```
R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> GroupData<-read.csv(file.choose(),header=T)
> head(GroupData)
     brand model_year milage                                accident HighValueCar
1     Ford       2013  51000 At least 1 accident or damage reported            0
2  Hyundai       2021  34742 At least 1 accident or damage reported            0
3    Lexus       2022  22372                            None reported            1
4 INFINITI       2015  88900                            None reported            0
5     Audi       2021   9835                            None reported            0
6    Acura       2016 136397                            None reported            0
> install.packages('dplyr')
Installing package into 'C:/Users/15039/AppData/Local/R/win-library/4.3'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
Warning: failed to download mirrors file (cannot open URL 'https://cran.r-project.org/CRAN_mirror
s.csv'); using local file 'C:/PROGRA~1/R/R-43~1.1/doc/CRAN_mirrors.csv'
trying URL 'https://ftp.osuosl.org/pub/cran/bin/windows/contrib/4.3/dplyr_1.1.3.zip'
Content type 'application/zip' length 1554796 bytes (1.5 MB)
downloaded 1.5 MB

package 'dplyr' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\15039\AppData\Local\Temp\Rtmp2VQgwT\downloaded_packages
Warning message:
In download.file(url, destfile = f, quiet = TRUE) :
  URL 'https://cran.r-project.org/CRAN_mirrors.csv': Timeout of 60 seconds was reached
> library(dplyr)

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union

Warning message:
package 'dplyr' was built under R version 4.3.2
> library(reshape2)
> GroupCor<-as.matrix(cor(GroupData))
Error in cor(GroupData) : 'x' must be numeric
> library(nnet)
> set.seed(1000)
> GroupNN<-nnet(HighValueCar ~., data=GroupData,size=8,maxit=10000)
# weights:  449
initial  value 680.230145
iter  10 value 518.422244
```

```
iter  20 value 508.815687
iter  30 value 508.617745
iter  40 value 479.646609
iter  50 value 479.394946
iter  60 value 478.270886
final  value 478.270440
converged
> GroupNN<-nnet(HighValueCar ~., data=GroupData,size=8,maxit=100000)
# weights:  449
initial  value 1031.994006
final  value 673.000000
converged
> GroupData<-read.csv(file.choose(),header=T)
> head(GroupData)
     brand model_year milage                                accident HighValueCar
1     Ford       2013  51000 At least 1 accident or damage reported            0
2  Hyundai       2021  34742 At least 1 accident or damage reported            0
3    Lexus       2022  22372                           None reported            1
4 INFINITI       2015  88900                           None reported            0
5     Audi       2021   9835                           None reported            0
6    Acura       2016 136397                           None reported            0
> GroupNN<-nnet(HighValueCar ~., data=GroupData,size=8,maxit=100000)
# weights:  449
initial  value 590.553661
iter  10 value 461.766115
iter  20 value 456.887098
iter  30 value 454.450864
iter  40 value 454.192167
iter  50 value 454.086010
iter  60 value 453.993514
iter  70 value 453.606660
iter  80 value 453.450354
iter  90 value 453.308337
final  value 453.308167
converged
> GroupNN <- nnet(HighValueCar ~ ., data = GroupData, size = 4, maxit = 100000, decay = 0.01)
# weights:  225
initial  value 1186.097376
iter  10 value 695.259666
iter  20 value 582.698432
iter  30 value 487.220136
iter  40 value 469.436941
iter  50 value 449.922819
iter  60 value 447.825623
iter  70 value 447.373527
iter  80 value 446.894881
iter  90 value 446.541407
iter 100 value 445.965617
iter 110 value 445.535616
iter 120 value 443.031956
iter 130 value 442.622567
iter 140 value 439.864562
iter 150 value 425.103713
iter 160 value 405.021717
