Introduction to Data Science

Finalterm Project

Name	ID	Section
Amit Podder	20-42273-1	C
Writhik Banik Barshan	20-43124-1	C

Topic: Cervical Cancer Dataset

Dataset Description:

The dataset is about Cervical Cancer. When cancer starts in the cervix, it is called cervical cancer.

Cervical cancer is cancer that starts in the cells of the cervix. The cervix is the lower, narrow end of the uterus (womb). The cervix connects the uterus to the vagina (birth canal). Cervical cancer usually develops slowly over time. Before cancer appears in the cervix, the cells of the cervix go through changes known as dysplasia, in which abnormal cells begin to appear in the cervical tissue. Over time, if not destroyed or removed, the abnormal cells may become cancer cells and start to grow and spread more deeply into the cervix and to surrounding areas. Anyone with a cervix is at risk for cervical cancer. It occurs most often in people over age 30. Long-lasting infection with certain types of human papillomavirus (HPV) is the main cause of cervical cancer. HPV is a common virus that is passed from one person to another during sex. At least half of sexually active people will have HPV at some point in their lives, but few women will get cervical cancer.

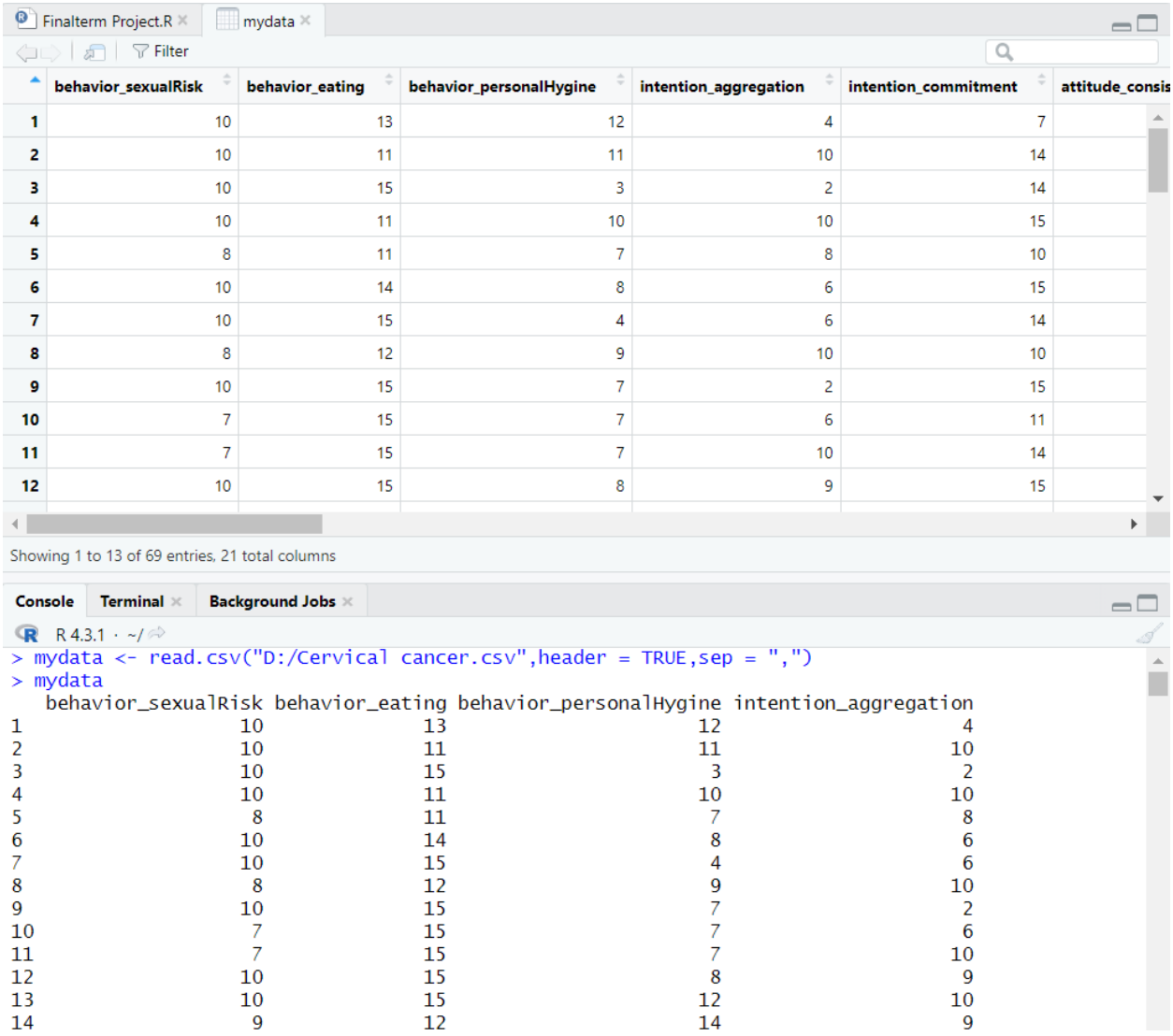
Screening tests and the HPV vaccine can help prevent cervical cancer. When cervical cancer is found early, it is highly treatable and associated with long survival and good quality of life.

