AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Science and Technology

FINAL TERM PROJECT REPORT

Assignment Title:	Credit Card Faud DetectingDataset Project Report		
Assignment No:	1	Date of Submission:	15 August 2023
Course Title:	Introduction to Data Science		
Course Code:	01153	Section:	C
Semester:	Summer 2022-23	Course Teacher:	Abdus Salam

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaborationhas been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
7. I/we understand thatPlagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a formofcheatingandisaveryseriousacademicoffencethatmayleadtoexpulsionfromtheUniversity. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of them arterial used is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

* *Student(s) must complete all details except the faculty use part.*
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.:

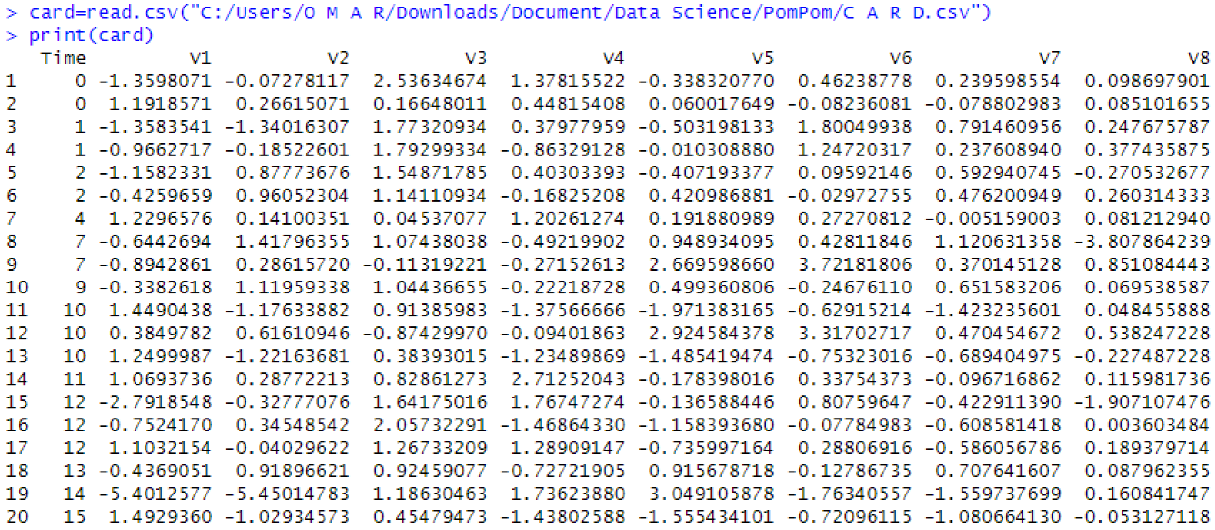
No	Name	ID	Program	Signature
1	FAISAL, MD. OMAR FARUK	20-43669-2	BSc [CSE]	
2			Choose an item.	
3			Choose an item.	
4			Choose an item.	
5			Choose an item.	
6			Choose an item.	
7			Choose an item.	
8			Choose an item.	
9			Choose an item.	
10			Choose an item.	

Faculty use only

FACULTYCOMMENTS	Marks Obtained	
	Total Marks	

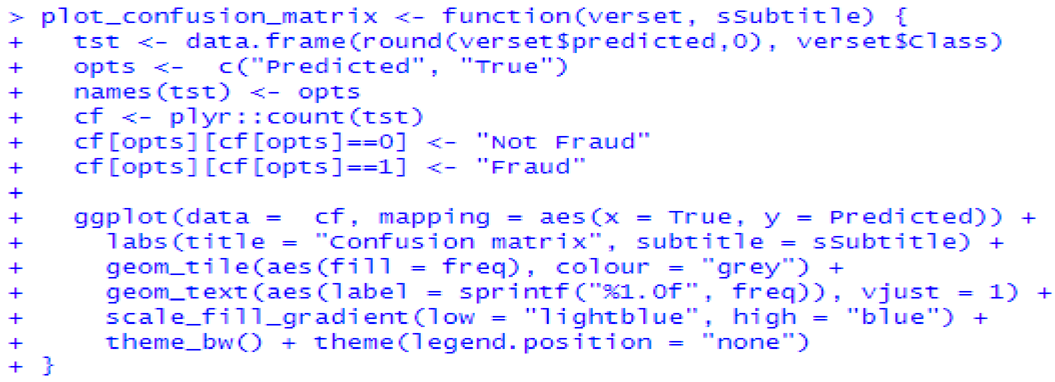
DATASET Link : https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

