**Finolex Academy of Management and Technology, Ratnagiri**

**Department of MCA**

**Course:- MCAL13 Advance Database Management System Lab**

**Practical No -07**

1. **Write a program to perform k means clustering on iris dataset. Perform data pre-processing if required.**

# load packages-tidyverse,datasets,ggplot2

install.packages(tidyverse)

library(tidyverse)

install.packages(datasets)

library(datasets)

install.packages(ggplot2)

library(ggplot2)

#load dataset iris

>iris

Sepal.Length Sepal.Width Petal.Length Petal.Width Species

1 5.1 3.5 1.4 0.2 setosa

2 4.9 3.0 1.4 0.2 setosa

3 4.7 3.2 1.3 0.2 setosa

4 4.6 3.1 1.5 0.2 setosa

5 5.0 3.6 1.4 0.2 setosa

6 5.4 3.9 1.7 0.4 setosa

7 4.6 3.4 1.4 0.3 setosa

8 5.0 3.4 1.5 0.2 setosa

9 4.4 2.9 1.4 0.2 setosa

10 4.9 3.1 1.5 0.1 setosa

11 5.4 3.7 1.5 0.2 setosa

12 4.8 3.4 1.6 0.2 setosa

13 4.8 3.0 1.4 0.1 setosa

14 4.3 3.0 1.1 0.1 setosa

15 5.8 4.0 1.2 0.2 setosa

16 5.7 4.4 1.5 0.4 setosa

17 5.4 3.9 1.3 0.4 setosa

18 5.1 3.5 1.4 0.3 setosa

19 5.7 3.8 1.7 0.3 setosa

20 5.1 3.8 1.5 0.3 setosa

21 5.4 3.4 1.7 0.2 setosa

22 5.1 3.7 1.5 0.4 setosa

23 4.6 3.6 1.0 0.2 setosa

24 5.1 3.3 1.7 0.5 setosa

25 4.8 3.4 1.9 0.2 setosa

26 5.0 3.0 1.6 0.2 setosa

27 5.0 3.4 1.6 0.4 setosa

28 5.2 3.5 1.5 0.2 setosa

29 5.2 3.4 1.4 0.2 setosa

30 4.7 3.2 1.6 0.2 setosa

31 4.8 3.1 1.6 0.2 setosa

32 5.4 3.4 1.5 0.4 setosa

33 5.2 4.1 1.5 0.1 setosa

34 5.5 4.2 1.4 0.2 setosa

35 4.9 3.1 1.5 0.2 setosa

36 5.0 3.2 1.2 0.2 setosa

37 5.5 3.5 1.3 0.2 setosa

38 4.9 3.6 1.4 0.1 setosa

39 4.4 3.0 1.3 0.2 setosa

40 5.1 3.4 1.5 0.2 setosa

41 5.0 3.5 1.3 0.3 setosa

42 4.5 2.3 1.3 0.3 setosa

43 4.4 3.2 1.3 0.2 setosa

44 5.0 3.5 1.6 0.6 setosa

45 5.1 3.8 1.9 0.4 setosa

46 4.8 3.0 1.4 0.3 setosa

47 5.1 3.8 1.6 0.2 setosa

48 4.6 3.2 1.4 0.2 setosa

49 5.3 3.7 1.5 0.2 setosa

50 5.0 3.3 1.4 0.2 setosa

51 7.0 3.2 4.7 1.4 versicolor

52 6.4 3.2 4.5 1.5 versicolor

53 6.9 3.1 4.9 1.5 versicolor

54 5.5 2.3 4.0 1.3 versicolor

55 6.5 2.8 4.6 1.5 versicolor

56 5.7 2.8 4.5 1.3 versicolor

57 6.3 3.3 4.7 1.6 versicolor

58 4.9 2.4 3.3 1.0 versicolor

59 6.6 2.9 4.6 1.3 versicolor

60 5.2 2.7 3.9 1.4 versicolor

61 5.0 2.0 3.5 1.0 versicolor

62 5.9 3.0 4.2 1.5 versicolor

63 6.0 2.2 4.0 1.0 versicolor

64 6.1 2.9 4.7 1.4 versicolor

65 5.6 2.9 3.6 1.3 versicolor

66 6.7 3.1 4.4 1.4 versicolor

67 5.6 3.0 4.5 1.5 versicolor

68 5.8 2.7 4.1 1.0 versicolor

69 6.2 2.2 4.5 1.5 versicolor

70 5.6 2.5 3.9 1.1 versicolor

71 5.9 3.2 4.8 1.8 versicolor

72 6.1 2.8 4.0 1.3 versicolor

73 6.3 2.5 4.9 1.5 versicolor

74 6.1 2.8 4.7 1.2 versicolor

75 6.4 2.9 4.3 1.3 versicolor

76 6.6 3.0 4.4 1.4 versicolor

77 6.8 2.8 4.8 1.4 versicolor

78 6.7 3.0 5.0 1.7 versicolor

79 6.0 2.9 4.5 1.5 versicolor

80 5.7 2.6 3.5 1.0 versicolor

81 5.5 2.4 3.8 1.1 versicolor

82 5.5 2.4 3.7 1.0 versicolor

83 5.8 2.7 3.9 1.2 versicolor

84 6.0 2.7 5.1 1.6 versicolor

85 5.4 3.0 4.5 1.5 versicolor

86 6.0 3.4 4.5 1.6 versicolor

87 6.7 3.1 4.7 1.5 versicolor

88 6.3 2.3 4.4 1.3 versicolor

89 5.6 3.0 4.1 1.3 versicolor

90 5.5 2.5 4.0 1.3 versicolor

91 5.5 2.6 4.4 1.2 versicolor

92 6.1 3.0 4.6 1.4 versicolor

93 5.8 2.6 4.0 1.2 versicolor

94 5.0 2.3 3.3 1.0 versicolor

95 5.6 2.7 4.2 1.3 versicolor

96 5.7 3.0 4.2 1.2 versicolor

97 5.7 2.9 4.2 1.3 versicolor

98 6.2 2.9 4.3 1.3 versicolor

99 5.1 2.5 3.0 1.1 versicolor

100 5.7 2.8 4.1 1.3 versicolor

101 6.3 3.3 6.0 2.5 virginica

102 5.8 2.7 5.1 1.9 virginica

103 7.1 3.0 5.9 2.1 virginica

104 6.3 2.9 5.6 1.8 virginica

105 6.5 3.0 5.8 2.2 virginica

106 7.6 3.0 6.6 2.1 virginica

107 4.9 2.5 4.5 1.7 virginica

108 7.3 2.9 6.3 1.8 virginica

109 6.7 2.5 5.8 1.8 virginica

110 7.2 3.6 6.1 2.5 virginica

111 6.5 3.2 5.1 2.0 virginica

112 6.4 2.7 5.3 1.9 virginica

113 6.8 3.0 5.5 2.1 virginica

114 5.7 2.5 5.0 2.0 virginica

115 5.8 2.8 5.1 2.4 virginica

116 6.4 3.2 5.3 2.3 virginica

117 6.5 3.0 5.5 1.8 virginica

118 7.7 3.8 6.7 2.2 virginica

119 7.7 2.6 6.9 2.3 virginica

120 6.0 2.2 5.0 1.5 virginica

121 6.9 3.2 5.7 2.3 virginica

122 5.6 2.8 4.9 2.0 virginica

123 7.7 2.8 6.7 2.0 virginica

124 6.3 2.7 4.9 1.8 virginica

125 6.7 3.3 5.7 2.1 virginica

126 7.2 3.2 6.0 1.8 virginica

127 6.2 2.8 4.8 1.8 virginica

128 6.1 3.0 4.9 1.8 virginica

129 6.4 2.8 5.6 2.1 virginica

130 7.2 3.0 5.8 1.6 virginica

131 7.4 2.8 6.1 1.9 virginica

132 7.9 3.8 6.4 2.0 virginica

133 6.4 2.8 5.6 2.2 virginica

134 6.3 2.8 5.1 1.5 virginica

135 6.1 2.6 5.6 1.4 virginica

136 7.7 3.0 6.1 2.3 virginica

137 6.3 3.4 5.6 2.4 virginica

138 6.4 3.1 5.5 1.8 virginica

139 6.0 3.0 4.8 1.8 virginica

140 6.9 3.1 5.4 2.1 virginica

141 6.7 3.1 5.6 2.4 virginica

142 6.9 3.1 5.1 2.3 virginica

143 5.8 2.7 5.1 1.9 virginica

144 6.8 3.2 5.9 2.3 virginica

145 6.7 3.3 5.7 2.5 virginica

146 6.7 3.0 5.2 2.3 virginica

147 6.3 2.5 5.0 1.9 virginica

148 6.5 3.0 5.2 2.0 virginica

149 6.2 3.4 5.4 2.3 virginica

150 5.9 3.0 5.1 1.8 virginica

#information about iris dataset

>head(iris,4)

