

**PURBANCHAL UNIVERSITY**

**KANTIPUR CITY COLLEGE (KCC)**

**Putalisadak, Kathmandu, Nepal**

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

Reg. No.

**A Proposal on**

**“A Comparative Analysis of Data Balancing Algorithms for Bank Loan Eligibility predictions system”**

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. To determine the most effective machine learning model for loan approval, various algorithms such as Logistic Regression, Naive Bayes and Support Vector Machines (SVM) will be compared. Each algorithm 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: Loan, Logistic Regression, Explainable, ML, Naive Bayes, Prediction, Random Forest, SVM*

Table of Contents

[ABSTRACT ii](#_Toc156753056)

[LIST OF FIGURES iii](#_Toc156753057)

[Chapter 1: INTRODUCTION 1](#_Toc156753058)

[1.1 Background 1](#_Toc156753059)

[1.2 Statement of Problem 2](#_Toc156753060)

[1.3 Research Objectives 2](#_Toc156753061)

[Chapter 2: LITERATURE REVIEW 3](#_Toc156753062)

[Chapter 3: RESEARCH METHODOLOGY 4](#_Toc156753063)

[3.1 Methods of Data Collection 4](#_Toc156753064)

[3.2 Data Preprocessing 5](#_Toc156753065)

[3.3 Model Development 5](#_Toc156753066)

[3.3.1 Support Vector Machine 5](#_Toc156753067)

[3.3.2 Naïve Bayes 8](#_Toc156753068)

[3.3.3 Logistic Regression 9](#_Toc156753069)

[3.4 Balancing Techniques 10](#_Toc156753070)

[3.4.1 SMOTE (Synthetic Minority Over-sampling Technique) 11](#_Toc156753071)

[3.4.2 ADASYN (Adaptive Synthetic Sampling) 13](#_Toc156753072)

[Chapter 4: Expected Outcome 16](#_Toc156753073)

[Chapter 5: References 18](#_Toc156753074)

# LIST OF FIGURES

[Figure 3‑1: Dataset Visualization 5](#_Toc156754549)

[Figure 3‑2: SVM Hyperplane 8](#_Toc156754550)

[Figure 3‑3: SVM Model 9](#_Toc156754551)

[Figure 3‑4: Sigmoid Function for Logistic Regression 11](#_Toc156754552)

[Figure 3‑5: Working Procedure of SMOTE 13](#_Toc156754553)

[Figure 3‑6: Working Procedure of ADASYN 15](#_Toc156754554)

[Figure 3‑7: Confusion Matrix 18](#_Toc156754555)

# INTRODUCTION

## 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, decision trees, 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.

The primary objective of this research is to identify the most effective ML algorithm for loan approval by analyzing and comparing the performance of Logistic regression, Naïve Bayes, and SVM. Each algorithm brings a unique set of strengths to the table, ranging from interpretability to the ability to handle complex, nonlinear relationships within large datasets. Through an in-depth exploration of these algorithms and their application in the banking domain, this research aims to provide valuable insights into optimizing loan approval processes for financial institutions.

By leveraging historical data or simulated scenarios, the study will assess the performance metrics of each algorithm, including accuracy, precision, recall, and F1 score. The comparative analysis will not only shed light on the strengths and weaknesses of each model but will also assist banks in making informed decisions regarding the adoption of ML algorithms based on the specific characteristics of their datasets and institutional goals.

## Statement of Problem

The primary problem at hand is the need for financial institutions to identify the most suitable machine learning algorithm for loan approval. Logistic regression, Naïve Bayes, and Support Vector Machines (SVM) are prominent contenders, each with its own strengths and weaknesses. The challenge lies in determining which algorithm offers the optimal balance of accuracy, interpretability, and adaptability to the dynamic nature of financial markets.

Moreover, as the banking industry grapples with an increasing volume of diverse loan applicants, the current processes are perceived as notably burdensome. The time-consuming nature of manual evaluations and the potential for inaccuracies underscore the urgency for a more streamlined and automated approach to credit risk assessment. Consequently, there is a critical need for research that systematically compares the performance of different machine learning algorithms in the context of loan approval, aiming to provide insights into optimizing these processes and addressing the challenges faced by financial institutions

## Research Objectives

* To compare different balancing technique with the help of machine learning algorithms.
* To build a robust model for predicting eligibility of loan according to past data

# LITERATURE REVIEW

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

# RESEARCH METHODOLOGY

## Methods of Data Collection

For this research work, datasets are collected from Kaggle, [Loan approval analysis | Kaggle](https://www.kaggle.com/code/jayrdixit/loan-approval-analysis) 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.

