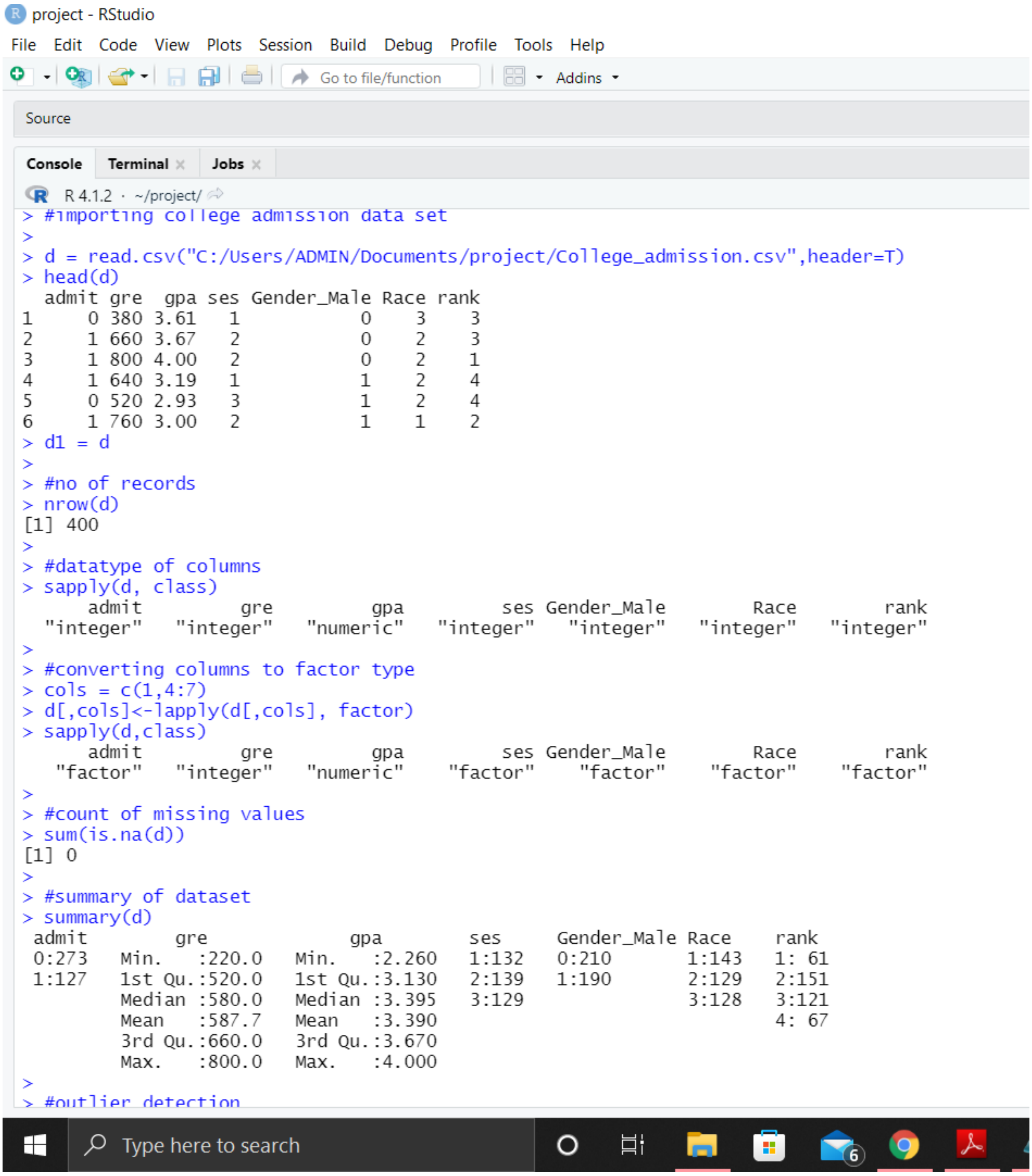
Find the missing values. (if any, perform missing value treatment)




```
> #importing college admission data set
>
> d = read.csv("C:/Users/ADMIN/Documents/project/College_admission.csv",header=T)
> head(d)
  admit gre  gpa ses Gender_Male Race rank
1     0 380 3.61   1           0    3    3
2     1 660 3.67   2           0    2    3
3     1 800 4.00   2           0    2    1
4     1 640 3.19   1           1    2    4
5     0 520 2.93   3           1    2    4
6     1 760 3.00   2           1    1    2
> d1 = d
>
> #no of records
> nrow(d)
[1] 400
>
> #datatype of columns
> sapply(d, class)
      admit         gre         gpa         ses Gender_Male        Race        rank
  "integer"   "integer"   "numeric"   "integer"   "integer"   "integer"   "integer"
>
> #converting columns to factor type
> cols = c(1,4:7)
> d[,cols]<-lapply(d[,cols], factor)
> sapply(d,class)
      admit         gre         gpa         ses Gender_Male        Race        rank
   "factor"   "integer"   "numeric"    "factor"    "factor"    "factor"    "factor"
>
> #count of missing values
> sum(is.na(d))
[1] 0
>
> #summary of dataset
> summary(d)
 admit        gre             gpa         ses     Gender_Male Race    rank
 0:273   Min.   :220.0   Min.   :2.260   1:132   0:210       1:143   1: 61
 1:127   1st Qu.:520.0   1st Qu.:3.130   2:139   1:190       2:129   2:151
         Median :580.0   Median :3.395   3:129               3:128   3:121
         Mean   :587.7   Mean   :3.390                               4: 67
         3rd Qu.:660.0   3rd Qu.:3.670
         Max.   :800.0   Max.   :4.000
>
> #outlier detection
```

