A Major Project-II Report

On

Survey On Dual Implementation In Sentiment Analysis

Submitted in Partial fulfilment of the Requirement for the Degree of

Master of Technology

in

Computer Science and Engineering

Submitted By

Diwakar Kumar

2K18/SWE/21

Under the Guidance of

Prof Rajni Jindal

(Assistant Professor)

DELHI TECHNOLOGICAL UNIVERSITY (Formerly Delhi College of Engineering)
Shahabad Daulatpur, Main Bawana Road, Delhi-110042

June 2020

CERTIFICATE

This is to certify that Project Report entitled **“Survey on Dual Implementation in Sentiment Analysis”** submitted by **Diwakar Kumar** (roll no. 2K18/SWE/21) in partial fulfilment of the requirement for the award of degree Master of Technology (Computer Science and Engineering) is a record of the original work carried out by him under my supervision.

Project Guide
Prof. Rajni Jindal
H.O.D.
Department of Computer Science & Engineering
Delhi Technological University

DECLARATION

I hereby declare that the Major Project-II work entitled **"Survey on Dual Implementation in Sentiment Analysis"** which is being submitted to Delhi Technological University, in partial fulfilment of requirements for the award of the degree of Master of Technology (Software Engineering) is a bonafide report of Major Project-II carried out by me. I have not submitted the matter embodied in this dissertation for the award of any other Degree or Diploma.

Diwakar Kumar
2K18/SWE/21

ACKNOWLEDGEMENT

First of all, I would like to express my deep sense of respect and gratitude to my project supervisor Prof Rajni Jindal (HOD), Computer Science & Engineering Department for providing the opportunity of carrying out this project and being the guiding force behind this work. I am deeply indebted to him for the support, advice and encouragement he provided without which the project could not have been a success.

I would also like to acknowledge Delhi Technological University library and staff for providing the right academic resources and environment for this work to be carried out.

Last but not the least I would like to express sincere gratitude to my parents and friends for constantly encouraging me during the completion of work.

Diwakar Kumar
Roll No – 2k18/SWE/21
M. Tech (Software & Engineering)

Delhi Technological University

ABSTRACT

Sentiment Analysis is one of the trending research topics on which people are working widely to enhance its effectiveness and availability. But still proposed methodologies are lagging when it comes to accuracy and the size of dataset. With the ongoing evolutions in the field of technology here I tried to emphasize the role of dual implementation in Sentiment Analysis Here we are working on the same to enhance the accuracy of the result set.

Also, here we included the different domains in which sentiment analysis are implemented and there comparision with other different applications and the challenges faced in there which involves the data related, accuracy issues related challenges and other similar issues. We are also discussing about the methodologies they are using in their paper. But here we are mainly focused on improving the accuracy by applying multiple implementations on training data set like adding multiple attributes based on the type of issue, training same data set with different methodologies and by data manipulation.

We are also comparing behavior of results of different cases using different classification methods so that we get a better understanding on type of issues and in return, accuracy on result set gets better.

TABLE OF CONTENTS

List of Figures

List of Tables

List of Abbreviations

1.	SA:	Sentiment Analysis
2.	SL:	Sentiment Label
3.	CM:	Classification methods
4.	SVM:	Support Vector Machine
5.	CNN:	Convolution Neural Network
6.	MLP:	Multilayer Perceptron
7.	KNN:	K- Nearest Neighbor
8.	TF:	Term Frequency
9.	IDF:	Inverse Document Frequency
10.	TF-IDF:	Term Frequency – Inverse Document Frequency
11.	CM:	Confusion Matrix

CHAPTER 1: INTRODUCTION

As we all know that in the field of SA there are lots of research work was introduced with different perspectives and goal. But when we are coming to the training data and prediction on the basis of training data set it is still lagging in terms of accuracy to assign correct classification label to some data set. Before we proceed for our problem statement first we need to understand some basic techniques which we are going to use here.

1.1 Simple Baseline Technique

For the baseline model, n-gram based language model is used. We have trained and tested the model for all unigram, bigram and trigram.

1.1.1 Preprocessing

- In preprocessing, firstly Data points in which NaN or null values are are present those data point are removed.

- Then stop words are removed and as per the usage of model lemmatization and stemming are used for preprocessing.

- Stemming: Stemming algorithms work by cutting off the end or the beginning of the word, taking into account a list of common prefixes and suffixes that can be found in an inflected word. Used porter stemmer for implementing it in our code.

- Lemmatization: Lemmatization, on the other hand, takes into consideration the morpho-logical analysis of the words. To do so, it is necessary to have detailed dictionaries which the algorithm can look through to link the form back to its lemma. Used W or net lemmatization for performing this lemmatizing task.

  Handling Multi-labeled Data: The given data is multi-labeled i.e., a single comment may belong to multiple classes at an instant. For making it single labeled the class labels are merged. For example, if a comment belongs to both toxic and sever toxic class then the new class label will be toxic sever toxic.

- Removing Garbage Data: The data given in test.csv there are some comments which have - 1 as entry for all labels. This indicates that the data was never labeled and this data is not to be used for evaluating our model. That’s why we drop such data points.

1.1.2 n-Gram Based Language Model

- Unigram: In this part the language model is trained and tested with the usage of unigrams. The complete data including all the comments are divided into the unigrams for a particular class label. Then the probability measures are used to classify the comments into the given labels.

Accuracy = 54%
Recall = 55%
Accuracy and F1 score = 65%

- **Bigram**: In this part the language model is trained and tested with the usage of bigrams. The occur frequency measures and probability measures are used to generate a language model using bigrams. This measures are used to classify the comments into the given class labels.

Accuracy = 00%

- **Trigram**: In this part the language model is trained and tested with the usage of trigrams. The term frequency for each trigram is measured and probability for each trigram is also measured. These measures are used to generate a language model using trigrams and classify the comments into the given class labels.

Accuracy = 00%

1.1.3 Word 2 Vec Model

For both the inquiries given in preparing set, word2vec model is being prepared which is a se - mantic learning system that utilizes a neural organization to get familiar with the portrayals of words/phrases in a specific book. The 'advantage' word2vec offers is in its use of a neural model in understanding the semantic significance behind those terms. More often than not it utilizes cosine closeness to measure the comparability. Genism apparatus of python is being utilized with word2vec which extricate semantic themes from records naturally in the most productive and easy way. For preparing word2vec model , size=200, window=10, min-count=2, sg=1, workers=10 boundaries are taken.

- Size: It characterizes the components of neural organization that is, size to speak to every token or word.

- Window: The size of the window decides the number of words when a given word would be incorporated as setting expressions of the given word.

- Min tally: Terms that happen not as much as min-check number of times are overlooked in estimations.

- sg: It determines which model is utilized for preparing constant Bag of Words model or skip gram model.

- Workers: The quantity of laborer strings used to prepare the model.

After this, on 20% of preparing dataset, Word Move Distance work is utilized to figure closeness between two inquiries. It can cover equivalent word issue. Exactness, accuracy and re - call are determined by contrasting and truth marks given in preparing dataset.

Accuracy = half

Review = 30%

Exactness and F1 score = 35%

As should be obvious this model isn't awesome.

1.2 Feature Extraction For Advance Model

Extraction of feature is the way toward getting valuable highlights from the crude information given which can encourage the ensuing learning and speculation steps, and sometimes prompts better human understandings. For current methodology, Different sort of highlights are inferred for each question pair. Highlights depend on similitudes and semantics be-tween the pair of inquiries. Extricated highlights are as per the following:

1.2.3 Google Pre-Trained Word 2 Vec Model

It contains word vectors for a jargon of 3 million words prepared on around 100 billion words from the google news dataset. In current methodology, word embeddings are being shaped for questions sets utilizing google pretrained word2vec model. For making word embeddings, two procedures have been utilized: TF-idf based word embeddings and basic normal of word vectors and framed two highlights by computing cosine comparability.

