ABSTRACT

Business data is information gathered and used in an organisation for assistance in decision-making. It can be saved in a database or spreadsheet in order to acquire insight, spot trends, and make wise decisions. Thus, the purpose of business data analysis is to discover business needs, analyse problems, and recommend solutions to enhance growth and efficiency. Credit card default datasets provide significant insights and information that assist financial institutions and credit card firms in managing credit risk, making educated lending decisions, preventing fraud, optimising portfolio performance, and ensuring regulatory compliance. The dataset "Default of Credit Card Clients" was uploaded by I-Cheng Yeh and is available at the UCI Machine Learning Repository. The dataset is a classification problem that determines whether the credit card holder will default or not. There are 25 attributes in total, and the raw dataset comprises 30,000 rows. There are 23,362 non-default credit card clients and 6,638 default credit card clients among the rows. There are five models used to compare their performance to determine which model is preferred for binary classification. Logistic Regression, K-Nearest Neighbour, Gaussian Naive Bayes, Decision Tree, and Extreme Gradient Boosting are among the models used. The results show that Gaussian Naive Bayes is the most successful in forecasting credit card holder default. However, when all of the algorithms were compared, the result was determined that there were no substantial differences between them. The accuracy and AUC values of all the models are more than 75%. As a result, this report shows that all of the algorithms here predict well with the chosen feature. The model is then used to construct a scorecard to assess the user's creditworthiness, allowing the financial loss from the credit card company to be avoided.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS / FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

AUC	Area Under ROC Curve
CI	Confidence Interval
DT	Decision Tree
d_i	Standardize Residuals / Semistudentized Residuals
EDA	Exploratory Data Analysis
e_i	i-th residual
FN	False Negative
FP	False Positive
GNB	Gaussian Naive Bayes
H_0	Null Hypothesis
H_1	Alternative Hypothesis
h_{ii}	Leverage of the i-th data point
IQR	Interquartile Range
IV	Information Value
k	Numbers of independent variable
KDD	Knowledge Discovery in Databases
KNN	K-Nearest Neighbor
LDA	Linear Discriminant Analysis
LR	Logistic Regression
MD	Model Development
MSE	Mean Squared Error of the model
n	Numbers of observation
N	node
NB	Naive Bayesian
PCA	Principal Component Analysis
PD	Probability of Default
PLS	Partial Least Squares
r_i	Studentized Residuals
R^2	R-Squared
R^2_{Adj}	Adjusted R-Squared
ROC	Receiver Operating Characteristic
TN	True Negative

TP	True Positive
t-SNE	t-Distributed Stochastic Neighbour Embedding
t_i	R-Student / Studentized Deleted Residuals
vi	Each test feature has the value
WoE	Weight of Evidence
XGBoost	Extreme Gradient Boosting
SS_E	sum of squares error
SS_R	sum of squares regression
SS_T	sum of squares total
z	Critical value

LIST OF APPENDICES

CHAPTER 2

INTRODUCTION

2.1 General Introduction on Business Data

In the reality of data science and analytics, "business data" refers to information gathered about a company's operations that can aid in making wise decisions. (Provost and Fawcett, 2013). These data points can come from a range of sources, such as transactions, customer engagements, and social media (Provost and Fawcett, 2013).

In order to ensure the accuracy, integrity, and relevance of data, a key aspect of business data is its quality and governance (Provost and Fawcett, 2013). Without these data, organizations can make bad decisions with serious consequences (Beynon-Davies, 2019). Companies can avoid mistakes and improve their decision-making by maintaining high data quality standards (Beynon-Davies, 2019)

Valuable insights, such as financial performance, customer behavior, and market trends, can be provided through the analysis of business data and aspects of a company's operations (Churchall and Walker, 1972). These insights can assist businesses in making informed decisions and implementing strategies to improve their business performance (Churchall and Walker, 1972) which most of them are not observable to the naked eye. Companies can gain a competitive advantage, respond quickly to changing market conditions, and identify new growth opportunities by utilizing business data (Churchall and Walker, 1972).

In order to identify patterns and trends that can help companies make better decisions, business data research often requires the analysis of large amounts of data. To improve operations, marketing strategies, and profitability, discovering hidden insights is needed. Advanced statistical and machine learning techniques can be used to uncover hidden insights (Provost and Fawcett, 2013). Therefore, companies must recognize the importance of business data and invest in its collection, storage, analysis, and governance.

2.2 Credit Card Default

Credit cards are considered business data because they can be used by the majority of households at all economic levels. According to Warwick and Mansfield (2000), credit sales are now 50% to 100% greater than cash purchases. Consumers today live at or above the financial margins, and they routinely spend all or more of what they earn and are unaware that their spending consistently exceeds their income (Warwick and Mansfield, 2000). A credit card account becomes delinquent before it goes into default. This happens after 30 days of late payments. Default usually occurs after 6 months of failing to make payments, indicating that your credit card is seriously in arrears.

Employers still commonly refer to credit reports on job applicants as part of the interview process. The credit report, not the credit score, is the primary concern of the employer. A series of delinquencies or defaults can reduce your competitiveness. The bank's loan interest rate will also be higher for the holder of this credit card or it may even will reject the loan request. The most important goal for financial institutions is to identify customers who are at high risk of default so that bank can take proactive measures to prevent defaults and minimise losses.

A creditworthiness assessment is a basic procedure used by banks or financial institutions to determine the risk associated with a borrower in order to prevent financial loss. It typically determines the credibility of this borrower by classifying credit risk as "high risk" or "low risk" based on demographics and payment history (Yap et al., 2011). To improve the accuracy of credit assessment, data mining techniques and analysis of past data are used to infer and investigate default prediction and creditworthiness (Yap et al., 2011). The goal of these models is to prevent financial losses through early intervention and a method for developing credit scoring models will be proposed. The main goal of this research is to examine credit card defaults based on cardholder behaviour. Classification methods were widely applied to credit card defaults in the early stages (Li et al., 2019). Shi, for example, provided data mining methods for classifying the behaviour of credit card holders using multi-criteria linear programming (Li et al., 2019). Lim and Sohn also used clustering and neural network methods to construct a dynamic scoring model that selects influencing variables based on logistic regression and uses these influencing variables to

train a target variable. The models they studied all included the customer's marital status, whether they owned a home, the distance between their birthplaces, and so on (Li et al., 2019).

Credit risk scorecards are often used to assess the creditworthiness of borrowers (Siddiqi, 2017). The higher the score, the more credible the borrower is. Careful consideration needs to be given when developing a credit risk scorecard for a corporate client, which includes financial data including revenue, profitability, cash flow, debt-to-equity ratio and other financial ratios (Machado and Karray, 2022).

Machado and Karray (2022) tested whether a hybrid machine learning approach combining decision trees, logistic regression, and neural networks could improve the accuracy of credit risk assessment. As a result of this test, lending judgments became more accurate due to the reduced frequency of false positives and false negatives in their credit risk assessment. In conclusion, creditworthiness assessment is an important process for banks or financial institutions to use when determining credit risk. Credit scoring models and credit risk scorecards can be frequently used to evaluate creditworthiness and integrated into operational data.

2.3 Problem Statement

Credit card default rates still remain high today, despite the widespread use of credit scoring models by banks and other financial institutions, the significant financial losses still resulted. Therefore, there is a need for an accurate and effective credit scoring model that must be able to predict the probability of credit card defaults based on historical business data. The study intends to address the problem of high credit card default rates and improve the overall financial performance of the institution by analyzing data and developing a scorecard.

2.4 Aim and Objectives

Firstly, this report is to identify which data mining algorithm is more suitable for binary classification output since our target is to find whether the user is defaulting for next month. The data mining algorithms include Logistic

Regression, K-nearest neighbor, Gaussian Naïve Bayes, Decision Tree and Extreme Gradient Boosting.

Secondly, some data preprocessing has to be done before running the model, such as feature selection and feature transformation. In business data, Weight of Evidence (WoE) and Information Value (IV) is more suitable for feature selection and feature transformation.

Thirdly, the dataset used here is about to create a scorecard to determine which user is more trustworthy. Hence, staff will determine whether the user will default on the credit card for the next month based on the scorecard.

2.5 Scope and Limitation of the Study

2.5.1 Introduction of Scope of Study

The primary goal of this project is to create Probability of Default (PD) risk rating models that are personalized to the banks or financial institutions' portfolio history and are usually known as scorecards. The end goal is to standardize and enhance credit rating processes. As a result, the bank's credit approval process would be more efficient, as well as the bank's or financial instituitions' credit risk management.

How to know whether the credit card defaults? In the financial system, there is a scorecard point to analyze whether this person is available and qualified to receive the amount of a credit card. The probability of credit card default will be considered in this scorecard point. Creating a model and count the accuracy based on the binning and weightage that are used to calculate the probability of default.

Then, data binning and data correlation should be applied to get a reasonable weightage and default probability. As a result, the accuracy of the default probability will be obtained. The greater the accuracy, the greater the confidence in this model. By using these binning and categorizing techniques, a standard scorecard that will assist banking and financial systems in easily approving or rejecting the amount of a credit card can be created.

Creating a scorecard is divided into two parts: Model Development (MD) and Probability Default (PD). In Project I, model development is built. In the model development process, outliers are capped at the lower (5%) or upper (95%) quantiles of the portfolio. Then, a correlation analysis must be performed.

The Pearson correlation coefficient method is used to calculate the linear correlation between two groups of variables. Python code is used to show virtualization because it has a crosstab function, which is a more straightforward way to see the relationship between variables. The continuous variable must then be binned into categorical data using the Weight of Evidence (WoE) transformation. Using this method, financial factors can be distributed equally, and the linear trend is selected, either decreasing or increasing. Typically, binning is about 3 to 5 groups of continuous financial factors since 2 of them are insufficient and inaccurate for predicting, and more than 5 bins are too complicated.

In Probability Default (PD) which will be done in Project II, final weightage is referred to from Model Development (MD) as factor weightage. The weightage of those factors greater than or equal to 0.5 is increased to 1. For example, CCRIS has a weightage of 13.50%, and thus its factor weightage is 14%. This means that the highest score for CCRIS in a 100% scorecard is 14% when combined with other variables. The closer the CCRIS score is to 14%, the more likely it is that the person will be trusted. Of course, other variables must also be considered and hence accumulate the score to determine whether the customer should default on their credit card or not.

2.5.2 Scope and Limitation of the Study (Model Development)

The collection of data is the first step in the data analysis process (Martinez et al., 2021). Once the data is collected, it needs to be organized and stored, as this data access and analysis will be access for future use. Key steps in identifying patterns, trends, and relationships among variables can be explored and visualized through data (Tukey, 1977). These steps help researchers gain insight into the data and address issues such as missing values or outliers. Pre-processing techniques such as cleaning, transformation, feature selection, integration and reduction prepare the data for analysis (Martinez et al., 2021). Data mining is a powerful tool to extract valuable insights from large databases (Yap et al., 2011). Companies can extract useful patterns or rules from data to drive strategic initiatives and make informed industry decisions (Yeh and Lien, 2009). Data mining yielded significant benefits, which include better decisions, lower costs, and greater efficiency (Yap et al., 2011). As data generation and

storage continue to grow, data mining is expected to become even more valuable in today's information-driven economy (Yap et al., 2011).

An important step in data analysis and data mining is data pre-processing. It is responsible for converting the original data into a format suitable for analysis or modeling (Mishra et al., 2020). Data cleaning is a key step in data preprocessing. Data cleaning includes identifying and handling errors or inconsistencies in the data. Examples of data that needs cleaning are missing values, outliers, or duplicate records (Davis and Clark, 2011). To ensure accurate results from analysis or modeling, the quality of the data is also critical. Mishra et al. (2020) argue that a combination of preprocessing techniques may be more effective in preparing data for analysis or modeling, especially for complex real-world data, which is often nonlinear. These techniques can be combined to improve the accuracy of the results of data analysis or modeling.

Feature selection is also a crucial step in data preprocessing. It involves selecting relevant features or variables for analysis while removing irrelevant or redundant features (Davis and Clark, 2011). Principal component analysis (PCA), linear discriminant analysis (LDA), and t-distributed Stochastic Neighbour Embedding (t-SNE) are examples of dimensionality reduction techniques (Davis and Clark, 2011). To identify patterns and relationships in data after it has been preprocessed, data mining techniques such as PCA and Partial Least Squares (PLS) regression (Lommen, 2009). Predictions or classifications of samples based on these structures and trends can be made using these techniques (Lommen, 2009). This highlights the importance of data preprocessing in producing meaningful insights from the data.

In conclusion, data analysis entails gathering, organizing, exploring, and preparing information for analysis. Cleaning and feature selection are important preprocessing techniques for achieving accurate results. Data mining can uncover useful patterns and rules. Dimensionality reduction techniques like PCA, LDA, and t-SNE aid in the identification of underlying structures and trends. Meaningful insights require reliable results. However, since this dataset is a business dataset, feature selection and dimensionality reduction techniques are ineffective. This is because doing so will almost certainly result in the loss of critical data. As a result, after cleaning the data, the Information Value (IV)

will be used to select important features. Weight of Evidence (WoE) will take the place of feature transformation and feature binning.

2.6 Contribution of the Study

This report will assist in the discovery of the optimal model for forecasting credit card default. As a result, construct the scorecard using the features specified in the model. Based on the scorecard, the credit card company can evaluate whether or not the customer will default on their credit card the next month. If the model has greater accuracy, it suggests that the feature chosen to run in the model is extremely beneficial in forecasting if a credit card customer will default the next month. The scorecard is developed based on the model and characteristics, so financial loss may be avoided when credit card companies adhere to the scorecard.

2.7 Outline of the Report

Logistic Regression, K-Nearest Neighbour, Gaussian Naive Bayes, Decision Tree, and Extreme Gradient Boosting are the models utilised to compare their performance. A confusion matrix with the fewest false positives and false negatives is preferred. Furthermore, the confusion matrix will yield accuracy, sensitivity, precision, recall, and so on. The classification report contains all of the results. This dataset focuses on accuracy since it is used to examine how well the model learns the dataset's data pattern and how well it can anticipate future data. Additionally, a bigger Area Under Receriver Operating Characteristic (ROC) Curve (AUC) is preferred. Based on a sample and a specified confidence level, a confidence interval is used to estimate the range with an upper and lower bound. Then, using all of the information, create a scorecard using the ratio technique.

CHAPTER 3

LITERATURE REVIEW

3.1 Introduction of Data Mining Algorithm

The most important key to dealing with big data is to scale up the large volume of data and provide the most reliable predictions or decisions. Data mining, also known as Knowledge Discovery in Databases (KDD), is a technique for making use of data (Jothi et al., 2015). Data mining is the process of detecting patterns and extracting meaningful knowledge from a lot of data (Jothi et al., 2015). There are two data mining tasks: classification and regression (Cortez and Embrechts, 2013b). Both tasks involve supervised learning to create a data-driven model with one target variable (Cortez and Embrechts, 2013b). These tasks can result in a variety of learning models or algorithms, which are referred to as data mining models (Cortez and Embrechts, 2013b). Each model has its own advantages and disadvantages (Cortez and Embrechts, 2013b), and the algorithm used is determined by the problem and data characteristics. Data mining models include predictive models and descriptive models (Jothi et al., 2015). Predictive models use supervised learning functions to predict the unknown, whereas descriptive models use unsupervised learning functions to discover data patterns (Jothi et al., 2015). Supervised learning can learn from labeled data and thus provide a specific output or target value. Labeled data is data that has previously known output values. Unsupervised learning discovers patterns and structure in data by learning from unlabeled data. Since the previous target variable is provided in this dataset, supervised learning algorithms must be used here.

3.1.1 Logistic Regression (LR)

Logistic regression (LR) is a method that employs a collection of features to analyse data that is either continuous, discrete, or a combination of both continuous and discrete data types (Jothi et al., 2015). Furthermore, logistic regression requires a binary target variable in the dataset (Jothi et al., 2015). The logistic function is then applied to the linear combination of the inputs (Jothi et al., 2015).

The relationship between two or more variables can be obtained from regression analysis (Stoltzfus, 2011). Linear regression is a type of regression that is commonly used to analyse continuous target variables and assumes that the relationship between the dependent variable and the independent variables are a straight line, either increasing or decreasing (Stoltzfus, 2011). Multivariate linear regression is used to determine the impact of multiple factors on an outcome at the same time (Stoltzfus, 2011). On the other hand, Logistic Regression is used for binary outcomes like mortality, where the estimated probability of falling into one of two categories is calculated (Stoltzfus, 2011). Many features of linear regression are retained in logistic regression, but the logit scale is used to solve the problem of predicted values that fall outside the range of 0 and 1. It also provides a probability value between 0 and 1. Logistic regression requires a linear connection between the independent variables and the log-odds of the occurrence. It also assumes that the observations are independent of each other and that there is little to no multicollinearity among the independent variables. The choice of independent variable and model building strategy are significant factors in making sure that logistic regression produces an accurate model.

The logistic function curve represents the probability of something. In this case, it will be used to predict if the credit card user will default for the next month.

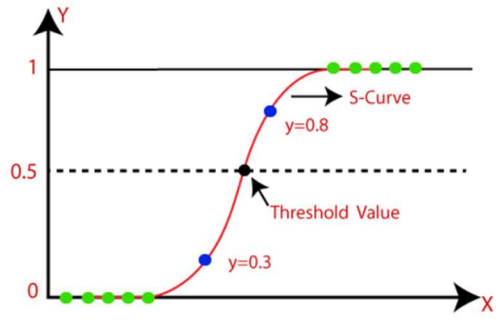


Figure 3.1.1.1: Logistic regression curve, S-Curve (Cramer, 2002)

As shown in Figure 2.1.1.1, the S-curve visualised in Logistic Regression is the sigmoid function or logistic function (Cramer, 2002). The sigmoid function is a mathematical function that transforms the expected values into probabilities within a range of 0 and 1. The threshold value is 0.5, which implies that if the probability is greater than 0.5, the expected value is 1. If the probability is less

than 0.5, the expected value is 0. For example, in this project, as the goal is to predict whether the credit card holder is in default for the month, 1 indicates that the credit card holder is in default for the next month, and 0 indicates that the credit card holder is not in default for the next month.

The logistic regression equation can be produced through linear regression. The following is the mathematical procedure for obtaining the logistic regression equation:

The equation for the straight line is written as Figure 2.1.1.2

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + ... + \beta_n x_n$$

Figure 3.1.1.2: Linear combination of the input features where y represent the target variable and the β_0, β_1, β_2, …, β_n

i. In Logistic Regression, y, which is the predicted value can be 0 or 1 only, hence, the equation can be write as Figure 2.1.1.3.

$$\frac{y}{1-y};\ 0\ for\ y = 0, and\ infinity\ for\ y = 1$$

Figure 3.1.1.3: The probability of the target variable occur

ii. Put the logarithm of the equation as Figure 2.1.1.4. The equation represents a linear relationship between the log-odds of the target variable and the input features.

$$log[\frac{y}{1-y}] = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + ... + b_n x_n$$

Figure 3.1.1.4: Final equation for Logistic regression (Logit Function)

Besides, there is a sigmoid function on linear regression as shown in Figure 2.1.1.5 and Figure 2.1.1.6.

$$p(x) = \frac{1}{1 + e^{-y}}$$

Figure 3.1.1.5: Sigmoid function

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + ... + \beta_n x_n)}}$$

Figure 3.1.1.6: Sigmoid function on linear regression

In Logistic Regression, the dependant variable follows the Bernoulli Distribution.

3.1.2 K-Nearest Neighbor (KNN)

According to Yeh and Lien (2009), a K-Nearest Neighbour (KNN) classifier finds the relationship between the current and previous data, then classifies the new data with the most similar historical data. Besides, KNN is a non-parametric algorithm, which means it makes no assumptions about the data's distribution and is applicable to both binary and multi-class classification issues. For example, the question asks how to tell the difference between a dog and a cat. The dataset includes all the characteristics of the dog and cat. Now, given some characteristics, the question asks whether those characteristics are considered cat or dog. The KNN model will look for similar traits in the new data and place them in either the cat or dog category, as shown in Figure 2.1.2.1.

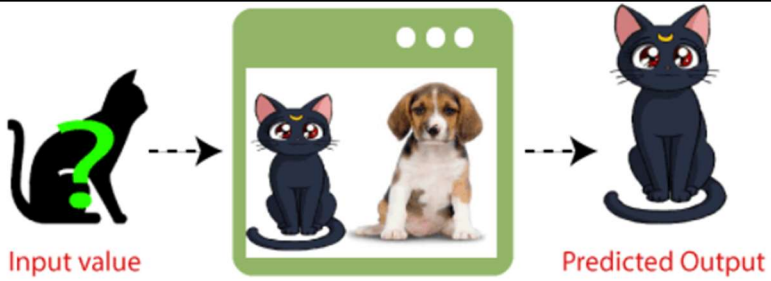


Figure 3.1.2.1: K-Nearest Neighbor (KNN) model

The primary benefit of this strategy is that no predictive model is necessary prior to categorization (Yeh and Lien, 2009).

The class of an unknown sample in the feature space is predicted using KNN based on the target variable of its k-nearest neighbours, k-value. The value of k is a hyperparameter that must be optimised for maximum performance of the KNN. It is a hyperparameter that the user must determine based on the dataset and the particular problem at hand. A good choice of k can have a vital influence on the KNN algorithm's performance. The KNN classifier extends this idea by taking the k closest points and assigning the majority sign to them. The small and unusual numbers to break ties, such as 1, 3, or 5 are the common k (Islam et al., 2010). Reduced effects of noisy points in the training data set are made possible by larger k values. K is typically determined using cross-validation. (Islam et al., 2010). A straightforward technique to calculate the value of k is to take the square root of the number of samples in the dataset.

Overall, the value of k should be chosen to strike a balance between overfitting (small k) and underfitting (large k) the data. Before selecting the ideal value of k, it is advised to test out a few different values of k and evaluate their effectiveness.

The distance between the samples is usually measured using Euclidean distance, as shown in Figure 2.1.2.2, but other distance functions can be used as well such as Manhattan distance, Chebyshev distance, and Minkowski distance. The calculated distance is the distance between all of the training and test points.

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

Figure 3.1.2.2: Euclidean distance

For example, if k=3, as illustrated in Figure 2.1.2.3, the algorithms consider the categorization region for the blue star to have three red circle neighbours.

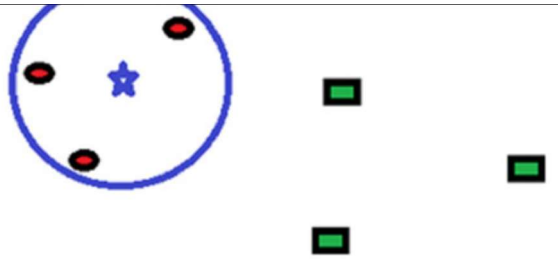


Figure 3.1.2.3: K=3

As a result, because it is closer to the red circle and the red circle is the majority average of the K points, the star is expected to be a red circle.

3.1.3 Gaussian Naïve Bayesian (GNB)

The Naive Bayesian (NB) classifier is a Bayesian algorithm that believes that a particular attribute value's impact on a target variable is independent and class conditional independence is the premise being used here (Yeh and Lien, 2009). According to Figure 2.1.3.1, the premise of class conditional independence greatly influences the algorithm's predictive accuracy (Yeh and Lien, 2009).

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Figure 3.1.3.1: Conditional probability: Bayes' Theorem

Despite the computational simplification provided by this assumption, dependencies between variables can exist in reality.

Based on Bayes' theorem, Gaussian Naive Bayes (GNB) is an algorithm that assumes the independence of features in a dataset given the class variable (Ontivero-Ortega et al., 2017). "Gaussian" is referred to as normal. The GNB method predicts the most likely class for a given set of input qualities using probability. Based on the input attributes, the probability of each class is computed and the class with the highest probability is the predicted class. Based on the input attributes, the probability of each class is computed and the class with the highest probability be the predicted class (Ontivero-Ortega et al., 2017). This feature independence assumption simplifies the probability calculation, resulting in a computationally efficient algorithm (Ontivero-Ortega et al., 2017).

