Credit Card Fraud Detection

Abstract:

This project implemented in R language analyzes a highly unbalanced credit card transaction dataset, containing 284,807 transactions made by European cardholders in September 2013. The dataset has 492 frauds, accounting for only 0.172% of all transactions. The input variables are numerical and result from a PCA transformation, except for 'Time' and 'Amount' features. 'Time' records the seconds elapsed between each transaction and the first transaction in the dataset, while 'Amount' records the transaction amount. The response variable 'Class' indicates the presence of fraud with 1, while 0 indicates otherwise.

We are implementing various machine learning algorithms like Random Forest, Decision Tree using C50, Decision Tree using rpart, XGBoost, KNN, KNN Smote, Gradient Boosted Tree model, SVM Radical, Neural Network, Neural Network with Class weight, SVM, SVM Smote. We are analyzing the performance of these algorithms Given the class imbalance ratio, the Area Under the Precision-Recall Curve (AUPRC) is recommended to measure accuracy instead of a confusion matrix.

Problem definition & goals:

The purpose of credit card fraud detection is to identify fraudulent transactions and prevent financial losses to both credit card companies and their customers. The problem involves developing a machine learning model that can accurately identify fraudulent transactions by analyzing various features of the transaction.

The dataset used for credit card fraud detection typically contains a large number of credit card transactions, both genuine and fraudulent. The dataset usually includes various features of the transaction such as transaction amount, location, time, and other relevant details. In addition, the dataset may also contain features that have been engineered to improve the accuracy of the machine learning model.

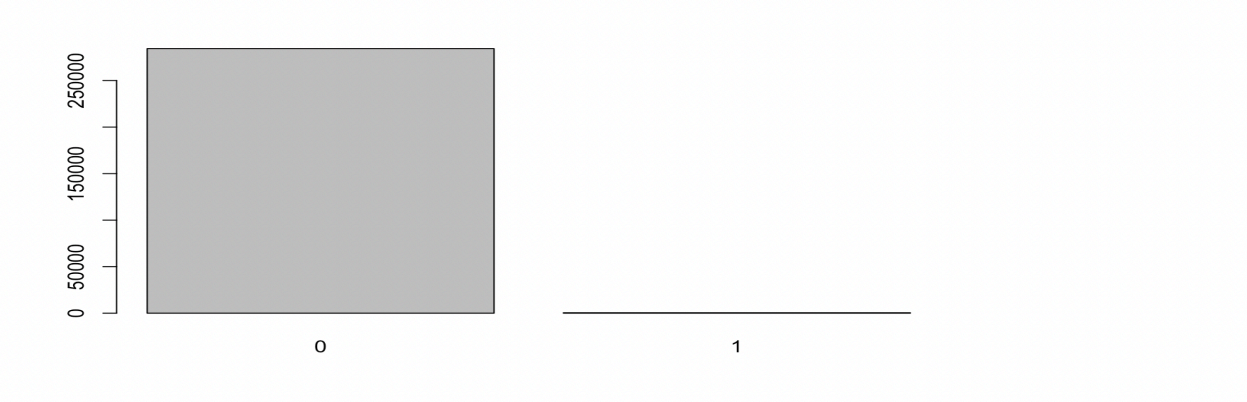
The features in the dataset are as follows:

Time: The number of seconds elapsed between each transaction and the first transaction in the dataset.
V1-V28: These are anonymized features that result from a principal component analysis (PCA) transformation applied to the original dataset. Due to confidentiality reasons, the original features could not be provided.
Amount: The transaction amount.
Class: The target variable, with a value of 1 indicating a fraudulent transaction and 0 indicating a non-fraudulent transaction.
The Time and Amount features are numeric, while the V1-V28 features are also numeric but have been transformed through PCA. The Class feature is binary, with values of 0 or 1
During data exploration, we found that the Time and Amount features had a wide range of values, and we normalized them to make them comparable with the
PCA-transformed features. We also noticed that the V1-V28 features had a near-normal distribution, while the Class feature was heavily imbalanced, with only 0.17% of transactions being fraudulent.

