Towards the Deployment of Machine Learning Solutions for Document Classification

Article *in* INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING · March 2019

DOI: 10.26438/ijcse/v7i3.193201

2 authors:

Bichitrananda Behera
Pondicherry University
6 PUBLICATIONS **1** CITATION
SEE PROFILE

G. Kumaravelan
4 PUBLICATIONS **3** CITATIONS
SEE PROFILE

IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-3, March 2019 E-ISSN: 2347-2693

Towards the Deployment of Machine Learning Solutions for Document Classification

Bichitrananda Behera[1*], G. Kumaravelan[2]

[1,2]Department of Computer Science, Pondicherry University, Karaikal, India

[*]*Corresponding Author: bbehera19@gmail.com, Tel.: +918658512232*

DOI: https://doi.org/10.26438/ijcse/v7i2.193201 | Available online at: www.ijcseonline.org

Accepted: 22/Mar/2019, Published: 31/Mar/2019

***Abstract*—** In the era of internet-connected devices, the amount of unstructured data is multiplying in many different types of file formats. In particular, a great deal of knowledge is hidden in the vast amounts of textual data such as emails, blogs, tweets, and log files. The primary issue in this kind of textual data is to classify its content into predefined classes expeditiously in real time. Hence this research paper investigates the deployment of the state-of-the-art Machine Learning (ML) algorithms such as decision tree, k-nearest neighbourhood, Rocchio, ridge, passive-aggressive, multinomial naïve Bayes, Bernoulli naïve Bayes, support vector machine, artificial neural network including perceptron, stochastic gradient descent, back-propagation neural network in automatic classification of text documents on benchmark datasets such as 20Newsgroup, BBC news, BBC sports and IMDB. Finally, the performance of all the aforementioned built-in classifiers is compared and empirically evaluated using the well-defined metrics such as *accuracy*, *error rate*, *precision*, *recall*, *f-measure* and *Kappa statistics*.

***Keywords*—**Text mining, Machine learning, Documents classification, Information Retrieval, Comparative study

I. INTRODUCTION

In every minute tremendous amount of data is generated by the present fast ever growing digital world from different sources like the business, society, science, engineering, medicine and almost every other aspect of daily life [1]. According to the DOMO's fifth annual iteration of the informative report on "Data Never Sleeps 5.0" info-graphic, "Text is the new talk" because, in every minute of the day users edited 600 Wikipedia web pages, Twitter users send 456,000 messages and 103,447,520 spam emails. Meanwhile, digitalization also produces a large volume of text data through e-newspapers, e-books, e-magazines, e-journals, etc. Besides, most of the businesses also provide a huge amount of text documents in different forms such as customer support documents, everyday transactions, company profiles, competitor data, emails, technical reports, news articles, user reviews, etc. Hence, it is inevitable to classify the text documents automatically with more accuracy for information organization, storage, and retrieval.

In general, an automatic document classification algorithm assigns a predefined label to the instances of text documents (test dataset) based on the classifier model developed using the supervised machine learning algorithm. The built-in classifier model captures the inherent patterns and relationship based on the corresponding labels assigned to the given text documents (training dataset). Depending on the type of machine learning algorithm, automatic document classification can be classified into three broad categories namely supervised document classification, unsupervised document classification, and semi-supervised document classification. In supervised document classification, some external mechanism is needed manually to the classifier model which contributes information related to the precise document classification. In unsupervised document classification, there is no scope of having an external mechanism to provide information to the classification model to the correct document classification. In semi-supervised document classification, the partial amount of the documents is labeled by an external mechanism.

This paper focuses on supervised machine learning algorithms such as Decision Tree(DT), k-Nearest Neighborhood (K-NN), Rocchio(RC), Ridge, Passive-Aggressive(PA), Multinomial Naïve Bayes(M_NB), Bernoulli Naïve Bayes(B_NB), Support Vector Machine (SVM), Artificial Neural Network (ANN) including Perceptron(PPN), Stochastic Gradient Descent(SGD), Back Propagation(BPN), ensemble classifier such as Random Forest(RF) in automatic classification of text documents. Ultimately, the efficiency of these classifier models is measured by various performance metrics such as accuracy, precision, recall, F1-score, kappa score and error rate.

These machine learning algorithms are applied on benchmark text datasets such as 20 Newsgroup dataset, Movie Review Dataset, BBC News Dataset, and BBC Sports Dataset. The selection of these dataset becomes more appropriate to evaluate the findings with the state-of-the-art competing machine learning algorithms for the automatic classification of text documents.

In the literature, very few research work has been carried out which examines the state-of-the-art machine learning algorithms with benchmark competing dataset in one platform. Therefore, the main aim of this research paper is to perform an end-to-end performance analysis of all the prominent supervised machine learning algorithms for automatic document classification.

The rest of the paper is organized as follows. Section II elaborates the background details for text classification including mathematic formulation for document classification, document representation and literature review for text document classification using machine learning algorithms. Section III depicts, in a nutshell, the various machine learning algorithms used in this paper for document classification purpose. Section IV describes the novel experiments conducted towards the deployment of machine learning solutions in document classification. Section V gives the conclusion and suggests topics for further research.

