*Yun Zhu*

*Boston University*

*data mining Project*

Data classification study

Table of Contents

[1. R language 3](#_Toc183113595)

[1.1. Introduction 3](#_Toc183113596)

[1.2. RStudio 3](#_Toc183113597)

[2. Models 4](#_Toc183113598)

[2.1. Balancing methods 4](#_Toc183113599)

[2.1.1. Undersampling 4](#_Toc183113600)

[2.1.2. Random Over-Sampling Examples 4](#_Toc183113601)

[2.2. Feature Selection Methods 5](#_Toc183113602)

[2.2.1. Variance Threshold 5](#_Toc183113603)

[2.2.2. Correlation Threshold 6](#_Toc183113604)

[2.2.3. Recursive Feature Elimination 6](#_Toc183113605)

[2.3. Classification Algorithms 7](#_Toc183113606)

[2.3.1. Logistic Regression 7](#_Toc183113607)

[2.3.2. Decision Tree 7](#_Toc183113608)

[2.3.3. Random Forest 8](#_Toc183113609)

[2.3.4. Support Vector Machine 8](#_Toc183113610)

[2.3.5. K-Nearest Neighbors 9](#_Toc183113611)

[2.3.6. Naive Bayes 9](#_Toc183113612)

[3. Data preprocessing 10](#_Toc183113613)

[4. Establishing methods 13](#_Toc183113614)

[4.1. Balancing methods 13](#_Toc183113615)

[4.1.1. apply\_balancing\_method\_b1 13](#_Toc183113616)

[4.1.2. apply\_balancing\_method\_b2 13](#_Toc183113617)

[4.2. Feature Selection Methods 13](#_Toc183113618)

[4.2.1. apply\_feature\_selection\_f1 13](#_Toc183113619)

[4.2.2. apply\_feature\_selection\_f2 13](#_Toc183113620)

[4.2.3. apply\_feature\_selection\_f3 13](#_Toc183113621)

[4.3. Classification Models 13](#_Toc183113622)

[4.3.1. train\_classification\_model\_c1 13](#_Toc183113623)

[4.3.2. train\_classification\_model\_c2 13](#_Toc183113624)

[4.3.3. train\_classification\_model\_c3 14](#_Toc183113625)

[4.3.4. train\_classification\_model\_c4 14](#_Toc183113626)

[4.3.5. train\_classification\_model\_c5 14](#_Toc183113627)

[4.3.6. train\_classification\_model\_c6 14](#_Toc183113628)

[5. Evaluation method establish 14](#_Toc183113629)

[5.1. Creation of the results Data Frame 14](#_Toc183113630)

[5.2. Helper Function ‘evaluate\_model’ 15](#_Toc183113631)

[5.2.1. Confusion Matrix and Metrics Calculation 15](#_Toc183113632)

[5.2.2. Calculating Performance Metrics 15](#_Toc183113633)

[5.2.3. Updating the results Data Frame 16](#_Toc183113634)

[6. Applying and evaluate methods 16](#_Toc183113635)

[6.1. Per combination analysis 16](#_Toc183113636)

[6.2. Conclusion 38](#_Toc183113637)

[6.2.1. Most Accurate Combination 39](#_Toc183113638)

[6.2.2. Least Accurate Combination 39](#_Toc183113639)

[7. Final best model 40](#_Toc183113640)

[7.1. Combination 34 40](#_Toc183113641)

[8. Conclusion 41](#_Toc183113642)

Abstract

This project explores data classification challenges and addresses them through the systematic application of data balancing methods, feature selection techniques, and machine learning algorithms. Using R programming and the RStudio environment, 36 combinations of methods were evaluated to identify the optimal approach for handling class imbalance and improving model accuracy. Data preprocessing steps, including outlier handling, normalization, and missing value treatment, ensured a clean and structured dataset. The study revealed that the combination of the ROSE balancing method, recursive feature elimination (RFE), and Support Vector Machines (SVM) yielded the best results, achieving a high F1-Score of 0.874. This combination demonstrated a robust balance between precision and recall, effectively managing class imbalance and optimizing feature relevance. Conversely, the least accurate combination highlighted the limitations of alternative approaches, reinforcing the importance of strategic method selection. The findings underscore the significance of preprocessing, model selection, and evaluation in developing effective classification systems, offering valuable insights for future research in data mining and machine learning applications.

# R language

## Introduction

R is a programming language and software environment commonly used for statistical computing, data analysis, and graphical representation. It was developed by statisticians Ross Ihaka and Robert Gentleman in the early 1990s.

R is a programming language and software environment widely recognized for statistical computing, data analysis, and data visualization. One of its primary strengths lies in statistical analysis. R comes with an extensive library of built-in functions, allowing users to perform various statistical tests, analyses, and modeling techniques efficiently. Additionally, R excels in data visualization, offering powerful packages like ggplot2 and lattice, which enable the creation of visually appealing and highly customizable plots.

## RStudio

RStudio is an integrated development environment (IDE) specifically designed for the R programming language. It offers a user-friendly interface that simplifies the process of writing and executing R code, making it a popular choice among data analysts, statisticians, and data scientists. The interface is organized into multiple panes, allowing users to efficiently write code, view plots, explore data, and manage files. This streamlined setup enhances productivity and ease of use, especially for those engaged in data analysis and research.

# Models

## Balancing methods

Because the dataset may have the problem of category imbalance (for example, the number of samples in some categories is much larger than in other categories), we need to balance the data before training the model to improve the performance of the model.

### Undersampling

Undersampling is a data balancing technique used in machine learning to address the problem of class imbalance in datasets. The core idea behind undersampling is to randomly or strategically select a subset of the majority class samples while retaining all of the minority class samples. This method can be particularly effective when the majority class significantly outnumbers the minority class, and reducing the majority class data does not lead to a substantial loss of information. However, one of the drawbacks of undersampling is the potential risk of discarding valuable data from the majority class, which may result in a loss of important information and, consequently, a less robust model. To counter this limitation, undersampling is often combined with other techniques or used in conjunction with ensemble learning methods to enhance model performance.