The primary goal of using this data is to develop a machine learning model that can accurately identify fraudulent transactions and minimize financial losses due to fraud. To achieve this, the data can be preprocessed and explored to gain insights into the underlying patterns and characteristics of fraudulent transactions. Machine learning algorithms can then be trained on this data to predict the likelihood of fraud in future transactions. Finally, the performance of the machine learning model can be evaluated and refined to improve its accuracy and effectiveness.

Data Exploration and Preprocessing Techniques:

In the credit card fraud detection project, the first step was to explore the data and understand the relationship between different variables. The dataset was visualized using appropriate plots such as histograms, box plots, and scatter plots to detect possible correlations between different features and the outcome variable (fraudulent transactions).

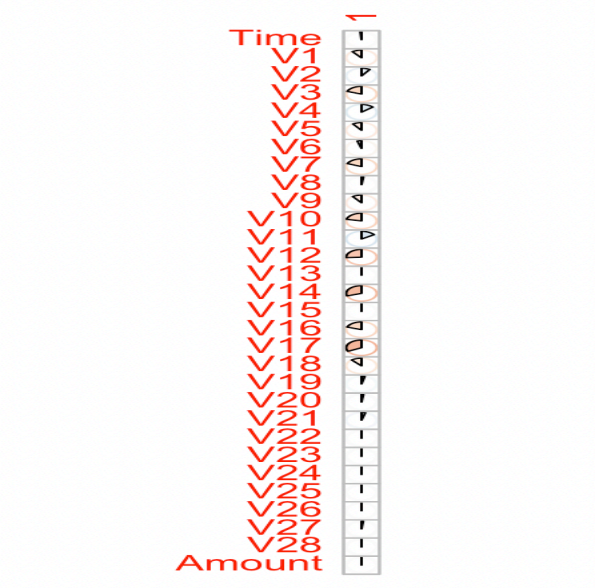


IMG: Fraud vs non Fraudulent transactions

The data cleaning and preprocessing steps included removing duplicate records, dealing with missing values, and removing irrelevant features. The missing values were handled by either removing the records or filling them with appropriate values such as mean, median, or mode. The feature engineering step involved creating new features from existing ones to improve the predictive power of the model.

But no missing values are reported in the data frame.

The variables used in the final model were selected based on their correlation with the outcome variable and their significance in predicting fraudulent transactions. The final set of features included V3, V4, V7, V10, V12, V14, V16, V17.



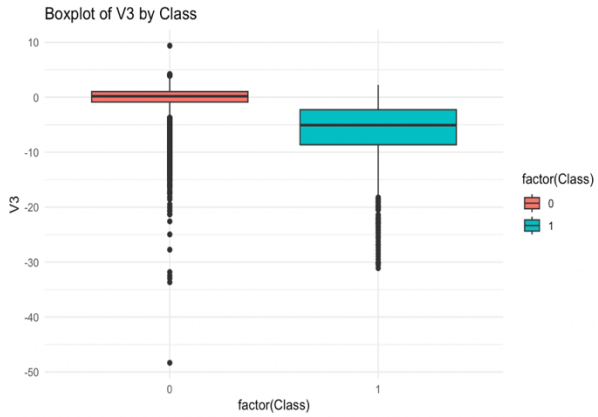
The data was imbalanced, with only a small fraction of transactions being fraudulent. To address this imbalance, we used techniques such as oversampling of the minority class, under sampling of the majority class, and the use of weighted models.

