VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", BELAGAVI - 590 018

A PROJECT REPORT

on

"DEPRESSION DETECTION USING NLP "

Submitted by

Rathan H V	4SF16CS124
Raviganesh M	4SF16CS125
Shyam Kishore	4SF16CS150

In partial fulfillment of the requirements for the award of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE & ENGINEERING

Under the Guidance of

Mrs. Soumya Patil

Assistant Professor, Department of CSE

at

SAHYADRI

College of Engineering & Management

Adyar, Mangaluru - 575 007

2019 - 2020

SAHYADRI

College of Engineering & Management

Adyar, Mangaluru - 575 007

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project entitled **"Depression Detection using NLP"** has been carried out by **Rathan H V (4SF16CS124), Raviganesh M (4SF16CS125) and Shyam Kishore (4SF16CS150)**, the bonafide students of Sahyadri College of Engineering & Management in partial fulfillment for the award of Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belagavi during the year 2018 - 19. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Signature of the Guide	**Signature of the HOD**	**Signature of the Principal**
Mrs. Soumya Patil	Dr. J V Gorabal	Dr. Rajesha S

External Viva:

Examiner's Name	Signature with Date
1.	.....
2.	.....

SAHYADRI

College of Engineering & Management

Adyar, Mangaluru - 575 007

Department of Computer Science & Engineering

DECLARATION

We hereby declare that the entire work embodied in this Project Report titled **"Depression Detection using NLP"** has been carried out by us at Sahyadri College of Engineering and Management, Mangaluru under the supervision of **Mrs. Soumya Patil,** for the award of **Bachelor of Engineering** in **Computer Science & Engineering**. This report has not been submitted to this or any other University for the award of any other degree.

Rathan H V (4SF16CS124)

Raviganesh M (4SF16CS125)

Shyam Kishore (4SF16CS150)

Dept. of CSE, SCEM, Mangaluru

Abstract

Youngsters often turn to social media platforms for mental health support. Referring to the comments and posts on such platforms can give us a brief idea of how people self disclose and discuss mental health issues such as depression. Depression can be considered as one of the major contributor to mental disability and a major reason for suicide. Usually, depressed person expresses his feelings in the form of text. The key objective of our study is to examine users' posts to detect any factors that may reveal the depression attitudes of various kinds of users.

Using a data set of scraped Twitter comments, which is obtained from Kaggle, this project aims to classify depression in comments. This project implements methods of machine learning and neural network architectures for identifying depression in digitally shared users comments. This project is implemented using machine learning algorithms such as logistic regression, support vector machines, a BERT-based model, and neural networks with Word2Vector for this classification task. An accessible website is often one of the easiest ways to interact and to find solution to the depression. The project consists of an user friendly website which is compatible with the inputs such as text, voice and the image.

Acknowledgement

It is with great satisfaction and euphoria that we are submitting the Project Report on **"Depression Detection using NLP"**. We have completed it as a part of the curriculum of Visvesvaraya Technological University, Belagavi for the award of Bachelor of Engineering in Computer Science & Engineering.

We are profoundly indebted to our guide, **Mrs. Soumya Patil**, Designation, Department of Computer Science & Engineering for innumerable acts of timely advice, encouragement and We sincerely express our gratitude.

We also thank **Mr. Shailesh S Shetty**, **Mr. Girisha S** and **Mrs. Shwetha R J**, Project Coordinators, Department of Computer Science & Engineering for their constant encouragement and support extended throughout.

We express our sincere gratitude to **Dr. J V Gorabal**, Head & Professor, Department of Computer Science & Engineering for his invaluable support and guidance.

We sincerely thank **Dr. Rajesha S**, Principal, Sahyadri College of Engineering & Management, **Dr. Umesh M. Bhushi**, Director Strategic and Planning, Sahyadri College of Engineering & Management, and **Dr. D. L. Prabhakara**, Director, Sahyadri Educational Institutions,who have always been a great source of inspiration.

Finally, yet importantly, We express our heartfelt thanks to our family & friends for their wishes and encouragement throughout the work.

Rathan H V (4SF16CS124)

Raviganesh M (4SF16CS125)

Shyam Kishore (4SF16CS150)

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

People nowadays share more posts on social media platforms like Facebook, Instagram, Twitter, Reddit etc about their moods pertaining to their daily lives. Due to the busy schedule, people are not interested to listen to the feelings of others. So, the depressed people uses the online platform to express their feelings. According to the study conducted by World Health Organization, there are approximately 322 million people estimated to be suffered from depression, which is equivalent to 4.4% of the total population. Nearly half of the in-risk individuals live in the south-east Asia (27%) and western pacific region (27%) including china and India. World Health Organization has predicted that the prevalence of depression will increase over the next 20 years.

Depressive disorders can affect one's general health and habits, including sleep patterns and eating behaviors. They can also affect one's interpersonal relationships. Many techniques have been implemented to determine the depression status of a person by observing and extracting the emotions from the text, speech and image, using emotion theories, machine learning approaches, and NLP techniques on different social media platforms. Depression related data can be collected from the sources such as Tweets, Depression blogs, Reddit. The nature of the post is decided by comparing it with dataset of scraped social media comments.

This project aims to classify depression level through their comments. Since the detection of depression in the clinical level is difficult, it can be made easy by extracting the symptoms related to depression in the form of text from the social media posts. The depression posts are collected only from Twitter is the drawback of this model. There are different methodologies to detect depression from the posts. This method uses the NLP and Text classification techniques. The framework consists of data-pre-processing, feature traction followed by the machine learning classifiers, features analysis and experimental results.

1.1 Purpose

The main purpose of this project is to presents a new model which determines the status of depression of the user. The existing model does not provide the solution to the user. But, our model suggests the solution to the depressed user by finding the cause of the depression. The project is compatible not only to the texts but also it takes the voice and image as the input. If the suggested solution is not satisfied, then it redirects to expert solutions.

1.2 Scope

This project contains a website which has solutions for different kinds of depression. This model selects the solution based on keyword mapping from the input given by the user. The model will select the best possible solution from the given solution set.

1.3 Overview

Since the number of depressive people are growing at higher rate day by day, it is very essential to come up with a new model which can handle large amount of depression data without reducing the performance of the model. The model implements advanced technologies of machine learning and NLP. This model will allow the user to input the depressive content through various forms such as text, voice and image.

Chapter 2

Literature Survey

In [1], Anees Ul Hassan, Jamil Hussain, Musarrat Hussain, Muhammad Sadiq, Sungyoung Lee *et al.* it proposes a method where there are two types of sentiment classification technique, binary classification and multi-class classification. In binary classification, documents are categorized into Positive and negative. There are two types of sentiment classification technique, binary classification and multi-class classification. In binary classification, documents are categorized into Positive and negative. In multi-class sentiment classification, the given document is categorized as Strong Positive, Positive, Neutral, Negative, and Strong Negative. In the proposed method there are four components they are pre-processing, feature extraction, Meta learning and training data. In data pre-processing, author first split the paragraph into sentences and then tokenize the sentences into words. From words vector then author remove the stop words. On the remaining words find the root words.

In [2], Tao, Xiaohui and Dharmalingam, Ravi and Zhang, Ji and Zhou, Xujuan and Li, Lin and Gururajan, Raj *et al.* the dataset for training the model is collected by scraping two subreddits: /r/depression and /r/AskReddit. Depending on the sub-reddit from which the comment is extracted, it is labeled with 1 or 0 for "depressed" or "non-depressed". The data were concatenated, randomly shuffled, and split in a 80%-20%-20% ratio for training, development, and testing sets respectively. From the sub-reddits /r/depression and /r/AskReddit, the "hot" and "top" posts were scrapped to create the custom dataset. The available Python Reddit API Wrapper (PRAW) is used to scrap the posts. All comments from /r/depression were considered "depressed" and those /r/AskReddit were "non-depressed." Data pre-processing is done to remove the unwanted content from the actual post. But one of the disadvantages of this model is, the sub-reddit /r/depression may contain the posts which are not depressive. It will be difficult for the model to predict the

proper output in these situations.

In [3],Long Ma, Zhibo Wang, and Yanqing Zhang *et al.* they proposed and implemented the techniques of data mining and NLP to collect symptoms by their posts from the social medias. The collected data is then preprocessed. During data analysis, frequent words are collected and that words are compared with depression symptoms. The word clustering is used to minimize the complexity of dataset. The data collections can be done from the sources such as Tweets(TW), Professional Twitter Accounts (PTA), Depression Blogs (DB). The collected data is then pre processed where the punctuations, special characters such as retweet tags "@RT: xxx" and link address "http://www.", contain less information are removed. Nonwords such as "hrt", "lmao" are filtered by the NLTK toolkit. During data analysis, frequency of each word is collected and the frequent words are considered as important and then that word is checked with the depression symptoms such as eg. words 'anxiety' and 'disorder'.

In [4], Michael M. Ta desse, Hongfei Lin, Bo Xu, and Liang Yang *et al.* introduced Attribute-Based Encryption (ABE) which allows each cipher text to be associated with an attribute, and the master-secret key holder can extract a secret key for a policy of these attributes in, so that a cipher text can be decrypted by this key if its associated attribute conforms to the policy. For example, with the secret key for the policy, one can decrypt cipher text tagged with class 2, 3, 6, or 8. However, the major concern in ABE is collusion resistance but not the compactness of secret keys.Indeed, the size of the key often increases linearly with the number of attributes it encompasses, or the cipher text-size is not constant. To delegate the decryption power of some cipher texts without sending the secret key to the delegatee, a useful primitive is proxy re-encryption (PRE).

In [5], Hu, Hsiao-Wei and Hsu, Kai-Shyang and Lee, Connie and Hu, Hung-Lin and Hsu, Cheng-Yen and Yang, Wen-Han and Wang, Ling-yun and Chen, Ting-An *et al.* it proposes a method to detect the major depressive disorders using the DSM-IV-TR. The expression is extracted by analyzing the text which is represented by the user. The process is also called as the opinion mining which uses the technique of Natural Language Processing(NLP) and the techniques of Machine Learning to determine the mental health of the user. This tool mainly makes use of the two main model architectures which are continuous bag-of-words(CBOW) model and the continuous skip-gram. Both of these architectures are used to learn the vector representation of the words which are included in the post which the user writes. In continuous bag-of-word, the different words are combined to predict the word in the middle. The predicted word will be having the similar context as the group

of words from which the new word is derived. Here, the order of word in the past will not influence the projection. The prediction of word is done based on the context. The skip-gram predicts the surrounding words by considering the single base word. The neighboring words are defined by the window size.

