A REVIEW ON DETECTION AND ANALYSIS OF PSYCHOLOGICAL DISORDERS USING MACHINE LEARNING

[1]RAMADOSS.S, [2]DR. S. Maruthuperumal,
[3]K. KAVYA, [4]K.SAI THULASI,[5]CH. MOUNIKA, [6]M. KAVYA

[1]Assistant Professor, [2]Professor, [3,4,5,6] Students
CSE
BHARATH INSTITUTE OF HIGHER EDUCATION AND RESEARCH

***Abstract*- An increasing number of social network mental disorders (SNMDs), such as Cyber-Relationship Addiction, Information Overload, and Net Compulsion, have been recently noted. Symptoms of these mental disorders are usually observed passively today, resulting in delayed clinical intervention. In this paper, we argue that mining online social behavior provides an opportunity to actively identify SNMDs at an early stage. It is challenging to detect SNMDs because the mental factors considered in standard diagnostic criteria (questionnaire) cannot be observed from online social activity logs. Our approach, new and innovative to the practice of SNMD detection, does not rely on self-revealing of those mental factors via questionnaires. Instead, we propose a machine learning framework, namely, Social Network Mental Disorder Detection (SNMDD), that exploits features extracted from social network data to accurately identify potential cases of SNMDs. We also exploit multi-source learning in SNMDD and propose a new SNMDbased Tensor Model (STM) to improve the performance. Our framework is evaluated via a user study with 3126 online social network users. We conduct a feature analysis, and also apply SNMDD on large-scale datasets and analyze the characteristics of the three SNMD types. The results show that SNMDD is promising for identifying online social network users with potential SNMDs.**

OBJECTIVE

Social media data has been suggested to be a convenient and readily available source of such data. This is because social media platforms are in widespread use, users produce high volumes of data, regularly and spanning many years, and these data often contain rich personal and emotional information of putative relevance to their mental state. We used gold-standard machine learning methods with out-of-sample testing to establish the predictive power of models trained to predict depression and other aspects of mental health, using linguistic features derived from Tweets.

BASE PAPER

https://ieeexplore.ieee.org/document/9377199

INTRODUCTION

Social media website is defined as “a website that facilitates meeting people, finding like minds, communicating and sharing content, and building community”; this kind of website allows or encourages various types of activities, such as commercial, social, or a combination of the two. Social media categories include digital library, e-commerce, entertainment, forum, geolocation, social bookmark, social review, social game, and social network. Social network is the subcategory of social media, which is the social structure of people who are joined by common interest. Social media are social channels of communication using web-based technologies, desktop computers, and mobile technologies. These technologies create highly interactive platforms through which individuals, communities, and organizations can share information, discuss, rate, comment, and modify user-generated and online contents. These advancements enable communication among businesses, organizations, communities, and individuals. Social media technologies change the way individuals and large organizations communicate, and they are increasingly being developed.

Wide range of applications in business and public policy uses sentiment analysis. Sentimental analysis is now being used from specific product marketing to antisocial behaviour recognition. Businesses and organizations have always been concerned about how they are perceived by the public. This concern results from a variety of motivations, including marketing and public relations. Before the era of Internet, the only way for an organization to track its reputation in the media was to hire someone for the specific task of reading newspapers and manually compiling lists of positive, negative and neutral references to the organization, it could undertake expensive surveys of uncertain validity. Today, many newspapers are published online. Some of them publish dedicated online editions, while others publish the pages of their print edition in PDF. In addition to newspapers, there are a wide range of opinionated articles posted online in blogs and other social media. This opens up the possibility of automatically detecting positive or negative mentions of an organization in articles published online, thereby dramatically reducing the effort required to collect this type of information. To this end, Organizations are becoming increasingly interested in acquiring fine sentiment analysis from news articles. Fine-grained sentiment analysis is an extremely challenging problem because of the variety of ways in which opinions can be expressed. News articles present an even greater challenge, as they usually avoid overt indicators of attitudes. However, despite their apparent neutrality, news articles can still bear polarity if they describe events that are objectively positive or negative. Many techniques used for sentiment analysis involve naïve approaches based on spotting certain keywords which reveal the author or speaker's emotions. We use naïve performs fine-grained sentiment analysis to classify sentences as positive, negative or neutral.

LITERATURE REVIEW

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

A google wave-based fuzzy recommender system to disseminate information in University Digital Libraries 2.0

Nowadays Digital Libraries 2.0 are mainly based on the interaction between users through collaborative applications such as wikis, blogs, etc. or new possible paradigms like the *waves* proposed by Google. This new concept, the *wave*, represents a common space where resources and users can work together. The problem arises when the number of resources and users is high, then tools for assisting the users in their information needs are necessary. In this case a fuzzy linguistic recommender system based on the Google Wave capabilities is proposed as tool for communicating researchers interested in common research lines. The system allows the creation of a common space by means a *wave* as a way of collaborating and exchanging ideas between several researchers interested in the same topic. In addition, the system suggests, in an automatic way, several researchers and useful resources for each wave. These recommendations are computed following several previously defined preferences and characteristics by means of fuzzy linguistic labels. Thus the system facilitates the possible collaborations between multi-disciplinar researchers and recommends complementary resources useful for the interaction. In order to test the effectiveness of the proposed system, a prototype of the system has been developed and tested with several research groups from the same university achieving successful results.

A hybrid fuzzy-based personalized recommender system for telecom products/services

The Internet creates excellent opportunities for businesses to provide personalized online services to their customers. Recommender systems are designed to automatically generate personalized suggestions of products/services to customers. Because various uncertainties exist within both product and customer data, it is a challenge to achieve high recommendation accuracy. This study develops a hybrid recommendation approach which combines user-based and item-based collaborative filtering techniques with fuzzy set techniques and applies it to mobile product and service recommendation. It particularly implements the proposed approach in an intelligent recommender system software called Fuzzy-based Telecom Product Recommender System (FTCP-RS). Experimental results demonstrate the effectiveness of the proposed approach and the initial application shows that the FTCP-RS can effectively help customers to select the most suitable mobile products or services.

Recommender systems based on social networks

The traditional recommender systems, especially the collaborative filtering recommender systems, have been studied by many researchers in the past decade. However, they ignore the social relationships among users. In fact, these relationships can improve the accuracy of recommendation. In recent years, the study of social-based recommender systems has become an active research topic. In this paper, we propose a social regularization approach that incorporates social network information to benefit recommender systems. Both users' friendships and rating records (tags) are employed to predict the missing values (tags) in the user-item matrix. Especially, we use a biclustering algorithm to identify the most suitable group of friends for generating different final recommendations. Empirical analyses on real datasets show that the proposed approach achieves superior performance to existing approaches.