Sepal.Length Sepal.Width

1 5.1 3.5

2 4.9 3.0

3 4.7 3.2

4 4.6 3.1

Petal.Length Petal.Width Species

1 1.4 0.2 setosa

2 1.4 0.2 setosa

3 1.3 0.2 setosa

4 1.5 0.2 setosa

**>tail(iris)**

Sepal.Length Sepal.Width

145 6.7 3.3

146 6.7 3.0

147 6.3 2.5

148 6.5 3.0

149 6.2 3.4

150 5.9 3.0

Petal.Length Petal.Width

145 5.7 2.5

146 5.2 2.3

147 5.0 1.9

148 5.2 2.0

149 5.4 2.3

150 5.1 1.8

Species

145 virginica

146 virginica

147 virginica

148 virginica

149 virginica

150 virginica

>dim(iris)

[1] 150 5

>names(iris)

[1] "Sepal.Length" "Sepal.Width"

[3] "Petal.Length" "Petal.Width"

[5] "Species"

>attributes(iris)

$names

[1] "Sepal.Length" "Sepal.Width"

[3] "Petal.Length" "Petal.Width"

[5] "Species"

$class

[1] "data.frame"

$row.names

[1] 1 2 3 4 5 6 7

[8] 8 9 10 11 12 13 14

[15] 15 16 17 18 19 20 21

[22] 22 23 24 25 26 27 28

[29] 29 30 31 32 33 34 35

[36] 36 37 38 39 40 41 42

[43] 43 44 45 46 47 48 49

[50] 50 51 52 53 54 55 56

[57] 57 58 59 60 61 62 63

[64] 64 65 66 67 68 69 70

[71] 71 72 73 74 75 76 77

[78] 78 79 80 81 82 83 84

[85] 85 86 87 88 89 90 91

[92] 92 93 94 95 96 97 98

[99] 99 100 101 102 103 104 105

[106] 106 107 108 109 110 111 112

[113] 113 114 115 116 117 118 119

[120] 120 121 122 123 124 125 126

[127] 127 128 129 130 131 132 133

[134] 134 135 136 137 138 139 140

[141] 141 142 143 144 145 146 147

[148] 148 149 150

Summary(iris)

Sepal.Length Sepal.Width

Min. :4.300 Min. :2.000

1st Qu.:5.100 1st Qu.:2.800

Median :5.800 Median :3.000

Mean :5.843 Mean :3.057

3rd Qu.:6.400 3rd Qu.:3.300

Max. :7.900 Max. :4.400

Petal.Length Petal.Width

Min. :1.000 Min. :0.100

1st Qu.:1.600 1st Qu.:0.300

Median :4.350 Median :1.300

Mean :3.758 Mean :1.199

3rd Qu.:5.100 3rd Qu.:1.800

Max. :6.900 Max. :2.500

Species

setosa :50

versicolor:50

virginica :50

iris[1:5,]

Sepal.Length Sepal.Width

1 5.1 3.5

2 4.9 3.0

3 4.7 3.2

4 4.6 3.1

5 5.0 3.6

Petal.Length Petal.Width Species

1 1.4 0.2 setosa

2 1.4 0.2 setosa

3 1.3 0.2 setosa

4 1.5 0.2 setosa

5 1.4 0.2 setosa

> iris[,1:1]

[1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6

[8] 5.0 4.4 4.9 5.4 4.8 4.8 4.3

[15] 5.8 5.7 5.4 5.1 5.7 5.1 5.4

[22] 5.1 4.6 5.1 4.8 5.0 5.0 5.2

[29] 5.2 4.7 4.8 5.4 5.2 5.5 4.9

[36] 5.0 5.5 4.9 4.4 5.1 5.0 4.5

[43] 4.4 5.0 5.1 4.8 5.1 4.6 5.3

[50] 5.0 7.0 6.4 6.9 5.5 6.5 5.7

[57] 6.3 4.9 6.6 5.2 5.0 5.9 6.0

[64] 6.1 5.6 6.7 5.6 5.8 6.2 5.6

[71] 5.9 6.1 6.3 6.1 6.4 6.6 6.8

[78] 6.7 6.0 5.7 5.5 5.5 5.8 6.0

[85] 5.4 6.0 6.7 6.3 5.6 5.5 5.5

[92] 6.1 5.8 5.0 5.6 5.7 5.7 6.2

[99] 5.1 5.7 6.3 5.8 7.1 6.3 6.5

[106] 7.6 4.9 7.3 6.7 7.2 6.5 6.4

[113] 6.8 5.7 5.8 6.4 6.5 7.7 7.7

[120] 6.0 6.9 5.6 7.7 6.3 6.7 7.2

[127] 6.2 6.1 6.4 7.2 7.4 7.9 6.4

[134] 6.3 6.1 7.7 6.3 6.4 6.0 6.9

[141] 6.7 6.9 5.8 6.8 6.7 6.7 6.3

[148] 6.5 6.2 5.9

> iris[1:10,"Sepal.Length"]

[1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6

[8] 5.0 4.4 4.9

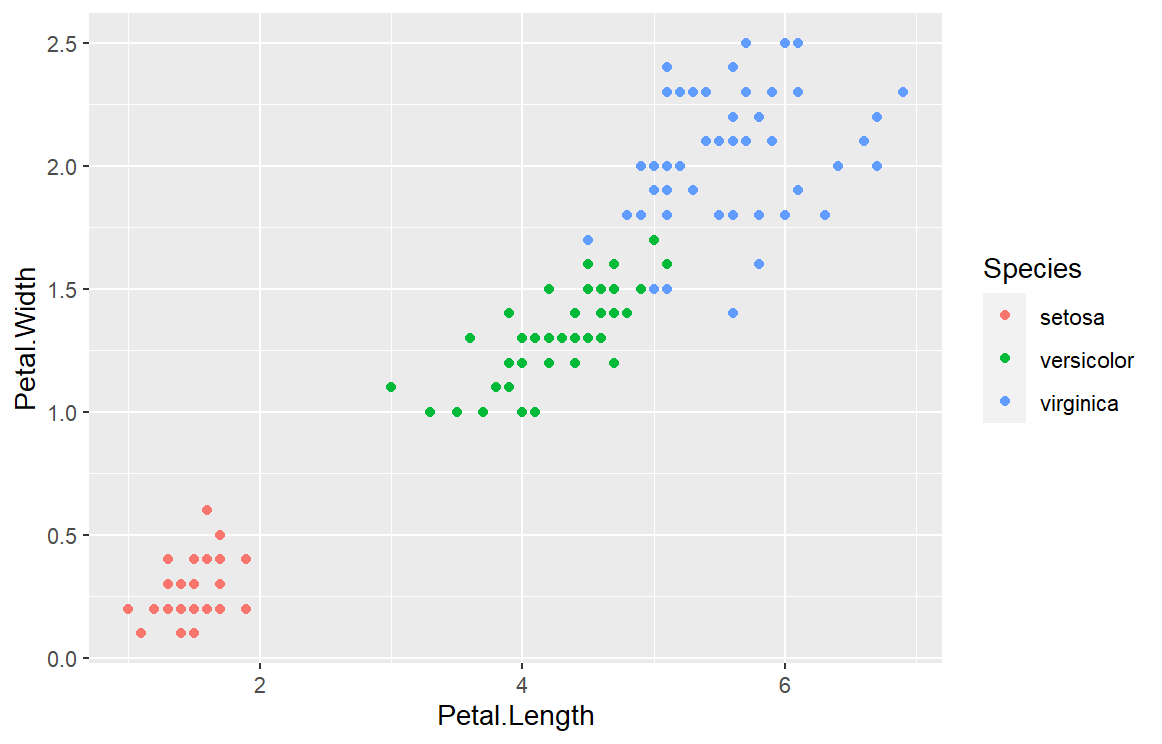
> sum(is.na(iris))

[1] 0

#plot data using ggplot() function of ggplot2 library

> library(ggplot2)

