A Project Report on

Online Detection of Depression Related Posts In Social Media

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

In

Computer Science & Engineering

By

K. DEEPTHI (193T1A0550)
K. SHIREESHA (193T1A0555)
K. SAAHITHYA (193T1A0553)
B. SONY (193T1A0506)

Under the esteemed guidance of

MS. P. PRIYANKA

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAVINDRA COLLEGE OF ENGINEERING FOR WOMEN

(Approved by AICTE – New Delhi, Affiliated to JNTU, Anantapur)
Kurnool-518002, Andhra Pradesh
Accredited by NAAC with A+ grade and Accredited by NBA
Near Venkayapalle, Pasupula Village, Nandikotkur Road, Kurnool – 518452, Andhra Pradesh.

May -2023

RAVINDRA COLLEGE OF ENGINEERING FOR WOMEN

(Approved by AICTE – New Delhi, Affiliated to JNTU, Anantapur)
Kurnool-518002, Andhra Pradesh
Accredited by NAAC with A+ grade and Accredited by NBA
Near Venkayapalle, Pasupula Village, Nandikotkur Road, Kurnool – 518452, Andhra Pradesh.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

The project report entitled **“ONLINE DETECTION OF DEPRESSION RELETED POSTS IN SOCIAL MEDIA”** is prepared and submitted by **K.DEEPTHI (193T1A0550), K.SHIREESHA (193T1A0555), K.SAAHITHYA (193T1A0553), B.SONY (193T1A0506) .**It has been found satisfactory in terms of scope, quality and presentation as partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** in Ravindra College of Engineering for Women, Kurnool, AP.

Signature of Project Guide. **Head of the Department.**

Internal Examiner. **External Examiner.**

RAVINDRA COLLEGE OF ENGINEERING FOR WOMEN

(Approved by AICTE – New Delhi, Affiliated to JNTU, Anantapur)
Kurnool-518002, Andhra Pradesh
Accredited by NAAC with A+ grade and Accredited by NBA
Near Venkayapalle, Pasupula Village, Nandikotkur Road, Kurnool – 518452, Andhra Pradesh.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We hereby declare that the project entitled **"Online Detection Of Depression Related Posts In Social Media"** submitted by us to the Department of Computer Science and Engineering, Ravindra College of Engineering for women, Kurnool, in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out by us under the supervision of **Ms. P. Priyanka,** Assistant professor, Department of Artificial Intelligence**.** We further declare that the work reported in this project work has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or any other institute or university.

Signatures of the Candidates

K. DEEPTHI (193T1A0550)

K. SHIREESHA (193T1A0555)

K. SAAHITHYA (193T1A0553)

B. SONY (193T1A0506)

ACKNOWLEDGEMENTS

We would like to thank Management and Chairman Sri **G.V.M. Mohan Kumar,** giving me an opportunity to work on this project report and supporting me throughout the period of work at Ravindra College of Engineering for Women, Kurnool.

We express our gratitude to **Dr.K.E. Srinivasa Murthy,** Principal of Ravindra College of Engineering for Women who permitted us to carry the project work as part of the academics.

We would like to thank **Mr.K. Mahesh Babu,** HOD for department of Computer Science and Engineering for his support and encouragement in completing our project.

We express our profound gratitude to our guide **Ms.P. Priyanka,** AssistantProfessor Department of Artificial Intelligence of Ravindra College of Engineering for women, by providing required information. Without her guidance, cooperation and encouragement we could not complete our project work.

Finally, we wish to take this opportunity to express our deep gratitude to our family members and all the people who have extended their cooperation in various ways during our project work. It is our pleasure to acknowledge the help of all those individuals.

.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

SVM	Support Vector Machine
MLP	Multilayer Perceptron
LR	Logistic Regression
LIWC	Linguistic Inquiry and Word Count
LDA	Linear Discriminant Analysis
RF	Random Forest
TF-IDF	Term Frequency-Inverse Document Frequency

LIST OF TABLES

Abstract

In this project we are detecting depression from users post, user can upload post in the form of text file, image file or audio file, this project can help peoples who are in depression by sending motivated messages to them. Now-a-days peoples are using online post services to interact with each other compare to human-to-human interaction. So by analyzing users post this application can detect depression and send motivation messages to them. Administrator of this application will send motivation messages to all peoples who are in depression. To detect depression we are using SVM (support vector machine) algorithm which analyze users post and give result as negative or positive. If users express depression words in post then SVM detect it as a negative post else positive post. Detecting depression through online posts is an innovative way to help people who may be struggling with mental health issues. With the increasing use of online communication, it's essential to leverage technology to address important health issues such as depression. This project aims to detect depression in users' posts, regardless of the format, and provide motivational messages to users who may need them.

The project relies on the SVM algorithm to analyze users' posts and classify them as positive or negative. The algorithm analyzes the words used in the posts to identify depressive expressions and provide an accurate diagnosis. This approach is beneficial as it enables the application to reach a large number of users while preserving their anonymity.

CHAPTER 1

INTRODUCTION

CHAPTER 1
INTRODUCTION

1.1 Overview of the project

In this project we are detecting depression from users post, user can upload post in the form of text file, image file or audio file, this project can help peoples who are in depression by sending motivated messages to them. Now-a-days peoples are using online post services to interact with each other compare to human to human interaction. So by analysing users post this application can detect depression and send motivation messages to them. Administrator of this application will send motivation messages to all peoples who are in depression. To detect depression we are using SVM (support vector machine) algorithm which analyses users post and give result as negative or positive. If users express depression words in post then SVM detect it as a negative post else positive post.

1.2 Objective of the project

To implement this project we are using python Speech Recognition API which will read text from audio files and then SVM will analyze that text to detect depression, user can also upload images via post and python Tesseract OCR (Optical Character Recognition) API can read text from uploaded image and then SVM will detect depression from that text, User can upload post in text file also the goal of this project is to provide early intervention and support for individuals who may be at risk of depression, and to help healthcare professionals and researchers better understand the prevalence and nature of depression in online communities. By identifying and flagging posts that may indicate depression, the system could help connect users with resources and support services, as well as provide valuable data for further research into the causes and treatment of depression. The main objective of the project is to develop a system that can accurately detect depression-related posts on the Reddit social media forum. This involves using machine learning and natural language processing techniques to analyze the text of posts and comments in the forum, with the goal of identifying

posts that suggest the presence of depression or related mental health issues. There are several potential benefits to developing such a system. For one, it could help to identify users who may be struggling with depression or related mental health issues and connect them with resources, support, and treatment options. This could potentially save lives, as depression is a leading cause of suicide

Another potential benefit is that by analyzing patterns in the language and content of depression-related posts, the system could provide insights into the experiences and needs of individuals with depression or related mental health conditions. This could inform the development of more effective interventions and treatments for depression.

In addition, the system could be used by mental health professionals, researchers, and public health officials to monitor trends in depression-related posts on Reddit and other social media forums. This could help to identify emerging issues and trends related to depression, which could inform public health policy and research efforts.

Overall, the goal of the project is to leverage the power of machine learning and natural language processing to improve our ability to detect and respond to depression-related posts on social media, with the ultimate aim of improving outcomes for individuals with depression and related mental health conditions.

The object of the project is to develop a machine learning model that can detect depression-related posts in the Reddit social media forum. The model would analyze the language used in posts and comments to identify potential signs of depression, such as feelings of sadness, hopelessness, or isolation.

The project's ultimate goal is to provide a tool that can help moderators and mental health professionals monitor and support individuals who may be struggling with depression. By identifying these individuals early on, the model could potentially help connect them with appropriate resources and support services.

Depression is a common mental health condition that affects millions of people worldwide. Many individuals may not seek professional help for their symptoms, which can lead to increased feelings of isolation and worsening mental health outcomes. Social media platforms like Reddit provide an opportunity for individuals to anonymously share their experiences and connect with others who may be going through similar challenges.

However, identifying individuals who may be struggling with depression in a social media forum can be challenging, as not all individuals may openly discuss their symptoms or seek help. This is where a machine learning model could potentially help. By analyzing the language used in posts and comments, the model could flag potential signs of depression, alerting moderators and mental health professionals to intervene and provide support

1.2 EXISTING SYSTEM

- Detecting depression from users post, user can upload post in the form of text file, image file or audio file, this project can help peoples who are in depression by sending motivated messages to them. Now-a-days peoples are using online post services to interact with each other compare to human to human interaction.
- we aim to analyze Facebook data to detect any factors that may reflect the depression of relevant Facebook's users. Various machine learning techniques are employed for such purpose. Considering the key objective of this study, the following are subsequent research challenges addressed in paper.
- As users express their feeling as a post or comments in the Face-book platform, sometimes their posts and comments refer to as emotional state such as 'joy', 'sadness', 'fear', 'anger', or 'surprise'.
- These systems use machine learning algorithms to analyze large datasets of posts and comments and identify patterns and features associated with depression-related language. The algorithms may be trained on labeled datasets of posts that have been manually annotated as either depression-related or not.

1.3 DISADVANTAGES:

- Reliable in classification of use.
- Easy to segregate and mark user as spam.
- The SVM algorithm segregates the posts in a less time.
- Existing Machine Learning algorithms are not accurate in the detection of depressed posts.
- They are poor in predicting the depression accurately.
- The System does not work for huge posts.

CHAPTER 2

LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

2.1 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is putforth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensurethat the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are :

- Economical Feasibility
- Technical Feasibility
- Operational Feasibility

2.1.1 Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developedsystem as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

2.1.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.1.3 Operational Feasibility

Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

2.2 Literature Survey

T. Nguyen, D. Phung, B. Dao, S. Venkatesh and M. Berk, "Affective and Content Analysis of Online Depression Communities," in *IEEE Transactions on Affective Computing*, vol. 5, no. 3, pp.217-226,1July-Sept.2014.
doi:10.1109/TAFFC.2014.2315623
Abstract: A large number of people use online communities to discuss mental health issues, thus offering opportunities for new understanding of these communities. This paper aims to study the characteristics of online depression communities (CLINICAL) in comparison with those joining other online communities (CONTROL). We use machine learning and statistical methods to discriminate online messages between depression and control communities using mood, psycholinguistic processes and content topics extracted from the posts generated by members of these communities. All aspects including mood, the written content and writing style are found to be significantly different between two types of communities. Sentiment analysis shows the clinical group have lower valence than people in the control group. For language styles and topics, statistical tests reject the hypothesis of equality on psycholinguistic processes and topics between two groups. We show good predictive validity in depression classification using topics and psycholinguistic clues as features. Clear discrimination between writing styles and contents, with good predictive power is an important step in understanding social media and its use in mental health.
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6784326&isnumber=6933964
B. Xu, Y. Ye and L. Nie, "An improved random forest classifier for image classification," *2012 IEEE International Conference on Information and Automation*, Shenyang, China, 2012, pp. 795-800.

doi:10.1109/ICInfA.2012.6246927

Abstract: This paper proposes an improved random forest algorithm for image classification. This algorithm is particularly designed for analyzing very high dimensional data with multiple classes whose well-known representative data is image data. A novel feature weighting method and tree selection method are developed and synergistically served for making random forest framework well suited to classify image data with a large number of object categories. With the new feature weighting method for subspace sampling and tree selection method, we can effectively reduce subspace size and improve classification performance without increasing error bound. Experimental results on image datasets with diverse characteristics have demonstrated that the proposed method could generate a random forest model with higher performance than

URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6246927&isnumber=6246755

M. M. Tadesse, H. Lin, B. Xu and L. Yang, "Personality Predictions Based on User Behavior on the Facebook Social Media Platform," in *IEEE Access*, vol. 6, pp. 61959-61969, 2018. doi:10.1109/ACCESS.2018.2876502

Abstract: With the development of social networks, a large variety of approaches have been developed to define users' personalities based on their social activities and language use habits. Particular approaches differ with regard to different machine learning algorithms, data sources, and feature sets. The goal of this paper is to investigate the predictability of the personality traits of Facebook users based on different features and measures of the Big 5 model. We examine the presence of structures of social networks and linguistic features relative to personality interactions using the my Personality project data set. We analyze and compare four machine learning models and perform the correlation between each of the feature sets and personality traits. The results for the prediction accuracy show that even if tested under the same data set, the personality prediction system built on the XGBoost classifier outperforms the average baseline for all the feature sets, with a highest prediction accuracy of 74.2%. The best prediction performance was reached for the extra version trait by using the individual social network analysis features set, which achieved a higher personality prediction accuracy of 78.6%.

URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8494744&isnumber=8274985

Z. -q. Wang, X. Sun, D. -x. Zhang and X. Li, "An Optimal SVM-Based Text Classification Algorithm," *2006 International Conference on Machine Learning and Cybernetics*, Dalian, China2006,pp.1378-1381.

doi:10.1109/ICMLC.2006.258708

Abstract: The goal of a text classification system is to determine whether a given document belongs to which of the predefined categories. An optimal SVM algorithm for text classification via multiple optimal strategies is proposed in this paper. The experimental results indicate that the proposed optimal classification algorithm yields much better performance than other conventional.

URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4028279&isnumber=4028022

L. -S. A. Low, N. C. Maddage, M. Lech, L. B. Sheeber and N. B. Allen, "Detection of Clinical Depression in Adolescents' Speech During Family Interactions," in *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 3, pp. 574-586, March 2011. doi:10.1109/TBME.2010.2091640

Abstract: The properties of acoustic speech have previously been investigated as possible cues for depression in adults. However, these studies were restricted to small populations of patients and the speech recordings were made during patients' clinical interviews or fixed-text reading sessions. Symptoms of depression often first appear during adolescence at a time when the voice is changing, in both males and females, suggesting that specific studies of these phenomena in adolescent populations are warranted. This study investigated acoustic correlates of depression in a large sample of 139 adolescents (68 clinically depressed and 71 controls). Speech recordings were made during naturalistic interactions between adolescents and their parents. Prosodic, cepstral, spectral, and glottal features, as well as features derived from the Teager energy operator (TEO), were tested within a binary classification framework. Strong gender differences in classification accuracy were observed. The TEO-based features clearly outperformed all other features and feature combinations, providing classification accuracy ranging between 81%–87% for males and 72%–79% for females. Close, but slightly less accurate, results were obtained by combining glottal features with prosodic and spectral features (67%–69% for males and 70%–75% for females). These findings indicate the importance of nonlinear mechanisms associated with the glottal flow formation as cues for clinical depression. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5629355&isnumber=5719227

A. Jan, H. Meng, Y. F. B. A. Gaus and F. Zhang, "Artificial Intelligent System for Automatic Depression Level Analysis Through Visual and Vocal Expressions," in *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 3, pp. 668-680, Sept. 2018. doi:10.1109/TCDS.2017.2721552.

Abstract: A human being's cognitive system can be simulated by artificial intelligent systems.

Machines and robots equipped with cognitive capability can automatically recognize a humans mental state through their gestures and facial expressions. In this paper, an artificial intelligent system is proposed to monitor depression. It can predict the scales of Beck depression inventory II (BDI-II) from vocal and visual expressions. First, different visual features are extracted from facial expression images. Deep learning method is utilized to extract key visual features from the facial expression frames. Second, spectral low-level descriptors and mel-frequency cepstral coefficients features are extracted from short audio segments to capture the vocal expressions. Third, feature dynamic history histogram (FDHH) is proposed to capture the temporal movement on the feature space. Finally, these FDHH and audio features are fused using regression techniques for the prediction of the BDI-II scales. The proposed method has been tested on the public Audio/Visual Emotion Challenges 2014 dataset as it is tuned to be more focused on the study of depression. The results outperform all the other existing methods on the sameDataset.

URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7997822&isnumber=8457402

Y. Zhu, Y. Shang, Z. Shao and G. Guo, "Automated Depression Diagnosis Based on Deep Networks to Encode Facial Appearance and Dynamics," in *IEEE Transactions on Affective Computing*, vol. 9, no. 4, pp. 578-584, 1 Oct.-Dec. 2018. doi:10.1109/TAFFC.2017.2650899

Abstract: As a severe psychiatric disorder disease, depression is a state of low mood and aversion to activity, which prevents a person from functioning normally in both work and daily lives. The study on automated mental health assessment has been given increasing attentions in recent years. In this paper, we study the problem of automatic diagnosis of depression. A new approach to predict the Beck Depression Inventory II (BDI-II) values from video data is proposed based on the deep networks. The proposed framework is designed in a two stream manner, aiming at capturing both the facial appearance and dynamics. Further, we employ joint tuning layers that can implicitly integrate the appearance and dynamic information. Experiments are conducted on two depression databases, AVEC2013 and AVEC2014. The experimental results show that our proposed approach significantly improve the depression prediction performance, compared to other.

URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7812588&isnumber=8547219W. C. de Melo, E. Granger and A. Hadid, "Combining Global and Local Convolutional 3D Networks for Detecting Depression from Facial Expressions," *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, Lille, France, 2019, pp. 1-8.

doi:10.1109/FG.2019.8756568

Abstract: Deep learning architectures have been successfully applied in video-based health monitoring, to recognize distinctive variations in the facial appearance of subjects. To detect patterns of variation linked to depressive behavior, deep neural networks (NNs) typically exploit

spatial and temporal information separately by, e.g., cascading a 2D convolutional NN (CNN) with a recurrent NN (RNN), although the intrinsic spatio-temporal relationships can deteriorate. With the recent advent of 3D CNNs like the convolutional 3D (C3D) network, these spatio-temporal relationships can be modeled to improve performance. However, the accuracy of C3D networks remain an issue when applied to depression detection. In this paper, the fusion of diverse C3D predictions are proposed to improve accuracy, where spatio-temporal features are extracted from global (full-face) and local (eyes) regions of subject. This allows to increasingly focus on a local facial region that is highly relevant for analyzing depression. Additionally, the proposed network integrates 3D Global Average Pooling in order to efficiently summarize spatio-temporal features without using fully-connected layers, and thereby reduce the number of model parameters and potential over-fitting. Experimental results on the Audio Visual Emotion Challenge (AVEC 2013 and AVEC 2014) depression datasets indicates that combining the responses of global and local C3D networks achieves a higher level of accuracy than state-of-the-artsystems.

URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8756568&isnumber=8756505 X. Geng, C. Yin and Z. -H. Zhou, "Facial Age Estimation by Learning from Label Distributions," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 10,pp.2401-2412,Oct.2013.doi:10.1109/TPAMI.2013.51

Abstract: One of the main difficulties in facial age estimation is that the learning algorithms cannot expect sufficient and complete training data. Fortunately, the faces at close ages look quite similar since aging is a slow and smooth process. Inspired by this observation, instead of considering each face image as an instance with one label (age), this paper regards each face image as an instance associated with a label distribution. The label distribution covers a certain number of class labels, representing the degree that each label describes the instance. Through this way, one face image can contribute to not only the learning of its chronological age, but also the learning of its adjacent ages. Two algorithms, named IIS-LLD and CPNN, are proposed to learn from such label distributions. Experimental results on two aging face databases show remarkable advantages of the proposed label distribution learning algorithms over the compared single-label learning algorithms, either specially designed for age estimation or for general purpose.

In this letter, we propose a deep cascaded multitask framework that exploits the inherent correlation between detection and alignment to boost up their performance. In particular, our framework leverages a cascaded architecture with three stages of carefully designed deep convolutional networks to predict face and landmark location in a coarse-to-fine manner. In addition, we propose a new online hard sample mining strategy that further improves the performance in practice. Our method achieves superior accuracy over the state-of-the-art techniques on the challenging face detection dataset and benchmark and WIDER FACE benchmarks for face detection, and annotated facial landmarks in the wild benchmark URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7553523&isnumber=7539715

L. Wen, X. Li, G. Guo and Y. Zhu, "Automated Depression Diagnosis Based on Facial Dynamic Analysis and Sparse Coding," in *IEEE Transactions on Information Forensics and Security*, vol. 10,no.7,pp.1432-1441,July2015.

doi:10.1109/TIFS.2015.2414392

Abstract: Depression is a severe psychiatric disorder preventing a person from functioning normally in both work and daily lives. Currently, diagnosis of depression requires extensive participation from clinical experts. It has drawn much attention to develop an automatic system for efficient and reliable diagnosis of depression. Under the influence of depression, visual-based behavior disorder is readily observable. This paper presents a novel method of exploring facial region visual-based nonverbal behavior analysis for automatic depression diagnosis. Dynamic feature descriptors are extracted from facial region subvolumes, and sparse coding is employed to implicitly organize the extracted feature descriptors for depression diagnosis. Discriminative mapping and decision fusion are applied to further improve the accuracy of visual-based diagnosis. The integrated approach has been tested on the AVEC2013 depression database and the best visual-based mean absolute error/root mean square error results have been achieved. Phase Quantization Histograms from Three Orthogonal

Abstract: One of the main difficulties in facial age estimation is that the learning algorithms cannot expect sufficient and complete training data. Fortunately, the faces at close ages look quite similar since aging is a slow and smooth process. Inspired by this observation, instead of considering each face image as an instance with one label (age), this paper regards each face image as an instance associated with a label distribution. The label distribution covers a certain number of class labels, representing the degree that each label describes the instance. Through this way, one face image can contribute to not only the learning of its chronological age, but also the learning of its adjacent ages. Two algorithms, named IIS-LLD and CPNN, are proposed to learn from such label distributions. Experimental results on two aging face databases show remarkable advantages of the proposed label distribution learning algorithms over the compared single-label learning algorithms, either specially designed for age estimation or for general purpose.

H. Pan, H. Han, S. Shan and X. Chen, "Mean-Variance Loss for Deep Age Estimation from a Face," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City,UT,USA,2018,pp.5285-5294.

doi:10.1109/CVPR.2018.00554

Abstract: Age estimation has wide applications in video surveillance, social networking, and human-computer interaction. Many of the published approaches simply treat age estimation as an exact age regression problem, and thus do not leverage a distribution's robustness in representing labels with ambiguity such as ages. In this paper, we propose a new loss function, called mean-variance loss, for robust age estimation via distribution learning. Specifically, the mean-variance loss consists of a mean loss, which penalizes difference between the mean of the estimated age distribution and the ground-truth age, and a variance loss, which penalizes the variance of the estimated age distribution to ensure a concentrated distribution. The proposed mean-variance loss and softmax loss are jointly embedded into Convolutional Neural Networks (CNNs) for age estimation. Experimental results on the FG-NET, MORPH Album II, CLAP2016, and AADB

2.3 Proposed System

The project will use Python's Speech Recognition API to read text from audio files. The API will convert the audio file into text, which will then be analyzed by the Support Vector Machine (SVM) algorithm to detect depression. This approach will allow users to share their experiences and feelings through audio recordings, which can be analyzed to identify potential signs of depression.

In addition to audio files, users will be able to upload images via posts. To read text from these images, the project will use Python's Tesseract OCR (Optical Character Recognition) API. Tesseract OCR is an open-source OCR engine that can extract text from images. Once the text has been extracted, the SVM algorithm will analyze it to detect depression.

Users will also be able to upload posts in a text file format. The SVM algorithm will analyze the text to detect depression. This approach will allow users who prefer to communicate through text to share their experiences and feelings.

The SVM algorithm is a supervised machine learning algorithm that has been shown to be effective in text classification tasks. The algorithm will be trained on a large dataset of Reddit posts and comments that have been annotated as depression-related. Once the algorithm has been trained, it will be able to identify potential signs of depression in new posts and comments.

Overall, the proposed implementation will use a combination of Python's Speech Recognition API, Tesseract OCR, and SVM algorithm to detect depression-related posts in the Reddit social media forum. This approach will allow for a variety of input types, including audio files, images, and text files, to be analyzed for potential signs of depression.

2.4Advantages of Proposed System:

- Security
- Communication.
- Reliable in classification of use.
- Easy to segregate and mark user as spam.
- The SVM algorithm segregates the posts in a less time.

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

System Analysis:

The first step in the system analysis would be to identify the requirements of the system. This would involve understanding the needs of the stakeholders, including the users, moderators, and mental health professionals. The requirements would be analyzed and documented to ensure that the system meets the needs of the stakeholders.

Once the requirements have been identified, the system analysis would involve designing a data model that can store the Reddit posts and comments. This data model would be used to store the data that would be used to train the SVM algorithm.

System Design:

The system design would involve designing the architecture of the system. The system would be designed to handle large amounts of data and process it in real-time. The architecture would be designed to be scalable and modular, allowing for easy maintenance and updates.

The system would be designed to incorporate the various APIs that would be used, including Python's Speech Recognition API, Tesseract OCR API, and SVM algorithm. The APIs would be integrated into the system to allow for seamless data processing.

The user interface (UI) design would be an important aspect of the system design. The UI would be designed to be user-friendly and intuitive, allowing users to easily navigate the system and upload their posts. The UI would also include a dashboard that moderators and mental health professionals can use to monitor the system and identify potential cases of depression.

The system design would also involve developing a testing plan to ensure that the system is functioning correctly. The testing plan would involve testing the system on various datasets to ensure that it is accurate and effective in detecting depression-related posts.

Overall, the system analysis and design would involve identifying the requirements of the system, designing a data model, developing a scalable architecture, integrating APIs, designing a user-friendly UI, and developing a testing plan. This would ensure that the system is accurate, effective, and meets the needs of the stakeholders.

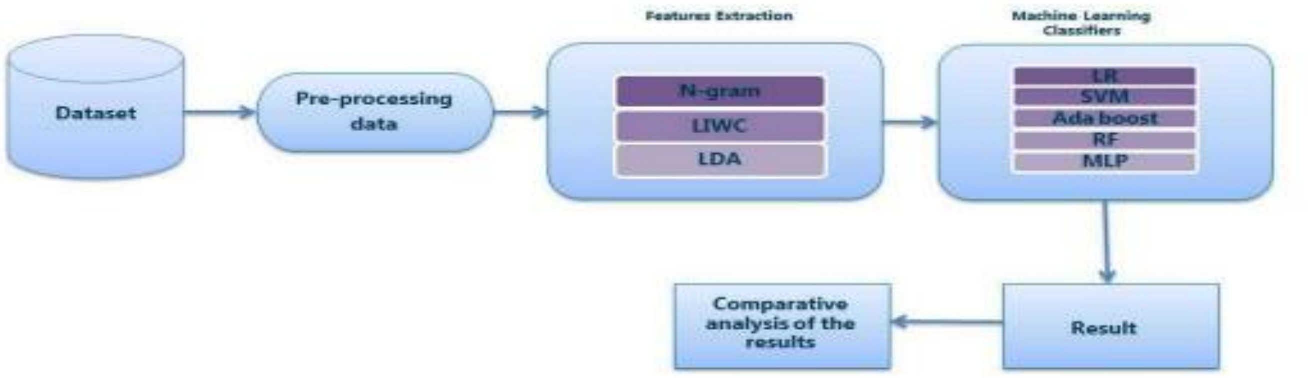


Fig-3.1 Process Flow Diagram

3.1 MODULES

The project working is divided into 2 modules.

