Strathmore Insitute of Mathematical Sciences

# Optimizing Credit Risk Assessment with Ensemble Sampling and Hybrid Machine Learning Models

Naomi Mucheru

*Supervisor:* Dr. Samuel Chege

Submitted in Partial fulfilment of the Requirements for the Degree of Master of Science in Mathematical Finance at Strathmore University

March 30, 2025

.

## Declaration

I declare that this work has not been previously submitted and approved for award of a degree by this or any other University. To the best of my knowledge and belief, the dissertation contains no material previously published or written by another person except where due reference is made in the dissertation itself.



Name: Naomi Wangari Mucheru

Signature: _____

Date: 27/03/2025

## Approval

The thesis of Naomi Wangari Mucheru was reviewed and approved for examination by the following:

Dr. Samuel Chege,
Senior Lecturer,Strathmore Institute of Mathematical Sciences
Sign: _____ Date: 30/03/2025

Dr. Godfrey Madigu,
Dean, Strathmore Institute of Mathematical Sciences,
Strathmore University
Sign: _____ Date: _____

Dr. Bernard Shibwabo,
Director of Graduate Studies,
Strathmore University
Sign: _____ Date: _____

# Abstract

Accurate credit risk modeling is essential for minimizing financial losses, but class imbalance, where defaulters make up a small fraction of the data, remains a challenge. This study tackles the issue using ensemble sampling and hybrid machine learning models. A Kaggle dataset with 32,582 entries was used in this study. SMOTE + Random Undersampling, ADASYN + Random Undersampling, Borderline-SMOTE + Random Undersampling, SVM-SMOTE + Random Undersampling, and SMOTE-TOMEK, were applied before training.

Our findings reveal that Random Forest with Borderline-SMOTE + Random Undersampling achieved the highest recall, while SMOTE + Random Undersampling with Random Forest achieved highest AUC. While hybrid machine learning models improved precision, they sacrificed recall.

This study reinforces the power of ensemble sampling and hybrid approaches in credit risk modeling, with future research focusing on dynamic thresholding and advanced ensemble strategies to refine predictions.

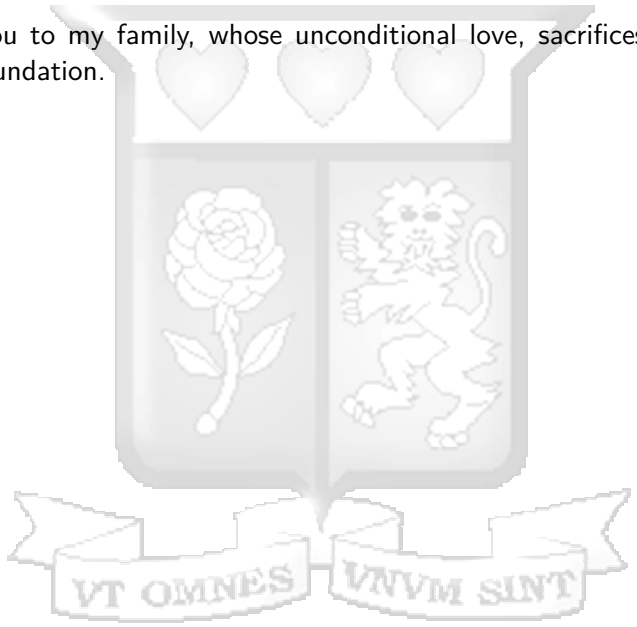
**Keywords:** Credit risk modeling, Class Imbalance,Ensemble sampling, Hybrid machine learning, Random Forest

# Acknowledgements

First and foremost, I am deeply grateful to God Almighty for His endless mercy, strength, and assurance throughout this journey.

Special thanks to my supervisor, Dr.Samuel Chege, for his invaluable guidance, patience, and unwavering support.

A heartfelt thank you to my family, whose unconditional love, sacrifices, and prayers have been my greatest foundation.



# Contents

# List of Figures



# List of Tables

# List of Abbreviations

| | |
|---|---|
| ADASYN | Adaptive Synthetic Sampling Approach |
| ANNs | Artificial Neural Networks |
| APT | Arbitrage Pricing Theory |
| AUC | Area Under the Curve |
| AUC - ROC | Area Under the Receiver Operating Characteristic curve |
| CAPM | Capital Asset Pricing Model |
| CART | Classification And Regression Tree |
| EM | Expectation Maximization |
| FN | False Negative |
| FP | False Positive |
| GBDT | Gradient Boosted Decision Trees |
| GDP | Gross Domestic Product |
| KNN | K-Nearest Neighbors |
| LDA | Linear Discriminant Analysis |
| MAP | Maximum A Posteriori |
| MLE | Maximum Likelihood Estimation |
| MSE | Mean Squared Error |
| RF | Random Forest |
| ROC | Receiver Operating Characteristic |
| SMOTE | Synthetic Minority Over-sampling Technique |
| SVM | Support Vector Machines |
| TN | True Negative |
| TP | True Positive |

# Chapter 1

# Introduction

Credit risk refers to the potential that borrowers may fail to meet their financial obligations, representing a critical concern for financial institutions. Banks employ credit scoring systems to classify customers as "good" (likely to repay) or "bad" (likely to default), supporting portfolio management and risk quantification. The Probability of Default (PD) stands as a key metric in credit risk assessment, combined with Loss Given Default (LGD) to compute Expected Loss (EL), which informs regulatory capital requirements established by financial regulators Berger and Di Patti (2006). The evolution of credit risk modeling began with expert judgment systems, which were costly and lacked standardization Valášková et al. (2014). The introduction of statistical approaches transformed the field, with linear regression, discriminant analysis, and logistic regression becoming foundational tools. Beaver's univariate discriminant analysis identified the cash-to-debt ratio as a strong predictor of corporate distress Beaver (1966), while Altman developed insolvency prediction models using linear discriminant analysis Altman (1968). Martin later applied logit models to predict bank failures Martin (1977).

Corporate credit risk modeling has historically dominated research, with structural models like Merton's approach Merton (1974) linking firm assets to credit risk through option pricing theory Merton (1973). Reduced-form models emerged later, treating default as a stochastic process rather than a deterministic outcome of asset values Jarrow et al. (1997). Value-at-Risk models, including Credit Metrics and credit portfolio view, further advanced risk assessment by incorporating portfolio perspectives Valášková et al. (2014). Consumer credit risk modeling gained prominence following the 2007 subprime mortgage crisis, highlighting the inadequacy of corporate-focused approaches for consumer lending portfolios. Research has since explored adapting corporate models to consumer contexts Perli and Nayda (2004) and developing specialized approaches incorporating economic variables and behavioral scores Malik and Thomas (2010).

Non-parametric estimation methods have gained appeal due to the limitations of parametric approaches, which rely on potentially misspecified assumptions. Machine learning Frydman et al. (1985) and neural networks Wilson and Sharda (1994) represent promising directions for future research in credit risk modeling. Machine learning, a subset of artificial intelligence, automates complex tasks by learning patterns from training data to generate hypotheses. It comprises two primary paradigms: supervised learning (using labeled input-output data) and unsupervised learning (analyzing input data alone). Hybrid approaches combining both paradigms have been explored for credit risk prediction. Common supervised algorithms include Support Vector Machines, Random Forests, Decision Trees, K-Nearest Neighbors, and Gradient Boosting, while unsupervised methods often rely on clustering techniques like K-

Means, K-Medoids, and X-Means. K-Means, noted for its simplicity, remains a dominant clustering algorithm in data mining Wu et al. (2008).

A critical challenge in credit risk modeling is data imbalance, where minority class instances (e.g., defaults) are vastly outnumbered by majority class instances. Standard classifiers tend to favor majority classes, reducing accuracy for rare events Kubat et al. (1998). Sampling techniques address this imbalance through undersampling (reducing majority instances) and oversampling (augmenting minority instances). Undersampling risks discarding valuable data and altering class distributions, while oversampling, particularly random replication, increases training time and overfitting risks. To mitigate overfitting, SMOTE (Synthetic Minority Oversampling Technique) was introduced in 2002, generating synthetic minority samples via interpolation Chawla et al. (2002). Numerous SMOTE variants have since emerged.

The 2008 global financial crisis underscored the critical role of credit risk assessment in financial stability, exposing vulnerabilities from relaxed lending standards and risky loans. In Kenya, escalating interest rates, inflation, and economic decline have driven loan defaults to an 18-year high of $4 billion (15% non-performing loan ratio as of August 2023), per Central Bank of Kenya (CBK) data. Such defaults threaten institutional profitability and systemic economic stability, necessitating robust credit risk evaluation. While traditional credit risk models rely on linear frameworks with limited predictive power, advancements in machine learning offer opportunities to address complex patterns and data imbalance inherent in credit datasets. Imbalanced data—where non-defaulters vastly outnumber defaulters—often biases models toward majority classes, exacerbating misclassification costs. Hybrid machine learning architectures, integrating supervised techniques, show promise in capturing non-linear relationships, yet their synergy with ensemble sampling methods remains underexplored.

This research aims to enhance the preciseness, robustness, and reliability of credit risk assessment by evaluating a hybridized framework that integrates two supervised machine learning algorithms with ensemble sampling techniques. The hybrid framework seeks to address two critical challenges in credit risk modeling: the complex, non-linear patterns inherent in credit data and the prevalence of data imbalance, where default instances are vastly outnumbered by non-default cases. By combining supervised learning algorithms, better generalization is expected due to reduced bias and variances, thereby improving prediction performance. Additionally, the research explores optimized ensemble sampling strategies—combining undersampling and oversampling methods—to create balanced, informative training datasets that mitigate bias toward majority classes. Systematic experimentation will identify synergistic combinations of sampling techniques to maximize model performance.

The evaluation of this integrated approach will involve rigorous testing using metrics such as F1-Score, recall, precision, and AUC-ROC, alongside comparative analysis against traditional single-machine learning models. Credit risk assessment datasets frequently suffer from imbalance, leading to models that underperform on minority-class predictions (defaults). This study addresses this issue by balancing default and non-default instances through strategic sampling. Furthermore, traditional linear models often fail to capture intricate relationships in credit data, prompting the adoption of hybrid machine learning architectures capable of harnessing both labeled and unlabeled data. Enhanced predictive accuracy not only improves risk management for financial institutions but also supports regulatory compliance by promoting robust risk assessment practices.

Finally, the integration of ensemble sampling with hybrid machine learning represents a novel methodological contribution to credit risk research. By advancing innovative techniques for imbalanced data and complex pattern recognition, this work offers transferable solutions to fields grappling with similar challenges, such as fraud detection or medical diagnosis. The research thus bridges theoretical innovation with practical applicability, aiming to provide financial institutions with a more reliable, accurate, and robust tool for credit risk evaluation.

# Chapter 2

# Literature Review

In this chapter, we provide a comprehensive overview of the literature surrounding credit risk assessment, tracing its development from foundational theories to modern empirical approaches. Understanding these developments is essential in contextualising contemporary methodologies in credit risk management.

The theoretical foundations of credit risk models have evolved considerably over time. One of the earliest and most influential contributions is the Modern Portfolio Theory, introduced in Markowitz (1952). This theory introduced the concept of constructing an investment portfolio by combining assets with varying levels of risk, with the objective of achieving optimal returns through diversification. The notion of the efficient frontier emerged from this work, offering investors a framework to balance risk and return by selecting portfolios that either maximise returns for a given risk level or minimise risk for a given expected return. Markowitz's work was further extended in the Capital Asset Pricing Model (CAPM) by Sharpe (1964), which provided a mechanism for estimating expected returns based on market risk factors. Both models, however, rest on assumptions such as normally distributed asset returns and constant correlations between assets, which have been challenged during periods of financial instability Hicks (1989).

Building upon these ideas, Arbitrage Pricing Theory (APT), developed by Ross (1976), presents a more flexible framework that considers multiple factors influencing asset returns. Unlike CAPM, which relies solely on market risk, APT accounts for macroeconomic variables such as inflation, GDP growth, and industry-specific factors, thus offering a more nuanced understanding of asset pricing. The theory rests on the principle that investors are risk-averse and that the expected return of an asset is linearly related to its exposure to various risk factors.

Parallel to these theoretical developments, practical credit assessment methods have also emerged. Among these, credit scoring models have become instrumental that estimate the probability of default based on financial and non-financial factors, using techniques such as logistic regression, discriminant analysis, and decision trees.. By drawing on data sources such as financial statements, credit histories, and industry benchmarks, these models enable lenders to systematically evaluate creditworthiness.

Empirical research in credit risk modelling began with seminal work of Durand (1941), which identified the potential of applicant characteristics in predicting loan outcomes. The widespread adoption of credit cards in the late 1960s catalysed the use of credit scoring, leading to substantial reductions in default rates. By the 1990s, scorecards became prevalent, further refining

these methods. Modern credit risk models encompass expert systems, statistical models, artificial intelligence approaches, and hybrid methodologies.

Statistical methods have long formed the backbone of credit risk modelling. Fisher (1936) pioneered Linear Discriminant Analysis (LDA), a foundational advancement in statistical classification. LDA projects the feature space onto a lower-dimensional subspace to maximize class separability, employing a linear discriminant function (LDF) that classifies customers based on a weighted combination of predictor variables:

$$LDF = a_0 + a_1x_1 + a_2x_2 + \cdots + a_nx_n, \tag{2.1}$$

where the coefficients $a_i$ quantify the contribution of each feature $x_i$. Despite its usefulness, LDA's reliance on assumptions of normality and equal covariance matrices across groups limits its applicability. Another method, logistic regression subsequently gained prominence due to its flexibility and interpretability. This approach models the log-odds of default as a linear function of predictor variables:

$$\log\left(\frac{P(Y=1 \mid X)}{1-P(Y=1 \mid X)}\right) = \beta_0 + \beta_1x_1 + \beta_2x_2 + \cdots + \beta_nx_n, \tag{2.2}$$

with $P(Y = 1|X)$ representing the probability of default given predictor values $X$. Studies such as Costa e Silva et al. (2020) demonstrate the relationship between loan characteristics and default probability, highlighting factors such as loan term and customer demographics.

Bayesian classification methods, rooted in Bayes' theorem, offer a probabilistic framework:

$$P(Y|X) = \frac{P(X|Y) \cdot P(Y)}{P(X)}, \tag{2.3}$$

with the Naive Bayes variant simplifying computations by assuming feature independence. Empirical evidence, such as the study by Choge (2012), supports the efficacy of this method in consumer credit evaluation.