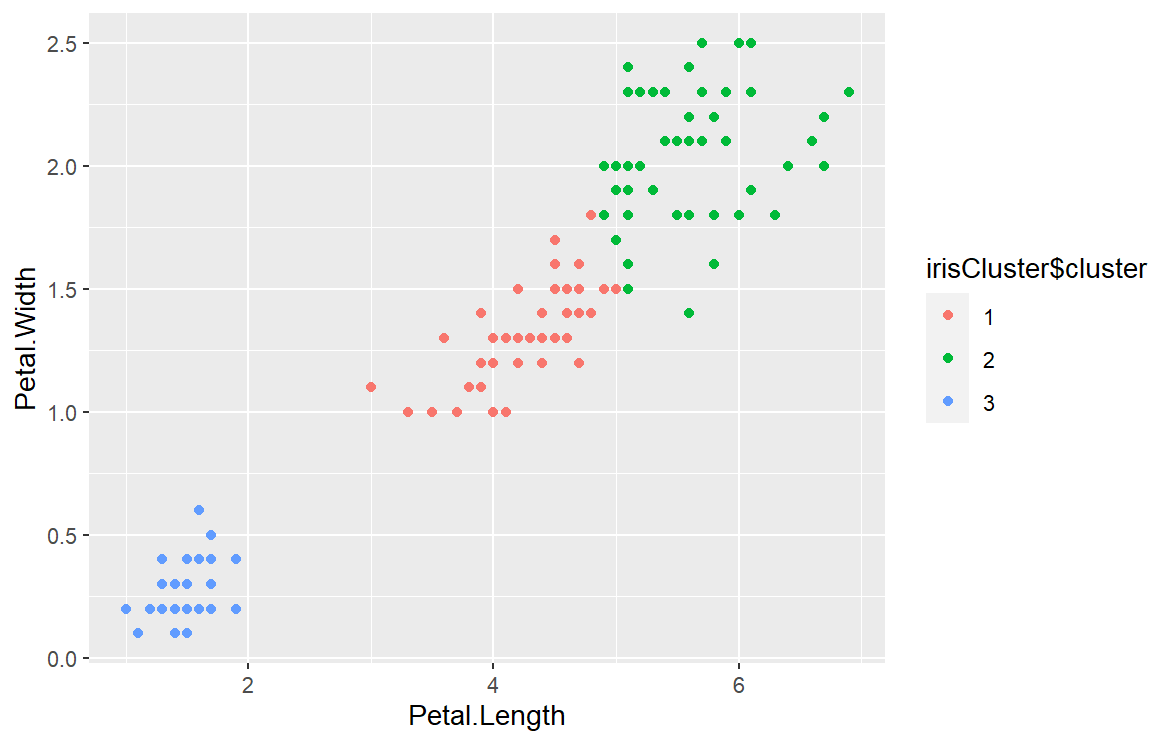
> ggplot(iris, aes(Petal.Length, Petal.Width, color = Species)) + geom\_point()



#clustering

>Set.seed(20)

|  |
| --- |
| > irisCluster <- kmeans(iris[, 3:4], 3, nstart = 20)  > irisCluster  K-means clustering with 3 clusters of sizes 52, 48, 50  Cluster means:  Petal.Length Petal.Width  1 4.269231 1.342308  2 5.595833 2.037500  3 1.462000 0.246000  Clustering vector:  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  [20] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  [39] 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1  [58] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  [77] 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1  [96] 1 1 1 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2  [115] 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2  [134] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2  Within cluster sum of squares by cluster:  [1] 13.05769 16.29167 2.02200  (between\_SS / total\_SS = 94.3 %)  Available components:  [1] "cluster" "centers"  [3] "totss" "withinss"  [5] "tot.withinss" "betweenss"  [7] "size" "iter"  [9] "ifault" |
|  |
| > table(irisCluster$cluster, iris$Species)    setosa versicolor virginica  1 0 48 4  2 0 2 46  3 50 0 0  #plot data to see the clusters  > irisCluster$cluster <- as.factor(irisCluster$cluster)  > irisCluster$cluster  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  [20] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3  [39] 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1  [58] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  [77] 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1  [96] 1 1 1 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2  [115] 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2  [134] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2  Levels: 1 2 3   |  | | --- | | >ggplot(iris, aes(Petal.Length, Petal.Width, color = irisCluster$cluster)) + geom\_point() | |



**2. Implement Regression Classification for following example using R**

**years=(3,8,9,13,3,6,11,21,1,16)**

**salary=(30,57,64,72,36,43,59,90,20,83)**

**Predict salary of a person having 10 years of experience in a company.**

**🡪**

#load packages

library(ggplot2)

library(tidyverse)

#create csv file of years=(3,8,9,13,3,6,11,21,1,16)

salary=(30,57,64,72,36,43,59,90,20,83) data

#import data from csv file

> rldata <- read.csv("linear01.csv")

> rldata

Years Salary

1 3 30

2 8 57

3 9 64

4 13 72

5 3 36

6 6 43

7 11 59

8 21 90

9 1 20

10 16 83

> relation <- lm(years~salary,data=rldata)

> relation

Call:

lm(formula = years ~ salary, data = rldata)

Coefficients:

(Intercept) salary

-5.7001 0.2671

> summary(relation)

Call:

lm(formula = years ~ salary, data = rldata)

Residuals:

Min 1Q Median

-2.3975 -0.8216 -0.1303

3Q Max

0.8751 2.6566

Coefficients:

Estimate

(Intercept) -5.70007

salary 0.26715

Std. Error

(Intercept) 1.35614

salary 0.02278

t value Pr(>|t|)

(Intercept) -4.203 0.00298

salary 11.728 2.55e-06

(Intercept) \*\*

salary \*\*\*

---

Signif. codes:

0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01

‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.57 on 8 degrees of freedom

Multiple R-squared: 0.945, Adjusted R-squared: 0.9382

F-statistic: 137.5 on 1 and 8 DF, p-value: 2.553e-06

> #predict salary of 10person having 10yrs experience

> a<-data.frame(years=10)

> result <-predict(relation,a)

> result

1 2

2.3144096 9.5274388

3 4

11.3974834 13.5346772

5 6

3.9173049 5.7873495

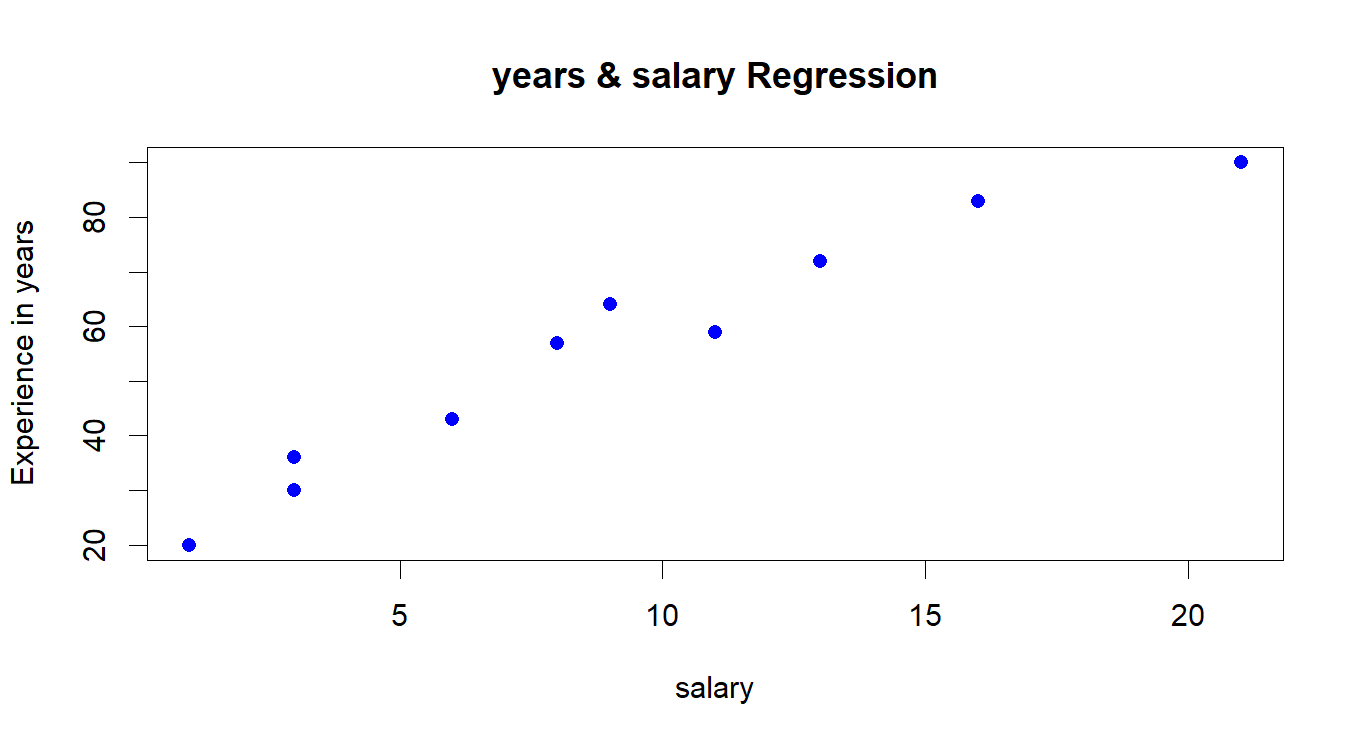
7 8

10.0617372 18.3433634

9 10

-0.3570827 16.4733187

> plot(rldata,col = "blue",pch = 16,main = "years & salary Regression",ylab = "Experience in years",xlab = "salary")