3.1.1 Admin Module: Administrator will login to application using username as ‘admin’ and password ‘admin’. After login admin can view all registered users and all posts send by each users. Admin can send motivation messages to all depressed users. All positive and negative depression users can also be seen in the form of graph.

3.1.2 User Module: Users need to register with the application and then login to application to access various sub modules such as :

- **Search Friends:** Using this module user can see all peoples register with the application
- **Upload Posts**: Using this module user can upload post in various formats such as text file, image or audio file. This application accepts only .WAV file format**.**
- **View Motivation Messages:** Using this module users can view all motivation messages send by administrator.

3.2 Data Flow Diagram

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

1. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used bythe process, an external entity that interacts with the system and the information flows in the system.

2. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and thetransformations that are applied as data moves from input to output.

3. DFD is also known as bubble chart. A DFD may be used to represent a system at any levelof abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

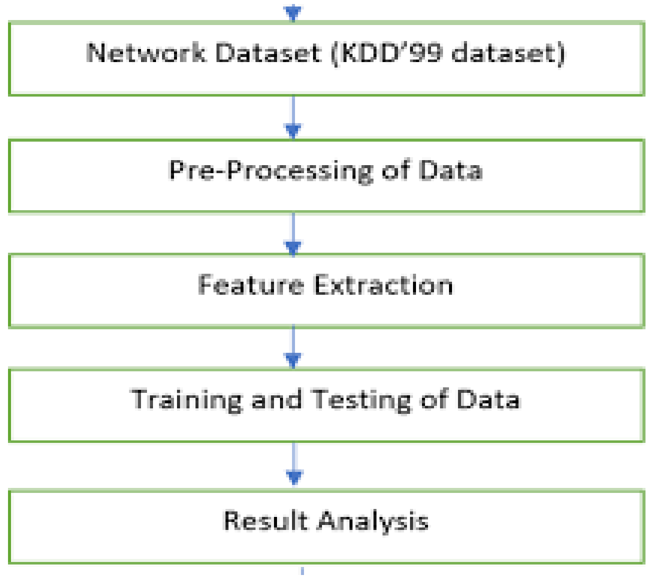

Network Dataset (KDD'99 dataset)
Pre-Processing of Data
Feature Extraction
Training and Testing of Data
Result Analysis

Fig-3.2 Data Flow Diagram of the Project

3.3UML Diagrams

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with UML. The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems. The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

Goals

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modelling Language so that they candevelop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modelling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.

3.3.1 Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to showwhat system functions are performed for which actor. Roles of the actors in the system can be depicted.

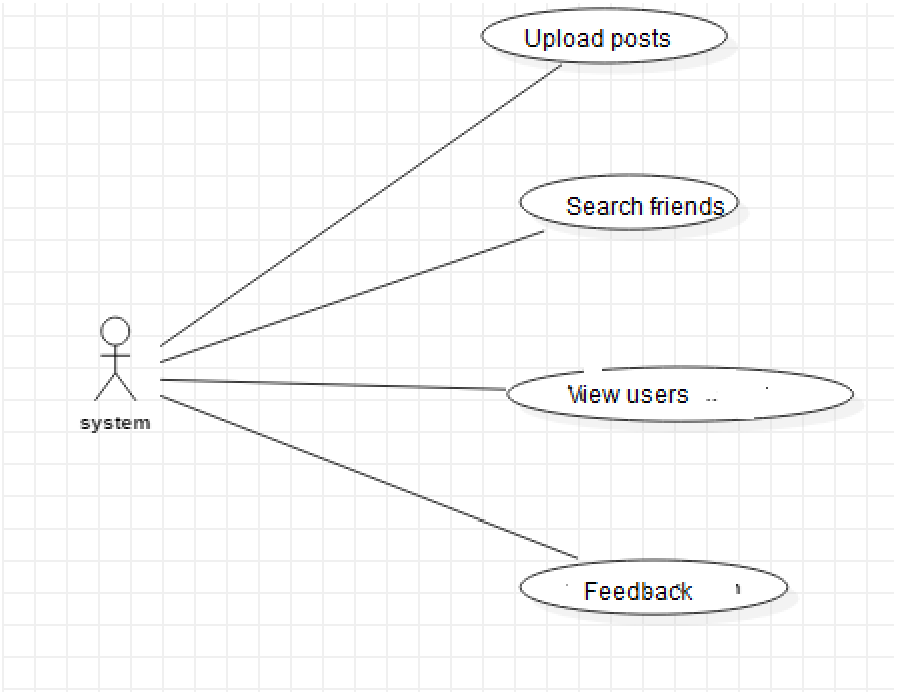


Fig-3.3 Use Case Diagram

3.3.2 Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains whichclass contains information.

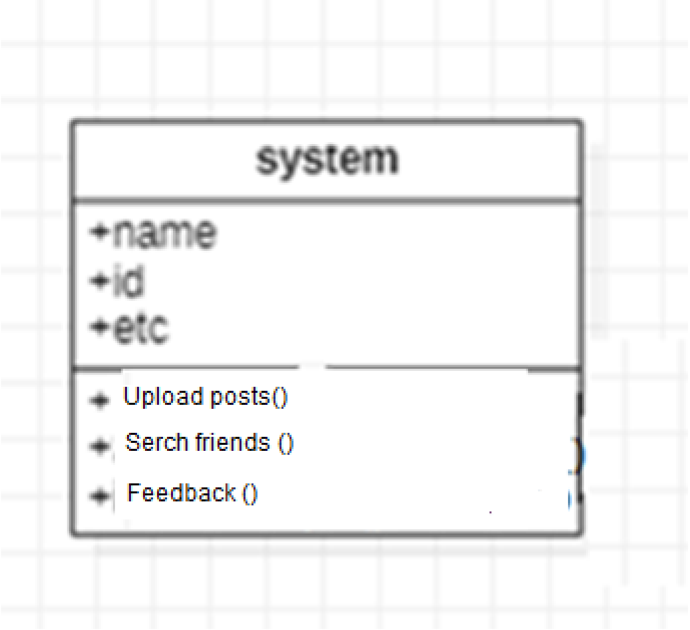


Fig-3.4 Class Diagram

3.3.3 Sequence Diagram

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

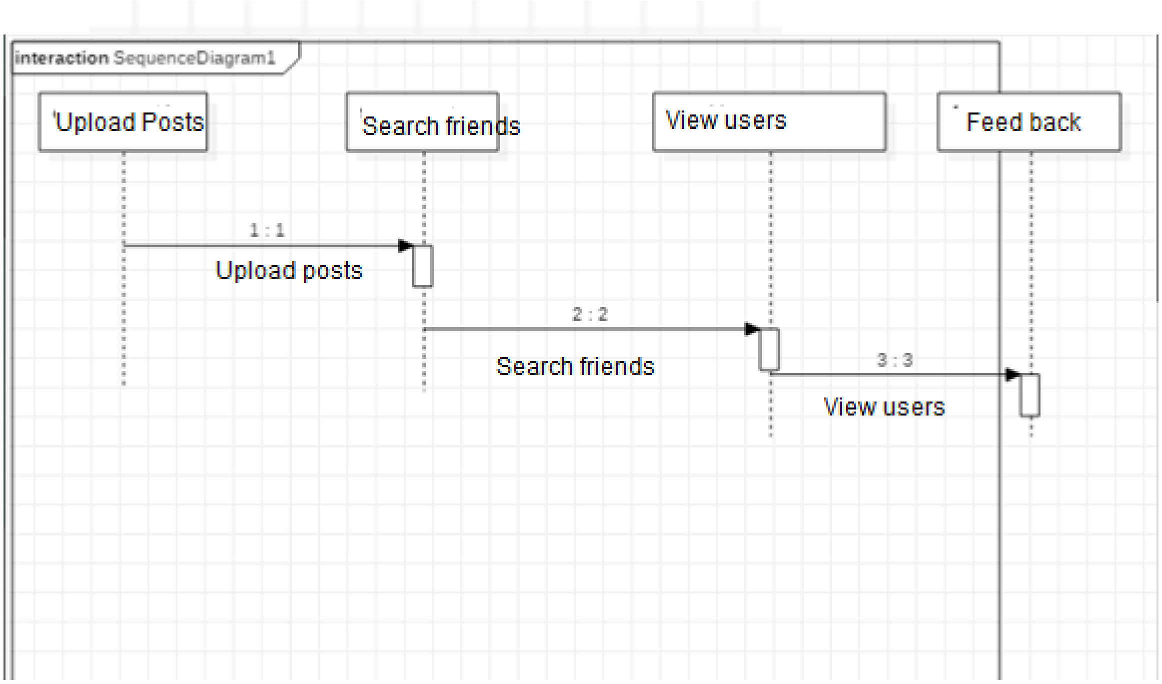


Fig-3.5 Sequence Diagram

3.3.4Activity DIAGRAM

An activity diagram is a visual representation of a process or workflow that shows the different activities or actions involved, as well as the flow between these activities. It is a type of UML diagram that uses nodes, edges, and actions to represent the various components of the process.

Activity diagrams can be used to model complex systems, workflows, or business processes, and are useful for identifying areas of inefficiency or bottlenecks. They provide a clear and concise representation of the process, making it easier for stakeholders to understand and analyze the different steps involved.

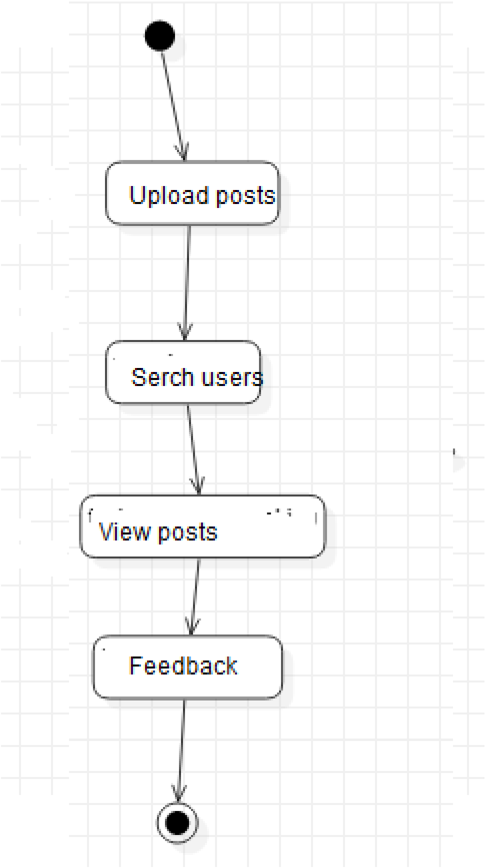


Fig-3.6 Activity DIAGRAM

3.4 DATA DESIGN

3.4.1 Databases SQLite

Name
Online depression

Table 3.4.1 SQLite Database

3.4.2 Tables

Name	Description
Users	Contains all the registered user details.
Upload posts	All the registered service provider details.
Services	Contains all the types of services available.

Table 3.4.2 List of Database Tables

3.5 TECHNOLOGIES USED

3.5.1Python

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly bracketsor keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. C, Python, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. C, Python is managed by the non-profit Python Software Foundation. Python features a dynamic type system and automatic memory management. It supports multipleprogramming paradigms, including object- oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

3.5.2 Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s freeand open source. Django's primary goal is to ease the creation of complex, database-driven websites.Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models.

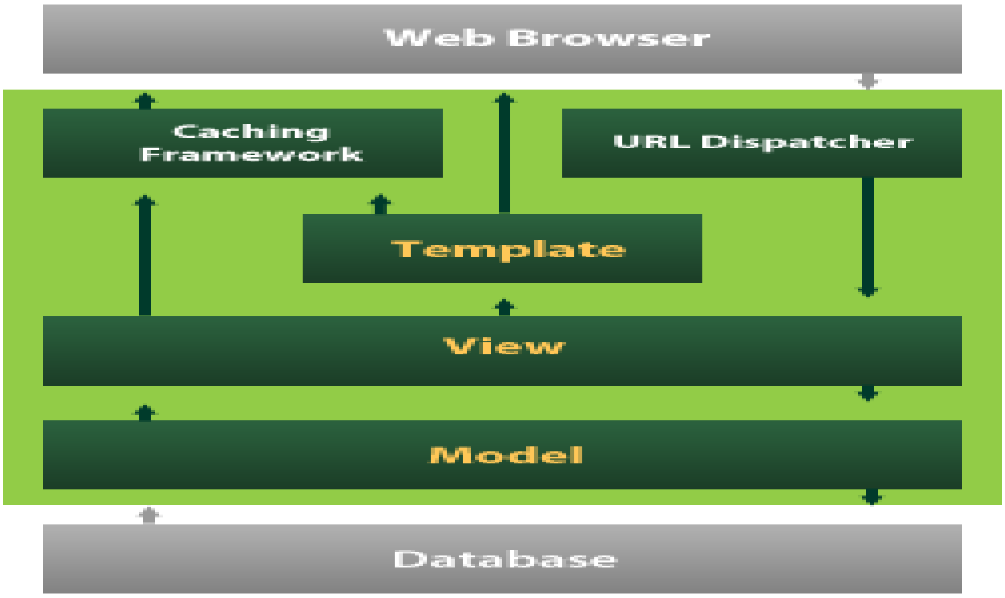


Fig-3.7 Database Diagram

Django also provides an optional administrative create, read, update and delete interface thatis generated dynamically through introspection and configured via admin models.

