Project Report

James Cook University

Subject: CP3404 – Data mining

Tutor: Eric Tham

Group 10:

- Van Phuong Nguyen - 13839051
- Le Hoang Nguyen - 13744451
- Duong Khai Nguyen - 13593335
- Le Huy Bao Nguyen - 13718013

Table of Content

1. Abstract

In the banking business, credit score cards are a frequent risk control approach. It predicts the likelihood of future defaults and credit card borrowings based on personal information and data provided by credit card applicants. The bank has the authority to decide whether or not to offer the applicant a credit card. Credit scores can be used to objectively measure the severity of a risk. Our purpose is to distinguish multiple factors that influence the probability of overdue payments, to categorize ‘good’ and ‘bad’ clients for future credit card approvals. Because of the complexity of the dataset, we first used preprocessing techniques to standardize the data. Then, classification algorithms including 1R, Decision Tree, Naive Bayesian, Artificial Neural Networks, Support Vector Machines, and Nearest Neighbor were implemented to place data into preset categories. We also utilized a clustering method to decide the importance of attributes in our task. We get the optimal model by utilizing Nearest Neighbor Classification, which is currently the best algorithm for creating the most useful model.

2. Introduction

a. Case study

Our mining team was hired by Commonwealth Bank Australia to distinguish multiple factors that predict ‘good’ and ‘bad’ clients for future credit card approvals. The credit records and customers in recent years were given for the mining tasks. We need to identify which aspects correlated to the overdue payments, then consult the bank whether to issue credit cards to specific groups of upcoming applicants.

b. Objective

Our goal of the project is to manufacture a classifier to predict if specific groups of new applicants will be the ‘good’ or ‘bad’ customer for Commonwealth Bank. By implementing such a system, the bank will be able to issue credit cards to trustworthy customers. The bank can organize a better management of available assets by focusing on possible clients "selected" by the classifier, which would increase their efficiency. They could also concentrate their resources on next marketing strategies to lower costs and increase earnings. It can help avoid a time-consuming and tedious process of gathering, checking, validating, and deciding on data, which was previously done manually, by utilizing data mining techniques and algorithms that should be readily available. As a result, a massive amount of data from consumers can be successfully handled before the bank can focus its attention on new clients. Obviously, having such a system is extremely persuasive and essential for any bank. It is capable of handling a large amount of data from consumers in a short period of time. It is an indisputable essential to have the system if the bank wants to stay competitive in a crowded country where many financial companies can speed up the process of checking customer information and quickly reach an inference based on given data from clients in just a few clicks.

3. Methodology

a. Data overview

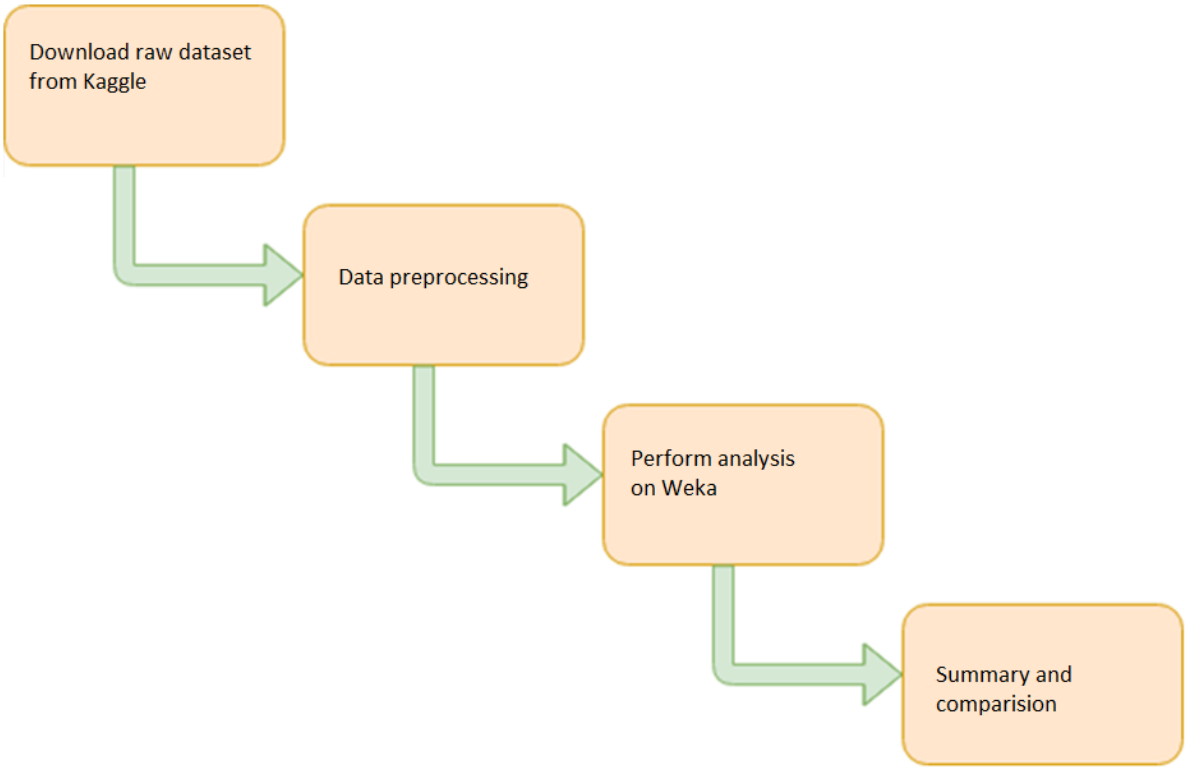
The dataset was given by Commonwealth Bank that recorded applicants and credit payments in recent years. This multivariate dataset consists of two csv files named “application_record.csv” and “credit_record.csv”. There are 20 attributes and 1048576 credit records, 438558 applicants in total. The numeric attributes are ID, CNT_CHILDREN, AMT_INCOME_TOTAL, CNT_FAM_MEMBERS and MONTHS_BALANCE. The boolean attributes are FLAG_OWN_CAR, FLAG_OWN_REALITY, FLAG_MOBIL, FLAG_WORK_PHONE, FLAG_EMAIL. There are 7 nominal attributes including CODE_GENDER, NAME_INCOME_TYPE, NAME_EDUCATION_TYPE, NAME_FAMILY_STATUS, NAME_HOUSING_TYPE, OCCUPATION_TYPE, STATUS. Date attributes are DAYS_BIRTH and DAYS_EMPLOYED. Classification was used to analyse the dataset in order to reach these targets. Despite the fact that the records were unbalanced, solutions to the problem were pursued in order to reduce biased performances. Furthermore, because the input dataset is multivariate, some attributes will be ignored during the mining process. Our data also had missing values, necessitating the use of preprocessing techniques afterwards.

