Adapting Naïve Bayes Model for Text Classification with One-of and Imbalanced Multi-Class Problems

Ahood Almaleh[1†], Muhammad Ahtisham Aslam [2††] and Kawther Saeedi[2††]

Faculty of Computing and Information Technology, King Abdulaziz University, 21589 Jeddah, Saudi Arabia

Abstract
Increasingly interested in research communities, the text classification area enables the text or part of the text to be classified into classes for extracting useful information. Expensive to scale, the manual classification tasks are becoming vulnerable to potential unreliability as documents in the world increase, especially if the classes number more than two (multiclass classification). As a classification technique based on algorithms, automatic classification facilitates the automatic categorization of text documents to classes, thus resulting in reliable and efficient classification. This paper aims to describe the process of using the Naïve Bayes classifier for text classification with one-of and multiclass, especially in cases where the probability of imbalanced classes is higher. Our proposed process consists of a number of steps such as data preprocessing, classification model building, evaluating and predicting classes as final classification results.

Key words:
text classification, multi-class problems, text mining, machine learning.

1. Introduction

The term classification refers to a very general notion that means assigning an object to one or more classes. Classification may be done manually or algorithmically [1]. The manual classification tasks are based on traditional methods. For instance, such tasks may include the manual assigning of library books in accordance with Library of Congress categories by the librarian. While the accuracy of this task can be high, the downside is that it is expensive to scale and labor-intensive. The algorithmic classification called classification techniques are based on algorithms such as handwriting recognition, document classification and language identification[2]. Such classification techniques have many applications within and beyond database, data mining and information retrieval [3]. Also known as a supervised learning method, classification is a well-known technique, particularly in the domain of machine learning or data mining whereby the algorithm trains and learns from the input dataset and uses this training to classify the new dataset. In other words, the problem of classification techniques can be defined as a training dataset consisting of records, with each record being labeled by a class from a set of predefined classes [2]. The training records used to build a classifier can be used to classify new data to one of the class labels. The general illustration of classification process is presented in Figure1.

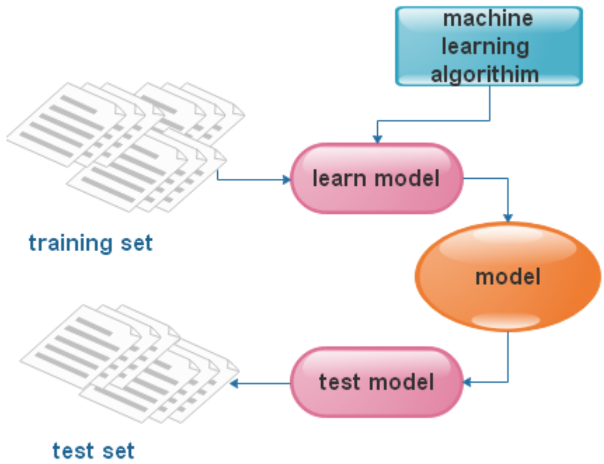

Fig 1 Classification process

In particular, the use of a classifier to classify the dataset in line with type of text is called a text classification problem [3]. Text classification, text categorization or topic classification can be used to map the documents to one or multiple predefined categories (also known as classes) based on the available information [4] as shown in Figure 2. In text classification, the classifier representing a document makes use of the bag of words model or lexicon model [5]. Most applications of text classification predominantly pertain to binary text classification. This refers to two-class classification problems such as spam detection (ham vs. spam), sentiment analysis (negative vs. positive), and gender identification (male vs. female). The real world problem needs to classify the text into more than two classes. This problem refers to a multiclass classification which is what will be explained in this paper. An example of this is organizing web pages into hierarchies and articles into topics. When the number of instances in one class make up

