Opinion Mining of Online Users' Comments
Using Natural Language Processing and Machine Learning

by

Anahita (Elham) Pazooki

A thesis submitted in partial fulfillment

of the requirements for the degree of

MSc Computational Sciences

The Faculty of Graduate Studies

Laurentian University

Sudbury, Ontario, Canada

THESIS DEFENCE COMMITTEE/COMITÉ DE SOUTENANCE DE THÈSE
Laurentian Université/Université Laurentienne
Faculty of Graduate Studies/Faculté des études supérieures

Title of Thesis Titre de la thèse	Opinion Mining of Online Users' Comments Using Natural Language Processing and Machine Learning	
Name of Candidate Nom du candidat	Pazooki, Anahita (Elham)	
Degree Diplôme	Master of Science	
Department/Program Département/Programme	Computational Science	Date of Defence Date de la soutenance August 298th, 2020

APPROVED/APPROUVÉ

Thesis Examiners/Examinateurs de thèse:

Dr. Kalpdrum Passi
(Supervisor/Directeur(trice) de thèse)

Dr. Ratvinder Grewal
(Committee member/Membre du comité)

Dr. Julia Johnson
(Committee member/Membre du comité)

Dr. Jinan Fiaidhi
(External Examiner/Examinateur externe)

Approved for the Faculty of Graduate Studies
Approuvé pour la Faculté des études supérieures
Dr. David Lesbarrères
Monsieur David Lesbarrères
Dean, Faculty of Graduate Studies
Doyen, Faculté des études supérieures

ACCESSIBILITY CLAUSE AND PERMISSION TO USE

Abstract

With the widespread popularity of World Wide Web, increasing number of people are active on social media and websites to post their opinions towards products or special events or to make decisions based on the opinions and experiences of people on social media. These Online opinions are unstructured or structured textual data containing insignificant as well as significant information which has attracted attention of researchers to extract knowledge from such textual data. Opinion mining and Natural Language Processing (NLP) techniques help to find information through the huge number of reviews in the form of unstructured comments. In this research a model is proposed for classification of online user's feedback and opinions to improve the accuracy and precision of the classification in comparison to the existing research on the same dataset. More-precisely, in this research, Natural Language Processing (NLP) techniques as well as various supervised machine learning techniques are used to classify users' opinions. The performances of all the classifiers are evaluated to find the best performance. The data set contains 689 comments extracted from the users' comments from Amazon.com, collected and annotated by Minqing Hu and Bing Liu. The selected comments are about the product "Speakers" on Amazon.com. Each comment is written by one user and it has a certain label that shows the author's desire to comment. This label can be classified as "positive", "negative" or "neutral". The data is provided in the form of XML file, a semi-structured format. The opinions are processed using natural language processing techniques, for instance by removing punctuations, removing URLs, removing numbers, removing spaces, removing stop-words, and their features are extracted using natural language processing techniques, for example, Word Tokenization, Stemming and Bag of words and Bag of N-grams and Term Frequency-Inverse

Document Frequency (TF_IDF). The proposed method was implemented using Python programming language and Natural Language Toolkit (NLTK) and other libraries in python. The proposed model gave a peak of 88% precision by Random Forest with 140 trees and bigram feature space. Also, Random Forest, Gradient Boosting, Artificial Neutral Network, and SVM gave 87% precision for trigram feature space.

Keywords

Data Mining, Opinion Mining, Natural Language Processing (NLP), Data pre-processing, Word Tokenization, Stemming, Term Frequency-Inverse Document Frequency, Supervised Machine Learning, Random Forest, Gradient Boosting, decision trees, SVMs, Gini-Index, Artificial Neural Network

Table of Contents

List of Tables

List of Figures

Acknowledgements

I want to start by first acknowledging my thesis supervisor Dr. Kalpdrum Passi. He is very respectful and dedicated professor, and he has always supported me in my research whenever I needed help. He has always welcomed any problems or questions that I had regarding the research and writing process of my thesis. Without his guidance, my thesis would not have taken this shape.

Moreover, I would also like to thank the committee members that were involved in the validation process of my thesis.

Additionally, I like to thank professors Minqing Hu and Bing Liu who annotated the data set and share it, which I used in this research.

I dedicate this thesis to my dedicated mother, father and brother who were the greatest inspiration to me to pursue higher education. Although they live in my home country and I could not see them about one year. They are always following my progress and encourage me to try to achieve my academic goal.

Preface

In most social networks, it is possible to create comments about goods, services, etc. It is important for a social network or website to provide its services in accordance with the interests of users. So user feedback is very important, because these ideas are directly effective in providing the right services. Therefore, analyzing customer feedback to extract valuable information is one of the most important and challenging issues. To analyze this vast and non-structured information, the field of Opinion Mining, which is a special type of Data Mining, is proposed. Opinion Mining means that users' opinions can be attributed to a positive or negative category. Their features are used for summarizing and extracting hidden valuable information.

Abbreviations

NLP	Natural Language Processing
NLTK	Natural Language Toolkit
TF-IDF	Term Frequency-Inverse Document Frequency
BOW	Bag of Words
SVM	Support Vector Machine
NN	Neural Network
URL	Uniform Resource Locator
GB	Gradient Boosting

Chapter 1

Introduction

1.1 Objectives of research

For a social network, websites, articles, service reviews and online stores providing facilities according to user's interest is an important issue. For the owners of these web pages and social networks, users' opinion is very important. Because the analysis of these opinions can help the managers provide suitable facilities, and improve future performances which can increase the users' satisfaction. However, the volume of users' comments is increasing and the manual analysis of these opinions is very difficult [Hemmatian & Sohrabi, 2017]. As a result, recommending a suitable method which can improve the accuracy of classification for analyzing the increasing amount of user's opinions is essential. Furthermore, the analysis of users' opinion by humans need much time and it can be very costly, because of these reasons there is a need for a new, effective and automatic method which can search a large amount of user's opinions every day. In this research, a suitable solution is recommended to solve this problem.

Goals of this research are listed below

- Recommending a suitable method for analysis of online user's opinions.
- Try to increase accuracy of classification of online user's opinions using recommended method.

1.2 Data-Mining Concept

The process of performing a computer-based methodology which includes new techniques to extract useful knowledge from data is called data mining. Data mining is the search for novel, noteworthy, and nontrivial information in huge volumes of data. The progress in modern technologies of computers, including data acquisition, storage technology and processing a huge amount of data, along with limited capabilities of humans in analyzing big databases have attracted scientists and researchers to search about extracting useful and hidden knowledge from huge data sets [Kantardzic , 2003 & 2011].

Because of growth in size and complexity of data sets, appropriate and sophisticated tools are needed to analyze the available data and achieve the knowledge [Kantardzic, 2003 & 2011]. Extracting valuable information from the mass of data and using it for organizational purposes needs advanced methods. These interests in finding the hidden knowledge from the data have been the cause in advances in data mining [Shahbaz, Masood, Shaheen & Khan, 2010].

1.3 Data Mining Process

Generally, data-mining problems have the following steps, namely stating the problem and making hypothesis, collecting the data, preprocessing the data, making the model and estimating the model. In Data Collection Stage, useful data are collected and stored. Data are usually collected from the existing databases, data warehouses. The old stored data is the only source of information to start the whole data mining process and therefore suggest the results according to the latest data. In data preprocessing stage, scaling, encoding, and selecting features, dimension reduction is done.

In the model manufacturing stage, a successful learning Process from the extracted or collected data is performed to develop an appropriate model. The model evaluation stage allows the developed data mining models to be evaluated. Finally, the results of the data mining process are used to optimize the system, gain knowledge about the system, or predict its behavior. Figure 1.1 shows that in a data mining process, the problem is first defined, and then the data is collected. This data is pre-processed and based on it, a model is created. Finally, this model is evaluated [Kantardzic, 2003 & 2011].

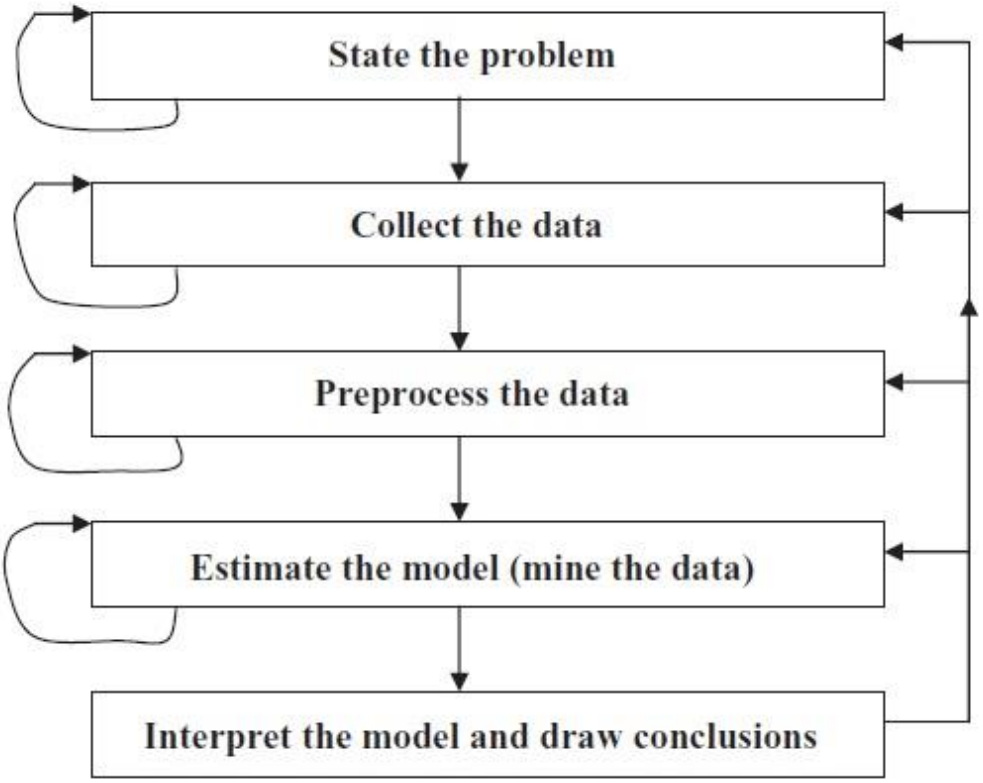
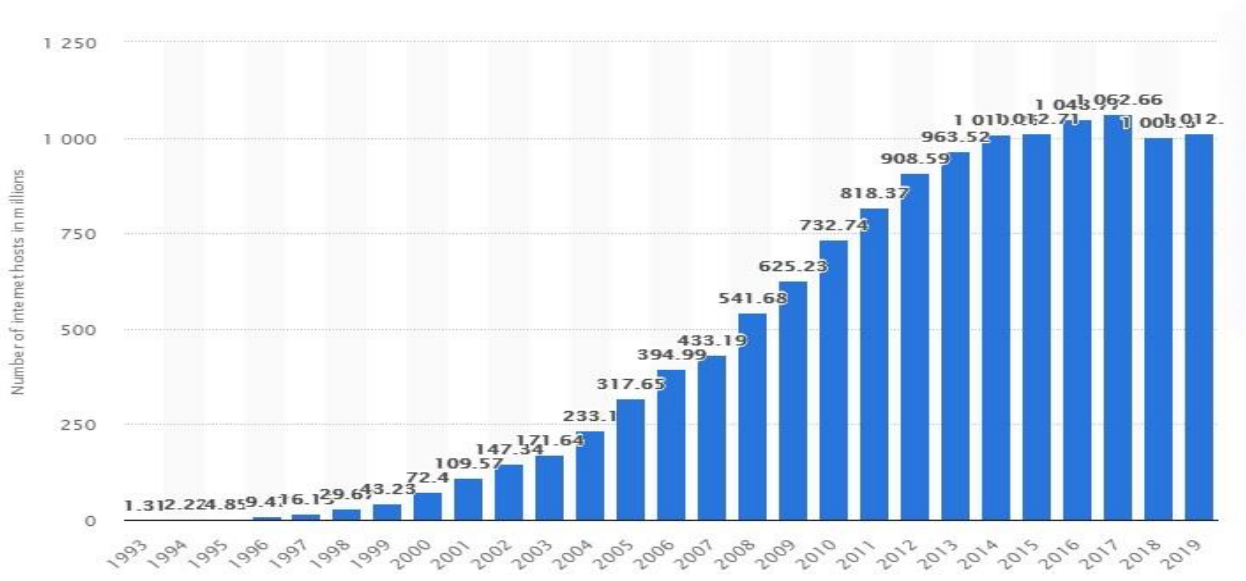


Figure 1.1: Data-mining process [Kantardzic, 2003 & 2011]

1.4 Opinion Mining

People's opinions have always been an important piece of information for the decision-making process. For instance, in the past, traditionally people asked their friends' views about products before shopping. But now with the widespread use of World Wide Web, increasing number of people make decisions based on the opinions and experiences of people via internet. According to two surveys, 81 percent of the Internet users (or 60 percent of Americans) have done online research about a product [Kantardzic, 2003 & 2011]. So, one of the most important factors in the success of services is the review of user feedback. Today, high percent of knowledge is stored in text, documents, and other media forms such as video and audio [Kantardzic, 2003 & 2011].

An illustrative example is given in Figure 1.2, where we can see a dramatic increase in Internet hosts from 1993 to 2017 then a little decrease from 2017 to 2019.



[Source: https://www.statista.com/statistics/264473/number-of-internet-hosts-in-the-domain-name-system/]

Figure 1.2: Growth of Internet hosts

If we look at these documents from the perspective of computer science, they all are unstructured [Pang & Lee, 2008]. Therefore, opinion mining technologies are needed for extracting opinions from the unstructured document. Other names of opinion mining are analysis of opinion, opinion classification, and sentiment analysis [Pang & Lee, 2008].

Opinion mining attracted attention of researchers due to its applications in different fields. Collecting opinions of people about products, social, political events and problems through the Web is becoming increasingly popular. Users' feedback is useful for the public and for companies when they need to make important decisions. Opinion mining is a way to find information through search engines, Web blogs and social networks. Because of the huge number of reviews in the form of unstructured comments, it is unattainable to summarize the huge amount of information manually. Accordingly, efficient computational techniques are needed for mining and summarizing the unstructured reviews from data sets and Web documents [Khan, Baharudin & Ullah, 2014].

1.4.1 Natural Language Processing

Natural Language Processing consists of different tasks such as text and speech processing, morphological analysis, syntactic analysis, lexical semantics, and relational semantics. The techniques used in this research consist of removing punctuations, stop-words, Word Tokenization, Stemming and Term Frequency-Inverse Document Frequency (TF_IDF). These techniques are used to preprocess the textual data to extract the features from the dataset.

1.5 Categorizing machine learning techniques towards opinion classification

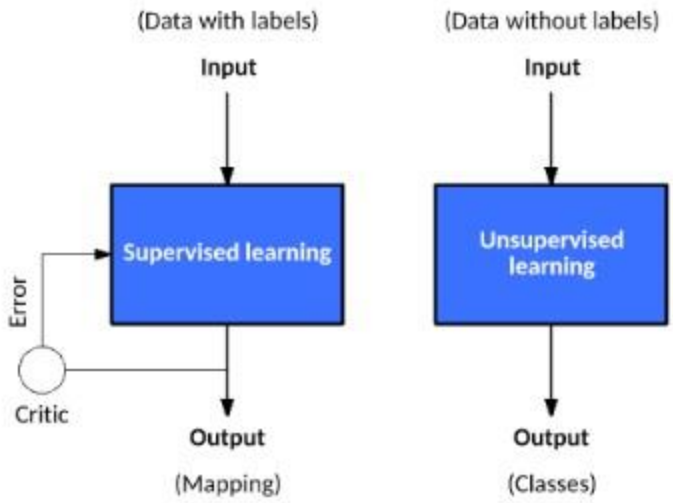
Several approaches have been used for opinion classification. These approaches can be divided into two main categories: supervised and unsupervised approaches. The semi-supervised approach, however, have also been used by some researchers.

1.5.1 Supervised learning methods

The supervised learning methods need the data set which is manually labeled. In this approach, a machine-learning model is trained on that data set to classify opinions. Supervised techniques lead to good results for feature extraction, but it needs the preparation of training sets manually, which can be time consuming and sometimes it is dependent on the domain. The most widely used supervised techniques are support vector machine (SVM), decision trees, Random Forest, K-nearest neighbor (KNN), neural network, and Naı¨ve Bayesian classifiers.

1.5.2 Unsupervised learning methods

In contrast to supervised learning methods, unsupervised techniques do not need the labeled data, and they automatically predict product features based on syntactic patterns and semantic correlation of an area to extract product features from opinions [Khan, Baharudin & Ullah, 2014]. One popular example of unsupervised learning algorithm is k-means for clustering analysis. Figure 1.3 shows an example of supervised and unsupervised learning.



[Source: https://developer.ibm.com/articles/cc-models-machine-learning/]

Figure 1.3: supervised and unsupervised learning

1.6 Structure of the thesis

After stating the generalities of the research topic in Chapter 1, Chapter 2 covers literature survey of opinion mining and techniques. Chapter 3 covers methods and materials used for opinion mining. Data used in the research, preprocessing of the data, methods used for opinion mining is discussed in this chapter. Chapter 4 presents the results and discussion of the performance of the classifiers. Chapter 5 concludes the research and discusses the findings and future work.

