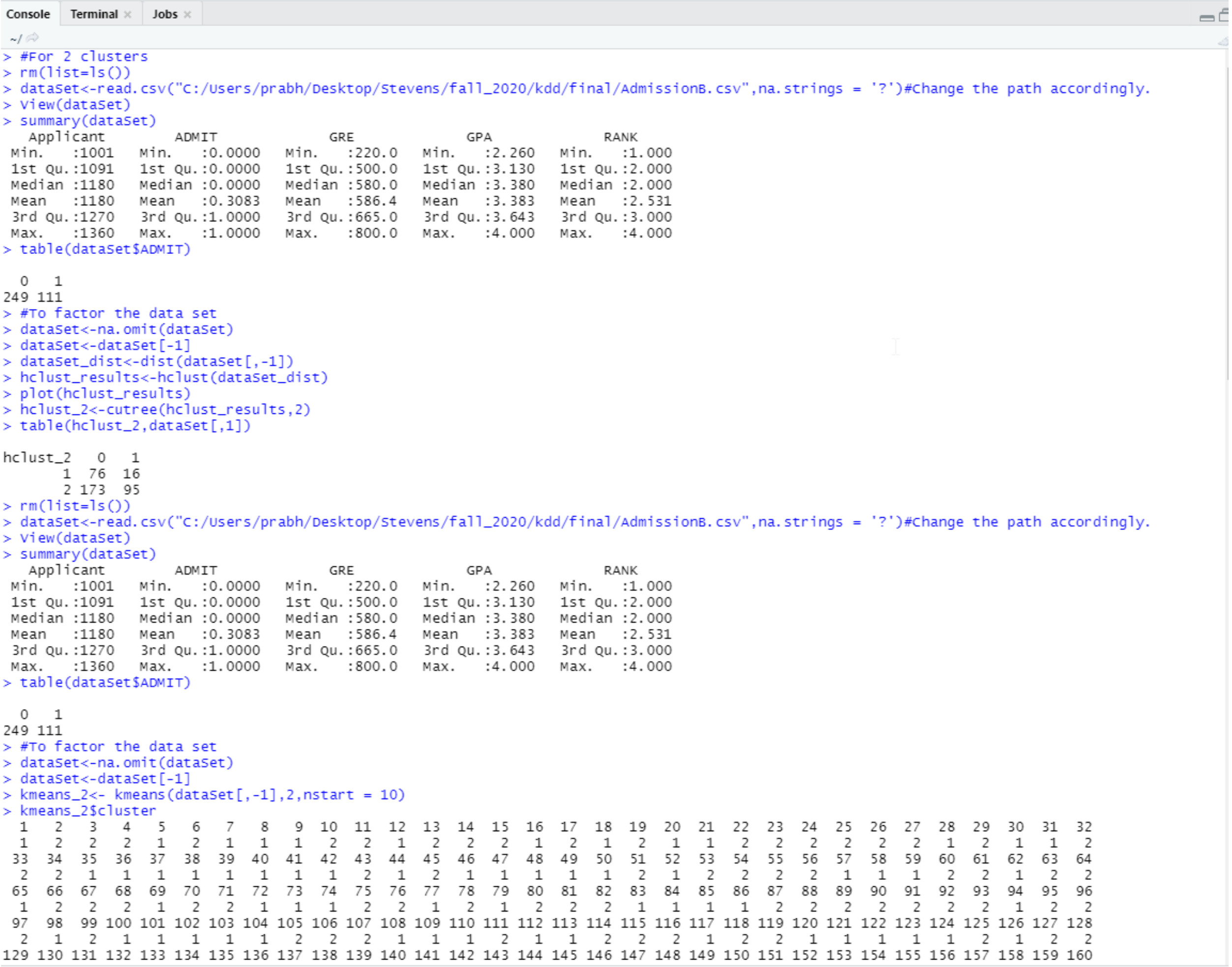
Final Exam

The following problems use “AdmissionB.csv” and “Admission_catB.csv” datasets. The datasets list the applicants that have been admitted (ADMIT=1) or rejected (ADMIT=0), based on GRE score, GRE and the rank of school attended (RANK1,2,3 or 4). Note: The Admission_catB.csv dataset contains categorized GRE scores and GPAs.

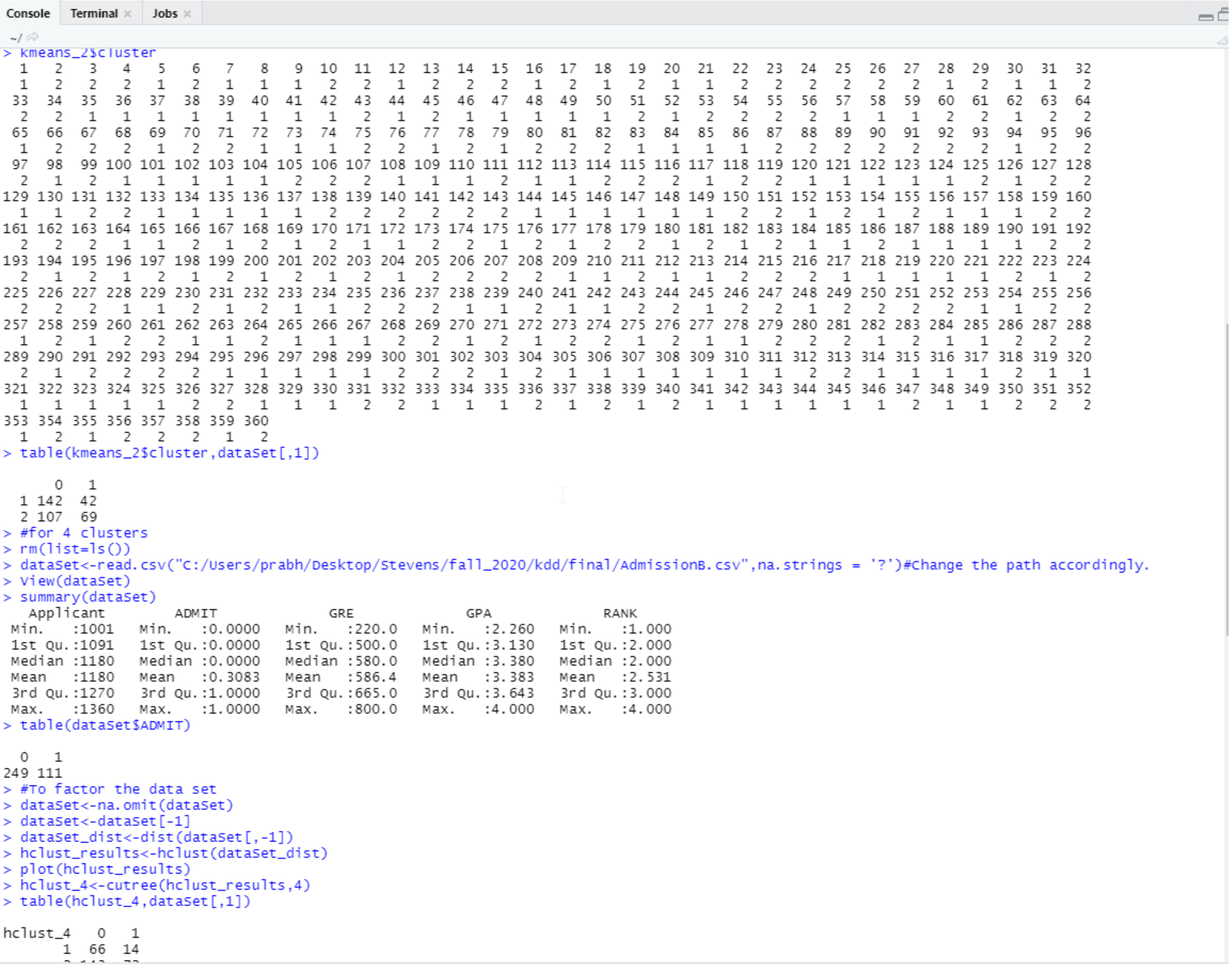
Problem #1: (40 points)

Cluster applicants in “AdmissionB.csv” into 2 clusters using the GRE, the GPA and the RANK variables. Compare the four clusters for each of the following two methods.