A Hybrid Trust-Based Recommender System for Online Communities of Practice

The needs for life-long learning and the rapid development of information technologies promote the development of various types of online Community of Practices. In online CoPs, bounded rationality and metacognition are two major issues, especially when learners face information overload and there is no knowledge authority within the learning environment. This study proposes a hybrid, trust-based recommender system to mitigate above learning issues in online CoPs. A case study was conducted using Stack Overflow data to test the recommender system. Important findings include: (1) comparing with other social community platforms, learners in online CoPs have stronger social relations and tend to interact with a smaller group of people only; (2) the hybrid algorithm can provide more accurate recommendations than celebrity-based and content-based algorithm and; (3) the proposed recommender system can facilitate the formation of personalized learning communities.

RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes
Recommender systems are used to provide filtered information from a large amount of elements. They provide personalized recommendations on products or services to users. The recommendations are intended to provide interesting elements to users. Recommender systems can be developed using different techniques and algorithms where the selection of these techniques depends on the area in which they will be applied. This paper proposes a recommender system in the leisure domain, specifically in the movie showtimes domain. The system proposed is called RecomMetz, and it is a context-aware mobile recommender system based on Semantic Web technologies. In detail, a domain ontology primarily serving a semantic similarity metric adjusted to the concept of "packages of single items" was developed in this research. In addition, location, crowd and time were considered as three different kinds of contextual information in RecomMetz. In a nutshell, RecomMetz has unique features: (1) the items to be recommended have a composite structure (movie theater + movie + showtime), (2) the integration of the time and crowd factors into a context-aware model, (3) the implementation of an ontology-based context modeling approach and (4) the development of a multi-platform native mobile user interface intended to leverage the hardware capabilities (sensors) of mobile devices. The evaluation results show the efficiency and effectiveness of the recommendation mechanism implemented by RecomMetz in both a cold-start scenario and a no cold-start scenario.

A novel hybrid approach improving effectiveness of recommender systems
Recommender systems support users by generating potentially interesting suggestions about relevant products and information. The increasing attention towards such tools is witnessed by both the great number of powerful and sophisticated recommender algorithms developed in recent years and their adoption in many popular Web platforms. However, performances of recommender systems can be affected by many critical issues as for instance, over-specialization, attribute selection and scalability. To mitigate some of such negative effects, a hybrid recommender system, called Relevance Based Recommender, is proposed in this paper. It exploits individual measures of perceived relevance computed by each user for each instance of interest and, to obtain a better precision, also by considering the analogous measures computed by the other users for the same instances. Some experiments show the advantages introduced by this recommender when generating potentially attractive suggestions.

A recommender system for researchers based on bibliometrics
Recommender systems (RSs) exploit past behaviors and user similarities to provide personalized recommendations. There are some precedents of usage in academic environments to assist users finding relevant information, based on assumptions about the characteristics of the items and users. Even if quality has already been taken into account as a property of items in previous works, it has never been given a key role in the re-ranking process for both items and users.
In this paper, we present REFORE, a quality-based fuzzy linguistic REcommender system FOr REsearchers. We propose the use of some bibliometric measures as the way to quantify the quality of both items and users without the interaction of experts as well as the use of 2-tuple linguistic approach to describe the linguistic information. The system takes into account the measured quality as the main factor for the re-ranking of the top-N recommendations list in order to point out researchers to the latest and the best papers in their research fileds. To prove the accuracy improvement, we conduct a study involving different recommendation approaches, aiming at measuring their performance gain. The results obtained proved to be satisfactory for the researchers from different departments who took part on the tests.

A peer-to-peer recommender system for self-emerging user communities based on gossip overlays
Gossip-based peer-to-peer protocols proved to be very efficient for supporting dynamic and complex information exchange among distributed peers. They are useful for building and maintaining the network topology itself as well as to support a pervasive diffusion of the information injected into the network. This is very useful in a world where there is a growing need to access and be aware of many types of distributed resources like Internet pages, shared files, online products, news and information. Finding flexible, scalable and efficient mechanisms addressing this topic is a key issue, even with relevant social and economic aspects. In this paper, we propose the general architecture of a system whose aim is to exploit the collaborative exchange of information between peers in order to build a system able to gather similar users and spread useful suggestions among them.

Social and Content Hybrid Image Recommender System for Mobile Social Networks
One of the advantages of social networks is the possibility to socialize and personalize the content created or shared by the users. In mobile social networks, where the devices have limited capabilities in terms of screen size and computing power, Multimedia Recommender Systems help to present the most relevant content to the users, depending on their tastes, relationships and profile. Previous recommender systems are not able to cope with the uncertainty of automated tagging and are knowledge domain dependant. In addition, the instantiation of a recommender in this domain should cope with problems arising from the collaborative filtering inherent nature (cold start, banana problem, large number of users to run, etc.). The solution presented in this paper addresses the abovementioned problems by proposing a hybrid image recommender system, which combines collaborative filtering (social techniques) with content-based techniques, leaving the user the liberty to give these processes a personal weight. It takes into account aesthetics and the formal characteristics of the images to overcome the problems of current techniques, improving the performance of existing systems to create a mobile social networks recommender with a high degree of adaptation to any kind of user.

A Personalized Recommender System Based on a Hybrid Model
Recommender systems are means for web personalization and tailoring the browsing experience to the users' specific needs. There are two categories of recommender systems; memory-based and model-based systems. In this paper we propose a personalized

recommender system for the next page prediction that is based on a hybrid model from both categories. The generalized patterns generated by a model based techniques are tailored to specific users by integrating user profiles generated from the traditional memory-based system's user-item matrix. The suggested system offered a significant improvement in prediction speed over traditional model-based usage mining systems, while also offering an average improvement in the system accuracy and system precision by 0.27% and 2.35%, respectively.

EXISTING SYSTEM

- E-commerce reviews reveal the customers' attitudes on the products, which are very helpful for customers to know other people's opinions on interested products. Meanwhile, producers are able to learn the public sentiment on their products being sold in E-commerce platforms.
- Generally, E-commerce reviews involve many aspects of products, e.g., appearance, quality, price, logistics, and so on.
- Therefore, sentiment analysis on E-commerce reviews has to cope with those different aspects.
- The problem with public auction is that the participation of the general public is very limited.

