Loan Default Prediction on Indian MFI Dataset

A Project Report

submitted by

V SAI KRISHNA

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

May 2022

THESIS CERTIFICATE

This is to certify that the thesis titled **Loan Default Prediction on Indian MFI Dataset**, submitted by **V SAI KRISHNA**, to the Indian Institute of Technology, Madras, for the award of credits for the course **EE6901, EE6902 and EE6903 (M.Tech Project1, M.Tech Project2 and M.Tech Project3 respectively)** in partial fulfilment of the requirements for the award of the degree of **Master of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Gaurav Raina
Research Guide
Associate Professor
Dept. of Electrical Engineering
IIT Madras, 600036
Place: Chennai
Date: 3rd June 2022

ACKNOWLEDGEMENTS

First and foremost I am extremely grateful to my guide, Prof. Gaurav Raina and my supervisors at Kaleidofin Private Limited, Mrs Suniti Srivastava, Mr Rohit Ganpath Raj, and Ms Dhaarini for their invaluable advice, continuous support, and patience during my M.Tech Project. Their immense knowledge and plentiful experience have encouraged me in all the time of my research and daily life.

I would like to express my gratitude to my parents. Without their tremendous understanding and encouragement in the past two semesters, it would be impossible for me to complete my study.

ABSTRACT

KEYWORDS: Credit risk estimation; Loan default prediction; Generic model; Segmentation; Machine learning.

In today's world, obtaining loans from banks and other financial institutions has become widespread. Every day many people apply for loans for a variety of purposes. However, not all the applicants are dependable, and not everyone can be endorsed. Several financial institutions (also called partners) approach fin-tech companies like Kaleidofin (with whom we collaborated for this project) for building a credit risk model. The aim of this thesis is to collect anonymized dataset from Indian micro-finance institutions (MFIs), perform feature engineering and build robust machine learning models to predict the loan default rate.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

MFI	Micro Finance Institutions
ML	Machine Learning
EDA	Exploratory Data Analysis
CB	Credit Bureau
EMI	Every Month Installment
RF	Random Forest
PCA	Principal Component Analysis
XGBoost	eXtreme Gradient Boosting
WOE	Weight of Expectations
IV	Information Value
AUC	Area Under Curve
ROC	Receiver Operating Characteristic

CHAPTER 1

INTRODUCTION

1.1 Background

In today's world, obtaining loans from banks and other financial institutions has become widespread. Every day many people apply for loans for a variety of purposes. However, not all the applicants are dependable, and not everyone can be endorsed. When these unreliable customers do not repay the bulk of the loan amount to the bank/MFIs, they lead to substantial monetary loss. Hence, the risk associated with deciding on a loan approval is immense. As a result, many financial institutions build machine learning models to estimate the credit risk of a customer.

Although big financial institutions can afford to hire their own ML team to build such models, small institutions cannot. This is where fin-tech companies like Kaleidofin come into play. Several financial institutions (also called partners) may approach fin-tech companies for building a credit risk model.

In this thesis, we will collect anonymized dataset from multiple partner institutions (predominantly Indian MFIs) and build a robust generalized ML model. Building partner wise models takes a long time right from data ingestion, EDA to final feature engineering and model building. The same process being followed across multiple partners adds to time complexity. Hence, this generalized model can later be used to build partner specific models.

1.2 Steps involved

The overall process of generic model build can be divided into following steps:

- Data standardization and mapping
- Core Feature Engineering
- Model Training and Tuning over a combined dataset of several partners data

1.2.1 Data Standardization and Mapping

All partners do capture common data in terms of loan application data, customer demographics data, credit bureau data and loan demand and repayment data. Besides this, some partners might capture additional information e.g. detailed asset ownership data, savings bank account data (in case of co-operative banks) etc. A standardized data schema can be created which is a super-set of all this information.

Once a partner data comes in, a simple mapper needs to be created which will map the partner variables to this standardized schema fields. This ensures different partner data, with varied variable names and data types is standardized to a common schema from this step onward. Data about the customer is obtained from partners (customer and loan data) as well as from the Credit Bureau. Hence we need to have different schema for each of them.

- Credit Bureau data schema : Inputs from both CRIF-High Mark and Equifax have been taken to build this standardized schema.
- Partner data schema : The data collected from the partner institution can be divided into Customer level data and Loan level data. The design of the schema for both sets of data should be versatile to ensure accommodation of different partner type data.

Schema is designed to be futuristic to include all potential data points that can be captured by an MFI regarding assets, savings, bank details , dependents etc.

1.2.2 Core Feature Engineering

Once both loan bureau data is standardized, one code can be used to create model features in a partner agnostic way. In case of additional data being captured by certain partners, there would be incremental set of features for those partners.

1.2.3 Model Training and Tuning over combined dataset consisting of several partner data

Several partner datasets needs to be combined to create training and validation datasets for generalized model training. The datasets should be chosen that they cover different geographies, different occupation types, different loan types, varied disbursement amounts etc. so that they can later work well on unseen datasets.

Detailed Study needs to be done on stability of created model features such that it holds across partner datasets and also across different years /economic conditions. Two or three partner datasets should ideally be used as unseen test sets to test the efficacy of the generic model on completely unseen datasets.

In case a partner requires a model that is fine tuned for its specific use case, the generic model score can be combined with additional data points captured specifically by the MFI to create a fine tuned model score for the particular MFI. This should be a quicker approach than creating a partner tuned model from scratch.

CHAPTER 2

Concepts involved

In the section, we will look at the theory behind binary classification problem, as we are dealing with default prediction. We will also take a look at the implemented variable selection technique and delve into the working principles of the ML algorithms and evaluation metrics used.

2.1 Binary Classification Problem

Binary classification tasks typically involve one class that is the normal state and another class that is the abnormal state. In this project, customers belong either to the non-default category or to the default category. The output (denoted by the random variable Y $\in \{0, 1\}$) is either 0 (for non-default customers) or 1 (for default customers). The random variable Y_i is the target variable and will take the value of y_i, where i corresponds to the i^{th} observation in the data set. For some methods, the variable $\bar{y}_i = 2y_i - 1$ will be used, since these methods require the response variable to take the values $\bar{y}_i \in \{1, 1\}$. Granström and Abrahamsson (2019)

Other information about the customers, such as the EMI paid, delay in payments, age of the customer, etc, can be modeled as the input variables. These variables can be both continuous and categorical, and are often referred to as features. Let $X_i \in R_p$ denote a real valued random input vector and an observed feature vector be represented by $x_i = [x_{i1}, x_{i2}, ..., x_{ip}]^T$, where p is the total number of features. Then the observation data set with N samples can be expressed as D $= \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$. Granström and Abrahamsson (2019)

We will use Logistic Regression and XGBoost in this project. These two methods consistently outperform other classification methods on loan default prediction tasks. The theory for these classifiers will be explained in more detail in the sections below.

