INTERNATIONAL JOURNAL OF SOFTWARE ENGINEERING & COMPUTER SYSTEMS (IJSECS)
ISSN: 2289-8522 e-ISSN: 2180-0650
VOL. 11, ISSUE 1, pp. 16 – 31
DOI: https://doi.org/10.15282/ijsecs.1.1.2025.2.0134

ORIGINAL ARTICLE

Optimizing Support Vector Machine For Imbalanced Datasets By Combining Posterior Probability And Correlation Methods

Canggih Ajika Pamungkas[1,2,*], Megat Farez Zuhairi [1]

[1]Malaysian Institute of Information Technology, Universiti Kuala Lumpur, 1016 Jalan Sultan Ismail, 50250 Kuala Lumpur
[2]Politeknik Indonusa Surakarta, Jalan KH. Samanhudi No 31, Surakarta, Indonesia

ABSTRACT – The challenge of classifying imbalanced data persists in machine learning, particularly in critical applications such as medical diagnosis, fraud detection, and anomaly identification, where detecting the minority class is essential. Conventional classifiers like Support Vector Machine (SVM) tend to favor the majority class, leading to reduced sensitivity in identifying minority instances. This study introduces Posterior Probability and Correlation-Support Vector Machine (PC-SVM), a novel approach that integrates posterior probability estimation with correlation analysis to enhance SVM's performance on imbalanced datasets. Unlike traditional SVM models, which struggle with class imbalance and require additional data balancing techniques, PC-SVM dynamically adjusts classification thresholds using posterior probability values and correlation-weighted features, simplifying the classification process while improving its effectiveness. The effectiveness of PC-SVM was evaluated using multiple imbalanced datasets from KEEL, UCI, and Kaggle repositories. Results demonstrate that PC-SVM achieves 100% recall for the minority class, significantly outperforming traditional SVM, which attained only 80% recall on average. This 20% improvement in recall underscores PC-SVM's ability to mitigate the imbalance issue without relying on oversampling or cost-sensitive adjustments. Furthermore, PC-SVM exhibits consistent performance across various evaluation metrics, including accuracy, precision, recall, and F1-score, ensuring robust classification results. By improving the detection of minority classes, PC-SVM offers a transformative solution for real-world applications that demand high sensitivity in identifying rare but crucial instances. Its ability to maintain classification integrity without additional balancing techniques positions it as a valuable model for industries such as healthcare, finance, and cybersecurity, where accurate minority class recognition is critical.

ARTICLE HISTORY
Received: 1 January 2025
Revised: 31 January 2025
Accepted: 28 April 2025
Published: 2 May 2025

KEYWORDS
Supervised classification
Imbalanced dataset
Posterior Probability
Correlation

1.0 INTRODUCTION

Supervised categorization is the process of using a training dataset and statistical learning techniques to classify items into distinct classes and apply this knowledge to categorize new data [1]. Supervised learning, a fundamental aspect of machine learning, involves algorithms that identify patterns in data by utilizing known independent and dependent variables to predict future outcomes, with supervised classification assuming cluster labels as parameters while addressing challenges like class distribution disparity [2].

The significance of imbalanced data categorization is increasing in the fields of machine learning [3]. A dataset is considered imbalanced when one class significantly outnumbers the other, with the minority class referred to as the positive (+) class and the majority class as the negative (−) class in data categorization. [3]. The issue of class imbalanced has garnered significant attention in recent study [4]–[10]. Sampling techniques are utilized to transform the distribution of imbalanced data into a balanced distribution [11]. Undoubtedly, the issue of learning from imbalanced data sets is a significant obstacle in the field of data mining. While conventional support vector machine can typically demonstrate strong performance in handling classification problems with imbalanced data sets, they treat all training samples equally in the learning process. This can lead to a bias in the final decision boundary towards the majority class, particularly when outliers or noises are present [12]. Imbalanced data categorization occurs when one class has more examples than the other, with the majority class often overshadowing the minority class, which is treated as noise by conventional classifiers, leading to bias towards the majority class and prompting the development of various methods to address this issue. [13].

The support vector machine (SVM) is a very efficient machine learning tool that is known for its speed, simplicity, reliability, and ability to give correct classification results [14]. SVM generates a model based on the available sample sizes of each class. The SVM learning formulation is derived from the ideas of structural risk minimization. Support Vector Machine (SVM) can be employed to mitigate the constraints of the generalization error, hence enhancing its performance when applied to data outside the training set [15]. The goal of Support Vector Machine (SVM) is to identify the hyperplane that separates two classes in a vector space [16]. The separating hyperplane lies between two parallel hyperplanes, with one placing vectors of the first class above it and the other placing vectors of the second class below it, where the margin is the distance between these hyperplanes, and in cases where misclassifications are allowed for

*CORRESPONDING AUTHOR | C. A. Pamungkas | ✉ pamungkas.canggih@s.unikl.edu.my

improved generalization, the margin is "soft," while SVM remains a highly efficient method for supervised classification. [1].

Traditional classification approaches assume equal probabilities for data from different classes, but in real-world scenarios, minority classes may have fewer data points than majority classes, causing traditional algorithms to show bias towards majority classes and resulting in reduced accuracy for minority class classification [2]. To address the challenges of imbalanced data, various techniques have been proposed, categorized into three groups: data-level techniques that modify the sample probabilities through oversampling or undersampling to balance the dataset, algorithm-level techniques that adjust classification systems with cost-sensitive approaches to penalize misclassifying minority samples more heavily, and fusion approaches that combine different tactics, such as sampling and cost-sensitive methods, to tackle the imbalance issue [2].

The proposed Posterior Probability and Correlation-Support Vector Machine (PC-SVM) improves upon traditional SVM-based methods by integrating posterior probability estimation and correlation-weighted feature analysis, addressing the longstanding issue of class imbalance in machine learning. Unlike standard SVM, which constructs decision boundaries without considering class distribution, PC-SVM dynamically adjusts classification thresholds using posterior probabilities, enhancing minority class recognition. Furthermore, unlike cost-sensitive SVM or resampling techniques (e.g., SMOTE), which require manual parameter tuning or risk overfitting, PC-SVM automatically balances class representation without modifying the dataset. Existing SVM solutions struggle with biased decision boundaries that favor the majority class, limiting their effectiveness in real-world applications such as fraud detection and medical diagnosis, where minority class detection is crucial. PC-SVM fills this gap by leveraging correlation analysis to refine feature selection, ensuring that the most relevant features influence classification, thus improving recall, accuracy, and generalization across datasets.

This article mainly introduces the development of a novel classification method called PC-SVM, which integrates posterior probability and correlation techniques to enhance SVM performance on imbalanced datasets. Section 1 discusses the background and motivation for addressing class imbalance in machine learning. Section 2 reviews related works and existing approaches to tackle this issue. Section 3 outlines the research methodology, including data sources, preprocessing, modeling, and evaluation techniques. Section 4 presents and analyzes experimental results, demonstrating that PC-SVM outperforms traditional SVM methods. The final section summarizes the findings and highlights the contributions and potential of PC-SVM for real-world applications.

2.0 RELATED WORKS

A key challenge in classification learning systems is the class distribution disparity, where one or more classes have a high frequency of instances while others are underrepresented, causing conventional algorithms, including SVM classifiers, to perform well on the dominant class but exhibit bias and fail to incorporate data distribution information in addressing class imbalance [17].

M. Li developed a novel technique called ant colony optimization resampling (ACOR) to tackle the issue of class imbalance [18]. ACOR consists of two stages: first, an oversampling technique is applied to balance the dataset, followed by the use of an ant colony optimization algorithm to select a suboptimal subset, enabling the creation of an optimal training set, which has shown superior performance compared to other oversampling methods, though challenges remain, such as quick descent into local optima, slow convergence, and low precision in convergence [19].

As mentioned before, the SVM classifier exhibits bias towards the dominant class as a result of class imbalance. Furthermore, the current SVM-based approaches for addressing class imbalance lack information on the data distribution. B. Richhariya et al offer a Reduced Universum Twin Support Vector Machine for Class Imbalance Learning (RUTSVM-CIL) that is motivated by the concept of previous information on data distribution [17]. This study combines universum learning with Support Vector Machine (SVM) to address class imbalance, using oversampling and undersampling techniques alongside universum data points for prior information, and employs a compact rectangular kernel matrix to reduce computational time and storage, with the RUTSVM-CIL method demonstrating superior generalization performance and minimal computational cost on diverse datasets [17]. The sampling strategy modifies the dataset before learning, using oversampling to increase the minority class size by adding data and undersampling to decrease the majority class size by removing data, with the downside of undersampling being the potential loss of valuable information. [20].

Classification algorithms often struggle with imbalanced datasets, making effective classification challenging, particularly in detecting loose particles in sealed electronic components, which is addressed using the Synthetic Minority Over-sampling Technique (SMOTE), a standard oversampling method [21], The LR-SMOTE algorithm, designed to create new samples close to the center of the dataset and avoid generating outliers or altering the distribution, was tested on publicly available UCI datasets and custom data, showing superior performance over SMOTE in terms of G-means, F-measure, and AUC, though SMOTE's insensitivity to majority class distribution can lead to the creation of redundant minority samples, worsening issues for borderline and noisy instances [19].

