Webster UNIVERSITY

GEORGE HERBERT WALKER SCHOOL
OF BUSINESS & TECHNOLOGY

DATA ANALYTICS

TERM PROJECT

FALL 2021

CSDA 6010 DATA ANALYTICS PRACTICUM

Case 1: Predicting Corporate Bankruptcy

Case 2: Classifying Book Sales Market

Case 3: Classifying Home Equity Loan Applicants

Developed by: Chau Hai Phuong Nguyen

Instructor: Dr. Ali Ovlia

Preface

In August 2020, I, Chau Hai Phuong "Mandy" Nguyen, started pursuing my graduate program in Data Analytics at Webster University in Saint Louis, Missouri, U.S.A. Going through many obstacles and achievements, as well as gaining more self-awareness of personal ability and capacity, this final practicum project reflects the hard work and knowledge I have gained throughout my nearly two-year journey of learning. I am grateful to have this wonderful opportunity to learn and to work on this four-month course (from August to December of 2021), organized by Webster University and under the instruction of my head professor, Dr. Ali Ovlia.

The project includes three study cases in the Financial and Marketing fields, which require using machine learning algorithms to find the best models for classifying target variables for prediction and classification purposes on future data sets. In particular, **Case 1 - Predicting Corporate Bankruptcy** - requires spending time to study financial ratios and select the best combinations among those attributes to classify most accurately bankrupt firms in order to use the best models to find the future potential firms to go bankrupt. The mission of **Case 2 - Classifying Book Sales Market** - is to find the best model to find a list of customers by classifying the ones who are most likely interested in buying the book in question, using the customers' interaction history with the book club, and then use that list to send mailing marketing to those potential future customers with similar profile features. The goal of **Case 3 - Classifying Home Equity Loan Applicants** – is to use the historical data of the loanees who paid and defaulted on home equity loans, and choose the best models to classify risky profiles so as to not approve loans for similar future portfolios of customers.

Personally, I believe the greatest achievement in this project is not to find the best models but to learn and enjoy the closest journey of real-world analytics processes. Despite many data mining methods, I used three machine-learning algorithms of choice including Logistic Regression, K-Nearest Neighbor, and Neural Networks throughout three cases with some other industrial analysis methods depending on each case. I found out that for Case 1, Neural Networks performed the best while K-Nearest Neighbor surpasses the other algorithms in both Cases 2 and 3. However, the learning is much more than the results as the first case shapes my understanding of an analysis paper structure, as well as, working effectively with coding R by using generic functions to be able to fit 69 models in less time-consuming. Case 2 opened up another view of using industrial expertise in analysis by classifying using RFM analysis – one of the famous methods in marketing fields. Case 3, the one with the most realistic data set allowed me to combine all empirical knowledge learned from two previous cases to sharpen my skills in data wrangling, dealing with unclean (missing values, null values, imbalanced data set, outliers, etc.), and presenting my analysis in the most captivated ways possible. The entire project was conducted using the R programming language in the RStudio integrated programming environment application, and therefore, it also upgrade my R usage skills to the highest confidence levels.

This documentation - my proud achievement, which includes one file of three analyses and one file of R codes used in the entire project, is an appreciation for my professor who led me through the journey from the beginning, as well as my supportive loved ones. It has been a great experience learning from my professors, classmate, and self-study. I hope this will be a successful reference for my future work.

Predict Cooperate Bankruptcy by Using Machine Learning Models

Case Study on Predicting Corporate Bankruptcy

By Chau Hai Phuong Nguyen

George Herbert Walker School of Business & Technology, Webster University

CSDA 6010: Data Analytics Practicum

Dr. Ali Ovlia

December 17, 2021

List of Tables

List of Figures

List of Equations

Table of Contents

Predict Cooperate Bankruptcy by Using Machine Learning Models

Bankruptcy Prediction has been facing several issues in the financial analysis practice due to two major problems including the lack of cash receipts and disbursement records and misinterpretation of historical financial disclosure information due to the constant development and reconstruction of accounting standards.

To solve the major problem in the business world, which is analyzing any firm's health and predicting its potential of bankruptcy without the aforementioned hassles, this analysis is conducted to learn and determine which financial indicators are highly capable of signaling a company to a foreseeable diminishment. From then, it is desirable to discover the best statistical models which result in the highest evaluation performance for future prediction analysis.

The dataset in use has financial data available on the Compustat[1] Research tape with 66 bankrupted companies in the manufacturing or retailing industry during the period from 1970 to 1982. With each failed firm, a healthy firm with similar sizes and backgrounds will be added for comparison which makes up the total number of observed firms to 132. The assigned outcome variable for the study is whether a company is bankrupt or non-bankrupt, while the other potential predictor variables that correlated to the outcome are fundamental financial indicators that meticulously computed by Moody's Industrial Manual with the list of 24 ratios.

The analysis undergoes a typical attributes analysis classification process including pre-processing data which involves cleaning; exploring and reducing dimensions;

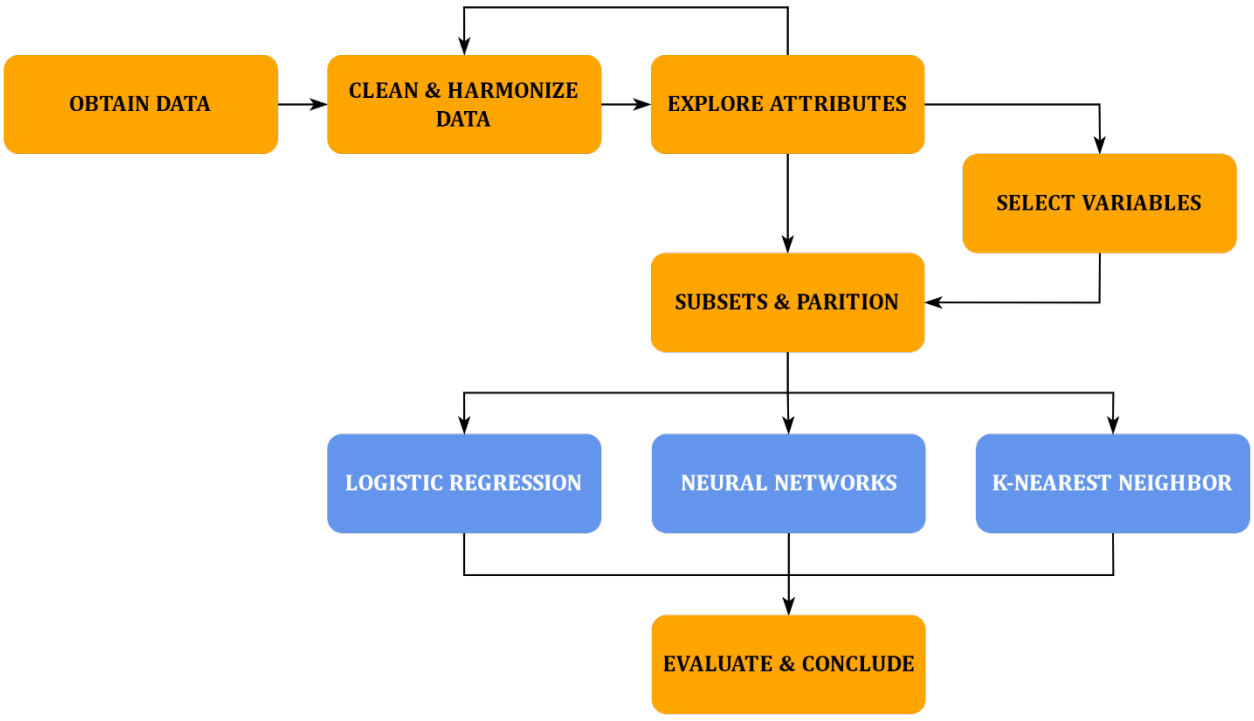
[1] https://www.investopedia.com/terms/c/compustat.asp

partitioning data into subsets for training and test purposes; using supervised learning techniques to choose the most suitable classifiers; fitting data into models; evaluating each model and selecting the best ones for this particular dataset; concluding the analysis. This analysis uses R version 4.1.1 as the main programing language and Rstudio version 1.4.1717 to compute and demonstrate all analytic steps which are summarized in Figure 1.

This analysis is divided into four parts as follows:

- **Introduction**: provides the business problem and overview; the data set, and the methods used for analysis.
- **Analysis**: provides five parts including data pre-processing, attribute analysis, dimensionality reduction, data partitioning, and machine learning models.
- **Results**: provides the final results of the analysis.
- **Conclusion**: provides the analyst view of the data set, analysis, and suggestions for improvement.

Figure 1: Diagram of Analysis Process



1. Introduction

1.1. Business Understanding

The initial step to begin this analysis is to understand the domain problems which is in the financial industry. The main goal is to classify which ratios are most influential and contribute greatly to a company's failure, and the best-supervised machine learning model for classification and prediction for future potentially bankrupt firms.

The list of 66 failed firms, provided by Dun and Bradstreet, during 1970 and 1982 in comparison with 66 other healthy ones is taken into account. The dataset was thoughtfully designed by the industry's professionals to consider 13 financial variables collected throughout 18 fiscal years including assets, cash, cash flow from operations (CFFO), cost of goods sold (COGS), current assets (CURASS), current debt (CURBEBT), total debts (DEBTS), income (INC), income plus depreciation (INCDEP), Inventory (INV), receivables (REC), sales (SALES), and working capital from operations (WCFO).

Then, the dataset was created by using 13 mentioned financial indicators, collected from Compustat and Moody's Industrial Manual, and dividing against each other to compute 24 fractional ratios for the list of 132 studied firms. The detailed list of the financial indicators and computed ratios can be found in Tables 1, 2, and 3.

Table 1: Identifier Attributes

Identifiers	Definition	Original Data Type
No	An assigned firm number, matched firms are given the same number	Integer
D	0 for failed firm, 1 for healthy firms.	Integer
YR	Year of bankruptcy for the failed firm.	Integer

Table 2: Financial Terms of Ratio Attributes

Abbreviation	Definition
ASSETS	Total assets
CASH	Cash
CFFO	Cash flow from operations
COGS	Cost of goods sold
CURASS	Current assets
CURBEBT	Current debt
DEBTS	Total debt
INC	Income
INCDEP	Income plus depreciation
INV	Inventory
REC	Receivables
SALES	Sales
WCFO	Working Capital from operations

Table 3: Ratio Predictors

Ratio	Definition	Original Data Type
R1	CASH / CURDEBT	Numeric
R2	CASH / SALES	Numeric
R3	CASH / ASSETS	Numeric
R4	CASH / DEBTS	Numeric
R5	CFFO / SALES	Numeric
R6	CFFO / ASSETS	Numeric
R7	CFFO / DEBTS	Numeric
R8	COGS / INV	Numeric
R9	CURASS / CURDEBT	Numeric
R10	CURASS / SALES	Numeric
R11	CURASS / ASSETS	Numeric
R12	CURBENT / DEBTS	Numeric
R13	INC / SALES	Numeric
R14	INC / ASSETS	Numeric
R15	INC / DEBTS	Numeric
R16	INCDEP / SALES	Numeric
R17	INCDEP / ASSETS	Numeric
R18	INCDEP / DEBTS	Numeric
R19	SALES / REC	Numeric
R20	SALES / ASSETS	Numeric
R21	ASSETS / DEBTS	Numeric
R22	WCFO / SALES	Numeric
R23	WCFO / ASSETS	Numeric
R24	WCFO / DEBTS	Numeric

1.2. Data Mining Methods and Process

In this particular business problem of predicting bankruptcy, the analyst decides to use three main supervised machine learning algorithms including Logistic Regression, Neural Networks, and K-Nearest Neighbor, and run 69 models to find the best results using confusion matrix for model evaluation. The main software for this analysis is RStudio and the programing language is R with the support from the list of R packages which can be found in the **R Code** document accompanied with this paper.

In particular, after studying the business problem, the dataset, and research for preliminary understanding, the analysis undergoes six major data mining steps as follows:

First is to pre-process the data for any defective values or records during the collecting procedure so as to provide a clean data set for learning algorithms.

Second is to learn each and every attribute, discover their correlations to each other and to the outcome variable and their significances.

Third is to remove the insignificant attributes after the second step for model fitting.

Forth is to gather the selected attributes and divide them into sets.

Fifth is to partition the original data set into 6 sub-sets to fit into models.

Sixth is to use the data sets, attributes, and knowledge gained from previous steps to fit into three algorithms and analyze the results.

2. Analysis

Figure 2 shows the map of six steps in the analysis procedure for the bankruptcy data set. The details and explanation of each process are described as follows:

2.1. Data Pre-processing

First, the "bankruptcy.csv" file, containing the dataset of bankrupt/ non-bankrupt firms with their 24 ratios for each firm, is loaded into Rstudio by the *read.csv()* function to be ready for work with the note of existence headers which requires a little extra care with the command *header = True* to avoid undesired input errors. After that, the *View()* function is used for an overview of the whole dataset in full length which is an objective way to skim through the set. From that, it is clear that the dataset was well clean ratio values of up to two decimal digits with positive, negative, and zero numbers. There are 27 columns in total in the dataset. The first 3 columns, in order, are the identification number of firms in columns "NO", the flag of bankrupt as "0" and non-bankrupt or healthy as "1" in column "D", the year in which the data of each firm collected in column "YR", and the rest of 24 variables are ratios marked as column "R1" to "R24" which explained in detail in Table 3.

Figure 2: Diagram of Models and Approaches

Second, the data is undergone the cleaning procedure for later analysis. In this particular data, there is no missing value to pre-process. However, there are 50 zeros in total which are needed to be looked into since most of the numerators are assets indicators, so zeros might depict not profitable business throughout the years. On that note, the rows with "zero" are filtered out into a subset for in-depth investigation. Table 4 below shows that among the records with zeros in their ratio attributes, 16 records belong to bankrupt firms and 12 records belong to healthy firms. Although there is a difference of 4 records, this is not significant enough to assume the zeros in the ratio may define the bankrupt firms or vice versa. However, there is one noteworthy sign that because all of these fractional ratios are presented in two-decimal values, it is possible that the software that recorded these numbers might have rounded up any decimal numbers with more than three decimals to zeros, which misleads that the dataset has zero values. However, due to the lack of original data of the financial values and insights from the data collected, it is not possible to give any satisfactory conclusion. For the moment, zero values will remain in the dataset for later analysis with attention.

Table 4: Class Distribution of Target Variable in Ratio with "Zero" Values

	Bankrupt	**Healthy**
Total Number of Firms	16	12

Next, to confirm everything learned about the overview of the dataset is correct, the function *str()* is used in R to provide a structural overview. The computational result depicts that there are 132 observations in the dataset which represents 132 firms, 27 variables which include the first three identifiable variables with the data type of "integer" and the last 24 variables are ratio classifiers with the data type of "numeric". The details of the data type

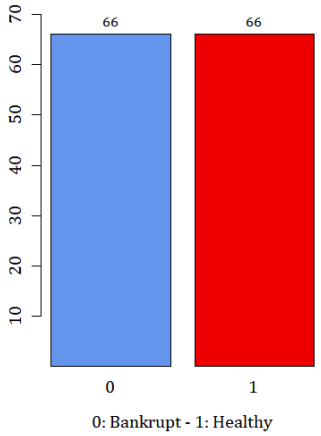
of each attribute can be found in Tables 1 and 3. For the 24 classifiers, the numerical data type is the best way to assign a decimal number from fractions. However, from common and industrial knowledge, it is understandable to believe that each value in column "D" does not have a quantitative meaning for our analysis, instead, they should be served as Boolean categorical values. However, for the moment, it is advisable to remain the data type of outcome variable "D" as the current integer for later models fitting step with the note to factor into categorical values where applicable and specify the significant value class is "0" (bankrupt) instead of "1" (healthy).

2.2. Attributes Analysis

2.2.1. Identifier Attributes

After the overview of the data set, it is necessary to investigate each and every column. The first column "NO" is simply an identification of attributes for records numbered from 1 to 132, organized in ascending order that does not have any statistical meaning to the analysis. Therefore, this column can be removed before the analyzing process.

Figure 3: Classes Distribution of Target Variable



Next, the outcome variable "D" is taken into account as visualized in Figure 3. It can be seen that within the dataset, both class "0" and "1" or "bankrupt" and "healthy" respectively are distributed with equal numbers of records. Hence, this data set is a perfectly balanced dataset that has even numbers of records for each class of expected outcome. However, in reality, it is understandable with practical experiences and common sense that

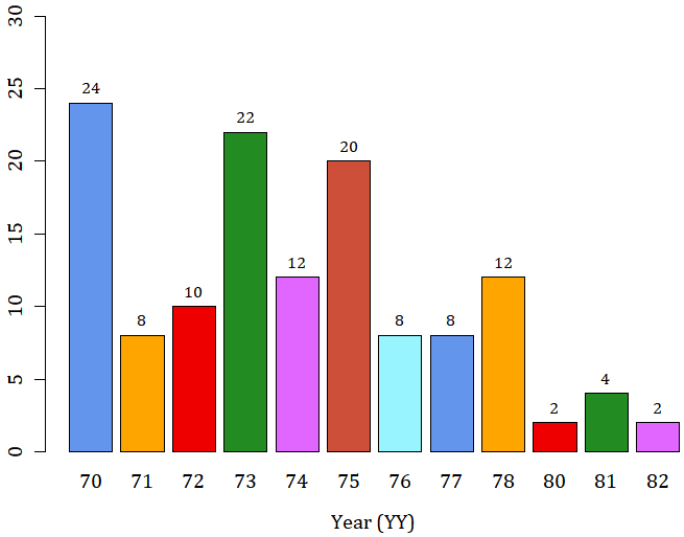
the number of bankrupt firms should be less than the number of successful firms. Therefore, this data set violates one of the criteria to be considered a "good data set" is that the sample does not represent the collective cases in question, which is a considerable element before making conclusions. Additionally, when viewing the whole data set in a spreadsheet, data is organized in the sense that all bankrupt firms are recorded within the first 66 rows and then the next 66 rows followed are the healthy ones. Without randomization among the records, the data set is considered biased which might lead to several problems when fitting into a machine learning algorithm. For this reason, the data set should be randomized before the partition step.

The last of the identifier attribute set is column "YR", describing the years that records were recorded throughout the course of 12 years from 1970 to 1982 excluding 1979 without any explanation. Table 5 below shows that the numbers of bankrupt and healthy firms are equally distributed each year. The histogram in Figure 4 shows that the year 1970 accounts for the highest number of records with 24, while the years 1982 and 1980 have the lowest with 2 records. In the beginning, it was stated that this dataset's designer intended to collect data for a prediction within two years after the most recent year of the data set which is 1982. Moreover, the biased dataset describes an ideal world where the numbers of bankruptcy and successful companies are equivalent, it is not reliable to believe this distribution of years has a significant role in the health of each company unless explained otherwise. Hence, the "YR" or year attribute shall be removed during the analysis process.

Table 5: Number of Bankrupt Firm and Healthy Firm grouped by Year

Year	Bankrupt Firm	Healthy Firm	Total Firm
1970	12	12	24
1971	4	4	8
1972	5	5	10
1973	11	11	22
1974	6	6	12
1975	10	10	20
1976	4	4	8
1977	4	4	8
1978	6	6	12
1980	1	1	2
1981	2	2	4
1982	1	1	2

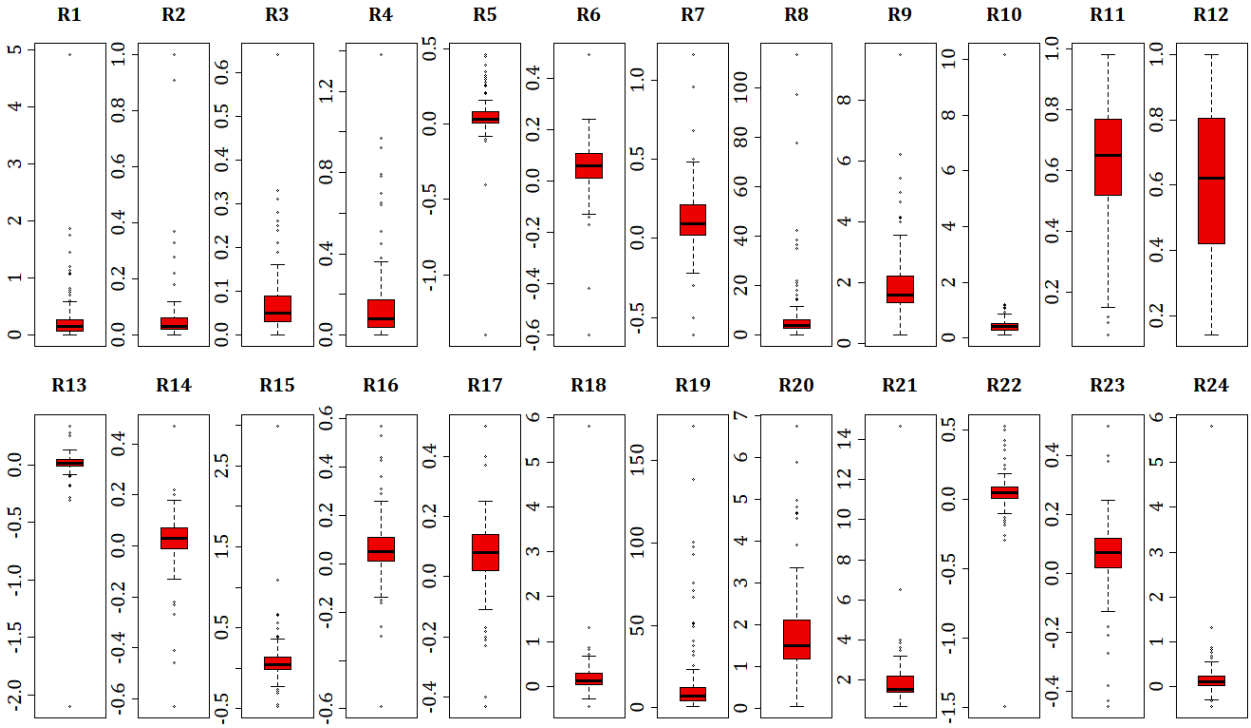
Figure 4: Frequency of Year



2.2.2. Ratio Attributes

Overall, Figure 5 shows that 24 ratios do not have a significant number of outliers to handle, but only the value scales might be taken under consideration when cleaning data.

Figure 5: Combined Box Plots of 24 Ratio Attributes



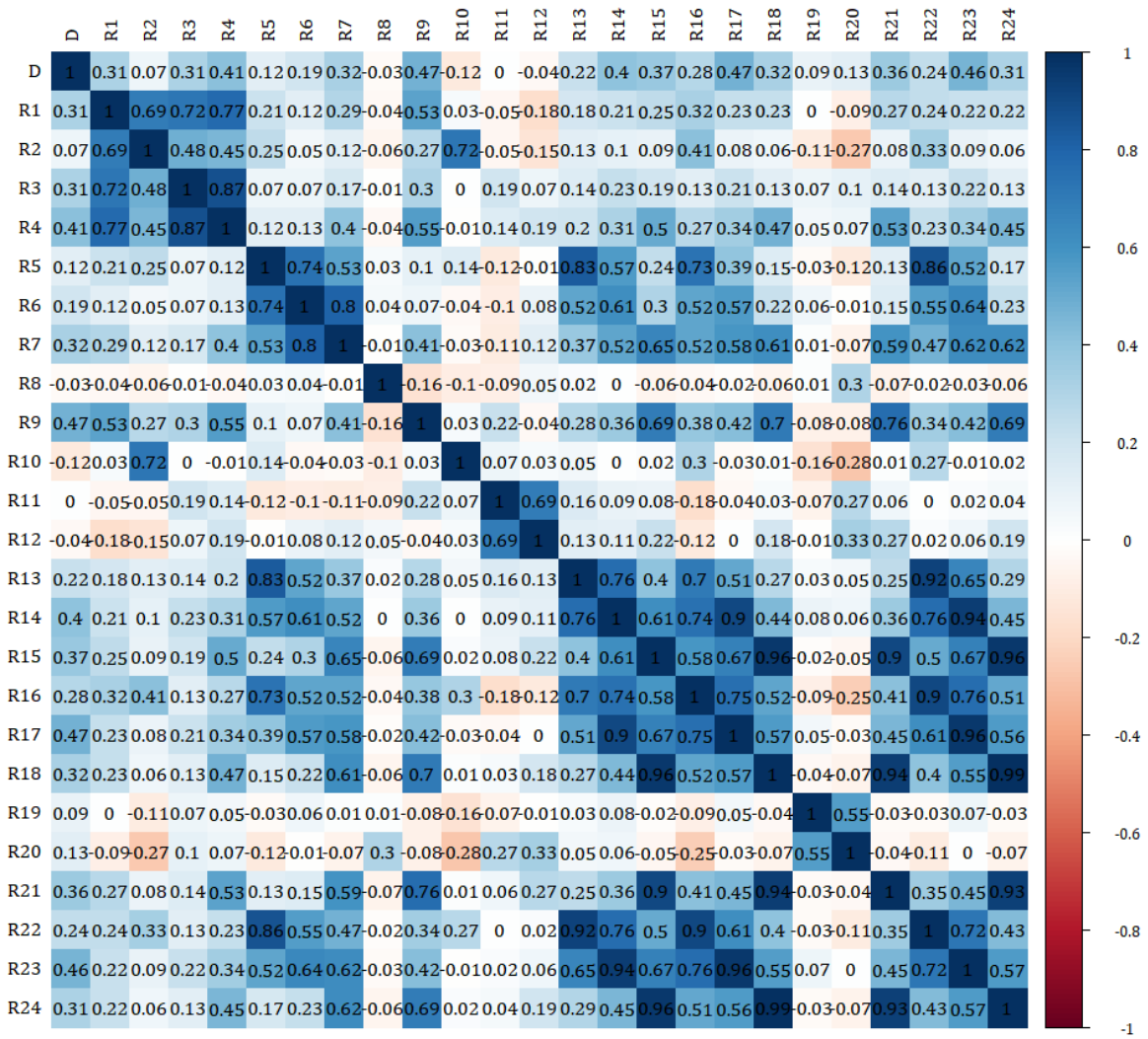
2.3. Dimensionality Reduction

While dealing with a large set of classifiers, a common practice is to analyze the ratios and select only the most significant attributes that have a high impact on the outcome.