PROPOSED SYSTEM

In this paper, we aim to explore data mining techniques to detect three types of SNMDs

1) Cyber-Relationship (CR) Addiction, which includes addiction to social networking, checking and messaging to the point where social relationships to virtual and online friends become more important than real-life ones with friends and families

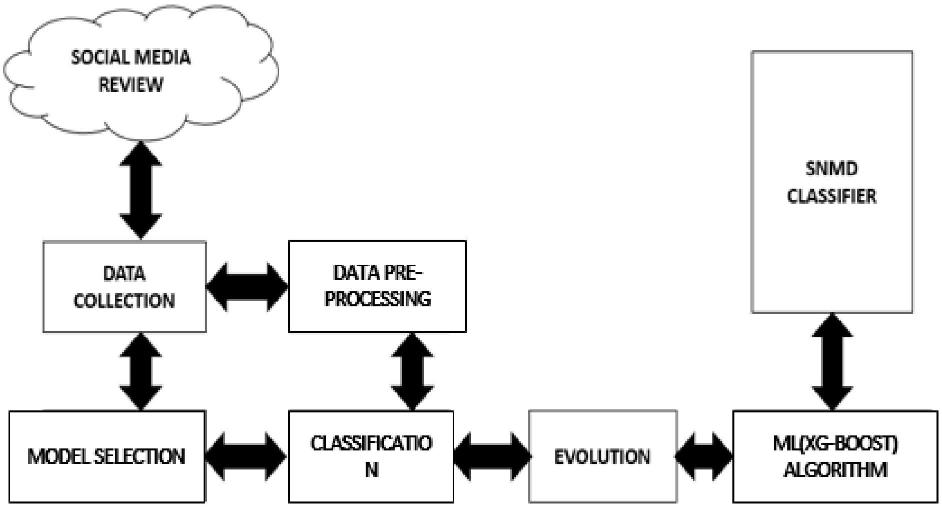
2) Net Compulsion (NC), which includes compulsive online social gaming or gambling, often resulting in financial and job-related problems.

3) Information Overload (IO), which includes addictive surfing of user status and news feeds, leading to lower work productivity and fewer social interactions with families and friends offline.

ADVANTAGES OF PROPOSED SYSTEM

- Collaborative Filtering
- Content Based Filtering
- Clustering
- Categorization

SYSTEM ARCHITECTURE



SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

- System : i3 Processor
- Hard Disk : 500 GB.
- Monitor : 15''LED
- Input Devices : Keyboard, Mouse
- Ram : 4GB.

SOFTWARE REQUIREMENTS

• Operating system	: Windows
• Coding Language	: Python
• IDE	: Pi-champ

SOFTWARE ENVIRONMENT

Python:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include −

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python https://www.python.org.

Windows Installation

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to https://www.python.org/downloads/.
- Follow the link for the Windows installer python-XYZ.msifile where XYZ is the version you need to install.
- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.

- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

First Python Program

Let us execute programs in different modes of programming.

Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt −

```
$ python
Python2.4.3(#1,Nov112010,13:34:43)
[GCC 4.1.220080704(RedHat4.1.2-48)] on linux2
Type"help","copyright","credits"or"license"for more information.
>>>
```

Type the following text at the Python prompt and press the Enter −

```
>>>print"Hello, Python!"
```

If you are running new version of Python, then you would need to use print statement with parenthesis as in **print ("Hello, Python!");**. However in Python version 2.4.3, this produces the following result −

```
Hello, Python!
```

Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension **.py**. Type the following source code in a test.py file −

```
print"Hello, Python!"
```

We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows −

```
$ python test.py
```

This produces the following result −

```
Hello, Python!
```

Flask Framework:

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

Http protocol is the foundation of data communication in world wide web. Different methods of data retrieval from specified URL are defined in this protocol.

The following table summarizes different http methods −

Sr. No	Methods & Description
1	**GET** Sends data in unencrypted form to the server. Most common method.
2	**HEAD** Same as GET, but without response body

3	**POST** Used to send HTML form data to server. Data received by POST method is not cached by server.
4	**PUT** Replaces all current representations of the target resource with the uploaded content.
5	**DELETE** Removes all current representations of the target resource given by a URL

By default, the Flask route responds to the **GET** requests. However, this preference can be altered by providing methods argument to **route()** decorator.

In order to demonstrate the use of **POST** method in URL routing, first let us create an HTML form and use the **POST** method to send form data to a URL.

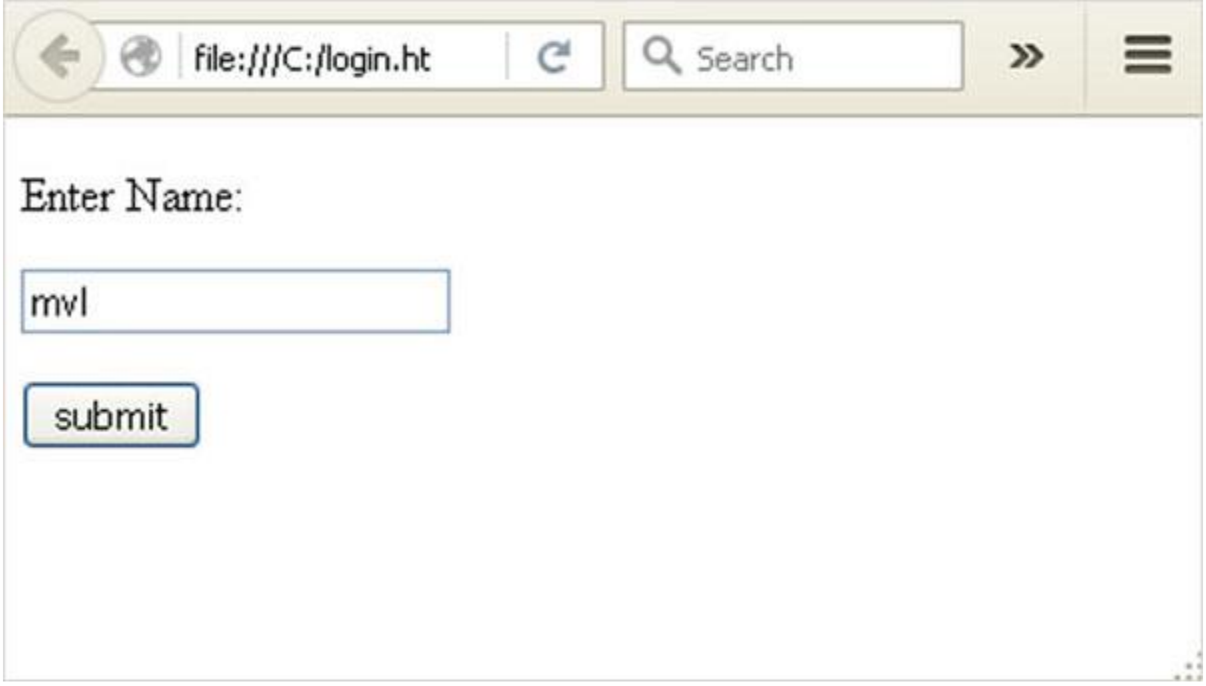
Save the following script as login.html

```
<html>
<body>
<formaction="http://localhost:5000/login"method="post">
<p>Enter Name:</p>
<p><inputtype="text"name="nm"/></p>
<p><inputtype="submit"value="submit"/></p>
</form>
</body>
</html>
```

Now enter the following script in Python shell.

```
from flask importFlask, redirect,url_for, request
app=Flask(__name__)
@app.route('/success/<name>')
def success(name):
return'welcome %s'% name
@app.route('/login',methods=['POST','GET'])
def login():
ifrequest.method=='POST':
user=request.form['nm']
return redirect(url_for('success',name= user))
else:
user=request.args.get('nm')
return redirect(url_for('success',name= user))
if __name__ =='__main__':
app.run(debug =True)
```

After the development server starts running, open **login.html** in the browser, enter name in the text field and click **Submit**.



