**A Project Report on**

# HUMAN ACTIVITY ANALYSIS USING MACHINE LEARNING CLASSIFICATION TECHNIQUES

***Submitted in the Partial Fulfillment of the Requirements of the Award of the Degree***

## BACHELOR OF TECHNOLOGY
## IN
## COMPUTER SCIENCE AND ENGINEERNG


**Submitted by**

M.Jyotsna(20W61A0552)

J.Sai(20W61A0532)

R.Harshavardan(20W61A0568)

S.Bhagyaraj(20W61A0574)

K.Satoshkumar(21W65A0504)

**Under the Esteemed Guidance of**

Mr M.Murali krishna

Assistant Professor




**Department of Computer Science & Engineering**

**Sri Sivani College of Engineering**

(Approved by AICTE, New Delhi,Affiliated to JNTU-GV, Vizianagaram

CC: W6,'A'Grade by AP Knowledge Mission, ISO 9001:2015 Certified campus)

Chilakapalem Jn,. Etcherla (M) Srikakulam (Dist), A.P,India.


**2020-2024**

# SRI SIVANI COLLEGE OF ENGINEERING

(Approved by AICTE, New Delhi, Permanently Affiliated to JNTU-GV,

'A' Grade by AP Knowledge Mission, ISO 9001:2015 Certified campus)

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

# Certificate

This is to certify that this project work entitled **"HUMAN ACTIVITY ANALYSIS USING MACHINE LEARNING CLASSIFICATION TECHNIQUES"** is the bonafide work carried out by **M.Jyotsna(20W61A0552), J.Sai(20W61A0532) ,R.Harshavardan(20W61A0568), S.Bhagyaraj(20W61A0574), K.Satoshkumar(21W65A0504)** submitted in partial fulfillment of the requirements for the Award of degree of Bachelor of Technology in Computer Science and Engineering, during the year 2020-2024.

**Signature of the guide**

**(Mr M.Murali Krishna)**

**Assistant Professor**

**Head of the Department**

**(Mr M.Murali krishna)**

**Assistant professor**

**External Examiner**

## Institute Vision

To be an institute of eminence, to produce highly Skilled, globally competent technocrats.

## Institute Mission

- Providing high quality, real world, industry relevant, career oriented Professional education to rural students towards their excellence and growth.
- Serving as a center of technical excellence, creating globally competent,human resources with ethical and moral values.

SRI SIVANI COLLEGE OF ENGINEERING
(Under the Management of Sri Sivani Educational Society, Srikakulam)
(Approved by AICTE, New Delhi and Affiliated to JNTUGV, Vizianagaram-CC-W6,
UGC Recognition under 2(f) & 12(B), ISO 9001:2015 Certified)
NH-16, Chilakapalem Jn., Srikakulam Dist. Andhra Pradesh -532410

## DEPARTMENT OF CSE

## VISION.

**Strive to be a center of eminence for developing holistic computer science professionals.**

## MISSION

**M1: To contribute to the advancement of knowledge, in both fundamental and applied Areas of computer Science Engineering.**

**M2: To promote a sense of excitement among the students in research, design, Development and entrepreneurship.**

**M3: To nurture communication skills, leadership, ethics and entrepreneurship among Students for their sustained growth.**

**M4: To forge mutuality beneficial relationship with industries and governmental entities to**

**Develop sustainable computing solutions for the benefits of the society.**

# ACKNOWLEDGEMENT

It is indeed with a great sense of pleasure and immense sense of gratitude that we acknowledge the help of these individuals. We feel elated in manifesting our sense of gratitude to our guide **M.Murali krishna**, Assistant Professor in the Department of Computer Science and Engineering for his valuable guidance. He has been a constant source of inspiration for us and we are very deeply thankful to him for his support and invaluable advice. We would like to thank our Head of the Department **M.Murali krishna**, for his constructive criticism throughout our project. We are highly indebted to Principal **Dr. Y Srinivasa Rao**, for the facilities provided to accomplish this project. We are extremely grateful to our Departmental staff members, lab technicians and non-teaching staff members for their extreme help throughout our project. Finally we express our heartfelt thanks to all of our friends who helped us in successful completion of this project.

## Project Member

M.Jyotsna(20W61A0552)

J.Sai(20W61A0532)

R.Harshavardan(20W61A0568)

S.Bhagyaraj(20W61A0574)

K.Satoshkumar(21W65A0504)

# DECLARATION

I do hereby declare that the work embodied in this project report entitled "**HUMAN ACTIVITY ANALYSIS USING MACHINE LEARNING CLASSIFICATION TECHNIQUES**" is the outcome of genuine research work carried out by me under the direct supervision of **M.Murali krishna** Asst prof, Department of Computer Science Engineering and is submitted by me to Sri Sivani College of Engineering. The work is original and has not been submitted elsewhere for the award of any other degree or diploma.


