1. INTRODUCTION

The purpose of the report is explaining how machine learning techniques are applied to the dataset. Machine learning algorithms are used to make prediction for target feature. Three machine learning techniques were use which are Decision Tree, Naive Bayes and K Nearst Neighbor.

1.1 Getting Familiar into Dataset

The dataset consists of 5 numerical and 9 categorical attributes. Raw form of the dataset has 1059 instances. Target feature is “banking_crisis” and it has two separate values as “crisis” and “no crisis”. Other features are explained below.

The “case”, “cc3”, and “country” columns have properties that define countries. The year column represents the observations of the countries indicated by years. For example, was the country independent on that line in the given year or was there a crisis? It answers these questions. In case the deterioration in the financial system prevents the markets to work, the systemic crisis is called as the shock of the national economy. The “systemic_crisis” column is a categorical value that determines whether this exists. The “exch_usd” column shows the value of that country's currency against the US dollar.The “domestic_debt_in_default” column indicates whether the country specified in that year has domestic debt, whether it owes its own currency in its own currency.Likewise, the “sovereign_external_debt_default” column represents the existence of the external debt of these countries. The “gdp_weighted_default” represent the total debt in default vis-a-vis the GDP. The “inflation_annual_cpi” column shows the inflation rate of the CPI. If this value is high, the purchasing power of the people has fallen and it is obvious that this value affects the target feature.The “independence” column is a categorical value that indicates whether that country is dependent on any other country.The “currency_crises” column answers the question of whether there is a sudden increase their own money in the currencies of countries.If one of the columns is evaluated, Algeria has a systemic crisis in 1870 and the exchange rate is 0.0522, with no internal or external debt. Although the inflation value is 3.44 and there is no inflation crisis, there is a bank crisis.

1.2 Exploratory Data Analysis

Before cleaning the data phase, the raw dataset has to be examined. Firstly, you can see that in the plot below how many missing values for each descriptive features.