For example, as shown in Figure 2.1.3.2, the colour with the highest probability is blue, which is Setosa. Hence, the predicted target variable is Setosa.

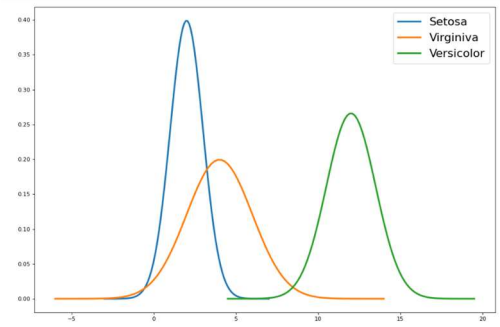


Figure 3.1.3.2: Example of Normal Probability

3.1.4 Decision Tree

Decision Tree work by segmenting the dataset into smaller subsets based on specific features, resulting in a tree-like structure that can be utilised to anticipate additional data points (Safavian and Landgrebe, 1991). The algorithm begins at the root node (Safavian and Landgrebe, 1991). The feature that divides the training data in the most effective way is the tree's root node (Bhavsar and Ganatra, 2012). There are numerous metrics, including

information gain, gain ratio, and Gini index are used for determining which feature divides the best training data (Bhavsar and Ganatra, 2012). Non-backtracking, greedy, top-down, and recursive divide and conquer strategies are used in the basic decision tree induction process (Bhavsar and Ganatra, 2012). Recursively repeating the algorithm for each subset results in the final leaf nodes having the predicted output values (Safavian and Landgrebe, 1991). This process stops when a stopping criteria is reached, such as reaching a maximum depth. Once the tree is built, it can be used to make predictions based on new data. To predict an outcome, the algorithm traverses the tree based on the input features, following the decision rules at each node. The prediction is made based on the target labels associated with the leaf node reached.

Decision Tree is used to develop a training model that can predict the target variable by learning the simple decision rules. Before discussing the operation of the Decision Tree, some terminology must be understood.

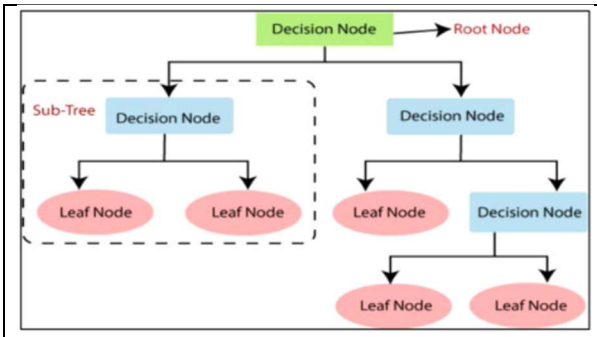


Figure 3.1.4.1: Schematic diagram of a decision tree (Bansal et al., 2022)

According to Bansal et al., as shown in Figure 2.1.4.1, the root node is the first branch of the decision tree, from which the full data set is further divided into multiple homogenous potential sets. The leaf node is the final outcome node; beyond it, no more tree separation is conceivable. Splitting is the process of subdividing the main node based on the limitations specified. A subtree or branch is formed when a hierarchy is divided. Irrelevant branches of the decision tree must be deleted to produce the best results. The parent node is sometimes referred to as the base node, while the remaining nodes are referred to as child nodes. Decision trees are used to categorise data by travelling down the tree from the root to a leaf node, with the leaf node supplying the category. Each node in the tree represents a test case for a certain attribute, and each edge

descending from the node represents one of the alternative solutions. This process is repeated for each subtree rooted at the new node.

There are some factors that influence which tree algorithm is utilised. When the response variable contains two categories, a standard classification tree is utilised. (Gulati et al., 2016). C4.5 is used when the response variable has numerous categories. (Gulati et al., 2016). It is also utilised when the response variable is continuous and the predictors and response variable have a linear relationship. (Gulati et al., 2016). The following is the decision tree's working step. Create a root node based on the metrics, which include information gain, gini index, and gain ratio. (Bhavsar and Ganatra, 2012). If all the samples belong to the same class, let this class be denoted as C, then the node (N) acts as a leaf node for the class labelled C. (Bhavsar and Ganatra, 2012). If no attribute is specified, N is used as the leaf node with the most prevalent class in the samples. Using the metrics once more to choose the best feature, which is the label node and specifically the test feature (Bhavsar and Ganatra, 2012). Each test feature has a value (vi), and for each vi, the samples are partitioned and a subtree is grown. Let ai be the set of tuples for which vi is the test feature. If vi is empty, attaching a leaf node with the is most often in samples.

3.1.5 Extreme Gradient Boosting (XGboost)

The Extreme Gradient Boosting (XGBoost) classifier is a widely used and powerful machine learning model for classification and regression tasks (Yu et al., 2019). XGBoost stands for "Extreme Gradient Boosting", as the name implies, and is based on the gradient boosting algorithm, which allows for the efficient creation of a series of decision trees. As a result, it is highly scalable and appropriate for large datasets and complex problems (Yu et al., 2019). During the training phase, the algorithm adds new trees to the model and focuses on previously misclassified instances, allowing for error correction. Regularisation techniques such as L1 and L2 aid in preventing overfitting.

The authors of the paper "An Overview of XGBoost Machine Learning Model and Its Applications" provide a thorough introduction to XGBoost. Because of its high efficiency and scalability, this model is well suited for analysing large and complex datasets (Yu et al., 2019). XGBoost's key features, such as its tree-based structure and regularisation techniques, enable it to handle

missing data and support parallel processing (Yu et al., 2019). Furthermore, XGBoost's diverse applications include image recognition, fraud detection, and financial forecasting (Yu et al., 2019).

XGBoost is preferable to gradient boosting algorithms because it provides a fair balance of bias and variation. On the other hand, gradient boosting is only optimised for variance and hence tends to overfit training data, whereas XGBoost provides regularisation terms that can improve model generalisation. Gradient boosting is the process of merging the predictions of numerous weak decision trees to produce a more accurate overall forecast. While XGBoost uses a gradient descent algorithm to minimise a loss function and construct a tree ensemble. XGBoost has several advanced features like regularisation, missing data handling, and parallel processing.

The gradient boosting algorithm functions as follows. Gradient boosting works by constructing a tree depending on the previous tree's mistake or residual. (Friedman, 2001). The tree is then gradient-boosted using the learning rate, and the prediction from the new tree is added to the prediction from the previous tree (Friedman, 2001). To begin, average the target variable as the initial starting point as prediction. Then, using the actual target variable minus the prediction, compute the list of errors (the average of the target variable). Creating a tree based on the residuals and replacing the leaves with the residuals' average Combining the prior forecast (the target variable's average) with the new tree to get a new prediction. The model forecast is exactly the same as the actual result, indicating that the model is overfitting the training data. To reduce overfitting, the learning rate is multiplied by the new prediction before combining it with the prediction from the previous stage (Friedman, 2001). Scaling is achieved by altering the newly added prediction from the new tree (Friedman, 2001). Adding a tree and scaling it with the learning rate helps to get closer to but not exactly to the goal variable. As a result, the model produces superior predictions on testing data with low variance. As is generally known, the lower the variance, the lower the model's bias and thus the better the model's prediction. Repeat the phase by calculating a new list of errors and building another tree, then combining the prediction with the previous stage. A new prediction is created by combining all the scaled predictions from all trees.

Extreme Gradient Boosting (XGBoost) adds a lot of optimisation and strategy from the Gradient Boosting technique. The most crucial component of XGBoost's success is its scalability in all scenarios (Chen and Guestrin, 2016). XGBoost is attempting to develop an algorithm that uses Gradient Boosting while also performing 10 times quicker than previous methods. (Chen and Guestrin, 2016). Furthermore, XGBoost scales to a large number of examples in distributed systems because it includes a lot of algorithmic optimisation and mathematics (Chen and Guestrin, 2016). The XGBooost results in faster learning and faster model exploration. This is due to the fact that XGBoost was created using parallel and distributed computing. It is also crucial that XGBoost explores out-of-score computation and allows data scientists to handle hundreds of millions of data points using only a simple computer (Chen and Guestrin, 2016).

The XGBoost algorithm forecasts the eventual outcome by combining the weights of the leaves from all trees (Tao et al., 2021).

$$\widehat{y_i} = \sum_{k=1}^{K} f_k(x_i), f_k \in \{f(x) = w_q(x)\}(q : \mathbb{R}^m \to T, w \in \mathbb{R}^T)$$

k = number of training's additive functions

T = number of leaves in a tree

q = tree structure

w = weight

Figure 3.1.5.1: Formula for predicting the output for XGBoost

The additive function may be used to optimise the trees by adding one at a time while optimising the goal (Tao et al., 2021). The XGBoost algorithm is wise owing to its exceptional regularisation capabilities; it avoids overfitting by combining L1 and L2 regularisation methods (Tao et al., 2021). The formula for XGBoost's regularisation ability is stated in the figure below (Tao et al., 2021). It is used to reduce the regularisation ability in each loop (Tao et al., 2021).

$$\mathscr{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$$

l = loss function

$\Omega(f_k)$ = function that penalizes the model's complexity

Figure 3.1.5.2: Formula for Regularization Ability of XGBoost

The training set will be introduced until the regularisation ability function shows no advancement (Tao et al., 2021). XGBoost is an excellent method since it focuses on fixing errors and updating weights to improve the final prediction (Tao et al., 2021).

3.2 Weight of Evidence (WoE) and Information Value(IV)

Feature binning is the process of grouping all the attributes in order to improve the prediction performance of the attriutes (Yap et al., 2011). Weight of Evidence (WoE) and Information Value (IV) are always used in banking and finance to group and select classes. The WoE of a variable is defined as the log of the ratio of 'events' in the dataset divided by the ratio of 'non-events' in the dataset, as shown in Illustration 2.2.1. (Yap et al., 2011). In this case, since this target variable is the default payment for the following month, using a 1 to represent default and a 0 to represent not default. As a result, high negative values for WoE here reflect the user's low default for the next month, whereas high positive values correspond to the user's high default for the next month.

$$woE = \ln\left(\frac{Event\ \%}{Non\ Event\ \%}\right)$$

Illustration 3.2.1: Formula for WoE

Following the binning of the feature with WoE, data transforming should be used as the "weightage" of the binning. To standardise the range of variables, feature scaling is usually required. This is because the computer will not understand the value's meaning. It only knows that 0 is less than 100. For instance, age in the dataset ranges from 21 to 75, whereas marital status only has 1, 2, and 3. The computer will have no idea what that is and will simply give more points for age because it has a higher value.

Normalisation is a technique used to standardise the range of values for different features or variables. It is also known as data normalisation or feature scaling. It is often used in data preprocessing to ensure that the features are on

the same scale, which is advantageous for some machine learning techniques. However, it is important to know that there are cases in business models where normalisation may not be suitable or necessary. This is due to the fact that it is critical for maintaining the interpretability of the data and model in various business scenarios. Normalisation alters the scale and distribution of the variables, making it more difficult for stakeholders to understand and interpret the results. Hence, in business data, there are two matrices to replace normalisation, which are WoE and IV.

In other words, the Weight of Evidence (WoE) method is a statistical technique used in credit risk modelling to assess the predictive power of various variables (Van Gool et al., 2013). It calculates the power of the association between a particular variable and a desired outcome, such as the probability of default in the context of credit risk (Van Gool et al., 2013). Continuous or categorical variables are transformed into discrete values that represent the likelihood of default in order to be used in credit scoring models (Van Gool et al., 2013). The WoE values are derived from the log-odds ratio of the proportion of good and bad borrowers within each interval of the variable under consideration as shown in Illustration 2.2.1. Credit scoring models that are more accurate and efficient in predicting credit risk can be developed by identifying important variables and reducing dimensionality (Van Gool et al., 2013). Thus, a WoE is a must in building credit scoring models.

To determine the predictive power of variables, the information value (IV) is introduced, which is an indicator widely used in credit risk modelling (Řezáč, 2011). IV determines the strength of the association between the binary target variable and the independent variable by calculating the difference between the percentage of positive events (targets) in a particular category of the independent variable and the percentage of negative events in the same category, using a logarithmic function as shown in Illustration 2.2.2. Variables with high IV values are considered strong predictors and are more likely to be included in the prediction model. Conversely, variables with low IV values are considered weak predictors. The use of IV is required to determine the essential variables in a credit scoring model, which can then be used to assess the variables' predictive potential (Wood and Piesse, 1988). IV considers variable correlation

and variable interaction with the target variable, enhancing the accuracy and reliability of the credit scoring model (Řezáč, 2011).

$$\text{Information Value} = \sum(p_{\text{goodattribute}} - p_{\text{badattribute}}) * \text{WeightofEvidence}.$$

Illustration 3.2.2: Formula for IV

The calculation of IV involves dividing a variable into bins and computing the Weight of Evidence (WoE) for each bin (Barthès et al., 2020). The IV considers both the strength of the association between the variable and the class as well as the variable's distribution in the population (Barthès et al., 2020). The IV is then calculated by adding the proportions of good and bad borrowers for each interval, weighted by the WoE for that interval. IV is typically used in conjunction with other methods to ensure robust and accurate model performance (Wood and Piesse, 1988). However, caution must be exercised to avoid overfitting or misinterpreting the study results, as the model's effectiveness may vary based on the specific data used and the methodologies employed.

Before transforming the data, Information value (IV) has to be measured so that it can be determined whether to keep or drop it. The table 2.1.5.1 descriping the meaning of the range of IV by Yap et al., 2011.

Table 3.1.5.1: The meaning of the range of IV

IV value	Description
>= 0.5	Over predicting variable
< 0.5	Strong variable
< 0.3	Medium variable
< 0.1	Weak variable
< 0.02	Unimportant variable

In summary, WoE and IV take on the roles of feature transformation and feature selection (Zdravevski et al., 2011). In WoE, all attributes can be discussed as having the same "weightage," which is helpful for classification algorithms (Zdravevski et al., 2011).

As a consequence, using WoE to bin and transform the feature are necessary steps. The rule that WoE must follow is that the trend must be linear, either decreasing or increasing. The default bin for WoE is ten bins; if the bins do not

follow a linear pattern, the bin must be reduced by one, and the minimum bin is three.

3.3 Statistical Analysis

The course entitled "Applied Regression Analysis" is available at UTAR. This course introduces the fundamental concepts of regression analysis and examines the estimation of regression model parameters. The investigation is widened to include diagnostics for leverage and influential observations, as well as the selection of independent variables. Subtitle 2.3.1 is used to identify the potential outlier; Subtitle 2.3.2 is used to identify the influential data; and Subtitle 2.3.3 is used to ensure that the features that were picked are relevant to the target variable. The purpose of subtitle 2.3.4 is to double-check if the feature chosen has proven successful in predicting the target variable.

3.3.1 Diagnostic for Leverage and Influence (Method of Scaling)

A. Standardized Residuals / Semistudentized Residuals, d_i

Standardized Residuals, $d_i = \frac{Residual - Mean\ Residual}{Standard\ Deviation\ of\ Residuals}$

Figure 3.3.1.1: Formula for Standardized Residuals

$Residual_{(i)}$ is the difference between the observed value of the features for the i-th observation and the predicted value based on the linear model. By assessing the deviation of each data point from the general trend, standardised residuals contribute to determine the reliability of a linear regression model. All data tends to reside 3 standard deviations from its mean; a standardised residual outside of this range indicates a suspected outlier observation.

B. Studentized Residuals, r_i

Studentized Residuals, $ri = \frac{ei}{\sqrt{MSE\ (1-hii)}}$

where

e_i = i-th residual

MSE = mean squared error

h_{ii} = leverage of the i-th data point

Figure 3.3.1.2: Formula for Studentized Residuals

The critical value for studentized residuals is $|ri| \geq t(1-\frac{\alpha}{2n}; n-p)$, where n is the number of observations and the p is the total number of features and intercepts. Outliers are data points that exceed critical levels.

C. Predicted Residual Sum of Squares, PRESS Residuals, $e_{(i)}$

The PRESS statistic measures how effectively the model predicts additional data points that were not included during the training phase of the model.

$$PRESS = \frac{ei}{1 - Hii}$$

Figure 3.3.1.3: Formula for PRESS

H_{ii} is the hat matrix, which is crucial in estimating the amount of a student's deleted residual. It is useful for spotting outliers in Y and X observations. The deleted residual will be bigger than the ordinary residual if Hii is large. So, the lower the PRESS value, the better the model's prediction performances and there will be low influence points.

D. R-Student / Studentized Deleted Residuals, t_i

The critical value for studentized residuals is $|ti| \geq t(1-\frac{\alpha}{2n}; n-p-1)$

R-Student, $t_i = \frac{ei}{\sqrt{MSE(i)\ (1-hii)}}$

Figure 3.3.1.4: Formula for R-Student

where

ei = the orifinal residual

MSEi = mean squared error of the model

Hii = diagonal of hat matrix

Outliers can be detected using t_i residual. Outliers are data that is above the critical value because their residuals vary dramatically after they are removed from the model.

E. Hat matrix, H (leverage values)

The hat matrix is important in understanding the idea of leverage since it measures the impact of each data point on the projected regression

coefficients and predictions. A linear regression model with n observations and p predictors (including the intercept) is used.

$H = X(X^TX)^{-1}X^T$

Figure 3.3.1.5: Formula for Hat matrix

X = n x p matrix

X^T = transpose of X

(X^TX) = cross-product mateix of X with its transpose

$(X^TX)^{-1}$ = inverse of the cross-product matrix

According to Li and Valliant (2009), The hat matrix have several important properties :

1. It is symmetric: $H^T = H$
2. It is idempotent: $H^2 = H$
3. It determines the predicted values: The predicted values $\hat{y} = Hy$ where y is the target variable / response variable
4. It determines the fitted values: The fitted values $\hat{y}$ are equal to the projection of y onto the column space of X.
5. It determines the residuals: The residuals can be calculated as e = y - $\hat{y}$, where e is the vector of residuals.

The hat matrix's diagonal elements, indicated as Hii, represent the leverage of each data point. The amount of effect an individual data point has on the computed regression coefficients is measured by leverage. Observations with high leverage values have extreme predictor values in comparison to the other data points and can have a large influence on the regression model.

In regression diagnostics, the hat matrix is used to identify influential observations and potential outliers. Figure 2.3.1.6 shows the Rule of Thumb of the Hat Matrix.

$H_{ii} > \frac{2p}{n}$ high leverage

$H_{ii} > 0.5$ high leverage

$0.2 < H_{ii} <= 0.5$ moderate leverage

Figure 3.3.1.6: Rule of Thumb

The H_{ii} with the high leverage must be identified for additional investigation since they have the potential to dramatically affect the regression model's outcomes.

3.3.2 Indentifying Influential Cases

A. Influence on Single Fitted Value, DFFITS

$(DFFITS)_i = \frac{\widehat{yi} - \widehat{y(i)}}{\sqrt{MSE(i)hii}}$ or $(DFFITS)_i = ti\left(\frac{hii}{1-hii}\right)^{1/2}$

$\hat{y}$ = i-th case when all the cases are used to fit the regression function

$\widehat{y(i)}$ = i-th case ontained when it is excluded from the fitting regression function

Figure 3.3.2.1: Formula for DFFITS

The critical value for DFFITS > $2\sqrt{\frac{(k+1)}{n}}$ for large data sets with more than 30 observations. When the number of observations is fewer than 30, the dataset is considered small, and the crucial value is DFFITS > 1. Any data point larger than the critical values is considered an influential case and must be observed for future investigation.

B. Influence on all Fitted Values, Cook's Distance

Cook's distance, abbreviated Di, is a statistical metric used in regression analysis to evaluate the impact of individual data points, i-th case, on the overall regression model. It measures the change in regression coefficients when certain information is removed from the model. Cook's distance is useful for regression model diagnostics and validation. It is one of the techniques used to assess the quality and reliability of regression models and make educated decisions regarding data inclusion and exclusion.

$D_i = \frac{\sum_{j=1}^{n}(\hat{y}j - \widehat{yJ(i)})^2}{(k+1)\,MSE}$ or $D_i = \frac{ei^2}{(k+1)MSE}\left[\frac{hii}{(1-hii)^2}\right]$

Figure 3.3.2.2: Formula for Cook's Distance

When training the regression model, each of the n fitted values, $\hat{y}$ is compared with the corresponding fitted value $\widehat{yJ(i)}$ for Cook's Distance. Based on Figre 2.3.2.2, it is evident that any instance of the residual and

leverage value, whether high or small, can influence the i-th case. Data points with Cook's distances greater than one are frequently pointed out as possibly important. Besides, the D_i is less than F(0.2, k+1, n-k-1), indicating that the impact of the i-th case on all fitted values is negligible. For those D_i that have a value greater than F (0.5, k+1, n-k-1) indicating that the i-th case has a major influence on the all fitted values. The value k is the number of variables and n is defined as the total number of observations.

C. Influence on the Regression Coeficients, DFBETAS

$(DFBETAS)_{j(i)} = \frac{\widehat{\beta J} - \widehat{\beta J(i)}}{\sqrt{MSE(i)\ cjj}}$

Figure 3.3.2.3: Formula for DFBETA

$\widehat{\beta J}$ = A measure of the i-th case's effect on each estimated regression coeeficient, where j = 0,1,2,…, k

c_{jj} = j-th diagonal element of $(X'X)^{-1}$

X = the n*(k+1) matrix

n = number of observation

k = the number of independent variables.

$\widehat{\beta J}$ is based on all n cases and the $\widehat{\beta J(i)}$ is the regression coefficient calculated by omitting the i-th case. Analysts can learn about the sensitivity of the regression coefficients to specific data points by studying the DFBETAS values. This information is vital for understanding which data points are critical for model estimate and how the model's outputs may change when certain observations are included or excluded. The value of DFBETAS that greater than 1 for small dataset and the value of DFBETAS that greater than $\frac{2}{\sqrt{n}}$ for large dataset are consider as potentially influential for that specific predictor variable.

3.3.3 Multiple Linear Regression

A regression model employs a linear model to determine whether the response variable is related to the numerical values of one or more variables that are quantitative (Freund et al., 2006). Multiple linear regression is a regression with multiple regressor variables. Illustration 2.3.1 represents the equation for the

multiple linear regression model (Freund et al., 2006). The test hypotheses are as follows:

H_0 = Determining the target variables using features as predictive variables are not useful.

H_1 = Determining the target variables using features as predictive variables are useful.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m + \epsilon,$$

where

y is the dependent variable
$x_j, j = 1, 2, \ldots, m$, represent m different independent variables
β_0 is the intercept (value when all the independent variables are 0)
$\beta_j, j = 1, 2, \ldots, m$, represent the corresponding m regression coefficients
ϵ is the random error, usually assumed to be normally distributed with mean zero and variance σ^2

Figure 3.3.3.1: Multiple Regression model equation

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1m} \\ 1 & x_{21} & x_{22} & \cdots & x_{2m} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix}, \; Y = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_n \end{bmatrix}, \; E = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \cdot \\ \cdot \\ \epsilon_n \end{bmatrix}, \text{ and } B = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \cdot \\ \cdot \\ \beta_m \end{bmatrix},$$

where x_{ij} represents the ith observation of the jth independent variable, $i = 1, \ldots, n$, and $j = 1, \ldots, m$.

Figure 3.3.3.2: Matrices to determining the model equation

The model equation for all observations is based on the matrices in Figure 2.3.3.2 (Freund et al., 2006). For the final validation of the linear regression model, use the Multiple Regression Model to determine whether the remaining attributes in the dataset are all useful for determining the target variable. The alpha level (probability) is usually set at 5% (Alexopoulos, 2010). Those variables whose p-values exceed the alpha level provide sufficient evidence that the variable is useful in determining the target variable.

