ASSORTING THE WEB NEWS WITH THE APPLICATIONS OF MACHINE LEARNING ALGORITHMS

A PROJECT REPORT

Submitted by

M.MOUNIKA(17B91A1294)
L.SRAVANI (17B91A1285)
M.DIVIJA(17B91A1297)
MD FAIROJUNNISA (17B91A12A3)

in Partial Fulfilment for the Award of the Degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

SAGI RAMAKRISHNAM RAJU ENGINEERING COLLEGE
(AUTONOMOUS)
BHIMAVARAM-534204,ANDHRA PRADESH

2020-2021

SAGI RAMAKRISHNAM RAJU ENGINEERING COLLEGE(A)

CHINAMIRAM,BHIMAVARAM-534204

BONAFIDE CERTIFICATE

Certified that this project report "**ASSORTING THE WEB NEWS WITH THE APPLICATIONS OF MACHINE LEARNING ALGORITHMS** " is the bonafide work of "**MOUNIKA M [17B91A1294], SRAVANI L [17B91A1285], DIVIJA M [17B91A1297], FAIROJUNNISA MD [17B91A12A3]**" who carried out the project work under my supervision.

SIGNATURE **SIGNATURE**

B H V S RAMAKRISHNAM RAJU
HEAD OF THE DEPARTMENT
Professor,
Dept.of Information Technology,
Sagi Rama krishnam Raju Engineering College(A)
Chinamiram Bhimavaram-534 202

T.RAJASRI
GUIDE ,
Assistant Professor,
Dept.of Information Technology
Sagi Rama krishnam Raju Engineering College(A)
Chinamiram Bhimavaram-534 202

EXTERNAL EXAMINER

SELF DECLARATION

We here by declare that the project work entitled "**ASSORTING THE WEB NEWS WITH THE APPLICATIONS OF MACHINE LEARNING ALGORITHMS"** is a genuine work carried out by us in B.Tech., (Information Technology) at SRKR Engineering College(A), Bhimavaram and has not been submitted either in part or full for the award of any other degree or diploma in any other institute or University.

1.M.Mounika (17B910A1294)

2.L.Sravani(17B91A1285)

3.M.Divija(17B91A1297)

4. MD.Fairojunnisa(17B91A12A3)

ABSTRACT

Along with the increase in internet facilities and Mobile phones, exchanging of information is in huge scale. One of the major data exchanging sources is web applications. Handling of this huge data with existing machine learning methodologies reduce the time complexity and makes it cost effective compared with conventional methods. So using those methods, web news documents are classified into different categories like business, entertainment, marketing and technology

First of all the news document undergoes some text pre-processing to reduce noise and computation cost. Next we apply the feature selection onto the document to further separate important words and less important words inside the document .After applying feature selection the document is classified by the classifier. Then comparison between ML algorithms like SVM, KNN and LSTM is done. Among these the LSTM gives more accuracy.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

LSTM	Long Short Term Memory
SVM	Support Vector Machine
kNN	k- Nearest Neighbors
ROC	Receiver Operating Characteristic
TF-IDF	Term Fequency Inverse Document Frequency

1.INTRODUCTION

1.1 Introduction Concept

News information was not easily and quickly available until the beginning of last decade. But now, news is easily accessible via content providers such as online news services. A huge amount of information exists in form of text in various diverse areas, whose analysis can be beneficial in several areas. Classification is quite a challenging field in text mining, as it requires prepossessing steps to convert unstructured data to structured information. With the increase in the number of news it has got difficult for users to access news of his web, which makes it a necessity to categorize the news, so that it could be easily accessed. Categorization refers to grouping that allows easier navigation among articles. Web news needs to be divided into categories. This will help users to access the news of their web in real-time without wasting any time. When it comes to news it is much difficult to classify, as news are continuously appearing that need to be processed and those news could be never-seen-before and could fall in a new category. In this paper a review of news classification based on its contents is presented . For classification we are using ML and DL algorithms like SVM , KNN and LSTM.

1.2.OBJECTIVE

The main objective of this project is to -
Classifying the web news articles into different categories like entertainment, politics, sports, business and technology using Algorithms namely SVM,KNN and LSTM .We will find the efficient algorithm among them using classification metrics.

1.3. PROBLEM DEFINITION

Text mining has gained quite a significant importance during the past few years. Data, now-a-days is available to users through many sources like electronic media, digital media and many more. This data is usually available in the most unstructured form and there exists a lot of ways in which this data may be converted to structured form. In many real life scenarios, it is highly

desirable to classify the information in an appropriate set of categories. News contents are one of the most important factors that have influence on various sections. In this paper we have considered the problem of classification of news articles. This paper presents algorithms for category identification of news and have analyzed the shortcomings of a number of algorithm approaches.

The objective of this paper is to efficiently classify the web news into the specified five categories like entertainment, politics, sports, business and technology. In order to achieve this initially the Natural Language Processing techniques are applied in order to get the interesting pattern and efficient Machine Learning classification algorithms are applied namely SVM, KNN and LSTM, thus high accuracy is expected to be obtained.

1.4. UNIQUE FEATURES OF PROJECT:

Our project is implemented using machine learning which is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

Here in our project we check the accuracy of various Machine Learning algorithms like support Vector Machine,k-Nearest Neighbors, and Long short term memory and we choose the best among them for higher performance.

2. LITERATURE REVIEW

2.1 FEATURE SELECTION

Yen kan [1] presents an approach to the segment the URL into important portions and adds components, sequential and orthographic features to model silent features. Their results show that URL based approach surpasses the performance of full content and link-based approaches content attributes classifies a web page conferring to the words and sentences it contains.

Surajit Chandhuri [2] The existing studies surveyed provide the overview of data warehousing and OLAP technologies, with an emphasis on their new requirements. It describe back end tools for extracting, cleaning and loading data into a data warehouse ,multidimensional data models typical of OLAP, front end client tools for querying and data analysis, server extensions for efficient query processing and tools for metadata management and for managing the warehouse.Data warehousing and on-line analytical processing (OLAP) are essential elements of decision support, which has increasingly become a focus of the database industry. Many commercial products and services are now available, and all of the principal database management system vendors now have offerings in these areas. Decision support places some rather different requirements on database technology compared to traditional on-line transaction processing applications. This paper provides an overview of data warehousing and OLAP technologies, with an emphasis on their new requirements. We describe back end tools for extracting, cleaning and loading data into a data warehouse; multidimensional data models typical of OLAP; front end client tools for querying and data analysis; server extensions for efficient query processing; and tools for metadata management and for managing the warehouse. In addition to surveying the state of the art, this paper also identifies some promising research issues, some of which are related to problems that the database research community has worked on for years, but others are only just beginning to be addressed. This overview is based on a tutorial that the authors presented at the VLDB Conference, 1996

S.P.Deshpande[8] represents overview of data mining system and some of its application. Information play important role in every sphere of human life. It is very important to gather data from different data sources, store and maintain the data. Generate the information. ,

generate knowledge and disseminate data, information and knowledge to every stakeholder. Due to vast use of computers and electronics devices and tremendous growth in computing power and storage capacity, there is explosive growth in data collection. The storing of the data in data warehouse enables entire enterprise to access a reliable current database

Mita K. Dalal[9] worked on text classification and feature extraction phases. Text classification can be automated successfully using machine learning techniques, however pre-processing and feature selection steps play a crucial role in the size and quality of training input given to the classifier, which in turn affects the classifier accuracy.

Dadgar[12] proposed an approach to classify news texts. This approach was comprised of three different steps: 1) text preprocessing, 2) feature extraction based on TF-IDF, and 3) classification based on SVM. They trained the approach through the SVM classifier which was selected because it could support data with high dimensions.

Xindong Wu presents the top 10 data mining algorithms .C4.5,k-means,SVM,Apriori ,EM,page rank,AdaBoost, kNN,Naïve Bayes and CART. These top 10 algorithms are among the most influential data mining algorithms in the research community. This paper presents the top 10 data mining algorithms identified by the IEEE International Conference on Data Mining (ICDM) in December 2006: C4.5, *k*-Means, SVM, Apriori, EM, PageRank, AdaBoost, *k*NN, Naive Bayes, and CART. These top 10 algorithms are among the most influential data mining algorithms in the research community. With each algorithm, we provide a description of the algorithm, discuss the impact of the algorithm, and review current and further research on the algorithm. These 10 algorithms cover classification, clustering, statistical learning, association analysis, and link mining, which are all among the most important topics in data mining research and development

2.2 SUPPORT VECTOR MACHINE:

Thorsten Joachims[3] introduces support vector machines for text categorization. It provides both theoretical and empirical evidence that SVMs are very well suited for text categorization.The paper considers the problem of automated categorization of web sites for systems used to block web pages that contain inappropriate content. In the paper we applied the techniques of analysis of the text, html tags, URL addresses and other information using

Machine Learning and Data Mining methods. Besides that, techniques of analysis of sites that provide information in different languages are suggested. Architecture and algorithms of the system for collecting, storing and analyzing data required for classification of sites are presented. Results of experiments on analysis of web sites' correspondence to different categories are given. Evaluation of the classification quality is performed. The classification system developed as a result of this work is implemented in F-Secure mass production systems performing analysis of web content

Hyeran Byun [4] presented a brief introduction on SVMs and several application of SVMs in pattern recognition problems . SVMs have been successfully applied to a number of applications ranging from face detection and recognition, speaker and speech recognition,information and image retrieval, prediction and etc. because they have yielded excellent generalization performance on many statistical problems without any prior knowledge and when the dimension of input space is very high. He present a comprehensive survey on applications of Support Vector Machines (SVMs) for pattern recognition. Since SVMs show good generalization performance on many real-life data and the approach is properly motivated theoretically, it has been applied to wide range of applications. This paper describes a brief introduction of SVMs and summarizes its numerous applications.

Chee Hong[5] experimented an automated approach to classify online news using the SVM (Support Vector Machine) classification method. SVM has been shown to give good classification results when ample training documents are given. Classiflcation of online news, in the past, has often been done manually. In our proposed Categorizor system, we have experimented an automated approach to classify online news using the Support Vector Machine (SVM). SVM has been shown to deliver good classiflcation results when ample training documents are given. In our research, we have applied SVM to personalized classiflcation of online news. In personalized classiflcation, users can deflne their personalized categories using a few keywords. By constructing search queries using these keywords, Categorizor obtains both positive and negative training documents required for the construction of personalized classiflers. In this paper, we describe the preliminary version of Categorizor and present its system architecture.