2.2 Logistic Regression

Logistic regression, despite its name, is a classification model rather than regression model. It is a powerful discriminating modeling approach, where we estimate the posterior probabilities of classes given $X = x_i$ directly without assuming the marginal distribution on X. The posterior probability for a customer to be in the default class with a given input $\mathbf{x}_i$ can be obtained with the logistic function as James *et al.* (2013)

$$P\left(Y_i = 1 \mid X_i = \mathbf{x}_i\right) = \frac{e^{\beta_0 + \boldsymbol{\beta}^\top \mathbf{x}_i}}{1 + e^{\beta_0 + \boldsymbol{\beta}^\top \mathbf{x}_i}}$$

where the parameters β_0 and β are parameters of a linear model with β_0 denoting an intercept and β denoting a vector of coefficients, $\beta = [\beta_1, \beta_2, \ldots, \beta p]^\top$. The logistic function from the above equation is derived from the relation between the log-odds of $P\left(Y_i = 1 \mid X_i = \mathbf{x}_i\right)$ and a linear transformation of $\mathbf{x}_i$, that is Granström and Abrahamsson (2019)

$$\log \frac{P\left(Y_i = 1 \mid X_i = \mathbf{x}_i\right)}{1 - P\left(Y_i = 1 \mid X_i = \mathbf{x}_i\right)} = \beta_0 + \boldsymbol{\beta}^\top \mathbf{x}_i$$

The class prediction can then be defined as

$$\hat{y}_i = \begin{cases} 1, & \text{if } P\left(Y_i = 1 \mid X_i = \mathbf{x}_i\right) \geq c \\ 0, & \text{if } P\left(Y_i = 1 \mid X_i = \mathbf{x}_i\right) < c \end{cases}$$

where c is a threshold parameter of the decision boundary which is usually set to $c = 0.5$ Finance (2017). Further, in order to find the parameters β_0 and $\boldsymbol{\beta}$, the maximization of the log-likelihood of Y_i is performed. After some manipulation, the expression can be rewritten as

$$p\left(\mathbf{x}_i; \beta_0, \boldsymbol{\beta}\right) = P\left(Y_i = 1 \mid X_i = \mathbf{x}_i; \beta_0, \boldsymbol{\beta}\right) = \frac{1}{1 + e^{-\left(\beta_0 + \boldsymbol{\beta}^\top \mathbf{x}_i\right)}}$$

Since $P\left(Y_i = 1 \mid X_i = \mathbf{x}_i; \beta_0, \boldsymbol{\beta}\right)$ completely specifies the conditional distribution, the multinomial distribution is appropriate as the likelihood function Hastie *et al.*

(2009). The loglikelihood function for N observations can then be defined as

$$\begin{aligned} l\left(\beta_0, \boldsymbol{\beta}\right) &= \sum_{n=1}^{N}\left[y_n \log p\left(\mathbf{x}_n; \beta_0, \boldsymbol{\beta}\right) + \left(1-y_n\right)\log\left(1-p\left(\mathbf{x}_n; \beta_0, \boldsymbol{\beta}\right)\right)\right] \\ &= \sum_{n=1}^{N}\left[y_n\left(\beta_0 + \boldsymbol{\beta}^\top \mathbf{x}_n\right) - \log\left(1+e^{\left(\beta_0+\boldsymbol{\beta}^\top \mathbf{x}_n\right)}\right)\right] \end{aligned}$$

Let $\theta = \{\beta_0, \boldsymbol{\beta}\}$ and assume that $\mathbf{x}_n$ includes the constant term 1 to accommodate β_0. Then, in order to maximize the log-likelihood, take the derivative of l and set to zero

$$\frac{\partial l(\theta)}{\partial \theta} = \sum_{n=1}^{N} \mathbf{x}_n \left(y_n - p\left(\mathbf{x}_n; \theta\right)\right) = 0.$$

The above equation generates $p+1$ equations nonlinear in θ. To solve these equations, the Newton-Rahpson method can be used. In order to use this method, the second derivative must be calculated Granström and Abrahamsson (2019)

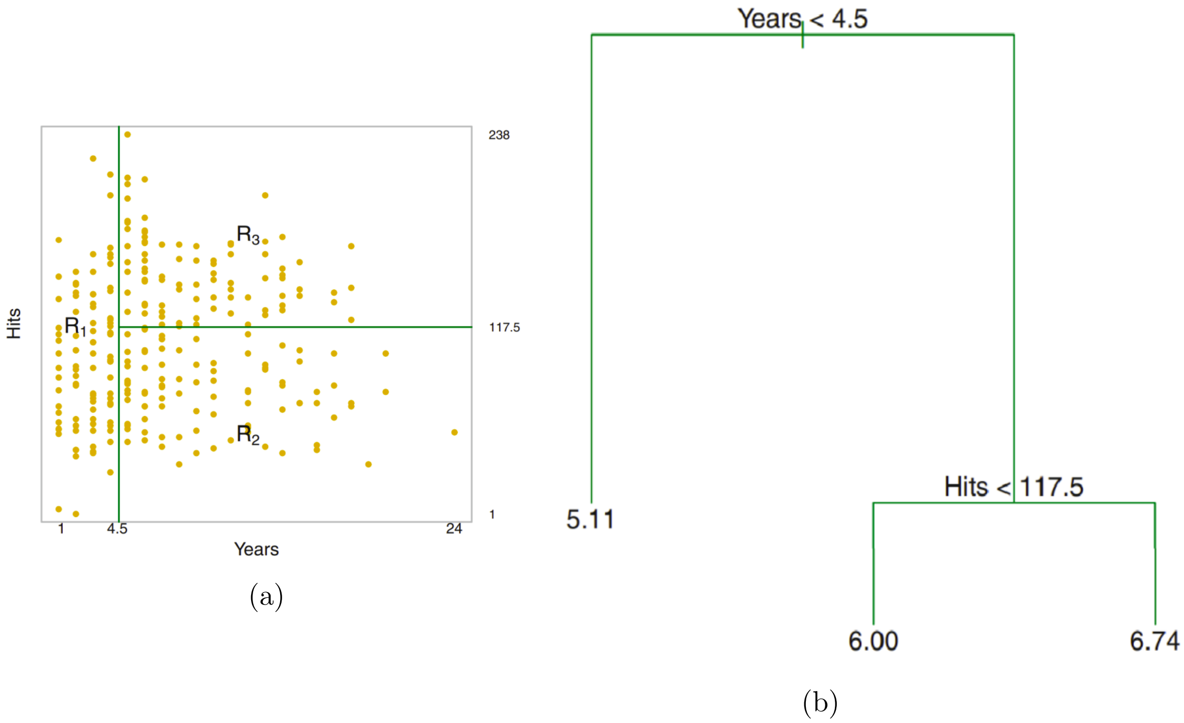
$$\frac{\partial^2 l(\theta)}{\partial \theta \partial \theta^\top} = -\sum_{n=1}^{N} \mathbf{x}_n \mathbf{x}_n^\top p\left(\mathbf{x}_n; \theta\right)\left(1 - p\left(\mathbf{x}_n; \theta\right)\right)$$