2.3.1. Correlation Plot

The first method used is a correlation plot between the outcome variable and ratio attributes, and between the attributes to each other. Figure 6 illustrates the coefficient values of the mentioned relations. The goal of using a correlation plot is to draw association strength between attributes in question and then try to compare them in pairs.

Figure 6: Correlation Matrix between Attributes



With the pairs that have a strong association (coefficient is equal to and greater than 0.5), the attribute with weaker association with the outcome variable will be removed. This step will be repeated subsequently with the next pairs until the last pair, and the last attributes remaining shall be those that are independent and have strong correlations with the outcome. The reason for this method of the selection process is because, the pairs that have a strong correlation, will provide a similar amount of information about the outcome which makes it redundant to keep both attributes in each pair. After finishing this step, it remains four independent ratios that have weak to the strong association with the outcome variable as follows:

Table 6: Significant Independent Ratios Selected from Correlation Matrix

Ratio	Correlation Coefficient	Association with D	Definition	Description
R9	0.47	Average	CURASS/ CURDEBT	Current Asset / Current Debt
R10	-0.12	Weak	CURASS / SALES	Current Asset / Sales
R17	0.47	Average	INCDEP / ASSETS	Income plus depreciation / Assets
R20	0.13	Weak	SALES / ASSETS	Sales / Assets

The overall meaning of these ratios are as follows:

- R9 - Current Asset / Current Debt[2]: is a liquidity ratio that measures a company's ability to pay short-term obligations.
- R10 - Current Asset / Sales[3]: indicates how efficiently a company is using its current assets to generate revenue.
- R17 - Income plus depreciation / Assets: describes the proportion of income generated per unit of assets.
- R20 - Sales / Assets[4]: indicates the return on sales.

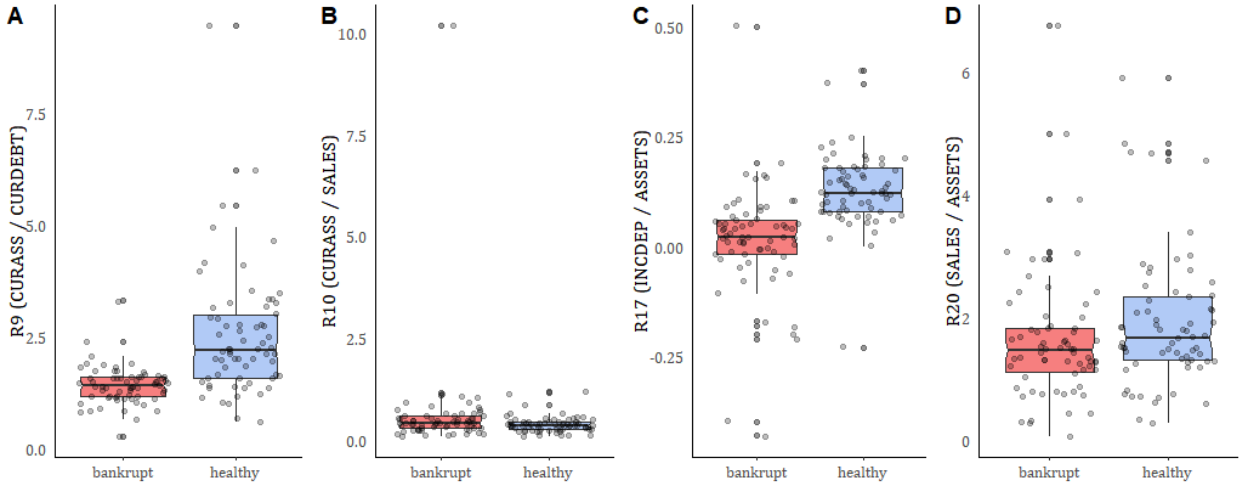
[2] https://www.investopedia.com/terms/c/currentratio.asp
[3] https://studyfinance.com/sales-to-current-assets-ratio/
[4] https://www-jstor-org.library3.webster.edu/stable/3666109?read-now=1&seq=7#page_scan_tab_contents

In order to understand more in-depth about these ratios, plotting them into a box plot is chosen as the main tools for the next step.

2.3.2. Box Plot

Figure 7 shows the figure combined four boxplots of the mentioned ratios including R9, R10, R17, and R20. In each boxplot of each ratio, the red whiskers depict values of each bankrupt firm, while the blue ones depict that of healthy firms. Overall, it is clear that healthy firms tend to have longer whiskers with ratio values varying from 1.5 to 3.75, while the failed firms that have shorter whiskers vary between 1.0 to 1.5 with a few exceptional outliers. Hence, it is plausible to believe that bankrupt firms are associated with a low ratio of the four in question. This belief can also be supported in practice since most of the numerators of these ratios are assets while denominators are liabilities, so the higher the ratios tell that the firms possess more assets than a liability.

Figure 7: Box Plots of R9, R10, R17, R20 ratios grouped by Target Variable



2.4. Data Partitioning

Data partitioning is an important process to divide the data into smaller subsets for training the models, and testing the models with new sets that the machine learning

algorithms have never seen. For this particular dataset, which has several concerning issues that affect partition decisions, it is essential to address the issue to clarify the partitioning decisions.

First, it is the insufficient numbers of records with only 132 that may be a problem with model fitting. Hence, the first approach is to use the whole randomized dataset for training the model and use and portion of 40% of it for testing. Henceforth, the training set of this approach shall be called *Original Dataset.*

Second, this approach shall use the conventional method of partitioning 60% of the original dataset into the training set and the rest of 40% shall be for the testing set. Henceforth, the training set of this approach shall be called the *Original Training Data set.*

Third, this approach shall use the normalization technique to transform the ratio values of the original dataset to the range of 0 to 1 using the min-max method with X' is the scaled data point, X is the original data point, Min(X) is the lowest value and Max(X) is the highest value in the attribute. The function for the min-max normalization technique is as follows:

$$\textbf{Equation 1: } X' = \frac{X - Min(X)}{Max(X) - Min(X)}$$

This dataset shall be called *Normalized Dataset*. The reason for using this dataset is that, even though the values seem to be relatively close to each other, the fact is that distance from the lowest data point to the highest can vary between 0.001 to over 170.000. Moreover, other potential machine learning algorithms also required normalized datasets for better performance.

Fourth, similarly to the second approach, this approach partitioned 60% of the normalized dataset for training and 40% for testing. Henceforth, this training set shall be called *Normalized Training Set.*

Fifth, the dataset in this approach shall be called *Reduced Dataset* using only four selected ratios (R9, R10, R17, and R20) of 132 observations from the dimensionality reduction process to fit into models and evaluate on a random 40% of its total records.

Sixth, similar to the second and fourth, this last approach uses 60% of the *Reduced Dataset* for training and 40% for testing. Henceforth, this dataset shall be called *Reduced Training Dataset.*

Respectively, for the testing dataset of six approaches, the names shall be called, *Original Testing Dataset, Normalized Testing Dataset, Reduced Testing Dataset*. In the following step of training the models, improving methods shall be applied to create better sets of each model, the improved sets of each approach shall be named respectively as *Improved Original Dataset, Improved Original Training Dataset, Improved Normalized Dataset, Improved Reduced Dataset,* and *Improved Reduced Training Dataset*. The list of 15 partitioned datasets is mentioned.

2.5. Classifier models selection

As explained in the Dimensionality Reduction and Data Partitioning steps, depending on each different approach, the set of classifiers can divide into two sets: (1) including 24 ratio attributes that can be original or normalized; (2) including four ratios after applying reduction methods that can also be original or normalized. However, new sets of classifiers may be created while fitting and improving models regarding the models of use in the following step.

2.6. Model Training, Performance, and Evaluation

After studying the preliminary data and business knowledge, cleaning data, and analyzing attributes, the primary step is to fit the pre-process data into machine learning models to find the best ones for bankruptcy classification. Three main algorithms used in this analysis are Logistic Regression, Neural Networks, and K-Nearest Neighbor. Subsequently, each algorithm shall process six partitioned sub-sets of data listed in Appendix C. After the first time fitting each sub-set of data into each model, certain modifications shall be made for improving performance, which shall be presented in detail in the following sections. Then, data shall be fitted into new adjusted models until finding the acceptably satisfied better ones. For evaluating the models, the confusion matrix is the main tool for this classification problem. The sequence of measures' priority order is accuracy, sensitivity, and specificity with the positive class is "0" or "bankrupt". The reason for prioritizing sensitivity over specificity is because based on financial practice, it is more important to find the correct potentially bankrupt firms, and it will damage business decisions if a company that is supposed to go bankrupt is out of the radar and diagnosed as "healthy". In total, there are 69 different models have been created by analysts' generic functions to find the best among the data sets and machine learning algorithms.

2.6.1. Logistic Regression Model

In this analysis, there are 24 regression models fitted to six data set so as to select the top three models with the highest evaluation measures as synthesized as in Table 7 as follows:

Table 7: Logistic Regression Evaluation Measures

Models	Accuracy	Sensitivity	Specificity
Improved Original Dataset (step = backward = both) *Retained Ratios: R3, R5, R6, R9, R10, R16, R17, R18, R22, R23, R24*	0.8679	0.9286	0.8000
Normalized Dataset	0.9434	0.9714	0.8889
Reduced Dataset	0.9245	0.9714	0.8333
Improved Reduced Dataset (step = backward = both) *Retained Ratios: R9, R10, R17*	0.9245	0.9714	0.8333

In detail, each of the six data sets is fitted into logistic regression using three generic functions created by the analyst including model and evaluation function, improving models with stepwise methods function, and fitting and evaluating new sets of attributes function. With these manually created functions, it allows the analyst to swiftly run over 24 regressions and receive instant results.

The process of fitting data sets into a logistic regression, first, is initiated by fitting the whole working data set into the model and then evaluating the accuracy, sensitivity, and specificity of the first model. Then, the same data set undergoes a stepwise method with three different steps including backward, forward, and both. The stepwise method will automatically run through many AIC cycles to output the results of best attributes among 24 ratios to fit in new logistic regression models. Next, the three new models generated from new sets of attributes selected by the stepwise method, together with the first fitted model are compared against each other to choose the best one among them. This process repeats for all six data sets.

As a result, the best logistic regression model is the one that is fitted by the normalized data set with all 24 ratios, which gives an accuracy of 94.34%, sensitivity of 97.14%, and specificity of 83.33%. The runner-up is the reduced data set with only four attributes selected from the dimensionality reduction step including R9, R10, R17, and R20, with an accuracy of 92.45%, specificity of 97.14%, and specificity of 83.33%. It seems that

after running the stepwise function to improve the reduced dataset and removing R20, the three new data set of the reduced data set still give the same results. Hence, R20 might not have a significant role that can affect the outcome variable. With the original dataset, the results of the best model ranked last among the Original, Reduced, and Normalized datasets after improving by running backward and both stepwise methods and selecting the 11 most significant attributes. The result of the improved original dataset is 86.79% of accuracy, 92.86% of sensitivity, and 80% of specificity. The detailed results of 24 logistic regression models ran can be found in Table 8.

2.6.2. K-Nearest Neighbor

K-Nearest Neighbor (KNN) is one of the most classic and simple but effective supervised machine learning algorithms that operate based on the premise that any data points close to each other have certain relations and should be in the same class. Among the three algorithms, KNN ranked third in terms of results delivery after running 15 models and the best ones among the data sets are selected and displayed in Table 9 below.

Table 8: K-Nearest Neighbor Evaluation Measures

Models	Accuracy	Sensitivity	Specificity
Improved Original Dataset (k = 1)	1.0000	1.0000	1.0000
Improved Original Dataset (k = 3)	0.8113	0.8571	0.7600
Improved Normalized Dataset (k = 1)	1.0000	1.0000	1.0000
Improved Normalized Dataset (k = 2)	0.8868	0.8286	1.0000
Improved Reduced Dataset (k = 1)	1.0000	1.0000	1.0000
Improved Reduced Dataset (k = 3)	0.9057	0.8857	0.9444

Table 9: Full Summary of Logistic Regression Models Evaluation Measures

Models	Accuracy	95% CI	No Info. Rate	P-Value	Kappa	Sensitivity	Specificity
Original Dataset							
Original Dataset	0.8491	(0.7241, 0.9325)	0.5283	9.065e-07	0.6945	0.9286	0.7600
Improved Original Dataset (step = backward = both)	0.8679	(0.7466, 0.9452)	0.5283	1.709e-07	0.7333	0.9286	0.8000
Retained Ratios: R3, R5, R6, R9, R10, R16, R17, R18, R22, R23, R24							
Original Training Dataset	0.6415	(0.498, 0.7686)	0.5283	0.06426	0.2792	0.6786	0.6000
Improved Original Training Dataset (step = backward = both)	0.6604	(0.5173, 0.7848)	0.5283	0.03587	0.3244	0.6071	0.7200
Retained Ratios: R2, R3, R5, R6, R9, R10, R12, R14, R15, R16, R19, R21							
Normalized Dataset							
Normalized Dataset	0.9434	(0.8434, 0.9882)	0.6604	1.006e-06	0.8721	0.9714	0.8889
Improved Normalized Dataset (step = backward = both)	0.9057	(0.7934, 0.9687)	0.6604	3.58e-05	0.7808	0.9714	0.7778
Retained Ratios: R3, R5, R6, R9, R10, R16, R17, R18, R22, R23, R24							
Normalized Training Dataset	0.6415	(0.498, 0.7686)	0.6604	0.672620	03201	0.5429	0.8333
Improved Normalized Training Dataset (step = backward = forward = both)	0.6415	(0.498, 0.7686)	0.6604	0.672620	03201	0.5429	0.8333
Retained Ratios: R5, R6, R7, R10, R11, R12, R14, R16, R17, R18, R22, R24							
Reduced Dataset							
Reduced Dataset	0.9245	(0.8179, 0.9791)	0.6604	6.766e-06	0.8271	0.9714	0.8333
Improved Reduced Dataset (step = backward = both)	0.9245	(0.8179, 0.9791)	0.6604	6.766e-06	0.8271	0.9714	0.8333
Retained Ratios: R9, R10, R17							
Reduced Training Dataset	0.8868	(0.7697, 0.9573)	0.6604	0.0001552	0.7605	0.8571	0.9444
Improved Original Training Dataset (step = backward = both)	0.9057	(0.7934, 0.9687)	0.6604	3.58e-05	0.7979	0.8857	0.9444
Retained Ratios: R9, R10, R17							

In this model, with the support from the "KNN" package for models fitting and "caret" package for evaluation using confusion matrix, the analyst manually created two functions, the first one is the find the value of k that gives the highest accuracy among a given range of k values and the plot the values of k inline chart for better visual analysis; next, after finding the best values of k, the analyst created a function to fully fit the data set with the selected k and evaluate the results.

For a starter, the analyst used a common practice when using KNN algorithms the find the assumably best k which is to take the square root of total observation, in this case, is 132, which is 11.5, and use as the first k. Then, the analyst uses the k with the highest accuracy in the table of k values to fit into new models for improvement.

In Table 9, there are six highest results with two results for each data set and all of the results derived from the data set with all observations (not the one using 60% training sets). The reason for choosing 6 results is because, with k = 1, all three full data set to give 100% of the three evaluation measures. Hence, the models are unreliable and unrealistic, and the reasons for this might because of the insufficient number of observations due to the small dataset with only 132 records. For this reason, the analyst decides to choose the next best things with the improved results of other k values. With this method, the model ranked best is the improved reduced data set with k = 3 that gives 90.57% of accuracy, 88.57% of sensitivity, and 94.44% of specificity. The second best is the improved normalized data set with k = 2 that gives 88.68% of accuracy, 82.86% of sensitivity, and 100% of specificity. The last is the improved original dataset with k = 3 gives 81.13% of accuracy, 85.71% of sensitivity, and 0.76% of specificity. The full results of 15 KNN models can be found in Table 10.

Table 10: Full Summary of K-Nearest Neighbor Models Evaluation Measures

Models	Accuracy	95% CI	No Info. Rate	P-Value	Kappa	Sensitivity	Specificity
Original Dataset							
Original Dataset (k = 11)	0.7358	(0.5967, 0.8474)	0.5283	0.00161	0.4723	0.7143	0.7600
Improved Original Dataset (k = 1)	1	(0.9328, 1)	0.5283	2.055e-15	1	1.0000	1.0000
Improved Original Dataset (k = 3)	0.8113	(0.6803, 0.9056)	0.5283	1.709e-05	0.6198	0.8571	0.7600
Original Training Dataset (k = 11)	0.7925	(0.6589, 0.8916)	0.6604	0.02629	0.5772	0.7429	0.8889
Improved Original Training Dataset (k = 3)	0.6981	(0.5566, 0.8166)	0.5283	0.008932	0.389	0.7857	0.7857
Normalized Dataset							
Normalized Dataset (k = 11)	0.7925	(0.6589, 0.8916)	0.6604	0.02629	0.5772	0.7429	0.8889
Improved Normalized Dataset (k = 1)	1	(0.9328, 1)	0.6604	2.812e-10	1	1.0000	1.0000
Improved Normalized Dataset (k = 2)	0.8868	(0.7697, 0.9573)	0.6604	0.0001552	0.7665	0.8286	1.0000
Normalized Training Dataset (k = 11)	0.7547	(0.6172, 0.8624)	0.6604	0.0936193	0.5348	0.6286	1.0000
Improved Normalized Training Dataset (k = 1)	0.6792	(0.5368, 0.8008)	0.6604	0.448508	0.4053	0.5429	0.9444
Reduced Dataset							
Reduced Dataset (k = 11)	0.8302	(0.702, 0.9193)	0.6604	0.004923	0.6362	0.8286	0.8333
Improved Reduced Dataset (k = 1)	1	(0.9328, 1)	0.6604	2.812e-10	1	1.0000	1.0000
Improved Reduced Dataset (k = 3)	0.9057	(0.7934, 0.9687)	0.6604	3.58e-05	0.7979	0.8857	0.9444
Reduced Training Dataset (k = 11)	0.8302	(0.702, 0.9193)	0.6604	0.004923	0.6362	0.8286	0.8333
Improved Reduced Training Dataset (k = 4)	0.8113	(0.6803, 0.9056)	0.6604	0.01202	0.6203	0.7429	0.9444

2.6.3. Neural Networks

As an advanced and complex supervised machine learning algorithm, Neural Networks gave the best results among the three algorithms. In this step, the analyst used the help of "neuralnet" and "caret" packages to manually create one functions that can fit the dataset into neural networks function, predict on the test data set, evaluate using confusion matrix and plot the neural network topology. With this one function, the analyst was able to quickly fit 30 models in total, with 5 different improving models among the 6 data sets, starting with only one hidden layer and one node, then changing to one hidden layer and two nodes, one hidden layer and three nodes, two hidden layers and 1:2 nodes each layer, and two hidden layers and 3:2 nodes each layer. The results of the best model within the original, normalized, and reduced data sets are concluded in Table 11 below.

Table 11: Neural Networks Evaluation Measures

Models	Accuracy	Sensitivity	Specificity
Improved Original Dataset (hidden: layer = 1; node = 3)	0.9623	0.9643	0.9600
Improved Normalized Dataset (hidden: layer = 1; node = 3)	0.9811	0.9714	1.0000
Reduced Dataset (hidden: layer = 1; node = 1)	0.9623	0.9714	0.9444

Overall, the normalized dataset after improving using three nodes gave the best results with 98.11% accuracy, 97.14% sensitivity, and 100% specificity. Both of the improved original data set with three nodes and reduced dataset with one node gave the same accuracy of 96.23% with the slighter higher sensitivity from the reduced dataset with 97.14% and 96.4% from the improved original dataset, while with specificity the former's slightly lower with 94.44% while the latter's is 96%. However, since sensitivity is prioritized over specificity, the reduced dataset ranks second among the three. The full results of 30 Neural Networks models can be found in Table 12 below.

Table 12: Full Summary of Neural Networks Models Evaluation Measures

Models	Accuracy	95% CI	No Info. Rate	P-Value	Kappa	Sensitivity	Specificity
Original Dataset							
Original Dataset (hidden: layer = 1; node = 1)	0.9057	(0.7934, 0.9687)	0.5283	3.765e-09	0.8095	0.9643	0.8400
Improved Original Dataset (hidden: layer = 1; node = 3)	0.9623	(0.8702, 0.9954)	0.5283	2.356e-12	0.9243	0.9643	0.9600
Original Training Dataset (hidden: layer = 1; node = 1)	0.7547	(0.6172, 0.8624)	0.5283	0.000599	0.5068	0.7857	0.7200
Improved Original Training Dataset (hidden: layer = 1; node = 2)	0.8113	(0.6803, 0.9056)	0.5283	1.709e-05	0.6246	0.7500	0.8800
Normalized Dataset							
Normalized Dataset (hidden: layer = 1; node = 1)	0.9623	(0.8702, 0.9954)	0.6604	1.104e-07	0.9159	0.9714	0.9444
Improved Normalized Dataset (hidden: layer = 1; node = 3)	0.9811	(0.8993, 0.9995)	0.6604	7.945e-09	0.9585	0.9714	1.0000
Normalized Training Dataset (hidden: layer = 1; node = 1)	0.7736	(0.6379, 0.8772)	0.6604	0.051967	0.5552	0.6857	0.9444
Improved Normalized Training Dataset (hidden: layer = 1; node = 2)	0.7925	(0.6589, 0.8916)	0.6604	0.02629	0.5874	0.7143	0.9444
Reduced Dataset							
Reduced Dataset (hidden: layer = 1; node = 1)	0.9623	(0.8702, 0.9954)	0.6604	1.104e-07	0.9159	0.9714	0.9444
Improved Reduced Dataset	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Reduced Training Dataset (hidden: layer = 1; node = 1)	0.9245	(0.8179, 0.9791)	0.6604	6.766e-06	0.8362	0.9143	0.9444
Improved Reduced Training Dataset (hidden: layer = 1, node = 2)	0.9434	(0.8434, 0.9882)	0.6604	1.006e-06	0.8755	0.9429	0.9444

3. Results

In conclusion, Neural Networks shows the best results, Logistic Regression is the second, the KNN is the third in terms of overall accuracy. The summary by the ranking of the three best models of each algorithm is displayed in Table 13. Although Neural Networks and KNN have the high result, there is one notice that these two algorithms have lower sensitivity compared to their own specificities, which in this case is concerning because it is more crucial to classify the correct bankrupted companies which defined by higher sensitivity instead of specificity.

Table 13: Summary of Best Models of Machine Learning Algorithms by Ranking

Rank	Algorithm	Models	Accuracy	Sensitivity	Specificity
1	Neural Networks	Improved Normalized Dataset *(hidden: layer = 1; node = 3)*	0.9811	0.9714	1.0000
2	Logistic Regression	Normalized Dataset	0.9434	0.9714	0.8889
3	k-Nearest Neighbor	Improved Reduced Dataset *(k = 3)*	0.9057	0.8857	0.9444

4. Conclusion

Overall, with this business problem of classifying potential bankrupt companies, it is possible to use three machine learning algorithms including Neural Networks, Logistic Regression, and k-Nearest Neighbor to find the best models for the job with over 90% of accuracy. However, the analyst is not confident in these results because of the lack of several elements during the process of designing the analysis and collecting data, which might be taken into account to reduce concerns during analyzing process. The suggestions for improvement comprise three points as follows:

First, the dataset designer should provide essentially provide more information of business knowledge as well as detailed information and description of the classifier ratios. The analyst was not given sufficient context of each financial indicators, reasons for

selections, and explanation of the divided fractional ratios for a good understanding to reduce the dimensions for fitting the best models. Additionally, the ratios are suggested to be given in the form of at least 4 decimals points so the machine can compute better calculations and avoid zeros values because of software rounding up settings.

Second, the dataset makes a crucial bias in data analytics is displaying the data without randomization. In this dataset, all records of bankrupt firms are displayed in the first 66 rows stacked of the healthy firms. With this type of display, machine learning algorithms are manipulated and cannot be objective while computing and fitting these records into models. Hence, it is possible that the data set collector, instead of displaying the order of attribute "D", may display the dataset in either ascending or descending order of year, as long as the records are not in order of the outcome attribute.

Third, the major of this dataset is the perfectly balanced number of classes within the outcome variable "D". As in reality, the number of bankrupt firms should be lower than the healthy firms, this data set created an ideal sample that does not represent the real world. Therefore, the analyst suggests that instead of collecting firms from various industries but the collectors did not include the industries which make the planned "industry" attribute useless, the collector may consider collecting all companies from one industry within a diverse and representative region such as the United States of America and then narrow down the time range. With this way of collecting data, the set should be more representative with fewer bankrupt firms than the healthy ones.

References

Baesens, B., Roesch, D., & Scheule, H. (2018). Credit risk analytics: Measurement techniques, applications, and examples in Sas. WILEY.

