

AN EXPLAINABLE COST-SENSITIVE CREDIT RISK ASSESSMENT MODEL

BHAWNA GUPTA

Final Thesis

MARCH 2022

## **Dedication**

**To my handsome husband & my beautiful princess**

## **Acknowledgment**

I would like to thank Dr Manoj Jayabalan, Abhishek Potnis and Dr. Rupal Bhargava for their support and guidance during my research.

## **Abstract**

Loan default is the biggest threat in the financial industry. It has a significant impact on the economy as well. So, designing a robust credit assessment model is the need of time. However, with the accuracy of the model's performance, the lending firm is legally bound to explain the reason behind the assessment result to the borrower. But the real challenge in designing such a model is the imbalance of financial dataset and black-box nature of the machine learning model. So, in this study, the ensemble classifiers XGBoost, LightGBM, CatBoost, and Stacking model of all three boosting classifiers were implemented on the imbalanced Lending Club dataset, because ensemble classifiers are known to have lower variance and low bias benefit, so the performance of these classifiers was compared with each other along with the oversampled data created by using SMOTE and ADASYN techniques. All the classifiers' performance was evaluated by misclassification cost metrics type-I and type-II error. And the final result shows that XGBoost and Stacking classifiers with SMOTE technique achieved the highest performance score and lowest misclassification cost. Moreover, the cost-sensitive metrics AUC score, F1-score, G-mean were the highest compared to the previous study performed on the same dataset. And the black-box nature of the classifiers was handled using the explainable AI SHAP technique. The top 5 features with a high contribution to the prediction result were plotted using the SHAP plots with their contribution value, which can be easily interpreted.

## TABLE OF CONTENTS

DEDICATION.....	ii
ACKNOWLEDGEMENTS .....	iii
ABSTRACT .....	iv
LIST OF TABLES .....	vii
LIST OF FIGURES.....	viii
LIST OF ABBREVIATIONS.....	ix
CHAPTER 1: INTRODUCTION.....	1
1.1 Background of the Study .....	1
1.2 Problem Statement.....	2
1.3 Research Questions.....	4
1.4 Aim and Objectives.....	4
1.5 Scope of the Study .....	5
1.6 Significance of the Study .....	5
1.7 Structure of the Study .....	6
CHAPTER 2: LITERATURE REVIEW.....	7
2.1 Introduction.....	7
2.2 Machine Learning in Credit Risk Assessment.....	8
2.2.1 Traditional Classifier .....	8
2.2.2 Advanced Classifier .....	9
2.3 Cost-sensitive learning in credit risk assessment.....	12
2.4 Role of Balancing Technique for Imbalanced Data.....	14
2.5 Need of Explainable AI .....	17
2.6 Discussion.....	18
2.7 Summary.....	23
CHAPTER 3: RESEARCH METHODOLOGY .....	24
3.1 Introduction.....	24
3.2 Research Approach .....	24
3.2.1 Data Acquisition and Description.....	26
3.2.2 Required Resource.....	27
3.2.3 Data Pre-processing .....	27
3.2.4 Data Transformation.....	29
3.2.5 Class Balancing .....	29
3.2.6 Cost-sensitive metrics.....	30
3.2.7 Explainable AI (XAI) .....	31
3.3 Proposed Method (Classification).....	32
3.4 Summary.....	34

CHAPTER 4: ANALYSIS AND DESIGN.....	35
4.1    Introduction.....	35
4.2    Exploratory Data Analysis.....	35
4.2.1    Analysis of loan_status.....	36
4.2.2    Analysis of application_type.....	36
4.2.3    Handling Null Values.....	37
4.2.4    Handling the outliers.....	38
4.2.5    Handling the Skewness.....	39
4.2.6    Handling the Redundant Features.....	40
4.2.7    Univariate Analysis.....	41
4.2.8    Bivariate Analysis.....	47
4.2.9    Multivariate Analysis.....	52
4.3    Preprocessing for the modeling.....	55
4.3.1    Label Encoding.....	55
4.3.2    Independent and dependent feature split.....	55
4.3.3    train and test split and Feature Scaling.....	55
4.4    Implementation.....	56
4.4.1    Boosting classifiers without balancing technique.....	56
4.4.2    Stacking classifier without balancing technique.....	57
4.4.3    Boosting and stacking classifiers with balancing technique.....	57
4.5    Model Interpretation.....	58
4.6    Summary.....	59
CHAPTER 5: RESULTS AND DISCUSSIONS.....	60
5.1    Introduction.....	60
5.2    Prediction result of Boosting classifiers before balancing technique.....	60
5.3    Prediction result of Stacking classifier before balancing technique.....	61
5.4    Prediction result after balancing technique.....	62
5.5    Evaluation of model on the test set.....	65
5.6    Result Analysis.....	67
5.7    Interpretation of the model result.....	68
5.8    Summary.....	72
CHAPTER 6: CONCLUSION AND RECOMMENDATIONS.....	73
6.1    Introduction.....	73
6.2    Conclusion and discussion.....	73
6.3    Contribution of the study.....	75
6.4    Recommendation.....	75
REFERENCES .....	76
APPENDIX A: RESEARCH PROPOSAL .....	81

## LIST OF TABLES

Table 2.3.1 Confusion matrix for credit risk.....	13
Table 2.6.1 Summarization of research studies.....	18
Table 3.2.1.1 Data Description of Lending Club.....	26
Table 3.2.2.1 Hardware and software requirements.....	27
Table 4.4.1.1 hyperparameter range for the boosting classifier.....	56
Table 5.2.1 confusion matrix of individual's train set of boosting classifier.....	59
Table 5.2.2 confusion matrix of joint's train set of boosting classifier.....	61
Table 5.3.1 confusion matrix of stacking classifier for the train set.....	62
Table 5.4.1 Confusion matrix of classifiers after balancing of individual's train set.....	62
Table 5.4.2 Confusion matrix of classifiers after balancing of joint's train set.....	64
Table 5.6.1 Comparison of performance.....	67
Table 5.7.1 Contribution of the top 10 features.....	68

## LIST OF FIGURES

Figure 2.2.2.1 Commonly used Advanced Classifier.....	9
Figure 2.2.2.2 Popular Boosting Classifier in Credit Risk Management.....	10
Figure 2.3.1 Commonly used Cost-sensitive Metrics.....	13
Figure 2.4.1 Distribution of class label in the dataset.....	15
Figure 2.5.1 Popular Explainable AI in Credit risk assessment.....	17
Figure 3.1.1 Workflow of Stacking Based explainable cost-sensitive model.....	25
Figure 3.2.3.1 Class labels for target feature.....	27
Figure 3.2.3.2 Class label distribution for application_type.....	28
Figure 3.2.5.1 Class label distribution for target feature.....	29
Figure 3.3.1 Stacking ensemble Architecture.....	32
Figure 3.3.2 Pseudocode for stacking Ensemble.....	33
Figure 4.2.1.1 Class labels for loan_status .....	36
Figure 4.2.4.1 Data distribution for annual income feature.....	38
Figure 4.2.5.1 skewed features for individual dataset.....	39
Figure 4.2.6.1 Correlation matrix of the loan amount for individual dataset.....	40
Figure 4.2.6.2 Correlation matrix of FICO score for individual dataset.....	40
Figure 4.2.7.1 Loan status data distribution.....	41
Figure 4.2.7.1.1 Analysis of loan_purpose for default account.....	41
Figure 4.2.7.2.1 Analysis of home_ownership for the default account.....	42
Figure 4.2.7.3.1 Analysis of an interest rate for the default account.....	43
Figure 4.2.7.3.1 Analysis of Last FICO score for the default account.....	43
Figure 4.2.7.5.1 Analysis of the usage of bank card account limit for the default account.....	44
Figure 4.2.7.6.1 Analysis of annual income for the default account.....	45
Figure 4.2.7.7.1 Analysis of debt to income ratio for the default account.....	46
Figure 4.2.7.8.1 Analysis of loan grades for the default account.....	46
Figure 4.2.8.1.1 Analysis of loan amount with loan purpose for the default.....	47
Figure 4.2.8.2.1 Analysis of avg loan amount with avg FICO score for default.....	48
Figure 4.2.8.3.1 Analysis of loan purpose with an interest rate for the default.....	48
Figure 4.2.8.4.1 Analysis of interest rate with loan term for the default.....	49
Figure 4.2.8.5.1 Analysis of loan term with instalments for the default.....	49
Figure 4.2.8.6.1. Analysis of annual income with FICO score for the default.....	50
Figure 4.2.8.7.1 Analysis of loan amount with dti for the default.....	51
Figure 4.2.9.1.1 Analysis of Avg loan amount with loan purpose for the default .....	52
Figure 4.2.9.2.1 Analysis of homeownership for the default and the non-default.....	53
Figure 4.2.9.3.1 Analysis of interest rate with FICO score for the default.....	54
Figure 5.5.1 Cost-sensitive metrics result for the individual test set.....	65
Figure 5.5.2 Cost-sensitive metrics result for the joint test set.....	66
Figure 5.7.1 Explainable graph of individual test's prediction result.....	70
Figure 5.7.2 Explainable graph of joint test's prediction result.....	71



## LIST OF ABBREVIATIONS

AI.....	Artificial Intelligence
ACC.....	Accuracy
ANN.....	Artificial Neural Network
AOD.....	Autoencoder Outlier Detection
AUC.....	Area Under Curve
AVG.....	Average
ADASYN.....	Adaptive Synthetic
CSDNN.....	Cost-Sensitive Deep Neural Network
CSDE.....	Cost-Sensitive Deep Neural Network Ensemble
CV.....	Cross Validation
DDR.....	Default Detection Rate
EBCA.....	Extended Balance Cascade approach
EDA.....	Exploratory Data Analysis
FPR.....	False Positive Rate
FP.....	False Positive
FN.....	False Negative
G-Mean.....	Geometric Mean
GBM.....	Gradient Boosting Machine
KS.....	Kolmogorov–Smirnov
LIME.....	Local Interpretable Model-agnostic Explanations
LightGBM.....	Light Gradient Boosting Machine
MC.....	Misclassification Cost
MLP.....	Multilayer perceptron
NN.....	Neural Network
ROUS.....	Random oversampling and under-sampling
ROC Curve.....	Receiver operating characteristic curve
SMOTE.....	Synthetic Minority Over-sampling Technique
SVM.....	Support Vector Machine
SHAP.....	SHapley Additive exPlanations
TPR.....	True Positive Rate
TP.....	True Positive
TN.....	True Negative
XAI.....	eXplainable Artificial Intelligence
XGBoost.....	eXtreme Gradient Boosting

## Chapter 1: Introduction

### 1.1 Background

Every financial firm all over the world does their business by sanctioning the loan to their customer. And every sanctioned loan has risks associated with it. If some loan defaults, it has a massive impact on the firm's profitability. 2008 Lehman brothers' bankruptcy is an excellent example of how loan default can cause the firm's insolvency (Malik, 2021). So, every loan proposal needs proper assessment, which can take time. The firm may lose its potential customer that also a huge loss for the firm. Here, the firm can use the benefit of machine learning, which helps develop a fast and accurate credit assessment system that improves the firm's profitability.

After introducing machine learning into the financial industry, it also faces many difficulties. For example, most classifier assumes that the dataset is balanced in the classification problem. The balanced dataset is where the class label of the target feature is distributed equally or almost equally. However, the financial dataset is imbalanced because default accounts are much less than non-default accounts. So, in this case, the machine learning model faces a misclassification problem. That means the model is always biased towards the majority class (non-default), and the model may predict some default labels as non-default. This will impact the business of the firm. So, need for a cost-sensitive model is the new essential requirement, which can evaluate the model performance not only by the accuracy but some misclassification cost metrics should also be introduced. These metrics are True positive rate, false-positive rate, True negative rate, and False-negative rate.

A robust model is also required with the evaluation metrics, reducing the misclassification cost and improving model performance. Moreover, this can be achieved by using two approaches:

1. Use of classifiers that are sensitive to the imbalanced dataset. These classifiers are known as a cost-sensitive classifiers.
2. Use of resampling technique to balance the dataset. Resampling can add a new sample to the minority class to balance with the majority class or delete the samples from the majority class to balance with the minority class.

This cost-sensitive approach will be part of this research study, and various state of the art will be reviewed and analysed.

After the significant advancement of artificial intelligence in individuals' lives, many regularities now imply various regulations for individual rights. If the decision of any machine learning algorithm significantly affects the individual, then it is their right to know the reason behind the decision. Under the equal credit opportunity act Code of Federal Regulations(eCFR :: 12 CFR 1002.9 -- Notifications., 2022) instruct all credit bureau of the United States of America that the customer should be notified if their application is denied with the list of reasons and proper explanation associated with the factors affect their application for the credit. Even under General Data Protection Regulation (GDPR) (EUR-Lex - 32016R0679 - EN - EUR-Lex, 2022), European Union law implied the right to explanation for transparency and interpretability of the decision made.

Now that the responsibility of machine learning is enhanced, it should be accurate and explainable in human terms. However, the machine learning model's nature is black-box. This means no model can explain which features have more importance to predict the probability. Hence need of additional technique is required to make the prediction result transparent. Nevertheless, some research studies (Ribeiro et al., 2016; Lundberg et al., 2017) explore some explainable AI that will be reviewed in this research study.

## 1.2 Problem Statement

The cost-sensitive learning aims to evaluate the machine learning model performance using additional misclassification cost metrics. Because of the imbalanced dataset, the accuracy of the metrics may result in high, but their misclassification cost may also be high in this case. The proposed model is not case-sensitive. Therefore, to calculate additional cost metrics confusion matrix (Feng et al., 2018; Yotsawat et al., 2021a) will be needed. It is a table that holds True positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). AUC score is the most commonly used metric to evaluate the classifier performance for identifying the correct class labels. Other than AUC score F-measure, G-mean, KS (Kolmogorov–Smirnov) (He et al., 2018; Biecek et al., 2021; Dzik-Walczak and Heba, 2021; Shen et al., 2021; Zhang et al., 2021) is the next most used cost-sensitive metrics for model evaluation. These metrics are very well formulated to calculate the misclassification cost for cost-sensitive learning. But are these evaluation metrics enough for cost-sensitive learning? The answer is No, and these metrics only can calculate the cost. However, additional steps need to be taken to reduce the misclassification cost. This can be done by using cost-sensitive learning or resampling techniques or a combination of both.

(Hamori et al., 2018) proposed a comparative analysis of 11 advanced classifiers. These are bagging, boosting, random forest, 4 Neural-networks based classifiers, and four deep neural network-based classifiers. Moreover, the performance of these classifiers was evaluated using the AUC score and F-Score. And the study showed that boosting outperformed all the individual classifiers. (Chengeta and Mabika, 2021a) compared the performance of advanced boosting classifier LightGBM and CatBoost with XGBoost, Convolutional neural network (CNN), k-NN, SVM, and decision tree. All boosting models performed better than the traditional classifier as well as their performance is almost similar to CNN. The ROC was highest for 99.76% LightGbm and Recall, True positive rate, and False positive rate is highest for CatBoost. Hence boosting algorithm is the best choice for cost-sensitive learning. However, these models are still not explainable.

These studies also proved that the classifier combination performed better than the single classifier. (Li et al., 2018; Shen et al., 2021) proposed heterogeneous ensemble of neural network with boosting classifier like AdaBoost, XGBoost improved the neural network's performance for misclassification cost metrics. (Song and Peng, 2019; Kun et al., 2020a; Shen et al., 2021; Yotsawat et al., 2021a) These studies add the resampling technique into consideration, and the result shows that the performance of each classifier significantly improved. For resampling, widely used techniques are Random under-sampling (RUS), Random over sampling (ROS), and Synthetic Minority Oversampling Technique (SMOTE).

They have their own pros and cons, but they improved the performance for cost-sensitive metrics, (Faris et al., 2020) study analysed the performance of various base classifiers and the ensemble of multiple base classifiers with or without sampling (SMOTE) method. The result showed that the performance (G-mean) of the neural network classifier (MLP) is improved from 42.7% to 63.9%. Performance (G-mean) of the ensemble of AdaBoost with Random tree (multi classifier) improved from 69% to 77.3%, and an ensemble of bagging with random tree improved from 49.1% to 69.9%. The resampling method improved the performance classifier for misclassification metrics, but these studies still failed to make the model explainable.

Some researchers (Bussmann et al., 2021; Dastile and Celik, 2021; Egan, 2021; Hadji Misheva et al., 2021; Moscato et al., 2021) explored the explainable AI (XAI) in their research studies. SHAP, LIME, and most commonly used XAI among researchers other XAI like Anchors, LORE, BEEF, Nearest-Unlike-Neighbour performance is also compared with them, but SHAP performance was better. All these XAI studies (Bussmann et al., 2021; Hadji Misheva et al., 2021; Moscato et al., 2021) compare the performance using misclassification cost for boosting classifier and some neural network classifier as well, and they made the model interpretable in terms of human. In the end, some features with rank were shown to have a high impact on the prediction result. (Moscato et al., 2021) this study showed the comparative analysis of some benchmark classifier (LR, RF, MLP) and resampling techniques (SMOTE, RUS, ROS, ADASYN, SMOTE-TOKEN, SMOTE-ENN ) with XAI, and their performance was evaluated using cost-sensitive metrics (AUC, TPR, TNR, FP-Rate, G-Mean, ACC). However, the performance of XAI with resampling technique based on multi-base ensemble classifier is yet to be explored.

Boosting ensemble LightGBM, CatBoost and XGBoost are outperformed in case of imbalanced data even their performance improved using the resampling technique. (He et al., 2018; Zhang et al., 2019, 2021; Kun et al., 2020a; Zhang and Li, 2021) Showed that the stacking ensemble really improved the model performance, and multi-classifiers with XAI are very interesting to explore.

Hence, in this study will analyze an explainable AI using a stacking ensemble of boosting classifiers based on cost-sensitive learning. The performance will be compared to the single benchmark classifier XGBoost, LightGBM, and CatBoost.

### **1.3. Research Question**

The focus of this research study is to improve the model's performance to reduce the misclassification caused due to data imbalance problem. And another important aspect of the research study is to explain the prediction result to the applicant. However, because of the black-box nature of the classifiers, the classification results are not self-explanatory. So, based on these facts following two research questions are formulated:

- Will oversampling technique improve the performance of the model in case of data imbalance?
- Will the result of the black-box model be interpretable?

### **1.4. Aim and Objectives**

This research study aims to find an optimal cost-sensitive model with interpretability to reduce the misclassification cost for a highly imbalanced dataset. The goal is to design a robust and trustworthy credit risk assessment application that helps the firm improve its business with a transparent decision process.

Following research objectives are formulated based on the research questions suggested above:

- To analyze the ensemble classifier that can handle the imbalanced data and reduce the misclassification cost.
- To find an optimal resampling technique to balance the majority (non-default) and minority (Default) class label for the financial dataset.
- To analyse additional misclassification cost metrics for cost-sensitive learning to evaluate the model performance.
- To suggest an explainable AI method to overcome the black-box nature of the classifier to make it more interpretable.

### **1.5. Significance of the Study**

This research study aims to design a robust and transparent cost-sensitive learning model that will outperform other benchmark models for the misclassification cost metrics. Transparency will be achieved by using the XAI model to help the model overcome its black-box nature, and it will explain the reason behind the risk probability result.

Multiple research studies (Hamori et al., 2018; Li et al., 2018; Faris et al., 2020; Barua et al., 2021; Egan, 2021; Hadji Misheva et al., 2021; Yotsawat et al., 2021a; Zhang and Li, 2021) showed boosting algorithm performed better in cost-sensitive learning. Their misclassification cost was reduced, and G-mean, AUC, F-measure were the highest. Moreover, they performed better as a single classifier and multi-base classifier. Even The neural network (Feng et al., 2018; Song and Peng, 2019; Faris et al., 2020) could not compete with them. This is may be because of their insensitivity towards imbalanced data. But NN performance improved (Li et al., 2018; Wong et al., 2020; Shen et al., 2021) when they worked with the combination of boosting classifiers (AdaBoost, XGBoost) in the heterogenous ensemble. So, boosting is the best choice for cost-sensitive learning.

The resampling technique also helps the classifier achieve their highest performance (Moscato et al., 2021) in case of data imbalance. In multiple studies, SMOTE (oversampling) proved its superiority with the boosting classifier.

This study will design a machine learning pipeline based on an ensemble of boosting classifiers (XGBoost, LightGBM, and CatBoost) with resampling method (SMOTE and ADAYSN) and explainable ai (SHAP), which make the cost-sensitive learning model more robust and interpretable, which will help the firm to grow its business while gaining the customer trust as well.

### **1.6. Scope of the Study**

This research study will incorporate popular cost-sensitive metrics (AUC, G-mean, F-measure, Precision, and Recall) to evaluate the stacked ensemble of boosting (XGBoost, LightGBM, and CatBoost) classifier. In addition, an oversampling method (SMOTE) will be used to improve the cost-sensitive metrics, and an XAI will be added to the final model to make the model explainable in terms of human terms.

Feature selection or feature extraction will not be part of this study because the proposed classifier ensemble will be examined for the high dimensional data. For XAI, all features will be needed to find their impact on the prediction result. The neural network is also not part of this research study. If the model performs well, future work using a heterogeneous ensemble of stacking with NN will be next to be explored.

## 1.7 Structure of the study

The structure of this research study is as follows:

**Chapter 1** presents an introduction and background study about the credit risk assessment model. Its need and benefits were discussed in section 1.1. Section 1.2 explained the problem statement associated with the credit risk model. Research questions were formulated after discussing the problem statement in section 1.3. Aims and objectives were discussed in section 1.4 based on the research question. And section 1.5 and 1.6 explained the significance and scope of this research study.

**Chapter 2** presents a detailed literature review of the credit risk prediction model studies. Section 2.2 will discuss the scope and challenges of the multiple classifiers used in the credit risk model. In section 2.3, various cost metrics will be explored and identified to evaluate the model's performance to reduce the misclassification cost. Section 2.4 will discuss the balancing technique and how it helps improve the classification model's performance. And finally, section 2.5 explains the model explainability and how it will help interpret the black-box model prediction result.

**Chapter 3** presents the workflow of the methodology for this research study. Section 3.2 explains how the dataset is acquired for this research and the important features with data description. Then, under subsections 3.2.2 and 3.2.3, explain the steps considered for the data preprocessing and transformation. Subsection 3.2.4 discusses the oversampling method, and 3.2.5 discusses cost-sensitive metrics. And explainable AI methodology is discussed under subsection 3.2.6. Finally, section 3.3 explains the proposed ensemble classification model.

**Chapter 4** presents the analysis and design of the methodology discussed in chapter 3. First, all the implemented EDA steps are discussed in section 4.2. data cleaning steps, handling null values, outliers, skewness, and redundant features are discussed under subsections 4.2.3, 4.2.4, 4.2.5, and 4.2.6. After that, univariate, bivariate, and multivariate analyses are discussed under subsections 4.2.7, 4.2.8, and 4.2.9. preprocessing steps label encoding, train-test split, and feature scaling are explained under section 4.3. Next, section 4.4 explains the model implementation of ensemble classifier for train set based on parameter tuning balancing technique. Finally, section 4.5 presents the interpretation step of the prediction result.

**Chapter 5** presents the analysis of the prediction and interpretation result. Section 5.2 analyses the boosting classifier's prediction result before the balancing techniques and section 5.3 analyses the prediction result of the stacking classifier before the balancing technique. Under section 5.4, the prediction result of ensemble classifiers is analysed after implementing the oversampling method (SMOTE and ADASYN). In section 5.5 performance of the classifiers is evaluated on the test set using cost-sensitive metrics. And finally, the interpretation result is discussed in section 5.6.

**Chapter 6** presents the conclusion of the overall study and important findings of the study will be discussed. Contribution of the study will be explained in section 6.3. And section 6.4 will explain the recommendation and future scope of the study.

## Chapter 2: Literature Review

### 2.1 Introduction

Thousands of paper was published on credit risk management. Many models were proposed with lots of scope in this area. With so much competition in the financial industry, every firm needs a vigorous risk assessment model with the highest accuracy. For the firm, accuracy means if the proposal is non-default and misclassified as fraudulent, it loses its profit, and if the fraud is misclassified as non-fraudulent, the firm will face a heavy loss. In both, the scenario misclassification is not profitable for the firm. So, they need a robust risk assessment model with low misclassification cost, not just high accuracy.

A machine learning model is evaluated against various metrics, and accuracy is the first. However, accuracy alone can not evaluate any model for misclassification because most of the model assumes that the dataset is balanced for classification problem. A balanced dataset means the distribution of the target variable label is distributed equally or almost equally. Nevertheless, that is not always true for a real-world dataset; they are highly imbalanced. Likewise, the Lending Club dataset has almost 48% Fully paid data and only approximately 12% data as Default. In this case, most models will be biased towards the majority class, which means default proposals are classified as non-default. Therefore, the firm will face a loss; besides the model's reasonable accuracy rate, the result is not suitable for the firm. So, the model needs to be cost-sensitive towards misclassification, and for that with accuracy metrics, additional evaluation metrics will be considered in this research paper.

Additional cost-sensitive metrics will help evaluate the model. First, however, some balancing technique is needed to reduce misclassification costs, which helps balance majority and minority classes. The basic idea behind the balancing technique is first to recreate samples for minority class to balance with majority class or reduce the data of majority class to match with minority class. Each approach has its pros and cons discussed in the next section.

Financial data is high in volume and dimension. So, need for a strong classifier that handles such an imbalanced dataset with high dimensionality. Various tree-based, neural network, and deep learning classifiers have already been proposed, and there are still lots of discussion for better performance. This research paper will discuss the various state of the art for credit risk assessments further in the literature review.

As already mentioned, many financial regularities obligate banks under the right to explanation to explain the reason behind the rejection of their loan proposal. So interpretability becomes an essential requirement for the machine learning pipeline for credit risk assessment. The most interpretable model for classification problems is Logistic Regression and Decision Tree. However, other advanced models are proposed that outperformed these models like Random Forest (RF), GBDT (Gradient Boosting Decision Tree), Deep Learning, Neural Network, and others. However, these models are Black-Box in nature; there are no transparency means which feature has more impact on default or non-default it can not be explained. So, the machine learning model can be explainable by using additional explainable AI techniques to solve this issue. SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) are popularly used.



## 2.2 Machine Learning in Credit Risk Assessment

The traditional financial industry reviews every loan proposal manually by reviewing the documents, credit score, annual income, and other factors. However, some of the loans still become the default. The review process also takes one or more days to decide whether to accept the proposal or not. Meanwhile, a good proposal may get missed, and other competitors benefit from that. Here the role of machine learning becomes very prominent to automate the process and make it more convenient, faster, and transparent than the traditional method. The most commonly used classification models for credit risk assessment are:

### 2.2.1 Traditional Classifier:

**Logistic regression** is the most interpretable and straightforward classifier, but its performance with imbalanced data is not worth it. Some researchers used the benefit of LR with the combination of other advanced models like neural networks (Dzik-Walczak and Heba, 2021). The proposed model reduces the potential risk by minimizing the misclassification cost. However, they suggest that the model performance may improve by introducing the ensemble technique. LR performed better when a heterogeneous ensemble framework (Li et al., 2018) was formed using XGBoost and a deep neural network. The framework overcame the LR difference by using XGBoost and a Deep neural network to handle imbalanced data using hyperparameter tuning. However, misclassification of cost metrics is not part of the research.

The other traditional classifier commonly used in credit risk prediction is **SVM** (Support Vector Machine). It is not transparent like LR, and its training time is relatively high, but its performance with cost-sensitive metrics is quite better (Song and Peng, 2019). They proposed an ensemble framework using four base classifiers, SVM, Decision Tree, Logistic Regression, and multilayer perceptron (MLP), combining sampling techniques. The result is better than the single classifier. (Hadi Misheva et al., 2021) the paper proposed a comparative analysis of an Explainable AI with SVM and other classifiers (LR, RF, XGBoost, and Neural Network). They concluded that the SVM performance is lower than the other because of the overfitting.

**Decision Tree** is another traditional and widely used classifier in the credit risk prediction field. DT has good stability and interpretability even for the high dimensional dataset, but it may bias the majority class for the imbalanced dataset. The research (Wong et al., 2020) shows the comparative analysis of SVM, DT, LR, Boosting, Bagging, Neural Network. The result shows SVM and DT are underperforming classifiers for the imbalanced dataset. On the other hand, some research (Feng et al., 2018) shows that using SVM, DT, or Neural networks as individual classifiers may underperform against cost-sensitive metrics. However, their performance improved when used in a heterogeneous ensemble framework.

So overall, the traditional classifier has its benefits but underperformed in the case of imbalanced datasets and cost-sensitive analysis.

### 2.2.2 Advanced Classifier

It has a broad category with many proposed classifiers which consistently outperform the traditional classifier. Based on the many research studies, figure 2.2.2.1 shows the categorization of the advanced classifier proposed in credit risk assessment.

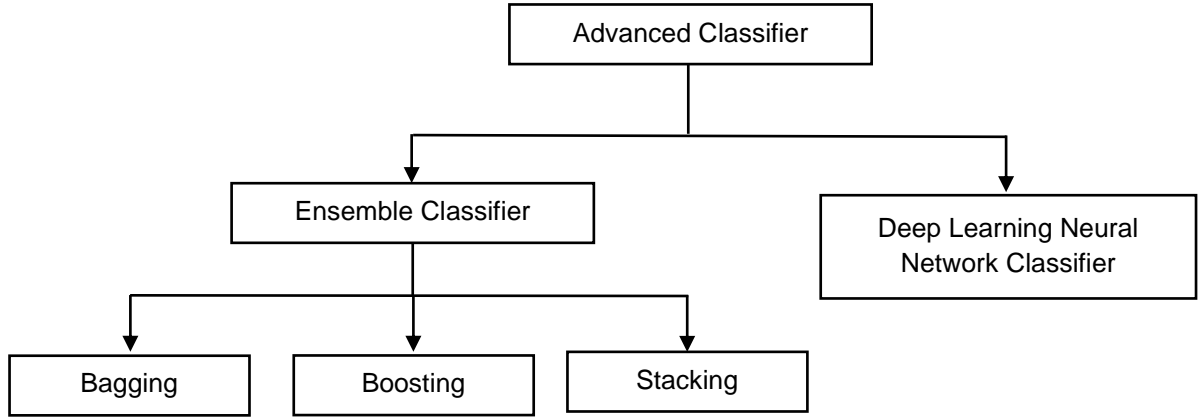


Figure 2.2.2.1 Commonly used Advanced Classifier

(Hamori et al., 2018) compare the performance of the 11 most used advanced classifiers for risk assessment based on random forest, bagging, boosting, neural network, and deep neural network. They concluded that boosting outperformed other methods because of the difficulty with hyperparameter tuning for neural networks.

**Ensemble Classifier** thrives in the financial industry. It has the most diversified nature but stable predictive power, making it more widespread. Moreover, this framework handles the high dimensionality with unbalancing and can provide better cost-sensitive results. However, this framework is black-box in nature.

**Random Forest Classifier** is one of the examples of an ensemble bagging framework. It is the combination of multiple decision trees. It handles missing values and outliers effectively, but this classifier's main disadvantage is overfitting. (Hamori et al., 2018) shows random forest has the highest standard deviation in training and test data result because of the overfitting. (Kun et al., 2020a; Barua et al., 2021; Egan, 2021; Zhang and Li, 2021) demonstrated that random forest underperforms as an individual classifier than another ensemble classifier but better than the traditional one. However, the overall performance improved significantly when RF was used in stacking (Zhang and Li, 2021) or heterogeneous (He et al., 2018) or homogeneous framework. (Sandica and Fratila, 2021) proposed a cost-sensitive regime-based credit assessment strategy using boosting and random forest classifier where the random forest proved its superiority over the other classifiers.

**Bagging (Bootstrap Aggregating) Classifier** is another excellent example of an ensemble classifier. It predicts the model's performance by building a model for a randomly taken subset of the train dataset and then aggregating their prediction result to calculate the final prediction. (Hamori et al., 2018) proposed a comparative analysis among 11 classifiers using random forest, bagging, boosting, neural network, deep learning. Moreover, the result shows accuracy rate for bagging and random forest for test data is less than 60% they both underperformed. (Wong et al., 2020) studied the

cost-sensitive model to handle the imbalanced dataset. They have used bagging for resampling to handle class imbalance, which improves the overall accuracy, but this method is biased towards the majority class, increasing the misclassification cost. (Feng et al., 2018; Yotsawat et al., 2021a) proposed different variants for bagging like DT-Bagging, Bagging-SVM NN-Bagging, NN-Bagging-SMOTE, NN-Bagging-RUS, and NN-Bagging (1:5) to build a cost-sensitive credit risk management model. These models are not given the best result against the cost-sensitive metrics but significantly improved compared to other bagging performances. (Faris et al., 2020) this research study concluded that if bagging resampling technique when used with balancing technique like SMOTE its performance improved.