Daniel I Morariu[6] investigated three approaches to build an efficient meta-classifier. In order to increase the classification accuracy. In this select 8 different SVM classifiers. For each

of the classified we modified the kernel, the degree of the kernel and the input data representation. Text categorization is the problem of classifying text documents into a set of predefined classes. In this paper, we investigated three approaches to build a meta-classifier in order to increase the classification accuracy. The basic idea is to learn a meta-classifier to optimally select the best component classifier for each data point. The experimental results show that combining classifiers can significantly improve the accuracy of classification and that our meta-classification strategy gives better results than each individual classifier. For 7083 Reuters text documents we obtained a classification accuracies up to 92.04%

D. Morariu, R. Cre¸tulescu [7] building up on the metaclassifier presented, based on 8 SVM components, they add to these a new Bayes type classifier which leads to a significant improvement of the upper limit that the meta- classifier can reach. Krishanalal developed the intelligent News Classifier and experimented with online news from web for the category Sports, Finance and Politics**.** Text categorization is the problem of classifying text documents into a set of predefined classes. In this paper, we investigated two approaches: a) to develop a classifier for text document based on Naive Bayes Theory and b) to integrate this classifier into a meta-classifier in order to increase the classification accuracy. The basic idea is to learn a meta-classifier to optimally select the best component classifier for each data point. The experimental results show that combining classifiers can significantly improve the classification accuracy and that our improved meta-classification strategy gives better results than each individual classifier. For Reuters2000 text documents we obtained classification accuracies up to 93.87%.

2.3 COMPARISION OF DIFFERENT CLASSIFIERS:

Yiming Yang[10] reported a controlled study with statistical significance tests on five text categorization methods: the Support Vector Machines (SVM), a k-Nearest Neighbor (kNN) classifier, a neural network (NNet) approach, the Linear Least- squares Fit (LLSF) mapping and a Naive Bayes (NB) classifier. They focus on the robustness of these methods in dealing with a skewed category distribution, and their performance as function of the training-set category frequency .

Rama Bharath Kumar[11] developed stock market prediction tool. Stock market prediction is an attractive research problem to be investigated. News contents are one of the most important factors that have influence on market. Considering the news impact in analyzing the stock market behavior, leads to more precise predictions and as a result more profitable trades. So far various prototypes have been developed which consider the impact of news in stock market prediction. In this paper, the main components of such forecasting systems have been introduced. The main objective is to predict the Classify the Financial News based on the contents of relevant news articles which can be accomplished by building a prediction model which is able to classify the news as either rise or drop.

Shri Zhong [13] compared generative models based on the multivariate Bernoulli and multinomials distributions have been widely used for text classification. Recently,the spherical; k-means algorithm ,which has desirable properties for text clustering,has been shown to be a special case of a generative model based on a mixture of von Mises-Fisher(vMF) distributions. Text classification, document clustering and similar document analysis are the most important areas of data mining. It is currently the subject of significant global research since such areas strengthen the enterprises of web intelligence, web mining, web search engine design, and so forth. Generative models based on the multivariate Bernoulli and multinomial distributions have been widely used for text classification. Recently, the spherical k-means algorithm, which has desirable properties for text clustering, has been shown to be a special case of a generative model based on a mixture of von Mises-Fisher (vMF) distributions. This paper compares these three probabilistic models for text clustering using a general model-based clustering framework. In this work, three implementations namely, Bernoulli model-based clustering (Bernoulli-based k-means), Multinomial model-based clustering (multinomial-based k-means), von Mises-Fisher model-based clustering (vMF-based k-means) have been implemented and evaluated using suitable metrics. These algorithms have been implemented in MATLAB and evaluated with additional metric Rand Index

3. SYSTEM DESIGN

3.1 SYSTEM ANALYSIS

3.1.1 Existing System

In Existing System, machine learning methodologies reduce the time complexity and makes it cost effective compared with conventional methods. Web applications are regarded as a popular platform to exchange information with users. These applications have to be able to process Big-Data quickly and to serve users in a timely manner with accurate information posted in news portals which can be a huge challenge to overcome. Huge computation power is needed to crawl the web and process big-data and the methods are needed to be developed to reduce space and time complexity of this process.

3.1.2 Proposed System

This paper proposes an efficient model for web that extracts news information and sorts news articles into five different categories business, technology , entertainment , sports and politics. First of all the news article undergoes some text preprocessing to reduce noise and computation cost. Next we apply the feature selection onto the article to further separate important words and less important words inside the article. After applying feature selection the article is classified by the classifier. Then comparison between ML algorithms like SVM, KNN and LSTM is done. Among these the LSTM gives more accuracy.

3.2 SYSTEM REQUIREMENTS

3.2.1 Hardware Requirements

Minimum hardware requirements are

- Processors - Intel® Core™ i3 processor and above processor
- RAM – 4 GB
- HARD DISC:1TB

3.2.2 Software Requirements

Minimum software requirements are

- Operating System - Windows 7/8/10 ,Linux
- Coding Language - Python 3.7.x or above with Pre installed packages such as NumPy, Pandas, Tkinter.
- chrome browser

3.2.3 Technologies And Tools

- Python 3.7.2 and above
- NLP packages
- Machine Learning Module

3.2.4 Software Specification

(1) Less Response Time
(2) High data utility
(3) Reduce the effort of the user
(4) Better Performance
(5) Less computational intensity

Requirements Model:

Requirements analysis, also called as requirements engineering, is the process of determining user expectations for a new or modified product, These features, called requirements, must be quantifiable ,relevant and detailed. In software engineering, such requirements are often called functional specifications. Requirements analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

Functional Requirements：

Data Preprocessing: It is extremely important that preprocess data before feeding it into our model.

Input: Input the dataset containing articles of the news.

Output: Returns the data frame of by removing stop words and break the sentences into tokens which are stored in the pandas data frame.

Feature Extraction: The main idea of feature selection is to choose a subset of input variables by eliminating features with little or no predictive information.

Input: Input the resulted tokens from the preprocessing stage.

Output: Returns the feature vector which contains the words and their frequencies in all news as key-value pairs.

Data Classification: Labeled records of specific category are to be used as the training data. When the classifier is trained accurately, it can be used to detect an unlabeled record.

Input: Input the feature vector generated from the previous step then SVM, KNN and LSTM Classifiers can be used to train the data.

Output: Return the label of the news article.

4.IMPLEMENTATION

4.1 ALGORITHMS:

The algorithms are used to cluster the time series datasets that are used in this work. Four classification algorithms are used in this experiment.

4.1.1. Support Vector Machine:

The main objective of the support vector machine algorithm is to find a hyper plane in an N-dimensional space that distinctly classifies the data points. To separate two classes of data points, there will many possible hyper planes that could be chosen. Our objective is to find a plane that has the maximum margin,

that means the maximum distance between the data points of both classes. Maximizing margin distance provides some reinforcement so that the future data points can be classified with more confidence.

The step wise algorithm is presented for an individual document below:-

Step1: Consider a random record from the newsCorpus record which contains the attributes of the news article

Step2: The considered news article is in raw form. To perform the feature extraction/selection and classification procedure, initially data is needed to pre-process. Pre-processing involves the steps of tokenization, lemmatization and stop word removal.

2.1. Initially, tokenize the text into individual keywords. Tokenization split each individual word into different token.

2.2. Remove the stop words from the obtained tokens.

2.3 .Remove the punctuation,Special characters and numbers

2.4. Perform lemmatization on the tokens obtained from the previous step. Lemmatization process reduces the size of word to its root word. For lemmatizing, a predefined list of possible words with their respective lemma words is considered.

2.4.1. For lemmatizing, a list of suffix words is stored into array with their respective root words.

2.4.2. Consider check token = availability in considered array of root word

2.4.3. If suffix of check token = true, lemma the word to its respective root word from the list of array.

2.4.4. Else, there is no need of lemmatizing. Word is already in its root word format. Move to next token.

Step 3: Apply Tf-idf vectorizer to each and every word such that for every unique word in our article a vector of float type is assigned. Now these Tf-idf values of each token is given as input to the SVM classifier.

Step 4: Based on the evaluated feature similarity using SVM classification model of tokens they are labeled according to the target categories list which contains target labels entertainment, politics, sports, business and technology.

Step 5: Store the record that was classified and repeat the process for all the records.

4.1.2 k-Nearest Neighbors classifier:

The kNN technique works by using articles from the trainng dataset after they are preprocessed to build its model. kNN algorithm assumes the similarity between the new case/data and available cases category that is most similar to the available categories.kNN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using kNN algorithm.

The step wise algorithm is presented for an individual document below:-

Step1: Consider a random record from the newsCorpus record which contains the attributes of the news article

Step2: The considered news article is in raw form. To perform the feature extraction/selection and classification procedure, initially data is needed to pre-process. Pre-processing involves the steps of tokenization, lemmatizing and stop word removal.

2.1. Initially, tokenize the text(sentence) into individual keywords. Tokenization split each individual word into different token.

2.2. Remove the stop words from the obtained tokens.

2.3 .Remove the punctuation,Special characters and numbers

2.4. Perform lemmatization on the tokens obtained from the previous step. Lemmatization process reduces the size of word to its root word. For lemmatizing, a predefined list of possible words with their respective lemma words is considered.

2.4.1. For lemmatizing, a list of suffix words is stored into array with their respective root words.

2.4.2. Consider check token = availability in considered array of root word

2.4.3. If suffix of check token = true, lemma the word to its respective root word from the list of array.

2.4.4. Else, there is no need of lemmatizing. Word is already in its root word format. Move to next token.

Step 3: Apply Tf-idf vectorizer to each and every word such that for every unique word in our article a vector of float type is assigned. Now these Tf-idf values of each token is given as input to the kNN classifier by setting the kneighbors attribute as 5.

Step 4: Based on the evaluated feature similarity using kNN tree classification model of tokens they are labeled according to the target categories list which contains target labels entertainment, politics, sports, business and technology.

Step 5: Store the record that was classified and repeat the process for all the records.

4.1.3 Long Short Term Memory

The long short-term memory classifier was implemented with the MATLAB train Network function. LSTM achieved an accuracy of 97.31 % making LSTM the first best solution. Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. This is a behavior required in complex problem domains like machine translation, speech recognition, and more.LSTMs are a complex area of deep learning. It can be hard to get your hands around what LSTMs are, and how terms like bidirectional and sequence-to-sequence relate to the field.

The step wise algorithm is presented for an individual document below:-

Step 1: Consider a random record from the newsCorpus record which contains the attributes of the news article

Step2: The considered news article is in raw form. To perform the feature extraction/selection and classification procedure, initially data is needed to pre-process. Pre-processing involves the steps of tokenization, lemmatizing and stop word removal.

2.1. Initially, tokenize the text into individual keywords. Tokenization split each individual word into different token.

2.2. Remove the stop words from the obtained tokens.

2.3 .Remove the punctuation, Special characters and numbers

2.4. Perform stemming on the tokens obtained from the previous step. Lemmatizing process reduces the size of word to its root word. For lemmatizing, a predefined list of possible words with their respective stem words is considered.