A single Newton-Rahpson update will then be performed as

$$\theta^{\text{new}} = \theta^{\text{old}} - \left(\frac{\partial^2 l(\theta)}{\partial \theta \partial \theta^\top}\right)^{-1} \frac{\partial l(\theta)}{\partial \theta}.$$

2.3 Decision Trees

A decision tree algorithm binary splits the feature space into subsets in order to divide the samples into more homogeneous groups. This can be implemented as a tree structure, hence the name decision trees. An example of a two-dimensional split feature space and its corresponding tree can be seen in Figure 2.1 Hastie *et al.* (2009) Granström and Abrahamsson (2019). The terminal nodes in the tree in Figure 2.1 are called leaves and are the predictive outcomes. In this particular example, a regression tree which predicts quantitative outcomes has been used. In a subset of the feature space, represented by the region R_m with N_m number



(a)

(b)

Figure 2.1: (a) Two dimensional feature space split into three subsets. (b) Corresponding tree to the split of the feature space. Hastie *et al.* (2009)

of observations, let the indicator function be $I(\cdot)$ and

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{\mathbf{x}_i \in R_m} I\left(y_i = k\right)$$

be the fraction of class k observations in R_m[19]. Then the observations lying in R_m will be predicted to belong to class $k(m) = \arg_{\max_k} \hat{p}_{mk}$. Since the Gini index, defined by

$$G = \sum_{k=1}^{K} \hat{p}_{mk} \left(1 - \hat{p}_{mk}\right)$$

is amenable for numerical optimization [20], it will be chosen as the criterion for binary splitting.

2.4 Random Forest

Before describing the random forest classifier, let us discuss two essential concepts: Bootstrapping and Bagging. The bootstrap method is a statistical technique for estimating quantities about a population by averaging estimates from multiple small data samples. Importantly, samples are constructed by drawing observations from a large data sample one at a time and returning them to the data sample after they have been chosen. This allows a given observation to be included in a given small sample more than once. This approach to sampling is called sampling with replacement.

Bootstrap aggregating also called bagging (from bootstrap aggregating), is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid overfitting. The method bootstraps the original data set to fit separate models for each bootstrapped data set and takes the average of the predictions made by each model. For a given data set z, the method can be expressed as Granström and Abrahamsson (2019)

$$\hat{f}_{bag}(z) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(z),$$

where B is the total amount of bootstrapped data sets and $\hat{f}^{*b}(z)$ is a model used for the bth bootstrapped data set. In a classification setting, instead of taking the average of the models, a majority vote is implemented. When applying bagging to decision trees, the following should be considered. If there is one strong predictor in the data set along with moderately strong predictors, most of the top splits will be done based on the strong predictor. This leads to fairly similar looking trees that are highly correlated. Averaging highly correlated trees does not lead to a large reduction of variance.

The random forest classifier has the same setup as bagging when building trees on bootstrapped data sets but overcomes the problem of highly correlated trees. It decorrelates the trees by taking a random sample of m predictors from the full set of p predictors at each split and uses randomly one among the m predictors to split. Granström and Abrahamsson (2019)

2.5 Boosting

Boosting works in a similar way as bagging regarding combining models and creating a single predictive model, but it does not build trees independently, it builds trees sequentially. Building trees sequentially means that information from the previous fitted tree is used for fitting the current tree. Rather than fitting separate trees on separate bootstrapped data sets, each tree is fit on a modified version of the original data set. Granström and Abrahamsson (2019)

2.5.1 XGBoost

XGBoost is an abbreviation of eXtreme Gradient Boosting. One of the evident advantages of XGBoost is its scalability and faster model exploration due to the parallel and distributed computing Chen and Guestrin (2016). In order to understand XGBoost's algorithm, some basic introduction to how gradient tree boosting methods works will be presented. Let N be a number of samples in the data set with p features, $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ $(|\mathcal{D}| = N, \mathbf{x}_i \in R^p$ and $y_i \in \{0,1\})$. To predict the output, M additive functions are being used Granström and Abrahamsson (2019)

$$\phi\left(\mathbf{x}_i\right) = \sum_{k=1}^{M} f_k\left(\mathbf{x}_i\right), \quad f_k \in \mathcal{S}, \quad \mathcal{S} = \left\{f(\mathbf{x}) = \mathbf{w}_{q(\mathbf{x})}\right)\Bigg\}$$

where $\mathcal{S}$ is the classification trees' space, q is the structure of a tree and $q : R^p \to T$, $\mathbf{w} \in R^M$ Chen and Guestrin (2016). Further, T is the number of leaves, f_k is an independent tree structure of q and leaf weights w, which can also be viewed as a score for i th leaf, w_i. Learning is being executed by minimization of the regularized objective and is derived as the following equation

$$\mathcal{L}(\phi) = \sum_{i=1}^{N} l\left(y_i, \phi\left(\mathbf{x}_i\right)\right) + \sum_{k=1}^{M} \Omega\left(f_k\right)$$