Find outliers (if any, then perform outlier treatment)

project - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Source

Console Terminal Jobs

R 4.1.2 · ~/project/

```
         Max.   :800.0   Max.   :4.000
>
> #outlier detection
> #for gre variable
> iqr1=IQR(d$gre)
> iqr1
[1] 140
>
> quantile(d$gre,na.rm=TRUE)
  0%  25%  50%  75% 100%
 220  520  580  660  800
>
> maz1 = 660+1.5*iqr1
> maz1
[1] 870
>
> min1= 520-1.5*iqr1
> min1
[1] 310
>
> #all the pointers above the upperinner fence
> print(which(d$gre > maz1))              #no outliers
integer(0)
>
> # all the pointers below the lowerinner fence
> print(which(d$gre < min1))              #4 outliers
[1]  72 180 305 316
>
> # for gpa variables
> iqr2= IQR(d$gpa)
> iqr2
[1] 0.54
>
> quantile(d$gpa,na.rm=TRUE)
   0%    25%    50%    75%   100%
2.260 3.130 3.395 3.670 4.000
>
> max2 = 3.67+1.5*iqr2
> max2
[1] 4.48
>
> min2 = 3.13-1.5*iqr2
> min2
[1] 2.32
```

Type here to search

Console | Terminal | Jobs

R 4.1.2 · ~/project/

```
>
> # all the pointers above the upperinner fence
> print(which(d$gpa > max2))                 # no outlier
integer(0)
>
> print(which(d$gpa < min2))                 # 1 outlier
[1] 290
>
> #removing outliers
>
> d=d[-c(72, 180, 290, 305, 316),]
> nrow(d)
[1] 395
>
```

Splitting the data into train and test

```
> #splitting of data set into train and test
>
> set.seed(0)
> library("caTools")
>
> set.seed(0)
> library("caTools")
> d[,2:3]=scale(d[,2:3])
> split=sample.split(d$admit,SplitRatio = .75)
> train=subset(d,split==T)
> test=subset(d,split==F)
>
```

Run logistic model to determine the factors that influence the admission process of a student (Drop insignificant variables)

project - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Source

Console Terminal Jobs

R 4.1.2 · ~/project/

```
> #logistic regression
> logit1=glm(admit~.,train,family='binomial')    #all variables
> summary(logit1)

Call:
glm(formula = admit ~ ., family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8458  -0.8294  -0.5794   0.9459   2.1850

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.03714    0.45482   2.280  0.02259 *
gre           0.27452    0.15262   1.799  0.07206 .
gpa           0.51793    0.16125   3.212  0.00132 **
ses2         -0.38459    0.33468  -1.149  0.25050
ses3         -0.40768    0.34356  -1.187  0.23538
Gender_Male1 -0.09887    0.27541  -0.359  0.71959
Race2        -0.34389    0.33981  -1.012  0.31153
Race3        -0.43592    0.33303  -1.309  0.19054
rank2        -1.29613    0.40854  -3.173  0.00151 **
rank3        -1.70399    0.43413  -3.925 8.67e-05 ***
rank4        -2.05159    0.51218  -4.006 6.19e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 370.01  on 295  degrees of freedom
Residual deviance: 319.50  on 285  degrees of freedom
AIC: 341.5

Number of Fisher Scoring iterations: 4

>
> # in the above model gre,ses,gender_male and race variable are not significant
> # building new model with only gpa rank variables
>
> logit2=glm(admit~gpa+rank,train,family = 'binomial')       # all the variables
> summary(logit2)

Call:
```

Type here to search

Second logistic model by removing insignificant variables

project - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Source

Console Terminal Jobs

R 4.1.2 · ~/project/

```
>
> # in the above model gre,ses,gender_male and race variable are not significant
> # building new model with only gpa rank variables
>
> logit2=glm(admit~gpa+rank,train,family = 'binomial')       # all the variables
> summary(logit2)

Call:
glm(formula = admit ~ gpa + rank, family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8203  -0.8527  -0.5997   1.0019   2.2480

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.4765     0.3386   1.407  0.15931
gpa           0.6037     0.1479   4.081 4.49e-05 ***
rank2        -1.2261     0.3978  -3.082  0.00205 **
rank3        -1.7363     0.4248  -4.087 4.37e-05 ***
rank4        -2.0552     0.5038  -4.079 4.52e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 370.01  on 295  degrees of freedom
Residual deviance: 326.90  on 291  degrees of freedom
AIC: 336.9

Number of Fisher Scoring iterations: 4

>
> # Accuracy of logit model
>
> predicted_val1 = predict(logit1,test,type="response")
>
> test$pred_admit1= ifelse(predicted_val1>0.5,1,0)
>
> #confusion matrix
>
> conf_mat1=table(predicted=test$pred_admit1,actual=test$admit)
> conf_mat1
```

