ABSTRACT

We use technology for our advantage by applying it in almost every aspect of our lives. Nowadays as people spend a great deal of time on internet, they tend to involve more in expressing their opinions, communicating with each other through various social media or business platforms. The fundamental aspect of our lives is that other's opinions and interests are important to us. Although measuring feelings and emotions can be quite difficult, there are still ways. Sentiment analysis and Emotional analysis is the computational study of sentiments and emotions expressed through various platforms by giving two different insights about the user. Sentiment analysis classifies user data into categories like positive and negative. Emotional analysis aims to understand the emotion conveyed by the user. The use of text is not new, but as people are opting simpler and quicker ways to communicate, the use of emoticons and gif became popular. This paper reflects the idea of taking user opinions expressed through text, emoticon and gif by performing sentiment, emotion analysis and establishing conclusions by using machine learning algorithm. This analysis essentially serves a gateway to improve consumer service, increase sales revenue and generates growth opportunities in businesses.

Keywords: Sentiment Analysis, Emotional Analysis, Support Vector Machines, Preprocessing, Tokenization, Lemmatization.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

The opinions of others matter a lot as they are an important reflection of our human behavior and have a significant influence in our daily decision-making process. Social media and other online tools had not only mediated communication in countless ways, but also a way to actively use information technologies to seek out and understand the opinions of others Sentimental and emotional analysis has served as a reliable source in day-to-day by providing insightful opinions about several products rolled over in the market, innovative ideas, people opinion about new policies framed by government etc.

Human emotions are extremely diverse and cannot be restricted to certain metrics alone. They are so complex, and reducing them to positive and negative can only give you a shallow understanding. Polarity analysis is too generic as it does not specify the actual intent of message delivered by author and just positive or negative classes are not sufficient to understand nuances of underlying tone of a sentence. This brings the need to take one step above sentiment analysis leading to emotion analysis which is based on a wide spectrum of moods rather than a couple of static categories. In this paper we throw light on methods we have used to derive sentiment analysis and how we have accomplished emotion analysis of user opinions expressed through text, emoticons and gif. Emoticons and Gifs are digital images, aims to help people convey more emotion and expression than the static written word, especially as images are processed quicker than text. A supervised learning technique provides labels to classifier to make it understand the insights among various features. Once the classifier gets familiarized with train data it can perform classification on unseen test data. We have chosen Support Vector Machine classification algorithm to carry out sentiment and emotional analysis. Emotion detection involves a wide platter of emotions classified into states like happiness, hate, love, relief, surprise and worry. We here examine sentiments and emotions of text combined with emoticons and gif.

Generally people discuss a lot of things daily but it is difficult to get insights just by reading through each of their opinions so there should be a way that helps us to get insights of users opinions in an unbiased manner, So this model helps in drawing out Sentiment, Emotions of users, classify them and finally present them to us. Sentiment analysis is the prediction of emotions in a word, sentence or corpus of documents. It is intended to serve as an application to understand the attitudes, opinions and emotions expressed within an online mention. The intention is to gain an overview of the wider public opinion behind certain topics.

Sentiment Analysis

Sentiment analysis (also known as opinion mining) refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is contextual mining of text which identifies and extracts subjective information in source material, and helping a business to understand the social sentiment of their brand, product or service while monitoring online conversations. However, analysis of social media streams is usually restricted to just basic sentiment analysis and count based metrics. This is akin to just scratching the surface and missing out on those high value insights that are waiting to be discovered.

Emotional Analysis

It is the process of identifying human emotions, most typically from facial expressions as well as from verbal expressions. It relies on a deeper analysis of human emotions and sensitivities. Just as with other data related to customer experience, emotional data is used to create strategies that will improve the business's customer relationship management (CRM).

1.1 MOTIVATION FOR WORK

Businesses primarily run over customer's satisfaction, customer reviews about their products. Shifts in sentiment on social media have been shown to correlate with shifts in stock markets. Identifying customer grievances thereby resolving them leads to customer satisfaction as well as trustworthiness of an organization. Hence there is a necessity of an unbaised automated system to classify customer reviews regarding any problem. In today's environment where we're justifiably suffering from data overload (although this does not mean better or deeper insights), companies might have mountains of customer feedback collected; but for mere humans, it's still impossible to analyze it manually without any sort of error or bias. Oftentimes, companies with the best intentions find themselves in an insights vacuum. You know you need insights to inform your decision making and you know that you're lacking them, but don't know how best to get them. Sentiment analysis provides some answers into what the most important issues are, from the perspective of customers, at least. Because sentiment analysis can be automated, decisions can be made based on a significant amount of data rather than plain intuition that isn't always.

1.2 PROBLEM STATEMENT

The basic task in sentiment and emotional analysis is to generate statistical information containing various emotions and also to predict the appropriate emotion in the given text, emoticon and gif. This can be done by using machine learning algorithms.

2. LITERATURE SURVEY

2.1 INTRODUCTION

The opinions of others matter a lot as they are an important reflection of our human behavior and have a significant influence in our daily decision-making process. What other people think has always been an important piece of information. The Internet and the Web have now (among other things) made it possible to find out about the opinions and experiences of those in the vast pool of people that are neither our personal acquaintances nor well-known professional critics — that is, people we have never heard of. And conversely, more and more people are making their opinions available to strangers via the Internet. The interest that individual users show in online opinions about products and services, and the potential influence such opinions wield, is something that is driving force for this area of interest. And there are many challenges involved in this process which needs to be walked all over in order to attain proper outcomes out of them.

2.2 EXISTING METHODS

2.2.1 Sentiment Analysis Using Machine Learning for Twitter:

It is an effort to implement Sentiment Analysis Using Machine Learning for Twitter. It used two classifiers SVM and Naive Bayes. It got achieved 63 percent accuracy using SVM and 58 percent accuracy using Naive Bayes classifier. Hence, we conclude that SVM is best classifier to be used for Sentiment Analysis Using Machine Learning for Twitter. From future perspective, we'd wish to extend this project by implementing some machine learning algorithms for applications like election results, product ratings, movies' outcomes and running the project on clusters to expand its functionality. Moreover, we'd wish to make an online applicationfor users to input keywords and find analyzed results. During this project, it got worked only with unigram models, but we'd wish to extend it to bi gram and further which is able to increase linkage between the info and supply accurate sentiment analysis result.

2.2.2 Classification of Emotions from text using SVM based Opinion Mining :

SVM classification using Quadratic programming was used. Steps included preparing the data set ,annotating the dataset with predefined emotions , using NLP prepare the database matrix of test emotions and training emotions, classify the training set with a support vector machine using quadratic programming algorithm. Compute the prediction of support vector machine using kernel function and its parameter for classification and finally compute the accuracy of the classification. The basic idea of SVM is to find the optimal hyperplane to separate two classes with largest margin of pre-classified data. After this hyperplane is determined it is used for classifying data into two classes based on which side they are located. By applying appropriate transformations to the data space after computing the separating hyperplane, SVM can be extended to cases where the margin between two classes is non-linear. Finally, on classifying the data set, superior results have been obtained.

2.2.3 Product opinion mining using sentiment analysis on smartphone reviews:

We are always eager and excited to know what people think what they feel and perceive about various aspects of living and non-living beings. In need to understand and analyze various traits of behavior and the varying personality there is a need of opinion mining. It tried to examine an unstructured Smartphone reviews that are being collected from a very well-known marketing site termed as Amazon. The data set was being classified amongst the classes that is the positive reviews and the negative reviews. The data set had been under gone through two well-known machine learning techniques ie Naïve Bayes and Support Vetor machine Classifier. The overll accuracy of the classifier thus trained using

Naïve Bayes Classification technique was around 40% which was quite un-satisfactory to deal with and to rely on. So in order to have a reliable trained model which is able to classify the data according to our needs we have opted another approach which is SVM. The same data set which when parsed under this approach [produces good result with an accuracy of around 90%.Hence the SVM is more realiable.

2.2.4 Sentiment analysis using n-gram algo and svm classifier:

The sentiment analysis is the technique which can analyze the behavior of the user. Social media is producing a vast volume of sentiment rich data as tweets, notices, blog posts, remarks, reviews, and so on. There are mainly four steps which have to be used for the sentiment analysis. The data pre- processing is done in first step. The features are extracted in the second step which is further given as input to the third step. For the sentiment analysis, data is classified in the third step. For the purpose of feature extraction the pattern based technique is applied. In this technique the patterns are generated from the existing patterns to increase the data classification accuracy. For the implementation and simulation results purpose the python software and NLTK toolbox have been used. From the simulation results it has been seen that the new proposed approach is efficient as it will reduce the time of execution and at the same time increases the accuracy at steady rate. the sentiment analysis is the efficient technique to analyze the people behavior. The sentiment analysis contains the four steps and in this work improvement in the feature extraction phase is using the pattern based technique. The proposed improvement is analyzed that execution time is reduced to 10 percent and accuracy is increased to 20 percent.

2.2.5 Methods for Sentiment Analysis:

In this paper, various approaches to sentiment analysis have been examined and analyzed, Techniques such as Streaming API SVM etc., discussed. These techniques all have different strengths and weaknesses. 1.Sentiment Analysis on Twitter using streaming API: It uses NLP where it helps in tokenization, stemming, classification, tagging, parsing and sentiment reasoning its basic feature is to convert unstructured data into structured data. It uses Naive Bayes for classification which requires number of linear parameters. To find out the sentiment an automated system must be developed “Support Vector Machine” can be used for this method. SVM is machine that takes the input and store them in a vector then using SentiWordNet it scores it decides the11 sentiment. It also classifies the opinion in overall way by positive, negative or Neutral. There are many more techniques but these are the most familiar ones , performs more efficiently. And selection of both features and techniques affect the final outcome. So proper analysis must be done to get intended , as accurate results as possible.

3. METHODOLOGY

The sentiment emotional analysis is an emerging field that needs much more attention. We have taken a text dataset from Kaggle and stored into a csv file. Thus our train data set without pre-processing is ready.Next we perform pre-processing to clean, remove unwanted characters out of the text.Then we train our classifier by fitting the train data to the classifier, there after prediction of results over unseen test data set is made. Which there after provides us with the accuracy with which the classifier had predicted the outcomes. The emoticons are taken with their respective Unicode to obtain their text description. The Graphics Interchange Format, or Gifs are taken from the internet in the form of a hyperlinks with their respective text descriptions. When a gif or emoticon is selected in the input its text description is retrieved from the input file to perform preprocessing on the obtained text. The above preprocessing steps are performed on the retrieved text for further processing in order to get the desired emotion. Thereafter, We present our results in a pictorial manner which is the best way to showcase results because of its easiness to understand information out of it.

3.1 PROPOSED SYSTEM

Data Collection:

The first step is to collect the data for predicting the emotion. We have the text dataset from kaggle.

Preprocessing of data:

The data is preprocessed before giving to classifier. This will convert the data into the format that is used to train the SVM model.

Tokenization:

Tokenization is the process of converting text into tokens before transforming it into vectors. It is also easier to filter out unnecessary tokens. For example, a document into paragraphs or sentences into words. In this case we are tokenizing the reviews into words.

Removing punctuations and special symbols:

The punctuations and symbols like !,\ are removed.

Stop words removal:

Stop words are the most commonly occurring words which are not relevant in the context of the data and do not contribute any deeper meaning to the phrase. In this case contain no sentiment. NLTK provide a library used for this.

"This is a cool picture, do you want to have it."

['This', 'is', 'a', 'cool', 'picture', ',', 'do', 'you', 'want', 'to', 'have', 'it', '.']

After stop words removal:

['This', 'cool', 'picture', ',', 'want', '.']

Lemmatization:

Sentences are always narrated in tenses, singular and plural forms making most words accompany with -ing, -ed, es and ies. Therefore, extracting the root word will suffice to identify sentiment behind the text.

Base forms are the skeleton for grammar lemmatization reduces inflectional forms and derivational forms to common base form.

Example: Cats is reduced to cat, ponies is reduced to poni.

The goal of lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. With that being said, stemming/lemmatizing helps us reduce the number of overall terms to certain “root” terms.

Rule			**Example**		
SSES	→	SS	caresses	→	caress
IES	→	I	ponies	→	poni
SS	→	SS	caress	→	caress
S	→		cats	→	cat

Fit data to classifier:

Train data is fitted to a suitable classifier upon feature extraction, then once the classifier is trained enough then we predict the results of the test data using the SVM classifier **.**

Predict the test data:

Now we predict the test data by using the same model to get the emotion.

The emoticons are taken with their respective Unicode to obtain their text description. The Graphics Interchange Format, or Gifs are taken from the internet in the form of a hyperlinks with their respective text descriptions. When a gif or emoticon is selected in the input its text description is retrieved from the input file to perform preprocessing on the obtained text. The above preprocessing steps are performed on the retrieved text for further processing in order to get the desired emotion.

Results:

Now based on the input we give in the form of text, emoticon or gif it predicts the emotion and also it divides as positive or negative based on the emotion.

The results are shown in bar graph.