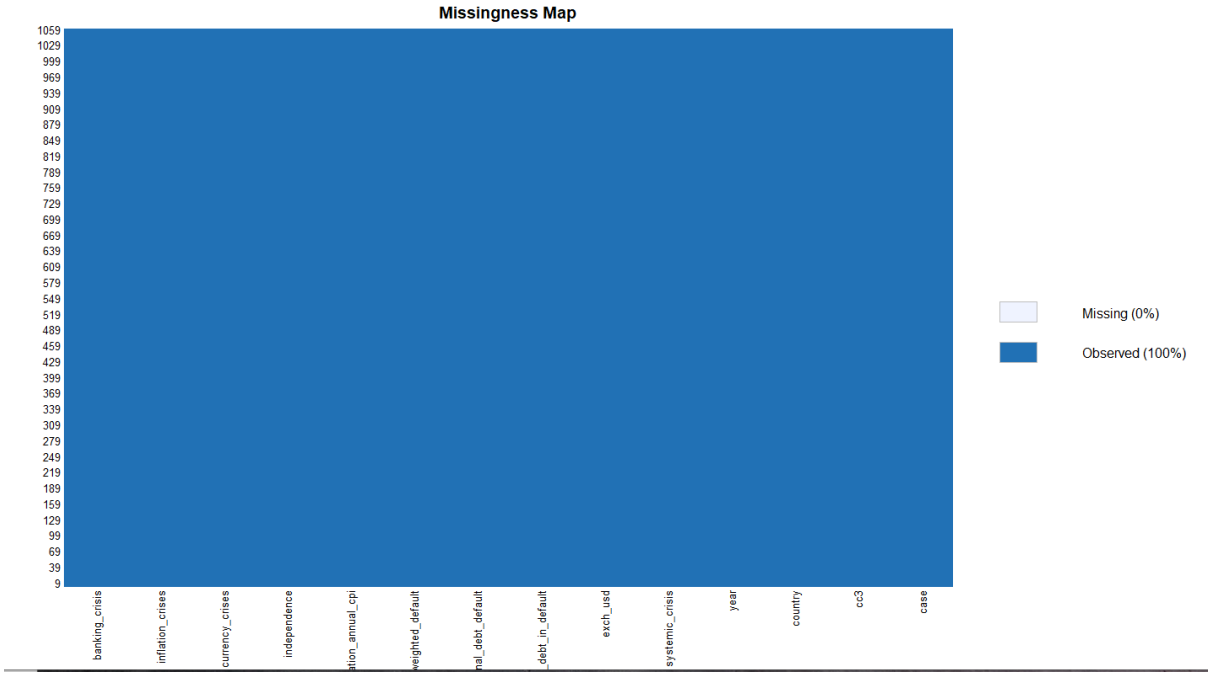

Figure 1 Data Analysis plot

There is no missing values of the raw dataset can be observed in this plot. In the second step, frequency distribution for each descriptive features are plotted in order to proper feature selection.

1.2.1 Data Visualization

In order to getting intuitive ideas about dataset, features should be plotted. This section includes many types of plots.

1.2.1.1 Ratio of Descriptive Features

First plot shows the count of values for each features.

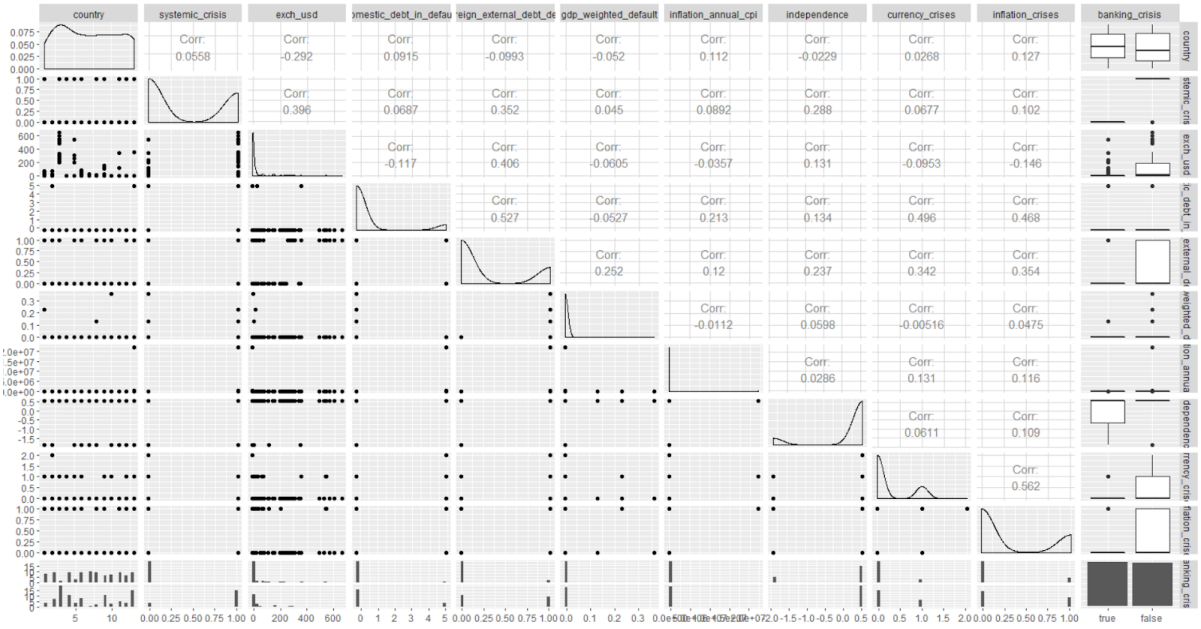

Figure 2 Data Analysis plot

Above, you can see the effect of all features on one another and the effect on the target feature. Accordingly, if the correlation value of a column with other columns is generally low, it may say that this column has no effect on the target feature. You can see correlation values and density plots from the table above. Now let's draw a colorful matrix that shows the values of the columns on top of each other.

