R Notebook

Objective

Aim of this project is to compare the performance of association rule mining and decision tree techniques and explore their performance in a disease prediction dataset. Apriori algorithm in Association rule mining and Rpart models will be used for decision tree classification.

Dataset – Liver patient dataset

http://archive.ics.uci.edu/ml/datasets/ILPD+%28Indian+Liver+Patient+Dataset%29

Background of data

The dataset used in this study is obtained from the UCI Machine learning repository. (http://archive.ics.uci.edu/ml/datasets/ILPD+%28Indian+Liver+Patient+Dataset%29). It is used to classify between patients with liver disorder or not.

Dataset Characteristics: Multivariate

Attribute characteristics: Integer, Real

Date Donated: 2012/ 05/21

Number of instances: 583

Number of Attributes: 10

Missing values: N/A

Attributes

1. Age Age of the patient
2. Gender Gender of the patient
3. TB Total Bilirubin
4. DB Direct Bilirubin
5. Alkphos Alkaline Phosphotase
6. Sgpt Alamine Aminotransferase

7. Sgot Aspartate Aminotransferase
8. TP Total Protiens
9. ALB Albumin
10. A/G Ratio Albumin and Globulin Ratio
11. Selector field used to split the data into two sets (labeled by the experts)

Association rule mining on liver patient dataset

```
#Meenakshi Nagarajan
#nagarajan.12@wright.edu


library("dplyr")

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

#Load the data into 'df.liverpatient'
df.liverpatient <- read.csv("/Users/meenakshinagarajan/Desktop/Datamining/Ind
ian Liver Patient Dataset (ILPD).csv", header=TRUE)
str(df.liverpatient)

## 'data.frame':    583 obs. of  11 variables:
##  $ Age                         : int  65 62 62 58 72 46 26 29 17 55 ...
##  $ Gender                      : Factor w/ 2 levels "Female","Male": 1 2 2
2 2 2 1 1 2 2 ...
##  $ TB                          : num  0.7 10.9 7.3 1 3.9 1.8 0.9 0.9 0.9 0
.7 ...
##  $ DB                          : num  0.1 5.5 4.1 0.4 2 0.7 0.2 0.3 0.3 0.
2 ...
##  $ Alkphos.Alkaline.Phosphotase: int  187 699 490 182 195 208 154 202 202
290 ...
##  $ SGPT                        : int  16 64 60 14 27 19 16 14 22 53 ...
##  $ SGOT                        : int  18 100 68 20 59 14 12 11 19 58 ...
##  $ TP                          : num  6.8 7.5 7 6.8 7.3 7.6 7 6.7 7.4 6.8
...
##  $ ALB                         : num  3.3 3.2 3.3 3.4 2.4 4.4 3.5 3.6 4.1
3.4 ...
##  $ A.G.ratio                   : num  0.9 0.74 0.89 1 0.4 1.3 1 1.1 1.2 1
```

```
...
##  $ Selector.field              : int  1 1 1 1 1 1 1 1 2 1 ...
```

Identify levels to convert numerical variables into factors

```
for(i in 3:10){
wfact=cut(df.liverpatient[,i],pretty(df.liverpatient[,i],3))
print(colnames(df.liverpatient)[i])
print(table(wfact))
}
```

```
## [1] "TB"
## wfact
##  (0,20] (20,40] (40,60] (60,80]
##     565      16       1       1
## [1] "DB"
## wfact
##   (0,5]  (5,10] (10,15] (15,20]
##     535      30      15       3
## [1] "Alkphos.Alkaline.Phosphotase"
## wfact
##         (0,500]     (500,1e+03] (1e+03,1.5e+03] (1.5e+03,2e+03]
##             522              46               8               6
## (2e+03,2.5e+03]
##               1
## [1] "SGPT"
## wfact
##         (0,500]     (500,1e+03] (1e+03,1.5e+03] (1.5e+03,2e+03]
##             570               7               3               3
## [1] "SGOT"
## wfact
##     (0,2e+03] (2e+03,4e+03] (4e+03,6e+03]
##           581             1             1
## [1] "TP"
## wfact
##  (2,4]  (4,6]  (6,8] (8,10]
##     13    188    348     34
## [1] "ALB"
## wfact
## (0,2] (2,4] (4,6]
##    60   451    72
## [1] "A.G.ratio"
## wfact
## (0,1] (1,2] (2,3]
##   402   174     3
```

```
df.liverpatient$Selector.field<-as.factor(df.liverpatient$Selector.field)
df.liverpatient$Gender<-as.factor(df.liverpatient$Gender)
wfact=cut(df.liverpatient[,1],pretty(df.liverpatient[,1],3))
print(colnames(df.liverpatient)[1])
```

```
## [1] "Age"
```

```
print(table(wfact))
```

```
## wfact
##   (0,50] (50,100]
##      376      207
```

