Contents

1 Methodology

1.1 Research Approach

This study is conducted to compare two cost-sensitive approaches using three machine learning models. The framework proposed has four phases:

1. Data Pre-processing
2. Resampling data
3. Fitting the model
4. Model Evaluation

The detailed flow is shown as follows:

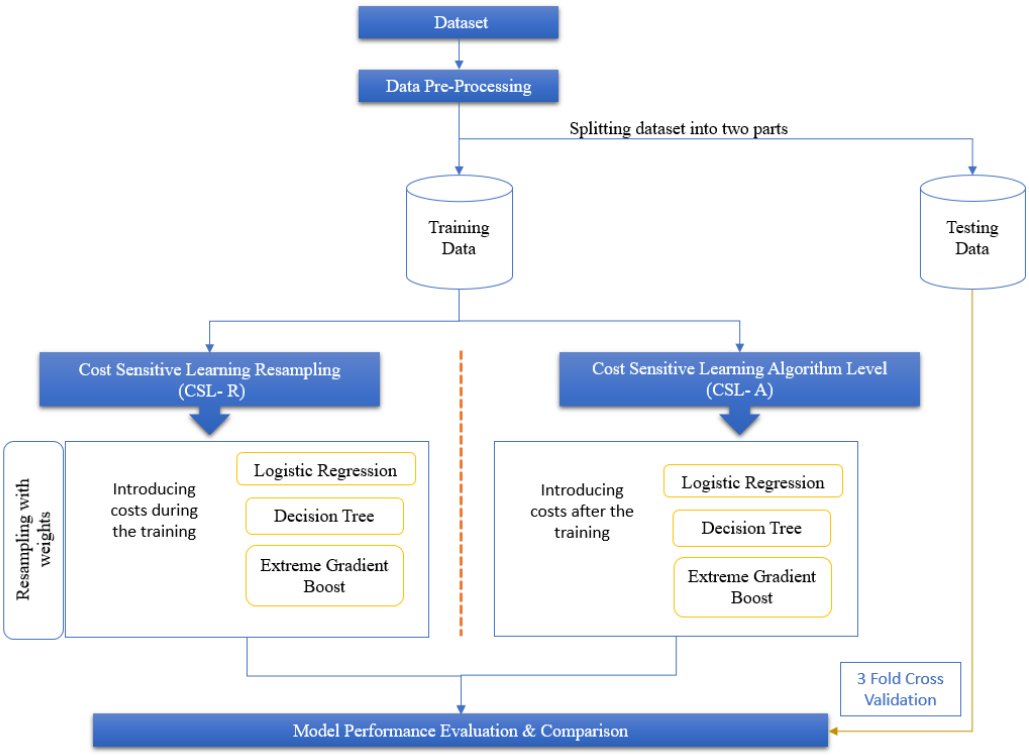

Figure 1: Research Approach

Under data pre-processing, the dataset is prepared for ML modelling. It includes selection of variables (also known as feature selection), data preparation including handling of missing values & dummy variable creation. The processed dataset is then split into two parts – training dataset for model learning purpose and testing dataset for model validation purpose. Thereafter, the training dataset is used to train models for both CSL methods:

Method 1: Cost Sensitive Learning with Resampling (CSL-R): In this method, costs are minimized by giving higher importance to less costly class during the model training phase. This is accomplished by adding class weights to rebalance the cost associated with classes. 'Weighted cost-sensitive wrapper' is introduced during the model training.

Method 2: Cost Sensitive learning at Algorithm level (CSL-A): Misclassification costs are introduced within the model algorithm after model training. This is carried out by adjusting the threshold values of the model's decision boundary, without affecting the learning process.

In comparing both the methods, this study aims to find effective method to minimize misclassification costs in imbalanced loan datasets. This will help in decreasing predictive bias in imbalanced datasets. This research intends to achieve the following objectives:

1. To compare performance of CSL techniques (CSL – R & CSL – A) in understanding its effectiveness in reducing misclassification costs.
2. To find the best CSL model in predicting default & decreasing risk exposure.

To achieve the first objective, three ML models, namely logistic regression, decision tree & extreme gradient boosting, are chosen based on the literature available on cost sensitive models and its effectiveness in handling misclassification errors. These three models will be trained on both techniques. Their performance will be evaluated based on calculating total costs and average misclassification cost. The least cost model with overall higher performance will be the best model.

