Assignment 4 Report

Student: Lina Mi @01377283

I. R commands used:

1. Load "votes.csv" dataset and store it into a local R variable "lenses":
   In order to load votes.csv as sparse matrix and use apriori analysis of arules package, it needs to load arules package first

```
>library(arules)
>votes<-read.transactions("M:/Data Mining/week4/votes.csv", sep=",")
```

2. Display the content of variable "votes"

```
> votes
transactions in sparse format with
 435 transactions (rows) and
 5 items (columns)
```

3. Display some basic information about the dataset using the "summary" R command

```
>summary(votes)
transactions as itemMatrix in sparse format with
 435 rows (elements/itemsets/transactions) and
 5 columns (items) and a density of 0.691954

most frequent items:
         y          n   democrat          ? republican    (Other)
       434        433        267        203        168          0

element (itemset/transaction) length distribution:
sizes
  2   3   4
  1 233 201

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2.00    3.00    3.00    3.46    4.00    4.00

includes extended item information - examples:
    labels
1        ?
2 democrat
3        n
```

4. Examine the frequency of *democrat* and *republican* using the "*itemFrequency*" R Command. Also plot these frequencies using "*itemFrequencyPlot*" R command.

```
> itemFrequency(votes)
        ?   democrat          n republican          y
0.4666667  0.6137931  0.9954023  0.3862069  0.9977011
```

5. Creating training dataset and testing dataset(training: 20, testing:4).

```
> set.seed(10203)
> train_sample<-sample(24,20, replace=FALSE)
```

```
> train_lenses<-lenses_attr[train_sample,]
>test_lenses<-lenses_attr[-train_sample,]
> train_lenses
     age    prescription     astigmatic    tear classification
 19   old nearsightedness non-astigmatic reduced           none
 21   old  farsightedness     astigmatic reduced           none
 3  young nearsightedness non-astigmatic reduced           none
 5  young  farsightedness     astigmatic reduced           none
 4  young nearsightedness non-astigmatic  normal           hard
 6  young  farsightedness     astigmatic  normal           soft
 22   old  farsightedness     astigmatic  normal           soft
 7  young  farsightedness non-astigmatic reduced           none
 16 adult  farsightedness non-astigmatic  normal           none
 8  young  farsightedness non-astigmatic  normal           hard
 13 adult  farsightedness     astigmatic reduced           none
 24   old  farsightedness non-astigmatic  normal           none
 17   old nearsightedness     astigmatic reduced           none
 2  young nearsightedness     astigmatic  normal           soft
 18   old nearsightedness     astigmatic  normal           none
 15 adult  farsightedness non-astigmatic reduced           none
 9  adult nearsightedness     astigmatic reduced           none
 12 adult nearsightedness non-astigmatic  normal           hard
 14 adult  farsightedness     astigmatic  normal           soft
 1  young nearsightedness     astigmatic reduced           none
 >test_lenses
     age    prescription     astigmatic    tear classification
 10 adult nearsightedness     astigmatic  normal           soft
 11 adult nearsightedness non-astigmatic reduced           none
 20   old nearsightedness non-astigmatic  normal           hard
 23   old  farsightedness non-astigmatic reduced           none
```

6. Train decision tree using C5.0 algorithm (C5.0 function) using training dataset, `train_lenses:`

```
> train_lenses$classification<-as.factor(train_lenses$classification)
> lenses_model<-C5.0(train_lenses[-5], train_lenses$classification)
> lenses_model

Call:
C5.0.default(x = train_lenses[-5], y = train_lenses$classification)

Classification Tree
Number of samples: 20
Number of predictors: 4

Tree size: 3

Non-standard options: attempt to group attributes

> summary(lenses_model)

Call:
C5.0.default(x = train_lenses[-5], y = train_lenses$classification)


C5.0 [Release 2.07 GPL Edition]        Sat Mar 03 11:52:20 2018
-------------------------------

Class specified by attribute `outcome'

Read 20 cases (5 attributes) from undefined.data
```

```
Decision tree:

tear = reduced: none (10)
tear = normal:
:...astigmatic = non-astigmatic: hard (5/2)
    astigmatic = astigmatic: soft (5/1)


Evaluation on training data (20 cases):

      Decision Tree
    ----------------
    Size      Errors

       3    3(15.0%)   <<


     (a)   (b)   (c)    <-classified as
    ----  ----  ----
       3                (a): class hard
       2    10     1    (b): class none
                   4    (c): class soft


  Attribute usage:

  100.00% tear
   50.00% astigmatic


Time: 0.0 secs
```