3.1.1 SYSTEM ARCHITECTURE

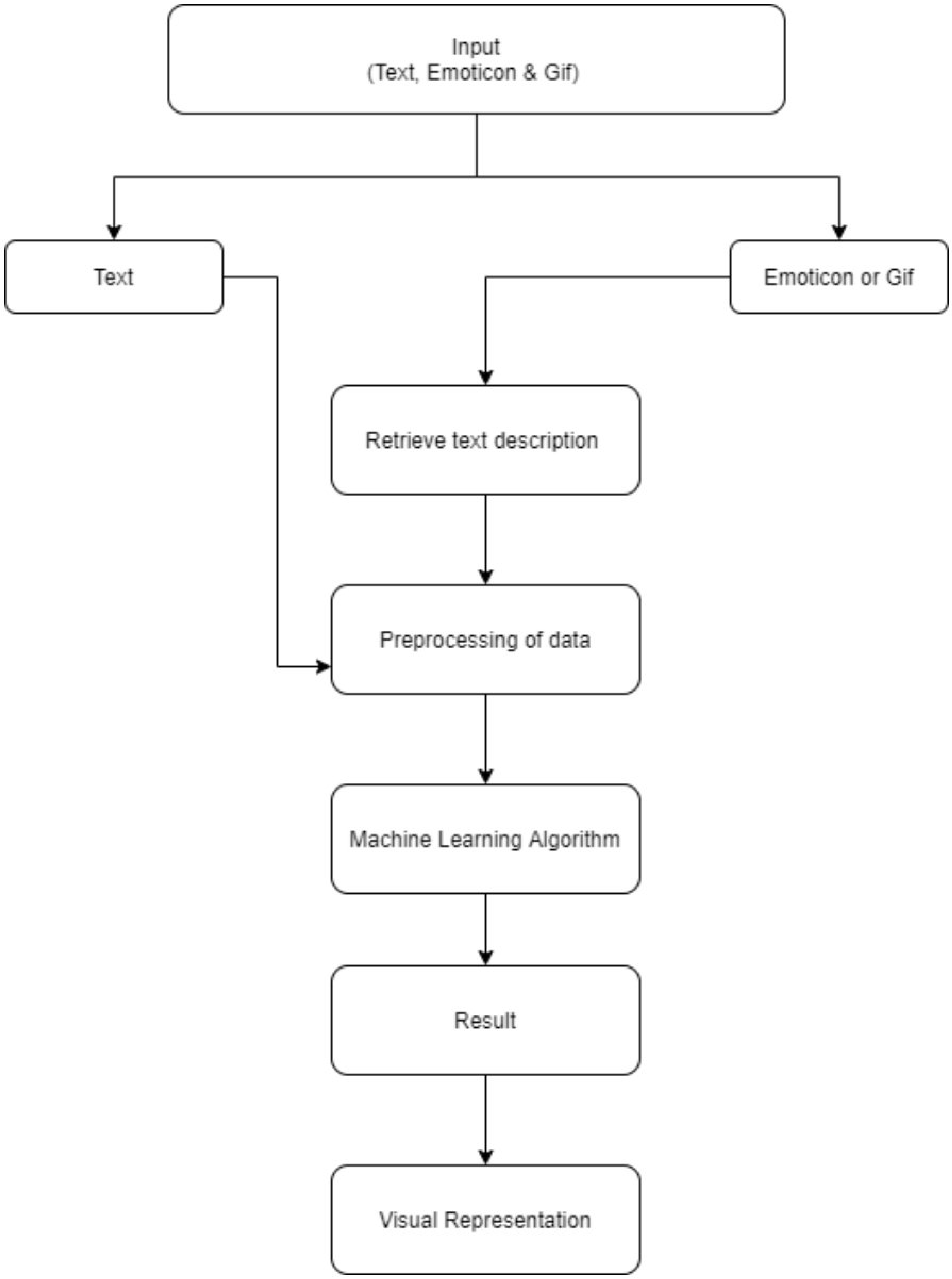


Fig 3.1.1 System Architecture of sentiment and emotion detection

3.1.2 Support Vector Machine:

Support Vector Machines is a supervised machine learning algorithm , adopted conventionally for classification as well as regression problems. SVMs for classification, work by figuring out the right hyperplane among the classes. After being trained by a labeled data set, SVM outputs an optimal hyperplane that categorizes new examples. Classification by SVMs for different data sets is governed by tuning parameters namely kernel, regularization, gamma and margin. When data is 2 dimensional Support vector classifier is a line, if it is 3D SVC forms a plane instead of a line. When data is more than 4D then classifier is a hyperplane. For highly distributed data Maximal margin and support vector classifier fail and hence SVMs are used. For linearly separable patterns optimal hyperplane is formed and for non-linearly separable patterns transformation of original data into a new space is performed determined by kernel function. The trouble of discovering an optimal hyperplane is an optimization problem and can be worked out using optimization techniques (eg. Lagrange). To classify tweets into different emotion classes a linear kernel has been utilized. Linear kernel is preferable for text classification problems because text has lot of features, linear kernel is faster and less parameters are optimized. When SVM is trained with a linear kernel only C regularization parameter need to be optimized whereas for other kernels you need to optimize gamma parameter also. Support Vector Machines (SVM) is a machine learning model proposed by V. N. Vapnik. The basic idea of SVM is to find an optimal hyperplane to separate two classes with the largest margin from pre-classified data. After this hyperplane is determined, it is used for classifying data into two classes based on which side they are located. By applying appropriate transformations to the data space before computing the separating hyperplane, SVM can be extended to cases where the margin between two classes is non-linear.

We have used Support Vector Machine algorithm for our project. It is a supervised machine learning algorithm. In this algorithm, the data item is plotted as a point in n dimension space having value of feature represented as the value of coordinate. After that we perform classification by finding the decision boundary that divides the classes. We use LinearSVC model for our project. Kernel is used to transform data into a suitable form to classify and train the data. In this we use linear kernel. For calculating the probabilities per each class CalibratedClassifierCV is used in LinearSVC to use predict_proba method.

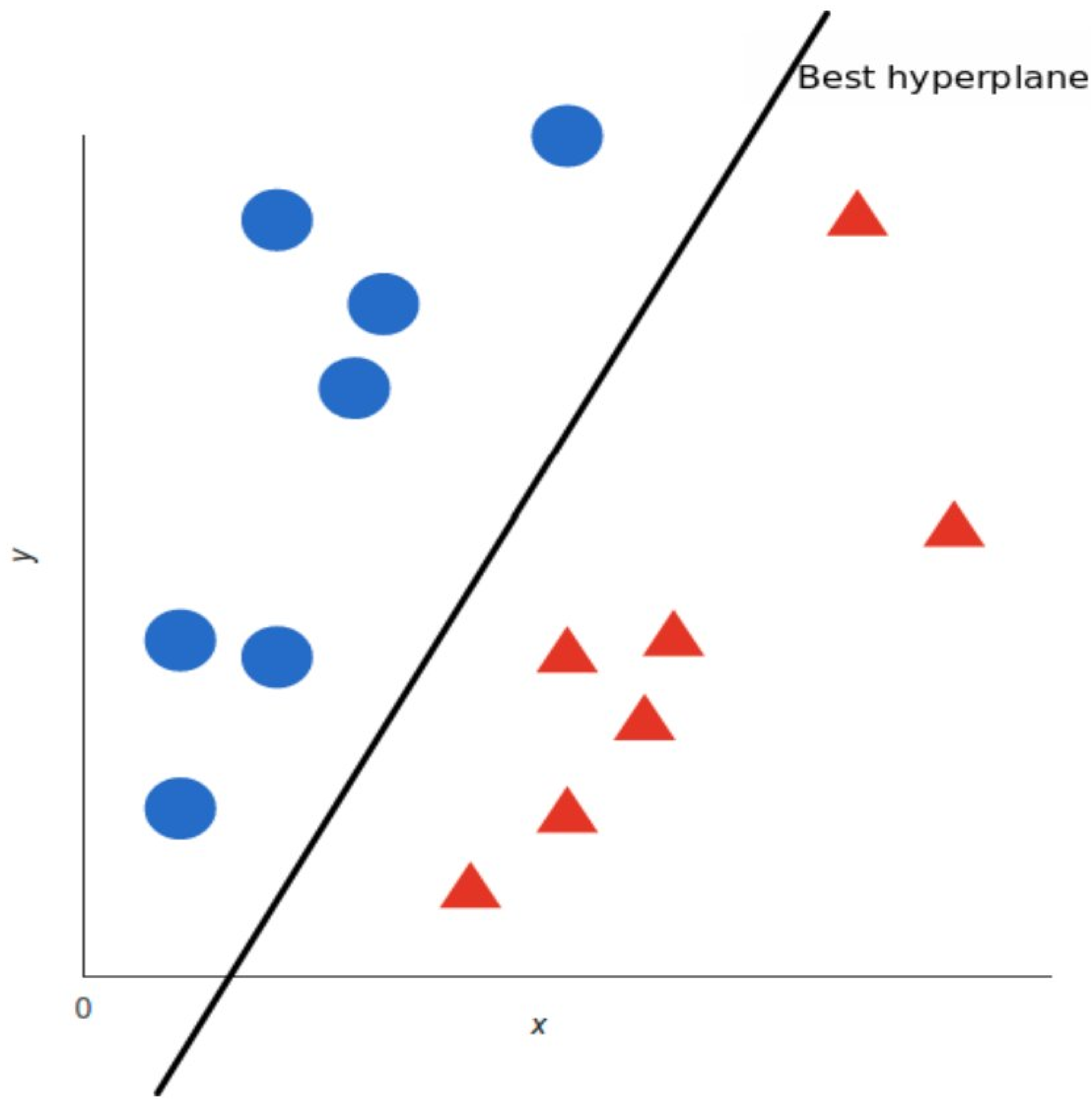


Fig 3.1.2 Support vector machine

4. REQUIREMENTS

4.1 SYSTEM CONFIGURATION

4.1.1SOFTWARE REQUIRMENTS

Following are the software and modules that needs to be installed for successful execution of the project. They are:

1.Nltk

2.Scikit-learn

3.Matplotlib

4.Pandas

5.Numpy

6.Csv

7.Re(Regular Expressions)

8.Windows

9.Flask

10.emoji

11.json

12.joblib

4.1.2 HARDWARE REQUIREMENTS

Following are the hardware requirements necessary for faster execution of the code.

1. A minimum of Intel Core I3 processor
2. A minimum of 4 GB Ram
3. Cpu with atleast 2 cores of clock speeds

4.Architecture: 32-bit or 64-bit

5.SAMPLE CODE

```
<html>
<head>
   <title>Emotion Detection</title>

   <script src="./static/jquery.min.js"></script>
   <script src="./static/chart.js"></script>

   <style>
      @import url('https://fonts.googleapis.com/css2?family=Caveat&display=swap');

      body{
         /*background-image: url('https://cliply.co/wp-content/uploads/2020/01/401911130_THINKING_FACE_BG_400.gif');*/
         background-image: url('https://pixy.org/download/99194/');
         /*background-image: url('https://ctl.s6img.com/society6/img/4zHY1GAQI-iWD4WevWVtxLXbRPw/w_700/canvas/~artwork/s6-original-art-uploads/society6/uploads/misc/ced1ea1bc6934696bab5f2e285b92e09/~~/fun-cool-happy-blue-smiley-faces-canvas.jpg');*/

         /*background-image: url('https://media.istockphoto.com/photos/blue-smileys-in-social-media-concept-3d-rendering-picture-id680897220');*/

         /*background-size: cover;*/
         /*position: center;*/
         /*background: linear-gradient(to top, #ffdde1, #ee9ca7);  W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+, Safari 7+ */
```

```
      /*background: linear-gradient(to left, #cb356b, #bd3f32);*/
       /*background-image: url("/This PC/Desktop/image.jpg");*/
       /*background: linear-gradient(to right, #141e30, #243b55);*/
       background-color: #cccccc;

    height: 100vh;
    padding:0;
    margin:0;
}
.split {
  height: 500px;
  width: 50%;
  position: fixed;
  z-index: 1;
  top: 0;
  overflow-x: hidden;
  padding-top: 20px;
}

.left {
    position: absolute;
    top: 100px;
    left: 2.5%;
    width: 45%;
    box-shadow: 0 15px 25px rgb(0 0 0 / 50%);
    padding: 5px;
  background-color:white;
}
```

```css
.right {
   position: absolute;
   top: 100px;
   right: 2.5%;
   width: 45%;
   box-shadow: 0 15px 25px rgb(0 0 0 / 50%);
   padding: 10px;
    background-color: pink;
}

.centered {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  text-align: center;
}

.centered img {
  width: 150px;
  border-radius: 50%;
}
.btn {
  border: 2px solid black;
  background-color: white;
  color: black;
  padding: 14px 28px;
  font-size: 16px;
  cursor: pointer;
```

```css
}

/* Green */
.text{
 border-color: #04AA6D;
 color: green;
}

.text:hover {
 background-color: #04AA6D;
 color: white;
}

/* Blue */
.emoji_btn {
 border-color: #2196F3;
 color: dodgerblue;
}

.emoji_btn:hover {
 background: #2196F3;
 color: white;
}

/* Orange */
.btn_gif {
 border-color: #993366;
 color: #993366;
}
```

```
.btn_gif:hover {
  background: #993366;
  color: white;
}
.submit {
  border-color:#5eb074;
  color: #5eb074;
}

.submit:hover {
  background: #5eb074;
  color: white;
}

.emojiDiv {
  width: 100%;
  height: 300px;
  background-color: white;
  overflow-y: scroll;
}

.emojiDiv .emoji {
    /*height: 40px;*/
    /*width: 40px;*/
    padding:5px;
    margin: 5px;
    float: left;
    text-align: center;
```

```
    cursor: pointer;
    transform:scale(1.5);
}

.emojiDiv .emoji:hover {
    background-color: rgba(0, 0, 0, 0.2);
}

.gifDiv {
  width: 100%;
  height: 300px;
  background-color: white;
  overflow-y: scroll;
}

.gifDiv .gif {
    height: 150px;
    width: 120px;
}

#myTextarea {
    width: 100%;
    height: 200px;
}

.prediction {
    text-align: center;
    margin-top: 10px;
    font-size: 30px;
```

```
        height: 30px;
        /*font-family: 'Caveat', cursive;*/
      }

      .graph {
        /*position: absolute;*/
        margin-top: 30px;
        width: 76%;
      }

       .graph1 {
        /*position: absolute;*/
        margin-top: 30px;
        margin-left: 50px;
        width: 50%;
      }


  </style>
  <link                                                              rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/emojionearea/3.4.2/emojionearea.min.css">
  <!-- /*what*/ -->
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/emojionearea/3.4.2/emojionearea.min.js"></sc
ript>
</head>
<body>
  <div class="split left">
    <div class="prediction"></div>
```

```
        <div class="graph">
            <div class="emotionGraphDiv">
                <canvas id="emotionGraph"></canvas>
            </div>
        <div class="graph1">
            <div class="emotionGraphDiv1">
                <canvas id="emotionGraph1"></canvas>
            </div>
        </div>
        </div>
    </div>

    <div class="split right">
        <textarea id="myTextarea"></textarea>

        <button onclick="showHideEmojis()" class="btn emoji_btn">emoji</button>
        <button onclick="showHideGifs()" class="btn btn_gif">gif</button><br />

        <div class="emojiDiv"></div>
        <div class="gifDiv"></div>

        <center><button onclick="predict()" style="margin-top: 30px" class="btn
submit">submit</button></center>
    </div>
</body>
</html>

<script>
```

```
function showHideEmojis() {
   $('.emojiDiv').toggle();
   $('.gifDiv').hide();
}
function showHideGifs() {
   $('.gifDiv').toggle();
   $('.emojiDiv').hide();
}

function addEmojiToText(emoji) {
   $('#myTextarea').val($('#myTextarea').val()+emoji);
}
function loadGraph1(res) {
   var labels = [];
   var labelData = [];

   for (var i in res) {
      labels.push(i);
      labelData.push(res[i]);
   }


   const data = {
     labels: labels,
     datasets: [
       {
         data: labelData,
         label: "",
         borderColor: "#2196f3",
```

```
      backgroundColor:"dodgerblue"
    }
  ]
};

const config = {
  type: 'bar',
  data: data,
  options: {
    responsive: true,
    plugins: {
      legend: {
        position: 'top',
        display: false
      },
      title: {
        display: false
      }
    }
  },
};

$('.emotionGraphDiv1').html(`<canvas id="emotionGraph1"></canvas>`);
//to display graph we pass canvas to div

var myChart1 = new Chart(
    document.getElementById("emotionGraph1"),
    config
);
```

```
}

function predict() {
    var text = $('#myTextarea').val();

    $.ajax({
        url: 'http://localhost:7676/predict',
        type: "POST",
        data: {"text": text},
        success: (res) => {
            console.log(res);

            $('.prediction').html("Predicted Emotion : "+ res.prediction);
            loadGraph(res.graphData);
            loadGraph1(res.sentiment);
        }
    });
}

function predictGif(gifid) {
    $.ajax({
        url: 'http://localhost:7676/predictGif',
        type: "POST",
        data: {"gifid": gifid},
        success: (res) => {
            console.log(res);

            $('.prediction').html("Predicted Emotion : "+ res.prediction);
            loadGraph(res.graphData);
```

```
            loadGraph1(res.sentiment);
        }
    });
}

function loadGraph(res) {
    var labels = [];
    var labelData = [];

    for (var i in res) {
        labels.push(i);
        labelData.push(res[i]);
    }


    const data = {
      labels: labels,
      datasets: [
        {
          data: labelData,
          label: "",
          borderColor: "#2196f3",
          backgroundColor:"dodgerblue"
        }
      ]
    };

    const config = {
      type: 'bar',
```

```
        data: data,
        options: {
          responsive: true,
          plugins: {
            legend: {
              position: 'top',
              display: false
            },
            title: {
              display: false
            }
          }
        },
      };

      $('.emotionGraphDiv').html(`<canvas id="emotionGraph"></canvas>`);
      //to display graph we pass canvas to div

      var myChart = new Chart(
        document.getElementById("emotionGraph"),
        config
      );
    }


//what
  $(document).ready(() => {

      $.ajax({
```

```
        url: 'http://localhost:7676/getEmojis',
        type: "POST",
        success: (res) => {
            // console.log(res);
            $('.emojiDiv').hide();
            var emojis = ``;

            for (var i in res) {
                var s = i.split(" ");
                var emoji = "";
                for (var j in s) emoji += '&#x' + s[j].substr(2) + ';';
                // console.log(emoji)


                emojis += `<div onclick="addEmojiToText('`+emoji+`')"
class="emoji">`+emoji+`</div>`;
            }

            $('.emojiDiv').html(emojis);
        }
    });

    $.ajax({
        url: 'http://localhost:7676/getGifs',
        type: "POST",
        success: (res) => {
            // console.log(gifs);
            $('.gifDiv').hide();
            var gifs = ``;
```

```
            for (var i in res) {
              gifs += `<img onclick="predictGif(`+i+`)" class="gif" src="`+res[i].link+`"
/>`;
            }

            $('.gifDiv').html(gifs);
          }
        });

        // getBarGraphData
        $.ajax({
          url: 'http://localhost:7676/getBarGraphData',
          type: "POST",
          success: (res) => {
            console.log(res);



            loadGraph(res);
          }
        });
        $.ajax({
          url: 'http://localhost:7676/getBarGraphData1',
          type: "POST",
          success: (res) => {
            console.log(res);
```

```
            loadGraph1(res);
        }
      });
  });

</script>
```