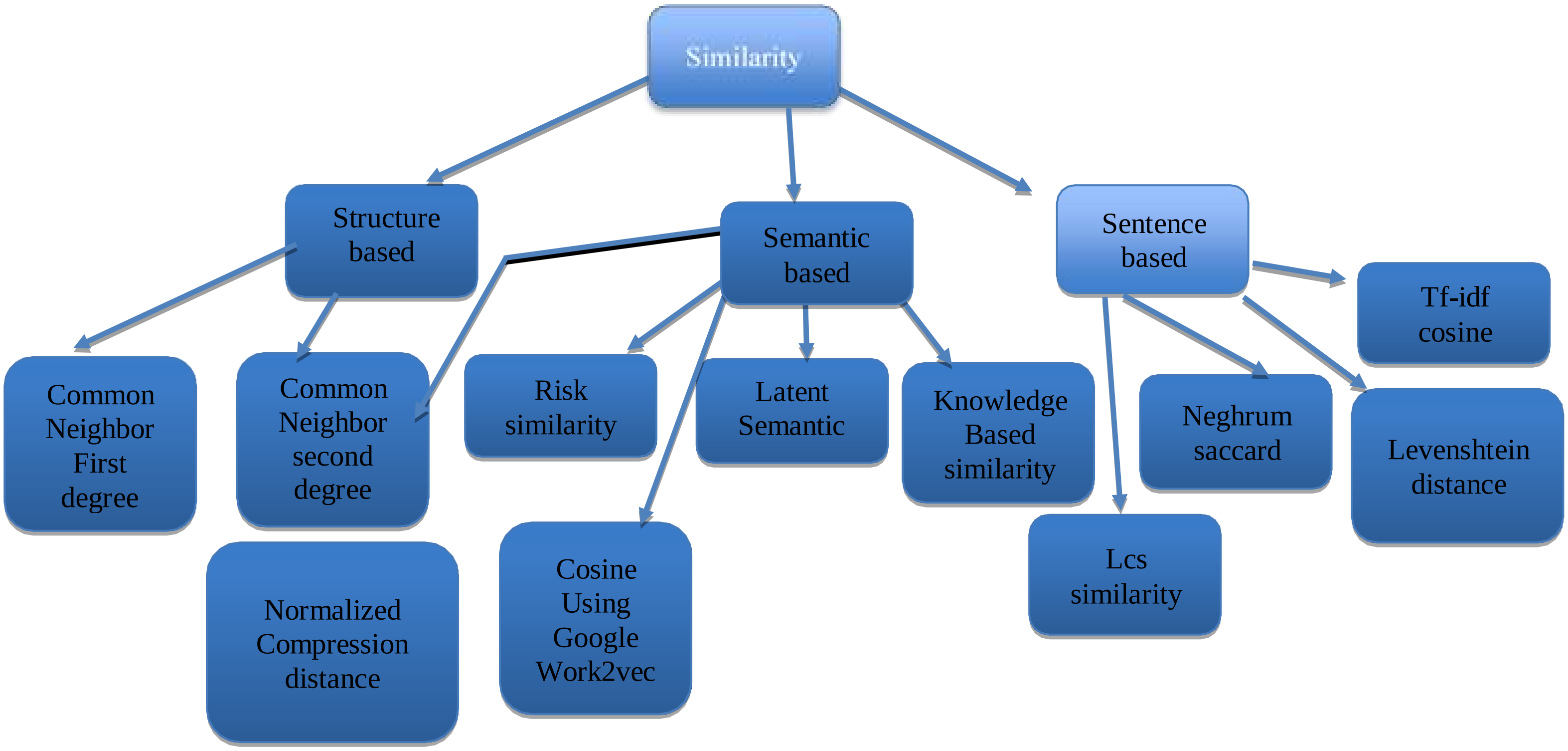


Figure 1: Different kind of similarities

TF-IDF based cosine likeness:- TF-IDF is short for term recurrence converse record recurrence. It is a mathematical measurement that is proposed to reflect how significant a word is to a report in an assortment or corpus. TF(Term Frequency) is taken as a proportion between basic include of that specific term in that specific inquiry. IDF(Inverse Document Frequency) is taken as the log of the proportion between all out number of inquiries and the quantity of

inquiries in which that term shows up. For every one of the inquiry pair, Word vectors are processed utilizing google pretrained model, at that point take weighted normal by increasing each word with its particular tf-idf and store it as highlight for the model. It follows Bag of Words model The recipe utilized for estimation of tfidf is:

t df(t; d; D) = tf(t; d) idf(t; D):

D: absolute number of records in the corpus

tf (t,d): term t recurrence in archive d

idf (t,D): converse record recurrence of term t in corpus D

A high weight in tf–idf is reached by a high term recurrence (in the given record) and a low report recurrence of the term in the entire assortment of archive.

Average cosine comparability: In this technique for each question, word embeddings are shaped for each word. Cosine similitude is registered by computing normal of word embeddings.

Normal words rate: It is determined by isolating no of basic of words in two inquiries by all out no of words present in two inquiries.

Length contrast rate: It is determined by isolating distinction of length between two inquiries by all out length of two inquiries.

Cosine comparability utilizing Word embeddings from self prepared word2vec: In this component extraction procedures, questions jargon is being utilized to shape word embeddings for each word present in inquiries rather than google pretrained model. At that point cosine likeness is being determined by taking weighted tf-idf normal of word embeddings for two inquiries.

Longest normal aftereffect (LCS): LCS coordinating is a usually utilized procedure to mea - sure the similitude between two strings (I, j). LCS measure the longest complete length of the apparent multitude of coordinated substrings between two strings where these sub-strings show up in a similar request as they show up in the other string. LCS comparability of Given Two string (I, j) will be.

$$\text{LCS}(I,j) = \begin{cases} 0 & \text{if } i=0 \text{ or } j=0 \\ 1+\text{LCS}(i-1,\ j-1) \text{ if } x[i]==y[j] & \\ \text{Max} & \begin{cases} \text{LCS}(I,j-1) \\ \text{LCS}(i-1,\ j) \text{ if } x[i]+y[j] \end{cases} \end{cases}$$

In current methodology, LCS for two inquiries is discover and standardized by partitioning all out no of aftereffects.

Leven shtein separation likeness: The Leven-shtein separation procedure additionally utilizes the separation factor to figure the similitude between two given strings. In real, this separation is tallying the base number of tasks expected to change one string into other string. The Leven - shtein separation between two string a, b is given by

$$\text{Lev a, b}(I,j) = \begin{cases} \text{Max}(i,j) \text{ if } \min(i,j)=0 \\ \begin{cases} \text{lev a,b } (I,\ j-1)+1 & \text{otherwise} \\ \text{Lev a,b } (i-1\ ,j-1)+1 \end{cases} \end{cases}$$

CHAPTER 2: RELATED WORK

There are different research papers published on the applications of sentiment analysis, we are doing a survey on few of them on the basis of different parameters. We are not going to write detailed explanation about the topic, we"ll just comparing the methodologies used in research papers and on the basis of some other parameters.

Before going ahead we"ll be discussing the topics of papers we are going to consider for the survey:

- **NADAQ: Natural Language Database Querying Based on Deep Learning**
- **Sentiment Analysis in A Cross-Media Analysis Framework**
- **Tweet Sentiment Analysis by Incorporating Sentiment-Specific Word Embedding and Weighted Text Features**
- **Entity-Level Sentiment Analysis of Issue Comments**
- **Dual Sentiment Analysis: Considering Two Sides of One Review**

The following are the outcomes of few papers referred for analysis purpose on their accuracy. We found out that most of them are working on methodologies part and in result could not be able to generate better results. So, here in our thesis we are more concerned towards the training dataset and accuracy of final result set through various means which is described better in methodologies part.

