DE-STRESS – A BATTLE AGAINST DEPRESSION

by

ABHISHEK R 2015013579

PRASANTH P 2015103601

A project report submitted to the

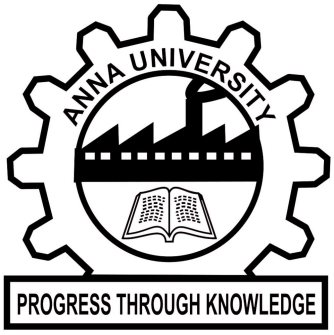
FACULTY OF COMPUTER SCIENCE

AND ENGINEERING

for

CREATIVE AND INNOVATIVE PROJECT

in the academic year of 2018-2019



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, ANNA UNIVERSITY, CHENNAI -25

ABSTRACT

The purpose of this project is to develop an application capable of detecting depression among people. Large amount of data is generated due to increase in social networking sites usage in recent years. This data can be analysed to detect emotions of people, thereby detecting depression, a state of mental illness.

The data generated is pre-processed using natural language processing techniques. The processed data is then fed into a machine learning classifier and the trained classifier is generated. This trained classifier is stored for faster access. A website is developed as the front-end for the user. A module for detection of extreme depression i.e) suicidal thoughts is also added.

Apart from these the application is designed to perform real time evaluation of twitter users by streamming user tweets, performing evaluation and thereby predicting depression.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Depression (major depressive disorder) is a common and serious medical illness that negatively affects how you feel, the way you think and how you act. Fortunately, it is treatable. Depression causes feeling of sadness and/or a loss of interest in activities once enjoyed. It can lead to a variety of emotional and physical problems and can also decrease a person's ability to function at work and at home.

Social Computing is an innovative and growing computing exemplar for the analysis and modelling of social activities taking place on various platforms. It is used to produce intellectual and interactive applications to derive efficient results. The wide availability of social media sites provides individuals to share their sentiments or opinions about a particular event, product or issue. Social media has recently emerged as a premier method to disseminate information online. Through these online networks, tens of millions of individuals communicate their thoughts, personal experiences, and social ideals. Thus we can analyse data from these and thereby predict depression.

1.1 PROBLEM DOMAIN

Natural Language Processing is a field of computer science which deals with interactions between the computer and human languages. It deals with making the computer to understand and interpret human language. Machine learning is a field of computer science that uses statistical techniques to give computer systems the ability to "learn" with data, without being explicitly programmed.

It is quite natural for everyone to be sad once in a while. But if this feeling prolongs over a long period of time it leads to depression. Some symptoms of depression which may go unnoticed are sadness, feeling of loneliness, changes in

sleep cycle, loss of interest, feeling tired all the time etc… We live in a world where people share their thoughts and views over social media constantly.

In this project we have developed a Machine Learning system which can detect depression from users' posts. People use twitter to share their views or feelings more often when compared to other social networking sites. Taking this into consideration we have developed a system which can detect depression from tweets.

1.2 PROBLEM DESCRIPTION

Given a sentence as input the system analyses it for signs of depression and accordingly suggests ideas to the user. The output would be motivational quotes or links to community groups or helplines that offers support.

1.3 SCOPE

Depression is a leading cause of mental ill health, which has been found to increase risk of early death. More than 300 million people suffer from depression worldwide. At its worst, depression can even lead to suicide. Close to 800,000 people die due to suicide every year. In recent years there has been a considerable increase in depression among teenagers. Being able to identify depression in early stages could make a big difference to those who are suffering from it. Hence this system could be of use in the field of medicine/health and in social networks.

1.4 CONTRIBUTION

This system suggests various methods that could help the users to identify and fight depression right from the beginning stages. This system demonstrates the efficiency of machine learning techniques to predict/detect disorders with high accuracy. This could be implemented on by social networking sites so that they can detect and help the individuals who are feeling depressed fight it.

1.5 SWOT ANALYSIS

Strengths:	**Weakness:**
1. Real time information 2. Powerful brand image 3. Helps monitor and follow up campaigns 4. Has 300M + active users 5. Web application access. 6. Easy to collect data and analyse. 7. Highly engaged users. 8. Free to use.	1. Low retention rate. 2. Too much reliability on dataset. 3. Inconsistency in dataset. 4. Limit in maximum allowed characters (280 characters) 5. Unequal distribution of tweets. 6. Data safety issues.
Opportunities:	**Threats:**
1. Easy to access. 2. Integration with other platforms.	1. Data might get lost. 2. Accounts might get hacked. 3. Presence of bots/fake profiles.

