Received: 5 July 2025 | Revised: 29 October 2025 | Accepted: 8 November 2025 | Published online: 3 February 2026

RESEARCH ARTICLE

Journal of Computational and Cognitive Engineering
2026, Vol. 00(00) 1–13
DOI: 10.47852/bonviewJCCE62026710

BON VIEW PUBLISHING

Performance Optimization of Naive Bayes Classification Using Filter-Based Feature Selection

Nalini Manogaran[1,2], Kalpana Vadivelu[3], Siva Subramanian Raju[4], Yamini Bhavani Shankar[5], Balamurugan Balusamy[6] and Sumendra Yogarayan[1,*]

[1]*Faculty of Information Science and Technology, Multimedia University, Malaysia*

[2]*Department of Computer Science and Engineering (Cyber Security), S.A. Engineering College (Autonomous), India*

[3]*Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, India*

[4]*Department of Computer Science and Engineering, SRM Institute of Science and Technology-Tiruchirappalli, India*

[5]*Department of Networking and Communications, SRM Institute of Science and Technology, India*

[6]*School of Engineering and IT, Manipal Academy of Higher Education, United Arab Emirates*

Abstract: In machine learning (ML), feature selection (FS) is considered an important preprocessing step that helps to find and pick the most relevant attributes from a dataset. FS minimizes computational complexity, enhances model efficiency, and improves generalization performance by removing unnecessary features. With an important emphasis on the use of filter FS techniques in the research, this study examines the importance and effects of FS in the context of ML. FS is important in ML because it maximizes the interpretability and performance of the ML models. This study explores how well the performance of the naive Bayes (NB) classifier is influenced by filter FS methods such as Symmetrical Uncertainty, Information Gain, Gain Ratio, Chi-Square (CHISQUARE), and RELIEFF. These filter FS techniques are selected because they make ranking of features according to certain criteria and are also computationally efficient. The results captured are projected using the validity scores like accuracy, specificity, precision, false negative rate, and false positive rate using the selected features. Further, using with FS and without FS, the classifier's performance is analyzed using these validity scores. The results represent the RELIEFF approach with NBs getting the best superior results with regards to accuracy, specificity, and precision when compared to other filter FS and NB without FS strategies. Finally, the study reveals the importance of filter FS techniques in maximizing the performance and efficacy of ML models. It also provides useful information on the subtle effects of FS strategies on classification performance, offering practitioners and academics useful knowledge that will further help them optimize their models for practical uses.

Keywords: machine learning filter, feature selection (FS), computational complexity, model efficiency, generalization performance

1. Introduction

Feature selection (FS) is an important phrase in machine learning (ML), which involves choosing the most relevant features from the given dataset [1]. The purpose of the FS is to improve the overall predictive performance, decrease computing complexity, and increase model efficiency [2]. This FS makes the ML models get the underlying patterns in the data and helps to create more accurate and understandable models, with the help of a feature subset obtained using the FS.

1.1. Applications of FS

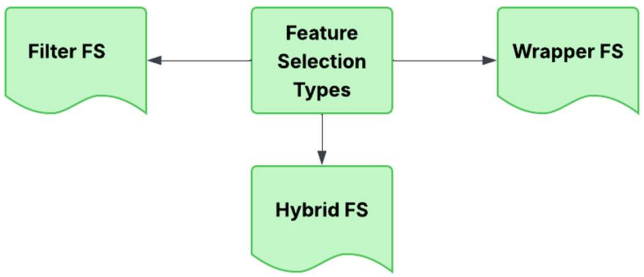
FS is an important phrase of any preprocessing stage in the ML approach. The main goal is to choose the most significant and informative feature subset for model training. The curse of dimensionality is reduced by eliminating unused features, and this helps keep models from being too complicated and prone to overfitting. Finally, FS helps to improve ML models' interpretability and makes it easier to comprehend the variables affecting predictions.

1.2. Machine learning integration

The use of FS with the ML has a great influence on the ML performance. The feature subset chosen from the FS process is further modeled with ML to improve the prediction and also to generalize to

***Corresponding author:** Sumendra Yogarayan, Faculty of Information Science and Technology, Multimedia University, Malaysia. Email: sumendra@mmu.edu.my

Figure 1
Feature selection types



new data. Further, the optimal FS approaches try to keep the model focused on the most relevant elements of the data and make robust, reliable models for a range of applications.

1.3. FS types

There are three kinds of FS techniques: filter, wrapper, and embedded approaches [3, 4]. Types of FS are given in Figure 1.

1. *Filter*

Definition : Filter techniques are straightforward, quick to compute, and solely dependent on the variable properties. The filter ranks the variables according to how relevant they are to the output label using many statistical metrics. Variables that have a strong correlation with the output label are chosen by the filter, followed by those that have a weaker correlation with other input predictors.
Benefits: Filter FS is faster and computationally efficient.
Drawbacks: Filter FS misses feature dependencies that are essential to certain models.

2. *Wrapper*

Definition : Wrapper is an efficient approach for selecting the best variable subset for use with an ML model. Compared to the filter approach, wrapper is a bit slow, but the variable subset chosen is an efficient one. Here, in the wrapper approach, the ML model is applied to select the variable subset [5].
Advantages: Feature dependencies are taken into account; these approaches provide more accurate findings.
Disadvantages: Wrapper causes overfitting on the training set due to its high computational demands.

3. *Embedded FS*

Definition : The process of training the model includes FS through the use of embedded approaches. During the training phase, the selection of features is made by the model's acquired knowledge.
Benefits: By selecting features adaptively during training, these techniques may enhance model performance.
Disadvantages: The underlying model's constraints may restrict the FS procedure.

1.4. Selection of filter features

The computational simplicity and effectiveness of filter FS led to its selection in this study. Because filter techniques assess characteristics individually, they are especially well-suited for datasets with large dimensions. They provide a concise synopsis of feature importance without the computational burden of wrapper techniques.

1.5. Selection of filter features in this study

In this study, a number of specialized filter FS techniques were used, including RELIEFF, Chi-Square, IG, GR, and SU. These techniques were chosen because they could evaluate the significance of features according to various information-theoretic and statistical standards. Every approach provides a distinct viewpoint on the significance of features, enabling a thorough assessment of their influence on the naive Bayes (NB) classifier's functionality.

1.6. Dataset

A customer dataset obtained from the UCI repository is used in the present study [6]. This dataset was selected because it has real-world importance and shows complex trends that impact customer behavior. This dataset provides a helpful foundation to determine how well different FS methods function in a context that is applicable to real-world scenarios.

1.7. Validity scores

Different validity scores such as accuracy, specificity, precision, false negative rate (FNR), and false positive rate (FPR) are applied in the study. The experimental results captured from NB with FS and NB without FS are projected using these validity scores.

1.8. Important of the research

The purpose of the work is to gain an insight into the application of filter FS that contributes to the enhancement of the ML model performance and reduces the computational cost of the ML model. These useful insights will help people understand the importance of the FS in ML. Moreover, compared to different FS, why the filter is chosen and experimented with is explored in the study.

1.9. Objective

The purpose of the study is to show how the FS can improve the ML model efficiency and minimize the complexity of the model. The study's particular aims and objectives are as follows:

1) Analyzing FS Methodologies: To thoroughly evaluate and compare the efficacy of several filter FS methods, including RELIEFF, Chi-Square, IG, GR, and SU.
2) Assessing the Performance of Classifiers: To determine how the selected FS techniques affect the NB classifier's performance using metrics like FNR, FPR, specificity, accuracy, and precision.
3) Practical Importance: To make use of a real-world relevant consumer dataset from the UCI repository, capturing complex patterns that accurately represent real-world customer behavior. This leads to the results obtained being reliable and useful in real-world situations.
4) FS Optimization: To provide researchers and practitioners with practical guidance and insights to improve FS in ML applications. To improve the performance and understanding of the NB classifier, which involves determining the optimal filter FS method.
5) Improving Knowledge: To increase knowledge about the general impact of filter FS on classification challenges. The study's objectives are to add to the volume of existing knowledge on the subject and provide valuable insights that will direct further research and practical applications.

6) While the use of filter-based FS techniques with NB is not new, this study's novelty stems from its comprehensive multi-metric performance evaluation and critical trade-off analysis, rather than a sole focus on accuracy. We also provide practical insights into selecting FS methods when metrics present conflicting trends, which is rarely emphasized in prior works.

