Journal of Data Science and Modeling, Vol. 2, No. 2, 1-14, June 2024

Research Manuscript

Bayesian Nonparametric Estimation for Big Data Classification

Rashin Nimaei[*], Farzad Eskandari

Department of Statistics, Allameh Tabataba'i University, Tehran, Iran.

Recieved: 18/09/2024 Accepted: 22/12/2024

Abstract: The classification is a fundamental methodology in the research of data mining in recent years. The Naive Bayes classifier is one of the commonly used data mining methods for classification. Although the strong feature independence assumption in the Naive Bayes classifier makes it a tractable method for learning, this assumption may not hold in real-world applications. Correlated Naive Bayes classification is a generalization of the Naive Bayes classification model, considering the dependencies between features. One approach to reduce the Naivety of the classifier is to incorporate feature weights into the conditional probability. In this paper, we propose a method to incorporate feature weights into Correlated Naive Bayes based on the MapReduce Model. The performance of all described methods is evaluated by considering accuracy, sensitivity, and specificity metrics.

Keywords: Classification; Map-Reduce; Correlative Naive Bayes; Feature weighting.

Mathematics Subject Classification (2010): 62H30, 68T05.

[*]Corresponding Author: rashinnimaee@gmail.com

1. Introduction

Data analytics is considered one of the most challenging research problems, and data mining and machine learning methods are used to perform analysis. The two main categories, namely clustering and classification, are included in data mining schemes. The data classification process is primarily influenced by the various classifiers, such as Naive Bayes, Support Vector Machine, and Extreme Learning Machine. The Naive Bayes classification algorithm is widely used in data analysis and other fields because of its simple and fast algorithm structure. The approach of data classification with a machine learning algorithm is mostly based on multiple classification (Chen (2016)). Data analysis involves extracting useful insights from data, and one of the main tasks related to it is classification. Many algorithms have been developed to perform data analysis tasks. The compatibility of the MapReduce algorithm in handling data processing makes it more suitable for analysis tasks. MapReduce is a programming model that was first proposed by Google researchers for distributed parallel computing on massive amounts of datasets. From a functional point of view, programmers are looking to eventually map each word to its number of occurrences in all documents. This leads to providing the mapping function, which is tasked with collecting data segments for processing nodes, and the reduction function, which is tasked with collecting temporary results from all processing nodes and sorting them to reach the final results.

Classification algorithms are used to solve data mining problems related to data collection from different sources. For example, classification techniques such as Bayesian networks, genetic algorithms, genetic programming, and decision trees are used to perform big data classification Fong *et al.* (2016). It is found in the literature that data classification is primarily performed using machine learning methods, optimization algorithms, and decision trees along with some other approaches. For example, Alessio Bechini (2016) developed a classifier by combining the MapReduce concept with association rules and then named it the MapReduce-based Associative Classifier (MRAC). The advantage of this method was related to speed and scalability. Despite this, the proposed method did not perform well on large datasets. Lopez *et al.* (2015) proposed an algorithm based on fuzzy rule-based classification named Chi-FRBCS-BigDataCS to deal with large imbalanced data. This algorithm addresses uncertainty in big data and enhances the strength of learning in minimal classes. This algorithm performs fuzzy logic and implements MapReduce processes to design processes using a cost-sensitive learning approach. The performance of this algorithm is significantly improved based on classification accuracy and computation time. Deng *et al.* (2017) proposed a classification method of big data based on k-nearest neighbor. The k-means clustering method was used to divide the entire dataset into different parts, and KNN classification was performed in each part of the cluster. Banchhor and Srinivasu (2021) presented optimization algorithms for enhancing Bayesian classification for big data based on different functions incorporated within the MapReduce framework. The Correlative Naive Bayes classifier has been considered a basic model and it has been integrated with some algorithms, like cuckoo search and grey wolf optimization. Prasetiyowati and Sibaroni (2024) developed a novel approach that makes the Naive Bayes classifier capable of predicting future classification. In this work, the process of expanding the dimension of the feature matrix based on historical data from several previous time periods was implemented to obtain a long-term classification prediction model using Naive Bayes. Sakthia *et al.* (2024) proposed the big data clustering and classification model with improved fuzzy-based Deep Architecture under the MapReduce framework. In this algorithm, a deep hybrid model, which is the combination of a DCNN and Bi-GRU, is used for the classification process. The improved score-level fusion procedure is used in this case to obtain the final classification result.

