Assignment 3 Report

Student: Lina Mi @01377283

I. R commands used:

1. Load "lenses.csv" dataset and store it into a local R variable "lenses":

```
>lenses<-read.csv("M:/Data Mining/week3/lenses.csv",header=FALSE, sep="")
```

2. Display the content of variable "lenses"

```
> lenses
   V1 V2 V3 V4 V5 V6
1   1  1  1  1  1  3
2   2  1  1  1  2  2
3   3  1  1  2  1  3
4   4  1  1  2  2  1
5   5  1  2  1  1  3
6   6  1  2  1  2  2
7   7  1  2  2  1  3
8   8  1  2  2  2  1
9   9  2  1  1  1  3
10 10  2  1  1  2  2
11 11  2  1  2  1  3
12 12  2  1  2  2  1
13 13  2  2  1  1  3
14 14  2  2  1  2  2
15 15  2  2  2  1  3
16 16  2  2  2  2  3
17 17  3  1  1  1  3
18 18  3  1  1  2  3
19 19  3  1  2  1  3
20 20  3  1  2  2  1
21 21  3  2  1  1  3
22 22  3  2  1  2  2
23 23  3  2  2  1  3
24 24  3  2  2  2  3
```

3. Get the data with only attributes and rename the column name as described in "lenses data description":

```
>lenses_attr<-lenses[,2:6]
>colnames(lenses_attr)<-c("age","prescription","astigmatic","tear","cl
assification")
>str(lenses_attr)
 'data.frame': 24 obs. of  5 variables:
  $ age           : int  1 1 1 1 1 1 1 1 2 2 ...
  $ prescription  : int  1 1 1 1 2 2 2 2 1 1 ...
  $ astigmatic    : int  1 1 2 2 1 1 2 2 1 1 ...
  $ tear          : int  1 2 1 2 1 2 1 2 1 2 ...
  $ classification: int  3 2 3 1 3 2 3 1 3 2 ...
```

4. Clean up the data in "lenses" by replacing all numeric values with descriptive labels as outlined in the "lenses data description" file

```
> lenses_attr$age<-replace(lenses_attr$age, lenses_attr$age==1, "young
")
> lenses_attr$age<-replace(lenses_attr$age, lenses_attr$age==2, "adult
")

>lenses_attr$age<-replace(lenses_attr$age, lenses_attr$age==3, "adult")
```

```
>lenses_attr$prescription<-replace(lenses_attr$prescription, lenses_attr
$prescription==1, "nearsightedness")
> lenses_attr$prescription<-replace(lenses_attr$prescription, lenses_att
r$prescription==2, "farsightedness")
> lenses_attr$astigmatic<-replace(lenses_attr$astigmatic, lenses_attr$as
tigmatic==1, "astigmatic")
> lenses_attr$astigmatic<-replace(lenses_attr$astigmatic, lenses_attr$as
tigmatic==2, "non-astigmatic")
> lenses_attr$tear<-replace(lenses_attr$tear, lenses_attr$tear==1, "redu
ced")
> lenses_attr$tear<-replace(lenses_attr$tear, lenses_attr$tear==2, "norm
al")
> lenses_attr$classification<-replace(lenses_attr$classification, lenses
_attr$classification==1, "hard")
> lenses_attr$classification<-replace(lenses_attr$classification, lenses
_attr$classification==2, "soft")
>
> lenses_attr$classification<-replace(lenses_attr$classification, lenses
_attr$classification==3, "none")
```

Cleaned dataset is shown as following

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

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 tra ined 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

II. Questions:

1. 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

2. 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%

3. 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)

4. 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.

5. 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.