

DT Scrum cycle Exercise

Phillip Llewellyn

7/19/2021

Decision Tree Scrum Cycle 7

Step 1

- Load data and get summaries

```
data <-read.csv("BankLoan Dataset 2021 - Clean .csv") #HR.csv
#str(data)
summary(data)
```

```
##           X.1           X           RefNum           agerange
## Min.      : 1.0    Min.      : 1.0    Min.      :10023467    18-30 : 97
## 1st Qu.: 281.5    1st Qu.: 281.5    1st Qu.:10023748    31-40 :430
## Median : 562.0    Median : 562.0    Median :10024028    41-60 :594
## Mean      : 562.0    Mean      : 562.0    Mean      :10024028    over 60: 2
## 3rd Qu.: 842.5    3rd Qu.: 842.5    3rd Qu.:10024308
## Max.      :1123.0    Max.      :1123.0    Max.      :10024589
##
##           age           job           marital           education
## Min.      :22.00    blue-collar:348    divorced:165    primary :198
## 1st Qu.:35.00    technician :167    married :709    secondary:626
## Median :42.00    admin.      :153    single  :249    tertiary :201
## Mean      :42.46    management :148                unknown  : 98
## 3rd Qu.:50.00    services    :145
## Max.      :61.00    retired     : 49
##           (Other)    :113
##           balance     housing     loan           month           date
## Min.      : -932.0    no :129    no :624    april      : 94    1/1/2018: 94
## 1st Qu.:   23.0    yes:994    yes:499    february: 94    2/1/2018: 94
## Median :  167.0                january : 94    3/1/2018: 94
## Mean      :  567.3                july      : 94    4/1/2018: 94
## 3rd Qu.:  446.0                june      : 94    5/1/2018: 94
## Max.      :58544.0                march     : 94    6/1/2018: 94
##           (Other) :559    (Other) :559
##           duration     deposit     lead           product
## Min.      : 2.0    Min.      : 1.80    Min.      :0.0000    auto      :217
## 1st Qu.: 130.0    1st Qu.: 43.75    1st Qu.:0.0000    business  : 25
## Median : 203.0    Median : 52.94    Median :1.0000    mortgage  :312
## Mean      : 270.5    Mean      : 84.09    Mean      :0.6883    no product:350
## 3rd Qu.: 315.5    3rd Qu.: 92.71    3rd Qu.:1.0000    unsecured :219
```

```
## Max. :2177.0 Max. :388.68 Max. :1.0000
##
## qualified contacted won loanvalue
## Min. :0.0000 Min. :-1.0000 Min. :0.0000 Min. : 1526
## 1st Qu.:0.0000 1st Qu.: 0.0000 1st Qu.:0.0000 1st Qu.: 3397
## Median :1.0000 Median : 0.0000 Median :0.0000 Median : 6530
## Mean :0.6073 Mean : 0.2787 Mean :0.4426 Mean : 5991
## 3rd Qu.:1.0000 3rd Qu.: 1.0000 3rd Qu.:1.0000 3rd Qu.: 7632
## Max. :1.0000 Max. : 1.0000 Max. :1.0000 Max. :12353
##
## NPS contacted_and_won qualified_and_contacted lead_and_qualified
## Min. : 3.000 Min. :0.0000 Min. :-1.0000 Min. :0.000
## 1st Qu.: 7.000 1st Qu.:0.0000 1st Qu.: 0.0000 1st Qu.:1.000
## Median : 7.000 Median :0.0000 Median : 0.0000 Median :1.000
## Mean : 7.874 Mean :0.4426 Mean : 0.3419 Mean :0.919
## 3rd Qu.: 9.000 3rd Qu.:1.0000 3rd Qu.: 1.0000 3rd Qu.:1.000
## Max. :10.000 Max. :1.0000 Max. : 1.0000 Max. :1.000
##
```

```
#data$left <- as.factor(data$left)
#str(data)
```

Step 2

- Split data into training and testing data

```
set.seed(21)
newDataset <- sample.split(Y=data$X, SplitRatio = 0.7)
trainData <- data[newDataset,]
testData <- data[!newDataset,]
```

Step 3

- Fit a Decision Tree using training data

```
# The . specifies all other columns ( Class ~ . )
DTmodel <- rpart(won ~ age + job + education , method="class", data=trainData, parms = list (split ="in
```

- Target Variable = Class,
- Input Variables = All,
- split =gini or information gain
- control = rpart.control for prepruning DT minsplit- min records at node for split to occur, maxdepth - depth of the DT
- Fitting the model

```
rpart.plot(DTmodel, type=3, extra = 2, fallen.leaves = F, cex = 0.8)
```

job = admin.,blue-collar,housemaid,management,retired,self-employed,technician,unemployed
entrepreneur,services,student,unknown



```
#try extra with 2,8,4, 101
```