EMOTION DETECTION:

```
import nltk
import pandas as pd
import numpy as np
import nltk
import re
import itertools
from io import StringIO
import matplotlib.pyplot as plt
import joblib
import functools
import operator
import re
import emoji
import json
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn import svm
from sklearn.model_selection import train_test_split
from nltk.tokenize import word_tokenize

from nltk.stem import WordNetLemmatizer
```

```
from nltk.stem.wordnet import WordNetLemmatizer
# from sklearn.ensemble import RandomForestClassifier
# import warnings
# warnings.filterwarnings("ignore")
```

```
data = pd.read_csv(r'./datasets/Emotion_final.csv')
print(data['Emotion'].unique())

h = {}
h2 = {"positive": 0, "negative": 0}

for emotion in data['Emotion']:
   if (emotion in h): h[emotion] += 1
   else: h[emotion] = 1

   if (emotion == "happy" or emotion=="love" or emotion=="surprise" ): h2["positive"]
+= 1
   elif (emotion == "sadness" or emotion=="fear" or emotion=="anger" ): h2["negative"]
+= 1
print(h)
print(h2)

# n = data.shape[0]
# for i in range(n):
```

```
#     if data['sentiment'][i]=="fun" or  data['sentiment'][i]=="enthusiasm":
#         data['sentiment'][i]="happiness"
#     elif data['sentiment'][i]=="sadness" or data['sentiment'][i]=="empty" or data['sentiment'][i]=="boredom":
#         data['sentiment'][i]="worry"


lem = WordNetLemmatizer()
data.head()

# data1 = pd.read_csv('./datasets/emoji_dataset.csv')
# data1.head()

# # x='🙈'
# # print('U+{:X}'.format(ord(x)))
# # print(ord('H'))
# def split_emojis(em):
#     em=str(em)
#     em_split_emoji = emoji.get_emoji_regexp().split(em)
#     em_split_whitespace = [substr.split() for substr in em_split_emoji]
#     # print(em_split_whitespace)
#     em_split = functools.reduce(operator.concat, em_split_whitespace)
#     emojis = []
#     for separated in em_split:
#         emojis.append(separated)
#     return emojis
```

```
# # j={}
# for row in data1.values:
#   j[row[2]]=row[4]

# j=json.dumps(j)
# f=open('./datasets/emoji_dataset.json','w')
# f.write(j)
# f.close()

# f=open('./datasets/emoji_dataset.json')
# s=f.readlines()[0]
# jsn = json.loads(s)
# jsn["U+1F46B"]

def cleaning(text):
    txt = str(text)
    txt = re.sub(r"http\S+", "", txt)
    if len(txt) == 0:
        return 'no text'
    else:
        txt = txt.split()
        index = 0
        for j in range(len(txt)):
            if txt[j][0] == '@':
                index = j
        txt = np.delete(txt, index)
        if len(txt) == 0:
            return 'no text'
        else:
```

```
        words = txt[0]
        for k in range(len(txt)-1):
            words+= " " + txt[k+1]
        txt = words
        txt = re.sub(r'[^\w]', ' ', txt)
        if len(txt) == 0:
            return 'no text'
        else:
          # for removing more than 2 times of char
            txt = ''.join(''.join(s)[:2] for _, s in itertools.groupby(txt))
            txt = txt.replace("'", "")
            txt = nltk.tokenize.word_tokenize(txt)

            for j in range(len(txt)):
                txt[j] = lem.lemmatize(txt[j], "v")
            if len(txt) == 0:
                return 'no text'
            else:
                return txt

print("Cleaning")
data['Text'] = data['Text'].map(lambda x: cleaning(x))
# progress=0
# tot_progress=data.shape[0]
# for x in data['content']:
#     progress+=1
#     print(progress,'/',tot_progress,end="\r")
#     cleaning(x)
```

```
# print("Emoji and text spliting")
data = data.reset_index(drop=True)
for i in range(len(data)):
    mul=data.Text[i]
    # mul = split_emojis(data.content[i])
    s1 = []
    for j in mul:
#        if (len(re.findall(r'[^\w\s,.]', j))==1): # emoji
#            s1.append(jsn['U+{:X}'.format(ord(j))])
#        else: # text
      s1.append(j)

    data.Text[i] = ' '.join(s1)
#    # print(data.content[i])



print("Test train split")

x_train, x_test, y_train, y_test = train_test_split(data.Text, data.Emotion, test_size=0.2,
random_state=10)

x_train = x_train.reset_index(drop = True)#old col indexs are not sent to new dataframes
x_test = x_test.reset_index(drop = True)

y_train = y_train.reset_index(drop = True)
y_test = y_test.reset_index(drop = True)
```

```
vectorizer = TfidfVectorizer(min_df=3, max_df=0.9)#used to cal word freq

train_vectors = vectorizer.fit_transform(x_train)

test_vectors = vectorizer.transform(x_test)



print("Training started...")

from sklearn.calibration import CalibratedClassifierCV

# model = svm.SVC(kernel='linear',probability=True,verbose=1,max_iter=1000) #82
# model=svm.LinearSVC()
# model.fit(train_vectors, y_train)#to train the data chooses best model
model = CalibratedClassifierCV(base_estimator=svm.LinearSVC())
# model = CalibratedClassifierCV(base_estimator=RandomForestClassifier(n_estimators=10), cv=4)
model.fit(train_vectors, y_train)
res = model.predict_proba(test_vectors)
# print(model.classes_)
# print(list(res[0]))


print("saving Model...")

joblib.dump(vectorizer, "vectorizer.sav")
joblib.dump(model, "model.sav")
```

```
print("Prediction")

predicted_sentiment = model.predict(test_vectors)#to predict for new data
# print(predicted_sentiment[0])

actual_pos, actual_neg = 0, 0
pred_pos, pred_neg = 0, 0

correct = 0
for i in range(len(y_test)):
    if predicted_sentiment[i]==y_test[i]:
        correct+=1
print("Accuracy",(correct/len(y_test))*100)
from sklearn.metrics import f1_score
print("f1_score", f1_score(y_test, predicted_sentiment, average="weighted"))
```

APP.PY

```
from flask import Flask, request
from flask_cors import CORS
import nltk
import emoji
import re
import functools
import operator
import pandas as pd
import joblib
import json
from nltk.stem import WordNetLemmatizer
```

```
def split_emojis(em):
    em=str(em)
    em_split_emoji = emoji.get_emoji_regexp().split(em)
    em_split_whitespace = [substr.split() for substr in em_split_emoji]
    # print(em_split_whitespace)
    em_split = functools.reduce(operator.concat, em_split_whitespace)
    emojis = []
    for separated in em_split:
        emojis.append(separated)
    return emojis

def prediction(inp):
    # if(inp == ""): return None
    def split_emojis(em):
        em=str(em)
        em_split_emoji = emoji.get_emoji_regexp().split(em)
        em_split_whitespace = [substr.split() for substr in em_split_emoji]
        # print(em_split_whitespace)
        em_split = functools.reduce(operator.concat, em_split_whitespace)
        emojis = []
        for separated in em_split:
            emojis.append(separated)
        return emojis

    wordnet_lemmatizer = WordNetLemmatizer()

    f=open('./datasets/emoji_dataset2.json')
    s=str(f.readlines())[2:-2]
```

```
# print(s)
# print(json.loads(s))
# s=json.loads(s)

jsn = json.loads(s)

i=inp

tokenization = nltk.word_tokenize(i)
v=[]
for w in tokenization:
  v.append(wordnet_lemmatizer.lemmatize(w,'v'))
i=' '.join(v)
# print(i)


mul = split_emojis(i)
s1 = []
for j in mul:
    # print(re.findall(r'[^\w\s,.]', j))
    if (len(re.findall(r'[^\w\s,.]', j))==1):
        s1.append(jsn['U+{:X}'.format(ord(j))])
    else:
        s1.append(j)

res = ' '.join(s1)

tokenization = nltk.word_tokenize(res)
v=[]
```

```
for w in tokenization:
  v.append(wordnet_lemmatizer.lemmatize(w,'v'))

res=' '.join(v)

mul = split_emojis(res)
# print("Mul: ", mul)
s1 = []
for j in mul:
    # print(re.findall(r'[^\w\s,.]', j))
    if (len(re.findall(r'[^\w\s,.]', j))==1):
        s1.append(jsn['U+{:X}'.format(ord(j))])
    else:
        s1.append(j)

res = ' '.join(s1)
# print("Res: ", res)
c=pd.Series([res])



c = vectorizer.transform(c)
#it converts to vector matrix



p = model.predict(c)
res = model.predict_proba(c)
# print(model.classes_)
res = list(res)
```

```
probabilities = {}
for i in range(len(model.classes_)):
    probabilities[model.classes_[i]] = res[0][i]
    # print(model.classes_[i], res[0][i])
if(inp != ""):
    print("\nProbabilities:")
    print(probabilities)
# prob=model.predict_proba(c)
# print("prob",prob)
p = p[0]
d = p

print("\nGraphs data:")
data1=None
data = None
with open('./resultGraph.json') as file:
    data = json.load(file)

if(inp != ""): data[p] += 1

open("resultGraph.json", "w").write(json.dumps(data))
print(data)



with open('./resultGraph1.json') as file:
    data1 = json.load(file)
```

```
    if (d == "happy" or d=="love" or d=="surprise" ): d="positive"
    elif (d == "sadness" or d=="fear" or d=="anger" ):    d="negative"

    if(inp != ""): data1[d] += 1
    open("resultGraph1.json", "w").write(json.dumps(data1))
    print(data1)



    if(inp == ""): p = "No emotion"

    return {"prediction": p, "graphData": data , "sentiment": data1}

def getGifData():
    gifs = None

    with open('./datasets/gifs.json') as file:
        gifs = json.load(file)

    return gifs

def getEmojiData():
    emojis = None

    with open('./datasets/emoji_dataset2.json') as file:
        emojis = json.load(file)

    return emojis

app = Flask("Emotion Detection")
```

```
CORS(app)

vectorizer = joblib.load("vectorizer.sav")
model = joblib.load("model.sav")



@app.route("/")
def index():
    return "Welcome to Emotion Detection API"

@app.route("/predict", methods=['POST', 'GET'])
def predict():
    if (request.method == "POST"):
        text = request.form['text']
        # print(split_emojis(text))
        data = prediction(text)
        return data
    else:
        return "This API accepts only POST requests"

@app.route("/predictGif", methods=['POST', 'GET'])
def predictGif():
    if (request.method == "POST"):
        gifid = request.form['gifid']
        gifData = getGifData()

        text = gifData[gifid]['description']
        print(text)
        data = prediction(text)
```

```
        return data
    else:
        return "This API accepts only POST requests"

@app.route("/getGifs", methods=['POST', 'GET'])
def getGifs():
    if (request.method == "POST"):
        gifs = getGifData()
        return gifs
    else:
        return "This API accepts only POST requests"

@app.route("/getEmojis", methods=['POST', 'GET'])
def getEmojis():
    if (request.method == "POST"):
        emojis = getEmojiData()
        return emojis
    else:
        return "This API accepts only POST requests"

@app.route("/getBarGraphData", methods=['POST', 'GET'])
def getBarGraphData():
    if (request.method == "POST"):
        data = None

        with open('./resultGraph.json') as file:
            data = json.load(file)

        return data
```

```
    else:
        return "This API accepts only POST requests"

@app.route("/getBarGraphData1", methods=['POST', 'GET'])
def getBarGraphData1():
    if (request.method == "POST"):
        data = None

        with open('./resultGraph1.json') as file:
            data = json.load(file)

        return data
    else:
        return "This API accepts only POST requests"

if __name__ == '__main__':
   app.run(debug = True, port=7676)
```

