Augmented Human Research (2020)5:12
https://doi.org/10.1007/s41133-020-00032-0

ORIGINAL PAPER

A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification

Kanish Shah[1] · Henil Patel[1] · Devanshi Sanghvi[1] · Manan Shah[2]

Received: 27 August 2019 / Revised: 6 February 2020 / Accepted: 13 February 2020

Abstract
In the current generation, a huge amount of textual documents are generated and there is an urgent need to organize them in a proper structure so that classification can be performed and categories can be properly defined. The key technology for gaining the insights into a text information and organizing that information is known as text classification. The classes are then classified by determining the text types of the content. Based on different machine learning algorithms used in the current paper, the system of text classification is divided into four sections namely text pre-treatment, text representation, implementation of the classifier and classification. In this paper, a BBC news text classification system is designed. In the classifier implementation section, the authors separately chose and compared logistic regression, random forest and K-nearest neighbour as our classification algorithms. Then, these classifiers were tested, analysed and compared with each other and finally got a conclusion. The experimental conclusion shows that BBC news text classification model gets satisfying results on the basis of algorithms tested on the data set. The authors decided to show the comparison based on five parameters namely precision, accuracy, *F*1-score, support and confusion matrix. The classifier which gets the highest among all these parameters is termed as the best machine learning algorithm for the BBC news data set.

Keywords Text classification · Machine learning algorithm · Logistic regression · Random forest · K-nearest neighbour · Natural language processing · Feature extraction

Introduction

In this digital world, the technology has shown its dominance as the humans have pushed their limit to think by binding artificial brain with human one [19, 35]. By performing this process, a new field has come in existence named Artificial Intelligence. AI is termed as the development of such systems which can perform the work which humans generally do, like distinction between two different objects, recognizing different voices and many more. It comes under the Computer Science branch and has gained so much importance and popularity worldwide that it is applicable to any working domains in this world. AI is something that works on past learning and helps in predicting the future values by giving the accuracy of the algorithm applied on a data set [1, 36–38]. There are many domains of AI like machine learning, deep learning, ANN (artificial neural network), CNN (convolutional neural network) which help in developing a more advance technology [23, 47].

Machine learning has gained so much importance and value in the recent years. There has been a recent resurrection in this field as implementers are now able to provide more transparency in the algorithms. ML, being one of the domains of AI, has made many things possible which once were thought as a tedious task. That's why this domain is so potent and influential. ML is completely based on mathematics and statistics. It is an approach to build intelligent systems. Similar to AI, it helps in the prediction of future by analysing past data and experience. There are many applications of ML like object differentiation and classification, speech recognition, text

✉ Manan Shah
manan.shah@spt.pdpu.ac.in

[1] Department of Computer Engineering, Indus University, Ahmedabad, Gujarat, India

[2] Department of Chemical Engineering, School of Technology, Pandit Deendayal Petroleum University, Gandhinagar, Gujarat, India

Published online: 05 March 2020

classification, prediction of weather, checking of destroyed crops, face recognition, medical diagnostics, etc. [14, 18, 25]. ML works purely on data; thus, Big Data has played a vital role in its evolution [37, 38, 43].

Another important concept which has played a vital role in the classification of text is natural language processing (NLP). In today's data generating generation, the data are generated in zettabytes and approximately around 80% of data generated is in unstructured form. So, it is necessary to convert data into structured form in order to derive its meaning. Here, NLP and text mining come into light. Moreover, there are various human languages in this world and each and every person writes the text in their own languages like Persian, Turkish, Chinese, English and many more. But it is the task of the computer to perform analysis and derive the meaning of those texts. NLP helps the computers to bring out the useful meaning in a smarter way. This algorithm has gained importance in the recent years. NLP algorithm is somewhat dependent on machine learning algorithms for learning the rules [44]. NLP is used in text classification, extraction and tracing of information, tagging of speech, opinion mining and lot more [13].

Text classification is similar to mapping used in mathematics and has the mathematical form as:

$$f : X \rightarrow Y$$

where X is the different sets of text to be classified and Y being the set of categories [32].

Text classification in recent years has made continuous success, and the application of this technology in various applications like emotion classification from text, classification of spam and many other has become immense and boundless [11, 26, 31]. The current paper talks about the text classification which is defined as the process of labelling texts with relevant categories from a data set that is already predefined. By classifying the content into different categories help users to search something very easily. After grouping the text, it is then analysed by different models which has the task of applying tags to the content. These models are nothing but the machine learning algorithms and are also known as classifiers. These classifiers need to be trained for making predictions on the textual data set. These classifiers are trained by assigning the tags and then making associations on the pieces of text. The authors have used three classifiers in this paper for text classification namely logistic regression, random forest and K-nearest neighbour. Logistic regression is used to measure the statistical significance of each independent variable with respect to probability. Random forest works on decision trees which are used to classify new object from input vector. K-nearest neighbour classifier helps in maintaining groups by keeping similar things together. These algorithms are then compared to find out which, among the three, is the best classifier. The contribution of this manuscript was that a higher accuracy and precision were obtained for the algorithms but there was still a room for improvement. The algorithms were successful in classifying the textual data and gave much better output than expected. Therefore, these algorithms can be used with better optimization than the current one.

Related Works

Related Study on Logistic Regression

Prabhat and Khullar [39] have presented that the enormous amount of data stored and flowing online cannot be mined effectively to extract valuable information and decision cannot be taken based on extracted information. Sentiment analysis is a method in which we judge people's ideas, opinions, feelings, attitude, thought and belief about a specific concept. Authors presented sentiment classification on Big Data using Naive Bayes classifier and logistic regression. Authors have used supervised and unsupervised learning algorithms. The performance of algorithms has been evaluated on the basis of different parameters like accuracy, precision and throughput. The analysis with logistic regression gives 10.1% more accuracy and 4.34% more precise results with almost one-fifth implementation time for same size of data set compared to Naive Bayes classifier.

Yen et al. [52] have implemented logistic regression model for Chinese text categorization. Instead of tokenizing the words which is a common method in text categorization, the authors have presented a new method which uses N-gram-based language model. This model takes word relations into account for Chinese text categorization without Chinese word tokenizer. To prevent from out-of-vocabulary, we also propose a novel smoothing approach based on logistic regression to improve accuracy. They used the logistic regression to smooth the probability of n-gram. They proposed a novel feature selection method which is suitable to N-gram-based model. Secondly, they proved that it could improve the F-measure in most case.

Liu et al. [30] put upfront that in the recent years multi-label classification has gained so much popularity, but still there is a need to tackle the ubiquitous data. So, the authors have presented novel framework for multi-label learning which can achieve the purposes of classification learning and selection of variables simultaneously. Therefore, they have used logistic regression to train the models for multi-label data for classification. They also solved the convex optimization problem of logistic regression with the elastic net penalty by a quadratic approximation technique for better performance. The results were improved

performance and better accuracy, and also the model is competitive among the other six models used.

Aseervatham et al. [4] have talked about how logistic regression is used for tackling the text categorization problems. The authors have stated that the ridge logistic regression has the performance as that of the support vector machine (SVM) algorithm. However, the advantage of using logistic regression is that of computing the probability value rather than calculating a score. They have presented a new selection method which approaches the ridge solution by a sparse solution. This selection method first computes the ridge solution and then performs the feature selection. The final result was that this method gave a solution that is good tradeoffs between ridge and LASSO solutions.

Related Study on Random Forest

Elghazel et al. [12] have talked about ensemble multi-label text categorization using rotation forest and latent semantic indexing. The authors have proposed a method based on four ideas. (1) to perform latent semantic indexing on lower dimensional space of concepts, (2) splitting the vocabulary randomly, (3) bootstrapping of document and (4) use the BoosTexter as a powerful multi-label base learner for text categorization for improving accuracy. Accuracy is promoted through underlying semantic structure in the text. The combination of latent semantic indexing and rotation forest brings about improvements in average precision, ranking, loss and error compared to five other state-of-the-art approaches across textual data.

Al Amrani et al. [2] put upfront that in the research area sentiment analysis has become more popular. This method allocates positive and negative polarity to the items with the help of natural language tools and also helps in predicting high or low performance of the classifiers that are being implemented. Here, random forest and support vector machines (SVM) are implemented. The authors have focused on the sentiment analysis that is resulted from the product reviews by using the original techniques for searching of text. The reviews are then been classified into positive and negative category based on a relation to a query. The result outperformed the classifiers that were used in the Amazon data set.