### Random Over-Sampling Examples

The ROSE (Random Over-Sampling Examples) model is a data balancing technique specifically designed to handle class imbalance in machine learning datasets. In many real-world datasets, certain classes are underrepresented, leading to biased models that may perform poorly on minority classes. ROSE mitigates this problem by generating synthetic samples for the minority class, enhancing the dataset’s balance and improving model accuracy across all classes. This synthetic data generation is accomplished by creating new examples based on the attributes of existing data, producing realistic variations that expand the representation of minority classes.

By using ROSE, models can better learn the features of underrepresented classes, which is especially useful in applications like fraud detection, medical diagnosis, and customer churn analysis, where imbalanced classes are common.

## Feature Selection Methods

Feature selection is used to reduce the number of features, and remove irrelevant or redundant features, to improve the generalization ability and efficiency of the model.

### Variance Threshold

Variance Threshold is a simple, yet effective feature selection method used in data preprocessing for machine learning. It works by removing features from a dataset that have low or zero variance, which means these features provide little to no information for distinguishing between classes or making accurate predictions. The underlying principle is that a feature with little variability across all data points is likely to be irrelevant, as it does not contribute significantly to the model’s ability to differentiate between different outcomes. By eliminating such features, Variance Threshold helps reduce the dimensionality of the dataset, making the training process faster and improving model performance.

One of the advantages of Variance Threshold is its simplicity and computational efficiency, making it suitable for initial data exploration and feature selection. However, it has limitations, as it does not consider the relationship between features and the target variable. This means that some features with low variance might still be valuable for prediction if they have a strong association with the target variable. Therefore, Variance Threshold is often used as a preliminary step, followed by more sophisticated feature selection techniques that consider the feature’s importance concerning the target outcome.

### Correlation Threshold

Correlation Threshold is a feature selection technique used to remove features that are highly correlated with each other in a dataset. In machine learning, having features that are highly correlated can lead to multicollinearity, which can cause problems for certain models, especially linear models. Multicollinearity can make it difficult to interpret the importance of individual features and may also result in unstable or unreliable model coefficients. By using the Correlation Threshold method, one can simplify the dataset, reduce redundancy, and enhance the model’s performance and interpretability.

The Correlation Threshold method involves calculating the correlation matrix of all the features and identifying pairs of features that have a correlation coefficient above a predefined threshold. Correlation coefficients range from -1 to 1, where values close to 1 or -1 indicate a strong positive or negative correlation, respectively. If two features have a correlation coefficient that exceeds the specified threshold, one of them is removed to eliminate redundancy. The choice of which feature to remove can depend on factors like domain knowledge, feature importance scores, or even random selection.

### Recursive Feature Elimination

Recursive Feature Elimination (RFE) is a powerful feature selection technique used in machine learning to identify the most important features in a dataset. The method works by recursively fitting a model and removing the least important features until a desired number of features is reached. Specifically, RFE begins by training the model on the entire set of features and then assigns importance scores to each feature based on the model’s coefficients or feature importances. The least important feature is then removed, and the model is refit on the remaining features. This process is repeated until the desired number of features, or a stopping criterion is met.

RFE is particularly useful because it not only selects a subset of relevant features but also improves model performance by reducing overfitting and decreasing computational complexity.

## Classification Algorithms

### Logistic Regression

Logistic Regression is a fundamental statistical method used for binary classification problems, where the goal is to predict one of two possible outcomes based on a set of input features. Unlike linear regression, which is used for continuous outcomes, logistic regression models the probability that an observation belongs to a particular class. It does this by applying the logistic (or sigmoid) function to a linear combination of the input features. The sigmoid function maps the output to a value between 0 and 1, which represents the probability of the observation belonging to the positive class. If the probability is greater than a specified threshold (commonly 0.5), the observation is classified into the positive class; otherwise, it is classified into the negative class.

### Decision Tree

A Decision Tree is a widely used machine learning algorithm for both classification and regression tasks. It is a tree-like model that makes decisions based on a series of hierarchical, if-then-else rules derived from the features of the dataset. The model starts at a root node and splits the data into subsets based on feature values, forming branches that lead to internal nodes or terminal nodes (also called leaves). Each internal node represents a decision point based on a feature and a splitting criterion, while each leaf node represents a final decision or prediction. This process continues recursively, partitioning the dataset until a stopping criterion is met, such as a maximum depth or a minimum number of samples per leaf.

One of the main advantages of Decision Trees is their interpretability. They provide a clear and intuitive representation of how decisions are made, making it easy to understand the logic behind the model’s predictions.

### Random Forest

Random Forest is a powerful and versatile ensemble learning method used for both classification and regression tasks. It builds upon the concept of Decision Trees by creating a collection (or “forest”) of multiple Decision Trees, each trained on different random subsets of the training data and features. The model makes predictions by aggregating the outputs of these individual trees: for classification, it takes a majority vote among the trees, while for regression, it averages the predictions. By combining the results of many Decision Trees, Random Forest reduces the risk of overfitting and improves the model’s generalization performance.

One of the key strengths of Random Forest is its ability to handle large datasets with a high number of features and maintain high accuracy. Another advantage of Random Forest is its ability to provide feature importance scores, which indicate the relative importance of each feature in making predictions. This is particularly useful for understanding the influence of different variables in the model and can guide feature selection in the data preprocessing stage.

### Support Vector Machine

Support Vector Machine (SVM) is a powerful and flexible supervised learning algorithm used for both classification and regression tasks, though it is more commonly used for classification. The main goal of SVM is to find the optimal hyperplane that best separates data points belonging to different classes. In a simple two-dimensional space, this hyperplane is a line that divides the feature space into two regions, each corresponding to one class. In higher-dimensional spaces, the hyperplane becomes a flat surface that maximizes the margin between the nearest data points of each class, known as support vectors. These support vectors are the critical elements of the dataset that define the position and orientation of the hyperplane.

