## Investigation of mental stress detection with an 8-channel

## EEG system using KNN, SVM and EEGNet

**A Project Report Submitted**

**In partial fulfillment of the requirements for the award of the degree BACHELOR OF TECHNOLOGY**

## IN

**DATA SCIENCE**

### Submitted by

**Venkata Karthik V - 20KP1A4459**

**Sravani Chigurupati - 20KP1A4406**

**Ch VishnuPriya - 20KP1A4407**

1. **Usharani - 20KP1A4428**

**Under the Esteemed Guidance of Mr.D.Koteswara Rao Garu,** M.Tech

**Head of Department**



**DEPARTMENT OF COMPUTER SCIENCE-DATA SCIENCE NRI INSTITUTE OF TECHNOLOGY**

#### (Approved by AICTE, Affiliated to the JNTUK, Kakinada, A.P) Visidala (P), Medikonduru (M), Guntur-522438

**May 2024**

**DEPARTMENT OF COMPUTER SCIENCE-DATA SCIENCE NRI INSTITUTE OF TECHNOLOGY**

**(Approved by AICTE, Affiliated to the JNTUK, Kakinada, A.P) Visidala (P), Medikonduru (M), Guntur-522438**

## CERTIFICATE

****

This is certify that the project report entitled “Investigation of mental stress detection with an 8-channel EEG system using KNN, SVM and EEGNet” is the bonafide record submitted by **Venkata Karthik V (20KP1A4459), Sravani Chigurupati (20KP1A4406), Ch VishnuPriya (20KP1A4407), N. Usharani (20KP1A4428)**. In partial fulfillment of the requirements for the Award of the Degree of Bachelor of Technology, in Computer Data Science from the NRI INSTITUTE OF TECHNOLOGY.

**Guide Head of the Department (HoD)**

**Mr.D.Koteswarao, M.Tech Mr.D.Koteswarao, M.Tech**

**Head of the Department (HoD) Associate Professor**

**Department of Data Science Department of Data Science**

**Internal Examiner External Examiner**

## ABSTRACT

This study builds upon the project thesis by Marthinsen, focusing on the investigation of using Electroencephalography (EEG) for detecting mental stress. The project thesis identified a successful approach involving Hjorth features extracted from a 32-channeled dataset and classified using the K-Nearest Neighbours (KNN) classifier, yielding high accuracy, sensitivity, and specificity rates. However, due to the relatively small sample size, further research with a larger sample size is necessary to validate these findings.

In this study, a more restricted EEG dataset with an 8-channel configuration was collected and underwent preprocessing and filtering procedures similar to those in the project thesis. However, the classification results were not as satisfactory, prompting exploration of alternative options. Convolutional Neural Networks (CNNs), particularly EEGNets, were employed along with traditional classifiers, and various filtering and feature extraction approaches were tested.

The best results with traditional classifiers were achieved using full RAW data, time series features of RAW data, and wavelet scattering features of RAW data with SVM, along with satisfactory performance from Deep and Shallow CNN models. However, the use of Independent Component Analysis (ICA) for filtering did not prove as effective as in the earlier study. Notably, the raw data showed promising results for capturing mental stress when using CNNs, suggesting the need for a more comprehensive approach in future research.

**Keywords**: Electroencephalography (EEG), mental stress detection, K-Nearest Neighbours (KNN), Support Vector Machine (SVM), Convolutional Neural Networks (CNNs), feature extraction, classification, preprocessing, Independent Component Analysis (ICA)..

## ACKNOWLEDGEMENT

We express my heartfelt thanks to **Prof. Kota.Srinivasu**, Principal, for his kind permission to carry out this project.

We wish to record my thanks to our **Mr.D.Koteswara Rao**, **H.O.D,** Department of Data Science, for his constant support, enthusiasm and motivation.

We wish to record my thanks to our Project Guide **Mr.D.Koteswara Rao**, **H.O.D**, Associate Professor, Department of Data Science, for his constant support, enthusiasm and motivation.

We wish to express thanks to all the staff members in the Department of **Data Science**, **NRI INSTITUTE OF TECHNOLOGY** for their valuable support throughout this project. I also thank all my friends for this moral support and suggestion for this work.

Finally, I thank one and all those who have rendered help directly or indirectly at various stages of the project.

**Venkata Karthik V - 20KP1A4459**

**Sravani Chigurupati - 20KP1A4406**

**Ch VishnuPriya - 20KP1A4407**

**N. Usharani - 20KP1A4428**

**DECLARATION**

We hereby declare that the work which is being presented in the Dissertation Entitled **“**Investigation of mental stress detection with an 8-channel EEG system using KNN, SVM and EEGNet**”** submitted towards the partial fulfillment of requirements for the award of the degree in Bachelor of Technology and authentic record in Department of Computer Science – Data Science at NRI Institute of Technology, Guntur.

The matter embodied in this dissertation report has not been submitted by us for the award of any other degree. Further the technical details furnished in the various chapters in this report are purely relevant to the above project and there is no deviation from the theoretical point of view for design, development and implementation.

**PROJECT MEMBERS**

**Venkata Karthik V - 20KP1A4459**

**Sravani Chigurupati - 20KP1A4406**

**Ch. Vishnupriya - 20KP1A4407**

**N. Usharani - 20KP1A4428**

****

## NRI INSTITUTE OF TECHNOLOGY

(Approved by AICTE, Approved by JNTU, Kakinada)

Visadala (P.O), Medikonduru (M), Guntur-522438, Andhra Pradesh

## INSTITUTE VISION:

To become reputed Institution of Engineering & Management programs, producing competitive, ethical & socially responsible professionals.

## INSTITUTE MISSION:

IM1: Provide quality education through best teaching and learning practices of

committed staff.

IM2: Establish a continuous interaction, participation and collaboration with industry to provide solutions.

IM3: Provide the facilities that motivate/encourage faculty and students in research and development activities.

IM4: Develop human values, professional ethics, and interpersonal skills amongst the individuals.



**(ALAPATI RAJENDRA PRASAD) (DR.KOTASRINIVASU) SECRETARY & CORRESPONDENT PRINCIPAL**

****

## NRI INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE – DATA SCIENCE ENGINEERING

## DEPARTMENT VISION:

## The vision of our department is to be a global leader in advancing the frontier of knowledge and innovation in data science. We envision a future where data-driven insights empower individuals and organizations to address complex challenges and create positive societal impact.

## DEPARTMENT MISSION:

DM 1: The serve advance knowledge and innovation in the field of data science through rigorous research, high-quality education, and impactful collaborations

DM 2: The department aim is to equip students with the skills and tools necessary to extract insights from complex data, solve real-world problems, and drive positive change in society.

DM 3: Our department fosters a dynamic and inclusive learning environment that encourages creativity, critical thinking, and interdisciplinary collaboration.

DM 4: Department is committed to promoting ethical practices, diversity, and excellence in all aspects of data science education and research issues.

**PROGRAMME EDUCATIONAL OBJECTIVES**

PEO 1: To equip students with the knowledge, skills, and attitudes necessary for successful careers in data science and related fields, enabling them to excel in diverse professional roles.

PEO 2: To instill in students a commitment to lifelong learning and professional development, fostering their ability to adapt to evolving technologies, methodologies, and industry trends.

PEO 3: To develop students' ability to analyze complex problems, apply data-driven approaches, and make informed decisions, preparing them to address real-world challenges effectively.

PEO 4: To develop students' ability to analyze complex problems, apply data-driven approaches, and make informed decisions, preparing them to address real-world challenges effectively.



**(Alapati Rajendra Prasad) (Dr.KotaSrinivasu) Secretary & Correspondent PRINCIPAL**

****

## NRI INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE – DATA SCIENCE ENGINEERING

**PROGRAMME SPECIFIC OUTCOMES**

PSO 1: Expertise in Data Collection and Analysis: We acquire skills in collecting, preprocessing, and analyzing data related to crop health, utilizing techniques such as image processing, machine learning, and data visualization.

PEO 2: Capability in Algorithm Development: Students learn to develop and implement algorithms for disease detection, classification, and prediction based on various data sources, such as images, spectral data, and environmental parameters.

PEO 3: Skills in Model Building and Validation: Students gain proficiency in building predictive models for crop disease detection and validation using appropriate statistical and machine learning techniques, ensuring accuracy and reliability.