At first I insert the Dataset(.csv) file of Credit card fraud detection



```
> card=read.csv("C:/Users/O M A R/Downloads/Document/Data Science/PomPom/C A R D.csv")
> print(card)
   Time         V1          V2          V3          V4           V5          V6           V7           V8
1     0 -1.3598071 -0.07278117  2.53634674  1.37815522 -0.338320770  0.46238778  0.239598554  0.098697901
2     0  1.1918571  0.26615071  0.16648011  0.44815408  0.060017649 -0.08236081 -0.078802983  0.085101655
3     1 -1.3583541 -1.34016307  1.77320934  0.37977959 -0.503198133  1.80049938  0.791460956  0.247675787
4     1 -0.9662717 -0.18522601  1.79299334 -0.86329128 -0.010308880  1.24720317  0.237608940  0.377435875
5     2 -1.1582331  0.87773676  1.54871785  0.40303393 -0.407193377  0.09592146  0.592940745 -0.270532677
6     2 -0.4259659  0.96052304  1.14110934 -0.16825208  0.420986881 -0.02972755  0.476200949  0.260314333
7     4  1.2296576  0.14100351  0.04537077  1.20261274  0.191880989  0.27270812 -0.005159003  0.081212940
8     7 -0.6442694  1.41796355  1.07438038 -0.49219902  0.948934095  0.42811846  1.120631358 -3.807864239
9     7 -0.8942861  0.28615720 -0.11319221 -0.27152613  2.669598660  3.72181806  0.370145128  0.851084443
10    9 -0.3382618  1.11959338  1.04436655 -0.22218728  0.499360806 -0.24676110  0.651583206  0.069538587
11   10  1.4490438 -1.17633882  0.91385983 -1.37566666 -1.971383165 -0.62915214 -1.423235601  0.048455888
12   10  0.3849782  0.61610946 -0.87429970 -0.09401863  2.924584378  3.31702717  0.470454672  0.538247228
13   10  1.2499987 -1.22163681  0.38393015 -1.23489869 -1.485419474 -0.75323016 -0.689404975 -0.227487228
14   11  1.0693736  0.28772213  0.82861273  2.71252043 -0.178398016  0.33754373 -0.096716862  0.115981736
15   12 -2.7918548 -0.32777076  1.64175016  1.76747274 -0.136588446  0.80759647 -0.422911390 -1.907107476
16   12 -0.7524170  0.34548542  2.05732291 -1.46864330 -1.158393680 -0.07784983 -0.608581418  0.003603484
17   12  1.1032154 -0.04029622  1.26733209  1.28909147 -0.735997164  0.28806916 -0.586056786  0.189379714
18   13 -0.4369051  0.91896621  0.92459077 -0.72721905  0.915678718 -0.12786735  0.707641607  0.087962355
19   14 -5.4012577 -5.45014783  1.18630463  1.73623880  3.049105878 -1.76340557 -1.559737699  0.160841747
20   15  1.4929360 -1.02934573  0.45479473 -1.43802588 -1.555434101 -0.72096115 -1.080664130 -0.053127118
```

Confusion Matrix function



```
> plot_confusion_matrix <- function(verset, sSubtitle) {
+   tst <- data.frame(round(verset$predicted,0), verset$Class)
+   opts <-  c("Predicted", "True")
+   names(tst) <- opts
+   cf <- plyr::count(tst)
+   cf[opts][cf[opts]==0] <- "Not Fraud"
+   cf[opts][cf[opts]==1] <- "Fraud"
+
+   ggplot(data =  cf, mapping = aes(x = True, y = Predicted)) +
+     labs(title = "Confusion matrix", subtitle = sSubtitle) +
+     geom_tile(aes(fill = freq), colour = "grey") +
+     geom_text(aes(label = sprintf("%1.0f", freq)), vjust = 1) +
+     scale_fill_gradient(low = "lightblue", high = "blue") +
+     theme_bw() + theme(legend.position = "none")
+ }
```

Column & Row number

sprintf("Rows: %d Columns: %d",nrow(card), length(names(card)))