In [6], Wenwen Li, Michael Chau *et al.* here, author use the design science approach and propose DK-LSTM, a novel design based on deep learning to identify people with depression and emotional distress. Based on LSTM networks are a variation of RNNs with Long ShortTerm Memory units, a type of deep learning networks. The LSTM detects domain-based information. DK-LSTM aims to capture both general information and domain-specific information at the same time. The LSTM detects domain-based information. DK-LSTM aims to capture both general information and domain specific information at the same time. There are two LSTM units in the LSTM Layer to process general representation and domain representation separately. Merge layer combines word embedding layer and LSTM layer. The Merge Layer combines the feature vectors produced by the two LSTM units and generate a mixed feature vector.

In [7], Irwan Oyong, Ema Utami, Emha Taufiq Luthfi *et al.* in this approach, Predictions were performed on a set of tweets with a time span of 2 weeks to 2 months according to the minimum limit of symptoms of Major Depressive Disorder and limit of manual review capability by human experts. [7] Data that has been collected through scraping and searching through the Twitter Search API undergoes standard text processing and text-specific processing for tweets. Data for building symptoms lexicon are obtained from diagnostic clinical manuals and statistics of the world standard mental disorders DSM-IV and CES-D. The data to construct the lexicon of frequency is derived from the general frequency dictionary and the frequency gradient adopted from CES-D and produces words such as never, sometimes, often, and always. Tweets data that has been collected undergo text pre-processing before then will be analysed and calculate the score of the tendency of depressive symptoms. The two text processing techniques used are standard and twitter text processing.

In [8], MarioEzraAragon, A.PastorLopez-Monroy, LuisC.Gonz´alez-Gurrola and Manuel Montes-y-G´omez *et al.* here author proposes a new representation called Bag of Sub-Emotions (BoSE). The social media documents are represented by a set of fine-grained emotions which are generated automatically using a lexical resource of emotions and sub-word embedding. The lexical resource based on eight recognized emotions(Anticipation, joy, fear, disgust, sadness,surprise, trust, anger) and the two main sentiments negative and

positive are used to generate the fine-grained emotions. There are two steps in converting text to fine-grained emotion and they are Text masking and Text representation. In text masking, documents are masked by replacing each word with the label of its closest fine-grained emotion. In text representation, based on the masked documents, author builds the BoSE representations computing a frequency histogram of their fine grained emotions.

Chapter 3

Problem Definition

Mental illness is the mostly faced problems in the world, the depression is most common psychological problem. The improper diagnosis of depression may lead to dangerous behavior of the patient. The main problem of detecting the depression is to recognize the depressive symptoms in the patient since the symptoms may differ from person to person based on his behavior and personality.

Clinical detection of depression is very difficult because of some restricted factors. Nowadays most of the people share their feelings through online posts on social media.

To overcome this problem, there is a need of a suitable model that predicts the presence of depression of a person by using the posts through which he has expressed his feelings.

Chapter 4

Software Requirements Specification

4.1 Introduction

Software Requirement Specification totally defines how the projected software behaves without unfolding how the software will perform it. The elementary objective of the requirement stage is to yield the software requirement specification that designates the peripheral performance of the projected software. Software requirement can be well- defined as a condition of a capability required by a user to solve a problem or attain an objective.

4.2 Purpose

The purpose of this project is to predict the status of depression of a person by analyzing his post in the form of sentence. The technique of NLP is used to predict the depression in the sentence. The input to the model is fed from the web page where user enters his feeling as the post. If depression is detected in the user entered text, the web page provides the related solutions to overcome the depression problems.

4.3 User Characteristics

A web page is provided to the user with an input field where he has to type his feeling.It also allows the user to input the text through voice. The web page provides the facility to identify the depression by inputting the images which are embedded with text contents. If the depression is detected in the input sentence, then it will redirect to the solution page.

4.4 Interfaces

4.4.1 Hardware Interfaces

- Processor : Any processor above 500 MHz
- RAM : 8GB
- Hard Disk : 1TB
- Processor : i5
- Output Device : Monitor

4.4.2 Software Interfaces

- Operating system : Windows or Linux
- Browser: Google Chrome or Firefox
- Text Editor : Gedit, Notepad
- Programming Language : Python

4.5 Functional Requirements

- The webpage will allow the user to give input as text posts.
- Webpage will be integrated with Machine Learning models that returns the depression level.
- The Webpage will provide an interface to the users to express their feelings through their text posts.
- Users will be able given necessary solution based upon the depression level that is found out by using our model.

4.6 Non-Functional Requirements

- The Webpage must be efficient with very little lag in response time to users reply.
- The model must be reliable to find depression level with no fault.

- The use of natural language used to find the depression level.
- Appropriate handling of unexpected input and correctly inform the user if it cannot provides solution.

Chapter 5

System Design

5.1 Architecture Design

The training data to the model is collected from Kaggle. The dataset has three attributes that are depression status, source of data and the text. Only the depression status and text are fed to the machine learning model. Before feeding the text to the model, all the sentences are pre-processed. All the special characters are removed from the text. Tokenization of the sentence is done.

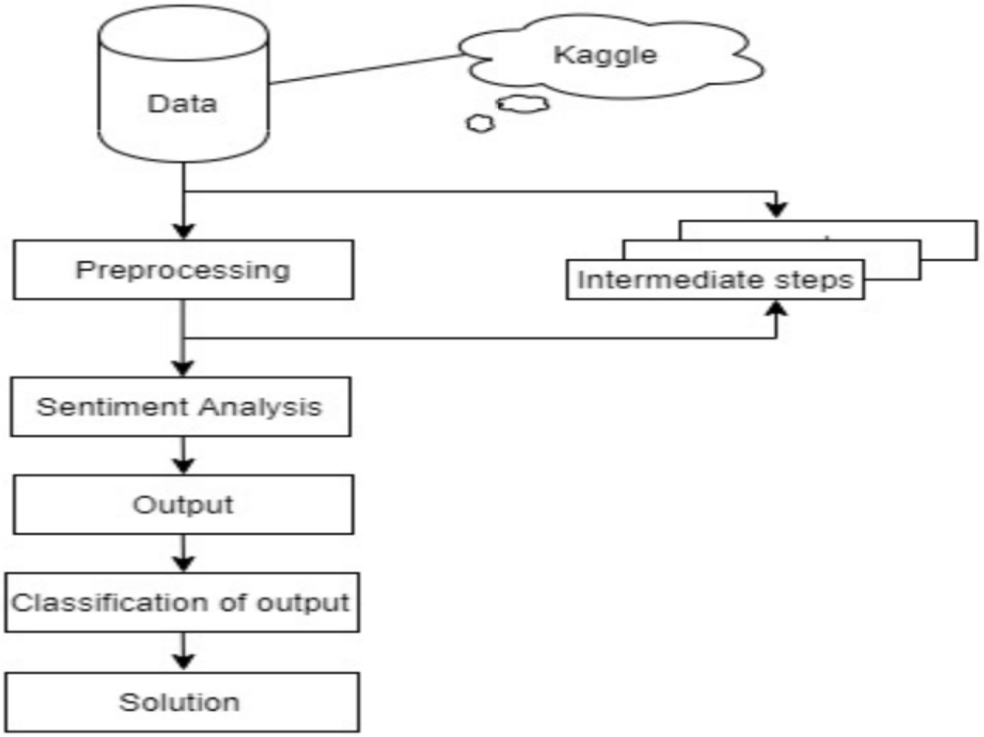


Figure 5.1: System Architecture Diagram

All the words are converted into their root words. The cleaned data is split into training and testing data. A word2vec model is created using the gensim library. An embedding matrix is created and added with the embedding layer. A sequential model is created to predict the depression. The convolution layer and LSTM layers were added to the model to get higher accuracy. A user friendly web page is made to get the status from the user. When the user types his feelings, the data is fed to the machine learning model and it will predict whether the sentence has depression or not. The result will be shown in a new web page. If the user has depression, it will be redirected to the page where he can find the appropriate solutions to overcome the depression problems.

5.2 Data Flow Design

The data for training the model is collected from Kaggle. The attribute columns in the dataset are the depression text and the status of depression. If the sentence has depression, it's status will be 1 or else the status will be 0. The data pre-processing is done to increase the efficiency of the model. All the unwanted symbols and other contents are removed from each sentence. Each word in a sentence is converted into it's root word. Only the decision making words are kept and other words are eliminated from the sentence. The data is splitted into training and testing data. The training data is fed into the machine learning model.

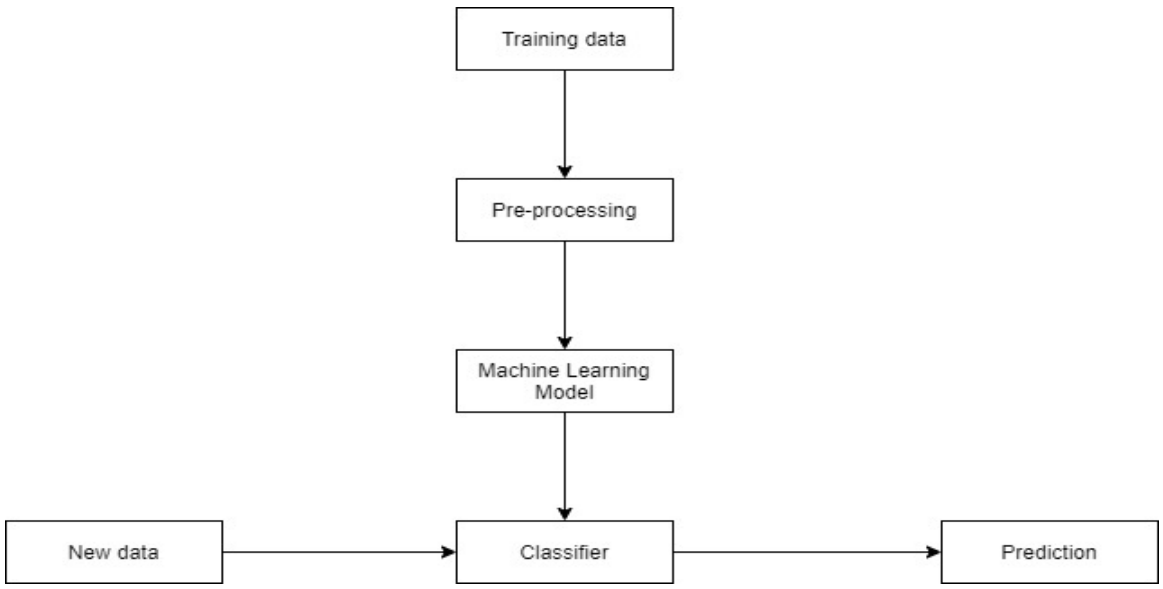


Figure 5.2: Data Flow Design

The new data from the user interface is then inputted into the classifier and the prediction will be made. The result will be again sent back to the user interface. The user will be able to see the depression status along with the associative result to the depression problem.