Divide the variables into categories

```
library(arules)
```

```
## Warning: package 'arules' was built under R version 3.4.2
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'arules'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
df.liverpatient[[ "TB"]] <- ordered(cut(df.liverpatient[[ "TB"]], c(0,20,40,60,80)),labels = c("Normal","High","Elevated","Very High"))
df.liverpatient[[ "DB"]] <- ordered(cut(df.liverpatient[[ "DB"]], c(0,5,10,15,20)),labels = c("Normal","High","Elevated","Very High"))
df.liverpatient[[ "Alkphos.Alkaline.Phosphotase"]] <- ordered(cut(df.liverpatient[[ "Alkphos.Alkaline.Phosphotase"]], c(0,500,1e+03,1.5e+03,2e+03,2.5e+03)),labels = c("Normal","Abnormal","High","Very high","Elevated"))
df.liverpatient[[ "SGPT"]] <- ordered(cut(df.liverpatient[["SGPT"]], c(0,500,1e+03,1.5e+03,2e+03)),labels = c("Normal","High","Elevated","Very high"))
df.liverpatient[[ "SGOT"]] <- ordered(cut(df.liverpatient[["SGOT"]], c(0,2e+03,4e+03,6e+03)),labels = c("Normal","High","Very High"))
df.liverpatient[[ "TP"]] <- ordered(cut(df.liverpatient[["TP"]], c(2,4,6,8,10)),labels = c("Very Low","Low","Normal","High"))
df.liverpatient[[ "ALB"]] <- ordered(cut(df.liverpatient[["ALB"]], c(0,2,4,6)),labels = c("Very Low","Low","Normal"))
df.liverpatient[["A.G.ratio"]] <- ordered(cut(df.liverpatient[["A.G.ratio"]], c(0,1,2,3)),labels = c("Low","Normal","High"))
df.liverpatient[["Age"]] <- ordered(cut(df.liverpatient[["Age"]], c(0,50,100)),labels = c("< 50","> 50"))
```

Coercing into transactions

```
LiverPatient<-as(df.liverpatient,"transactions")
```

Compute the timeliness of apriori algorithm and find all the rules with minimum support of 1% and confidence of 90%

```
system.time({rules<-apriori(LiverPatient,parameter = list(support = 0.01, con
fidence = 0.9))})
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.9    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 5
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[36 item(s), 583 transaction(s)] done [0.00s].
## sorting and recoding items ... [27 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 9 10

## Warning in apriori(LiverPatient, parameter = list(support = 0.01,
## confidence = 0.9)): Mining stopped (maxlen reached). Only patterns up to a
## length of 10 returned!

##  done [0.02s].
## writing ... [66419 rule(s)] done [0.02s].
## creating S4 object  ... done [0.03s].

##    user  system elapsed
##   0.437   0.031   0.214
```

Rules for liverpatient and not a liver patient with lift measure greater than 1

```
rulesNotaLiverPatient<-subset(rules,subset=rhs %in% "Selector.field=2" & lift
>1)
rulesLiverPatient<-subset(rules,subset=rhs %in% "Selector.field=1" & lift>1)
```

Compare rules for both sets with highest confidence

```
inspect(head(rulesNotaLiverPatient,n=3,by="confidence"))
inspect(head(rulesLiverPatient,n=3,by="confidence"))
```

```
##     lhs                                       rhs                    support c
onfidence     lift count
## [1] {SGPT=High}                           => {Selector.field=1} 0.01200686
1 1.401442     7
## [2] {Alkphos.Alkaline.Phosphotase=High} => {Selector.field=1} 0.01372213
1 1.401442     8
## [3] {DB=Elevated}                         => {Selector.field=1} 0.02572899
1 1.401442    15
```

In association rule mining, accuracy is computed using lift, support and confidence. From the output we can see that, if SGPT is above normal level (or) Alkphos.Alkaline.Phosphotase is high (or) DB is Elevated, then there is 90% likelihood of liver disorder.