Chapter 2

Literature Review

2.1 Introduction

This chapter provides a review of literature on opinion mining, advantages and drawbacks of the used methods. Furthermore, a brief overview of all techniques in this area is summarized.

2.2 Related Work

On the Internet, we deal with a mass volume of data which is not structured, or is semi-structured. For example: feedback from surveys, complaints sent via email, comments about products on the different websites. These comments are available in various forms on different sites, they can be found on commercial sites selling products (such as Amazon) with full descriptions and professional comment sites (such as www.zdnet .com, www.dpreview.com, and www.cnet.com), YouTube, and social media [Chaovalit& Zhou, 2005]. These comments are usually in unstructured form. Therefore, to study them, a set of methods is needed to deal with them efficiently. To analyze users' opinions, there are different tools and methods from a various set of disciplines:

• Machine learning approach

• Linguistic approach

• Semantic approach

Chaovalitand & Zhou, 2005 used semantic approach for idea mining. In this paper, reviews were tagged and then the two-word phrases which have a certain patterns based on parts-of speech are selected; these words can be placed in 5 possible states. Then these phrases are scored based on special formula. If the value of phrases is negative, it is known as a negative opinion and if the value of phrases is positive, it is known as a positive opinion. Moreover, one comparison was done between semantic approach and machine learning approach and it is shown that machine learning approach is more accurate than semantic approach but more time is needed to train the model [Chaovalitand & Zhou, 2005].

To understand better challenges and available solutions for mining online users 'opinions, various machine learning methods are analyzed by Hemmatian & Sohrabi in 2017 and benefits and drawbacks of each are presented as: 1- Supervised machine learning methods are slow; 2- they are very dependent on tagged training samples so they have relatively low efficiency (they are slow on training but fast on testing); 3- they need human participation and linguist knowledge; 4- they are costly. But supervised techniques have very high accuracy for classification; it has the ability to categorize numerous categories, shows good performance against noise in data [Hemmatian & Sohrabi in 2017]. They have effectiveness in the discovery of subject of the issue. In the semantic approaches, lexicon-based approaches have relatively low accuracy and are unable to find the opinion of the words which have the specific content and are not in the lexicon. On the other hand, they have some benefits for example, easy access to words lexicon and their orientation, and executing them is very fast [Hemmatian & Sohrabi, 2017]. In recent years, a new classification method, called forest rotation has been proposed [Rodrı´guez & Kuncheva, 2006]. Combining categories is currently an active area of research in machine learning and pattern recognition. Many theoretical and empirical studies have been published

that show the advantages of hybrid models over individual classification models (such as support vector machines) [Rodrıguez & Kuncheva].

Keshwani, Agarwal & Kumar, 2018 analyzed user's opinions with machine learning methods, especially Neural Networks to predict the trend of stock market for the gold, silver and crude oil. For example, the text "the Price of Gold will rise again" is given as an input to the proposed model. In the first step, after breaking the text into words and stemming them, the sentence is transformed to the set of words "Price", "of", "Gold", "will", "rise", "again" and they are identified as noun (NN, it represents noun, singular) verb (VB), Preposition or subordinating conjunction (IN), adverb (RB).

('Price', 'NN'), ('of', 'IN'), ('Gold', 'NN'), ('will', 'VB'), ('rise', 'VB'), ('again', 'RB')

In the second step, positive and negative scores are calculated. In the last step, the neural network is trained and predicts future prices [Keshwani, Agarwal & Kumar, 2018]. The problem of this method is that when there is a noise in data, it does not have good performance.

Bertola and Patti, 2015 applied sentiment analysis on the visitors' comments about online art gallery and they used tools and methods from a set of disciplines such as Semantic web, Social Web, and Natural Language Processing which have some rules. They created a social space that organizes art works based on users' feedbacks so that users were effective and involved in creating a semantic space to recognize opinions. The output of this work is an effective classification model for identifying comments. In short, a tagged database was given to the model and the relationship between tags and comments was explored using a combination of Semantic Web methods and Natural Language Processing and lexicon resources. The proposed

model was evaluated using a dataset of tagged multimedia artwork which was tagged and was for different artworks [Bertola & Patti, 2015].

The combination of classifiers is now an active area of research in machine learning and pattern recognition [Lasota, Luczak & Trawiński, 2012; Rodrıguez & Kuncheva, 2006]. Many theoretical and experimental studies having been published show the advantages of combination of them over individual classifiers (e.g. SVM). Many researchers have tried to design the systems which have multiple classifiers based on the same classifier model but they are trained on various data subsets or feature subsets. Such combined classifiers are named classifier ensembles. In recent years, a new classifier ensemble method which is called Rotation Forest has been proposed. It applies PCA in order to rotate the original feature axes to have various training set for learning classifier [Lasota, Luczak & Trawiński, 2012; Rodrıguez & Kuncheva, 2006]. The idea of the rotation approach is to have accuracy and variety in our categories at the same time. Diversity is achieved by extracting features from each of classifiers. Decision trees were chosen because they are sensitive to the rotation of the feature axes. Furthermore, the accuracy was improved while maintaining all the main components and using all the samples to teach each of the classifier [Rodrıguez & Kuncheva, 2006].

Two of the most popular effective methods in ensemble models are Bagging and Random Forest. Bagging was first introduced by Breiman in 1996. It has relatively a simple algorithm but has very good performance. Bagging is based on bootstrap selection. In fact, the training data with replacements from original dataset is given to each individual learner. Some instances can be represented multiple times [Lasota1, Luczak & Trawi´nski, 2012] so its disadvantage is that the classifier trained on training set having repeated instance may obtain a higher error in test

time than the classifier using all of the data. However, when classifiers are combined, they produce lower error on test data than the individual classifiers; the diversity among these classifiers can compensate for the increase in the amount of error in any individual classifier [Poria ,Cambria & Gelbukh, 2016]. Then individual learners are combined using an algebraic expression, for instance, minimum, maximum, sum, mean, product, median, etc. When each individual learner has learnt from the samples of bootstrap, individual classifiers in bagging ensemble shows relatively high accuracy in classification. Only thing that brings diversity into ensemble learning approaches is the amount of varied data in training dataset. The classifiers used in Bagging are sensitive to small changes in data but the bootstrap sampling leads to ensembles with low diversity compared to other ensemble creating methods. Hence, Bagging needs larger ensemble sizes to perform well [Lasota1, Luczak, and Trawi´nski, 2012].

To increase diversity, another method of ensemble learning Random Forest was suggested by Breiman [Rodriguez & Kuncheva, 2006; Breiman, 1996 & 2001]. Rodriguez and Kuncheva in their paper showed that Rotation Forest is similar to Bagging but Rotation Forest is slightly more accurate and slightly more diverse than Bagging [Rodriguez & Kuncheva, 2006; Breiman, 1996 & 2001].

Kouloumpiz et al., 2011 performed opinion mining on Twitter messages using hashtagged dataset (HASH) which is a subset of Edinburgh Twitter corpus. They use three different corpuses of Twitter messages in their experiment. To train the model they used Edinburgh Twitter corpus and EMOT corpus (emoticon dataset) and to test it they used ISIEVE dataset. In this article, tweets were divided into three categories: positive, negative and neutral. Initially, the data was pre-processed, consisting of three steps: 1- Tokenization 2- normalization

3- Part-of speech tagging (POS). In the normalization stage, the abbreviations were found and replaced with the correct meaning. For instance, “BRB“is replaced with”be right back”. 4- All capital words are made into lower case. The advantage of normalization is that it improves the labeling performance of POS. A variety of features were used, for example, unigram, bigram, lexicon and POS features. Tweets were categorized into three categories: positive, negative, and neutral. Furthermore, they used AdaBoost ensemble learning. The benefits of their model were that useful features were extracted and ensemble learning was used, but extracting all useful features required more time.

Abbasi et al., 2008 implemented opinion mining for Internet forums and blogs. The language used in forums and blogs can be in different languages or can be slang and contrary to grammar. These messages became logical messages with two types of features extracted from them: 1- Features based on Lexicon resources, the words in database were labelled positive, negative and neutral, 2-Features based on speech components. The number of speech components, nouns, verbs, pronouns, adverbs, etc. was counted in each message. Then the features were reduced and selected automatically. Support Vector Machine (SVM) was used for classification. Selecting features and reducing the dimensions lead to increase in the performance of the classification. But the disadvantage of the proposed method in this paper was that it was not able to identify some words in different languages [Abbasi, Chen & Salem, 2008].

Severyn and Moschitti in 2015 did opinion mining for YouTube in three stages: 1- the classifiers that predict the type and polarity of the comment were modeled, while distinguishing whether the polarity was related to the products or the videos; 2- Proposed a structural model that adapts well when tested with domains; and 3- evaluating the effectiveness of the proposed model

on two languages, English and Italian. Tree kernels were used to better extract features. The proposed structure was compared with bag of words (BOW) models with the same domain. The results showed that when available data was more than 4k or even when available data was limited, the proposed structural model performed well [Severyn & Moschitti, 2015].

Poria, Cambria and Gelbukh, 2016 used a 7-layer deep Convolutional Neural Network to examine all aspects of a text or idea and they examined the emotions. Moreover, a set of linguistic patterns were combined with the neural network [Severyn, & Moschitti, 2015]. The results showed better accuracy in classification than the state-of-the-art methods, but the combination of speech patterns with the deep neural networks to improve the diagnostic pattern increased the complexity and time.

Huang et al., 2008, used apriority algorithm for the classification of opinions where each opinion was considered as a transaction. In this study, the numbers 0, 1 and 2 indicated positive, negative and neutral comments, respectively. For positive and negative comments, the classification accuracy of this algorithm was high, however, for neutral comments it failed to increase the classification accuracy. Nakov, Rosenthal, Kiritchenko, Mohammad, Kozareva, Ritter, Stoyanov and Zhu in 2016 examined opinions and their polarities. They presented Semantic analysis and Natural Language Processing of the text in social media to do semantic evaluation. They performed their task on 44 participating teams in 2013 and on participating 46 teams in 2014. They continued to perform their task in 2015 on a large corpus consisting of tweets, SMS messages, LiveJournal messages, and a special set of sarcastic tweets. In this paper the process of collecting data was shown and the words which expressed sentiment were gathered using SentiWordNet as a repository of sentiment words. One message was classified

based on weather a marked word was positive or negative or neutral. The message that contained words with both positive and negative comments, the one which was stronger was selected [Nakov, Rosenthal, Kiritchenko, Mohammad, Kozareva, Ritter, Stoyanov & Zhu, 2016]. In both years (2014 and 2013), most systems were supervised, and different features were extracted using n-grams, stemming, punctuation, part-of-speech (POS) tags and Twitter-specific encodings such as emoticons, hashtags, and abbreviations. Support Vector Machines (SVM), Maximum Entropy and Naı¨ve Bayes were used to classify [Nakov, Rosenthal, Kiritchenko, Mohammad, Kozareva, Ritter, Stoyanov & Zhu, 2016]. The benefits of Natural Language Processing are that it helps to find the meaning of a word better and it simplifies the extraction of features. Rill, Reinel, Scheidt, Zicari and PoliTwi in 2014 tried to detect top political topics on Twitter. Sentiment analysis was done to find the polarity of topics marked by sentiment hashtags. Chen and Liu, 2014 focused on topic modeling to mine the topics of the documents. Knowledge-based topic models including Latent Dirichlet Allocation (LDA) and Probabilistic Latent Semantic Analysis (PLSA) and their extensions have been widely used to extract the topic of document but the disadvantages of these approaches is that they need high amount of data of one special domain in order to generate coherent topic to produce reliable results. 100 reviews were tested on the classic topic model LDA and it produced very poor results. Chen and Liu, 2014 proposed a new topic model called AMC (topic modeling with automatically generated Must-links) and compared AMC model with five state-of-the-art models and showed that AMC could produce more accurate topics [Chen and Liu, 2014].

Evermann, Rehse and Fettke in 2017 proposed a new approach to predict the future behavior in a business process using Natural Language Processing and Deep Learning with recurrent neural networks (RNN). Their approach can contribute to process management. This

method used unique features to form the feature vector for the input of network. Although this model had acceptable functionality, but there was a problem with the loss of some data and information.

2.3 Overview

The online users' comments are usually in unstructured form. Therefore, to study them, a set of methods is needed to deal with them efficiently. To analyze users' opinions, there are different tools and methods from a different set of disciplines including Machine learning approaches, Linguistic approaches, Semantic approaches. The Machine Learning approach includes Artificial Neural Network, Support Vector Machine and Ensemble learning methods. The learning methods were used in this area is mostly supervised classifiers because they are more accurate. But the semantic approach for exploring ideas is unsupervised learning because it does not require prior training. The language approach involves natural language processing which includes tokenization, stemming, stop-word removal, punctuation removal and POS tagging. Some of the methods, having been used by some researchers to categorize the opinions of online users were mentioned above. Also, some benefits and drawbacks were mentioned to some extent. In this research, an attempt is made to suggest a new approach to solve the problems in existing methods. Many researchers have tried to use different individual supervised machine learning approaches to categorize the opinions of online users and some of them presented that the classification accuracy of machine learning approach is more than the semantic approaches. Moreover, the performance of ensemble learning methods was higher than individual machine learning approaches. In this research Random Forest was used as an ensemble method because of the mentioned benefits as well as other classifiers such as Gradient Boosting, Neutral Network, and Support Vector Machine (SVM) were applied to categorize online users' opinions. Natural

Language Processing techniques were used to preprocess and to extract useful features from semi-structured (XML) Textual data. Figure 2.1 shows the approaches for opinion mining.

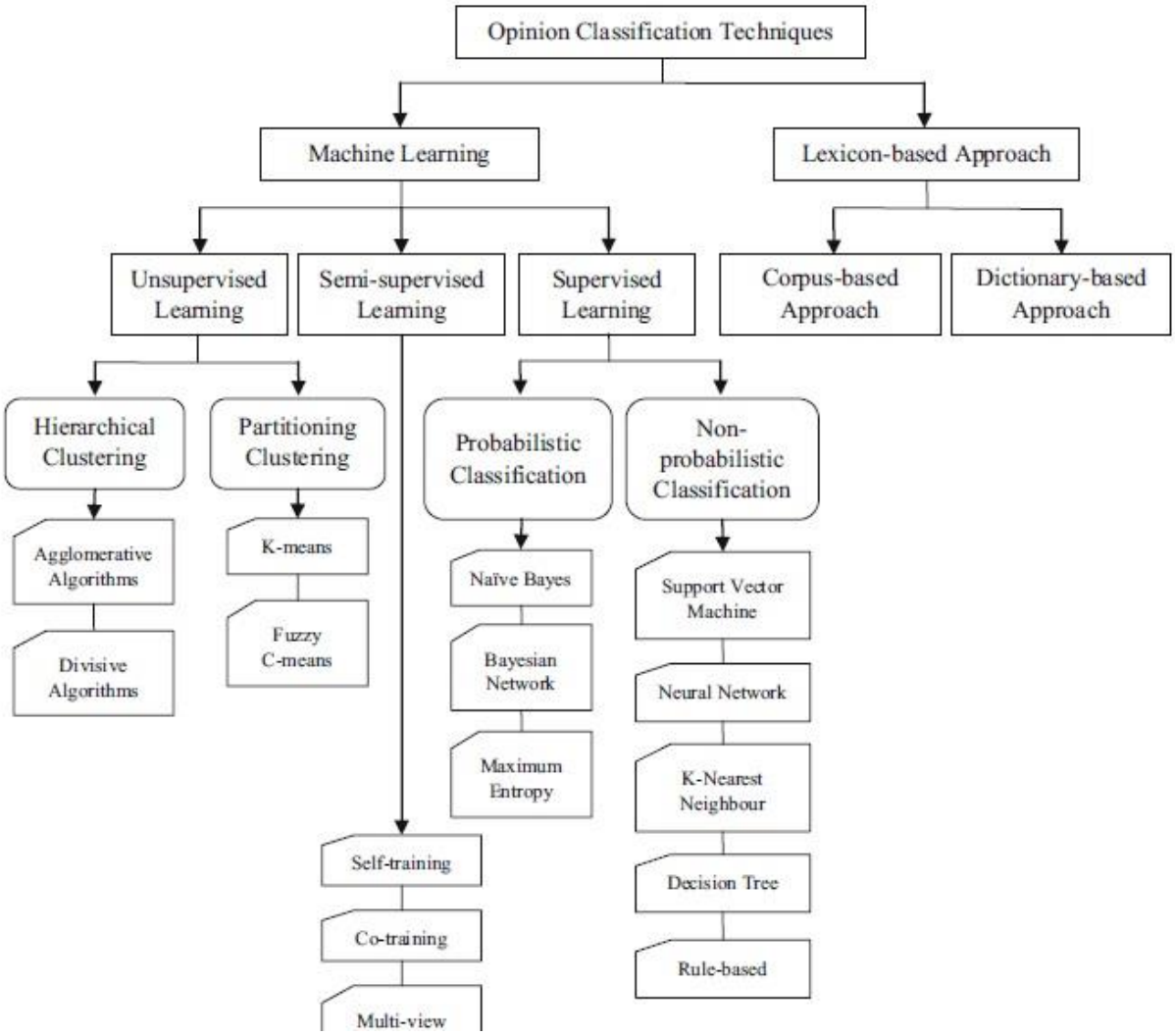


Figure 2.1: Overall view of several classification techniques in opinion mining [Hemmatian & Sohrabi, 2017]

Chapter 3

Materials and Methods

3.1 Data

The data set contains 689 comments, which are extracted from the users' comments from Amazon.com annotated by Minqing Hu and Bing Liu, 2004. The selected comments are about “Speakers”. In fact, each comment is written by one user and it has a certain label that shows the author's desire to comment. This label can be "positive", "negative" or "neutral". The data is provided in the form of XML file. An example of the comments in the XML files is shown in Figure 3.1. In these files, each comment is presented in XML tags. Each sentence is placed in a sentence tag. The properties of the author of the comment (same as the comment tag) are given in the aspectTerm tag. Along with that, there is an aspect of the product which has been commented, as well as the positions of its beginning and end in the sentence are given in the aspectTerm tag. More precisely, most of the user sentences have one aspectTerm tag but there are some sentences which have more than one aspectTerm tag which may have different terms and values for their polarity.

