© 2018 IJSRSET | Volume 4 | Issue 8 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section : Engineering and Technology

Comparative Study of Classification Algorithms in Sentiment Analysis

N. Lokeswari*, K. Amaravathi
CSE, Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, Andhra Pradesh, India

ABSTRACT

In daily life customer reviews or opinions play a very key role. Whenever anyone has to take a decision, they are likely to consider opinions of others. Analyzing the people's feeling is significant for many applications such as companies, ventures, enterprises trying to find out the review of their goods in the marketplace, predicting socio-economic situations like stock exchange and predicting political elections. Since there is a good quantity of collective data present on the internet automatically therefore, it's very important to employ techniques that automatically classify them. Sentiment Classification is also called as Opinion Mining. This project addresses sentiment analysis in Hotel reviews that classifies the hotel reviews according to the sentiment expressed by the public into: positive or negative. To classify the hotel reviews i.e. the given sentiment we used natural language tool kit (NLTK) to pre-process the natural language tasks. While pre-processing the text review first we remove stop words, punctuations and regular expressions .We also used n-grams to improve the sentimental analysis. After pre-processing the sentiment we choose a classifier to classify the sentiment. So, in order to selecting the best classifier in this paper we use different classifiers of different models to classify the sentiment and we also give the brief comparison of classifiers in terms of Accuracy, Precision, Recall and F-Measure. With these evaluation metrics we can show the best classifier. Multinomial NaiveBayes is the most accurate classifier to our data set and it gives quite efficient results than other classifiers.

Keywords: Natural language tool kit (NLTK), Multinomial NaiveBayes, Accuracy, Precision, Recall and F-Measure, n-grams, classifier.

I. INTRODUCTION

Sentimental Analysis is simply known as Opinion mining. In other words it is determined as the process of determining the attitudes, emotions and opinions which are behind a series of words called Natural language processing. It is now and then referred to as opinion mining, although the importance in this case is on extraction. Using NLP, machine learning methods to extract, identify, or otherwise characterize the sentiment content of a text unit is called as sentimental analysis.

Travel scheduling and booking on the internet has become one of its most key marketable uses. Particularly for hotel booking, such customer reviews are significant since they are more definite and comprehensive than reviews originate in usually printed hotel magazines etc., they are not unfair by marketing consideration as e.g. the hotels front page and reflects genuine experiences of visitors. In general words, sentiment analysis aims to find out the attitude of a speaker or a writer with respect to some subject or the overall related division of a document. The applications of sentiment analysis are wide and powerful. The capability to pull out insights from

social data is a practice that is being commonly adopted by organizations across the world.

To be specific, word Sentiment is extremely broad and it constitute feelings, opinions, dispositions, and so on. In this hypothesis, we talk just about the reviews communicated in written pattern which are collected in texts which are written in human readable natural language.

In this paper, we focus on a review on a website which may be largely positive about a hotel, but also negative concerning how bad the hotel of it is. Able to classify this kind of data in an ordered way gives the client a much clearer demonstration of public view than surveys or focus groups do, because the statistics is created by the client.

Clients can be contradictory in their statements. Most reviews will have both good and bad remarks, which is rather controlled by analyzing sentences one at a time. Recommender system gives people to obtain decisions in these compound information places. Recommender systems are a sort of data filtering that presents lists of objects (films, videos, products.) which are based on user attention

II. PROPOSED ARCHITECTURE

The steps involved in the whole process of classifying the sentiment of the hotel reviews are depicted in the below figure.

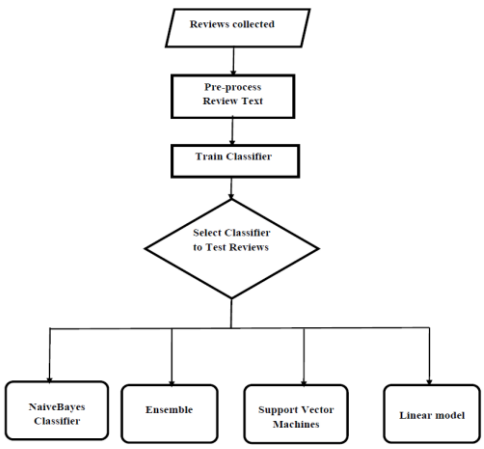

Figure 1. Architecture

A. Collecting Data

We collected 800 trip advisor reviews (https://www.tripadvisor.in) for performing sentiment analysis. The reviews we collected are plain text reviews without containing id and original sentiment and also we collect each review as a single text file for making the task easy while sentiment analyzing.