2.4.1. For lemmatizing, a list of suffix words is stored into array with their respective root words.

2.4.2. Consider check token = availability in considered array of root word

2.4.3. If suffix of check token = true, lemma the word to its respective root word from the list of array.

2.4.4. Else, there is no need of lemmatizing. Word is already in its root word format. Move to next token.

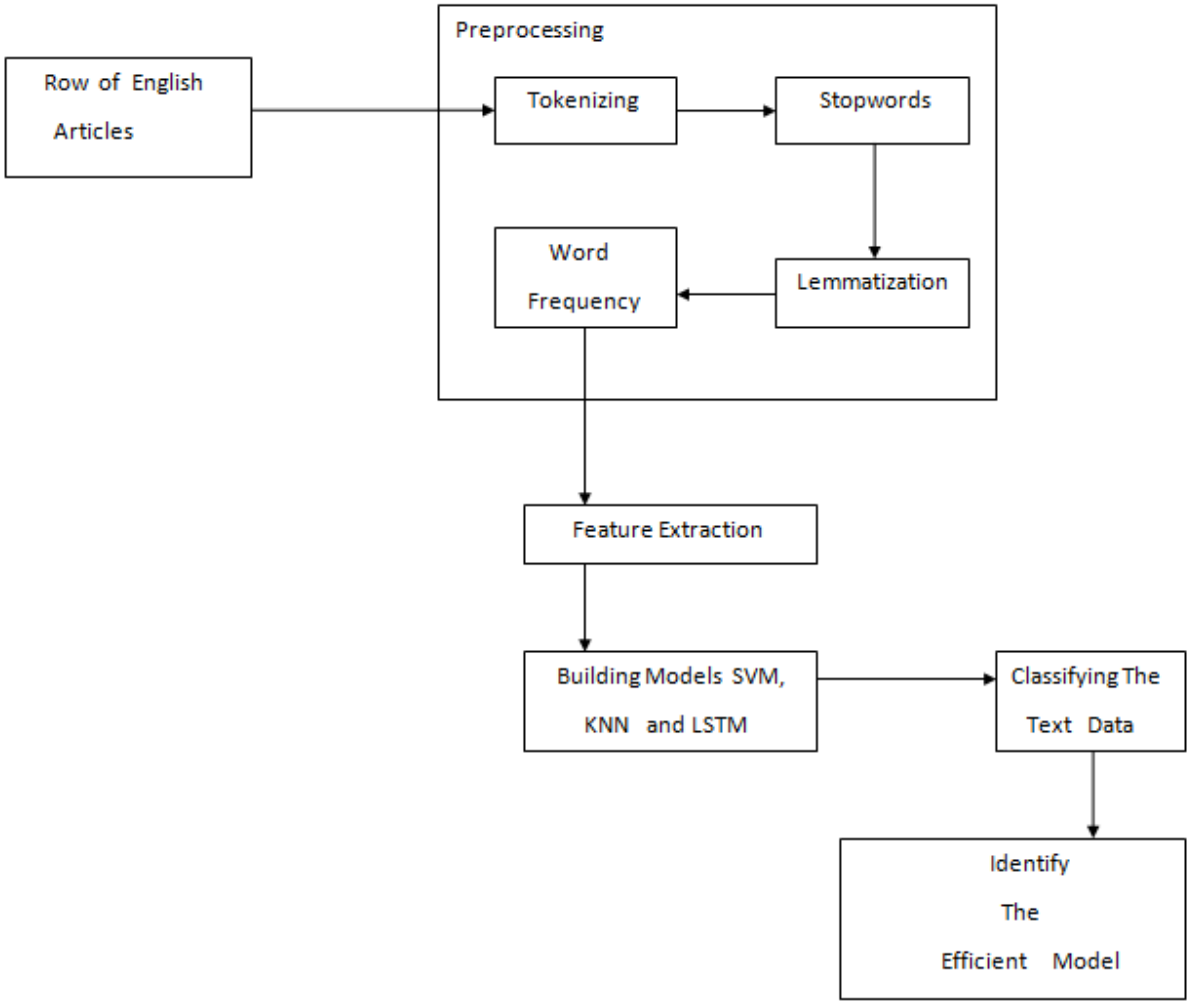
Step3: The text is converted into the sequences ,inorder to decrease the overhead of unequal lengths in the sequences padding of the sequences is done.The target variables list are encoded using the one-hot encoding model.

Step4: The available sequences and encoded target variables are given for recurrent neural network to perform the training phase.

Step5: Based on the evaluated feature similarity using LSTM recurrent neural network model of tokens they are labeled according to the target categories list which contains target labels entertainment, politics, sports, business and technology.

Step6: Store the record that was classified and repeat the process for all the records.

4.2 ARCHITECTURE



4.2.1 Modules Description

(Definitions):

Data Preprocessing:

Data preprocessing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of model to learn therefore, it is extremely important that preprocess data before feeding it into our model. Three techniques to preprocess the data those are performed in order to remove unnecessary , redundant , Noisy data and computational cost. They are:

• Tokenization

• Stop Word Removal

• Lemmatization

Preprocessing step is significant in order to make the dataset ready for classification. The steps involved in preprocessing are,

Firstly 1)checking the Null values that is checking whether the dataset contains any null values.

2)checking whether the dataset is balanced or not using count of each category.

3)removing of punctuation, and special symbols are removed including ".", "%" and "@".

The preprocessing steps are discussed below

- **Tokenization:**

Given a character sequence and a defined article unit (blurb of texts), tokenization is the task of chopping it up into pieces, called tokens, perhaps at the same time throwing away certain characters/words, such as punctuation. Ordinarily, there are two types of tokenization:

i) Word Tokenization:-Used to separate words via unique space character. Depending on the application, word tokenization may also tokenize multi-word expressions like "New York". This is often times is closely tied to a process called "Named Entity Recognition".
Example "Machine learning makes benefit to humans" can be converted to ["Machine" ,"Learning" ,"makes" ,"benefit", "to", "humans"]

- **Stop Word Removal**:-

A stop word is a commonly used word (such as "the", "a", "an", "in") that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query. We would not want these words taking up space in our database, or taking up valuable processing time. For this, we can remove them easily, by storing a list of words that you consider to be stop words. After generating the tokens in the previous step. Now we remove the stop words from the generated tokens i. e, words. Example In the list of tokens ["Machine" ,"Learning" ,"makes" ,"benefit" ,"to" ,"humans"] we remove "to" because which does not contribute to the meaning of the sentence.

- **Stemming And Lemmatization:-**

Word stemming is a fundamental rule based process that strips common suffixes and prefixes from the word to normalize it to its root. Lemmatization on the other hand is a step by step process of converting a word to its root by considering the vocabulary and morphological analysis.

Lemmatization is found to be more powerful and organized as compared to stemming as it uses dictionaries which are formed using in depth linguistic knowledge. It helps to obtain better features as compared to stemming.

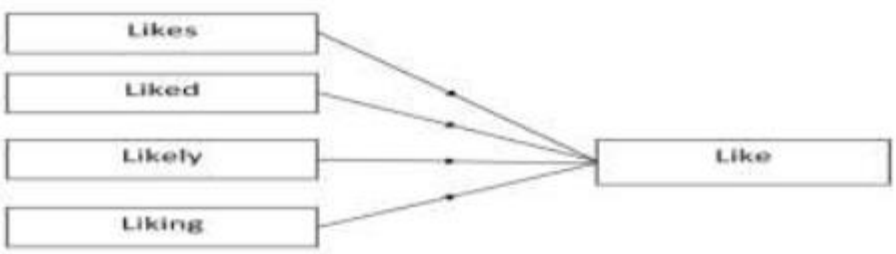


Fig:Example of Lemmatization

Figure demonstrates an example of lemmatization where different words ‘Likes’, ‘Liked ‘, ‘Likely’, ‘Liking’ are all converted into its base word ‘Like’ by slicing off the suffixes. Similarly, the words ‘Playing’, ‘Played’, ’Plays’ get transformed into its root form ‘Play’.

- **Label coding**

Machine learning models require numeric features and labels to provide a prediction. For this reason we must create a dictionary to map each label to a numerical ID. We have created this mapping scheme:

	ArticleId	Text	Category	News_length	Text_parsed	Category_target
0	1833	worldcom ex-boss launches defence lawyers defe...	business	1866	worldcom exboss launch defence lawyer defend f...	0
1	154	german business confidence slides german busin...	business	2016	german business confidence slide german busine...	0
2	1101	bbc poll indicates economic gloom citizens in ...	business	3104	bbc poll indicate economic gloom citizen major...	0
3	1976	lifestyle governs mobile choice faster bett...	tech	3618	lifestyle govern mobile choice fast good funky...	4
4	917	enron bosses in $168m payout eighteen former e...	business	2190	enron boss 168m payout eighteen former enron d...	0

- **Word Frequencies**

Word counts are a good starting point, but are very basic.

Here,we do not just capture the existence of words for a given articles but also capture how many times it occurs. Thus, for each word in a vocabulary that occurs in a articles we capture

the number of times it occurs. Thus, is to find the frequency of different words in all these articles.

Feature Extraction with TfidfVectorizer:

Machine Learning algorithms learn from a pre-defined set of features from the training data to produce output for the test data. But the main problem in working with language processing is that machine learning algorithms cannot work on the raw text directly. So, we need some feature extraction techniques to convert text into a matrix(or vector) of features. and by far the most popular method is called TF-IDF. This is an acronym than stands for "*Term Frequency – Inverse Document*" Frequency which are the components of the resulting scores assigned to each word.

- Term Frequency: This summarizes how often a given word appears within a document.
- Inverse Document Frequency: This downscales words that appear a lot across documents.

TF-IDF is the product of TF and IDF. It is formulated as:

tf-idf = log(word_count_in_particular_doc)* log(total no of docs / no of docs that has word)

The TfidfVectorizer will tokenize documents, learn the vocabulary and inverse document frequency weightings, and allow you to encode new documents. Alternately, if you already have a learned CountVectorizer, you can use it with a TfidfTransformer to just calculate the inverse document frequencies and start encoding documents.

The same create, fit, and transform process is used as with the CountVectorizer.

- **Train — Test split**

We need to set apart a test set in order to prove the quality of our models when predicting unseen data. We have chosen a random split with 75% of the observations composing the training test and 25% of the observations composing the test set. We will perform the hyperparameter tuning process with cross validation in the training data, fit the final model to it and then evaluate it with **totally unseen** data so as to obtain an evaluation metric as less biased as possible.

- **Building Models**

We have used machine learning models to figure out which one may fit better to the data and properly capture the relationships across the points and their labels. We have only used classic

machine learning algorithms and one deep learning algorithms,and the overall steps involved in those algorithms are discussed in (4.1)

We have tried the following models:

- Support Vector Machine
- K Nearest Neighbors
- Long Short Term Memory

- **Predict The Text Data or Performance Measurement:**

After Building the models with the training data and fitting the model to this training data, we need to evaluate its performance on totally unseen data (the Dataset). When dealing with classification problems, there are several metrics that can be used to gain insights on how the model is performing. Some of them are:

- Accuracy
- Recall
- F1 score
- Precision

  (these are discussed in evaluation metrics (4.6)

WORKING OF ARCHITECTURE:

Initially consider a dataset of columns like web news text and category etc..