X. Tao has presented a novel approach called Affinity and Probability-based Fuzzy Support Vector Machine (ACF SVM) [12]. The proposed ACFSVM approach utilizes an SVDD model trained on majority class samples in kernel space

to detect anomalies and edge samples, applies the kernel k-nearest neighbor technique to reduce noise impact, and enhances classification by prioritizing minority class importance while achieving superior performance on imbalanced UCI datasets.

Class imbalance in real-world datasets causes bias towards the dominant class, resulting in poor performance for the minority class, which can lead to unreliable outcomes in critical applications like illness detection, prompting researchers to focus on addressing this issue through hybrid techniques [22]. The study uses simulated annealing for undersampling and applies support vector machine, decision tree, k-nearest neighbor, and discriminant analysis for classification, validating the approach on 51 real-world datasets and demonstrating superior effectiveness in reducing misclassification compared to previous methods.

The conventional SVM models are extensively employed across many domains. However, research indicates that they lack a coherent geometric definition, which might potentially undermine their theoretical performance, particularly in high-dimensional scenarios [23]. K. Qi et al [23] The study explores the use of a combined penalty and introduces an elastic net support vector machine (ENSVM), which penalizes slack variables rather than the hyperplane's normal vectors, showing that ENSVM outperforms standard SVM and Doubly Regularized SVM (DrSVM) in terms of logical specification, stability, and high-dimensional characteristics, while the integration of fused weights leads to the adaptive weighted ENSVM (AWENSVM), which enhances adaptability and robustness in handling imbalanced data and shows superior performance compared to other common SVM models.

Most classification techniques assume even sample distribution across classes, but this leads to biased performance toward the dominant class; the proposed Enhanced Automatic Twin Support Vector Machine (EATWSVM) method addresses this issue by integrating a kernel representation into the optimization of Twin Support Vector Machines (TWSVM) using a Gaussian similarity based on Mahalanobis distance, enhancing data distinguishability with a centered kernel alignment strategy, and determining regularization parameters based on the imbalance ratio and dataset overlap, with experimental results showing superior performance and training efficiency compared to state-of-the-art methods [13].

Imbalanced data can result in unsatisfactory classification models, since it often leads to frequent misclassification of minority cases and hinders the achievement of optimal performance. H. Shamsudin presents an enhanced approach called SVM-GA for handling imbalanced data. This method optimizes the SVM algorithm using Genetic Algorithm (GA) in combination with a synthetic minority oversampling strategy [24]. The experimental results show that the proposed method improves performance by 97% compared to the baseline and optimized models, outperforming SVM with Grid and Randomized search, especially for datasets with rare instances; however, the study is limited by small sample sizes (under 5000) and fewer features, suggesting the model's performance may need further testing with more complex datasets to assess its efficacy in more intricate settings.

Imbalanced data categorization is crucial in machine learning as it focuses on detecting abnormalities, which are often of interest, but occur less frequently than regular instances in real-world systems. Developing classifiers using imbalanced data can be challenging since there is no definitive criterion for determining the extent of imbalance that can be considered as imbalanced or balanced [25]. In order to tackle this problem, this article suggests enhancing the current Support Vector Machine technique.

A key aspect of classification involves estimating the likelihood of a new point x belonging to a specific class, where most classifiers generate a score linked to posterior probability through an implicit relationship, with models like SVM providing an approximate posterior probability that represents the likelihood of a class being assigned to a dataset, which can be refined for greater accuracy.

A major challenge in data mining is handling imbalanced data in classification tasks, where some classes (majority classes) have a large amount of data, and others (minority classes) have only a few instances, leading to biased performance in traditional classifiers that focus on data distribution rather than error rates, often neglecting the minority classes in the classification outcome, a problem common in many real-world applications [26].

3.0 METHODS AND MATERIAL

The research methodology consists of five distinct phases : Data Sources, Data Preparation, Experiment, Modelling, Model Evaluation.

3.1 Data Sources

The research issue necessitates the use of data in order to provide a response. The research resources utilized in this study consist of publicly available data sets obtained from the Knowledge Extraction based on Evolutionary Learning (KEEL) Database, the UC Irvine (UCI) Machine Learning Repository, and Kaggle.

Details of the dataset employed in this article is shown in Table 1. The highest imbalanced ratio is the chrun dataset with IR 5,36 and the lowest imbalanced ratio is heart with IR 2.11. While the lowest instances are 299 in the heart dataset, the highest instances are 3150 in churn dataset. The highest features are churn which has features 13, and the lowest features are yeast dataset which has 8 features.

Table 1. Detail Dataset

Dataset	Features	Instances	Positive Instances (%)	Negative Instances (%)	IR	Public dataset
Yeast	8	1484	28,9	71,1	2,46	Kaggle
Heart	12	299	68	32	2,11	Keel
Churn	13	3150	16	84	5,36	UCI

3.2 Data Preparation

Data preprocessing is a crucial undertaking in machine learning that has the potential to greatly enhance the results of a model [27] [28] [29]. Data pre-processing is an essential and pivotal stage in the life cycle of machine learning. A significant obstacle in the healthcare industry is obtaining a comprehensive and uncontaminated dataset. The data quality is of utmost importance as it can significantly impact the model's learning capacity and its final generalizability [30]. Efficient and accurate algorithms may be achieved by employing effective data preparation methods and procedures. This serves as a strong basis for making data-driven decisions and developing applications [31]. The data underwent preprocessing techniques including data encoding, data imputation, transformation of skewed data, data balancing, feature scaling, and feature selection [32] [33] [34]. Data preprocessing encompasses a collection of methods aimed at improving the integrity of the original data, including the elimination of outliers and the filling in of missing values [35]. An essential stage in the data analysis process is preprocessing, which entails transforming unprocessed data into a format that can be comprehended by computers and machine learning algorithms. This crucial stage significantly influences the accuracy and effectiveness of machine learning models [36]. The data preparation process involves key stages, including missing value detection and data transformation.

3.3 Experiment

This research includes a series of experiments to evaluate classification performance under various conditions: (1) assessing the performance of SVM with an imbalanced dataset to examine its bias towards the majority class, (2) evaluating SVM with a balanced dataset to observe improvements after addressing class imbalance, and (3) testing the PC-SVM model with an imbalanced dataset to determine its effectiveness in mitigating class imbalance while maintaining classification accuracy.

3.4 Modelling

Model development refers to the process of creating and refining models in the context of data analysis and machine learning. A model is a mathematical or statistical representation of a system or process that aims to uncover patterns or relationships within data.

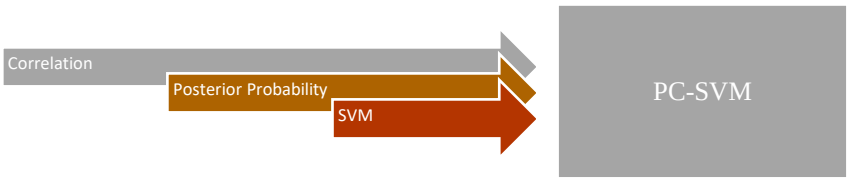


Figure 1. Proposed Algorithm

The proposed idea shown by Figure 1 combines the algorithm referred to as PC-SVM stands for Posterior Probability and Correlation-Support Vector Machine. The fundamental principle behind the PC-SVM method is combination of Posterior Probability and Correlation Techniques which is very effective in improving SVM performance on imbalanced data sets. In an imbalanced data set, when one class is larger than the other, posterior probability provides the advantage of calculating class probabilities based on feature likelihood, thereby providing insight into the probability of a sample belonging to the minority or majority class. The probability distribution is optimized by multiplying the prior probability with the total sum of the multiplication of the R Square of feature i of class Y with the independent probability of all feature vectors X. The attribute weights in the proposed method are obtained from the correlation coefficient value between the attribute and the class. The correlation coefficient has a value range from -1 to 1, so there is a possibility that the attribute weighting value will be negative. To prevent negative values from occurring, what is used for attribute weighting is the R Square value. Attribute weighting is a method where the R Square value of each attribute for the class is multiplied by the probability of each attribute in calculating the conditional probability of the Naive Bayes Classifier using the join probability method. Figure 2 illustrates the general architecture of the proposed algorithm :

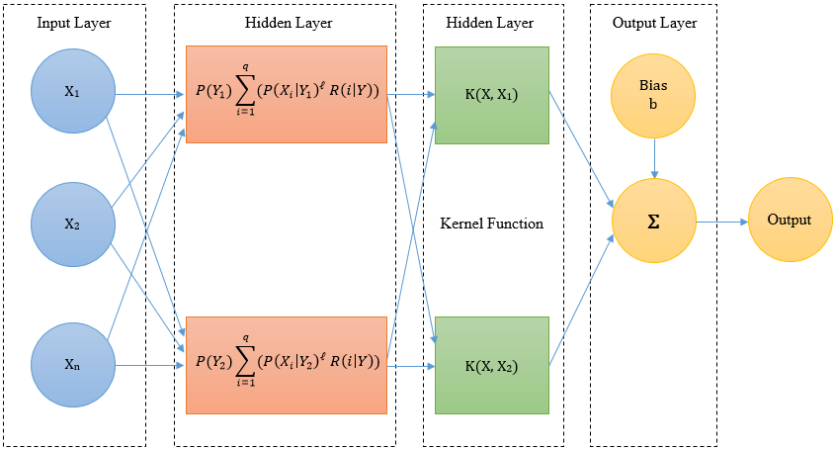


Figure 2. General architecture of the proposed algorithm

This attribute weighting can increase the accuracy by considering how strongly the attribute is related to the specified class. To maintain consistency that the R Square value always influences the classification process, the Laplacian method is used which prevents probability 0 from occurring. The Laplacian method needs to be applied because if there is a probability of 0, then whatever the R Square value of the attribute will have no effect. The application of the Laplacian method is to force the frequency of data occurrences in the data set to be greater than 0 by adding an occurrence value of 1 to each conditional probability calculation. The basic concept of attribute weighting is to assume that each attribute has a different influence and priority on the class. To optimize classification results, the research that will be carried out will also use the Laplacian method to overcome the Zero Probability problem.