Feature weighting is practical for alleviating Naive Bayes' primary weakness (see Zhang *et al.* (2021)). The naïveté in the classifier is that all the attributes are assumed to be independent given the class. This assumption simplifies the computation to infer the probability of a class given the data. Although the attribute independence assumption in the Naive Bayes classifier makes it a tractable method for learning, this assumption may not hold in real-world applications. Correlated Naive Bayes classification is a generalization of the Naive Bayes classification model, considering the dependencies between attributes.

In the attribute set, some attributes are more important than others, so they should have more influence on the final model than less-important attributes. Thus, feature weighting assigns a discriminative weight to each attribute. Then the feature weights are incorporated into the Correlated Naive Bayesian classification formula to build a feature-weighted Correlative Naive Bayesian model.

In this paper, we propose a new improved classification model called Feature-Weighted Correlative Naive Bayes (FWCNB) and give the general framework of FWCNB, which pays attention to feature weighting. Classification models include Naive Bayes classifier, Correlative Naive Bayes, and Correlative Naive Bayes with feature weighting. Feature weighting techniques are used to improve the simple Correlated Bayes classifier. Since the Naive Bayes approach cannot be directly used to classify data with a large number of features, we improve the Correlated Naive Bayes classifier by introducing a feature-weighted technique for handling a large number of features. The performance of all described methods is evaluated based on their accuracy, sensitivity, and specificity. All classification methods are implemented using the R programming language.

The organization of the paper is as follows: Section 2 presents a brief description of the big data classification algorithms. Section 3 discusses the performance of the proposed methods, and Section 4 concludes the paper.

2. Descriptions of Bayesian classification methods

2.1 Naive Bayes classifier

Naive Bayes (NB) classifier is one of the most efficient and practical algorithms for machine learning and data mining, which uses Bayes' theorem with the assumption of independence between attributes. Suppose $X = (A_1, A_2, \cdots, A_k)$ is an attribute vector, and A_i is the ith attribute with different values of x_i. Using the probability rule, we have:

$$P(X|C_k) = \prod_{i=1}^{n} P(A_i = x_i|C_k). \tag{2.1}$$

The maximum posterior classification in Naive Bayes can be given as

$$\arg\max_{k\in K}\{P(C_k) \times P(X|C_k)\}, \tag{2.2}$$

where K is the collection of all possible class labels k, n is the number of attributes, $P(C_k)$ is the prior probability of the class C_k, and $P(A_i = x_i|C_k)$ is the conditional probability of $A_i = x_i$ given the class C_k. NB predicts the class label with the highest posterior probability.

2.2 Correlative Naive Bayes classifier

In this section, the model construction of the Correlative Naive Bayes (CNB) classifier is presented. The CNB classifier is a probabilistic algorithm that is often used for classification tasks. It is an extension of the Naive Bayes classifier, which assumes that all attributes are independent of each other. In the training phase, the classifier creates the probability index table based on the features of the input training data samples and the class values (Banchhor and Srinivasu (2020)). Let B denote the database, represented as follows,

$$B = \{d_{j,l}\},\ l \leq j \leq r,\ 1 \leq l \leq m \tag{2.3}$$

where r is the total number of data points, $d_{j,l}$ is the data value, m is the total number of features, and m is the number of features in the data. The class to which the object belongs is expressed as,

$$C = \{C_1, C_2, \cdots, C_i, \cdots, C_k\},\ 1 \leq i \leq k \tag{2.4}$$

where k is the total number of classes.

The number of samples in the training data set depends on the unique class value that represents the number of classes corresponding to the data. Then, the mean and the variance of the data attribute in each class are computed as follows,

$$\begin{aligned} \mu_l^i &= \sum_{j=1}^{n} \frac{d_{j,l}}{n}, \\ \sigma_l^{2^i} &= \sum_{j=1}^{n} \frac{(d_{j,l} - \mu_l^i)^2}{n} \end{aligned} \tag{2.5}$$

where n is the number of samples in the ith class, μ_l^i and $\sigma_l^{2^i}$ represent the mean and the variance of the data values in the ith class, respectively. The CNB classifier performs a correlative analysis to improve the classification using a correlation function that is data-dependent. This includes a correlation factor represented as,

$$R = \frac{2}{m(m-1)} \sum_{l=1}^{m} \sum_{q=l+1}^{m} D(f_q, f_l), \tag{2.6}$$

where f_q and f_l are the qth and lth feature in the dataset, and $D(f_q, f_l)$ is the function that finds the relationship between f_q and f_l, as given below,

