PURBANCHAL UNIVERSITY

KANTIPUR CITY COLLEGE (KCC)

Putalisadak, Kathmandu, Nepal

Masters of Computer Application (M.C.A.)

Reg. No.

A Thesis Proposal on

"A Comparative Analysis of Data Balancing Techniques in Machine Learning for Enhancing Bank Loan Default Risk Predictions"

IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF COMPUTER APPLICATION

Submitted by

Manoj Rai

2024

ABSTRACT

Due to substantial technological advancements, people's needs have expanded. Consequently, there has been an increase in the number of loan approval requests in the banking industry. Several criteria are considered while selecting a candidate for loan approval in order to ascertain the loan's status. Banks encounter major challenges in evaluating loan applications and mitigating the risks linked to prospective borrower defaults. Due to the need to thoroughly assess the eligibility of every borrower for a loan, banks consider this process as notably burdensome. First the balancing of dataset will the performed. For this two oversampling techniques SMOTE and ADASYN will be compared. To determine the most effective balancing technique for loan approval, various algorithms such as Logistic Regression and Support Vector Machines (SVM) will be compared. Each technique comes with its own set of strengths and weaknesses, making the choice dependent on the specific characteristics of the dataset and the goals of the lending institution.

Keywords: ADASYN, Loan, Logistic Regression, ML, Prediction, SMOTE, SVM

Table of Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1: INTRODUCTION

1.1 Background

In an era marked by unprecedented technological progress, the banking industry is undergoing a transformative shift in response to the growing and diverse needs of its clientele. As individuals seek financial support for a myriad of purposes, from homeownership to education and entrepreneurship, the volume of loan approval requests has surged, necessitating a more sophisticated approach to credit risk assessment. Recognizing the limitations of traditional manual underwriting methods, financial institutions are increasingly turning to machine learning (ML) algorithms to enhance the efficiency and accuracy of their loan approval processes. This research seeks to delve into the intricate landscape of ML applications within the banking sector, with a particular focus on evaluating and comparing three prominent algorithms – logistic regression, and support vector machines (SVM) – in the context of loan approval. As the banking industry grapples with the complexities of assessing a myriad of factors influencing creditworthiness, from financial histories to debt-to-income ratios, the need for a robust and streamlined approach to credit risk management has never been more pressing.

In the contemporary financial landscape, particularly within the banking sector, the ability to accurately predict loan eligibility has become increasingly critical. This process, traditionally based on manual assessment, has evolved significantly with the integration of machine learning (ML) techniques. These advancements have enhanced the efficiency and accuracy of predictions, making them indispensable tools for financial institutions. However, an inherent challenge in this process is the prevalence of class imbalance in datasets – a situation where instances of one class (e.g., loan defaults) are far fewer than those of another (e.g., loan approvals). This disparity can lead to skewed predictive models that favor the majority class, resulting in decisions that might be unfair or misrepresentative of the minority class. Addressing this imbalance is crucial, not just for achieving technical accuracy, but also for maintaining ethical standards and regulatory compliance in financial decision-making.

To counter the issues posed by class imbalance, SMOTE (Synthetic Minority Over-sampling Technique) and ADASYN (Adaptive Synthetic Sampling) have emerged as prominent solutions. These techniques are designed to artificially balance datasets by generating synthetic examples of the minority class. SMOTE achieves this by creating samples that are interpolations of existing minority instances and their nearest neighbors. In contrast, ADASYN places more emphasis on generating samples near the minority instances that are harder to learn, thus creating a more adaptive approach to balancing. The application of these techniques has shown potential in various fields, but their impact in the specific context of bank loan eligibility prediction necessitates further exploration.

The effectiveness of machine learning models such as SVM, and Logistic Regression is well-recognized in predictive analytics. Each of these models has unique characteristics that make them suitable for different types of data and prediction tasks. SVM is known for its effectiveness in high-dimensional spaces and its ability to model non-linear decision boundaries. Logistic Regression, a more straightforward model, is prized for its interpretability and efficiency in binary classification tasks. However, the interplay between these models and balancing techniques in the context of imbalanced datasets, especially for bank loan eligibility, is a subject that warrants comprehensive study.

1.2 Statement of Problem

In the realm of financial services, specifically within the banking sector, the decision-making process for loan eligibility is of paramount importance. With the advent of machine learning (ML) techniques, this process has seen significant advancements in terms of efficiency and accuracy. However, a critical and pervasive issue in this domain is the challenge posed by imbalanced datasets. Typically, in loan eligibility datasets, one class (e.g., loan defaults or rejections) is underrepresented compared to another (e.g., loan approvals), leading to a class imbalance. This imbalance can significantly skew the performance of predictive models, resulting in biases towards the majority class and potentially unfair or inaccurate loan eligibility decisions.

The challenge is further compounded when considering different machine learning models such as Support Vector Machine (SVM), and Logistic Regression, each

with its unique approach to data analysis and prediction. The effectiveness of these models in handling imbalanced datasets, particularly in the context of loan eligibility, is a subject that has not been exhaustively explored. Furthermore, while balancing techniques like SMOTE (Synthetic Minority Over-sampling Technique) and ADASYN (Adaptive Synthetic Sampling) have been proposed to address class imbalance, their impact on the predictive accuracy and bias of these specific models in the banking sector remains unclear.

1.3 Research Objectives

- To compare SMOTE and ADASYN balancing technique with the help of machine learning algorithms.
- To build a robust model for bank loan default risk predictions

Chapter 2: LITERATURE REVIEW

This section presents a literature survey. Relevant literature from multiple sources is referred for analysis of loan prediction system:

In 1998, Andrew Callun et al.[1] conducted a series of Monte Carlo simulations to analyze how different data characteristics affect the classifier's performance. Despite the fundamental assumption of feature independence in Naive Bayes – often considered a weak point – the classifier was found to perform surprisingly well in practice. The study revealed that Naive Bayes shows good performance when feature distributions have low entropy, and it operates effectively in scenarios with completely independent features, as well as in cases with nearly-functional feature dependencies. A striking finding was that the accuracy of Naive Bayes did not directly correlate with the degree of feature dependency, but rather with the amount of class information lost due to the independence assumption. This insight challenges the conventional understanding of Naive Bayes and highlights its potential effectiveness even when its basic assumptions are not strictly met.

In 2002, Nitesh V Chawla et al.[2] an approach for constructing classifiers from imbalanced datasets, where the representation of classification categories is uneven. This imbalance is common in real-world datasets, often dominated by 'normal' examples with a relatively small proportion of 'abnormal' or 'interesting' examples. The study emphasized that the misclassification of an abnormal example as normal often carries a higher cost than the reverse error. A method involving under-sampling of the majority (normal) class was proposed to enhance the sensitivity of classifiers to the minority class. The paper's key contribution is the demonstration that combining over-sampling of the minority class with under-sampling of the majority class results in superior classifier performance, as measured in ROC (Receiver Operating Characteristic) space, compared to solely under-sampling the majority class. This finding was also compared against alternative approaches like adjusting the loss ratios in Ripper or class priors in Naive Bayes, with the combined over- and under-sampling method showing better performance. The method for

over-sampling the minority class involved creating synthetic examples of this class. The research utilized classifiers such as C4.5, Ripper, and Naive Bayes, and the evaluation was based on the area under the ROC curve (AUC) and the ROC convex hull strategy. This approach offers a significant advancement in dealing with imbalanced datasets, providing a more effective way to handle the challenges of misclassification costs in such scenarios.

In 2008, Haibo He et al.[3] introduced a novel approach known as the Adaptive Synthetic (ADASYN) sampling method, designed to enhance learning from imbalanced data sets. The core concept of ADASYN revolves around creating a weighted distribution for different examples within the minority class, based on their learning difficulty. This approach prioritizes generating more synthetic data for those minority class examples that are more challenging to learn, as opposed to easier ones. The innovation of the ADASYN method lies in its dual impact on learning: firstly, it significantly reduces the bias that typically arises from class imbalance, and secondly, it adaptively modifies the classification decision boundary in favor of the more difficult examples. This shift in the decision boundary is crucial for addressing the issues posed by imbalanced data sets. The effectiveness of the ADASYN method was validated through simulation analyses on various machine learning data sets. The results, assessed across five different evaluation metrics, demonstrated the robustness and efficiency of this method in improving learning outcomes in scenarios where data imbalance is a prevalent challenge.

In 2019, Chauhan et al.[4] presented a detailed review of the Support Vector Machine (SVM), a key classification technique in machine learning known for its optimal margin-based approach. Initially developed as a binary linear classifier, SVM has been expanded to accommodate non-linear data through the use of Kernels and multi-class data via various methods such as one-versus-one, one-versus-rest, Crammer Singer SVM, Weston Watkins SVM, and Directed Acyclic Graph SVM (DAGSVM). Distinguished by the type of Kernel used, SVMs are categorized into linear and non-linear types, with linear SVMs being particularly

efficient for high-dimensional data applications like document classification, word-sense disambiguation, and drug design. This is due to the comparable test accuracy of linear SVMs to their non-linear counterparts and their faster training time. The paper underscores the continuous evolution of SVM since its inception, highlighting the development of various problem formulations, solvers, and strategies for its application. It also addresses the challenges posed by 'Big Data' in training classifiers, reflecting on the advancements in technology that have led to the generation of large-scale datasets. This review not only provides an overview of the evolution of linear SVM classification but also delves into the solvers, strategies for solver improvement, experimental results, current challenges, and future research directions in the field, making it a comprehensive resource for understanding the dynamics and potential of SVM in the modern landscape of machine learning.

In 2022, Mirza Muntasir Nishat et al.[5] tackled the challenge of predicting patient survival in cases of heart failure, a severe cardiac condition. Utilizing a dataset from the UCI Machine Learning Repository that includes 299 individuals, they explored six supervised machine learning algorithms: Decision Tree Classifier, Logistic Regression, Gaussian Naïve Bayes, Random Forest Classifier, K-Nearest Neighbors, and Support Vector Machine. The study involved preprocessing steps like data scaling (standard and min–max scaling) and employed hyperparameter optimization techniques including grid search and random search cross-validation. Notably, the study incorporated the Synthetic Minority Oversampling Technique and Edited Nearest Neighbor (SMOTE-ENN) for addressing the imbalance in the dataset. A comprehensive comparison of these methods revealed that the Random Forest Classifier, particularly when combined with SMOTE-ENN and standard scaling, significantly outperformed the others with a test accuracy of 90%. This finding underscores the efficacy of specific machine learning algorithms and preprocessing techniques in managing imbalanced datasets for critical healthcare predictions, such as survivability in heart failure cases.

In 2022, Ugochukwu Orji et al.[6] delves into the transformative role of machine learning algorithms across various sectors, with a particular focus on the financial industry's loan approval process. The study investigates the application of six machine learning algorithms – Random Forest, Gradient Boost, Decision Tree, Support Vector Machine, K-Nearest Neighbor, and Logistic Regression – to enhance the speed, efficacy, and accuracy of loan approval processes. Utilizing the 'Loan Eligible Dataset' from Kaggle, processed with Python libraries in Kaggle's Jupyter Notebook environment, the research demonstrates notable results. The algorithms showed high-performance accuracy, with Random Forest leading at a 95.55% accuracy rate, while Logistic Regression was at 80%, the lowest among the tested models. These models not only achieved high accuracy but also outperformed two of the three existing loan prediction models from the literature in terms of precision-recall and overall accuracy. This finding highlights the significant potential of machine learning in streamlining and improving the efficiency of loan eligibility predictions in the banking sector.