Salles et al. [42] have solved the problem of high-dimension noisy data by using random forest algorithm. The authors have presented the lazy version of traditional random forest classifier also named as Lazy NN_RF. This model is specially designed for high dimension classification tasks. The training projection is comprised of examples that resemble the examples to be classified and obtained through nearest neighbourhood training projection. The main contributions include: (1) implementation of the random forest classifier and nearest neighbourhood projection of training set and (2) a detailed experimental analysis. This showed that their approach is very effective and feasible and can be considered for text classification.

Nadi and Moradi [34] put upfront that random forest is one of the most powerful ensemble method with high performance when it comes to high-dimension data. The authors in this paper have proposed a novel approach to increase the performance of random forest by increasing the number of trees and decreasing the number of levels for each tree in random forest. In this approach, the trees are bounded to certain depth for allowing increase in views. Each tree that is being bounded is considered as a local view of problem and more the local view better is the classification. The results showed that by binding the trees the accuracy can be improved for high-dimension problems.

Wu et al. [50] have used random forest for imbalanced text categorization. The authors have presented a new random forest ensemble method named ForesTexter. This algorithm uses a simple random sampling of features in building their decision trees. The main idea is to stratify the features into two different groups and generate term weighting for the features. One of the groups contains positive features for minority class, whereas the other group contains negative features for the majority class. Therefore, the tree model becomes more robust for text categorization task with imbalanced data set. The result showed that the proposed method is competitive against standard random forest classifier and SVM algorithms.

Related Study on KNN algorithm

Moldagulova and Sulaiman [33] put upfront that the recent years have arisen the need of generation and structuring of textual documents. It has become one of the vital things which seek the attention to resolve the problems of computational resources and bring forth the optimal result. Authors have used method for handling unstructured text data, especially document classification problems. Authors have performed document classification using KNN algorithm by term vector space reduction. The purpose is to justify the choice of the algorithm term vector space model and enhance the KNN text classifier algorithm for implementation of document classification tasks. It was concluded that while using K-NN classifier, even by shortening the size of feature space by factor of 10, there were minimal changes in the accuracy whereas time cost decreased rapidly, almost exponentially.

Hmeidi et al. [17] have implemented classifiers for Arabic articles text categorization. These authors have done a comparative study by comparing two different classifiers/machine learning models for categorization of

Arabic text. The data set was divided into training set which contained news articles and the same with test set. Considering the full word features they used the tf-idf Vectorizer as the weighting method for the selection of features. CHI statistics was used as a ranking metrics. The result of the experiment showed that both the method used had an excellent performance on the data set, while the SVM performed well on average, *F*1-score and the time of prediction.

Tan [46] has implemented K-nearest neighbour classifier for the effective refinement strategy. It is being implemented due to its simplicity and better efficiency. However, KNN suffers from a problem of model misfitting due to its assumptions. Due to this the author has proposed a new refinement strategy named as DragPushing for the K-nearest neighbour classifier to solve this problem. The outcome was that experiment showed that DragPushing achieved an improvement in the performance of the K-nearest neighbour classifier.

Trstenjak et al. [48] have carried out somewhat similar work as that of Ismail Hmeidi. The authors of this paper also used K-nearest neighbour classifier for the text classification. They focused on the possibility of using TF-IDF Vectorizer method and framework for the classification of text in KNN classifier. This framework enabled classification according to various parameters and analysis of results. Speed and quality of classification were considered the judgment parameters. The outcome was that the experiment classified the good and the bad features of the algorithm.

Jiang et al. [21] also improved the K-nearest neighbour for the categorization of text. The authors have proposed a new method by combining constrained one pass clustering algorithm with that of KNN classifier. The result of the experiments showed that this method reduces the text similarity computation substantially and outperforms many other models like Naïve Bayes, support vector machines (SVM) and state-of-the-art K-nearest neighbour classifier.

Other Related Algorithms for Text Classification

Wahiba and Ahmed [49] have evaluated text documents using fuzzy decision trees. This paper talks about a heuristic approach that has not been tested till now, and the proposed method will get it tested. The learning space is developed, and the weight of the attribute is calculated using the TF-IDF Vectorizer method. Then, fuzzy rule is applied and the proposed method has preprocessing phase, membership degrees calculation, fuzzy decision tree induction and classification. The result was that the proposed model gave better classification results but the maximum value of *F*1-score did not exceed 48% which is one of the drawback of the fuzzy systems but can be effective in classification or when compared with fuzzy ID3 algorithm.

Aydogan and Karci [5] have calculated the improvement in accuracy for classification of texts that were in Turkish language. They had used deep neural networks as the algorithm for getting the result. Word2Vec method was used for training word vectors on the two different Turkish data sets. The methods employed here were deep neural networks, convolutional neural networks, long short term memory, gated recurrent unit and bidirectional LSTM. The propped model contains variety of stages like word embedding, continuous bag of words model, skip gram model and skip gram model. The outcome was that the values of accuracy, precision, recall and *F*1-score achieved different values for different models, but the highest was of the gated recurrent unit. However, none of the values exceeded more than 0.85 for any model.

Zhu et al. [54] had discussed about a problem of class discrimination for improving the performance of text classification. Naïve Bayes algorithm has been applied for the experiment. Two models namely multi-variate Bernoulli model and multinomial models are used here. For the extraction of features, mutual information and document frequency thresholding are employed here. Then, comes the Gaussian divergence which helps in measuring the dissimilarity between two classes. The main aim is to get better and accurate ranking order. After this step, selection of features takes place. The result showed that the discrimination-based feature selection can be effective in enhancing discriminative power in text classification.

Raychaudhuri et al. [41] had done a comparative study on three algorithms namely neural networks, decision trees and support vector machines for the purpose of text classification. The data set consisted of 435 instances, 335 samples of training data, 50 samples of test data and 50 samples of validation data. In terms of efficiency between SVM and neural networks, it turned out that SVM performed better than neural networks when the value of C was 1 whereas neural network was better when the value of C was 1000. Fully grown decision tree performed better than smaller decision tree. Therefore, the classification techniques depend on many factors.

Garla et al. [15] used Laplacian and linear SVMs to work on the clinical text classification. The main target was to evaluate the actual effect of unlabelled training dada on Laplacian SVM. The data set consisted of 820 abdominal CT, MRI and ultrasound reports, and the samples used by Laplacian SVM were around 19,845. The main aim of using semi-supervised algorithm is that they can sometimes perform better than supervised algorithms. On the training data, the feature extraction takes place and the computing graph for Laplacian is constructed. Then, kernel parameters are used, SVM optimization technique is applied, and

regularization is used. The final result was that the Macro $F1$-scores were around 0.77 and sensitivity was around 091.

Jiang et al. [20] have presented a technique that improves the performance of Naïve Bayes text classification. The problem of skewed distribution of training data that was causing poor results was solved with this technique. Since the shorts data consisting of text consist of more important words required for classification, while the long data consist of many unnecessary words which decrease the probability. Therefore, this method transforms to estimate the probability of the feature that can be calculated both in a global as well as local view. Different graphs were constructed for the calculation of $F1$-score versus the alpha parameter and achieved different values for different parameters.

Comparative Analysis of the Past Study

From the above different literature reviews that were done on different algorithms like Naive Bayes, linear and Laplacian support vector machine, neural networks like convolutional neural network, recurrent network etc., decision trees and fuzzy logic algorithms, it was observed that almost all the algorithms did not perform up to the mark. Talking about the support vector machine, it achieved great accuracy that did not went above 90% and there has to be many parameters that need to set for performing the classification algorithms and one parameter that might give good accuracy in one might perform poor in other. In neural networks, there is always a problem of overfitting, increasing the load due to more hidden layers and the empirical nature of its model too. Talking about Naïve Bayes classifier, there is always a chance that the algorithm might lose the accuracy and sometimes a zero probability is assigned to the category which does not have categorical variable. In fuzzy logic, the results are achieved on the basis of assumption so it might not be acceptable everywhere and is not that accurate compared to logistic regression, random forest and many others. The problem with the decision tree is that many a time they are unstable as a small change might result in a drastic change. The inaccuracy is always a problem with decision trees as it performs better when its size is big and poor when smaller in size.

Methodology

There are ample numbers of machine learning algorithms that are used in for the classification of texts. But all do not have the same accuracy or precision, i.e. one might have low accuracy, while other might have higher accuracy than the other. The classification of text is done using three different machine learning algorithms which has been implemented on a data set. The classification algorithms used are logistic regression classifier, random forest classifier and K-nearest neighbours classifier. They have played an important role in evaluating machine learning strategies by giving a much easier and understandable solution. Each of these three algorithms has a complete different working from one another as one works on a particular formula for classifying and predicting, the other works by constructing nodes and trees (random forest). A solution has been developed about the classification of texts by each of these three algorithms.