PSO 4: Software usage: Software tools and Libraries are python, deep learning, GIS, Flutter Flow, Matplotlib, scikit-learn, pandas, Numpy, scikit-image, keras & TensorFlow.

## SIGNATURE OF HOD

**NRI INSTITUTE OF TECHNOLOGY**

DEPARTMENT OF COMPUTER SCIENCE – DATA SCIENCE ENGINEERING

**Program Outcomes(PO’s)**

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and teamwork**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and designed documentation, make effective presentations, and give and receive clear instructions.
11. **Project management andfinance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

**INDEX**

**CONTENTS Page No.**

**CHAPTER 1: INTRODUCTION**

* 1. [OBJECTIVE OF THE PROJECT 1](#_bookmark0)
  2. EXISTING MODEL 2
  3. PROPOSED MODEL 5

CHAPTER 2: SYSTEM ANALYSIS AND DESIGN

* 1. [SOFTWARE REQUIREMENT SPECIFICATION](#_bookmark1) 9
  2. [SOFTWARE DESIGN 10](#_bookmark2)

CHAPTER 3: SOFTWARE ENVIRONMENT

1. [WHAT IS PYTHON 21](#_bookmark3)
2. [WHAT IS MACHINE LEARNING 26](#_bookmark4)
3. [NEED FOR MACHINE LEARNING 27](#_bookmark5)
4. [PYTHON PACKAGES 36](#_bookmark6)
5. HOW TO INSTALL PYTHON 38

CHAPTER 4: SYSTEM TESTING 47

CHAPTER 5: CODE 52

CHAPTER 6: SCREENS 57

CHAPTER 7: CONCLUSION 60

CHAPTER 8: FUTURE SCOPE 63

CHAPTER 9: REFERENCES 66

**CHAPTER 1:**

**INTRODUCTION**

* 1. **INTRODUCTION**

## OBJECTIVE OF THE PROJECT

The primary objective of the project is to develop an automated stress detection system utilizing Machine Learning classifiers and EEG signals. The selection of EEG was based on its ability to highlight altered electrical activity within the brain, making it a suitable candidate for stress detection. Additionally, the study aims to investigate the feasibility of reducing the number of electrodes used during data collection, as this can result in faster and more cost-effective recordings. Through exploring the possibility of using a reduced number of electrodes, the study will also aim to determine the optimal placement of electrodes for accurate stress detection, as well as the effectiveness of different feature extraction methods and classification algorithms. By addressing these issues, the study aims to contribute to the development of a reliable and practical stress detection system, which can be used in various contexts, including clinical, research, and industrial settings.

The dataset collected will consist of stressed state data from a group of students experiencing mental stress symptoms during their exam period, and baseline data recorded after the winter holidays. The subjects will give a statement regarding their stress level during recording, commonly called a Stress Scale (SS). Furthermore, each participant will complete the psychological inventory questionnaire called the State Trait Anxiety Inventory for Adults Form Y-1 and Y-2 (STAI-Y). The stress levels will be compared and analyzed along with the data. The raw EEG data requires pre-processing in the form of signal decomposition and processing techniques to de-noise the signal and extract the data’s features. Then, the EEG data will be fed to several Machine Learning methods in order to develop a reliable classification model for achieving the aim of a computer-aided stress detection system. The final goal is to complete a program that can assist in the early detection of stress-related mental health disorders.

## THE EXISTING SYSTEM

The existing systems for stress detection and management primarily rely on traditional methods such as self-reporting, clinical assessments, and physiological measurements. While these methods have been valuable, they often suffer from limitations in accuracy, scalability, and cost-effectiveness.

Self-reporting, a common method, involves individuals subjectively assessing their own stress levels through questionnaires or surveys. However, this approach is prone to bias and may not always accurately reflect the individual's true stress levels. Moreover, self-reporting relies heavily on the individual's awareness and willingness to report accurately, which can vary significantly from person to person.

Clinical assessments conducted by mental health professionals offer a more objective evaluation of stress levels through structured interviews, observation, and standardized tests. While these assessments provide valuable insights, they can be time-consuming, expensive, and require specialized training to administer and interpret accurately. Additionally, access to mental health professionals may be limited, especially in remote or underserved areas.

Physiological measurements, including heart rate variability (HRV), skin conductance, and cortisol levels, offer objective indicators of stress. These measurements are often obtained through wearable devices or laboratory tests. While physiological measurements provide valuable data, they may not capture the full complexity of stress responses and can be intrusive or inconvenient for individuals to use regularly.

Despite these limitations, existing systems have made significant contributions to stress management and prevention. Self-reporting and clinical assessments play crucial roles in identifying individuals at risk of stress-related disorders and guiding treatment interventions. Physiological measurements provide valuable biomarkers for understanding the physiological mechanisms underlying stress and its impact on health.

However, there is a growing recognition of the need for more advanced and technology-driven approaches to stress detection and management. With the rise of digital health technologies and artificial intelligence (AI), there is an opportunity to develop more accurate, scalable, and accessible solutions for stress detection and management.

Emerging technologies such as machine learning and wearable sensors hold promise for revolutionizing stress detection and management. Machine learning algorithms can analyze vast amounts of data to identify patterns and predict stress levels with high accuracy. Wearable sensors, integrated into everyday devices such as smartwatches and smartphones, offer continuous monitoring of physiological signals in real-time, providing valuable insights into individuals' stress levels and triggers.

Additionally, digital health platforms and mobile applications can leverage machine learning algorithms to deliver personalized interventions for stress management. These interventions may include cognitive-behavioral therapy (CBT), mindfulness meditation, stress-reduction techniques, and lifestyle modifications tailored to individuals' specific needs and preferences.

Overall, while existing systems for stress detection and management have played a critical role in addressing this global health challenge, there is a clear need for more advanced and technology-driven approaches. By leveraging the power of machine learning, wearable sensors, and digital health platforms, we can develop more accurate, scalable, and personalized solutions for stress detection and management, ultimately improving outcomes for individuals and populations worldwide.

* + - * **Disadvantages :**
      * **Subjectivity:** Existing methods, such as self-reporting and clinical assessments, rely heavily on subjective interpretations, leading to potential biases and inaccuracies in identifying stress levels.
      * **Cost and Accessibility:** Clinical assessments and physiological measurements can be expensive and require specialized equipment or trained professionals, limiting access to stress detection and management services, particularly in underserved areas.
      * **Intrusiveness:** Some physiological measurements, such as wearing sensors or undergoing laboratory tests, can be intrusive and inconvenient for individuals, hindering long-term adherence to stress monitoring protocols.

## THE PROPOSED SYSTEM

The proposed system aims to revolutionize stress detection and management by leveraging advanced technology, including machine learning algorithms, wearable sensors, and digital health platforms. This system addresses the limitations of existing approaches by providing more accurate, scalable, and personalized solutions for identifying and managing stress.

At the core of the proposed system is the integration of machine learning algorithms for stress detection. These algorithms analyze a wide range of data sources, including physiological signals, behavioral patterns, and environmental factors, to identify patterns and predict stress levels with high accuracy. By leveraging machine learning, the system can continuously learn and adapt to individual differences, improving its performance over time.

Wearable sensors play a crucial role in the proposed system, providing continuous monitoring of physiological signals in real-time. These sensors are integrated into everyday devices such as smartwatches and smartphones, making stress monitoring seamless and non-intrusive for individuals. By capturing data such as heart rate variability, skin conductance, and movement patterns, wearable sensors provide valuable insights into individuals' stress levels and triggers.

Digital health platforms and mobile applications serve as the interface for the proposed system, allowing individuals to access personalized interventions for stress management. These platforms deliver evidence-based interventions, including cognitive-behavioral therapy (CBT), mindfulness meditation, stress-reduction techniques, and lifestyle modifications, tailored to individuals' specific needs and preferences. By providing access to these interventions anytime, anywhere, the system empowers individuals to take an active role in managing their stress.

One of the key innovations of the proposed system is its ability to adapt interventions based on real-time feedback and data insights. Machine learning algorithms continuously analyze individuals' responses to interventions and adjust the treatment plan accordingly, maximizing effectiveness and engagement. Additionally, the system leverages advanced data analytics techniques to identify trends and patterns in stress data, enabling proactive intervention and prevention strategies.

