STRESS DETECTION IN SOCIAL INTERACTION USING MACHINE LEARNING

AN INDUSTRIAL INTERNSHIP REPORT

Submitted in partial fulfilment for the award of the degree of

M.Tech

in

Software Engineering

by

M.PAVITHRA-16MIS0452

School of Information Technology and Engineering

Department of Software and Systems Engineering

School of Information Technology and Engineering

Department of Software and Systems Engineering

DECLARATION BY THE CANDIDATE

I hereby declare that the Industrial Internship report entitled **"STRESS DETECTION IN SOCIAL INTERACTION USING MACHINE LEARNING"** submitted by me to VIT, Vellore, in partial fulfillment of the requirement for the award of the degree of **M.Tech (Software Engineering)** is a record of bonafide **Industrial Internship - SWE3099** carried out by me under the guidance of **Mr.KUMAR, Sr.Software developer**. I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree in this institute or any other institute or university.

Place: Vellore

Date:11-05-2020 **Signature of theCandidate**

School of Information Technology and Engineering

Department of Software and Systems Engineering

BONAFIDE CERTIFICATE

This is to certify that the Industrial Internship report entitled **"STRESS DETECTION IN SOCIAL INTERACTION USING MACHINE LEARNING"** by **M.PAVITHRA (16MIS0452)** to VIT, Vellore, in partial fulfillment of the requirement for the award of the degree of **M.Tech (Software Engineering)** is a record of bonafide work carried out by him /her under my guidance. The project fulfills the requirements as per the regulations of this Institute and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Dr. VIJAY ANAND.R
Signature of Internal Guide

Examiner(s) Signature

1.

ABSTRACT

Psychological wellness conditions influence a noteworthy level of the world's population every year. The stress investigation of emotional wellness phenomena in openly accessible social networking sites like twitter ,sinaweibo and facebook. A set of stress-related textual, visual, and social attributes from various aspects are first defined and then propose a novel hybrid model. The work has demonstrated the utility of online social information for contemplating despondency, be that as it may, there have been limited assessments of other mental well beingconditions.It is not easy to access the user posts on their facebook page. In order to obtain the user data from facebook, system have to get the access token from facebook developer page. The API act as an intermediate system that will help the system to analysis the user information from the facebook page. The system will also help to Recommending users with different links for psychological counseling centers, soft music or articles to help release their stress according to users' stress level

ACKNOWLEDGEMENT

I wish to express our heartfelt gratitude to **Dr.G.Viswanathan**, Chancellor, VIT, Vellore, for providing facilities for the Industrial Internship. I am highly grateful to our Vice President, **Dr.G. Sekar Viswanathan**, Vice chancellor **Dr. Anand A. Samuel,** and Pro-Vice Chancellor **Dr.S.Narayanan,** for providing the necessary resources.

My sincere gratitude to **Dr.B.K.Tripathy**, Dean, School of Information Technology and Engineering, for giving me the opportunity to undertake the project.

I wish to express my sincere gratitude to **Dr. S. SreeDharinya**, Head of the Department, Software and Systems Engineering, **Prof. Sujatha.R& Prof. Elango.N.M**, Industrial Internship Coordinators, M.Tech (Software Engineering), School of Information Technology and Engineering for providing me continuous support to do my projectwork**.**

I would like to express my special gratitude and thanks to my external guide **Mr.Kumar,** Sr.Software developer,Cloudgate systems India Pvt Ltd and internal guide **DR. Vijay Anand R ,** Assistant Professor Sr.Grade1, School of Information Technology and Engineering for their esteemed guidance, immense support and encouragement to complete the internship successfully.

I thank the management of VIT, Vellore for permitting me to use the library resources. I also thank all the faculty members of VIT, Vellore for giving me the courage and strength I needed to complete my goals. This acknowledgement would be incomplete without expressing my whole hearted thanks to my family and friends who motivated me during the course of the work.

Place: Vellore

Date:11-05-2010 M.PAVITHRA

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	4
	LIST OF SYMBOLS	8
	LIST OF FIGURES	8
	LIST OF ABBREVATION	8
1.	**INTRODUCTION**	9
	1.1.Problem Statement	10
	1.2.Motivation	10
	1.3.Objective	10
	1.3.1.Proposed System	10
	1.3.2.Advantages of Proposed system 1.3.2.1 Feasibility Study 1.3.2.2Technical Feasibility 1.3.2.3Economic Feasibility 1.3.2.4Social Feasibility 1.3.2.5Operational Feasibility	11 11 11 11 11 11
2.	**TECHNOLOGIES LEARNT**	12
3.	**SYSTEMDESIGN**	16
	3.1 System Architecture	16
	3.2 Module description	16
	3.3 System Specification	18
	3.3.1 Software Requirements	18
	3..3.2 Hardware Requirements	18
	3.4 DetailedDesign	19

	3.4.1 Use case Diagram	19
	3.4.2 Sequence Diagram	20
	3.4.3 Class Diagram	21
	3.4.4 Activity diagram	22
4.	**IMPLEMENTATION**	22
	4.1 Implementation details	22
5.	**TEST RESULTS**	25
	5.1 Test cases	25
	5.1.1 Unit Testing	25
	5.1.2 Integration Testing	25
6.	**RESULTS AND DISCUSSIONS**	27
7.	**CONCLUSION AND FUTURE WORK**	29
	7.1 Conclusion	29
	7.2 Future Work	30
8.	**REFERENCES**	30

LIST OF SYMBOLS

SYMBOLS	NAME	PAGE NO*
*	**Multiply**	**17**
/	**Divide**	**17**
=	**Validation**	**17**

LIST OF FIGURES

Figure No	Title	Page No
FIG1	System Architecture	16
FIG2	Decision Tree Classifier	18
FIG 3	Use case Diagram	19
FIG 4	Sequence Diagram	20
FIG 5	Class Diagram	21
FIG 6	Activity diagram	22

LIST OF ABBREVIATIONS

ACRONYM	EXPANSION
ML	Machine learning
GUI	Graphical User Interface
NUMpy	Numerical Python
NLKT	Natural Language Toolkit
UML	Unified Modeling Language
SOC	Sense of Coherenc

CHAPTER - 1

INTRODUCTION

More and more teenagers today are overloaded with adolescent stress from different aspects: academic future, self cognition, inter-personal, and affection. Long-lasting stress may lead to anxiety, withdrawal, aggression, or poor coping skills such as drug and alcohol use, threatening teenagers' health and development. Hence, it is important for both teenagers and their guardians/teachers to be aware of the stress in advance, and manage the stress before it becomes severe and starts causing health problems. The current social media micro-blog offers an open channel for us to timely and unobtrusively sense teenager's stress based on his/her tweeting contents and behaviors. This study describes a framework to further predict teenager's future adolescent stress level from micro-blog, and discusses how we address the challenges (data incompleteness and multifaceted prediction) using machine learning and multi-variant time series prediction techniques. Forthcoming events that may possibly influence teenager's stress levels are also incorporated into our prediction method. Our experimental results demonstrate the effectiveness of considering correlated features and event influence in prediction. To the best of our knowledge, this is the first work on predicting teenager's future stress level via micro-blog.College can be stressful for many freshmen as they cope with a variety of academic, personal, and social pressures . Although not all stress is negative, a certain level of stress can be beneficial to help improve performance. However, too much stress can adversely affect health in the annual survey of the American Freshman; the number of students reported feeling overwhelmed and stressed has increased steadily in the last decade. Over 50% of college students suffer significant levels of stress during a typical college semester