where $\Omega(f)$ is defined as follows

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j}^{T} w_j^2$$

The function $\Omega(f)$ penalizes the complexity of the model by the parameter γ, which penalizes the number of leaves, and λ which penalizes the leaf weights. The loss function l measures the difference between the prediction $\phi\left(\mathbf{x}_i\right)$ and the target y_i Chen and Guestrin (2016). Further, let $\phi\left(\mathbf{x}_i\right)^{(t)}$ be the prediction of the i th observation at the t-th iteration, then f_t is needed to add in order to minimize the following objective

$$\mathcal{L}^{(t)} = \sum_{i=1}^{N} l\left(y_i, \phi\left(\mathbf{x}_i\right)^{(t-1)} + f_t\left(\mathbf{x}_i\right)\right) + \Omega\left(f_t\right),$$

where f_t is chosen greedily so that it improves the model the most. Second-order approximation can be used to quickly optimize the objective in the general setting Chen and Guestrin (2016)

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{N} \left[l\left(y_i, \phi\left(\mathbf{x}_i\right)^{(t-1)}\right) + g_i f_t\left(\mathbf{x}_i\right) + \frac{1}{2} h_i f_t^2\left(\mathbf{x}_i\right)\right] + \Omega\left(f_t\right)$$

where $g_i = \partial_{\phi(\mathbf{x}_i)^{(t-1)}} l\left(y_i, \phi\left(\mathbf{x}_i\right)^{(t-1)}\right)$ and $h_i = \partial^2_{\phi(\mathbf{x}_i)^{(t-1)}} l\left(y_i, \phi\left(\mathbf{x}_i\right)^{(t-1)}\right)$. Simplification of the function can be made by removing a constant term $l\left(y_i, \phi\left(\mathbf{x}_i\right)^{(t-1)}\right)$. and by expanding the Ω function the following expression can be obtained Granström and Abrahamsson (2019)

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{N} \left[g_i f_t\left(\mathbf{x}_i\right) + \frac{1}{2} h_i f_t^2\left(\mathbf{x}_i\right)\right] + \gamma T + \frac{1}{2}\lambda \sum_{j}^{T} w_j^2$$

Let $I_j = \{i \mid q\left(\mathbf{x}_i\right) = j\}$ be the instance of leaf j. Further, the equation is being simplified to

$$\tilde{\mathcal{L}}^{(t)} = \sum_{j}^{T} \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right) \right] + \gamma T.$$

Now, the expression for the optimal weight w_j^* can be derived from the above equation

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}.$$

Thus, the optimal value is given by

$$\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2} \sum_{j}^{T} \frac{\left(\sum_{i \in I_j} g_i\right)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T$$

The final classification is then

$$\hat{y}_i = \begin{cases} 1, & \text{if } \phi\left(\mathbf{x}_i\right) \geq c \\ 0, & \text{if } \phi\left(\mathbf{x}_i\right) < c \end{cases}$$

where c is a chosen decision boundary and $\phi\left(\mathbf{x}_i\right) \in (0, 1)$.

2.6 Feature Selection Techniques

Credit/Loan data is generally humongous and not all features available to us are useful in model prediction. Feature selection is the process of isolating the most consistent, non-redundant, and relevant features to use in model construction. Methodically reducing the size of datasets is important as the size and variety of datasets continue to grow. The main goal of feature selection is to improve the performance of a predictive model and reduce the computational cost of modeling. In this project we will be using Variable Clustering and Information Value (IV) obtained using Weight of Evidence (WOE) for feature selection.

2.6.1 Variable Clustering

Variable clustering is a useful tool for data reduction, such as choosing the best variables or cluster components for analysis. Variable clustering removes collinearity, decreases variable redundancy, and helps reveal the underlying structure of the input variables in a data set. Before proceeding further, we need to understand the R-squared measure.

R-squared

The coefficient of determination, R-squared or R^2, is used to analyze how differences in one variable can be explained by a difference in a second variable. It doesn't tell you whether our chosen model is good or bad, nor will it can tell whether the data and predictions are biased. The formula to calculate R-squared between two features/variables x and y are: sta (2021)

$$R^2_{xy} = \frac{(\sum xy - \sum x \sum y)^2}{(\sum x^2 - (\sum x)^2) - (\sum y^2 - (\sum y)^2)}$$

Now that we saw how to measure R-squared metric, let us continue discussing about the variable clustering method. In this method, we first divide a set of numeric variables into either disjoint or hierarchical clusters. Associated with each cluster is a linear combination of the variables in the cluster, which may be either the first principal component or the centroid component. The rule dictates to select the variable with the minimum 1-R^2 as the cluster representative. The $1 - R^2$ is defined as: Datalab (2018)

$$1 - R^2 = \frac{(1 - R^2_{\text{own cluster}})}{(1 - R^2_{\text{nearest cluster}})}$$

Intuitively, we want the cluster representative to be as closely correlated to its own cluster and as uncorrelated to the nearest cluster. Therefore, the optimal representative of a cluster is a variable where 1-R^2 tends to zero.

Typically, in the clustering literature, there is a rule for selecting the cluster representative, the 1-R^2. Business “knowledge from subject matter expert should also complement this rule to guide the selection of variables. For this reason, we could decide to use more than one variable per duster. Also, for business justification alternate variable may provide a better intuitive interpretation of the model than the cluster representative.

2.6.2 Weight of Evidence

The weight of evidence tells the predictive power of an independent variable in relation to the dependent variable. Since it evolved from credit scoring world, it is generally described as a measure of the separation of good and bad customers. "Bad Customers" refers to the customers who defaulted on a loan. and "Good Customers" refers to the customers who paid back loan. In general, for each class i of an independent variable x, we want to find the ratio of the proportion/percentage of the population, whose dependent variable y belongs to a certain class, that has the class i, followed by natural log. Quant (2020) Bhalla

$$WOE_{x=i} = ln \left(\frac{\text{\% of y = 0 where x = i}}{\text{\% of y = 1 where x = i}} \right)$$

The steps for calculating WOE are: Bhalla

- For a continuous variable, split data into 10 parts (or lesser depending on the distribution).
- For a categorical variable, you do not need to split the data (Ignore the above step follow the remaining steps)
- Calculate the number of events and non-events in each group (bin)
- Calculate the % of events and % of non-events in each group.
- Calculate WOE by taking natural log of division of % of non-events and % of events

The benefits of WOE are: Bhalla

- It can treat outliers. Suppose you have a continuous variable such as annual salary and extreme values are more than 500 million dollars. These values would be grouped to a class of (let's say 250-500 million dollars). Later, instead of using the raw values, we would be using WOE scores of each classes.
- It can handle missing values as missing values can be binned separately.
- Since WOE Transformation handles categorical variable so there is no need for dummy variables.
- WOE transformation helps you to build strict linear relationship with log odds. Otherwise it is not easy to accomplish linear relationship using other transformation methods such as log, square-root etc. In short, if you would not use WOE transformation, you may have to try out several transformation methods to achieve this.