First few rows of data

head(card,10) %>%

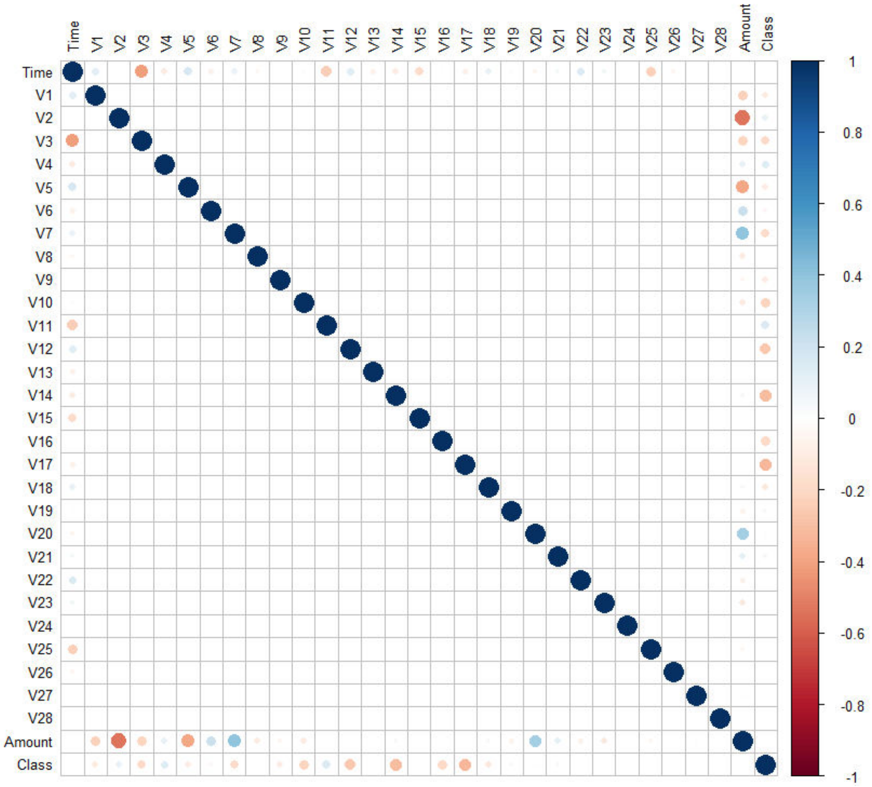
kable("html", escape=F, align="c") %>%

kable_styling(bootstrap_options = "striped", full_width = F, position = "center")

Correlations : Pearson Correlation

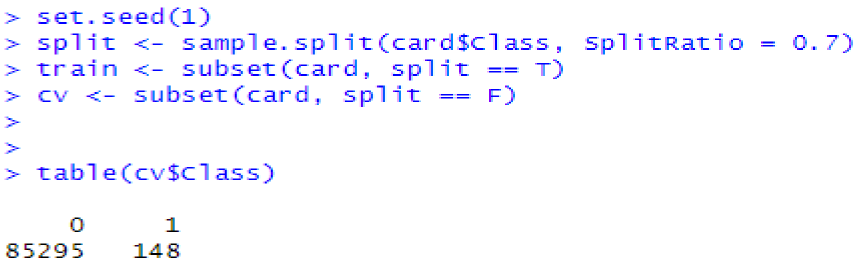
correlations <- cor(card,method="pearson")

corrplot(correlations, number.cex = .9, method = "circle", type = "full", tl.cex=0.8,tl.col = "black")



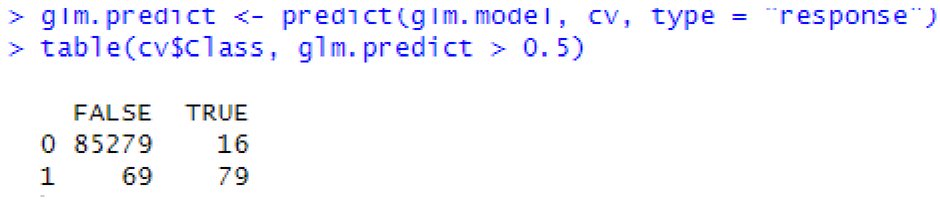
Predictive Modeling

Split data 70:30 where baseline accuracy **99.826785%**



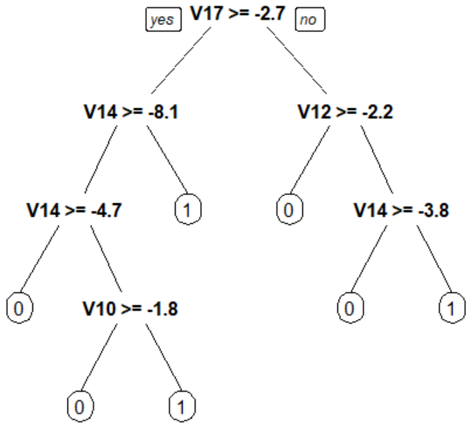
```
> set.seed(1)
> split <- sample.split(card$Class, SplitRatio = 0.7)
> train <- subset(card, split == T)
> cv <- subset(card, split == F)
>
>
> table(cv$Class)

    0     1
85295   148
```

Logistic regression (Accuracy -> 99.900518%)