II. BACKGROUND

A. Mathematical formulation

More specifically, the text classification problem needs three sets to define. First one is the training document set $D=\{d_1, d_2, \ldots d_n\}$, the second one is the class label set $C=\{c_1, c_2, \ldots c_n\}$ and finally is the test document set $T=\{d_1, d_2, \ldots d_n\}$. Each document d_i of the training document set D is labelled with a class label c_i from the class label set C, but each document of the test document set T has not been labelled. The main aim of text classification is to construct a text classification model, i.e., a *text classifier* from the training document set by relating the features in the text documents to one of the target class labels. After the classification model is fully trained, it can predict the class labels of the test document set. A mathematical formula of text classification algorithm both for training and testing is given below.

$$f:D \rightarrow C \qquad f(d)=c \tag{1}$$

B. Document representation

Document representation is a two-step process, namely *feature extraction* and *feature selection*. Feature extraction step includes several pre-processing activities to scale down the document complexity and to carry out the classification process in an accessible way. The pre-processing operation usually involves routines such as *stop-word removal*, *stemming of words*, *punctuation removal* and finally results in the tokenization process [2, 3]. Feature extraction step includes the calculation of *Term Frequency* (TF) and *Inverse Document Frequency* (IDF) from the tokenized documents. Finally, all the documents are normalized to unit length to perform classification in an efficient way. Basically, there are three most used models available in the literature for document representation namely *Vector Space Method* (VSM), *probabilistic models*, and the *inference network model* [4, 5, 6]. Once the document representation phase is finalized, then the document classifier is developed using the machine learning algorithm. Figure1 shows the general architecture of an automatic document classification process.

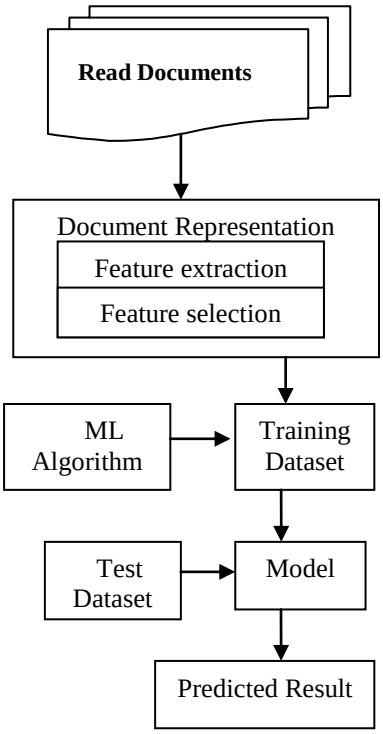

Figure 1. Text Document Classification Process Using ML Algorithm [7]

C. Literature review

F. Sebastiani [8] surveys about the different types of text document classification, application of text document classification, and discussed in detail the role of machine learning algorithms in automatic text document classification.

Colas et al. [9] compared the performance of SVM with K-NN and Naïve Bayes classification algorithms on Reuters-21570 dataset. The 10 fold cross-validation, precision, recall, F1-score and global execution time were used to evaluate the performance of each classifier. Overall the performance of SVM classifier works quite well, but if the pre-processing and

their internal parameters are adjusted well, then K-NN and Naïve Bayes also performs well.

S. Z. Mishu et al.[10] analyzed the performance of various supervised machine learning algorithms such as multinomial Naïve Bayes, Bernoulli Naïve Bayes, logistic regression, stochastic gradient descent, SVM, backpropagation neural network for classification on Reuters corpus, brown corpus and movie review corpus and concludes that backpropagation neural network is best among them.

A. Singh et al.[11] performed classification with Ridge classifier and Linear Discriminant Analysis(LDA) classifier on twitter dataset and concludes that Ridge classifier gives more classification accuracy than LDA classifier.

Rasjida et al. [12] compared the text classification performance of K-NN and Naïve Bayes classification algorithms by optimizing the parameters of both classifiers.

Samal et al. [13] measured the performance of most of the supervised classifiers for sentiment analysis using movie review dataset and concluded that SVM classifiers performed best among all classifiers for large movie review datasets.

M. Ghosh et al. [14] proposed composite feature vector CompUniBi for feature selection and applied four machine learning algorithm such as SVM, multinomial Naive Bayes (MNB), K-NN, and Maximum Entropy (ME) for text document classification. The proposed feature selection based classification performs better than unigram based feature selection based classification. The experiments were conducted on IMDB movie review, electronics review and kitchen product review dataset.

III. ALGORITHM FOR TEXT CLASSIFICATION

A. Decision Tree Classifier(DT)

In the decision tree classification model, the instances are the documents and attributes of each document are itself a *bag of words* or *terms*. The decision tree classifier [15] performs hierarchical decomposition of text documents of training text dataset by labelling its internal nodes with names of the text documents, branches of the tree with the test condition on *terms* and leaves of the tree with categories (labels). The test condition on *terms* may be of two types based on the document representation model. The first category of the test is to check whether a particular term available in the documents or not. The second type of test is to examine the weight of the *terms* in the text document. The first category of test will be used if document representation is of the form of the binary or Boolean model and the second category of test will be used if document representation is of the form of TF-IDF model.

