Tribhuvan University

Institute of Science & Technology

Comparative Analysis of Machine Learning based Classification Algorithms for Sentiment Analysis

Dissertation

Submitted To:

Central Department of Computer Science & Information Technology

Tribhuvan University

Kirtipur, Kathmandu

Nepal

In partial Fulfillment of the requirements for the Degree of Master of Science in Computer Science and Information Technology

Submitted By:

Tekendra Nath Yogi (08/071)

November, 2019

Supervisor

Asst. Prof. Nawaraj Paudel

Central Department of Computer science and Information Technology

Tribhuvan University, Kirtipur, Kathmandu

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

Student's Declaration

I hereby declare that I am the only author of this work and that no sources other than the listed here have been used in this work.

… … … … … … …….

Tekendra Nath Yogi

Date: Nov, 2019

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

Supervisor's Recommendation

I hereby recommend that the dissertation prepared under my supervision by **Mr. Tekendra Nath Yogi** entitled **"Comparative Analysis of Machine Learning based Classification Algorithms for Sentiment Analysis"** be accepted as in fulfilling partial requirement for completion of master Degree of science in computer science and information Technology.

… … … … … … … … … …

Asst. Prof. Nawaraj Paudel

Central Department of Computer science and Information Technology,

Tribhuvan University,

Kirtipur, Kathmandu

Date: Nov, 2019

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

LETTER OF APPROVAL

We certify that we have read this dissertation and in our opinion it is appreciable for the scope and quality as a dissertation in the partial fulfillment for the requirement of Masters Degree in Computer Science and Information Technology.

Evaluation Committee

…… ……. …….

Asst. Prof. NawarajPaudel
Head of Department
Central Department of Computer science
& Information Technology
Tribhuvan University
Kirtipur, Kathmandu

Asst. Prof. NawarajPaudel
(Supervisor)
Central Department of Computer science
& Information Technology
Tribhuvan University
Kirtipur, Kathmandu

……. …….

(External Examiner) **(Internal Examiner)**

Date: Nov, 2019

Acknowledgement

I would like to express my sincere thanks to my supervisor **Asst. Prof. Nawaraj Paudel**, Central Department of Computer science and Information Technology Tribhuvan University, Kirtipur, Kathmandu, Nepal for his support, motivation, suggestions and guidance. His advice was inevitable and with his help I was able to work on my own interested field and complete my dissertation on time.

I am also thankful to **Asst. Prof. Nawaraj Paudel**, Head of Department, CDCSIT who has provided all the help and facilities, which I required, for the completion of my dissertation.

Moreover, I would like to express my heartfelt gratitude to all my teachers at Central Department of Computer Science and Information Technology, Tribhuvan University who have imparted knowledge in various subjects.

Last but not the least; I would like to express my thanks to all my friends and my lovely parents for direct and indirect supports for the completion of this thesis.

Tekendra Nath Yogi

Date: Nov, 2019

Abstract

Sentiment analysis is the process of predicting the sentiment polarity of review data based on a given data set. Nowadays, sentiment analysis is more popular in Internet in general and in social media in particular. In the web huge amount of review data generated in each day is rapidly increasing day to day so there need to process these data to detect the sentiment polarity of large review dataset as early as possible.

In this research, comparison of three different machine learning based classification algorithms for sentiment analysis i.e. Multinomial naïve Bayes (MNB), K-Nearest-Neighbors (KNN) and Support Vector Machines (SVM) is presented. The main aim of this research is to evaluate their performance of those three different machine learning based classification algorithms for sentiment labeled sentences datasets with different size. The sentiment labeled sentences datasets used for this research is chosen such way that they are different in size, mainly in terms of number of instances. When comparing the performance of all three machine learning based classification algorithms for sentiment analysis, SVM is found to be better algorithm to detect sentiment polarity in all three sentiment labeled sentence datasets in every aspect, whereas MNB and KNN had got less performance in every aspect as compared to SVM.

Keywords: Sentiment analysis, KNN, SVM, MNB.

Table of Contents

List of Figures

Figures ..... pages

List of Tables

Tables..... pages

List of Abbreviations

Abbreviations	Full form
chi2	Chi-Squared
DF	Document Frequency
FN	False Negative
FP	False Positive
IDE	Integrated Development Environment
IMDB	Internet Movie Database
KNN	K-Nearest Neighbors
ML	Machine Learning
MLBCA	Machine Learning Based Classification Algorithms
MNB	Multinomial Naive Bayes
NB	Naive Bayes
NLP	Natural Language Processing
SMO	Sequential Minimal Optimization
SVM	Support Vector Machines
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
TN	True Negative
TP	True Positive

CHAPTER 1

1. INTRODUCTION

1.1. Introduction to sentiment analysis

An opinion is a viewpoint or judgment about a specific thing that acts as a key influence on an individual process of decision making. Collectively Opinions reflects the "wisdom of crowds" and can be good indicator of the future [1]. In addition People's belief and the choices they make are always dependent on how others see and evaluate the world. So opinion holds high values in many aspect of life. Sentiment analysis is the process of determining opinions or sentiments as positive or negative from review which are expressed by people over a particular subject, area, product, or services offered by companies, governments and organizations [2]. In recent years, this field is widely appreciated by researchers due to its dynamic range of application in various numbers of fields. There are several areas such as marketing; politics; news analytics etc. which are benefited from the result of sentiment analysis [3].

Due to the vast range of product and services these days, it has become difficult for the users to select their preferred product. Product reviews turn out to be very useful reference. Despite of the willingness of people to share their thoughts and views about the product, a problem persists due to the huge amount of total reviews [3]. This develops a need for technology of data mining to uncover information automatically and assist in decision making. Such data mining technology is sentiment analysis for classifying opinion based on the review polarity [4].

There are two main approaches for sentiment analysis: machine learning approach and lexicon-based approach to solve the problem of sentiment classification [5]. The former approach is applied to classify the sentiments based on training as well as test data sets. The second category doesn't require any prior training data set; it performs the task by identifying a list of words, phrases that consists of a semantic value. It mainly concentrates on patterns of unseen data [3].

This field becomes more challenging due to the fact that many demanding and interesting research problems still exist in this field to solve. Sentiment based analysis of a document

is quite tough to perform in comparison with topic based text classification. The opinion words and sentiments are always varied with situations. Therefore, an opinion word can be considered as positive in one circumstance but may be that becomes negative in some other circumstance. The opinionated word 'unpredictable' is used in different senses in a different domain. For example, "an unpredictable plot in the movie" gives a positive opinion about the movie, while "an unpredictable steering wheel" is a negative expression considering the product, car [3].

Sentiment classification process has been classified into three levels: document level, sentence level, and feature level. In Document level the whole document is classify either into positive or negative class. Sentiment classification at the sentence level, considers the individual sentence to identify whether the sentence is positive or negative. Feature level sentiment classification concerns with identifying and extracting product features from the source data [5].

During this research study, the focus has been made on sentiment polarity classification based on three sentiment labeled sentence datasets, namely Yelp restaurant review dataset, Amazon cell phones and accessories review dataset and IMDB movie review dataset, by using three text classification algorithms, namely MNB, KNN and SVM. To improve the processing efficiency and sentiment polarity classification performance the data has been preprocessed, such as case folding, stop word removing, stemming and lemmatization, and chi-squared method has been used for feature selection and TFIDF method has been used for feature weighting before fed as input to the sentiment polarity classification algorithms. The sentiment polarity classification algorithms have been evaluated based four performance evaluation parameters accuracy, precision, recall and F-measure.

1.2. Problem statement

Several review messages express opinions about events, products, and services, political views or even their author's emotional state and mood. Sentiment analysis has been used in several applications including analysis of the repercussions of events in social

networks, analysis of opinions about products and services, and simply to better understand aspects of social communication in Social Networks. There are multiple methods for measuring sentiments, including lexical-based approaches and supervised machine learning methods. Despite the wide use and popularity of some methods, it is unclear which method is better for identifying the polarity (i.e., positive or negative) of a review message as the current literature does not provide a method of comparison among existing methods in combination with preprocessing, feature selection and feature weighting methods. Such a comparison is crucial for understanding the potential limitations, advantages, and disadvantages of popular methods in analyzing the content of review messages. This study aims at filling this gap by presenting comparisons of three popular machine learning based classification algorithms (MLBCA) for sentiment analysis in combination with preprocessing, feature selection and feature weighting methods.

1.3. Objective of thesis

The main objectives of this research are:

- To analyze the sentiment of text data whose sentiment category is unknown based on given training data set by using MLBCAs, namely MNB, KNN and SVM, in combination with preprocessing and feature selection methods.
- To perform comparative analysis of these MLBCAs for sentiment analysis based on accuracy, precision, recall and F-measure.

