



ARBAZ KHAN NIAZI

01-134192-011

SHAHEER REHMAN KHAN NIAZI

01-134192-077

DEPRESSION ANALYSIS ON SOCIAL MEDIA PLATFORM

Bachelor of Science in Computer Science

Supervisor: Ms.Umarah Qaseem

Department of Computer Science
Bahria University, Islamabad

May 2023

Certificate

We accept the work contained in the report titled “Depression Analysis on Social Media platform”, written by Mr.Arbaz Khan Niazi and Mr. Shaheer Rehman Khan Niazi as a confirmation of the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by . . . :

Supervisor: Ms.Umarah Qaseem

Internal Examiner:

External Examiner:

Project Coordinator: Ms. Maryam Khalid Multani (Assistant Professor)

Head of the Department: Dr. Arif ur Rahman (Sr. Associate Professor)

September 31st, 2015

Abstract

Depression is a mental health disorder that affects a significant portion of the population, and social media platforms have become a popular place for individuals to express their thoughts and feelings. In this study, we aim to analyze social media data to identify patterns and trends related to depression. This will involve collecting data from Reddit using natural language processing techniques to identify key themes and sentiment related to depression. Additionally, we will use machine learning algorithms to classify posts and identify users who may be at risk for depression. The goal of this study is to potentially identify early warning signs of depression in order to improve intervention and treatment efforts. Chapter 1 mentions a brief introduction to the system that will be used. Chapter 2 we will discuss the comparative study and similar work. In Chapter 3 we shall show the specification that are required where as in Chapter 4 show in detail the system design.

Acknowledgments

First and foremost, we would like to thank our Creator; Allah Almighty. This project would not have been possible without His blessings. We would also like to thank our extremely supportive supervisor Ms.Umarah Qaseem (Senior Lecturer) for guiding and helping us through our project-related issues. Their support, cooperation, encouragement, and constructive suggestions helped us take the project from just an idea to finished product. We would also like to thank our friends, family and especially our parents for supporting us throughout this project and the four years of the degree. SHAHEER REHMAN KHAN NIAZI ARBAZ KHAN NIAZI Islamabad, Pakistan June 2023

AUTHOR NAME
Islamabad, Pakistan

June 2023

*“We think someone else, someone smarter than us,
someone more capable, someone with more resources will solve that problem.
But there isn’t anyone else.”*

Regina Dugan

Contents

- 1 Introduction 1**
 - 1.1 Introduction 1
 - 1.2 objective 1
 - 1.3 Problem Description 1
 - 1.4 Proposed Method 2
 - 1.5 Report Organization 2

- 2 Literature Review 3**
 - 2.1 Reddit Web Crawler 3
 - 2.2 NLP model 4
 - 2.3 User Interface 4
 - 2.4 Existing solutions 4
 - 2.4.1 Phantom Buster 5
 - 2.4.2 Depression Detection Using Machine Learning Techniques on Twitter Data 5
 - 2.4.3 Deep Learning for Depression Detection of Twitter Users 5
 - 2.4.4 Comparative Study 6

- 3 Requirement Specifications 7**
 - 3.1 Existing System 7
 - 3.2 Proposed System: 7
 - 3.3 Requirement Specification: 8
 - 3.3.1 Software reuirements: 8
 - 3.3.2 Hardware Requirements 8
 - 3.3.3 Functional Requirement 8
 - 3.3.4 Non-Functional Requirements 8
 - 3.4 Use Cases 9
 - 3.4.1 Sign-Up Use Case 9
 - 3.4.2 Login Use Case 9
 - 3.4.3 Search User Use Case 10
 - 3.4.4 Perform Depression analysis Use Case 11
 - 3.4.5 Show Final Record Use Case 12

- 4 Design 14**
 - 4.1 System Architecture 14
 - 4.2 Design Methodology: 15
 - 4.3 High Level Design 16

4.3.1	Component Diagram:	16
4.4	Low Level Design	16
4.4.1	Use Case	16
4.4.2	Sequence Diagram	17
4.5	Database Design	19
4.6	GUI Design	20
4.6.1	Login:	20
4.6.2	Sign Up:	20
4.6.3	Search Page	21
4.6.4	Results	21
4.6.5	Circle Graph	22
5	System Implementation	23
5.1	Tools and Technologies	23
5.1.1	Explanation:	23
5.2	Front end:	24
5.2.1	Django(Django,n.d.)[1]	24
5.2.2	Login	24
5.2.3	Sign Up:	25
5.2.4	Search UserID:	26
5.2.5	Security	26
5.2.6	Final Result	26
5.2.7	Circle Graph	27
5.2.8	SQLite	28
5.3	Back End:	28
5.3.1	Scrapy(Scrapy, n.d.)	28
5.3.2	Selenium (Selenium, n.d.)[2]	28
5.3.3	FastAPI (FastAPI, n.d.)	29
5.3.4	Dataset	29
5.3.5	PostgreSQL	31
5.3.6	Natural Language Processing[3]	31
5.4	Work Flow	34
5.5	Dictionary	35
5.5.1	Tokenized	36
5.6	LSTM	36
5.7	NGRAMS	37
6	System Testing and Evaluation	38
6.1	Graphical user interface testing	38
6.2	Usability Testing	38
6.3	Software performance testing	39
6.4	Security Testing	39
6.5	Confusion Matrix:	40
6.5.1	Different Model Results:	40
7	Conclusions	42
A	User Manual	43

References

44

List of Figures

2.1	6
3.1	Sign-Up	9
3.2	Login Use Case	10
3.3	Search User Case	11
3.4	Perform Depression Analysis use case	12
3.5	Show Final Record Use Case	13
4.1	System Architecture Diagram	14
4.2	agile methodology incorporated for the development of Depression analysis service	15
4.3	Component diagram of Depression Analysis on Social media	16
4.4	Use Case	17
4.5	Sign Up sequence diagram	17
4.6	Login sequence diagram	18
4.7	Search keyword Sequence Diagram	18
4.8	Analysis sequence diagram	19
4.9	Database Design	19
4.10	Login Page	20
4.11	Sign Up Page	20
4.12	Search User ID	21
4.13	Result Page	21
4.14	Circle Graph	22
5.1	Login Page	25
5.2	Sign Up Page	25
5.3	Search UserID	26
5.4	Result	27
5.5	Circle Graph	27
5.6	Training Dataset	30
5.7	Comparison Table	34
5.8	Flow Chart	34
5.9	Dictionary	35
5.10	Tokenized Data	36
5.11	Ngram Parameters	37
6.1	Confuse Matrix	40

List of Tables

3.1	Sign-Up Use Case	9
3.2	Login Use Case	10
3.3	Search User Use Case	11
3.4	Perform Depression Analysis Use Case	12
3.5	Show Final Record Use Case	13
6.1	GUI General and Forms Test Case	38
6.2	Usability test case	39
6.3	Performance test case	39
6.4	Security test case	40
6.5	Models	41

Acronyms and Abbreviations

DSA	Data Structure and Algorithms
OOP	Object Oriented Programming
PF	Programming Fundamentals
SE	Software Engineering
SQL	Structured Query Language
PE	Portable Executable
EXE	Executable
Retdec	Retargetable machine-code decompiler based on LLVM
ML	Machine Learning

Chapter 1

Introduction

1.1 Introduction

Depression is a mental health disorder characterized by persistent feelings of sadness, hopelessness, and a loss of interest in activities. It can also cause physical symptoms such as fatigue, changes in appetite and sleep patterns, and difficulty concentrating. Depression is a serious condition that can affect a person's ability to function in daily life and can lead to serious problems if left untreated. Depression can be caused by a combination of genetic, biological, environmental, and psychological factors. Imbalance of chemicals in brain called neurotransmitters may also play a role in depression. Some people may have a genetic predisposition to depression while may develop the condition in response to stress or such as a difficult life event or relationship problems.

1.2 objective

The objective of this study is to analyze social media data to identify patterns and trends related to depression, and to use machine learning algorithms to classify posts and identify users who may be at risk for depression, in order to improve intervention and treatment efforts. The Project aims to use the Natural Language Processing modules to perform sentiment analysis on the Posts to detect depression-related posts on Reddit (Reddit, 2022) [4]

1.3 Problem Description

This Project is basically a Natural language processing (Natural language processing, 2022) system that will detect depression-related posts on social media. we are making this project

because Depression has an impact on the behavior of the affected individuals. The key objective of our project is to examine Reddit (Reddit, 2022) users' posts to detect whether the user has posted a depressing post or not and through that, get the average of depressed users on Reddit social media. The Dataset will be made by using Scrapy selenium (pypi, 2022)[5]. We will make our own Dataset and perform an analysis on the Dataset.

1.4 Proposed Method

The project is built using Python as the main language. While using this project the Dataset will have binary labeling. We will have Online Labelled Dataset and if Online Dataset is not available then we will generate our own Dataset through Scrapy through this Dataset, we will train our Dataset and perform the analysis based on that, and through that, we will calculate the accuracy of the predictions.

1.5 Report Organization

Chapter 1 starting off with the introduction while Chapter 2 revolves around the literature review. After that we have Chapter 3 which showcases the basic requirements and lastly Chapter 4 gives a detail overview of the system design.

Chapter 2

Literature Review

Depression is a common mental disorder that affects individuals of all ages and can cause a range of symptoms, including feelings of sadness, hopelessness, and loss of interest in activities. In recent years, there has been a significant increase in the number of people experiencing depression, with rates of depression rising across society. This may be due to a variety of factors, such as increased stress and pressure, changes in societal norms and expectations, and access to mental health resources. Nowadays the generation z uses a lot of social media and express their feelings on social media. Reddit is one of most commonly used social media in today's era and contain extensive textual data about experiences of people and their daily life. So therefore depression could be diagnosed from these social media posts. For that purpose our team has built a web crawler that gets data from different reddit post's and further depression detection could be done using NLP techniques.

2.1 Reddit Web Crawler

The Reddit web crawler is built using the scrapy-selenium framework. Scrapy-Selenium is a package that allows Scrapy [6], a popular web scraping framework, to interact with web pages rendered by JavaScript using Selenium, a browser automation tool. This allows Scrapy to scrape websites that require JavaScript to load content, as well as interact with pages in ways that would not be possible using just Scrapy alone. Scrapy-Selenium allows the user to configure a Selenium web driver in the Scrapy settings and use it in the spider to navigate and interact with the website. The crawler requires the input in form of the link and it scrapes data from that particular web page. The web-crawler gets different textual data.

- Post Date

- Post link
- Post Text
- User profile

link other data maybe included based on the need of the model.

2.2 NLP model

This model uses a combination of machine learning techniques such as natural language processing (NLP) and sentiment analysis to classify posts as either depressed or non-depressed based on the language used in the Reddit Posts. The model will be trained on a dataset of posts labeled as depressed or non-depressed, and uses features such as the presence of certain keywords and the sentiment of the tweet to make its classification. The model's performance is evaluated using metrics such as accuracy, precision, and recall. Another approach could be to use unsupervised learning to label data gathered from the crawler and separate it as depressed or non-depressed and then use NLP model to train the model to predict depression. It is worth noting that unsupervised learning methods alone may not be able to achieve high accuracy results in depression detection task, because, unsupervised methods only group similar posts together, but can not explicitly label them as depressed or non-depressed. Therefore, unsupervised methods should be combined with other techniques such as supervised learning, NLP and sentiment analysis to achieve good results.

2.3 User Interface

This is an important factor while considering the whole system as it displays the analysis gathered from the profile . The user interface will have an input of the link of user profile and will predict the depression analysis. The user interface will be a web application and will be made using django framework.

2.4 Existing solutions

Depression analysis is a field being worked on by different researchers and has come into light after the development of different machine learning and artificial intelligence techniques some of them are following

2.4.1 Phantom Buster

Phantombuster is a web automation platform that allows users to automate various online tasks, such as data extraction, social media automation, and web scraping. The platform utilizes a browser extension and a cloud-based automation engine to perform these tasks, allowing users to automate repetitive and time-consuming processes without the need for coding or programming knowledge. Phantombuster's automation engine can be used to extract data from websites and social media platforms, such as LinkedIn, Twitter, and Facebook. The platform also offers pre-built scripts and APIs that can be used to automate tasks such as lead generation, email scraping, and account creation. Additionally, the platform provides analytics and visualization tools that allow users to better understand the data they have collected. Phantombuster is primarily used by businesses and marketing professionals to automate repetitive tasks and gain insights from online data. It's designed to help users increase their productivity, save time and reduce human error. we took inspiration from this system to create our web crawler and do analysis on the data. PhantomBuster at its core is a web-crawler and has extended its limits to different social media sites. Our project is basically scaled down version of PhantomBuster[7]

2.4.2 Depression Detection Using Machine Learning Techniques on Twitter Data

[8]

Twitter is an application which is focused on 140 characters per tweets enables the user to share their opinion and thought shortly and directly. Each tweet enables the researchers to extract and analyze the information shared in the tweet. The sentiment analysis technique is applied to each tweet to identify the sentiment score and labelled them as positive, negative, or neutral. The labeled tweets are fed into a machine learning algorithm that enables the classification of the tweets into correct groups. Naive Bayes and NBTree which are the selected machine learning algorithm have been implemented on two different sizes of tweet datasets to find the accuracy of the algorithm on classifying the depressive and non-depressive tweets.