XML files were processed with the help of Python libraries. Users’ feedbacks are extracted and are sent as an initial input to the proposed method. Figure 3.1 shows part of corpus in XML format. The data belongs to three different classes, positive, negative and neutral. Table 3.1 shows a sample of sentence categories with their class labels.

```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<!DOCTYPE sentences SYSTEM "xhtml_lat1.ent">
- <sentences>
  - <sentence id="1">
      <text>These speakers are incredibly amazing .</text>
    - <aspectTerms>
        <aspectTerm pos="nn" term="speakers" polarity="positive" to="13" from="6"/>
      </aspectTerms>
    </sentence>
  - <sentence id="2">
      <text>Before this I had the jbl on stage , which were also incredibly good .</text>
    </sentence>
  - <sentence id="3">
      <text>But I wanted something that I can play without having to plug into a power
        outlet and would work on rechargable batteries .</text>
    </sentence>
  - <sentence id="4">
      <text>I didnt want one in which I would have to put in batteries myself -LRB- that is a
        pain -RRB- I did alot of research and read a bunch of reviews and im glad I
        purchased these speakers .</text>
    - <aspectTerms>
        <aspectTerm pos="nn" term="speakers" polarity="positive" to="181" from="174"/>
      </aspectTerms>
    </sentence>
  - <sentence id="5">
      <text>Most of the reviews about the sound quality is true .</text>
    </sentence>
  - <sentence id="6">
      <text>The speakers have a very rich sound and good bass also , obviously not
        thumping bass for which you would need a huge subwoofer !!</text>
    - <aspectTerms>
        <aspectTerm pos="nn" term="speakers" polarity="positive" to="11" from="4"/>
        <aspectTerm pos="nn" term="sound" polarity="positive" to="34" from="30"/>
        <aspectTerm pos="nn" term="bass" polarity="positive" to="49" from="45"/>
      </aspectTerms>
    </sentence>
  - <sentence id="7">
      <text>Although the bass im getting out of these speakers was unexpected .</text>
    </sentence>
```

Figure 3.1 Part of the raw data in XML form

Table 3.1 a sample of each category in the data set with their classes

Sentence	Class label
These speakers are incredibly amazing	Positive
The only downside is the remote which requires a line of sight	Negative
The main thing I was looking for was sound quality	Neutral

3.1.1 Preprocessing

As it is mentioned, user comments are textual and non-structured data that is not normally usable in opinion mining. In other words, machine learning methods will not be able to use textual data not having any basic structure and concept. In any data mining task, machine learning algorithms are trained with a set of training examples, each of which is described in a vector of characteristics. In fact, with help of the features in existing examples, a learning algorithm will be trained. Therefore, an initial phase is needed to prepare textual data.

Data preparation refers to the processing at the end of which the necessary features of the user opinions are extracted, and are evaluated. To do this, some pre-processing steps are required as shown in the Figure 3.2.

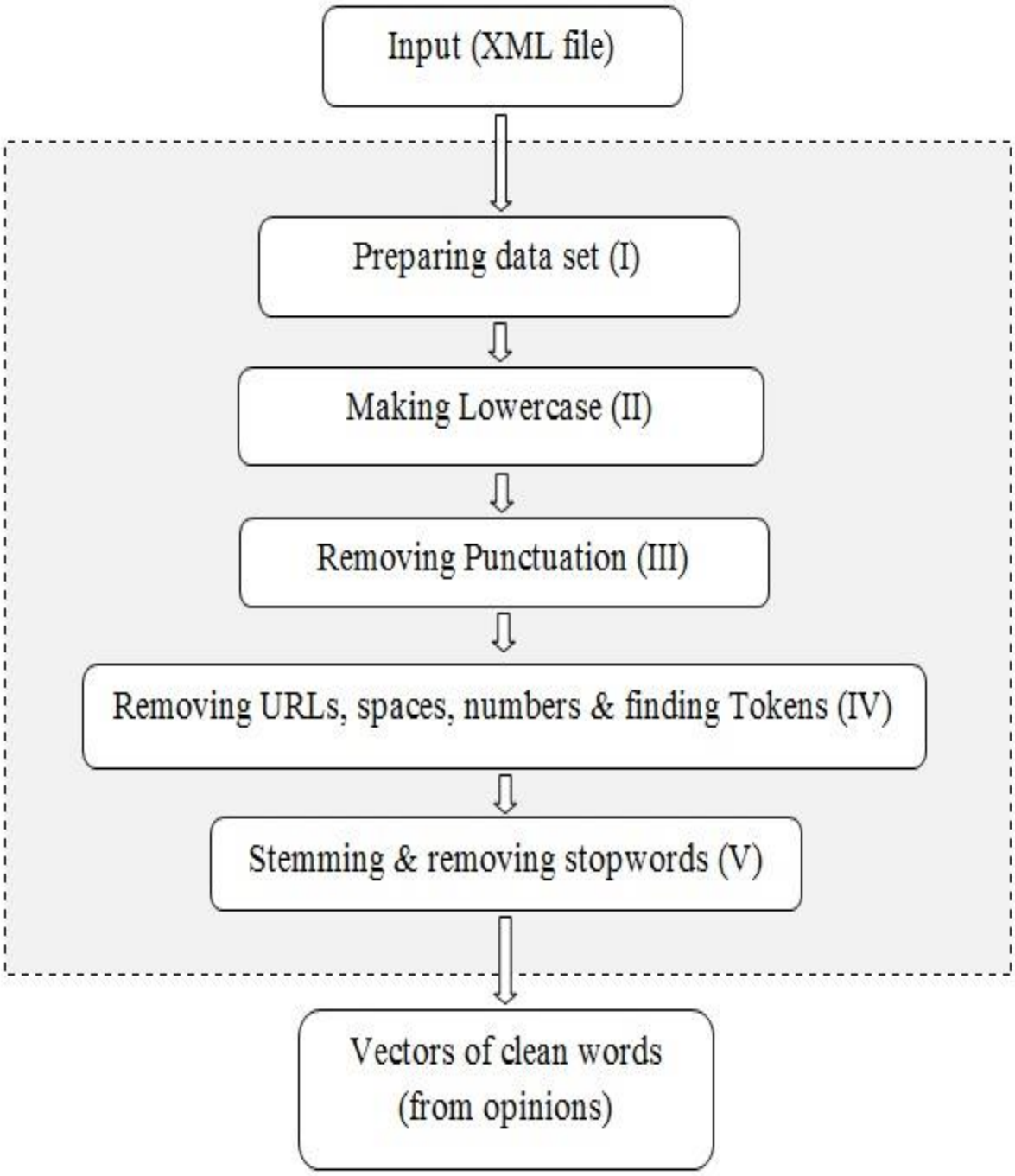


Figure 3.2: preprocessing steps for opinion mining, used in this thesis

Step 1: Preparing data

Text data and labels are extracted from XML file using python library

Output of reading XML data in python

The text is extracted and shown in Figure 3.3.

```
step 1: Preparing Dataset related to Speaker-------
689
['These speakers are incredibly amazing .', 'Before this I had the jbl on stage , which were also incredibly good .', 'But I wanted somethi
ng that I can play without having to plug into a power outlet and would work on rechargable batteries .', 'I didnt want one in which I woul
d have to put in batteries myself -LRB- that is a pain -RRB- I did alot of research and read a bunch of reviews and im glad I purchased the
se speakers .', 'Most of the reviews about the sound quality is true .', 'The speakers have a very rich sound and good bass also , obviousl
y not thumping bass for which you would need a huge subwoofer !!', 'Although the bass im getting out of these speakers was unexpected .', '
Without a doubt I would say that these speakers have the same sound quality as the jbl on stage , and are truly portable -LRB- can be playe
d without a power outlet , I love that feature -RRB- .', 'The only downside is the remote which requires a line of sight .', 'Its not a big
 deal for me .', 'The main thing I was looking for was sound quality !', 'I read all the reviews herein , and saw all the concerns .', 'Fir
st , i was looking for an iPhone dock and FM tuner , not necessarily an alarm clock - as its for the office .', 'I must say : - IT WORKS WI
TH IPHONE4 , EVEN WITH THIN CASE ON IT .', 'just push it into the slot - It charges the iPhone4 .', '- It allows incoming calls to interrup
t the music , and you can answer the call without having to undock it .', '- It allows incoming email and TXT messages to be read and respo
nded to w\\/o undocking it - essentially you can do ALL iPhone apps while its docked .', '- Sound quality is good .', '- Tuner quality is g
ood .', '- Looks fantastic , esp w\\/iPhone4 docked !!', '- Setup , from in the box to completely working , and programmed for 20 preset FM
 stations , was under 5 minutes .', '- I bought from an Amazon Prime Fullfiller , used like new , and it is just that - like new , not a bl
emish or issue , and for $ 56 ... I have no qualms about this .', 'Yes , the buttons are a bit small and close , but they are functional .'
, 'My only recommendation - include a remote ... BUY THIS !', 'I purchased this phone set for work , thinking that inexpensive replacement
phones would ease the pain of things being spilled on them or dropped .', 'The sound quality has been poor since day one , the base constan
tly drops calls , and now there seems to be some sort of short circuit in the base -- we can no longer use the AC adapter with the base or
it turns off completely .', 'When I purchased this set , I never imagined I would be running my business telephones off of 4 AA batteries .
', "Other complaints : you ca n't put line 1 on hold and answer line 2 with the same handset without a `` conference call '' beginning .",
'The incoming call volume is incredibly low and it is often difficult to hear my customers -LRB- it was so bad I had my hearing tested beca
use I thought something was wrong with me .', "Yes , I 'm blonde , but this proves to me it 's the phones ! -RRB-", "I got this for use on
my truck and I 'm happy with it for the most part , it is loud and super easy to install , however you are going to want an amp for i", 'I
will never buy another GE phone again .', 'Period .', 'These speakers have great sound but they will buzz if you leave them on with the sou
nd turned up .', 'These serve as my primary speakers in my living room -LRB- about 14X22 -RRB- .', "They sit on stands about 8 '' high , an
d are pretty close to being in the corners on one of the long walls .", "I am sure that if I had spent more and gone with bigger , floor-st
anding units from Klipsch , I would be even more impressed , but frankly , these get the job done and do n't seem to break a sweat .", "Mos
t of our listening is at modest volumes , but I have turned them up on some demanding tracks and they do n't flinch .", 'Even the bass is p
```

Figure 3.3: Text from the XML file

Extracting proper polarity of each opinion

Most of the user sentences have one individual polarity tag, namely positive, negative or neutral; in such samples, label "1" was applied when the polarity of a sentence is positive and "-1" for negative sentences and 0 for neutral ones. Some sentences have more than one polarity. In fact a user may give different views or the same views about different parts of the product, in this case,

the polarities are added for positive or negative and if sum of number of positive polarities is more than the negative ones, label “1” is selected, otherwise label “-1” is selected for that sentence. If the number of positive and negative polarities is equal, a label “1” 0r “-1” is randomly selected**.** Figure 3.4 depicts one sample that has three positive polarities, and Figure 3.5 depicts one sample which has two different polarities, a positive and a negative.

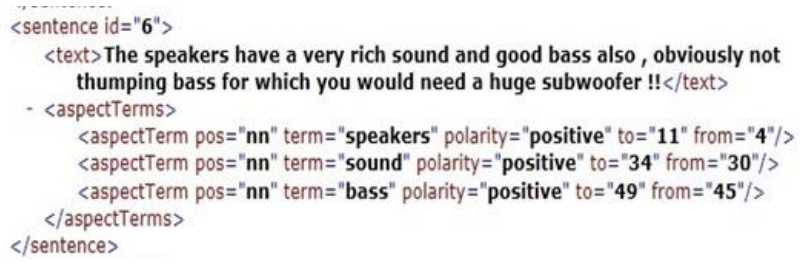


Figure 3.4: One sample that has three positive polarities

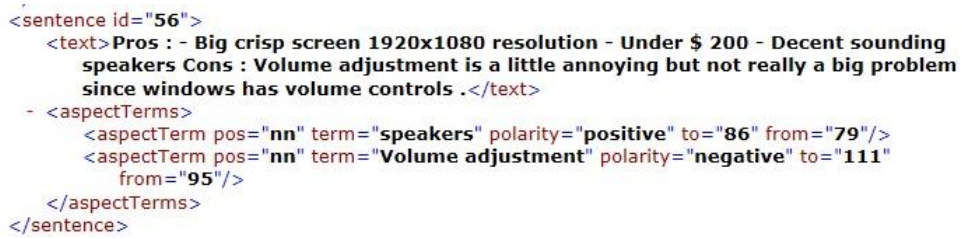


Figure 3.5: One sample which has two different polarities (positive and negative)

After saving the polarity of all sentences, each sentence has one individual label (class), -1, 0 or 1. Figure 3.6 shows the selected labels for the data set.

Figure 3.6: Labels of the data set

Step 2: Making Lowercase

In the second step of pre-processing, all the words are converted into lowercase letters. Figure 3.7 shows an example after step 2 is applied.

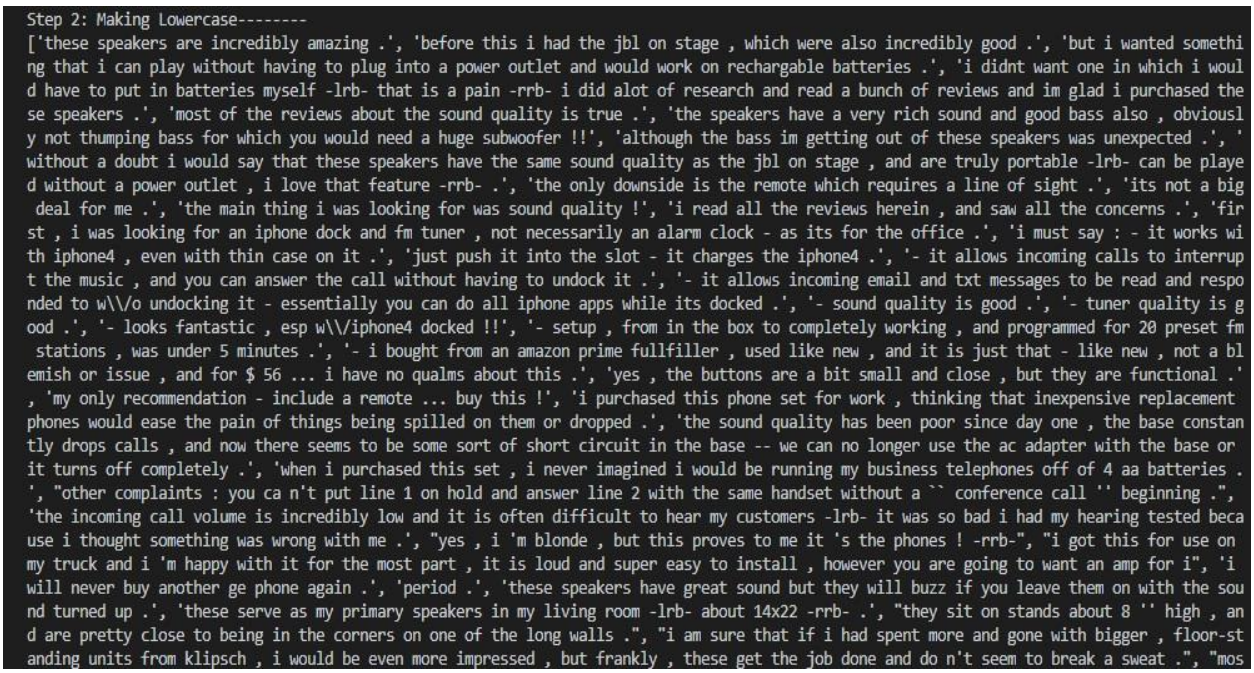


Figure 3.7: Output after making lowercase

Step 3: Removing Punctuation

In this step, punctuation is deleted from the opinions. Punctuation marks are shown below and Figure 3.8 shows the output after removing punctuation

```
~{|}`_^[\]@?<=>;:/.-,+*()'&%$#"!
```

```
step 3: Removing Punctuation------
i bought this item along with a couple of other external speakers in an effort to find a good set to use in my elderly car lrb speaker syst
em died rrb
step 3: Removing Punctuation------
this set is not my preferred to use in my car lrb i use the jbl on tour  great sound for a small  compact speaker rrb  but i was very impre
ssed with the phillip speaker sound and use them very regularly around the house
step 3: Removing Punctuation------
for the price  the sound is great and their portability lrb pack down to a large eggshape rrb is quite good
step 3: Removing Punctuation------
you ca nt go wrong with these if what you are looking for is a good sound in a small package
step 3: Removing Punctuation------
for what a pain in the ass setting clocks are is im walking on freaking sunshine
step 3: Removing Punctuation------
the soft orange glow at the base is a nice nightlight
step 3: Removing Punctuation------
and the sound well there is no complaints about that
```

Figure 3.8: Output after removing punctuation

Step 4: Removing URLs, spaces, numbers & finding Tokens

If there are URLs in users' comments, they will be deleted. Also, additional characters such as blank spaces and numbers are removed from the comments. We know that each sentence represents an opinion and has a label that indicates its tendency. In the following, each sentence is broken down into its Compound words. In fact, we can think of text as a sequence of tokens. This process is called tokenizing. The output of this step is the tokenization of the comments, each of which is converted into a vector of words. The output of (IV) and Figure 3.9 depict some of the output after performing tokenization step.