$$D(f_q, f_l) = \left[\frac{C(f_q, f_l) + 1}{2} \right], \tag{2.7}$$

where $C(f_q, f_l)$ is known as Pearson's correlation coefficient, which is used to compute the linear correlation between the two datasets. Therefore, the resulting training data matrix of the CNB classifier depends on the mean, the variance, and the correlation function, expressed as,

$$\{\mu_{k\times m}, \sigma_{k\times m}, R_{k\times m}\}$$

where $\mu_{k\times m}$ is the mean vector, $\sigma_{k\times m}$ is specified as variance, and $R_{k\times m}$ denotes the correlation function, and it is illustrated in vector form. The testing results are classified based on the probability index of the selected class with the attributes combined with the correlation function, by analyzing the training data:

$$\arg\max_{k\in K} \left[P(C_k) \times P(X|C_k) \times R^k \right]. \tag{2.8}$$

The CNB classifier selects the class having a maximum posterior probability value as the final class.

2.3 Feature-weighted Correlative Naive Bayes classifier

It is generally believed that the more an attribute feature appears, the more important it is, and the greater the corresponding weight in the model. Therefore, the weight coefficient of the feature is set as

$$w_i = \frac{N(A_i = x_i)}{N(D)},$$

w_i represents the proportion of the number of samples to the total number of samples when attribute A_i is x_i. In the FWCNB method, the weight of each attribute is incorporated into the CNB classification formula as shown in Eq. 2.9

$$C(X) = \arg\max_{k \in K} \left[P(C_k) \times P_w(X|C_k) \times R^k \right]. \tag{2.9}$$

where

$$P_w(X|C_k) = \prod_{i=1}^{n} P(A_i = x_i|C_k)^{w_i}.$$

To define the weight of each attribute, Hall (2007) suggested a decision tree feature-weighted Naive Bayes (DTAWNB) model, which calculates the dependencies between attributes through an unpruned decision tree trained from the training data. Jiang *et al.* (2019) proposed another improved model called correlation-based attribute-weighted Naive Bayes (CAWNB). In CAWNB, the weight of each attribute is defined as the difference between the mutual relevance and the average mutual redundancy. Later, a sigmoid transformation is performed to ensure that the weight is within a realistic range. The classification accuracy of CAWNB is higher than NB, while it maintains the simplicity of the final model. Recently, Chen *et al.* (2021) proposed an improvement of the NB algorithm by using feature weighting and Laplace calibration to obtain the improved Naive Bayesian classification algorithm. Their results show that when the sample size is large, the improved Naive Bayesian classification algorithm has high accuracy and is very stable.

The general framework of FWCNB is described in the following Algorithm. By executing the algorithm below, we can learn the feature weight vector w. To learn w, we employ a correlation-based feature-weighting approach.

```
Algorithm   FWCNB
Input:      $T$-a training dataset, $X$-a test dataset
Output:     the class label $C(X)$ of $X$
1:          Employ a feature-weighting method to learn the
            weight of each feature $w_i (i = 1, 2, \cdots, n)$
2:          Estimate the prior probability $P(C_k)$
3:          Estimate the conditional probability $P_w(X|C_k)$
4:          Predict the class label $C(X)$ of $X$ by Eq. 2.9
5:          return $C(X)$
```

Laplace calibration

When the number of training samples is small and the number of features is large, the training samples are not enough to cover so many features, so the number of samples of $A_i = x_i$ may be 0, and the whole category conditional probability $P(X|C_k)$ will be equal to 0. If this happens frequently, it is impossible to achieve accurate classification. The way to solve the problem is to use Laplace's calibration, which can completely solve the problem of category conditional probability being 0. In this case, the weight coefficient is defined as follows:

$$w_i = \frac{N(A_i = x_i) + 1}{N(D) + q_i},$$

where q_i represents the number of possible values of attribute A_i.

3. Numarical analysis and results

The purpose of this section is to investigate the classification performance of our proposed FWCNB. The comparative analysis provided compares the performance of the proposed algorithm with the standard NB and CNB competitors. For data classification, the method included in the developed classifiers is implemented using R programming.

3.1 Dataset description

The dataset utilized was taken from the UCI machine learning repository for experimentation (Localization dataset. https://archive.ics.uci.edu/dataset/850/raisin). The data relates to images of Kecimen and Besni raisin varieties grown in Turkey. These varieties of raisin are of great value and are thus important. A total of 900 raisin grains were used, including 450 pieces from each variety. These images were subjected to various stages of preprocessing, and seven morphological features were extracted. The morphological features and descriptions used in feature inference are given below.