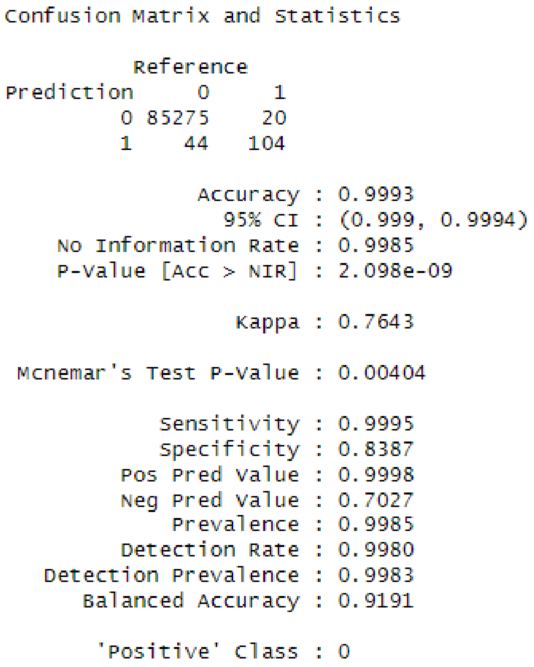


```
> glm.predict <- predict(glm.model, cv, type = "response")
> table(cv$Class, glm.predict > 0.5)

     FALSE  TRUE
  0 85279    16
  1    69    79
```

Decision Tree Model

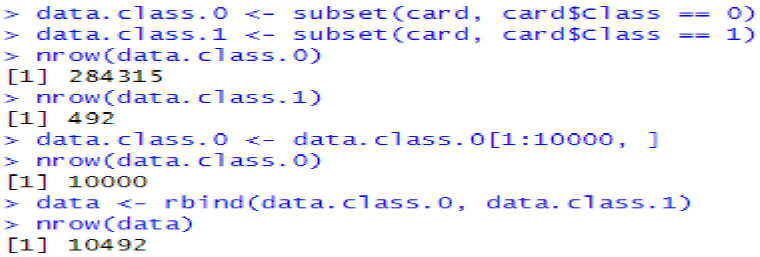


Confusion Matrix & Statistics -> 99.925096 % accuracy (best) using decision tree



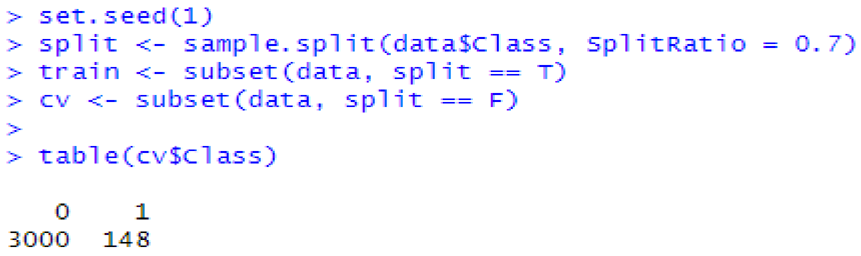
```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 85275    20
         1    44   104
                                          
               Accuracy : 0.9993          
                 95% CI : (0.999, 0.9994)
    No Information Rate : 0.9985          
    P-Value [Acc > NIR] : 2.098e-09       
                                          
                  Kappa : 0.7643          
                                          
 Mcnemar's Test P-Value : 0.00404         
                                          
            Sensitivity : 0.9995          
            Specificity : 0.8387          
         Pos Pred Value : 0.9998          
         Neg Pred Value : 0.7027          
             Prevalence : 0.9985          
         Detection Rate : 0.9980          
   Detection Prevalence : 0.9983          
      Balanced Accuracy : 0.9191          
                                          
       'Positive' Class : 0               
```

Now we only keep 10000 rows of data with class = 0



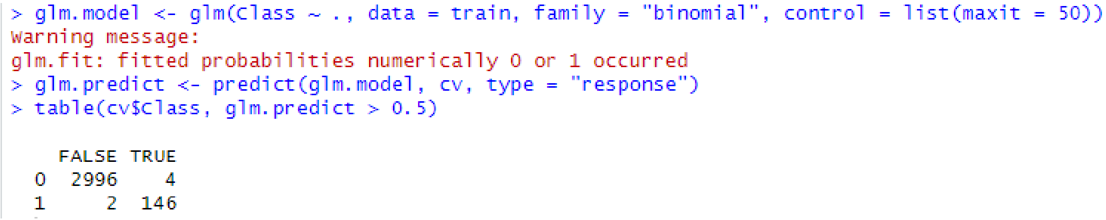
```
> data.class.0 <- subset(card, card$Class == 0)
> data.class.1 <- subset(card, card$Class == 1)
> nrow(data.class.0)
[1] 284315
> nrow(data.class.1)
[1] 492
> data.class.0 <- data.class.0[1:10000, ]
> nrow(data.class.0)
[1] 10000
> data <- rbind(data.class.0, data.class.1)
> nrow(data)
[1] 10492
```

Split Data 70:30 (Baseline accuracy -> 95.298602%)