5.3 Sequence Diagram

This model contains two phases, training phase and testing phase. In the training phase, the data is fed into the pre-processor. During pre-processing, the unwanted content is removed from the data and the special characters are also removed. All the words are converted into their root form. A word2vec model is built and the pre-processed data is inputted into the word to vector model. Then the tokenization of the data is done. After the completion of tokenization, the tokenized data is fed into the depression detection model for training. The user can input the statements in the forms of text, voice and image. The extracted text will be given to the model. The depression status will be predicted by the model and returned back. The result will be displayed on the web page of the user. If depression is detected in the user inputted sentence, then the associative solution will also be displayed.

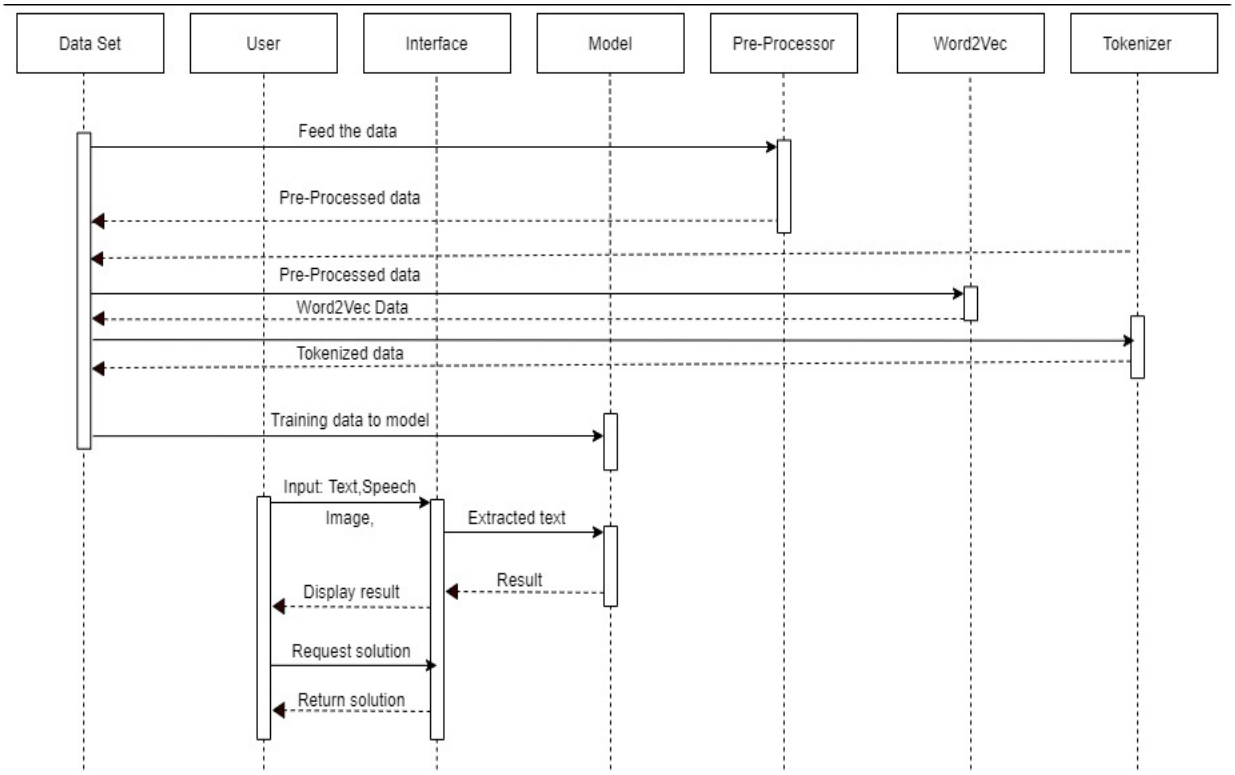


Figure 5.3: Sequence diagram

5.4 Use case Diagram

This model helps in predicting the presence of depression in a sentence. During the training stage, the collected dataset will be pre-processed. The special characters and unwanted contents are removed from the dataset. All the words are converted into their root words. A word to vector model is built and the document is fed into that model. The data is undergone the process of tokenization and all sentences are divided into a group of words. An embedding matrix is created by feeding the tokenized data and word2vec model. An embedding layer is created by using the embedding matrix. A sequential model is built to detect the depression. The embedding layer is added to the sequential model along with the convolution layer and LSTM layer.

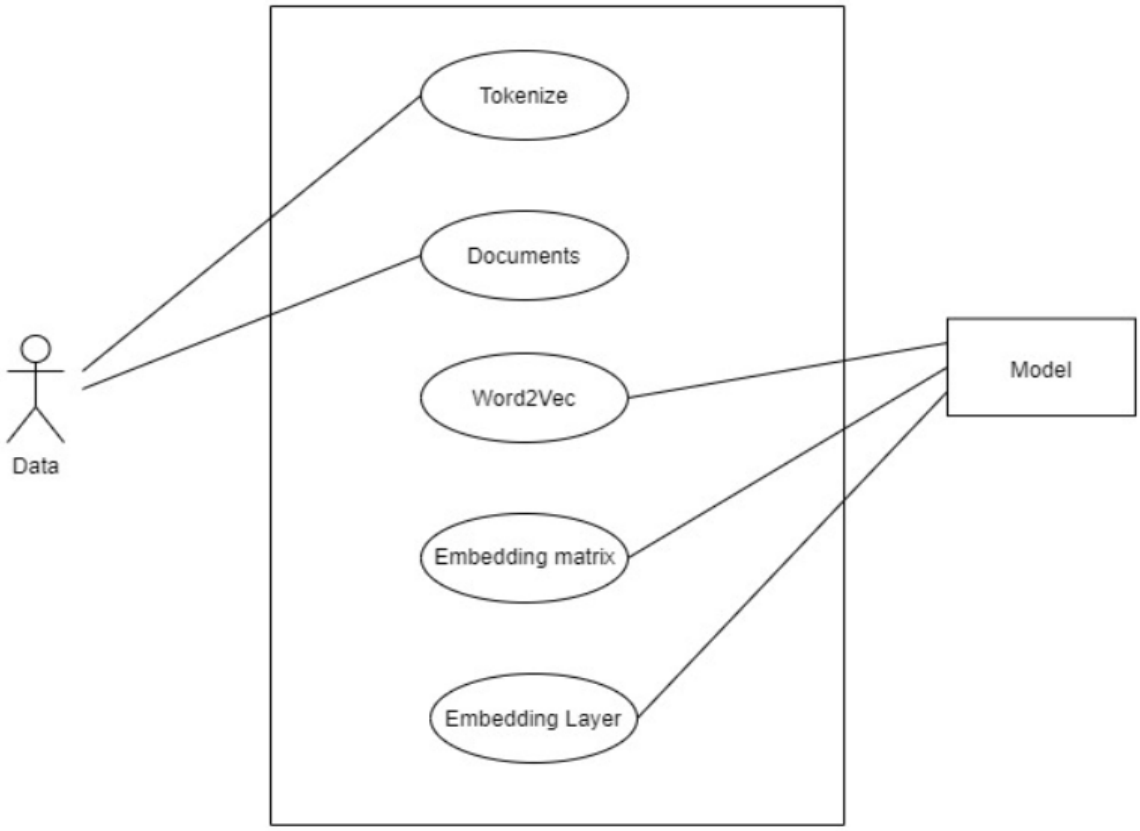


Figure 5.4: Use case Diagram for training the model

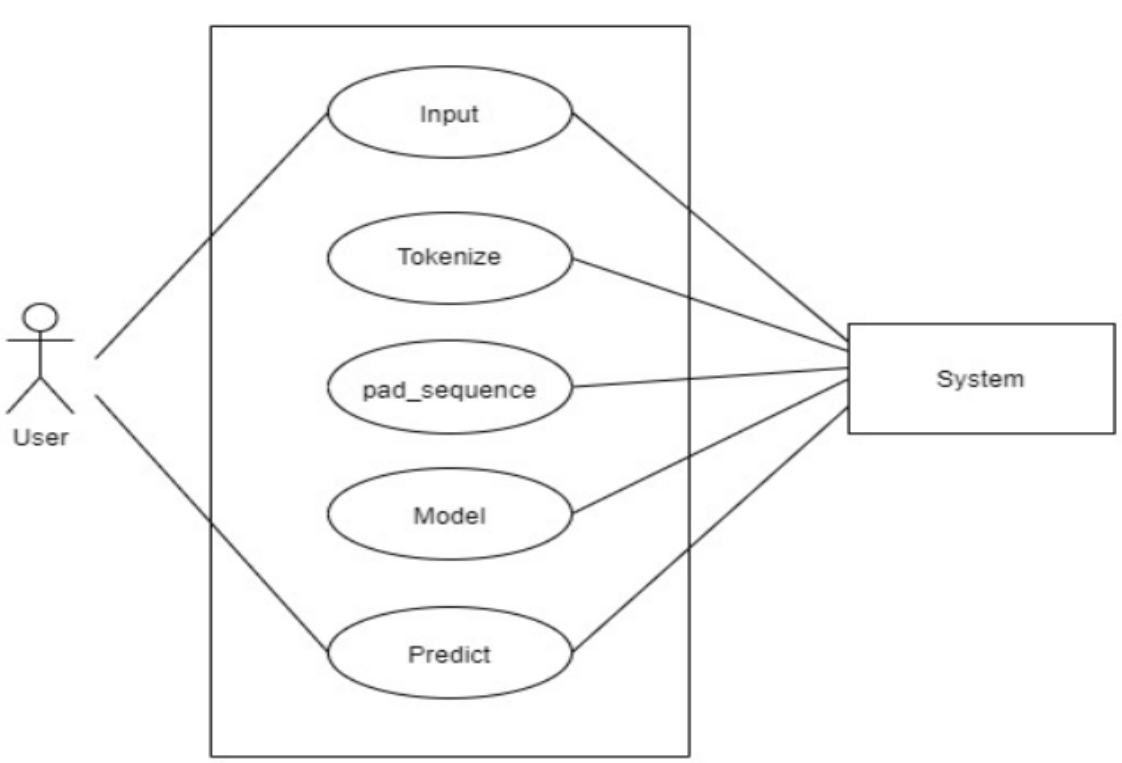


Figure 5.5: Use case Diagram for user input

A web page is created to provide the user friendly interface. The input can be taken in the form of text, voice and image. The user inputted text will be tokenized and the padding sequence will be calculated. This data will be fed into the machine learning model. The predicted result will be shown in the web page. If the user has depression, it will redirect to the page containing the solution to depression.