2.4.3 Deep Learning for Depression Detection of Twitter Users

Deep learning is a type of machine learning that utilizes neural networks to model and analyze data. In the context of depression detection on Twitter, researchers may use deep learning algorithms to analyze the text of tweets and identify patterns or language that may indicate depression. The authors of this paper may have used a dataset of tweets from known depressed individuals and trained a deep learning model on this data in order to identify similar patterns in tweets from other users. The goal of this research may be to

create a tool that can automatically identify individuals on Twitter who may be suffering from depression and provide them with resources or support [9]

2.4.4 Comparative Study

TITLE	MEHTOD	TECHNIQUE	SCRIPT	DATASET	RESULTS
<u>PhantomBuster</u>	Web-Automation	Web crawler	English	Social media sites	
Depression Detection Using Machine Learning Techniques on Twitter Data	Textual	Naive Bayes and <u>NBTree</u>	English	Twitter	<ul style="list-style-type: none"> • The <u>NBTree</u> algorithm gives an accuracy of 97.31% to classify the depressive and non-depressive tweets • <u>Naïve Bayes</u> show 97.31% on the 3000 tweets dataset and 92.34 % on 1000 tweets datasets.
Deep Learning for Depression Detection of Twitter Users	Textual	CNN-based and <u>RNN-based</u> models to determine the best models and parameters across different settings for depression detection.	English	The training data consists of 1,145 Twitter users labeled as Control, Depressed, and PTSD (<u>Coppersmith et al., 2015b</u>)	<ul style="list-style-type: none"> • experiments showed that our <u>CNNbased</u> models perform better than <u>RNN-based</u> models. Models with optimized <u>embeddings</u> managed to maintain performance with the generalization ability.

Figure 2.1

Chapter 3

Requirement Specifications

3.1 Existing System

1. **Phantom Buster:** Phantom buster is primarily used by businesses and marketing professionals to automate repetitive tasks and gain insights from online data. Phantom Buster at its core is a web-crawler and has extended its limits to different social media sites.
2. **Depression Analysis on Twitter:** Naive Bayes and NBTree which are the selected machine learning algorithm have been implemented on two different sizes of tweet datasets to find the accuracy of the algorithm on classifying the depressive and non-depressive tweets.
3. **Deep Learning for Depression Detection of Twitter Users:** The goal of this research may be to create a tool that can automatically identify individuals on Twitter who may be suffering from depression and provide them with resources or support.

3.2 Proposed System:

The system our team is trying to build consists of a NanP model that detects the depression of a particular user from reddit. The system first collects data from the user's profile and then process and saves data to the database. The data is collected through a web crawler and then pre- processing is done using conventional techniques of python. The user can access this system through web-application.

3.3 Requirement Specification:

We now formally present you the requirement specification of our system in the following sections.

3.3.1 Software requirements:

The Software requirements for this project are:

1. **Python** [10]
2. **PostgreSQL** [11]
3. **Microsoft Excel**
4. **Overleaf(latex Editor)**

3.3.2 Hardware Requirements

1. **Computer or laptop/browser**

3.3.3 Functional Requirement

1. **An interface that makes user able to register, login and manage their accounts**
2. **Interface that will allow user to search the Username of the person on the search page and data will be crawled.**
3. **allow user to search the Username of the person and analysis will be performed on the crawled data**
4. **Providing a database for the storage of crawled data**

3.3.4 Non-Functional Requirements

1. **Performance**
2. **Scalability**
3. **Flexibility**
4. **Security Requirement**
5. **Internet Connectivity**

3.4 Use Cases

3.4.1 Sign-Up Use Case

This use case allows users to sign up for our database so he/she can get the benefits of the system.as shown in table 3.1 and figure 3.2

Table 3.1: Sign-Up Use Case

Use Case ID	UC001	
Use Case Name	Sign up	
Actor(s)	User	
Pre-condition	The user wants to use the website but does not have an account	
Description	Normal Flow of Events This Use case allows to sign up to our database so he/she can log in to our system.	Alternative Flow of Events Wrong Input. Enter the data again.
Post Conditions	The user now has an account and can freely use the Website.	
Comments	None	

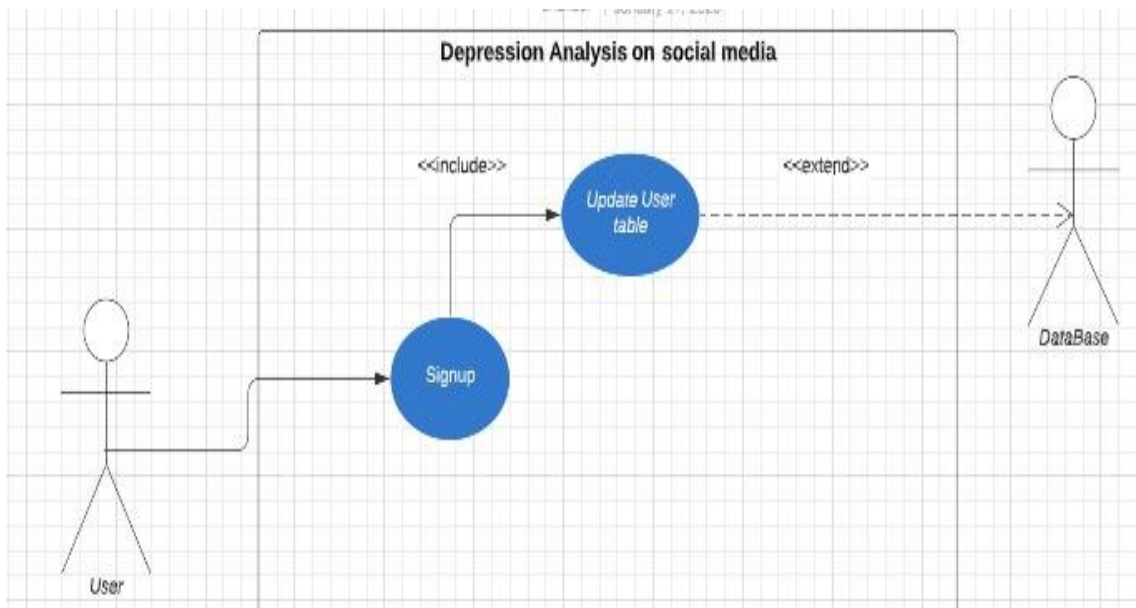


Figure 3.1: Sign-Up

3.4.2 Login Use Case

This use case offers you to get yourself logged into the system. To login to the system the user has to provide his/her specific username and password. Upon successful login, the user will be redirected to the next page. As shown in table 3.2 and figure 3.2

Table 3.2: Login Use Case

Use Case ID	UC002	
Use Case Name	Login	
Actor(s)	User	
Pre-condition	User wants to use the website but isn't logged in	
Description	<p>Normal Flow of Events The user opens the Login page, enters the required credentials, and upon successful completion, is logged into his/her account.</p>	<p>Alternative Flow of Events Error is displayed if the user enters wrong login credentials, the error will be "invalid login credentials."</p>
Post Conditions	User can resume using the website	
Comments	None	

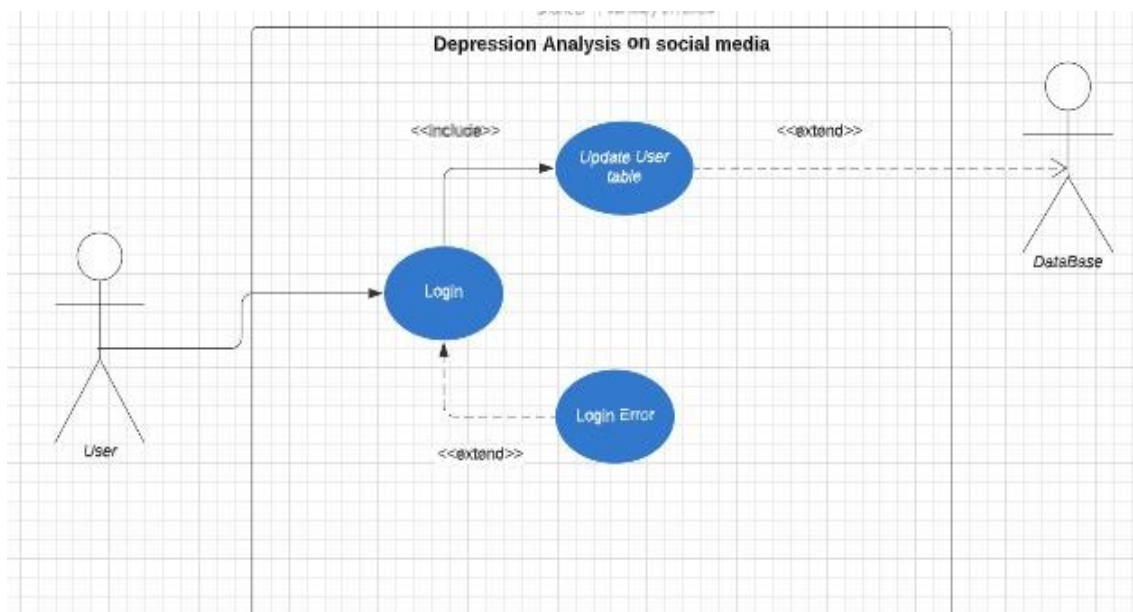


Figure 3.2: Login Use Case

3.4.3 Search User Use Case

The user will enter the username of the specific user In the search bar and then press enter button to continue. As shown as in the table 3.3 and figure 3.3

Table 3.3: Search User Use Case

Use Case ID	UC003	
Use Case Name	Search User	
Actor(s)	User	
Pre-condition	User must be logged in first.	
Description	Normal Flow of Events The user enter the username of the specific user and then data of the that specific username will be crawled.	Alternative Flow of Events Wrong userID will give null data and error will be displayed as “no record Found”
Post Conditions	Data will be crawled	
Comments	None	

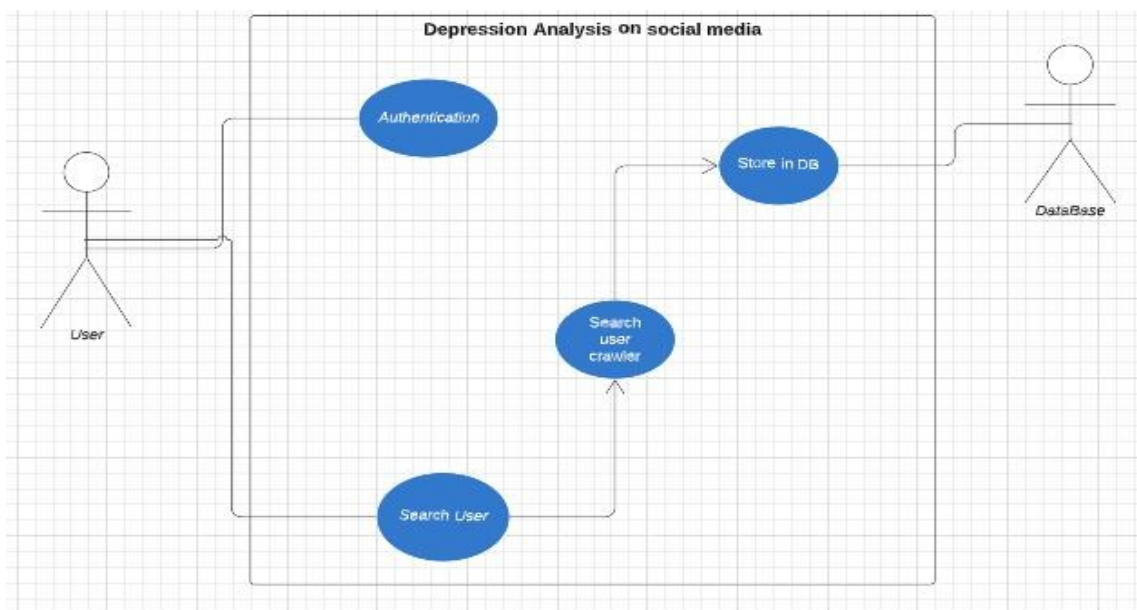


Figure 3.3: Search User Case

3.4.4 Perform Depression analysis Use Case

After searching the crawler search the data will be saved in Database and on that textual data Depression analysis will be performed and save the records to the database. As shown in table 3.4 and figure 3.4

