Optional Individual Assignment

Alireza Houshidari

Course Rating the Institute for Statistics Education at Statistics.com asks students to rate a variety of aspects of a course as soon as the student completes it. The Institute is contemplating instituting a recommendation system that would provide students with recommendations for additional courses as soon as they submit their rating for a completed course. Consider the except from student ratings of online statistics courses shown in Table 14.16, and the problem of what to recommend to student E.N.

a. First consider a user-based collaborative filter. This requires computing correlations between all student pairs. For which students is it possible to compute correlations with E.N.? Compute them.

First, we read the data. The first column of the dataframe is the name of each row (students), therefore, for conducting further analysis, we add the names to the dataframe using row.names function and remove the first column of the dataframe.

```
courseRating <- read.csv("/Users/alireza/Desktop/DTI/Semester 1/Fundamentals
of Applied Data Science/Optional assignment/Resources/courserating (1)
(1).csv", header = T)
row.names(courseRating) <- courseRating[,1]
courseRating <- courseRating[,-1]
courseRating
```

```
##    SQL Spatial PA1 DM.in.R Python Forecast R.Prog Hadoop Regression
## LN   4      NA  NA      NA      3        2      4     NA          2
## MH   3       4  NA      NA      4       NA     NA     NA         NA
## JH   2       2  NA      NA     NA       NA     NA     NA         NA
## EN   4      NA  NA       4     NA       NA      4     NA          3
## DU   4       4  NA      NA     NA       NA     NA     NA         NA
## FL  NA       4  NA      NA     NA       NA     NA     NA         NA
## GL  NA       4  NA      NA     NA       NA     NA     NA         NA
## AH  NA       3  NA      NA     NA       NA     NA     NA         NA
## SA  NA      NA   4      NA     NA       NA     NA     NA         NA
## RW  NA      NA   2      NA     NA       NA     NA      4         NA
## BA  NA      NA   4      NA     NA       NA     NA     NA         NA
## MG  NA      NA   4      NA     NA        4     NA     NA         NA
## AF  NA      NA   4      NA     NA       NA     NA     NA         NA
## KG  NA      NA   3      NA     NA       NA     NA     NA         NA
## DS   4      NA  NA       2     NA       NA      4     NA         NA
```

We can only compute correlations for students who have at least one course rating in common with E.N including LN, MH, JH, DU, and DS.

```r
data <- courseRating[c(1,2,3,4,5,15),]
#data <- as.matrix(data)
data
```

```
##    SQL Spatial PA1 DM.in.R Python Forecast R.Prog Hadoop Regression
## LN   4      NA  NA      NA      3        2      4     NA          2
## MH   3       4  NA      NA      4       NA     NA     NA         NA
## JH   2       2  NA      NA     NA       NA     NA     NA         NA
## EN   4      NA  NA       4     NA       NA      4     NA          3
## DU   4       4  NA      NA     NA       NA     NA     NA         NA
## DS   4      NA  NA       2     NA       NA      4     NA         NA
```

Then, using the cor function we can calculate the correlations. However, the dataframe consists various NA values, thus, we should add "use='pairwise.complete.obs'" argument so that R knows to only use pairwise observations where both values are present. Also, we want to calculate correlations for each row, therefore, we have to use the transposed version of our dataframe.

```r
cor(t(data[]), use='pairwise.complete.obs')
```

```
## Warning in cor(t(data[]), use = "pairwise.complete.obs"): the standard
## deviation is zero

##    LN MH JH EN DU DS
## LN  1 -1 NA  1 NA NA
## MH -1  1 NA NA NA NA
## JH NA NA NA NA NA NA
## EN  1 NA NA  1 NA NA
## DU NA NA NA NA NA NA
## DS NA NA NA NA NA  1
```

b. Based on the single nearest student to E.N., which single course should we recommend to B.N.? Explain why.

As can be seen in the correlation matrix, the only user with correlation to student EN is LN. According to the dataset, student LN has rated only two other courses that EN has not taken yet, Python (3) and Forecast (2). Therefore, as these two students are much alike we can conclude that student EN should take the Python course as student LN has rated that course higher than the Forecast course.

c. Use R (function similarity()) to compute the cosine similarity between users.

For calculating the cosine similarity between users we used the "proxy" library.

```r
library(proxy)
```

```
##
## Attaching package: 'proxy'

## The following objects are masked from 'package:stats':
##
##     as.dist, dist
```

```
## The following object is masked from 'package:base':
##
##     as.matrix
```

```r
data <- as.matrix(data)

result <- proxy::dist(data, method = "cosine", na.option = "mean")

print(result)
```

```
##              LN           MH           JH           EN           DU
## MH 4.000000e-02
## JH 0.000000e+00 1.005051e-02
## EN 1.089951e-02 0.000000e+00 0.000000e+00
## DU 0.000000e+00 1.005051e-02 2.220446e-16 0.000000e+00
## DS 2.220446e-16 0.000000e+00 0.000000e+00 3.774955e-02 0.000000e+00
```

d. Using the csv file for course ratings, apply item-based collaborative filtering to this dataset (using R) and based on the results, recommend a course to E.N.