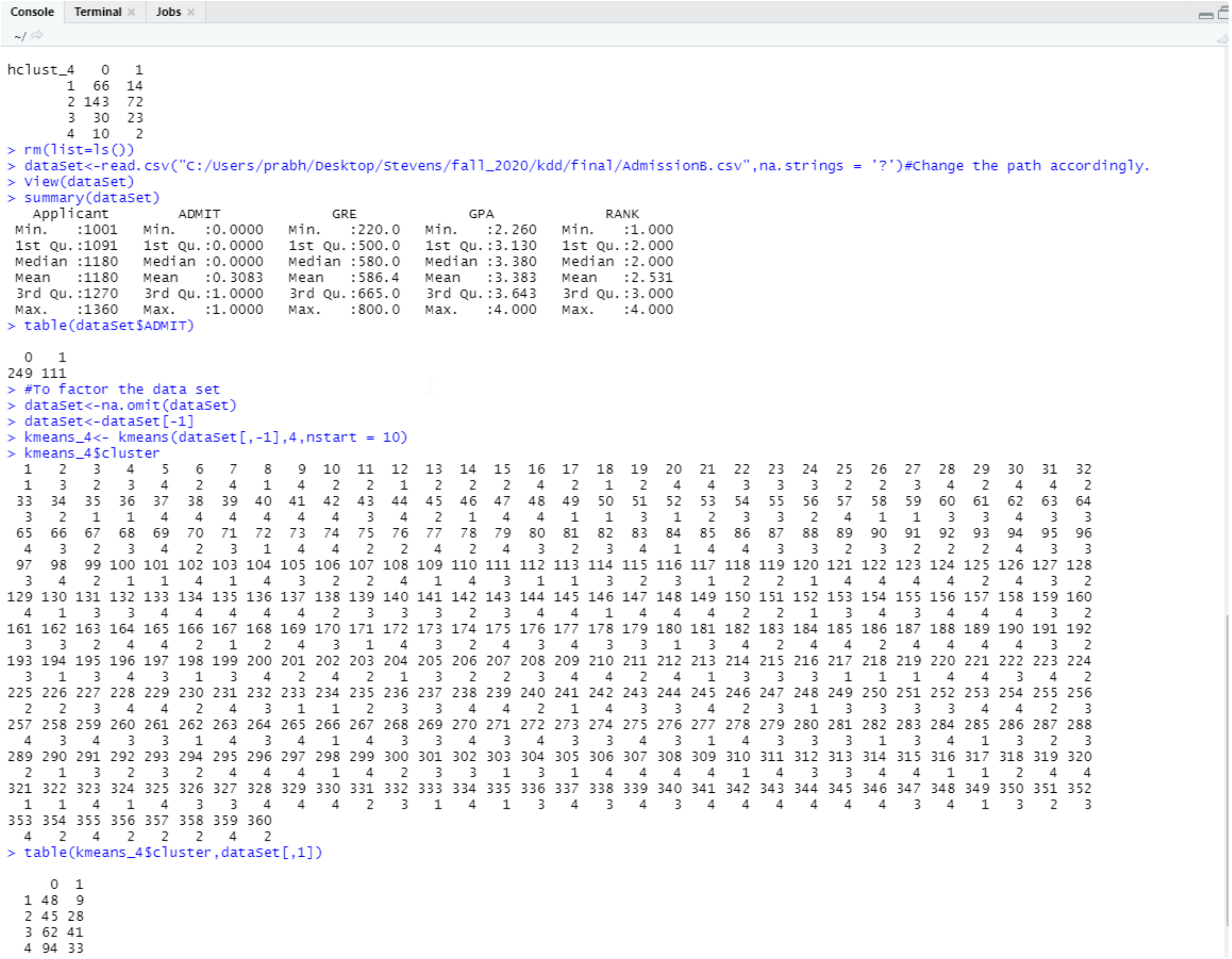
- Hierarchical clustering
- K-means



```
> #For 2 clusters
> rm(list=ls())
> dataSet<-read.csv("C:/Users/prabh/Desktop/Stevens/fall_2020/kdd/final/AdmissionB.csv",na.strings = '?')#Change the path accordingly.
> View(dataSet)
> summary(dataSet)
   Applicant        ADMIT              GRE             GPA             RANK      
 Min.   :1001   Min.   :0.0000   Min.   :220.0   Min.   :2.260   Min.   :1.000  
 1st Qu.:1091   1st Qu.:0.0000   1st Qu.:500.0   1st Qu.:3.130   1st Qu.:2.000  
 Median :1180   Median :0.0000   Median :580.0   Median :3.380   Median :2.000  
 Mean   :1180   Mean   :0.3083   Mean   :586.4   Mean   :3.383   Mean   :2.531  
 3rd Qu.:1270   3rd Qu.:1.0000   3rd Qu.:665.0   3rd Qu.:3.643   3rd Qu.:3.000  
 Max.   :1360   Max.   :1.0000   Max.   :800.0   Max.   :4.000   Max.   :4.000  
> table(dataSet$ADMIT)

  0   1 
249 111 
> #To factor the data set
> dataSet<-na.omit(dataSet)
> dataSet<-dataSet[-1]
> dataSet_dist<-dist(dataSet[,-1])
> hclust_results<-hclust(dataSet_dist)
> plot(hclust_results)
> hclust_2<-cutree(hclust_results,2)
> table(hclust_2,dataSet[,1])
       
hclust_2   0   1
       1  76  16
       2 173  95
> rm(list=ls())
> dataSet<-read.csv("C:/Users/prabh/Desktop/Stevens/fall_2020/kdd/final/AdmissionB.csv",na.strings = '?')#Change the path accordingly.
> View(dataSet)
> summary(dataSet)
   Applicant        ADMIT              GRE             GPA             RANK      
 Min.   :1001   Min.   :0.0000   Min.   :220.0   Min.   :2.260   Min.   :1.000  
 1st Qu.:1091   1st Qu.:0.0000   1st Qu.:500.0   1st Qu.:3.130   1st Qu.:2.000  
 Median :1180   Median :0.0000   Median :580.0   Median :3.380   Median :2.000  
 Mean   :1180   Mean   :0.3083   Mean   :586.4   Mean   :3.383   Mean   :2.531  
 3rd Qu.:1270   3rd Qu.:1.0000   3rd Qu.:665.0   3rd Qu.:3.643   3rd Qu.:3.000  
 Max.   :1360   Max.   :1.0000   Max.   :800.0   Max.   :4.000   Max.   :4.000  
> table(dataSet$ADMIT)

  0   1 
249 111 
> #To factor the data set
> dataSet<-na.omit(dataSet)
> dataSet<-dataSet[-1]
> kmeans_2<- kmeans(dataSet[,-1],2,nstart = 10)
> kmeans_2$cluster
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 
  1   2   2   2   1   2   1   1   1   2   2   1   2   2   2   1   2   1   2   1   1   2   2   2   2   2   2   1   2   1   1   2 
 33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64 
  2   2   1   1   1   1   1   1   1   1   2   1   2   1   1   1   1   1   2   1   2   2   2   2   1   1   1   2   2   1   2   2 
 65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96 
  1   2   2   2   1   2   2   1   1   1   2   2   1   2   1   2   2   2   1   1   1   1   2   2   2   2   2   2   2   1   2   2 
 97  98  99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 
  2   1   2   1   1   1   1   1   2   2   2   1   1   1   2   1   1   2   2   2   1   2   2   1   1   1   1   1   2   1   2   2 
129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 
```

