NAME: RUDRADITYO SAHA

REGISTRATION NUMBER: 16BCE1062

CSE 3025 DIGITAL ASSIGNMENT 1

NAIVE BAYES CLASSIFICATION

INTRODUCTION:

Naive Bayes Classification is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. It computes the conditional a-posterior probabilities of a categorical class variable. Naive Bayes model is easy to build and particularly useful for very large datasets.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c)through the equation:

$$\underbrace{P(c \mid x)}_{\text{Posterior Probability}} = \frac{\overbrace{P(x \mid c)}^{\text{Likelihood}}\,\overbrace{P(c)}^{\text{Class Prior Probability}}}{\underbrace{P(x)}_{\text{Predictor Prior Probability}}}$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

- P(c|x) is the posterior probability of class (c,target) given predictor (x,attributes)
- P(c) is the prior probability of class
- P(x|c) is the likelihood which is the probability of predictor given class
- P(x) is the prior probability of predictor

TOOL USED: R

DATASET

The dataset contains information about admission of a student to an institute depending on the cumulative score of gre, gpa and rank of university from where the student is coming from.

The dataset contains of 4 columns with 400 rows. The four columns being “admit”,”gre”,”gpa” and “rank”. Each row shows the data about a particular student.

The column “admit“ is a categorical variable with only two values ‘0’ and ‘1’. ‘0’ indicates that the student is not admitted to the institute and ‘1’ indicates that the student is admitted to the institute.

The column “admit” also the class variable in the dataset.

The column “gre” contains gre scores as integer values.

The column “gpa” contains gpa scores as floating point values.

The column “rank” is a categorical variable having values ranging from ‘1’ to ‘4’ with ’1’ indicating the highest rank and ‘4’ indicating the lowest rank.

Data Link:
https://www.youtube.com/redirect?redir_token=jKws6gvs8mm4TeeUCqD9RcHnwvx8MTUzNTg4NDU3M0AxNTM1Nzk4MTcz&event=video_description&v=RLjSQdcg8AM&q=https%3A%2F%2Fgoo.gl%2FnCFX1x

R CODE WITH COMMENTS

1) LIBRARIES

```
#libraries

library(naivebayes)

library(dplyr)

library(ggplot2)

library(psych)
```

2) DATASET

CODE:

data <- read.csv(file.choose(), header = T) #To read data file

str(data) #To view structure of dataset

View(data)

OUTPUT:

```
> str(data)  #To view structure of dataset
'data.frame':   400 obs. of  4 variables:
 $ admit: int  0 1 1 1 0 1 1 0 1 0 ...
 $ gre  : int  380 660 800 640 520 760 560 400 540 700 ...
 $ gpa  : num  3.61 3.67 4 3.19 2.93 3 2.98 3.08 3.39 3.92 ...
 $ rank : int  3 3 1 4 4 2 1 2 3 2 ...
```

Filter

	admit	gre	gpa	rank
1	0	380	3.61	3
2	1	660	3.67	3
3	1	800	4.00	1
4	1	640	3.19	4
5	0	520	2.93	4
6	1	760	3.00	2
7	1	560	2.98	1
8	0	400	3.08	2
9	1	540	3.39	3

Showing 1 to 9 of 400 entries

CODE:

data$rank <- as.factor(data$rank) #To convert rank variable from integer to factor variable

data$admit <- as.factor(data$admit) #To convert admit variable from integer to factor variable

OUTPUT:

```
> data$rank <- as.factor(data$rank)  #To convert rank variable from integer to
factor variable
> data$admit <- as.factor(data$admit) #To convert admit variable from integer t
o factor variable
> str(data)  #To view structure of dataset
'data.frame':   400 obs. of  4 variables:
 $ admit: Factor w/ 2 levels "0","1": 1 2 2 2 1 2 2 1 2 1 ...
 $ gre  : int  380 660 800 640 520 760 560 400 540 700 ...
 $ gpa  : num  3.61 3.67 4 3.19 2.93 3 2.98 3.08 3.39 3.92 ...
 $ rank : Factor w/ 4 levels "1","2","3","4": 3 3 1 4 4 2 1 2 3 2 ...
```

3) VISUALIZATION

CODE:

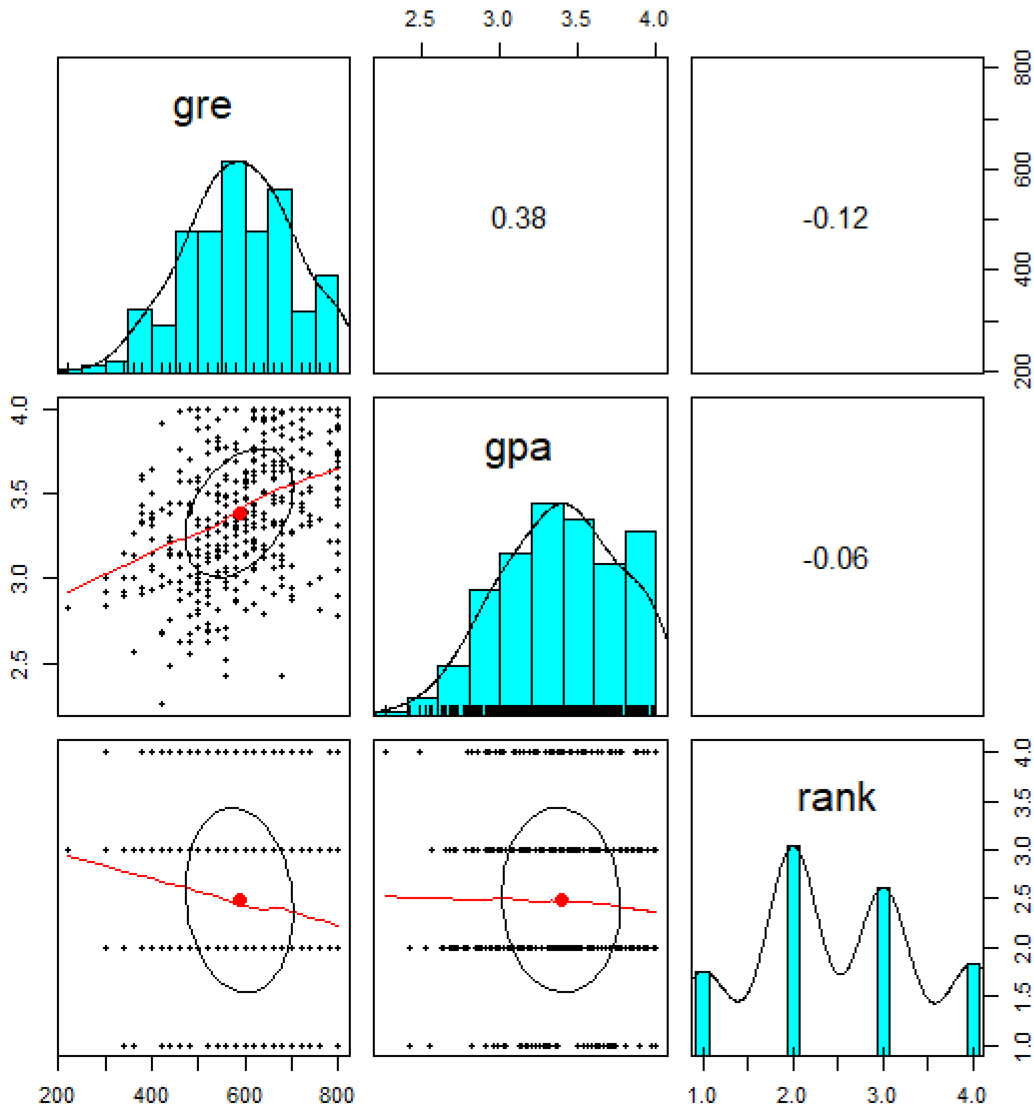
pairs.panels(data[-1])

#To check the correlation between independent variables. For developing a naive bayes classification

model, the independent variables should not be highly correlated. The first variable 'admit' is

excluded.

OUTPUT:



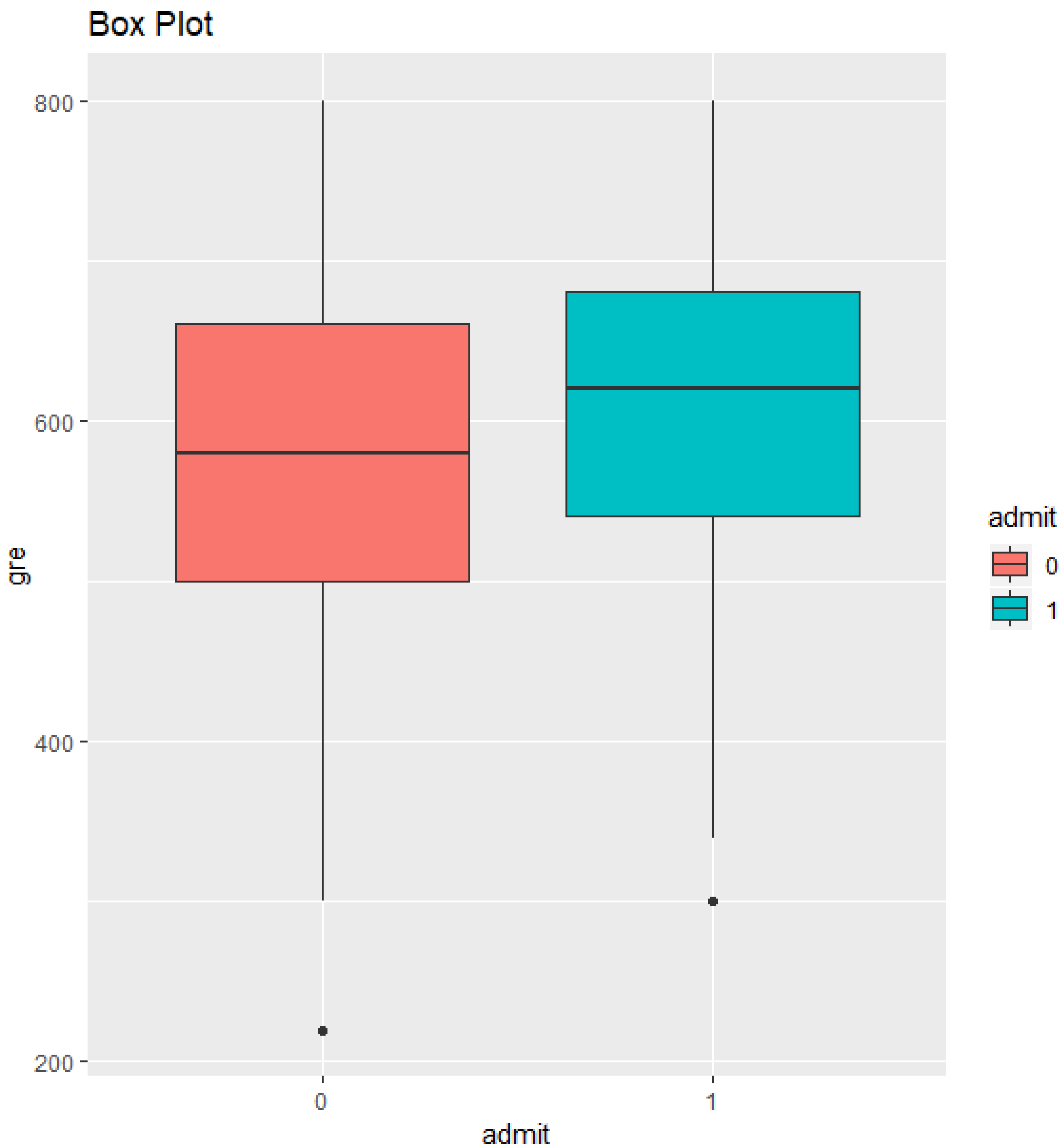
EXPLANATION FOR OUTPUT

i. The correlation coefficient is 0.38 which is low enough for naive bayes classification to be applied on the dataset.

CODE:

```
ggplot(data,aes(x=admit, y=gre, fill = admit)) + geom_boxplot() + ggtitle("Box Plot")
```