The overview of architecture is shown in Fig. 1. At the beginning, we import the necessary libraries which can also simultaneously be included in code as we proceed further. Next, we load the data set on which we need to perform the classification. To evaluate the representation, we used the BBC news data set [45]. The data set contains two columns (i) category consisting of five different classes and (ii) text which contains the text lines. Then, we perform the pre-

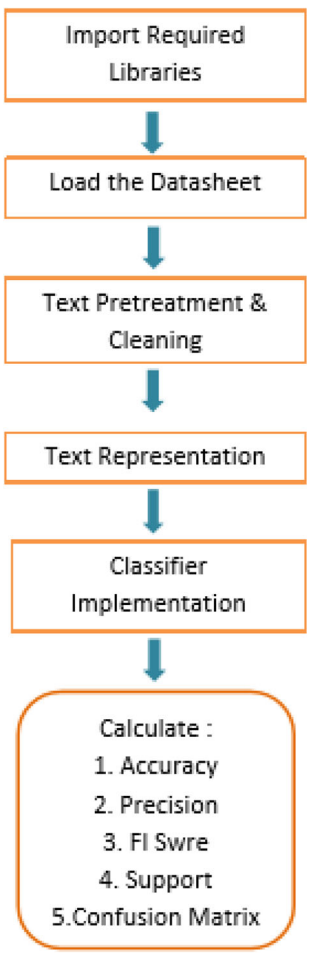

Fig. 1 Architecture of the implementation

treatment or cleaning of text. After this step, the focus comes on the representation of text. In the testing phase, the three classifiers are applied on the data set which gave five parameters as the output and also to compute which feature has the maximum value for a particular class in the data set [40, 51]. The parameters computed are accuracy, precision, *F*1-score, support and confusion matrix.

Text Pre-treatment and Representation

Before this step, the authors decided to plot the five classes of data set on a bar graph to determine what number of text belongs to a particular class and one can get a clear picture about the data set without actually looking it.

The authors obtained this graph after the implementation of code. The graph was plotted with category on *x*-axis versus number of text lines on *y*-axis. For instance, the number of text in business category is above 500, entertainment close to 400, politics above 400 and so on. After demonstrating this graph, the steps for pre-processing initiates (Fig. 2).

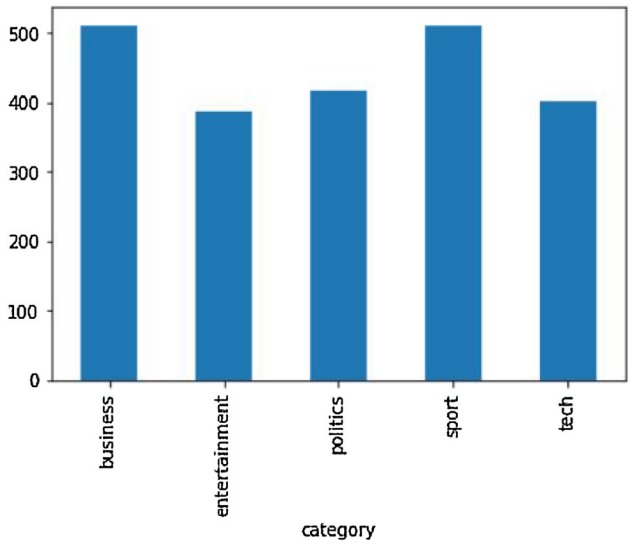

Fig. 2 Bar graph depicting the number of texts lines for different classes

To represent a document for classification, two major aspects play an important role which are text pre-treatment or pre-processing and the other one is term weighting [16]. We shall discuss the steps of text pre-treatment or text cleaning. There are some words which have no part to play in discrimination of classes. Those words can be prepositions, conjunctions and pronouns. These words have no context in the text because they do not contribute in classifying the classes [53]. These words are known as stop words. So, it is necessary to remove these stop words like 'the', 'a', 'and', 'but', 'or', etc. Therefore, the English stop words were downloaded. After downloading the stop word list, we have to check the word against that list and filter the words from the list [24]. After that, we perform stemming on the text which is kind of normalizing the text. A sentence can be written in much different form by changing the tenses but has the same meaning. So, stemmer helps in removing those tenses by bringing the sentences in same meaning. Stemmer's main task is to condense the sentences. The algorithm we have used for stemming is the Porter Stemmer which performs the condensing task. Next, we focus on the representation of text. First, we have applied lambda function and then used the joint operation on the text that we had obtained after stemming. After that, we had used the sub-operation for checking whether all the text is in the small and capital alphabets or not. After checking that, we convert all the text into lower letters in order to unify all the text so that the classification becomes easier. After pre-processing and cleaning, we obtained the output of cleaned text as shown in Table 1.

In Table 1, the first column shows name of categories that we used for implementation. The next column displays the text before performing pre-processing on it. The last column displays how the text has become cleaned by removing irrelevant (not useful for text classification) words from the text, for instance removing words like of, in, the, etc.

After cleaning, it is very important to represent the text in which machine can easily understand. Therefore, Vectorizer has been used which converts the sentence or text into an array or vector of numbers. TF-IDF (term frequency-inverse document frequency) Vectorizer is

Table 1 Text before and after cleaning

Sr. no.	Category	Text	Cleaned
1	Tech	Tv future in the hands of viewers with home th…	Tv futur hand viewer home theatr system plasma…
2	Business	Worldcom boss left books alone former worldc…	Worldcom boss left book alon former worldcom b…
3	Sport	Tigers wary of farrell gamble Leicester say…	Tiger wary farrel gamble leicest say rush make…
4	Sport	Yeadling face newcastle in fa cup premiership s…	Yead face Newcastl fa cup premiership side new…
5	Entertainment	Ocean s twelve raids box office ocean s twelve…	Ocean twelv raid box offic ocean twelv crime c…

implemented for transforming the text into a representation of numbers which has certain meaning. The term frequency is used for normalizing the occurrence of each word with size of the data set, whereas the inverse document frequency is used for removing the words which do not contribute much for deciding the meaning of the sentence. So, if the term occurs in the text, it would become zero. The TF-IDF technique eliminates the common words and also extracts the relevant features from the corpus [6]. The TF-IDF algorithm's main focus is on the word that has the higher frequency in the text, and at the same time, it appears in the corpus in a smaller range. Then, this word has the strongest capability to distinguish different classes in the text [32].

The mathematical formula of TF-IDF algorithm is given by:

$$\text{TF-IDF}(w) = \text{TF}(w) * \text{IDF}(w)$$

Authors had passed n-gram as an argument which consists of adjacent letters or words in the text that helps in predicting the next item in a sequence. N-gram captures the structure of a language like what letter or word would follow the previous one. N-gram is to generate word vector based on the context of the text [32]. Moreover, we had used norm function for returning one of different matrix depending on the value Ord parameter. l2 norm is used for minimizing the sum of squares of differences between target value and estimated value. After performing this, we returned all the elements in form of an array. Finally, we applied logistic regression, random forest and K-NN classifiers separately in order to calculate accuracy, precision, $F1$-score and support.

True positives is defined as the predicted values predicted correctly, i.e. the value of predicted class is yes and that to of actual class is also yes.

True negatives is defined as negative values predicted correctly, i.e. the value of predicted class is no and that to of actual class is also no.

False positives is defined as the condition when the predicted class is yes whereas the actual class is no.

False negatives is defined as the condition when the predicted class is no whereas the actual class is yes (Tables 2, 3, 4).

Table 2 Diagram depicting four parameters

	Predicted class		
Actual class	S	Class = Yes	Class = No
	Class = Yes	True positive	False negative
	Class = No	False positive	True negative

Table 3 True positive + false positive = total predicted positive

	Negative	Positive
Negative	True Negative	False Positive
Positive	False Negative	True Positive

Table 4 True positive + false negative = actual positive

	Negative	Positive
Negative	True Negative	False Positive
Positive	False Negative	True Positive

Accuracy is defined as the ratio of observations predicted correctly to the total number of observations. The mathematical formula is defined as

$$\begin{aligned}\text{Accuracy} = {} & \text{True Positives} + \text{True Negatives}/ \\ & (\text{True Positives} + \text{True Negatives} \\ & + \text{False Positives} + \text{False Negatives})\end{aligned}$$

Precision is defined as the ratio of observations that are predicted positively correct to the total number of observations predicted positively. The mathematical formula is defined as

$$\begin{aligned}\text{Precision} = {} & \text{True Positives}/\text{True Positives} \\ & + \text{False Positives}\end{aligned}$$

Recall is defined as the ratio of observations that are predicted positively correct to the total number of observations in an actual class. The mathematical formula is defined as

$$\text{Recall} = \text{True Positives}/\text{True Positives} + \text{False Negatives}$$

$F1$-score is defined as the harmonic average of recall and precision. The mathematical formula is defined as

$$F1 \text{ score} = 2 * (\text{recall} * \text{precision})/(\text{recall} + \text{precision})$$

Support is defined as the total number of samples of true response which lies in a class.