2.6.3 Information Value

Information value is one of the most useful technique to select important variables in a predictive model and this is what we'll be using in this project for feature selection. It helps to rank variables on the basis of their importance. The IV is calculated using the following formula: Quant (2020)

$$IV_x = \sum_{x,i} \text{(given x = i, \% of y = 0 - \% of y = 1)} * WOE_{x=i}$$

Once we have the IV of the variable, we can check against table 2.1 to see the predictive power of the variable. Siddiqi (2006)

Table 2.1: IV value vs Predictive power

Information value (IV)	**Predictive power**
Less than 0.02	Not useful for prediction
0.02 to 0.1	Weak predictive Power
0.1 to 0.3	Medium predictive Power
0.3 to 0.5	Strong predictive Power
Greater than 0.5	Suspicious Predictive Power

2.7 Evaluation Metrics

2.7.1 Confusion Matrix

One common way to evaluate the performance of a model with binary responses is to use a confusion matrix. The observed cases of default are defined as positives and non-default as negatives Finance (2017). The possible outcomes are then true positives (TP) if defaulted customers have been predicted to be defaulted by the model. True negatives (TN) if non-default customers have been predicted to be non-default. False positives (FP) if non-default customers have been predicted to be defaulted, and false negatives (FN) if defaulted customers have been predicted to be non-default. A confusion matrix can be presented as in the Figure 2.2. From a confusion matrix there are certain metrics that can be taken into consideration. The most common metric is accuracy which is defined as the fraction of the total number of correct classifications and the total number of observations. It is mathematically defined as: Granström and Abrahamsson (2019)

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

The issue with using accuracy as a metric is when applying it for imbalanced data. If the data set contains 99% of one class it is possible to get an accuracy of 99%, if all of the predictions are made for the majority class. A metric that is more

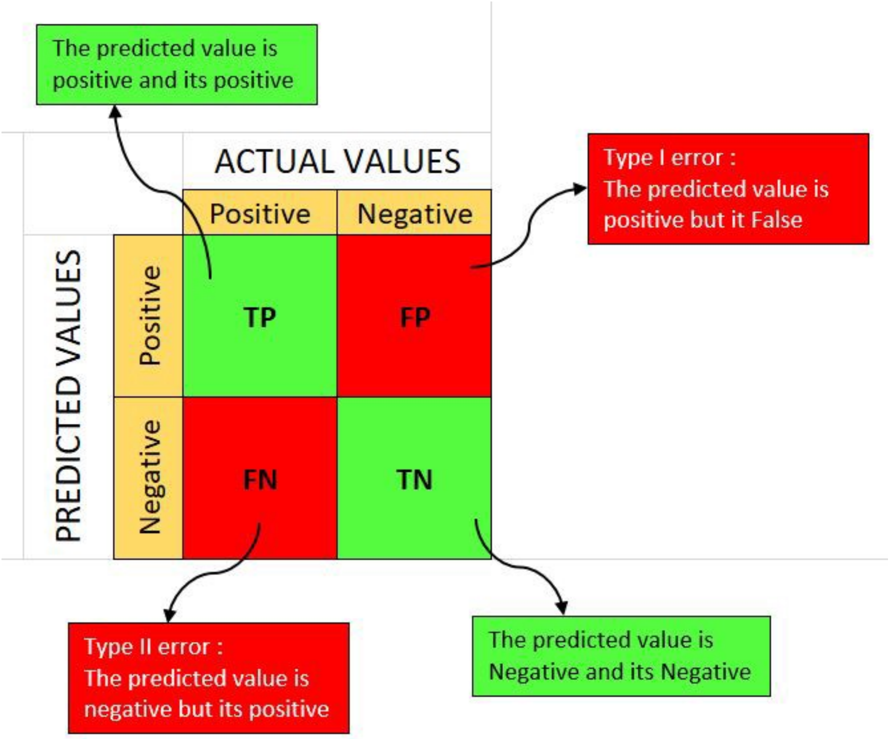

The predicted value is positive and its positive
ACTUAL VALUES
Positive
Negative
Type I error :
The predicted value is positive but it False
PREDICTED VALUES
Positive
Negative
TP
FP
FN
TN
Type II error :
The predicted value is negative but its positive
The predicted value is Negative and its Negative

Figure 2.2: Confusion Matrix

relevant in the context of this project is specificity. It is defined as:

$$Specificity = \frac{TN}{TN + FP}$$

and will be used to illustrate the theory behind the ROC-AUC. In terms of business sense, the aim is to balance a trade-off between losing money on non-performing customers and the opportunity cost caused by declining a potentially performing customer. Thus, there is a high pertinence in analyzing how sensitivity and precision are influenced by various methods, as sensitivity estimates how many customers defaulted. In contrast, precision relates to the potential opportunity cost.

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

Since sensitivity and precision are of equal importance in this project, a trade-off between these metrics is considered. The F-score is the weighted harmonic average of precision and sensitivity Goutte and Gaussier (2005). The definition of F-score can be expressed as:

$$F = (1+\beta^2)\frac{Precision * Sensitivity}{Sensitivity + \beta^2 * Precision} = \frac{(1+\beta^2))TP}{(1+\beta^2)TP + \beta^2 FN + FP}$$

where β is a weight parameter. As mentioned before, both measures of precision and sensitivity are equally relevant and therefore the weight is set to $= 1$. Further, the F-score takes both of these measures into consideration, and thus performance of every method will be primarily evaluated and compared with the regards to this metric.