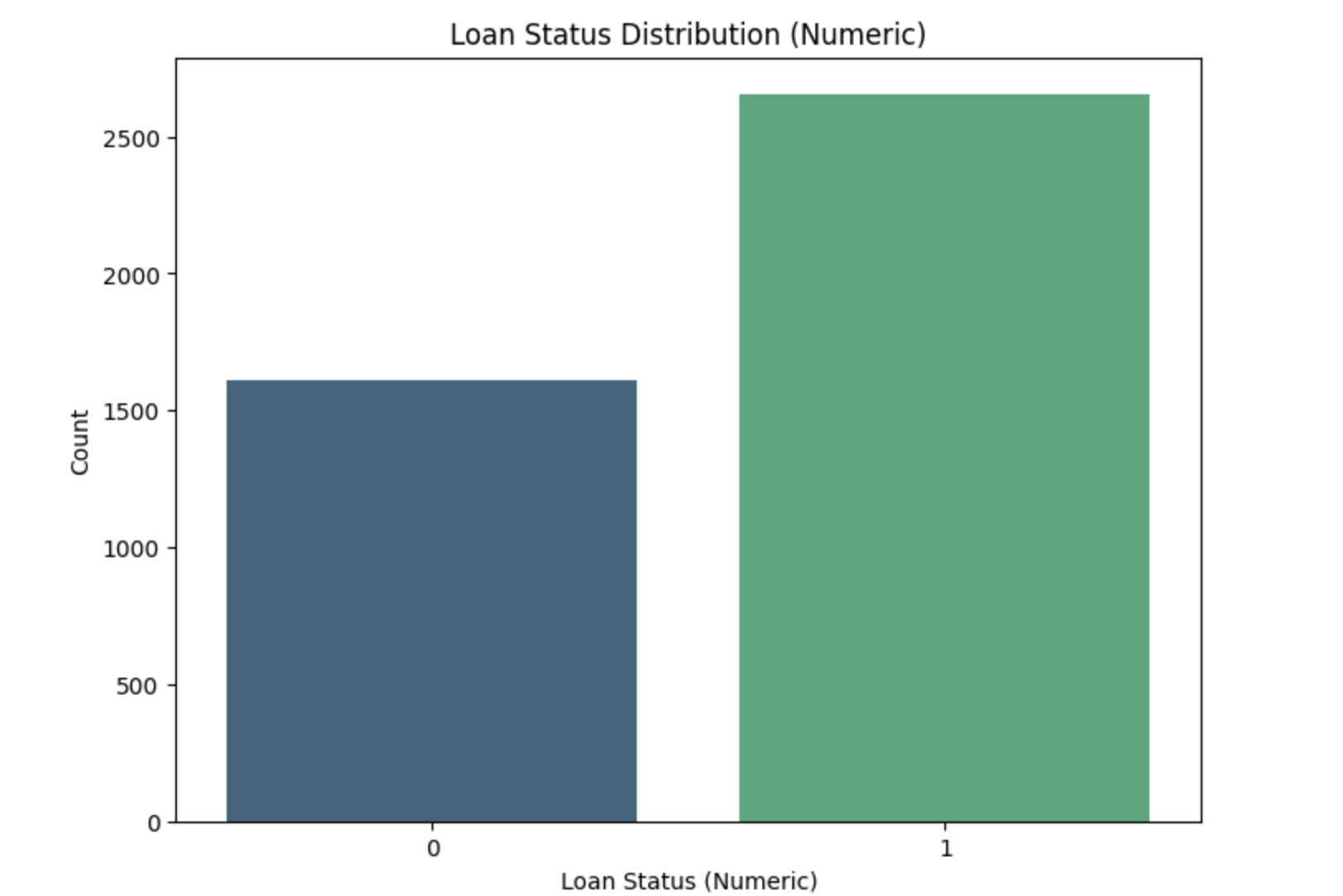


Figure 3‑1: Dataset Visualization

## Data Preprocessing

To prepare the data for modeling, the first step is to dropped the loan status column from the categorical data of each field and assign the binary labels for each values. The Correlation will be calculated to determine the similarities and dissimilarities between the categorical data. For the filling of missing values, the median will be calculated and assigned to those fields.

