Application of Supervised Machine Learning techniques for COVID-19 Text document Categorization

Sasmita Sahoo, Brojo Kishore Mishra, N.V. Jagannadha Rao

GIET University, Gunupur, India
Email: sasmitasahoo179@gmail.com, brojomishra@gmail.com, drnvj.rao@gmail.com

ABSTRACT : The COVID-19 pandemic forced the research community to discover different methods, ideas, and medicines to handle the pandemic. The research articles in COVID-19 literature have been growing exponentially, and manual classification of these articles is an impossible task. Therefore, automatic extraction and classification of COVID-19 related articles from the vast COVID-19 literature emerge as a significant task. Hence, thisthesis implements the vital Machine Learning (ML) algorithms like decision tree, k-nearest neighbourhood, Rocchio, ridge, passive-aggressive, multinomial naïveBayes, Bernoulli naïve Bayes, support vector machine, and artificial neural network classifiers such as perceptron, random gradient descent, and Backpropagation neural network in automatic classification of COVID-19 text documents on benchmark PubMed Abstract dataset. Finally, the performance of all the said constitutional classifiers are compared and evaluated utilizing the well-defined metrics like accuracy, error rate, precision, recall, and f-measure.

Keywords : COVID-19, Machine Learning, Classification Algorithms

1. INTRODUCTION

Ever since COVID-19 pandemic broke out in China, it took approximately 0.36 million valuable life and 5.8 million people were infected. The research community has been developing different innovative techniques and materials to tackle COVID-19 pandemic. More specifically, COVID-19 research publishes a large volume of electronic research articles leads to significant COVID-19 literature. Due to a large number of COVID-19 text documents in literature, manual categorization of COVID-19 documents becomes a tough task. Consequently, the automatic classifier model design for classifying COVID-19 documents grows a thrust area of research. Machine learning (ML) is a sub-field of artificial intelligence, which disseminates intelligence to the classifier model from the training data set. So that the build-in classifier model captures the inherent patterns and relationship based on the corresponding labels assigned to the given text documents (training data set). After the classifier developed from the training dataset, then it can automatically predict or classify the class label for a new text document.

Depending on the usage of the ML algorithms, automatic document classification task is often classified into three broad classes specifically 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, a partial amount of the documents is labelled by an external mechanism. This thesis focuses on the application of the supervised ML algorithms for COVID-19 text document classification. The Decision Tree(DT),k-Nearest Neighborhood (KNN), Rocchio(RC), Ridge, Passive-Aggressive(PA) classifier, Multinomial Naïve Bayes(MNB), Bernoulli Naïve Bayes(BNB), Support Vector Machine (SVM), Artificial Neural Network (ANN) classifier including Perceptron(PPN), Stochastic Gradient Descent(SGD), Back Propagation neural network(BPN) are the most prominent classifier found in the literature of supervised ML community[1].

In the COVID-19 literature, researchers have developed lots of ML techniques to fight with COVID-19 pandemic.Only a little work has done for COVID-19 text document classification and analysis using all the progressive machine-learning algorithms in one platform. This thesis exhibits the application of prominent supervised ML algorithms in COVID-19 text document classification. The well-defined performance metrics such as accuracy, error rate, precision, recall, and f-measure compares and evaluates the performance of the built-in classifiers on benchmark PubMed Abstract datasets. Therefore, the primary aim of this thesis is to perform an end-to-end performance analysis

of all the unique supervised ML algorithms for automatic text document categorization.

The organization of this thesis is as follows. Section 2 explains the background details for text classification process including preprocessing along with document representation, document classification that includes mathematic formulation for document classification, and literature review for COVID-19 text document classification using machine-learning algorithms: section 3presents, the various machine learning algorithms considered in this thesis for COVID-19 document categorization purpose. Section 4 holds the results and discussion on the experiments conducted for application of ML for COVID-19 text document classification. Section 5concludes the thesis and explain the possibilities of further research.

2. BACKGROUND:

2.1 TEXT CLASSIFICATION PROCESS

Text classification deals with unstructured text documents from differentrepositories like PubMed and Medline, web blogs, e-newspapers, medical reports, and social media. The primary aim of the text classification process is to predict a class label of the given test document with the prior knowledge of trained dataset. In general, text classification process involves three crucial steps: text preprocessing, text classification, and post-processing. Fig. 1 shows the various steps involved in building an automatic document classification model.

2.1.1 TEXT PREPROCESSING

Generally, in document classification, the first and crucial key component is text preprocessing, which has a high impact on classification performance. It usually consists of three tasks, namely*feature extraction*, *feature reduction*, and *document representation*.