B. Pre-processing Of Data

Removing unnecessary word is a basic function when we mine unstructured data. Before applying any of the sentiment mining methods, it is a common practice to perform data pre-processing [1].We need to clean the data through basic operation. Therefore collected reviews are preprocessed by using the NLTK (natural language toolkit) platform in python.

1) Removing stop words: We have used the corpus stopword from NLTK that removes the stopwords [1]. It removes words like a, an, is, the, etc from our reviews.

2) **Removing Special Characters***:* Special characters like, [] {} ()/' should be removed. We use the RE (regular expression) operation module in python.

3) Stemming: We have used the corpus wordnet from NLTK in python which lemmatizes the word using wordnet's built in morphy function from nltk.stem.wordnet.

4) Punctuation removal: For getting text free of any punctuation like! Or ': etc we use the RE (Regular expression) operation module in python.

C. N-gram

An n-gram is an adjacent sequence of k items from a given sequence of text. The items may be base pairs, syllables, words, phonemes, or letters according to the application. These n-grams are normally collected from a speech corpus or text. If the items in the n-gram are words, then n-grams are also known as shingles.

In N-grams,

- ✓ If N=1, then it unigrams and it is basically each word in a sentence.
- ✓ If N=2, then it is bigrams and it is combination of two words in a sentence,
- ✓ If N=3, then it is trigrams and it is combination of three words in a sentence.
- ✓ If N>3, then these are referred as four grams or five grams etc…
- ✓ N-grams sentence: If W=Num of words in a given sentence S, the number of n-grams for sentence S would be:

$$Ngrams_s = W - (N - 1)$$

III. CLASSIFERS USED

In this paper we have employed Linear Model Classifiers present in Natural Language Toolkit package in Python [2].

A. Bagging

Bagging generates different classifiers only if the base learning algorithm is unbalanced that is, if small changes to the training set cause large changes in the learned classifier. Bagging can be viewed as a ways of exploiting the volatility in learning algorithms to improve classification accuracy.

Bagging is also called as Bootstrap aggregating. This is designed to improve accuracy of statistical classification and regression. To improve the classification by combining classifications of randomly generated training sets bagging was proposed. Bagging set generates m latest training sets S, each of size n1, some observations may be repeated in each S is expected to have a fraction of (1-1/e) of the unique samples of S, the rest being duplicates for a standard training set S of size n, which are given by sampling from uniformly with replacement. The type of example is called as a bootstrap sample.

The numerous editions are developed by considering bootstrap replicates about the learning set and by means of each of these learning sets as new learning sets. Tests those are performed on real and simulated datasets using categorization by means of regression trees and subset selection in the linear regression have been showing that the considered bagging can be given substantial gains in accuracy.

1) Random Forest Classifier:

Random Forest is measured to be a universal remedy of all data science problems. On a different note, when you may not think of any algorithm use random forest.

Random Forest is a adaptable machine learning technique which is able of achieving both regression and classification tasks. It may also undertake dimensional reduction methods; treat missing values, outlier values and other necessary actions of data exploration, and do a quite good quality job. It is a kind of ensemble learning method, wherever a collection of weak models merge to figure a powerful model.

In the Random Forest, we create multiple trees as contrasting to a single tree in CART model. To categorize a new object depending on attributes, each and every tree offers a categorization and we state that the tree "votes" for that particular class considered.

It working mechanism of it is in the following manner. Each and every tree is planted & grown in the following way:

1. Consider an assumption that the number of cases that are in the training set is N. Then, the example of those N cases is considered at a random rate but with a specific replacement. This above taken sample will be grouped as the training set for growing the tree.
    1. Suppose there are M input variables those are taken, a number m<M is particular such that at each node, m variables are chosen at random rate out of the M. The top split on that m is made to split the above node. The value of m is measured constant whereas we grow the forest.

2. Each and every tree is grown to the maximum extent possible and there is no pruning to be used.
3. Expect new data by cumulating the calculation of the n trees.