In 2022, Ch. Naveen Kumar et al. investigated the application of various machine learning techniques to predict loan eligibility, a pressing need given the rapid growth of the banking and financial sector and increasing reliance of individuals on bank loans. The study involves collecting customer data from multiple banks and analyzing customer profiles using key parameters integral to machine learning. Unlike traditional loan approval systems, this project adopts a machine learning approach to analyze data and determine loan eligibility based on customer profiles. The project's core objectives include data cleansing, selection of key attributes, and a performance comparison of several machine learning methods: Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbor, and Decision Tree with AdaBoost. The data is split into training and testing segments, with the model trained on the training dataset and evaluated on the testing dataset. The results indicate that the ensemble model, Decision Tree with AdaBoost, outperforms the other models in accuracy, showcasing the potential of advanced machine learning techniques in enhancing the efficiency and effectiveness of loan eligibility predictions in the banking sector.

In 2023, Anthony Anggrawan et al. [7] addressed the crucial need for higher education institutions to accurately predict student graduation timelines, a key indicator of institutional success and a vital component of accreditation. The study recognizes the challenge in classifying students who graduate on time versus those who do not, especially given the limitations of educational technology and machine learning with data mining approaches in handling unbalanced data. To improve the classification performance on such unbalanced class data, the study introduces the Synthetic Minority Oversampling Technique (SMOTE) for enhancing the effectiveness of the Support Vector Machine (SVM) data mining method. The application of SMOTE resulted in a significant increase in the accuracy, precision, and sensitivity of the SVM method for classifying unbalanced student graduation data. Specifically, the study found that the SVM performance improved by 3% in classification accuracy, 8% in precision, and 25% in sensitivity, demonstrating the effectiveness of combining SMOTE with SVM in addressing the challenges of unbalanced data in educational settings.

In 2023, Dina Elreedy et al. [8] addressed the class imbalance problem in data, a common issue in real-world applications where one class (the minority) is under-represented compared to another (the majority). The Synthetic Minority Over-sampling Technique (SMOTE) is a widely recognized method for handling such imbalances by creating synthetic data patterns through linear interpolation between minority class samples and their nearest neighbors. However, a critical concern is that these SMOTE-generated patterns may not accurately reflect the original minority class distribution. In this study, the authors develop a novel theoretical analysis of SMOTE by deriving the probability distribution of the SMOTE-generated samples, marking the first attempt to mathematically formulate the probability distribution of SMOTE patterns. This development allows for a comparison between the density of the generated samples and the true underlying class-conditional density, providing insights into the representativeness of the generated samples. The validity of the derived formula is confirmed through calculations on various densities and comparing them with densities empirically

estimated, thereby offering a significant advancement in understanding and evaluating the effectiveness of the SMOTE method in addressing class imbalances.

In 2023, Tarid Wongvorachan et al.[9] explored the impact of class imbalance in educational datasets on the accuracy of predictive models used in educational data mining, a field known for developing data-driven applications like early warning systems and predicting students' academic achievement. Recognizing that many predictive models are built assuming balanced classes, the study addresses the challenge posed by imbalanced classes, which can significantly affect model accuracy. While previous research primarily focused on technical improvements of various techniques, this study shifts the focus towards their application, particularly for data with varying imbalance ratios. Using the High School Longitudinal Study of 2009 dataset, the authors compared several sampling techniques tailored for different degrees of imbalance: random oversampling (ROS), random undersampling (RUS), and a hybrid approach combining the synthetic minority oversampling technique for nominal and continuous data (SMOTE-NC) with RUS. The Random Forest algorithm was employed to evaluate the effectiveness of these techniques. The findings suggest that random oversampling is more effective for moderately imbalanced data, while the hybrid resampling approach is better suited for extremely imbalanced data. The study concludes with implications for educational data mining applications and recommendations for future research, highlighting the importance of selecting appropriate sampling techniques based on the degree of class imbalance in educational datasets.

In 2023, Ishani Dey et al.[10] delves into the realm of machine learning and its significant role in solving real-world problems such as credit card fraud detection, cancer susceptibility and survival prediction, spam identification, and customer segmentation. Central to the effectiveness of machine learning is its reliance on large volumes of data to deliver accurate predictions. The accuracy of a machine learning model is intrinsically linked to the quality and balance of the dataset it processes. This brings into focus the techniques of oversampling and under-sampling, crucial for dataset balancing. Under-sampling involves reducing the

majority class by removing samples, whereas oversampling entails adding synthetic samples to the minority class to achieve balance. This particular study focuses on three specific methods: SMOTE (Synthetic Minority Over-sampling Technique), Borderline-SMOTE, and ADASYN (Adaptive Synthetic Sampling). It encompasses a comprehensive comparison of these oversampling techniques, evaluating them based on their impact on accuracy, precision, recall, F1-measure, and ROC (Receiver Operating Characteristic) curve performance. The study aims to provide a clear understanding of how each technique affects the overall effectiveness of machine learning models in varied applications, thereby aiding in the selection of the most appropriate method for dataset balancing in machine learning tasks.

In 2023, Saini et al. addresses the crucial task of predicting loan approvals, a challenge that financial institutions have historically navigated through manual and subjective processes, often leading to inconsistent decisions and a heightened risk of loan defaults. With the emergence of machine learning, there's an opportunity to create more accurate and consistent predictive models to aid in lending decisions. The study conducts a comparative analysis of various machine learning algorithms – Random Forest Classifier, K-Nearest Neighbors Classifier, Support Vector Classifier, and Logistic Regression – applied to loan approval prediction. Through exploratory data analysis and feature engineering, the dataset is prepared and the performance of each algorithm is evaluated using metrics such as accuracy score, F1 score, and ROC score. The results demonstrate that the Random Forest Classifier outperforms the others with a high accuracy of 98.04%, followed by K-Nearest Neighbors Classifier, Logistic Regression, and Support Vector Classifier. These findings underscore the capability of machine learning algorithms in refining the loan approval process and minimizing the risk of defaults. This study not only provides valuable insights for financial institutions looking to enhance their decision-making processes but also sets a precedent for applying similar approaches to other domains where classification is essential.

In 2023, Robert C. Moore et al. [11] studied the issue of data balancing. This issue is particularly evident in AudioSet, which encompasses a broad spectrum of 527 sound event classes with varying frequencies of occurrence. Commonly, the evaluation of classification performance on AudioSet is conducted by averaging metrics across all classes, assigning equal importance to both rare and common classes. Although recent studies have implemented dataset balancing techniques to enhance performance on AudioSet, this paper presents a critical finding: while such balancing appears to improve performance on the public AudioSet evaluation data, it adversely affects performance on an unpublished, similarly collected evaluation set. Through experimentation with varying degrees of dataset balancing, it becomes apparent that the effectiveness of balancing is highly contingent on the chosen evaluation set. Additionally, there is no substantial evidence to suggest that dataset balancing preferentially enhances the performance of rare classes over common ones. Therefore, the study advises caution against the indiscriminate use of balancing techniques and emphasizes the need for skepticism towards minor performance improvements reported on public evaluation sets. This cautionary note highlights the complexity of balancing techniques and their impact on model performance, especially in datasets with a wide range of class priors like AudioSet.

In 2023, Archana Archana[12] discussed the pivotal role of banking in bridging the gap between individuals with surplus funds and those in need of financial resources, underscoring the bank's integral part in the financial system. People rely on banks for loans, which are disbursed with an interest rate and are subject to repayment. The decision to approve these loans is traditionally based on a variety of borrower-specific factors. With the advent of artificial intelligence, financial institutions have found a way to expand their lending operations while managing financial risks effectively. The introduction of machine learning and deep learning models into the loan approval process promises to enhance its speed, efficiency, and accuracy. This study embarks on a comprehensive comparison of ten different Machine Learning models — including Decision Tree, Logistic Regression, K Nearest Neighbour (KNN), Random Forest Classifier, Support Vector Machine (SVM), Linear Discriminant Analysis (LDA), Gaussian Naive Bayes, XGBoost, Gradient

Boosting, and Adaboost — along with Deep Learning models like the Deep Neural Network (DNN) and Long Short Term Memory network (LSTM). The aim is to evaluate their effectiveness in predicting which loan applicants are most deserving of funding, highlighting the potential of these advanced technological models in revolutionizing the loan approval process in the banking sector.

In 2023, Viswanatha et al. [13] addressed the heightened demand for loan approvals in the banking sector, a consequence of significant technological advancements and escalating financial needs of individuals. Faced with the challenge of meticulously assessing loan applications to minimize the risk of borrower defaults, banks find the loan approval process arduous and time-consuming. To tackle this, the study proposes the integration of machine learning (ML) models and ensemble learning approaches to enhance the accuracy in identifying eligible loan applicants. This approach is designed to streamline the approval process, benefiting both loan applicants and bank employees by dramatically reducing the sanctioning time. As the banking sector expands, leading to an increased volume of loan applications, the study evaluates the efficacy of four algorithms: Random Forest, Naive Bayes, Decision Tree, and KNN. The findings indicate that the Naïve Bayes algorithm, with an accuracy of 83.73%, is the most effective, showcasing the potential of ML in revolutionizing the loan approval process in banks by offering a more efficient, accurate, and reliable method for evaluating loan applications.

Chapter 3: RESEARCH METHODOLOGY

3.1 System Flow Diagram

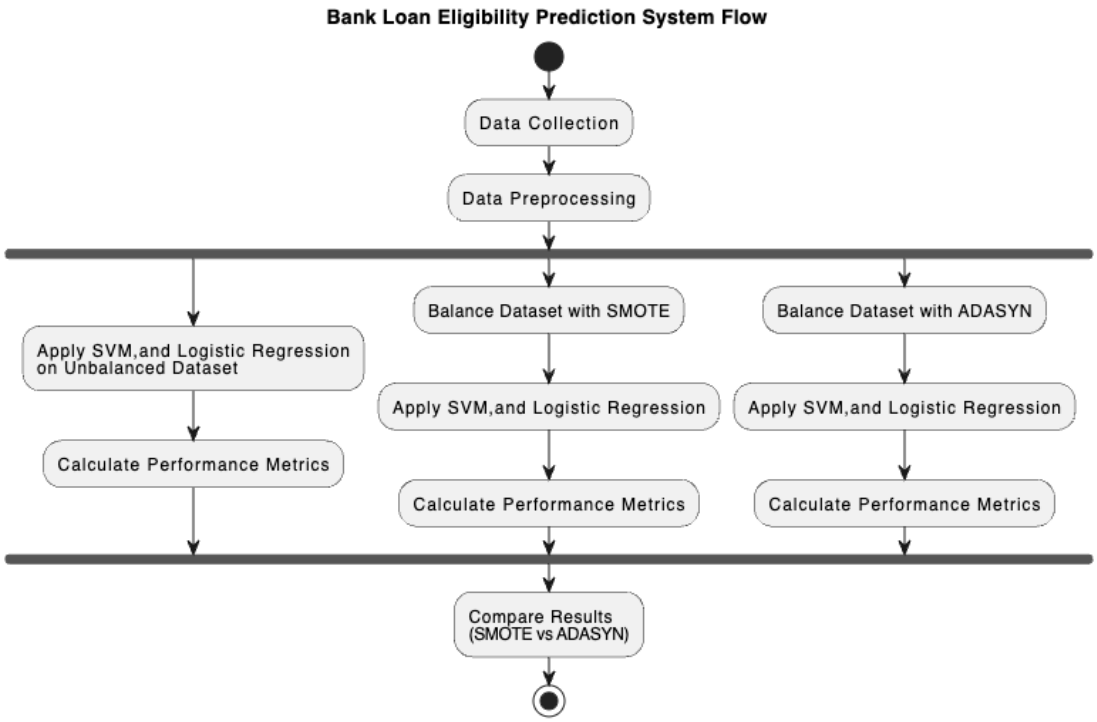

Figure 3-1:System Flow Diagram

3.1.1 Data Collection and Preprocessing

The process initiates with the collection of relevant data, which forms the foundation of the study. This data is likely to include various attributes of loan applicants that are critical for determining loan eligibility. Once the data was collected, it undergoes preprocessing. This crucial phase involves cleaning the data to ensure its quality, which may include handling missing values, encoding categorical variables, normalizing data for uniformity, and selecting features that are most relevant to the loan approval process.