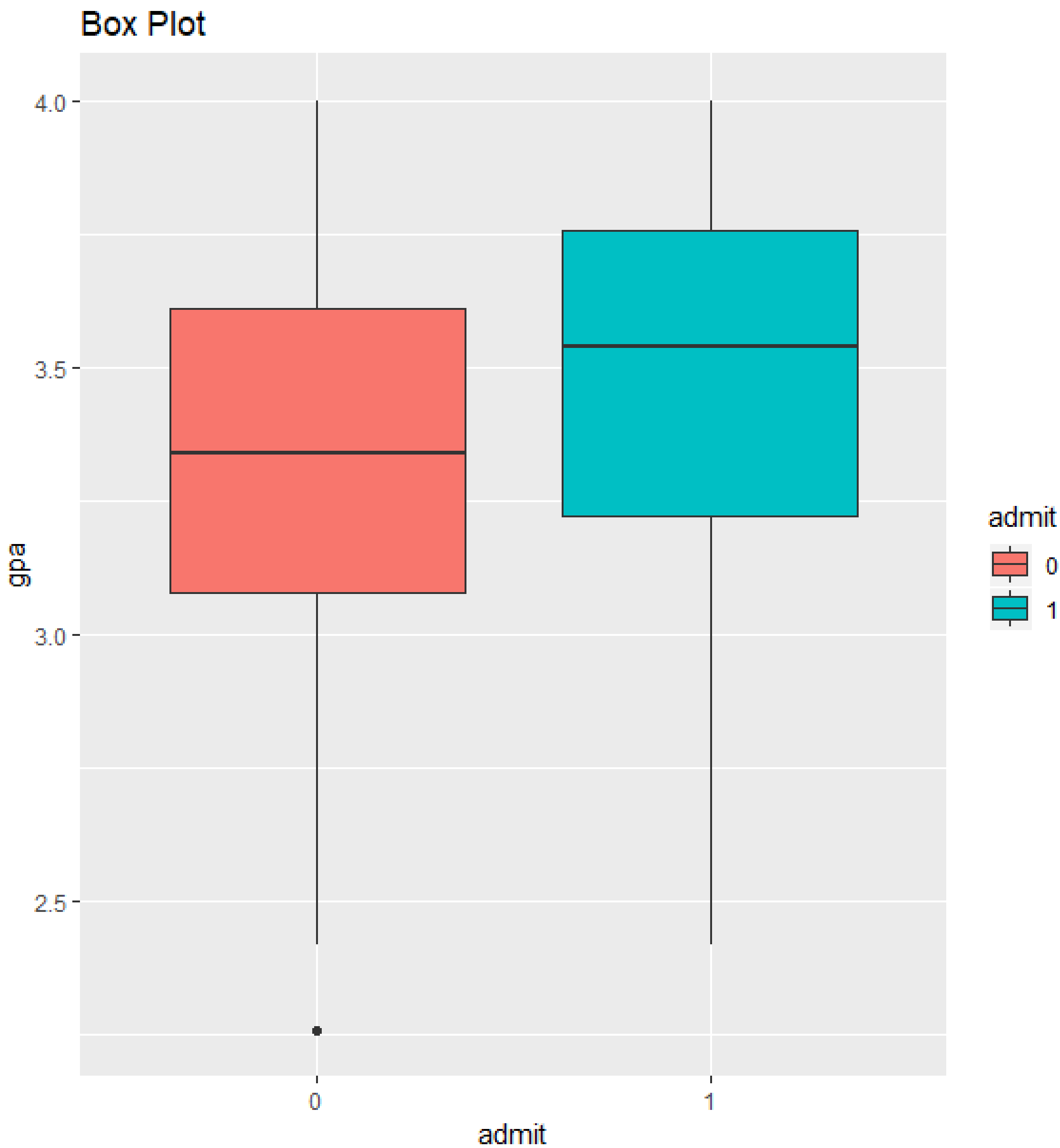
OUTPUT:



CODE:

```
ggplot(data,aes(x=admit, y=gpa, fill = admit)) + geom_boxplot() + ggtitle("Box Plot")
```