First, we transform our dataframe to matrix. Then, we replace the NAs with zero and calculate the similarity matrix using cosine function.

```r
library(lsa)
```

```
## Loading required package: SnowballC
```

```r
data2 <- as.matrix(courseRating)
data3 <- data2
data3[is.na(data3)] = 0

d <- cosine(data3)
print(d)
```

```
##                  SQL   Spatial       PA1   DM.in.R    Python  Forecast
## SQL        1.0000000 0.4155844 0.0000000 0.6115766 0.5470108 0.2038589
## Spatial    0.4155844 1.0000000 0.0000000 0.0000000 0.3646738 0.0000000
## PA1        0.0000000 0.0000000 1.0000000 0.0000000 0.0000000 0.4077178
## DM.in.R    0.6115766 0.0000000 0.0000000 1.0000000 0.0000000 0.0000000
## Python     0.5470108 0.3646738 0.0000000 0.0000000 1.0000000 0.2683282
## Forecast   0.2038589 0.0000000 0.4077178 0.0000000 0.2683282 1.0000000
## R.Prog     0.7895420 0.0000000 0.0000000 0.7745967 0.3464102 0.2581989
## Hadoop     0.0000000 0.0000000 0.2279212 0.0000000 0.0000000 0.0000000
## Regression 0.6321395 0.0000000 0.0000000 0.7442084 0.3328201 0.2480695
##               R.Prog    Hadoop Regression
## SQL        0.7895420 0.0000000  0.6321395
## Spatial    0.0000000 0.0000000  0.0000000
## PA1        0.0000000 0.2279212  0.0000000
## DM.in.R    0.7745967 0.0000000  0.7442084
## Python     0.3464102 0.0000000  0.3328201
## Forecast   0.2581989 0.0000000  0.2480695
```

```
## R.Prog     1.0000000 0.0000000  0.8006408
## Hadoop     0.0000000 1.0000000  0.0000000
## Regression 0.8006408 0.0000000  1.0000000
```

By using recommenderlab library we conducted our IBCF, predict ratings, and provided recommendations. According to our prediction, Spatial course is the most recommended course for student EN as this course has the most predicted rating.

```
library(recommenderlab)

## Loading required package: Matrix

## Loading required package: arules

##
## Attaching package: 'arules'

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

## Registered S3 methods overwritten by 'registry':
##   method               from
##   print.registry_field proxy
##   print.registry_entry proxy

d <- as(data2, "realRatingMatrix")
rec <- Recommender(d, "IBCF")
pred <- predict(rec, d, type = "ratings")
as(pred, "matrix")

##    SQL Spatial PA1 DM.in.R   Python Forecast   R.Prog Hadoop Regression
## LN  NA       3   2       4       NA       NA       NA     NA         NA
## MH  NA      NA  NA       3       NA        4 3.333333     NA    3.92733
## JH  NA      NA  NA       2 2.000000        2 2.000000     NA    2.00000
## EN  NA       4  NA      NA 3.591052        3       NA     NA         NA
## DU  NA      NA  NA       4 4.000000        4 4.000000     NA    4.00000
## FL   4      NA  NA      NA 4.000000       NA       NA     NA         NA
## GL   4      NA  NA      NA 4.000000       NA       NA     NA         NA
## AH   3      NA  NA      NA 3.000000       NA       NA     NA         NA
## SA  NA      NA  NA      NA       NA        4       NA      4         NA
## RW  NA      NA  NA      NA       NA        2       NA     NA         NA
## BA  NA      NA  NA      NA       NA        4       NA      4         NA
## MG   4      NA  NA      NA 4.000000       NA 4.000000      4    4.00000
## AF  NA      NA  NA      NA       NA        4       NA      4         NA
## KG  NA      NA  NA      NA       NA        3       NA      3         NA
## DS  NA       4  NA      NA 4.000000        4       NA     NA    4.00000
```

Q2. Association Rule

Identifying Course Combinations. The Institute for Statistics Education at Statistics.com offers online courses in statistics and analytics, and is seeking information

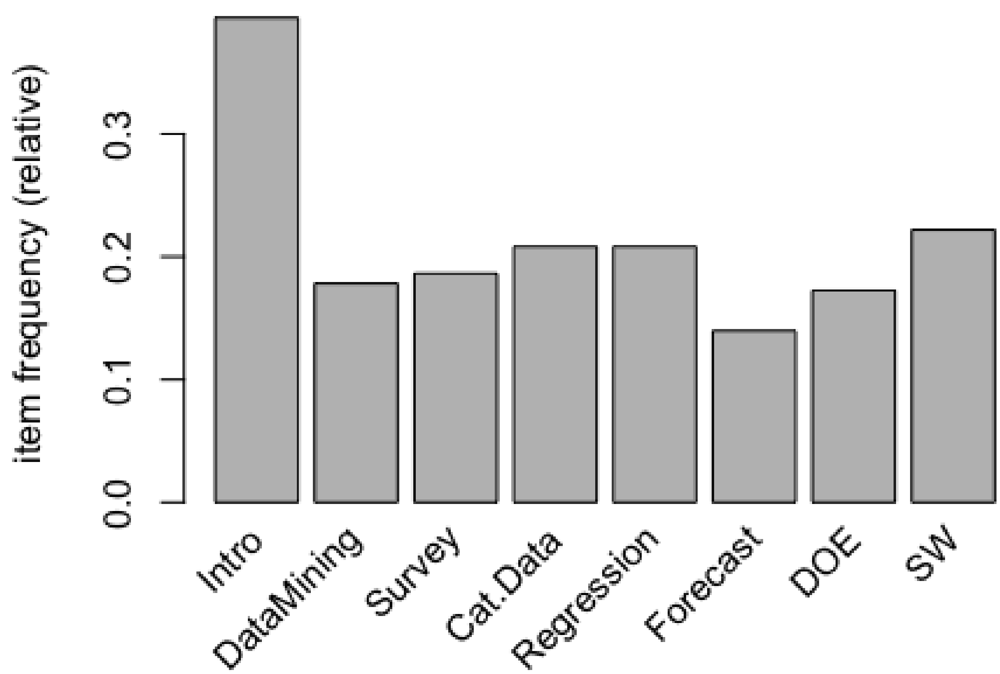
that will help in packaging and sequencing courses. Consider the data in the file Course-Topics.cs, the first few rows of which are shown in Table 14.13. These data are for purchases of online statistics courses at Statistics.com. Each row represents the courses attended by a single customer. The firm wishes to assess alternative sequencing and bundling of courses. Use association rules to analyze these data, and interpret several of the resulting rules.