During the training phase, the decision tree is built from the training dataset. While creating the decision tree from the training data set, different splitting criteria are used and most of the decision tree classifiers use single attribute split combination where the single attribute is used to perform the division [16]. The attribute or *term* whose information gain is high is considered as a base node and the procedure is repeated accordingly for selecting the remaining nodes.

Meanwhile in the testing phase, to predict the class label of a new unlabelled document $\vec{d}_j$, the decision tree classifier tests the *terms* of the $\vec{d}_j$ against the decision tree starting from the root node(base node) to until it reaches a leaf node and assigns the class label of the leaf node to the $\vec{d}_j$.

B. Naïve Bayes Classifier(NB)

Naïve Bayes classifier is a probabilistic classifier based on Bayesian posterior probability distribution. It holds the restriction with the independent relationship among the attributes through conditional probability. There are two variant of naïve Bayes classifier namely *Multivariate Bernoulli model (B_NB)* and *multinomial model (M_NB)* [17]. The multivariate Bernoulli naive Bayes model works only on binary data. Hence, in document pre-processing steps, each attributes corresponding to the list of documents in VSM must be either one or zero depending on the presence or absence of that particular attribute in that document [18]. On the other hand, the multinomial model works on the frequencies of attributes available in VSM representation of the documents. If the vocabulary size is small, then the Bernoulli model performs better than the multinomial model.

C. K-Nearest Neighborhood Classifier(K-NN)

Most of the classifiers in the literature spend more time in the training phase for building the classification model are considered as an *eager learner*. Nevertheless, k-NN classifier spends more time in the testing phase for predicting the class label of the new unlabelled test document. Hence, it is called as a *lazy learner*.

In the training phase of the model construction, k-nearest neighbor classifier stores all the training documents along with their respective target class. Meanwhile, in the testing phase, when any new test document comes for classification whose target class is unknown, k-nearest-neighborhood classifier finds the distance of the test document from all the other documents and assigns the class label of the training documents that is closest or most similar to the unknown document [19]. For this reason, k-nearest neighborhood classifier is known as an instant-based learning algorithm [19]. Euclidian distance and cosine similarity are the most frequently used approaches for measuring similarity quotient in finding the nearest neighborhood.

D. Support Vector Machine(SVM)

SVM is a type of classifier has the capability to classify both linear and nonlinear data [20]. The core idea behind the SVM classifier is, it first nonlinearly maps the original training data into sufficiently higher dimension let be *n* so that the data in the higher dimension can be separated easily by *n-1* dimension decision surface called *hyperplanes*. Out of all hyperplanes, the SVM classifier determines the best hyperplane which has maximum margin from the *support vectors*. Due to nonlinearity mapping, SVM classifier works efficiently on a large dataset and has been successfully applied in text classification [21].

E. Artificial Neural Network(ANN)

ANN is a kind of a data processing nonlinear model akin to the neural structure of the brain and it can learn from the existing training data to perform tasks like classification, prediction, decision-making, visualization, and others. It consists of a compilation of *nodes* otherwise called as *neurons* which are the centre of data processing in ANN. With context to the problem statement, these neurons are organized into three different layers namely *input layer*, *an output layer*, and *hidden layer*. In the context of text classification, the number of *words* or *terms* defines the neuron numbers in the input layer and the categories (class label) of documents define the number of neurons in the output layer. ANN can have at least one input layer and one output layer, but it may have many hidden layers depending upon the chosen problem. All links from the input layer to the output layer through hidden layers are assigned with some weights that represent the dependence relation between the nodes. When the neurons get weighted data, it calculates the weighted sum and it is processed by a well-known *activation function*. The output value from the activation function is fed forward to all the neurons in the input layer to map the correct neuron in the output layer. Some examples of well-known activation functions are *Binary step*, *Sigmoid*, *TanH*, *Softmax*, and *Rectifier Linear Unit* (ReLU) functions. ANN can be more flexible and more powerful by employing additional hidden layers. In particular, Perceptron (PPN), Stochastic Gradient Descent (SGD) neural network and Back-propagation neural network (BPN) are the three popular neural network based classifiers that extensively used for text classification.

F. Rocchio Classifier(RC)

Rocchio classification algorithm is defined on the concept of *relevance feedback* theory established in the field of Information Retrieval (IR) [22]. It uses the properties of *centroid* and *similarity measure* computations among the documents in the training and testing phase of model construction and usage respectively. In the training phase, the Rocchio classifier computes the centroid for each class from the relevant documents and establishes the centroid of each class as its representative. In testing phase to predict the class label of a test document, Rocchio classifier calculates its distance from the centroid of each class and assigns that class label with minimum distance.

G. Ridge Classifier(Ridge)

The Ridge classification algorithm is based on subspace assumption which states that samples of a particular class lie on a linear subspace and a new test sample to a class can be represented as a linear combination of training samples of the relevant class [23]. Let *X* is the training set and *X-test* is the testing dataset. For any new test sample $x \in X\text{-}test$, the goal of the ridge regression is to find the regression parameter vector $\hat{\alpha}_i$ to minimize the residual error as

$$\hat{\alpha}_i = \arg_{\alpha_i} \min \left\| x - X_i \alpha_i \right\|_2^2 + \lambda \left\| \alpha_i \right\|_2^2 \quad (2)$$

Here λ is the regularization parameter.