- Print out the information

```
#(DTmodel) # detailed summary of splits
DTmodel #prints the rules
```

```
## n= 786
##
## node), split, n, loss, yval, (yprob)
##      * denotes terminal node
##
## 1) root 786 358 0 (0.5445293 0.4554707)
##   2) job=admin.,blue-collar,housemaid,management,retired,self-employed,technician,unemployed 647 280
##   3) job=entrepreneur,services,student,unknown 139 61 1 (0.4388489 0.5611511) *
```

- Run the second model

```
#DTmodel2 <- J48(as.factor(Class) ~., trainData, control = Weka_control(R = TRUE, M = round(NROW(trainD
#DTmodel2 <- J48(as.factor(left) ~., trainData, control = Weka_control(R = TRUE, M = 50))
#IGDT5model <- J48(as.factor(eReader_Adoption)~., trainData ,control = Weka_control(R = TRUE, M = round
#IGDT10model <- J48(as.factor(eReader_Adoption)~., trainData ,control = Weka_control(R = TRUE, M = roun
```

- Plot the model

```
#plot(DTmodel)
```

Step 4

- Use the fitted model to do predictions for the test data

```
predTest <- predict(DTmodel, testData, type="class")
probTest <- predict(DTmodel, testData, type="prob")

actualTest <- testData$won
```

Step 5

- Create Confusion Matrix and compute the misclassification error

```
t1 <- table(predictions=predTest, actual = actualTest)
t1 # Confusion matrix
```

```
##           actual
## predictions  0   1
##           0 163 116
##           1  35  23
```

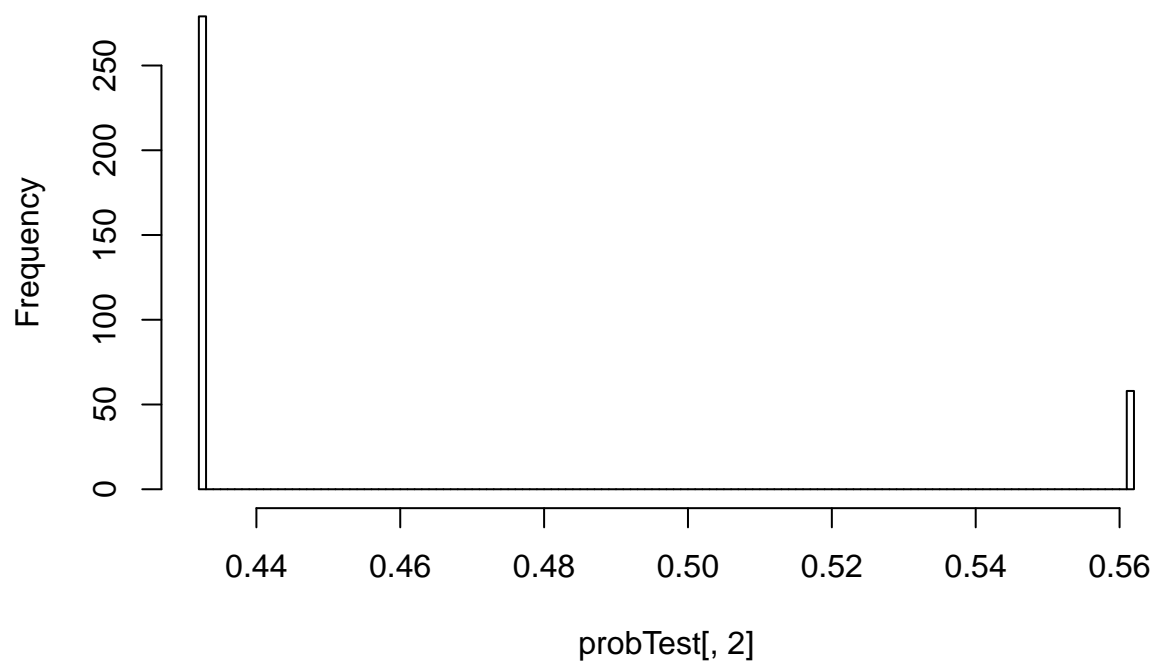
```
accuracy1 <- sum(diag(t1))/sum(t1)
accuracy1
```

```
## [1] 0.5519288
```

- Visualization of probabilities

```
hist(probTest[,2], breaks = 100)
```

Histogram of probTest[, 2]