application_record.csv	
Feature Name	*Explanation*
ID	Client number
CODE_GENDER	Gender
FLAG_OWN_CAR	Is there a car
FLAG_OWN_REALITY	Is there a property
CNT_CHILDREN	Number of children
AMT_INCOME_TOTAL	Annual income
NAME_INCOME_TYPE	Income category
NAME_EDUCATION_TYPE	Education level
NAME_FAMILY_STATUS	Marital status
NAME_HOUSING_TYPE	Way of living
DAYS_BIRTH	Birthday
DAYS_EMPLOYED	Start date of employment
FLAG_MOBIL	Is there a mobile phone
FLAG_WORK_PHONE	Is there a work phone

FLAG_EMAIL	Is there an email
OCCUPATION_TYPE	Occupation
CNT_FAM_MEMBERS	Family size

credit_record.csv	
Feature name	*Explanation*
ID	Client number
MONTHS_BALANCE	Record month
STATUS	Status

b. Project Process



The process of our project is depicted in the diagram above. The dataset was downloaded from the Kaggle site right away, and then we selected our goal and conducted an audit of the dataset. From there, we'll erase repetitive attributes before applying the filter to ensure that the dataset is in the best possible state before applying the algorithms. We selected 6 excellent classification algorithms: Decision Tree (J48), Naive Bayes, Multilayer Perceptron, Logistic Regression, OneR, KNN; and 1 clustering algorithm: SimpleKMeans after evaluating and considering them. Those algorithms will load the preprocessed dataset into WEKA till the finish, and the results and comparisons will show.

4. Preprocessing Details

a. Data reduction

For this campaign only, the bank requested us to do mining tasks for the current month. This luckily reduced the great amount of instances, which is later convenient for our jobs. Therefore, we needed to remove the value except 0 from the MONTHS_BALANCE attribute. We used python and its famous pandas library to do so . We saved the new file and named it "credit_record_clean.csv".

b. Data merge and transformation

Due to the bank policy, the status that is equal or greater than 2 will be considered as "BAD". We used python to merge the data from two seperate files by IDs and transform the STATUS attribute (*Image 4.1)*

c. Remove unnecessary attributes.

We removed ID, CODE_GENDER, DAYS_EMPLOYED, which are not suitable subjects for mining by python. We have also removed MONTHS_BALANCE since we keep the current month only, then export to a new file called “credit_approval.csv”. The results will show in *Image 4.2.*

d. Handling missing data

OCCUPATION_TYPE is the only attribute with 31% missing values (*Image 4.3*). We used weka to fill in missing ones with mean data (*Image 4.4)*.

e. Handling asymmetric data of dataset:

Asymmetric data occurs when the bulk of occurrences have one response and only a few have the opposite response. In other words, the data's mean is skewed to one side, causing data classification and prediction to be distorted. In the chosen dataset after preprocessing, there are only 89 "BAD" customers, despite the fact that there are 24583 "GOOD" ones, accounting for 99,993 percent of all responses.

