ONLINE MENTAL STRESS DETECTION BY USING MACHINE LEARNING

A Project Report Submitted

JNTUA, ANANTHAPURAMU

In partial fulfillment of the requirements for awards of degree of

BACHELOR OF TECHNOLOGY

in

MECHANICAL ENGINEERING

by

SOMALA VIKAS	**(20KB5A0313)**
V.SHARATH KUMAR REDDY	**(20KB5A0317)**
M.SIVA KUMAR	**(20KB5A0322)**
K. BHARGAV	**(20KB5A0331)**
K. BHARATH CHANDRA	**(20KB5A0333)**

Under the guidance of

DR. M. SREENIVASULU

Professor

DEPARTMENT OF MECHANICAL ENGINEERING

N.B.K.R. INSTITUE OF SCIENCE AND TECHNOLOGY

(AUTONOMOUS)

VIDYANAGAR - 524413

2022-2023

N.B.K.R. INSTITUE OF SCIENCE AND TECHNOLOGY

(AUTONOMOUS)

Affiliated to JNTUA, ANANTAPURAMU & Approved by AICATE

B. TECH., Courses Accredited by NBA under Tier I: Re-Accredited by NAAC with 'A' Grade

ISO 9001:2008 Certificated Institute

VIDYA NAGAR - 524413

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"Online Mental Stress Detection by Using Machine Learning"** is a Bonafide work done by

SOMALA VIKAS	**20KB5A0313**
V.SHARATH KUMAR REDDY	**20KB5A0317**
M.SIVA KUMAR	**20KB5A0322**
K. BHARGAV	**20KB5A0331**
K. BHARATH CHANDRA	**20KB5A0333**

In the Department of Mechanical Engineering, N.B.K.R. INSTITUE OF SCIENCE AND TECHNOLOGY, Vidyanagar and is submitted to JNTUA, Ananthapuramu in partial fulfillment for the award of degree of Bachelor of Technology in MECHANICAL ENGINEERING. This work has been carried out under my supervision

DR. M. SREENVISAULU
Professor
Department of Mechanical Engineering
N.B.K.R.I.S.T., VIDYANAGAR

DR. CH. R. VIKRAM KUMAR
Professor & Head
Department of Mechanical Engineering
N.B.K.R.I.S.T., VIDYANAGAR

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of a project would be incomplete without the people who made it possible of their constant guidance and encouragement crowned the efforts with success.

We would like to express the profound sense of gratitude to the project guide

Dr. M. SREENIVASULU, Professor, Department of Mechanical Engineering, N.B.K.R. Institute of Science & Technology, for his masterful guidance and the constant encouragement throughout the project. The sincere appreciations for his suggestions and unmatched services without, which this work would have been an unfulfilled dream.

We convey the special thanks to the management of **N.B.K.R. Institute of Science & Technology,** for providing excellent infrastructure in the campus for the completion of the project.

We convey the special thanks to **Sri N. RAM KUMAR** respectable correspondent of **N.B.K.R. Institute of Science & Technology**, for providing excellent infrastructure in the campus for the completion of the project.

We are grateful to **Dr. V. VIJAYA KUMAR REDDY, Director of N.B.K.R Institute of Science & Technology** for allowing us to avail all the facilities in the college.

We express the sincere gratitude to **Dr. CH.R. VIKRAM KUMAR, Professor & Head of the Department, Mechanical Engineering,** for providing exceptional facilities for successful completion of the project work.

We would like to convey the heartfelt thanks to the Parents, Staff members, Lab Technicians and the friends, who extended their cooperation in making this project as successful one.

We would like to thank one and all who have helped us directly and indirectly to complete this project successfully

ABSTRACT

Mental stress is a major issue nowadays, it affects day-to-day human life. The age that was considered once the most carefree is now under a large amount of stress. Stress increases nowadays leading to many problems like depression, suicide, heart attack, and stroke. Stress is one of the main factors that are affecting millions of lives. The objective is to analyse stress on human behaviour at different points in the student life. By using different machine learning algorithms like Naïve Bayes, KNN, Random Forest, and SVM is applied sensitivity, specificity, and accuracy are used as performance parameters. The accuracy and performance of data are further enhanced by applying 10-Fold Cross-Validation. The dataset is taken by answering set of questions on stress under different conditions of student life ,It help to analyse the stress level of the individuals. After creating an ML model by using the dataset we will create a website attaining the parameters from the dataset.

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Mental stress is a common problem that affects millions of people worldwide. It can cause various negative effects on an individual's physical and mental health, leading to anxiety, depression and other health issues. Therefore, it is crucial to detect mental stress at an early stage and take necessary measures to prevent it from worsening.

Traditional methods of detecting mental stress involve self-reporting or clinical observation which can be subjective and unreliable. With the advent of wearable technology and machine learning algorithms, online mental stress detection has become a promising approach to detect and monitor mental stress in real-time.

The stress we experience is a physical reaction to mental excitement or actual difficulties. Modest quantities of stress might be wanted, useful, and surprisingly solid. In any case, unnecessary pressure may prompt real mischief and increment the danger of sickness say for example: High blood pressure, sugar, spondylitis, asthma, organ failure and ulcers etc., Stress can be a dangerous factor for the undisputed amounts of mental instability and pain.

As per Lancet report (2012) Every hour , a student commits suicide in the different part of the country. The country has reported large suicide cases of the youngsters aging between the age group of fifteen and twenty-nine. This motivated us to take up this work.

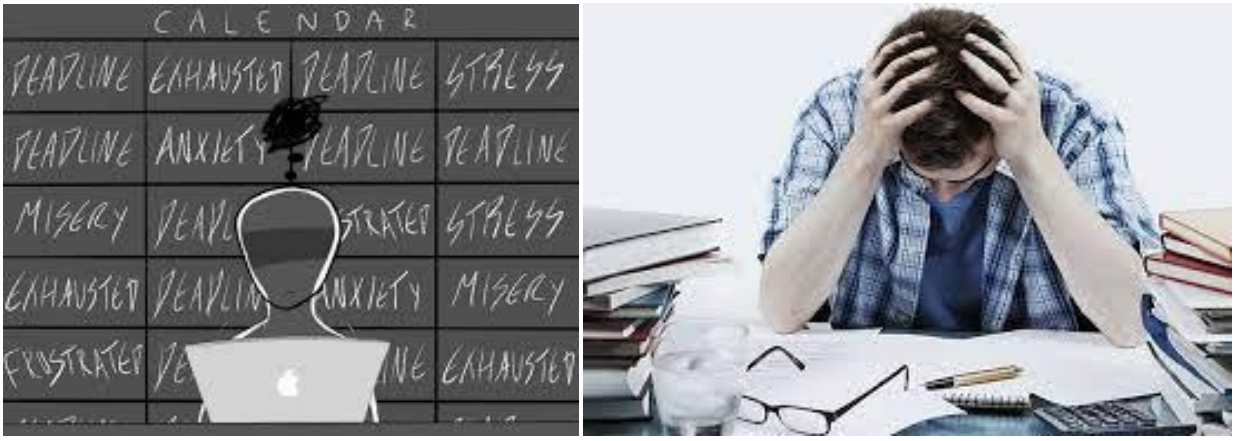


Fig: 1.1 Mental stress

1.1 Benefits of detecting mental stress at an early stage:

1.1.1 Prevention: Early detection of mental stress can help prevent it from escalating into more severe mental health problems, such as anxiety or depression. By identifying stressors and

providing appropriate support and coping strategies, students can learn to manage their stress and prevent it from interfering with their academic performance and personal life.

1.1.2 Academic performance: Stress can impair cognitive functions such as memory and attention. This makes it difficult for students to concentrate, learn and recall the information. By detecting and addressing stress early on, students can improve their academic performance.

1.1.3 Career prospects: Managing stress effectively can improve students' confidence, motivation, and self-efficacy, which can enhance their career prospects and success.

In overall, detecting and addressing mental stress at an early stage can have significant benefits for students' academic performance, health, well-being, and future success. It is important to create a supportive and inclusive environment that promotes mental health and well-being and provides early intervention and support for those who need it.

1.2 Machine learning algorithms used for online mental stress detection

There are several machine learning algorithms that can be used for online mental stress detection

1.2.1 Support Vector Machines (SVM): SVM is a popular algorithm for binary classification, which can be used to predict whether an individual is experiencing stress or not. SVM works by identifying the hyperplane that best separates the two classes and uses this to make predictions on new data.

1.2.2 Artificial Neural Networks (ANN): ANN is a type of machine learning algorithm that is modelled after the structure of the human brain. It can be used for complex classification tasks, such as identifying different types of stress. ANN works by creating a network of interconnected nodes that process information and make predictions.

1.2.3 Decision Trees: Decision Trees are a simple but effective algorithm that can be used for classification tasks. They work by creating a tree-like model of decisions and their possible consequences, allowing for easy interpretation and understanding of the results.

1.2.4 Random Forests: Random Forests are an ensemble learning algorithm that combines multiple decision trees to improve accuracy and reduce overfitting. They work by randomly selecting subsets of features and samples to build a collection of decision trees, and then combining their predictions to make a final prediction.

1.2.5 Logistic Regression: Logistic Regression is a statistical algorithm that can be used for binary classification tasks, such as predicting whether an individual is experiencing stress or not. It works by modelling the relationship between the dependent variable and one or more independent variables, allowing for the calculation of probabilities and predictions.

In overall, machine learning algorithms can be useful tools for online mental stress detection by asking questions. The choice of algorithm will depend on the specific task and the available data. It is important to carefully select and evaluate the algorithm to ensure accuracy and avoid bias.

CHAPTER 2

LITERATURE SURVEY

Ravinder Ahujaa *et al.*, [1] The paper "Mental Stress Detection in University Students using Machine Learning Algorithms" aims to develop a model for detecting mental stress in university students using machine learning algorithms. The authors collected data from 100 students from their university and used physiological signals such as heart rate, skin conductivity, and body temperature to classify stress levels. The authors used various machine learning algorithms, including K-nearest neighbours (KNN), decision tree, and support vector machine (SVM), to classify the stress levels of students. They found that the SVM algorithm performed the best, achieving an accuracy rate of 93.9%.Overall, the paper provides an interesting approach to detect mental stress in university students using machine learning algorithms. However, the study has certain limitations, including a relatively small sample size and the use of physiological signals to detect stress levels, which may not be reliable indicators of mental stress.

Mr. Purnendu Shekhar Pandey BML *et al.*, [2] The paper begins by providing an overview of the concept of stress, its causes, and its impact on individuals. The authors then discuss the importance of early detection and intervention in managing stress. The paper then describes how machine learning and IoT can be used to predict and detect stress. The authors explain that IoT devices can be used to collect data on an individual's physiological responses, such as heart rate and blood pressure, which can be used as inputs to machine learning models. The authors provide a detailed explanation of the different types of machine learning algorithms that can be used for stress prediction and detection, such as decision trees, support vector machines, and neural networks. They also discuss the importance of feature selection and data pre -processing in developing accurate models. Finally, the authors provide a survey of existing literature on the use of machine learning and IoT for stress prediction and detection. They conclude that the combination of machine learning and IoT has great potential for improving the early detection and management of stress in individuals. Overall, the paper provides a useful overview of the potential applications of machine learning and IoT for stress prediction and detection. However, it would benefit from more specific examples of how these technologies have been applied in practice, as well as a discussion of potential challenges and limitations.

Russell Li1 and Zhidong Liu *et al.*, [3] This paper begins by discussing the importance of stress detection and its potential impact on healthcare, productivity, and quality of life. The authors then introduce their approach, which involves using deep neural networks to analyse physiological data collected from wearable sensors The authors provide a detailed explanation of their methodology, which involves pre-processing the physiological data and training a deep neural network to classify the data into different stress levels. They also discuss the importance of feature selection and model optimization in achieving accurate results. This paper presents the results of experiments conducted on a dataset of physiological data collected from 20 individuals. The authors report that their approach achieved an accuracy of 87.5% in detecting stress levels, outperforming other machine learning algorithms such as decision trees and support vector machines. The authors also discuss the potential applications of their approach, such as in the development of personalized stress management tools and in monitoring the stress levels of individuals in high-stress environments. Overall, the paper provides a well-researched and well-documented approach to stress detection using deep neural networks. The authors provide a clear explanation of their methodology and present convincing evidence of the effectiveness of their approach. However, it

would benefit from a more detailed discussion of the limitations and potential biases of their approach, as well as a more thorough evaluation of its real-world applicability.

Shruti Gedam *et al.*, [4] The paper begins by discussing the importance of stress detection and its potential impact on health and wellbeing. The author then explains how wearable sensors can be used to collect physiological data, such as heart rate variability and skin conductance, which can be used as inputs to machine learning models for stress detection. The author provides a detailed review of the literature on the use of wearable sensors and machine learning for stress detection, discussing different types of sensors and machine learning algorithms that have been used in previous studies. The paper also discusses the potential applications of automatic stress detection, such as in the development of personalized stress management tools and in monitoring the stress levels of individuals in high-stress environments. Overall, the paper provides a useful review of the state-of-the-art in automatic stress detection using wearable sensors and machine learning. The author provides a comprehensive overview of the different approaches and techniques that have been used in previous studies, and highlights the potential applications of this technology. However, the paper would benefit from a more detailed discussion of the limitations and potential biases of this approach, as well as a more thorough evaluation of its real-world applicability.

Aravind Pradeep *et al.*, [5] The paper begins by discussing the importance of stress detection and the limitations of traditional approaches to stress assessment. The authors then explain how they used biosignals, such as heart rate variability and electrodermal activity, as inputs to their stress detection model.The authors describe their approach, which involves using a decision tree to select the most relevant features from the bio signal data and then using an SVM model to classify the data into different stress levels. They also discuss the importance of feature selection and model optimization in achieving accurate results. The paper presents the results of experiments conducted on a dataset of bio signal data collected from 20 individuals. The authors report that their approach achieved an accuracy of 85% in detecting stress levels, outperforming other machine learning algorithms such as k-nearest neighbours and random forest. The authors also discuss the potential applications of their approach, such as in the development of personalized stress management tools and in monitoring the stress levels of individuals in high-stress environments. Overall, the paper presents a well-researched and well-documented approach to stress detection using a combination of decision trees and SVM. The authors provide a clear explanation of their methodology and present convincing evidence of the effectiveness of their approach. However, it would benefit from a more detailed discussion of the limitations and potential biases of their approach, as well as a more thorough evaluation of its real-world applicability.

S. Binny *et al.*, [6]

The paper titled "Classification of Lung disease using deep learning neural networks" by S. Binny and Dr. P. Sardar Maran published in the June 2021 issue of PENSEE, focuses on the application of deep learning techniques to classify different types of lung diseases. The authors present a novel approach to identify three types of lung diseases, namely pneumonia, bronchiectasis, and COVID-19, from chest X-ray images. They used a convolutional neural network (CNN) model, which is a type of deep learning neural network that is commonly used in image recognition tasks. The

authors trained the CNN model using a dataset of 1,501 chest X-ray images, which included 500 images of pneumonia, 500 images of bronchiectasis, and 501 images of COVID-19. The dataset was divided into training and testing sets, and the performance of the CNN model was evaluated based on accuracy, sensitivity, specificity, precision, and F1 score. The experimental results showed that the CNN model achieved an accuracy of 97.07%, sensitivity of 97.36%, specificity of 98.23%, precision of 96.90%, and F1 score of 97.33%. These results indicate that the proposed CNN model is effective in accurately classifying lung diseases from chest X-ray images. Overall, the paper provides valuable insights into the use of deep learning techniques for the classification of lung diseases, and the results obtained by the authors are promising. However, it should be noted that the dataset used in the study is relatively small, and further research using larger datasets is needed to validate the effectiveness of the proposed approach.

2.2. APPROACH IN THIS WORK

Mental stress is a major issue nowadays, especially among youngsters. The age that was considered once most carefree is now under a large amount of stress. Stress increase nowadays leads to many problems like depression, suicide, heart attack, and stroke. In this paper, we are calculating the mental stress of students one week before the exam and during the usage of the internet. The objective is to analyze stress in the college students at different points in his life. The effect that exam pressure or recruitments stress has on the student which often goes unnoticed. We will perform an analysis on how these factors affect the mind of a student and will also correlate this stress with the time spent on the internet

The dataset was taken from the 206 students of Jaypee Institute of Information Technology Noida. We have classified the data in two conditions one is before the exams and other is stress due to the usage of the internet. The dataset was collected for PSS test which includes 14 questions overall including the entire emotional question. The marking for the questions was in 5 ways (a) Never (b) Almost Never (c) Sometimes (d) fairly Often (e) Very Often. Then the weighted average model is used, preference is given to every question. The students are divided into 3 categories highly stressed, stressed and normal. The dataset was collected from students in the college. They were asked basic questions about their feelings in situations that they might have encountered in the last month and their reactions to it [9]. Their answers are given some number of weights and the weights thus help to calculate a score to analyze the stress level of the individuals.

CHAPTER 3

METHODOLOGY

Stress is a term frequently utilized synonymously with negative life experiences or life occasions. Logical research on pressure and uneasiness offers different points of view on the issue. The expanding pace of life hurried and focused ways of life imply that stress is an integral part of human life. A man in a condition of adjusting to pressure demonstrates conduct resistances.

Fig 3.1 Stress due to live experiences

Evaluation is also known is endurance testing in software engineering. Software is tested for a short period of time to determine its endurance capacity .The most common application of testing is to determine the level at which the system, software, or hardware breaks. It also evaluates if the system displays right error management under these scenarios.

The testing process can be categorised into major steps:

Step 1) Test Planning: This is the stage when the acquire system data, analyse the system. Data was collected through a survey. Machine Learning models are used to analyse the data to get the output.

Step 2) Create Automation Scripts: as a part of this step created the anxiety testing automation scripts using python and its libraries Pandas, NumPy, seaborn, Matplotlib .

Script Execution: At this stage, we can run the testing automation scripts and save the results.

Step 3) Analysis of findings: At this point, we can analyse the Test findings and identify from student stress at part they feel stress in the education system of by check their stress in website holdups.