These posterior probabilities are then used as input features for the SVM, which excels at finding the optimal separating hyperplane between classes. By using psoterior probabilities to convert the original feature space into probability estimates, SVM can focus on maximizing the margin between classes using these probabilities. This helps mitigate imbalances by making decision boundaries more sensitive to minority class examples, increasing classification accuracy, and reducing bias towards the majority class.

The PC-SVM algorithm employs an attribute weighting technique based on R Square. The R Square weighting in the PC-SVM technique may be shown in Figure 3.

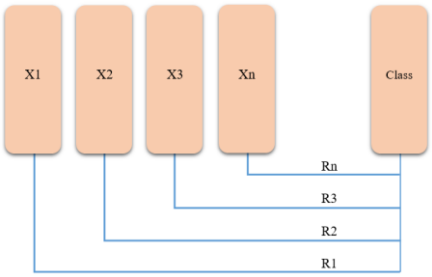


Figure 3. R Square Weighting

The SVM method utilizes joint probability to calculate conditional probability. characteristic weighting is a method that assigns a numerical value to each characteristic to indicate its relative importance. Thus, the accurate approach in conditional probability involves employing the method of total addition rather than complete multiplication. The posterior probability is employed to ascertain the degree of confidence in a categorization.

Posterior probability alone only provides probabilities without considering the strength of relationships between features. By incorporating R Square, the model becomes richer in information, as now each feature is not only measured based on its likelihood distribution but also on its relevance to the target class. This results in a deeper probability structure, which is important when dealing with imbalanced datasets, where most features may be more relevant to the majority class and less helpful in detecting the minority class. By combining R Square with the independent probabilities of features, we effectively make the model more sensitive to variations and patterns present in features closely related to the minority class. In the context of an imbalanced dataset, this approach helps the model capture subtle patterns important for detecting the minority class, which often gets lost amidst the dominance of the majority class.

3.5 Model Evaluation

Model evaluation in this study involves the confusion matrix, ROC curve, and scatter plot. The confusion matrix, applicable to both binary and multiclass problems, presents actual versus predicted classifications and supports performance metrics such as Accuracy, Precision, Recall, and F1 Score, based on TTP, TFN, TTN, and TFP [37] [38].

The ROC curve assesses classifier performance across thresholds, with AUC indicating the model's ability to distinguish between classes particularly useful in imbalanced data scenarios. Additionally, scatter plots visualize relationships and correlations between two variables, aiding in the interpretation of data patterns.

4.0 RESULTS AND DISCUSSION

4.1 Correlation

Correlations are widely utilized statistical processes that serve as a foundation in several applications, including exploratory data analysis, structural modeling, and data engineering [39]. Equation (1) is used to determine the correlation value and equation (2) is used to obtain R Square.

$$r = \frac{\sum(X_i - \bar{X})\,(Y - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2 \sum(Y - \bar{Y})^2}} \tag{1}$$

$$R = r^2 \tag{2}$$

Let X be a vector with an unknown class label, consisting of features indexed by $i = \{1, \ldots, q\}$, where $R(i|Y)$ denotes the attribute weights computed based on the coefficient of determination (R-squared) for each feature with respect to class Y. The R is determined through the application of the equation (2). The coefficient of determination R Square for all features within vector X in relation to the dependent variable Y is calculated using equation (3):

$$R(i|Y) = R \tag{3}$$

with

R	:	r Square
r	:	correlation coefficient value
$\bar{X}$	:	Mean of the attribute X_i
$\bar{Y}$	:	Mean of Y
$R(i\|Y)$	:	r Square attribute i to class Y

When posterior probability is used in classification tasks, it generally only evaluates the probability of class membership based on the observed data. However, this approach can be limiting as it assumes that all features contribute equally and independently to the classification task, ignoring the relationship strength between individual features and the target class.

The use of R Square in the equation changes this dynamic by providing a measure of how much of the variance in the target class is explained by each feature. By incorporating R Square into the model, gained additional information about how relevant each feature is to the target class, going beyond simple probability estimations. For example, if a particular feature has a high R Square value, it is strongly correlated with the target class and should be given more weight in the classification decision.

4.2 Posterior Probability

Probability theory is a scientific discipline that use statistical methods to understand occurrences that occur randomly [40]. Possibility theory is based on two basic principles. These Prior probability and posterior probability. The posterior probability is the likelihood of an event occurring, which is computed after taking into account all available information or data [40]. In order to calculate the posterior probability for the PC-SVM method, it is necessary to choose the highest value among the numerous prior probabilities, using conditional probability. The equation (4) is utilized to calculate the posterior probability.

$$Posterior\ Probability = \max\left(P(Y) \sum_{i=1}^{q} P(X_i|Y) \right) \tag{4}$$

Posterior probability and correlation analysis are powerful techniques that can complement each other in predictive modeling. This probabilistic output allows for more nuanced decisions in classification tasks. On the other hand, correlation analysis evaluates the strength and direction of relationships between variables. By combining these methods can enhance model performance by understanding feature dependencies and selecting the most relevant variables. The equation (5) is Posterior Probability with Correlations.

$$Posterior\ Probability\ with\ \text{Correlations} = \max\left(P(Y) \sum_{i=1}^{q} (P(X_i|Y)\ R(i|Y)) \right) \tag{5}$$

The Laplacian method is employed as a technique for estimating conditional probabilities by addressing the issue of zero-frequency occurrences. This is achieved by adding one to the frequency count of each occurrence of X_i in the dataset. As outlined in Equation (5), the application of the Laplacian adjustment leads to a revised formulation of the posterior probability, which is mathematically represented in Equation (6).

$$Posterior\ Probability\ with\ \text{Correlations and Laplacian} = \max\left(P(Y) \sum_{i=1}^{q} (P(X_i|Y)^{\ell}\ R(i|Y)) \right) \tag{6}$$

Based on equation (6) by combining Posterior Probability with Correlation and Laplacian, it can create a classification model that is more robust and able to handle shortcomings that arise due to the assumption of independence and lack of data in the minority class. This combination provides several advantages:

- Overcoming zero probability in data that does not appear in training, especially in minority classes.
- Gives additional weight to features that are more correlated with the target class, which is especially important in the case of imbalanced datasets.
- Ensure that the posterior probability is more representative, both in terms of feature probability and its relevance to the target, thereby increasing the model's ability to detect minority classes.

This approach is effective in situations where imbalance in class distribution makes classification a challenge, by increasing the model's sensitivity to underrepresented classes.

4.3 Posterior Probability And Correlation-Support Vector Machine (PC-SVM)

In the combination of posterior probability and SVM, it uses the product of the prior probability and the total sum of the products of R Square feature i of class Y with Independent Probability of all attribute vectors X as input features for the SVM model. The SVM will learn to use these features to separate the classes by the largest margin.

4.3.1 Probability Posterior from Naive Bayes

Naive Bayes calculates the posterior probability of each class Y_1 and Y_2 based on the attribute $X = (x_1, x_2, \ldots, x_n)$ of the sample.

$$P(Y_1|X) = P(Y_1) \sum_{i=1}^{q} (P(X_i|Y_1)^{\ell}\ R(i|Y)) \tag{7}$$

$$P(Y_2|X) = P(Y_2) \sum_{i=1}^{q} (P(X_i|Y_2)^{\ell}\ R(i|Y)) \tag{8}$$

Where $P(Y_1)$ and $P(Y_2)$ are the prior probabilities of classes Y_1 and Y_2, and $P(X_1|Y_1)$, $P(X_2|Y_2)$ are the conditional probabilities of attribute X in each class. After getting the probabilities $(Y_1|X)$ and $P(Y_2|X)$, these probabilities are used as new input attributes for the SVM model.

4.3.2 SVM Formulation with Naive Bayes Features

SVM aims to find a hyperplane $f(z)$ that separates two classes Y_1 and Y_2 based on a new input attribute $z = (P(Y_1|X), P(Y_2|X))$. The decision function for SVM, given a feature vector z is mathematically expressed by equation (9):

$$f(z) = w^{\top} z + b \tag{9}$$

subject to,

$$\hat{y} = sign\big(f(z)\big) = sign(w_1.z + b\,) \tag{10}$$

Where :

- w is the weight vector of the SVM
- $z = (P(Y_1|X), P(Y_2|X))$ is a feature vector consisting of Naive Bayes posterior probabilities
- b is the bias of the SVM.
- $f(z)$ determine which class the sample belongs to:
  - If $f(z) > 0$, then the sample is predicted as Y_1 (positive class).
  - If $f(z) \leq 0$, then the sample is predicted as Y_2 (negative class).