By addressing these objectives, the research aims to improve our understanding of how filter FS impacts the performance of the NB classifier, particularly in the context of using customer datasets. The findings should be applicable in the real world, where ML model optimization depends on having efficient and intelligible models.

1.10. Motivation

The need to improve ML models for real-world uses, particularly consumer data classification jobs, motivated this research. Several main reasons drive the investigation of filter FS techniques and how they affect the NB classifier:

1. Improving model efficiency:

High-dimensional datasets are a typical cause of difficulty for ML models, especially in practical applications. By reducing the number of features, FS helps you maximize model efficiency while lowering computational load and enhancing overall performance.

2. Interpretable decision-making models:

In situations when model-based decision-making requires a full understanding of the relevant aspects, interpretability is crucial. By highlighting the most crucial information, FS aims to improve interpretability and empower stakeholders to make better-informed decisions.

3. Addressing the dimensionality curse:

Higher feature dimensions can give rise to model overfitting and reduced generalization, an issue known as the "curse of dimensionality" in ML. By determining and maintaining just the most important attributes, FS techniques provide a solution to the problems related to high-dimensional data.

4. Model performance optimization for business applications:

In business and industrial applications, it is essential to deploy ML models with optimal performance. The research evaluates several filter FS procedures in an effort to provide practitioners with insights that might help them pick the best approach for increasing model accuracy and efficiency.

5. FS method options:

For practitioners, the wide variety of FS approaches might be bewildering. This study intends to help the selection of filter feature approaches by offering practical suggestions based on their influence on the performance of the NB classifier.

The practical consequences of optimizing ML models for real-world applications are the driving force behind this study. The emphasis on customer datasets and filter FS techniques intends to bridge the theoretical and practical gaps, supporting the development of more effective and interpretable ML solutions in customer-centric domains.

2. Related Study

The literature review offers a thorough summary of several works on FS in ML from a variety of fields. A recurring issue is how crucial FS is to improving classification model performance, cutting down on computing complexity, and meeting particular needs in various application domains.

FS is an important feature of ML, especially in domains such as bioinformatics. Filter techniques are important for FS since they may cut run time and forecast accuracy greatly. This research compared the effectiveness of 22 filter techniques with classification methods on 16 high-dimensional classification datasets. It determined that no one filter approach consistently outperforms all others, but it gave suggestions for those that perform well on a variety of datasets. For the investigation, the R ML package mlr was utilized since it provides a consistent programming API for FS utilizing filter techniques [7].

In high-dimensional datasets with limited samples and features, gene selection is critical. X variance and mutual congestion are two heuristic methods for gene selection that are proposed in this article. X variance employs internal feature attributes to classify labels, whereas mutual congestion operates on the basis of frequency. The outcomes of evaluations on eight binary medical datasets indicate that X variance performs admirably with standard datasets, whereas mutual congestion increases accuracy in high-dimensional datasets by a substantial margin [8].

A wrapper-filter combination of ant colony optimization for FS is proposed in this paper. The computational complexity is diminished through the implementation of a filter method for subset evaluation. In addition, a memory is employed to retain the most optimal ants and to implement a feature dimension-dependent pheromone update for multi-objective FS. On real-world datasets, the procedure has been shown to outperform the majority of contemporary algorithms for FS. Furthermore, the model's resilience was evaluated using microarray and facial emotion recognition datasets [9].

To avoid overfitting, FS is an important step in ML. It aids in the reduction of dimensionality, memory savings, and computation time. Spearman Correlation, Fisher Score, Pearson Correlation, Count-based, Kendall Correlation, Chi-squared, and Mutual Information are among the seven filter-based FS approaches available in Azure Machine Learning studio. A study on the Automobile Price Prediction dataset using Bayesian linear regression and the elbow technique, with Fisher Score emerging as the most efficient [10].

The issue for higher education is to improve academic performance prediction models in order to detect substandard pupils. Although data mining provides several strategies for prediction, building good models is difficult. Model accuracy is driven by classifiers and FS, with each offering the best outcome provided it meets the right categorized data. Some research has produced great predictions of students' academic achievement, although it relies on classification methods rather than FS. Based on categorized data, this study provides filter-based FS techniques and classifiers, assisting researchers in finding the appropriate combination of filter-based FS methods and classifiers. Tests demonstrate that using suitable classifiers for certain categorized data and selecting acceptable features improves prediction model accuracy [11].

The study's goal is to find the most efficient filtering technique for extracting features from Head CT Scan pictures. Due to comparable outcomes, visually recognizing the sorts of scanned medical photos is difficult. Filtering techniques such as the median, bandpass filter, XYZ color transformer filter, improved local contrast filter, and histogram equalization were used. The most important filtered pictures were segmented using graph cut segmentation, and statistical feature extraction was used to obtain the features. The most important filtering approaches were discovered to be histogram equalization and improved local contrast filtering [12].

The significance of mental and physical well-being is growing as the contemporary economy improves. Depression and anxiety are two factors that lead to mental health issues. Previous models analyzed brain activity and mental state using medical-based approaches such as electroencephalograms and electrocardiograms. These procedures, however, are complicated, time-consuming, and costly. The K-Nearest Neighbors (KNN) ML model is presented with better FS, a nonparametric approach for classifying distinct input points. Following preprocessing, filtering methods such as forward and reverse filtering are used. KNN is the greatest classifier since it predicts outcomes with 99% accuracy [13].

Plant disease diagnosis is still a major concern in the scientific community. In agriculture, artificial intelligence (AI) is critical, and image analysis and classification algorithms are used to anticipate plant illnesses. Current approaches, however, can only identify plant illnesses. An enhanced Convolutional Neural Network (CNN) approach has been developed to overcome this. To identify medicinal plant properties, CNN preprocessing includes a wrapping filter followed by LDR FS. Changes in plant foliar disease states are used to build automated disease detection tools. CNNs are very accurate in picture classification and recognition, enabling rapid and reliable image identification [14].

The NB method is commonly employed in classification issues; however, real-time data often invalidates its conditional assumptions, resulting in less ideal prediction accuracy. Structure Learning, FS, Data Expansion, and Attribute Weighting are some of the approaches presented by researchers to reduce the NB assumption. FS has gained popularity in recent years because it allows for the extraction of significant variables from a dataset without reducing the amount of information known about the dataset. To ease the conditional assumption, this method employs FS in combination with NB. The suggested study applies the filter FS methodology, which is faster and more successful than other FS techniques. The characteristics are modeled using the NB algorithm and two separate filter FS techniques. Various validity ratings are used to analyze, compare, and predict the findings. The study finds that Filter NB outperforms NB and has a higher effective time complexity [15].

Identification of wheat leaf diseases is essential to agriculture and wheat quality. This work presents an integrated ML strategy that combines RGB, Linear, and Quadratic discriminant analysis with Colour Layout filters with Linear discriminant analysis, Quadratic discriminant analysis, and Linear discriminant analysis to improve wheat leaf disease identification. With the aid of images, processing, feature extraction, selection, and learning, farmers can identify plant diseases with greater speed and accuracy thanks to the agricultural autonomous leaf infection detection system. Wheat illnesses are effectively classified using the Linear Discriminant Analysis Colour Layout Filter; CLF's 0.88 LDA F-Measure is the most accurate model. Additionally, the research discovered that Colour Layout Filters based on Linear Discriminant Analysis performed better than other models [16].

Problem addressed

This study addresses the need for appropriate FS in ML to improve model performance and reduce computational complexity and also explores the particular challenges in a variety of fields, including bioinformatics, agriculture, mental health, and education. The studies highlight how important it is to use suitable FS strategies based on the features of the application domain and the dataset.

3. Methodology

The proposed methodology explores evaluating filter FS techniques for the NB classifier using a customer dataset that was obtained from the UCI repository. The need to improve ML models for real-world applications, particularly in customer-centric domains, drives approach selection. The approach used is strongly influenced by the importance of FS and the desire to tackle this problem.

3.1. FS's significance

A key phase in ML is FS. This FS involves choosing a subset of relevant features from a dataset in order to improve the model's interpretability, effectiveness, and generalizability [17]. In high-dimensional datasets, redundant or unnecessary features may lead to overfitting, increase computing costs, and degrade model performance [18]. As a result, selecting features wisely is essential to raising ML models' overall effectiveness.