The limitations of traditional statistical approaches have prompted the adoption of artificial intelligence techniques. Artificial Neural Networks (ANNs), have shown superior classification accuracy compared to linear models, as demonstrated by Crook et al. (1996). However, their lack of interpretability remains a concern in regulatory environments Baesens et al. (2003). Support Vector Machines (SVMs), introduced by Vapnik and Chervonenkis (1963), have also found applications in credit scoring. By defining optimal separating hyperplanes and employing hinge loss functions, these offer robust classification, albeit with computational challenges Chen et al. (2005).

Ensemble methods further enhance predictive performance by combining multiple models. Approaches such as bagging, boosting, and stacking allow for the aggregation of diverse classifiers. Random Forests and Gradient Boosting Decision Trees (GBDT) have demonstrated notable success, with studies by Nanni and Lumini (2009) and Charpignon et al. (2014) showing their ability to handle imbalanced data and achieve high accuracy. Hybrid models, which integrate clustering and classification techniques, leverage the strengths of multiple methodologies. In their work, Tsai and Chen (2010) classified hybrid models into various categories, with empirical results indicating the effectiveness of combinations such as EM+Logistic Regression and Logistic Regression+ANN.

Addressing class imbalance is also critical in credit risk modelling. Class imbalance arises when the proportion of non-default observations substantially exceeds that of default observations within the dataset. Such imbalance can impair model performance, leading to biased predictions that underestimate the likelihood of default and compromise risk assessment accuracy. In this context, sampling techniques are frequently applied to address class imbalance. Oversampling methods, such as the Synthetic Minority Oversampling Technique (SMOTE) introduced by Chawla et al. (2002), generate synthetic minority class examples through interpolation to reduce overfitting. Extensions, including Borderline-SMOTE Han et al. (2005), focus on generating synthetic instances near the decision boundary for improved performance. Undersampling, on the other hand, reduces the number of majority class observations, though it may result in loss of valuable information Kubat et al. (1997). Comparative analysis by Marqués et al. (2013) indicates that oversampling methods generally yield superior results compared to undersampling, especially in the presence of large datasets.

The regulatory landscape plays a pivotal role in shaping credit risk management practices. The Basel Accords, beginning with Basel I (Basel Committee on Banking Supervision (1988)), established the foundation for risk-based capital requirements, although its emphasis on fixed capital ratios and credit risk alone was later deemed inadequate. Basel II (Basel Committee on Banking Supervision (2004)), extended the framework through its three-pillar approach: minimum capital requirements, supervisory review, and market discipline, while also incorporating operational risk considerations. Following the 2007–2008 financial crisis, Basel III (Basel Committee on Banking Supervision (2011)) was introduced to strengthen the resilience of financial institutions, adding leverage ratios and liquidity requirements to better mitigate systemic risk. These regulatory milestones, alongside theoretical and empirical advancements, underpin the current framework for credit risk assessment, which continues to adapt in response to evolving market conditions and emerging risks.

# Chapter 3

# Methodology

This section examines the machine learning models for analyzing the credit dataset, the chosen study design and the methodology for evaluating the performance of the algorithm. This study aims to use ensemble sampling algorithms and hybrid supervised machine learning models to contribute to the existing literature on credit scoring.
Ensemble algorithms that combine undersampling and oversampling algorithms to address the imbalanced nature of the dataset, were applied.This research investigates several combinations of SMOTE variations and undersampling approaches.

The research consists of five combinations.

- SMOTE and Random Undersampling.
- ADASYN and Random Undersampling.
- Borderline-SMOTE and Random Undersampling.
- SVM SMOTE and Random Undersampling.
- SMOTE-TOMEK.

Regularized logistic regression was used in the data pre-processing stage for feature selection. Hybrid machine learning algorithms were employed to develop credit scoring models within the dataset. Finally, to assess their predictive accuracy and effectiveness, the appropriate metrics were used to evaluate the performance. For the implementation, the programming was done in Python for various tasks.

Logistic regression with a regularization term added to it's cost function, is regularized logistic regression. Logistic regression is a supervised learning method for classification. The term logistic refers to the logit(log odds) that is being modeled. Odds refer to the ratio of the probability that an event occurs to the probability that an event does not occur. The objective of logistic regression is to choose the coefficients that minimize misclassification. The logistic regression model is expressed as:

$$P(Y = 1 \mid X) = \frac{1}{1 + \exp(-(X\beta))}, \tag{3.1}$$

where:

- $P(Y = 1 \mid X)$ is the probability that the output $Y$ is 1, given the input $X$,

- $\beta$ is the vector of coefficients (parameters) associated with the features in $X$,
- $X\beta$ represents the dot product of the feature vector $X$ and the parameter vector $\beta$,
- exp is the exponential function.

The logistic regression model can also be expressed in terms of the **log-odds** (also known as the logit function):

$$\log\left(\frac{P(Y=1 \mid X)}{1-P(Y=1 \mid X)}\right) = X\beta. \tag{3.2}$$

where:

- The left-hand side represents the log of the odds ratio of $Y=1$ to $Y=0$,
- $X\beta$ is the linear combination of the features and the coefficients.

The derivation of the above expression has been shown in Appendix C.
Logistic regression estimates the coefficients $\beta$ that best fit the data by maximizing the likelihood of observing the given set of outcomes. The model predicts $P(Y=1 \mid X)$ and not the class label. When $P(Y=1 \mid X)$ is greater than a critical probability the class label ( 1 ) is assigned.Because the likelihood functions are a product of probabilities, the logarithm of the likelihood is used instead. In binary classification problems, where the outcome variable $y \in \{0,1\}$, logistic regression is a commonly used method for modeling the probability that the response variable equals one, given a set of independent features $\mathbf{x}$. Logistic function, estimates the probability $P(y=1 \mid \mathbf{x})$ as a function of the input ,which ensures that the predicted probabilities is between 0 and 1. In logistic regression model, the binary outcome variable $y$ can be expressed using a Bernoulli distribution. Probability of $y=1$ given the input features $\mathbf{x}$ is represented as:

$$P(y=1 \mid \mathbf{x};\mathbf{w}) = h(\mathbf{x};\mathbf{w}) = \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x})}, \tag{3.3}$$

where $h(\mathbf{x};\mathbf{w})$ is the logistic function, and $\mathbf{w}$ represents the parameters of the model. The logistic function maps the linear combination of input features, $\mathbf{w}^T\mathbf{x}$, to a probability value between 0 and 1. Since the dependent variable is binary, the probability mass function (PMF) of the Bernoulli distribution is used to explain the probability of getting a particular outcome $y$, given by:

$$P(y \mid \mathbf{x};\mathbf{w}) = h(\mathbf{x};\mathbf{w})^y \cdot (1-h(\mathbf{x};\mathbf{w}))^{1-y}. \tag{3.4}$$

In this term, $h(\mathbf{x};\mathbf{w})$ is the predicted probability of obtaining $y=1$, and $1-h(\mathbf{x};\mathbf{w})$ is the predicted probability of obtaining $y=0$. This expression shows the likelihood of the binary outcome $y$, conditional on the input $\mathbf{x}$ and the model parameters $\mathbf{w}$.
To estimate the parameters $\mathbf{w}$ in logistic regression, we maximize the likelihood function, which expresses the probability of observing the data given the model parameters. The likelihood for a single data point $(\mathbf{x}_i, y_i)$ is:

$$\mathcal{L}_i(\mathbf{w}) = P(y_i \mid \mathbf{x}_i;\mathbf{w}) = \Big(\sigma(\mathbf{w}^T\mathbf{x}_i)\Big)^{y_i} \cdot \Big(1-\sigma(\mathbf{w}^T\mathbf{x}_i)\Big)^{1-y_i}. \tag{3.5}$$

where $\sigma(\mathbf{w}^T\mathbf{x}_i) = \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x}_i)}$ is the logistic function. Taking the natural logarithm of the likelihood function simplifies the product into a sum, giving us the log-likelihood for a

single observation. To facilitate optimization, the negative log-likelihood (NLL) is defined and rewritten as:

$$\text{NLL}(\mathbf{w}) = -\sum_{i=1}^{N} \left[ y_i \log \sigma(\mathbf{w}^T \mathbf{x}_i) + (1 - y_i) \log(1 - \sigma(\mathbf{w}^T \mathbf{x}_i)) \right]. \tag{3.6}$$

The negative log likelihood serves as the loss function in logistic regression. During model training, the goal is to minimize this function to obtain the optimal parameters $\mathbf{w}$. By minimizing the NLL, we ensure that the model's predictions are as close as possible to the true binary outcomes. This is typically achieved using iterative optimization methods, such as gradient descent, which adjust the model parameters in the direction that decreases the NLL, thereby improving the fit of the model to the data.
One major drawback of logistic regression in feature selection is overfitting, especially when there are many features or when some features are highly correlated. This happens because logistic regression tries to fit the data as well as possible, which can lead to complex models that do not generalize well to new data.
Regularization in Logistic regression, is used to prevent overfitting. When dealing with high - dimensional data, regularization adds penalties to the logistic regression loss function to prevent overfitting. L1 Regularization (Lasso) and L2 Regularization (Ridge) are the two types of regularization.

- **L1 Regularization (Lasso):** Adds a penalty proportional to the absolute value of the coefficients. This encourages the model to shrink less important feature coefficients to zero, effectively performing feature selection by eliminating unnecessary features.

$$J(\theta) = -\frac{1}{m} \left[ \sum_{i=1}^{m} y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right] + \lambda \sum_{j=1}^{n} |\theta_j|. \tag{3.7}$$

- **L2 Regularization (Ridge):** Adds a penalty proportional to the square of the coefficients. This helps reduce the magnitude of coefficients without eliminating them completely, which prevents overfitting by discouraging extreme values.

$$J(\theta) = -\frac{1}{m} \left[ \sum_{i=1}^{m} y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right] + \frac{\lambda}{2} \sum_{j=1}^{n} \theta_j^2. \tag{3.8}$$

In this study Lasso regularization was used in feature selection because it removes irrelevant features by shrinking the coefficients to zero.
Instead of using Maximum Likelihood Estimation (MLE), which assumes no prior knowledge about the parameters, regularized logistic regression uses Maximum A Posteriori (MAP) estimate. It combines the prior belief about the parameters with the likelihood of the observed data. The objective of the Maximum A Posteriori (MAP) estimate in regularized logistic regression is to find the most probable values of the model parameters $\theta$ given the observed data, while incorporating prior knowledge or assumptions about these parameters.
In essence, MAP estimation combines the likelihood of the data (as used in Maximum Likelihood Estimation) with a prior distribution over the parameters, effectively leading to regularization. The MAP estimate can be expressed as:

$$\theta_{\text{MAP}} = \arg\max_{\theta} \left[ \log P(\mathcal{D} \mid \theta) + \log P(\theta) \right], \tag{3.9}$$

where:

- $P(\mathcal{D} \mid \theta)$ is the likelihood of the data $\mathcal{D}$ given the parameters $\theta$,
- $P(\theta)$ is the prior distribution over $\theta$.

The objective of MAP estimation is to obtain the parameter vector $\theta$ that maximizes the posterior distribution, which can be explained using Bayes' Theorem:

$$P(\theta|\mathbf{X}, \mathbf{y}) = \frac{P(\mathbf{y}|\mathbf{X}, \theta)P(\theta)}{P(\mathbf{y}|\mathbf{X})}, \tag{3.10}$$

where $P(\theta|\mathbf{X}, \mathbf{y})$ is the posterior distribution (the probability of the parameters $\theta$ given the data), $P(\mathbf{y}|\mathbf{X}, \theta)$ is the likelihood (the probability of observing the data $\mathbf{y}$ given the parameters $\theta$), $P(\theta)$ is the prior (the belief about the parameters $\theta$ before observing the data), and $P(\mathbf{y}|\mathbf{X})$ is the evidence (the marginal likelihood, which normalizes the posterior). The aim of MAP estimation is to find the $\theta$ that maximizes the posterior distribution:

$$\theta_{\text{MAP}} = \arg\max_{\theta} P(\theta|\mathbf{X}, \mathbf{y}). \tag{3.11}$$

Using Bayes' Theorem, this can be rewritten as:

$$\theta_{\text{MAP}} = \arg\max_{\theta} \frac{P(\mathbf{y}|\mathbf{X}, \theta)P(\theta)}{P(\mathbf{y}|\mathbf{X})}. \tag{3.12}$$

Since the evidence $P(\mathbf{y}|\mathbf{X})$ does not depend on $\theta$, we can ignore it during the maximization:

$$\theta_{\text{MAP}} = \arg\max_{\theta} P(\mathbf{y}|\mathbf{X}, \theta)P(\theta). \tag{3.13}$$

For logistic regression, the likelihood of observing the labels $\mathbf{y}$ given features $\mathbf{X}$ and parameters $\theta$ is:

$$P(\mathbf{y}|\mathbf{X}, \theta) = \prod_{i=1}^{n} \sigma(x_i^\top \theta)^{y_i}(1 - \sigma(x_i^\top \theta))^{1-y_i}, \tag{3.14}$$

where $\sigma(\cdot)$ is the sigmoid function:

$$\sigma(z) = \frac{1}{1 + \exp(-z)}. \tag{3.15}$$

The log-likelihood function can be expressed as:

$$\log P(\mathbf{y}|\mathbf{X}, \theta) = \sum_{i=1}^{n} \left[ y_i \log \sigma(x_i^\top \theta) + (1 - y_i) \log(1 - \sigma(x_i^\top \theta)) \right]. \tag{3.16}$$

By the Central Limit Theorem, the distribution of the sum of a large number of independent, identically distributed random variables is asymptotically normally distributed whatever the distribution of the underlying variables. So, we introduce a Gaussian prior over $\theta$:

$$P(\theta) = \mathcal{N}(\theta|0, \lambda^{-1}\mathbf{I}) = \frac{1}{(2\pi\lambda^{-1})^{d/2}} \exp\left( -\frac{\lambda}{2} \|\theta\|^2 \right), \tag{3.17}$$

where $\lambda > 0$ is the regularization parameter (inverse of variance), controlling the strength of the prior. Taking the log of the prior, we have:

$$\log P(\theta) = -\frac{\lambda}{2} \|\theta\|^2 + \text{constant}. \tag{3.18}$$

The log of the posterior can be written as:

$$\log P(\theta|\mathbf{X}, \mathbf{y}) = \log P(\mathbf{y}|\mathbf{X}, \theta) + \log P(\theta) + \text{constant}. \tag{3.19}$$

Substituting the expressions for the log-likelihood and the log-prior, we get:

$$\log P(\theta|\mathbf{X}, \mathbf{y}) = \sum_{i=1}^{n} \Big[ y_i \log \sigma(x_i^\top \theta) + (1 - y_i) \log(1 - \sigma(x_i^\top \theta)) \Big] - \frac{\lambda}{2} \|\theta\|^2 + \text{constant}. \tag{3.20}$$

To find $\theta_{\text{MAP}}$, we maximize the log-posterior:

$$\theta_{\text{MAP}} = \arg\max_{\theta} \left[ \sum_{i=1}^{n} \Big( y_i \log \sigma(x_i^\top \theta) + (1 - y_i) \log(1 - \sigma(x_i^\top \theta)) \Big) - \frac{\lambda}{2} \|\theta\|^2 \right]. \tag{3.21}$$

Maximizing this expression is equivalent to minimizing the following regularized cost function:

$$J(\theta) = -\sum_{i=1}^{n} \Big[ y_i \log \sigma(x_i^\top \theta) + (1 - y_i) \log(1 - \sigma(x_i^\top \theta)) \Big] + \frac{\lambda}{2} \|\theta\|^2. \tag{3.22}$$

The first term corresponds to the standard logistic regression loss (negative log-likelihood), and the second term is the regularization penalty. By minimizing $J(\theta)$, we are effectively performing MAP estimation, balancing the data fit and the prior belief.