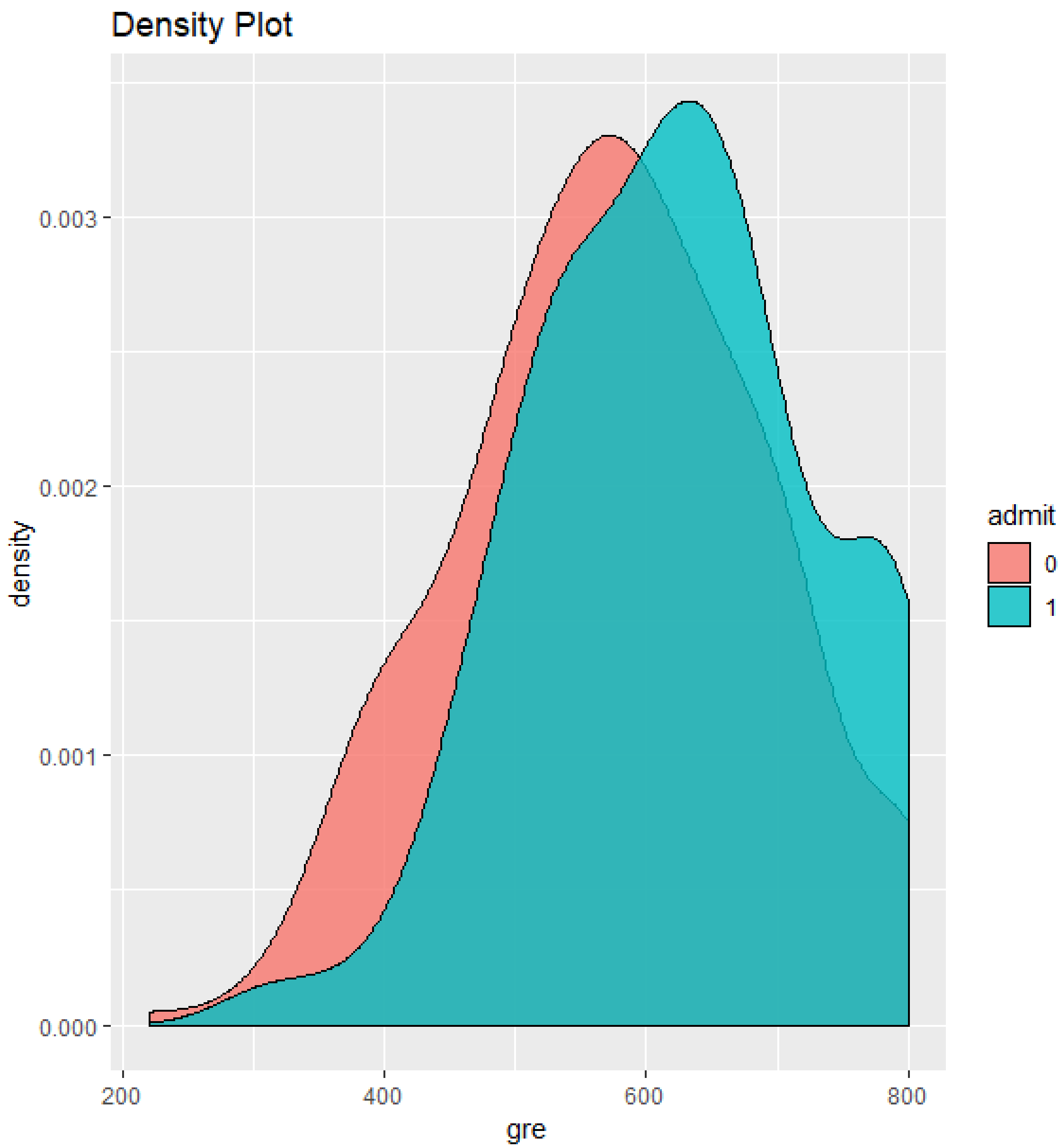
OUTPUT:



CODE:

```
ggplot(data,aes(x=gre, fill = admit)) + geom_density(alpha=0.8, color='black') + ggtitle("Density Plot")
```

OUTPUT:



4) DATA PARTITION

CODE:

set.seed(1234)

ind <- sample(2, nrow(data), replace = T, prob = c(0.8,0.2))

train <- data[ind == 1,]

test <- data[ind == 2,]

5) NAIVE BAYES MODEL

CODE:

model <- naive_bayes(admit ~ ., data = train, usekernel = T)

model

OUTPUT:

```
> model <- naive_bayes(admit ~ ., data = train, usekernel = T)
> model
===================== Naive Bayes =====================
Call:
naive_bayes.formula(formula = admit ~ ., data = train, usekernel = T)

A priori probabilities:

        0         1 
0.6861538 0.3138462 

Tables:
$`0`

Call:
	density.default(x = x, na.rm = TRUE)

Data: x (223 obs.);	Bandwidth 'bw' = 35.5

       x               y            
 Min.   :193.5   Min.   :6.010e-07  
 1st Qu.:371.7   1st Qu.:2.924e-04  
 Median :550.0   Median :1.291e-03  
 Mean   :550.0   Mean   :1.401e-03  
 3rd Qu.:728.3   3rd Qu.:2.405e-03  
```

```
 Max.   :906.5   Max.   :3.199e-03

$`1`

Call:
	density.default(x = x, na.rm = TRUE)

Data: x (102 obs.);	Bandwidth 'bw' = 39.59

       x               y
 Min.   :181.2   Min.   :1.145e-06
 1st Qu.:365.6   1st Qu.:2.007e-04
 Median :550.0   Median :1.129e-03
 Mean   :550.0   Mean   :1.354e-03
 3rd Qu.:734.4   3rd Qu.:2.375e-03
 Max.   :918.8   Max.   :3.465e-03


$`0`

Call:
	density.default(x = x, na.rm = TRUE)

Data: x (223 obs.);	Bandwidth 'bw' = 0.1134

       x               y
 Min.   :2.080   Min.   :0.0002229
 1st Qu.:2.645   1st Qu.:0.0924939
 Median :3.210   Median :0.4521795
 Mean   :3.210   Mean   :0.4419689
 3rd Qu.:3.775   3rd Qu.:0.6603271
 Max.   :4.340   Max.   :1.1433285

$`1`

Call:
	density.default(x = x, na.rm = TRUE)

Data: x (102 obs.);	Bandwidth 'bw' = 0.1234

       x              y
 Min.   :2.25   Min.   :0.0005231
 1st Qu.:2.78   1st Qu.:0.0800747
 Median :3.31   Median :0.4801891
 Mean   :3.31   Mean   :0.4710851
 3rd Qu.:3.84   3rd Qu.:0.8626207
 Max.   :4.37   Max.   :1.0595464



rank          0          1
   1 0.10313901 0.24509804
   2 0.36771300 0.42156863
   3 0.33183857 0.24509804
   4 0.19730942 0.08823529
```