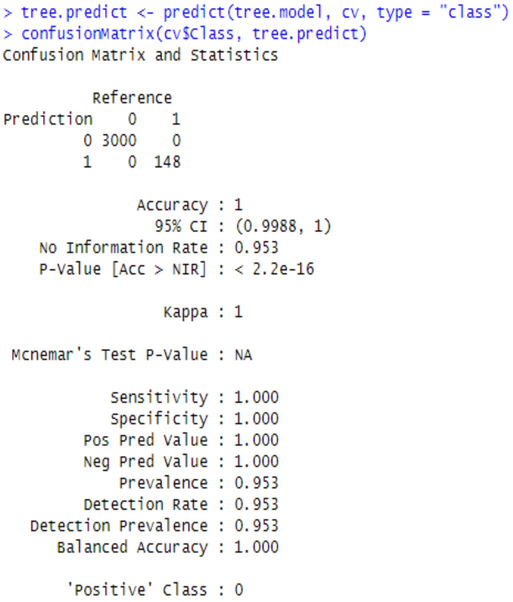
```
> set.seed(1)
> split <- sample.split(data$Class, SplitRatio = 0.7)
> train <- subset(data, split == T)
> cv <- subset(data, split == F)
>
> table(cv$Class)

   0    1 
3000  148 
```

Logistic regression (Accuracy 99.809402%)



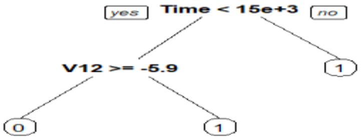
```
> glm.model <- glm(Class ~ ., data = train, family = "binomial", control = list(maxit = 50))
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred
> glm.predict <- predict(glm.model, cv, type = "response")
> table(cv$Class, glm.predict > 0.5)

    FALSE TRUE
  0  2996    4
  1     2  146
```

SVM Model with accuracy of 98.856416%

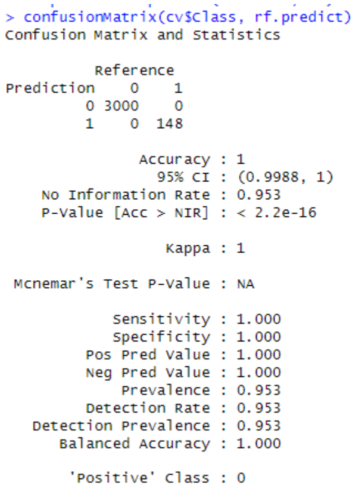


```
> tree.predict <- predict(tree.model, cv, type = "class")
> confusionMatrix(cv$Class, tree.predict)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 3000    0
         1    0  148
                                     
               Accuracy : 1          
                 95% CI : (0.9988, 1)
    No Information Rate : 0.953      
    P-Value [Acc > NIR] : < 2.2e-16  
                                     
                  Kappa : 1          
                                     
 Mcnemar's Test P-Value : NA         
                                     
            Sensitivity : 1.000      
            Specificity : 1.000      
         Pos Pred Value : 1.000      
         Neg Pred Value : 1.000      
             Prevalence : 0.953      
         Detection Rate : 0.953      
   Detection Prevalence : 0.953      
      Balanced Accuracy : 1.000      
                                     
       'Positive' Class : 0          
```

Decision Tree Model



Confusion Matrix and Statistics



```
> confusionMatrix(cv$Class, rf.predict)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 3000    0
         1    0  148
                                    
               Accuracy : 1         
                 95% CI : (0.9988, 1)
    No Information Rate : 0.953     
    P-Value [Acc > NIR] : < 2.2e-16 
                                    
                  Kappa : 1         
                                    
 Mcnemar's Test P-Value : NA        
                                    
            Sensitivity : 1.000     
            Specificity : 1.000     
         Pos Pred Value : 1.000     
         Neg Pred Value : 1.000     
             Prevalence : 0.953     
         Detection Rate : 0.953     
   Detection Prevalence : 0.953     
      Balanced Accuracy : 1.000     
                                    
       'Positive' Class : 0         
```

