Machine Learning Based Text Classifier Centered On TF-IDF Vectoriser

Hema Kiran Yadla, Dr.PVRD Prasada Rao

Abstract: In 21st Century, Data is considered as New Oil. Given the spurt of Globalisation followed by Digitization, the size of Data has grown to an extent where classification of Text has become an exacting task. In the myriad of uncertainties task of text classification using Machine Learning can be enriched using various pre-processing techniques like stemming, lemmatization but usage of Term Frequency - Inverse Document Frequency (TF-IDF) helps further to refinement of text data as TF-IDF produces feature values for training a classifier. To further improvise the classification process, comparison amid the machine learning classification algorithms has been presented in the paper.

Index Terms: TF-IDF, Text Classification, Machine Learning, Neural Network, SVM, Feature Extraction, Document Classification.

1. INTRODUCTION

According to Reports from a survey undertaken by Ericsson, by 2024, mobile data usage will extend up to 131 exabytes per month. Mobile data is just one of the way data is created given the gamut of digital options available. Given the rapid increase in data text classification plays a vitals role in segregating the data into different classes based on the text content and nuances involved in the data. In mathematical terms, it's a mapping function where f i.e. the classifier, maps text in set x and categories of data in set y.

$$F : x \rightarrow y$$

Cataloguing of raw text is an extensive process, which needs exacting efforts. Preprocessing of original data makes the text more predictable and analysable for classifiers to work effectively therefore the input data is first channeled through various stages. Cleaning the HTML off from the text using unescap function, removing non-ASCII characters, stopwords and punctuation smoothens the text data and two key preprocessing techniques Stemming and Lemmatisation are also performed to attain the base forms of text and reduce size of corpus and improve text quality to make it more analysable. After the preprocessing stage, Feature extraction from the corpus has to be performed using TF-IDF this vectorization method helps in dimensionality reduction as it takes both the Term Frequency and Inverse Document Frequency into account. After certain extent of refinement, to extricate features from the corpus TF-IDF is used, it helps to reduce the dimensionality of data and makes it fit for preparing the machine learning algorithms.

- *Hema Kiran Yadla is currently pursuing masters degree program in Machine Learning and computation in Koneru Lakshmaiah Education Foundation (KLEF), Vaddeswaram, Guntur District, AP, India. PH-8712345427, Email: hemakiranyadla@gmail.com.*
- *Dr.PVRD Prasada Rao is professor in Computer Science Engineering in Koneru Lakshmaiah Education Foundation (KLEF), Vaddeswaram, Guntur District, AP, India, Email:pvrdprasad@kluniversity.in pvrdprasad@kluniversity.in*

Classification algorithms – Random Forest, Nearest Neighbours, Linear SVM, Decision Tree, RBF SVM, Neural Net, and Naive Bayes have been used to train on BBC news dataset consisting of 2225 tuples and categorized into 5 classes - business, entertainment, politics, sport, and tech.

2. LITERATURE SURVEY

In the Task of Document classification mining the essential features is critical to do this job we can use diverse preprocessing practices of which TF-IDF is pivot in document classification using Naïve Bayes this is emphasised in [1]. Rather than tackling with the raw text inputs converting the content to vector spaces and then applying the Support Vector Machine Classifier has shown good results under different kernel function and gamma parameters[3]. Documents with heterogeneous content like photographs, text and graphs etc. are challenging to handle the heterogeneous content Support Vector Machine Modal is deployed [2]. Document classification requires various filtering preprocesses before performing classification algorithms this decreases the originality of the content. To prevent this filtered classifier C4.5 Decision Tree is used to perform the task of classification. In preprocessing stage, Fayyad and Irani's discretization method is deployed to discretize numerical attributes into nominal attributes [5]. Classification of News is a complex, which can further delay the classification task due to high feature dimension of data. Softmax Regression algorithm, a generalization of logistic regression, is one way to handle the high feature dimension of data. Thus acquire good classification results [8]. Higher dimensionality and unstructured data gives a skewed classification result, to lessen this effect Deep Learning technique Word2vec is used along with improved TF-IDF to perform the classification task [4]. Classification of huge data has associated problems like sparseness and higher feature dimensions in the extraction method this reduces the models generalization ability. To dodge this problem Deep learning based Bi-LSTM-CNN is employed to get higher accuracy in predicting the class.

3. PROPOSED SYSTEM

Text Classification is performed in 3 progressive stages in an ordered manner:

i. Preprocessing phase

ii. Feature Extraction using TF-IDF
iii. Train and build ML classification algorithms to predict the class of input data.