1.2.1.2 Heat Map of Correlation Matrix

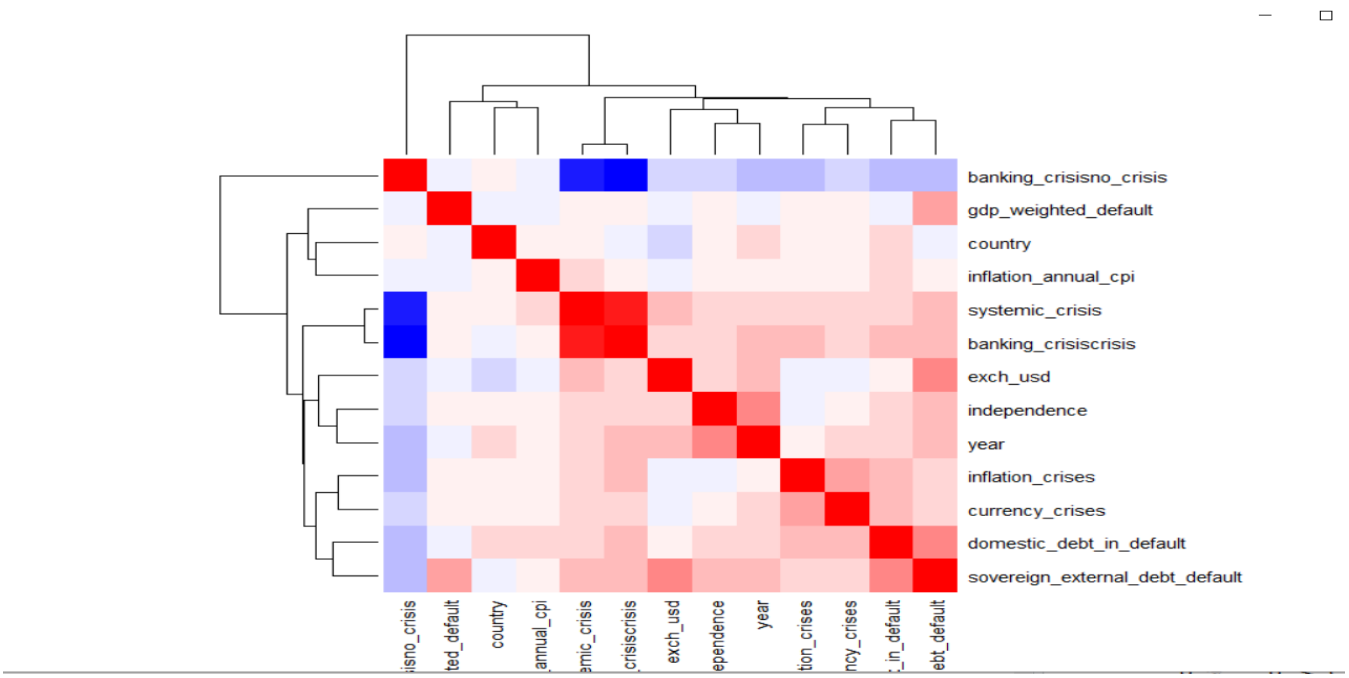

Figure 3 Heat Map

As we have seen, the systemic crisis has a huge impact on the target feature. It is very important for modeling and learning. On the other hand, columns such as gdp-weighted_default, country and currency crisis have no significant effect on the target feature and it is considered that removing some of these columns will not affect learning.

2. DATA PREPARATION

Data cleaning is applied for obtaining results closer to reality in dataset and facilitating analysis. Data cleaning has lots of steps.

2.1. Clean Data

2.1.1. Remove Duplicates and Missing Values

Before modelling the data the necessary step is to check whether the data has been repeated. Repeated data is unwanted situation when performing data training and analysis. It affects the project about time and cost poorly. The process that will done is to determine repeated instances or rows and convert them to just one instance (or row). Therefore, operations will be applied to an instance instead of multiple instances.Function duplicated () was used for finding repeated data and removing from the dataset. It was clearly seen that there is no duplicated data.

This technique is used to check whether the data is or not. It was analysed which instance contain null or NA values. In result of done technique, it is applied imputation or central method for removing missing values from dataset.There is no missing value in the selected data set as you can see in data analysis plot 1.