- The above data set undergoes preprocessing such as
  - Checking null values, Category balancing.
- Tokenization is performed for
  - Removal of punctuations, special characters like '\n' , '\r', '-', etc
  - Removal of stop words.
  - Performing Text Normalization using Lemmatization.
- Now label encoding is performed.

- Apply Tf-idf vectorizer to each and every word of independent feature such that for every unique word in our document a vector of float type is assigned. Now these Tf-idf values of each token is given as input to the classifiers.
- Data is split into training and test data.
- Now the below Models are built, trained with training data.
    - SVM
    - KNN
    - LSTM
- Predict the results for our test data using above models.
- Finally identifying the efficient model .

4.3. UML DESIGN

UML:

The Unified Modeling Language (UML) is a general purpose visual modeling language that is used to specify, visualize, construct, and document the artifacts of a software system. It captures decisions and understanding about systems that must be constructed. It is used to understand, design, browse, configure, maintain, and control information about such systems. The Unified Modeling Language is very important parts of developing object oriented software and the software development process. The Unified Modeling Language uses mostly graphical notations to express the design of software projects. Using the Unified Modeling Language helps project teams communicate, explore potential designs, and validate the architectural design of the software. The primary goals in the design of the Unified Modeling Language are:

• Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models.

• Provide extensibility and specialization mechanisms to extend the core concepts.

• Be independent of particular programming languages and development processes.

• Provide a formal basis for understanding the modeling language.

• Encourage the growth of the object oriented tools market.

• Support higher-level development concepts such as collaborations, frameworks, patterns, and components.

Each Unified Modeling Language diagram is designed to let developers and customer view a software system from a different perspective and invarying degrees of abstraction.

UML diagrams commonly created in visual modeling tools include.

• Use Case Diagram
• Class Diagram
• Sequence Diagram
• Collaboration Diagram
• State chart Diagram
• Activity Diagram.

The software that is used to draw the above diagrams is Rational Rose.

RATIONAL ROSE: Rational Rose is an object-oriented Unified Modelling Language (UML) software design tool intended for visual modelling and component construction of enterprise-level software applications. In much the same way a theatrical director blocks out a play, a software designer uses Rational Rose to visually create (model) the framework for an application by blocking out classes with actors (stick figures), use case elements (ovals), objects (rectangles) and messages/relationships (arrows) in a sequence diagram using drag-and-drop symbols. Rational Rose documents the diagram as it is being constructed and then generates code in the designer's choice of C++, Visual Basic, Java, Oracle8, or Data Definition Language.

UseCase Model

USECASE:

A use case diagram shows a set of use cases and actors and their relationships. Use case diagrams commonly contain

• Use cases

• Actors

• Dependency, generalization and association relationships

Use case

A use case describes a set of sequences, in which each sequence represents the interaction of the things outside the system (its actors) with the system itself.

Fig:Use case

Actor

An actor represents a coherent set of roles that users of use cases play when interacting with these use cases. Actors can be human or they can be automated systems.

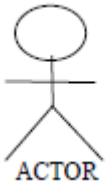


Fig: Actor

Dependency

A dependency is a semantic relationship between two things in which a change to one thing (the independent thing) may affect the semantics of the other thing (the dependent thing).

Fig : Dependency

Generalization

A generalization is a specialization/generalization relationship in which objects of the specialized element (the child) are substitutable for objects of the generalized element (the parent). With this the child shares the structure and the behavior of the parent.

Fig: Generalization

4.3.1 UseCase Diagram:

A use case diagram in the UML is a type of behavior diagram defined by and created from a use-case analysis. The purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those two use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

Identification of Actors : Actors represent system users. They help delimit the system and give a clearer picture of what the system should do. It is important to note that an actor interacts with, but has no control over the use cases.
An actor is someone or something that:

- Interacts with or uses the system.
- Provides input to and receives information from the system.
- Is external to the system and has no control over the use cases.

An actor can be represented as shown below:
1. A class rectangle with the label
2. The stick figure with the name of the actor below.

Identification of UseCases:

In its simplest form, a use case can be described as a specific way of using the system from a user's (actors) perspective . A more detailed description might characterize a use case as:

- A pattern of behavior the system exhibits.
- A sequence of related transactions performed by an actor in the system.
- Delivering something of value to the actor.

Use cases provide a means to:

- Capture system requirements.
- Communicate with the end users and domain experts.
- Test the system.

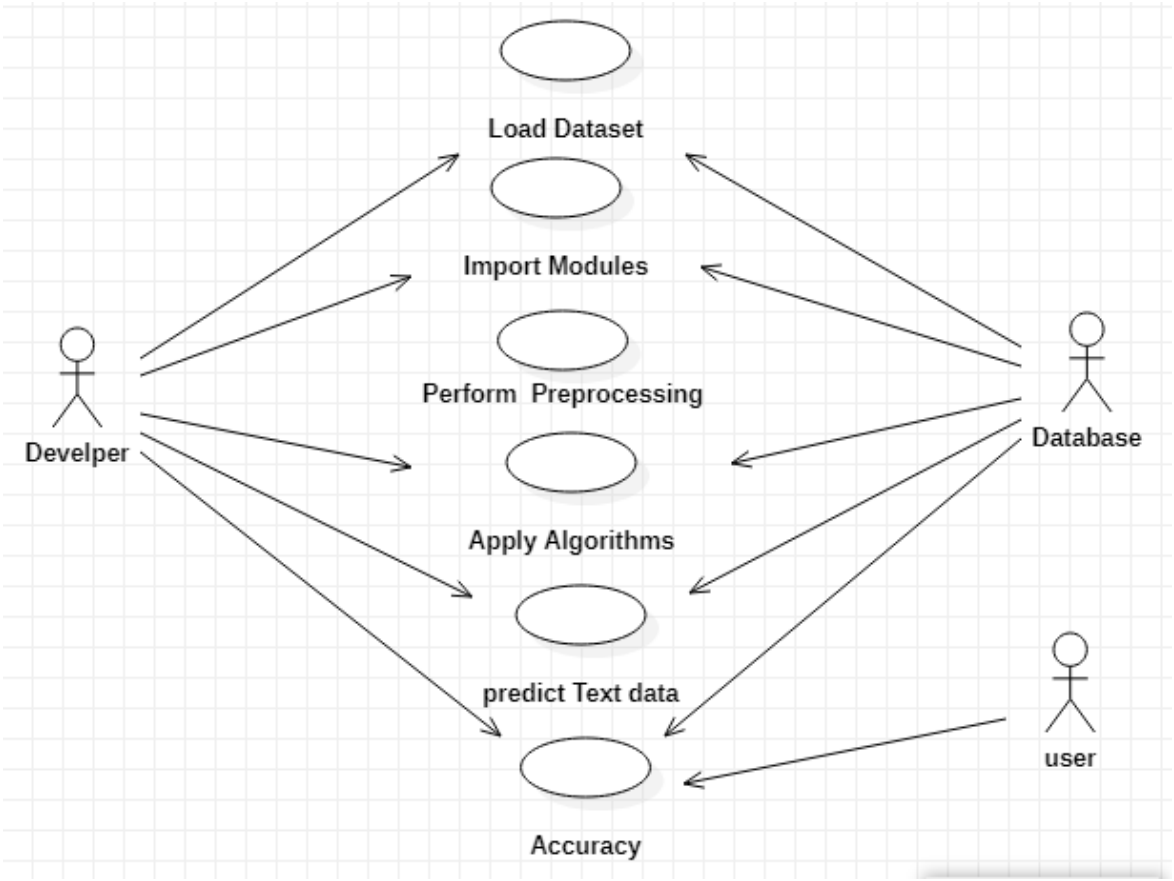


Fig 4.3.1.1:Usecase diagram for classification of web news

4.3.2 Sequence Diagram

A Sequence diagram is a graphical view of a scenario that shows interaction in a time-based sequence what happens first, what happens next. These diagrams establish the roles of objects and help to provide essential information to determine interfaces and class responsibilities.
The following are the tools located on the sequence diagram toolbox that enable to model sequence diagrams:

Message Icons:

A message icon represents the communication between the objects that indicates an action will follow. The message icon is a solid, horizontal arrow connecting two life lines together.

Focus of control:

Focus of Control (FOC) is an advanced technique that enhances sequence diagrams. It represents the period of time during which an object is performing an action, either directly or through an underlying procedure.

Message to Self:

A message to self is a tool that sends a message from one object back to the same object. It does not involve other objects because the message returns to the same object. The sender of the message is the same as the receiver.

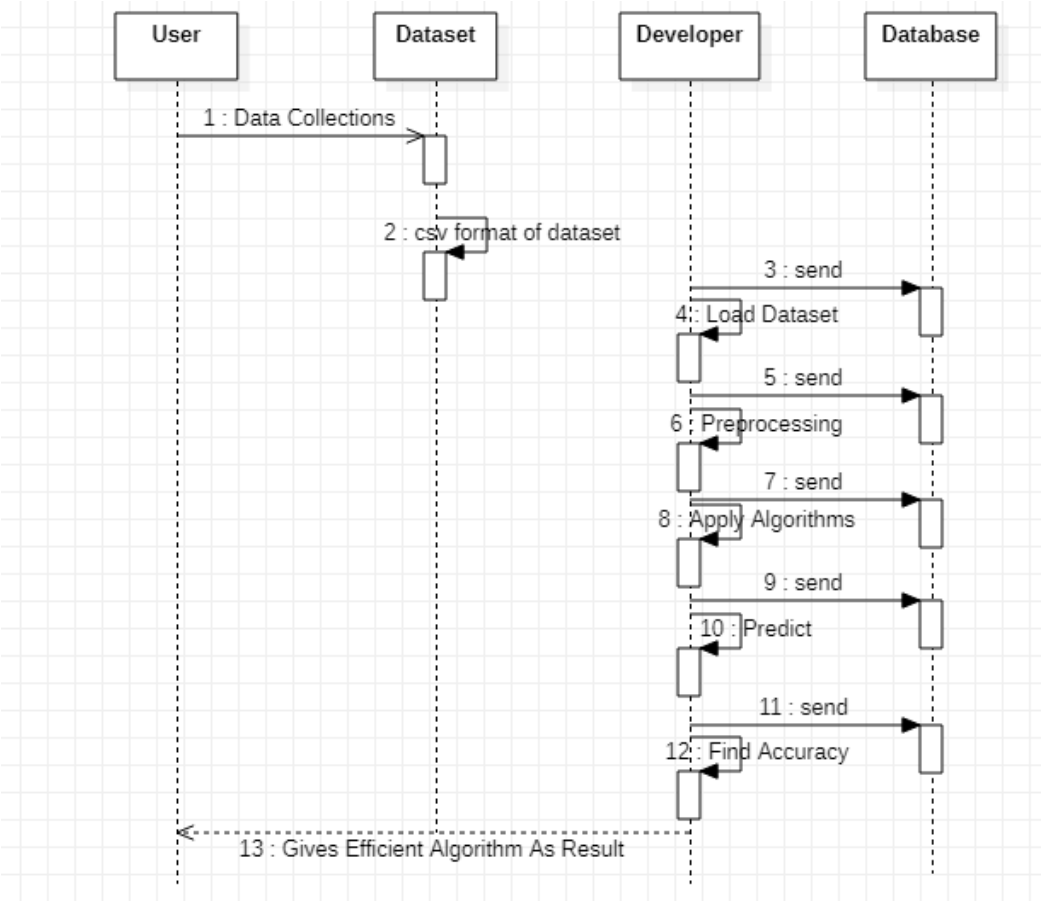


Fig 4.3.2.1:Sequence Diagram for classification of web news

4.3.3 State Chart Diagram

State chart diagram model the dynamic behavior of individual classes or any other kind of object. They show sequence of state that an object goes through, the events that cause transition from one state to another state and the actions that result from a state change.