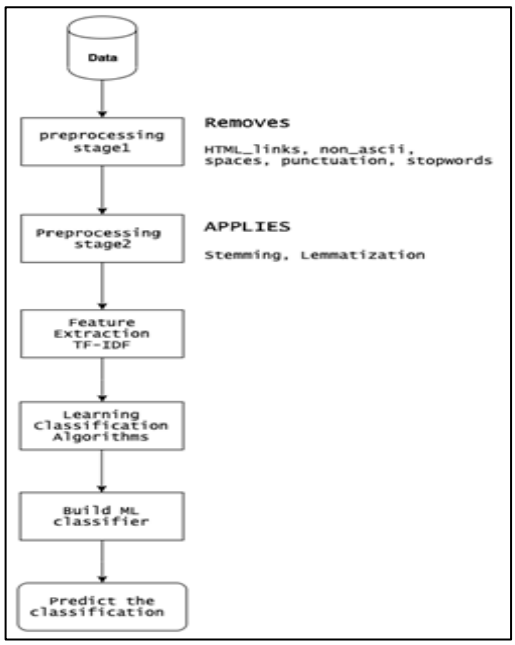

***Fig. 1**. Flowchart of Proposed System*

4. METHODOLOGY

4.1 Dataset Selection:

The BBC News dataset consists of articles related to business, entertainment, politics, sport, technology. It consists of 2225 tuples of heterogeneous text content so comprises of real world text classification problems like immense dimensionality unnecessary features, presence of trivial diction, case insensitive content.

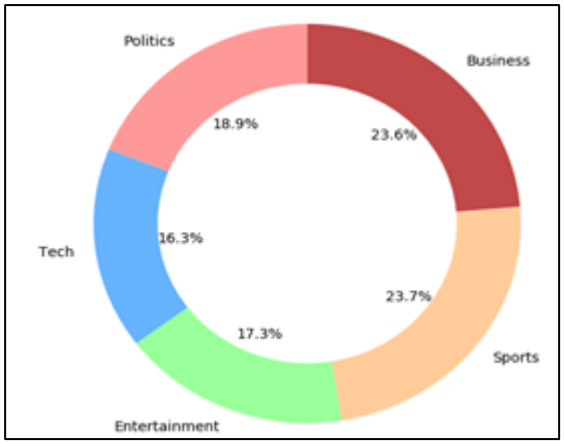

***Fig. 2**. Dataset Composition*

4.2 Data Preprocessing:

Data preprocessing refers to amending, substituting, scrapping imperfect or inappropriate data from the corpus thus making the data more utilisable and accomplishes a crucial role in functionality and performance of Machine Learning Algorithms. The data has to be preprocessed from all the irrelevant content like Non-ASCII characters, Punctuation, Stopwords in stage 1. Stage 2 of preprocessing includes stemming and lemmatization where the words from text corpus are converted to the base form to understand the context in which the word is used.

Stemming: Its normalization of words where the base form of the word is achieved by shedding the prefixes and suffixes.

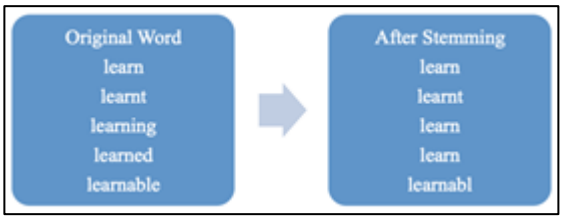

***Fig. 3**. Stemming transformation*

Lemmatization: It's more powerful preprocessing technique as it takes morphological analysis of the words into consideration. It converts the word into its base form and makes it a meaningful word for understanding the context in which it's used.

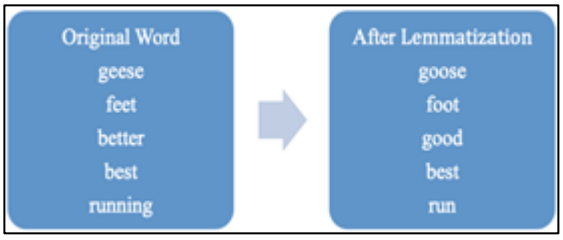

***Fig. 4**. Lemmatization transformation*

4.3 Feature Extraction:

Term Frequency – Inverse Document Frequency is deployed for vectorization of text, which can be further used in feature mining. TF-IDF entails of two factors Firstly, Term frequency i.e. total numeral times a given term appears in the text document alongside (per) the total tally of words in the text and secondly, The inverse document frequency which gauges how much information the word provides. It calculates the weight of a given word in the entire document. IDF show how common or rare a given word is across all documents. TF-IDF can be computed as tf * idf.

$$TF\ (t_1) = \frac{c_{i,j}}{\sum_k c_{i,j}}$$

$c_{i,j}$ is the total count of term t in a document.
$\sum_k c_{i,j}$ is the overall count of terms in the entire text.