So, Bagging reduces the overfitting and variation, but when the dataset is imbalanced, it is biased towards the majority class, increasing the misclassification cost.

**Boosting Classifier** is the most popular ensemble classifier, which works on multiple weak classifiers to create a robust classifier. After creating the model for the training dataset, keep adding a new model over the previous model to correct the errors until the prediction result is correct or the maximum number of models are added.

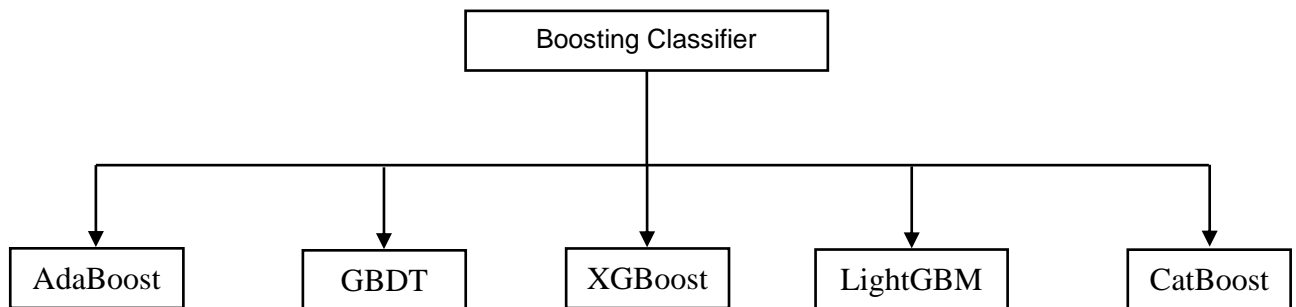


Figure 2.2.2.2 Popular Boosting Classifier in Credit Risk Management

The main aim of the boosting classifier is to minimize training errors. Hence this classifier is sensitive towards misclassification.

**AdaBoost (Adaptive Boosting)** is the first boosting classifier based on a decision tree to boost its performance. AdaBoost performance is not good with the high dimensional dataset or imbalanced dataset; hence the researchers (He et al., 2018; Kun et al., 2020a; Yotsawat et al., 2021a; Zhang et al., 2021) conclude that AdaBoost accuracy is not promising when it is implemented as a single classifier. (Faris et al., 2020; Wong et al., 2020) these studies' combination of SMOTE (oversampling technique) and AdaBoost achieved good accuracy of about 98%. Furthermore, the performance of a new variant of AdaBoost with a cost function named AdaCost gives accuracy and a True Positive rate equivalent to a cost-sensitive neural-network-based model of more than 60%. (Shen et al., 2021) proposed an integrated model of AdaBoost and LSTM neural network classifier which reduces the overfitting issue and received more than 70% AUC score and Kolmogorov–Smirnov statistic (KS) is 48%.

**GBDT (Gradient Boosting Decision Tree)** this classifier is similar to AdaBoost, but it provides a more optimal solution using gradient function while adjusting the weak learners with a strong classifier. However, this classifier is prone to overfitting, and

tuning the hyperparameter is very difficult and time-consuming, but once the result is obtained its very optimal hence lots of researchers used this classifier as a benchmark model(He et al., 2018; Xia et al., 2020; Zhang and Li, 2021; Zhang et al., 2021).

**XGBoost (eXtreme Gradient Boosting)** is an advanced implementation of GBDT where the optimal solution can be achieved by using fast hyperparameter tuning within less training time. It has the same sequential process to combine the weak learners to strong learners like GBDT and XGBoost handles imbalanced data and reduces the chance of overfitting. This is the most popularly used classifier in credit risk management. (Wu et al., 2020a) implemented the multilayer cascaded model of XGBoost to deal with the imbalanced dataset, calculate the misclassification cost, and achieve the outstanding accuracy of 99.97% with an outstanding F1-score of 99.98%. Because of its fast computation time, it is also popular in (Bussmann et al., 2021) explainable machine learning modeling.

**LightGBM** and **CatBoost** are very similar to XGBoost in terms of performance, except LightGBM support gradient-based one-side sampling and CatBoost supports Minimal Variance Sampling. This makes these boosting classifiers perform better with imbalanced datasets than XGBoost and improve the misclassification cost.

(Ma et al., 2018; Xia et al., 2020; Chengeta and Mabika, 2021a; Egan, 2021; Zhang and Li, 2021) show the comparative performance analysis where when the dataset is highly imbalanced, LightGBM outperformed the XGBoost, and CatBoost outperformed all the classifiers. However, This classifier is the most favourite for the cost-sensitive model.

(Li et al., 2021) implement an early default using LightGBM without using any sampling technique still the result is better than 70% AUC for the cost-sensitive model.

### **Stacking Classifier**

**Deep Learning Neural Network Classifier** Structure of neural is inspired by the nervous system of the human brain, where brain neurons are interconnected and working in parallel processing. The architecture of a neural network has three layers: input, output, and hidden layer, which makes the structure a little complex because of which it is hard to explain the process and trace out the error.

(Hamori et al., 2018) demonstrate how boosting (AdaBoost) classifier over power the deep neural network classifier where DNN underperforms with less than 70% accuracy for test data set with the high std deviation. The study shows this may be because of the difficulty of choosing an appropriate hyper parameter for the complex structure of the neural network.(Dzik-Walczak and Heba, 2021) They tested the hypothesis performance of neural network over the logistic regression against many costs' sensitive metrics like Gini index, KS (Kolmogorov-Smirnov), and AUC score. The proposed Logistic regression with WOE (Weight of Evidence) performs better than the Multilayer perceptron (MLP) neural network. However, the ensemble combination of each model proves its superiority by a slight improvement in the AUC score, Gini Index, and KS.

(Hadi Misheva et al., 2021) studies the performance of multiple classifier LR, XGBoost, RF, SVM, and neural network with explainable ai technique SHAP and

LIME. Accuracy for NN is the highest, but other XGBoost perform better in precision and F1-score. This concludes that NN may be susceptible to overfitting. (Moscato et al., 2021) even the combination of Sampling techniques like SMOTE or RUS with MLP underperforms against the cost-sensitive metrics like G-mean and False positive rate. (Kun et al., 2020a) Neural networks do not provide the promising cost-sensitive result as a single classifier. However, (Li et al., 2018) heterogeneous model using XGBoost, LR, and Deep neural network reached up to the 78.91% AUC score. (Shen et al., 2021) the study concludes promising results with more than 70% AUC score and more than 40% KS score for an ensemble of deep learning-based LSTM (long-short-term-memory) models with AdaBoost.

A Convolutional Neural Network (CNN) based research was proposed in the study (Dastile and Celik, 2021) and (Chengeta and Mabika, 2021a). The CNN has an additional multilayer of convolution and filters. This method is commonly used in image recognition, but these two studies show the use of CNN in credit risk management and compare their value with another benchmark classifier. (Chengeta and Mabika, 2021a) conclude CatBoost and LightGBM outperform the CNN for misclassification cost and AUC score. (Dastile and Celik, 2021) this study uses 2-D CNN with explainable AI, and with SHAP, the result is better.

(Oreški and Oreški, 2018; Wong et al., 2020; Yotsawat et al., 2021a) Demonstrate cost-sensitive neural network designed explicitly for imbalanced dataset to improve the model performance for misclassification cost. Overall, the CS-NN-based model performs better than other neural network-based models. For example, the overall AUC score is more than 70%, and other metrics like True positive rate, False positive rate, G-mean give promising results for imbalanced datasets for CS-NN.

### 2.3 Cost-sensitive learning in credit risk assessment

Because of the imbalanced dataset, the machine learning model does not always perform at its best because most models assume the class label distribution is almost balanced. This issue causes misclassification of data means the model becomes biased towards the majority class. In this scenario, cost-sensitive learning (CSL) plays a significant role. The key feature of cost-sensitive learning is the consideration of additional performance measures misclassification metrics. Therefore, the goal of the CSL is not only to measure the performance of the model using accuracy but also to consider additional misclassification error measures.

(Feng et al., 2018; Yotsawat et al., 2021a) Misclassification errors can be classified as type I and type II errors. Furthermore, for credit risk assessment, these errors are defined as:

**Type I error** is misclassifying the default data as non-default.

**Type II error** is misclassifying the non-default as default.

In both, the case firm will face a heavy loss in business. So, CSL helps measure the machine learning model performance for these two errors to reduce it.

The classification problem confusion matrix is the best way to calculate these type I and type II errors. The structure of the confusion matrix for credit risk assessment is demonstrated in table 2.3.1. where True Positive (TP) indicates the number of actual non-default data predicted

Table 2.3.1 Confusion matrix for credit risk

Predicted Labels	Actual Labels	
	Non-Default	Default
Non-Default	True Positive (TP)	False Negative (FN)
Default	False Positive (FP)	True Negative (TN)

Correctly and True Negative (TN) number of actual default label predicted correctly. The misclassification error focus on False Positive (FP) and False Negative (FN). So, type I and type II error can be calculated as (Feng et al., 2018) :

$$Type\ I\ error = \frac{FP}{FP + TN} \quad (2.3.1)$$

$$Type\ II\ error = \frac{FN}{TP + FN} \quad (2.3.2)$$

The most research study used this confusion matrix for their cost-sensitive learning model. The most commonly used metrics for calculating the misclassification error are explained in figure 2.3.1.

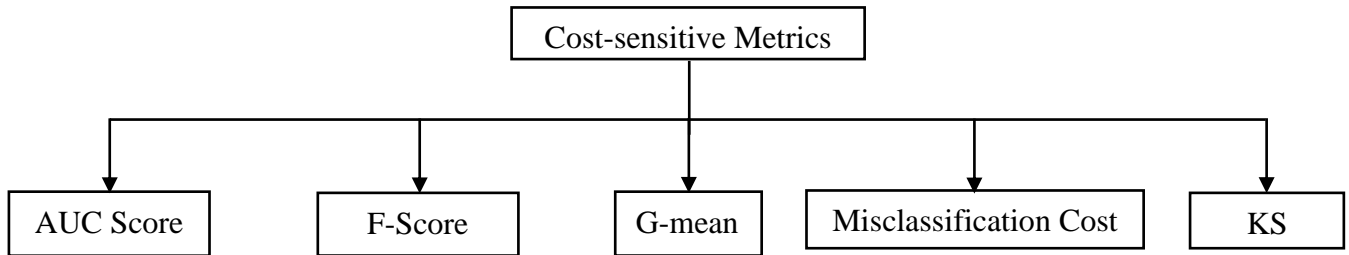


Figure 2.3.1 Commonly used Cost-sensitive Metrics

The model's accuracy determines how well the model performs for all class labels. However, it can not measure the misclassification of class labels. The alternative of accuracy for classification problems is the AUC score. **AUC** (Area under the curve) measures how well the model correctly classified the class label. This measure is used in every classification model.

**F-Score** is the second most popular measure used in classification problems (He et al., 2018; Oreški and Oreški, 2018; Shen et al., 2019; Song and Peng, 2019; Jin et al., 2021; Pes and Lai, 2021; Zhang et al., 2021). It is a weighted mean of precision and recall, making this measure more perfect for calculating the misclassification cost for imbalanced data. Higher the F-score lower the misclassification cost. The most cost-sensitive based model used this measure to evaluate the model.

(He et al., 2018; Shen et al., 2019; Song and Peng, 2019; Zhang et al., 2019; Faris et al., 2020; Pes and Lai, 2021; Yotsawat et al., 2021a) The next most popular measure G-mean (geometric

mean), measures the balance of the classification performance using specificity and sensitivity. Higher the G-mean better the model performance.

(Feng et al., 2018; Yotsawat et al., 2021a) Proposed **misclassification cost** metrics. This measure calculates the actual misclassification error cost by using the below formula:

$$\text{Misclassification Cost} = \frac{FN}{TP + FN} * P(0) * 5 + \frac{FP}{(FP + TN)} * P(1) * 1 \quad (2.3.3)$$

Where P(0) and P(1) are the probability of non-default and default, and the studies (Feng et al., 2018; Yotsawat et al., 2021a) proposed a weightage cost to non-default and default to 5 and 1, respectively.

**KS (Kolmogorov-Smirnov)** is also a very important metric used in cost-sensitive learning (He et al., 2018; Biecek et al., 2021; Dzik-Walczak and Heba, 2021; Shen et al., 2021; Zhang et al., 2021). It measures the ability of the model to distinguish between the class labels. Its values vary between 1 to 100. If KS is 100 means all class labels are predicted perfectly by the model.

## 2.4 Role of Balancing Technique for Imbalanced Data

In classification problems, the goal is to predict the class label for the target feature. However, in the case of imbalanced data, the classifier's performance is a big issue. In the last section, various cost metrics have been discussed to evaluate the model performance. This section will discuss techniques through which this imbalancing issue can be improved. The ratio of default cases is much less than the non-default in the financial dataset. The lending club dataset used in this study has the target feature 'loan\_status'. Figure 2.4.1 shows the label's distribution for the target feature, where the class label Fully Paid is non-default, which holds almost 47.6% of data, and the default class label Charged Off and Default together hold only 12% data. This vast difference between class labels causes data imbalancing.

Fully Paid	47.629771
Current	38.852100
Charged Off	11.879630
Late (31-120 days)	0.949587
In Grace Period	0.373164
Late (16-30 days)	0.192377
Does not meet the credit policy. Status:Fully Paid	0.087939
Does not meet the credit policy. Status:Charged Off	0.033663
Default	0.001769

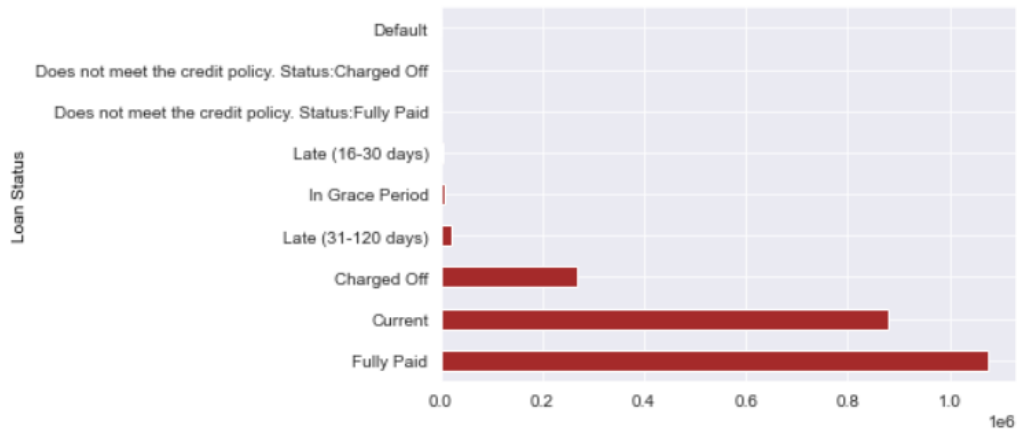


Figure 2.4.1 Distribution of class label in the dataset

So, whenever any model is implemented on such an imbalanced dataset, they may be biased towards the majority class labels that are non-default. The result will be that the default data may be classified as non-default, and the firm will face heavy loss besides the highest accuracy rate of the model. (García et al., 2019) study analysis the performance of ensemble model using bagging, boosting (AdaBoost), Random Forest, and DECORATE (Diverse Ensemble Creation by Oppositional Relabelling of Artificial Training Example) for 14 real financial datasets. Moreover, the result shows that model configuration depends on sample type.

Most models are biased towards the positive class (majority class). (Faris et al., 2020) this study concludes that after using the sampling technique to balance the dataset, the accuracy of all base classifiers (random tree, J48, Rep-Tree, Naïve Bayes, MLP, k-NN) and ensemble classifier (Bagging, AdaBoost, Random Forest) and DECORATE achieved very high accuracy rate, even the other misclassification measure True positive rate, False positive rate, and G-mean also improved as the same rate of accuracy.

The balancing resampling technique is used, which works on the data level before implementing the model to handle the imbalanced issue. It works in three ways:

- **Undersampling**, this resampling approach works on the majority class. It keeps all samples from minority class and decreases the size of majority class to balance the ratio with minority class. In this process majority class losses lots of potential information.
- **Oversampling**, this approach is the opposite of undersampling by keeping all data samples for the majority class and recreating new samples for the minority class to balance the ratio with the majority class. However, this process takes time, and it may cause overfitting.
- **Hybrid Sampling** this process is the combination of oversampling and undersampling.

Every resampling technique has its own pros and cons, and the researcher has used all three for credit risk assessment. (Namvar et al., 2018) demonstrate a comparative study of the combination of various classifiers (Logistic Regression, Linear Discriminate Analysis, and Random Forest) with undersampling technique (RUS, IHT), oversampling technique (ROS, SMOTE, ADASYN), and hybrid technique (SMOTE-TOMEK, SMOTE-ENN). As a result, RF with RUS gives the highest G-mean, about 65%, and AUC score about 69%. LDA with SMOTE is the next best-performing model.

(Chi et al., 2020; Niu et al., 2020; Jin et al., 2021; Zhang et al., 2021) These studies show how the undersampling method helps the classifier achieve the best performance by deleting the



sample from the majority class. The first paper proposed a bagging-based under-sampling method, and all base classifiers achieved their best AUC score, F-measure, and KS metrics. The performance result, compared with the stacking ensemble-based model and sampling-based, gives a better result. Hence sampling technique with the stacking-based model is an excellent idea to be considered. (Chi et al., 2020) studied a novel undersampling method TRainable Undersampling with Self Training (TRUST), which results in an AUC score of up to 87% for the proposed model. The proposed method works well on the small dimensional dataset.

For a cost-sensitive model oversampling technique is very acceptable. (Oreški and Oreški, 2018) explored the impact of the resampling method in a combination of hybrid genetic algorithm and neural networks (HGA-NN) classifier. The random oversampling (ROS) achieved a better result than the synthetic minority oversampling technique with an AUC score of 79% and 65% F-score. The next cost-sensitive approach (Stern et al., 2021) analyzed the comparative study of the oversampling method with a threshold (Theoretical thresholding and Empirical thresholding) approach and cost-sensitive method, Metacost. Moreover, the study shows that the true positive rate for SMOTE with base classifier (Random Forest, Gradient Boosting, and Logistic Regression) is better.

(Shen et al., 2019) proposed a novel ensemble model based on back propagation neural network (BP-NN) and AdaBoost. This model used particle swarm optimization (PSO) to optimize the weak learner. The oversampling method SMOTE is also used, which improved the overall performance for two imbalanced datasets, and G-mean and F-score for the proposed model achieved more than 90%. Even the true positive rate improved with a 91% AUC score.

SMOTE is performed well in a multi-classifier-based model, for the single classifier (Niu et al., 2020), its performance is not up to the benchmark. (Yotsawat et al., 2021a) examined the various single classifier (SVM, LR, k-NN, LDA, DT, NN) with random undersampling and SMOTE and cost-sensitive neural network ensemble (CS-NNE). The CS-NNE outperformed all the classifiers, but RUS is superior to the SMOTE using a single classifier.

Another heterogenous ensemble model (Shen et al., 2021) based on long-short-term-memory (LSTM) neural network and AdaBoost was studied with an improved oversampling SMOTE method. The improved SMOTE idea was to select the appropriate original minority class sample rather than considering each minority class as a sample. It would remove the noisy data and reduce the newly created synthetic sample variance. Furthermore, the improved SMOTE would help identify the inter-correlation among the features, making the model more robust and accurate. The proposed model achieved an 80% AUC score for one imbalanced dataset and 75% for the second dataset. Even the KS score showed a better result.

(Song and Peng, 2019) performed an experimental analysis to evaluate the model by combining the ensemble method with preprocessing balancing technique to handle the imbalanced dataset. Under-Bagging and SMOTE-Boosting are introduced; performance would analyze with various base classifiers (SVM, MLP, LR, C4.5) and compared with random undersampling and SMOTE method. The performance was measured with the accuracy of six cost-sensitive metrics: AUC, G-mean, F1-score, false positive rate, and false negative rate. After implementing the model, SMOTE-Boost ranked one by achieving the 96% AUC score and more than 90% G-mean and F1-score.

So balancing techniques improve the model performance, and other cost-metrics also improve. Of course, Undersampling and oversampling both have their own benefits. However, SMOTE performed better with ensemble-based multi classifier.

## 2.5 Need of Explainable AI

Artificial intelligence brings a revolution in the finance industry. Credit risk assessment automation makes the process faster and accurate. But after the decision of the proposal, a rejection of the loan will create a question of why the loan got rejected? The firm has to explain to the customer in human terms on what basis the loan was not approved. The current machine learning classifier can not explain because most of the model has black-box nature. They can provide the default risk probability, but the model cannot explain the reason behind the result.

The need for eXplainable AI (XAI) is an important requirement nowadays. The XAI aims to find out the features responsible for the classifier's decision result and explain them to the customer when needed.

(Biecek et al., 2021; Bussmann et al., 2021; Dastile and Celik, 2021; Egan, 2021; Hadji Misheva et al., 2021; Moscato et al., 2021) These recent studies explain the importance of the XAI in credit risk assessment. Based on the studies, figure 2.4.1 shows the most popular XAI in the finance machine learning field.

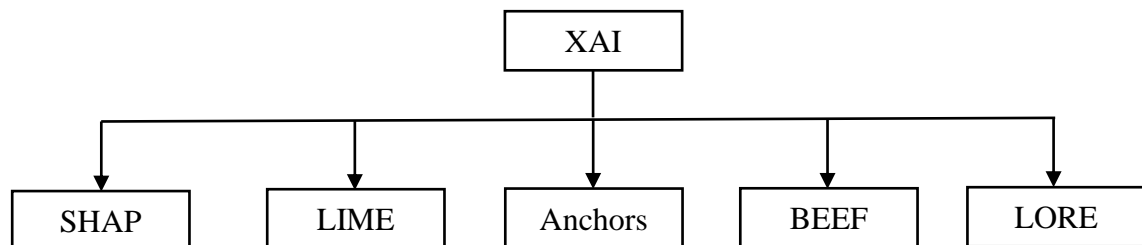


Figure 2.4.1 Popular Explainable AI in Credit risk assessment

(Lundberg et al., 2017) introduced an explainable AI SHapley Additive exPlanation (SHAP), which explains the classifier's black-box nature. This model works along with the prediction algorithm and measures each feature's contribution to the prediction result. (Biecek et al., 2021; Bussmann et al., 2021; Dastile and Celik, 2021; Egan, 2021; Hadji Misheva et al., 2021; Moscato et al., 2021) Multiple research studies explore the SHAP performance, and it always proved its superiority over the other proposed method. SHAP is the most stable and consistent method. It is not classifier dependent, which means it will give the same Shapley value to the feature even the model changes the classifier.

**LIME** (Local Interpretable Model-agnostic Explanations) is a subset of SHAP and the subsequent popular XAI after SHAP. (Dastile and Celik, 2021; Hadji Misheva et al., 2021; Moscato et al., 2021) However, in comparison with SHAP, LIME underperforms because it is not stable. Every time a new classifier was introduced in the model, the feature weightage changed in LIME. (Hadji Misheva et al., 2021) done the comparative analysis between LIME and SHAP using the different classifiers (XGBoost, SVM, Neural Network). The shapely value for default and the non-default case features was always the same for every classifier. However, LIME gave every classifier a different feature rank for default and non-default. The study concludes that the SHAP outperformed the LIME but the processing time for SHAP is higher than the LIME.

(Moscato et al., 2021) the paper did the comparative analysis among five XAI SHAP, LIME, Anchors, BEEF (Balanced English Explanations of Forecasts), and LORE (Local Rule-Based Explanations). The **BEEF** concept is based on a clustering algorithm, whereas LORE is based on the decision rule. All five methods combine with three base classifiers (LR, RF, and MLP) with various balancing techniques (ROS, RUS, SMOTE, IHT, SMOTE-TOKEN). Moreover, the study concluded that SHAP with logistic regression and random oversampling results in the highest precision of about 92%, which outperformed all the methods.

(Bussmann et al., 2021) studied a TreeSHAP method that uses the benefit of minimum spanning tree, and the proposed XAI was combined with XGBoost to develop an explainable predictive model. The TreeSHAP method is faster than the SHAP, which overcomes the SHAP disadvantage. The performance of the TreeSHAP with imbalanced data still needs to be explored.

So, SHAP is the most reliable, stable, and consistent explainable AI method proposed by most studies. And it is easy to incorporate nature with any classifier makes it more demanding for machine learning applications.

## 2.6 Discussion

Recent studies focus more on cost-sensitive learning and explainable AI to improve the credit risk assessment application. The cost-sensitive learning will provide a more accurate and robust model, and explainable AI will provide human trust in the model. For the cost-sensitive learning, many classifiers proposed, and multiple evaluation metrics are also being discussed to reduce the misclassification cost. Furthermore, many researchers are also exploring multiple resampling techniques to make cost-sensitive learning more robust to handle imbalanced data. The next milestone step is to add an explainable AI method to overcome the classifier's black-box nature. Table 2.6.1 will summarise all the recent studies related to the objective of this research study

Table 2.6.1 Summarization of research studies

Research Paper	Cost-sensitive	Classifier	Evaluation metrics	Balancing Technique	Explainable AI
(Hamori et al., 2018)	No	11 Classifier based on Bagging, Boosting and Deep Neural Network	AUC, ROC and F-score	None	None
(Li et al., 2018)	No	multiple-round ensemble learning are XGBoost, DNN, and LR	AUC and ROC	None	None
(He et al., 2018)	No	stacking ensemble using RF and XGBoost as base classifier	AUC, H-measure, KS, F-measure, GMean, Logloss	supervised undersampling approach (Extended BalanceCascade)	None

<b>Research Paper</b>	<b>Cost-sensitive</b>	<b>Classifier</b>	<b>Evaluation metrics</b>	<b>Balancing Technique</b>	<b>Explainable AI</b>
(Ma et al., 2018)	No	LightGBM and XGBoost	logloss	None	None
(Oreški and Oreški, 2018)	Yes	HGA-NN	Accuracy, AUC, F-score, F- $\beta$ , TP, FN, FP, TN	ROS and SMOTE	None
(Saidi Meryem et al., 2018)	Yes	CS-CART	Error, Specificity, Sensibility, Cost, Friedman test	None	None
(Feng et al., 2018)	Yes	Dynamic ensemble of SVM, DT and NN based on soft probability	Type I error, Type II error and the misclassification cost	None	None
(Shen et al., 2019)	Yes	Ensemble of BP-NN and AdaBoost with PSO algorithm	Type-I accuracy, Type-II accuracy, Total accuracy, G-Mean, F-Measure, AUC	SMOTE	None
(Zhang et al., 2019)	No	enhanced multi-population niche genetic algorithm (EMPNGA)	Accuracy, AUC, H measure, Brier score, AvgRank	None	None
(Song and Peng, 2019)	Yes	SVM, MLP, LR and C4.5	G-mean, F-measure, AUC, FP rate, FN rate, and time	SMOTE	None
(García et al., 2019)	No	BAGGING, AdaBoost, random subspace, DECORATE, rotation forest, random forest, and stochastic gradient boosting	AUC, True Positive Rate, True Negative Rate	None	None

Research Paper	Cost-sensitive	Classifier	Evaluation metrics	Balancing Technique	Explainable AI
(Faris et al., 2020)	Yes	The basic classifiers: J48, random tree, Rep tree, k-NN, NB, and MLP, and the ensemble classifiers: AdaBoost, Bagging, random forest, rotation forest, and DECORATE	Accuracy, Type I, Type II, G-mean, AUC	SMOTE	None
(Niu et al., 2020)	Yes	Resampling ensemble model based on data distribution (REMDD)	AUC, G-mean, Maj_Recall, Min_Recall	ROS, RUS, SMOTE, UnderBagging, SBD, SBC	None
(Wong et al., 2020)	Yes	Cost-Sensitive Deep Neural Network Ensemble (CSDE) and Cost-Sensitive Deep Neural Network (CSDNN)	Avg. TPR Avg. TNR Avg. AUC Avg. G-mean	SMOTE, ROS and RUS combined with other base classifiers to compare the result	None
(Wu et al., 2020a)	Yes	Three layers cascaded XGBoost	Accuracy and F1-Score	None	None
(Kun et al., 2020a)	Yes	Stacking ensemble classification model ANN, RF, AdaBoost, and XGBoost	Accuracy, Precision, Recall, F1-Score, AUC	SMOTE	None
(Xia et al., 2020)	No	Ensemble of RF, GBDT, XGBoost, LightGBM, CatBoost	Accuracy, AUC, H-measure, Brier-score	None	None
(Chi et al., 2020)	No	Decision Tree	AUC	TRUST	None

<b>Research Paper</b>	<b>Cost-sensitive</b>	<b>Classifier</b>	<b>Evaluation metrics</b>	<b>Balancing Technique</b>	<b>Explainable AI</b>
(Wang and Zhang, 2020)	No	hybrid system HFES and HGIES.	F-score and Gini index	None	None
(Dzik-Walczak and Heba, 2021)	Yes	Heterogeneous ensemble of Logistic Regression, Neural networks	AUC, Gini coefficient and KS	None	None
(Egan, 2021)	No	Random Forest, LightGbm, XGBoost	Accuracy, Precision, Recall, AUC	None	Nearest-Unlike-Neighbor Counterfactual
(Busmann et al., 2021)	No	XGBoost	ROC, TPR, FPR	None	TreeSHAP
(Hadi Misheva et al., 2021)	No	Logistic Regression, Random Forest, XGBoost, SVM, and Neural Network	AUC score	None	LIME and SHAP
(Moscato et al., 2021)	No	Logistic regression, Random Forest, Multi-Layer Perceptron	AUC, Sensitivity, Specificity	None	LIME, SHAP, BEEF, Anchors, LORE
(Dastile and Celik, 2021)	No	2D Convolutional Neural Networks	Accuracy, AUC, Brier Score, H-measure	None	LIME and SHAP
(Li et al., 2021)	Yes	LightGBM	AUC and KS	None	None
(Barua et al., 2021)	No	CatBoost, RF, XGBoost	Accuracy	None	None

<b>Research Paper</b>	<b>Cost-sensitive</b>	<b>Classifier</b>	<b>Evaluation metrics</b>	<b>Balancing Technique</b>	<b>Explainable AI</b>
(Chengeta and Mabika, 2021a)	No	Deep learning CNN, XGBoost, CatBoost and LightGBM	TPR, ROC, FPR, F-measure, Recall	None	None
(Yotsawat et al., 2021a)	Yes	Cost-sensitive Neural Network Ensemble (CS-NNE)	ROC Curve, DDR, G-Mean	RUS and SMOTE	None
(Shen et al., 2021)	Yes	long-short-term-memory (LSTM) neural network and AdaBoost	AUC and KS	Improved SMOTE	None
(Zhang and Li, 2021)	No	Logistic Regression, Random Forest, GBDT, XGBoost, and LightGBM were used to train, and the SVM model was used to perform Stacking integration	Precision, Recall, F-Score	None	None
(Zhang et al., 2021)	Yes	XGBoost, GBDT, AdaBoost, RF, LR, Bagging, and ExtraTree,	AUC, F-score, Brier score, KS	Bagging based Random undersampling	None
(Jin et al., 2021)	Yes	hybrid genetic algorithm (HGA)	BACC, F-score, G-mean, Recall	Voting Instance Hardness Threshold (VIHT)	None
(Sandica and Fratila, 2021)	Yes	Ensemble-based AdaBoost, LogitBoost, GentelBoost, and Random Forest	TP, FN, FP, TN	None	None

Research Paper	Cost-sensitive	Classifier	Evaluation metrics	Balancing Technique	Explainable AI
(Sterner et al., 2021)	Yes	LR, RF and Gradient Boosting	MMC, ACC, SENS, SPEC, AUC	SMOTE, Theoretical thresholding, Empirical thresholding	None
(Biecek et al., 2021)	No	random forest, gbm, and xgboost	KS and Gini value	None	Global interpretations
(Wu et al., 2022)	Yes	COst Sensitive Loan Evaluation (COSLE)-DT, COSLE-RF, CSXGBoost	Type1 error, Type2 error, AUC, G-mean and lender's Return (LER) evaluation	None	None

Table 2.6.1 lists 34 research studies with their important model approach, and only six studies have used explainable AI to interpret the prediction result. So, in the credit assessment model field, explainability needs more focus to be explored. In this study, the ensemble classifier that benefits from lower variance and lower bias, which is the prime requirement in case of data imbalance problems, will be integrated with the XAI to make the credit assessment model better and more interpretable.