1.6 PESTLE ANALYSIS

Political:	**Economical:**
1. Capable of changing the political landscape	1. Increase in e-commerce industry which needs advertising.
Social:	**Technological:**
1. Identify social behaviour. 2. identify trending content. 3. Wide reach.	1. Everything is accessible using mobile phones. 2. Technology keeps improving day to day.
Legal:	**Environmental:**
1. Risk of cyber-stalking and other cyber security issues. 2. Free speech dilemma 3. Defamation and malicious tweets	1. Promotion of environment campaigns. 2. Promoting campaigns for a cause. 3. Comfortable environment.

1.7 ORGANISATION OF THESIS

Chapter 2 discusses the existing approaches to depression detection in greater detail. Chapter 3 gives the requirements analysis of the system. It explains the functional and nonfunctional requirements, constraints and assumptions made in the implementation of the system.

CHAPTER 2

RELATED WORK

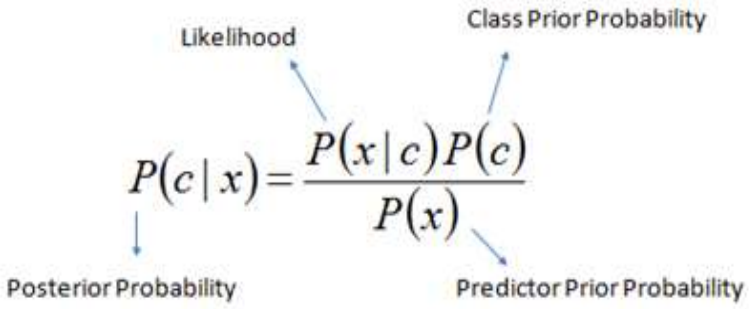
This gives a survey of various machine learning models that can be designed for depression detection. The models include the likes of Multinomial Naïve Bayes Classifier, Support Vector Machine, Decision tree and Random Forest. This helped us to study in detail the various methods and choose the best which suits our needs.

2.1 MODELS

2.1.1 NAIVE BAYES CLASSIFIER

Naïve Bayes classifier is a simple probabilistic classifier that uses the concept of mixture models to perform classification. Naïve Bayes remains a popular method for text categorization, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as the features. With appropriate pre-processing, it is competitive in this domain with more advanced methods including support vector machines. It also finds application in automatic medical diagnosis.

Given a problem instance to be classified, represented by a vector representing some n features (independent variables), it assigns to this instance probabilities for each of K possible outcomes or *classes.*



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

2.1.2 SUPPORT VECTOR MACHINE

Support vector machine (SVM) solves the traditional text categorization problem effectively; generally outperforming Naïve Bayes as it supports the concept of maximum margin. The main principle of SVMs is to determine a linear separator that separates different classes in the search space with maximum distance i.e. with maximum margin The idea of SVM is to determine a boundary or boundaries that separate distinct clusters or groups of data. SVM performs this task constructing a set of points and separating those points using mathematical formulas.

2.1.3 DECISION TREE

Random Forest classifier is a tree-based classifier. It consists of numerous classification trees that can be used to predict the class label for a given data point based on the categorical dependent variable. In the classifier tree, the internal nodes are represented as the features, the edges leaving a node are represented as tests on the feature's weight, and the leaves are represented as class categories. It performs classification preliminary from the root node and moves incrementally downward until a leaf node is detected. The document is then classified in the category that labels the leaf node.

2.1.4 RANDOM FORESTS

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

2.2 DRAWBACKS AND INFERENCES

A linear SVM would be preferred when we wish to emphasize the ability to identifying most depressed individuals at the risk of identifying a few false positives. A Naive Bayes method is preferred when accuracy is given priority.All the above mentioned models are language dependent. A language independent model should be designed.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FUNCTIONAL REQUIREMENTS

- Based on the intensity of depression, the system outputs either a helpline or community group or motivational quotes.
- The system must be optimised for space complexities.
- The system must be able to make prediction based on the entered text input.
- Server response time must be optimal.
- Content of website must be motivating and helpful.

3.2 NON FUNCTIONAL REQUIREMENTS

3.2.1 USER INTERFACE

There must be a simple to use and attractive UI where the user can enter the input (tweets).

3.2.2 HARDWARE REQUIREMENTS

The client side requirement is only a browser with good internet connection The server side must be able to handle multiple requests & tools to run machine learning python scripts. Server must be able to run javascript runtime environment (NodeJs)

3.2.3 SOFTWARE REQUIREMENTS

OS - Windows or Linux

Programming Language - Javascript, Python.

Frameworks - Jquery, Bootstrap.

Libraries - Pandas, Scikit-learn, Numpy, NLTK, Tweepy.