Consequently, there is a need to find innovative and costeffective strategies to help identify those students experiencing high levels of stress and negative emotions early on so that they can receive the appropriate treatment in order to prevent future mental illnesses.Social media use, such as Twitter and Facebook, has been rapidly growing, and research has already shown that data from these technologies can be used for novel approaches to public health surveillance . Twitter usage among young adults has increased 16% from 2012 to 2014. Currently, 32% of adults of the ages 18-29 years use Twitter, and the usage is expected to increase steadily in the future.People often have the need to share their emotions and experiences . Researchers have theorized that emotional sharing may fulfill a socio-affective need by eliciting attention, affection, and social support. Consequently, this may help individuals cope with their emotions and provide an immediate relief . Users often share their thoughts, feelings, and opinions on these social media platforms, and as a result, social media data may be used to provide real-time monitoring of stress and emotional state among college students . Previous studies have shown that Twitter data can be used to monitor a wide range of health outcomes, such as detecting human immunodeficiency virus infection outbreaks and predicting an individual’s risk of depression . For example, De Choudhury et al conducted one of the first studies that used an individual’s tweets to predict the risk of depression . The authors found that certain features extracted from a person’s tweets collected over a 1-year period were highly associated with the risk of depression in adults, such as raised negative sentiment in the tweets, frequent mentions of antidepressant medication, and greater expression of religious involvement. Currently, no studies have examined whether Twitter data can be used to monitor stress level and emotional state among college students. Studying this topic is important because the large amount of social media data from college students’ frequent use of social media can be used to help university officials and researchers

monitor and reduce stress among college students.

1.1ProblemStatement

Psychological stress is threatening people's health. It is non-trivial to detect stress timely for proactive care. With the popularity of social media, people are used to sharing their daily activities and interacting with friends on social media platforms, making it feasible to leverage online social network data for stress detection. In this paper, we find that users stress state is closely related to that of his/her friends in social media, and we employ a large-scale dataset from real-world social platforms to systematically study the correlation of users' stress states and social interactions.

1.2**Motivation**

With the development of living standards stress is increasingly common in people 's daily life.To detect how much stress they undergo in while they are using social media.machine learning can help people make a preliminary judgement about stress detection through decision tree and guassiannb.

1.3 Objective

It may be possible to detect stress through social network data. There are many ways to detect stress levels. Many researchers had tried to use machine learning techniques including the use of various algorithms such as Decision Tree, Naïve Bayes, Random Forest, etc. which gives a lower accuracy of 70% on average.

1.3.1ProposedSystem

1.The proposed work, we build a practical application to detect and release user's psychological stress by leveraging user's social media data in online social networks, and provide an interactive user interface to present users and friends psychological stress states.

2.With the given users online social media data as input, our proposed system intelligently and automatically detects users stress states.

3.Moreover, it will recommend users with different links to help release their stress.

4. The main technology of this project is a clustering model, which can leverage social media content and social interaction information for stress detection.

1.3.2Advantages of Proposedsystem

1.This will incredibly improve their productivity, streamline their activity line the board, and eventually upgrade the framework performance.

2. The proposed technique will identify the stress in social interactions effectively and accurately

Feasibility Study

1.3.2.1

A significant result of starter examination is the assurance that the framework solicitation is attainable. “The possibility of the venture is broke down in this stage and strategic agreement is advanced with a general arrangement for the task and some cost evaluations. During framework investigation the attainability investigation of the proposed framework is to be done. This is to guarantee that the proposed framework isn't a weight to the organization. For attainability investigation, some comprehension of the significant necessities for the framework is essential”. This is conceivable just in the event that it is practical inside restricted asset and time.The prime feasibility considerations are:

1.3.2.2 Technical feasibility:

This examination is done to check the specialized practicality, that is, the specialized necessities of the framework. “Any framework created must not have an intense interest on the accessible specialized assets. This will prompt levels of popularity on the accessible specialized assets. This will prompt levels of popularity being put on the customer. The created framework must have a humble necessity, as just negligible or invalid changes are required for actualizing this framework”.

1.3.2.3 Economical feasibility:

This investigation is done to check the financial effect that the framework will have on the association. “The measure of reserve that the organization can fill the innovative work of the framework is restricted. The uses must be defended. In this manner the created framework also inside the spending limit and this was accomplished in light of the fact that the vast majority of the innovations utilized are openly accessible. Just the modified items must be bought”.

1.3.2.4 Social feasibility:

The part of study is to check the degree of acknowledgment of the framework by the client. “This incorporates the way toward preparing the client to utilize the framework productively. The client must not feel undermined by the framework, rather should acknowledge it as a need. The degree of acknowledgment by the clients exclusively relies upon the strategies that are utilized to instruct the client about the framework and to make him acquainted with it. His degree of certainty must be raised with the goal that he is additionally ready to make some useful analysis, which is invited, as he is the last client of the framework”.

1.3.2.5 Operational feasibility:

This area of feasibility makes sure that the project satisfies all the requirements that were discusses during initial requirement analysis phase. Our project met all the requirements proposed earlier. It made sure that the modules discussed during the proposing of the system were implemented with stable accuracy and without any errors.

CHAPTER - 2

TECHNOLOGIES LEARNT

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freelydistributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Python Applications

Python is known for its general purpose nature that makes it applicable in almost each domain of software development. Python as a whole can be used in any sphere of development. Here, we are specifying applications areas where python can be applied.

1) Desktop GUIApplications

Python provides Tk GUI library to develop user interface in python based application. Some other useful toolkits wxWidgets, Kivy, pyqt that are useable on several platforms. The Kivy is popular for writing multitouch applications.

2) Scientific andNumeric

Python is popular and widely used in scientific and numeric computing. Some useful library and package are SciPy, Pandas, IPython etc. SciPy is group of packages of engineering, science and mathematics.

3) SoftwareDevelopment

Python is helpful for software development process. It works as a support language and can be used for build control and management, testing etc.

4) Enterprise Applications

Python can be used to create applications which can be used within an Enterprise or an Organization. Some real time applications are: OpenErp, Tryton, Picaloetc.

5) Console BasedApplication

We can use Python to develop console based applications. For example: IPython.

6) Audio or Video basedApplications

Python is awesome to perform multiple tasks and can be used to develop multimedia applications. Some of real applications are: TimPlayer, cplay etc.

7) 3D CADApplications

To create CAD application Fandango is a real application which provides full features of CAD.

8) Applications forImages

Using Python several application can be developed for image. Applications developed are: VPython, Gogh, imgSeek etc.

Machine learning is nothing but using data to make a machine to make intelligent decisions. It is based on recognizing and learning through patterns in data. Intelligent algorithms are then built by extracting, processing, defining, cleaning, arranging and then understanding the data. To perform such tedious tasks, why should one use Python? The answer is simple – It is easy to understand! Adding Python to the implementation process has helped engineers to validate various ideas.

Why Python for Machine Learning :

1. Simple andconsistent

Python's simple syntax allows developers to write codes that are reliable, concise and readable. This allows them to only struggle with the solution of the problem and not the code syntax, increasing the productivity of the overall process of development. This simplicity becomes appealing to other developers and urges them to learn python. It's greater human understanding makes it easier for coders/developers to invent various functionalmodels.

Multiple developments and Collaborative implementation type of projects majorly use Python as the language for code. Prototypes are built faster as the complex machine learning tasks and the testing process can be done quickly. While we find the reason behind greater functionality- frameworks, libraries, and extensions are the reasons that come back as responses.Now, another reason to learn Python for machine learning is the amazing libraries and frameworks of Python.

2. Libraries andFrameworks

Libraries Used:

NumPy:
NumPy stands for 'Numerical-Python' or 'Numeric-Python'. It is often pronounced as num-pai or num-pee. It is a library for the Python programming language mainly used for interacting with multidimensional and large arrays and matrices. It has a large number of built in mathematical functions which can be used on the large arrays for analyzing them. It is a very good package that can be used on any mathematical and statistics related operation. It works well with multi-dimensional array, data science programming, linear-algebra, random number problems, Fourier-transform etc. This library is more efficient in terms of memory and can manage large amount of data than any other library.