2.8 Area Under the Receiver Operator Characteristic Curve

Another way to evaluate results from the models is to analyze the Receiver Operator Characteristic (ROC) curve and its Area Under the Curve (AUC). In this section, the definition of ROC will be provided, followed by the explanation of AUC. Let V_0 and V_1 denote two independent random variables with cumulative distribution functions F0 and F1 respectively. The random variables V_0 and V_1 describe the outcomes predicted by a model if a customer has defaulted or not. Let c be a threshold value for the default classification such that if the value from the model is greater or equal to c, a customer is classified as default and non-default otherwise. Further, in this setting, sensitivity and specificity are defined then in the following way Granström and Abrahamsson (2019)

$$\text{Sensitivity }(c) = P\left(V_1 \geq c\right) = 1 - F_1(c),$$

$$\text{Specificity }(c) = P\left(V_0 < c\right) = F_0(c).$$

The ROC curve uses the false positive fraction in order to describe the trade-offs between sensitivity and (1-specificity). Let m express $1 - F_0(c)$, then the following definition for the ROC curve is obtained

$$ROC(m) = 1 - F_1\left\{F_0^{-1}(1-m)\right\}$$

where $0 \leq m \leq 1$ and $F_0^{-1}(1-m) = \inf\{z : F_0(z) \leq 1-m\}$. A ROC curve can also be summarized by AUC score, which represents an index for ROC curve and is defined in following way

$$AUC = \int_0^1 ROC(m)dm.$$

2.9 GINI

The Gini index or coefficient is a way to adjust the ROC-AUC so that it can be clearer and more meaningful. It's more natural for us to see a perfectly random model having 0, reversing models with a negative sign and the perfect model having 1. The range of values now is [-1, 1].

Perfectly reversing model

This model is doing the exact opposite of a perfect model. It's predicting every positive observation as a negative one and vice-versa. This means if we invert all the outputs we'll have a perfect model. It has a Gini=-1 and AUC=0. And if you have a model like this, or a model having a negative Gini, you've surely done something wrong.

Imperfect model

The imperfect model is the worst model we can have. It means this model has no discrimination ability to distinguish between the two classes. It's a perfectly random model. It has a Gini=0 and AUC=0.5

Perfect model

The perfect model is the model that predicts every observation correctly for positive and negative classes. It means in every threshold at least one of FP and TP is equal to zero. This model has an AUC=1 and a Gini=1.

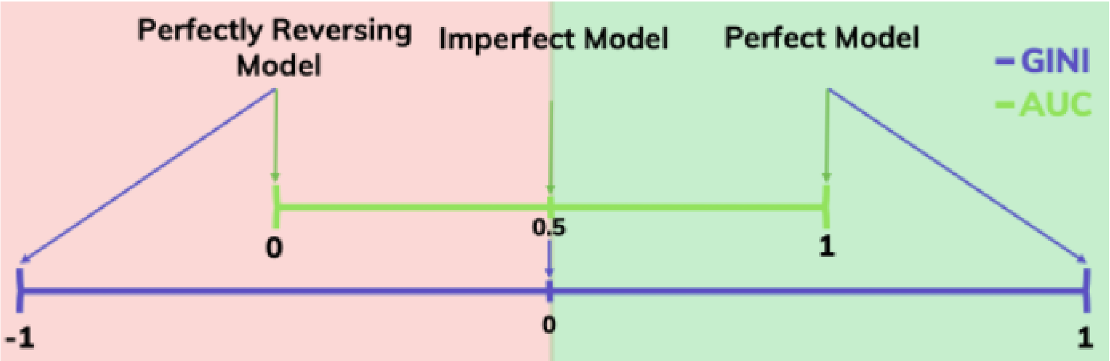


Figure 2.3: ROC-AUC vs GINI

CHAPTER 3

Dataset

The dataset can be divided into 2 broad categories. They are:

1) **Segment 1 (Seg-1)**: Customers in this segment have previously taken loans from the partner institution. In other words, these are old customers and we can use CB features, customer level features and loan level features (the last two are obtained from the partner institution) to predict the credit risk.

2) **Segment 2 (Seg-2)**: Customers in this segment are new to the partner institution. We will have to predict the credit risk using only the CB features, and customer level features.

We will now look at the features obtained from each of our sources (CB and Partner institution)

3.1 Credit Bureau

Before looking at the features we need to know what Credit Bureau is. The simplest answer is that credit bureaus, like Equifax, **are data collectors**. Credit bureaus, also known as credit reporting agencies, do two things:

1. We compile your credit history based on your credit accounts, using your Social Security number or other identification information.

2. We provide your credit information, in the form of credit reports, to lenders and creditors to help them determine your creditworthiness. We also provide credit reports to you, so you can better understand your credit situation. Your credit history, including factors such as your payment history and your amounts owed, are used along with other factors to calculate your credit scores.

A frequent misconception about the three nationwide credit bureaus (Equifax, Experian and TransUnion) is that they make lending decisions. Credit bureaus provide some of the information creditors and lenders use to help them make important lending decisions. While credit bureaus collect credit information in order to make it available to certain third parties, the decision to deny or approve someone credit ultimately lies with the lender or creditor. Each lender and creditor may have its own criteria.

3.1.1 Credit Bureau Features

Now that we know what Credit Bureau is, let us take a look at the features obtained from them. Note that different CB institutions give data in different formats. The table presented here just represents the features obtained. The actual name of these features will vary across different Credit Bureaus. The features obtained from the Credit Bureau are as follows: (Note that a single customer (unique Customer ID) can have multiple addresses/contact details/loan details).

Table 3.1: Credit Bureau Features

Basic Details	**General Details**	**Loan Details**
Customer ID	Total no of loan accounts	Last Payment Date
Loan ID	No of accounts in other institutions	Institution
Date of Birth	No of default accounts	Loan type
Address Features	No of closed accounts	Loan frequency
Address	No of active accounts	Disbursement Amount
State	No of accounts in other institutions	Current Balance
Pin-code	No of accounts in partner institutions	Installment Amount
Contact details	Total disbursed amount in partner institutions	Overdue Amount
Phone Number	Total disbursed amount in other institutions	Write off Amount
Email ID	Total current EMI	Disbursement Date
Loan Enquiry Details	Total EMI towards other institutions	Loan close date
Enquiry Date	Total installment amount	Loan update date
Enquiry Purpose	Total installment amount to other institutions	Loan cycle
Enquiry Amount	Maximum worst delinquency	

3.2 Features obtained from partner institutions

As mentioned in chapter 1, the features obtained from partner institutions can be divided into two categories. They are Customer level features and Loan level features. Let us look at each of them in detail.