Step 4) Modifying and Optimisation: At this stage, we are going to fine-tune the system, alter configurations, and optimise the code with the goal to meet the set benchmark.

Here, introduction to Machine Learning, Pandas, NumPy, seaborn and Matplotlib etc., was mentioned below.

3 MACHINE LEARNING

3.1 SYSTEM MODULES.

3.1.1 Working on dataset:

System checks for data whether it is available or not and load the data in csv files.

3.1.2 Pre-processing:

Data need to be pre-processed according the models it helps to increase the accuracy of the model and better information about the data.

3.1.3 Training the data:

After pre-processing the data will split into two parts as train and test data before training with the given algorithms.

3.1.4 Model Building

To create a model that predicts the personality with better accuracy, this module will help user.

3.1.5 Generated Score:

Here user view the score in percentage

3.1.6 Generate Results:

We train the machine learning algorithm and predict the sentiment of the tweet.

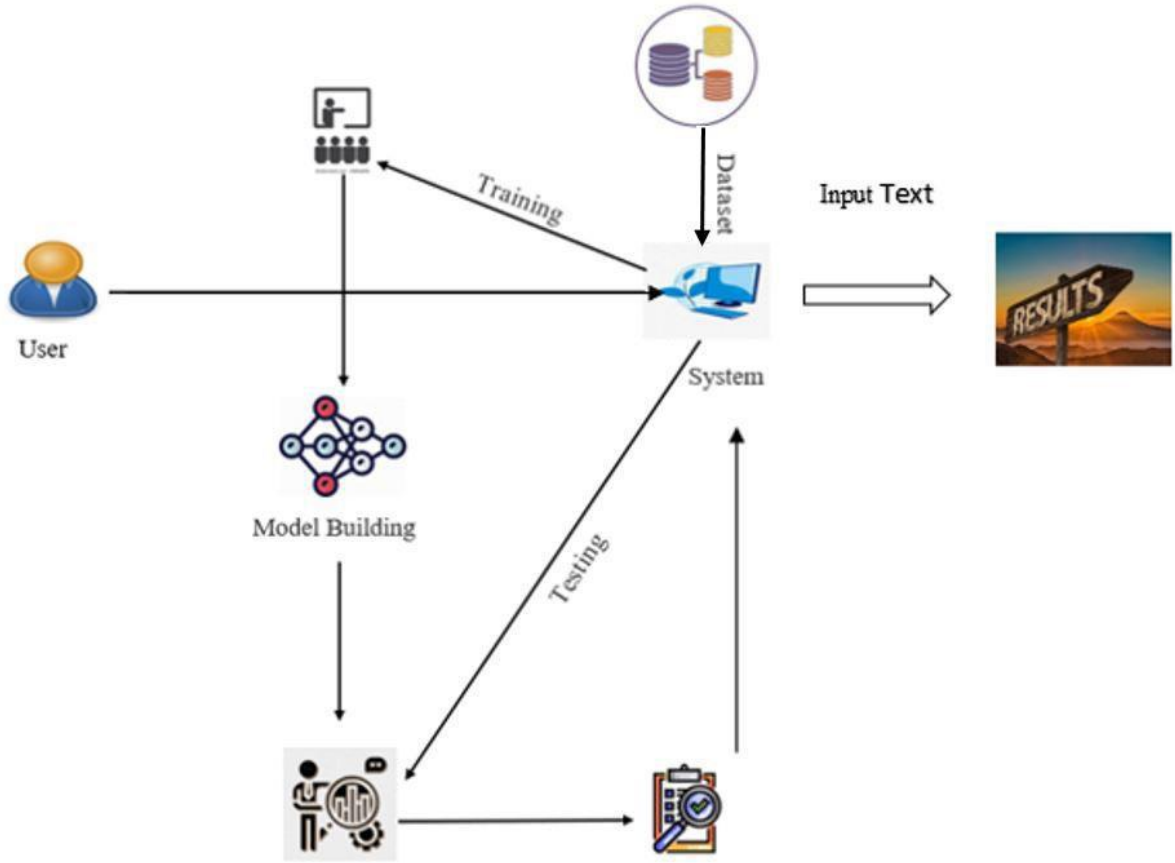


Fig 3.2 Machine Learning approach for appropriate prediction

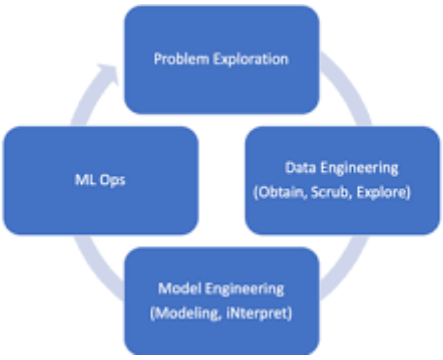


Fig 3.3 Machine Learning Cycle

3.2. CLASSIFICATION ALGORITHM:

3.2.1 K- Nearest Neighbour:

KNN also called K- nearest neighbour is a supervised machine learning algorithm that can be used for classification and regression problems. This is one of the simplest algorithms to learn and uses non-parametric representation i.e., it does not make any assumptions for underlying data. KNN is also termed as a lazy algorithm as it does not learn during the training phase rather it stores the data points but learns during the testing phase. It is a distance-based algorithm.

3.2.2 Working Principle of KNN:

Fig 3.4 Working Principle of KNN

3.2.3 Naive Bayes:

Naive Bayes is a probabilistic machine learning algorithm that is widely used for classification tasks. It is based on Bayes' theorem, which states that the probability of a hypothesis (in this case, a class label) given some evidence (the input features) is proportional to the probability of the evidence given the hypothesis times the prior probability of the hypothesis.

The "naive" in Naive Bayes refers to the assumption that the input features are conditionally independent in the given the class label. This simplifies the calculation of the probabilities and makes the algorithm computationally efficient.

In Naive Bayes classification, the algorithm first calculates the prior probability of each class label based on the frequency of each label in the training data. It then calculates the conditional probability of each feature given each class label, also based on the frequency of each feature given each label in the training data. These probabilities are used to calculate the posterior probability of each class label given the input features using Bayes' theorem.

The class label with the highest posterior probability is then assigned to the input. Naive Bayes is known for its simplicity and speed, and it can work well even with a small amount of training data. However, the assumption of conditional independence may not always hold in real-world data, which can lead to suboptimal performance.

3.2.4 SVM:

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

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

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

3.2.5 Random Forest:

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest splits out a class prediction and the class with the most votes become the model's prediction.

Bagging stands for bootstrap aggregation. It combines multiple learners in a way to reduce the variance of estimates.

Random Forest algorithm is a supervised classification algorithm implemented based on the concept of bagging. There is a direct relationship between the number of trees in the forest and the results it can get: the larger the number of trees, the more accurate the result.

The advantages of using Random Forest over Decision Trees are:

- Overfitting is one critical problem that may make the results worse, but for Random Forest algorithm, if there are enough trees in the forest, the classifier won't overfit the model.

- For both classification and regression task, the same random forest algorithm can be used.
- The Random Forest algorithm can be used for identifying the most important features from the training dataset, in other words, feature engineering.

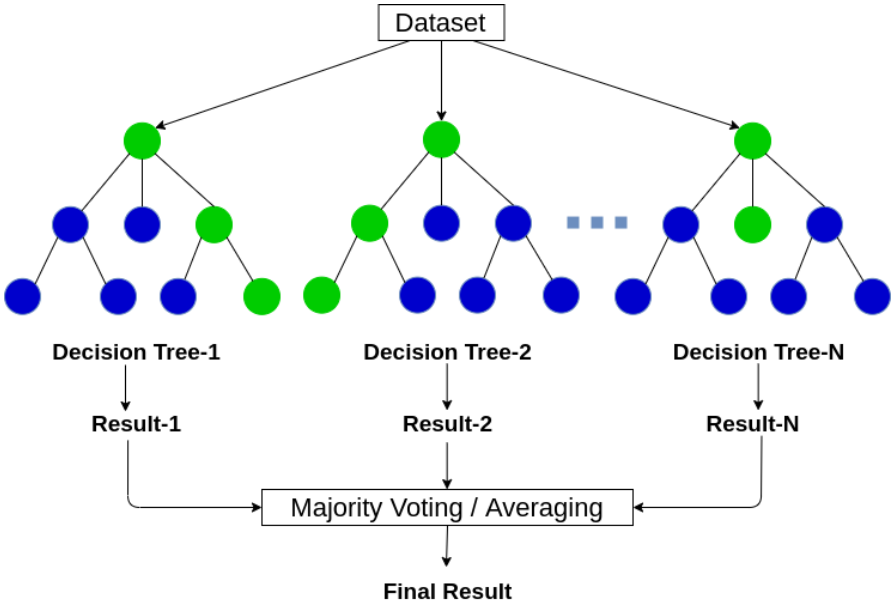


Fig 3.5 Random Forest regression flow chart

3.3 Introduction to Python and web technologies.

3.3.1 What Is a Script?

A Python script, the file containing the commands, is structured to be executed like a program. These files are designed to contain various functions and import various modules. Python interactive shell or the respective command line is used to execute the script, to perform a specific task.

3.3.2 Scripts are reusable

Basically, a script is a text file containing the statements that comprise a Python program. Once we have created the script, this can execute it over and over without having to retype it each time.

3.3.3 Scripts are editable

Perhaps, more importantly, we can make different versions of the script by modifying the statements from one file to the next using a text editor. Then we can execute each of the individual versions. In this way, it is easy to create different programs with a minimum amount of typing.

3.3.4 It will need a text editor

Just about any text editor will suffice for creating Python script files.

We can use Microsoft Notepad, Microsoft WordPad, Microsoft Word, or just about any word processor they want to.

3.4 Difference between a script and a program

3.4.1 Script:

Scripts are distinct from the core code of the application, which is usually written in a different language, and are often created or at least modified by the end-user. Scripts are often interpreted from source code or byte code, whereas the applications they control are traditionally compiled to native machine code.

3.4.2 Program:

The program has an executable form that the computer can use directly to execute the instructions. The same program in its human-readable source code form, from which executable programs are derived(e.g., compiled)

3.5 Python

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis.

3.5.1 Python concepts

If they not interested in the how's and whys of Python, feel free to skip to the next chapter. In this chapter I will try to explain to the reader why I think Python is one of the best languages available and why it's a great one to start programming with.

• Open-sthe ce general-purpose language.
• Object Oriented, Procedural, Functional
• Easy to interface with C/Java/Fortran
• Easy-is to interface with C++ (via SWIG)
• Great interactive environment

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. do not need to compile the program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – This can actually sit at a Python prompt and interact with the interpreter directly to write the 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.

3.5.2 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** − we 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.

3.5.3 Variables

Variables are nothing but reserved memory locations to store values. This means that when create a variable we reserve some space in memory.

Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in that reserved memory. Therefore, by assigning different data types to variables, we can store integers, float, characters etc., in them.

3.5.4 Standard Data Types

The data stored in memory can be of many types. For example, a person's age is stored as a numeric value and his or her address is stored as alphanumeric characters. Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.

Python has five standard data types:

a) Numbers
b) String
c) List
d) Tuple
e) Dictionary

a) **Numbers**

Number data types store numeric values. Number objects are created when it assign a value to them

b) Python Strings

Strings in Python are identified as a contiguous set of characters represented in the quotation marks. Python allows for either pairs of single or double quotes. Subsets of strings can be taken using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.

c) Lists

Lists are the most versatile of Python's compound data types. A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data type. The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. The plus (+) sign is the list concatenation operator, and the asterisk (*) is the repetition operator.

d) Tuples

A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.

The main differences between lists and tuples are: Lists are enclosed in brackets ([ ]) and their elements and size can be changed, while tuples are enclosed in parentheses (( )) and cannot be updated. Tuples can be thought of as **read-only** lists.

e) Python Dictionary

Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python data type, but are usually numbers or strings.

Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).

3.6 Different modes in python

Python has two basic modes: normal and interactive.

The normal mode is the mode where the scripted and finished .py files are run in the Python interpreter.

Interactive mode is a command line shell which gives immediate feedback for each statement, while running previously fed statements in active memory. As new lines are fed into the interpreter, the fed program is evaluated both in part and in whole

Some Python Libraries are listed below,

1) **Pandas**
2) **NumPy**
3) **Matplotlib**
4) **Seaborn**
5) OpenCV
6) Keras
7) TensorFlow
8) NLTK
9) Scikit-Learn
10) SciPy
11) Beautiful Soup
12) Text Blob
13) Pillow
14) Request
15) SQLAlchamy
16) PyTorch

17) Selenium

1) Pandas:

- Pandas provide us with many Series and Data Frames. It allows to easily organize, explore, represent and manipulate data.
- Smart alignment and indexing featured in Pandas offer a perfect organization and data labelling.
- Pandas has some special features that allows to handle missing data or value with a proper measure.
- This package offers us such a clean code that even people with no or basic knowledge of programming can easily work with it.
- It provides a collection of built-in tools that allows to both read and write data in different web services, data-structure and databases as well.
- Pandas can support JSON, Excel, CSV, HDF5 and many other formats. In fact, we can merge different databases at a time with Pandas.

2) NumPy:

- Arrays of NumPy offer modern mathematical implementations on huge amount of data. NumPy makes the execution of these projects much easier and hassle-free.
- NumPy provides masked arrays along with general array objects. It also comes with functionalities such as manipulation of logical shapes, discrete Fthe ier transform, general linear algebra and many more.
- Once the shape of any N-dimensional array changes, NumPy immediately creates new array according to the changes and delete the old ones.
- This package provides useful tools for integration. We can easily integrate NumPy with programming languages such as C, C++ and Fortran code.
- NumPy provides such functionalities that are comparable to MATLAB. They both allow users to get faster with operations.

3) Matplotlib:

- Matplotlib can create such quality figures that are really good for publication. Figures create with Matplotlib are available in hardcopy formats across different interactive platforms.
- It can use Matplotlib with different toolkits such as Python Scripts, Python Shells, Jupiter Notebook and many other for graphical user interfaces.

- A number of third-party libraries can be integrated with Matplotlib applications. Such as seaborn, plot, other projection and mapping toolkits such as base map.
- An active community of developers is dedicated to helping it with any of the inquiries with Matplotlib. Their contribution to Matplotlib is highly praisable.
- Good thing is that can track any bugs, new patches, and feature requests on the issue tracker page from GitHub. It is an official page for featuring different issues related to Matplotlib.

4) Seaborn:

Seaborn is built on top of Python's core visualization library Matplotlib. It is meant to serve as a complement and not a replacement. However, Seaborn comes with some very important features. Let us see a few of them here. The features help in −

- Built in themes for styling matplotlib graphics
- Visualizing univariate and bivariate data
- Fitting in and visualizing linear regression models
- Plotting statistical time series data
- Seaborn works well with NumPy and Pandas data structures
- It comes with built in themes for styling Matplotlib graphics

In most of the cases, one will still use Matplotlib for simple plotting. The knowledge of Matplotlib is recommended to tweak Seaborn's default plots.

5) Scikit-Learn:

- Scikit Learn comes with a clean and neat API. It also provides very useful documentation for beginners.
- It comes with different algorithms – classification, clustering, and regression. It also supports random forests, k-means, gradient boosting, DBSCAN and others
- This package offers easy adaptability. Once we get well with the general functionalities of Scikit Learn, switching to other platforms will be no problem at all.
- Scikit Learn offers easy methods for data representation. Whether one wants to present data as a table or matrix, it is all possible with Scikit Learn.
- It allows the user to explore through digits that are written in hands. The user can not only load but also visualize digits-data as well.

3.7 DJANGO

Django is a Python framework that makes it easier to create web sites using Python.

Django takes care of the difficult stuff so that one can concentrate on building web applications.

3.8 MODULES

a) User Modules: User accounts, groups, permissions and cookie-based user sessions

b) View Home page: Here user view the home page of the sentiment classification web application.

c)3 View Upload page: In the ‘about’ page in the web application, users can learn more about the sentiment prediction.

d) Input Model: The user must provide input values for the certain fields in order to get results.

e) View Results: User view’s the generated results from the model.

f) View score: Here user have ability to view the score in percentage.

3.9 DJANGO INTERFACE FROM END TO END

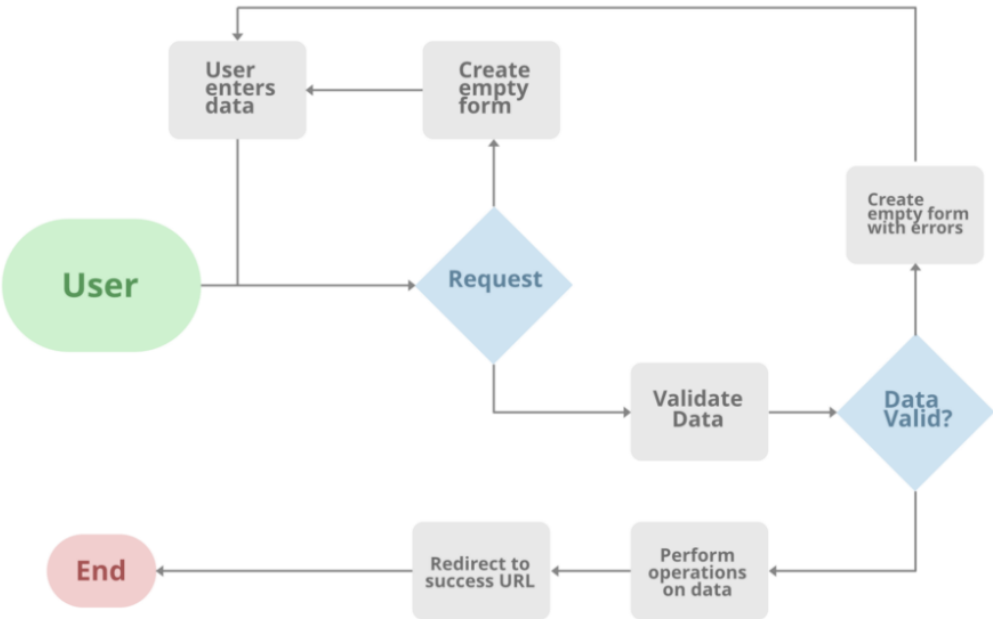


Fig 3.6 Django Interface flow chart

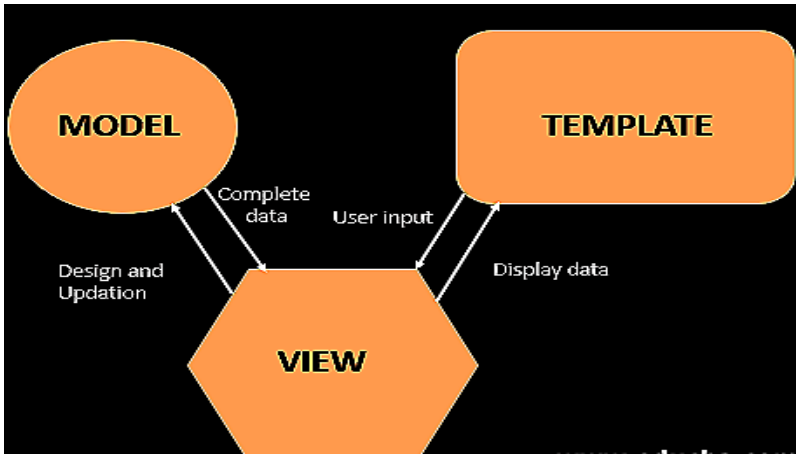


Fig 3.7 Django architecture.