The regression parameter vector $\hat{\alpha}_i$ is used for the projection of a new test sample *x* onto the subspace of the i^{th} class as

$$\vec{x}_i = X_i \hat{\alpha}_i \quad (3)$$

After projecting the new test sample onto every class-specific subspace, the minimum distance is calculated between the new test sample *x* and the class-specific subspace $\vec{x}_i$ accordingly, the test sample is assigned to the class whose distance is minimum.

H. Passive Aggressive Classifier(PA)

The passive-aggressive classifiers belong to the family of large-scale learning algorithm [24]. The working principle of this kind of classifier is similar to that of *Perceptron* classifier; meanwhile, they do not require a learning rate. However, it includes a regularization parameter *c*. Figure 2 shows the pseudo code description of the Passive aggressive classifier.

INPUT: cost function $\rho(y, y')$
INITIALIZE: $\mathbf{w}_1 = (0, \ldots, 0)$
For $t = 1, 2, \ldots$

- receive instance: $\mathbf{x}_t \in \mathbb{R}^n$
- predict: $\hat{y}_t = \arg\max_{y \in \mathcal{Y}} (\mathbf{w}_t \cdot \Phi(\mathbf{x}_t, y))$
- receive correct label: $y_t \in \mathcal{Y}$
- define: $\tilde{y}_t = \arg\max_{r \in \mathcal{Y}} \left(\mathbf{w}_t \cdot \Phi(\mathbf{x}_t, r) - \mathbf{w}_t \cdot \Phi(\mathbf{x}_t, y_t) + \sqrt{\rho(y_t, r)} \right)$
- define:

$$q_t = \begin{cases} \hat{y}_t & \text{(PB)} \\ \tilde{y}_t & \text{(ML)} \end{cases}$$

- suffer loss: $\ell_t = \mathbf{w}_t \cdot \Phi(\mathbf{x}_t, q_t) - \mathbf{w}_t \cdot \Phi(\mathbf{x}_t, y_t) + \sqrt{\rho(y_t, q_t)}$
- set: $\tau_t = \frac{\ell_t}{\|\Phi(\mathbf{x}_t, y_t) - \Phi(\mathbf{x}_t, q_t)\|^2}$
- update: $\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t (\Phi(\mathbf{x}_t, y_t) - \Phi(\mathbf{x}_t, q_t))$

Figure 2. Pseudo Code for Passive aggressive Classifier [24].

I. Random Forest(RF)

Random forest is a *bagging* type ensemble learning algorithm for classification. In the training phase, it builds several decision tree classifiers from the random sub-sample of documents. In the testing phase, each decision tree performs prediction for a new test document and assigns that class label which is mostly predicted by all of the decision tree classifiers. The main advantage of random forest over the decision tree is it eliminates the problem of over-fitting and increases the classification accuracy.

IV. RESULTS AND DISCUSSION

A. Experimental Setup

Machine learning solutions for document classification has been implemented in python 3.6.7 and the experimentation is performed on a machine having Intel(R) Pentium(R) CPU 3825U processor 1.90 GHz with 4.00 GB of RAM. Four benchmark dataset namely, 20 Newsgroup, IMDB, BBC News, and BBC Sports has been used to perform an empirical evaluation of various machine learning algorithms mentioned in section 3. The description of this dataset is detailed below:

- **20 Newsgroup dataset:** This dataset size is 43.9MB and contains nearly 20000 documents. All the documents are divided into twenty newsgroups, and each newsgroup contains nearly one thousand Usenet articles. These articles are about atheism, computer graphics, motorcycle, pc hardware, etc and belong to any one category out of twenty. One subfolder represents one category of the 20newsgroup dataset. This dataset was donated by Tom Mitchell of Carnegie Mellon University in the year 1999. This dataset is available at UCI machine learning repository(https://archive.ics.uci.edu/ ml/ datasets/Twenty+Newsgroups)
- **Movie Review Dataset:** The Movie Review dataset has been collected from the NLTK corpora, and the dataset size is 7.42MB. This dataset is the processed subset of the unprocessed HTML files of IMDB archive, and it contains a total of 2000 text documents [25]. This dataset is available at http://www.nltk.org/nltk_data/
- **BBC News Dataset:** This dataset consists of 2225 documents, and those were collected in the year 2004-2005 by the BBC news website. The dataset size is 4.80MB, and all the documents are divided into five different categories like business, entertainment, politics, sports, and technology [26].
- **BBC Sports Dataset:** This dataset consists of 737 documents with a total of 4613 terms and its dataset size is 1.38MB. The entire document in this dataset is categorized into five sports group such as athletics, football, tennis, rugby, and cricket. BBC Sports collected these documents in the year of 2004-2005 [26].

Extensive experimentation was carried out with eighty percent of dataset contemplated for training and the remaining twenty percentage of dataset intended for testing respectively. Using python *Scikit-learn* ML library [27], and *TfidfVectorizer* envisages all the text pre-processing routines to build a dictionary and finally to transform all the documents to VSM representation. Subsequently, classification is performed with different ML algorithms and finally, the classifiers are evaluated using the well-established performance measures.