3.1.2 Application of ML algorithms on unbalanced dataset

Following preprocessing, the methodology takes its first path by applying three distinct machine learning algorithms - Support Vector Machine (SVM) and Logistic Regression - directly on the unbalanced dataset. This step is essential to establish a baseline performance of these algorithms on the dataset without any balancing

techniques applied. The effectiveness of each algorithm is assessed by calculating key performance metrics such as accuracy, precision, recall, and F1-score.

3.1.3 Balancing the Dataset with SMOTE and ADASYN

Parallel to the first path, two additional paths explore the impact of data balancing techniques on the performance of the same machine learning algorithms. The first of these uses the Synthetic Minority Over-sampling Technique (SMOTE) to balance the dataset by creating synthetic samples for the minority class. The second path employs Adaptive Synthetic Sampling (ADASYN), another technique for balancing the dataset by generating synthetic samples, but with a focus on those samples that are difficult to learn. In both paths, after applying the respective data balancing technique, the study re-applies the SVM, and Logistic Regression algorithms to these balanced datasets. The performance of these algorithms on the balanced datasets is then evaluated using the same metrics as before.

3.1.4 Comparative Analysis

The final step of the methodology involves a comparative analysis of the results obtained from the unbalanced dataset and the datasets balanced using SMOTE and ADASYN. This comparison is pivotal to determine which data balancing technique - SMOTE or ADASYN - is more effective in improving the predictive performance of the machine learning models.

3.2 Methods of Data Collection

For this research work, datasets are collected from Kaggle, Loan approval analysis | Kaggle The size of the dataset is 4268 samples, which have nine fields, where 11 fields are for input charactertics and one field for an output field. are representing the input fields, while the output field pertains to the presence of heart attack (class), which is divided into two categories (negative and positive); negative refers to the rejection of a loan, while positive refers to the approval of a loan.

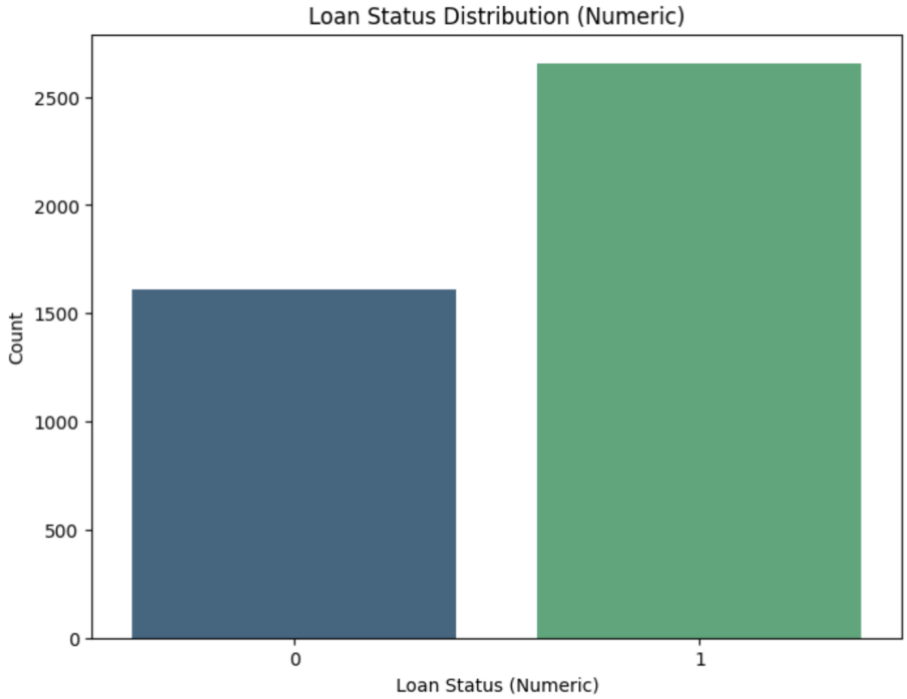

Figure 3-2: Dataset Visualization

3.3 Data Preprocessing

3.3.1 Label Encoding of Categorical Data

The preprocessing procedure commenced with the conversion of categorical variables into a machine-readable format. This was executed by employing the LabelEncoder functionality provided within the sklearn.preprocessing package. Each distinct category within the categorical variables was assigned a unique integer identifier. This encoding was imperative for facilitating the processing of categorical data by subsequent machine learning algorithms, which necessitate numerical input.

3.3.2 Training and Testing Dataset Split

Following the encoding, the dataset was bifurcated into training and testing subsets. The division was realized through the train_test_split method from the sklearn.model_selection suite. An 80:20 ratio was adhered to for the split, allocating 80% of the data to the training subset for the purpose of model training, and the remaining 20% to the testing subset for model evaluation.

3.3.3 Data Standardization

The next phase entailed the standardization of numerical features within the dataset. The StandardScaler from the sklearn.preprocessing library was utilized for this process. It standardized the features to a common scale by subtracting the mean and scaling to unit variance. This standardization was pivotal in ensuring that all features contributed equivalently to the model's predictions and that no single feature with a larger scale unduly influenced the model output.

Concluding the preprocessing stages, the dataset was subjected to balancing techniques to address class imbalance issues. The ADASYN (Adaptive Synthetic Sampling) and SMOTE (Synthetic Minority Over-sampling Technique) methods from the imblearn.over_sampling module were employed. Both techniques generated synthetic instances of the under-represented class in the dataset. ADASYN placed an emphasis on generating synthetic samples adjacent to those instances that were misclassified, while SMOTE created synthetic instances by interpolating between existing minority instances. These methods effectively balanced the class distribution, reducing the model's inherent bias towards the majority class and enhancing its predictive performance across all classes.

3.4 Machine Learning Algorithms

For the development of model, various ML algorithm will be taken into consideration such as Support Vector Machine (SVM), and Logistic Regression.

3.4.1 Support Vector Machine

Support Vector Machine (SVM) was first heard in 1992, introduced by Boser, Guyon, and Vapnik in COLT-92. Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression Support Vector Machine (SVM), in other words, is a classification and regression prediction tool that automatically detects over-fitting to the data while maximizing predictive accuracy using machine learning theory. Systems can be defined as Support Vector machines. This uses the linear function's hypotheses space in the high-dimensional feature space and was trained using an optimization theory-based learning algorithm that uses a learning bias obtained from theory of statistical learning.

The basic idea of SVM is to construct an optimal hyper plane, which can be used for classification, The optimal hyper plane is a hyper plane selected from the set of hyper planes for classifying patterns that maximizes the margin of the hyper planes.

- **Linear SVM**

  When the data is perfectly linearly separable only then we can use Linear SVM. Perfectly linearly separable means that the data points can be classified into 2 classes by using a single straight line(if two dimensional).

- **Non-Linear SVM**

  When the data is not linearly separable then we can use Non-Linear SVM, which means when the data points cannot be separated into 2 classes by using a straight line (if 2D) then we use some advanced techniques like kernel tricks to classify them. In most real-world applications we do not find linearly separable datapoints hence we use kernel trick to solve them.

The main two terms that are used in SVM are: -

Support Vectors: These are the points that are closest to the hyperplane. A separating line will be defined with the help of these data points.

Margin: it is the distance between the hyperplane and the observations closest to the hyperplane (support vectors). In SVM large margin is considered a good margin. There are two types of margins hard margin and soft margin.

The equation for hyperplane can be given as:

$$aX + bY = C \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \textit{(1)}$$

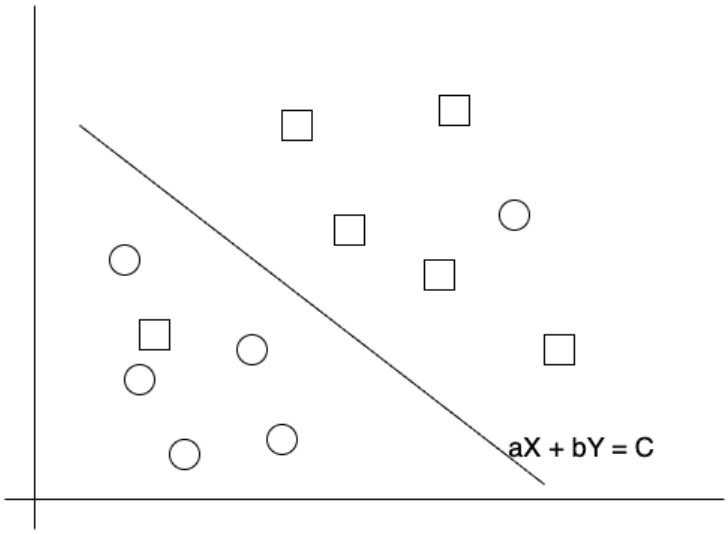

Figure 3-3: SVM Hyperplane

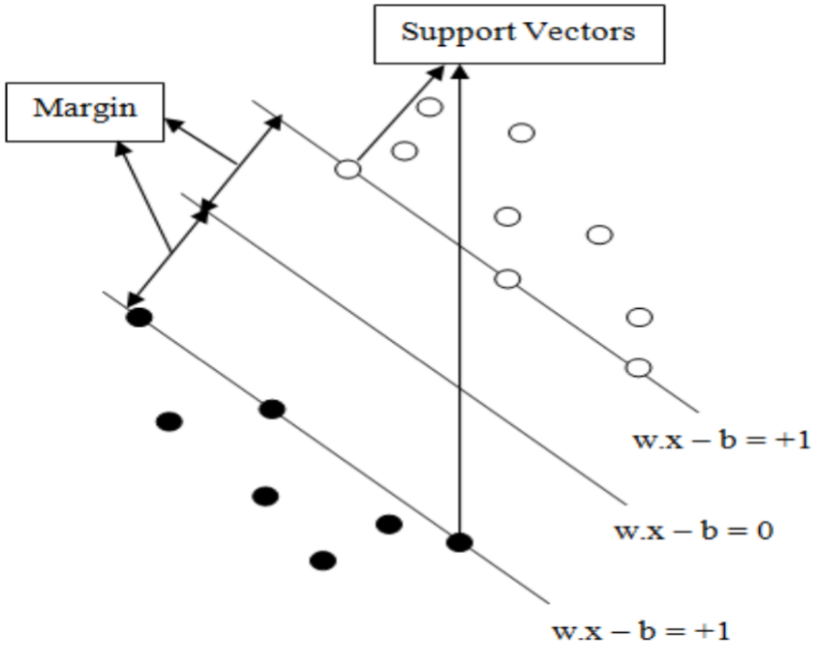

Figure 3-4: SVM Model

The figure 3.4 shows that the model consists of three different lines. These three lines construct the hyper plane that separates the given patterns and the pattern that lies on the edges of the hyper plane is called support vectors. The perpendicular distance between the line of margin and the edge of hyper plane is known as margin. The objective of support vectors is to optimize the margin so that it can classify the given problem.