1.4. Structure of report

This report is organized in the following five chapters.

- Chapter 1 of this dissertation is introduction, which is organized into subsequent four sub-sections.
    - First section is about introduction and overview of sentiment analysis.
    - Second section is about stating the existing problem in previous works in sentiment analysis along with the need of this study to select the better algorithm for sentiment analysis.

- Third section is about objective of this dissertation.
- Fourth section is about limitation of this dissertation.

- Chapter 2 provides the systematic summaries of all the existing research works to this topic in detail under literature review.
- Chapter 3 includes details of research methodological steps such as data collection, preprocessing techniques, feature selection and feature vector construction techniques, classification algorithms for sentiment analysis, evaluation matrices to measure the performance of algorithms and tools that were studied and used to conduct this research.
- Chapter 4 contains all the details of data which is applied for analysis purpose and comparative performance measure of three MLBCAs for sentiment analysis. The result of the comparative study is shown in tabular form as well as in graph..
- Chapter 5 provides final conclusion and future works of the study.

CHAPTER 2

2. LITERATURE REVIEW

2.1. Background and Literature Review

Sentiment analysis is one of the sought application area of text mining and NLP in recent years due to its wide practicality in determining what the users want from the review data in the web. Now days this field is rapidly growing and there are various research aspects that need to be considered in order to get the better result of analysis helps in better decision making.

M. Annette and G. Kondark proposed a novel approach based on Support Vector Machines and compares lexical-based and Machine learning based approaches. From this comparative study they conclude that machine learning based approach for sentiment classification is quite successful and outperforms the lexical based approach [6].

Lopes et al. has been employed both supervised and unsupervised ML algorithms for automatic classification of sentiments from 2000 social network users. The researchers found that supervised machine learning technique outperformed the unsupervised machine learning techniques with low classification error [7].

Pang et al. carried out an experiment on automatic classification of Sentiments in text documents using classification algorithms. The researchers classified the text documents by topic, and overall sentiment of documents according to negative and positive sentiments. From their experiment the researchers found that classification algorithm perform poorly on the sentiment classification by topic [8] [9].

A. Kathuria, and s. Upadhyay provide a comparative study of machine learning based, lexical based and rule based approaches for sentiment analysis. They found that machine learning based approach outperforms both lexical based and rule based approaches and additionally shows that more the cleaner the data more correct the knowledge [10].

E. Elmurngi, and A. Gherbi have been compared five classification algorithms on movie review data set in order to identify fake review data set. In this comparative study data are used without preprocessing and found that SVM outperforms all other four classification algorithms namely Naïve Bayes, KNN, K*, and Decision Tree-J48 [11].

Zhao Jiananng and GUI Xialin proposed a method which combines machine learning based methods with preprocessing techniques to determine the sentiment of twitter data and then the researcher performs comparative analysis with usual machine learning methods. It is found that the proposed method outperforms the usual machine learning methods [12].

S.D. Sarkar, and S. Goswami proposed a method to compare four feature selection methods chi-squared, Information gain, Mutual information and symmetrical uncertainty in combination with two machines learning methods SVM and NB and found that SVM with Information gain outperforms all others. But the chi- Squared method has better noise tolerance [13].

K. Huda and et al. analyzed twitter data and proposed a new machine leaning based method in combination with preprocessing and feature selection methods for sentiment analysis. The proposed Machine learning base method has four steps that followed for the sentiment analysis in the first step, the first step is applied in which data pre-processed. In the second step feature of the data will be extracted which is given as input to the third step in which data is classified for the sentiment analysis. They found that the proposed method outperformed the usual machine learning methods [14].

B. Trstenjak et al. proposed a framework for text classification base on the KNN algorithm and the TF-IDF method. This framework proves a good result and provides the ability to upgrade and improve the present embedded classification algorithm [15].

S. Ahmed, et al. proposed a method in combination with preprocessing and TFIDF weighting and performed a comparative analysis with the usual machine learning algorithms by using twitter data. The researchers found that the proposed method outperforms the usual machine learning methods for sentiment analysis [16].

In [17] M. Mowafy et al. proposed a method by combining Multinomial Naive Bayes as a selected machine learning technique for classification, and TF-IDF as a vector space model for text extraction, and chi2 technique for feature selection. This proposed method outperforms the framework proposed in [15].

In [18] U. I. Larasati proposed a method by combining Support Vector Machine as a selected machine learning technique for classification, and TF-IDF as a vector space

model for text extraction, and chi2 technique for feature selection. This proposed method outperforms the normal SVM.

Besides this, there is no task of comparative analysis of MNB, KNN, and SMO in combination with preprocessing, Feature selection, and TF-IDF weighting in order to select the best text classification algorithm to perform the sentiment analysis of text data.

CHAPTER 3

3. RESEARCH METHODOLOGY

There are different techniques to find the polarity of a review data. The most popular and efficient one is Machine learning based sentiment analysis technique. The machine learning based sentiment analysis technique decides the polarity of a data point as either positive class or negative class. To decide the polarity of a review data and to find the most efficient algorithm following steps were used in this research as shown in figure below.

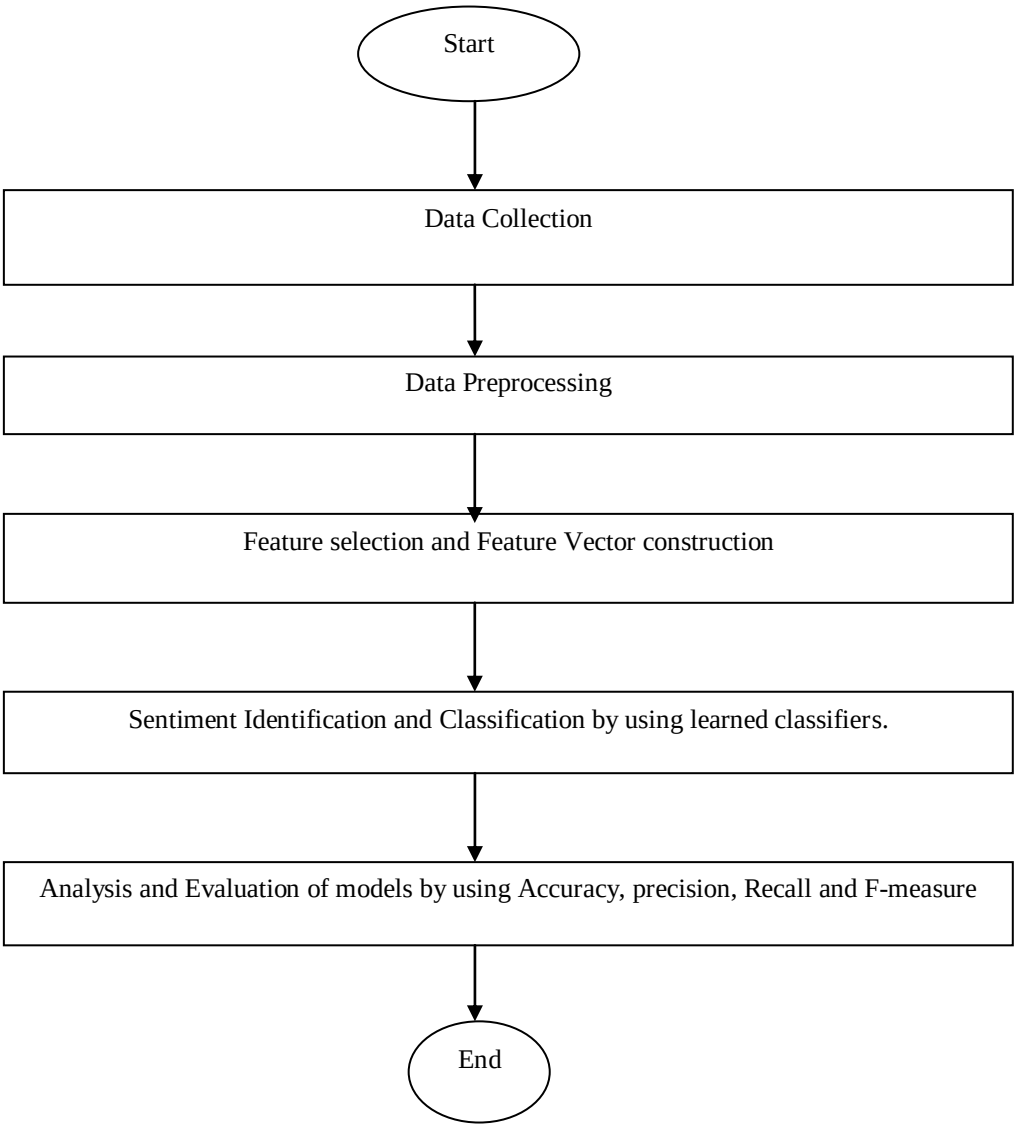

Figure 3.1: Flow chart for entire process of research

3.1. Data collection

In this research three types, sentiment labeled sentence, data sets have been used which are collected from Kaggle machine learning repository. The data sets are different in size in terms of number of sentences. These data sets can be used for any sorts of text classification task but in this research all three data sets were used for sentiment analysis after minor preprocessing such as case folding, stop word removing, etc.