Perimeter: It calculates the perimeter by measuring the distance between the boundaries of the raisin grain and the pixels around it.

MajorAxisLength: Gives the length of the major axis, which is the longest line that can be drawn within the raisin grain.

Eccentricity: Measures the eccentricity of the ellipse that has the same moments as the raisins.

Extent: Gives the ratio of the region formed by the raisin grain to the total pixels in the bounding box.

3.2 Data split

The purpose of splitting the data into training and testing sets is to evaluate the performance of the model on new and unobserved data. In this study, a percentage of data was split for the training set and the rest for the testing set using the dplyr library.

3.3 Implementation of algorithms

The implementation stage involves using the algorithm to process the data and generate predictions for the testing data. The predictions were then evaluated to determine the accuracy of the model. In this step, all three described algorithms were applied to the training dataset to build the Bayesian classification model.

During the implementation stage, the data is fed into the algorithm, and the algorithm calculates the conditional probabilities and correlation parameters for each attribute given the class label. The CNB classifier model is built by considering the correlation between features. The FWCNB classifier is obtained using the w-naive-bayes function in the rbooster library. These models are used to predict the test dataset. The predictions are then evaluated to determine the accuracy of the models. The performance evaluation process is carried out by varying the number of mappers and the training data. The number of mappers used is 4, representing the number of desktops used for the simulation of big data analysis, and the data sample for training varies from 75% to 95%.

The Confusion Matrix technique is a commonly used method for evaluating the performance of classification models. It provides a summary of the model's performance by showing the number of true positives, true negatives, false positives, and false negatives. These matrices are shown in figures 1 to 4. The results of the analysis of the classification models are presented in table 1 for the data.

3.4 Performance evaluation metrics

In this section, accuracy, sensitivity, and specificity metrics are introduced to evaluate the performance of classifiers. The degree of veracity is measured using an accuracy metric defined as the proportion of true results. The sensitivity and specificity are defined as the proportion of correctly identified true positives and true negatives.

$$Acc = \frac{TN + TP}{TN + TP + FN + FP}$$
$$Sens = \frac{TP}{TP + FN}$$
$$Spec = \frac{TN}{TN + FP}$$

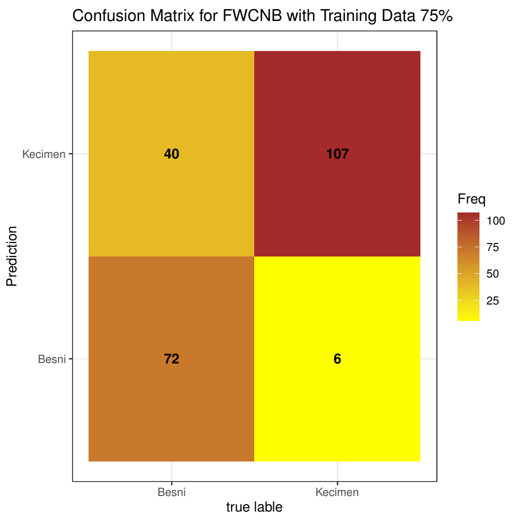

Figure 1: Confusion matrix for FWCNB with 75% training data

3.5 Comparative analysis of different Bayesian classifiers

This section illustrates the comparative analysis results obtained using dataset in all the comparative methods for the three mappers by varying the training data from 75% to 90%.

3.5.1 Performance evaluation of different Bayesian classifiers

The developed classifiers CNB and FWCNB are evaluated based on accuracy, sensitivity, and specificity on the localization dataset. The performance evaluation is presented in this section. The performance evaluation process is carried out with a mapper size of 3 and on the training data. Table 1 shows the analysis of NB, CNB, and FWCNB classifiers based on localization data.

Table 1: Comparative analysis using dataset.