Table I Comparison of Accuracy of different Model Proposed

Ref. No.	SA Challeng e type	SA Challenge	Technique Used	Lexicon Type	Data Set	Accuracy
BOYAN XU1 Et. Al. (2019)	Technical	Complex Syntax	NADAQ System	SQL	Kaggle data for complex SQL Querry	>80%
Andrius Et. Al. (2012)	Technical	Huge lexicon	Bag-of-word11sSVM	pSenti	The firrst data set SoftwareReview,sec ond dataset MovieReviews	82.30%
Yonas Woldemariam m(2017)	Technical	SA Pipeline	LEXICON-BASED SENTIMENT CLASSIFICATION USING HADOOP	MICO text	MICO Dataset	72.3%
Quanzhi Li Et.Al. (2016)	Theoritical	Incorrect words or Sentence	SSWE, WTFM & Rocchio text classification	Comments	Twitter comments	62.4
Erikand MarieFrancin e (2009)	Technical	Nlp overheads (Multilingu al)	Integrated approach combining from information retrieval, natural language processing and machine Learning	English,Dutch andFrenchte x	Blog, review and forum texts found on the WorldWide Web	83% 70% and 68%

Yonas Woldemariam (2016)	Technical	Lexicon Based Sentiment Prediction	Recursive Neural Tensor Network (RNTN) model.	Amazon Turk	Tree bank Phrases Labeled with Amazon turk	Improved by 9.88%
Alexandra etal.(2013)	Theoretic al	Domain Dependenc e	WordNet-lexiconbased	Newsreviews	News paper articles (the set of 1292 quotes)	82%improve the baseline 21%

CHAPTER 3: SA TECHNIQUE AND DIFFERENT CLASSIFICATION MODEL

3.1 Basic Understanding of SA techniques

Sentiment analysis is field of study that investigation of individuals assessment, estimations, disposition and emotions towards entities for example items, services organizations individual events issues, subjects and their attributes.

Levels In SA

Different 3 levels in sentiment analysis that is document level, sentence level and facet level. In document level i.e. known that's the review is positive or negative. In sentence level i.e., known each sentence is positive or negative and in facet level entities and their features/aspects Sentiments is positive and negative.

Document level :

In Document level analysis task is characterize whether or not a whole opinion of document level communicates a positive or negative supposition for example, given issue audit, the framework figures out if the survey communicates a general positive or negative call concerning something. This enterprise is frequently called document level sentiment classification.

Sentence Level :

In Sentence level the basic enterprise is goes to the Sentence and is smart of if each sentence communicated a positive, negative, or neutral sentiment. Neutral means that no opinion concerning any sentence. This level of investigation is immovably associated with the judgement arrangement. that is acknowledges sentences (called target sentences) that's specific real data from the sentences (called subjective sentences) that specific subjective views and opinions.in any case, we have a tendency to have to be compelled to observe that judgement isn't admire supposition a similar range of target sentences will counsel feelings for e.g., “We purchased new car a month past and also the mechanical device has tumbled off’.

Aspect Level :

In facet Level each the document level and also the sentence level analyses don't discover what precisely individuals likable and didn‟t like. facet level performs better-grained investigation. facet level is directly appearance at the opinion itself. within the facet level is rely on the chance that associate opinion consists of a sentiment positive, negative or neutral or associate objective of sentiment.

For e.g. Sentence is "The Sony telephone's decision quality is wonderful, however its battery life is short" assesses 2 focuses 1st is decision quality second is battery life, of Sony (component). The conclusion on Sony's decision quality is for certain in sentence but the opinion on its battery life is negative. Sony telephone's decision quality and battery lifetime of Phone area unit the sensation targets. during this level of investigation, associate organized of assessments concerning components and their viewpoints will be created, that turns unstructured content to organized.

Sentiment Analysis Technique :

It is broadly divided into two separate technologies which is Machine Learning and Lexicon

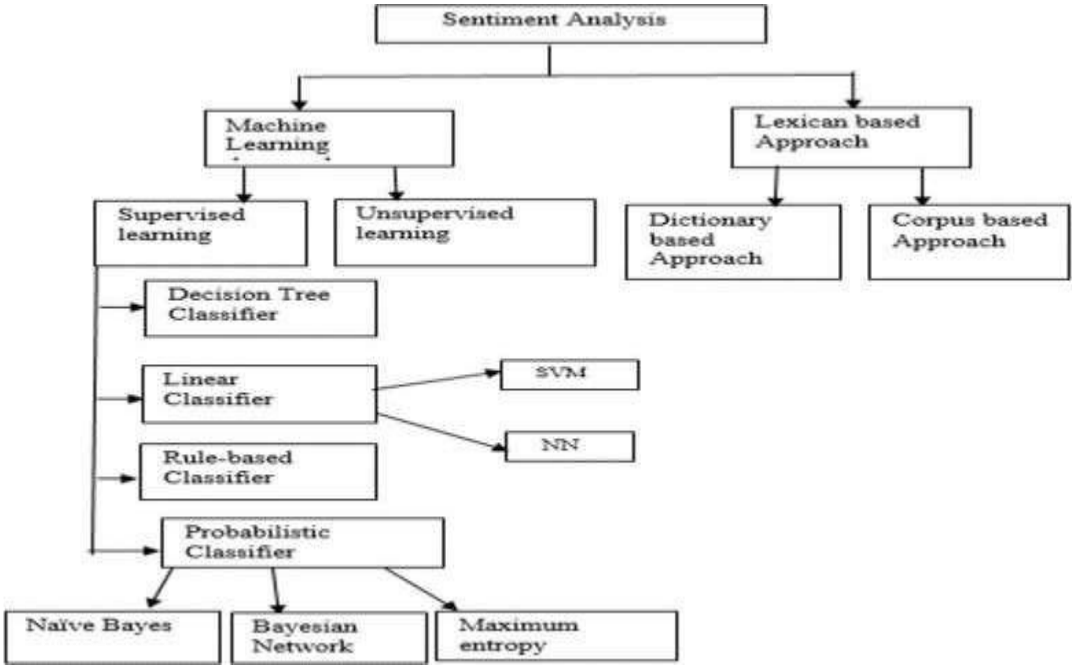


Figure 2 : Flow Chart of Sentiment Analysis Technique

Based Learning.

Here in our thesis we are mainly focused on **Machine learning based approach** (Machine learning approach is depends on upon cubic centimeter algorithms to unravel the Sentiment Analysis as a regular substance classification issue that creates use of syntactical yet as linguistic options.)

3.2 Classification Model

Features extracted by the methodologies we"ll read in the below section are used to train different classification models on training data such as Naive Bayes Classification, K-nearest Neighbors, Support Vector Machine and Convolutional Neural Network. So what we are about to do with these classification method is to we"ll partition the complete data into two set one is for training purpose and one is for testing purpose in the ratio of 70:30 respectively and we are going to perform different models on them and check how they vary with different classification models and we are going to add our classification methods as well with the training and testing data in order to compensate with accuracy and to get more better results. So, now we"ll discussing about all of them in this chapter.

3.2.1 Naive's Bayes Classification

Naïve Bayes is an order calculation for paired (two-class) and multi-class arrangement issues. The procedure is simplest to comprehend when portrayed utilizing parallel or all out info esteems.

Gullible bayes classifier is a probabilistic model that is utilized for the order task, the core of classifier depends on the Bayes hypothesis :

$$P(A\backslash B)=(P(B\backslash A)P(A))/P(A)$$

Utilizing the bayes hypothesis, we can assess the likelihood of class given some archive by extending he bayes hypothesis :

$P(y \mid D) = P(D \mid y) * P(y)/P(D)$, "y" is a sure class mark worth, and "D" is the record whose class name we need to discover.

The above likelihood can be extended:

$P(y \mid X1, X2 ,\dots. Xn) = P(X1, X2, \dots .. Xn \mid y) * P(y)/P(X1, X2, \dots .. Xn)$

X1, X2, … .. , Xn are the badge of the archive D.

$P (y \mid X1, X2 , \dots.Xn)$ is the back likelihood.

$P(X1, X2, \quad Xn \mid name)$ is the probability likelihood.

$P(y)$ is the earlier likelihood.

If there should be an occurrence of Naive bayes, it utilizes the autonomy suspicion i.e given the class mark the likelihood of a tokens are autonomous of one another which implies that

P(token1 | name) is autonomous from P(token2 | mark) which is obviously an off-base supposition. Now and again on the off chance that the freedom suspicion holds, at that point Naive bayes perform well, and in the event that the autonomy supposition doesn't hold, at that point Naive bayes perform more regrettable. After the utilization of autonomy presumption:

$P(y\backslash x1,\dots ..,xn)=p(x1|y)p(x2|y)..p(xn|y)p(y)$

$P(x1)p(x2)\dots p(xn)$

The likelihood of the archive in the denominator is the likelihood which is autonomous from the class mark which will stay same for all the class names regarding that specific record, so we can disregard this likelihood, so the last back likelihood is :

$P(y|x1, \quad ,xn) \infty p(y) \prod n,i=1 \; P(XI|Y)$

So we will locate the back likelihood for all the class names regarding specific report and the class mark which is having most extreme back likelihood is alloted as the anticipated class name for that archive.