When it comes to handling class imbalance, SMOTE is a popular oversampling method. It generates synthetic examples after identifying K-nearest neighbors for each minority class instance. It generates patterns from the minority class by performing linear interpolation between the original instance and a randomly chosen K-nearest neighbor.Many methods have expanded upon the SMOTE technique due to it's simplicity and performance. One variation is Borderline-SMOTE that is particularly effective in instances where the decision boundary between classes is complex. It targets instances in the minority class that are near the decision boundary.Another variation is the Adaptive Synthetic Sampling Approach for Imbalance Learning(ADASYN). It uses an adaptive mechanism to focus on areas of the feature space where the imbalance is more severe. It adjusts the density of the synthetic samples based on the distribution of the minority instance. Another one is the SVM-SMOTE that focuses on increasing the minority class instances along the decision boundary. The argument behind this is that instances around the boundary are critical for estimating the optimal decision boundary.SMOTE-TOMEK, SMOTE-EMV and SMOTE-CUT are ensemble algorithms that combine both oversampling and undersampling techniques. They integrate SMOTE with undersampling techniques(Tomek links, EMV and cluster-based undersampling) respectively, to achieve better balance in imbalanced datasets.SMOTE-TOMEK combines SMOTE oversampling and Tomek links undersampling. Tomek links are pairs of instances, one from the majority class and another from the minority class but are very close to each other. They are considered as noise which are removed by Tomek links undersampling technique. SMOTE-EMV combines SMOTE with Edited Multi-layer Voronoi where the latter is an undersampling technique that removes noisy examples by examining the Voronoi diagram of the data. SMOTE-CUT used both SMOTE and clustering-based undersampling techniques. The efficacy of SMOTE as a reliable sampling algorithm relies on it's capacity to generate patterns that adhere closely to the true one.
The SMOTE procedure is as follows;

**Algorithm 1** SMOTE Algorithm

**Input:** Minority class samples $X_{min}$, number of synthetic samples $N_{synth}$, $k$ neighbors
**Output:** Synthetic samples $X_{synth}$

$X_{synth} \leftarrow$ Empty list
**for** each $x \in X_{min}$ **do**
    Find $k$ nearest neighbors of $x$ from $X_{min}$ using some distance metric
    **for** $i = 1$ to $N_{synth}$ **do**
        Choose a random nearest neighbor $x_{nn}$ from $k$ nearest neighbors of $x$
        Generate a synthetic sample $x_{new}$ as:
        $x_{new} \leftarrow x + \text{random\_uniform}(0, 1) \times (x_{nn} - x)$
        Add $x_{new}$ to $X_{synth}$
    **end for**
**end forreturn** $X_{synth}$

Elreedy and Atiya (2019), offered both theoretical and experimental analysis of the SMOTE method and found that the mean vector of the patterns generated by SMOTE closely approximates the true mean vector. It was noted that the covariance matrix exhibited some discrepancies, as the patterns generated by SMOTE tend to be more compact than the true patterns. This indicates that the SMOTE generation process positions the new patterns more towards the center.
Later, Elreedy et al. (2023) performed a new theoretical analysis of the SMOTE technique which is obtained by deriving the probability distribution of sample created with SMOTE. This development permits the synthetic samples' density to be compared with that of the true underlying class-conditional density. In the theoretical analysis the next theorem was proved to be true.
Let $x$ be a random sample of a random variable X. Let Z be a random variable defined as a random linear interpolation between K-nearest neighbors of $x_k$ as given in Algorithm 1. Then, the probability density of Z is given by:

**Theorem 3.0.1.** *Let $\{x_i\}_{i=1}^{n}$ be a random sample of a random variable $X$. Let $Z$ be a random variable defined as a random linear interpolation between $K$-nearest neighbors of $x$ as given in Algorithm 1. Then, the probability density of $Z$ is given by:*

$$p_Z(z) = (N-K)\left(\frac{N-1}{K}\right)\int_x p_X(x)p_X\left(x+\frac{(z-x)r}{\|z-x\|}\right)\left(\frac{r^{d-2}}{\|z-x\|^{d-1}}\right) * B(1-I_{B(x,r)}; N-K-1, K)drdx$$

*where $N$ is the total number of samples, $K$ is the number of nearest neighbors, $p_X(x)$ is the probability density function of the random variable $X$, $r$ is the interpolation ratio, and $d$ is the dimensionality of the random variable $X$.*

Transitioning to machine learning, in unsupervised learning, data lacks the target attribute. The aim is to describe hidden structures from the unlabeled data. This is done by exploring the data to find some intrinsic structures in them. One form of unsupervised learning, clustering, does the grouping of a set of objects in such a way that objects in the same cluster are more similar to each other. It is useful as a data pre-processing technique for further analysis and automatic data organization. Major clustering approaches include partitioning, hierarchical, model-based and density-based. Partitioning involves constructing a number of different partitions and evaluating those according to some criteria. Hierarchical approach

creates a hierarchical decomposition of the set of objects using a established criteria. Model-based, hypothesizes a model for each cluster and finds best fit of models to data. The guiding principles of density-based methods are connectivity and density functions.

K-means Clustering is an unsupervised machine learning algorithm, known for it's efficiency and simplicity in partitioning data into clusters. It has a strong ability in handling large datasets efficiently. K-means algorithm operates by iteratively refining cluster assignments and centroid locations until a convergence criterion is met.
Consider a training set with input data points $x_1, x_2, x_3, ...., x_n$ in $R^d$ and value of k as the number of clusters needed. The steps below are followed;

- Choose k random points from the dataset as the starting centroids.
- The Euclidean distance between each point in the dataset and the preselected k points - centroids, should be calculated.
- Assign each point to the closest centroid in terms of the distance that has been calculated.
- Now, find the new centroids, which are actually the average points in each cluster group.
- Repeat the second and fourth steps for a fixed number of iterations or when centroids no longer change.

The euclidean distance between two points in space is;

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}. \tag{3.23}$$

if x=$(x_1, x_2)$ and y=$(y_1, y_2)$
The K-means optimization problem is;
Among all K-partitions $C_1 U C_2 U ..... U C_K = P$, find one that minimizes ;

$$\min_{C_1 U C_2 U ..... U C_K = P} \sum_{i=1}^{k} \sum_{x \epsilon C_i} \parallel x - \frac{1}{\mid C_i \mid} \sum_{x_j \epsilon C_i} x_j \parallel^2. \tag{3.24}$$

In supervised learning, a training set S of examples $x \epsilon X$ and their corresponding output value $y \epsilon Y$ is given. X = $x_1, x_2, x_3, ....x_n$ which is the set of all possible values of X. Each example $x$ is a vector of feature values. The feature data type can either be numeric or categorical.
The training set $T$ is composed of n instances;

$$T = (x_1, y_1), (x_2, y_2), (x_3, y_3), .....(x_n, y_n). \tag{3.25}$$

The aim of supervised learning is to approximate a function $h : X \rightarrow Y$ that can map $x_i$ to it's output $y_i$. The mapping is conducted using a learning algorithm known as an inducer. In training set, a single instance of an inducer is referred to as a classifier Rokach (2010).

Support vector machines (SVM) are a machine learning algorithm which finds the separating hyperplane with the largest margin of separation between two classes, when given a set of linearly separable points in $R^n$. Kecman (2001) has described the model clearly.
SVM finds the optimal hyperplane by solving the following optimization problem for the weight $w$;

$$\min_{w,b} \frac{1}{2} w^T w,$$
$$subject\ to: y_i(\langle w.x_i \rangle + b) - 1 \geq 0$$

Consider a set of training data $\{x_i\}_{i=1}^n$ drawn from a space $X$ and associated classes $\{y_i\}_{i=1}^m$; we assume $X \subseteq R^n$ and each $y_i \epsilon \{1, -1\}$. We also assume that there exists a hyperplane in $R^n$ such that each data point $x_i$ with $y_i = -1$ is on one side of the hyperplane and point $x_i$ with $y_i = 1$, is on the opposite side of the hyperplane. The classifier is defined as $f(x) = w^T x + b$ where $w$ and $b$ are the parameters that must be learned. The sign of $f(x) \epsilon \{-1, 1\}$ determines which of the two classes the classifier predicts as the class of data point $x$. Support vectors define the decision boundary and have non-zero Lagrangian multiples. To find the optimal hyperplane that separates different classes, the Lagrange function is used in the formulation of the optimization problem. It is given by;

$$L_p(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{M} (\alpha_i \{y_i(\langle w.\overline{x}_i \rangle + b) - 1\} \tag{3.26}$$

where $\alpha_i$ is the Lagrange multipliers, thus $\alpha_i \geq 0$.

$$\frac{\partial L}{\partial \overline{w}} = \overline{w} - \sum_{i=1}^{m} \alpha_i y_i \overline{x}_i = 0,$$
$$\overline{w} = \sum_{i=1}^{m} \alpha_i y_i \overline{x}_i$$

This suggests that $\overline{w}$ is a linear sum of some of the samples where $\alpha_i$ is non-zero.

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{m} \alpha_i y_i = 0,$$
$$\frac{1}{2} \sum_{i=1}^{m} \alpha_i y_i \overline{x}_i \sum_{j=1}^{m} \alpha_j y_j \overline{x}_j - \sum_{i=1}^{m} \alpha_i y_i \overline{x}_i \sum_{j=1}^{m} \alpha_j y_j \overline{x}_j - b \sum_{i=1}^{M} \alpha_i y_i + \sum_{i=1}^{M} \alpha_i.,$$
$$L_D(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=i} \sum_{j=i} \alpha_i \alpha_j y_i y_j x_i . x_j.$$

Above is the dual Lagrangian $L_D(\alpha)$ that is maximized with respect to the non negative $\alpha_i$ to obtain the optimal hyperplane with parameters $w$ and $b$.

K-Nearest Neighbors algorithm is a method in machine learning used for both classification and regression. It's key assumption is that similar data points tend to have similar labels. It is a non-parametric algorithm which makes no assumptions about the underlying distribution of the data. We assume we have training data D given by:

$$D = (x_1, y_1), (x_2, y_2), ...., (x_n, y_n) \subseteq X^d * Y \tag{3.27}$$

y = 1, 2, ..., $M$ M-class classification. For a point $x \epsilon X^d$, we define a set $S_x \subseteq D$ as a set of K neighbors. Using a function 'dist' that computes the distance between two points in $X^d$, we can define a set $S_x$ of size K as:

$$dist(x, x') \geq \max_{(x'', y'') \epsilon S_x}, \forall (x', y') \epsilon D \backslash S_x \tag{3.28}$$

The choice of the hyper-parameter K is important as it impacts the algorithm's performance. The choice of distance metric such as Minkowski distance, Euclidean distance, Manhattan distance, should fit with the characteristics of the data.

Another supervised learning algorithm, is a decision tree which is actually non-parametric and could be applied to classify or to perform regression. It consists of a root node, branches, internal nodes and leaf nodes. In this research, since our target variables(Pseudo-residuals) are continuous, the best approach for splitting the nodes, is selecting a split the yields the highest reduction in the variance of the values stored. When using decision trees, there are instances where one leaf gets more than one residual. And since the gamma($\gamma$) is added to the $log(odds)$ not to the probabilities directly, optimal output $\gamma_{jm}$ for each leaf node need to be found.

$$\gamma_{jm} = argmin_{\gamma} \sum_{x_i \epsilon R_j m} L(yi, F_{m-1}(x_i) + \gamma). \tag{3.29}$$

Region $R_{jm}$ is the pseudo-residuals, while j is the leaf number in the tree. The above equation can be rewritten as:

$$\gamma_{jm} = argmin_{\gamma} \sum_{x_i \epsilon R_j m} -y_i * [F_{m-1}(x_i) + \gamma] + \log(1 + e^{F_{m-1}(x_i)+\gamma}). \tag{3.30}$$

Appendix B, shows how to solve for the optimal gamma;

$$\gamma_{jm} = \frac{\sum\limits_{x_i \epsilon R_{jm}} (y_i - pi)}{\sum\limits_{x_i \epsilon R_{jm}} p_i(1 - p_i)}. \tag{3.31}$$

There have been a number of advancements aimed at improving decision trees. Among the advancements, the utilization of random forest algorithms stands out. Random forest algorithm, was developed to respond to the issues of tree instability. Breiman (2001)'s paper which defines Random Forests, serves as the cornerstone for improving this algorithm.
It creates an ensemble decision trees to improve the accuracy and robustness of predictions. Random Forest is constructed using multiple decision trees and the final decision is obtained by the majority votes of the decision trees. If the problem is a classification type, decision tree algorithm takes the majority votes. If the problem is of regression type, decision tree algorithm takes the mean of the decisions. Random Forests, uses bagging method to draw multiple training sample sets that are different from each other. On every single sample set, a decision tree is built with randomly selected attributed. Classification and Regression Trees models are normally used in problems with high-dimensional feature spaces. Since Random Forest method uses a large number of decision trees, it is characterized with good ability to resist noise and good performance in the classification accuracy. It is defined as a set of decision trees $h(x, \theta_k), k = 1, ....$, where $h(x, \theta_k)$ is a meta-classifier, which is a unpruned decision tree; $x$ serves as the input vector, while $\theta_k$ is an independent and identically distributed random vector. The Random Forest model, uses every variable to give predictions unlike CART where the variables not selected do not interfere with the response. This estimates the importance of each feature and also reduces the generalization error. To improve the hierarchy of the predictors, since a prediction is based on all explanatory variables, RF gives a ranking based on two measures. These two measures are Mean Decrease Accuracy and Mean Decrease