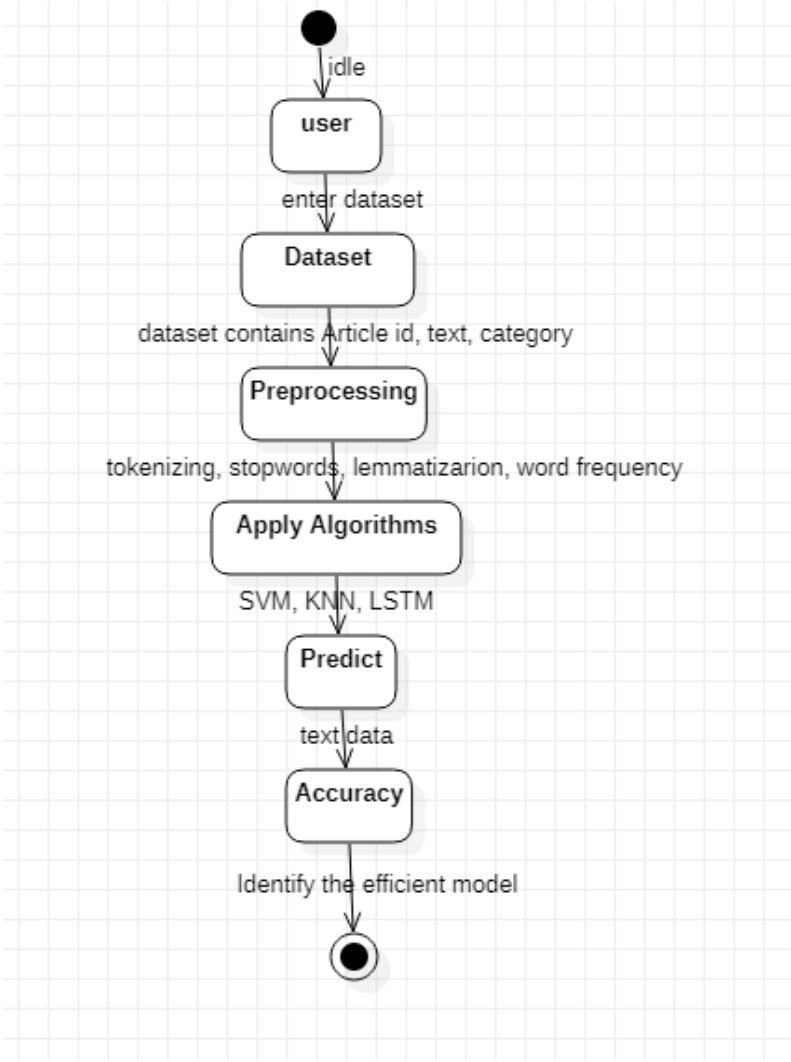


Fig 4.3.3.1:StateChart Diagram for classification of web news

4.3.4 Activity Diagram

Activity diagrams provide a way to model the workflow of a business process. An activity diagram is used for modeling the sequence of workflows. Activity diagrams models a software development process in a software company. The following are the tools used on the activity diagram toolbox to model activity diagrams:

Decisions

A decision represents a specific location in the activity diagram where the workflow may branch based upon guard conditions.

Synchronizations

These are visually defining forks and join represents parallel workflow.

Forks and Joins

A fork is used to model a single flow of control that divides into two or more separate and simultaneous flows. A join has two or more flows of control that unite into a single flow of control.

States

A state represents a condition during the life of an object during which it satisfies some condition or waits for some event.

Transitions

A state transition indicates an object in the source state that perform certain specified actions and enter the destination state when a specified event occurs or when certain conditions are satisfied.

Start States

A start state (initial state) explicitly shows the beginning of a workflow.

End States

An end state(last) represents a final state or terminal state.

Swim lane

A unique activity diagram feature that defines who or what is responsible for carrying out the activity or state.

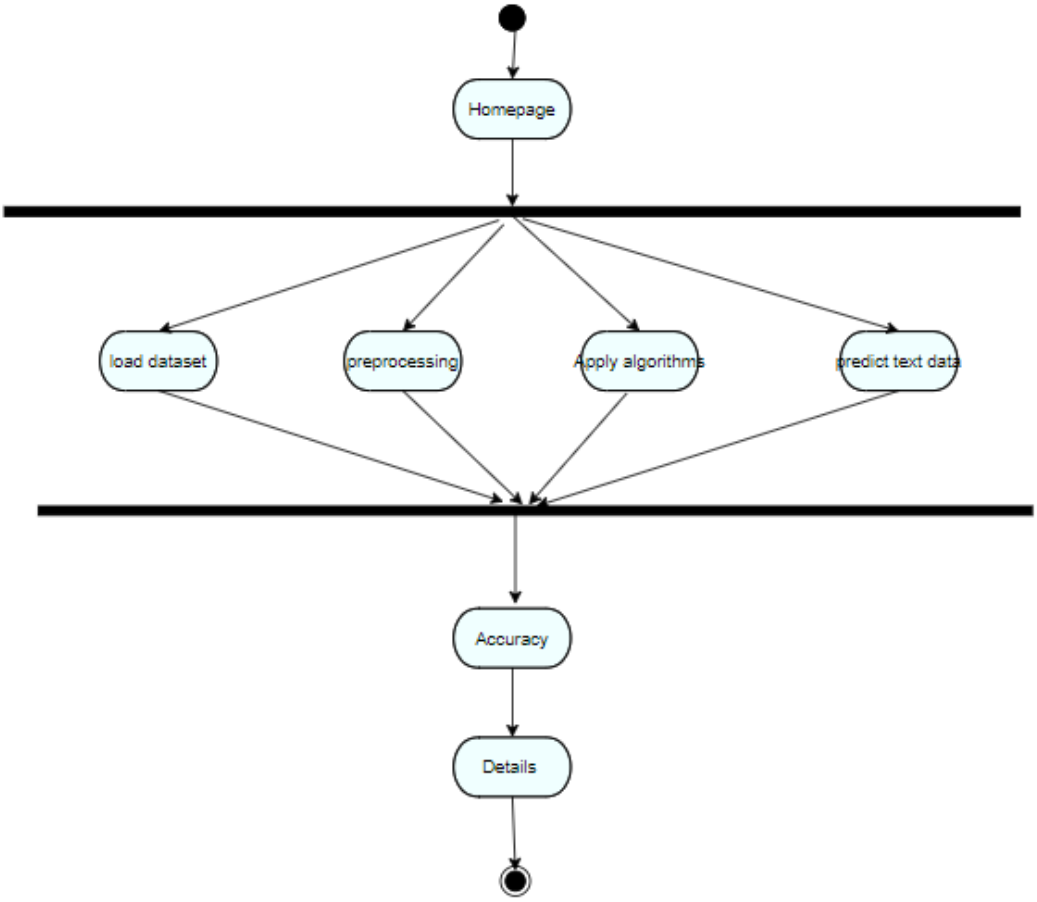


Fig 4.3.4.1:Activity Diagram for classification of web news

4.3.5 Class Diagram

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

1. The top part contains the name of the class. It is printed in Bold, centered and the first letter capitalized.

2.The middle part contains the attributes of the class. They are left aligned and the first letter is lower case.

3.The bottom part gives the methods or operations the class can take or undertake. They are also left aligned and the first letter is lower case.