2.2. Normalization

As seen below, the range of numeric values “exch_usd” and “inlation_annual_cpi” are very wide. For this reason, normalization method is applied to these columns. Range normalization technique guarantees all features will have the exact same scale but does not handle outliers well. Due to these results, the range normalization technique was used in this data set for having more consistent results. Standardization (Z-score Normalization) was used.

```
      case            year      systemic_crisis      exch_usd       domestic_debt_in_default sovereign_external_debt_default gdp_weighted_default
 Min.   : 1.00   Min.   :1860   Min.   :0.00000   Min.   :  0.0000   Min.   :0.00000          Min.   :0.000                   Min.   :0.000000
 1st Qu.:15.00   1st Qu.:1951   1st Qu.:0.00000   1st Qu.:  0.1954   1st Qu.:0.00000          1st Qu.:0.000                   1st Qu.:0.000000
 Median :38.00   Median :1973   Median :0.00000   Median :  0.8684   Median :0.00000          Median :0.000                   Median :0.000000
 Mean   :35.61   Mean   :1968   Mean   :0.07743   Mean   : 43.1408   Mean   :0.03966          Mean   :0.153                   Mean   :0.006402
 3rd Qu.:56.00   3rd Qu.:1994   3rd Qu.:0.00000   3rd Qu.:  8.4627   3rd Qu.:0.00000          3rd Qu.:0.000                   3rd Qu.:0.000000
 Max.   :70.00   Max.   :2014   Max.   :1.00000   Max.   :744.3061   Max.   :1.00000          Max.   :1.000                   Max.   :0.400000
 inflation_annual_cpi  independence    currency_crises  inflation_crises   banking_crisis
 Min.   :     -29     Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   crisis    : 94
 1st Qu.:       2     1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:0.0000   no_crisis:965
 Median :       6     Median :1.0000   Median :0.0000   Median :0.0000
 Mean   :   20849     Mean   :0.7762   Mean   :0.1322   Mean   :0.1294
 3rd Qu.:      12     3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.0000
 Max.   :21989695     Max.   :1.0000   Max.   :2.0000   Max.   :1.0000
```

Figure 4 Before Normalization

```
      case            year      systemic_crisis      exch_usd       domestic_debt_in_default sovereign_external_debt_default gdp_weighted_default
 Min.   : 1.00   Min.   :1860   Min.   :0.00000   Min.   :-0.3870   Min.   :0.00000          Min.   :0.000                   Min.   :0.000000
 1st Qu.:15.00   1st Qu.:1951   1st Qu.:0.00000   1st Qu.:-0.3852   1st Qu.:0.00000          1st Qu.:0.000                   1st Qu.:0.000000
 Median :38.00   Median :1973   Median :0.00000   Median :-0.3792   Median :0.00000          Median :0.000                   Median :0.000000
 Mean   :35.61   Mean   :1968   Mean   :0.07743   Mean   : 0.0000   Mean   :0.03966          Mean   :0.153                   Mean   :0.006402
 3rd Qu.:56.00   3rd Qu.:1994   3rd Qu.:0.00000   3rd Qu.:-0.3111   3rd Qu.:0.00000          3rd Qu.:0.000                   3rd Qu.:0.000000
 Max.   :70.00   Max.   :2014   Max.   :1.00000   Max.   : 6.2899   Max.   :1.00000          Max.   :1.000                   Max.   :0.400000
 inflation_annual_cpi  independence    currency_crises  inflation_crises   banking_crisis
 Min.   :-0.03090      Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   crisis    : 94
 1st Qu.:-0.03085      1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:0.0000   no_crisis:965
 Median :-0.03085      Median :1.0000   Median :0.0000   Median :0.0000
 Mean   : 0.00000      Mean   :0.7762   Mean   :0.1322   Mean   :0.1294
 3rd Qu.:-0.03084      3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.0000
 Max.   :32.51140      Max.   :1.0000   Max.   :2.0000   Max.   :1.0000
```

Figure 5 After Normalization

2.3. Undersampling Methods

There are 2 type in target feature of dataset. This values are crisis and no_crisis. Distributions of target feature values in data set have been analyzed. Results are shown in below.