Table 3.4: Perform Depression Analysis Use Case

Use Case ID	UC004	
Use Case Name	Perform Depression Analysis use case	
Actor(s)	User,Database	
Pre-condition	User must be logged in first.	
Description	Normal Flow of Events After data is stored in Database, on that data Prediction will be performed that whether Post is depressed or not and the records will be stored in Database.	Alternative Flow of Events Error will be displayed as “no Record found” if the User didn’t post anything Wrong.
Post Conditions	Predictions will be made.	
Comments	None	

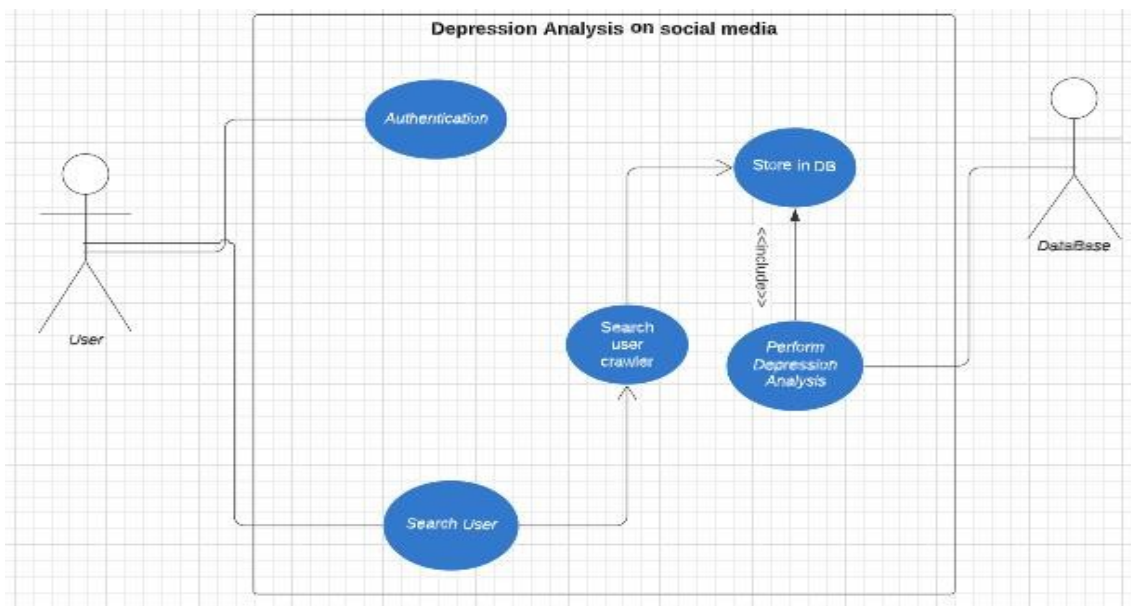


Figure 3.4: Perform Depression Analysis use case

3.4.5 Show Final Record Use Case

After the analysis is done then it will display the result on the system.as shown in Table 3.5 and Figure 3.5.

Table 3.5: Show Final Record Use Case

Use Case ID	UC005	
Use Case Name	Show Final Record	
Actor(s)	User	
Pre-condition	Analysis must be completed.	
Description	Normal Flow of Events After the analysis the result of the analysis will be displayed on the system.	Alternative Flow of Events Error will be displayed as “no Record found” if the User didn’t post anything.
Post Conditions	Result has been displayed.	
Comments	None	

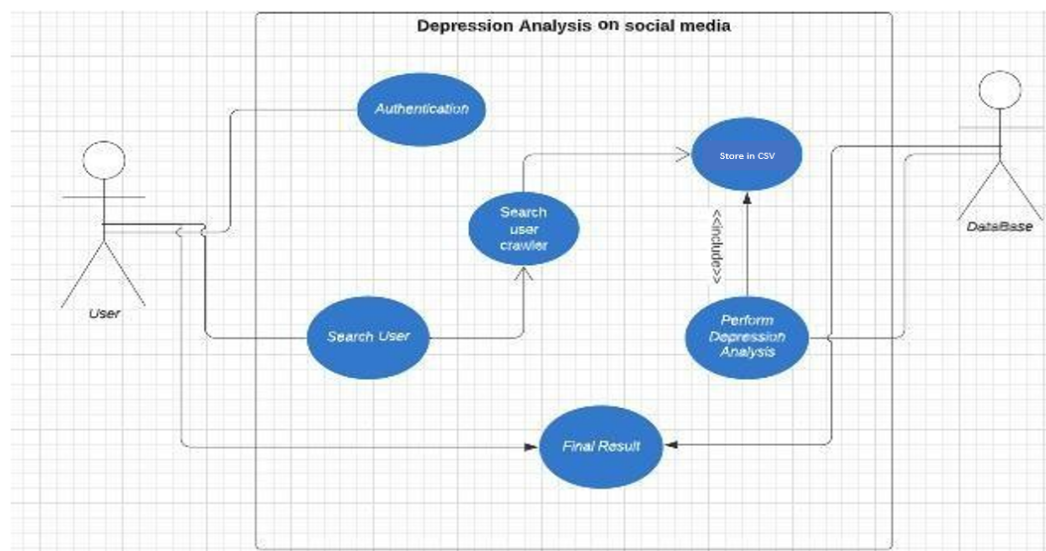


Figure 3.5: Show Final Record Use Case

Chapter 4

Design

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. The following sections constitute this chapter:

4.1 System Architecture

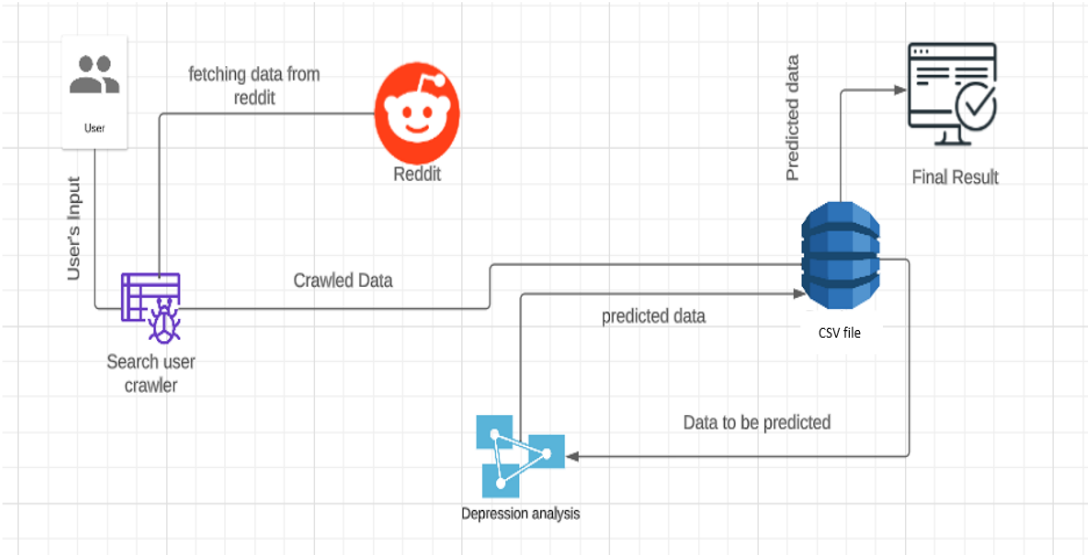


Figure 4.1: System Architecture Diagram

In above Figure 4.1, the system architecture is shown in which the data is crawled for Reddit social media according to the input that the user gave, and after that data is populated

in the database. The data is further going to the analysis phase and after prediction, the predicted data is populated to the database and that data is further displayed on the web.

4.2 Design Methodology:

The proposed project will be implemented using an agile methodology. When it comes to accomplishing this goal, the agile technique will prove very beneficial. According to the agile methodology, the planned project is divided into various phases. Each phase demands continuous communication and feedback from team members, as well as continuous improvement. The agile model incorporates the following steps:

1. Planning
2. Requirements Analysis
3. Design
4. Development
5. Testing
6. Evaluation

The following diagram 4.2 shows the design methodology followed for our project.

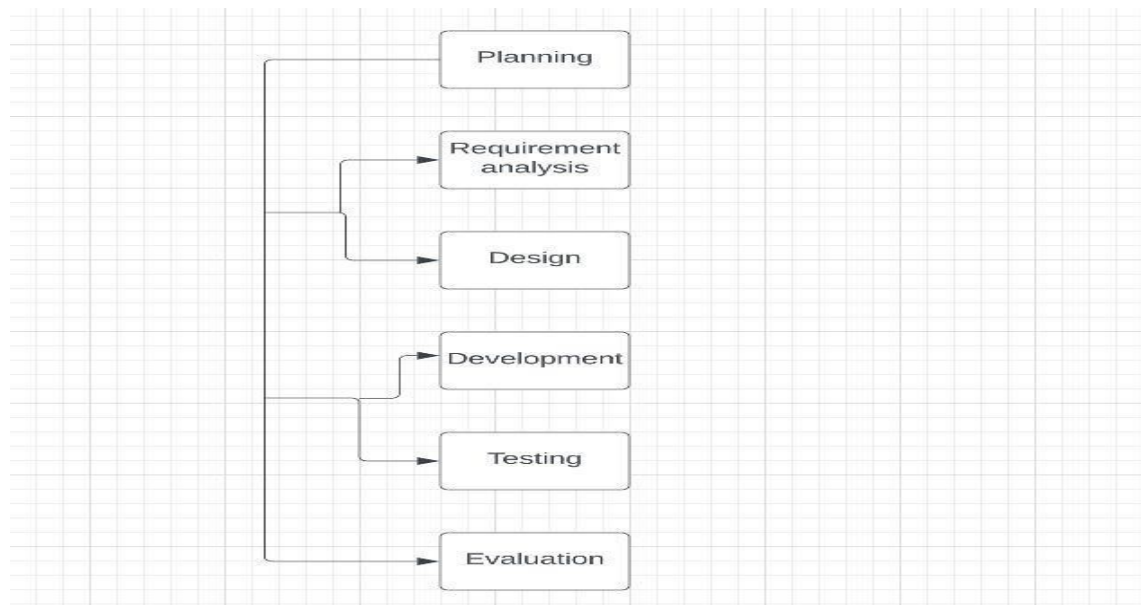


Figure 4.2: agile methodology incorporated for the development of Depression analysis service

4.3 High Level Design

This section describes in further detail elements discussed in the Architecture. High-level designs are most effective if they attempt to model groups of system elements from a number of different views. Typical viewpoints are:

4.3.1 Component Diagram:

A component Diagram is used to break down large component into small components for better managing and the component diagram for our system is illustrated in Figure 4.3

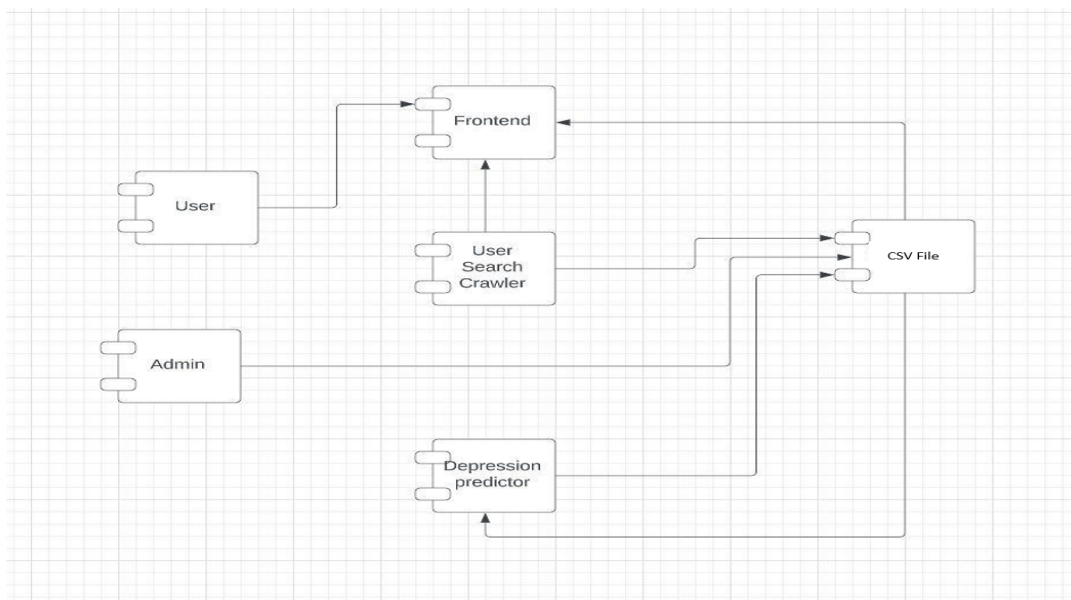


Figure 4.3: Component diagram of Depression Analysis on Social media

4.4 Low Level Design

4.4.1 Use Case

The system's use case shown in Figure 4.4 visualizes how an actor uses the system. At first, the actor needs to log in to the system to make use of the facilities. After login, the user enters the main page of the system where the user enters the search to query that the system starts crawling relevant data from the Reddit platform. The crawling of the data will be populated in the database. The sentimental analysis will be performed over the stored data and then the final result will be shown.