3.10 Creating Django Project:

Before creating a Django project, it has to be installed Django into a development server using the below command.

pip install Django

This will automatically install all the dependencies that the Django framework required in order to function properly.

After installing Django, at command prompt the following commands are used:

1. **cd Desktop**
2. **mkdir local disk**
3. **cd STRESS_DETECTION**
4. **Django-admin start project STRESS DETECTION**
5. **cd STRESS DETECTION**
6. Now it follows the above steps correctly, just type in **python manage.py run server** into a command prompt screen and then copy the link that appear on the command prompt (http://127.0.0.1:8000/) into a web browser.

3.11 GIT & GITHUB

3.11.1 GIT

Git is a distributed version control system that is commonly used for software development but can also be used for other types of projects, including documentation, writing, and more. It allows users to track changes to files over time, collaborate with others, and manage different versions of a project.

Here are some key concepts and commands in Git:

1. Repository: A repository (or "repo") is a collection of files that is stored in a central location, typically on a remote server. Users can clone a repository to create a local copy on their own computer, make changes, and then push those changes back to the remote server.

2. Commit: A commit is a snapshot of the current state of the files in a repository. When we make changes to a file, we can commit those changes to create a new version of the file. Each commit has a unique identifier, called a "hash," which allows us to track changes to the file over time.

3. Branch: A branch is a separate line of development that allows us to work on a new feature or bug fix without affecting the main codebase. It can create a new branch, make changes, and then merge those changes back into the main branch when we are ready.

4. Merge: Merging is the process of combining changes from one branch into another. When wea are ready to add changes from a feature branch back into the main branch, it can merge the changes using the "git merge" command.

5. Pull request: A pull request is a way to propose changes to a repository. When create a pull request, it asking the owner of the repository to review and merge the changes. This can be a helpful way to collaborate with others and ensure that changes are reviewed before they're merged into the main branch.

These are just a few of the key concepts and commands in Git. There are many more, including "git clone" for cloning a repository, "git push" for pushing changes to a remote server, and "git log" for viewing the history of a repository.

3.11.2 GITHUB

GitHub is a web-based platform that allows developers to store, manage, and share their code repositories. It is commonly used for version control and collaborative software development.

GitHub provides a range of features, including:

-Code hosting: GitHub provides a platform for developers to store and share their code repositories, which can be accessed and modified by other developers.

-Version control: GitHub uses the Git version control system to manage changes to code, allowing developers to easily track changes and collaborate on projects.

-Collaboration: GitHub allows developers to work together on code repositories, making it easy to share code and review changes.

-Issue tracking: GitHub provides tools for tracking and managing issues, bugs, and feature requests, making it easy for developers to communicate about and prioritize work.

-Documentation: GitHub provides a platform for documenting code repositories, making it easy for developers to share information about their projects and collaborate on documentation.

GitHub is widely used by developers and organizations across a range of industries and sectors, and is considered one of the most popular and influential platforms for opensource software development.

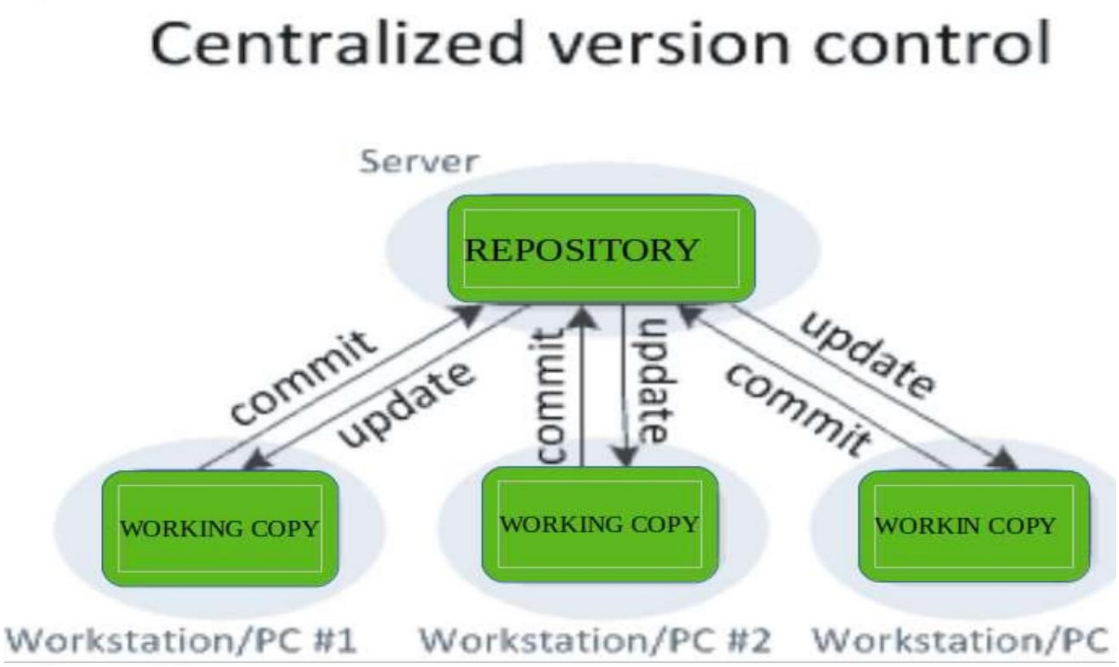


Fig 3.8 Git-Hub Working

3.12 WEB Hosting

Deploying website documentation typically involves the process of publishing the documentation on a web server or hosting service, so that it can be accessed by users. Here are some steps we can follow to deploy website documentation:

Choose a hosting service: There are various hosting services available that can be used to deploy website documentation, including Amazon Web Services (AWS), Microsoft Azure, Google Cloud, or GitHub Pages.

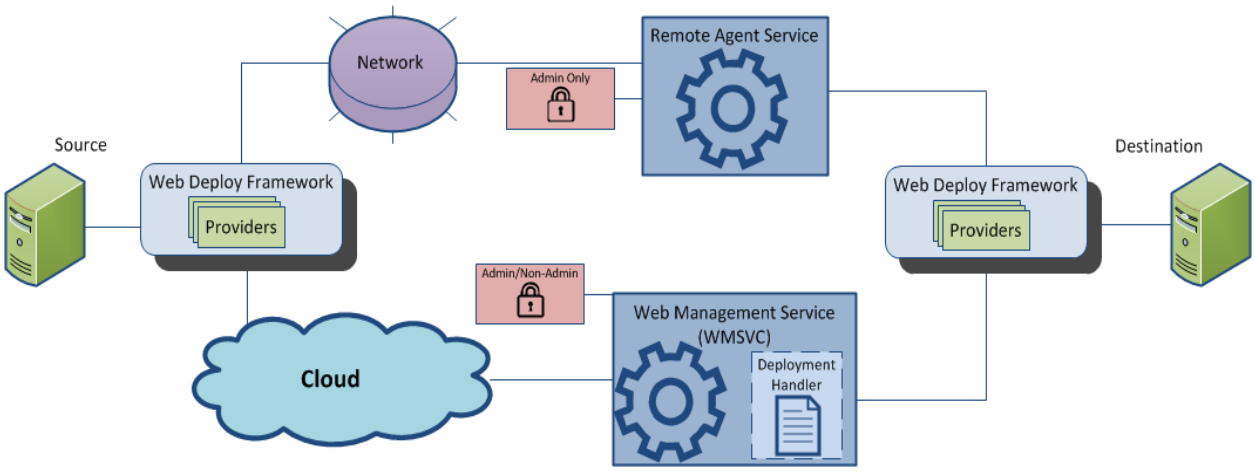


Fig 3.9 Hosting a web application process

Choose a static site generator: If the documentation is in a format that can be converted into HTML, CSS, and JavaScript, we can use a static site generator such as Jekyll, Hugo, or Gatsby to generate static HTML files.

Create a GitHub repository: If we plan to use GitHub Pages as the hosting service, it can create a GitHub repository and push the documentation to the repository. GitHub Pages will automatically deploy the documentation as a static site.

Configure the hosting service: If choose to use a hosting service other than GitHub Pages, it will need to configure the hosting service to deploy documentation. The exact steps will depend on the hosting service we choose.

Test and deploy: Once it have configured the hosting service, test the documentation to ensure it works correctly. Then deploy it to the web server or hosting service, so that it can be accessed by users.

Set up a custom domain (optional): If we want to use a custom domain for the documentation, it can configure the hosting service to use the custom domain. This will involve updating domain's DNS settings to point to the hosting service.

CHAPTER 4

SOFTWARE DEVELOPMENT

In implementing the design steps for further proceeding the with the work following are used as aid, Machine learning algorithms, Django and Creating a website check and to know the present Mental stress .

Machine Learning is said as a subset of **artificial intelligence** that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. The term machine learning was first introduced by **Arthur Samuel** in **1959**. We can define it in a summarized way as:

A machine has the ability to learn if it can improve its performance by gaining more data.

How Machine Learning works

A Machine Learning system **learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it**. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

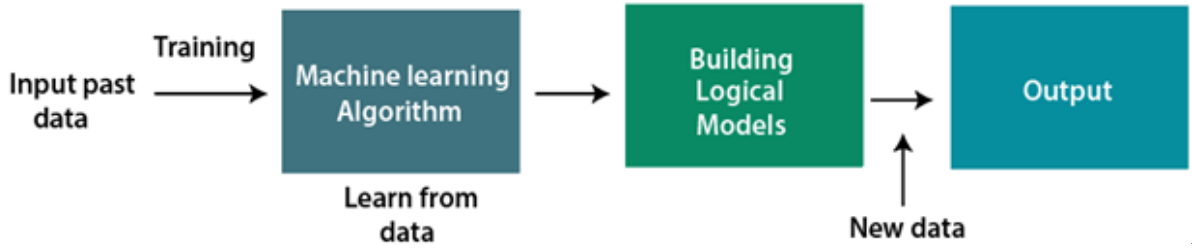


Fig 4.1 Block diagram of Software Deployment

4.1 SOFTWARE INSTALLATION FOR MACHINE LEARNING PROJECT:

Procedure followed for software installation is as follows,

From the Anaconda download page (Anaconda.com/downloads) Anaconda and Jupiter Note book were installed.

2. at the command prompt using ‘pip’ command install the required packages say, NumPy, Pandas, seaborne, scikit learn, matplotlib, pyplot.

Ex: pip install NumPy

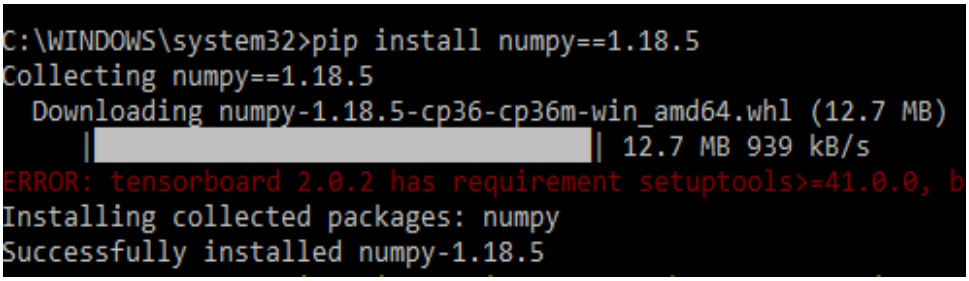


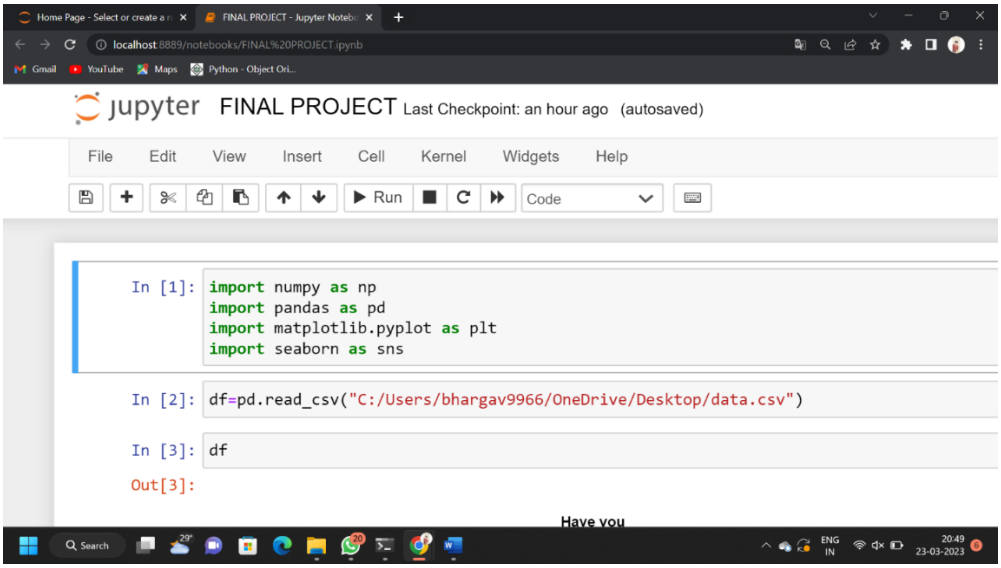
Fig: 4.2 Installation of NumPy

4.2 MODULES

4.2.1 Import Libraries:

Import all required libraries for the processing of data

Libraries that were used in the present work is shown below.



4.2.2 Upload:

Uploading the dataset which was stored in MSEXCEL .

With the help of pandas Library we can read and convert into dataframe

Data frame that was used in the present work is shown below.

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns

In [2]: df=pd.read_csv("C:/Users/bhargav9966/OneDrive/Desktop/data.csv")