Manuscript received September 5, 2020
Manuscript revised September 20, 2020

DOI: 10.22937/IJCSNS.2020.20.09.11

more than 20% of the other classes, this is known as imbalanced dataset which leads to problems in training the classifier. Also, the text classification problem can comprise either any-of classification or one-of classification problem. Any-of problem means that the record is a member of more than one class or has more than two label values, whereas a one-of problem indicates that the record is a member of only one class value (classes are mutually exclusive) [1]. The most popular machine learning algorithms used for text classification are Naïve Bayes Classifier (NB) Logistic Regression(LR), , k-Nearest Neighbor (k-NN), Support Vector Machines (SVM) and Neural Networks (NN) [6]. The model explained in this paper makes use of the Naïve Bayes Algorithm. It is a practical and straightforward classification method that can apply to many different learning problems and is unlikely to produce models that fail in classification [1], [7]–[9].

Fig 1 Text classification task

Input
A document d
A predefined classes C= {c1,c2,c3,….cn}
Output
A predicted class c ∈ C for document d

The rest of the paper is organized as follows. Section 2 discusses the Naïve Bayes for text classification in detail. Section 3 illustrates the process of text classification with Naive Bayes for one of and imbalanced multiclass problems. Section 4 presents the conclusion of this paper.

2. Naive Bayes Algorithm

The Naïve Bayes (also known as Bayes Theorem) is a supervised learning algorithm based on probability prediction [10]. It is called "naïve" because the classifier assumes that there is independence amongst features and that the occurrence of one feature is unrelated to the occurrence of the other. This assumption is made to simplify the computations. However, when features are highly correlated, this assumption is considered as a disadvantage of the Naïve Bayes classifier. Despite this assumption, the Naïve Bayes' classifier has been producing satisfactorily results in many domains because of its design simplicity, efficiency, swiftness in classification, lack of reliance on large amounts of data for learning and superb performance with any sophisticated classification problem [4], [7], [8], [11]–[14]. Previous studies have proven that Naïve Bayes due to its simplicity and performance is a popular algorithm for text classification [4], [8], [11]–[14] [1], [7]–[9].Most studies on text classification are about computing the binary outcomes like spam filtering (spam vs. ham). This paper explains how to employ Naïve Bayes algorithm for one-of and multi-classification problems. Naïve Bayes (NB) finds the class of a given text document by using the combined probabilities of terms and classes based on the Bayes' Theorem. Considering a record R (text document) and a class as C, the Naïve Bayes classifies R as C which has the highest conditional probability using Bayes' rule. The formula for Bayes' theorem (as described in Equation 1) can be used to split any conditional probability into three sub probabilities:

$$p(C|R) = \frac{p(R|C)p(C)}{p(R)} \tag{1}$$

P(C) and P(R) are the probability of class and record without regarding each other.

P(C|R) is the probability of C conditional on R.

P(R|C) is the probability of R conditional on C.

Naïve Bayes estimates the best class C out of all classes for the record R by assigning the record to the class with the maximum of a posterior (MAP) probability while using the following Equation 2:

$$c_{map} = argmax_{c\in\{C\}}\, p(C|R) \tag{2}$$

We can then substitute Equation 1 for Equation 2 to get Equation 3:

$$c_{map} = argmax_{c\in\{C\}}\, p(C|R) = argmax_{c\in\{C\}} \frac{p(R|C)p(C)}{p(R)} \tag{3}$$

Since for each class C, the value of p(R) is the same, therefore, the value can be dropped to simplify Equation 3. All the matching classes for the matching record R are supposed to have the same probability P(R). This can result into a simpler equation such as shown in Equation 4:

$$c_{map} = argmax_{c\in\{C\}}\, p(R|C)p(C) \tag{4}$$

As a result, we can compute the MAP class by selecting the class which has the highest value of two probabilities:

P(R|C) the likelihood of the record R

P(C) the prior probability of the class

Each record contains a set of words. Equation 5 replaces a record with words w1, w2,….on:

$$c_{map} = argmax_{c\in\{C\}}\, p(w1, w2, w3, \dots \dots wn|C)p(C) \tag{5}$$

Often in text classifiers, the dataset is represented as a bag of words (BOW) which defines the words included in the dataset and the frequency time of each word without order. However, to simplify the computation of the probabilities of every possible set of words, Naïve Bayes classifiers make two simplified assumptions [14]. The first assumption is about the set of words. It is assumed that there is no impact of the position of word, and it has the same effect on classification regardless of where it occurs in the record. The second thing assumed in this case is the conditional independence. It is assumed that each word $p(w1, w2, w3, \dots \dots wn|C)$ is independent given the class c and hence can be 'naively' multiplied as follows in Equation 6:

$$p(w1, w2, w3, \dots \dots wn|C) = p(w1|c) \times p(w2|c) \dots \dots \times p(wn|c) \quad (6)$$

So, $p(w1, w2, w3, \dots \dots wn|C)$ in equation written as follow Equation (7):

$$p(w1, w2, w3, \dots \dots wn|C) = \prod p(wn|C) \quad (7)$$

Thus, the final equations of Naive Bayes (NB) classifier for classifying the records can be presented as follows in the form of Equation 8:

$$c_{NB} = \mathrm{argmax}_{c \in \{C\}}\, p(c) \prod_{w \in W} p(w|C) \quad (8)$$

Where NB is Naive Bayes
c the selected class from all classes C
w the word from all words W

Equation 8 shows multiplication of different conditional probabilities wherein the result can be in floating underflow[1], [14]. It is also recommended to do log space based calculation to increase the speed and to avoid the underflow. Equation 10 shows the addition of algorithms of probabilities in place of the multiplication of algorithms of probabilities:

$$c_{NB} = \mathrm{argmax}_{c \in \{C\}} \log p(c) + \sum_{w \in W} \log p(wi|c) \quad (9)$$

Equation 9 it is used in most implementations of NB. For estimating the maximum likelihood of the class prior P(c) compute the terms frequencies in training data, use Equation 10 as below:

$$p(c) = \frac{Nc(\mathit{the\ records\ number\ of\ class\ c\ in\ training\ data})}{Nr\ (\mathit{the\ total\ records\ number\ in\ dataset})} \quad (10)$$

For estimate probability of the word in the record for class c P(wi|c), use Equation 11:

$$p(wi|c) = \frac{\mathit{the\ frequency\ of\ the\ word\ in\ the\ class\ c}}{\sum_{w \in V} \mathit{the\ counts\ of\ all\ words\ in\ the\ class\ c}} \quad (11)$$

To create big-record for class c by grouping all records in this class, use frequency of the words. Here, V is the vocabulary (the distinct words in training data) rather than group of words in a class (C). When one or more of the probability terms P(wi|c) is zero, computing the probabilities of the class will be zero which is a potential complication in NB. In other words, with this maximum likelihood estimation of the training data not containing a word with the given class in order to determine the probability of this word, the likelihood of this word will be zero [14]. Since it naively multiplies all words likelihoods together, zero word probabilities cause the class probabilities to be zero[1]. The solution to eliminating zeros is add-one or Laplace smoothing as in Equation 12:

$$p(wi|c) = \frac{\mathit{the\ frequency\ of\ the\ word\ in\ the\ class\ c} + 1}{(\sum_{w \in V} \mathit{the\ counts\ of\ all\ words\ in\ the\ class\ c}) + 1} \quad (12)$$

Note that if there are some words that exist in test data but do not exist in the training data, these words will not occur in vocabulary[14]. The solution is to remove such words from the test data as well as any kind of probability for them. Such kind of resolution of non-useful words or noise will improve the performance as well as the accuracy of algorithm when applied on real life datasets. The Naive Bayes algorithm training and testing are shown in Figure 3.