Python pseudo code:

```
##Import python Libraries and required libraries from sklearn.ensemble import RandomForestClassifier
###Let us assume we have, A (guessing (or) predictor) and B (output target) for training data set and A_test (guess) of test dataset
## Create object using Random Forest classifier
Model= RandomForestClassifier (n_estimators=1000)
# Train the model using the training sets and check score
Model1.fit (A, b)
Model1. Score (A, b)
#guessed Output
Predicted= Model1. Predict (x_test)
```

B. NaiveBayes Classifier

The Bayesian Classification symbolizes a supervised learning method as well as a statistical method for categorization. Suppose an essential probabilistic model and it permit us to capture uncertainty regarding the model in a principled way by explaining chances of the results.

Bayesian reasoning is useful for decision making and inferential statistics those deal with probability inference. It is used the knowledge of prior proceedings to predict future events. So, known the training data which holds the number of positive and negative labelled reviews, the NaiveBayes Classifier can expect the polarity of any review given as a test case.

This classifier is completely based on Bayes Theorem with independent assumptions between predictors. A Naive Bayesian model is simple to construct, with no complicated iterative parameter evaluation which make it mainly helpful for very huge datasets. Despite its ease, the Naive Bayesian classifier frequently does surprisingly well and is broadly applied because it often outperforms more sophisticated classification methods [6].

Bayes theorem offers a method of evaluating the posterior probability, P(c/p), from P(c), P (p), and P (p/c). NaiveBayes classifier supposes that the result of the value of a predictor (p) on a given class (c) is not dependent of the values of other predictors. This assumption is known as class conditional independence.

$$P\ (c/p) = \frac{P(p/c)P(c)}{P(p)}$$

- P(c/p) is the posterior probability of class (target) given predictor (attribute).
- P(c) is the prior probability of class.
- P (p/c) is the likelihood which is the probability of predictor given class.
- P (p) is the prior probability of predictor.

NaiveBayes Classification is based on the Bayesian theorem equation. This can be implified by assuming that features are independent when given the class. Despite this assumption, the classifier is remarkably successful in practice.

$$P\ (p/c) = \pi_{i-1}^{n} P\ (p_i/c)$$

Where $p= \{p_1, p_2, \ldots\ldots p_i\}$ is a feature vector.

It is mostly suitable as the dimensionality of the inputs is big or high. Parameter estimation for NaiveBayes models use the way of maximum probability. In spite of over-simplified assumptions, it repeatedly performs better in most of the complex real-world scenarios. In machine learning, considering theoretical and experimental data has been shown that a training procedure depending on the NaiveBayes assumptions is able to give way an optimal classifier in a variety of conditions where the assumptions are wildly violated. Recent classifications of machine learning methods for text categorization have been to some extent less favorable to NaiveBayes

models while still presenting them to achieve respectable efficiency.

The classifiers used in NaiveBayes are:

1) Multinomial NaiveBayes
2) Bernoulli NaiveBayes
3) Gaussian NaiveBayes

C. Support Vector Machine

In Machine Learning, Support Vector Machines are supervised learning models with associated learning algorithm that analyze data used for classification and regression analysis. New examples are then mapped into that same space and predicted that it belongs to a category based on which side of the gap they fall. The support vector machines which are in scikit learn holds both sparse and dense sample vectors as its input. Although, to make use of an SVM and to create predictions for sparse data, it should fit over such data, for best performance, use C-ordered numpy. ndarray (dense) (or) scipy.sparse.csr_matrix (sparse) with dtype=float36.

In this paper we used three classifiers from Support Vector Machine, they are

1) SVC Classifier
2) NuSVC Classifier
3) Linear SVC Classifier

SVC and NuSVC are alike techniques, but accept slightly dissimilar sets of parameters and contain unlike mathematical formulations. On the other side, Support Vector Classification is also put into practice by Linear SVC for the case of a linear kernel. Note down that Linear SVC does not allow keyword kernel, as this is understood to be linear. It also is short of some of the members of SVC and NuSVC like support. SVMs decision function which is not independent on subset of the training data, called the support vectors. Some characteristics of these support vectors be able to be found in members support vector, support and n support.

Let us assume a training dataset of n points of the form.

(x, y) ……(xi,yi)

Where yi denotes the class i.e. 1 or 0 to which xi belongs. Any hyper plane can be written as set of point x satisfying

$$W.x-b = 0$$

Where w is a normal vector to the hyper plane from the origin along the normal vector. For cases in which the data are not literally separable, hinge loss function is introduced.

$$\text{Maximum } (0, 1-yi\ (W.x-b))$$

The value of this function is zero if x vector lies on the correct side of the margin. The function's value is proportional to the distance from the margin if data on the wrong side of the margin.

