**Text Classification through NLP and Machine Learning Models**

***Abstract*** – Exponential growth of documents and their text, categorization all be formulated as text classification problems. Natural Language play key role with a complex problem “huge dimension of text or document features” to classify the documents and unstructured text into single class or multiple classes. After a major investigation of document classification and classification algorithms it get to know different steps to produce effective results like appropriate preprocessing, embedding methods and classification algorithms like Adaptive, Dynamic. Accuracy cannot be achieved good even multiple classifiers combined for productivity, so applying relative steps to result in more accuracy even for a single classifier by tuning parameters, selecting different classifiers and etc. It may be expensive and not feasible with time constraints of number of documents and huge text content involvement.

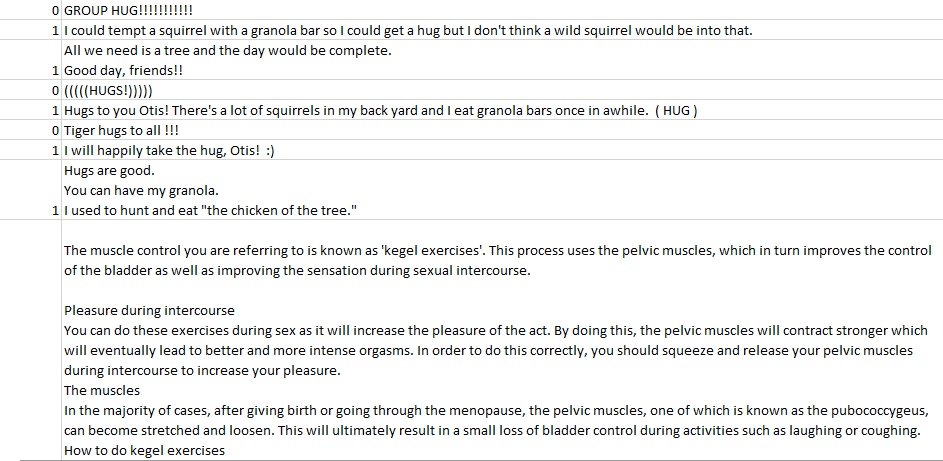
**1. INTRODUCTION**

Text classification has been very important and necessaire since when digital documentation comes into the modern world due to deal with very large amount of text documents and their data. Texts can be written in many categories like articles, reviews and advertisements. Text classification also includes topic based text classification but category classification differs from topic based one. Mostly data for category or genre classification is gathered/collected from web, they’re multi source and consequently resides in many formats and multiple vocabularies and often in different styles.

Initially text classification is task of classifying a document under a predefined category. More formally, if ‘d1’ is document under set of documents D and {c1, c2, c3,…, cn} is the set of all categories under the document ‘d1’ and text classification assigns one category to one document.

**2. DATA**

As in supervised machine learning task, an initial dataset is needed most importantly. Data can be in pure text format and also in impure format which requires some cleaning steps to make it pure.



Here total data is represented in text format which needs some preprocessing steps to make it pure and to feed the machine learning models but the models cannot work on text representation, to work with machine learning models so they require another set of representation which explained under Word Embedding.

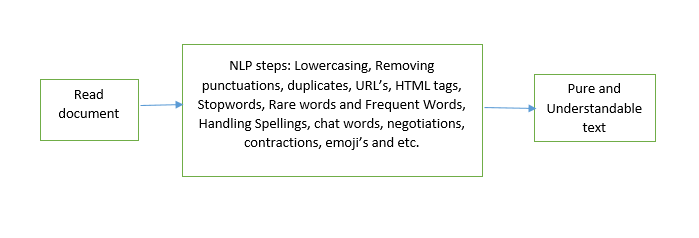
**3. Natural Language Processing (NLP)**

It refers to the field in machine learning with the ability of a computer to understand, analyze and manipulate human language.

In real life it can be used for many purposes like,

* Information Extraction
* Machine Translation
* Text Simplification & Summarization
* Sentiment Analysis

NLTK – popular open source package in python to preprocess the raw text to make it pure and understandable.



There are more preprocessing steps like Stemming, Lemmatization, Parts of speech tagging, chunking, Named Entity Recognition (NER) and etc.

**4. WORD EMBEDDING**

Word embedding’s are a type of word representation that allow text words with similar meaning but with a different type of representation like numeric. Individual words in text are represented as real-valued vectors often hundreds or thousands of dimensions in predefined vector space where each word mapped to one vector and the vector values are learned in the way that resembles neural network.

Words distributed representation is learned based on usage of words and where the words are considered as features for the machine learning model. These features can be selected based on text type and its way of transformation like using Principle Component Analysis (PCA) in order to reduce the feature dimension to make machine learning model work faster because the low number of features can be low execution time.

There are multiple word embedding techniques to convert the text words into vector space like,

* Bag of Words (BOW)
* Term Frequency and Inverse Document Frequency (TF-IDF)
* Word2Vec
* GloVe

**5. Machine Learning Models**

Post feature selection and transformation the documents can be easily represented in the form to feed machine learning algorithms. However there are several supervised learning algorithms like Decision tree, Random forest, Support vector machines, Naïve Bayes, KNN and etc, but initially we cannot predict which algorithm is going to produce better results. In order to overcome that kind of situation prefer to apply every supervised learning algorithm to check for better accuracy and results.