Import the dataset as CSV and print the dataset:

Code:

```
mydata <- read.csv("D:/Cervical cancer.csv",header = TRUE,sep = ",")
mydata
```

Output:



Finalterm Project.R | mydata

Filter

	behavior_sexualRisk	behavior_eating	behavior_personalHygine	intention_aggregation	intention_commitment	attitude_consis
1	10	13	12	4	7	
2	10	11	11	10	14	
3	10	15	3	2	14	
4	10	11	10	10	15	
5	8	11	7	8	10	
6	10	14	8	6	15	
7	10	15	4	6	14	
8	8	12	9	10	10	
9	10	15	7	2	15	
10	7	15	7	6	11	
11	7	15	7	10	14	
12	10	15	8	9	15	

Showing 1 to 13 of 69 entries, 21 total columns

Console | Terminal | Background Jobs

R 4.3.1 · ~/

```
> mydata <- read.csv("D:/Cervical cancer.csv",header = TRUE,sep = ",")
> mydata
   behavior_sexualRisk behavior_eating behavior_personalHygine intention_aggregation
1                   10              13                      12                     4
2                   10              11                      11                    10
3                   10              15                       3                     2
4                   10              11                      10                    10
5                    8              11                       7                     8
6                   10              14                       8                     6
7                   10              15                       4                     6
8                    8              12                       9                    10
9                   10              15                       7                     2
10                   7              15                       7                     6
11                   7              15                       7                    10
12                  10              15                       8                     9
13                  10              15                      12                    10
14                   9              12                      14                     9
```

Description:

Here we have imported the code of the dataset as a csv file. We can also see the output of the dataset imported in RStudio.

To see the column name of the dataset:

Code:

names(mydata)

Output:

```
> names(mydata)
 [1] "behavior_sexualRisk"         "behavior_eating"             "behavior_personalHygine"
 [4] "intention_aggregation"       "intention_commitment"        "attitude_consistency"
 [7] "attitude_spontaneity"        "norm_significantPerson"      "norm_fulfillment"
[10] "perception_vulnerability"    "perception_severity"         "motivation_strength"
[13] "motivation_willingness"      "socialSupport_emotionality"  "socialSupport_appreciation"
[16] "socialSupport_instrumental"  "empowerment_knowledge"       "empowerment_abilities"
[19] "empowerment_desires"         "ca_cervix"                   "Result"
```

Description:

In this code, we can see the column name of the dataset. Here, with the help of the code names(), we can see all the attribute names present in the dataset.

Finding the Missing(Null) values:

Code:

colSums(is.na(mydata))

Output:

```
> colSums(is.na(mydata))
       behavior_sexualRisk             behavior_eating    behavior_personalHygine
                         0                           0                          0
     intention_aggregation       intention_commitment       attitude_consistency
                         0                           0                          0
      attitude_spontaneity     norm_significantPerson           norm_fulfillment
                         0                           0                          0
  perception_vulnerability        perception_severity        motivation_strength
                         0                           0                          0
    motivation_willingness socialSupport_emotionality socialSupport_appreciation
                         0                           0                          0
socialSupport_instrumental      empowerment_knowledge      empowerment_abilities
                         0                           0                          0
       empowerment_desires                   ca_cervix                     Result
                         0                           0                          0
```

Description:

In this code, we can see all the null values of the dataset. Here, with the help of the code colSums(is.na()), we can check missing values in each column.