Pandas:

Pandas is also a library used for dealing with data in data science. Apart from Numpy which deals with multi-dimensional arrays, Pandas deals with data in tabular form i.e. data in the form of rows and columns. In this library a 1D array called as Series, a 2D-table called as DataFrames and the

combination of DataFrames called as Panel objects are used. We can edit the Series by adding or deleting elements into the array like objects. We can even add, delete or edit the rows and columns in the DataFrame. In similar lines with Numpy, Pandas also provides the primary mathematical operations such as adding subtracting, conditional-operations and even broadcasting.

NLTK:

NLTK is the abbreviation for Natural Language Toolkit. It is often a platform that is used to build the python programs that function with data represented in human language. It is mainly used for Natural Language Processing. It mainly comprises of libraries used in tokenizing, classifying, parsing, stemming, tagging, lemmatization, punctuation and also semantic reasoning. It consists of a large number (>50) of corpora and the lexical sources. It is available free as it an open source and community driven project.

Tensorflow:

Tensorflow is a framework for generating ML models. It is an open source platform mainly used in Machine Learning. The word Tensor means the n-dimensional array which we give as input and then we develop a flow of operations in between the input and the output. The input we give goes through all these operations and produces an output at the other end. So we call this as Tensorflow. The Tensorflow architecture comprises of mainly three parts. They are a) Preprocess the data, 2) Building the model and 3) Training the model and then estimating it.

3. PlatformIndependence

It basically means one can freely shift from one machine to another without making changes to the actual code. This functionality is allowed by python's framework. This is also one of the keys to the popularity of the python language. It supports many platforms such as Windows, macOS, and Linux. Most of the companies use their own machine containing powerful GPUs to train their machine learning models. Python being platform-independent can make this overall training very cheap and easier.

CHAPTER - 3

SYSTEM DESIGN

3.1 System Architecture

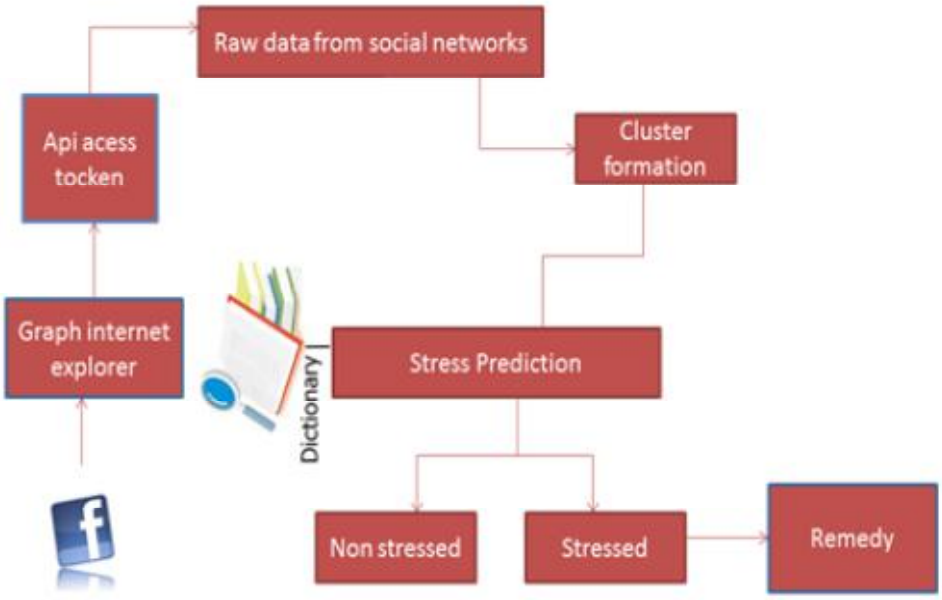


Fig1

3.2 Module description

Dataset :

A data set is a collection of related, discrete items of related data that may be accessed individually or in combination or managed as a whole entity. A data set is organized into some type of data structure in a database. The database itself can be considered a data set, as can bodies of data within it related to a particular type of information, such as sales data for a particular corporate department.

Data Processing :

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent,

and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing. Data preprocessing is used database-driven applications such as customer relationship management and rule-basedapplications.

Data goes through a series of steps during preprocessing:

Data Cleaning: Data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data.

Data Integration: Data with different representations are put together and conflicts within the data are resolved.

Data Transformation: Data is normalized, aggregated and generalized.

Data Reduction: This step aims to present a reduced representation of the data in a data warehouse.

Data Discretization: Involves the reduction of a number of values of a continuous attribute by dividing the range of attribute intervals.

Algorithm Implementation:

A calculation in information mining is a lot of heuristics and estimations that makes a model from information. To make a model, the calculation initially investigates the information you give, searching for explicit sorts of examples or patterns. The calculation utilizes the aftereffects of this investigation over numerous emphases to locate the ideal parameters for making the mining model. These parameters are then applied over the whole informational index to remove noteworthy examples and natty gritty measurements. In this venture we are utilizing AI calculation called Naive Bayes. Guileless Bayes classifie area**Moduledescription** group of straightforward "probabilistic classifiers" in view of applying Bayes' hypothesis with solid (gullible) autonomy presumptions between the highlights.

Naive Bayes Classifier:

Credulous Bayes classifier accepts that the impact of a specific element in a class is free of different highlights. For instance, an advance candidate is attractive or not relying upon his/her pay, past credit and exchange history, age, and area. Regardless of whether these highlights are associated, these highlights are as yet considered freely. This supposition improves calculation, and that is the reason it is considered as credulous. This supposition that is called class contingent autonomy.

$$P(A/B)=P(B/A)*P(A)/P(B)$$

A classifier is an ML model that is utilized to separate various articles dependent on specific highlights. A Naive Bayes classifier is a probabilistic ML model that is utilized for order task. The core of the classifier depends on the Bayes hypothesis.

Utilizing Bayes hypothesis, we can discover the likelihood of An event, given that B has happened. Here, B is the proof and An is the theory. The suspicion made here is that the indicators/highlights are free. That is nearness of one specific component doesn't influence the other. Henceforth it is called innocent.

Decision Tree Classifier:

Choice Tree is a white box kind of ML calculation. It shares inner basic leadership rationale, which isn't accessible operating at a profit box kind of calculations, for example, Neural Network. Its preparation time is quicker contrasted with the neural system calculation. The time intricacy of choice trees is an element of the quantity of records and number of characteristics in the given information. The choice tree is a dispersion free or non-parametric technique, which doesn't rely on likelihood conveyance suspicions. Choice trees can deal with high dimensional information with great precision.