3.3.4 R Square and Adjusted R Square

The coefficient of determination, R-Squared, R^2 and Adjusted R-Squared, R^2_{Adj} represent a model's degree of fit (A.Y.J & R, 2013).

$$R^2 = \frac{SS_R}{SS_T} = 1 - \frac{SS_E}{SS_T}$$

where

SS_R = sum of squares regression

SS_T = sum of squares total

SS_E = sum of squares error

Figure 3.3.4.1: Formula for R^2

The R^2 number is usually considered the amount of variance in the target variable that can be explained by utilising the independent factors to predict the target variables. However, R^2 behaves in such a way that if more variables are added to the model, R^2 will always grow, even if the added new variable performs poorly in estimating the target variable. As a result, the Adjusted R-Squared is introduced.

$$R^2_{Adj} = 1 - \frac{\frac{SS_E}{n-k-1}}{\frac{SS_T}{n-1}} = 1 - \frac{n-1}{n-k-1}(1-R^2)$$

Where

SS_E = sum of square error

SS_T = sum of square total

n = number of observation

k =number of independent variable

Figure 3.3.4.2: Formula for Adjusted R-Squared

The R^2_{Adj} is preferred because, no matter how many variables are in the model, the R^2_{Adj} will only rise when a variable decreases the residual mean square, $\frac{SS_E}{n-k-1}$ when the variable is added to the model.

CHAPTER 4

METHODOLOGY AND WORK PLAN

As stated in the subtitle 1.5 scope study, Project I is primarily concerned with model development. Model development will indicate which data mining model is optimal for binary classification problems. Aside from that, in Project II, use the WoE and IV determined here to create a scorecard to determine whether the credit card holder can be trusted.

In the section 3.1, the dataset is introduced. Following data collection, data preparation must be carried out. Data cleaning is one of the data preprocessing steps. Hence, data cleaning is completed in subtitle 3.2. In subtitle 3.3, focusing on the data preprocessing of feature selection, feature binning and feature transformation using WoE and IV. Then, using statistical analysis to make sure that all the features used to determine the target variable are useful in section 3.4. After confirmation of all the features, using the hold-out method, divide the dataset into a 70% training set and a 30% testing set in subtitle 3.5.

4.1 Introduction of Default of Credit Card Clients Dataset

The link (https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients) provides the raw variable name and description. There are a total of 25 attributes and the raw dataset has 30000 rows.

The first step is to conduct exploratory data analysis, which is to review all the data in the dataset. All the data in the dataset should be complete and understandable. In this dataset, for example, the data for the Marriage column does not have a description for the data object 0, so the value cannot be determined. Hence, any data that does not provide the description from the link must be removed from the dataset since there is no predictive function for the model.

Illustration 3.1.1 shows that there are 26893 rows and 25 columns after manually inspecting the data and deleting those that do not provide information. In the 26893 clients, there are 20651 clients (rows) that will not default for the next month and 6242 clients (rows) that will default for the next month.

```
(26893, 25)
```

Illustration 4.1.1: Rows and Columns in raw dataset from Python

4.2 Data Preprocessing (Data Cleaning)

Data preprocessing is one of the steps in preparing and transforming data so that it can be fit into the data mining algorithm and produce accurate and useful results (Garcia et al., 2016). Data preprocessing encompasses data cleaning, reduction, transformation, and integration (Garcia et al., 2016). There is a lot of potentially incorrect data in real-world datasets, so data cleaning is required. Incomplete data, noisy data, inconsistent data, and intentional data are all examples of incorrect data. These incorrect data are typically caused by faulty instruments, human or computer error, and transmission error.

If missing values are not handled appropriately, the algorithm will produce incorrect results. In most cases, incomplete data with missing target variables is dropped. The missing value for the other attribute can be observed to see if there is a formula to get that column. If the missing column is deemed significant, the categorical missing data can be filled with "unknown" or "NAN" (Garcia et al., 2016). While numerical missing data can typically be filled using the mean or median (Garcia et al., 2016). Redundant attributes are those that are duplicated. For example, monthly and annual earnings. As a result, two of them will produce the same result, and using one of them should suffice.

Exploratory Data Analysis (EDA) is a type of data analysis in which a researcher examines the data to discover patterns, trends, and relationships in large amounts of data using automated or semi-automatic methods (Martinez et al., 2021). According to Martinez et al. (2021), John W. Tukey defined the step as EDA methodology. First, a problem definition is needed to figure out what to do (Myatt, 2007). This is the step in which a focused plan for execution is developed (Myatt, 2007). For example, the first question should always be, What is the title that you want to give? What is the purpose of doing this title, and what is the objective? Then, the question and design can be asked to iterate. Third, the datasets that correspond to the title of this object can be collected. Transform data will be used to collect an appropriate form for analysis (Myatt, 2007). As a result, conducting a statistical investigation of the information and

developing a model to obtain the response The only way to discover patterns, trends, and data structure is through scientific and statistical visualisation, which is central to EDA (Martinez et al., 2021). EDA is typically used to detect the number of zeros. If there are zero values, the researcher must examine the column and determine why the column data contains so many zeros and hence decide whether it should be dropped.

Illustration 3.1.1 shows that there are 26893 rows and 25 columns after manually inspecting the data and deleting those that do not provide information. Summarise the data, which is the process of reducing data for interpretation while retaining critical information (Myatt, 2007). Illustration 3.2.1 presents the full raw column names from the dataset.

```
# print column names of dataset
file.columns
```

```
Index(['ID', 'AMOUNT_OF_GIVEN_CREDIT', 'SEX', 'EDUCATION', 'MARITAL_STATUS',
       'AGE', 'REPAYMENT_STATUS_IN_SEPTEMBER', 'PAY_2', 'PAY_3', 'PAY_4',
       'PAY_5', 'PAY_6', 'AMOUNT_OF_BILL_STATEMENT_IN_SEPTEMBER', 'BILL_AMT
2',
       'BILL_AMT3', 'BILL_AMT4', 'BILL_AMT5', 'BILL_AMT6',
       'AMOUNT_OF_PREVIOUS_PAYMENT_IN_SEPTEMBER', 'PAY_AMT2', 'PAY_AMT3',
       'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6', 'default payment next month'],
      dtype='object')
```

Illustration 4.2.1: Complete raw columns name

The count, mean, minimum, and maximum value of data attributes are examples of description statistics as shown as Illustration 3.2.2.

	ID	AMOUNT_OF_GIVEN_CREDIT	SEX	EDUCATION	MARITAL_STATUS	AGE
count	26893.000000	26893.000000	26893.000000	26893.000000	26893.000000	26893.000000
mean	14821.985572	159093.581229	1.594876	1.833154	1.560592	35.312089
std	8635.518569	127198.611394	0.490925	0.705326	0.518601	9.256918
min	1.000000	10000.000000	1.000000	1.000000	1.000000	21.000000
25%	7362.000000	50000.000000	1.000000	1.000000	1.000000	28.000000
50%	14709.000000	130000.000000	2.000000	2.000000	2.000000	34.000000
75%	22197.000000	230000.000000	2.000000	2.000000	2.000000	41.000000
max	30000.000000	1000000.000000	2.000000	4.000000	3.000000	79.000000

8 rows × 25 columns

REPAYMENT_STATUS_IN_SEPTEMBER	PAY_2	PAY_3	PAY_4	...	BILL_AMT4	BILL_AMT5	BILL_AMT6
26893.000000	26893.000000	26893.000000	26893.000000	...	26893.000000	26893.000000	26893.000000
0.394415	0.049046	0.003049	-0.063213	...	46490.189008	43288.065891	41753.867624
0.790098	1.104099	1.117348	1.099277	...	65709.999931	62172.798571	60916.463009
0.000000	-2.000000	-2.000000	-2.000000	...	-170000.000000	-81334.000000	-339603.000000
0.000000	-1.000000	-1.000000	-1.000000	...	4041.000000	3040.000000	2173.000000
0.000000	0.000000	0.000000	0.000000	...	21289.000000	19715.000000	19227.000000
1.000000	0.000000	0.000000	0.000000	...	59696.000000	55063.000000	52624.000000
8.000000	8.000000	8.000000	8.000000	...	891586.000000	927171.000000	961664.000000

AMOUNT_OF_PREVIOUS_PAYMENT_IN_SEPTEMBER	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	default payment next month
26893.000000	2.689300e+04	26893.000000	26893.000000	26893.000000	26893.000000	26893.000000
5566.773547	5.657438e+03	5101.945488	4692.604135	4658.448555	5031.837095	0.232105
16056.347407	2.034086e+04	17420.001522	15260.059059	14682.315334	17211.480945	0.422183
0.000000	0.000000e+00	0.000000	0.000000	0.000000	0.000000	0.000000
1100.000000	1.000000e+03	504.000000	370.000000	349.000000	268.000000	0.000000
2200.000000	2.058000e+03	1901.000000	1529.000000	1600.000000	1500.000000	0.000000
5005.000000	5.000000e+03	4500.000000	4011.000000	4016.000000	4000.000000	0.000000
873552.000000	1.227082e+06	896040.000000	621000.000000	426529.000000	528666.000000	1.000000

Illustration 4.2.2: Description Statistics for all attributes

EDA includes correlation statistics, which are used to quantify data and can also be included in summarization (Myatt, 2007). What is the definition of the correlation coefficient? It is a statistic used to assess the power of a linear relationship between two variables (Ratner, 2009). Any value between -1 and +1 can theoretically be taken as the correlation coefficient. (Ratner, 2009). Due to the challenge of gathering all responses from all over the world, all data from the real world is usually a sample. As an outcome, the formulas used are typically formulas in sample data. The Pearson correlation coefficient is the name given to the linear correlation coefficient (Haomiao et al., 2016). It is used to assess the degree of linear correlation between two random variables. It is commonly used in statistics, such as data analysis for decision-making. Illustration 3.2.3 depicts the Pearson Sample Correlation Coefficient formula.

$$r_{xy} = \frac{\sum(x_i - \overline{x})\sum(y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2}\sqrt{\sum(y_i - \overline{y})^2}}$$

where $\overline{x} = \frac{1}{n}\sum_{i=1}^{N} x_i$ denotes the mean of x and $\overline{y} = \frac{1}{n}\sum_{i=1}^{N} y_i$ denotes the mean of y.

Illustration 4.2.3: Formula for Pearson Sample Correlation Coefficient

The value of the correlation coefficient ranges from 0 to 1 (0 to -1), where the positive indicates a positive linear relationship between two variables and the negative indicates a negative linear relationship between two variables. In addition, the strong linear relationship between two variables is from 0.7 to 1.0 (-0.7 to 1.0).

The scatter plot for all variables is shown in Illustration 3.2.4. A scatter plot can be used to determine the correlation coefficient. Illustration 3.2.4 shows that the Amount of Bill Statement 1 and Amount of Bill Statement 6 have a strong relationship and represent the same property but in different months. According to the information provided in the link, Amount of Bill Statement 1 to Amount of Bill Statement 6 indicate the Amount of Bill Statement from September 2005 to April 2005, respectively. The more recent Amount of Bill Statement will be used to forecast the default rate for the following month and others will be dropped. Since the Amount of Bill Statement is closely linked in Illustration 3.2.4, those attributes with the same attribute 1 to attribute 6 are also considered highly associated, and hence the most current attribute is chosen, which is the Amount of Bill Statement at September 2005.

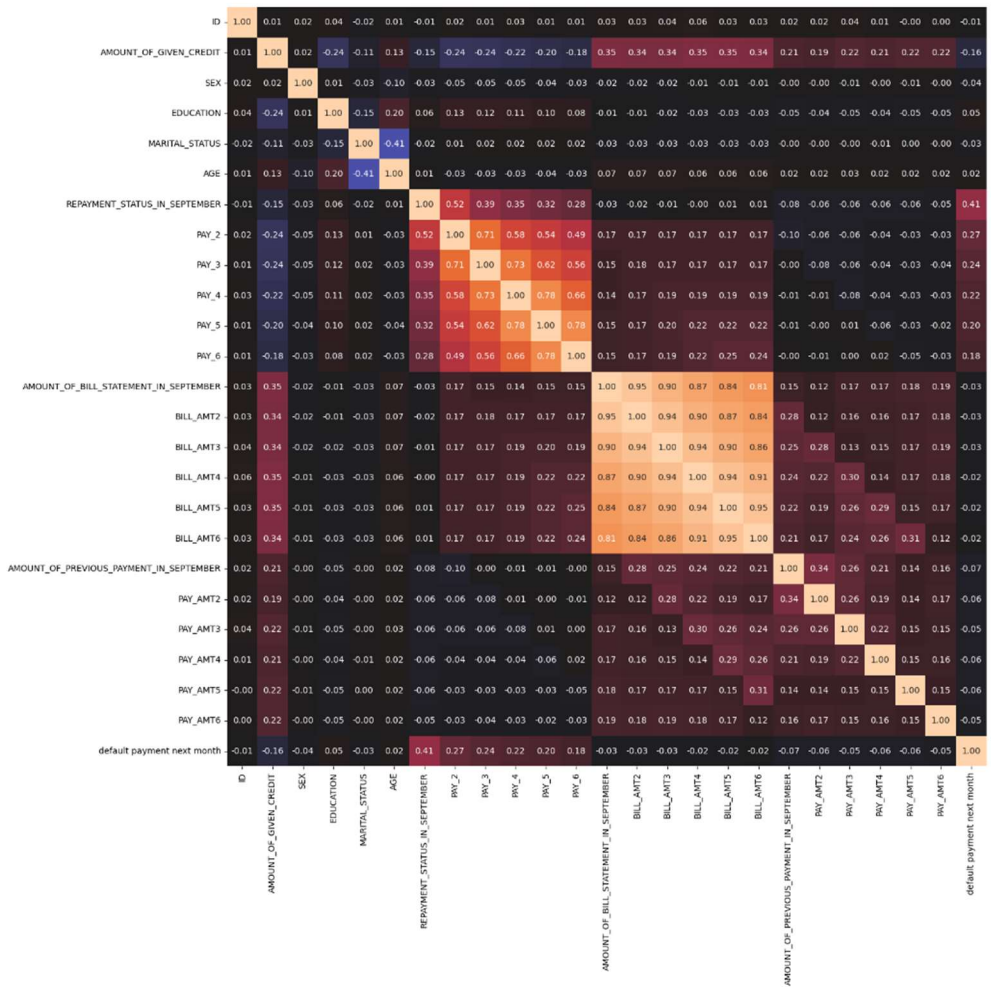


Illustration 4.2.4: Relationship between all the attribute and the target variable (default rate for next month)

After removing those attributes from the dataset that are replicated or do not contain any predictive variables, such as Amount of Bill Statement 1 to Amount of Bill Statement 6. There are still 26893 rows and 10 columns in the dataset. The columns remaining include ID, Amount of given credit, Sex, Education, Marital Status, Age, Repayment Status in September, Bill Amount, Pay Amount and default payment for next month. The description for the remaining attributes will still be the same since only the columns of data that are not necessary will be dropped, not the data object (row data). The number of zeros in attributes must be checked to ensure that there is data in the specific attribute. For example, in this dataset, some attributes have a number of zeros.

```
Count of zeros in column ID is:  0
Count of zeros in column AMOUNT_OF_GIVEN_CREDIT is:  0
Count of zeros in column SEX is:  0
Count of zeros in column EDUCATION is:  0
Count of zeros in column MARITAL_STATUS is:  0
Count of zeros in column AGE is:  0
Count of zeros in column REPAYMENT_STATUS_IN_SEPTEMBER is:  20132
Count of zeros in column AMOUNT_OF_BILL_STATEMENT_IN_SEPTEMBER is:  1301
Count of zeros in column AMOUNT_OF_PREVIOUS_PAYMENT_IN_SEPTEMBER is:  4300
Count of zeros in column default payment next month is:  20651
```

Illustration 4.2.5: Numbers of Zeros in dataset

From the result on Illustration 3.2.5, Repayment Status in September is a classifier; 0 indicates that it has been paid in full, which cannot be removed. Amount of Bill Statement in September and Amount of Previous Payement in September are numerical values that will be further examined. The output which is the target variable that is denoted by Default Payment for Next Month, cannot be removed.

Then, to avoid incorrect analysis, the missing values must be determined. From analysis, it shows that the missing values in this dataset do not have to be handled as there are none, as shown in Illustration 3.2.6.

```
ID                                    0
AMOUNT_OF_GIVEN_CREDIT                0
SEX                                   0
EDUCATION                             0
MARITAL_STATUS                        0
AGE                                   0
REPAYMENT_STATUS_IN_SEPTEMBER         0
AMOUNT_OF_BILL_STATEMENT_IN_SEPTEMBER 0
AMOUNT_OF_PREVIOUS_PAYMENT_IN_SEPTEMBER 0
default payment next month            0
dtype: int64
```

Illustration 4.2.6: Missing value in dataset

Then, for the remaining columns, double-check the relationship. The scatter plot in Illustration 3.2.7 is visualised. Since it has so many attributes, it is difficult to interpret.

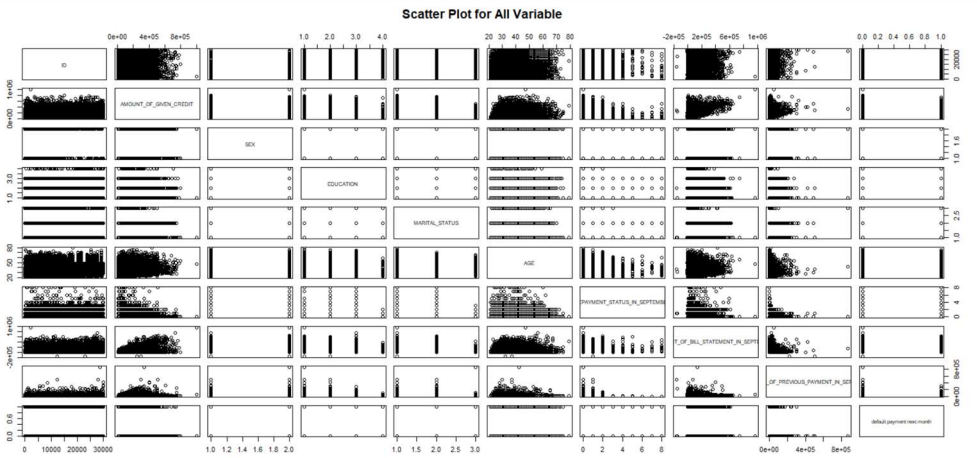


Illustration 4.2.7: Scatter plot for all the remaining variable

However, the correlation heatmap clearly demonstrated the correlation of each attribute to the target variable and other attributes, as shown in Illustration 3.2.8. This visualisation shows that there is no strong correlation via a linear rule. As a result, there are no columns that have to be dropped or inverted.

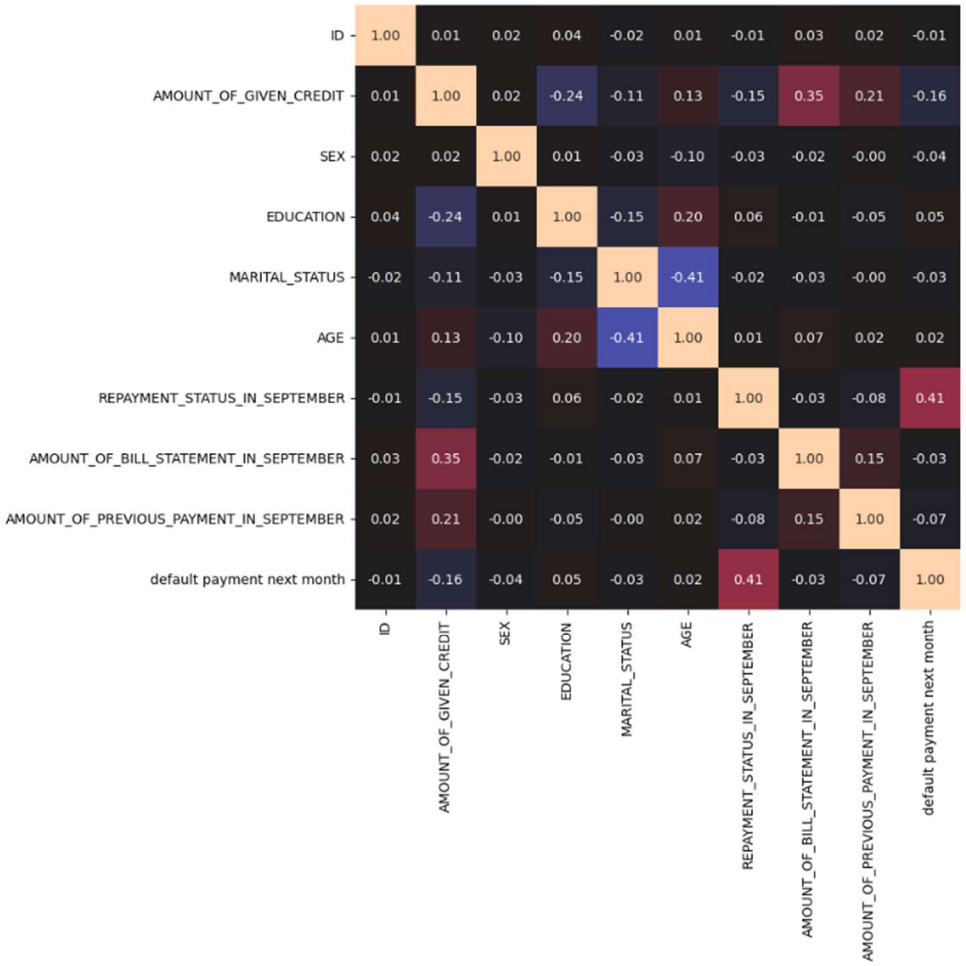


Illustration 4.2.8: Analysis of Correlation Heatmap between the remaining attributes and target variable

In view of the increasing amount of data collected day after day, data mining is required to extract useful data. Data visualisation is used in data analysis to provide information and insight (Andrienko et al., 2020). Data visualisation can be used to identify patterns in information and thus provide accurate decision-making. The fact that visual impressions have such a strong influence on humans is the reason for data visualisation's success (Egger, 2022). The main purpose of data visualisation is to show numerical and categorical data (Egger, 2022). Data can be divided into three types: univariate data, bivariate data, and multivariate data. Exactly one characteristic characterises univariate data, two characteristics characterise bivariate data, and several attributes characterise multivariate data. Data visualisation is concerned with exploring, analysing, and presenting data. The data visualisation is used to explore the data

and identify patterns. Data visualisation can be used to analyse and present data. Categorical data is commonly visualised using different colours, a bar chart, and a crosstab.

Visualisation for numerical data can include a bar chart and a box plot. Histograms are one type of visualisation that can be used to show the distribution of central tendency on two axes: x and y (Mowbray et al., 2019). The X-axis usually represents the data value, and the Y-axis usually represents the frequency (Mowbray et al., 2019). According to Mowbray et al. (2019), boxplot is one method used to identify outliers. Outlier cases are those where the value exceeds the whiskers (Mowbray et al., 2019). Finding outliers in numerical data using these two visualisations and handling them using the Range (IQR). The conclusion from the visualisation is that the outlier must be replaced or removed. The interquartile range (IQR) can be used to identify outlier cases (Mowbray et al., 2019). An outlier is defined as a case value that extends beyond either side of the IQR box's edge by more than 1.5 times, according to Mowbray et al. (2019). The interquartile range is used to handle the outlier for the remaining attributes. Once the outlier is identified, it will be tested by removing or replacing it with the 5th and 95th percentiles for the smaller and larger outliers, respectively.

The target variable's portion can be seen in Illustration 3.2.9 and Illustration 3.2.10. Both show that approximately 76.79% of users will not default in the next month, while approximately 23.21% will default in the following month.