Check whether all the data is numeric or not:

all_numeric <- sapply(mydata, is.numeric)

print(all_numeric)

Output:

```
> all_numeric <- sapply(mydata, is.numeric)
> print(all_numeric)
       behavior_sexualRisk             behavior_eating    behavior_personalHygine
                      TRUE                        TRUE                       TRUE
     intention_aggregation       intention_commitment       attitude_consistency
                      TRUE                        TRUE                       TRUE
      attitude_spontaneity     norm_significantPerson           norm_fulfillment
                      TRUE                        TRUE                       TRUE
  perception_vulnerability        perception_severity        motivation_strength
                      TRUE                        TRUE                       TRUE
    motivation_willingness socialSupport_emotionality socialSupport_appreciation
                      TRUE                        TRUE                       TRUE
socialSupport_instrumental      empowerment_knowledge      empowerment_abilities
                      TRUE                        TRUE                       TRUE
       empowerment_desires                   ca_cervix                     Result
                      TRUE                        TRUE                      FALSE
```

Description:

Here, we are using this code to check whether all the data is numeric or not. After implementing the code, we can see that all the data is showing true which means all the data is numeric.

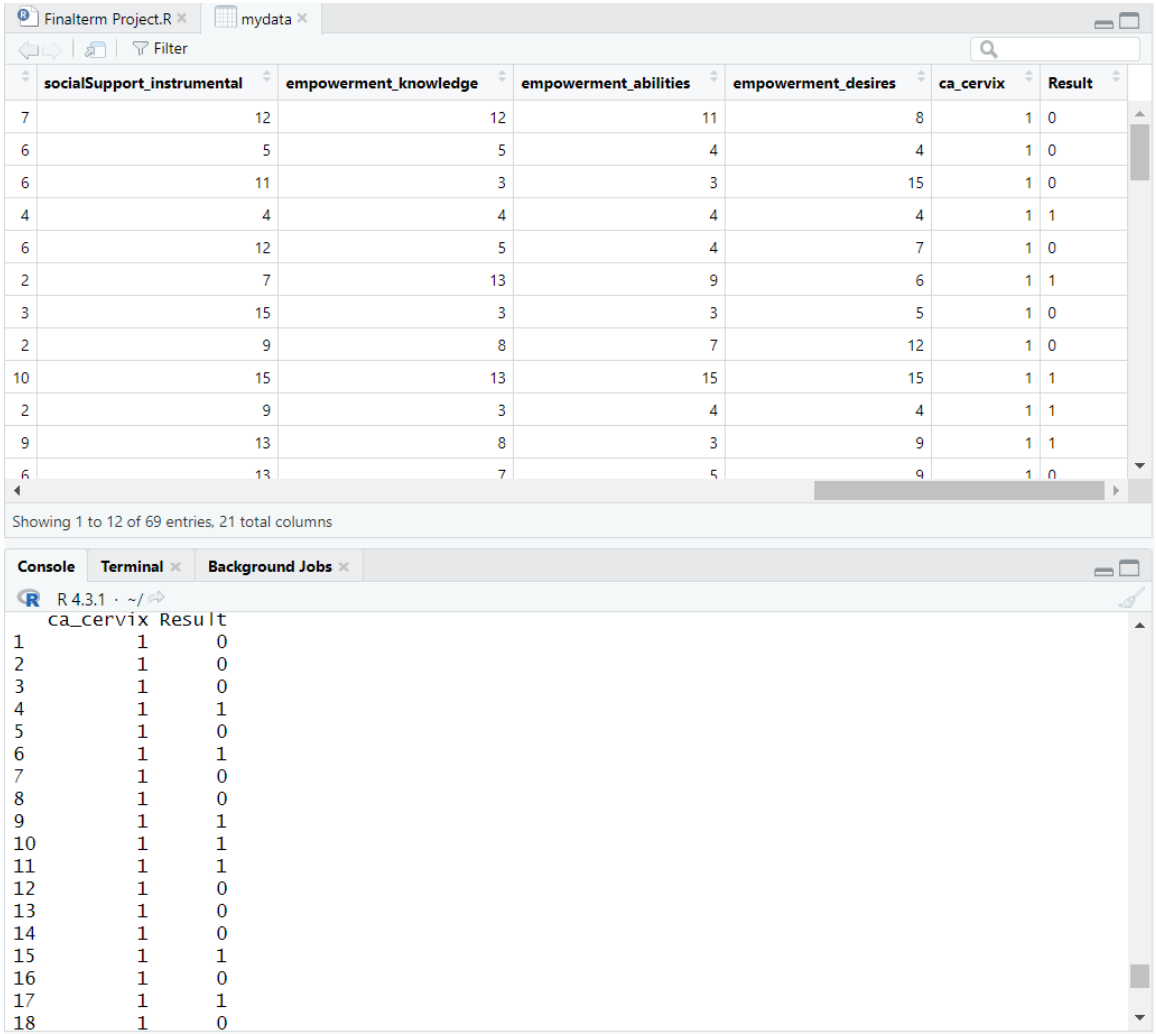
Annotating Datasets:

Code:

mydata$Result <- factor(mydata$Result, levels = c("Positive","Negetive"),labels = c("0","1"))

mydata

Output:



Description:

Here, the “Result” column is converted into numeric(0 and 1) where ‘0’ represents ‘Positive’ and ‘1’ represents ‘Negetive’. With the help of the code mydata$Result <- factor(mydata$Result, levels = c("Positive","Negetive"),labels = c("0","1")), we were able to successfully convert ‘Positive’ and ‘Negetive’ into numeric ‘0’ and ‘1’.

Summary of the structure of the dataset:

Code:

```
str(mydata)
```

Output:

```
> str(mydata)
'data.frame':   69 obs. of  21 variables:
 $ behavior_sexualRisk       : int  10 10 10 10 8 10 10 8 10 7 ...
 $ behavior_eating           : int  13 11 15 11 11 14 15 12 15 15 ...
 $ behavior_personalHygine   : int  12 11 3 10 7 8 4 9 7 7 ...
 $ intention_aggregation     : int  4 10 2 10 8 6 6 10 2 6 ...
 $ intention_commitment      : int  7 14 14 15 10 15 14 10 15 11 ...
 $ attitude_consistency      : int  9 7 8 7 7 8 6 5 6 8 ...
 $ attitude_spontaneity      : int  10 7 10 7 8 10 10 10 10 8 ...
 $ norm_significantPerson    : int  1 5 1 1 1 1 5 5 1 5 ...
 $ norm_fulfillment          : int  8 5 4 5 5 3 3 5 3 3 ...
 $ perception_vulnerability  : int  7 4 7 4 3 4 7 5 5 3 ...
 $ perception_severity       : int  3 2 2 2 2 2 2 2 2 4 ...
 $ motivation_strength       : int  14 15 7 15 15 14 7 10 9 15 ...
 $ motivation_willingness    : int  8 13 3 13 5 8 13 9 15 3 ...
 $ socialSupport_emotionality: int  5 7 3 7 3 7 3 13 13 8 ...
 $ socialSupport_appreciation: int  7 6 6 4 6 2 3 2 10 2 ...
 $ socialSupport_instrumental: int  12 5 11 4 12 7 15 9 15 9 ...
 $ empowerment_knowledge     : int  12 5 3 4 5 13 3 8 13 3 ...
 $ empowerment_abilities     : int  11 4 3 4 4 9 3 7 15 4 ...
 $ empowerment_desires       : int  8 4 15 4 7 6 5 12 15 4 ...
 $ ca_cervix                 : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Result                    : Factor w/ 2 levels "0","1": 1 1 1 2 1 2 1 1 2 2 ...
>
```

Description:

The structure of the dataset is displayed with the help of the code str().