B. Performance Measures

Measures such as *Accuracy*, *Error Rate*, *Precision*, *Recall*, and *F1-Score* are used to evaluate the performance of the classifier [28]. Aforementioned measures are defined by means of the following features which actually defines the properties of the *confusion matrix* as shown in table1.

a. True Positive (tp_i): The documents which belong to class C_i is correctly predicted to class C_i by the classifier.
b. True Negative (tn_i): The documents which do not belong to the class C_i are correctly predicted to other class rather than class C_i.
c. False Positive (fp_i): The documents which do not belong to the class C_i are wrongly predicted to the class C_i.
d. False Negative (fn_i): The documents which belong to the class C_i are wrongly predicted to different class rather than class C_i.

Table 1.Confusion Matrix for class C_i

Total Documents		***Predicted Class***	
		C_i	***Not C_i***
Actual Class	***C_i***	*True Positive(tp_i)*	*False Negative(fn_i)*
	Not C_i	*False Positive (fp_i)*	*True Negative(tn_i)*

Now the performance measures are defined as follows

- Accuracy: *It is the average of per class ratio of correctly classified documents to the total documents.*

$$\frac{\sum_{i=1}^{n} \frac{tp_i + tn_i}{tp_i + fp_i + fn_i + tn_i}}{n} \quad (4)$$

- Error Rate: *It is the average of per class ratio of incorrectly classified documents to the total documents.*

$$\frac{\sum_{i=1}^{n} \frac{fp_i + fn_i}{tp_i + fp_i + fn_i + tn_i}}{n} \quad (5)$$

- Kappa Score (κ):

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \quad (6)$$

Where, $Pr(a)$ is the probability of the actual agreement and $Pr(e)$ is the probability of chance agreement between the classifier and the true values. However, the calculation of $Pr(a)$ and $Pr(e)$ depend on tp_i, fp_i, fn_i and tn_i.

- Precision: *It is the average of per class ratio of true positive prediction to total positive prediction.*

$$\frac{\sum_{i=1}^{n} \frac{tp_i}{tp_i + fp_i}}{n} \quad (7)$$

- Recall: *It is the average of per class ratio of true positive prediction to the total number of actual positive documents in the test set.*

$$\frac{\sum_{i=1}^{n} \frac{tp_i}{tp_i + fn_i}}{n} \quad (8)$$

- F1-Score: *It is evaluated as follows*

$$\frac{2(\mathrm{Pr}ecision \times \mathrm{Re}call)}{\mathrm{Pr}ecision + \mathrm{Re}call} \quad (9)$$

In all the above cases, *n* is the no of classes or labels in the dataset.

A. Hyper-parameters for different classifiers

The initialization of the input parameters among the different classifiers has a great impact on the classification performance measurements. Table 2 highlights the respective parameter setting procedures adopted in the experimental process of building the corresponding classifier.

B. Performance Analysis

The extensive experiment is conducted on different machine learning solutions or algorithms such as DT, M_NB, B_NB, K-NN, SVM, PPN, SGD, Ridge, RC, PA, RF and BPN algorithms for benchmark dataset like 20News group, Movie Review BBC News and BBC Sports. The performance of all the machine learning solutions is evaluated using different performance measures like accuracy, error rate, kappa score, precision, recall, and F1 score. These performance measurements will provide a general overview of each machine learning solution performance from a different perspective. Table 4 shows the performance measurements of machine learning solutions on various benchmark dataset for automatic document classification.

Table 2.Hyper-parameters settings of different classifiers

Classifiers	Parameters		
DT	Splitting= "Gini"	splitter="best"	min_samples_split =2
M_NB	alpha=0.01	fit_prior=True	class_prior=None
B_NB	alpha=0.01	binarize=0.0	fit_prior=True
K-NN	K=10	metric= "minkowski"	weights="uniform"
SVM	penalty factor="l2"	tolerance(tol)= "1e-4"	loss= "square_hinge"
PPN	max_iter= "50"	tolerance(tol)="1e -3"	n_iter_no_change= 5
SGD	alpha= "0.0001"	Maximum iteration="50",	loss="hinge"
Ridge	solver= "sag"	tolerance(tol)="1e -2"	max_iter=None
RC	metric= "Euclidean"	shrink_threshold= "None"	
PA	max_iter=" 50",	tolerance(tol)= "1e-3"	loss="hinge"
RF	n_estimator ="100"	Splitting= "Gini"	min_samples_split =2
BPN	max_iter=" 200"	Hidden layer size="100"	activation function="relu"

Table 3.Performance of classifiers with respect to execution time

Algorithms	Execution time in seconds for each dataset			
	20 News Group	**Movie Review**	**BBC News**	**BBC Sports**
DT	1.9245830	1.8230620	1.0971900	0.2343850
M_NB	0.0312410	0.0156080	0.0312620	0.0156490
B_NB	0.0625210	0.0312570	0.0282960	0.0156160
K-NN	0.7674350	0.3614240	0.1875060	0.0155980
SVM	0.6440480	0.1738950	0.3141970	0.1249910
PPN	0.0904940	0.0468730	0.0937660	0.0198900
SGD	1.4347270	0.5445320	1.0930660	0.3262330
Ridge	0.8590630	0.3015270	0.6878870	0.2343590
RC	0.0432680	0.0781250	0.0312480	0.0159990
PA	0.1405960	0.0675460	0.1446090	0.0468710
RF	6.1928300	2.6099860	2.4861950	0.7048670
BPN	691.8678800	256.6748260	194.8566910	51.8221200

The Execution time of different algorithms is provided in table 3.Execution time is the sum of training and testing time of the classification algorithm. Execution time plays a great role along with performance measures for comparing different classification algorithms.