6. RESULTS

Result graph after training and testing the dataset with the SVM model

Fig 6.1 Result analysis for dataset

6.1 INPUT AND OUTPUT

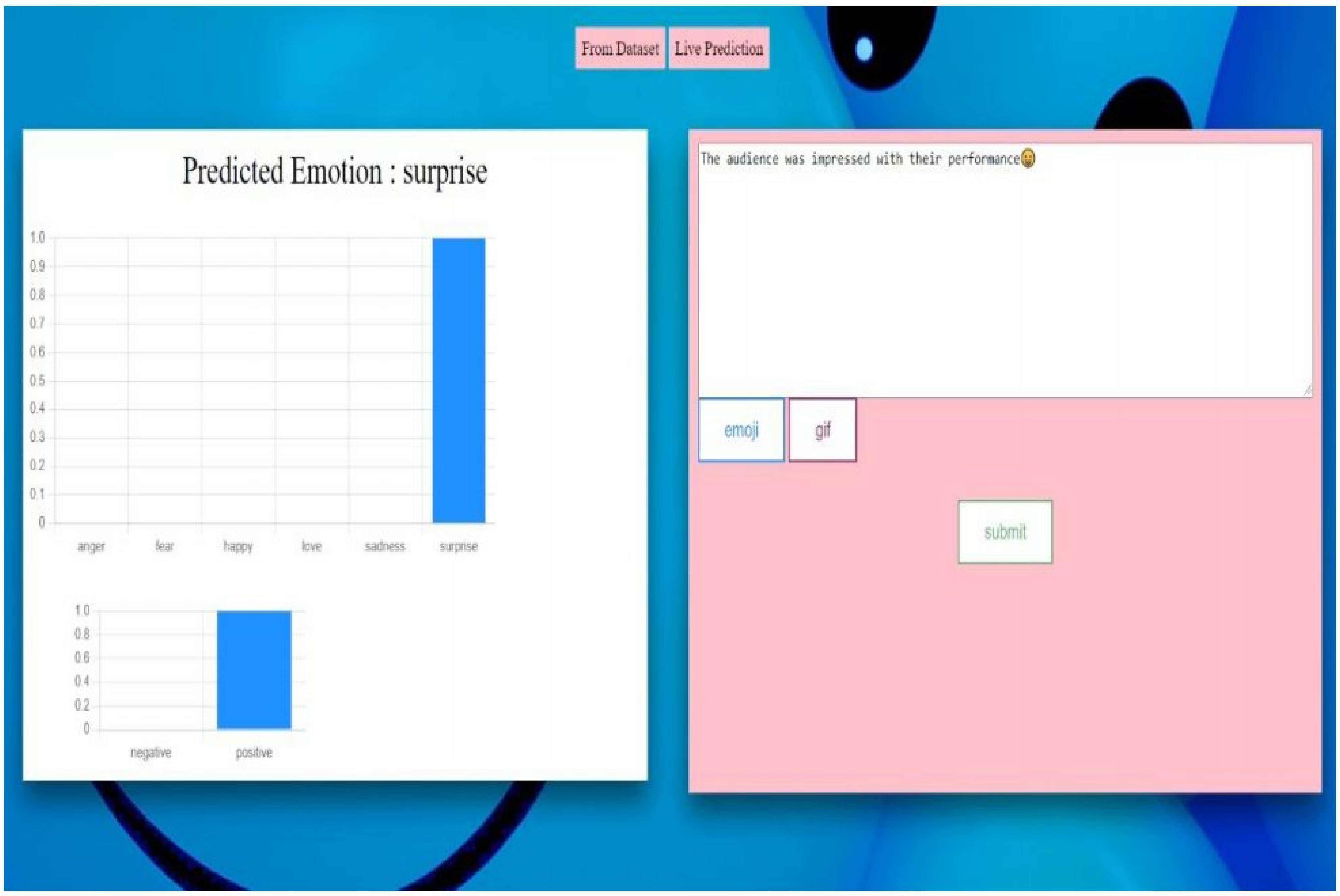


Fig 6.1.1 Input and output for text

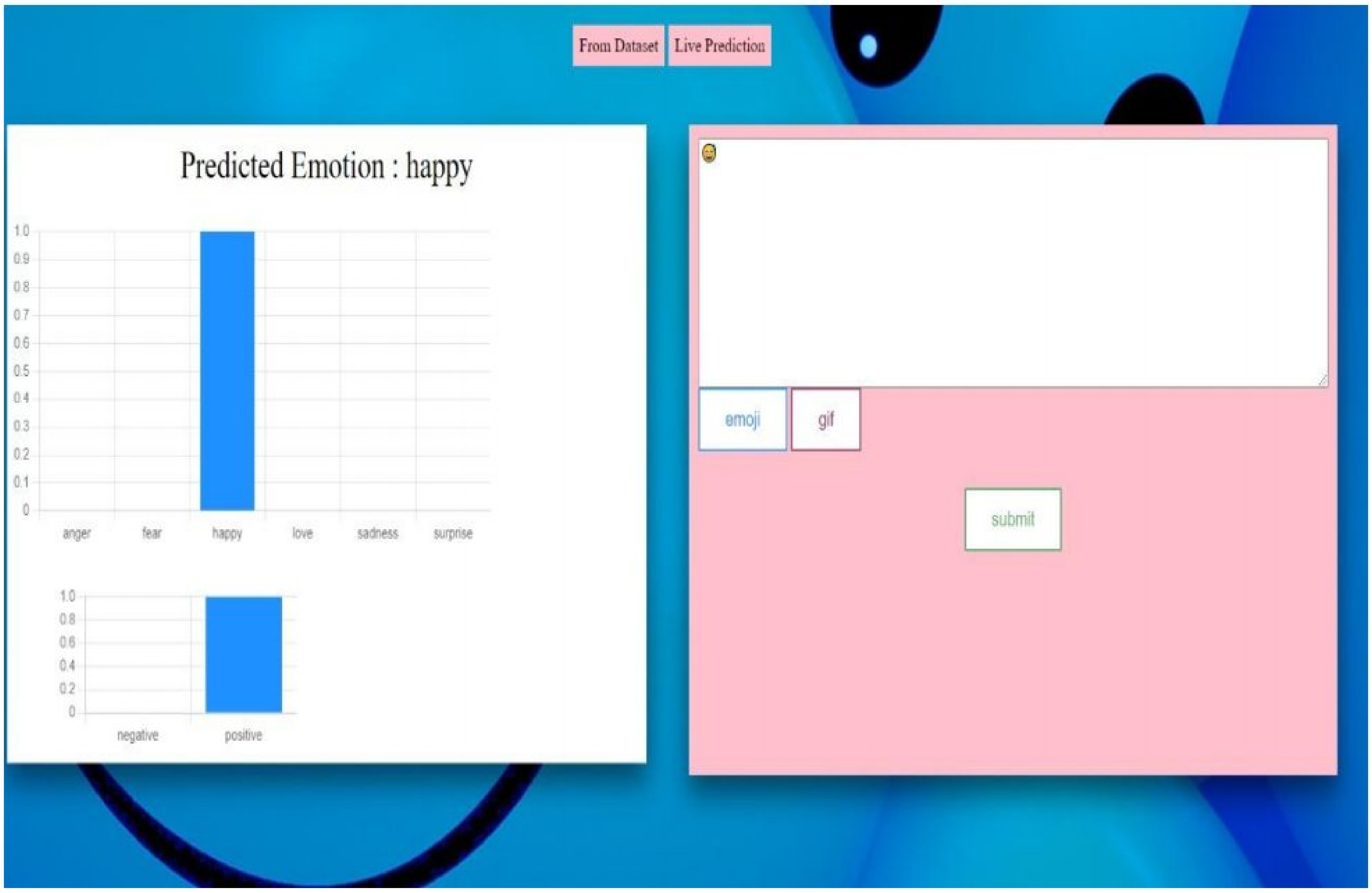


Fig 6.1.2 Input and output for emoticon

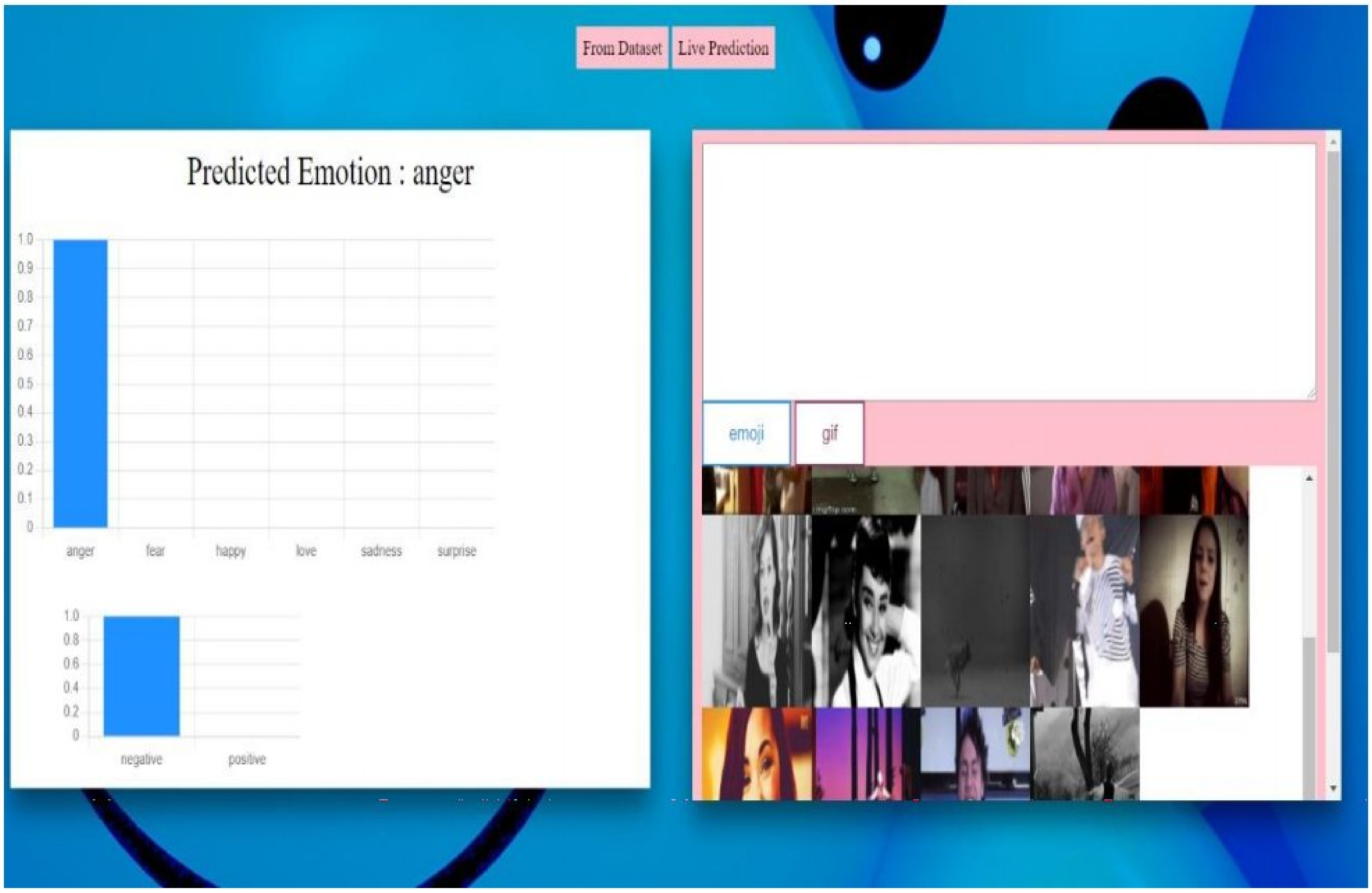


Fig 6.1.3 Input and output for gif

6.2 PERFORMANCE MEASURE

For Performance measure we can use the following to find whether our model is accurate or not

Accuracy: Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by
dividing the number of correct predictions by the number of total predictions.

accuracy=correctpredictions/allpredictions

the accuracy for our model is :

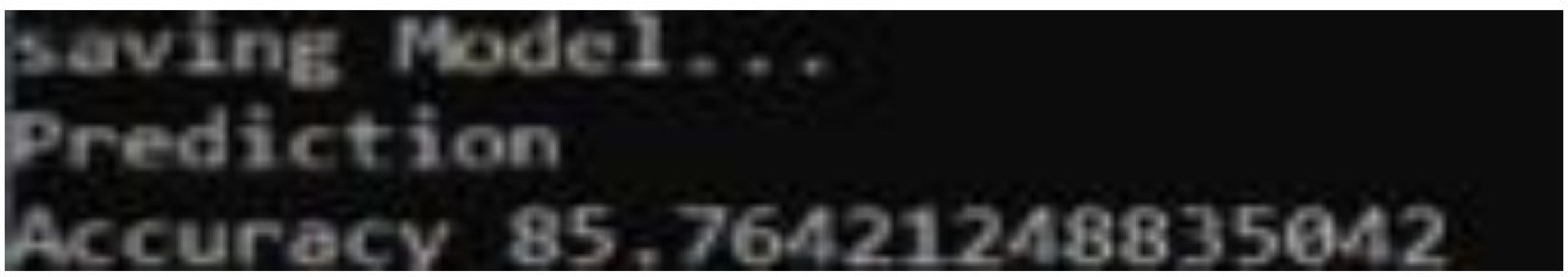


Fig 6.2 Accuracy measure

7. CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

In this paper, we discuss about how to predict the desired sentiment and emotion using sentiment and emotion analysis using machine learning algorithm. Here we predict the emotion expressed in the form of text, emoticon and gif given by the user. In this we used Support Vector Machine classification model and it gives 85.7642% accuracy while compare to other algorithms that are used previously like naive Bayes, SVM is fast and gives accurate results compared to other algorithms. We firmly conclude that implementing sentiment analysis and emotional analysis using these algorithms will help in deeper understanding of textual data and visually represented data, which can essentially serve a potential platform for businesses.

7.2 FUTURE WORK

In future work , we aim to handle text in images, dive deep into emotional analysis to further detect idiomatic statements. We will also explore richer linguistic analysis.

8. REFERENCES

[1] Akshay Chavhan , Prof. Sneha A. Khaire , Ankit Kumar , Saurabh Mate ,Vipul Thakare(2020) Sentiment Analysis Using Machine Learning for Twitter, Department of Information Technology, SITRC, Nashik, India.

https://www.academia.edu/43433185/Sentiment_Analysis_Using_Machine_Learning_for_Twitter

[2] Shilpi Chawla, Gaurav Dubey, Ajay Rana, Product Opinion Mining Using Sentiment Analysis on Smartphone Reviews.

Shilpi Chawla, Gaurav Dubey, Ajay Rana, Product Opinion Mining Using Sentiment Analysis on Smartphone Reviews.