Confusion matrix is defined as how much a classifier is able to predict the correct value, i.e. true positives in

classification or what number of values belongs to the correct class rather than the other class.

Implementation of Classifier

After performing text pre-treatment and representation, the classifiers are now to be implemented. We had considered three classifiers namely logistic regression, random forest and KNN to determine which has the best output. In the beginning, we had split the data set into training and testing set, the size of the testing set being 25% and training set being 75%. After splitting, we had used the pipeline for implementing the classifiers. Pipelining is used for better flow of an algorithm. Pipelines work by enabling sequence of the data that needs to be transformed and to correlate in a model that needs to be tested and evaluated for achieving an outcome. As all the classifiers implemented are supervised so the data sets are provided and one has to just apply classification algorithms. Generally a machine learning pipeline consists of four stages namely pre-processing, learning, evaluation and prediction. There were many reasons behind the use of pipeline. Firstly, pipelining improves the overall functioning of the model. Secondly, it also helps in pre-processing and enhancing of the data, better handling of the over fitting caused by the data set and better tuning with the hyper-parameters of the pipeline. While implementing the classifier with the help of pipeline, the use of pickle has also been included. Pickle is generally used for serializing and de-serializing objects in python. With the help of pickle, the program is stored in the disk so it becomes easy to work at ones ease. It is also helpful in making new predictions without rewriting everything again (Fig. 3).

Random Forest Algorithm

In this algorithm, a large number of decision trees are built as they operate together. Decision trees act as pillars in this algorithm. Random forest is defined as the group of decision trees whose nodes are defined at the pre-processing step [7]. After constructing multiple trees, the best feature is selected from the random subset of features [8, 22]. To generate a decision tree is another concept which is formed using decision tree algorithm. So, random forest consists of these trees which are used to classify new object from input vector. Each decision tree built is used for classification. Suppose we give the tree votes to that class then the random forest chooses the classification that has the most number of votes among all the trees in the forest. There are also some chances of error in the random forest depending upon two parameters:

(i) There are chances that two trees in a forest may have correlation between each other which leads to increase in error rate.
(ii) Each tree has its own strength. So, a tree which has lower error rate is a strong classifier and vice versa.

- Some of the features of random forest are:

(i) Handles huge number of input variables without deletion of variables.
(ii) It suggests the variables which are important in classification.
(iii) The databases that are large also runs efficiently.
(iv) The trees or forests generated can be saved for future use also.

- Steps for random forest algorithm:

Step 1: From the training data, pick K random data points.
Step 2: Construct a decision tree with these K data points.
Step 3: Before repeating steps 1 and 2, choose the number NTree of tree you want to construct.
Step 4: Predict the value of y by making each one of NTree trees for a new data point and assign new data point average across all predicted y values.

The mathematical formula for random forest classifier is:

$$n_{ij} = w_l C_j - w_{\text{left}(j)} C_{\text{left}(j)} - w_{\text{right}(j)} C_{\text{right}(j)}$$

ni sub(j) = the importance of node j
w sub(j) = weighted number of samples reaching node j
C sub(j) = the impurity value of node j
left(j) = child node from left split on node j
right(j) = child node from right split on node j

K-Nearest Neighbour Algorithm

This algorithm focuses on the keeping the similar things nearest to each other. This model works on class labels and

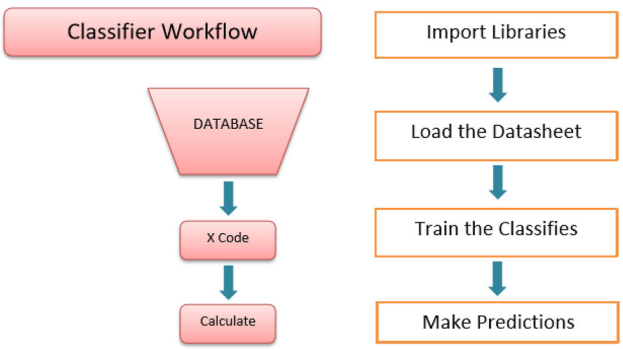

Fig. 3 Workflow of classifiers

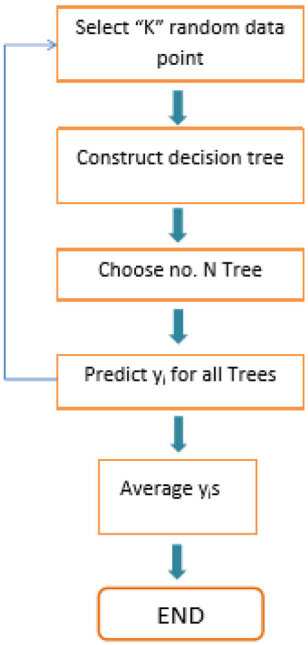

Fig. 4 Flow chart for random forest classifier

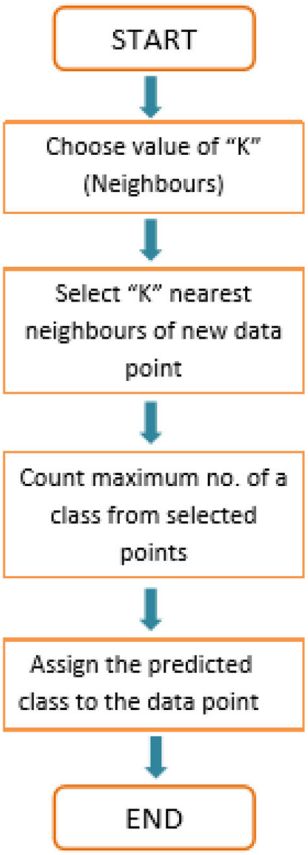

Fig. 5 Flow chart for K-NN classifier

feature vectors in a data set [9]. KNN stores all the cases and helps in classifying new cases with the help of similarity measure. In K-nearest neighbours, text is represented using a spatial vector which is denoted by $S = S(T1, W1; T2, W2; \ldots Tn, Wn)$. For any text, similarity is found and calculated using the training text and the texts with highest similarity are selected. Finally, the classes are determined based on K neighbours (Fig. 4).

- Steps for performing KNN algorithm

  Step 1: At the beginning, choose number K of neighbours.
  Step 2: According to the Euclidean distance, take the K-nearest neighbours of new data point.
  Step 3: In each category, perform the count of number of data points among the K neighbours.
  Step 4: After counting, assign the new data point where you counted the most neighbour (Fig. 5).

- The steps of KNN involved in classification of text are:

  1. Both training text and incoming text are expressed as feature vectors in vector space.
  2. 2) Then, comparison between the feature vector of the incoming text and that of each training text is calculated with a mathematical formula.

  $$\mathrm{sim}(di, dj) = \frac{\sum_{k=1}^{M} WikWjk}{\sqrt{\sum_{k=1}^{M} Wik^2}\sqrt{\sum_{k=1}^{M} Wjk^2}}$$

  where di and dj are the feature vectors of the incoming and training text, respectively, M being the dimension of feature vector. Wik and Wjk are the k-th elements of vectors di and dj, respectively.
  3. Lastly, the K-nearest neighbours of that incoming text are selected based on the similarity or comparison of texts [47].

  $$\mathrm{sim}\ (d_i, d_j)\ \delta\,(d_i, C_m)$$

The formulas behind the KNN are

$$Q(di, Cm) = \sum_{j=1}^{K} \mathrm{sim}(di, dj)\delta(di, Cm)$$

$$\delta(di, Cm) = \begin{cases} 1, & \text{if } di \in Cm \\ 0, & \text{if } di \in Cm \end{cases}$$

Logistic Regression Algorithm

Logistic regression comes under the supervised classification algorithm. This algorithm has achieved importance in recent times, and the use has increased extensively. This

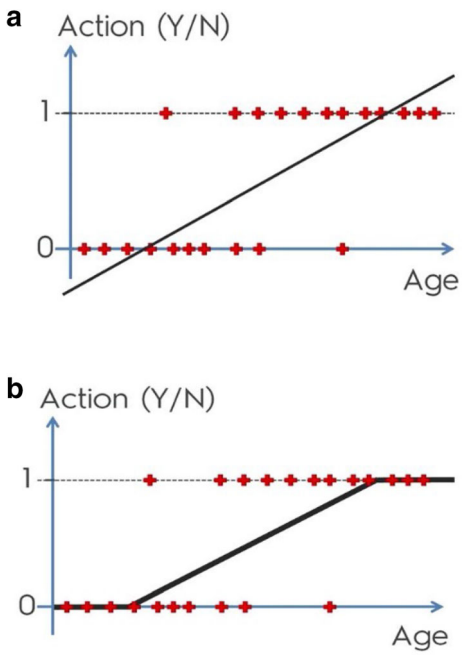

Fig. 6 **a** Graph when data points do not fit properly. **b** Graph when logistic regression is applied and one gets a perfect curve

algorithm is used to classify individuals in the categories based on logistic function [29]. There are many instances when one does not get a perfect graph that fits all the data points. For instances, we might encounter with problems like the graph mentioned in Fig. 6a.