The result for the 20News group dataset in table 2 shows that SVM classifier performs best among all the classifiers with respect to all the classification performance measures. However, Decision Tree classifier yields the lowest classification performance among all the classifiers for the 20Newsgroup dataset. Meanwhile, the remaining classifiers provide an average classification performance.

equal classification accuracy of 82 percentages. However, If Kappa score, precision, and recall are taken for ranking the classifiers then SGD classifier performs better than PPN and BPN classifier. On the other hand, if F1 score is alone chosen as the performance measure than the BPN classifier outperforms both the PPN and SGD classifier. Usually, for movie review, dataset Decision Tree classifier generates the lowest classification performance.

Table 4.Performance of classifiers (in %) using a different dataset

Dataset	Performance Measure	Classification Algorithms											
		DT	*M_NB*	*B_NB*	*K-NN*	*SVM*	*PPN*	*SGD*	*Ridge*	*RC*	*PA*	*RF*	*BPN*
20 News Group	**Accuracy**	71.70	91.13	89.40	84.87	92.48	89.96	90.71	92.20	88.60	92.29	87.81	91.64
	Error rate	28.30	08.87	10.60	15.13	07.52	10.04	09.29	07.80	11.40	07.71	12.19	08.36
	Kappa Score	65.64	89.29	87.20	81.74	90.93	87.89	88.79	90.59	86.25	90.70	85.28	89.92
	Precision	70.14	90.53	89.24	84.43	86.00	89.10	89.63	91.77	87.65	91.36	86.88	90.75
	Recall	69.85	90.14	88.27	83.80	91.61	88.94	89.55	91.02	87.17	91.15	86.20	90.57
	F1-measure	69.84	90.29	99.51	83.91	91.40	88.99	89.53	91.12	87.07	85.05	86.38	90.63
Movie Review	**Accuracy**	59.50	79.00	79.75	60.75	83.50	82.00	82.00	82.75	68.50	91.24	78.25	82.50
	Error rate	40.50	21.00	20.25	39.25	16.50	18.00	18.00	17.50	31.50	17.00	21.75	18.00
	Kappa Score	33.12	58.01	59.54	22.88	67.05	64.08	64.13	65.07	37.13	66.04	56.40	64.05
	Precision	66.65	78.02	79.84	68.02	83.66	82.26	82.65	82.77	68.70	83.11	78.30	82.16
	Recall	66.60	79.04	79.83	61.65	83.61	82.13	82.21	82.63	68.62	83.09	78.16	82.10
	F1-measure	66.49	79.00	79.75	57.35	83.50	81.99	81.96	82.49	68.49	83.00	78.19	82.00
BBC News	**Accuracy**	80.90	97.75	96.18	94.83	97.98	97.53	97.53	98.20	95.96	97.75	95.73	93.75
	Error rate	19.10	2.25	3.82	5.17	2.02	2.47	2.47	1.80	4.04	2.25	4.27	2.25
	Kappa Score	76.00	97.18	95.20	93.51	97.46	96.90	96.89	97.74	94.91	97.17	94.62	97.18
	Precision	80.68	97.03	96.00	94.83	97.88	97.28	97.43	98.15	95.87	97.62	95.94	97.61
	Recall	80.44	97.88	96.12	94.90	98.08	97.77	97.49	98.27	95.78	97.75	95.42	97.82
	F1-measure	80.26	97.72	96.00	94.74	97.96	97.47	97.45	98.20	95.82	97.68	95.65	97.70
BBC Sports	**Accuracy**	87.84	98.65	96.62	95.27	98.65	96.62	100.00	98.65	94.59	98.65	97.30	98.65
	Error rate	4.16	1.35	3.38	4.73	1.35	3.38	0.00	1.35	5.41	1.35	2.70	1.35
	Kappa Score	83.91	98.22	95.53	93.79	98.22	95.57	100.00	98.22	92.88	98.22	96.43	98.22
	Precision	87.88	99.30	98.33	95.67	99.30	96.97	100.00	99.30	95.46	99.30	98.64	99.30
	Recall	85.99	98.52	95.91	95.18	98.52	96.79	100.00	98.52	94.57	98.52	96.78	98.52
	F1-measure	86.47	98.87	97.01	95.22	98.87	96.76	100.00	98.87	94.83	98.87	97.62	98.87

For movie review dataset, SVM stands top among all the classifiers with the classification accuracy of 83.5 percentages. Not only with the classification accuracy, for other measures also SVM classifier ranked top among all the classifiers. Next to SVM, PA and Ridge classifier performs well. Meanwhile, SGD, PPN, and BPN classifier have an

The Ridge classifier shows the best classification performance among all the classifiers for BBC News dataset. Meanwhile, after Ridge classifiers SVM shows good performances. Next to SVM, M_NB, BPN and PA classifiers provide the same classification accuracy of 97.75 percentages. However, Kappa score is the same for all the

three classifiers. While considering the measures precision, recall, and F1 score, M_NB classifier shows better performance than the BPN and PA classifier. Hence, M_NB is preferred than BPN and PA for BBC News dataset.