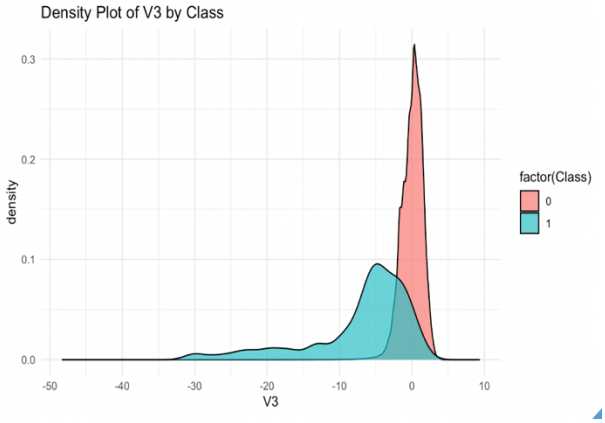
#Under Sampling and Over Sampling of data

```{r}
credit_card_0 <- credit_card[credit_card$Class == 0, ]
credit_card_1 <- credit_card[credit_card$Class == 1, ]

#Undersampling
credit_card_0_down <- credit_card_0[sample(nrow(credit_card_0), 50000), ]

# Oversampling
credit_card_1_over <- credit_card_1[sample(nrow(credit_card_1), 50000, replace = TRUE), ]

#Combining
credit_card <- rbind(credit_card_0_down, credit_card_1_over)

credit_card
```

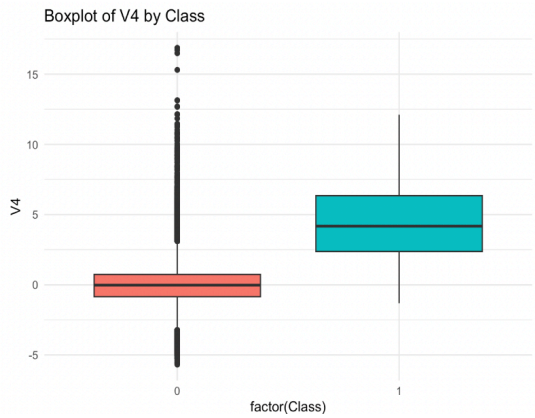
Box plot and Density plot for each numerical feature in the credit card dataset, split by the target variable Class. This is a useful way to visualize the distribution of each numeric feature in the dataset and how it varies across the two classes.