Output (IV)-*['i', 'm', 'not', 'sure', 'that', 'any', 'speakers', 'could', 'offer', 'me', 'high', 'fidelity', 'at', 'very', 'low', 'volumes', 'though']*

```
['so', 'besides', 'the', 'annoyances', 'this', 'is', 'a', 'fine', 'speaker', 'for', 'the', 'amount', 'i', 'paid']
Step 4: Finding Tokens------------
['for', 'the', 'cost', 'of', 'this', 'product', 'i', 'could', 'not', 'ask', 'for', 'more']
Step 4: Finding Tokens------------
['the', 'sound', 'quality', 'is', 'very', 'exceptable']
Step 4: Finding Tokens------------
['the', 'only', 'issue', 'i', 'have', 'against', 'the', 'speakers', 'is', 'its', 'a', 'little', 'on', 'the', 'low', 'tone', 'side', 'but',
'still', 'is', 'very', 'good', 'and', 'you', 'can', 'set', 'your', 'equalizer', 'to', 'make', 'up', 'for', 'it']
Step 4: Finding Tokens------------
['all', 'in', 'all', 'could', 'nt', 'be', 'more', 'happy', 'with', 'the', 'purchase']
Step 4: Finding Tokens------------
['easy', 'set', 'up', 'and', 'the', 'picture', 'was', 'crystal', 'clear']
Step 4: Finding Tokens------------
['i', 'use', 'it', 'for', 'the', 'xbox', 'as', 'well', 'as', 'my', 'computer', 'and', 'i', 'cried', 'with', 'joy']
Step 4: Finding Tokens------------
['ok', 'enough', 'said', 'just', 'buy', 'it']
Step 4: Finding Tokens------------
['love', 'this', 'speaker']
Step 4: Finding Tokens------------
['sounds', 'awesome']
Step 4: Finding Tokens------------
['works', 'super', 'easy']
Step 4: Finding Tokens------------
['looks', 'great']
```

Figure 3.9: Output after applying tokenization step

Step 5: Stemming & removing stopwords

In this step, Repetitive words such as articles, prepositions or auxiliary verbs and Stop words that are frequently repeated in daily sentences are omitted. Indeed, insignificant words are deleted, additionally, stemming is performed.

In terms of stopwords, as it is mentioned before there is no universal stopwords list. In this research, the Standard English language stopwords list from NLTK was used. In Figure 3.10 lines of code in python in order to see stopwords in nltk is shown. The contents of *nltk.corpus.stopwords.words ('english')* are shown in Figure 3.11 to get an idea of the stop words. Also, the output of Stemming and removing stopwords is depicted in Figure 3.12.

```
from nltk.corpus import stopwords
stop_words = set(stopwords.words ('english'))
print(stop_words)
```

Figure 3.10: Lines of code in python to see stopwords in nltk

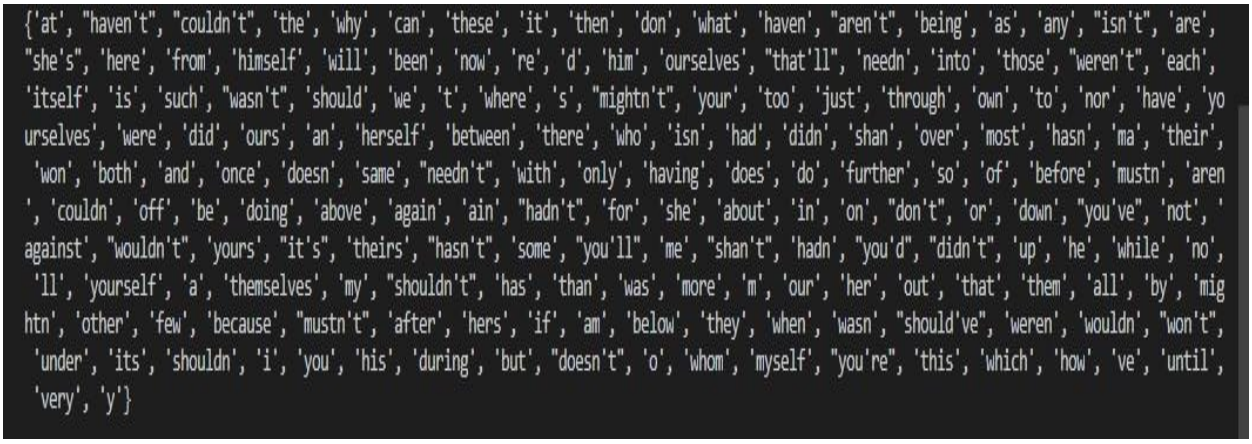


Figure 3.11: Standard English language stopwords list from NLTK

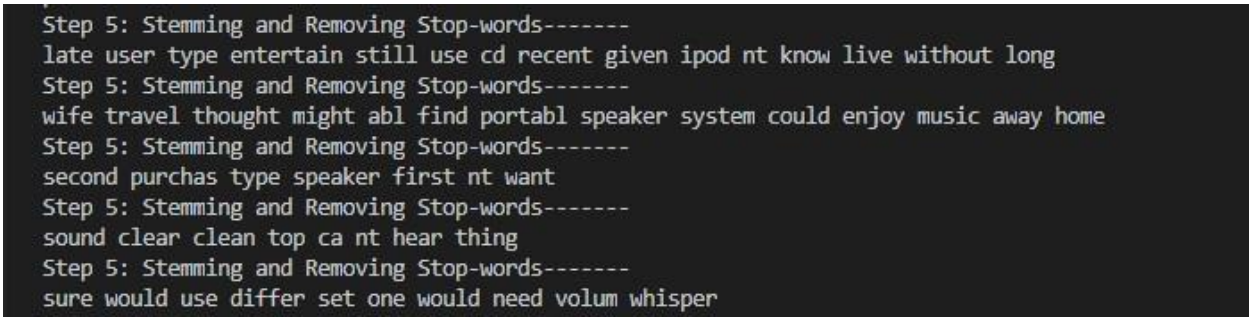


Figure 3.12: Stemming and removing stopwords

3.1.2 Feature extraction

Once all the important words from all the comments in the data set have been extracted, in the form of a vector of significant words from each opinion, statistical analysis is used to extract meaningful features for the learning methods. The implementation of this approach is described

below. Figure 3.13 shows all possible 1-grams, 2-grams and 3-grams are extracted from the comments in the data set.

Its sound is great

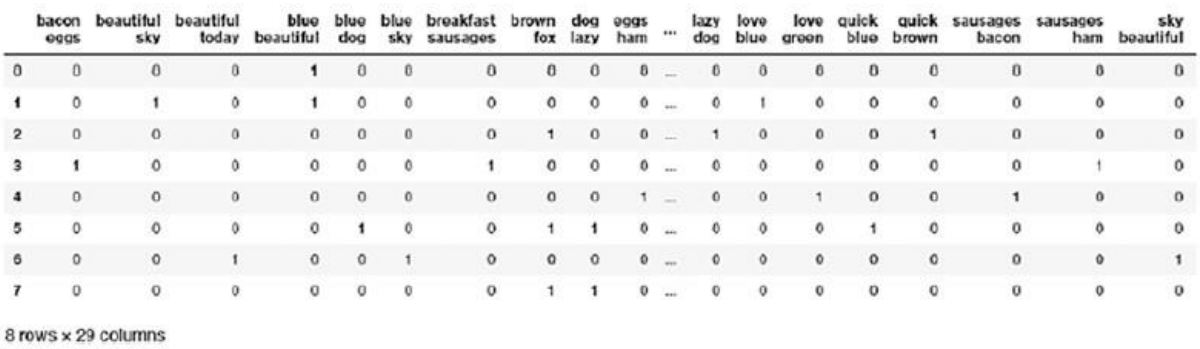
Unigram	Its	Sound	is	great
Bigram	Its sound	Sound is	Is great	
Trigram	Its sound is	Sound is great		

Figure 3.13: unigrams, Bigrams and trigrams is extracted

Using a statistical approach all possible 1-grams, 2-grams and 3-grams are extracted from the comments in the data set. Moreover, to quantify the features a statistical measure such as TF-IDF is used. The following examples in Figure 3.14 and Figure 3.15 show unigrams (1-grams), bi-grams respectively, which are extracted from features in a sample text document. Then value for each term is quantified by TF-IDF; as it is seen, the result is a sparse matrix [Sarkar, 2019].

	bacon	beans	beautiful	blue	breakfast	brown	dog	eggs	fox	green	ham	jumps	kings	lazy	love	quick	sausages	sky	toast	today
0	0.0	0.0	1.81	1.59	0.0	0.00	0.00	0.0	0.00	0.0	0.0	0.0	0.0	0.00	0.0	0.00	0.0	1.81	0.0	0.0
1	0.0	0.0	1.81	1.59	0.0	0.00	0.00	0.0	0.00	0.0	0.0	0.0	0.0	0.00	2.1	0.00	0.0	1.81	0.0	0.0
2	0.0	0.0	0.00	0.00	0.0	1.81	1.81	0.0	1.81	0.0	0.0	2.5	0.0	1.81	0.0	1.81	0.0	0.00	0.0	0.0
3	2.1	2.5	0.00	0.00	2.5	0.00	0.00	2.1	0.00	0.0	2.1	0.0	2.5	0.00	0.0	0.00	2.1	0.00	2.5	0.0
4	2.1	0.0	0.00	0.00	0.0	0.00	0.00	2.1	0.00	2.5	2.1	0.0	0.0	0.00	2.1	0.00	2.1	0.00	0.0	0.0
5	0.0	0.0	0.00	1.59	0.0	1.81	1.81	0.0	1.81	0.0	0.0	0.0	0.0	1.81	0.0	1.81	0.0	0.00	0.0	0.0
6	0.0	0.0	1.81	1.59	0.0	0.00	0.00	0.0	0.00	0.0	0.0	0.0	0.0	0.00	0.0	0.00	0.0	3.62	0.0	2.5
7	0.0	0.0	0.00	0.00	0.0	1.81	1.81	0.0	1.81	0.0	0.0	0.0	0.0	1.81	0.0	1.81	0.0	0.00	0.0	0.0

Figure 3.14: Constructing the raw TF-IDF matrix [Sarkar, 2019]



	bacon eggs	beautiful sky	beautiful today	blue beautiful	blue dog	blue sky	breakfast sausages	brown fox	dog lazy	eggs ham	...	lazy dog	love blue	love green	quick blue	quick brown	sausages bacon	sausages ham	sky beautiful
0	0	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0
1	0	1	0	1	0	0	0	0	0	0	...	0	1	0	0	0	0	0	0
2	0	0	0	0	0	0	0	1	0	0	...	1	0	0	0	1	0	0	0
3	1	0	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0	0	1	...	0	0	1	0	0	1	0	0
5	0	0	0	0	1	0	0	1	1	0	...	0	0	0	1	0	0	0	0
6	0	0	1	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0	1	1	0	...	0	0	0	0	0	0	0	0

8 rows × 29 columns

Figure 3.15: Bigram based feature vectors using the Bag of N-Grams model [Sarkar, 2019]

3.2 Methodology

Assumptions of the Proposed Method

In this study, it is assumed that a collection of training and test data is available as it is mentioned in Section 3.1.

Steps of the Proposed Method

The objective of this study is to identify the author's desire or inclination of comment on a product or service. These tendencies can be positive, negative, or neutral. The flowchart in Figure 3.16 shows an overview of the proposed method. It is divided into various steps to classify opinions.

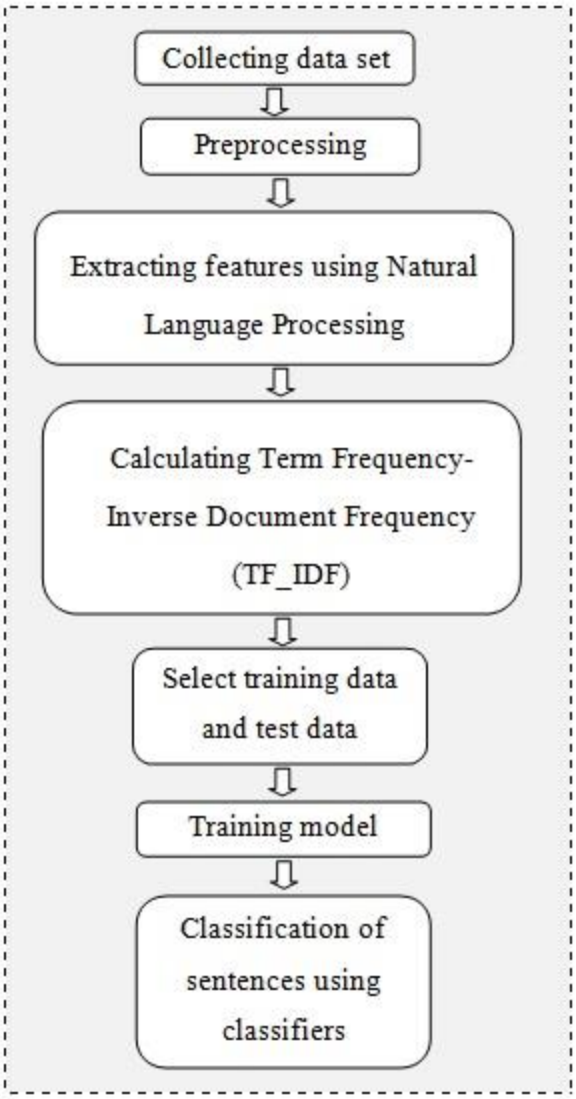


Figure 3.16: Proposed method

In the first step of Figure 3.16, existing data set (people's opinions) which is in text form and conversational sentences are processed. In fact, users' comments are raw, textual, and unstructured which is not usable without initial processing. Some steps need to be taken to make new data sets which can be used to train the model. These steps will be described in the following section Such as pre-processing step which is necessary because user feedback is raw and textual and cannot be used to train the learning models. In the second step, appropriate

features using Natural Language techniques are extracted; the output of this step will be user comments in a new feature space, then those features are vectorized by Term Frequency- Inverse Document Frequency (TF_IDF) approach, so that they can be used to train the learning models. The detailed steps for preprocessing and extracting features are described in Preprocessing section and Feature extraction section. After building a new data set, the main step is implementing and training the classifiers. This step is also done in a number of sub-steps. Finally, the trained model is used to classify new opinions.

Classification of user comments using some classifiers

After completing the pre-processing phase, the next steps include analyzing the opinions and categorizing them. In this study, some classifiers like Gradient Boosting, Neural Network, SVMs and Random Forest which is based on ensemble learning technique will be used to analyze the opinions.

In fact, Random forest algorithm is an ensemble learning algorithm which uses a large number of decision trees to learn. A new sample is given to that algorithm to classify where each tree gives its “vote” for a class, then the class with the most votes is selected as the class of new sample. The purpose of creating this model is to analyze users' opinions to identify the author's attitude and to determine if it is positive, negative or neutral.

Random forest is used as it can manage over fitting of the data. When a sample of opinion (which is converted to a vector of features) is given to each tree, it moves from its root to its leaf. Finally, the new sample’s class is the class which a large number of trees have predicted. After applying Random forest, some other classifiers namely, SVMs, Neural Network, Gradient Boosting are applied on the data set and results will be evaluated in chapter 4.

3.3 Natural Language Processing (NLP)

We are in the age of Big Data where organizations and businesses are facing difficulty in managing all the data generated by different systems, processes, and transactions. For instance, retail and online stores generate a large amount of textual data from new product information and customer reviews and feedback, as well as large amounts of data in the form of tweets, messages, hash tags, articles, blogs, wikis, and much more on social media is seen [Sarkar, 2019].