One of the key strengths of SVM is its effectiveness in handling high-dimensional data and its ability to find complex decision boundaries using a technique called the kernel trick. SVM also has the advantage of being robust to overfitting, particularly when the data has a clear margin of separation. It works well in situations where the number of features is greater than the number of samples and can be effective even with noisy or overlapping data, depending on the choice of the regularization parameter.

### K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a simple, yet effective machine learning algorithm used for both classification and regression tasks. It operates on the principle of similarity, making predictions based on the “k” closest data points in the feature space. In a classification setting, KNN assigns a class to a data point based on a majority vote from its k-nearest neighbors. For regression tasks, KNN predicts the value of a data point as the average (or weighted average) of the values of its k-nearest neighbors. The value of k is a critical hyperparameter that determines the performance of the model: a smaller k value makes the model sensitive to noise, while a larger k value makes the decision boundary smoother but may overlook local patterns.

KNN is an intuitive algorithm that does not make strong assumptions about the underlying data distribution, making it useful for a variety of problems. It is easy to implement and understand, as the algorithm simply calculates the distance between data points using metrics like Euclidean distance, Manhattan distance, or Minkowski distance. One of the main advantages of KNN is its ability to adapt to the data’s shape, making it suitable for non-linear decision boundaries. KNN can also handle multi-class classification problems and works well for datasets with a relatively small number of features.

### Naive Bayes

Naive Bayes is a family of simple, yet powerful probabilistic algorithms based on Bayes’ Theorem, widely used for classification tasks. Bayes’ Theorem provides a way to calculate the posterior probability of a class given a set of features, using the prior probability of the class, the likelihood of the features given the class, and the overall probability of the features. Naive Bayes classifiers come in several variants, tailored to different types of data. The most common ones are Gaussian Naive Bayes, which is used for continuous data assuming a normal distribution; Multinomial Naive Bayes, used for discrete data such as word counts in text classification problems; and Bernoulli Naive Bayes, used for binary or boolean features. These models are widely applied in tasks such as spam detection, sentiment analysis, document classification, and medical diagnosis.

One of the main advantages of Naive Bayes classifiers is their efficiency and simplicity. They are easy to implement and can handle large datasets with ease, making them suitable for real-time applications. The algorithm requires relatively little training data and can make quick predictions, as it only needs to calculate probabilities using simple arithmetic operations. Additionally, Naive Bayes is less affected by the curse of dimensionality and works well even when the number of features is large.

# Data preprocessing

#### Loading Necessary Packages

First, necessary R packages were loaded to perform various data preprocessing tasks. If a package was not installed, it was automatically installed and loaded into the session. Several packages are loaded:

- `ggplot2`: For data visualization.

- `dplyr`: For data manipulation.

- `rpart.plot`: For plotting decision trees.

- `MASS`: For statistical functions.

#### Loading the Dataset

The dataset was loaded using the `read.csv()` function from a specified path. This step reads the CSV file into an R data frame for subsequent analysis.

#### Data Exploration

Initial exploration was done to understand the basic structure and contents of the dataset, and to check for missing values.

Firstly, we do summary statistics. It provides an overview of the data, including count, mean, min, and max for numerical variables, as well as counts for categorical variables. As we can see in the results, we get only one character column `Class`, which serves as the category. The other columns are all numeric ones.

Then, we do a missing value check. `colSums(is.na(data))` provides the count of missing values for each column. As we can see from the result, from column ` SAFETIME` to ` HHADULT`, we get 2102 missing values for each column. We need to process these in future steps. However, there’s no missing value in the `Class` column, so we don’t have to deal with it.

#### Handling Missing Values

To manage missing data, columns with more than 50% missing values were removed. This is mainly because if more than half the data in a column is missing, the remaining values may not offer enough information to capture meaningful trends or patterns, reducing the column’s contribution to model performance. Also, filling in many missing values can introduce bias or noise.

For the remaining columns, missing values in numeric columns were filled with the median of the respective column. Median is less sensitive to outliers compared to the mean, so it is more accurate.

#### Outlier Detection and Handling

Outliers in numeric columns were handled using the interquartile range (IQR) method. This calculates the first (Q1) and third quartile (Q3) of the column x, which correspond to the 25th and 75th percentiles. These two quartiles help define the IQR, which is the range between Q1 and Q3. The na.rm = T argument ensures that any NA values are ignored when calculating the quantiles.

This approach preserves the overall distribution while mitigating the impact of extreme values.

#### Feature Engineering

Categorical variables were transformed into factors to prepare them for modeling.

Columns with near-zero variance were removed to reduce redundancy. A custom function is defined with one argument df, which represents the dataset. The purpose of this function is to identify and remove columns where the variance is lower than 0.01. A variance below 0.01 indicates that the values in the column are almost constant (low variance), meaning the column doesn’t provide much useful information for modeling.

Features with very little variance provide little to no predictive power and can increase the complexity of the model unnecessarily.

#### Data Normalization

Numeric columns were standardized to ensure all features are on the same scale.

Standardization is crucial when using machine learning models that are sensitive to feature scaling, such as SVMs, KNN, and neural networks.

By transforming each numeric feature to have a mean of 0 and a standard deviation of 1, the model will not disproportionately favor features with larger numeric ranges.

#### Final Data Structure

The preprocessed dataset was summarized using `str()` to check the final structure. The dataset is now clean, well-structured, and ready for building robust machine learning models.

# Establishing methods

Once we get the preprocessed data, we can now establish 2 balancing methods, 3 feature selection methods, and 6 classification algorithms.

## Balancing methods

### apply\_balancing\_method\_b1

Uses the ROSE library to undersample the majority class in the data, aiming to balance the class distribution.

### apply\_balancing\_method\_b2

Uses the ROSE method from the ROSE package to o address class imbalance by generating synthetic examples to create a more balanced dataset, improving model performance on underrepresented classes.

## Feature Selection Methods

### apply\_feature\_selection\_f1

Removes features with near-zero variance using nearZeroVar from caret.

### apply\_feature\_selection\_f2

Identifies and removes features with high correlation using findCorrelation.

### apply\_feature\_selection\_f3

Uses recursive feature elimination (RFE) with random forest as the underlying algorithm to select important features.