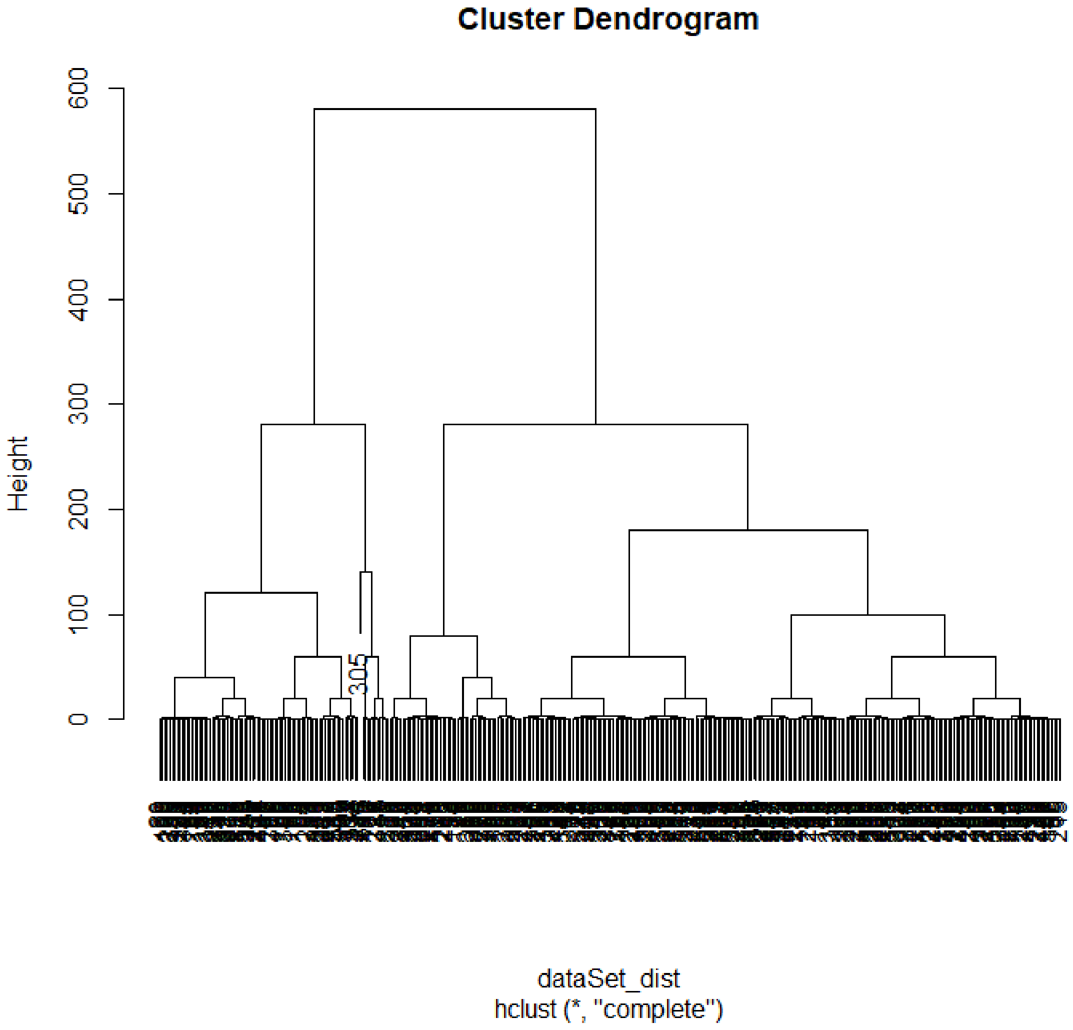


```
> kmeans_2$cluster
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 
  1   2   2   2   1   2   1   1   1   2   2   1   2   2   2   1   2   1   2   1   1   2   2   2   2   2   2   1   2   1   1   2 
 33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64 
  2   2   1   1   1   1   1   1   1   1   2   1   2   1   1   1   1   1   2   1   2   2   2   2   1   1   1   2   2   1   2   2 
 65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96 
  1   2   2   2   1   2   2   1   1   1   2   2   1   2   1   2   2   2   1   1   1   1   2   2   2   2   2   2   2   1   2   2 
 97  98  99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 
  2   1   2   1   1   1   1   1   2   2   2   1   1   1   2   1   1   2   2   2   1   2   2   1   1   1   1   1   2   1   2   2 
129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 
  1   1   2   2   1   1   1   1   1   2   2   2   2   2   2   1   1   1   1   1   1   2   2   1   2   1   2   1   1   1   2   2 
161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 
  2   2   2   1   1   2   1   2   1   2   1   1   2   2   1   2   1   2   2   1   2   1   2   1   1   2   1   1   1   1   2   2 
193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 
  2   1   2   1   2   1   2   1   2   1   2   1   2   2   2   2   1   1   2   1   1   2   2   2   1   1   1   1   1   2   1   2 
225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 
  2   2   2   1   1   2   1   2   1   1   2   2   2   1   1   2   1   1   2   2   1   2   2   1   2   2   2   2   1   1   2   2 
257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 
  1   2   1   2   2   1   1   2   1   1   1   2   2   1   2   1   2   2   1   2   1   1   2   2   2   1   2   1   1   2   2   2 
289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 
  2   1   2   2   2   2   1   1   1   1   1   2   2   2   1   2   1   1   1   1   1   1   1   2   2   1   1   1   1   2   1   1 
321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 
  1   1   1   1   1   2   2   1   1   1   2   2   1   1   1   2   1   2   1   2   1   1   1   1   1   1   2   1   1   2   2   2 
353 354 355 356 357 358 359 360 
  1   2   1   2   2   2   1   2 
> table(kmeans_2$cluster,dataSet[,1])
   
      0   1
  1 142  42
  2 107  69
> #for 4 clusters
> rm(list=ls())
> dataSet<-read.csv("C:/Users/prabh/Desktop/Stevens/fall_2020/kdd/final/AdmissionB.csv",na.strings = '?')#Change the path accordingly.
> View(dataSet)
> summary(dataSet)
   Applicant        ADMIT             GRE             GPA             RANK      
 Min.   :1001   Min.   :0.0000   Min.   :220.0   Min.   :2.260   Min.   :1.000  
 1st Qu.:1091   1st Qu.:0.0000   1st Qu.:500.0   1st Qu.:3.130   1st Qu.:2.000  
 Median :1180   Median :0.0000   Median :580.0   Median :3.380   Median :2.000  
 Mean   :1180   Mean   :0.3083   Mean   :586.4   Mean   :3.383   Mean   :2.531  
 3rd Qu.:1270   3rd Qu.:1.0000   3rd Qu.:665.0   3rd Qu.:3.643   3rd Qu.:3.000  
 Max.   :1360   Max.   :1.0000   Max.   :800.0   Max.   :4.000   Max.   :4.000  
> table(dataSet$ADMIT)

  0   1 
249 111 
> #To factor the data set
> dataSet<-na.omit(dataSet)
> dataSet<-dataSet[-1]
> dataSet_dist<-dist(dataSet[,-1])
> hclust_results<-hclust(dataSet_dist)
> plot(hclust_results)
> hclust_4<-cutree(hclust_results,4)
> table(hclust_4,dataSet[,1])
       
hclust_4   0   1
       1  66  14
```