Trainin data(%)	Mappers(M)		NB	CNB	FWCNB
75	2	Acc	75.6	77.8	79.6
		Sens	64.3	64.3	64.3
		Spec	86.7	92.0	94.7
	3	Acc	75.3	77.7	79.8
		Sens	64.5	63.6	64.7
		Spec	86.3	92.1	94.9
	4	Acc	75.3	77.9	79.7
		Sens	63.1	63.5	64.7
		Spec	86.5	92.3	94.8
80	2	Acc	74.4	78.9	81.7
		Sens	64.8	65.9	69.2
		Spec	84.3	92.1	94.4
	3	Acc	74.3	78.7	81.9
		Sens	68.9	65.7	69.4
		Spec	84.2	92.0	94.5
	4	Acc	74.2	79.0	81.9
		Sens	68.9	65.7	69.7
		Spec	84.5	92.0	94.4
85	2	Acc	80.0	81.5	84.4
		Sens	72.7	71.2	74.2
		Spec	87.0	92.3	94.6
	3	Acc	79.9	81.4	84.5
		Sens	72.8	71.5	74.4
		Spec	86.8	92.6	96.8
	4	Acc	79.8	81.7	84.6
		Sens	72.9	71.5	74.5
		Spec	86.8	92.3	96.8
90	2	Acc	82.2	83.3	84.7
		Sens	73.3	75.6	75.6
		Spec	91.1	91.1	95.3
	3	Acc	82.1	83.4	84.5
		Sens	73.2	75.7	75.3
		Spec	91.0	91.0	95.3
	4	Acc	82.0	83.4	84.7
		Sens	73.1	75.7	75.2
		Spec	91.1	91.1	95.5

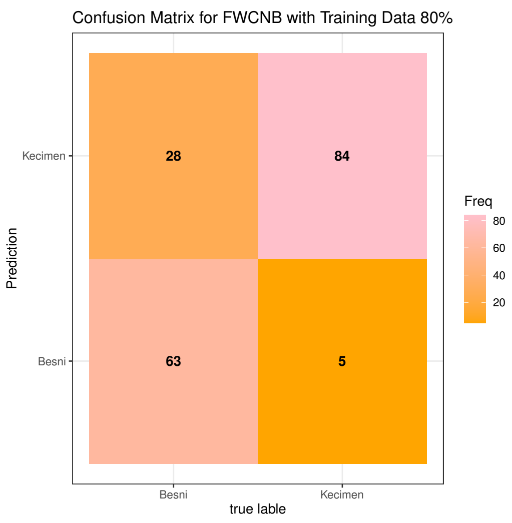

Figure 2: Confusion matrix for FWCNB with 80% training data

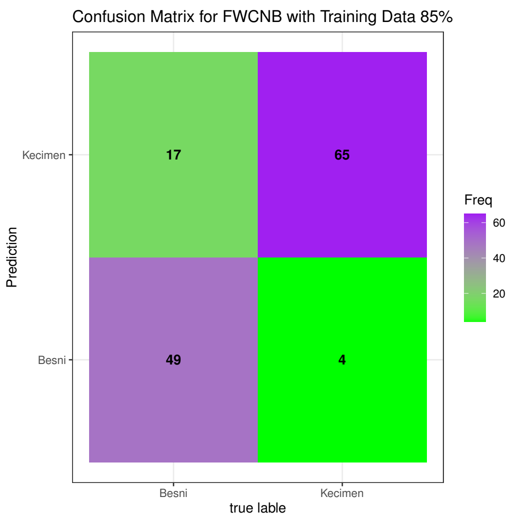

Figure 3: Confusion matrix for FWCNB with 85% training data

3.5.2 Analysis based on training percentage

Consider Table 1. The accuracy of the FWCNB classifier with 75% of training data is 79.6%. However, with an increase in the training percentage, the accuracy of the classifier improves. Similarly, for all the metrics, like sensitivity and specificity, improved performance is achieved with an increased training percentage.

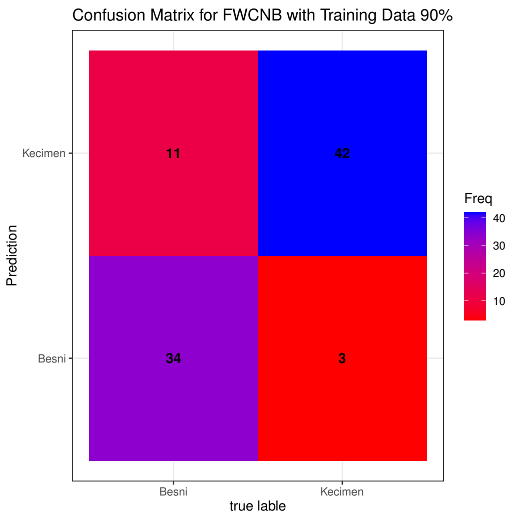

Figure 4: Confusion matrix for FWCNB with 90% training data

3.5.3 Analysis based on mappers

In Table 1, when considering the FWCNB classifier for $M = 3$ with a training percentage of 90%, the performance metrics, like accuracy, sensitivity, and specificity, are 84.5, 75.3, and 95.3, respectively. Similarly, for $M = 4$ with the training percentage of 90%, the performance metrics, like accuracy, sensitivity, and specificity, are 84.7, 75.2, and 95.5. From the above analysis, we can interpret that the sensitivity decreases with the increase in the mapper size. Moreover, it is observed that the FWCNB classifier has enhanced performance in accuracy, sensitivity, and specificity compared to other techniques. In this proposed work, the mapper size depicts the number of desktops used.