Dinov, I. D. (2018). Data Science and Predictive Analytics: Biomedical and health applications using R. Springer.

Scheule, H., Rösch Daniel, & Baesens, B. (2017). Credit risk analytics: The R companion. CreateSpace, a DBA of On-Demand Publishing, LLC.

Shmueli, G. (2018). Data mining for Business Analytics: Concepts, techniques, and applications in R. John Wiley & Sons.

https://www.investopedia.com/terms/c/compustat.asp

https://library.ulethbridge.ca/apa7style/formatting_guidelines

https://www.statmethods.net/stats/descriptives.html

https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html

https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/

http://www.sthda.com/english/wiki/ggplot2-themes-and-background-colors-the-3-elements

https://www.lexingtonlaw.com/education/derogatory-marks

https://www.socr.umich.edu/people/dinov/courses/DSPA_notes/05_DimensionalityReduction.html

https://www.investopedia.com/terms/s/stepwise-regression.asp

Classify Customers for Mail Marketing by Using RFM and Machine Learning Models

Case Study on Classifying Book Sales Market

By Chau Hai Phuong Nguyen

George Herbert Walker School of Business & Technology, Webster University

CSDA 6010: Data Analytics Practicum

Dr. Ali Ovlia

December 17, 2021

List of Tables

List of Figures

List of Equations

Table of Contents

Classify Customers for Direct Mail Marketing by Using Machine Learning Models

Nowadays, since the rise of online e-commerce businesses that offer user-friendly and easy-to-use methods for customers to conveniently search and shop for any particular products in general and books in particular, Amazon and many other websites do not spare any market shares for conventional physical bookstore retailers.

However, a distinctive business model followed suit by Charles Book Club (CBC), subscription-based book clubs, seems to survive the dominance of Amazon with concerning suffers. Book clubs model enterprise offers different types of memberships with exclusive benefits, but the general idea is each customer who signs up for the monthly subscription gets more competitive prices on the titles of choice within the book clubs' catalogs. Although persisted in the business, Charles Book Club is facing drops in revenue which require the assistance of marketing analysis to understand and solve the problem.

Charles Book Clubs was established in December 1986, positioning at a book enterprise that understands customers and tailors most suitable titles that fit their customers' needs. CBC's main method to sell books is direct marketing through a variety of channels including media advertising such as TV, magazines, and newspaper or mailing. CBC strictly defines itself as a distributor, not a publisher, therefore, their most asset is the customer database which includes over 50,000 readers mostly acquired by magazines advertisements.

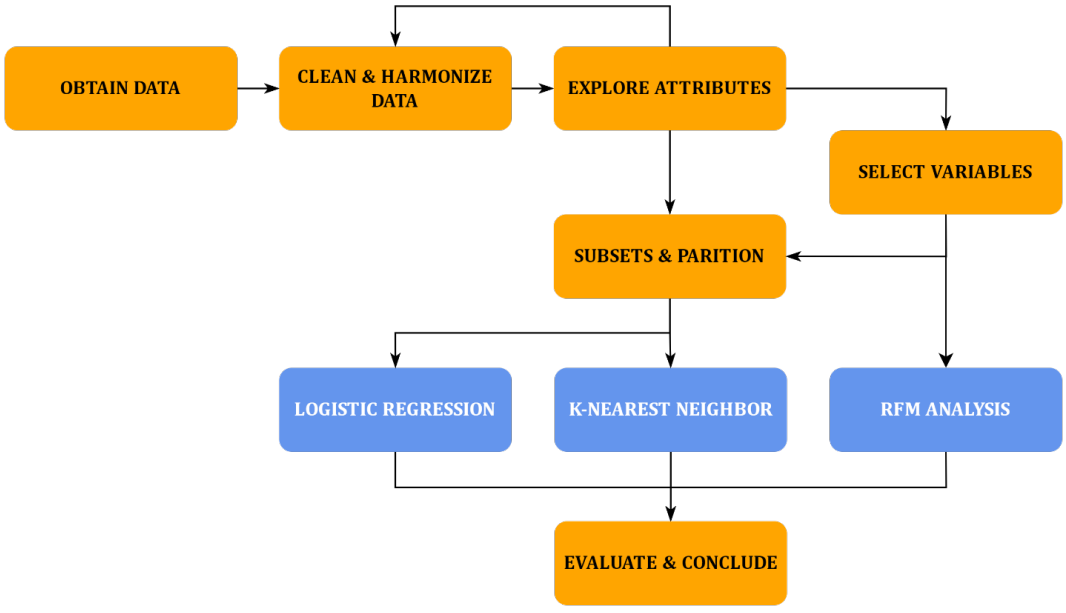
The problem is that CBC falls into a trap that makes them appear to be successful due to the increase of mailing volume, growth of book selection, and customer database, but the profit gradually decreases. After consolidating their plan, CBC decided to improve using their primary asset, customer database, to understand their customers' preferences and then

tailor the books' promotions to only customers who are most likely to enjoy and buy the targeted title they have in mind. With this strategy, it might reduce various costs such as printing and then mass-mailing to not potential customers that breaking even revenues.

To enable this strategy, CBC ran a test campaign to which groups of a randomly selected set of 4,000 customers are more likely to on a book named "The Art of Florence History" (Florence). The dataset includes the customers' purchase history of other books, their description, interactions with CBC in terms of transactions, and their decision to buy the Florence book. The details of customers' meta-data attributes can be found in Table 1 below.

This analysis uses two main approaches: (1) RFM segmenting to manually categorize groups of customers which shall be explained below; (2) Machine Learning methods including Logistic Regression and k-Nearest Neighbor. Before classifications, the CBC data set also required several data wrangling methods such as exploring, harmonization, subsetting, partition, etc. before it is ready to use. The process is summarized in Figure 1.

Figure 1: Diagram of Analysis Process



1. Data

The collected Charles Book Club data set mentioned above contains 4,000 records with 24 variables including one outcome variable "Florence", its two dummy variables "Yes_Florence" and "No_Florence"; two identification columns including "Seq#" and "ID#"; 19 classifier variables are predictors with six variables related to **RFM Analysis** which will be explained in **Classification** section; the rest are quantities of other books each customer bought. Each column is a variable, with the header row giving the name of the variable. The variable names and descriptions are given in Table 1 below:

Table 1: Descriptions of Variables

Variable Name	Description
Seq#	Sequence number in the partition
ID#	Identification number in the full (unpartitioned) market test data set
Gender	0=Male; 1=Female
M	Monetary- Total money spent on books
R	Recency- Months since last purchase
F	Frequency - Total number of purchases
FirstPurch	Months since first purchase
ChildBks	Number of purchases from the category: Child books
YouthBks	Number of purchases from the category: Youth books
CookBks	Number of purchases from the category: Cookbooks
DoItYBks	Number of purchases from the category: Do It Yourself books I
RefBks	Number of purchases from the category: Reference books (Atlases, Encyclopedias, Dictionaries)
ArtBks	Number of purchases from the category: Art books
GeoBks	Number of purchases from the category: Geography books
ItalCook	Number of purchases of book title: "Secrets of Italian Cooking"
ItalAtlas	Number of purchases of book title: "Historical Atlas of Italy"
ItalArt	Number of purchases of book title: "Italian Art"
Florence	=1 'The Art History of Florence" was bought, = 0 if not
Related purchase	Number of related books purchased
MCode	Bins of Monetary amount ranges scored 1-5
RCode	Bins of Recency amount ranges scored 1-4
FCode	Bins of Frequency amount ranges scored 1-3
Yes_Florence	=1 'The Art History of Florence" was bought, = 0 if not
No_Florence	=0 'The Art History of Florence" was bought, = 1 if not

For the sake of calling throughout the entire analysis, I will categorize the attributes as follows:

- Identifier Group: “Seq#”, “ID#”, “Gender”
- RFM Group: “R”, “F”, “M”, “RCode”, “FCode”, “MCode”, “RFM_score” (created in **RFM Analysis** section)
- Target Group: “Florence”, “Yes_Florence”, “No_Florence”
- Book Group: "FirstPurch", "ChildBks", "YouthBks", "CookBks", "DoItYBks", "RefBks", "ArtBks", "GeogBks", "ItalCook", "ItalAtlas", "ItalArt", "Related.Purchase"

2. Attributes Exploration

First, we take a close look at the statistical summary of all attributes in Charles Book Club set in Table 2 to learn about the characteristics of the attributes to devise suitable methods for cleaning, partition, and classification. All of the values in the data set are numerical integers (no decimals) with four binary attributes include “Gender” and all attributes in the target group. Since the “Seq#” and “ID#” columns are not meaningful to this analysis, they will be removed from the data set and all contexts of speaking from this point forward. There are no missing values and most of the attributes have null values, which is acceptable in this case. The range of values in all attributes as a whole varies extremely with the RFM group having averages up to 200 while the other attributes have a maximum of ten in average measures. This is a potential threat when we try to fit data into algorithms such as Logistic Regression.

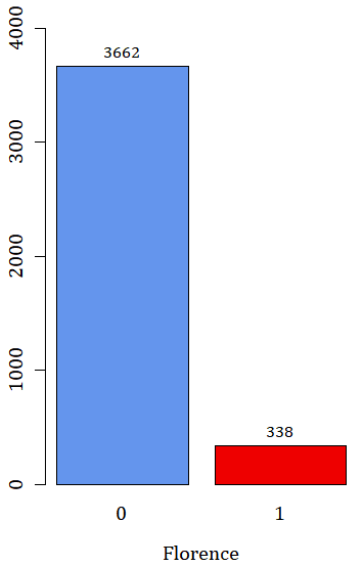
Table 2: Statistical Summary of All Attributes in the Raw Data Set

Statistical Measures	Variables											
	Seq.	*ID.*	*Gender*	*M*	*R*	*F*	*FirstPurch*	*ChildBks*	*YouthBks*	*CookBks*	*DoItYBks*	*RefBks*
Values	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000
Null Values	0	0	1182	0	0	0	0	2424	3047	2338	2981	3181
Missing Values	0	0	0	0	0	0	0	0	0	0	0	0
Minimum	1	25	0	15	2	1	2	0	0	0	0	0
Maximum	4000	32977	1	479	36	12	99	7	5	7	5	4
Range	3999	32952	1	464	34	11	97	7	5	7	5	4
Sum	8002000	66378492	2818	832366	53562	15333	106029	2559	1219	2925	1403	1025
Median	2000	16581	1	208	12	2	20	0	0	0	0	0
Mean	2000	16595	1	208	13	4	27	1	0	1	0	0
SE of Mean	18	150	0	2	0	0	0	0	0	0	0	0
95% CI of Mean	36	294	0	3	0	0	1	0	0	0	0	0
Variance	1333667	89954484	0	10191	66	12	337	1	0	1	0	0
Standard Deviation	1155	9484	0	101	8	3	18	1	1	1	1	1
CFC of Variance	1	1	1	0	1	1	1	2	2	1	2	2

Statistical Measures	Variables											
	ArtBks	*GeogBks*	*ItalCook*	*ItalAtlas*	*ItalArt*	*Florence*	*Related.Purchase*	*Mcode*	*Rcode*	*Fcode*	*Yes_Florence*	*No_Florence*
Values	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000
Null Values	3108	2933	3570	3870	3827	3662	2095	0	0	0	3662	338
Missing Values	0	0	0	0	0	0	0	0	0	0	0	0
Minimum	0	0	0	0	0	0	0	1	1	1	0	0
Maximum	5	6	3	2	2	1	8	5	4	3	1	1
Range	5	6	3	2	2	1	8	4	3	2	1	1
Sum	1156	1550	501	150	183	338	3540	17125	12680	8343	338	3662
Median	0	0	0	0	0	0	0	5	3	2	0	1
Mean	0	0	0	0	0	0	1	4	3	2	0	1
SE of Mean	0	0	0	0	0	0	0	0	0	0	0	0
95% CI of Mean	0	0	0	0	0	0	0	0	0	0	0	0
Variance	0	1	0	0	0	0	2	1	1	1	0	0
Standard Deviation	1	1	0	0	0	0	1	1	1	1	0	0
CFC of Variance	2	2	3	6	5	3	1	0	0	0	3	0

Second, we focus on the target attribute "Florence". Looking at Figure 2, it is clear that this data set committed a bias of having an imbalanced observation in terms of outcome variable's classes. First, the data set was into 60% of the original data set including 2,400 records and the rest goes to the Validation set. A simple code was written to create the tree distribution of outcome variables' class below shows a clear discrepancy of 91.3% (2,191 records) of the records belonging to the class "0" (do not respond to "Florence" and only 8.7% (209 records) belong to the class "1".

Figure 2: Classes Distribution of "Florence" Outcome Variable in Raw Data Set



Even though, this reflects the true reality that among the general list of customers who have different references, those who might be interested in purchasing the Florence book should account for a low number. However, this will become a problem in future analyses if we try to fit the imbalanced data set into machine learning methods because how they work is that those algorithms need balanced numbers of each class for computation so as to avoid biases.

Therefore, a solution proposed is to use a random sampling method called Synthetically Generating Records with the help of the ROSE package in R. Overall, these methods will populate the data set with more records until the two classes are relatively equivalent. The predictors will be randomly multiplied for a number between 0 to 1 to create new records. With this approach, two machine learning methods will be used to analyze both

the imbalanced and the improved balanced sets regarding each algorithms' own segmentation approaches.

In conclusion, the CBC data set were nicely collected with no missing values or major outliers to deal with. However, the two major problems encountered are the extreme value scales of attributes and the large differences between target classes' frequencies. Therefore, several measures will be taken into account to make this data set suitable and available for analysis.

3. Data Harmonization

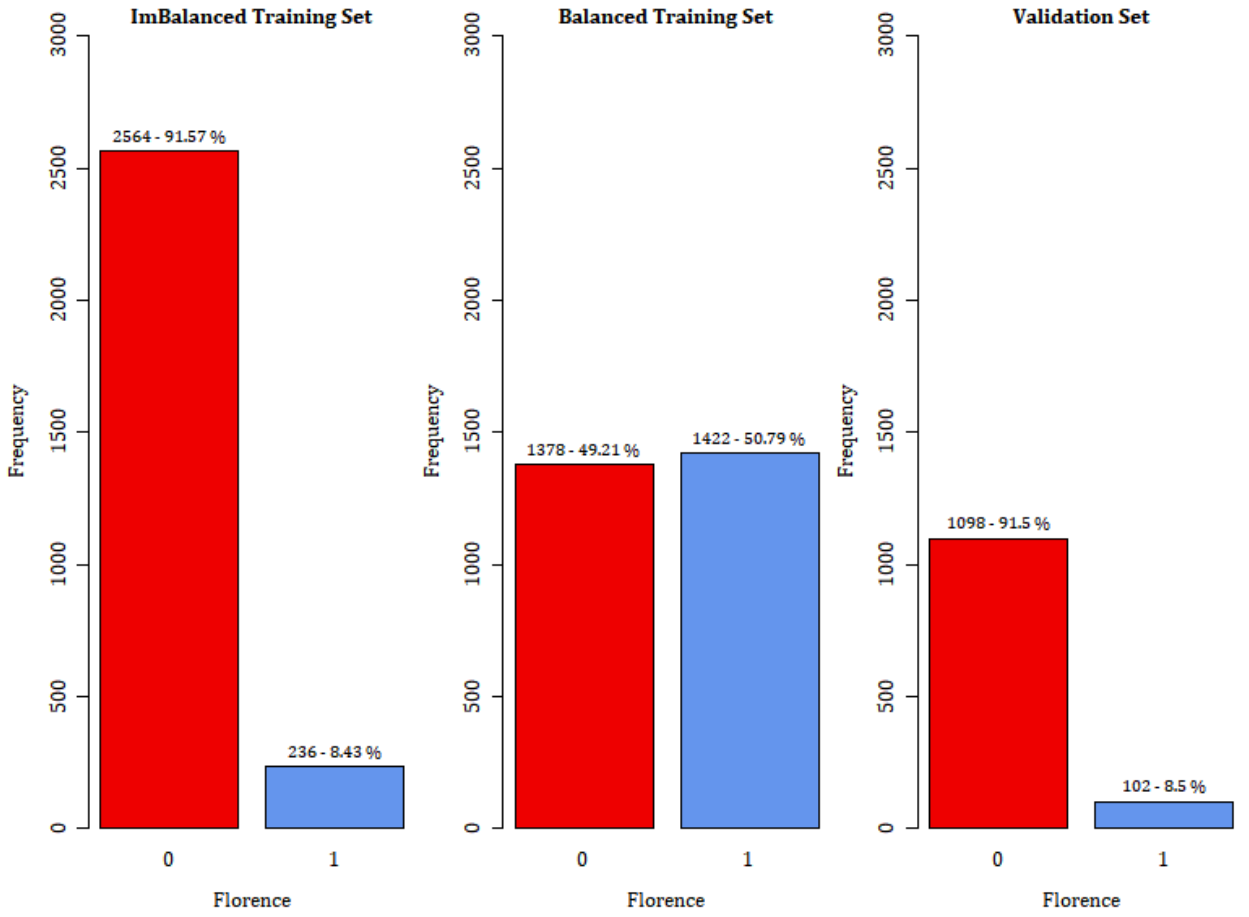
As we have two main approaches including using analysts' marketing expertise and machine learning to learn and classify the data set, the **Data Harmonization** section is generally dedicated to machine learning methods since the machines require special care to be able to learn our data set. Any other data preparation for RFM analysis will be discussed in the **RFM Analysis** section. Overall, this data set is well-collected with no missing or null values. Two major issues are the extreme distances of value scales and the disproportionate distribution of classes in target attributes.

With the first problem of value scales, I used the Min-Max normalization method to bring all attributes to closer scales. This approach shall use the normalization technique to transform the ratio values of the original dataset to the range of 0 to 1 using the min-max method with X' is the scaled data point, X is the original data point, Min(X) is the lowest value and Max(X) is the highest value in the attribute. The function for the Min-Max normalization technique is as follows:

Equation 1: $X' = \frac{X - Min(X)}{Max(X) - Min(X)}$

With the class distribution problem, I used the ROSE package to try to balance the number of target variable "Florence"'s classes to be relatively equal in the training set while keeping the validation set intact and authentic. This method tries to synthesize artificial records with similar features in other attributes to increase the records of minority class "1" and remove those of the majority class "0", while keeping the same number of records in the balanced training set. The contrast in the class distribution of the imbalanced training set, balanced training set, and validation set can be seen in Figure 3.

Figure 3: Classes Distribution of Target Variable in Imbalanced, Balanced Training Sets and Validation Set



4. Subsets and Partition

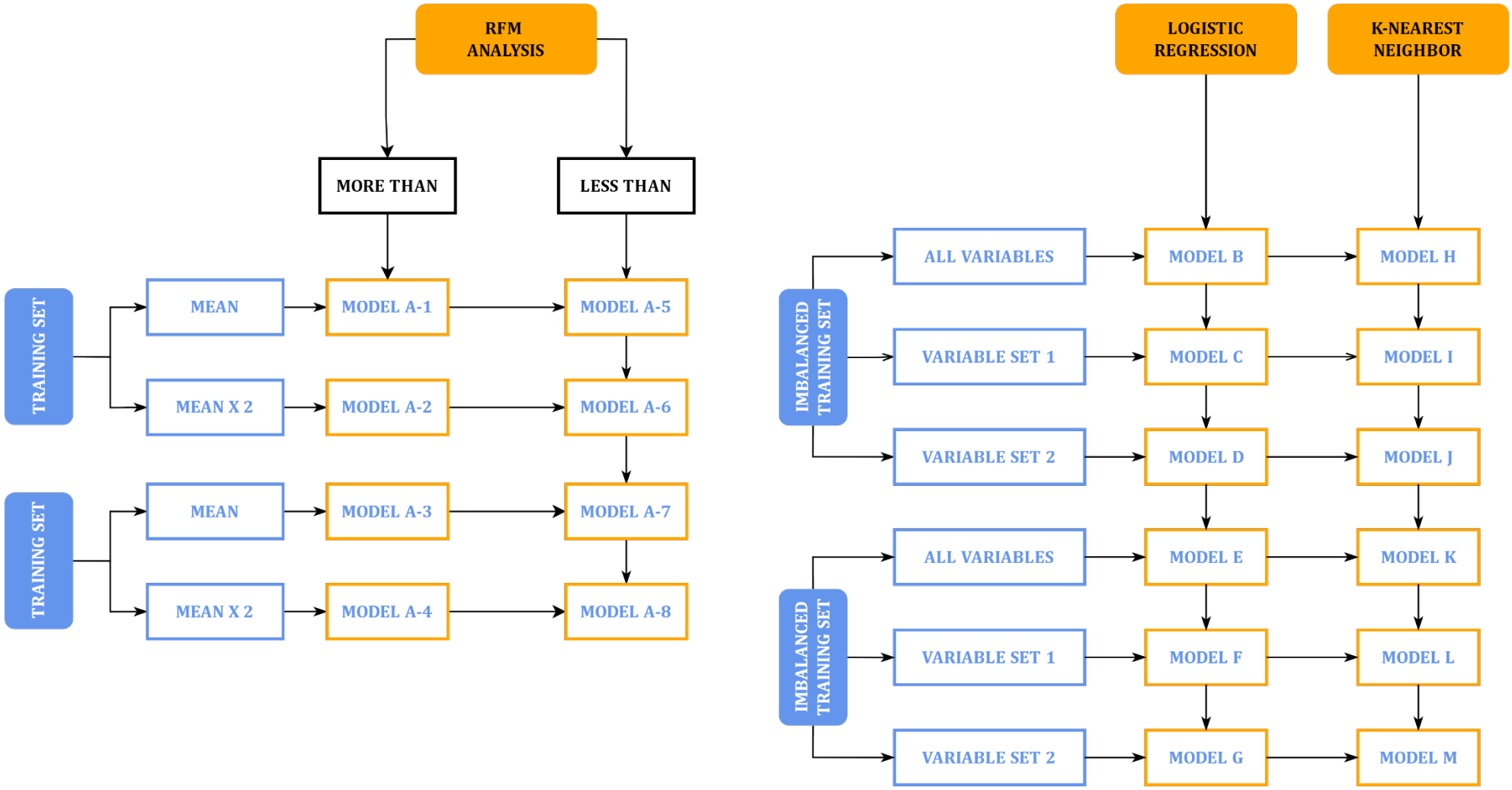
The **Subsets and Partition** section also contribute to the machine learning algorithms. The only common step of RFM analysis and machine learning algorithms is that they both use the partition ratio of 70% for the training set and 30% for validation.

In terms of subsets, three lists of variables were created below including Set All with 16 original variables; Set 1 with five variables related to the Florence book and RFM scores; Set 2 with 11 most significant variables generated by using both-direction stepwise regression.

- Set All: "Gender", "M", "R", "F", "FirstPurch", "ChildBks", "YouthBks", "CookBks", "DoItYBks", "RefBks", "ArtBks", "GeogBks", "ItalCook", "ItalAtlas", "ItalArt", "Related.Purchase"
- Set 1: "Rcode", "Fcode", "Mcode", "FirstPurch", "Related.Purchase"
- Set 2: "Gender", "R", "F", "ChildBks", "YouthBks", "CookBks", "DoItYBks", "RefBks", "ArtBks", "GeogBks", "ItalArt"

In conclusion, for machine learning models, we use six different subsets created by applying three variable sets above into the balanced and imbalanced training sets. The imbalanced and balanced sets are both normalized using Min-Max methods because of the extreme scales of values. The mentioned subsets are clearly visualized in Figure 4 below.

Figure 4: Diagram of Classification Approaches and Models



5. Classification

For this particular purpose to find the customers who might not willing to buy the Florence book so that we can exclude them from our mailing list to reduce the cost, this analysis on the Charles Book Club database uses three classifications methods including RFM analysis which requires marketing expertise and two other machine learning methods which are Logistic Regression and K-Nearest Neighbor. The RFM analysis was conducted on the raw data set with eight models segmented using cut-offs and comparison types, while the other two machine learning algorithms use the processed sets discussed in the **Data Harmonization** section and partitions mentioned in the **Subsets and Partition** section, which include six subsets in total. The map of models and approaches can be found in Figure 4 above.

5.1. RFM Analysis

RFM Analysis is developed in marketing fields to classify potential customer databases using purchase portfolios. In this section, RFM analysis will be discussed in order of explaining concepts, its application to CBC data set, and then report the results after applying the method.

5.1.1. Concepts

Recency, frequency, monetary value is a marketing analysis tool used to identify a company's or an organization's best customers by measuring and analyzing spending habits[1]. The RFM model is based on three quantitative factors:

- Recency: How recently a customer has made a purchase
- Frequency: How often a customer makes a purchase

[1] https://www.investopedia.com/terms/r/rfm-recency-frequency-monetary-value.asp

- Monetary Value: How much money a customer spends on purchases

RFM analysis numerically ranks a customer in each of these three categories, generally on a scale of 1 to 5 (the higher the number, the better the result). The "best" customer would receive a top score in every category. The premise of this method is that the more recent the purchase made, the higher chance customers will still remember the brand on making a purchase from it; the higher the frequency indicates their loyalty to the business and the higher chance they will make more purchases from the enterprise of study; and the more monetary value customers pay from previous purchases and the higher chance they will be willing to pay more.