4.3.3 Combination Formula of SVM and Naive Bayes

By combining the Naive Bayes posterior probabilities into the SVM, the combination formula becomes:

$$f(z) = w_1.P(Y_1|X) + w_2.P(Y_2|X) + b \tag{11}$$

Input to SVM: $P(Y_1|X)$ and $P(Y_2|X)$ are the posterior probabilities.

4.3.4 Margin Optimization in SVM

SVM maximizes the margin between classes Y_1 and Y_2 by solving the following optimization problem:

$$\min_{w,b} \frac{1}{2} \|w\|^2 \tag{12}$$

with condition

$$Y_i(w^\top z_i + b) \geq 1 \; \forall_i \tag{13}$$

Where:
- Y_1 is the original class label of the i sample (1 for Y_1, -1 for Y_2).
- $z_i = [P(Y_1|X_i), P(Y_2|X_i)]$ is the feature vector from Naive Bayes for the i sample.

4.3.5 Final Decision

The final prediction is determined by the decision function $f(z)$, where Naive Bayes and SVM work synergistically:

$$Posterior\ Probability = \max\left(P(Y) \sum_{i=1}^{q} (P(X_i|Y)\ R(i|Y)) \right) \tag{14}$$

In mathematical terms, $Posterior\ Probability\ (Y|X)$ is computed using equation (15):

$$Posterior\ Probability\ (Y|X) = \max\left(\frac{P(Y) \sum_{i=1}^{q} (P(X_i|1)^{\ell}\ R(i|Y))}{P(X)} \right) \tag{15}$$

where w is the weight vector, z is the feature vector, and b is the bias

After training, for a new sample X, the Naive Bayes component computes the posterior probabilities $P(Y_1|X)$ and $P(Y_2|X)$, which are then fed into the SVM. The SVM makes the final prediction using the decision function:

$$\hat{y} = sign\big(f(z)\big) = sign(w_1.P(Y_1) \sum_{i=1}^{q} (P(X_i|\ Y_1)^{\ell}\ R(i|Y)) +\ w_2.P(Y_2) \sum_{i=1}^{q} (P(X_i|\ Y_2)^{\ell}\ R(i|Y)) + b) \tag{16}$$

The final SVM decision function is:

$$\hat{y} = sign\big(f(z)\big) = sign(w_1.P(Y_1|X) +\ w_2.P(Y_2|X) + b) \tag{17}$$

Where:
- $f(z)$ is the decision score: the sign of $f(z)$ determines the class assignment.
- $P(Y_1|X)$ and $P(Y_2|X)$ are posterior probabilities from Naive Bayes.
- w_1 and w_2 are the SVM weights.
- b is the bias term learned by SVM.

The class prediction is made by evaluating the sign of $f(z)$:
- If $f(z) > 0$, the prediction is Y_1.
- If $f(z) \leq 0$, the prediction is Y_2.

4.1 Data Preprocessing

4.1.1 Missing Value Detection

The result analysis of missing values in the yeast dataset indicates a complete dataset, as evidenced by the absence of missing values across all attributes. Each attribute, including Mcg, Gvh, Alm, Mit, Erl, Pox, Vac, Nuc, and Class, contains a total of 1484 valid entries, with no entries recorded as missing. This results in a 0% missing value percentage for each attribute, demonstrating that the dataset is fully populated. The completeness of the yeast dataset is a significant advantage, ensuring that the data is ready for further analysis and modeling without the need for imputation or any corrective measures. This reliable data quality enhances the confidence in the analytical outcomes and predictive performance of any models developed using this dataset.

The result examination of missing values in the heart dataset indicates that there are no missing entries for any attributes, which is a noteworthy discovery. Every attribute comprises a total of 299 valid entries. This yields a 0% incidence of missing values universally. The lack of missing values across all attributes signifies that the dataset is comprehensive and prepared for analysis without requiring data imputation or rectification. The high-quality data is essential for guaranteeing the robustness and reliability of analyses or predictive modeling conducted on the dataset, hence improving the possibility for precise insights into the factors affecting heart-related outcomes.

The examination of missing values in the churn dataset reveals the absence of any missing data entries for any of the attributes, which is a commendable result. Each attribute contains 3150 valid entries. This results in a 0% missing value percentage across all attributes. The complete dataset signifies a high level of data integrity, as the absence of missing

values eliminates the need for data imputation or any corrections. This completeness is crucial for conducting thorough analyses and building predictive models, ensuring that the insights derived from the data are based on a robust foundation and enhancing the reliability of the conclusions drawn regarding customer behavior and churn factors.

4.2 Data Transformation

The transformation of the yeast dataset was crucial for preparing it for posterior probability calculations and correlation analysis, as it involved converting various attributes from float64 to Category type. Initially, attributes were represented as continuous numerical values, which could misrepresent their categorical nature in subsequent analyses. By transforming these attributes into categorical data types, the analysis can accurately interpret and handle the relationships between different classes without assuming a linear relationship, which is common with continuous variables. This conversion is particularly important for calculating posterior probabilities, where the models need to treat these attributes as distinct categories rather than as continuous scales. Additionally, transforming the data into categories enhances the correlation analysis by ensuring that the statistical methods applied are appropriate for categorical data, allowing for the identification of meaningful relationships between different yeast characteristics. Overall, this data transformation optimizes the dataset for effective statistical modeling and ensures accurate results in both posterior probability and correlation computations.

The transformation of the heart dataset was essential for its preparation for posterior probability calculations and correlation analysis, involving the conversion of various attributes from their original numerical types to Category. Initially, attributes like Age, Platelets, and serum_creatinine were represented as float64, while others, such as Anaemia, creatinine_phosphokinase, and DEATH_EVENT, were int64. By converting these variables into categorical data types, the dataset better reflects the inherent characteristics of the variables, allowing for appropriate statistical analyses. This is particularly important for posterior probability estimation, as treating variables like Age and ejection_fraction as categories prevents misinterpretation of their values as continuous data, which could lead to erroneous conclusions. Additionally, transforming all relevant attributes to Category enhances the validity of correlation analyses, as it allows for the examination of relationships between discrete classes rather than assuming a linear relationship between continuous variables. Ultimately, this transformation aligns the dataset with the requirements of the analytical methods to be applied, facilitating more accurate insights into the relationships among various heart-related factors and their impact on the outcome of interest.

The transformation of the churn dataset was vital for preparing it for posterior probability calculations and correlation analysis, entailing the conversion of various attributes from their original integer types (Int64) and floating-point types (float64) into Category types. Initially, attributes such as Call Failure, Complains, and Subscription Length were represented as integer values, which could imply a numerical relationship that does not accurately reflect their categorical nature. By transforming these attributes into categorical types, the dataset aligns more closely with the intended analysis, allowing each attribute to be treated as a distinct category rather than a continuous variable. This is particularly important for posterior probability estimation, as categorical data can provide clearer insights into the likelihood of churn based on various factors, such as Age Group or Tariff Plan, without the misleading implications of numerical scales. Furthermore, the transformation enhances the correlation analysis by facilitating the exploration of relationships between categorical variables, enabling a more meaningful interpretation of how different attributes interact and influence customer churn. Overall, this data transformation optimizes the dataset for effective statistical modeling, ensuring accurate results in posterior probability and correlation computations related to customer behavior.

4.3 Performance SVM With Imbalanced Dataset

The results of performing SVM with an imbalanced dataset are displayed in Figure 4, Figure 5, and Figure 6. The performance evaluation include metrics such as accuracy, precision, recall, and FI-score.

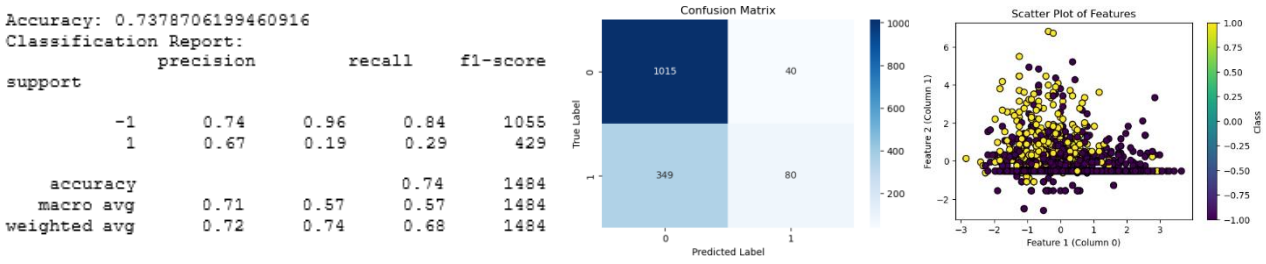


Figure 4. Performance of SVM on Imbalanced Datasets Utilizing the Yeast Dataset

Figure 4 shows that the SVM model's accuracy on the imbalanced yeast dataset is 73.79%, which seems low due to the class imbalance. The majority class (-1) has 1055 instances, while the minority class (1) has only 429, causing the model to be biased towards the majority class. For the majority class, the model performs well with a precision of 0.74 and a high recall of 0.96, leading to a strong F1-score of 0.84. However, the performance for the minority class is much weaker, with a precision of 0.67 and a very low recall of 0.19, resulting in a poor F1-score of 0.29. The confusion matrix shows many False Negatives for the minority class, indicating the model struggles to identify these instances. The scatter

plot confirms this, showing clear separation for the majority class but less distinct separation for the minority class, leading to misclassifications. The macro average shows a precision of 0.71 and a recall of 0.57, reflecting the imbalance, while the weighted average improves slightly but still highlights the issue with the minority class.for the imbalance, gives slightly better scores, but the low recall for class 1 remains a key issue.