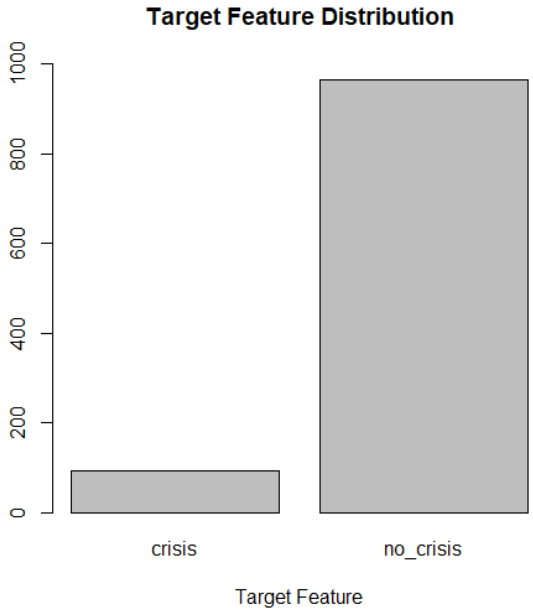

Figure 6 Target Feature Distribution

There is no equaly distribution. No_crisis values are much more than crisis values. There should not be that much difference between these values for a good prediction. Therefore, it is almost identical to the code below.

```
> balanced_africa <- ovun.sample(banking_crisis~., data=cleaned_africa, method="under",seed=1)$data
> summary(balanced_africa$banking_crisis)
no_crisis    crisis
       95        94
```

Figure 7 Undersampling Method

3. MACHINE LEARNING ALGORITHMS

3.1. K-Fold Cross Validation

The k-fold cross validation was used to improve the accuracy of the predictions. By using this method, models are analyzed better with the randomly choosed data. The “k” vaue is determined as 3 because only 198 sample left after the undersampling. Every step, different train and test set were created.

3.2. Training and Testing Sets

The training and testing subsets were used for 3 different machine learning model. In this process, %75 of dataset for training and %25 of dataset for testing were used. Every subset gave us different accuracy results.

3.3. Decision Tree

Datasets were trained with different subsets using the cross validaiton method. Three different decision trees were created and the decision tree with the best accuracy was choosed. The best decision tree is shown below and the metrics are as follows. According to the decision tree, four attributes are actually used in the tree construction which are "systemic_crisis", "year", "exch_usd", "case". Also “systemic_crisis” attribute mostly affects the banking_crisis target feature.