## 2.7 Summary

As discussed, a credit risk assessment model needs robust cost-sensitive learning with explainable AI is the need of the current time. XGBoost and LightGBM gave an outstanding result for the cost-sensitive metrics AUC score, TPR, FPR, and G-mean. Even XGBoost proves its superiority in explainable AI as well. With the SHAP method, XGBoost outperformed the model accuracy. The resampling technique SMOTE also plays a significant role in the cost-sensitive learning to reduce the misclassification cost (Type I and Type II error). So the combination of resampling technique with gradient boosting classifier (XGBoost, LightGBM, and CatBoost) and explainable AI (SHAP) is exciting to be explored.

This research study will study the ensemble of XGBoost, LightGBM, and CatBoost. Furthermore, the resampling technique SMOTE will be used to reduce the Type I and Type II error, and the model will be evaluated for G-mean, AUC score, and F-measure to make the model more robust. Moreover, SHAP XAI will finally be used to explain the model in terms of a layman.



## **Chapter 3: Research Methodology**

### **3.2 Introduction**

The evolution of machine learning in credit risk assessment faces multiple issues, high dimensionality, high volume, class imbalance, and interpretability of the prediction result. So proposed research study will deal with class imbalance and interpretability to make the final model robust and transparent.

In this research methodology section, each step of methodology will be discussed. This section explains the data description and essential features necessary for this research study. Furthermore, preprocessing of data cleaning, transformation, and resampling steps will also be briefed. The modeling step stacking ensemble of boosting classifier will be explained and evaluated for misclassification cost by the cost-sensitive metrics. Furthermore, as a final step, an explainable AI will be demonstrated how it will help the model explain the prediction result with the factor in terms of human terms.

### **3.2 Research Approach**

The approach of this model is to design a most accurate machine learning pipeline in terms of misclassification, and it can also be interpretable for the prediction result explanation. Figure 3.2.1 demonstrates a workflow of methodology which will be implemented in this research study.

The whole methodology is divided into three parts. Part one consists of data preprocessing, where all data cleaning, by handling the null values and outliers, will explain the process in detail. Furthermore, the next step, label encoding for categorical features, data splitting, and scaling, will prepare the dataset for modeling.

In the second part, the architecture of the stacking ensemble will be discussed. The stacking architecture has two layers one estimator layer and a second meta-classifier. The estimator layer will be explained using three base model classifiers based on boosting, XGBoost, LightGBM, and CatBoost. Moreover, Logistic regression will be briefed for the meta-classifier to help combine all base model classifiers into one for stacking ensemble. Finally, multiple cost metrics will also be discussed, which will evaluate the final model for the misclassification cost.

And the last part will discuss the explainable AI (SHAP) methodology, how it will help the model to interpret the factors behind the prediction result for the final model

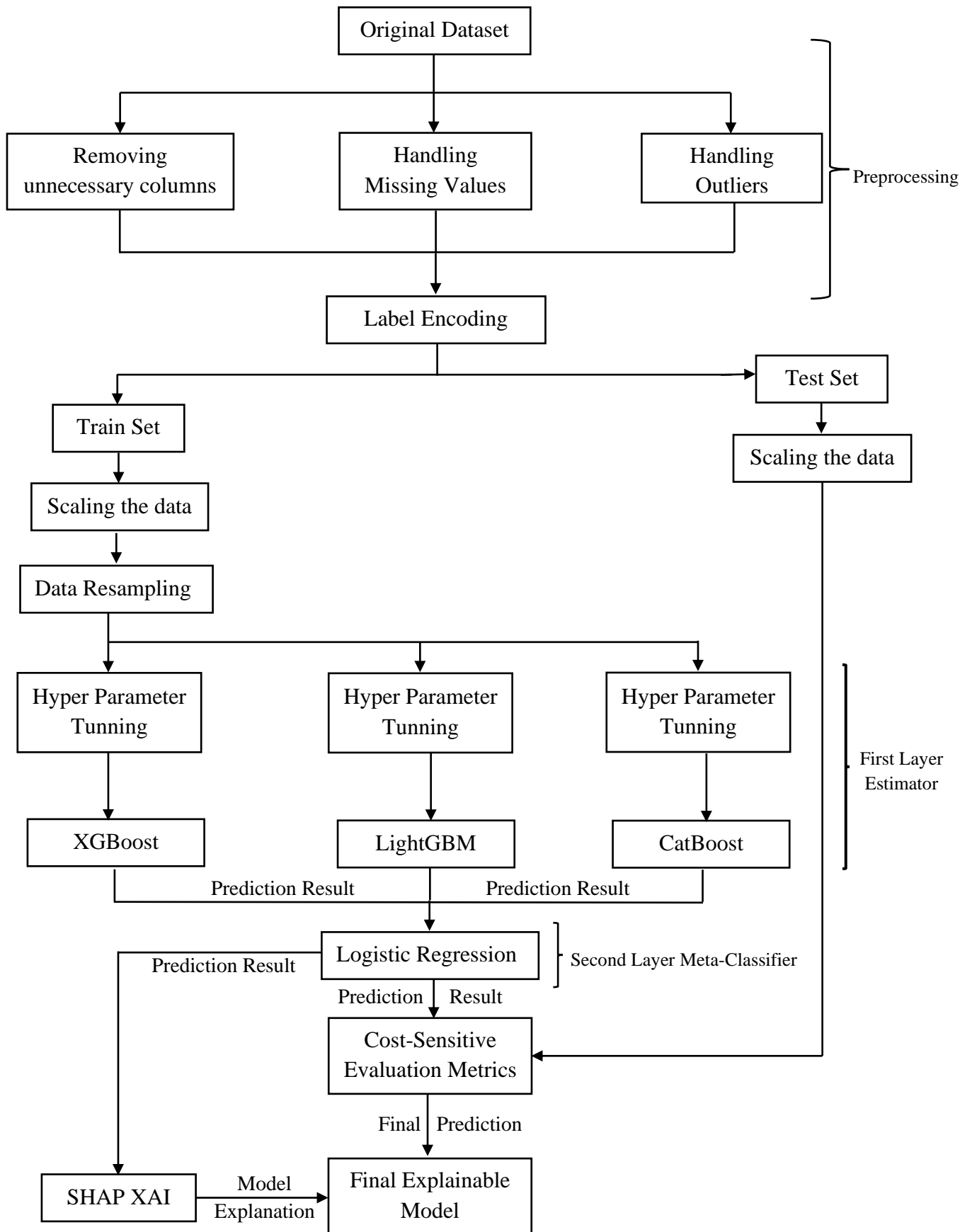


Figure 3.2.1 Workflow of Stacking Based explainable cost-sensitive model

### 3.2.1 Data Acquisition and Description

The American peer-to-peer lending company named Lending club's data set is being used for this research study collected from the kaggle(Lending club, 2021). It has two csv files for accepted and rejected loans, which hold information from 2007 to 2018. The rejected loan data set only holds precise information about the rejected loan proposal, but the accepted file holds all the detailed information about the lending. So only the accepted dataset will be considered in this research study. There are 151 columns in accepted csv, and the **loan\_status** field holds a class label for default and non-default, so it is the target variable. For any risk assessment, some of the information was asked from the borrower while filling the application for the credit. Based on that, below important feature is explained of lending club dataset:

Table 3.2.1.1 Data Description of Lending Club

Column Name	Description
loan_amnt	The loan amount sanctioned to the borrower
term	A number of months for loan repayment (36 months or 60 months).
int_rate	The applicable interest rate on loan.
installment	The monthly payment amount for the sanctioned loan.
grade	Loan grades are assigned by the lending club based on the borrower's FICO Score.
emp_title	Employment title of the borrower
emp_length	Length of service of the borrower.
home_ownership	Status of home (rented/owned/mortgaged) of the borrower
annual_inc	Annual income of the borrower.
verification_status	Indicates if Lending Club verified income.
issue_d	Issue date of the loan.
loan_status	Status of the loan (fully paid/default/charged off)
purpose	Purpose of the loan.
zip_code	The first three digits of the zip code borrower's address.
dti	It is a ratio of the total debt of the borrower to his total income
earliest_cr_line	Earliest date of open credit of the borrower.
fico_range_low	Lower FICO score boundary of the borrower.
open_acc	How many open credits the borrower has
revol_util	The usage rate of the revolving credit by the borrower
total_acc	A total number of credit account of the borrower.
application_type	Type of loan application (individual/ joint)
annual_inc_joint	Total annual income of the applicants (borrower & co-borrower)

### 3.2.2 Required Resource

The resources that will be going to be part of this study is represented by the table 3.2.2.1, which explains the hardware and software requirement for this study.

Table 3.2.2.1 Hardware and software requirements

Hardware Requirement	
Feature	Specification
Operating System	Windows 10/ Mac X or higher
RAM	8 GB or higher
Processor	Intel Core i5 or higher
GPU	NVIDIA GTX equivalent or higher
Software Requirement	
System Requirement	Anaconda Jupyter Notebook/ PyCharm/Google Colab
	Python 3.7 or higher environment set up
Python Package Requirement	pandas 1.3, numpy 1.21 or higher
	matplotlib 3.4, seaborn 0.11.1 or higher
	scikit-learn 0.24.2, imbalanced-learn 0.8.0, xgboost 1.5.0, lightgbm 3.3.2, catboost 1.0.4, mlxtend 0.19.0, shap 0.40.0 or higher

### 3.2.3 Data Preprocessing

The machine learning model needs quality data. If data has null values, high variance in data columns means outliers. Then there will be a huge variance in model performance; hence, it needs to be cleaned up before feeding the data to the model. So for cleaning up the dataset, some important approach is to handle the null values, the outliers and the transformation which is needed to be done for the sake of good risk assessment is as follows:

- The main goal of this research is to predict the risk probability of any loan proposal for default and non-default. So, the target feature should have only a non-default and default class label. Figure 3.2.3.1 shows the class labels available for the target feature loan\_status in the lending club dataset. In this research study, the non-default Fully Paid class label will be considered and for default Charged Off, and Default class label will be used.

```
Fully Paid
Current
Charged Off
Late (31-120 days)
In Grace Period
Late (16-30 days)
Does not meet the credit policy. Status:Fully Paid
Does not meet the credit policy. Status:Charged Off
Default
```

Figure 3.2.3.1 Class labels for target feature

Except for these three class labels, other values will be removed from the dataset for further processing. Because other class labels like Current, Late, In Grace period, and others do not provide clear information that the particular loan is the default, these class labels will be removed.

- The next step, dropping unnecessary columns that hold non-meaningful information, like id, member\_id, url, and so on, will be done. In this research study, the important feature application\_type needs additional transformation. Because this column holds the information about loan application type is individual or joint. In a real case scenario, when an applicant applies for the credit, it has two ways to apply, either individual or joint applicants. Each application needs different input information from the applicant, and risk assessment will be done based on that only.

So, special handling is required for this column, and figure 3.2.3.2 shows the distribution of applicant\_type for the lending club dataset, and it shows that 94.66% of the applicant is individual and only 5.34% is joint. So, for the joint applicant, most of the values will be null that is 94.66% and most of will be removed during the null value handling process, then no data for the joint applicant will be available for analysis.

```
Individual    94.660428
Joint App     5.339572
Name: application_type, dtype: float64
```

Figure 3.2.3.2 Class label distribution for application\_type

So, in this research study, the accepted data frame will be split into Individual application type and Joint applications, and for both data frames, the whole methodology will be implemented.

- Afterward, missing values will be identified by summing the null values count for each column. If the null value % is more than 50% for any column, it will be dropped because the imputation of more than 50% null value may make the column skewed towards the imputed value. After dropping the columns, the next step is to impute the null values. The best way to impute the null values is either using mean() or mode() or median() function. Mean() or median() is for the numerical features, and mode() is for categorical. However, for high variance, column mode() is the best way to impute the null values to avoid the variance in the column data.
- After successfully handling the null values, the next approach is to identify the outliers by plotting them using a boxplot. Outliers are the extreme values that lie out of the column average data range. Handling outlier is significant because its high variance will affect (Nyitrai and Virág, 2019; Chakravarty et al., 2020; Yotsawat et al., 2021a) the model performance. There are two ways to handle the outliers, either by eliminating them (Yotsawat et al., 2021a) or capping(Nyitrai and Virág, 2019) them. Elimination of the outlier for the financial dataset may lead to losing some important information for the risk assessment. So, in this research study, the capping of outliers will be considered.

### 3.2.4 Data Transformation

Data transformation is the next important step before the modeling. Unfortunately, most of the models understand only numerical data. So, to make the dataset model understandable, all categorical features will be converted into numerical features using label encoding.

There are other techniques available to convert categorical features to numerical features, one-hot encoding. In label coding, all categorical labels will be encoded from 0 to number\_of\_class-1. No extra features will be created, but in the one-hot encoding process for each class label new column will be created, which will increase the high dimensionality; hence label encoding will be used for the data transformation step in this research study.

The next step is to split the dataset into train and test sets, and afterward, data scaling will be applied to normalize the data range of the dataset.

### 3.2.5 Class Balancing

The lending club dataset has the target feature loan\_status on which classification model will be implemented to predict the credit risk. The machine learning model assumes that the target feature class label is balanced for classification problems. From figure 3.2.5.1, it is clear that the target feature loan\_status is imbalanced 47.63% of class labels are Fully Paid (non-default), and only 11.88% of class labels are default (charged-off + Default).

Fully Paid	47.629771
Current	38.852100
Charged Off	11.879630
Late (31-120 days)	0.949587
In Grace Period	0.373164
Late (16-30 days)	0.192377
Does not meet the credit policy. Status:Fully Paid	0.087939
Does not meet the credit policy. Status:Charged Off	0.033663
Default	0.001769

Figure 3.2.5.1 Class label distribution for target feature

So, to solve this issue, this research study will use a resampling technique. Multiple studies proposed many resampling methods (Faris et al., 2020; Kun et al., 2020a; Moscato et al., 2021; Shen et al., 2021; Yotsawat et al., 2021a) and SMOTE (Synthetic Minority Oversampling Technique) proved its performance with multiple classifiers (RF, LR, AdaBoost, NN, XGBoost). Hence, SMOTE will be used as a resampling technique to balance the dataset for the target feature and ADASYN technique will also be used to compare the result with SMOTE.

#### **SMOTE (Synthetic Minority Oversampling Technique):**

SMOTE is an oversampling technique that works on minority classes and creates a new synthetic by replicating minority classes using the nearest neighbour concept. It was first proposed in (Chawla et al., 2002). this technique is widely used because it does not

create duplicates of the sample, but it generates new synthetic values slightly different from the original minority class values. Furthermore, the concept of nearest neighbor value makes this technique more prominent for the oversampling method because the created data points are always within the minority sample range. There are the following four significant steps for SMOTE:

Step 1: Identify any minority data points.

Step 2: Select the value of  $k$  for a number of nearest neighbor points.

Step 3: Make a line between the minority data point and any chosen neighbor point and create a synthetic point anywhere on the line.

Step 4: Repeat all steps from 1 to 3 for all the minority data points until they are balanced.

### 3.2.6 Cost-sensitive metrics

the performance of the classification model is always evaluated using accuracy metrics. It calculates the ratio of the number of correctly predicted class labels to the total number of class labels. If the class labels are balanced, then the accuracy is the best choice to measure the model performance. However, in case of imbalance, a number of misclassified class labels is also required, which can not be evaluated by accuracy.

Hence cost-sensitive metrics will be used in this research to evaluate the model performance, especially for misclassification. The confusion matrix (Feng et al., 2018; Oreški and Oreški, 2018; Shen et al., 2019; Sandica and Fratila, 2021) will summarize the prediction result classifier, and then, based on that, multiple cost metrics will be calculated. Table 2.3.1 confusion matrix explains that the True Positive (TP) indicates a number of actual non-default data predicted correctly and True Negative (TN) number of actual default labels predicted correctly, and False Positive (FP) is a number of Default misclassified as non-default and False Negative (FN) shows a number of non-defaults misclassified as Default. To reduce the misclassification cost, Type I and Type II is defined as below, which will be used in this research methodology

**Type I error** is misclassifying the default data as non-default that is False Positive.

**Type II error** is misclassifying the non-default as default that is False Negative.

So, to reduce these two-error model performance will be evaluated by AUC-score, F1-Score, and G-mean (He et al., 2018; Biecek et al., 2021; Dzik-Walczak and Heba, 2021; Shen et al., 2021; Zhang et al., 2021).

**AUC -score (Area Under the Curve):** these metrics measure the separability among class labels predicted by the classifier. Its value ranges from 0 to 1, higher the AUC score (Yotsawat et al., 2021a) better the model for correctly predicting the class labels. AUC score is calculated by the ROC curve, which draws a plot between True Positive Rate and False Positive rate. So more the curve bend towards the TPR better the model in terms of misclassification.

**G-Mean (Geometric Mean):** this metric is best for class imbalance problems (Ri and Kim, 2020). It measures the classifier performance for both the majority and minority classes. Lower G-mean shows poor performance of the model. Even the majority class label classified perfectly, but if there is misclassification happens for minority class, G-

mean will be low for the model. It can also measure the overfitting of the model by lowering its value. It can be calculated using a confusion matrix value that is True Positive, False Positive, True Negative, and False Negative.

$$G - Mean = \sqrt{\left(\frac{True\ Positive}{True\ Positive + False\ Negative}\right) * \left(\frac{True\ Negative}{True\ Negative + False\ Positive}\right)} \quad (3.2.5.1)$$

**F1- Score:** Another popular metrics for imbalanced classification problems is F1-measure. This metric measures the model performance based on the balance between precision and recall. **Precision** measures the resulting quality of the classifier, low precision means high false positive rate that is high number of default loans misclassified as non-default. And **Recall** measures the true positive rate, that is, how many class labels are correctly classified by the model. F1 measure the balance between them using the below formula:

$$F1 - Score = 2 * \left(\frac{Precision * Recall}{Precision + Recall}\right) \quad (3.2.5.2)$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (3.2.5.3)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (3.2.5.4)$$

### 3.2.7 Explainable AI (XAI)

As previously discussed, the machine learning model is not self-explainable to interpret the prediction result. However, an explainable model is more trustful than the traditional one. So to make the model explainable, one additional step needs to be taken by introducing an explainable ai into the machine learning pipeline of the credit risk prediction.

In the literature review, multiple XAI was reviewed and analysed, and finally, SHAP (SHapley Additive exPlanations) proved its worth. It estimates the Shapley value for each feature using the prediction result of the training dataset and then will decide to include the feature into consideration or not based on the Shapley value.

Its global and local explainability make it unique from other XAI (Biecek et al., 2021; Bussmann et al., 2021; Dastile and Celik, 2021; Hadji Misheva et al., 2021; Moscato et al., 2021), which makes its result consistent for the different classifier. Furthermore, its graphical representation of the feature contribution into the predicational result is outstanding.



### 3.3 Proposed Method (Classification)

In the literature review, the various method was discussed for credit risk assessment and multiple individual classifier-based models (Hamori et al., 2018; Ma et al., 2018; Chengeta and Mabika, 2021a; Egan, 2021; Li et al., 2021; Yotsawat et al., 2021a) was compared with the multi-base classifier model (Feng et al., 2018; He et al., 2018; Xia et al., 2020; Sandica and Fratila, 2021; Zhang and Li, 2021; Zhang et al., 2021) and result showed that the performance of multi-base classifier performed better than the individual classifier. And the most performing classifier was gradient boosting based XGBoost, LightGBM, and CatBoost (Ma et al., 2018; Xia et al., 2020; Barua et al., 2021; Li et al., 2021). Their performance for cost-sensitive learning outperformed another classifier. However, even the neural network or deep learning-based classifier did not achieve the highest G-mean or AUC score compared to them.

So in this research study, a multi-base classifier-based model will be implemented using XGBoost, LightGBM, and CatBoost. There are two popular approaches for classification problems to combine multi-base classifiers into a single classifier. One approach is based on voting(Wong et al., 2020; Zhang et al., 2021), and another one is stacking (He et al., 2018; Jin et al., 2021; Zhang and Li, 2021). The stacking-based ensemble model performed better for cost-sensitive learning. Therefore, this research methodology will use a stacking ensemble classification approach based on three base classifiers, XGBoost, LightGBM, and CatBoost.

**Stacking Ensemble:** A stacking ensemble is the extension of a voting-based ensemble. Figure 3.3.2 explains the pseudo-code for the stacking ensemble. Figure 3.2.1 shows the architecture of the stacking ensemble, which is divided into two layers. In the first layer, a multiple base classifier will be trained. The dataset and prediction output will be combined by the meta classifier by stacking all the prediction result to predict the final prediction output. For the first level of base classifier, XGBoost, LightGBM, and CatBoost and their prediction result will be stacked using Logistic regression. Some time boosting algorithm has a tendency to overfit. Logistic regression will help to avoid the overfitting issue if it finds. The two-layer architecture makes it more robust and helps to achieve the best prediction result by combining the multiple base classifier. After the final prediction, the result will be evaluated by the cost-sensitive metrics using AUC-score, Precision, Recall, F1-Score, and G-mean. And then, all metrics will be compared with the evaluation metrics of the individual base classifier (XGBoost, LightGBM, and CatBoost).

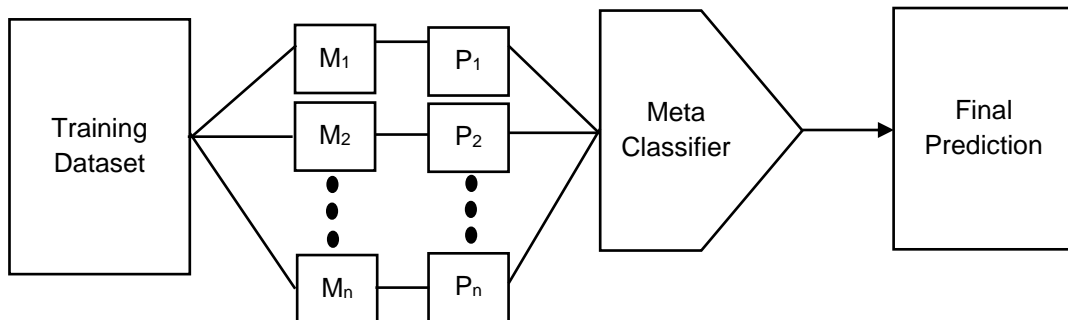


Figure 3.3.1 Stacking ensemble Architecture

Pseudo Code for Stacking Ensemble		
<b>Input:</b>	Training dataset $D = \{X_i, y_i\}$	where $i = \{1, 2, 3, \dots, m\}$
	First level base learning classifier $clf_n$	where $n = \{1, 2, 3, \dots, N\}$
	Second level meta learning classifier $clf$	
<b>Process:</b>		
<b>Step 1:</b>	Learning base-level classifier to the original dataset D.	
	<b>for</b> $n=1$ <b>to</b> $N$ :	
	$p_n = clf_n(D)$	
	<b>end</b>	
<b>Step 2:</b>	New dataset creation based on the base level prediction.	
	<b>for</b> $i=1$ <b>to</b> $m$ :	
	$D_p = \{X_{pi}, y_i\}$ where $i = \{1, 2, 3, \dots, m\}$	
	<b>end</b>	
<b>Step 3:</b>	Learning meta-classifier to the new predicted dataset from the base classifier.	
	$p = clf(D_p)$	
<b>Output:</b>	$P(X) = p$	

Figure 3.3.2 Pseudocode for stacking Ensemble

**XGBoost (eXtreme Gradient Boosting) Classifiers:** this boosting algorithm was successfully implemented by (Chen and Guestrin, 2016). This is an advancement of the gradient boosting decision tree (GBDT). XGBoost is the most favorite machine learning classifier among many researchers (Li et al., 2018; Kun et al., 2020; Bussmann et al., 2021; Egan, 2021; Hadji Misheva et al., 2021). The reason behind its popularity is its ability to handle imbalanced datasets. Moreover, it has an inbuilt function of lasso and ridge regularization, which helps the model avoid overfitting. Its parallel processing nature reduces the model complexity for the high dimensional dataset because of that it is very easy and fast to tune the hyperparameter for the XGBoost.

XGBoost works on a sequential decision tree, and all independent features are assigned with a weight number. Afterward, all these features will be input to one of the decision trees, for every wrong prediction will increase the number of trees, and again the feature will be fed into the next decision tree to convert the weak learner into a strong one.

**LightGBM (Light Gradient Boosting Machine) Classifiers:** proposed by (Ke et al., 2017) in the advancement of GBDT like XGBoost. It has many advantages of XGBoost, and it has two novel techniques which make this classifier achieve the highest performance compared to other

benchmark classifiers (Egan, 2021; Li et al., 2021; Zhang and Li, 2021), one gradient-based one side sampling (GOSS) and another one is exclusive feature bundling (EFB). GOSS helps to keep instances with large gradient samples while performing sampling on small gradient instances to provide higher accuracy. This makes this classifier favorite for cost-sensitive learning. And EFB helps the LightGbm to work with categorical features without any transformation.

LightGBM architecture is based on the leaf-wise approach for the decision tree instead of growing tree level-wise. This helps decrease the loss and handle the high volume data, but this may cause an overfitting problem for small volume, which can be fixed by its in-built technique, a histogram-based decision tree for the best split point to improve the overall performance of the model.

**CatBoost Classifier:** This is another most favorite classifier for cost-sensitive learning. It was proposed (Prokhorenkova et al., 2017) to handle the categorical feature using boosting. This was designed to handle an imbalanced dataset with the categorical feature. It proved its superiority over others (Xia et al., 2020; Barua et al., 2021; Chengeta and Mabika, 2021), boosting classifier, neural network, or traditional classifier for cost-sensitive learning.

It has two novel techniques, ordered boosting and ordered target encoding. Using ordered boosting avoids overfitting issues, and the ordered target encoding approach helps to handle categorical features using one-hot encoding or advanced mean encoding.

The approach of working on the decision tree of CatBoost is to develop a balanced decision tree using the binary tree concept to restrict the number of splitting per level. This way, its computation time improves, and it helps to handle high-dimensional data.

### 3.4 Summary

The machine learning pipeline of this research methodology will have three-level. Data pre-processing, transformation, and class balancing will be performed first. Furthermore, at the second level, stacking ensemble using XGBoost, LightGBM, and CatBoost as a base classifier and Logistic Regression as meta-classifier will predict the classification result for the training dataset. Afterward, model performance will be evaluated using the cost-sensitive metrics AUC score, F1-measure, Precision, Recall, and G-mean. And at the final level, an explainable ai SHAP will be applied to the training prediction result to interpret the factors behind it.

Using the proposed methodology will definitely answer the research question of this study to make the black-box model explainable in cost-sensitive learning.

## Chapter 4: Analysis and Design

### 4.1 Introduction

This chapter demonstrates the exploratory data analysis (EDA), modeling preprocessing, and model implementation performed on the dataset. EDA involved critical feature analysis like target feature **loan\_status** has multiple class labels but only Fully Paid, charged off, and the default is considered in this research paper. The second important feature is **application\_type**, any loan proposal has two types of application category individual and joint, and both have different features to affect the loan risk, so both applications need to be treated differently hence accepted dataset split into individual and joint based on the `application_type` and all the EDA, modeling preprocessing and implementation steps are performed on both datasets.

Under the data cleaning steps, identification of null values and handling them will be discussed. For handling null values, if the null value percentage is more than forty, those columns are dropped, and the remaining columns mean and mode has been used to impute the missing value. Outlier handling is the next step that keeps the data within the range to improve the prediction result by capping the extreme values. While handling the outlier, skewed columns are identified and removed from the dataset. In addition, redundant features are also identified and removed from the dataset. After the data cleaning, multiple feature analyses like univariate, bivariate, and multivariate are performed on the cleaned dataset to identify the feature behaviour for increasing the loan risk for the application.

Afterward, the modeling preprocessing step explains the process to prepare the dataset for the modeling by label encoding the categorical features and normalizing the numerical feature by scaling the data. Furthermore, the dataset is split into independent features (X) and dependent features or target features (y). Then, dividing the dataset into train and test in 70:30 ratio using train and test split is performed to prepare the dataset for the modeling.

And implementation steps share the model implantation process, how the hyper-parameter tuning is performed on the classifier (XGBoost, LightGBM, CatBoost), how the balancing technique (SMOTE and ADASYN) is applied on the dataset, and how the stacking classifier is worked. Finally, the explainable AI method is applied to the classification result and interprets the black-box model prediction result in human terms.

### 4.2 Exploratory Data Analysis (EDA)

The lending club dataset is explored and analysed in this research study. The imported csv file has 2260701 rows and 151 columns. And 113 features are numerical, and 38 features are categorical. Some features like `id`, `member_id`, and `url` do not hold any meaningful information about the data; they are only for identifying the loan application, so these features are dropped from the dataset for further processing. The `loan_status` is the dependent feature of the dataset. Some of the features need extra analysis, which will be discussed further.

#### 4.2.1. Analysis of loan\_status

The target feature of the dataset is loan\_status which shows the loan application status whether it is fully paid or charged off or current or default. Figure 4.2.1.1 shows all the class labels of the loan status with the counts.

Fully Paid	1076751
Current	878317
Charged Off	268559
Late (31-120 days)	21467
In Grace Period	8436
Late (16-30 days)	4349
Does not meet the credit policy. Status:Fully Paid	1988
Does not meet the credit policy. Status:Charged Off	761
Default	40

Figure 4.2.1.1 class labels for loan\_status

The current status shows that the loan application is active and running. The labels Late (31-120 days), In Grace Period, and Late (16-30 days) shows that the EMIs are late for this account; it may get default or not. The Fully Paid shows that these applications are non-default, which has paid all their debts on time, and the class labels Charged off and Default class labels hold information about the default accounts. Only these three class labels are part of the further learning. So, a new target feature is created, with 0 for fully paid (non-default) and 1 for charged off or default. And the loan\_status is removed from the dataset.

#### 4.2.2 Analysis of application\_type

There are two types of loan applications individual and joint. The individual has only one applicant, but the joint has two primary and secondary. For each applicant type, different information is needed for credit assessment; hence they both need different feature information. For example, for individual type required information are fico score, dti, annual income, and for required joint applicant information is fico score for primary applicant and fico score for the secondary applicant, dti for the primary applicant and secondary applicant, annual income for the first applicant and second applicant. So, some of the feature for the individual is null, which is only applicable for joint and vice versa. So, creating two different datasets will help make an accurate prediction for each dataset. So, the dataset is divided into the individual and joint datasets based on the application\_type, and all the further steps will be performed on both of the datasets.

### 4.2.3 Handling Null Values

Null values are missing or unknown values like hyphen, blank space, or none. Most machine learning models cannot handle these values; there will be prediction errors or low-performance accuracy. So very first step of data cleaning is to identify the null values and handle them. In this research study following approach has been used to manage the null values:

- Some columns have the value 'NONE', which is the same as null, so for better null value identification, these values are replaced by the null values for both individual and joint datasets.
- For null value identification, the null value percentage is calculated for each column and both datasets (individual and joint) by summing up the null value count and dividing them by the length of the dataset.
- There are 103 out of 148 columns that have missing values for individual datasets; out of the 56 columns have more than 50% of data is null. For the joint dataset, there are 70 columns out of 148 columns have missing values; out of the 27 columns have, more than 50% data is null. Imputation for more than 50% null values is hard, and mean, mode or median value imputation skewed the column towards the imputed value because more than 50% value will be the same. So, the best way to handle these columns is to drop them.
- After dropping the high null value columns, 40 numerical features and 7 categorical features have missing values in the individual dataset. And in the joint dataset, there are 36 numerical features and 7 categorical features with null values. Imputation for the numerical feature has two options: mean and median. Using the mean, the null values will be replaced by the average value of the column. The null values will be replaced by the middle value when the data set is in ascending order using the median. The median value is better when the column is skewed; otherwise, the mean is a better option. Null values holder numerical features for both datasets (individual and joint) are not skewed; hence in this research study, mean is used to impute the null value for numerical feature and mode for categorical features.