Tools - Visual Studio Code

3.2.4 PERFORMANCE

The system should provide consistent, reliable with low false positives.

3.3 CONSTRAINTS AND ASSUMPTIONS

3.3.1 CONSTRAINTS

- The system can’t differentiate between sarcasm and seriousness.
- System can’t identify internet lingos like rofl, lol, bff.
- There could be false positives.

3.3.2 ASSUMPTIONS

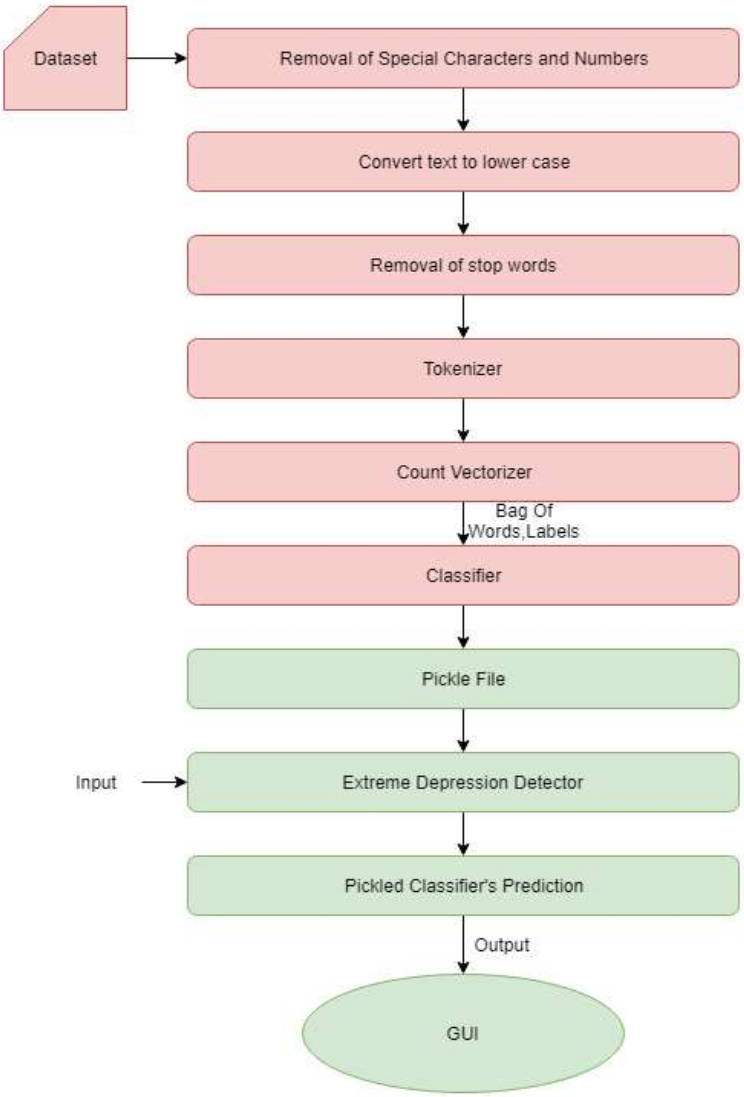
- The text is in English and is void of foreign letters/characters.
- The user input is genuine.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

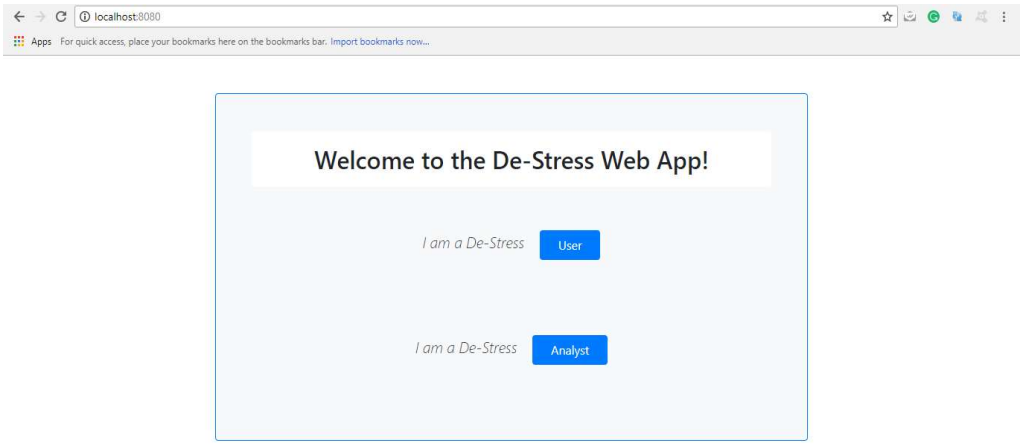
The block diagram of the entire system is shown in figure 4.1. A web application has been developed.