****

**3.Write a program to perform market basket analysis on Groceries dataset and display the top 5 important rules after sorting by confidence.­­­**

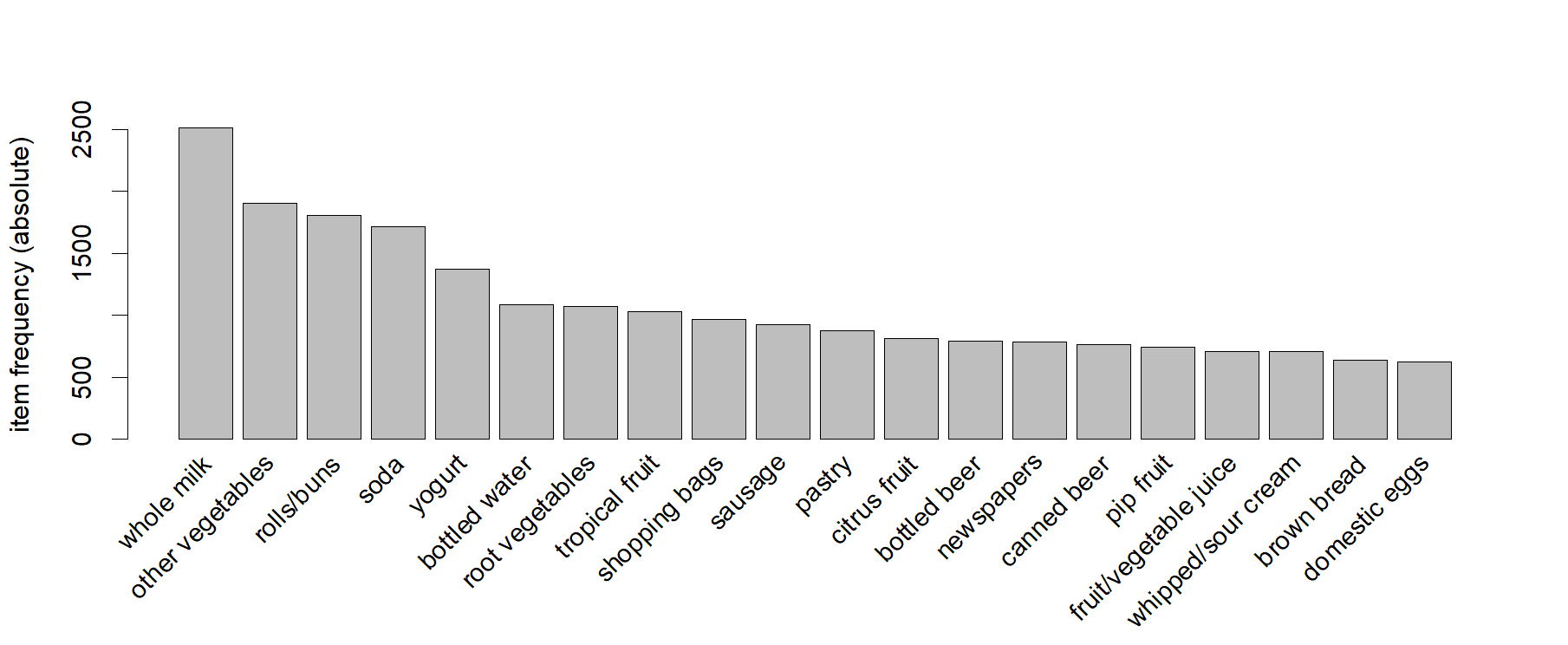
> library(arules)

> library(arulesViz)

> data("Groceries")

>#explore the data before making any rules

> itemFrequencyPlot(Groceries,topN=20,type="absolute")



**> performing apriori algorithm and generating association rules**

> rules <- apriori(Groceries, parameter = list(supp = 0.001, conf = 0.8))

Apriori

Parameter specification:

confidence minval smax arem

0.8 0.1 1 none

aval originalSupport maxtime

FALSE TRUE 5

support minlen maxlen target

0.001 1 10 rules

ext

TRUE

Algorithmic control:

filter tree heap memopt load

0.1 TRUE TRUE FALSE TRUE

sort verbose

2 TRUE

Absolute minimum support count: 9

set item appearances ...[0 item(s)] done [0.00s].

set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].

sorting and recoding items ... [157 item(s)] done [0.00s].

creating transaction tree ... done [0.01s].

checking subsets of size 1 2 3 4 5 6 done [0.03s].

writing ... [410 rule(s)] done [0.00s].

creating S4 object ... done [0.04s].

|  |
| --- |
| > options(digits=2)  > inspect(rules[1:5])  lhs rhs support confidence coverage lift count  [1] {liquor,  red/blush wine} => {bottled beer} 0.0019 0.90 0.0021 11.2 19  [2] {curd,  cereals} => {whole milk} 0.0010 0.91 0.0011 3.6 10  [3] {yogurt,  cereals} => {whole milk} 0.0017 0.81 0.0021 3.2 17  [4] {butter,  jam} => {whole milk} 0.0010 0.83 0.0012 3.3 10  [5] {soups,  bottled beer} => {whole milk} 0.0011 0.92 0.0012 3.6 11 |
|  |
| |  | | --- | | > | |

> rules<-sort(rules, by="confidence", decreasing=TRUE)

> rules <- apriori(Groceries, parameter = list(supp = 0.001, conf = 0.8,maxlen=3))

Apriori

Parameter specification:

confidence minval smax arem aval

0.8 0.1 1 none FALSE

originalSupport maxtime support

TRUE 5 0.001

minlen maxlen target ext

1 3 rules TRUE

Algorithmic control:

filter tree heap memopt load sort

0.1 TRUE TRUE FALSE TRUE 2

verbose

TRUE

Absolute minimum support count: 9

set item appearances ...[0 item(s)] done [0.00s].

set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].

sorting and recoding items ... [157 item(s)] done [0.00s].

creating transaction tree ... done [0.01s].

checking subsets of size 1 2 3 done [0.01s].

writing ... [29 rule(s)] done [0.00s].

creating S4 object ... done [0.00s].

>rules<-apriori(data=Groceries, parameter=list(supp=0.001,conf = 0.15,minlen=2),appearance = list(default="rhs",lhs="whole milk"),control = list(verbose=F))rules<-sort(rules, decreasing=TRUE,by="confidence")

inspect(rules[1:5])