Descriptive Statistics:

Code:

summary(mydata)

Output:

```
> summary(mydata)
 behavior_sexualRisk behavior_eating behavior_personalHygine intention_aggregation
 Min.   : 2.000      Min.   : 3.00   Min.   : 3.00           Min.   : 2.000
 1st Qu.:10.000      1st Qu.:11.00   1st Qu.: 9.00           1st Qu.: 6.000
 Median :10.000      Median :13.00   Median :11.00           Median :10.000
 Mean   : 9.667      Mean   :12.86   Mean   :11.01           Mean   : 7.928
 3rd Qu.:10.000      3rd Qu.:15.00   3rd Qu.:14.00           3rd Qu.:10.000
 Max.   :10.000      Max.   :15.00   Max.   :15.00           Max.   :10.000
 intention_commitment attitude_consistency attitude_spontaneity norm_significantPerson
 Min.   : 6.00        Min.   : 2.000       Min.   : 5.000       Min.   :1.000
 1st Qu.:11.00        1st Qu.: 6.000       1st Qu.: 8.000       1st Qu.:1.000
 Median :15.00        Median : 7.000       Median : 9.000       Median :3.000
 Mean   :13.42        Mean   : 7.217       Mean   : 8.725       Mean   :3.072
 3rd Qu.:15.00        3rd Qu.: 8.000       3rd Qu.:10.000       3rd Qu.:5.000
 Max.   :15.00        Max.   :10.000       Max.   :10.000       Max.   :5.000
 norm_fulfillment perception_vulnerability perception_severity motivation_strength
 Min.   : 3.000   Min.   : 3.000           Min.   : 2.000      Min.   : 3.00
 1st Qu.: 3.000   1st Qu.: 5.000           1st Qu.: 2.000      1st Qu.:11.00
 Median : 7.000   Median : 7.000           Median : 4.000      Median :14.00
 Mean   : 8.246   Mean   : 8.391           Mean   : 5.217      Mean   :12.67
 3rd Qu.:14.000   3rd Qu.:13.000           3rd Qu.: 9.000      3rd Qu.:15.00
 Max.   :15.000   Max.   :15.000           Max.   :10.000      Max.   :15.00
 motivation_willingness socialSupport_emotionality socialSupport_appreciation
 Min.   : 3.00          Min.   : 3                 Min.   : 2.000
 1st Qu.: 7.00          1st Qu.: 3                 1st Qu.: 3.000
 Median :10.00          Median : 9                 Median : 6.000
 Mean   : 9.58          Mean   : 8                 Mean   : 6.087
 3rd Qu.:13.00          3rd Qu.:12                 3rd Qu.: 9.000
 Max.   :15.00          Max.   :15                 Max.   :10.000
 socialSupport_instrumental empowerment_knowledge empowerment_abilities empowerment_desires
 Min.   : 3.00              Min.   : 3.00         Min.   : 3.000        Min.   : 3.00
 1st Qu.: 6.00              1st Qu.: 7.00         1st Qu.: 5.000        1st Qu.: 6.00
 Median :12.00              Median :12.00         Median :10.000        Median :11.00
 Mean   :10.29              Mean   :10.48         Mean   : 9.174        Mean   :10.14
 3rd Qu.:15.00              3rd Qu.:15.00         3rd Qu.:13.000        3rd Qu.:15.00
 Max.   :15.00              Max.   :15.00         Max.   :15.000        Max.   :15.00
```

Description:

Here, we are using this code to see the descriptive statistics. To see the descriptive statistics, we use the summary() function. We can also see the min, max, mean, and median values of the dataset.

Using the correlation technique:

Code:

```
all_numeric <- mydata[, sapply(mydata, is.numeric)]

correlation_matrix <- cor(all_numeric)

dim(correlation_matrix)

correlation_matrix
```