First, I convert our dataframe to a transaction database format and display it in a readable form using "arules" library.

```
library(arules)
ct.df <- read.csv("/Users/alireza/Desktop/DTI/Semester 1/Fundamentals of
Applied Data Science/Optional assignment/Resources/Coursetopics (1).csv")

ct.mat <- as(ct.df, "matrix")

ct.trans <- as(ct.mat, "transactions")
```

Then, an item frequency plot has been drawn.

```
itemFrequencyPlot(ct.trans)
```



Finally, an association rule model has been built in this section. It's support value has been set as 0.01 and the confidence value has been set as 0.5. The first ten rules sorted by their lift values has been illustrated.

```
rules <- apriori(ct.trans, parameter = list(support = 0.01, confidence = 0.5,
target = "rules"))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 3
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[8 item(s), 365 transaction(s)] done [0.00s].
## sorting and recoding items ... [8 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [54 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
ruleshead <- inspect(head(sort(rules, by = "lift"), 10))
```

```
##      lhs                                rhs          support    confidence
## [1]  {Intro, Regression, Forecast}   => {DataMining} 0.01369863 0.7142857
## [2]  {Intro, Survey, DOE}            => {Cat.Data}   0.01095890 0.8000000
## [3]  {Intro, DataMining, Cat.Data}   => {Regression} 0.01643836 0.7500000
## [4]  {Intro, DataMining, Regression} => {Forecast}   0.01369863 0.5000000
## [5]  {Intro, Survey, Cat.Data}       => {Forecast}   0.01369863 0.5000000
## [6]  {Intro, Regression, DOE}        => {SW}         0.01917808 0.7777778
## [7]  {Intro, DataMining, Forecast}   => {Regression} 0.01369863 0.7142857
## [8]  {Intro, Cat.Data, Forecast}     => {Survey}     0.01369863 0.6250000
## [9]  {DataMining, DOE}               => {Cat.Data}   0.01643836 0.6666667
## [10] {Survey, Regression}            => {Cat.Data}   0.01643836 0.6666667
##      coverage   lift     count
## [1]  0.01917808 4.010989 5
## [2]  0.01369863 3.842105 4
## [3]  0.02191781 3.601974 6
## [4]  0.02739726 3.578431 5
## [5]  0.02739726 3.578431 5
## [6]  0.02465753 3.504801 7
## [7]  0.01917808 3.430451 5
## [8]  0.02191781 3.354779 5
```

```
## [9]  0.02465753 3.201754 6
## [10] 0.02465753 3.201754 6
```

```
rules
```

```
## set of 54 rules
```

```
ruleshead
```