$$Idf = \log\left(\frac{N}{df_t}\right)$$

N – Total documents taken
df_t – Total documents that have term t_1

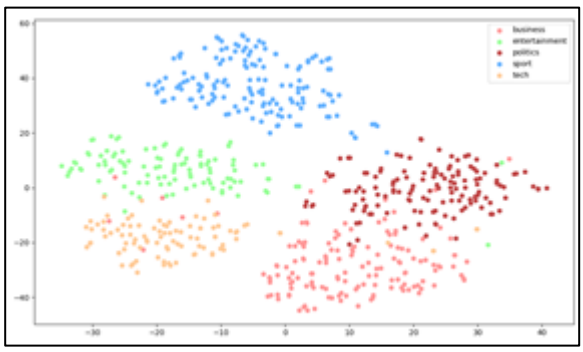

Figure 5. *Classification based on TF-IDF Vectorization*

4.4 Machine Learning Algorithms applied:

1. Decision Tree:
Decision tree is supervised machine learning algorithm that can be deployed for both Regression and Classification therefore also named as CART algorithm. It uses the concept of Recursive Partitioning or the Divide and Conquer approach to perform the classification problem at hand. Minor changes in the data can have repercussive changes in the entire decision tree structure, which in turn impacts the classification process. Accuracy attained using Decision Tree is 82.86%

2. K-Nearest Neighbour
K-Nearest Neighbour is a modest and easy use algorithm there's no necessity to pickle a model, refine several factors, or make further assumptions. But the major disadvantage is that it gets slower when the data set is large and has numerous classifications. k-Nearest Neighbour has its own drawbacks like less capable to handle high dimensionality. Accuracy acquired using k-Nearest Neighbour is 94.36%.

3. SVM Linear kernel and RBF kernel
Support Vector Machine uses the concept of Hyperplane and maximising the margin to segregate the classes. SVM deploys kernels to map the input spaces to feature spaces and perform the classification using hyperplane. SVM using Linear Kernel out performs RBF kernel because of lot of Features in the training data. A total of 15384 features are extracted from data set. SVM using Linear Kernel performed at 97.18% accuracy where as SVM RBF kernel performed slightly less accurate 96%.

4. Naïve Bayes:
Naïve Bayes algorithm is centred on Bayesian classification techniques hence rely on equation describing the association of restricted probabilities of statistical quantities. The uniqueness of Naïve Bayes is even if the features are related to each other, a Naive Bayes classifier would deliberate all of these properties individually when computing the probability of a specific outcome.

$$P(X|\,features) = \frac{P(features\,|X).P(X)}{P(features)}$$

5. Random Forest:
The Random forest is a blend of many decision trees and centred on ensemble machine learning techniques like Bootstrap Aggregation or bagging. Two key features of Random Forest is that Training data points are sampled while building trees and key features are extracted even when dimensionality is high. But Random Forest takes up large memory space when the data is huge because of tree size. Random forests acquired 95.53% accuracy.

6. Neural-Networks Multi-layer Perceptron classifier:
The basic Neural network Multi-layer Perceptron are comprised of one or more hidden layers made of neurons with specific weights and activation functions. Performance of Multi-layer perceptron classifier is 97.65%, highest among all the ML algorithms because of ability to extract informative features from heterogeneous data and can learn nonlinear functions.

4.5 Performance Measure:
Various evaluations metrics are used to make a conclusive result:

1. **Accuracy:** Accuracy is the fraction of correct predictions and total predictions made by the classifier.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TP = True positive, TN = True Negative, FP = False Positive, FN = False Negative.

2. **Kappa:** Kappa Score measures inter-rater reliability, Kappa Score of range 0.81 – 0.99 means near perfect agreement.

$$k = \frac{k_0 - k_e}{1 - k_e} = 1 - \frac{1 - k_0}{1 - k_e}$$

Where:
k_0 = The relative detected agreement between raters.
k_e = The hypothetical probability of chance agreement.