However, the support value for all the 3 rules is less which means that not a large amount of similar transaction involve High Alkphos Alkaline Phosphotase or abnormal SGPT or elevated DB. Therefore, this association could be misleading. While the lift value is greater than 1 suggests that the lhs and rhs of the rule are positively correlated. They appear together more often than expected. This shows strong relation between high SGPT, high Alkphos.Alkaline.Phosphotase, elevated DB and presence of liver disorder.

We could also see that with 90% confidence, no rule exists to identify people without a liver disorder.

Decision tree classification

```
#Load the data into 'mydata'
mydata=read.csv(file="/Users/meenakshinagarajan/Desktop/Datamining/Indian Liv
er Patient Dataset (ILPD).csv",head=TRUE,sep=",")
head(mydata)
```

```
##   Age Gender   TB  DB Alkphos.Alkaline.Phosphotase SGPT SGOT  TP ALB
## 1  65 Female  0.7 0.1                          187   16   18 6.8 3.3
## 2  62   Male 10.9 5.5                          699   64  100 7.5 3.2
## 3  62   Male  7.3 4.1                          490   60   68 7.0 3.3
## 4  58   Male  1.0 0.4                          182   14   20 6.8 3.4
## 5  72   Male  3.9 2.0                          195   27   59 7.3 2.4
## 6  46   Male  1.8 0.7                          208   19   14 7.6 4.4
##   A.G.ratio Selector.field
## 1      0.90              1
## 2      0.74              1
## 3      0.89              1
## 4      1.00              1
## 5      0.40              1
## 6      1.30              1
```

```
str(mydata)
```

```
## 'data.frame':    583 obs. of  11 variables:
##  $ Age                         : int  65 62 62 58 72 46 26 29 17 55 ...
##  $ Gender                      : Factor w/ 2 levels "Female","Male": 1 2 2
```

```
2 2 2 1 1 2 2 ...
##  $ TB                          : num  0.7 10.9 7.3 1 3.9 1.8 0.9 0.9 0.9 0
.7 ...
##  $ DB                          : num  0.1 5.5 4.1 0.4 2 0.7 0.2 0.3 0.3 0.
2 ...
##  $ Alkphos.Alkaline.Phosphotase: int  187 699 490 182 195 208 154 202 202
290 ...
##  $ SGPT                        : int  16 64 60 14 27 19 16 14 22 53 ...
##  $ SGOT                        : int  18 100 68 20 59 14 12 11 19 58 ...
##  $ TP                          : num  6.8 7.5 7 6.8 7.3 7.6 7 6.7 7.4 6.8
...
##  $ ALB                         : num  3.3 3.2 3.3 3.4 2.4 4.4 3.5 3.6 4.1
3.4 ...
##  $ A.G.ratio                   : num  0.9 0.74 0.89 1 0.4 1.3 1 1.1 1.2 1
...
##  $ Selector.field              : int  1 1 1 1 1 1 1 1 2 1 ...
```

Converting Selector field to factors

```
mydata$Selector.field<-factor(mydata$Selector.field,levels=sort(unique(mydata
$Selector.field)))
str(mydata)
```

```
## 'data.frame':    583 obs. of  11 variables:
##  $ Age                         : int  65 62 62 58 72 46 26 29 17 55 ...
##  $ Gender                      : Factor w/ 2 levels "Female","Male": 1 2 2
2 2 2 1 1 2 2 ...
##  $ TB                          : num  0.7 10.9 7.3 1 3.9 1.8 0.9 0.9 0.9 0
.7 ...
##  $ DB                          : num  0.1 5.5 4.1 0.4 2 0.7 0.2 0.3 0.3 0.
2 ...
##  $ Alkphos.Alkaline.Phosphotase: int  187 699 490 182 195 208 154 202 202
290 ...
##  $ SGPT                        : int  16 64 60 14 27 19 16 14 22 53 ...
##  $ SGOT                        : int  18 100 68 20 59 14 12 11 19 58 ...
##  $ TP                          : num  6.8 7.5 7 6.8 7.3 7.6 7 6.7 7.4 6.8
...
##  $ ALB                         : num  3.3 3.2 3.3 3.4 2.4 4.4 3.5 3.6 4.1
3.4 ...
##  $ A.G.ratio                   : num  0.9 0.74 0.89 1 0.4 1.3 1 1.1 1.2 1
...
##  $ Selector.field              : Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1
1 2 1 ...
```