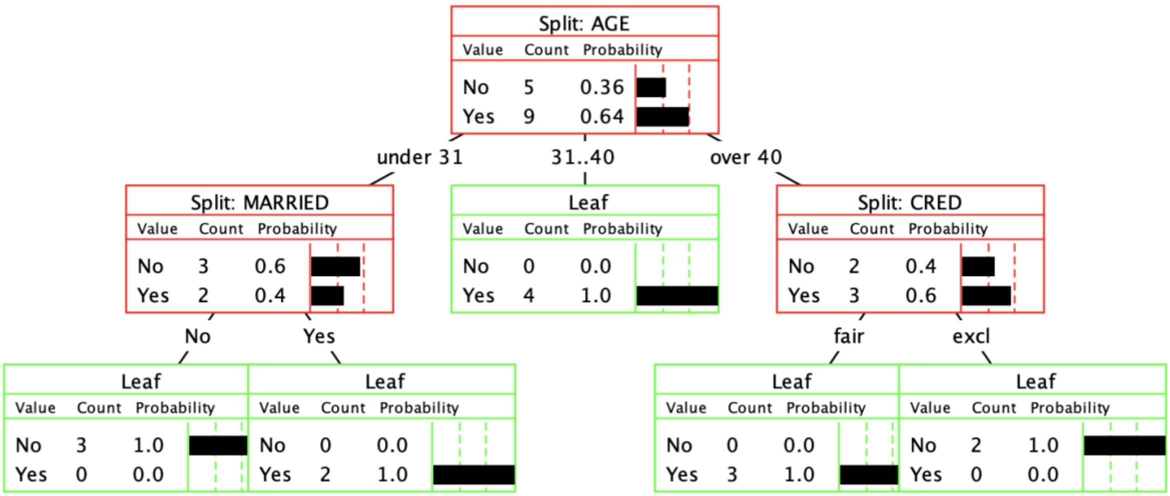
```
##                                    lhs              rhs    support confidence
## [1]    {Intro, Regression, Forecast} => {DataMining} 0.01369863  0.7142857
## [2]             {Intro, Survey, DOE} =>   {Cat.Data} 0.01095890  0.8000000
## [3]    {Intro, DataMining, Cat.Data} => {Regression} 0.01643836  0.7500000
## [4]  {Intro, DataMining, Regression} =>   {Forecast} 0.01369863  0.5000000
## [5]       {Intro, Survey, Cat.Data} =>   {Forecast} 0.01369863  0.5000000
## [6]        {Intro, Regression, DOE} =>         {SW} 0.01917808  0.7777778
## [7]    {Intro, DataMining, Forecast} => {Regression} 0.01369863  0.7142857
## [8]      {Intro, Cat.Data, Forecast} =>     {Survey} 0.01369863  0.6250000
## [9]                {DataMining, DOE} =>   {Cat.Data} 0.01643836  0.6666667
## [10]            {Survey, Regression} =>   {Cat.Data} 0.01643836  0.6666667
##        coverage     lift count
## [1]  0.01917808 4.010989     5
## [2]  0.01369863 3.842105     4
## [3]  0.02191781 3.601974     6
## [4]  0.02739726 3.578431     5
## [5]  0.02739726 3.578431     5
## [6]  0.02465753 3.504801     7
## [7]  0.01917808 3.430451     5
## [8]  0.02191781 3.354779     5
## [9]  0.02465753 3.201754     6
## [10] 0.02465753 3.201754     6
```

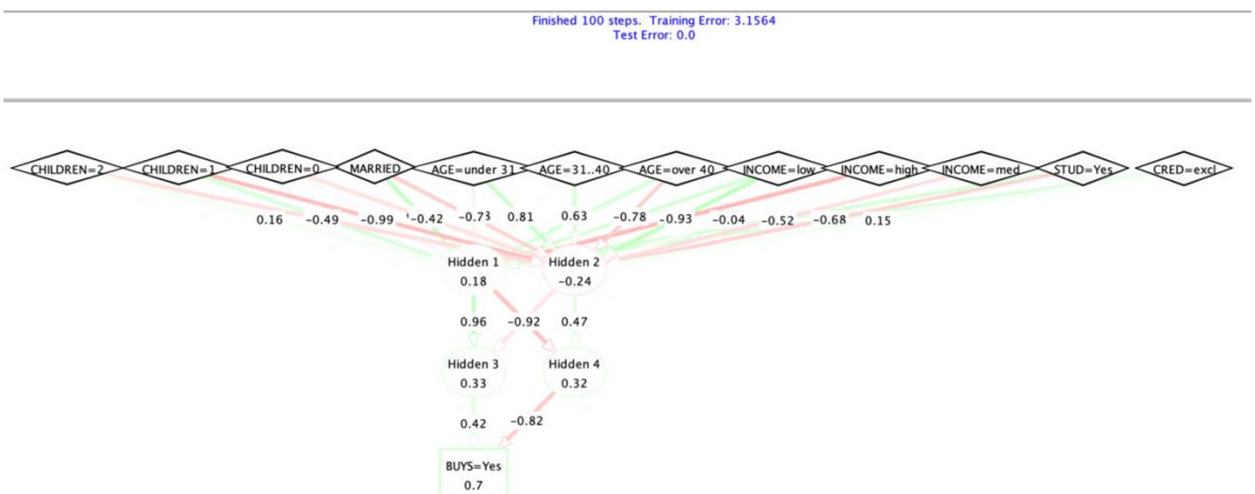
According to the results, if Intro, Survey, and DOE are taken by a student we can be around 80% sure that the student will also take Cat.Data. Also, if Intro, DataMining, and Regression are taken, we can be 50% sure that they will take Forecast. These examples are the highest and lowest confidences in this data set which means these rules are relatively strong.

Q3. A comparison between neural networks and decision trees

1- If Age > 40 and Cred = Excel then NO (People aged more than 40 with excellent credit will not buy a computer) If Age > 40 and Cred = Fair then YES (People aged more than 40 with fair credit will buy a computer) If 31 > Age > 40 then YES (People aged between 31 and 40 w will buy a computer) If Age < 31 and Married = Yes then YES (People aged less than 31 and married will buy a computer) If Age < 31 and Married = No then NO (People aged less than 31 and unmarried will not buy a computer)