## Classification Models

### train\_classification\_model\_c1

Logistic regression using glm.

### train\_classification\_model\_c2

Decision tree model using rpart.

### train\_classification\_model\_c3

Random forest model using randomForest.

### train\_classification\_model\_c4

Support vector machine (SVM) using svm from e1071.

### train\_classification\_model\_c5

k-Nearest Neighbors (k-NN) using class::knn.

### train\_classification\_model\_c6

Naive Bayes classifier using e1071::naiveBayes.

# Evaluation method establish

## Creation of the results Data Frame

We initialize an empty data frame named ‘results’. It has the following columns to store the evaluation metrics for each model combination:

Combination: A character column indicating the specific combination of balancing, feature selection, and classification method used.

Class: A character column to differentiate between the two classes (e.g., “N” and “Y”).

TPR (True Positive Rate or Sensitivity): A numeric column to store the rate of correctly identified positive instances.

FPR (False Positive Rate): A numeric column to store the rate of incorrectly identified negative instances.

Precision: A numeric column for the ratio of true positives to the total predicted positives.

Recall: A numeric column, which is another name for Sensitivity or TPR.

F\_measure: A numeric column to store the harmonic mean of Precision and Recall.

ROC\_AUC: A numeric column that would typically store the Area Under the Receiver Operating Characteristic Curve; however, it is currently set to NA because probabilities are needed for this calculation.

Kappa: A numeric column to store Cohen’s Kappa, a measure of agreement between predicted and actual classifications beyond chance.

## Helper Function ‘evaluate\_model’

This function, ‘evaluate\_model’, is designed to evaluate the performance of a model’s predictions.

This function has these inputs:

predicted: The predicted labels from the model.

actual: The actual labels from the dataset.

model\_name: The name of the classification model used (not directly used in this code but can be useful for debugging or further enhancements).

combination: A string describing the combination of balancing, feature selection, and model used.

In this function, we created several steps to output the model’s evaluation.

### Confusion Matrix and Metrics Calculation

We create a confusion matrix using the caret package’s confusionMatrix function, comparing the predicted labels to the actual ones.

### Calculating Performance Metrics

tpr: Extracts the True Positive Rate (TPR) or Sensitivity from the confusion matrix.

fpr: Calculates the False Positive Rate (FPR) as 1 - Specificity.

precision: Retrieves Precision, the ratio of true positives to the total predicted positives.

recall: Recall is the same as Sensitivity or TPR.

f\_measure: Calculates the F-measure, which is the harmonic mean of Precision and Recall.

roc\_auc: The ROC AUC is set to NA since the code does not currently calculate probabilities needed for this metric.

kappa: Retrieves Cohen’s Kappa from the confusion matrix.

### Updating the results Data Frame

This section appends a new row to the results data frame for Class “N” and “Y”. The <<- operator updates the global results data frame within the function.

Finally, the current state of the results data frame is printed to the console, showing the accumulated evaluation metrics for each model combination tested so far.

# Applying and evaluate methods

As we have 2 balancing methods, 3 feature selection methods, and 6 classification algorithms, we have all together 36 combinations. Now we apply the methods one by one.

## Per combination analysis

#### Combination 1: b1 + f1 + c1

After applying the balance method b1, feature selection method f1, and classification algorithm c1, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 969 | 37 |
| Y | 347 | 141 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.73632219 | 0.26367781 | 0.96322068 | 0.73632219 | 0.83462532 |  | 0.36503854 | 0.30145773 |
| Class Y | 0.79213483 | 0.20786517 | 0.28893443 | 0.79213483 | 0.42342342 |  | 0.36503854 | 0.30145773 |
| Wt.  Average | 0.74297189 | 0.25702811 | 0.88288403 | 0.74297189 | 0.7856334 | 0.84139886 | 0.36503854 | 0.30145773 |

#### Combination 2: b1 + f1 + c2

After applying the balance method b1, feature selection method f1, and classification algorithm c2, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1004 | 53 |
| Y | 312 | 125 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.76291793 | 0.23708207 | 0.94985809 | 0.76291793 | 0.84618626 |  | 0.33125838 | 0.28553086 |
| Class Y | 0.70224719 | 0.29775281 | 0.28604119 | 0.70224719 | 0.40650407 |  | 0.33125838 | 0.28553086 |
| Wt.  Average | 0.75568942 | 0.24431058 | 0.87076879 | 0.75568942 | 0.7938011 |  | 0.33125838 | 0.28553086 |

#### Combination 3: b1 + f1 + c3

After applying the balance method b1, feature selection method f1, and classification algorithm c3, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 949 | 37 |
| Y | 367 | 141 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.72112462 | 0.27887538 | 0.96247465 | 0.72112462 | 0.82450043 |  | 0.35099796 | 0.28489306 |
| Class Y | 0.79213483 | 0.20786517 | 0.27755906 | 0.79213483 | 0.41107872 |  | 0.35099796 | 0.28489306 |
| Wt.  Average | 0.72958501 | 0.27041499 | 0.88087158 | 0.72958501 | 0.77524403 |  | 0.35099796 | 0.28489306 |

#### Combination 4: b1 + f1 + c4

After applying the balance method b1, feature selection method f1, and classification algorithm c4, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 968 | 36 |
| Y | 348 | 142 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.73556231 | 0.26443769 | 0.96414343 | 0.73556231 | 0.83448276 |  | 0.36800773 | 0.30338825 |
| Class Y | 0.79775281 | 0.20224719 | 0.28979592 | 0.79775281 | 0.4251497 |  | 0.36800773 | 0.30338825 |
| Wt.  Average | 0.74297189 | 0.25702811 | 0.88379948 | 0.74297189 | 0.78571349 |  | 0.36800773 | 0.30338825 |