$Y=argmaxy\prod n,n=1\ p(xi|y)$

3.2.2 K-Nearest Neighbor Classification

K Nearest neighbor grouping which is abridged as 'KNN' is one of the least difficult order models which depends on the likeness between the records. So as to run any characterization model, we need two datasets: the train dataset, and the test dataset. The train dataset is utilized to prepare our model with the goal that it can anticipate the class names for the new concealed information. We will discover the class marks of the test dataset utilizing the model which is prepared on the train dataset. On account of KNN, for the test archive for which we need to discover the class name, we will discover the closeness between the train reports and the test record. There are different likeness or separation estimates which can be utilized to discover the closeness between two archives:

- Euclidean Distance
- Manhattan Distance
- Minkowski Distance
- Cosine Similarity

So by utilizing any of the similitude measures between the archives which are indicated above, we can discover the closeness between the test record and train reports, and we will choose "k" train records which are having higher likeness with the test archive.

Presently for these "k" train records we realize their class marks as of now, so we will apply some sort of surveying method i.e the class name which happens the greater part of the occasions in those "k" archives will be doled out as the anticipated class name for the test report.

Advantages:

- The calculation is basic and simple to actualize.
- There's no compelling reason to fabricate a model, tune a few boundaries, or make extra suppositions.
- The calculation is adaptable. It very well may be utilized for characterization, relapse, and search (as we will find in the following segment).

Disadvantages : The calculation gets altogether more slow as the quantity of models and additionally indicators/free factors increment.

3.2.3 Decision Tree Classification

A tree has numerous analogies, in actuality, and turns out that it has affected a wide zone of AI, covering both characterization and relapse. In choice examination, a choice tree can be utilized to outwardly and expressly speak to choices and dynamic. As the name goes, it utilizes a tree-like model of choices. Despite the fact that a generally utilized device in information digging for determining a technique to arrive at a specific objective, its additionally broadly utilized in AI, which will be the fundamental focal point of this article.

For this current we should consider a fundamental model that utilizes titanic informational collection for foreseeing if a traveler will endure. Beneath model uses 3 highlights/ascribes/sections from the informational collection, to be specific sex, age and sibsp (number of companions or youngsters along).

Albeit, a genuine dataset will have significantly more highlights and this will simply be a branch in an a lot greater tree, yet you can't overlook the effortlessness of this calculation. The element significance is clear and relations can be seen without any problem. This strategy is all the more usually known as taking in choice tree from information or more tree is called Classification tree as the objective is to group traveler as endure or passed on. Relapse trees are spoken to in a similar way, just they foresee consistent qualities like cost of a house. As a rule, Decision Tree calculations are alluded to as CART or Classification and Regression Trees.

Anyway, what is really going on out of sight? Growing a tree includes settling on which highlights to pick and what conditions to use for parting, alongside realizing when to stop. As a tree for the most part develops subjectively, you should manage it down for it to look lovely. Lets start with a typical procedure utilized for parting.

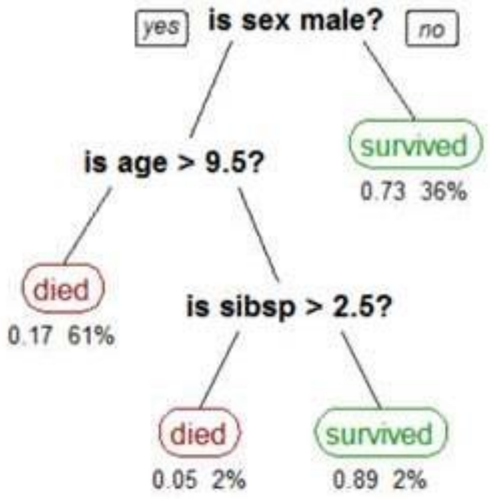


Figure 3 : Basic Decision Tree Example

When to quit parting?

You may request that when quit growing a tree? As an issue for the most part has a huge arrangement of highlights, it brings about enormous number of split, which thus gives a tremendous tree. Such trees are unpredictable and can prompt overfitting. Anyway, we have to realize when to stop? One method of doing this is to set a base number of preparing contributions to use on each leaf. For instance we can utilize at least 10 travelers to arrive at a choice (kicked the bucket or endure), and disregard any leaf that takes under 10 travelers. Another route is to set greatest profundity of your model. Most extreme profundity alludes to the length of the longest way from a root to a leaf.

3.2.4 Support Vector Machine(SVM) Classification

SVM is supervised learning model. Support Vector Machines work on plan of call planes that specify call boundaries. Set of objects happiness to varied category memberships are participations by call planes. Shown in Fig. example for example the conception of linear SVMs within the objects either belong to inexperienced category (or RED class) during this example.

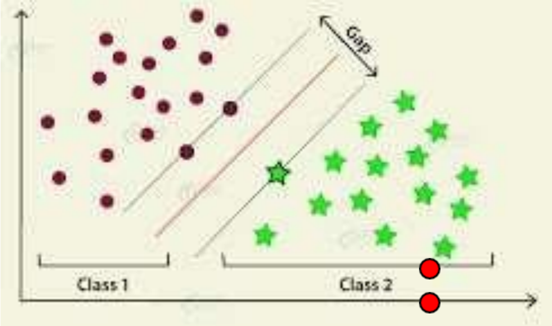


Figure 4 : Example of Linear SVM

Isolated line confirm the selection. On the proper hand facet of the boundary, all objects are inexperienced and to the facet hand facet of boundary, all articles are RED. a brand new object white circle are going to be classified as inexperienced if it falls to the proper hand facet of the boundary or classified as RED if it falls to the one facet of the boundary.

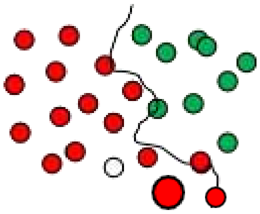


Figure 5 : Example of Hyperplane SVM

A classifier partitions a set of objects into their respective domains with a line is called linear classifier and partitioning with a curve is known as hyperplane classifier. An example of hyperplane classifier is shown in figure.

3.2.5 Multilayer Perceptron Model

A multilayer perceptron (MLP) is a profound, counterfeit neural organization. It is made out of more than one perceptron. They are made out of an information layer to get the sign, a yield layer that settles on a choice or forecast about the information, and in the middle of those two, a discretionary number of concealed layers that are the genuine computational motor of the MLP. MLPs with one shrouded layer are equipped for approximating any ceaseless capacity.

3.2.6 Convolution Neural Network

CNNs, as neural organizations, are comprised of neurons with learnable loads and inclinations. Each neuron gets a few sources of info, takes a weighted total over them, go it through an enactment

work and reacts with a yield. In contrast to neural organizations, where the info is a vector, here the information is a multi-directed picture (3 diverted for this situation).

• CNN is prepared with one information layer, two covered up lair layers and one yield layer to anticipate the yield

• Hyperparameter tuning is being done to have the boundaries which can give better precision Different boundaries utilized are as following:

• epochs: The quantity of ages is a hyperparameter that characterizes the number occasions that the learning calculation will work through the whole preparing dataset. In current methodology, ages are taken as 50.

• batch size: The cluster size characterizes the quantity of tests that will be engendered through the organization.

• Optimizer: Optimization calculations encourages us to limit (or amplify) an Objective capacity (another name for Error work) E(x) which is essentially a numerical capacity reliant on the Model's inner learn-capable boundaries which are utilized in registering the objective values(Y) from the arrangement of predictors(X) utilized in the model. In current methodology, Adam streamlining agent is utilized.

learn rate: Learning rate is a hyper-boundary that controls the amount we are changing the loads of our organization with deference the misfortune inclination. The lower the worth, the more slow we travel along the descending slant. In current methodology, learning rate is 0.04.