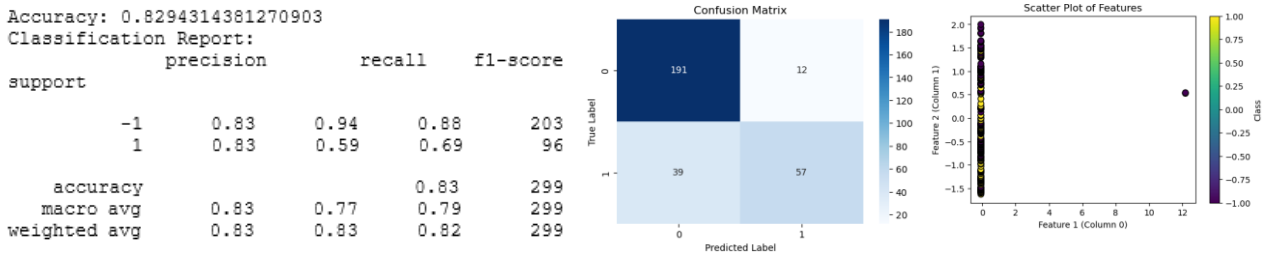


Figure 5. Performance of SVM on Imbalanced Datasets Utilizing the Heart Dataset

Figure 5 shows that the SVM model achieves an overall accuracy of 82.94% on the imbalanced heart dataset, suggesting good general prediction. However, accuracy alone doesn't fully reflect the model's performance due to class imbalance. The dataset has a majority class (-1) with 203 instances and a minority class (1) with 96 instances, showing clear performance differences. For the majority class, the model performs well, with a precision of 0.83, a recall of 0.94, and an F1-score of 0.88. However, for the minority class, precision is also 0.83, but recall drops to 59%, resulting in a lower F1-score of 0.69. The confusion matrix shows a high number of True Negatives for the majority class and many False Negatives for the minority class, indicating the model struggles to identify the minority class. The scatter plot further shows that the majority class is well-separated, but the minority class instances are more scattered, leading to misclassifications. The macro average shows high precision at 0.83 but a lower recall of 0.77, highlighting the imbalance, while the weighted average reflects the overall accuracy, still showing the model's difficulty with the minority class.

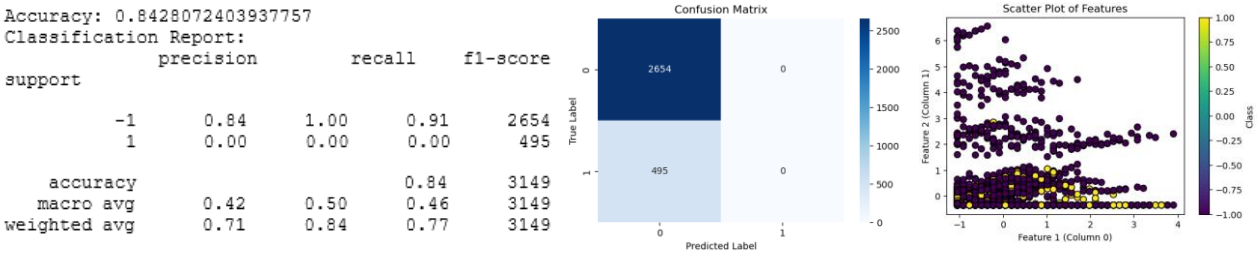


Figure 6. Performance of SVM on Imbalanced Datasets Utilizing the Churn Dataset

Figure 6 shows that the SVM model's performance on the imbalanced churn dataset highlights the issue of class imbalance, with an overall accuracy of 84.28%. However, this accuracy is misleading due to the stark difference between the majority class (-1, non-churn) with 2654 instances and the minority class (1, churn) with 495 instances. The model performs well for the majority class, achieving a precision of 0.84, recall of 100%, and an F1-score of 0.91, meaning it correctly identifies nearly all non-churners. However, the confusion matrix shows that the model misclassifies all churn instances as non-churn, resulting in a recall of 0% for the minority class. The scatter plot further highlights this, showing that the decision boundary favors the majority class, causing many churn instances to be misclassified. As a result, the model fails to predict churners, with a recall and F1-score of 0 for class 1. The macro average scores (precision 0.42, recall 0.50, F1-score 0.46) emphasize the model's poor generalization across both classes, while the weighted average F1-score of 0.77 reflects the model's strong performance on the majority class but hides its failure with the minority class.

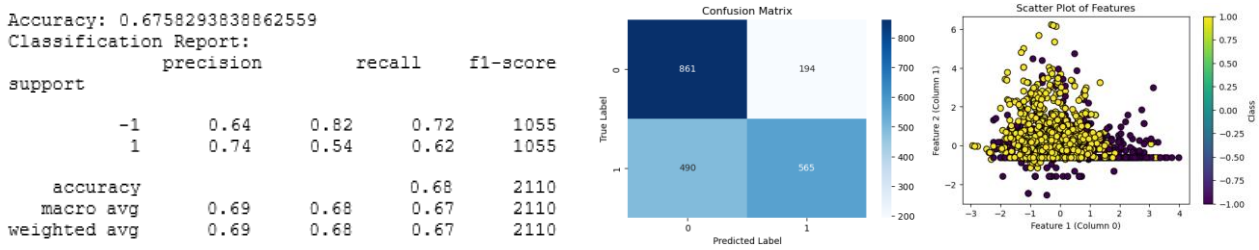
4.4 Performance Svm With Balanced Dataset

The results of performing SVM with a balanced dataset are presented in

Figure 7,

Figure 8, and

Figure 9. The preprocessing technique used to mitigate class imbalance is SMOTE. The performance measurements utilized comprise accuracy, precision, recall, and F1-score.

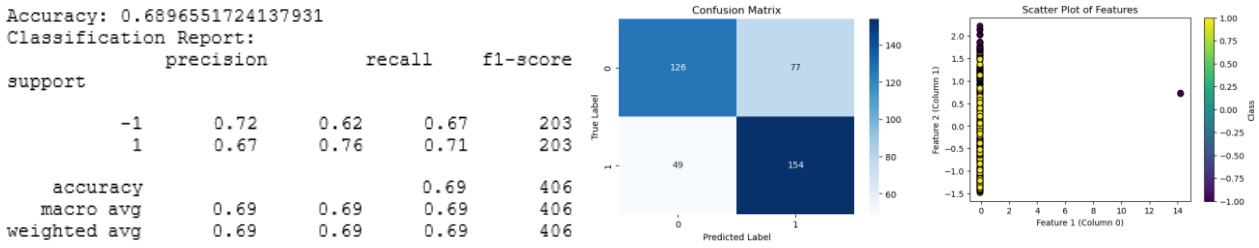


Accuracy: 0.6758293838862559
Classification Report:

	precision	recall	f1-score	support
-1	0.64	0.82	0.72	1055
1	0.74	0.54	0.62	1055
accuracy			0.68	2110
macro avg	0.69	0.68	0.67	2110
weighted avg	0.69	0.68	0.67	2110

Figure 7. Performance of SVM on Balanced Datasets Utilizing the Yeast Dataset

Figure 7 shows that the SVM model's performance on the balanced yeast dataset is moderately effective, with an overall accuracy of 67.58%, meaning it correctly predicts more than two-thirds of the time. However, this accuracy doesn't fully reflect how the model performs for both classes, which have an equal number of instances (1055 each for classes -1 and 1). The confusion matrix reveals that for the majority class (-1), the model has a high recall of 82% but a lower precision of 0.64, indicating it misclassifies some class 1 instances as class -1. This results in an F1-score of 0.72. For the minority class (1), the model's precision is higher at 0.74, but the recall drops to 54%, showing it misses many class 1 instances, leading to an F1-score of 0.62. The scatter plot shows that class -1 instances are more accurately classified, while class 1 instances are more spread out and misclassified. The macro average scores (precision 0.69, recall 0.68, F1-score 0.67) suggest the model is fairly balanced across the classes, but with slightly better performance for the majority class. The weighted average F1-score of 0.67 confirms that while the model is consistent, it still shows some bias toward the majority class despite the balanced dataset.