3.4.2 Logistic Regression

Logistic Regression uses the sigmoid (logistic) function to model the probability of a binary outcome. The sigmoid function transforms any real-valued number into the range [0, 1]. The model is trained using a set of labeled data, where the features are used to predict the probability of belonging to the positive class. The parameters (coefficients) are adjusted during training to minimize the difference between predicted probabilities and actual class labels. Logistic Regression creates a decision boundary that separates the data into two classes. If the predicted probability is above a certain threshold (usually 0.5), the instance is classified as the positive class; otherwise, it is classified as the negative class.

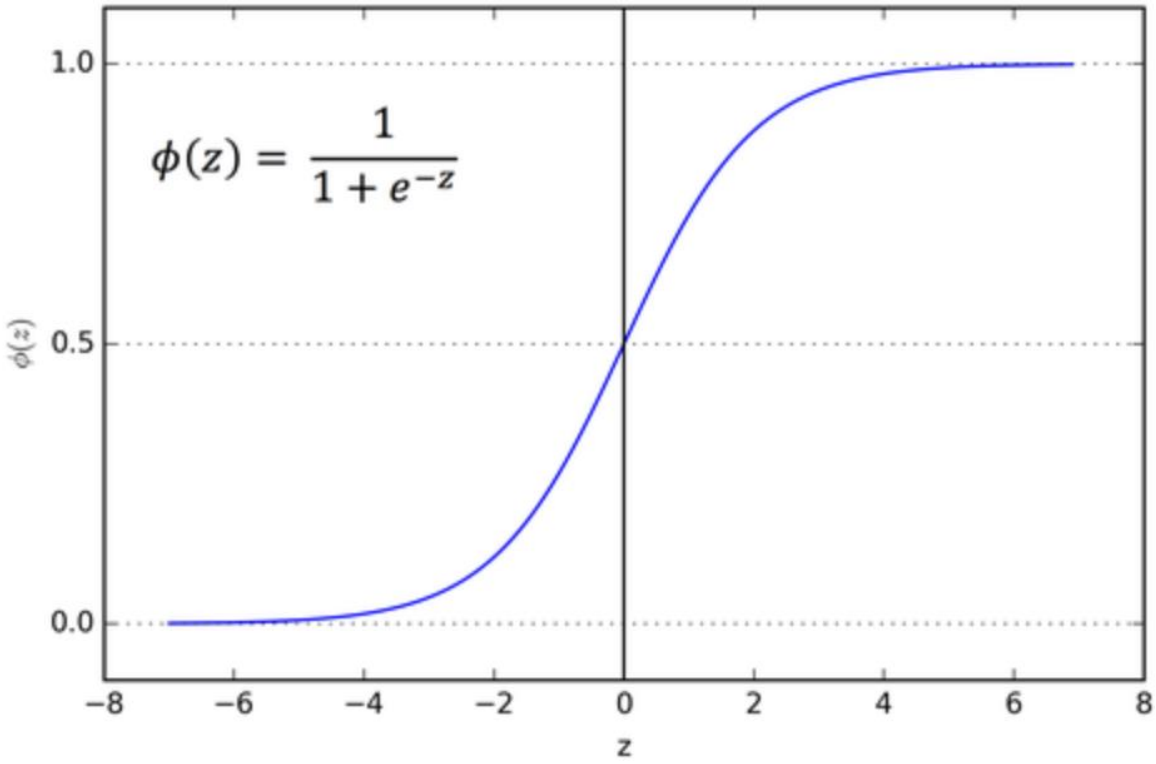

Figure 3-5: Sigmoid Function for Logistic Regression

Logistic Regression provides interpretable coefficients, allowing for a clear understanding of the impact of each feature on the predicted outcome. It performs well when the relationship between features and the log-odds of the outcome is

approximately linear. Logistic Regression is less prone to overfitting, especially when the number of features is relatively small.

3.5 Balancing Technique

In the realm of data analysis, particularly in scenarios where dataset imbalance is a significant challenge, balancing techniques play a crucial role. Two prominent techniques are SMOTE (Synthetic Minority Over-sampling Technique) and ADASYN (Adaptive Synthetic Sampling). These techniques are particularly valuable in handling imbalanced datasets, a common occurrence in many real-world applications such as fraud detection, medical diagnosis, and sentiment analysis.

3.5.1 SMOTE (Synthetic Minority Over-sampling Technique)

Imbalanced class distribution is a common challenge in classification tasks, where one class significantly outnumbers the other. SMOTE, or Synthetic Minority Over-sampling Technique, is a resampling technique designed to address this issue by oversampling the minority class. Developed by Nitesh Chawla et al., SMOTE works by generating synthetic instances of the minority class, thereby balancing the class distribution.

SMOTE creates synthetic instances by interpolating between existing minority class instances. For each minority instance, it selects its k nearest neighbors and generates synthetic samples along the line segments connecting the instance to its neighbors. SMOTE introduces adaptability by allowing the user to control the level of oversampling through the parameter k. A higher k value results in a more aggressive oversampling, whereas a lower value creates a more conservative oversampling strategy.

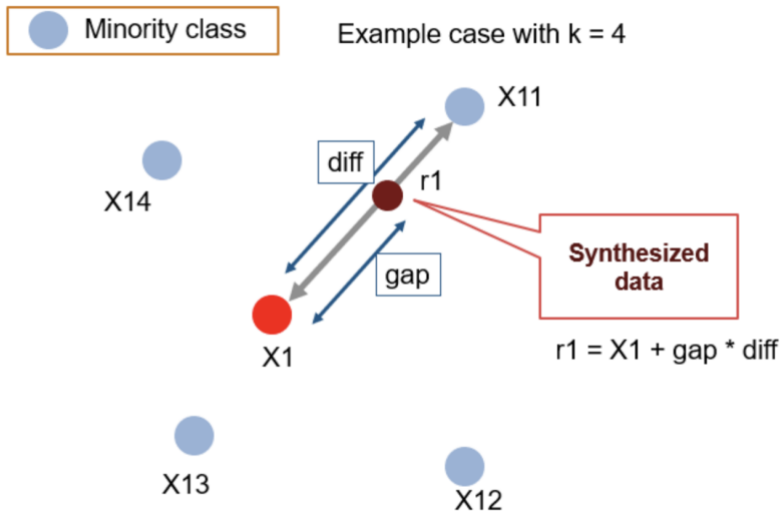

Figure 3-6: Working Procedure of SMOTE

Input:

- **X**: Feature matrix with shape (n_samples, n_features) containing minority and majority class instances.
- **y**: Target variable with shape (n_samples,) indicating the class labels.
- **k_neighbors**: Number of nearest neighbors to consider for synthetic sample generation.

Algorithm Steps:

1. **Identify Minority Instances:**
    - Identify instances belonging to the minority class.
2. **Calculate Synthetic Sample Count:**
    - Determine the number of synthetic samples to be generated for each minority instance. This can be user-defined or set as a multiple of the original minority class size.
3. **For Each Minority Instance:**
    - For each minority instance, find its k nearest neighbors within the minority class.
4. **Generate Synthetic Samples:**
    - For each minority instance, generate synthetic samples along the line segments connecting it to its k nearest neighbors.

- The synthetic samples are created by combining the minority instance with a randomly selected neighbor.

5. **Combine Synthetic Samples with Original Data:**
    - Combine the synthetic samples with the original minority class instances.
6. **Update Feature Matrix and Target Variable:**
    - Update the feature matrix **X** and target variable **y** to include the synthetic samples.

3.5.2 ADASYN (Adaptive Synthetic Sampling)

ADASYN, or Adaptive Synthetic Sampling, is an extension of SMOTE that aims to address some of its limitations. Like SMOTE, ADASYN focuses on generating synthetic samples for the minority class but adapts its approach to the local distribution of the minority instances.

ADASYN assesses the density distribution of the minority class instances locally. Instances that are more challenging to learn are given higher importance in synthetic sample generation. ADASYN adapts the level of oversampling dynamically based on the density of the minority class instances. It generates more synthetic samples for instances in areas with fewer minority class examples.

It was designed to adaptively generate synthetic samples based on the local distribution of the minority class. ADASYN aims to address the limitation of SMOTE, where the same number of synthetic samples is generated for all minority instances, regardless of their difficulty level.

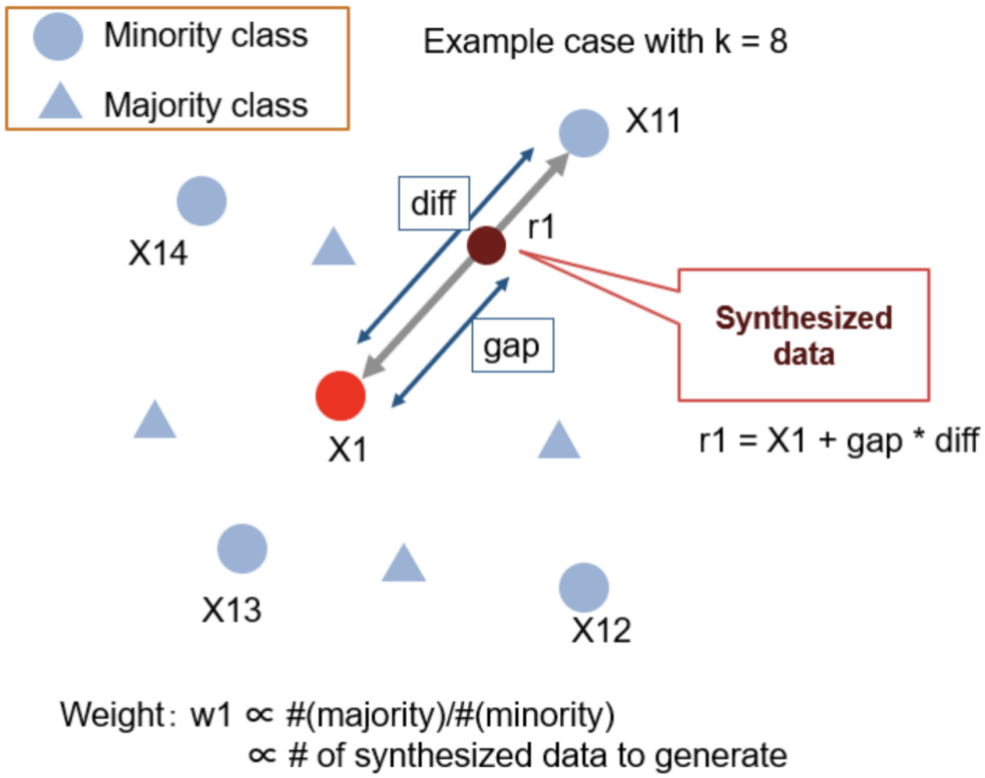

Figure 3-7: Working Procedure of ADASYN

Input:

X: Feature matrix with shape (n_samples, n_features) containing minority and majority class instances.

y: Target variable with shape (n_samples,) indicating the class labels.

n_neighbors: Number of nearest neighbors to consider for density estimation.

beta: Tuning parameter to control the level of adaptive oversampling.