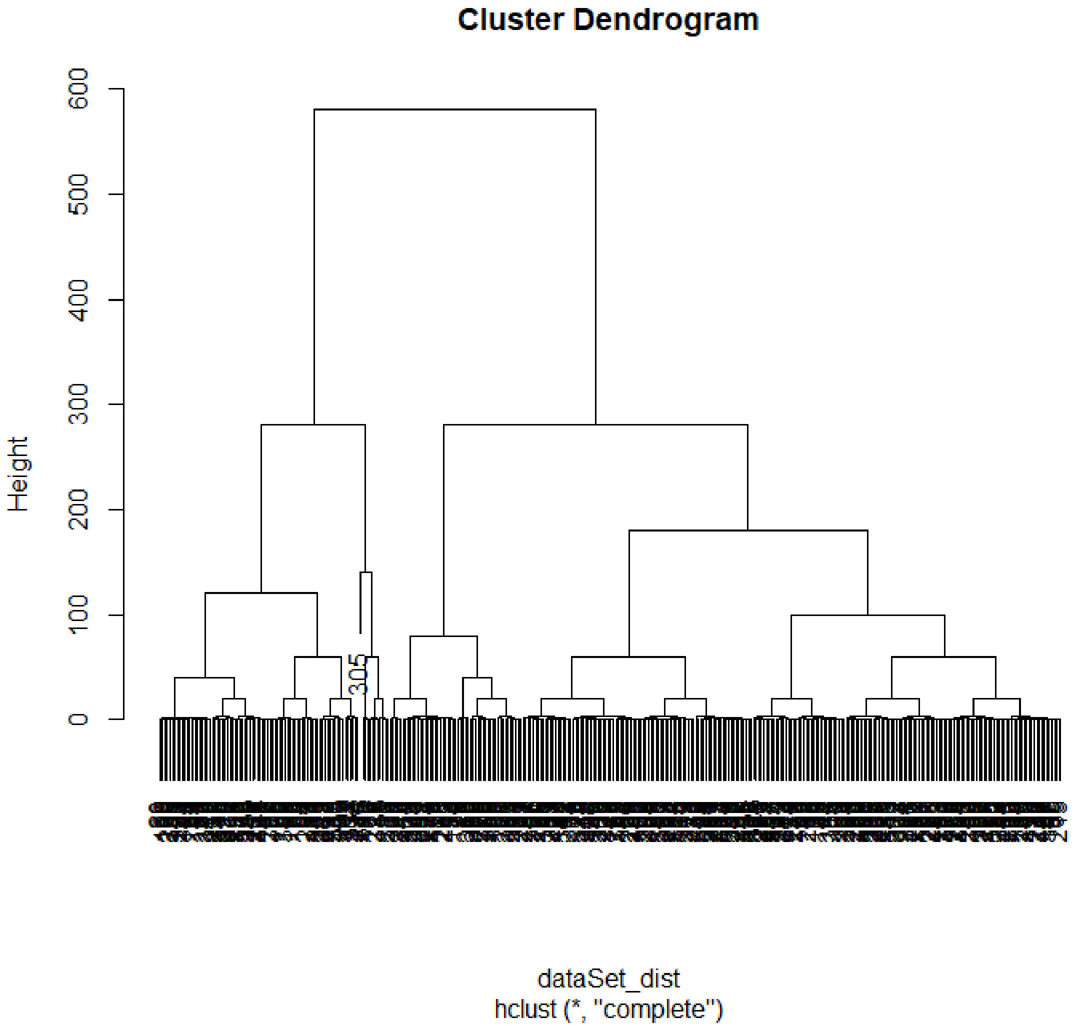


```
hclust_4   0   1
       1  66  14
       2 143  72
       3  30  23
       4  10   2
> rm(list=ls())
> dataSet<-read.csv("C:/Users/prabh/Desktop/Stevens/fall_2020/kdd/final/AdmissionB.csv",na.strings = '?')#Change the path accordingly.
> View(dataSet)
> summary(dataSet)
   Applicant        ADMIT             GRE             GPA             RANK      
 Min.   :1001   Min.   :0.0000   Min.   :220.0   Min.   :2.260   Min.   :1.000  
 1st Qu.:1091   1st Qu.:0.0000   1st Qu.:500.0   1st Qu.:3.130   1st Qu.:2.000  
 Median :1180   Median :0.0000   Median :580.0   Median :3.380   Median :2.000  
 Mean   :1180   Mean   :0.3083   Mean   :586.4   Mean   :3.383   Mean   :2.531  
 3rd Qu.:1270   3rd Qu.:1.0000   3rd Qu.:665.0   3rd Qu.:3.643   3rd Qu.:3.000  
 Max.   :1360   Max.   :1.0000   Max.   :800.0   Max.   :4.000   Max.   :4.000  
> table(dataSet$ADMIT)

  0   1 
249 111 
> #To factor the data set
> dataSet<-na.omit(dataSet)
> dataSet<-dataSet[-1]
> kmeans_4<- kmeans(dataSet[,-1],4,nstart = 10)
> kmeans_4$cluster
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 
  1   3   2   3   4   2   4   1   4   2   2   1   2   2   2   4   2   1   2   4   4   3   3   3   2   2   3   4   2   4   4   2 
 33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64 
  3   2   1   1   4   4   4   4   4   4   3   4   2   1   4   4   1   1   3   1   2   3   3   2   4   1   1   3   3   4   3   3 
 65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96 
  4   3   2   3   4   2   3   1   4   4   2   2   4   2   4   3   2   3   4   1   4   4   3   3   2   3   2   2   2   4   3   3 
 97  98  99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 
  3   4   2   1   1   4   1   4   3   2   2   4   1   4   3   1   1   3   2   3   1   2   2   1   4   4   4   4   2   4   3   2 
129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 
  4   1   3   3   4   4   4   4   4   2   3   3   3   2   3   4   4   1   4   4   4   2   2   1   3   4   3   4   4   4   3   2 
161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 
  3   3   2   4   4   2   1   2   4   3   1   4   3   2   4   3   4   3   3   1   3   4   2   4   4   2   4   4   4   4   3   2 
193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 
  3   1   3   4   3   1   3   4   2   4   2   1   3   2   2   3   4   4   2   4   1   3   3   3   1   1   1   4   4   3   4   2 
225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 
  2   2   3   4   4   2   4   3   1   1   2   3   3   4   4   2   1   4   3   3   4   2   3   1   3   3   3   3   4   4   2   3 
257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 
  4   3   4   3   3   1   4   3   4   1   4   3   3   4   3   4   3   3   4   3   1   4   3   3   3   1   3   4   1   3   2   3 
289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 
  2   1   3   2   3   2   4   4   4   1   4   2   3   3   1   3   1   4   4   4   4   1   4   3   3   4   4   1   1   2   4   4 
321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 
  1   1   4   1   4   3   3   4   4   4   2   3   1   4   1   3   4   3   4   3   4   4   4   4   4   4   3   4   1   3   2   3 
353 354 355 356 357 358 359 360 
  4   2   4   2   2   2   4   2 
> table(kmeans_4$cluster,dataSet[,1])
   
     0  1
  1 48  9
  2 45 28
  3 62 41
  4 94 33
```