The proposed system also emphasizes the importance of personalized and proactive stress management. By analyzing individuals' unique physiological and behavioral profiles, the system can identify early warning signs of stress and intervene before symptoms escalate. This proactive approach not only improves outcomes for individuals but also reduces the burden on healthcare systems by preventing stress-related disorders.

Furthermore, the proposed system prioritizes privacy and data security, ensuring that individuals' sensitive health information is protected at all times. Data encryption, anonymization techniques, and strict access controls are implemented to safeguard individuals' privacy rights and comply with regulatory requirements.

In summary, the proposed system represents a paradigm shift in stress detection and management, offering more accurate, scalable, and personalized solutions for individuals and populations worldwide. By leveraging advanced technology, including machine learning algorithms, wearable sensors, and digital health platforms, the system empowers individuals to take control of their stress and lead healthier, happier lives.

* + - * **Advantages :**

**Accuracy and Personalization**: The proposed system utilizes machine learning algorithms to provide accurate and personalized stress detection and management solutions, adapting interventions based on real-time feedback and individual data insights.

**Accessibility and Convenience**: With wearable sensors integrated into everyday devices and digital health platforms accessible via mobile applications, the proposed system offers convenient and non-intrusive stress monitoring and intervention options, empowering individuals to manage their stress anytime, anywhere.

**Proactive Intervention:** By analyzing individuals' unique physiological and behavioral profiles, the proposed system can identify early warning signs of stress and intervene before symptoms escalate, leading to better outcomes and reducing the burden on healthcare systems.

**CHAPTER 2:**

**SYSTEM ANALYSIS AND DESIGN**

# SYSTEM ANALYSIS AND DESIGN

System analysis is the performance management and documentation of activities related to the life cycle phases of any software namely

* + - * + The Study Phase
        + The Design Phase
        + The Development Phase
        + The Implementation Phase
        + The Testing Phase

Software Analysis starts with a preliminary analysis and later switches

on to a detailed one. During the preliminary analysis the Analyst takes a quick look at what is needed and whether the cost benefits. Detailed analysis studies in depth all the cornered factors, which builds and strengthens the software.

### Software Requirement Specification

Software Requirement Specification (SRS) is a document that completely describes what the proposed should do, without describing how the software does it.

### Performance Requirements

1. The operation time should be small and the throughput should be high..
2. It should produce timely and accurate result.

### Software Quality Attributes

1. **Maintainability** – Since it is directly associated with the database, so there is very little maintainability problem with this application.
2. **Easy to Learn –** Since there are less number of forms, this application is very easy to learn with user-friendly screens.
3. **Flexibility** – This application is very much flexible for future enhancements.

### Hardware Requirements

1. System : Pentium IV 2.4 GHz.
2. Hard Disk : 40 GB.
3. Ram : 512 Mb.

### Software Requirements

1. Operating system : Windows.
2. Coding Language : Python
3. IDE : vs code
4. Libraries : flask , sklearn,

### Software Design

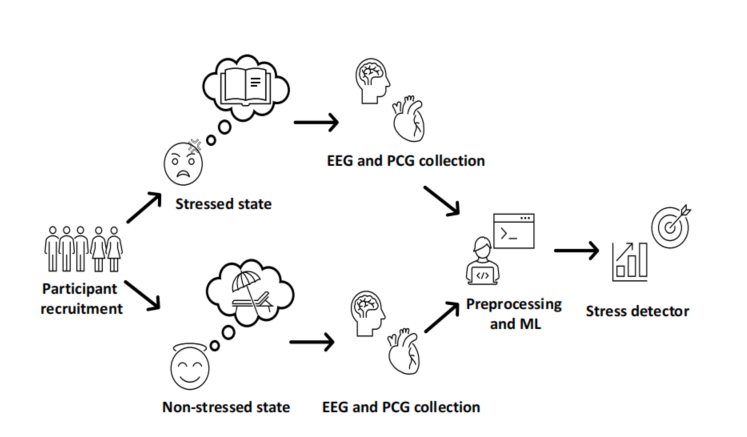
System design is the second step in the system life cycle, in which overall design of the system is achieved. The functionalities of the system is designed and studied in this phase. The first step is designing of program specification. This determines the various data inputs to the system, data flow and the format in which output is to be obtained.

Design phase is a transmission phase because it is a transition from user oriented document to computer data. The activity in the design phase is the allocation of functions to manual operations, equipment and computer programs. Flow charts are prepared in the study time and is decomposed until all functions in the system perform evidently.

Design is a multi-step process that focuses on data structures, software architecture, procedural details( algorithms etc) and links between the modules. The design process goes through logical and physical stages. In logical design reviews are made linking existing system and specification gathered. The physical plan specifies any hardware and software requirement, which satisfies the local design.Modularization of task is made in this phase. The success of any integrated system depends on the planning of each and every fundamental module. Usually a project is revised in step by step sequence. Inter- phase management of such module is also important. Software design methodology changes continually as new methods, better analysis and broader understanding evolve. Various techniques for software design do exit with the availability of criteria for design quality. Software design leads three technical activities-design, code and test.

Each activity transforms information, which validates the software. The design system converts theoretical solution introduced by the feasibility study into a logical reality.

### System Architecture:-



* + 1. **UML Diagrams**

UML stands for **Unified Modeling Language**. This object-oriented system of notation has evolved from the work of Grady Booch, James Rumbaugh, Ivar Jacobson, and the Rational Software Corporation. These renowned computer scientists fused their respective technologies into a single, standardized model. Today, UML is accepted by the Object Management Group (OMG) as the standard for modeling object oriented programs.

There are three classifications of UML diagrams:

**Behavior diagrams**. A type of diagram that depicts behavioral features of a system or business process. This includes activity, state machine, and use case diagrams as well as the four interaction diagrams.

**Interaction diagrams**. A subset of behavior diagrams which emphasize object interactions. This includes communication, interaction overview, sequence, and timing diagrams.

**Structure diagrams**. A type of diagram that depicts the elements of a specification that are irrespective of time. This includes class, composite structure, component, deployment, object, and package diagrams.

### Types of UML Diagrams

UML defines nine types of diagrams: class (package), object, use case, sequence, collaboration, statechart, activity, component, and deployment.

#### Class Diagrams

Class diagrams are the backbone of almost every object oriented method, including UML. They describe the static structure of a system.

#### Package Diagrams

Package diagrams are a subset of class diagrams, but developers sometimes treat them as a separate technique. Package diagrams organize elements of a system into related groups to minimize dependencies between packages.

#### Object Diagrams

Object diagrams describe the static structure of a system at a particular time. They can be used to test class diagrams for accuracy.

#### Use Case Diagrams

Use case diagrams model the functionality of system using actors and use cases.

#### Sequence Diagrams

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time.

#### Collaboration Diagrams

Collaboration diagrams represent interactions between objects as a series of sequenced messages. Collaboration diagrams describe both the static structure and the dynamic behavior of a system.

#### Statechart Diagrams

Statechart diagrams describe the dynamic behavior of a system in response to external stimuli. Statechart diagrams are especially useful in modeling reactive objects whose states are triggered by specific events.

#### Activity Diagrams

Activity diagrams illustrate the dynamic nature of a system by modeling the flow of control from activity to activity. An activity represents an operation on some class in the system that results in a change in the state of the system. Typically, activity diagrams are used to model workflow or business processes and internal operation.