Once confirmed that there are no missing values, the entire dataset is split into two sets: training (4:5) and testing (1:5)

## Model Development

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

### 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………………………… (1)**

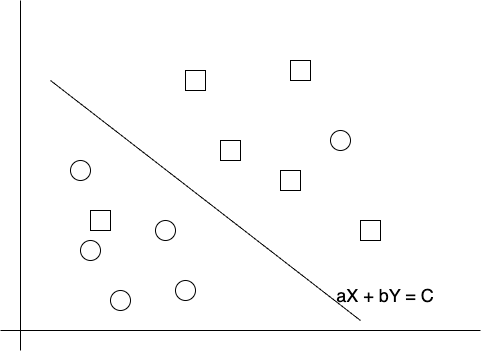


Figure 3‑2: SVM Hyperplane

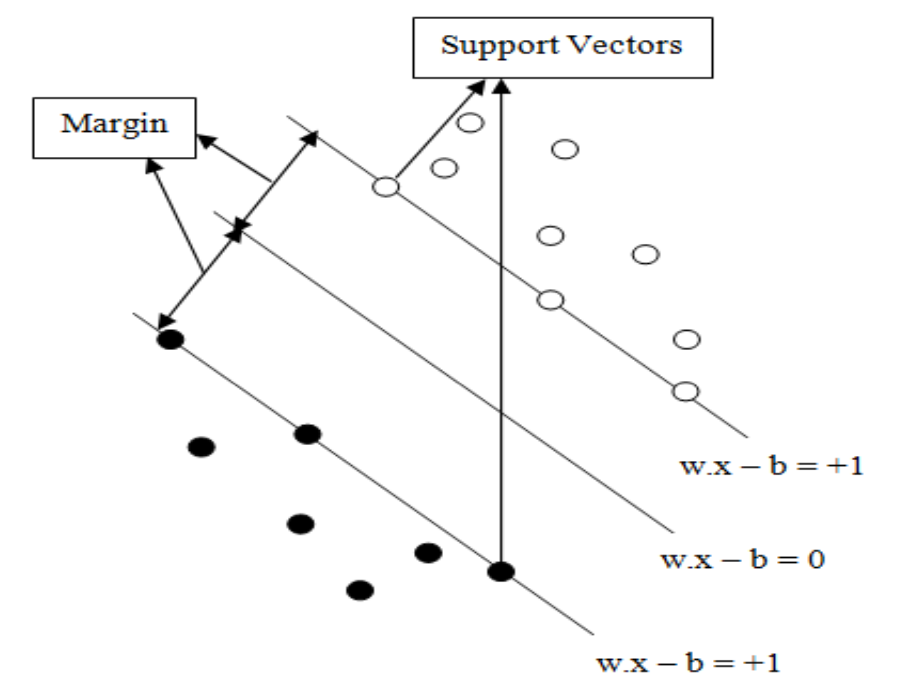


Figure 3‑3: SVM Model

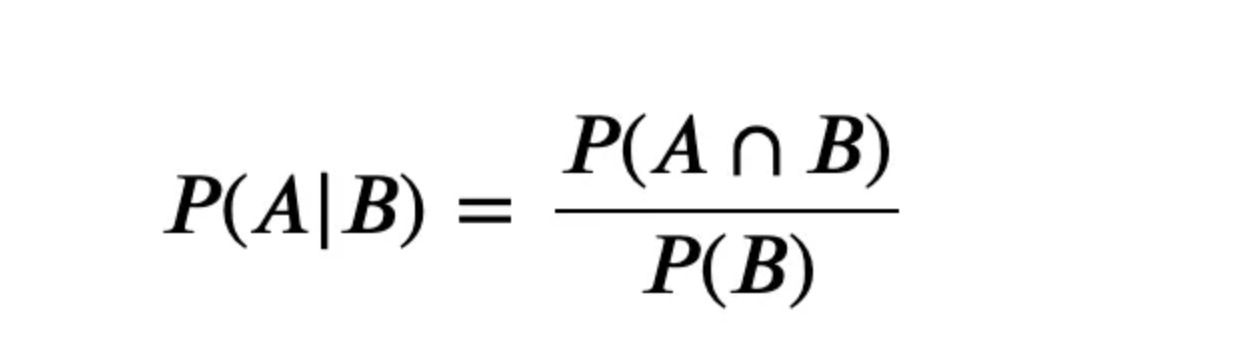
The figure 3.4 shows that the model consists of three different lines. These three lines construct the hyper plane that separates the given patterns and the pattern that lies on the edges of the hyper plane is called support vectors. The perpendicular distance between the line of margin and the edge 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.