The graph in Fig. 6a states how the action varies with respect to age. So, this graph is not at all appropriate and it does not fit to all the data points. The solution is to implement logistic regression algorithm. When we apply this algorithm to these set of data points, we get the graph as drawn in Fig. 6b. This graph is very much appropriate as it perfectly fits to all the data points. This graph or curve can be properly visualized as drawn in Fig. 7.

This is the speciality behind using logistic regression. The curve in Fig. 7 is obtained because of the use of sigmoid function in logistic regression. Sigmoid function is a mathematical function which is responsible for this S-shaped curve. The curve obtained above is also known as

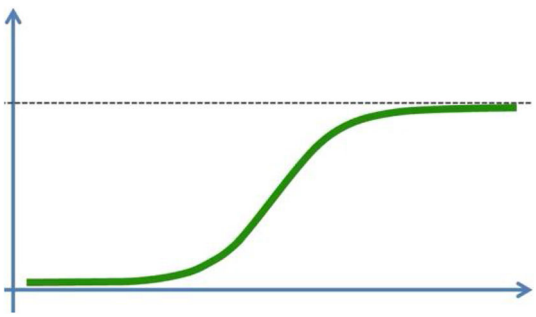

Fig. 7 Sigmoid curve

sigmoid curve. It is special case of logistic regression. To understand the mathematical version of the explanation, we will begin with a simple linear regression formula

$$y = b0 + b1 * x$$

So, now the sigmoid function is applied on it, and it is given by the formula

$$p = \frac{1}{1 + e^{-y}}$$

Now the value of y is calculated by substituting one formula into the other; we get our logistic regression formula as

$$\ln\left(\frac{p}{1-p}\right) = b0 + b1 * x$$

Or

$$\text{logit}(S) = b0 + b1M1 + b2M2 + b3M3 \ldots bkMk \ldots$$

where, S is probability of the presence of interest features.
$M1, M2, M3 \ldots\ldots Mk$ are the predictor value
$bo, b1, b2, b3 \ldots\ldots bk$ are the intercept of model.

- Assumptions in the logistic regression classifier:

1. There is no linear relationship between dependent and independent variables in logistic regression.
2. The dependent variable cannot be divided into two parts.
3. The dependent variables must not be normally distributed instead they should be linearly related.

In text classification, LR model recognizes a vector containing variables and then evaluates the coefficients for each input variable and predicts the class of text in the form of word vector.

Results and Outcome

After running the code successfully, the required output was obtained. As mentioned earlier, the comparison between the algorithms is done on the basis of five parameters namely accuracy, precision, $F1$-score, support and confusion matrix. Let us compare the three algorithms implemented one by one.

Logistic Regression

This model is used to measure the statistical significance of each independent variable with respect to probability. It is a powerful way of modelling binomial outcome. For example: if the person is going to suffer from cancer or not by taking values 0 and 1 with the use of one or many explanatory variables. The outcome variable in logistic

Table 5 Resultant outcome of all parameters in five different categories using logistic regression classifier

Category	Precision	Accuracy	*F*1-score	support
Business	0.94	0.99	0.97	133
Entertainment	1.00	0.98	0.99	91
Politics	0.97	0.94	0.96	103
Sports	0.99	0.99	0.99	131
Technology	0.98	0.96	0.97	99
Accuracy			0.97	557

regression is dichotomous. It is used to assign observation to a discrete set of classes and being a classification algorithm it very much depends on probability.

On applying logistic regression to the data set, the output was obtained in this manner:

```
[[132   0   1   0   0]
 [ 0  89   2   0   0]
 [ 3   0  97   1   2]
 [ 1   0   0 130   0]
 [ 4   0   0   0  95]]
```

Output Confusion Matrix

It is observed from Table 5 that an accuracy of 97% is obtained. This figure shows that all the parameters for each of the individual classes are calculated. For instance, the precision of business class obtained is 94%, accuracy is 99%, *F*1-score is 97%, and value of support is 133. Similarly, for entertainment, the values obtained are 100% precision, 98% accuracy, 99% *F*1-score and 91 supports. In politics, a precision of 97% is obtained, accuracy of 94%, 96% *F*1-score and 103 support is obtained. For sports, class precision obtained is 99%, accuracy of 99%, *F*1-score of 99% and support of 131. For technology class, a precision of 98% is obtained, accuracy of 96%, *F*1-score of 97% and support of 99 is obtained. In confusion matrix, each row and column corresponds to class so the value 132 in first row and first column indicates that the classifier is able to properly classify 132 lines of text belonging to business class and 1 on the first row third column indicates that the classifier has put it in another class. Similarly in second row, the value 89 means that the classifier is able to classify 89 lines of text that belongs to entertainment class and the rest it puts in another class. The third row states that 97 lines of text belongs to politics and the classifier is able to properly separate it whereas the other lines are put in another classes which the classifier is not able to do that. In the fourth row, this classifier is able to classify 130 lines of text for politics whereas the classifier is not able to classify that 1 line and hence puts it in business class. Lastly, the classifier is able to classify 95 lines of text that belongs to technology class and is not able to classify those 4 lines and hence puts it in business class.

Random Forest algorithm

Random forest is an ensemble method which is used to build predictive models for both regression and classification problems. It consists of the random number of trees which is used to give a desired output. It follows the ensemble method or learning. In problems that have classification, those decision trees vote for the most popular class whereas in regression problems the response of tree is an estimate of dependent variables given the predictors.

On applying logistic regression to the data set, the output was obtained in this manner:

```
[[134   2   6   0   2]
 [ 4  86   1   3   2]
 [ 6   2  84   1   0]
 [ 2   0   1 132   1]
 [ 3   2   0   0  83]]
```

Output Confusion Matrix

It is observed from Table 6 that an accuracy of 93% is obtained. This figure shows that all the parameters for each of the individual classes are calculated. For instance, the precision of business class obtained is 90%, accuracy is 93%, *F*1-score is 91% and value of support is 144. Similarly, for entertainment, the values obtained are 93% precision, 90% accuracy, 91% *F*1-score and 96 supports. In politics, a precision of 91% is obtained, accuracy of 90%, 91% *F*1-score and 93 supports is obtained. For sports class, precision obtained is 97%, accuracy of 97%, *F*1-score of 97% and support of 136. For technology class, a precision of 94% is obtained, accuracy of 94%, *F*1-score of 94% and support of 88 is obtained. In confusion matrix, each row and column corresponds to class so the value 134 in first row and first column indicates that the classifier is able to properly classify 134 lines of text belonging to business class and the rest the classifier puts in another class. Similarly in second row, the value 86 means that the classifier is able to classify 86 lines of text that belongs to entertainment class and the rest it puts in another class. The third row states that 84 lines of text belongs to politics and the classifier is able to properly separate it whereas the other

Table 6 Resultant outcome of all parameters in five different categories using random forest classifier

Category	Precision	Accuracy	*F*1-score	support
Business	0.90	0.93	0.91	144
Entertainment	0.93	0.90	0.91	96
Politics	0.91	0.90	0.91	93
Sports	0.97	0.97	0.97	136
Technology	0.94	0.94	0.94	88
Accuracy			0.93	557

lines are puts in another classes which the classifier is not able to do that. In the fourth row, this classifier is able to classify 132 lines of text for politics whereas the classifier is not able to classify the rest of the text lines and hence puts it in business, politics and technology class. Lastly, the classifier is able to classify 83 lines of text that belongs to technology class and is not able to classify those 5 lines and hence puts it in different classes.