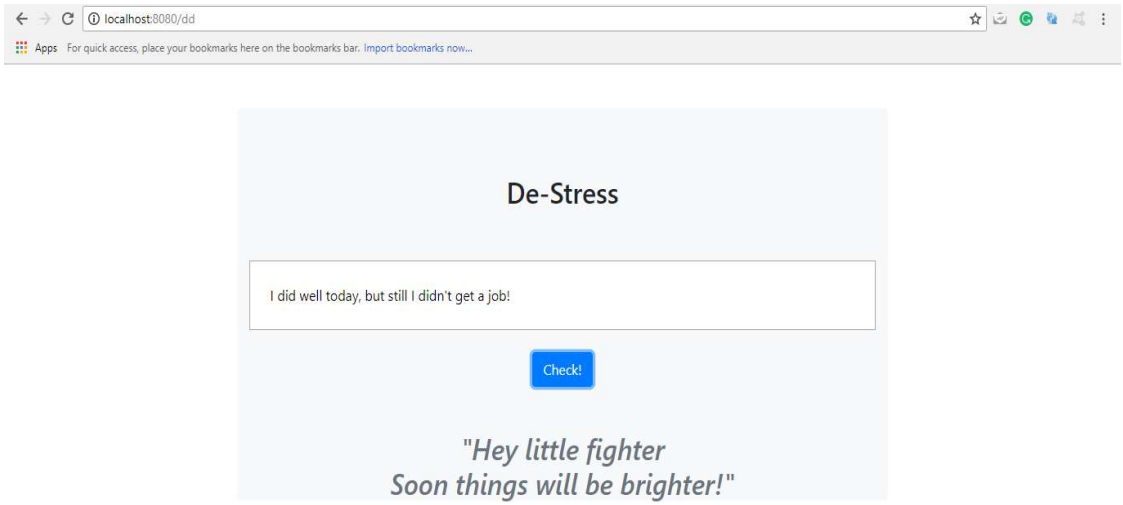


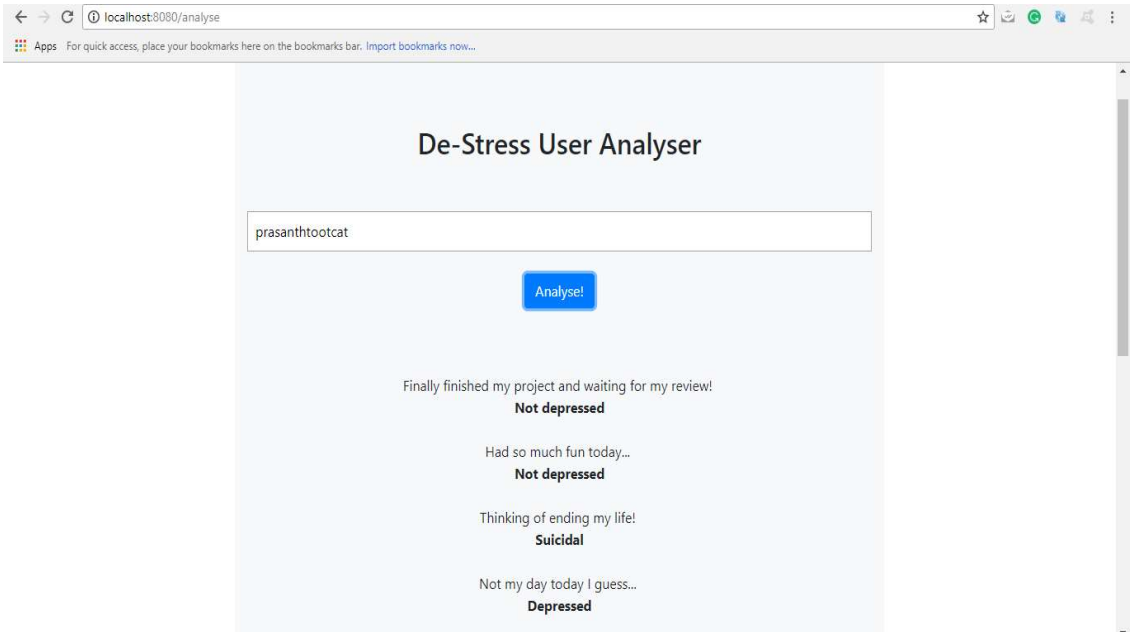
The system aims at validating the user input for possibility of depression. The depression dataset and user's input are initially subjected to preprocessing. The preprocessor takes the English text/tweet, removes the special characters and converts the text to lowercase. This is followed by tokenizing the text and removal of stop words. This preprocessed dataset is then sent to the count vectoriser which in turn produces a bag of words along. The bag of words and labels is sent to the Multinomial Naive Bayes classifier and a trained classifier is generated which is in turn stored in a pickle file. The preprocessed input is fed to the extreme depression detector and pickled classifier where the intensity of depression is determined. The output is displayed in the UI.