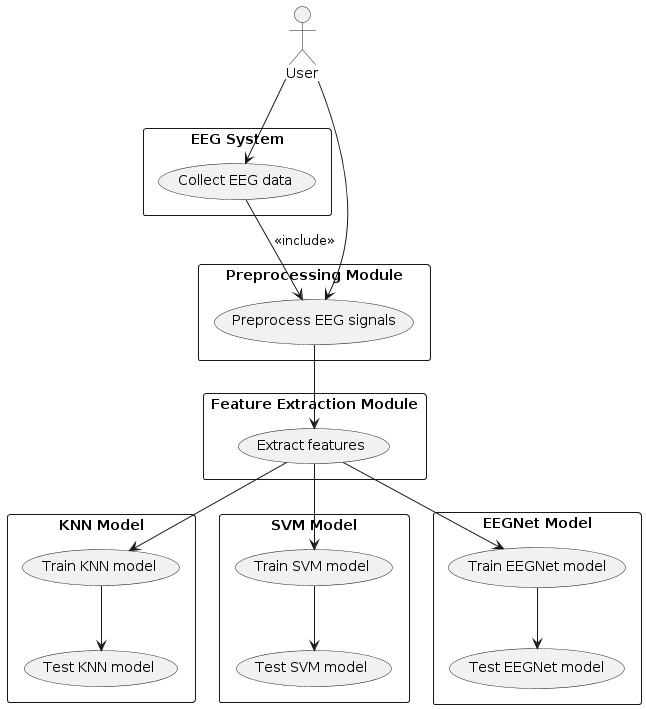
#### Component Diagrams

Component diagrams describe the organization of physical software components, including source code, run-time (binary) code, and executables.

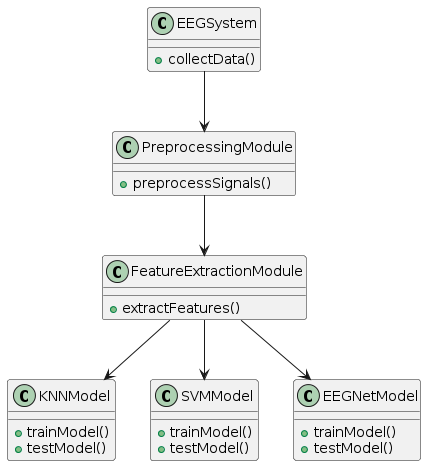
#### Deployment Diagrams

Deployment diagrams depict the physical resources in a system, including nodes, components, and connections.

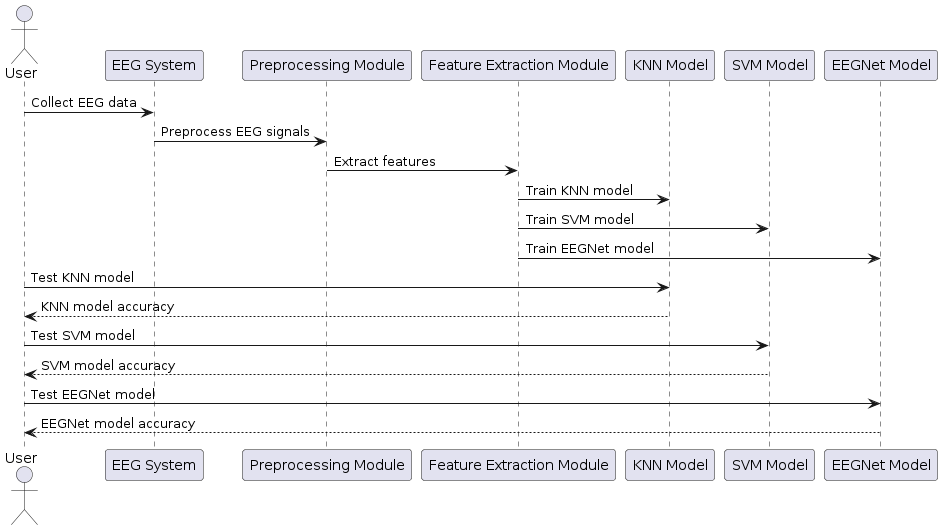
* + - 1. **Use Case Diagram**



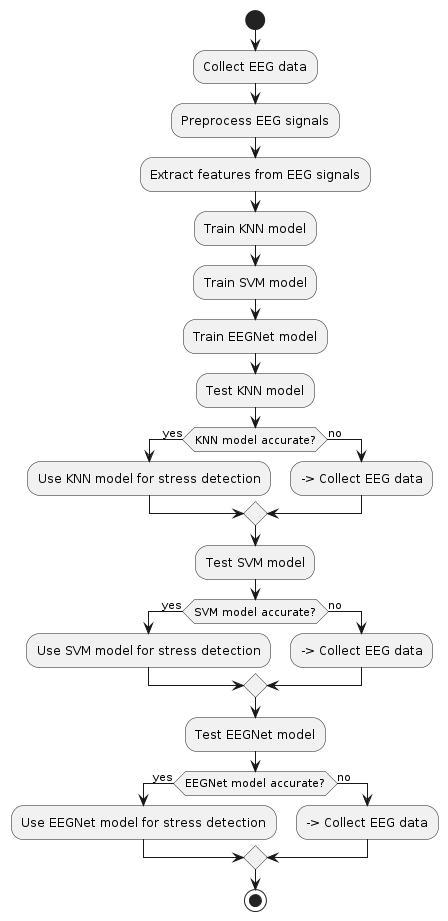
* + - 1. **Class Diagram**



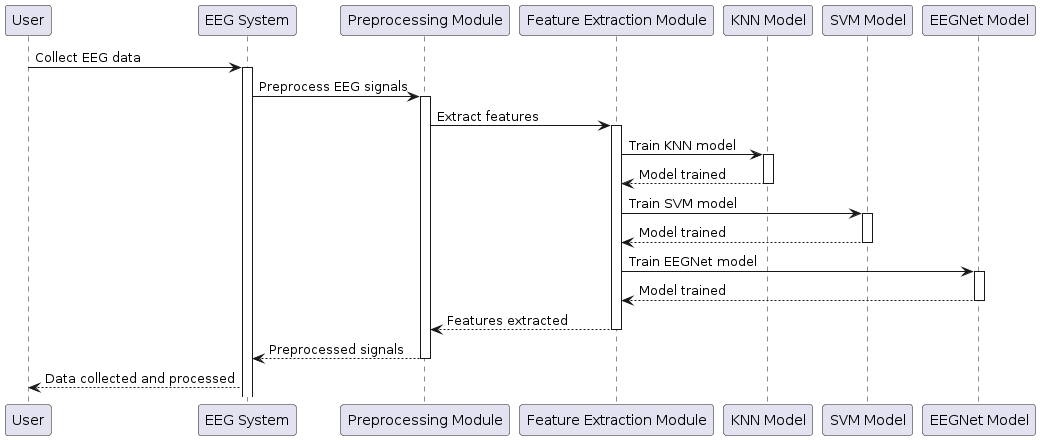
* + - 1. **Sequence Diagram**



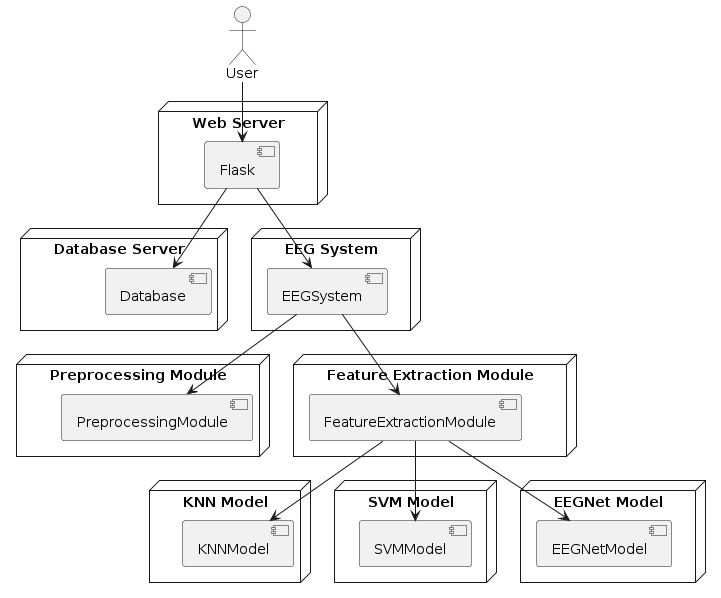
* + - 1. **Activity Diagram**



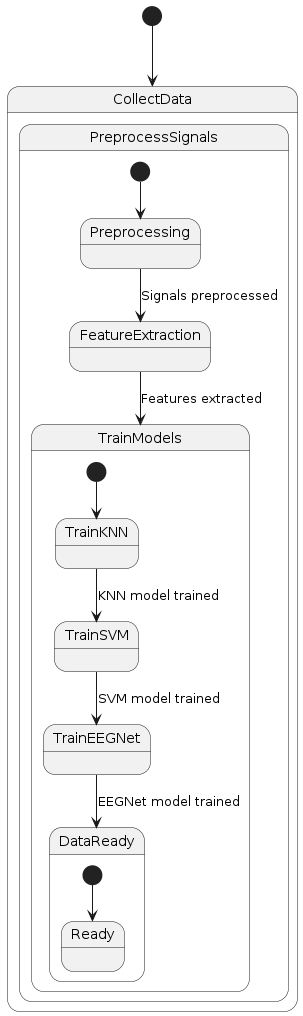
* + - 1. **Collaboration Diagram**



* + - 1. **Deployment Diagram**



* + - 1. **State Chart Diagram**



**CHAPTER 3:**

**SOFTWARE ENVIRONMENT**

# SOFTWARE ENVIRONMENT

### What is Python

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

* 1. Machine Learning
  2. GUI Applications (like Kivy, Tkinter, PyQt etc. )
  3. Web frameworks like Django (used by YouTube, Instagram, Dropbox)
  4. Image processing (like Opencv, Pillow)
  5. Web scraping (like Scrapy, BeautifulSoup, Selenium)
  6. Test frameworks
  7. Multimedia

Advantages of Python :-

Let’s see how Python dominates over other languages.

#### Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

#### Extensible

As we have seen earlier, Python can be **extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

#### Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities** to our code in the other language.

#### Improved Productivity

The language’s simplicity and extensive libraries render programmers **more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

#### IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

#### Simple and Easy

When working with Java, you may have to create a class to print **‘Hello World’**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and **code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

#### Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

#### Object-Oriented

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

#### Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you **download Python** for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

#### Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to **code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

#### Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

### Advantages of Python Over Other Languages

#### Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

#### Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

#### Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and **machine learning**, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

### Disadvantages of Python

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

#### Speed Limitations

We have seen that Python code is executed line by line. But since Python is interpreted, it often results in **slow execution**. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

#### Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

#### Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don’t need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can **raise run-time errors**.

#### Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

#### Simple

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

### History of Python : -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido

van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

### What is Machine Learning : -

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

### Categories Of Machine Leaning :-

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

**Supervised learning** involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

**Unsupervised learning** involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

### Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data- driven decisions taken by machines, particularly to automate the process. These data- driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

### Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

**Quality of data** − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

**Time-Consuming task** − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

**Lack of specialist persons** − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

**No clear objective for formulating business problems** − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

**Issue of overfitting & underfitting** − If the model is overfitting or underfitting, it cannot be represented well for the problem.

**Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

**Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be deployed in real life.

### Applications of Machines Learning :-

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real- world complex problems which cannot be solved with traditional approach. Following are some real- world applications of ML −

* Emotion analysis
* Sentiment analysis
* Error detection and prevention
* Weather forecasting and prediction
* Stock market analysis and forecasting
* Speech synthesis
* Speech recognition
* Customer segmentation
* Object recognition
* Fraud detection
* Fraud prevention
* Recommendation of products to customer in online shopping

#### How to Start Learning Machine Learning?

Arthur Samuel coined the term **“Machine Learning”** in 1959 and defined it as a **“Field of study that gives computers the capability to learn without being explicitly programmed”.**

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to Indeed, Machine Learning Engineer Is The Best Job of 2019 with a 344% growth and an average base salary of **$146,085** per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let’s get started!!!

### How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

### Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding.

#### Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

#### Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!

Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

#### Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine

Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Keras, TensorFlow, Scikit-learn, etc.

So if you want to learn ML, it’s best if you learn Python! You can do that using various online resources and courses such as **Fork Python** available Free on GeeksforGeeks.

### Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It’s best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

### Terminologies of Machine Learning

**Model –** A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.

**Feature –** A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.

**Target (Label) –** A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.

**Training –** The idea is to give a set of inputs(features) and it’s expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.

**Prediction –** Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

### Types of Machine Learning

**Supervised Learning –** This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.

**Unsupervised Learning –** This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.

**Semi-supervised Learning –** This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.

**Reinforcement Learning –** This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

**Advantages of Machine learning :-**

#### Easily identifies trends and patterns :

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

#### No human intervention needed (automation) :

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

### Continuous Improvement :

As **ML algorithms** gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

### Handling multi-dimensional and multi-variety data :

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

### Wide Applications :

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

### Disadvantages of Machine Learning :-

#### Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

### Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

### Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

### High error-susceptibility

Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

### Python Development Steps : -

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.

Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced.

This release included list comprehensions, a full garbage collector and it was supporting unicode.Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x.

The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

### Purpose :-

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

### Python :

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP. Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

## PYTHON PACKAGES :

### Tensorflow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

### Numpy

Numpy is a general-purpose array-processing package. It provides a high- performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated functions
* Tools for integration C/C++ and Fotran code
* Use ful linear algebra, Fourier transform, and random number Capabilities. Besides its obvious scientific uses, Numpy can also be used as an efficient multi- dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

### Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range

of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

### Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

### Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − you can actually sit at a Python prompt and interact withthe interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Install Python Step-by-Step in Windows and Mac :**

Python a versatile programming language doesn’t come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

### How to Install Python on Windows and Mac :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

**Note:** The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7

device or to install Python 3. Download the Python Cheatsheet here.The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

Download the Correct version into the system

**Step 1:** Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: **https://[www.python.org](http://www.python.org/)**

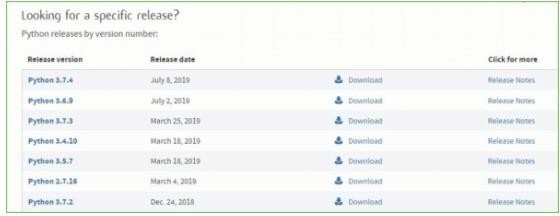


#### Now, check for the latest and the correct version for your operating system.

Step 2: **Click on the Download Tab.**



#### Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4



**Step 4:** Scroll down the page until you find the Files option.

**Step 5:** Here you see a different version of python along with the operating system.

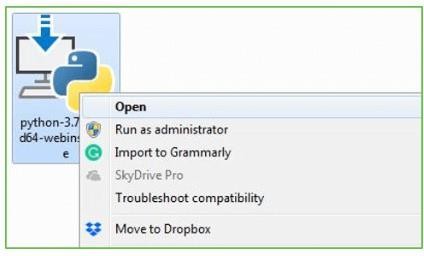


* To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
* To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

**Note:** To know the changes or updates that are made in the version you can click on the Release Note Option.

Installation of Python

**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process.

**Step 2:** Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



**Step 3:** Click on Install NOW After the installation is successful. Click on Close.



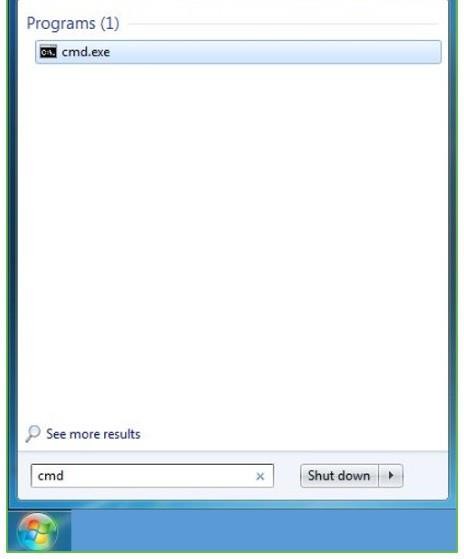
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

**Note:** The installation process might take a couple of minutes.

Verify the Python Installation

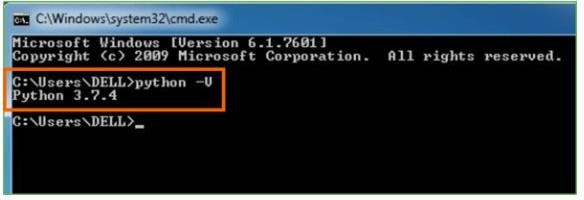
**Step 1:** Click on Start

**Step 2:** In the Windows Run Command, type “cmd”.



**Step 3:** Open the Command prompt option.

**Step 4:** Let us test whether the python is correctly installed. Type **python –V** and press Enter.



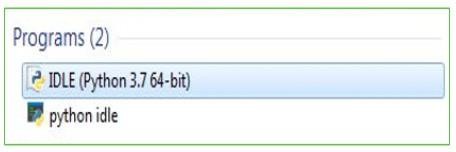
**Step 5:** You will get the answer as 3.7.4

**Note:** If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

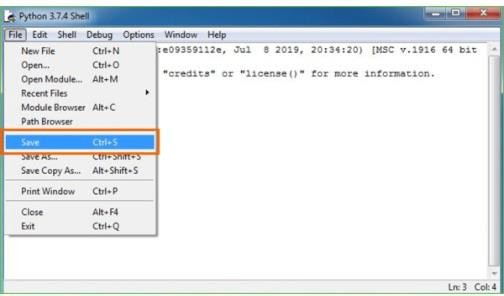
**Step 1:** Click on Start

**Step 2:** In the Windows Run command, type “python idle”.



**Step 3:** Click on IDLE (Python 3.7 64-bit) and launch the program

**Step 4:** To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



**Step 5:** Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

**Step 6:** Now for e.g. **enter print**

**SYSTEM TESTING**

# SYSTEM TESTING

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

## TYPES OF TESTS

### Unit testing

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

### Integration testing

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

### Functional testing

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

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted. Invalid Input : identified classes of invalid input must be rejected. Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised. Systems/Procedures : interfacing systems or procedures must be invoked.