Data in text and speech is unstructured which can contain important information; such textual data presents in all organizations, irrespective of their domain. However, because of the inherent complexity of processing and analyzing such an unstructured and noisy data, people usually refrain from spending extra time to analyze these unstructured data. This issue lead data scientists to introduce Natural language processing (NLP) to extract useful information from text data. Generally, Natural language Processing (NLP) is about the techniques and algorithms which help people or robots to process and understand natural language-based data. Because of increasing demand in today's fast-paced world, it is expected of data scientists to be proficient in natural language processing [Sarkar, 2019].

Natural language Processing (NLP) is defined as a specialized field of computer science and engineering and artificial intelligence with roots in computational linguistics. It is primarily concerned with designing and building applications and systems that enable interaction between machines and natural languages created by humans. This also makes NLP related to the area of human-computer interaction (HCI). NLP techniques enable computers to process and understand human natural language and utilize it further to provide useful output [Sarkar, 2019].

NLP preprocessing pipeline

The text is passed on to the NLP preprocessing pipeline, which consists of the following steps:

1. Removal stop words

2. Tokenization

3. Stemming

4. Lemmatization [Kulkarni & Shivananda, 2019]

3.3.1 Removal stop words

What are Stopwords

Stopwords are words that have little importance or they have no significance and often are deleted from text document, when the text is being processed so the words having maximum significance and context are kept. Stopwords usually occur most frequently if you aggregate a corpus of text based on singular tokens and check their frequencies. Words like “a”, “the”, “and” and so on are stopwords. There is no universal or exhaustive list of stopwords and often each domain or language has its own set of stopwords [Sarkar, 2019].

In this thesis, the Stop words that are frequently repeated in daily sentences are omitted. Indeed, insignificant words are deleted.

3.3.2 Tokenization

In Word tokenization process, each sentence is split into its words which constitute the sentence. Using tokenization, a sentence is essentially split into a list of words which can be used to reconstruct the sentence. Word tokenization is fundamental in many processes involving text

data, especially in cleaning and normalizing text data where operations like stemming is performed on each individual word based on its respective stems.

Main interfaces in NLTK for word tokenization

NLTK provides varied useful interfaces for word tokenization, similar to sentence tokenization [Kulkarni & Shivananda, 2019]. The main interfaces are given below:

- word_tokenize
- TreebankWordTokenizer
- TokTokTokenizer
- RegexpTokenizer
- Inherited tokenizers from RegexpTokenizer [Kulkarni & Shivananda, 2019].

3.3.3 Word Stemming

To understand the process of stemming, understanding what word stemming represents is necessary. Firstly for clarification, the term morphemes need an explanation, which are the smallest independent units in any natural language. Morphemes consist of units which are stems and affixes. Affixes are units like prefixes, suffixes, and so on, which are attached to word stems in order to change their meaning or create a new word altogether. Word stems are also often known as the base form of a word and we can create new words by attaching affixes to them. This process is known as inflection. The reverse of this is obtaining the base word from its inflected form and this is known as stemming. For instance, consider the word “JUMP”, affixes can be added to it and form several new words like “JUMPS”, “JUMPED”, and “JUMPING”. In

this example, “JUMP” is the base word and this is the word stem. If stemming is applied on any of its three inflected forms, you would get the base form. This is shown in Figure 3.17.

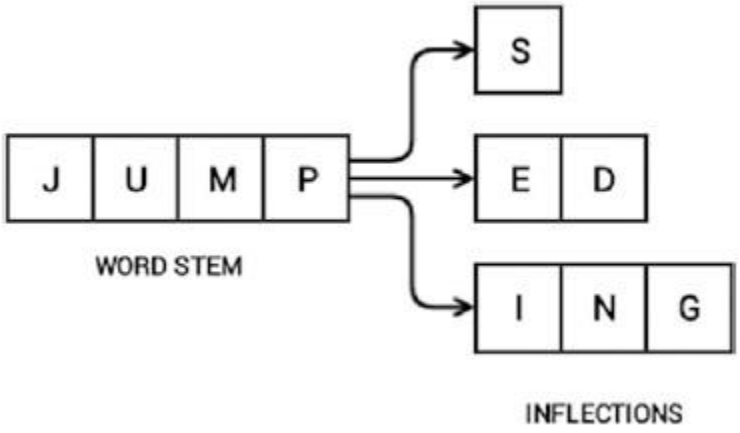


Figure 3.17: Word stem and inflections [Sarkar, 2019]

Figure 3.17 illustrates the stem of word in all its inflections. Stemming helps application to find the base stem of the words regardless of their inflections, this technique helps many applications, namely classifying or clustering text or even in information retrieval. This technique largely is used by Search engines in order to achieve better accurate results regardless of the word form. The NLTK package has some implementations for stemmers. The Porter stemmer is one of the most common. Porter stemmer is based on the algorithm created by Martin Porter [Sarkar, 2019].

3.3.4 Lemmatization

The process of extracting a root word by considering the vocabulary is called lemmatization. For instance, “good”, “better,” or “best” is lemmatized into good. Lemmatization determines Part of speech (PoS) of word. It returns the dictionary form of a

word, which should be a valid word while stemming just extracts the root word. Stemming was used in this research instead of lemmatization.

• Lemmatization handles matching “car” to “cars” along with matching “car” to “automobile.”

• Stemming handles matching “car” to “cars” [Sarkar, 2019].

3.3.5 Feature Engineering

Bag of N-Grams Model

One word is just an individual token, which usually is known as a *unigram* or *1-gram*. The Bag of Words model doesn’t keep the order of words. But if extracting phrases or collection of words which occur in a sequence is needed, N-grams model can help to deal with that issue. More precisely, one N-gram is a collection of tokens (words) in a text document where the tokens are contiguous and occur in a sequence of the text document. Bi-grams or 2-grams represent n-grams which have order 2 (consist of two words in a sequence), tri-grams are n-grams which have order 3 (consist of three words in a sequence), and so on. The Bag of N-Grams model is only an extension of the Bag of Words model which use N-gram extracted from features [Sarkar, 2019]. For instance, the output of applying bi-gram is feature vectors for text documents, where each feature is made of a bi-gram depicting a sequence of two words. In this research, *unigram-based features, bigram-based features and tri-gram-based features* are extracted from the corpus [Sarkar, 2019].

TF-IDF (Term Frequency- Inverse Document Frequency) Model

When the Bag of Words model is used on large corpora, some potential problems may arise. Due to the fact that the feature vectors are based on absolute term frequencies in Bag of Word model, therefore, there might be some terms which occur frequently across all documents and these

terms may cause the other terms in the feature set seem less important. Particularly, the words which occur rarely, but those words might be more effective as features to identify specific categories. As a result, TF-IDF is introduced to deal with the issue [Sarkar, 2019].

TF-IDF represents *term frequency-inverse document frequency as a short form.* It's a combination of two metrics, term frequency (tf) and inverse document frequency (idf). This technique was originally developed as a metric to rank the results of search engine based on user queries and TF-IDF has come to be a part of information retrieval and text feature extraction. Moreover, the whole idea of having TF-IDF is to reflect on how important a word is to a document in a collection [Sarkar, 2019].

Extract TF-IDF Features using NLTK

I discuss how to use TF-IDF to vectorize the features (being in the form of words) where word features are converted into numeric vectors.

It is not always necessary to create features using a Bag of Words or count-based model before engineering *TF-IDF* features. The *TfidfVectorizer* in *Scikit-Learn* is used to calculate *tf-idf* directly by taking the raw text data as input and then calculating the term frequencies as well as the inverse document frequencies. Hopefully, this eliminates the need to use *CountVectorizer* to calculate the term frequencies based on the *Bag of Words model*. Moreover, *TfidfVectorizer* enable us to add n-grams to the feature vectors. Figure 3.18 shows a small piece of code for finding numeric vector of features from raw data using TfidfVectorizer and Figure 3.19 shows its output, as it is seen, the result is a sparse matrix [Sarkar, 2019], being general in NLP field.

```
44
45  tfidf_vect = TfidfVectorizer(ngram_range=(1,1), min_df=0., max_df=1., norm='l2',
46  use_idf=True, smooth_idf=True)
47
48  X_tfidf = tfidf_vect.fit_transform(full_corpus['body_text'])
49
50  print("TF-IDF-Matrix Shape: ",X_tfidf.shape)
51  print("Its type: ", type(X_tfidf))
52
53  X_tfidf_df = pd.DataFrame(X_tfidf.toarray())
54
55  X_tfidf_df.columns = tfidf_vect.get_feature_names()
56
57  print(X_tfidf_df.head())
```

Figure 3.18: Small piece of code for finding numeric vector of features from raw data using TfidfVectorizer

```
TF-IDF-Matrix Shape:  (1241, 2969)
Its type:  <class 'scipy.sparse.csr.csr_matrix'>
    00   05   10  100  1000  1080p   11   12  120  1200  1200w  ...  yesterday  yet  york  you  young  your  yours  yourself  youth  zippers  zone
0  0.0  0.0  0.0  0.0   0.0    0.0  0.0  0.0  0.0   0.0    0.0  ...        0.0  0.0   0.0  0.0    0.0   0.0    0.0       0.0    0.0      0.0   0.0
1  0.0  0.0  0.0  0.0   0.0    0.0  0.0  0.0  0.0   0.0    0.0  ...        0.0  0.0   0.0  0.0    0.0   0.0    0.0       0.0    0.0      0.0   0.0
2  0.0  0.0  0.0  0.0   0.0    0.0  0.0  0.0  0.0   0.0    0.0  ...        0.0  0.0   0.0  0.0    0.0   0.0    0.0       0.0    0.0      0.0   0.0
3  0.0  0.0  0.0  0.0   0.0    0.0  0.0  0.0  0.0   0.0    0.0  ...        0.0  0.0   0.0  0.0    0.0   0.0    0.0       0.0    0.0      0.0   0.0
4  0.0  0.0  0.0  0.0   0.0    0.0  0.0  0.0  0.0   0.0    0.0  ...        0.0  0.0   0.0  0.0    0.0   0.0    0.0       0.0    0.0      0.0   0.0
```

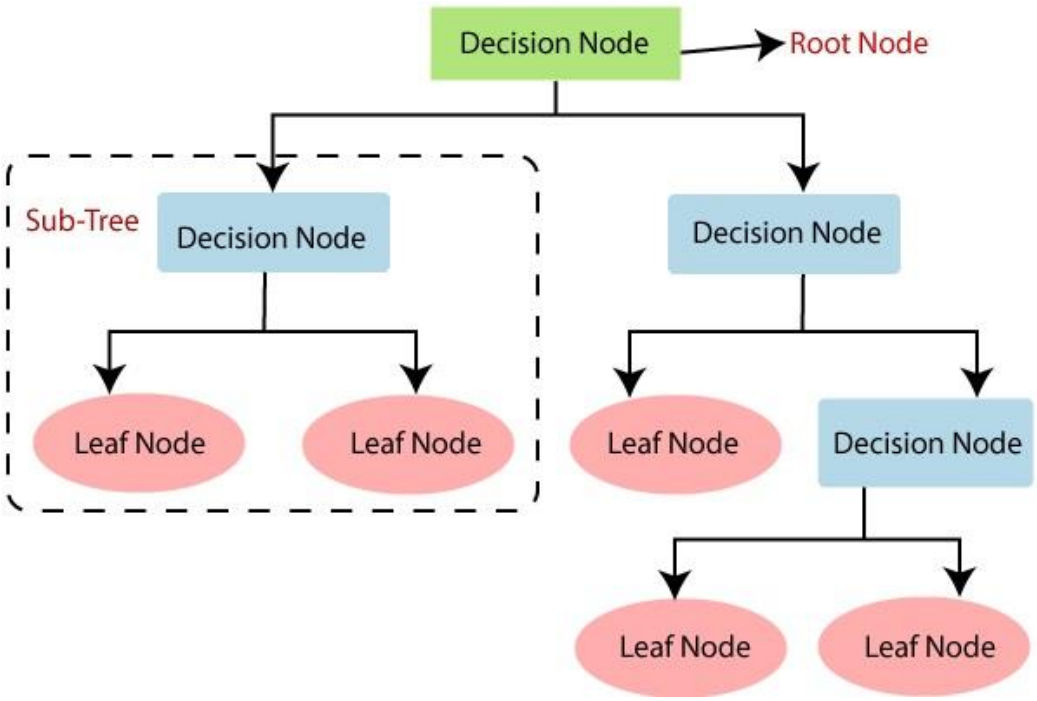
Figure 3.19: A TF-IDF model-based document feature vectors from raw text using TfidfVectorizer

3.4 Classification Methods

3.4.1 Decision Trees

The decision tree is a non-parametric method of data classification that it does not require setting up parameters in advance, as well as initial knowledge of the data. This method is one of the Supervised Learning Methods. In this method, a structure called the decision tree is created using the training data. Using this tree, you can find rules for the problem and these rules are used to categorize the data without labels [Kantardzic, 2003 & 2011].

Generally, a decision-tree learning algorithm adopts a top - down approach which looks for a solution in a part of the search space. A decision tree consists of nodes and leaf. It starts with all the training sets at the root node and a feature is selected to divide these samples. A branch is created for each value of the feature, and the corresponding subset of samples which have the feature value are specified by the branch and are moved to the newly created child node. The algorithm is performed recursively to each child node until all samples at a node are of one class. In the decision tree each path to the leaf shows a classification rule [Kantardzic, 2003 & 2011]. Figure 3.20 depicts a general illustration of Decision Tree.



[Source: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm]

Figure 3.20: A general illustration of Decision Tree

Splitting Criteria in Decision Trees

In decision tree algorithm, the selection of feature which should be tested at each node in the tree is very important. It is clear that we tend to select the feature that is most useful for classifying samples. There are several criteria for determining the feature on which a split should be made [Mitchel, 1997]; two of them are listed below:

- Information Gain
- Gini Index

Definition of Information Gain

To calculate the information gain of a feature, we need to calculate entropy which is used in information theory widely [Mitchell, 1997].

Definition of entropy

Entropy calculates the purity or impurity of that set of data; it is called the amount of clutter in a set. The equation 3.1 shows how it is calculated generally, if the target attribute can take on n different values, and then the entropy D related to this n-wise classification is defined as

$$Entropy(D) = -\sum_{i=1}^{n} p_i \log_2(p_i) \qquad (3.1)$$

$$Pi = |C_{iD}|/|D| \qquad (3.2)$$

In the equation 3.1, |***D***| is the total samples, and ***pi*** is the proportion of ***D*** belonging to class ***i.*** Note the logarithm is still base 2 because entropy is a measure of the expected encoding length measured in ***bits.*** Note also that if the target feature can take on **n** possible values, the entropy can be as large as $\boldsymbol{log_2 n}$.

This probability is estimated by equation 3.2. |***D***| shows the number of samples in ***D*** and $|\boldsymbol{C_{iD}}|$ shows the number of samples in ***D***, which has the class label ***ci***, and ***n*** is the number of existing classes.

Definition of Entropy shows that for a data set which has two classes, value of Entropy will have lowest value when all samples ***D*** belong to a class. In this situation, entropy will be zero. Also, Entropy reaches its maximum value when half of the data belongs to the first class and the other half to the second class, in this situation, the value of entropy will be 1. Figure 3.21 confirms this [Mitchell, 1997].

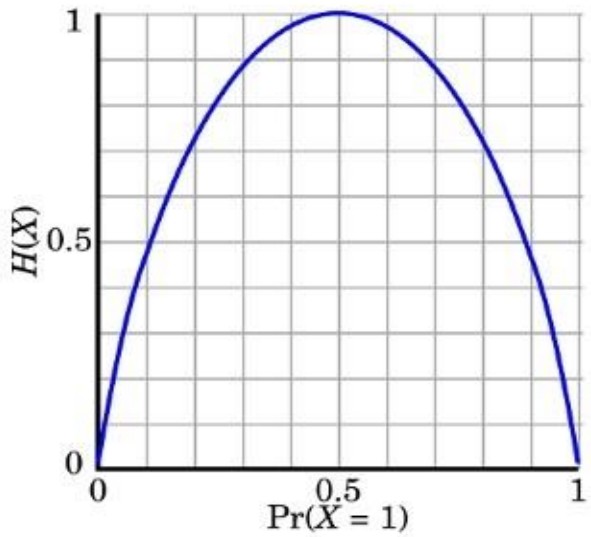


Figure 3.21: Entropy behavior of a data set with two distinct classes [Mitchell, 1997 & Miblog.faradars.org]

If the problem has more than two classes, then the range of changes in entropy will satisfy the equation 3.3.

$$Entropy\ (D) <= log_2\ (n), \quad Entropy\ (D) >= 0 \qquad (3.3)$$

Information Gain

Using entropy as a measure for finding the impurity in a set of training samples, *information gain* now can be calculated. *Information gain* is a measure for finding the effectiveness of an attribute in classifying the training samples. The *Information Gain* of attribute *A* is the amount of *entropy reduction* which is obtained by partitioning the samples according to this attribute. For instance, the *information gain*, ***Gain (D, A)*** of *an* attribute ***A,*** relative to a set of samples ***D,*** is defined in equation 3.4. Assume that the attribute *A* has a distinct value of *v* as *{a1, a2,..., av}*. In other words, *A* is a discrete attribute. If we want to divide *D* by the attribute *A*, *v* Section or subsets such as *{D1, D2,..., Dv}* are obtained, where the *DJ* contains tuples of *D* (a finite ordered list (sequence) of *D*). The value of the attribute *A* in them is equal to *aj*. If we assume that *D* is in a node such as *N*, then *D* segments correspond to the branches that exist in *N*. The Information Gain of feature *A* is the amount of entropy reduction which is obtained by separating the tuples through this property. Equation 3.4 reflects this definition.

$$Gain\ (D, A) = Entropy\ (D) - \sum (|D_v| / |D| * Entropy\ (D_v)) \qquad (3.4)$$

More precisely, in the equation (3.4), values (A) represent the set of all possible values for attribute A, and Dv shows the Subset of D, where A has the value v. The first term in

Equation (3.4) is just the entropy of the original collection *D*, and the second term is the expected value of the entropy after *D* is partitioned using attribute *A*. The expected entropy described by this second term is simply the sum of the entropies of each subset *D*, weighted by the fraction of samples that belong to *D***.** *Gain (D, A)* is therefore the expected reduction in entropy caused by knowing the value of attribute *A* [Mitchell, 1997].