For 2 clusters

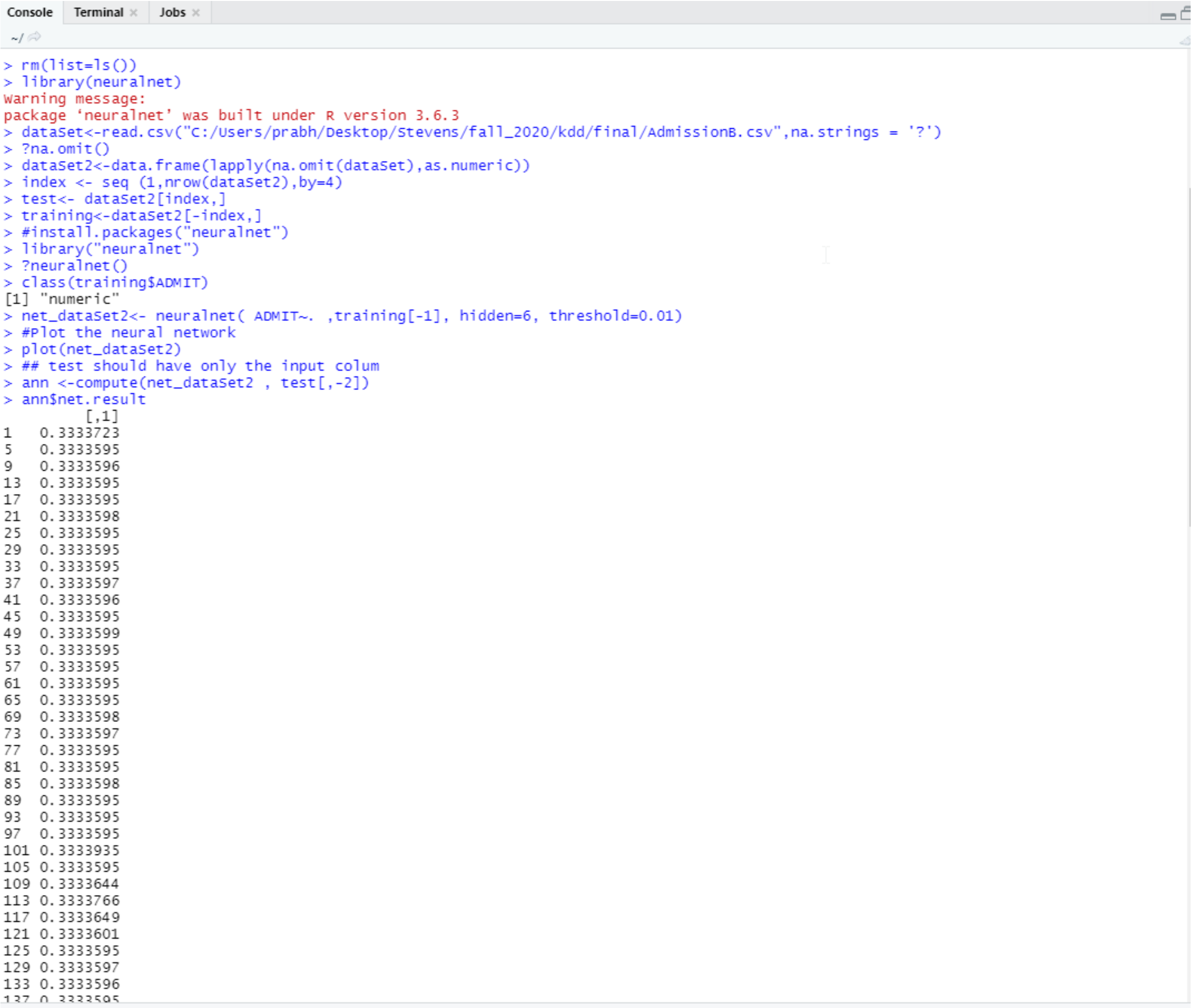


For 4 clusters

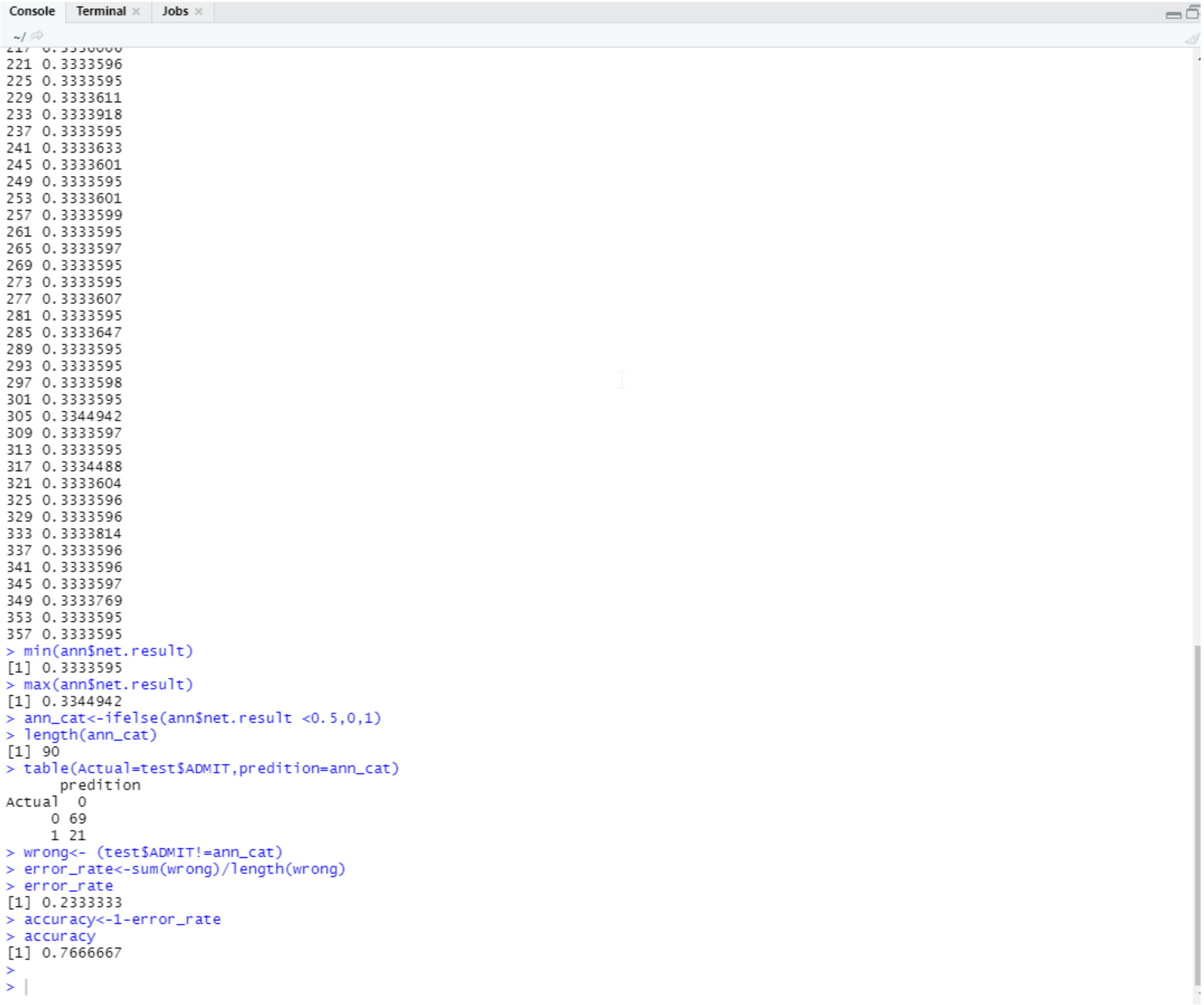


Problem #2: (20 points)

- Load the AdmissionB dataset from CANVAS
- Store every fourth record in a “test” dataset starting with the first record
- Store the rest in the “training” dataset
- Use ANN with 6 hidden nodes to classify applicants (ADMIT=1 vs. 0)
- Measure the performance of the model against the test data.

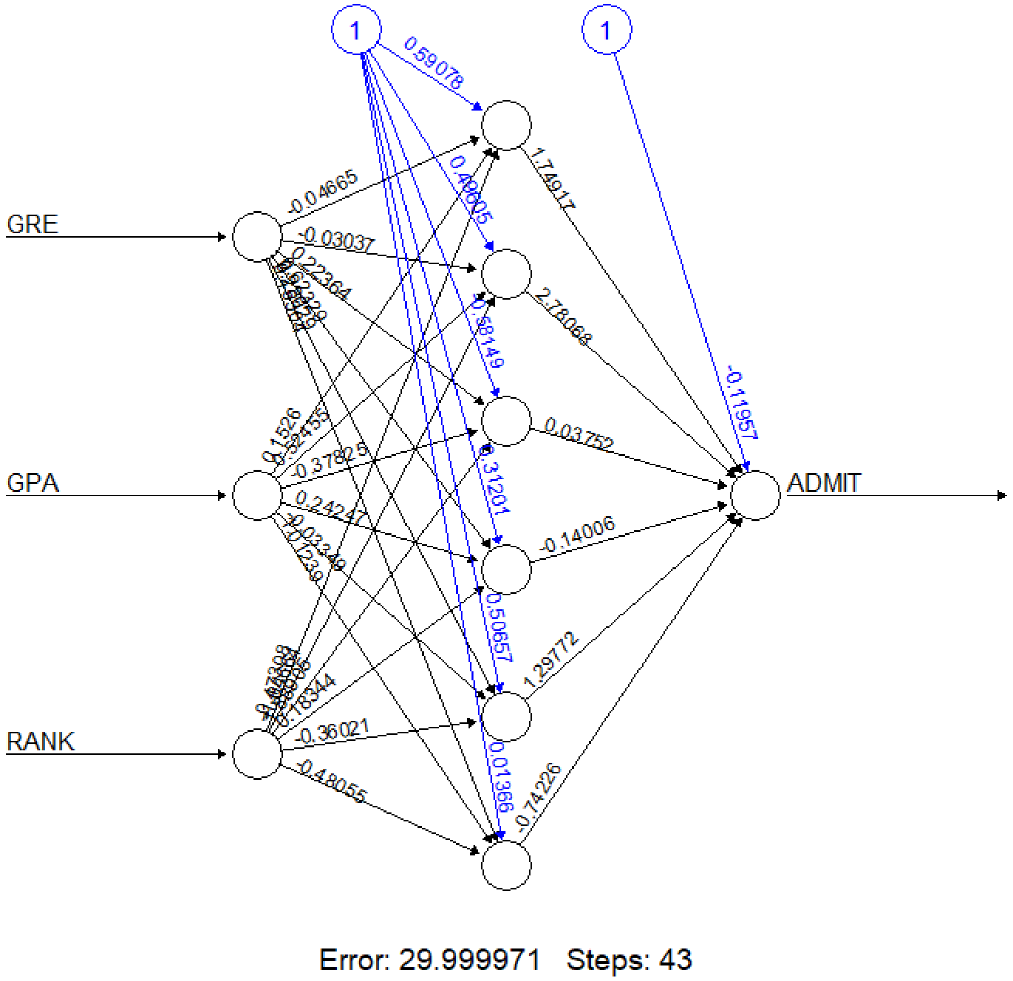


```
> rm(list=ls())
> library(neuralnet)
Warning message:
package ‘neuralnet’ was built under R version 3.6.3
> dataSet<-read.csv("C:/Users/prabh/Desktop/Stevens/fall_2020/kdd/final/AdmissionB.csv",na.strings = '?')
> ?na.omit()
> dataSet2<-data.frame(lapply(na.omit(dataSet),as.numeric))
> index <- seq (1,nrow(dataSet2),by=4)
> test<- dataSet2[index,]
> training<-dataSet2[-index,]
> #install.packages("neuralnet")
> library("neuralnet")
> ?neuralnet()
> class(training$ADMIT)
[1] "numeric"
> net_dataSet2<- neuralnet( ADMIT~. ,training[-1], hidden=6, threshold=0.01)
> #Plot the neural network
> plot(net_dataSet2)
> ## test should have only the input colum
> ann <-compute(net_dataSet2 , test[,-2])
> ann$net.result
         [,1]
1   0.3333723
5   0.3333595
9   0.3333596
13  0.3333595
17  0.3333595
21  0.3333598
25  0.3333595
29  0.3333595
33  0.3333595
37  0.3333597
41  0.3333596
45  0.3333595
49  0.3333599
53  0.3333595
57  0.3333595
61  0.3333595
65  0.3333595
69  0.3333598
73  0.3333597
77  0.3333595
81  0.3333595
85  0.3333598
89  0.3333595
93  0.3333595
97  0.3333595
101 0.3333935
105 0.3333595
109 0.3333644
113 0.3333766
117 0.3333649
121 0.3333601
125 0.3333595
129 0.3333597
133 0.3333596
137 0.3333595
```