After successfully identifying and imputing null values, the individual and the joint dataset are now cleaned from the null values.

#### 4.2.4 Handling the outliers

Machine learning algorithms are susceptible to data distribution and its range. And an outlier is the extreme data point compared to the average data range of the features. For example, the annual\_inc column had a high variance in data points figure 4.2.4.1 shows the distribution of data using a boxplot.

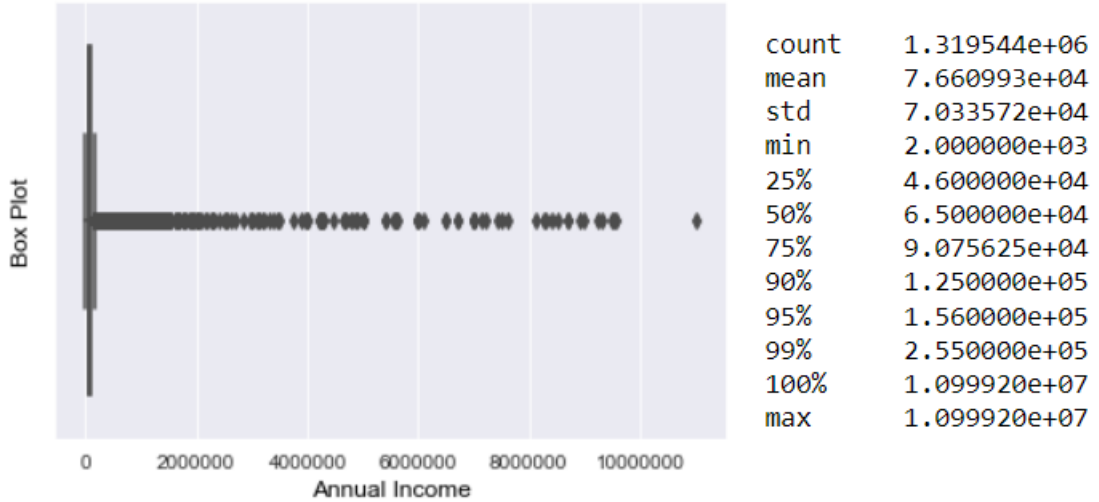


Figure 4.2.4.1 Data distribution for annual income feature

The 99<sup>th</sup> percentile data points for annual income in thousands (255k), but the 100<sup>th</sup> percentile data points hold the value of 10 million (1.099 M). There is a huge difference between the 99<sup>th</sup> percentile and 100<sup>th</sup> percentiles. These were known as outliers, affecting the training time and prediction result. So, most of the time, it was better to drop these values, but they hold important information that helps in credit assessment, so outlier capping was used in this study to keep the data points intact.

Using boxplot there were 51 columns are identified with outliers in individual dataset, and each feature was capped with the right percentile values as per required. For example, features loan\_amnt, funded\_amnt, funded\_amnt\_inv, dti, revol\_util, total\_rec\_prncp, bc\_util, percent\_bc\_gt\_75 and last\_fico\_range\_high were capped using their 1<sup>st</sup> and 99<sup>th</sup> percentile value while last\_fico\_range\_low, num\_actv\_bc\_tl, pct\_tl\_nvr\_dlq, mort\_acc, total\_acc, int\_rate with 1<sup>st</sup> and 98<sup>th</sup> percentile. Annual\_inc was capped using 1<sup>st</sup> and 95<sup>th</sup> percentile. Finally, the individual dataset was freed from outliers.

For the joint dataset, 68 columns have outliers, and these outliers were also successfully handled by capping them with the correct percentile value after studying each feature.

## 4.2.5 Handling the Skewness

Skewness is the measure of data asymmetry; skewed features may shift to the right or left. The skewed features would not add meaning to the prediction result for machine learning as it is always biased towards the skewed value. Figure 4.2.5.1 shows the data distribution for skewed features for the individual dataset. And most of the features were left-skewed towards value 0, and the feature policy\_code had only one value, i.e., 1.

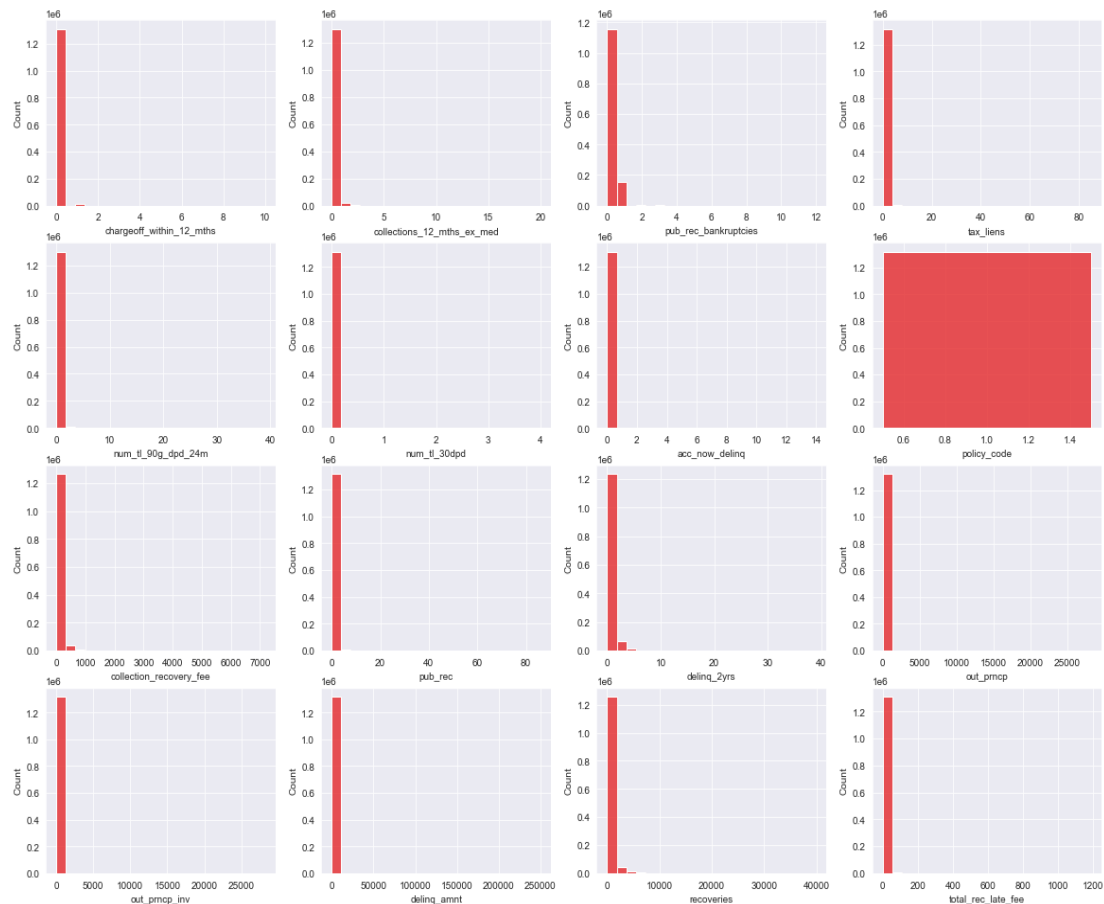


Figure 4.2.5.1 skewed features for individual dataset

All these features would not contribute to the prediction result as it always results in skewed data points, mostly 0. So, it was better to remove these columns from the dataset for better learning performance.

There are 16 and 20 skewed columns in the individual and joint dataset, respectively, and all were dropped from the dataset for further learning. Other features application\_type, hardship\_flag, pymnt\_plan have only one class label so these columns were also dropped from the both dataset.



#### 4.2.6 Handling the Redundant Features

Redundant features hold duplicate data points or almost the same data points. They were highly correlated; their correlation value was nearly 1. These features increase the learning training time and decrease the model's performance. So, cleaning these features was very important before the learning process. Figure 4.2.6.1 shows the correlation among three independent features, loan\_amnt, funded\_amnt, and funded\_amnt\_inv, and dependent feature target.

	loan_amnt	funded_amnt	funded_amnt_inv	target
loan_amnt	1.000000	0.999540	0.998539	0.066192
funded_amnt	0.999540	1.000000	0.999090	0.066247
funded_amnt_inv	0.998539	0.999090	1.000000	0.066028
target	0.066192	0.066247	0.066028	1.000000

Figure 4.2.6.1 Correlation matrix of the loan amount for individual dataset

All three independent features were highly correlated, with more than a 99.8% correlation value. These features were important to the dependent feature as well, with a correlation value of more than 66.6%. The same features have almost the same correlation matrix for the joint dataset. So, only one feature loan\_amnt was kept, and two funded\_amnt and funded\_amnt\_inv were dropped from both datasets (individual and joint).

	fico_range_high	fico_range_low	last_fico_range_high	last_fico_range_low
fico_range_high	1.00000	1.00000	0.341440	0.340460
fico_range_low	1.00000	1.00000	0.341440	0.340460
last_fico_range_high	0.34144	0.34144	1.000000	0.999856
last_fico_range_low	0.34046	0.34046	0.999856	1.000000

Figure 4.2.6.2 Correlation matrix of FICO score for individual dataset

Fico\_range\_high was highly correlated with fico\_range\_low, and last\_fico\_range\_high was highly correlated to last\_fico\_range\_low for both individual and joint datasets. A new feature is created by averaging the high and low values to remove redundant features. A new feature, avg\_fico\_range, was created by averaging the value of fico\_range\_high and fico\_range\_low, and the new feature avg\_last\_fico\_range was derived by averaging the last\_fico\_range\_high and last\_fico\_range\_low for both datasets (individual and joint). Afterwards, duplicate features fico\_range\_high, fico\_range\_low, last\_fico\_range\_high and last\_fico\_range\_low was dropped from the both datasets. Likewise, a new feature for joint dataset sec\_avg\_last\_fico\_range was created by mean of sec\_last\_fico\_range\_low and sec\_last\_fico\_range\_high, and duplicate features were dropped from the joint dataset.

#### 4.2.7 Univariate Analysis

In univariate analysis, data analysis of a single variable was performed to showcase the data pattern based on the aim of the study. As the primary goal of this research study is to predict the credit risk of the application. As figure 4.2.7.1 mentioned 80% of data points are non-default and only 20% are default. So, for deep analysis of default account

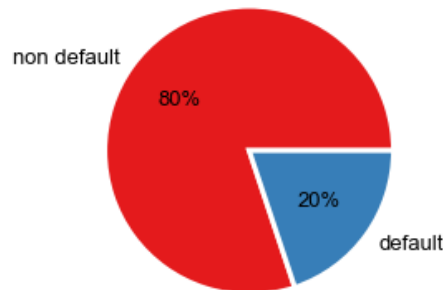


Figure 4.2.7.1 Loan status data distribution

all the variable was analysed for only default accounts trying to find the pattern of the data that might lead to fraudulent.

##### 4.2.7.1 Analysis of loan\_purpose

loan\_purpose is the reason why the customer needs a loan. And figure 4.2.7.1.1 shows a bar graph for the number of loan purposes for the default account. And from the chart, the maximum number of loans was sanctioned for debt consolidation.

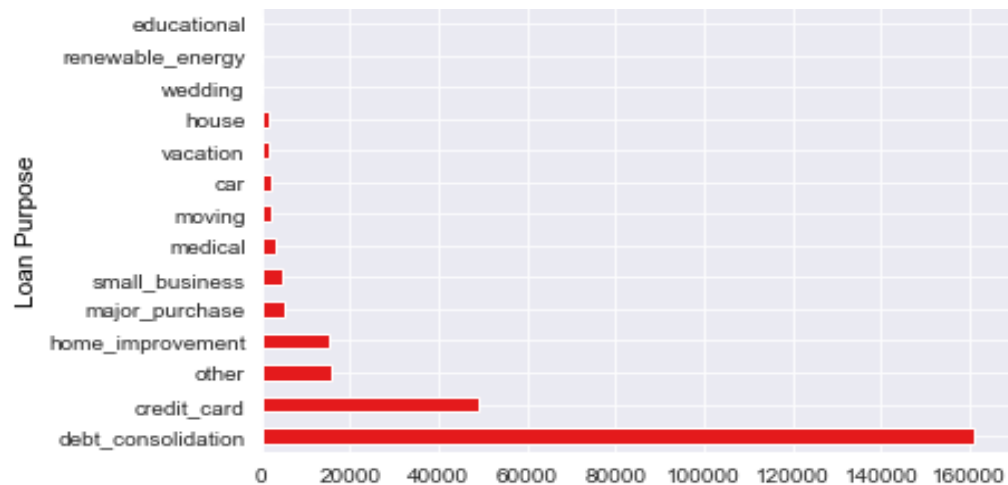


Figure 4.2.7.1.1 Analysis of loan\_purpose for default account

Debt consolidation is a way of refinancing debt. However, this kind of loan has a huge risk because it did not come with any security. And as the bar graph figure 4.2.7.1.1 shows that most of the default account loan purpose was debt consolidation, so while passing any loan proposal, if the purpose is debt\_consolidation, it needs more attention to take care of.

The most secured loan house was the least five purposes for the default account, which makes sense because a housing loan comes with security and its loan tenure is higher, and interest rate is lower, so there is a low chance of default in the case of housing loan.

#### 4.2.7.2 Analysis of home\_ownership

Homeownership features hold information about whether the customer owns the home property or the home property is already mortgaged or rents the property. Figure 4.2.7.2.1 demonstrates the data distribution of home\_ownership for default accounts using a pie chart, and it shows that 47% of customers rent the home and only 11% of customers own it.

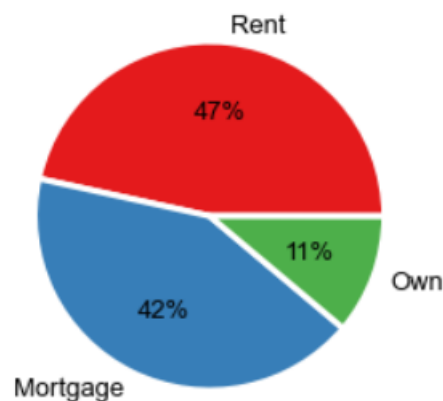


Figure 4.2.7.2.1 Analysis of home\_ownership for the default account

Renting a home or mortgage increases the monthly expenses for the customer, so sanctioning a loan to such customer is riskier than the customer who owns the property because they have lesser expenses; hence there will be a low chance that they have missed any due EMI. This is why default risk is lower for the customers who own their home property.

#### 4.2.7.3 Analysis of Interest Rate

Interest rate plays an essential role in increasing the loan risk. Lowering the interest rate reduces the EMI and less chance of default. But if there are some payment dues, the interest rate rises with penalties, which will increase the risk, and the loan may default. The interest rate depends on loan purpose, fico score, employment, and loan term. The interest rate is lowest for secured loans like home loans and highest for loans like debt consolidation or credit card or personal loans.

Figure 4.2.7.1.1 already show that most of the loan has been sanctioned for debt\_consolidation for default account; hence interest rate must be higher for the same. figure 4.2.7.3.1 show a histogram of the interest rate bin for default account, and it represents that more than 40,000 accounts have more 15% interest rate, which is very high

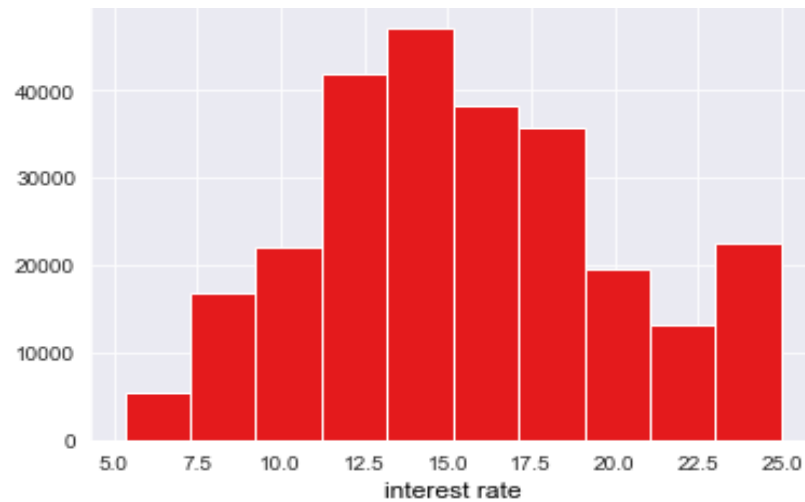


Figure 4..2.7.3.1 Analysis of an interest rate for the default account

Higher interest risk increases the loan cost for the customer, leading to the default. So before sanctioning, a loan detail analysis for interest rate analysis must be performed to reduce the default risk.

#### 4.2.7.4 Analysis of FICO score

FICO score is a credit score calculated based on the credit performance of the customer. It ranges from 300-850 lower the score higher the credit risk. It tells about the likelihood of the customer to repay the loan.

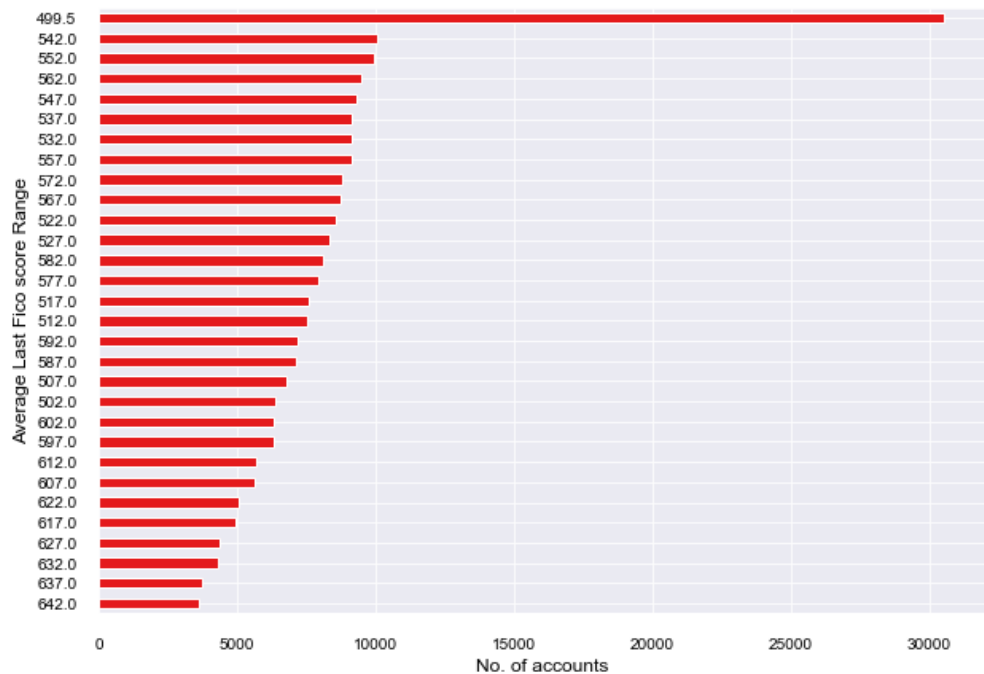


Figure 4.2.7.3.1 Analysis of Last FICO score for the default account

Figure 4.2.7.3.1 uses a bar graph to show the range of FICO score of default account, and most of the default account has a score less than 500, which is very low. So, the FICO score plays a significant role in analyzing any loan proposal; it will tell about the credit history, and the loan amount is calculated based on that interest rate. Hence, the FICO score must be checked, and if it is low, there will be a high chance of the loan default.

#### 4.2.7.5 Analysis of Bank-card usage limit

The bank card usage limit explains the expenses of the customer. There will be a high credit risk if the customer has high expenses. So analyzing the customer's expenses are also significant.

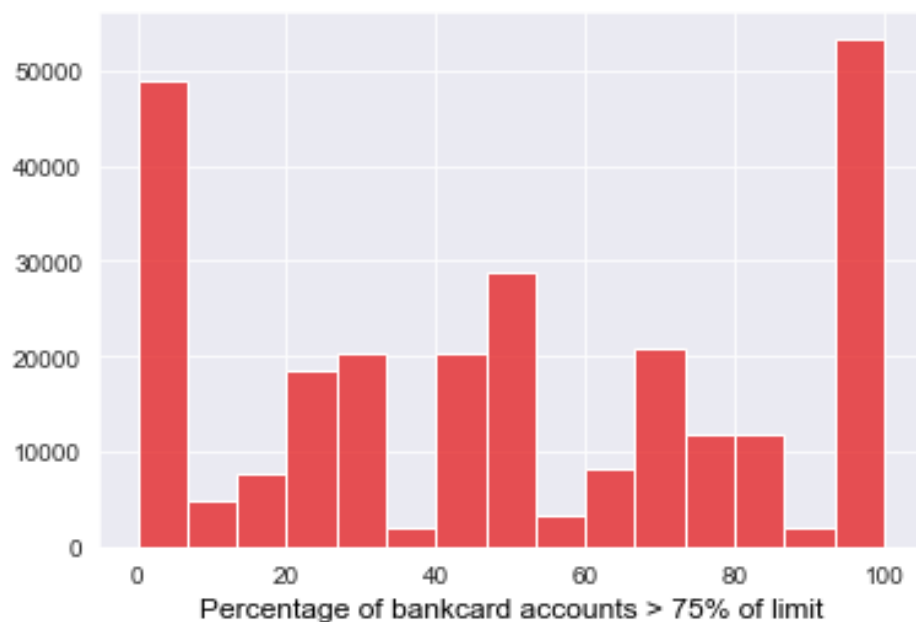


Figure 4.2.7.5.1 Analysis of the usage of bank card account limit for the default account

Figure 4.2.7.5.1 shows a histogram representation of bankcard account limit more than 75% for the default account and chart shows 100% account is more than 75% usage limit for the most of the default account holder that explains they have high monthly expenses that cause the account payment overdue and makes the account default.

If the expenses are too high, it is tough to bear the loan EMI, which will make the account default; hence, analysing the customer expenses will help in credit risk calculation.

#### 4.2.7.6 Analysis of Annual Income

Annual income is also an important feature to analyze before sanctioning the credit. It will help calculate the loan amount, debt to income ratio, and loan term. Figure 4.2.7.6.1 represents annual income data values distribution for default account using boxplot.

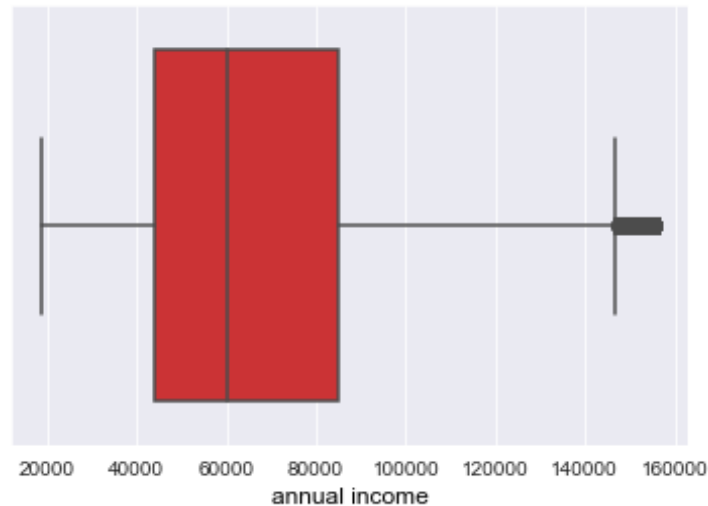


Figure 4.2.7.6.1 Analysis of annual income for the default account

The median of the boxplot is not equal to the mean value, and it is slightly skewed towards the lower whisker. The median annual income for default income is 60,000, and most of the annual income is between 45,000 to 85,000. Some outliers represent some customers with an annual income of more than 1,50,000, but they still default.

So, it is not necessary that if the customer has an excellent annual income, it can not get the default. So, annual income should be analyzed, but other factors like FICO score, expenses, and homeownership should also be considered.

#### 4.2.7.7 Analysis of dti

The debt to income (dti) ratio is the expense ratio with the annual income. The higher expenses dti will be higher. This is the best measure to calculate the monthly expense cost of any customer. This will help the firm calculate the loan proposal's credit risk.

Figure 4.2.7.7.1 explains the data distribution of dti for the default account. The overall data is normally distributed. The average dti is in the range of 15-25, which is good, but there is a high bar near about 40 towards the left side of the graph.

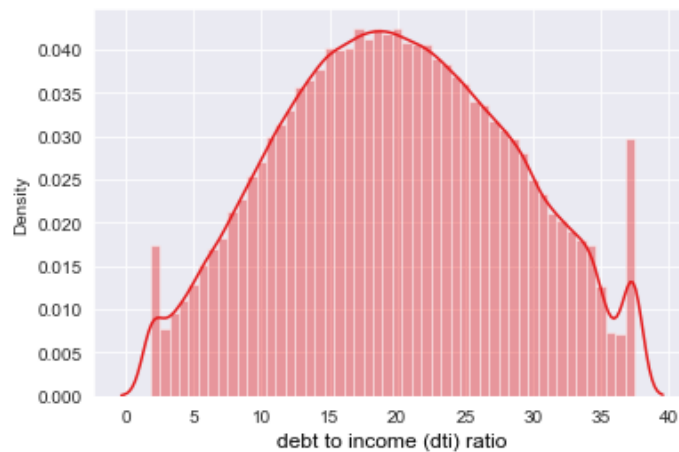


Figure 4.2.7.7.1 Analysis of debt to income ratio for the default account

Dti value 40 represents the high expense cost of the customer. So before sanctioning the loan, dti should be calculated for credit risk assessment. Higher the dti, the higher the credit risk.

#### 4.2.7.8 Analysis of loan grade

After analyzing the loan applicant's credit profile, the loan grade is assigned to each loan. Each grade has its interest rate range; grades range from A to G. Grade A has a low-interest rate, and grade G has a high-interest rate. Figure 4.2.7.8.1 shows a bar graph of the default account number in each loan grade.

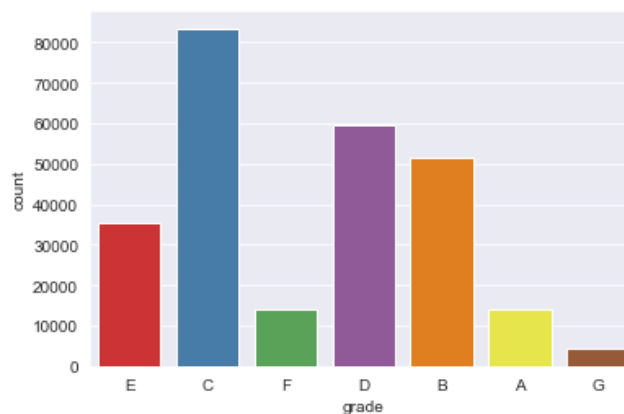


Figure 4.2.7.8.1 Analysis of loan grades for the default account

The graph shows that grade C has a maximum number of default accounts. And interest rate for loan grade C ranges from 6.0% to 17.27%. On the other hand, the least number of default accounts are in grade G and grade A, with the highest and least interest rate range.

## 4.2.8 Bivariate Analysis

After univariate analysis, it's time to explore the empirical relationship between two sets of features that can help analyze the goal of this research study. First, because of the low data points available for the default account in this section, all two variables were analysed only for the default account, trying to find the factors for the high credit risk.

### 4.2.8.1 Analysis of Loan amount with loan purpose

The bar graph represented in figure 4.2.8.1.1 shows that the maximum average loan amount has been sanctioned for small businesses for the default account. A business loan is riskier; it depends on competition, location, demand-supply. So sanctioning business loan needs lots of research about the market, customers, and products.

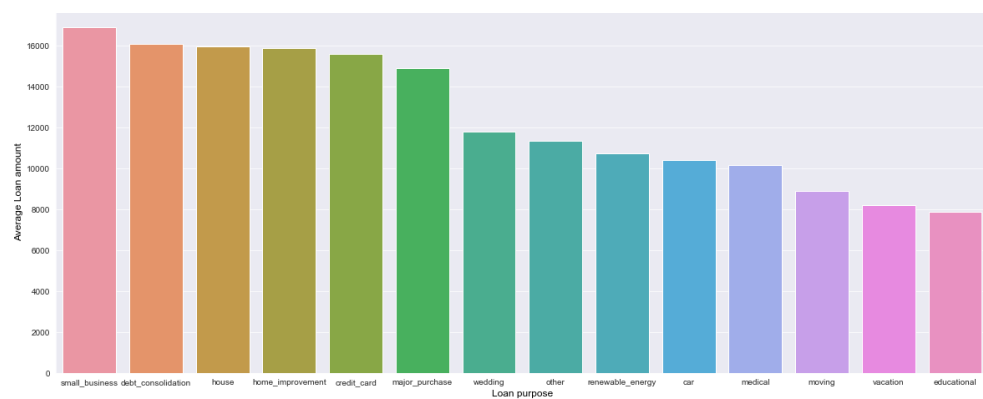


Figure 4.2.8.1.1 Analysis of loan amount with loan purpose for the default

Even the debt consolidation came second in the maximum average loan sanctioned list, and home loans came third. Therefore, for lowering the credit risk, the purpose of the loan needs to be secured, and refinancing the debt needs more detailed scrutiny of the borrower's credit profile.

### 4.2.8.2 Analysis of loan amount with FICO score

Generally, if the FICO score is higher, there is a high probability of getting a sanction high loan amount. For example, figure 4.2.8.2.1 represents a scatter plot of the average loan amount sanctioned for the FICO score range for the default account. It shows that the average loan amount increases until the range of 700 FICO score increase FICO score increases.



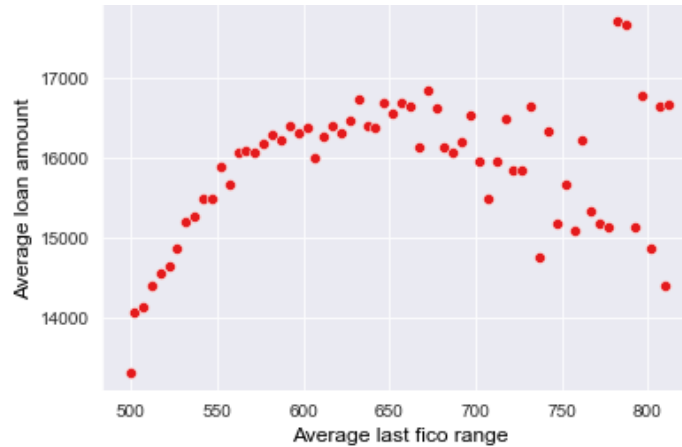


Figure 4.2.8.2.1 Analysis of avg loan amount with avg FICO score for default

But after 700, the average loan amount decreases with the higher FICO score. Some outliers can be seen within the range of 750-850. One or two has been sanctioned for an average amount of more than 17,000.

#### 4.2.8.3 Analysis of loan purpose with the interest rate

The interest rate varies according to the loan purpose. A secured loan has the lowest interest rate, and a loan with no security has the highest interest rate. Figure 4.2.8.3.1 shows a boxplot of loan purpose with interest rate range for the default account.

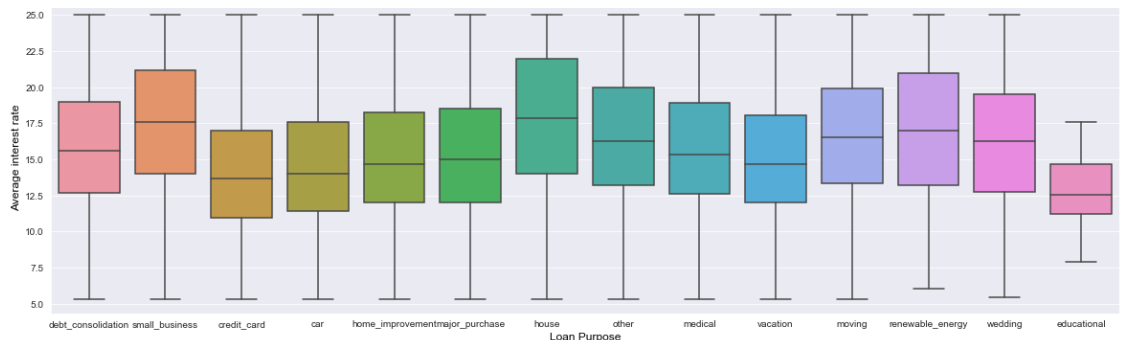


Figure 4.2.8.3.1 Analysis of loan purpose with an interest rate for the default

And loan for a house has the highest median interest rate, about 17.5%, which is a little surprising because the interest rate for housing loans is usually the lowest. And personal loan or refinancing or business loan has a high-interest rate. However, the average interest rate for debt consolidation, credit cards, or small businesses has a lower interest rate than housing. This may be because the borrower of a housing loan may have a low FICO score or their credit profile is not performing well.