3.1.1. Dataset 1

The first data set that has been used in this research is Yelp restaurant review data set, which was collected from the Kaggle machine learning repository. The data set contains two attributes namely review and sentiment, the value of first one attribute is the review text and the value of next one is the sentiment category, 0 for negative sentiment category and 1 for positive sentiment category, for the corresponding review text. This data set has 992 reviews with their corresponding sentiment categories [19].

3.1.2. Dataset 2

The second data set that has been used in this research is Amazon cell phones and accessories review data set, which was collected from the Kaggle machine learning repository. The data set contains two attributes namely review and sentiment, the value of first one attribute is the review text and the value of next one is the sentiment category of the corresponding review text. This data set has 1000 reviews with their corresponding sentiment categories [19].

3.1.3. Dataset 3

The third data set that has been used in this research is movie review data set, which was collected from the Kaggle machine learning repository. The data set contains two attributes namely review and sentiment, the value of first one attribute is the review text and the value of next one is the sentiment category of the corresponding review text. This data set has 25000 reviews with their corresponding sentiment categories [19].

3.2. Tools Used

In this research, all the algorithms used to conduct entire research methodology were implemented in python programming language by using Pycharm IDE.

3.2.1. Programming language

Python is a powerful, widely used, general-purpose, high level programming language and is becoming an increasingly popular tool in research. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. The popularity of Python grows rapidly in the recent year because of it is intuitive to learn, has a flourishing online community, easy-to-use, versatile functionality and is open-source. In addition, Python can accomplish most day-to-day research tasks and can be used at multiple steps of the research methodology such as data collection and preprocessing, feature selections and data representation and organization, statistical analysis and modeling, and visualization. So, Instead of using different software packages to accomplish different tasks python was used in this research which helps to save a significant amount of time and efforts needed to conduct the research [20].

3.2.2. Pycharm IDE

PyCharm is a cross-platform IDE for Python which was develop by JetBrains. PyCharm's smart code editor provides first-class support for Python. In addition it has other features also such as code completion, error detection, and on-the-fly code fixes. Smart search can be used to jump to any class, file or symbol, or even any IDE action or tool window. One click is sufficient to switch to the declaration, super method, test, usages, implementation, and more. PyCharm has a huge collection of tools, including an integrated debugger and test runner, Python profiler, a built-in terminal and many more. PyCharm integrates with Jupyter Notebook, has an interactive Python console, and supports Anaconda as well as multiple scientific packages including Matplotlib and NumPy. These features of Pycharm IDE make this research more comfortable [21].

3.3. Data Preprocessing

The preprocessing phase aims to prepare unstructured opinions text data (reviews) ready for further processing. Preprocessing step that were conducted in this research includes:

Case folding: DO NOT waste your time on this 'film -> do not waste your time on this film.

Stop word removing: do not waste your time on this film->do not waste your time this film.

Stemming: do not waste your time this film ->do not wast yo tim thi film. And

Lemmatization: do not waste your time this film-> do not waste your time this film.

3.4. Feature selection and feature vector construction

One inherent problem of a computer is that it cannot process text data directly. So need to represent text data in numeric form. Generally, terms are used as features to represent the text. This leads to high dimension in the text representation. To improve the classification performance and processing efficiency, features need to be filtered to reduce dimension and remove noise [17].

3.4.1. TFIDF

TFIDF is a method to calculate the numeric weight for each term (t_i) in each document (d_j) as:

$$TF - IDFij(ti, dj, D) = TFij * \log\left(\frac{N}{1 + DFi}\right)$$

Where, TF_{ij}, is the occurrence frequency of the term t_i in the document d_j. N, is the total number of document in training set. DF_i, total number of documents containing term t_i.

TF-IDF represents the importance of terms in the training set (D). But the problem with this method is that it is unable to represent the association between features and categories. One possible solution to this problem is to use feature selection method called the chi-square method. The chi-square method

helps to measure the association between term t_i and class c_k. In addition chi-square method holds the strongest noise tolerance ability [17][18].

3.4.2. Combined Chi-square and TFIDF method

To achieve the better sentiment analysis result TFIDF and Chi-square method can be combined as [18]:

1. First, feature selection value i.e., Chi-square value will be calculated as:

$$\chi 2(\mathrm{ti}, \mathrm{ck}) = \frac{\mathrm{N} * (\mathrm{ad} - \mathrm{bc})2}{(\mathrm{a} + \mathrm{c}) * (\mathrm{b} + \mathrm{d}) * (\mathrm{a} + \mathrm{b}) * (\mathrm{c} + \mathrm{d})}$$

   Where,

   N, Total number of documents in training set.

   a, Number of documents with term ti and belong to a category ck.

   b, Number of documents with term ti and do not belong to category ck.

   c, Number of documents without term ti and belong to a category ck.

   d, Number of documents without term ti and do not belong to a category ck.

2. The Higher value of $\chi 2(\mathrm{ti}, \mathrm{ck})$ indicates the closer relationship between term t_i and class c_k and $\chi 2(\mathrm{ti}, \mathrm{ck}) = 0$ indicates independent relationship between term t_i and class c_k. The score of the term t_i in the entire training set (D) is obtained as:

$$\mathrm{Max}\ \chi 2(\mathrm{ti}, \mathrm{ck}) = \underset{\mathrm{k}=1}{\overset{|\mathrm{c}|}{\mathrm{Max}}} \{(\mathrm{ti}, \mathrm{ck})\}$$

3. Rank the features in descending order in terms of the feature selection function values.
4. Choose the top K features to achieve the goal of feature selection.
5. Finally, to obtain the combined feature weight multiply feature selection values and TF-IDF value and normalize the product.

In this way the document-feature weight matrix was formed.

3.5. Classification algorithms for sentiment analysis

There are many popular and widely used classification algorithms for sentiment polarity identification of opinions of users based on the given opinion data. Among them the most commonly used and popular classification algorithms [22] that were compared in this research are as follows:

3.5.1. Multinomial Naïve Bayes algorithm

Naive Bayes is a family of algorithms based on applying Bayes theorem with a strong (naive) assumption, that every feature is independent of the others, in order to predict the category of a given sample. They are probabilistic classifiers, calculate the probability of each category using Bayes theorem, and the category with the highest probability is output [22].

MNB is a probabilistic classifier, meaning that for a document d, out of all classes $c_k \in C$ the classifier returns the class c_k which has the maximum posterior probability. MNB is always a preferred method for any sort of text classification as taking the frequency of the word into consideration, and get back better accuracy than just checking for word occurrence [23].

Algorithm:

Input:

- D, a training data set consisting of the 'm' training documents and their associated target values i.e., D={(d_1,c_1), (d_2, c_2),…….(d_m, c_j)};
- A fixed set of classes C={c_1, c_2,c_3,……..,c_j}
- A test document set testdoc

Output:

- A learned classifier
- A predicted class c ∈C for document testdoc

Method:

1. Function TRAIN MNB(D, C) returns log P(c) and log P(w|c)
    a. for each class c ∈ C # Calculate P(c_j) term

N_{doc} ← number of documents in D

N_c ← number of documents from D in class c.

-logpriori[c] ← $\log(\frac{Nc}{Ndoc})$

- doc V← vocabulary of D

- bigdoc[c]←append(d) for d ∈ D with class c

- for each word w in V # Calculate P(w|c) term

- count(w,c)← number of occurrences of w in bigdoc[c]
- loglikelihood [w, c] ← $\log \frac{count(w,c) + 1)}{\sum_{w \text{ in } V}((count\ (w\ ,c) + 1)}$

b. return logprior, loglikelihood, V

2. function TEST MNB(testdoc, logprior, loglikelihood, C, V) returns best class c
   a. for each class c ∈ C
      i. sum[c]← logprior[c]
      ii. for each position i in testdoc
          word← testdoc[i]
          if word ∈ V
              sum[c]←sum[c]+ loglikelihood[word, c]
   b. return argmax_c sum[c]

3.5.2. K-Nearest Neighbor algorithm

KNN is a non-parametric, lazy learning algorithm [24]. To identify the sentiment of new test document KNN classifier computes the similarity between a new test document and every training document. Then KNN classifier sort the training documents in descending order of their similarity to the test document in order to pick the top K most similar training documents with a test document. Finally the KNN classifier assigns this new test document to a sentiment category that has the highest score of similarity [25] [26].