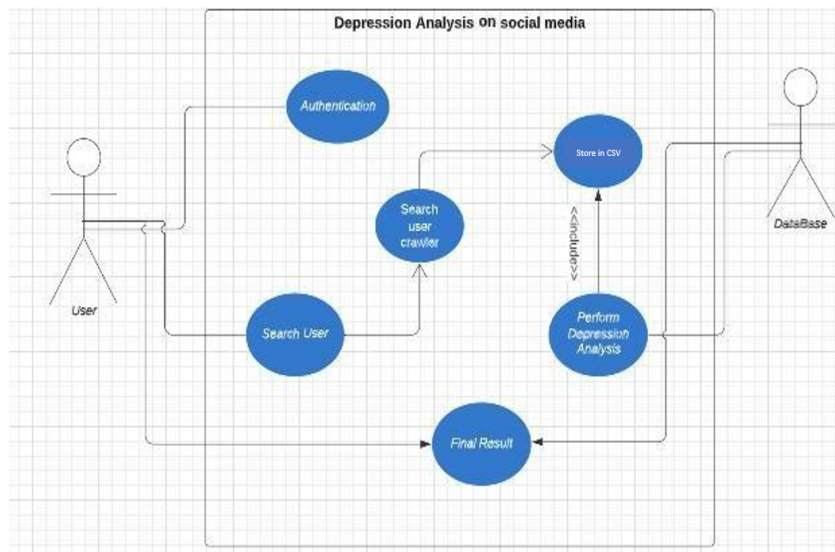


Figure 4.4: Use Case

4.4.2 Sequence Diagram

Here we have defined the sequence Diagram.

4.4.2.1 Sign Up Sequence diagram

Figure 4.5 shows how a user will sign up while following the sequence as depicted in the diagram.

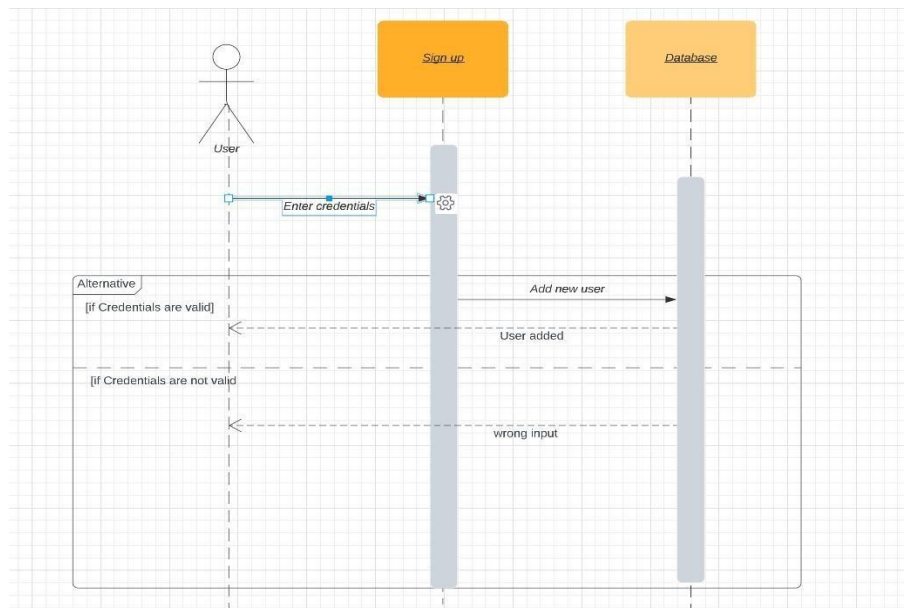


Figure 4.5: Sign Up sequence diagram

4.4.2.2 Login Sequence diagram

Figure 4.6 shows a user will log in to the system.

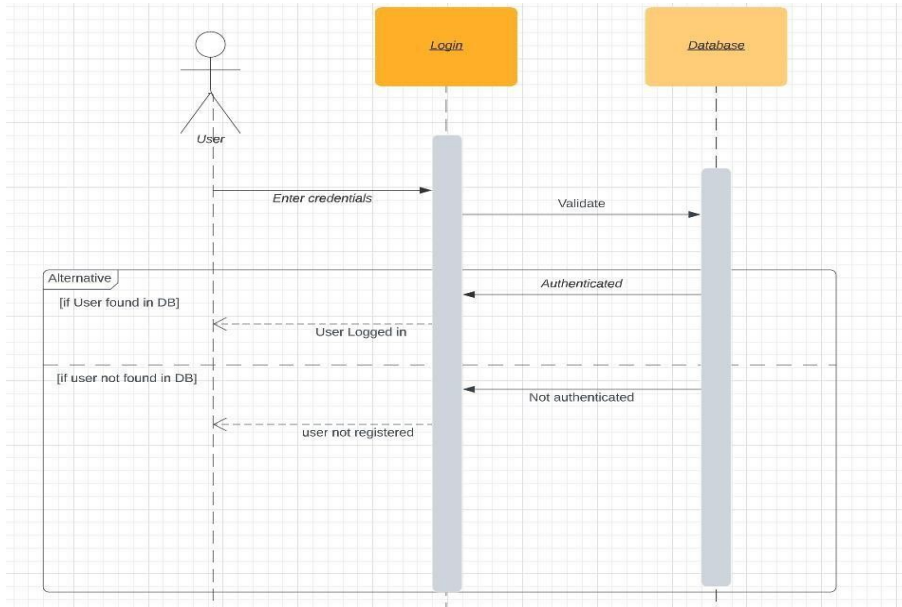


Figure 4.6: Login sequence diagram

4.4.2.3 Search Keyword Sequence Diagram:

Figure 4.7 shows how a user will be able to search the specific Username and the data of that username will be crawled and populated in the CSV file.

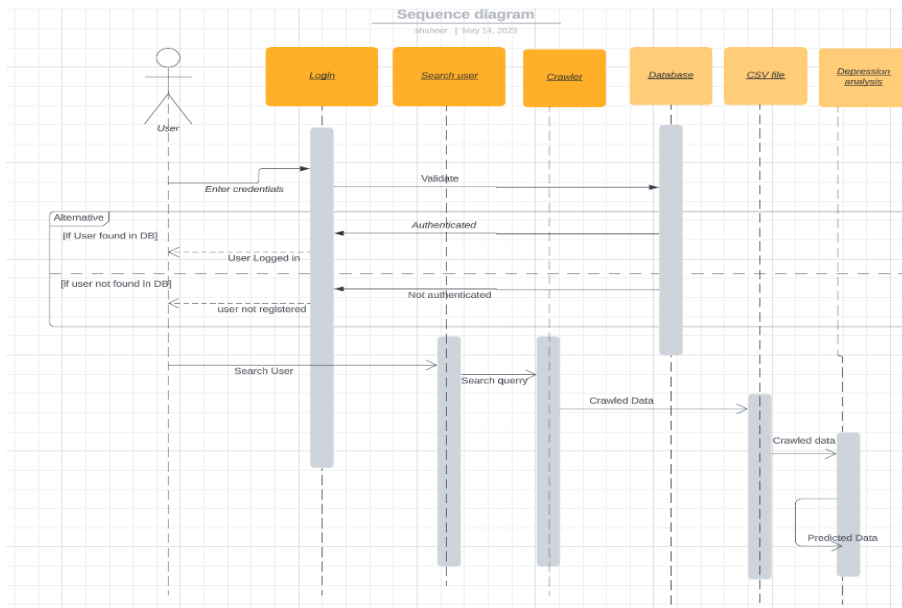


Figure 4.7: Search keyword Sequence Diagram

4.4.2.4 Analysis Sequence diagram

Figure 4.8 shows how analysis will be performed.

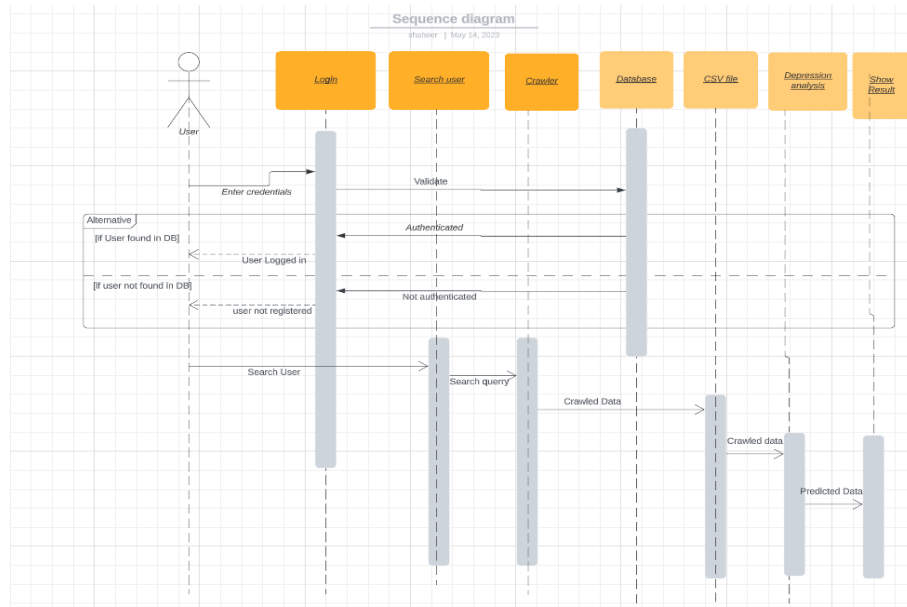


Figure 4.8: Analysis sequence diagram

4.5 Database Design

The following Figure 4.9 is the database design. It highlights the attributes and relation of thesearch User Crawler.

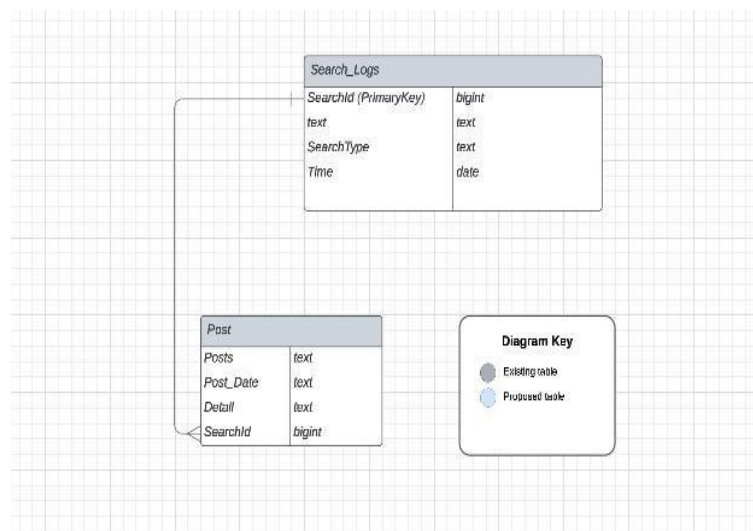


Figure 4.9: Database Design

4.6 GUI Design

The GUI is designed using Django Framework GUI design is shown below:

4.6.1 Login:

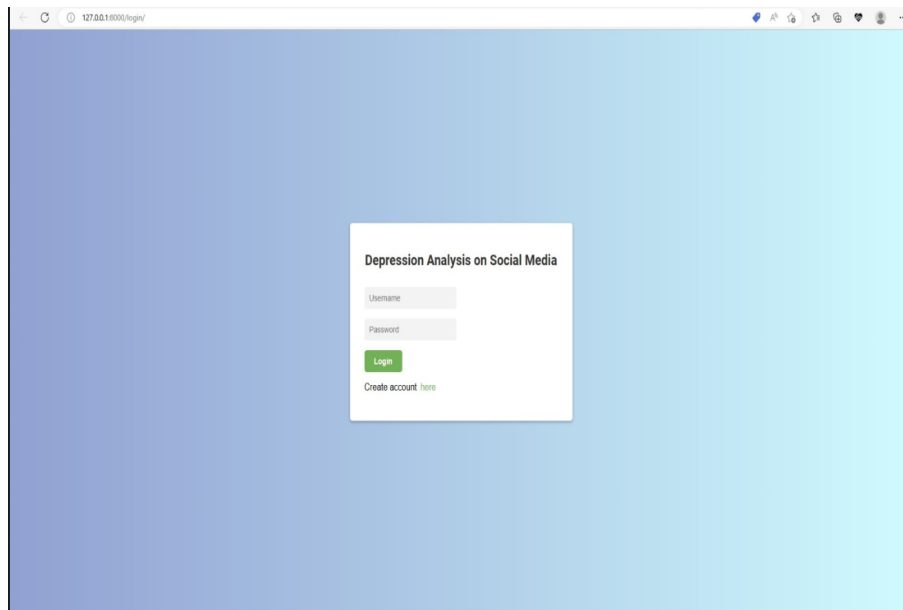


Figure 4.10: Login Page

4.6.2 Sign Up:

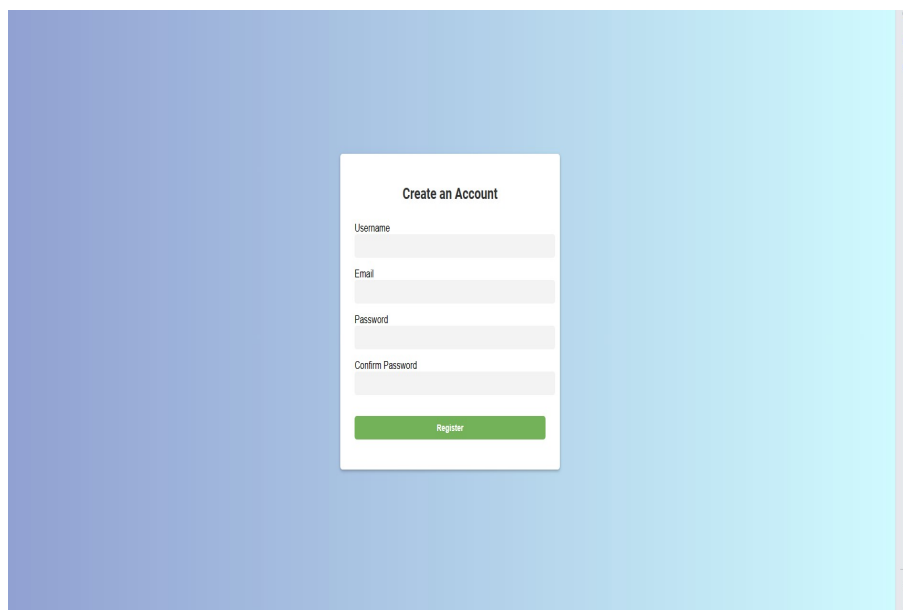


Figure 4.11: Sign Up Page

4.6.3 Search Page

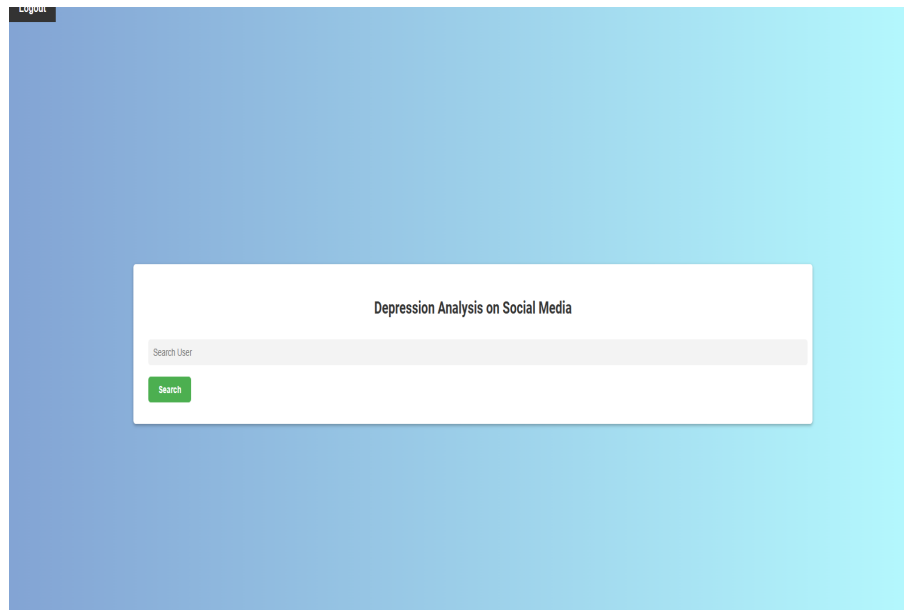


Figure 4.12: Search User ID

4.6.4 Results

Results

#	Posts	Predictions
1	the researchers learned that one of the fats in the mediterranean diet oleic acid increases the number of two key cellular structures or organelles and protects cellular membranes from damage by a chemical reaction called oxidation this protective effect has a big payoff worms fed food rich in oleic acid lived about longer than those consigned to standard worm rations the researchers found oleic acid the researchers	0
2	almeno nei vermi da laboratorio	0
3	students whose families used the handbook reported their alcohol use over the past days had increased once they got to college compared to a increase among students whose parents didn't receive the book cannabis use went up for those control students but only for students whose families used the book families used whose families students whose the book used the	1
4	the study assessed the interaction between physical exertion and short term memory performance when distractors were present or absent in younger and older adults	0
5	punto cardine della causa è il fatto tuonano da greenpeace che per la prima volta avvenga l'accertamento danni che un giudice riconosca le responsabilità in termini di crisi climatica del soggetto di diritto italiano eni oggi da considerare il maggiore responsabile di emissioni climalteranti in italia	0
6	in che modo mettere un minimo a k toglie potere ai sindacati di contrattare a k numeri a caso se non sono capaci di negoziare per i propri iscritti che chiudano a k	0
7	paper	0
8	source although it makes sense because it might narrow the gap between economies hard to believe polls in china russia even turkey	0
9	paper	0
10	our findings provide new insights into the computational mechanisms used by the human brain in perceptual judgments about the relation between ourselves and the external world	0
11	source	0
12	non ho capito se bannano gli ip che trasmettono o bloccano anche gli ip che ricevono perchè se vogliono bloccare anche chi guarda voglio vedere come fanno a chi è dietro nat gli bannano l'intera famiglia condominio perchè un tizio solo stava guardando ip che gli ip	0
13	the detail numbers for every country	0
14	paper	0
15	paper	0
16	source	0

Figure 4.13: Result Page

4.6.5 Circle Graph

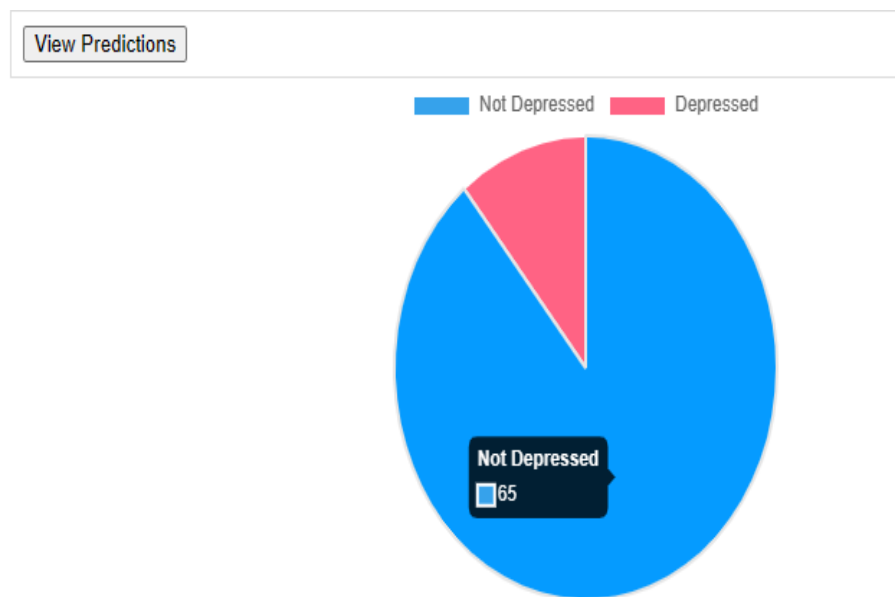


Figure 4.14: Circle Graph

Chapter 5

System Implementation

This chapter presents in detail the implementation details of the system in the following sections. There were two major parts of the system architecture: back-end, and front-end. The back end consists of FastAPI which helps to crawl the Reddit Posts and perform analysis on the crawled data. The Front-End consists of a Web application that allows the user to give the id of the product on which he/she wants to analyze depression.

5.1 Tools and Technologies

The tools used in the system implementation are given below:

- PostgreSQL
- SQLite (SQLite, n.d.) [12]
- Web Browser
- PyCharm (pyCharm, n.d.) [13]

5.1.1 Explanation:

- Latex: We have used Latex version 2022 for documentation on Overleaf.
- MS PowerPoint: We have used Ms. PowerPoint 2019 for the presentation of our system.
- PyCharm: We have used PyCharm for creating the system.
- Django: We have used the Django web app for the Front-end Development of the Systems Interface.

- Python Language: We have used Python as the main language for the development of our System.
- PostgreSQL: We have used PostgreSQL as a database for the collection of training datasets because it is a popular open-source relational database management system that offers many advantages for developers and organizations.
- SQLite: We have used SQLite as a database for the authorized user records for the Web Page.

5.2 Front end:

The front end of the system includes the Web Application. Let us see the functionality of the Web Application.

5.2.1 Django(Django,n.d.)(1]

We have used the Django framework for the front-End because Django is a popular open-source web Framework for Python that offers many advantages for developers building web applications. Some of the advantages are as follows:

- Security: Django provides built-in security features such as protection against SQL injection, cross-site scripting (XSS) attacks, etc., making it a secure choice for web applications
- Rapid Development: Django provides a high-level, Pythonic syntax that allows developers to write code quickly and efficiently.
- Compatibility: Django is compatible with many popular Python libraries and frameworks, making it easy to integrate with other tools and technologies.

5.2.2 Login

Provided Login functionality for user authentication. Login features are an essential component that requires to authenticate their identity.

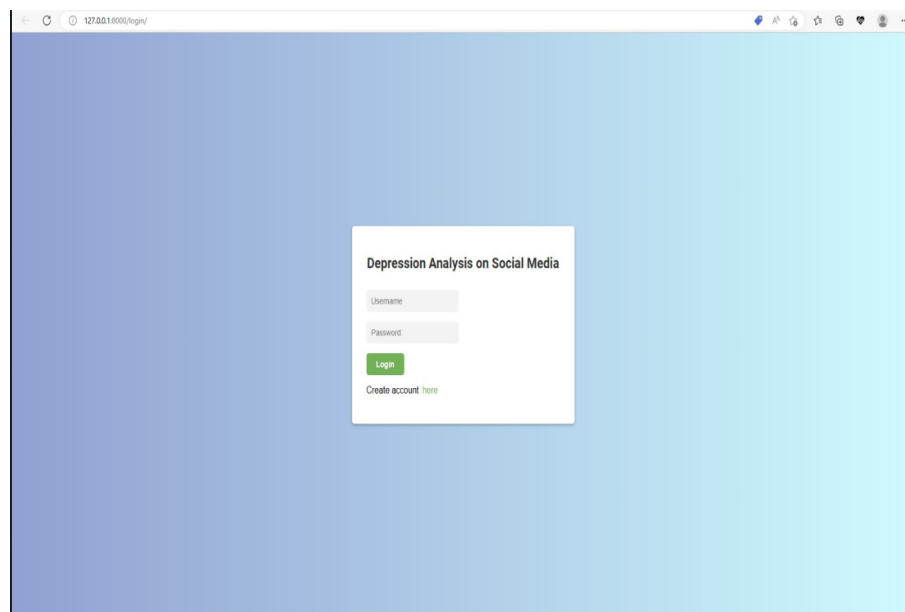


Figure 5.1: Login Page

5.2.3 Sign Up:

Provided Sign Up functionality for the users to get registered for the legal usage of the Web Applications. The records are being saved in SQLite database.

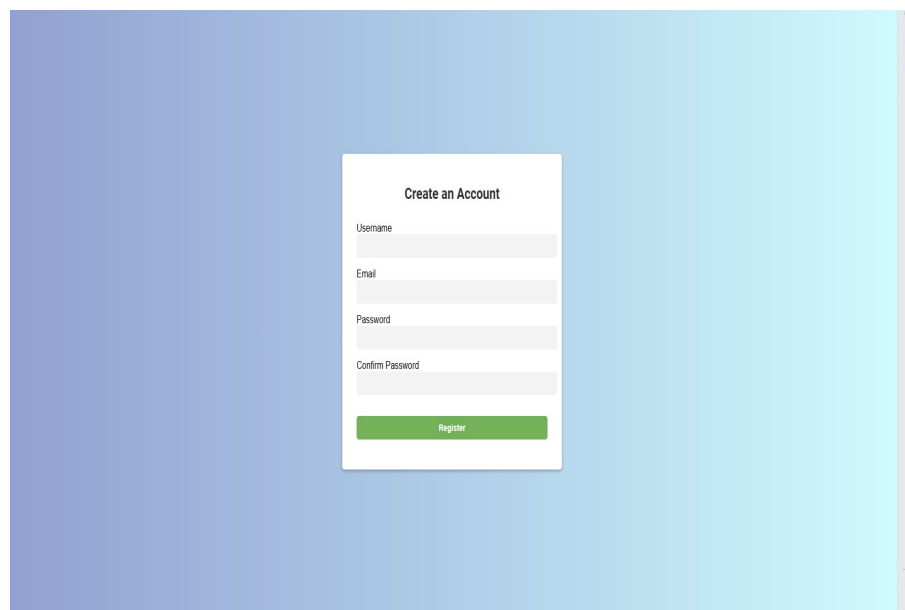


Figure 5.2: Sign Up Page

5.2.4 Search UserID:

Provided Search bar for the user to type the exact Username of the User on which he/she wants to perform the analysis.