#### 4.2.8.4 Analysis of interest rate with Loan term

It is exciting to analyze whether the short-term loan has a better interest rate than the long-term or not. Figure 4.2.8.4.1 explains this analysis using a boxplot between the default account's loan term and interest rate.

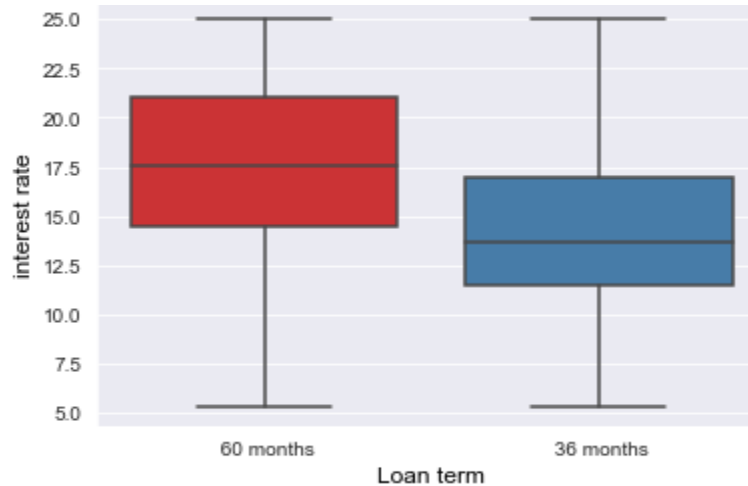


Figure 4.2.8.4.1 Analysis of interest rate with loan term for the default

The loan term 60 months has a higher median interest rate than the loan term 36 months. So, if the loan term is low, then the interest rate is low, but EMI may be increased as the loan term is short; hence borrower prefers a long term loan plan which leads to paying a higher rate of interest which make the interest amount higher than the principal and this may lead the high risk of credit because the principal amount is not repaid much because of the high-interest rate.

#### 4.2.8.5 Analysis of loan interest with the installment

if the loan term is high, the installment or EMI becomes low, and figure 4.2.8.4.1 shows that there is a higher rate of interest applicable for the long term

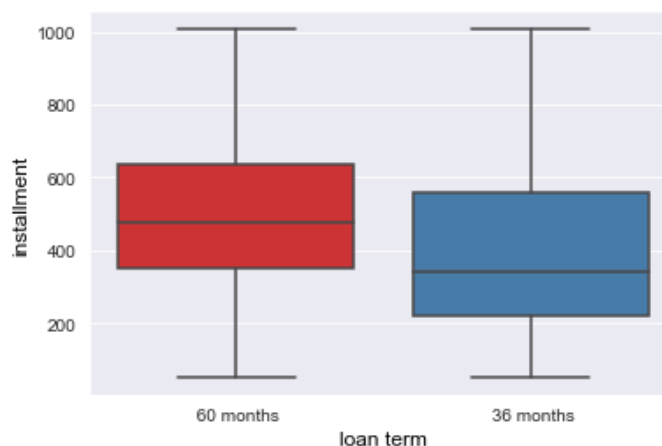


Figure 4.2.8.5.1 Analysis of loan term with installments for the default

loan for the default account, so is it better to offer a long-term loan to the borrower to reduce credit risk?

Figure 4.2.8.5.1 represents a boxplot graph for loan interest and loan installment. The relationship between these two features for the default account shows that the average installment amount is higher for the long-term loan (60 months) than the short-term (36 months) loan.

So, in conclusion, the long-term loan plan costs a higher interest rate and higher installment, which makes the borrower bear extra loan cost, which may lead to delay in EMI, and the loan may get the default.

#### 4.2.8.6 Analysis of Annual income with FICO score

Annual income does not affect the FICO score directly. However, credit scores build on the past payment done in the credit account, and payment can be affected by the annual income. So, analysing the relationship between annual income and FICO score could be interesting. Figure 4.2.8.6.1 shows a scatter plot representing a relationship between average annual income and FICO score for the default account.



Figure 4.2.8.6.1. Analysis of annual income with FICO score for the default

Till the range of 600 credit score with the increase in annual income, the credit score is also increasing, but afterward, some accounts have low average annual income with a good credit score. Their credit score is more than 700; some have about 800 credit scores, and some accounts have high scores and high annual income. So, annual income influences the FICO score, but other factors are also involved in building the FICO score.

#### 4.2.8.7 Analysis of loan amount with dti

If the dti is low, there is a high chance of sectioning the loan. High dti represents the high monthly expenses of the borrower. So using a scatterplot, figure 4.2.8.7.1 represents the relationship between dti and the average sanctioned loan amount for the default account.

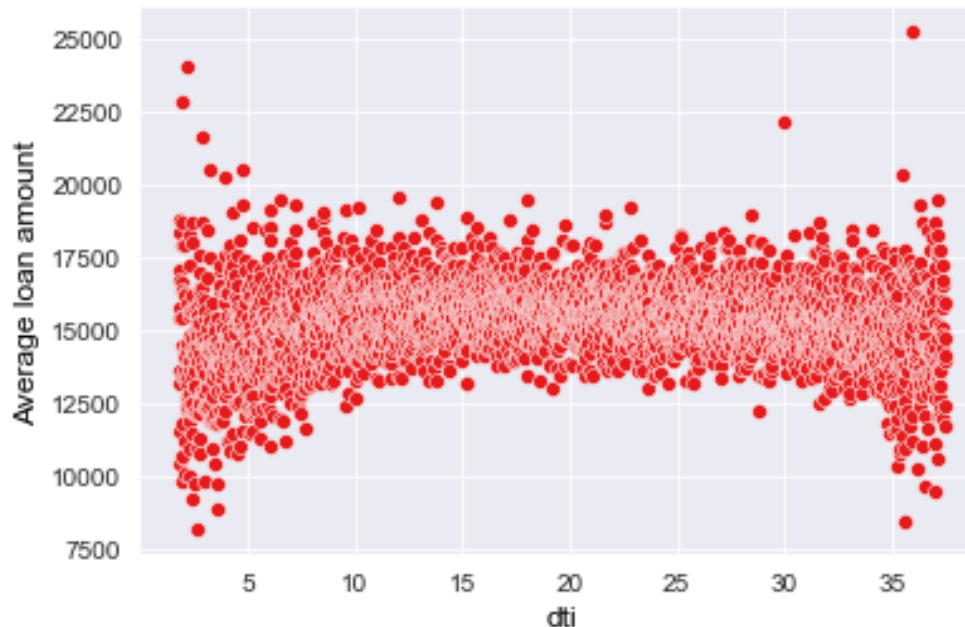


Figure 4.2.8.7.1 Analysis of loan amount with dti for the default

And there is a high amount of loans sanctioned for low dti holder customers, which is good, but for dti value of more than 25, there is still a high amount of loans approved to the customer, which increases the credit risk. The highest average loan amount has been sanctioned to the customer whose dti value is 35. This is because these loans get the default.

So, it is always better to calculate the dti value before sanctioning the loan in the credit assessment process.

#### 4.2.9 Multivariate Analysis

In univariate and bivariate analysis, features were analysed only for the default account for trying to find the pattern in data that would make the credit risk higher. Finally, a comparative analysis between default and non-default was performed using the multiple set of features in multivariate analysis.

##### 4.2.9.1 Comparative analysis of default and non-default using loan amount and loan purpose

Loan purpose is a factor that decides the credit risk of the loan application, so to analyze this feature with the default and the non-default accounts, figure 4.2.9.1.1 used a bar graph using average loan amount and loan purpose.

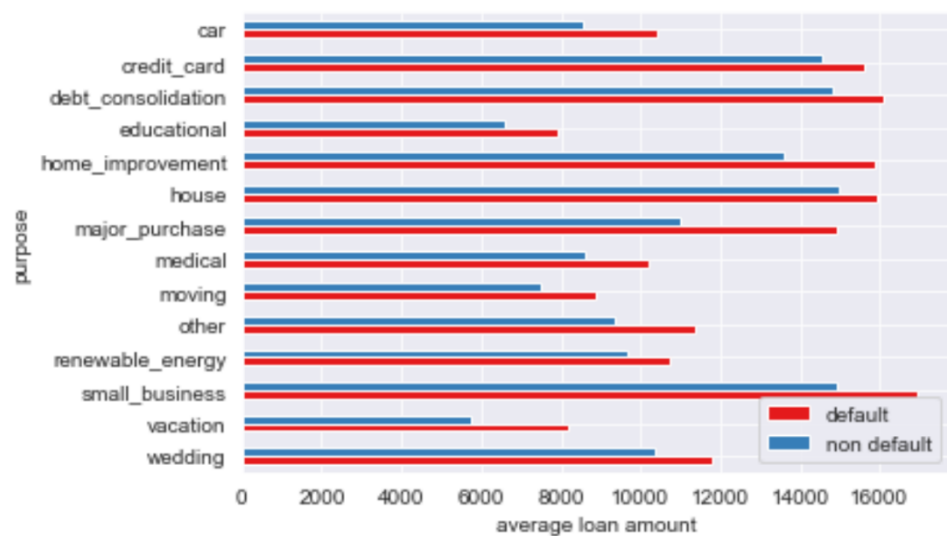


Figure 4.2.9.1.1 Analysis of Avg loan amount with loan purpose for the default and the non-default

The average loan amount is higher for the default account for all loan purposes. Small business has the highest average loan amount for default and non-default. Housing, debt consolidation is the next highest for both of the accounts. For purpose major\_purchase sanctioned loan amount is more than 14,000 for the default and the non-default accounts, the amount is less than 12,000 which is a big difference because for almost every purpose sanctioned amount is almost same with little difference for the default and non-default but major\_purchase has high amount in the default account, which makes this loan purpose riskier than others.

#### 4.2.9.2 Comparative analysis of default and non-default using homeownership

Homeownership plays an important role in calculating the credit risk of the loan proposal. It has three categories rented, mortgage, and own. Rented and mortgage add extra costs to the borrower's expenses. So, using figure 4.2.9.2.1, a comparative analysis is performed for the default and non-default accounts based on homeownership.

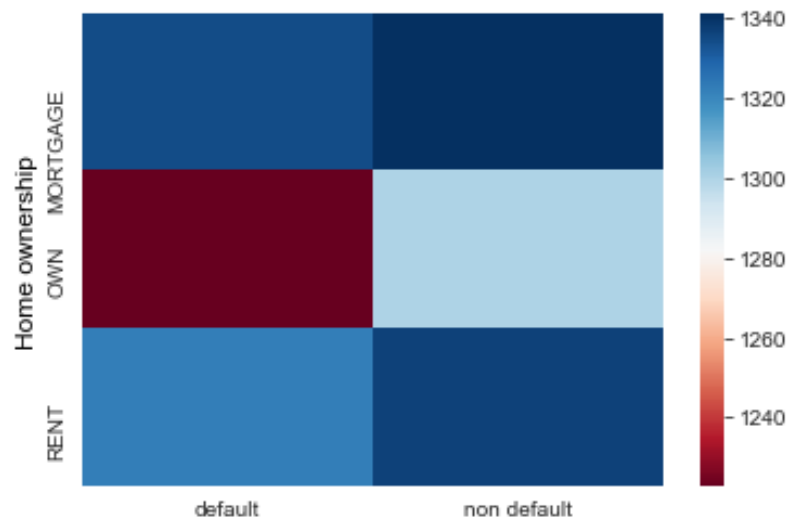


Figure 4.2.9.2.1 Analysis of homeownership for the default and the non-default

The default accounts have the least number of accounts for owning the house. And most of the borrowers rent their house for the default account. And for the non-default number of the borrower who owns their house is higher than the non-default account, and most of the borrower has more mortgage house than rent, which increases the creditworthiness of the non-default account.

#### 4.2.9.3 Comparative analysis of default and non-default with interest rate and FICO score

The interest rate is based on the FICO score; the better the score, the better the interest rate. Figure 4.2.9.3.1 represents a comparative analysis for the default and the non-default accounts based on the interest rate and FICO score.

The graph shows that as the credit score gets better, the interest rate lowers.

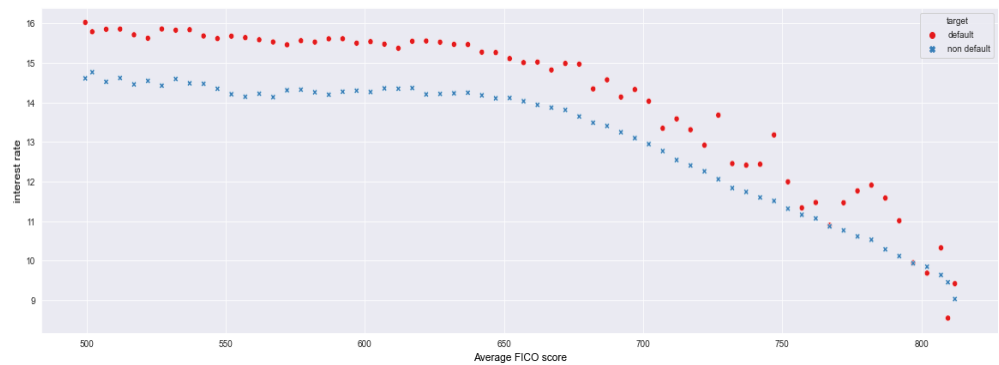


Figure 4.2.9.3.1 Analysis of interest rate with FICO score for the default and the non-default

And for the non-default account, the interest rate is always lower than the default account, which is correct because additional penal interest is applied when the account gets default. But for some default accounts, there is a high-interest rate of more than 14% despite a good credit score of more than 700, which could be because of applied penal interest.

#### 4.2.9.4 Comparative analysis of default and non-default with verification status and loan amount

Verification status is the verification of the borrower's source, which is an important step for credit assessment. And using figure 4.2.9.4.1, a comparative

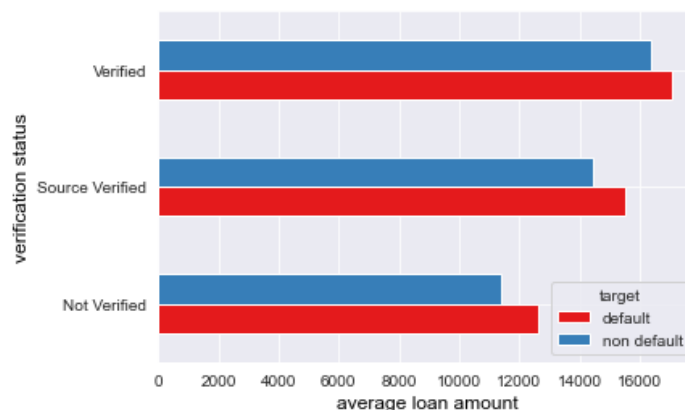


Figure 4.2.9.3.1 Analysis of verification status with avg loan amount for the default and the non-default

Analysis of the average sanctioned loan for the default and non-default is performed. The average sanctioned loan for source not verified is higher than the non-default account for the default account. If the borrower's source is not verified, it increases the credit risk and higher chance of fraud. As the graph shows, the average sanctioned loan amount is higher for all verification statuses for the default account. However, not verified source has more than an average 12,000 approved loan amount for the default account, which is quite a big amount with high credit risk, so every borrower's source must be verified to reduce the risk of the credit.

### **4.3 Preprocessing for the modeling**

After the data cleaning, data should be prepared for the modeling. Most machine learning models can interpret only numerical data, so all categorical features should be converted into numbers. Afterward, all numerical features would be scaled to normalize the data range to reduce the variance in the prediction result because of the high magnitude of the data value after the scaling step dependent feature target split from the independent feature and then train and test set was created for the further modeling process.

#### **4.3.1 Label Encoding**

Using label encoding, categorical features value was converted into a numerical value. Label encoding assigned each class label of the feature with a unique integer value based on alphabetical order. To use this technique, LabelEncoder was imported from the sklearn python library, and all the categorical features value was encoded to numerical feature for both individual and joint dataset.

Thirteen categorical features of the individual dataset and fifteen of the joint dataset were encoded. And five-date features were also converted into numerical using the strftime method for both datasets.

#### **4.3.2 Independent and dependent feature split**

The prediction result would be compared with the original value for supervised-learning dependent features extracted from the dataset to perform the modeling on the independent features. So, dependent feature target of both dataset individual and joint extracted and saved into individual\_y and joint\_y respectively. And the independent feature set was saved into individual\_X and joint\_X. The shape of individual\_X was (1319544, 65), and joint\_X was (25806, 89).

#### **4.3.3 train and test split and Feature Scaling**

Train and test sets are needed for model fitting and model evaluation. The model was performed on the train set, and on the test set, it was evaluated for unbiased results using a similar dataset. So, using train\_test\_split from python model selection library individual\_X, individual\_y, joint\_X and joint\_y split into individual\_X\_train and individual\_X\_test, individual\_y\_train and individual\_y\_test, joint\_X\_train and joint\_X\_test, joint\_y\_train and joint\_y\_test in 70:30 ratio. Train sets have 70% of data, and test sets have 30% of data. Now train dataset was used as model input, and the test set was used to evaluate the model performance.

And the final step of preprocessing is to scale the numerical data to intact the data range. Because not all numerical values use the same unit, some have high data magnitude, and some have low, and if the data is not scaled, the prediction result gets influenced by data value variance. So, to overcome this issue, all independent numerical features were scaled using the StandardScaler python library to normalize the data range. Forty-eight numerical columns were normalized of the individual dataset and seventy columns of the joint dataset.



## 4.4 Implementation

Boosting classifiers XGBoost, LightGBM, and CatBoost were fitted on the training set of the individual and joint dataset. Afterward, their prediction result was compared with each other and with the prediction result of the stacking classifier based on all three classifiers. The modeling phase for this research study was divided into two parts. Part one had no balancing technique involved, and part two had a balancing technique SMOTE and ADASYN, in which data resampling was performed. All the classifiers were learned on resampled train dataset. Finally, all the results were compared and analysed on the test dataset.

### 4.4.1 Boosting classifiers without balancing technique

Boosting classifiers were implied on the training dataset first with a default parameter, and afterward, hyperparameters were tuned. This is because the dataset has high volume and dimension, and parameter tuning for such a dataset took a lot of execution time. So, the comparative analysis of the default and tuned parameter performance would be interesting.

Initially XGBoost, LightGBM and CatBoost were initialised with default parameter and `random_state=100` using `XGBClassifier()`, `LGBMClassifier()` and `CatBoostClassifier()` imported from python package. All the classifiers were fitted for model prediction on `individual_X_train` and `individual_y_train` for the individual dataset and `joint_X_train` and `joint_y_train` for the joint dataset.

Afterward, new classifiers were initialized for each boosting classifier for hyperparameter tuning. Then, `RandomizedSearchCV` was used to tune the hyperparameter for all the classifiers using `cv=5` because cross-validation less than 5 was not enough to validate the result, and more than five was taking too much execution time. So, `cv=5` was used for each classifier to tune the parameter. Table 4.4.1.1 shows the parameter range used for XGBoost, LightGBM, and CatBoost classifier's parameter tuning.

Table 4.4.1.1 hyperparameter range for the boosting classifier

<b>XGBoost</b>	<code>learning_rate</code> : [0.05, 0.10, 0.15, 0.20, 0.25, 0.30] <code>max_depth</code> : [ 3, 4, 5, 6, 8, 10, 12, 15] <code>min_child_weight</code> : [ 1, 3, 5, 7] <code>gamma</code> : [ 0.0, 0.1, 0.2, 0.3, 0.4] <code>colsample_bytree</code> : [ 0.3, 0.4, 0.5, 0.7]
<b>LightGBM</b>	<code>num_leaves</code> : <code>sp_randint(6, 50)</code> <code>min_child_samples</code> : <code>sp_randint(100, 500)</code> <code>min_child_weight</code> : [1e-5, 1e-3, 1e-2, 1e-1, 1, 1e1, 1e2, 1e3, 1e4] <code>subsample</code> : <code>sp_uniform(loc=0.2, scale=0.8)</code> <code>colsample_bytree</code> : <code>sp_uniform(loc=0.4, scale=0.6)</code> <code>reg_alpha</code> : [0, 1e-1, 1, 2, 5, 7, 10, 50, 100] <code>reg_lambda</code> : [0, 1e-1, 1, 5, 10, 20, 50, 100]

<b>CatBoost</b>	depth :[3,1,2,6,4,5,7,8,9,10] iterations :[250,100,500,1000] learning_rate :[0.03,0.001,0.01,0.1,0.2,0.3] l2_leaf_reg :[3,1,5,10,100] border_count :[32,5,10,20,50,100,200]
-----------------	---

Hyperparameter tuning for the XGBoost classifier took the highest execution time. However, using the `best_estimator_` attribute of the random search cv, parameters were extracted for each classifier. Then, new classifiers were initialized and fitted on the train set of the individual and the joint dataset using the best parameter value.

Finally, for both datasets, predictions for train and test sets were performed, and results were compared based on multiple cost-sensitive metrics.

#### 4.4.2 Stacking classifier without balancing technique

Using the stacking ensemble classifier all three boosting classifier (XGBoost, LightGBM and CatBoost) were combined using logistic regression as meta classifier. Stacking ensemble classifier were initialized using `StackingClassifier()` from python package and `XGBClassifier(random_state=100)`, `LGBMClassifier(random_state=100)` and `CatBoostClassifier(random_state=100)` were used as base classifier and `LogisticRegression(random_state=100)` as meta classifier. Afterwards, initialized classifier was implemented on train set of the individual and the joint dataset.

Using predict function model prediction was performed for train and test set for the individual and the joint dataset.

#### 4.4.3 Boosting and stacking classifiers with balancing technique

the target feature was highly imbalanced; about 80% of class labels belonged to non-default class labels, and only 20% of data belonged to default. Usually, the machine learning model is very sensitive to class imbalance; The result may get biased towards majority class labels. So, to analyze this problem class balancing technique was used to resample the minority class by creating new synthetic data points. Two oversampling methods were used in this research study SMOTE and ADASYN The result was compared and analysed.

SMOTE was initialized using `SMOTE("minority", random_state=100)` and ADASYN was initialized using `ADASYN(sampling_strategy='minority', random_state=100, n_neighbors=4)` from the python package, `sampling_strategy` was defined as 'minority' for both of the oversampling methods because resampling was needed to perform only for minority class labels. Afterward, both oversampling methods were fitted on the `individual_X_train` and the `individual_y_train` for the individual dataset and the `joint_X_train` and the `joint_y_train` for the joint dataset. And new resampled train dataset was recreated, and all three boosting classifiers (XGBoost, LightGBM, and CatBoost) and stacking classifier were implied on the newly resampled train set for both datasets.

After model fitting prediction for resampled train and test set was performed, results were evaluated based on confusion matrix and multiple cost-sensitive metrics.

#### 4.5 Model Interpretation

One of the objectives of this research study is to interpret the black-box model's result. And the result of the implemented model, XGBoost, LightGBM, CatBoost, and stacking classifiers, could not be interpretable in human terms. To resolve this issue, SHAP was used to interpret the result of the classifiers. Two python packages were imported to use the SHAP features, shap, and SmartExplainer from shapash. Shapash has advanced interpretable summary reports for better understanding. Finally, the model interpretation was performed for all three boosting classifiers with and without the balancing technique for the individual and the joint dataset.

SmartExplainer() was initialized, and it was compiled with the test set using boosting (XGBoost, LightGBM and CatBoost) classifier and afterward compiled result was used with to\_smartpredictor() method to interpret the result and then using detail\_contributions() method all features contribution was interpreted in tabular form for the predicted result for both datasets (individual and joint).

For graphical representation shap.Explainer() was initialized using boosting (XGBoost, LightGBM, and CatBoost) classifier, and shap values were calculated for the test set for all the features. Afterward, the waterfall and force graph first value of shap for all the features was plotted for interpretation. Finally, shap values were used to calculate each feature contribution for the predicted result.

#### 4.6 Summary

In this chapter, the lending-club dataset was analysed in detail. The dataset was split into individual and joint datasets for better model implementation based on the application type. Null values, outliers, and redundant features were identified and handled properly in the EDA step. Higher missing values (> 50%) columns were removed from the dataset, and rest values were imputed using mean and mode function for numerical and categorical features, respectively. Outliers were identified using boxplot, and they were capped to intact the data range. And redundant columns were identified and removed. Datasets were properly cleaned from missing values, outliers, skewed columns, and redundant features.

Afterward, based on the objective of this research study, univariate and bivariate analyses were performed for default accounts to find out the data pattern in some of the important features that increased the credit risk. And the comparative analysis was performed between default and non-default accounts in multivariate analysis based on some of the feature sets.

For further modeling, data was prepared by label encoding of the categorical features. First, the dataset was split into train and test set in 70:30 ratio for both datasets (individual and joint). Then, to reduce the variance in the prediction result, data values were normalized using a standard scaling process.

To analyze the data imbalance, issue the boosting classifiers (XGBoost, LightGBM, and CatBoost) and stacking classifier was implied on the imbalanced and oversampled datasets.

For oversampling, SMOTE and ADASYN were used on minority class labels.

Finally, SHAP and shapash were used to interpret the classification result for each boosting classifier for both datasets.

## Chapter 5: Results and Discussions

### 5.1 Introduction

After the model implementation the next step is to evaluate the prediction result. In this chapter prediction result (Confusion matrix) of the train set for the individual and the joint datasets will be discussed. Performance of boosting classifiers (XGBoost, LightGBM and CatBoost) based on before and after hyper parameter tuning and before and after balancing techniques (SMOTE and ADASYN) will be compared with stacking classifier prediction result for both dataset (individual and joint).

After the analysis of train set the performance of the classifiers will be evaluated on similar data using test set of the individual and the joint dataset. Cost-sensitive metrics like AUC score, F1-Score, G-mean, Precision, Recall, Type-I and Type-II error will evaluate the model performance on the test dataset using the confusion matrix. As one of the objectives of this research study is to reduce the misclassification cost that is caused due to class imbalance problem, so the type-I and type-II error metrics will help to evaluate the misclassification cost for each model. For this research study type-I error is the misclassification cost of the default class label predicted as non-default and type-II error is the misclassification cost of the non-default predicted as default. Based on the lowest misclassification cost and highest other cost-sensitive metrics best model will be identified for the individual and joint dataset.

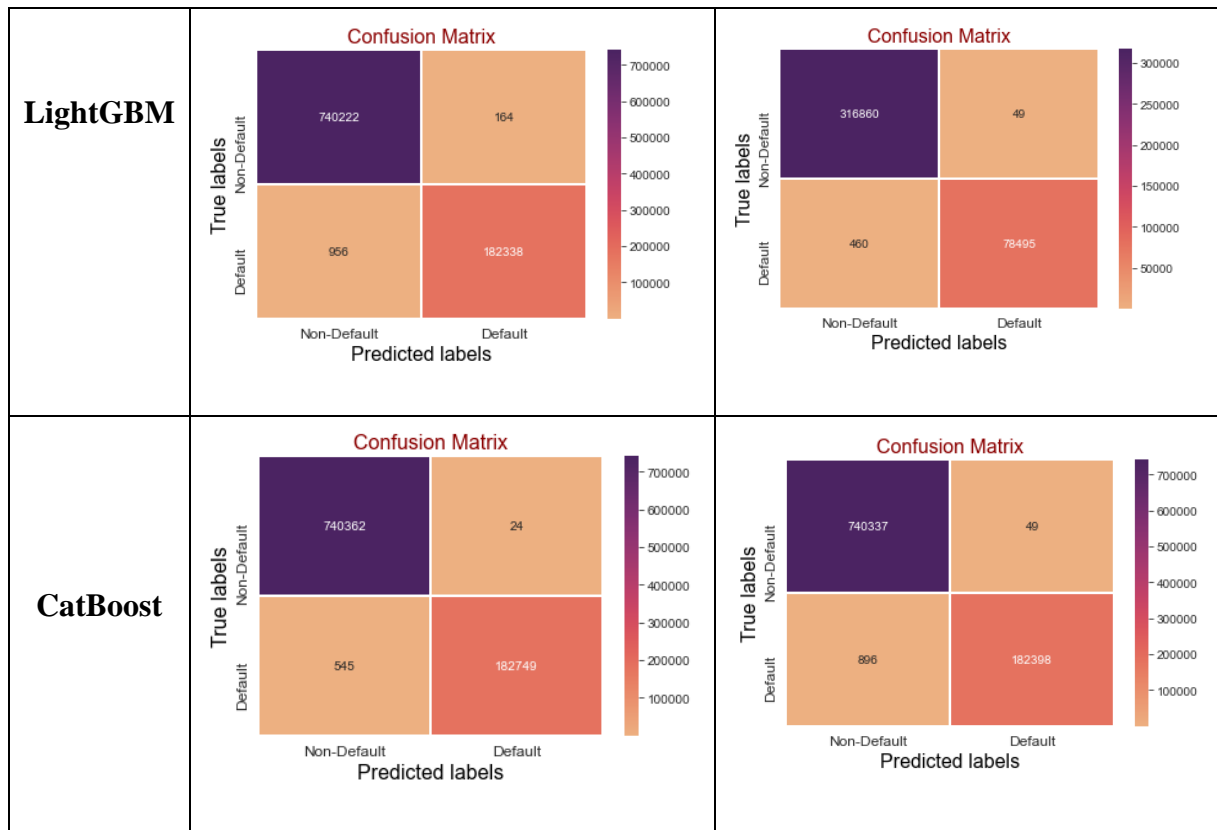
Afterward, interpretability of the model prediction result will be explained based on the SHAP.

### 5.2 Prediction result of Boosting classifiers before balancing technique

The three-boosting classifiers XGBoost, LightGBM, and CatBoost, were implemented on train sets of the individual and joint dataset with and without hyperparameter tuning. Table 5.2.1 shows the confusion matrix for the individual's train set prediction result before and after parameter tuning.

Table 5.2.1 confusion matrix of individual's train set of boosting classifier

Model	Before Parameter Tuning	After Parameter Tuning																		
<b>XGBoost</b>	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True labels \ Predicted labels</th> <th>Non-Default</th> <th>Default</th> </tr> </thead> <tbody> <tr> <th>Non-Default</th> <td>740381</td> <td>5</td> </tr> <tr> <th>Default</th> <td>157</td> <td>183137</td> </tr> </tbody> </table>	True labels \ Predicted labels	Non-Default	Default	Non-Default	740381	5	Default	157	183137	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True labels \ Predicted labels</th> <th>Non-Default</th> <th>Default</th> </tr> </thead> <tbody> <tr> <th>Non-Default</th> <td>740381</td> <td>5</td> </tr> <tr> <th>Default</th> <td>390</td> <td>182904</td> </tr> </tbody> </table>	True labels \ Predicted labels	Non-Default	Default	Non-Default	740381	5	Default	390	182904
True labels \ Predicted labels	Non-Default	Default																		
Non-Default	740381	5																		
Default	157	183137																		
True labels \ Predicted labels	Non-Default	Default																		
Non-Default	740381	5																		
Default	390	182904																		



And the results show that the performance of LightGBM was improved after the parameter tuning for the individual dataset. But for XGBoost and CatBoost, the false positive and false negative were increased after the parameter tuning. This was because of data imbalance problem, the machine learning models are sensitive to the class imbalance problem. These classifiers were biased towards the majority class and default class labels were misclassified as non-default, increasing the type I error.

Overall, XGBoost performance for the individual dataset was better before hyper-parameter tuning.

The joint dataset had a high dimension (90 features) but a small data volume (25,806 rows). First, the performance of the boosting classifiers was analyzed on a high volume, the individual dataset (13,19,544 rows). Now its performance was analyzed on a small volume. Table 5.2.2 shows the confusion matrix of the boosting classifier's prediction result for the joint's train set before and after parameter tuning.

For the joint train set before the parameter tuning, all class labels were predicted perfectly for XGBoost and LightGBM. For the CatBoost, it was almost perfect; only one default class label incorrectly predicted non-default. This is because there is a high chance of overfitting for a small dataset, and from the prediction result of the joint dataset from table 5.2.2, it was observed. After the parameter tuning, small misclassification happened only for the default class labels, and XGBoost's performance was better among all three.