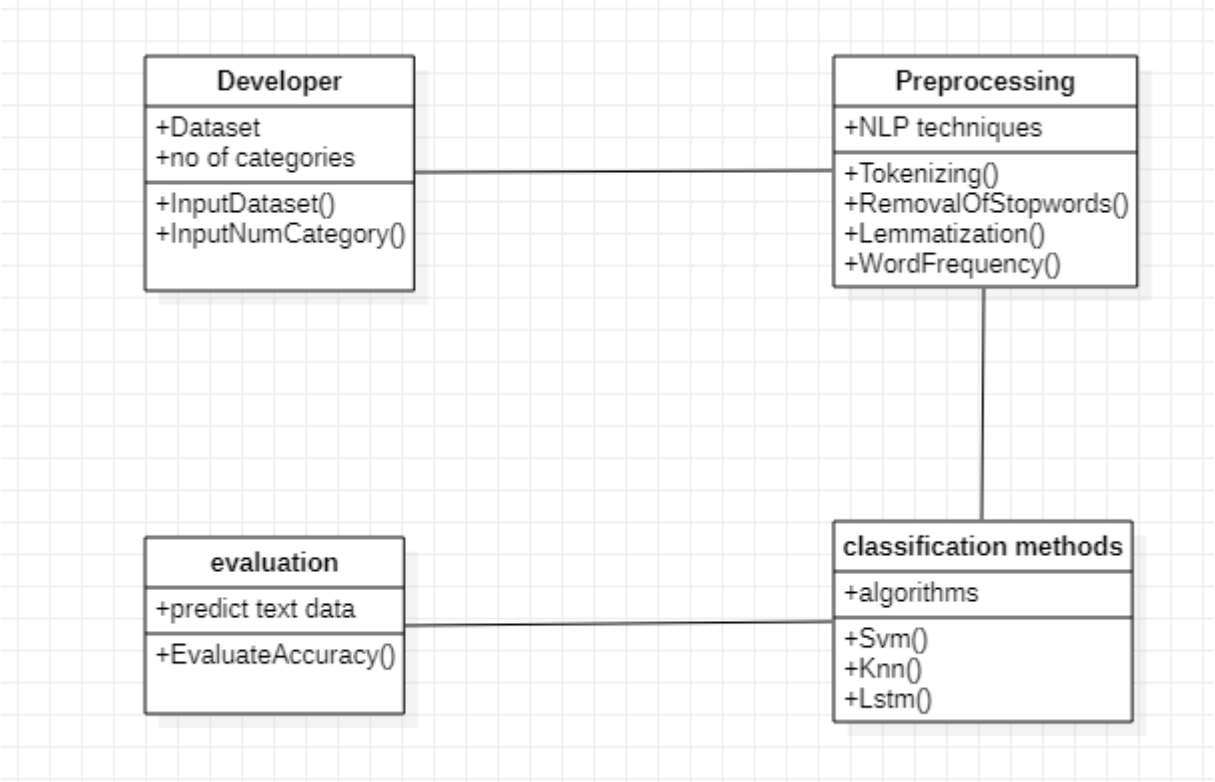


Fig 4.3.5.1:Class Diagram for classification of web news

4.3.6 Component Diagram

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

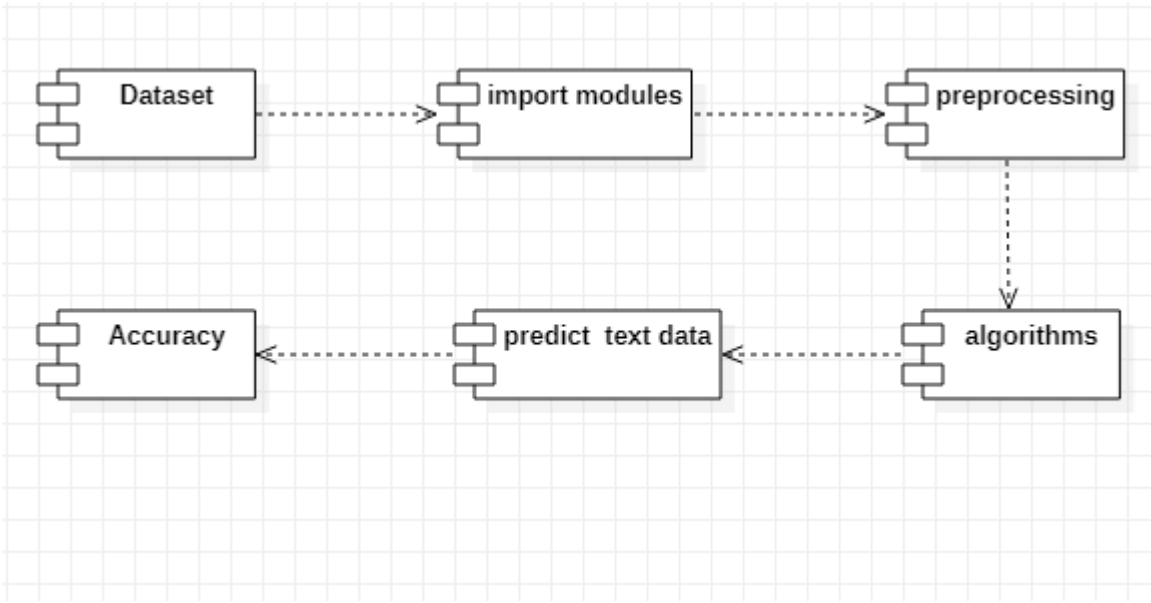


Fig 4.3.6.1:Component Diagram for classification of web news

4.4 Datasets

The dataset contains the data about the news articles .These data were collected from different websites. In the data set, present study considers five different types of news namely Business, Entertainment, Technology ,Sports and Politics. All these news data are stored in text file (.csv)with specific domain . Final dataset consist of total 1490 news distributes shown in Table.Here we are considering structured data, Other data like images, videos, and text, so-called unstructured data is not considered at this time.

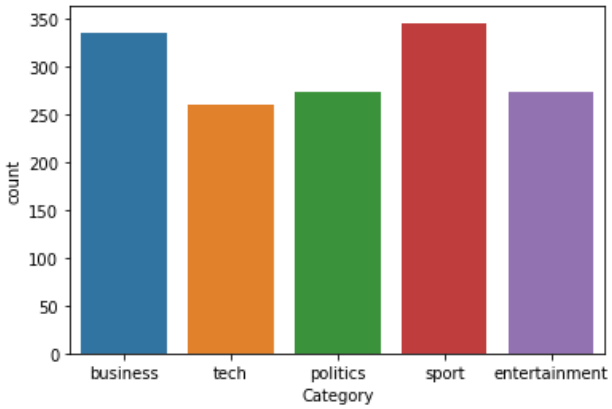
Because It is important to have good grasp of input data.

Training Dataset : A dataset that we feed into our machine learning algorithm to train our model.

Testing Dataset: A dataset that we use to validate the accuracy of our model but is not used to train the model. It may be called the validation dataset.

The fields(Features) that are present in the dataset are Article Id,Text and Category. Article id is same as page number or serial number of the text.Text represents paragraphs that belongs to any categories(business,technology etc).and Category represents ,for which category that the text belongs to. Fig() represents the count of the three features and Fig() for the graphical representation of the Category in the dataset as shown below.

Category	ArticleId	Text
business	336	336
entertainment	273	273
politics	274	274
sport	346	346
tech	261	261



4.5. Evaluation Metrics

The performance measure provides the efficiency, accuracy and reliability of a training model. Many factors that contributes to the performance of classifier and model. Some of those we consider four parameters precision,recall,f1-score and accuracy for SVM, kNN and LSTM

Metrics for Classification:

Precision:-

It can be defined as the probability of predicted target value with true value using the classifier. It also defines the effectiveness of classifier. It can be formulated as mentioned below

Precision P=TP/(TP+FP)

Recall:-

It can be defines as the probability of actual label of the class. It can be formulated as mentioned below.

Recall R=TP/(TP+FN)

F1-Score:-

It is a measure to define the overall performance of the classifier. It is evaluated from the precision and recall values as mentioned in below.

F1-Score F= 2*P*R/(P+R)

Accuracy:-

It is the ratio of number of correct predictions to the total number of input samples.

Sensitivity:-

Sensitivity is a measure of the proportion of actual positive cases that got predicted as positive.

Specificity:-

Specificity is defined as the proportion of actual negatives, which got predicted as the negative.

$$accuracy = \frac{\sum (Y_{pre} = Y_{test})}{Y_{test}} * 100$$

$$sensitivity = \frac{TP}{TP + FN}$$

$$specificity = \frac{TN}{TN + FP}$$

In this experiment, the confusion matrix is used to demonstrate the preferred result against the predicted result. Each column of the matrix represents samples in the predicted class while each row represents samples in the actual class. Receiver operating characteristics (ROC) curves are also used throughout the article to demonstrate a true positive rate against the false positive rate at various thresholds. The dataset is being organized into four classes instead of two so in order to use ROC curves the one verse all approach is used. The area under a ROC curve (AUC) and accuracy of a classification technique are good indicators for comparison.

5.TESTING

5.1 INTRODUCTION:

Introduction to testing:

Testing is a fault detection technique that tries to create failure and erroneous states in a planned way. This allows the developer to detect failures in the system before it is released to the customer. Note that this definition of testing implies that a successful test is test that identifies faults. We will use this definition throughout the definition phase. Another often used definition of testing is that it demonstrates that faults are not present.

Testing can be done in two ways:

1.Top down approach
2.Bottom up approach

1.Top down approach:

This type of testing starts from upper level modules. Since the detailed activities usually performed in the lower level routines are not provided stubs are written.

2.Bottom up Approach:

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded within the larger system. In this project, bottom up approach is used where the lower level modules are tested first and the next ones having much data in them.

5.2 TESTING METHODOLOGIES

The following are the Testing Methodologies:

• Unit Testing :

Unit testing method considers a module as single unit and checks every unit at interfaces and communicates with other modules rather than getting into details at statement level. The module here will be treated as a black box, which will take some input and generate output. Outputs for given set of input combination are pre-calculated and are generated by this module.

• Integration Testing:

Testing is a major quality control measure employed at the phase of software development. It is basic function to detect errors. Sub functions, when combined may not produce than it desired. Global data structures can represent the problems. Integrated testing is a systematic technique for constructing program structure while conducting the tests. To uncover errors that are associated with the interfacing objective is to make unit test modules and built a program structure that has been detected by the design. In a non-incremental integration, all the modules are combined in advance and the program is tested as a whole. Errors will appear in an endless loop function in this case. In incremental testing the program is constructed and tested in small segments where the errors are isolated and corrected.
Different incremental integration strategies are bottom – up integration, top – down integration, regression testing.

• Usability Testing:

Usability testing is a testing method that measures an application's ease-of-use from the end-user perspective and is often performed during the system or acceptance testing stages. The goal is to determine whether or not the visible design and aesthetics of an application meet the intended workflow for various processes, such as logging into an application. Usability testing is a great way for teams to review separate functions, or the system as a whole, is intuitive to use.

• System Testing:

In system testing, a system as a whole i.e. a completely developed system is tested to find defects. All the modules are integrated to test as a complete system. Scenarios prepared for system testing are run to find the defects.

All the modules of a product are integrated and tested to verify that the built product is as per the customer requirement. End-to-end testing is performed to cover the complete scenarios.

5.3 TEST CASES

Test Case 1:- In this test, we upload a valid .csv data file and print the dataset

Test Id	Test Scenario	Test Case	Test Data	Test Steps	Expected Result	Actual Result	Status
1	Verify the dataset	Uploading valid Dataset	Valid Dataset	Upload respective .csv data file	No error	Dataset is Printed	Pass

Table(5.3.1) Test case 1

Test Case 2:- In this test, we upload an invalid data file other than .csv data file and print the dataset

Test Id	Test Scenario	Test Case	Test Data	Test Steps	Expected Result	Actual Result	Status
2	Verify Dataset	Uploading Dataset	Invalid Dataset	Upload another file	Has to show error	Invalid format Exception	Fail

Table(5.3.2)Test case 2

Data Preprocessing:

Test Case 3:-In this test, we test whether the stopwords are removed from the data or not

Test Id	Test Scenario	Test Case	Test Data	Test Steps	Expected Result	Actual Result	Status
3	Preprocessing	Remove the Stopwords	Uploaded dataset of news arcticles	Import stopwords from NLTK	Has to show no error	Shows the test without stopwords	pass

Table(5.3.3)Test case 3

Feature Extraction:

Test Case 4:- In this test,the feature vector generation is done

Test Id	Test Scenario	Test Case	Test Data	Test Steps	Expected Result	Actual Result	Status
4	Feature vector generation	Using Count vectorization and TFIDF transformer	Uploaded dataset of news article	Import TFIDF transformer	Has to convert each and every word to vector	Converted each and every word to vector form of float value	Pass

Table(5.3.4)Test case 4

Test Case 5:- In this test, testing whether the classifier is correctly classified or not is checked

Test Id	Test Scenario	Test Case	Test Data	Test Steps	Expected Result	Actual Result	Status

5	Perform classification	Perform the SVM classification	The feature vectors that are generated for each and every article	Using SVM	Get the label of the news article	Returns the correct label of the news	Pass

Table(5.3.5)Test case 5

Test Id	Test Scenario	Test Case	Test Data	Test Steps	Expected Result	Actual Result	Status
6	Perform classification	Perform the SVM classification	The feature vectors that are generated for each and every article	Using SVM	Get the label of the news article	Returns the wrong label of the news	fail

Table(5.3.6)Test case 6

Test Case 7:- In this test, we test the initialize_K nearest neighbors function of kNN by supplying valid dataset and k value.