Algorithm:

Input:

- D, a training data set consisting of the 'm' training documents and their associated target values i.e., D={(d_1,c_1), (d_2, c_2),…….(d_m, c_j)};
- A fixed set of classes C={c_1, c_2,c_3,……..,c_j}
- Value of K
- A test document set testdoc

Output:

- A predicted class c ∈ C for document testdoc

Method:

1. Read the training data set D, data sample to be classified X and the value of k (number of nearest neighbors)
2. For each document in test data (X_i)and For each document in training data (D_j).
   a. Calculate similarity

$$\text{Similarity (Xi, Dj)} = \frac{\sum_{k=1}^{m}(\text{Wjk} * \text{Wik})}{\sqrt{\sum_{k=1}^{m}(\text{Wjk})^2 * \sum_{k=1}^{m}(\text{Wik})^2}}$$

   Where, X_i is the test document vector, D_j is the training document vector, W_{ik} corresponds to the weight of the k^{th} element of the term vector X_i and W_{jk} is the weight of the k^{th} element of the term vector D_j.

   b. Find the K-Nearest training document for each test document (X_i). This can be done by sorting the training documents in descending order of similarity with test document (X_i) and then pick the top K documents only.
   c. Assign the class which is most common in the k-Nearest training tuples to the test document (X_i). The most common class is decided as:

Category (X_i)=arg $max_{ci}(\sum_{Dj \in KNN(Xi)}$ Similarity $(Xi, Dj) *$ y(Dj,Ci)

Where, $y(Dj, Ci) = \begin{cases} 1, & \text{if } Dj \in Ci \\ 0, & \text{otherwise} \end{cases}$

d. Return the predicted class.

3. End

3.5.3. Support Vector Machines algorithm

A Support Vector Machine (SVM) performs classification by finding the hyper plane (classifier) that maximizes the margin between the two classes subject to the constraint that all the training tuples should be correctly classified. Hyper plane is defined by using the data points that are closest to the boundary. These points are called support vectors and the decision boundary itself is called support vector machine. The main advantage of SVM classifier is that it minimizes the training set error and the test set error [27].

To obtain a SVM classifier with the best generalization performance, appropriate training is required. The most commonly used and popular algorithm for training SVM is the SMO algorithm. The main advantage of SMO algorithm is that it works analytically on a fixed size working set by decomposing the large training data set. So, that it can works fine even for large data sets and it also gives superb performances in almost all kinds of training data sets [28].

SVM algorithm:

Input:

- D, a training data set consisting of the 'm' training documents and their associated target values i.e., D={(x_1,y_1), (x_2, y_2),…….(x_m, y_j)}; where xi ∈ R^N is a feature vector and yi ∈ {-1,1} is a class label.

- The Value of regularization parameter, C, is the margin parameter that determines the trade-off between maximizing the margin and minimizing the classification error and is chosen by means of a validation set
- A test document set testdoc

Output:

- A learned classifier
- A predicted class c ∈ C for document testdoc

Method:

1. Read the training data set D, data sample to be classified X and the value of regularization parameter C.
2. Fit the model's parameters w, b and α's to a training data set by using SMO algorithm.
3. Construct a decision boundary or classifier by using the model parameters obtained in step 2, to make a prediction.

$$f(x) = \sum_{\alpha!=0}^{l} yi * \alpha i * xi.x + b$$

4. Make a prediction at a new point input x by using the model constructed at step 3 as:

$$\text{class}(x) = \begin{cases} +1 \text{ , if } f(x) \geq 0 \\ -1 \text{ , if } f(x) < 0 \end{cases}$$

5. End

SMO Algorithm:

Input:

- Training data $D=\{(x_1,y_1), (x_2, y_2),\ldots\ldots(x_m, y_j)\}$; where $x_i \in R^N$ is a feature vector and $y_i \in \{-1,1\}$ is a class label.
- C: Regularization parameter
- tol: Numeric tolerance

- max passes: max number of times to iterate over α's without changing

Output:

- $\alpha \in R^m$: Lagrange multipliers for solution
- $b \in R$: threshold for solution
- w: Weight vector

Method:

1. procedure takeStep(i1,i2)

```
if (i1 == i2) return 0
alph1 = Lagrange multiplier for i1
y1 = target[i1]
E1 = SVM output on point[i1] – y1
s = y1*y2
Compute L, and H as:
        If (y1!=y2) then L=max(0, a2-a1) and H=min(C,C+a2-a1).
        Else L=max (0, a2+a1-C) and H=min(C, a2+a1).
if (L == H) return 0
eta = point[i1].point[i1]+ point[i2].point[i2]-2* point[i1].point[i2]
if (eta > 0)
{
        a2 = alph2 + y2*(E1-E2)/eta
        if (a2 < L) a2 = L
        else if (a2 > H) a2 = H
}
else
{
        Lobj = objective function at a2=L
        Hobj = objective function at a2=H
        if (Lobj < Hobj-eps)
                a2 = L
```

```
            else if (Lobj > Hobj+eps)
                    a2 = H
            else
                    a2 = alph2
    }
    -if (|a2-alph2| < eps*(a2+alph2+eps)) return 0
    -a1 = alph1+s*(alph2-a2)
    -Update threshold (b),weight vector(w) and error cache to reflect
    change in a1 & a2,
    -Store a1 and a2 in the alpha array
    return 1
endprocedure
```

2. procedure examineExample(i2)

```
y2 = target[i2]
alph2 = Lagrange multiplier for i2
E2 = SVM output on point [i2] – y2
r2 = E2*y2
if ((r2 < -tol && alph2 < C) || (r2 > tol && alph2 > 0))
{
    if (number of non-zero & non-C alpha > 1)
    {
            i1 = result of second choice heuristic
            if takeStep(i1,i2)
                    return 1
        }

        loop over all non-zero and non-C alpha, starting at a random
        point
        {
            i1 = identity of current alpha
            if takeStep(i1,i2)
                    return 1
```

```
        }
        loop over all possible i1, starting at a random point
        {
            i1 = loop variable
            if (takeStep(i1,i2)
                    return 1
        }
    }
    return 0
endprocedure
```

3. **main routine:**

```
    numChanged = 0;
    examineAll = 1;
    while (numChanged > 0 | examineAll)
    {
            numChanged = 0;
            if (examineAll)
                    loop I over all training examples
                            numChanged += examineExample(I)
            else
                    loop I over examples where alpha is not 0 & not C
                            numChanged += examineExample(I)
            if (examineAll == 1)
                    examineAll = 0
            else if (numChanged == 0)
                    examineAll = 1
    }
```

3.6. Evaluation Metrics

The comparative analysis of MLBCAs for sentiment analysis was made by measuring the performance of each algorithm with the help of following parameters.

3.6.1. Confusion matrix

A confusion matrix is a table for analyzing the result of sentiment analysis by using classification algorithms. It deals with how classification algorithm can recognize documents of different sentiment class (Either positive or negative). In order to develop the confusion matrix, the following terms should be considered [17].

- **True Positive (TP):** Positive sentiment documents that are correctly labeled by the MLBCAs for sentiment analysis.
- **True Negative (TN):** Negative sentiment documents that are correctly labeled by MLBCAs for sentiment analysis.
- **False Positive (FP):** Negative sentiment documents that are incorrectly labeled as positive.
- **False Negative (FN):** Positive sentiment documents that are mislabeled as negative.

	Predicted positive sentiment	Predicted negative sentiment
Actual positive sentiment	TP	FN
Actual negative sentiment	FP	TN

Figure 3.2: Confusion Matrix

3.6.2. Accuracy

Accuracy of classification algorithm for sentiment analysis on given data dataset is the percentage of documents in a data set that are correctly classified as positive sentiment or negative sentiment. It also refers to the polarity detection rate of the classification algorithm for sentiment analysis.