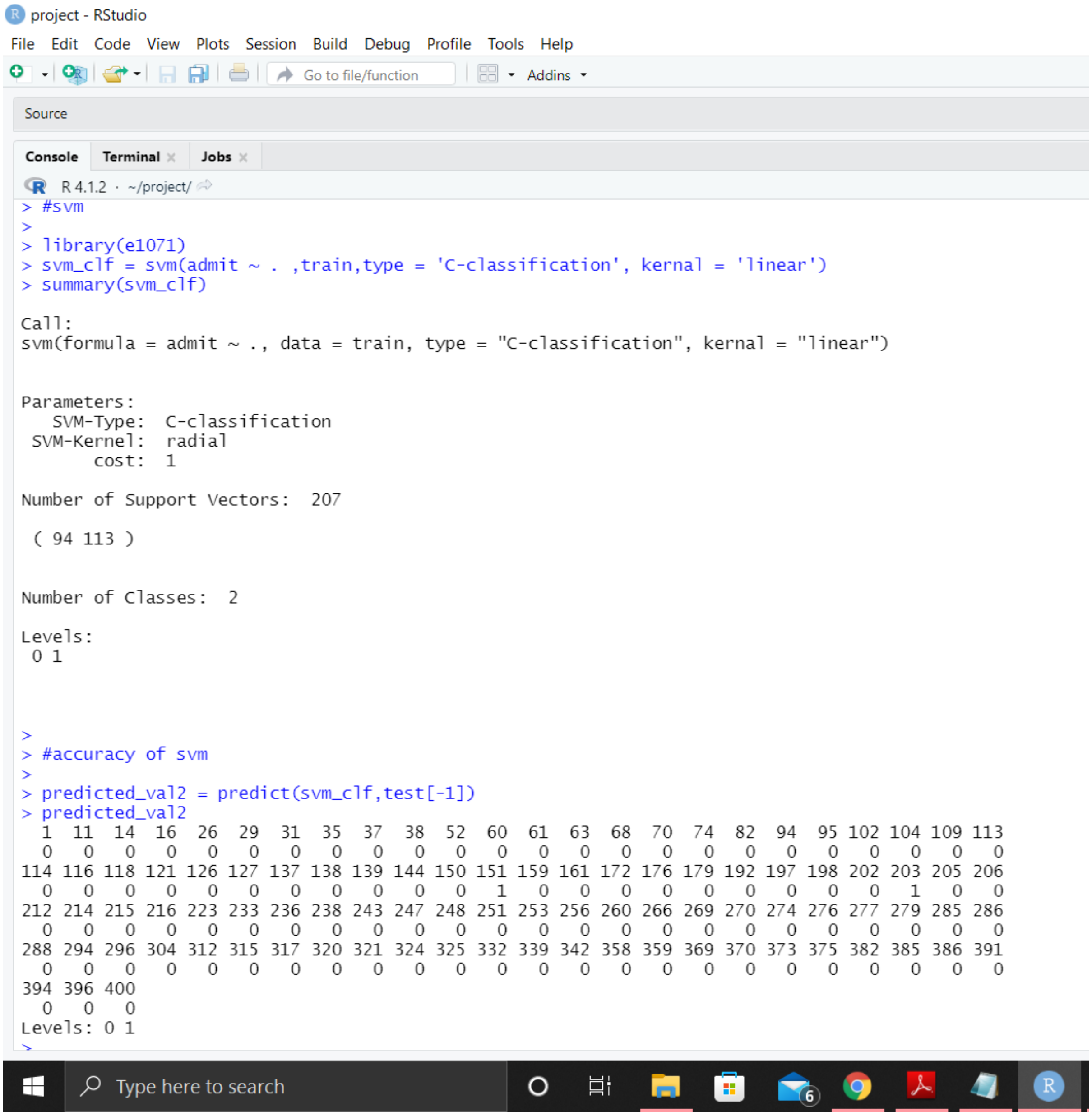
Type here to search

Here residual deviation increases so we will use first model

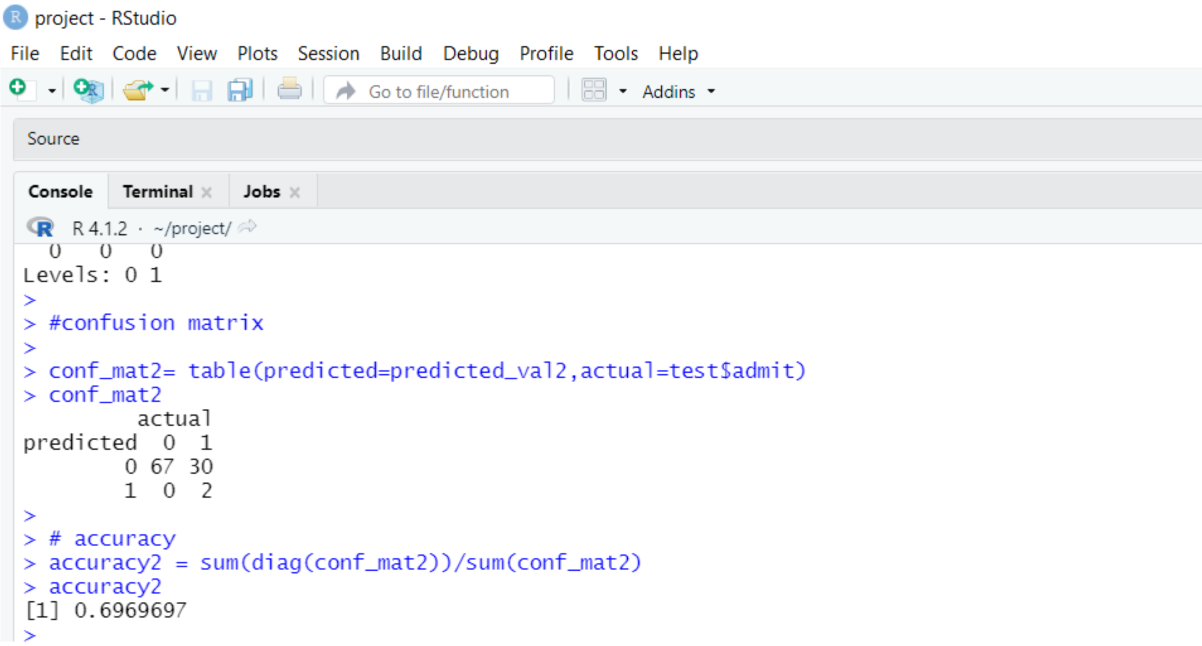
Accuracy of Logistic model

```
>
> # Accuracy of logit model
>
> predicted_val1 = predict(logit1,test,type="response")
>
> test$pred_admit1= ifelse(predicted_val1>0.5,1,0)
>
> #confusion matrix
>
> conf_mat1=table(predicted=test$pred_admit1,actual=test$admit)
> conf_mat1
         actual
predicted  0  1
        0 55 26
        1 12  6
>
> # accuracy
> accuracy1 = sum(diag(conf_mat1))/sum(conf_mat1)
> accuracy1
[1] 0.6161616
>
```