Changing the method for quantifying algorithm performance is the solution to this problem. Because the proportion of Correctly Classified Instances is commonly used to assess classifier performance, the ROC Curve Area and ROC Area were used instead. The Receiver Operating Characteristics Area, or ROC Area, is a formula for calculating the area under the curve of classified cases. Because the Correctly Classified Cases assessment only measures the mean accuracy of instances that are predisposed due to the majority of "GOOD" clients, the ROC Area would highlight the general exactness of the occurrences, resulting in instances with increasingly sensible performance.

```python
# import files
import pandas as pd
application_record = pd.read_csv('application_record.csv')
credit_record = pd.read_csv('credit_record_clean.csv')

# merge files
credit_approval = pd.merge(application_record, credit_record)

# drop unnessary columns
credit_approval.drop('ID', inplace=True, axis=1)
credit_approval.drop('CODE_GENDER', inplace=True, axis=1)
credit_approval.drop('MONTHS_BALANCE', inplace=True, axis=1)
credit_approval.drop('DAYS_EMPLOYED', inplace=True, axis=1)

# transform status
for i, x in enumerate(credit_approval['STATUS']):
    if x == 'C' or i == 'X':
        credit_approval['STATUS'][i] = 'GOOD'
    elif x == '3' or x =='2' or x == '4' or x == '5':
        credit_approval['STATUS'][i] = 'BAD'
    elif x:
        credit_approval['STATUS'][i] = 'GOOD'


# export files
credit_approval.to_csv("credit_approval.csv")
```

Image 4.1

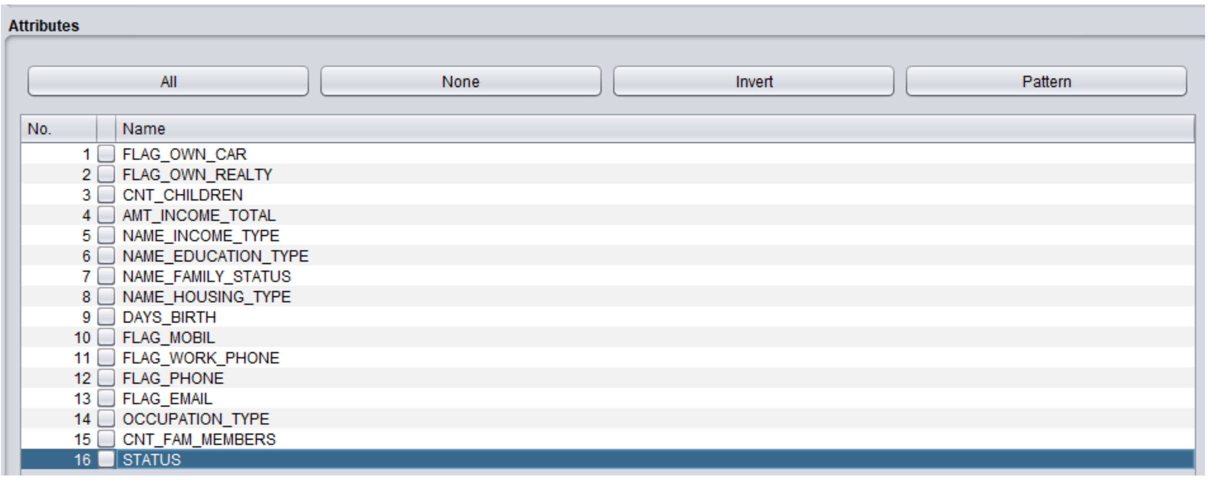


Image 4.2

Selected attribute

Name: OCCUPATION_TYPE — Type: Nominal
Missing: 7629 (31%) — Distinct: 18 — Unique: 0 (0%)

No.	Label	Count	Weight
1	Security staff	406	406.0
2	Sales staff	2328	2328.0
3	Accountants	853	853.0
4	Laborers	4293	4293.0
5	Managers	2008	2008.0
6	Drivers	1504	1504.0
7	Core staff	2381	2381.0
8	High skill tech staff	950	950.0
9	Cleaning staff	385	385.0
10	Private service staff	235	235.0
11	Cooking staff	449	449.0
12	Low-skill Laborers	123	123.0
13	Medicine staff	796	796.0
14	Secretaries	98	98.0
15	Waiters/barmen staff	88	88.0
16	HR staff	53	53.0
17	Realty agents	47	47.0
18	IT staff	46	46.0

Class: STATUS (Nom) — Visualize All

Image 4.3

Selected attribute

Name: OCCUPATION_TYPE — Type: Nominal
Missing: 0 (0%) — Distinct: 18 — Unique: 0 (0%)

No.	Label	Count	Weight
1	Security staff	406	406.0
2	Sales staff	2328	2328.0
3	Accountants	853	853.0
4	Laborers	11922	11922.0
5	Managers	2008	2008.0
6	Drivers	1504	1504.0
7	Core staff	2381	2381.0
8	High skill tech staff	950	950.0
9	Cleaning staff	385	385.0
10	Private service staff	235	235.0
11	Cooking staff	449	449.0
12	Low-skill Laborers	123	123.0
13	Medicine staff	796	796.0
14	Secretaries	98	98.0
15	Waiters/barmen staff	88	88.0
16	HR staff	53	53.0
17	Realty agents	47	47.0
18	IT staff	46	46.0