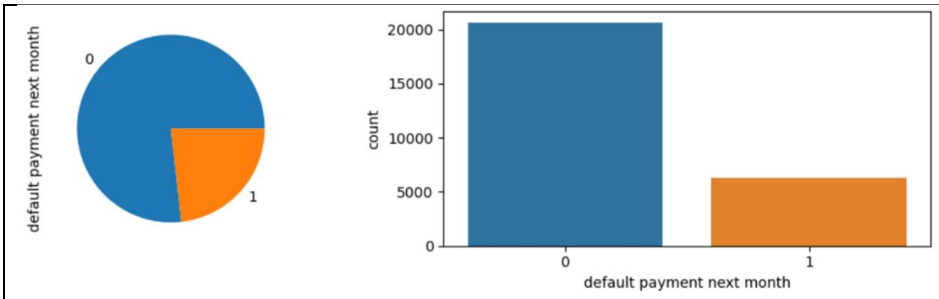


Illustration 4.2.9: Vizualization for Target Variable (Default Payment Next Month)

```
# no of count for target variable
file_want_column['default payment next month'].value_counts()
```

```
0    20651
1     6242
Name: default payment next month, dtype: int64
```

```
# probability of default for target variable
file_want_column['default payment next month'].value_counts(normalize=True)
```

```
0    0.767895
1    0.232105
Name: default payment next month, dtype: float64
```

Illustration 4.2.10: Portion of the target variable from the remaining dataset

Aside from that, given those category data, EDA may calculate the default rate for credit card users next month (target) and display it using a side-by-side bar chart and a crosstab. This visualisation clearly shows the relationship between the specific variable and the target variable (default rate). Out of the 10 variables mentioned, 4 of them are categorical data, and one of them is the target variable, namely default payment for the following month. Sex, Education, Marital status, and Repayment status in September are among the category data.

I. SEX

default payment next month	0	1	All	Default Rate
SEX				
1	8145	2750	10895	25.240936
2	12506	3492	15998	21.827728
All	20651	6242	26893	23.210501

Illustration 4.2.11: Crosstab for 'SEX'

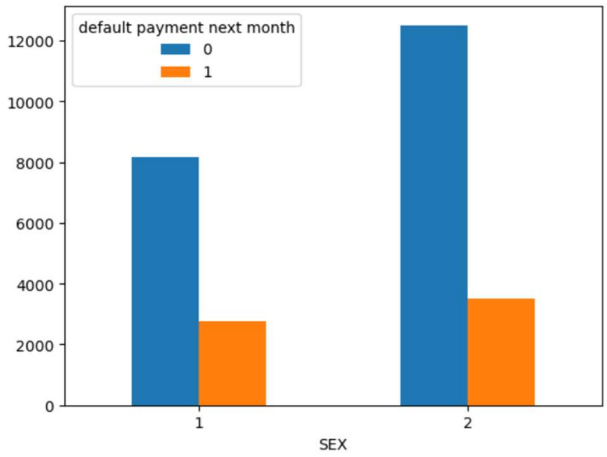


Illustration 4.2.12: Side-by-side bar chart for 'SEX'

Illustrations 3.2.11 and 3.2.12 show that for the 'SEX' variable, 1 represents male and 2 represents female. Male default rates are around 25%, and female default rates are around 22%, which is roughly the same. As a result, in the scorecard, it will assign the same score to males and females for 'SEX' feature.

II. EDUCATION

default payment next month	0	1	All	Default Rate
EDUCATION				
1	7357	1856	9213	20.145447
2	9844	3204	13048	24.555487
3	3360	1178	4538	25.958572
4	90	4	94	4.255319
All	20651	6242	26893	23.210501

Illustration 4.2.13: Crosstab for 'EDUCATION'

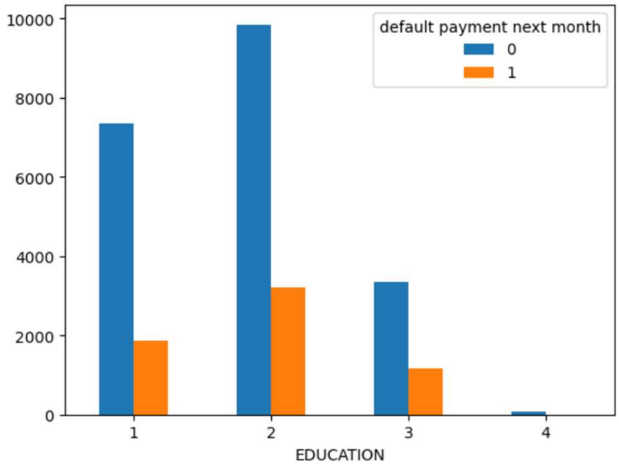


Illustration 4.2.14: Side-by-side bar chart for 'EDUCATION'

Illustrations 3.2.13 and 3.2.14 show that the 'EDUCATION' variable is represented by 1 for graduated school (master or PhD programme), 2 for university, 3 for high school, and 4 for others (non-graduate). According to Illustration 3.2.13, the lower the graduate level, the higher the default rate, which means the credit card user may not pay for the next month and the scorecard point should be assigned accordingly. However, notice that non-graduates pay on time and have the lowest default rate, so the scorecard point for non-graduates will be the highest in this attribute.

III. MARITAL STATUS

default payment next month	0	1	All	Default Rate
MARITAL_STATUS				
1	9118	3003	12121	24.775184
2	11311	3157	14468	21.820570
3	222	82	304	26.973684
All	20651	6242	26893	23.210501

Illustration 4.2.15: Crosstab for 'MARITAL STATUS'

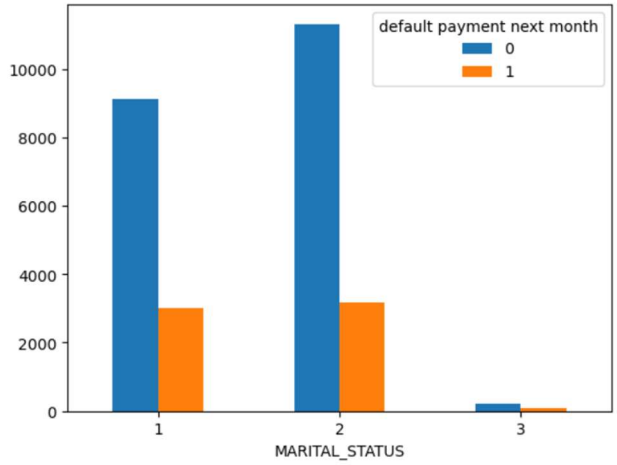


Illustration 4.2.16: Side-by-side bar chart for 'MARITAL STATUS'

Knowing that 1 represents married, 2 represents single, and 3 represents others (widows or divorces). According to the above crosstab, widows and divorcees have the highest default rate. The reason for this might be that it is difficult for one person to bear and handle all of the burden of a family, and hence the default rate is high. As a result, a lower score will be assigned to credit card users who are widowed or divorced.

IV. REPAYMENT STATUS IN SEPTEMBER

default payment next month	0	1	All	Default Rate
REPAYMENT_STATUS_IN_SEPTEMBER				
0	17311	2821	20132	14.012517
1	2411	1251	3662	34.161660
2	803	1837	2640	69.583333
3	76	244	320	76.250000
4	24	52	76	68.421053
5	11	13	24	54.166667
6	5	6	11	54.545455
7	2	7	9	77.777778
8	8	11	19	57.894737
All	20651	6242	26893	23.210501

Illustration 4.2.17: Crosstab for 'REPAYMENT STATUS IN SEPTEMBER'

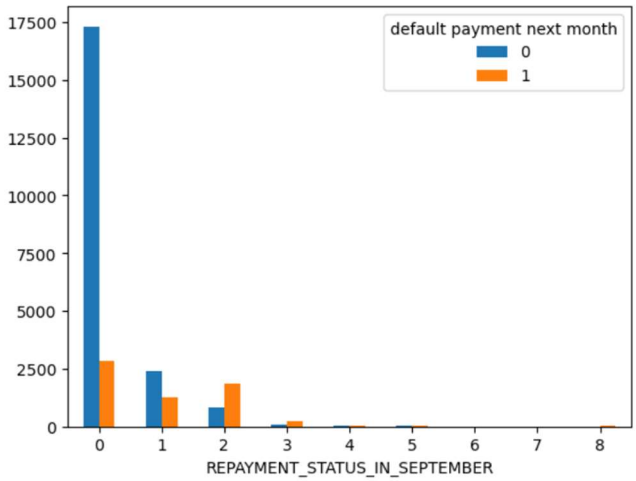


Illustration 4.2.18: Side-by-side bar chart for 'REPAYMENT STATUS IN SEPTEMBER'

Illustrations 3.2.17 and 3.2.18 show that the 'REPAYMENT STATUS IN SEPTEMBER' variables are represented by 0 for timely payment, 1 for a one-month payment delay, 2 for a two-month payment delay, and so on. According to Illustration 3.2.17, payments made on time have the lowest default rate, while payments delayed for seven months have the highest default rate. As a result, the category for this attribute with the highest scorecard point will be paid accordingly.

There are only 4 numerical attributes chosen in this report, which include Amount of given credit, Age, Amount of bill statement in September and Amount of previous payment in September. The boxplot represented this attribute situation, as shown in Illustration 3.2.19.

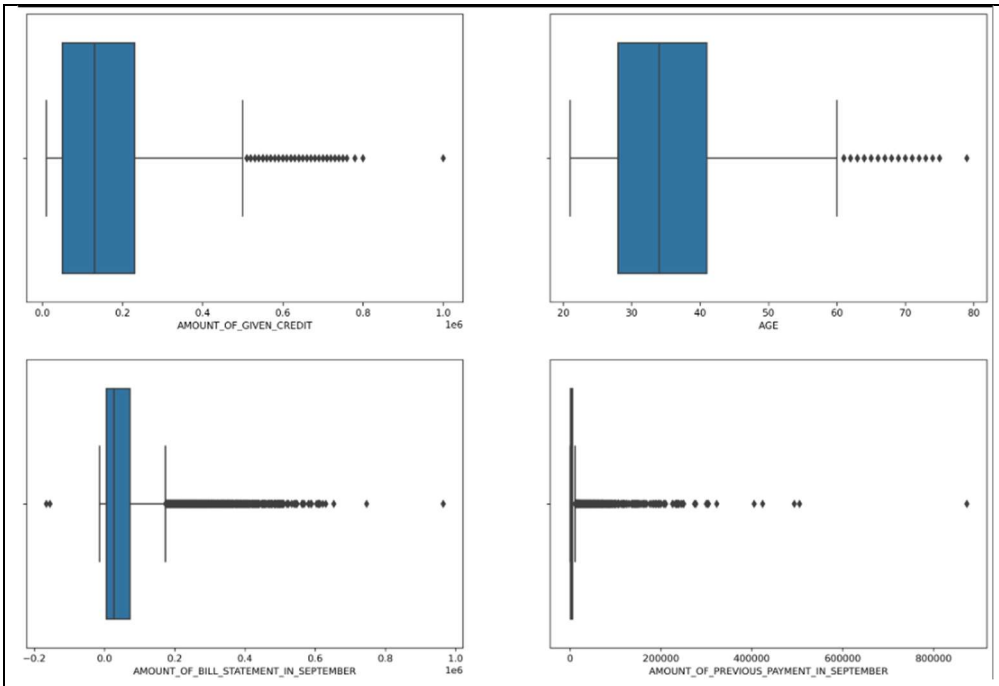


Illustration 4.2.19: Visualization for all numerical attributes

From Illustration 3.2.19, The outlier for 'AMOUNT OF GIVEN CREDIT' can be accepted since there is no limit to the number of credit cards you can own unless you earn less than RM36,000 per annum. Due to the terms and conditions, the credit card user must be over 21 years old to obtain a credit card, and the age in this report is all older than 21 years old, so the age attribute will not need to be bothersome. For the rest of the two numerical data sets, the interquartile range is used to handle the outlier for the remaining attributes. Once the outlier is identified, it will be tested by removing or replacing it with the 5th and 95th percentiles for the smaller and larger outliers, respectively.

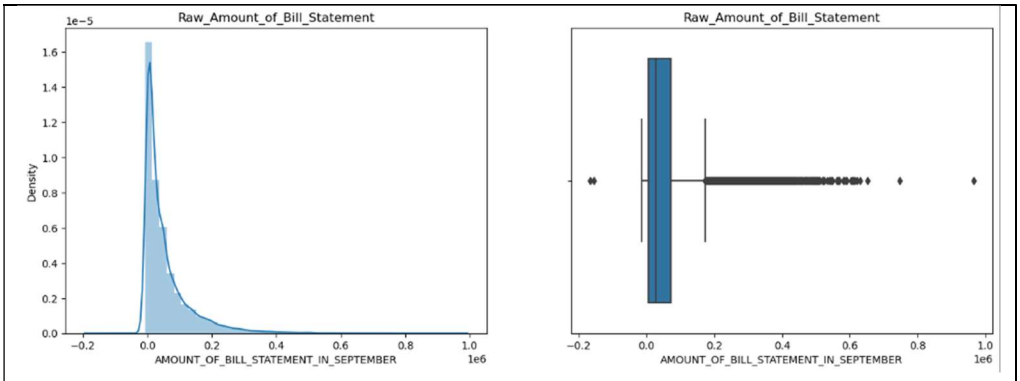


Illustration 4.2.20: Visualization of the raw 'AMOUNT OF BILL STATEMENT IN SEPTEMBER'

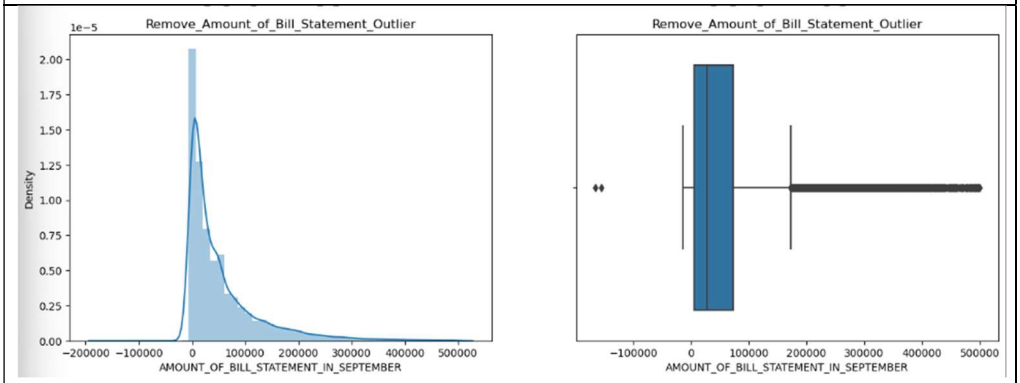


Illustration 4.2.21: Visualization of the 'AMOUNT OF BILL STATEMENT IN SEPTEMBER' after removing outlier

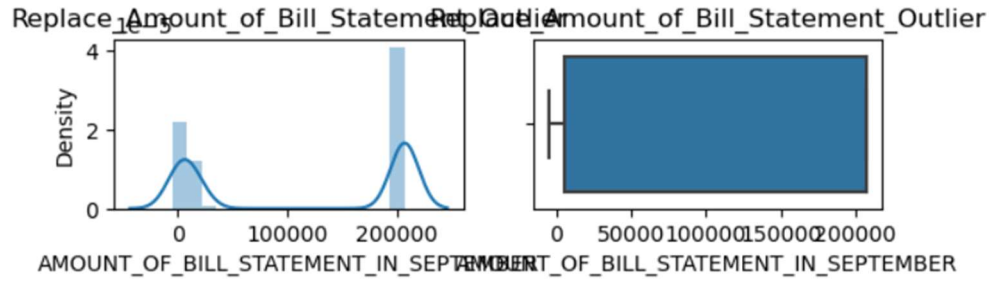


Illustration 4.2.22: Visualization of the 'AMOUNT OF BILL STATEMENT IN SEPTEMBER' after replacing outlier

When compared to Illustrations 3.2.21 and 3.2.22, the data looks more logical after the outlier is removed. Hence, the outlier in 'AMOUNT OF BILL STATEMENT IN SEPTEMBER' should be removed from the dataset.

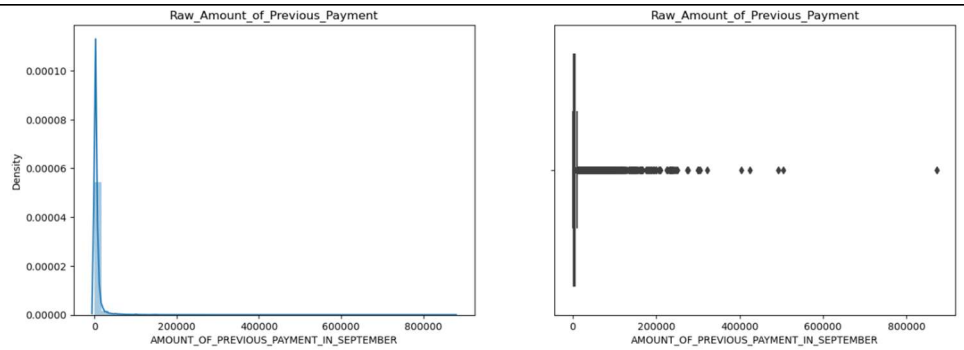


Illustration 4.2.23: Visualization of the 'AMOUNT OF PREVIOUS PAYMENT IN SEPTEMBER' after replacing outlier

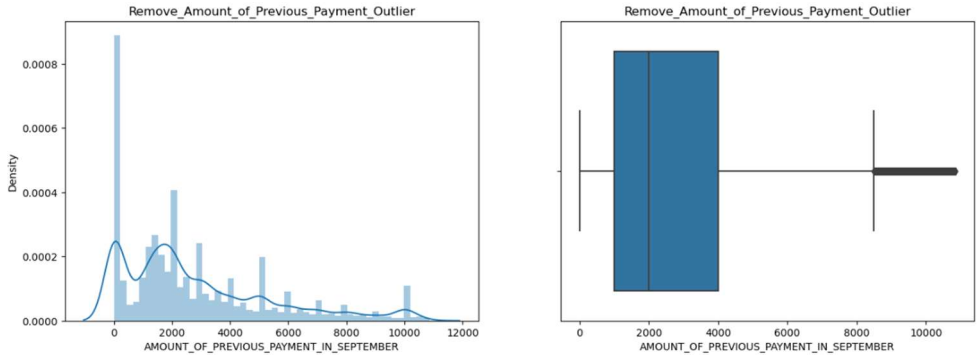


Illustration 4.2.24: Visualization of the 'AMOUNT OF PREVIOUS PAYMENT IN SEPTEMBER' after removing outlier

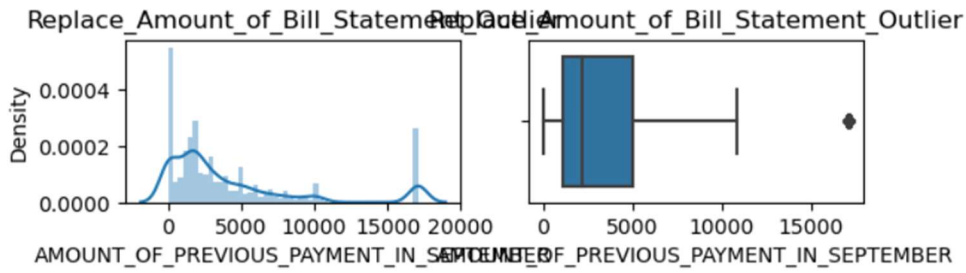


Illustration 4.2.25: Visualization of the 'AMOUNT OF PREVIOUS PAYMENT IN SEPTEMBER' after replacing outlier

When compared to Illustrations 3.2.24 and 3.2.25, the data looks more logical after the outlier is removed. Hence, the outlier in 'AMOUNT OF PREVIOUS PAYMENT IN SEPTEMBER' should be removed from the dataset.

After removing all the outliers, there are 24437 rows and 10 columns in the dataset.

Correlation again needs to be checked to make sure there is no highly correlated data, as illustrated in Illustration 3.2.26.

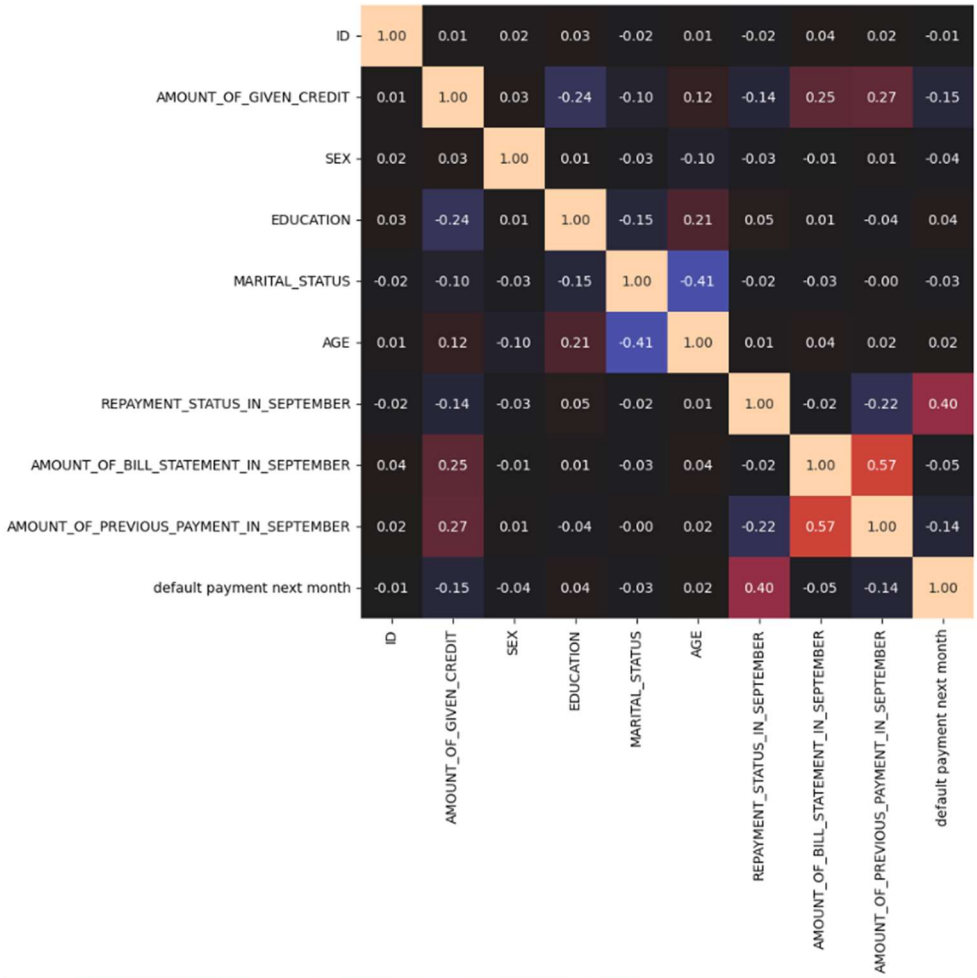


Illustration 4.2.26: Analysis of Correlation between attributes and target variable after removing all the outliers

4.3 Data Preprocessing (Feature Selection, Feature Binning and Feature Transforming)

Before running the model algorithm, the attribute's Information Value (IV) and Weight of Evidence (WoE) must be determined to select the important feature. As discussed in the previous section, IVs that are less than 0.2 are considered unimportant variables and can thus be removed from the dataset.

	Attributes_Name	IV_Value
0	REPAYMENT_STATUS_IN_SEPTEMBER	0.894160
1	AMOUNT_OF_PREVIOUS_PAYMENT_IN_SEPTEMBER	0.181639
2	AMOUNT_OF_GIVEN_CREDIT	0.170389
3	AMOUNT_OF_BILL_STATEMENT_IN_SEPTEMBER	0.035564
4	AGE	0.025855
5	ID	0.024945
6	EDUCATION	0.019325
7	SEX	0.009657
8	MARITAL_STATUS	0.008256

Illustration 4.3.1: IV value for all the attribute

As shown in Illustration 3.3.1, three attributes should be removed from the dataset because they are deemed unimportant by calculation IV, namely Education, Sex and Marital Status. However, in real life, Marital Status and Sex are important features for estimating the default rate. As a consequence, Education can be removed from the dataset. Thus, the remaining columns are 9, and the remaining rows are 24437.

