14 0.0010000     28   0.73201 0.79781 0.014088
```

Only want to include first 6 layers based on oneSE

```r
lrTree <- prune(lrTree,cp=0.0049692)
```

Full Variable Model Tree

```
printcp(bigTree)
```

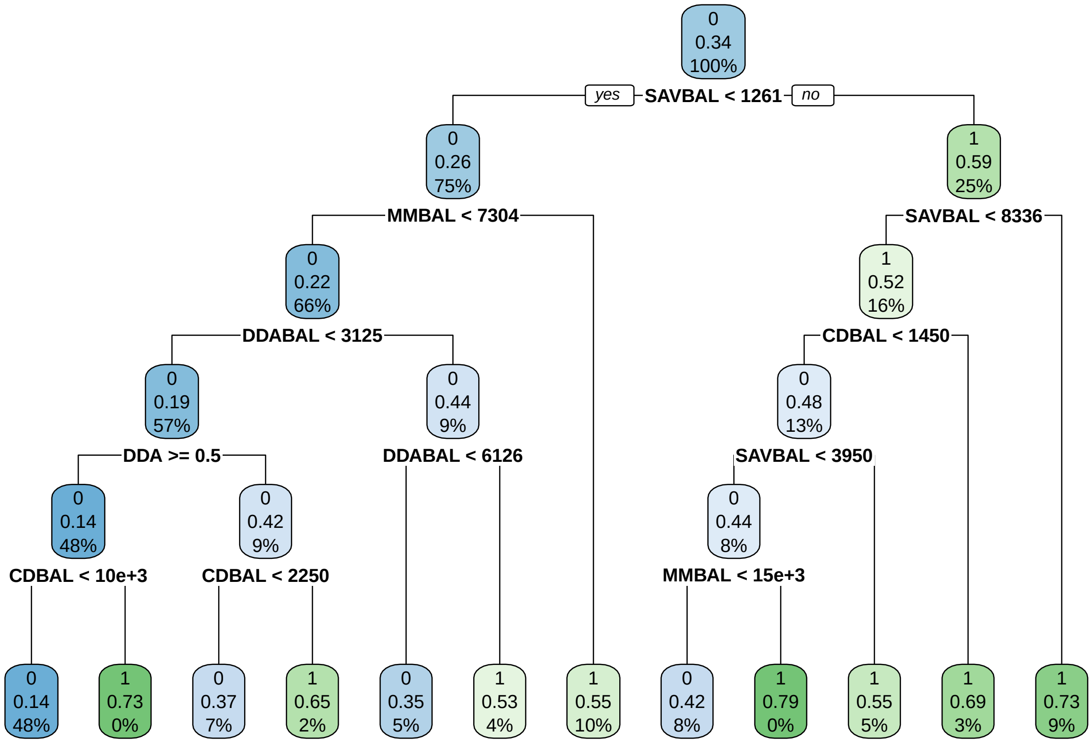
```
##
## Classification tree:
## rpart(formula = INS ~ ., data = training, method = "class", parms = list(split = "gini"),
##     control = rpart.control(minsplit = 30, cp = 0.001, maxdepth = 6))
##
## Variables actually used in tree construction:
##  [1] ACCTAGE ATMAMT  BRANCH  CDBAL   CHECKS  CRSCORE DDA     DDABAL  DEP
## [10] MM      SAVBAL
##
## Root node error: 2918/8495 = 0.3435
##
## n= 8495
##
##           CP nsplit rel error  xerror     xstd
## 1  0.1329678      0   1.00000 1.00000 0.014999
## 2  0.0277587      1   0.86703 0.87217 0.014469
## 3  0.0119945      2   0.83927 0.83413 0.014281
## 4  0.0111378      3   0.82728 0.82865 0.014253
## 5  0.0090816      5   0.80500 0.82522 0.014235
## 6  0.0065798      7   0.78684 0.82557 0.014237
## 7  0.0065113     12   0.75394 0.81734 0.014194
## 8  0.0034270     13   0.74743 0.81220 0.014166
## 9  0.0030843     14   0.74400 0.80295 0.014117
## 10 0.0020562     15   0.74092 0.80295 0.014117
## 11 0.0017135     17   0.73681 0.80809 0.014144
## 12 0.0015422     19   0.73338 0.80912 0.014150
## 13 0.0013708     22   0.72824 0.80980 0.014154
## 14 0.0010281     24   0.72550 0.81323 0.014172
## 15 0.0010000     25   0.72447 0.82180 0.014217
```

Only want first 10 layers

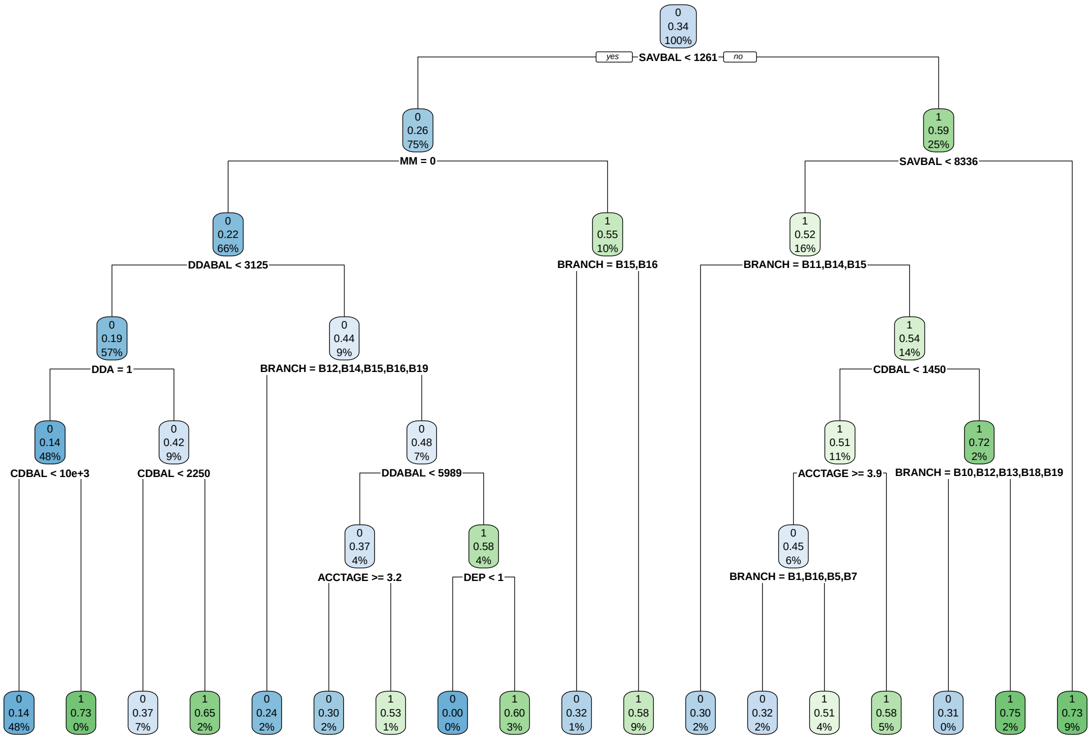
```
bigTree <- prune(bigTree,cp=0.0020562)
```

Visualizing

Subset Variable Model



Full Variable Model



Accuracy scores

Predictions and Fitted Values