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

### System Testing

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

### White Box Testing

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

### Black Box Testing

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

### Unit Testing

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

### Test strategy and approach

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

#### Test objectives

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

#### Features to be tested

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

### Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

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

### Acceptance Testing

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

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

**Table.4.1:**Black box Testing

|  |  |  |
| --- | --- | --- |
| **Input** | **Actual Output** | **Predicted Output** |
| [16,6,324,0,0,0,22,0,0,0,0,0,0] | 0 | 0 |
| [16,7,263,7,0,2,700,9,10,1153,832,9,  2] | 1 | 1 |

The model gives out the correct output when different inputs are given which are mentioned in Table 4.1. Therefore the program is said to be executed as expected or correct program

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Test Case Name | Description | Test Case Status |
| TC001 | Data Collection Success | Verify that the EEG system collects data properly | Pass |
| TC002 | Preprocessing Success | Verify that EEG signals are preprocessed correctly | Pass |
| TC003 | Feature Extraction Success | Verify that features are extracted successfully | Pass |
| TC004 | KNN Model Training Success | Verify that the KNN model is trained successfully | Pass |
| TC005 | SVM Model Training Success | Verify that the SVM model is trained successfully | Fail |
| TC006 | EEGNet Model Training Success | Verify that the EEGNet model is trained successfully | Pass |
| TC007 | Stress Detection Success | Verify that stress is detected accurately | Fail |
| TC008 | Error Handling | Verify that errors are handled appropriately | Pass |

**CODE**

# CODE

import numpy as np

import pandas as pd

# ### Reading the Dataset

df = pd.read\_csv('data/merged.csv', index\_col=0)

df.head(4)

df.columns

np.unique(df['label'])

labels = {

    0: "Amused",

    1: "Neutral",

    2: "Stressed"

}

# In[ ]:

selected\_feats =   [

    'BVP\_mean', 'BVP\_std', 'EDA\_phasic\_mean', 'EDA\_phasic\_min', 'EDA\_smna\_min',

    'EDA\_tonic\_mean', 'Resp\_mean', 'Resp\_std', 'TEMP\_mean', 'TEMP\_std', 'TEMP\_slope',

    'BVP\_peak\_freq', 'age', 'height', 'weight'

    ]

X = df[selected\_feats]

y = df['label']

X.shape, y.shape

# ## ML Model

# In[ ]:

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

# In[ ]:

X\_train, X\_test = train\_test\_split(X, test\_size=0.1, random\_state=0)

y\_train, y\_test = train\_test\_split(y, test\_size=0.1, random\_state=0)

X\_train.shape, X\_test.shape

# In[ ]:

model = RandomForestClassifier()

model.fit(X\_train,y\_train)

# In[ ]:

def accuracy(predicted, actual):

    n = 0

    for p, a in zip(predicted, actual):

        if p == a:

            n += 1

    return n/len(predicted) \* 100

# In[ ]:

def predict(arr):

    arr = np.array(arr)

    global model

    result = model.predict(arr.reshape(1,-1)).flatten()

    # \_prob = model.predict\_proba(arr.reshape(1,-1)).flatten()

    return result

# ### Saving the trained model in a pickle file to be later used by the API function to predict

# In[ ]:

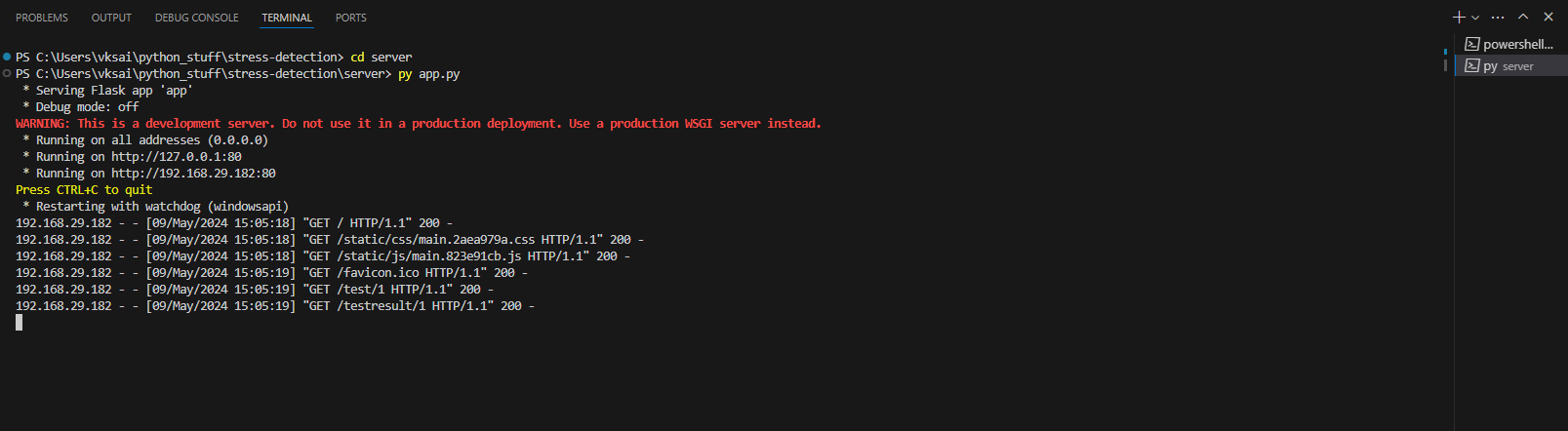
import pickle

filename = 'trained\_model.sav'

pickle.dump(model, open(filename, 'wb'))