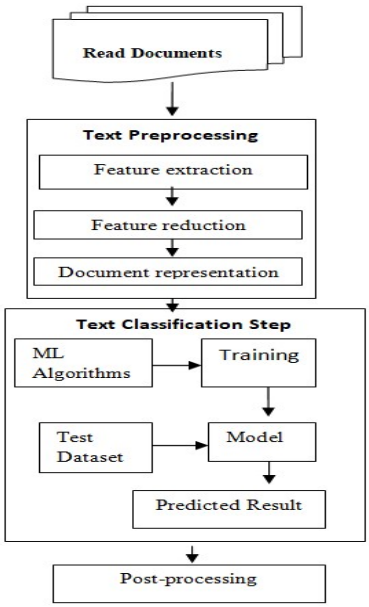

FIGURE .1 Text Document Classification Process Using ML Algorithm

FEATURE EXTRACTION:It includes many activities such as tokenization, filtering or stop-word removal, lemmatization, and stemming of words to scale down the document complexity and to present the classification method in an accessible manner.

- ***Tokenization:*** The input for tokenization activity is the raw text data or text document. It breaks [2] the sequence of strings from the given raw text data into small character pieces that can be a distinctive word, phrases, or keywords known as tokens.
- ***Filtering:*** It removes unwanted words from the documents so that more focus is given to essential words. *Stop-Words removal* is a well-known filtering method in which those words that are often used without meaningful content is get removed [3-4]. Examples of such stop-words are prepositions, conjunctions, and determiners.
- ***Lemmatization:*** In documents, there is varied inflected sort of words whose meaning are almost in the same nature. In such a situation, lemmatization is a kind of task which performs grouping of those words having similar meaning into one word by using vocabulary and morphological analysis of those words in that cluster of words.
- ***Stemming:*** It is the task of reducing derived words to their base or root form. Otherwise, it is like a crude chopping of affixes. For example, words like "running", and "runs" will be reduced to their base form like "run." Several stemming algorithms have been developed with time. In the field of Text Mining, Porter Stemmer is the mostly used stemming technique [5-7].

FEATURE REDUCTION: Normally, in a text document, the numbers of words otherwise called *features* are incredibly large, and those words play a vital role in document representation. Therefore, it is necessary to use the feature reduction methods to make a useful representation of the given text documents without changing the meaning of text data. Feature reduction methods are loosely divided into two categories, namely, *feature selection* and *feature transformation*.

- ***Feature Selection:*** It involves the selection of a subset of features that can equivalently represent the original physical meaning with a better understanding of data which leads to the elegant learning process [8]. The primary goal of the feature selection method is to reduce the curse of dimensionality to make the training dataset in the smaller size that can lead to lesser computational time. The advantage of reducing the curse of dimensionality is to increase the classification accuracy and to decrease the over-fitting problem. There are different types of feature

selection methods [9] available text mining literature namely Term Frequency (TF), Mutual Information (MI), Information Gain (IG), CHI-square statistic (CHI) and Term Strength (TS).

- ***Feature Transformation:*** It generates a new and smaller set of features by transforming or mapping the original set of features. Some well-known feature transformations methods are Latent Semantic Indexing (LSI) [10], Probabilistic Latent Semantic Indexing (PLSI) [11], Linear Discriminant Analysis (LDA)[12-13], and Generalized singular value decomposition methods [14-15].

DOCUMENT REPRESENTATION: Once the features are extracted from the raw text data, all the given documents are normalized to unit length to perform classification economically. There are three most used models on the market within the literature for document representation namely, Vector Space Method (VSM) [16], probabilistic models [17], and the inference network model [18]. Among the three models, VSM is the most used model, and the following section describes briefly about VSM.

It initially used for indexing and information retrieval. It converts documents into numerical vectors with the document set *D*; vocabulary set *V* and the term vector $\vec{t}_d$ for document *d*. Set $D = \{d_1, d_2, ..., d_D\}$ is a collection of Documents, the set $V = \{w_1, w_2, ..., w_v\}$ is a set of unique words or terms in the set *D* and the term vector $\vec{t}_d = (f_d(w_1), f_d(w_2), ..., f_d(w_v))$ where $f_d(w)$ represents the frequency of term $w \in V$ in the document $d \in D$ and $f_D(w)$ represents several documents contain the word w.