For BBC Sports dataset, SGD classifiers estimate 100 percent of classification accuracy, kappa score, precision, recall, and F1 score. Next to SGD classifier BPN, PA, Ridge, and SVM classifiers have equal classification performance. For BBC News and BBC Sports dataset also Decision Tree classifier shows the least performance among all the classifiers. But DT gives more than 80 percent of classification accuracy performance compare to 20 Newsgroup and movie review dataset.

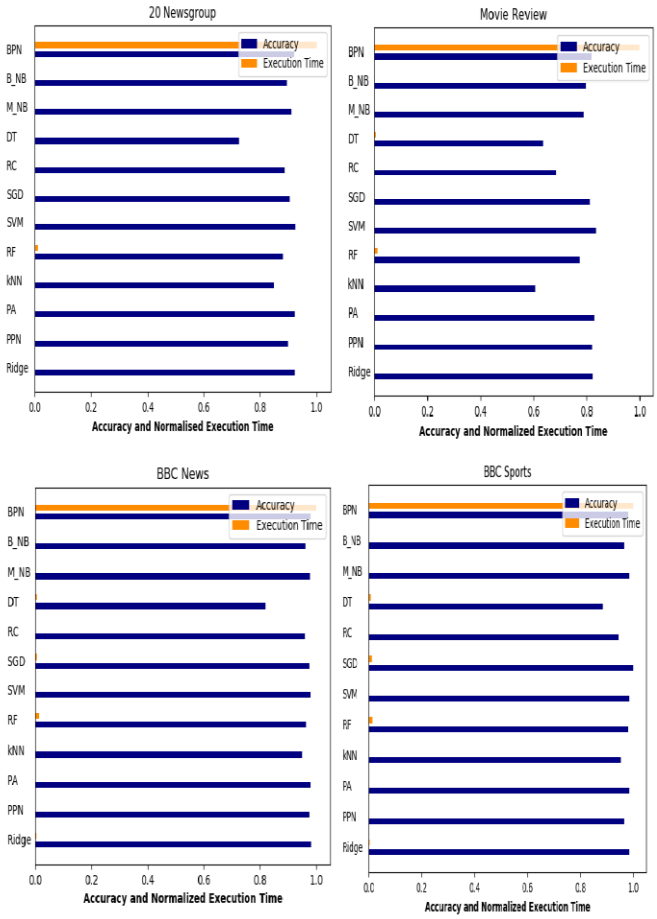

Figure 3. Performance comparison in Accuracy and Execution Time

Thus from table 3 it is clear that no one classifier is best for all the benchmarking dataset. From dataset to dataset the performance of the classifier varies. But, among all the classifiers BPN, SGD, Ridge, PA and SVM provide good classification performances and they have approximately posses the same classification performances. In particular, BPN classifier provides good classification accuracy, but it consumes more execution time.

The comparison of different algorithms with respect to accuracy and execution time is shown in figure 3. In figure 3, the accuracy values are in the range of 0 to 1 and the execution time of various ML algorithms on each dataset are normalized by dividing execution time of each algorithm by maximum execution time of an algorithm for concern dataset. The main purpose of having a normalized execution time is to compare the accuracy of the respective algorithms.

V. CONCLUSION AND FUTURE SCOPE

Automatic text document classification is one of the crucial areas of research in the field of Text Mining (TM). Consequently, machine learning based solutions provide a proven way for automatic text document classification. In this paper, the state-of-the-art ML algorithms have been applied to four benchmark dataset. In particular, classifiers like SGD, Ridge, Passive Aggressive, MLP and SVM provides good results on the chosen dataset compared to the other classifiers. However, the performance of KNN, Rocchio and Decision Tree classifiers has shown poor results for the chosen dataset compared to other classifiers. Other classifiers have average classification performance. The future scope is to make an improvement among those classifiers to adapt well in connection to the large-scale dataset. As a result, application of deep learning based models like multi-layer feed-forward neural networks, Convolution Neural Networks (CNN), Recurrent Neural Networks (RNN) and ensemble deep learning models becomes an evitable avenue of further research.

REFERENCES

[1] J. Han, M. Kamber, J. Pei, *"Data Mining: Concepts and Techniques"*, Morgan Kaufmann Publishers Inc., San Francisco, CA, 2011.

[2] S. Murugan, R. Karthika, "*A Literature Review on Text Mining Techniques and Methods*", International Journal of Computer Sciences and Engineering, Vol.06, Issue.02, pp.96-99, 2018.

[3] R. Lourdusamy, S. Abraham, "*A Survey on Text Pre-processing Techniques and Tools*", International Journal of Computer Sciences and Engineering, Vol.06, Issue.03, pp.148-157, 2018.

[4] A. McCallum, R. Rosenfeld, T.M. Mitchell, and A.Y. Ng,*"Improving Text Classification by Shrinkage in a Hierarchy of Classes"*. In ICML, Vol. 98. 359–367, 1998.

[5] C.D. Manning, P. Raghavan, and H. Schütze, *"Introduction to information retrieval"*,Vol. 1, Cambridge university press, Cambridge, 2008.