The second objective will be to find the best performing model among the six ML models trained, i.e. CSL – R Logistic regression, CSL – R Decision tree, CSL – R XG-Boost, CSL – A logistic regression, CSL – A decision tree, CSL – A XG-Boost. Its real-life applicability such as the model's capability to minimize the credit risk exposure of an organisation will determine its effectiveness. In this case, the focus will be to minimize probability of default.

1.2 Machine Learning Models

There are various techniques are available in ML modelling, each categorized by distinctive advantages and limitations. Despite substantial innovations, certain aspects of credit risk modelling utilizing sophisticated data sampling techniques remain underexplored (Barbaglia

et al., 2023). As per the literature, research on the application of machine learning in credit risk management has predominantly focused on model's inherent limitations, such as challenges in handling data imbalances, assumptions, and biases (Shi et al., 2022). Literature review presents that regression and tree models exhibit superior performance in addressing data imbalance issues in loan prediction as compared to other ML classifier models (Shen, 2021; Pławiak, 2019).

This study uses three models:

1. **Logistic Regression**

Logistic regression is a one of the popular algorithms for binary classification tasks (Abdou et al., 2016). The algorithm forms a linear function between the probability of an event occurring—such as probability of a loan default in this study—and the predictor variables available in the dataset (Hosmer, Lemeshow, & Sturdivant, 2013). The mathematical representation of the function is shown below:

$$logit(p) = ln\left(\frac{p}{1-p}\right) = \beta 0 + \beta 1X1 + \beta 2X2 + \cdots + \beta nXn \quad (2)$$

where,

p is probability of the event,

$\frac{p}{1-p}$ is the probability of event happening,

$\beta 0$ is the intercept and $\beta 1, \beta 2, \ldots \beta n$ are coefficients, and

$X1, X2, \ldots Xn$ are predictor variables in the dataset.

The logit function in the equation ensures a linear relationship on its predictor variables. The output remains in the range of 0 and 1, which efficient for a binary predictor tasks.

2. **Decision Tree**

One of the most adaptable algorithms is decision tree. It is useful in both binary & multi classification tasks and regression problems (Madane & Nande, 2019). The algorithm visualisation is presented in figure 2. Each nodes are leaf-life arrangements that split

variables. Its shape is similar to a downward facing tree structure.

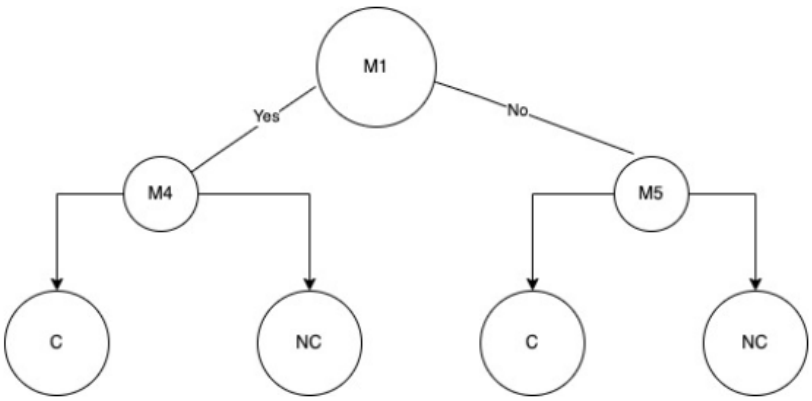

Figure 2: Sample Decision Tree Structure

Source: https://en.wikipedia.org/wiki/en:Creative_Commons

3. Extreme Gradient Boosting (XG-Boost)

Chen and Guestrin (2016) built XG-Boost algorithm based on the gradient boosting ensemble decision tree model. This algorithm creates an ensemble of decision trees, each tree corrects errors made by previous trees. Unlike other algorithms, XG-Boost utilises parallel processing to make the model learning process faster and efficient. It is therefore scalable for big data (Chen and Guestrin, 2016). Its parameters such as tree depth can be tuned and optimised for specific requirements (Chen and Guestrin, 2016).

1.3 Cost Matrix

Studying and understanding misclassification costs is crucial to achieve research objectives. The misclassification errors in predictive modelling such as false positives and false negatives are associated with certain costs in cost sensitive learning techniques. In order to increase model's interpretability in predicting correct outcomes, these misclassification errors must be minimized. Charles Elkan (2001) in his publication '*The foundations of cost-sensitive learning'* introduces a cost matrix.