In [3]: df

Out[3]:
```

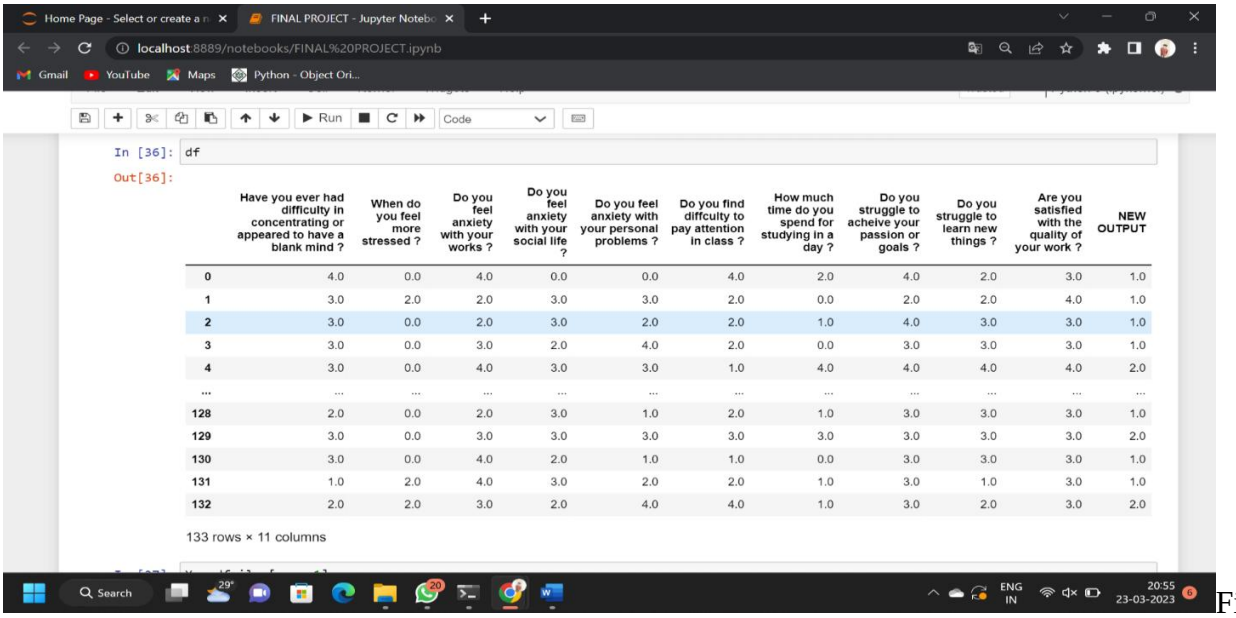
	NAME	AGE	CLASS	BRANCH	Have you ever had difficulty in concentrating or appeared to have a blank mind ?	When do you feel more stressed ?	Do you feel anxiety with your works ?	Do you feel anxiety with your social life ?	Do you feel anxiety with your personal problems ?	Do you find diffculty to pay attention in class ?	How much time do you spend for studying in a day ?	Do you struggle to acheive your passion or goals ?	Do you struggle to learn new things ?	What a Psychol or Em effe stress noticed you
0	Bhargav	22.0	4th	MECH	STRONGLY AGREE	EVENING	STRONGLY AGREE	STRONGLY DISAGREE	STRONGLY DISAGREE	STRONGLY AGREE	2-3 HOURS	STRONGLY AGREE	NEUTRAL	ANXIE
1	Ch. Vineeth	20.0	4th	MECH	AGREE	NIGHT	NEUTRAL	AGREE	AGREE	NEUTRAL	0-1 HOURS	NEUTRAL	NEUTRAL	IRRIT OR A
2	MV SUBASH	21.0	4th	MECH	AGREE	EVENING	NEUTRAL	AGREE	NEUTRAL	NEUTRAL	1-2 HOURS	STRONGLY AGREE	AGREE	FE ISOLATI LONEL
3	Prakash	22.0	4th	MECH	AGREE	EVENING	AGREE	NEUTRAL	STRONGLY AGREE	NEUTRAL	0-1 HOURS	AGREE	AGREE	
4	Manjunath	20.0	4th	MECH	AGREE	EVENING	STRONGLY AGREE	AGREE	AGREE	DISAGREE	4+ HOURS	STRONGLY AGREE	STRONGLY AGREE	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
128	Surendra	21.0	4th	ECE	NEUTRAL	EVENING	NEUTRAL	AGREE	DISAGREE	NEUTRAL	1-2 HOURS	AGREE	AGREE	TIRE
129	N Manjunath	22.0	4th	MECH	AGREE	EVENING	AGREE	AGREE	AGREE	AGREE	3-4 HOURS	AGREE	AGREE	

4.2.3 Pre-processing:

Data Pre-processing is a technique that is used to convert the raw data into a clean data set. Cleaning the data refers to removing the null values, filling the null values with meaningful value, removing duplicate values, removing outliers, removing unwanted attributes. If dataset contains any categorical records means convert those categorical **Pre-processing** variables to numerical values.

Here, the columns of the dataset is adjusted.

Pre-processing did in the present work is shown below.



	Have you ever had difficulty in concentrating or appeared to have a blank mind ?	When do you feel more stressed ?	Do you feel anxiety with your works ?	Do you feel anxiety with your social life ?	Do you feel anxiety with your personal problems ?	Do you find diffculty to pay attention in class ?	How much time do you spend for studying in a day ?	Do you struggle to acheive your passion or goals ?	Do you struggle to learn new things ?	Are you satisfied with the quality of your work ?	NEW OUTPUT
0	4.0	0.0	4.0	0.0	0.0	4.0	2.0	4.0	2.0	3.0	1.0
1	3.0	2.0	2.0	3.0	3.0	2.0	0.0	2.0	2.0	4.0	1.0
2	3.0	0.0	2.0	3.0	2.0	2.0	1.0	4.0	3.0	3.0	1.0
3	3.0	0.0	3.0	2.0	4.0	2.0	0.0	3.0	3.0	3.0	1.0
4	3.0	0.0	4.0	3.0	3.0	1.0	4.0	4.0	4.0	4.0	2.0
...	...	...	...	...	...	...	...	...	...	...	...
128	2.0	0.0	2.0	3.0	1.0	2.0	1.0	3.0	3.0	3.0	1.0
129	3.0	0.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	2.0
130	3.0	0.0	4.0	2.0	1.0	1.0	0.0	3.0	3.0	3.0	1.0
131	1.0	2.0	4.0	3.0	2.0	2.0	1.0	3.0	1.0	3.0	1.0
132	2.0	2.0	3.0	2.0	4.0	4.0	1.0	3.0	2.0	3.0	2.0

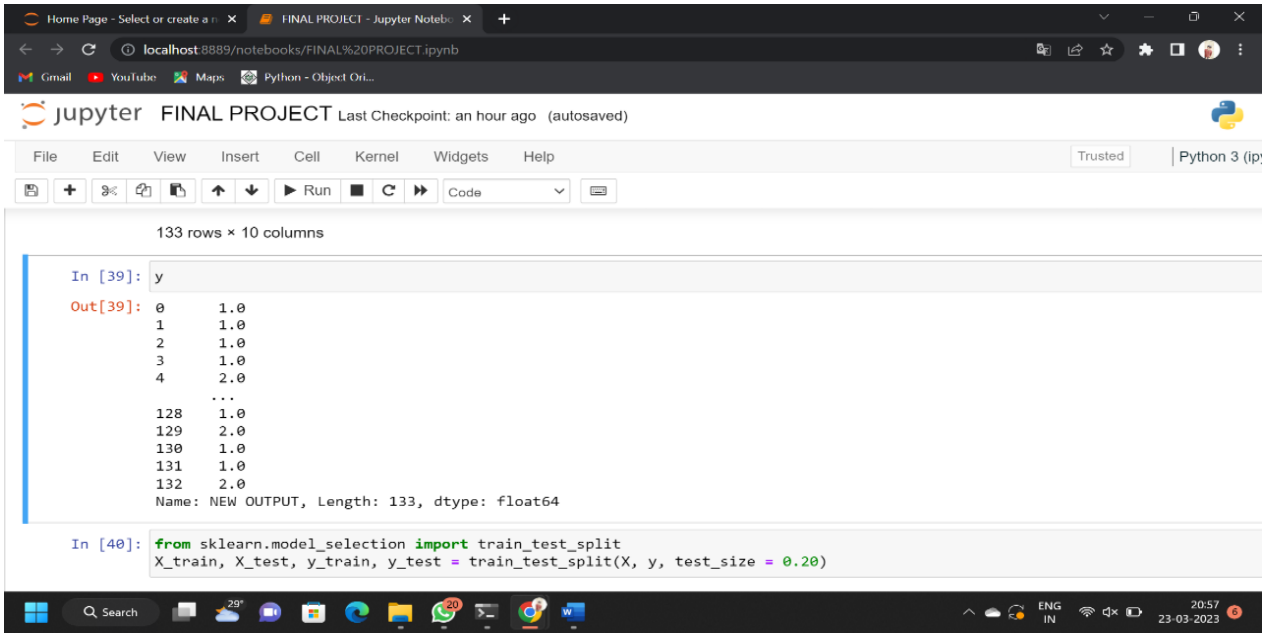
133 rows × 11 columns

Fi

4.2.4 Split dataset:

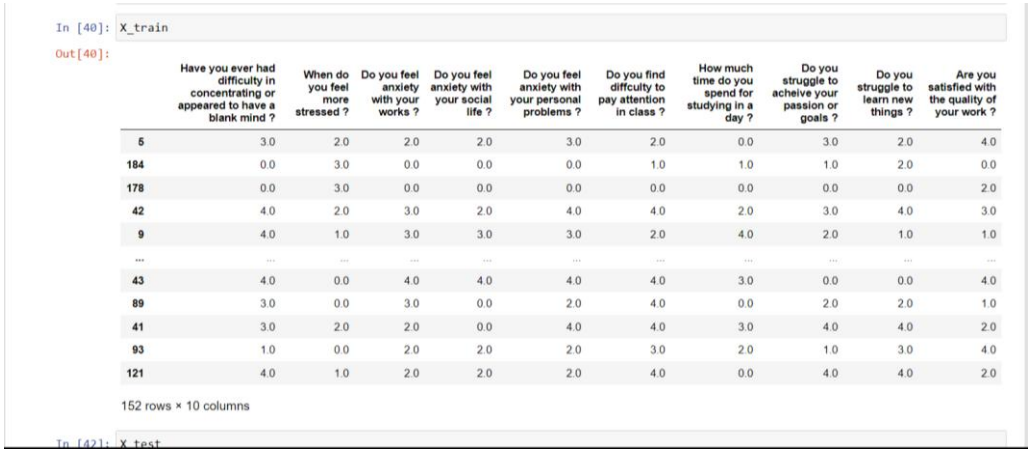
The datasets are split into test and train dataset with a test size of 20%

Splitting of data set that was used in the present work is shown below.



The above code is used to split the dataset into training data set and test data set where we train the data with training data set which is used for predicting the output.

Input that was used in the present work is shown below.



In [40]: X_train

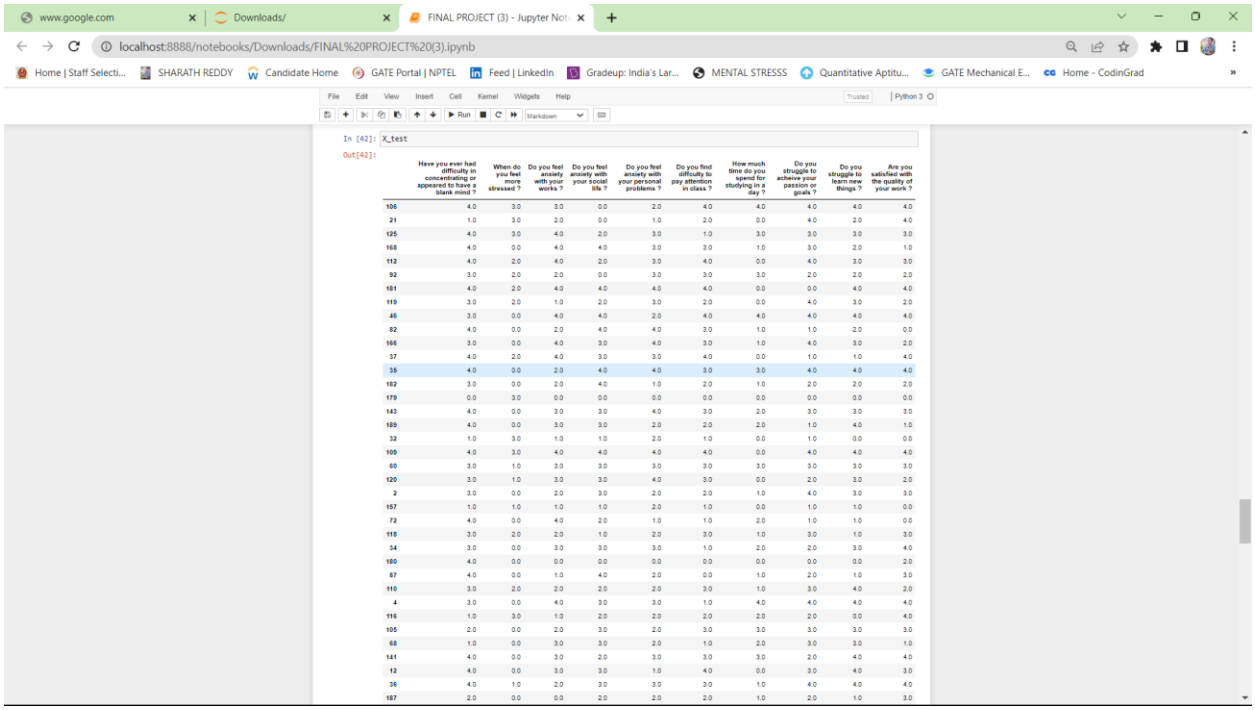
Out[40]:

	Have you ever had difficulty in concentrating or appeared to have a blank mind ?	When do you feel more stressed ?	Do you feel anxiety with your works ?	Do you feel anxiety with your social life ?	Do you feel anxiety with your personal problems ?	Do you find diffculty to pay attention in class ?	How much time do you spend for studying in a day ?	Do you struggle to acheive your passion or goals ?	Do you struggle to learn new things ?	Are you satisfied with the quality of your work ?
5	3.0	2.0	2.0	2.0	3.0	2.0	0.0	3.0	2.0	4.0
184	0.0	3.0	0.0	0.0	0.0	1.0	1.0	1.0	2.0	0.0
178	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0
42	4.0	2.0	3.0	2.0	4.0	4.0	2.0	3.0	4.0	3.0
9	4.0	1.0	3.0	3.0	3.0	2.0	4.0	2.0	1.0	1.0
...	...	...	...	...	...	...	...	...	...	...
43	4.0	0.0	4.0	4.0	4.0	4.0	3.0	0.0	0.0	4.0
89	3.0	0.0	3.0	0.0	2.0	4.0	0.0	2.0	2.0	1.0
41	3.0	2.0	2.0	0.0	4.0	4.0	3.0	4.0	4.0	2.0
93	1.0	0.0	2.0	2.0	2.0	3.0	2.0	1.0	3.0	4.0
121	4.0	1.0	2.0	2.0	2.0	4.0	0.0	4.0	4.0	2.0

152 rows × 10 columns

In [42]: X_test

Input test data that was used in the present work is shown below.

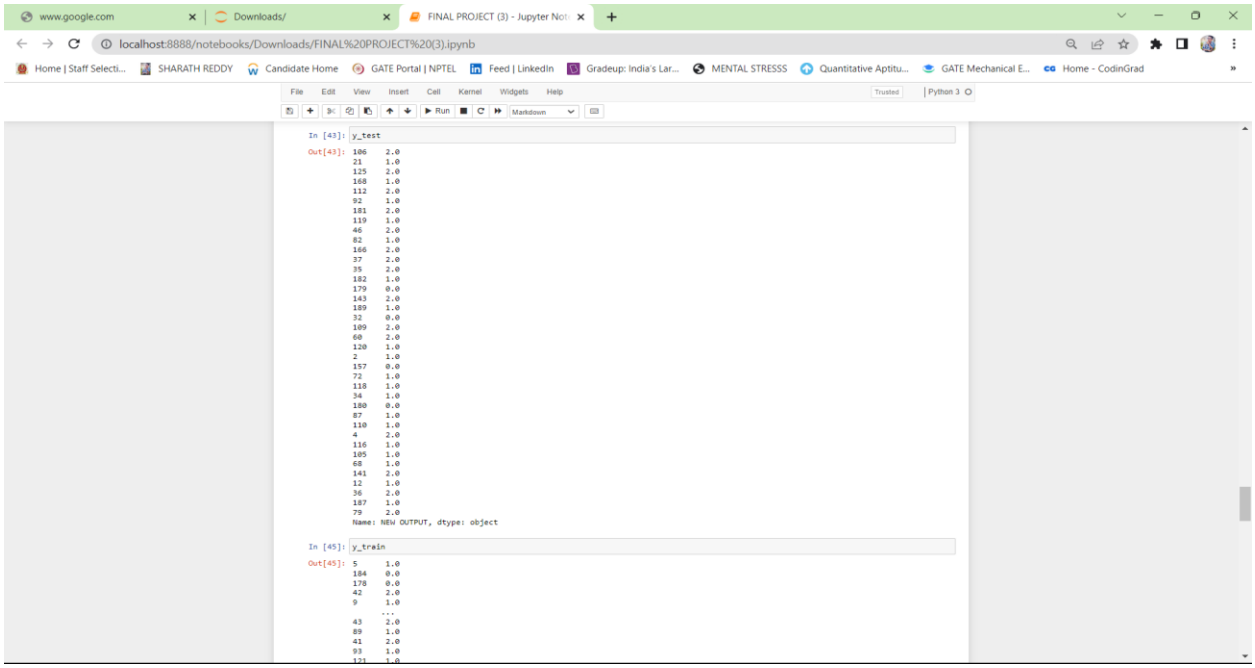


In [42]: X_test

Out[42]:

	Have you ever had difficulty in concentrating or appeared to have a blank mind ?	When do you feel more stressed ?	Do you feel anxiety with your works ?	Do you feel anxiety with your social life ?	Do you feel anxiety with your personal problems ?	Do you find difficulty to pay attention in class ?	How much time do you spend for studying in a day ?	Do you struggle to acheive your passion or goals ?	Do you struggle to learn new things ?	Are you satisfied with the quality of your work ?
106	4.0	3.0	3.0	0.0	2.0	4.0	4.0	4.0	4.0	4.0
21	1.0	3.0	2.0	0.0	1.0	2.0	0.0	4.0	2.0	4.0
125	4.0	3.0	4.0	2.0	3.0	1.0	3.0	3.0	3.0	3.0
168	4.0	0.0	4.0	4.0	3.0	3.0	1.0	3.0	2.0	1.0
112	4.0	2.0	4.0	2.0	3.0	4.0	0.0	4.0	3.0	3.0
92	3.0	2.0	2.0	0.0	3.0	3.0	3.0	2.0	2.0	2.0
181	4.0	2.0	4.0	4.0	4.0	4.0	0.0	0.0	4.0	4.0
119	3.0	2.0	1.0	2.0	3.0	2.0	0.0	4.0	3.0	2.0
46	3.0	0.0	4.0	4.0	2.0	4.0	4.0	4.0	4.0	4.0
82	4.0	0.0	2.0	4.0	4.0	3.0	1.0	1.0	2.0	0.0
166	3.0	0.0	4.0	3.0	4.0	3.0	1.0	4.0	3.0	2.0
37	4.0	2.0	4.0	3.0	3.0	4.0	0.0	1.0	1.0	4.0
35	4.0	0.0	2.0	4.0	4.0	3.0	3.0	4.0	4.0	4.0
182	3.0	0.0	2.0	4.0	1.0	2.0	1.0	2.0	2.0	2.0
179	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
143	4.0	0.0	3.0	3.0	4.0	3.0	2.0	3.0	3.0	3.0
189	4.0	0.0	3.0	3.0	2.0	2.0	2.0	1.0	4.0	1.0
32	1.0	3.0	1.0	1.0	2.0	1.0	0.0	1.0	0.0	0.0
109	4.0	3.0	4.0	4.0	4.0	4.0	0.0	4.0	4.0	4.0
60	3.0	1.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0
120	3.0	1.0	3.0	3.0	4.0	3.0	0.0	2.0	3.0	2.0
2	3.0	0.0	2.0	3.0	2.0	2.0	1.0	4.0	3.0	3.0
157	1.0	1.0	1.0	1.0	2.0	1.0	0.0	1.0	1.0	0.0
72	4.0	0.0	4.0	2.0	1.0	1.0	2.0	1.0	1.0	0.0
118	3.0	2.0	2.0	1.0	2.0	3.0	1.0	3.0	1.0	3.0
34	3.0	0.0	3.0	3.0	3.0	1.0	2.0	2.0	3.0	4.0
180	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0
87	4.0	0.0	1.0	4.0	2.0	0.0	1.0	2.0	1.0	3.0
110	3.0	2.0	2.0	2.0	2.0	3.0	1.0	3.0	4.0	2.0
4	3.0	0.0	4.0	3.0	3.0	1.0	4.0	4.0	4.0	4.0
116	1.0	3.0	1.0	2.0	2.0	2.0	2.0	2.0	0.0	4.0
105	2.0	0.0	2.0	3.0	2.0	3.0	3.0	3.0	3.0	3.0
68	1.0	0.0	3.0	3.0	2.0	1.0	2.0	3.0	3.0	1.0
141	4.0	0.0	3.0	2.0	3.0	3.0	3.0	2.0	4.0	4.0
12	4.0	0.0	3.0	3.0	1.0	4.0	0.0	3.0	4.0	3.0
36	4.0	1.0	2.0	3.0	3.0	3.0	1.0	4.0	4.0	4.0
187	2.0	0.0	0.0	2.0	2.0	2.0	1.0	2.0	1.0	3.0

Output train data set that was used in the present work is shown below and Output test data set that was used in the present work is shown below.



4.2.5 The models:

- Decision Tree:
  - A Decision tree is a flowchart like tree structure.
  - Each internal node denotes a test on an attribute.
  - Each branch represents an outcome of the test.
  - Each leaf node (terminal node) holds a class label.
  - An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute.
  - Gini method or Information Gain (entropy) is used for attribute selection.
- Random Forest:
  - Random forest, like its name implies, consists of a large number of individual decision trees that operates as an ensemble.
  - Each individual tree in the random forest generates a class prediction.
  - The class with the most votes become the model's prediction.
  - By using many trees as an ensemble, it reduces overfitting.
  - It can be used for both classification and regression but is mostly preferred for classification.

4.2.6 Visualizations:

Each dataset is trained on different machine learning algorithms and their respective accuracies are recorded. These accuracies are then used for comparison between the models. The below shows the matplotlib code and plot the diagram.

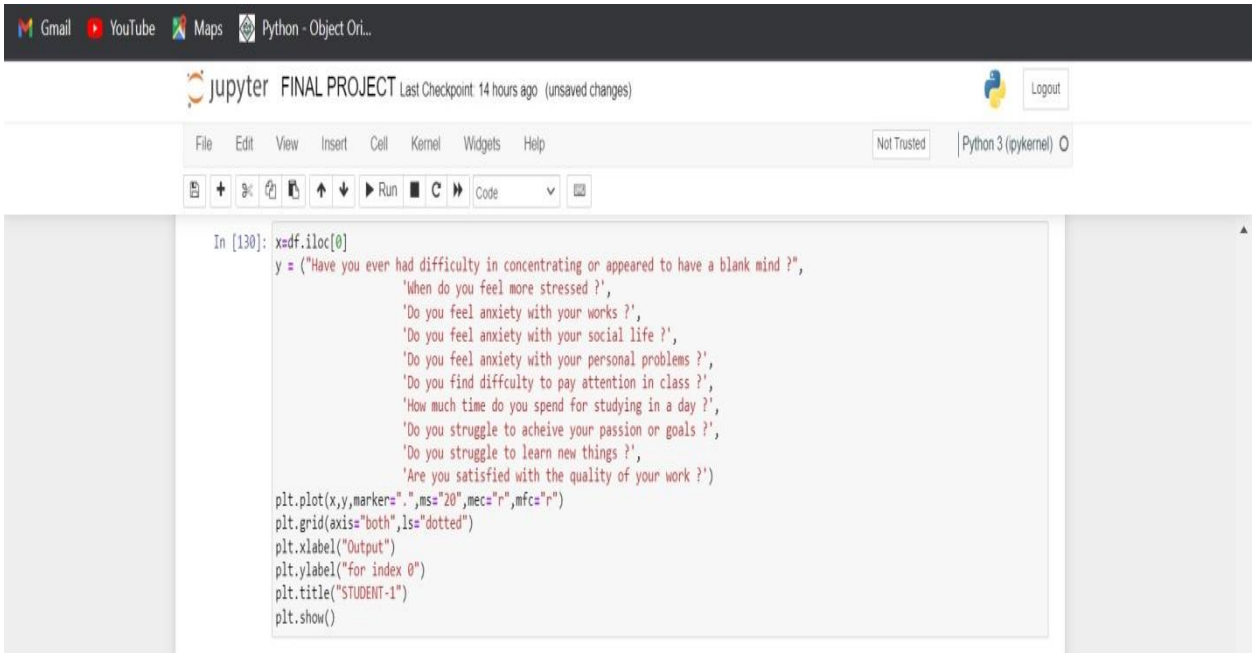


Fig 4.3 Visualizations of data

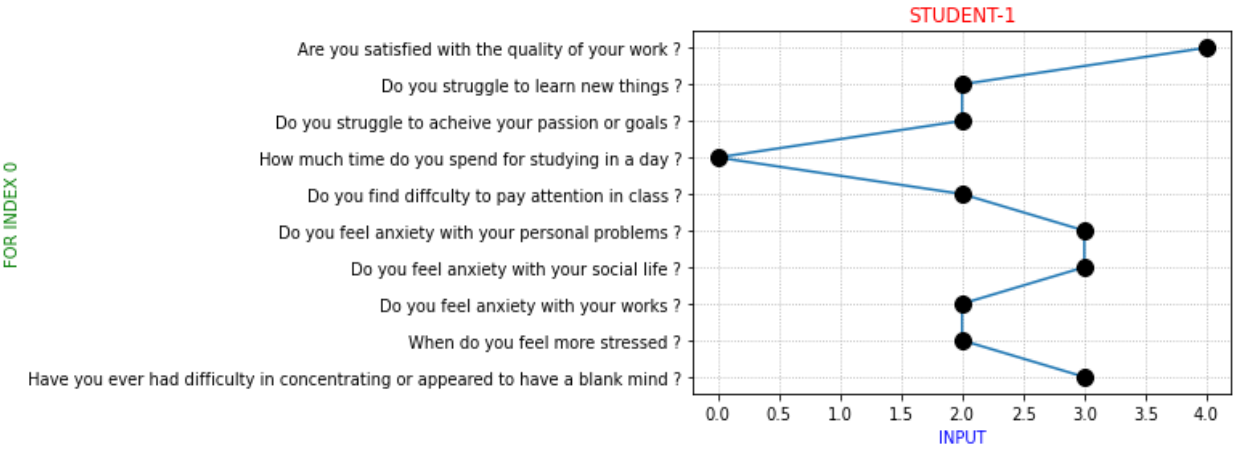


Fig 4.4 Responses of student 1

Let us consider two categories i.e., category A and category B. It is desired to classify which category the new datapoint lies in. To classify this, one can use the KNN algorithm which observes the behaviour of the nearest points and classify it accordingly. In this case, in order to classify the behaviour is which category it belongs to, let us consider an XY plane with data points plotted in the graph fig 4.8.

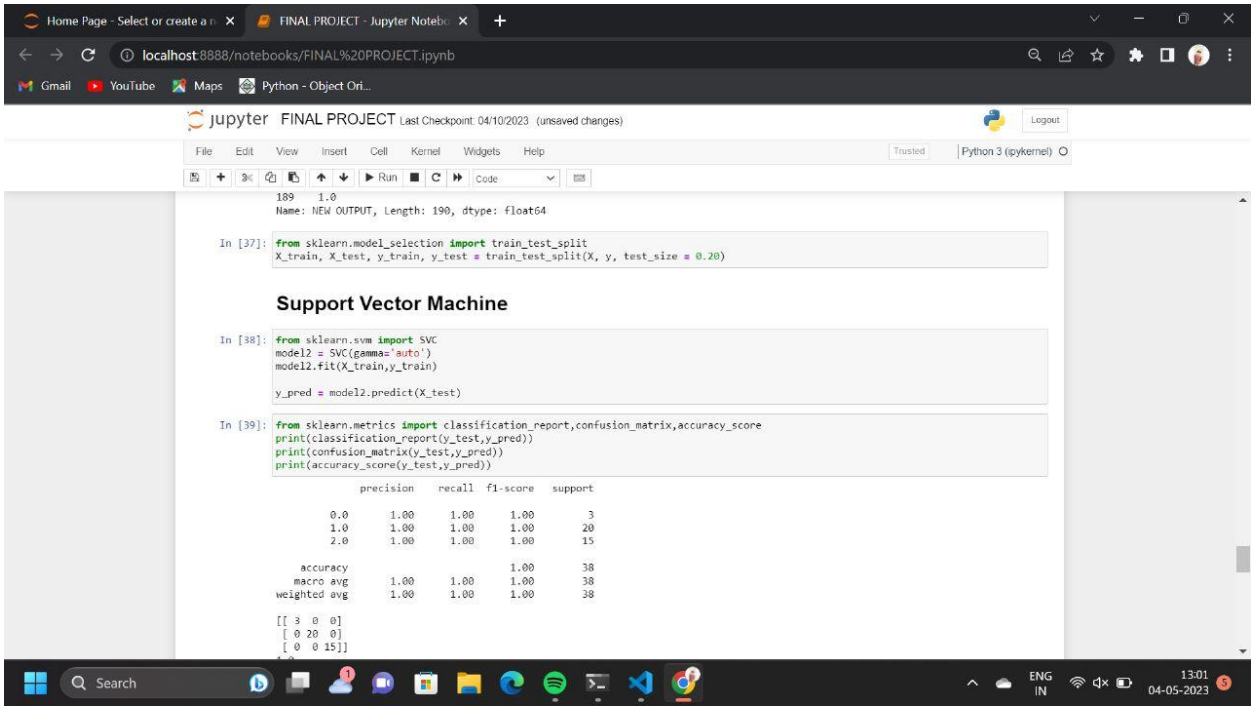


Fig 4.5 SVM algorithm applied to the dataset collected.

4.3 DJANGO

Django is a Python framework that makes it easier to create web sites using Python.

Django takes care of the difficult stuff so that it can concentrate on building web applications.

Django emphasizes reusability of components, also referred to as DRY (Don't Repeat Yourself), and comes with ready-to-use features like login system, database connection and CRUD operations (Create Read Update Delete).

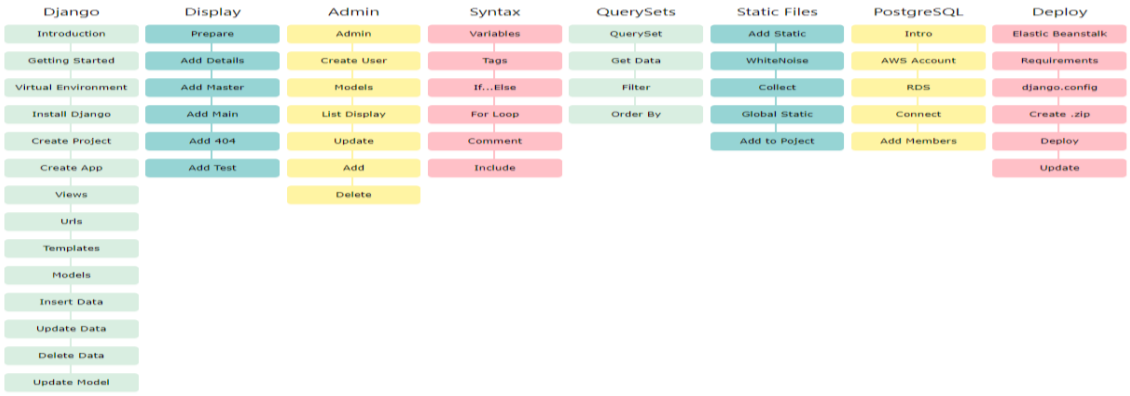


Fig 4.6 Django flow chart

Django installed already. It can tell Django is installed and which version by running the following command in a shell prompt

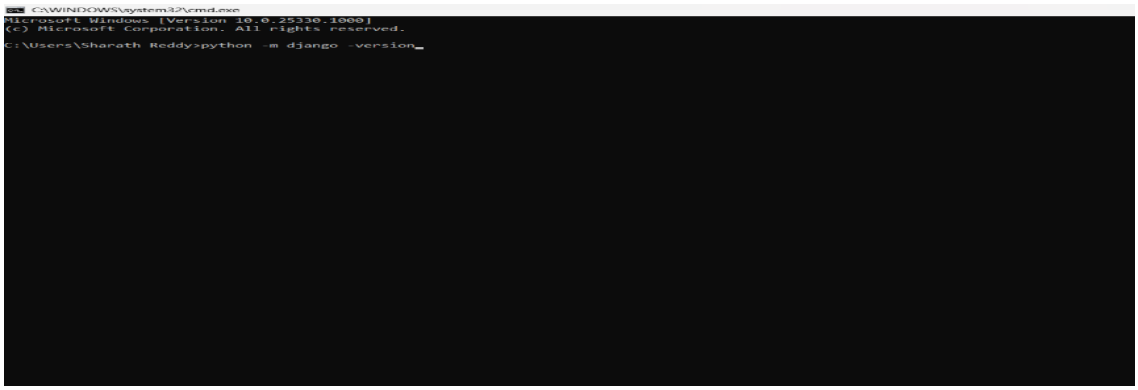


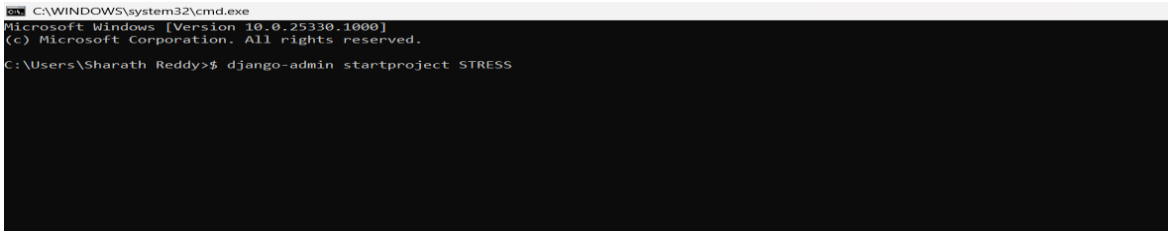
Fig 4.7 Django version

4.3.13 python -m Django –version

Creating a project

First-time using Django, it has to make some initial setup. Namely, we will need to auto-generate some code that establishes a Django project – a collection of settings for an instance of Django, including database configuration, Django-specific options and application-specific settings.

From the command line, **cd** into a directory where we like to store the code, then run the following command:



|Fig 4.8 creating a Django project

4.3.14 Django-admin start project STRESS

This will create a **STRESS** directory in the current directory. If it didn't work, see Problems running Django-admin.

Put the code in some directory **outside** of the document root, such as **/home/STRESS**.

Let's look at what **start project** created:

```
STRESS/
    manage.py
    STRESS/
        __init__.py
        settings.py
```

urls.py

asgi.py

wsgi.py

These files are:

- The outer stress (root) directory is a container for the project. ‘manage.py’ is a command-line utility that lets interact with this Django project in various ways. It can read all the details about manage.py. The inner **stress directory in the Stress** directory is the actual Python package for the project. This is a Python package this need to be imported. (e.g., **stress. URLs**).
- **stress/__init__.py**:
- **stress/settings.py**: Settings/configuration for this Django project.
- **stress/urls.py**: The URL declarations for the Django project; a “table of contents” of the Django-powered site.
- **stress/asgi.py**: An entry-point for ASGI-compatible web servers to serve the project.
- **stress/wsgi.py**: An entry-point for WSGI-compatible web servers to serve the project.

The development server

Change into the outer **stress** directory, if this haven’t already, and run the following commands:

Command1. python manage.py run server

The following output is seen at the command line:

Performing system checks...

System check identified no issues (0 silenced).

You have unapplied migrations; the app may not work properly until they are applied.

Run 'python manage.py migrate' to apply them.

April 10, 2023 - 15:50:53

Django version 4.2, using settings 'stress. Settings'

Starting development server **http://127.0.0.1:8000/**

Quit the server with CONTROL-C.

This is shown in the figs 4.13, 4.14.

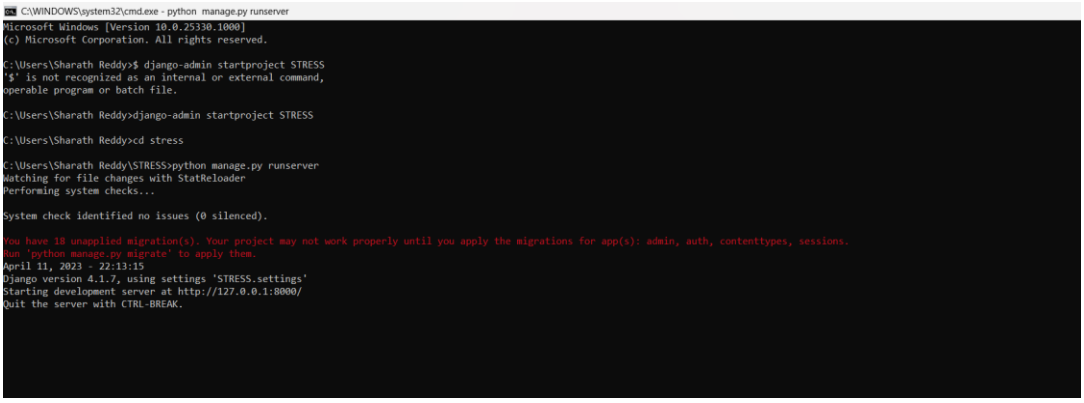


Fig 4.9 local host server

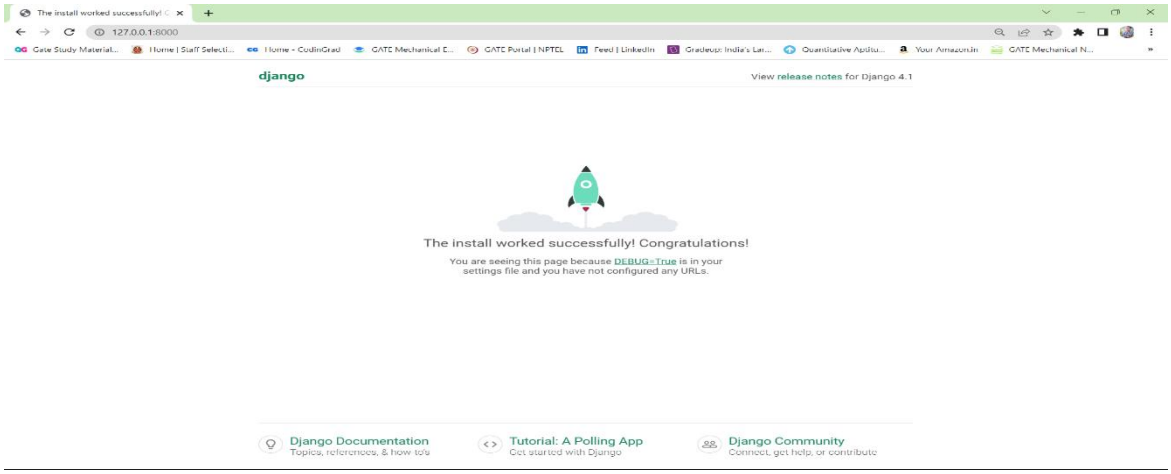


Fig 4.10 Launching or hosting the local server

4.4 Creating the Stress detection app

From this ,we have created the stress-detection app in the same directory as the **manage.py** file so that it can be imported as its own top-level module, rather than a submodule of **stress**.

To create the app, make sure we are in the same directory as **manage.py** and type this command:

python manage.py start app stress detection

That'll create a directory **stress detection**, which is laid out like this:

Stress detection/

__init__.py

admin.py

apps.py

migrations/

__init__.py

models.py

tests.py

views.py

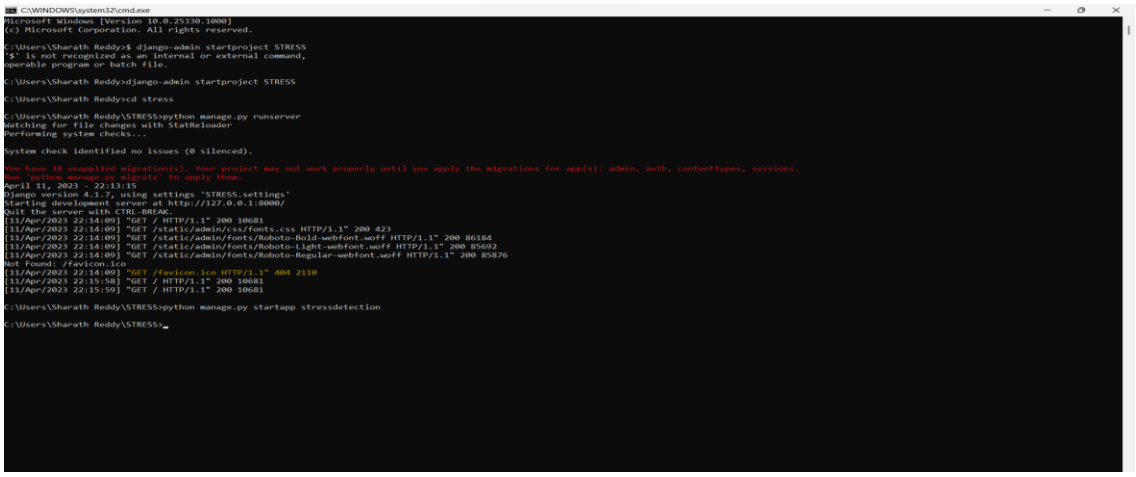


Fig 4.11 Creating Django app

Write the first view

Open the file **stressdetection/views.py** following command has been used :

from Django. http import HTTP Response

def index(request):

return Http Response("Hello, world. You're at the stress detection index.")

Views.py code that was used in the present work was shown below.

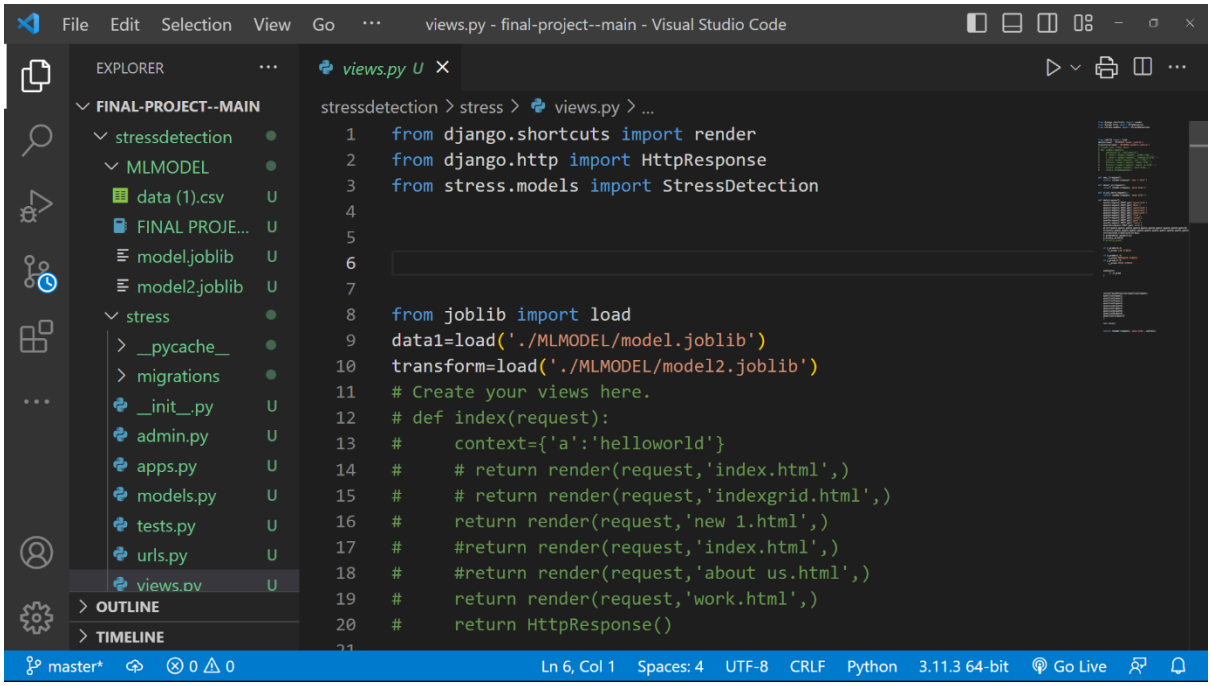


Fig 4.12 Views.py

To create a URL conf in the stress detection directory, create a file called **urls.py**. the app directory should now look like:

In the **stressdetection/urls.py** file include the following code:

Stress detection/urls.py

from Django. URLs import path

from . import views

URL patterns = [

path("", views. Index, name="index"),

] In **stress/urls.py**, add an import for **django.urls.include** and insert an **include()** in the **URL patterns** list, it can command:

stress/urls.py

from django.contrib import admin

from Django. URLs import include, path

URL patterns = [

path("stress detection/", include("stressdetection.urls")),

path("admin/", admin.site.urls),

urls.py code that was used for the present work was shown below

the command can be seen in Figures 4.21& 4.22

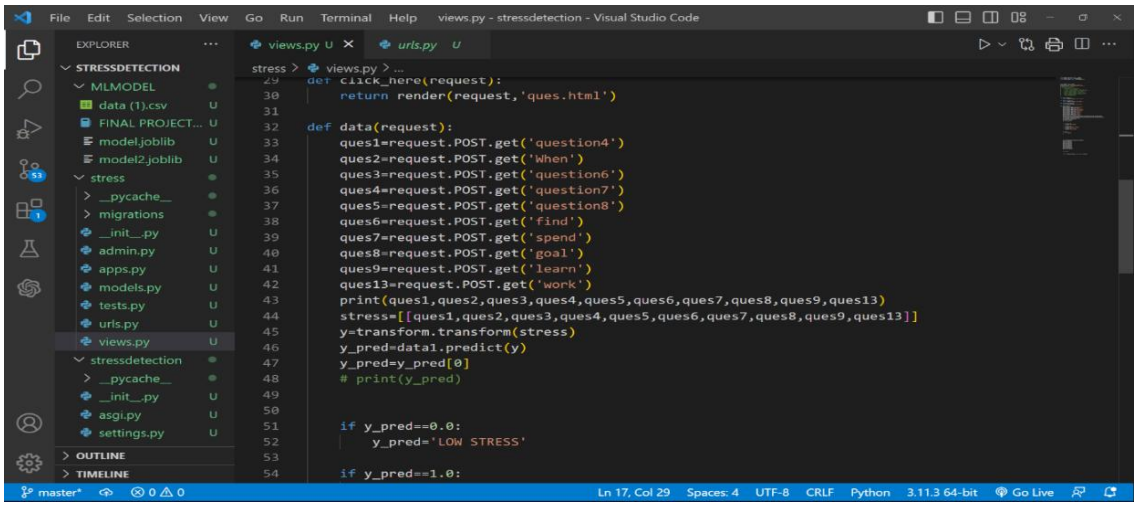


Fig 4.13 (a) views.py

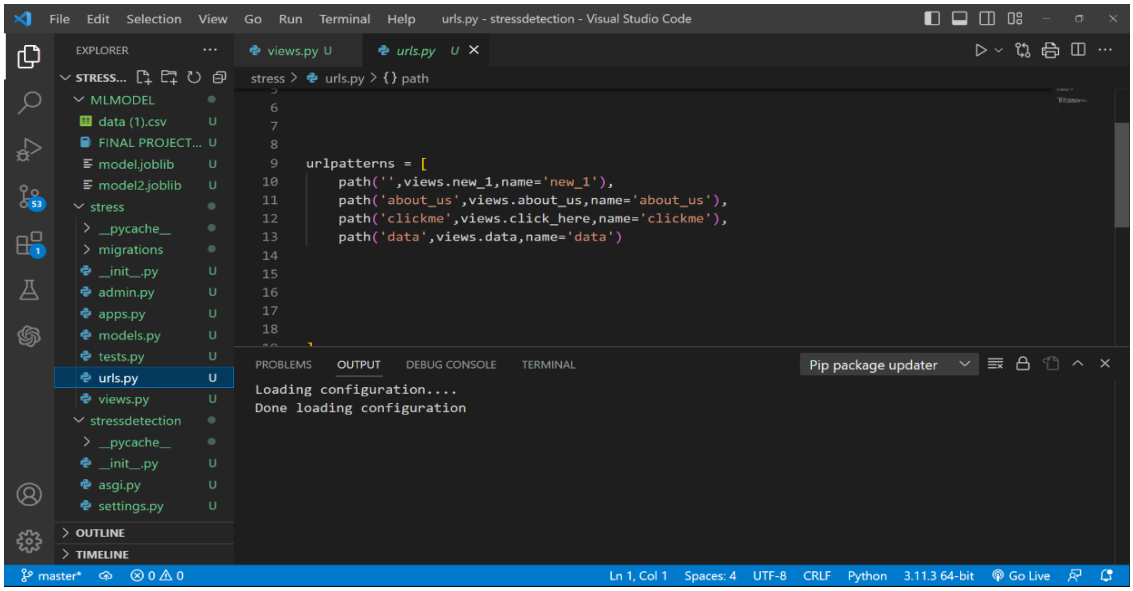


Fig 4.14 urls.py

4.4.6 python manage.py run server

Go to http://localhost:8000/stressdetection/ in the browser, and we can see the text "*Hello, world. You're at the stress detection index.*",

4.5 Django Models

Output has been static data from Python or HTML templates.

Now we will see how Django allows us to work with data, without having to change or upload files in the process.

In Django, data is created in objects, called Models, and is actually tables in a database.

Create Table (Model)

To create a model, navigate to the models.py file in the /members/ folder.

To create a model, navigate to the models.py file in the/members/folder. Django Models.py code is shown below.

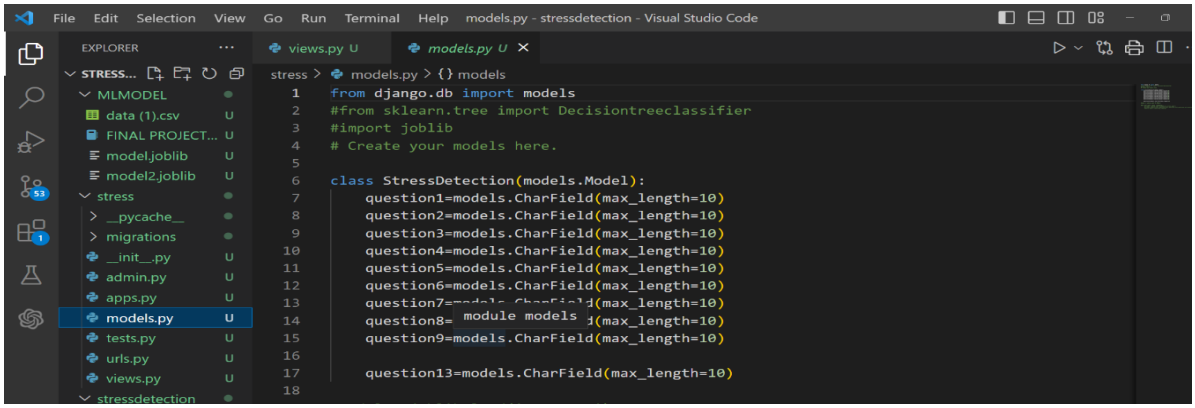


Fig 4.15 models.py

Django templates

4.6HTML

HTML, or Hypertext Markup Language, is a markup language used to create web pages. It provides the structure and content of a web page, including text, images, and other multimedia elements. In this document, we will provide an basics of HTML, including its syntax, basic tags, and some best practices for writing HTML code.

HTML Syntax HTML documents are made up of elements, which are defined using tags. Tags are enclosed in angle brackets, and can contain attributes that modify the behaviour or appearance of the element. The basic syntax of an HTML element is as follows:

<tag name attribute1="value1" attribute2="value2">content</tag name>

The opening tag contains the name of the element, along with any attributes, and is enclosed in angle brackets. The content of the element is placed between the opening and closing tags. The closing tag has the same name as the opening tag, but is preceded by a forward slash (/) to indicate that it is the closing tag.

Basic HTML Tags There are many HTML tags available for use, but some of the most commonly used tags include:

<html>: The top-level element in an HTML document, which contains all other elements.

<head>: Contains meta information about the document, such as the title and links to stylesheets and scripts.

<title>: Defines the title of the document, which is displayed in the browser's title bar.

<body>: Contains the main content of the document, such as text, images, and multimedia elements.

<h1>, <h2>, <h3>, etc.: Heading tags, which define the hierarchy of headings in the document.

<p>: Defines a paragraph of text.

<image>: Defines an image element, which can be used to display images on the page.

<a>: Defines a hyperlink, which can be used to link to other pages or content on the web.

, <ol>, <li>: List tags, which can be used to create bulleted or numbered lists.

4.7CSS

CSS (Cascading Style Sheets) is used to apply styles to web pages. Cascading Style Sheets are fondly referred to as CSS. It is used to make web pages presentable. The reason for using this is to

simplify the process of making web pages presentable. It allows us to apply styles on web pages. More importantly, it enables us to do this independently of the HTML that makes up each web page. Tags for formatting a web page were never intended in HTML. HTML was established to define a web page's content. The addition of tags like <font> and color attributes to HTML created a big problem for web developers. The creation of large websites, where fonts and color information were added to each page, became a time-consuming and costly procedure. CSS was established to address this issue. CSS eliminated the HTML page's style formatting.

The selector points to the HTML element that we want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

Example

In this example all <p> elements will be center-aligned, with a red text color:

```
p {
color: red;
text-align: center;
}
```

Example Explained

- p is a selector in CSS (it points to the HTML element where we want to style: <p>).
- color is a property, and red is the property value
- text-align is a property, and center is the property value

4.8.1 CSS Selectors

CSS selectors are used to "find" (or select) the HTML elements where we want to style.

We can divide CSS selectors into five categories:

- Simple selectors (select elements based on name, id, class)
- Combinator selectors (select elements based on a specific relationship between them)
- Pseudo-class selectors (select elements based on a certain state)
- Pseudo-elements selectors (select and style a part of an element)
- Attribute selectors (select elements based on an attribute or attribute value)

This page will explain the most basic CSS selectors.

The CSS element Selector

The element selector selects HTML elements based on the element name

4.8.2 Example

Here, all <p> elements on the page will be center-aligned, with a red text color:

P {

text-align: center;

color: red;

The CSS id Selector

The id selector uses the id attribute of an HTML element to select a specific element.

The id of an element is unique within a page, so the id selector is used to select one unique element!

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

4.8.3 Example

The CSS rule below will be applied to the HTML element with id="paral":

#paral {

text-align: center;

color: red;

}

The CSS class Selector

The class selector selects HTML elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the class name.

4.8.4 Example

In this example all HTML elements with class-'center" will be red and center-aligned: center {

text-align: center;

color: red;

}

4.8.5 CSS Padding

The CSS padding properties are used to generate space around an element's content, inside of any defined borders.

With CSS, we have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).

Padding - Individual Sides

CSS has properties for specifying the padding for each side of an element:

- padding-top
- padding-right

- padding-bottom
- padding-left

Main interface of the website

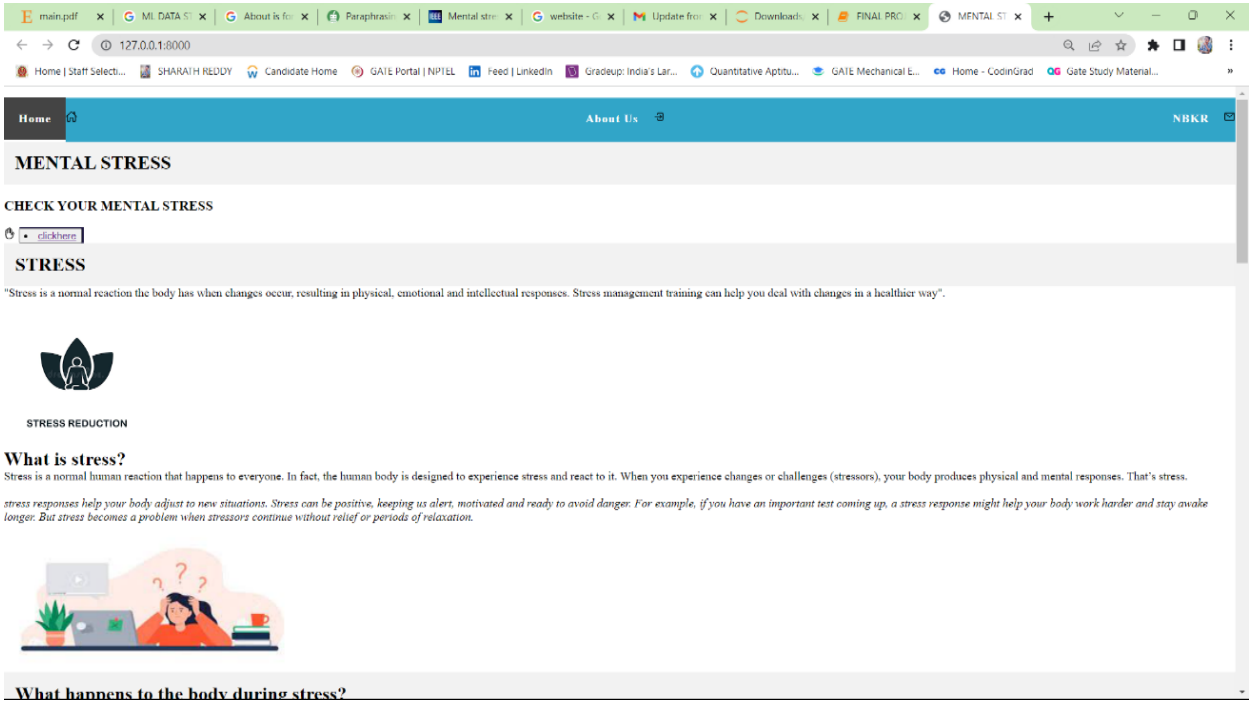


Fig: 4.16 Home page

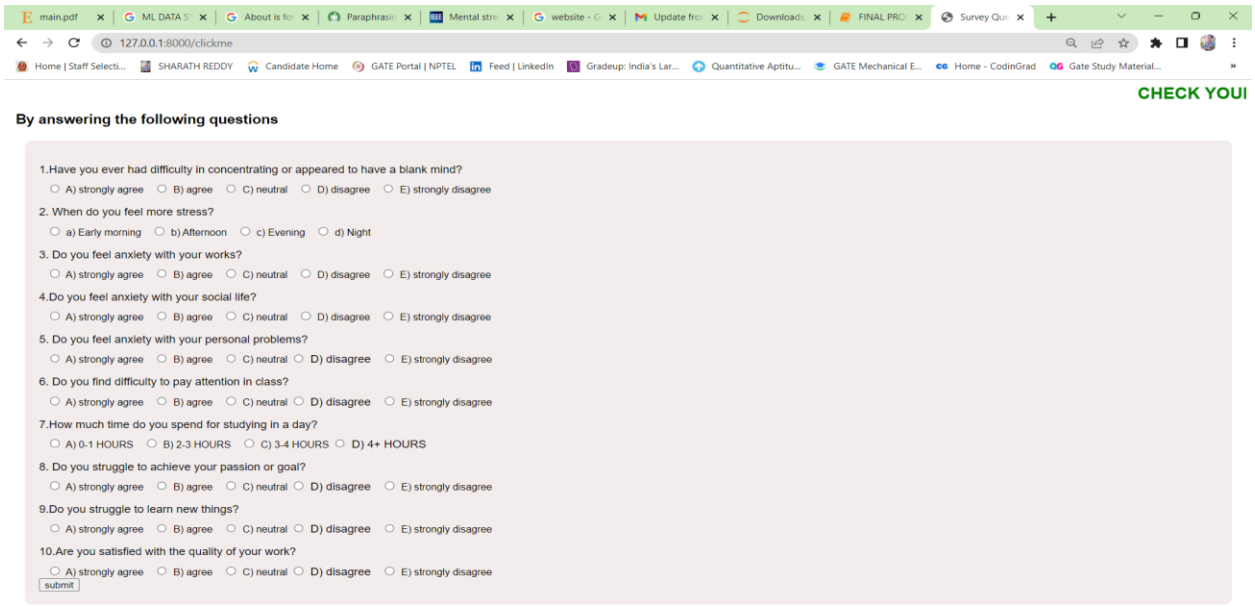


Fig 4.17 questions page

4.8 DEPLOYMENT

Git and Git-hub

Step 1: Create a local git repository

To initialize a git repository in the root of the folder, run the git in it command:

Step 2: Add a new file to the repo

Add a new file to the project, using any text editor like or running a touch command. `touch newfile.txt` just creates and saves a blank file named newfile.txt.

After creating the new file, It can use the **git status** command to see which files git knows exist.

Step 3: Add a file to the staging environment

Add a file to the staging environment using the git add command.

Step 4: Create a commit

Run the command git commit -m "the message about the commit"

Step 5: Create a new branch

Say we want to make a new feature but are worried about making changes to the main project while developing the feature. This is where git branches come in.

Branches allows us to move back and forth between 'states' of a project. Official git docs describe branches this way: ‘A branch in Git is simply a lightweight movable pointer to one of these commits.’ For instance, if they want to add a new page to the website can create a new branch just for that page without affecting the main part of the project. Once it is done with the page, we can merge the changes from the branch into the primary branch. When we create a new branch, Git keeps track of which commit the branch 'branched' off of, so it knows the history behind all the files

Step 6: Run git checkout -b <my branch name>.

This command will automatically create a new branch and then 'check you out' on it, meaning git will move us to that branch, off of the primary branch.

After running the above command, we can use the **git branch** command to confirm that the branch was created.

By default, every git repository’s first branch is named `master` (and is typically used as the primary branch in the project).

Step 7: Create a new repository on GitHub

If it only want to keep track of the code locally, we don't need to use GitHub. But if we want to work with a team, we can use GitHub to collaboratively modify the project's code.

To create a new repo on GitHub, log in and go to the GitHub home page. We can find the “New repository” option under the “+” sign next to the profile picture, in the top right corner of the navbar.

Step 8: Push a branch to GitHub

Now we'll push the commit in the branch to the new GitHub repo. This allows other people to see the changes have made. If they're approved by the repository's owner, the changes can then be merged into the primary branch.

To push changes onto a new branch on GitHub, It has to run git push origin the branch name. GitHub will automatically create the branch for on the remote repository

Step 9: Create a pull request (PR)

A pull request (or PR) is a way to alert a repo's owners that we want to make some changes to their code. It allows them to review the code and make sure it looks good before putting the changes on the primary branch.

Step 10: Merge a PR

This will merge the changes into the primary branch.

Step 11: Get changes on GitHub back to the computer

Right now, the repo on GitHub looks a little different than what we have on the local machine. For example, the commit we have made in the branch and merged into the primary branch doesn't exist in the primary branch on the local machine.

In order to get the most recent changes that others have merged on GitHub, use the git pull origin master command (when working on the primary branch). In most cases, this can be shortened to “git pull

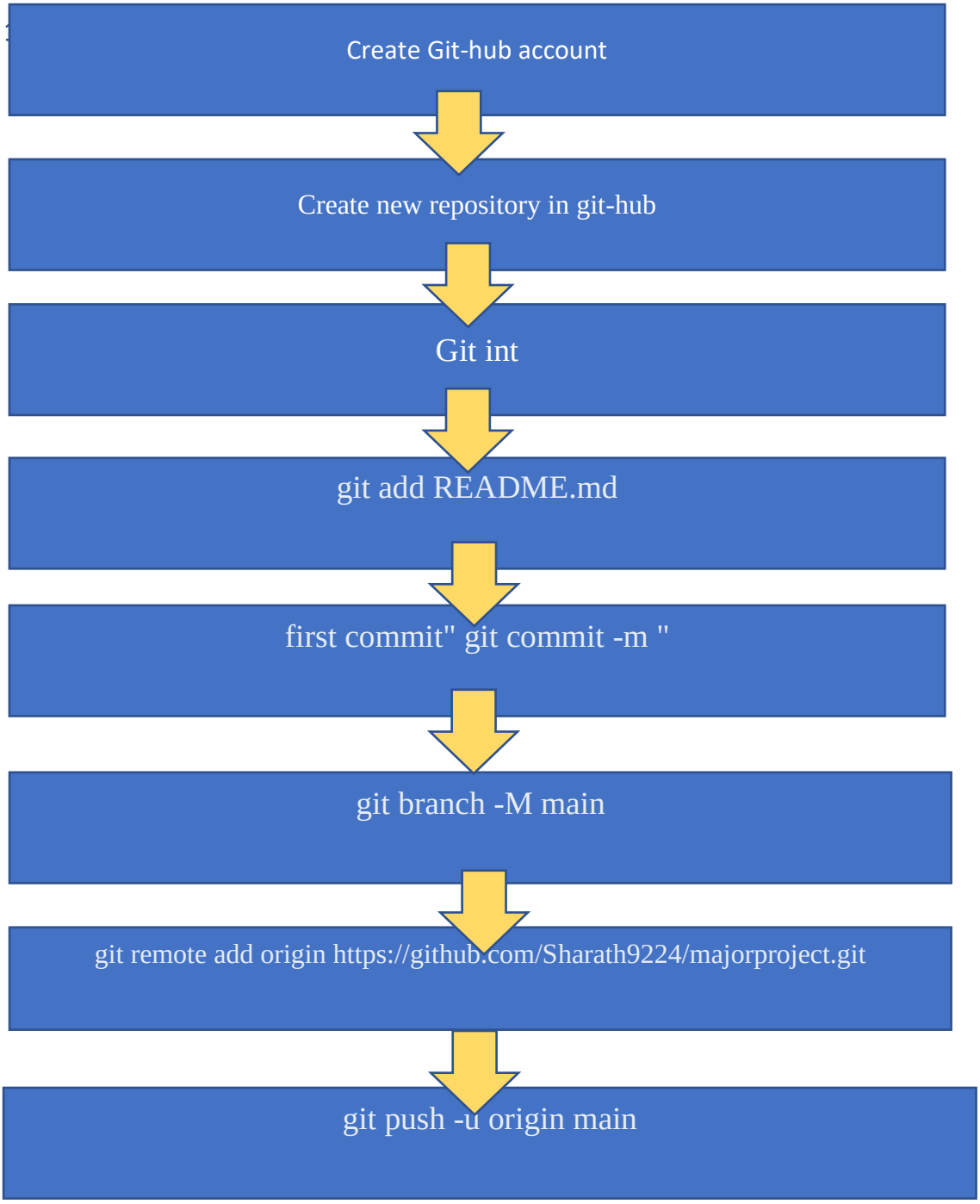


Fig 4.18 Git process flow chart

Introduction

Railway App is a web application designed to simplify the process of booking and managing train tickets. It is built using the MEAN stack (MongoDB, Express.js, AngularJS, and Node.js) and can be deployed on a variety of platforms, including Linux and Windows.

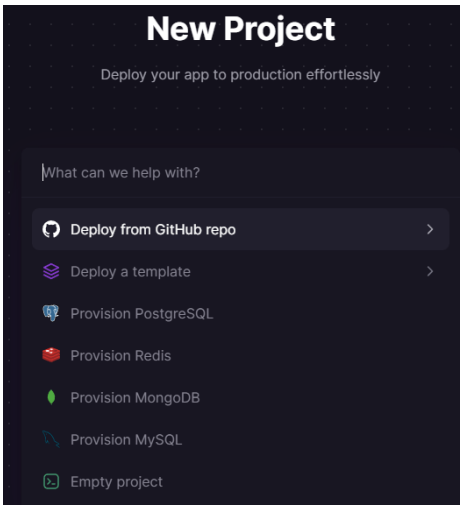


Fig 4.19 Deployment

System Requirements

To deploy Railway App, we will need the following software:

Node.js: Railway App is built using Node.js, so we will need to install Node.js on the system.

MongoDB: Railway App uses MongoDB as its database. we will need to install MongoDB on the system.

Git: Git is a version control system that is used to manage the source code of Railway App. we will need to install Git on the system.

Web server: This will need a web server to host the application. So that we can use Apache or Nginx as the web server.

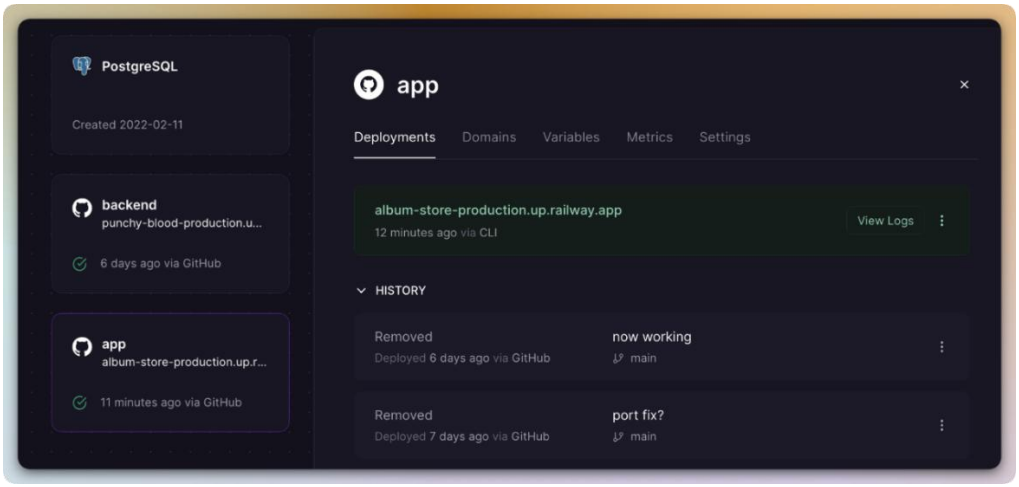


Fig :4.20 Deployment status page

Deployment Steps Follow the below steps to deploy Railway App:

Step 1: Clone the Repository Clone the Railway App repository from GitHub using the following command:

git clone https://github.com/username/RailwayApp.git

Step 2: Install Dependencies Navigate to the project directory and install the required dependencies using the following command:

npm install

Step 3: Configure the Application Configure the application by creating a .env file in the project directory and adding the following details:

PORT=3000 MONGODB_URI=mongodb://localhost:27017/railway

SECRET_KEY= my secret key

Step 4: Start the Application Start the application by running the following command:

npm start

The application will be available at

https://stressdetection.up.railway.app/

CHAPTER 5
PROGRAM DEVELOPED

5.1 Program 1