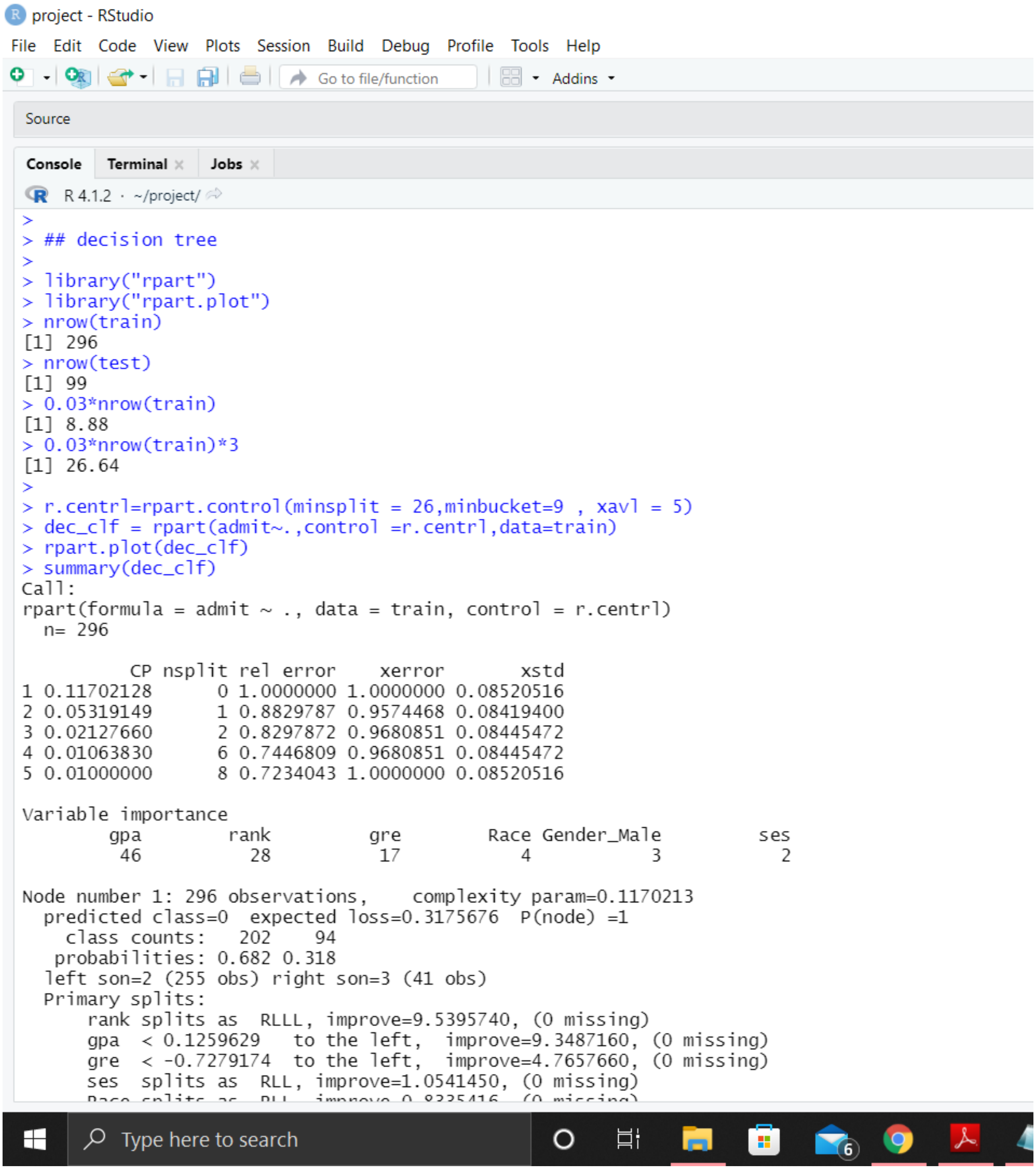
Try other modelling techniques like decision tree and SVM and select a champion model