EXPLANATION FOR OUTPUT

- P(Admit=1|Rank=1) = (P(Admit=1) * P(Rank=1|Admit=1))/P(Rank=1)
- P(Rank=1|Admit=0) = 0.103
- P(Rank=1|Admit=1) = 0.245

CODE:

train %>% filter(admit == "0") %>% summarise(mean(gre), sd(gre))

train %>% filter(admit == "1") %>% summarise(mean(gre), sd(gre))

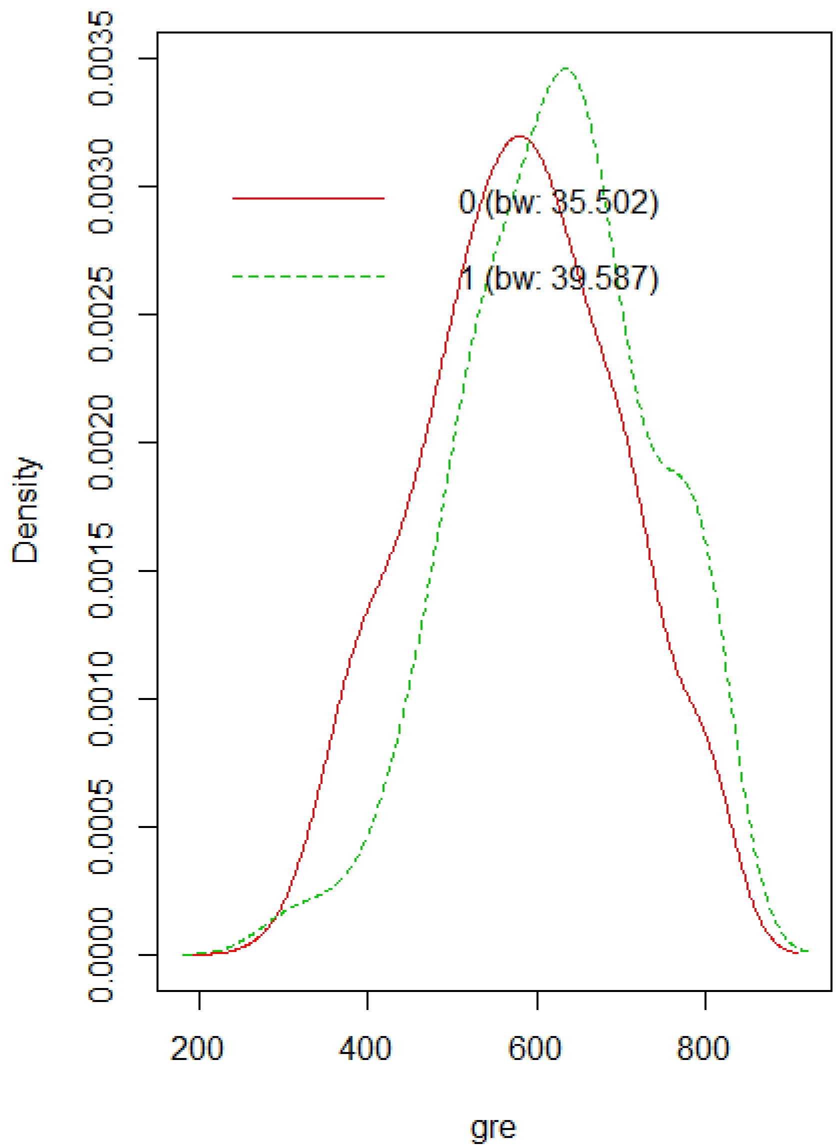
plot(model)