5.5 Flow chart

The user is allowed to input the status through text, voice or image on the web page provided. If the input from the user is an image, the texts from the image will be extracted. The input from the user will be fed into the machine learning model. The model will calculate the weight of positive and negative words in the sentence. If the weight of negative words is high, then the sentence will be classified as depressive. If the weight of positive words is high, then the sentence will be classified as non-depressive. Or else, the sentence will be known as neutral. The result will be displayed on the web page. If the result is negative, then the user input will be splitted and each word is compared with the solution dataset. The proper solution will be displayed on the page.

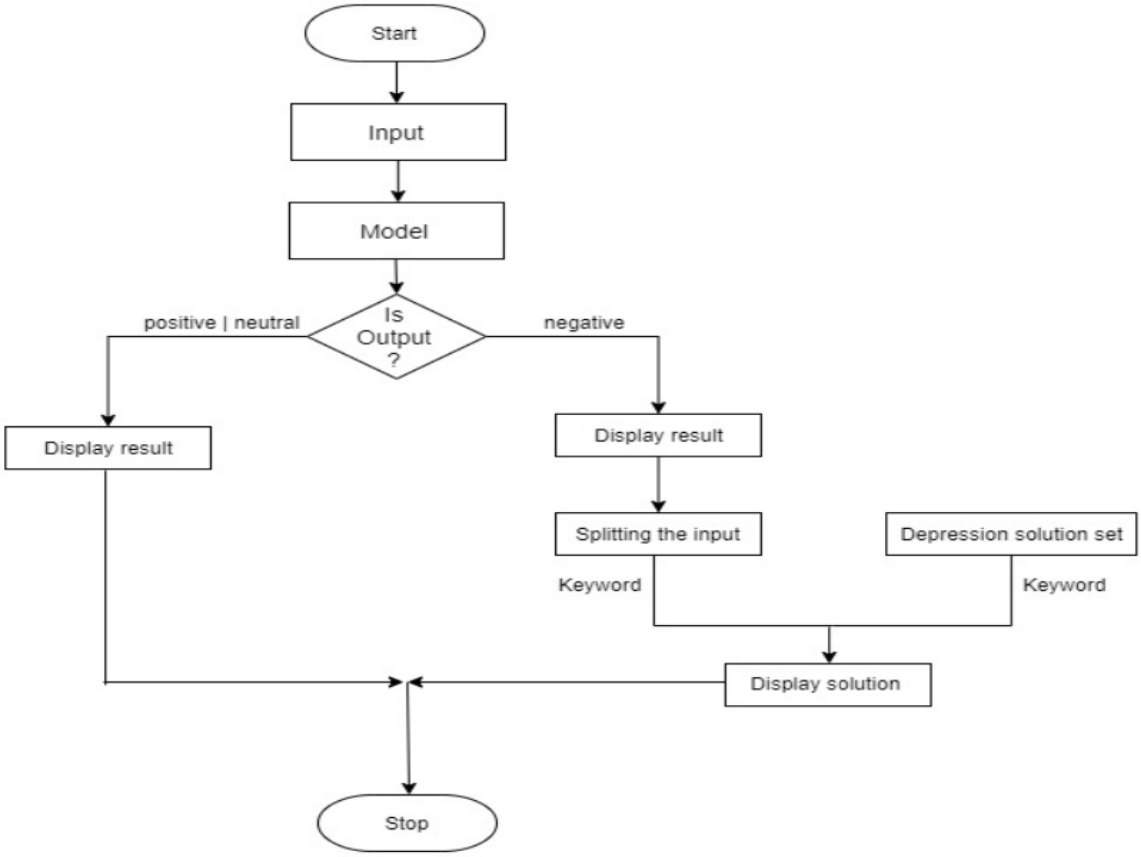


Figure 5.6: Flow chart

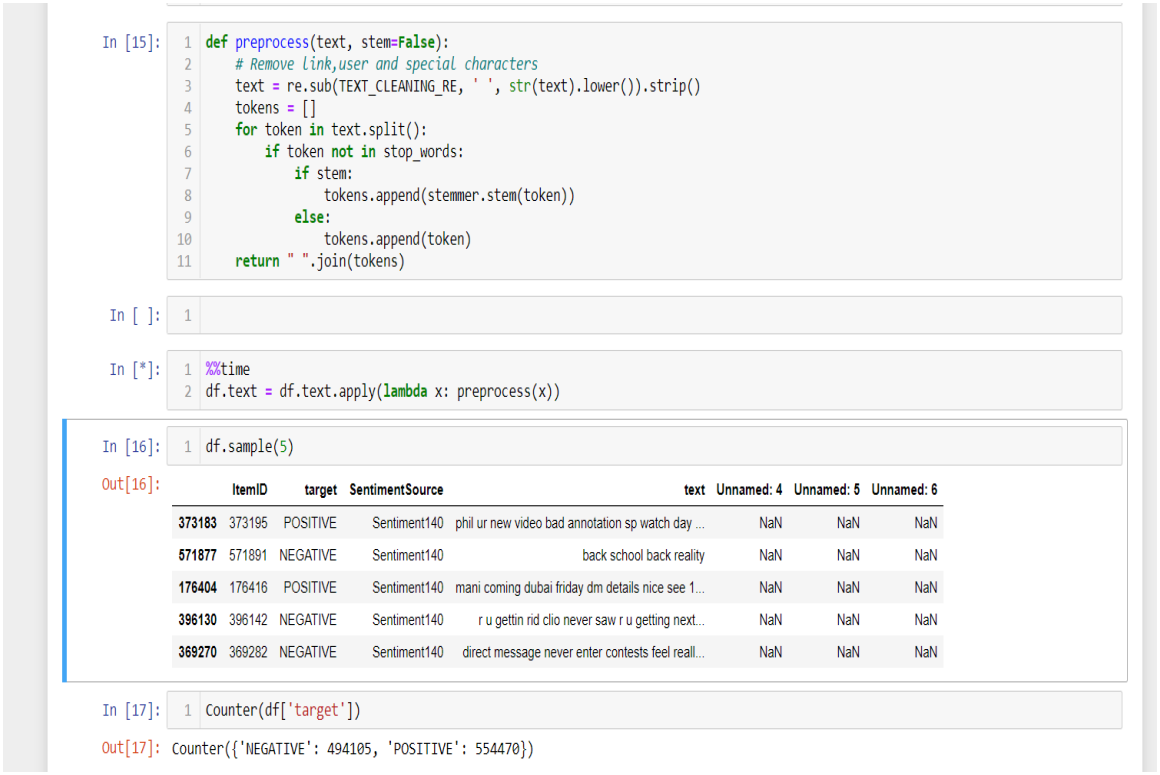
Chapter 6

Implementation

Implementation plays the most important part in any project report. Implementation is used to execute, or practice of a plan, a method, design, model, specification, idea standard or policy for doing something. The implementation phase is when the end user interacts with the product.

Our model mainly works on the input given by the user such as text, speech, and image. It categorize the output as positive, neutral and negative. If the output is negative it will suggest the solution. If the given solution is not satisfied then it redirects to the web page containing various solutions.

The first phase in implementation consists of collection and pre-processing of data. The collection of data is done from kaggle which consists of the scraped twitter comments of various users. The collected data may contain some special characters and some unwanted contents which is pre-processed using regular expression. Then stemming is performed to obtain root word.



```
In [15]:
def preprocess(text, stem=False):
    # Remove link,user and special characters
    text = re.sub(TEXT_CLEANING_RE, ' ', str(text).lower()).strip()
    tokens = []
    for token in text.split():
        if token not in stop_words:
            if stem:
                tokens.append(stemmer.stem(token))
            else:
                tokens.append(token)
    return " ".join(tokens)
```

```
In [ ]:
```

```
In [*]:
%%time
df.text = df.text.apply(lambda x: preprocess(x))
```

```
In [16]:
df.sample(5)
```

Out[16]:

	ItemID	target	SentimentSource	text	Unnamed: 4	Unnamed: 5	Unnamed: 6
373183	373195	POSITIVE	Sentiment140	phil ur new video bad annotation sp watch day ...	NaN	NaN	NaN
571877	571891	NEGATIVE	Sentiment140	back school back reality	NaN	NaN	NaN
176404	176416	POSITIVE	Sentiment140	mani coming dubai friday dm details nice see 1...	NaN	NaN	NaN
396130	396142	NEGATIVE	Sentiment140	r u gettin rid clio never saw r u getting next...	NaN	NaN	NaN
369270	369282	NEGATIVE	Sentiment140	direct message never enter contests feel reall...	NaN	NaN	NaN

```
In [17]:
Counter(df['target'])
```

```
Out[17]: Counter({'NEGATIVE': 494105, 'POSITIVE': 554470})
```