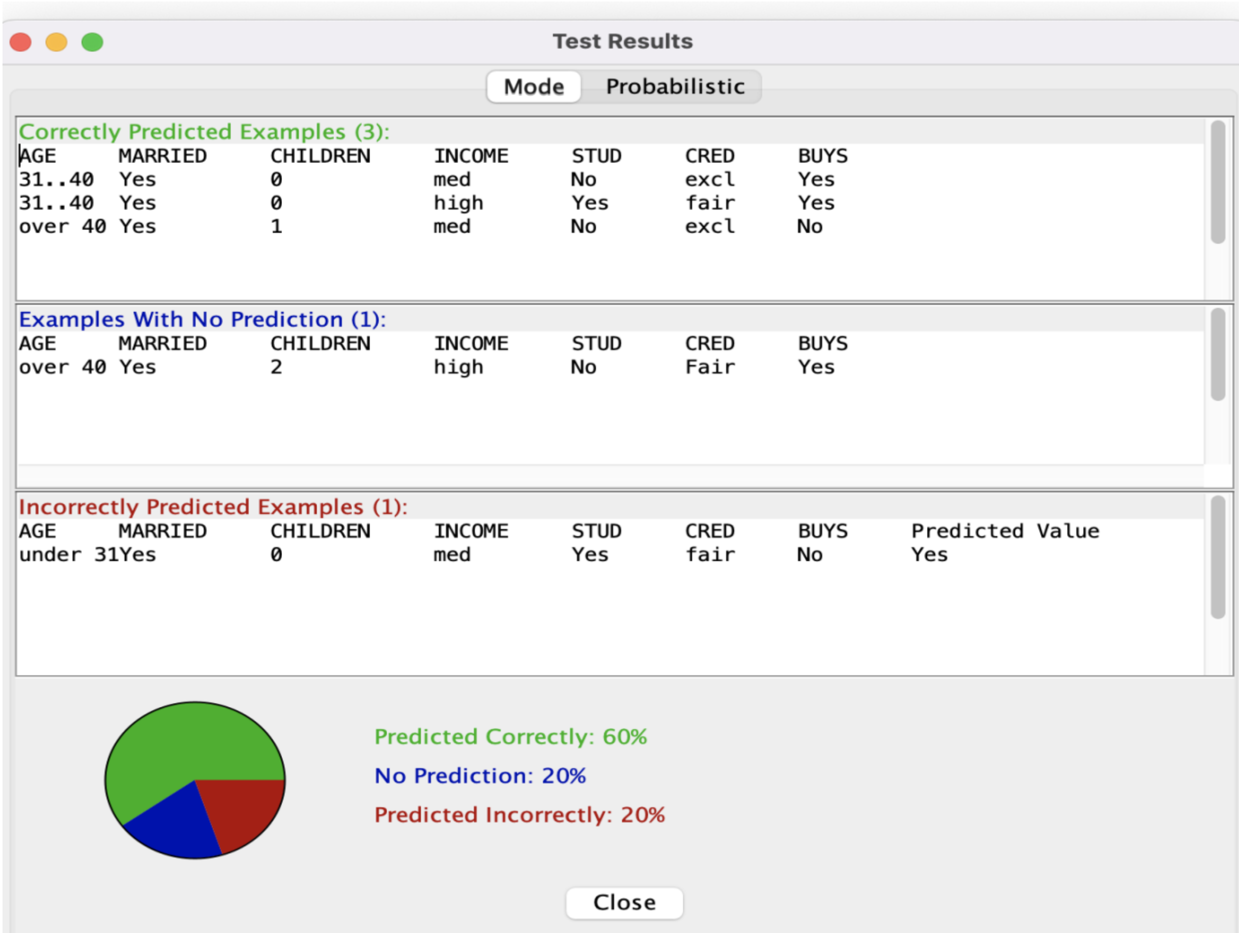
2-

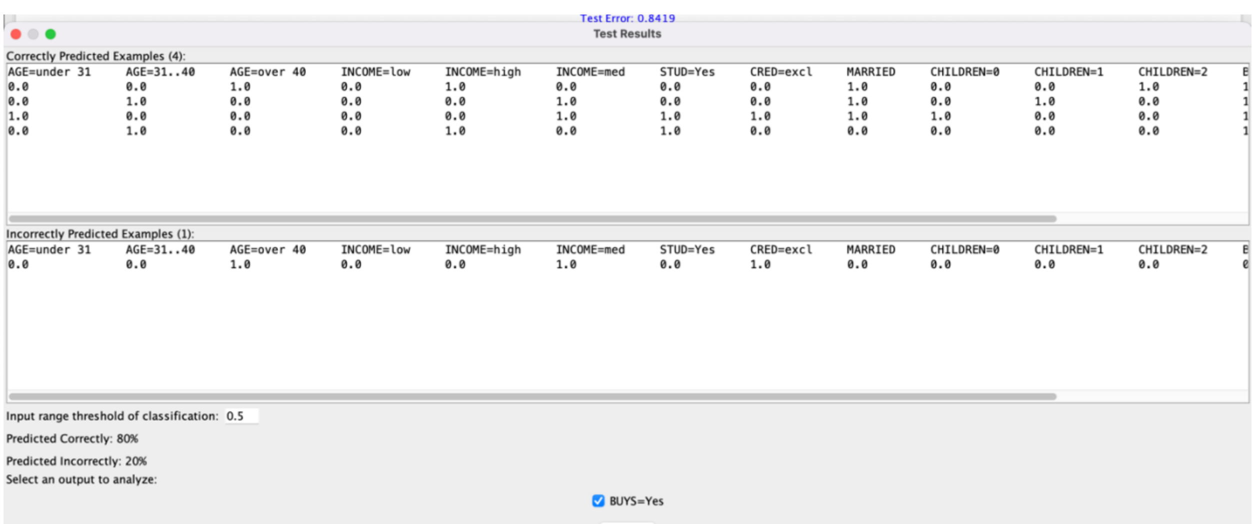




3.A Yes, I believe decision tree model is significantly comprehensible to human. However, I did not find the neural network model comprehensible.