[3] Ankush Mittal, Amarvir Singh (2017) Sentiment Analysis Using N-GRAM Algo and SVM Classifier, Department of Computer Science, Punjabi University, Patiala, India.

http://www.ijcrt.org/papers/IJCRT1704093.pdf

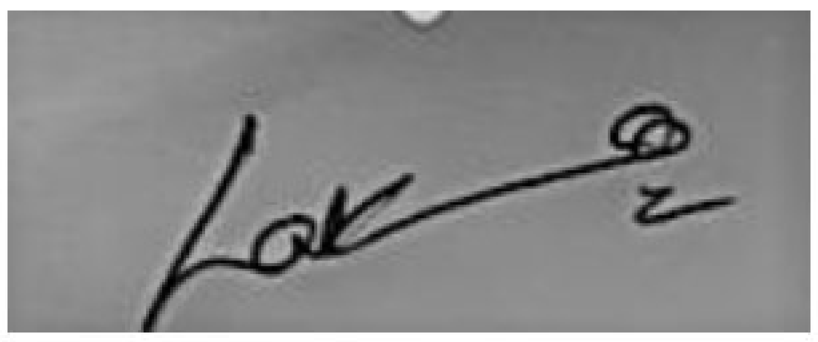
[4]Anitha, Dr.S.Sivakumar, A COMPARATIVE STUDY AND SENTIMENT ANALYSIS USING KONSTANZ INFORMATION MINER IN SOCIAL NETWORKS.

https://www.ijrar.org/viewfull.php?&p_id=IJRAR19K4870

[5]Shanshan Yi,Machine learning based customer sentiment analysis for recommending shoppers, shops based on customers' review. Chengdu University of technology,China

https://link.springer.com/article/10.1007/s40747-020-00155-2

Project Guide:



S.V.S.S. Lakshmi

(Assistant Professor)

Computer Science Engineering

Anits

V.S.S. Gowri	317126510176	
S.Pratyusha	317126510160	
S. Nidhin Prasad	317126510162	

Product Opinion Mining Using Sentiment Analysis on Smartphone Reviews

Shilpi Chawla, Gaurav Dubey, Ajay Rana

Amity University Uttar Pradesh, Noida, India

Shilpichawla13@gmail.com, gdubey@amity.edu, ajay_rana@amity.edu

Abstract: **An unexpected increase in the application of social web sites has led to the need in development of a very robustful and trust worthy systems and varied machinery forms in order to analyze such large forms of data and their ware houses which is being collected from various industries. We are always eager and excited to know what people think what they feel and perceive about various aspects of living and non-living beings. In need to understand and analyze various traits of behavior and the varying personality there is a need of opinion mining. It is a collection of all the extracted information from all the available resources and possible instances which could be in the form of hidden emotions, paragraphs and could be text , urban language and various un-identified representations. It is not only related topics like sensex, politics, finance and other meaningful words , but represents many things in the wide area of application. Almost all the sites have provided the option to display and present various medias and their views on various real life events. They can share various aspects of philosophy too and many more. They can represent various opinions on various streams of life which has really become crucial in our daily life to analyze the pattern in which things are exponentially growing. This research paper provides you with sentimental analysis of various smart phone opinions on smart phones dividing them Positive, Negative and Neutral Behavior. This is basically being obtained by studying the various posts being posted by varied number of users considering their areas of interest categorizing the smart phones. Analysis of plenty of words coupled in a sentence represent various sentiments of users and the various experiences and impact that product has given them. This analysis compiles a structural modeling approach and Bayesian Interface system to identify the polarity of the opinion which subsequently classifies positive and negative opinions.**

Keywords: **Text Mining--Opinion mining--sentiment analysis--text analysis--Sentiment Mining.**

I. INTRODUCTION:

1. Sentiment Classification

This type of classification also known as Polarity Classification . It is used to analyze in ample amount of text in which every sample is being labeled as either positive, negative or neutral sample depending on the overall response received that is being expressed in that particular text.

This type of classification can be carried out at various levels to gain surety over the produced opinion on the set of texts.

The higher level of classification in the pyramid will be the difficult level amongst all the levels as it would be having the opinions as well as the things.

1.1. Document Level Sentiment Classification:

At this level of classification, the sentiment is being analyzed in the whole document and is categorized as positive, negative or neutral or is just objective. In this level of classification , the utmost challenge is to extract that subjective test which could be inferred in the complete sentiment of the overall document.[2]Thus for polarity classification it is supposed that the document should be focused on a single object and should contain opinion from a single holder so as not to have any conflicts in the opinions of various opinion holders.

1.2. Sentence Level Sentiment Classification:

At this level of sentiment analysis the sentiments are being analyzed in the text appearing at the sentence level. It is a considered to fined grained classification as compared to the classification made at document level in which the polarity was expressed only in three forms either positive, negative or neutral format.[2] This type of sentence classification can be done in the below mentioned 2 ways- Grammatical Syntactic Approach or semantic approach . [33] The former approach considers the grammatical hierarchy or structure of the various sentences occurring in the text by taken into consideration various parts of speech. And the latter one counts the occurrences of positive as well as negative words in order to propose the polarity of the analyzed sentence.

1.3. Feature Level Sentiment Classification:

Product features or attributes referred to as components are being analyzed under this classification covering the sentence level as well as document level.Opinion is being derived from the already extracted features.

<u>General flow of the complete process-</u>

978-1-5090-3012-5/17/$31.00 ©2017 IEEE

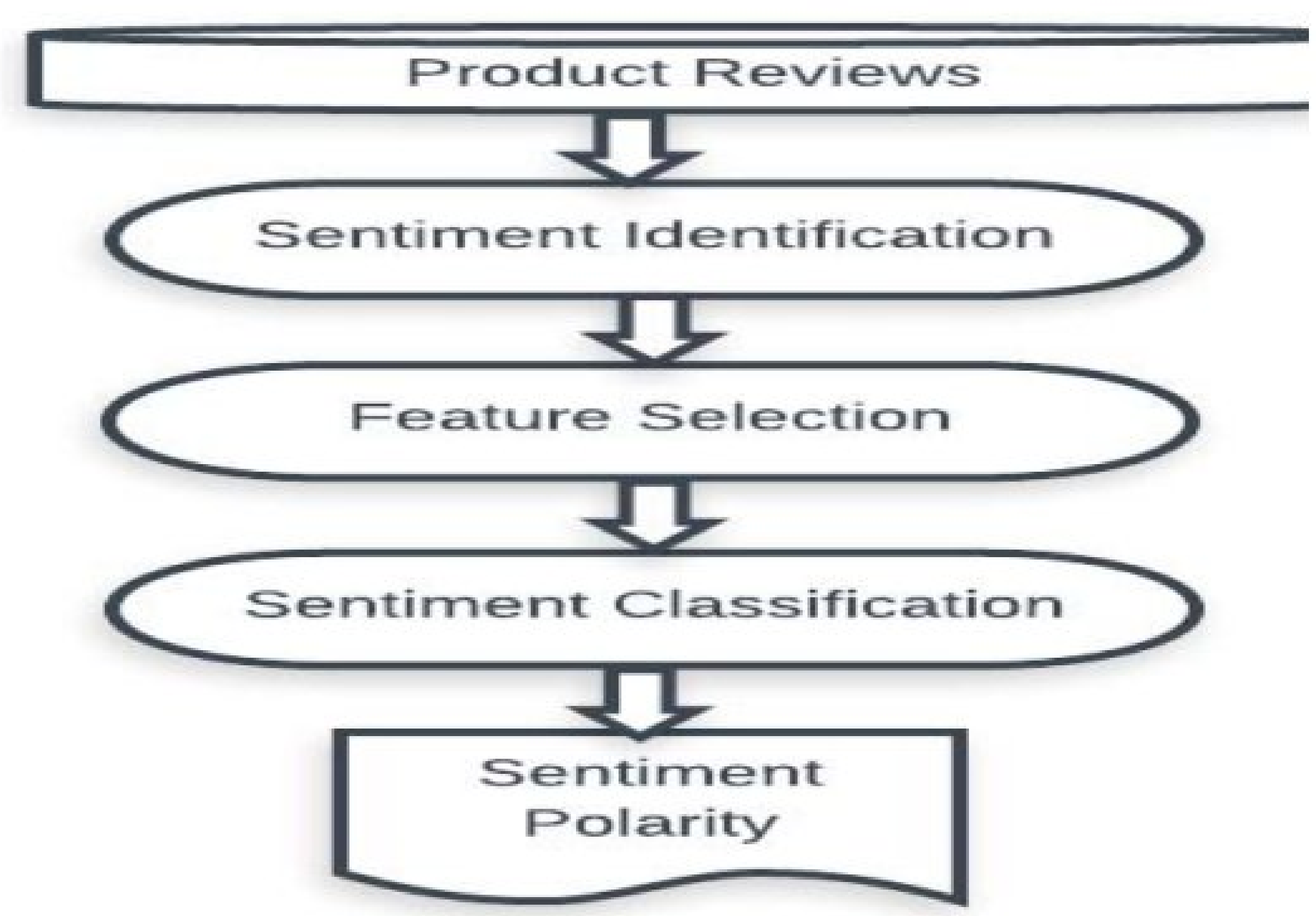


Figure 1: Sentiment Analysis process on product reviews.

Sentiment analysis has following features

 Term presence and their frequency:
Includes UNI-grams or N-grams along with their frequency of the presence. In terms of Moview reviews, the concluded sentiment analysis by ,Pang et al. [14] was that uni-grams always give better performance results as compared to that than of bi-grams. Moreover, Dave et al. [16] concluded that bi-grams and tri-grams proved to give much higher polarity classification in case of product reviews.

 Part of speech information:
Tagging specific words to their respective POS tags supports in disambiguating manner which gives help in feature selection [17]. Every word in a sentence is being assigned a label which represents the role or the position held by that word in the grammatical sense. Also, POS Tags could be helpful in identifying adjectives and various adverbs which can be rigorously used in sentiment indicators or classifiers [13].

 Negations:
Negative words have the potential to reverse the selected sentiment so they should always be taken into consideration[17].

 Opinion words and phrases:
Phrases or words(also known as opinion words) are those words which represents some positive or negative sentiments. In order to detect the semantic orientation of the mentioned words in the sentences , Lexicon and statistical based methods are the best ones. Taking an example of WordNet which was used by Hu and Liu et al. [4] to examine the polarity of the various extracted adjectives. Machine Learning Techniques widely uses Maximum Entropy , Support Vector Machines and Naïve Bayes Algorithms and have achieved success in the categorization of text.

Commonly used learning techniques in the natural language processing area are the K nearest neighborhood, winnow classifier, centroid classifier, ID3, C5, and the N-gram model.

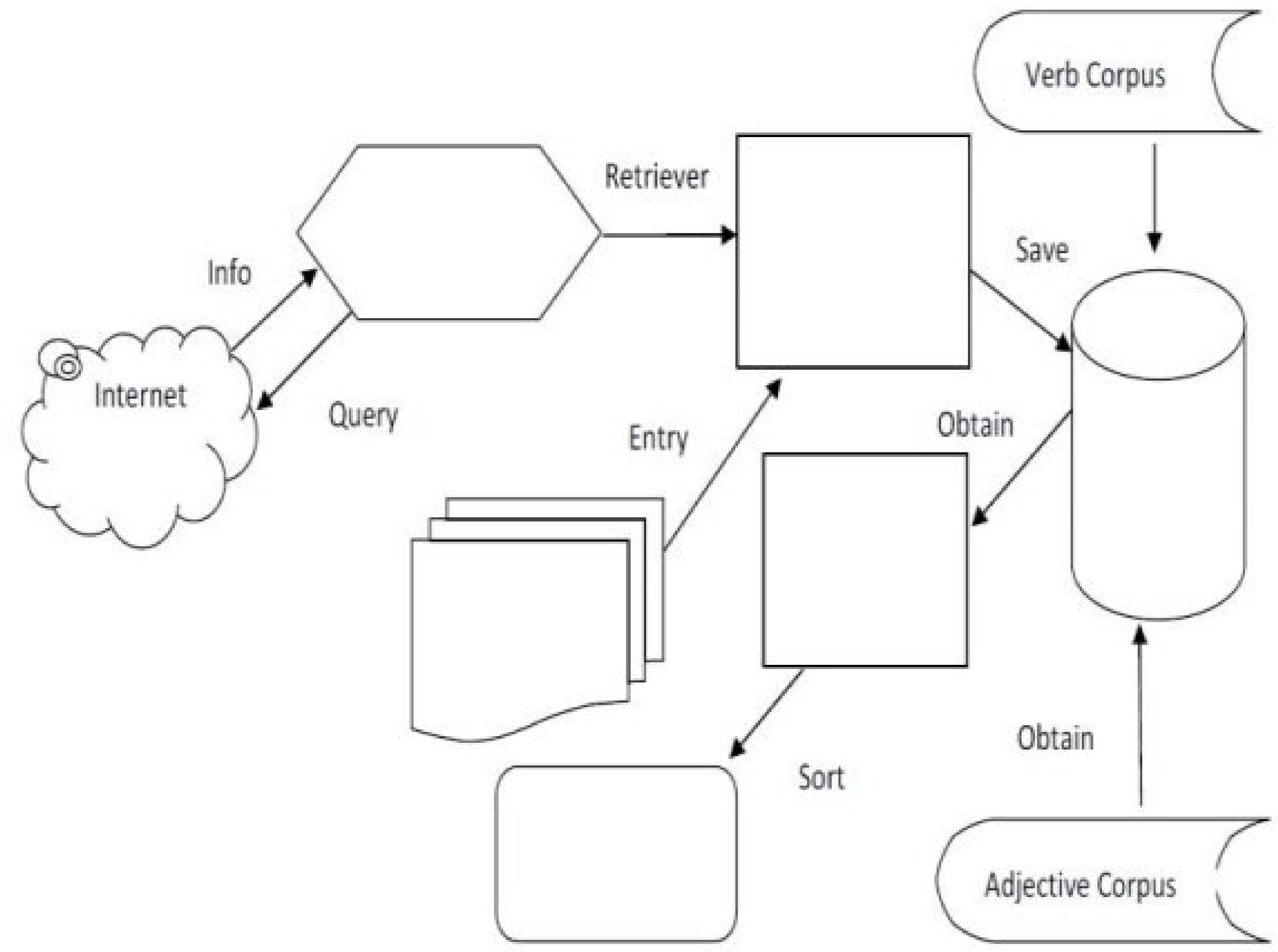


Figure 2: Opinion Mining Process.

II. RELATED WORK

The word "sentiment" which is used in context to the automated analyzing of the text which is being evaluated along with the conclusion or the judgment in terms of analytics that has already appeared in a paper published in 2001 by "Das" and "Chen" [2]. On the further evaluation and research , the same was opted and enhancement was made in it by "Turney" [3] and "Pang et al [4]. In the upcoming years the same concept was adopted by various other researchers as well ie "Nasukawa" & "Yi" [5] and "Yi et al." [6]. Above mentioned events collaboratively can summarize the effect and importance of "the sentiment analysis" among various domains and communities which are basically focused on NLP(Natural Language Processing). A numerous number of papers have been published on the term "sentiment—analysis" which are specialized in classifying, collecting and categorizing the reviews collected by customer in two – either positive or negative. Sentiment analysis in all the published papers have been carried out at almost all the levels of categorization ie document, sentence, and attribute levels.