3.2. Addressing the problem

The main purpose of the research is to show how the FS helps to improve the ML model performance. FS works by selecting the relevant variables from the correlated variables and further helps to enhance the ML performance. In real time, most real data collected consists of highly correlated variables and is high dimensional. To overcome the problem with these datasets and to get an accurate prediction from the ML model, the FS method is applied. By eliminating irrelevant and redundant attributes, feature selection (FS) reduces dimensionality, improves generalization, and enhances the overall predictive performance of machine learning models.

3.3. Methodology

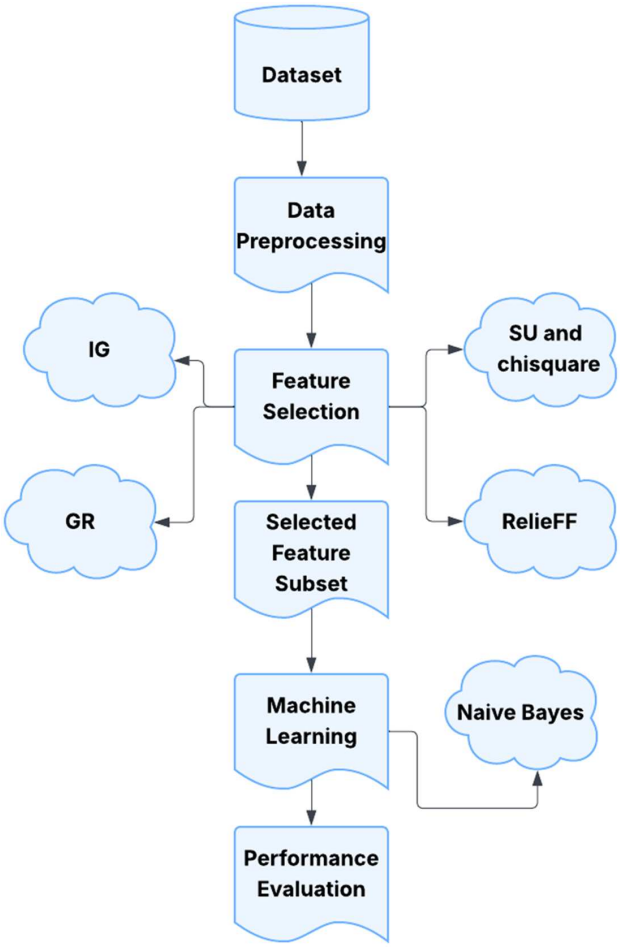
Here, the proposed methodology filter FS methods are considered and experimented with and given in Figure 2. These methods are fast and not dependent upon any ML models to select the variables. The filter FS methods applied are SU, IG, GR, Chi-Square, and RELIEFF. Using these methods, the customer dataset acquired from the UCI repository is experimented [6]. The selected variables subset is modeled using the NB model. The results captured from the experimental procedure are represented using different validity measures such as accuracy, specificity, precision, FNR, and FPR. All experiments used 10-fold cross-validation to ensure robustness of the results. The dataset was randomly partitioned into ten equal folds; in each iteration, nine folds were used for training and one for testing, and the process was repeated until each fold served as a test set. Average scores and standard deviations were reported for each metric.

In conclusion, the use of FS aids to improve the NB model by using the relevant dataset from the correlated dataset. The study attempts to aid in the creation of more effective, comprehensible, and precise ML models in fields where comprehending consumer behavior is crucial by tackling the FS challenge.

4. Feature Selection

FS is also referred to as variable selection or attribute selection in ML. FS is the procedure of choosing the relevant variables from the given dataset by using any evaluation metrics or with the ML process. The purpose of the FS is to minimize the computational complexity of the ML model and also to improve the prediction

Figure 2
Overall methodology



accuracy of the model [19]. The variables that are highly correlated with the class label are highly ranked, and the variables that are less correlated with the class label are ranked low. For example, consider the D= {y_1,.....y_n |C_k}, where 〖(y〗 _1,. . .,y_n) represent the input variables and C_k represent the output. The purpose of FS is to apply any FS method to choose relevant variables from the D and improve the ML model.

4.1. Feature selection types

There are three primary categories of FS techniques:

4.1.1. Filter

Explanation: Filter techniques are straightforward, quick to compute, and solely dependent on the variable properties. The filter ranks the variables according to how relevant they are to the output label using many statistical metrics. Variables that have a strong correlation with the output label are chosen by the filter, followed by those that have a weaker correlation with other input predictors [20].
Benefits: Fast overview of feature importance; computationally efficient.
Drawbacks: May ignore feature dependencies that are essential to certain models.

4.1.2. Wrapper methods

Explanation: Wrapper is an efficient approach for selecting the best variable subset for use with an ML model. Compared to the filter approach, wrapper is bit a slow, but the variable subset choosen is an efficient one. Here, in the wrapper approach, the ML model is applied to select the variable subset [21].
Advantages: Feature dependencies are taken into account; these approaches provide more accurate findings.
Disadvantages: Wrapper causes overfitting on the training set due to its high computational demands.

4.1.3. Embedded methods

Explanation: FS is incorporated into the classification algorithm in an embedded manner. In order to achieve the greatest classification accuracy, the classifier modifies its internal parameters throughout the training phase and chooses the proper weights and priorities for each feature. As a result, with an embedded technique, finding the ideal feature subset and building the model are done in one step.

Benefits: Reduce overfitting, improve efficiency, and handle multicollinearity.
Disadvantages: Model dependency, high computational.
Difficulty in feature interaction understanding

Overall, these feature selection approaches are widely used, and the choice of method depends on the dataset characteristics and the specific machine learning task. Normally, the filter approach is fast and efficient and also computationally good in analyzing high-dimensional datasets. But the wrapper approach is good at choosing the feature subset, but computationally inefficient. In the embedded method, FS is incorporated into the classification algorithm in an embedded manner. The optimal strategy depends on a number of factors, including the size, dimensionality, and complexity of the ML model being employed.

4.2. Filter Feature Selection

Filter techniques are straightforward, quick to compute, and solely dependent on the variable properties. The filter ranks the variables according to how relevant they are to the output label using many statistical metrics. Variables that have a strong correlation with the output label are chosen by the filter, followed by those that have a weaker correlation with other input predictors.

Filter selection algorithm

filter_feature_selection (data, target_variable, num_features_to_select):
data: Input dataset D= {y_1,.....y_n |Z_k}
class_variable: Z_k target variable
num_features_to_select: Number of top features to choose
1. Define the dataset:
Let D be the dataset with features F and target variable T.
2. Calculate feature scores:
Let S(f) be the score for a feature f.
For each feature f in F:
S(f) = calculate_feature_score(D[f],D[T])
3. Select top features:
Let K be the number of top features to select.
Let S_sorted be the sorted list of features based on their scores.
Select the top K features: SelectedFeatures = S_sorted [:K]
Filter Feature Selection Types Employed in the Study:

A. *Symmetrical Uncertainty (SU)*

An information-theoretic measure termed symmetric uncertainty (SU) is used in filter FS to evaluate the mutual information between two variables, typically a feature and the target variable. It is determined by normalizing the mutual information using the variable entropy. Attributes that have a major influence on a model's prediction ability may be found via SU. Based on the features' ability to gather information and lessen ambiguity in classification tasks, the study uses SU as a criterion to assess the significance of features [22].

Algorithm:

function symmetrical_uncertainty_feature_selection(data, target_variable, num_features_to_select):
data: Input dataset D= {y_1,.....y_n |Z_k}
class_variable: Z_k target variable
num_features_to_select: Number of top features to choose
// Step 1: Calculate the entropy of the target variable
entropy_target = calculate_entropy(data[target_variable])
// Step 2: Calculate Symmetrical Uncertainty for each feature
symmetrical_uncertainties = calculate_symmetrical_uncertainties(data,target_variable,entropy_target)
// Step 3: Select the top features based on Symmetrical Uncertainty
selected_features = select_top_features(symmetrical_uncertainties,num_features_to_select)
// Step 4: Return the selected features
return selected_features

B. *Information Gain (IG)*

Information Gain (IG) is a filter FS approach that quantifies the amount of uncertainty removed from the target variable when a certain feature is known. It is a useful technique for sorting features based on how effective they are in discriminating, as it measures the amount of information a feature offers for the categorization. The NB classifier's overall effectiveness and accuracy are increased when IG is used in research to help select and prioritize the qualities that provide the most significant insights into case classification [23].