Accuracy of SVM Model



Decision tree model



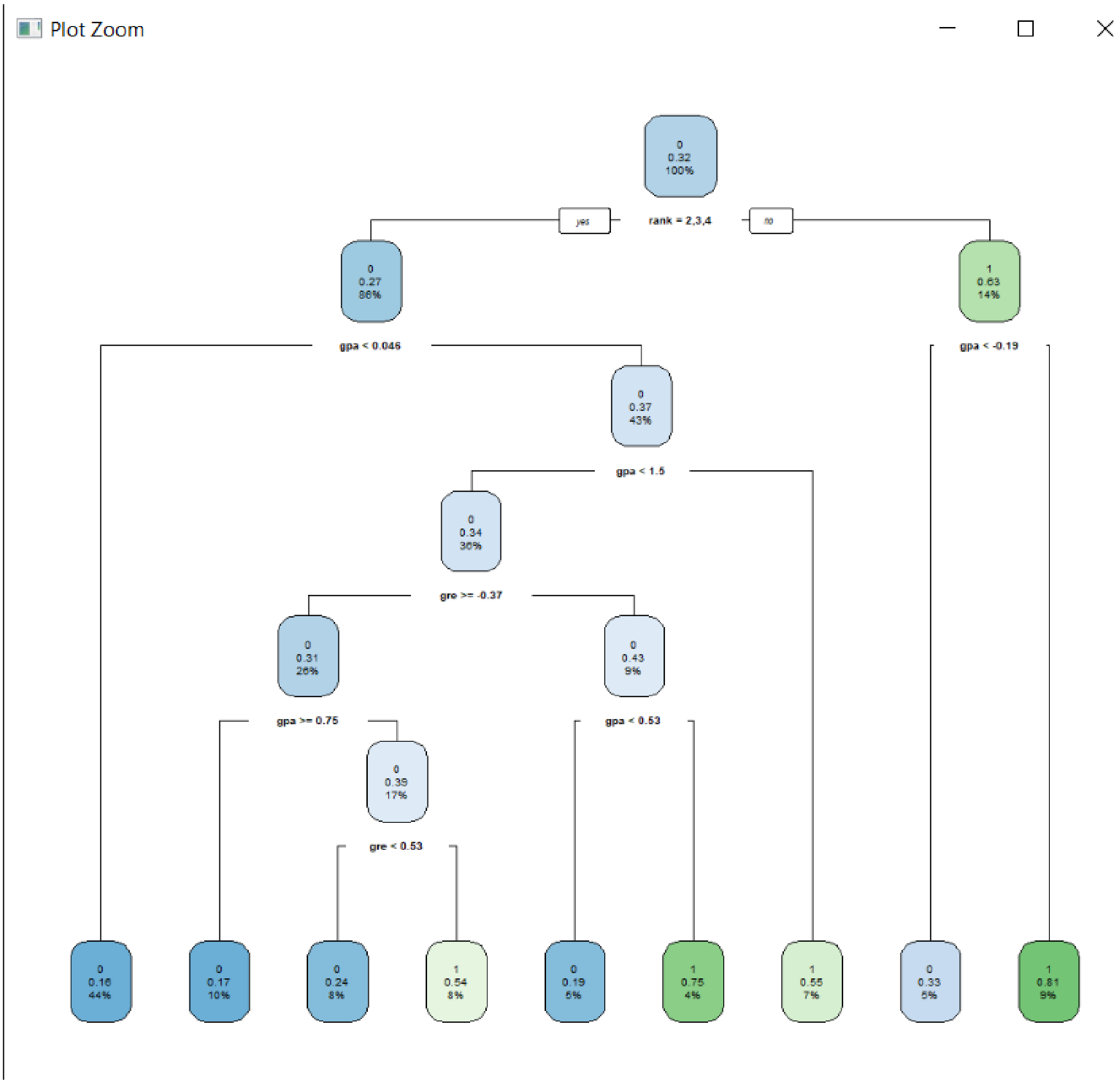

```
>
> ## decision tree
>
> library("rpart")
> library("rpart.plot")
> nrow(train)
[1] 296
> nrow(test)
[1] 99
> 0.03*nrow(train)
[1] 8.88
> 0.03*nrow(train)*3
[1] 26.64
>
> r.centrl=rpart.control(minsplit = 26,minbucket=9 , xavl = 5)
> dec_clf = rpart(admit~.,control =r.centrl,data=train)
> rpart.plot(dec_clf)
> summary(dec_clf)
Call:
rpart(formula = admit ~ ., data = train, control = r.centrl)
  n= 296

          CP nsplit rel error    xerror       xstd
1 0.11702128      0 1.0000000 1.0000000 0.08520516
2 0.05319149      1 0.8829787 0.9574468 0.08419400
3 0.02127660      2 0.8297872 0.9680851 0.08445472
4 0.01063830      6 0.7446809 0.9680851 0.08445472
5 0.01000000      8 0.7234043 1.0000000 0.08520516

Variable importance
        gpa        rank         gre        Race Gender_Male         ses
         46          28          17           4           3           2

Node number 1: 296 observations,    complexity param=0.1170213
  predicted class=0  expected loss=0.3175676  P(node) =1
    class counts:   202    94
   probabilities: 0.682 0.318
  left son=2 (255 obs) right son=3 (41 obs)
  Primary splits:
      rank splits as  RLLL, improve=9.5395740, (0 missing)
      gpa  < 0.1259629   to the left,  improve=9.3487160, (0 missing)
      gre  < -0.7279174  to the left,  improve=4.7657660, (0 missing)
      ses  splits as  RLL, improve=1.0541450, (0 missing)
```