• Neuron Activation work: Activation work chooses, if a neuron ought to be initiated by ascertaining weighted total and further including predisposition with it. The reason for the enactment work is to bring non-linearity into the yield of a neuron. In current methodology, sigmoid is utilized for shrouded layers and delicate max for yield layer.

CHAPTER 4: PROPOSED WORK

Here let's discuss about the content we are working on and what we have introduced here. So here we are working to improve the accuracy of classification methods using dual implementation or we can say by adding multiple attributes and functionality in addition with the classification model. So here I have worked on two major topics in order to improve it's accuracy out of which one is IMDb rating prediction and other is toxic comment classification. So, we'll be discussing about these topics in detail and their problem statement and solution proposed.

4.1 Problem Statement

Since here we are working on multiple problem statement, so we'll be discussing about both one by one and their requirement background and all other related stuff. Coming to the first problem is about classifying the given comments into multiple toxic comment class label. Each label will define the level of toxicity and i.e. how toxic the comment is which like ordinary, humiliating, insulting, mild, semi severe, severe etc. so we have some dataset which is already described with their result set so we had partitioned the dataset into two parts one of them is training dataset which will be given to the classification models and other is testing data set from which result attribute or we can sentiment label which is the outcome is omitted and used to compare while testing with resulted and actual sentiment label. Now coming to second issue it is also similar to the previous one and the only difference here is the topic, sentiment labels, the set of data and attributes.

4.2 Problem Background

The set of comments and movie names and their result set is taken from the Kaggle. Dataset named toxic comment classification and Rating prediction IMDb. The provided dataset for this problem contains 3 different files. Let's first discuss about toxic comment issue, so, the files are train.csv. It has 160k rows in which each row represents the class labels for each comment. Each data point has seven columns id, toxic, sever toxic, obscene, threat, insult, identity hate. The second file is, test.csv. It has only 2 columns in it i.e., id and comments. We are supposed to predict the label for these comments which are stored in this file. For the true value of these class labels and accuracy measure, the labels are given in another file named test labels.csv. It has a column corresponding to every class label. Each column has only 2 possible values 0 or 1 in train data. 0 means that comment does not belong to the class and 1 means comment belongs to that class. In test label -1 also a possible value which states that the corresponding comment is not used for labelling. And coming to the second problem similar to the above we'll be getting three different files one is train.csv which is of size nearly 5k and the second file is result.csv which contains the result set which is used to compare the calculated result and the actual result and the last file is the file where the output of the classification methods will be there.

4.3 Discussion

In this whole scenario we are trying to obtain the sentiments of the user and trying to build the system which can identify the toxicity of the comments or prediction of rating as per the audience and considering there different thoughts and considering all relative points which can affect these. Here we are identifying the severity of comments based on different labels like toxic, severe toxic, hate, insult etc.

We have already got a data set of nearly one lack fifty four thousand comments which is in the tabular form with certain labels and results. These data set will be used accordingly:

i) In this we are using 70 % of the data as a training set data which will be used with the different classification methods to check their efficiency respectively.

ii) Remaining 30 % of data will be used as a testing data set which will be demonstrate the efficiency of classification methods and our data featuring and the techniques that we are implementing.

Moving further we are doing modification with the data to make it more efficient and and in order to improve the efficiency of the system. This method is also called **Data Featuring.** Here we are adding some more labels like no. of characters, capital letters, proportions etc.

Multilabel vs Multiclass classification?

As the task was to figure out whether the data belongs to zero, one, or more than one categories out of all the labels listed in the model, the first step before working on the problem was to distinguish between multi-label and multi-class classification.

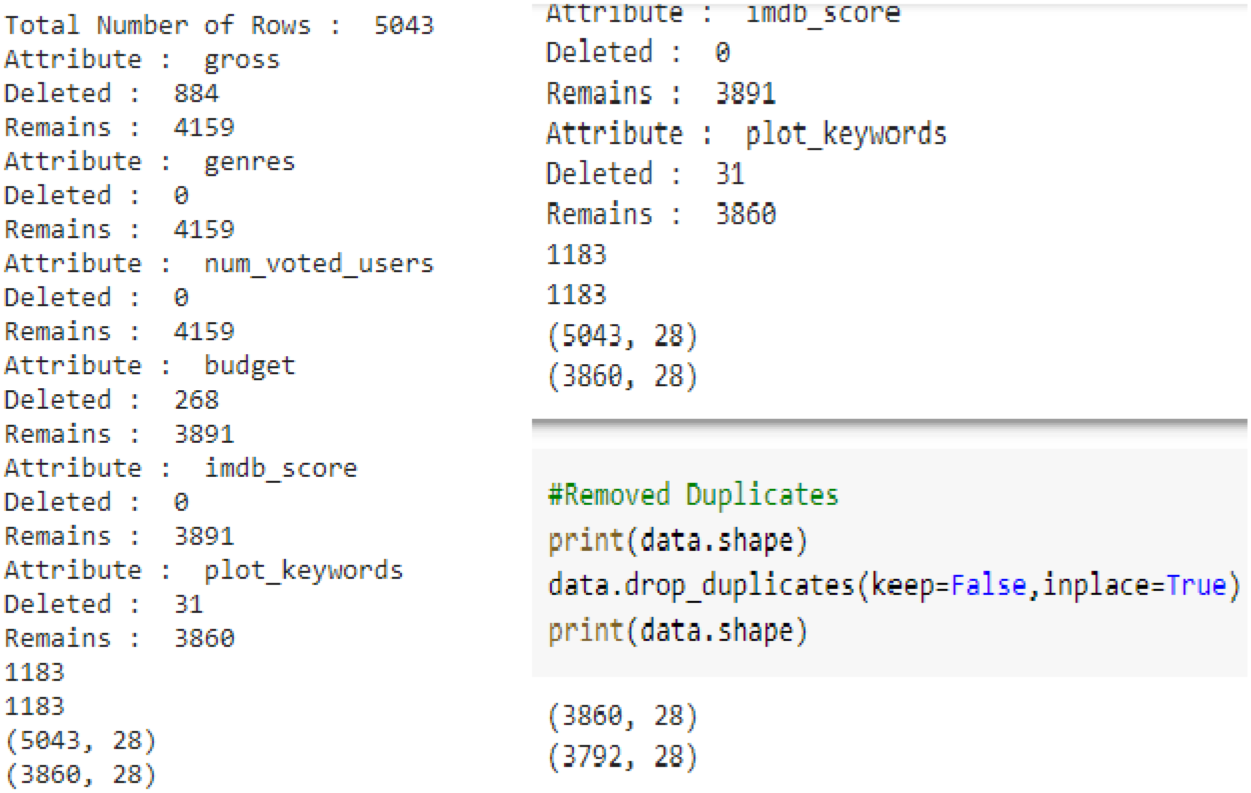
In multi-class classification, we have one basic assumption that our data can belong to only one label out of all the labels we have. For example, a given picture of a fruit may be an apple, orange or guava only and not a combination of these.

In multi-label classification, data can belong to more than one label simultaneously. For example, in our case a comment may be toxic, obscene and insulting at the same time. It may also happen that the comment is non-toxic and hence does not belong to any of the six labels. Hence, I had a multi-label classification problem to solve.