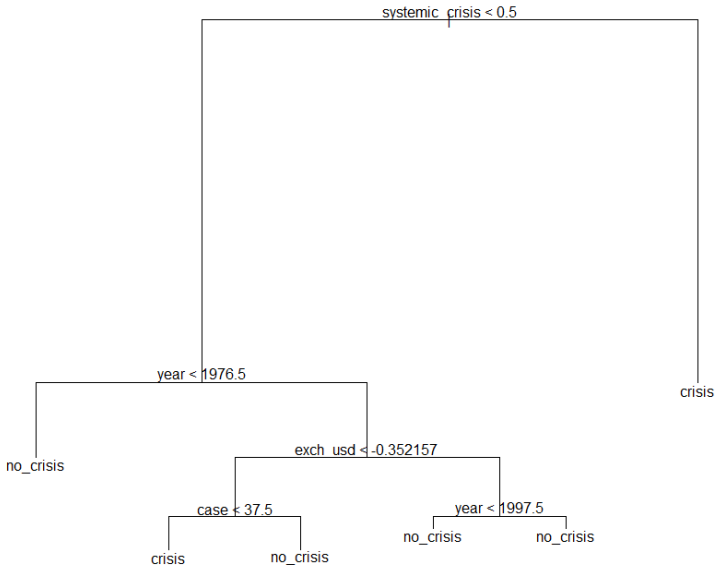

Figure 8 The Best Decision Tree

The all metric values are represented below. The best decision tree were created in the fold 2. The accuracy is 1. After the all folds, average metric values are calculated also. Average accuracy for decision tree is calculated as 0.9236111.

```
Accuracy Matrix
> print(accuracy_metric)
              Fold 1    Fold 2    Fold 3   Average
Decision Tree 0.8125 1.0000000 0.9583333 0.9236111
Naive Bayes   0.6875 0.8750000 0.8750000 0.8125000
KNN           0.8750 0.9166667 0.9166667 0.9027778
> cat("Precision Matrix\n")
Precision Matrix
> print(precision_metric)
              Fold 1    Fold 2    Fold 3   Average
Decision Tree 1.0000000 1.0000000 0.9629630 0.9876543
Naive Bayes   0.8636364 0.8928571 0.8888889 0.8817941
KNN           0.9090909 0.8571429 0.8888889 0.8850409
> cat("Recall Matrix\n")
Recall Matrix
> print(recall_metric)
              Fold 1    Fold 2    Fold 3   Average
Decision Tree 0.7096774 1.0000000 0.9629630 0.8908801
Naive Bayes   0.6129032 0.8928571 0.8888889 0.7982164
KNN           0.8333333 1.0000000 0.9600000 0.9311111
> cat("F1 Score Matrix\n")
F1 Score Matrix
> print(f1score_metric)
              Fold 1    Fold 2    Fold 3   Average
Decision Tree 0.8301887 1.0000000 0.9629630 0.9310505
Naive Bayes   0.7169811 0.8928571 0.8888889 0.8329091
KNN           0.8695652 0.9230769 0.9230769 0.9052397
```

Figure 9 The Metrics for the Decision Tree

```
Classification tree:
tree(formula = banking_crisis ~ ., data = balanced_africa, subset = africa_train_idx)
Variables actually used in tree construction:
[1] "systemic_crisis" "year"            "exch_usd"        "case"
Number of terminal nodes:  6
Residual mean deviance:  0.149 = 20.12 / 135
Misclassification error rate: 0.03546 = 5 / 141
```

Figure 10 Summary of the Decision Tree

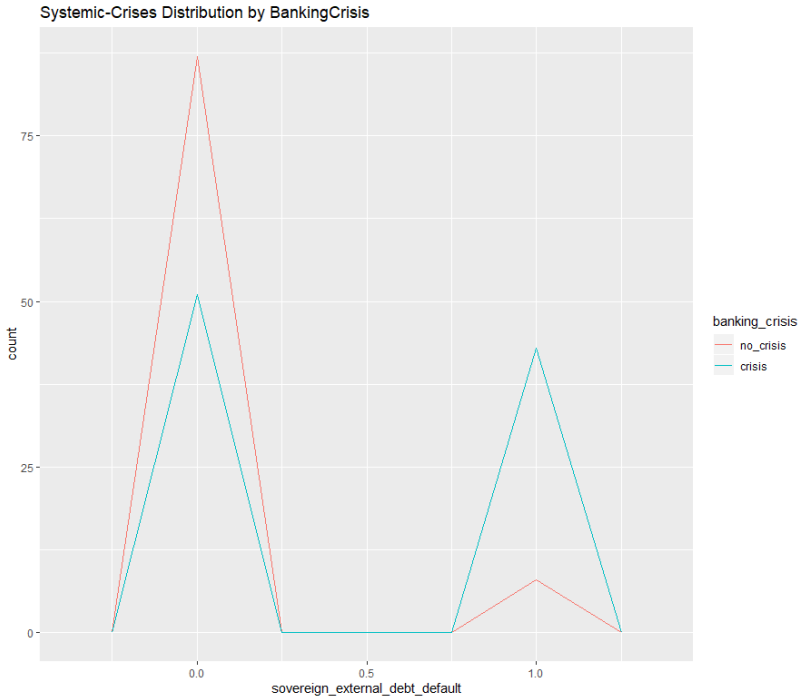

Figure 11 Systemic Crisis Distribution by Banking_crisis Plot

The run-time of the decision tree was calculated. The results are below. The average run-time is 0.005651.

```
Run-time of the Algorithms in each fold
> print(run_time_matrix)
                Fold 1      Fold 2       Fold 3      Average
Decision Tree 0.0099730492 0.003989935 0.0029928684 0.0056519508
Naive Bayes   0.0099768639 0.012968063 0.0090229511 0.0106559594
KNN           0.0009970665 0.000000000 0.0009548664 0.0006506443
```

Figure 11 The run-time for decision tree

3.4. Naive Bayes

Function naiveBayes() has been used. This algorithm is usedto calculate a probability for each feature (Bayes Theorem) on this data set. After calculation, dataset that is divided for test has been given to model and test dataset was analysed according to this probability. The model creates the conditional probability for each feature separately.