Supervised Learning Techniques can be applied to "Naïve Bayes", "Support Vector Machine" and many more. As mentioned in [7] , the authors have tried to provide a comparison of "Naïve Bayes" and "Support Vector Machines" with the use of an Artificial Neural Network Methods in terms of document level classification of sentiment analysis.

The authors or the researchers contributed to the comparison of the sentiment – classification – literature survey-

a) Dominant comparison survey along with an efficient approach which computationally proposed using an ANN methodology in the same respect (SVN and NB)
b) a mixture of real life contexts including the positives and the negatives in un-equal.
c) A dataset of new movie reviews was used to produce an evaluation based on performance using ANN approach [1].
Many supervised methodology was used in a classic manner for the selection of features and conducted various experiments manually using ANN [1] , along with the reviews which were short listed from AMAZON web-site in Cameras, GPS and Books.
They have found that ANN has performed tremendously outstanding SVM specially when the input data set is highly un-balanced in respect to the accuracy of the classification of the obtained dataset of various smartphones reviews ((Pang & Lee, 2004).
When the balanced data set was referred that ANN was over SVM with approx 13 test cases , while SVM was over ANN in only 2 test cases.
Overall conclusion is that ANN has over achieved the accuracy in classification in all the data sets. SVM in the experiments has been graded as less effected by the Noisy data as compared to ANN when there is an increase in the imbalance.
"Pang et al. [4]" has carried out three commonly used Machine learning Methods ie Maximum Entrophy , Naïve Baiyesian and SVM on the data that is being collected from smartphones present on Internet.
The extracted ratings were being classified into 3 broad chunks: the positive ratings, the negative ratings, and the neutral ones too. Moreover the concentration was more on the pros and the crons. They accepted the standard chunks of the feature framework.
A far as performance is considered , worst results are been produced by Naïve Bayes Theorem while the SVM's give the best results, more over the differences are not that large.
A Supervised Learning can too be converted into an Un – supervised Supervised learning method as reported in [3,8] by many researchers.
In the paper presented [3] authors have used an un-supervised learning algorithms to classify an opinion | result | review as being recommended or not.
Algorithm has been widely divided into 3 main broad categories –

Step-1:

Extracting of those phrases or idioms in the sentences that represent an adverb or adjective.
In many direction there may be some isolated form of adjectives which may indicate in a subjective manner and could be insufficient enough to find in an semantic manner.

Step-2:

Usage of PMI-R algorithm in order to estimate the orientation (semantic analysis) of the phases that were extracted in step-1

Step-3:

Last step is to find out the average of the various semantic orientation found out in the second step in the mentioned review and the same is classified as positive or negative review depending on the recommendation.
In the paper [8] , the researchers proposed a learning approach in a depth with the aim to extract the way to extract a represent able format for each review using an un-supervised learning method.
Algorithm based on finding some intermediate format of representing the whole structure in an hierarchical manner. The data set has been taken from a very well – known famous site ie Amazon.com which proposed rules in lacs, out which around 50 products were reviewed and were being categorized as a positive or a negative recommendation.

III. DATA SET

A diverse data set was used to evaluate the results using Naïve Bayes Technique. The following set of data was obtained from a very reputated merchant like Amazon that act as a epic for opinion mining.
Collected various beneficial customer reviews about various smart phones including One Plus, Micromax, Samsung, Nokia ,Lava and many more.
About 1000 reviews collected including positive negative and the rest of them were neutral ones. These reviews consists of approximately 3000 sentences
We have deleted some inconsistent data and annoying data and sentences from the set to be more precise and specific.

Table 1: depicts most of the typical features and the sentences number in the corpus.

	Samsung	MI	OnePlus
Reviews	200	500	300
Sentences	600	1500	900

Table1: Representing The count of reviews and corresponding sentences.
Total distribution of the data set in terms of positive and negative reviews can further be classified as-

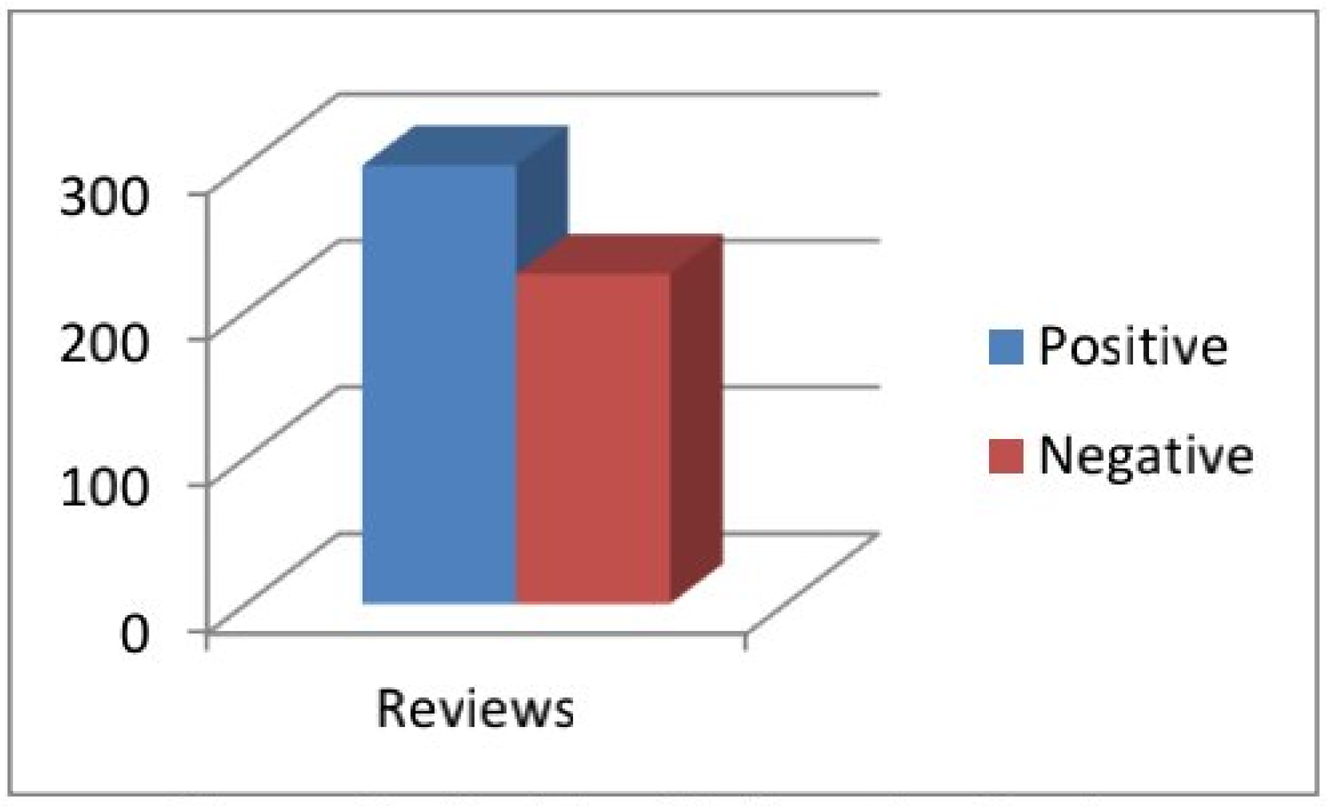


Figure-3 : Positive Vs Negative Reviews.

5. Implementation:

Flow chart of the process done can be summed by including various steps-

After installing all the necessary libraries and packages needed to execute the task following are the actions done-

Step1:

Set up the Current working directory in which R is being installed and activate the current session.

Read the CSV file which consists of all the reviews.

Step2:

Generate the factor type of the collected data that is categorized under positive and negative.

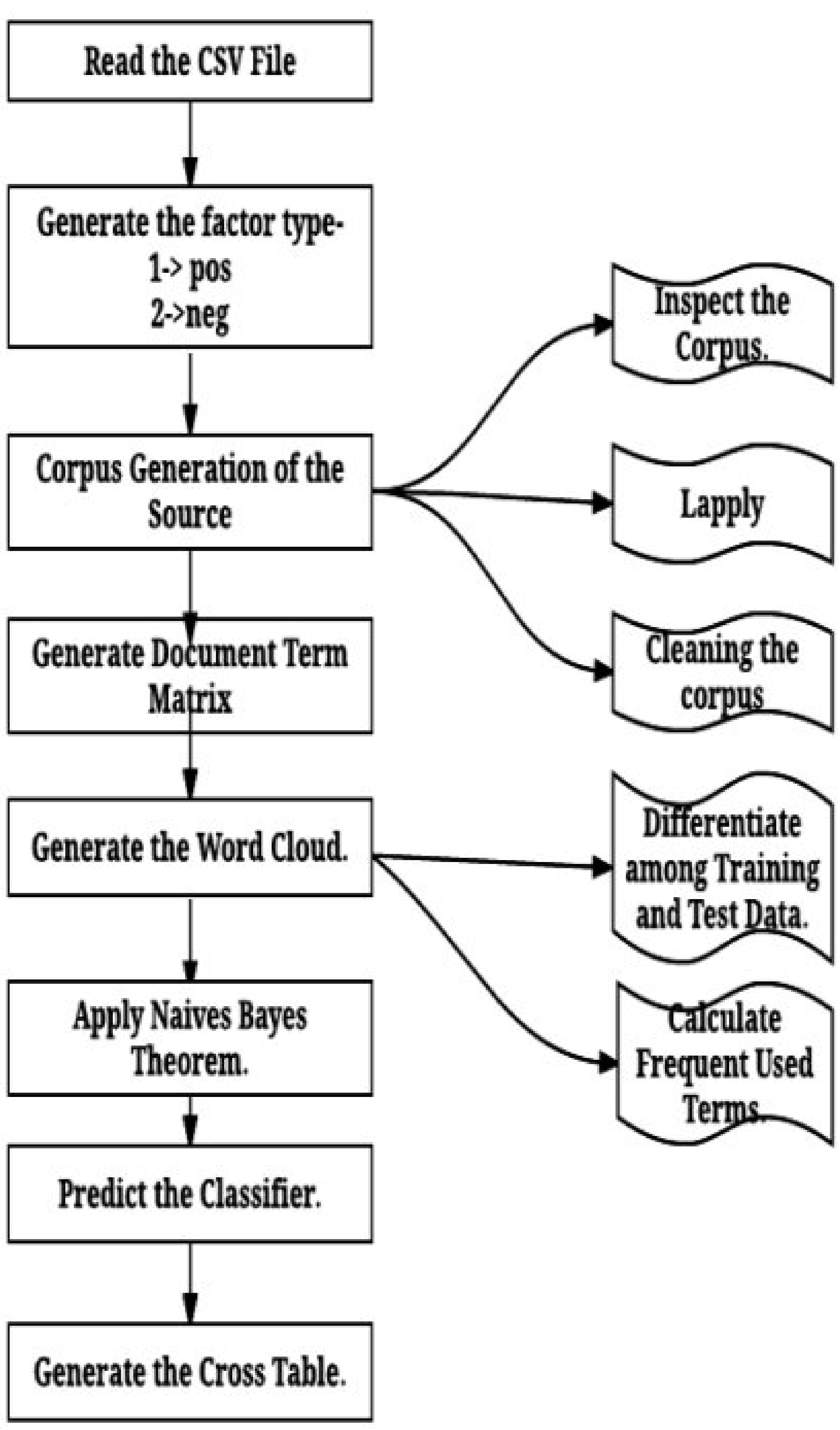


Figure 4: Flow of the process.

Step3:

Used Corpus based approach of Lexicon Tree to go through the sentiment analysis process.

- Inspecting the corpus
- Lapply to the corpus
- Cleaning the generated corpus(by removing punctuation marks, white spaces, duplicates and null values)

Step4:

Generate the Term Document Matrix which seems like as-

<<DocumentTermMatrix (documents: 524, terms: 1727)>>

Non-/sparse entries: 7380/897568

Sparsity : 99%

Maximal term length: 25

Weighting : term frequency (tf)

Step5:

Form the Word Cloud depending on the frequency of the words encountered in the textual data. Applying rainbow colors to the will give an attractive feel to the user.

Figure 4: Word Cloud from for Document Matrix.

Creating the subsets for Training Data and Testing data. The data set assigned training should be well equipped to evaluate to conclude the relevancy of the data set.

Accumulated Word cloud for Negative reviews with a minimum frequency of 6 is as-

Figure 5: Word Cloud for Negative Reviews

Word Cloud for Positive Reviews with a minimum frequency of 5 -

Figure 6: Word Cloud for Positive Reviews

```
"bad"      "get"       "good"         "now"          "oneplus"
"review"   "used"      "using"        "will"         "battery"
"charge"   "day"       "life"         "long"         "nice"
"phone"    "user"      "amazing"      "bought"       "got"
"time"     "via"       "always"       "android"      "buy"
"camera"   "can"       "design"       "experience"   "fast"
"first"    "front"     "give"         "great"        "happy"
"makes"    "money"     "one"          "product"      "provide"
"ram"      "software"  "worth"        "box"          "instead"
"budget"   "decent"    "even"         "light"        "like"
"look"     "perfect"   "performance"  "phones"       "price"
```

Figure 7: Words with minimum frequency of 20 appearing in the Document Term Matrix

Step 6:

Application of Naïve Bayes theorem for textual classification.

It depends on the posterior probability of any class selected which designates the distribution of the words in the referred document.

$$P(label|features) = \frac{P(label) * P(features|label)}{P(features)}$$

$$P(label|features) = \frac{P(label) * P(f1|label) * \ldots\ldots * P(fn|label)}{P(features)}$$

Step 7:

Generate the cross table to compare the actual values and predicted value producing the Row as well as column total.

Predicted/ Actual	Negative	Positive	Row total
Negative	5	45	50
	0.1	0.9	0.305
	0.078	0.45	
positive	59	55	114
	0.518	0.482	0.695
	0.922	0.55	
Column Total	64	100	164
	0.39	0.61	

Figure 8: Cross table of predicted and actual values using Naïve Bayes Classifier

According to the shown classifier the model produced an accuracy of around 40% which is not that much satisfactory to mine a data with this kind of classifier.

In order to produce much more refine results , we have opted another approach to classify our unstructured data.

So SVM (Support Vector Machine) approach which is one step ahead the Nave Bayes Classifier and overcomes all the drawback of the existing classifier.