Table 1: Cost Matrix by Charles Elkan (Elkan, 2001)

	Actual Negative	**Actual Positive**
Predict Negative	C (0,0)	C (0,1)
Predict Positive	C (1,0)	C (1,1)

The costs associated with correct predictions (C (0,0) and C (1,1)) are assigned 0. They don't incur any real cost in predicting correct outcomes. Although, some studies include a minimal costs for implementation of the model (Mienye I & Sun Y, 2021). In this study, the costs are kept zero as they don't incur affect model predictions and model objectives.

C (0,1) and C (1,0) positions in the matrix are the misclassification errors. In default prediction, lending organisations lose money when loan defaults. Hence, misclassifying default increases high debt for the organisation. Yanka A and Mariya A (2022) calculated the misclassification costs for defaults in the Lending Club dataset. Cost for misclassifying defaults, i.e. C(1,0), is 5957.99 USD and costs for wrong predictions (i.e. C (0,1)) is 2615.14 USD.

Table 2: Cost Matrix from The Lending Club (Yanka and Mariya, 2022)

	Actual Negative	**Actual Positive**
Predict Negative	0	2615.14
Predict Positive	5957.99	0

The cost matrix is thereby adapted in this study for cost sensitive learning comparison.

1.4 Model Validation & Evaluation

An important aspect in machine learning model development is assessing the model's performance and its effectiveness. This is done by model validation and assessing model's outcome with certain evaluation metrics.

These metrics are:

1. **Confusion Matrix**

   A confusion matrix is performance evaluation tool for machine learning models. It is represented in a tabular format as shown in table 3. It helps breakdown model's prediction into four components:

   1. True Positive (TP): Positive class predicted correctly
   2. False Positive (FP): Negative class incorrectly predicted to be positive
   3. True Negative (TN): Negative class predicted correctly
   4. False Negative (FN): Positive class incorrectly predicted to be negative

   Table 3: Example of a Confusion Matrix

	Actual Positive	**Actual Negative**
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

2. **Accuracy**

   Accuracy (A) is ratio of correctly predicted instances of the model.

$$Accuracy = \frac{TP + TN}{TN + FN + TP + FP} \quad (2)$$

3. Kappa metric

Although in imbalanced datasets, accuracy may not be ideal metric in understanding true model performance. Kappa metric (also called Cohen's Kappa) offers a better judgement of model's performance. This is because the metric, unlike the accuracy, is not inflated by the majority class. Kappa metric measures agreement level between actual and predicted observations.

$$kappa = \frac{2 * (TP * TN - FN * FP)}{(TP + FP) * (FP + TN) + (TP + FN) * (FN + TN)} \quad (3)$$

A higher value of kappa suggests a higher level of agreement, meaning a better model performance in handling imbalanced data. A higher accuracy value with lower kappa values may suggest biasness due to class imbalances. It indicates the model is predicting majority class (i.e. non-defaults) majority of the time while failing to capture patterns of the minority class (i.e. defaults), which is of higher importance in this case. Therefore, kappa metric is preferred over accuracy to check model's robustness against class imbalances (Boughorbel, S et al., 2017).

4. G-mean

Geometric mean (G-mean) is another measure for imbalanced dataset for a comprehensive model evaluation (Shen et al., 2019). Higher value of G-mean indicates a good balance between the two classes (i.e defaults and non-defaults). Its mathematical equation is:

$$G - mean = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}} \quad (4)$$

5. Precision

Precision (P) is a performance metric. It is the ratio of correctly predicted positive class to the total number of positive predicted. It is also known as '*Positive Predicted Value'* (PPV) and is calculated as:

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

6. Recall

Recall (R) is ratio of correctly predicted positive class to the total number of positives. It is also known as '*True Positive Rate*' (TPR) and '*sensitivity'*.