K-Nearest Neighbours Algorithm

Similar to random forest, KNN is also used for both classification and regression problems, but in industries it is widely known for solving classification problems rather than regression. The input contains k-closest training examples lying in the feature space. It determines the group where a data point lies by looking all the data points.

On applying K-nearest neighbour to the data set, the output was obtained in this manner:

```
[[112   0  17   0   1]
 [ 2  90   6   0   1]
 [ 0   1 108   0   0]
 [ 1   0   3 125   0]
 [ 2   0   7   1  80]]
```

Output Confusion Matrix

It is observed from Table 7 that an accuracy of 92% is obtained. For instance, the precision of business class obtained is 96%, accuracy is 86%, *F*1-score is 91% and value of support is 130. Similarly, for entertainment, the values obtained are 99% precision, 91% accuracy, 95% *F*1-score and support of 99. In politics, a precision of 77% is obtained, accuracy of 99%, 86% *F*1-score and 109 supports is obtained. For sports class, precision obtained is 99%, accuracy of 97%, *F*1-score of 98% and support of 129. For technology class, a precision of 98% is obtained, accuracy of 89%, *F*1-score of 93% and support of 90 is obtained. In confusion matrix, each row and column corresponds to five different classes so the value 134 in first row and first column indicates that the classifier is able to properly classify 112 lines of text belonging to business class and the rest the classifier puts in another class. Similarly in second row, the value 90 means that the classifier is able to classify those 90 lines of text that belongs to entertainment class and the rest it puts in another class. The third row states that 108 lines of text belongs to politics and the classifier is able to properly separate it whereas the other lines are puts in another classes which the classifier is not able to do that. In the fourth row this classifier is able to classify 125 lines of text for politics whereas the classifier is not able to classify the rest of the text lines and hence puts it in business, politics and technology class. Lastly, the classifier is able to classify 80 lines of text that belongs to technology class and is not able to classify those other lines and hence puts it in different classes.

Table 7 Resultant outcome of all parameters in five different categories using K-NN classifier

Category	Precision	Accuracy	*F*1-score	support
Business	0.96	0.86	0.91	130
Entertainment	0.99	0.91	0.95	99
Politics	0.77	0.99	0.86	109
Sports	0.99	0.97	0.98	129
Technology	0.98	0.89	0.93	90
Accuracy			0.92	557

Comparison Analysis of Three Algorithms

After discussing the results, the authors decided to compare all the three algorithms based on the four parameters, i.e. precision, accuracy, *F*1-score and support. These four parameters are compared on the bar graph in order to display a perfect comparison. The four comparisons are shown below as:

1) Precision

The graph obtained is shown in Fig. 8

In the business section, logistic regression obtained precision of 0.94 (means 94%), random forest of 0.9 (means 90%) and KNN has a precision of 0.96 (means 96%). In the entertainment section, logistic regression obtained precision of 1 (100%), random forest had 0.93 (93%) precision and KNN has a precision of 0.99 (99%). In the politics class, precision for logistic regression obtained is 0.97 (97%), random forest has precision of 0.91 (91%) and KNN has 0.77 (77%). In the section for sports, logistic regression obtained a precision of 0.99 (99%), random forest with a precision of 0.97 (97%) and KNN with 0.99 (99%). At last in the technology section, logistic regression obtained a precision of 0.98 (98%), random forest has precision of 0.94 (94%) and KNN has 0.98 (98%) precision (Figure 8).

2) Accuracy

The graph for accuracy is shown is Fig. 9

In the business section, logistic regression obtained an accuracy of 0.99 (means 99%), random forest has an accuracy of 0.93 (means 93%) and KNN has an accuracy of 0.86 (means 86%). In the entertainment section, logistic regression obtained an accuracy of 0.98 (98%), random forest has accuracy of 0.90 (90%) and KNN has an accuracy of 0.91 (91%). In the politics class, accuracy obtained for logistic regression is 0.94 (94%), random forest has an accuracy of 0.9 (90%) and KNN has accuracy of 0.99 (99%). In the section for sports, logistic regression obtained

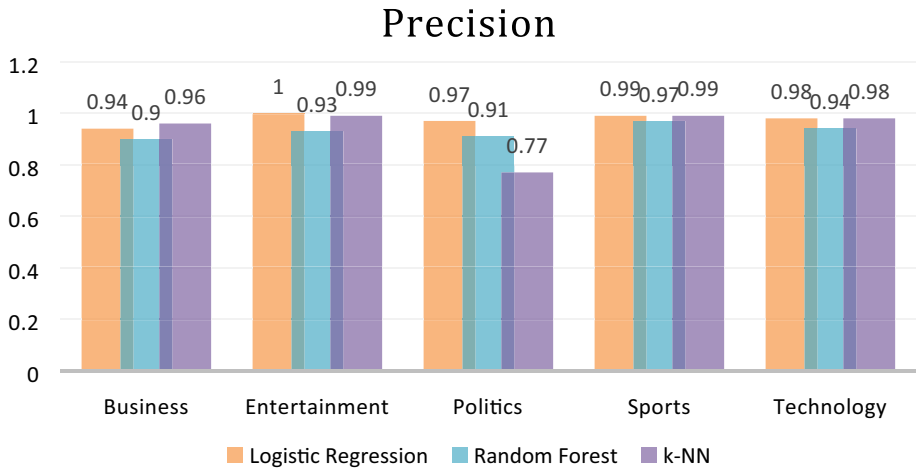

Fig. 8 Categories versus precision showing variations of data set classes with respect to change in precision

an accuracy of 0.99 (99%), random forest with accuracy of 0.97 (97%) and KNN has the accuracy of 0.97 (97%). Finally in the technology section, logistic regression obtained an accuracy of 0.96 (96%), random forest has an accuracy of 0.94 (94%) and KNN has an accuracy of 0.89 (89%) precision (Figure 9).

3) *F*1-score

The bar chart obtained for *F*1-score is shown in Fig. 10

In the business section, logistic regression obtained *F*1-score of 0.97 (means 97%), random forest with the *F*1-score of 0.91 (means 91%) and KNN has *F*1-score of 0.91 (means 91%). In the entertainment graph, logistic regression obtained a *F*1-score of 0.99 (99%), random forest has *F*1-score of 0.91 (91%) and KNN has *F*1-score of 0.95 (95%). In the politics section, *F*1-score obtained for logistic regression is 0.96 (96%), random forest has *F*1-score of 0.91 (91%) and KNN has the *F*1-score of 0.86 (86%). In the graph obtained for sports, logistic regression obtained *F*1-score of 0.99 (99%), random forest with the *F*1-score of 0.97 (97%) and KNN has the *F*1-score of 0.98 (98%). Finally, in the technology section, logistic regression obtained *F*1-score of 0.97 (97%), random forest has the *F*1-score of 0.94 (94%) and KNN with the *F*1-score of 0.93 (93%) (Figure 10).

4) Support

For the support parameter, the bar chart obtained is shown in Fig. 11

In the graph obtained for business class, logistic regression obtained support of 133, random forest with the support of 144 and KNN having a support of 130. In the entertainment graph, logistic regression obtained a support of 91, random forest has the support of 96 and KNN having the support of 99. In the politics section, the support obtained for logistic regression is 103, random forest has the support of 93 and KNN has the support of 109. In the sports graph, logistic regressions obtained have the support

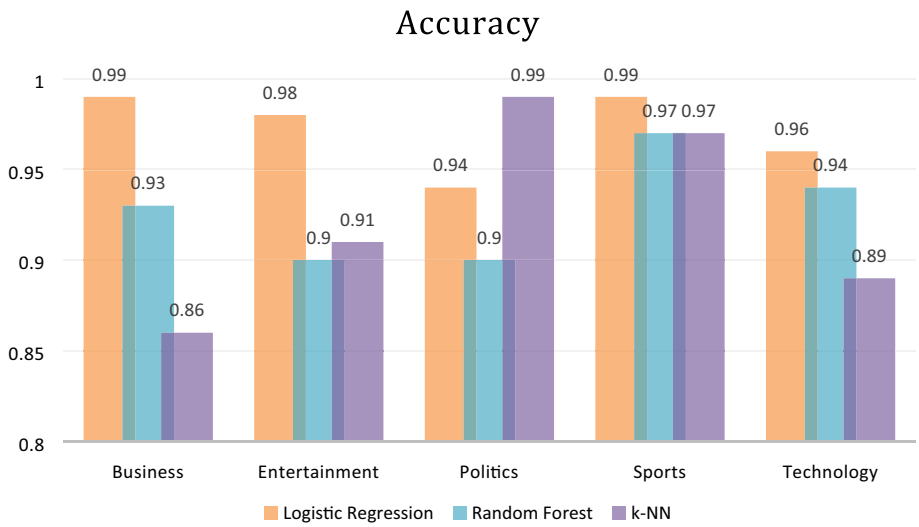

Fig. 9 Categories versus Accuracy showing variations of data set classes with respect to change in accuracy