Output:

```
> all_numeric <- mydata[, sapply(mydata, is.numeric)]
> correlation_matrix <- cor(all_numeric)
> dim(correlation_matrix)
[1] 20 20
> correlation_matrix
                        behavior_sexualRisk behavior_eating behavior_personalHygine
behavior_sexualRisk              1.000000000    -0.168530745              0.005284053
behavior_eating                 -0.168530745     1.000000000              0.230107989
behavior_personalHygine          0.005284053     0.230107989              1.000000000
intention_aggregation            0.005863048     0.109909165              0.453658955
intention_commitment             0.136041523     0.082417866              0.019238824
attitude_consistency            -0.071004758     0.102829883              0.163930285
attitude_spontaneity            -0.063617649     0.243652346             -0.108341853
norm_significantPerson           0.063318462     0.036371391              0.218990634
norm_fulfillment                 0.164061111    -0.026564712              0.232852396
perception_vulnerability         0.183407730     0.002794813              0.121751042
perception_severity              0.072328708    -0.061464379              0.230322278
motivation_strength             -0.039848614    -0.128303938              0.403446985
motivation_willingness           0.312342151    -0.066599413              0.424340170
socialSupport_emotionality       0.079145323    -0.074638185              0.379441651
socialSupport_appreciation       0.103843702    -0.008882999              0.341608848
socialSupport_instrumental       0.104569339     0.064536167              0.083834615
empowerment_knowledge            0.172344254     0.069094734              0.441018308
empowerment_abilities            0.211977130    -0.016930187              0.375885706
empowerment_desires              0.295242518     0.050961110              0.179499739
ca_cervix                       -0.315012669     0.190873278             -0.364836573
                        intention_aggregation intention_commitment attitude_consistency
behavior_sexualRisk                0.005863048         0.1360415233          -0.07100476
behavior_eating                    0.109909165         0.0824178658           0.10282988
behavior_personalHygine            0.453658955         0.0192388240           0.16393028
intention_aggregation              1.000000000         0.2542574112          -0.04105323
intention_commitment               0.254257411         1.0000000000          -0.02520478
attitude_consistency              -0.041053234        -0.0252047778           1.00000000
attitude_spontaneity              -0.213627130         0.1801790396           0.16405191
norm_significantPerson             0.118341808         0.0228986164           0.20488417
norm_fulfillment                   0.072278227         0.0010825834           0.23750732
perception_vulnerability          -0.048120761        -0.0004973524           0.19930385
perception_severity                0.071221537         0.0086054660           0.26850357
```

Description:

Here, we are using this code to create a correlation matrix. After implementing the code, we were able to create the correlation matrix for all numeric variables.

Find the significant attributes using the correlation technique:

Code:

```
significant_attributes <- names(which(abs(correlation_matrix[1, ]) > 0.2))

significant_attributes
```

Output:

```
> significant_attributes <- names(which(abs(correlation_matrix[1, ]) > 0.2))
> significant_attributes
[1] "behavior_sexualRisk"    "motivation_willingness" "empowerment_abilities"
[4] "empowerment_desires"    "ca_cervix"
>
```

Description:

Here, we are using this code to find the significant attributes of the correlation matrix. After implementing the code, we were able to find the significant attributes.

Applying the Naïve Bayes algorithm:

Code:

```
install.packages("e1071")

install.packages("caret")

install.packages("naivebayes")


library("e1071")

library("caret")

library("naivebayes")


naive_bayes_model <- naiveBayes(mydata[, -ncol(mydata)], mydata[,
ncol(mydata)])

naive_bayes_model
```

Output:

```
> naive_bayes_model <- naiveBayes(mydata[, -ncol(mydata)], mydata[, ncol(mydata)])
> naive_bayes_model

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = mydata[, -ncol(mydata)], y = mydata[,
    ncol(mydata)])

A-priori probabilities:
mydata[, ncol(mydata)]
        0         1
0.5507246 0.4492754

Conditional probabilities:
                      behavior_sexualRisk
mydata[, ncol(mydata)]     [,1]      [,2]
                     0 9.763158 0.7861714
                     1 9.548387 1.5882634

                      behavior_eating
mydata[, ncol(mydata)]     [,1]     [,2]
                     0 12.78947 1.974950
                     1 12.93548 2.743908

                      behavior_personalHygine
mydata[, ncol(mydata)]     [,1]     [,2]
                     0 10.97368 3.208829
                     1 11.06452 2.943190

                      intention_aggregation
mydata[, ncol(mydata)]     [,1]     [,2]
                     0 8.342105 2.385500
                     1 7.419355 3.138728
```

Description:

Here, we have used this code to apply the naïve Bayes algorithm. After implementing the code, we were able to apply the naïve Bayes algorithm for all attributes.