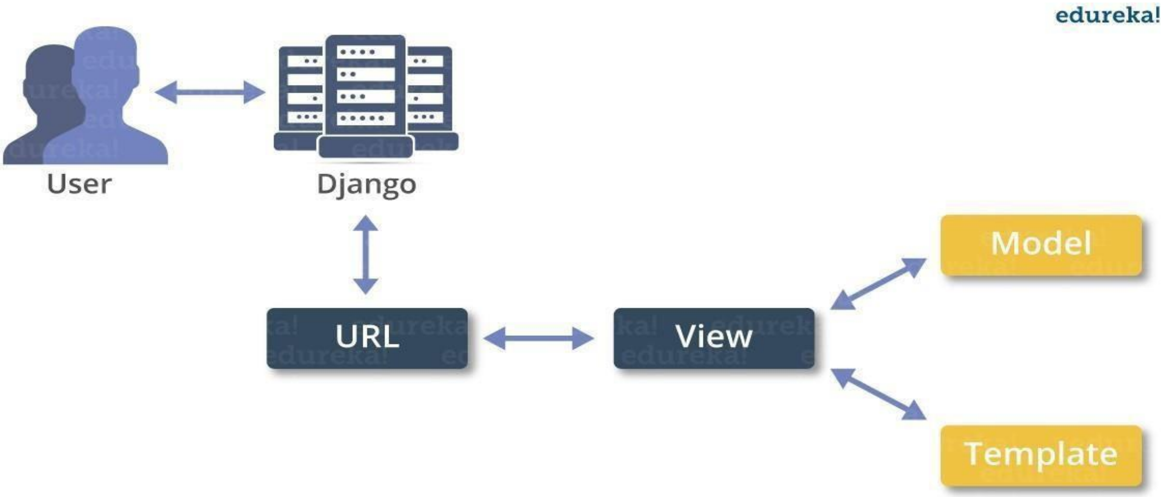


Fig-3.8 Django

CHAPTER 4

IMPLEMENTATION

CHAPTER 4

IMPLEMENTATION

4.1 SYSTEM SPECIFICATIONS:

HARDWARE REQUIREMENTS:

- **Processor** : Intel i5 Processor
- **Hard Disk** : 1 TB
- **Floppy Drive** : 1.44 Mb.
- **Monitor** : SVGA
- **Mouse** : Two or Three Button Mouse
- **Ram** : 6 GB

SOFTWARE REQUIREMENTS:

- **Operating system** : Windows 11
- **Coding Language** : Python
- **Front-End** : Python.
- **Designing** : Python, Django, MYSQL, WampServer 2.4
- **Data Base** : MySQL.

4.2 ALGORITHMS

4.2.1 Support Vector Machine (SVM) Algorithm

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, whichis used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane.

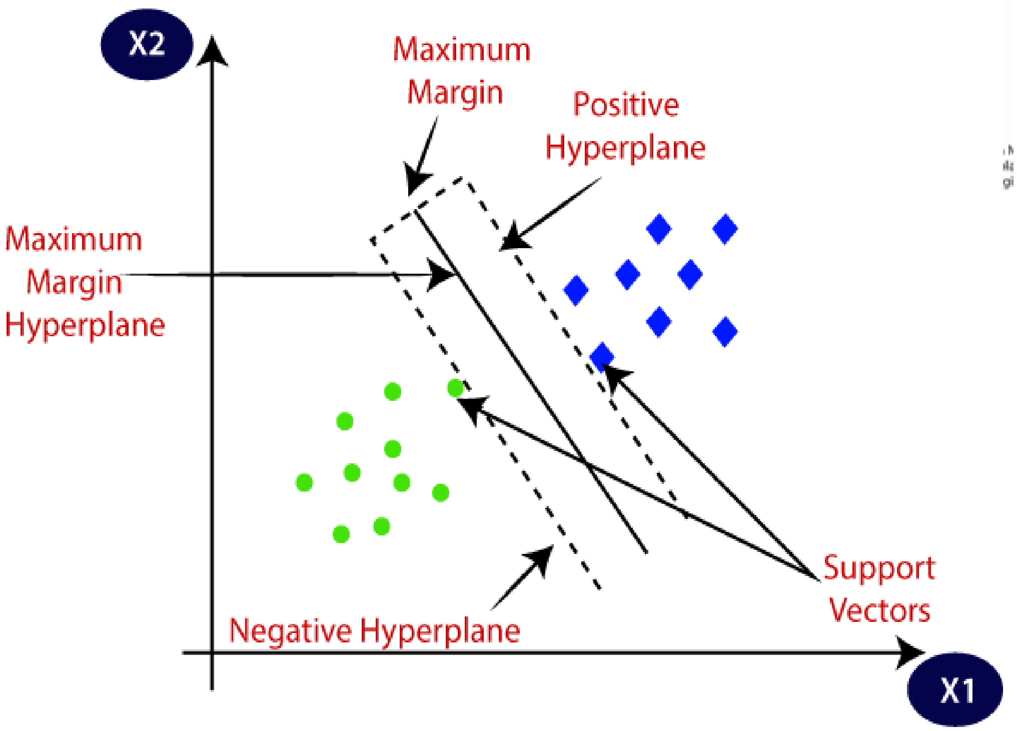


Fig-4.1 Support Vector Machine

Sample code:

```
def runSVM(): text.delete('1.0',
    END)global svm_acc global
    classifier
    global X, Y, X_train, X_test, y_train, y_test total =
    X_train.shape[1];
    X_train1 = SelectKBest(chi2,15).fit_transform(X_train, y_train) X_test1 =
    SelectKBest(chi2,15).fit_transform(X_test,y_test) text.insert(END,"Total
    Features : "+str(total)+"\n")
    text.insert(END,"Features set reduce after applying features selection concept : "+str((total -
X_train.shape[1]))+"\n\n")
    cls = svm.SVC(kernel='rbf', class_weight='balanced', probability=True)cls.fit(X_train,
    y_train)
    text.insert(END,"Prediction Results\n\n")
    prediction_data = prediction(X_test, cls)
    svm_acc = cal_accuracy(y_test, prediction_data,'SVM Accuracy, ClassificationReport & Confusion
Matrix')
    classifier = cls
```

4.2.2 Multilayer Perceptron (MLP):

Multilayer Perceptron (MLP) is a type of artificial neural network that is commonly used in machine learning for classification and regression tasks. MLPs consist of multiple layers of neurons, including an input layer, one or more hidden layers, and an output layer. Each neuron in a layer is connected to all neurons in the next layer, and the network is trained using backpropagation to adjust the weights and biases of the connections between the neurons.

In an MLP, each neuron in the hidden layers applies a non-linear transformation to the weighted sum of the outputs from the previous layer. This allows the network to learn non-linear relationships between the inputs and outputs. The output layer of an MLP typically consists of one or more neurons that produce the predicted output values.

MLPs can be used for a wide range of classification and regression tasks, including image classification, speech recognition, natural language processing, and financial forecasting. They are particularly useful when dealing with complex, high-dimensional data, such as images or text. MLPs have been shown to perform well in many benchmark machine learning tasks, but they can be computationally expensive to train and may require large amounts of labeled training data.

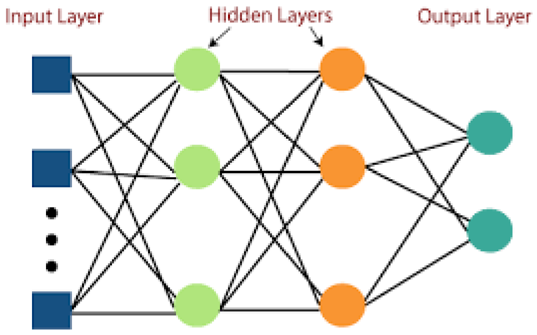


Fig 4.2 Multilayer Perceptron

Sample code

```
from keras.models import Sequential
 from keras.layers import Dense

 from sklearn.datasets import make_classification

 from sklearn.model_selection import train_test_split


 # Generate a random binary classification dataset

 X, y = make_classification(n_samples=1000, n_features=10, n_classes=2, random_state=42)


 # Split the dataset into training and testing sets

 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


 # Initialize the MLP using the Sequential API

 model = Sequential()


 # Add an input layer with 10 neurons and a hidden layer with 5 neurons

 model.add(Dense(5, input_dim=10, activation='relu'))


 # Add an output layer with 1 neuron and a sigmoid activation function

 model.add(Dense(1, activation='sigmoid'))


 # Compile the model with binary cross-entropy loss and the Adam optimizer

 model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])


 # Train the model on the training set
```

```
        model.fit(X_train, y_train, epochs=100, batch_size=32, verbose=1)

        # Evaluate the model on the testing set

        score = model.evaluate(X_test, y_test, verbose=0)


        # Print the accuracy score

print('Accuracy:', score[1])
```