5.1.2. Application to Charles Book Club Data Set

In the use case of Charles Book Club, to apply the RFM technique, we must extract the following information from the customer database to compute the RFM score as follows:

- Monetary ("M" attribute): Total money the customer spent on books from CBC
- Recency ("R" attribute): Total number of purchases of the customer at CBC
- Frequency ("F" attribute): Total number of purchases of the customer at CBC

Then, from the above raw data obtained, 3 new attributes are created containing categorical values from 1 to up to 5 as bins of value ranges as follows:

Recency ("Rcode" attribute):	**Frequency ("Fcode" attribute):**	**Monetary ("Mcode" attribute):**
0–2 months (Rcode = 1)	1 book (Fcode = l)	$0–$25 (Mcode = 1)
3–6 months (Rcode = 2)	2 books (Fcode = 2)	$26–$50 (Mcode = 2)
7–12 months (Rcode = 3)	3 books and up (Fcode = 3)	$51–$100 (Mcode = 3)
13 months and up (Rcode = 4)		$101–$200 (Mcode = 4)
		$201 and up (Mcode = 5)

Finally, the way to create the RFM score ("RFM_score" attribute) from the segmented attributes in this analysis is to concatenate the 3 codes into one string value with 3 digital

characters together, separated with "_", and becoming a category itself. Then we find the response rate of each RFM score category and compare to find the best one. With this coding for RFM in this case, we have a possible number of different combinations of RFM equal to 4 x 5 x 3 = 60.

An example of RFM Score formation of the first customer in the database is demonstrated in Table 3. First, we extract some relevant information the first customer information. The customer with ID 25 has purchased a total of 297 dollars which is in the range of $201 and up so the Mcode is assigned with 5. The last time they made the purchase was 14 months prior to the analysis so Rcode is assigned with 4 indicating the range of 13 months and up. They bought books from CBC 2 times so the Fcode is assigned with 2. From these codes' values, the RFM score is concatenated in the order of Recency, Frequency, Monetary as the categorical value of 425. It can also be seen that their response to whether they bought the Florence book is "No". Sequentially, with the support from the R programming language, these codes and scores are computed for all 4,000 customers.

Table 3: An Example of the First Customer's Profiles with RFM related Attributes

Seq.	ID	M	R	F	Mcode	Rcode	Fcode	Yes_Florence	No_Florence	RFM_score
1	25	297	14	2	5	4	2	0	1	4_2_5

After understanding the method to compute RFM_score, we shall explain the classification and evaluation methods. First, the training and validation sets are partitioned by a 7:3 ratio with 2,800 observations in the training set and 1,200 in the validation set from the 4,000-record raw set. The analogy of RFM analysis in this paper is to compute certain cut-off values specified by the analyst and use them to decide whether the response rate of each record belongs to which targeted responses (positive or negative) of training and

validation sets. Finally, that classification method is evaluated using a confusion matrix comparing the result by matching responses of the same RFM categories in training and validation sets. The full response rates of the raw, training, and validation sets can be found in Appendix A, B, and C respectively.

The response rate for each customer is calculated by dividing the total number of customers who bought the Florence book within an RFM group by the sum of all customers in that group including the ones who bought the book and the ones who did not buy the book. The formula for the response rate is as follows:

$$\textbf{Equation 2: } Response\ Rate = \frac{Purchase\ Florence}{Purchase\ Florence + Not{-}Purchase\ Florence} \times 100$$

5.1.3. Results

First, we can look at Figure 5 to understand the nature of target class distribution in the raw, training, and validation sets in percentage. All three sets are relatively equivalent in terms of class distribution, which indicates the subsets might be able the represent the whole raw data set.

Figure 5: Distribution of Response Rate Florence in Raw, Training, and Validation Sets

After that, we compute the average response rates of the raw and training set are 0.12 and 0.089 respectively. The manual calculation can be made using the information in Appendix A and B, but in this analysis, I used R. Then, I tried to create four different cut-offs by using the two average values and their time-two values. With the comparison methods, I tried less than or more than the cut-offs to classify whether the response rate of an RFM category will have a buy "1" or not-buy "0" response to the Florence book on both training and validation sets. Then, I use a confusion matrix to compare For example, in Table 4, the sixth Model A-6, the cut-off computed by using the average response in the training set multiplied by 2 (0.089 x 2 = 0.178). Then, all of the response rates in training and validation sets, if less than or equal to 0.178, will be classified as positive or interested class (in this case is "0") and any of which are more than 0.178 will be "1". With this method, the accuracy measures and confusion matrix of RFM analysis are synthesized in Table 4 and Figure 6.

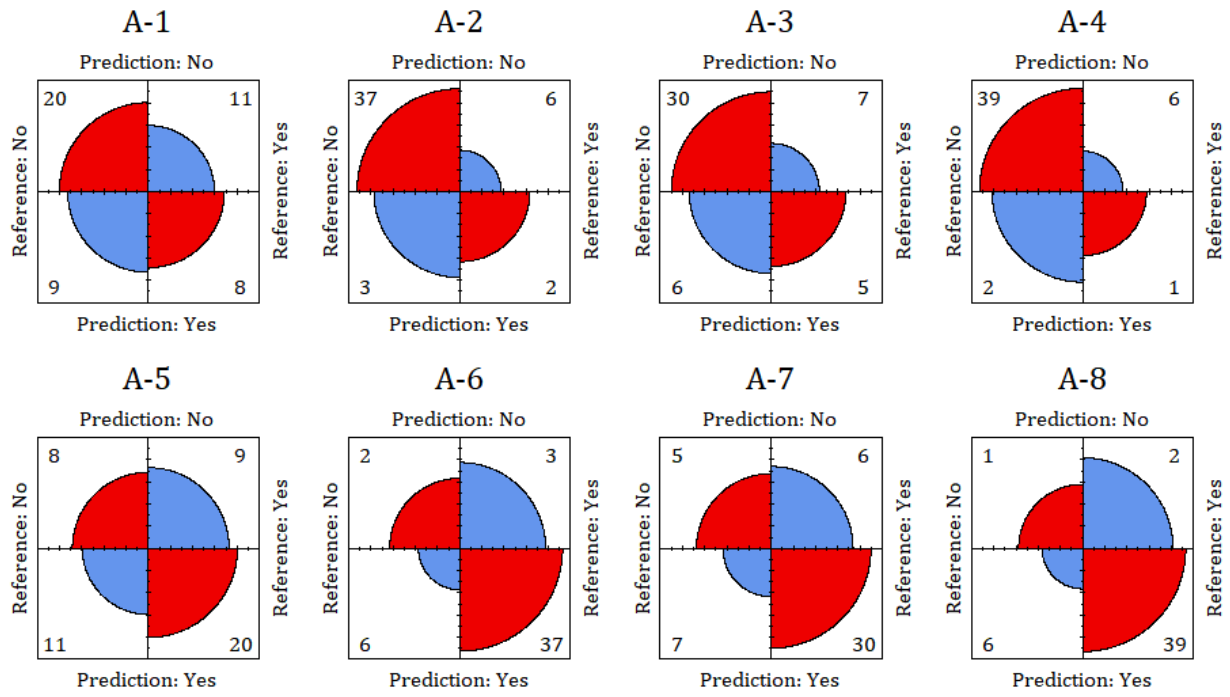
Overall, it appears that when flipping the comparison directions (more or less than) on the models with the same cut-off, it will increase sensitivity and decrease the specificity ("more than" comparison type) and vice versa. Investigating the best performance models (A-4 and A-8) with the same cut-off of 0.24, the overall accuracy is the same with 83.3%, while the sensitivity and specificity of the models flip to one another. In the more-than comparison model A-4, the sensitivity is higher with 95.1% and lower specificity of 14.3% than its counterparts – Model less-than A-8. Therefore, it depends on the analyst's purpose, but in this case, since we prioritize finding class "0", so I choose the model with high sensitivity which is Model A-4.

Table 4: Accuracy Measurements of RFM Analysis

Comparison Type	More Than				Less Than			
Data Set	*Training*		*Raw*		*Training*		*Raw*	
Mean	*0.089*		*0.12*		*0.089*		*0.12*	
Mean of Response	×1	×2	×1	×2	×1	×2	×1	×2
Model	A-1	A-2	A-3	A-4	A-5	A-6	A-7	A-8
Accuracy	0.5830	0.8120	0.7290	0.8330	0.5830	0.8120	0.7290	0.8330
95% CI	(0.432, 0.724)	(0.674, 0.911)	(0.582, 0.847)	(0.698, 0.925)	(0.432, 0.724)	(0.674, 0.911)	(0.582, 0.847)	(0.698, 0.925)
No Information Rate	0.6040	0.8330	0.7500	0.8540	0.6040	0.8330	0.7500	0.8540
P-Value [Acc > NIR]	0.6740	0.7290	0.699	0.7400	0.6740	0.729	0.6990	0.7400
Kappa	0.1130	0.2060	0.2570	0.1230	0.1130	0.2060	0.2570	0.1230
Mcnemar's Test P-Value	0.8230	0.5050	1.0000	0.2890	0.823	0.5050	1.0000	0.2890
Sensitivity	0.6900	0.9250	0.8330	0.9510	0.4210	0.2500	0.4170	0.1429
Specificity	0.4210	0.2500	0.4170	0.1430	0.6900	0.9250	0.8330	0.9512
Positive Predicted Values	0.6450	0.8600	0.8110	0.8670	0.4710	0.4000	0.4550	0.3333
Negative Predicted Values	0.4710	0.4000	0.4550	0.3330	0.6450	0.8605	0.8110	0.8667
Prevalence	0.6040	0.8330	0.7500	0.8540	0.3960	0.1667	0.2500	0.1458
Detection Rate	0.4170	0.7710	0.6250	0.8120	0.1670	0.0417	0.1040	0.0208
Detection Prevalence	0.6460	0.8960	0.7710	0.9380	0.3540	0.1042	0.2290	0.0625
Balanced Accuracy	0.5550	0.5880	0.6250	0.5470	0.5550	0.5875	0.6250	0.5470

Note: All models in this table were evaluated by the confusion matrix method with the seed 2021 for randomization, analyst's cut-off values, and positive class "0". Each accuracy measure is highlighted according to descending rank of best to last values in the top four out of six values with respective colors red, blue, green, and yellow.

Figure 6: Confusion Matrices of RFM Analysis Models



5.2. Logistic Regression

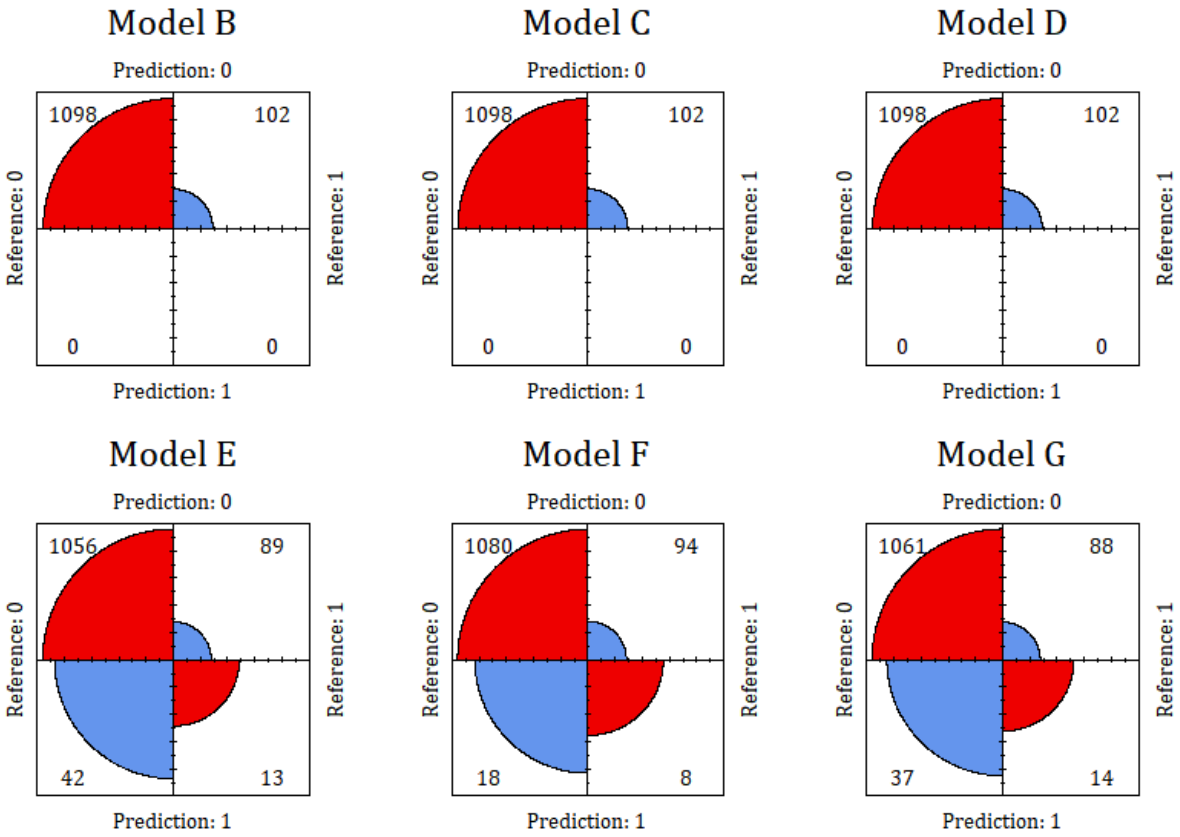
In this analysis, logistic regression is used to fit 6 different models. First, with the imbalanced data set, it will be divided into three different segments of variable sets mentioned in the **Subsets and Partition** section. After improving the data set by creating new balanced training data set, we will continue to fit the models into three segments. The results in Table 5 tells us that all three models using an imbalanced training set have 100% accuracy and the ability to classify not-buying customers while they are not able to find any potential customers to buy Florence book. This might due to the overfitting problem since the "0" accounts for the majority of records. Therefore, it might not be trustworthy to use the models of the imbalanced training set. The next best thing is Model G of the balanced training set with high scores while to able to classify some potential customers. Therefore, model G is chosen in the Logistic Regression algorithm for the purpose of finding the "0" class.

Table 5: Accuracy Measurements of Logistic Regression Algorithms

	Imbalanced Training Set			**Balanced Training Set**		
Variable	All	Set 1	Set 2	All	Set 1	Set 2
Model	B	C	D	E	F	G
Accuracy	0.9150	0.9150	0.9150	0.8910	0.5450	0.8960
95% CI	(0.898, 0.93)	(0.898, 0.93)	(0.898, 0.93)	(0.872, 0.908)	(0.889, 0.923)	(0.877, 0.913)
No Information Rate	0.9150	0.9150	0.9150	0.9150	0.9150	0.9150
P-Value [Acc > NIR]	0.5260	0.5260	0.5260	0.9980	0.8610	0.9910
Kappa	0.0000	0.0000	0.0000	0.1130	0.0940	0.1340
Mcnemar's Test P-Value	<2e-16	<2e-16	<2e-16	5.84e-05	1.37e-12	7.74e-06
Sensitivity	1.0000	1.0000	1.0000	0.9620	0.9836	0.9660
Specificity	0.0000	0.0000	0.0000	0.1270	0.0784	0.1370
Positive Predicted Values	0.9150	0.9150	0.9150	0.9220	0.9199	0.9230
Negative Predicted Values	NaN	NaN	NaN	0.2360	0.3077	0.2750
Prevalence	0.9150	0.9150	0.9150	0.9150	0.9150	0.9150
Detection Rate	0.9150	0.9150	0.9150	0.8800	0.9000	0.8840
Detection Prevalence	1.0000	1.0000	1.0000	0.9540	0.9783	0.9580
Balanced Accuracy	0.5000	0.5000	0.5000	0.5450	0.5310	0.5520

Note: All models in this table were evaluated by the confusion matrix method with the seed 2021 for randomization, cut-off 0.5, and positive class "0". Each accuracy measure is highlighted according to descending rank of best to last values in the top four out of six values with respective colors red, blue, green, and yellow.

Figure 7: Confusion Matrices of Logistic Regression Algorithms



5.3. K-Nearest Neighbor

A total of six models were fitted in KNN with the approaches to the imbalanced and balanced sets similar to the logistic regression algorithm. First, the most optimal k was calculated by half of the square root of a total number of observations in the training set which is 26. Then, the accuracies plot of all k from one to 26 of six models was created as in Figure 8. It appears that models using the imbalanced training set have higher accuracies overall but it also may be resulted in the overfitting of majority class "0". The best k's of Models H to M are 5, 21, 3, 13, 9, and 9 respectively. The results in Table 6 and Figure 9 indicates that the best model to find "0" class is J, the best model to find "1" class is K, and the best one overall with the lowest error of type I and II is model J.

Figure 8: Accuracies of k values

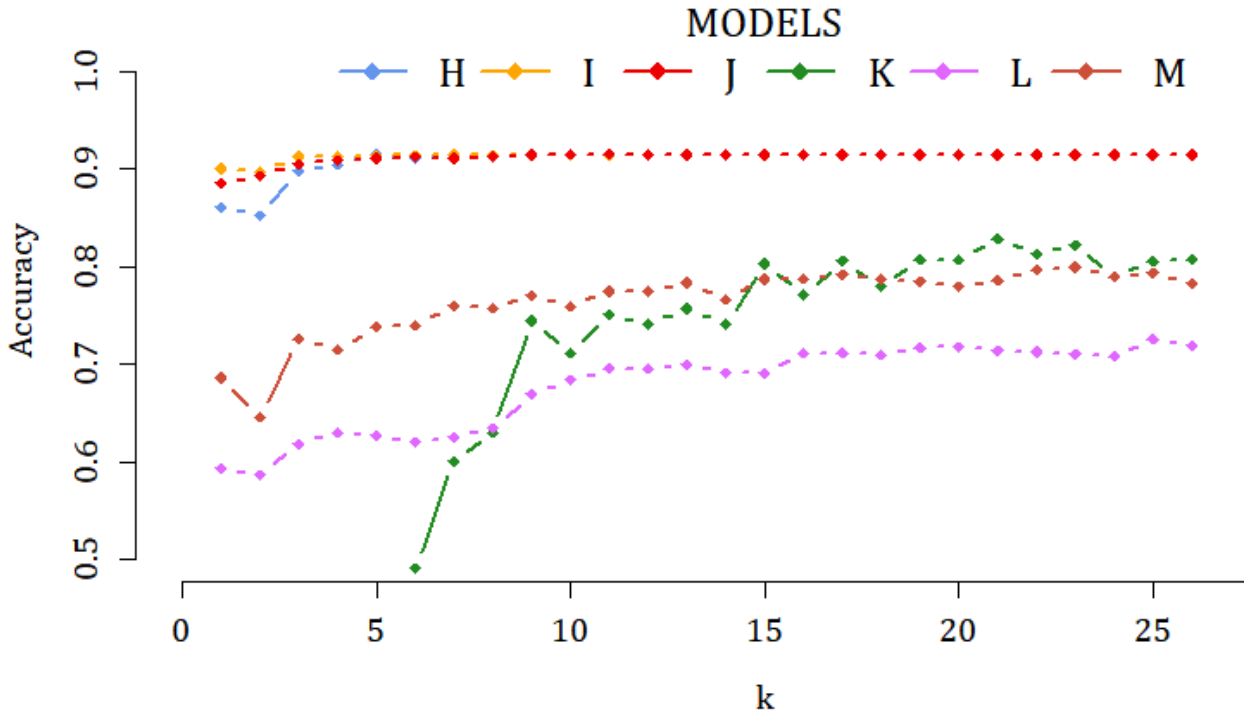
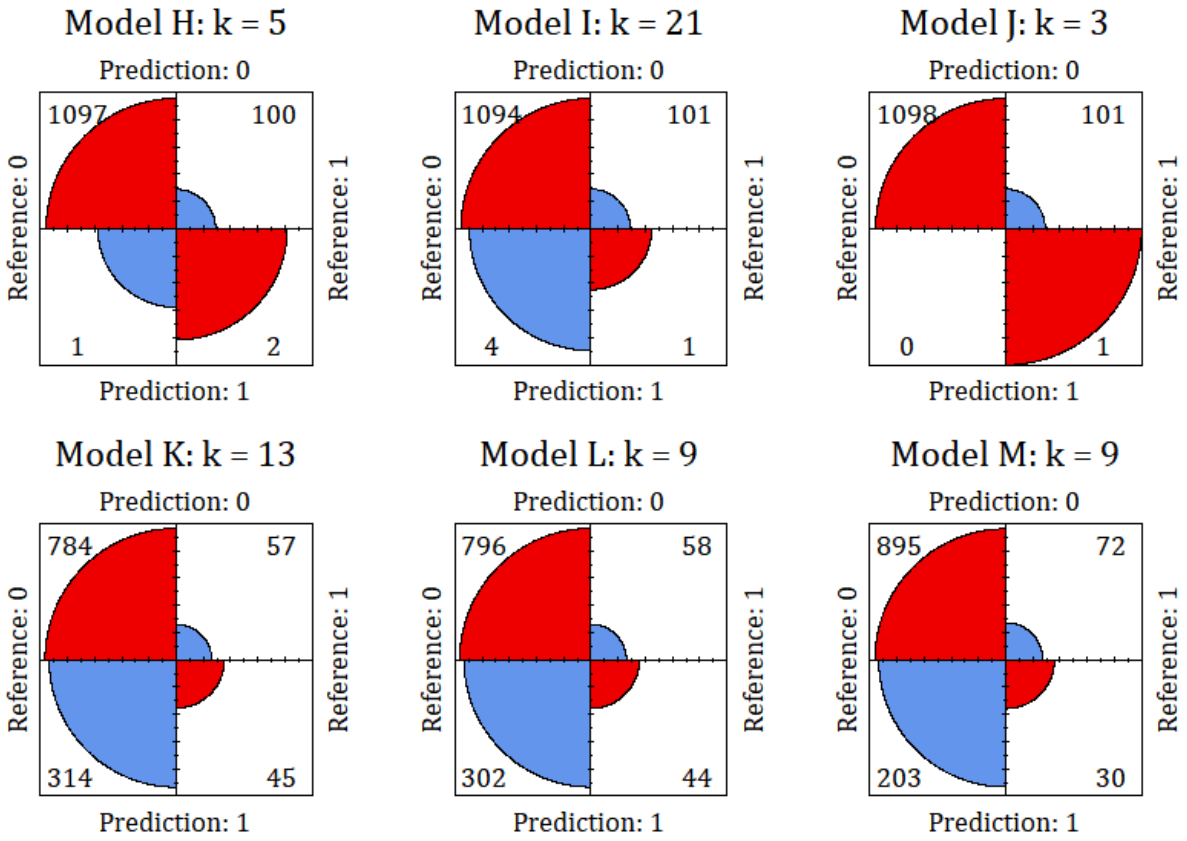


Table 6: Accuracy Measurements of K-Nearest Neighbor Algorithms

	Imbalanced Training Set			Balanced Training Set		
Variable	All	Set 1	Set 2	All	Set 1	Set 2
Model	H	I	J	K	L	M
k	5	21	3	13	9	9
Accuracy	0.916	0.9120	0.9160	0.6910	0.7000	0.7710
95% CI	(0.899, 0.931)	(0.895, 0.928)	(0.899, 0.931)	(0.664, 0.717)	(0.673, 0.726)	(0.746, 0.794)
No Information Rate	0.915	0.9150	0.9150	0.9150	0.9150	0.9150
P-Value [Acc > NIR]	0.485	0.6460	0.4850	1.0000	1.0000	1.0000
Kappa	0.033	0.0110	0.0180	0.0720	0.0750	0.0690
Mcnemar's Test P-Value	<2e-16	<2e-16	<2e-16	<2e-16	<2e-16	4.53e-15
Sensitivity	0.9991	0.9964	1.0000	0.7140	0.7250	0.8150
Specificity	0.0196	0.0098	0.0098	0.4410	0.4310	0.2940
Positive Predicted Values	0.9165	0.9155	0.9158	0.9320	0.9320	0.9260
Negative Predicted Values	0.6667	0.2000	1.0000	0.1250	0.1270	0.1290
Prevalence	0.9150	0.9150	0.9150	0.9150	0.9150	0.9150
Detection Rate	0.9142	0.9117	0.9150	0.6530	0.6630	0.7460
Detection Prevalence	0.9975	0.9958	0.9992	0.7010	0.7120	0.8060
Balanced Accuracy	0.5093	0.5031	0.5049	0.5780	0.5780	0.5550

Note: All models in this table were evaluated by the confusion matrix method with the seed 2021 for randomization, cut-off 0.5, and positive class "0". Each accuracy measure is highlighted according to descending rank of best to last values in the top four out of six values with respective colors red, blue, green, and yellow.

Figure 9: Confusion Matrices of K-Nearest Neighbor Algorithms



6. Conclusions

In the result summary in Table 7, we can see that the best model for RFM analysis is by using twice the average response rate in the entire data set will give the best results, while for machine learning models, Set 2 of variables results in higher accuracy. In terms of accuracy measurements, using RFM gives the best overall result among its eight models while the machine learning models vary from significantly low to high. However, Model J of K-Nearest Neighbor performed best of all so it is highly recommended if the requirement is to use data-driven models.

Table 7: Summary of the best Models of three Classification Methods

Specification	Classification Methods		
	RFM Analysis	**Logistic Regression**	**K-Nearest Neighbor**
Model	A-4	G	J
Data Set	Imbalanced	Balanced	Imbalanced
Variable Set	RFM_score	Set 2	Set 2
Cut-off	≥ 0.24	0.5	0.5
Other Modification	NA	NA	k = 3

Due to the limitation of time and industrial knowledge, this analysis only explores three different methods to solve the problem of finding the potential customer to promote the Florence book to. However, some questions of improvement that remained for future analysis are stated as follows:

In terms of industrial expertise, the question remains out of light whether the collected attribute/ information is sufficient or appropriate, or whether there are any other marketing analysis methods asides from RFM analysis. And with more empirical knowledge, maybe other analysts can learn more in the **Attributes Exploration** step to devise new ways or new sets of variables for classification.