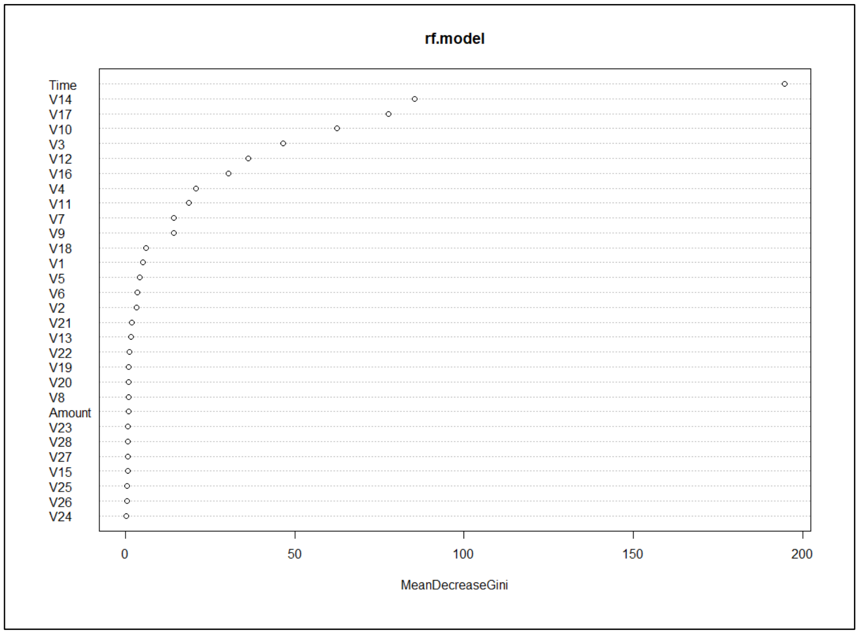
Random Forest(rf) model

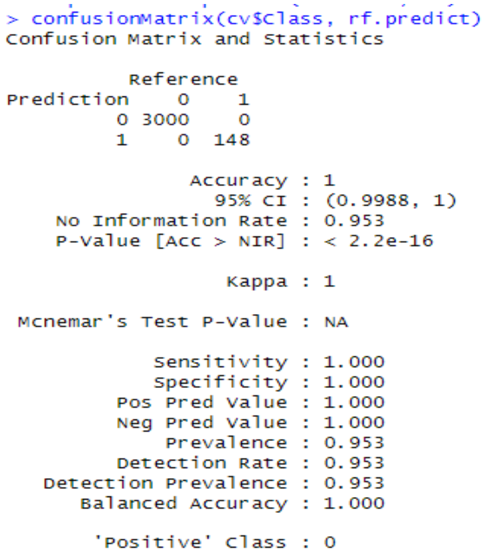
set.seed(10)

rf.model <- randomForest(Class ~ ., data = train,ntree = 2000, nodesize = 20)

rf.predict <- predict(rf.model, cv)

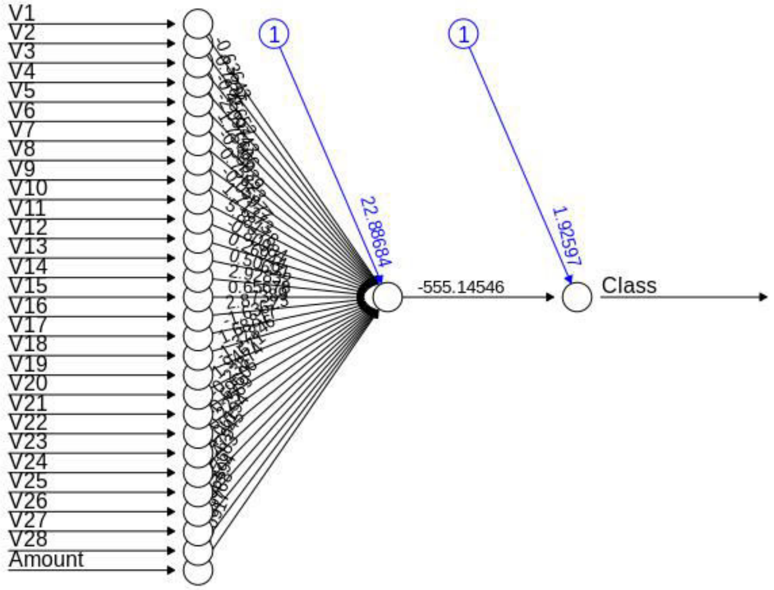
confusionMatrix(cv$Class, rf.predict)





```
> confusionMatrix(cv$Class, rf.predict)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 3000    0
         1    0  148
                                          
               Accuracy : 1               
                 95% CI : (0.9988, 1)     
    No Information Rate : 0.953           
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 1               
                                          
 Mcnemar's Test P-Value : NA              
                                          
            Sensitivity : 1.000           
            Specificity : 1.000           
         Pos Pred Value : 1.000           
         Neg Pred Value : 1.000           
             Prevalence : 0.953           
         Detection Rate : 0.953           
   Detection Prevalence : 0.953           
      Balanced Accuracy : 1.000           
                                          
       'Positive' Class : 0               
```

ANN Model (Artificial Neural Nertwok)