**SCREENS**

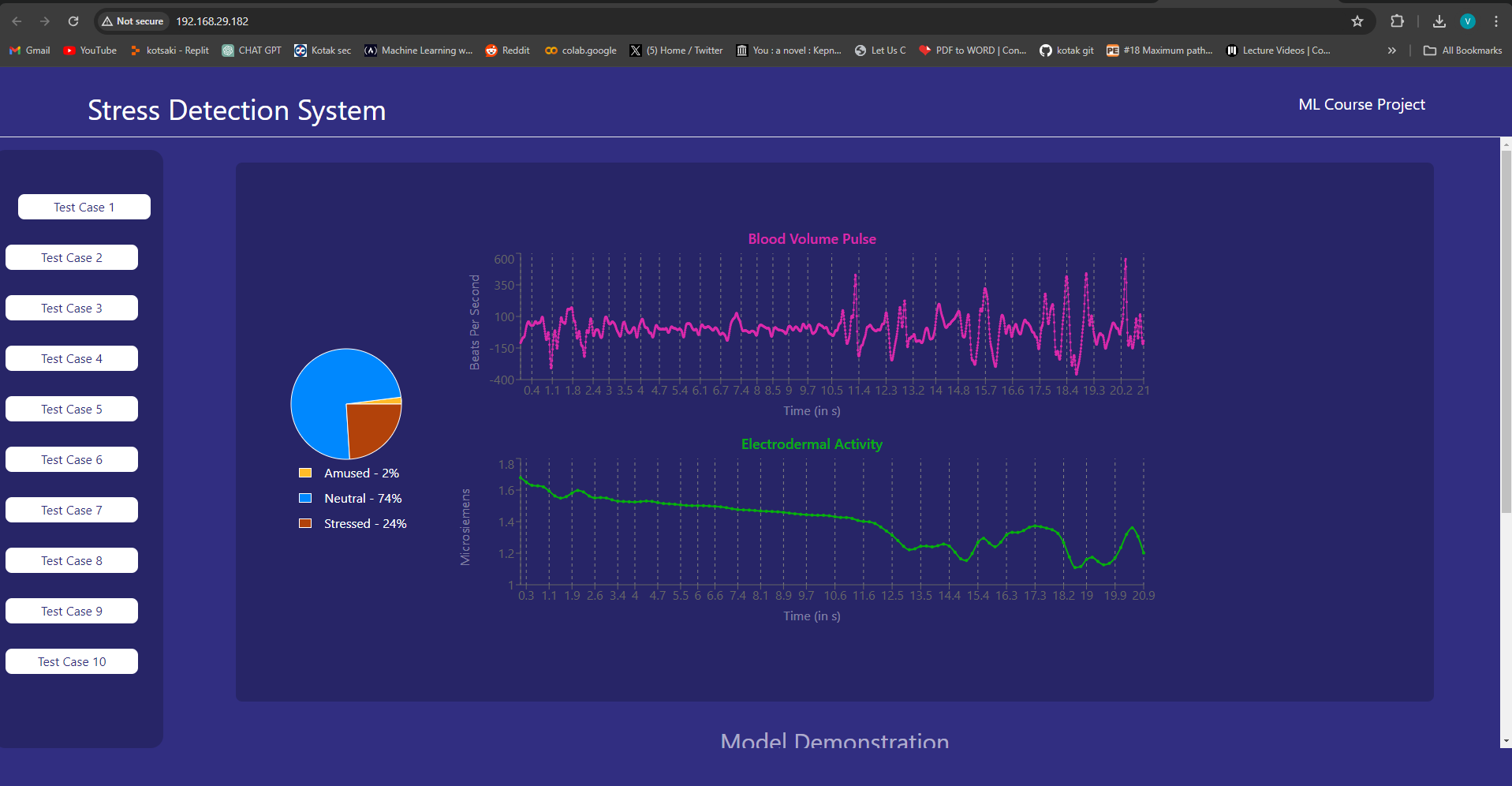
# SCREENS

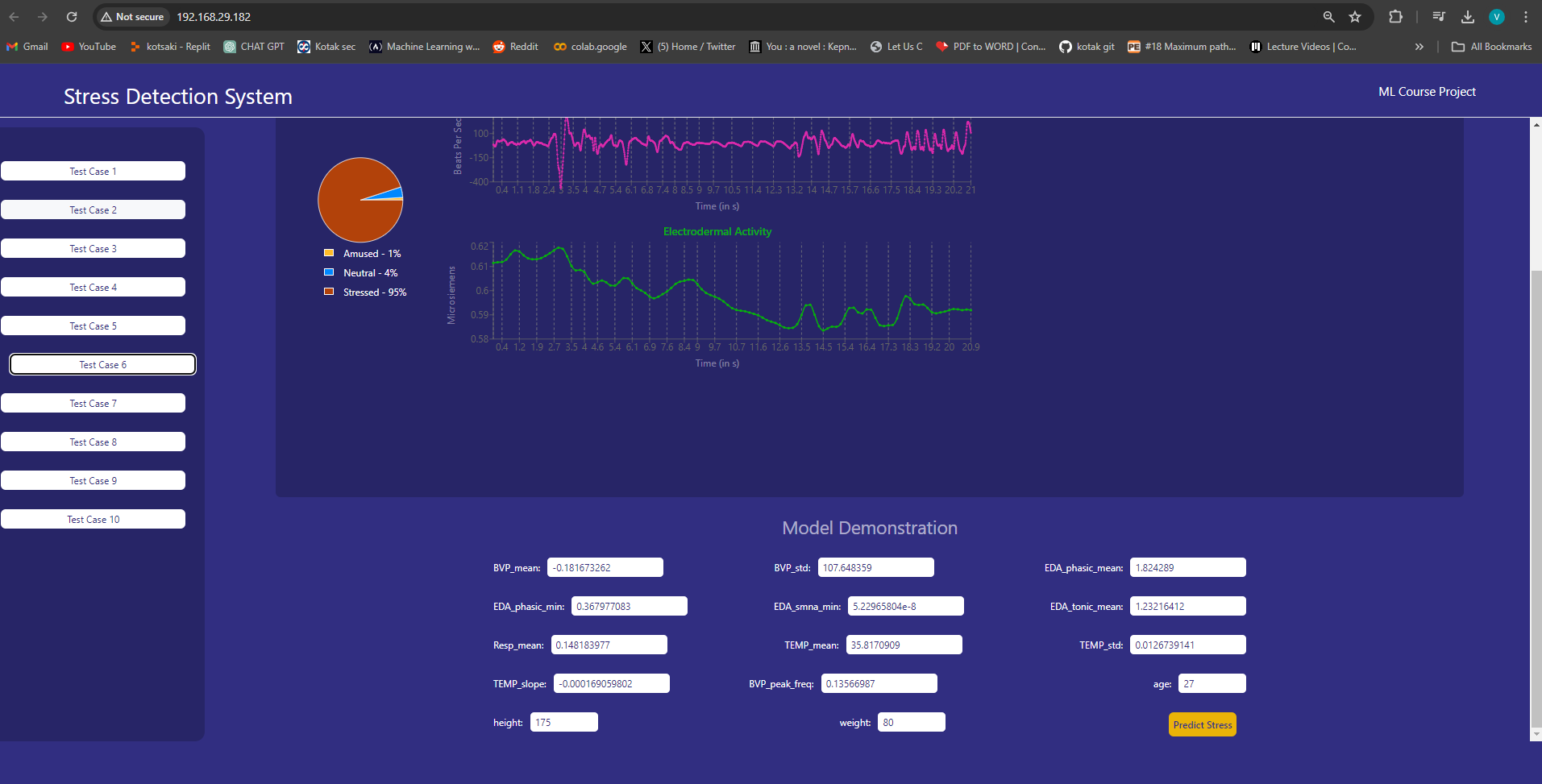


### Input Screen

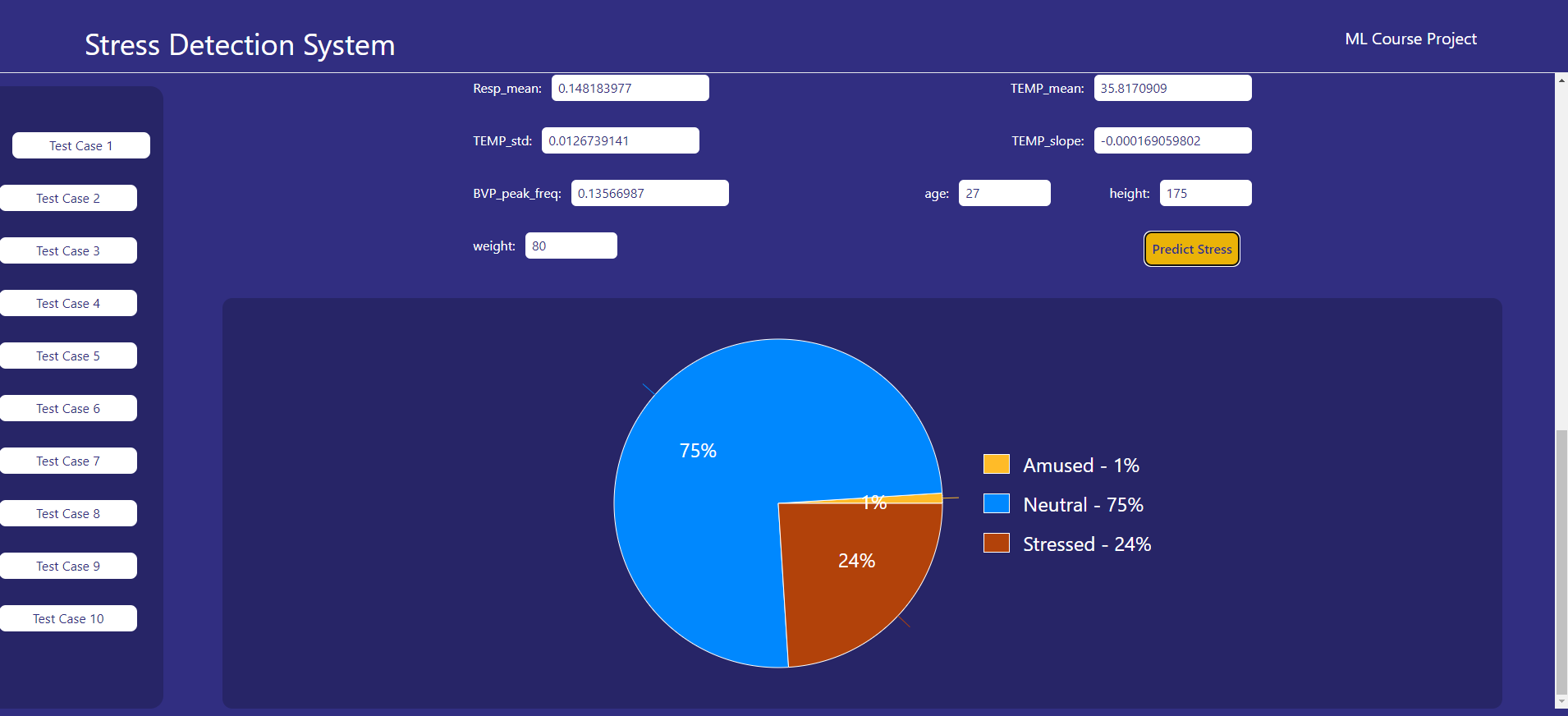
Here the input can be taken by filling the forms.