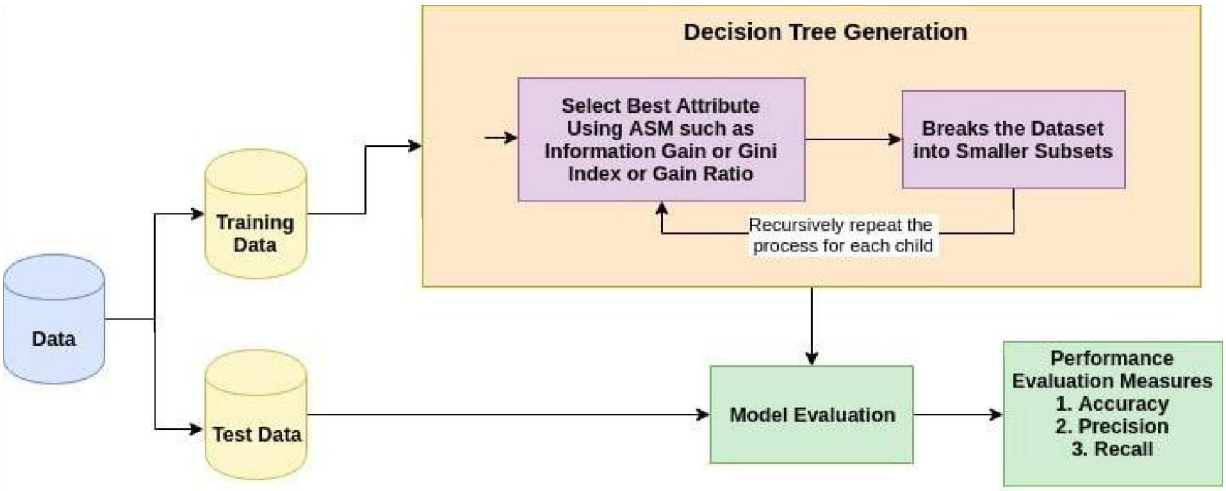


FIG2

3.3 SystemSpecification

3.3.1 Hardware Requirements

- **Processor:** AMD PRO A4-4350B R4,5 COMPUTE CORES 2C+3G 2.50GHZ
- **HardDisk:** Min 160 GB
- **Ram:** Min 4GB

3.3.2 Software requirements:

windows operating System version-1803

Python

Anaconda-Anaconda3-2020.02-Windows-x86_64

Launch jupyter(python3)