Function LEARN NAIVE BAYES(R, C) **returns** log P(c) and log P(w_i|c)

for each class c ∈ C # Calculate P(c)

N_r = number of all records in dataset
Nc = number of records in class c
$logprior(c) = log\frac{Nc}{Nr}$

V ← extract vocabulary of training data R
big-record(r) ← append all records(r) for R with class c
for each word *w* in V # Calculate P(w_i|c) terms
count(w,c)← # of occurrences of w in big-record[r]
loglikelihood(w,c)← log$\frac{count(W|C)+1}{\sum_{wi\ inV}(count(wi|C)+1)}$
returns logprior, loglikelihood, V

Function TEST NAIVE BAYES(test_record, logprior, loglikelihood, C, V) **returns** best c

for each class c ∈ C
sum(c) ← logprior(c)
for each position i in test_record
word ←test_record [i]
if word ∈ V
sum(c)← sum(c)+ loglikelihood(word,c)
return $argmax_c$ sum[c]

Fig. 3 Proposed beam former.

3. Text Classification Process with Naive Bayes Algorithm

The general text classification processes are different from text classification process with one-of, imbalanced, and multiclass problems. Text preprocessing and representation steps are the same as the binary classification, while the changes are in the build of the model and evaluation steps. The classification process uses a supervised algorithm to predict the label or category as output for data instances based on their content. A Naïve Bayes is a supervised learning algorithm that can deal with the requirement of this paper[11]. Its needs labeled data as training and testing datasets. The data for training and testing the classifier must be the same in structure and proprieties as the data that requires classification. Therefore we begin by assuming, we have predefined classes to categorize the datasets to it and that we have labeled data (datasets with known classes) or in other words the training and testing datasets. Once the labeled dataset has been gathered, we preprocess the datasets, train and test the model, and predict the classes for the new data, as shown in the following subsections.

3.1 Data Preprocessing

Data Preprocessing helps to transform the raw data with noisy items into clean data to be processed efficiently and improves the performance of the classification model [15]. Preprocessing includes reviewing a sample from raw data (sampling) to determine what steps are needed to clean and to preprocess. The text data are varied in structure, and length. This is a hard type of data to preprocess and requires application of extensive cleaning and preprocessing techniques. In text datasets, we must first apply tokenization to separate individual terms. Tokenization tokens the text into a sequence of characters excluding punctuation and spaces and into one word “unigram”, two words “bigram”, or any number of words “n-gram” or sentence by using white-space as a word delimiter [16]. Then, using to lower techniques “converts all letters into lower cases”. Remove punctuation techniques “remove all special characters”. Remove stop words “remove the common English words (e.g., the, a, an, or) that are meaningless for the evaluation of the document content”. Remove the words addresses that rarely appeare (i.e. the frequency of words = 1), or the words appearing too frequently which do not contribute to identifying the class of the document. Remove whitespace is applied “if there is more than one space between words”. Remove numbers addresses “any numbers within the text”[17]. Stemming function is used to convert the words to their root forms [18]. For example, if we have the word “manager,” “management,” and “managers,” all is returned to one word “manag” as the root. These functions and techniques change according to the content of the data. This means that while one preprocessing technique may be useful for a specific dataset or language it may not be suitable for the others[19]. After preprocessing, the dataset is used for building the classification model.

3.2 Build the Naïve Bayse Classifier

In this work as a proof of concept, the datasets are classified into more than two predefined classes “multiclass text classification problem” where the intersection of classes must be zero “one- of” and the classes are imbalanced. The aim is to build the Naive Bayes classifier using the training dataset from the labeled data. Training dataset is a set of categorized or classified data used as an example to learn and to train the classifier. To get a high accuracy of the classifier and achieve good results in text classification, the model must train on data which is similar to new data which need to be labeled. A fair amount of data is required to get a good result by the model. The training of the classifier is the process of computing the probabilities of record r being in class c [9]. The goal in the text classification is to find the best probability class for the record. In Naïve Bayes classification, the best class is maximum *a posteriori* (MAP) class $C_{map,}$ as Equation 13:

$$c_{map} = argmax_{c\in\{c\}}\, p(c|r) = argmax_{c\in\{c\}} \log p(c) + \sum_{1\le i\le n_r} \log p(w_i\ |c) \qquad (13)$$

- Where, $P(w_i|c)$ is the conditional probability of word w_i occurring in a record belonging to class c. $P(w_i|c)$ is about how much the word w_i contributes that c is the correct class.
- P(c) is the prior probability of a record occurring in class c, choosing the class has a higher prior probability. $p(c) = \frac{N_c}{N}$ (The value of each parameter given the training data corresponds to relative frequency). Nc is the number of records in class c and N is the total numbers of records.
- w1,w2,……wn are the words in r we use for classification and n_r is the number of the words in the record r.