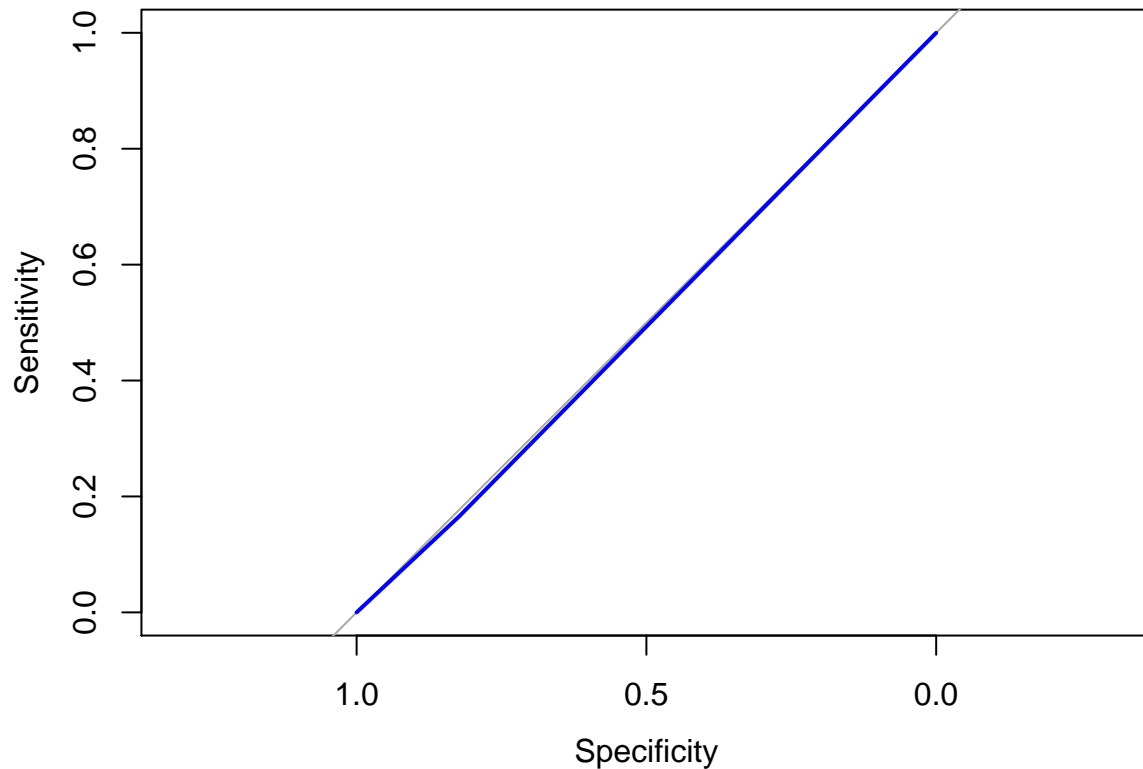
- ROC and Area Under the Curve

```
ROC <- roc(actualTest, probTest[,2])
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot(ROC, col="blue")
```



```
AUC <- auc(ROC)
AUC
```