On the existing data , we have applied another classifier which works as follows-

Document Term Matrix Formed for SVM is-

<<DocumentTermMatrix (documents: 524, terms: 1735)>>
Non-/sparse entries: 7656/901484
Sparsity : 99%
Maximal term length: 16
Weighting : term frequency (tf)

The training model this formed after having a ratio of around 70:30 against the training data as wellas the test data is as follows-

Call:
svm.default(x = container@training_matrix, y = container@training_codes,
kernel = kernel, cost = cost, cross = cross, probability = TRUE,
method = method)

Parameters:
SVM-Type: C-classification

SVM-Kernel: linear
cost: 1
gamma: 0.0005763689

Number of Support Vectors: 172

SVM_LABEL	SVM_PROB
	Min. :0.5160
	1st Qu.:0.8752
	Median :0.9619
	Mean :0.9068
neg: 51	3rd Qu.:0.9799
pos:174	Max. :1.0000

Cross table thus generated between the actual and predicted values is as follows-

Predicted/ Actual	Negative	Positive	Row total
Negative	49	5	54
	0.907	0.093	0.241
	0.662	0.033	
positive	25	145	170
	0.147	0.853	0.759
	0.338	0.967	
Column Total	74	150	224
	0.33	0.67	

Figure 9: Cross table of predicted and actual values using SVM Classifier

The accuracy of the model thus produced is 90% which is far better than that of Naïve Bayes Classifier.

IV. CONCLUSION

With this data set we have just tried to examine an unstructured Smartphone reviews that are being collected from a very well-known marketing site termed as Amazon. The data set was being classified amongst the classes that is the positive reviews and the negative reviews. The data set had been under gone through two well-known machine learning techniques ie Naïve Bayes and Support Vetor machine Classifier. The overll accuracy of the classifier thus trained using Naïve Bayes Classification technique was around 40% which was quite un-satisfactory to deal with and to rely on. So in order to have a reliable trained model which is able to classify the data according to our needs we have opted another approach which is SVM. The same data set which when parsed under this approach [produces good result with an accuracy of around 90%.Hence the SVM approach seems to be much better and reliable to work with the mining of data.

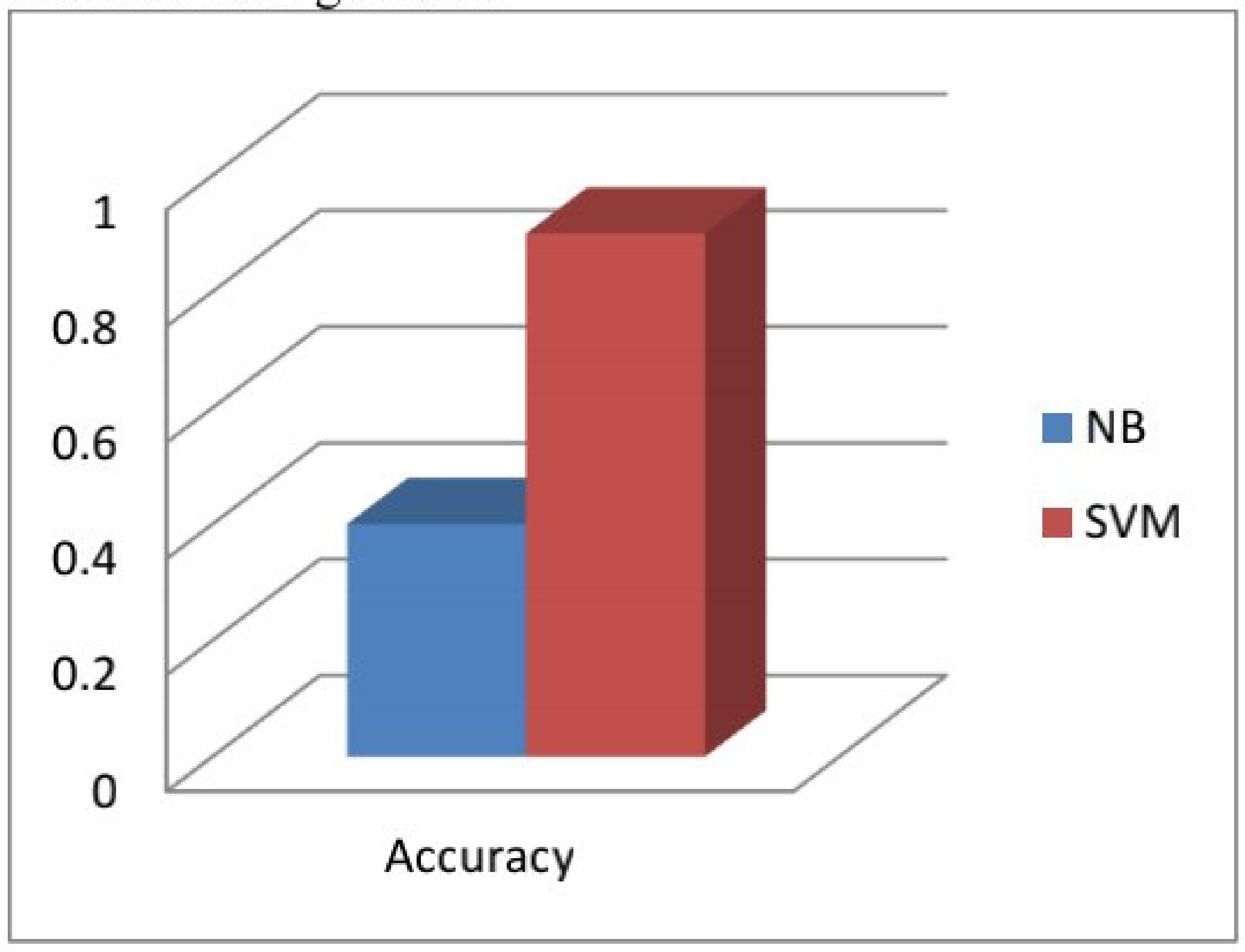


Figure 10: Comparison of the accuracy of Naïve Bayes Vs SVM

References

[1] Luciano Barbosa and Junlan Feng. 2010. Robust sentiment detection on twitter from biased and noisy data.Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pages 36–44.

[2] Alec Go, Richa Bhayani, and Lei Huang. 2009. Twit-ter sentiment classification usingdistant supervision. Technical report, Stanford.

[3] B.Pang and L. Lee. 2004. A sentimental education: Sentiment analysis using subjectivity analysis using subjectivity summarization based on minimum cuts. ACL.

[4] Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining.Proceedings of LREC.

[5] Bing Liu, Minqing Hu and Junsheng Cheng. "Opinion Observer: Analyzing and Comparing Opinions on the Web." Proceedings of the 14th International World Wide Web

[6] B.Pang and L. Lee. 2004. A sentimental education: Sentiment analysis using subjectivity analysis using subjectivity summarization based on minimum cuts. ACL.

[7] Bing Liu, Minqing Hu and Junsheng Cheng. "Opinion Observer: Analyzing and Comparing Opinions on the Web." Proceedings of the 14th International World Wide Web conference (WWW-2005), May 10-14, 2005, Chiba, Japan.

[8] Zhongwu Zhai, Bing Liu, Hua Xu and PeifaJia. "Clustering Product Features for Opinion Mining." Proceedings of Fourth ACM International Conference on Web Search and Data Mining (WSDM-2011), Feb. 9-12, 2011, Hong Kong, China

[9] Aue, A. and M. Gamon.Customizing sentiment classifiers to new domains: a case study.In Proceedings of Recent Advances in Natural Language Processing (RANLP-2005), 2005.

[10] Blitzer, J., M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In Proceed-ings of Annual Meeting of the Association for Computational Linguistics (ACL-2007), 2007.

[11] Bo Pang and Lillian Lee , Opinion Mining and Sentiment Analysis , Computer Science Department, Foundations and Trends in Information Retrieval Vol. 2, No 1-2 (2008) 1–135, 2008 , Cornell University, Ithaca, NY 14853, U.S.A.

[12] Daniel Lowd and Pedro Domingos , Naïve Bayes Model for Probability Estimation , Department of Computer Science and Engineering, University of Washington, Seattle, WA 98195-2350, USA.

[13] Behrooz Parhami , IEEE Transactions on reliability , Voting Algorithms, Vol. 43. , No. 4 , 1994 December , University of California , Santa Barbara.

[14] Simon Tong , Support Vector Machine Active Learning with Applications to Text Classification , Journal of Machine Learning Research (2001) 45-66 , Computer Science Department Stanford University Stanford CA 94305-9010, USA

[15] Vincent J. Della Pietra , Adam L. Berger , Stephen A. Della Pietra , A Maximum Entropy Approach to Natural Language Processing(1996).

[16] Guo-Xun Yuan , Chia-Hua Ho and Chih-Jen Lin , An Improved GLMNET for L1-regularized Logistic Regression(2011).

[17] S. B. Kotsiantis , Supervised Machine Learning: A Review of Classification Techniques , Informatica 31 (2007) 249-268 , Department of Computer Science and Technology.

SENTIMENT AND EMOTIONAL ANALYSIS FOR TEXTS, EMOTICONS AND GIFS USING MACHINE LEARNING ALGORITHMS

[1]S.V.S.S. Lakshmi, [2]V.S.S. Gowri, [3]S. Pratyusha, [4]K. Meenakshi ,[5]S. Nidhin Prasad

[1]Assistant Professor, [2,3,4,5] Student

Department Of Computer Science & Engineering,

Anil Neerukonda Institute Of Technology and Sciences, Visakhapatnam, India.

ABSTRACT

We use technology for our advantage by applying it in almost every aspect of our lives. Nowadays as people spend a great deal of time on internet, they tend to involve more in expressing their opinions, communicating with each other through various social media or business platforms. The fundamental aspect of our lives is that other's opinions and interests are important to us. Although measuring feelings and emotions can be quite difficult, there are still ways. Sentiment analysis and Emotional analysis is the computational study of sentiments and emotions expressed through various platforms by giving two different insights about the user. Sentiment analysis classifies user data into categories like positive and negative. Emotional analysis aims to understand the emotion conveyed by the user. The use of text is not new, but as people are opting simpler and quicker ways to communicate, the use of emoticons and gif became popular. This paper reflects the idea of taking user opinions expressed through text, emoticon and gif by performing sentiment, emotion analysis and establishing conclusions by using machine learning algorithm. This analysis essentially serves a gateway to improve consumer service, increase sales revenue and generates growth opportunities in businesses.

Keywords: Sentiment Analysis, Emotional Analysis, Support Vector Machines, Preprocessing, Tokenization, Lemmatization.

1. INTRODUCTION

The opinions of others matter a lot as they are an important reflection of our human behavior and have a significant influence in our daily decision-making process. Social media and other online tools had not only mediated communication in countless ways, but also a way to actively use information technologies to seek out and understand the opinions of others Sentimental and emotional analysis has served as a reliable source in day-to-day by providing insightful opinions about several products rolled over in the market, innovative ideas, people opinion about new policies framed by government etc.

Human emotions are extremely diverse and cannot be restricted to certain metrics alone. They are so complex, and reducing them to positive and negative can only give you a shallow understanding. Polarity analysis is too generic as it does not specify the actual intent of message delivered by author and just positive or negative classes are not sufficient to understand nuances of underlying tone of a sentence. This brings the need to take one step above sentiment analysis leading to emotion analysis which is based on a wide spectrum of moods rather than a couple of static categories. In this paper we throw light on methods we have used to derive sentiment analysis and how we have accomplished emotion analysis of user opinions expressed through text, emoticons and gif. Emoticons and Gifs are digital images, aims to help people convey more emotion and expression than the static written word, especially as images are processed quicker than text. A supervised learning technique provides labels to classifier to make it understand the insights among various features. Once the classifier gets familiarized with train data it can perform classification on unseen test data. We have chosen Support Vector Machine classification algorithm to carry out sentiment and emotional analysis. Emotion detection involves a wide platter of emotions classified into states like happiness, hate, love, relief, surprise and worry. We here examine sentiments and emotions of text combined with emoticons and gif.

2. PREVIOUS WORK

[1]Ankush Mittal, Amarvir Singh (2017) presented in this paper, that sentiment analysis is the efficient technique to analyze the people behavior. The sentiment analysis contains the four steps and in this work improvement in the feature extraction phase is using the pattern based technique. The proposed improvement is analyzed that execution time is reduced to 10 percent and accuracy is increased to 20 percent. [2]A. Anitha ,Dr.S.Sivakumar(2019) in this paper proposed Natural Language (NLP) based approach to enhance the sentiment classification by converting into vectors and then using methods for classification. The sentiment analysis was conducted using machine learning algorithms on Facebook data. The results of the study were interpreted by calculating accuracy. [3]Shilpi Chawla, Gaurav Dubey, Ajay Rana(2017) proposed Product Opinion Mining Using Sentiment Analysis on Smartphone Reviews in this

paper the overll accuracy of the classifier thus trained using Naïve Bayes Classification technique was around 40% which was not satisfactory ,so to have a reliable model which will classify the data according to our needs we have used another approach which is SVM. We used this approach on the data set we got good result with accuracy nearly 90%. So SVM is more reliable to use. [4]Akshay Chavhan, Prof. Sneha A. Khaire, Ankit Kumar, Saurabh Mate, Vipul Thakare proposed Sentiment Analysis Using Machine Learning for Twitter used two classifiers SVM and Naive Bayes. In this work achieved 63 percent accuracy by SVM and 58 percent accuracy by Naive Bayes classifier. [5]Shanshan Yi and Xiaofang Liu[2020] in this page developing a suitable hybrid recommendation system that can use the shopping data of customer which is in the form of reviews, the system predicts the interest of customer in buying the product in a particular shop. Hybrid Recommendation System is evaluated by Mean squared error, Mean absolute error.

3. PROPOSED SYSTEM

In this paper we have used SVM (Support vector machine) which is a supervised machine learning algorithm. This works by making a straight line between two classes. The line is called decision boundary or hyperplane. We use LinearSVC model to classify our data.

1.Data Collection:

The first step is to collect the data for predicting the emotion. We have the text dataset from kaggle.

2.Preprocessing of data:

The data is preprocessed before giving to classifier. This will convert the data into the format that is used to train the SVM model.

2.1: Tokenization:

Tokenization is the process of converting text into tokens before transforming it into vectors. In this we divide sentences into words.

2.2: Removing punctuations and special symbols:

The punctuations and symbols like !,\ are removed.

2.3: Lemmatization:

Sentences in some cases will be in the form of -ing, -ed. So, we need to remove them to get the root word so we can get correct emotion from the data.

We have also used TFID Vectorizer to convert into the required format.

3.Fit data to classifier:

Now we use LinearSVC model to fit the data.

4.Predict the test data:

Now we predict the test data by using the same model to get the emotion.

The emoticons are taken with their respective Unicode to obtain their text description. The Graphics Interchange Format, or Gifs are taken from the internet in the form of a hyperlinks with their respective text descriptions. When a gif or emoticon is selected in the input its text description is retrieved from the input file to perform preprocessing on the obtained text. The above preprocessing steps are performed on the retrieved text for further processing in order to get the desired emotion.