Now the classifier estimates the value from the training sets by multiplying the conditional probabilities, one for each position $1 \le i \le n_r$. The class with the highest probability is the most potent class. The result of this model contains knowledge used as input for classifying the new data. When one or more of the probability terms $P(w_i|c)$ is zero, computing the probabilities of the class will be zero which is a potential complication in NB. In other words, with this maximum likelihood of estimating, if the training data do not contain a word with the given class which want to estimate the probability of this word, the likelihood of this word will be zero [14]. Since it naively multiplies all the word likelihoods together, zero word probability causes the class probabilities to be zero[1]. The solution is add-one or which called Laplace smoothing. However, the classes distributed in labeled datasets are imbalanced. For this, use some techniques for ensuring all classes are divided equally and the model is trained for all types of class. These techniques can include generating a random number of samples from the training data, reproducing the starting point of the sequence and using technique to create balanced dataset by selecting the random sampling from every class and preserving the overall distribution of the dataset [20]. After that, the datasets are transformed into a word frequencies format which allow the model to make sense of this data. Representing the data as a corpus to count the words of records is called a bag-of-words approach. The frequency table for each word against each class in the training data is created by using the model. It is also used to calculate the initial weight of every record, and the probability for each word in the class [21]. The classifier learns what each class looks like and classifies the new data into the predefined classes based on the content of that data. The new data based on its contents is classified into the predefined classes. These classes are also used by the classifier for learning purposes. After training the Naive Bayes model, its performance is evaluated by using the testing data.

3.3 Prediction of Classes

The Naïve Bayes algorithms and the prior knowledge from the saved words and classes probabilities generated a classification model [25]. This model can be used for further classification at word level, both for training and test datasets. We set out to build the Naïve Bayes model that could classify the new data into classes based on the entire text. Here, we can use the existing set of classes as well as new classes can be proposed for further classification of words. Naturally, a subsequent step to evaluate the model validity was to compare the classes predicted by model with the classes expected by experts. It means, in addition to auto generated model, manual involvement of domain experts can improve the results as well as the accuracy of classification. The experts checked the classes against standards. The results of this comprised all datasets classified into one of the predefined classes. This result is the goal of the classification stage which builds the model and classifies the new data. The resulting classified data can be used for further prediction.

3.4 Evaluation

Before employing the Naïve Bayes model, its performance is tested by using the testing data. Testing dataset is a set of categorized or classified data which has never been used during the process of training. It is used to test the performances of the classifier. The model computes the probability of each class label in the usual way and picks the class with the largest probability [22]. The model compares each word in testing data with the saved words from the training step to determine the class of each instance. After developing the model and predicting the classes for each test instance, we have to compare the model prediction with the actual label to measure the performance of the model. The performance of machine learning algorithms is generally measured using the Accuracy metric [23]. The perfect performance of the classifier is when the accuracy is one. With imbalanced classification, the Accuracy metric is not appropriate [14]. The Alternative measures of classifier performance with imbalanced and multiclass datasets are Precision, Recall, and F1- metrics [12]. These evaluation measures are computed for each class as Equations 14,15, and 16. After obtaining the Precision, Recall, and F-measure for each class, the results are averaged to get the classifier performance[24].

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (14)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (15)$$

$$\text{F1} - \text{measure} = 2\text{X}\frac{\text{Precision.Recall}}{\text{Precision} + \text{Recall}} \quad (16)$$

TP stands for the correct classification "true positive" and TN for the instances classified correctly as not the class "true negative", while FP stands for negative instances incorrectly classified into the class "false positive", and FN stands for positive instances that are not recognized as a class "False Negative".