4.3 Implementation

4.3.1 Sample code:

```
import pymysql
print(pymysql._file_)
pymysql.install_as_MySQLdb()

from django.apps import AppConfig



class DepressionappConfig(AppConfig):
    name = 'DepressionApp'

from django.db import models

# Create your models here.

from django.test import TestCase

# Create your tests here.

from django.urls import path

from . import views

urlpatterns = [path("index.html", views.index, name="index"),
               path("Register.html", views.Register, name="Register"),
               path("Signup", views.Signup, name="Signup"),
               path("Login.html", views.Login, name="Login"),
               path("UserLogin", views.UserLogin, name="UserLogin"),
               path("SearchFriends.html", views.SearchFriends, name="SearchFriends"),
               path("UploadPost.html", views.UploadPost, name="UploadPost"),
               path("UploadPostData", views.UploadPostData, name="UploadPostData"),
               path("AdminLogin", views.AdminLogin, name="AdminLogin"),
path("Admin.html", views.Admin, name="Admin"),
              path("ViewUsers.html", views.ViewUsers, name="ViewUsers"),
              path("ViewPosts.html", views.ViewPosts, name="ViewPosts"),
              path("SendMotivatedPost.html", views.SendMotivatedPost,
name="SendMotivatedPost"),
              path("SendMotivatedPostData", views.SendMotivatedPostData,
name="SendMotivatedPostData"),
              path("ViewMotivatedPost.html", views.ViewMotivatedPost,
name="ViewMotivatedPost"),
              path("MotivatedText.html", views.MotivatedText, name="MotivatedText"),
           ]

from django.shortcuts import render
from django.template import RequestContext
from django.contrib import messages
import pymysql
from django.http import HttpResponse
from django.conf import settings
from django.core.files.storage import FileSystemStorage
import datetime
from sklearn.externals import joblib
import PIL.Image
import pytesseract
import matplotlib.pyplot as plt
import re
import numpy as np
import speech_recognition as sr
# Create your views here.

svm_classifier = joblib.load('svmClassifier.pkl')

def index(request):
    if request.method == 'GET':
       return render(request, 'index.html', {})

def UploadPost(request):
    if request.method == 'GET':
        return render(request, 'UploadPost.html', {})

def Register(request):
    if request.method == 'GET':
        return render(request, 'Register.html', {})

def Admin(request):
    if request.method == 'GET':
        return render(request, 'Admin.html', {})

def Login(request):
    if request.method == 'GET':
        return render(request, 'Login.html', {})

def SendMotivatedPost(request):
    if request.method == 'GET':
        return render(request, 'SendMotivatedPost.html', {})



def predict(textdata,classifier):
   text_processed = textdata
   X =  [text_processed]
   sentiment = classifier.predict(X)
   return (sentiment[0])



def predictSentiment(textdata):
    result = predict(textdata, svm_classifier)
    predicts = ""
    if result == 0:
      predicts = "Negative"
    if result == 1:
      predicts = "Positive"
    return predicts

def SendMotivatedPostData(request):
     if request.method == 'POST':
        username = request.POST.get('t1', False)
        time = request.POST.get('t2', False)
        text = request.POST.get('t3', False)
        db_connection = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password = 'root',
database = 'depression',charset='utf8')
        db_cursor = db_connection.cursor()
        student_sql_query = "update postdata set motivate_post='"+text+"' where
username='"+username+"' and post_time='"+time+"' and motivate_post='Pending'"
        db_cursor.execute(student_sql_query)
        db_connection.commit()
        print(db_cursor.rowcount, "Record Inserted")
        context= {'data':'Your motivated text sent to user '+username}
        return render(request, 'SendMotivatedPost.html', context)

def UploadPostData(request):
    if request.method == 'POST' and request.FILES['t1']:
        output = ''
        myfile = request.FILES['t1']
        fs = FileSystemStorage()
        name = str(myfile)
        if name.lower().endswith(('.txt')):
            name = 'text.txt'
        elif name.lower().endswith(('.png', '.jpg', '.jpeg', 'gif')):
            name = 'img.jpg'
        filename = fs.save(name, myfile)
        if name.lower().endswith(('.txt')):
            with open("text.txt", "r") as file:
                for line in file:
                    line = line.strip('\n')
                    output+=line+' '
        elif name.lower().endswith(('.png', '.jpg', '.jpeg', 'gif')):
            output = pytesseract.image_to_string(PIL.Image.open(name))
            output = output.replace('\n',' ')
        elif name.lower().endswith(('.wav')):
            r = sr.Recognizer()
            with sr.WavFile(name) as source:
                audio = r.record(source)
        try:
            output = r.recognize_google(audio)
        except:
            pass
        user = ''
        with open("session.txt", "r") as file:
          for line in file:
              user = line.strip('\n')
        now = datetime.datetime.now()
        option = 'Pending'
        output = re.sub('\W+',' ', output)
        current_time = now.strftime("%Y-%m-%d %H:%M:%S")
        sentiment = predictSentiment(output.lower())
        db_connection = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password = 'root',
database = 'depression',charset='utf8')
        db_cursor = db_connection.cursor()
        student_sql_query = "INSERT INTO
postdata(username,post_data,post_time,depression,motivate_post)
VALUES('"+user+"','"+output+"','"+current_time+"','"+sentiment+"','"+option+"')"
        db_cursor.execute(student_sql_query)
        db_connection.commit()
        print(db_cursor.rowcount, "Record Inserted")
        if db_cursor.rowcount == 1:
            context= {'data':'Detected Depression From Uploaded File : '+sentiment}
            return render(request, 'UploadPost.html', context)
        else:
            context= {'data':'Error in signup process'}
            return render(request, 'UploadPost.html', context)



def ViewUsers(request):
    if request.method == 'GET':
        strdata = '<table border=1 align=center
width=100%><tr><th>Username</th><th>Password</th><th>Contact No</th><th>Email
ID</th><th>Address</th></tr><tr>'
        con = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password = 'root', database =
'depression',charset='utf8')
        with con:
            cur = con.cursor()
            cur.execute("select * FROM users")
            rows = cur.fetchall()
            for row in rows:

strdata+='<td>'+row[0]+'</td><td>'+row[1]+'</td><td>'+row[2]+'</td><td>'+str(row[3])+'</td><td
>'+str(row[4])+'</td></tr>'
    context= {'data':strdata}
    return render(request, 'ViewUsers.html', context)


def ViewPosts(request):
    if request.method == 'GET':
        positive = 0
        negative = 0
        strdata = '<table border=1 align=center width=100%><tr><th>Username</th><th>Post
Data</th><th>Post Time</th><th>Depression</th><th>Motivated Post</th></tr><tr>'
        con = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password = 'root', database =
'depression',charset='utf8')
        with con:
            cur = con.cursor()
            cur.execute("select * FROM postdata")
            rows = cur.fetchall()
            for row in rows:
                if row[3] == 'Negative':
                    negative = negative + 1
                else:
                    positive = positive + 1

strdata+='<td>'+row[0]+'</td><td>'+row[1]+'</td><td>'+str(row[2])+'</td><td>'+str(row[3])+'</td
><td>'+str(row[4])+'</td></tr>'
    height = [positive,negative]
    bars = ('Depression Posts', 'Non Depression Post')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, height)
    plt.xticks(y_pos, bars)
    plt.show()
    context= {'data':strdata}
    return render(request, 'ViewPosts.html', context)
```

```
def MotivatedText(request):
    if request.method == 'GET':
        user = ''
        with open("session.txt", "r") as file:
            for line in file:
                user = line.strip('\n')
        strdata = '<table border=1 align=center width=100%><tr><th>Username</th><th>Post
Data</th><th>Post Time</th><th>Depression</th><th>Motivated Post</th></tr><tr>'
        con = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password = 'root', database =
'depression',charset='utf8')
        with con:
            cur = con.cursor()
            cur.execute("select * FROM postdata")
            rows = cur.fetchall()
            for row in rows:
                if row[0] == user:

strdata+='<td>'+row[0]+'</td><td>'+row[1]+'</td><td>'+str(row[2])+'</td><td>'+str(row[3])+'</td
><td>'+str(row[4])+'</td></tr>'
    context= {'data':strdata}
    return render(request, 'MotivatedText.html', context)

def ViewMotivatedPost(request):
    if request.method == 'GET':
        strdata = '<table border=1 align=center width=100%><tr><th>Username</th><th>Post
Data</th><th>Post Time</th><th>Depression</th><th>Motivated Post</th></tr><tr>'
        con = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password = 'root', database =
'depression',charset='utf8')
        with con:
            cur = con.cursor()
            cur.execute("select * FROM postdata")
            rows = cur.fetchall()
            for row in rows:
                if row[4] != 'Pending':

strdata+='<td>'+row[0]+'</td><td>'+row[1]+'</td><td>'+str(row[2])+'</td><td>'+str(row[3])+'</td
><td>'+str(row[4])+'</td></tr>'
    context= {'data':strdata}
    return render(request, 'ViewMotivatedPost.html', context)

def SearchFriends(request):
    if request.method == 'GET':
        user = ''
        with open("session.txt", "r") as file:
            for line in file:
                user = line.strip('\n')
        strdata = '<table border=1 align=center width=100%><tr><th>Username</th><th>Contact
No</th><th>Email ID</th><th>Address</th></tr><tr>'
        con = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password = 'root', database =
'depression',charset='utf8')
        with con:
            cur = con.cursor()
            cur.execute("select * FROM users")
            rows = cur.fetchall()
            for row in rows:
                if row[0] != user:

strdata+='<td>'+row[0]+'</td><td>'+row[2]+'</td><td>'+row[3]+'</td><td>'+str(row[4])+'</td></tr
>'
    context= {'data':strdata}
    return render(request, 'SearchFriends.html', context)



def UserLogin(request):
    if request.method == 'POST':
      username = request.POST.get('t1', False)
      password = request.POST.get('t2', False)
      index = 0
      con = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password = 'root', database =
'depression',charset='utf8')
      with con:
          cur = con.cursor()
          cur.execute("select * FROM users")
          rows = cur.fetchall()
          for row in rows:
             if row[0] == username and password == row[1]:
                index = 1
                break
        if index == 1:
         file = open('session.txt','w')
         file.write(username)
         file.close()
         context= {'data':'welcome '+username}
         return render(request, 'UserScreen.html', context)
        else:
         context= {'data':'login failed'}
         return render(request, 'Login.html', context)

def Signup(request):
    if request.method == 'POST':
        username = request.POST.get('t1', False)
        password = request.POST.get('t2', False)
        contact = request.POST.get('t3', False)
        email = request.POST.get('t4', False)
        address = request.POST.get('t5', False)
        db_connection = pymysql.connect(host='127.0.0.1',port = 3308,user = 'root', password = 'root', 
database = 'depression',charset='utf8')
        db_cursor = db_connection.cursor()
        student_sql_query = "INSERT INTO users(username,password,contact_no,email,address) 
VALUES('"+username+"','"+password+"','"+contact+"','"+email+"','"+address+"')"
        db_cursor.execute(student_sql_query)
        db_connection.commit()
        print(db_cursor.rowcount, "Record Inserted")
        if db_cursor.rowcount == 1:
         context= {'data':'Signup Process Completed'}
         return render(request, 'Register.html', context)
        else:
         context= {'data':'Error in signup process'}
         return render(request, 'Register.html', context)



def AdminLogin(request):
    if request.method == 'POST':
            username = request.POST.get('t1', False)
            password = request.POST.get('t2', False)
            if username == 'admin' and password == 'admin':
             context= {'data':'welcome '+username}
             return render(request, 'AdminScreen.html', context)
            else:
             context= {'data':'login failed'}
return render(request, 'Admin.html', context)
```

CHAPTER 5

EXPERIMENTAL RESULTS

CHAPTER 5

EXPERIMENTAL RESULTS

5.1 Experimental Results

The Android application is developed using a combination of XML, Java, and SQL with Firebase connectivity. The application aims to provide household services to individuals in need. The application was designed to provide the following functionalities:

Home screen with different fragments: The application displays a home screen with different fragments that provide various functionalities.

User authentication: Users can authenticate themselves by using the login screen that uses Firebase.

Home screen based on user or service provider: After successful login, the home screen is displayed based on whether the user is a service provider or a user seeking services.

Booking services: Users can select the service they need and book a slot with a particular service provider from the displayed list.

User management: Users can add, update, view, and delete their details.

Service provider management: Service providers can view all the bookings that are made by the users and attend to them one by one. They can also set their preferences to not available if they are too busy or if many users have already booked them.

Location preferences: Service providers have the ability to change their radius of location for servicing. They can set up to a 10 km radius.

Logout and end the session: The users and service providers can log out and end their session.

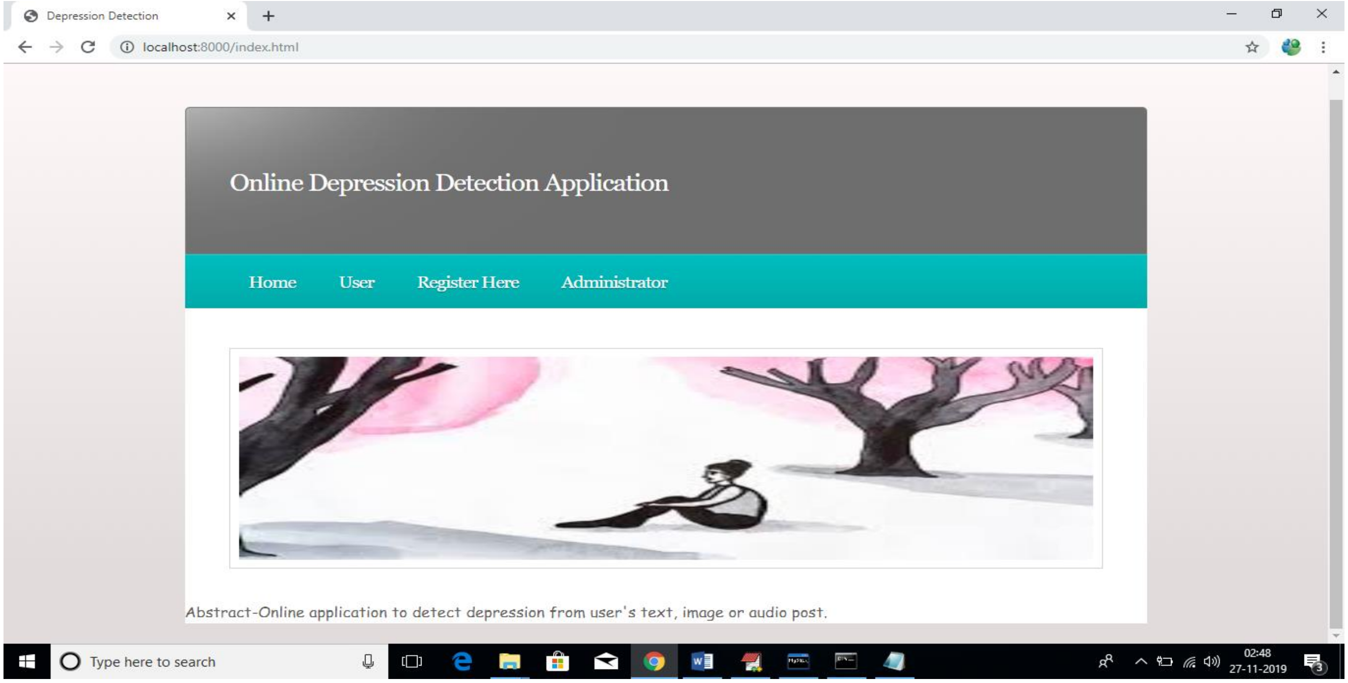
The application has been designed to provide an easy-to-use interface for both users and service providers. The use of Firebase for user authentication provides a secure and reliable login mechanism. The use of SQL provides efficient and effective data management. The application is capable of providing a reliable and efficient platform for household services, making it convenient for users to book services and for service providers to manage their bookings and communicate with service providers.

Other features of the application include chat functionality, multi-language support, analytics, personalization, and social media integration. The application provides a personalized experience for users based on their past bookings, preferences, and ratings. The use of advanced features, such as push notifications, chat functionality, and payment integration, makes the app user-friendly and provides a seamless experience.