Gini Index

Another Splitting criterion for selecting attributes to split is Gini index. In order to calculate this criterion, the two equations (3.5) and (3.6) are used.

The Gini index is used to select the feature at each internal node of the decision tree. We define the Gini index for a data set S as follows:

$$\text{Gini(S)} = 1 - \sum_{i=0}^{c-1} pi^2 \quad (3.5)$$

Where

• c is the number of predefined classes,

• Ci are classes for i = 1. . . c − 1,

• si is the number of samples belonging to class Ci ,

• p i = s i /S is a relative frequency of class Ci in the set.

The quality of a split on a feature into k subsets Si is then computed as the weighted sum of the Gini indices of the resulting subsets:

$$\text{Gini}_{\text{Split}} = \sum_{i=1}^{k-1} \frac{ni}{n} Gini(Si) \quad (3.6)$$

where

• ni is the number of samples in subset Si after splitting, and

• n is the total number of samples in the given node.

Thus, $\text{Gini}_{\text{split}}$ is calculated for all possible features, and the feature with minimum $\text{Gini}_{\text{split}}$ is selected as the split point [Kantardzic, 2003 & 2011].

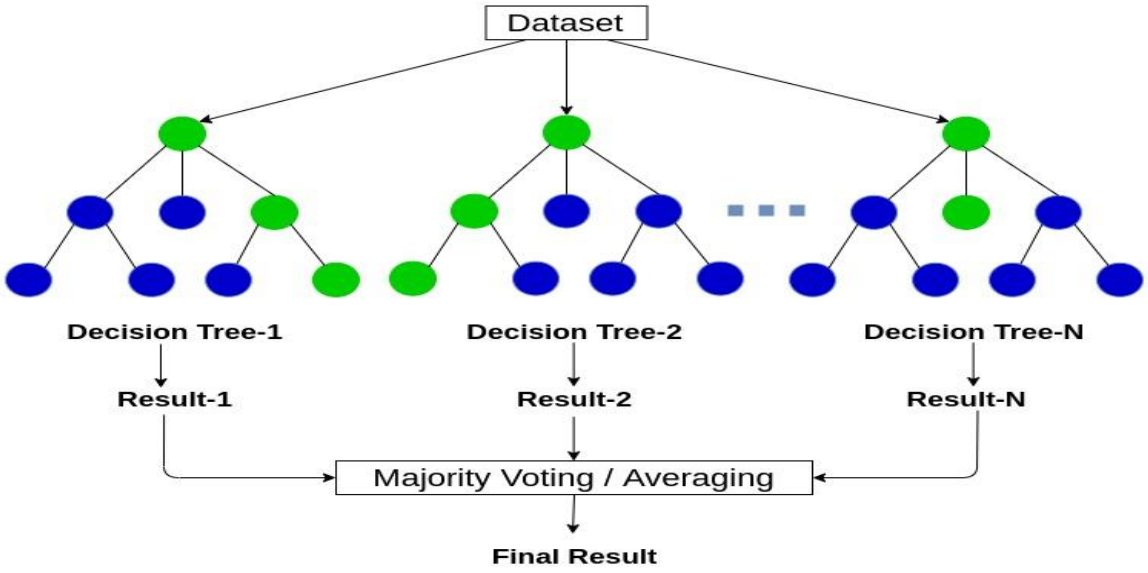
3.4.2 Random Forest

Due to the great attractiveness of decision trees, many researchers have tried to improve its prediction accuracy. It has recently been discovered that one of the best ways to increase the performance of decision tree algorithms is to use an ensemble of trees.

Random Forest is one of the effective algorithm used in both classification and regression applications. Random Forest is a group of unpruned classification or regression trees which are made by using bootstrap samples of the training data set and randomly-selected features [Svetnik et al., 2003]. To construct each decision tree, a top-down partitioning approach is used. A decision tree divides the search space into a set of separate areas and it assigns a response (a vote) to each area. For classification issues, prediction or response of the random forest is based on majority votes. But in regression, prediction or response is based on average responses of all trees for that particular area. At each stage of tree growth using training examples, search is done among features to select the best split point so that impurity reduction (entropy) in the node is achieved.

In fact, the random forest classifier consists of a combination of tree classifiers where each classifier is created using a random vector sampled independently from the input vector, and each tree gives a vote for selecting the class for an input vector [Pal, 2005].

Moreover, Random Forest adds more randomness to bagging algorithm. Not only constructing each tree using a different bootstrap sample of the data set, but also it changes how the classification or regression trees are made. In standard trees, each node is split using the best split among all variables. In a random forest, each node is split using the best features among a subset of features which are chosen at that node randomly. In addition, Random Forest just needs two parameters, namely the number of features in the random subset at each node and the number of trees in the forest to be set [Liaw & Wiener, 2002]. Figure 3.22 shows an example of Random Forest and Figure 3.23 gives the algorithm.



[Source: https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/]

Figure 3.22: A sample of Random Forest

1. For $b = 1$ to B:
   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.
   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      i. Select m variables at random from the p variables.
      ii. Pick the best variable/split-point among the m.
      iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

Figure 3.23: Algorithm of Random Tree [Liaw & Wiener, 2002]

3.4.3 Gradient Boosting

Gradient Boosting Decision Tree (GBDT) is one of the Ensemble Learning Method. It is largely used because of its efficiency and accuracy. According to Ke et al., 2017, GBDT achieves state-of-the-art performances in many machine learning tasks, such as multi-class classification, click prediction, and learning to rank [Ke, et al., 2017]. However, in conventional implementations of GBDT, it calculates the information gain of all the possible split points, every feature, in all instances in the data set. Therefore, their computational complexities will be proportional to both the number of features and the number of instances. This makes these implementations very time consuming when handling big data.

Gradient Boosting algorithm

Gradient Boosting Method is based on decision trees, which are trained in sequence. It takes an iterative approach to combine weak learners to create a strong learner by focusing on mistakes of prior iterations. In each iteration, GBDT learns the decision trees by fitting the negative gradients (also known as residual errors).

Complexity Analysis in Gradient Boosting

The main cost in GBDT lies in learning the decision trees, and the most time-consuming part in learning a decision tree is to find the best split points. One of the most common algorithms to find split points is the pre-sorted algorithm, which enumerates all possible split points on the pre-sorted feature values. This algorithm is simple and can find the optimal split points; however, it is not efficient in training speed and memory consumption. Another popular algorithm is the histogram-based algorithm.

Difference between GBDT and RF

Random Forest and Gradient Boosting are based on decision trees. But the difference is that Gradient Boosting uses a method which is called boosting.

3.4.4 Support Vector Machine (SVM)

The support vector machine (SVM) is a supervised learning method which creates an input-output mapping function from the available training data set with labels [Wang *et al.* 2005]. This method has been largely used for classification and nonlinear regression. The mapping function

can be either a classification function which classifies the training data set, or a regression function. Joachims mentioned in 1998 that one of the significant features of SVMs is that they are able to learn the training data independent of dimensionality of the feature space also they can be used to learn polynomial classifiers [Joachims, 1998]. Joachims in 1998 mentioned also that SVMs are very useful in text classification and most text classification problems can be linearly separable.

SVM theory is mainly derived from the problem of binary classification. Its main idea can be concluded as the following two points: First, it constructs a nonlinear kernel function to present an inner product of feature space, which corresponds to mapping the data from the input space into a possibly high-dimensional feature space by a nonlinear algorithm. Thus it is possible to analyze the nonlinear properties of samples in the feature space with linear algorithm. Secondly, it implements the structural risk minimization principle in statistical learning theory by generalizing optimal hyper-plane with maximum margin between the two classes. Although intuitively simple, this idea actually plays the role of capacity controlling and makes the learned machine not only has small empirical risks, but also has good generalization performance. Therefore, SVM has many advantages in both theoretical base and practical prospect [Xiao, Wang & Zhang, 2000].

In this research SVM algorithm is used to classify user opinion about some products in Amazon website which is extracted from the repository collected by Hu. SVM classifier is trained with a part of the labeled data set and in the second step (which is prediction part) the trained data in first step is used to classify the test set.

3.4.5 Neural Networks

A neural network consists of many processing elements joined together to form an appropriate network with adjustable weighting functions foreach input. These processing elements are usually organized into a sequence of layers with full or random connections between layers. Typically, there are three or more layers: an input layer where data are presented to the network through an input buffer, an output layer with a buffer that holds the output response to a given input, and one or more intermediate or "hidden" layers.

The operation of an artificial neural network involves two processes: learning and recall. Learning is the process of adapting the connection weights in response to stimuli presented at the input buffer. The network "learns" in accordance with a learning rule which governs how the connection weights are adjusted in response to a learning example applied at the input buffers. Recall is the process of accepting an input and producing a response determined by the learning of the network [Uhrig, 1995].

Chapter 4

Results and Discussion

4.1 Introduction

In this chapter, to investigate the effect of various parameters of the proposed model on the accuracy of identification, some experiments have been designed and implemented. With these tests, acceptable values are determined for the free parameters of the model. In the following, it has been compared with other available methods to show the power and efficiency of the proposed method and the obtained results have been shown. Generally, in all experiments, the proposed NLP techniques are used to pre-process the corpus and to extract features (like finding all 1-gram, 2-gram and 3-gram). After that each opinion is vectorized by the mentioned methods like TF-IDF approach. Therefore, texts (or words of each opinion) are converted into vectors of numbers with the help of TF-IDF method. Then various classifiers are used to classify the data.

4.2 Investigate the various parameters in the Random Forest

In general, in each algorithm there are a number of free and effective parameters for the efficiency of the algorithm. Random forest algorithms are no exception and it has a number of parameters, the optimal setting of which increases the accuracy and strength of the algorithm in

In this section, the effect of these parameters is shown. All tests, which are done in this section, are performed on the data set (introduced in previous chapter) and the results are presented. To compare and to evaluate this method with other methods, some criteria for evaluating

classification algorithms such as accuracy, recall, (sensitivity), precision and F-Measure are used. Equation 4.1 shows accuracy formula, equation 4.2 shows precision formula, equation 4.3 shows recall formula and equation 4.4 shows F1 score.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (4.1)$$

$$\text{Precision} = TP / TP + FP \quad (4.2)$$

$$\text{Recall} = TP / TP + FN \quad (4.3)$$

$$\text{F1 score} = 2*(\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision}) \quad (4.4)$$

A **true positive (TP)** is a result that the model *correctly* predicts the *positive* class and a **true negative** is a result that the model *correctly* predicts the *negative* class. A **false positive (FP)** is a result that the model *incorrectly* predicts the *positive* class. And a **false negative (FN)** is a result that the model *incorrectly* predicts the *negative* class.

4.2.1 Investigate number of trees parameter

One of the most important and effective parameters in the accuracy of classification and precision is the number of trees in the forest. Each of the trees in the forest alone does not have very well efficiency. Indeed, the classification strength of random forest depends on the strength of all voted trees in the forest. It is a feature of ensemble methods that uses a kind of voting system between members for final decision-making.

Increase in the number of trees in the forest lead to the improvement of confidence in the final diagnosis based on the tree's votes. To see the effect of this parameter on algorithm's performance and choosing the right number of trees in the forest, the following experiment was performed.

Experiment 1- different number of trees

In this experiment, a random forest algorithm with the following conditions is implemented:

- Splitting Criteria: Gini Index
- "m": number of variables
- Number of trees: varied from 10 to 150, with an incremental step of 10.

The parameter "m" (which shows the number of extracted features) is equal to the total number of features which is extracted in statistical approach. The possible values for n in the extraction of n-grams will be 1, 2 and 3.

For each separate n, the obtained feature space is different. If n = 1, then the feature space is equal to 1539. If n = 2, then the feature space is equal to 4509, and finally, if n = 3, we are dealing with a feature space which has 4279 dimension (By writing codes in python software, I found the dimension of feature spaces). Hence, this experiment is repeated for each feature space.

Part A- 10% of the data set is used for testing

In this experiment, 90% of the data was used to train the model and construction of original random forest and 10% data is used to test the model and to evaluate the model. For evaluating the proposed method various evaluations were performed and the results are listed in table 4.1.

Table 4.1: Performance measures on unigrams, bigram and trigrams on different number of trees, with 10% of data (test size)

Feature Space	**Measures in%**	**Number of Trees**														
		10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**	**110**	**120**	**130**	**140**	**150**
Speaker /Unigram	***Accuracy***	67	70	70	72	72	72	72	72	72	72	71	71	71	71	71
	Precision	45	79	79	81	81	81	81	81	81	81	80	80	80	80	80
Speaker /Bigram	***Accuracy***	59	59	62	62	61	59	59	59	59	59	59	61	59	61	62
	Precision	72	72	76	76	74	72	72	72	75	72	72	74	72	74	76
Speaker /Trigram	***Accuracy***	58	58	58	58	58	58	58	58	58	58	58	58	58	58	58
	Precision	86	86	86	86	86	86	86	86	86	86	86	86	86	86	86

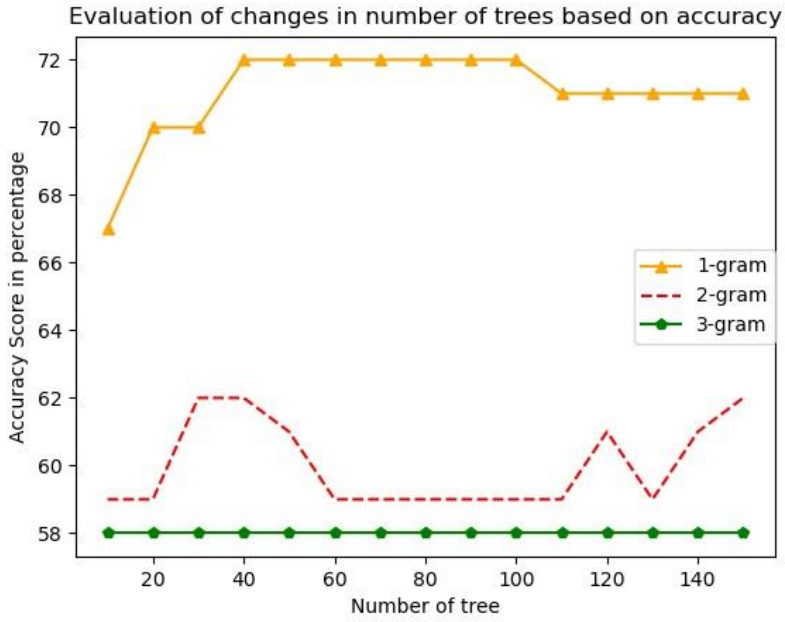


Figure 4.1: Accuracy measure for 1-gram, 2-gram and 3-gram as a function of number of trees, with 10% of data

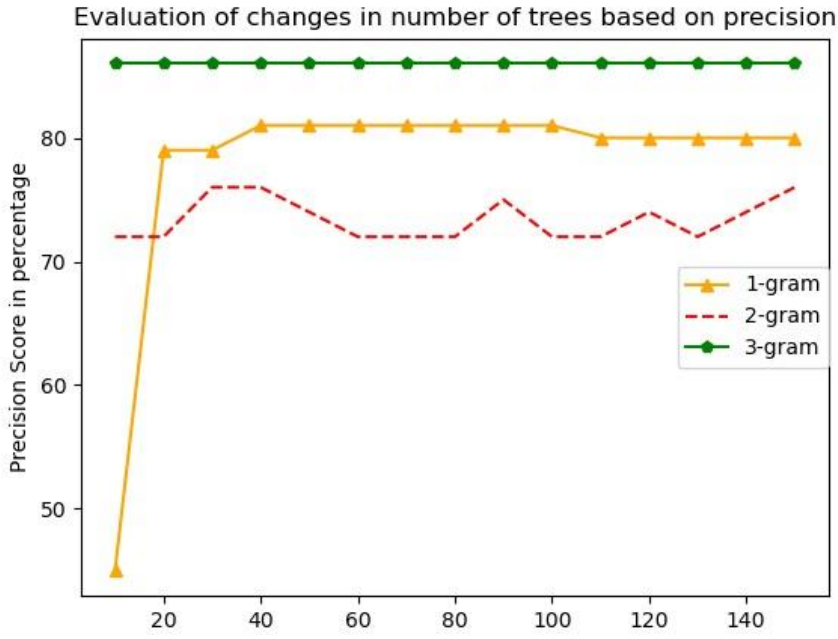


Figure 4.2: Precision on 1-gram, 2-gram and 3-gram as a function of number of trees, with 10% of data

Discussion

The results in Table 4.1, Figure 4.1 and Figure 4.2 show that precision reaches its highest point of 81% when the number of trees is 40-100 in terms of unigrams. And as the number of trees increase from 10 to 40, there is a little improvement in accuracy, from 67% to 72%, and it remains stable when the number of trees increases from 40-100.