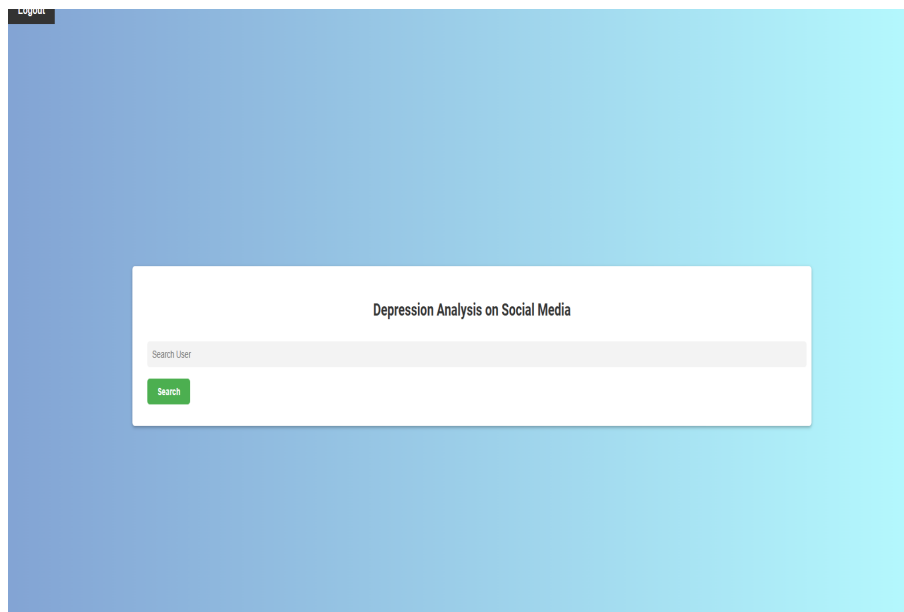


Figure 5.3: Search UserID

5.2.5 Security

For Security, we have used the Django built-in security features. Encrypted the Password with * for security measures.

5.2.6 Final Result

This Result table shows the Final predicted result of the Searched User.

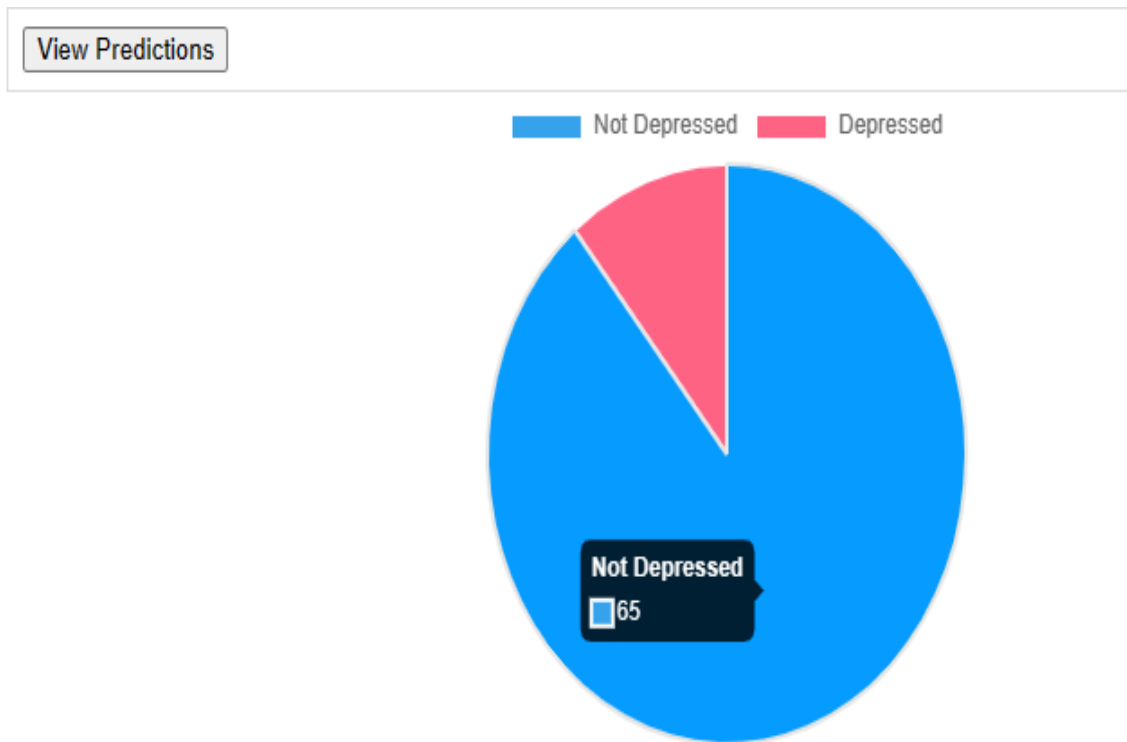
Results

#	Posts	Predictions
1	the researchers learned that one of the fats in the mediterranean diet oleic acid increases the number of two key cellular structures or organelles and protects cellular membranes from damage by a chemical reaction called oxidation this protective effect has a big payoff worms fed food rich in oleic acid lived about longer than those consigned to standard worm rations the researchers found oleic acid the researchers	0
2	almeno nei vermi da laboratorio	0
3	students whose families used the handbook reported their alcohol use over the past days had increased once they got to college compared to a increase among students whose parents didn t receive the book cannabis use went up for those control students but only for students whose families used the book families used whose families students whose the book used the	1
4	the study assessed the interaction between physical exertion and short term memory performance when distractors were present or absent in younger and older adults	0
5	punto cardine della causa è il fatto tuonano da greenpeace che per la prima volta avvenga l accertamento danni che un giudice riconosca le responsabilità in termini di crisi climatica del soggetto di diritto italiano eni oggi da considerare il maggiore responsabile di emissioni climalteranti in italia	0
6	in che modo mettere un minimo a k toglie potere ai sindacati di contrattare a k numeri a caso se non sono capaci di negoziare per i propri iscritti che chiudano a k	0
7	paper	0
8	source although it makes sense because it might narrow the gap between economies hard to believe polls in china russia even turkey	0
9	paper	0
10	our findings provide new insights into the computational mechanisms used by the human brain in perceptual judgments about the relation between ourselves and the external world	0
11	source	0
12	non ho capito se bannano gli ip che trasmettono o bloccano anche gli ip che ricevono perchè se vogliono bloccare anche chi guarda voglio vedere come fanno a chi è dietro nat gli bannano l intera famiglia condominio perchè un tizio solo stava guardando ip che gli ip	0
13	the detail numbers for every country	0
14	paper	0
15	paper	0
16	source	0

Figure 5.4: Result

5.2.7 Circle Graph

this graph shows the amount of the predicted posts of the user are depressed and not depressed



5.2.8 SQLite

We have used SQLite for making a record of the users that are authenticated to use the Web Applications. We have used SQLite because SQLite is a great choice for smaller projects or applications that require a lightweight, efficient, and easy-to-use database management system. Its integration with Django and compatibility with multiple operating systems make it an attractive option for developers looking for a reliable and efficient database management system.

5.3 Back End:

In the back-end the major work is being performed where the data is being crawled and analysis is being performed.

5.3.1 Scrapy(Scrapy, n.d.)

Scrapy is a Python-based open-source framework for web crawling that is used for extracting structured data from websites. Its functionality includes data mining, information processing, and automated testing. The framework works by sending HTTP requests to websites and parsing their HTML content to extract desired data. Scrapy provides developers with the ability to create powerful and flexible spiders that can navigate complex websites, follow links, and scrape data in various formats, such as JSON, CSV, and XML. Scrapy is highly scalable, speedy, and robust, which makes it a widely used tool for web crawling and scraping projects of various sizes by data scientists, researchers, and developers.

We have used Scrapy to extract data from Reddit Posts. Our Spider navigated through Reddit's web pages and extracted the relevant information which we required to perform the analysis. Then the data we extracted went for sentiment analysis.

5.3.2 Selenium (Selenium, n.d.)[\[2\]](#)

Selenium is an open-source, portable framework for automating web browsers. It provides a suite of tools for testing web applications, including a playback tool for authoring functional tests without the need to learn a test scripting language. Selenium can also be used for web scraping and automating repetitive tasks, such as data entry or form submission. Selenium supports a variety of programming languages, including Python, Java, Ruby, and C#. It can automate interactions with web browsers, including clicks, form submissions, and text inputs. Selenium also supports headless browser testing, which allows for testing without a visible user interface. Overall, Selenium is a powerful tool for automating web-related tasks and testing web applications.

In our project, we have used Selenium to automate the web browser to move to the Reddit Site and we have also used Selenium to scroll through the Reddit Posts to get the posts as much as we can then we made a local file of the scrolled webpage and pass it down to Scrapy using scrapy Request for data crawling.

5.3.3 FastAPI (FastAPI, n.d.)

[14] FastAPI is a modern, fast (high-performance), web framework for building APIs with Python 3.7+ based on standard Python type hints. It is designed to be easy to use and to provide high performance, leveraging the asynchronous programming capabilities of Python. Some of the key features of FastAPI are:

- FastAPI is one of the fastest web frameworks available, thanks to its use of asynchronous programming and the performance benefits it provides.
- With FastAPI, you can get started building APIs quickly and easily. It has a simple, intuitive API that makes it easy to write code.
- FastAPI uses Python type hints to define the structure of your API, which makes it easy to understand and maintain your code.
- FastAPI automatically generates OpenAPI and JSON Schema documentation for your API, making it easy to share and understand.

In our project, we have used FastAPI for the integration between back-end and front-end of our project. In our FastAPI the crawler is being executed and the sentiment analysis is being perform in the API. The use of FastAPI allowed us to quickly build a robust and scalable API that could handle a large amount of data. With FastAPI's support for asynchronous programming and Python's type hints, we were able to write efficient and readable code that could handle complex data structures.

Furthermore, FastAPI's automatic documentation generation feature proved to be extremely helpful in our project. Overall, the use of FastAPI in our depression analysis project allowed us to build a high-performance backend that could handle a large amount of data and integrate well with our front end built on Django. It also allowed us to easily document our API and write efficient code.

5.3.4 Dataset

The dataset selected for training the model was different Reddit posts related to Pakistan Communities and Depression related posts. The dataset was collected by using scrapy-selenium. through scrapy-selenium, we crawled through the Subreddits of the Pakistan Communities and Depression Communities. The Crawled data was then sent to the

database and the database we used is PostgreSQL. The dataset contains more than 20,000 text data. Text data contain the English language. We had to create our own dataset because there was less amount of datasets available online. we consulted professional psychologist for labelling our dataset.

	Post_Date text	Post [PK] text	Detail text	Img text	Link text
1	2 hours ago	'I would p...		None	https://w...
2	1 hour ago	'A welco...		None	https://w...
3	15 hours a...	'Absolutel...		None	https://w...
4	12 hours a...	'Abusive' ...		None	https://w...
5	1 day ago	'Adani Gr...		None	https://w...
6	1 day ago	'At its curr...		None	https://w...
7	1 day ago	'Atomic A...		None	None
8	2 days ago	'Better th...		None	https://w...
9	4 hours ago	'Bureaucr...		None	https://tri...
10	9 hours ago	'Centre sh...		None	https://w...
11	2 days ago	'Chop Off ...		None	https://ze...
12	1 day ago	'Class wa...		None	https://w...
13	3 days ago	'Complete...		None	https://w...
14	54 minute...	'Cooperat...		None	https://w...
15	14 hours a...	'Cow Kille...		None	https://w...
16	2 days ago	'Cow vigil...		None	https://w...
17	3 days ago	'Day of di...		None	https://w...
18	28 minute...	'Dear SBI ...		None	https://ec...
19	6 hours ago	'Devastati...		None	https://w...

Total rows: 1000 of 21120 Query complete 00:00:00.462

Figure 5.6: Training Dataset

5.3.4.1 Hurdles

The hurdles we faced in the training dataset are as follows:

- Took about 1-2 weeks to collect this amount of data.
- Dataset labeling of large amount of data.
- To label the data for depression analysis we required psychology specialists.
- Hurdle of finding number of psychology specialist for dataset labeling.
- It took about 4-5 weeks to get our dataset labelled.