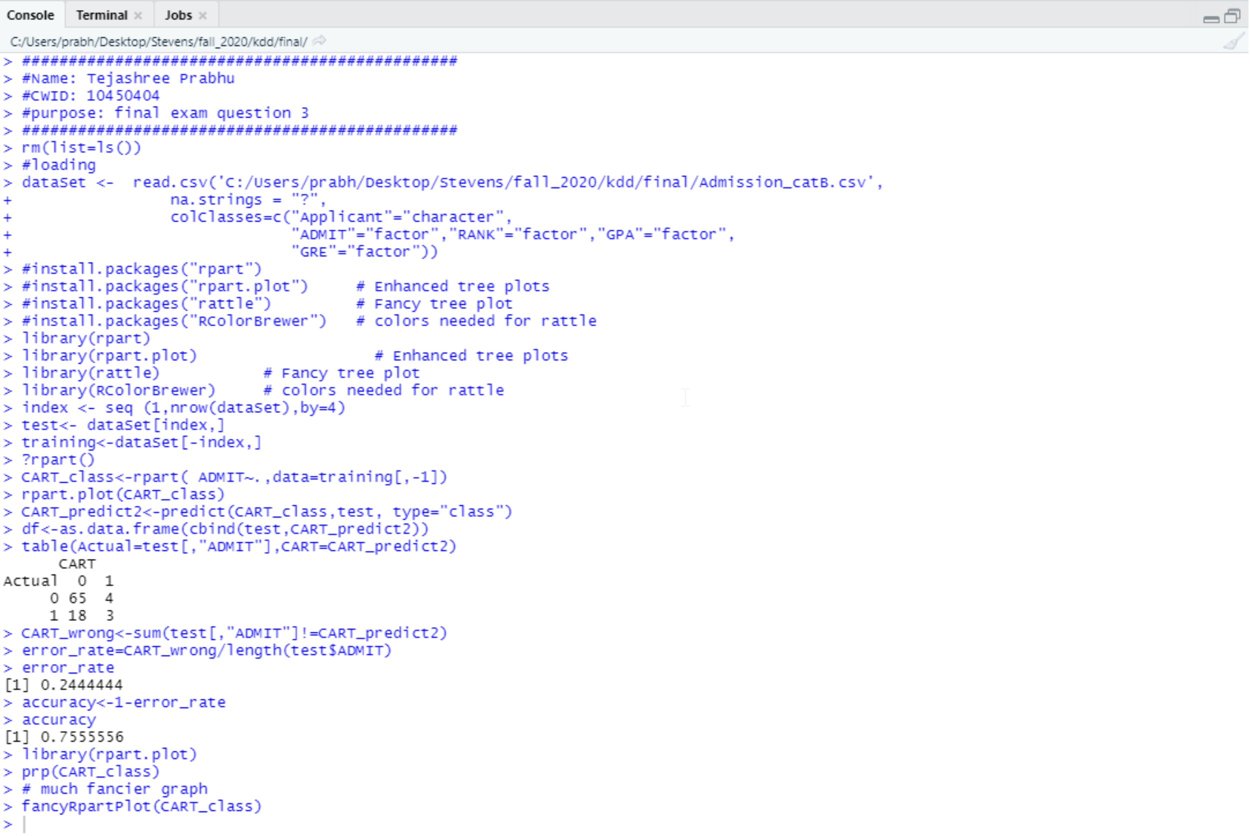
```
217 0.3336000
221 0.3333596
225 0.3333595
229 0.3333611
233 0.3333918
237 0.3333595
241 0.3333633
245 0.3333601
249 0.3333595
253 0.3333601
257 0.3333599
261 0.3333595
265 0.3333597
269 0.3333595
273 0.3333595
277 0.3333607
281 0.3333595
285 0.3333647
289 0.3333595
293 0.3333595
297 0.3333598
301 0.3333595
305 0.3344942
309 0.3333597
313 0.3333595
317 0.3334488
321 0.3333604
325 0.3333596
329 0.3333596
333 0.3333814
337 0.3333596
341 0.3333596
345 0.3333597
349 0.3333769
353 0.3333595
357 0.3333595
> min(ann$net.result)
[1] 0.3333595
> max(ann$net.result)
[1] 0.3344942
> ann_cat<-ifelse(ann$net.result <0.5,0,1)
> length(ann_cat)
[1] 90
> table(Actual=test$ADMIT,predition=ann_cat)
      predition
Actual  0
     0 69
     1 21
> wrong<- (test$ADMIT!=ann_cat)
> error_rate<-sum(wrong)/length(wrong)
> error_rate
[1] 0.2333333
> accuracy<-1-error_rate
> accuracy
[1] 0.7666667
>
> |
```

Final Exam

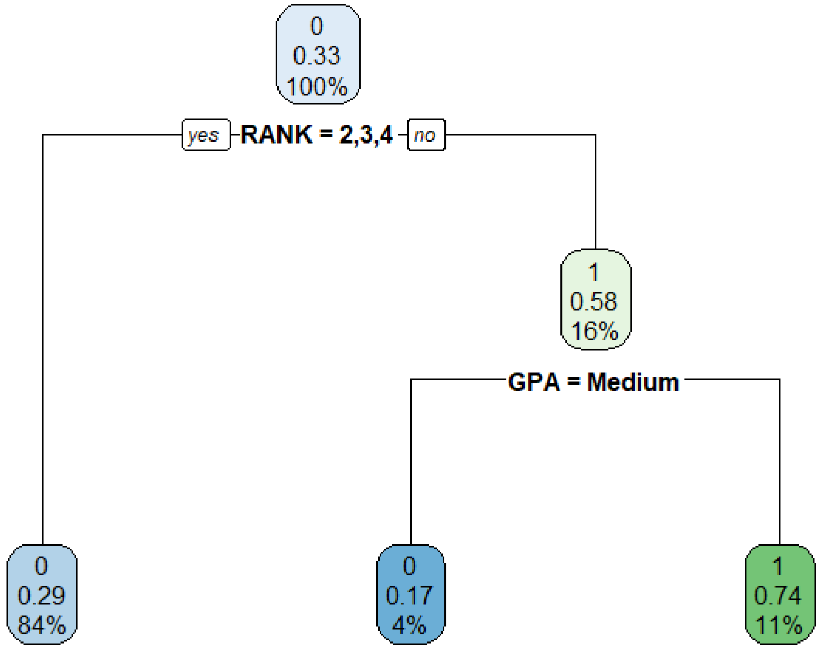

1
1
GRE
GPA
RANK
ADMIT
0.59078
0.40605
-0.58149
0.31201
0.50657
0.01366
-0.11957
1.74917
2.78068
0.03752
-0.14006
1.29772
-0.74226
-0.04665
-0.03037
0.22364
0.1526
-0.37825
0.24247
-0.03349
-0.18344
-0.36021
-0.48055
Error: 29.999971 Steps: 43

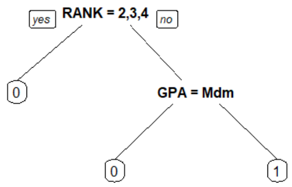
Problem #3: (20 points)

- Load the Admission_catB.csv dataset from CANVAS
- Store every fourth record in a “test” dataset starting with the first record
- Store the rest in the “training” dataset
- Use CART to classify applicants
- Measure the performance of the model against the test data.

