Table of Contents

1 Data Modelling

1.1 Feature Selection

Selection of the predictor variables with the target variable loan defaults is crucial step in machine learning modelling. Carefully each predictor variable has to be selected with reasoning and correlation with the target variable. These variables are broadly divided into two categories:

1. **Borrower's information**: Its annual income, employment history, credit history, credit score, past delinquency history, credit lines available, home ownership status, total revolving credit balance, etc.
2. **Loan related information**: Information related to the borrower's loan such as current loan status, loan amount, loan duration, loan grade, debt to income ratio, purpose, meets credit policy, instalment amount, etc.

The variables falling under both the categories were selected for modelling. These 23 variable description is given in Table 6:

Table 1: List of Variables in the dataset

Variables	Description
Annual Income	Borrower's income
Debt to Income Ratio	Proportion of Borrower's monthly debt payments and his/her income
Employee length	Borrower's employment length: from less than 1 year, 2 years, 3 years… 10 years and more than 10 years
Delinquent within 2 years	Number of times borrower is 30 Days Past Due in his/her credit history

FICO range	Borrower's credit score, values ranging between 300 to 850. (Higher the score, higher credit worthiness of the individual)
Open Account	The number of open credit lines with the borrower at the time of registration
Total Account	Total number of credit lines available the borrower at the time of registration
Revolving Balance	Borrower's total credit revolving balance in all his/her credit accounts
Revolving Utilization	The proportion of utilization of borrower's revolving balance. The amount of credit utilized to the total amount available in his/her credit accounts
Home Ownership Status	Borrower's ownership status: rented, owned by individual, or mortgage
Public Record	Number of derogatory public records of the borrower at the time of registration
Public Record Bankruptcy	Number of bankruptcy public records of the borrower at the time of registration
Interest Rate	Interest rate charged on the loan amount
Loan Amount	The total loan amount credited
Loan Status	Status assigned based on the borrower's delinquency: current, delinquent or paid off
Instalment	The instalment amount owed by the borrower on monthly basis
Outstanding Principal	Outstanding principal remaining from the total loan amount
Last FICO low	The lower range of credit score recorded at the last instalment date
Purpose	The reason for borrowing loan: purchasing vehicle, credit card, educational, home improvement, medical, business, mortgage, etc.
Term	The length of the loan borrowed.
Verification Status	Indicates if the borrower's annual income was verified or not by the agency.
Meets Credit Policy	Indicates if the borrower's application met the internal credit policy or not by the agency.
Grade	A grade is assigned based on borrower's credit profile. The grade ranges from A to G – A being the best lowest risk grade to G having the highest risk.

1.2 Data Preparation

Examining the given dataset, there are missing values present in various variables. ML models trained on missing data may lead to learning incorrect patterns, resulting in poor generalisation of the dataset. The three models used in this dissertation are inherently capable of handling missing data. Although these models can handle missing values, during model cross-validation

could show inconsistencies. Each fold during cross validation process may display inconsistent results due to missing values at each fold. In order to ensure there is no data leakage and model can accurately perform cross-validation, it is necessary to address missing values.

There are many strategies available in ML modelling to handle null values. Imputation is one of the popular strategies for handling missing data (Hastie R et al., 2001). In imputation, missing values are replaced with the mean, median, or mode of the variable. Since the variables in the data are not distributed randomly, it might introduce bias. Thereby, missing values are totally removed from the final dataset to increase data consistency. Having no null values will help ensure effectiveness in comparison of different ML models. Additionally, models trained on the same dataset without any null values increases model's interpretability and comparability.

It is also worth noting, the dataset consists a high number of instances, which may lead to high computational time. Therefore, removing instances with null values present will help increase computational efficiency in learning and validation processes. 28% of rows containing null values were disregarded. Hence, the final dataset contains only 30,461 instances.

For machine learning modelling, categorical variables are required to be converted into dummy variables. Hot-encoding, binning or combination of both are popular techniques used in creating dummy variables in the dataset.