3. **Precision:** Quantity of positive identifications that are really exact or True.

$$Precision = \frac{\sum True\ Positive}{\sum Total\ Positive\ condition}$$

4. **Recall:** Quantity of actual positives was identified right.

$$Recall = \frac{\sum True\ Positive}{\sum Condition\ Positive}$$

5. **F1 Score:** Transfers the equilibrium between the precision and recall. The F1 score is the harmonic mean of the precision and recall.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

5. RESULTS

TABLE 1

COMPARISON TABLE OF VARIOUS PERFORMANCE METRICS OF VARIOUS CLASSIFICATION METHODS

Algorithms	Accuracy	Kappa
Decision Tree	85.68	81.91
KNN	94.36	92.92
Random Forest	95.53	93.50
SVM - RBF	96.00	94.96
Naïve Bayes	96.47	95.56
SVM - Linear	97.18	96.45
Neural Network	98.36	97.93

*All values are in percentages

TABLE 2

COMPARISON TABLE OF VARIOUS PERFORMANCE METRICS OF VARIOUS CLASSIFICATION METHODS

Algorithms	Precision	Recall	F1 score
Decision Tree	86.17	84.63	85.05
KNN	94.54	94.26	94.34
Random Forest	94.96	94.33	94.60
SVM - RBF	96.46	95.63	95.92
Naïve Bayes	96.64	96.31	96.44
SVM - Linear	97.14	97.19	97.15
Neural Network	98.33	98.20	98.26

*All values are in percentages

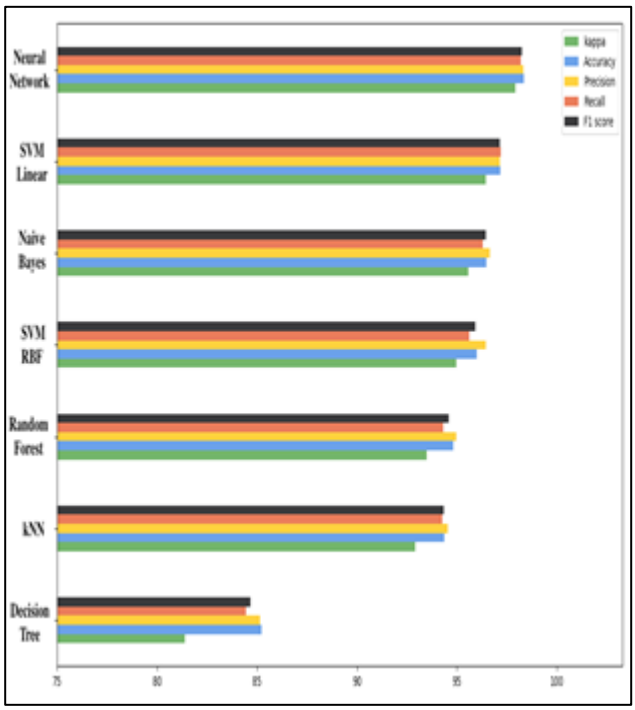

***Fig. 6**. Evaluation Metrics Comparison*

6. CONCLUSION

Groundbreaking Ideas are being projected on online platforms and substantial headways have been made in other applications like Speech Recognitions, Financial applications, Pattern Recognition, Text Recognition. Neural Networks using Multi-layer Perceptron composed of hidden layers where Neurons as basic unit, has attained an accuracy of 98.36%. Adam solver - a stochastic gradient-based optimizer is used with Hidden layer size attribute of MLP classifier fixed at [100,100], alpha is given a value 1, and Maximum Iterations are fixed at 400. With these parameters in hand the Neural Network-MLP has attained the highest accuracy in Text Classification task among all the Classifiers compared. Unlike single layer perceptron that is restricted to computing a single line of separation among classes, Multi-Layer Perceptron Performs best to obtain valuable features from heterogeneous data and can work with nonlinear functions. Neural Network-MLP has proved competent in Text Classification because of its Generalization and Fault Tolerance ability and thus improved the accurateness amid all classifications.

7. REFERENCES

[1] Basarkar et al., "Document Classification Using Machine Learning", 2017

[2] Kowsari et al. “Text Classification Algorithms: A Survey”, 2019

[3] J. Mtimet and H. Amiri, "Document class recognition using a support vector machine approach," 2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Monastir, 2016

[4] Chandrika et al. “An Efficient Filtered Classifier for Classification of Unseen Test Data in Text Documents”,2017.

[5] Li et al. “News text classification model based on topic model”, 2016.

[6] C. Liu et al. "Research of Text Classification Based on Improved TF-IDF Algorithm," 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE), Lanzhou, 2018.

[7] P.V.R.D.Prasad Rao et al. “Optimizing Genetic Algorithm For Neural Networks” published in International Journal of Pure and Applied Mathematics Volume 115 No. 8 2017, 219-225 ISSN: 1311-8080.

[8] Sujatha, M.M et al. ” Metrics for assessing quality of a web site”, International Journal of Innovative Technology and Exploring Engineering,2019.

[9] A.Yasaswini et al. “Automation of an IoT hub using artificial intelligence techniques” titled published in International Journal of Engineering & Technology, 7 (2.7) (2018) 25-27.

[10] PVRD. Prasada Rao et al. “Ailment Prognosis and Propose Antidote for Skin using Deep Learning” published in International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-4, February 2019.