4. Conclusion

Text classification is a widely researched field. This paper demonstrated how to classify a text with *imbalanced* and *one-of multiclass* problems using the Naïve Bayes algorithm. The Naïve Bayes was chosen for its simplicity and efficiency, the ability to execute classification in many different problems and its capacity to learn from small training datasets. We discussed the process of text classification in detail. By supposing this, we labeled the data that can be used to train and test the model and we displayed the most used preprocessing techniques in text datasets. Then, we trained the Naive Bayes algorithms using the probability equations. After that, we explained how to evaluate the classifier when we have imbalanced and multi classes. Finally, we outlined the model validation and prediction. The text classification process of imbalanced multiclass and one-of is different from the general text classification wherein the instances in labeled data must have only one class (one of problem) to learn the classifier correctly. In the model training, we used some functions to distribute the train and test datasets in equal proportions to ensure the model learning for all classes (multiclass problem). Also in model evaluation, we excluded the Accuracy measure (imbalanced problem) and we used the Precision, Recall, and F-measure to determine the model performance carefully.

References

[1] C. D. Manning, P. Ragahvan, and H. Schutze, "An Introduction to Information Retrieval," Inf. Retr. Boston., no. c, pp. 1–18, 2009.

[2] A. Purohit, D. Atre, and P. Jaswani, "Text Classification in Data Mining," Int. J. Sci. Res. Publ., vol. 5, no. 6, pp. 1–6, 2015.

[3] C. Aggarwal and C. Zhai, "A Survey of Text Classification Algorithms," Min. Text Data, pp. 163–222, 2012.

[4] S. S. Roy, D. Kaul, R. Roy, C. Barna, S. Mehta, and A. Misra, "Prediction of Customer Satisfaction Using Naive Bayes, MultiClass Classifier, K-Star and IBK," in Soft Computing Applications, 2018, pp. 153–161.

[5] T. Elias and G. Reiten, "Classification with Multiple Classes using Naïve Bayes and Text Generation with a Small Data Set using a Recurrent Classification with Multiple Classes using Na ̈ ıve Bayes and Text Generation with a Small Data Set using a Recurrent Neural Network," 2017.

[6] F. S. Al-Anzi and D. AbuZeina, "Toward an enhanced Arabic text classification using cosine similarity and Latent Semantic Indexing," J. King Saud Univ. - Comput. Inf. Sci., vol. 29, no. 2, pp. 189–195, 2017.

[7] "DOCUMENT CLASSIFICATION USING MULTINOMIAL NAIVE BAYES CLASSIFIER." [Online]. Available: https://www.3pillarglobal.com/insights/document-classification-using-multinomial-naive-bayes-classifier.

[8] R. P. Rajeswari and K. Juliet, "Text Classification for Student Data Set using Naive Bayes Classifier and KNN Classifier," Int. J. Comput. Trends Technol., vol. 43, no. 1, pp. 8–12, 2017.

[9] C. D. Manning, P. Raghavan, and H. Schütze, "Text classification and Naive Bayes," Introd. to Inf. Retr., no. c, p. 260, 2008.

[10] C. Slamet, R. Andrian, D. S. Maylawati, Suhendar, W. Darmalaksana, and M. A. Ramdhani, "Web Scraping and Naïve Bayes Classification for Job Search Engine," IOP Conf. Ser. Mater. Sci. Eng., vol. 288, no. 1, 2018.

[11] S. Wang, L. Jiang, and C. Li, "Adapting naive Bayes tree for text classification," Knowl. Inf. Syst., vol. 44, no. 1, pp. 77–89, 2015.

[12] A. Ashari, I. Paryudi, and A. Tjoa, "Performance Comparison between Naïve Bayes, Decision Tree and k-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool," Int. J. Adv. Comput. Sci. Appl., vol. 4, no. 11, pp. 33–39, 2013.