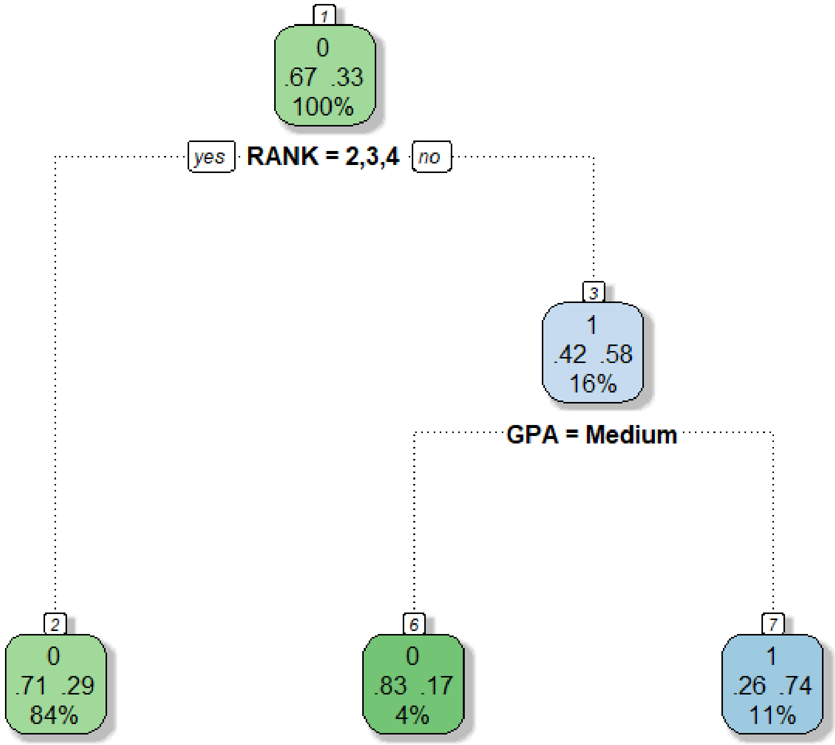


```
> ###############################################
> #Name: Tejashree Prabhu
> #CWID: 10450404
> #purpose: final exam question 3
> ###############################################
> rm(list=ls())
> #loading
> dataSet <-  read.csv('C:/Users/prabh/Desktop/Stevens/fall_2020/kdd/final/Admission_catB.csv',
+                 na.strings = "?",
+                 colClasses=c("Applicant"="character",
+                              "ADMIT"="factor","RANK"="factor","GPA"="factor",
+                              "GRE"="factor"))
> #install.packages("rpart")
> #install.packages("rpart.plot")     # Enhanced tree plots
> #install.packages("rattle")         # Fancy tree plot
> #install.packages("RColorBrewer")   # colors needed for rattle
> library(rpart)
> library(rpart.plot)                   # Enhanced tree plots
> library(rattle)          # Fancy tree plot
> library(RColorBrewer)    # colors needed for rattle
> index <- seq (1,nrow(dataSet),by=4)
> test<- dataSet[index,]
> training<-dataSet[-index,]
> ?rpart()
> CART_class<-rpart( ADMIT~.,data=training[,-1])
> rpart.plot(CART_class)
> CART_predict2<-predict(CART_class,test, type="class")
> df<-as.data.frame(cbind(test,CART_predict2))
> table(Actual=test[,"ADMIT"],CART=CART_predict2)
      CART
Actual  0  1
     0 65  4
     1 18  3
> CART_wrong<-sum(test[,"ADMIT"]!=CART_predict2)
> error_rate=CART_wrong/length(test$ADMIT)
> error_rate
[1] 0.2444444
> accuracy<-1-error_rate
> accuracy
[1] 0.7555556
> library(rpart.plot)
> prp(CART_class)
> # much fancier graph
> fancyRpartPlot(CART_class)
> 
```


0
0.33
100%
yes
RANK = 2,3,4
no
1
0.58
16%
GPA = Medium
0
0.29
84%
0
0.17
4%
1
0.74
11%


yes
RANK = 2,3,4
no
0
GPA = Mdm
0
1

Final Exam

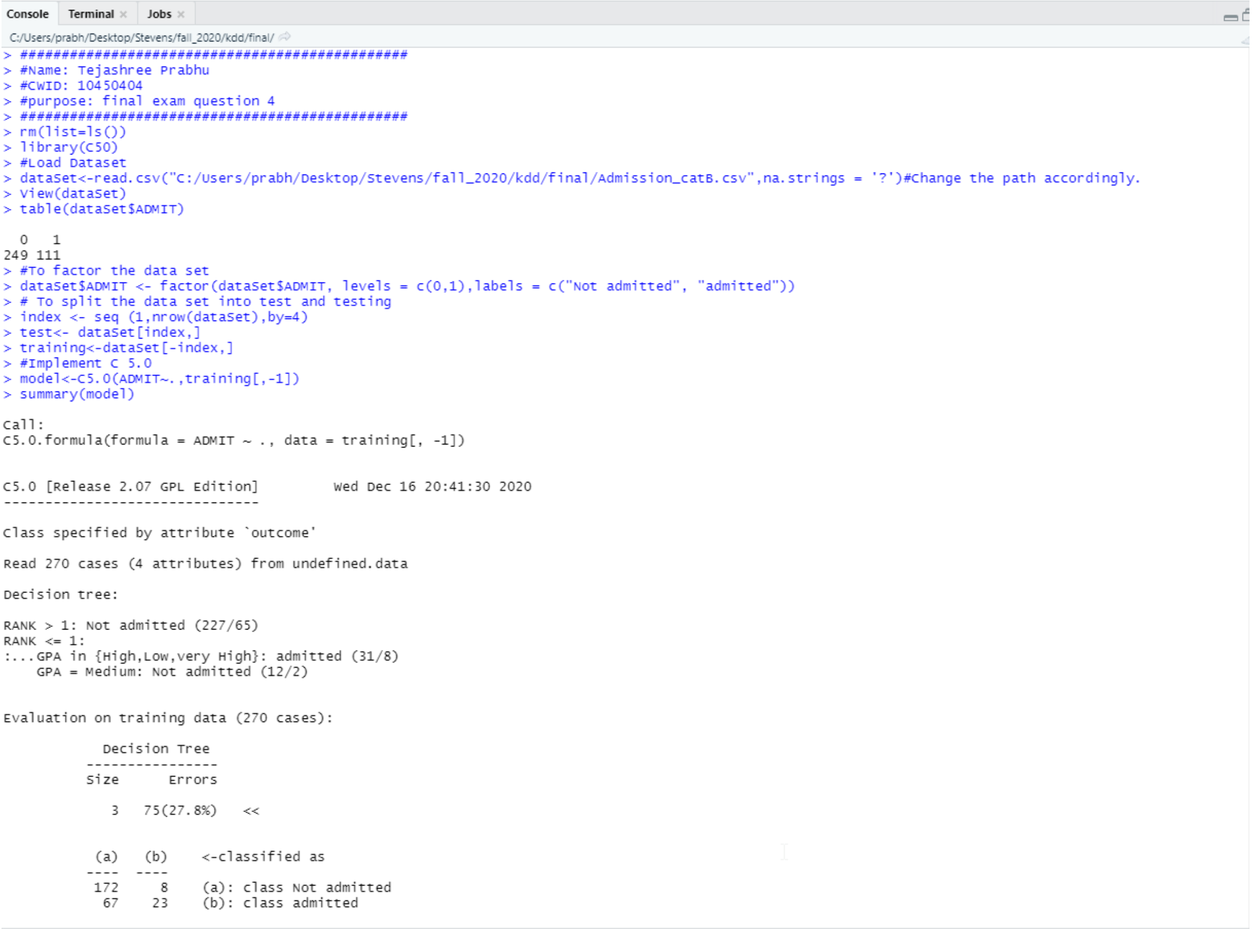


Rattle 2020-Dec-16 20:49:13 prabh

Tejashree Prabhu 10450404

Problem #4: (20 points)

- Load the Admission_catB.csv dataset from CANVAS
- Store every fourth record in a “test” dataset starting with the first record
- Store the rest in the “training” dataset Use C5.0 to classify
- Measure the performance of the model against the test data.