In VSM, the Boolean model and TF-IDF are the two-term weight schemes are used to calculate the weight of each feature. The Boolean model assigns $w_{ij} > 0$ to each term w_i if $w_i \in d_j$ and assigns $w_{ij} = 0$ $w_i \notin d_j$ if. However, the TF-IDF scheme calculates the term weight of each word $w \in d$ as follows.

$$q(w) = f_d(w) * \log \frac{|D|}{f_D(w)} \quad (1)$$

Where $|D|$ is the number of documents in the set D.

3. TEXT CLASSIFICATION STEP

Mathematically, the text classification problem wants three sets to outline. First one is the training document set $D = \{d_1, d_2, ..., d_n\}$, the second one is the category label set $C = \{c_1, c_2, ..., c_n\}$ and third one is the test document set $T = \{d_1, d_2, ..., d_n\}$. Every document d_i of the training document set D is labelled with a category label c_i from the category label set C; however, not every document of the test document set T has been labelled. The most 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 within the text documents to one of the target class labels. When the classification model is trained, it will predict the category labels of the test document set. Mathematical formula of text classification algorithm both for training and testing is given below.

$$f: D \rightarrow C \qquad f(d) = c \quad (2)$$

In equation 2, classifier assigns the proper class label to new document d (test instance). If a class label is assigned to the test instance, then this sort of classification is termed hard or multi-class classification, and on the other hand, classification is termed soft if a probability value is assigned to the test instance. In multi-label classification, multiple class labels are allotted to a test instance.

POST PROCESSING STEP

In post-processing step evaluation of the classifier is performed. The evaluation of the classification models is performed through various elegant performance measures like accuracy, precision, recall, and F-1 scores.

3.1 LITERATURE REVIEW

Nowadays, the COVID-19 information organization and access becomes a prominent requirement for research community because of the exponential growth of COVID-19 text documents. F. Sebastiani surveys [19] concerning the various types of text document classification, application of text document classification, and mentioned the role of machine learning algorithms in automatic text document classification thoroughly. Aaron M. Cohen developed [20] a replacement classification algorithm by assembling SVM with rejection sampling and chi-square based feature selection technique for automatic document classification. The TREC 2005 genomics track biomedical dataset was used to compare the classification performance of the classifier with a different variant of SVM classifier.

Hayda Almeida et al. conferred supervised machine learning approaches like Naive Bayes, Support Vector Machine and provision Model Trees to perform text classification of PubMed abstracts, to support the triage of documents [21].Samal et al. measured the [22] 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.S. Z. Mishu et al. 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 concluded that backpropagation neural network is best among them [23].Xiangying Jiang et al. applied Naïve Bayes (NB) and Random Forest (RF) for classifying biomedical publication documents associated with mouse gene expression database [24].

T.T. Nguyen [25] did a survey on different AI methods like data analytics, text mining and natural language processing which to fight against COVID-19 pandemic. L. Li et al. [26] in their research paper tried to classify social Medias data related to COVID-19 into seven categories of situational information using ML techniques like support vector machines (SVM), random forest (RF), and Naive Bayes (NB). The situational information is helpful to understand public sentiments and also forecast the spread of COVID-19 pandemic.J. Samuel et.al.[27] applied ML methods like Naïve Bayes, logistic regression, linear regression and K-Nearest Neighbor algorithms on twitter text data for fear-sentiment analysis of United States people over COVID-19 pandemic.

4 SUPERVISED ML ALGORITHMS FOR TEXT DOCUMENT CLASSIFICATION

4.1 DECISION TREE CLASSIFIER(DT)

In the decision tree classification model, the instances are the documents and attributes of every document are itself a bag of words or terms. The decision tree classifier [28] performs hierarchical decomposition of text documents of training 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 could also be of two varieties supported the document representation model.

The first category test is to test whether a selected term out there within the documents or not. The second kind of test is to look at the weight of the terms within the text document. The primary class of the test is used if document representation is of the shape of the binary or Booleanmodeland also the second category of the test will be used if document representation is of the form of TF-IDF model. During the training phase, the decision tree is made from the training dataset, whereas making the decision tree from the training data set, totally different splitting criteria are used, and most of the decision tree classifiers use single attribute split combination wherever the one attribute is employed to perform the division [29]. The attribute or term whose information gain is high is considered as a base node, and also the procedure is continual consequently for choosing the remaining nodes. Meanwhile within the testing phase, to predict the category label of a new untagged document, the decision tree classifier tests the terms of the against the decision tree ranging from the root node (base node) to until it reaches a leaf node and assigns the category label of the leaf node.