[13] L. Dey, S. Chakraborty, A. Biswas, B. Bose, and S. Tiwari, "Sentiment Analysis of Review Datasets Using Naïve Bayes' and K-NN Classifier," Int. J. Inf. Eng. Electron. Bus., vol. 8, no. 4, pp. 54–62, 2016.

[14] J. Daniel and J. Martin, "Naive Bayes and Sentiment Classification," Speech Lang. Process. Stanford Univ., 2017.

[15] P. C. Gaigole, L. H. Patil, and P. M. Chaudhari, "Preprocessing Techniques in Text Categorization," Int. J. Comput. Appl., pp. 1–3, 2013.

[16] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," Inf. Process. Manag., vol. 50, no. 1, pp. 104–112, 2014.

[17] C. A. Gon, “The impact of Pre-Processing on the Classification of MEDLINE Documents,” 2015.
[18] M. Anjali and G. Jivani, “A Comparative Study of Stemming Algorithms,” no. November 2011, 2016.
[19] V. B. Kobayashi, S. T. Mol, H. A. Berkers, G. Kismihók, and D. N. Den Hartog, “Text Classification for Organizational Researchers: A Tutorial,” Organ. Res. Methods, vol. 21, no. 3, pp. 766–799, 2018.
[20] M. L. Jockers, Text Analysis with R for Students of Literature. 2014.
[21] G. A. A.Jabbar Alkubaisi, S. S. Kamaruddin, and H. Husni, “Stock Market Classification Model Using Sentiment Analysis on Twitter Based on Hybrid Naive Bayes Classifiers,” Comput. Inf. Sci., vol. 11, no. 1, p. 52, 2018.
[22] H. Zhang and D. Li, “Naive Bayes text classifier,” Proc. - 2007 IEEE Int. Conf. Granul. Comput. GrC 2007, no. 3, pp. 708–711, 2007.
[23] T. R. Patil, “Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification,” Int. J. Comput. Sci. Appl. ISSN 0974-1011, vol. 6, no. 2, pp. 256–261, 2013.
[24] Kavita Ganesan, “Text Mining, Analytics & More: Computing Precision and Recall for Multi-Class Classification Problems.,” 2015. [Online]. Available: http://text-analytics101.rxnlp.com/2014/10/computing-precision-and-recall-for.html. [Accessed: 10-May-2018].
[25] P. Domingos, “A few useful things to know about machine learning,” Commun. ACM, vol. 55, no. 10, p. 78, 2012.

Author’s Profiles

Ahood Almaleh received her master degree in Information System from King Abdulaziz University, Jeddah, Saudi Arabia, in 2019 and BS degree in Information System from King Khaled University, Abha, Saudi Arabia, in 2011. Currently she is working as Lecturer at the College of Computer Science and Information System, Najran University, Saudi Arabia. Ahood has published several journal papers in the domain of text classification and data analytics. Her, research interests includes big data, classification models and data analytics.

Muhammad Ahtisham Aslam is currently working as Associate Professor at Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia. He has been working as Senior Staff Research at Artificial Intelligence Lab, Knowledge Technolog Cluster, Malaysian Institute of Microelectronic Systems (MIMOS), Kuala Lumpur, Malaysia. He also has been working as Assistant Professor at COMSATS Institute of Information Technology, Lahore, Pakistan. He received his Ph.D. degree from University of Leipzig, Germany. His research interests include semantic web, linked open data, data science, and social network.

Kawther Saeedi is Assistant professor in information system department at king Abdulaziz University, Jeddah, Saudi Arabia. She is interested in applied research that involves new technology trends. Her core focus is software engineering, Blockchain and Data mining. With regards to professional experience, she has worked in projects of different scale and technologies. She is certified Professional Scrum Master, Senior Software Engineer and System architect. She is also an active team player and leading IS department since 2006. Kawther received her PhD Degree in Computer Science from the University of Manchester in 2014 and Master Degree in Engineering and Physical Science from the University of Manchester.