lhs rhs support confidence coverage lift

[1] {whole milk} => {other vegetables} 0.075 0.29 0.26 1.5

[2] {whole milk} => {rolls/buns} 0.057 0.22 0.26 1.2

[3] {whole milk} => {yogurt} 0.056 0.22 0.26 1.6

[4] {whole milk} => {root vegetables} 0.049 0.19 0.26 1.8

[5] {whole milk} => {tropical fruit} 0.042 0.17 0.26 1.6

count

[1] 736

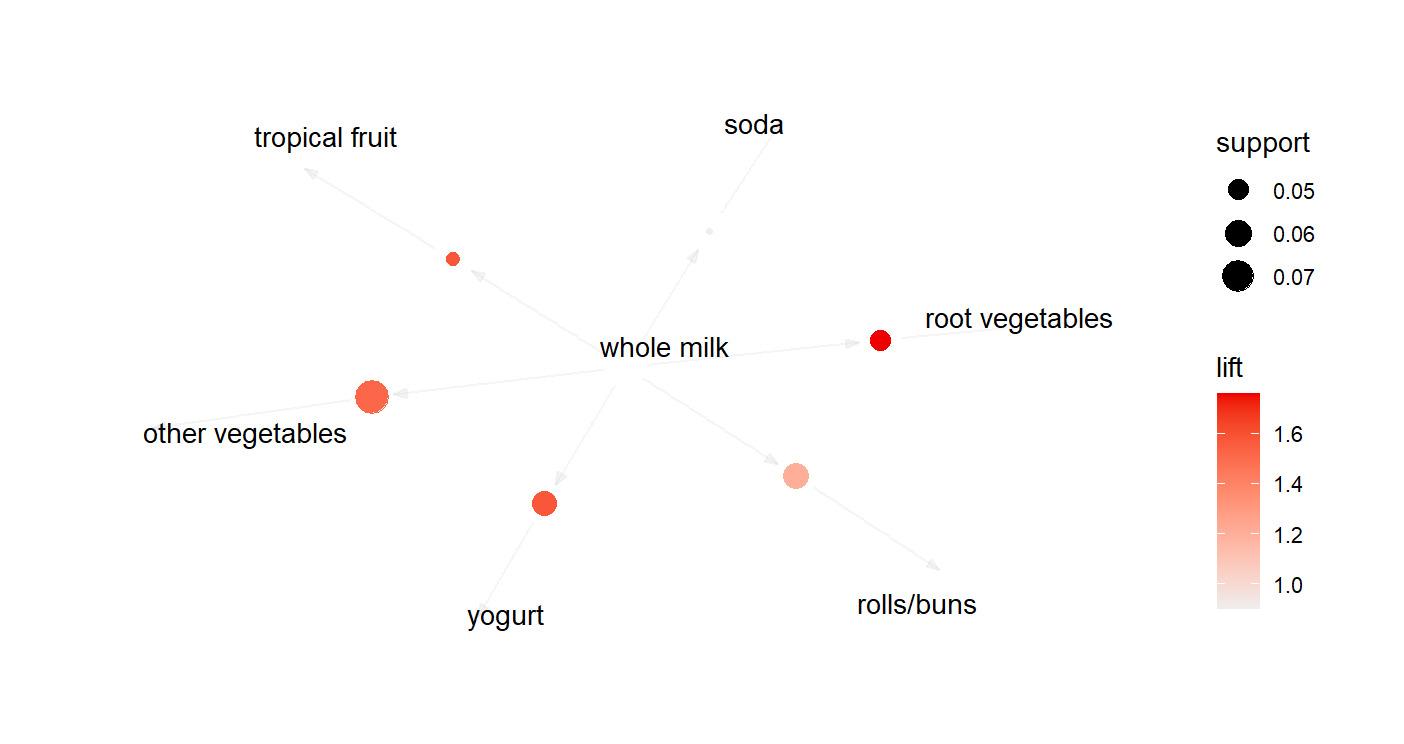
[2] 557

[3] 551

[4] 481

[5] 416

> plot(rules, method="graph")

****

**4. Write a Program to perform naïve bayes classification on iris dataset. Perform data pre-processing if required.**

Install.packages(caTools)

Library(caTools)

Install.packages(“e1071”)

Library(“e1071”)

Install.packages(“caret”)