4.4 Problem Evaluation

Coming to problem evaluation we'll be discussing about the majority steps taken in the mean while of processing the data from cleansing to testing or we can say processing the file, we are working on it from the very beginning phase to the ending which are as follows :

- Very first step is data cleansing where we run the data through a process which cleans it means, since we are working on very large data set so we need to clean data before we proceed for next step so what we are doing here is like and described in following figures:
    1. Removing the Null Values.
    2. Deleting duplicate values.
    3. Clearing value



```
Total Number of Rows :  5043
Attribute :  gross
Deleted :  884
Remains :  4159
Attribute :  genres
Deleted :  0
Remains :  4159
Attribute :  num_voted_users
Deleted :  0
Remains :  4159
Attribute :  budget
Deleted :  268
Remains :  3891
Attribute :  imdb_score
Deleted :  0
Remains :  3891
Attribute :  plot_keywords
Deleted :  31
Remains :  3860
1183
1183
(5043, 28)
(3860, 28)
```

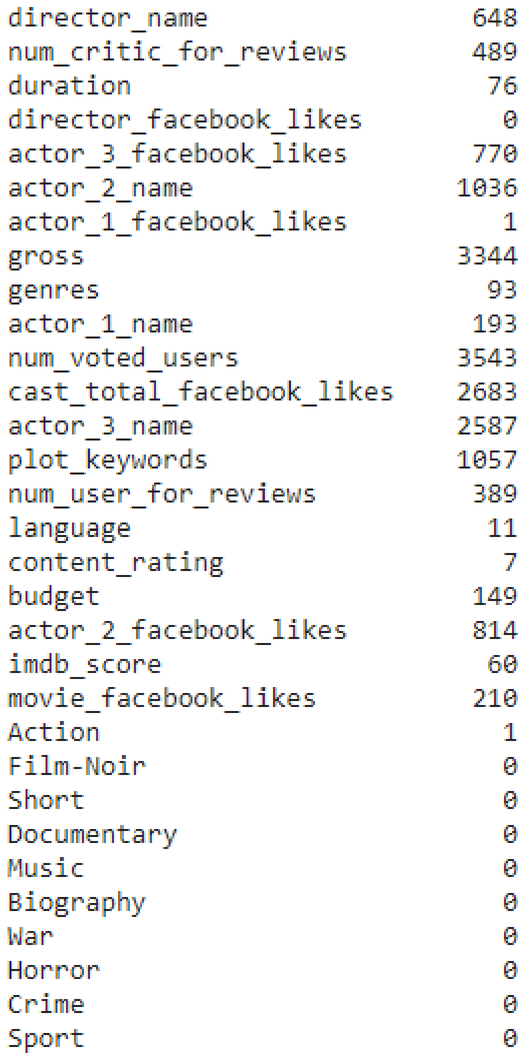
```
Attribute :  imdb_score
Deleted :  0
Remains :  3891
Attribute :  plot_keywords
Deleted :  31
Remains :  3860
1183
1183
(5043, 28)
(3860, 28)
```

```
#Removed Duplicates
print(data.shape)
data.drop_duplicates(keep=False,inplace=True)
print(data.shape)
```

```
(3860, 28)
(3792, 28)
```

Figure 6 : Output of deleted Records

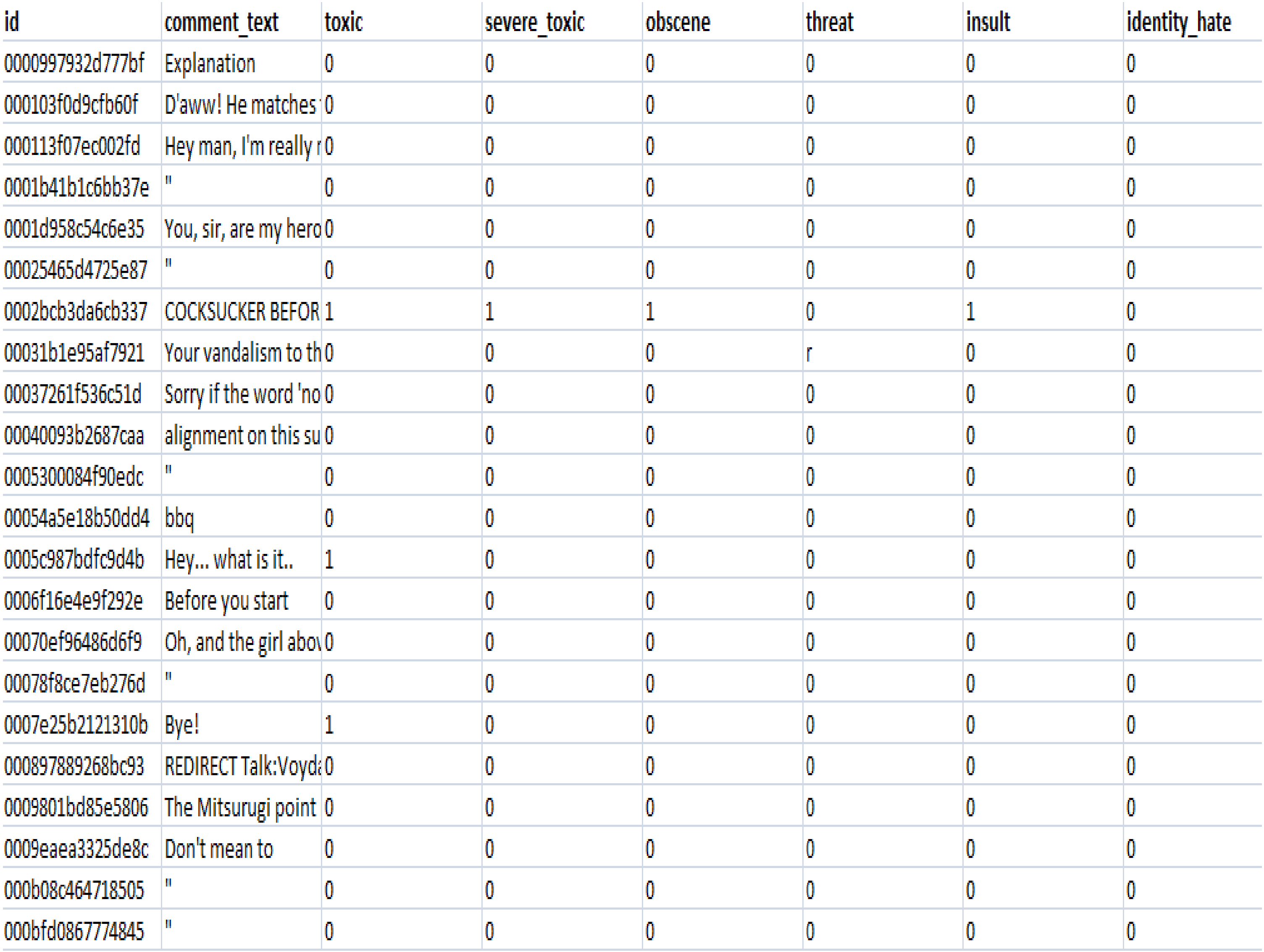
- Moving to the next step we are adding features or attributes to the dataset in order to make decisions more accurately based on different parameters classification methods can work more accurately and can give more precise results for e.g. a toxic comment is usually shorter in length so it can help better in this way … so we have added a list of attributes in order to improve the accuracy.



```
director_name                 648
num_critic_for_reviews        489
duration                       76
director_facebook_likes         0
actor_3_facebook_likes        770
actor_2_name                 1036
actor_1_facebook_likes          1
gross                        3344
genres                         93
actor_1_name                  193
num_voted_users              3543
cast_total_facebook_likes    2683
actor_3_name                 2587
plot_keywords                1057
num_user_for_reviews          389
language                       11
content_rating                  7
budget                        149
actor_2_facebook_likes        814
imdb_score                     60
movie_facebook_likes          210
Action                          1
Film-Noir                       0
Short                           0
Documentary                     0
Music                           0
Biography                       0
War                             0
Horror                          0
Crime                           0
Sport                           0
```