Algorithm Steps:

1. **Identify Minority Instances:**
    - Identify instances belonging to the minority class.
2. **Calculate Density Ratio:**
    - For each minority instance, calculate the density ratio, which represents the ratio of minority class instances to majority class instances among its k nearest neighbors.
3. **Calculate Synthetic Sample Count:**
    - Determine the number of synthetic samples to be generated for each minority instance. This is proportional to the density ratio.

4. **For Each Minority Instance:**
    - For each minority instance, find its k nearest neighbors within the minority and majority classes.
5. **Generate Synthetic Samples:**
    - For each minority instance, generate synthetic samples along the line segments connecting it to its k nearest neighbors. The number of synthetic samples is determined by the density ratio.
6. **Combine Synthetic Samples with Original Data:**
    - Combine the synthetic samples with the original minority class instances.
7. **Update Feature Matrix and Target Variable:**
    - Update the feature matrix **X** and target variable **y** to include the synthetic samples.

3.6 Performance Evaluation Metrics

3.6.1 K-Fold Cross Validation

Cross-validation is a statistical method used to estimate the skill of machine learning models. K-fold cross validation is a procedure used to estimate the skill of the model on new data. To achieve K-Fold Cross Validation, dataset have to split into three sets, Training, Testing, and Validation. Based on the K value, the data set would be divided, and train/testing will be conducted in a sequence way equal to K time.

During K-fold cross validation, Accuracy, Precision, F1-Score will be evaluated.

3.6.2 Confusion Matrix

A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class.

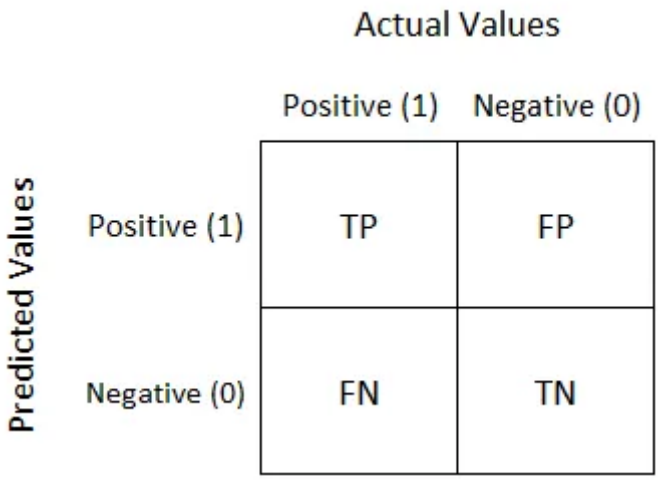

Figure 3-8: Confusion Matrix

It comprises 4 different parts naming:

- **True Positive (TP):** It is the sum of all actually positive records + the records our machine learning model predicted positive.
- **True Negative (TN):** Likewise, it is the sum of all records of the actual negative + model predicted it negative.
- **False Negative (FN or Type II error):** It is the sum of all records which actually positive, but our model falsely predicted negative.
- **False Positive (FP or Type I Error):** Similarly, it is the sum of all records which model predicted positive but actually it belongs to negative class.

We will evaluate accuracy, precision, error rate, sensitivity, specificity, F1 Score using confusion matrix as:

- Accuracy= (TP+TN)/Total
- Precision= TP/(TP+FP)
- Error Rate= 1-Accuracy
- Sensitivity=TP/(TP+FN)
- Specificity=TN/(TN+FP)
- F1 Score= 2*((Precision * Sensitivity) / (Precision + Sensitivity)

Chapter 4: IMPLEMENTATION DETAILS

4.1 Unbalanced Dataset

The two machine learning model were implemented on a dataset that initially presented a significant imbalance in the distribution of the target variable. In the presence of an unbalanced dataset, the model's ability to discern patterns across the minority class could be substantially compromised. This concern was particularly acute in the context of the classification task at hand, where the predictive accuracy across both classes was of paramount importance.

During the implementation, the model was first trained on this unbalanced dataset without any preliminary balancing techniques applied. This was done to establish a baseline performance, against which the effectiveness of subsequent balancing interventions could be measured. It was observed that the model, when trained on this unbalanced data, demonstrated a propensity towards higher predictive performance for the majority class while exhibiting suboptimal performance for the minority class.

The training process followed conventional practices. The model parameters were initialized without any bias towards either class, and the learning algorithm processed the input features and target labels in their original, skewed distribution. This preliminary phase was critical in demonstrating the inherent challenges associated with training on an unbalanced dataset and underscored the need for employing strategies to mitigate these issues.

Following the training phase, the model was evaluated using standard metrics. However, these metrics revealed that the high accuracy rate was largely reflective of the model's success in predicting the majority class, with a considerable decline in accuracy for the minority class instances. This outcome validated the initial concerns and set the stage for exploring advanced techniques that could enable the

model to treat both classes equitably, thereby improving the overall model performance.

4.2 SMOTE Technique

Prior to the application of the SMOTE (Synthetic Minority Over-sampling Technique) balancing technique, the training set was characterized by an imbalance in the distribution of the target classes. The 'Approved' class encompassed 2,120 instances, whereas the 'Rejected' class was represented by 1,295 instances, indicating a substantial disparity where the 'Approved' class outnumbered the 'Rejected' class.

In an effort to rectify this imbalance, the SMOTE algorithm was employed. SMOTE is an over-sampling approach that synthesizes new, synthetic examples in the minority class by interpolating between existing ones. The algorithm operates by randomly picking a point from the minority class and computing the k-nearest neighbors for this point. The synthetic points are then added between the chosen point and its neighbors.

Figure 4-1: Dataset after SMOTE

Following the implementation of SMOTE, the class distribution in the training set achieved parity. The 'Rejected' class's count was augmented to equal the 'Approved' class, with both classes now comprising 2,120 instances each. This newly balanced class distribution was expected to enhance the model's ability to learn from an equal representation of both classes, thereby improving its generalization capabilities when making predictions on unseen data.

The bar chart above depicting the class distribution post-SMOTE implementation visually confirmed the successful balancing of classes, as evidenced by the equal heights of the bars corresponding to both the 'Approved' and 'Rejected' classes.

4.3 ADASYN Technique

Prior to the invocation of the ADASYN (Adaptive Synthetic Sampling) technique, an evident imbalance was observed within the training set regarding the classification labels. The majority class, labeled as 'Approved', contained 2,120 instances, while the minority class, labeled as 'Rejected', comprised only 1,295 instances. This discrepancy highlighted a skewness towards the 'Approved' class, which could potentially lead to a bias in the predictive model towards the more prevalent class.

To address this issue, the ADASYN algorithm was applied to the dataset. ADASYN is an oversampling technique that aims to adaptively generate synthetic samples for the minority class. Unlike SMOTE, which generates the same number of synthetic samples for each minority class instance, ADASYN focuses on generating more synthetic data for those instances that are difficult to learn. The method involves creating synthetic data points based on the density distribution of samples. For each minority class sample, it computes the k-nearest neighbors and generates synthetic data points proportionally to the number of neighbors that belong to the majority class.

Figure 4-2: Dataset after ADASYN

The application of ADASYN resulted in an increase in the 'Rejected' class count, bringing the total to 2,204 instances, as opposed to the unchanged count of 2,120 in the 'Approved' class. The outcome was a more balanced dataset, with the minority class being better represented. The new synthetic points generated by ADASYN not only aimed to balance class distribution but also to emphasize the learning of those minority class instances that were more challenging to classify.

The modified class distribution was reflected in the visualization that followed the ADASYN application. The comparative height of the bars in the chart after applying ADASYN demonstrated a significant increase in the minority class, bringing the distribution closer to a balanced state, with 2,204 instances in the 'Rejected' class against the 2,120 instances in the 'Approved' class.

4.3.1 Adaptive Nature of Synthetic Sample Generation

Upon the completion of the ADASYN algorithm's execution, it was discerned that the quantity of synthetic samples generated did not result in precisely equal class proportions. This was attributed to the inherent nature of the ADASYN algorithm, which adaptively generates synthetic samples based on the density distribution of the minority class. The algorithm's design is such that it preferentially augments

areas where the minority class is underrepresented, particularly near the decision boundary where classification is most challenging. Consequently, the synthetic samples were not uniformly distributed across the minority class but were concentrated around the regions where the learning algorithm faced more difficulty distinguishing between classes.

The parameter sampling_strategy was set to a value of 1.0 with the intention of balancing the class distribution. However, the adaptive nature of ADASYN meant that the ultimate count of the minority class post-synthesis was contingent upon the data's complexity and the classifier's needs, rather than a fixed target ratio. The strategy was aimed at achieving balance, but the algorithm's internal dynamics allowed for flexibility in the final count of generated samples. This flexibility is central to ADASYN's approach, which prioritizes the enhancement of the classifier's performance in problematic regions over achieving exact numerical class balance.

The algorithmic process underpinning ADASYN meticulously considered the distributional characteristics of the data. It accounted for the intricacies within the minority class's distribution, focusing on areas with a high degree of class overlap or proximity to the majority class. Due to this approach, the number of synthetic samples introduced to achieve a balance was not predetermined but varied in accordance with the minority class's neighborhood characteristics. This variability is a testament to ADASYN's goal of reinforcing the classifier's ability to discern between classes in regions where the minority class is not well represented.

In the aftermath of applying the ADASYN technique, the resultant class distribution reflected a closer approximation to balance, with the minority class's count rising to 2,204, as opposed to the majority class's unchanged count of 2,120. Despite the minority class not achieving exact parity with the majority class, the synthetic sampling served its intended purpose. It equipped the classifier with a more representative training set, especially in the regions critical for accurate classification, thereby potentially enhancing the predictive performance across the entire feature space.

Chapter 5: RESULTS AND ANALYSIS

5.1 Unbalanced Dataset

5.1.1 Support Vector Machine

5.1.1.1 Hyperparameter Optimization

In the pursuit of optimizing the Support Vector Machine (SVM) for the loan prediction task, a comprehensive search across a predefined grid of hyperparameters was conducted. This grid encompassed various values for the regularization parameter 'C', the type of kernel, and the kernel coefficient 'gamma'. Specifically, the 'C' parameter was varied amongst [0.1, 1, 10], the 'kernel' parameter amongst ['linear', 'rbf'], and 'gamma' amongst ['scale', 'auto', 0.01, 1].