5 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 is two variant of naïve Bayes classifier, namely *the multivariate Bernoulli model(B_NB)* and *multinomial model(M_NB)* [30]. 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 [31]. On the other hand, the multinomial model works on the frequencies of attributes available in the VSM representation of the documents [32]. If the vocabulary size is small, then the Bernoulli model performs better than the multinomial model.

6 K-NEAREST NEIGHBORHOOD CLASSIFIER(K-NN)

Most of the classifiers within the literature pay longer in the training part for building the classification model are considered as an *eager learner*. However, k-NN classifier spends longer within the testing part for predicting the category label of the new untagged test document. Hence, it is known as a *lazy learner*.

In the training section of the model construction, k-nearest neighbour classifier stores all the training documents together with their target class. Meanwhile, in the testing phase, once any new test document comes for classification whose target class is unknown, k-nearest-neighbourhood classifier finds the distance of the test document from all the training documents and assigns the category label of the training documents that is nearest or most like the unknown document [33]. For this reason, k-nearest neighbourhood classifier is thought of as an instant-based learning algorithm [33]. Euclidian distance and cosine similarity are the foremost oftentimes-used approaches for measurement similarity quotient to find the nearest neighbourhood.

7 SUPPORT VECTOR MACHINE (SVM)

SVM is a kind of classifier has the potential to classify each linear and nonlinear data[34]. The core plan behind the SVM classifier is that it first non-linearly maps the initial training data into sufficiently higher dimension let be *n*, so the data within the higher dimension is separated simply by *n-1* dimension decision surface known as *hyperplanes*. Out of all hyperplanes, the SVM classifier determines the simplest hyperplane that has most margins from the *support vectors*. Thanks to non-linearity mapping, SVM classifier works expeditiously on an oversized data set and has been with success applied in text classification [35].

8 ARTIFICIAL NEURAL NETWORK (ANN)

ANN is a reasonably a data processing nonlinear model cherish the structure of the brain, and it will learn from the prevailing training data to perform tasks like categorization, prediction or forecast, decision-making, visualization, and others. It consists of a compilation of nodes otherwise known as neurons that are the middle of data processing in ANN. With context to the problem statement, these neurons are organized into three different layers, specifically the input layer, an output layer, and hidden layer. Within the context of text classification, the quantity of words or terms outlines the neuron numbers within the input layer, and therefore the classes (class label) of documents define the number of neurons in the output layer. ANN will have a minimum of one input layer and one output layer; however, it is going to have several hidden layers relying upon the chosen drawback. All links from the input layer to the output layer through hidden layers are appointed with some weights that represent the dependence relation between the nodes. Once the neurons get weighted data, it calculates the weighted sum, and a well-known activation function processes it. The output value from the activation function is fed forward to all the neurons within the input layer to map the proper 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 additional versatile and more potent by employing additional hidden layers. In particular, Perceptron (PPN), Stochastic Gradient Descent (SGD) neural network, and Back-propagation neural network (BPN) are the three widespread neural network primarily based classifiers that extensively used for text classification.

9 ROCCHIO CLASSIFIER(RC)

Rocchio classification algorithm [36] is outlined on the conception of relevance feedback theory established within the field of Information Retrieval (IR). It uses the properties of centroid and similarity measure computations among the documents within the training and testing phase of model construction and usage, respectively. If $D=<d_1,d2,\ldots,d_n>$ represents Document set which holds all the training documents and If $C=<c_1,c_2,\ldots,c_m>$ represents class set which have all the distinct class labels. For each class $c_i \in C$, D_{c_i} represent all the documents of D *the set* which belong to class c_i and $\vec{v_d}$ represents the VSM document representation for each document. In the training phase, the Rocchio classifier computes the centroid $\vec{\mu}(c_i)$ for each class from the relevant documents and establishes the centroid of each class as its representative Rocchio classifier computes the centroid $\vec{\mu}(c_i)$ for the class c_i using the equation.

$$\vec{\mu}(c_i) = \frac{1}{|D_{c_i}|} \sum_{d \in D_{c_i}} \vec{v_d} \qquad (3)$$

In testing phase to predict the category label $c_i \in C$ of an untagged test document $d \notin D$, Rocchio classifier calculates its Euclidean distance from the centroid of every class $\vec{\mu}(c_i)$ and assigns that class label which has a minimum distance from untagged test document using the following equation.

$$dist = \arg_c \min \left\| \vec{\mu}(c_i) - \vec{v_d} \right\| \qquad (4)$$

10 RIDGE CLASSIFIER(RIDGE)

The Ridge classification algorithm [37]relies on subspace assumption, which states that samples of a specific class lie on a linear subspace and a new test sample to a category will be described as a linear combination of training samples of the relevant class. The ridge classification algorithm is presented in Fig.X.2.