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

3.6.3. Precision

Precision refers to the measure of exactness that means what percentage of documents labeled as positive sentiment category are actually such.

$$\text{Precision} = \frac{TP}{TP+FP}$$

3.6.4. Recall

Recall refers to the true positive or positive polarity that means the proportion of positive polarity documents that are correctly identified.

$$\text{Recall} = \frac{TP}{TP+FN}$$

3.6.5. F-Measure

The F-measure combines both measures precision and recall as the harmonic mean.

$$\text{F-measure} = \frac{2 \text{ x precision x recall}}{\text{precsion}+\text{recall}}$$

CHAPTER 4

4. RESULTS ANALYSIS AND COMPARISON

4.1. Result Analysis and Comparison

In this research study, the analysis of all three MLBCAs for sentiment analysis mentioned in section 3.5 of chapter 3 has been compared for all three sentiment labeled sentence datasets mentioned in section 3.1 of chapter 3, which are compared based on four performance criteria's namely accuracy, precision, recall and F-measure. The results have been achieved by using whole test dataset for all three algorithms after training phase. In this research work the value of K- parameter for KNN algorithm has been taken k=9, because for the datasets taken in this research KNN algorithm gave better result on this value of K than other possible value of K.

4.1.1. Performance result of MLBCAs for sentiment analysis on dataset1 and their comparison

The dataset1 (i.e. restaurant review dataset) has 992 tuples in total, but after preprocessing only 702 tuples among them 348 belongs to 1 (i.e., positive sentiment) category and 354 belongs to 0 (i.e., negative sentiment) category. For training only 561 tuples has been taken and remaining 141 (65 belongs to 0 and 76 belongs to 1) for testing purpose.

Table 4.1 shows the classification report that has been obtained after three MLBCAs applied on test dataset that is obtained from dataset1.

	KNN		MNB		SVM	
	Predicted +ve	Predicted -ve	Predicted +ve	Predicted -ve	Predicted +ve	Predicted -ve
Actual +ve	54	22	58	18	60	16
Actual -ve	16	49	12	53	13	52

Table 4.1: confusion matrix on dataset1

Based on the classification report shown in Table 4.1 the calculated summary performance result for the comparison of all three algorithms applied on

dataset1 is shown in Table 4.2. The precision, recall and F-measure value shown in Table 4.2 is the average of precision, recall and F-measure for both sentiment categories.

Algorithms	Accuracy	Precision	Recall	F-Measure
KNN	73%	73%	73%	73%
MNB	78.7%	79%	79%	78.5%
SVM	79.5%	79.5%	79%	79.5%

Table 4.2: Performance result on dataset1

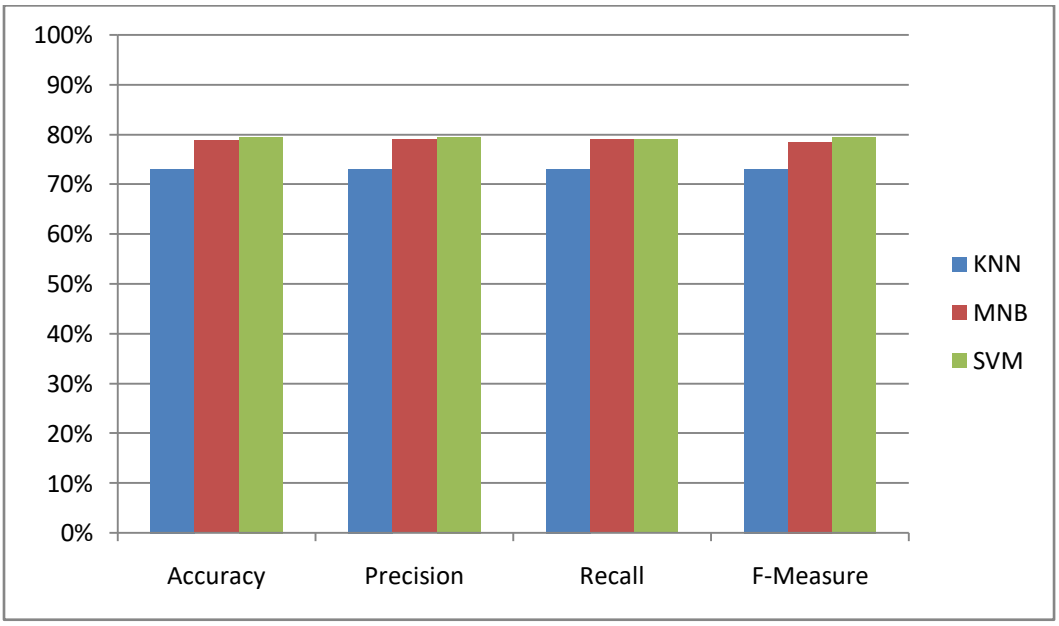

Figure 4.1: Graph of table 4.2

Based on the Figure 4.1, it is clearly seen that the accuracy value of SVM is got high level of 79.5% and KNN got less accuracy of level 73%. In case of precision and the Recall value of implemented SVM for sentiment analysis had got high precision and recall level of 79.5% and 79% respectively. Whereas KNN got less precision and recall level of 73% and 73 % respectively.

Figure 4.1 also shows the F-measure of table 4.2 observed by implemented Machine learning based classification algorithms for sentiment analysis. Again

SVM had outperformed two other compared algorithms with value of 79.5% and KNN had got minimum value of 73%.

4.1.2. Performance result of MLBCAs for sentiment analysis on dataset2 and their comparison

The dataset2 (i.e. Amazon cell phones and accessories review dataset) has 1000 tuples in total, but after preprocessing only 770 tuples among them 387 belongs to 1 (i.e., positive sentiment) category and 383 belongs to 0 (i.e., negative sentiment) category. For training only 616 tuples has been taken and remaining 154 (77 belongs to 0 and 77 belongs to 1) for testing purpose.

Table 4.3 shows the classification report that has been obtained after three MLBCAs applied on test dataset that is obtained from dataset2.

	KNN		MNB		SVM	
	Predicted +ve	Predicted -ve	Predicted +ve	Predicted -ve	Predicted +ve	Predicted -ve
Actual +ve	66	11	65	12	60	17
Actual -ve	25	53	25	52	16	61

Table 4.3: confusion matrix on dataset2

Based on the classification report shown in Table 4.3 the calculated summary performance result for the comparison of all three algorithms applied on dataset2 is shown in Table 4.4. The precision, recall and F-measure value shown in Table 4.4 is also the average of precision, recall and F-measure for both sentiment categories.

Algorithms	Accuracy	Precision	Recall	F-Measure
KNN	77.3%	77.5%	78%	77%
MNB	75.9%	76%	76.5%	76%
SVM	78.6%	78.5%	78.5%	78.5%

Table 4.4: Performance result on dataset2

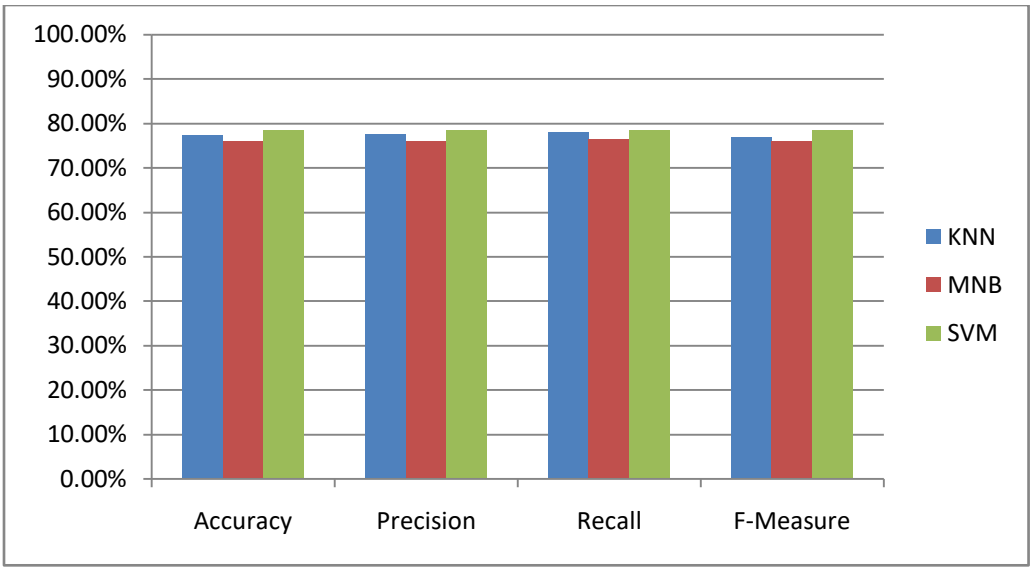

Figure 4.2: Graph of table 4.4

Based on the Figure 4.2, it is clearly seen that the accuracy value of SVM is got high level of 78.6% and MNB got less accuracy of level 75.9%. In case of precision and the Recall value of implemented SVM for sentiment analysis had got high precision and recall level of 78.5% and 78.5% respectively. Whereas MNB got less precision and recall level of 76% and 76.5 % respectively.