In terms of technology, using a different application such as Python, SAS, SAP Predictive Analytics may give better results than R-4.1.1 with RStudio version 2021.9.0 used in this analysis. It is recommended to try Association Rules or Regression Trees, which are popular classification in marketing analyses, as machine learning models, as well as trying other data set balancing like random resampling to give better-balanced data set.

References

Baesens, B., Roesch, D., & Scheule, H. (2018). *Credit risk analytics: Measurement techniques, applications, and examples in Sas*. WILEY.

Dinov, I. D. (2018). *Data Science and Predictive Analytics: Biomedical and health applications using R*. Springer.

Scheule, H., Rösch Daniel, & Baesens, B. (2017). *Credit risk analytics: The R companion*. CreateSpace, a DBA of On-Demand Publishing, LLC.

Shmueli, G. (2018). *Data mining for Business Analytics: Concepts, techniques, and applications in R*. John Wiley & Sons.

Do, H.X., Rösch, D. and Scheule, H., 2019. *Liquidity constraints, home equity and residential mortgage losses.* Journal of Real Estate Finance and Economics.

Do, H.X., Rösch, D. and Scheule, H., 2018. *Predicting loss severities for residential mortgage loans: A three-step selection approach.* European Journal of Operational Research.

Rösch, D. and Scheule, H., 2010. *Downturn credit portfolio risk, regulatory capital and prudential incentives.* International Review of Finance, 10(2), pp.185-207.

IFRS 9/ CECL:

Krüger, S., Rösch, D. and Scheule, H., 2018*. The impact of loan loss provisioning on bank capital requirements.* Journal of Financial Stability, 36, pp.114-129.

Aniruddho “Oni” Sanyal, Phoenix Computing Solutions: Using SAS Studio (via SAS OnDemand for Academics) for “Credit Risk Analytics”

Appendix A

Table 8: Statistical Summary of RFM Categories in Raw Data Set

No.	RFM_score	Yes	No	Response_Rate	Count
1	132	1	0	1.000	1
2	122	2	3	0.400	5
3	211	2	3	0.400	5
4	133	1	3	0.250	4
5	233	2	6	0.250	8
6	212	3	11	0.214	14
7	114	5	20	0.200	25
8	112	1	5	0.167	6
9	135	14	70	0.167	84
10	235	22	124	0.151	146
11	335	52	305	0.146	357
12	311	2	12	0.143	14
13	234	9	58	0.134	67
14	215	6	43	0.122	49
15	222	1	8	0.111	9
16	413	12	97	0.110	109
17	223	4	33	0.108	37
18	225	8	69	0.104	77
19	115	4	36	0.100	40
20	224	6	56	0.097	62
21	213	3	29	0.094	32
22	324	10	110	0.083	120
23	334	10	112	0.082	122
24	434	15	173	0.080	188
25	412	3	35	0.079	38
26	414	14	164	0.079	178
27	125	3	36	0.077	39
28	435	38	460	0.076	498
29	425	17	222	0.071	239
30	312	3	41	0.068	44
31	323	4	55	0.068	59
32	134	2	28	0.067	30
33	314	9	126	0.067	135
34	433	2	29	0.065	31
35	415	13	193	0.063	206
36	424	9	187	0.046	196
37	423	4	88	0.043	92
38	325	7	167	0.040	174
39	214	2	50	0.038	52
40	315	7	176	0.038	183
41	322	1	25	0.038	26
42	124	1	26	0.037	27
43	422	1	26	0.037	27
44	313	2	54	0.036	56
45	333	1	30	0.032	31
46	111	0	3	0.000	3
47	113	0	16	0.000	16
48	123	0	14	0.000	14
49	332	0	1	0.000	1
50	411	0	22	0.000	22
51	432	0	2	0.000	2

Appendix B

Table 9: Statistical Summary of RFM Categories in Training Data Set

No.	RFM_score	Yes	No	Response_Rate	Count
1	211	2	2	0.500	4
2	122	2	3	0.400	5
3	233	1	3	0.250	4
4	215	6	27	0.182	33
5	311	2	9	0.182	11
6	114	3	14	0.176	17
7	135	10	51	0.164	61
8	234	7	36	0.163	43
9	213	3	17	0.150	20
10	235	15	95	0.136	110
11	335	32	211	0.132	243
12	223	3	22	0.120	25
13	413	9	71	0.112	80
14	224	4	34	0.105	38
15	225	6	52	0.103	58
16	334	9	78	0.103	87
17	134	2	20	0.091	22
18	324	8	82	0.089	90
19	414	11	114	0.088	125
20	433	2	22	0.083	24
21	435	27	320	0.078	347
22	115	2	26	0.071	28
23	125	2	26	0.071	28
24	412	2	26	0.071	28
25	323	3	40	0.070	43
26	415	11	146	0.070	157
27	312	2	28	0.067	30
28	422	1	14	0.067	15
29	434	8	117	0.064	125
30	425	11	164	0.063	175
31	214	2	33	0.057	35
32	424	8	132	0.057	140
33	333	1	18	0.053	19
34	313	2	37	0.051	39
35	325	6	112	0.051	118
36	423	3	62	0.046	65
37	124	1	21	0.045	22
38	315	4	114	0.034	118
39	314	3	95	0.031	98
40	111	0	2	0.000	2
41	112	0	4	0.000	4
42	113	0	10	0.000	10
43	123	0	11	0.000	11
44	133	0	1	0.000	1
45	212	0	5	0.000	5
46	222	0	5	0.000	5
47	322	0	17	0.000	17
48	332	0	1	0.000	1
49	411	0	13	0.000	13
50	432	0	1	0.000	1

Appendix C

Table 10: Statistical Summary of RFM Categories in Validation Data Set

No.	RFM_score	Yes	No	Response_Rate	Count
1	132	1	0	1.000	1
2	112	1	1	0.500	2
3	133	1	2	0.333	3
4	212	3	6	0.333	9
5	114	2	6	0.250	8
6	222	1	3	0.250	4
7	233	1	3	0.250	4
8	235	7	29	0.194	36
9	335	20	94	0.175	114
10	135	4	19	0.174	23
11	115	2	10	0.167	12
12	314	6	31	0.162	37
13	322	1	8	0.111	9
14	434	7	56	0.111	63
15	225	2	17	0.105	19
16	413	3	26	0.103	29
17	412	1	9	0.100	10
18	425	6	58	0.094	64
19	125	1	10	0.091	11
20	223	1	11	0.083	12
21	224	2	22	0.083	24
22	234	2	22	0.083	24
23	435	11	140	0.073	151
24	312	1	13	0.071	14
25	324	2	28	0.067	30
26	323	1	15	0.062	16
27	414	3	50	0.057	53
28	315	3	62	0.046	65
29	415	2	47	0.041	49
30	423	1	26	0.037	27
31	334	1	34	0.029	35
32	325	1	55	0.018	56
33	424	1	55	0.018	56
34	111	0	1	0.000	1
35	113	0	6	0.000	6
36	123	0	3	0.000	3
37	124	0	5	0.000	5
38	134	0	8	0.000	8
39	211	0	1	0.000	1
40	213	0	12	0.000	12
41	214	0	17	0.000	17
42	215	0	16	0.000	16
43	311	0	3	0.000	3
44	313	0	17	0.000	17
45	333	0	12	0.000	12
46	411	0	9	0.000	9
47	422	0	12	0.000	12
48	432	0	1	0.000	1
49	433	0	7	0.000	7

Classify Portfolios for Home Equity Loan by Using Machine Learning Models

Case Study on Classifying Home Equity Loan Applicants

By Chau Hai Phuong Nguyen

George Herbert Walker School of Business & Technology, Webster University

CSDA 6010: Data Analytics Practicum

Dr. Ali Ovlia

December 17, 2021

List of Tables

List of Figures

List of Equations

Table of Contents

Classify Portfolios for Home Equity Loan by Using Machine Learning Models

Nowadays, commercial banks are typically large in size with enormous banking systems, and their fundamental business model continues to rely on financial intermediate by raising finance through deposit-taking, wholesale funding, shareholder capital, and lending, which is a major source of risk. Recognizing trustworthy customers with a high potential of paying back the loan to credit is an essential process that requires financial experts to go through the customers' profiles (or portfolios), and analyzes which are prospect profiles and which are risk, which consumes time and money resources. The world of over eight billion people and thousands of loan requests per day demands better productiveness in terms of volume and velocity solutions. Therefore, with the resourceful historical data of successfully paid loans of customers, the required solution is possible with the help of machine learning algorithms to increase the processing speed and capacity of analyzing large data, as well as avoiding human errors of bias in experts analysis.

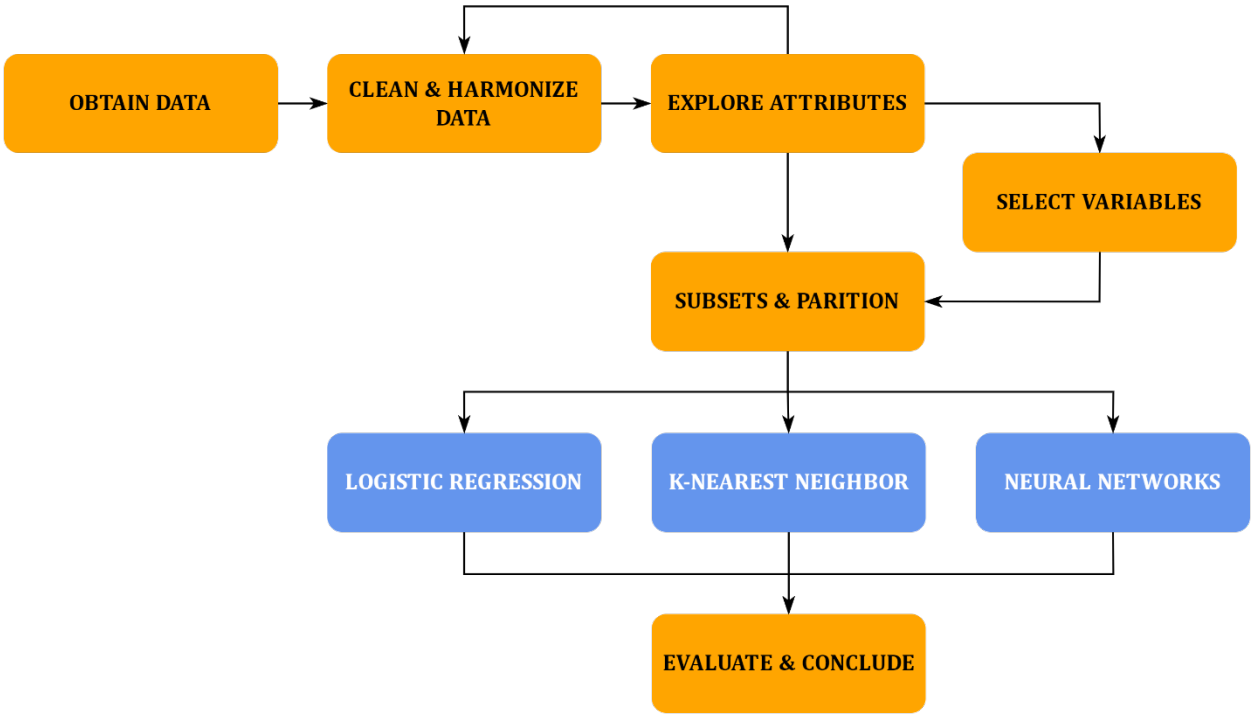
As the result of the mentioned motivation, this paper is conducted with the aim of finding the most effective machine learning model for the given historical data set "Home Equity" in order to most accurately classify those profiles that defaulted their loans so that we can use that model for future prediction of new profiles to decide whether we should give them home equity loans. With the determined purpose, this paper will prioritize the significance of classifying the profiles with higher possibilities to pay loans for approving loan decisions and may or may not discard most of the ineffective results of defaulted loans.

In terms of technology in use, the majority of this analysis is conducted using R programming language version R-4.1.1 implemented in RStudio version 2021.9.0, incorporated occasionally by Microsoft Excel version 2111. Several R packages are required

to download and load into R working environments in order to run the R script for this analysis, which are included in the R Codes companion documents accompanied with this paper.

The entire working process of this analysis in R is summarized in the flow chart of Figure 1 from obtaining the data set process to processing, learning, and evaluation to the conclusion with the arrows indicating the learning steps. This paper will also follow a similar procedure to report the learning journey of this data set to learning using three algorithms including Logistic Regression, K-Nearest Neighbor, and Neural Networks.

Figure 1: Diagram of Home Equity Data Analysis Process



Overall, this data set is very realistic as it represents the world big data example of being disproportionate in classes distribution, unclean because of missing, blank, and outliers values, which is great for learning but requires more time on exploring and cleaning

process. In the end, with this particular data set, After fitting in three machine learning algorithms with 18 different Models created by different sets of balance levels in the target variable's classes distribution, and more than 5 combinations in each model by modifying specific indicator particularly of each algorithm, I discover that k-Nearest Neighbor should be the best fit for the job.

1. Data

The Home Equity data set (HMEQ) in use contains 13 variables with one binary target variable "BAD", and 12 predictor variables either numeric or character, and either continuous or discrete categorical. The total number of observations is 5,960 of home equity loan portfolios approved under expert analysis and recommendations of the Equal Credit Opportunity Act. Our goal is to use the data set to create an empirically derived and statistically sound classification machine learning model to detect potential good portfolios with high potential to pay their loan for loans approval decisions. The nature of variables in the HMEQ data set is described in Table 1 while their statistical summary is in Table 2 below:

Table 1: Description and Definitions of Attributes in Home Equity Data Set

Attribute	Data Type	Description
BAD	integer	1 = applicant defaulted on loan or seriously delinquent 0 = applicant paid loan
LOAN	integer	Amount of the loan request
MORTDUE	numeric	Amount due on existing mortgage
VALUE	numeric	Value of current property
REASON	character	DebtCon = debt consolidation HomeImp = home improvement
JOB	character	Occupational categories
YOJ	numeric	Years at present job
DEROG	integer	Number of major derogatory reports
DELINQ	integer	Number of delinquent credit lines
CLAGE	numeric	Age of oldest credit line in months
NINQ	integer	Number of recent credit inquiries
CLNO	integer	Number of credit lines
DEBTINC	numeric	Debt-to-income ratio

Note: The official HMEQ data set is free to download after filling out the required information by the website: http://www.creditriskanalytics.net/datasets.html.

Since the type of target attribute “BAD” is numeric binary categorical with value “1” for the profiles with defaulted loan and “0” for those with paid loans is quite ambiguous and not instinctively recognizable, a new target variable “STATUS” with character binary categorical values including “defaulted” for those with value “1” and “paid” for those with value “0” in “BAD” attributes is created. Henceforth, the two target variables “BAD” and “STATUS” will be used interchangeably in whichever circumstances that need either numeric and or character input. However, most frequently, this analysis will prefer to attribute “STATUS” as the main target variable. Moreover, the significant values between the binary values of target variables are “1” in the “BAD” attribute and “defaulted” in the “STATUS” attribute because it is more important to identify profiles that are at high risk of committing default to loan than those whose pay. Further explanation for the significant value will be discussed in detail in the **Attributes Exploration** section.

Table 2: Statistical Descriptions of Variables in Home Equity Data Set

Statistical Measures	Variables												
	BAD	**LOAN**	**MORTDUE**	**VALUE**	**REASON**	**JOB**	**YOJ**	**DEROG**	**DELINQ**	**CLAGE**	**NINQ**	**CLNO**	**DEBTINC**
Values	5960	5960	5442	5848	NA	NA	5445	5252	5380	5652	5450	5738	4693
Null Values	4771	0	0	0	NA	NA	415	4527	4179	2	2531	62	0
Missing Values	0	0	518	112	NA	NA	515	708	580	308	510	222	1267
Minimum	0	1100	2063	8000	NA	NA	0	0	0	0	0	0	0.524499
Maximum	1	89900	399550	855909	NA	NA	41	10	15	1168.234	17	71	203.3121
Range	1	88800	397487	847909	NA	NA	41	10	15	1168.234	17	71	202.7876
Sum	1189	1.11E+08	4.01E+08	5.95E+08	NA	NA	48581.75	1337	2418	1016039	6464	122197	158529.1
Median	0	16300	65019	89235.5	NA	NA	7	0	0	173.4667	1	20	34.81826
Mean	0.199497	18607.97	73760.82	101776	NA	NA	8.922268	0.25457	0.449442	179.7663	1.186055	21.2961	33.77992
SE of Mean	0.005177	145.1727	602.6523	750.4134	NA	NA	0.102642	0.011674	0.015369	1.141398	0.023416	0.133848	0.125563
95% CI of Mean	0.010148	284.591	1181.44	1471.088	NA	NA	0.201219	0.022887	0.030129	2.237579	0.045905	0.262393	0.246162
Variance	0.159725	1.26E+08	1.98E+09	3.29E+09	NA	NA	57.36521	0.715795	1.270728	7363.372	2.988317	102.798	73.99004
Standard Deviation	0.399656	11207.48	44457.61	57385.78	NA	NA	7.573982	0.846047	1.127266	85.81009	1.728675	10.13893	8.601746
CFC of Variance	2.003319	0.602295	0.602727	0.563844	NA	NA	0.848885	3.323439	2.508143	0.477343	1.4575	0.476094	0.254641

*Note: This statistical summary of the data set was generated by stat.desc() function in "**pastecs**" package in R instead of the common summary() function.*

2. Data Cleaning and Harmonization

In the real world, there is no perfect data set because during the collection process, there are countless reasons for corrupting the records. Therefore, **Data Cleaning and Harmonization** is the most time-consuming but essential step to prepare adequate sets of input data for the machine to learn. Hence, if the data is not well-prepared and structurally proper, the algorithms will not be able to process. In this Home Equity set, there are three main issues with the data that need to be accessed including missing values, typographical errors in character attributes, and outliers in numerical attributes.

2.1. Missing Values

Since machine learning algorithms do not support processing missing values, identifying, and cleaning them is an essential step. There are several ways to count those values in a data set. Summing up all of them using *is.na()* function in R is commonly used.

Figure 2: Total number of Missing Values of Target Attribute, Rows, and Cells

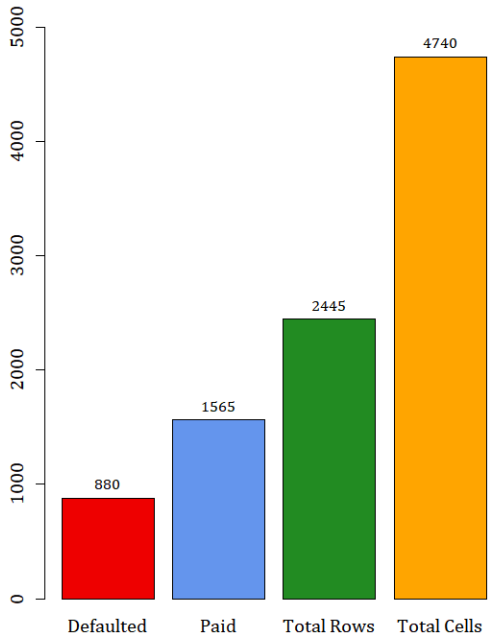


Figure 2 visually describes the overall missing values statistics. In total, the 5,960-record data set contains 2,445 observations (rows) with missing values, and among which, 4,740 cells have missing values. Among those missing values, 880 of which were distributed to the defaulted profiles, and 1,565 were distributed to the paid profiles. This leaves the data set a total of 3,515 observations with no empty cells, which accounts for 59% of the data set, while the missing-value accounts for 41%.

Zooming in the missing value territories, the *stat.desc()* function in "***pastecs***" package in R provides us an overview of statistical measures of all 13 (original) attributes in the entire data set including the missing values. Regarding the first three rows of Table 2, of each attribute, we can compare their total number of values, null, and missing values. Null values appearing in the attributes "LOAN", "MORTDUE", "VALUE", and "DEBTINC" are acceptable since they are attributes denoting quantity. In respect of missing values, only "BAD" and "LOAN" are free of the concerning matter, while the other numeric predictors contain missing values. In the table, only "REASON" and "JOB" do not show any statistical measure because their values are character, so it is not possible to compute numbers, instead, different approaches will be adopted to access these attributes in later steps.

An intuitively better way to visualize the missing values without the data set is to use heat map in Figure 3 created by the function *missmap()* in "***Amelia***" package in R. The vertical y-axis represents the row index, and the horizontal x-axis represents all 13 attributes in ascending order of missing values. "DEBTINC", "DEROG", and "DELINGQ" are in the top three of attributes with the highest number of missing values, while the two categorical variables "REASON", "JOB", as well as "LOAN" and target attribute "BAD" have none. For clarification, the total numbers of cells with missing values in each attribute are aggregated in Table 3.

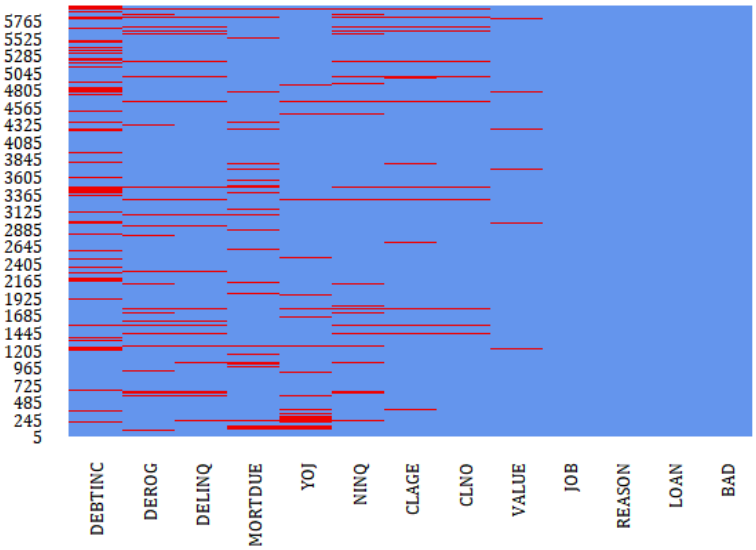
The two common ways to handle missing value is to omit all records with missing values or impute them with substitutes using statistical methods such as mean, median, min, max, or more sophisticated procedures such as linear regression or moving average. Although imputing data seem to be a great choice to maintain a sufficient number of records in the set, such a technique will understate the variability in the data set and violate the integrity of actual information. In the case of Home Equity, the omission choice will disregard

2,445 observations which are 41 % of the total of 5,960 records, and retain 59% of which with 3,515 observations. Based on empirical practice this amount is sufficient for machine learning algorithms, so imputing missing value is opted out of the handling decision.

Table 3: Total Numbers of Missing Value cells in each Attribute

Attribute	Total Missing Value
BAD	0
LOAN	0
MORTDUE	518
VALUE	112
REASON	0
JOB	0
YOJ	515
DEROG	708
DELINQ	580
CLAGE	308
NINQ	510
CLNO	222
DEBTINC	1,267
Total	4,740

Figure 3: Heat Map Chart of Missing Values in ascending order of total missing values with red color representing missing values and blue representing observed values



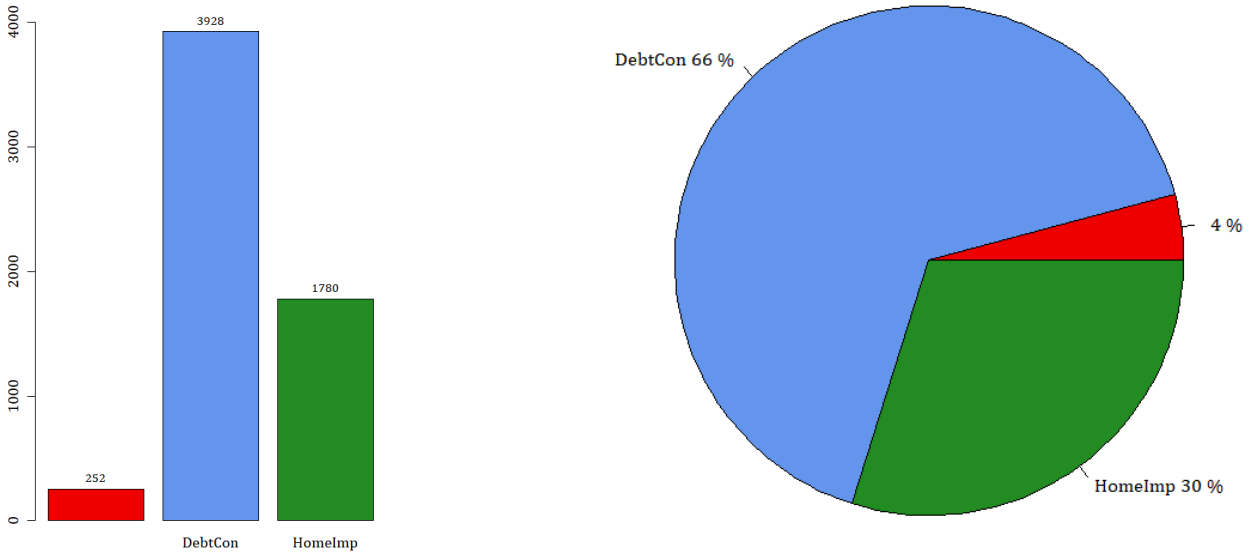
2.2. Typographical Errors

As for the two categorical variables "REASON" and "JOB", the simple adopted approach is to plot out the total frequency of each value in the attribute so as to automatically compare the correctness of the actual categorical types of value in each attribute against their predetermined regulation. Then, we use an appropriate method to manually or automatically adjust the improper value strings. In this section, the original data set of 5,960 observations including missing values is used for typographical error check.