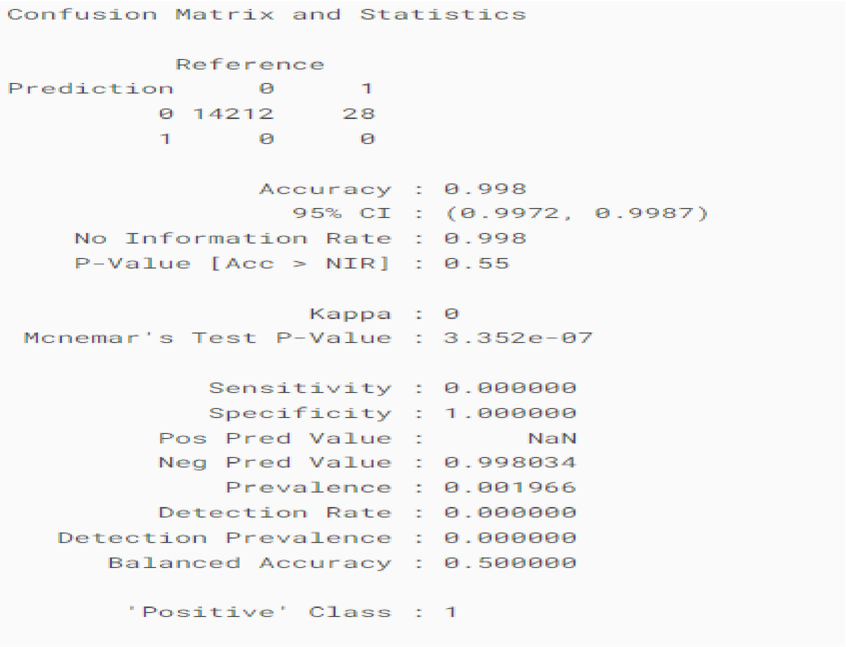
“Class” is of a class “Integer” , the factor transformationwas performed:

card$Class <- factor(card$Class)

K-Nearest Neighbours

set.seed(1998)

knn1 <- knn(train = train[,-31], test = test[,-31], cl = train$Class, k = 5) confusionMatrix(knn1, test$Class, positive = "1")

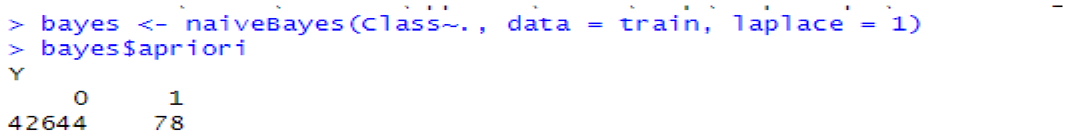


```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 14212    28
         1     0     0

               Accuracy : 0.998
                 95% CI : (0.9972, 0.9987)
    No Information Rate : 0.998
    P-Value [Acc > NIR] : 0.55

                  Kappa : 0
 Mcnemar's Test P-Value : 3.352e-07

            Sensitivity : 0.000000
            Specificity : 1.000000
         Pos Pred Value :      NaN
         Neg Pred Value : 0.998034
             Prevalence : 0.001966
         Detection Rate : 0.000000
   Detection Prevalence : 0.000000
      Balanced Accuracy : 0.500000

       'Positive' Class : 1
```

Naive Bayes

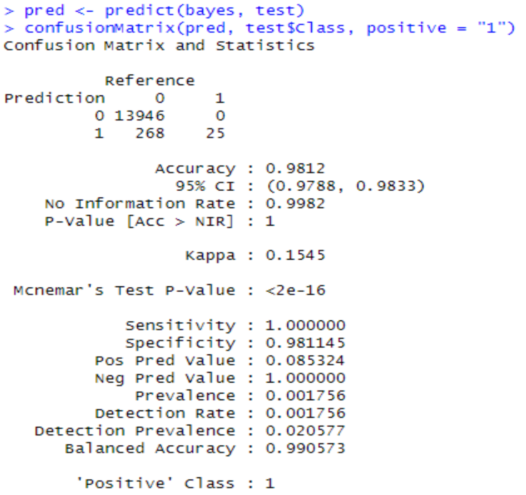
bayes <- naiveBayes(Class~., data = train, laplace = 1)