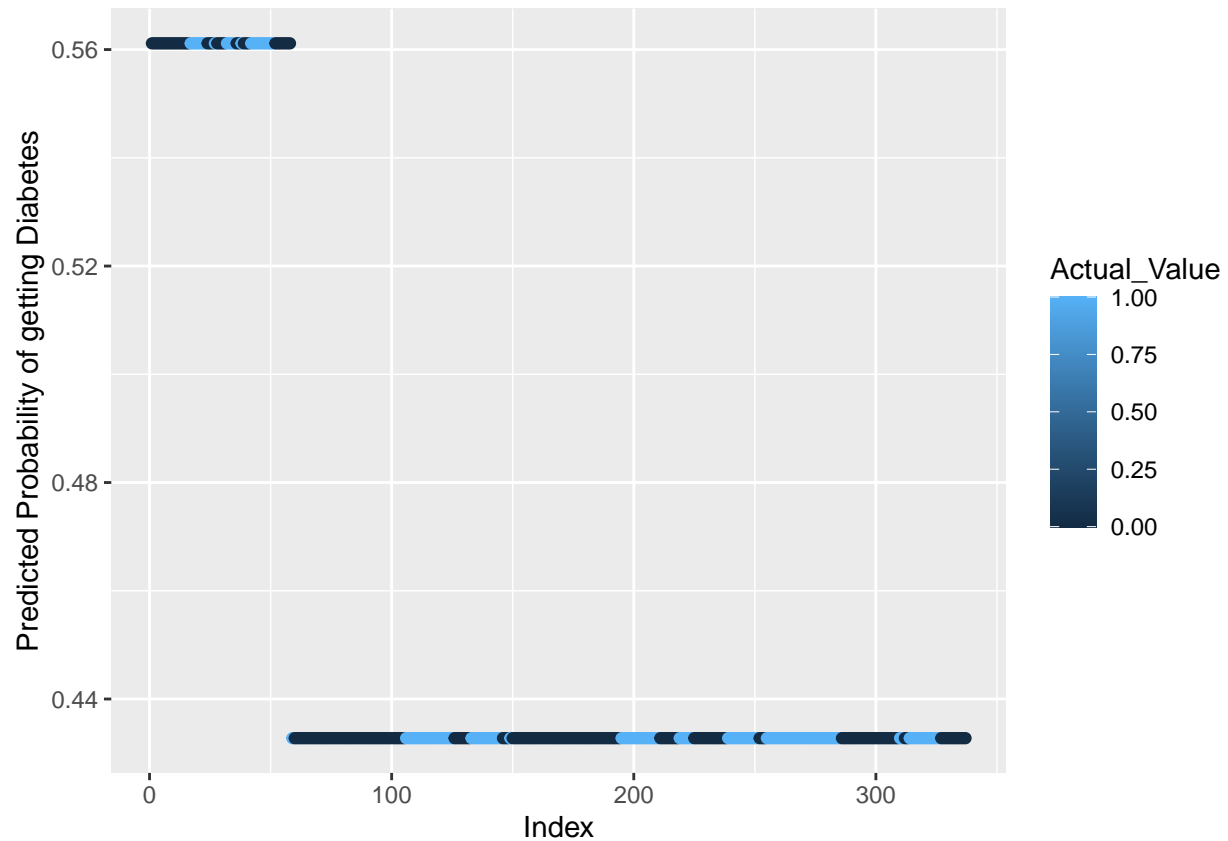
```
## Area under the curve: 0.4943
```

- A new dataframe with Predicted Prob, Actual Value and Predicted Value

```
predicted_data <- data.frame(Probs = probTest, Actual_Value= actualTest ,Predicted_Value = predTest )
#predicted_data$Probs.0 <- Class 0 Probability
#predicted_data$Probs.1 <- Class 1 Probability
predicted_data <- predicted_data[order(predicted_data$Probs.1, decreasing=TRUE),] # Sort on Probability
predicted_data$Rank <- 1:nrow(predicted_data) # Add a new variable rank
```

- plot the graph

```
ggplot(data=predicted_data, aes(x=Rank, y=Probs.1)) +
  geom_point(aes(color = Actual_Value)) + xlab("Index") + ylab("Predicted Probability of getting Diabet")
```



Step 6

- Use model to make predictions on newdata. Note we can specify the newData as data.frame with one or many records

```
#newData <- data.frame(Nbr_Preg = 4 , Glucose_test = 100, Triceps_SF=40,BP =95, S_insulin = 150, BMI= 30)

#predProbability <-predict(DTmodel, newData, type='prob')
#predProbability

## Performnce measures -
#setseed(1), gini
# Simplicity = 15 leaves
# Accuracy = 0.734
# AUC = 0.7627

#setseed(1), information
# Simplicity = 10 leaves
# Accuracy = 0.71
# AUC = 0.7834
```

Step 7

- EXAMINING STABILITY - Creating Decile Plots for Class 1 or 0 Sort

```

#-----Create empty df-----
#decileDF<- data.frame(matrix(ncol=3,nrow = 0))
#colnames(decileDF)<- c("Decile","per_correct_preds","No_correct_Preds","cum_preds")
#-----Initialize variables
#num_of_deciles=10
#Obs_per_decile<-nrow(predicted_data)/num_of_deciles
#decile_count=1
#start=1
#stop=(start-1) + Obs_per_decile
#prev_cum_pred<-0
#x=0
#-----Loop through DF and create deciles
#while (x < nrow(predicted_data)) {
#  subset<-predicted_data[c(start:stop),]
#  correct_count<- ifelse(subset$Actual_Value==subset$Predicted_Value,1,0)
#  no_correct_Preds<-sum(correct_count,na.rm = TRUE)
#  per_correct_Preds<-(no_correct_Preds/Obs_per_decile)*100
#  cum_preds<-no_correct_Preds+prev_cum_pred
#  addRow<-data.frame("Decile"=decile_count,"per_correct_preds"=per_correct_Preds,"No_correct_Preds"=no
#  decileDF<-rbind(decileDF,addRow)
#  prev_cum_pred<-prev_cum_pred+no_correct_Preds
#  start<-stop+1
#  stop=(start-1) + Obs_per_decile
#  x<-x+Obs_per_decile
#  decile_count<-decile_count+1
#}
#-----Stability plot (correct preds per decile)
#plot(decileDF$Decile,decileDF$per_correct_preds,type = "l",xlab = "Decile",ylab = "Percentage of corre

```