Test Id	Test Scenario	Test Case	Test Data	Test Steps	Expected Result	Actual Result	Status

7	Perform Classification	Initializing random values for k	Dataset and valid number of neighbors(k)	Execute initializing random number of neighbors function	Selection of random number of neighbor	K nearest neighbors are selected	pass

Table(5.3.7)Test case 7

6.RESULTS

6.1 Actual Results

Table 6.1.1 Results of different classifiers

ALGORITHM	ACCURACY (In percent)	PRECISION (In Percent)	RECALL (In Percent)	F1-SCORE (In Percent)
KNN	91.15	91.15	91.15	91.15
SVM	93.02	93.02	93.02	93.02
LSTM	97.31	97.31	97.31	97.31

Out of all the classifiers implemented LSTM coming out with the best accuracy of 97.31 % and KNN coming out the least at 91.15% of accuracy.

6.2 Analysis of the Results obtained

Confusion Matrix:

A **confusion matrix** is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.

ROC Curve:

An **ROC curve** (**receiver operating characteristic curve**) is a **graph** showing the performance of a classification model at all classification thresholds.

This **curve** plots two parameters:

1.True Positive Rate

2. False Positive Rate

6.2.1 Analysis of Results for SVM

Confusion Matrix for SVM

```
[[75  0  2  1  0]
 [ 3 65  3  2  1]
 [ 2  1 58  0  1]
 [ 2  2  0 92  0]
 [ 1  2  2  1 57]]
```

Fig 6.2.1.1:Confusion Matrix for SVM

ROC Curve for SVM

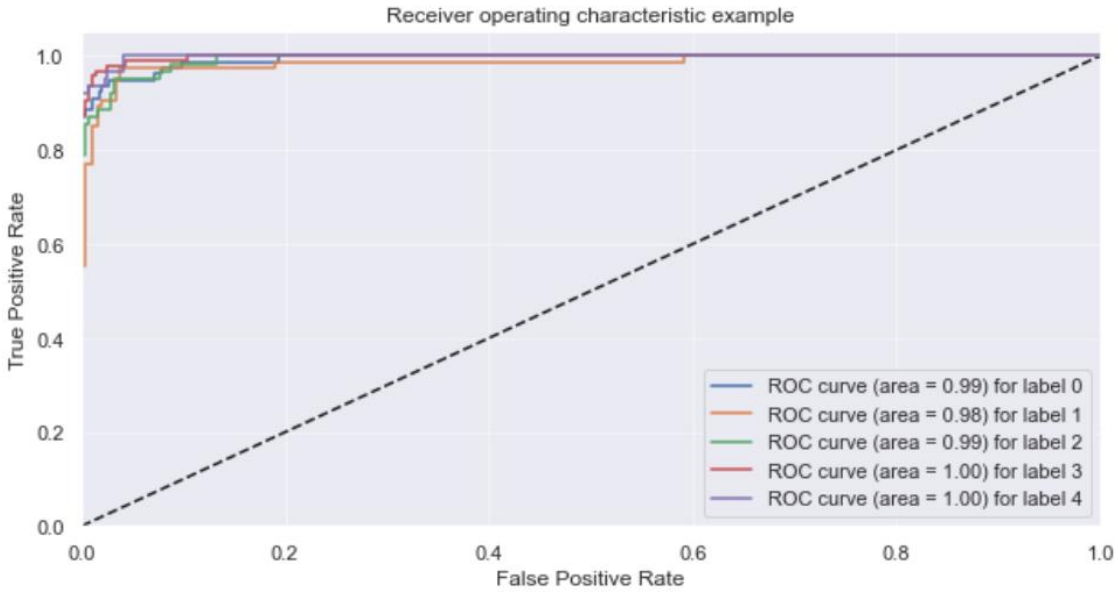


Fig 6.2.1.2:ROC curve for SVM

6.2.2 Analysis of Results for KNN

Confusion Matrix for SVM

```
[[75  0  2  1  0]
 [ 3 65  3  2  1]
 [ 2  1 58  0  1]
 [ 2  2  0 92  0]
 [ 1  2  2  1 57]]
```

Fig 6.2.2.1:Confusion Matrix for KNN

ROC Curve for KNN

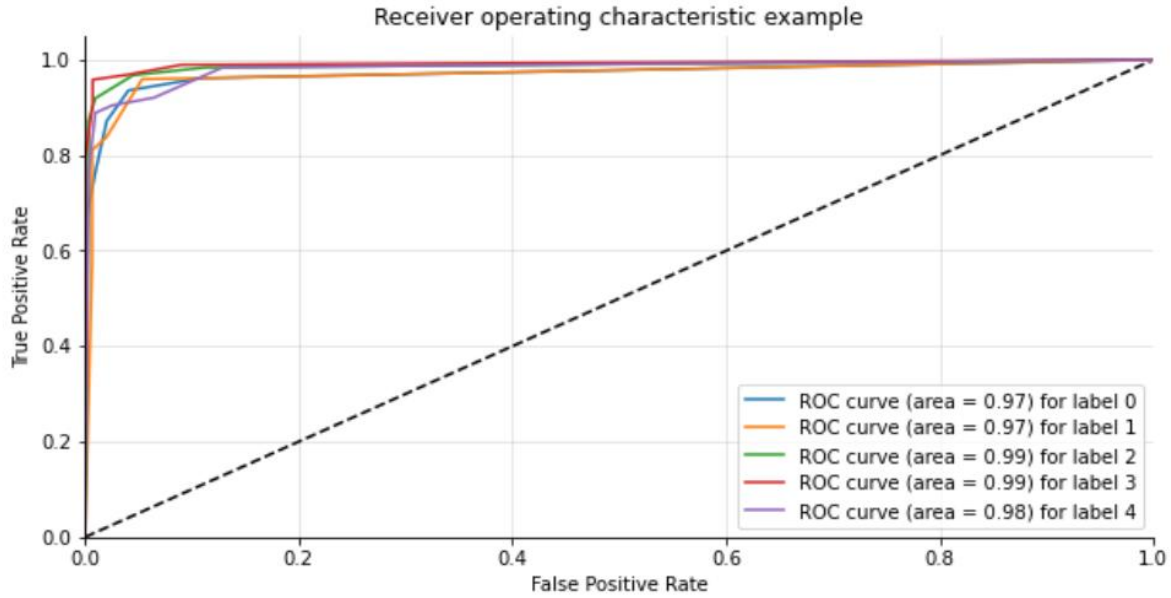


Fig 6.2.2.2 :ROC curve for KNN

6.2.3 Analysis of Results for LSTM

ROC Curve for LSTM

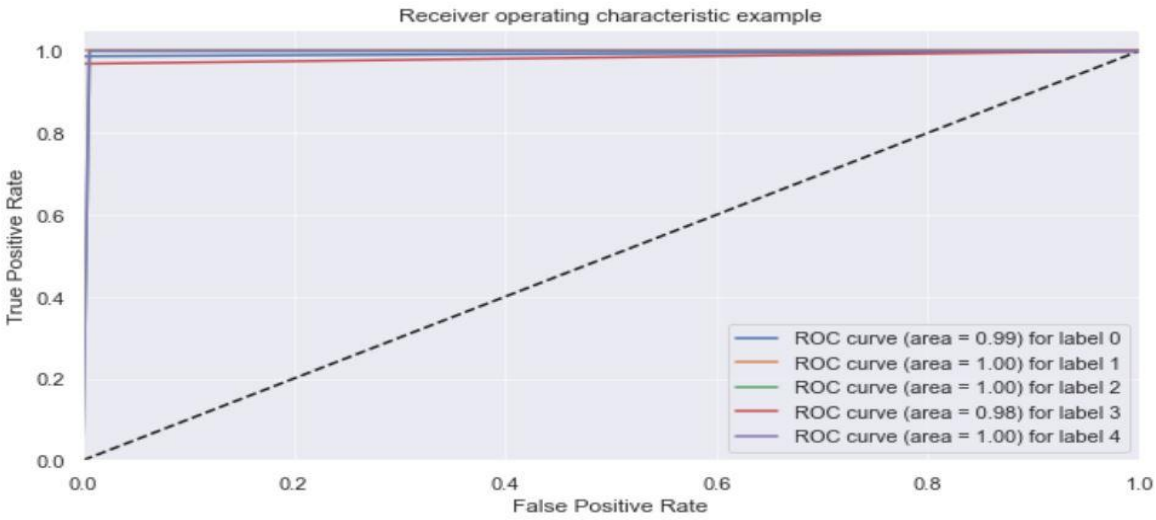


Fig 6.2.3.1:ROC curve for LSTM

7. CONCLUSION AND FUTURE WORK

This project presented the classification of web news in a website using the applications of ML algorithms. Almost all the three ML models gave good accuracy values but the most accurate one is LSTM model.

In future our project will extract videos from web such as movies,videos etc. Also user can get live updated news from internet. User will be added his own interested news and share that news with another user. User will be personalised his own category as per his demand and provide more security for each user. And also we will try to use regional languages which are user friendly.

REFERENCES

[1]https://www.academia.edu/2038312/Fast_webpage_classification_using_features

[2] Surajit Chandhuri, Umeshwar Dyal,“ An overview of data warehousing and OLAP technologies” Published in ACM Sigmod record, Vol. 26, Issue 1, pp. 65-74, USA, 1997

[3] Thorsten Joachims,“Text Categorization With Support Vector Machines: Learning with many relevant features,” Published in: European Conference on Machine Learning (ECML 1998), Lecture notes in Computer Science Book series Springer Publications, Vol. 1398, pp. 137-139, 1998

[4] Hyeran Byun Pattern Recognition with Support Vector Machines, First International Workshop, SVM 2002, Niagara Falls, Canada, August 10, 2002, Proceedings

[5] Chee-Hong Chan Aixin Sun Ee-Peng Lim,“Automated Online News Classification with Personalization,” 4th International Conference on Asian Digital Libraries (ICADL2001), Bangalore, pp. 320-329, 2001

[6] Daniel I. Morariu, Lucian N. Vintan, Volker Tresp, “MetaClassification using SVM Classifiers for Text Documents”, World Academy of Science, Engineering and Technology 21, pp. 15-20, 2006

[7]D. Morariu, R. Cre¸tulescu, L. Vin¸tan,“Improving a SVM Meta-classifier for Text Documents by using Naïve-Bayes,” Int. J. of Computers, Communications & Control, Vol. 3, pp. 351-361, 2010.

[8] Mr. S.P Deshpande, Dr. V.M Thakre,"Data mining system and Applications: A review,” In International Journal of Distributed and Parallel System (IJDPS), Issue 1, Vol. 1, pp. 32-44, 2010

[9] Mita K. Dalal, Mukesh A.Zaveri,“Automatic text classification: A technical review,” In International Journal of Computer Applications, Vol. 28, Issue 2, pp. 37-40, 2011.