3.B Decision tree prediction accuracy was 60 percent and neural network model's prediction accuracy was 80 percent.





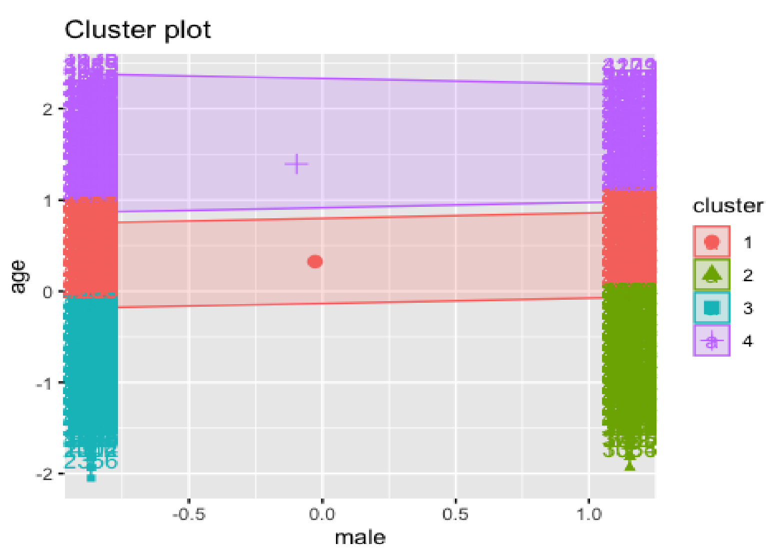
3.C Neural network models have higher accuracy rates in their predictions but the are not easily comprehensible for humans. On the other hand, decision trees might be slightly less accurate but they are much easier to comprehend for human brain.

Q4 K-Means Clustering Part A a. First, lets load the data, build sexAge.df which is a dataframe consists only the two demanded column and standardize the age in it.

```
farmingham.df <- read.csv("/Users/alireza/Desktop/DTI/Semester 1/Fundamentals
of Applied Data Science/Optional assignment/Resources/framingham (1).csv")

sexAge.df <- farmingham.df[,c(1,2)]

sexAge.norm <- sexAge.df
sexAge.norm[,2] <- as.data.frame(scale(sexAge.norm[, 2]))

head(sexAge.norm)

##   male        age
## 1    1 -1.2341374
## 2    0 -0.4176149
## 3    1 -0.1843228
## 4    0  1.3320761
## 5    0 -0.4176149
## 6    0 -0.7675531
```

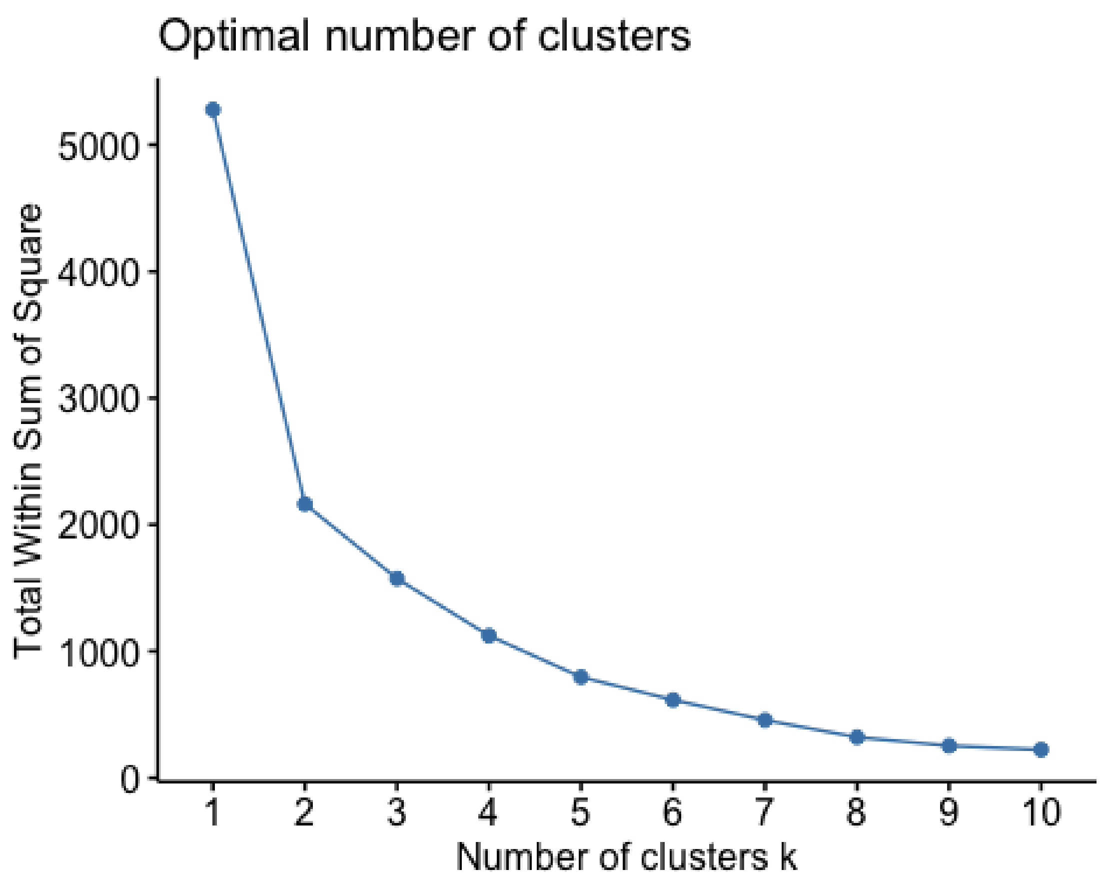
Then, we preform and plot the k-means clustering using “factoextra” library.

```
library(factoextra)

## Loading required package: ggplot2

## Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa

set.seed(123)
k4 <- kmeans(sexAge.norm, centers = 4, nstart = 10)

fviz_cluster(k4, data = sexAge.norm)
```