3.4DetailedDesign

3.4.1 Use caseDiagram

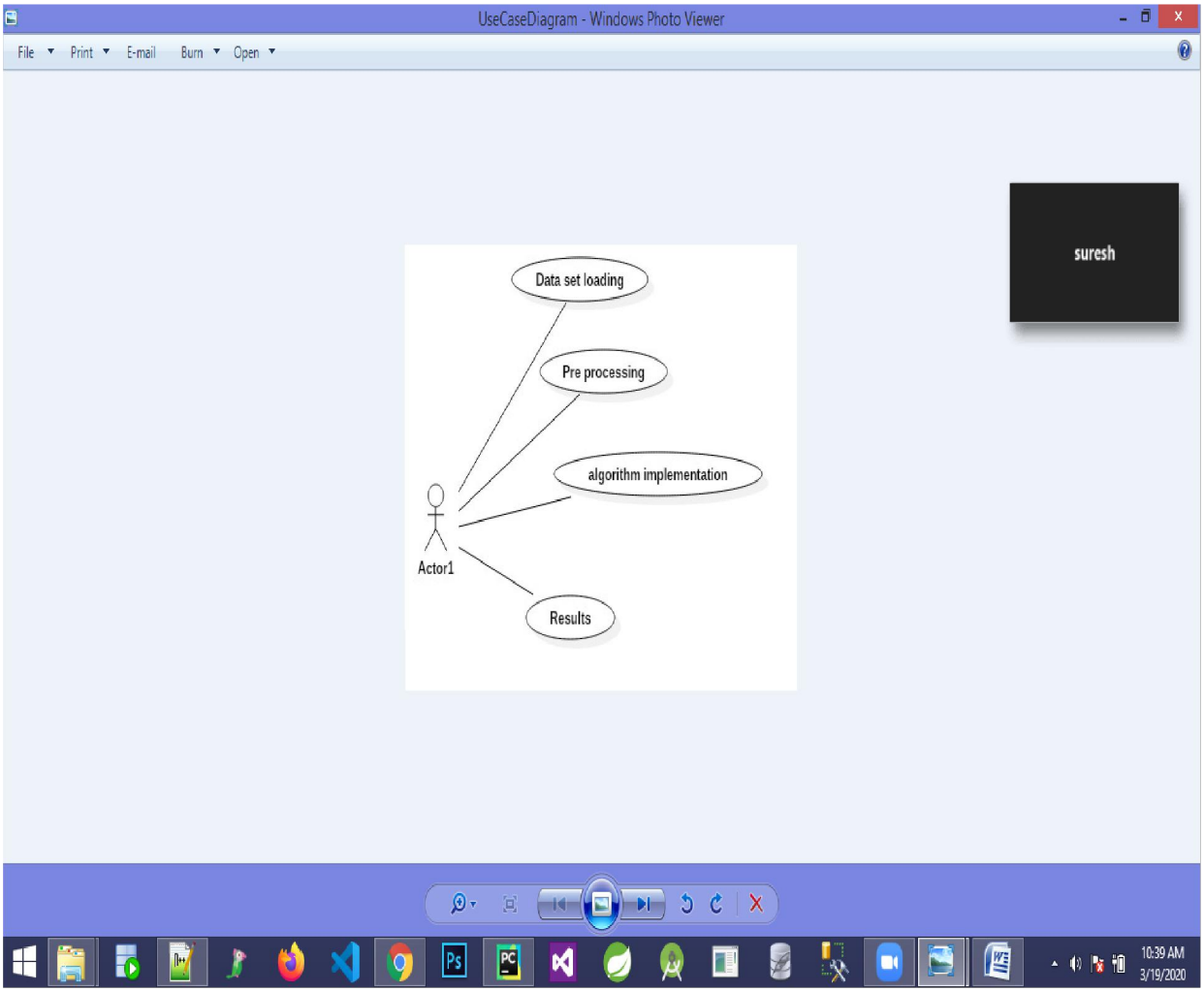


FIG3

3.4.2 Sequence diagram

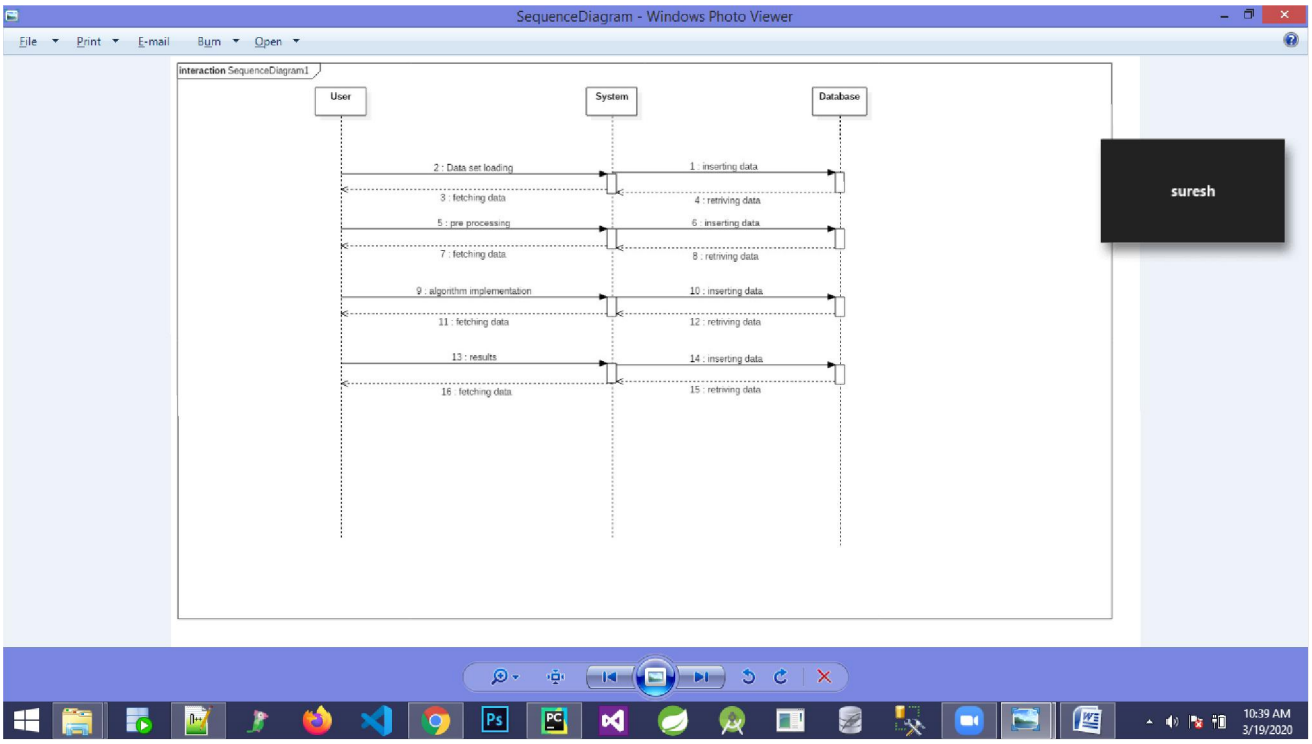


FIG4

3.4.3Classdiagram

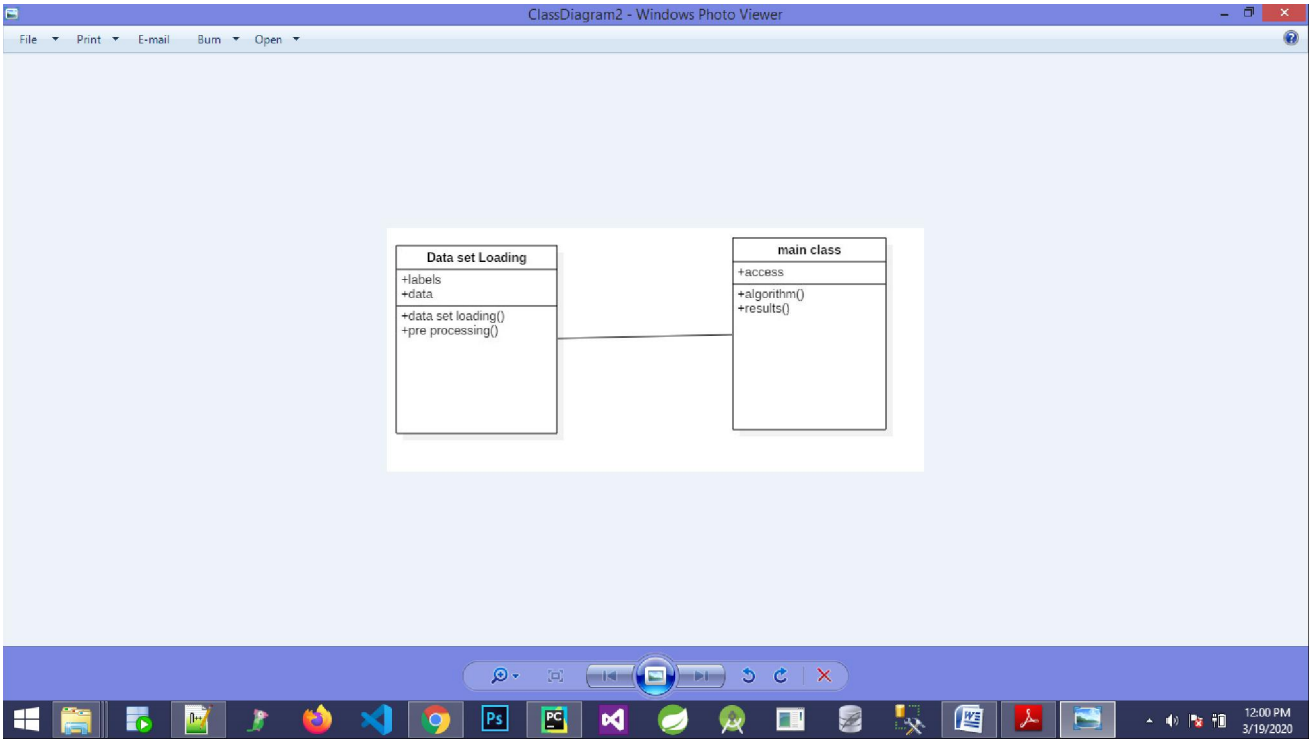


FIG5

3.4.4 Activity diagram

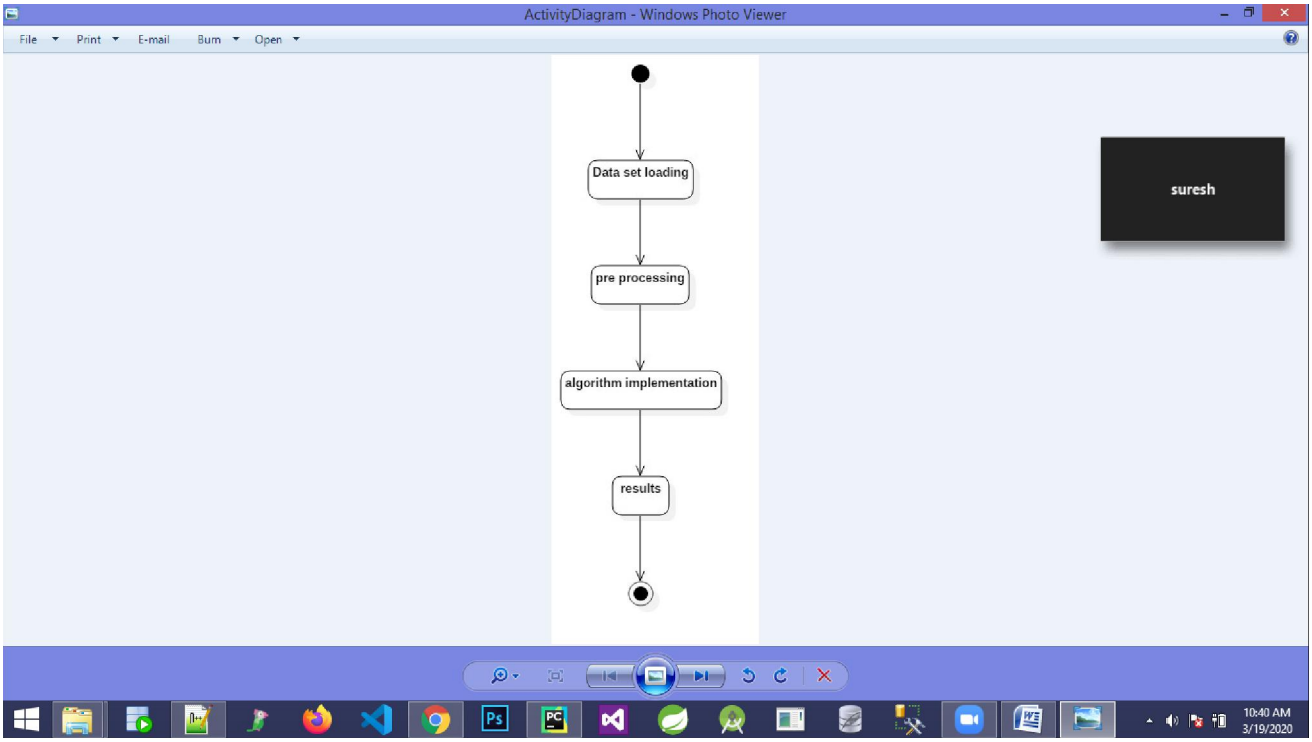


FIG 6

CHAPTER 4

IMPLEMENTATION

4.1 Implementation details

stressdetectionusingdecisiontree:

```
import pandas as pd
from sklearn.model_selectionimport train_test_split
from sklearnimport preprocessing
from sklearn.decompositionimport PCA
from matplotlib import pyplotas plt
from sklearn.naive_bayesimport GaussianNB
from sklearnimport metrics
from sklearn.treeimport DecisionTreeClassifier

df = pd.read_excel('C:\\Users\HEMNATH SIN\Desktop\Stress_Detection-
master\stress_data.xlsx',header=None)

df.columns=['Target', 'ECG(mV)', 'EMG(mV)','Foot GSR(mV)','Hand GSR(mV)',
'HR(bpm)','RESP(mV)']
X_train, X_test, y_train, y_test = train_test_split(df[['ECG(mV)', 'EMG(mV)','Foot
GSR(mV)','Hand GSR(mV)', 'HR(bpm)','RESP(mV)']], df['Target'],
test_size=0.30, random_state=12345)

# Min-Max Scaling

minmax_scale = preprocessing.MinMaxScaler().fit(df[['ECG(mV)', 'EMG(mV)','Foot
GSR(mV)','Hand GSR(mV)', 'HR(bpm)', 'RESP(mV)']])
df_minmax = minmax_scale.transform(df[['ECG(mV)', 'EMG(mV)','Foot GSR(mV)','Hand GSR(mV)',
'HR(bpm)','RESP(mV)']])
X_train_norm, X_test_norm, y_train_norm, y_test_norm = train_test_split(df_minmax,
df['Target'],
test_size=0.30, random_state=12345)

def plot():
plt.figure(figsize=(8,6))

plt.scatter(df['Hand GSR(mV)'], df['HR(bpm)'],
color='green', label='input scale', alpha=0.5)

plt.scatter(df_minmax[:,0], df_minmax[:,1],
color='blue', label='min-max scaled [min=0, max=1]', alpha=0.3)

plt.title('Hand GSR and HR content of the physiological dataset')
plt.xlabel('Hand GSR')
plt.ylabel('HR')
plt.legend(loc='upper left')
plt.grid()

plt.tight_layout()

plot()
plt.show()

# on non-normalized data
```

```
model = DecisionTreeClassifier(max_leaf_nodes=3)
fit = model.fit(X_train, y_train)

# on normalized data
model_norm = DecisionTreeClassifier(max_leaf_nodes=3)
fit_norm = model_norm.fit(X_train_norm, y_train)

pred_train = model.predict(X_train)

pred_test = model.predict(X_test)

print('Accuracy measure for dataset')
print('{:.2%}\n'.format(metrics.accuracy_score(y_test, pred_test)))

pred_train_norm = model_norm.predict(X_train_norm)

print('Accuracy measure for normalized dataset')

pred_test_norm = model_norm.predict(X_test_norm)

print('{:.2%}\n'.format(metrics.accuracy_score(y_test, pred_test_norm)))


pred = model.predict([[0.001,0.931,5.91,19.773,99.065,35.59]])
print(pred)
pred = model.predict([[-0.005,0.49,8.257,9.853,66.142,45.998]])
print(pred)

pred = model_norm.predict([[0.001,0.931,5.91,19.773,99.065,35.59]])
print(pred)
pred = model_norm.predict([[0.005,0.49,8.257,5.853,80.142,45.998]])
print(pred)
```

stressdetectionusingnaivebayes:

```
import pandas as pd
from matplotlib import pyplotas plt
from sklearnimport preprocessing
from sklearn.model_selectionimport train_test_split
from sklearn.naive_bayesimport GaussianNB
from sklearnimport metrics

df = pd.read_excel('C:\\Users\HEMNATH SIN\Desktop\Stress_Detection-
master\stress_data.xlsx', header=None)

df.columns=['Target', 'ECG(mV)', 'EMG(mV)','Foot GSR(mV)','Hand GSR(mV)',
'HR(bpm)','RESP(mV)']
X_train, X_test, y_train, y_test = train_test_split(df[['ECG(mV)', 'EMG(mV)','Foot
GSR(mV)','Hand GSR(mV)', 'HR(bpm)','RESP(mV)']], df['Target'],
test_size=0.30, random_state=12345)

# Min-Max Scaling

minmax_scale = preprocessing.MinMaxScaler().fit(df[['ECG(mV)', 'EMG(mV)','Foot
GSR(mV)','Hand GSR(mV)', 'HR(bpm)','RESP(mV)']])
df_minmax = minmax_scale.transform(df[['ECG(mV)', 'EMG(mV)','Foot GSR(mV)','Hand GSR(mV)',