5.2 Screenshots

User Login Screen:

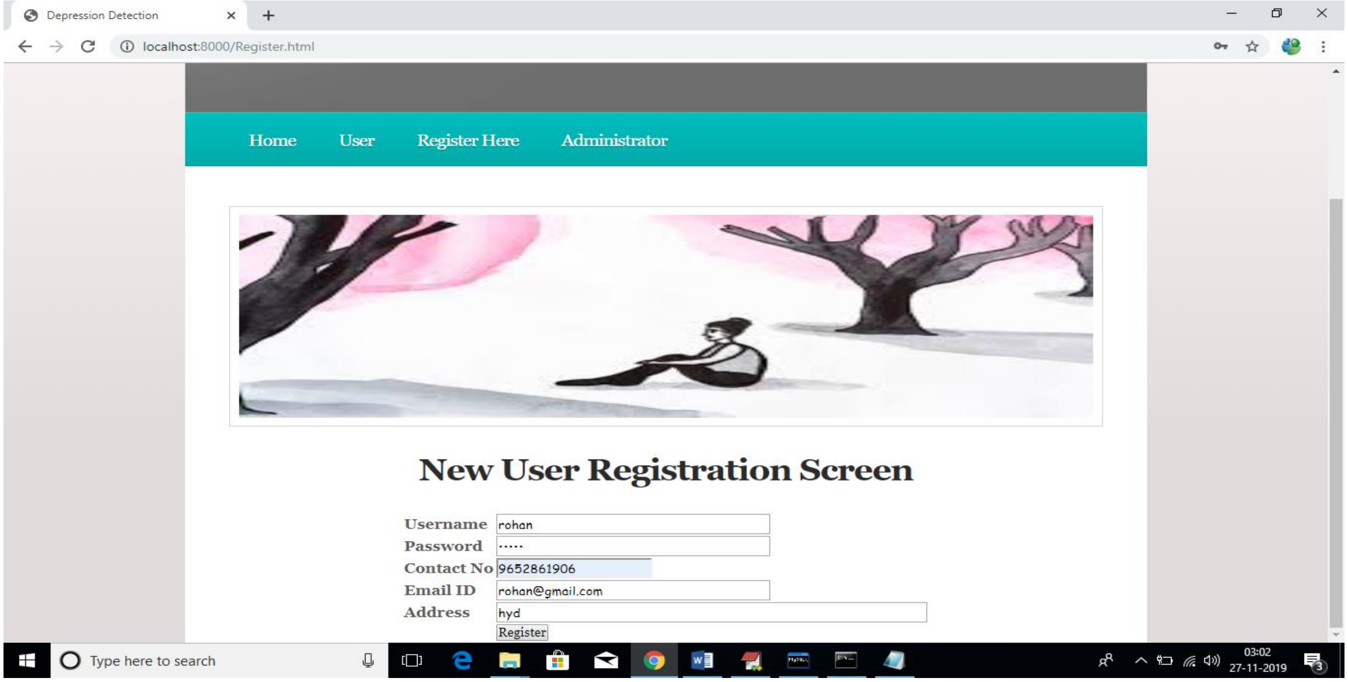
Now deploy code in DJANGO and start server and run in browser to get below screen.



Online Depression Detection Application

Abstract-Online application to detect depression from user's text, image or audio post.

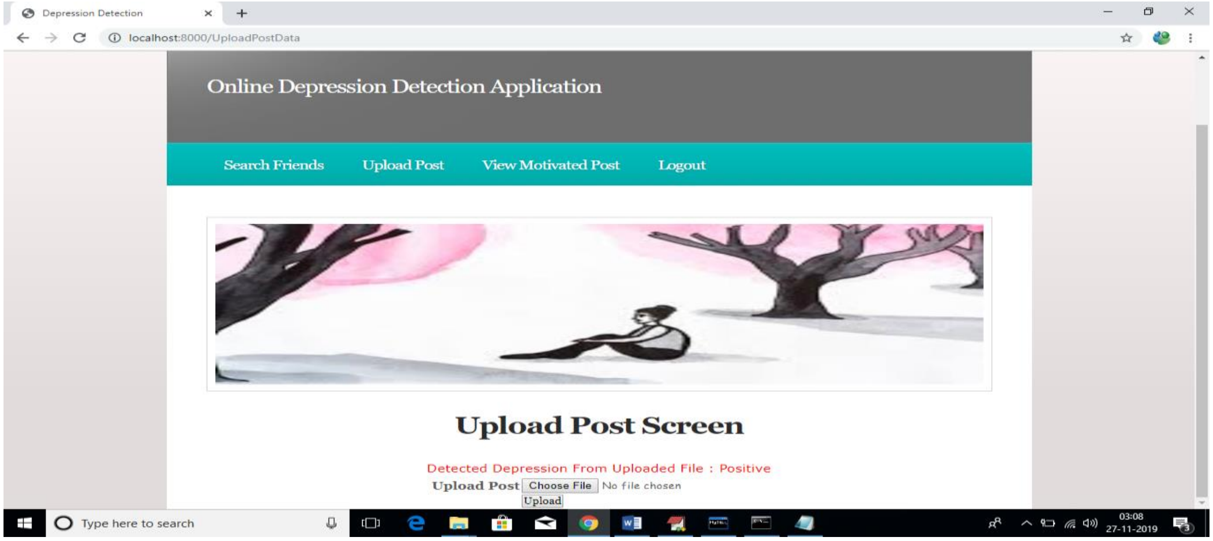
In above screen click on 'Register Here' link to add new user to application



New User Registration Screen

Username rohan
Password ·····
Contact No 9652861906
Email ID rohan@gmail.com
Address hyd
Register

In above screen I am uploading one recording file, after upload will get below message from recording data



Online Depression Detection Application

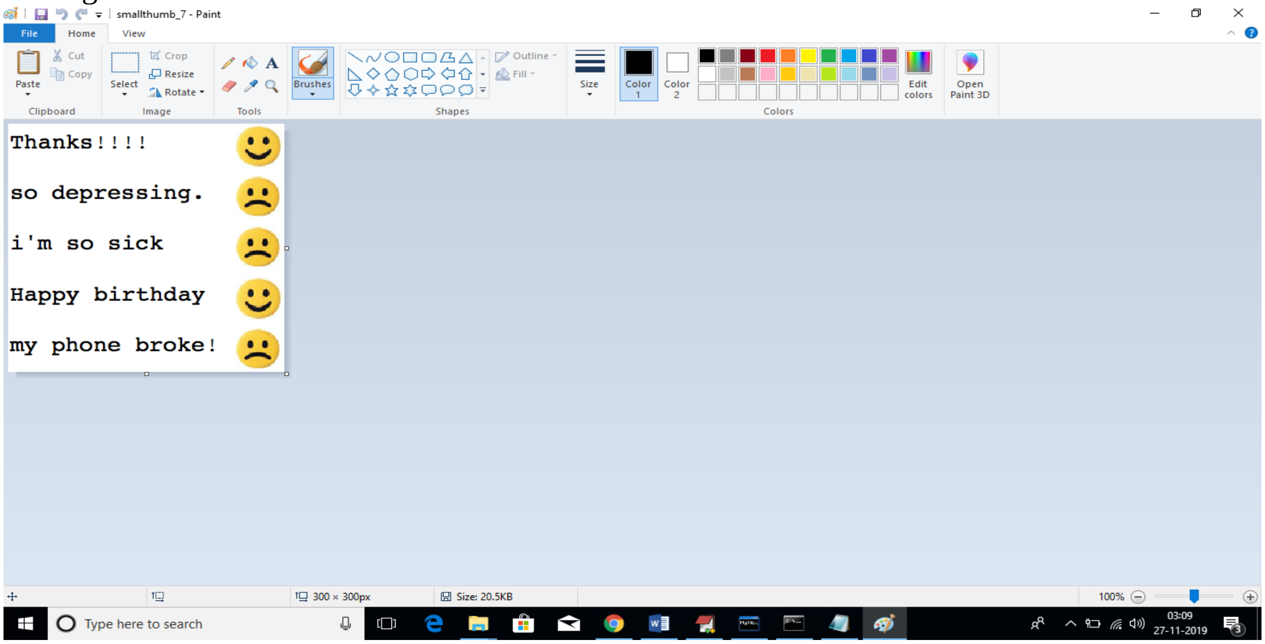
Upload Post Screen

Detected Depression From Uploaded File : Positive

Upload Post Choose File No file chosen

Upload

In above screen its says uploaded file contains message which indicate user is happy and gave positive recording. In below screen I am uploading one image which contains depress negative message



Thanks!!!!

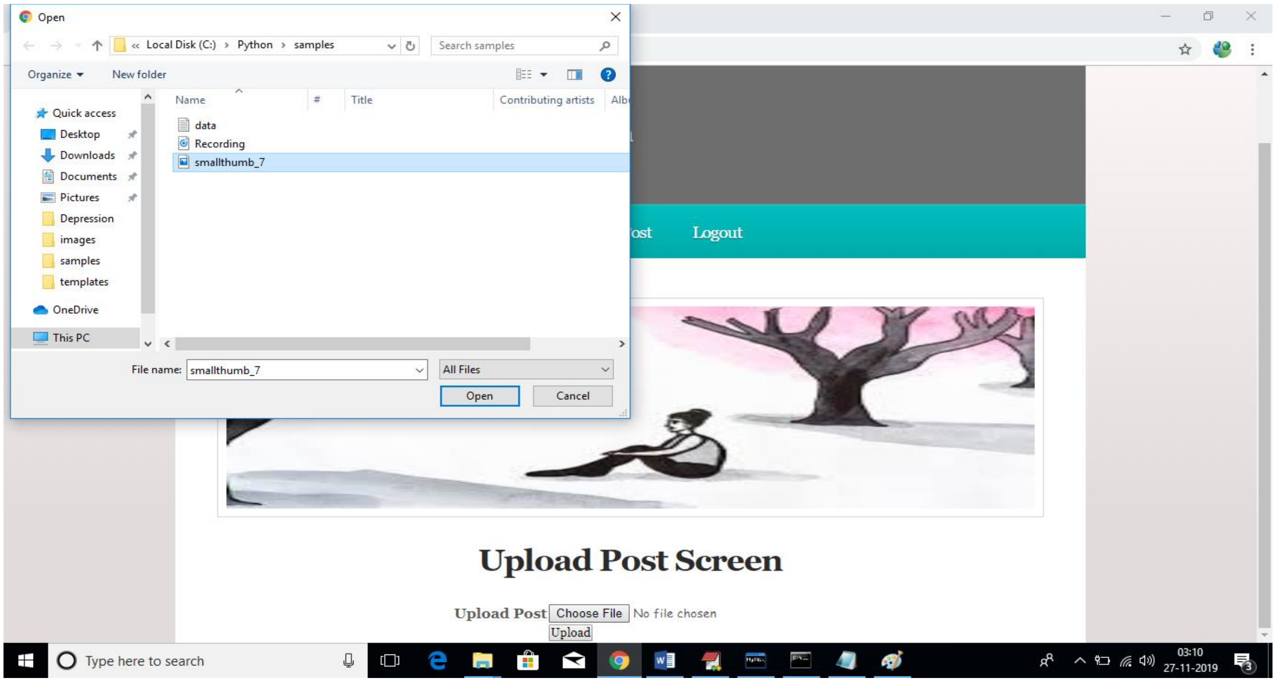
so depressing.

i'm so sick

Happy birthday

my phone broke!

In the above screen in title image file name is ‘smallthumb-7’ and it contains depress negative message and I will upload same image to application and see results

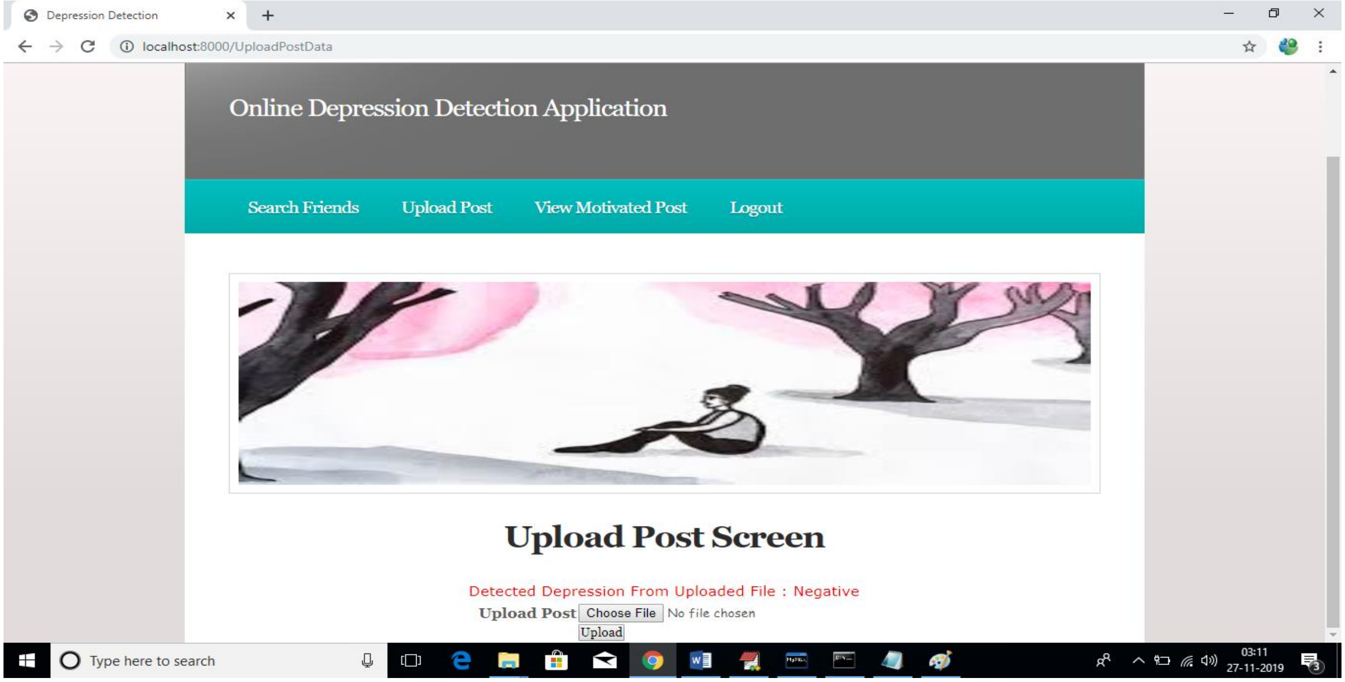


File name: smallthumb_7

Upload Post Screen

Upload Post No file chosen

In above screen we can see I am uploading same image and below are is the output screen.