Table 5.2.2 confusion matrix of joint's train set of boosting classifier

Model	Before Parameter Tunning	After Parameter Tunning																		
<b>XGBoost</b>	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True labels \ Predicted labels</th><th>Non-Default</th><th>Default</th></tr> </thead> <tbody> <tr> <th>Non-Default</th><td>13651</td><td>0</td></tr> <tr> <th>Default</th><td>0</td><td>4413</td></tr> </tbody> </table>	True labels \ Predicted labels	Non-Default	Default	Non-Default	13651	0	Default	0	4413	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True labels \ Predicted labels</th><th>Non-Default</th><th>Default</th></tr> </thead> <tbody> <tr> <th>Non-Default</th><td>13651</td><td>0</td></tr> <tr> <th>Default</th><td>9</td><td>4404</td></tr> </tbody> </table>	True labels \ Predicted labels	Non-Default	Default	Non-Default	13651	0	Default	9	4404
True labels \ Predicted labels	Non-Default	Default																		
Non-Default	13651	0																		
Default	0	4413																		
True labels \ Predicted labels	Non-Default	Default																		
Non-Default	13651	0																		
Default	9	4404																		
<b>LightGBM</b>	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True labels \ Predicted labels</th><th>Non-Default</th><th>Default</th></tr> </thead> <tbody> <tr> <th>Non-Default</th><td>13651</td><td>0</td></tr> <tr> <th>Default</th><td>0</td><td>4413</td></tr> </tbody> </table>	True labels \ Predicted labels	Non-Default	Default	Non-Default	13651	0	Default	0	4413	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True labels \ Predicted labels</th><th>Non-Default</th><th>Default</th></tr> </thead> <tbody> <tr> <th>Non-Default</th><td>13651</td><td>0</td></tr> <tr> <th>Default</th><td>20</td><td>4393</td></tr> </tbody> </table>	True labels \ Predicted labels	Non-Default	Default	Non-Default	13651	0	Default	20	4393
True labels \ Predicted labels	Non-Default	Default																		
Non-Default	13651	0																		
Default	0	4413																		
True labels \ Predicted labels	Non-Default	Default																		
Non-Default	13651	0																		
Default	20	4393																		
<b>CatBoost</b>	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True labels \ Predicted labels</th><th>Non-Default</th><th>Default</th></tr> </thead> <tbody> <tr> <th>Non-Default</th><td>13651</td><td>0</td></tr> <tr> <th>Default</th><td>1</td><td>4412</td></tr> </tbody> </table>	True labels \ Predicted labels	Non-Default	Default	Non-Default	13651	0	Default	1	4412	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True labels \ Predicted labels</th><th>Non-Default</th><th>Default</th></tr> </thead> <tbody> <tr> <th>Non-Default</th><td>13651</td><td>0</td></tr> <tr> <th>Default</th><td>16</td><td>4397</td></tr> </tbody> </table>	True labels \ Predicted labels	Non-Default	Default	Non-Default	13651	0	Default	16	4397
True labels \ Predicted labels	Non-Default	Default																		
Non-Default	13651	0																		
Default	1	4412																		
True labels \ Predicted labels	Non-Default	Default																		
Non-Default	13651	0																		
Default	16	4397																		

### 5.3 Prediction result of Stacking classifier before balancing technique

The stacking model of XGBoost, LightGBM, and CatBoost using logistic regression as meta classifier was implemented on the train set of the individual and the joint dataset. Table 5.3.1 shows the confusion matrix of the model's prediction.

Table 5.3.1 confusion matrix of stacking classifier for the train set

Model	Individual Train set	Joint Train set																		
Stacking	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True labels \ Predicted labels</th><th>Non-Default</th><th>Default</th></tr> </thead> <tbody> <tr> <th>Non-Default</th><td>740381</td><td>5</td></tr> <tr> <th>Default</th><td>157</td><td>183137</td></tr> </tbody> </table>	True labels \ Predicted labels	Non-Default	Default	Non-Default	740381	5	Default	157	183137	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True labels \ Predicted labels</th><th>Non-Default</th><th>Default</th></tr> </thead> <tbody> <tr> <th>Non-Default</th><td>13651</td><td>0</td></tr> <tr> <th>Default</th><td>0</td><td>4413</td></tr> </tbody> </table>	True labels \ Predicted labels	Non-Default	Default	Non-Default	13651	0	Default	0	4413
True labels \ Predicted labels	Non-Default	Default																		
Non-Default	740381	5																		
Default	157	183137																		
True labels \ Predicted labels	Non-Default	Default																		
Non-Default	13651	0																		
Default	0	4413																		

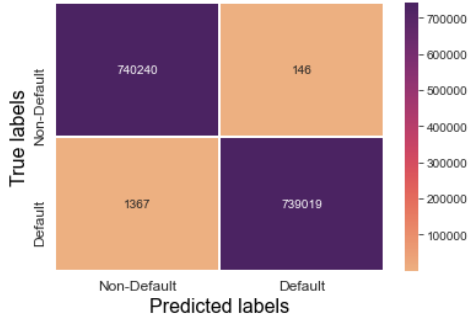
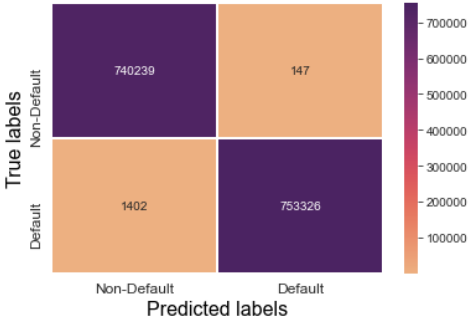
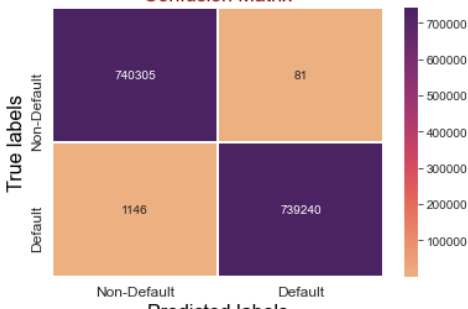
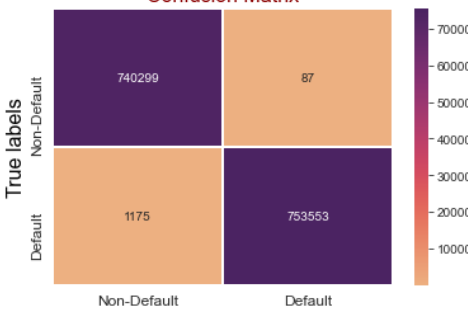
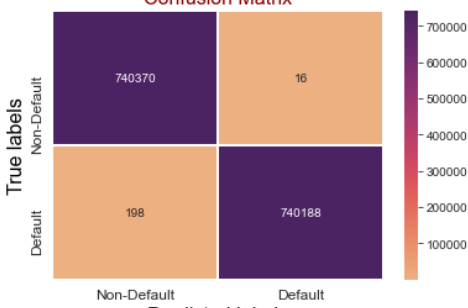
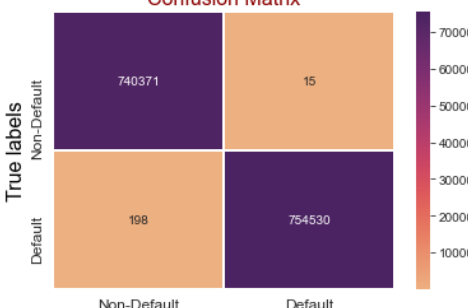
The confusion matrix for the individual train set of stacking classifiers is the same as the XGBoost prediction result before parameter tuning. And for the joint train set, the class labels were predicted perfectly. For the joint dataset, almost all the classifiers showed the perfect prediction result; after analysing the performance of these classifiers on the test set, it will confirm whether it is a case of overfitting.

#### 5.4 Prediction result after balancing technique

Two oversampling methods were used, SMOTE and ADASYN, to create new sample data for the individual and the joint train set. Afterward, all three boosting and stacking classifiers were implemented. Table 5.4.1 shows the confusion matrix result of the prediction for the individual train set.

Table 5.4.1 Confusion matrix of classifiers after balancing of individual's train set

Model	SMOTE	ADASYN																		
XGBoost	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True labels \ Predicted labels</th><th>Non-Default</th><th>Default</th></tr> </thead> <tbody> <tr> <th>Non-Default</th><td>740370</td><td>16</td></tr> <tr> <th>Default</th><td>198</td><td>740188</td></tr> </tbody> </table>	True labels \ Predicted labels	Non-Default	Default	Non-Default	740370	16	Default	198	740188	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True labels \ Predicted labels</th><th>Non-Default</th><th>Default</th></tr> </thead> <tbody> <tr> <th>Non-Default</th><td>740371</td><td>15</td></tr> <tr> <th>Default</th><td>198</td><td>754530</td></tr> </tbody> </table>	True labels \ Predicted labels	Non-Default	Default	Non-Default	740371	15	Default	198	754530
True labels \ Predicted labels	Non-Default	Default																		
Non-Default	740370	16																		
Default	198	740188																		
True labels \ Predicted labels	Non-Default	Default																		
Non-Default	740371	15																		
Default	198	754530																		

<b>LightGBM</b>	<p>Confusion Matrix</p> 	<p>Confusion Matrix</p> 
<b>CatBoost</b>	<p>Confusion Matrix</p> 	<p>Confusion Matrix</p> 
<b>Stacking</b>	<p>Confusion Matrix</p> 	<p>Confusion Matrix</p> 

The confusion matrix for the train set of the individual dataset after the oversampling techniques (SMOTE and ADASYN) was not performing as well as before the balancing technique for all the classifiers.

Table 5.4.2 shows the prediction result for the joint train set after the balancing technique was implemented, and for each classifier, except CatBoost all class labels were predicted correctly. This is because of the low number of rows.

Even after the balancing techniques were applied, the prediction result for the small dataset (the joint dataset) was the same. All the case labels were predicted perfectly for all the classifiers; if it is the case of overfitting or not will be confirmed after analyzing the test set of the joint dataset.



Table 5.4.2 Confusion matrix of classifiers after balancing of joint's train set

Model	SMOTE	ADASYN																		
<b>XGBoost</b>	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True \ Pred</th> <th>Non-Default</th> <th>Default</th> </tr> </thead> <tbody> <tr> <th>Non-Default</th> <td>13651</td> <td>0</td> </tr> <tr> <th>Default</th> <td>0</td> <td>13651</td> </tr> </tbody> </table>	True \ Pred	Non-Default	Default	Non-Default	13651	0	Default	0	13651	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True \ Pred</th> <th>Non-Default</th> <th>Default</th> </tr> </thead> <tbody> <tr> <th>Non-Default</th> <td>13651</td> <td>0</td> </tr> <tr> <th>Default</th> <td>0</td> <td>14204</td> </tr> </tbody> </table>	True \ Pred	Non-Default	Default	Non-Default	13651	0	Default	0	14204
True \ Pred	Non-Default	Default																		
Non-Default	13651	0																		
Default	0	13651																		
True \ Pred	Non-Default	Default																		
Non-Default	13651	0																		
Default	0	14204																		
<b>LightGBM</b>	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True \ Pred</th> <th>Non-Default</th> <th>Default</th> </tr> </thead> <tbody> <tr> <th>Non-Default</th> <td>13651</td> <td>0</td> </tr> <tr> <th>Default</th> <td>0</td> <td>13651</td> </tr> </tbody> </table>	True \ Pred	Non-Default	Default	Non-Default	13651	0	Default	0	13651	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True \ Pred</th> <th>Non-Default</th> <th>Default</th> </tr> </thead> <tbody> <tr> <th>Non-Default</th> <td>13651</td> <td>0</td> </tr> <tr> <th>Default</th> <td>0</td> <td>14204</td> </tr> </tbody> </table>	True \ Pred	Non-Default	Default	Non-Default	13651	0	Default	0	14204
True \ Pred	Non-Default	Default																		
Non-Default	13651	0																		
Default	0	13651																		
True \ Pred	Non-Default	Default																		
Non-Default	13651	0																		
Default	0	14204																		
<b>CatBoost</b>	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True \ Pred</th> <th>Non-Default</th> <th>Default</th> </tr> </thead> <tbody> <tr> <th>Non-Default</th> <td>13651</td> <td>0</td> </tr> <tr> <th>Default</th> <td>5</td> <td>13646</td> </tr> </tbody> </table>	True \ Pred	Non-Default	Default	Non-Default	13651	0	Default	5	13646	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True \ Pred</th> <th>Non-Default</th> <th>Default</th> </tr> </thead> <tbody> <tr> <th>Non-Default</th> <td>13651</td> <td>0</td> </tr> <tr> <th>Default</th> <td>6</td> <td>14198</td> </tr> </tbody> </table>	True \ Pred	Non-Default	Default	Non-Default	13651	0	Default	6	14198
True \ Pred	Non-Default	Default																		
Non-Default	13651	0																		
Default	5	13646																		
True \ Pred	Non-Default	Default																		
Non-Default	13651	0																		
Default	6	14198																		
<b>Stacking</b>	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True \ Pred</th> <th>Non-Default</th> <th>Default</th> </tr> </thead> <tbody> <tr> <th>Non-Default</th> <td>13651</td> <td>0</td> </tr> <tr> <th>Default</th> <td>0</td> <td>13651</td> </tr> </tbody> </table>	True \ Pred	Non-Default	Default	Non-Default	13651	0	Default	0	13651	<p>Confusion Matrix</p> <table border="1"> <thead> <tr> <th>True \ Pred</th> <th>Non-Default</th> <th>Default</th> </tr> </thead> <tbody> <tr> <th>Non-Default</th> <td>13651</td> <td>0</td> </tr> <tr> <th>Default</th> <td>0</td> <td>14204</td> </tr> </tbody> </table>	True \ Pred	Non-Default	Default	Non-Default	13651	0	Default	0	14204
True \ Pred	Non-Default	Default																		
Non-Default	13651	0																		
Default	0	13651																		
True \ Pred	Non-Default	Default																		
Non-Default	13651	0																		
Default	0	14204																		

The prediction was done only for the train set on which models were fitted, but this is not enough to measure the performance. The correct way to measure the model prediction result is for the test set if the model works better for train and test set both with the best performance result. The prediction result for the test set will be discussed in the next section.

### 5.5 Evaluation of model on the test set

Model evaluation on train set is not enough until its performance is validated on the similar data on which classifiers were not fitted. Such validation test set was created for the individual and the joint dataset before the model implementation. All the classifiers were used to predict the result for the test set, and the results are shown in figure 5.5.1 for the individual test set and figure 5.5.2 for the joint test set. Model accuracy and the cost-sensitive metrics AUC score, precision, recall, F1-score, G-mean, Type-I error, and Type-II error were used to evaluate the model performance based on the confusion matrix of the prediction result. As both figure 5.5.1 and figure 5.5.2 shows that the accuracy (ACC) for all classifiers is the same that is 99.90%, so model accuracy is not enough to evaluate the model's performance. Hence other cost-sensitive metrics were also used to evaluate the model's performance.

Compared to XGBoost and Stacking classifier, the LightGBM and CatBoost are not as good even after the class balancing technique. Type-I error is the highest of 0.621% for the LightGBM after the resampling technique SMOTE was used.

Model Name	ACC	Confusion Matrix				AUC	Precision	Recall	F1-score	G-mean	Type-I Error	Type-II Error
		TP	FN	FP	TN							
XGBoost	99.90%	316890	19	222	78733	99.856%	99.930%	99.994%	99.962%	99.856%	0.281%	0.006%
XGBoost-Tunned	99.90%	316892	17	455	78500	99.709%	99.857%	99.995%	99.926%	99.709%	0.576%	0.005%
XGBoost-SMOTE	99.90%	316885	24	220	78735	99.857%	99.931%	99.992%	99.962%	99.857%	0.279%	0.008%
XGBoost-ADASYN	99.90%	316884	25	238	78717	99.845%	99.925%	99.992%	99.959%	99.845%	0.301%	0.008%
Lightgbm	99.80%	316870	129	467	78488	99.684%	99.853%	99.959%	99.906%	99.684%	0.591%	0.041%
LightGBM-Tunned	99.90%	316860	49	460	78495	99.701%	99.855%	99.985%	99.920%	99.701%	0.583%	0.015%
LightGBM-SMOTE	99.90%	316844	65	490	78465	99.679%	99.846%	99.979%	99.912%	99.679%	0.621%	0.021%
LightGBM-ADASYN	99.90%	316857	52	489	78466	99.682%	99.846%	99.984%	99.915%	99.682%	0.619%	0.016%
Catboost	99.90%	316884	25	302	78653	99.805%	99.905%	99.992%	99.948%	99.805%	0.382%	0.008%
Catboost-Tunned	99.90%	316873	36	421	78534	99.728%	99.867%	99.989%	99.928%	99.727%	0.533%	0.011%
CatBoost-SMOTE	99.90%	316853	56	433	78522	99.717%	99.864%	99.982%	99.923%	99.717%	0.548%	0.018%
CatBoost-ADASYN	99.90%	316856	53	444	78511	99.710%	99.860%	99.983%	99.922%	99.710%	0.562%	0.017%
Stacking	99.90%	316890	19	222	78733	99.856%	99.930%	99.994%	99.962%	99.856%	0.281%	0.006%
Stacking-SMOTE	99.90%	316885	24	220	78735	99.857%	99.931%	99.992%	99.962%	99.857%	0.279%	0.008%
Stacking-ADASYN	99.90%	316884	25	238	78717	99.845%	99.925%	99.992%	99.959%	99.845%	0.301%	0.008%

Figure 5.5.1 Cost-sensitive metrics result for the individual test set

The prediction result for the stacking classifier is the same as the XGBoost classifier; hence XGBoost and Stacking classifier with the SMOTE technique gave the highest AUC score of 99.857%, and type-I error is the lowest is 0.279%. Their other cost-sensitive metrics, F1-Score, G-Mean, Precision, and Recall, also gave the highest result that is 99.962%, 99.857%, 99.931%, and 99.992%, respectively. Their type-II error was the second-lowest of 0.008%, which is not bad.

One of the objectives of this research study is to improve the model's performance using the oversampling technique. And after the result analysis of the individual's test set after using the resampling technique, the performance of XGBoost and Stacking classifier improved type-I error was reduced from 0.281% to 0.279%. Of course, the improvement was little, but small

improvements still make a big difference in credit risk. But for the LightGBM and CatBoost, after the oversampling technique, the type-I error was increased from 0.591% to 0.621% for LightGBM, and for CatBoost, it was increased from 0.533% to 0.562%. So overall, LightGBM and CatBoost are not performing as well as XGBoost with the imbalance and high volume dataset.

The performance of the stacking classifiers was influenced because of integration of the XGBoost.

Confusion matrix results for the joint train set for most of the classifiers had perfectly predicted labels, and there was no misclassification for any labels except for CatBoost. The confusion matrix of the CatBoost had only 5 or 6 misclassified default labels predicted as non-default. Figure 5.5.2 shows the prediction result for the joint's test set and CatBoost classifier with the resampling technique. SMOTE with CatBoost had the lowest type-I error of 0.361%, and its AUC score, precision, recall, and G-mean are the highest, 99.793%, 99.879%, 99.948%, and 99.793%, respectively. F1-score and Type-II error value is the second-best value for the CatBoost-SMOTE.

Model Name	ACC	Confusion Matrix				AUC	Precision	Recall	F1-score	G-mean	Type-I Error	Type-II Error
		TP	FN	FP	TN							
XGBoost	99.80%	5803	2	11	1926	99.699%	99.811%	99.966%	99.888%	99.698%	0.568%	0.034%
XGBoost-Tunned	99.80%	5802	3	12	1925	99.664%	99.794%	99.948%	99.871%	99.664%	0.620%	0.052%
XGBoost-SMOTE	99.80%	5803	2	11	1926	99.699%	99.811%	99.966%	99.888%	99.698%	0.568%	0.034%
XGBoost-ADASYN	99.90%	5805	0	12	1925	99.690%	99.794%	100.000%	99.897%	99.690%	0.620%	0.000%
Lightgbm	99.90%	5804	1	9	1928	99.759%	99.845%	99.983%	99.914%	99.759%	0.465%	0.017%
LightGBM-Tunned	99.80%	5803	2	10	1927	99.725%	99.828%	99.966%	99.897%	99.724%	0.516%	0.034%
LightGBM-SMOTE	99.90%	5804	1	9	1928	99.759%	99.845%	99.983%	99.914%	99.759%	0.465%	0.017%
LightGBM-ADASYN	99.90%	5805	0	11	1926	99.716%	99.811%	100.000%	99.905%	99.716%	0.568%	0.000%
Catboost	99.80%	5803	2	10	1927	99.725%	99.828%	99.966%	99.897%	99.724%	0.516%	0.034%
Catboost-Tunned	99.40%	5780	25	20	1917	99.268%	99.655%	99.569%	99.612%	99.268%	1.033%	0.431%
CatBoost-SMOTE	99.90%	5802	3	7	1930	<b>99.793%</b>	<b>99.879%</b>	<b>99.948%</b>	99.914%	<b>99.793%</b>	<b>0.361%</b>	<b>0.052%</b>
CatBoost-ADASYN	99.80%	5805	0	13	1924	99.664%	99.777%	100.000%	99.888%	99.664%	0.671%	0.000%
Stacking	99.90%	5804	1	9	1928	99.759%	99.845%	99.983%	99.914%	99.759%	0.465%	0.017%
Stacking-SMOTE	99.90%	5804	1	8	1929	99.785%	99.862%	99.983%	<b>99.923%</b>	99.785%	0.413%	0.017%
Stacking-ADASYN	99.80%	5805	0	13	1924	99.664%	99.777%	100.000%	99.888%	99.664%	0.671%	0.000%

Figure 5.5.2 Cost-sensitive metrics result for the joint test set

After the ADASYN resampling type-II error is 0% for all the classifiers, ADASYN worked well for both the majority class labels for the train and test dataset. However, the minority class labels were not performing as well as the majority class labels.

For a small dataset, the resampling technique only improves the performance of the CatBoost classifiers, but for other classifiers, the result was the same or poorer. And for the CatBoost classifier, the type-I error was improved from 1.033% to 0.361% after using the SMOTE.

The type-I error percentage for a small dataset (joint test set) is the same as the big dataset (individual test set), but no misclassification happened for a small train set. All the class labels were predicted accurately for XGBoost, LightGBM, and Stacking classifier. But the variation in the result of the train and test set for the small dataset (the joint) was not very much high; hence these classifiers were not facing the issue of overfitting.

## 5.6 Result Analysis

Multiple studies worked on the credit assessment model. Using table 5.6.1, a performance comparison is performed. The table shows the evaluation score performed on the same dataset used in this study (Lending Club). Some of the studies proposed cost-sensitive neural-network (Chengeta and Mabika, 2021a; Yotsawat et al., 2021a), and the study (He et al., 2018) proposed The Extended Balance Cascade approach (EBCA) used the undersampling method to handle the imbalance ratio, and stacking classifiers based on random forest and XGBoost were used for model prediction. Both studies achieved an AUC score not more than 75% using EBCA with stacking, the AUC reached 73.06%, and the F1-score is 99.38%. But the undersampling with stacking did not provide a good G-mean score, which is an important measure in data imbalance to calculate the misclassification cost. Higher the G-mean better the model.

Table 5.6.1 Comparison of performance

Paper	Model	AUC	F1-Score	G-mean	Sensitivity	Specificity
<b>Best performing model in the previous study</b>						
(Yotsawat et al., 2021a)	CS-NNE	70.82	-	65.00	62.69	67.41
	XGBoost	69.69	-	30.87	9.79	98.21
(He et al., 2018)	EBCA	73.06	99.38	1.638	-	-
	XGBoost	71.45	99.35	0.00	-	-
(Chengeta and Mabika, 2021a)	CNN	99.74	95.39	-	92.30	-
	XGB	99.44	86.23	-	94.30	-
	LightGBM	99.76	89.54	-	97.04	-
	CatBoost	99.55	87.19	-	99.00	-
(Kun et al., 2020a)	Stacking	98.11	98.32	-	98.68	-
	XGBoost	97.69	97.95	-	98.40	-
<b>Best performing model in this study</b>						
Individual Dataset	XGBoost-SMOTE	99.86	99.96	99.86	99.92	99.72
	Stacking-SMOTE	99.86	99.96	99.86	99.92	99.72
Joint Dataset	CatBoost-SMOTE	99.79	99.91	99.79	99.95	99.64

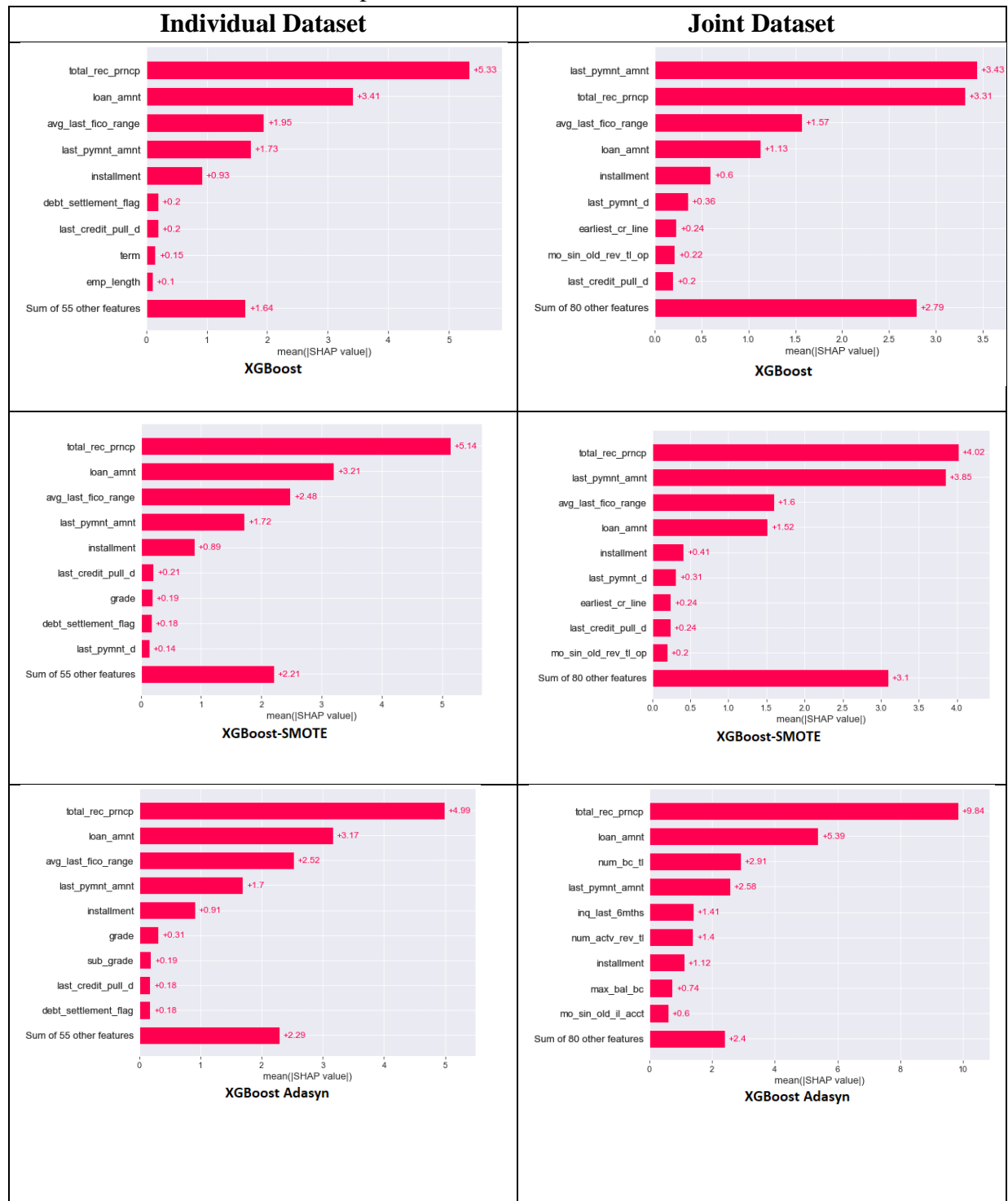
(Kun et al., 2020a) proposed a Stacking classification model based on ANN, RF, AdaBoost, and XGBoost with oversampling method SMOTE. The proposed model achieved a good AUC score of 98.11%, F1-score of 98.32%, and sensitivity of 98.68%.

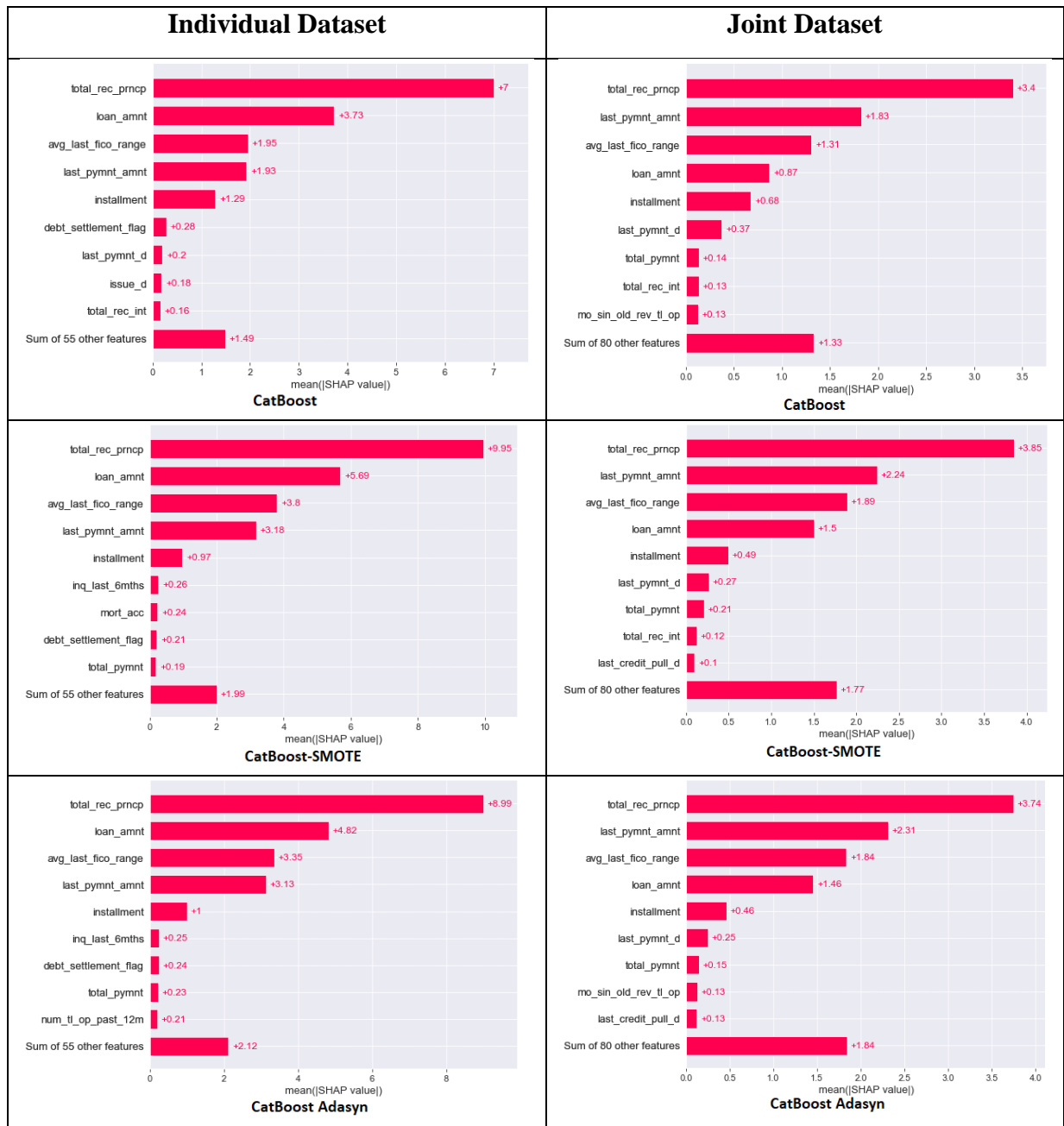
This research study achieved the highest AUC score 99.86% compared to other credit assessment models using XGBoost with smote. The F1-score is 99.96%, G-mean 99.86%, sensitivity 99.92% and specificity is 99.72%. the imbalance problem was handled using SMOTE, and extreme boosting tree helped in reducing the overfitting caused due to resampling of the minority class. The CatBoost with smote achieved the highest AUC score of 99.79% for small volume and other cost-sensitive metrics results are higher than other studies.