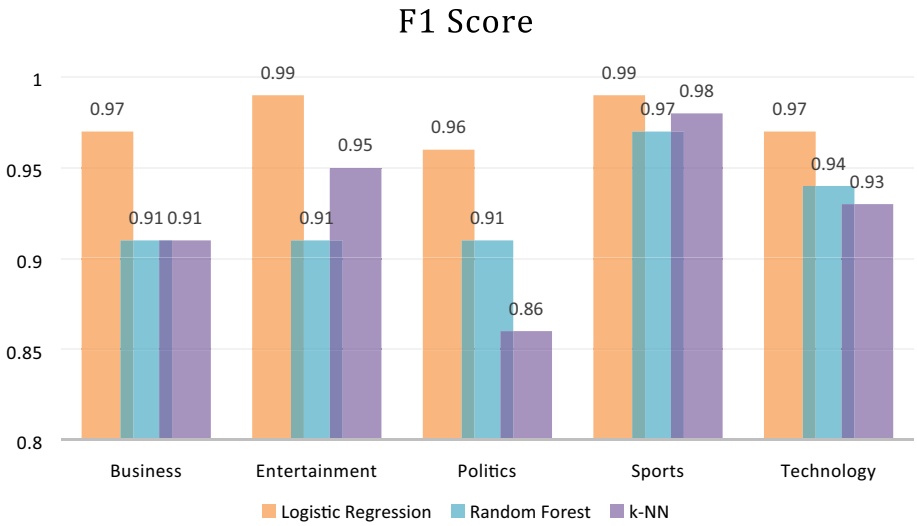

Fig. 10 Categories versus *F*1-score showing variations of data set classes with respect to change in *F*1-score

of 131, random forest with the support of 136 and KNN having the support of 129. Finally in the technology graph, logistic regression obtained the support of 99, random forest has the support of 88 and KNN having 90 as the support (Figure 11).

Challenges and Future Scope

While the model has been implemented with great accuracy and precision rate, there are certain challenges that can be looked upon for the future development of this work. The data set does not provide accuracy while using SVM (support vector machine) algorithm [28]. Also, the data set used here is completely statistical and text based. Random forest has shown great success in many real-world applications. The problem of learning from text data with class imbalance is the problem which we are being faced (Wu et al. [50]. The scope of these algorithms can extend to different data sets that can have features based on images and audios. Currently existing technologies for these challenges are image recognition for the images data set and POS (Part-Of-Speech) text recognition. Using these would provide a wide area of application for this research. A general analysis is being done for several classification algorithms in machine learning (e.g. logistic regression, decision trees, etc.) it is possible to explain and understand the model and the decisions given by the model [3].

The current development towards automation can be much progressed by the use of text classification applications. These can directly alter the easy of application by transforming the commands that we speak, into direct actions by machines [10]. Computer security susceptibility will always exist long as we have faulty security policies, computer system which is poorly configured. There is one important intrusion detection system which helps in

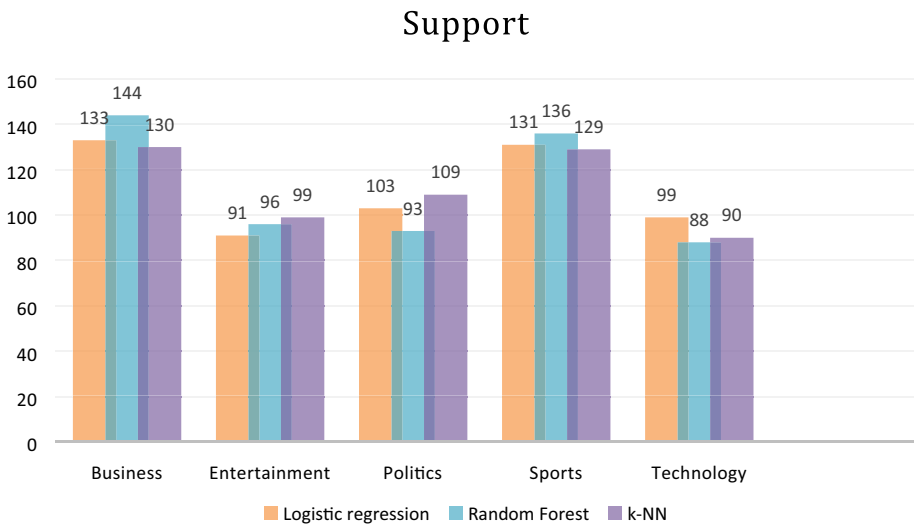

Fig. 11 Categories versus support showing variations of data set classes with respect to change in support

detecting attacks on susceptibility or the faulty security policies [27]. There is also a major problem which text classification faces is the high dimensionality of the feature space. There are several tens of thousands of features text domain has, though most of these features are not used for text classification task. Even some of them may sharply reduce the classification accuracy [10].

Conclusion

The current paper is constructing a BBC news text classification model based on machine learning algorithms. This paper proposes the logistic regression, random forest and K-nearest neighbour algorithms which describes every aspect of model in detail by providing the evaluation metrics. When machine learning algorithms are implemented on a particular data set, the most important parameter that matters is the accuracy. Hence, the result shows that logistic regression classifier with the TF-IDF Vectorizer feature attains the highest accuracy of 97% for the data set. This algorithm has emerged as the most stable classifier in a small data set. The second best was the random forest classifier with the accuracy of 93%. The algorithm with the least accuracy among the three was K-nearest neighbour with the overall accuracy of 92%. The logistic regression classifier gave a performance as expected in terms of all parameters. Hence, the output was obtained as per the expectations. The reason behind taking these three algorithms is mentioned in the related works section. Therefore, in order to find out the best fit algorithm, the authors decided to write this manuscript.

Acknowledgements The authors are grateful to Indus University and School of Technology, Pandit Deendayal Petroleum University for the permission to publish this research.

Authors Contribution All the authors make substantial contribution in this manuscript. KS, HP, DS and MS participated in drafting the manuscript. KS, HP and DS wrote the main manuscript; all the authors discussed the results and implication on the manuscript at all stages.

Funding Not Applicable.

Availability of Data and Material All relevant data and material are presented in the main paper.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no competing interests.

Consent for publication Not applicable.

Ethics approval and consent to participate Not applicable.

References

1. Ahir K, Govani K, Gajera R, Shah M (2020) Application on virtual reality for enhanced education learning, military training and sports. Augment Hum Res 5:7
2. Al Amrani Y, Lazaar M, El Kadiri KE (2018) Random forest and support vector machine based hybrid approach to sentiment analysis. Proc Comput Sci 127:511–520
3. Altınel B, Ganiz MC (2018) Semantic text classification: a survey of past and recent advances. Inf Process Manag 54(6):1129–1153
4. Aseervatham S, Antoniadis A, Gaussier E, Burlet M, Denneulin Y (2011) A sparse version of the ridge logistic regression for large-scale text categorization. Pattern Recogn Lett 32(2):101–106. https://doi.org/10.1016/j.patrec.2010.09.023
5. Aydoğan M, Karci A (2019) Improving the accuracy using pre-trained word embedding on deep neural networks for Turkish text classification. Stat Mech Its Appl, Physica A. https://doi.org/10.1016/j.physa.2019.123288
6. Bafna P, Pramod D, Vaidya A (2016) Document clustering: TF-IDF approach. In: 2016 international conference on electrical, electronics, and optimization techniques (ICEEOT), Chennai, pp 61–66
7. Bouaziz A, Dartigues-Pallez C, da Costa Pereira C, Precioso F, Lloret P (2014) Short text classification using semantic random forest. In: Bellatreche L, Mohania MK (eds) Data warehousing and knowledge discovery. DaWaK 2014. Lecture notes in computer science, vol 8646. Springer, Cham
8. Breiman L (2001) Random forests. Mach Learn 45(1):5–32
9. Chatzigeorgakidis G, Karagiorgou S, Athanasiou S, Skiadopoulos S (2018) FML-kNN: scalable machine learning on Big Data using k-nearest neighbor joins. J Big Data 5:4. https://doi.org/10.1186/s40537-018-0115-x
10. Chen J, Huang H, Tian S, Qu Y (2009) Feature selection for text classification with Naïve Bayes. Expert Syst Appl 36(3–1):5432–5435
11. Cheng Y, Rui K (2017) Text classification of minimal risk with three-way decisions. J Inf Optim Sci 39(4):973–987
12. Elghazel H, Aussem A, Gharroudi O, Saadaoui W (2016) Ensemble multi-label text categorization based on rotation forest and latent semantic indexing. Expert Syst Appl 57:1–11. https://doi.org/10.1016/j.eswa.2016.03.041
13. Ferrari A (2018) Natural language requirements processing: from research to practice. In: IEEE/ACM 40th international conference on software engineering: companion (ICSE-Companion), Gothenburg, pp 536–537
14. Gandhi M, Kamdar J, Shah M (2020) Preprocessing of Non-symmetrical images for edge detection. Augment Hum Res 5:10. https://doi.org/10.1007/s41133-019-0030-5
15. Garla V, Taylor C, Brandt C (2013) Semi-supervised clinical text classification with Laplacian SVMs: an application to cancer case management. J Biomed Inf 46(5):869–875
16. Genkin A, Lewis DD, Madigan D (2007) Large-scale Bayesian logistic regression for text categorization. Technometrics 49(3):291–304
17. Hmeidi I, Hawashin B, El-Qawasmeh E (2008) Performance of KNN and SVM classifiers on full word Arabic articles. Adv Eng Inf 22(1):106–111
18. Jani K, Chaudhuri M, Patel H, Shah M (2019) Machine learning in films: an approach towards automation in film censoring. J Data Inf Manag. https://doi.org/10.1007/s42488-019-00016-9
19. Jha K, Doshi A, Patel P, Shah M (2019) A comprehensive review on automation in agriculture using artificial intelligence. Artif Intell Agric 2:1–12