```r
probLRTree <- predict(lrTree,validation,type = "prob")
probBigTree <- predict(bigTree,validation,type = "prob")

predLRTree <- predict(lrTree,validation,type = "class")
predBigTree <- predict(bigTree,validation,type = "class")

fittedLRTree <- predict(lrTree,training,type = "prob")
fittedBigTree <- predict(bigTree,training,type = "prob")
```

Subset Model

```r
lrAccuracy <- (length((which(predLRTree == validation$INS))) / nrow(validation))

lrAccuracy
```

```
## [1] 0.7123352
```

Full Variable Model

```r
bigAccuracy <- (length((which(predBigTree == validation$INS))) / nrow(validation))

bigAccuracy
```

```
## [1] 0.7306968
```

Logistic Regression Accuracy

```r
# Taken from logistic regression ROC curve
cutoff <-  0.2970672
pred <- predict(finalModel,validationBin,type = "response")
pred <- data.frame(pred = pred) %>% mutate(pred = if_else(pred > cutoff,1,0))
pred <- pred$pred

# Create accuracy vector
accDF <- data.frame(pred = pred, observed = validation$INS) %>% mutate(accuracy = if_else(pred == observ

accuracy <- round(mean(accDF$accuracy),4)
# Accuracy
accuracy
```

```
## [1] 0.702
```

Seeing as how the full variable tree outperforms both the logistic regression and the model built on a smaller set of variables, we should go forward with the full variable model tree as it maximizes accuracy with only a slight tradeoff of complexity.