Decision tree shapes classification simulation in usage of tree structure. Main aim is to construct a model that predicts the value of target variable based on input variables. In these tree structures, always leaves characterize the class labels and the branches denotes features of class labels. It disturbs the dataset into small subsets while at identical time it the tree established incrementally and the ultimate result is a tree with decision nodes and leaf nodes. These decision trees can knob both categorical and numeric data with pre-defined conditions while constructing the tree by using Information Gain and Entropy Splitting while splitting the nodes.

Random forest is an ensemble learning technique consists of multiple decision tree, where decision tree is a single tree but random forest have multiple number of decision tree to improve the accuracy by using multiple trees.

Support vector machine (SVM) is an impressive approach for classifying the high dimensional data with use of hyperplanes. SVM has been conveyed as discriminative classifier that requires an optimal hyperplane to split the training data points from different classes by maximizing the classification margin. Similarly SVM applicable to non-linear decision surface data points by engaging a system identified as “kernel” method that designs input data to higher dimensional feature spaces where a linear hyperplane can be launch to classify the documents or text.

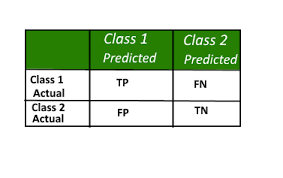
Naïve Bayes is often used for text classification applications and experiments because of its simplicity and effectiveness. However its performance is often degraded because it does not model the text as well like numeric data so to move for Multi-nominal Naïve Bayes classification which produce better results and accuracy with text as comparable with Naïve Bayes with numeric data. This model gives class labels to problem instances, represented as vectors of feature values. The value of particular feature is autonomous of the value of other feature. Prior probability is known as posterior probability is checked in Naïve Bayes technique which helps to assign the documents to particular categories.

KNN, the most commonly and widely used distance function for the kNN classifier is the Euclidean distance formula and it is used to calculate the distance between the new unlabeled data point and the training data points. The main step in the classification stage of the KNN is to measure the distance in order to identify the nearest neighbors of the new input data point.

**6. EVALUATION**

There are various methods to determine effectiveness, however confusion matrix, precision, recall, and accuracy are most often used. To determine these, one must first begin by understanding if the classification of a document was a true positive (TP), false positive (FP), true negative (TN), or false negative (FN).

Confusion Matrix, is a table often used to describe the performance of classification models on set of test data for which the true values are known. It allows the visualizations of the performance of the model or algorithm. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made.



|  |  |
| --- | --- |
| TP | Determined as a document being classified correctly as relating to a category |
| FP | Determined as a document that is said to be related to the category incorrectly |
| FN | Determined as a document that is not marked as related to a category but should be |
| TN | Documents that should not be marked as being in a particular category and are not |

Precision, is determined as the conditional probability that a random document d is classified under ci, or what would be deemed the correct category. It represents the classifiers ability to place a document as being under the correct category as opposed to all documents place in that category, both correct and incorrect:

Precision = (TP) / (TP + FP)

Recall, is defined as the probability that, if a random document dx should be classified under category (ci), this decision is taken.

Recall = (TP) / (TP + FN)

Accuracy, is commonly used as a measure for categorization techniques. Accuracy values, however, are much less reluctant to variations in the number of correct decisions than precision and recall.

Accuracy = (TP + TN) / (TP + FP + FN + TN)

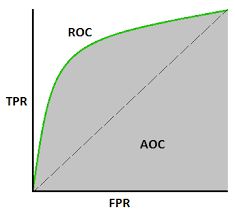
Furthermore Precision and Recall combined to F1-score, in order to get a better picture of the performance of the classifier.

F1-score = 2\*[(Precision \* Recall) / (Precision + Recall)]

Classification Report, it gives total report on how model predicted the classes correctly or incorrectly including evaluation metrics like precision, recall and etc.

AUC-ROC curve: (Area Under the Curve – Receiver Operating Characteristics), It is one of the most important evaluation metrics for checking any classification model’s performance at various thresholds settings. ROC is probability curve and AUC represents degree or measure of separability.

It tells how much model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0’s as 0’s and 1’s as 1’s



Total area under the curve determines the accuracy of our model indirectly.

**7. CONCLUSION**

It has observed that even for a specified classification method, classification performances of the classifiers based on different training text corpuses are different, in some cases differences are sustainable.

This observation implies that Classifier performance is relevant to its training corpus in some degree by tuning parameters and good or high quality training corpuses may derive classifiers of good performance.

The hybrid approach of these techniques also very helpful in text classification. In today’s world the demand of support vector machine is increased because of its kernel functions. Kernel functions plot the data into higher dimensional spaces and then data could convert effortlessly separated. The hybrid approach of linear SVM and k-NN provides better accuracy but with the use of kernel function gives better performance than linear SVM. Now the Multinomial Naïve Bayes comes into picture to compete SVM with low execution time and comparable accuracy with SVM whereas remaining models like Decision tree, Random forest takes long time to build tree and also producing comparable results.