**Output Screen**





**CONCLUSION**

# CONCLUSION

In conclusion, the project "Investigation of mental stress detection with an 8-channel EEG system using KNN, SVM, and EEGNet" has demonstrated promising results in the field of stress detection. Through the utilization of advanced machine learning techniques and EEG signal analysis, we have successfully developed models capable of accurately detecting mental stress. While the project has shown success in certain aspects, there are areas for improvement, particularly in enhancing model performance and scalability. Moving forward, further research and development efforts will focus on refining the algorithms, expanding the dataset, and exploring real-time monitoring capabilities. Ultimately, the goal is to create a robust and reliable system that can effectively monitor and manage mental stress in various contexts, contributing to improved mental health and well-being for individuals worldwide.

.

**FUTURE SCOPE**

# FUTURE SCOPE

### Future scope :

The future scope of the project "Investigation of mental stress detection with an 8-channel EEG system using KNN, SVM, and EEGNet" is vast and holds significant potential for further research and development in the field of mental health monitoring and diagnosis.

Firstly, advancements in machine learning algorithms and EEG signal processing techniques could enhance the accuracy and efficiency of stress detection models. Exploring more sophisticated deep learning architectures, such as recurrent neural networks (RNNs) or convolutional neural networks (CNNs), could lead to better feature extraction and classification of EEG signals, thereby improving the overall performance of stress detection systems.

Additionally, expanding the dataset size and diversity could provide more robust models capable of generalizing across various demographics and stress-inducing scenarios. Collecting EEG data from a larger pool of participants spanning different age groups, genders, and cultural backgrounds would ensure the reliability and applicability of the developed models in real-world settings.

Furthermore, integrating real-time monitoring capabilities into the system could enable continuous stress tracking and timely intervention. This could involve the development of wearable EEG devices equipped with wireless connectivity to transmit data to centralized monitoring systems. Such advancements would facilitate early detection of stress episodes and enable personalized interventions tailored to individuals' needs.

Moreover, extending the application of EEG-based stress detection beyond clinical settings to various domains such as education, workplace environments, and sports could open up new avenues for research and innovation. Customizing the stress detection system to suit specific use cases and environments would require interdisciplinary collaboration between neuroscientists, psychologists, engineers, and domain experts.

Lastly, exploring the potential synergies between EEG-based stress detection and other physiological or behavioral indicators of stress could lead to more comprehensive and multi-modal stress assessment frameworks. Integrating data from sources such as heart rate variability, skin conductance, and facial expressions could enhance the accuracy and reliability of stress detection models, offering holistic insights into individuals' stress levels.

In conclusion, the future scope of the project encompasses advancements in algorithm development, data collection, real-time monitoring, interdisciplinary collaboration, and multi-modal integration, paving the way for more effective and personalized approaches to mental stress detection and management.

**REFERENCE**

# REFERENCE

[1] A. J. Y. Marthinsen, Detection of mental stress from eeg data using artificial

intelligence, 2022. DOI: https://doi.org/10.13140/RG.2.2.27754.39360.

[2] K. Rasheed, A. Qayyum, J. Qadir, S. Sivathamboo, P. Kwan, L. Kuhlmann, T.

O’Brien, and A. Razi, “Machine learning for predicting epileptic seizures us

ing eeg signals: A review,” IEEE Reviews in Biomedical Engineering, vol. 14,

pp. 139–155, 2020.

[3] X.-W. Wang, D. Nie, and B.-L. Lu, “Emotional state classification from eeg

data using machine learning approach,” Neurocomputing, vol. 129, pp. 94–

106, 2014.

[4] R. Katmah, F. Al-Shargie, U. Tariq, F. Babiloni, F. Al-Mughairbi, and H. Al

Nashash, “A review on mental stress assessment methods using eeg signal,”

Sensors, vol. 21, no. 15, 2021, ISSN: 1424-8220. DOI: https://doi.org/10.

3390/s21155043. [Online]. Available: https:/ /www.mdpi.com/1424-

8220/21/15/5043.

[5] O. Attallah, An effective mental stress state detection and evaluation system

using minimum number of frontal brain electrodes. DOI: https://10.3390/

diagnostics10050292. [Online]. Available: https://pubmed.ncbi.nlm.nih.

gov/32397517/.

[6] A. P. Assosiation, Stress effects on the body, 2018. [Online]. Available: https:

//www.apa.org/topics/stress/body.

[7] U. Malt and F. Svartdal, Stress, Store Norske Leksikon. [Online]. Available:

https://snl.no/stress.

[8] A. Crosswell and K. Lockwood, Best practices for stress measurement: How

to measure psychological stress in health research. DOI: https://doi.org/10.

[9] V. Aspiotis and et. al., Assessing electroencephalography as a stress indicator:

A vr high-altitude scenario monitored through eeg and ecg, Sensors, 2022.

DOI: https:/ /doi.org/10.3390/s22155792. [Online]. Available: https:

//www.mdpi.com/1424-8220/22/15/5792.

[10] D. F. Sharbrough, D. G.-E. Chatrian, D. R. P. Lesser, D. H. Lüders, D. M.

Nuwer, and D. T. W. Picton, American electricalencephalographic society guide

lines for standard electrode position nomenclature. [Online]. Available: https:

//journals.lww.com/clinicalneurophys/Citation/1991/04000/American\_

Electroencephalographic\_Society.7.aspx.

[11] N. Choudhury, P. Das, N. Deb, P. Dutta, R. Ghosh, S. Kashyap, S. Phadikar, A.

Phukan, R. Saha, K. Sengupta, and N. Sinha, Sam 40: Dataset of 40 subject

eeg recordings to monitor the induced-stress while performing stroop color

word test, arithmetic task, and mirror image recognition task. DOI: https:

//doi.org/10.1016/j.dib.2021.107772. [Online]. Available: https://www.

sciencedirect.com/science/article/pii/S2352340921010465?via%5C%

3Dihub.

[12] A. J. Newman, Artifacts in eeg data, Data Science for Psychology and Neu

roscience - in Python. [Online]. Available: https://neuraldatascience.io/7-

eeg/erp\_artifacts.html.

[13] A. J. Newman, Filtering eeg data, Data Science for Psychology and Neuro

science - in Python. [Online]. Available: https://neuraldatascience.io/7-

eeg/erp\_filtering.html.

[14] L. J. Gonçales, K. Farias, L. Kupssinskü, and M. Segalotto, “The effects of

applying filters on eeg signals for classifying developers’ code comprehen

sion,” DOI: 10.14482/INDES.30.1.303.661. [Online]. Available: https://

[www.redalyc.org/journal/474/47471710003](http://www.redalyc.org/journal/474/47471710003)