Additionally, In terms of Bigram, Precision is steady at 72% when the number of trees increases from 10 to 20. Then it increases to 76% when the number of trees is increased to 30. And it remained steady when the number of trees is increased to 40, followed by some fluctuations when the number of trees is between 50 and 130.Then there is an improvement from 130 to 150 where it reaches 76%. So, precision reaches its highest point of 76% when the numbers of trees are 30, 40 and then 150, and accuracy is quite low at the whole trend, between (59%-62%). Moreover, In terms of trigram, precision stay steady at 86% at the whole trend, and accuracy is quite low, 58%.

Part B- 20% of the data set is used for testing

In this experiment, 80% of the data was used to train the model and construction of original random forest and 20% data was used to test the model and to evaluate the model. After performing this experiment, various evaluations were performed and the results are listed in Table 4.2. Figure 4.3 shows the graph for the accuracy measure for unigrams, bigrams, and trigrams on different number of trees. Figure 4.4 shows the precision on unigrams, bigrams and trigrams for different number of trees.

Table 4.2: Performance measures on unigrams, bigrams and trigrams for different number of trees, with 20% of data (test size)

Feature Space	**Measures in%**	**Number of Trees**														
		10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**	**110**	**120**	**130**	**140**	**150**
Speaker /Unigram	***Accuracy***	73	74	74	74	74	74	74	75	75	75	75	75	75	74	74
	Precision	50	83	82	82	82	82	82	83	83	83	83	84	83	82	82
Speaker /Bigram	***Accuracy***	61	61	61	62	59	59	61	61	60	61	61	61	61	60	61
	Precision	78	80	80	81	75	75	78	78	76	78	78	78	78	76	78
Speaker /Trigram	***Accuracy***	57	57	57	57	57	57	57	57	57	57	57	57	57	57	57
	Precision	86	86	86	86	86	86	86	86	86	86	86	86	86	86	86

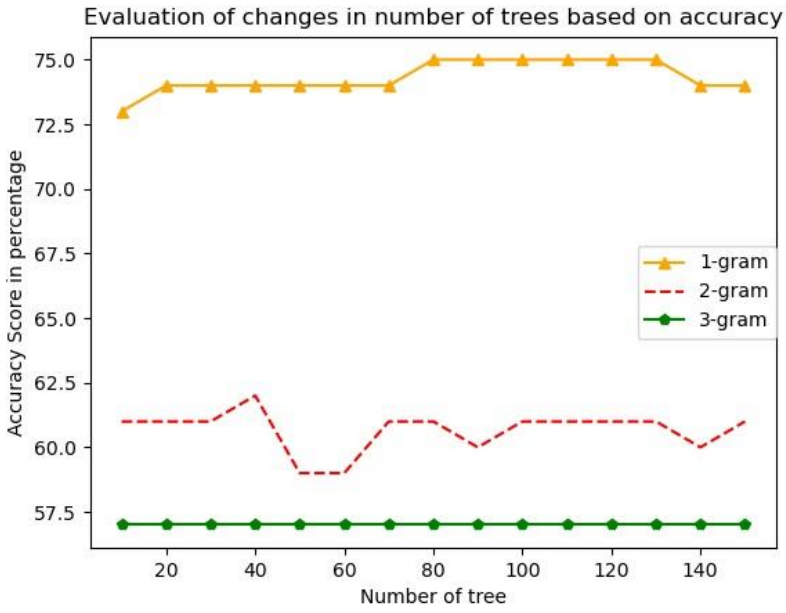


Figure 4.3: Accuracy for 1-gram, 2-gram and 3-gram as a function of number of trees, with 20% of data

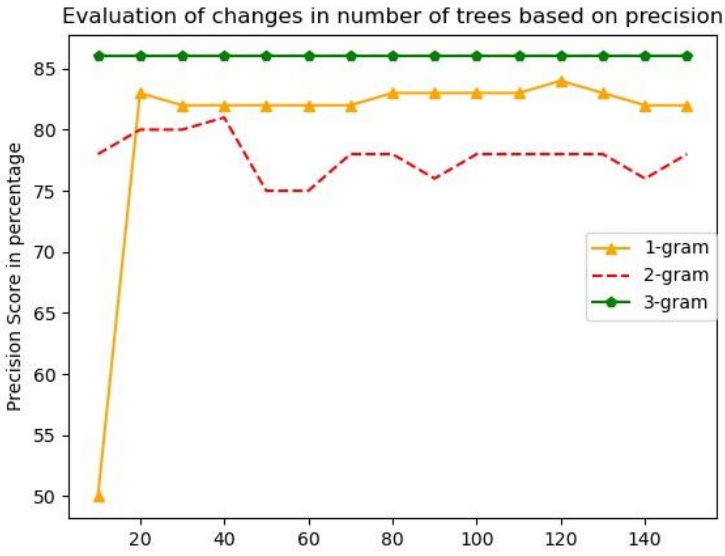


Figure 4.4: Precision for 1-gram, 2-gram and 3-gram as a function of number of trees, with 20% of data

Discussion

The result in Table 4.2, Figure 4.3 and Figure 4.4 shows that there is dramatic increase in precision from 50 to 83 when the number of trees increases from 10-20, followed by fluctuation at the whole trend also reach the maximum value about 84% in 120 trees. Moreover, accuracy increases from 73% to 74% when the number of trees increases from 10 to 20, and then it stays stable (at 74%) when the number of trees increases from 20 to 70. After that it increases to 75% when the number of trees increases to 80, then it stays stable (75%) until the number of tree increases to 130. Then there is slight decrease in the rest of the trend.

In terms of Bigram, there is improvement in precision when the number of trees going from 10-20 (from 78%-75%), after that it is stable until the number of trees is 30. Then it reaches its highest point (81%) when the number of trees is 40. And Accuracy is quite low at the whole trend, between (60%-62%).

In terms of trigram, as it is seen in the table, precision remained stable at 86% with the increase in the number of trees as well as this, Accuracy remains stable at 57% which is quite low. Overall, if we want to have highest precision we need to use trigram and the number of trees can be 10-150. But highest accuracy can be achieved in this test when unigram is used and the number of trees is 80-130.

Part C- 30% of data is used for testing

Table 4.3: Accuracy, precision for unigram, bigram and trigrams with different number of trees, with 30% of data (test size)

Feature Space	**Measures in%**	**Number of Trees**														
		10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**	**110**	**120**	**130**	**140**	**150**
Speaker /Unigram	***Accuracy***	73	73	76	73	75	75	75	75	75	75	75	75	75	75	74
	Precision	49	81	84	81	82	83	83	82	82	83	83	83	83	83	82
Speaker /Bigram	***Accuracy***	62	63	62	63	63	64	63	64	64	64	64	63	63	64	64
	Precision	44	83	77	81	83	84	83	84	84	84	84	83	83	88	88
Speaker /Trigram	***Accuracy***	61	61	61	61	61	61	61	61	61	61	61	61	61	61	61
	Precision	87	87	87	87	87	87	87	87	87	87	87	87	87	87	87

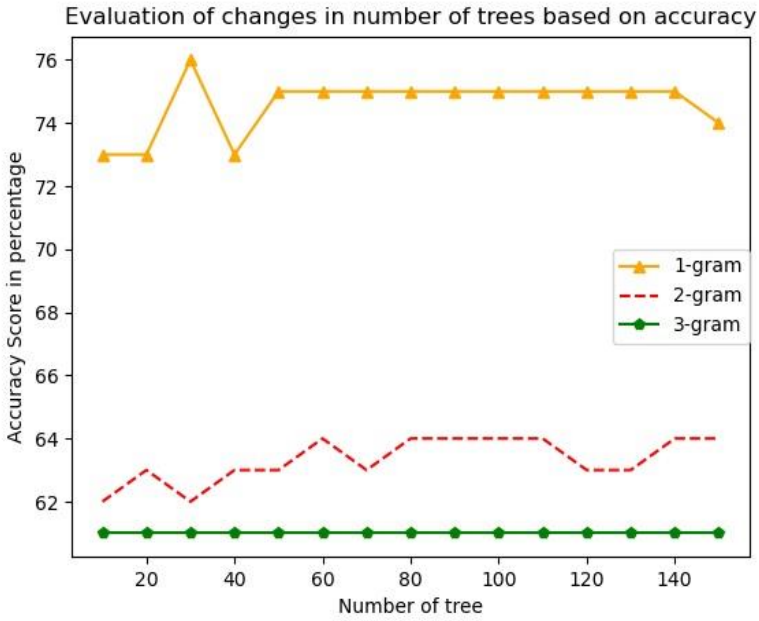


Figure 4.5: Accuracy as a function of number of trees for 1-gram, 2-gram and 3-gram, with 30% of data

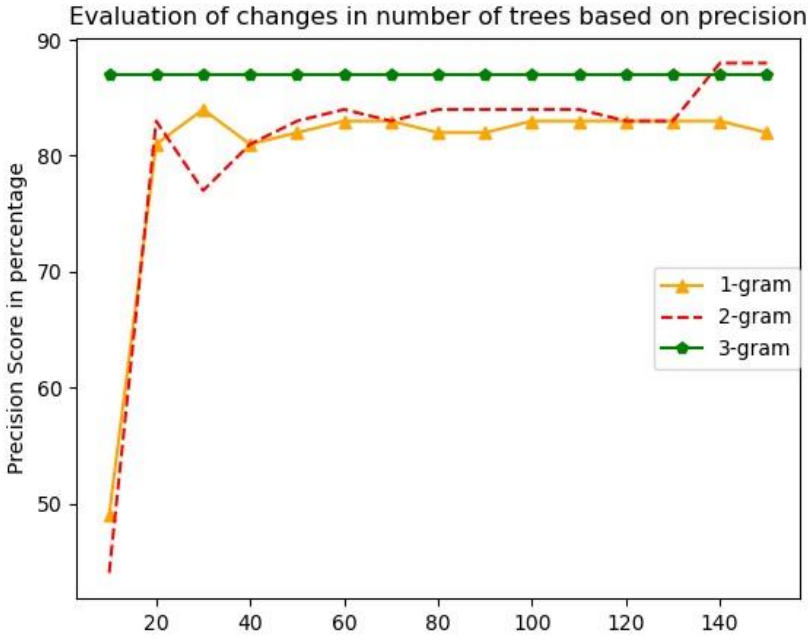


Figure 4.6: Precision as a function of number of trees for 1-gram, 2-gram and 3-gram, with 30% of data

Discussion

In different parts of experiment one, I tested my model with different test size. Overall, I found that when my test size is 30%, the proposed model (using Random Forest) represents the better accuracy and precision, particularly, when Bi-gram feature space is used. Therefore, in this research I select 30% of the data as test set for evaluating and comparing other parameters and classifiers in the next sections.

As it is seen in table 4.3 and figure 4.6, when I use 2-gram and 3-gram, overall precision is higher (where test size is 30% of data set). As long as my data set is unbalanced, accuracy is not a good criterion for evaluating a classifier with very unbalanced data. In such cases, precision is an appropriate tool for evaluating a classifier performance, so I focused on precision due to this issue in my data set.

Generally, I tried to find the best values for number of trees of Random Forest by performing experiment 1 .And according to the results shown in figures 4.6, I understood that a forest which can categorize test data with the best amount of precision must contain at least 140 trees (where test size is 30% of data set).

4.2.2 Investigate parameter M, number of features

Experiment 1- evaluating different parameters for M, number of features on Speaker data

In this experiment, random forest algorithm with the following conditions is implemented:

- Splitting Criteria: Gini Index
- “m”: number of variable
- Number of trees: 140

- Test data: 30 % of the data set

The parameter “m” (which shows the number of extracted features) is equal to the total number of features that are extracted in statistical approach. The possible values for n in the extraction of n-grams will be 1, 2 and 3.

For each separate n, the obtained feature space is different. If n = 1, then the feature space is equal to 1539. If n = 2, then the feature space is equal to 4509, and finally, if n = 3, we are dealing with a feature space which has 4279 dimension. Hence, this experiment is repeated for each feature space.

The common values for” m” is Square root of the number of variables, logarithm 2 of the number of variable and all variables.

Table 4.4: Performance of unigrams, bigrams and trigrams with different values of parameter m

Parameter m	**Feature Space**	**Accuracy**	**Precision**	**Recall**	**F1**
Sqrt(nvariables)	***Unigram***	75	83	52	50
	Bigram	64	88	37	32
	Trigram	61	87	34	26
Log2(nvariables)	***Unigram***	75	84	52	51
	Bigram	63	87	36	31
	Trigram	61	87	34	26
Nvariables	***Unigram***	75	83	52	50
	Bigram	64	88	37	32
	Trigram	61	87	34	26

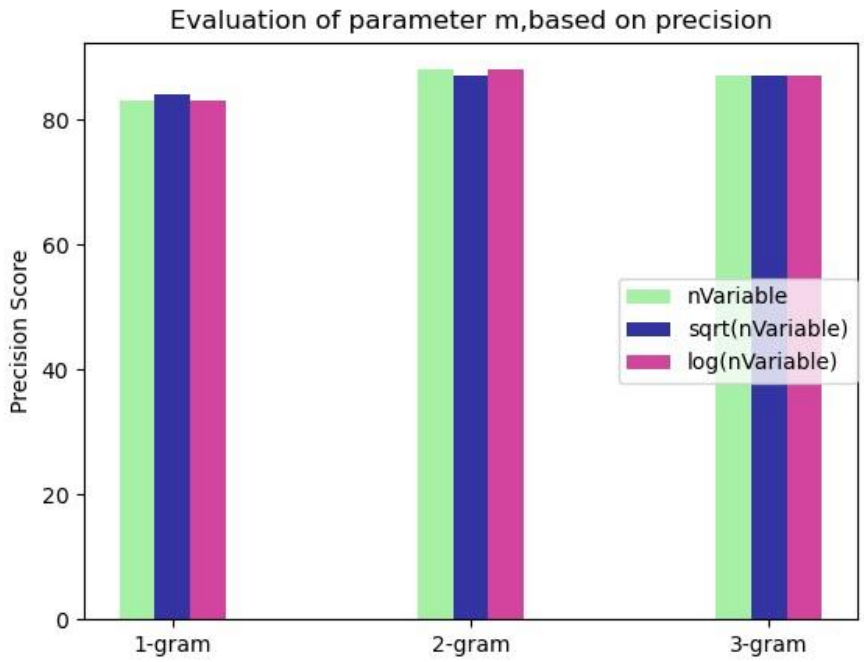


Figure 4.7: Comparison of precision score in terms of different values of parameter m

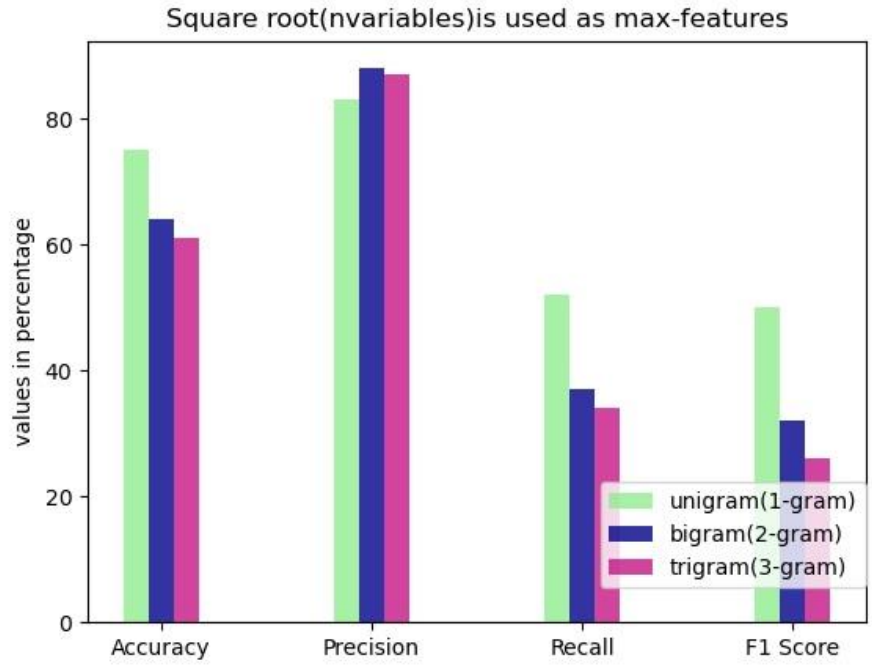


Figure 4.8: Comparison of performance criteria based on m = Square root (nvariables)