### Naïve Bayes

Naive Bayes is a probabilistic machine learning algorithm based on Bayes' theorem, named after the 18th-century mathematician and philosopher Thomas Bayes. It is particularly popular for classification tasks, such as spam detection or sentiment analysis, including its application in credit risk assessment in the banking industry.

#### Bayes' Theorem

At the core of Naive Bayes is Bayes' theorem, which mathematically describes the probability of an event based on prior knowledge of conditions that might be related to that event. The formula for Bayes' theorem is as follows:



Where,

P(A|B) is the probability of event A given that event B has occurred.

P(B|A) is the probability of event B given that event A has occurred.

P(A) and P(B) are the probabilities of events A and B independently.

#### Naïve Assumption

The "naive" aspect of Naive Bayes comes from the assumption that features used to describe data are conditionally independent given the class label. In other words, the presence or absence of a particular feature does not influence the presence or absence of any other feature.

Naive Bayes is simple and computationally efficient. It requires less training data compared to other algorithms. It is robust to irrelevant features due to its conditional independence assumption.

### 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.

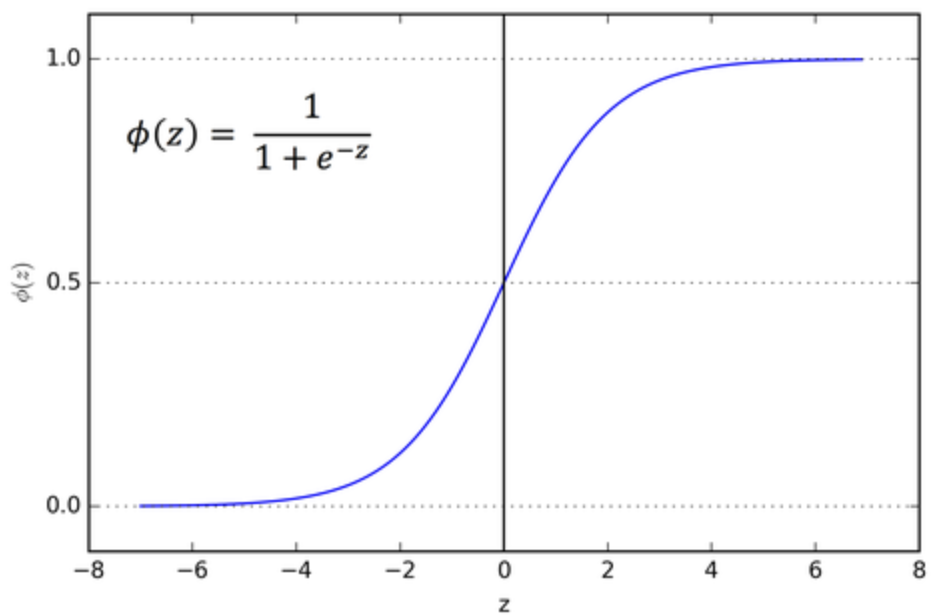


Figure 3‑4: Sigmoid Function for Logistic Regression

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.

## Balancing Techniques

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.