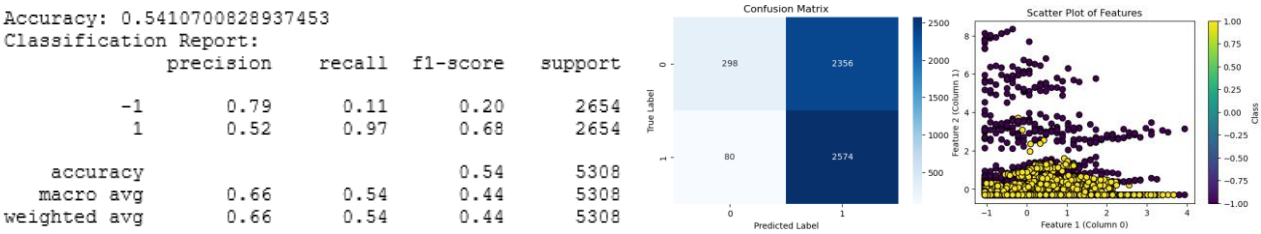


Accuracy: 0.6896551724137931
Classification Report:

	precision	recall	f1-score	support
-1	0.72	0.62	0.67	203
1	0.67	0.76	0.71	203
accuracy			0.69	406
macro avg	0.69	0.69	0.69	406
weighted avg	0.69	0.69	0.69	406

Figure 8. Performance of SVM on Balanced Datasets Utilizing the Heart Dataset

Figure 8 shows the classification results of the balanced heart dataset using SVM, where the use of SMOTE (Synthetic Minority Over-sampling Technique) helped address class imbalance and improve the model's ability to distinguish between the two classes. While the model's accuracy is 68.97%, indicating moderate effectiveness, there's still room for improvement. The confusion matrix shows that for class -1, the model has a precision of 0.72 but a lower recall of 0.62, meaning it misses some true class -1 instances. For class 1, the recall is higher at 0.76, meaning it correctly identifies more class 1 instances, but the precision is lower at 0.67, indicating a higher rate of False Positives. The scatter plot shows that class 1 instances are more spread out, which leads to some overlap with class -1 and challenges the model in perfectly separating the classes. The F1-scores are 0.67 for class -1 and 0.71 for class 1, showing slightly better performance for class 1, especially in terms of recall, which is important in medical contexts. The macro averages for precision, recall, and F1-score are all 0.69, indicating a balanced performance between the classes. The weighted averages confirm the model's consistent performance, making it reliable for this balanced dataset.



Accuracy: 0.5410700828937453
Classification Report:

	precision	recall	f1-score	support
-1	0.79	0.11	0.20	2654
1	0.52	0.97	0.68	2654
accuracy			0.54	5308
macro avg	0.66	0.54	0.44	5308
weighted avg	0.66	0.54	0.44	5308

Figure 9. Performance of SVM on Balanced Datasets Utilizing the Churn Dataset

Figure 9 shows the classification results of the balanced churn dataset using SVM, which highlights mixed performance with notable differences between the two classes. The overall accuracy of 54.11% suggests that the model is barely performing better than random guessing. Despite the balanced dataset, the model struggles to classify churn and non-churn instances effectively. Looking at the confusion matrix, for class -1 (non-churn), the model has a low recall of 11%, meaning it fails to identify most non-churn instances. While its precision is higher at 0.79, indicating that when it predicts non-churn, it is often correct, the low recall results in a poor F1-score of 0.20, showing poor performance in identifying non-churn cases. For class 1 (churn), the recall is much higher at 97%, showing the model is good at detecting churn cases. However, its precision is lower at 0.52, meaning many predicted churn instances are incorrect, which leads

to a higher rate of False Positives. This reduces precision but gives a better balance between precision and recall, reflected in the F1-score of 0.68. The scatter plot suggests a lot of overlap between the classes, particularly for class -1, which likely causes the low recall for non-churn instances. The decision boundary may be placed in a way that favors detecting churn but misclassifies non-churn instances. The macro average scores show limited ability to generalize, with a precision of 0.66 and recall of 0.54, leading to an F1-score of 0.44, indicating the model is biased towards churn detection. The weighted averages also show an F1-score of 0.44, confirming that the model needs improvement to balance its predictions better between churn and non-churn cases.

4.5 Performance Evaluation of the PC-SVM Model on Imbalanced Datasets

The results of performance PC-SVM with an imbalanced dataset are presented in Figure 10,

Figure 11, and

Figure 12. The performance metrics assessed include accuracy, precision, recall, and F1-score.

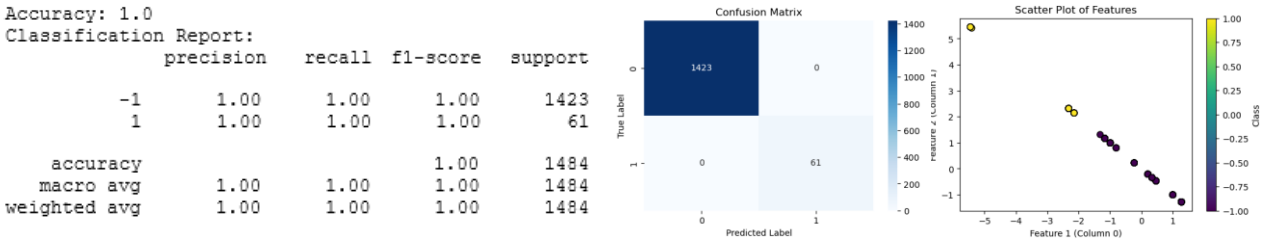


Figure 10. Performance of PC-SVM on Imbalanced Datasets Utilizing the Yeast Dataset**Figure 10** shows that the PC-SVM model performs exceptionally well on the imbalanced yeast dataset, achieving a perfect accuracy of 100%. This means the model correctly predicted all instances without any errors, demonstrating its effectiveness and reliability in classifying the yeast data. The confusion matrix confirms this strong performance, as all instances from both classes (-1 and 1) are correctly classified, with no misclassifications. This matrix only shows True Positives and True Negatives, meaning there are no False Positives or False Negatives. This indicates that the PC-SVM model can accurately distinguish between the two classes, even with the class imbalance (61 instances of class 1 vs. 1423 of class -1). A scatter plot of the results would likely show a clear separation between the two classes, with no overlap. Class -1 instances would form a dense cluster, while class 1 instances would be clearly distinguishable, reinforcing the model's ability to handle class imbalance and classify both classes accurately. The classification report shows perfect scores for both classes, with precision, recall, and F1-score all at 1.00. This means the model is 100% accurate in its predictions, with no false positives or false negatives, and an optimal balance between precision and recall. The support values show 1423 instances for class -1 and 61 for class 1. Despite the imbalance, the PC-SVM method handles the disparity well, achieving perfect classification for both classes. The macro and weighted average scores are also 1.00, further highlighting the model's consistent performance across the dataset. This outstanding performance demonstrates the power of PC-SVM for handling imbalanced datasets in classification tasks.

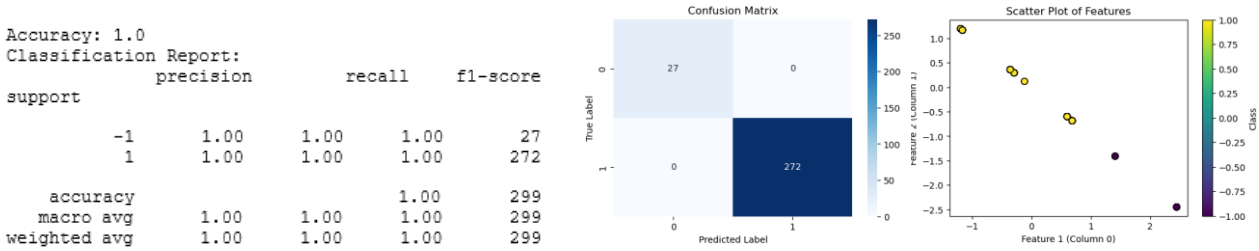


Figure 11. Performance of PC-SVM on Imbalanced Datasets Utilizing the Heart Dataset

Figure 11 shows that the PC-SVM model performs exceptionally well on the imbalanced heart dataset, achieving a perfect accuracy of 100%. This means the model correctly classified all instances without any errors, highlighting its ability to accurately distinguish between the two classes. The confusion matrix further confirms this, showing that all instances for both classes (-1 and 1) were classified correctly, with no False Positives or False Negatives. This suggests that the PC-SVM model is highly effective, even with the class imbalance in the dataset. A scatter plot of the results would likely show clear separation between the two classes, with no overlap. It would clearly identify all instances of class -1 (no heart disease) and class 1 (with heart disease), reinforcing the model's perfect classification and its ability to handle the class imbalance effectively. The classification report shows perfect results for both classes, with precision, recall, and F1-score all at 1.00. This means that the model's predictions are flawless, with no false positives or false negatives, and a perfect balance between precision and recall. The support values show 27 instances of class -1 and 272 of class 1. Despite the imbalance, the PC-SVM model classifies all instances perfectly. Both the macro and weighted average metrics are also 1.00, highlighting the model's consistent performance across both classes. This impressive

performance demonstrates the power of PC-SVM in classifying imbalanced datasets, particularly in important fields like healthcare.