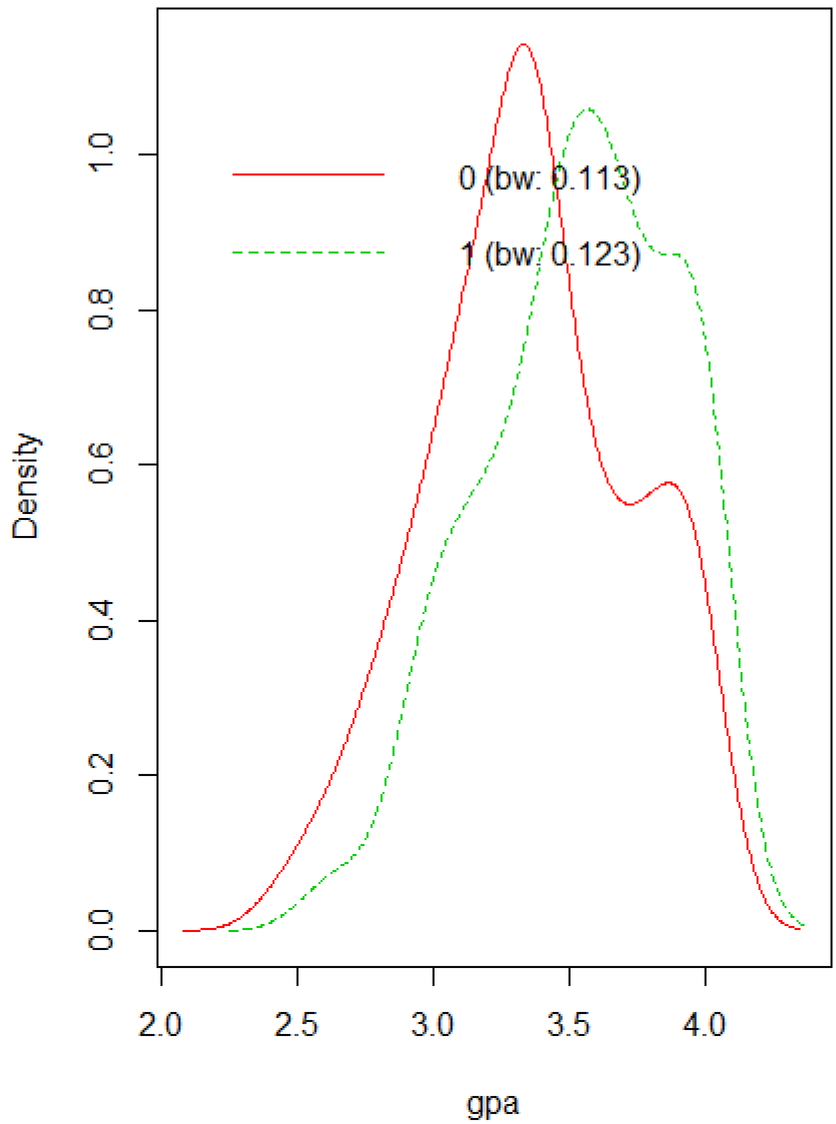
OUTPUT:

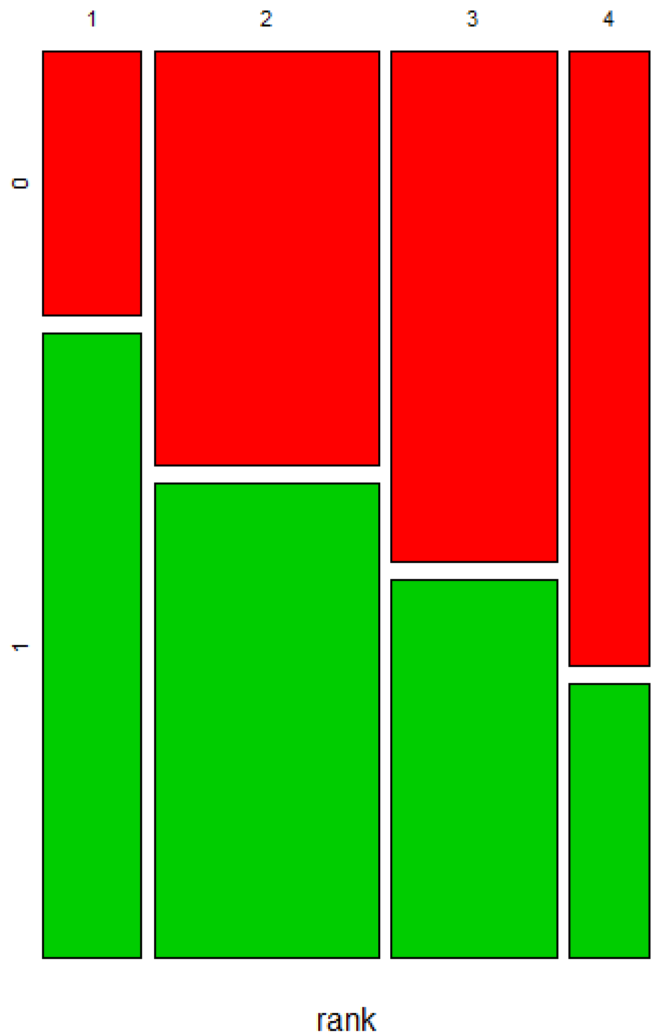
```
> train %>% filter(admit == "0") %>% summarise(mean(gre), sd(gre))
  mean(gre) sd(gre)
1  578.6547 116.325
```

```
> train %>% filter(admit == "1") %>% summarise(mean(gre), sd(gre))
  mean(gre) sd(gre)
1  622.9412 110.924
```

```
> plot(model)
```