localhost:8000/UploadPostData

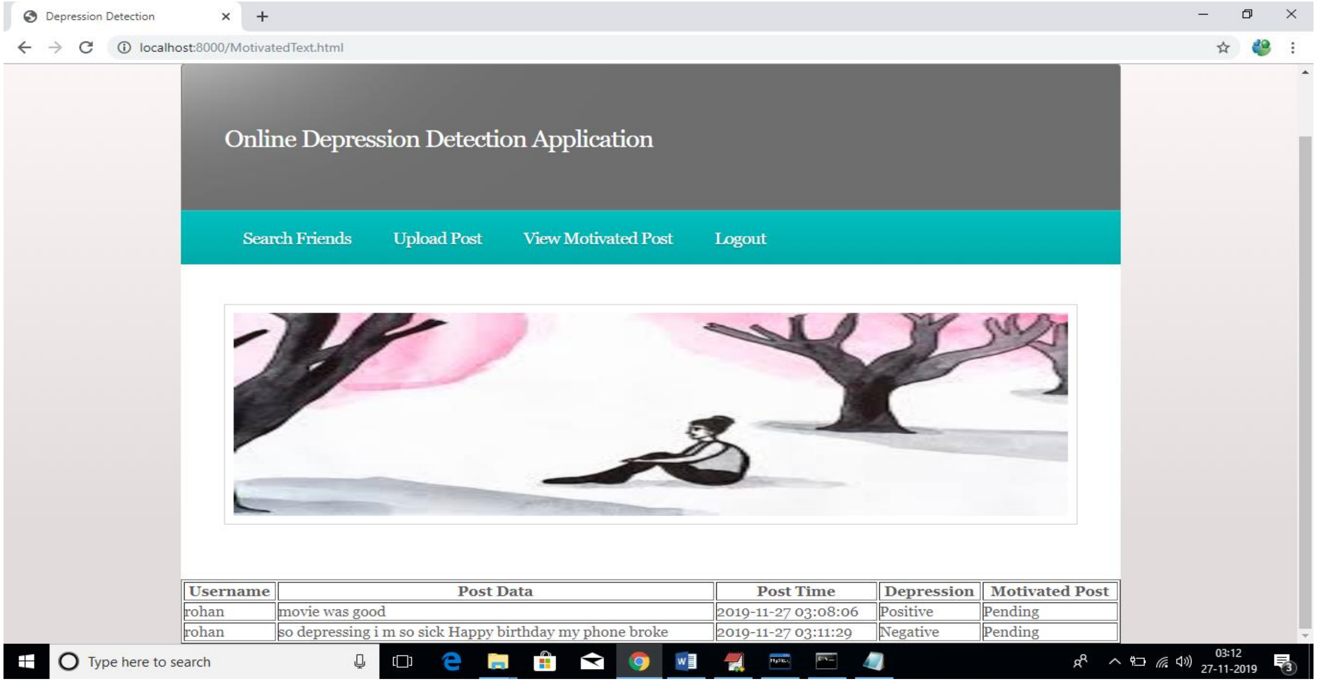
Online Depression Detection Application

Upload Post Screen

Detected Depression From Uploaded File : Negative

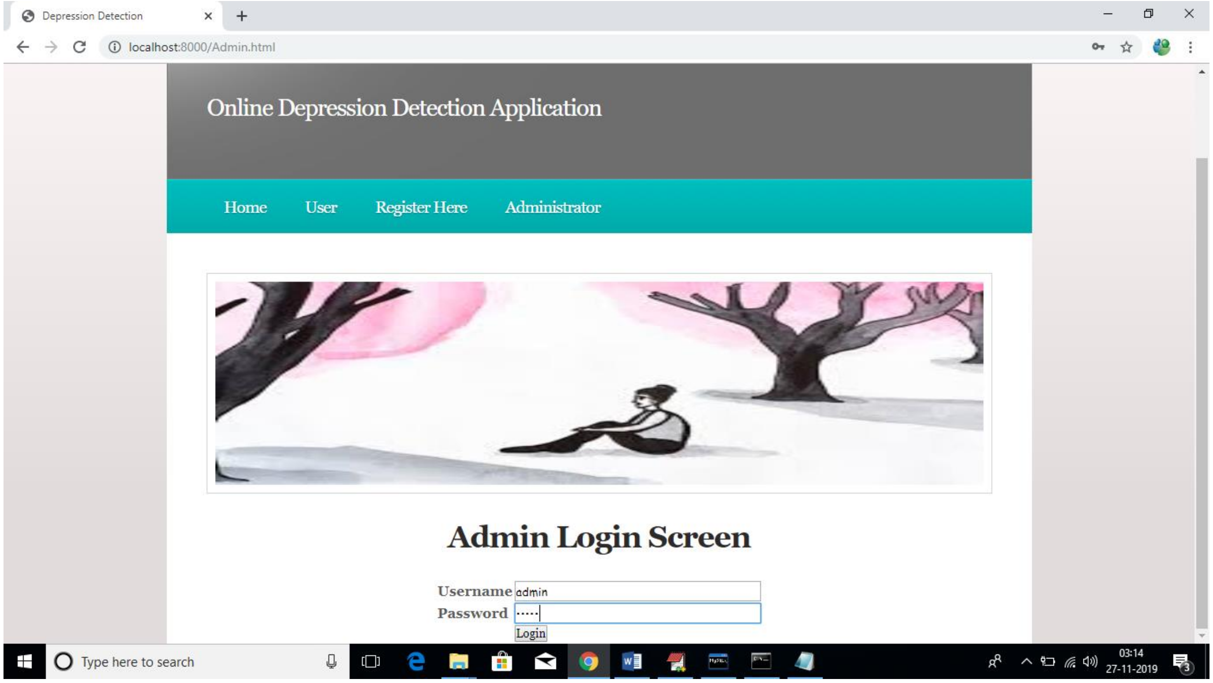
Upload Post No file chosen

In above screen message is negative and we can say user is depressed Similarly click on 'View Motivated Post' link to get admin message if he posted otherwise message will mark as 'Pending'



Username	Post Data	Post Time	Depression	Motivated Post
rohan	movie was good	2019-11-27 03:08:06	Positive	Pending
rohan	so depressing i m so sick Happy birthday my phone broke	2019-11-27 03:11:29	Negative	Pending

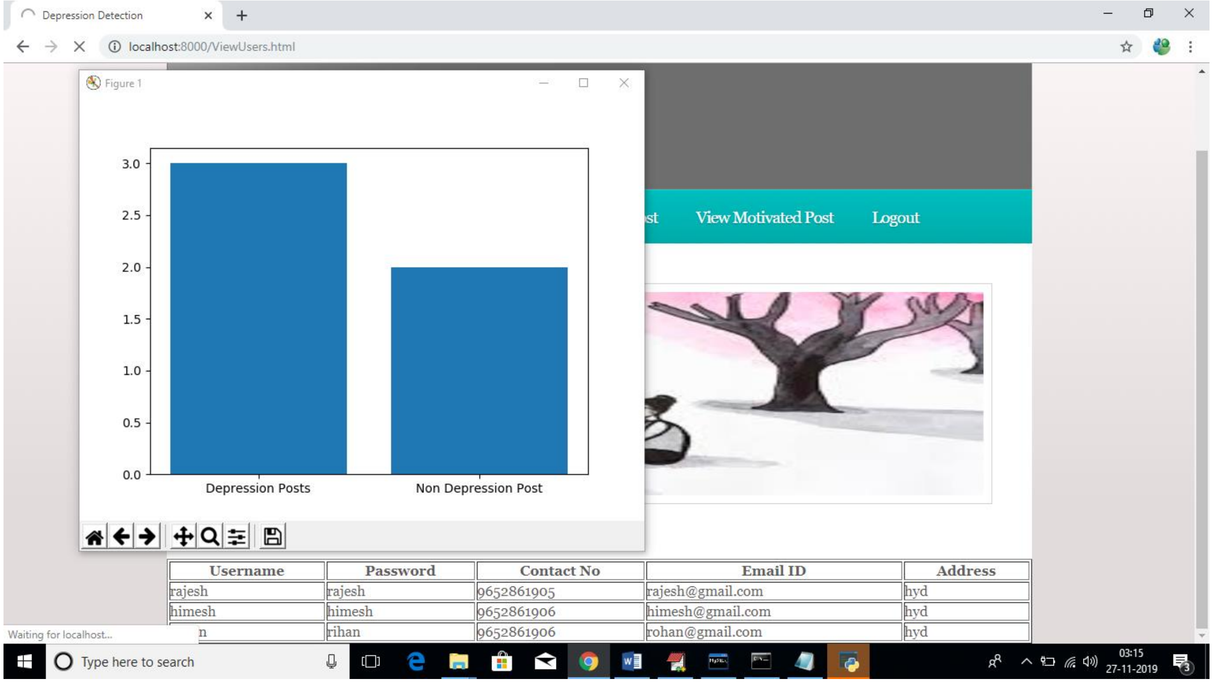
In above screen we can see this user uploaded two files one is recording and other one is image and both file data we can see as post data column and depression result as positive or negative also we can see. Admin has not sent any motivated post so the field is marked as ‘Pending’. Now logout andloginas‘admin’.



Admin Login Screen

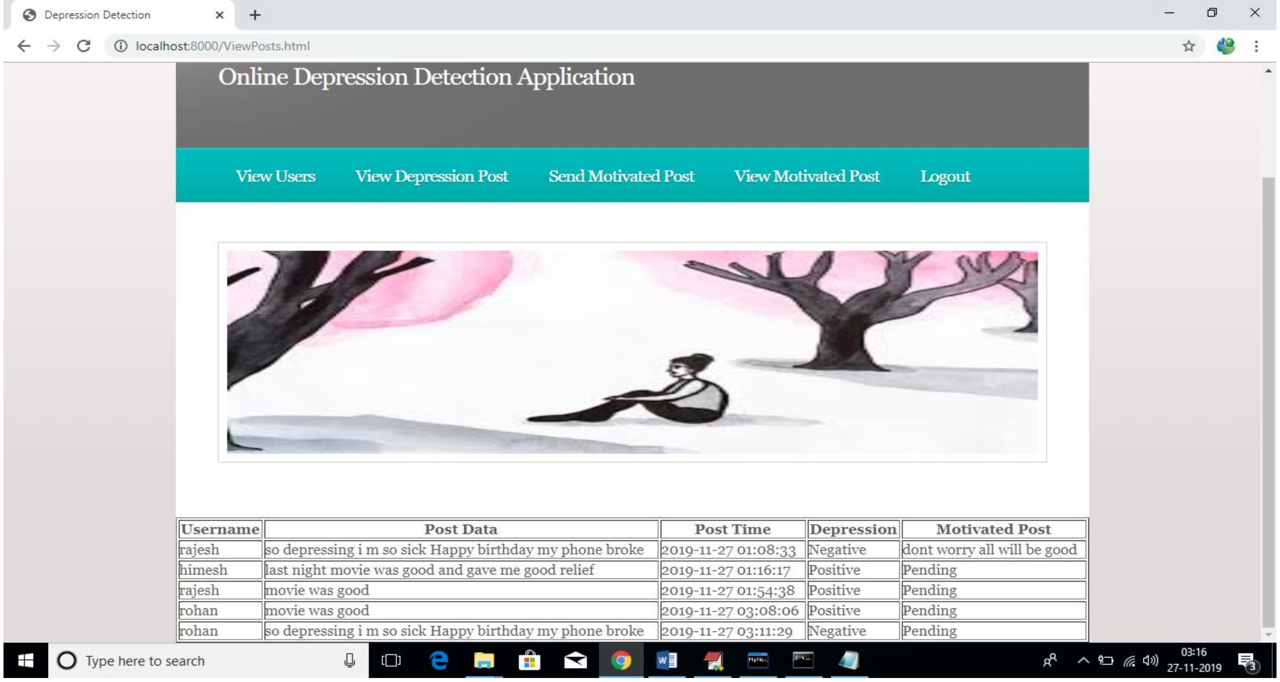
Username admin
Password ·····
Login

After login we can get below admin screen. In above screen admin can click on ‘View Users’ to view all users list. Similarly admin can click on ‘View Depression Post’ to view all post in graph and text.