#load dataset iris

>iris

Sepal.Length Sepal.Width Petal.Length Petal.Width

1 5.1 3.5 1.4 0.2

2 4.9 3.0 1.4 0.2

3 4.7 3.2 1.3 0.2

4 4.6 3.1 1.5 0.2

5 5.0 3.6 1.4 0.2

6 5.4 3.9 1.7 0.4

7 4.6 3.4 1.4 0.3

8 5.0 3.4 1.5 0.2

9 4.4 2.9 1.4 0.2

10 4.9 3.1 1.5 0.1

11 5.4 3.7 1.5 0.2

12 4.8 3.4 1.6 0.2

13 4.8 3.0 1.4 0.1

14 4.3 3.0 1.1 0.1

15 5.8 4.0 1.2 0.2

16 5.7 4.4 1.5 0.4

17 5.4 3.9 1.3 0.4

18 5.1 3.5 1.4 0.3

19 5.7 3.8 1.7 0.3

20 5.1 3.8 1.5 0.3

21 5.4 3.4 1.7 0.2

22 5.1 3.7 1.5 0.4

23 4.6 3.6 1.0 0.2

24 5.1 3.3 1.7 0.5

25 4.8 3.4 1.9 0.2

26 5.0 3.0 1.6 0.2

27 5.0 3.4 1.6 0.4

28 5.2 3.5 1.5 0.2

29 5.2 3.4 1.4 0.2

30 4.7 3.2 1.6 0.2

31 4.8 3.1 1.6 0.2

32 5.4 3.4 1.5 0.4

33 5.2 4.1 1.5 0.1

34 5.5 4.2 1.4 0.2

35 4.9 3.1 1.5 0.2

36 5.0 3.2 1.2 0.2

37 5.5 3.5 1.3 0.2

38 4.9 3.6 1.4 0.1

39 4.4 3.0 1.3 0.2

40 5.1 3.4 1.5 0.2

41 5.0 3.5 1.3 0.3

42 4.5 2.3 1.3 0.3

43 4.4 3.2 1.3 0.2

44 5.0 3.5 1.6 0.6

45 5.1 3.8 1.9 0.4

46 4.8 3.0 1.4 0.3

47 5.1 3.8 1.6 0.2

48 4.6 3.2 1.4 0.2

49 5.3 3.7 1.5 0.2

50 5.0 3.3 1.4 0.2

51 7.0 3.2 4.7 1.4

52 6.4 3.2 4.5 1.5

53 6.9 3.1 4.9 1.5

54 5.5 2.3 4.0 1.3

55 6.5 2.8 4.6 1.5

56 5.7 2.8 4.5 1.3

57 6.3 3.3 4.7 1.6

58 4.9 2.4 3.3 1.0

59 6.6 2.9 4.6 1.3

60 5.2 2.7 3.9 1.4

61 5.0 2.0 3.5 1.0

62 5.9 3.0 4.2 1.5

63 6.0 2.2 4.0 1.0

64 6.1 2.9 4.7 1.4

65 5.6 2.9 3.6 1.3

66 6.7 3.1 4.4 1.4

67 5.6 3.0 4.5 1.5

68 5.8 2.7 4.1 1.0

69 6.2 2.2 4.5 1.5

70 5.6 2.5 3.9 1.1

71 5.9 3.2 4.8 1.8

72 6.1 2.8 4.0 1.3

73 6.3 2.5 4.9 1.5

74 6.1 2.8 4.7 1.2

75 6.4 2.9 4.3 1.3

76 6.6 3.0 4.4 1.4

77 6.8 2.8 4.8 1.4

78 6.7 3.0 5.0 1.7

79 6.0 2.9 4.5 1.5

80 5.7 2.6 3.5 1.0

81 5.5 2.4 3.8 1.1

82 5.5 2.4 3.7 1.0

83 5.8 2.7 3.9 1.2

84 6.0 2.7 5.1 1.6

85 5.4 3.0 4.5 1.5

86 6.0 3.4 4.5 1.6

87 6.7 3.1 4.7 1.5

88 6.3 2.3 4.4 1.3

89 5.6 3.0 4.1 1.3

90 5.5 2.5 4.0 1.3

91 5.5 2.6 4.4 1.2

92 6.1 3.0 4.6 1.4

93 5.8 2.6 4.0 1.2

94 5.0 2.3 3.3 1.0

95 5.6 2.7 4.2 1.3

96 5.7 3.0 4.2 1.2

97 5.7 2.9 4.2 1.3

98 6.2 2.9 4.3 1.3

99 5.1 2.5 3.0 1.1

100 5.7 2.8 4.1 1.3

101 6.3 3.3 6.0 2.5

102 5.8 2.7 5.1 1.9

103 7.1 3.0 5.9 2.1

104 6.3 2.9 5.6 1.8

105 6.5 3.0 5.8 2.2

106 7.6 3.0 6.6 2.1

107 4.9 2.5 4.5 1.7

108 7.3 2.9 6.3 1.8

109 6.7 2.5 5.8 1.8

110 7.2 3.6 6.1 2.5

111 6.5 3.2 5.1 2.0

112 6.4 2.7 5.3 1.9

113 6.8 3.0 5.5 2.1

114 5.7 2.5 5.0 2.0

115 5.8 2.8 5.1 2.4

116 6.4 3.2 5.3 2.3

117 6.5 3.0 5.5 1.8

118 7.7 3.8 6.7 2.2

119 7.7 2.6 6.9 2.3

120 6.0 2.2 5.0 1.5

121 6.9 3.2 5.7 2.3

122 5.6 2.8 4.9 2.0

123 7.7 2.8 6.7 2.0

124 6.3 2.7 4.9 1.8

125 6.7 3.3 5.7 2.1

126 7.2 3.2 6.0 1.8

127 6.2 2.8 4.8 1.8

128 6.1 3.0 4.9 1.8

129 6.4 2.8 5.6 2.1

130 7.2 3.0 5.8 1.6

131 7.4 2.8 6.1 1.9

132 7.9 3.8 6.4 2.0

133 6.4 2.8 5.6 2.2

134 6.3 2.8 5.1 1.5

135 6.1 2.6 5.6 1.4

136 7.7 3.0 6.1 2.3

137 6.3 3.4 5.6 2.4

138 6.4 3.1 5.5 1.8

139 6.0 3.0 4.8 1.8

140 6.9 3.1 5.4 2.1

141 6.7 3.1 5.6 2.4

142 6.9 3.1 5.1 2.3

143 5.8 2.7 5.1 1.9

144 6.8 3.2 5.9 2.3

145 6.7 3.3 5.7 2.5

146 6.7 3.0 5.2 2.3

147 6.3 2.5 5.0 1.9

148 6.5 3.0 5.2 2.0

149 6.2 3.4 5.4 2.3

150 5.9 3.0 5.1 1.8

Species

1 setosa

2 setosa

3 setosa

4 setosa

5 setosa

6 setosa

7 setosa

8 setosa

9 setosa

10 setosa

11 setosa

12 setosa

13 setosa

14 setosa

15 setosa

16 setosa

17 setosa

18 setosa

19 setosa

20 setosa

21 setosa

22 setosa

23 setosa

24 setosa

25 setosa

26 setosa

27 setosa

28 setosa

29 setosa

30 setosa

31 setosa

32 setosa

33 setosa

34 setosa

35 setosa

36 setosa

37 setosa

38 setosa

39 setosa

40 setosa

41 setosa

42 setosa

43 setosa

44 setosa

45 setosa

46 setosa

47 setosa

48 setosa

49 setosa

50 setosa

51 versicolor

52 versicolor

53 versicolor

54 versicolor

55 versicolor

56 versicolor

57 versicolor

58 versicolor

59 versicolor

60 versicolor

61 versicolor

62 versicolor

63 versicolor

64 versicolor

65 versicolor

66 versicolor

67 versicolor

68 versicolor

69 versicolor

70 versicolor

71 versicolor

72 versicolor

73 versicolor

74 versicolor

75 versicolor

76 versicolor

77 versicolor

78 versicolor

79 versicolor

80 versicolor

81 versicolor

82 versicolor

83 versicolor

84 versicolor

85 versicolor

86 versicolor

87 versicolor

88 versicolor

89 versicolor

90 versicolor

91 versicolor

92 versicolor

93 versicolor

94 versicolor

95 versicolor

96 versicolor

97 versicolor

98 versicolor

99 versicolor

100 versicolor

101 virginica

102 virginica

103 virginica

104 virginica

105 virginica

106 virginica

107 virginica

108 virginica

109 virginica

110 virginica

111 virginica

112 virginica

113 virginica

114 virginica

115 virginica

116 virginica

117 virginica

118 virginica

119 virginica

120 virginica

121 virginica

122 virginica

123 virginica

124 virginica

125 virginica

126 virginica

127 virginica

128 virginica

129 virginica

130 virginica

131 virginica

132 virginica

133 virginica

134 virginica

135 virginica

136 virginica

137 virginica

138 virginica

139 virginica

140 virginica

141 virginica

142 virginica

143 virginica

144 virginica

145 virginica

146 virginica

147 virginica

148 virginica

149 virginica

150 virginica

> dim(iris)

[1] 150 5

> table(iris$Species)

setosa versicolor virginica

50 50 50

> set.seed(123)

> split = sample.split(iris$Species, SplitRatio = 0.7)#

> split

[1] TRUE FALSE TRUE FALSE FALSE TRUE TRUE FALSE

[9] TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE

[17] TRUE TRUE TRUE FALSE FALSE TRUE TRUE FALSE

[25] TRUE FALSE TRUE TRUE TRUE TRUE FALSE FALSE

[33] TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE

[41] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

[49] TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE

[57] TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE

[65] FALSE TRUE FALSE FALSE FALSE TRUE FALSE TRUE

[73] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

[81] TRUE FALSE TRUE FALSE TRUE TRUE FALSE FALSE

[89] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE

[97] FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE

[105] TRUE FALSE FALSE TRUE TRUE TRUE FALSE TRUE

[113] TRUE FALSE FALSE TRUE TRUE FALSE TRUE TRUE

[121] TRUE TRUE TRUE TRUE TRUE FALSE TRUE TRUE

[129] TRUE TRUE TRUE FALSE TRUE FALSE TRUE TRUE

[137] FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE

[145] FALSE TRUE TRUE TRUE TRUE FALSE

> #Creating the training set and test set separately

> training\_set = subset(iris, split == TRUE)

> test\_set = subset(iris, split == FALSE)