Impurity.
Mean Decrease Accuracy(Breiman,2001);

$$MDA(X_j) = \frac{1}{B}\sum_{1}^{B}(e_{OBB} - e_{OBBj}i) \tag{3.32}$$

where $e_{OBB}$ is the error rate on the out-of-bag sample.
Mean Decrease Impurity(Breiman,2001);

$$MDA(X_j)_{classification} = \frac{1}{B}\sum_{1}^{b}\sum_{t\epsilon T_b} I(j_t* = j)(\frac{N_t}{N}\delta_i(s,t)) \tag{3.33}$$

Gradient-Boosted Decision Trees is an ensemble algorithm based on multiple weak learners(decision trees). It is based on a combination of boosting and gradient descent. In Machine Learning there are mainly two types of errors, namely bias error and variance error. This algorithm mainly helps to minimize the bias error by reducing the underfitting problem. The main objective behind the Gradient Boosting algorithm is to train weak learners sequentially by reducing the errors of the previous weak learner. This is achieved by building the next weak learner on the residuals of the previous weak learner.By doing this, GBDT can reach the target by decreasing the residual error in the training process. When the dependent variable is continuous, Gradient Boosting Regressor is used, whereas when it's a classification problem, Gradient Boosting Classifier is used. The main difference between the two algorithms is the Loss function. For regression problems the Loss function can be like the Mean Squared Error (MSE) while for classification problems it can be the Log-likelihood. Since our problem is a classification problem, Log-likelihood loss function is used. The main objective is to minimize the Loss function, thus the first step is as follows:

$$F_0(x) = argmin_\lambda \sum_{i=1}^{n} L(y_i, \lambda). \tag{3.34}$$

L is the loss function, Gamma ($\lambda$) is the predicted value and $y(i)$ is the observed values. The first step involves finding a gamma value for which the loss function is minimum. In regression problems, this is normally the mean. In classification problems, we get the constant value using log(odds).
Then the model is expanded in a greedy fashion:

$$F_m(X) = F_{m-1}(X) + vh_m(X). \tag{3.35}$$

M is the sum of the weak classifiers and $h_m$ is the newly added base learner. $v(0 < v < 1)$ is the learning rate, which controls the step size of the gradient descent. This is for regularization to reduce overfitting. The Loss function has to be differentiable to enable the derivation of the gradient.In binary classification problem:

$$p = P(Y = 1|x) = \frac{1}{1 + e^{-F(x)}}. \tag{3.36}$$

$$F(x) = \log(\frac{p}{1-p}). \tag{3.37}$$

$F(x)$ is the logarithm of the odds ratio, where by it's a function of $p$.
Below is the loss function of a classification problem:

$$L = -\sum_{i=1}^{n} y_i \log(p) + (1 - y_i) \log(1 - p). \tag{3.38}$$

Appendix A, shows the transformation of the above loss function to

$$L = -y \log(odds) + \log(1 + e^{\log(odds)}). \tag{3.39}$$

Above is the loss function in terms of $log(odds)$. We then differentiate it so as to compute the gradient at a point $F_{m-1}(X_i)$.
Using Chain rule, the derivative of the $log$ of something, is 1 over that something multiplied by the derivative of that something, we have:

$$\frac{dL}{d[\log(odds)]} = -y + \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}. \tag{3.40}$$

Since:

$$\frac{e^{\log(odds)}}{1 + e^{\log(odds)}} = p. \tag{3.41}$$

$$\frac{dL}{d[\log(odds)]} = -y + p. \tag{3.42}$$

It is then multiplied by -1, to get the negative gradient.

$$= -(-y + p). \tag{3.43}$$

$$= y + p, = (Observed - Predicted). \tag{3.44}$$

The above formula will give the residuals, which we build the weak learners(decision trees), with all independent variables and dependent variables as residuals.

In measuring the effectiveness and capability of machine learning models, performance metrics play an important role. These metrics assist practitioners to understand the models' strength and weaknesses by providing quantitative measures. The data is split into two sets; training and test data. This allows the model to learn relevant parameters on the training data, then assess how the model performs on the test data. Efficiency of a model is calculated using a number of measures which are: Confusion matrix, Accuracy, Precision, Recall, F1-score, and Area Under the Curve(AUC).

Table 3.1 represents a Confusion matrix which is utilized in describing the performance of machine learning models on test data for which the true values are known.

Table 3.1: Confusion Matrix

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

True Positive(TP); This is the number of positive examples correctly classified. In this study

it will involve the correct number of defaults that have been predicted by the model that was actually defaults in the actual data..
True Negative(TN); This is the number of correctly classified negative examples. In our case, these are the number of non-defaults correctly predicted by the model.
False Positive(FP); That would be the number of negative examples that have been classified as positive instances. These are non-defaults that have been classified as defaults in our case.
False Negative(FN); This is the number of positive examples which have been classified as are those which must be classified as non-defaults. In our case these are defaults that have been classified as non-defaults.

Accuracy is the ratio of the number of correctly classified instances to the total number of instances. It is given by the following formula(Al-Shayea et al., 2010);

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.45}$$

Accuracy may not always be the most informative metric especially when dealing with imbalanced datasets.To address this limitation, precision, recall and F1-score come into play.
Precision: It is the ratio of true positive predictions to all positive predictions.

$$Precision = \frac{TP}{TP + FP} \tag{3.46}$$

Recall: Measures the proportion of true positive predictions of all actual positive instances. It focuses on the model's ability to capture all positive instances.

$$Recall = \frac{TP}{TP + FN} \tag{3.47}$$

F1- score: Is the harmonic mean between the Recall and Precision given by:

$$F1 - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.48}$$

It provides a balanced measure that considers both false positives and false negatives.

The Receiver Operating Characteristic(ROC) curve, is made by plotting False Positive Rate on the X-axis and True Positive Rate on the Y-axis. It visualizes the trade-offs between the positive and false positive predictions. Each point on the curve represents a specific threshold, indicating how the model's performance changes as the threshold for classifying instances as positive or negative varies. A classifier is said to be perfect if an ROC curve passes through the point (0,1). It represents a 100% sensitivity and 0% false positive rate. The area under the ROC curve is a single scalar metric that describes the performance of the classifier for all points. It provides insights into the ability to discriminate between positive and negative instances across various thresholds.
The figure 3.1 shows an example of an ROC curve:

Precision-Recall curve illustrates the trade-off between precision and recall across different thresholds.

It visualizes how precision and recall change as the classification threshold varies. Research in credit risk modeling indicates that the precision-recall curve is a more informative metric for

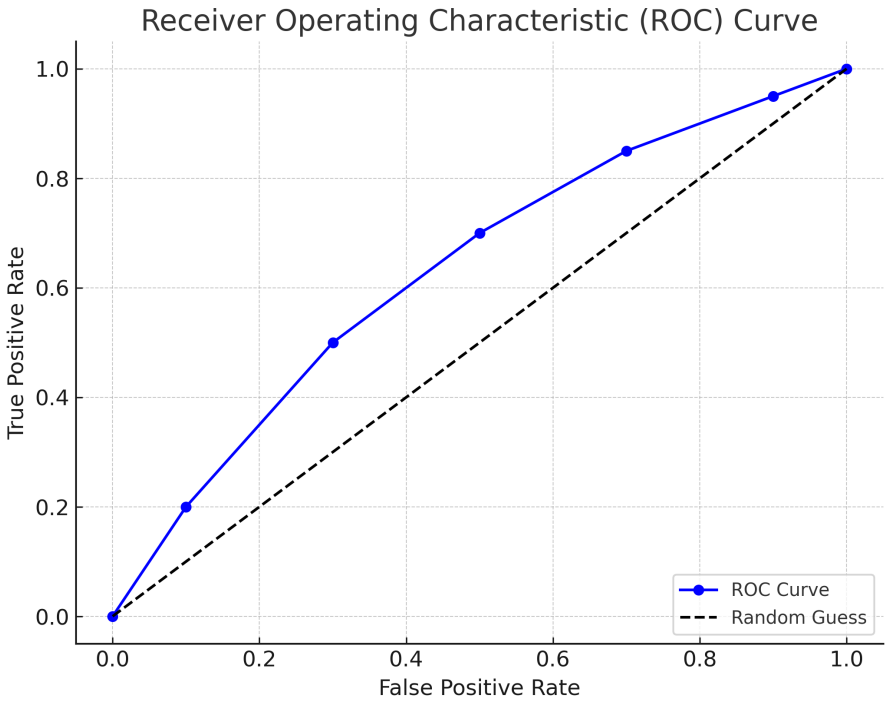



Figure 3.1: Receiver Operating Characteristic Curve

assessing the performance of binary classifiers on imbalanced datasets. This is because the precision-recall curve focuses on the positive class, which is of greater importance in these scenarios, unlike the ROC curve(Saito & Rehmsmeier, 2015).

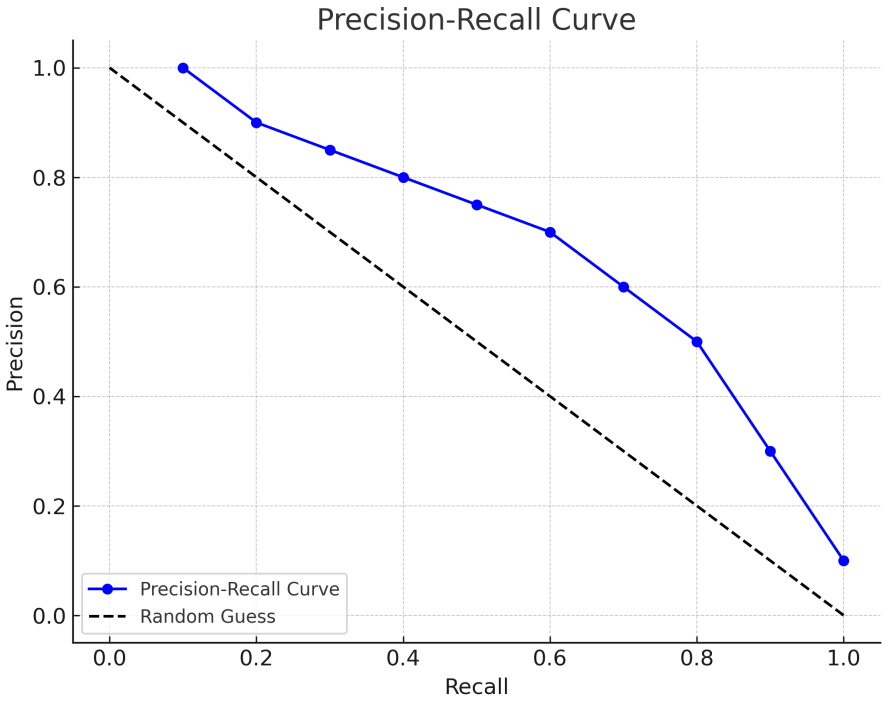



Figure 3.2: Precision-Recall Curve

# Chapter 4

# Data Analysis

The dataset used in this study on optimizing credit risk was obtained from Kaggle. It comprises 32,582 entries and includes the following features:

Table 4.1: Credit Risk Assessment Dataset Variables

| Variable Name | Description | Type |
|---|---|---|
| Age | The age of the borrower. | Numerical |
| Annual Income | The yearly income of the borrower. | Numerical |
| Home Ownership | The type of home ownership (e.g., rent, own, mortgage). | Categorical |
| Employment Length | The duration of employment in years. | Numerical |
| Loan Intent | The purpose of the loan (e.g., personal, debt consolidation). | Categorical |
| Loan Grade | The credit grade assigned to the loan. | Ordinal |
| Loan Amount | The amount of the loan requested. | Numerical |
| Interest Rate | The interest rate on the loan. | Numerical |
| Loan Status | The target variable indicating whether the loan is a non-default or a default. | Binary |
| Loan Percent to Income | The percentage of the loan amount relative to the borrower's income. | Numerical |
| Historical Defaults | The number of historical defaults by the borrower. | Numerical |
| Credit History Length | The duration of the borrower's credit history. | Numerical |

Prior to analysis, the dataset underwent comprehensive pre-processing to ensure its suitability for predictive modeling. The specific steps included:

## 4.1 Data Pre-processing

To prepare the data for analysis, categorical variables were encoded numerically using one-hot and label encoding, ensuring compatibility with machine learning models. Missing values were imputed: numerical features with the median and categorical features with the mode. Numerical features were rescaled using the robust scaler, which is robust to outliers and ensures uniformity in feature impact. The dataset was split into training and test sets (80:20). Regularized logistic regression was used for feature selection, reducing dimensionality by penalizing

less informative features and identifying relevant predictors. The dataset exhibited significant class imbalance in the target variable *Loan Status*, which was addressed using various ensemble resampling techniques on the training data:

- **SMOTE and Random Undersampling:** Synthetic Minority Oversampling Technique (SMOTE) was used to generate synthetic samples for the minority class, while random undersampling reduced the majority class to balance the dataset.
- **ADASYN and Random Undersampling:** Adaptive Synthetic Sampling (ADASYN) was applied to create synthetic samples focused on harder-to-classify instances, coupled with random undersampling of the majority class.
- **Borderline-SMOTE and Random Undersampling:** Borderline-SMOTE generated synthetic samples along the decision boundary of the minority class to improve separability, combined with random undersampling.
- **SVM SMOTE and Random Undersampling:** Support Vector Machine (SVM)-based SMOTE created synthetic samples guided by SVM decision boundaries, complemented with random undersampling.
- **SMOTE-TOMEK:** This combined approach utilized SMOTE for oversampling and Tomek Links to remove overlapping samples between classes.

Gridsearch was used to identify the optimal combination for the sampling strategies. The most effective combination for improving the performance of the credit risk models, was selected for each ensemble sampling strategy.

Finally, the processed and balanced datasets were utilized to train and evaluate hybrid machine learning models. These models combined the strengths of ensemble techniques and advanced sampling strategies to achieve robust predictions. The prediction process involved testing various models and assessing their performance using standard metrics such as accuracy, precision, recall, and the F1-score.

## 4.2 Feature Analysis

Table 4.2 provides a summary of the features used in analysis. The analysis of default rates across various borrower characteristics provided the following key insights into credit risk factors. Age significantly impacts default probability, with borrowers aged 60–70 showing the highest rate (33.85%), while those 70–80 have lower rates (16.67%). Income levels negatively correlate with default risk: borrowers earning below $100,000 have a 23.92% default rate, while those above $800,000 show 0%. Employment length also influences risk, with less than 10 years showing 22.49% default rate versus 0% for over 40 years. We observe that the loan characteristics are crucial:

- Loans between $20,000-$40,000 have the highest default rate (33.68%),
- Interest rates above 20% show extreme default rates (85.14%),
- Loan-to-income ratios over 50% correlate with very high default rates (76.74%-80%).