#### Combination 5: b1 + f1 + c5

After applying the balance method b1, feature selection method f1, and classification algorithm c5, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1054 | 76 |
| Y | 262 | 102 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.80091185 | 0.19908815 | 0.93274336 | 0.80091185 | 0.86181521 |  | 0.28219961 | 0.25757317 |
| Class Y | 0.57303371 | 0.42696629 | 0.28021978 | 0.57303371 | 0.37638376 |  | 0.28219961 | 0.25757317 |
| Wt.  Average | 0.77376171 | 0.22623829 | 0.85499959 | 0.77376171 | 0.80397933 |  | 0.28219961 | 0.25757317 |

#### Combination 6: b1 + f1 + c6

After applying the balance method b1, feature selection method f1, and classification algorithm c6, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 934 | 45 |
| Y | 382 | 133 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.70972644 | 0.29027356 | 0.95403473 | 0.70972644 | 0.81394336 |  | 0.31144455 | 0.25124824 |
| Class Y | 0.74719101 | 0.25280899 | 0.25825243 | 0.74719101 | 0.38383838 |  | 0.31144455 | 0.25124824 |
| Wt.  Average | 0.71419009 | 0.28580991 | 0.87113697 | 0.71419009 | 0.76269926 | 0.79045285 | 0.31144455 | 0.25124824 |

#### Combination 7: b1 + f2 + c1

After applying the balance method b1, feature selection method f2, and classification algorithm c1, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 953 | 30 |
| Y | 363 | 148 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.72416413 | 0.27583587 | 0.96948118 | 0.72416413 | 0.82905611 |  | 0.37943061 | 0.3071662 |
| Class Y | 0.83146067 | 0.16853933 | 0.28962818 | 0.83146067 | 0.42960813 |  | 0.37943061 | 0.3071662 |
| Wt.  Average | 0.73694779 | 0.26305221 | 0.88848129 | 0.73694779 | 0.78146458 | 0.85307879 | 0.37943061 | 0.3071662 |

#### Combination 8: b1 + f2 + c2

After applying the balance method b1, feature selection method f2, and classification algorithm c2, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1004 | 53 |
| Y | 312 | 125 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.76291793 | 0.23708207 | 0.94985809 | 0.76291793 | 0.84618626 |  | 0.33125838 | 0.28553086 |
| Class Y | 0.70224719 | 0.29775281 | 0.28604119 | 0.70224719 | 0.40650407 |  | 0.33125838 | 0.28553086 |
| Wt.  Average | 0.75568942 | 0.24431058 | 0.87076879 | 0.75568942 | 0.7938011 |  | 0.33125838 | 0.28553086 |

#### Combination 9: b1 + f2 + c3

After applying the balance method b1, feature selection method f2, and classification algorithm c3, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 909 | 32 |
| Y | 407 | 146 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.69072948 | 0.30927052 | 0.96599362 | 0.69072948 | 0.80549402 |  | 0.34281715 | 0.26739019 |
| Class Y | 0.82022472 | 0.17977528 | 0.26401447 | 0.82022472 | 0.3994528 |  | 0.34281715 | 0.26739019 |
| Wt.  Average | 0.70615797 | 0.29384203 | 0.88235755 | 0.70615797 | 0.75711695 |  | 0.34281715 | 0.26739019 |

#### Combination 10: b1 + f2 + c4

After applying the balance method b1, feature selection method f2, and classification algorithm c4, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 936 | 29 |
| Y | 380 | 149 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.7112462 | 0.2887538 | 0.96994819 | 0.7112462 | 0.82069268 |  | 0.37143633 | 0.29597642 |
| Class Y | 0.83707865 | 0.16292135 | 0.28166352 | 0.83707865 | 0.42149929 |  | 0.37143633 | 0.29597642 |
| Wt.  Average | 0.72623829 | 0.27376171 | 0.88794372 | 0.72623829 | 0.77313149 |  | 0.37143633 | 0.29597642 |

#### Combination 11: b1 + f2 + c5

After applying the balance method b1, feature selection method f2, and classification algorithm c5, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1056 | 85 |
| Y | 260 | 93 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.80243161 | 0.19756839 | 0.92550394 | 0.80243161 | 0.85958486 |  | 0.24777768 | 0.22798908 |
| Class Y | 0.52247191 | 0.47752809 | 0.26345609 | 0.52247191 | 0.35028249 |  | 0.24777768 | 0.22798908 |
| Wt.  Average | 0.76907631 | 0.23092369 | 0.84662542 | 0.76907631 | 0.79890492 |  | 0.24777768 | 0.22798908 |

#### Combination 12: b1 + f2 + c6

After applying the balance method b1, feature selection method f2, and classification algorithm c6, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 925 | 46 |
| Y | 391 | 132 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.70288754 | 0.29711246 | 0.95262616 | 0.70288754 | 0.80891998 |  | 0.30186359 | 0.2418146 |
| Class Y | 0.74157303 | 0.25842697 | 0.25239006 | 0.74157303 | 0.37660485 |  | 0.30186359 | 0.2418146 |
| Wt.  Average | 0.70749665 | 0.29250335 | 0.86919776 | 0.70749665 | 0.75741256 | 0.78706755 | 0.30186359 | 0.2418146 |

#### Combination 13: b1 + f3 + c1

After applying the balance method b1, feature selection method f3, and classification algorithm c1, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 980 | 47 |
| Y | 336 | 131 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.74468085 | 0.25531915 | 0.95423564 | 0.74468085 | 0.83653436 |  | 0.33590038 | 0.28239555 |
| Class Y | 0.73595506 | 0.26404494 | 0.28051392 | 0.73595506 | 0.40620155 |  | 0.33590038 | 0.28239555 |
| Wt.  Average | 0.74364123 | 0.25635877 | 0.87396625 | 0.74364123 | 0.78526311 | 0.83400072 | 0.33590038 | 0.28239555 |

#### Combination 14: b1 + f3 + c2

After applying the balance method b1, feature selection method f3, and classification algorithm c2, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1004 | 53 |
| Y | 312 | 125 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.76291793 | 0.23708207 | 0.94985809 | 0.76291793 | 0.84618626 |  | 0.33125838 | 0.28553086 |
| Class Y | 0.70224719 | 0.29775281 | 0.28604119 | 0.70224719 | 0.40650407 |  | 0.33125838 | 0.28553086 |
| Wt.  Average | 0.75568942 | 0.24431058 | 0.87076879 | 0.75568942 | 0.7938011 |  | 0.33125838 | 0.28553086 |