> training\_set

Sepal.Length Sepal.Width Petal.Length Petal.Width

1 5.1 3.5 1.4 0.2

3 4.7 3.2 1.3 0.2

6 5.4 3.9 1.7 0.4

7 4.6 3.4 1.4 0.3

9 4.4 2.9 1.4 0.2

10 4.9 3.1 1.5 0.1

12 4.8 3.4 1.6 0.2

13 4.8 3.0 1.4 0.1

14 4.3 3.0 1.1 0.1

15 5.8 4.0 1.2 0.2

17 5.4 3.9 1.3 0.4

18 5.1 3.5 1.4 0.3

19 5.7 3.8 1.7 0.3

22 5.1 3.7 1.5 0.4

23 4.6 3.6 1.0 0.2

25 4.8 3.4 1.9 0.2

27 5.0 3.4 1.6 0.4

28 5.2 3.5 1.5 0.2

29 5.2 3.4 1.4 0.2

30 4.7 3.2 1.6 0.2

33 5.2 4.1 1.5 0.1

35 4.9 3.1 1.5 0.2

36 5.0 3.2 1.2 0.2

38 4.9 3.6 1.4 0.1

39 4.4 3.0 1.3 0.2

40 5.1 3.4 1.5 0.2

41 5.0 3.5 1.3 0.3

42 4.5 2.3 1.3 0.3

43 4.4 3.2 1.3 0.2

44 5.0 3.5 1.6 0.6

45 5.1 3.8 1.9 0.4

46 4.8 3.0 1.4 0.3

47 5.1 3.8 1.6 0.2

48 4.6 3.2 1.4 0.2

49 5.3 3.7 1.5 0.2

51 7.0 3.2 4.7 1.4

52 6.4 3.2 4.5 1.5

54 5.5 2.3 4.0 1.3

55 6.5 2.8 4.6 1.5

56 5.7 2.8 4.5 1.3

57 6.3 3.3 4.7 1.6

60 5.2 2.7 3.9 1.4

61 5.0 2.0 3.5 1.0

62 5.9 3.0 4.2 1.5

63 6.0 2.2 4.0 1.0

64 6.1 2.9 4.7 1.4

66 6.7 3.1 4.4 1.4

70 5.6 2.5 3.9 1.1

72 6.1 2.8 4.0 1.3

74 6.1 2.8 4.7 1.2

75 6.4 2.9 4.3 1.3

76 6.6 3.0 4.4 1.4

77 6.8 2.8 4.8 1.4

78 6.7 3.0 5.0 1.7

79 6.0 2.9 4.5 1.5

80 5.7 2.6 3.5 1.0

81 5.5 2.4 3.8 1.1

83 5.8 2.7 3.9 1.2

85 5.4 3.0 4.5 1.5

86 6.0 3.4 4.5 1.6

90 5.5 2.5 4.0 1.3

91 5.5 2.6 4.4 1.2

92 6.1 3.0 4.6 1.4

93 5.8 2.6 4.0 1.2

94 5.0 2.3 3.3 1.0

95 5.6 2.7 4.2 1.3

96 5.7 3.0 4.2 1.2

98 6.2 2.9 4.3 1.3

99 5.1 2.5 3.0 1.1

100 5.7 2.8 4.1 1.3

101 6.3 3.3 6.0 2.5

102 5.8 2.7 5.1 1.9

103 7.1 3.0 5.9 2.1

105 6.5 3.0 5.8 2.2

108 7.3 2.9 6.3 1.8

109 6.7 2.5 5.8 1.8

110 7.2 3.6 6.1 2.5

112 6.4 2.7 5.3 1.9

113 6.8 3.0 5.5 2.1

116 6.4 3.2 5.3 2.3

117 6.5 3.0 5.5 1.8

119 7.7 2.6 6.9 2.3

120 6.0 2.2 5.0 1.5

121 6.9 3.2 5.7 2.3

122 5.6 2.8 4.9 2.0

123 7.7 2.8 6.7 2.0

124 6.3 2.7 4.9 1.8

125 6.7 3.3 5.7 2.1

127 6.2 2.8 4.8 1.8

128 6.1 3.0 4.9 1.8

129 6.4 2.8 5.6 2.1

130 7.2 3.0 5.8 1.6

131 7.4 2.8 6.1 1.9

133 6.4 2.8 5.6 2.2

135 6.1 2.6 5.6 1.4

136 7.7 3.0 6.1 2.3

140 6.9 3.1 5.4 2.1

141 6.7 3.1 5.6 2.4

142 6.9 3.1 5.1 2.3

143 5.8 2.7 5.1 1.9

144 6.8 3.2 5.9 2.3

146 6.7 3.0 5.2 2.3

147 6.3 2.5 5.0 1.9

148 6.5 3.0 5.2 2.0

149 6.2 3.4 5.4 2.3

Species

1 setosa

3 setosa

6 setosa

7 setosa

9 setosa

10 setosa

12 setosa

13 setosa

14 setosa

15 setosa

17 setosa

18 setosa

19 setosa

22 setosa

23 setosa

25 setosa

27 setosa

28 setosa

29 setosa

30 setosa

33 setosa

35 setosa

36 setosa

38 setosa

39 setosa

40 setosa

41 setosa

42 setosa

43 setosa

44 setosa

45 setosa

46 setosa

47 setosa

48 setosa

49 setosa

51 versicolor

52 versicolor

54 versicolor

55 versicolor

56 versicolor

57 versicolor

60 versicolor

61 versicolor

62 versicolor

63 versicolor

64 versicolor

66 versicolor

70 versicolor

72 versicolor

74 versicolor

75 versicolor

76 versicolor

77 versicolor

78 versicolor

79 versicolor

80 versicolor

81 versicolor

83 versicolor

85 versicolor

86 versicolor

90 versicolor

91 versicolor

92 versicolor

93 versicolor

94 versicolor

95 versicolor

96 versicolor

98 versicolor

99 versicolor

100 versicolor

101 virginica

102 virginica

103 virginica

105 virginica

108 virginica

109 virginica

110 virginica

112 virginica

113 virginica

116 virginica

117 virginica

119 virginica

120 virginica

121 virginica

122 virginica

123 virginica

124 virginica

125 virginica

127 virginica

128 virginica

129 virginica

130 virginica

131 virginica

133 virginica

135 virginica

136 virginica

140 virginica

141 virginica

142 virginica

143 virginica

144 virginica

146 virginica

147 virginica

148 virginica

149 virginica

> test\_set

Sepal.Length Sepal.Width Petal.Length Petal.Width

2 4.9 3.0 1.4 0.2

4 4.6 3.1 1.5 0.2

5 5.0 3.6 1.4 0.2

8 5.0 3.4 1.5 0.2

11 5.4 3.7 1.5 0.2

16 5.7 4.4 1.5 0.4

20 5.1 3.8 1.5 0.3

21 5.4 3.4 1.7 0.2

24 5.1 3.3 1.7 0.5

26 5.0 3.0 1.6 0.2

31 4.8 3.1 1.6 0.2

32 5.4 3.4 1.5 0.4

34 5.5 4.2 1.4 0.2

37 5.5 3.5 1.3 0.2

50 5.0 3.3 1.4 0.2

53 6.9 3.1 4.9 1.5

58 4.9 2.4 3.3 1.0

59 6.6 2.9 4.6 1.3

65 5.6 2.9 3.6 1.3

67 5.6 3.0 4.5 1.5

68 5.8 2.7 4.1 1.0

69 6.2 2.2 4.5 1.5

71 5.9 3.2 4.8 1.8

73 6.3 2.5 4.9 1.5

82 5.5 2.4 3.7 1.0

84 6.0 2.7 5.1 1.6

87 6.7 3.1 4.7 1.5

88 6.3 2.3 4.4 1.3

89 5.6 3.0 4.1 1.3

97 5.7 2.9 4.2 1.3

104 6.3 2.9 5.6 1.8

106 7.6 3.0 6.6 2.1

107 4.9 2.5 4.5 1.7

111 6.5 3.2 5.1 2.0

114 5.7 2.5 5.0 2.0

115 5.8 2.8 5.1 2.4

118 7.7 3.8 6.7 2.2

126 7.2 3.2 6.0 1.8

132 7.9 3.8 6.4 2.0

134 6.3 2.8 5.1 1.5

137 6.3 3.4 5.6 2.4

138 6.4 3.1 5.5 1.8

139 6.0 3.0 4.8 1.8

145 6.7 3.3 5.7 2.5

150 5.9 3.0 5.1 1.8

Species

2 setosa

4 setosa

5 setosa

8 setosa

11 setosa

16 setosa

20 setosa

21 setosa

24 setosa

26 setosa

31 setosa

32 setosa

34 setosa

37 setosa

50 setosa

53 versicolor

58 versicolor

59 versicolor

65 versicolor

67 versicolor

68 versicolor

69 versicolor

71 versicolor

73 versicolor

82 versicolor

84 versicolor

87 versicolor

88 versicolor

89 versicolor

97 versicolor

104 virginica

106 virginica

107 virginica

111 virginica

114 virginica

115 virginica

118 virginica

126 virginica

132 virginica

134 virginica

137 virginica

138 virginica

139 virginica

145 virginica

150 virginica

> table(test\_set$Species)

setosa versicolor virginica

15 15 15

> iris\_classifier=naiveBayes(Species ~ ., data = training\_set)

> iris\_classifier

Naive Bayes Classifier for Discrete Predictors

Call:

naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:

Y

setosa versicolor virginica

0.3333333 0.3333333 0.3333333

Conditional probabilities:

Sepal.Length

Y [,1] [,2]

setosa 4.940000 0.3541352

versicolor 5.920000 0.5166635

virginica 6.634286 0.5422952

Sepal.Width

Y [,1] [,2]

setosa 3.405714 0.3685766

versicolor 2.777143 0.3144423

virginica 2.925714 0.2831990

Petal.Length

Y [,1] [,2]

setosa 1.445714 0.1930298

versicolor 4.217143 0.4462166

virginica 5.565714 0.5075563

Petal.Width

Y [,1] [,2]

setosa 0.2428571 0.1092372

versicolor 1.3114286 0.1827429

virginica 2.0428571 0.2714728

> iris\_test\_pred=predict(iris\_classifier,test\_set)

> iris\_test\_pred

[1] setosa setosa

[3] setosa setosa

[5] setosa setosa

[7] setosa setosa

[9] setosa setosa

[11] setosa setosa

[13] setosa setosa

[15] setosa virginica

[17] versicolor versicolor

[19] versicolor versicolor

[21] versicolor versicolor

[23] virginica versicolor

[25] versicolor virginica

[27] versicolor versicolor

[29] versicolor versicolor

[31] virginica virginica

[33] versicolor virginica

[35] virginica virginica

[37] virginica virginica

[39] virginica versicolor

[41] virginica virginica

[43] virginica virginica

[45] virginica

3 Levels: setosa ... virginica

> table(test\_set$Species)

setosa versicolor virginica

15 15 15

> table(iris\_test\_pred)

iris\_test\_pred

setosa versicolor virginica

15 14 16

> table(iris\_test\_pred,test\_set$Species,dnn=c("Prediction","Actual"))

Actual

Prediction setosa versicolor virginica

setosa 15 0 0

versicolor 0 12 2

virginica 0 3 13

> iris\_classifier\_lap=naiveBayes(Species ~ ., data = training\_set,laplace=1)

> iris\_classifier\_lap

Naive Bayes Classifier for Discrete Predictors

Call:

naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:

Y

setosa versicolor virginica

0.3333333 0.3333333 0.3333333

Conditional probabilities:

Sepal.Length

Y [,1] [,2]

setosa 4.940000 0.3541352

versicolor 5.920000 0.5166635

virginica 6.634286 0.5422952

Sepal.Width

Y [,1] [,2]

setosa 3.405714 0.3685766

versicolor 2.777143 0.3144423

virginica 2.925714 0.2831990

Petal.Length

Y [,1] [,2]

setosa 1.445714 0.1930298

versicolor 4.217143 0.4462166

virginica 5.565714 0.5075563

Petal.Width

Y [,1] [,2]

setosa 0.2428571 0.1092372

versicolor 1.3114286 0.1827429

virginica 2.0428571 0.2714728

> table(iris\_test\_pred\_lab)

iris\_test\_pred\_lab

setosa versicolor virginica

15 14 16

> table(iris\_test\_pred,test\_set$Species,dnn=c("Prediction","Actual"))