Other significant factors include:

- Credit history: Longer history correlates with lower default rates,
- Homeownership: Renters (31.57%) and "Other" (30.84%) have higher rates than homeowners (7.47% for full owners),
- Loan purpose: Debt consolidation (28.59%) has higher default rates than education (17.22%) or venture loans (14.81%),
- Loan grade: Default rates increase progressively from Grade A (9.96%) to Grade G (98.44%),
- Historical defaults: Previous defaulters have significantly higher default rates (37.81% vs. 18.39% for no prior defaults).

| Feature | Subcategory | Percentage Default | Percentage Non-Default |
|---|---|---|---|
| Age | 18 <30 | 22.35% | 77.65% |
| | 30 <40 | 20.24% | 79.76% |
| | 40 <50 | 20.42% | 79.58% |
| | 50 <60 | 23.92% | 76.08% |
| | 60 <70 | 33.85% | 66.15% |
| | 70 <80 | 16.67% | 83.33% |
| Annual Income | 0 <100000 | 23.92% | 76.08% |
| | 100000 <200000 | 8.78% | 91.22% |
| | 200000 <300000 | 10.78% | 89.22% |
| | 300000 <400000 | 6.02% | 93.98% |
| | 400000 <500000 | 11.11% | 88.89% |
| | 500000 <600000 | 12.50% | 87.50% |
| | 600000 <700000 | 27.27% | 72.73% |
| | 700000 <800000 | 7.69% | 92.31% |
| | 800000 <900000 | 0.00% | 100.00% |
| | 900000 <1000000 | 0.00% | 100.00% |
| | 1000000 <3000000 | 0.00% | 100.00% |
| Employment Length | 0 <10 | 22.49% | 77.51% |
| | 10 <20 | 16.05% | 83.95% |
| | 20 <30 | 24.65% | 75.35% |
| | 30 <40 | 25.00% | 75.00% |
| | 40 <50 | 0.00% | 100.00% |
| Loan Amount | 0 <5000 | 20.78% | 79.22% |
| | 5000 <10000 | 17.96% | 82.04% |
| | 10000 <20000 | 23.17% | 76.83% |
| | 20000 <40000 | 33.68% | 66.32% |
| Interest Rate | 0 <10 | 10.80% | 89.20% |
| | 10 <15 | 21.93% | 78.07% |
| | 15 <20 | 57.41% | 42.59% |
| | 20 <30 | 85.14% | 14.86% |
| Loan Percent to Income | 0.0 <0.25 | 14.28% | 85.72% |
| | 0.25 <0.5 | 47.95% | 52.05% |
| | 0.5 <0.75 | 76.74% | 23.26% |
| | 0.75 <1.0 | 80.00% | 20.00% |
| Credit History Length | 0 <5 | 22.72% | 77.28% |
| | 5 <10 | 20.71% | 79.29% |
| | 10 <20 | 20.48% | 79.52% |
| | 20 <30 | 26.58% | 73.42% |
| Home Ownership | MORTGAGE | 12.57% | 87.43% |
| | OTHER | 30.84% | 69.16% |
| | OWN | 7.47% | 92.53% |
| | RENT | 31.57% | 68.43% |
| Loan Intent | DEBTCONSOLIDATION | 28.59% | 71.41% |
| | EDUCATION | 17.22% | 82.78% |
| | HOMEIMPROVEMENT | 26.10% | 73.90% |
| | MEDICAL | 26.70% | 73.30% |
| | PERSONAL | 19.89% | 80.11% |
| | VENTURE | 14.81% | 85.19% |
| Loan Grade | A | 9.96% | 90.04% |
| | B | 16.28% | 83.72% |
| | C | 20.73% | 79.27% |
| | D | 59.05% | 40.95% |
| | E | 64.42% | 35.58% |
| | F | 70.54% | 29.46% |
| | G | 98.44% | 1.56% |
| Historical Defaults | N | 18.39% | 81.61% |
| | Y | 37.81% | 62.19% |

Table 4.2: Feature Analysis Table

## 4.3 Model Analysis

### 4.3.1 SMOTE

The initial evaluation of credit risk modeling was conducted using SMOTE as the primary sampling technique to address class imbalance. The results presented in Table 3 indicate that hybrid models generally outperformed individual classifiers in terms of precision, though recall values varied. The Random Forest model exhibited the highest standalone performance, achieving a precision of 0.8736, accuracy of 0.9142, and an AUC of 0.9226, suggesting its robustness in distinguishing high-risk borrowers from low-risk ones.

| Model | Precision | Recall | F1 Score | Accuracy | AUC |
|---|---|---|---|---|---|
| SVM | 0.4955 | 0.7707 | 0.6032 | 0.7787 | 0.8559 |
| Decision Tree | 0.7710 | 0.6913 | 0.7290 | 0.8878 | 0.8788 |
| KNN | 0.5215 | 0.7328 | 0.6094 | 0.7950 | 0.8337 |
| Random Forest | 0.8736 | 0.7096 | 0.7831 | 0.9142 | 0.9226 |
| GBDT | 0.7713 | 0.7208 | 0.7452 | 0.8924 | 0.9131 |
| SVM + Decision Tree | 0.8085 | 0.6589 | 0.7261 | 0.9131 | 0.9131 |
| SVM + KNN | 0.6616 | 0.6406 | 0.6509 | 0.8501 | 0.9131 |
| SVM + Random Forest | 0.8807 | 0.6385 | 0.7403 | 0.9023 | 0.9131 |
| SVM + GBDT | 0.7816 | 0.6617 | 0.7167 | 0.8858 | 0.9131 |
| Decision Tree + KNN | 0.8115 | 0.6449 | 0.7187 | 0.8898 | 0.9131 |
| Decision Tree + Random Forest | 0.9408 | 0.6821 | 0.7909 | 0.9213 | 0.9131 |
| Decision Tree + GBDT | 0.9260 | 0.6779 | 0.7828 | 0.9179 | 0.9131 |
| KNN + Random Forest | 0.8830 | 0.6371 | 0.7402 | 0.9024 | 0.9131 |
| KNN + GBDT | 0.8386 | 0.6287 | 0.7186 | 0.8926 | 0.9131 |
| GBDT + Random Forest | 0.9194 | 0.7096 | 0.7827 | 0.9174 | 0.9131 |

Table 4.3: Performance Metrics of Different Models and Their Combinations using SMOTE

The study highlights the effectiveness of hybrid machine learning models combined with SMOTE in enhancing credit risk assessment. Key findings reveal that Random Forest (RF) emerged as the top-performing individual model, achieving precision of 0.8736, accuracy of 0.9142, and an AUC of 0.9226, underscoring its robustness in handling imbalanced datasets. Gradient Boosting Decision Trees (GBDT) also demonstrated strong performance, with an AUC of 0.9131 and accuracy of 0.8924, leveraging its iterative error-minimization approach to address complex, non-linear patterns inherent in credit risk data. Conversely, Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) exhibited lower precision (0.4955 and 0.5215, respectively), indicating challenges in managing imbalanced data even with SMOTE. Notably, SVM's recall of 0.7707 suggests it may fail to identify a substantial portion of high-risk cases—a critical drawback in credit risk evaluation. The hybrid models, which integrate multiple algorithms, consistently outperformed individual models, emphasizing the value of combining diverse methodologies to harness complementary strengths. These results underscore the importance of ensemble techniques and data balancing strategies like SMOTE in improving the accuracy and reliability of credit risk classification, particularly in identifying high-risk borrowers despite dataset imbalances. For instance:

- **Decision Tree + Random Forest** achieved the highest precision (0.9408) and accuracy

(0.9213) among all models. This combination likely benefits from the interpretability of decision trees and the robustness of random forests.

- **Decision Tree + GBDT** also performed exceptionally well, with a precision of 0.9260 and an AUC of 0.9131. This hybrid model leverages the iterative boosting of GBDT and the simplicity of decision trees, making it highly effective for credit risk prediction.

These results suggest that hybrid models can capture a wider range of patterns in the data, improving both predictive power and reliability. The high AUC values (ranging from 0.8337 to 0.9226) across all models indicate strong discriminatory power, which is crucial for credit risk assessment. A high AUC ensures that the models can effectively distinguish between low-risk and high-risk borrowers, reducing the likelihood of financial losses for lenders. The superior performance of hybrid models, particularly **Decision Tree + Random Forest** and **Decision Tree + GBDT**, suggests that these approaches should be prioritized in real-world credit risk applications. Their ability to achieve high precision and recall simultaneously makes them ideal for minimizing both false positives (approving high-risk borrowers) and false negatives (rejecting low-risk borrowers). The lower precision of SVM and KNN highlights the importance of model selection in credit risk assessment.

### 4.3.2 SMOTE + Random Undersampling

In this study, ensemble sampling, which combines SMOTE and random undersampling, was employed to address class imbalance in credit risk modeling. The optimal sampling strategy was determined using grid search. The results indicate that ensemble sampling provided improved precision across multiple models while maintaining competitive recall and F1 scores. On the performance of individual models, the Random Forest classifier demonstrated the

| Model | Precision | Recall | F1 Score | Accuracy | AUC |
|---|---|---|---|---|---|
| SVM | 0.6372 | 0.6238 | 0.6304 | 0.8404 | 0.8548 |
| Decision Tree | 0.7930 | 0.6653 | 0.7235 | 0.8891 | 0.8766 |
| KNN | 0.6677 | 0.6315 | 0.6491 | 0.8510 | 0.8434 |
| Random Forest | 0.8609 | 0.7138 | 0.7805 | 0.9124 | 0.9243 |
| GBDT | 0.8460 | 0.7032 | 0.7680 | 0.9073 | 0.9200 |
| SVM + Decision Tree | 0.8537 | 0.5703 | 0.6838 | 0.8849 | 0.9200 |
| SVM + KNN | 0.8094 | 0.5077 | 0.6240 | 0.8665 | 0.9200 |
| SVM + Random Forest | 0.8880 | 0.5520 | 0.6808 | 0.8871 | 0.9200 |
| SVM + GBDT | 0.8743 | 0.5577 | 0.6810 | 0.8860 | 0.9200 |
| Decision Tree + KNN | 0.8611 | 0.5928 | 0.7022 | 0.8903 | 0.9200 |
| Decision Tree + Random Forest | 0.8611 | 0.5928 | 0.7022 | 0.8903 | 0.9200 |
| Decision Tree + GBDT | 0.9310 | 0.6737 | 0.7817 | 0.9179 | 0.9200 |
| KNN + Random Forest | 0.8911 | 0.5872 | 0.7079 | 0.8943 | 0.9200 |
| KNN + GBDT | 0.9072 | 0.5703 | 0.7003 | 0.8935 | 0.9200 |
| GBDT + Random Forest | 0.9114 | 0.6730 | 0.7743 | 0.9144 | 0.9200 |

Table 4.4: Performance Metrics of Different Models and Their Combinations using SMOTE and Random Undersampling

highest overall performance, achieving an accuracy of 91.24%, an F1 score of 0.7805, and the highest AUC of 0.9243. This aligns with expectations, as ensemble-based models, particularly Random Forest, tend to perform well in imbalanced classification tasks by leveraging multiple

decision trees to improve generalization. Similarly, GBDT (Gradient Boosting Decision Trees) exhibited strong performance, with an accuracy of 90.73% and an F1 score of 0.7680, making it another robust choice for credit risk classification. Among the base classifiers, SVM and KNN showed moderate performance, with lower recall values compared to tree-based models. The Decision Tree classifier achieved an F1 score of 0.7235, indicating better generalization than SVM and KNN while maintaining a competitive recall.

On the performance of hybrid models, the combination of Decision Tree + GBDT emerged as the most effective hybrid approach, with the highest F1 score (0.7817) among all tested models, alongside an accuracy of 91.79%. This indicates that blending an interpretable model (Decision Tree) with a powerful boosting technique (GBDT) enhanced predictive performance by leveraging both diversity and learning depth. Interestingly, several hybrid models, including SVM + Decision Tree, SVM + Random Forest, and SVM + GBDT, exhibited lower recall values, suggesting that while ensemble sampling improved precision, it led to a trade-off in capturing positive instances. This may indicate that SVM-based combinations were more conservative in classification, leading to fewer false positives but a higher rate of false negatives. The study also found that ensemble sampling techniques improved precision across multiple credit risk assessment models compared to SMOTE-only approaches, particularly in hybrid models. This precision improvement suggests models became more confident in their predictions, reducing misclassifications of negative instances (default risk). However, this came with a slight reduction in recall for some models. Despite this trade-off, hybrid models maintained consistently high AUC values of approximately 0.9200, indicating that ensemble sampling preserved the classifiers' ability to effectively distinguish between low-risk and high-risk borrowers. These findings suggest that ensemble sampling methods represent a valuable advancement in credit risk assessment, offering a better balance of predictive confidence and discriminatory power compared to traditional SMOTE approaches.