5.3.4.2 Dataset Creation:

The dataset used in this study was labeled in collaboration with Dr. Arslan, a renowned psychiatrist, and with the assistance of the Islamabad Psychiatric Clinic and Rehabilitation Centre. Dr. Arslan's expertise and the clinic's resources provided valuable insights and guidance in the labeling process, ensuring the accuracy and reliability of the dataset used for analysis

5.3.4.3 Preprocessing

we have cleaned non English data to make it English dependant only. Provided exception handling for preventing adding duplicate data.

5.3.5 PostgreSQL

PostgreSQL, often simply called Postgres, is a powerful open source relational database management system (RDBMS) that uses and extends the SQL language. It is known for its robustness, scalability, and extensibility, making it a popular choice for enterprise-level applications and data-heavy workloads.

PostgreSQL was developed at the University of California, Berkeley in the 1980s and has since become one of the most advanced and feature-rich open source databases available. It is known for its reliability, with a focus on data integrity and transaction management.

Some of the key features of PostgreSQL include:

- PostgreSQL is fully ACID compliant, ensuring that transactions are processed in a reliable and consistent manner.
- PostgreSQL provides powerful full-text search capabilities, allowing users to search for specific words or phrases within text documents.
- PostgreSQL has built-in support for storing and querying JSON data.

5.3.6 Natural Language Processing[3]

NLP (Natural Language Processing) is a subfield of artificial intelligence and computational linguistics that focuses on the interaction between computers and human language. It involves the development of algorithms, models, and systems that enable computers to understand, interpret, and generate natural language text or speech.

NLP aims to bridge the gap between human language and computer language by providing computers with the ability to process and analyze human language in a meaningful way.

NLP utilizes various techniques from machine learning, deep learning, statistical modeling, and linguistic analysis to accomplish these tasks. It finds applications in various fields, including information retrieval, chatbots, virtual assistants, sentiment analysis, document summarization, machine translation, and many more.

5.3.6.1 Multilayer Perceptron (MLP, n.d.)[15]

A Multilayer Perceptron (MLP) is a type of artificial neural network (ANN) that consists of multiple layers of interconnected artificial neurons, or nodes. It is a feedforward neural network model, which means that the information flows in one direction, from the input layer through the hidden layers to the output layer, without any loops or feedback connections.

The MLP is composed of three main types of layers:

- **Input layer:** It represents the features or input variables of the problem. Each node in the input layer corresponds to a feature, and the values of these nodes are fed as inputs to the subsequent layers.
- **Hidden layers:** These are intermediate layers between the input and output layers. Each hidden layer consists of multiple nodes, and the nodes in one layer are fully connected to the nodes in the previous and next layers. The hidden layers are responsible for learning complex patterns and extracting relevant features from the input data.
- **Output layer:** It produces the final output of the network. The number of nodes in the output layer depends on the type of problem being solved. For example, in a binary classification problem, there will be a single node representing the probability of belonging to one class, while in a multi-class classification problem, there will be multiple nodes representing the probabilities for each class.

5.3.6.2 Latent Dirichlet Allocation (LDA) (LDA, n.d.)[16]

LDA is an unsupervised learning algorithm that discovers hidden or latent topics within a collection of documents. It assumes that each document is a mixture of various topics, and each topic is a distribution over words. The goal of LDA is to estimate the probability distributions that generate the observed documents.

The result of LDA is a set of topics, each characterized by a distribution of words. These topics can be interpreted and analyzed to gain insights into the underlying themes present in the document collection. LDA is often used for tasks such as document clustering, topic identification, and information retrieval in large text datasets.

5.3.6.3 Term Frequency-Inverse Document Frequency (TF-IDF) (TF-IDF, n.d.)[17]

TF-IDF (Term Frequency-Inverse Document Frequency) is a numerical statistic used to evaluate the importance of a term (word) in a document within a larger collection of documents, typically in the field of natural language processing and information retrieval.

The TF-IDF value for a term in a document is calculated based on two factors:

Term Frequency (TF):

- Term Frequency measures the frequency of a term within a document.
- It represents how often a term appears in a document relative to the total number of terms in that document.
- The intuition is that a term that appears more frequently in a document is more likely to be important to the document's content.

Inverse Document Frequency (IDF):

- Inverse Document Frequency measures the rarity or uniqueness of a term across the entire collection of documents.
- It is calculated by taking the logarithm of the ratio between the total number of documents and the number of documents containing the term.
- The IDF value is higher for terms that appear in fewer documents, suggesting that such terms are more discriminative or informative.

5.3.6.4 BIGRAM (BIGRAM, n.d.)[18]

In the context of natural language processing and text analysis, a bigram refers to a sequence of two adjacent words occurring together in a document or a sentence. It is a type of n-gram, where "n" represents the number of consecutive words considered as a unit.

5.3.6.5 Comparison Table

The table 5.1 shows the comparison of different Models that we used for the Analysis to get the best accuracy model.

Models	Accuracy	Precision	Recall	F1 Score
LDA+TF-IDF+Bigram with MLP	81.5%	83.1%	81.5%	81.5%
LDA+TF-IDF+Bigram with SVM	71.4%	71.4%	71.4%	71.3%
Linear Regression	66.5%	66.9%	66.5%	65.5%
ADA boost	80%	80%	81%	81%
Random Forest	80%	80%	80%	80%

Figure 5.7: Comparison Table

We have used MLP model because the test accuracy was better as compared to other models the test accuracy we got from MLP model was 81 percent , using SVM (SVM, n.d.) [19] we got 70 percent , Linear Regression (LR, n.d.) [20] accuracy was too low which was 34 percent and AdaBoost (AdaBoost, n.d.) [21] and RandomForest (Random Forest, n.d.) [22] gave the accuracy of approx. 80 percent.

5.4 Work Flow

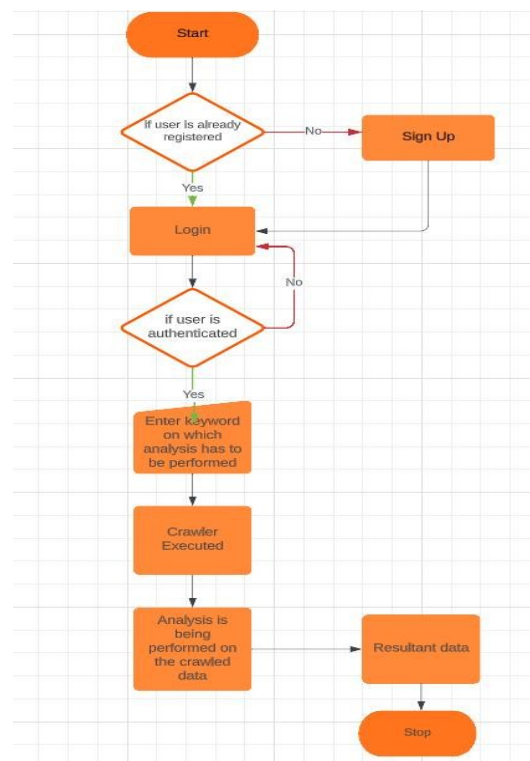
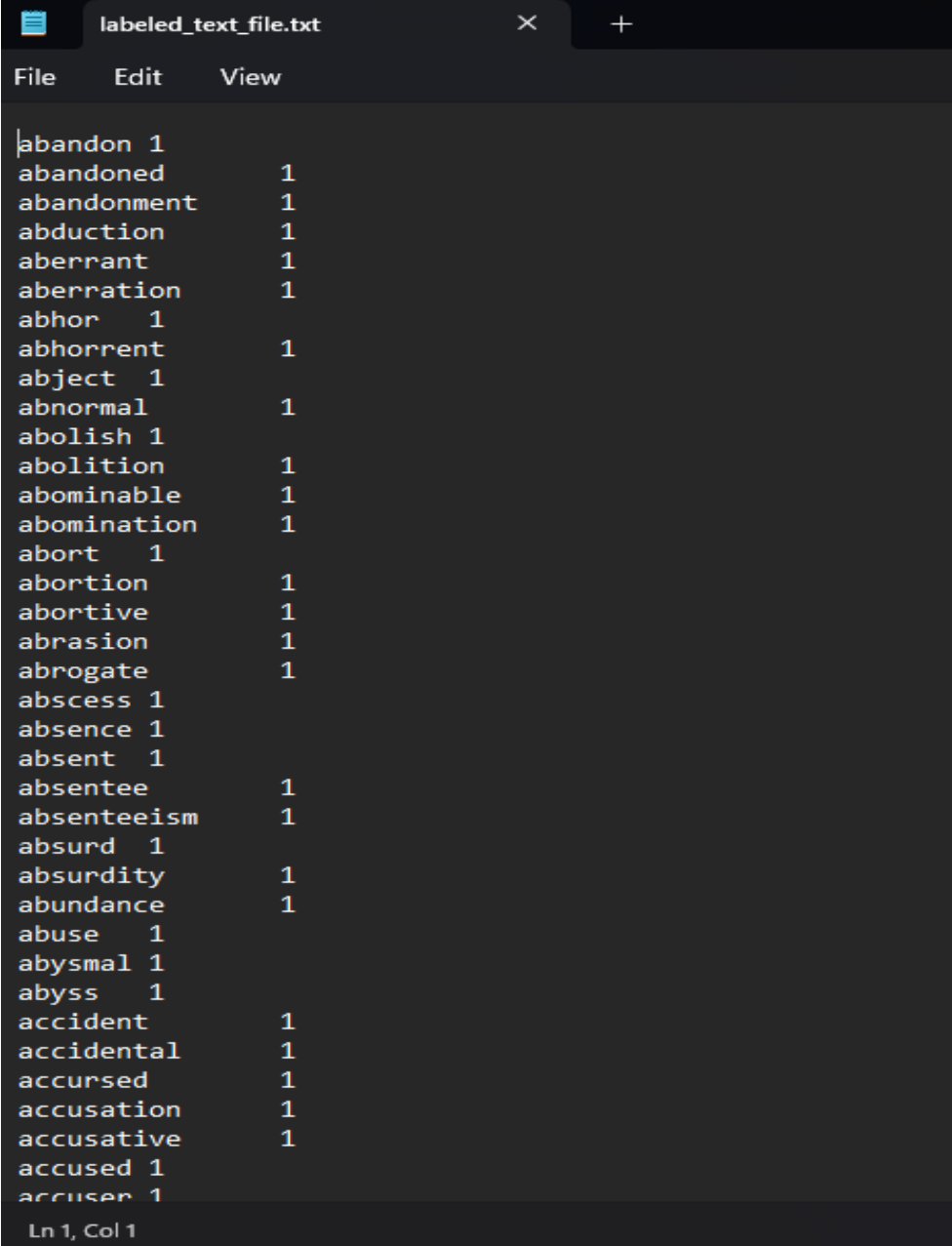


Figure 5.8: Flow Chart

5.5 Dictionary

We have created a dictionary of depressed words and it contains about 3000+ depressed words and we labelled them as 1 for depressed words because we know the words stored in the dictionary are depressed words



```
labeled_text_file.txt
File Edit View
|abandon 1
abandoned 1
abandonment 1
abduction 1
aberrant 1
aberration 1
abhor 1
abhorrent 1
abject 1
abnormal 1
abolish 1
abolition 1
abominable 1
abomination 1
abort 1
abortion 1
abortive 1
abrasion 1
abrogate 1
abscess 1
absence 1
absent 1
absentee 1
absenteeism 1
absurd 1
absurdity 1
abundance 1
abuse 1
abysmal 1
abyss 1
accident 1
accidental 1
accursed 1
accusation 1
accusative 1
accused 1
accuser 1
Ln 1, Col 1
```

Figure 5.9: Dictionary

The above image shows the sample of our dictionary's text file

We pass this dictionary to get them tokenized and during pre processing the data that is being analysed is passed to the pre processing section the posts get tokenized and the

words matching the dictionary words or related get the token 1 and other gets the token 0 for depressed and not depressed words.