bayes$apriori



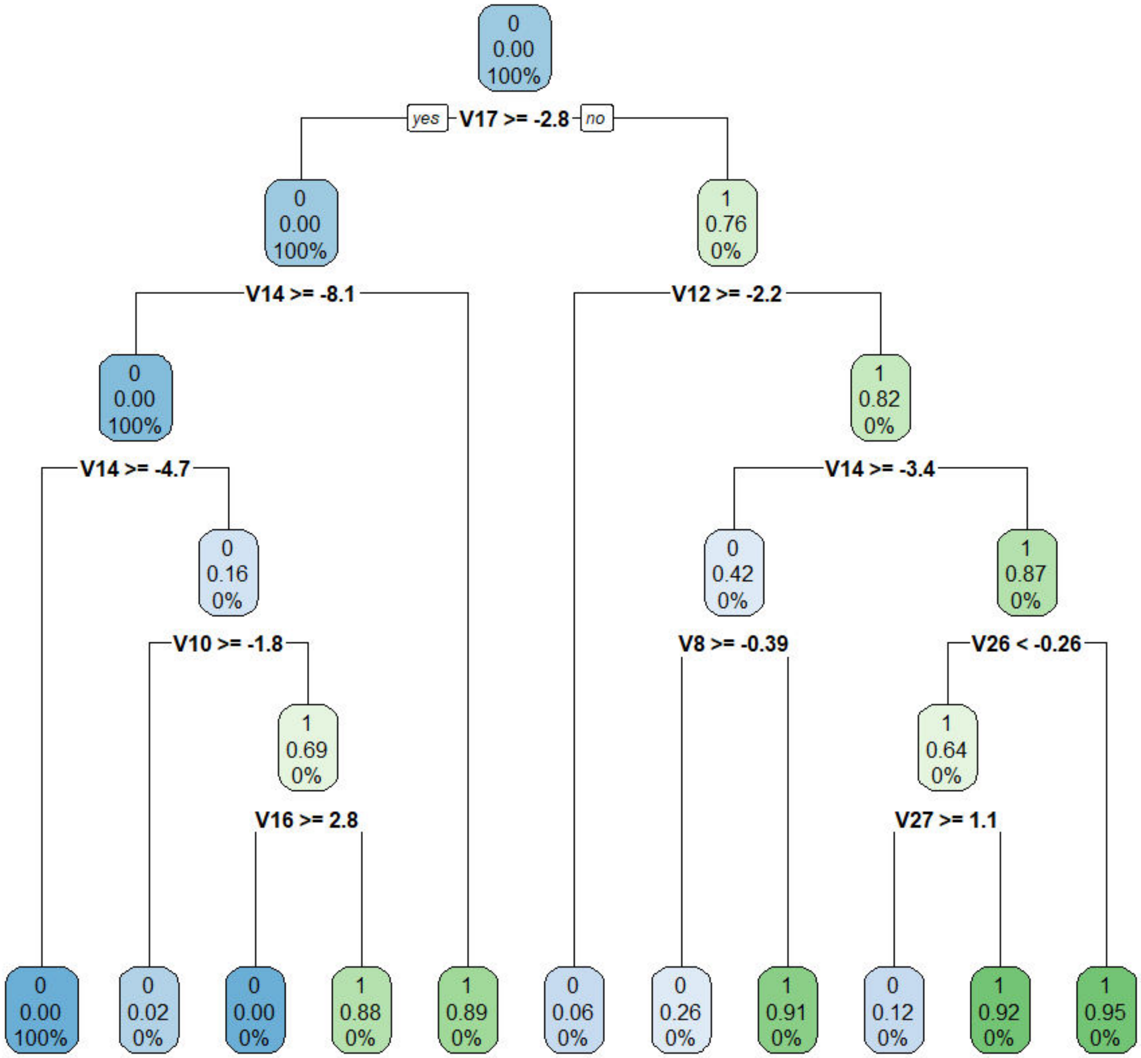
```
> bayes <- naiveBayes(Class~., data = train, laplace = 1)
> bayes$apriori
Y
    0     1 
42644    78 
```

Confusion Matrix



```
> pred <- predict(bayes, test)
> confusionMatrix(pred, test$Class, positive = "1")
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 13946     0
         1   268    25
                                          
               Accuracy : 0.9812          
                 95% CI : (0.9788, 0.9833)
    No Information Rate : 0.9982          
    P-Value [Acc > NIR] : 1               
                                          
                  Kappa : 0.1545          
                                          
 Mcnemar's Test P-Value : <2e-16          
                                          
            Sensitivity : 1.000000        
            Specificity : 0.981145        
         Pos Pred Value : 0.085324        
         Neg Pred Value : 1.000000        
             Prevalence : 0.001756        
         Detection Rate : 0.001756        
   Detection Prevalence : 0.020577        
      Balanced Accuracy : 0.990573        
                                          
       'Positive' Class : 1               
```

Overall Decison Tree


0
0.00
100%
yes
V17 >= -2.8
no
0
0.00
100%
1
0.76
0%
V14 >= -8.1
V12 >= -2.2
0
0.00
100%
1
0.82
0%
V14 >= -4.7
V14 >= -3.4
0
0.16
0%
0
0.42
0%
1
0.87
0%
V10 >= -1.8
V8 >= -0.39
V26 < -0.26
1
0.69
0%
1
0.64
0%
V16 >= 2.8
V27 >= 1.1
0
0.00
100%
0
0.02
0%
0
0.00
0%
1
0.88
0%
1
0.89
0%
0
0.06
0%
0
0.26
0%
1
0.91
0%
0
0.12
0%
1
0.92
0%
1
0.95
0%