Figure 6.1: Snapshot of pre-processing the dataset

Pre-processed data is spitted into bag of word. Here each sentence is split into list of words. Word2Vec model is created using the list of words.

```
In [19]:
%%time
documents = [_text.split() for _text in df_train.text]
documents[0:5]
```

```
Wall time: 1.91 s
```

```
Out[19]: [['lt', '3', 'shall', 'continue', 'following', 'jorge', 'joyce', 'gave'],
 ['well',
  'get',
  'uncomfortable',
  'audience',
  'members',
  'go',
  'stage',
  'tonys'],
 ['problem', 'ses', 'wifi'],
 ['stomach', 'flu', 'fucking', 'spectacular'],
 ['guess', 'mine']]
```

```
In [20]:
w2v_model = gensim.models.word2vec.Word2Vec(size=W2V_SIZE,
                                            window=W2V_WINDOW,
                                            min_count=W2V_MIN_COUNT,
                                            workers=8)
```

```
In [21]:
w2v_model.build_vocab(documents)
```

```
2020-03-09 15:28:17,958 : INFO : collecting all words and their counts
2020-03-09 15:28:17,959 : INFO : PROGRESS: at sentence #0, processed 0 words, keeping 0 word types
2020-03-09 15:28:17,987 : INFO : PROGRESS: at sentence #10000, processed 71201 words, keeping 13744 word types
2020-03-09 15:28:18,021 : INFO : PROGRESS: at sentence #20000, processed 142042 words, keeping 21348 word types
2020-03-09 15:28:18,061 : INFO : PROGRESS: at sentence #30000, processed 213255 words, keeping 27388 word types
2020-03-09 15:28:18,092 : INFO : PROGRESS: at sentence #40000, processed 284100 words, keeping 32605 word types
2020-03-09 15:28:18,122 : INFO : PROGRESS: at sentence #50000, processed 356026 words, keeping 37406 word types
2020-03-09 15:28:18,159 : INFO : PROGRESS: at sentence #60000, processed 427275 words, keeping 41772 word types
2020-03-09 15:28:18,201 : INFO : PROGRESS: at sentence #70000, processed 498215 words, keeping 45865 word types
2020-03-09 15:28:18,232 : INFO : PROGRESS: at sentence #80000, processed 569873 words, keeping 49838 word types
```

Figure 6.2: Snapshot of pre-processing the dataset

After splitting the pre-processed dataset into training and testing set, tokenization is performed on the training data.

```
In [26]:
%%time
tokenizer = Tokenizer()
tokenizer.fit_on_texts(df_train.text)
print()
vocab_size = len(tokenizer.word_index) + 1
print("Total words", vocab_size)

Total words 218065
Wall time: 15.3 s

In [27]:
tokenizer=pickle.load(open('tokenizer.pkl', 'rb'))
```

Figure 6.3: Snapshot of tokenization of the text in dataset

After tokenization is performed, embedding matrix is created using word2vec model.

```
In [35]:
embedding_matrix = np.zeros((vocab_size, W2V_SIZE))
for word, i in tokenizer.word_index.items():
 if word in w2v_model.wv:
    embedding_matrix[i] = w2v_model.wv[word]
print(embedding_matrix.shape)

(218065, 300)
```

Figure 6.4: Snapshot of creation of embedding matrix

Next phase of implementation involves in building a sequential model for depression detection. This model consists of embedding layer which has 65419500 parameters, convolution layer which has 28832 parameters, LSTM layer which has a 399600 parameters and dense layer which has 301 parameters. To prevent overfitting of data, dropout layer is added.

```
In [39]:
model = Sequential()
# Embedded Layer
model.add(Embedding(len(embedding_matrix), 300, weights=[embedding_matrix],
                    input_length=300, trainable=False))
# Convolutional Layer
model.add(Conv1D(filters=32, kernel_size=3, padding='same', activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.2))
# LSTM Layer
model.add(LSTM(300))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))

In [40]:
model.summary()

Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 300, 300)          65419500
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 300, 32)           28832
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 150, 32)           0
_________________________________________________________________
dropout_1 (Dropout)          (None, 150, 32)           0
_________________________________________________________________
lstm_1 (LSTM)                (None, 300)               399600
_________________________________________________________________
dropout_2 (Dropout)          (None, 300)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 301
=================================================================
Total params: 65,848,233
Trainable params: 428,733
Non-trainable params: 65,419,500
```

Figure 6.5: Snapshot of building the model

The model will return the threshold value based on the sentimental analysis performed on the given text. The model returns the score for the given text where comparison is done with the threshold value to determine whether the given text is positive, neutral or negative.

Predict

In [42]:
```
def decode_sentiment(score, include_neutral=True):
    if include_neutral:
        label = NEUTRAL
        if score <= SENTIMENT_THRESHOLDS[0]:
            label = NEGATIVE
        elif score >= SENTIMENT_THRESHOLDS[1]-0.1:
            label = POSITIVE

        return label
    else:
        return NEGATIVE if score < 0.5 else POSITIVE
```

In [43]:
```
def predict1(text, include_neutral=True):
    start_at = time.time()
    # Tokenize text
    x_test = pad_sequences(tokenizer.texts_to_sequences([text]), maxlen=SEQUENCE_LENGTH)
    #print(x_test)
    # Predict
    score = model.predict([x_test])[0]
    print(score)
    # Decode sentiment
    label = decode_sentiment(score, include_neutral=include_neutral)

    return {"label": label, "score": float(score),
       "elapsed_time": time.time()-start_at}
```

Figure 6.6: Snapshot of method to decode a text

Chapter 7

System Testing

7.1 Introduction

Testing is a procedure of executing the program with unequivocal intension of discovering mistakes, assuming any, which makes the program, fall flat. This stage is an essential piece of the product improvement.

It plays out an exceptionally basic part for quality affirmation and for guaranteeing unwavering quality of programming. It is the way toward finding the mistakes and missing operation and furthermore an entire confirmation to decide if the targets are met the client prerequisites are fulfilled.

The objective of testing is to reveal prerequisites, outline or coding blunders in the projects. Therefore, unique levels of testing are utilized in programming frameworks. The testing results are utilized amid upkeep.

This area manages the points of interest in the various classes of the test which should be directed to approve capacities, imperatives and execution. This can be accomplished fundamentally by using the methods for testing, which assumes a crucial part in the improvement of a product.

The structure of the program is not being considered in useful testing. Test cases are exclusively chosen on the premise of the prerequisites or particulars of a program or module of program but the internals of the module or the program are not considered for determination of experiments.

The program to be tried is executed with an arrangement of experiments and the yield of the program for the experiments is assessed to decide whether the program is executing not surprisingly. The accomplishment of testing in uncovering mistakes in projects depends basically on the experiments. There are two fundamental ways to deal with testing Black

Box and White Box.

7.2 Testing Objectives

- To verify the interaction between objects.
- To verify proper integration of all the components of the software.
- To verify all the requirements have been correctly implemented.
- To identify and ensure defects are detected before the implementation of the software.

7.3 Test Approach

7.3.1 Black Box Testing

Black box testing or functional testing allows testing the errors that are conducted at the software interface. Itis approaches were the tests are derived from theprogram specification. In black box testing examination of some fundamental aspect of a system with little regard for the internal logical structure of the system. The system which is a black box whose behavior can be found out by studying its inputs and related output. This approach is equally applicable to systems that are organized as functions.

7.3.2 White Box Testing

White box testing strategy works with structure and internal logic of the code. White box testing is also recognized as glass, structural, open box or clear box testing. It mainly focuses on validating the flow of inputs and outputs through the application, improving design and usability, strengthening security.

7.4 Testing Strategies

7.4.1 Unit testing

Unit testing can be used to test a unit as a whole. This would test the interaction of many functions but combine the test within one unit. The scope of a unit is left to interpretation. Supporting test code, sometimes called scaffolding, maybenecessary to support an individual test. This type of testing is driven by architecture and implementation teams .

This type of testing is also called black box testing because of the details of interface are visible to the test. Global units are tested here with limits.

7.4.2 Integration Testing

One of the important phase of the software development project is the integration strategy. Integration can be done all at once , top-down, bottom-up critical piece first, or else by first integrating functioning sub-systems and then integrating the subsystems in separate phases using any one of the basic strategies. UsuaIly, the larger the project the more important is the integration strategy.

7.4.3 System Testing

System testing is performed on complete system, or on the integrated system to evaluate whether the system meets the specific requirement. System testing falls under the scope of black, so it should not require knowledge of the inner design of the code or logic.

7.4.4 Security Testing

The process of determining that an Information System can be used to protect the data and maintains functionality as intended. The six basic concepts in security testing are: confidentiality, integrity, authentication, authorization, availability and non-repudiation.

7.4.5 Validation Testing

This testing method focuses on the need for consistent and through quality assurance processes,and its standards or its guidelines. It is Implemented through test planning and management, and it can also be independently executed software testing process becomes a important element in ensuring quality in mission-critical projects and the long-term success of any organization.

Table 7.1: Work Flow

Sl No	Work	Duration(in Weeks)
1	Literature Survey	2
2	Dataset collection from various sources	2
3	Information collection on the different libraries used	1
4	Analysis of dataset	2
5	Model building	6
6	Developement of front end	2
7	Testing	2

Table 7.2: Test cases

TC#	Description	Expected Result	Actual Result	Status
TC-1	I am feeling happy	Positive	Positive	Pass
TC-2	Juuussst Chillin!!	Positive	Positive	Pass
TC-3	thanks to all the my face all day!	Positive	Positive	Pass
TC-4	congrats to helio though	Positive	Positive	Pass
TC-5	Just got school	Neutral	Neutral	Pass
TC-6	Hand quilting it is then	Neutral	Neutral	Pass
TC-7	I am going to die today	Negative	Negative	Pass
TC-8	I am depressed today	Negative	Negative	Pass
TC-9	Total tragedy of my life	Negative	Negative	Pass
TC-10	My situation is bad	Negative	Negative	Pass

Chapter 8

Results and Disscussion

The following are the results obtained from our project.