The optimization process leveraged the GridSearchCV method from the Scikit-learn library, which systematically worked through multiple combinations of parameter values, cross-validating as it went to determine which tune yielded the most effective model performance. The cross-validation was performed using a 5-fold strategy to ensure that the optimization was robust and not biased by the partitioning of the data.

Upon completion of the grid search, the optimal parameters were found to be 'C=10', 'kernel='rbf'', and 'gamma='auto''. The 'rbf' kernel, known for its ability to handle non-linear data, along with the higher 'C' value, suggested that the model required a greater penalty for misclassification to perform optimally. The choice of 'gamma='auto'' allowed the model to automatically adjust the kernel coefficient for the 'rbf' kernel based on the data features, which likely contributed to the model's ability to better capture the complexity of the decision boundary.

The hyperparameter tuning process was instrumental in enhancing the SVM model's capacity to predict loan defaults accurately. The selected hyperparameters were instrumental in refining the model's complexity, ensuring a delicate balance

between bias and variance, and ultimately yielding a model that was well-suited for the nuances of the dataset at hand.

5.1.1.2 Cross-Validation

In the evaluation of the Support Vector Machine (SVM) model, a K-Fold cross-validation approach was adopted, utilizing the KFold class from the Scikit-learn library. This technique involved partitioning the data into five distinct subsets, or 'folds', to validate the model's performance and to ensure that the assessment was not biased by any particular sample of the data.

The KFold cross-validation was initialized with the number of splits set to five, a random state for reproducibility set to 42, and shuffling enabled to guarantee randomness in the selection of data points for each fold. Throughout this process, the SVM model was trained and evaluated five separate times, with each fold serving once as the validation set while the remaining folds comprised the training set.

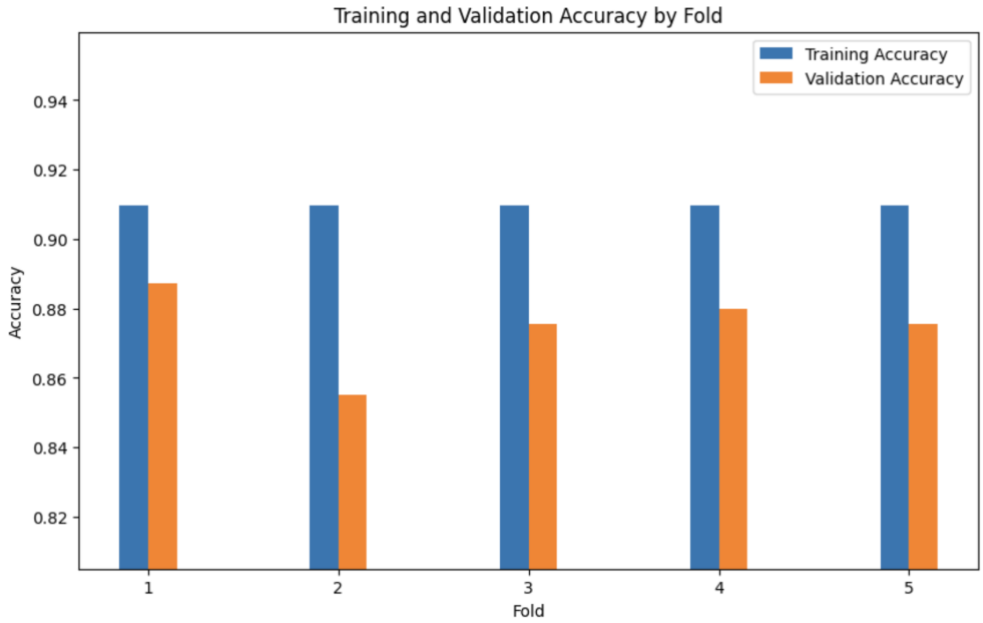

Figure 5-1: Cross Validation for SVM

The resulting graph, illustrating the training and validation accuracy for each fold, revealed that the model's training accuracy was consistently higher than the validation accuracy across all folds. This disparity between training and validation

performance suggested that the model might be overfitting to the training data to some extent—excelling at predictions on the data it had seen, but not generalizing quite as well to unseen data.

However, the validation accuracies were not drastically lower than the training accuracies, indicating that while the model may have been overfitting, it still maintained a reasonable level of predictive power on the validation sets. This was a crucial observation, as it meant that with some adjustments, the model had the potential to perform well on new, unseen data.

5.1.1.3 Model Evaluation

The evaluation of the Support Vector Machine (SVM) model was meticulously carried out by analyzing both the classification report and the Precision-Recall (PR) curve. The classification report provided a detailed summary of the model's performance metrics, while the PR curve offered a graphical representation of the model's trade-off between precision and recall for the positive class.

Table 5-1: Classification Report

	precision	recall	f1-score	support
0	0.88	0.94	0.91	536
1	0.89	0.79	0.84	318
accuracy			0.89	854
macro avg	0.89	0.87	0.88	854
weighted avg	0.89	0.89	0.89	854

The classification report revealed an accuracy of approximately 88.76%, indicating a high level of overall correctness in the model's predictions. Class-specific metrics were particularly insightful:

- For Class 0, presumed to be non-defaulting loans, the model achieved a precision of 0.88 and a recall of 0.94, resulting in an F1-score of 0.91. This high recall rate suggested that the model was particularly adept at identifying the majority of non-defaulting loans.

- Class 1, which likely represented defaulting loans, had a precision of 0.89 and a recall of 0.79, with an F1-score of 0.84. The slightly lower recall for this class indicated that there was some room for improvement in capturing all potential defaults.

As part of the evaluation process for the Support Vector Machine (SVM) model, the Precision-Recall Curve was plotted, and the Area Under the Curve (AUC) for this graph was calculated. The AUC associated with the Precision-Recall Curve was found to be 0.9256, indicating a high level of performance.

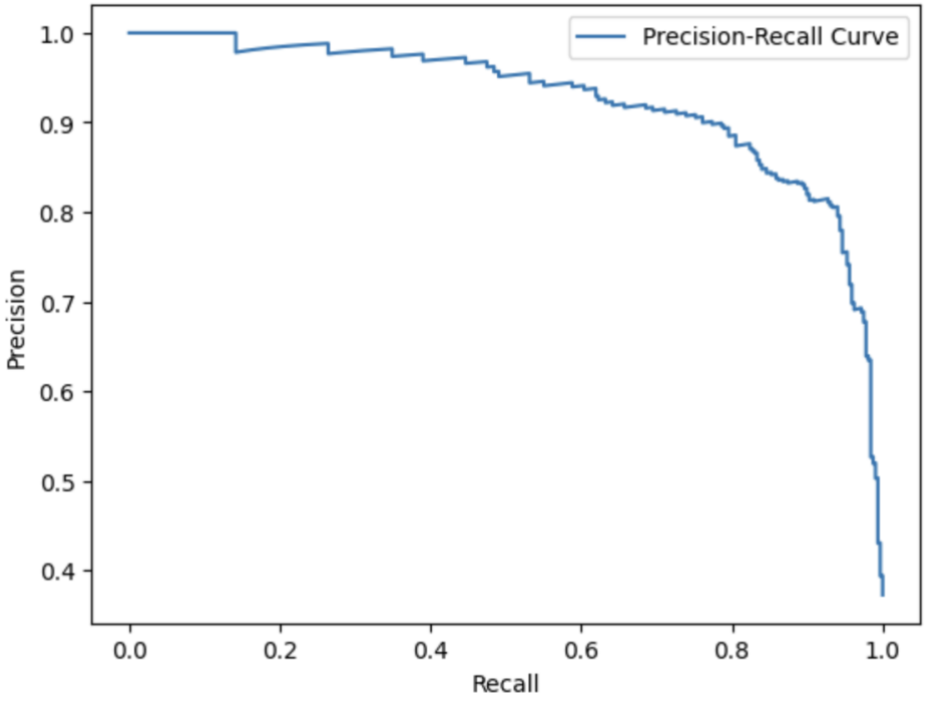

Figure 5-2: Precision-Recall Curve for SVM

The Precision-Recall Curve depicted a strong relationship between precision and recall, two critical measures of a model's predictive capabilities, particularly in scenarios with imbalanced classes. In the initial range of the curve, the model sustained a high precision while recall incrementally increased, illustrating the model's ability to correctly identify a high proportion of actual positives (defaulting loans) while minimizing the number of false positives (non-defaulting loans incorrectly classified as defaults).

As recall continued to rise towards 1.0, which corresponds to the model identifying all actual positives, a gradual decline in precision was observed. This is a typical behavior for precision-recall curves and represents the trade-off between capturing as many positives as possible and the increasing likelihood of false positives within those identified.

The high AUC value of approximately 0.93 reflected that the SVM model, overall, maintained a favorable balance between precision and recall across various threshold levels. This balance is particularly valuable in the context of loan default prediction, where the ability to detect as many defaults as possible (high recall) while maintaining a high confidence in these predictions (high precision) is essential to minimize risk and financial loss.

5.1.1.4 Confusion Matrix Analysis

The confusion matrix for the Support Vector Machine (SVM) model provided a quantifiable breakdown of the model's predictions compared to the true labels. The analysis of this matrix was integral to understanding the model's performance in classifying loans as 'Approved' or 'Rejected'.

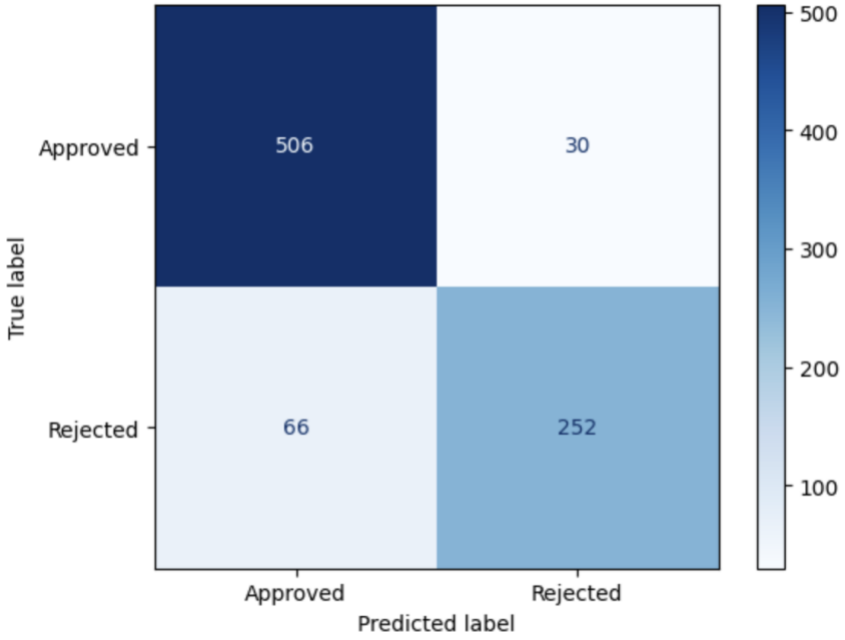

Figure 5-3: Confusion Matrix for SVM

Upon examination, the confusion matrix::The model accurately predicted 'Approved' loans 506 times, which represented the true positives for the non-defaulting class.For loans that were actually 'Rejected' (defaulting), the model correctly predicted 252 cases, indicating the true positives for the defaulting class.
However, there were instances where the model predicted loans as 'Approved' when they were actually 'Rejected', amounting to 66 cases. These were the false negatives and represented missed opportunities for identifying potential defaults.The model also predicted 30 loans as 'Rejected' which were actually 'Approved', classified as false positives. These instances represented potential lost opportunities where the model incorrectly identified a loan as a default risk.
These findings underscored the model's strengths and weaknesses. The high number of true positives for 'Approved' loans suggested that the model was quite effective at identifying loans that were not at risk of defaulting. Meanwhile, the respectable amount of true positives for 'Rejected' loans indicated that the model could identify a significant number of high-risk loans, albeit with some room for improvement given the false negatives.