First, regarding the "REASON" attribute in Figure 4, the bar chart shows three types of values in the variable including debt consolidation "DebtCon", home improvement

"HomeImp", and empty spaces containing the respective total record numbers of 3,928, 1,780, and 252. In particular, the pie chart shows that the debt consolidation reason accounts for the largest portion of 66% of the whole data set followed by the home improvement reason with 30% and non-specified empty spaces with 4%.

Figure 4: Distribution of Categorical Values of "REASON" Attribute



Second, regarding the "JOB" attribute in Figure 5, the charts show seven types of values in ascending order of total number of values including other "Other", professional executive "ProfExe", office workers "Office", managers "Mgr", unspecified empty spaces, self-employed "Self", and salespeople "Sales". Among the seven occupations, "Other" account for the majority with 40%, followed by specific occupations including "Professional Executives" with 21%, and office positions in the last of top three with 16%.

In this case, harmonization of "REASON" and "JOB" includes two steps:

- Convert all unspecified empty spaces into "Other" categories for both attributes.
- Create new numeric versions of "REASON" and "JOB" with two new attributes ".REASON" and ".JOB" respectively with the "." prefix for distinguishing purposes. The details of number coding of new attributes and explanations can be found in Table 4.

Figure 5: Distribution of Categorical Values of "JOB" Attribute

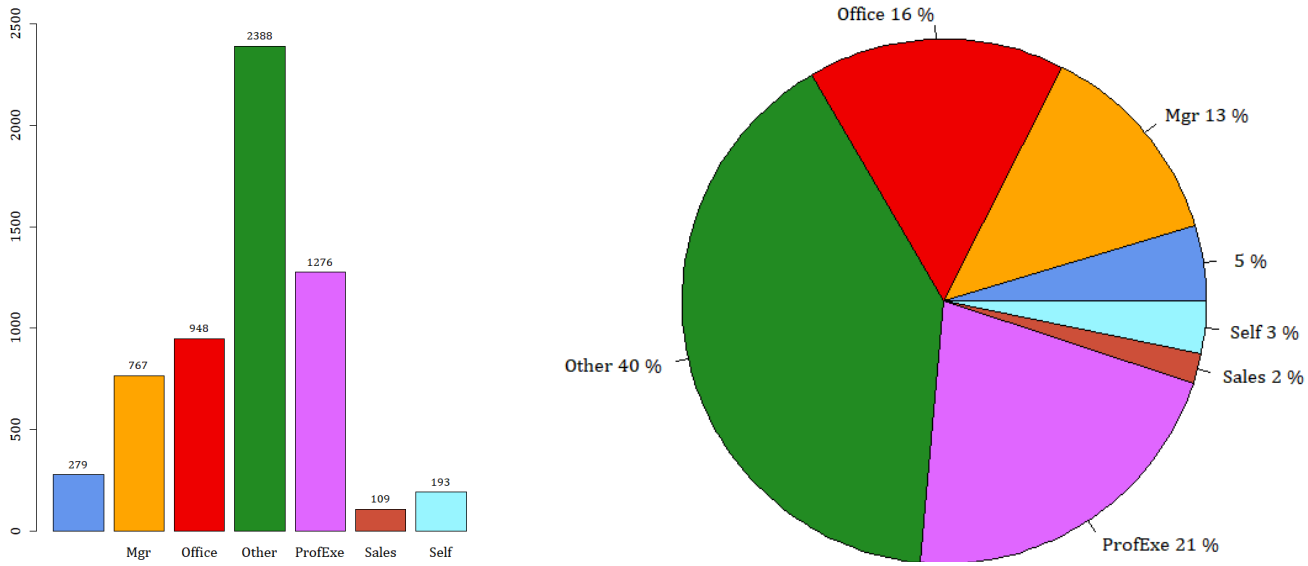


Table 4: New Number Coded Attributes ".REASON" and ".JOB" with Definition and Distribution of Categorical Predictors "REASON" and "JOB"

Coded Attribute	Original Attribute	Attribute Definition	Distribution
		"REASON" Attribute	
1	DeptCon	Debt Consolation	3,928
2	HomeImp	Home Improvement	1,780
3	Other	Other	252
		"JOB" Attribute	
1	Other	Other	2,667
2	ProfExe	Professional Executive	1,276
3	Office	Office Worker	948
4	Mgr	Manager	767
5	Self	Self-Employed	193
6	Sales	Salespeople	109

2.3. Outliers

Outliers are extreme values that sparsely exist outside of the common range of other converged data points of the same attributes. The more data we are dealing with, the greater the chance of encountering erroneous values resulting from measurement error, data-entry

error, or the like. Therefore, it is important to assess outliers and decide whether to remove or keep them. However, assessing each and every individual outlier requires well domain knowledge of the industry, in this case, is Finance, which is out of the scope and capacity of the Data Analyst, the approach is to use pure statistic methods to remove the outliers and evaluate them.

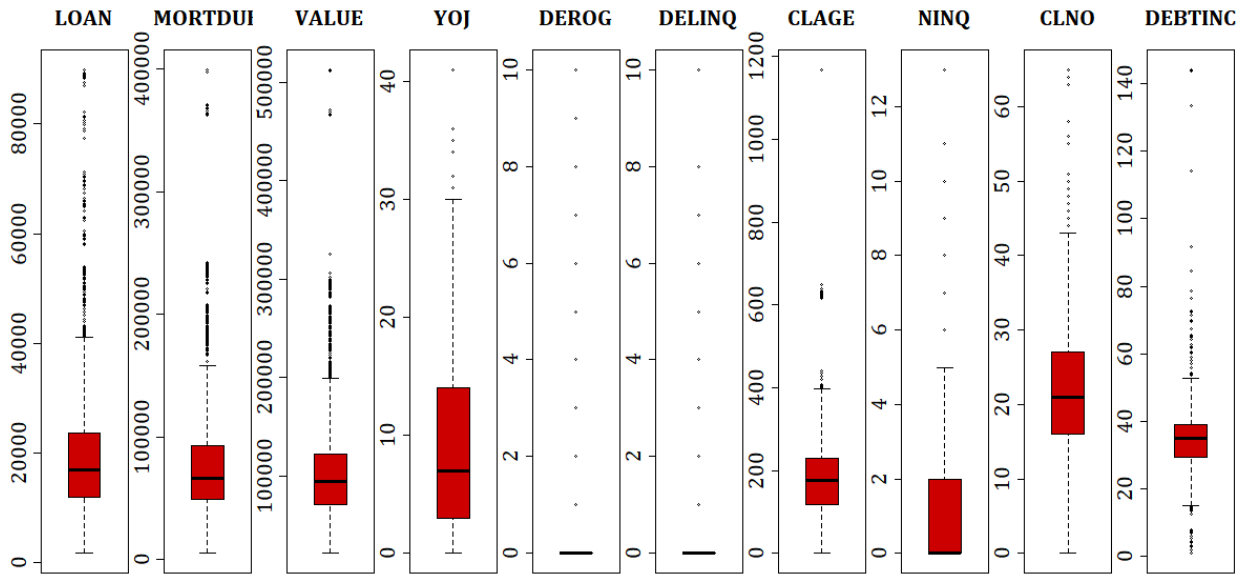
First, let us look at the upper combined box plot in Figure 6of all numeric variables in the clean data set with no missing value. It is clear that all numeric predictors have many outliers above the 75% quartile and only the "DEBTINC" variable also has outliers under the 25% quartiles.

Second, after identifying them, the outliers got removed from the attributes, which can be seen clearly from the lower chart of Figure 6. At this time, the 3,515-observation data set is reduced to 2,302 observations after removing the rows with outliers.

Astonishingly, although up until this point, it seems that the data set is clean and ready for analysis and machine learning algorithms, the problem is unveiled during the **Attributes Exploration** step. It reveals that after removing outliers, the two numeric categorical attributes "DEROG" and "DELINGQ" lost all of their classes of values and only retained the category "0". The details of their value classes' distribution can be found in Table 5 below. Losing all other data types means that we lose all of the important information of the two attributes, or in other words, their existence will have no effects on the analysis. Therefore, the final decision on outliers, in this case, is to keep them for all machine learning algorithms.

Figure 6: : Combined Box Plots of Numerical Attributes in Raw and Clean Datasets Before (Upper Plot) and After (Lower Plot) Removing Outliers

Plot (A): Raw Data set with Outliers



Plot (B): Clean Data set with Outliers Removed

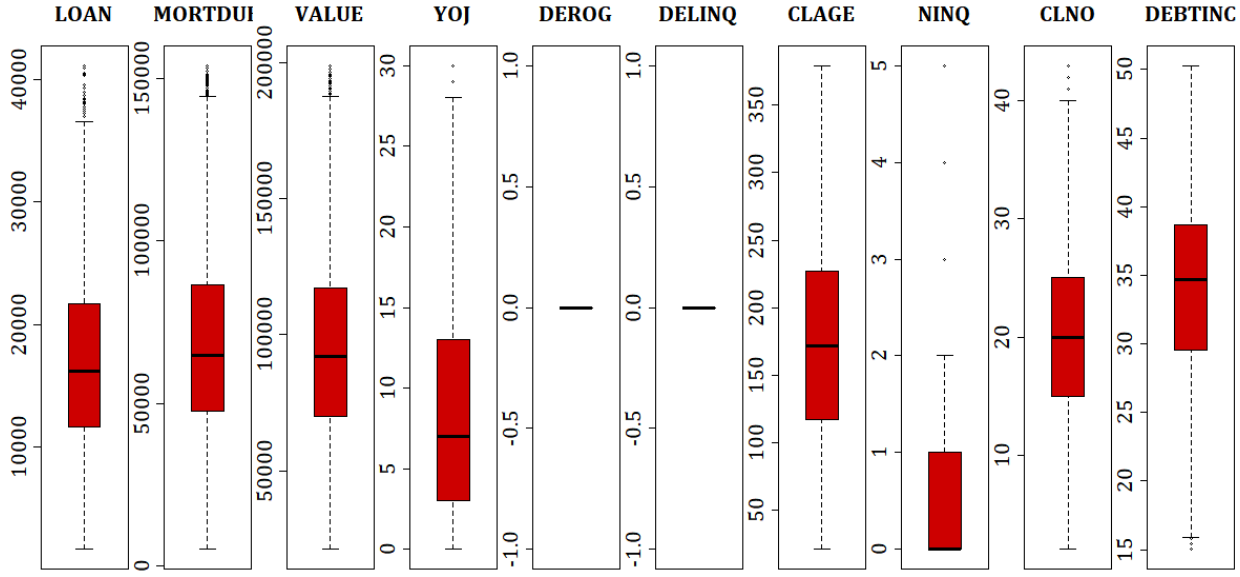


Table 5: Distribution of Categorical Values of "DEROG" and "DELINQ" Attributes Before and After Removing Outliers

Frequency of Values	Categories												
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**13**
	DEROG												
Before	3188	212	80	23	4	1	2	2	1	1	1		
After	2302												
Grouped	3188	212	115										
	DELINGQ												
Before	2964	333	126	50	21	6	7	6	1		1		
After	2302												
Grouped	2964	333	218										
	CLNO												
Before	1763	853	470	241	81	31	22	15	12	7	17	2	1
After	1127	556	304	143	51	14							
Grouped	1763	853	470	429									

In conclusion of this **Data Cleaning and Harmonization** section, three inspections were implemented on missing values of the entire data set, typographical errors of character categorical attributes, and outliers of numeric attributes. After accessing missing values, 41% of records were removed and the data set retains 59% with 3,515 records. During the typographical errors check, all unspecified values were converted to “Other” values, and new numeric versions of the two attributes were also created. As for the outliers, after a thorough assessment, it is decided to keep them since removing them, two numeric categorical attributes will lose all of their information. The final data set, after the initial pre-processing steps containing 3,515 records with retained outliers, and a total of 16 attributes including two target variables and 14 predictors.

3. Exploratory Data Analysis

Exploratory Data Analysis is a significant step to gaining insights and getting familiar with the attributes and their characteristics. However, it is not a linear process as we have been exploring since the initial steps of obtaining and cleaning data. In this step, the data set is analyzed in detail by using One-dimensional Attributes Analysis to investigate

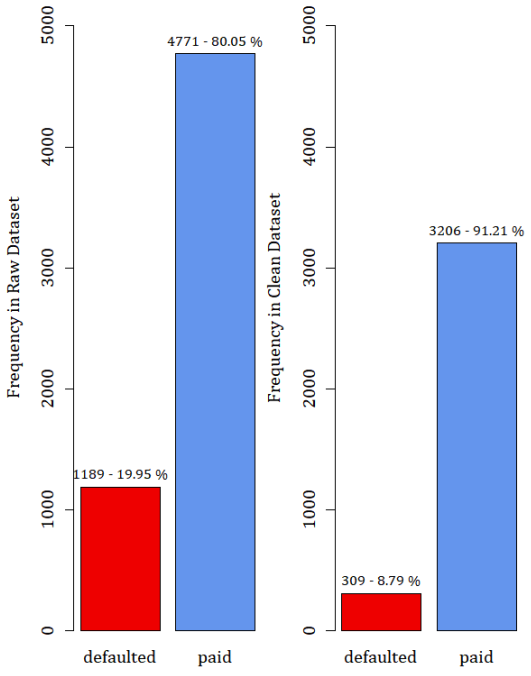
important attributes individually and then using Multi-dimensional Attributes Analysis to combine them appropriately to discover their aggregated information. After this step, we expect to learn the essential characteristics of the data to devise the most optimal ways to harmonize data and to select the most effective sets of attributes to reduce the cost of time and budget.

3.1. One-dimensional Attributes Analysis

3.1.1. Target Attribute

First, we inspect the target variable "STATUS" by plotting their value distribution in the original data set the clean one. Figure 7 shows that in both data set, the "Paid" profiles outnumbers the "Defaulted" ones, which resembles the real-world loan cases where the number of people who do not pay their debts is lower than that of those who pay. However, our main working dataset, the clean data set, suffers a severe imbalanced distribution in values with only 4.65% of "Defaulted" and 95.35% of "Paid". Imbalanced data sets usually confuse the machine learning algorithms, so in preparation steps, various methods will be adopted to create more balanced sets to fit into those algorithms. After that evaluation methods will be used to compare models fitted on the imbalanced and balanced sets.

Figure 7: Classes Distribution of Target Variable in Raw (Left Plot) and Clean (Right) Data Sets



3.1.2. Predictor Attributes

In the Home Equity data set, there is a total of 13 original predictors, which have been discovered in the previous steps. Two more numerical versions of "REASON" and "JOB" have been created for further analysis, namely ".REASON" and ".JOB", making up the total of 14 predictors, and their overall value distributions are shown in Figure 6, while, the statistical summary of all attributes in the new clean data set is in Table 13. Although there are 16 attributes with 14 predictors, only 13 attributes are included in Table 13 because three uncalculated character attributes "STATUS", "REASON", and "JOB" are excluded and replaced by their numerical version "BAD", ".REASON", and ".JOB" respectively. Regardless, it remains difficult to understand the implications underneath the values of those stand-alone attributes without comparing them against each other and the target attribute. Therefore, further attributes analysis on predictors will be conducted in the following **Multi-dimensional Attributes Analysis** section.

3.2. Multi-dimensional Attributes Analysis

3.2.1. Empirical Classification

Before using data-driven methods, it is a common practice to use empirical knowledge to first investigate the given attributes. By using common senses of industrial knowledge, it is possible to classify them into three groups of three distinct characteristics including groups with attributes related to the loan (Loan group), to the loanee occupation (Persona group), and credit scores (Credit group) with standardized colored border coded of yellow, green and purple respectively among the figures in following sections. The details of group classification by characteristics with each group's attributes are as follows:

Table 6: Statistical Descriptions of Variables in No Missing Values and Classes Aggregated Subset

Statistical Measures	Variables												
	BAD	**LOAN**	**MORTDUE**	**VALUE**	**.REASON**	**.JOB**	**YOJ**	**DEROG**	**DELINQ**	**CLAGE**	**NINQ**	**CLNO**	**DEBTINC**
Values	3515	3515	3515	3515	3515	3515	3515	3515	3515	3515	3515	3515	3515
Null Values	3206	0	0	0	0	0	297	3188	2964	0	1763	1	0
Missing Values	0	0	0	0	0	0	0	0	0	0	0	0	0
Minimum	0	1700	5076	21144	1	1	0	0	0	0	0	0	1
Maximum	1	89900	399412	512650	3	6	41	2	2	1168	3	65	144
Range	1	88200	394336	491506	2	5	41	2	2	1168	3	65	143
Sum	309	67445400	2.66E+08	3.76E+08	4662	7655	32216	442	769	636506	3080	77170	119934
Median	0	16900	66590	94071	1	2	7	0	0	177	0	21	35
Mean	0	19188	75623	106876	1	2	9	0	0	181	1	22	34
SE of Mean	0	191	755	914	0	0	0	0	0	1	0	0	0
95% CI of Mean	0	374	1480	1792	0	0	0	0	0	3	0	0	0
Variance	0	1.28E+08	2E+09	2.94E+09	0	2	59	0	0	6789	1	88	66
Standard Deviation	0	11316	44762	54200	1	1	8	0	1	82	1	9	8
CFC of Variance	3	1	1	1	0	1	1	3	2	0	1	0	0

Note: This is the statistical summary of the new working data set with 3,515 records and 16 attributes. The character attributes "REASON" and "JOB" are replaced by the numeric versions ".REASON" and ".JOB" respectively. The character version of the "BAD", "STATUS" attribute, is excluded from this table because it is incalculable.

Loan Group: has four attributes describing each profile's information of the requested loan and the collateral mortgage. The attributes include:

- LOAN
- MORTDUE
- VALUE
- REASON

Persona Group: has three attributes describing the job; its longevity and the job's income as the form of debt-to-income of each loanee. The attributes include:

- JOB
- YOJ
- DEBTINC

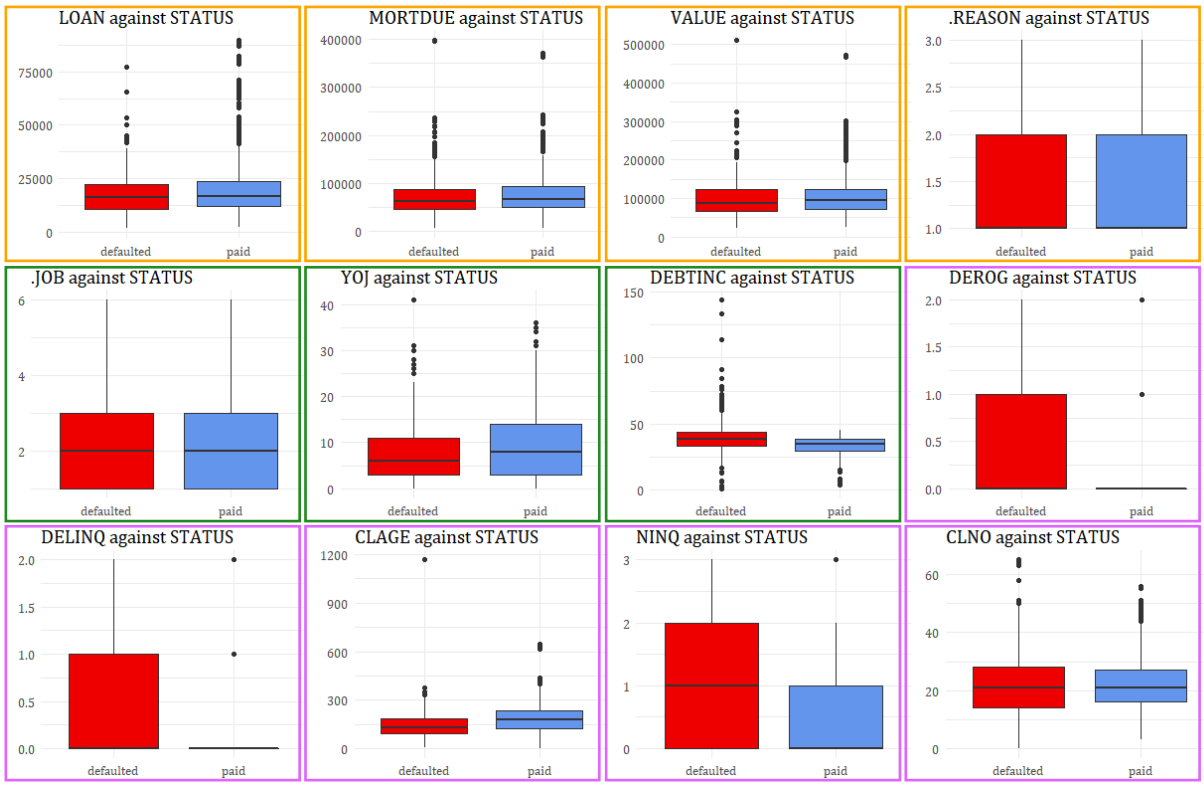
Credit Group: has five attributes contains the credit history of each profile including the attributes as follows:

- DEROG
- DELINQ
- CLAGE
- NINQ
- CLNO

3.2.2. Predictor against Target Attributes

As the goal is to analyze the impact of attributes of each loan portfolio and learn which types of the portfolio are more likely to commit a default on loan, each variable is inspected against the target attribute "STATUS". It is worth mentioning that this is the step, mentioned in the **Data Cleaning and Harmonization** process, where we discovered that after removing the outliers, the categorical attributes "DEROG", "DELINQ", and "CLNO" profiles only contain the "defaulted" class of "STATUS" attribute. Thus, we had to go back to the cleaning process to devise a different way to reduce the outliers in these attributes by grouping minor classes together. After that, we went through the exploration process with the new clean data set with similar attribute analysis methods (one-dimensional and multi-dimensional).

Figure 8: Side-By-Side Boxplots of Target Attribute against Numerical Attributes



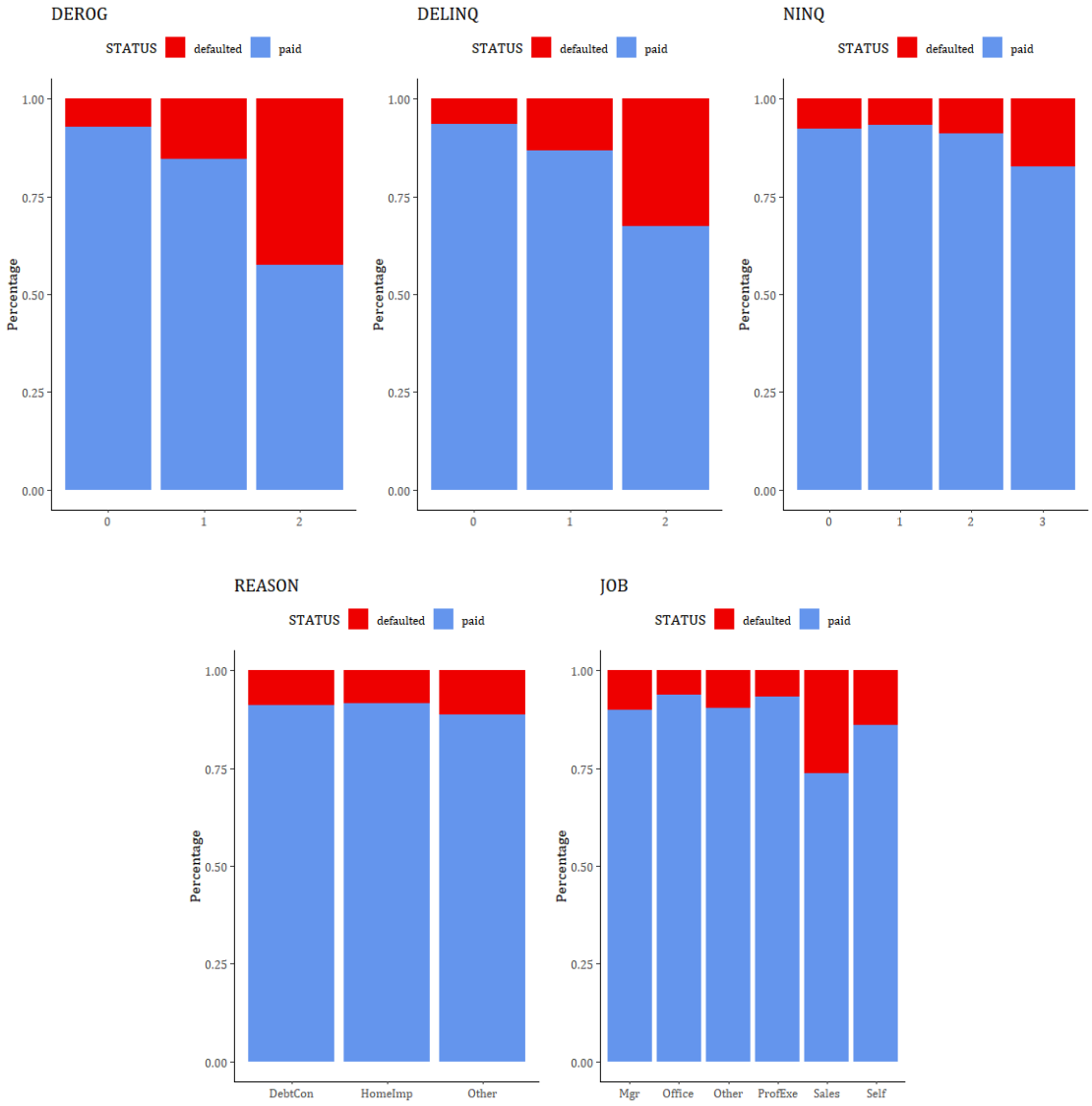
As for numerical variables, Figure 8's side-by-side boxplots allows us to explore the distribution of the target variable's outputs against different numerical predictors with the first row are variables in the Loan group with yellow border, the first three plots in the second rows with green borders are the Persona group and the rest with purple border are of Credit group. Generally in most variables, the distributions of class "defaulted" and "paid" are relatively equal with few outstanding values. Only three credit risk indicators include the number of major derogatory reports[1] "DEROG", number of delinquent credit lines[2]

[1] Derogatory reports of a user's portfolio may include many types of serious negative items which leads to high reductions of scores in credit and the ability to be accepted for loans, new credit lines, or rent apartment, ect. Derogatory items are including but not limited to late payment over 180 days, bankruptcies, civil judgement, debt settlement, collections, etc.