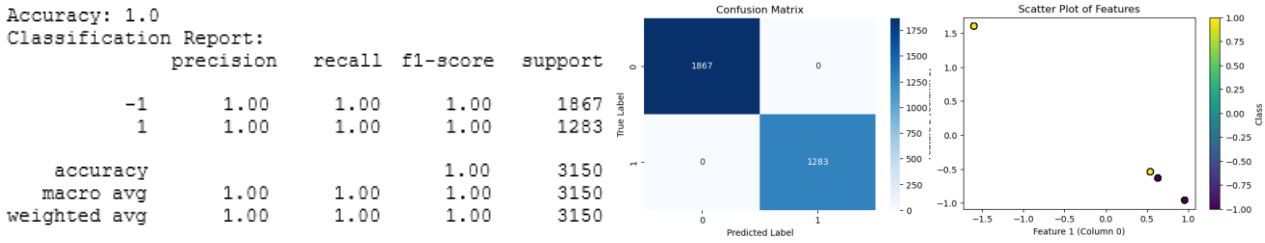


Figure 12. Performance of PC-SVM on Imbalanced Datasets Utilizing the Churn Dataset

Figure 12 shows that the PC-SVM model performs exceptionally well on the imbalanced churn dataset, achieving perfect accuracy of 100%. This means the model accurately classified every instance in the dataset, distinguishing between the two classes without any errors, even with the class imbalance. The confusion matrix confirms this, showing that all instances for both classes (-1 for non-churned customers and 1 for churned customers) were correctly classified. There are no false positives or false negatives, demonstrating the model's strong performance and ability to handle the class imbalance. A scatter plot would likely show clear separation between the two classes, with distinct groups for churned and non-churned customers. The lack of overlap would visually confirm that the model accurately identified all churned customers and did not misclassify any non-churned ones, further supporting the model's effectiveness. The classification report confirms perfect results, with precision, recall, and F1-scores all at 1.00 for both classes. This indicates the model correctly predicted all true positives and avoided false positives and false negatives, maintaining a perfect balance between precision and recall. The support values show 1867 instances of class -1 and 1283 of class 1, indicating a slight imbalance. Despite this, the PC-SVM model classified all instances correctly, proving its ability to handle imbalanced datasets effectively. Both the macro and weighted averages are 1.00, showing consistent performance across both classes. This impressive performance highlights the PC-SVM's potential as a powerful tool for classifying imbalanced datasets, such as customer churn prediction.

Table 2 provides a concise overview of the performance findings obtained from experiments conducted on SVM with Imbalanced Dataset, SVM with Balanced Dataset, and PC-SVM with Imbalanced Dataset.

Table 2. The Summary of Experimental Performance

No	Dataset	SVM Imbalanced Dataset				SVM Balanced Dataset (SMOTE)				PC-SVM Imbalanced Dataset			
		Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
1.	Yeast	0.74	0.72	0.74	0.68	0.68	0.69	0.68	0.67	1.00	1.00	1.00	1.00
2.	Heart	0.83	0.83	0.83	0.82	0.69	0.69	0.69	0.69	1.00	1.00	1.00	1.00
3.	Churn	0.84	0.71	0.84	0.77	0.54	0.69	0.54	0.44	1.00	1.00	1.00	1.00
	Mean	0.80	0.75	0.80	0.76	0.64	0.69	0.64	0.60	1.00	1.00	1.00	1.00

Table **2** compares three machine learning models—SVM (with and without balancing), and PC-SVM—on three datasets (Yeast, Heart, and Churn). The models are evaluated based on accuracy, precision, recall, and F1 score to see how well they handle imbalanced data. The results show that PC-SVM performs the best. Here's a breakdown of each model's performance.

SVM on imbalanced data performs reasonably well, with accuracy between 0.74 and 0.84 across the datasets. On the Yeast dataset, SVM achieves an accuracy of 0.74, with precision and recall of 0.72 and 0.74, respectively. While these results are consistent, SVM struggles to balance precision and recall, with an F1 score of 0.68. On the Heart and Churn datasets, SVM performs better with F1 scores of 0.82 and 0.77, but it still has issues with recall, missing some relevant instances. Overall, SVM can handle imbalanced data decently, but its performance can be improved, especially for more complex datasets like Churn.

When the dataset is balanced using SMOTE, SVM's performance drops significantly, especially on the Churn dataset. Its accuracy falls to 0.54, and the F1 score drops to 0.44, with lower precision and recall. SMOTE, which is intended to improve learning from the minority class, seems to introduce noise and reduce model precision, particularly for the Churn dataset. This suggests that while balancing techniques can help, they may not always improve performance and can even make things worse, especially for complex models like SVM.

PC-SVM stands out as the best-performing model, achieving perfect scores (accuracy, precision, recall, and F1 score) across all datasets. It correctly classifies all instances with no false positives or false negatives. This performance shows that PC-SVM is particularly good at handling imbalanced datasets, likely due to its ability to better differentiate between classes, especially the minority class, something that SVM struggle with.

In summary, applying SMOTE to balance the dataset did not improve SVM's performance and even worsened the results on the Churn dataset. Conversely, PC-SVM demonstrated outstanding performance across all datasets, establishing itself as a reliable classifier for handling imbalanced data. These findings indicate that for datasets such as Yeast, Heart, and Churn, PC-SVM is the most effective model, whereas traditional SVM requires further optimization. Moreover, balancing techniques like SMOTE do not guarantee performance enhancement and can, in some cases, negatively impact the model.

5.0 CONCLUSION

5.1 Achievement Of The Research Objective

This study aimed to enhance the performance of Support Vector Machine (SVM) in addressing imbalanced datasets by integrating Posterior Probability and Correlation techniques, resulting in the development of the PC-SVM model. The research objectives were achieved through comprehensive evaluation of the proposed model across various imbalanced datasets. The findings indicate that the PC-SVM model yields notable improvements in classification metrics, including accuracy, precision, recall, and F1-score. In all evaluated cases, PC-SVM consistently outperformed conventional models by effectively reducing the bias toward majority classes. These results suggest that the combined application of posterior probability and correlation techniques substantially strengthens SVM's ability to manage imbalanced data distributions.

5.2 Contribution Of Research

In this study, we propose a novel classification approach named PC-SVM, designed to overcome the shortcomings of conventional SVMs when dealing with imbalanced datasets. The main contributions of this research are summarized as follows:

- Development of PC-SVM: The fusion of Posterior Probability and Correlation techniques into the SVM framework resulted in a novel algorithm that effectively addresses the imbalance issue in datasets, making it a practical solution for real-world problems.
- Improved Performance Metrics: The PC-SVM model consistently demonstrated superior performance compared to traditional SVM models on imbalanced datasets.
- Evaluation on Multiple Datasets: The proposed method was rigorously tested on several publicly available datasets, further validating its effectiveness and generalizability across different domains.
- Algorithm Adaptation: The integration of correlation coefficients into posterior probabilities for classification improves the understanding of relationships between features and their impact on class separation, a novel adaptation within the SVM framework.
- PC-SVM can be recommended for industries where detecting rare events is crucial. In healthcare, it can help identify rare diseases more accurately, reducing misdiagnoses. In cybersecurity, it can enhance intrusion detection, identifying rare but serious threats. Additionally, in manufacturing and aviation, PC-SVM can predict equipment failures, preventing costly breakdowns. These applications show how PC-SVM can be practically useful in real-world scenarios.

5.3 Limitation And Future Research

While the research yielded promising results, several limitations were identified that open up opportunities for further investigation:

- Dataset Variety: Although multiple datasets were used, the research was limited to publicly available data with specific imbalance characteristics.
- Computational Complexity: The fusion of posterior probability and correlation techniques increased the computational complexity of the PC-SVM model. Future work could focus on optimizing the algorithm for faster computation, especially for large-scale datasets.
- Parameter Tuning: The performance of the PC-SVM relies on hyperparameter tuning, which was done manually in this research. Future research could incorporate automated hyperparameter optimization techniques to further enhance model performance.

- Exploration of Other Techniques: While this research focused on posterior probability and correlation techniques, future studies could explore the integration of other statistical methods or ensemble learning strategies to further improve classification performance on imbalanced datasets.

In conclusion, this research presents an effective solution for improving SVM performance on imbalanced datasets through PC-SVM, setting a foundation for future research in this area.

AUTHOR CONTRIBUTION

Canggih Ajika Pamungkas – Developed the research concept, designed the methodology, conducted experiments, and wrote the initial draft of the manuscript.
Megat F. Zuhairi – Provided theoretical insights and reviewed the manuscript for clarity and accuracy.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

REFERENCES

[1] J. Alcaraz, M. Labbé, and M. Landete, "Support Vector Machine with feature selection: A multiobjective approach," *Expert Syst. Appl.*, vol. 204, no. May, p. 117485, 2022, doi: 10.1016/j.eswa.2022.117485.

[2] J. Liu, "Fuzzy support vector machine for imbalanced data with borderline noise," *Fuzzy Sets Syst.*, vol. 1, pp. 1–10, 2020, doi: 10.1016/j.fss.2020.07.018.

[3] C. Wu, N. Wang, and Y. Wang, "Increasing Minority Recall Support Vector Machine Model for Imbalanced Data Classification," *Discret. Dyn. Nat. Soc.*, vol. 2021, 2021, doi: 10.1155/2021/6647557.

[4] H. Liu, Z. Liu, W. Jia, D. Zhang, and J. Tan, "A Novel Imbalanced Data Classification Method Based on Weakly Supervised Learning for Fault Diagnosis," *IEEE Trans. Ind. Informatics*, vol. 18, no. 3, pp. 1583–1593, 2022, doi: 10.1109/TII.2021.3084132.

[5] S. Shaikh, S. M. Daudpota, A. S. Imran, and Z. Kastrati, "Towards improved classification accuracy on highly imbalanced text dataset using deep neural language models," *Appl. Sci.*, vol. 11, no. 2, pp. 1–20, 2021, doi: 10.3390/app11020869.

[6] R. A. Hamad, M. Kimura, and J. Lundström, "Efficacy of Imbalanced Data Handling Methods on Deep Learning for Smart Homes Environments," *SN Comput. Sci.*, vol. 1, no. 4, pp. 1–10, 2020, doi: 10.1007/s42979-020-00211-1.