```
{% load static %}
<!DOCTYPE html>
<html>
<head>
<title>MENTAL STRESSS  </title>
<style>
                        </style>
                        <p id="current Time"></p>
</head>
<!--img src="{% static 'stress/NBKR.jpg' %}"alt=" Image" class="post-image" width="1100px"
height="200px" -->
                        <!--script >document.getElementById('currentTime').innerHTML = new
Date().toUTCString();
}
</script-->
<style>
                                body {
                                        margin: 0;
                                        padding: 0;
                                }
                                nav {
                                        background-color: #31A5C8;
                                        height: 60px;
                                        display: flex;
                                        align-items: center;
                                }

                                nav ul {
                                        margin: 0;
```

```
        padding: 0;
        list-style-type: none;
        display: flex;
        flex-grow: 1;
        justify-content: space-between;
}
nav li {
        margin: 0;
        padding: 0;
        display: inline-block;
}
nav a {
        display: inline-block;
        color: #fff;
        text-align: center;
        padding: 20px;
        text-decoration: none;
        transition: all 0.3s ease;
        font-size: 16px;
        font-weight: bold;
        letter-spacing: 1px;
        border-bottom: 3px solid transparent;
}
nav a:hover {
        background-color: #444;
        border-bottom: 3px solid #fff;
}
nav a.active {
        background-color: #444;
        border-bottom: 3px solid #fff;
}
.home {
        background-image: url("home-icon.png");
```

```
                    }

                    .about {
                        background-image: url("about-icon.png");
                    }
                    .services {
                        background-image: url("services-icon.png");
                    }
                    .contact {
                        background-image: url("contact-icon.png");
                    }
                    body {
                        background-image: url('C:\Users\Sharath
Reddy\Downloads.jpg');
                        background-size: cover;
                        background-repeat: no-repeat;
                    }
                </style>
</head>
<body>
<style>
<h1><center> MENTAL STRESS </CENTER> </h1>
<STYLE>
</STYLE>
                <nav>
                    <ul>
                        <li><a href="# " class="active">Home</a><ion-icon
name="home-outline"></ion-icon>
                        </li>
                        <li><a href="about_us">About Us</a><ion-icon
name="log-in-outline"></ion-icon>
                        </li>
```

```
                                        <li><a href="https://nbkrist.co.in/index.php">NBKR
</a><ion-icon name="mail-outline"></ion-icon>
                                        </li>
                                    </ul>
                                </nav>
<style>
*/.grid-item {
  background-color: orange;
  padding: 10px;
  text-align: center;
} */
                                </style>
                                <div><h1 class="hero-content_title lede"><b>MENTAL
STRESS</b></h1></div>
<div><h3>CHECK YOUR MENTAL STRESS  <br></li> </h3></div><ion-icon name="hand-
left-outline"></ion-icon>
                                <button> <li><a class="clickme" href="clickme">clickhere
</button></a>
<style>
.clickme {
  position: relative;
  display:inline-flex;
}
.clickme::before {
  content: "";
  position: absolute;
  top: -5px;
  left: -25px;
  right: -10px;
  bottom: -5px;
  z-index: -1;
  opacity: 20;
  background-color: #1C163F;
```

```
  transition: opacity 0.10s ease-in-out;
                              padding: 5px;
}
.clickme:hover::before {
  opacity: 1;
}
</style>
<!--img src="images/stress9" alt="" class="" width="1300px" height="900px"-->
<article><div>            <h2 class="hero-content_title lede">STRESS</h2> </div>
<div class="hero-content_subtitle">"Stress is a normal reaction the body has when changes
occur, resulting in physical, emotional and intellectual responses. Stress management training
                              text-align: left;
                              background-color: #f2f2f2;
                              padding: 15px;
}
/* Style for the first div */
div:first-of-type {
  text-align: left;
  background-color: #f2f2f2;
  padding: 15px;
}
.hero-content_title {
  font-size: 26px;
  margin: 0;
                              margin-left: inherit;
/*                            animation: fadeInOut 10s ease-in-out infinite;
}
@keyframes fadeInOut {
  0% {
    opacity: 0;
  }
  50% {
    opacity: 1;
```