The false negative, while fewer than the true positives for the 'Rejected' class, were particularly concerning given the objective to prevent loan defaults. These represented cases where the model's predictions could potentially lead to financial losses. The false positives, although not financially detrimental, could result in lost revenue opportunities due to unnecessary loan rejections.

In conclusion, the confusion matrix analysis revealed that while the SVM model was quite adept at identifying non-defaulting loans, it also highlighted the need for further refinement to reduce the number of false negatives, thereby enhancing the model's predictive accuracy for loan defaults.

5.1.2 Logistic Regression

5.1.2.1 Hyperparameter Optimization

The Logistic Regression model underwent a systematic hyperparameter tuning process using GridSearchCV, with the goal of optimizing the model for the task of loan default prediction. The GridSearchCV function from Scikit-learn was configured to explore a comprehensive grid of hyperparameters, which included:

- Regularization strength **C**, tested across a broad range of values: [0.01, 0.1, 0.5, 1, 2, 10, 100].

- The penalty parameter, which dictates the norm used in the penalization, with candidates: ['l1', 'l2'].

- The solver used in the optimization problem, with options: ['lbfgs', 'liblinear', 'newton-cg', 'newton-cholesky'].

This grid was chosen to cover a spectrum of model complexities—from underfitting to overfitting scenarios—and to identify the most suitable combination of parameters for the Logistic Regression model. A five-fold cross-validation strategy was employed during the grid search to ensure that the optimization was not biased toward a specific subset of the dataset.

The max_iter parameter was set to 1000 to give the optimization algorithm sufficient iterations to converge to the best solution, especially for more complex models with larger datasets or those requiring more iterations due to the complexity of the penalty.

Upon completion of the hyperparameter tuning, the optimal parameters were identified as C=0.01, employing an 'l1' penalty, and utilizing the 'liblinear' solver. The 'l1' penalty, also known as Lasso regularization, has the capability to drive some coefficients to zero, thus performing feature selection and resulting in a sparse

model. The 'liblinear' solver is particularly well-suited for small datasets and for optimization problems with 'l1' penalty.

Selecting C=0.01 indicated that a stronger regularization was preferred, which helped to prevent overfitting and to create a model that generalized better to unseen data. The combination of these parameters suggested that the final model aimed to strike a balance between maintaining model simplicity and ensuring adequate learning from the data.

The outcome of this hyperparameter tuning phase was a Logistic Regression model that was tailored to the characteristics of the dataset, with an enhanced ability to predict loan defaults while minimizing the risk of overfitting.

5.1.2.2 Cross-Validation

The Logistic Regression (LR) model's cross-validation was executed with the intent to authenticate the model's predictive stability and generalizability. For this purpose, the KFold class from Scikit-learn was employed, establishing a five-fold cross-validation framework. The n_splits parameter was set to 5 to divide the data into equal partitions, with random_state as 42 ensuring reproducibility of the results, and shuffle set to True to randomize the data points prior to the split, thus mitigating any potential bias due to the original ordering of the data.

Throughout the cross-validation process, the Logistic Regression model was trained and validated across five distinct subsets of the data. This ensured that each segment of the data was used for both training and validation, providing a comprehensive evaluation of the model's performance.

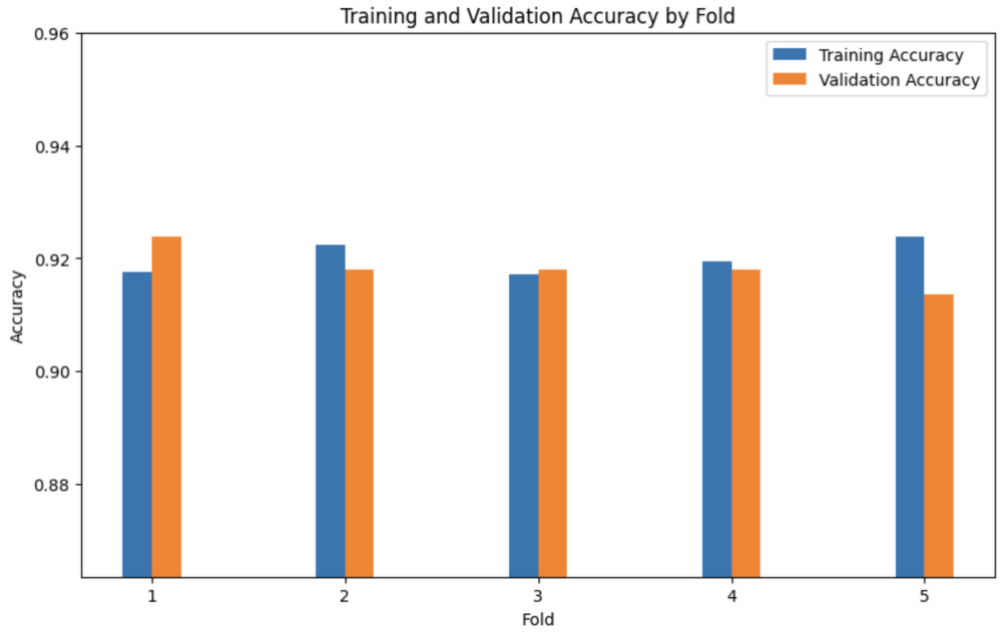

Figure 5-4: Cross validation result for Logistic Regression

The graph generated from the cross-validation exhibited training and validation accuracies that were notably similar for each fold. This close alignment between training and validation performance indicated a well-fitting model with good generalization properties. The slight variations in accuracy across the folds were minimal, suggesting that the model was not overly sensitive to the specific data on which it was trained and that it was stable across different data samples.

In the context of predicting loan defaults, the close correspondence between the training and validation accuracy scores, which consistently hovered around the same values, was indicative of the model's reliability. This uniformity in performance affirmed the robustness of the Logistic Regression model, suggesting that it was capable of delivering reliable predictions when applied to unseen data.

The similarity in accuracy between the training and validation sets across all folds provided evidence against significant overfitting or underfitting. The Logistic Regression model demonstrated an ability to learn from the training data without being unduly influenced by noise, thereby ensuring that its predictions were based on the underlying patterns in the data that were relevant to the task of loan default prediction.

5.1.2.3 Model Evaluation

Table 5-2: Classification Report for Logistic Regression

	precision	recall	f1-score	support
0	0.92	0.93	0.92	536
1	0.88	0.86	0.87	318
accuracy			0.90	854
macro avg	0.90	0.90	0.90	854
weighted avg	0.90	0.90	0.90	854

The classification report showed that the model achieved an overall accuracy of approximately 90.39%. This high level of accuracy was indicative of the model's adeptness in correctly identifying the majority of the outcomes.

- Class 0 (presumed to represent 'Approved' loans) had a precision of 0.92 and a recall of 0.93, yielding an F1-score of 0.92. The model was notably proficient at identifying true non-defaulting loans and had a high probability of correctly classifying loans that would not default.

- Class 1 (presumably corresponding to 'Rejected' or defaulting loans) exhibited a precision of 0.88 and a recall of 0.86, with an F1-score of 0.87. The model was quite effective at predicting defaults, with a substantial proportion of the positive predictions indeed corresponding to defaults and most actual defaults being correctly identified.

The macro and weighted averages for precision, recall, and F1-score were all around 0.90, reflecting a balanced model performance across both classes.

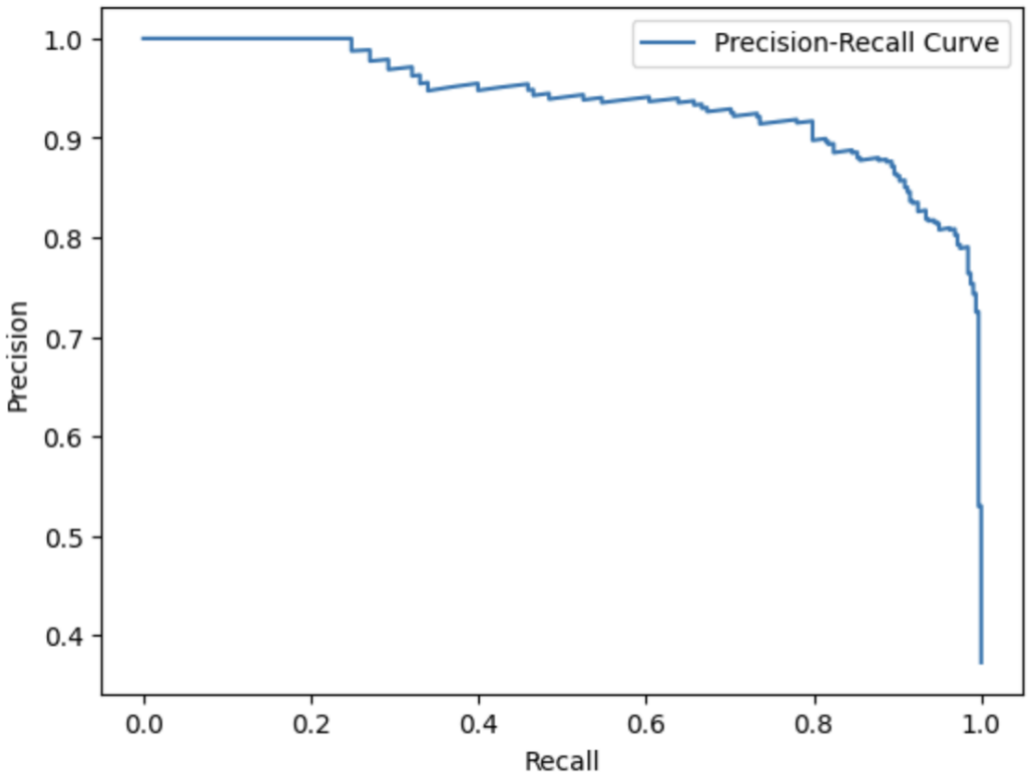

Figure 5-5: Precision Recall Curve for Logistic Regression

The Precision-Recall Curve was accompanied by an AUC of 0.9370, signifying the model's excellent capability in discriminating between the classes. The curve began with high precision at lower recall levels, indicating that when the model predicted a default, it did so with high confidence. As recall increased, a gradual decrease in precision occurred, which is typical as the model started to predict more loans as likely to default, including some false positives.

The curve's high AUC suggested that the Logistic Regression model effectively balanced precision and recall across different thresholds. This balance is particularly vital in loan default prediction, where predicting defaults accurately (high recall) without misclassifying too many non-defaults as defaults (high precision) is crucial for minimizing financial risk.