[10] Yiming Yang, Xin Liu,"A re-examination of text categorization methods,” Caenegie Mellon University Pittsburg, PA 15213- 3702, USA, Vol. 18, Issue 2, 2012.

[11] Rama Bharath Kumar, Bangari Shravan Kumar, Chandragiri Shiva Sai Prasad,“Financial news classification using SVM,” International Journal of Scientific and Research Publications, Vol. 2, Issue 3, pp. 1-6, 2012 45

#SOURCE CODE

1) Checking for NULL Values

```
data.info()

data.isnull().any()
```

2) Checking Data Balance using Count of Each Category

```
sns.countplot(data["Category"])

'"# can do by using groupby tooo

data.groupby('Category').count()

#FREQUENCY OF WORDS

def create_wordcloud(single_text):

    wordcloud = WordCloud(font_path="C:\my_fonts/ITCBLKAD.TTF", width=800, height=500, random_state=21, max_font_size=110).generate(single_text)  # generate will convert text into wordcloud

    plt.figure(figsize=(200, 70)) #adjusting fig and it return object

    plt.imshow(wordcloud)    # above object is printed

    plt.axis('off')

  #plt.figure(figsize=(10, 7))

    #plt.imshow(wordcloud, interpolation="bilinear")

    #plt.axis('off')

    #plt.show()

def create_wordcloud(words):

    wordcloud = WordCloud( width=800, height=500, random_state=21, max_font_size=130).generate(words)

    plt.figure(figsize=(10, 7))
```

```
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis('off')
    plt.show()


subset=data[data.Category=="tech"]
text=subset.Text.values
words =" ".join(text)
create_wordcloud(words)


subset=data[data.Category=="business"]
texts=subset.Text.values
single_text=" ".join(texts)
create_wordcloud(single_text)
#3) Removal of '\n', '\r'  etc
# Removal of stopwords
# Text Normalizing using Lemmatization
def process_text(text):
    text = text.lower().replace('\n',' ').replace('\r','').strip()
    text = re.sub(' +', ' ', text)
    text = re.sub(r'[^\w\s]','',text)
   stop_words = set(stopwords.words('english'))
    word_tokens = word_tokenize(text)
    #filtered_sentence = [w for w in word_tokens if not( w in stop_words)]
    filtered_sentence = []
```

```
    for w in word_tokens:
        if w not in stop_words:
            filtered_sentence.append(w)
   lemma_word=[]
    for w in filtered_sentence:
        word1 = wordnet_lemmatizer.lemmatize(w, pos = "n")
 # Transformation based tagging is applied here (i.e all POS are applied and modifeid until no modification is requiered .... it is combo of rule based and stochastic tagging)
        word2 = wordnet_lemmatizer.lemmatize(word1, pos = "v")
        word3 = wordnet_lemmatizer.lemmatize(word2, pos = ("a"))
        lemma_word.append(word3)
       text = " ".join(lemma_word)
    return text
#downloading wordnet package
nltk.download('wordnet')
from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()
data['Text_parsed'] = data['Text'].apply(process_text)
print(data.head(5))
```

#Label Encoding

```
"""from sklearn.preprocessing import LabelEncoder
data['Category_target']= LabelEncoder().fit_transform(y=data['Category'])"""
#or can write as
```

```
from sklearn import preprocessing
label_encoder = preprocessing.LabelEncoder()
data['Category_target']= label_encoder.fit_transform(data['Category'])
```

```
#SPLITTING DATA INTO TRAINING AND TESTING
import random
from sklearn.model_selection import train_test_split
ran_val=random.randint(3,12)
print(ran_val)
X_train, X_test, y_train, y_test = train_test_split(data['Text_parsed'],
                                                    data['Category_target'],
                                                    test_size=0.25,
                                                    random_state=10)
print(X_train)
print(y_test)
```

TF-IDF

tf-idf = log(word_count_in_particular_doc)* log(total no of docs / no of docs that has word)

as the tf-idf value increases - the word is more relavant to that particular doc.

```
#ACTUAL IMPLEMENTATION OF TFIDF
ngram_range = (1,2)
min_df = 10
max_df = 1.
max_features = 300
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf = TfidfVectorizer(encoding='utf-8',
                        ngram_range=ngram_range,
                        stop_words=None,
                        lowercase=False,
                        max_df=max_df,
                        min_df=min_df,
                        max_features=max_features,
                        norm='l2',
                        sublinear_tf=True)

features_train = tfidf.fit_transform(X_train).toarray()
labels_train = y_train
print(features_train)
features_test = tfidf.transform(X_test).toarray()
labels_test = y_test
print(features_test.shape)
```

SVM

```
#for confusion matrix,accuracy,precision,recall,f1 score
bifrom sklearn.model_selection import train_test_split
#SVM classification
from sklearn import svm
import numpy as np
import seaborn as sns
```

```
import matplotlib.pyplot as plt

from sklearn.metrics import confusion_matrix

from sklearn.metrics import roc_curve, auc

from sklearn.preprocessing import label_narize

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

'''X_train, X_test, y_train, y_test = train_test_split(X_train_tfidf, training_data.flag,
test_size=0.25, random_state=42)

y_test2=y_test

y_test3=y_test'''

clf_svm = svm.LinearSVC()

clf_svm.fit(features_train,labels_train)

predicted = clf_svm.predict(features_test)

confusion_mat = confusion_matrix(labels_test,predicted)

print(confusion_mat)

print("accuracy_score:",accuracy_score(labels_test, predicted))

print("precision_score:",precision_score(labels_test, predicted, average='micro'))

print("recall_score:",recall_score(labels_test,predicted,average="micro"))

print("f1_score:",f1_score(labels_test,predicted,average="micro"))


#.....For Plotting ROC Curve…..

labels_score = clf_svm.fit(features_train, labels_train).decision_function(features_test)

#y_test = label_binarize(y_test, classes=[0, 1, 2,3,4])

fpr = dict()

tpr = dict()

roc_auc = dict()
```

```
target_list=[0,1,2,3,4]
labels_test_dummies = pd.get_dummies(labels_test, drop_first=False).values
n_classes=5
for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(labels_test_dummies[:, i], labels_score[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

# roc for each class
figsize=(10,5)
fig, ax = plt.subplots(figsize=figsize)
ax.plot([0, 1], [0, 1], 'k--')
ax.set_xlim([0.0, 1.0])
ax.set_ylim([0.0, 1.05])
ax.set_xlabel('False Positive Rate')
ax.set_ylabel('True Positive Rate')
ax.set_title('Receiver operating characteristic example')
for i in range(n_classes):
    ax.plot(fpr[i], tpr[i], label='ROC curve (area = %0.2f) for label %s' % (roc_auc[i],
target_list[i]))
ax.legend(loc="best")
ax.grid(alpha=.4)
sns.despine()
plt.show()
```

KNN

```
#for confusion matrix,accuracy,precision,recall,f1 score
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=5)
classifier.fit(features_train, labels_train)
labels_pred = classifier.predict(features_test)
print(confusion_mat)
print("accuracy_score:",accuracy_score(labels_test, labels_pred))
print("precision_score:",precision_score(labels_test, labels_pred, average='micro'))
print("recall_score:",recall_score(labels_test,labels_pred,average="micro"))
print("f1_score:",f1_score(labels_test,labels_pred,average="micro"))

#...For Plotting ROC Curve…
labels_score = classifier.fit(features_train, labels_train).predict_proba(features_test)
#-----for plotting ROC curve-----
#y_test = label_binarize(y_test, classes=[0, 1, 2,3])
fpr = dict()
tpr = dict()
roc_auc = dict()
target_list=[0,1,2,3,4]
labels_test_dummies = pd.get_dummies(labels_test, drop_first=False).values
n_classes=5
for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(labels_test_dummies[:, i], labels_score[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])
```

```
# roc for each class
figsize=(10,5)
fig, ax = plt.subplots(figsize=figsize)
ax.plot([0, 1], [0, 1], 'k--')
ax.set_xlim([0.0, 1.0])
ax.set_ylim([0.0, 1.05])
ax.set_xlabel('False Positive Rate')
ax.set_ylabel('True Positive Rate')
ax.set_title('Receiver operating characteristic example')
for i in range(n_classes):
    ax.plot(fpr[i], tpr[i], label='ROC curve (area = %0.2f) for label %s' % (roc_auc[i], target_list[i]))
ax.legend(loc="best")
ax.grid(alpha=.4)
sns.despine()
plt.show()
```

LSTM

```
# For Checking accuracy,recall,precision,f1 score
yhat_probs = model.predict(features_test2, verbose=0)
# predict crisp classes for test set
yhat_classes = model.predict_classes(features_test2, verbose=0)
# reduce to 1d array
yhat_probs = yhat_probs[:, 0]
```

```
#yhat_classes = yhat_classes[:, 0]

import numpy as np

labels_test=np.argmax(labels_test2, axis=1)

# accuracy: (tp + tn) / (p + n)

accuracy = accuracy_score(labels_test, yhat_classes)

print('Accuracy_score: %f' % accuracy)

# precision tp / (tp + fp)

precision = precision_score(labels_test, yhat_classes,average="micro")

print('Precision_score: %f' % precision)

# recall: tp / (tp + fn)

recall = recall_score(labels_test, yhat_classes,average = "micro")

print('Recall_score: %f' % recall)

# f1: 2 tp / (2 tp + fp + fn)

f1 = f1_score(labels_test, yhat_classes,average ="micro")

print('F1 score: %f' % f1)


#.....For Plotting ROC Curve……

import numpy as np
from scipy import interp
import matplotlib.pyplot as plt
from itertools import cycle
from sklearn.metrics import roc_curve, auc

# Plot linewidth.

lw = 2
labels_score=yhat_classes
labels_score=label_binarize (labels_score, classes=[0, 1, 2,3,4])
```

```
# Compute ROC curve and ROC area for each class

fpr = dict()
tpr = dict()
roc_auc = dict()
#labels_test_dummies = pd.get_dummies(labels_test4, drop_first=False).values

n_classes=5
for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(labels_test_dummies[:, i], labels_score[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])


figsize=(10,5)
fig, ax = plt.subplots(figsize=figsize)
ax.plot([0, 1], [0, 1], 'k--')
ax.set_xlim([0.0, 1.0])
ax.set_ylim([0.0, 1.05])
ax.set_xlabel('False Positive Rate')
ax.set_ylabel('True Positive Rate')
ax.set_title('Receiver operating characteristic example')
for i in range(n_classes):
    ax.plot(fpr[i], tpr[i], label='ROC curve (area = %0.2f) for label %s' % (roc_auc[i],
target_list[i]))
ax.legend(loc="best")
ax.grid(alpha=.5)
sns.despine()
plt.show()
```