Input: Data matrix X holds training dataset and data matrix X-test holds the test dataset
Procedure:

Step-1: For each test data x ∈ X-test, calculate the regression parameter vector $\hat{\alpha}$ as

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

where λ the regularization parameter and irepresent each class.

Step-2: Perform projection of new test sample x onto the subspace of each class i using $\hat{\alpha}_i$ as follows

$$\vec{x}_i = X_i \hat{\alpha}_i$$

Step-3: Calculated the distance between the test sample x and the class-specific sub-space $\vec{x}_i$.

Step-4: The test sample x is assigned to that class whose distance is minimum.

FIGURE X.2Ridge Classification Algorithm

PASSIVE AGGRESSIVE CLASSIFIER (PA)

The passive-aggressive classifiers belong to the family of a large-scale learning algorithm [38]. 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*. Fig. X.3 shows the pseudo-code description of the Passive aggressive classifier.

Algorithm: Passive-Aggressive (PA) classifier for multi-class classification
Input:
- D=<X, Y> is the dataset where X holds training instances and Y holds class labels
- Cost function $\rho(y, \overline{y})$

Initialize: Weight vector $w_1 = (0, \ldots, 0)$

1. **for** $i = 1, 2, \ldots$
2. Consider, instance $x_i \in X$ and its corresponding label $y_i \in Y$
3. **if** PA method == prediction-based(PB)
4. return $\overline{y}_i = \arg\max_{y \in Y} (w_i . \phi(x_i, y))$
5. **if** PA method==Max-loss(ML)
6. return $\overline{y} = \arg\max_{r \in Y} (w_i . \phi(x_i, r) - w_i . \phi(x_i, y_i) + \sqrt{\rho(y_i, r)})$
7. Loss function: $\ell_i = w_i . \phi(x_i, \overline{y}_i) - w_i . \phi(x_i, \overline{y}_i) + \sqrt{\rho(y_i, \overline{y}_i)}$
8. Compute: $\tau_i = \frac{\ell_i}{\|\phi(x_i, y_i) - \phi(x_i, \overline{y}_i)\|^2}$
9. Update weight $w_{i+1} = w_i + \tau_i(\phi(x_i, y_i) - \phi(x_i, \overline{y}_i))$

FIGURE X.3Pseudo Code for Passive-aggressive Classifier.

11 RANDOM FOREST(RF)

Random forest classifier is a *bagging* type ensemble-learning algorithm. Fig. X.4 shows the overall architecture of the random forest classifier. 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 that it eliminates the problem of over-fitting and increases the classification accuracy.

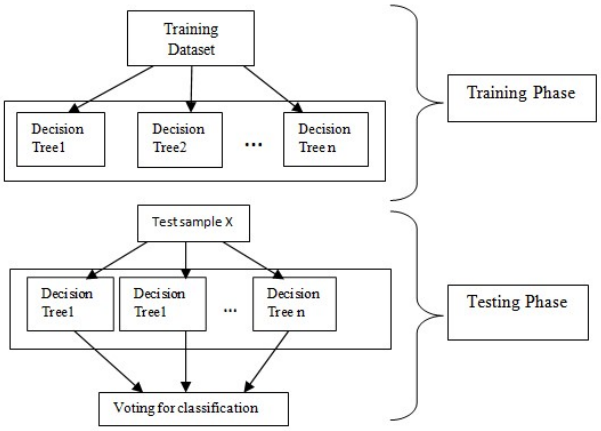

FIGURE X.4Random Forest classification.

12 RESULTS AND DISCUSSION

12.1 *EXPERIMENTAL SETUP*

The experimentation is performed on a machine having Intel(R) Pentium(R) CPU 3825U processor 1.90 GHz with 4.00 GB of RAM. All the coding implemented using python programming language on Jupiter notebook of Google Colaboratory.