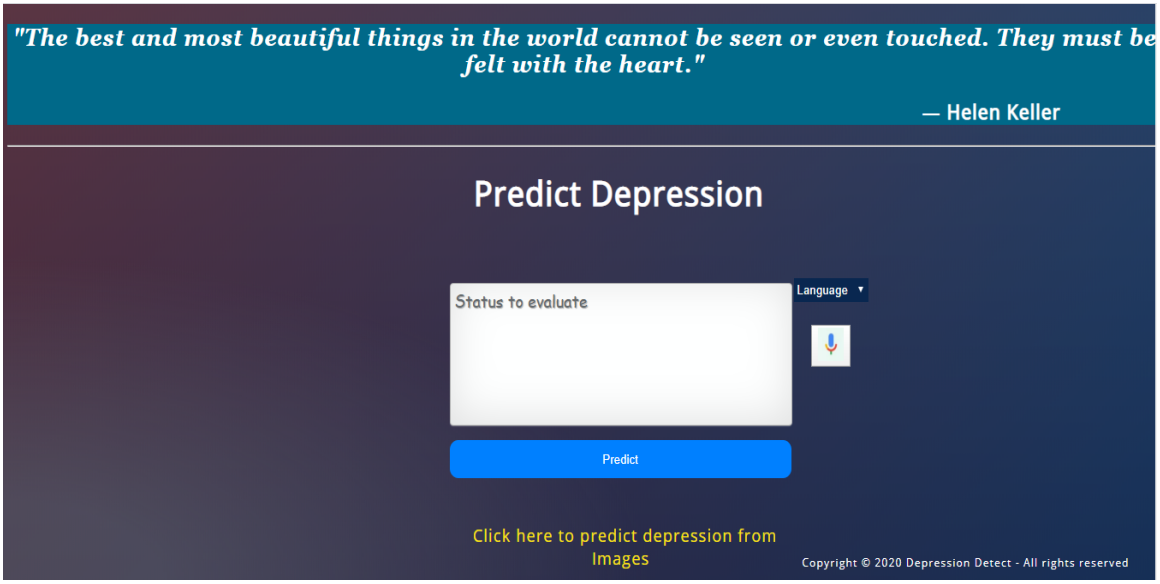


Figure 8.1: Home Page of the Project

The Home Page consists of field to enter various types of input and Predict button. The inputs may include text, voice or image. The text and voice input can be of any language. Some of the languages compatible are English, Hindi, Kannada, Malayalam, Tamil and Telugu.

Images that contain text can also be given as input. Any language input given by the user is internally converted into English and fed into the Depression Detection model. The user input is validated before it is processed further.

Following are the images that show various types of input given to the project.

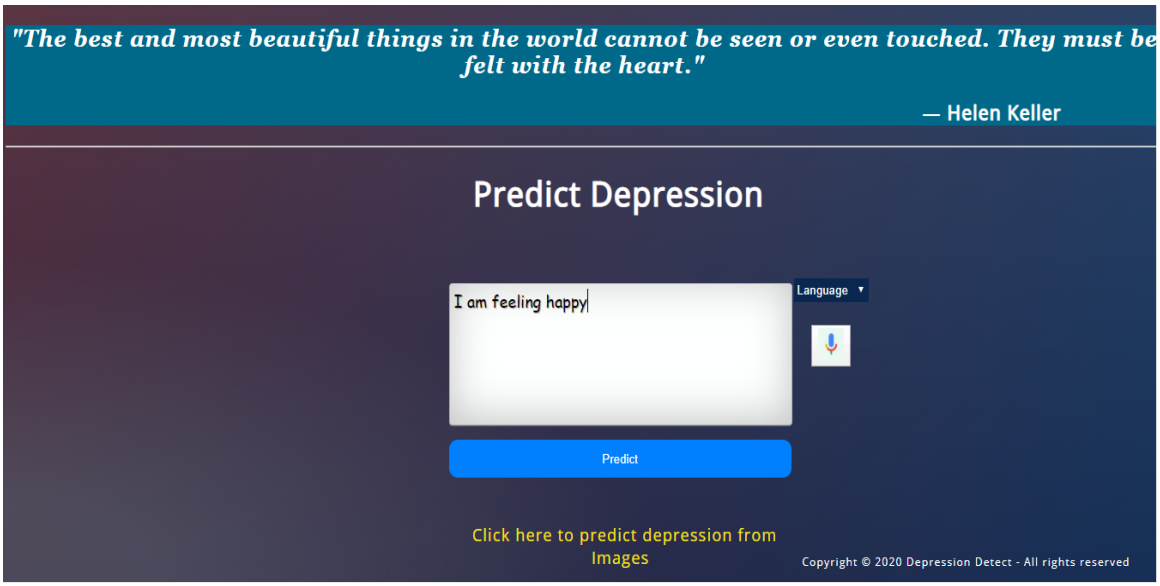


Figure 8.2: Text Input given to the project

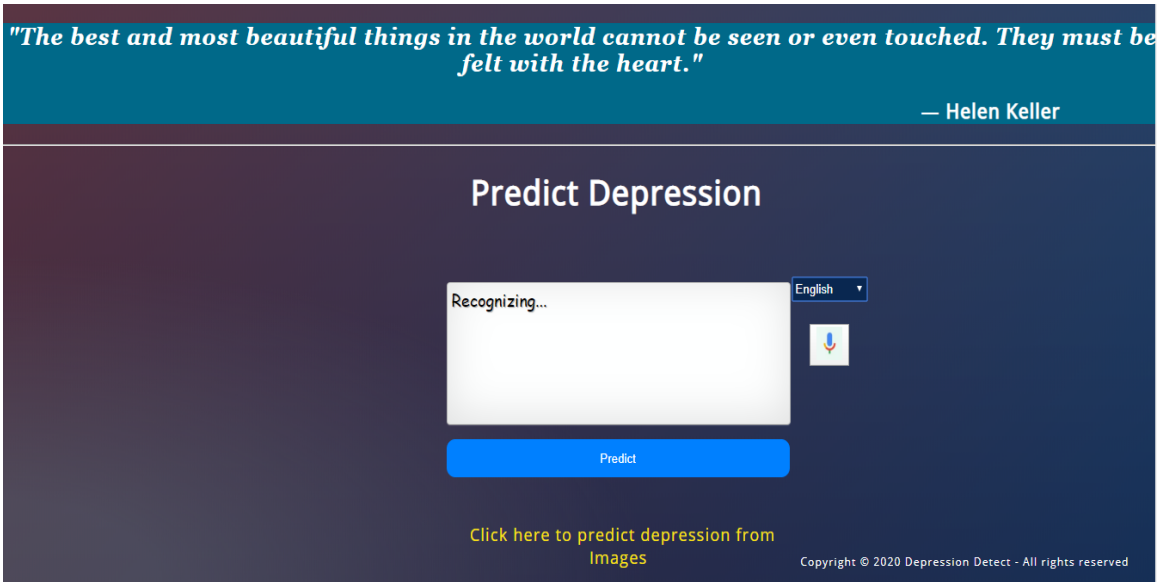


Figure 8.3: Voice Input given to the project

The conversion of voice input from the user to text is performed by using SpeechRecognition library. When microphone icon is pressed it will start recognizing the voice through microphone of the system.

Figure 8.4: Image Input given to the project

Extraction of text from the image is performed using PyTesseract library. The extracted text is fed into the model and output is predicted.

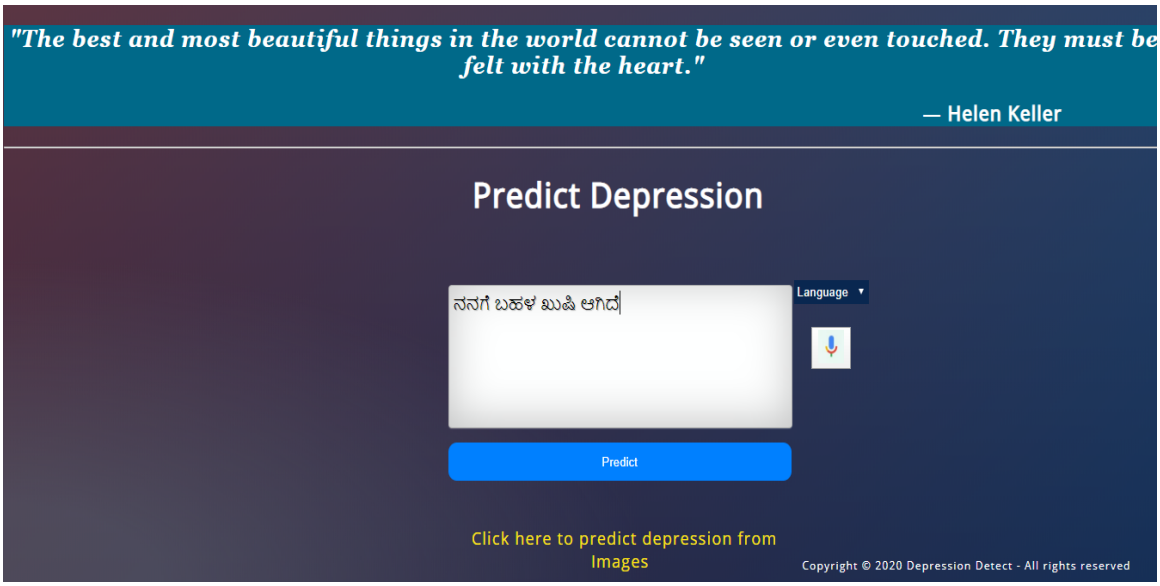


Figure 8.5: Text Input given in different languages

The user can input the text in any language. By using the library named GoogleTrans, which converts text input given in any language to English language. The above snapshot shows the text input given in Kannada langauge.