4.2 UI DESIGN

A simple and easy to use User Interface (UI) has been designed for the system using Visual Studio IDE. The home screen of the UI contains two buttons which enables the user to switch between user mode and analyst mode. Each button takes you to the respective screen which has a text box and a button. The input sentences/paragraph is typed in the textbox . On clicking the Check/Analyze button, the appropriate output will be displayed.




localhost:8080/dd
Apps For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
De-Stress
I did well today, but still I didn't get a job!
Check!
"Hey little fighter
Soon things will be brighter!"


localhost:8080/analyse
Apps For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
De-Stress User Analyser
prasanthtootcat
Analyse!
Finally finished my project and waiting for my review!
Not depressed
Had so much fun today...
Not depressed
Thinking of ending my life!
Suicidal
Not my day today I guess...
Depressed

4.3 CLASS DIAGRAM

The class diagram of the entire system is shown in figure 4.3. This diagram depicts the functions of various modules in the system clearly. It also shows the interaction between the modules of the system thereby providing a clear idea for implementation.

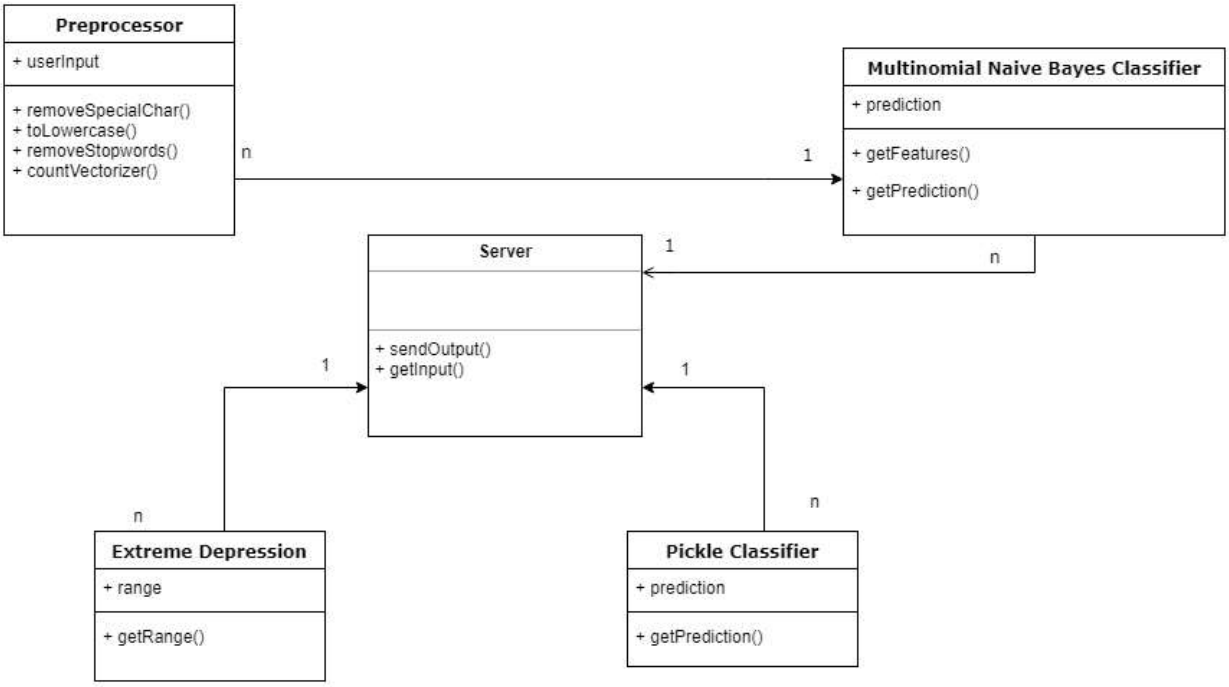


Figure 4.3 Class Diagram

4.4 MODULE DESIGN

4.4.1 PRE-PROCESSING

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues.

- Data cleaning: fill in missing values, smooth noisy data and resolve inconsistencies.
- Data transformation: normalization and aggregation.
- Data reduction: reducing the volume but producing the same or similar analytical results.
- Data discretization: part of data reduction, removing numerical attributes and special characters.