Class: STATUS (Nom) — Visualize All

Image 4.4

5. Algorithms Used

a. Simple K-means

The basic and widely used clustering approach is KMeans. It is dependent on the partitioning mechanism used. It divides n data items into k-groups, where k denotes the number of clusters a client specifies. Clusters are framed so that each item in the cluster is as close to the centroid as possible. The K-Means algorithm uses the Euclidean distance measurement to determine the distance between an item and the centroid. This is the result of Simple K-means clustering

b. 1R Classification

OneR is a straightforward yet precise classification technique that generates one rule for each data predictor and then identifies the rule with the smallest overall error as the one rule. To establish a rule for a predictor, we build a frequency table against the goal for each predictor.

c. Decision Tree

Decision trees can be drawn by hand or with the help of a graphics application. They can be used to assign money, time, or other values to enable computerized decision-making. In data mining, decision tree software is used to simplify complex problems and assess the cost-effectiveness of research and business decisions. In a decision tree, variables are commonly represented by circles.

d. Naive Bayesian

The Bayes Theorem is used to create the Naive Bayes classification algorithm. It forecasts participation probability for each class, such as the possibility that a particular record or data point will be assigned to a particular class. The most likely class is defined as the one having the highest probability. The Naive Bayes classifier assumes that all of the features are unrelated. The presence or absence of a component has no bearing on the presence or absence of other features. The Naive Bayes model is simple to put together and is very useful for huge data sets. Naive Bayes is renowned to outperform even the most powerful classifiers because of its simplicity.

e. Artificial Neural Networks

Multilayer perceptrons are perceptron systems, often known as direct classifier systems. A perceptron is a model of a single neuron that predates larger neural networks. It's a branch of computer science that studies how simple models of biological brains may be used to solve difficult computing problems like the predictive modeling tasks seen in machine learning. The goal is to construct strong algorithms and data structures that can be used to represent challenging situations, rather than to create actual brain models. The varied levels or multi-layered structure of brain systems is what gives them their precognitive ability.

f. Logistic Regression

Logistic regression is a statistical analysis tool for predicting information value based on previous data set perceptions. In the field of machine learning, logistic regression has become an important tool. The method allows a machine learning application to classify incoming data using an algorithm based on historical data. As more critical data is received, the algorithm should show evidence of improving its ability to predict classifications within data sets. Logistic regression can also help with information planning by allowing data sets to be placed into specified containers throughout the extract, convert, and load processes.

g. Nearest Neighbor Classification

The K-Nearest Neighbour (KNN) algorithm is a non-parametric approach in which the input consists of the k nearest training instances in the future space; the output is a class membership when the K-Nearest Neighbour algorithm is used for classification. A new item's classification is determined by a majority vote of its neighbors, with the object being assigned to the class that is most common among its k nearest neighbors.

6. Data mining processing

a. Simple K-means

```
Final cluster centroids:
                                                                  Cluster#
Attribute                                     Full Data                  0                          1
                                              (19738.0)          (12312.0)                   (7426.0)
=====================================================================================================
FLAG_OWN_CAR                                          N                  N                          Y
FLAG_OWN_REALTY                                       Y                  Y                          Y
CNT_CHILDREN                                     0.4229              0.367                     0.5158
AMT_INCOME_TOTAL                            185464.3786        168870.8527                212975.7564
NAME_INCOME_TYPE                                Working            Working                    Working
NAME_EDUCATION_TYPE        Secondary / secondary special Secondary / secondary special Secondary / secondary special
NAME_FAMILY_STATUS                              Married            Married                    Married
NAME_HOUSING_TYPE                     House / apartment  House / apartment          House / apartment
FLAG_MOBIL                                            1                  1                          1
FLAG_WORK_PHONE                                  0.2409             0.2327                     0.2545
FLAG_PHONE                                       0.2967             0.2996                     0.2918
FLAG_EMAIL                                       0.0846             0.0807                      0.091
OCCUPATION_TYPE                                Laborers           Laborers                   Laborers
CNT_FAM_MEMBERS                                  2.1891             2.0887                     2.3556
STATUS                                             GOOD               GOOD                       GOOD




Time taken to build model (full training data) : 0.23 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      12312 ( 62%)
1       7426 ( 38%)
```

b. 1R Classification

10 fold cross validation for training dataset:

```
Time taken to build model: 0.08 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       19650               99.5542 %
Incorrectly Classified Instances        88                0.4458 %
Kappa statistic                          0
Mean absolute error                      0.0045
Root mean squared error                  0.0668
Relative absolute error                 49.913  %
Root relative squared error            100.2235 %
Total Number of Instances            19738

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.996      1.000    0.998      ?        0.500     0.996     GOOD
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.004     BAD
Weighted Avg.    0.996    0.996    ?          0.996    ?          ?        0.500     0.991

=== Confusion Matrix ===

     a     b   <-- classified as
 19650     0 |     a = GOOD
    88     0 |     b = BAD
```

	OneR: Cross-validation folds = 10
ROC Area	0.5
Time Taken	0.08
Kappa	0
Accuracy	99.5542%