3.2.1 Customer level features

These features contain the personal information of the customer. These customer features are split across four different tables. They are:

- Customer Information: Contains personal information about the customer like Customer's full name, Email ID, etc
- Customer Address: Contains information about the customer's location
- Customer Cashflow: Contains information about the assets and liabilities owned by the customer
- Customer other: Contains other useful information that doesn't fall under the above categories, like family size.

3.2.2 Loan level features

These features contain details about the loans (current loan included) taken by the customer. Each loan goes through multiple stages before getting disbursed. Hence, these loan features are obtained from five different tables. They are:

- Proposal Data: Contains information about the loan at the time of proposal by the customer
- Sanction Data: Contains information about the loan after sanctioning (stage before final approval)
- Approval Data: Contains information about the loan post approval
- Disbursement Data: Some loans get cancelled or get the terms changed post sanctioning. This table contains details of loans at the time of disbursement.
- Transaction details: Contains information about every repayment made by the customer. This table also contains information about the expected repayment amount from the customer.

CHAPTER 4

Feature Engineering and Selection

In the previous section we took a look at the dataset in our hand. However, the features mentioned above are too crude to be used in our model. Some of them mostly or entirely contain null values. Moreover, it is not ethical to use certain features for predicting the default rate (for example using gender or religion of the customer). Hence, it is required to transform these raw features into something more practical for model building. These features should also inculcate the business/domain knowledge. Since this project is related to financial domain we cannot reveal the actual features used. We instead would mention a broad category to which the feature belongs and the number of features included in that category.

4.1 Features derived from the CB

Table 4.1 contains the features obtained/derived from the CB features in the previous chapter.Since this project is related to financial domain we cannot reveal the actual features used. We instead would mention a broad category to which the feature belongs and the number of features included in that category.

Table 4.1: Final Credit Bureau Features

Feature type	**No of features**
Features that capture variation in reported data	28
Features that capture high Credit Leverage	18
Features that capture delinquency in payment	31
Features that capture loan closure history	13
Total number of features	**90**

4.2 Features derived from customer and loan level features

Table 4.2 contains the partner features. Similar to 4.1, a lot of features are related to each other, and these features are represented together in the table for easier understanding. Moreover, some features in 4.1 are repeated in 4.2. These repeated features are used if available for a customer as they are obtained using partner data, because features obtained using CB data can be slightly outdated. However, most of these repeated features are loan level features, which are available only for Seg1 customers. Hence, as a rule of thumb, we take repeated features from 4.2 if it is a Seg1 customer and from 4.1 for Seg2 customer.

Table 4.2: Final Partner Features

Feature type	**No of features**
Features that capture loan tenure history	4
Features that capture high credit leverage	26
Features that capture delinquency in payment	64
Features that capture loan closure history	15
Features that capture customer income	1

4.3 Feature Selection

As you have already seen, we have Feature Engineered 90 CB features and 110 Partner features (which is a lot!). Partner institutions (especially in financial domain) do not condone black box models (one reason why we won't be using Neural Networks in this thesis for model building) Finance (2017). Partner institutions expect us to build models using fewer, more explainable features rather than a black box model that uses all the features. Hence, the task of feature selection is very crucial.

In order to select the most useful features, we will first calculate the Information Value(IV) using Weight of Expectations (WOE) for all the variables (refer chapter

2). Then we will construct the Variable Clusters and from each cluster we will pick the feature with highest IV. If this feature has an IV less than 0.02 we will neglect this feature as it has no useful predictive power 2.1

4.3.1 Final features for Segment 1 customer

As mentioned before, we obtain Credit Bureau, Customer-level and Loan-level features for Segment 1 customers. The final set of features and their IVs for this set of customers are given in the table 4.3. (Features that start with the prefix CB are Credit Bureau features). One can observe that most of the features with high predictive power are from the CB, while some features from loan level features also show promising predictive power.

Table 4.3: Final features for Seg1 customers

Variable type	**IV**
CB Variation in reported data	0.435854
CB Variation in reported data	0.403947
CB High Credit Leverage	0.388317
CB High Credit Leverage	0.351663
CB Variation in reported data	0.3492
CB Variation in reported data	0.342153
CB High Credit Leverage	0.328047
CB High Credit Leverage	0.321432
CB Delinquency in payment	0.302556
CB High Credit Leverage	0.300428
CB Delinquency in payment	0.300151
CB Variation in reported data	0.299963
CB Delinquency in payment	0.262813
CB Delinquency in payment	0.241394
High Credit Leverage	0.140705
CB Loan closure history	0.06525
Delinquency in payment	0.044842
Loan closure history	0.038139
Loan closure history	0.035865
Loan tenure history	0.030214
Delinquency in payment	0.02863

4.3.2 Final features for Segment 2 customer

As mentioned before, we obtain Credit Bureau, and Customer-level for Segment 2 customers. The final set of features and their IVs for this set of customers are given in the table 4.4. (Features that start with the prefix CB are Credit Bureau features). Since Seg2 customers have no loan-level features the final features contain only CB features.

Table 4.4: Final features for Seg2 customers

Variable	**IV**
CB Variation in reported data	0.36709
CB Variation in reported data	0.330246
CB High Credit Leverage	0.259546
CB Variation in reported data	0.227702
CB Loan closure history	0.145573
CB High Credit Leverage	0.142729
CB Variation in reported data	0.133407
CB High Credit Leverage	0.095839
CB Variation in reported data	0.081403
CB Delinquency in payment	0.027413
CB Delinquency in payment	0.026433

CHAPTER 5

Model Building

In the last chapter we have obtained the final set of features for Seg1 and Seg2 customers (35 for Seg1 and 17 for Seg2). We will now use these features to build models. We will be using Logistic Regression, Random Forest and XGBoost algorithms for our models. For XGBoost and Random Forest, we used Gridsearch to find the best set of hyper-parameters. Refer to 5.1 for the AUC and Gini score obtained using different algorithms.