Figure 4.2 also show the F-measure of table 4.4 observed by implemented MLBCAs for sentiment analysis. Again SVM had outperformed other two compared algorithms with value of 78.5% and MNB had got minimum value of 76%.

4.1.3. Performance result of MLBCAs for sentiment analysis on dataset3 and their comparison

The dataset3 (i.e. IMDB movie review dataset) has 25000 tuples in total, but after preprocessing all remains as it is among them 12500 belongs to 1 (i.e., positive sentiment) category and 12500 belongs to 0 (i.e., negative sentiment) category. For training only 20000 tuples has been taken and remaining 5000 (2548 belongs to 0 and 2452 belongs to 1) for testing purpose.

Table 4.5 shows the classification report that has been obtained after three MLBCAs applied on test dataset that is obtained from dataset3.

	KNN		MNB		SVM	
	Predicted +ve	Predicted -ve	Predicted +ve	Predicted -ve	Predicted +ve	Predicted -ve
Actual +ve	1911	541	2163	289	2213	239
Actual -ve	497	2051	283	2265	288	2260

Table 4.5: confusion matrix on dataset3

Based on the classification report shown in Table 4.5 the calculated summary performance result for the comparison of all three algorithms applied on dataset3 is shown in Table 4.6. The precision, recall and F-measure value shown in Table 4.6 is also the average of precision, recall and F-measure for both sentiment categories.

Algorithms	Accuracy	Precision	Recall	F-Measure
KNN	79.24%	79%	79%	79.5%
MNB	88.56%	88.5%	88.5%	88.5%
SVM	89.46%	89.5%	89%	89.5%

Table 4.6: performance result on dataset3

Based on the Figure 4.3, it is clearly seen that the accuracy value of SVM is got high level of 89.46% and KNN got less accuracy of level 79.24%. In case of precision and the Recall value of implemented SVM for sentiment analysis had got high precision and recall level of 89.5% and 89% respectively. Whereas KNN got less precision and recall level of 79% and 79 % respectively.

Figure 4.3 also show the F-measure of table 4.6 observed by implemented MLBCAs for sentiment analysis. Again SVM had outperformed other two

compared algorithms with value of 89.5% and KNN had got minimum value of 79.5%.

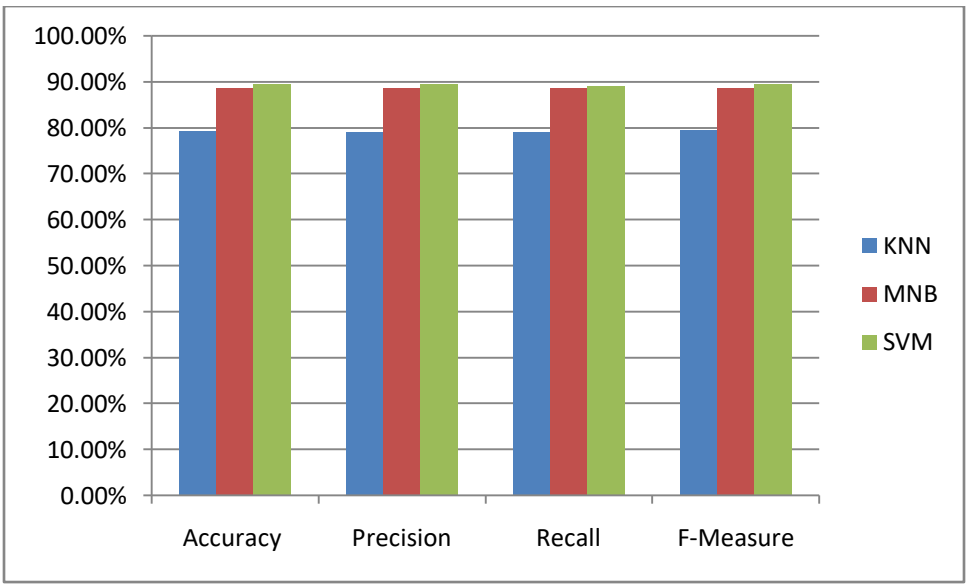

Figure 4.3: Graph of table 4.6

CHAPTER 5

5. CONCLUSION AND FUTURE ENHANCEMENT

5.1. Conclusion

In this research, the comparative analysis of machine learning based classification algorithms for sentiment analysis (i.e. MNB, KNN and SVM) using various performance measure parameter like accuracy, precision, recall and F-measures over the three different dataset (i.e. Yelp Restaurant review dataset, Amazon cell phones and accessories review dataset and IMDB movie review dataset) with different size are evaluated.

From this comparative study it has been found that, in Yelp Restaurant review dataset SVM had got higher performance in every aspect, whereas MNB and KNN had got less performance in every aspect as compared to SVM. The SVM algorithm has accuracy, precision, recall and F-measure with level of 79.5%, 79.5%, 79% and 79.5% respectively. In Amazon cell phones and accessories review dataset again SVM had got higher performance in every aspect, whereas KNN and MNB had got less performance in every aspect as compared to SVM. The SVM algorithm has accuracy, precision, recall and F-measure with level of 78.6%, 78.5%, 78.5% and 78.5% respectively. Also in the IMDB movie review dataset SVM had got higher performance in every aspect, whereas MNB and KNN had got less performance in every aspect as compared to SVM. The SVM algorithm has accuracy, precision, recall and F-measure with level of 89.46%, 89.5%, 89% and 89.5% respectively.

Therefore, it has been concluded that, on balance datasets, SVM algorithm has predicted better Sentiment category result than other machine learning based classification algorithms for sentiment analysis studied for all three datasets.

5.2. Future Enhancement

In this research study only three traditional machine learning based classification algorithm has been study for sentiment polarity detection in three sentiment labeled

sentence datasets. In the future more algorithms from the classification, clustering, and deep learning approach can be incorporated for further study to the studied datasets or other datasets which have text as well as image, audio or video type. Moreover some algorithms can be customized for the specific domain so that sentiment analysis could have more accurate and reliable results.

References

[1]A. Z.H. Khan, et al, "Sentiment Analysis Using Support Vector Machine", *International Journal of Advanced Research in Computer Science and Software Engineering,* Volume 5, Issue 4, April 2015.

[2] C L. Rakshitha, S. Gowrishankar , " Machine Learning based Analysis of Twitter Data to Determine a Person's Mental Health Intuitive Wellbeing", *International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13,* Number 21 (2018).

[3] M. Ghosh, G. Sanyal, "An ensemble approach to stabilize the features for multi-domain sentiment analysis using supervised machine learning", *Journal of big data,* 2018.

[4] G. Isabelle, "Analysis on Opinion Mining Using Combining Lexicon-Based Method and Multinomial Naive Bayes", *International Conference on Industrial Enterprise and System Engineering (IcoIESE),* 2018.

[5] S. M. Vohra, J. B. Teraiya, "a comparative study of sentiment analysis techniques", *Journal of information, knowledge and research in computer engineering volume – 02,* issue – 02 page 313.

[6] M. Annett, G. Kondrak, "A Comparison of Sentiment Analysis Techniques: Polarizing Movie Blogs", *Department of Computing Science, University of Alberta*.

[7] Lopes et al., "Automatic cluster labeling through Artificial Neural Networks", *International Joint Conference on Neural Networks (IJCNN),*(2014).

[8] B. Pang et al., "Thumbs up? Sentiment Classification using Machine Learning Techniques", *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP),* Philadelphia, July 2002.

[9] H. Rahmath, T. Ahmad, "Sentiment analysis techniques: A comparative study", *IJCEM International Journal of Computational Engineering & Management, Vol. 17 Issue 4,* July 2014.

[10] A. Kathuria, s. Upadhyay., " A Novel Review of various sentiment analysis technique", *International Journal of Computer Science and Mobile Computing (IJCSMC), Vol. 6, Issue. 4*, April 2017.

[11] E. Elmurngi, A. Gherbi, "Fake Reviews Detection on Movie Reviews through Sentiment Analysis Using Supervised Learning Techniques", *International Journal on Advances in Systems and Measurements, vol 11*, no 1 & 2, year 2018.

[12] Z. Jianqiang, G. Xiaolin, "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis *", IEEE Access*, February 22, 2017.

[13] S.D. Sarkar, S. Goswami, "Empirical Study on Filter based Feature Selection Methods for Text Classification", *International Journal of Computer Applications (0975 – 8887) ,Volume 81 – No.6*, November 2013.