10 fold cross validation for test dataset:

```
Time taken to build model: 0.06 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        4933               99.9797 %
Incorrectly Classified Instances         1                0.0203 %
Kappa statistic                          0
Mean absolute error                      0.0002
Root mean squared error                  0.0142
Relative absolute error                 32.1589 %
Root relative squared error             99.9763 %
Total Number of Instances             4934

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    1.000      1.000    1.000      ?        0.500     1.000     GOOD
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.000     BAD
Weighted Avg.    1.000    1.000    ?          1.000    ?          ?        0.500     1.000

=== Confusion Matrix ===

    a    b   <-- classified as
 4933    0 |    a = GOOD
    1    0 |    b = BAD
```

	OneR: Cross-validation folds = 10
ROC Area	**0.5**
Time Taken	**0.06**
Kappa	**0**
Accuracy	**99.9797%**

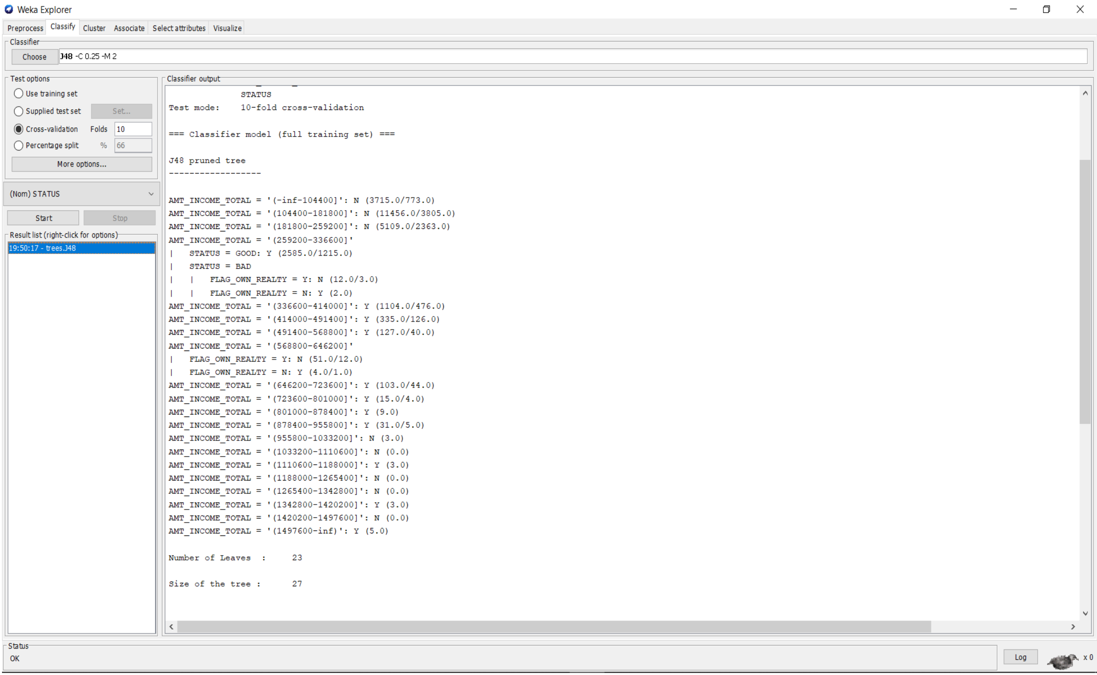
c. Decision Tree

We remove the unnecessary value for the dataset and keep 4 values FLAG_OWN_CAR,FLAG_OWN_RELTY, AMT_INCOME_TOTAL, STATUS . In addition, we use Discretize for AMT_INCOME_TOTAL with 20 bins.

This is the result when we tried the 10 fold cross validation on both training data set and testing data set:

10 fold cross validation on training set: Pick FLAG_OWN_CAR variable and then Start Algorithm