### 4.3.3 ADASYN + Random Undersampling

Among the individual classifiers, Random Forest achieved the highest performance, with an accuracy of 90.81%, F1 score of 0.7739, and AUC of 0.9218, reinforcing its strength as an ensemble learning method. Gradient Boosting Decision Trees (GBDT) followed closely, demonstrating robust classification ability with an accuracy of 89.24% and an F1 score of 0.7450. Decision Trees also performed competitively, achieving a recall of 72.01%, which is crucial for identifying high-risk borrowers. Meanwhile, SVM and KNN showed lower precision and F1 scores compared to tree-based models, which aligns with expectations given their sensitivity to data imbalance. The hybrid models exhibited diverse performance patterns. The Decision Tree + GBDT combination emerged as the most effective, achieving an F1 score of 0.7617 and an accuracy of 90.92%, surpassing most individual classifiers. Similarly, GBDT + Random Forest maintained a strong balance between recall and precision, achieving an F1 score of 0.7684 and AUC of 0.9182. The comparison between SMOTE and ADASYN-based ensemble sampling indicates that both approaches effectively handle class imbalance but with different trade-offs. SMOTE improves recall, making it preferable when the goal is to minimize false negatives (e.g., correctly identifying high-risk borrowers). On the other hand, ADASYN + undersampling improves precision, which may be beneficial in reducing false positives and avoiding unnecessary loan rejections.

| Model | Precision | Recall | F1 Score | Accuracy | AUC |
|---|---|---|---|---|---|
| SVM | 0.5351 | 0.7342 | 0.6190 | 0.8028 | 0.8557 |
| Decision Tree | 0.6896 | 0.7201 | 0.7045 | 0.8682 | 0.8835 |
| KNN | 0.5089 | 0.7243 | 0.5978 | 0.7873 | 0.8299 |
| Random Forest | 0.8354 | 0.7208 | 0.7739 | 0.9081 | 0.9218 |
| GBDT | 0.7717 | 0.7201 | 0.7450 | 0.8924 | 0.9182 |
| SVM + Decision Tree | 0.7825 | 0.6273 | 0.6963 | 0.8806 | 0.9182 |
| SVM + KNN | 0.6810 | 0.6125 | 0.6449 | 0.8528 | 0.9182 |
| SVM + Random Forest | 0.8525 | 0.6259 | 0.7218 | 0.8947 | 0.9182 |
| SVM + GBDT | 0.7804 | 0.6371 | 0.7015 | 0.8817 | 0.9182 |
| Decision Tree + KNN | 0.7701 | 0.6456 | 0.7024 | 0.8806 | 0.9182 |
| Decision Tree + Random Forest | 0.7701 | 0.6456 | 0.7024 | 0.8806 | 0.9182 |
| Decision Tree + GBDT | 0.8908 | 0.6653 | 0.7617 | 0.9092 | 0.9182 |
| KNN + Random Forest | 0.8574 | 0.6428 | 0.7347 | 0.8987 | 0.9182 |
| KNN + GBDT | 0.8344 | 0.6238 | 0.7139 | 0.8909 | 0.9182 |
| GBDT + Random Forest | 0.8809 | 0.6814 | 0.7684 | 0.9104 | 0.9182 |

Table 4.5: Performance Metrics of Different Models and Their Combinations using ADASYN and Random Undersampling

| Model | Precision | Recall | F1 Score | Accuracy | AUC |
|---|---|---|---|---|---|
| SVM | 0.5337 | 0.7342 | 0.6181 | 0.8021 | 0.8557 |
| Decision Tree | 0.6960 | 0.7166 | 0.7062 | 0.8699 | 0.8836 |
| KNN | 0.5322 | 0.7152 | 0.6103 | 0.8007 | 0.8331 |
| Random Forest | 0.8324 | 0.7264 | 0.7758 | 0.9084 | 0.9224 |
| GBDT | 0.7698 | 0.7243 | 0.7464 | 0.8926 | 0.9172 |
| SVM + Decision Tree | 0.7709 | 0.6364 | 0.6972 | 0.8794 | 0.9172 |
| SVM + KNN | 0.6805 | 0.6125 | 0.6447 | 0.8527 | 0.9172 |
| SVM + Random Forest | 0.8529 | 0.6280 | 0.7234 | 0.8952 | 0.9172 |
| SVM + GBDT | 0.7803 | 0.6470 | 0.7074 | 0.8832 | 0.9172 |
| Decision Tree + KNN | 0.7772 | 0.6329 | 0.6977 | 0.8803 | 0.9172 |
| Decision Tree + Random Forest | 0.8919 | 0.6906 | 0.7784 | 0.9142 | 0.9172 |
| Decision Tree + GBDT | 0.8893 | 0.6723 | 0.7657 | 0.9102 | 0.9172 |
| KNN + Random Forest | 0.8468 | 0.6414 | 0.7299 | 0.8964 | 0.9172 |
| KNN + GBDT | 0.8375 | 0.6308 | 0.7196 | 0.8927 | 0.9172 |
| GBDT + Random Forest | 0.8919 | 0.6850 | 0.7749 | 0.9132 | 0.9172 |

Table 4.6: Performance Metrics of Different Models and Their Combinations using Borderline-SMOTE and Random Undersampling

### 4.3.4 Borderline-SMOTE + Random Undersampling

On performance of individual models, the results indicate that Random Forest consistently outperforms other models across all sampling strategies, achieving a high F1 Score (0.7758), accuracy (0.9084), and AUC (0.9224) when using Borderline-SMOTE + Random Undersampling. Gradient Boosting Decision Trees (GBDT) also demonstrated competitive performance, with an F1 Score of 0.7464 and an AUC of 0.9172, making it one of the strongest individual classifiers.

Compared to SMOTE alone, which produced lower recall but higher precision, the use of Borderline-SMOTE enhanced recall values, ensuring that more default cases were correctly identified. Among hybrid models, Decision Tree + Random Forest and GBDT + Random Forest stood out, achieving the highest F1 Scores (0.7784 and 0.7749, respectively) and AUC values (0.9172 and 0.9172, respectively). This suggests that combining decision trees with ensemble methods yields robust models capable of handling class imbalance effectively.
Compared to ADASYN + Random Undersampling, which slightly increased recall but sacrificed precision, Borderline-SMOTE + Random Undersampling balanced the trade-off between precision and recall more effectively.

### 4.3.5 SVM-SMOTE + Random Undersampling

| Model | Precision | Recall | F1 Score | Accuracy | AUC |
|---|---|---|---|---|---|
| SVM | 0.5463 | 0.7300 | 0.6249 | 0.8088 | 0.8551 |
| Decision Tree | 0.7500 | 0.7194 | 0.7344 | 0.8865 | 0.8813 |
| KNN | 0.5768 | 0.6814 | 0.6248 | 0.8214 | 0.8393 |
| Random Forest | 0.8551 | 0.7222 | 0.7831 | 0.9127 | 0.9232 |
| GBDT | 0.7759 | 0.7159 | 0.7447 | 0.8929 | 0.9157 |
| SVM + Decision Tree | 0.7979 | 0.6385 | 0.7094 | 0.8858 | 0.9157 |
| SVM + KNN | 0.7150 | 0.5928 | 0.6482 | 0.8596 | 0.9157 |
| SVM + Random Forest | 0.8717 | 0.6210 | 0.7253 | 0.8973 | 0.9157 |
| SVM + GBDT | 0.7920 | 0.6399 | 0.7079 | 0.8848 | 0.9157 |
| Decision Tree + KNN | 0.8199 | 0.6245 | 0.7090 | 0.8881 | 0.9157 |
| Decision Tree + Random Forest | 0.9191 | 0.6955 | 0.7918 | 0.9202 | 0.9157 |
| Decision Tree + GBDT | 0.8983 | 0.6772 | 0.7723 | 0.9128 | 0.9157 |
| KNN + Random Forest | 0.8770 | 0.6217 | 0.7276 | 0.8984 | 0.9157 |
| KNN + GBDT | 0.8508 | 0.6097 | 0.7104 | 0.8915 | 0.9157 |
| GBDT + Random Forest | 0.8970 | 0.6800 | 0.7736 | 0.9132 | 0.9157 |

Table 4.7: Performance Metrics of Different Models and Their Combinations using SVM-SMOTE and Random Undersampling

Random Undersampling demonstrated a notable improvement in classification effectiveness. Specifically, the Random Forest classifier achieved the highest F1-score of 0.7831, accuracy of 91.27%, and an AUC of 0.9232, indicating strong discrimination ability between default and non-default cases. Gradient Boosting Decision Trees (GBDT) also showed competitive results, with an F1-score of 0.7447 and an accuracy of 89.29%, reflecting its robustness in handling imbalanced data.
When examining hybrid models, the combination of Decision Tree and Random Forest performed exceptionally well, achieving an F1-score of 0.7918 and an accuracy of 92.02%, surpassing individual model performances. Similarly, the combination of Decision Tree and GBDT yielded promising results with an F1-score of 0.7723 and an accuracy of 91.28%.

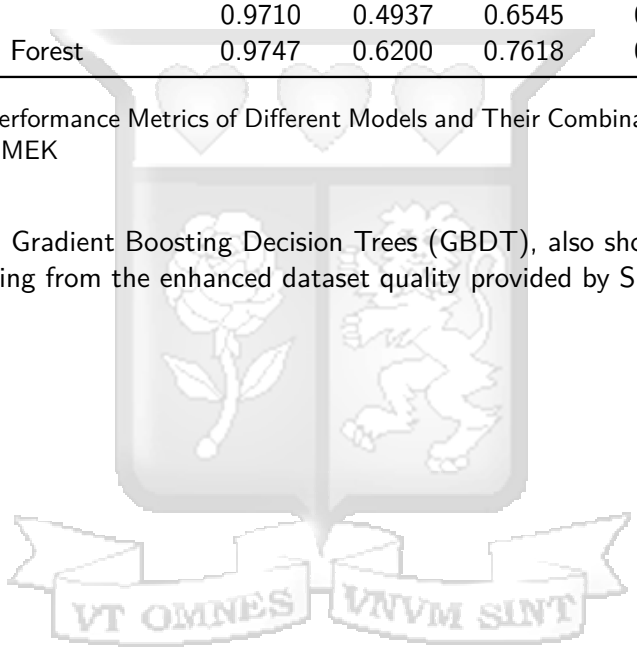
### 4.3.6 SMOTE-TOMEK

The Random Forest classifier achieves the highest F1-score and AUC, highlighting its ability to balance false positives and false negatives effectively. Additionally, the combination of Decision Tree and Random Forest also performs exceptionally well, demonstrating enhanced model robustness and decision boundary refinement. Hybrid models, particularly the ensemble

| Model | Precision | Recall | F1 Score | Accuracy | AUC |
|---|---|---|---|---|---|
| SVM | 0.7578 | 0.4599 | 0.5724 | 0.8501 | 0.8521 |
| Decision Tree | 0.9522 | 0.5605 | 0.7056 | 0.8980 | 0.8678 |
| KNN | 0.8021 | 0.5331 | 0.6405 | 0.8694 | 0.8295 |
| Random Forest | 0.9518 | 0.6660 | 0.7836 | 0.9197 | 0.9200 |
| GBDT | 0.9414 | 0.6435 | 0.7644 | 0.9135 | 0.9163 |
| SVM + Decision Tree | 0.9444 | 0.4177 | 0.5792 | 0.8676 | 0.9163 |
| SVM + KNN | 0.9138 | 0.3727 | 0.5295 | 0.8555 | 0.9163 |
| SVM + Random Forest | 0.9688 | 0.4149 | 0.5810 | 0.8694 | 0.9163 |
| SVM + GBDT | 0.9548 | 0.4163 | 0.5798 | 0.8683 | 0.9163 |
| Decision Tree + KNN | 0.9363 | 0.5063 | 0.6572 | 0.8848 | 0.9163 |
| Decision Tree + Random Forest | 0.9766 | 0.6449 | 0.7768 | 0.9191 | 0.9163 |
| Decision Tree + GBDT | 0.9821 | 0.6181 | 0.7587 | 0.9142 | 0.9163 |
| KNN + Random Forest | 0.9625 | 0.5056 | 0.6630 | 0.8878 | 0.9163 |
| KNN + GBDT | 0.9710 | 0.4937 | 0.6545 | 0.8867 | 0.9163 |
| GBDT + Random Forest | 0.9747 | 0.6200 | 0.7618 | 0.9147 | 0.9163 |

Table 4.8: Performance Metrics of Different Models and Their Combinations using SMOTE-TOMEK

of Decision Tree and Gradient Boosting Decision Trees (GBDT), also show strong predictive performance, benefiting from the enhanced dataset quality provided by SMOTE-TOMEK.



# Chapter 5

# Discussion and Findings

This chapter presents the key discoveries of this study and indepth comparisons between sampling strategies.

Below is a summary of the key findings:

- **Random Forest** consistently outperformed other individual models across all sampling strategies, achieving the highest AUC and a good balance of precision and recall.
- **SMOTE + Random Undersampling** achieved the highest AUC (0.9243) and provided a robust balance between precision and recall.
- **Borderline SMOTE + Undersampling** achieved the highest recall (0.7264) using Random Forest.
- **SMOTE-TOMEK** achieved the highest precision (0.9766) using Decision Tree + Random Forest.
- **Decision Tree + Random Forest** was the best-performing hybrid model, achieving high precision and F1 score across most sampling strategies.
- **SVM** and **KNN** performed poorly across all sampling strategies, with low precision and recall.

Table 5.1, shows a comparison across the sampling techniques using the Random Forest model, which emerged as the best performing individual model.

| Sampling Strategy | Best Model | Precision | Recall | F1 Score | AUC |
|---|---|---|---|---|---|
| SMOTE | Random Forest | 0.8736 | 0.7096 | 0.7831 | 0.9226 |
| SMOTE + Undersampling | Random Forest | 0.8609 | 0.7138 | 0.7805 | 0.9243 |
| ADASYN + Undersampling | Random Forest | 0.8354 | 0.7208 | 0.7739 | 0.9218 |
| Borderline SMOTE + Undersampling | Random Forest | 0.8324 | 0.7264 | 0.7758 | 0.9224 |
| SVM SMOTE + Undersampling | Random Forest | 0.8551 | 0.7222 | 0.7831 | 0.9232 |
| SMOTE-TOMEK | Random Forest | 0.9518 | 0.6660 | 0.7836 | 0.9200 |

Table 5.1: Performance of Individual Models

Table 5.2 shows a comparison across sampling techniques using the best hybrid model, Decision Tree + Random Forest.

| Sampling Strategy | Best Hybrid Model | Precision | Recall | F1 Score | AUC |
|---|---|---|---|---|---|
| SMOTE | Decision Tree + Random Forest | 0.9408 | 0.6821 | 0.7909 | 0.9131 |
| SMOTE + Undersampling | Decision Tree + Random Forest | 0.9408 | 0.6821 | 0.7909 | 0.9131 |
| ADASYN + Undersampling | Decision Tree + Random Forest | 0.8611 | 0.5928 | 0.7022 | 0.9182 |
| Borderline SMOTE + Undersampling | Decision Tree + Random Forest | 0.8919 | 0.6906 | 0.7784 | 0.9172 |
| SVM SMOTE + Undersampling | Decision Tree + Random Forest | 0.9191 | 0.6955 | 0.7918 | 0.9157 |
| SMOTE-TOMEK | Decision Tree + Random Forest | 0.9766 | 0.6449 | 0.7768 | 0.9163 |

Table 5.2: Performance of Hybrid Models

Figure 5.1 shows the Precision-Recall Curve for the SMOTE + Random Undersampling sampling ensemble with Random Forests machine learning algorithm.

The Precision - Recall Area Under the Curve score of 0.87, indicates that it is a strong model, and it performs well in distinguishing between positive and negative classes.

