3. RESEARCH METHOD

3.1 Research Step

The "Default of Credit Card Clients Dataset" (Yeh, 2016) is a dataset that offers sources of data regarding the payment history, demographics, and behaviors of credit card holders. The main goal is to use the dataset to build and assess models that forecast the likelihood of customer default on their payments in the following month. This analysis is crucial for banks and financial institutions to minimize costly defaults, thereby ensuring a more stable and secure financial environment. This comprehensive comparison will help to determine the strategy for credit card default prediction problems and contribute risk management in the financial industry.

Research method shows the research's steps to be conducted in a flowchart on figure 3.1. This flowchart aims to ensure that the research runs clearly and in a consequence structure, and also provides clear steps for the readers.

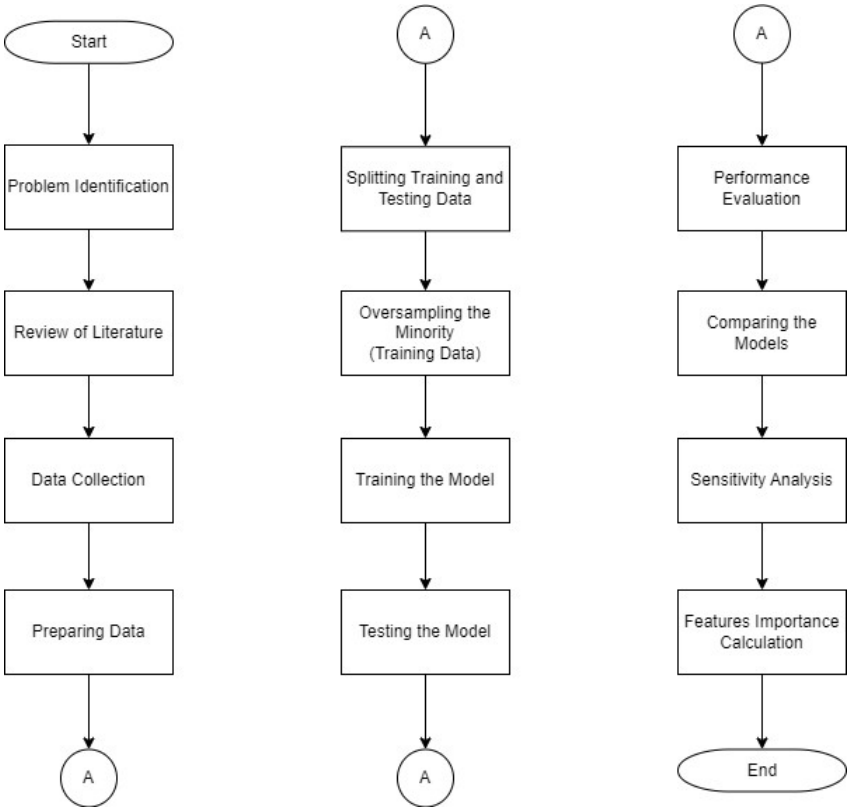

Figure 3.1 Research Method Flowchart

3.2 Problem Identification

Problem identification is the first stage of the research to determine the main objective of this research. A well-defined problem will guide the researcher to understand the main problem that needs to be accomplished based on the identification.

3.3 Review of Literature

At this stage, the theories and literature related to the research will be explored further. It helps the researcher to identify the precise aspects of the problem. Related theories can be gained from books, journals, proceedings, theses, and previous research about credit card default, machine learning, machine learning algorithms. The study from previous research can provide its conclusion and how these studies were conducted.

3.4 Data Collection

In the data gathering step, the data set “Default of Credit Card Clients Dataset” will be downloaded from Kaggle. The information that includes in this data set are default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005.

Dataset link source :

https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset

3.5 Preparing Data

At this stage, putting together the collected data and cleaning the data to remove unrequired data, missing values, data type conversion, etc. It could potentially restructure the dataset, the features and target in the dataset will be split into X (features) and y (target). It is important to separate the target from features in order to facilitate the development, training, and effectiveness of the model. Before splitting the data, there is one stage, which is standardization. Standardization is performed for numerical data. Then, the data will be split into training data and testing data.