Split into training and testing data

```
set.seed(100)
#90-10 split
ind <- sample(2,nrow(mydata),replace=TRUE,prob=c(0.9,0.1))
```

```
training<-mydata[ind==1, ]
test <- mydata[ind==2, ]
```

Model formation

```
myModel <- Selector.field ~ .
```

Compute timeliness

```
library(rpart)
system.time({rpart(myModel, data=training, method="class")})
```

```
##    user  system elapsed
##   0.020   0.001   0.022
```

Print Rpart model

```
liverpatient_rpart_tree<-rpart(myModel, data=training, method="class")
print(liverpatient_rpart_tree)
```

```
## n= 520
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##    1) root 520 146 1 (0.71923077 0.28076923)
##      2) TB>=1.65 193  15 1 (0.92227979 0.07772021) *
##      3) TB< 1.65 327 131 1 (0.59938838 0.40061162)
##        6) Alkphos.Alkaline.Phosphotase>=203 128  34 1 (0.73437500 0.265625
00) *
##        7) Alkphos.Alkaline.Phosphotase< 203 199  97 1 (0.51256281 0.487437
19)
##         14) Age>=28.5 160  71 1 (0.55625000 0.44375000)
##           28) ALB< 2.15 9   1 1 (0.88888889 0.11111111) *
##           29) ALB>=2.15 151  70 1 (0.53642384 0.46357616)
##             58) Age< 51.5 97  39 1 (0.59793814 0.40206186)
##              116) TB< 0.65 13   2 1 (0.84615385 0.15384615) *
##              117) TB>=0.65 84  37 1 (0.55952381 0.44047619)
##                234) Age>=45.5 22   5 1 (0.77272727 0.22727273) *
##                235) Age< 45.5 62  30 2 (0.48387097 0.51612903)
##                  470) SGOT< 18.5 12   2 1 (0.83333333 0.16666667) *
##                  471) SGOT>=18.5 50  20 2 (0.40000000 0.60000000)
##                    942) A.G.ratio>=0.98 34  16 1 (0.52941176 0.47058824)
##                     1884) ALB< 3.55 15   2 1 (0.86666667 0.13333333) *
##                     1885) ALB>=3.55 19   5 2 (0.26315789 0.73684211) *
##                    943) A.G.ratio< 0.98 16   2 2 (0.12500000 0.87500000) *
##             59) Age>=51.5 54  23 2 (0.42592593 0.57407407)
##              118) A.G.ratio< 1.05 27  12 1 (0.55555556 0.44444444)
##                236) Alkphos.Alkaline.Phosphotase< 158.5 8   1 1 (0.8750000
```

```
0 0.12500000) *
##                 237) Alkphos.Alkaline.Phosphotase>=158.5 19   8 2 (0.421052
63 0.57894737) *
##               119) A.G.ratio>=1.05 27   8 2 (0.29629630 0.70370370) *
##          15) Age< 28.5 39  13 2 (0.33333333 0.66666667)
##            30) TP>=7.8 8   2 1 (0.75000000 0.25000000) *
##            31) TP< 7.8 31   7 2 (0.22580645 0.77419355) *
```

Predict on test data

```
predSelector <- predict(liverpatient_rpart_tree,newdata=test,type="class")
table(predSelector,test$Selector.field)

##
## predSelector  1  2
##             1 33 15
##             2  9  6
```

Accuracy calculation

```
cat("MisClassification Error Rate:",mean(as.character(predSelector) != as.cha
racter(test$Selector.field)))

## MisClassification Error Rate: 0.3809524
```

From the Rpart model, we could see that the misclassification error rate is 38% which is not very high. Also, in terms of timeliness rpart method perfom well. Therefore, decision tree is desirable in healthcare sector.

Conclusion

We found that, apriori rules on classifying patients into liver disorder or not could be misleading as there are no enough itemset to support the rule. We could assume that apriori rules could perform well in case of large dataset. Also, apriori algorithm does not always produce rules relating to the selector field.

Therefore in terms of timeliness and accuracy with respect to this dataset, we could conclude that rpart model performs well and it produces rules which are based on selector field. This study can be continued with a larger dataset and with more rule based mining and decision tree algorithm.

Reference

1) IHE-7510-01 - Data Mining – LatexSlides – RdataDecisionTree and RDataAssociationRule
2) https://archive.ics.uci.edu/ml/datasets.html