Project 3

Random Forest

Brooke O’Donnell

October, 25th 2020

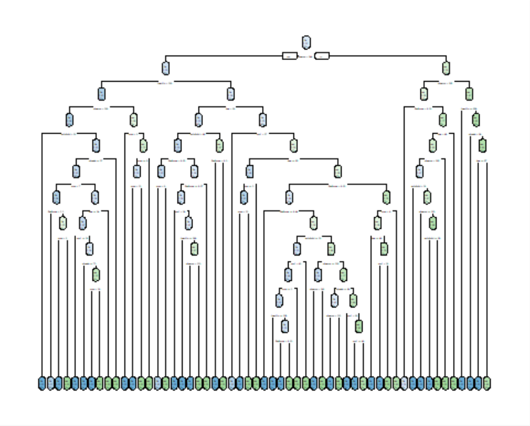
Predict the values of the “class” variable using a sample of women who were diagnosed with or without diabetes. The target variable is “class”. The feature variables are (1) Pregnancy (2) Plasma (3) Bloodp (4) Skinfold (5) Insulin (6) BMI (7) Pedigree (8) Age. Please prepare your data you see fit and split the data set into training and testing sets with 70% and 30% respectively before you run the models.

```
project3<-read.csv("Diabetes.csv", header=TRUE, sep=",", fileEncoding="UTF-8-BOM")
drop3<-na.omit(project3)

drop3$Class<-as.factor(drop3$Class)
drop3<-drop3%>%dplyr::mutate(Class=recode_factor(Class, "0"="No", "1"="Yes"))
set.seed(123)
split<-sample.split(drop3$Class, SplitRatio=0.7)
dia.train<-subset(drop3, split==TRUE)
dia.test<-subset(drop3, split==FALSE)
```

1. Please estimate a decision tree to predict the values of “Class” by including all feature variables and setting cp=0.005 (this is close to the best model). Then cross-validate the tree model using the testing set, record and report the indexes of accuracy, sensitivity, specificity, and kappa.

```
d.tree<-rpart(Class~.,
          data=dia.train,
          method="class",
          minsplit=2,
          minbucket=1,
          cp=0.005)
printcp(d.tree)

summary(d.tree)
```

```
rpart.plot(d.tree)
```



```
test.pred<-predict(d.tree, dia.test, type="class")
confusionMatrix(test.pred, (dia.test$Class))

## Confusion Matrix and Statistics
##           Reference
## Prediction  No Yes
##        No  112  34
##        Yes  38  46
##                Accuracy : 0.687
##                  95% CI : (0.6227, 0.7463)
##     No Information Rate : 0.6522
##     P-Value [Acc > NIR] : 0.1494
##                   Kappa : 0.318
##  Mcnemar's Test P-Value : 0.7237
##             Sensitivity : 0.7467
##             Specificity : 0.5750
##        'Positive' Class : No
```

- The simple decision tree model reported an accuracy of 69%, a sensitivity of 75%, a specificity of 58% and a kappa of 32%. A researcher would use random forest methods to improve these indexes.

2. Estimates 3 random forest models with 501 trees by setting mtry=3, 4, 5, respectively and including all feature variables. Cross-validate the 3 models using the testing set. Based on the three key indexes of accuracy, sensitivity, and specificity, which model is optimal? (Note: you have to argue for your choice)

```
set.seed(123)
system.time(rf3<-randomForest(Class~., data=dia.train, mtry= 3, ntree=501))

##    user  system elapsed
##    0.19    0.00    0.18

set.seed(123)
system.time(rf4<-randomForest(Class~., data=dia.train, mtry= 4, ntree=501))

##    user  system elapsed
##    0.17    0.00    0.17

set.seed(123)
system.time(rf5<-randomForest(Class~., data=dia.train, mtry= 5, ntree=501))

##    user  system elapsed
##    0.17    0.00    0.17

set.seed(123)
test.pred.3<-predict(rf3, dia.test, type="class")
confusionMatrix(test.pred.3, as.factor(dia.test$Class))

## Confusion Matrix and Statistics
##           Reference
## Prediction  No Yes
##        No  122  38
##        Yes  28  42
##                Accuracy : 0.713
##                  95% CI : (0.6499, 0.7706)
##     No Information Rate : 0.6522
##     P-Value [Acc > NIR] : 0.02944
##                   Kappa : 0.3485
##  Mcnemar's Test P-Value : 0.26794
##             Sensitivity : 0.8133
##             Specificity : 0.5250
##        'Positive' Class : No
```

```
set.seed(123)
test.pred.4<-predict(rf4, dia.test, type="class")
confusionMatrix(test.pred.4, as.factor(dia.test$Class))
```

```
## Confusion Matrix and Statistics
##           Reference
## Prediction  No Yes
##        No  121  36
##        Yes  29  44
##                Accuracy : 0.7174
##                  95% CI : (0.6545, 0.7746)
##     No Information Rate : 0.6522
##     P-Value [Acc > NIR] : 0.02107
##                   Kappa : 0.3641
##  Mcnemar's Test P-Value : 0.45675
##             Sensitivity : 0.8067
##             Specificity : 0.5500
##        'Positive' Class : No
```