Decision tree



Accuracy of Decision Tree

project - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Source

Console Terminal Jobs

R 4.1.2 · ~/project/

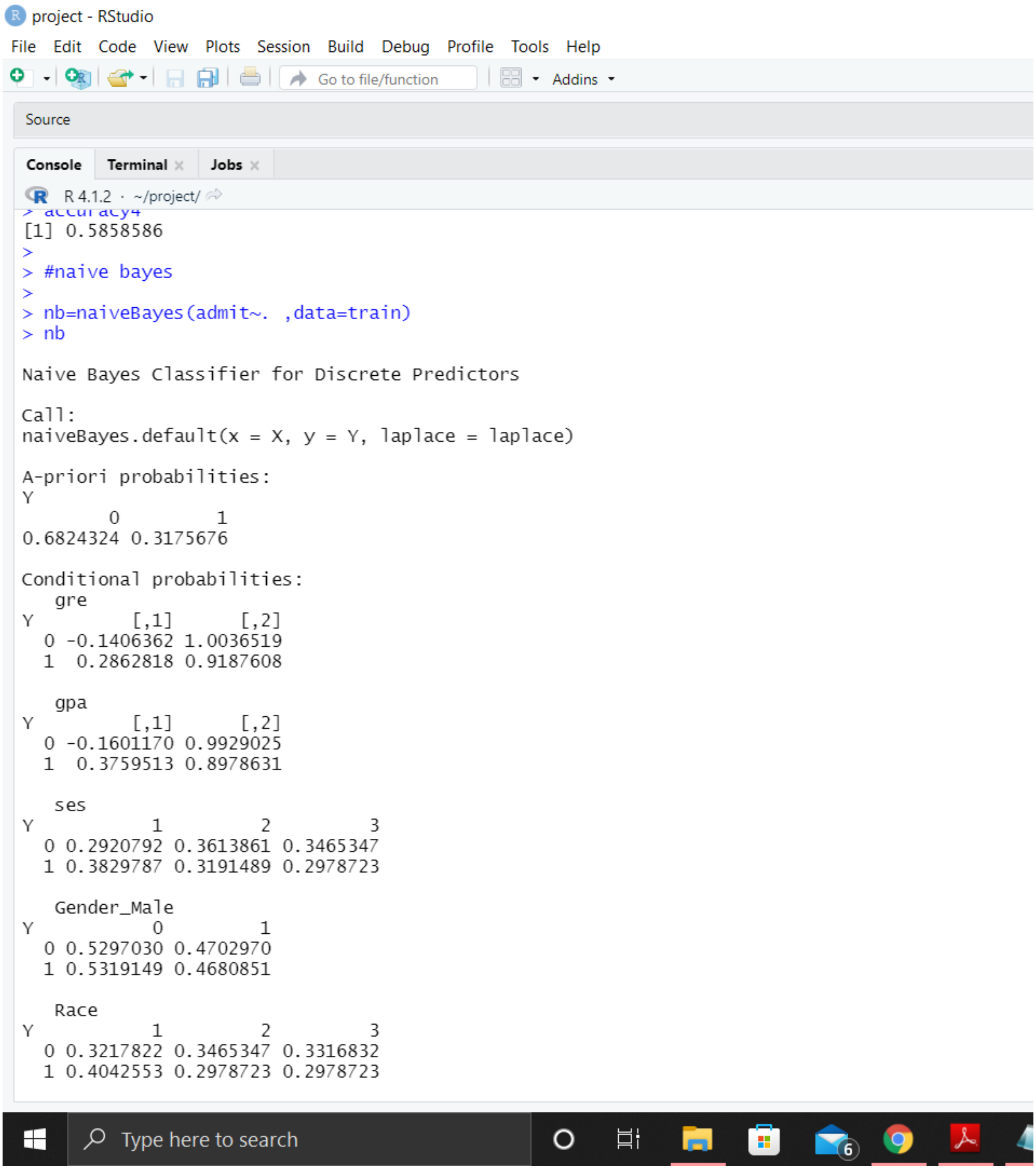
```
   probabilities: 0.458 0.542

>
> #accuracy of decision tree
>
> predicted_val3 = predict(dec_clf,test[-1],type="class")
> predicted_val3
  1  11  14  16  26  29  31  35  37  38  52  60  61  63  68  70  74  82  94  95 102 104 109 113
  1   1   0   0   1   0   1   0   0   0   0   0   0   0   0   1   1   0   0   1   0   1   0   0
114 116 118 121 126 127 137 138 139 144 150 151 159 161 172 176 179 192 197 198 202 203 205 206
  0   1   0   1   0   1   0   1   0   0   0   1   1   0   0   0   0   1   0   0   0   1   1   0
212 214 215 216 223 233 236 238 243 247 248 251 253 256 260 266 269 270 274 276 277 279 285 286
  0   0   0   0   0   0   0   1   0   0   0   0   1   0   0   0   1   0   0   0   1   0   0   0
288 294 296 304 312 315 317 320 321 324 325 332 339 342 358 359 369 370 373 375 382 385 386 391
  0   1   0   1   1   0   0   0   0   0   0   1   0   0   0   0   1   0   0   0   0   0   0   0
394 396 400
  0   1   0
Levels: 0 1
>
> #confusion matrix
>
> conf_mat3 = table(predicted=predicted_val3,actual=test$admit)
> conf_mat3
         actual
predicted  0  1
        0 50 22
        1 17 10
>
> #accuracy
>
> accuracy3= sum(diag(conf_mat3))/sum(conf_mat3)
> accuracy3
[1] 0.6060606
>
```