[2] Delinquent credit lines refers credit lines with less serious negative items compared to derogatory such as a late payment less than 180 days.

"DELINGQ", and number of recent credit inquiries "NINQ". Delving into these credit risk categories' bar plots in Figure 9, it is clear that when the number of their classes increases, the more in numbers of distributed "defaulted" profiles with the highest classes always accounts for the most among those three categories. This implies that the credit risk attributes have a strong correlation with the defaulted profiles with the premise that if a loanee has high credit risk, which indicates high delinquent tendencies in credit score, they will also implant a high risk of jeopardizing their home equity loans.

Figure 9: Classes Distributions of Target Attribute against Categorical Attributes

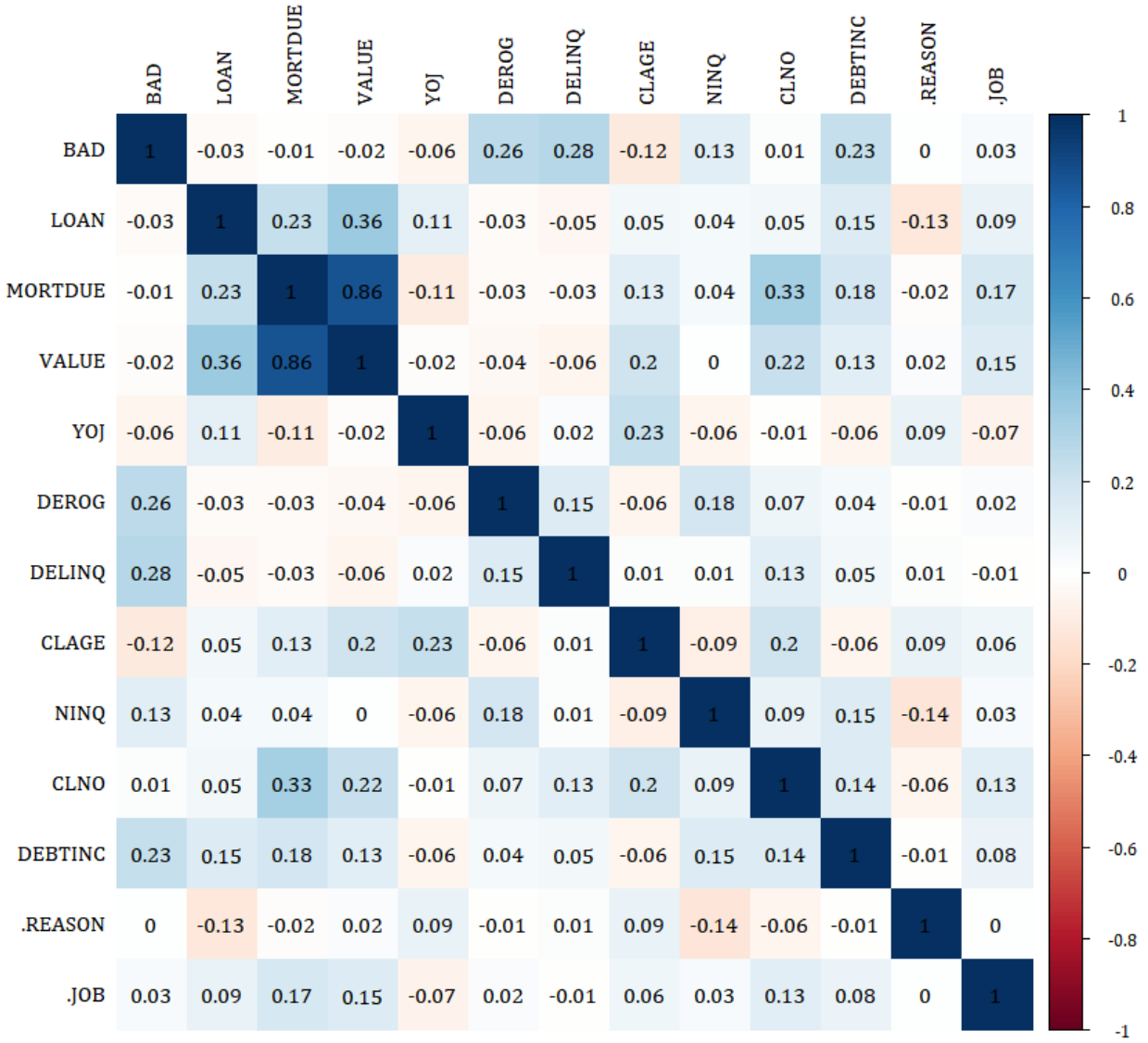


3.2.3. Variables Selections

As mentioned, the purpose of attributes is to discover sets of predictors that have high impacts on the outcomes of the target variable and eliminate the insignificant ones so as to reduce the cost of time and money. In this section, three methods are used including correlation plot, stepwise regression, and information values.

3.2.3.1. Correlation Matrix: The matrix in Figure 16 graphically plotted the correlation coefficient between each variable. Since most of the variables are not highly correlated with one another (coefficients are greater than 0.5), so we do not implement an elimination process between variables that are strongly correlated to the target variable. Instead, we only focus on five predictors with the strongest relations to "STATUS" in the matrix (coefficients over 0.1) descendingly including "DELINQ" (0.28), "DEROG" (0.26), "DEBTINC" (0.23), "NINQ" (0.13), and "CLAGE" (- 0.12). Interestingly, our first set of predictors includes the whole Credit group of predictors that we empirically classified using financial understanding. It is worth noticing that four out of five predictors that have positive to the "BAD" target variable (numerical version of target variable "STATUS"), whose "1" class means negativity with defaulted on loans and "0" class indicates positivity with paid loan status) are variables that indicate credit risk or negativity in credit scores. This is because we have discovered that our data set is imbalanced with more than 90% of records belonging to the class "1" so not only do the negative variables have positive relations with the target, they also have stronger relations. Whilst "CLAGE" is a positive indicates because that the longer the age of your oldest credit lines, the better your profile seems to be (paying on time and being able to maintain the oldest credit line).

Figure 10: Correlation Matrix of Variable Pairs



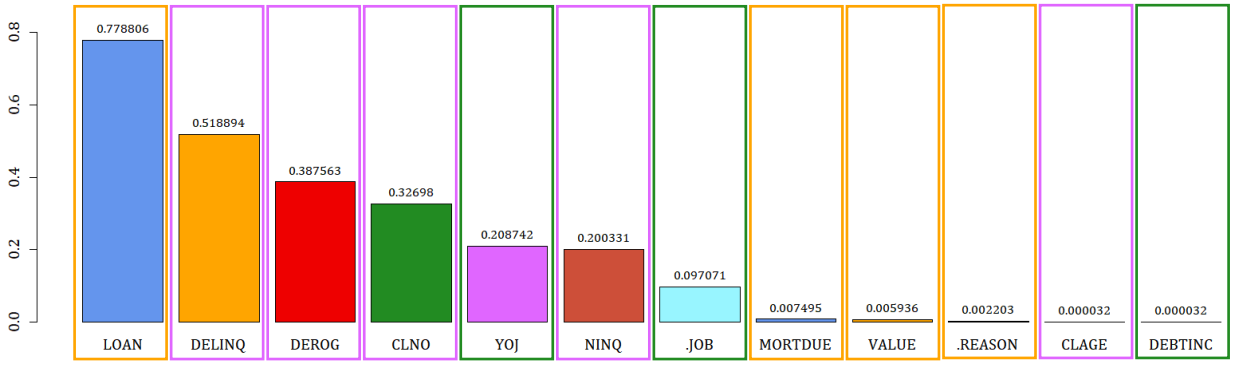
3.2.3.2. Stepwise Regression: After seven steps of adding and removing variables, the stepwise iterative construction of regression computed a set of statistically significant variables with large values of R-Squared including six variables “DEROG”, “DELINQ”, “CLAGE”, “DEBTINC”, and “NINQ”, which are the same ones we discovered in correlation matrix above.

3.2.3.3. Information Values[3] (IV): Information value is one of the popular variable selection methods based on the Weight of Evidence invented in the credit scoring world with the same purpose of logistic regression which is to find the highest possibility of binary outcome variables. Equations 1 and 2 describe two-step calculations for IV. However, in R, we can simply use *IV()* function to create a table of Information Values of each variable. Figure 17 plots the IV table in descending order. As the IV smaller than 0.1 indicates week predictive power, we only select six variables with IVs are equal to 0.1 including variables "LOAN", "DELINQ", "DEROG", "CLNO", "YOJ", "NINQ". Again, the majority of variables in this set are in the credit group with the other two being "LOAN" is in the Loan group and "YOJ" is in the Persona group.

$$\textbf{Equation 1: Weight of Evidence: WOE} = \ln\left(\frac{\%\ \textbf{of Non}-\textbf{Events}}{\%\ \textbf{o fEvents}}\right)$$

$$\textbf{Equation 2: Information Value: IV} = \sum(\%\textbf{ofNon} - \textbf{Events} - \%\textbf{ofEvents}) \times \textbf{WOE}$$

Figure 11: Information Values of Predictor Variables in Ascending Order of Values



In summary, exploring all attributes of this Home Equity data set in an individual fashion allows us to discover the imbalance in terms of classes distribution in target values which suggests further balancing methods as solutions, as well as handling outliers of some

[3]https://www.listendata.com/2015/03/weight-of-evidence-woe-and-information.html#What-is-Information-Value-IV

predictors by grouping minority classes. In a collective manner, studying their correlation in matrix form and their significance using Stepwise Regression and Information Values methods allows us to discover two sets of significant variables containing the most amount of information that might be used on behalf of all variables to save computational cost in terms of time and money.

4. Subsets and Partition

In this final step of preparation before fitting data sets into machine learning algorithms, we need to divide our data set into subsets appropriately and partition them into the training set for model fitting validation set for evaluating models. We first define the subsets created during variable selection during attribute exploration. Second, we partition the sets into training and validation with the ratio of 70% and 30% respectively. Lastly, we use "ROSE" package to balance the classes of outcome variables in training sets.

4.1. Variable Sets

The sets of variables that will be used in this analysis are as follows:

- All Variables (All): a set with all 13 predictors

- Variable Set 1 (Set 1): a set of five variables selected from correlation matrix and stepwise regression including "DEROG", "DELINQ", "CLAGE", "DEBTINC", and "NINQ" variables

- Variable Set 2 (Set 2): a set of six variables selected using information values including "LOAN", "DELINQ", "DEROG", "CLNO", "YOJ", "NINQ" variables

4.2. Partition

Due to the fact that the purpose is classification (not prediction) to find the best models and indicators of defaulted profiles, the third testing subset is not necessary. Therefore, in

this Home Equity analysis, the partition includes 70% of the clean data (2,460 observations) set divided for training and 30% (1,055 observations) for validation. Moreover, because we planned to balance the training set using the "ROSE" package, it is worth mentioning that 70% of the training set (in imbalanced data set) will be used to create another training whose "STATUS" attribute's classes are relatively equal in number.

4.3. Balance Training Set with ROSE Package

Once again, this Home Equity set is a great representation of real-world situations where the classes distributions of the response or target variables are not perfectly equal (imbalanced). However, the current machine learning algorithms at our disposal tend to tremble when faced with imbalanced classification data sets. Additionally, they result in biased predictions and misleading accuracies because they do not get the necessary information about the minority class to make an accurate prediction. For this reason, we use the "ROSE" (Random Over Sampling Examples) package to help by generating artificial data based on sampling methods and smoothed bootstrap approach. After using the "ROSE" package, Figure 18 clearly shows the contrast of imbalance in the initial training set and validation set versus the balanced training set. The new balanced set now consists of 48.46% of records with defaulted status on loan and 51.54% with paid status on loan.

In summary, after defining three sets of variables, partitioning into training and validation sets, and balancing outcome variables classes to create a new training set, we have in total six sets of training to fit into machine learning algorithms. For clarification, the diagram in Figure 19 shows six subsets mentioned.

Figure 12: Classes Distribution of Target Attribute in Imbalanced Training Set, Balanced Training Set, and Validation Set

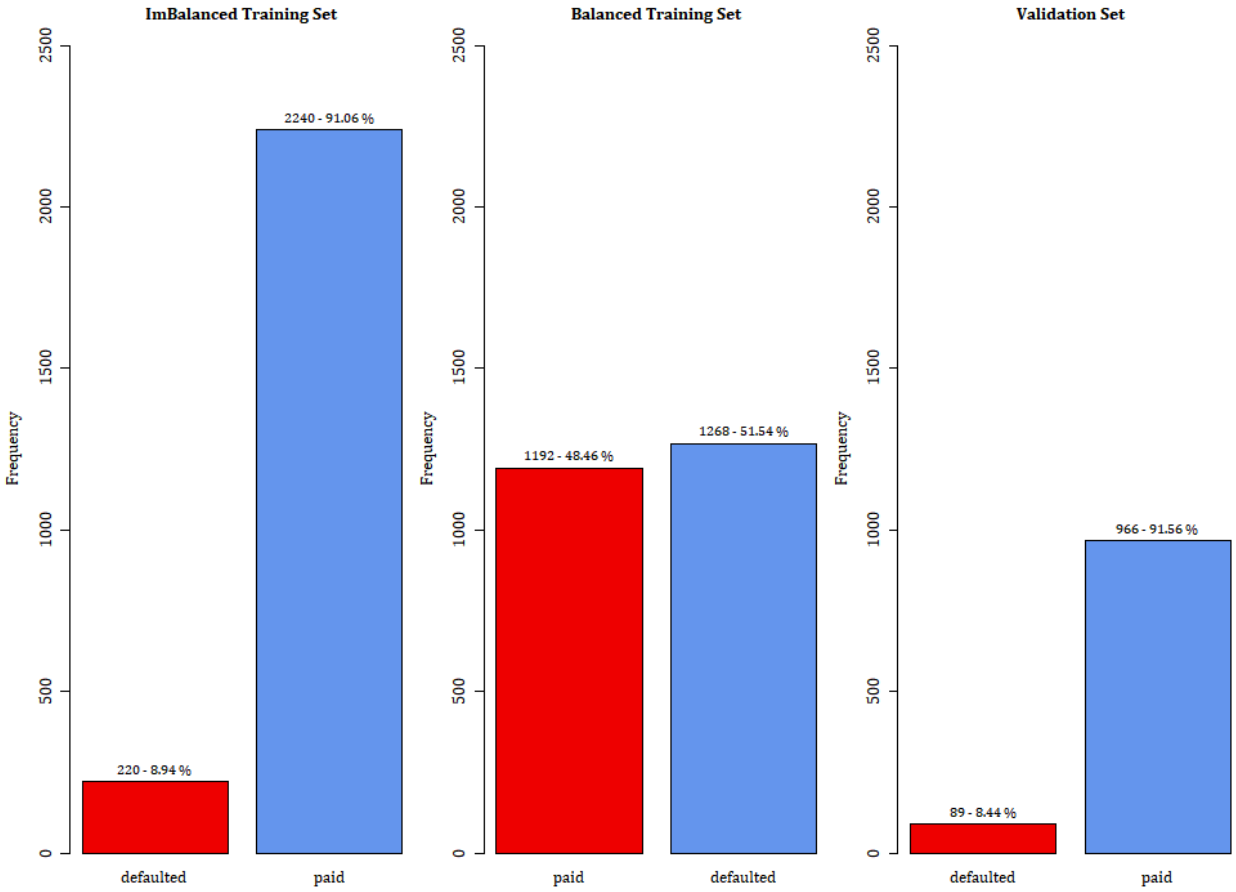
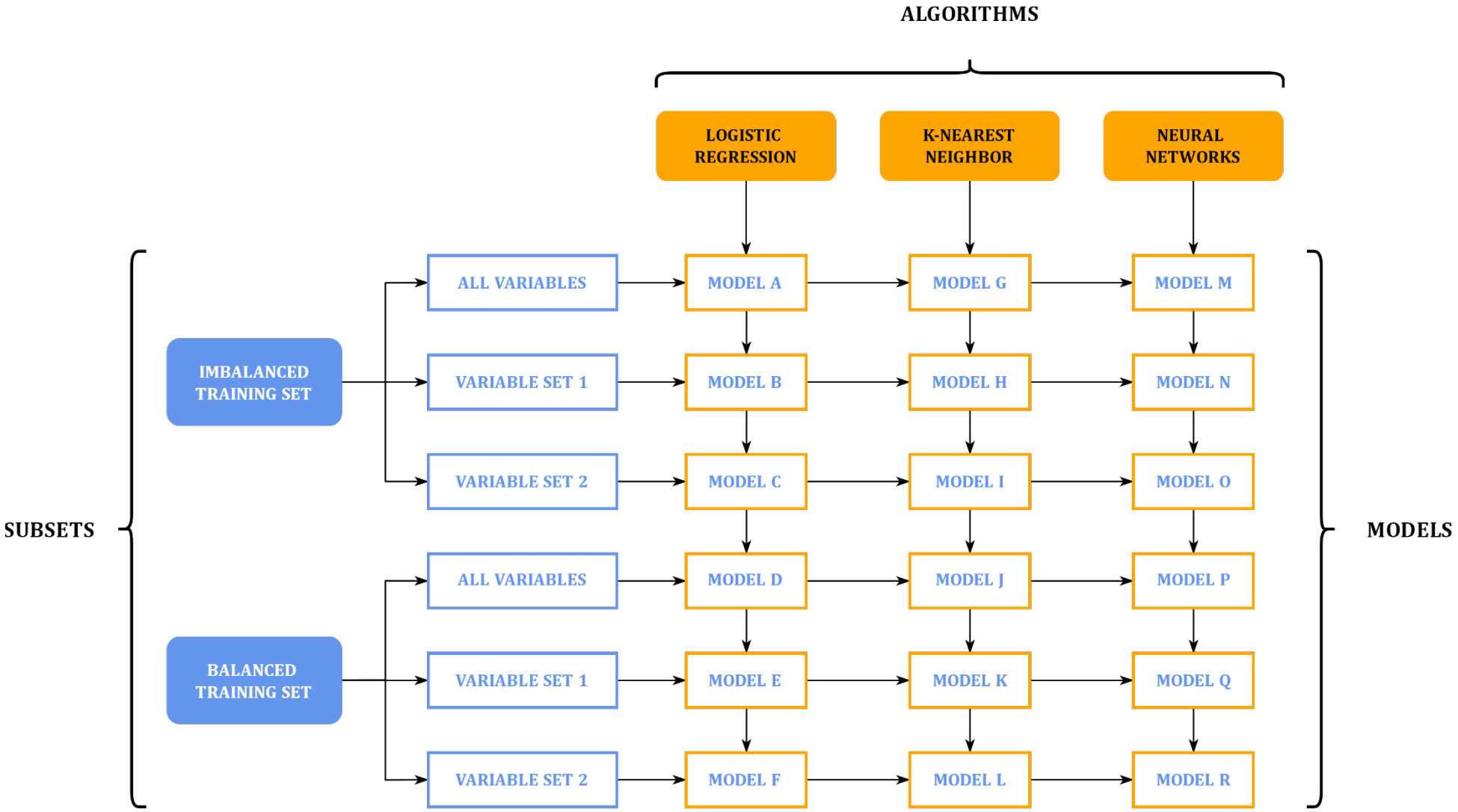


Figure 13: Diagram of Classification Approaches and Models



5. Classifications

Finally, with the purpose to find the best models to classify the portfolios with the high potential to pay off their loans, in the **Classifications**, six determined subsets are fed into three machine algorithms including Logistic Regression, k-Nearest Neighbor, and Neural Networks with suitable modifications to receive the best models evaluated by the confusion matrix in the "caret" package. In total, there are 18 models named from A to R were implemented. Figure 13 shows the three algorithms in use, six subsets made by different sets of selected variables and balance level of outcome variable's classes, as well as 18 models' names. The arrows do not connect relations but they show the order directions of approaches. This analysis presents in vertical order, in which the six subsets are subsequently fitted on each algorithm at the time for internal evaluation. Finally, we compare the best models in each algorithm. To avoid different results generated by the same model, a fixed number of seeds for the machine's automatic randomization is set at 2021, which is the number of years when the analysis is complete.

5.1. Logistic Regression

As for Logistic Regression, in which six different models namely from A to F using the imbalanced and balanced subsets with three selections of attributes including all variables, set 1, and set 2 were fitted in. One notice that with each model, the range of confusion matrix's cut-offs from 0.5 – 0.9 was used, but since we kept receiving similar results, 0.5 cut-offs made it in this paper for all models for more generalized and equal comparisons. From the color-coded accuracy measurements in Table 7 and Figure 14, it is easy to see that Model A is the one with the maximized accuracies when it comes to classifying both "defaulted" and "paid" with zero error in Type I and Type II. However, it may result from the disproportional

Figure 14: Four-Fold Plots of Confusion Matrices for Logistic Regression Models

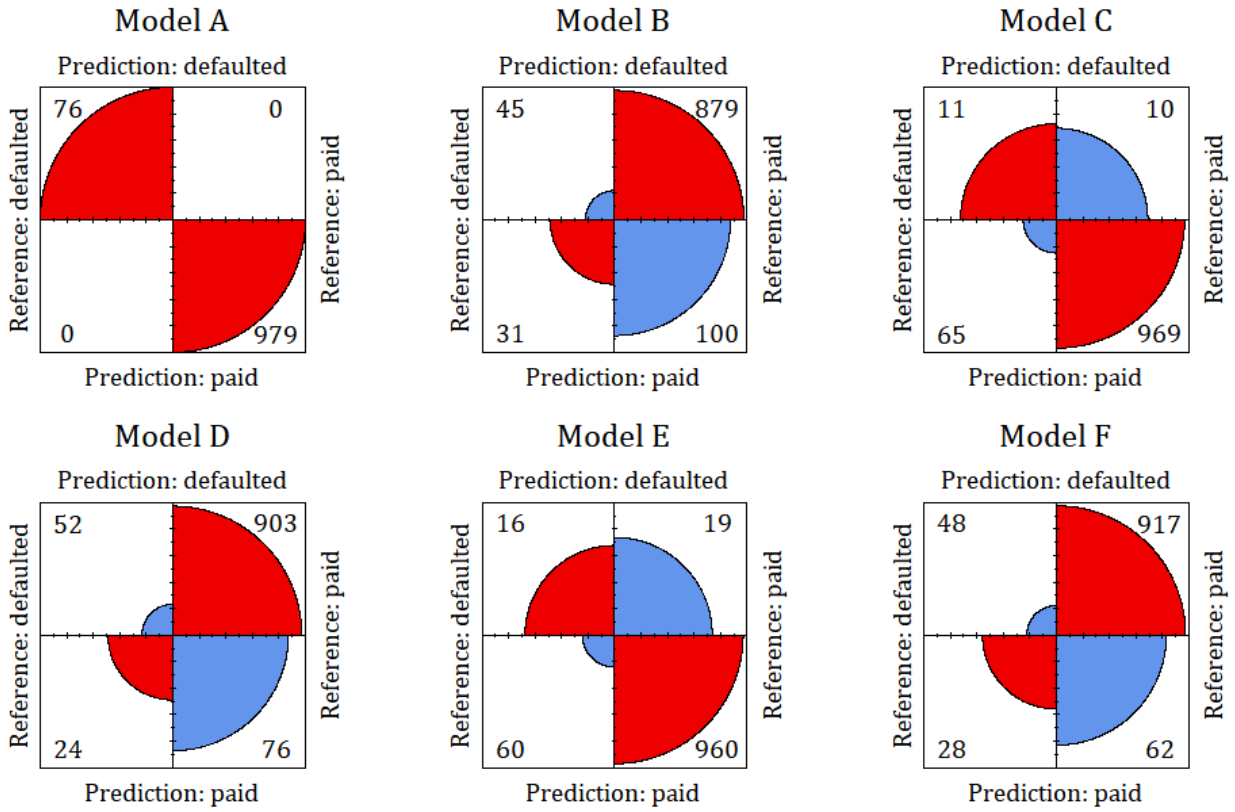


Table 7: Accuracy Measures of Logistic Regression Models

	Imbalanced Training Set			Balanced Training Set		
Variable	All	Set 1	Set 2	All	Set 1	Set 2
Model	A	B	C	D	E	F
Accuracy	1.0000	0.9290	0.9250	0.1370	0.1210	0.1040
95% CI	(0.997, 1)	(0.912, 0.944)	(0.908, 0.94)	(0.117, 0.16)	(0.102, 0.143)	(0.086, 0.124)
No Information Rate	0.9280	0.9280	0.9280	0.9280	0.9280	0.9280
P-Value [Acc > NIR]	<2e-16	0.4830	0.6670	1.0000	1.0000	1.0000
Kappa	1.000	0.2020	0.2540	(0.0500)	(0.0380)	(0.0480)
Mcnemar's Test P-Value	NA	4.51e-10	6.78e-06	<2e-16	<2e-16	<2e-16
Sensitivity	1.0000	0.9900	0.9810	0.1021	0.0776	0.0633
Specificity	1.0000	0.1450	0.2110	0.5921	0.6842	0.6316
Positive Predicted Values	1.0000	0.9370	0.9410	0.7634	0.7600	0.6889
Negative Predicted Values	1.0000	0.5240	0.4570	0.0487	0.0545	0.0497
Prevalence	0.9280	0.9280	0.9280	0.9280	0.9280	0.9280
Detection Rate	0.9280	0.9180	0.9100	0.0948	0.0720	0.0588
Detection Prevalence	0.9280	0.9800	0.9670	0.1242	0.0948	0.0853
Balanced Accuracy	1.0000	0.5670	0.5960	0.3471	0.3809	0.3475

Note: All models in this table were evaluated by the confusion matrix method with the seed 2021 for randomization, cut-off 0.5, and positive class "paid". Each accuracy measure is highlighted according to descending rank of best to last values in the top four out of six values with respective colors red, blue, green, and yellow.

distribution of outcome variable's classes that mislead the Logistic Regression to discard the minority class "defaulted" due to inferiority. To fix the problem with the "paid" class overthrowing the "default" class, the balanced set was used and the best model overall is D. However, the balanced subset only gives around 10% of the overall accuracy among its three models with the specificity higher than their sensitivity counterparts, which mean they have better ability to classify "defaulted" class. Therefore, if the business owner desire to find the potential defaulted portfolio, using the balanced set or specifically Model E with the highest specificity (68.42%) with the lowest error type II (60 out of 960) is recommended. However, for this paper, Model B (with the highest overall accuracy of 92.90% and sensitivity of 99%) is selected as the best Logistic Regression model.