Dataset:The experimentation has performed on ***PubMed Abstract*** dataset, which iscreated by scraping a large number of research articles from the National Library of Medicine.The PubMed Abstract dataset contains 45399 article abstracts related to different areas of research such as Deep learning, COVID-19, Human Connectome, Virtual reality, Brain-machine interfaces, Electroactive polymers, PEDOT electrodes, and Neuroprosthetics. The dataset is freely available on the Kaggle website at https://www.kaggle.com/bonhart/pubmed-abstracts. In this experiment, COVID-19 documents in one category and Human Connectome, Brain-machine interfaces, Electroactive polymers, PEDOT electrodes, and Neuroprosthetics in another category. The summary of these datasets are presented in Table X.1, and their descriptions are detailed below:

TABLE X. 1Summary of PubMed Abstract dataset

Document type	No. of documents	category	Class Label
COVID-19	8954	COVID	0
Human Connectome	4877	Non-COVID	1
Brain-Machine Interfaces	4377	Non-COVID	1
Electroactive Polymers	907	Non-COVID	1
PEDOT electrodes	206	Non-COVID	1
Neuroprosthetics	715	Non-COVID	1

13 PERFORMANCE MEASURE

The classification performance measures [39] like*Accuracy*, *Precision*, *Recall*, and *F1-Score* are used to evaluate the performance of the classification algorithms [40]. The properties of the *confusion matrix*such as True Positive (tp_i), True Negative (tn_i), False Positive (fp_i), and False Negative (fn_i) defines aforementioned performance metrics.The *confusion matrix* has shown in Table X.2.

TABLE X.2.Confusion Matrix for class C_i

Total Documents		***Predicted Class***	
		C_i	***Not C_i***
Actual Class	***C_i***	*True Positive (TP)*	*False Negative (FN)*
	Not C_i	*False Positive (FP)*	*True Negative (TN)*

a. True Positive (tp_i): The documents, which belong to class C_i, are 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.

The performance measures explained below.

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

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

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

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

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

$$\frac{\sum_{i=1}^{n} \frac{tp_i}{tp_i + fp_i}}{n} \quad (9)$$

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

$$\frac{\sum_{i=1}^{n} \frac{tp_i}{tp_i + fn_i}}{n} \quad (10)$$

- F1-Score:

$$\frac{2(\mathrm{Precision} \times \mathrm{Recall})}{\mathrm{Precision} + \mathrm{Recall}} \quad (11)$$

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

14 HYPER-PARAMETERS FOR DIFFERENT CLASSIFIERS

The initialization of the input parameters among the different classifiers has a great impact on the classification performance measurements. TableX.3highlights the respective parameter setting procedures adapted in the experimental process of building the corresponding classifier.

TABLE X.3 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="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="1000"	activation function=relu

15 PERFORMANCE ANALYSIS

This section elaborates the implementation of ML algorithms on PubMed Abstract dataset and investigates the outcomes of the ML algorithmsto comparetheir performance. All the ML methods have implemented in the Scikit-learn ML library. The text preprocessing methods like tokenization, stop word removal, stemming and lemmatization has been performed by the *TfidfVectorizer*of Python Scikit-learn library, and it finally transforms all the documents to *TF-IDF* [41] based *vector space model (VSM)*[42] document representation. In this experiment, the TF-IDF based VSM representation generates 24365 number of features forPubMed Abstract dataset. Once the features of the documents present in *TF-IDF* based *VSM document representation*, subsequently, training of all the ML algorithms performs with *10-fold cross-validation*.In *10-fold cross-validation*, the model training of ML algorithms performs in ten iterations. In every iteration, all the documents of the dataset equally divided into ten parts and each part select documents randomly from the whole dataset. Out of ten parts, nine parts usually consider for training, and one part uses for testing. After ten iterations, the mean and standard deviation of all the performance measures are evaluated. All the ML algorithms used the default hyper-parameters defined by *Scikit-learn* ML library presented in table X.3 Theclassification performance with mean and deviation has shown below in *Table 4*. The classification accuracy of the ML algorithms has been compared and presented graphically in *Figure 2.*

Table 4. Performance of ML algorithms for PubMed Abstract dataset

Classification Algorithm	Performance Measure(Mean±Deviation)			
	Accuracy	Precision	Recall	F1-Score
KNN	0.9745±0.0134	0.9756±0.0119	0.9745±0.0134	0.9745±0.0134
DT	0.9924±0.0024	0.9924±0.0024	0.9924±0.0024	0.9924±0.0024
MNB	0.9944±0.0027	0.9944±0.0027	0.9944±0.0027	0.9944±0.0027
BNB	0.9721±0.0074	0.9733±0.0069	0.9721±0.0074	0.9721±0.0074
RF	0.9950±0.0016	0.9951±0.0015	0.9950±0.0016	0.9950±0.0016
SVM	0.9953±0.0017	0.9953±0.0017	0.9953±0.0017	0.9953±0.0017
PPN	0.9946±0.0021	0.9946±0.0021	0.9946±0.0021	0.9946±0.0021
SGD	0.9933±0.0021	0.9933±0.0021	0.9933±0.0021	0.9932±0.0021
Ridge	0.9963±0.0015	0.9963±0.0015	0.9963±0.0015	0.9963±0.0015
RC	0.9826±0.0030	0.9828±0.0029	0.9826±0.0030	0.9826±0.0031
PA	0.9965±0.0009	0.9965±0.0009	0.9965±0.0009	0.9964±0.0009
BPN	0.9957±0.0013	0.9957±0.0013	0.9957±0.0013	0.9957±0.0013