D. Linear Model Classifier

In machine learning and statistics, classification is the difficulty of recognizing to which set of category a new observation belongs to, on the accordance of a training set of data containing observations whose category membership is identified. In this paper we have used three types of Linear Model classifiers, they are:

1) SGD Classifier
2) Logistic Regression Classifier
3) BayesRidge Classifier

1) Stochastic Gradient Descent Classifier:

Stochastic Gradient Descent is also known as Incremental Gradient Descent. Stochastic Gradient Descent is linear model in scikit library [3]. Stochastic Gradient Descent comes under the Support Vector Machine it is a stochastic approximation of the gradient descent optimization method for minimizing an objective function that is written as a sum of differentiable functions. It is a simple and very efficient to discriminative learning of linear classifiers under convex loss function such as Support Vector Machines and Logistic Regression.

The Stochastic Gradient Descent has been successfully applied to large scale and sparse machine learning problems often encountered in text classification and Natural Language Processing tasks and motivate the fact to use this in our problem.
The SGD Classifier is written in python as sklearn.linear_model. SGD Classifier.

SGD is the well-organized approach to be set for linear models which is used to solve error function called as the cost function that calculates the known line showing how good the data is fine on the line. SGD is one of the vital classifiers in Machine Learning Platform. These algorithms gain significance for large level data.
SGD is a simple learning routine which supports different loss functions and penalties for classification.

2) Logistic Regression Classifier:

The Logistic Regression is a regression representation where not independent variable is categorical. Logistic Regression measures the link between categorical dependent variable and one or more independent variables by approximating the probabilities by means of logistic function which is a cumulative logistic distribution. Logistic Regression is one of the most commonly used linear models. The Logistic Regression can be seen as generalized liner model and thus analogous to linear regression. However it is based on quite different assumptions from those of the linear regression. Logistic Regression is about the relationship between the dependent and independent variables.

Logistic Regression is not only used in Machine Learning but also various fields such as medical fields and social sciences. Logistic Regression classifier is present in scikit-learn machine learning library for python programming language.

The Logistic Regression model is used to estimate the probability of a binary response based on one or more independent features. Logistic Regression can also be termed as a special case linear regression but logistic regression is different in the sense that conditional distribution q/p is a Bernoulli distribution rather than Gaussian distribution, because the dependent variable is binary. Also, predicted values are probabilities and are restricted 0 and 1 through the logistic distribution function.
The logistic function Π(p) is defined as follows:

$$\Pi\,(p) = \frac{1}{(1+e^{-a})}$$

Where x is a linear combination and multiple explanatory variables t is treated similarly. So, p can be expressed as follows:

$$p = \beta_0 + \beta_1 t$$

And the logistic function can now be written as:

$$F\,(t) = \frac{1}{1+e^{-(\beta_0+\beta_1 t)}}$$

Also, we can now define the inverse of the logistic function, g, the logistic function is

$$g\,(F(t)) = \ln\left(\frac{F(t)}{1-F(t)}\right) = \beta_0 + \beta_1 t$$

E. Classifier Evaluation Metrics

Evaluation metrics are the solution to understanding how your classification model performs when applied to a test dataset .When we use a classifier model for a problem, we almost always want to look at the correctness of that model as the number of correct predictions from all predictions made. This is the classification accuracy. When we need to make a decision whether it is a good sufficient model to solve the problem, for evaluating the effectiveness of classifier accuracy is not the only metric we have also another metrics that evaluates the effectiveness of classifier. The other two useful metrics are recall and precision. These other two metrics can offer more superior impending into the performance uniqueness of a classifier.

A ***false positive*** error, or in short false positive, commonly called a 'false alarm', is a result that indicates a given situation has been satisfied; when it

actually has not been satisfied that is inaccurately a positive effect has been assumed.

A ***false negative*** error, or in short false negative, is where a test result indicates that a condition failed; while it actually was successful that is incorrectly no effect has been assumed.