The text must be parsed to remove words, called tokenization. Then the words need to be encoded as integers or floating point values for use as input to a machine learning algorithm, called feature extraction (or vectorization). CountVectorizer builds a count matrix where rows are occurrences counts of different words taking into account the high-dimensional sparsity.The scikit-learn library is used to perform both tokenization and feature extraction of the input text data.

BAG OF WORDS

We cannot work with text directly when using machine learning algorithms. Each tweet is an "*input*" and a class label is the "*output*" for our predictive algorithm. Algorithms take vectors of numbers as input, therefore we need to convert documents to fixed-length vectors of numbers.

A simple and effective model for thinking about text documents in machine learning is called the Bag-of-Words Model, or BoW.

The bag-of-words and labels is fed to the Multinomial Naive Bayes Classifier.

4.4.2 MULTINOMIAL NAIVE BAYES CLASSIFIER

Naive Bayes is a family of algorithms based on applying Bayes theorem with a strong(naive) assumption, that every feature is independent of the others, in order to predict the category of a given sample. They are probabilistic classifiers, therefore will calculate the probability of each category using Bayes theorem, and the category with the highest probability will be output. Naive Bayes classifier is a general term which refers to conditional independence of each of the features in the model, while Multinomial Naive Bayes classifier is a specific instance of a Naive Bayes classifier which uses a multinomial distribution for each of the features. Multinomial Naive Bayes simply assumes multinomial distribution for all the pairs, which seem to be a reasonable assumption in some cases, i.e. for word counts in documents.

4.4.3 PICKLE FILE

Pickle is used for serializing and de-serializing Python object structures, also called marshalling or flattening. Serialization refers to the process of converting an object in memory to a byte stream that can be stored on disk or sent over a network. Later on, this character stream can then be retrieved and de-serialized back to a Python object. Pickling is the conversion of an object from one representation (data in Random Access Memory (RAM)) to another (text on disk). Pickle is very useful for when you're working with machine learning algorithms, where you want to save them to be able to make new predictions at a later time,

without having to rewrite everything or train the model all over again.It can also be used to send data over a Transmission Control Protocol (TCP) or socket connection, or to store python objects in a database.

4.4.4 SERVER

The server side scripting language is NodeJs. It uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. NodeJs represents a "JavaScript everywhere" paradigm, unifying web application development around a single programming language, rather than different languages for server side and client side scripts. The user is redirected to the page according to the url and the input is transferred as a query through REST calls. The query is processed and the output is displayed in the GUI.

4.4.5 GUI

HTML and Bootstrap are used to develop the front end. The graphical user interface, is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation. Nearly all digital interfaces are GUIs.

4.5 COMPLEXITY ANALYSIS

4.5.1 TIME COMPLEXITY

The time complexity of all preprocessing steps is O(N).

In case of Multinomial naive bayes classifier the

Training Time: O(|D|Ld + |C||V|)) where Ld is the average length of a document in D. Assumes V and all Di , ni , and nij pre-computed in O(|D|Ld) time during one pass through all of the data.

Test Time: O(|C| Lt) where Lt is the average length of a test document. Very efficient overall, linearly proportional to the time needed to just read in all the data.

4.5.2 COMPLEXITY OF THE PROJECT

The complexity of the project depends to a great extent upon the time taken to train the classifier and size of the data set. The efficiency depends upon the accuracy of the data set used. To improve access speed, the trained classifier is dumped into a pickle file. Response time depends on the latency of the server.

CHAPTER 5

SYSTEM DEVELOPMENT

The system described consists of various packages like pandas, numpy, regex, sklearn etc… The overall code overview showing the organisation of these various packages of the Machine Translation system can be seen in figure 5.1.

```
import pandas as pd
import numpy as np
import re,sys
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from sklearn.externals import joblib
from sklearn.feature_extraction import text
```

Figure5.1 Code Overview

5.1 PROTOTYPE ACROSS THE MODULES

The input and output to each module of the system is described in this section.

PREPROCESSOR

This module takes the English text/tweet as input and produces as intermediary output the text without special characters and numbers.This is followed by tokenizing the text and removing stop words (am,is,are,..). A bag of words approach is used.The countvectorizer produces the bag of words. The bag of words and the labels are fed to the Multinomial Naive Bayes Classifier.

MULTINOMIAL NAIVE BAYES CLASSIFIER

This module take the bag of words and the labels as its input. The classifier classifies the tweets as depressed or not depressed.The model is trained with 80% of dataset and tested with 20%.

PICKLE FILE

The trained classifier is pickled so that access time is reduced. This ensures that output is determined faster that under normal circumstances.

SERVER

The text entered by the user/fetched from the user's account is subjected to the extreme depression detection sub-module which checks for high intensity of depression and the text is validated for depression against the pickle file.