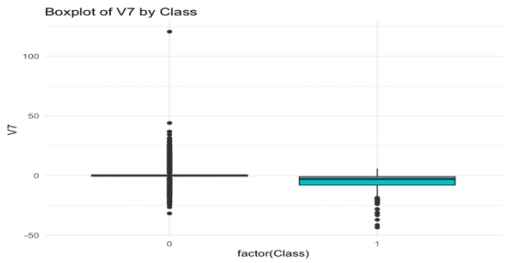
Box and Density Plot V3 by Class

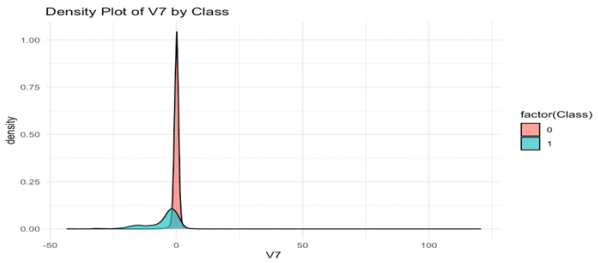


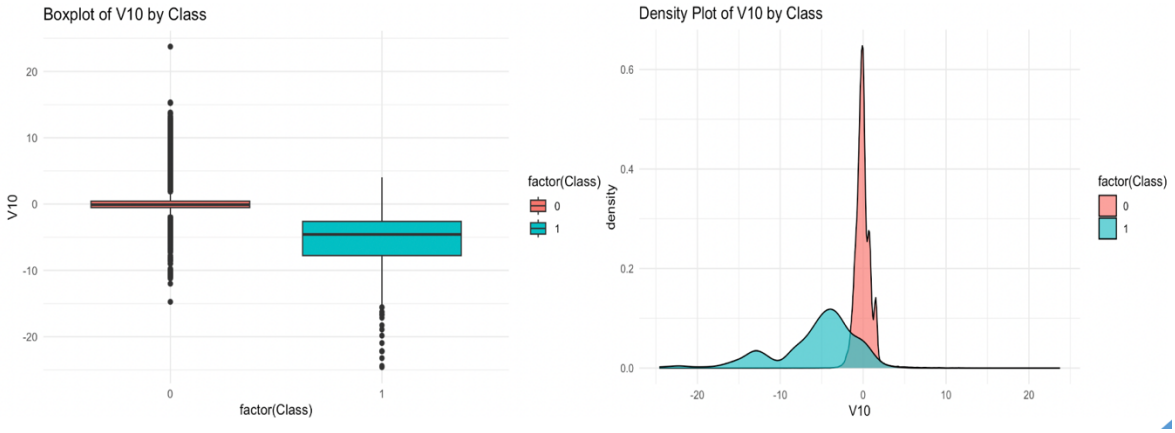


Box and Density Plot V4 by Class

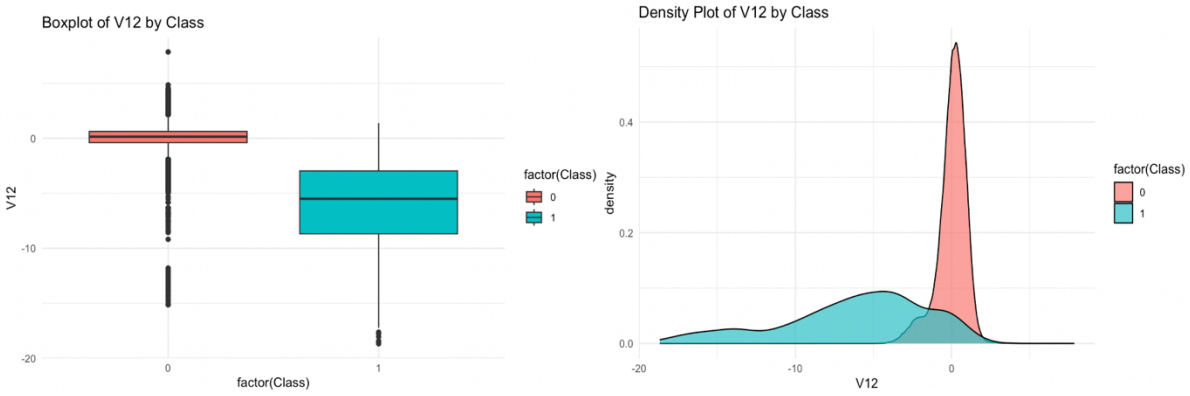




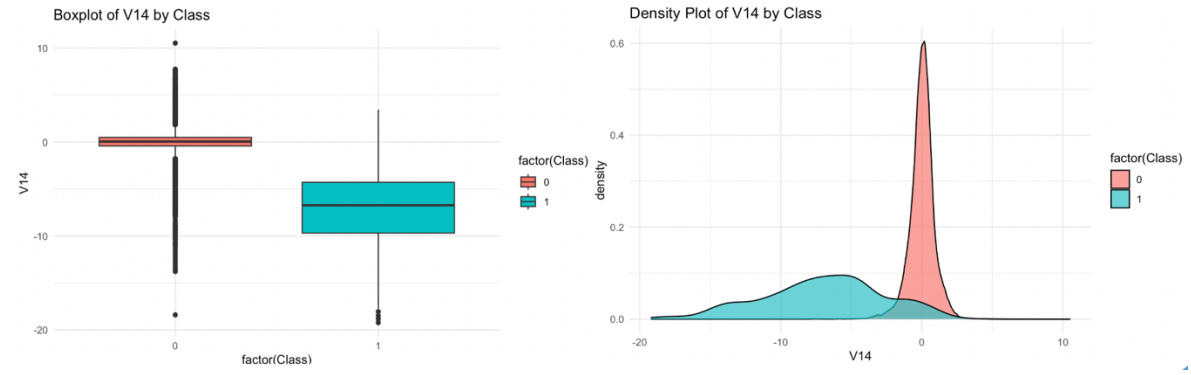




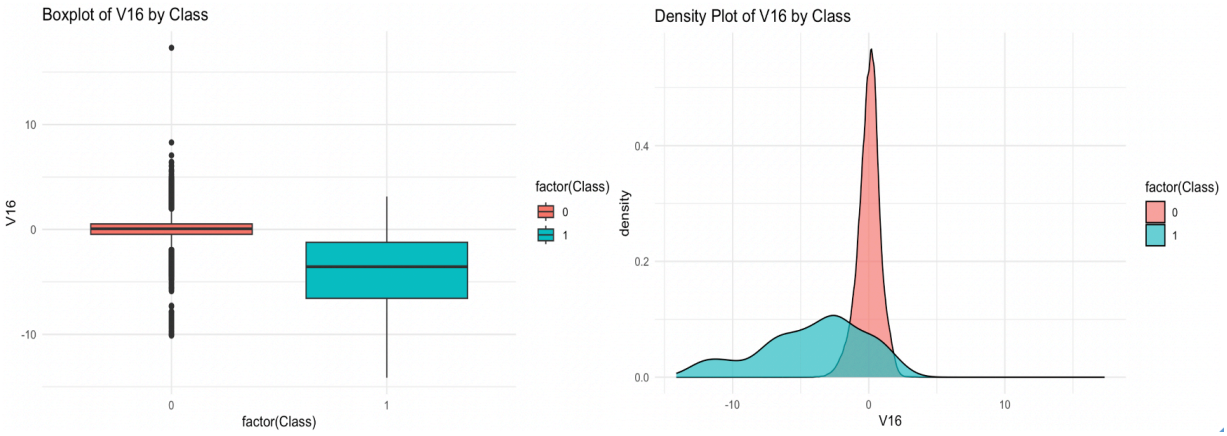
Box and Density Plot V12 by Class



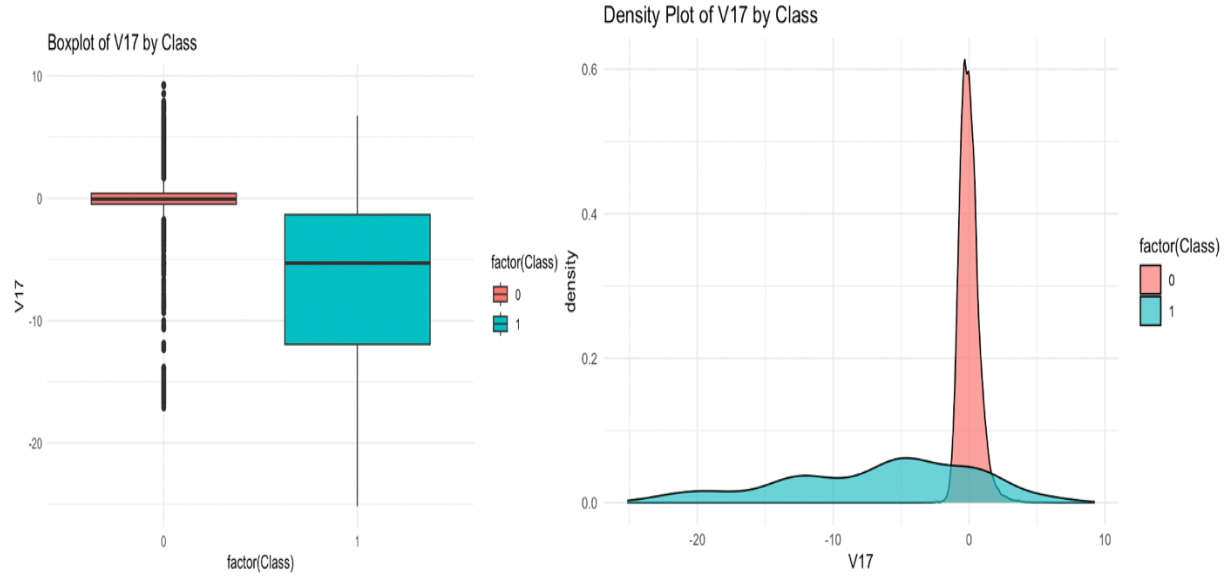
Box and Density Plot V14 by Class



Box and Density Plot V16 by Class



Box and Density Plot V17 by Class



T-test for all variables with Class:

t-test between variables by Class

```
Variable: V3
t-test p-value: 4.78608140742888e-75

Variable: V4
t-test p-value: 4.58731692138804e-136

Variable: V7
t-test p-value: 4.29223131944152e-52

Variable: V10
t-test p-value: 3.38288365061388e-93

Variable: V12
t-test p-value: 1.48131650283816e-112

Variable: V14
t-test p-value: 1.0401507099026e-140

Variable: V16
t-test p-value: 7.90532563943685e-84

Variable: V17
t-test p-value: 1.55809380578438e-71
```

Experimental Results:

Trained several Machine learning model to classify weather a transaction fraudulent or not.