5.2. K-Nearest Neighbors

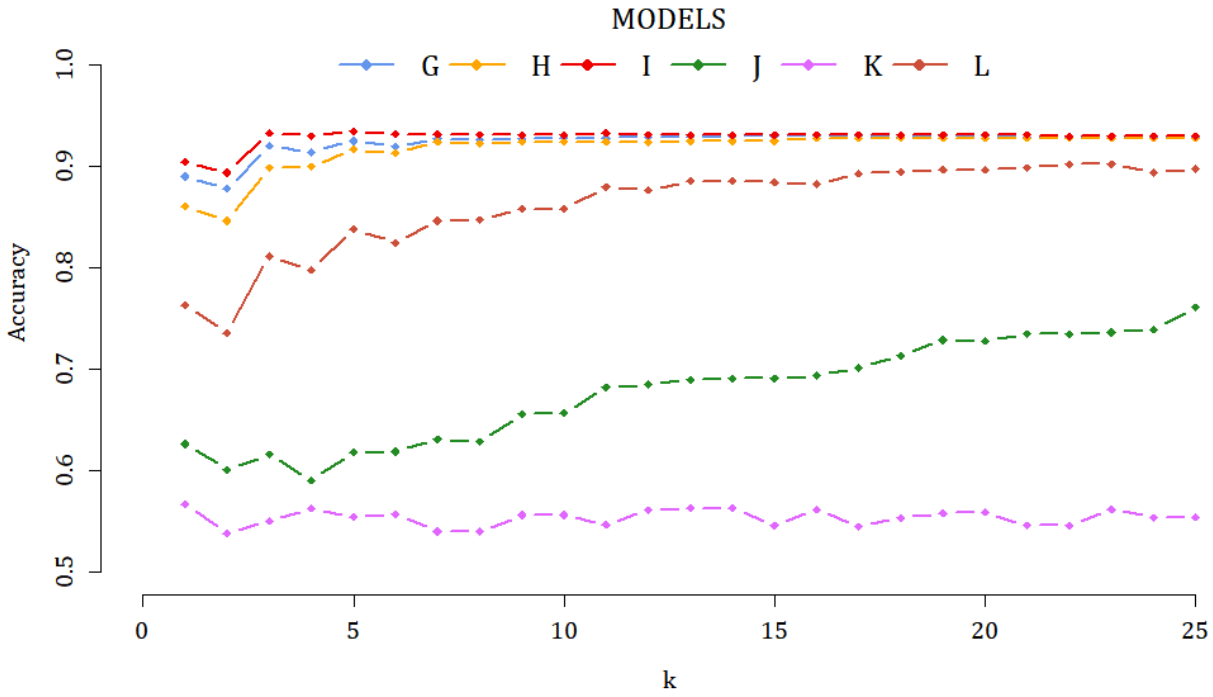
K-Nearest Neighbors (KNN) is based on a simple premise that any point (records) close to each other with the distance of k must be in the same class. In this algorithm, six models subsetted from balanced and imbalanced sets with three selection of variable sets are used. With k-Nearest Neighbors, the initial step is to find the optimal number of k which is calculated by the following equation:

$$\textbf{Equation 3: } \boldsymbol{Optimal\ k} = \frac{\sqrt{\boldsymbol{Total\ number\ of\ training\ records}}}{\mathbf{2}}$$

In this case, the optimal k is 25, so it is used as the maximum limit of k we would use. Then, I wrote a function so that R can automatically calculate the accuracy of each k from one to 2 5 and plot it into the line chart in Figure 15 with each line representing all accuracies of k's of six models from G to L. Generally, the three lines of models G, H, J belonging to the imbalanced set are over 80% bar while the rest belonging to the balanced ones have lower

overall accuracies. However, we should remember that with a simple premise of classifying based on distance, it is easy for KNN to only be able to find the majority class "paid" for distance computation. Whilst with the classes relative equivalent, each class's points are more populated so KNN will have more observations to learn and classify.

Figure 15: Accuracies of K Values in K-Nearest Neighbor Models



After plotting the accuracies of k's, we can find the best k for each model and, therefore, we could use KNN algorithms to classify the validation's observations with the cut-off of 0.5 because of the same reason in the **Logistic Regression** section (replacing cut-offs 0.5 – 0.9 gives similar results in all models). Figure 16 shows that all models, except for I, have a high ability to classify the correct "paid" records but also extremely high errors in type II, and none of them have an outstanding ability to classify the "defaulted" class. Therefore, for the k-Nearest Neighbor algorithms, Model I is the best in terms of overall scores of 93.3% of accuracy, 99.5% of sensitivity, and 13.2% of specificity.

Figure 16: Four-Fold Plots of Confusion Matrices for K-Nearest Neighbor Models

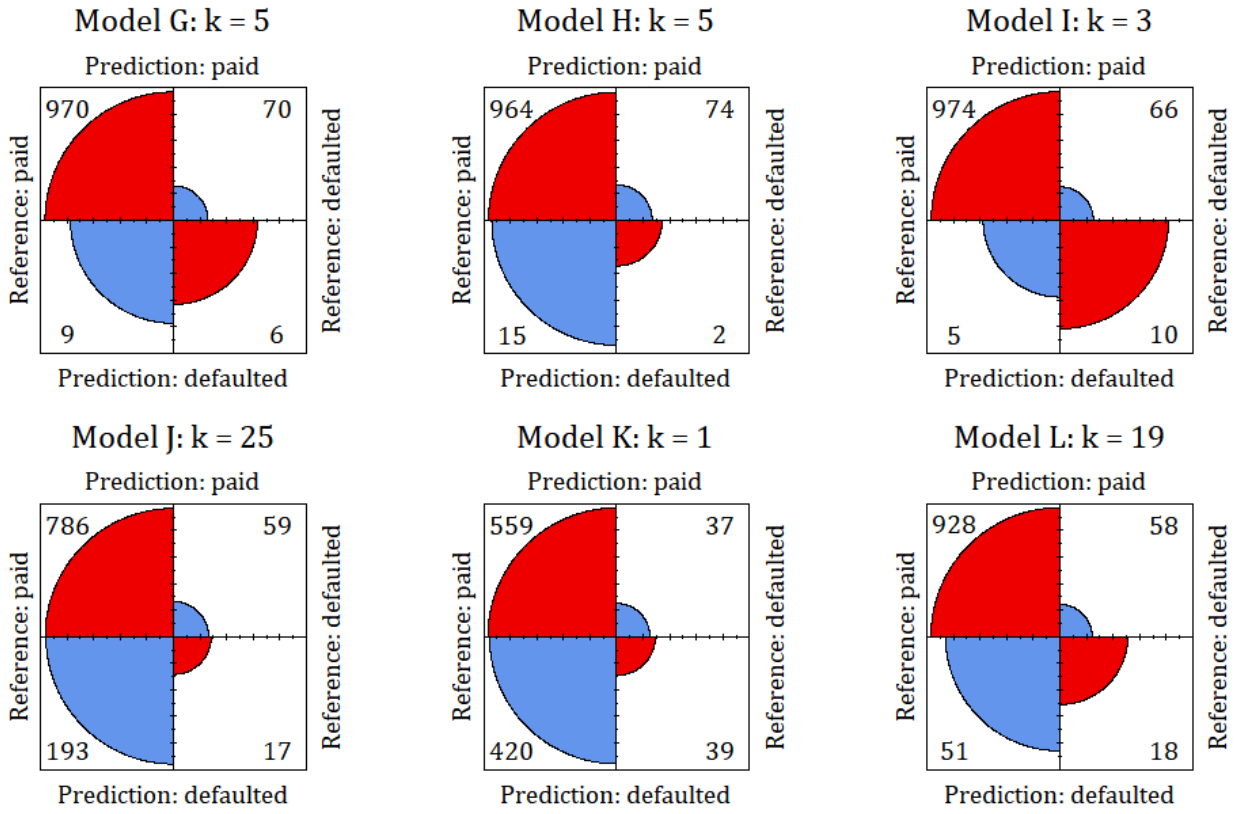


Table 8: Accuracy Measures of K-Nearest Neighbor Models

	Imbalanced Training Set			**Balanced Training Set**		
Variable	All	Set 1	Set 2	All	Set 1	Set 2
Model	G	H	I	J	K	L
k	5	5	3	25	1	19
Accuracy	0.9250	0.9160	0.9330	0.7610	0.5670	0.8970
95% CI	(0.908, 0.94)	(0.897, 0.932)	(0.916, 0.947)	(0.734, 0.787)	(0.536, 0.597)	(0.877, 0.914)
No Information Rate	0.9280	0.9280	0.9280	0.9280	0.9280	0.9280
P-Value [Acc > NIR]	0.6670	0.9430	0.3000	1.0000	1.0000	1.0000
Kappa	0.1110	0.0170	0.2010	0.0150	0.0250	0.1930
Mcnemar's Test P-Value	1.47e-11	7.85e-10	1.07e-12	<2e-16	<2e-16	0.5650
Sensitivity	0.9908	0.9847	0.9950	0.8030	0.5710	0.9480
Specificity	0.0789	0.0263	0.1320	0.2240	0.5130	0.2370
Positive Predicted Values	0.9327	0.9287	0.9370	0.9300	0.9380	0.9410
Negative Predicted Values	0.4000	0.1176	0.6670	0.0810	0.0850	0.2610
Prevalence	0.9280	0.9280	0.9280	0.9280	0.9280	0.9280
Detection Rate	0.9194	0.9137	0.9230	0.7450	0.5300	0.8800
Detection Prevalence	0.9858	0.9839	0.9860	0.8010	0.5650	0.9350
Balanced Accuracy	0.5349	0.5055	0.5630	0.5130	0.5420	0.5920

Note: All models in this table were evaluated by the confusion matrix method with the seed 2021 for randomization, cut-off 0.5, and positive class "paid". Each accuracy measure is highlighted according to descending rank of best to last values in the top four out of six values with respective colors red, blue, green, and yellow.

5.3. Neural Networks

The Neural Networks algorithm, as the result, is the trickiest one among the three. It has many pros such as the ability to handle complex data sets with numerous input variables of different types of data sources such as images, audio, tabular, etc. It also can deal with categorical or numeric continuous or discrete target variables. However, Neural Networks is also the most time-consuming in terms of computational time and infrastructure resources among the three algorithms, while it also requires the analyst to take special care to modify different thresholds for different types of hidden layers/nodes modification. Hence, the cons of Neural Networks surpass its pros.

In this analysis, binary dummy variables of the "STATUS" target were not created. Instead, the "BAD" column is used as a target with binary values of "1" for "defaulted" and "0" for "paid". Then, since the Home Equity contains different number scales in each variable, I normalize all numerical variables using the Min-Max normalization method automatically iterate by manual R function, with X' is a normalized numeric value, X is the original value while A is all values in the same variable, as follows:

$$\textbf{Equation 4: } X' = \frac{X - Min(A)}{Max(A) - Min(A)}$$

Within the four models M to R, I also tried six combinations of hidden layers and nodes shown in Table 9, as well as, a range of cut-off values between 0.5 to 0.9 with the increment of 0.1. While exploring those different combinations, I encounter one of the famous problems only encountered with Neural Networks with the warming below:

```
Warning: Algorithm did not converge in 1 of 1 repetition(s) within the stepmax.
Error in cbind(1, pred) %*% weights[[num_hidden_layers + 1]] :
  requires numeric/complex matrix/vector arguments
```

This problem[4] happens with giving Neural Networks unsatisfactory thresholds (or stepmax), so when using the different threshold besides the default 0.01, I was able to receive no error but the results are still similar to those using the user's specification of 0.5 cut-off and 0.01 threshold. Therefore, I believe it is not wise to invest time and computation resources to try more combinations, but instead, I will evaluate the six models using my default specification since it has already taken 5 and up to 90 for the Neural Network algorithm to learn. Table 9 shows the different models and layers/nodes combinations that work with 0.01 threshold and 0.5 cut-offs. Before looking at the accuracy measurements in Table 10, it is clear that Models N, P, Q do not have any ability to find the "defaulted" profiles, and set 1 of variables are no longer in use in this algorithm, despite there 100% of their ability to recognize "paid" profiles. Among the rest of the models, Model R shows the best results with 100% of specificity and low errors of both types I and II while still able to retain the ability to classify "defaulted" records.

Table 9: Cut-offs matrix with default thresholds = 0.01 of Neural Networks Models

Hidden Layers/Nodes	MODEL					
	M	**N**	**O**	**P**	**Q**	**R**
1	0.5	0.5	0.5	0.5	0.5	0.5
2		0.5	0.5	0.5	0.5	0.5
1:1		0.5	0.5	0.5	0.5	0.5
1:2		0.5		0.5	0.5	0.5
2:1		0.5	0.5	0.5	0.5	
2:2		0.5	0.5		0.5	0.5

[4]https://stackoverflow.com/questions/65388609/neuralnet-requires-numeric-complex-matrix-vector-arguments

Figure 17: Four-Fold Plots of Confusion Matrices for Neural Network Models

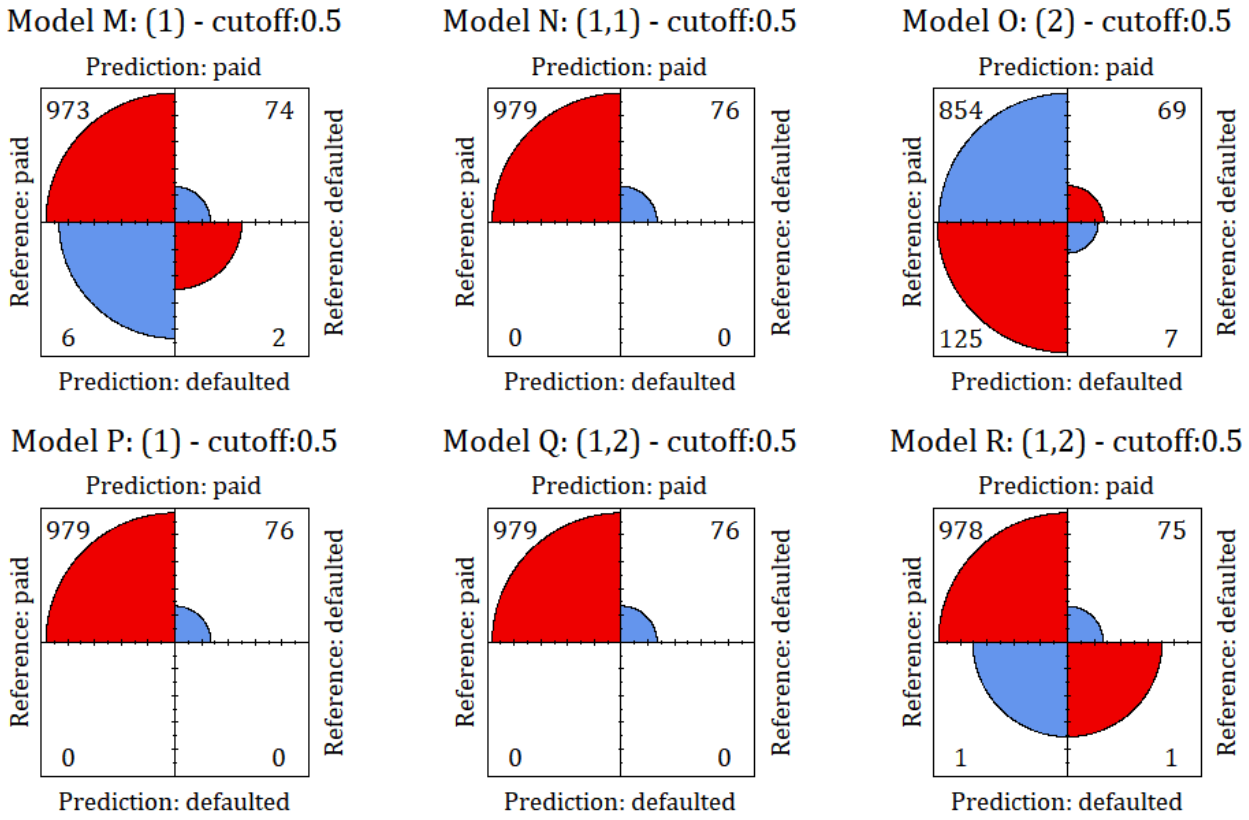


Table 10: Accuracy Measures of Neural Networks Models

	Imbalanced Training Set			Balanced Training Set		
Variable	All	Set 1	Set 2	All	Set 1	Set 2
Model	M	N	O	P	Q	R
Hidden Layers/Nodes	1	1:1	2	2	1:2	2:1
Cut-off	0.5	0.5	0.5	0.5	0.5	0.5
Accuracy	0.9240	0.9280	0.8160	0.9280	0.9280	0.9290
95% CI	(0.907, 0.939)	(0.911, 0.943)	(0.791, 0.839)	(0.911, 0.943)	(0.911, 0.943)	(0.912, 0.944)
No Information Rate	0.9280	0.9280	0.9280	0.9280	0.9280	0.9280
P-Value [Acc > NIR]	0.7080	0.5300	1.0000	0.5300	0.5300	0.4830
Kappa	0.0340	0.0000	(0.0270)	0.0000	0.0000	0.0240
Mcnemar's Test P-Value	6.84e-14	<2e-16	7.86e-05	<2e-16	<2e-16	<2e-16
Sensitivity	0.9939	1.0000	0.8723	1.0000	1.0000	1.0000
Specificity	0.0263	0.0000	0.0921	0.0000	0.0000	0.0132
Positive Predicted Values	0.9293	0.9280	0.9252	0.9280	0.9280	0.9288
Negative Predicted Values	0.2500	NaN	0.0530	NaN	NaN	1.0000
Prevalence	0.9280	0.9280	0.9280	0.9280	0.9280	0.9280
Detection Rate	0.9223	0.9280	0.8095	0.9280	0.9280	0.9280
Detection Prevalence	0.9924	1.0000	0.8749	1.0000	1.0000	0.9991
Balanced Accuracy	0.5101	0.5000	0.4822	0.5000	0.5000	0.5066

Note: All models in this table were evaluated by the confusion matrix method with the seed 2021 for randomization and positive class "paid", while the cut-off values will vary depending on each model. Each accuracy measure is highlighted according to descending rank of best to last values in the top four out of six values with respective colors red, blue, green, and yellow.

6. Conclusions

Finally, Table 11 shows that B, I, R are the best models among Logistic Regression, K-Nearest Neighbor, and Neural Networks algorithms for classifying good "paid loan" profiles for the Home Equity loan data set. In terms of data-driven reasons, Model R of Neural Networks results in the highest accuracy for this paper's purpose. Since this model did not use the imbalanced set, it reduces the chances of biased computation from machine learning algorithms because our data set has high disproportionate distribution leaning toward the "paid" class. However, in real-world, approving loan might sometimes encounter questions of clarification in terms of classification, Neural Network, which is also called the "Black Box" method because of its unclarity, may not satisfy investigations with the explanation of some complex mathematical equations, and therefore, gives loaner trouble. For that reason, I will have more confidence in using the next best model which is I with K-Nearest Neighbor.

Table 11: Summary of Selected Best Models

Specification	Algorithms		
	Logistic Regression	**K-Nearest Neighbor**	**Neural Networks**
Model	B	I	R
Data Set	Imbalanced	Imbalanced	Balanced
Variable Set	Set 1	Set 2	Set 2
Cut-off	0.5	0.5	0.5
Other Modification	NA	k = 3	Layers/Nodes (2,1)

With the limitation of industrial expertise and time, I am not able to fully analyze the data set to serve the purpose of finding the best models for "paid" loan profile classification to my heart desire of factory. However, there are several suggestions hereby to improve further research as follows:

In terms of data collection and data set, there are several suggested modifications including the addition of identifiers, new aggregated significant attributes, and a variety of classes in variables. First, we can see that there are no time-specific variables to know the time of collection or data entry. If we were to know more about time and geographical indicators, we might have been able to add, exclude or aggregate appropriate variables to suit the current industry standards of approving loans. Second, it appears that the variety of profiles in this data set is with the white-collar[5] professions while many pieces of research show that at the same professional level, blue-collar[6] occupations earn equivalent or more wage per hour. Therefore, if we only consider the income factor as the potential to not default on the loan, the blue-collar professions might have the same, if not higher, possibility to be accepted for home equity loans. Hence, inclusion factors in classes of various attributes may be reconsolidated for this data set.

In terms of industrial expertise in Finance, the scorecard, one of the most recent popular methods developed just a few decades ago by using user-determined different bins for continuous-valued attributes, might also be a great way to classify the Home Equity data set. Another way is to analyze credit score-related values in deep and give appropriate modifications. An advantage of having statistical credit scoring models is consistency. We no longer have to rely upon the experience, intuition, or common sense of one or multiple business experts. Now it's just a mathematical formula, and the formula will always evaluate in the same way if given the same set of inputs, like age, marital status, income, and so on.

---

[5] White-collar workers are known as suit-and-tie workers who work in service industries and often avoid physical labor.

[6] The blue-collar stereotype refers to any worker who engages in hard manual labor, such as construction, mining, or maintenance.

However, using expertise must be taken with great regard since it requires the high intuition ability of a professional in the field to give the right judgments.

In terms of building data-driven models, it is advisable to try more different algorithms well-known for classification in the financial area such as association, regression trees that have more clarity of machine learning. Moreover, many other models proving methods such as bagging and boosting, and models evaluation such as ROC curve, gain-lift charts should also be implemented to test the results. Other manual programming languages or semi to fully automatic applications such as SAS or SAP Predictive Analytics are available tools that may give better results than R.

Reference

Baesens, B., Roesch, D., & Scheule, H. (2018). *Credit risk analytics: Measurement techniques, applications, and examples in Sas*. WILEY.

Dinov, I. D. (2018). *Data Science and Predictive Analytics: Biomedical and health applications using R*. Springer.

Scheule, H., Rösch Daniel, & Baesens, B. (2017). *Credit risk analytics: The R companion*. CreateSpace, a DBA of On-Demand Publishing, LLC.

Shmueli, G. (2018). *Data mining for Business Analytics: Concepts, techniques, and applications in R*. John Wiley & Sons.

Do, H.X., Rösch, D. and Scheule, H., 2019. *Liquidity constraints, home equity and residential mortgage losses.* Journal of Real Estate Finance and Economics.

Do, H.X., Rösch, D. and Scheule, H., 2018. *Predicting loss severities for residential mortgage loans: A three-step selection approach.* European Journal of Operational Research.

Rösch, D. and Scheule, H., 2010. *Downturn credit portfolio risk, regulatory capital and prudential incentives.* International Review of Finance, 10(2), pp.185-207.

IFRS 9/ CECL:

Krüger, S., Rösch, D. and Scheule, H., 2018*. The impact of loan loss provisioning on bank capital requirements.* Journal of Financial Stability, 36, pp.114-129.

Aniruddho “Oni” Sanyal, Phoenix Computing Solutions: Using SAS Studio (via SAS OnDemand for Academics) for “Credit Risk Analytics”

https://www.statmethods.net/stats/descriptives.html

https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html

https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/

https://www.creditkarma.com/advice/i/what-does-derogatory-mean

http://www.sthda.com/english/wiki/ggplot2-themes-and-background-colors-the-3-elements

https://www.lexingtonlaw.com/education/derogatory-marks

https://www.investopedia.com/terms/s/stepwise-regression.asp

https://jaiprakashml.medium.com/credit-risk-application-scorecard-modeling-using-r-f6e9f842bdd9

https://cran.rproject.org/web/packages/cvms/vignettes/Creating_a_confusion_matrix.html

https://www.investopedia.com/articles/wealth-management/120215/blue-collar-vs-white-collar-different-social-classes.asp