Algorithm:

function information_gain_feature_selection(data, target_variable, num_features_to_select):
data: Input dataset D= {y_1,.....y_n |Z_k}
class_variable: Z_k target variable
num_features_to_select: Number of top features to choose
// Step 1: Calculate the entropy of the target variable
entropy_target = calculate_entropy(data[target_variable])
// Step 2: Calculate Information Gain for each feature information_gains = calculate_information_gains(data,target_variable,entropy_target)
// Step 3: Select the top features based on Information Gain
selected_features = select_top_features(information_gains, num_features_to_select)
// Step 4: Return the selected features
return selected_features

C. *GR (Gain Ratio)*

By adding a normalization factor that takes into consideration a feature's inherent information and corrects for any biases, the Gain Ratio (GR) expands upon the idea of IG. When selecting filter features, this approach performs especially effectively for qualities with a wide range of values. By preventing selection bias toward variables with a large number of categories, GR contributes to the creation of a more balanced evaluation of study features. In the end, this strengthens the NB classifier's robustness [24].

Algorithm:

function gain_ratio_feature_selection(data, target_variable, num_features_to_select):
data: Input dataset D= {y_1,.....y_n |Z_k}
class_variable: Z_k target variable
num_features_to_select: Number of top features to choose
// Step 1: Calculate the entropy of the target variable
entropy_target = calculate_entropy(data[target_variable])
// Step 2: Calculate Gain Ratio for each feature
gain_ratios = calculate_gain_ratios(data,target_variable,entropy_target)
// Step 3: Select the top features based on Gain Ratio
selected_features = select_top_features(gain_ratios,num_features_to_select)
// Step 4: Return the selected features
return selected_features

D. *CHISQUARE*

A statistical technique called Chi-Square (CHISQUARE) is used to evaluate the degree of independence between two categorical variables. It is used to assess the relationship between each feature and the target variable in filter FS. CHISQUARE is an excellent tool for evaluating if characteristics have statistically significant correlations with the categorization result when dealing with categorical data. To make sure that the chosen features have a meaningful impact on the NB classifier's prediction ability, CHISQUARE is required in the investigation [25].

Algorithm:

function chi_square_feature_selection(data, target_variable, num_features_to_select):
data: Input dataset D= {y_1,.....y_n |Z_k}
class_variable: Z_k target variable
num_features_to_select: Number of top features to choose
// Step 1: Calculate observed and expected frequencies for each feature
observed_frequencies = calculate_observed_frequencies(data, target_variable)
expected_frequencies = calculate_expected_frequencies(data, target_variable)
// Step 2: Calculate Chi-Square statistic for each feature
chi_square_statistics = calculate_chi_square_statistics(observed_frequencies,expected_frequencies)
// Step 3: Select the top features based on Chi-Square statistics
selected_features = select_top_features(chi_square_statistics, num_features_to_select)
// Step 4: Return the selected features
return selected_features

E. *RELIEFF*

RELIEFF is a distance-based filter FS technique that evaluates a feature's relevance by taking into account an instance's closest neighbors [26]. It is able to capture feature dependencies and interactions by comparing each instance's feature values to those of its neighbors. Because RELIEFF has a reputation for finding characteristics that contribute to the local structure of the data, it is a popular option for datasets containing complex relationships. By offering insights into feature dynamics that other methods could miss, RELIEFF improves the research's FS process and raises the NB classifier's overall accuracy and adaptability.

These filter FS strategies are used in the context of the research on the customer dataset in order to identify the most informative features for the NB classifier. Each method offers a unique perspective on the importance of characteristics, contributing to a comprehensive analysis of how each influences classification success. The diversity of these methods allows for in-depth examination of several aspects of feature importance, enhancing the interpretability and robustness of the FS process.

Algorithm:

function reliefF_feature_selection(data, target_variable, num_features_to_select, k_neighbors):
data: Input dataset D= {y_1,.....y_n |Z_k}
class_variable: Z_k target variable
num_features_to_select: Number of top features to choose
// Step 1: Initialize feature scores
feature_scores = initialize_feature_scores(data.features)
// Step 2: Iterate through each instance in the dataset
for each instance in data:
// Step 3: Find k nearest neighbors (ignoring the current instance)
nearest_neighbors = find_k_nearest_neighbors(data,instance, target_variable,k_neighbors)
// Step 4: Update feature scores based on differences in feature values update_feature_scores(instance,nearest_neighbors, feature_scores,target_variable)
// Step 5: Select the top features based on scores
selected_features = select_top_features(feature_scores,num_features_to_select)
// Step 6: Return the selected features
return selected_features

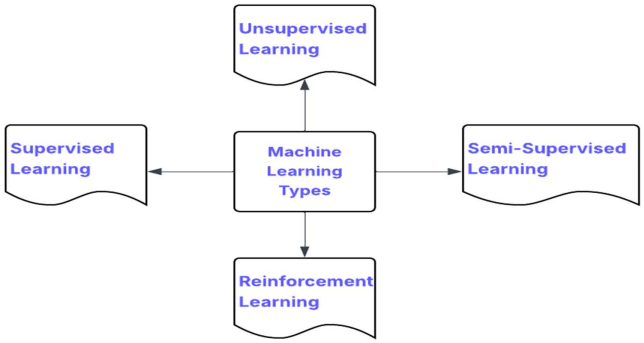
5. The Use of Machine Learning (ML)

Within the discipline of AI, ML is the study of creating models and algorithms that allow computer systems to learn from data and make judgments or predictions without explicit programming [27, 28]. The fundamental notion is that robots should be able to recognize links, patterns, and insights in data so they can generalize and adjust to previously unheard-of circumstances. ML is a broad field that includes many different methods and strategies. It is essential for addressing difficult issues and reaching wise choices in a variety of fields [29, 30]. The types of ML are shown in Figure 3.

5.1. Machine learning types

1. *Supervised learning*

Figure 3
Machine learning types



One of the types of ML is a supervised learning approach. In this approach, the ML models are trained using a labeled dataset, which means the dataset modeled consists of input and its associated output. Regression and classification are performed in supervised learning. Examples of supervised learning are NB, KNN, Support Vector Machine (SVM), and logistic regression (LR).

2. *Unsupervised learning*

One of the types of ML is an unsupervised learning approach. In this approach, the ML models are trained using an unlabeled dataset, which means the dataset modeled consists of input and no associated output. Clustering and association are performed in unsupervised learning. An example of supervised learning is KNN.

3. *Semi-supervised learning*

It includes both supervised and unsupervised learning approaches in semi-supervised learning. Because the algorithm is trained on a dataset that includes both labeled and unlabeled instances, it can make use of the labeled data that is readily accessible and use the unlabeled data to gain new insights. It is practical in situations where getting labeled data is costly or time-consuming.

4. *Reinforcement learning*

A particular type of ML called reinforcement learning trains an agent to make decisions by interacting with its surroundings. As it acts, the agent gets input in the form of rewards or penalties, and its goal is to eventually learn a policy that maximizes the cumulative reward. Examples: Game playing, robotics, and autonomous systems.

5.2. Naive Bayes

In the study, an ML approach called the NB classifier is used. NB is a simple method for classification tasks because of its ease of use, efficacy, and efficiency, particularly when dealing with moderate- to high-dimensional datasets.