Enter Name:

mvl

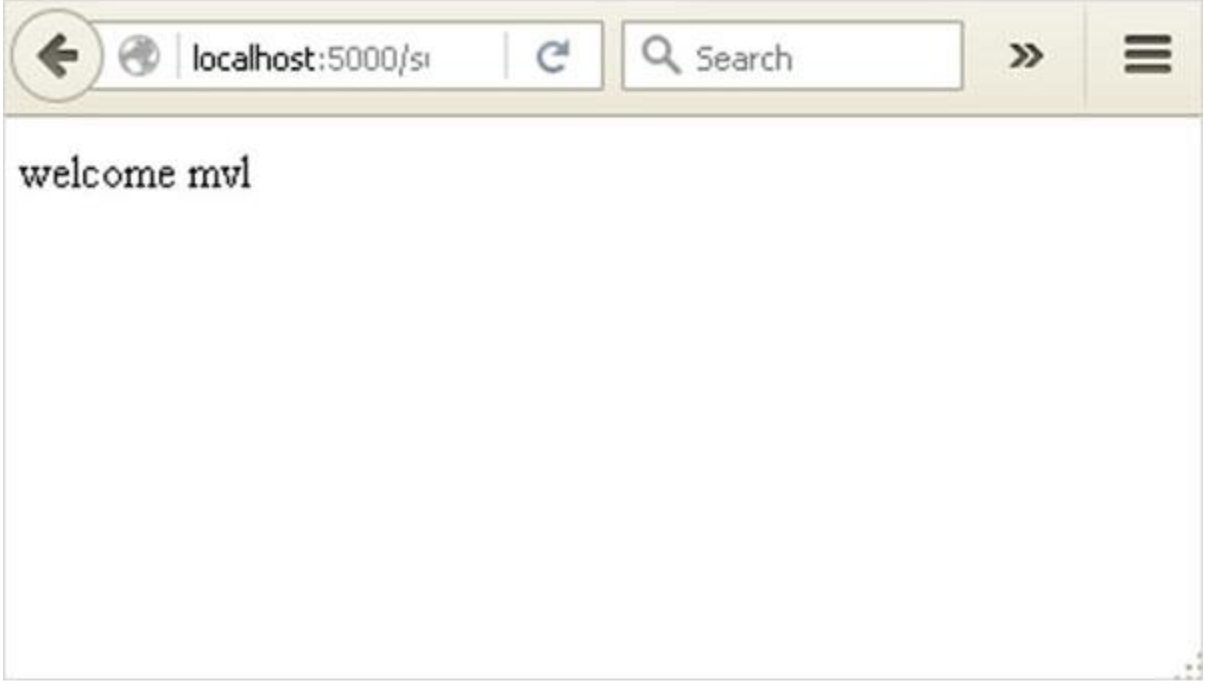
submit

Form data is POSTed to the URL in action clause of form tag.

http://localhost/login is mapped to the **login()** function. Since the server has received data by **POST** method, value of ‘nm’ parameter obtained from the form data is obtained by –

```
user = request.form['nm']
```

It is passed to **‘/success’** URL as variable part. The browser displays a **welcome** message in the window.



welcome mvl

Change the method parameter to **‘GET’** in **login.html** and open it again in the browser. The data received on server is by the **GET** method. The value of ‘nm’ parameter is now obtained by –

```
User = request.args.get(‘nm’)
```

Here, **args** is dictionary object containing a list of pairs of form parameter and its corresponding value. The value corresponding to ‘nm’ parameter is passed on to ‘/success’ URL as before.

What is Python?

Python is a popular programming language. It was created in 1991 by Guido van Rossum.
It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

- Python was designed to for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Python Install

Many PCs and Macs will have python already installed.
To check if you have python installed on a Windows PC, search in the start bar for Python or run the following on the Command Line (cmd.exe):

```
C:\Users\Your Name>python --version
```

To check if you have python installed on a Linux or Mac, then on linux open the command line or on Mac open the Terminal and type:

```
python --version
```

If you find that you do not have python installed on your computer, then you can download it for free from the following website: https://www.python.org/

Python Quickstart

Python is an interpreted programming language, this means that as a developer you write Python (.py) files in a text editor and then put those files into the python interpreter to be executed.
The way to run a python file is like this on the command line:

```
C:\Users\Your Name>python helloworld.py
```

Where "helloworld.py" is the name of your python file.
Let's write our first Python file, called helloworld.py, which can be done in any text editor.

```
helloworld.py
print("Hello, World!")
```

Simple as that. Save your file. Open your command line, navigate to the directory where you saved your file, and run:

```
C:\Users\Your Name>python helloworld.py
```

The output should read:

```
Hello, World!
```

Congratulations, you have written and executed your first Python program.

The Python Command Line

To test a short amount of code in python sometimes it is quickest and easiest not to write the code in a file. This is made possible because Python can be run as a command line itself.

Type the following on the Windows, Mac or Linux command line:

```
C:\Users\Your Name>python
```

From there you can write any python, including our hello world example from earlier in the tutorial:

```
C:\Users\Your Name>python
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, World!")
```

Which will write "Hello, World!" in the command line:

```
C:\Users\Your Name>python
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, World!")
Hello, World!
```

Whenever you are done in the python command line, you can simply type the following to quit the python command line interface:

```
exit()
```

Execute Python Syntax

As we learned in the previous page, Python syntax can be executed by writing directly in the Command Line:

```
>>> print("Hello, World!")
Hello, World!
```

Or by creating a python file on the server, using the .py file extension, and running it in the Command Line:

```
C:\Users\Your Name>python myfile.py
```

Python Indentations

Where in other programming languages the indentation in code is for readability only, in Python the indentation is very important.

Python uses indentation to indicate a block of code.

Example

```
if 5 > 2:
  print("Five is greater than two!")
```

Python will give you an error if you skip the indentation:

Example

```
if 5 > 2:
print("Five is greater than two!")
```

Comments

Python has commenting capability for the purpose of in-code documentation.

Comments start with a #, and Python will render the rest of the line as a comment:

Example

Comments in Python:

```
#This is a comment.
print("Hello, World!")
```

Docstrings

Python also has extended documentation capability, called docstrings.

Docstrings can be one line, or multiline.