4. Conclusion

This paper presents a relevant and innovative approach to enhancing Bayesian classification using the MapReduce framework. FWCNB is a classifier developed by modifying the CNB classifier using a feature weighting technique. To evaluate the performance of the proposed approach, it is compared with the performance of two existing methods, such as NB and CNB, in terms of accuracy, sensitivity, and specificity. The CNB shows improved performance compared to NB, because the highest posterior value is only selected as a consequential class. FWCNB performs better than both NB and CNB, because the FWCNB algorithm is used to train the CNB classifier. While the CNB model achieved 78.9% accuracy, the proposed

FWCNB classifier achieved 81.7% accuracy. Therefore, it can be concluded that the proposed FWCNB approach can effectively perform big data classification using the CNB classifier, which could provide an accuracy of 81.7%, a sensitivity of 69.2%, and a specificity of 94.4%. Although the proposed classifier suggests high performance, the results may be affected by the datasets.

Acknowledgment

The authors would like to thank the Editor, Associate Editor, and anonymous reviewers for their constructive and valuable suggestions, which improved an earlier version of this article.

References

Alessio Bechini, F, M. A. (2016). Map Reduce solution for associative classification of big data. *Inf Sc.i*, **332**, 33–55.

Banchhor, C., and Srinivasu, N. (2020). Integrating Cuckoo search-Grey wolf optimization and Correlative Naive Bayes classifier with MapReduce model for big data classification, *Data & Knowledge Engineering*, **1(27)**, 101-788.

Banchhor, C., and Srinivasu, N. (2021). Analysis of Bayesian optimization algorithms for big data classification based on Map Reduce framework. *Banchhor and Srinivasu J Big Data*, https://doi.org/10.1186/s40537-021-00464-4.

Chen, J., Chen, H., Wan, X., and Zheng, G. (2016). MR-ELM: a MapReduce-based framework for large-scale ELM training in big data era. *Neural Comput Appl*, **27(1)**, 10-110.

Chen, H., Hu, S., Hua, R., and Zhao, X. (2021). Improved naive Bayes classification algorithm for traffic risk management, *EURASIP Journal on Advances in Signal Processing*, **30**, https://doi.org/10.1186/s13634-021-00742-6.

Deng, Z., Zhu, X., Cheng, D., Zong, M., and Zhang, S. (2016). Efficient kNN classification algorithm for big data. *Neurocomputing*, **195**, 143-8.

Fong, S., Wong, R., and Vasilakos, A, V. (2016). Accelerated PSO swarm search feature selection for data stream mining big data. *IEEE Trans Serv Comput*, **9(1)**, 33-45.

Lopez, V., del Rio, S., Benitez, J.M., and Herrera, F. (2015). Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data. *Fuzzy Sets Syst.*, **258**, 5-38.

Hall, M. A. (2007). A decision tree-based attribute weighting filter for naive bayes, *Knowl. Based Syst*, **20(2)**,120-126.

Jiang, L., Zhang, L., Li, C., and Wu, J. (2019). A correlation-based feature weighting filter for naive bayes, *IEEE Trans. Knowl. Data Eng.*, **31(2)**, 201-213.

Chen, C. L.P., and Zhang, C.Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data, *Information Sciences*, **275**, 314-347.

Prasetiyowati, S.S., and Sibaroni, Y. (2024). Unlocking the potential of Naive Bayes for spatio temporal classification: a novel approach to feature expansion, *Journal of Big Data*, **11**,106.

Sakthia, V.D., Valarmathib, V., Suryac, V., Karthikeyand, A., and Malathie, E. (2024). Bigdata clustering and classification with improved fuzzy based deep architecture under MapReduce framework, *Journal of Intelligent Decision Technologies*, **18(2)**, 1511-1540.

Tsai, C.F., Lin, W.C, and Ke, S.W. (2016). Big data mining with parallel computing: a comparison of distributed and MapReduce methodologies. *J Syst Softw.*, **122**, 83-92.

Zhang, H., Jiang, L., and Yu, L. (2021). Attribute and instance weighted naive Bayes, *Pattern Recognition*, **111**, 107-674.