Table 5.1: Results obtained

Dataset	**Train-Val-Test split**	**Model**	**ROC-AUC**	**Gini**
Segment 1	Train: 646545 (60%) Val: 215515 (20%) Test: 215515 (20%)	Random Forest	Train: 0.948 Val: 0.839 Test: 0.841	Train: 0.896 Val: 0.678 Test: 0.682
		XGBoost	Train: 0.804 Val: 0.794 Test: 0.788	Train: 0.608 Val: 0.588 Test: 0.576
		Logistic Regression	Train: 0.767 Val: 0.767 Test: 0.762	Train: 0.534 Val: 0.535 Test: 0.525
Segment 2	Train: 298830 (60%) Val: 99625 (20%) Test: 99625 (20%)	Random Forest	Train: 0.966 Val: 0.810 Test: 0.813	Train: 0.932 Val: 0.620 Test: 0.625
		XGBoost	Train: 0.823 Val: 0.813 Test: 0.815	Train: 0.646 Val: 0.625 Test: 0.630
		Logistic Regression	Train: 0.796 Val: 0.795 Test: 0.794	Train: 0.593 Val: 0.590 Test: 0.590

From 5.1 we can see that Random Forest and XGBoost give similar test AUC and Gini scores although Random Forest seem to overfit more on the training dataset (hence better training AUC and Gini). Logistic Regression performs the worst in both the datasets.

5.1 Segmentation and Model Building

Instead of building one model on the entire dataset, one can divide the dataset into smaller segments and build separate models for each segment. This process of segmentation+model building might improve the predictive power, but creating separate model for separate segments may be time consuming and not worth the effort ana (2020). In this section, we will try to further divide Seg1 and Seg2 customers, build separate models on each segment and see if we get significant improvement in the results.

But how do we decide the segments? For this, we will build a Decision tree of depth = 5, note down all the node splits (feature used to split and the split value) and use these splits to segment the dataset. Since we are using Decision trees to find the split nodes, we will be training the segmented dataset using **Logistic Regression** only (as XGBoost and Random Forest use Decision trees to split the nodes by default). The best results obtained for Seg1 and Seg2 dataset using segmentation are given in 5.2.

Table 5.2: Results obtained

Dataset	**Best feature for segmentation**	**Segment**	**AUC**	**Gini**	**Combined AUC**	**Combined Gini**
Segment 1	Disbursed Amount	< 16528.5	Train: 0.851 Val: 0.847 Test: 0.859	Train: 0.704 Val: 0.694 Test: 0.719	Train: 0.914 Val: 0.917 Test: 0.918	Train: 0.829 Val: 0.834 Test: 0.837
		>=16528.5	Train: 0.914 Val: 0.917 Test: 0.917	Train: 0.829 Val: 0.834 Test: 0.834		
Segment 2	CB Total EMI	< 3702.7	Train: 0.789 Val: 0.784 Test: 0.784	Train: 0.579 Val: 0.568 Test: 0.567	Train: 0.800 Val: 0.800 Test: 0.800	Train: 0.600 Val: 0.600 Test: 0.600
		>=3702.7	Train: 0.767 Val: 0.772 Test: 0.772	Train: 0.535 Val: 0.544 Test: 0.545		

We can see Segmentation + Logistic Regression has produced better results than simply using Logistic Regression 5.1. In case of Seg1 dataset, Segmentation + Logistic Regression has produced the best results (better than Random Forest and XGBoost too 5.1. In case of Seg2, Random Forest still outperformed Segmentation + Logistic Regression.

CHAPTER 6

Conclusions and Future Work

In this thesis we attempted to build a loan default predictor on the dataset created using multiple MFI datasets. Before building the model, we first mapped the partner features to a standard template, feature engineered and selected the most useful features using Variable Clustering and Information Value. We observed that most of the highly predictive features were from the Credit Bureau. We then build loan default models using Random Forest, XGBoost and Logistic Regression algorithms. Later we tried to improve the performance by combining Segmentation and Logistic Regression. The splits for the segments were obtained using a Decision tree with a depth of 5. The best result for Seg1 dataset was obtained using Segmentation + Logistic Regression 5.2, and for Seg2 dataset was obtained using XGBoost 5.1.

Although segmentation didn't lead to the best results in case of Seg2 dataset, it did produce better results. Hence, exploring different types of Segmentation/Clustering algorithms is a promising way forward for building better models. These models were built using the data collected from only 2 MFIs. Hence, approaching more partners and incorporating their data into our models can also lead to better performance.

REFERENCES

1. (2020). Segmentation: Building predictive models using segmentation. URL `https://www.analyticsvidhya.com/blog/2016/02/guide-build-predictive-models-segmentation/`.

2. (2021). Coefficient of determination (r squared): Definition, calculation. URL `https://www.statisticshowto.com/probability-and-statistics/coefficient-of-determination-r-squared/`.

3. **Bhalla, D.** (). Weight of evidence (woe) and information value (iv) explained. URL `https://www.listendata.com/2015/03/weight-of-evidence-woe-and-information.html`.

4. **Chen, T.** and **C. Guestrin**, Xgboost: A scalable tree boosting system. *In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016.

5. **Datalab, A.** (2018). Learn about variable clustering. URL `https://medium.com/@analyttica/learn-about-variable-clustering-4f765a33d592`.

6. **Finance, J.** (2017). Machine learning in credit risk modeling: Efficiency should not come at the expense of explainability.

7. **Goutte, C.** and **E. Gaussier**, A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. *In European conference on information retrieval*. Springer, 2005.

8. **Granström, D.** and **J. Abrahamsson** (2019). Loan default prediction using supervised machine learning algorithms.

9. **Hastie, T.**, **R. Tibshirani**, **J. H. Friedman**, and **J. H. Friedman**, *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

10. **James, G.**, **D. Witten**, **T. Hastie**, and **R. Tibshirani**, *An introduction to statistical learning*, volume 112. Springer, 2013.

11. **Quant, J.** (2020). Model? or do you mean weight of evidence (woe) and information value (iv)? URL `https://towardsdatascience.com/model-or-do-you-mean-weight-of-evidence-woe-and-information-value`.

12. **Siddiqi, M. N.** (2006). Islamic banking and finance in theory and practice: A survey of state of the art. *Islamic economic studies*, **13**(2).