The steps I have followed to clean the data to get the desired outcomes. First, we must convert the file to .csv and import the data to R by using “read.csv” command, then let’s cleanup the data by using “names” and “replace” commands. Here “names” command is used to name the columns, “replace” command is used to replace the values in the columns. All these commands are shown below:

**R Commands:**

1. **Lenses<-read.csv(“lenses.csv”,FALSE)**
2. **Lenses**
3. **Names(lenses)[names(lenses)==”V2”]<-“Age”**

**OR**

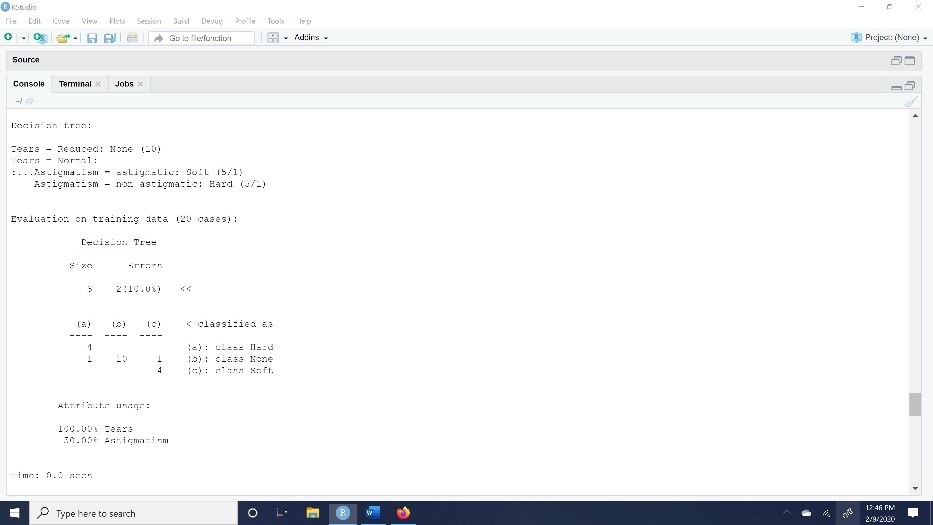
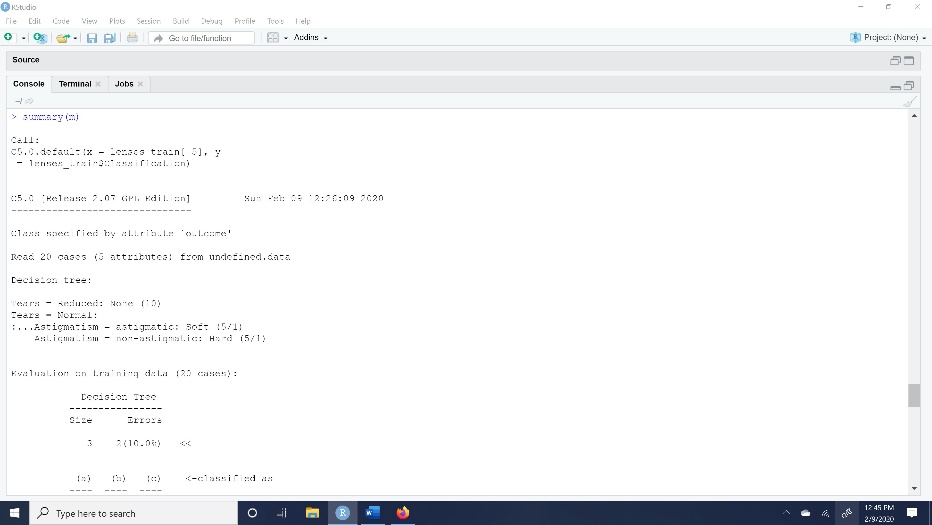
1. **Names(lenses)<-gsub(“v3”,”Perscription”,names(lenses))**
2. **Lenses$Age<-Replace(lenses$Age,lenses$Age==1,’Young’)**
3. **Lenses**

Next, we need to install C5.0 algorithm in RStudio. Then we split the data into training data and testing data by using seed value of “10203”. To get the train decision tree and including the outcome of “summary” command is represented below:

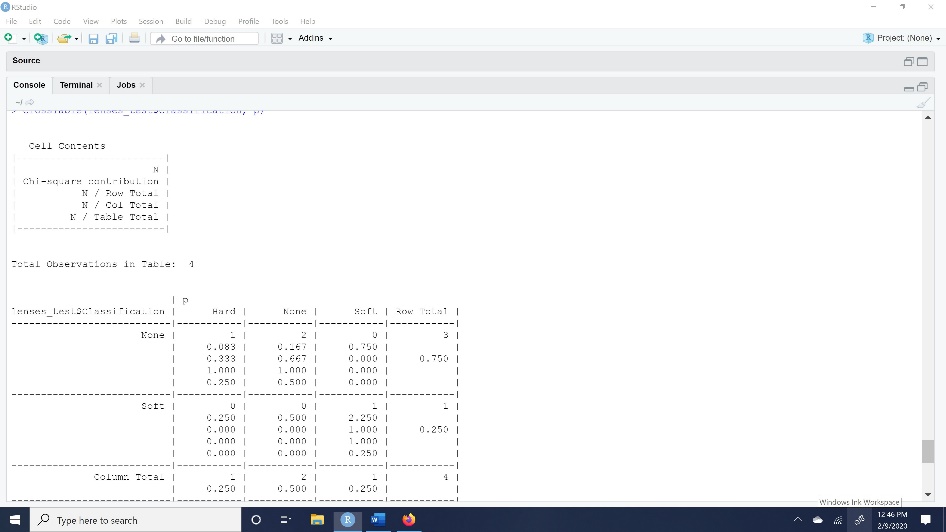
**R Commands:**

1. **Install.packages(“C50”)**
2. **Library(C50)**
3. **lenses\_train<-lenses[1:20,]**
4. **lenses\_test<-lenses[21:24,]**
5. **nrow(lenses\_train)**
6. **nrow(lenses\_test)**
7. **require(C50)**
8. **str(lenses)**
9. **set.seed(10203)**
10. **str(lenses)**
11. **str(lenses\_train)**
12. **str(lenses\_test)**
13. **dim(lenses\_train)**
14. **dim(lenses\_test)**
15. **view(lenses)**
16. **lenses\_train$Classification<-as.factor(lenses\_train$Classification)**
17. **lenses\_train$Age<-as.factor(lenses\_train$Age)**
18. **lenses\_train$Prescription<-as.factor(lenses\_train$Prescription)**
19. **lenses\_train$Astigmatism<-as.factor(lenses\_train$Astigmatism)**
20. **lenses\_train$Tears<-as.factor(lenses\_train$Tears)**
21. **lenses\_test$Age<-as.factor(lenses\_test$Age)**
22. **lenses\_test$Prescription<-as.factor(lenses\_test$Prescription)**
23. **lenses\_test$Astigmatism<-as.factor(lenses\_test$Astigmatism)**
24. **lenses\_test$Tears<-as.factor(lenses\_test$Tears)**
25. **lenses\_test$Classification<-as.factor(lenses\_test$Classification)**
26. **lenses<-as.data.frame(lenses)**
27. **str(lenses)**
28. **m<C5.0(lenses\_train [-5],lenses\_train$classification)**
29. **m**
30. **summary(m)**
31. **p<-predict(m,lenses[21:24,])**
32. **p**
33. **table(lenses[21:24,1],p)**
34. **plot(m)**
35. **install.packages(“gmodels”)**
36. **library(gmodels)**
37. **crosstable(lenses\_test$Classification,p)**

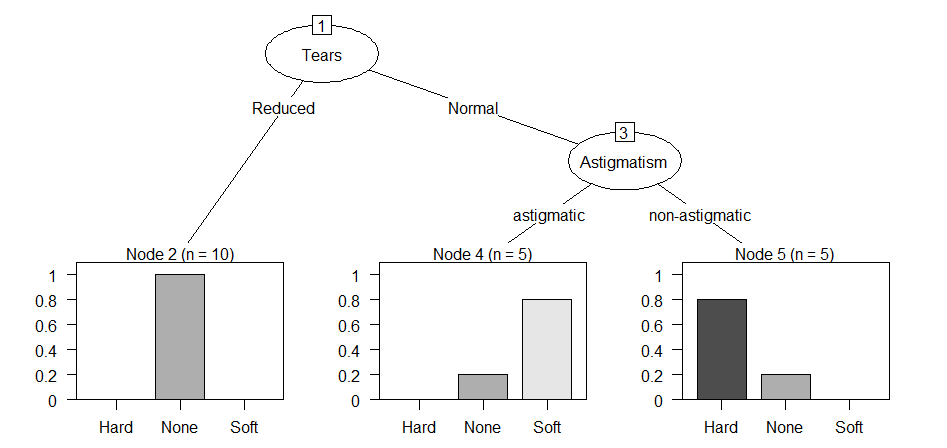
After running this R commands, the outcomes of summary are below:



After running the R commands, then the result of the crosstable is shown below:



The decision tree classification plot is shown below:

****

According to my experience of building decision tree above is bit challenging. After going through all the material provided and seeing tutorials online, I cracked the construction of decision tree. But the tree size is bit varying, when I run this R commands for the first time the result tree size is 4, later I ran it again and then I got the tree size as 3.

By looking at the decision tree plot above we can see how many members are free from getting lenses. There is also missing nodes in the plot too.

The best-case scenario is to sort the classify the data into nodes. The worst-case scenario is this is not 100% perfect data classification. As, I see in the above constructed decision tree, I have faced challenges in doing so.

I have presented this report with few doubts and there are few more alterations needed, but I couldn’t figure out those issues. Please give me some feedback so that I can learn from my mistakes?