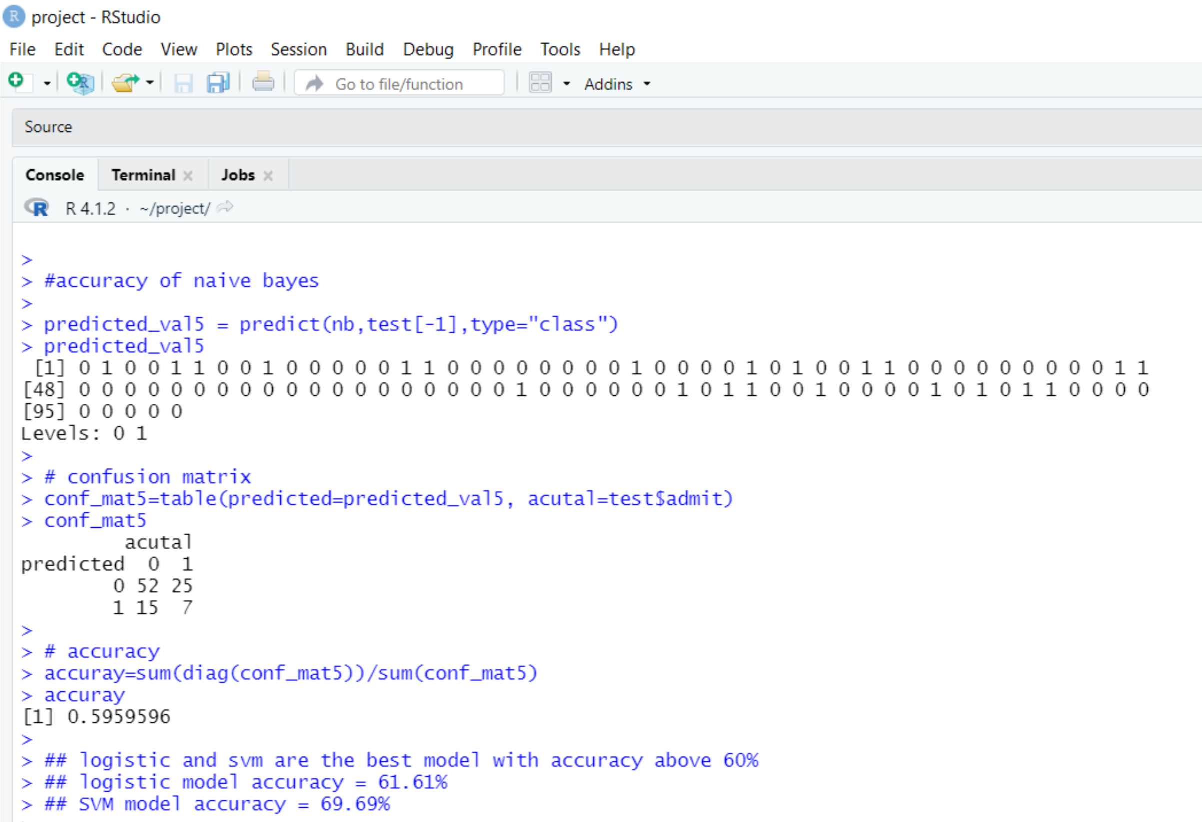
KNN and its Accuracy

```
project - RStudio
File  Edit  Code  View  Plots  Session  Build  Debug  Profile  Tools  Help
Go to file/function    Addins
Source
Console  Terminal  Jobs
R 4.1.2 · ~/project/
> accuracy3
[1] 0.6060606
>
> #KNN and its accuracy
>
> library("class")
> knn = knn(train, test[-1],train$admit, k=19)
> knn
 [1] 0 1 1 0 1 1 0 0 0 0 0 0 1 0 0 1 0 1 1 1 0 0 0 0 0 1 1 0 0 0 0 1 1 1 1 1 0 1 1 1 1 1 0 0 1 0 0
[48] 1 1 1 1 1 0 0 1 0 1 1 0 1 0 0 0 0 1 0 1 0 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0
[95] 0 1 0 0 0
Levels: 0 1
>
> #confussion matrix
>
> conf_mat4=table(predicted=knn,actual=test$admit)
> conf_mat4
         actual
predicted  0  1
        0 41 15
        1 26 17
>
> # accuracy
>
> accuracy4= sum(diag(conf_mat4))/sum(conf_mat4)
> accuracy4
[1] 0.5858586
```

Naïve Bayes


project - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Source
Console
Terminal
Jobs
R 4.1.2 · ~/project/
[1] 0.5858586
>
> #naive bayes
>
> nb=naiveBayes(admit~. ,data=train)
> nb
Naive Bayes Classifier for Discrete Predictors
Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)
A-priori probabilities:
Y
0 1
0.6824324 0.3175676
Conditional probabilities:
gre
Y [,1] [,2]
0 -0.1406362 1.0036519
1 0.2862818 0.9187608
gpa
Y [,1] [,2]
0 -0.1601170 0.9929025
1 0.3759513 0.8978631
ses
Y 1 2 3
0 0.2920792 0.3613861 0.3465347
1 0.3829787 0.3191489 0.2978723
Gender_Male
Y 0 1
0 0.5297030 0.4702970
1 0.5319149 0.4680851
Race
Y 1 2 3
0 0.3217822 0.3465347 0.3316832
1 0.4042553 0.2978723 0.2978723
Type here to search
6