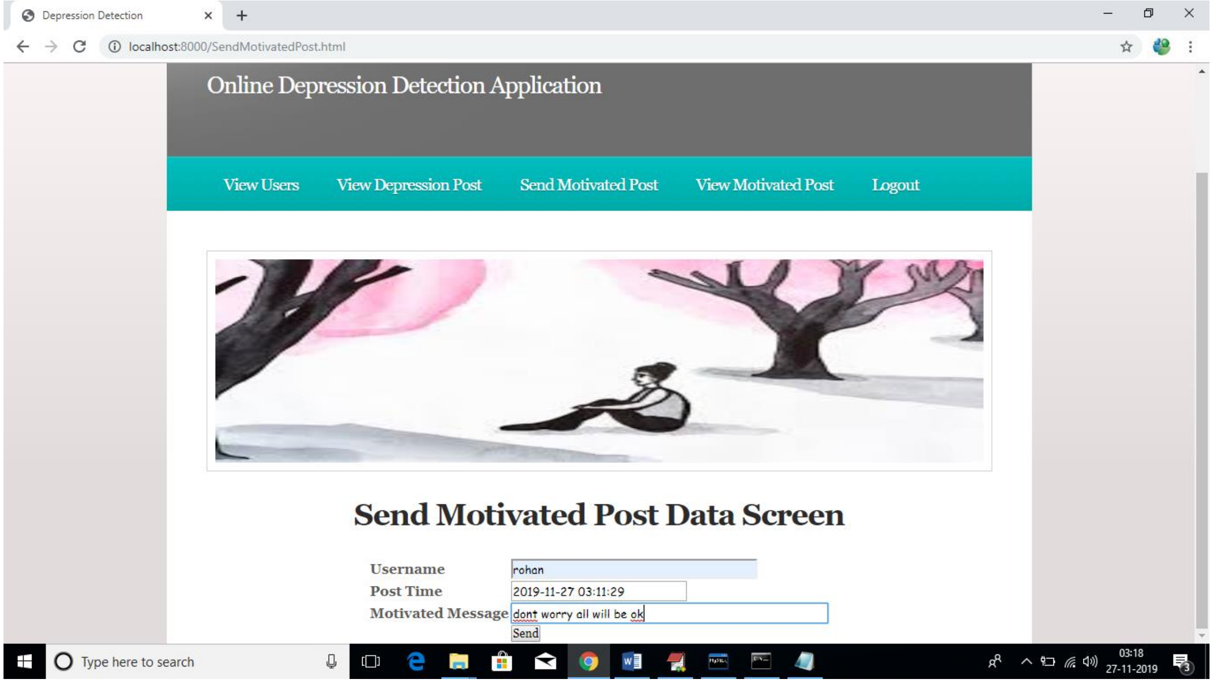
Username	Password	Contact No	Email ID	Address
rajesh	rajesh	9652861905	rajesh@gmail.com	hyd
himesh	himesh	9652861906	himesh@gmail.com	hyd
n	rihan	9652861906	rohan@gmail.com	hyd

In above graph we can see total depress and non-depress users, see below screen for all posts from all users.



Username	Post Data	Post Time	Depression	Motivated Post
rajesh	so depressing i m so sick Happy birthday my phone broke	2019-11-27 01:08:33	Negative	dont worry all will be good
himesh	last night movie was good and gave me good relief	2019-11-27 01:16:17	Positive	Pending
rajesh	movie was good	2019-11-27 01:54:38	Positive	Pending
rohan	movie was good	2019-11-27 03:08:06	Positive	Pending
rohan	so depressing i m so sick Happy birthday my phone broke	2019-11-27 03:11:29	Negative	Pending

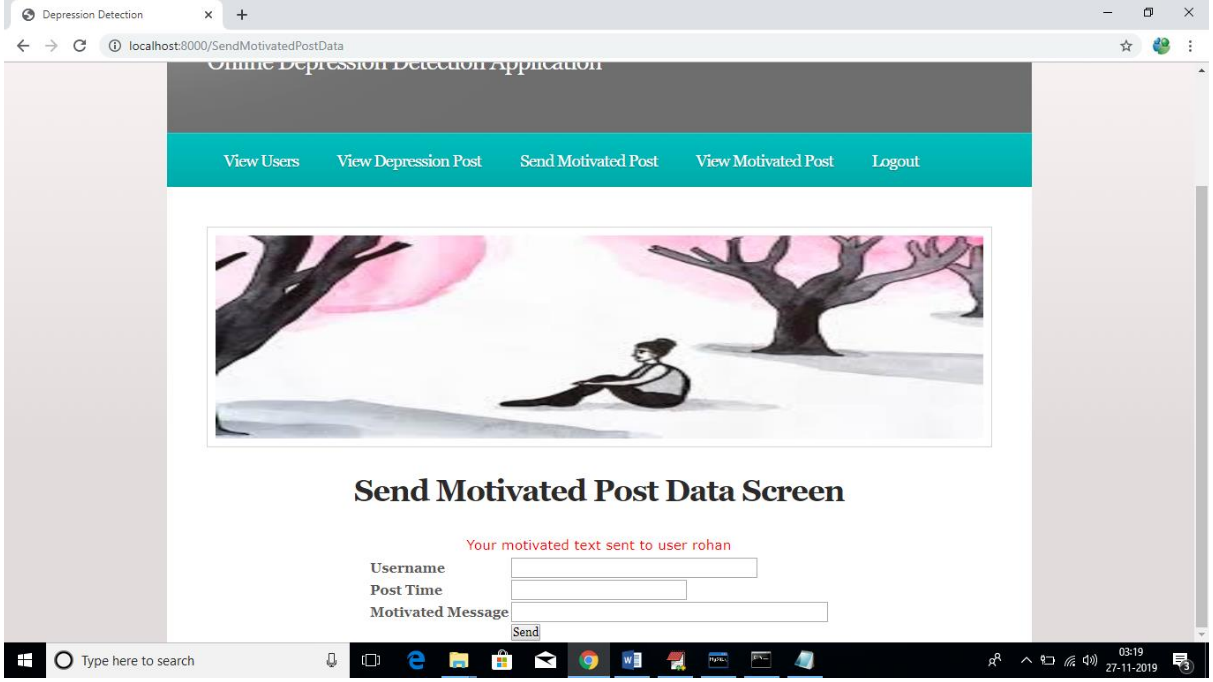
While sending messages admin has to enter username and post time. Post time he can copy from above screen.



Send Motivated Post Data Screen

Username	rohan
Post Time	2019-11-27 03:11:29
Motivated Message	dont worry all will be ok

After sending motivated messages will get below screen.



Send Motivated Post Data Screen

Your motivated text sent to user rohan

Username	
Post Time	
Motivated Message	

Now admin can click on 'View Motivated Post' link to view all motivated messages sent by him.

CHAPTER 6

CONCLUSION

CHAPTER 6

CONCLUSION

6.1 Conclusion

In this paper, we tried to identify the presence of depression in Reddit social media; and searched for affective performance increase solutions of depression detection. We characterized a closer connection between depression and a language usage by applying NLP and text classification techniques. We identified a lexicon of words more common among the depressed accounts. According to our findings, the language predictors of depression contained the words related to preoccupation with themselves, feelings of sadness, anxiety, anger, hostility or suicidal thoughts, with a greater emphasis on the present and future.This project has the potential to contribute to early detection and intervention of depression, which is crucial for improving mental health outcomes. However, it is important to note that the algorithms used in this project should not be used as a replacement for traditional diagnostic methods such as clinical interviews and assessment.

In this research paper, the aim was to detect the presence of depression in the popular social media platform, Reddit. To achieve this goal, the researchers applied natural language processing (NLP) and text classification techniques to identify patterns in language usage that are indicative of depression. The study involved the analysis of a large dataset of Reddit posts to identify the language predictors of depression.

The researchers found that there was a closer connection between depression and language usage, with certain words and phrases being more common among depressed individuals. They identified a lexicon of words that were commonly used by depressed individuals, such as those related to preoccupation with oneself, feelings of sadness, anxiety, anger, hostility, or suicidal thoughts. These words were identified as strong indicators of depression, with a greater emphasis on the present and future.

The study's findings suggest that language usage can be used as a potential tool for identifying depression in individuals. By analyzing language patterns, it may be possible to detect depression at an early stage and provide timely interventions to prevent further progression. The study also highlights the potential of NLP and text classification techniques in the field of mental health research.

Overall, this research paper provides important insights into the language predictors of depression and the potential for NLP and text classification techniques to be used in identifying depression in social media platforms. The findings have implications for the development of new approaches for detecting depression and providing timely interventions to those in need.

References

M. R. Islam, M. A. Kabir, A. Ahmed, A. R. M. Kamal, H. Wang, and A. Ulhaq, “Depression detection from social network data using machine learning techniques,” Health Information Science and Systems, vol. 6, no. 1, pp. 1–12, Dec. 2018, doi: 10.1007/s13755-018-0046-0 .

[1] M. A. Ilgen et al., “Exploratory data mining analysis identifying subgroups of patients with depression who are at high risk for suicide,” The Journal of Clinical Psychiatry, vol. 70, no. 11, pp. 1495–1500, Nov. 2009, doi: 10.4088/JCP.08m04795.

[3] A. Kumar, A. Sharma, and A. Arora, “Anxious depression prediction in real-time social data,” in proceeding of International Conference on Advanced Engineering, Science, Management and Technology–2019, 2019, pp. 1–7, doi: 10.2139/ssrn.3383359.

[2] C. Karmen, R. C. Hsiung, and T. Wetter, “Screening internet forum participants for depression symptoms by assembling and enhancing multiple NLP methods,” Computer Methods and Programs in Biomedicine, vol. 120, no. 1, pp. 27–36, Jun. 2015, doi: 10.1016/j.cmpb.2015.03.008.

[3] N. Narziev, H. Goh, K. Toshnazarov, S. A. Lee, K.-M. Chung, and Y. Noh, “STDD: Short-term depression detection with passive sensing,” Sensors, vol. 20, no. 5, pp. 1–1796, Mar. 2020, doi: 10.3390/s20051396.

[4] H. S. Alsagri and M. Ykhlef, “Machine learning-based approach for depression detection in twitter using content and activity features,” IEICE Transactions on Information and Systems, vol. 103, no. 8, pp. 1825–1832, Aug. 2020, doi: 10.1587/transinf.2020EDP7023.

[5] M. L. Reyes-Rodríguez, C. L. Rivera-Medina, L. Cámara-Fuentes, A. Suárez-Torres, and G. Bernal, “Depression symptoms and stressful life events among college students in Puerto Rico,” Journal of Affective Disorders, vol. 145, no. 3, pp. 324–330, Mar. 2013, doi: 10.1016/j.jad.2012.08.010.

[6] S. Dey, I. Sarkar, S. Chakraborty, and S. Roy, “Depression detection using intelligent algorithms from social media context-state of the art, trends and future roadmap,” Journal of Xidian University, vol. 14, no. 8, pp. 45–62, Aug. 2020, doi: 10.37896/jxu14.8/007.

[7] E. R. Vieira, E. Brown, and P. Raue, “Depression in older adults: Screening and referral,” Journal of Geriatric Physical Therapy, vol. 37, no. 1, pp. 24–30, Jan. 2014, doi: 10.1519/JPT.0b013e31828df26f.

[10] B. Zohuri and S. Zadeh, “The utility of artificial intelligence for mood analysis, depression

detection, and suicide risk management," Journal of Health Science, vol. 8, no. 2, pp. 67–73, Apr. 2020, doi: 10.17265/2328-7136/2020.02.003.

[11] F. Cacheda, D. Fernandez, F. J. Novoa, and V. Carneiro, "Early detection of depression: social network analysis and random forest techniques," Journal of Medical Internet Research, vol. 21, no. 6, pp. 1–18, Jun. 2019, doi: 10.2196/12554.

[12] 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, doi: 10.1109/ACCESS.2019.2909180.

[13] S. Smys and J. S. Raj, "Analysis of deep learning techniques for early detection of depression on social media network-a comparative study," Journal of Trends in Computer Science and Smart Technology, vol. 3, no. 1, pp. 24–39, May 2021, doi: 10.36548/jtcsst.2021.1.003.

[14] U. M. Haque, E. Kabir, and R. Khanam, "Detection of child depression using machine learning methods," PLOS ONE, vol. 16, no. 12, pp. 1–13, Dec. 2021, doi: 10.1371/journal.pone.0261131.

[15] I. A. N. Arachchige, P. Sandanapitchai, and R. Weerasinghe, "Investigating machine learning & natural language processing techniques applied for predicting depression disorder from online support forums: A systematic literature review," Information, vol. 12, no. 11, pp. 1–18, Oct. 2021, doi: 10.3390/info12110444.

[16] M. de Choudhury, E. Kiciman, M. Dredze, G. Coppersmith, and M. Kumar, "Discovering shifts to suicidal ideation from mental health content in social media," in Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, May 2016, pp. 2098–2110, doi: 10.1145/2858036.2858207.

[17] R. Chatterjee, R. K. Gupta, and B. Gupta, "Depression detection from social media posts using multinomial naive theorem," in IOP Conference Series: Materials Science and Engineering, Jan. 2021, pp. 1–12, doi: 10.1088/1757-899X/1022/1/012095.

[18] C.-T. Wu, D. Dillon, H.-C. Hsu, S. Huang, E. Barrick, and Y.-H. Liu, "Depression detection using relative EEG power induced by emotionally positive images and a conformal kernel support vector machine," Applied Sciences, vol. 8, no. 8, pp. 1–18, Jul. 2018, doi: 10.3390/app8081244.

[19] M. V. Smith, R. A. Rosenheck, M. A. Cavaleri, H. B. Howell, K. Poschman, and K. A. Yonkers, "Screening for and detection of depression, panic disorder, and PTSD in public-sector obstetric clinics," Psychiatric Services, vol. 55, no. 4, pp. 407–414, Apr. 2004, doi: 10.1176/app.

[20] H. Jung, H.-A. Park, and T.-M. Song, "Ontology-based approach to social data sentiment analysis: Detection of adolescent depression signals," Journal of Medical Internet Research.