Categorical data such as loan grades, purpose, loan status, etc have been transformed into a binary format: 0 or 1. "1" indicates presence of the variables. Such binary formats 0 or 1 help ML models to form relationships with existing numerical variables. The table below displays the dummy variables created for this dissertation:

Table 1: List of dummy variables created for the analysis

Categorical Variable Name	Dummy variable name
Employment length	1. Less than 1 2. 1 to 5 3. 5 to 10 4. > 10 years
Loan Purpose	1. Car 2. Home 3. Personal

	4. Business 5. Credit card 6. Debt consolidation 7. Others
Home ownership	1. Owned 2. Mortgage 3. Rent
Verification Status	Verfication = yes ('Y')
Meets Credit Policy	Meets credit policy = yes ('Y')
Loan grade	1. Grade A (best) 2. Grade B 3. Grade C 4. Grade D 5. Grade E 6. Grade F 7. Grade G (worst)

The target variable "loan default" is also converted into a dummy variable. "1" is assigned against the defaulted borrower and "0" when non-defaulted.

Binning is a method that is used to transformed continuous numerical variables into categorical attribute (Fayyad and Irani, 1993). Binning has several benefits. It helps increase interpretability of the model output (Kuhn and Johnson, 2013). The list of variables are listed below:

Table 3: List of variables that were used for binning

Continuous Numerical Variable	**Binning variable**
Employment length	Bin 1: Less than 1 Bin 2: 1 to 5 Bin 3: 5 to 10 Bin 4: > 10 years
Loan term length	1. 36 months 2. 60 months

Data normalisation and scaling is commonly recommended when dummy variables exist in the data model. For decision tree, centering is not compulsory. Therefore, a sensible option is to leave the data in its originality.

1.3 Training and Testing Set Split

In order to evaluate ML models, it is one of the common techniques to split dataset into two parts: training set and testing set. For the purpose of this study, the dataset is split in a 70%-30% ratio. 70% of the data is used for model learning and rest 30% test set is utilised in model performance evaluation. The final dataset consists of 30,461 observations. Training dataset has the first 21,323 observations, while testing dataset has the remaining 9,138 observations. For comparison of models, all models were trained on the same training dataset and tested on the same testing dataset. This was achieved by fixing the seed (i.e. a pseudo-random number 12345678) before the data is split into training and testing sets respectively.

1.4 Balancing the Dataset

In order to avoid bias and over fitting the model, handling imbalance is crucial in ML modelling. During the training, balanced classes help generalise relationships to enhance its robustness. When classes are fairly distributed, it also decreases misclassification error.

One of the common methods is measuring 'imbalance ratio'(IR). It is the ratio of the minority to majority class in the dataset. The highter the value of IR, the higher is the class imbalance. It is represented as:

$$IR = \frac{X_1}{X_0} \tag{11}$$

Where $X_1 =$ number of majority class observations and $X_0 =$ number of minority class observations.

Fadi et al. (2020) studied IR's usefulness in handling class imbalance with various datasets. In the given dataset, the majority class (i.e. non-defaults) has 26,012 (85.39%) instances, while minority class (i.e. defaults) cases are only 4,449 (14.60%) instances. Hence, IR is 5.8.

R Studio (Posit team, 2023) supports class weights within model algorithms in '*mlr'* package (Bischl et al., 2016). IR is used in the '*weights'* or '*class_weights*' parameter in baseline classifiers.

1.5 Model Optimisation

Hyper parameter tuning is performed in ML modelling to optimise the performance of the model by finding ideal sets of hyper parameters. Model parameters are learned from the data, but hyper parameters are set before training. Hyper parameters therefore directly influences model's learning capabilities. Finding optimal hyperparameters can help increase model performance and decrease overfitting.

For this study, a three fold cross validation is set for resampling method and average credit costs measure is used to find the ideal parameters of the model. The following tables shows in detail the parameters tuned into the model:

Table 4: List of hyperparamters tuning for each ML model

ML Model	Hyper-parameters tuned	Software packages used in R Studio
Decision Tree	complexity parameter" (cp) = {0.0041}	"rpart" Package (Therneau and Atkinson, 2022)
XG-Boost	• "booster" = {"gbtree"} • "max_depth" = {5} • "min_child_weight" = {6.22} • "subsample" = {0.541} • "colsample_bytree" = {0.881}	"xgboost" package (Chen et al.,2024)

Model Optimisation in CSL-R approach

Using "mlr" package (Bischl et al., 2016), model learners are transformed into "*makeWeightedClassesWrapper*". Imbalanced ratio (IR) is used as a sample weight for reference for tuning weights in the models. Each model has a unique tuned weight to handle data imbalance. The tuned weights are as follows:

- *Logistic regression*: $w = 3.5$
- *Decision tree*: $w = 2.5$
- *XG-Boost*: $w = 5$

Model Optimisation in CSL-A approach

Threshold method in CSL-A approach is adjusting decision threshold used in categorizing instances within the classes (default and non-default). It is by default set at 0.5, meaning it gives equal weightage to both classes. By finding the optimal threshold value, it minimizes cost function to reduce misclassification errors in the model output.