Python uses triple quotes at the beginning and end of the docstring:

Example

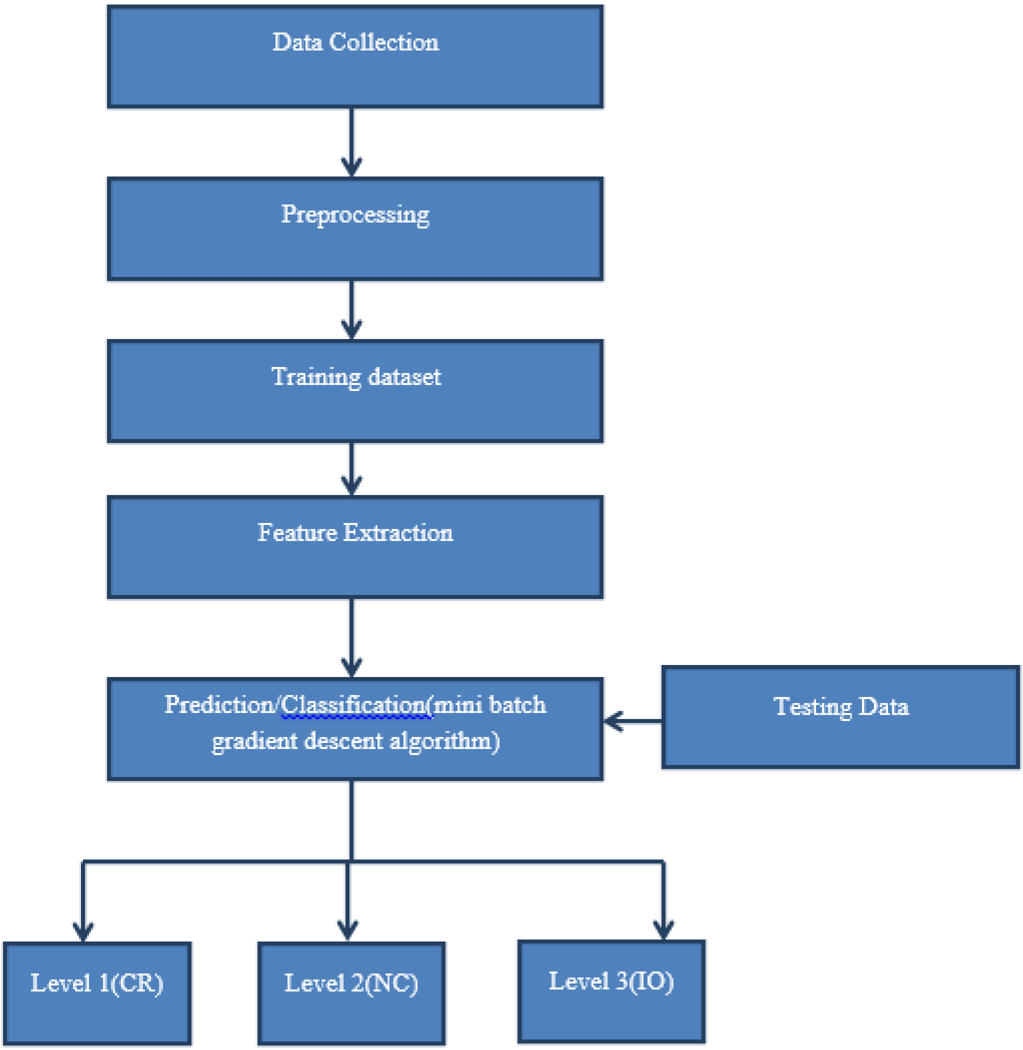
Docstrings are also comments:

```
"""This is a
multiline docstring."""
print("Hello, World!")
```

DATA FLOW DIAGRAM:

1. 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.
2. 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 by the process, an external entity that interacts with the system and the information flows in the system.
3. 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 the transformations that are applied as data moves from input to output.

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



UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling 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 Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling 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:

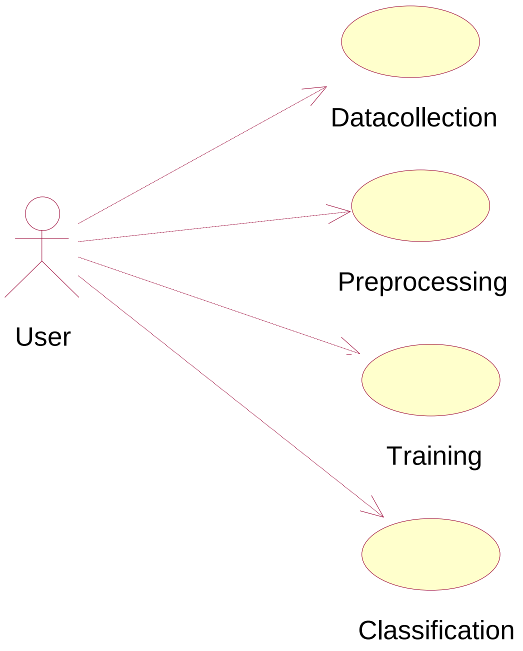
The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop 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 modeling language.
5. Encourage the growth of OO tools market.

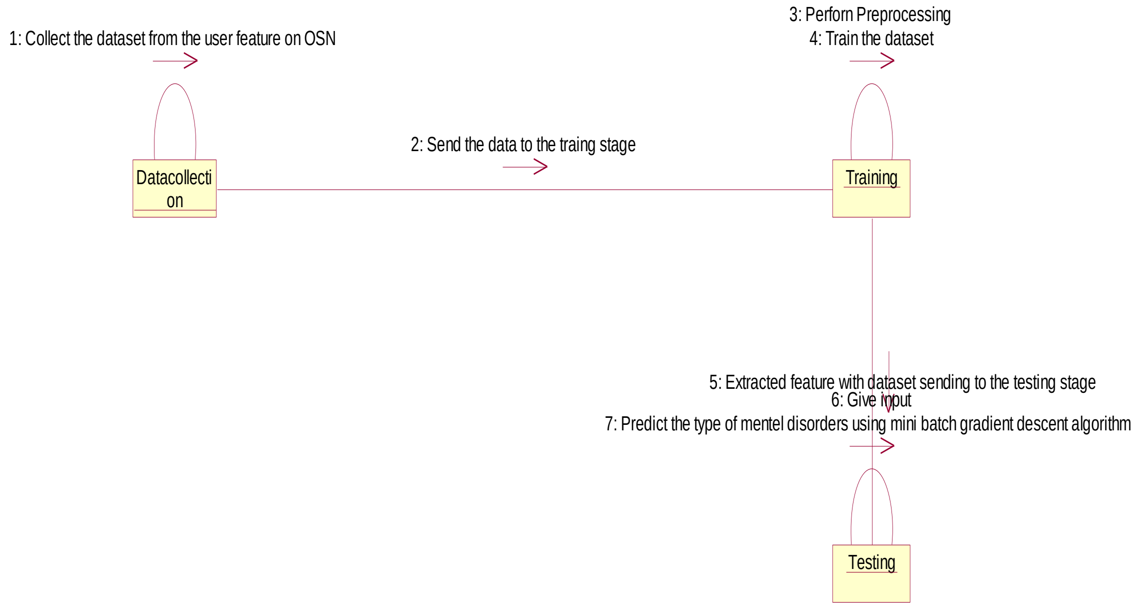
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