#### Combination 15: b1 + f3 + c3

After applying the balance method b1, feature selection method f3, and classification algorithm c3, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 931 | 33 |
| Y | 385 | 145 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.70744681 | 0.29255319 | 0.96576763 | 0.70744681 | 0.81666667 |  | 0.35348952 | 0.28142663 |
| Class Y | 0.81460674 | 0.18539326 | 0.27358491 | 0.81460674 | 0.40960452 |  | 0.35348952 | 0.28142663 |
| Wt.  Average | 0.72021419 | 0.27978581 | 0.88329874 | 0.72021419 | 0.76816796 |  | 0.35348952 | 0.28142663 |

#### Combination 16: b1 + f3 + c4

After applying the balance method b1, feature selection method f3, and classification algorithm c4, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 919 | 29 |
| Y | 397 | 149 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.69832827 | 0.30167173 | 0.96940928 | 0.69832827 | 0.81183746 |  | 0.36018153 | 0.28270219 |
| Class Y | 0.83707865 | 0.16292135 | 0.27289377 | 0.83707865 | 0.41160221 |  | 0.36018153 | 0.28270219 |
| Wt.  Average | 0.71485944 | 0.28514056 | 0.88642417 | 0.71485944 | 0.76415213 |  | 0.36018153 | 0.28270219 |

#### Combination 17: b1 + f3 + c5

After applying the balance method b1, feature selection method f3, and classification algorithm c5, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1077 | 81 |
| Y | 239 | 97 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.81838906 | 0.18161094 | 0.93005181 | 0.81838906 | 0.87065481 |  | 0.28191535 | 0.26256363 |
| Class Y | 0.54494382 | 0.45505618 | 0.28869048 | 0.54494382 | 0.37743191 |  | 0.28191535 | 0.26256363 |
| Wt.  Average | 0.78580991 | 0.21419009 | 0.85363795 | 0.78580991 | 0.81189064 |  | 0.28191535 | 0.26256363 |

#### Combination 18: b1 + f3 + c6

After applying the balance method b1, feature selection method f3, and classification algorithm c6, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 926 | 44 |
| Y | 390 | 134 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.70364742 | 0.29635258 | 0.95463918 | 0.70364742 | 0.81014873 |  | 0.30987443 | 0.24801332 |
| Class Y | 0.75280899 | 0.24719101 | 0.25572519 | 0.75280899 | 0.38176638 |  | 0.30987443 | 0.24801332 |
| Wt.  Average | 0.70950469 | 0.29049531 | 0.8713683 | 0.70950469 | 0.75910987 | 0.79210922 | 0.30987443 | 0.24801332 |

#### Combination 19: b2 + f1 + c1

After applying the balance method b2, feature selection method f1, and classification algorithm c1, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1005 | 41 |
| Y | 311 | 137 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.76367781 | 0.23632219 | 0.96080306 | 0.76367781 | 0.85097375 |  | 0.37708378 | 0.32209612 |
| Class Y | 0.76966292 | 0.23033708 | 0.30580357 | 0.76966292 | 0.43769968 |  | 0.37708378 | 0.32209612 |
| Wt.  Average | 0.7643909 | 0.2356091 | 0.8827643 | 0.7643909 | 0.80173494 | 0.84888238 | 0.37708378 | 0.32209612 |

#### Combination 20: b2 + f1 + c2

After applying the balance method b2, feature selection method f1, and classification algorithm c2, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 960 | 47 |
| Y | 356 | 131 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.72948328 | 0.27051672 | 0.95332671 | 0.72948328 | 0.82651743 |  | 0.32167765 | 0.26587721 |
| Class Y | 0.73595506 | 0.26404494 | 0.26899384 | 0.73595506 | 0.39398496 |  | 0.32167765 | 0.26587721 |
| Wt.  Average | 0.73025435 | 0.26974565 | 0.87179308 | 0.73025435 | 0.77498411 |  | 0.32167765 | 0.26587721 |

#### Combination 21: b2 + f1 + c3

After applying the balance method b2, feature selection method f1, and classification algorithm c3, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1062 | 55 |
| Y | 254 | 123 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.80699088 | 0.19300912 | 0.95076097 | 0.80699088 | 0.8729963 |  | 0.37142563 | 0.33572101 |
| Class Y | 0.69101124 | 0.30898876 | 0.32625995 | 0.69101124 | 0.44324324 |  | 0.37142563 | 0.33572101 |
| Wt.  Average | 0.79317269 | 0.20682731 | 0.87635589 | 0.79317269 | 0.82179413 |  | 0.37142563 | 0.33572101 |

#### Combination 22: b2 + f1 + c4

After applying the balance method b2, feature selection method f1, and classification algorithm c4, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1240 | 105 |
| Y | 76 | 73 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.94224924 | 0.05775076 | 0.92193309 | 0.94224924 | 0.93198046 |  | 0.38095374 | 0.37906378 |
| Class Y | 0.41011236 | 0.58988764 | 0.48993289 | 0.41011236 | 0.44648318 |  | 0.38095374 | 0.37906378 |
| Wt.  Average | 0.87884873 | 0.12115127 | 0.87046318 | 0.87884873 | 0.87413674 |  | 0.38095374 | 0.37906378 |

#### Combination 23: b2 + f1 + c5

After applying the balance method b2, feature selection method f1, and classification algorithm c5, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1204 | 126 |
| Y | 112 | 52 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.91489362 | 0.08510638 | 0.90526316 | 0.91489362 | 0.91005291 |  | 0.2145459 | 0.21431665 |
| Class Y | 0.29213483 | 0.70786517 | 0.31707317 | 0.29213483 | 0.30409357 |  | 0.2145459 | 0.21431665 |
| Wt.  Average | 0.84069612 | 0.15930388 | 0.8351843 | 0.84069612 | 0.83785695 |  | 0.2145459 | 0.21431665 |