Applying the Naïve Bayes algorithm for significant attributes, dividing the data into training and test sets, finding its accuracy, and generating the confusion matrix:

Code:

```
set.seed(123)

predictors <- colnames(mydata)[colnames(mydata) != "ca_cervix"]
target <- "ca_cervix"

split_index <- sample(1:nrow(mydata), 0.7 * nrow(mydata))
train_mydata <- mydata[split_index, ]
test_mydata <- mydata[-split_index, ]

nb_model <- naiveBayes(train_mydata[, predictors], train_mydata$ca_cervix)

unseen_instance <- test_mydata[1, predictors, drop = FALSE]

prediction <- predict(nb_model, unseen_instance)

cat("Original Class:", test_mydata[1, "ca_cervix"], "\n")
cat("Predicted Class:", prediction, "\n")
predictions <- predict(nb_model, test_mydata[, predictors])

conf_matrix <- table(predictions, test_mydata$ca_cervix)
print("Confusion Matrix:")
print(conf_matrix)
```

accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)

cat("Accuracy:", round(accuracy, 2), "\n")

Output:

```
> set.seed(123)
> predictors <- colnames(mydata)[colnames(mydata) != "ca_cervix"]
> target <- "ca_cervix"
> split_index <- sample(1:nrow(mydata), 0.7 * nrow(mydata))
> train_mydata <- mydata[split_index, ]
> test_mydata <- mydata[-split_index, ]
> nb_model <- naiveBayes(train_mydata[, predictors], train_mydata$ca_cervix)
> unseen_instance <- test_mydata[1, predictors, drop = FALSE]
> prediction <- predict(nb_model, unseen_instance)
> cat("Original Class:", test_mydata[1, "ca_cervix"], "\n")
Original Class: 1
> cat("Predicted Class:", prediction, "\n")
Predicted Class: 2
> predictions <- predict(nb_model, test_mydata[, predictors])
> conf_matrix <- table(predictions, test_mydata$ca_cervix)
> print("Confusion Matrix:")
[1] "Confusion Matrix:"
> print(conf_matrix)

predictions  0  1
          0 12  1
          1  1  7
>
```

```
> accuracy <- sum(diag(conf_matrix)) / sum(conf_matrix)
> cat("Accuracy:", round(accuracy, 2), "\n")
Accuracy: 0.9
>
```

Description:

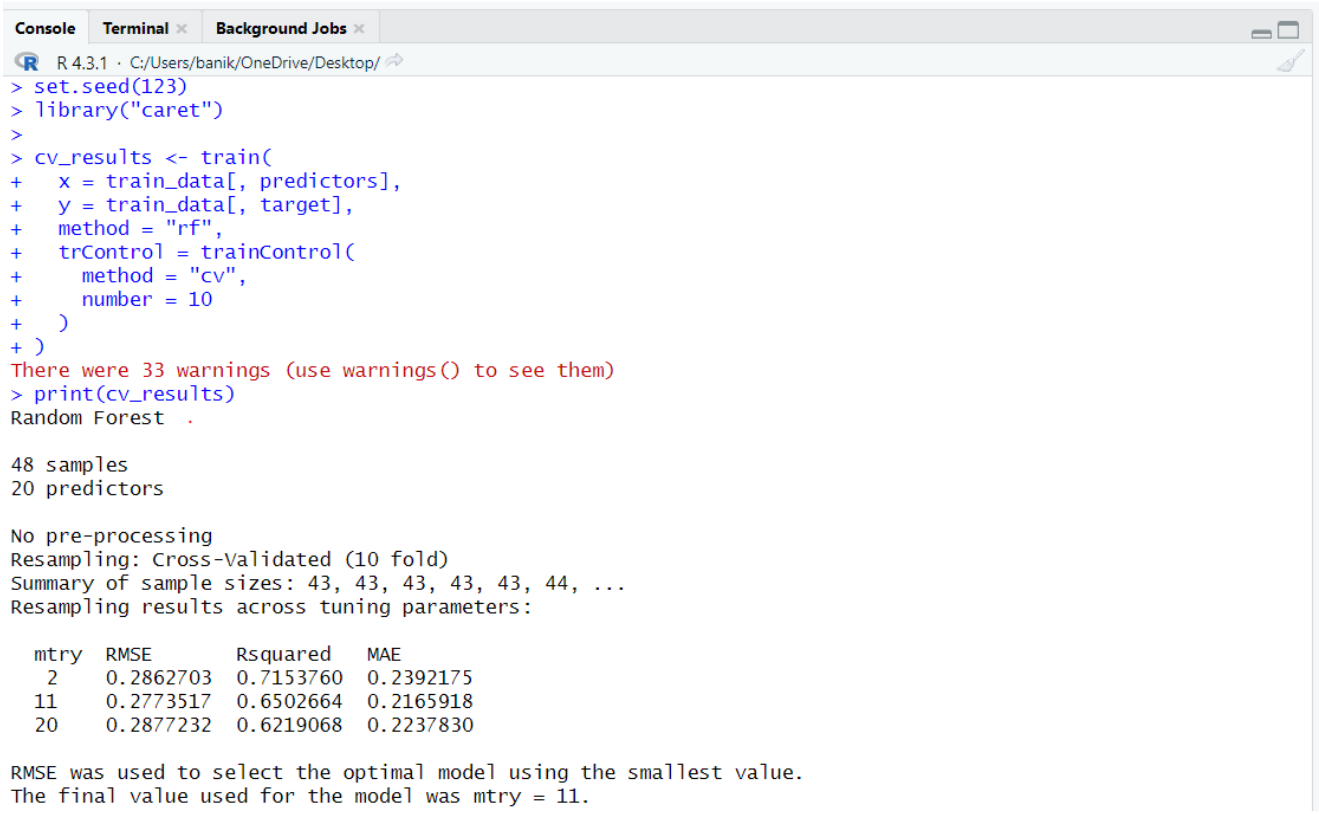
Here, at first, we have used the code to apply the naïve Bayes algorithm for one of the significant attributes “ca_cervix”. We have also divided the data into training and test sets to find its accuracy. Finally, we have generated the confusion matrix for our dataset using the naïve Bayes classifier.