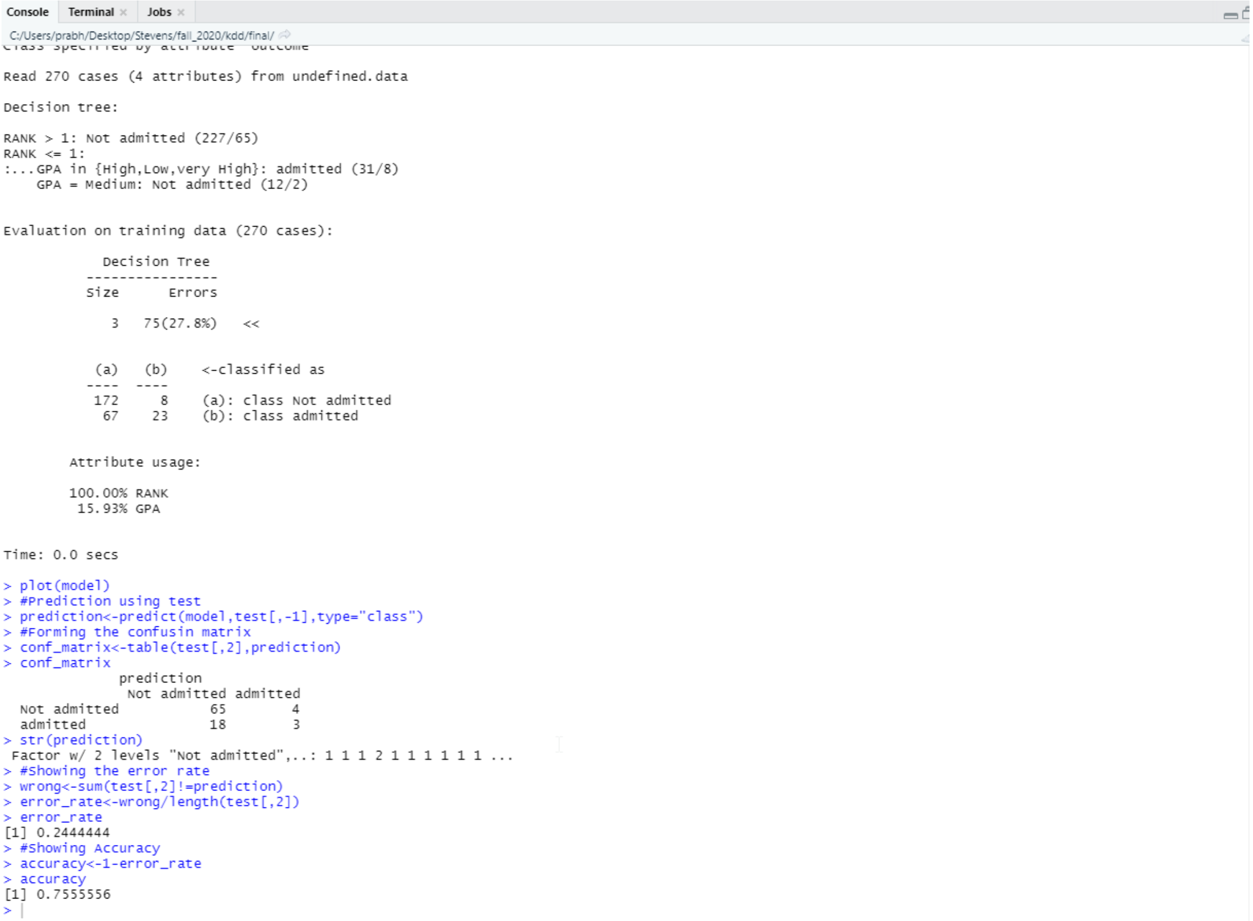


Console Terminal Jobs

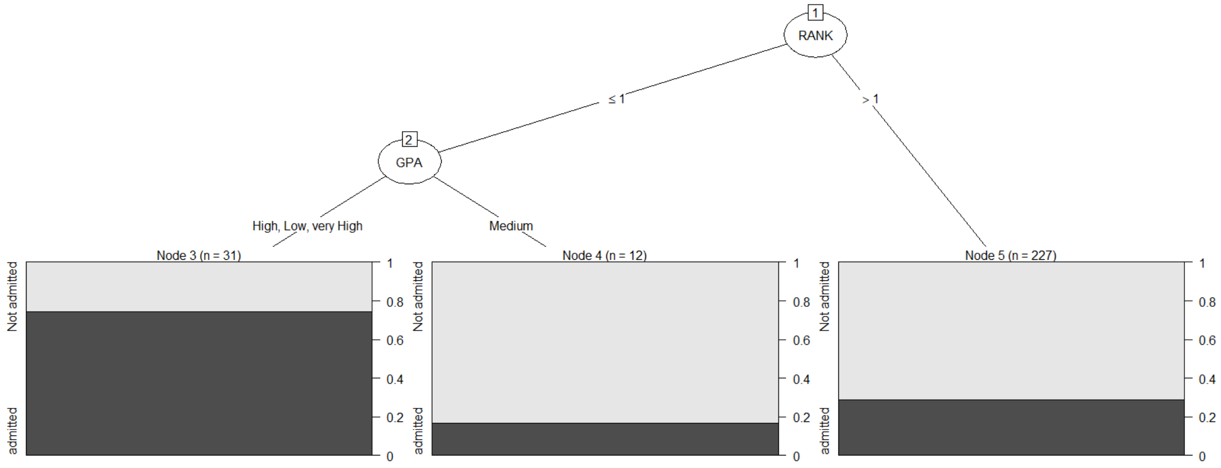
C:/Users/prabh/Desktop/Stevens/fall_2020/kdd/final/

```
> ###############################################
> #Name: Tejashree Prabhu
> #CWID: 10450404
> #purpose: final exam question 4
> ###############################################
> rm(list=ls())
> library(C50)
> #Load Dataset
> dataSet<-read.csv("C:/Users/prabh/Desktop/Stevens/fall_2020/kdd/final/Admission_catB.csv",na.strings = '?')#Change the path accordingly.
> View(dataSet)
> table(dataSet$ADMIT)

  0   1 
249 111 
> #To factor the data set
> dataSet$ADMIT <- factor(dataSet$ADMIT, levels = c(0,1),labels = c("Not admitted", "admitted"))
> # To split the data set into test and testing
> index <- seq (1,nrow(dataSet),by=4)
> test<- dataSet[index,]
> training<-dataSet[-index,]
> #Implement C 5.0
> model<-C5.0(ADMIT~.,training[,-1])
> summary(model)

Call:
C5.0.formula(formula = ADMIT ~ ., data = training[, -1])


C5.0 [Release 2.07 GPL Edition]        Wed Dec 16 20:41:30 2020
-------------------------------

Class specified by attribute `outcome'

Read 270 cases (4 attributes) from undefined.data

Decision tree:

RANK > 1: Not admitted (227/65)
RANK <= 1:
:...GPA in {High,Low,Very High}: admitted (31/8)
    GPA = Medium: Not admitted (12/2)


Evaluation on training data (270 cases):

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

	     3   75(27.8%)   <<


	   (a)   (b)    <-classified as
	  ----  ----
	   172     8    (a): class Not admitted
	    67    23    (b): class admitted
```

Final Exam



```
Class specified by attribute `outcome'

Read 270 cases (4 attributes) from undefined.data

Decision tree:

RANK > 1: Not admitted (227/65)
RANK <= 1:
:...GPA in {High,Low,very High}: admitted (31/8)
    GPA = Medium: Not admitted (12/2)


Evaluation on training data (270 cases):

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

             3   75(27.8%)   <<


           (a)   (b)    <-classified as
          ----  ----
           172     8    (a): class Not admitted
            67    23    (b): class admitted


        Attribute usage:

        100.00% RANK
         15.93% GPA


Time: 0.0 secs

> plot(model)
> #Prediction using test
> prediction<-predict(model,test[,-1],type="class")
> #Forming the confusin matrix
> conf_matrix<-table(test[,2],prediction)
> conf_matrix
              prediction
               Not admitted admitted
  Not admitted           65        4
  admitted               18        3
> str(prediction)
 Factor w/ 2 levels "Not admitted",..: 1 1 1 2 1 1 1 1 1 1 ...
> #Showing the error rate
> wrong<-sum(test[,2]!=prediction)
> error_rate<-wrong/length(test[,2])
> error_rate
[1] 0.2444444
> #Showing Accuracy
> accuracy<-1-error_rate
> accuracy
[1] 0.7555556
> 
```



Tejashree Prabhu 10450404