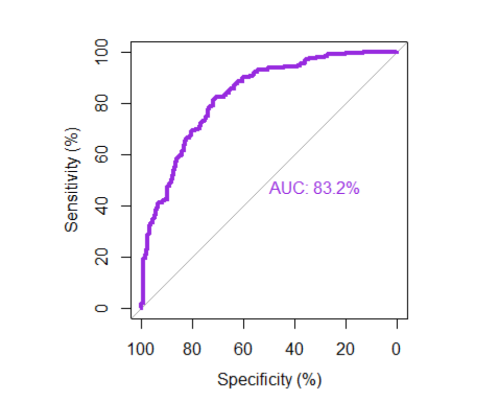
```
set.seed(123)
test.pred.5<-predict(rf5, dia.test, type="class")
confusionMatrix(test.pred.5, as.factor(dia.test$Class))
```

```
## Confusion Matrix and Statistics
##           Reference
## Prediction  No Yes
##        No  122  37
##        Yes  28  43
##                Accuracy : 0.7174
##                  95% CI : (0.6545, 0.7746)
##     No Information Rate : 0.6522
##     P-Value [Acc > NIR] : 0.02107
##                   Kappa : 0.3603
##
##  Mcnemar's Test P-Value : 0.32106
##             Sensitivity : 0.8133
##             Specificity : 0.5375
##       Balanced Accuracy : 0.6754
##        'Positive' Class : No
```

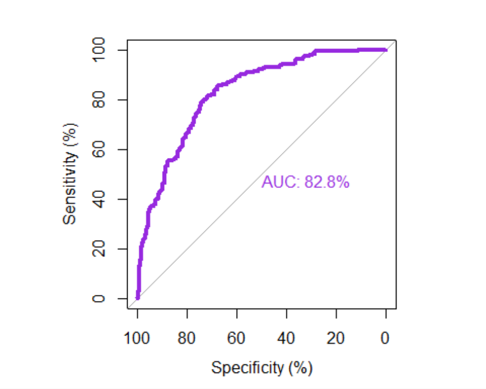
- Based on the three key indexes of accuracy, sensitivity and specificity I would pick the model that uses mtry=4. I would pick this model because it reported increased accuracy at 72% and an increased sensitivity at 81%. More importantly I would pick this model because it reported the highest specificity at 55%. I would argue that this is the most important index because I want a model that is good a detecting or predicting women with diabetes.

3. Create a ROC curve and obtain the AUC value from the optimal random forest model you decide in the previous step.

```
par(pty="s")
plot.roc(dia.train$Class,
        rf3$votes[,1],
        percent=TRUE,
        col="#972ADF",        lwd=4,
        print.auc=TRUE)
```



```
par(pty="s")
plot.roc(dia.train$Class,
        rf4$votes[,1],
        percent=TRUE,
        col="#972ADF",
        lwd=4,
        print.auc=TRUE)
```



- Out of the three randon forest models mtry=3 produced the highest AUC at 83.2%. The model I chose had an AUC slightly reduced at 82.8%.

4. Compare your optimal random forest model with your decision tree. Which performs better? Please justify your answer.

```
test.pred<-predict(d.tree, dia.test, type="class")
confusionMatrix(test.pred, (dia.test$Class))
```

```
## Confusion Matrix and Statistics
##           Reference
## Prediction  No Yes
##        No  112  34
##        Yes  38  46
##                Accuracy : 0.687
##                  95% CI : (0.6227, 0.7463)
##     No Information Rate : 0.6522
##     P-Value [Acc > NIR] : 0.1494
##                   Kappa : 0.318
##  Mcnemar's Test P-Value : 0.7237
##             Sensitivity : 0.7467
##             Specificity : 0.5750
##        'Positive' Class : No
```

```
set.seed(123)
test.pred.4<-predict(rf4, dia.test, type="class")
confusionMatrix(test.pred.4, as.factor(dia.test$Class))
```

```
## Confusion Matrix and Statistics
##           Reference
## Prediction  No Yes
##        No  121  36
##        Yes  29  44
##                Accuracy : 0.7174
##                  95% CI : (0.6545, 0.7746)
##     No Information Rate : 0.6522
##     P-Value [Acc > NIR] : 0.02107
##                   Kappa : 0.3641
##  Mcnemar's Test P-Value : 0.45675
##             Sensitivity : 0.8067
##             Specificity : 0.5500
##        'Positive' Class : No
```

- Compared to the simple decision tree model the random forest model preformed slightly better. The accuracy increased from 69% to 72%, the Kappa increased from 32% to 36%, and the sensitivity increased from 75% to 81%. The specificity decreased which is not ideal. The decision tree model had a specificity of 58% while the random forest model produced a 55% specificity.

5. Please tune the hyperparameter of “mtry” using either the “caret” or the “tidymodels” package by setting your number of trees to 501. If your decision is solely based on accuracy, what “mtry” yields the optimal model? Please show your evidence.

```
rfcontrol<-trainControl(method = "repeatedcv",
                        number = 501,
                        repeats = 3)
```

```
rf.final<-train(Class~., data=dia.train, method="rf", trControl=rfcontrol)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

```
rf.final
```

```
## Random Forest
## 538 samples
##   8 predictor
##   2 classes: 'No', 'Yes'
## No pre-processing
## Resampling: Cross-Validated (501 fold, repeated 3 times)
## Summary of sample sizes: 536, 536, 537, 537, 537, 537, ...
## Resampling results across tuning parameters:
##   mtry  Accuracy   Kappa
##   2     0.7761194  0.3161634
##   5     0.7711443  0.3008850
##   8     0.7649254  0.2954545
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 2.
```

- After tuning the hyperparameter based on accuracy, mtry=2 yields to be the optimal model at 78%.