### 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.

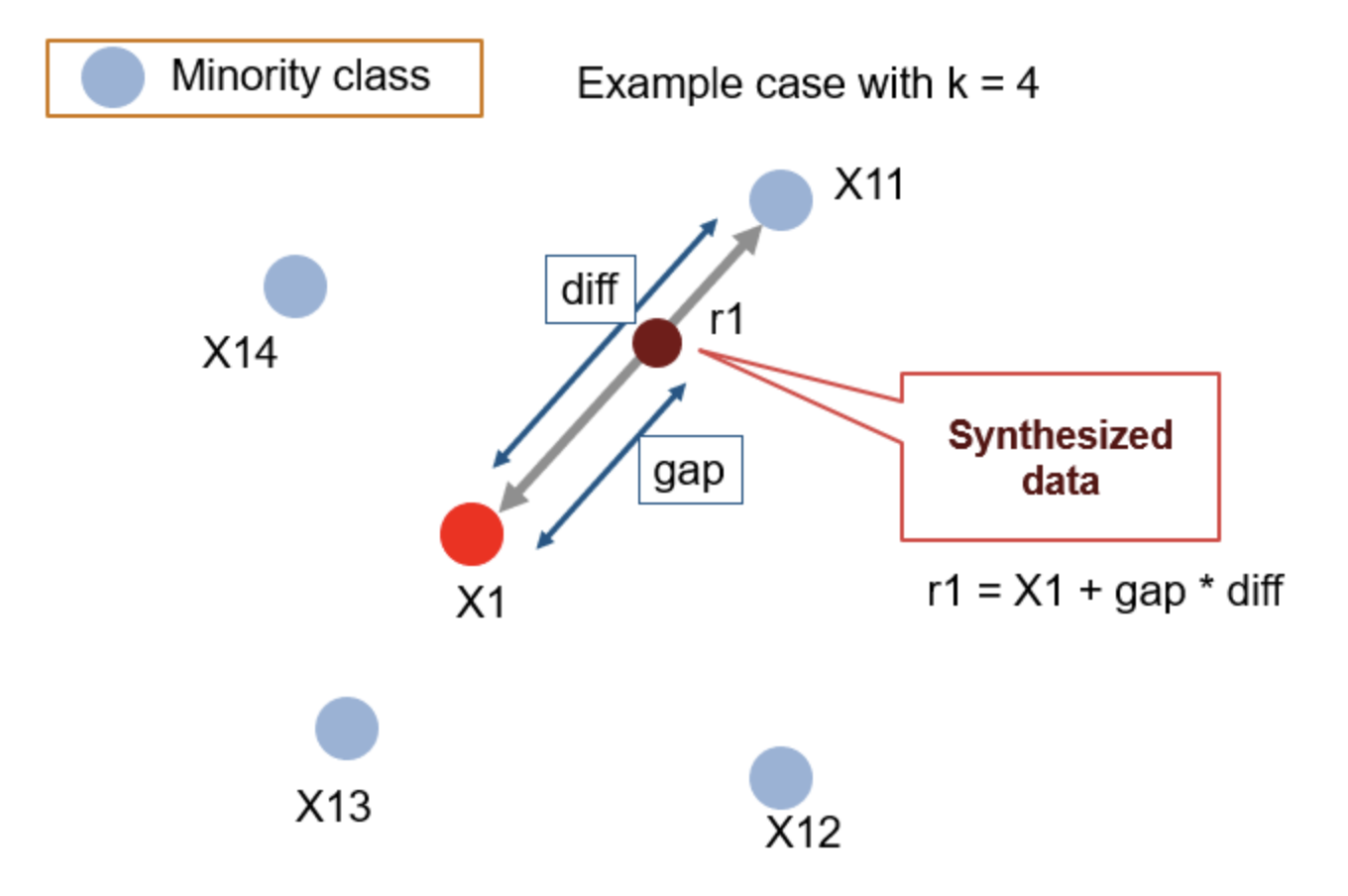


Figure 3‑5: Working Procedure of SMOTE

**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.

### 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.

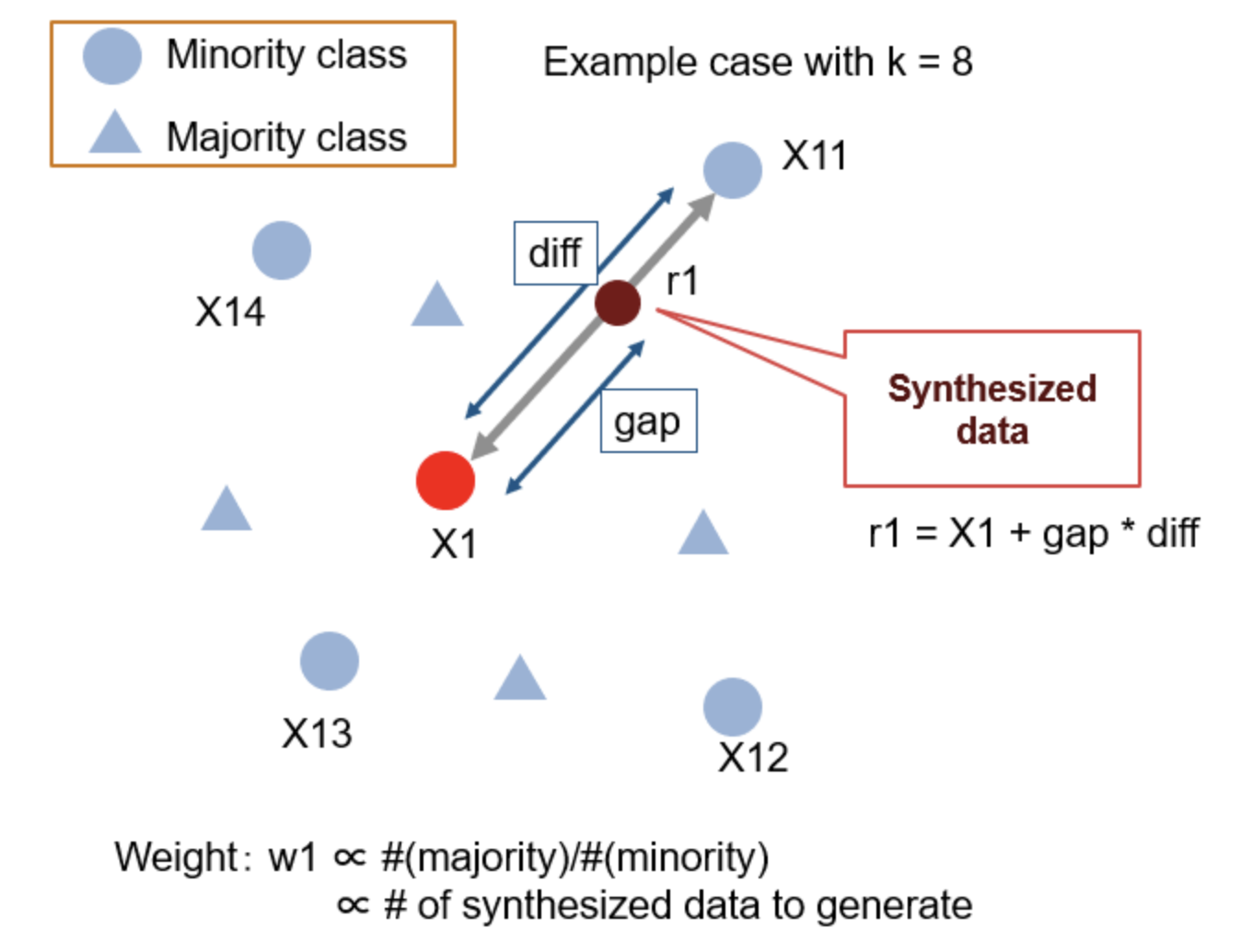


Figure 3‑6: Working Procedure of ADASYN

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.

## Performance Evaluation Metrics

### 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.

### 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.

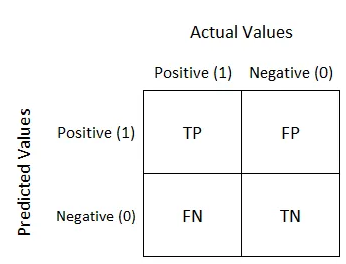


Figure 3‑7: Confusion Matrix

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))

# Expected Outcome

The anticipated outcome of this research is to comprehensively assess the impact of balancing techniques, specifically SMOTE and ADASYN, on predictive models used for bank loan eligibility decisions. The study will focus on three aspects: the effectiveness of these balancing techniques, the comparative performance of different predictive models, and the practical implications for financial decision-making in the context of class imbalance.

The first aim is to evaluate the effectiveness of SMOTE and ADASYN in enhancing the accuracy and reducing the bias in machine learning models, namely SVM, Naive Bayes, and Logistic Regression. This evaluation will provide insights into how these balancing techniques influence the performance of each model when applied to imbalanced datasets typical in bank loan eligibility assessments. The research aims to ascertain whether these techniques not only improve the predictive accuracy but also contribute to a more equitable decision-making process by mitigating bias.

The second aim of the study is to determine the most suitable model for predicting bank loan eligibility in the presence of balancing techniques. This involves a comparative analysis of the three models to identify which one demonstrates the highest efficacy in handling imbalanced data, post-application of SMOTE or ADASYN. The criteria for model effectiveness will encompass a range of performance metrics, emphasizing the importance of accuracy and fairness in predictions. This aspect of the research is expected to yield valuable recommendations for financial institutions on the optimal combination of predictive models and balancing techniques.

# References

[1] P. S. Saini, A. Bhatnagar, and L. Rani, “Loan Approval Prediction using Machine Learning: A Comparative Analysis of Classification Algorithms,” in *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, May 2023, pp. 1821–1826. doi: 10.1109/ICACITE57410.2023.10182799.

[2] “Machine Learning Models for Predicting Bank Loan Eligibility | IEEE Conference Publication | IEEE Xplore.” Accessed: Jan. 20, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/9803172

[3] 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

[4] R. Purswani, S. Verma, and Y. Jaiswal, “Loan Approval Prediction using Machine Learning: A Review,” vol. 08, no. 06, 2021.