1. **Knn and Knn Smote**

Knn and Knn Smote

```
A performance instance
  'Area under the ROC curve'
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  14817     0
       Yes   183 15000

               Accuracy : 0.9939
                 95% CI : (0.993, 0.9947)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9878

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 1.0000
            Specificity : 0.9878
         Pos Pred Value : 0.9879
         Neg Pred Value : 1.0000
              Precision : 0.9879
                 Recall : 1.0000
                     F1 : 0.9939
             Prevalence : 0.5000
         Detection Rate : 0.5000
   Detection Prevalence : 0.5061
      Balanced Accuracy : 0.9939

       'Positive' Class : Yes
```

```
A performance instance
  'Area under the ROC curve'
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  14817     0
       Yes   183 15000

               Accuracy : 0.9939
                 95% CI : (0.993, 0.9947)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9878

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 1.0000
            Specificity : 0.9878
         Pos Pred Value : 0.9879
         Neg Pred Value : 1.0000
              Precision : 0.9879
                 Recall : 1.0000
                     F1 : 0.9939
             Prevalence : 0.5000
         Detection Rate : 0.5000
   Detection Prevalence : 0.5061
      Balanced Accuracy : 0.9939

       'Positive' Class : Yes
```

Both Knn and Knn Smote Models returned accuracy of 99.39

2. SVM- Linear and SVM- Linear with Smote:

SVM Linear and SVM Linear with Smote

```
               Accuracy : 0.9401
                 95% CI : (0.9374, 0.9428)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8803

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9039
            Specificity : 0.9763
         Pos Pred Value : 0.9745
         Neg Pred Value : 0.9104
              Precision : 0.9745
                 Recall : 0.9039
                     F1 : 0.9379
             Prevalence : 0.5000
         Detection Rate : 0.4520
   Detection Prevalence : 0.4638
      Balanced Accuracy : 0.9401

       'Positive' Class : Yes
```

```
No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 63000, 63000, 63000, 63000, 63000, 63000, ...
Additional sampling using SMOTE

Resampling results:

  ROC        Sens       Spec
  0.9801161  0.9760571  0.9099429

Tuning parameter 'C' was held constant at a value of 1
A performance instance
  'Area under the ROC curve'
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  14645  1441
       Yes   355 13559

               Accuracy : 0.9401
                 95% CI : (0.9374, 0.9428)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8803

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9039
            Specificity : 0.9763
         Pos Pred Value : 0.9745
         Neg Pred Value : 0.9104
              Precision : 0.9745
                 Recall : 0.9039
                     F1 : 0.9379
             Prevalence : 0.5000
         Detection Rate : 0.4520
   Detection Prevalence : 0.4638
      Balanced Accuracy : 0.9401

       'Positive' Class : Yes
```

SVM Linear and with Smote Models returned accuracy of 94.21

3. SVM- Radial

SVM Radical

```
70000 samples
    8 predictor
    2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 63000, 63000, 63000, 63000, 63000, 63000, ...
Resampling results across tuning parameters:

  C     Accuracy   Kappa
  0.25  0.9581714  0.9163429
  0.50  0.9635143  0.9270286
  1.00  0.9685143  0.9370286

Tuning parameter 'sigma' was held constant at a value of 1.030363
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were sigma = 1.030363 and C = 1.
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  14840   785
       Yes   160 14215

               Accuracy : 0.9685
                 95% CI : (0.9665, 0.9704)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.937

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9893
            Specificity : 0.9477
         Pos Pred Value : 0.9498
         Neg Pred Value : 0.9889
             Prevalence : 0.5000
         Detection Rate : 0.4947
   Detection Prevalence : 0.5208
      Balanced Accuracy : 0.9685

       'Positive' Class : No
```