Algorithm:

Step 1: Training Naive Bayes
function train_naive_bayes(training_data, class_labels):
class_probabilities = calculate_class_probabilities(class_labels)
feature_probabilities = calculate_feature_probabilities(training_data, class_labels)
return {'class_probabilities': class_probabilities, 'feature_probabilities': feature_probabilities}
Step 2: Calculate Class Probabilities:
function calculate_class_probabilities(class_labels):
total_samples = length(class_labels)
class_probabilities = {}
for class_label in unique_values(class_labels):
class_count = count_occurrences(class_labels, class_label)
class_probabilities[class_label] = class_count / total_samples
return class_probabilities
In this step, the probability of each class is measured. It is done by

$$P(C_i)\frac{Count\ of\ samples\ with\ class\ C_i}{Total\ number\ of\ samples}$$

Step 3: Calculate Feature Probabilities:
function calculate_feature_probabilities(training_data, class_labels):
feature_probabilities = {}

for class_label in unique_values(class_labels):
class_data = get_data_for_class(training_data, class_labels, class_label)
feature_probabilities[class_label] = {}
for feature_index in range(num_features(training_data)):
feature_values = get_values_for_feature(class_data, feature_index)
unique_feature_values = unique_values(feature_values)
for value in unique_feature_values:
probability = calculate_feature_probability(feature_values, value)
feature_probabilities[class_label][(feature_index, value)] = probability
return feature_probabilities

In this step, compute the probability of a particular feature value given the class

$$P(X_j) = x|C_i = \frac{Count\ of\ samples\ with\ X_j = x\ and\ class\ C_i}{Count\ of\ samples\ with\ C_i}$$

Step 4: Prediction with Naive Bayes

function predict_naive_bayes(model, new_instance):
class_probabilities = {}
for class_label in model['class_probabilities'].keys():
class_probability = model['class_probabilities'][class_label]
feature_probabilities = calculate_instance_feature_probabilities(model, new_instance, class_label)
combined_probability = class_probability * product_of_feature_probabilities(feature_probabilities)
class_probabilities[class_label] = combined_probability
predicted_class = argmax(class_probabilities)
return predicted_class

$$P(C_i|new\ instance) \propto P(C_i)\ X \prod_{j=1}^{n} P(X_j = x_j|\ C_i)$$

Predict the class

$$Predicted\ class\ =\ argmax_{C_i} P(C_i) new\ instance$$

In conclusion, NB is used in the study because it is straightforward, effective, and compatible with the dataset's properties, particularly when filter FS techniques are used. In the context of the study aims, the method is a practical option for classification problems because of its interpretability, resistance to the curse of dimensionality, and capacity to handle categorical data.

6. Experimental Results

Figure 4 shows the outcomes of several FS strategies, Filter NB and NB without a filter approach, using accuracy as the evaluation criterion. SU, IG, GR, CHISQUARE, and RELIEFF are among the FS techniques shown in Figure 4. The percentages of accuracy for each approach are presented. Notably, the RELIEFF approach outperformed the other strategies with the greatest accuracy of 89.5755%. This suggests that RELIEFF, a relief-based FS technique, was especially helpful in improving the performance of the NB classifier in this setting. In comparison, the classic NB strategy generated an accuracy of 88.0073%, suggesting that the integration of FS approaches enhanced classification accuracy in general. Compared to five filter FS, SU and GR get similar accuracy of NB (88.4475%), whereas IG and CHISQUARE get lower accuracy of 88.4187%. From the findings, it is clear that the RELIEFF and NB get a better feature subset and improve the NB model performance.

Figure 5 shows the outcomes of several FS strategies, Filter NB and NB without a filter approach, using Specificity as the evaluation criterion. SU, IG, GR, CHISQUARE, and RELIEFF are among the FS techniques shown in Figure 5. The percentages of Specificity for each approach are presented. Notably, the RELIEFF approach outperformed the other strategies with the greatest accuracy of 95.3%. This suggests that RELIEFF, a relief-based FS technique, was especially helpful in improving the performance of the NB classifier in this setting. In comparison, the classic NB strategy generated an accuracy of 92.6%, suggesting that the integration of FS approaches enhanced classification accuracy in general. In comparison, the classic NB approach had a specificity of 92.6%, while SU and GR had a specificity of 93.8%. IG and CHISQUARE did somewhat better, with a specificity of 93.9%. From the findings, it is clear that the

Figure 4
Results of Filter NB and NB without filter approach with accuracy parameter

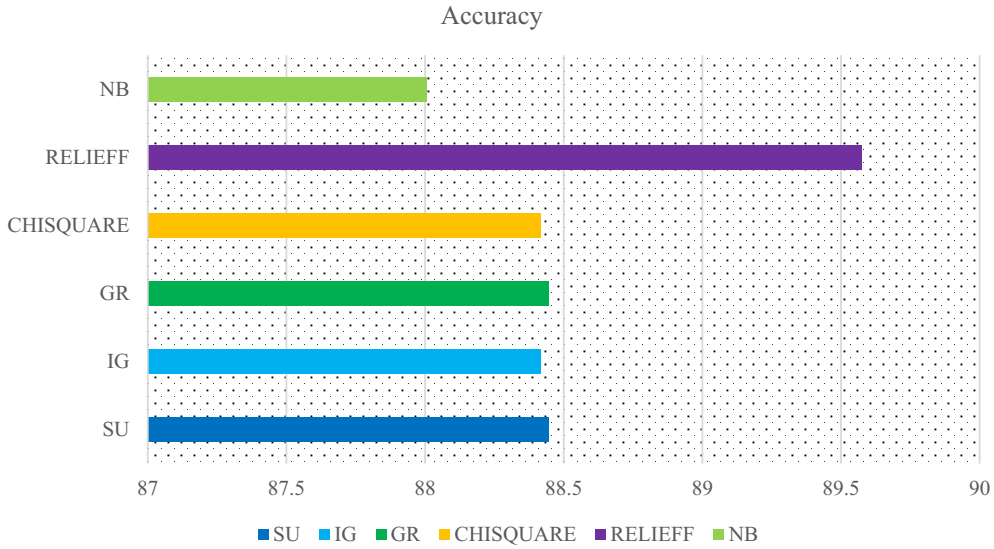


Figure 5
Results of Filter NB and NB without filter approach with specificity parameter

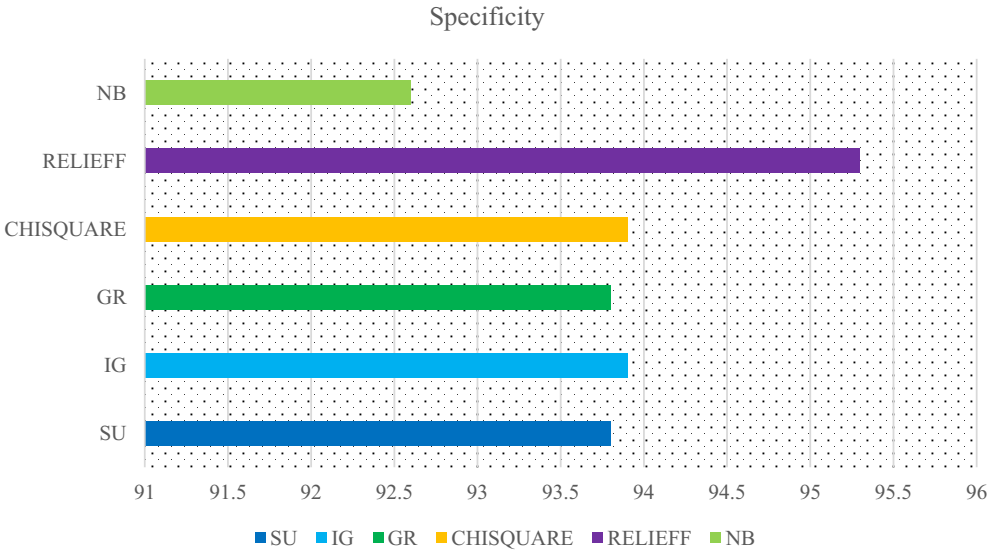
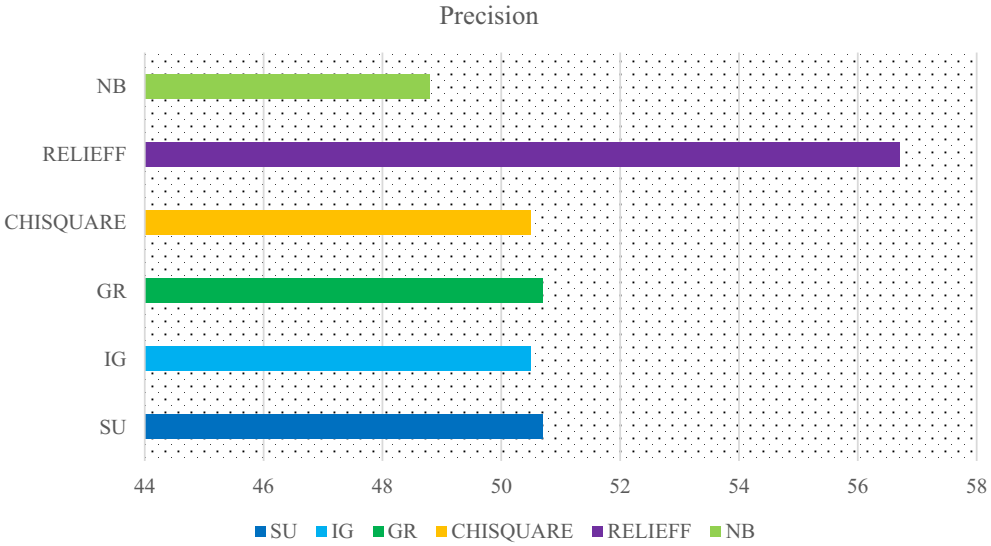


Figure 6
Results of Filter NB and NB without filter approach with precision parameter



RELIEFF and NB get a better feature subset and improve the NB model performance.