To analyze performance of data in dataset that was calculated with conditional probability has been applied predict() function. So , the result of test data in the model has been shown as below. In first iteration, result of data that is predicted is in below.

```
> (tab2 <- table(Conf_matrix, test_subset_naive_bayes$banking_crisis))

Conf_matrix  0  1
          0 24  3
          1  3 18
> |
```

Figure 12 Predicted Result for Naive Bayes

In the table below, you can see the values of Naive bayes algorithm such as run time, accuracy, precision, f1 Score and recall.

```
Accuracy Matrix
> print(accuracy_metric)
              Fold 1    Fold 2    Fold 3   Average
Decision Tree 0.8125 1.0000000 0.9583333 0.9236111
Naive Bayes   0.6875 0.8750000 0.8750000 0.8125000
KNN           0.8750 0.9166667 0.9166667 0.9027778
> cat("Precision Matrix\n")
Precision Matrix
> print(precision_metric)
                Fold 1    Fold 2    Fold 3   Average
Decision Tree 1.0000000 1.0000000 0.9629630 0.9876543
Naive Bayes   0.8636364 0.8928571 0.8888889 0.8817941
KNN           0.9090909 0.8571429 0.8888889 0.8850409
> cat("Recall Matrix\n")
Recall Matrix
> print(recall_metric)
                Fold 1    Fold 2    Fold 3   Average
Decision Tree 0.7096774 1.0000000 0.9629630 0.8908801
Naive Bayes   0.6129032 0.8928571 0.8888889 0.7982164
KNN           0.8333333 1.0000000 0.9600000 0.9311111
> cat("F1 Score Matrix\n")
F1 Score Matrix
> print(f1score_metric)
                Fold 1    Fold 2    Fold 3   Average
Decision Tree 0.8301887 1.0000000 0.9629630 0.9310505
Naive Bayes   0.7169811 0.8928571 0.8888889 0.8329091
KNN           0.8695652 0.9230769 0.9230769 0.9052397


Run-time of the Algorithms in each fold
> print(run_time_matrix)
                 Fold 1      Fold 2      Fold 3     Average
Decision Tree 0.0038020611 0.005043983 0.002993822 0.003946622
Naive Bayes   0.0163271427 0.015697002 0.015581131 0.015868425
KNN           0.0009989738 0.001198053 0.001365185 0.001187404
```

Figure 13 The metrics for Naive Bayes

When the results of the Naive Bayes probabilistic classification algorithm are examined, the accuracy is low in Fold 2 but quite high in the other two fold. However, it should not be decided only by looking at the accuracy value. Therefore, precision, recall and f1 score analysis are performed. In these investigations, the precision (percentage of what is classified as crisis is actually a crisis) is 0.88 on average. Recall (crisis detection rate correctly) is lower in the first example, but it is very high in others. Another important feature is the F1 score, which directly shows the success rate of the model. Accordingly, our model is successful.

3.5. K Nearest Neighbor

In this algorithm, different k values is found for each fold. To decide the best k value, the for loop turned 15 times but the results were generally between 1 and 3

```
Best K values for each fold
> print(best_k_each_fold)
             Fold 1 Fold 2 Fold 3
Best K Value      3      1      3
>
```