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 show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

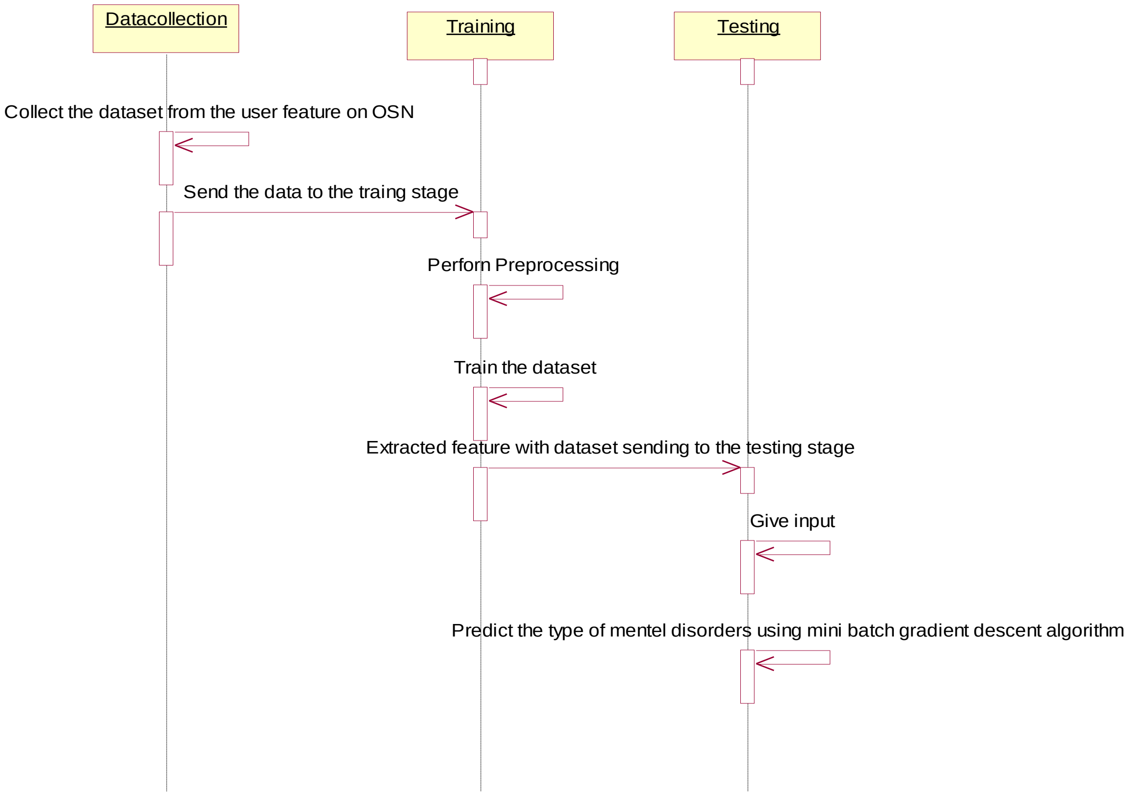


COLLABORATION DIAGRAM:


1: Collect the dataset from the user feature on OSN
3: Perforn Preprocessing
4: Train the dataset
2: Send the data to the traing stage
Datacollecti on
Training
5: Extracted feature with dataset sending to the testing stage
6: Give input
7: Predict the type of mentel disorders using mini batch gradient descent algorithm
Testing

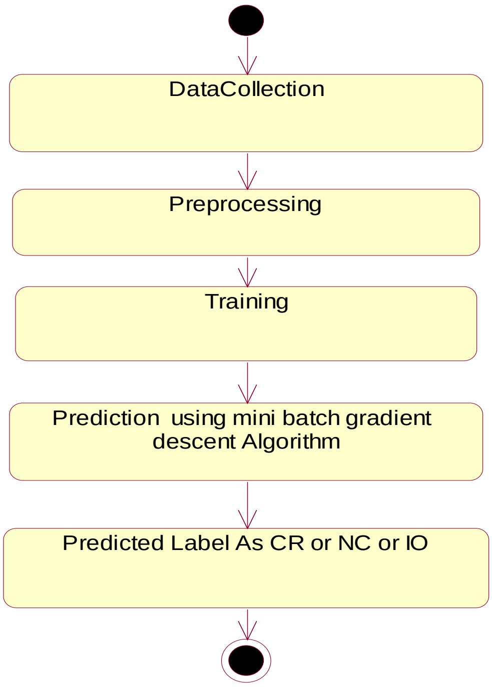
SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling 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.


Datacollection
Training
Testing
Collect the dataset from the user feature on OSN
Send the data to the traing stage
Perforn Preprocessing
Train the dataset
Extracted feature with dataset sending to the testing stage
Give input
Predict the type of mentel disorders using mini batch gradient descent algorithm

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.


DataCollection
Preprocessing
Training
Prediction using mini batch gradient descent Algorithm
Predicted Label As CR or NC or IO

INPUT DESIGN AND OUTPUT DESIGN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

➢ What data should be given as input?
➢ How the data should be arranged or coded?
➢ The dialog to guide the operating personnel in providing input.
➢ Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

❖ Convey information about past activities, current status or projections of the
❖ Future.
❖ Signal important events, opportunities, problems, or warnings.
❖ Trigger an action.
❖ Confirm an action.

SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth 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 ensure that 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
♦ SOCIAL FEASIBILITY

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 developed system 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.

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.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the
Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