Figure 6 shows the outcomes of several FS strategies, Filter NB and NB without a filter approach, using precision as the evaluation criterion. Precision is a statistic that measures the accuracy of a classification model's positive predictions, expressing the ratio of true positives to the total of true positives and false positives. SU, IG, GR, CHISQUARE, and RELIEFF are among the FS techniques shown in Figure 6. The percentages of precision for each approach are presented. Notably, the RELIEFF approach outperformed the other strategies with the greatest precision of 56.7%. This suggests that RELIEFF, a relief-based FS technique, was especially helpful in improving the performance of the NB classifier in this setting. In comparison, the classic NB strategy generated an accuracy of 48.8%, suggesting that the integration of FS approaches enhanced classification precision in general. The precision of SU and GR was 50.7%, whereas IG and CHISQUARE attained a marginally lower precision of 50.5%. From the findings, it is clear that the RELIEFF and NB get a better feature subset and improve the NB model performance.

Figure 7 shows the outcomes of several FS strategies, Filter NB and NB without a filter approach, using FNR as the evaluation criterion. SU, IG, GR, CHISQUARE, and RELIEFF are among the FS techniques shown in Figure 7. The percentages of FNR for each approach are presented. Notably, the RELIEFF approach outperformed the other strategies with the greatest FNR of 53.7%. This suggests that RELIEFF, a relief-based FS technique, was especially helpful in improving the performance of the NB classifier in this setting. In comparison, the classic NB strategy generated an accuracy of 47.2%, suggesting that the integration of FS approaches enhanced classification precision in general. The FNR of CHISQUARE and IG was 53%, whereas GR and SU attained a marginally lower FNR of 52.6%. From the findings, it is clear that the RELIEFF and NB get a better feature subset and improve the NB model performance.

Figure 7
Results of Filter NB and NB without filter approach with FNR parameter

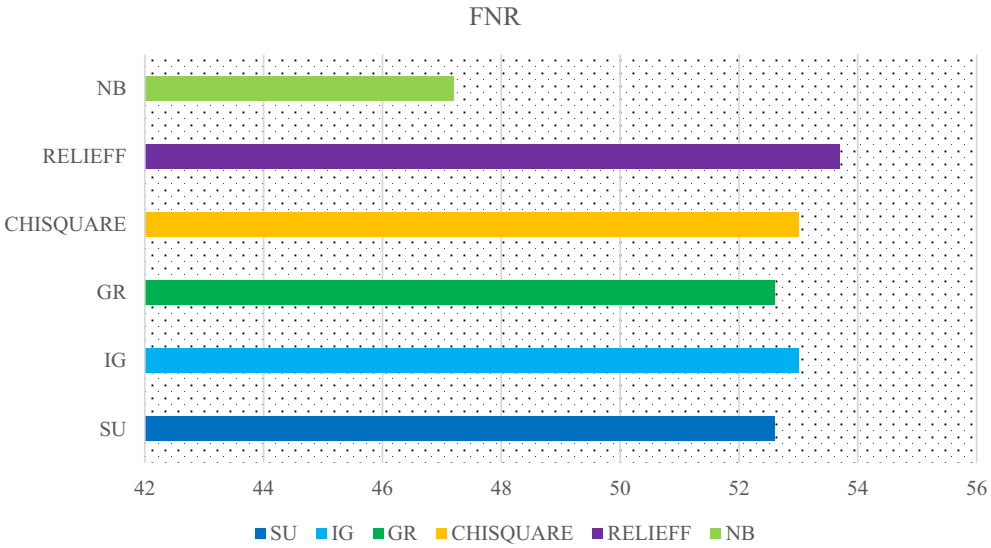


Figure 8
Results of Filter NB and NB without filter approach with FPR parameter

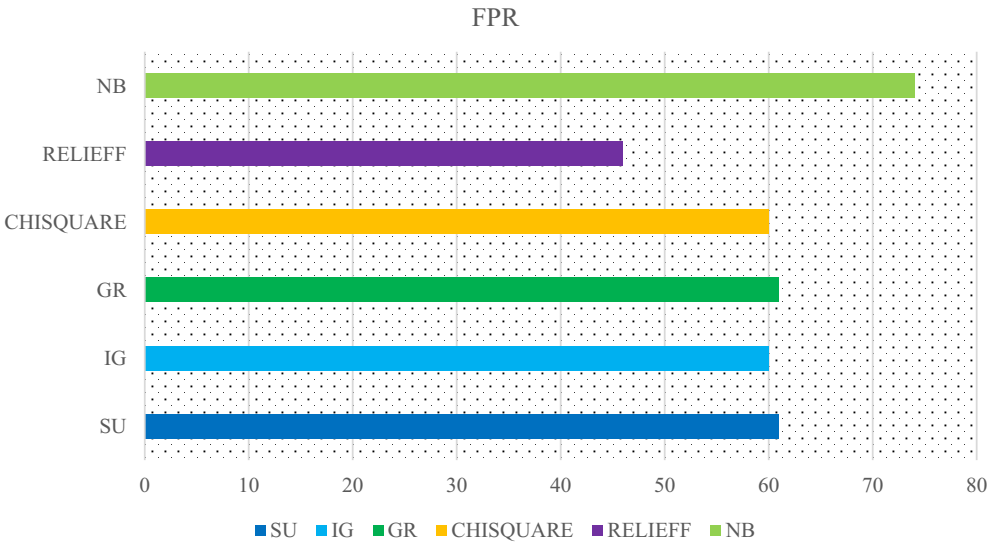


Figure 8 depicts the outcomes of several FS strategies, including Filter NB and NB without a filter, with an emphasis on the FPR parameter. The FPR quantifies the fraction of true negative cases that the model mistakenly identifies as positives, revealing the model's propensity to produce false alarms. FPR values for various FS approaches are given as percentages. RELIEFF had the lowest FPR at 46%, suggesting its efficiency in reducing false positive predictions. This shows that RELIEFF is effective at lowering false alarms, making it a suitable option for circumstances where avoiding false positives is critical. In comparison, the standard NB technique had the greatest FPR (74%), showing a greater proclivity for false positive predictions. FPR scores for SU, GR, IG, and CHISQUARE were in the middle, ranging between 60 and 61%. These findings highlight the importance of FS in reducing false positive predictions, with RELIEFF standing out as especially successful in this respect. Without FS, the NB model takes more time to run because it processes all available attributes. In contrast, applying FS before running NB reduces the execution time, as the model is trained and tested only on the most relevant attributes.