[14] K. Huda et al., "Classification Technique for Sentiment Analysis of Twitter Data", *International Journal of Advanced Research in Computer Science,* Volume 8, No. 5, May-June 2017.

[15] B. Trstenjak et al., "KNN with TF-IDF Based Framework for Text Categorization", *ScienceDirect 24th DAAAM International Symposium on Intelligent Manufacturing and Automation,* 2013.

[16] S. Ahmed, et al."Detection of Sentiment Polarity of Unstructured Multi-Language Text from Social Media", *(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9*, No. 7, 2018.

[17] M. Mowafy et al., "An Efficient Classification Model for Unstructured Text Document", *American Journal of Computer Science and Information Technology, Vol.6 No.1:16,* 2018.

[18] U. I. Larasati, "Improve the Accuracy of Support Vector Machine Using Chi Square Statistic and Term Frequency Inverse Document Frequency on Movie Review Sentiment Analysis", *Scientific Journal of Informatics Vol. 6, No. 1*, May 2019.

[19] https://www.kaggle.com/marklvl/sentiment-labelled-sentences-data-set.

[20] https://www.apa.org/science/about/psa/2019/07/python-research.

[21]https://semanti.ca/blog/?recommended-ide-for-data-scientists-and-machine-learning-engineers.

[12] M. Abbas, ”Multinomial Naive Bayes Classification Model for Sentiment Analysis”, *IJCSNS International Journal of Computer Science and Network Security, VOL.19 No.3,* March 2019.

[23] D. Jurafsky, J. H. Martin, “Naive Bayes and Sentiment Classification”, *Speech and Language Processing,* 2019.

[24] R. Alhutaish, N. Omar, “Arabic Text Classification using K-Nearest Neighbour Algorithm”, *The International Arab Journal of Information Technology, Vol. 12, No. 2,* March 2015.

[25] H. Deka, P. Sarma, “A Machine Learning Approach for Text and Document Mining”, *International Journal of Computer Science Engineering* (IJCSE).

[26]S. Tan,”An effective refinement strategy for KNN text classifier”,*Software Department, Institute of Computing Technology, Chinese Academy of Sciences,*2006.

[27] J. C. Platt, “Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines”, *Microsoft Research,* April 21, 1998.

[28] K.P Bennett, C. Champbell, “Support Vector Machine: Hype”, *department of Engineering and Mathematics Bristol University UK.*

Appendix

1. Dataset1 sample (Yelp restaurant review dataset)

Review	Sentiment
Not tasty and the texture was just nasty.	0
The selection on the menu was great and so were the prices.	1
Now I am getting angry and I want my damn pho.	0
The fries were great too.	1
A great touch.	1
Service was very prompt.	1
Would not go back.	0
I tried the Cape Cod ravoli	1
I was disgusted because I was pretty sure that was human hair.	0
I was shocked because no signs indicate cash only.	0
Highly recommended.	1
Waitress was a little slow in service.	0
This place is not worth your time	0
Service is also cute.	1
I could care less... The interior is just beautiful.	1
So they performed.	1
I found this place by accident and I could not be happier.	1
Poor service	0
My first visit to Hiro was a delight!	1
Service sucks.	0
On a positive note	1
Frozen pucks of disgust	0
The only thing I did like was the prime rib and dessert section.	1
It's too bad the food is so damn generic.	0
Their chow mein is so good!	1
Service is fantastic	1

2. Dataset2 sample (Amazon cell phones and accessories review dataset)

Review	Sentiment
Highly recommend for anyone who has a blue tooth phone.	1
The design is very odd	0
What a waste of money and time!.	0
He was very impressed when going from the original battery to the extended battery.	1
I bought this to use with my Kindle Fire and absolutely loved it!	1
The commercials are the most misleading.	0
I bought it for my mother and she had a problem with the battery.	0
Great Pocket PC / phone combination.	1
I didn't think that the instructions provided were helpful to me.	0
Doesn't hold charge.	0
This product is ideal for people like me whose ears are very sensitive.	1
It is unusable in a moving car at freeway speed.	0
I have two more years left in this contract and I hate this phone.	0
The case is great and works fine with the 680.	1
Worthless product.	0
Nice headset priced right.	1
I only hear garbage for audio.	0
Excellent Bluetooth headset.	1
Not loud enough and doesn't turn on like it should.	0
Good protection and does not make phone too bulky.	1
This phone is pretty sturdy and I've never had any large problems with it.	1
This case seems well made.	1
Disappointed with battery.	0
It has all the features I want	1

3. Dataset3 sample (IMDB Movie review dataset)

Review	Sentiment
The acting company exhibits all the emotions of the play itself	1
brilliant performance by Timothy Spall	1
simply for the complete absurdity	0
In point of fact, the whole movie is really disturbing	0
this sequel doesn't match up to its predecessor	0
This movie is one among the very few Indian movies	1
definitely keep some viewers glued to the screen	1
I would love to have that two hours of my life back	0
a loosely constructed script	0
Wow! It was amazing!	1
disappointing movie	0
without a doubt, be the biggest waste of time	0
I liked the film	1
why aren't other films like this?	1
It is the worst film ever	0
I highly recommend this movie	1
thoroughly uninteresting	0
a delightful and original surprise	1
A pleasant simple	1
romantic-comedy that deserves to be seen by more	1
Full of entertainment	1
Full of fun	1
This is a film with nothing	0
one of the best Italian horror movies	1
I hated it and I'm not afraid to say so	0
I loved the episode	1
I'm glad I saw it	1

this movie was absolutely awful	0
Okay, sorry, but I loved this movie	1
heart-breaking and uplifting	1
this movie deserves an audience	1
This is pretty bad	0
DO NOT waste your time on this 'film	0
movie was o.k. but it could have been much better	0
The acting is exceptionally good,	1
the location filming and photography is at time breathtaking	1
had excellent acting and writing	1
IT IS So Sad	0
A somewhat dull made for tv movie	0
the playful scene	1
a beautifully told story	1
the story is ridicules	0
the acting by the four main characters is solid	1
The two lead actors are awesome	1
worst film of all time'	0
it is a fun movie that should be seen at least once	1
It is so bad, it isn't even good for being bad	0
most fantastic films	1
It's very un-funny	0
I totally agree that	1
a fantastic film	1
really work well	1
the worst movie I've ever seen	0
bad selection of the actors	0