Before converting the remaining significant data into WoE format, it is necessary to ensure that there is no irrelevant data. The ID is the data object's key, but it has no predictive purpose for estimating the credit card user's default rate. As a result, the dataset contains 8 columns and 24437 rows, including the target variable, the default payment next month.

WoE is a data transformation that ensures each attribute has the same weight. The WoE groups some features' characteristics and assigns a weightage to them, as shown in Illustrations 3.3.3 and 3.3.4.

	AMOUNT_OF_GIVEN_CREDIT
0	20000
1	120000
2	90000
3	50000
4	50000
...	...
26887	80000
26888	220000
26889	150000
26890	30000
26892	50000

Illustration 4.3.2: Example of Amount of Given Credit before bining by WoE

Information value of AMOUNT_OF_GIVEN_CREDIT is 0.159877

	Variable	Cutoff	N	Events	% of Events
0	AMOUNT_OF_GIVEN_CREDIT	(9999.999, 30000.0]	3861	1428	0.241665
1	AMOUNT_OF_GIVEN_CREDIT	(30000.0, 50000.0]	3387	950	0.160772
2	AMOUNT_OF_GIVEN_CREDIT	(50000.0, 60000.0]	772	226	0.038247
3	AMOUNT_OF_GIVEN_CREDIT	(60000.0, 80000.0]	2098	554	0.093755
4	AMOUNT_OF_GIVEN_CREDIT	(80000.0, 120000.0]	2732	707	0.119648
5	AMOUNT_OF_GIVEN_CREDIT	(120000.0, 150000.0]	2153	460	0.077847
6	AMOUNT_OF_GIVEN_CREDIT	(150000.0, 200000.0]	3102	591	0.100017
7	AMOUNT_OF_GIVEN_CREDIT	(200000.0, 230000.0]	1488	271	0.045862
8	AMOUNT_OF_GIVEN_CREDIT	(230000.0, 320000.0]	2580	418	0.070740
9	AMOUNT_OF_GIVEN_CREDIT	(320000.0, 780000.0]	2264	304	0.051447

Illustration 4.3.3: Example of Amount of Given Credit after bining by WoE

	Non-Events	% of Non-Events	WoE	IV
0	2433	0.131315	0.609956	0.067309
1	2437	0.131531	0.200745	0.005870
2	546	0.029469	0.260723	0.002289
3	1544	0.083333	0.117839	0.001228
4	2025	0.109294	0.090512	0.000937
5	1693	0.091375	-0.160224	0.002167
6	2511	0.135525	-0.303814	0.010788
7	1217	0.065684	-0.359219	0.007120
8	2162	0.116688	-0.500501	0.022997
9	1960	0.105786	-0.720866	0.039171

Illustration 4.3.4: WoE value of Amount of Given Credit after bining

The numerical data can be transformed into WoE as long as the WoE is increasing or decreasing linearly, as shown in Illustration 3.3.4.

Then, as shown in Illustrations 3.3.3 and 3.3.4, the Amount of Given Credit that falls between 9999.999 and 30000 is grouped as 0.60995. Hence, the number of data objects has been reduced. Repeating the same thing for other numerical and categorical data, the data has been cleaned and transformed and is ready to be fitted into the model.

4.4 Statistical Analysis

Several crucial aspects were looked at in the framework of the statistical study, as mentioned in subtitle 2.3. By closely examining the X and Y observations, outlier detection was investigated in subheading 3.4.1. Through this procedure, any anomalies that would bias our data were able to be found and evaluated. The relevance of each individual data point was then examined in section 3.4.2 to determine which ones should be excluded. It must be carefully reviewed if the removal of specific data points caused the analysis to change significantly. The Multiple Linear Regression was used to assess the usefulness of the chosen features as predictive variables, ensuring that the chosen features significantly contributed to the model. Lastly, in subtile 3.4.4, R-square and Adjusted R-square were relied on to ascertain the overall benefit of the chosen variables, validating their collective contribution to the predictive accuracy of our model.

4.4.1 Diagnostic for Leverage and Influence

The greater the residual, e_i, the greater the possibility of an outlier. There is no outlier when both the residual, e_i, and standardised residuals are checked, d_i because there is no absolute value of them larger than ±3 (standard deviation). As the standard deviation for d_i is extremely close to one, any point close to -3 or 3 is considered an outlier, but there is no such point in this case. The level of significance here is 0.05 because it is the typical alpha level, the total number of observations is 24437, and the total number of features picked is 7, and there is one target variable, namely default payment next month.

H_0 = Not an outlier

H_1 = Is an outlier

A. Test for Outlying Y Observations

As previously indicated, the crucial value for evaluating the default credit card user outlier differs between R-student residuals, t_i and studentized residuals, r_i.

The critical value for R-Student residuals is:

$$|ti| \geq t(1-\frac{0.05}{2(24437)}; 24437-7-1)$$

Figure 4.4.1.1: Enter the value into the Formula to calculate the critical value of R-Student Residuals

R-Student residuals have a critical value of 4.749964 here. As a result, each observation of the target variable that exceeds 4.749964 is considered an outlier of Y observations.

The critical value for Studentized residuals is:

$$|ri| \geq t(1-\frac{0.05}{2(24437)}; 24437-7)$$

Figure 4.4.1.2: Enter the value into the Formula to calculate the critical value of Studentized Residuals

Studentized residuals have a critical value of 4.749964 here. The critical values for R-Student Residuals and Studentized Residuals are the same.

The graph below shows that there are no outlier Y observations in this report by utilising the critical value because there is no absolute value bigger than the critical value for Studentized residuals and R-Student.

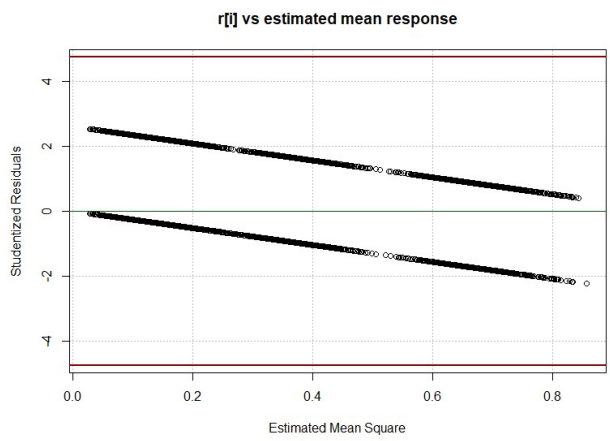


Figure 4.4.1.3: Studentized Residuals, r_i

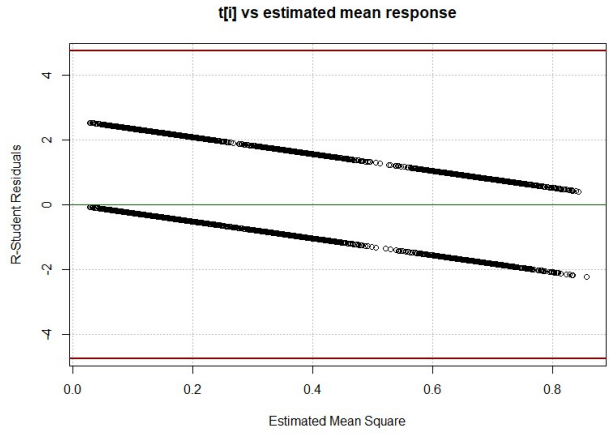


Figure 4.4.1.4: Bonferroni Test; R-Student Residuals, t_i

B. Test for Outlying X Observations

$$Hii > \frac{2\ (7)}{24437}$$

Figure 4.4.1.5: Enter the value into the Formula to calculate the critical value of Hat Matrix

The Hat Matrix value is roughly 0.0007, hence the 53 observations with more than 0.0007 are considered outliers with respect to X values. However, according to the Rule of Thumb, there are no high leverage values because no observation is greater than 0.5. Thus, the outlier can be ignored and kept in the dataset.

4.4.2 Identifying Influential Cases

Identifying influential cases entails determining whether or not any outliers left in this report will have a significant impact on the outcome. As the number of observations in this dataset exceeds 30, it counts as a big dataset; hence, all critical values are based on the critical value of the large dataset.

A. Influence on Single Fitted Value, DFFITS

$$|\text{DFFITS}| > 2\sqrt{\frac{(7)}{24437}}$$

Figure 4.4.2.1: Enter the value into the formula to calculate the critical value of DFFITS

The crucial value is around 0.0338, and each absolute observation larger than 0.0338 has an effect on the single fitted value. The single fitted value is influenced by 3185 observations in this dataset.

B. Influence on all Fitted Values, Cook's Distance

$$\text{Cook.i} < F(0.2;8,24437-7-1)$$
$$\text{Cook.i} > F(0.5;8,24437-7-1)$$

Figure 4.4.2.2: Enter the value into the formula to calculate the critical value of Cook's Distance

For the first critical value, those smaller than 1.378903, the regression fit is influenced, and all observations have a minor influence. There are no observations with values greater than the second critical value of 0.9180404, indicating that there are no major influence points on all fitted values.

C. Influence on the Regression Coeficients, DFBETAS

$$|\ \text{DFBETAS}\ | > \frac{2}{\sqrt{24437}}$$

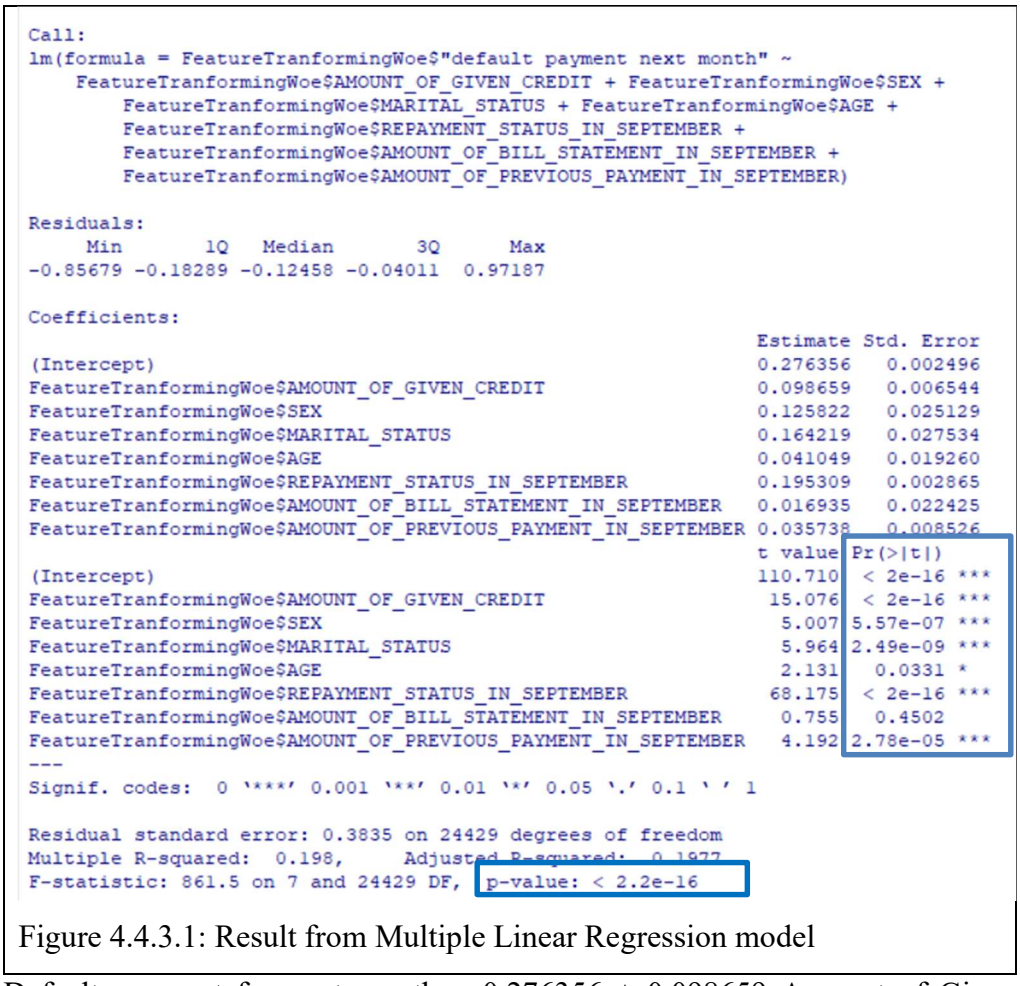
Figure 4.4.2.3: Enter the value into the formula to calculate the critical value of DFBETAS

The critical value is about 0.01279399, and over 1750 observations have been labelled as possible impact situations. However, no DFBETAS numbers surpass the critical values by a significant margin, therefore the situations may be less influential to the default credit card

user. As a result, there is no need to remove any outliers from this dataset.

4.4.3 Multiple Linear Regression

After confirming all the important features, Multiple Linear Regression is used to determine whether the response variable is associated with the numerical values of one or more quantitative variables.



```
Call:
lm(formula = FeatureTranformingWoe$"default payment next month" ~
    FeatureTranformingWoe$AMOUNT_OF_GIVEN_CREDIT + FeatureTranformingWoe$SEX +
        FeatureTranformingWoe$MARITAL_STATUS + FeatureTranformingWoe$AGE +
        FeatureTranformingWoe$REPAYMENT_STATUS_IN_SEPTEMBER +
        FeatureTranformingWoe$AMOUNT_OF_BILL_STATEMENT_IN_SEPTEMBER +
        FeatureTranformingWoe$AMOUNT_OF_PREVIOUS_PAYMENT_IN_SEPTEMBER)

Residuals:
     Min       1Q   Median       3Q      Max
-0.85679 -0.18289 -0.12458 -0.04011  0.97187

Coefficients:
                                                    Estimate Std. Error
(Intercept)                                         0.276356   0.002496
FeatureTranformingWoe$AMOUNT_OF_GIVEN_CREDIT        0.098659   0.006544
FeatureTranformingWoe$SEX                           0.125822   0.025129
FeatureTranformingWoe$MARITAL_STATUS                0.164219   0.027534
FeatureTranformingWoe$AGE                           0.041049   0.019260
FeatureTranformingWoe$REPAYMENT_STATUS_IN_SEPTEMBER 0.195309   0.002865
FeatureTranformingWoe$AMOUNT_OF_BILL_STATEMENT_IN_SEPTEMBER 0.016935   0.022425
FeatureTranformingWoe$AMOUNT_OF_PREVIOUS_PAYMENT_IN_SEPTEMBER 0.035738   0.008526
                                                    t value Pr(>|t|)
(Intercept)                                         110.710  < 2e-16 ***
FeatureTranformingWoe$AMOUNT_OF_GIVEN_CREDIT         15.076  < 2e-16 ***
FeatureTranformingWoe$SEX                             5.007 5.57e-07 ***
FeatureTranformingWoe$MARITAL_STATUS                  5.964 2.49e-09 ***
FeatureTranformingWoe$AGE                             2.131   0.0331 *
FeatureTranformingWoe$REPAYMENT_STATUS_IN_SEPTEMBER  68.175  < 2e-16 ***
FeatureTranformingWoe$AMOUNT_OF_BILL_STATEMENT_IN_SEPTEMBER   0.755   0.4502
FeatureTranformingWoe$AMOUNT_OF_PREVIOUS_PAYMENT_IN_SEPTEMBER   4.192 2.78e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3835 on 24429 degrees of freedom
Multiple R-squared:  0.198,     Adjusted R-squared:  0.1977
F-statistic: 861.5 on 7 and 24429 DF,  p-value: < 2.2e-16
```

Figure 4.4.3.1: Result from Multiple Linear Regression model

Default payment for next month = 0.276356 + 0.098659 Amount of Given Credit+ 0.125822 Sex + 0.164219 Marital Status + 0.041049 Age + 0.195309 Repayment Status in September + 0.016935 Amount of Bill Statement in September + 0.035738 Amount of Previous Payment in September

H_0 = Determining the default rate for the next month (target variable) using Amount of Given Credit, Sex, Marital Status, Age, Repayment Status in September, Amount of Bill Statement in September and Amount of Previous Payment in September as predictive variables are not useful.

H_1 = Determining the default rate for the next month (target variable) using Amount of Given Credit, Sex, Marital Status, Age, Repayment Status in September, Amount of Bill Statement in September and Amount of Previous Payment in September as predictive variables are useful.

As pointed out previously, the commonly used certain probability (alpha level) is 0.05 because the overall p-value (blue rectangle below in Figure 3.4.3.1) is close to zero and thus the p-value is less than 0.05. With alpha at 0.05, H_0 rejects and has sufficient evidence to conclude that determining the default rate for the next month (target variable) using Amount of given credit, Sex, Marital Status, Age, Repayment Status in September, Amount of bill statement in September and Amount of Previous Payment in September as predictive variables is useful.

Then, variables are checked to see which are less important in determining the target variable.

Table 4.4.3.1: The hypothesis for all important features

H_0 = Amount of Given Credit is not useful in determing the target variable H_1 = Amount of Given Credit is useful in determing the target variable
H_0 = Sex is not useful in determing the target variable H_1 = Sex is useful in determing the target variable
H_0 = Marital Status is not useful in determing the target variable H_1 = Marital Status is useful in determing the target variable
H_0 = Age is not useful in determing the target variable H_1 = Age is useful in determing the target variable
H_0 = Repayment Status in September is not useful in determing the target variable H_1 = Repayment Status in September is useful in determing the target variable
H_0 = Amount of Bill Statement in September is not useful in determing the target variable H_1 = Amount of Bill Statement in September is useful in determing the target variable
H_0 = Amount of Previous Payment in September is not useful in determing the target variable

H_1 = Amount of Previous Payment in September is useful in determing the target variable

Checking all the p-values for Amount of Given Credit to Amount of Previous Payment in September (blue rectangle above in Figure 3.4.3.1), only age has a p-value greater than 0.05, so it does not reject H_0 and concludes that age has no significant contribution to the model. However, since age is a required feature in the building scorecard, it can still be present in the dataset. Other features' p-values are all less than 0.05, providing sufficient evidence to conclude that the Amount of Given Credit, Sex, Marital Status, Age, Repayment Status in September, Amount of Bill Statement in September and Amount of Previous Payment in September all play a significant role in determining the default rate, and thus should all be included in the model.

4.4.4 R-Squared and Adjusted R-Squared Values

In the table below, the values of R^2 and R^2_{Adj} are compared. Using the first fifth independent variables, Amount of Given Credit, Sex, Marital Status, Age, and Repayment Status, as the first model to verify the value of the R^2 and R^2_{Adj}. Then, keep adding the feaure to the model.

Table 4.4.4.1: The R-Squared and Adjusteed R-Squared values

Model	R^2	R^2_{Adj}
Amount of Given Credit, Sex, Marital Status, Age, Repayment Status in September	0.19710274	0. 19693842
Amount of Given Credit, Sex, Marital Status, Age, Repayment Status in September, Amount of Bill Statement in September	0.19739938	0.19720226
Amount of Given Credit, Sex, Marital Status, Age, Repayment Status in September, Amount of Bill Statement in September, Amount of Previous Payment in September	0.19797627	0.19774646

When a variable that might not be beneficial to the model is included, the values of R-Square and Adjusted R-Squared decrease. Hence, the independent features listed above were strongly associated with the default credit card as the values of R^2 increased. Therefore, there is strong evidence that those independent variables that were chosen, which include Amount of Given Credit, Sex, Marital Status, Age, Repayment Status in September, Amount of Bill Statement in September and Amount of Previous Payment in September are useful for predicting the default credit card.

4.5 Model Development

There are several ways to assess classifier accuracy, including Resubstituting, Hold-out, K-Fold Cross-validation, Leave One Out Cross-validation and Repeated K-fold Cross-validation. Resubstituting Validation presents a drawback by training and testing the model on the same data, leading to overfitting (Yadav and Shukla, 2016). Hold-out Validation divides the data into two parts to counter this limitation by training on one half and testing on the other (Yadav and Shukla, 2016). K-Fold By dividing the data into k equal sets, with one set set aside for testing, and repeating the procedure k times, cross-validation is a useful method for evaluating the model's generalisation performance (Yadav and Shukla, 2016). The size of the validation set depends on the data type (parametric or nonparametric). Cross-validation helps in selecting the appropriate modelling or model selection technique and comparing performance across algorithms for optimal results (Yadav and Shukla, 2016). Nevertheless, using Leave One Out Cross-validation with only one instance for testing can lead to high variability (Yadav and Shukla, 2016).

The repeated holdout method involves randomly selecting and withholding a portion of the training set for testing purposes (Kim, 2009). This is done when there are no independent testing samples available from the training set. Typically, one-third of the training sample is reserved for this purpose (Kim, 2009). The hold-out method is commonly used to split data into training and testing sets, and techniques such as k-fold cross-validation, hold-out validation, and bootstrapping can be used to improve the classifier. In this case, using the holdout method to split the dataset into a 70% training set and a 30% testing set with the state 42. As illustrated in Illustration 3.5.1, the training set contains 70%

of the dataset, which is 17105 rows and 8 attributes. One of the attributes is the target variable (default credit card for the next month). While the testing set, which contains 30% of the dataset, comprises 7332 rows and 8 attributes, one of the attributes is the target variable.

```
train size X :  (17105, 7)
train size y :  (17105,)
test size X :  (7332, 7)
test size y :  (7332,)
```

Illustration 4.5.1: Holdout method to split dataset in this report into 70% training and 30% testing set

CHAPTER 5

RESULTS AND DISCUSSION

The classification output can be obtained after running all the algorithms mentioned above. As a result, the model's accuracy will be determined for determining credit card default next month. The greater the accuracy, the greater the confidence in the model. This increases confidence in the features used to determine the next month's default credit card. Hence, financial institutions can reduce their financial losses.

In subtitle 4.1, since some of the algorithms have specific outputs, there is a need to show those outputs. The confusion matrix for all the algorithms is obtained from Python, so in subtilte 4.2 the information from the confusion matrix is demonstrated on how to extract it. In Subsection 4.3, classification reports are generated for all the algorithms. The ROC curve is used to analyse which model is the best model for a binary classifier. In subtitle 4.4, the ROC curve for all algorithms will be explained. The confidence interval of the model is used to estimate the range with an upper and lower bound based on a sample and a desired confidence level is shown in subtitle 4.5. Subtitles 4.1 to 4.5 cover all the results of the model development process. Subtitle 4.6 then covers the scorecard's development.