'HR(bpm)','RESP(mV)']])
X_train_norm, X_test_norm, y_train_norm, y_test_norm = train_test_split(df_minmax,
df['Target'],
test_size=0.30, random_state=12345)

def plot():
plt.figure(figsize=(8,6))
```

```
plt.scatter(df['Hand GSR(mV)'], df['HR(bpm)'],
color='green', label='input scale', alpha=0.5)

plt.scatter(df_minmax[:,0], df_minmax[:,1],
color='blue', label='min-max scaled [min=0, max=1]', alpha=0.3)

plt.title('Hand GSR and HR content of the physiological dataset')
plt.xlabel('Hand GSR')
plt.ylabel('HR')
plt.legend(loc='upper left')
plt.grid()

plt.tight_layout()

plot()
plt.show()

# on non-normalized data
gnb = GaussianNB()
fit = gnb.fit(X_train, y_train)

# on normalized data
gnb_norm = GaussianNB()
fit_norm = gnb_norm.fit(X_train_norm, y_train)

pred_train = gnb.predict(X_train)
pred_test = gnb.predict(X_test)

# Accuracy measure for datasets

print('Accuracy measure for dataset')
print('{:.2%}\n'.format(metrics.accuracy_score(y_test, pred_test)))

print('Accuracy measure for normalized dataset')
pred_test_norm = gnb_norm.predict(X_test_norm)
print('{:.2%}\n'.format(metrics.accuracy_score(y_test, pred_test_norm)))


# comparing the true and predicted responses

print('True target values: ',y_test.values[0:25])
print('Predicted target values: ',pred_test_norm[0:25])

# Confusion Matrix
print(metrics.confusion_matrix(y_test,pred_test_norm))
print('True: ',y_test.values[0:25])
print('Pred: ',pred_test_norm[0:25])
print()
confusion = metrics.confusion_matrix(y_test,pred_test_norm)
TP = confusion[1,1]
TN = confusion[0,0]
FP = confusion[0,1]
FN = confusion[1,0]

# Metrics calclulation using confusion matrix

print()
# Classsification accuracy:- how often is the classifier correct
print('Classification Accuracy:- ' , metrics.accuracy_score(y_test,pred_test_norm))

# Classification error/Misclassification rate:- how often is the classifier is incorrect
print('Classification Error:- ' , 1-metrics.accuracy_score(y_test,pred_test_norm))

# Sensitivity :- when the actual value is positive , how often is the prediction correct?
print('Sensitivity:- ' , metrics.recall_score(y_test,pred_test_norm))
```

```
# Specificity:- when the actual value is negative ,how often the prediction is the
correct?
print('Specificity:- ' , TN/float(TN+FP))

# False positive rate:- when the actual value is negative ,how often the prediction is the
incorrect?
print('False positive rate:- ' , FP/float(TN+FP))


# Precision:- when a positive value is predicted , how often is the prediction correct?
print('Precision:- ' , metrics.precision_score(y_test,pred_test_norm))

# Prediction of stress/no stress class on new dataset
print()
pred_data_norm = minmax_scale.transform([[-0.005,0.49,8.257,5.853,66.142,45.998]])
pred = gnb_norm.predict(pred_data_norm)
print('Predicted class for dataset [-0.005,0.49,8.257,5.853,66.142,45.998]:- ', pred)

pred_data_norm = minmax_scale.transform([[0.001,0.931,5.91,19.773,99.065,35.59]])
pred = gnb_norm.predict(pred_data_norm)
print('Predicted class for dataset [0.001,0.931,5.91,19.773,99.065,35.59]:- ', pred)
```

CHAPTER-5

TEST RESULTS

5.1 TEST CASES

The reason for testing is to find blunders. Testing is the way toward attempting to find each possible shortcoming or shortcoming in a work item. “It gives an approach to check the usefulness of parts, sub congregations, gatherings or potentially a completed item It is the way toward practicing software with the plan of guaranteeing that the Software framework lives up to its prerequisites and client desires and does not bomb in an unsuitable way. There are different kinds of test”. Each test type tends to a particular testing necessity.