Code:

```
import numpy as np
import pandas as pd
import os
from sklearn.model_selection import train_test_split
import re
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.stem import PorterStemmer
from nltk.stem import LancasterStemmer
import nltk
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

#nltk.download('punkt')
#nltk.download('wordnet')

def selected_Analysis(up, algorithm):
    if up==1:
        data = pd.read_csv('yelp_labelled.csv', delimiter=",")
      # data.head()
    elif up==2:
        data = pd.read_csv('amazon_cells_labelled.csv', delimiter=",")
    else:
        data = pd.read_csv('labeledTrainData.tsv', delimiter="\t")
        data.head()
```

```
    # dropping extra column in data
    data.drop(data.columns[data.columns.str.contains('unnamed', case=False)], axis=1,
inplace=True)

    # Removing non numberic row in a Category column
    data = data[pd.to_numeric(data['Category'], errors='coerce').notnull()]

    data = data.dropna()

    data.reset_index(inplace=True)

    #print(data.head())

    review_data = data['Review'][0]

    #print(review_data)
    return_review = review_preprocessing(review_data, stem=True, lemm=True)
    #print("  ")
    #print(return_review)

    data_set = []
    data_label = data['Category']
    for Review in data['Review']:
        data_set.append(review_preprocessing(Review, stem=True, lemm=True))

    data_set = np.array(data_set)

    #print(data_set.shape)

    # split data to train & test set
```

```
    x_train, x_test, y_train, y_test = train_test_split(data_set, data_label, test_size=0.2,
random_state=0)

   # print(x_train.shape)
    #print(x_test.shape)

    tfidf = TfidfVectorizer(
        ngram_range=(1, 4),
        use_idf=1,
        smooth_idf=1,
        stop_words='english')
    data_train_count = tfidf.fit_transform(x_train)
    test_train_count = tfidf.transform(x_test)

    if (algorithm==1):
        clf_model = MultinomialNB()
        clf_model.fit(data_train_count, y_train)
        pred = clf_model.predict(test_train_count)
        #print(pred)
    elif algorithm == 2:
        clf_model = KNeighborsClassifier(n_neighbors=9, algorithm='auto', metric='cosine')
        clf_model = clf_model.fit(data_train_count, y_train)
        pred = clf_model.predict(test_train_count)
        #print(pred)
    elif algorithm ==3:
        clf_model = LinearSVC()
        clf_model.fit(data_train_count, y_train)
        pred = clf_model.predict(test_train_count)
        #print(pred)
    else:
```

```
    print("select algorithm properly")

if up==1 and algorithm==1:
    print("Classification Report For MNB Classifier on Yelp Resturant Review Dataset:")
elif up==1 and algorithm==2:
    print("Classification Report For KNN Classifier on Yelp Resturant Review Dataset:")
elif up == 1 and algorithm == 3:
    print("Classification Report For SVM Classifier on Yelp Resturant Review Dataset:")
elif up == 2 and algorithm == 1:
    print("Classification Report For MNB Classifier on Amazon Review Dataset:")
elif up == 2 and algorithm == 2:
    print("Classification Report For KNN Classifier on Amazon Review Dataset:")
elif up == 2 and algorithm == 3:
    print("Classification Report For SVM Classifier on Amazon Review Dataset:")
elif up == 3 and algorithm == 1:
    print("Classification Report For MNB Classifier on Movie Review Dataset:")
elif up == 3 and algorithm == 2:
    print("Classification Report For KNN Classifier on Movie Review Dataset:")
elif up == 3 and algorithm == 3:
    print("Classification Report For SVM Classifier on Movie Review Dataset:")
else:
    print("select data set and algorithm prperly")



print("Acccuracy:", accuracy_score(pred, y_test))

print("Classification report:")
print(classification_report(pred, y_test))

print("Confusion Matrix:")
```

```
    print(confusion_matrix(pred, y_test))



lancaster=LancasterStemmer()
lemmatizer = WordNetLemmatizer()
def review_preprocessing(Review, stem = True, lemm = True):
    revised_words = []
    review_text = re.sub("[^a-zA-Z]"," ", Review)
    review_text = review_text.lower()
    #Stemming or Lemmatization
    for word in word_tokenize(review_text):
        if stem:
            word = lemmatizer.lemmatize(word)
        if lemm:
            word = lancaster.stem(word)
        revised_words.append(word)
    return_words = " ".join(revised_words)
    return(return_words)

def main():
    dataset=int(input("Enter the choice for dataset 1 or 2 or 3 for Yelp Resturant,Amazon and Movie Review Datasets respectively:"))
    algorithm=int(input("Enter the choice for algorithm 1 for MNB, 2 for KNN and 3 for SVM:"))
    selected_Analysis(dataset,algorithm)

if __name__ == "__main__":
    main()
```

Sample Run and their corresponding outputs:

1. RUN1:

Enter the choice for dataset 1 or 2 or 3 for Yelp Resturant,Amazon and Movie Review Datasets respectively:1

Enter the choice for algorithm 1 for MNB, 2 for KNN and 3 for SVM:1

Classification Report For MNB Classifier on Yelp Resturant Review Dataset:

Accuracy: 0.7872340425531915

Classification report:

	precision	recall	f1-score	support
0	0.82	0.75	0.78	71
1	0.76	0.83	0.79	70
micro avg	0.79	0.79	0.79	141
macro avg	0.79	0.79	0.79	141
weighted avg	0.79	0.79	0.79	141

Confusion Matrix:

[[53 18]
[12 58]]

2. Run 2:

Enter the choice for dataset 1 or 2 or 3 for Yelp Resturant,Amazon and Movie Review Datasets respectively:1

Enter the choice for algorithm 1 for MNB, 2 for KNN and 3 for SVM:2

Classification Report For KNN Classifier on Yelp Resturant Review Dataset:

Accuracy: 0.7304964539007093

Classification report:

	precision	recall	f1-score	support
0	0.75	0.69	0.72	71

1	0.71	0.77	0.74	70
micro avg	0.73	0.73	0.73	141
macro avg	0.73	0.73	0.73	141
weighted avg	0.73	0.73	0.73	141

Confusion Matrix:

[[49 22]
[16 54]]

3. Run 3:

Enter the choice for dataset 1 or 2 or 3 for Yelp Resturant,Amazon and Movie Review Datasets respectively:1

Enter the choice for algorithm 1 for MNB, 2 for KNN and 3 for SVM:3

Classification Report For SVM Classifier on Yelp Resturant Review Dataset:

Accuracy: 0.7943262411347518

Classification report:

	precision	recall	f1-score	support
0	0.80	0.76	0.78	68
1	0.79	0.82	0.81	73
micro avg	0.79	0.79	0.79	141
macro avg	0.79	0.79	0.79	141
weighted avg	0.79	0.79	0.79	141

Confusion Matrix:

[[52 16]
[13 60]]

4. Run 4:

Enter the choice for dataset 1 or 2 or 3 for Yelp Resturant,Amazon and Movie Review Datasets respectively:2

Enter the choice for algorithm 1 for MNB, 2 for KNN and 3 for SVM:1

Classification Report For MNB Classifier on Amazon Review Dataset:

Accuracy: 0.7597402597402597

Classification report:

	precision	recall	f1-score	support
0	0.68	0.81	0.74	64
1	0.84	0.72	0.78	90
micro avg	0.76	0.76	0.76	154
macro avg	0.76	0.77	0.76	154
weighted avg	0.77	0.76	0.76	154

Confusion Matrix:

[[52 12]
[25 65]]

5. Run5:

Enter the choice for dataset 1 or 2 or 3 for Yelp Resturant,Amazon and Movie Review Datasets respectively:2

Enter the choice for algorithm 1 for MNB, 2 for KNN and 3 for SVM:2

Classification Report For KNN Classifier on Amazon Review Dataset:

Accuracy: 0.7727272727272727

Classification report:

	precision	recall	f1-score	support
0	0.69	0.83	0.75	64
1	0.86	0.73	0.79	90

micro avg	0.77	0.77	0.77	154
macro avg	0.77	0.78	0.77	154
weighted avg	0.79	0.77	0.77	154

Confusion Matrix:

[[53 11]
[24 66]]

6. Run 6:

Enter the choice for dataset 1 or 2 or 3 for Yelp Resturant,Amazon and Movie Review Datasets respectively:2

Enter the choice for algorithm 1 for MNB, 2 for KNN and 3 for SVM:3

Classification Report For SVM Classifier on Amazon Review Dataset:

Accuracy: 0.7857142857142857

Classification report:

	precision	recall	f1-score	support
0	0.79	0.78	0.79	78
1	0.78	0.79	0.78	76
micro avg	0.79	0.79	0.79	154
macro avg	0.79	0.79	0.79	154
weighted avg	0.79	0.79	0.79	154

Confusion Matrix:

[[61 17]
[16 60]]

7. Run7:

Enter the choice for dataset 1 or 2 or 3 for Yelp Resturant,Amazon and Movie Review Datasets respectively:3

Enter the choice for algorithm 1 for MNB, 2 for KNN and 3 for SVM:1

Classification Report For MNB Classifier on Movie Review Dataset:

Accuracy: 0.8856

Classification report:

	precision	recall	f1-score	support
0	0.89	0.89	0.89	2554
1	0.88	0.88	0.88	2446
micro avg	0.89	0.89	0.89	5000
macro avg	0.89	0.89	0.89	5000
weighted avg	0.89	0.89	0.89	5000

Confusion Matrix:

[[2265 289]

[283 2163]]

8. Run 8:

Enter the choice for dataset 1 or 2 or 3 for Yelp Resturant,Amazon and Movie Review Datasets respectively:3

Enter the choice for algorithm 1 for MNB, 2 for KNN and 3 for SVM:2

Classification Report For KNN Classifier on Movie Review Dataset:

Accuracy: 0.7924

Classification report:

	precision	recall	f1-score	support
0	0.80	0.79	0.80	2592
1	0.78	0.79	0.79	2408

micro avg	0.79	0.79	0.79	5000
macro avg	0.79	0.79	0.79	5000
weighted avg	0.79	0.79	0.79	5000

Confusion Matrix:

[[2051 541]
[497 1911]]

9. Run 9:

Enter the choice for dataset 1 or 2 or 3 for Yelp Resturant,Amazon and Movie Review Datasets respectively:3

Enter the choice for algorithm 1 for MNB, 2 for KNN and 3 for SVM:3

Classification Report For SVM Classifier on Movie Review Dataset:

Accuracy: 0.8946

Classification report:

	precision	recall	f1-score	support
0	0.89	0.90	0.90	2499
1	0.90	0.88	0.89	2501
micro avg	0.89	0.89	0.89	5000
macro avg	0.89	0.89	0.89	5000
weighted avg	0.89	0.89	0.89	5000

Confusion Matrix:

[[2260 239]
[288 2213]]