## 5.7 Interpretation of the model result

Interpretation of the prediction result is an important step; if the applicant gets to know the reason behind their loan rejection, then the trust in the assessment model increase. So to interpret the prediction result of the black-box model (XGBoost, LightGBM, CatBoost), additional explainable AI SHAP were used to explain the result in terms of feature contribution in the prediction. Table 5.7.1 shows the top 10 features contribution with their average contribution value (shap value) for the different classifiers.

Table 5.7.1 Contribution of the top 10 features





the top 5 features (total\_rec\_prncp, loan\_amnt, avg\_last\_fico\_range, last\_pymnt\_amnt and installment) are same for all the classifiers and for both datasets. Their average contribution values (SHAP value) are different, positively impacting the prediction result. This is the advantage of SHAP explainable AI that does not depend on the model. Its interpretation is the same for all the classifiers.

Figure 5.7.1 represents a waterfall and force plot of the feature contributions of the prediction result of a first row of the individual test set. The waterfall plot of shap represents the top 9 features that contribute to the prediction result. Y-axis shows the feature value of the dataset, and the bars represent their contribution. The blue bar of the waterfall represents the feature's negative impact, and the red bar shows the positive impact of features on the prediction result. the value  $f(x)$  represents the prediction result probabilities if it is less than 0.5

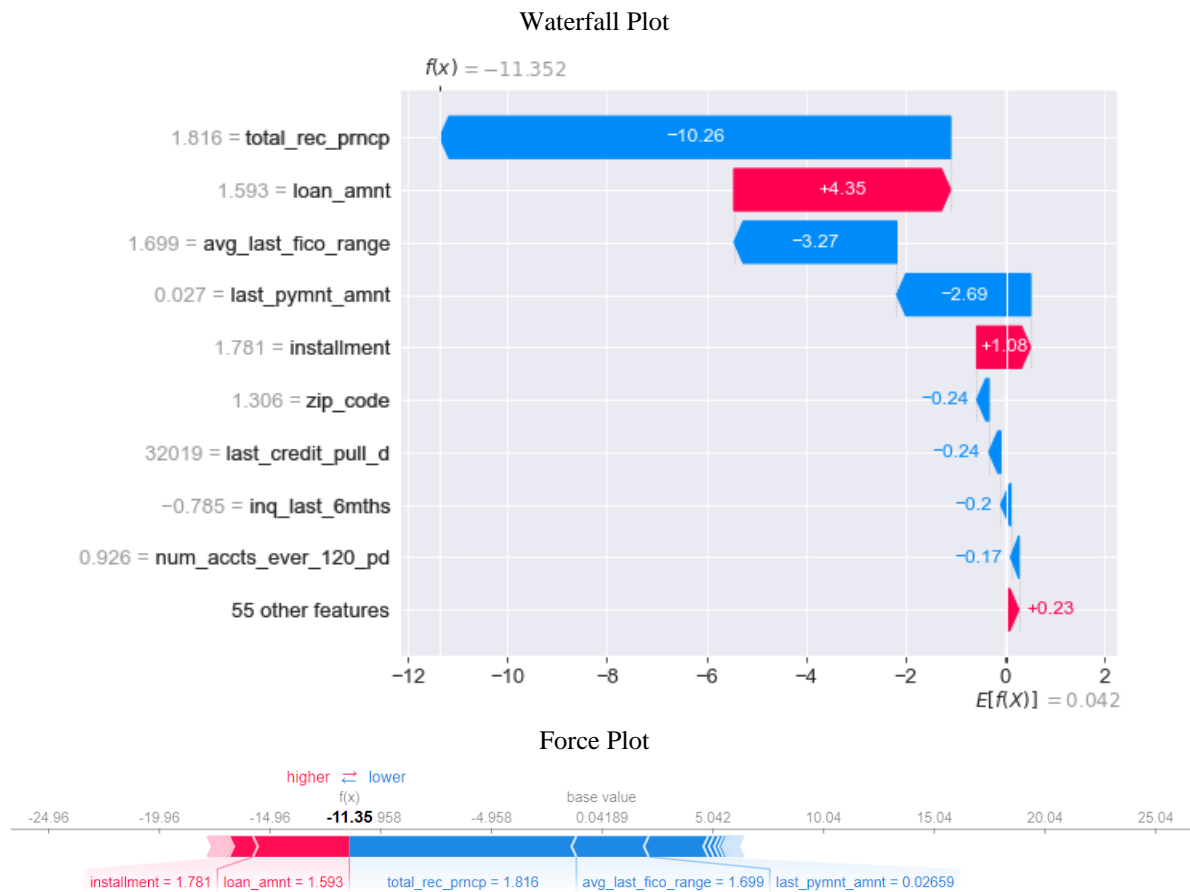


Figure 5.7.1 Explainable graph of individual test's prediction result

the prediction probability for that row is 0 and greater or equal to 0.5; then prediction probability is 1. And the  $f(x)$  value -11.32 says that the prediction result is 0, which means non-default. And for this prediction result, the total\_rec\_prncp has a negative impact with a 10.26 factor; that is true because if the loan is non-default, then the total reconciliation principal amount should be low. The higher the principal loan amount adjustment, the higher the default risk. Loan\_amnt and installment features had a positive impact. The interesting observation is that the average last fico range has a negative impact on the prediction result. The lower average last fico range helps in the higher probability of the non-default prediction for this row. And same with the last payment amount has the same negative impact with factor -2.69, which says if the last payment amount increases by one, the probability of non-default will decrease with the factor 2.69.

From figure 5.7.1, the force plot visualizes the feature contribution towards the prediction result. It represents how the value of each feature increases/decreases for the base value -11.35. The red zone represents the features (installments and loan\_amnt) that have a positive impact, and the blue zone represents the features (total\_rec\_prncp, avg\_last\_fico\_range, and last\_pymnt\_amnt) with negative impact.

Figure 5.7.2 represents the interpretation result of the first row of shap values for each feature of the joint test set. The predicted base value  $f(x)$  is -6.498 that is 0 (non-default). And for this row `last_pymnt_amnt` feature has the highest negative impact on the probability result ( $f(x)$ ) with the factor of 2.84. which indicates higher the last payment amount lowers the probability.

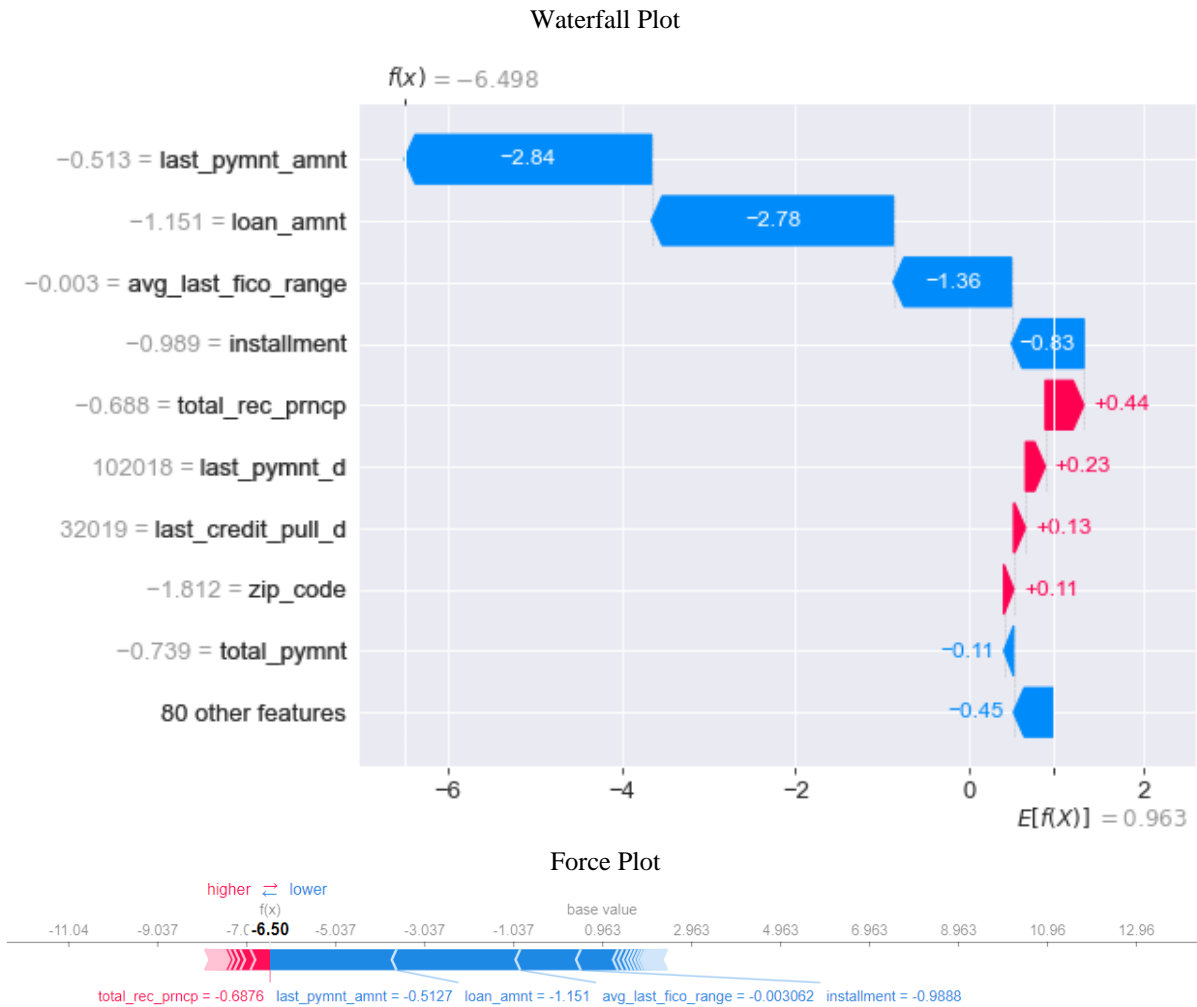


Figure 5.7.2 Explainable graph of joint test’s prediction result

Features `total_rec_prncp`, `last_pymnt_d`, `last_credit_pull_d` and `zip_code` have positive contribution to the prediction result.

Using SHAP, the interpretation of the prediction result of the black-box model is easy now. Moreover, all the plots are easy to interpret, and features contribution can be explained based on their shap values for each loan application.



## 5.8 Summary

All the prediction results of ensemble classifiers were compared before and after the resampling technique. XGBoost, LightGBM and CatBoost are sensitive to data imbalance; the misclassification costs type-I and type-II were increased for all the classifiers after the hyperparameter tuning for both datasets (individual and joint). After implying resampling techniques SMOTE and ADASYN, the misclassification costs were improved for the XGBoost with SMOTE and Stacking classifier with SMOTE of the individual dataset. And for the joint dataset, CatBoost performance was improved after applying the SMOTE.

But the oversampling method did not improve the performance of CatBoost and LightGBM in case of high data volume (individual dataset). they were still biased towards the majority class their misclassification for default as non-default increased even after using the oversampling technique. Their misclassification costs type -I error and type-II error were increased by 0.38% and 0.02% for LightGBM, and for CatBoost, they were increased by 0.18% and 0.007%, respectively. For the joint dataset, the XGBoost, LightGBM, and Stacking classifiers' performance remained the same after the resampling method because of low data volume. However, the misclassification cost for the CatBoost model was improved after the SMOTE oversampling; the misclassification metrics type-I error was improved by 0.2% and type-II error by 0.37%.

Explainability was achieved by using SHAP explainable AI to interpret the prediction results with the help of SHAP visualization plots (waterfall, force plot, and bar plot). For example, the shap bar plot demonstrated the top ten features contribution based on the average shap values. And using waterfall and force plots, the impact of the feature can be easily interpreted for any prediction result.

## **Chapter 6: Conclusion and Recommendations**

### **6.1 Introduction**

The lending club dataset is highly imbalanced, representing the real-world data scenario—and working with such data helped in finding the solution to handle the imbalanced dataset. This chapter will provide the best findings of this research study; in this chapter, the conclusion of studying the imbalanced dataset with optimal prediction results will be discussed. Achievements of the study in terms of objective and answer to the research question will also be explained in detail. Furthermore, the contribution of the research study in credit assessment will be explored. And finally, the improvement and future scope of the research study will be discussed.

### **6.2 Conclusion and discussion**

The classification of non-default and default in the credit assessment model faces two major challenges: handling the imbalance dataset and the second challenge explaining the prediction result to the borrower. The imbalance dataset is real in the financial industry, which creates a problem in the classification model because machine learning models are very sensitive to this issue. Hence, misclassification cost is higher for minority class labels (default) because the classifiers are biased towards the majority class (non-default).

So, to handle this issue, one of the objectives of this research study was to add cost-sensitive metrics to evaluate the classifier's performance, specifically to calculate misclassification cost error which helps in credit risk evaluation. So, type-I and type-II error metrics have been considered along with AUC score, precision, recall, F1-score, and G-mean. And all the performance of classifiers were evaluated based on these metrics only, lowering the misclassification cost better the model.

So, after identifying the metrics for evaluation, another objective of this study is to analyze the performance of the ensemble classifiers. And this study included three boosting classifiers (XGBoost, LightGBM, and CatBoost) and the Stacking ensemble model using three boosting classifiers (XGBoost, LightGBM and CatBoost) as the base and logistic regression as meta classifier. As the lending club dataset was split into two parts based on the application type (individual and joint dataset) for better credit assessment, so after the model implementation on both of the datasets, XGBoost were performing better than LightGBM and CatBoost for the individual dataset, which had high volume data before the parameter tuning. As a result, XGBoost achieved the lowest type-I error, 0.281%, and type-II error, 0.006%.

Misclassification cost was the highest for LightGBM, so LightGBM was more sensitive to class imbalance. After the parameter tuning, the misclassification costs were increased for all the classifiers. So, parameter tuning may improve the biasing issue caused due to data imbalance hence more default applicant (minority labels) was predicted as non-default applicant (majority class labels), which increased the type-I error. Even after combining all three boosting classifiers using the Stacking ensemble model, the prediction result was the same as XGBoost. So, in the case of high volume and highly imbalanced data XGBoost and Stacking model gave the best result.

But when the data volume was low (the joint dataset), the type-I error was lowest for LightGBM and Stacking classifier, both achieved the same result for type-I error 0.465%, and type-II error

was also low 0.017%. But the misclassification also increased hereafter the parameter tuning for all the classifiers. So, hyperparameter tuning did not help improve the misclassification cost. However, the Stacking ensemble model achieved the best performance for both of the datasets in case of a data imbalance problem.

All the classifiers worked on imbalanced data where the percentage of default class labels was 20% only. So, to manage this imbalance ratio, another objective of this research study is to find an optimal resampling technique. Two oversampling techniques, SMOTE and ADASYN, were used to create the synthetic sample for minority class to balance the class labels for the individual and joint dataset. Afterward, all classifiers were implemented on the new resampled dataset. The performance of all the classifiers was improved for SMOTE resample data, and XGBoost and stacking model with SMOTE outperformed among all the classifiers for the individual dataset. They both gave the same result. They both have achieved 99.86% in the AUC score, 99.93% in precision, 99.92% in the recall, 99.96% in F1-score, and 99.86% in G-mean. Their type-I error was the lowest, 0.279%, and type-II error was the second-lowest that is 0.008%.

And for the small volume dataset (the joint dataset), CatBoost with SMOTE outperformed with the lowest type-I error of 0.361% and type-II error of 0.052%. So, the optimal resampling technique that improves the model's performance is SMOTE, which works well for both datasets (low and high volume). In addition, the misclassification cost was reduced after introducing the SMOTE oversampling method.

Finally, the imbalanced problem was handled by SMOTE and XGBoost, and the Stacking classifier achieved the highest model's performance with SMOTE for individual datasets and CatBoost with SMOTE for the joint dataset. So, to handle the second challenge usually faced in developing a credit assessment model is to explain the prediction result to the borrower, which is not an easy way to do because machine learning model is black-box in nature, hence to achieve this last objective of the study an explainable AI SHAP technique was used on prediction result, and after applying shap explainer on the prediction result of the classifier contribution of the features were plotted using shap visualization plot waterfall, force and bar plot. Based on the average shap value, the top five features that impacted the prediction result were total\_rec\_prncp, loan\_amnt, avg\_last\_fico\_range, last\_pymnt\_amnt, and installment.

Based on the achieved objective of this research study, the answer to the research questions are:

- Yes, the oversampling technique (SMOTE) improves the model's performance (XGBoost, CatBoost, and Stacking classifier), and the highest performance was achieved for XGBoost and Stacking model.
- And with the help of explainable AI (SHAP) integration with the model framework, the prediction result is interpretable and can be easily explained to the borrower. So, the answer to the second research question is yes, the result of the black-box model can be interpretable.

### **6.3 Contribution of the study**

The approach proposed in this study XGBoost with SMOTE and Stacking with SMOTE achieved better performance than previous studies. And integration of the explainable AI makes the whole model framework a complete package that provides better prediction results of credit risk and borrower will get to know why his loan was not approved. As a result, this study will help the lending institution to its business by gaining the borrower's trust, and the chance of misclassification is lower than the other model.

### **6.4 Recommendation**

The stacking model combines the multiple base classifiers and achieves a better performance result. So, the initial approach of this research study was to utilize the benefits of all the three boosting-classifier by stacking them and achieving better performance than a single classifier. And XGBoost and Stacking gave the best but same result. So, in the future, the Stacking of other classifier like neural network will be interesting to analyze.

The approach of this study was not to lose any information, so only oversampling techniques were used and compared in the future other resampling technique like random undersampling is good to be explored.

## References

- Anon (2022) eCFR:: 12 CFR 1002.9 -- Notifications. [online] Available at: <https://www.ecfr.gov/current/title-12/chapter-X/part-1002/section-1002.9> [Accessed 5 Jan. 2022].
- Anon (2022) EUR-Lex - 32016R0679 - EN - EUR-Lex. [online] Available at: <https://eur-lex.europa.eu/eli/reg/2016/679/oj> [Accessed 5 Jan. 2022].
- Barua, S., Gavandi, D., Sangle, P., Shinde, L. and Ramteke, J., (2021) Swindle: Predicting the Probability of Loan Defaults using CatBoost Algorithm. In: *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*. Institute of Electrical and Electronics Engineers Inc., pp.1710–1715.
- Biecek, P., Chlebus, M., Gajda, J., Gosiewska, A., Kozak, A., Ogonowski, D., Sztachelski, J. and Wojewnik, P., (2021) *Enabling Machine Learning Algorithms for Credit Scoring - Explanatory Artificial Intelligence (XAI) methods for clear understanding complex predictive models*.
- Bussmann, N., Giudici, P., Marinelli, D. and Papenbrock, J., (2021) Explainable Machine Learning in Credit Risk Management. *Computational Economics*, 571, pp.203–216.
- Chakravarty, S., Demirhan, H. and Baser, F., (2020) Fuzzy regression functions with a noise cluster and the impact of outliers on mainstream machine learning methods in the regression setting. *Applied Soft Computing Journal*, 96.
- Chawla, N. v, Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., (2002) SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, .
- Chen, T. and Guestrin, C., (2016) XGBoost: A scalable tree boosting system. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, pp.785–794.
- Chengeta, K. and Mabika, E.R., (2021a) Peer to Peer Social Lending Default Prediction with Convolutional Neural Networks. In: *icABCD 2021 - 4th International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems, Proceedings*. Institute of Electrical and Electronics Engineers Inc.
- Chengeta, K. and Mabika, E.R., (2021b) Peer To Peer Social Lending Default Prediction With Convolutional Neural Networks. In: *2021 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*. IEEE.
- Chi, J., Zeng, G., Zhong, Q., Liang, T., Feng, J., Xiang, A. and Tang, J., (2020) Learning to undersampling for class imbalanced credit risk forecasting. In: *Proceedings - IEEE International Conference on Data Mining, ICDM*. Institute of Electrical and Electronics Engineers Inc., pp.72–81.
- Dastile, X. and Celik, T., (2021) Making Deep Learning-Based Predictions for Credit Scoring Explainable. *IEEE Access*, 9, pp.50426–50440.
- Dzik-Walczak, A. and Heba, M., (2021) An implementation of ensemble methods, logistic regression, and neural network for default prediction in peer-to-peer lending. *Zbornik Radova Ekonomskog Fakultet au Rijeci*, 391, pp.163–197.

Egan, C., (2021) *Improving Credit Default Prediction Using Explainable AI MSc Research Project Data Analytics Improving Credit Default Prediction Using Explainable AI*. [online] Available at: <http://norma.ncirl.ie/5146/> [Accessed 28 Dec. 2021].

Faris, H., Abukhurma, R., Almanaseer, W., Saadeh, M., Mora, A.M., Castillo, P.A. and Aljarah, I., (2020) Improving financial bankruptcy prediction in a highly imbalanced class distribution using oversampling and ensemble learning: a case from the Spanish market. *Progress in Artificial Intelligence*, 91, pp.31–53.

Feng, X., Xiao, Z., Zhong, B., Qiu, J. and Dong, Y., (2018) Dynamic ensemble classification for credit scoring using soft probability. *Applied Soft Computing Journal*, 65, pp.139–151.

García, V., Marqués, A.I. and Sánchez, J.S., (2019) Exploring the synergetic effects of sample types on the performance of ensembles for credit risk and corporate bankruptcy prediction. *Information Fusion*, 47, pp.88–101.

Hadji Misheva, B., Hirs, A., Osterrieder, J., Kulkarni, O. and Fung Lin, S., (2021) *EXPLAINABLE AI IN CREDIT RISK MANAGEMENT A PREPRINT*. [online] Available at: <https://ssrn.com/abstract=3795322>.

Hamori, S., Kawai, M., Kume, T., Murakami, Y. and Watanabe, C., (2018) Ensemble Learning or Deep Learning? Application to Default Risk Analysis. *Journal of Risk and Financial Management*, 111, p.12.

He, H., Zhang, W. and Zhang, S., (2018) A novel ensemble method for credit scoring: Adaption of different imbalance ratios. *Expert Systems with Applications*, 98, pp.105–117.

Jin, Y., Zhang, W., Wu, X., Liu, Y. and Hu, Z., (2021) A Novel Multi-Stage Ensemble Model with a Hybrid Genetic Algorithm for Credit Scoring on Imbalanced Data. *IEEE Access*, 9, pp.143593–143607.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.-Y., (2017) *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*. [online] Available at: <https://github.com/Microsoft/LightGBM>.

Kun, Z., Weibing, F. and Jianlin, W., (2020a) Default Identification of P2P Lending Based on Stacking Ensemble Learning. In: *Proceedings - 2020 2nd International Conference on Economic Management and Model Engineering, ICEMME 2020*. Institute of Electrical and Electronics Engineers Inc., pp.992–1006.

Kun, Z., Weibing, F. and Jianlin, W., (2020b) Default Identification of P2P Lending Based on Stacking Ensemble Learning. In: *2020 2nd International Conference on Economic Management and Model Engineering (ICEMME)*. IEEE.

Laborda, J. and Ryoo, S., (2021) Feature selection in a credit scoring model. *Mathematics*, 97.

Lending club, (2021) *All Lending Club loan data | Kaggle*. [online] Available at: [https://www.kaggle.com/wordsforthewise/lending-club?select=accepted\\_2007\\_to\\_2018Q4.csv.gz](https://www.kaggle.com/wordsforthewise/lending-club?select=accepted_2007_to_2018Q4.csv.gz) [Accessed 7 Nov. 2021].

Li, W., Ding, S., Chen, Y. and Yang, S., (2018) Heterogeneous ensemble for default prediction of peer-to-peer lending in China. *IEEE Access*, 6, pp.54396–54406.

- Li, Z., Zhang, J., Yao, X. and Kou, G., (2021) How to identify early defaults in online lending: A cost-sensitive multi-layer learning framework. *Knowledge-Based Systems*, 221.
- Lundberg, S.M., Allen, P.G. and Lee, S.-I., (2017) *A Unified Approach to Interpreting Model Predictions*. [online] Available at: <https://github.com/slundberg/shap>.
- Ma, X., Sha, J., Wang, D., Yu, Y., Yang, Q. and Niu, X., (2018) Study on a prediction of P2P network loan default based on the machine learning LightGBM and XGboost algorithms according to different high dimensional data cleaning. *Electronic Commerce Research and Applications*, 31, pp.24–39.
- Malik, R., (2021) An Impact of Loan Defaults and Impact on Profitability of Bank. *International Journal for Research in Engineering Application & Management (IJREAM)*, 0611, pp.2454–9150.
- Moscato, V., Picariello, A. and Sperlí, G., (2021) A benchmark of machine learning approaches for credit score prediction. *Expert Systems with Applications*, 165.
- Namvar, A., Siami, M., Rabhi, F. and Naderpour, M., (2018) *Credit risk prediction in an imbalanced social lending environment*.
- Niu, K., Zhang, Z., Liu, Y. and Li, R., (2020) Resampling ensemble model based on data distribution for imbalanced credit risk evaluation in P2P lending. *Information Sciences*, 536, pp.120–134.
- Nyitrai, T. and Virág, M., (2019) The effects of handling outliers on the performance of bankruptcy prediction models. *Socio-Economic Planning Sciences*, 67, pp.34–42.
- Oreški, S. and Oreški, G., (2018) Cost-sensitive learning from imbalanced datasets for retail credit risk assessment. *TEM Journal*, 71, pp.59–73.
- Pes, B. and Lai, G., (2021) Cost-sensitive learning strategies for high-dimensional and imbalanced data: a comparative study. *PeerJ Computer Science*, [online] 7, p.e832. Available at: <https://peerj.com/articles/cs-832>.
- Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V. and Gulin, A., (2017) CatBoost: unbiased boosting with categorical features. [online] Available at: <http://arxiv.org/abs/1706.09516>.
- Ri, J.H. and Kim, H., (2020) G-mean based extreme learning machine for imbalance learning. *Digital Signal Processing: A Review Journal*, 98.
- Ribeiro, M.T., Singh, S. and Guestrin, C., (2016) Model-Agnostic Interpretability of Machine Learning. [online] Available at: <http://arxiv.org/abs/1606.05386>.
- Roberto Lopez, (2021) *How to benchmark the performance of machine learning platforms: data capacity, training speed, inference speed and model precision | Neural Designer*. [online] Available at: <https://www.neuraldesigner.com/blog/how-to-benchmark-the-performance-of-machine-learning-platforms> [Accessed 7 Nov. 2021].

- Saidi Meryem, Habib Daho Mostafa El, Settouti Nesma and Amine Bechar Mohammed El, (2018) Comparison of ensemble cost sensitive algorithms application to credit scoring prediction. *International Conference on Advanced Aspects of Software Engineering*, 2326.
- Sandica, A.M. and Fratila, A., (2021) Implications of macroeconomic conditions on Romanian portfolio credit risk. A cost-sensitive ensemble learning methods comparison. *Economic Research-Ekonomska Istrazivanja* .
- Shen, F., Zhao, X., Kou, G. and Alsaadi, F.E., (2021) A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique. *Applied Soft Computing*, 98.
- Shen, F., Zhao, X., Li, Z., Li, K. and Meng, Z., (2019) A novel ensemble classification model based on neural networks and a classifier optimisation technique for imbalanced credit risk evaluation. *Physica A: Statistical Mechanics and its Applications*, 526.
- Song, Y. and Peng, Y., (2019) A MCDM-Based Evaluation Approach for Imbalanced Classification Methods in Financial Risk Prediction. *IEEE Access*, 7, pp.84897–84906.
- Sterner, P., Goretzko, D. and Pargent, F., (2021) *COST-SENSITIVE LEARNING 1 Everything has its Price: Foundations of Cost-Sensitive Learning and its Application in Psychology*. [online] Available at: <https://osf.io/cvks7/>.
- Tripathi, D., Edla, D.R., Bablani, A., Shukla, A.K. and Reddy, B.R., (2021) *Experimental analysis of machine learning methods for credit score classification*. *Progress in Artificial Intelligence*, .
- Wang, D. and Zhang, Z., (2020) Enterprise Credit Risk Assessment Using Feature Selection Approach and Ensemble Learning Technique. In: *Proceedings - 2020 16th International Conference on Computational Intelligence and Security, CIS 2020*. Institute of Electrical and Electronics Engineers Inc., pp.228–233.
- Wang, H., Kou, G. and Peng, Y., (2021) Multi-class misclassification cost matrix for credit ratings in peer-to-peer lending. *Journal of the Operational Research Society*, 724, pp.923–934.
- Wong, M.L., Seng, K. and Wong, P.K., (2020) Cost-sensitive ensemble of stacked denoising autoencoders for class imbalance problems in business domain. *Expert Systems with Applications*, 141.
- Wu, D., Guo, P. and Wang, P., (2020a) Malware Detection based on Cascading XGBoost and Cost Sensitive. In: *Proceedings - 2020 International Conference on Computer Communication and Network Security, CCNS 2020*. Institute of Electrical and Electronics Engineers Inc., pp.201–205.
- Wu, D., Guo, P. and Wang, P., (2020b) Malware Detection based on Cascading XGBoost and Cost Sensitive. In: *Proceedings - 2020 International Conference on Computer Communication and Network Security, CCNS 2020*. Institute of Electrical and Electronics Engineers Inc., pp.201–205.
- Wu, S., Gao, X. and Zhou, W., (2022) COSLE: Cost sensitive loan evaluation for P2P lending. *Information Sciences*, 586, pp.74–98.



Xia, Y., Zhao, J., He, L., Li, Y. and Niu, M., (2020) A novel tree-based dynamic heterogeneous ensemble method for credit scoring. *Expert Systems with Applications*, 159.

Yotsawat, W., Wattuya, P. and Srivihok, A., (2021a) A Novel Method for Credit Scoring Based on Cost-Sensitive Neural Network Ensemble. *IEEE Access*, 9, pp.78521–78537.

Yotsawat, W., Wattuya, P. and Srivihok, A., (2021b) A Novel Method for Credit Scoring Based on Cost-Sensitive Neural Network Ensemble. *IEEE Access*, 9, pp.78521–78537.

Zhang, T. and Li, J., (2021) Credit Risk Control Algorithm Based on Stacking Ensemble Learning. In: *Proceedings of 2021 IEEE International Conference on Power Electronics, Computer Applications, ICPECA 2021*. Institute of Electrical and Electronics Engineers Inc., pp.668–670.

Zhang, W., He, H. and Zhang, S., (2019) A novel multi-stage hybrid model with enhanced multi-population niche genetic algorithm: An application in credit scoring. *Expert Systems with Applications*, 121, pp.221–232.

Zhang, W., Yang, D. and Zhang, S., (2021) A new hybrid ensemble model with voting-based outlier detection and balanced sampling for credit scoring. *Expert Systems with Applications*, 174.

## **APPENDIX A: RESEARCH PROPOSAL**

PREDICTING THE LOAN DEFAULT RISK FOR IMBALANCED DATA USING THE FASTEST AND MORE  
ACCURATE MACHINE LEARNING PIPELINE

BHAWNA GUPTA

Research Proposal

NOVEMBER 2021

## **Abstract**

Loan default is the biggest threat in the financial industry. It has a significant impact on the economy as well. So robust credit assessment is a real challenge. Financial data is very imbalanced and high in volume, and handling them within less computation time, and good accuracy result is the objective of this research. So this research proposes an embedded feature selection method to reduce the high dimensional dataset after the proper data cleaning. Afterward, the benefit of random oversampling and undersampling will be used to balance the dataset to avoid the overfitting of minority class due to oversampling and loss of important information of majority class due to undersampling. Since Boosting classification gives excellent accuracy results and works well with the imbalanced dataset, this research will use the cascading ensemble classification model to fit the training dataset, using XGBoost, LightGBM, and CatBoost algorithm. The classification report and ROC curve will evaluate the model. However, additional credit measures like Default Detection Rate, Misclassification Cost, and G-mean also evaluate the model against domain-specific metrics. The overall approach will apply to the highly imbalanced dataset Lending Club dataset.