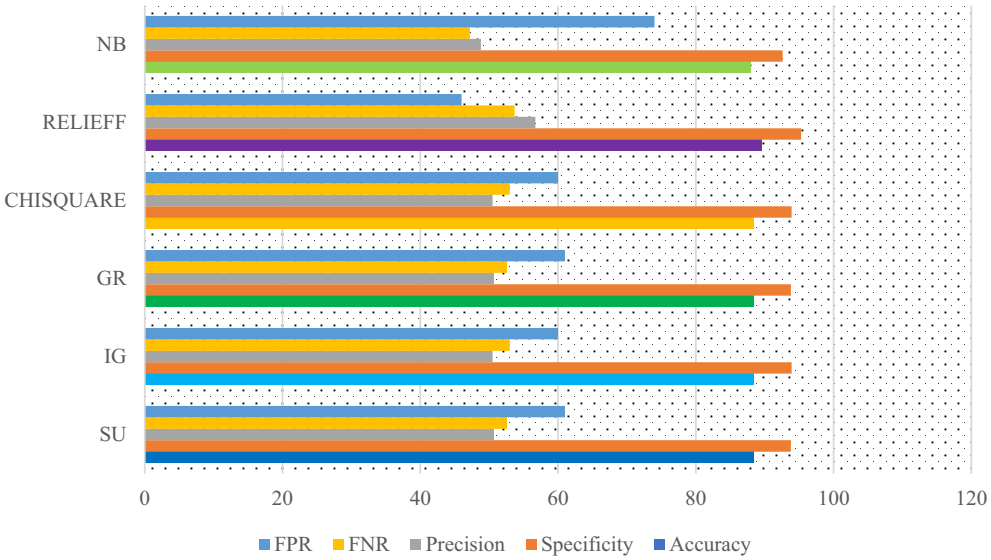
In summary, Figure 5 shows how various FS strategies affect the FPR of the NB classifier. Because RELIEFF may significantly minimize false positives, it is a potential technique for situations where minimizing false alarms is crucial to overall model performance.

6.1. Result discussion

Figure 9 depicts the performance of multiple FS strategies, Filter NB and NB without a filter, across a variety of evaluation criteria, including accuracy, specificity, precision, FNR, and FPR. Let's take a closer look at each statistic and the ramifications of the findings.

1) Accuracy: RELIEFF obtained the maximum accuracy of 89.5755%, beating all other approaches. This shows that

Figure 9
Results Filter NB and NB without filter approach with different parameters



RELIEFF is beneficial in terms of total classification accuracy. Comparative Accuracy: SU, IG, GR, and CHISQUARE all obtained approximately 88.4% accuracy, whereas NB without a filter technique reached 88.0073% accuracy.

2) Specificity: From the results, it shows that RELIEFF achieves a high specificity of 95.3%, showing its effectiveness in predicting real negatives. This shows that RELIEFF is superior at reducing false positives. SU, IG, GR, and CHISQUARE have comparable specificity values ranging from 93.8 to 93.9%, whereas NB had a slightly lower specificity of 92.6%.
3) Precision: From Figure 6, it shows that RELIEFF achieves a high precision of 56.7%, suggesting its capacity to make accurate positive predictions. This means that RELIEFF is effective at reducing false positives and increasing accuracy. SU and GR attained a precision of 50.7%, whereas IG and CHISQUARE produced a little lesser accuracy of 50.5%. NB had a precision of 48.8% without a filter.
4) FNR: RELIEFF achieves a high value of FNR of 53.7%, indicating a slightly greater rate of missing positive cases when compared to other approaches. This indicates a trade-off, since RELIEFF outperforms in other measures but may fall short in FNR. The FNR values for SU, IG, GR, and CHISQUARE were comparable, ranging from 52.6 to 53.0%, whereas NB without a filter technique obtained a lower FNR of 47.2%.
5) False Positive Rate (FPR): RELIEFF has the lowest FPR of 46%, suggesting its efficiency in reducing false positive predictions. As a result, RELIEFF is well-suited for situations where preventing false alarms is crucial. SU, IG, GR, and CHISQUARE had intermediate FPR values of 60–61%, whereas NB without a filter technique had the greatest FPR of 74%.

In summary, the overall RELIEFF approach performs better in choosing the feature subset and improving the NB performance. However, there are drawbacks, since it has a slightly higher FNR. Understanding these trade-offs is critical in determining the best FS strategy based on the application's unique needs. The findings also highlight the influence of FS on many elements of classification performance, demonstrating the significance of examining numerous metrics for a thorough review.

6.2. Research summary

The purpose of the study is to show how the FS approach helps to improve the ML model performance. The experimental results obtained are projected from Figures 4 to 9. The summary of the research is given below.

Compared to other approaches, RELIEFF achieves the best accuracy (89.5755%), specificity (95.3%), and precision (56.7%). RELIEFF had the lowest FPR of 46%, suggesting its efficacy in reducing false positive predictions. However, RELIEFF had a somewhat higher FNR of 53.7% when compared to other techniques.

Other FS strategies, such as SU, IG, GR, and Chi-Square, performed well across several measures. The conventional NB without a filter technique served as a reference, with a decent accuracy of 88.0073% but poorer specificity, precision, and a larger FPR.

The research focuses on the subtle influence of FS approaches on many elements of classification performance. Multiple metrics must be considered in order to make educated conclusions regarding the best FS strategy based on individual application needs.

The study provides practitioners and scholars with significant insights into FS approaches in the context of classification problems.

6.3. Research limitations

1) Dataset Dependency: The study outcomes are dependent on the unique dataset utilized for testing. The narrow breadth of the dataset may not adequately reflect the variety of real-world circumstances, and the findings may be difficult to generalize to other datasets with different features.
2) Algorithm Sensitivity: In this study, the methodology is considered for the NB algorithm with features and without features. Another different ML algorithm is not explored.
3) Limited FS strategies: The research investigated a limited number of FS strategies, including SU, IG, GR, CHISQUARE, and RELIEFF. Addressing these constraints in future studies might lead to a more nuanced and thorough knowledge of the

efficacy and application of FS approaches in the context of the NB classifier.

4) Generalizability of RELIEFF Performance: Currently, the study's experiments are limited to a single customer dataset from the UCI repository, so the generalizability of RELIEFF's and FS methods' performance is not explored.
5) A promising direction for future research is to employ a hybrid approach using filter methods for rapid dimensionality reduction, followed by wrapper or embedded methods to refine the feature set. This combined strategy can offer a balance between computational efficiency and the ability to model complex feature interactions, potentially leading to more robust and interpretable ML models.
6) While this work uses a single real-world dataset to establish a performance baseline, future research should replicate the methodology across diverse datasets (e.g., medical, financial, text mining) to evaluate generalization. This will also help test whether RELIEFF's observed advantage is domain-specific or consistent across contexts.

7. Conclusion

FS is finally illustrated to be a key component of the ML setting, aiding in the enhancement of model performance and addressing the challenges presented by high-dimensional datasets. The present study explored the essential role of FS and explained its intricate implications, particularly with the NB classifier optimization. Since FS may enhance model performance, lessen the chance of overfitting, and make interpretation easier, it is essentially significant. This work has shown the significance of this process and provided a comprehensive analysis of its use using filter FS approaches as a lens. A thorough evaluation framework defined the recommended strategy, which has been helpful in systematically assessing the performance of SU, IG, GR, CHISQUARE, and RELIEFF. Based on a wide variety of metrics, including accuracy and FNR, the study ensures a thorough understanding of the classifiers' advantages and disadvantages. The results of this comprehensive method provide an in-depth understanding of how filter FS approaches function. While RELIEFF's specificity, accuracy, and precision are commendable, a thorough trade-off analysis is necessary due to the slight increase in FNR. This emphasizes the need of carrying out an impartial evaluation and demonstrates that the optimal FS approach relies on the specific objectives of the application. In summary, this research contributes to the theoretical underpinnings of financial science and provides practitioners with useful assistance through the complexities of ML model optimization. The study explores the connection between theory and practice, and the findings provide helpful guidance for future research, broadening our understanding of how FS approaches impact real-world ML scenarios.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in the UCI repository at https://doi.org/10.1016/j.dss.2014.03.001, reference number [6].

Author Contribution Statement

Nalini Manogaran: Methodology, Resources. **Kalpana Vadivelu:** Formal analysis, Writing – original draft, Visualization. **Siva Subramanian Raju:** Conceptualization, Investigation, Writing – review &; editing. **Yamini Bhavani Shankar:** Software, Data curation, Visualization. **Balamurugan Balusamy:** Software, Validation, Data curation. **Sumendra Yogarayan:** Formal analysis, Writing – review & editing, Supervision, Project administration.