Applying the 10-fold cross-validation:

Code:

```
set.seed(123)
library("caret")


cv_results <- train(
  x = train_data[, predictors],
  y = train_data[, target],
  method = "rf",
  trControl = trainControl(
    method = "cv",
    number = 10
  )
)
print(cv_results)
```

Output:



Description:

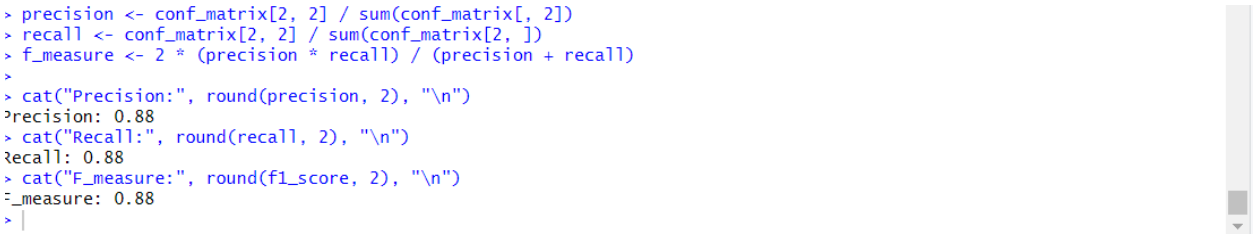
Here, we are displaying the result of cross-validation for the random forest model. The performance of a random forest model of our dataset is demonstrated by this output, which uses various values of mtry. We may get insight into the performance of the model using the metrics RMSE, Rsquared, and MAE. With lower RMSE and MAE values higher Rsquared values indicate greater performance. We may assess and pick the ideal hyperparameter configuration for the random forest model using the values of these metrics for each model setup.

Calculate the Precision, Recall, and F-measure value of the confusion matrix:

Code:

```
precision <- conf_matrix[2, 2] / sum(conf_matrix[, 2])

recall <- conf_matrix[2, 2] / sum(conf_matrix[2, ])

f_measure <- 2 * (precision * recall) / (precision + recall)


cat("Precision:", round(precision, 2), "\n")

cat("Recall:", round(recall, 2), "\n")

cat("F_measure:", round(f1_score, 2), "\n")
```

Output:



Description:

Here, we are using this code to calculate the recall, precision, and f-measure values of the confusion matrix. After implementing the code, we were able to calculate the recall, precision, and f-measure values of the confusion matrix.