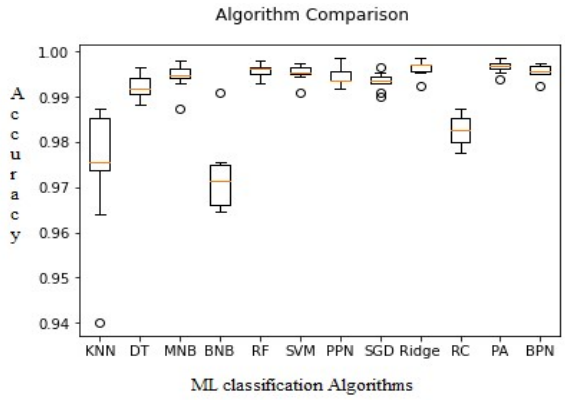

Figure 5:Box plot for Algorithm comparison on TREC dataset

From *Table 4*, it is clear that for PubMed Abstract dataset, the PA classifier performs best among all the classifiers with respect to all the classification performance measures. The classification accuracy of the PAclassifiers is 0.9965±0.0009. The Ridge classifier performs well next to PA classifier. After Ridge, BPN and SVM classifiers perform well. However, KNN and BNB classifiers yield the lowest classification performance among all the classifiers for the dataset. Meanwhile, the remaining classifiers provide an average classification performance. Figure 2 shows the statistical comparison among the classifiers with the help of boxplots.

16 CONCLUSION AND FUTURE SCOPE

Since the outbreak of COVID-19 pandemic, a vast number of research articles have been published to tackle the disease. Therefore the time demandsto classify COVID-19 related articles from other large volumes of research articles. Hence, this research paper summarizes in detail the procedures involved automatic document classification process for COVID-19 related articles,exemplifies the working logic of the supervised ML algorithms and empirically evaluates how all the ML algorithmswhich are constituted to act as a classifier to the benchmark PubMed Abstract dataset. Notable, for COVID-19 text document classification, classification algorithms like PA, Ridge BPN and SVM outperforms among all the classification algorithms. However, the performance of KNN and BNB classifiers has shown poor results for the chosen dataset compared to otherclassifiers.Meanwhile, other classifiers have an average classification performance. The future scope is to perform COVID-19 document classification using Machine learning and Deep learning classifiers for the large-scale dataset.

REFERENCES

[1] B. Behera, and G. Kumaravelan. "Towards the Deployment of Machine Learning Solutions for Document Classification", International Journal of Computer Sciences and Engineering, Vol.7(3), Mar 2019, E-ISSN: 2347-2693.

[2] Webster, J.J.; Kit, C. Tokenization as the initial phase in NLP. In Proceedings of the 14th conference on Computational linguistics. Association for Computational Linguistics.2010, 4, 1106–1110.

[3] Saif, H.; Fernández, M.; He, Y.; Alani, H. On stopwords, filtering and data sparsity for sentiment analysis of twitter. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014), Reykjavik, Iceland, 26–31 May 2014.

[4] Silva, C.; Ribeiro, B. The importance of stop word removal on recall values in text categorization. In Proceedings of the International Joint Conference on Neural Networks. Portland, OR,USA.2003,3,1661–1666.

[5] Lovins, J.B. Development of a stemming algorithm. MIT Information Processing Group, Electronic Systems Laboratory.1968

[6] Porter, M.F. An algorithm for suffix stripping. Program: electronic library and information systems.1980, 14, 3, 130–137.

[7] Hull, D.A. Stemming algorithms: A case study for detailed evaluation. JASIS 47.1996, 1, 70–84

[8] Liu, H.; Motoda H. Feature Extraction, construction, and selection: A Data Mining Perpective. Boston, Massachusetts(MA): Kluwer Academic Publishers,1998.

[9] Yang Y.; Pederson, J.O. A comparative study on feature selection in text categorization, ACM SIGIR Conference, 1995.