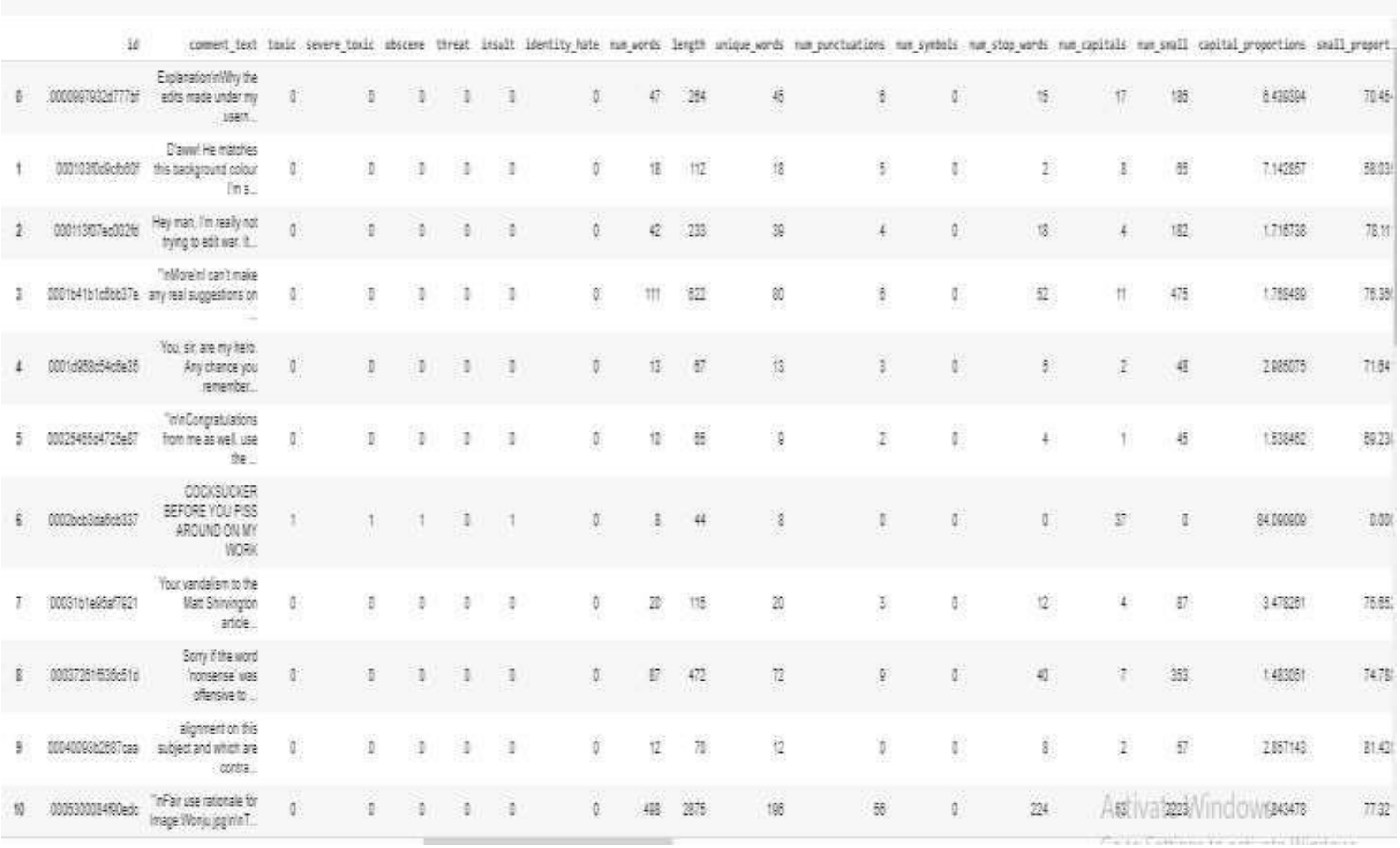
Figure 7 : List of Attributes added

- And then we worked to enhance the sentiment label(SL) or we can say polarity. In the field of sentiment analysis, the main issue is to find **polarity** of a sentence whether it is positive, negative or neutral. Coming to positive or negative still these two somehow seems feasible but the most challenging task is to find the neutrality of any comment in the field of SA, so moving ahead previously we were having only 6 SL, due to which a particular comment can belong to more than one SL which then creates a problem to the classification model and hence reduces the accuracy. So in order to improve the efficiency and accuracy of the system we have increased the no of labels to 52 after which we found out that now a single comment can belong to a single SL which are described in the below image.



id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0000997932d777bf	Explanation	0	0	0	0	0	0
000103f0d9cfb60f	D'aww! He matches	0	0	0	0	0	0
000113f07ec002fd	Hey man, I'm really r	0	0	0	0	0	0
0001b41b1c6bb37e	"	0	0	0	0	0	0
0001d958c54c6e35	You, sir, are my hero	0	0	0	0	0	0
00025465d4725e87	"	0	0	0	0	0	0
0002bcb3da6cb337	COCKSUCKER BEFOR	1	1	1	0	1	0
00031b1e95af7921	Your vandalism to th	0	0	0	r	0	0
00037261f536c51d	Sorry if the word 'no	0	0	0	0	0	0
00040093b2687caa	alignment on this su	0	0	0	0	0	0
0005300084f90edc	"	0	0	0	0	0	0
00054a5e18b50dd4	bbq	0	0	0	0	0	0
0005c987bdfc9d4b	Hey... what is it..	1	0	0	0	0	0
0006f16e4e9f292e	Before you start	0	0	0	0	0	0
00070ef96486d6f9	Oh, and the girl abo	0	0	0	0	0	0
00078f8ce7eb276d	"	0	0	0	0	0	0
0007e25b2121310b	Bye!	1	0	0	0	0	0
000897889268bc93	REDIRECT Talk:Voyd	0	0	0	0	0	0
0009801bd85e5806	The Mitsurugi point	0	0	0	0	0	0
0009eaea3325de8c	Don't mean to	0	0	0	0	0	0
000b08c464718505	"	0	0	0	0	0	0
000bfd0867774845	"	0	0	0	0	0	0

Figure 8 : Original Dataset with Sentiment Label



	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate	num_words	length	unique_words	num_punctuations	num_symbols	num_stop_words	num_capitals	num_small	capital_proportions	small_proport...
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0	47	264	45	6	0	15	17	186	6.439394	70.45
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0	18	112	18	5	0	2	8	65	7.142857	58.03
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0	42	233	39	4	0	18	4	182	1.716738	78.11
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0	111	622	80	6	0	52	11	475	1.768489	76.39
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0	13	67	13	3	0	5	2	48	2.985075	71.64
5	00025465d4725e87	"\n\nCongratulations from me as well, use the ...	0	0	0	0	0	0	12	65	9	2	0	4	1	45	1.538462	69.23
6	0002bcb3da6cb337	COCKSUCKER BEFORE YOU PISS AROUND ON MY WORK	1	1	1	0	1	0	8	44	8	0	0	0	37	0	84.090909	0.00
7	00031b1e96af7621	Your vandalism to the Matt Shirvington article...	0	0	0	0	0	0	20	115	20	3	0	12	4	87	3.478261	76.65
8	00037261f536c51d	Sorry if the word 'nonsense' was offensive to ...	0	0	0	0	0	0	87	472	72	9	0	40	7	353	1.483051	74.78
9	00040093b2687caa	alignment on this subject and which are contra...	0	0	0	0	0	0	12	70	12	0	0	8	2	57	2.857143	81.42
10	0005300084f90edc	"\nFair use rationale for Image:Wonju.jpg\n\nT...	0	0	0	0	0	0	498	2875	196	56	0	224	53	2223	1.843478	77.32

Figure 9 : Modified Dataset with Sentiment Label

- Now finally what we are doing here is partitioning the data into two parts in the ratio of 70:30 where 70% is for training data and remaining 30% is for testing purpose. So what we are doing here is we are training the 70% of data with different classification methods and with different parameters and conditions.

4.5 Evaluation Criterian :

Accuracy :

Order Accuracy is the thing that we generally mean, when we utilize the term precision. It is the proportion of number of right forecasts to the all out number of info tests.

Accuracy=number of correct expectation / Complete number of prediction made

It functions admirably just if there are equivalent number of tests having a place with each class. For instance, consider that there are 98% examples of class An and 2% tests of class B in our preparation set. At that point our model can without much of a stretch get 98% preparing exactness by essentially anticipating each preparation test having a place with class A.

At the point when a similar model is tried on a test set with 60% examples of class An and 40% examples of class B, at that point the test exactness would drop down to 60%. Characterization Accuracy is incredible, yet gives us the misguided feeling of accomplishing high exactness.

The genuine issue emerges, when the expense of misclassification of the minor class tests are high. In the event that we manage an uncommon yet deadly illness, the expense of neglecting to analyze the sickness of a wiped out individual is a lot higher than the expense of sending a solid individual to more tests.

Confusion Matrix:

Confusion Matrix as the name proposes gives us a grid as yield and portrays the total presentation of the model.

Lets accept we have a parallel order issue. We have a few examples having a place with two classes : YES or NO. Likewise, we have our own classifier which predicts a class for a given information test. On testing our model on 165 examples, we get the accompanying outcome.