Classifier Accuracy (or) recognition rate: Fraction of test set reviews that are properly classified.

$$A = \frac{T_p + T_n}{T_p + F_p + F_n + F_n}$$

Where,
Tp , is the number of true positives
Fp , is the number of false positives
Tn , is the number of true negatives
Fn , is the number of false negatives

This metric measures the correctness of a classifier. A lower precision means more false positives, while higher precision means less false positives. This is frequently at chances with recall, as an easy way to get better
precision is to decrease recall. Precision is defined as the number of true positives over the number of true positives plus the number of false positives.

$$P = \frac{T_p}{T_p + F_p}$$

Where,
Tp, is the number of true positives
Fp , is the number of false positives

Recall:

This metric measures the completeness, or understanding, of a classifier. Lower recall means more number of false negatives, while higher recall means less number of false negatives. Improving recall will reduce precision because it becomes progressively harder to be precise as the model space increases. Recall is defined as the number of true positives over the number of true positives plus the number of false negatives.

$$R = \frac{T_p}{T_p + F_n}$$

Where,
Tp , is the number of true positives.
Fn , is the number of false negatives.

F-measure:

This metric was generated by the combination of two metrics that are Precision and recall and the single generated metric is known as F-measure.
F-Measure is the weighted harmonic mean of recall and precision.

$$F1 = 2\frac{P \times R}{P + R}$$

Where,
P, is the precision.
R, is the recall.
We evaluate and calculate the performance of the four models that we use based on their Accuracy, Precision, Recall and F-measure.

IV. EXPERIMENTAL RESULTS

In this paper we used different models and compare the classifiers that are present in these models. Below tables compare the classifiers.

Table 1. Comparison of classifiers of uni-gram, bi-gram and tri-gram in Ensemble Model

UNI+BI+TRI	Random Forest	Ada Boost	Gradient Boost	Bagging	Extra Trees
Accuracy	88.50	86.00	86.50	88.00	87.50
Precision	92.00	86.00	93.00	92.00	91.00
Recall	85.98	86.00	82.30	85.18	85.05
F_Measure	88.88	85.99	87.32	88.46	87.92

Table 2. Comparison of classifiers of uni-gram, bi-gram, and tri-gram in SVM

UNI+BI+TRI	SVC	NuSVC	Linear SVC
Accuracy	85.50	86.00	85.50

Precision	91.00	91.00	89.00
Recall	81.98	82.72	83.18
F_Measure	86.25	86.67	86.00

Table 3. Comparison of classifiers of uni-gram, bi-gram, and tri-gram in NaiveBayes Model

UNI+BI+TRI	Multinomial NB	Bernoulli NB	Gaussian NB
Accuracy	87.50	86.50	85.50
Precision	95.00	92.00	95.00
Recall	82.61	82.88	79.83
F_Measure	88.37	87.20	86.76

Table 4. Comparison of classifiers of uni-gram, bi-gram, and tri-gram in Linear Model

UNI+BI	SGDC	Logistic Regression
Accuracy	86.50	88.00
Precision	97.00	93.00
Recall	80.16	84.54
F_Measure	87.78	88.57

GRAPHS:

In order to compare and understand these classifiers we plot these classifiers in the below graphs.

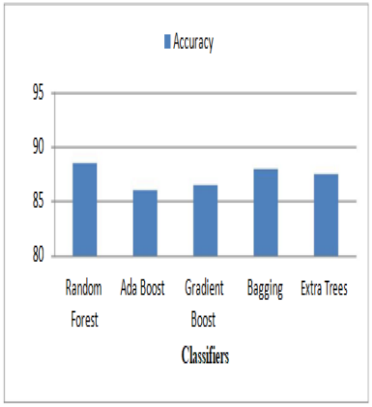

Figure 2. Comparison of Accuracy percentage among Ensemble Classifiers

On observation Random forest shows the highest accuracy percentage of 88.50%.

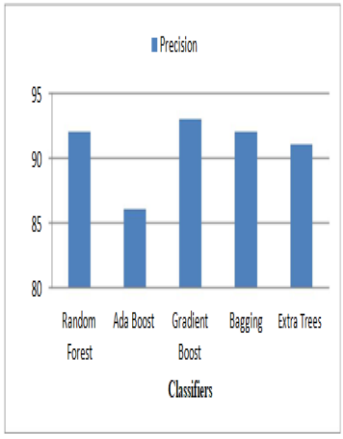

Figure 3. Comparison of Precision percentage among Ensemble Classifiers

On observation Gradient Boost shows the highest Precision percentage of 93%.