5.5.1 Tokenized

The sample of our tokenized testing data is given below

```
[ 'we_0', 'understand_0', 'that_0', 'most_0', 'people_0', 'who_0', 'reply_0', 'immediately_1', 'to_0',
[ 'welcome_0', 'to_0', 'r_0', 'depression_1', 's_0', 'check_0', 'in_0', 'post_0', 'a_0', 'place_0',
[ 'anyone_0', 'else_0', 'instead_0', 'of_0', 'sleeping_0', 'more_0', 'when_0', 'depressed_1', 'stay_0',
[ 'i_0', 've_0', 'kind_0', 'of_0', 'stuffed_0', 'around_0', 'a_0', 'lot_0', 'in_0', 'my_0', 'life_0',
[ 'sleep_0', 'is_0', 'my_0', 'greatest_0', 'and_0', 'most_0', 'comforting_0', 'escape_1', 'whenever_0',
[ 'i_0', 'm_0', 'year_0', 'old_0', 'turning_0', 'soon_0', 'in_0', 'a_0', 'few_0', 'month_0', 'i_0',
[ 'i_0', 'live_0', 'alone_0', 'and_0', 'despite_0', 'me_0', 'being_0', 'prone_0', 'to_0', 'loneline_0',
[ 'i_0', 'm_0', 'not_0', 'looking_0', 'for_0', 'sympathy_1', 'just_0', 'simply_0', 'to_0', 'state_0',
[ 'i_0', 'don_0', 't_0', 'know_0', 'how_0', 'to_0', 'communicate_0', 'all_0', 'of_0', 'my_0', 'thou_0',
[ 'mom_0', 'i_0', 'm_0', 'sad_0', 'it_0', 'hurt_1', 'in_0', 'my_0', 'heart_0', 'the_0', 'feeling_1',
[ 'i_0', 've_0', 'been_0', 'struggling_0', 'with_0', 'depression_1', 'for_0', 'a_0', 'long_0', 'tin_0',
[ 'idk_0', 'how_0', 'to_0', 'elaborate_0', 'on_0', 'it_0', 'i_0', 'just_0', 'started_0', 'suddenly_0',
[ 'i_0', 'tried_0', 'to_0', 'help_0', 'his_0', 'family_0', 'abandoned_1', 'him_0', 'so_0', 'it_0',
[ 'to_0', 'me_0', 'it_0', 'seems_0', 'like_0', 'an_0', 'empty_0', 'meaningless_1', 'phrase_0', 'pec_0',
[ 'my_0', 'father_0', 'committed_0', 'suicide_1', 'day_0', 'before_0', 'my_0', 'th_0', 'birthday_0',
[ 'i_0', 'don_0', 't_0', 'think_0', 'i_0', 'have_0', 'the_0', 'ball_0', 'to_0', 'do_0', 'it_0', 'bu_0',
[ 'tw_0', 'suicide_1', 'yea_0', 'so_0', 'my_0', 'recent_0', 'symptom_1', 'of_0', 'depression_1', 'w_0',
```

Figure 5.10: Tokenized Data

5.6 LSTM

LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) architecture that is designed to overcome the limitations of traditional RNNs in capturing long-term dependencies in sequential data. It is particularly effective in tasks that involve processing and analyzing sequences of data, such as natural language processing and time series forecasting.

5.7 NGRAMS

NGRAMS refers to the contiguous sequence of n items (typically words or characters) within a given text. It is a technique commonly used in natural language processing and text analysis to capture the contextual information and relationships between adjacent elements. For example, consider the sentence: "I love to eat ice cream."

- When using unigrams ($n=1$), each word is treated as a separate unit: ["I", "love", "to", "eat", "ice", "cream"].
- When using bigrams ($n=2$), pairs of consecutive words are considered: ["I love", "love to", "to eat", "eat ice", "ice cream"].
- When using trigrams ($n=3$), triplets of consecutive words are considered: ["I love to", "love to eat", "to eat ice", "eat ice cream"].

In our project, the ngram range parameter is set in the TfidfVectorizer object to determine the range of n -grams to consider when creating the feature vectors from the text data. For example, `ngram range=(1, 3)` means that the vectorizer will consider unigrams, bigrams, and trigrams while creating the feature vectors. This allows the model to capture information from individual words as well as sequences of multiple words, potentially improving the model's ability to understand the context and meaning of the text.

The Result of using different ngram parameters are given below:

Ngrams	Accuracy	Precision	Recall	F1 Score
1-2	79.2%	79.1%	79.2%	79.1%
1-3	82.6%	82.4%	82.6%	82.1%
1-4	81%	80.7%	81%	80.8%

Figure 5.11: Ngram Parameters

Chapter 6

System Testing and Evaluation

This chapter presents the evaluations we carried out to test the functionality of the developed system. Different evaluations are described in the following sections.

6.1 Graphical user interface testing

In this section, we have tested the graphical user interface by the following steps which are define in Table 6.1

Table 6.1: GUI General and Forms Test Case

Test Case ID	Test Case 1	
Description	Test the Graphical User Interface	
Initial Condition	Equipment is set up as per requirements	
Steps	Task	Results
1.	Open the Web Application.	Pass
2.	Verify all the pages are working properly.	Pass
3.	Verify all Buttons are working properly.	Pass
4.	Verify Login system works properly.	Pass

6.2 Usability Testing

Usability testing helps in identifying the design and user interface-related issues in the system. The main aim of this testing is to identify the UI/UX-related issues in the system. In this test we

- Judge if users are able to complete tasks successfully

- Identify how long time it needs or takes to complete that task
- Find how satisfied users are with your application
- Note those changes needed to improve Ui/Ux
- And evaluate the performance to check if it meets your objectives or not

Table 6.2: Usability test case

Test Case ID	Test Case 2	
Description	Tests the usability of the system	
Initial Condition	Equipment is set up as per requirements	
Steps	Task	Results
1.	Open the Web Application.	Pass
2.	Verify that the buttons are relatable to the user.	Pass
3.	Verify that the user can easily navigate through different pages from the home page	Pass
4.	Verify that the user can easily complete all tasks	Pass

6.3 Software performance testing

In this we have tested the system speed, its responsiveness and the stability of the system. This gives us the highlights that where our system might fail.

Table 6.3: Performance test case

Test Case ID	Test Case 3	
Description	Tests the performance of the application.	
Initial Condition	Equipment is set up as per requirements	
Steps	Task	Results
1.	Open the Web Application.	Pass
2.	Verify that the web load time is minimum.	Pass
3.	Verify that the response time to user input is small.	Pass
4.	Verify that the application works under bad connection.	Fail

6.4 Security Testing

For the Security testing, we have provided the Login technique in which the data is being saved in the SQLite database

Table 6.4: Security test case

Test Case ID	Test Case 4	
Description	Tests the security of the application.	
Initial Condition	Equipment is set up as per requirements	
Steps	Task	Results
1.	Open the Web Application.	Pass
2.	Verify that data is in encrypted form.	Pass
3.	Verify that application is allowing only authorized users to access the information.	Pass

Here we have tested the installation phase of our system. As our system is a web application and by using the local IP address <http://127.0.0.1:8000/> we have tested that our GUI Running phase.

6.5 Confusion Matrix:

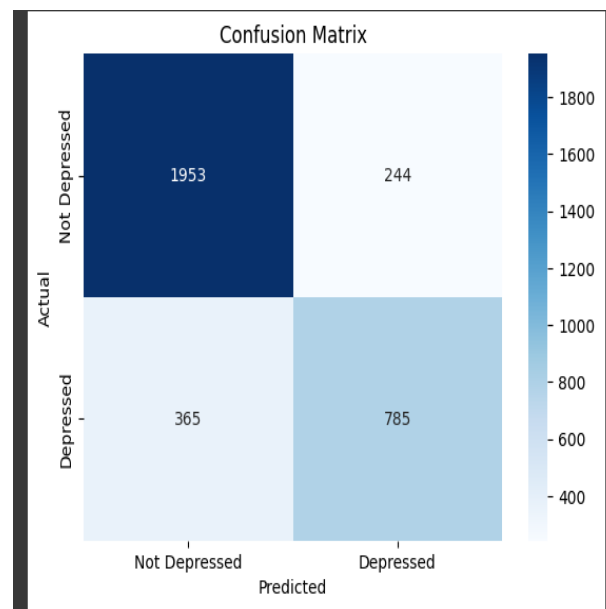


Figure 6.1: Confuse Matrix

6.5.1 Different Model Results:

This is the result of using different models for predictions:

6.5.1.1 Results:

Models	Accuracy	Precision	Recall	F1 Score
MLP	81%	83%	81%	79%
SVM	65%	42%	65%	51%
Random Forest	80.5%	80.3%	80.5%	79.7%
Linear Regression	75%	76%	75%	72%
LSTM	82.6%	82.4%	82.6%	82.1%

Table 6.5: Models

Chapter 7

Conclusions

This final year project aimed to analyze depression-related Reddit posts using web crawling techniques and Django for the front-end and FastAPI for the back-end. By collecting and analyzing a large amount of data from specific subreddits using Scrapy and Selenium, valuable insights were gained. Django provided a user-friendly interface, while FastAPI ensured efficient back-end handling. This project has the potential to impact mental health research by providing insights into the experiences of people with depression. The combination of web crawling, Django, and FastAPI streamlined analysis and presented data effectively, making it a successful project.

Our project demonstrates the power of web crawling techniques and modern web frameworks for analyzing large datasets of unstructured data, specifically focusing on depression-related language use on Reddit. This work serves as a foundation for further research, potentially exploring the relationship between depression-related language use and variables like demographics or location. By combining Reddit data with other sources, a more nuanced understanding of factors contributing to depression-related language use online can be achieved. Machine learning techniques can also be applied to identify subtle patterns or develop predictive models for identifying at-risk users.

In conclusion, this project contributes to mental health research by providing valuable insights into the experiences and challenges faced by individuals with depression. It has the potential to inspire further research and inform policy decisions promoting mental health and well-being.

Appendix A

User Manual

This user manual details how to use the Depression analysis Website:

- Start by making a virtual environment and installing all the libraries.
- If you have server access then assign 1 port to Backend and another port to Frontend.
- Else make a local host and run the FastAPI by using the command as given below
`uvicorn RedditFast:app --host 127.0.0.1 --port 8001`.
- Then make another local host and run the WebPage by using the command given below: `py manage.py runserver`.
- Also don't forget to activate venv before using these commands.
- After the page is loaded successfully open the webpage if you already have an account the sign in else sign up and then sign in.
- When you are in the Home page write the Correct UserID of the user you want to perform the analysis on.
- Wait till the process is completed.
- The final result will be shown in the form of table and in the form of a pie chart.

References

- [1] Django. <https://www.djangoproject.com/>, 2005. [Online; accessed 19-July-2008]. Cited on pp. v and 24.
- [2] selenium. <https://www.selenium.dev/>, Year Accessed. Cited on pp. v and 28.
- [3] Natural Language Processing. <https://www.ibm.com/topics/natural-language-processing>, Year Accessed. Cited on pp. v and 31.
- [4] reddit. <https://en.wikipedia.org/wiki/Reddit>, Year Accessed. Cited on p. 1.
- [5] scrapy-selenium. <https://pypi.org/project/scrapy-selenium/>, Year Accessed. Cited on p. 2.
- [6] Scrapy. <https://docs.scrapy.org/en/latest/intro/overview.html>, Year Accessed. Cited on p. 3.
- [7] PhantomBuster. <https://phantombuster.com/?deal=brock75>, 2008. [Online; accessed 19-July-2008]. Cited on p. 5.
- [8] Cited on p. 5.
- [9] Ahmed Hussein Orabi, Prasadith Buddhitha, Mahmoud Hussein Orabi, and Diana Inkpen. Deep learning for depression detection of twitter users. In *Proceedings of the fifth workshop on computational linguistics and clinical psychology: from keyboard to clinic*, pages 88–97, 2018. Cited on p. 6.
- [10] python. <https://www.python.org/shell/>, Year Accessed. Cited on p. 8.
- [11] PostgreSQL. <https://www.postgresql.org/>, Year Accessed. Cited on p. 8.
- [12] SQLite. <https://www.sqlite.org/index.html>, Year Accessed. Cited on p. 23.
- [13] pyCharm. <https://www.jetbrains.com/pycharm/>, Year Accessed. Cited on p. 23.
- [14] fastapi. <https://fastapi.tiangolo.com/>, 2018. [Online; accessed 19-July-2008]. Cited on p. 29.

- [15] MLP. https://en.wikipedia.org/wiki/Multilayer_perceptron, Year Accessed. Cited on p. 32.
- [16] Latent Dirichlet allocation. https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation, Year Accessed. Cited on p. 32.
- [17] TF-IDF. <https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/>, Year Accessed. Cited on p. 33.
- [18] Bigram. <https://en.wikipedia.org/wiki/Bigram>, 2010. [Online; accessed 19-July-2008]. Cited on p. 33.
- [19] SVM. <https://www.geeksforgeeks.org/introduction-to-support-vector-machines-svm/>, Year Accessed. Cited on p. 34.
- [20] Linear regression. https://en.wikipedia.org/wiki/Linear_regression, Year Accessed. Cited on p. 34.
- [21] Adaboost. <https://en.wikipedia.org/wiki/AdaBoost>, 1995. [Online; accessed 19-July-2008]. Cited on p. 34.
- [22] Random forest. https://en.wikipedia.org/wiki/Random_forest, Year Accessed. Cited on p. 34.