J48 pruned tree



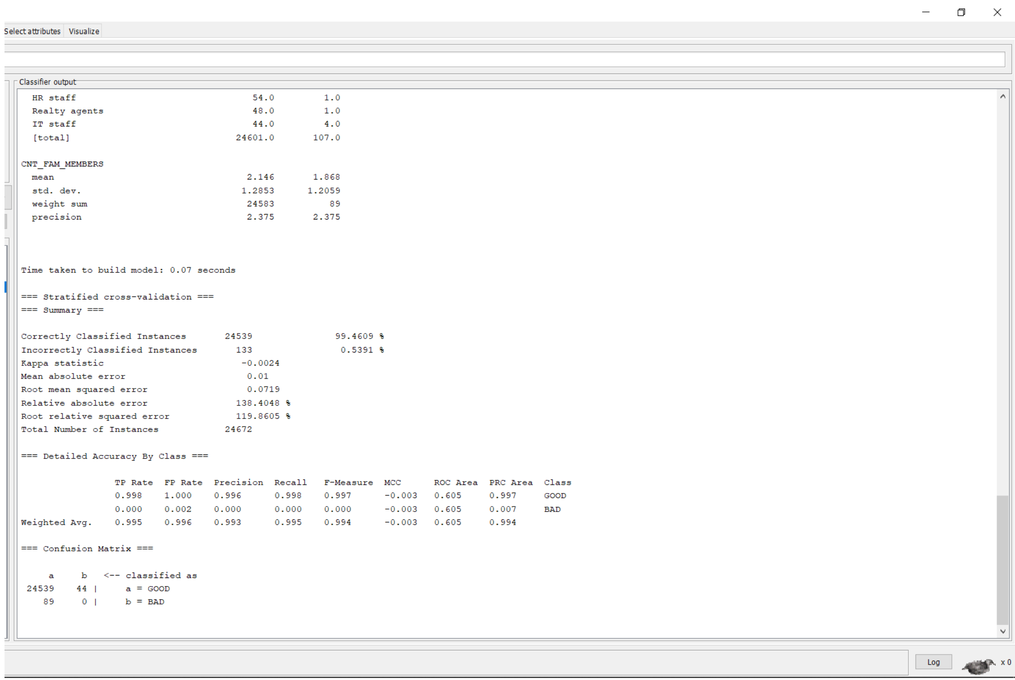
	Decision Tree -J48
ROC Area	0.624
Kappa	0.1461
Accuracy	0.6398

```
Number of Leaves  :     23

Size of the tree :      27


Time taken to build model: 0.06 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       15787               63.9875 %
Incorrectly Classified Instances      8885               36.0125 %
Kappa statistic                          0.1461
Mean absolute error                      0.4453
Root mean squared error                  0.472
Relative absolute error                 94.5148 %
Root relative squared error             97.2599 %
Total Number of Instances            24672

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.255    0.125    0.556      0.255    0.350      0.167    0.624     0.495     Y
                 0.875    0.745    0.657      0.875    0.751      0.167    0.624     0.709     N
Weighted Avg.    0.640    0.509    0.619      0.640    0.599      0.167    0.624     0.628

=== Confusion Matrix ===

     a     b   <-- classified as
  2394  6977 |     a = Y
  1908 13393 |     b = N
```

d. Naive Bayesian

10 fold cross validation for test dataset:

	Naive Bayesian
ROC Area	0.605
Kappa	-0.0024
Accuracy	0.9946



Select attributes Visualize

Classifier output

```
  HR staff                          54.0        1.0
  Realty agents                     48.0        1.0
  IT staff                          44.0        4.0
  [total]                        24601.0      107.0

CNT_FAM_MEMBERS
  mean                             2.146      1.868
  std. dev.                       1.2853     1.2059
  weight sum                       24583         89
  precision                        2.375      2.375



Time taken to build model: 0.07 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       24539               99.4609 %
Incorrectly Classified Instances       133                0.5391 %
Kappa statistic                         -0.0024
Mean absolute error                      0.01
Root mean squared error                  0.0719
Relative absolute error                138.4048 %
Root relative squared error            119.8605 %
Total Number of Instances            24672

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.998    1.000    0.996      0.998    0.997      -0.003   0.605     0.997     GOOD
                 0.000    0.002    0.000      0.000    0.000      -0.003   0.605     0.007     BAD
Weighted Avg.    0.995    0.996    0.993      0.995    0.994      -0.003   0.605     0.994

=== Confusion Matrix ===

     a     b   <-- classified as
 24539    44 |     a = GOOD
    89     0 |     b = BAD
```

Log

e. Artificial Neural Networks

Training set:

	Multilayer perceptrons
ROC Area	0.600
Kappa	0.1272
Accuracy	0.9958

```
=== Evaluation on training set ===

Time taken to test model on training data: 0.35 seconds

=== Summary ===

Correctly Classified Instances       19656               99.5846 %
Incorrectly Classified Instances        82                0.4154 %
Kappa statistic                          0.1272
Mean absolute error                      0.0043
Root mean squared error                  0.0638
Relative absolute error                 48.3078 %
Root relative squared error             95.7341 %
Total Number of Instances            19738

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.932    0.996      1.000    0.998      0.261    0.600     0.997     GOOD
                 0.068    0.000    1.000      0.068    0.128      0.261    0.600     0.106     BAD
Weighted Avg.    0.996    0.928    0.996      0.996    0.994      0.261    0.600     0.993

=== Confusion Matrix ===

     a     b   <-- classified as
 19650     0 |     a = GOOD
    82     6 |     b = BAD
```

Test result:

	Multilayer perceptrons
ROC Area	0.322
Kappa	0
Accuracy	0.9998

```
Time taken to build model: 165.03 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.09 seconds

=== Summary ===

Correctly Classified Instances        4933               99.9797 %
Incorrectly Classified Instances         1                0.0203 %
Kappa statistic                          0
Mean absolute error                      0.0005
Root mean squared error                  0.0147
Relative absolute error                 11.0148 %
Root relative squared error             98.7416 %
Total Number of Instances             4934

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    1.000      1.000    1.000      ?        0.322     1.000     GOOD
                 0.000    0.000    ?          0.000    ?          ?        0.322     0.000     BAD
Weighted Avg.    1.000    1.000    ?          1.000    ?          ?        0.322     1.000

=== Confusion Matrix ===

    a    b   <-- classified as
 4933    0 |    a = GOOD
    1    0 |    b = BAD
```

ROC Area of training set is 0.600, for test set is 0.322

f. Logistic Regression

This is the result when we tried the 10 fold cross validation on both training data set and testing data set:

10 fold cross validation on training set:

	Logistic Regression
ROC Area	0.591
Kappa	0
Accuracy	0.9955

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       19650               99.5542 %
Incorrectly Classified Instances        88                0.4458 %
Kappa statistic                          0
Mean absolute error                      0.0087
Root mean squared error                  0.0663
Relative absolute error                 97.8394 %
Root relative squared error             99.5902 %
Total Number of Instances            19738

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.996      1.000    0.998      ?        0.591     0.996     GOOD
                 0.000    0.000    ?          0.000    ?          ?        0.591     0.026     BAD
Weighted Avg.    0.996    0.996    ?          0.996    ?          ?        0.591     0.992

=== Confusion Matrix ===

     a     b   <-- classified as
 19650     0 |     a = GOOD
    88     0 |     b = BAD
```

Test result:

	Logistic Regression
ROC Area	0.123
Kappa	0
Accuracy	0.9998

```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.02 seconds

=== Summary ===

Correctly Classified Instances        4933               99.9797 %
Incorrectly Classified Instances         1                0.0203 %
Kappa statistic                          0
Mean absolute error                      0.0048
Root mean squared error                  0.0157
Relative absolute error                101.9655 %
Root relative squared error            105.612  %
Total Number of Instances             4934

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    1.000      1.000    1.000      ?        0.123     1.000     GOOD
                 0.000    0.000    ?          0.000    ?          ?        0.123     0.000     BAD
Weighted Avg.    1.000    1.000    ?          1.000    ?          ?        0.123     0.999

=== Confusion Matrix ===

    a    b   <-- classified as
 4933    0 |    a = GOOD
    1    0 |    b = BAD
```

ROC Area of training set is 0.591 and for test set is 0.123.

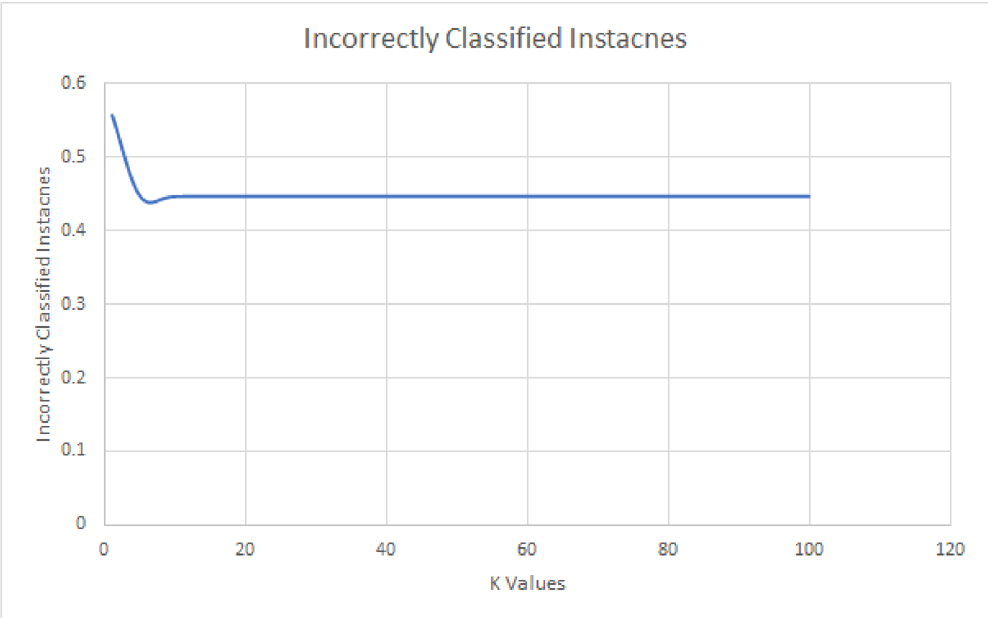
g. Nearest Neighbor Classification

As k increases, bias increases but variance drops extensively. To deal with this, we need to make use of 10-fold cross-validation. The best result generated with k-value is the one that minimizes the misclassification rate for the validation data set.

The data was then tested with k = 1, k = 5 , k = 10, k = 20, k = 30 , k = 50, k = 60, k = 70, k = 80, k = 90 and k=100 to determine variations between misclassification percentages of each parameter.

K Values	Incorrectly Classified Instances (%)
1	0.5573
5	0.4458
10	0.4458

20	0.4458
30	0.4458
50	0.4458
60	0.4458
70	0.4458
80	0.4458
90	0.4458
100	0.4458



Training set:

	Nearest Neighbor Classification
ROC Area	0.741
Kappa	0.1106
Accuracy	0.9944

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       19628               99.4427 %
Incorrectly Classified Instances       110                0.5573 %
Kappa statistic                          0.1106
Mean absolute error                      0.0075
Root mean squared error                  0.0763
Relative absolute error                 83.9817 %
Root relative squared error            114.539  %
Total Number of Instances            19738

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.999    0.920    0.996      0.999    0.997      0.122    0.741     0.998     GOOD
                 0.080    0.001    0.194      0.080    0.113      0.122    0.741     0.053     BAD
Weighted Avg.    0.994    0.916    0.992      0.994    0.993      0.122    0.741     0.994

=== Confusion Matrix ===

     a     b   <-- classified as
 19621    29 |     a = GOOD
    81     7 |     b = BAD
```