Types of Testing:

The different kinds of testing performed on the developed project are listed below:

5.1.1 Unit Testing:

Unit testing includes the “plan of experiments that approve that the inward program rationale is working appropriately, and that program information sources produce substantial yields. All choice branches and interior code stream ought to be approved. It is the trying of individual software units of the application .it is done after the finishing of an individual unit before joining. This is an auxiliary testing, that depends on information of its development and is obtrusive. Unit tests perform essential tests at part level and test a particular business procedure, application, as well as framework design. Unit tests guarantee that every novel way of a business procedure performs precisely to the reported particulars and contains unmistakably characterized sources of info and anticipated outcomes”.

5.1.2 Integration Testing:

Coordination tests are intended to test incorporated “software segments to decide whether they really keep running as one program. Testing is occasion driven and is increasingly worried about the essential result of screens or fields. Coordination tests exhibit that in spite of the fact that the segments were exclusively fulfillment, as appeared by effectively unit testing, the mix of segments is

right and reliable". Coordination testing is explicitly planned for uncovering the issues that emerge from the blend of segments.

Practical testing is focused on the accompanying things:
Legitimate Input : recognized classes of substantial information must be acknowledged.

Invalid Input : distinguished classes of invalid information must be rejected.
Capacities : recognized capacities must be worked out.
Yield : recognized classes of use yields must be worked out.
Frameworks/Procedures: interfacing frameworks or systems must be summoned.
Association and arrangement of utilitarian tests is "centered around prerequisites, key capacities, or uncommon experiments. Moreover, efficient inclusion relating to recognize Business procedure streams; information fields, predefined procedures, and progressive procedures must be considered for testing. Before utilitarian testing is finished, extra tests are recognized and the viable estimation of current tests is resolved".

System Test

Framework testing guarantees that the whole coordinated software framework meets prerequisites. "It tests a design to guarantee known and unsurprising outcomes. A case of framework testing is the design situated framework reconciliation test". Framework testing depends on procedure portrayals and streams, underlining pre-driven procedure connections and combination focuses.

White Box Testing

White Box Testing is a "trying wherein in which the software analyzer knows about the inward operations, structure and language of the software, or if nothing else its motivation. It is reason. It is utilized to test regions that can't be come to from a discovery level".

Block Testing

Discovery Testing will be trying the software with no information of the internal operations, structure or language of the module being tried. Discovery tests, "as most different sorts of tests, must be composed from an authoritative source record, for example, particular or prerequisites report, for example, detail or necessities archive. It is a trying wherein the software under test is dealt with, as a discovery .you can't "see" into it". The test gives data sources and reacts to yields without thinking about how the software functions.

Unit Testing:

"Unit testing is normally led as a feature of a joined code and unit test period of the software lifecycle, in spite of the fact that it isn't remarkable for coding and unit testing to be directed as two unmistakable stages".

Test methodology and approach
Field testing will be performed physically and practical tests will be written in detail. \
Test goals

- All field passages must work appropriately.
- Pages must be actuated from the recognized connection.
- The section screen, messages and reactions must not be deferred.

Highlights to be tried

- Verify that the sections are of the right design

- No copy sections ought to be permitted
- All connections should take the client to the right page.

Integration Testing

Software reconciliation testing is the gradual mix testing of at least two coordinated software parts on a solitary stage to deliver disappointments brought about by interface deserts.

The undertaking of the "reconciliation test is to watch that segments or software applications, for example parts in a software framework or – one stage up – software applications at the organization level – collaborate without blunder".

Test outcomes: All the experiments referenced above passed effectively. No imperfections experienced.

Acceptance Testing

Client Acceptance Testing is a "basic period of any task and requires critical cooperation by the end client. It additionally guarantees that the framework meets the utilitarian prerequisites".

Test outcomes: All the experiments referenced above passed effectively. No imperfections experienced.

MAINTAINENCE

This covers a wide scope of exercises including rectifying code and structure blunders. "To lessen the requirement for upkeep over the long haul, we have all the more precisely characterized the client's prerequisites during the procedure of framework advancement. Contingent upon the prerequisites, this framework has been created to fulfill the requirements to the biggest conceivable degree. With improvement in innovation, it might be conceivable to include a lot more highlights based the necessities in future". The coding and planning is basic and straightforward which will make support simpler.

TESTING STRATEGY :

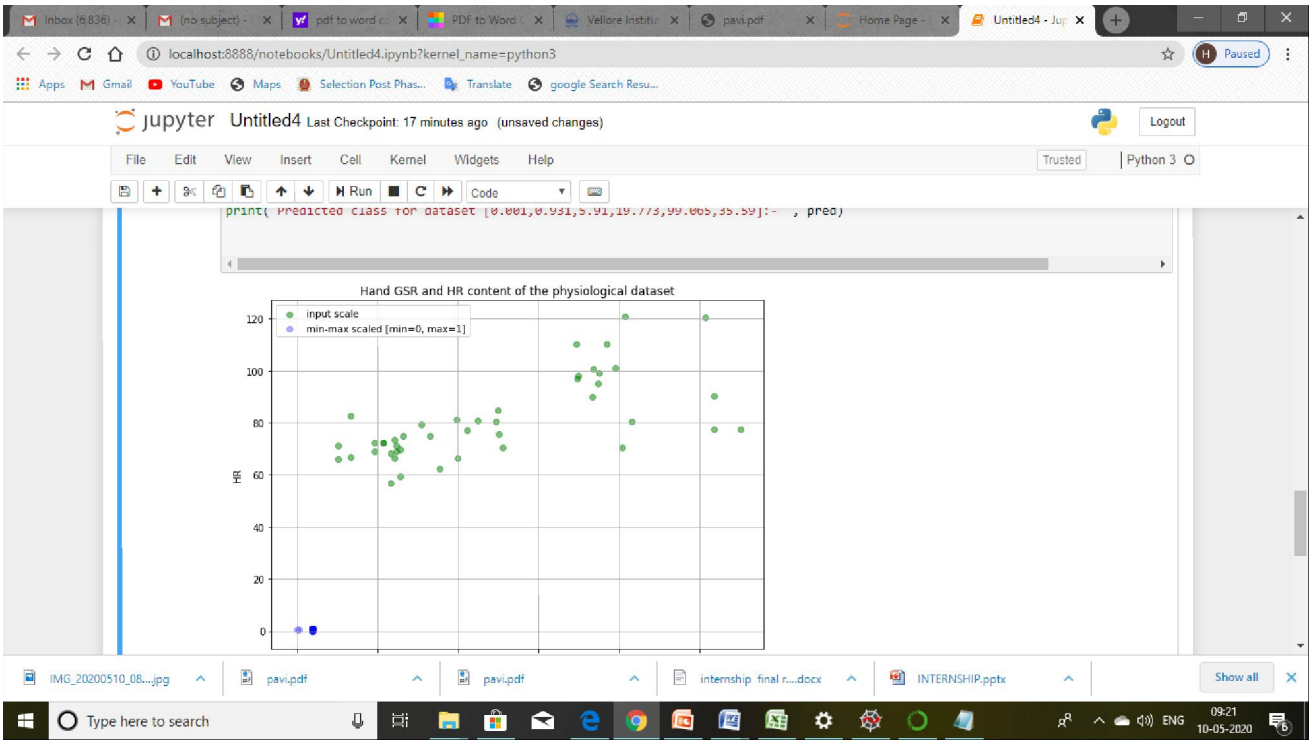
A procedure for framework testing coordinates framework experiments and "structure systems into a very much arranged arrangement of steps that outcomes in the effective development of software. The testing methodology must co-work test arranging, experiment configuration, test execution, and the resultant information gathering and assessment.A procedure for software testing must suit low-level tests that are important to check that a little source code fragment has been accurately actualized just as abnormal state tests that approve significant framework capacities against client prerequisites".