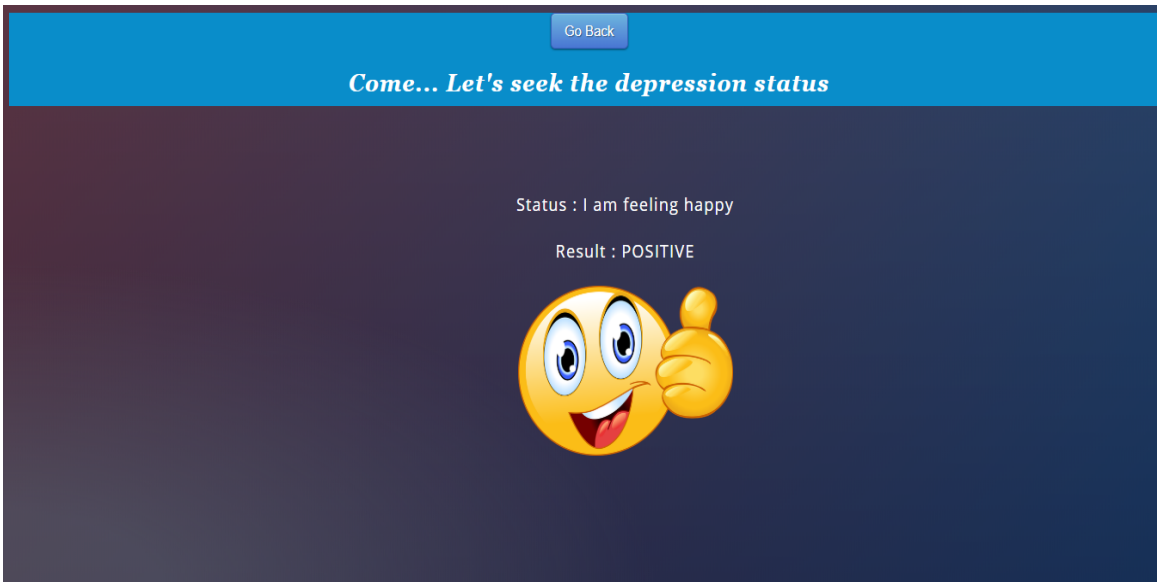


Figure 8.6: Snapshot showing positive output predicted by model

After the text input is given, the model gives a score based on the depression status of the input. Later this score is compared with the predefined threshold. If the predicted output is positive, the user will be redirected to the page where the positive result is shown.

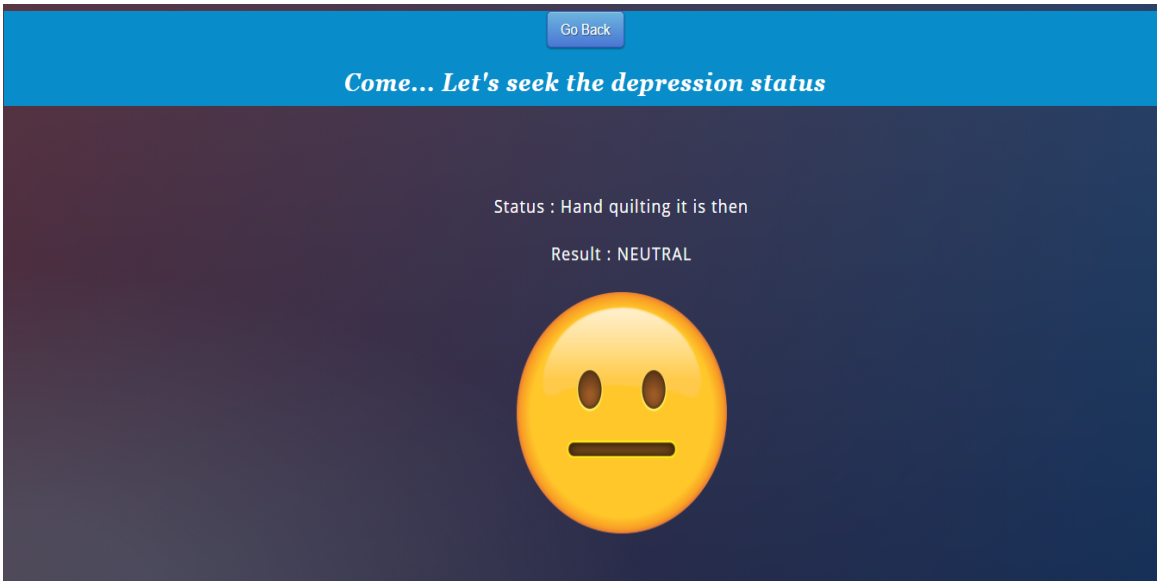


Figure 8.7: Snapshot showing neutral output predicted by model

If the calculated score lies between positive and negative threshold values, then the given input is predicted as neutral. If the predicted output is neutral, the user will be redirected to the page where the neutral result is shown.

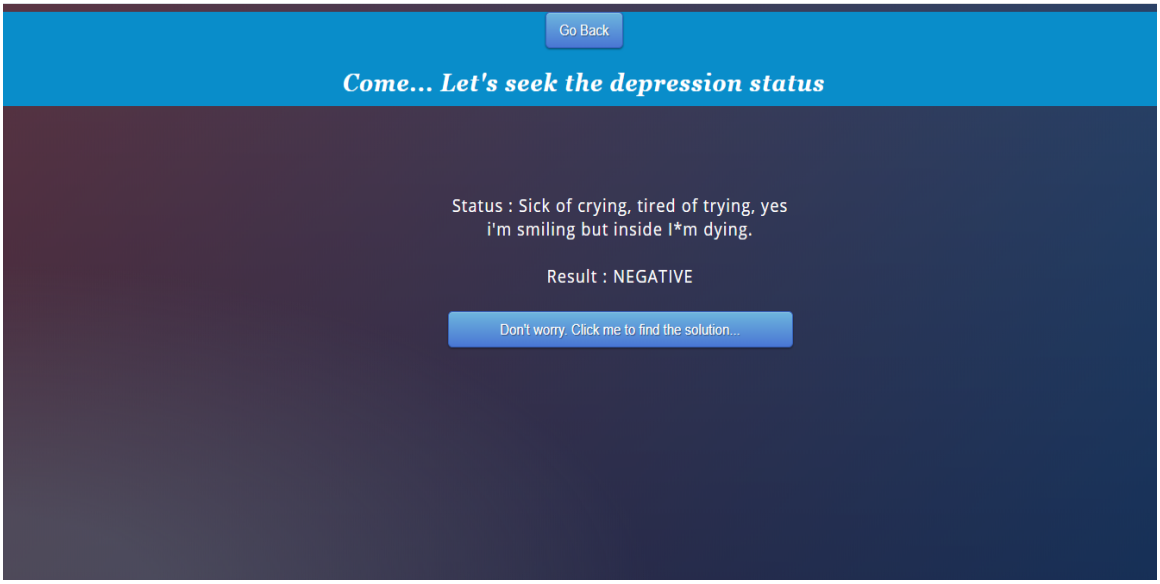


Figure 8.8: Snapshot showing negative output predicted by model

After the text input is given, the model gives a score based on the depression status of the input. Later this score is compared with the predefined threshold. If the predicted output is negative, the user will be redirected to the page where the negative result is shown. The user is given an option to find solution to the depression.

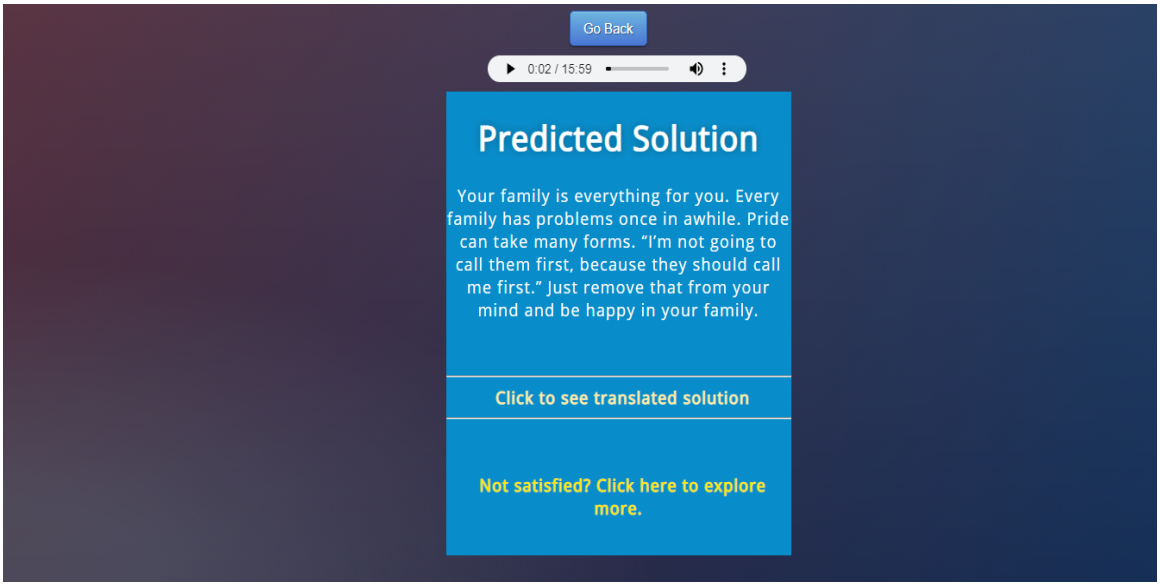


Figure 8.9: Snapshot suggesting the solution to the depressed user

If the predicted output is negative, then a solution will be suggested based on the depression category. A pleasant audio will be played in the background to supress the depression. If the suggested solution is not satisfied then the user can be redirected to website containing various depression solutions.

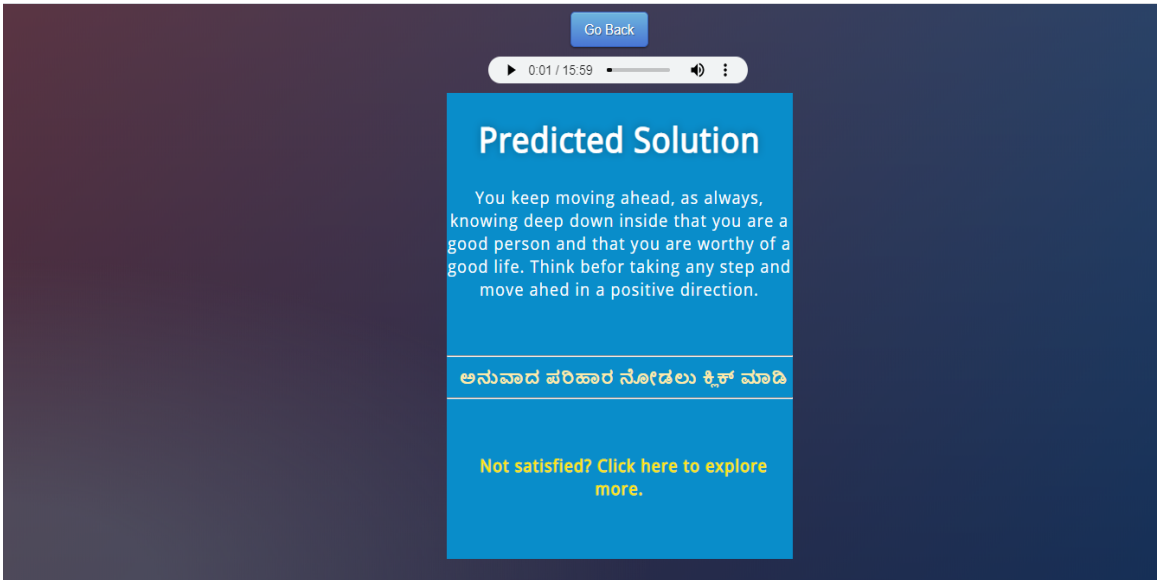


Figure 8.10: Snapshot suggesting the solution in user opted language

The solution suggested will be in the English language as well as the link will provided to the user in his opted language. In the above snapshot, the user has opted Kannada language. So the link for the translation appears in Kannada.