Results:

Now based on the input we give in the form of text, emoticon or gif it predicts the emotion and also it divides as positive or negative based on the emotion.

The results are shown in bar graph.

Accuracy:

The SVM model gives 85.7642% accuracy for the emotion prediction.

4. ARCHITECTURE DIAGRAM:

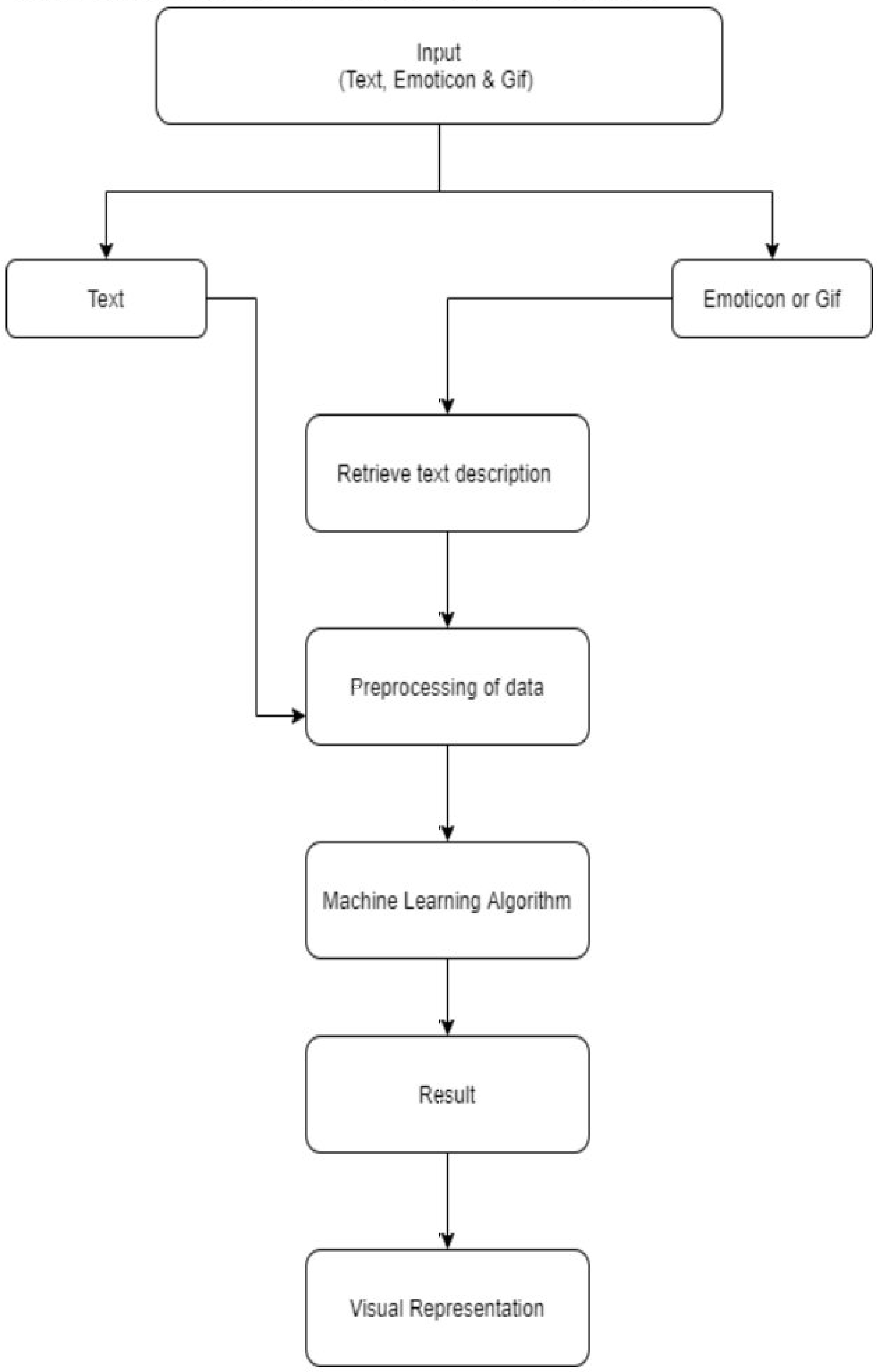


Fig.4.1 Architecture diagram for emotion prediction

Support Vector Machine:

We have used Support Vector Machine algorithm for our project. It is a supervised machine learning algorithm. It works by dividing two classes

with a straight line between them. This line is called decision boundary or hyperplane. SVM is used for both classification and

regression problems but it is usually used for classification problems. In this algorithm, the data item is plotted as a point in n dimension space having value of feature represented as the value of coordinate. After that we perform classification by finding the decision boundary that divides the classes. We use LinearSVC model for our project.

Kernel is used to transform data into a suitable form to classify and train the data. In this we use linear kernel. For calculating the probabilities per each class CalibratedClassifierCV is used in LinearSVC to use predict_proba method.

5. RESULTS

We took a text dataset for the purpose of training and testing. After training and testing the dataset with the SVM model it gives 85.7642% accuracy for the emotion prediction.

We used the same model to predict the emotion for the given input.

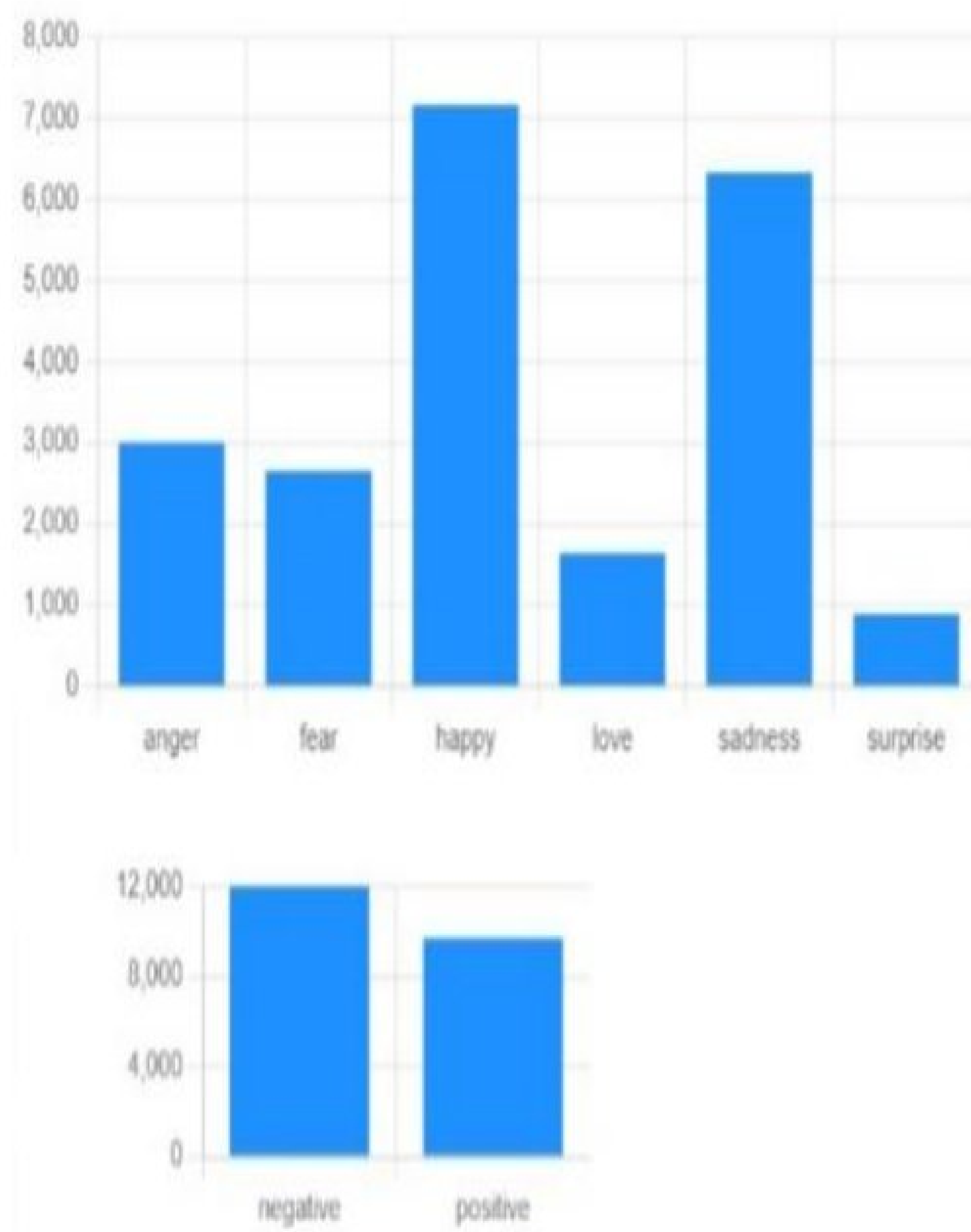


Fig 5.1 Result Graph of dataset

USER INPUT

Fig 5.2 Input given by the user

OUTPUT

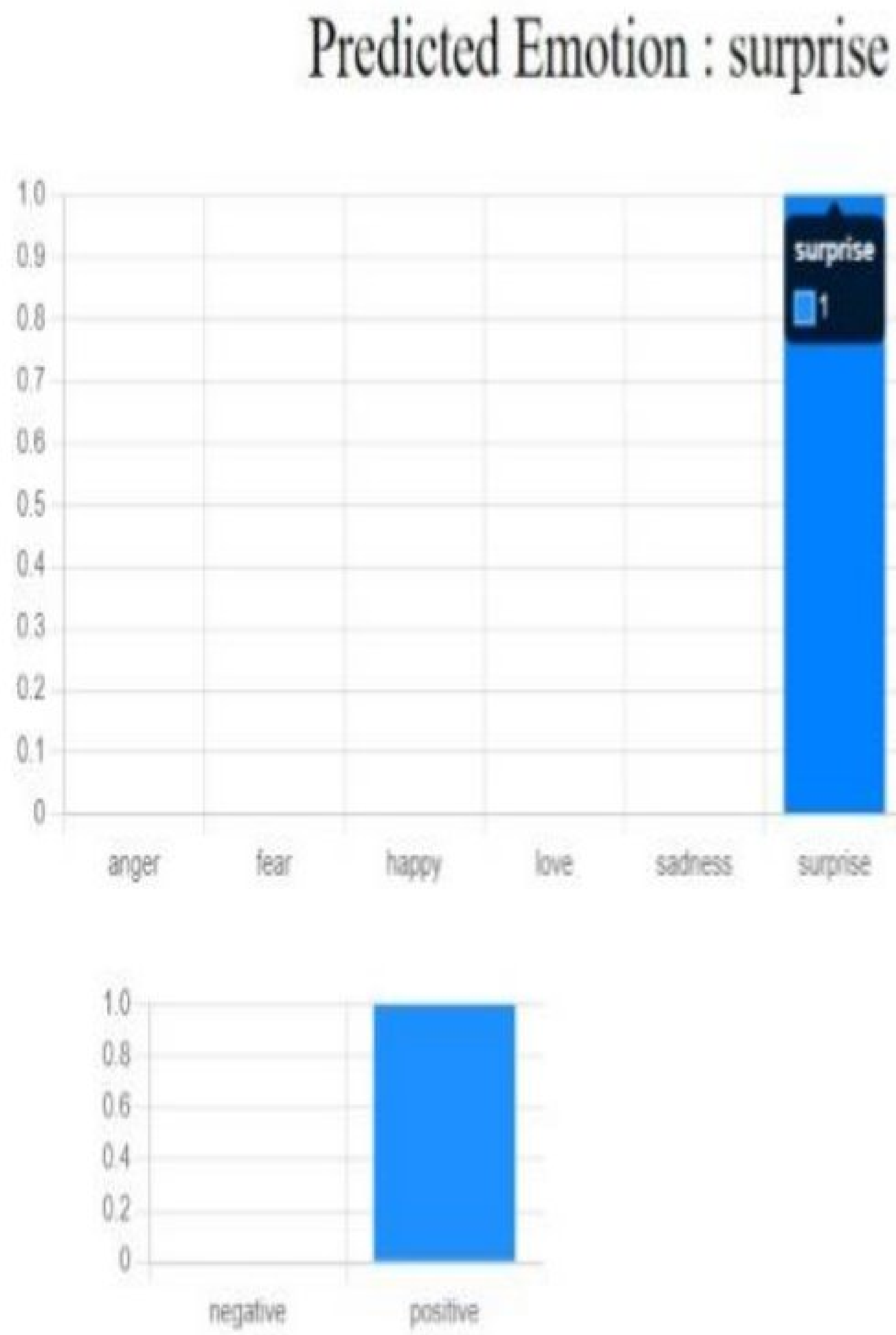


Fig 5.3 Result for user input

6. CONCLUSION:

In this paper, we discuss about how to predict the desired sentiment and emotion using sentiment and emotion analysis using machine learning algorithm. Here we predict the emotion expressed in the form of text, emoticon and gif given by the user. In this we used Support Vector Machine classification model and it gives 85.7642% accuracy while compare to other algorithms that are used previously like naive Bayes, SVM is fast and gives accurate results compared to other algorithms. We firmly conclude that implementing sentiment analysis and emotional analysis using these algorithms will help in deeper understanding of textual data and visually represented data, which can essentially serve a potential platform for businesses.

7. REFERENCES

1.Ankush Mittal, Amarvir Singh (2017) Sentiment Analysis Using N-GRAM Algo and SVM Classifier, Department of Computer Science, Punjabi University, Patiala, India.

http://www.ijcrt.org/papers/IJCRT1704093.pdf

2.Anitha, Dr.S.Sivakumar, A COMPARATIVE STUDY AND SENTIMENT ANALYSIS USING KONSTANZ INFORMATION MINER IN SOCIAL NETWORKS.

https://www.ijrar.org/viewfull.php?&p_id=IJRAR19K4870

3.Shilpi Chawla, Gaurav Dubey, Ajay Rana, Product Opinion Mining Using Sentiment Analysis on Smartphone Reviews.

https://www.researchgate.net/publication/324940438_Product_opinion_mining_using_sentiment_analysis_on_smartphone_reviews

4.Akshay Chavhan , Prof. Sneha A. Khaire , Ankit Kumar , Saurabh Mate ,Vipul Thakare(2020) Sentiment Analysis Using Machine Learning for Twitter, Department of Information Technology, SITRC, Nashik, India.

https://www.academia.edu/43433185/Sentiment_Analysis_Using_Machine_Learning_for_Twitter

5.Shanshan Yi,Machine learning based customer sentiment analysis for recommending shoppers, shops based on customers' review. Chengdu University of technology,China

https://link.springer.com/article/10.1007/s40747-020-00155-2