5.1 Output for algorithm

5.1.1 Logistic Regression

Formula for Linear Regression in Figure 4.1.1.1.

```
Linear Regression Formula:
 Default payment next month =
   [-1.15104506](Intercept)
 + [0.6291877707329642] (AMOUNT_OF_GIVEN_CREDIT)
 + [0.8365100029168979] (SEX)
 + [0.9311502649702703] (MARITAL_STATUS)
 + [0.19456493423596663] (AGE)
 + [0.9495974479369806] (REPAYMENT_STATUS_IN_SEPTEMBER)
 + [0.03990530262512369] (AMOUNT_OF_BILL_STATEMENT_IN_SEPTEMBER)
 + [0.3350665584263777] (AMOUNT_OF_PREVIOUS_PAYMENT_IN_SEPTEMBER)
```

Figure 5.1.1.1: Equation for Linear Regression obtained

Figure 5.1.1.2: Sigmoid Function for Amount of Given Credit

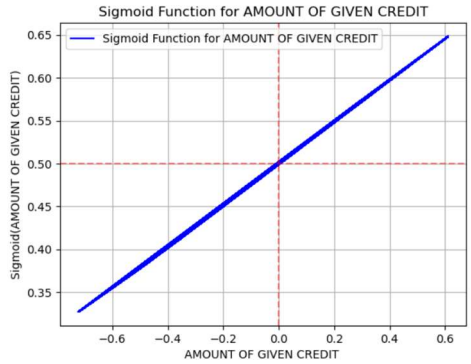


Figure 5.1.1.3: Sigmoid Function for SEX

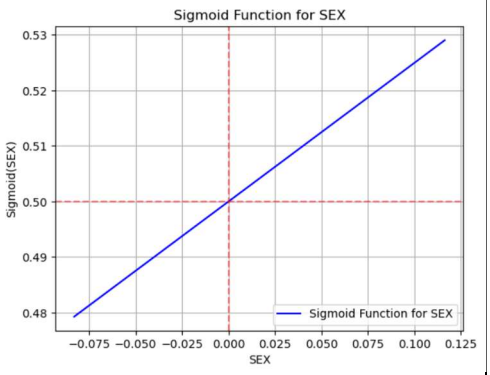


Figure 5.1.1.4: Sigmoid Function for MARITAL STATUS

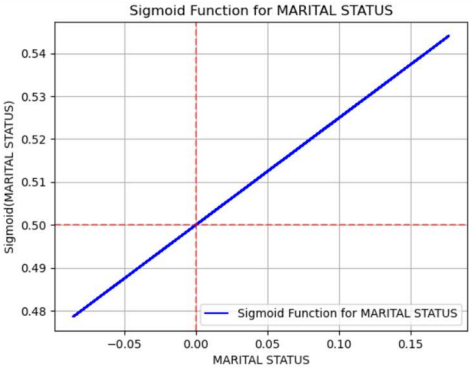


Figure 5.1.1.5: Sigmoid Function for AGE

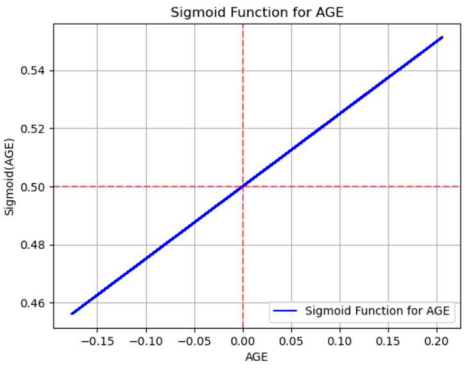


Figure 5.1.1.6: Sigmoid Function for REPAYMENT STATUS IN SEPTEMRBER

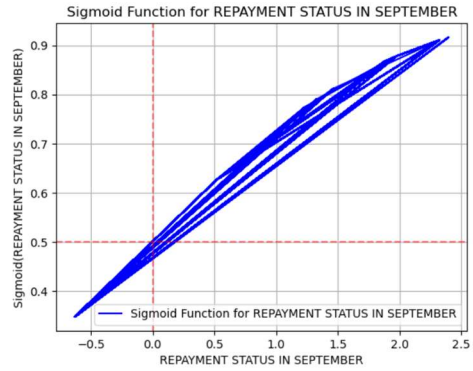


Figure 5.1.1.7: Sigmoid Function for AMOUNT OF BILL STATEMENT IN SEPTEMBER

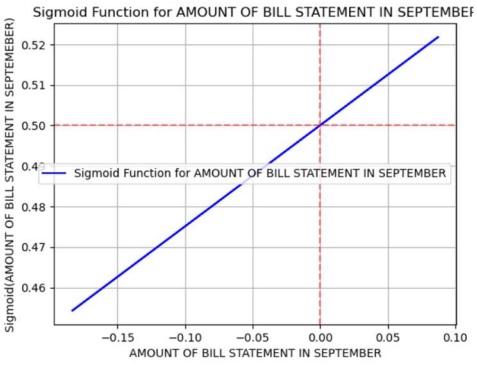
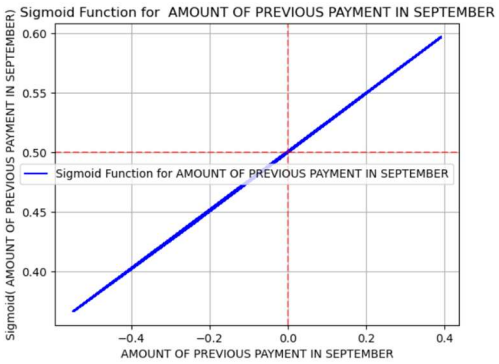


Figure 5.1.1.8: Sigmoid Function for AMOUNT OF PREVIOUS PAYMENT IN SEPTEMBER



The sigmoid function is an S-shaped curve that cannot be defined as a straight line. It converts every real integer to a value between 0 and 1, but it never quite reaches 0 or 1 for any finite input. It might be a data use issue. Let's have a look at the Logistic Regression's performance in the subtitles below (Subtitles 4.2, 4.3, and 4.4).

5.1.2 K - Nearest Neighbour

As previously stated, there is no proper approach to find K values, but K values must be known. In the preliminary stage, a range of K values ranging from 10 to 31 were examined on two metrics: accuracy score and error rate.

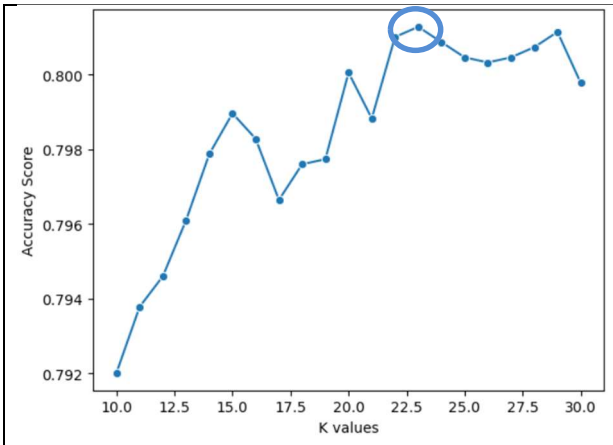


Figure 5.1.2.1: K values against Accuracy Score

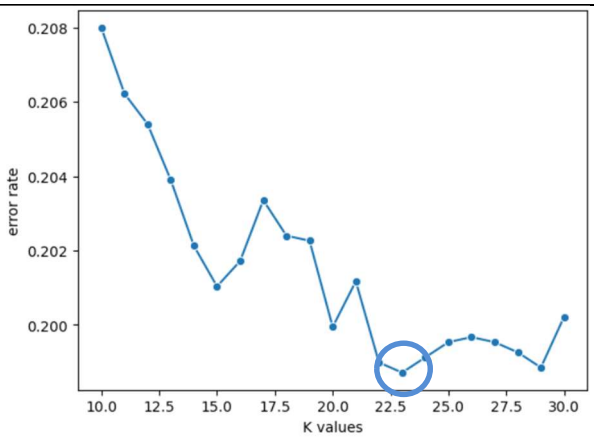


Figure 5.1.2.2: K values against Error Rate

The error rate, also known as the misclassification rate, is used to calculate the proportion of misclassified cases in the predictions of a classification model. It works in the same way as the confusion matrix, with false positives and false negatives. It is obtained by dividing the total number of cases in the dataset by the number of misclassified instances. The error rate indicates how well the classifier works overall, with lower values suggesting better performance.

The accuracy rate is the proportion of correctly categorised instances in the predictions of a classification model. It is determined by dividing the total number of occurrences in the dataset by the number of appropriately classified instances. The accuracy rate is a popular metric because it provides a basic and straightforward measurement of how well the classifier works. Better performance is indicated by higher accuracy values.

According to Figures 4.1.2.1 and 4.1.2.2, the best-fitting K value is 23, as it has the highest accuracy score and lowest error rate. As a result, the K values of 23 are used to train and test the KNN model. The model's output included a confusion matrix (shown in subtitle 4.2), a classification report (shown in subtitle 4.3), and a ROC curve (shown in subtitle 4.4).

5.1.3 Gaussian Naïve Bayes

The compatibility with the data in Gaussian Naive Bayes can be shown by a density plot and serves as a suitable starting point for understanding its classification behavior. In the density plots below, 1 indicates a default payment

for the following month in blue and 0 indicates no default payment for the following month in orange.

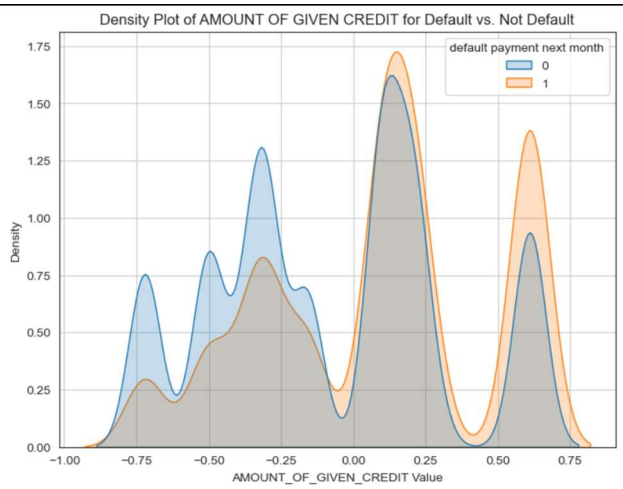


Figure 5.1.3.1: Density plot of Amount of Given Credit

Figure 4.1.3.1 shows that the peak value range for Amount of Given Credit is the same whether it is a default or not.

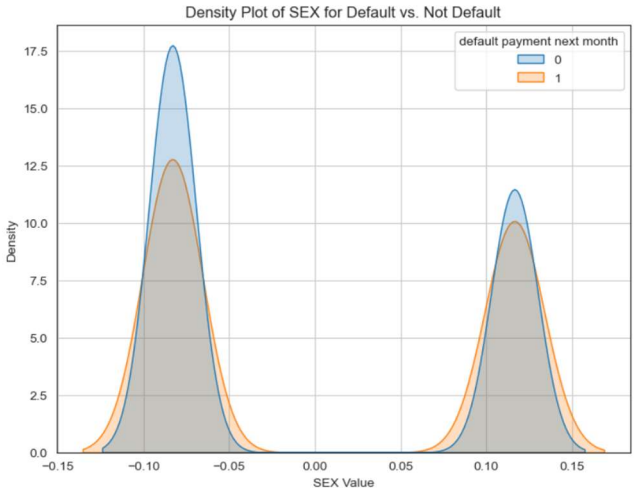


Figure 5.1.3.2: Density plot of SEX

Knowing that 1 represents male and 2 represents female in Sex. According to Figure 4.1.3.2, whether male or female, they are mostly not default.

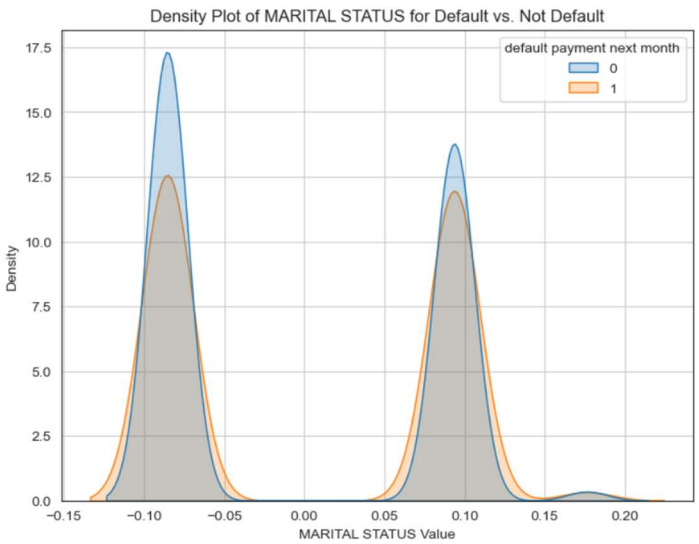


Figure 5.1.3.3: Density plot of Marital Status

Knowing that 1 denotes married, 2 denotes single, and 3 denotes other marital statuses. However, regardless of the credit card user's marital status, they are unlikely to miss payment for the next month.

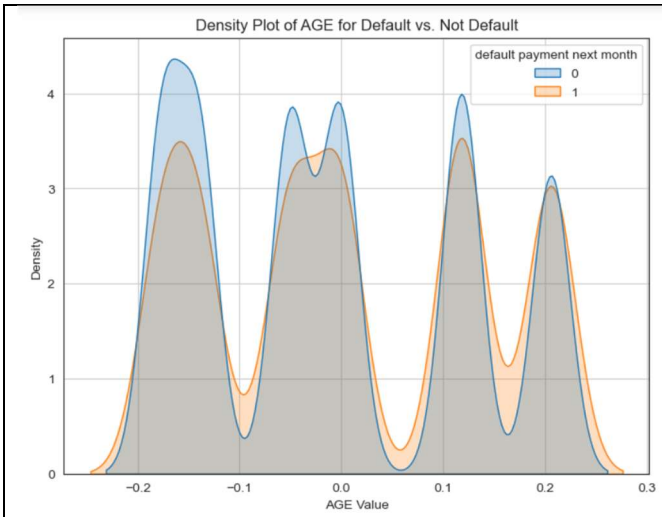


Figure 5.1.3.4: Density plot of Age

The entire age range is not about to be the default payment for the following month.

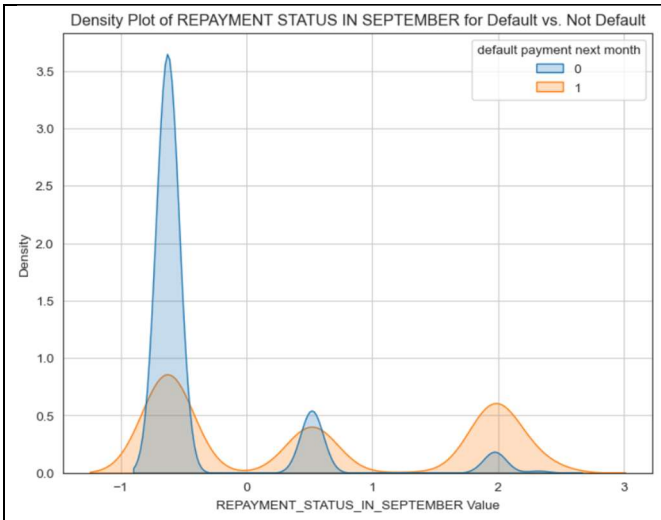


Figure 5.1.3.5: Density plot of Repayment Status in September

Knowing that 0 indicates payment on time, 1 represents a one-month delay, 2 denotes a two-month delay, and so on. As demonstrated by Figure 4.1.3.5, the longer the payment delay, the greater the likelihood of a default payment occurring.

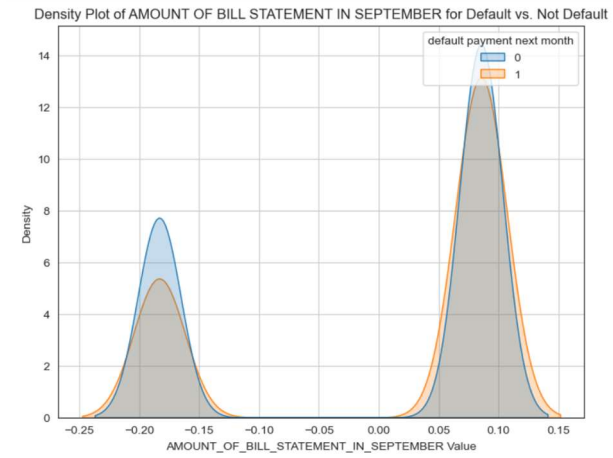


Figure 5.1.3.6: Density plot of Amount of Bill Statement in September

Credit card users are more likely to become non-default for the next month, regardless of the value range of the Amount of Bill Statement in September, as shown in Figure 4.1.3.6.

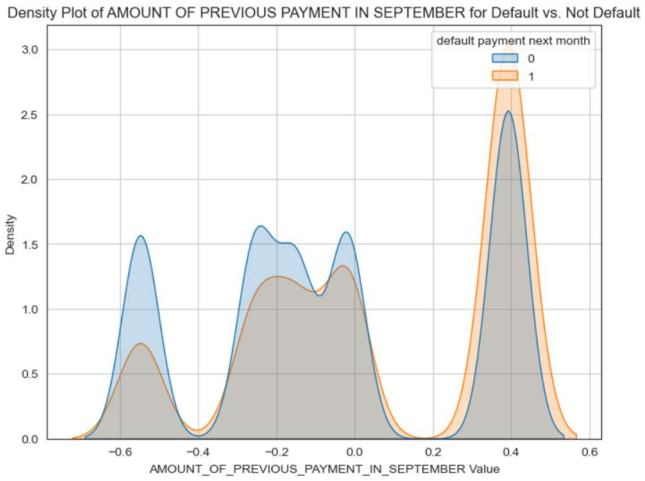


Figure 5.1.3.7: Density plot of Amount of Previous Payment in September

Figure 4.1.3.7 shows that the greater the Amount of Previous Payment in September, the more likely it is that the credit card user will default on the next month's payment.

5.1.4 Decision Tree

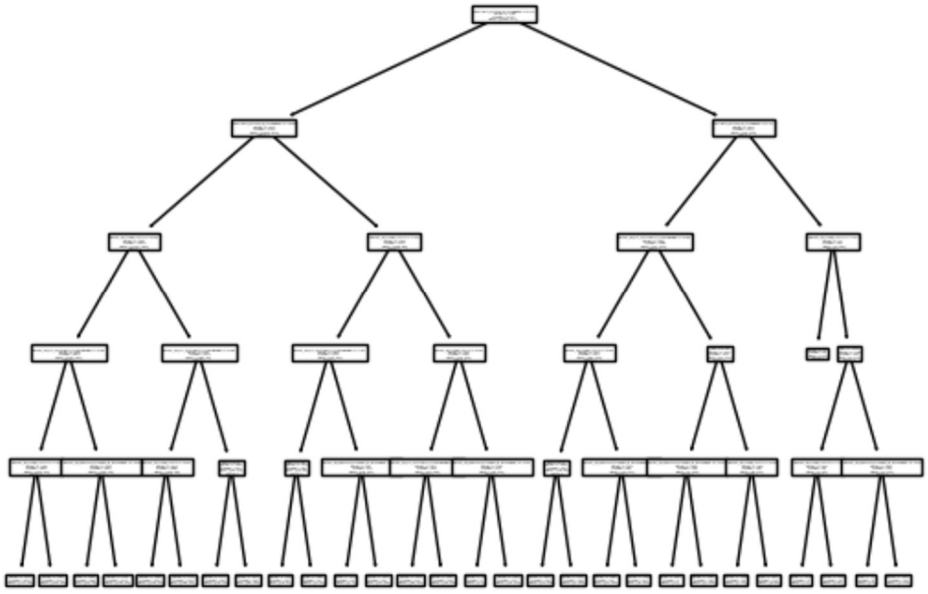


Figure 5.1.4.1: Decision Tree of 5 levels of depth

The level or depth of a decision tree is an essential element that can have a substantial influence on the tree's performance and behaviour. In this case, a decision tree of 5 levels of depth was used as shown in Figure 4.1.4.1. The model's output included a confusion matrix (shown in subtitle 4.2), a classification report (shown in subtitle 4.3), and a ROC curve (shown in subtitle 4.4).

5.1.5 Extreme Gradient Boosting

Since Extreme Gradient Boosting is an enhancement to the Gradient Boosting model to minimise overfitting, and Gradient Boosting is primarily used to reduce model error. As a result, the model's error before and after can be observed by applying gradient boosting and extreme gradient boosting. The Mean Square Error (MSE) is a frequent error used to assess the efficiency of a regression model, including Gradient Boosting. MSE calculates the average squared difference between predicted and actual values.

```
Mean Squared Error for Gradient Boosting: 0.20444626295690124
Mean Squared Error for Extreme Gradient Boosting: 0.20444626295690124
```

Figure 5.1.5.1: The MSE for Gradient Boosting and Extreme Gradient Boosting

Figure 4.1.5.1 shows that both the Gradient Boosting and Extreme Gradient Boosting models obtain the same Mean Square Error value. It implies that both models perform equally well on this dataset. As a result, the conclusion may be drawn that there is no possible overfitting and that feature selection is successful in this case.

5.2 Confusion Matrix

The performance of classification models is determined by a confusion matrix on a set of test data (Agarwal et al., 2021). According to Guo et al. (2008), the number of examples accurately identifying positive and negative examples is denoted as True Positive (TP) and True Negative (TN). The number of misclassified positive and negative examples is denoted as False Negative (FN) and False Positive (FP), respectively (Guo et al., 2008). Illustration 4.2.1 depicts the confusion matrix arrangement. Equations from Eq. 1 to Eq. 5 in Illustration 4.2.2 can be used to represent accuracy, precision, recall, FP rate, and TP rate.

		Predicted	
		0	1
Actual	0	TN	FP
	1	FN	TP

Illustration 5.2.1: Arrangement of Confusion Matrix

Accuracy = (TP + TN)/(TP + FN + FP + TN)

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

FP = FP/(FP + TN)

TP = TP/(TP+FN)

Illustration 5.2.2: Equation to calculate accuracy, precision, recall, FP rate and TP rate

For example, the confusion matrix for Logistic regression that obtained as Illustration 4.2.3.

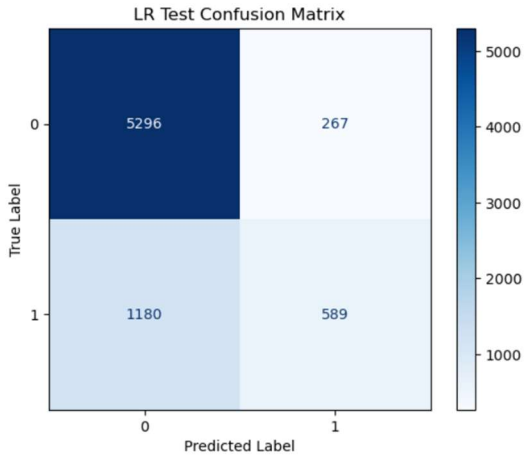
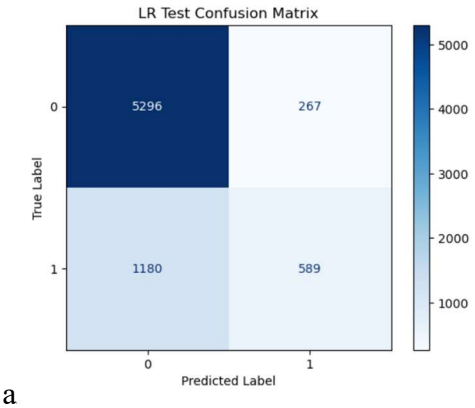


Illustration 5.2.3: Confusion matrix for logistic regression for testing data

True Negatives (TN) equal 5296 in this scenario, indicating that 5296 users are not default for both predicted and actual results. The credit card holder is not in default, and after checking, it is discovered that the credit card holder is not in default for the following month. False Positives (FP) equal 267, indicating that the predicted result is "yes," but the actual result is "no". The default of 267 credit card holders is predicted for next month, but the actual result is not a default. The number of False Negatives (FN) is 1180, indicating that the predicted result is "no," but the actual result is "yes". In this case, 1180 users are predicted to be non-default for the next month, but upon further investigation, they are default credit card users. True Positives (TP) in this case are 589, which means that 589 users are predicted to default on the next month's credit card and must default after checking.