6.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.3 Acceptance Testing

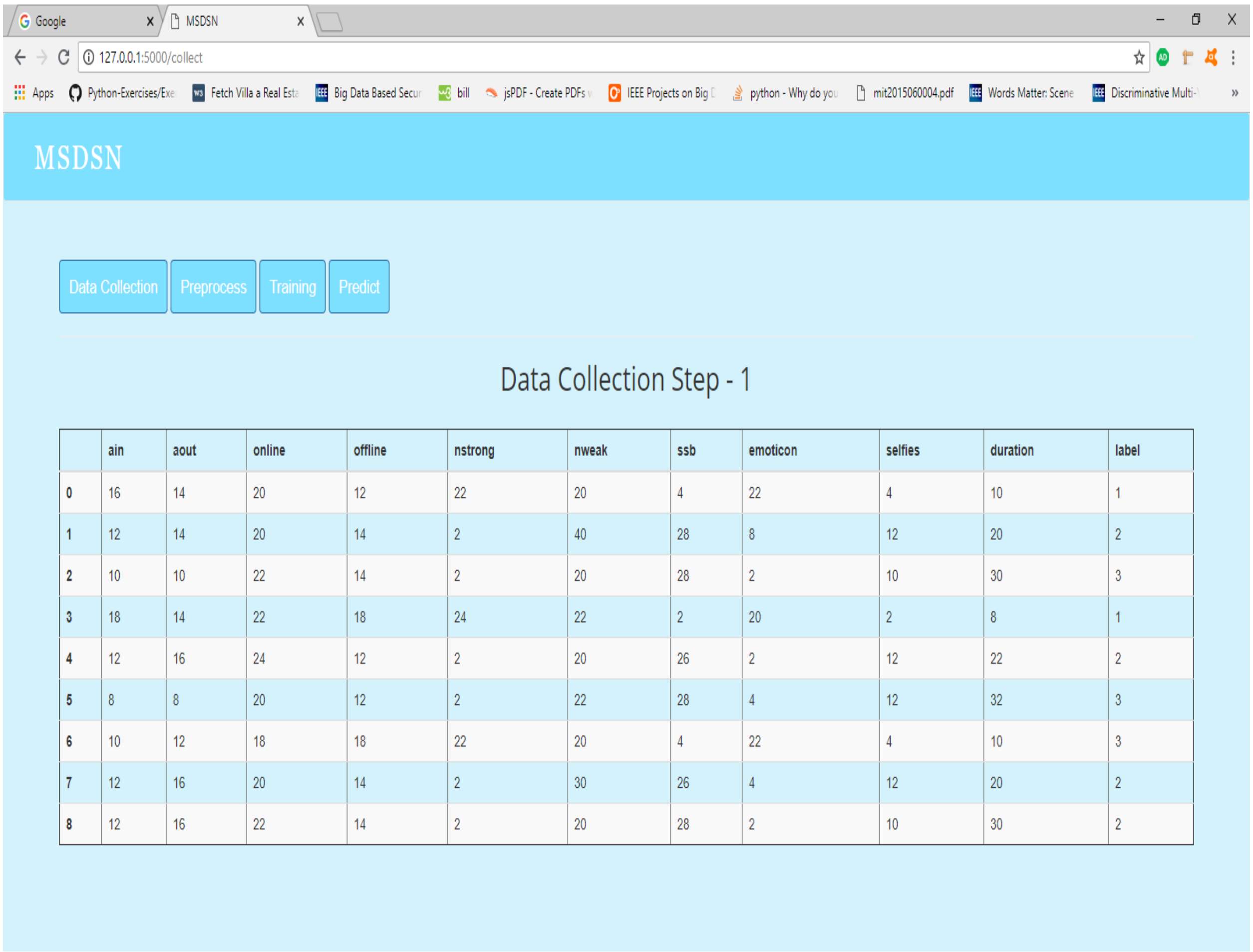
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

IMPLEMENTATION

Home:

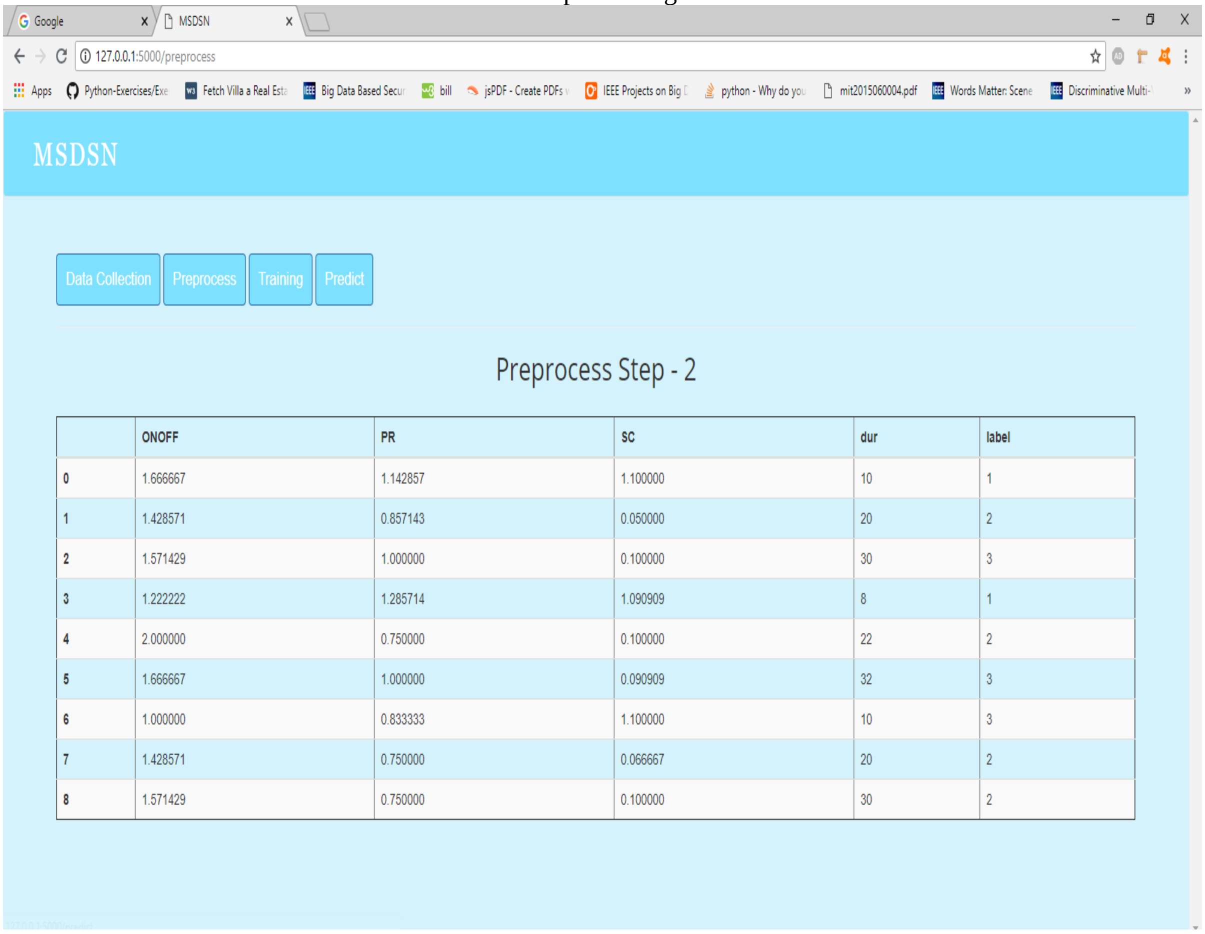
Dataset Collection:



Data Collection Step - 1

	ain	aout	online	offline	nstrong	nweak	ssb	emoticon	selfies	duration	label
0	16	14	20	12	22	20	4	22	4	10	1
1	12	14	20	14	2	40	28	8	12	20	2
2	10	10	22	14	2	20	28	2	10	30	3
3	18	14	22	18	24	22	2	20	2	8	1
4	12	16	24	12	2	20	26	2	12	22	2
5	8	8	20	12	2	22	28	4	12	32	3
6	10	12	18	18	22	20	4	22	4	10	3
7	12	16	20	14	2	30	26	4	12	20	2
8	12	16	22	14	2	20	28	2	10	30	2

Preprocessing

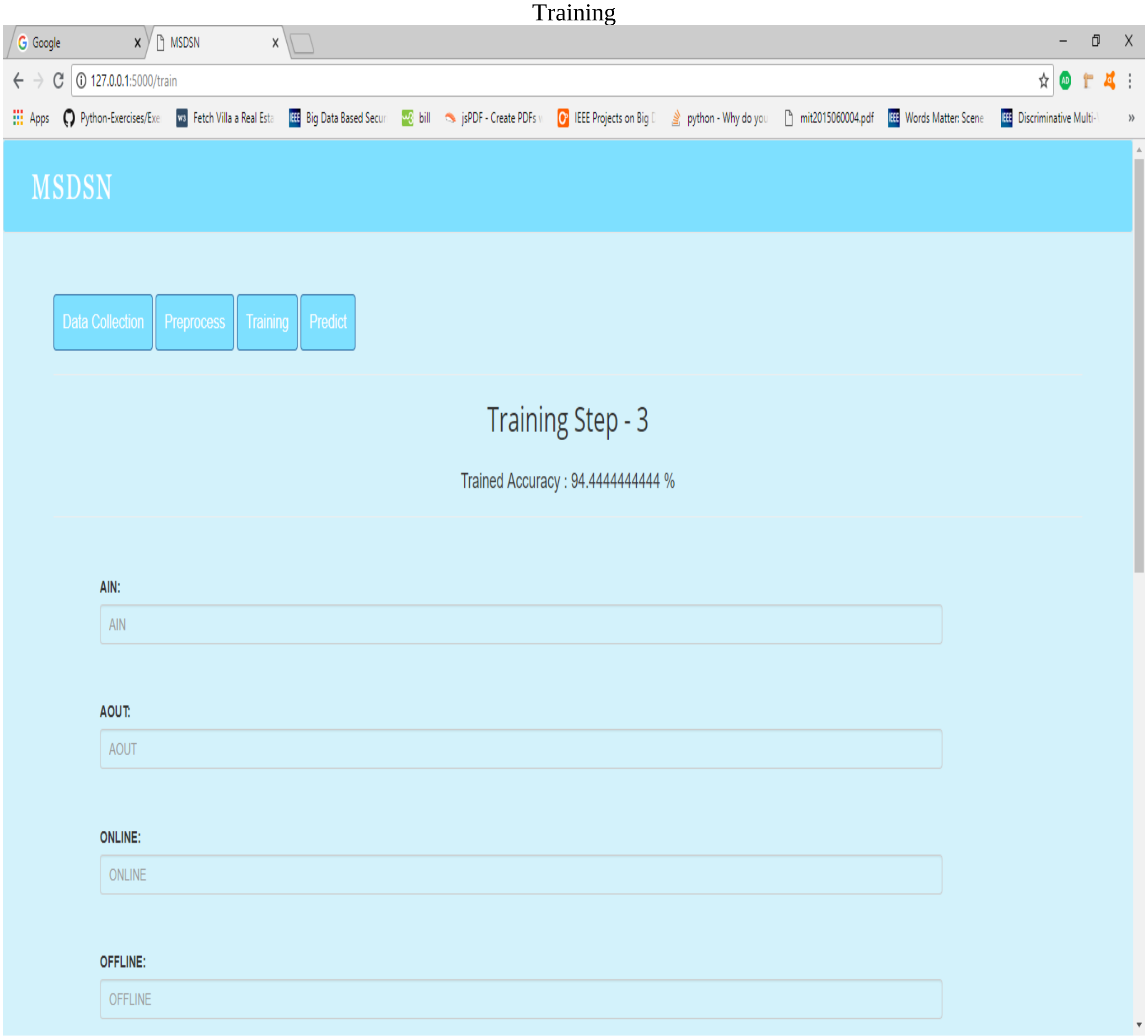


MSDSN

Preprocess Step - 2

	ONOFF	PR	SC	dur	label
0	1.666667	1.142857	1.100000	10	1
1	1.428571	0.857143	0.050000	20	2
2	1.571429	1.000000	0.100000	30	3
3	1.222222	1.285714	1.090909	8	1
4	2.000000	0.750000	0.100000	22	2
5	1.666667	1.000000	0.090909	32	3
6	1.000000	0.833333	1.100000	10	3
7	1.428571	0.750000	0.066667	20	2
8	1.571429	0.750000	0.100000	30	2

Training



Training Step - 3

Trained Accuracy : 94.4444444444 %

AIN: AIN

AOUT: AOUT

ONLINE: ONLINE

OFFLINE: OFFLINE

REFERENCES:

[1] K. Young, M. Pistner, J. O'Mara, and J. Buchanan. Cyber-disorders: The mental health concern for the new millennium. Cyberpsychol. Behav., 1999.

[2] J. Block. Issues of DSM-V: internet addiction. American Journal of Psychiatry, 2008.

[3] K. Young. Internet addiction: the emergence of a new clinical disorder, Cyberpsychol. Behav., 1998. [4] I.-H. Lin, C.-H. Ko, Y.-P. Chang, T.-L. Liu, P.-W. Wang, H.-C. Lin, M.-F. Huang, Y.-C. Yeh, W.-J. Chou, and C.-F. Yen. The association between suicidality and Internet addiction and activities in Taiwanese adolescents. Compr. Psychiat., 2014.

[5] Y. Baek, Y. Bae, and H. Jang. Social and parasocial relationships on social network sites and their differential relationships with users' psychological well-being. Cyberpsychol. Behav. Soc. Netw., 2013.

[6] D. La Barbera, F. La Paglia, and R. Valsavoia. Social network and addiction. Cyberpsychol. Behav., 2009.

[7] K. Chak and L. Leung. Shyness and locus of control as predictors of internet addiction and internet use. Cyberpsychol. Behav., 2004.

[8] K. Caballero and R. Akella. Dynamically modeling patients health state from electronic medical records: a time series approach. KDD, 2016.

[9] L. Zhao and J. Ye and F. Chen and C.-T. Lu and N. Ramakrishnan. Hierarchical Incomplete multi-source feature learning for Spatiotemporal Event Forecasting. KDD, 2016.

[10] E. Baumer, P. Adams, V. Khovanskaya, T. Liao, M. Smith, V. Sosik, and K. Williams. Limiting, leaving, and (re)lapsing: an exploration of Facebook non-use practices and experiences. CHI, 2013.