[7] V. Rupapara, F. Rustam, H. F. Shahzad, A. Mehmood, I. Ashraf, and G. S. Choi, "Impact of SMOTE on Imbalanced Text Features for Toxic Comments Classification Using RVVC Model," *IEEE Access*, vol. 9, pp. 78621–78634, 2021, doi: 10.1109/ACCESS.2021.3083638.

[8] H. Qin, H. Zhou, and J. Cao, "Imbalanced learning algorithm based intelligent abnormal electricity consumption detection," *Neurocomputing*, vol. 402, no. xxxx, pp. 112–123, 2020, doi: 10.1016/j.neucom.2020.03.085.

[9] S. S. Mullick, S. Datta, S. G. Dhekane, and S. Das, "Appropriateness of performance indices for imbalanced data classification: An analysis," *Pattern Recognit.*, vol. 102, p. 107197, 2020, doi: 10.1016/j.patcog.2020.107197.

[10] H. Shamsudin, U. K. Yusof, A. Jayalakshmi, and M. N. Akmal Khalid, "Combining oversampling and undersampling techniques for imbalanced classification: A comparative study using credit card fraudulent transaction dataset," *IEEE Int. Conf. Control Autom. ICCA*, vol. 2020-Octob, pp. 803–808, 2020, doi: 10.1109/ICCA51439.2020.9264517.

[11] K. H. Kim and S. Y. Sohn, "Hybrid neural network with cost-sensitive support vector machine for class-imbalanced multimodal data," *Neural Networks*, vol. 130, pp. 176–184, 2020, doi: 10.1016/j.neunet.2020.06.026.

[12] X. Tao *et al.*, "Affinity and class probability-based fuzzy support vector machine for imbalanced data sets," *Neural Networks*, vol. 122, pp. 289–307, 2020, doi: 10.1016/j.neunet.2019.10.016.

[13] C. Jimenez-Castaño, A. Alvarez-Meza, and A. Orozco-Gutierrez, "Enhanced automatic twin support vector machine for imbalanced data classification," *Pattern Recognit.*, vol. 107, 2020, doi: 10.1016/j.patcog.2020.107442.

[14] R. Abo Zidan and G. Karraz, "Gaussian Pyramid for Nonlinear Support Vector Machine," *Appl. Comput. Intell. Soft Comput.*, vol. 2022, 2022, doi: 10.1155/2022/5255346.

[15] Y. S. Solanki *et al.*, "A hybrid supervised machine learning classifier system for breast cancer prognosis using feature selection and data imbalance handling approaches," *Electron.*, vol. 10, no. 6, pp. 1–16, 2021, doi: 10.3390/electronics10060699.

[16] R. Kumar R *et al.*, "Investigation of nano composite heat exchanger annular pipeline flow using CFD analysis for crude oil and water characteristics," *Case Stud. Therm. Eng.*, vol. 49, p. 104908, 2023, doi: 10.1016/j.csite.2023.103297.

[17] B. Richhariya and M. Tanveer, "A reduced universum twin support vector machine for class imbalance learning," *Pattern Recognit.*, vol. 102, p. 107150, 2020, doi: 10.1016/j.patcog.2019.107150.

[18] M. Li, A. Xiong, L. Wang, S. Deng, and J. Ye, "ACO Resampling: Enhancing the performance of oversampling methods for class imbalance classification," *Knowledge-Based Syst.*, vol. 196, no. xxxx, p. 105818, 2020, doi: 10.1016/j.knosys.2020.105818.

[19] A. S. Hussein, T. Li, C. W. Yohannese, and K. Bashir, "A-SMOTE: A new preprocessing approach for highly imbalanced datasets by improving SMOTE," *Int. J. Comput. Intell. Syst.*, vol. 12, no. 2, pp. 1412–1422, 2019, doi: 10.2991/ijcis.d.191114.002.

[20] P. Gnip, L. Vokorokos, and P. Drotár, "Selective oversampling approach for strongly imbalanced data," *PeerJ Comput. Sci.*, vol. 7, pp. 1–22, 2021, doi: 10.7717/PEERJ-CS.604.

[21] X. W. Liang, A. P. Jiang, T. Li, Y. Y. Xue, and G. T. Wang, "LR-SMOTE — An improved unbalanced data set oversampling based on K-means and SVM," *Knowledge-Based Syst.*, vol. 196, 2020, doi: 10.1016/j.knosys.2020.105845.

[22] A. S. Desuky and S. Hussain, "An Improved Hybrid Approach for Handling Class Imbalance Problem," *Arab. J. Sci. Eng.*, vol. 46, no. 4, pp. 3853–3864, 2021, doi: 10.1007/s13369-021-05347-7.

[23] K. Qi, H. Yang, Q. Hu, and D. Yang, "A new adaptive weighted imbalanced data classifier via improved support vector

machines with high-dimension nature," *Knowledge-Based Syst.*, vol. 185, p. 104933, 2019, doi: 10.1016/j.knosys.2019.104933.

[24] H. Shamsudin, U. K. Yusof, Y. Haijie, and I. S. Isa, "an Optimized Support Vector Machine With Genetic Algorithm for Imbalanced Data Classification," *J. Teknol.*, vol. 85, no. 4, pp. 67–74, 2023, doi: 10.11113/jurnalteknologi.v85.19695.

[25] Y. Park and J. S. Lee, "A Learning Objective Controllable Sphere-Based Method for Balanced and Imbalanced Data Classification," *IEEE Access*, vol. 9, pp. 158010–158026, 2021, doi: 10.1109/ACCESS.2021.3130272.

[26] H. Patel, D. Singh Rajput, G. Thippa Reddy, C. Iwendi, A. Kashif Bashir, and O. Jo, "A review on classification of imbalanced data for wireless sensor networks," *Int. J. Distrib. Sens. Networks*, vol. 16, no. 4, 2020, doi: 10.1177/1550147720916404.

[27] S. Strasser and M. Klettke, "Transparent Data Preprocessing for Machine Learning," *HILDA 2024 - Work. Human-In-the-Loop Data Anal. Co-located with SIGMOD 2024*, 2024, doi: 10.1145/3665939.3665960.

[28] J. Nalic and A. Svraka, "Importance of data pre-processing in credit scoring models based on data mining approaches," *2018 41st Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2018 - Proc.*, pp. 1046–1051, 2022, doi: 10.23919/MIPRO.2018.8400191.

[29] H. F. Tayeb, M. Karabatak, and C. Varol, "Time Series Database Preprocessing for Data Mining Using Python," *8th Int. Symp. Digit. Forensics Secur. ISDFS 2020*, pp. 20–23, 2020, doi: 10.1109/ISDFS49300.2020.9116260.

[30] S. Albahra *et al.*, "Artificial intelligence and machine learning overview in pathology & laboratory medicine: A general review of data preprocessing and basic supervised concepts," *Semin. Diagn. Pathol.*, vol. 40, no. 2, pp. 71–87, 2023, doi: 10.1053/j.semdp.2023.02.002.

[31] Z. Liu, "Research on data preprocessing method for artificial intelligence algorithm based on user online behavior," *J. Comput. Electron. Inf. Manag.*, vol. 12, no. 3, pp. 74–78, 2024, doi: 10.54097/qf6fv8j1.

[32] A. Q. Md, S. Kulkarni, C. J. Joshua, T. Vaichole, S. Mohan, and C. Iwendi, "Enhanced Preprocessing Approach Using Ensemble Machine Learning Algorithms for Detecting Liver Disease," *Biomedicines*, vol. 11, no. 2, 2023, doi: 10.3390/biomedicines11020581.

[33] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Glob. Transitions Proc.*, vol. 3, no. 1, pp. 91–99, 2022, doi: 10.1016/j.gltp.2022.04.020.

[34] H. T. Duong and T. A. Nguyen-Thi, "A review: preprocessing techniques and data augmentation for sentiment analysis," *Comput. Soc. Networks*, vol. 8, no. 1, pp. 1–16, 2021, doi: 10.1186/s40649-020-00080-x.

[35] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data," *Front. Energy Res.*, vol. 9, no. March, pp. 1–17, 2021, doi: 10.3389/fenrg.2021.652801.

[36] V. Chernykh, A. Stepnov, and B. O. Lukyanova, "Data preprocessing for machine learning in seismology," *CEUR Workshop Proc.*, vol. 2930, pp. 119–123, 2021.

[37] A. J. Mohammed, "Improving Classification Performance for a Novel Imbalanced Medical Dataset using SMOTE Method," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 3, pp. 3161–3172, 2020, doi: 10.30534/ijatcse/2020/104932020.

[38] A. Kulkarni, D. Chong, and F. A. Batarseh, *Foundations of data imbalance and solutions for a data democracy*. Elsevier Inc., 2020. doi: 10.1016/B978-0-12-818366-3.00005-8.

[39] D. Makowski, M. Ben-Shachar, I. Patil, and D. Lüdecke, "Methods and Algorithms for Correlation Analysis in R," *J. Open Source Softw.*, vol. 5, no. 51, p. 2306, 2020, doi: 10.21105/joss.02306.

[40] M. S. Vural and M. Telceken, "Modification of posterior probability variable with frequency factor according to Bayes Theorem," *J. Intell. Syst. with Appl.*, vol. 5, no. 1, pp. 19–26, 2022, doi: 10.54856/jiswa.202205195.