SVM Radical returned accuracy of 96.85

4. **Random Forest**

Random Forest

```
Random Forest

70000 samples
    8 predictor
    2 classes: 'No', 'Yes'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 63000, 63000, 63000, 63000, 63000, 63000, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  2     0.9997714  0.9995429
  5     0.9997000  0.9994000
  8     0.9989000  0.9978000

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
Confusion Matrix and Statistics

          Reference
Prediction    No    Yes
       No  14996     0
       Yes     4 15000

               Accuracy : 0.9999
                 95% CI : (0.9997, 1)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9997

 Mcnemar's Test P-Value : 0.1336

            Sensitivity : 0.9997
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9997
             Prevalence : 0.5000
         Detection Rate : 0.4999
   Detection Prevalence : 0.4999
      Balanced Accuracy : 0.9999
```

Random Forest returned accuracy of 96.85

5. **Gradient Boosted Model**

Gradient Boosted model

```
Confusion Matrix and Statistics

          Reference
Prediction     No    Yes
        No   14810    716
        Yes    190 14284

               Accuracy : 0.9698
                 95% CI : (0.9678, 0.9717)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9396

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9873
            Specificity : 0.9523
         Pos Pred Value : 0.9539
         Neg Pred Value : 0.9869
             Prevalence : 0.5000
         Detection Rate : 0.4937
   Detection Prevalence : 0.5175
      Balanced Accuracy : 0.9698

       'Positive' Class : No
```

GB Tree Model returned accuracy of 96.98

6. Extreme Gradient Boosted Model

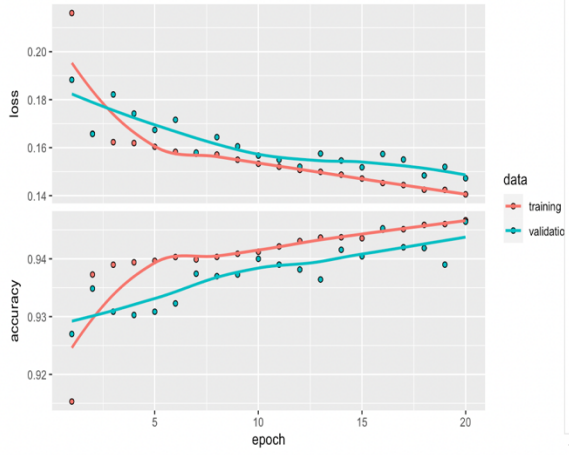
eXtreme Gradient Boosted model

```
 0.4 3         0.8              1.00      150      0.9992429 0.9984857

Tuning parameter 'gamma' was held constant at a value of 0
Tuning parameter 'min_child_weight' was held constant at
 a value of 1
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were nrounds = 150, max_depth = 3, eta = 0.4, gamma = 0, colsample_bytree =
 0.8, min_child_weight = 1 and subsample = 1.
Confusion Matrix and Statistics

          Reference
Prediction    No   Yes
       No  14972     0
       Yes    28 15000

               Accuracy : 0.9991
                 95% CI : (0.9987, 0.9994)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9981

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

            Sensitivity : 0.9981
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.9981
             Prevalence : 0.5000
         Detection Rate : 0.4991
   Detection Prevalence : 0.4991
      Balanced Accuracy : 0.9991

       'Positive' Class : No
```