[10] Deerwester, S.; Dumais, S.; Landauer, T.; Furnas, G.; Harshman, R. Indexing by Latent Semantic Analysis. JASIS.1990, 41(6), 391–407.

[11] Hofmann, T. Probabilistic latent semantic indexing. ACM SIGIR Conference, 1999.

[12] Fisher, R. The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics.1936, 7, 179–188.

[13] Chakrabarti, S.; Roy, S.; Soundalgekar, M. Fast and Accurate Text Classification via Multiple Linear Discriminant Projections, VLDB Journal. 2003, 12(2), 172–185.

[14] Howland, P.; Jeon, M.; Park, H. Structure Preserving Dimension Reduction for Clustered Text Data based on the Generalized Singular Value Decomposition. SIAM Journal of Matrix Analysis and Applications.2003, 25(1), 165–179.

[15] Howland, P.; Park, H. Generalizing discriminant analysis using the generalized singular value decomposition, IEEE Transactions on Pattern Analysis and Machine Intelligence.2004, 26(8), 995–1006.

[16] Salton, G.; Wong, A.; Yang, C.S. A vector space model for automatic indexing. Commun. ACM 18.1975, 11, 613–620.

[17] Manning, C.D.; Raghavan, P.; Schütze, H. Introduction to information retrieval. Cambridge university press Cambridge.2008,1.

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

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

[20] Cohen, AM. An effective general purpose approach for automated biomedical document classification. AMIA Annu Symp Proc. 2006,161–165.

[21] Almeida, H.; Meurs, M. J.; Kosseim, L.; Butler, G.; Tsang,A. Machine learning for biomedical literature triage, Plos One. 2014, 9(12).

[22] Samal, B.R.; Behera, A.K.; Panda, M. 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.2017,128-133.

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

[24] Jiang,X.; Ringwald,M.; Blake,J.; Shatkay,H. Effective biomedical document classification for identifying publications relevant to the mouse Gene Expression Database (GXD).2017.

[25] T.T. Nguyen. "Artificial intelligence in the battle against coronavirus (COVID-19): a survey and future research directions." *Preprint, DOI* 10 (2020).

[26] L. Li et al., "Characterizing the Propagation of Situational Information in Social Media During COVID-19 Epidemic: A Case Study on Weibo," in IEEE Transactions on Computational Social Systems, vol. 7, no. 2, pp. 556-562, April 2020, doi: 10.1109/TCSS.2020.2980007.

[27] J. Samuel, G.G. Ali, M. Rahman, E. Esawi and Y. Samuel,"Covid-19 public sentiment insights and machine learning for tweets classification." Nawaz and Rahman, Md. Mokhlesur and Esawi, Ek and Samuel, Yana, COVID-19 Public Sentiment Insights and Machine Learning for Tweets Classification (April 19, 2020) (2020).

[28] Li, Y. ; Jain, A. Classification of text documents. The Computer Journal.1998, 41(8), 537–546.

[29] Aggarwal, C.C. ; Zhai, C. X. Mining text data, Springer. 2012.

[30] McCallum, A.; Nigam,K. A comparison of event models for naive bayes text classification. In AAAI-98 workshop on learning for text categorization.1998, 752, 41–48.

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

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

[33] Han, E.S.; Karypis, G.; Kumar, V. Text categorization using weight adjusted k-nearest neighbor classification. Springer.2001.

[34] Cortes, C. ; Vapnik, V.; Support-vector networks. Machine Learning. 1995, 20, 273–297.

[35] Drucker, H.; Wu, D.; Vapnik, V. Support Vector Machines for Spam Categorization, IEEE Transactions on Neural Networks. 1999, 10(5), 1048–1054.

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

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

[38] Crammer, K.; Dekel, O.; Keshet, J.; Shalev-Shwartz, S.; Singer, Y. Online passive aggressive algorithms, Journal of Machine Learning Research. 2006, 7,551–585.

[39] Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks, Inform. Process. Manage. 2009, 45(4), 427-437.

[40] B. Behera, G. Kumaravelan and P. Kumar.B, "Performance Evaluation of Deep Learning Algorithms in Biomedical Document Classification," 2019 11th International Conference on Advanced Computing (ICoAC), Chennai, India, 2019, pp. 220-224, doi: 10.1109/ICoAC48765.2019.246843.

[41] S. Robertson, “Understanding inverse document frequency: on theoretical arguments for IDF”, Journal of documentation.2004.

[15] G. Salton, A. Wong and C.S. Yang,“A vector space model for automatic indexing. Communications of the ACM 18, no. 11(1975): 613-620.