CHAPTER -6

RESULTS AND DISCUSSION

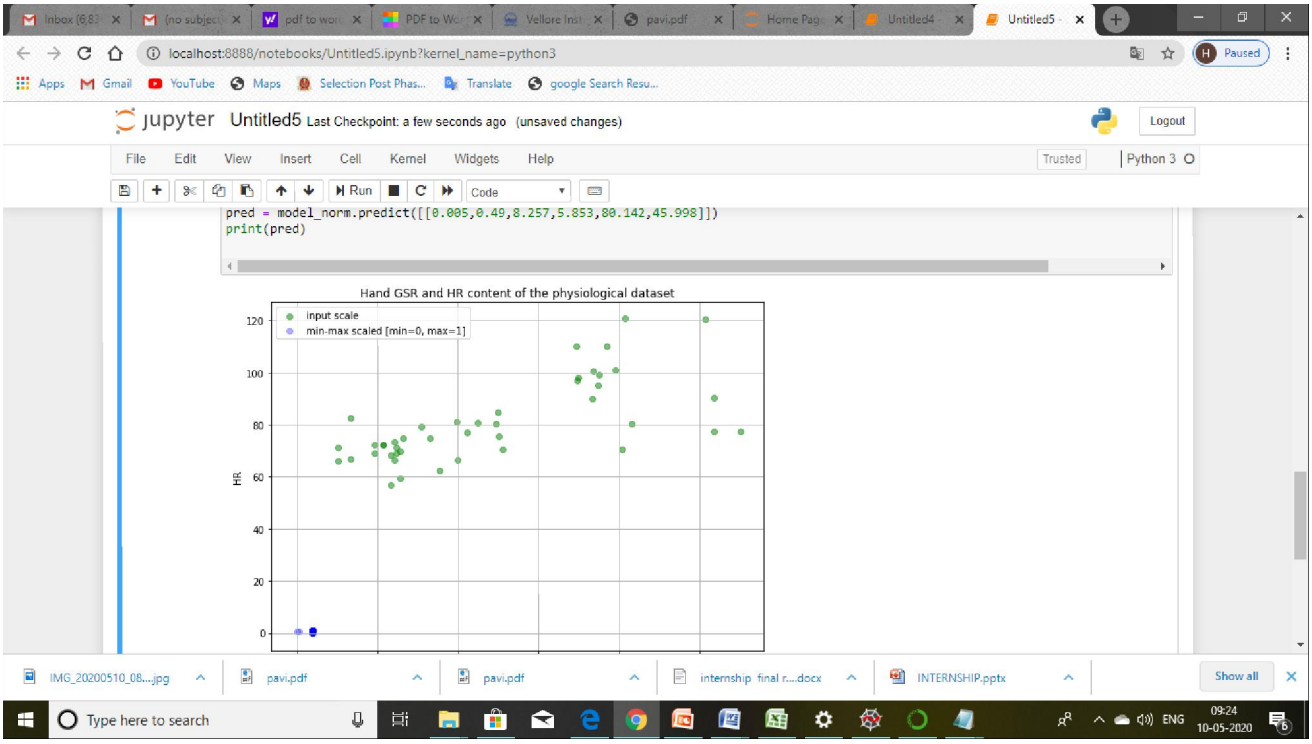
OUTPUT:

GUASSIAN NB




```
Accuracy measure for dataset
100.00%

Accuracy measure for normalized dataset
100.00%

True target values:  [0 1 0 0 0 1 0 0 1 0 0 0 1 1]
Predicted target values:  [0 1 0 0 0 1 0 0 1 0 0 0 1 1]
[[9 0]
 [0 5]]
True:  [0 1 0 0 0 1 0 0 1 0 0 0 1 1]
Pred:  [0 1 0 0 0 1 0 0 1 0 0 0 1 1]


Classification Accuracy:-  1.0
Classification Error:-  0.0
Sensitivity:-  1.0
Specificity:-  1.0
False positive rate:-  0.0
Precision:-  1.0

Predicted class for dataset [-0.005,0.49,8.257,5.853,66.142,45.998]:-  [0]

Predicted class for dataset [0.001,0.931,5.91,19.773,99.065,35.59]:-  [1]
```

DECISION TREE.PY




```
Accuracy measure for dataset
100.00%

Accuracy measure for normalized dataset
100.00%

[1]
[0]
[1]
[1]
```

CHAPTER -7

CONCLUSION AND FUTURE AND WORK

7.1 Conclusions:

We made use of decision tree and naive bayes techniques in order to cluster the user data and provide an accuracy for the user stress levels.that are gathered and provided by graph internet explorer.In this system, we displayed a system for distinguishing users psychological stress states from clients

7.2 Future Scope:

We can make use of image for their detection of stress as it is posted by a user.Using these images we have to calculate the user Stress level.

CHAPTER-8

REFERENCES

[1] Huijie Lin, Jia Jia, JiezhonQiu, Yongfeng Zhang, LexingXie, Jie Tang, Ling Feng, and Tat-Seng Chua, "Detecting Stress Based on Social Interactions inSocial Networks", IEEE Transactions on Knowledge and Data Engineering, 2017.

.

[2] Aasia Manzoor, Hadia Awan &Sabita Mariam,-INVESTIGATING THE IMPACT OF WORK STRESS ON JOB PERFORMANCE: A Study on Textile Sector of Faisalabad, Asian Journal of Business and Management Sciences ISSN: 2047-2528 Vol. 2 No. 1 [20-28]

[3] Dr. Jyotsna Codaty, - Key to Stress Free Living V&S publications,2013,New Delhi,pp14,15,45,46.

[4] P. Kavitha, Role of stress among women employees forming majority workforce at IT sector in Chennai and Coimbatore, Tier-I & Tier-II centers, SONA- Global Management Review , Volume 6, Issues 3, May 2012.

[5] Kayoko Urakawa and KazuhitoYokoyam, ―Sense of Coherence (SOC) may reduce the Effects of Occupational Stress on Mental Health Status among Japanese Factory Workers, Journal ofIndustrial Health, Vol. 47 , No. 5 pp.503-508.

[6] Amir Shani and Abraham Pizam, ―Work-Related Depression among Hotel Employees, Cornell Hospitality Quarterly, Vol. 50, No. 4, 446-459 (2009) .

[7] Li-fang Zhang, -Occupational stress teaching approach among Chinese academics, Educational Psychology, Volume 29, Issue 2, March 2009, pages 203 – 219.

[8] Andrey Bogomolov, Bruno Lepri, Michela Ferron, Fabio Pianesi,and Alex Pentland. Daily stress recognition from mobile phone data,weather.

[9] Chris Buckley and EllenM Voorhees. Retrieval evaluation with incomplete information. In Proceedings of the 27th annual international ACM SIGIR conference on and development in informationretrieval, pages 25–32, 2004.

[10] Xiaojun Chang, Yi Yang, Alexander G Hauptmann, Eric P Xing,and Yao-Liang Yu. Semantic concept discovery for large-scale zero-shot event detection. In Proceedings of International Joint Conference on Artificial Intelligence, pages 2234–2240, 2015.