5.2 ALGORITHM

The Multinomial Naive Bayes algorithm used for prediction of depression is as show below

TRAINMULTINOMIALNB($\mathbb{C}, \mathbb{D}$)
1 $V \leftarrow$ EXTRACTVOCABULARY($\mathbb{D}$)
2 $N \leftarrow$ COUNTDOCS($\mathbb{D}$)
3 **for each** $c \in \mathbb{C}$
4 **do** $N_c \leftarrow$ COUNTDOCSINCLASS($\mathbb{D}, c$)
5 $prior[c] \leftarrow N_c/N$
6 $text_c \leftarrow$ CONCATENATETEXTOFALLDOCSINCLASS($\mathbb{D}, c$)
7 **for each** $t \in V$
8 **do** $T_{ct} \leftarrow$ COUNTTOKENSOFTERM($text_c, t$)
9 **for each** $t \in V$
10 **do** $condprob[t][c] \leftarrow \frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$
11 **return** $V, prior, condprob$

APPLYMULTINOMIALNB($\mathbb{C}, V, prior, condprob, d$)
1 $W \leftarrow$ EXTRACTTOKENSFROMDOC(V, d)
2 **for each** $c \in \mathbb{C}$
3 **do** $score[c] \leftarrow \log prior[c]$
4 **for each** $t \in W$
5 **do** $score[c] += \log condprob[t][c]$
6 **return** $\arg\max_{c \in \mathbb{C}} score[c]$

5.3 DEPLOYMENT DETAILS

Python libraries such as pandas, numpy, sklearn, tweepy etc… must be available in the deployment environment with support to deploy NodeJs application. Apart from these front end libraries such as bootstrap and jquery must be made available either locally in server or through CDN.

CHAPTER 6

RESULTS AND DISCUSSION

6.1 DATASET FOR TESTING

The dataset has around 35,000 tweets. It training dataset consists of tweets with depressed/not depressed labels. Each module of the system was also tested separately. The results of this module testing as well as the testing of the entire system are summarized below.

6.2 OUTPUT OBTAINED IN VARIOUS STAGES

This section shows the results obtained during module testing

6.2.1 INPUT

The input is a tweet in case of a user or twitter username incase of an analyst.

6.2.2 PRE-PROCESSING

The output obtained from the pre-processing module is as show in the figure 6.1 and figure 6.2

Figure 6.1

BAG OF WORDS

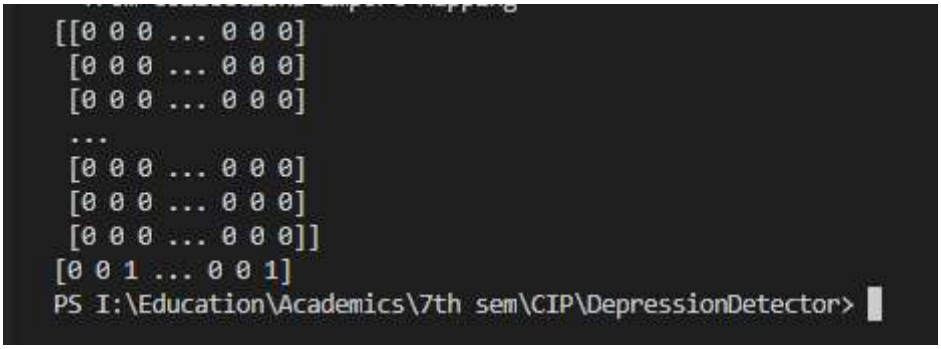


Figure 6.2

6.2.3 CLASSIFIER

The classifier is trained and is dumped into a pickle file as show in the figure 6.3

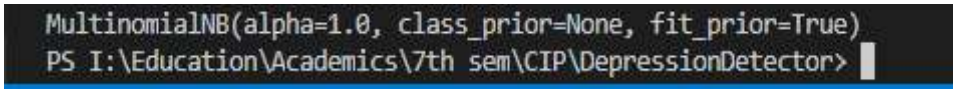


Figure 6.3

6.2.4 SERVER

The server runs the algorithm and sends the prediction to the GUI as output. The output is rendered as motivational quote, helpline or as an analysis depending on the user profile.

6.2.5 OUTPUT

The sample output obtained is of various forms depending on the user profile as shown in figure 6.4

Figure 6.4

6.3 SAMPLE SCREENSHOTS OF INPUT AND OUTPUT WHILE TESTING

Figure 6.5 and 6.6 shows the different outputs obtained during testing.