#### Combination 24: b2 + f1 + c6

After applying the balance method b2, feature selection method f1, and classification algorithm c6, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1074 | 72 |
| Y | 242 | 106 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.81610942 | 0.18389058 | 0.93717277 | 0.81610942 | 0.87246141 |  | 0.31546214 | 0.29131846 |
| Class Y | 0.59550562 | 0.40449438 | 0.3045977 | 0.59550562 | 0.40304183 |  | 0.31546214 | 0.29131846 |
| Wt.  Average | 0.78982597 | 0.21017403 | 0.86180573 | 0.78982597 | 0.81653324 | 0.79557136 | 0.31546214 | 0.29131846 |

#### Combination 25: b2 + f2 + c1

After applying the balance method b2, feature selection method f2, and classification algorithm c1, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1005 | 41 |
| Y | 311 | 137 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.76367781 | 0.23632219 | 0.96080306 | 0.76367781 | 0.85097375 |  | 0.37708378 | 0.32209612 |
| Class Y | 0.76966292 | 0.23033708 | 0.30580357 | 0.76966292 | 0.43769968 |  | 0.37708378 | 0.32209612 |
| Wt.  Average | 0.7643909 | 0.2356091 | 0.8827643 | 0.7643909 | 0.80173494 | 0.84888238 | 0.37708378 | 0.32209612 |

#### Combination 26: b2 + f2 + c2

After applying the balance method b2, feature selection method f2, and classification algorithm c2, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 960 | 47 |
| Y | 356 | 131 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.72948328 | 0.27051672 | 0.95332671 | 0.72948328 | 0.82651743 |  | 0.32167765 | 0.26587721 |
| Class Y | 0.73595506 | 0.26404494 | 0.26899384 | 0.73595506 | 0.39398496 |  | 0.32167765 | 0.26587721 |
| Wt.  Average | 0.73025435 | 0.26974565 | 0.87179308 | 0.73025435 | 0.77498411 |  | 0.32167765 | 0.26587721 |

#### Combination 27: b2 + f2 + c3

After applying the balance method b2, feature selection method f2, and classification algorithm c3, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1062 | 55 |
| Y | 254 | 123 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.80699088 | 0.19300912 | 0.95076097 | 0.80699088 | 0.8729963 |  | 0.37142563 | 0.33572101 |
| Class Y | 0.69101124 | 0.30898876 | 0.32625995 | 0.69101124 | 0.44324324 |  | 0.37142563 | 0.33572101 |
| Wt.  Average | 0.79317269 | 0.20682731 | 0.87635589 | 0.79317269 | 0.82179413 |  | 0.37142563 | 0.33572101 |

#### Combination 28: b2 + f2 + c4

After applying the balance method b2, feature selection method f2, and classification algorithm c4, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1240 | 105 |
| Y | 76 | 73 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.94224924 | 0.05775076 | 0.92193309 | 0.94224924 | 0.93198046 |  | 0.38095374 | 0.37906378 |
| Class Y | 0.41011236 | 0.58988764 | 0.48993289 | 0.41011236 | 0.44648318 |  | 0.38095374 | 0.37906378 |
| Wt.  Average | 0.87884873 | 0.12115127 | 0.87046318 | 0.87884873 | 0.87413674 |  | 0.38095374 | 0.37906378 |

#### Combination 29: b2 + f2 + c5

After applying the balance method b2, feature selection method f2, and classification algorithm c5, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1204 | 126 |
| Y | 112 | 52 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.91489362 | 0.08510638 | 0.90526316 | 0.91489362 | 0.91005291 |  | 0.2145459 | 0.21431665 |
| Class Y | 0.29213483 | 0.70786517 | 0.31707317 | 0.29213483 | 0.30409357 |  | 0.2145459 | 0.21431665 |
| Wt.  Average | 0.84069612 | 0.15930388 | 0.8351843 | 0.84069612 | 0.83785695 |  | 0.2145459 | 0.21431665 |

#### Combination 30: b2 + f2 + c6

After applying the balance method b2, feature selection method f2, and classification algorithm c6, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1074 | 72 |
| Y | 242 | 106 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.81610942 | 0.18389058 | 0.93717277 | 0.81610942 | 0.87246141 |  | 0.31546214 | 0.29131846 |
| Class Y | 0.59550562 | 0.40449438 | 0.3045977 | 0.59550562 | 0.40304183 |  | 0.31546214 | 0.29131846 |
| Wt.  Average | 0.78982597 | 0.21017403 | 0.86180573 | 0.78982597 | 0.81653324 | 0.79557136 | 0.31546214 | 0.29131846 |

#### Combination 31: b2 + f3 + c1

After applying the balance method b2, feature selection method f3, and classification algorithm c1, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1005 | 41 |
| Y | 311 | 137 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.76367781 | 0.23632219 | 0.96080306 | 0.76367781 | 0.85097375 |  | 0.37708378 | 0.32209612 |
| Class Y | 0.76966292 | 0.23033708 | 0.30580357 | 0.76966292 | 0.43769968 |  | 0.37708378 | 0.32209612 |
| Wt.  Average | 0.7643909 | 0.2356091 | 0.8827643 | 0.7643909 | 0.80173494 | 0.84888238 | 0.37708378 | 0.32209612 |

#### Combination 32: b2 + f3 + c2

After applying the balance method b2, feature selection method f3, and classification algorithm c2, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 960 | 47 |
| Y | 356 | 131 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.72948328 | 0.27051672 | 0.95332671 | 0.72948328 | 0.82651743 |  | 0.32167765 | 0.26587721 |
| Class Y | 0.73595506 | 0.26404494 | 0.26899384 | 0.73595506 | 0.39398496 |  | 0.32167765 | 0.26587721 |
| Wt.  Average | 0.73025435 | 0.26974565 | 0.87179308 | 0.73025435 | 0.77498411 |  | 0.32167765 | 0.26587721 |