[6] H. Turtle and W.B. Croft,*"Inference networks for document retrieval"*, In Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, pp. 1–24, 1989.

[7] P.V. Arivoli, T.Chakravarthy, G.Kumaravelan, *"International Journal of Advanced Research in Computer Science"*, 8, (8), 299-302, 2017.

[8] F. Sebastiani. *"Machine Learning in Automated Text Categorization", ACM Computing Surveys*, 34(1), 2002.

[9] F. Colas ,P. Brazdil., *"Artificial Intelligence in Theory and Practice"*, ed. M. Bramer, (Boston: Springer), pp. 169-178, 2006.

[10] S. Z. Mishu, S. M. Rafiuddin, *"Performance Analysis of Supervised Machine Learning Algorithms for Text Classification"*, 19th Int. Conf. Comput. Inf. Technol, pp. 409-413, 2016.

[11] A. Singh, B. S. Prakash, K. Chandrasekaran*," A comparison of linear discriminant analysis and ridge classifier on Twitter data",* International Conference on Computing, Communication and Automation (ICCCA), pp. 133-138,2016.

[12] Z.E. Rasjida, R. Setiawan, *"Performance comparison and optimization of text document classification using k-NN and naïve bayes classification techniques",*Procedia Computer Science 2017; 116(C),pp.107-12, 2017.

[13] B.R. Samal, A.K. Behera, M. Panda, *" Performance analysis of supervised machine learning techniques for sentiment analysis"* Proceedings of the 1st ICRIL international conference on sensing, signal processing and security (ICSSS). Piscataway, IEEE, pp. 128–133. 2017.

[14] M. Ghosh and G. Sanyal, "*Performance Assessment of Multiple Classifiers Based on Ensemble Feature Selection Scheme for Sentiment Analysis"*, Applied Computational Intelligence and Soft Computing, vol. 2018, Article ID 8909357, 12 pages, 2018.

[15] Y. Li, A. Jain, *"Classification of text documents",*The Computer Journal, 41(8), pp. 537–546, 1998.

[16] C.C. Aggarwal and C. X. Zhai, *"Mining text data",* Springer, 2012.

[17] A. McCallum, Kamal Nigam, *"A comparison of event models for naive bayes text classification",* In AAAI-98 workshop on learning for text categorization, Vol. 752. Citeseer, pp.41–48, 1998.

[18] D.D. Lewis, *"Naive (Bayes) at forty: The independence assumption in information retrieval",* In Machine learning:ECML-98,Springer, pp.4–15. 1998.

[19] E.S. Han, G. Karypis, and V. Kumar, "*Text categorization using weight adjusted k-nearest neighborclassification. Springer",*2001

[20] C. Cortes, V. Vapnik*.," Support-vector networks. Machine Learning",* 20, pp. 273–297, 1995.

[21] H. Drucker, D. Wu, V. Vapnik, "*Support Vector Machines for Spam Categorization"*, IEEE Transactions on NeuralNetworks, vol. 10(5), pp.1048–1054, 1999.

[22] J.J. Rocchio, *"Relevance Feedback in Information Retrieval"* The SMART Retrieval System, pp. 313–323 ,1971.

[23] J. He, L. Ding, L. Jiang, L. Ma, "*Kernel ridge regression classification"*, Proceedings of the International Joint Conference on Neural Networks. pp.2263-2267, 2014.

[24] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. *"Online passive aggressive algorithms"*, Journal of Machine Learning Research, vol. 7, pp. 551–585, 2006.

[25] B. Pang and L. Lee*,"A Sentimental Education: Sentiment Analysis Using Subjectivity SummarizationBased on Minimum Cuts"*, Proceedingsof the ACL, 2004.

[26] D. Greene and P. Cunningham. *"Practical Solutions to the Problem of Diagonal Dominance in Kernel DocumentClustering"*, Proc. ICML 2006.

[27] F. Pedregosa et al,"*Scikit-learn: Machine learning in Python",* Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[28] M. Sokolova, G. Lapalme, *"A systematic analysis of performance measures for classification tasks"*, Inform. Process.Manage., vol. 45, no. 4, pp. 427-437, 2009.

Authors Profile

Mr. Bichitrananda Behera received his B.Tech. Degree in Computer science and Engineering from DRIEMS, Cuttack, Odisha, India, in 2011 and the M.Tech Degree in Computer science and Engineering from College of Engineering and Technology, Bhubaneswar, Odisha, India, in 2014 and Now pursuing Ph.D. Degree in Department of Computer Science, Pondicherry University, Karaikal Campus, Karaikal, India. He is having three years of teaching experience in relevant fields. His broad research area is in Machine Learning and Data Science.

Dr. G. Kumaravelan received his M.Tech Degree in Advanced Information Technology and Ph.D. in Computer Science from Bharathidasan University, Trichy, India in the year 2009 and 2013 respectively. He is currently working as Assistant Professor (Stage III) in Department of Computer Science, Pondicherry University, Karaikal Campus, Karaikal, India since 2009. He has published more than 20 research papers in reputed international journals including Thomson Reuters (SCI & Web of Science) and conferences including IEEE and ACM. His primary research work focuses on Machine Learning, Natural Language Processing, Big Data Analytics, and Human-Computer Interaction. He has 13 years of teaching experience and 4 years of Research Experience.