iter 170 value 366.021552
iter 180 value 356.837027
iter 190 value 354.239817
iter 200 value 352.812151
iter 210 value 349.851303
iter 220 value 349.304307
iter 230 value 348.719780
iter 240 value 348.154263
iter 250 value 347.874725
iter 260 value 347.775561
iter 270 value 347.692716
iter 280 value 347.661454
iter 290 value 347.653778
iter 300 value 347.650352
iter 310 value 347.648677
iter 320 value 347.647893
```

```
iter 330 value 347.646407
iter 340 value 347.645900
final  value 347.645881
converged
> GroupData<-read.csv(file.choose(),header=T)
> GroupNN <- nnet(HighValueCar ~ ., data = GroupData, size = 4, maxit = 100000, decay = 0.01)
# weights:  225
initial  value 753.655316
iter  10 value 565.831266
iter  20 value 532.704136
iter  30 value 530.339970
iter  40 value 476.636097
iter  50 value 475.744667
iter  60 value 465.088071
iter  70 value 458.941738
iter  80 value 438.462661
iter  90 value 420.577864
iter 100 value 370.919985
iter 110 value 360.635485
iter 120 value 356.116797
iter 130 value 352.192101
iter 140 value 351.436817
iter 150 value 350.863324
iter 160 value 350.491425
iter 170 value 350.243808
iter 180 value 350.082753
iter 190 value 349.991247
iter 200 value 349.613560
iter 210 value 349.129997
iter 220 value 348.455380
iter 230 value 346.727563
iter 240 value 346.282809
iter 250 value 346.243720
iter 260 value 346.218451
iter 270 value 346.140538
iter 280 value 346.119992
iter 290 value 346.118633
final  value 346.118525
converged
> library(NeuralNetTools)
Warning message:
package 'NeuralNetTools' was built under R version 4.3.2
> plotnet(GroupData)
Error in UseMethod("neuralweights") :
  no applicable method for 'neuralweights' applied to an object of class "data.frame"
> plotnet(GroupNN)
> install.packages("e1071")
Installing package into 'C:/Users/15039/AppData/Local/R/win-library/4.3'
(as 'lib' is unspecified)
trying URL 'https://ftp.osuosl.org/pub/cran/bin/windows/contrib/4.3/e1071_1.7-13.zip'
Content type 'application/zip' length 653309 bytes (637 KB)
downloaded 637 KB

package 'e1071' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\15039\AppData\Local\Temp\Rtmp2VQgwT\downloaded_packages
> set.seed(123)
> sample_size <- floor(0.7 * nrow(GroupData))
> train_data <- GroupData[sample(sample_size, nrow(GroupData)), ]
Error in sample.int(x, size, replace, prob) :
  cannot take a sample larger than the population when 'replace = FALSE'
> data <- GroupData[, c("HighValueCar", "brand", "model", "model_year", "milage", "accident", )]
Error in c("HighValueCar", "brand", "model", "model_year", "milage", "accident",  :
  argument 7 is empty
> data <- GroupData[, c("HighValueCar", "brand", "model", "model_year", "milage", "accident" )]
Error in `[.data.frame`(GroupData, , c("HighValueCar", "brand", "model",  :
  undefined columns selected
> head(GroupData)
     brand model_year milage                                    accident HighValueCar
```

```
1     Ford       2013  51000 At least 1 accident or damage reported           0
2  Hyundai       2021  34742 At least 1 accident or damage reported           0
3    Lexus       2022  22372                          None reported           1
4 INFINITI       2015  88900                          None reported           0
5     Audi       2021   9835                          None reported           0
6    Acura       2016 136397                          None reported           0
> formula <- HighValueCar ~ brand + model_year + mileage + accident
> sample_size <- floor(0.7 * nrow(GroupData))
> train_data <- GroupData[sample(sample_size, nrow(GroupData)), ]
Error in sample.int(x, size, replace, prob) :
  cannot take a sample larger than the population when 'replace = FALSE'
> set.seed(123)
> sample_indices <- sample(nrow(GroupData), size = sample_size, replace = TRUE)