Test set:

	Nearest Neighbor Classification
ROC Area	0.480
Kappa	-0.004
Accuracy	0.9976

```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 4.7 seconds

=== Summary ===

Correctly Classified Instances        4922               99.7568 %
Incorrectly Classified Instances        12                0.2432 %
Kappa statistic                         -0.0004
Mean absolute error                      0.0051
Root mean squared error                  0.0575
Relative absolute error                109.2056 %
Root relative squared error            386.6674 %
Total Number of Instances             4934

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.998    1.000    1.000      0.998    0.999      -0.001   0.480     1.000     GOOD
                 0.000    0.002    0.000      0.000    0.000      -0.001   0.480     0.000     BAD
Weighted Avg.    0.998    1.000    1.000      0.998    0.999      -0.001   0.480     1.000

=== Confusion Matrix ===

    a    b   <-- classified as
 4922   11 |    a = GOOD
    1    0 |    b = BAD
```

7. Summary:

7.1. Accuracy:

	Simple K - mean	1R classification	Decision Tree	Naive Bayes ian	Artificial Neural Networks	Logistic Regression	Nearest Neighbor Classification
AUG	**0.5**	**0.5**	**0.624**	**0.605**	**0.6**	**0.591**	**0.741**
Accuray	**0.995**	**0.997**	**0.6398**	**0.9946**	**0.9958**	**0.591**	**0.9944**

Following the table above, we realize that Nearest Neighbor Classification is the best method. Although Simple K-mean has a high accuracy ratio(99,7%), it is considered the worst algorithm which we used in this study.

7.2. Time taken:

	Simple K - mean	1R classification	Decision Tree	Naive Bayes ian	Artificial Neural Networks	Logistic Regression	Nearest Neighbor Classification
Time taken (seconds)	**0.08**	**0.06**	**0.06**	**0.07**	**0.35**	**0.02**	**4.7**

Although Nearest Neighbor Classification is the best method, it took a lot of time to process (4.7 seconds) around 100 times compared to the rest. Otherwise, Logistic Regression spends 0.02 to run the process and become the fastest method.

8. Issues

Since we used data for the current month only according to the bank's request, the result may not be as reliable as we expected if we used data in the last 30 months. However, the data for such months was huge with more than 1 million records which may not be the suitable subject for mining. In addition, we found out that the numbers between "BAD" and "GOOD" clients are very imbalanced.

9. References

Artificial neural Networks applications and algorithms. XenonStack. (2021, September 20). Retrieved September 25, 2021, from https://www.xenonstack.com/blog/artificial-neural-network-applications#:~:text=A%20 neural%20network%20is%20a,without%20redesigning%20the%20output%20procedure.

Brownlee, J. (2020, August 20). *One-Class classification algorithms for imbalanced datasets*. Machine Learning Mastery. Retrieved September 25, 2021, from https://machinelearningmastery.com/one-class-classification-algorithms/.

Data Mining - Overview. Data mining - overview. (n.d.). Retrieved September 25, 2021, from https://www.tutorialspoint.com/data_mining/dm_overview.htm.

Decision tree Algorithm, Explained. KDnuggets. (n.d.). Retrieved September 25, 2021, from https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html#:~:text=Decision%20Tree%20algorithm%20belongs%20to%20the%20family%20of%20supervised%20learning%20algorithms.&text=The%20goal%20of%20using%20a,prior%20data(training%20data).

Garbade, D. M. J. (2018, September 12). *Understanding k-means clustering in machine learning*. Medium. Retrieved September 25, 2021, from https://medium.com/m/global-identity?redirectUrl=https%3A%2F%2Ftowardsdatascience.com%2Funderstanding-k-means-clustering-in-machine-learning-6a6e67336aa1.

Harrison, O. (2019, July 14). *Machine learning basics with the k-nearest Neighbors ALGORITHM*. Medium. Retrieved September 25, 2021, from https://medium.com/m/global-identity?redirectUrl=https%3A%2F%2Ftowardsdatascience.com%2Fmachine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761#:~:text=Summary-,The%20k%2Dnearest%20neighbors%20(KNN)%20algorithm%20is%20a%20simple,that%20data%20in%20use%20grows.

Learn naive BAYES Algorithm: Naive Bayes Classifier examples. Analytics Vidhya. (2021, August 26). Retrieved September 25, 2021, from https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/.

Pant, A. (2019, January 22). *Introduction to logistic regression*. Medium. Retrieved September 25, 2021, from https://medium.com/m/global-identity?redirectUrl=https%3A%2F%2Ftowardsdatascience.com%2Fintroduction-to-logistic-regression-66248243c148#:~:text=Logistic%20Regression%20is%20a%20Machine,on%20the%20concept%20of%20probability.&text=The%20hypothesis%20of%20logistic%20regression,function%20between%200%20and%201%20.