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

7. **F1 Score**

F-1 Score is harmonic mean of precision and recall. It shows the number of occurrences that are incorrectly predicted. F1 score is a useful in data imbalance.

$$F1 = 2\left(\frac{P * R}{P + R}\right) \tag{7}$$

8. **Area under curve (AUC) based on ROC plot**

The area under the Receiver Operating Characteristics Curve (ROC) is one of the popular metrics that measure the model's ability to distinguish between positive and negative classes. It is a two-dimensional graph of true positive rate and false positive rate. The curve closer to the top left corner is a good performing model. A sample ROC curve is shown in Fig 3:

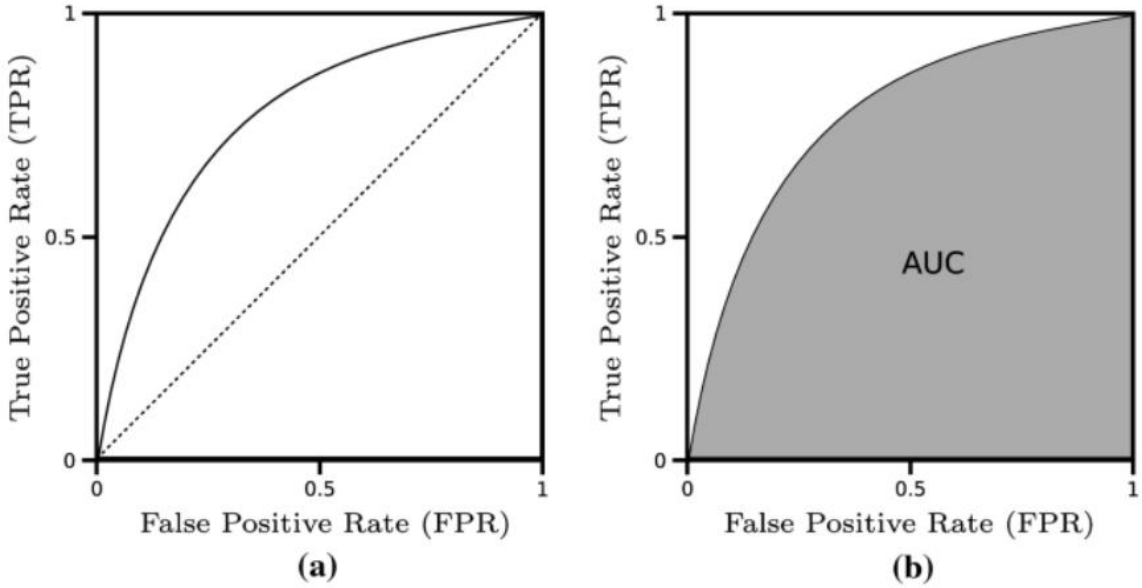

Figure 3: Example of a ROC curve and AUC representation

Source: (Jaskowiak et al., 2022)

9. Total costs

Utilizing the confusion matrix outputs and the defined cost matrix in Table 1, total misclassification costs are calculated to find the model with lowest misclassification costs.

$$TC = (TP \times Cost\ of\ TP) + (FP \times Cost\ of\ FP) + (TP \times Cost\ of\ TP) + (FN \times Cost\ of\ FN) \quad (8)$$

The cost of TP and cost of TN is set as 0 in the cost matrix in Table 1.

$$TC = 0 + (FP \times Cost\ of\ FP) + 0 + (FN \times Cost\ of\ FN) \quad (9)$$

Hence, the total cost is represented by:

$$TC = (FP \times Cost\ of\ FP) + (FN \times Cost\ of\ FN) \quad (10)$$

Understanding the total costs is one of the key metrics to interpret how the model is predicting errors (i.e. the false negatives and false positives). Thus, this metric helps in ensuring the model aligns with the business's operational objectives of minimizing expenditure and maximizing revenue stream. Hence, the model with least total costs will be an ideal model with least number of misclassification errors.

10. Average misclassification cost

The 'mlr' package (Bischl et al., 2016) has a custom '*makeCostMeasure*' function calculates average misclassification cost based on the cost matrix. Such a custom cost measure is created based on the misclassification costs from the cost matrix in table 1. Penalty is imposed when the prediction is incorrect. The minimum penalty is zero for correct predictions and 5957.99 when predictions are incorrect. In summary, if the model has higher false negatives and false positives then the cost function penalises and therefore will have a higher average misclassification cost. The higher the value, higher the misclassification errors in the model.

11. Mean misclassification error (mmce)

Mean misclassification error ('*mmce*') (He & Gracia, 2009) is a predefined metric for cost sensitive classification in '*mlr'* package (Bischl et al., 2016) in R Programming Language (R core team, 2023). It measures the mean score of the misclassification errors. It is the proportion of incorrect classification errors to total instances. Lower value of mmce suggests lower misclassification errors, indicating it has higher proportion of correct predictions (He & Gracia, 2009).