Since False Positives and False Negatives are incorrectly predicted, the preferred result is a low frequency of False Positives and False Negatives. The confusion matrix for all of the algorithms selected above, including Logistic Regression (LR), K-Nearest Neighbour (KNN), Gaussian Nave Bayes (GNB), Decision Tree (DT), and Extreme Gradient Boosting (XGboost), is shown in Illustrations 4.2.4 to 4.2.8.



a

Illustration 5.2.4: Testing for Logistic regression

FN + FP = 1180 + 267 = 1447

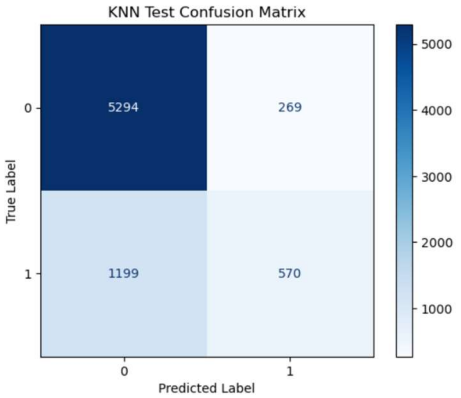


Illustration 5.2.5: Testing set for KNN

FN + FP = 1199 + 269 = 1468

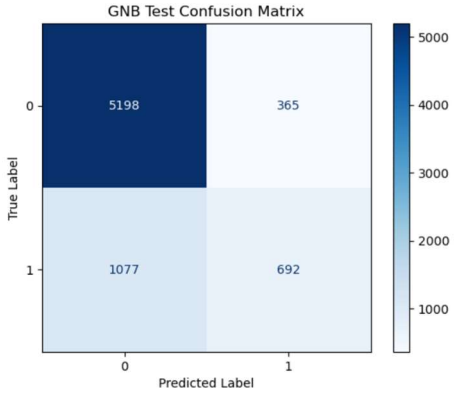


Illustration 5.2.6: Testing set for GNB

FN + FP = 1077 + 365 = 1442

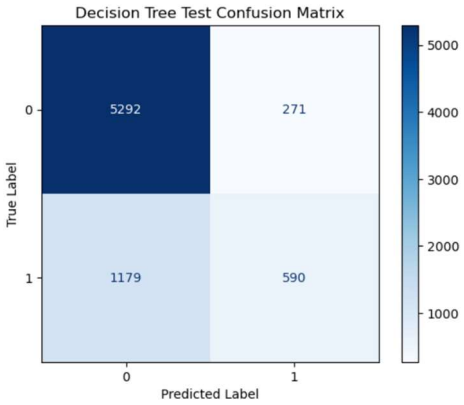


Illustration 5.2.7: Testing set for Decision Tree

FN + FP = 1179 + 271 = 1450

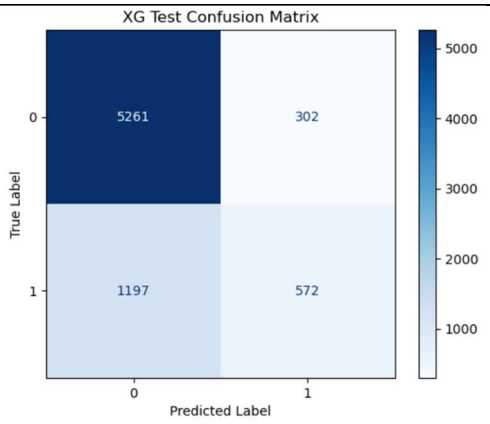


Illustration 5.2.8: Testing set for XGboost

FN + FP = 1197 + 302 = 1499

The least frequent combination of False Positive and False Negative is 1442, which is the Gausian Nave Bayes. However, all of the algorithms' frequent combinations of False Positive and False Negative are nearly the same. As a result, the conclusion here is that all of the algorithms are performing well when using those selected features to estimate the default payment.

5.3 Classification Report

The classification report can be used to obtain the classification model parameters, such as accuracy, precision, recall, and F1-score (Agarwal et al., 2021) as shown in Illustration 4.3.1. A classification model's accuracy base parameter is used to assess how accurately the model learns the dataset's data pattern and how well it can forecast future data (Agarwal et al., 2021). Precision is defined as the proportion of properly interpreted positive findings compared to all positive results (Agarwal et al., 2021). The recall is calculated by dividing the percentage of correctly observed positive findings by the total number of observations (Agarwal et al., 2021). The F1 score is used when the false positives and false negatives are not the same (Agarwal et al., 2021).

The training set is used to train the model and teach it how to recognise patterns, as previously stated, whereas the testing set is used to determine whether the built model is on the right track. As a result, the classification report can only look at the testing set and see if the models are working properly.

	Model	Accuracy	Precision	Recall	F1 Score
0	Test Logistic Regression	0.802646	0.688084	0.332956	0.448762
1	Test KNN	0.801282	0.668467	0.349915	0.459369
2	Test GNB	0.803328	0.654683	0.391181	0.489738
3	Test DT	0.802237	0.685250	0.333522	0.448669
4	Test XG	0.795554	0.654462	0.323347	0.432841

Illustration 5.3.1: Classification Report for all algorithm

It is discovering that all the algorithms had an accuracy of around 80%. As previously stated, the greater the accuracy, the more confident the model and features are in predicting the credit card user's default which means that all the models have about 80% confidence in predicting the default credit card holder. According to Sharma et al. (2018), the best algorithm is the one with the highest accuracy, recall, precision, and F1-score. According to Illustration 4.3.1, Gaussian Naïve Bayes is the most effective in determining the default of a credit card holder because it has the best accuracy, precision, recall, and F1 score when compared to others. By comparing all of the algorithms, it was discovered that there was no significant difference between them. Hence, proving that all of the algorithms here predict well with the feature selected.

5.4 Receiver Operating Characteristic (ROC) Curve

The ROC curve is used to evaluate the usefulness of a model. The larger the area of the ROC curve, the more useful the model (Ekelund, 2012). It is a graph in which the y-axis indicates the True Positive rate, and the x-axis represents the False Positive rate. It is also known as precision-recall curves (Ekelund, 2012). The ROC curve has a value ranging from 0 to 1. The more closely the ROC curve approaches 1, the more it approaches the upper left corner, indicating that the model is more significant. (Ekelund, 2012). As shown in Illustration 4.4.1, Model A is more useful than Model B because it has a larger area.

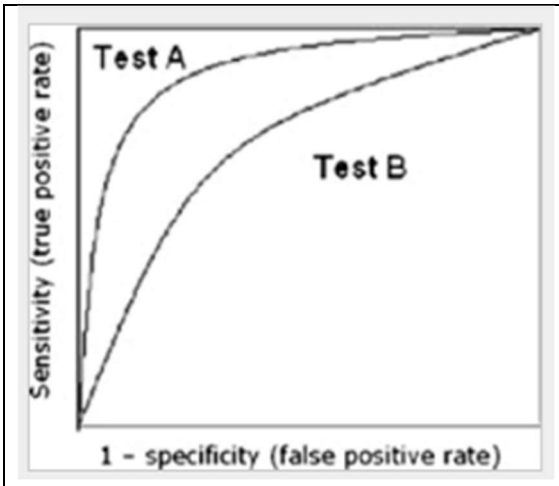


Illustration 5.4.1: ROC curve for both Model A and Model B.

The ROC curve is describing by using Rule of Thumb by using the Area Under ROC Cuve (AUC) value (as Table 4.4.2) (Ekelund, 2012).

Value AUC	**Category of Model**
0.9 to 1.0	The model is very good for prediction.
0.8 to 0.9	The model is good for prediction.
0.7 to 0.8	The model is fair for prediction.
0.6 to 0.7	The model is poor for prediction.
0.5 to 0.6	The model is fail for prediction.

Table 5.1.5.1: Rule of Thumb for describing ROC curve

As shown in Table 4.1.3.1, if the AUC value is 0.8, the model is good for prediction. In this case, it is useful for predicting which user will be the default. From Illustration 4.4.2 to Illustration 4.4.6, showing the ROC curve for all the algorithms including Logistic Regression (LR), K-Nearest Neighbour (KNN), Gaussian Nave Bayes (GNB), Decision Tree (DT), and Extreme Gradient Boosting (XGboost),

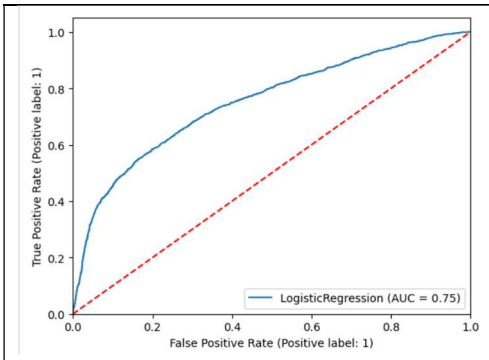


Illustration 5.4.2: ROC curve for LR

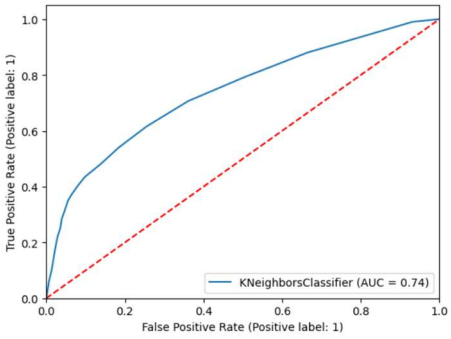


Illustration 5.4.3: ROC curve for KNN

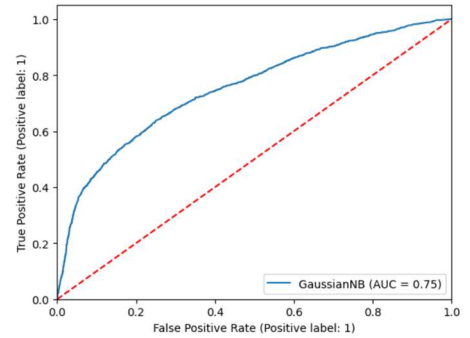


Illustration 5.4.4: ROC curve for GNB

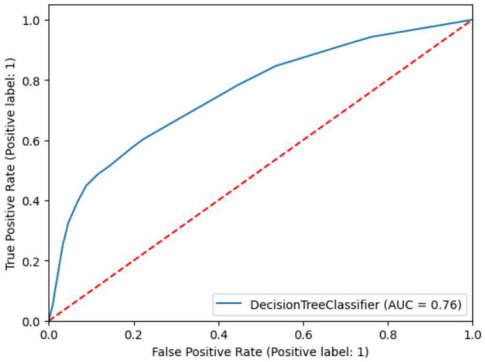


Illustration 5.4.5: ROC curve for DT

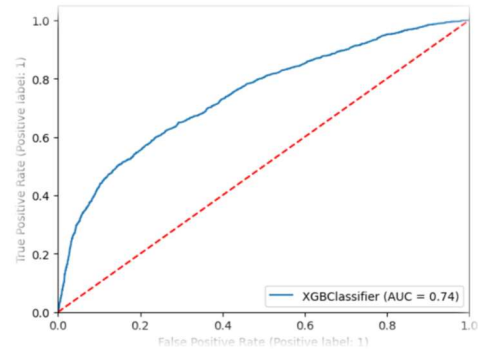


Illustration 5.4.6: ROC curve for XGboost

	Model	AUC value
0	Test Logistic Regression	0.75
1	Test KNN	0.74
2	Test GNB	0.75
3	Test DT	0.76
4	Test XG	0.74

Illustration 5.4.7: AUC value for all ROC curve

Table 4.4.1 shows that the higher the value of the area under the ROC curve (AUC), the better the prediction and thus the more useful the model. When comparing the AUC values in Illustration 4.4.7, it is clear that Decision Tree has the highest AUC values compared to others, making it the best model for predicting credit card default rates. However, since there is no big difference between them, all the algorithms are working well with the feature selected above. According to the results of subtitles 4.2 (Confusion Matrix) and 4.3 (Classification Report), all the algorithms have roughly the same result. As a result of combining subtitles 4.2, 4.3 and 4.4, all the models used here can correctly predict the credit card default rate.

In conclusion, one of the goals is fulfilled, which is to identify which data mining algorithm is more suitable for binary classification output since our target is to find whether the user is defaulting for next month. From all the results that getting above, it is clearly stated that all the algorithms used here, including Logistic Regression, K- Nearest Neighbour, Gaussian Naïve Byaes, Decision Tree and Extreme Gradient Boosting can determine the binary classifier problem.

5.5 Confidence Interval

The purpose of a confidence interval (CI) is to estimate the range with an upper and lower bound based on a sample and a desired confidence level (Hazra, 2017). Illustration 4.5.1 illustrates the formula for calculating CI.

$$\text{CI} = \text{Point estimate} \pm \text{Margin of error}$$

$$\text{Point estimate} \pm \text{Critical value (z)} \times \text{Standard error of point estimate}$$

Illustration 5.5.1: Formula for calculating Confidence Interval (CI)

Any level of certainty can be used for confidence intervals (CI), even though the 95% CI is the most frequently used (Hazra, 2017). In this case, a 75% confidence interval is used because the algorithm has at least a 75% chance of correctly predicting the default credit card holder from subtitle 4.3.

```
                                                                  12.5 %
(Intercept)                                                   0.273484538
FeatureTranformingWoe$AMOUNT_OF_GIVEN_CREDIT                  0.091130749
FeatureTranformingWoe$SEX                                     0.096914546
FeatureTranformingWoe$MARITAL_STATUS                          0.132545129
FeatureTranformingWoe$AGE                                     0.018892689
FeatureTranformingWoe$REPAYMENT_STATUS_IN_SEPTEMBER           0.192013602
FeatureTranformingWoe$AMOUNT_OF_BILL_STATEMENT_IN_SEPTEMBER  -0.008862553
FeatureTranformingWoe$AMOUNT_OF_PREVIOUS_PAYMENT_IN_SEPTEMBER 0.025930634
                                                                  87.5 %
(Intercept)                                                   0.27922771
FeatureTranformingWoe$AMOUNT_OF_GIVEN_CREDIT                  0.10618689
FeatureTranformingWoe$SEX                                     0.15473023
FeatureTranformingWoe$MARITAL_STATUS                          0.19589366
FeatureTranformingWoe$AGE                                     0.06320486
FeatureTranformingWoe$REPAYMENT_STATUS_IN_SEPTEMBER           0.19860488
FeatureTranformingWoe$AMOUNT_OF_BILL_STATEMENT_IN_SEPTEMBER   0.04273168
FeatureTranformingWoe$AMOUNT_OF_PREVIOUS_PAYMENT_IN_SEPTEMBER 0.04554601
```

Illustration 5.5.2: 75% Confidence Interval for all features

Using Amount of Givne Credit as an interpretation, with a 75% confidence interval, estimate that the change in the probability of the mean default rate when the amount of given credit is increased by 1 NT dollar, holding the other variables value constant, is somewhere between 0.2735 and 0.2792.

5.6 Scorecard Building

The creation of a credit risk scorecard involves two crucial steps. In subtitle 4.6.1, factor weightage is calculated by considering the feature coefficients derived from the model. The ratio of each feature's coefficient to the sum of all feature coefficients is determined, and points are assigned to each feature based on this ratio. This step ensures that each feature's contribution to the scorecard accurately reflects its importance in assessing credit risk. In subtitle 4.6.2, the feature binning, previously determined from selected features, plays a pivotal role. By referencing the weight assigned to each binning of a selected feature and employing the ratio method once more, points are allocated to each binning

category. This comprehensive approach to scoring factors and feature binning ensures the credit risk scorecard's effectiveness in precisely evaluating and managing credit risk for individuals or entities. Subsection 4.6.3 uses the dataset to create individual credit scores based on the criteria and binning weights determined in previous stages. Data from each borrower or entity is compared to these predetermined standards, and the resulting points are totaled. These factors are finally used to calculate an overall credit score, which offers a clear and accurate indication of creditworthiness. Financial institutions can use this carefully crafted credit scorecard, which was created through factor weighting and feature binning, as a useful tool to help them manage credit risk and make educated lending decisions while maintaining fairness and openness in the evaluation process.

5.6.1 Count Factor Weightage

Creating a score system using the weighted variables. This can be accomplished via a mathematical formula or by establishing bins or categories for each variable. Determining how the individual scores will be added together to yield an overall score. Firstly, the factor weightage is calculated by using the coefficient of the factor (coefficient taken from Multiple Linear Regression because the relationship between other variables and the target variable must be related) against to sum of the factor coefficient and multiplying with 100, as shown in Figure 4.6.1.1. Figure 4.6.1.2 depicts all of the factor weights for each selected attribute.

$$Factor\ weight = \frac{factor\ coef}{sum(factor\ coef)} * 100$$

Figure 5.6.1.1: Formula to find the factor weight

	factor coef	sum of coef	factor weight
INTERCEPT	0.276356		
AMOUNT_OF_GIVEN_CREDIT	0.098659	0.677731	14.55725059
SEX	0.125822		18.56518294
MARITAL_STATUS	0.164219		24.2307051
AGE	0.041049		6.056827856
REPAYMENT_STATUS_IN_SEPTEMBER	0.195309		28.81807089
AMOUNT_OF_BILL_STATEMENT_IN_SEPTEMBER	0.016935		2.498779014
AMOUNT_OF_PREVIOUS_PAYMENT_IN_SEPTEMBER	0.035738		5.273183608

Figure 5.6.1.2: Factor weight for all selected attribute

The factor point is calculated using the factor weight's ceiling as Figure 4.6.1.3.

Factor	Point
AMOUNT_OF_BILL_STATEMENT_IN_SEPTEMBER	2
AMOUNT_OF_PREVIOUS_PAYMENT_IN_SEPTEMBER	5
AGE	6
AMOUNT_OF_GIVEN_CREDIT	15
SEX	19
MARITAL_STATUS	24
REPAYMENT_STATUS_IN_SEPTEMBER	29
Total	100

Figure 5.6.1.3: The factor weight's integer is used to determine the factor point.

As illustrated in Figure 4.61.3, knowing that the Amount of Bill Statement has the least influence on whether the credit card holder will default the following month and Repayment Status has the highest influence on the credit card holder's behaviour

5.6.2 Counting Score for each Binning of Factor

$$Score\ for\ each\ binning = \frac{the\ WoE\ of\ the\ binning - lowest\ WoE\ for\ each\ binning}{highest\ WoE\ for\ each\ binning - lowest\ WoE\ for\ each\ binning} * factor\ weight$$

Figure 5.6.2.1: Formula for calculate the score for each binning of Factor

Using the formula as Figure 4.6.2.1 to count the score for each binning of factor.

AMOUNT_OF_BILL_STATEMENT_IN_SEPTEMBER	WoE	Score
(-14386.001, 10915.333]	0.087271	2.00
(10915.333, 49614.333]	0.083587	1.97
(49614.333, 495559.0]	-0.18335	0.00
lowest	-0.18335	
highest	0.087271	

Figure 5.6.2.2: Score for each binning of Amount of Bill Statement in September

Given that the weight of the Amount of Bill Statement in September is 2 from Figure 4.6.1.3, the greatest binning score is 2 and the lowest binning score is 0. In subtitle 2.2, it is mentioned that binning must be linear, and in this section, it is more fully utilised and explained why binning must be linear. Knowing that the lesser the Amount of Bill Statement will have the greater the score in this factor. This can be explained by the fact that smaller bills are easier to pay.

AMOUNT_OF_PREVIOUS_PAYMENT_IN_SEPTEMBER	WoE	Score
(-0.001, 1362.0]	0.391268	5.00
(1362.0, 2000.0]	-0.02029	2.81
(2000.0, 3044.667]	-0.15044	2.12
(3044.667, 5100.0]	-0.25425	1.56
(5100.0, 10850.0]	-0.54791	0.00
lowest	-0.54791	
highest	0.391268	

Figure 5.6.2.3: Score for each binning of Amount of Previous Payment in September

Given that the weight of the Amount of Previous Payment in September is 2 from Figure 4.6.1.3, the greatest binning score is 5 and the lowest binning score is 0. The concept of the Amount of Previous Payment is the same as the Amount of Bill Statement, the less debt people have, the less pressure the credit card user is under and the easier it is to pay off the bills, and thus their creditworthiness and score are higher.

AGE	WoE	Score
(20.999, 26.0]	0.1178	4.62
(26.0, 29.0]	-0.13978	0.57
(29.0, 33.0]	-0.17588	0.00
(33.0, 38.0]	-0.05036	1.97
(38.0, 45.0]	-0.00162	2.74
(45.0, 75.0]	0.205613	6.00
lowest	-0.17588	
highest	0.205613	

Figure 5.6.2.4: Score for each binning of Age

Given that the weight of the Age is 6 in Figure 4.6.1.3, the greatest binning score is 6 and the lowest binning score is 0. Individuals between the ages of 21 and 26 may have completed their education, obtained work experience, and begun to establish themselves in many sectors of life. Depending on the context, this may signify a certain level of dependability or proficiency in various areas. Individuals between the ages of 45 and 75 may have longer-standing relationships with banks, lenders, and other financial institutions. These longstanding relationships can contribute to a sense of trust built over time, as they have proven their creditworthiness in earlier exchanges. Hence, the individuals between the ages of 21 and 26 and the individuals between the ages of 45 and 75 hold the higher score.

AMOUNT_OF_GIVEN_CREDIT	WoE	Score
(9999.999, 30000.0]	0.609956	15.00
(30000.0, 50000.0]	0.200745	10.39
(50000.0, 60000.0]	0.260723	11.06
(60000.0, 80000.0]	0.117839	9.45
(80000.0, 120000.0]	0.090512	9.15
(120000.0, 150000.0]	-0.16022	6.32
(150000.0, 200000.0]	-0.30381	4.70
(200000.0, 230000.0]	-0.35922	4.08
(230000.0, 320000.0]	-0.5005	2.48
(320000.0, 780000.0]	-0.72087	0.00
lowest	-0.72087	
highest	0.609956	

Figure 5.6.2.5: Score for each binning of Amount of Given Credit

Given that the weight of the Amount of Given Credit is 15 from the Figure 4.6.1.3, the greatest binning score is 15 and the lowest binning score is 0. The concept of the Amount of Given Credit is same as the Amount of Previous Payment and Amount of Bill Statement, the less debt people have, the less pressure the stress is under and the easier it is to pay off the bills, and thus their creditworthiness and score are higher.

SEX	WoE	Score
Male	-0.08309	0.00
Female	0.116317	19.00
lowest	-0.08309	
highest	0.116317	

Figure 5.6.2.6: Score for each binning of Sex

SEX	WoE	Score
Male	-0.083086	9.50
Female	0.116317	9.50
lowest	-0.083086	
highest	0.116317	

Figure 5.6.2.7: Score for each binning of Sex (Average)

Given that the weight of the Sex is 19 in Figure 4.6.1.3, the greatest binning score is 19 and the lowest binning score is 0. However, Illustration 3.2.11 shows that either male or female yield about the same default rate of 25.24% and 21.83%, respectively. Therefore, the sex binning score must be the average of the two, as shown in Figure 4.6.2.7.

MARITAL_STATUS	WoE	Score
Married	0.093583	16.40
Single	-0.08547	0.00
Others	0.176505	24.00
lowest	-0.08547	
highest	0.176505	

Figure 5.6.2.8: Score for each binning of Marital Status

MARITAL_STATUS	WoE	Score
Married	0.093583	8.00
Single	-0.085467	8.00
Others	0.176505	8.00
lowest	-0.085467	
highest	0.176505	

Figure 5.6.2.9: Score for each binning of Marital Status (Average)

Given that the weight of the Marital Status is 24 in Figure 4.6.1.3, the greatest binning score is 24 and the lowest binning score is 0. However, from Illustration 3.2.15, it appears that either married, single, or others yield about the same default rate of 24.78%, 21.82% and 26.97%, respectively. Therefore, the marital status binning score must be the average of the three, as shown in Figure 4.6.2.9.

REPAYMENT_STATUS_IN_SEPTEMBER	WoE	Score
Pay Duly	-0.62827	0.00
Payment Delay for 1 Month	0.518127	10.99
Payment Delay for 2 Month	1.972851	24.95
Payment Delay for 3 Month	2.31838	28.26
Payment Delay for 4 Month	1.876776	24.02
Payment Delay for 5 Month	1.229818	17.82
Payment Delay for 6 Month	1.325128	18.73
Payment Delay for 7 Month	2.395569	29.00
Payment Delay for 8 Month	1.46126	20.04
lowest	-0.62827	
highest	2.395569	

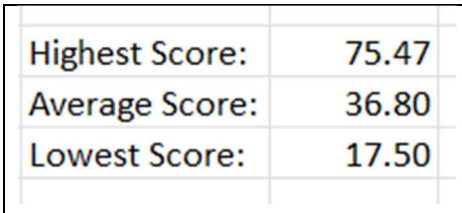
Figure 5.6.2.10: Score for each binning of Repayment Status in September

Given that the weight of the Repayment Status in September is 29 in Figure 4.6.1.3, the greatest binning score is 29 and the lowest binning score is 0. The linear binning is still mentioned above, but there are some circumstances where the linear form cannot be followed, such as when the original attribute is known as nominal data from the beginning. As a result, changing the trend is

impossible. Furthermore, Illustration 3.2.17 shows that the default rate only makes sense for timely payments with a four-month payment delay. Thus, these attributes will follow the score, but they can be discussed further with the stakeholder.