b. In this section, we apply the elbow method to determine the best k and plot it.

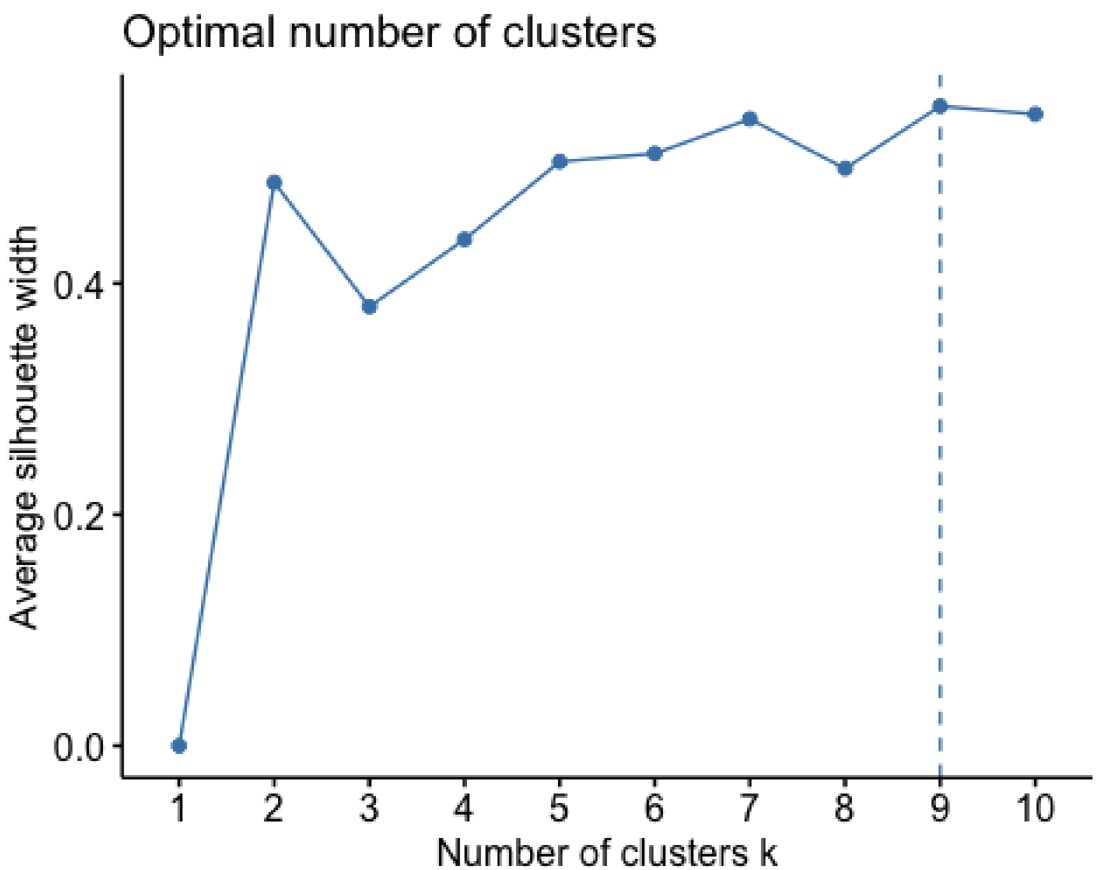
```r
set.seed(777)
fviz_nbclust(sexAge.norm, kmeans, method = "wss")
```



According to the results, k = 4 is a optimal point using the elbow method.

C. I used the following function to plot the average Silhouette to find the optimal number of clusters. Accoding to the results, K = 9 is the optimal number for k.

```r
fviz_nbclust(sexAge.norm, kmeans, method = "silhouette")
```



2.