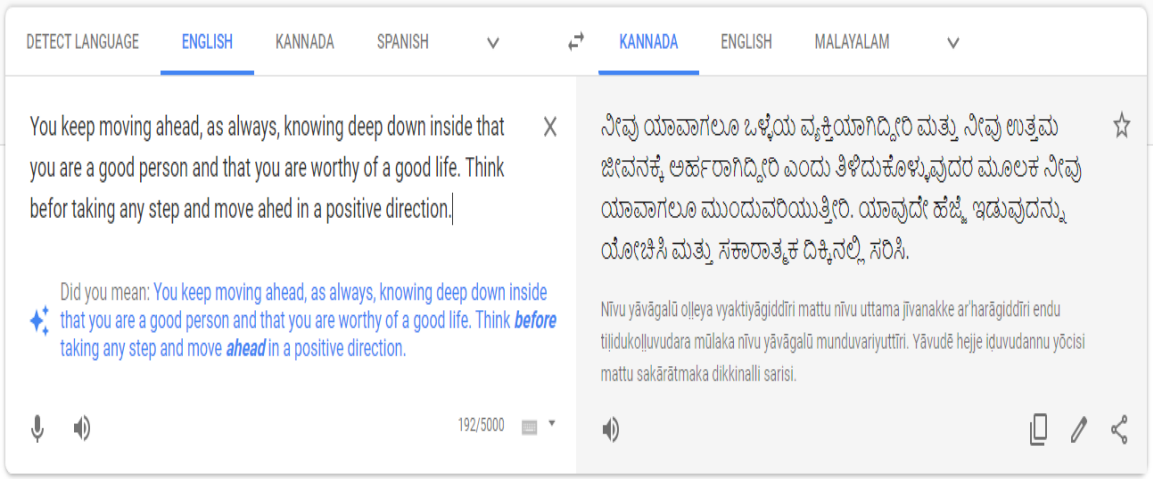


Figure 8.11: Snapshot showing translated solution in user opted language

The user can opt solution in any language. The above snapshot shows that the user has opted for Kannada language for suggested solution. A link will be provided for the solution translation. By clicking on the link, the user will be redirected to google translation website where he can view the solution in English language as well as the user opted language.

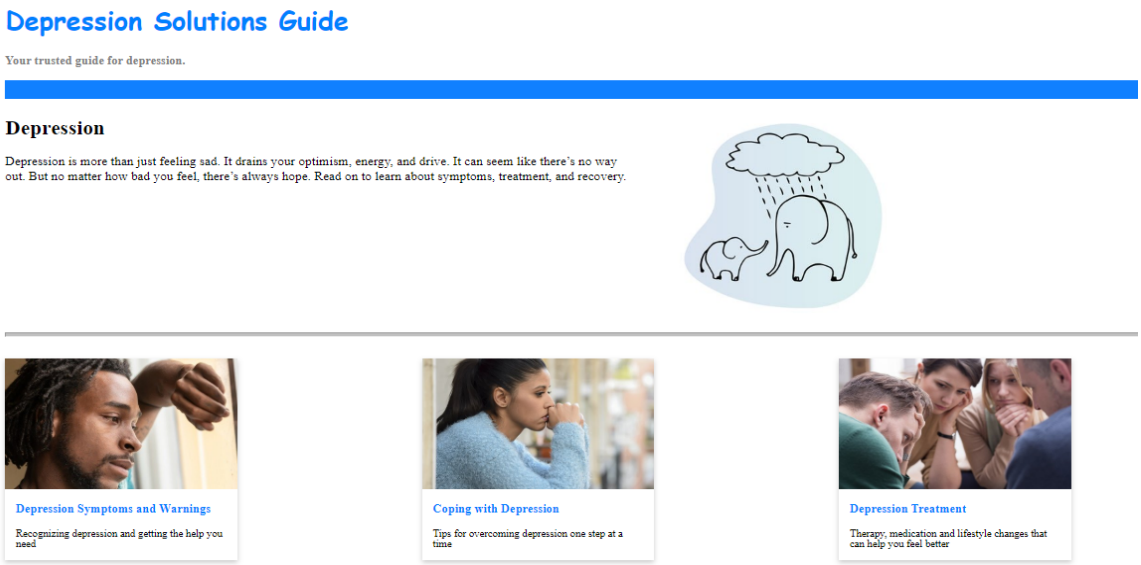


Figure 8.12: Snapshot showing home page of Depression solution website

If the suggested solution is not satisified to the user, then he can click to the link which redirects to the webpage containing the different categories of depression solution. The above snapshot shows the home page of the depression solution, which contains symptoms and treatments.

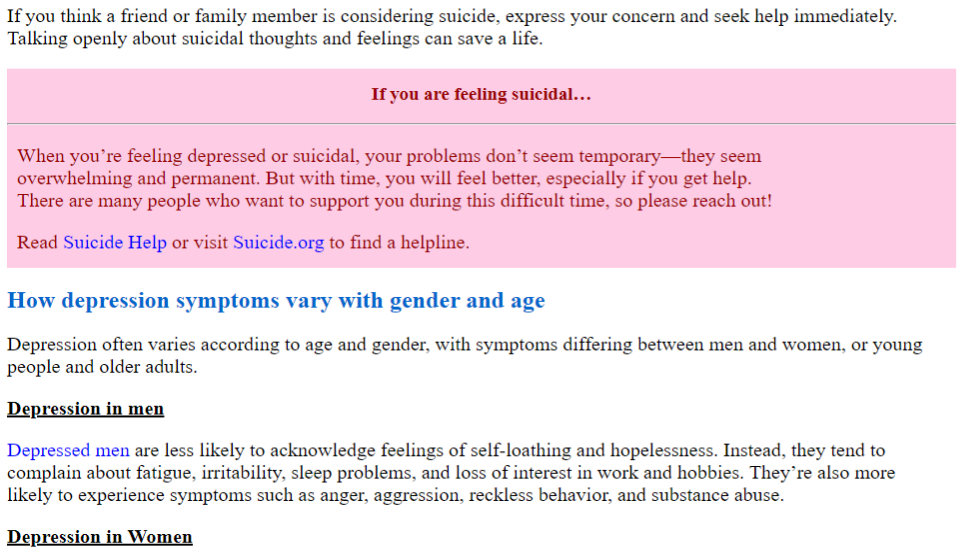


Figure 8.13: Snapshot showing sub-category of Depression solution

After choosing any of the given category from the home page of the depression solution website, it will redirect to choosen category which shows more details about it. In the above snapshot the user has choosen Depression symptoms and warnings as sub-category.

Chapter 9

Conclusion and Future work

The target is to determine whether the given text entered by user is depressive or not and to give solution to the user. A larger dataset to the model will ensure better reliability. Sentimental analysis has to be done to determining the status of the text whether it is positive or negative. The main aim is that it should give a result of 100 % accuracy. The future work to be carried out is the formation of more precise dataset to the train the model. After formation of dataset then feature extraction has to be done by performing word2vec and tokenization to the formation of the embedding matrix. Then the test set has to be given and compared with the dataset to determine the accuracy of the model. The given text has to be determine as positive ,negative or neutral by the model. Among all the nine paper it gives the maximum accuracy in identifying the given text. Our future work will be focusing on efficient model building and building better solution set to determine the solution to the depressive text.

References

[1] A. U. Hassan, J. Hussain, M. Hussain, M. Sadiq, and S. Lee, "Sentiment analysis of social networking sites (sns) data using machine learning approach for the measurement of depression," in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 138–140, IEEE, 2017.

[2] X. Tao, R. Dharmalingam, J. Zhang, X. Zhou, L. Li, and R. Gururajan, "Twitter analysis for depression on social networks based on sentiment and stress," in *2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC)*, pp. 1–4, IEEE, 2019.

[3] L. Ma, Z. Wang, and Y. Zhang, "Extracting depression symptoms from social networks and web blogs via text mining," in *International Symposium on Bioinformatics Research and Applications*, pp. 325–330, Springer, 2017.

[4] M. M. Tadesse, H. Lin, B. Xu, and L. Yang, "Detection of depression-related posts in reddit social media forum," *IEEE Access*, vol. 7, pp. 44883–44893, 2019.

[5] H.-W. Hu, K.-S. Hsu, C. Lee, H.-L. Hu, C.-Y. Hsu, W.-H. Yang, L.-y. Wang, and T.-A. Chen, "Keyword-driven depressive tendency model for social media posts," in *International Conference on Business Information Systems*, pp. 14–22, Springer, 2019.

[6] W. Li and M. Chau, "Applying deep learning in depression detection.," in *PACIS*, p. 333, 2018.

[7] I. Oyong, E. Utami, and E. T. Luthfi, "Natural language processing and lexical approach for depression symptoms screening of indonesian twitter user," in *2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pp. 359–364, IEEE, 2018.

[8] M. E. Aragón, A. P. López-Monroy, L. C. González-Gurrola, and M. Montes, "Detecting depression in social media using fine-grained emotions," in *Proceedings of the 2019*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 1481–1486, 2019.

Appendix