7. Make predictions on test dataset and using CrossTable to evaluate the prediction result of the trained decision tree model.

```
> lenses_pred<-predict(lenses_model, test_lenses)
> install.packages("gmodels")
> library(gmodels)
> CrossTable(test_lenses$classification, lenses_pred)
```

```
Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  4


                           | lenses_pred
test_lenses$classification |      hard |      none |      soft | Row Total |
---------------------------|-----------|-----------|-----------|-----------|
                      hard |         1 |         0 |         0 |         1 |
                           |     2.250 |     0.500 |     0.250 |           |
                           |     1.000 |     0.000 |     0.000 |     0.250 |
                           |     1.000 |     0.000 |     0.000 |           |
```

```
                           |      0.250 |      0.000 |      0.000 |            |
---------------------------|------------|------------|------------|------------|
                      none |          0 |          2 |          0 |          2 |
                           |      0.500 |      1.000 |      0.500 |            |
                           |      0.000 |      1.000 |      0.000 |      0.500 |
                           |      0.000 |      1.000 |      0.000 |            |
                           |      0.000 |      0.500 |      0.000 |            |
---------------------------|------------|------------|------------|------------|
                      soft |          0 |          0 |          1 |          1 |
                           |      0.250 |      0.500 |      2.250 |            |
                           |      0.000 |      0.000 |      1.000 |      0.250 |
                           |      0.000 |      0.000 |      1.000 |            |
                           |      0.000 |      0.000 |      0.250 |            |
---------------------------|------------|------------|------------|------------|
              Column Total |          1 |          2 |          1 |          4 |
                           |      0.250 |      0.500 |      0.250 |            |
---------------------------|------------|------------|------------|------------|
```

From the result of crosstable displayed above, it can be seen that the decision tree model we built based on training dataset performs very well on the testing dataset

24 Questions:

a. it easy or difficult to build the decision tree model?
Ans: it is very easy to build the decision tree model once the training dataset and testing dataset are ready, just use C5.0() function

b. Is it intuitive or hard to understand and interpret?
Ans: it is quite intuitive and easy to understand and interpret the decision tree model build on training dataset. From the output of `summary(lenses_model)`, we know that the decision tree model is 3 depth. It used 20 observations with 5 attributes as training data. the first split is based on attribute tear: if the value of attribute "tear"= "reduced", then 10 out of 20 observations are classified as "none"(patient should not be fitted with contact lenses) without any error, namely 10 observations are correctly classified as "none" ; if "tear"= "normal", then need to investigate the feature of "astigmatic", if value of feature "astigmatic" = "non-astigmatic", then 5 observations are classified as "hard" with 2 observations misclassified, namely, two observations with "classification" of other than "hard" is mistakenly classified into "hard"; if value of feature "astigmatic"= "astigmatic", then 5 observations are classified into "soft" class with 1 observation is misclassified. The total error rate is 15%

c. What are the possible decisions that the tree can make?
The decisions that the tree model can make include "none"(patient should not be fitted with contact lenses), "soft"(patient should be fitted with soft contact lenses) and "hard"(patient should be fitted with hard contact lenses)

d. What is the best-case scenario and what is the worst case scenario of using the model you generated?
Ans: The best-case scenario is using decision tree model I generated to correctly predict the classifications of unseen observations with rate of 100%, just like the situation of testing dataset, the generated decision tree model predicted the classification of observations on testing dataset with rate of 100%. The worst-case scenario is tree model built on training dataset could not provide

correct prediction on the classification for any unseen observation, or provide correct prediction at very low rate.

e. Are there any risky decisions or consequences that can result from that model?

   Ans: yes, there are risky decisions or consequences could result from the model, if the model gave wrong classification, the patient will be prescribed wrong type of lenses, that would lead to the det erioration of patient’s condition.