## LIST OF TABLES

Table 7.1.1 Data Description .....	50
Table 8.1 Hardware Requirements .....	52
Table 8.2 Software Requirements .....	53

## LIST OF FIGURES

Figure 7.1 Workflow of methodology .....	49
Figure 7.5.1 Imbalanced data of loan status .....	50
Figure 9.1 Gantt Chart .....	52

## LIST OF ABBREVIATIONS

ANN.....	Artificial Neural Network
AOD.....	Autoencoder Outlier Detection
DDR.....	Default Detection Rate
EDA.....	Exploratory Data Analysis
GM.....	G-Mean
MC.....	Misclassification Cost
ROUS.....	Random oversampling and under-sampling
ROC Curve.....	Receiver operating characteristic curve
SMOTE.....	Synthetic Minority Over-sampling Technique
SVM.....	Support Vector Machine

## 1. Background

Every financial firm all over the world does their business by sanctioning the loan to their customer. And every sanctioned loan has risks associated with it. If some loan gets defaults, it has a massive impact on the firm's profitability. 2008 Lehman brothers bankruptcy is an excellent example of how loan default can cause insolvency of the firm (Malik, 2021). So every loan proposal needs proper assessment, which can take time. The firm may lose its potential customer that also a huge loss for the firm. Here, the firm can use the benefit of machine learning, which helps develop a fast and accurate credit assessment system that improves the firm's profitability.

The past research proposes various credit assessments, and all the studies focused on improving accuracy. Some researchers also proposed some additional credit measures like Default Detection Rate (DDR), G-Mean (GM), and Misclassification Cost (MC) (Yotsawat et al., 2021b) to improve the model performance for more finance-oriented because every domain has its performance metrics. The above metrics are also going to be part of this study.

There are various key performance measures for machine learning model, but below four can define most of the model characteristics by (Roberto Lopez, 2021):

- **Data Capacity Test** measures how big a dataset a model can process without any crash.
- **Training Speed Test** measures the processing time of a number of samples per second while fitting the train data set by a model.
- **Model Precision Test** evaluates the model's accuracy using the test dataset.
- **Inference Speed Test** measures the model's response time in real-time.

Training speed test and model precision test will be explored in this research study with the hypothesis of whether the machine learning pipeline with the embedded feature selection, random oversampling and undersampling, and boosting ensemble model improve the training speed and model precision significantly.

## 2. Related work

Much work has already been done in the prediction of loan default. Most algorithms or combinations of algorithms are already performed to make the prediction more accurate. However, the finance-related dataset has high data volume and is imbalanced. So decreasing the computation time and balancing the data is an important factor as well. Some research study has been discussed below to highlight the important points which are going to be considered in this research

In the paper (Kun et al., 2020b), Zhang Kun, Feng Weibing, Wu Jianlin proposed a stacking ensemble classification model using ANN (Artificial Neural Network), Random Forest, AdaBoost & XGBoost, and they have achieved good accuracy. In addition, the SMOTE algorithm has been used for oversampling the imbalanced data. However, there is a need to improve the learning efficiency while reducing the computation time.

Kennedy Chengeta & Elizabeth Rutendo Mabika (Chengeta and Mabika, 2021b) built the new class of classifiers for XGBoost, Deep Learning, CatBoost, and LightGBM compared the performance in terms of computation time and accuracy. The result shows that CNN deep learning shows better accuracy than others, but the boosting model XGBoost and LightGBM results in better processing time and parallel processing.

Wirot Yotsawat, Pakaket Wattuya & Anongnart Srivihok (Yotsawat et al., 2021b) proposed a cost-sensitive Neural Network ensemble model for balanced and imbalanced datasets. A comparative study with other models shows the superiority of the proposed model in terms of accuracy, Default Detection Rate (DDR), G-Mean (GM), and Misclassification Cost (MC). In addition, they have used Autoencoder Outlier Detection (AOD) to remove the noisy and outlier data to improve the computation speed.

Diwakar Tripathi, Damodar Reddy Edla, Annushree Bablani, Alok Kumar Shukla & B. Ramachandra Reddy (Tripathi et al., 2021) suggested that we work on feature selection that will improve the performance as well. They have proposed comparative analysis of state of the art approaches as classification, rule-based classifications, ensembles classification, and hybrid model. They have concluded that ensemble classification approaches considering the proper feature selection method may perform better accuracy.

Di Wu, Peiqi Guo, Peng Wang (Wu et al., 2020b) discussed in their project to detect malware based on a 3-tier cascading XGBoost and cost-sensitive method to manage the unbalanced dataset. They have used the random forest to reduce the dimensions of features and then put data into a model based on a 3-tier cascade XGBoost. The cascading model technique helps to filter the imbalanced data and improve the overall performance of the model.

Haomin Wang, Gang Kou & Yi Peng (Wang et al., 2021) suggested a multiclass cost matrix that measures misclassification costs of P2P lending. However, they have used Cost-sensitive

classification, which results in lowering down the accuracy of the model. In conclusion, it is pretty challenging to achieve the highest accuracy and the lowest total cost simultaneously. Juan Laborda and Seyong Ryoo (Laborda and Ryoo, 2021) proposed three different feature selection methods one filter-based (Chi-squared test and correlation coefficients) and second wrapper based (forward stepwise and backward stepwise) and compare the performance of multiple classification algorithms (Logistic Regression, SVM, K-nearest neighbors and random forest). And research concluded that Forward stepwise gives the best result for the mentioned classification model. Furthermore, Embedded-based method selection will be the subsequent study to be explored to measure the performance.

All the above discussed research study shows the excellent result and proves their points with evidence. Various approaches have been used to improve the model performance, and the overall boosting modeling technique gives impressive results. For the data balancing, oversampling technique (SMOTE) is prevalent, but this may increase the case of overfitting and for high data volume the processing time may also increase. Some research proposed that if a good feature selection technique is used, this may also decrease the model's training speed and improve the accuracy.

So this research study will explore which feature selection method, data balancing technique, and ensemble boosting model results in good training speed and model precision.

### **3. Research Question**

The focus of this study is to reduce the model computation time and improve the model performance. Therefore, the following research question is formulated, which are going to be dealt with in this study:

- What are the factors that will help to improve the performance of training speed and model cost matrix?

### **4. Aim and Objectives**

This research aims to find the machine learning model in which computation time is the lowest. The goal is to find the best steps like feature selection and data balancing technique, which helps improve the performance of the whole modeling.

Following research objectives are formulated based on the research question suggested above:

- To suggest the best feature selection technique for faster computation time.
- To find a suitable balancing technique that will not lead to overfitting.
- To analyze various ensemble modeling technique which helps to improve the performance of training speed and accuracy.

- To evaluate the model's performance based on other cost matrices of the machine learning model.

## **5. Significance of the Study**

In the related work section, various comparative analysis of algorithms like XGBoost, Deep learning, LightGBM, AdaBoost, and Artificial Neural network has already been done. They all worked to improve the model accuracy, and some of the research papers also used some credit measures like G-mean and default detection rate (Yotsawat et al., 2021). Moreover, a data balancing technique like SMOTE was already used (Kun et al., 2020b). Finally, everything was checking the accuracy of the machine learning pipeline with some computation cost factor (Wang et al., 2021).

In this research study benefits of the feature selection method with proper data balancing technique will be considered. The benefit of cascading ensemble learning based on boosting technique is also going to be covered here. The machine learning pipeline performance is also evaluated by credit measures metrics like G-mean, Default Detection Rate, and Misclassification cost (Yotsawat et al., 2021b).

The overall focus of this research study is on how to improve the model performance on various measures using less computational time for default risk prediction.

The implication of this research will help any financial firm get the best prediction accuracy of default risk for any given loan proposal with faster computation time. It does not matter how huge their data volume is and how imbalanced they are.

## **6. Scope of the Study**

This research study will explore the cascading ensemble learning based on boosting algorithm to check the model accuracy will perform better or not. A combination of oversampling and undersampling will be used to check the improvement in model training speed and accuracy. Embedded-based feature selection is also going to be covered here to achieve the objective.

Deep learning and neural network algorithm is not part of this research study. However, suppose the feature selection and data balancing helps improve the training speed and accuracy of the model. In that case, deep learning and neural network-based algorithms will be explored for future work.



## 7. Research Methodology

The purpose of this research study is to predict the default risk percentage associated with any given loan proposal within less computation time and with the best prediction accuracy. So the workflow of methodology for the machine learning pipeline for this research is below.

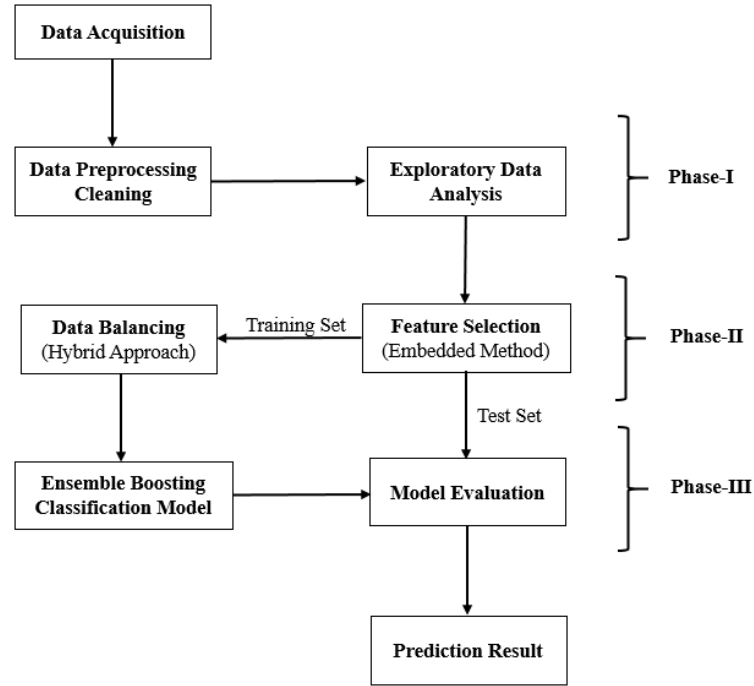


Figure 7.1 workflow of methodology

### 7.1 Data Acquisition:

The American peer-to-peer lending company named Lending club's data set is being used for this research study collected from the kaggle(Lending club, 2021). It has two csv files for accepted and rejected loans, which hold information from 2007 to 2018. The rejected loan data set only holds precise information about the rejected loan proposal, but the accepted file holds all the detailed information about the lending. There is 151 column in accepted csv, and the **loan\_status** field shows a particular loan is a default or not, is the target variable. Description of some of the important features is as follow:

#### Data Description

Column Name	Description
loan_amnt	The loan amount sanctioned to the borrower
term	A number of months for loan repayment (36 months or 60 months).
int_rate	The applicable interest rate on loan.
installment	The monthly payment amount for the sanctioned loan.

grade	Loan grades are assigned by the lending club based on the borrower's FICO Score.
emp_title	Employment title of the borrower
emp_length	Length of service of the borrower.
home_ownership	Status of home (rented/owned/mortgaged) of the borrower
annual_inc	Annual income of the borrower.
verification_status	Indicates if Lending Club verified income.
issue_d	Issue date of the loan.
loan_status	Status of the loan (fully paid/default/charged off)
purpose	Purpose of the loan.
zip_code	The first three digits of the zip code borrower's address.
dti	It is a ratio of the total debt of the borrower to his total income
earliest_cr_line	Earliest date of open credit of the borrower.
fico_range_low	Lower FICO score boundary of the borrower.
open_acc	How many open credits the borrower has.
revol_util	The usage rate of the revolving credit by the borrower
total_acc	A total number of credit account of the borrower.
application_type	Type of loan application (individual/ joint)
annual_inc_joint	Total annual income of the applicants (borrower & co-borrower)

Table 7.1.1

## 7.2 Data Pre-processing

This step transforms the structure of data to improve the performance of the model. Preprocessing involves pre-analysis of raw data by handling the null values, outliers, restructuring the column data type for deriving meaningful information, and binning column data.

- For null value handling 30-40% criteria will be used. If any column has more than 40% null values, it is better to drop it because any null value imputation for such a column may skew data that may not be useful.
- For handling outlier capping method will be used, some column like annual\_inc or loan\_amount has an extreme value which may affect the model performance, so for better model accuracy and speeding the model training outlier should be appropriately handled.

## 7.3 Exploratory Data Analysis

After the preprocessing of the data set, EDA will analyze the important characteristic of the dataset. This step will help in finding the relationship between the target variable, loan\_status, and other features. For example, visualizing the data distribution of features like annual\_inc, home\_ownership, fico\_range, earliest\_cr\_line concerning the target variable will give good insight.

## 7.4 Feature Selection

The purpose of this research is to minimize the training speed and improve accuracy. As concluded from the related work studies, feature selection plays a significant role in this. The lending club data set has 151 features, and if all the features are imputed to the model, it will increase the training speed, which may cause overfitting that will badly affect the model performance. So feature selection step will help to fix all these issues. So next question is which type of feature selection should be used. There are three categories:

1. **Filter-Based**, this method is based on the correlation between variables. It is faster in computing but may result in poor accuracy.
2. **Wrapper Based**, this method selects all possible combinations of a feature subset that results in the best accuracy for the machine learning model. Therefore, computation time is very high for this method.
3. **Embedded**, this method is a combination of the above two. It has a faster computation speed like filter-based and the accuracy result is like wrapper. So this method will be used in this research study. There are various algorithms like Lasso, Ridge, Decision tree, Random forest, etc. From the related work studies, feature selection based on random forest performs better, so Random forest will be used in this research study.

## 7.5 Data Balancing

If the dataset is balanced, then the accuracy result is not biased. But if the dataset is imbalanced, then accuracy for majority class results is good but not for the minority class. And the lending club dataset is highly imbalanced. Figure.2 represents the loan\_status values, where the Default case is only 0.001769 % while the majority class has 47.629% of data.

```
1  lending_club.loan_status.value_counts(normalize=True)*100
Fully Paid          47.629771
Current            38.852100
Charged Off        11.879630
Late (31-120 days)  0.949587
In Grace Period     0.373164
Late (16-30 days)   0.192377
Does not meet the credit policy. Status:Fully Paid  0.087939
Does not meet the credit policy. Status:Charged Off 0.033663
Default             0.001769
Name: loan_status, dtype: float64
```

Figure 7.5.1 imbalanced data of loan\_status

So to improve model performance, a data balancing technique should be used. There are two ways to resample the data:

1. **Under-Sampling**, here some data points from majority class are got deleted to balance with minority class. In this case, some vital information may also get deleted.
2. **Over-Sampling**, here duplicate sample data points created for minority class to balance with majority class. But because of the duplicity of the data points, minority class may lead to overfitting.

Here data is highly imbalanced. If oversampling is used on minority class with only 0.0017% observation to match with majority class that is 47.62%, it will cause the overfitting. Moreover, if under-sampling is used, then the majority class will lose lots of observation points. So combining these two will improve the overall performance. Hence random oversampling and under-sampling (ROUS) methods will be used for data balancing in this research.

## 7.6 Machine learning Model

This research aims to reduce the training speed, and from the related work research boosting algorithms like XGBoost, LightGBM and CatBoost perform very well in computation time for fitting the train data set. In the previous work, only comparative analysis has been done on these algorithms. This research study will apply cascading ensemble learning using these three algorithms on the lending club dataset.

## 7.7 Model Evaluation

Classification report with ROC curve is widely used for classification model evaluation. In addition, some other metrics like Default Detection Rate (DDR), G-Mean (GM), and Misclassification Cost (MC) (Yotsawat et al., 2021b) will be used to showcase other credit classification performance measures.

If the model evaluation does not give a good accuracy result, all the steps will be reiterated until better accuracy is achieved.

## 8. Requirements Resources

Reducing the computation time is the objective of this research study, but if the system resources are slow, no machine model runs fast. So the essential hardware required for this research is as follows:

- **Hardware Requirement**

Feature	Specification
Operating System	Windows 10/ Mac X or higher
RAM	8 GB or higher

Processor	Intel Core i5 or higher
GPU	NVIDIA GTX equivalent or higher

Table 8.1

Some of the software requirements are mentioned below, which help set up the machine learning modeling environment.

- **Software Requirement**

<b>System Requirement</b>	Anaconda Jupyter Notebook/Pycharm
	Google Colab
	Python 3.7 or higher environment set up
<b>Python Package Requirement</b>	pandas 1.3, numpy 1.21 or higher
	matplotlib 3.4, seaborn 0.11.1 or higher
	scikit-learn 0.24.2 ,imbalanced-learn 0.8.0 or higher

Table 8.2

Above mentioned resource requirement is basic. It may get changed as per additional modeling requirements.

## 9. Research Plan

The below Gantt chart shows the overall research plan from topic selection to final thesis submission. The plan period used in the below chart is in the week.

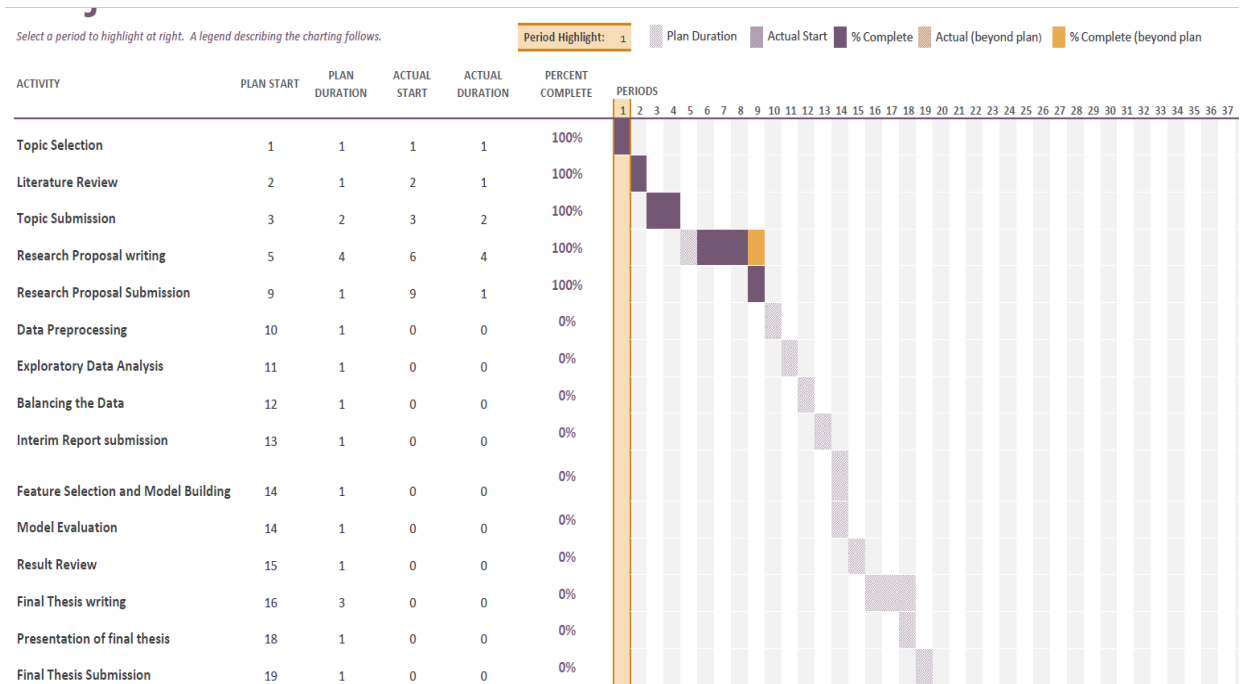


Figure 9.1 Gantt Chart

## References

- Anon (2022) eCFR:: 12 CFR 1002.9 -- Notifications. [online] Available at: <https://www.ecfr.gov/current/title-12/chapter-X/part-1002/section-1002.9> [Accessed 5 Jan. 2022].
- Anon (2022) EUR-Lex - 32016R0679 - EN - EUR-Lex. [online] Available at: <https://eur-lex.europa.eu/eli/reg/2016/679/oj> [Accessed 5 Jan. 2022].
- Barua, S., Gavandi, D., Sangle, P., Shinde, L. and Ramteke, J., (2021) Swindle: Predicting the Probability of Loan Defaults using CatBoost Algorithm. In: *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*. Institute of Electrical and Electronics Engineers Inc., pp.1710–1715.
- Biecek, P., Chlebus, M., Gajda, J., Gosiewska, A., Kozak, A., Ogonowski, D., Sztachelski, J. and Wojewnik, P., (2021) *Enabling Machine Learning Algorithms for Credit Scoring - Explanatory Artificial Intelligence (XAI) methods for clear understanding complex predictive models*.
- Busmann, N., Giudici, P., Marinelli, D. and Papenbrock, J., (2021) Explainable Machine Learning in Credit Risk Management. *Computational Economics*, 571, pp.203–216.
- Chakravarty, S., Demirhan, H. and Baser, F., (2020) Fuzzy regression functions with a noise cluster and the impact of outliers on mainstream machine learning methods in the regression setting. *Applied Soft Computing Journal*, 96.
- Chawla, N. v, Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., (2002) SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, .
- Chen, T. and Guestrin, C., (2016) XGBoost: A scalable tree boosting system. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, pp.785–794.
- Chengeta, K. and Mabika, E.R., (2021a) Peer to Peer Social Lending Default Prediction with Convolutional Neural Networks. In: *icABCD 2021 - 4th International Conference on Artificial*

*Intelligence, Big Data, Computing and Data Communication Systems, Proceedings*. Institute of Electrical and Electronics Engineers Inc.

Chengeta, K. and Mabika, E.R., (2021b) Peer To Peer Social Lending Default Prediction With Convolutional Neural Networks. In: *2021 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*. IEEE.

Chi, J., Zeng, G., Zhong, Q., Liang, T., Feng, J., Xiang, A. and Tang, J., (2020) Learning to undersampling for class imbalanced credit risk forecasting. In: *Proceedings - IEEE International Conference on Data Mining, ICDM*. Institute of Electrical and Electronics Engineers Inc., pp.72–81.

Dastile, X. and Celik, T., (2021) Making Deep Learning-Based Predictions for Credit Scoring Explainable. *IEEE Access*, 9, pp.50426–50440.

Dzik-Walczak, A. and Heba, M., (2021) An implementation of ensemble methods, logistic regression, and neural network for default prediction in peer-to-peer lending. *Zbornik Radova Ekonomskog Fakulteta u Rijeci*, 391, pp.163–197.

Egan, C., (2021) *Improving Credit Default Prediction Using Explainable AI MSc Research Project Data Analytics Improving Credit Default Prediction Using Explainable AI*. [online] Available at: <http://norma.ncirl.ie/5146/> [Accessed 28 Dec. 2021].

Faris, H., Abukhurma, R., Almanaseer, W., Saadeh, M., Mora, A.M., Castillo, P.A. and Aljarah, I., (2020) Improving financial bankruptcy prediction in a highly imbalanced class distribution using oversampling and ensemble learning: a case from the Spanish market. *Progress in Artificial Intelligence*, 91, pp.31–53.

Feng, X., Xiao, Z., Zhong, B., Qiu, J. and Dong, Y., (2018) Dynamic ensemble classification for credit scoring using soft probability. *Applied Soft Computing Journal*, 65, pp.139–151.

García, V., Marqués, A.I. and Sánchez, J.S., (2019) Exploring the synergetic effects of sample types on the performance of ensembles for credit risk and corporate bankruptcy prediction. *Information Fusion*, 47, pp.88–101.

Hadji Misheva, B., Hirska, A., Osterrieder, J., Kulkarni, O. and Fung Lin, S., (2021) *EXPLAINABLE AI IN CREDIT RISK MANAGEMENT A PREPRINT*. [online] Available at: <https://ssrn.com/abstract=3795322>.

Hamori, S., Kawai, M., Kume, T., Murakami, Y. and Watanabe, C., (2018) Ensemble Learning or Deep Learning? Application to Default Risk Analysis. *Journal of Risk and Financial Management*, 111, p.12.

He, H., Zhang, W. and Zhang, S., (2018) A novel ensemble method for credit scoring: Adaption of different imbalance ratios. *Expert Systems with Applications*, 98, pp.105–117.

Jin, Y., Zhang, W., Wu, X., Liu, Y. and Hu, Z., (2021) A Novel Multi-Stage Ensemble Model with a Hybrid Genetic Algorithm for Credit Scoring on Imbalanced Data. *IEEE Access*, 9, pp.143593–143607.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.-Y., (2017) *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*. [online] Available at: <https://github.com/Microsoft/LightGBM>.

Kun, Z., Weibing, F. and Jianlin, W., (2020a) Default Identification of P2P Lending Based on Stacking Ensemble Learning. In: *Proceedings - 2020 2nd International Conference on Economic Management and Model Engineering, ICEMME 2020*. Institute of Electrical and Electronics Engineers Inc., pp.992–1006.

Kun, Z., Weibing, F. and Jianlin, W., (2020b) Default Identification of P2P Lending Based on Stacking Ensemble Learning. In: *2020 2nd International Conference on Economic Management and Model Engineering (ICEMME)*. IEEE.

Laborda, J. and Ryoo, S., (2021) Feature selection in a credit scoring model. *Mathematics*, 97.

Lending club, (2021) *All Lending Club loan data | Kaggle*. [online] Available at: [https://www.kaggle.com/wordsforthewise/lending-club?select=accepted\\_2007\\_to\\_2018Q4.csv.gz](https://www.kaggle.com/wordsforthewise/lending-club?select=accepted_2007_to_2018Q4.csv.gz) [Accessed 7 Nov. 2021].

- Li, W., Ding, S., Chen, Y. and Yang, S., (2018) Heterogeneous ensemble for default prediction of peer-to-peer lending in China. *IEEE Access*, 6, pp.54396–54406.
- Li, Z., Zhang, J., Yao, X. and Kou, G., (2021) How to identify early defaults in online lending: A cost-sensitive multi-layer learning framework. *Knowledge-Based Systems*, 221.
- Lundberg, S.M., Allen, P.G. and Lee, S.-I., (2017) *A Unified Approach to Interpreting Model Predictions*. [online] Available at: <https://github.com/slundberg/shap>.
- Ma, X., Sha, J., Wang, D., Yu, Y., Yang, Q. and Niu, X., (2018) Study on a prediction of P2P network loan default based on the machine learning LightGBM and XGboost algorithms according to different high dimensional data cleaning. *Electronic Commerce Research and Applications*, 31, pp.24–39.
- Malik, R., (2021) An Impact of Loan Defaults and Impact on Profitability of Bank. *International Journal for Research in Engineering Application & Management (IJREAM)*, 0611, pp.2454–9150.
- Moscato, V., Picariello, A. and Sperlí, G., (2021) A benchmark of machine learning approaches for credit score prediction. *Expert Systems with Applications*, 165.
- Namvar, A., Siami, M., Rabhi, F. and Naderpour, M., (2018) *Credit risk prediction in an imbalanced social lending environment*.
- Niu, K., Zhang, Z., Liu, Y. and Li, R., (2020) Resampling ensemble model based on data distribution for imbalanced credit risk evaluation in P2P lending. *Information Sciences*, 536, pp.120–134.
- Nyitrai, T. and Virág, M., (2019) The effects of handling outliers on the performance of bankruptcy prediction models. *Socio-Economic Planning Sciences*, 67, pp.34–42.
- Oreški, S. and Oreški, G., (2018) Cost-sensitive learning from imbalanced datasets for retail credit risk assessment. *TEM Journal*, 71, pp.59–73.
- Pes, B. and Lai, G., (2021) Cost-sensitive learning strategies for high-dimensional and imbalanced data: a comparative study. *PeerJ Computer Science*, [online] 7, p.e832. Available at: <https://peerj.com/articles/cs-832>.
- Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V. and Gulin, A., (2017) CatBoost: unbiased boosting with categorical features. [online] Available at: <http://arxiv.org/abs/1706.09516>.
- Ri, J.H. and Kim, H., (2020) G-mean based extreme learning machine for imbalance learning. *Digital Signal Processing: A Review Journal*, 98.
- Ribeiro, M.T., Singh, S. and Guestrin, C., (2016) Model-Agnostic Interpretability of Machine Learning. [online] Available at: <http://arxiv.org/abs/1606.05386>.
- Roberto Lopez, (2021) *How to benchmark the performance of machine learning platforms: data capacity, training speed, inference speed and model precision | Neural Designer*. [online] Available at: <https://www.neuraldesigner.com/blog/how-to-benchmark-the-performance-of-machine-learning-platforms> [Accessed 7 Nov. 2021].
- Saidi Meryem, Habib Daho Mostafa El, Settouti Nesma and Amine Bechar Mohammed El, (2018) Comparison of ensemble cost sensitive algorithms application to credit scoring prediction. *International Conference on Advanced Aspects of Software Engineering*, 2326.
- Sandica, A.M. and Fratila, A., (2021) Implications of macroeconomic conditions on Romanian portfolio credit risk. A cost-sensitive ensemble learning methods comparison. *Economic Research-Ekonomska Istrazivanja*.
- Shen, F., Zhao, X., Kou, G. and Alsaadi, F.E., (2021) A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique. *Applied Soft Computing*, 98.
- Shen, F., Zhao, X., Li, Z., Li, K. and Meng, Z., (2019) A novel ensemble classification model based on neural networks and a classifier optimisation technique for imbalanced credit risk evaluation. *Physica A: Statistical Mechanics and its Applications*, 526.
- Song, Y. and Peng, Y., (2019) A MCDM-Based Evaluation Approach for Imbalanced Classification Methods in Financial Risk Prediction. *IEEE Access*, 7, pp.84897–84906.



- Sterner, P., Goretzko, D. and Pargent, F., (2021) *COST-SENSITIVE LEARNING 1 Everything has its Price: Foundations of Cost-Sensitive Learning and its Application in Psychology*. [online] Available at: <https://osf.io/cvks7/>.
- Tripathi, D., Edla, D.R., Bablani, A., Shukla, A.K. and Reddy, B.R., (2021) *Experimental analysis of machine learning methods for credit score classification. Progress in Artificial Intelligence*, .
- Wang, D. and Zhang, Z., (2020) Enterprise Credit Risk Assessment Using Feature Selection Approach and Ensemble Learning Technique. In: *Proceedings - 2020 16th International Conference on Computational Intelligence and Security, CIS 2020*. Institute of Electrical and Electronics Engineers Inc., pp.228–233.
- Wang, H., Kou, G. and Peng, Y., (2021) Multi-class misclassification cost matrix for credit ratings in peer-to-peer lending. *Journal of the Operational Research Society*, 724, pp.923–934.
- Wong, M.L., Seng, K. and Wong, P.K., (2020) Cost-sensitive ensemble of stacked denoising autoencoders for class imbalance problems in business domain. *Expert Systems with Applications*, 141.
- Wu, D., Guo, P. and Wang, P., (2020a) Malware Detection based on Cascading XGBoost and Cost Sensitive. In: *Proceedings - 2020 International Conference on Computer Communication and Network Security, CCNS 2020*. Institute of Electrical and Electronics Engineers Inc., pp.201–205.
- Wu, D., Guo, P. and Wang, P., (2020b) Malware Detection based on Cascading XGBoost and Cost Sensitive. In: *Proceedings - 2020 International Conference on Computer Communication and Network Security, CCNS 2020*. Institute of Electrical and Electronics Engineers Inc., pp.201–205.
- Wu, S., Gao, X. and Zhou, W., (2022) COSLE: Cost sensitive loan evaluation for P2P lending. *Information Sciences*, 586, pp.74–98.
- Xia, Y., Zhao, J., He, L., Li, Y. and Niu, M., (2020) A novel tree-based dynamic heterogeneous ensemble method for credit scoring. *Expert Systems with Applications*, 159.
- Yotsawat, W., Wattuya, P. and Srivihok, A., (2021a) A Novel Method for Credit Scoring Based on Cost-Sensitive Neural Network Ensemble. *IEEE Access*, 9, pp.78521–78537.
- Yotsawat, W., Wattuya, P. and Srivihok, A., (2021b) A Novel Method for Credit Scoring Based on Cost-Sensitive Neural Network Ensemble. *IEEE Access*, 9, pp.78521–78537.
- Zhang, T. and Li, J., (2021) Credit Risk Control Algorithm Based on Stacking Ensemble Learning. In: *Proceedings of 2021 IEEE International Conference on Power Electronics, Computer Applications, ICPECA 2021*. Institute of Electrical and Electronics Engineers Inc., pp.668–670.
- Zhang, W., He, H. and Zhang, S., (2019) A novel multi-stage hybrid model with enhanced multi-population niche genetic algorithm: An application in credit scoring. *Expert Systems with Applications*, 121, pp.221–232.
- Zhang, W., Yang, D. and Zhang, S., (2021) A new hybrid ensemble model with voting-based outlier detection and balanced sampling for credit scoring. *Expert Systems with Applications*, 174.