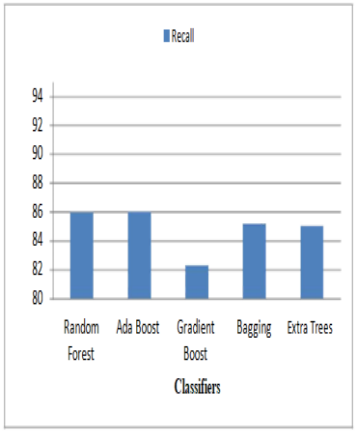

Figure 4. Comparison of Recall percentage among Ensemble Classifiers

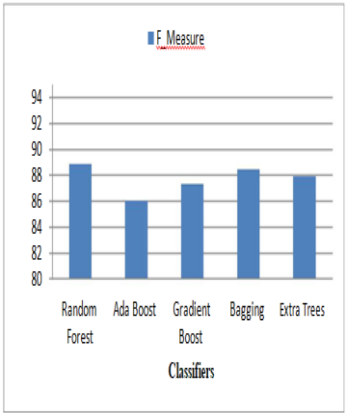

Figure 5. Comparison of F-Measure percentage among Ensemble Classifiers

V. CONCLUSION

Above results shows the accuracy, precision, recall and F-measure of models Ensemble, Support vector

machines, NaiveBayes Classifier and Linear models and compare their classifiers in each model with the help of graphs. In Ensemble methods Bagging Classifier is the most efficient classifier using unigram and bigram classification, whereas Random forest classifier is the most effective classifier while using uni, bi, trigrams. In Support vector machine methods SVC Classifier is the most efficient classifier using unigram classification, Linear SVC Classifier is the most efficient classifier using uni, bigrams, whereas Nu SVC classifier is the most effective classifier while using uni, bi, trigrams. In NaiveBayes Classifier, Multinomial Classifier is the most efficient classifier using unigram, bigrams and trigrams. In Linear models, Stochastic Gradient Descent Classifier (SGDC) is the most efficient classifier using unigram, bigrams and trigrams. Overall Multinomial NaiveBayes Classifier is the most efficient Classifier to our data set as we know that NaiveBayes Classifiers are mostly efficient to the small data sets. Since our data set is taken from trip advisor India hotels and we took only 800 reviews that is very small data set. In order to implement this project we collected data manually, so we collected only 800 reviews. NaiveBayes Classifier is the most efficient classifier to our dataset although the remaining classifiers also give fair results.

VI. FUTURE WORK

In this paper we classify the given sentiment into two categories whether it is positive or negative only. For example, reviews like good, awesome, extraordinary, great, best, nice good are considered as positive reviews whereas reviews like bad, poor, awful, worst are considered as negative reviews. But in this paper good, very good, very- very good are considered as positive words but we do not classify them into more generous way.

Recent years there has been a rapid growth in online review sites and discussion reviews such as www.tripadvisor.com where the key characteristics of customer review are drawn from their overall opinion/sentiment. So, in future we want to work on the sentiment to classify into more categories that gives rating in range from 1 to 5. The more classifier sentiment will always give best opinion of user.

VII. REFERENCES

[1]. H. Saif, M. Fernandez, Y. He, and H. Alani, "On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter," Proc. Ninth Int. Conf. Lang. Resour.Eval., no. i, pp. 810–817, 2014.

[2]. "Python programming." Online]. Available: https://pythonprogramming.net/.

[3]. "Machine Learning Tutorials." Online]. Available: http://machinelearningmastery.com/.

[4]. F. Pedregosa, R. Weiss, and M. Brucher, "Scikit-learn : Machine Learning in Python," vol. 12, pp. 2825–2830, 2011.

[5]. G. Vinodhini and R. Chandrasekaran, "Sentiment Analysis and Opinion Mining: A Survey," Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 2, no. 6, pp. 282–292, 2012.

[6]. V. Narayanan, I. Arora, and a Bhatia, "Fast and accurate sentiment classification using an enhanced Naive Bayes model," Int. Data Eng. Autom. Learn. Lect. Notes Comput.Sci., vol. 8206, pp. 194–201, 2013.

[7]. R. S. Michalski, J. G. Carbonell, and T. M. Mitchell,"Machine Learning: An Artificial Intelligence Approach," vol. 2, no. 12, pp. 3400–3405, 1984.

[8]. W. Pan, X. Shen, and B. Liu, "Cluster Analysis: Unsupervised Learning via Supervised Learning with a Non-convex Penalty.," J. Mach. Learn. Res., vol. 14, no. 7, p. 1865, 2013.