XGB Tree Model returned accuracy of 99.91

7. Neural Network Model:

Neural Network Model

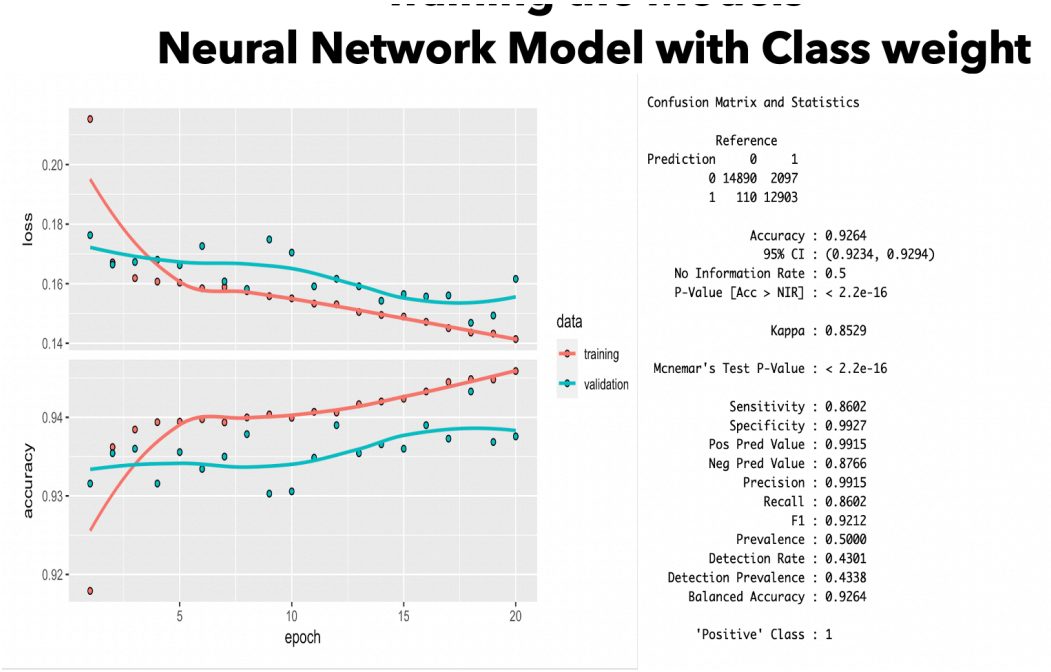



```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 13556  1347
         1  1444 13653

               Accuracy : 0.907
                 95% CI : (0.9036, 0.9102)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.8139

 Mcnemar's Test P-Value : 0.06919

            Sensitivity : 0.9102
            Specificity : 0.9037
         Pos Pred Value : 0.9044
         Neg Pred Value : 0.9096
              Precision : 0.9044
                 Recall : 0.9102
                     F1 : 0.9073
             Prevalence : 0.5000
         Detection Rate : 0.4551
   Detection Prevalence : 0.5032
      Balanced Accuracy : 0.9070

       'Positive' Class : 1
```

The Accuracy of Neural Network Model is 90.7

Neural Network Model returned accuracy of 90.7

8. NEURAL NETWORK MODEL WITH CLASS WEIGHT



Neural Network Model returned accuracy of 92.64

Conclusion:

On performing various analysis and training various models we conclude that Random Forest performed the best with 99.99% accuracy.

```
#Summary of Accuracy
Random Forest: 99.99%
DecisionTreeModel using C50: 99.97%
XGBoost: 99.93797%
KNN Smote: 99.38%
KNN: 99.38%
Gradient Boosted Tree model: 97.17%
SVM Radical: 97.04%
NeuralNetwork with class weight: 92.64%
DecisionTreeModel using rpart: 93.50195%
SVM Smote: 94.21%
SVM: 94.21%
Neural Network Model: 90.7%
```

References

•Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson and Gianluca Bontempi. Calibrating Probability with Undersampling for Unbalanced Classification. In Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, 2015

•Dal Pozzolo, Andrea; Caelen, Olivier; Le Borgne, Yann-Ael; Waterschoot, Serge; Bontempi, Gianluca. Learned lessons in credit card fraud detection from a practitioner perspective, Expert systems with applications,41,10,4915-4928,2014, Pergamon

•Dal Pozzolo, Andrea; Boracchi, Giacomo; Caelen, Olivier; Alippi, Cesare; Bontempi, Gianluca. Credit card fraud detection: a realistic modeling and a novel learning strategy, IEEE transactions on neural networks and learning systems,29,8,3784-3797,2018,IEEE

Dal Pozzolo, Andrea Adaptive Machine learning for credit card fraud detection ULB MLG PhD thesis (supervised by G. Bontempi)

Websites and Dataset Links:

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

Dataset link: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/download?datasetVersionNumber=3