20. Jiang Y, Lin H, Wang X, Lu D (2011) A Technique for improving the performance of Naive Bayes text classification. In: Lecture notes in computer science, pp 196–203
21. Jiang S, Pang G, Wu M, Kuang L (2012) An improved K-nearest-neighbour algorithm for text categorization. Expert Syst Appl 39(1):1503–1509
22. Kabir M, Jahangir M, Xu S, Badhon B (2019) An empirical research on sentiment analysis using machine learning approaches. Int J Comput Appl. https://doi.org/10.1080/1206212x.2019.1643584
23. Kakkad V, Patel M, Shah M (2019) Biometric authentication and image encryption for image security in cloud framework. Multiscale Multidiscip Model Exp Des. https://doi.org/10.1007/s41939-019-00049-y
24. Kumar R, Kaur J (2020) Random forest-based sarcastic tweet classification using multiple feature collection. In: Tanwar S, Tyagi S, Kumar N (eds) Multimedia big data computing for IoT applications. Intelligent systems reference library, vol 163. Springer, Singapore
25. Kundalia K, Patel Y, Shah M (2020) Multi-label movie genre detection from a movie poster using knowledge transfer learning. Augment Hum Res 5:11. https://doi.org/10.1007/s41133-019-0029-y
26. Li J, Deng X, Yao Y (2013) Multistage email spam filtering based on three-way decisions. In: Lingras P, Wolski M, Cornelis C, Mitra S, Wasilewski P (eds) Rough sets and knowledge technology. RSKT 2013. Lecture notes in computer science, vol 8171. Springer, Berlin, pp 313–324
27. Liao Y, Vemuri VR (2002) Use of K-Nearest Neighbor classifier for intrusion detection. Comput Secur 22(5):439–448
28. Liu Y, Loh HT, Tor SB (2005) Comparison of extreme learning machine with support vector machine for text classification. In: Ali M, Esposito F (eds) Innovations in applied artificial intelligence. IEA/AIE 2005. Lecture notes in computer science, vol 3533. Springer, Berlin, pp 390–399
29. Liu YY, Yang M, Ramsay M, Li XS, Coid JW (2011) A comparison of logistic regression, classification and regression tree, and neural networks models in predicting violent re-offending. J Quant Criminol 27(4):547–553
30. Liu H, Zhang S, Wu X (2014) MLSLR: multilabel learning via sparse logistic regression. Inf Sci 281:310–320
31. Mehmood RM, Lee HJ (2015) Emotion classification of EEG brain signal using SVM and KNN. In: IEEE international conference on multimedia and expo workshops. IEEE, pp 1–5
32. Miao F, Zhang P, Jin L, Wu H (2018) Chinese news text classification based on machine learning algorithm. In: 2018 10th international conference on intelligent human-machine systems and cybernetics (IHMSC), Hangzhou, pp 48–51
33. Moldagulova A, Sulaiman RB (2018) Document classification based on KNN algorithm by term vector space reduction. In: 18th international conference on control, automation and systems (ICCAS), Daegwallyeong, pp 387–391
34. Nadi A, Moradi H (2019) Increasing the views and reducing the depth in random forest. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2019.07.018
35. Pandya R, Nadiadwala S, Shah R, Shah M (2019) Buildout of methodology for meticulous diagnosis of K-complex in EEG for aiding the detection of Alzheimer's by artificial intelligence. Augment Hum Res. https://doi.org/10.1007/s41133-019-0021-6
36. Parekh V, Shah D, Shah M (2020) Fatigue detection using artificial intelligence framework. Augment Hum Res 5:5
37. Patel D, Shah Y, Thakkar N, Shah K, Shah M (2020) Implementation of artificial intelligence techniques for cancer detection. Augment Hum Res. https://doi.org/10.1007/s41133-019-0024-3
38. Patel D, Shah D, Shah M (2020) The intertwine of brain and body: a quantitative analysis on how big data influences the system of sports. Ann Data Sci. https://doi.org/10.1007/s40745-019-00239-y
39. Prabhat A, Khullar V (2017) Sentiment classification on big data using Naïve bayes and logistic regression. In: International conference on computer communication and informatics (ICCCI), pp 1–5
40. Ranjitha KV (2018) Classification and optimization scheme for text data using machine learning Naïve Bayes classifier. In: IEEE world symposium on communication engineering (WSCE), pp 33–36
41. Raychaudhuri K, Kumar M, Bhanu S (2017) A comparative study and performance analysis of classification techniques: support vector machine, neural networks and decision trees. In: Advances in computing and data sciences, pp 13–21
42. Salles T, Gonçalves M, Rodrigues V, Rocha L (2018) Improving random forests by neighborhood projection for effective text classification. Inf Syst 77:1–21
43. Shah G, Shah A, Shah M (2019) Panacea of challenges in real-world application of big data analytics in healthcare sector. Data Inf Manag. https://doi.org/10.1007/s42488-019-00010-1
44. Solangi YA, Solangi ZA, Aarain S, Abro A, Mallah GA, Shah A (2018) Review on natural language processing (NLP) and its toolkits for opinion mining and sentiment analysis. In: IEEE 5th international conference on engineering technologies and applied sciences (ICETAS), pp 1–4
45. Szymaski J (2014) Comparative analysis of text representation methods using classification. Cybern Syst 45(2):180–199
46. Tan S (2006) An effective refinement strategy for KNN text classifier. Expert Syst Appl 30(2):290–298
47. Tan Y (2018) An improved KNN text classification algorithm based on K-Medoids and rough set. In: 10th international conference on intelligent human–machine systems and cybernetics (IHMSC), pp 109–113
48. Trstenjak B, Mikac S, Donko D (2014) KNN with TF-IDF based framework for text categorization. Proc Eng 69:1356–1364
49. Wahiba BA, Ahmed BEF (2015) New fuzzy decision tree model for text classification. In: The 1st international conference on advanced intelligent system and informatics (AISI2015), November 28–30, 2015, Beni Suef, Egypt, pp 309–320. https://doi.org/10.1007/978-3-319-26690-9_28
50. Wu Q, Ye Y, Zhang H, Ng MK, Ho S (2014) ForesTexter: an efficient random forest algorithm for imbalanced text Categorization. Knowl Based Syst 67:105–116
51. Yao H, Liu C, Zhang P, Wang L (2017) A feature selection method based on synonym merging in text classification system. EURASIP J Wirel Commun Netw 2017:166. https://doi.org/10.1186/s13638-017-0950-z
52. Yen SJ, Lee YS, Ying JC, Wu YC (2011) A logistic regression-based smoothing method for Chinese text categorization. Expert Syst Appl 38(9):11581–11590
53. Yuntao Z, Ling G, Yongcheng W, Yin Z (2003) An effective concept extraction method for improving text classification performance. Geo-Spatial Inf Sci 6(4):66–72
54. Zhu J, Wang H, Zhang X (2006) Discrimination-based feature selection for multinomial Naïve Bayes text classification. In: Lecture notes in computer science, pp 149–156