References

[1] Remeseiro, B., & Bolon-Canedo, V. (2019). A review of feature selection methods in medical applications. *Computers in Biology and Medicine*, *112*, 103375. https://doi.org/10.1016/j.compbiomed.2019.103375

[2] Venkatesh, B., & Anuradha, J. (2019). A review of feature selection and its methods. *Cybernetics and Information Technologies*, *19*(1), 3–26. https://doi.org/10.2478/cait-2019-0001

[3] Bolón-Canedo, V., Alonso-Betanzos, A., Morán-Fernández, L., & Cancela, B. (2022). Feature selection: From the past to the future. In *Advances in selected artificial intelligence areas: World outstanding women in artificial intelligence* (pp. 11–34). Springer International Publishing. https://doi.org/10.1007/978-3-030-93052-3_2

[4] Deng, X., Li, Y., Weng, J., & Zhang, J. (2019). Feature selection for text classification: A review. *Multimedia Tools and Applications*, *78*(3), 3797–3816. https://doi.org/10.1007/s11042-018-6083-5

[5] Maldonado, J., Riff, M. C., & Neveu, B. (2022). A review of recent approaches on wrapper feature selection for intrusion detection. *Expert Systems with Applications*, *198*, 116822. https://doi.org/10.1016/j.eswa.2022.116822

[6] Moro, S., Cortez, P., & Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, *62*, 22–31. https://doi.org/10.1016/j.dss.2014.03.001

[7] Bommert, A., Sun, X., Bischl, B., Rahnenführer, J., & Lang, M. (2020). Benchmark for filter methods for feature selection in high-dimensional classification data. *Computational Statistics & Data Analysis*, *143*, 106839. https://doi.org/10.1016/j.csda.2019.106839

[8] Alirezanejad, M., Enayatifar, R., Motameni, H., & Nematzadeh, H. (2020). Heuristic filter feature selection methods for medical datasets. *Genomics*, *112*(2), 1173–1181. https://doi.org/10.1016/j.ygeno.2019.07.002

[9] Ghosh, M., Guha, R., Sarkar, R., & Abraham, A. (2020). A wrapper-filter feature selection technique based on ant colony optimization. *Neural Computing and Applications*, *32*(12), 7839–7857. https://doi.org/10.1007/s00521-019-04171-3

[10] Pandit, A., Gupta, A., Bhatia, M., & Gupta, S. C. (2022). Filter based feature selection anticipation of automobile price prediction in azure machine learning. In *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)* (vol. *1*, pp. 256–262). IEEE, https://doi.org/10.1109/COM-IT-CON54601.2022.9850615

[11] Dafid, ., & Ermatita, . (2022). Filter-Based FS Method for Predicting Students' Academic Performance. In *2022 International Conference on Data Science and Its Applications (ICoDSA)*, *Bandung, Indonesia*. 309–314. https://doi.org/10.1109/ICoDSA55874.2022.9862883

[12] Nurhayati, O. D., & Surarso, B. (2021). Filter selection and feature extraction to distinguish types of CT scan images. In *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia*, 80–85. https://doi.org/10.1109/ISRITI54043.2021.9702847

[13] Maria, A. S., Sunder, R., & Antony, A. M. (2023). Stress prediction using enhanced FS and KNN model. In *2023 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA), Ernakulam, India*, 1–5. https://doi.org/10.1109/ACCTHPA57160.2023.10083348

[14] Lakineni, P. K., Reddy, D. J., Chitra, M., Umapriya, R., Kannan, L. V., & Barkunan, S. R. (2023). Optimal FS and classification using convolutional neural network-based plant disease prediction. In *2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS), Raichur, India*, 1–6. https://doi.org/10.1109/ICICACS57338.2023.10099630

[15] Girija, P., Sudha, K., Aswini, J., SivaKumar, S., & Nattesan, N. V. S. (2023). Alleviating the Naive Bayes assumption using filter approaches. In *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India*, 1430–1436. https://doi.org/10.1109/ICSSIT55814.2023.10061030

[16] Kanna, R. Rajesh., & Ulagamuthalvi, V. (2023). A novel detection on wheat disease through CL and RGB filters by LDA and QDA. In *2023 2nd International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN), Villupuram, India*, 1–5. https://doi.org/10.1109/ICSTSN57873.2023.10151536

[17] Dokeroglu, T., Deniz, A., & Kiziloz, H. E. (2022). A comprehensive survey on recent metaheuristics for feature selection. *Neurocomputing*, *494*, 269–296. https://doi.org/10.1016/j.neucom.2022.04.083

[18] Albulayhi, K., Al-Haija, Abu., Q, Alsuhibany., A, S., Jillepalli, A. A., Ashrafuzzaman, M., & Sheldon, F. T. (2022). IoT intrusion detection using machine learning with a novel high performing feature selection method. *Applied Sciences*, *12*(10), 5015. https://doi.org/10.3390/app12105015

[19] El-Kenawy, E. S. M., Mirjalili, S., Alassery, F., Zhang, Y. D., Eid, M. M., El-Mashad, S. Y., & Abdelhamid, A. A. (2022). Novel meta-heuristic algorithm for feature selection, unconstrained functions and engineering problems. *IEEE Access*, *10*, 40536–40555. https://doi.org/10.1109/ACCESS.2022.3166901

[20] Bommert, A., Welchowski, T., Schmid, M., & Rahnenführer, J. (2022). Benchmark of filter methods for feature selection in high-dimensional gene expression survival data. *Briefings in Bioinformatics*, *23*(1), bbab354. https://doi.org/10.1093/bib/bbab354

[21] Liu, H., & Setiono, R. (2022). Feature selection and classification–A probabilistic wrapper approach. In *Industrial and engineering applications or artificial intelligence and expert systems* (pp. 419–424). CRC Press. https://doi.org/10.1201/9780429332111

[22] Hashemi, A., Pajoohan, M. R., & Dowlatshahi, M. B. (2023). An election strategy for online streaming feature selection. In *2023 28th International Computer Conference, Computer Society of Iran (CSICC)*, 01–04. IEEE,https://doi.org/10.1109/CSICC58665.2023.10105319

[23] Sharma, A., & Mishra, P. K. (2022). Performance analysis of machine learning based optimized feature selection approaches for breast cancer diagnosis. *International Journal of Information Technology*, *14*(4), 1949–1960. https://doi.org/10.1007/s41870-021-00671-5

[24] Thakkar, A., & Lohiya, R. (2022). A survey on intrusion detection system: Feature selection, model, performance measures, application perspective, challenges, and future research directions. *Artificial Intelligence Review*, *55*(1), 453–563. https://doi.org/10.1007/s10462-021-10037-9

[25] Sahu, B., Panigrahi, A., Rout, S. K., & Pati, A. (2022). Hybrid multiple filter embedded political optimizer for FS. In *2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP), Hyderabad, India*, 1–6. https://doi.org/10.1109/ICICCSP53532.2022.9862419, 2022

[26] Zhang, B., Li, Y., & Chai, Z. (2022). A novel random multi-subspace based ReliefF for feature selection. *Knowledge-Based Systems*, *252*, 109400. https://doi.org/10.1016/j.knosys.2022.109400

[27] Badillo, S., Banfai, B., Birzele, F., Davydov, I. I., Hutchinson, L., Kam-Thong, T., . . . , & Zhang, J. D. (2020). An introduction to machine learning. *Clinical Pharmacology & Therapeutics*, *107*(4), 871–885. https://doi.org/10.1002/cpt.1796

[28] Wickramasinghe, I., & Kalutarage, H. (2021). Naive Bayes: Applications, variations and vulnerabilities: A review of literature with code snippets for implementation. *Soft Computing*, *25*(3), 2277–2293. https://doi.org/10.1007/s00500-020-05297-6

[29] Cheng, S., Quilodrán-Casas, C., Ouala, S., Farchi, A., Liu, C., Tandeo, P., . . . , & Arcucci, R. (2023). Machine learning with data assimilation and uncertainty quantification for dynamical systems: A review. *IEEE/CAA Journal of Automatica Sinica*, *10*(6), 1361–1387. 10.1109/JAS.2023.123537

[30] Zhong, C., Cheng, S., Kasoar, M., & Arcucci, R. (2023). Reduced-order digital twin and latent data assimilation for global wildfire prediction. *Natural Hazards and Earth System Sciences*, *23*(5), 1755–1768. https://doi.org/10.5194/nhess-23-1755-2023

How to Cite: Manogaran, N., Vadivelu, K., Raju, S. S., Shankar, Y. B., Balusamy, B., & Yogarayan, S. (2026). Performance Optimization of Naive Bayes Classification Using Filter-Based Feature Selection. *Journal of Computational and Cognitive Engineering*. https://doi.org/10.47852/bonviewJCCE62026710