3.6 Splitting Training and Testing Data

The data will be divided into two sets with an 80 : 20 ratio, where 80% is allocated to the training data set and the remaining 20% belong to the testing data set. The training set is the data which the model learns patterns and gains insight for building the model. The

hyperparameter which is used to optimize the model in the next step is also determined using training data. Whereas, testing data is used to assess the model's performance and to verify its performance. The next step after the model has been trained on training data, the testing data will apply to the model to evaluate how well it generalizes to unseen data. This separation into training and testing data is essential to ensure that the model's performance can be effectively assessed and fine-tuned if necessary.

3.7 Oversampling the Minority

One technique for addressing class imbalance in machine learning involves oversampling the minority class. Class imbalance arises when one class has significantly fewer samples than another class within the dataset. In order to achieve balance between the minority and majority classes, oversampling is used to increase the number of samples in the minority class. It is important to note that the oversampling technique is only applied to the training data, and in this study, the selected oversampling method is SMOTE-NC (Synthetic Minority Over-sampling Technique for Nominal and Continuous data). This decision of using this method is because the dataset comprises both categorical and numerical data, making SMOTE-NC a suitable option for generating synthetic samples and enhancing the model's performance.

3.8 Training the Model

After the class in the training data set is balanced, the training data set will be fit into the machine learning model to find the patterns and make predictions. In order to make a good performance, hyperparameters need to be adjusted. The combination of hyperparameters optimized the model's performance by producing better results with fewer errors. The hyperparameters should be adjusted before the learning process begins. Additionally, for finding the right hyperparameters in order to build the good model and avoid the overfitting, there will be an extra step which is cross validation. The cross validation technique that used in this study is K-Fold Cross Validation using 5 folds.

Each fold may have different hyperparameters, and different sets of hyperparameters can be applied to the model to identify the best hyperparameters among the 5 folds. The decision for choosing which hyperparameters suit for building the model classifier depends on the priority of performance evaluation. The performance evaluation that will be considered the most is F1-Score and G-Mean since the data is imbalanced.

3.9 Testing The Model

After the training phase, it is important to check its performance by testing the model on testing data (the unseen data that has been split earlier). This involves conducting comprehensive testing using a separate set of data known as the testing data. In evaluating the model's performance against this unseen dataset, one can determine whether the model can generalize and make accurate predictions on new, previously unknown information. This testing step ensures the reliability and effectiveness of the trained model in handling the data set.

3.10 Performance Evaluation

After the performance of machine learning can be assessed, the machine learning performance measurements such as accuracy, precision, recall, F1-score, AUC, G-Mean are considered as a benchmark for machine learning models. The performance measurement is used to see how the model generalizes and makes predictions. Each performance evaluation metric plays a distinct role, and the selection of which performance measurement will be considered the most depends on which one best aligns with the specific task or problem at hand.

3.11 Comparing The Models

Based on machine learning performance measurements, it can be compared to evaluate which algorithm among single classifiers (Naive Bayes, Logistic Regression, Decision Tree) and ensemble learning (AdaBoost, XGBoost, Random Forest) have the best performance. At this stage, it can be determined which parameter is given the most consideration. The chosen parameter depends on our goals and the characteristics of the dataset. The model with the best performance can finally be used to make predictions.

3.12 Sensitivity Analysis

Sensitivity analysis is a stage conducted to examine the impact of SMOTE-NC across various imbalance data ratios, 1:4 (the original dataset's imbalance ratio), 1:10, 1:20, and 1:50. The initial step involves undersampling the minority class to achieve the specified imbalance ratio. Following that, the undersampled minority class will undergo oversampling using the SMOTE-NC method. Thus, the sensitivity analysis will compare the performance of models built with undersampled data and the newly oversampled SMOTE-NC data.

3.13 Features Importance

After comparing the model performance in several imbalance ratios, the features' importance is the final stage of this research. In this stage, the impact of each feature on various models and on each fold of cross-validation results will be examined. Every fold is examined, since the results of feature importance may vary for each fold. Then, the features will be ranked from the largest to the smallest percentage of importance in each fold. The feature rankings from each fold will then be averaged and sorted from smallest to largest. Following that, this ranking order will be compared between models.

In determining feature importance, five commonly used algorithms employ distinct methodologies. Decision Trees assess importance by measuring how frequently a feature is used for node splits and the reduction in impurity achieved. As part of an ensemble, Random Forest combines the contributions from each individual decision tree. The AdaBoost algorithm emphasizes features prone to misclassification across iterations by assigning different weights to instances across iterations. XGBoost derives feature importance from the involvement of features in splitting decisions. On the other hand, Logistic Regression quantifies importance through the absolute values of coefficients assigned to each feature.