0 (bw: 35.502)
1 (bw: 39.587)
Density
0.0000
0.0005
0.0010
0.0015
0.0020
0.0025
0.0030
0.0035
200
400
600
800
gre


0 (bw: 0.113)
1 (bw: 0.123)
Density
0.0
0.2
0.4
0.6
0.8
1.0
2.0
2.5
3.0
3.5
4.0
gpa



EXPLANATION OF OUTPUT

i. Rank is a categorical variable and it has 4 values which are 1, 2, 3 and 4.
ii. The Green colour indicates that the student is admitted to the institute.
iii. The Red colour indicates that the student is not admitted to the institute.

6) PREDICT

CODE:

p <- predict(model, train, type = 'prob')

head(cbind(p, train))

OUTPUT:

```
> p <- predict(model, train, type = 'prob')
> head(cbind(p, train))
          0         1 admit gre  gpa rank
1 0.8528794 0.1471206     0 380 3.61    3
2 0.5621460 0.4378540     1 660 3.67    3
3 0.2233490 0.7766510     1 800 4.00    1
4 0.8643901 0.1356099     1 640 3.19    4
6 0.6263274 0.3736726     1 760 3.00    2
7 0.5933791 0.4066209     1 560 2.98    1
```

7) CONFUSION MATRIX – TRAINING DATA

CODE:

p1 <- predict(model, train)

(tab1 <- table(p1, train$admit))

1 - sum(diag(tab1)) / sum(tab1)

OUTPUT:

```
> p1 <- predict(model, train)
> (tab1 <- table(p1, train$admit))

p1    0   1
  0 203  69
  1  20  33
> 1 - sum(diag(tab1)) / sum(tab1)
[1] 0.2738462
```

EXPLANATION OF OUTPUT:

i. 203 students are admitted to the institute which is correctly predicted as admitted by the model.
ii. 33 students are not admitted to the institute which is correctly predicted as not admitted by the model.
iii. 20 students are not admitted to the institute which is incorrectly predicted as admitted by the model.
iv. 69 student s are admitted to the institute which is incorrectly predicted as not admitted by the model.
v. The misclassification for the training data is 27.3%

8) CONFUSION MATRIX – TESTING DATA

CODE:

```
p2 <- predict(model, test)

(tab2 <- table(p2, test$admit))

1 - sum(diag(tab2)) / sum(tab2)
```

OUTPUT:

```
> p2 <- predict(model, test)
> (tab2 <- table(p2, test$admit))

p2    0  1
  0 47 20
  1  3  5
> 1 - sum(diag(tab2)) / sum(tab2)
[1] 0.3066667
```

EXPLANATION OF OUTPUT:

i. 47 students are not admitted to the institute which is correctly predicted as not admitted by the model.
ii. 5 students are admitted to the institute which is correctly predicted as admitted by the model.
iii. 3 students are not admitted to the institute which is incorrectly predicted as admitted by the model.

iv. 20 students are admitted to the institute which is incorrectly predicted as not admitted by the model.
v. The misclassification for the testing data is 30.67%

CONCLUSION

- The misclassification for the training data is 27.3% and the misclassification for the testing data is 30.67%.
- Lower percentages of misclassification give better accuracy of predictions done by the model.