Actual

Prediction setosa versicolor virginica

setosa 15 0 0

versicolor 0 12 2

virginica 0 3 13

cm=confusionMatrix(test\_set$Species,iris\_test\_pred)

> print(cm)

Confusion Matrix and Statistics

Reference

Prediction setosa versicolor virginica

setosa 15 0 0

versicolor 0 12 3

virginica 0 2 13

Overall Statistics

Accuracy : 0.8889

95% CI : (0.7595, 0.9629)

No Information Rate : 0.3556

P-Value [Acc > NIR] : 1.581e-13

Kappa : 0.8333

Mcnemar's Test P-Value : NA

Statistics by Class:

Class: setosa Class: versicolor

Sensitivity 1.0000 0.8571

Specificity 1.0000 0.9032

Pos Pred Value 1.0000 0.8000

Neg Pred Value 1.0000 0.9333

Prevalence 0.3333 0.3111

Detection Rate 0.3333 0.2667

Detection Prevalence 0.3333 0.3333

Balanced Accuracy 1.0000 0.8802

Class: virginica

Sensitivity 0.8125

Specificity 0.9310

Pos Pred Value 0.8667

Neg Pred Value 0.9000

Prevalence 0.3556

Detection Rate 0.2889

Detection Prevalence 0.3333

Balanced Accuracy 0.8718

**5. Write a Program to perform naïve bayes classification on Titanic dataset. Perform data pre-processing if required.**

> Titanic

, , Age = Child, Survived = No

Sex

Class Male Female

1st 0 0

2nd 0 0

3rd 35 17

Crew 0 0

, , Age = Adult, Survived = No

Sex

Class Male Female

1st 118 4

2nd 154 13

3rd 387 89

Crew 670 3

, , Age = Child, Survived = Yes

Sex

Class Male Female

1st 5 1

2nd 11 13

3rd 13 14

Crew 0 0

, , Age = Adult, Survived = Yes

Sex

Class Male Female

1st 57 140

2nd 14 80

3rd 75 76

Crew 192 20

> class(Titanic)

[1] "table"

> head(Titanic)

, , Age = Child, Survived = No

Sex

Class Male Female

1st 0 0

2nd 0 0

3rd 35 17

Crew 0 0

, , Age = Adult, Survived = No

Sex

Class Male Female

1st 118 4

2nd 154 13

3rd 387 89

Crew 670 3

, , Age = Child, Survived = Yes

Sex

Class Male Female

1st 5 1

2nd 11 13

3rd 13 14

Crew 0 0

, , Age = Adult, Survived = Yes

Sex

Class Male Female

1st 57 140

2nd 14 80

3rd 75 76

Crew 192 20

> str(Titanic)

'table' num [1:4, 1:2, 1:2, 1:2] 0 0 35 0 0 0 17 0 118 154 ...

- attr(\*, "dimnames")=List of 4

..$ Class : chr [1:4] "1st" "2nd" "3rd" "Crew"

..$ Sex : chr [1:2] "Male" "Female"

..$ Age : chr [1:2] "Child" "Adult"

..$ Survived: chr [1:2] "No" "Yes"

> dfdata <- as.data.frame(Titanic)

> dfdata

Class Sex Age Survived Freq

1 1st Male Child No 0

2 2nd Male Child No 0

3 3rd Male Child No 35

4 Crew Male Child No 0

5 1st Female Child No 0

6 2nd Female Child No 0

7 3rd Female Child No 17

8 Crew Female Child No 0

9 1st Male Adult No 118

10 2nd Male Adult No 154

11 3rd Male Adult No 387

12 Crew Male Adult No 670

13 1st Female Adult No 4

14 2nd Female Adult No 13

15 3rd Female Adult No 89

16 Crew Female Adult No 3

17 1st Male Child Yes 5

18 2nd Male Child Yes 11

19 3rd Male Child Yes 13

20 Crew Male Child Yes 0

21 1st Female Child Yes 1

22 2nd Female Child Yes 13

23 3rd Female Child Yes 14

24 Crew Female Child Yes 0

25 1st Male Adult Yes 57

26 2nd Male Adult Yes 14

27 3rd Male Adult Yes 75

28 Crew Male Adult Yes 192

29 1st Female Adult Yes 140

30 2nd Female Adult Yes 80

31 3rd Female Adult Yes 76

32 Crew Female Adult Yes 20

> names(dfdata)

[1] "Class" "Sex"

[3] "Age" "Survived"

[5] "Freq"

> dim(dfdata)

[1] 32 5

> set.seed(123)

> split=sample.split(df\_data$Survived,SplitRatio = 0.7)

> split

[1] TRUE TRUE TRUE FALSE FALSE TRUE TRUE FALSE TRUE TRUE

[11] FALSE TRUE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE

[21] FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE TRUE TRUE

[31] FALSE FALSE

> training\_set1=subset(dfdata,split==TRUE)

> training\_set1

Class Sex Age Survived Freq

1 1st Male Child No 0

2 2nd Male Child No 0

3 3rd Male Child No 35

6 2nd Female Child No 0

7 3rd Female Child No 17

9 1st Male Adult No 118

10 2nd Male Adult No 154

12 Crew Male Adult No 670

13 1st Female Adult No 4

14 2nd Female Adult No 13

15 3rd Female Adult No 89

17 1st Male Child Yes 5

18 2nd Male Child Yes 11

19 3rd Male Child Yes 13

22 2nd Female Child Yes 13

23 3rd Female Child Yes 14

25 1st Male Adult Yes 57

26 2nd Male Adult Yes 14

27 3rd Male Adult Yes 75

28 Crew Male Adult Yes 192

29 1st Female Adult Yes 140

30 2nd Female Adult Yes 80

> nrow(training\_set1)

[1] 22

> ncol(training\_set1)

[1] 5

> test\_set1 = subset(dfdata, t\_split == FALSE)

> test\_set1

Class Sex Age

5 1st Female Child

11 3rd Male Adult

16 Crew Female Adult

20 Crew Male Child

24 Crew Female Child

31 3rd Female Adult

Survived Freq

5 No 0

11 No 387

16 No 3

20 Yes 0

24 Yes 0

31 Yes 76

>table(test\_set1$Survived)

No Yes

3 3

> titanic\_classifier=naiveBayes(Survived ~ ., data = training\_set1)

> titanic\_classifier

Naive Bayes Classifier for Discrete Predictors

Call:

naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:

Y

No Yes

0.5 0.5

Conditional probabilities:

Class

Y 1st 2nd

No 0.2307692 0.3076923

Yes 0.3076923 0.3076923

Class

Y 3rd Crew

No 0.2307692 0.2307692

Yes 0.2307692 0.1538462

Sex

Y Male Female

No 0.5384615 0.4615385

Yes 0.5384615 0.4615385

Age

Y Child Adult

No 0.5384615 0.4615385

Yes 0.4615385 0.5384615

Freq

Y [,1] [,2]

No 84.61538 183.27645

Yes 48.84615 59.15917

> titanic\_test\_pred=predict(titanic\_classifier,test\_set1)

> titanic\_test\_pred

[1] Yes No Yes Yes Yes Yes

Levels: No Yes

> table(titanic\_test\_pred)

titanic\_test\_pred

No Yes

1 5

> table(titanic\_test\_pred, test\_set1$Survived,dnn=c("Prediction","Actual"))

Actual

Prediction No Yes

No 1 0

Yes 2 3

table(titanic\_test\_pred, test\_set1$Survived,dnn=c("Prediction","Actual"))

Actual

Prediction No Yes

No 1 0

Yes 2 3

>

> cm\_titanic = confusionMatrix(test\_set1$Survived, titanic\_test\_pred)

>

> cm\_titanic

Confusion Matrix and Statistics

Reference

Prediction No Yes

No 1 2

Yes 0 3

Accuracy : 0.6667

95% CI : (0.2228, 0.9567)

No Information Rate : 0.8333

P-Value [Acc > NIR] : 0.9377

Kappa : 0.3333

Mcnemar's Test P-Value : 0.4795

Sensitivity : 1.0000

Specificity : 0.6000

Pos Pred Value : 0.3333

Neg Pred Value : 1.0000

Prevalence : 0.1667

Detection Rate : 0.1667

Detection Prevalence : 0.5000

Balanced Accuracy : 0.8000

'Positive' Class : No