#### Combination 33: b2 + f3 + c3

After applying the balance method b2, feature selection method f3, and classification algorithm c3, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1053 | 54 |
| Y | 263 | 124 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.80015198 | 0.19984802 | 0.95121951 | 0.80015198 | 0.86917045 |  | 0.36734471 | 0.32950231 |
| Class Y | 0.69662921 | 0.30337079 | 0.32041344 | 0.69662921 | 0.43893805 |  | 0.36734471 | 0.32950231 |
| Wt.  Average | 0.78781794 | 0.21218206 | 0.87606323 | 0.78781794 | 0.81791117 |  | 0.36734471 | 0.32950231 |

#### Combination 34: b2 + f3 + c4

After applying the balance method b2, feature selection method f3, and classification algorithm c4, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1240 | 105 |
| Y | 76 | 73 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.94224924 | 0.05775076 | 0.92193309 | 0.94224924 | 0.93198046 |  | 0.38095374 | 0.37906378 |
| Class Y | 0.41011236 | 0.58988764 | 0.48993289 | 0.41011236 | 0.44648318 |  | 0.38095374 | 0.37906378 |
| Wt.  Average | 0.87884873 | 0.12115127 | 0.87046318 | 0.87884873 | 0.87413674 |  | 0.38095374 | 0.37906378 |

#### Combination 35: b2 + f3 + c5

After applying the balance method b2, feature selection method f3, and classification algorithm c5, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1204 | 126 |
| Y | 112 | 52 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.91489362 | 0.08510638 | 0.90526316 | 0.91489362 | 0.91005291 |  | 0.2145459 | 0.21431665 |
| Class Y | 0.29213483 | 0.70786517 | 0.31707317 | 0.29213483 | 0.30409357 |  | 0.2145459 | 0.21431665 |
| Wt.  Average | 0.84069612 | 0.15930388 | 0.8351843 | 0.84069612 | 0.83785695 |  | 0.2145459 | 0.21431665 |

#### Combination 36: b2 + f3 + c6

After applying the balance method b2, feature selection method f3, and classification algorithm c6, we get the confusion matrix and evaluation of the model.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Actual | |
| Predicted |  | N | Y |
| N | 1074 | 72 |
| Y | 242 | 106 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TPR | FPR | Precision | Recall | F-measure | ROC | MCC | Kappa |
| Class N | 0.81610942 | 0.18389058 | 0.93717277 | 0.81610942 | 0.87246141 |  | 0.31546214 | 0.29131846 |
| Class Y | 0.59550562 | 0.40449438 | 0.3045977 | 0.59550562 | 0.40304183 |  | 0.31546214 | 0.29131846 |
| Wt.  Average | 0.78982597 | 0.21017403 | 0.86180573 | 0.78982597 | 0.81653324 | 0.79557136 | 0.31546214 | 0.29131846 |

## Conclusion

A graph of a number of blue bars

Description automatically generated with medium confidence

A graph of a graph with numbers and a number

Description automatically generated with medium confidence

### Most Accurate Combination

Combination 34 achieves the highest F-measure of approximately 0.874, indicating a balanced performance between precision and recall. The precision and recall are relatively high, both above 0.87, reflecting a reliable prediction capacity across classes.

While this combination lacks a specific ROC AUC score in the data, the high F-measure suggests strong overall accuracy.

### Least Accurate Combination

Combination 9 has the lowest F-measure of approximately 0.757, indicating it struggles to balance precision and recall compared to other combinations. It has a lower TPR (sensitivity) of about 0.706, showing room for improvement in correctly identifying positive cases.

This combination is therefore less effective for applications where high accuracy is critical.

# Final best model

## Combination 34

#### Balancing the Data with ROSE (b2)

The ROSE function (ROSE(Class ~ ., data = trainData, seed = 123)) is applied to the dataset.

ROSE generates synthetic samples for the minority class by sampling from a smoothed distribution of the existing data. This ensures both classes are equally represented, eliminating bias toward the majority class.

Outcome: A balanced training dataset is created, ensuring fairness and improved generalization for models trained in the next steps.

Feature Selection with Recursive Feature Elimination (f3)

apply\_rfe applies Recursive Feature Elimination (RFE) on the balanced dataset.

RFE ranks features based on their importance to the classification task and iteratively removes the least important ones. This process continues until the optimal subset of features is identified.

Outcome: A reduced dataset with only the most relevant features is created, reducing overfitting, computational cost, and noise.

#### Building the Classifier with SVM (c4)

The Support Vector Machine (SVM) model is trained using the reduced dataset (svm(Class ~ ., data = selected\_data)).

SVM works by identifying the optimal hyperplane that separates classes in the feature space.

Outcome: A trained SVM model that leverages the benefits of balanced and optimized data, enabling it to classify the test data with high precision and recall.

#### Result

The model was evaluated on the test dataset using metrics such as Precision, Recall, F1-Score, MCC, and Kappa. The confusion matrix revealed excellent performance for both classes, and the weighted metrics (TPR, Precision, F1) were among the best across all combinations:

Weighted F1-Score: 0.874137

Weighted Recall (TPR): 0.878849

MCC: 0.380954

# Conclusion

In this project, a comprehensive approach was taken to address classification tasks using various balancing methods, feature selection techniques, and machine learning algorithms. A total of 36 combinations were evaluated to determine the most effective strategies for maximizing model performance.

The results demonstrated that Combination 34, which used the ROSE balancing method, recursive feature elimination (RFE) for feature selection, and Support Vector Machines (SVM) as the classification model, outperformed all other combinations. This combination achieved a high F1-Score of 0.874, indicating a strong balance between precision and recall. The ROSE method effectively handled data imbalance, while RFE optimized feature selection, reducing noise and improving the model’s generalization capabilities.

Conversely, Combination 9, which yielded the lowest F1-Score of 0.757, highlighted areas where the selected methods struggled to balance sensitivity and precision. This underperformance underscored the importance of carefully choosing the right combination of balancing, feature selection, and classification techniques.

In conclusion, the findings emphasize the critical role of data preprocessing, method selection, and hyperparameter tuning in building robust classification models. The project not only identified an optimal combination but also provided valuable insights into the impact of different approaches on classification accuracy, making it a useful reference for similar tasks in the future.