Figure 14 The best k values

Thw metric results are always as the same as the decision tree algorithm.

```
Accuracy Matrix
> print(accuracy_metric)
              Fold 1    Fold 2    Fold 3   Average
Decision Tree 0.8125 1.0000000 0.9583333 0.9236111
Naive Bayes   0.6875 0.8750000 0.8750000 0.8125000
KNN           0.8750 0.9166667 0.9166667 0.9027778
> cat("Precision Matrix\n")
Precision Matrix
> print(precision_metric)
                Fold 1    Fold 2    Fold 3   Average
Decision Tree 1.0000000 1.0000000 0.9629630 0.9876543
Naive Bayes   0.8636364 0.8928571 0.8888889 0.8817941
KNN           0.9090909 0.8571429 0.8888889 0.8850409
> cat("Recall Matrix\n")
Recall Matrix
> print(recall_metric)
                Fold 1    Fold 2    Fold 3   Average
Decision Tree 0.7096774 1.0000000 0.9629630 0.8908801
Naive Bayes   0.6129032 0.8928571 0.8888889 0.7982164
KNN           0.8333333 1.0000000 0.9600000 0.9311111
> cat("F1 Score Matrix\n")
F1 Score Matrix
> print(f1score_metric)
                Fold 1    Fold 2    Fold 3   Average
Decision Tree 0.8301887 1.0000000 0.9629630 0.9310505
Naive Bayes   0.7169811 0.8928571 0.8888889 0.8329091
KNN           0.8695652 0.9230769 0.9230769 0.9052397


Run-time of the Algorithms in each fold
> print(run_time_matrix)
                 Fold 1       Fold 2       Fold 3      Average
Decision Tree 0.0038020611 0.005043983 0.002993822 0.003946622
Naive Bayes   0.0163271427 0.015697002 0.015581131 0.015868425
KNN           0.0009989738 0.001198053 0.001365185 0.001187404
```

Figure 15 The metrics for KNN

4. CONCLUSION & COMPARISONS

In this project, three machine learning techniques are evaluated which are decision tree, naive bayes and k-nearest neighbor algorithms. In order to compare these models, several metrics were used. These metrics are accuracy, precision, recall and f1 score. After that, the run-time of the algorithms are compared too.

As a result of the term project, the most successful and the fastest algorithm is found as Decision Tree with the accuracy 0.923611. The all comparisons are shown below.

```
Accuracy Matrix
> print(accuracy_metric)
              Fold 1    Fold 2    Fold 3   Average
Decision Tree 0.8125 1.0000000 0.9583333 0.9236111
Naive Bayes   0.6875 0.8750000 0.8750000 0.8125000
KNN           0.8750 0.9166667 0.9166667 0.9027778
> cat("Precision Matrix\n")
Precision Matrix
> print(precision_metric)
                Fold 1    Fold 2    Fold 3   Average
Decision Tree 1.0000000 1.0000000 0.9629630 0.9876543
Naive Bayes   0.8636364 0.8928571 0.8888889 0.8817941
KNN           0.9090909 0.8571429 0.8888889 0.8850409
> cat("Recall Matrix\n")
Recall Matrix
> print(recall_metric)
                Fold 1    Fold 2    Fold 3   Average
Decision Tree 0.7096774 1.0000000 0.9629630 0.8908801
Naive Bayes   0.6129032 0.8928571 0.8888889 0.7982164
KNN           0.8333333 1.0000000 0.9600000 0.9311111
> cat("F1 Score Matrix\n")
F1 Score Matrix
> print(f1score_metric)
                Fold 1    Fold 2    Fold 3   Average
Decision Tree 0.8301887 1.0000000 0.9629630 0.9310505
Naive Bayes   0.7169811 0.8928571 0.8888889 0.8329091
KNN           0.8695652 0.9230769 0.9230769 0.9052397
> cat("Run-time of the Algorithms in each fold\n")
Run-time of the Algorithms in each fold
> print(run_time_matrix)
                 Fold 1       Fold 2      Fold 3     Average
Decision Tree 0.003999949 0.0039989948 0.002993822 0.003664255
Naive Bayes   0.015999079 0.0133380890 0.013287067 0.014208078
KNN           0.000000000 0.0009970665 0.001001120 0.000666062
```

Figure 16 The metrics for all algorithms

REFERENCES

Fundamental Techniques of Feature Engineering for Machine Learning (2019) from https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114

Dealing with imbalanced data: undersampling, oversampling and proper cross-validation (2015) from https://www.marcoaltini.com/blog/dealing-with-imbalanced-data-undersampling-oversampling-and-proper-cross-validation

Africa Economic, Banking and Systemic Crisis Data (2015) from https://www.kaggle.com/chirin/africa-economic-banking-and-systemic-crisis-data

Cross Validation Essentials in R (2018) from http://www.sthda.com/english/articles/38-regression-model-validation/157-cross-validation-essentials-in-r/