5.1.2.4 Confusion Matrix Analysis

The Logistic Regression model's predictive accuracy was further scrutinized through an analysis of the confusion matrix, which provided a clear visual representation of the model's classification performance.

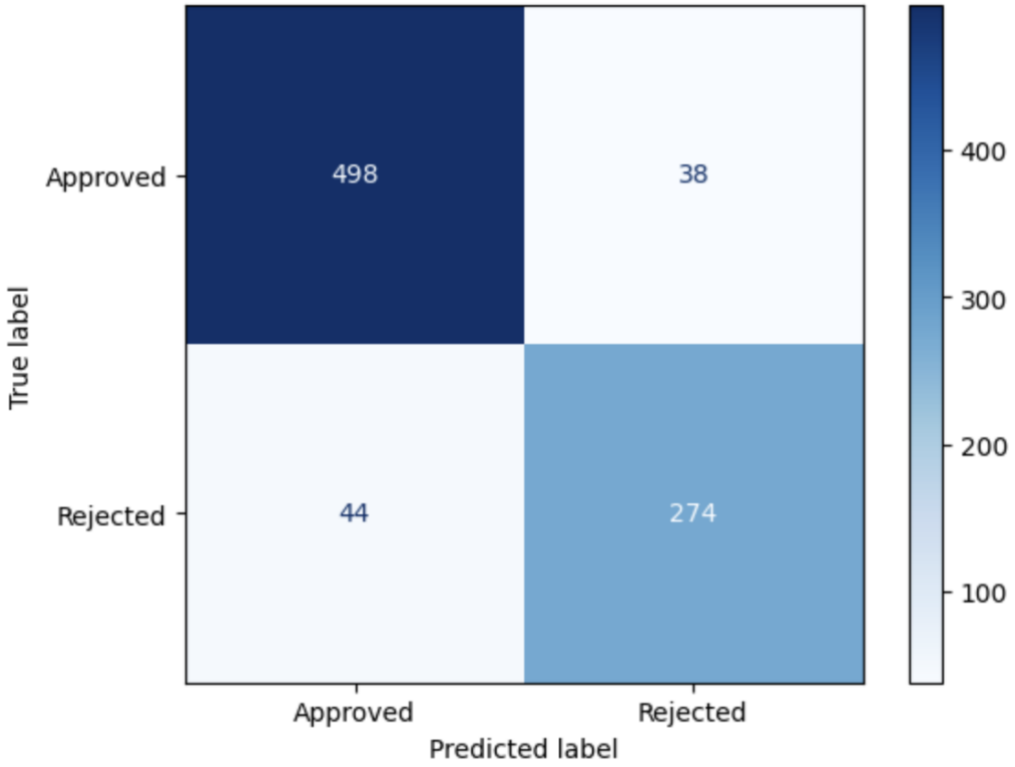

Figure 5-6: Confusion Matrix for Logistic Regression

- **True Positives (TP)**: The model correctly identified 498 cases as 'Approved', which indicated that it was highly effective at recognizing loan applications that were not at risk of defaulting.
- **True Negatives (TN)**: It accurately predicted 274 cases as 'Rejected', demonstrating its capability to identify a substantial number of loan applications that were at risk of defaulting.
- **False Positives (FP)**: There were 38 instances where the model incorrectly labeled loan applications as 'Rejected' when they were actually 'Approved'. These errors represent cases where potentially viable loan applications were mistaken for defaults, which could lead to lost opportunities for the lender.
- **False Negatives (FN)**: The model labeled 44 loan applications as 'Approved' that were actually 'Rejected'. These instances were particularly

critical since they represent loans that were likely to default but were not identified as such, thus posing a financial risk.

The confusion matrix's diagonal cells (TP and TN) contained the majority of the predictions, illustrating the model's strong predictive power. However, the presence of false positives and false negatives provided valuable insights into the types of errors the model was prone to, and the potential impact of these errors on lending decisions.

The matrix highlighted the model's strength in correctly classifying both non-defaulting and defaulting loans while also revealing areas for potential improvement. Specifically, the number of false negatives pointed to opportunities to enhance the model's sensitivity to default risk, thereby reducing the likelihood of approving risky loans. Conversely, the false positives indicated a chance to refine the model's specificity to avoid rejecting safe loans.

In conclusion, the analysis of the confusion matrix confirmed the Logistic Regression model's competency in distinguishing between 'Approved' and 'Rejected' loan applications, with a strong overall accuracy. Nevertheless, it also underscored the importance of further model optimization to minimize the financial implications associated with incorrect loan approval or rejection decisions.

Upon completion of the training and validation process, the Support Vector Machine (SVM) and Logistic Regression models were compared to determine which was more effective for predicting loan defaults.

- **Accuracy**: Both models demonstrated high overall accuracy, with the SVM achieving approximately 88.75% and Logistic Regression slightly higher at around 90.51%. While accuracy is a useful metric, it does not always provide the full picture, especially in the context of imbalanced datasets where the cost of false negatives can be high.
- **Precision and Recall**: The SVM model showed strong precision and recall values, but Logistic Regression slightly outperformed SVM in terms of recall for the defaulting class. This is critical in loan default prediction,

where identifying as many actual defaults as possible (high recall) is often more important than the precision of those predictions.

- **F1-Score**: The balance between precision and recall, as measured by the F1-score, was also important. The SVM had an F1-score of 0.84 for the defaulting class, while Logistic Regression showed a slightly better F1-score of 0.87 for the same class, indicating a better balance between precision and recall.
- **AUC-PR**: The AUC for the Precision-Recall Curve is a crucial metric when dealing with imbalanced classes. The SVM had an AUC-PR of 0.925, while the Logistic Regression model had an AUC-PR of 0.937. This suggests that the Logistic Regression model had a better performance in distinguishing between the positive (default) and negative (non-default) classes across all thresholds.
- **Confusion Matrix**: Analysis of the confusion matrices for both models provided insights into their predictive capabilities. The Logistic Regression model had fewer false negatives compared to the SVM, which is favorable for the goal of minimizing financial risk due to loan defaults.

In conclusion, while both models performed well, Logistic Regression showed a slight edge over the SVM in terms of recall and F1-score for the defaulting class, as well as a higher AUC-PR score.Here the cost of false negatives is higher (i.e., failing to predict a default is costlier than incorrectly predicting a default), the higher recall of the Logistic Regression model would likely make it the preferred choice.

Chapter 6: EPILOGUE

6.1 Remaining tasks

The research of creating an effective predictive model for loan defaults does not conclude with the initial model evaluation. Several tasks remain to refine and enhance the model's performance further. The epilogue of this report outlines the steps that are yet to be undertaken.

6.1.1 Model Training on Balanced Dataset

The initial model training was conducted on an unbalanced dataset, which may have introduced biases towards the majority class. To rectify this and potentially improve the model's performance, the subsequent step involves training the model on a balanced dataset. This task will include applying techniques such as SMOTE or ADASYN to create synthetic samples of the minority class, thus ensuring an equal representation of both classes. The model will then be retrained using this newly balanced dataset, with the expectation that the balance will enable the model to learn more generalizable patterns, especially for the minority class.

6.1.2 Comparision of Balancing Techniques

There are multiple techniques available for balancing datasets, and each has its nuances and effects on model training. A comparative study will be conducted to assess the effectiveness of different balancing techniques. This comparison will not only consider the resulting class distributions but will also evaluate how each technique affects the model's ability to generalize its predictions to unseen data. Metrics such as precision, recall, and the F1-score for the minority class will be of particular interest in this comparison.

6.1.3 Selection of Best Balancing Technique

Following the comparison of different balancing techniques, a decision will be made on which balanced dataset yields the best model performance. The selection

criteria will consider several factors, including but not limited to, the model's accuracy, precision-recall balance, and the overall AUC-PR score. The dataset that allows the model to achieve the highest level of performance across these metrics will be chosen for final model training and deployment. The selection will be based on the model's efficacy in predicting loan defaults, ensuring that the most predictive and least biased model is utilized for decision-making.

Chapter 7: References

[1] Andrew McCallum, A. McCallum, Kamal Nigam, and K. Nigam, "A comparison of event models for naive bayes text classification," *AAAI Conference on Artificial Intelligence*, pp. 41–48, Jan. 1998.

[2] Nitesh V. Chawla *et al.*, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 321–357, Jan. 2002, doi: 10.1613/jair.953.

[3] Haibo He *et al.*, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," *IEEE World Congress on Computational Intelligence*, pp. 1322–1328, Jun. 2008, doi: 10.1109/ijcnn.2008.4633969.

[4] V. K. Chauhan *et al.*, "Problem formulations and solvers in linear SVM: a review," *Artificial Intelligence Review*, vol. 52, no. 2, pp. 803–855, Aug. 2019, doi: 10.1007/s10462-018-9614-6.

[5] Mirza Muntasir Nishat *et al.*, "A Comprehensive Investigation of the Performances of Different Machine Learning Classifiers with SMOTE-ENN Oversampling Technique and Hyperparameter Optimization for Imbalanced Heart Failure Dataset," *Scientific Programming*, vol. 2022, pp. 1–17, Mar. 2022, doi: 10.1155/2022/3649406.

[6] Ugochukwu Orji *et al.*, "Machine Learning Models for Predicting Bank Loan Eligibility," *2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON)*, Apr. 2022, doi: 10.1109/nigercon54645.2022.9803172.

[7] Anthony Anggrawan, Hairani Hairani, and Christofer Satria, "Improving SVM Classification Performance on Unbalanced Student Graduation Time Data Using SMOTE," *International Journal of Information and Education Technology*, vol. 13, no. 2, pp. 289–295, Jan. 2023, doi: 10.18178/ijiet.2023.13.2.1806.

[8] Dina Elreedy, Amir F. Atiya, and Firuz Kamalov, "A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning," *Machine-mediated learning*, Jan. 2023, doi: 10.1007/s10994-022-06296-4.

[9] Tarid Wongvorachan, Surina He, and Okan Bulut, “A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining,” *Inf.*, vol. 14, no. 1, pp. 54–54, Jan. 2023, doi: 10.3390/info14010054.

[10] Ishani Dey and Vibha Pratap, “A Comparative Study of SMOTE, Borderline-SMOTE, and ADASYN Oversampling Techniques using Different Classifiers,” *2023 3rd International Conference on Smart Data Intelligence (ICSMDI)*, Mar. 2023, doi: 10.1109/icsmdi57622.2023.00060.

[11] Robert C. Moore, Daniel P. W. Ellis, Eduardo Fonseca, Shawn Hershey, Aren Jansen, and Manoj Plakal, “Dataset Balancing Can Hurt Model Performance,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Jun. 2023, doi: 10.1109/icassp49357.2023.10095255.

[12] Archana Archana, “A Comparison of Various Machine Learning Algorithms and Deep Learning Algorithms for Prediction of Loan Eligibility,” *International Journal for Research in Applied Science and Engineering Technology*, vol. 11, no. 6, pp. 4558–4564, Jun. 2023, doi: 10.22214/ijraset.2023.54495.

[13] V. V, R. A. C, V. K N, and A. G, “Prediction of Loan Approval in Banks Using Machine Learning Approach.” Rochester, NY, Aug. 04, 2023. Accessed: Jan. 20, 2024. [Online]. Available: https://papers.ssrn.com/abstract=4532468