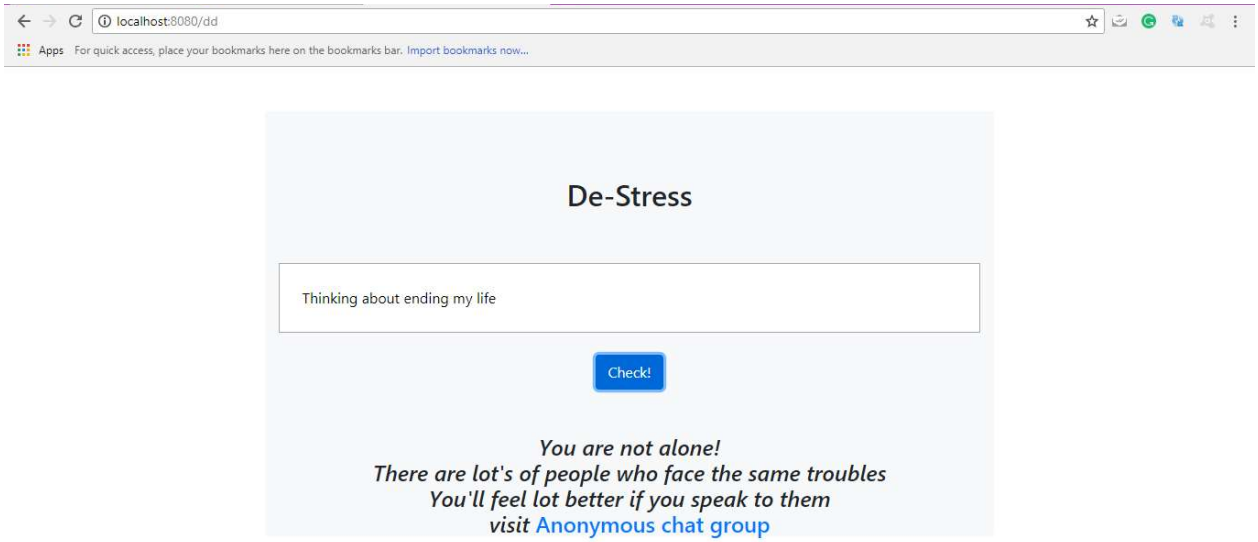


Figure 6.5

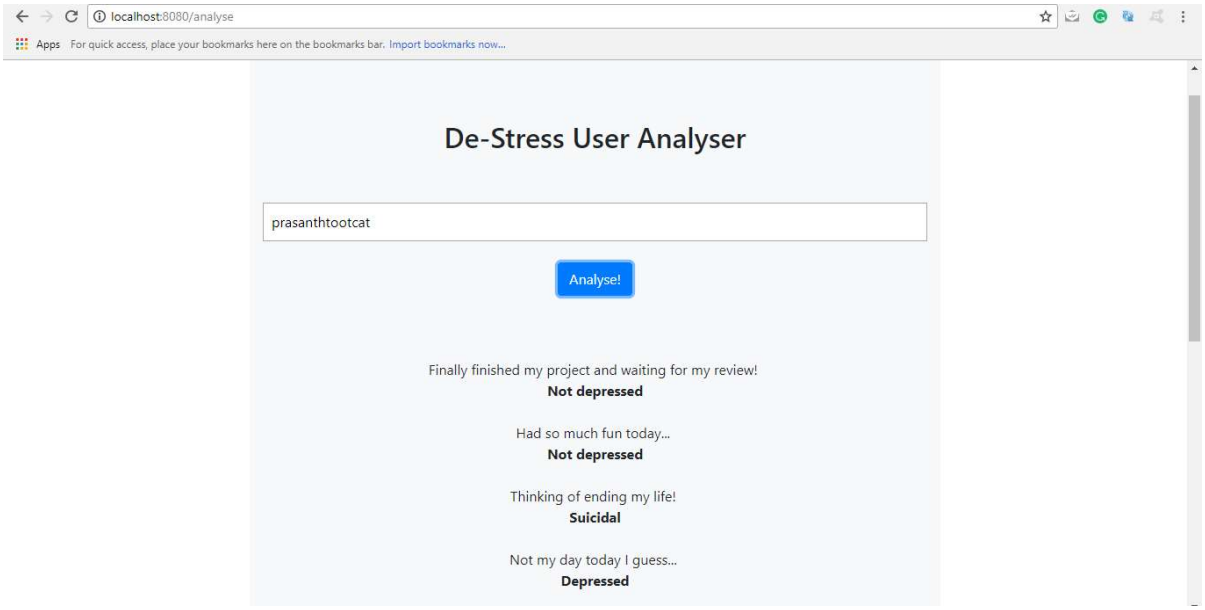


Figure 6.6