```
  }
  100% {
    opacity: 0;
  }
}
}
}*/
/* Style for the second div */
div:nth-of-type(2)--{
  margin: 20px 0;
}
.hero-content_subtitle {
  font-size: 30px;
  line-height: 10;
}
.clickhere {
  color: #0072C6;
  text-decoration: none;
}
.clickhere:hover {
  text-decoration:line-through;
}
/* Style for the article tag */
article {
  margin: 20px 0;
}
/* Style for the third div */
div:nth-of-type(3) {
  margin-top: 20px;
}
/* Style for the fthe th div */
div:nth-of-type(4) {
  margin: 20px 0;
```

```
}
/* Style for the fifth div */
div:nth-of-type(5) {
  font-size: 24px;
  font-weight: bold;
  margin: 20px 0;
}
/* Style for the h3 tag */
h3 {
  font-size: 24px;
  margin: 20px 0;
}
/* Style for the list inside the h3 tag */
ul {
  margin: 0;
  padding-left: 20px;
}
/* Style for the article tag inside the h3 tag */
h3 > article {
  margin-top: 10px;
}
/* Style for the sixth div */
div:nth-of-type(6) {
  margin: 20px 0;
}
/* Style for the last div */
div:last-of-type {
  margin-top: 20px;
}
/* Style for the headings inside the article tags */
article > div > h2 {
  font-size: 24px;
  font-weight: bold;
```

```
  margin: 0;
}
article > div > h3 {
  font-size: 20px;
  font-weight: bold;
  margin: 0;
}
div:nth-of-type(6) {
  margin-top: 20px;
}
</style>
<script type="module"
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>
<script nomodule src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js"></script>
                                </div>
                                <div>
<footer class="copyright"
<Copyright (c) 2023 Copyright Holder All Rights Reserved.>
</footer>
</body>
</html>
```

5.2 Program 2

```
{% load static %}
<!DOCTYPE html>
<!---Coding By CoderGirl!--->
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
   <title> An About Us Page | </title>
  <!---Custom Css File!--->
  <!--link rel="stylesheet" href="style.css"-->
```

```
</head>
<body>
  <section class="about-us">
    <div class="about">
      <img src="{% static 'stress/guide.jpg' %}" class="pic">
      <div class="text">
        <h2>DR M. SREENIVASULU</h2>
        <h4> B.Tech., M.Tech.,Ph.D</h4>
        <h5>PROFESSOR <br>DEPARTMENT OF MECHANICAL ENGINEERING</h5>
        <div class="data">
        <!--a href="mslu544@yahoo.com" class="email"> email</a-->
        <a href="https://www.nbkrist.co.in/fp.php?id=55" class="hire">READ MORE</a>
        </div>
      </div>
    </div>
  </section>
  <section class="about-us">
    <div class="about">
      <img src="{% static 'stress/vikas.jpg' %}" class="pic">
      <div class="text">
        <h3>SOMALA VIKAS </h3>
        <h4> contact  : 7989771183</h4>
        <div class="data"-->
        <a href="#" class="hire"> CONTACT</a>
        </div>
      </div>
    </div>
  </section>
  <section class="about-us">
    <div class="about">
      <img src="{% static 'stress/Sharath.jpg' %}"class="pic">
      <div class="text">
        <h3>V.SHARATH REDDY</h3>
```

```
      <h4>contact : 9704109224</h4>
      <div class="data">
      <a href="https://sharath9224.netlify.app/" class="hire">CONTACT </a>
      </div>
    </div>
  </div>
</section>
<section class="about-us">
  <div class="about">
    <img src="{% static 'stress/sivakumar.jpg' %}" class="pic">
    <div class="text">
      <h3>M.SIVA KUMAR </h3>
      <h4> contact: 7893356522</h4>
      <div class="data">
      <a href="#" class="hire">CONTACT </a>
      </div>
    </div>
  </div>
</section>
<section class="about-us">
  <div class="about">
    <img src="{% static 'stress/bhargav.jpg' %}" class="pic" >
    <div class="text">
      <h3>K.BHARGAV</h3>
      <h4> contact : 6304631664</h4>
      <div class="data">
      <a href="#" class="hire">contact</a>
      </div>
    </div>
  </div>
</section>
<section class="about-us">
  <div class="about">
```

```
      <img src="{% static 'stress/chand.jpg' %}" class="pic">
      <div class="text">
        <h3>K.BHARATH CHANDRA</h3>
        <h4> contact : 7893345730</h4>
        <div class="data">
        <a href="#" class="hire">CONTACT </a>
        </div>
      </div>
    </div>
  </section>
<style>
    @import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500&display=swap');
*{
margin: 0;
padding: 0;
box-sizing: border-box;
font-family: 'Poppins', sans-serif;
}
.about-us{
height: 100vh;
width: 100%;
padding: 90px 0;
background: #ddd;
}
.pic{
height: auto;
width:  302px;
}
.about{
width: 1130px;
max-width: 85%;
margin: 0 auto;
```

```
display: flex;
align-items: center;
justify-content: space-around;
}
.text{
width: 540px;
}
.text h2{
font-size: 90px;
font-weight: 600;
margin-bottom: 10px;
}
.text h5{
font-size: 22px;
font-weight: 500;
margin-bottom: 20px;
}
span{
color: #4070f4;
}
.text p{
font-size: 18px;
line-height: 25px;
letter-spacing: 1px;
}
.data{
margin-top: 30px;
}
.hire{
font-size: 18px;
background: #4070f4;
color: #fff;
text-decoration: none;
```

```
border: none;
padding: 8px 25px;
border-radius: 6px;
transition: 0.5s;
}
.hire:hover{
background: #000;
border: 1px solid #4070f4;
}
</style>
</body>
</html>
```

Project code for Django

```
def index(request):
    return Http Response("Hello, world. You're at the lab index.")
```

This is the simplest view possible in Django. To call the view, we need to map it to a URL - and for this we need a URL conf.

To create a URL conf in the ML detection directory, create a file called **urls.py**. Your app directory should now look like:

In the **ml/urls.py** file include the following code:

Convection/urls.py

```
from Django.urls import path
```

from Django urls import path

```
from .import views
stress\views.py
from django.shortcuts import render
from django.http import HttpResponse
from stress.models import StressDetection
from joblib import load
```

```
data1=load('./MLMODEL/model.joblib')
transform=load('./MLMODEL/model2.joblib')
def new_1(request):
return render(request,'new 1.html')
def about_us(request):
return render(request,'work.html')
def click_here(request):
return render(request,'ques.html')
def data(request):
ques1=request.POST.get('question4')
ques2=request.POST.get('When')
ques3=request.POST.get('question6')
ques4=request.POST.get('question7')
ques5=request.POST.get('question8')
ques6=request.POST.get('find')
ques7=request.POST.get('spend')
ques8=request.POST.get('goal')
ques9=request.POST.get('learn')
ques13=request.POST.get('work')
print(ques1,ques2,ques3,ques4,ques5,ques6,ques7,ques8,ques9,ques13)
stress=[[ques1,ques2,ques3,ques4,ques5,ques6,ques7,ques8,ques9,ques13]]
print(stress)
y=transform.transform(stress)
y_pred=data1.predict(y)
y_pred=y_pred[0]
# print(y_pred)
if y_pred==0.0:
y_pred='LOW STRESS'
if y_pred==1.0:
y_pred='MODERATE STRESS'
if y_pred==2.0:
y_pred='HIGH STRESS'
context={
```

'y':y_pred

}

5/4/23, 9:23 AM views.py

localhost:54963/b970e212-3c9c-427c-9dcf-a1e835537ee8/ 2/2

ins=StressDetection(question1=ques1,

question2=ques2,

question3=ques3,

question4=ques4,

question5=ques5,

question6=ques6,

question7=ques7,

question8=ques8,

question9=ques9,

question13=ques13

)

ins.save()

return render(request,'ques.h**tml',context)**

views.py

```
urlpatterns = [
    path("", views.index, name="index"),
]
```

The next step is to point the root URLconf at the **ml.urls** module. In **ml/urls.py**, add an import for **django.urls.include** and insert an **include()** in the **urlpatterns** list, so you have:

convection/urls.py¶

```
from django.contrib import admin
from django.urls import include, path
```

```
from django.urls import path
from . import views

urlpatterns = [
   path('',views.new_1,name='new_1'),
```

```
    path('about_us',views.about_us,name='about_us'),
    path('clickme',views.click_here,name='clickme'),
    path('data',views.data,name='data')
]
```

Models.py

```
from django.db import models
class StressDetection(models.Model):
    question1=models.CharField(max_length=10)
    question2=models.CharField(max_length=10)
    question3=models.CharField(max_length=10)
    question4=models.CharField(max_length=10)
    question5=models.CharField(max_length=10)
    question6=models.CharField(max_length=10)
    question7=models.CharField(max_length=10)
    question8=models.CharField(max_length=10)
    question9=models.CharField(max_length=10)

    question13=models.CharField(max_length=10)
```

CHAPTER 6

RESULT & DISCUSSUIONS

RESULT

Steps to view the website

- https://stressdetection.up.railway.app/ is the link of the website.
- Click the website link or type the website name any search engine.
- By clicking the link the website interface will look's as shownbelow
- On the interface of website there is a navbar.

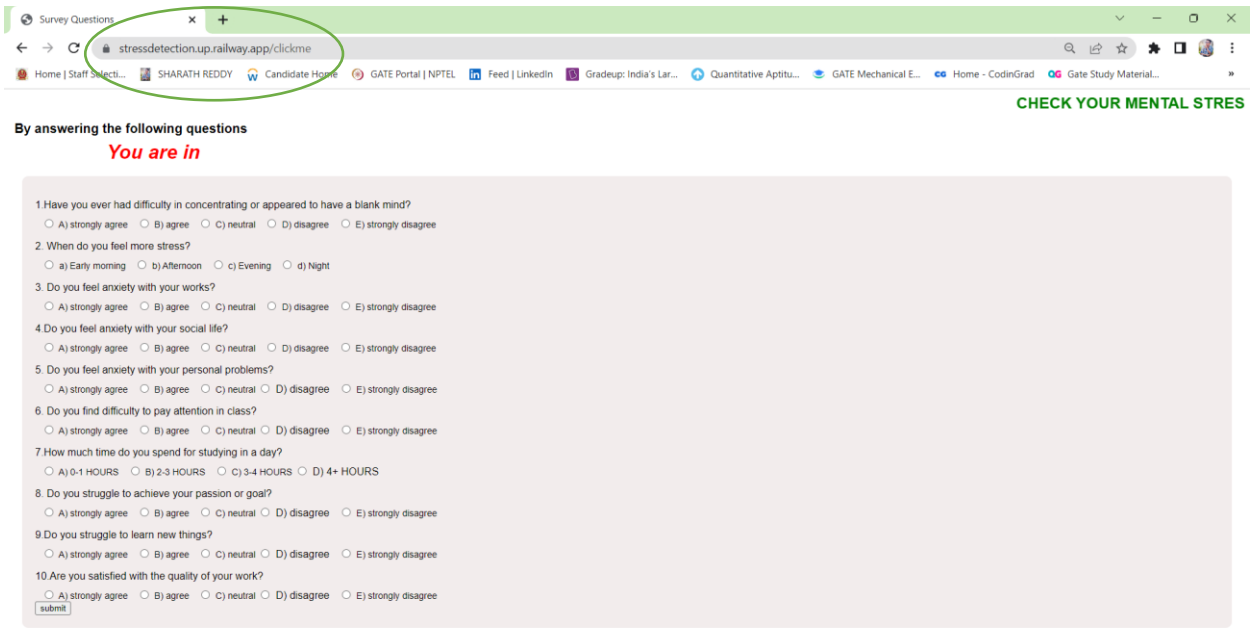


Fig 5.1 website main page

- After completion of given questions, click the below button which is used to predict the stress and situation at present level.

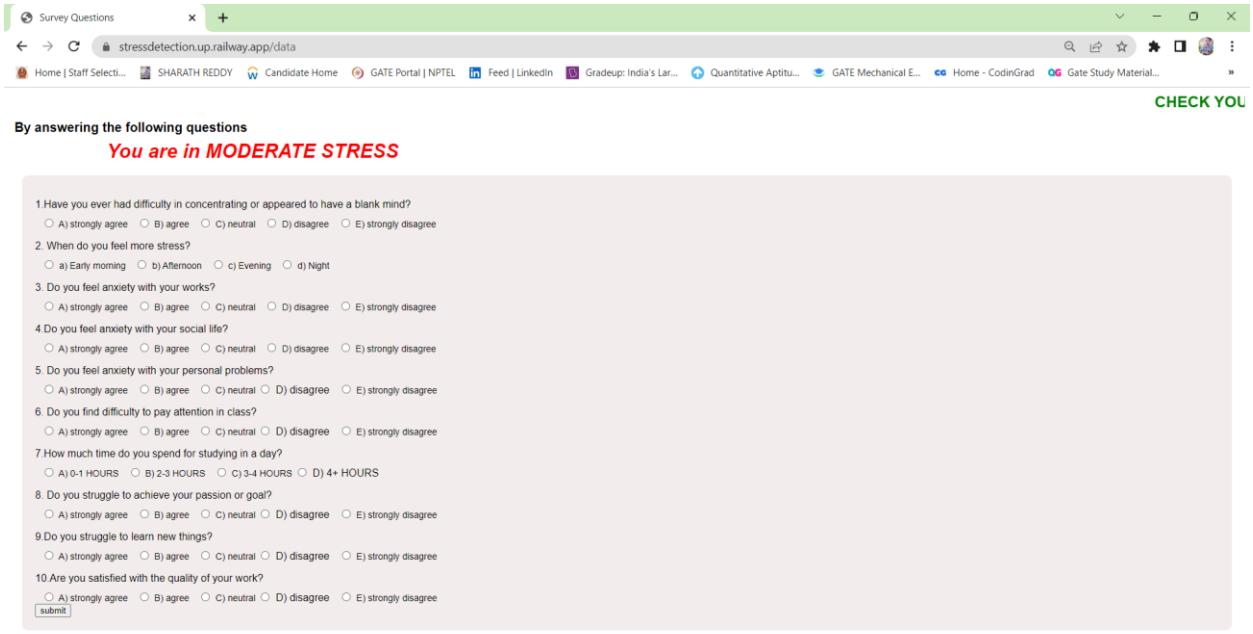


Fig 5.2 Result page of the website.

- The output data shows that the present scenario of mental stress and predict the stress level in three ranges high stress, moderate stress and Low stress.

CHAPTER 6

CONCLUSIONS & FUTURE ENHANCEMENTS

Conclusions

In the present work Random Forest, Naïve Bayes, Support Vector Machine, and K-Nearest Neighbour machine learning algorithms are used for calculating specificity, sensitivity, and accuracy. From these calculations it is found that the support vector machine is performing well when compared with the other algorithms. Support vector machine has given an accuracy of 85.71%, specificity 100% and sensitivity of 75%. Random Forest is performing next to support vector machine showed an accuracy of 83.33%, specificity of 66.66% and sensitivity of 100%. Hence it is concluded that the SVM is performing well out of the other algorithms worked with.

FUTURE ENHANCEMENTS

This study used a supervised machine-learning algorithm, feature selection methods to select the best subset features while developing the models. It is better to see the difference in performance results using unsupervised or deep learning algorithms models.

There are several potential future enhancements that could be made to improve the accuracy, specificity and sensitivity of the present work on mental stress prediction using machine learning. Some of these enhancements are as follows:

1. **Feature Engineering:** The accuracy of the machine learning model depends heavily on the quality and relevance of the features used for training. Therefore, it would be beneficial to explore different feature engineering techniques to improve the performance of the models.
2. **Hybrid Approaches:** In this project, we have used multiple machine learning algorithms to predict chronic kidney disease. In the future, we can explore hybrid approaches that combine the strengths of different models to improve accuracy and performance.
3. **Advanced Machine Learning Techniques:** There are several advanced machine learning techniques such as neural networks, deep learning, and ensemble learning, which can be explored to improve the accuracy of the models.
4. **Big Data Analytics:** With the increasing volume of data, there is a need to develop big data analytics techniques that can handle large-scale data and extract insights from it. In the future, we can explore big data analytics to improve the accuracy of the models.

5. **Real-Time Monitoring:** In this project, we have developed models that predict chronic kidney disease based on historical data. In the future, we can explore real-time monitoring techniques that can provide alerts and recommendations based on real-time data to prevent from stress.
6. **Mobile Application:** A mobile application can be developed that can help users monitor their health and provide personalized recommendations based on their data. The application can also integrate with wearable devices to collect data in real-time.

REFERENCES

Following IEEE papers on mental stress detection are referred:

1) “Automatic Stress Detection Using Wearable Sensors and Machine Learning: A Review” Shruti Gedam Dept. of Computer Science & Engg. Birla Institute of Technology, Mesra Ranchi, India.
2) Machine Learning and IoT for Prediction and Detection of Stress (Mr. Purnendu Shekhar Pandey BML, Munja University, Gurgaon Har“Binny S , A survey concept on Deep Learning, International Jthe nal of Scientific & Engineering Research “ Volume10,Issue 6, June2019 1570.
3) Setha Pan-ngum , and Pasin Israsena Noppadon Jatupaiboon , "Real-Time EEG-Based Happiness Detection System," in Hindawi Publishing Corporation the Scientific World Jthe nal Volume 2013 , 2013, p.12.
4) S. Binny, Dr. P. Sardar Maran, “Classification of Lung disease using deep learning neural networks, ”-- PENSEE (penseeresearch.com) ISSN: 0031-4773. Volume 51, Issue 6, Page No: 466-470, June 2021.
5) Mental Stress Detection in University Students using Machine Learning Algorithms Ravinder Ahujaa, Alisha Bangab aJaypee Institute of Information Technology, Noida 201304, India bSatyug Darshan Institute of Engineering and Technology, Faridabad-121002, India.
6) Yana, India purnendu.pandey@bml.edu.in).
7) “Stress detection using deep neural networks” Russell Li1 and Zhidong Liu2,3.
8) “A Decision Tree Optimised SVM Model for Stress Detection using Bio signals “ Alana Paul Cruz, Aravind Pradeep, Kavali Riya Sivasankari and Krishnaveni K.S.