5.6.3 Scorecard

Subsections 4.6.1 and 4.6.2 determine the score for each factor as well as the binning. The scorecard can now be constructed. The scorecard is typically stated by the stakeholder; for example, the data scientist just provides the highest, average, and lowest score from historical information, and the stakeholder determines the lower and upper bounds of the score to assess whether this customer can be trusted.



Highest Score:	75.47
Average Score:	36.80
Lowest Score:	17.50

Figure 5.6.3.1: The highest, average and lowest score

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

Table 5.6.3.1: Conclusion for the accuracy and AUC for all the model

Models	Accuracy	AUC
Logistic Regression	0.802646	0.75
Gaussian Naïve Bayes	0.803328	0.75
K-Nearest Neighbours	0.801282	0.74
Decision Tree	0.802237	0.76
Extreme Gradient Boosting	0.795554	0.74

The model with the best accuracy in predicting the default credit card is Gaussian Naive Bayes, while the model with the highest AUC value is Decision Tree. However, there are no significant variations in the accuracy or AUC value of the model used in this research. All of the models have higher than 75% accuracy, implying that the specified characteristics can effectively predict the default credit card with greater than 75% accuracy. Then, utilising these features and the values of WoE, IV, and factor coefficient, the credit card company may prevent financial loss by following the scorecard.

Figure 4.6.1.1 depicts the scorecard built on factor weight and the calculation for factor weight. The scorecard can be built using Figures 4.6.1.3 to 4.6.2.10. Assume the credit card male single user of age 25 has NT 10200 in bill statements in September, NT 1500 in previous payments in September, and NT 20000 in given credit and pays on time. When all of the scores are summed up together, a credit card user can get 41.9 out of 100, which is higher than the dataset's average score. Then, follow the stakeholder's instructions to see if this user can be trusted.

6.2 Recommendations for future work

There are 25 variables, including ID and the target variable, the default payment next month. However, the final variables employed are eight, including the target variable.

The first suggestion is to use all of the variables that were dropped in this report and try to add them one by one to observe the value of R-Squared and the value of Adjusted R-Squared, as well as the model's accuracy. Aside from linear regression, several other statistical analyses, such as generalised linear models and nonparametric statistics, can be used for further analysis.

Considering our objective variable is the default payment next month, which only includes default and non-default, it serves as a classification and binomial issue because there are only two outcomes. Resampling, such as oversampling and undersampling, can be considered here to balance the binomial classification outcome. Besides, since the target variable is binomial, two machine learning techniques, including Support Vector Machines (SVM) and Neural Networks, are proposed for further analysis,

Other than that, the Sigmoid function for Logistic Regression model in this report is a straight line and the Sigmoid function should not produce a straight line. The issue is that the data preparation part of this report has some obstacles. However, Logistic Regression achieved an accuracy of around 80%. Instead of using financial terminology like WoE and IV, the Recursive Feature is one method that might be considered to solve this problem. Recursive Feature is an elimination that removes less important elements from the collection recursively based on their relevance.

On the other hand, the unsupervised learning model, such as clustering algorithms can be used to extracting more insights, patterns and relationships from the raw data. Unsupervised learning seeks to discover underlying structure or representations in data without the use of predetermined output labels.

REFERENCES

A.Y.J, A., & R, P. (2013). https://www.researchgate.net/profile/Arcadius-Akossou/publication/289526309_Impact_of_data_structure_on_the_estimators_R-square_and_adjusted_R-square_in_linear_regression/links/586a90ce08ae329d621114f2/Impact-of-data-structure-on-the-estimators-R-square-an. *20*(3), 10. https://www.researchgate.net/profile/Arcadius-Akossou/publication/289526309_Impact_of_data_structure_on_the_estimators_R-square_and_adjusted_R-square_in_linear_regression/links/586a90ce08ae329d621114f2/Impact-of-data-structure-on-the-estimators-R-square-an

Agarwal, A., Sharma, P., Alshehri, M., Mohamed, A. A., and Alfarraj, O. (2021). Classification model for accuracy and intrusion detection using machine learning approach. *PeerJ*, *7*, e437. https://doi.org/10.7717/peerj-cs.437

Alexopoulos, E. C. (2010). Introduction to multivariate regression analysis. *Hippokratia*.

Andrienko, G., Andrienko, N., Drucker, S. M., Fekete, J. D., Fisher, D., Idreos, S., Kraska, T., Li, G., Liu, K., MA, Mackinlay, J. D., Oulasvirta, A., Schreck, T., Schmann, H. J., Stonebraker, M., Auber, D., Bikakis, N., Chrysanthis, P. K., Papastefanatos, G., and Sharaf, M. A. (2020). Big Data Visualization and Analytics: Future Research Challenges and Emerging Applications. *HAL (Le Centre Pour La Communication Scientifique Directe)*. https://hal.inria.fr/hal-02568845/document

Bansal, M., Goyal, A., and Choudhary, A. (2022). A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning. *Decision Analytics Journal*, *3*, 100071. https://doi.org/10.1016/j.dajour.2022.100071

Barthès, J. A., Loezer, L., Enembreck, F., and Lanzuolo, R. (2020). Lessons learned from data stream classification applied to credit scoring. *Expert Systems With Applications*, *162*, 113899. https://doi.org/10.1016/j.eswa.2020.113899

Beynon-Davies, P. (2019). *Business Information Systems*. Bloomsbury Publishing.

Bhavsar, H., and Ganatra, A. (2012). A Comparative Study of Training Algorithms for Supervised Machine Learning. *International Journal of Soft Computing and Engineering (IJSCE)*, *2*(4). https://citeseerx.ist.psu.edu/document?repid=rep1andtype=pdfanddoi=18ca69ec35a0ab52922cb8a81d5041ac99005f3a

Chen, T., and Guestrin, C. (2016). XGBoost. *ArXiv (Cornell University)*. https://doi.org/10.1145/2939672.2939785

Cortez, P., and Embrechts, M. J. (2013b). Using sensitivity analysis and visualization techniques to open black box data mining models. *Information Sciences*, *225*, 1–17. https://doi.org/10.1016/j.ins.2012.10.039

Cramer, J. (2002). The Origins of Logistic Regression. *Social Science Research Network*. https://doi.org/10.2139/ssrn.360300

Davis, J. J., and Clark, A. G. (2011). Data preprocessing for anomaly based network intrusion detection: A review. *Computers and Security*, *30*(6–7), 353–375. https://doi.org/10.1016/j.cose.2011.05.008

Egger, R. (2022). *Applied Data Science in Tourism: Interdisciplinary Approaches, Methodologies, and Applications*. Springer Nature.

Ekelund, S. (2012). ROC Curves—What are They and How are They Used? *Point of Care: The Journal of Near-Patient Testing and Technology*, *11*(1), 16–21. https://doi.org/10.1097/poc.0b013e318246a642

Freund, R. J., Wilson, W. J., and Sa, P. (2006). *Regression Analysis*. Elsevier.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*(5). https://doi.org/10.1214/aos/1013203451

García, S., Luengo, J., and Herrera, F. (2016). Tutorial on practical tips of the most influential data preprocessing algorithms in data mining. *Knowledge Based Systems*, *98*, 1–29. https://doi.org/10.1016/j.knosys.2015.12.006

Gulati, P., Sharma, A., and Gupta, M. (2016). Theoretical Study of Decision Tree Algorithms to Identify Pivotal Factors for Performance Improvement: A Review. *International Journal of Computer Applications*, *141*(14), 19–25. https://doi.org/10.5120/ijca2016909926

Guo, X., Yin, Y., Dong, C., Yang, G., and Zhou, G. (2008). On the Class Imbalance Problem. *Fourth International Conference on Natural Computation*, 192–201. https://doi.org/10.1109/ICNC.2008.871.

Haomiao, Z., Deng, Z., Xia, Y., and Fu, M. (2016). A new sampling method in particle filter based on Pearson correlation coefficient. *Neurocomputing*, *216*, 208–215. https://doi.org/10.1016/j.neucom.2016.07.036

Hazra, A. (2017). Using the confidence interval confidently. *Journal of Thoracic Disease*, *9*(10), 4124–4129. https://doi.org/10.21037/jtd.2017.09.14

Islam, S. M. S., Wu, Q., Ahmadi, M., and Sid-Ahmed, M. A. (2010). Investigating the Performance of Naive- Bayes Classifiers and K- Nearest Neighbor Classifiers. *Journal of Convergence Information Technology*, *5*(2), 133–137. https://doi.org/10.4156/jcit.vol5.issue2.15

Jothi, N., Rashid, N. A., and Husain, W. (2015). Data Mining in Healthcare – A Review. *Procedia Computer Science*, *72*, 306–313. https://doi.org/10.1016/j.procs.2015.12.145

Kim, J. (2009). Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational Statistics and Data Analysis*, *53*(11), 3735–3745. https://doi.org/10.1016/j.csda.2009.04.009

Li, J., & Valliant, R. (2009). Component of Statistics Canada. *Survey Weighted Hat Matrix and Leverages*, 12-001–X.

Li, Y., Li, Y., and Li, Y. (2019). What factors are influencing credit card customer's default behavior in China? A study based on survival analysis. *Physica D: Nonlinear Phenomena*, *526*, 120861. https://doi.org/10.1016/j.physa.2019.04.097

Lommen, A. (2009). MetAlign: Interface-Driven, Versatile Metabolomics Tool for Hyphenated Full-Scan Mass Spectrometry Data Preprocessing. *Analytical Chemistry*, *81*(8), 3079–3086. https://doi.org/10.1021/ac900036d

Machado, M. A., and Karray, S. (2022). Assessing credit risk of commercial customers using hybrid machine learning algorithms. *Expert Systems With Applications*, *200*, 116889. https://doi.org/10.1016/j.eswa.2022.116889

Martinez, W. L., Martinez, A. R., and Solka, J. (2021). *Exploratory Data Analysis with MATLAB (Chapman and Hall/CRC Computer Science and Data Analysis)* (3rd ed.). Chapman and Hall.

Mishra, P., Biancolillo, A., Roger, J., Marini, F., and Rutledge, D. N. (2020). New data preprocessing trends based on ensemble of multiple preprocessing techniques. *Trends in Analytical Chemistry*, *132*, 116045. https://doi.org/10.1016/j.trac.2020.116045

Mowbray, F., Fox-Wasylyshyn, S. M., and El-Masri, M. M. (2019). Univariate Outliers: A Conceptual Overview for the Nurse Researcher. *Canadian Journal of Nursing Research Archive*, *51*(1), 31–37.

Myatt, G. J. (2007). Making Sense of Data. *A Practical Guide to Exploratory Data Analysis and Data Mining*. Wiley.

Ontivero-Ortega, M., Lage-Castellanos, A., Valente, G., Goebel, R., and Valdés-Sosa, M. (2017). Fast Gaussian Naïve Bayes for searchlight classification analysis. *NeuroImage*, *163*, 471–479. https://doi.org/10.1016/j.neuroimage.2017.09.001

Provost, F., and Fawcett, T. (2013). *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*. "O'Reilly Media, Inc."

Purchall, F., and Walker, R. S. (1972). *Case Studies in Business Data Processing*. Springer.

Ratner, B. (2009). The correlation coefficient: Its values range between +1/−1, or do they? *Journal of Targeting, Measurement and Analysis for Marketing*, *17*(2), 139–142. https://doi.org/10.1057/jt.2009.5

Řezáč, M. (2011). Advanced empirical estimate of information value for credit scoring models. *Acta Universitatis Agriculturae Et Silviculturae Mendelianae Brunensis*, *59*(2), 267–274. https://doi.org/10.11118/actaun201159020267

Safavian, S., and Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, *21*(3), 660–674. https://doi.org/10.1109/21.97458

Sharma, S., Aggarwal, A., and Choudhury, T. (2018). Breast Cancer Detection Using Machine Learning Algorithms. *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*. https://doi.org/10.1109/ctems.2018.8769187

Siddiqi, N. (2017). *Intelligent Credit Scoring: Building and Implementing Better Credit Risk Scorecards*. John Wiley and Sons.

Stoltzfus, J. (2011). Logistic Regression: A Brief Primer. *Academic Emergency Medicine*, *18*(10), 1099–1104. https://doi.org/10.1111/j.1553-2712.2011.01185.x

Tao, H., Habib, M., Aljarah, I., Faris, H., Afan, H. A., & Yaseen, Z. M. (2021). An intelligent evolutionary extreme gradient boosting algorithm development for modeling scour depths under submerged weir. *Information Sciences*, *570*, 172–184. https://doi.org/10.1016/j.ins.2021.04.063

Van Gool, J., Baesens, B., Sercu, P., and Verbeke, W. (2013). An analysis of the applicability of credit scoring for microfinance. Expert Systems with Applications, 40(15), 5988-5998. doi: 10.1016/j.eswa.2013.05.026

Warwick, J., and Mansfield, P. (2000). Credit card consumers: college students' knowledge and attitude. *Journal of Consumer Marketing*, *17*(7), 617–626. https://doi.org/10.1108/07363760010357813

Wood, D. E., and Piesse, J. (1988). The information value of failure predictions in credit assessment. Journal of Banking and Finance, 12(2), 275–292. https://doi.org/10.1016/0378-4266(88)90040-4

Yadav, S., and Shukla, S. (2016). Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification. *International Conference on Advanced Computing*. https://doi.org/10.1109/iacc.2016.25

Yap, B. W., Ong, S. H., and Husain, N. A. (2011). Using data mining to improve assessment of credit worthiness via credit scoring models. *Expert Systems With Applications*, *38*(10), 13274–13283. https://doi.org/10.1016/j.eswa.2011.04.147

Yeh, I. C., and Lien, C. H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems With Applications*, *36*(2), 2473–2480. https://doi.org/10.1016/j.eswa.2007.12.020\

Yu, D., Liu, Z., Su, C., Han, Y., Duan, X., Zhang, R., Liu, X., Yang, Y., and Xu, S. (2019). Copy number variation in plasma as a tool for lung cancer prediction using Extreme Gradient Boosting (XGBoost) classifier. *Thoracic Cancer*, *11*(1), 95–102. https://doi.org/10.1111/1759-7714.13204

Zdravevski, E., Lameski, P., and Kulakov, A. (2011). Weight of evidence as a tool for attribute transformation in the preprocessing stage of supervised learning algorithms. *The 2011 International Joint Conference on Neural Networks*. https://doi.org/10.1109/ijcnn.2011.6033219

APPENDICES

Appendix A: Python Code

Refer to FYP 2. ipynb

Appendix B: R – codes

```
# ========
# scatter plot for all the veriable
# ========
library("readxl")
setwd("C:/Users/hojk8/OneDrive/Desktop/Project I/data/")
data <- read_excel("2. file_want_column.xlsx")
plot(data, main = "Scatter Plot for All Variable")

# ========
# check is all the data inside the final clean file are all related to target variable
# ========
setwd("C:/Users/hojk8/OneDrive/Desktop/Project I/data/Manually/")
FeatureTranformingWoe<- read_excel("7. file_FeatureTranformingWoe.xlsx")
names(FeatureTranformingWoe)

model <- lm(FeatureTranformingWoe$"default payment next month" ~
FeatureTranformingWoe$"AMOUNT_OF_GIVEN_CREDIT" +
FeatureTranformingWoe$"SEX" +
FeatureTranformingWoe$"MARITAL_STATUS" +
FeatureTranformingWoe$"AGE" +
FeatureTranformingWoe$"REPAYMENT_STATUS_IN_SEPTEMBER" +
FeatureTranformingWoe$"AMOUNT_OF_BILL_STATEMENT_IN_SEPTE
MBER" +
FeatureTranformingWoe$"AMOUNT_OF_PREVIOUS_PAYMENT_IN_SEP
TEMBER")

summary(model)
```

```
confint(model, level=.95)
anova(model)

# =======
# table of R square, Adjusted R square
# =======
model.fit1 <- lm(formula = FeatureTranformingWoe$"default payment next month" ~ FeatureTranformingWoe$"AMOUNT_OF_GIVEN_CREDIT")
sum.fit1 <- summary(model.fit1)

model.fit2 <- lm(formula = FeatureTranformingWoe$"default payment next month" ~ FeatureTranformingWoe$"AMOUNT_OF_GIVEN_CREDIT" + FeatureTranformingWoe$"SEX")
sum.fit2 <- summary(model.fit2)

model.fit3 <- lm(formula = FeatureTranformingWoe$"default payment next month" ~ FeatureTranformingWoe$"AMOUNT_OF_GIVEN_CREDIT" + FeatureTranformingWoe$"SEX" + FeatureTranformingWoe$"MARITAL_STATUS")
sum.fit3 <- summary(model.fit3)

model.fit4 <- lm(formula = FeatureTranformingWoe$"default payment next month" ~ FeatureTranformingWoe$"AMOUNT_OF_GIVEN_CREDIT" + FeatureTranformingWoe$"SEX" + FeatureTranformingWoe$"MARITAL_STATUS" + FeatureTranformingWoe$"AGE")
sum.fit4 <- summary(model.fit4)

model.fit5 <- lm(formula = FeatureTranformingWoe$"default payment next month" ~ FeatureTranformingWoe$"AMOUNT_OF_GIVEN_CREDIT" + FeatureTranformingWoe$"SEX" + FeatureTranformingWoe$"MARITAL_STATUS" +
```

```
FeatureTranformingWoe$"AGE" +
FeatureTranformingWoe$"REPAYMENT_STATUS_IN_SEPTEMBER")
sum.fit5 <- summary(model.fit5)

model.fit6 <- lm(formula = FeatureTranformingWoe$"default payment next month" ~ FeatureTranformingWoe$"AMOUNT_OF_GIVEN_CREDIT" +
FeatureTranformingWoe$"SEX" +
FeatureTranformingWoe$"MARITAL_STATUS" +
FeatureTranformingWoe$"AGE" +
FeatureTranformingWoe$"REPAYMENT_STATUS_IN_SEPTEMBER" +
FeatureTranformingWoe$"AMOUNT_OF_BILL_STATEMENT_IN_SEPTEMBER")
sum.fit6 <- summary(model.fit6)

model.fit7 <- lm(formula = FeatureTranformingWoe$"default payment next month" ~ FeatureTranformingWoe$"AMOUNT_OF_GIVEN_CREDIT" +
FeatureTranformingWoe$"SEX" +
FeatureTranformingWoe$"MARITAL_STATUS" +
FeatureTranformingWoe$"AGE" +
FeatureTranformingWoe$"REPAYMENT_STATUS_IN_SEPTEMBER"+
FeatureTranformingWoe$"AMOUNT_OF_BILL_STATEMENT_IN_SEPTEMBER" +
FeatureTranformingWoe$"AMOUNT_OF_PREVIOUS_PAYMENT_IN_SEPTEMBER")
sum.fit7 <- summary(model.fit7)

R.sq.values <- data.frame(Model = c("AMOUNT_OF_GIVEN_CREDIT",
"AMOUNT_OF_GIVEN_CREDIT, SEX",
"AMOUNT_OF_GIVEN_CREDIT, SEX, MARITAL_STATUS",
"AMOUNT_OF_GIVEN_CREDIT, SEX, MARITAL_STATUS, AGE",
"AMOUNT_OF_GIVEN_CREDIT, SEX, MARITAL_STATUS, AGE, REPAYMENT_STATUS_IN_SEPTEMBER",
"AMOUNT_OF_GIVEN_CREDIT, SEX, MARITAL_STATUS, AGE,
```

```
REPAYMENT_STATUS_IN_SEPTEMBER,
AMOUNT_OF_BILL_STATEMENT_IN_SEPTEMBER",
"AMOUNT_OF_GIVEN_CREDIT, SEX, MARITAL_STATUS, AGE,
REPAYMENT_STATUS_IN_SEPTEMBER,
AMOUNT_OF_BILL_STATEMENT_IN_SEPTEMBER,
AMOUNT_OF_PREVIOUS_PAYMENT_IN_SEPTEMBER"),
                R.Square = c(sum.fit1$r.squared, sum.fit2$r.squared,
sum.fit3$r.squared, sum.fit4$r.squared, sum.fit5$r.squared, sum.fit6$r.squared,
sum.fit7$r.squared),
                Adjusted.R.Square = c(sum.fit1$adj.r.squared,
sum.fit2$adj.r.squared, sum.fit3$adj.r.squared, sum.fit4$adj.r.squared,
sum.fit5$adj.r.squared, sum.fit6$adj.r.squared, sum.fit7$adj.r.squared))
R.sq.values

# ======
# Diagnostic for Leverage and Influence
# ======
# fitted values, y hat
yhat <- round(model$fitted.values,4)

# residuals, ei
ei <- round(model$residuals,4)

# stardized residuals, di
di <- round(model$residuals/sum.fit$sigma,4)



# studentized residuals, ri
ri <- round(rstandard(model),4)

# PRESS residuals, e(i)
hii <- round(hatvalues(model),4)
press <- round(ei/(1-hii),4)
```

```
# R-student residuals, ti
ti <- round(rstudent(model),4)

diagnostic_table <- cbind(yhat, ei, di, hii, ri, press, ti)
diagnostic_df <- as.data.frame.matrix(diagnostic_table)
diagnostic_df

# put this table into excel file
library("writexl")
#           write_xlsx(diagnostic_df,"C:/Users/hojk8/OneDrive/Desktop/Project
II/dataset/Manually/10. diagnostic.xlsx")

n <- length(model$fitted.values)
cv1 <- qt(p=1-0.05/(2*n), df=model$df.residual-1)
cv1
cv2 <- qt(p=1-0.05/(2*n), df=model$df.residual)
cv2

# plot ri vs yhat
plot(x=model$fitted.values, y=ri, xlab="Estimated Mean Square",
ylab="Studentized Residuals", main="r[i] vs estimated mean response"
    , panel.first = grid(col="gray", lty="dotted"), ylim=c(min(qt(p=0.10/(2*n),
df=model$df.residual), min(ri)), max(qt(p=1-0.10/(2*n)
    , df=model$df.residual), max(ri))))
abline(h=0, col="darkgreen")
abline(h=c(qt(p=0.05/(2*n), df=model$df.residual), qt(p=1-0.05/(2*n),
df=model$df.residual)), col="darkred", lwd=2)

# plot ti vs yhat
plot(x=model$fitted.values, y=ti, xlab="Estimated Mean Square", ylab="R-
Student Residuals", main="t[i] vs estimated mean response"
```

```
    , panel.first = grid(col="gray", lty="dotted"), ylim=c(min(qt(p=0.10/(2*n),
df=model$df.residual-1), min(ti)), max(qt(p=1-0.10/(2*n)
    , df=model$df.residual-1), max(ti))))
abline(h=0, col="darkgreen")
abline(h=c(qt(p=0.05/(2*n), df=model$df.residual-1), qt(p=1-0.05/(2*n),
df=model$df.residual-1)), col="darkred", lwd=2)

# =====
# Influential Cases
# =====
dffits.i <- round(dffits(model),4)
cook.i <- round(cooks.distance(model),4)
dfbeta.all <- round(dfbetas(model),4)

influential_table <- cbind(dffits.i, cook.i, dfbeta.all)
influential_df <- as.data.frame.matrix(influential_table)
influential_df

# put this table into excel file
library("writexl")
#write_xlsx(influential_df,"C:/Users/hojk8/OneDrive/Desktop/Project
II/dataset/Manually/11. influential.xlsx")

# critical value for cook's distance
# little influence
# major influence
cook_little <- qf(p=0.2, df1=8, df2=24429, lower.tail=FALSE)
cook_little
cook_major <- qf(p=0.5, df1=8, df2=24429, lower.tail=FALSE)
cook_major
```

2/sqrt(24437)