> train_data <- GroupData[sample_indices, ]
> test_data <- GroupData[-sample_indices, ]
> naive_bayes_model <- naiveBayes(formula, data = train_data)
Error in naiveBayes(formula, data = train_data) :
  could not find function "naiveBayes"
> library(e1071)
Warning message:
package 'e1071' was built under R version 4.3.2
> naive_bayes_model <- naiveBayes(formula, data = train_data)
Error in eval(predvars, data, env) : object 'mileage' not found
> formula <- HighValueCar ~ brand + model_year + mileage + accident
> naive_bayes_model <- naiveBayes(formula, data = train_data)
Error in eval(predvars, data, env) : object 'mileage' not found
> formula <- HighValueCar ~ brand + model_year + milage + accident
> naive_bayes_model <- naiveBayes(formula, data = train_data)
> predictions <- predict(naive_bayes_model, newdata = test_data, type = "class")
> confusion_matrix <- table(Actual = test_data$HighValueCar, Predicted = predictions)
> accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
> print(confusion_matrix)
      Predicted
Actual    0    1
     0 1188  247
     1   94  264
> cat("Accuracy:", accuracy)
Accuracy: 0.809816> summary(predictions)
   0    1
1282  511
> predictions_all <- predict(naive_bayes_model, newdata = GroupData, type = "class")
> GroupData$Predicted_HighValueCar <- predictions_all
> write.csv(GroupData, "GroupData_with_predictions.csv", row.names = FALSE)
>
>
>
> # Naive Bayes Model Evaluation:
> predictions <- predict(naive_bayes_model, newdata = test_data, type = "class")
> confusion_matrix <- table(Actual = test_data$HighValueCar, Predicted = predictions)
>
> # Calculate Precision, Recall, and F1-score
> precision <- confusion_matrix[2, 2] / sum(confusion_matrix[, 2])
> recall <- confusion_matrix[2, 2] / sum(confusion_matrix[2, ])
> f1_score <- 2 * (precision * recall) / (precision + recall)
>
> # Print the confusion matrix and additional metrics
> print(confusion_matrix)
      Predicted
Actual    0    1
     0 1188  247
     1   94  264
> cat("Accuracy:", accuracy, "\n")
Accuracy: 0.809816
> cat("Precision:", precision, "\n")
Precision: 0.5166341
> cat("Recall:", recall, "\n")
Recall: 0.7374302
> cat("F1-score:", f1_score, "\n")
F1-score: 0.6075949
>
```

```
>
>
> # Neural Network Model Evaluation:
> GroupData$HighValueCar <- as.factor(GroupData$HighValueCar)
>
> # Check for missing values
> missing_values <- sum(is.na(GroupData))
> if (missing_values > 0) {
+   cat("Dataset contains missing values. Imputation or data cleaning is required.\n")
+ } else {
+   tryCatch({
+     # Train neural network (intentionally causing an error)
+     GroupNN <- nnet(HighValueCar ~ ., data = GroupData, size = 8, maxit = 100000)
+   }, error = function(e) {
+     cat("Neural network training encountered an error. See details below:\n")
+     cat(paste("Error: ", e$message, "\n"))
+     cat("Neural network evaluation is not working on this database.\n")
+   })
+ }
# weights:  457
initial  value 3860.384267
iter  10 value 1543.115403
iter  20 value 1401.035087
iter  30 value 1323.026264
iter  40 value 1209.610501
iter  50 value 1182.301717
iter  60 value 1149.225849
iter  70 value 1131.696834
iter  80 value 1128.469771
iter  90 value 1122.750396
iter 100 value 1117.822180
iter 110 value 1117.029446
iter 120 value 1094.619875
iter 130 value 1080.141138
iter 140 value 1079.942507
iter 150 value 1079.857431
iter 160 value 1079.819492
iter 170 value 1079.760681
iter 180 value 1078.724565
iter 190 value 1078.546001
iter 200 value 1078.400268
iter 210 value 1078.211645
iter 220 value 1077.936323
iter 230 value 1077.861902
iter 240 value 1077.851547
iter 250 value 1077.799761
iter 260 value 1077.735143
iter 270 value 1077.729466
final  value 1077.727670
converged
> q()
>
```