There are two methods in determining optimal threshold value – Theoretical threshold & empirical threshold.

The first method, theoretical threshold, calculates optimal threshold values with the help of cost matrix.

$$th = \frac{C(0,1)}{\left(C(0,1) + C(1,0)\right)} \tag{12}$$

Referring to the cost matrix in Table 1, the value of theoretical threshold is $th = 0.30$.

The second method, empirical threshold, utilises the theoretical threshold value as a base to perform a three-fold cross validation and plots a threshold vs. performance plot. The minimum average misclassification cost will be achieved at the ideal threshold value. The performance plots for each models are shown below:

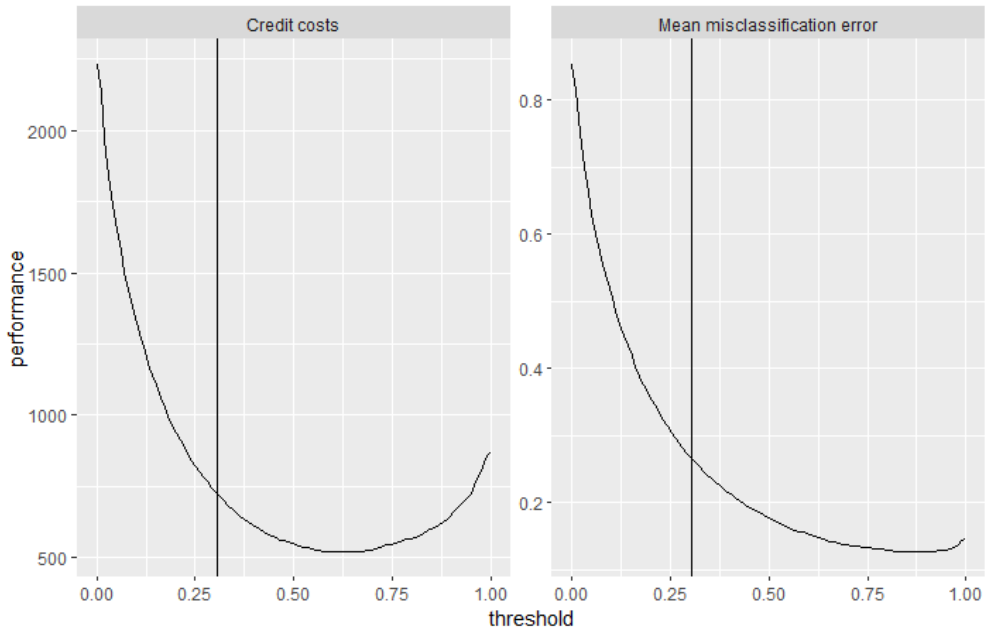

Figure 1: Logistic regression: threshold vs. performance plot

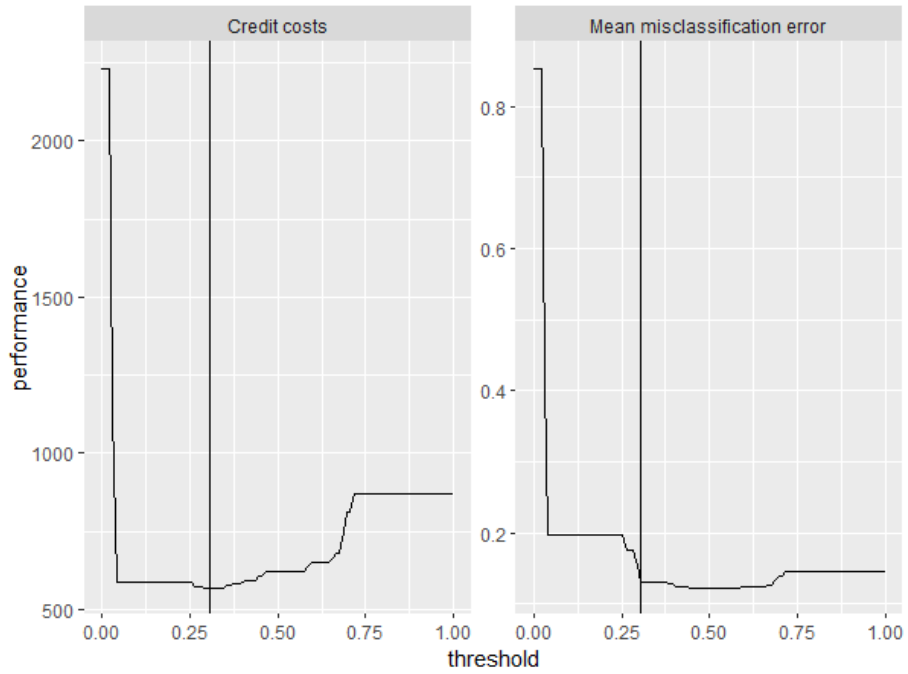

Figure2:: Decision Tree: threshold vs. performance plot

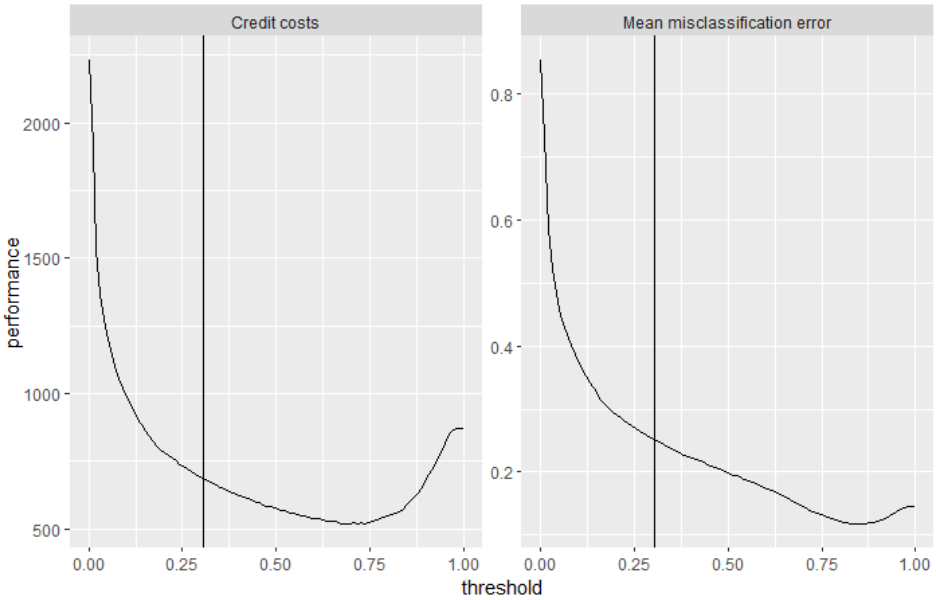

Figure3: XG-Boost: threshold vs. performance plot

In each graph, the vertical line represents the theoretical threshold values (th). Based on the lowest misclassification error in model's performance plots, empirical values are derived below:

- Logistic Regression: $e - th = 0.62$
- Decision Trees: $e - th = 0.61$
- XG-Boost: $e - th = 0.70$

Using *e-th* and *th* values, Table 10 shows comparison of model's average costs and mmce to find the best threshold values for optimisation.

Table 5: Comparison of theoretical and empirical threshold values

Model	Threshold values	Average Credit costs	Mean misclassification error (mmce)
Logistic Regression	$th = 0.30$	686.55	0.2525
	$e - th = 0.62$	491.57	0.1409
Decision Trees	$th = 0.30$	638.85	0.1230
	$e - th = 0.61$	568.28	0.1253
Extreme Gradient Boosting	$th = 0.30$	547.78	0.1931
	$e - th = 0.70$	469.27	0.1360

It is evident, empirical threshold values have lower misclassification costs than their theoretical counterparts. Hence, *e-th* values are used to optimise CSL-A models.