Accuracy of **Naïve Bayes**



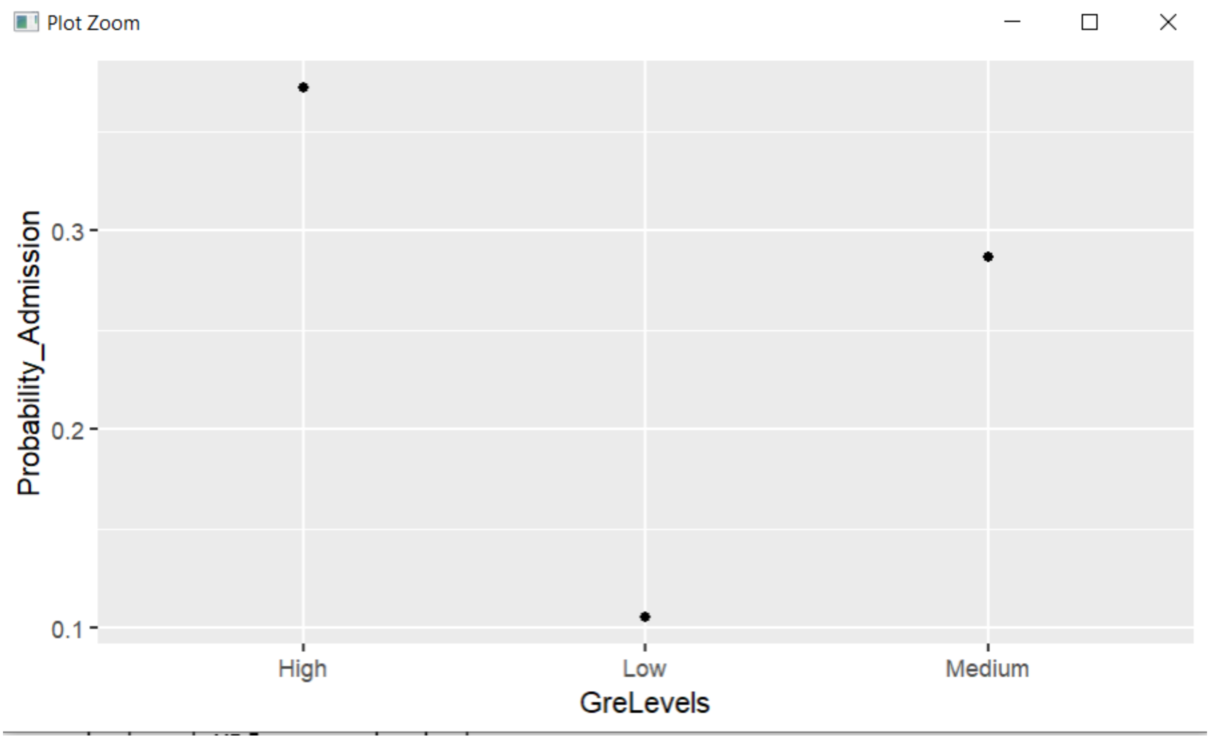

```
>
> #accuracy of naive bayes
>
> predicted_val5 = predict(nb,test[-1],type="class")
> predicted_val5
 [1] 0 1 0 0 1 1 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1
[48] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 0 0 1 0 0 0 0 1 0 1 0 1 1 0 0 0 0
[95] 0 0 0 0 0
Levels: 0 1
>
> # confusion matrix
> conf_mat5=table(predicted=predicted_val5, acutal=test$admit)
> conf_mat5
         acutal
predicted  0  1
        0 52 25
        1 15  7
>
> # accuracy
> accuray=sum(diag(conf_mat5))/sum(conf_mat5)
> accuray
[1] 0.5959596
>
> ## logistic and svm are the best model with accuracy above 60%
> ## logistic model accuracy = 61.61%
> ## SVM model accuracy = 69.69%
```

Categorize the average of grade point into High, Medium, and Low (with admission probability percentages) and plot it on a point chart.

```
> #Descriptiven Categorize the average of grade point into High, Medium, and Low
>
> Descriptive = transform(d1,GreLevels=ifelse(gre<440,"Low",ifelse(gre<580,"Medium","High")))
> View(Descriptive)
> Sum_Desc=aggregate(admit~GreLevels,Descriptive,FUN=sum)
> length_Desc=aggregate(admit~GreLevels,Descriptive,FUN=length)
> Probability_table = cbind(Sum_Desc,Recs=length_Desc[,2])
> Probability_table_final = transform(Probability_table,Probability_Admission=admit/Recs)
> Probability_table_final
  GreLevels admit Recs Probability_Admission
1      High    84  226             0.3716814
2       Low     4   38             0.1052632
3    Medium    39  136             0.2867647
>
> library("ggplot2")
> ggplot(Probability_table_final,aes(x=GreLevels , y=Probability_Admission))+geom_point()
>
> #Cross grid for admission variable with GRE categorized
>
> table(Descriptive$admit,Descriptive$GreLevels)

    High Low Medium
  0  142  34     97
  1   84   4     39
>
>
>
> |
```

Point Chart



Descriptive showing GreLevels

project - RStudio

	admit	gre	gpa	ses	Gender_Male	Race	rank	GreLevels
1	0	380	3.61	1	0	3	3	Low
2	1	660	3.67	2	0	2	3	High
3	1	800	4.00	2	0	2	1	High
4	1	640	3.19	1	1	2	4	High
5	0	520	2.93	3	1	2	4	Medium
6	1	760	3.00	2	1	1	2	High
7	1	560	2.98	2	1	2	1	Medium
8	0	400	3.08	2	0	2	2	Low
9	1	540	3.39	1	1	1	3	Medium
10	0	700	3.92	1	0	2	2	High
11	0	800	4.00	1	1	1	4	High
12	0	440	3.22	3	0	2	1	Medium
13	1	760	4.00	3	1	2	1	High
14	0	700	3.08	2	0	2	2	High
15	1	700	4.00	2	1	1	1	High
16	0	480	3.44	3	0	1	3	Medium
17	0	780	3.87	2	0	3	4	High
18	0	360	2.56	3	1	3	3	Low
19	0	800	3.75	1	1	3	2	High
20	1	540	3.81	1	0	3	1	Medium

Showing 1 to 20 of 400 entries, 8 total columns