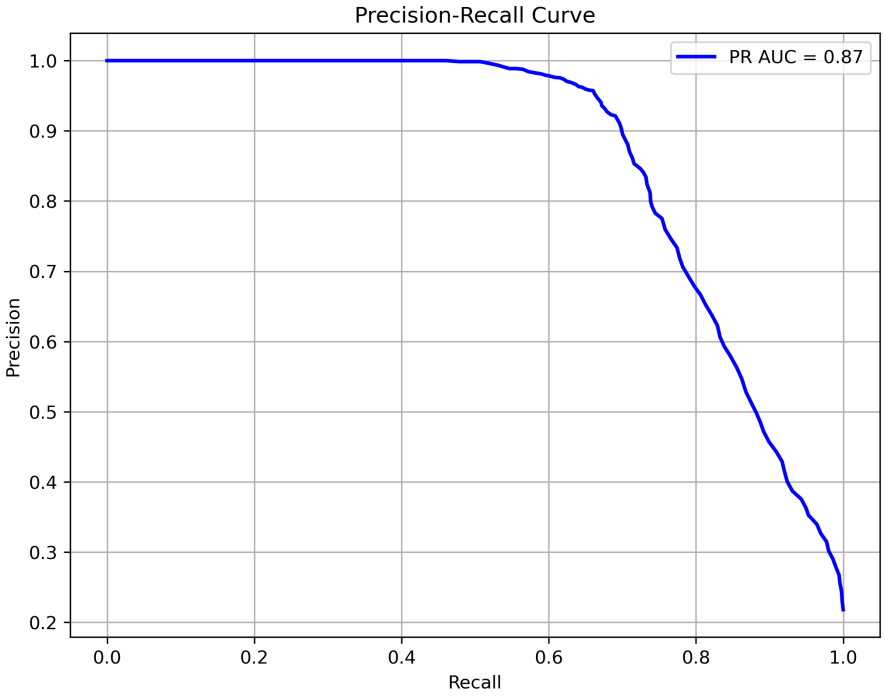



Figure 5.1: SMOTE + Random Undersampling Precision-recall curve

Figure 5.2 shows the Precision-Recall Curve for the SMOTE + Random Undersampling sampling ensemble with Random Forests machine learning algorithm.The Precision - Recall Area Under the Curve score of 0.86, also indicates that it is a strong model, and it performs well in distinguishing between positive and negative classes.

The findings show that the two models have almost similar Precision-Recall Area Under the Curve, but SMOTE + Random Undersampling + Random Forests has a significantly higher precision than Borderline SMOTE + Random Undersampling + Random Forests. Also, the latter achieves a significantly higher recall than SMOTE + Random Undersampling + Random Forests.

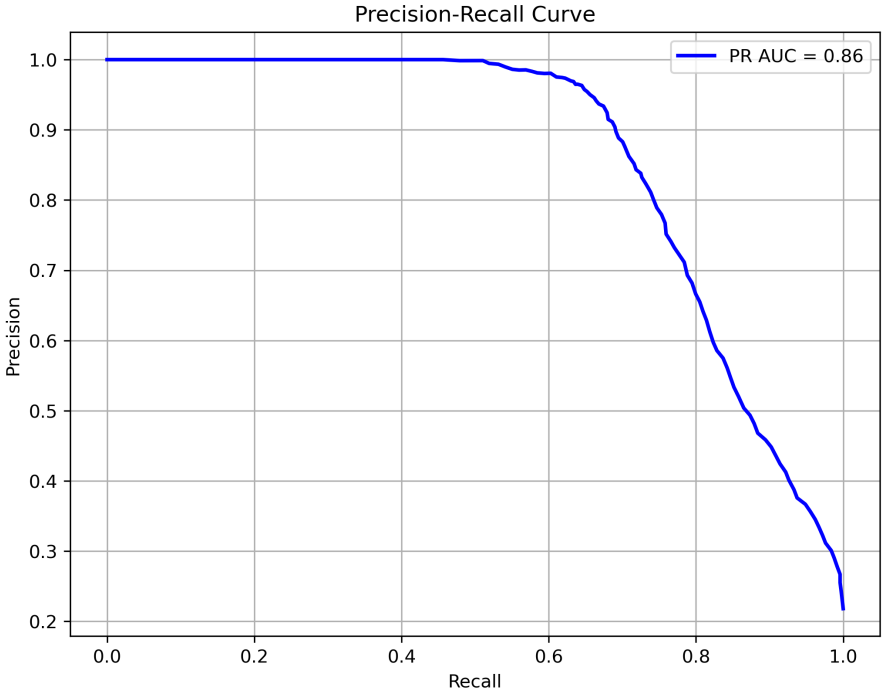



Figure 5.2: Borderline SMOTE + Random Undersampling Precision-recall curve

# Chapter 6

# Conclusions and Future Work

In this study, we conducted a comprehensive comparison of six sampling techniques. SMOTE, SMOTE + Random Undersampling, ADASYN + Random Undersampling, Borderline SMOTE + Random Undersampling, SVM SMOTE + Random Undersampling, and SMOTE-TOMEK for addressing class imbalance in credit risk modeling. The results indicate that individual models consistently exhibit higher recall compared to hybrid models, suggesting their effectiveness as aggressive default detectors. Conversely, hybrid models demonstrate significantly higher precision, likely due to their complex decision boundaries. Furthermore, the use of ensemble sampling, combining SMOTE variants with random undersampling, yielded improved results compared to the individual SMOTE sampling, demonstrating the effectiveness of integrating oversampling and undersampling techniques to enhance model performance. Given that the primary objective of this study is to enhance default detection, the findings strongly support the use of individual models, particularly the **Borderline SMOTE + Random Undersampling** with **Random Forest**, which achieved the highest recall, and the **SMOTE + Random Undersampling** with **Random Forest**, which demonstrated superior AUC performance. For applications where a balance between recall and precision is required, adjusting the decision threshold presents a viable approach. Additionally, alternative ensemble strategies, such as weighted voting within hybrid models, may further enhance predictive performance. Future research could explore advanced sampling techniques, such as GAN-based oversampling, and the application of deep learning models to further enhance predictive accuracy in imbalanced datasets. This study underscores the importance of selecting appropriate sampling strategies and models to build reliable and effective credit risk models.

# References


Altman, E. I. (1968), 'Financial ratios, discriminant analysis and the prediction of corporate bankruptcy', *The journal of finance* **23**(4), 589–609.

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J. and Vanthienen, J. (2003), 'Benchmarking state-of-the-art classification algorithms for credit scoring', *Journal of the operational research society* **54**, 627–635.

Basel Committee on Banking Supervision (1988), International convergence of capital measurement and capital standards, Technical report, Bank for International Settlements. Basel I Accord.
**URL:** *https://www.bis.org/publ/bcbs04a.htm*

Basel Committee on Banking Supervision (2004), International convergence of capital measurement and capital standards: A revised framework, Technical report, Bank for International Settlements. Basel II Accord.
**URL:** *https://www.bis.org/publ/bcbs107.htm*

Basel Committee on Banking Supervision (2011), Basel iii: A global regulatory framework for more resilient banks and banking systems, Technical report, Bank for International Settlements. Basel III Accord.
**URL:** *https://www.bis.org/publ/bcbs189.htm*

Beaver, W. H. (1966), 'Financial ratios as predictors of failure', *Journal of accounting research* pp. 71–111.

Berger, A. N. and Di Patti, E. B. (2006), 'Capital structure and firm performance: A new approach to testing agency theory and an application to the banking industry', *Journal of Banking & Finance* **30**(4), 1065–1102.

Breiman, L. (2001), 'Random forests', *Machine learning* **45**, 5–32.

Charpignon, M.-L., Horel, E. and Tixier, F. (2014), 'Prediction of consumer credit risk', *Standford University* .

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* **16**, 321–357.

Chen, W.-H., Hsu, S.-H. and Shen, H.-P. (2005), 'Application of svm and ann for intrusion detection', *Computers & Operations Research* **32**(10), 2617–2634.

Choge, J. K. (2012), Credit evaluation model using naïve bayes classifier a case of a Kenyan Commercial Bank, PhD thesis, University of Nairobi.

Costa e Silva, E., Lopes, I. C., Correia, A. and Faria, S. (2020), 'A logistic regression model for consumer default risk', *Journal of Applied Statistics* **47**(13-15), 2879–2894.

Crook, J., Desai, V. and Overstreet, G. (1996), 'A comparison of a neural network and classical techniques for credit scoring', *European Journal of Operational Research* **95**, 24–36.

Durand, D. (1941), Appendix b: Application of the method of discriminant functions to the good-and bad-loan samples, *in* 'Risk Elements in Consumer Instalment Financing, Technical Edition', NBER, pp. 125–142.

Elreedy, D. and Atiya, A. F. (2019), 'A comprehensive analysis of synthetic minority oversampling technique (smote) for handling class imbalance', *Information Sciences* **505**, 32–64.

Elreedy, D., Atiya, A. F. and Kamalov, F. (2023), 'A theoretical distribution analysis of synthetic minority oversampling technique (smote) for imbalanced learning', *Machine Learning* pp. 1–21.

Fisher, R. A. (1936), 'The use of multiple measurements in taxonomic problems', *Annals of eugenics* **7**(2), 179–188.

Frydman, H., Altman, E. I. and Kao, D.-L. (1985), 'Introducing recursive partitioning for financial classification: the case of financial distress', *The journal of finance* **40**(1), 269–291.

Han, H., Wang, W.-Y. and Mao, B.-H. (2005), Borderline-smote: a new over-sampling method in imbalanced data sets learning, *in* 'International conference on intelligent computing', Springer, pp. 878–887.

Hicks, J. R. (1989), A suggestion for simplifying the theory of money, *in* 'General Equilibrium Models of Monetary Economies', Elsevier, pp. 7–23.

Jarrow, R. A., Lando, D. and Turnbull, S. M. (1997), 'A markov model for the term structure of credit risk spreads', *The review of financial studies* **10**(2), 481–523.

Kecman, V. (2001), *Learning and soft computing: support vector machines, neural networks, and fuzzy logic models*, MIT press.

Kubat, M., Holte, R. C. and Matwin, S. (1998), 'Machine learning for the detection of oil spills in satellite radar images', *Machine learning* **30**, 195–215.

Kubat, M., Holte, R. and Matwin, S. (1997), Learning when negative examples abound, *in* 'Machine Learning: ECML-97: 9th European Conference on Machine Learning Prague, Czech Republic, April 23–25, 1997 Proceedings 9', Springer, pp. 146–153.

Malik, M. and Thomas, L. C. (2010), 'Modelling credit risk of portfolio of consumer loans', *Journal of the Operational Research Society* **61**(3), 411–420.

Markowitz, H. (1952), 'The utility of wealth', *Journal of political Economy* **60**(2), 151–158.

Marqués, A. I., García, V. and Sánchez, J. S. (2013), 'On the suitability of resampling techniques for the class imbalance problem in credit scoring', *Journal of the Operational Research Society* **64**, 1060–1070.

Martin, D. (1977), 'Early warning of bank failure: A logit regression approach', *Journal of banking & finance* **1**(3), 249–276.

Merton, R. C. (1973), 'Theory of rational option pricing', *The Bell Journal of economics and management science* pp. 141–183.

Merton, R. C. (1974), 'On the pricing of corporate debt: The risk structure of interest rates', *The Journal of finance* **29**(2), 449–470.

Nanni, L. and Lumini, A. (2009), 'An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring', *Expert systems with applications* **36**(2), 3028–3033.

Perli, R. and Nayda, W. I. (2004), 'Economic and regulatory capital allocation for revolving retail exposures', *Journal of Banking & Finance* **28**(4), 789–809.

Rokach, L. (2010), 'Ensemble-based classifiers', *Artificial intelligence review* **33**, 1–39.

Ross, S. A. (1976), 'The arbitrage theory of capital asset pricing', *Journal of Economic Theory* **13**(3), 341–360.
**URL:** *https://www.sciencedirect.com/science/article/pii/0022053176900466*

Sharpe, W. F. (1964), 'Capital asset prices: A theory of market equilibrium under conditions of risk', *The Journal of Finance* **19**(3), 425–442.

Tsai, C.-F. and Chen, M.-L. (2010), 'Credit rating by hybrid machine learning techniques', *Applied soft computing* **10**(2), 374–380.

Valášková, K., Gavlakova, P. and Dengov, V. (2014), Assessing credit risk by moody's kmv model, *in* '2nd International Conference on Economics and Social Science (ICESS 2014), Information Engineering Research Institute, Advances in Education Research', Vol. 61, pp. 40–44.

Vapnik, V. N. and Chervonenkis, A. Y. (1963), 'A note on one class of perceptrons', *Automation and Remote Control* **25**, 821–837.

Wilson, R. L. and Sharda, R. (1994), 'Bankruptcy prediction using neural networks', *Decision support systems* **11**(5), 545–557.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S. et al. (2008), 'Top 10 algorithms in data mining', *Knowledge and information systems* **14**, 1–37.

# Appendix A

# Proof 1

We first transform the Loss function to a function of log(odds).

$$= -[y\log(p) + (1-y)\log(1-p)], \tag{A.1}$$

$$= -y\log(p) - (1-y)\log(1-p), \tag{A.2}$$

$$= -y\log(p) - \log(1-p) + y\log(1-p), \tag{A.3}$$

$$= -y(\log(p) - \log(1-p)) - \log(1-p), \tag{A.4}$$

$$= -y(\frac{\log(p)}{\log(1-p)}) - \log(1-p), \tag{A.5}$$

$$= -y\log(\frac{p}{1-p}) - \log(1-p), \tag{A.6}$$

Since:

$$\log(odds) = \log(\frac{p}{1-p}), \tag{A.7}$$

$$= -y\log(odds) - \log(1-p), \tag{A.8}$$

Also:

$$p = \frac{e^{\log(odds)}}{1+e^{\log(odds)}}, \tag{A.9}$$

So:

$$\log(1-p) = \log(1 - \frac{e^{\log(odds)}}{1+e^{\log(odds)}}), \tag{A.10}$$

Which is the same as:

$$= \log(\frac{1+e^{\log(odds)}}{1+e^{\log(odds)}} - \frac{e^{\log(odds)}}{1+e^{\log(odds)}}), \tag{A.11}$$

$$= \log(\frac{1}{1+e^{\log(odds)}}, \tag{A.12}$$

$$= \log(1) - \log(1+e^{\log(odds)}), \tag{A.13}$$

Since $\log(1) = 0$:

$$\log(1-p) = -\log(1+e^{\log(odds)}), \tag{A.14}$$

Thus:

$$L = -y\log(odds) - (-\log(1+e^{\log(odds)})). \tag{A.15}$$

# Appendix B

# Proof 2

To solve for the optimal gamma, we can approximate the loss function with a second-order Taylor expansion.
The second-order Taylor expansion of f around $x_k$ is:

$$f(x_k + t) \approxeq f(x_k) + f'(x_k) + \frac{1}{2}f''(x_k)t^2, \tag{B.1}$$

Similarly, we write the second-order Taylor expansion of the loss function around the point $F_{m-1}(x)$.

$$L(y_i, F_{m-1}(x_i) \approxeq L(y_i, F_{m-1}(x_i) + \frac{dL(y_i, F_{m-1}(x_i)}{dF_{m-1}(x_i)}\gamma + \frac{1}{2}\frac{d^2L(y_i, F_{m-1}(x_i)}{dF_{m-1}(xi)^2}\gamma^2, \tag{B.2}$$

We take the derivative of the Loss function with respect to $gamma$.