Part B

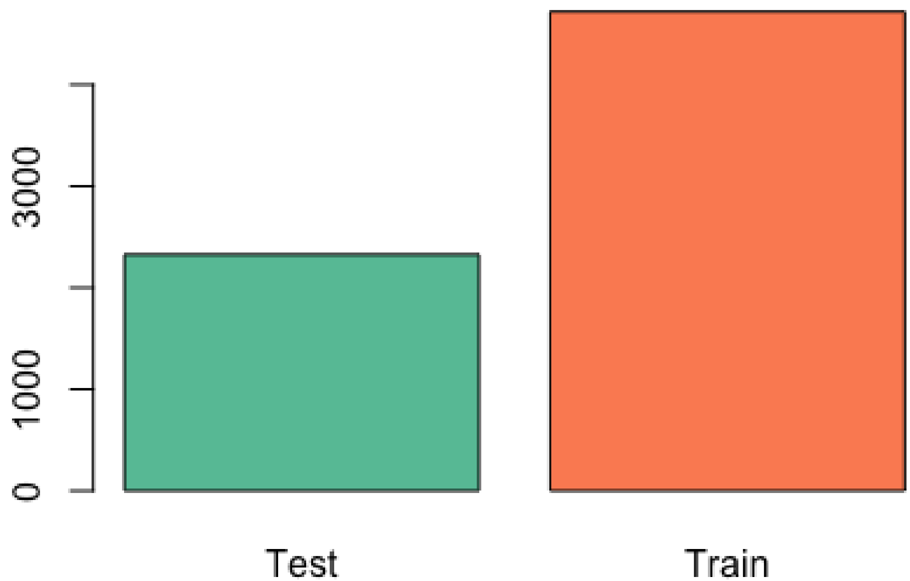
a. Firstly, data should be imported

```
churn.df <- read.csv("/Users/alireza/Desktop/DTI/Semester 1/Fundamentals of
Applied Data Science/Optional assignment/Resources/customer_churn.csv")

head(churn.df)

##   customerID gender SeniorCitizen Partner Dependents tenure PhoneService
## 1 7590-VHVEG Female             0     Yes         No      1           No
## 2 5575-GNVDE   Male             0      No         No     34          Yes
## 3 3668-QPYBK   Male             0      No         No      2          Yes
## 4 7795-CFOCW   Male             0      No         No     45           No
## 5 9237-HQITU Female             0      No         No      2          Yes
## 6 9305-CDSKC Female             0      No         No      8          Yes
##      MultipleLines InternetService OnlineSecurity OnlineBackup
DeviceProtection
## 1 No phone service             DSL             No          Yes
No
## 2               No             DSL            Yes           No
Yes
```

```
## 3                No             DSL             Yes          Yes
No
## 4 No phone service             DSL             Yes           No
Yes
## 5                No     Fiber optic              No           No
No
## 6               Yes     Fiber optic              No           No
Yes
##   TechSupport StreamingTV StreamingMovies       Contract PaperlessBilling
## 1          No          No              No Month-to-month              Yes
## 2          No          No              No       One year               No
## 3          No          No              No Month-to-month              Yes
## 4         Yes          No              No       One year               No
## 5          No          No              No Month-to-month              Yes
## 6          No         Yes             Yes Month-to-month              Yes
##             PaymentMethod MonthlyCharges TotalCharges Churn
## 1          Electronic check          29.85        29.85    No
## 2              Mailed check          56.95      1889.50    No
## 3              Mailed check          53.85       108.15   Yes
## 4 Bank transfer (automatic)          42.30      1840.75    No
## 5          Electronic check          70.70       151.65   Yes
## 6          Electronic check          99.65       820.50   Yes
```

Then, we split the data and plot the number of observations.

```
library(RColorBrewer)
set.seed(111)
dt = sort(sample(nrow(churn.df), nrow(churn.df)*0.67))
train<-churn.df[dt,]
test<-churn.df[-dt,]

tt <- data.frame(Test = nrow(test), Train = nrow(train))
tt <- as.matrix(tt)

coul <- brewer.pal(5, "Set2")
barplot(height=tt[1,], col=coul )
```



B.

```
t1 <- table(train$Churn)
t2 <- table(test$Churn)
t3 <- table(churn.df$Churn)



t1.ratio <- t1[2]/(t1[1] + t1[2])
t1.ratio

##       Yes
## 0.2666384

t2.ratio <- t2[2]/(t2[1] + t2[2])
t2.ratio

##       Yes
## 0.2627957
```

Thus, we have to add 2300 rows with false Churn value in order to have 20 percent true churn value in the data. Or we can reduce the number of true churn values. We can also do both of them using ROSE package as demonstrated below.

C. In this section rebalanced dataset is built using ROSE package and the ratio has been illustrated to confirm that the sample now has 20 percent True churn values.

```
library(ROSE)

## Loaded ROSE 0.0-4

churn.df$Churn <- as.factor(churn.df$Churn)

rebalanced <- ovun.sample(Churn~., data = churn.df, method = "both", p =
0.212, seed = 213)$data
t1n <- table(rebalanced$Churn)

t1n.ratio <- t1n[2]/(t1n[1] + t1n[2])
t1n.ratio

##       Yes
## 0.2009386
```

Now again I build the Training and Test sets.

```
set.seed(313)
re.dt = sort(sample(nrow(rebalanced), nrow(rebalanced)*0.67))
re.train<-rebalanced[dt,]
re.test<-rebalanced[-dt,]
```

D.

Unfortunately, I did not have time to complete these parts.

```
re.train$Churn<-ifelse(re.train$Churn=="Yes",1,0)

table(re.train$Churn)

##
##    0    1
## 3755  957

re.test$Churn<-ifelse(re.test$Churn=="Yes",1,0)

table(re.test$Churn)

##
##    0    1
## 1864  456
```