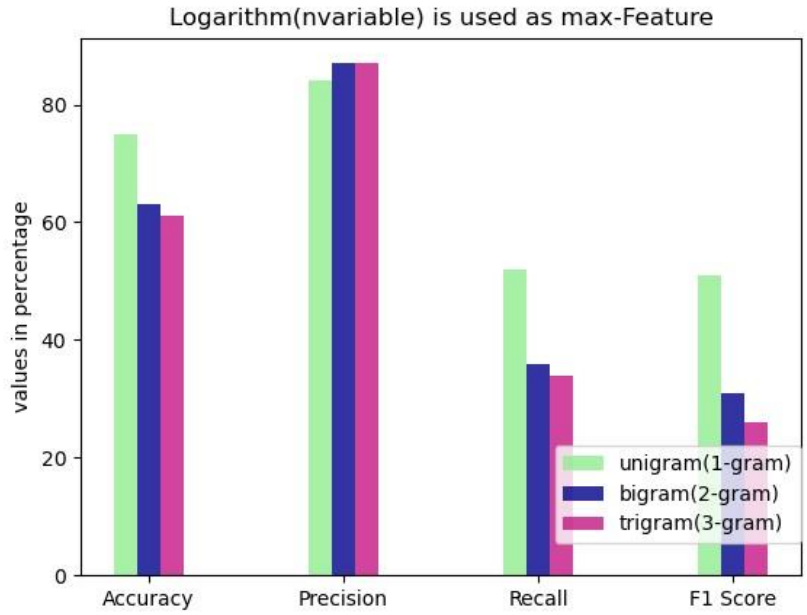


Figure 4.9: Comparison of performance criteria based on m = Log (nvariable)

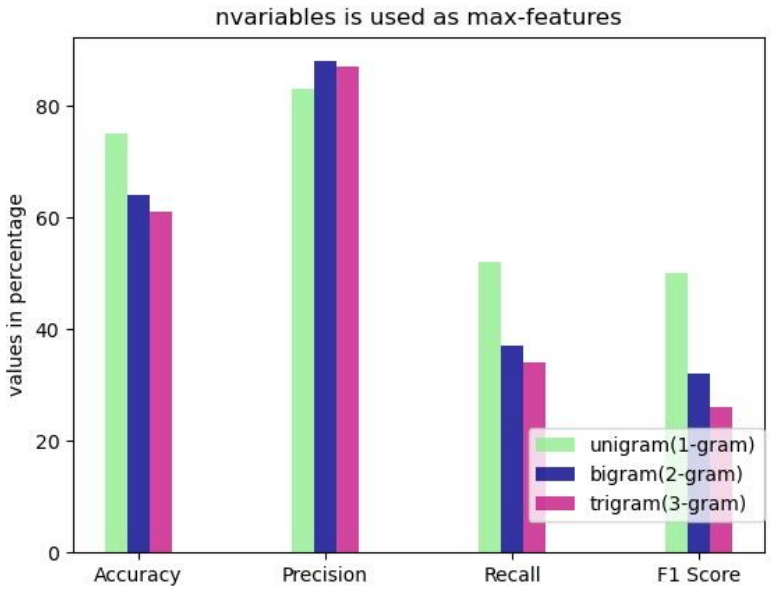


Figure 4.10: Comparison of performance criteria based on m = nvariable

Discussion

By performing experiment 1, I tried to find the best value for number of features need to consider for splitting in each node in Random Forest algorithm (I selected parameter m to show it). Based on results in table 4.4 and figure 4.7, 4.8, 4.9 and 4.10, it is clear that reducing the number of features to sqrt (nvariable) and log2 (nvariable) does not lead to improvement. In fact, according to the results in Table 4.4, if I set this parameter to the number of features or square root of it in the data set, I get the highest precision about 88%. (Of course not much difference from the score in log2).

Generally, the remarkable thing is that this feature space is very large, and every opinion is converted to a vector; so if the features do not exist in this opinion, it takes a value of zero in the vector. Therefore, the number of zeros in the samples will be very large. And by decreasing the value of m, the valuable features which have opposite values of zero may be lost for selecting the splitting features in tree nodes. Consequently, trees are made with less important features which are zero in most samples. As a result, the precision, accuracy and efficiency of the model decrease.

4.3 Comparing the Classifiers

In this experiment, different classifiers are implemented and compared.

Conditions:

- Number of trees: 140
- Number of max features: square root (number of variables)
- Split criteria: Gini

- Extracted different features: 1-gram, 2-gram, 3-gram
- Test data: 30 % of the data set
- Neural Network: 2 layers, in first layer it has 5 neurons and 3 neurons in hidden layer, and active function is hyperbolic tangent

Table 4.5: Evaluating different classifiers with different measures and feature spaces

Feature space	**Measures in %**	**ANN(5,3) ,hyperbolic tangent**	**SVMs (linear)**	**Gradient Boosting**	**Random Forest (tree=140)**
Unigram	***Accuracy***	73	76	74	75
	Precision	65	84	82	83
	Recall	55	52	51	52
	F1-score	57	51	50	50
Bigram	***Accuracy***	68	66	64	64
	Precision	77	85	78	88
	Recall	44	39	38	37
	F1-score	43	36	35	32
Trigram	***Accuracy***	61	61	61	61
	Precision	87	87	87	87
	Recall	34	34	34	34
	F1-score	26	26	26	26

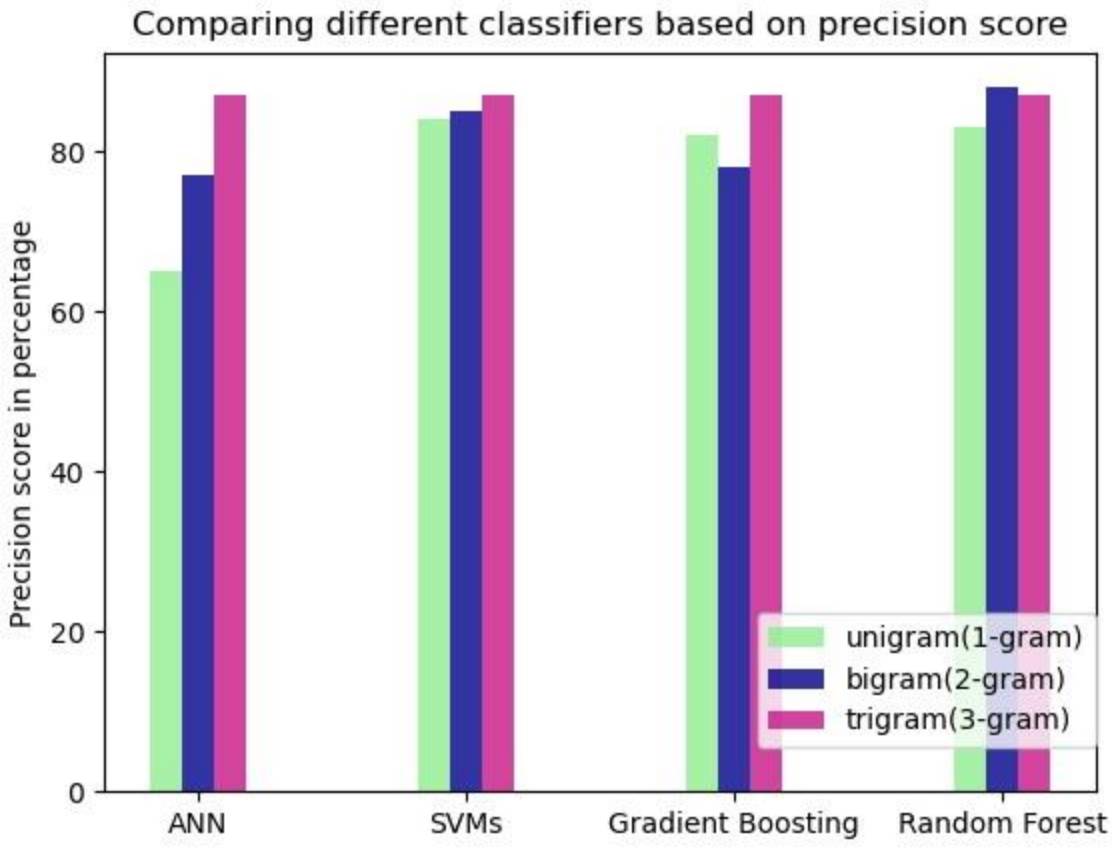


Figure 4.11: Comparing different classifiers based on precision score

Discussion

According to table 4.5 and figure 4.11, Random Forest reaches a peak of 88% at precision with 140 trees and bigram feature space. Also, the proposed Random Forest and Gradient Boosting, Artificial Neutral Network (having 2 layers), SVMs reach 87% precision when trigram feature space is used. Therefore, as I mentioned, selecting Bigram for Random Forest is the best choice also in the next level, trigram is good as well. In fact, According to results, Random Forest represents better precision than other classifiers when the test size is 30% and Bi-gram feature space is used. But, when unigram is used, precision of SVMs is better than

Random Forest about 1%. Moreover, when trigram feature space is used, score of all criteria like accuracy (61%), precision (87%), recall (34%) and F1-score (26%) is the same in Random Forest, Gradient Boosting ANN, and SVMs.

As long as my data set is unbalanced, accuracy is not a good criterion for evaluating a classifier with unbalanced data. In such cases precision is a good tool for evaluating a classifier, so I focused more on precision due to this issue in my data set.

Chapter 5

Conclusions and Future Work

5.1 Conclusion

In this thesis, a unique model in order to do opinion mining on raw textual data set (user comments) has been introduced using various machine learning algorithms and Natural Language Processing techniques. The detailed steps of removing noise and pre-processing using Natural Language Processing techniques (NLP) is explained and represented in this thesis such as converting all words to lowercase, removing punctuations, removing URLs, numbers, unnecessary spaces, tokenization, stemming and removing stop-words. As user comments are conversational sentences, raw and unstructured or semi-structured and contain insignificant characters (noise), which are not usable without initial processing, it is fundamental to apply some sequential pre-processing steps. After that Bag of word (BOW) and Bag of N-grams are used to extract features. Appropriate features, namely 1-grams, 2-grams, 3-grams were extracted. Bag of word and N-gram is known as NLP techniques, the output being user comments in a new feature space. Then the extracted features were quantified by using term frequency-inverse document frequency (*TF-IDF*). With TF-IDF technique, which is a Natural Language Processing technique there is no need for lexicons. Four classifiers, Random Forest, ANN, SVMs and Gradient Boosting were used to predict the polarity of the comments.

The result of this research was introduction of a model where *TF-IDF* technique is used for opinion mining on categorizing user comments. The proposed model gave a peak of 88% precision by Random Forest with 140 trees and bigram feature space. Also, Random Forest, Gradient Boosting, Artificial Neutral Network, SVMs gave 87% precision for trigram feature space. Therefore, selecting bigram with random forest is the best choice and at the second level trigram with all the classifiers. As the data set is unbalanced, accuracy is not a good criterion for evaluating a classifier with very unbalanced data. In such cases precision is a good measure for evaluating a classifier. Moreover, when trigram feature space is used, score of all criteria like accuracy (61%), precision (87%), recall (34%) and F1-score (26%) is the same in Random Forest, Gradient Boosting ANN, and SVMs; these classifiers have the same performance when trigram feature space is used.

5.2 Future work

The present study focuses specifically on special category of opinions. To evaluate the proposed method, this model can be used on other datasets which have more data or can be used in other similar fields such as opinion mining on social networks having more users and resources. Furthermore, other criteria can be used for better evaluation of a model.

In this study, the existing opinions were classified into only three categories: "positive", "negative" and "neutral". In order to examine more comments and extract accurate information from them, comments can be labeled into more categories such as "very good", "good", "ineffective", "bad" and "very bad." Moreover, a bigger training data is needed to fully train the learning model. Additionally, the data set should be of good quality.

References

[1] Chaovalitand P, Zhou L. “Movie review mining: A comparison between supervised and unsupervised classification approaches”. In Proceedings of the 38th annual Hawaii international conference on system sciences, IEEE, 2005

[2] Hemmatian F, Sohrabi M.K, “A survey on classification techniques for opinion mining and sentiment analysis”, Springer Science&Business Media, 2017

[3] Keshwani K, Agarwal P, Kumar D, “Prediction of Market Movement of Gold, Silver and Crude Oil Using Sentiment Analysis, in Advances in Computer and Computational Sciences, 2018, Springer, PP 101-109

[4] Bertola F, Patti V, “Ontology-based affective models to organize artworks in the social semantic web”. In Information Processing & Management, 2015, Elsevier, PP 1-24

[5] Lasota T, Luczak T, Trawiński B, “Investigation of Rotation Forest Method Applied to Property Price Prediction”. In Artificial Intelligence and Soft Computing, 2012, Springer, pp 403–411

[6] Rodriguez J, Kuncheva L, Alonso C, “Rotation Forest: A new classifier ensemble method”. IEEE transactions on pattern analysis and machine intelligence, 2006, PP 1619-1630

[7] Kouloumpis E, Wilson T, Moore J, “Twitter sentiment analysis: The good the bad and the omg!”, In Proceedings of the Fifth International Conference on Weblogs and Social Media, Spain, 2011, PP 538-541

[8] Abbasi A, Chen H, Salem A, “Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums”, ACM Transactions on Information Systems (TOIS), 2008, PP 12-34

[9] Severyn A, Moschitti A, "Multi-lingual Opinion Mining on YouTube” ,In Information Processing and Management, 2015, Elsevier, PP 46-60

[10] Poria S, Cambria E, Gelbukh A, "Aspect extraction for opinion mining with a deep convolutional neural network", Knowledge-Based Systems, 2016, Elsevier, PP 42–49

[11] Huang W, Zhao Y, “Analysis of the user behavior and opinion classification based on the BBS”. Applied mathematics and computation, 2008, PP 668-676

[12] Nakov ,P, Rosenthal ,S, Kiritchenko ,s, Mohammad ,S.M, Kozareva , Z, Ritter ,A, Stoyanov ,V, Zhu ,X,"Developing a Successful SemEval Task in Sentiment Analysis of Twitter and Other Social Media Texts", data mining, 2016, PP 35-65

[13] Chen Z, Liu B, “Mining topics in documents: standing on the shoulders of big data”, in Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, PP 1116–1125

[14] Hu, M, Liu, B, “Mining and summarizing customer reviews”. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-04), 2004, PP 168-177

[15]. Hu, M, Liu, B, "Mining Opinion Features in Customer Reviews”. In Proceedings of nineteenth National Conference on Artificial Intelligence (AAAI-2004), 2004, PP 755-760

[16] https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume11/opitz99a-html/node3.html

[17] Juan Ramos, Department of Computer Science, Rutgers University "Using TF-IDF to Determine Word Relevance in Document Queries"

[18] Andy Liaw and Matthew Wiener, "Classification and Regression by Random Forest", 2002

[19] Hastie, Tibshirani & Friedman, "The Elements of Statistical Learning", Second Edition, springer, 2017

[20] Svetnik, Liaw, Tong, Culberson, Sheridan & Feuston, "Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling", American Chemical Society, 43, 1947-1958, 2003

[21] Mitchell T, "Machine Learning". Published by McGraw-Hill, Maidenhead, U.K., International Student Edition, 1997

[22] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, and Qiwei Ye, Tie-Yan Liu, LightGBM: A Highly Efficient Gradient Boosting Decision Tree, 2017

[23] Lipo Wang, Support Vector Machines: Theory and Applications

[24] Thorsten Joachims, Text Categorization with Support Vector Machines: Learning with Many Relevant Features, Universit□at Dortmund Informatik LS8, Baroper Str. 301 44221 Dortmund, Germany

[25] Kulkarni, Shivananda, Natural Language Processing Recipes Unlocking Text Data with Machine Learning and Deep Learning using Python, 2019

[26] Dipanjan Sarkar, Text Analytics with Python, A Practitioner's Guide to Natural Language Processing, 2019, *Second Edition*

[27] Kantardzic M, DATA MINING Concepts, Models, Methods, and Algorithms, Second Edition, 2011, IEEE

[28] Kantardzic M, DATA MINING Concepts, Models, Methods, and Algorithms, John Wiley & Sons, 2003

[28] Guolin Ke, Qi Meng, Thomas Finley, TaifengWang, Wei Chen, Weidong Ma, and Qiwei Ye, and Tie-Yan Liu, LightGBM: A Highly Efficient Gradient Boosting Decision Tree

[29] L. Breiman, "Bagging Predictors," Machine Learning, vol. 24, no. 2, pp. 123-140, 1996

[30] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001

[31] S. Rill, D. Reinel, J. Scheidt, Roberto V. Zicari, PoliTwi: "Early detection of emerging political topics on twitter and the impact on concept-level sentiment analysis", 2014

[32] J. Evermann, Rehse J, P. Fettke , "Predicting process behavior using deep learning",2017

[33] R. Xiao, J. Wang, F. Zhang, "An Approach to Incremental SVM Learning Algorithm", 2000

[34] Robert E. Uhrig, "Introduction to Artificial Neural Networks", 1995