There are 4 significant terms:

- True Positives: The cases in which we anticipated YES and the genuine yield was additionally YES.
- True Negatives: The cases in which we anticipated NO and the real yield was NO.
- False Positives: The cases in which we anticipated YES and the genuine yield was NO.
- False Negatives: The cases in which we anticipated NO and the genuine yield was YES.

Accuracy for the network can be determined by taking normal of the qualities lying over the "primary inclining" i.e

Accuracy=total positive+false negative /Complete number of tests
Accuracy=(100+50)/165 =0.96
Precisitve= genuine positive /(Genuine positive +false negative)

Precision :

Accuracy is a decent measure to decide, when the expenses of False Positive is high. For example, email spam discovery. In email spam discovery, a bogus positive implies that an email that is non-spam (genuine negative) has been distinguished as spam (anticipated spam). The email client may lose significant messages if the accuracy isn't high for the spam recognition model.

Recall: recall= genuine positive/(total negative + false negative)

So Recall really ascertains the number of the Actual Positives our model catch through naming it as Positive (True Positive). Applying a similar comprehension, we realize that Recall will be the model metric we use to choose our best model when there is a significant expense related with False Negative.

F1 Score:

F1= 2* (Precision*Recall / Precision*Recall)
F1 Score is required when you need to look for a harmony among Precision and Recall. Right… so what is the contrast between F1 Score and Accuracy at that point? We have recently observed that exactness can be generally contributed by countless True Negatives which in many business conditions, we don't zero in on a lot though False Negative and False Positive for the most part has business costs (unmistakable and elusive) hence F1 Score may be a superior measure to utilize on the off chance that we have to look for a harmony among Precision and Recall AND there is a lopsided class dispersion (enormous number of Actual Negatives).

CHAPTER 5: RESULT AND ANALYSIS

5.1 Experimental Results

Following table shows different models with their accuracy scores, F1-Scorees, precision and Recall. From the results of following table, It is being observed that the complex models perform better than the simpler models. CNN and MLP (Multilayer Perceptron) model perform better than other and these two are equally good with minor gap only. CNN model with 4 layers provide the accuracy of 74.6%. With all the changes made and feature implementation it has been improved from the previous one.

Technique used	Accuracy	Precision	Recall	F1-score
Gaussian nave Bayes	67.4%	32.9%	61.3%	70.4%
KNN (K=3)	72.55%	57.4%	64.5%	72.9%
Support vector machine	72.23	43.12%	70.45%	74.1%
Multilayer perceptron	74.3%	55.8%	68.8%	74.9%
CNN with four layer	74.6%	63%	66.6%	74.0%

Table II Experimental Result of Different Classification Models

5.2 Analysis

For finding the features importance in the model, ablative analysis is used, features are removed one by one and in sequence to find out how much there is drop in accuracy if that feature is removed. As shown in figure the accuracy differs various features are removed in multilayer perceptron model. Without removing any of the feature it provides 74.346% accuracy.

The removal of features generated by Structural similarity affects the accuracy most which implies that these features play the key role in this model. Whereas removal of Trigram and Quad-gram do not alter the accuracy with a considerable amount which implies that the importance of these features is the least. Structural Similarity feature is consist of first degree common neighbors, first degree union neigh-bours, Second degree common neighbors and Second degree union neighbors.

In Features are removed in sequence one after one. So the Accuracy is decreasing as fea-tures are removed. It is being shown that TF-idf based similarity features and structural features are dropping the accuracy with a considerable amount which prove their importance.

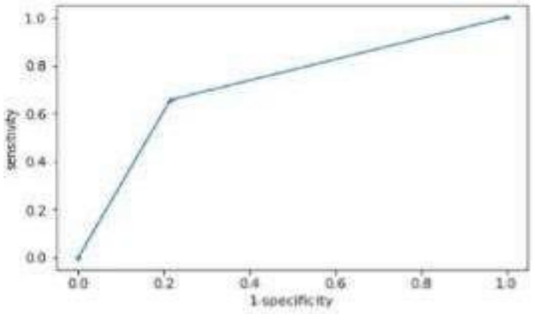


Figure 10 : ROC Curve for CNN

CHAPTER 6: CONCLUSION

In this thesis, we have worked upon to improve the accuracy of results with dual implementation or feature extraction. Sentiment analysis makes ease in identifying people‟s emotional and attitudes states. People‟s feeling that can be expressed in positive or negative ways. This paper talks about in suitable elements the different ways to deal with sentiment Analysis, mostly ML and Lexicon-based approaches. This thesis gives a point by point perspective of the distinctive applications and challenges of Sentiment Analysis. Sentiment analysis can be extremely compelling inforeseeing decision comes about, securities exchange or motion picture survey like Imdb audits of facebook and twitter can be likewise used to give helpful information which can be utilized to anticipate future.

Here we have seen that we can improve the efficiency or accuracy of any system related to emotion mining, sentiment labeling or prediction based on the sentiment from different profiles with the help of hybrid classification as we are adding multiple functionality like data cleansing, attributes based on different features and functionality, classification models with different parameters.

CHAPTER 7: REFERENCES

1. NADAQ: Natural Language Database Querying Based on Deep Learning by BOYAN XU1, RUICHU CAI 1, ZHENJIE ZHANG2, XIAOYAN YANG2, ZHIFENG HAO1,3, ZIJIAN LI1, AND ZHIHAO LIANG on February 20, 2019.

2. Sentiment Analysis in A Cross-Media Analysis Framework Yonas Woldemariam Department of Computing Science Umea University Umea, Sweden e-mail: yonasd@cs.umu.se

3. Saeed, H. H., Shahzad, K., & Kamiran, F. (2018). Overlapping Toxic Sentiment Classification Using Deep Neural Architectures. 2018 IEEE International Conference on Data Mining Workshops (ICDMW). doi:10.1109/icdmw.2018.00193

4. Z. Zhang, D. Robinson, and J. Tepper, "Detecting hate speech on twitter using a convolution-gru based deep neural network," in European Semantic Web Conference. Springer, 2018

5. S.Padmaja, S.Sameen, and S.Bandu, "Evaluating sentiment analysis methods and identifying scope of negation in newspaper articles," vol.3 , No.11, no. 11, 2014.

6. Quanzhi Li, Sameena Shah, Rui Fang, Armineh Nourbakhsh, Xiaomo Liu Research and Development Thomson Reuters 3 Times Square, NYC, NY 10036 {quanzhi.li, sameena.shah, rui.fang, armineh.nourbakhsh, xiaomo.liu}@thomsonreuters.com

7. Jin Ding∗†, Hailong Sun∗†, Xu Wang∗†, Xudong Liu∗† SKLSDE Lab, School of Computer Science and Engineering, Beihang University∗ Beijing Advanced Innovation Center for Big Data and Brain Computing† Beijing, China
{dingjin,sunhl,wangxu,liuxd}@act.buaa.edu.cn

8. R. Xia, C. Zong, and S. Li, "Ensemble of Feature Sets and Classification Algorithms for Sentiment Classification," Information Sciences, vol. 181, no. 6, pp. 1138-1152, 2011.

9. Comparative Analysis of Customer Sentiments on Competing Brands using Hybrid Model Approach Comparative Analysis of Customer Sentiments on Competing Brands using Hybrid Model Approach

10. For data set and the applications on which these methodologies are implemented are taken from GitHub.

11. Dataset for the experimental results have been taken from Kaggle one from Toxic Comment Classification and other is from IMDb rating prediction.

12. Dasha Bogdanova, Cicero dos Santos, Luciano Bartions in online user forums," in Computational

Natural Language Learning, Volume: 3 , Issue: 5,2015.

13. Nitesh Pradhan and Manasi Gyanchandani, “A Re-view on Text Similarity Technique used in IR and its Application,” in A Review on Text Similarity Tech-nique used in IR and its Application,2015.

14. https://en.wikipedia.org/wiki/Sentiment_analysis.

15. https://machinelearningmastery.com/types-of-classification-in-machine-learning/

16. https://www.geeksforgeeks.org/regression-classification-supervised-machine-learning/

17. Kajal Sarawgi, Vandana Pathak, “Opinion Mining: Aspect Level Sentiment Analysis using SentiWordNet and Amazon Web Services”, International Journal of Computer Applications, pp.31- 36, 2015