$$\frac{d}{d\gamma}L(y_i, F_{m-1}(x_i) \approxeq \frac{dL(y_i, F_{m-1}(x_i)}{dF_{m-1}(x_i)} + \frac{d^2L(y_i, F_{m-1}(x_i)}{dF_{m-1}(xi)^2}\gamma, \tag{B.3}$$

We set the derivative equal to 0, and solve for $\gamma$

$$\gamma = -\frac{\frac{dL(y_i, F_{m-1}(x_i)}{dF_{m-1}(x_i)}}{\frac{d^2L(y_i, F_{m-1}(x_i)}{dF_{m-1}(xi)^2}}, \tag{B.4}$$

The numerator of the equation is: $Observed - Predicted(Residual)$. The denominator is the second derivative of the loss function:

$$= \frac{d^2Observed * \log(odds) + log(1 + e^{\log(odds)})}{d\log(odds)^2}, \tag{B.5}$$

$$= \frac{dObserved + \frac{e^{log(odds)}}{1 + e^{\log(odds)}}}{d\log(odds)}, \tag{B.6}$$

$$= \frac{dObserved + (1 + e^{\log(odds)})^{-1} * e^{log(odds)}}{d\log(odds)}, \tag{B.7}$$

Using the product rule, we get:

$$= -(1 + e^{\log(odds)})^{-2}e^{\log(odds)}e^{\log(odds)} + (1 + e^{\log(odds)})^{-1}e^{\log(odds)}, \tag{B.8}$$

$$= \frac{-e^{2log(odds)}}{(1+e^{log(odds)})^2} + \frac{e^{log(odds)}}{(1+e^{\log(odds)})} * \frac{(1+e^{log(odds)})}{(1+e^{log(odds)})}, \tag{B.9}$$

$$= \frac{-e^{2log(odds)}}{(1+e^{log(odds)})^2} + \frac{e^{log(odds)} + e^{2\log(odds)}}{(1+e^{\log(odds)})^2}, \tag{B.10}$$

$$= \frac{-e^{2log(odds)} + e^{log(odds)} + e^{2\log(odds)}}{(1+e^{\log(odds)})^2}, \tag{B.11}$$

$$= \frac{e^{log(odds)}}{(1+e^{\log(odds)})^2}, \tag{B.12}$$

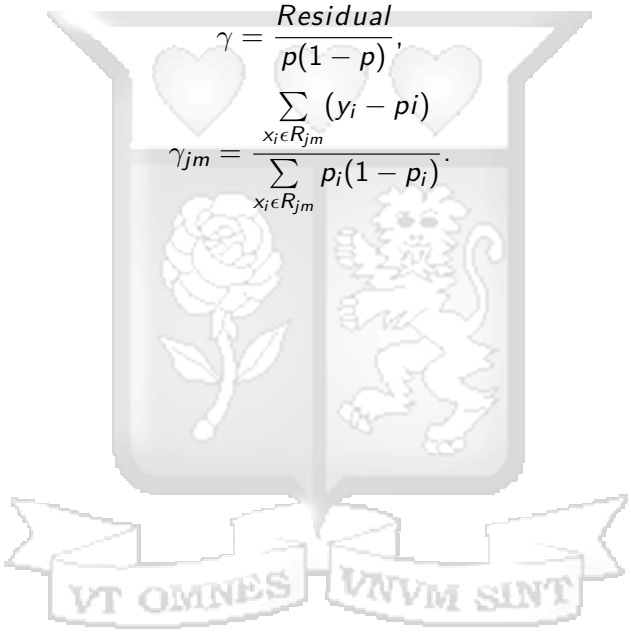
$$= \frac{e^{log(odds)}}{(1+e^{\log(odds)})} * \frac{1}{(1+e^{\log(odds)})}, \tag{B.13}$$

$$= p * (1-p), \tag{B.14}$$

So the optimal $\gamma$ is:

$$\gamma = \frac{Residual}{p(1-p)}, \tag{B.15}$$

$$\gamma_{jm} = \frac{\sum\limits_{x_i \epsilon R_{jm}} (y_i - pi)}{\sum\limits_{x_i \epsilon R_{jm}} p_i(1-p_i)}. \tag{B.16}$$



# Appendix C

# Proof 3

In logistic regression, we model the probability that the outcome $y$ is equal to 1 (i.e., $P(y = 1)$) using the following equation:

$$P(y = 1 \mid \mathbf{x}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)}} \tag{C.1}$$

where:

- $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is the vector of input features.
- $\beta_0, \beta_1, \ldots, \beta_n$ are the model parameters.

The odds of $y = 1$ is given by:

$$\text{Odds}(y = 1 \mid \mathbf{x}) = \frac{P(y = 1 \mid \mathbf{x})}{P(y = 0 \mid \mathbf{x})} \tag{C.2}$$

where $P(y = 0 \mid \mathbf{x}) = 1 - P(y = 1 \mid \mathbf{x})$. Therefore:

$$\text{Odds}(y = 1 \mid \mathbf{x}) = \frac{P(y = 1 \mid \mathbf{x})}{1 - P(y = 1 \mid \mathbf{x})} \tag{C.3}$$

Using the logistic regression formula, we have:

$$P(y = 1 \mid \mathbf{x}) = \frac{1}{1 + e^{-z}} \quad \text{where} \quad z = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n \tag{C.4}$$

$$1 - P(y = 1 \mid \mathbf{x}) = 1 - \frac{1}{1 + e^{-z}} = \frac{e^{-z}}{1 + e^{-z}} \tag{C.5}$$

Thus:

$$\text{Odds}(y = 1 \mid \mathbf{x}) = \frac{\frac{1}{1+e^{-z}}}{\frac{e^{-z}}{1+e^{-z}}} = e^{z} \tag{C.6}$$

The log-odds, also known as the logit function, is the natural logarithm of the odds:

$$\text{Log-Odds} = \text{logit}(P(y = 1 \mid \mathbf{x})) = \log\left(\text{Odds}(y = 1 \mid \mathbf{x})\right) = \log(e^{z}) \tag{C.7}$$

$$\text{Log-Odds} = z = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n \tag{C.8}$$

Therefore, the logistic regression model can be expressed in terms of the log-odds as:

$$\log\left(\frac{P(y=1 \mid \mathbf{x})}{1-P(y=1 \mid \mathbf{x})}\right) = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n \tag{C.9}$$

# Appendix D

# Additional Figures

The figure D.1 shows the sequential steps followed in the data preprocessing phase.

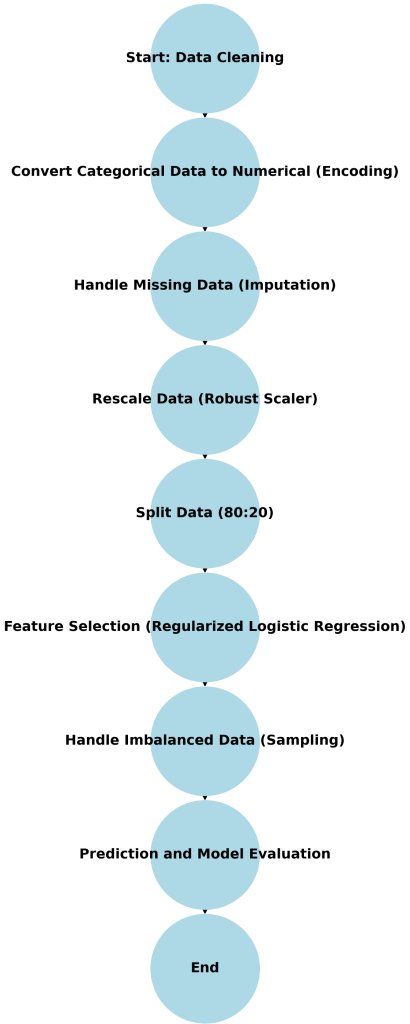



Figure D.1: Flowchart of Data Processing Steps

# Appendix E

# Python code

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from imblearn.over_sampling import SVMSMOTE
from sklearn.neighbors import KNeighborsClassifier
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import f1_score, classification_report,
    precision_recall_curve, make_scorer

df = pd.read_csv(r'C:\Users\Naomi\Desktop\credit_risk_dataset masters.csv'
    )

# Handling Missing Values
num_cols = df.select_dtypes(include=['float64', 'int64']).columns
imputer = SimpleImputer(strategy='median')
df[num_cols] = imputer.fit_transform(df[num_cols])

cat_cols = df.select_dtypes(include=['object']).columns
imputer_cat = SimpleImputer(strategy='most_frequent')
df[cat_cols] = imputer_cat.fit_transform(df[cat_cols])

missing_values = df.isna().sum().sum()

# Label Encoding Loan Grade
encoder = LabelEncoder()
df['Loan Grade'] = encoder.fit_transform(df['Loan Intent'])

# One-Hot Encoding
df = pd.get_dummies(df, columns=['Home Ownership', 'Loan Intent', '
    Historical Defaults'])

X = df.drop('Loan Status', axis=1)
y = df['Loan Status']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
    random_state=42, stratify=y)

from sklearn.preprocessing import RobustScaler
scaler = RobustScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

```

```
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import SelectFromModel

log_reg = LogisticRegression(penalty='l1', solver='liblinear', C=1.0,
    random_state=42)
log_reg.fit(X_train_scaled, y_train)

selector = SelectFromModel(log_reg, prefit=True)
selected_features = X.columns[selector.get_support()]

X_train_selected = selector.transform(X_train_scaled)
X_test_selected = selector.transform(X_test_scaled)

from imblearn.pipeline import Pipeline
from imblearn.combine import SMOTETomek

sampling_pipeline = Pipeline([
    ('smote_tomek', SMOTETomek(random_state=42))
])

param_grid = {'smote_tomek__sampling_strategy': [0.3, 0.5, 0.7, 1.0]}

from sklearn.model_selection import GridSearchCV
grid_search = GridSearchCV(sampling_pipeline, param_grid, scoring=
    make_scorer(f1_score), cv=3, n_jobs=-1)
grid_search.fit(X_train_selected, y_train)

best_params = grid_search.best_params_

best_SMOTETomek = SMOTETomek(sampling_strategy=best_params['
    smote_tomek__sampling_strategy'], random_state=42)
X_train_resampled, y_train_resampled = best_SMOTETomek.fit_resample(
    X_train_selected, y_train)

from sklearn.svm import SVC
svm_model = SVC(kernel='linear', probability=True, random_state=42)
svm_model.fit(X_train_resampled, y_train_resampled)

y_pred = svm_model.predict(X_test_selected)
y_pred_proba = svm_model.predict_proba(X_test_selected)[:, 1]

from sklearn.metrics import accuracy_score, precision_score, recall_score,
     roc_auc_score
import matplotlib.pyplot as plt

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
auc = roc_auc_score(y_test, y_pred_proba)

from sklearn.ensemble import VotingClassifier
from sklearn.tree import DecisionTreeClassifier

hybrid_clf = VotingClassifier(
    estimators=[
        ('dt', DecisionTreeClassifier(random_state=42)),
        ('svm', SVC(kernel='linear', probability=True, random_state=42))
    ],
    voting='hard'
)
```

```
97
98 hybrid_clf.fit(X_train_resampled, y_train_resampled)
99 y_pred = hybrid_clf.predict(X_test_selected)
100
101 accuracy = accuracy_score(y_test, y_pred)
102 precision = precision_score(y_test, y_pred)
103 recall = recall_score(y_test, y_pred)
104 f1 = f1_score(y_test, y_pred)
```

...

## Appendix F: Similarity Report

Optimizing Credit Risk Assessment with Ensemble Sampling and Hybrid Machine Learning Models.pdf

ORIGINALITY REPORT

| 16% | 16% | 16% | 8% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | su-plus.strathmore.edu<br>Internet Source | 1% |
| 2 | Submitted to Strathmore University<br>Student Paper | 1% |
| 3 | eitca.org<br>Internet Source | 1% |
| 4 | Nekuri Naveen. "Application of fuzzyARTMAP for churn prediction in bank credit cards", International Journal of Information and Decision Sciences, 2009<br>Publication | 1% |
| 5 | Submitted to Colegio Universitario de Estudios Financiero<br>Student Paper | 1% |
| 6 | www.mdpi.com<br>Internet Source | 1% |
| 7 | "Practical Statistical Learning and Data Science Methods", Springer Science and Business Media LLC, 2025<br>Publication | 1% |
| 8 | arxiv.org<br>Internet Source | <1% |
| 9 | "Intelligent Systems and Pattern Recognition", Springer Science and Business Media LLC, | <1% |

| | | |
|---|---|---|
| | 2025<br>Publication | |
| 10 | Submitted to Massey University<br>Student Paper | <1 % |
| 11 | link.springer.com<br>Internet Source | <1 % |
| 12 | Xinyuan Song, HSIEH,WEI-CHE, Ziqian Bi, Chuanqi Jiang, Junyu Liu, Benji Peng, Sen Zhang, Xuanhe Pan, Jiawei Xu, Jinlang Wang. "A Comprehensive Guide to Explainable AI: From Classical Models to LLMs", Open Science Framework, 2024<br>Publication | <1 % |
| 13 | S. Prasad Jones Christydass, Nurhayati Nurhayati, S. Kannadhasan. "Hybrid and Advanced Technologies", CRC Press, 2025<br>Publication | <1 % |
| 14 | "Computational Science – ICCS 2019", Springer Science and Business Media LLC, 2019<br>Publication | <1 % |
| 15 | www.fastercapital.com<br>Internet Source | <1 % |
| 16 | Dina Elreedy, Amir F. Atiya, Firuz Kamalov. "A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning", Machine Learning, 2023<br>Publication | <1 % |
| 17 | www.geeksforgeeks.org<br>Internet Source | <1 % |
| 18 | www.math.colostate.edu | |

## Appendix G: Strathmore University Institutional Ethics Review Clearance Certificate

**14th November 2024**

Ms Mucheru Naomi,
naomi.mucheru@strathmore.edu

Dear Ms Mucheru,

**RE: Optimizing Credit Risk Assessment with Ensemble Sampling and Hybrid Machine Learning Models**

This is to inform you that SU-ISERC has reviewed and **approved** your above **SU-masters** proposal. Your application reference number is **SU-ISERC2238/24.** The approval period is from **14th November 2024 to 13th November 2025.**

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used.
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-ISERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-ISERC within 72 hours of notification.
iv. Any changes anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-ISERC within 72 hours.
v. Clearance for the export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to the expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days of completion of the study to SU-ISERC.

Before commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology, and Innovation (NACOSTI) https://research-portal.nacosti.go.ke/ and obtain other clearances needed.

Yours sincerely,

**Mr Ambrose Rachier,**
**Chairperson; SU-ISERC**