**Project Member**

M.Jyotsna(20W61A0552)

J.Sai(20W61A0532)

R.Harshavardan(20W61A0568)

S.Bhagyaraj(20W61A0574)

K.Satoshkumar(21W65A0504)

# ABSTRACT

Human activity recognition is gaining increasing importance because of its implication in remote monitoring application including security, health and fitness apps. This paper provides an analysis of different machine learning techniques for recognizing human activity. Firstly, all the recent work related to human activity recognition using accelerometer data is analyzed and presented in the paper. In this study the accelerometer used in smartphones as well as those embedded in wearable devices are compared and recognition methodologies applied on both the devices are presented. The dataset used in this project is a transformed version of "Activity Recognition using Cell Phone Accelerometers," by the Wireless Sensor Data Mining WSDM. Some important features were extracted from the data and based on it different models were assessed using python Classification Learner App. Four distinct machine learning techniques were applied on the dataset, namely, linear regression, logistic regression, support vector machine and neural network. For the purposed of applying classifier Weka tool is used. The results of these algorithms are compared and presented in the form of tables and graphs and Bagged Tree is identified to be the best algorithm based on accuracy results.

**Keywords:** Activity analysis, classification techniques , smartphone data.

# TABLE OF CONTENT

# CHAPTER-1

# INTRODUCTION

Human Activity Recognition is an engaging topic of research since quite some time but still has a lot of space for improvement. Various method presented by researchers through which human activity can be recognized are based on computer vision using RGB camera, sensors like Wifi sensors, magnetometers, gyroscope and accelerometer sensors. Recognizing activity through computer vision methods has limitations due to lightening, accurate foreground extraction technique and most importantly the person to be monitored needs to be in camera range of vision. Monitoring human activity of a more mobile person is far easier and pervasive through accelerometer data as it is embedded on a mobile device like smart phone or some other device and is easier to carry around. By using accelerometer data on a smart phone, a person's activity can be stored ubiquitously and sent to a server where it can be processed to recognize activity. an overview of the whole process. HAR is becoming more and more interesting in many application. Assistive technology like eldercare, storing calories lost based on exercise done in fitness app, and monitoring human gait in security application are few of amazing applications of HAR. Major challenge in HAR is that human body activities are very complex and its real-time recognition is very difficult. To overcome this challenge machine learning is applied on sensors data like accelerometer data. In this paper application of different methods of machine learning is analyzed and presented. Different researchers have explored and presented different ways of recognizing and classifying human activity. The data is processed and classified into predefined learnt classes like walking, climbing, running etc. In any application of machine learning, feature selection is one of the most important step for creating a classifier. The features which are more informative, needs be selected before classification for generalized results. The features selected for classification are mean, standard deviation, peak and resultant values. These features help classify data from the tri-axial accelerometer, x, y, and z values into classes of walking, standing, running, sitting etc. The rest of the paper is organized as follows. Section 2 presents a literature review of the recent work done on human activity recognition through accelerometer data. First different techniques applied on smartphone accelerometer data are discussed and then the work done recognizing activity from wearable accelerometer data . the data set used in experimentation in detail. presents classification done on python

Classification Learning App, its analysis and results. The complete methodology of developing classifier, graphical analysis and results is presented

**IDENTIFICATION/ NEED:**

Human activity analysis using smartphone datasets and machine learning classification techniques addresses several critical needs and challenges across various domians. Below are some key points outlining the identification of the need for this approach:

**Health Monitoring and wellness Management:** Human activity analysis can provide valuable insights into individuals' physical activities, sleep patterns, and overall lifestyle, enabling early detection of health issues and personalized interventions.

Leveraging smartphone sensor data for activity analysis offers a non-intrusive and cost-effective way to monitor individuals' health status in real-time.

**Fitness Tracking and Performance Improvement:**

- In the realm of fitness and sports, understanding human activities is essential for tracking workouts, measuring performance, and setting fitness goals.
- By analyzing smartphone sensor data, users can receive real-time feedback on their exercise routines, including the type, intensity, and duration of activities performed.
- Machine learning classification techniques can enhance the accuracy of activity recognition, providing users with more reliable insights into their fitness progress and areas for improvement.
- Context-aware Computing and Personalized Services:
- Context-aware computing aims to adapt system behavior based on users' context, including their location, activity, and preferences.
- Human activity analysis enables smartphones and other smart devices to infer users' activities and context, allowing for personalized services and recommendations.
- For example, smartphones can automatically adjust settings or suggest relevant apps based on users' current activities (e.g., driving, walking, or working out).

**Enhanced User Experience and Interaction:**

Understanding human activities can significantly enhance the user experience and interaction with digital devices and applications.

**Safety and Security Applications:**

- Human activity analysis has applications in safety and security, including fall detection, abnormal behavior detection, and emergency response.
- By analyzing smartphone sensor data, systems can detect deviations from normal activity patterns and trigger alerts or interventions in case of emergencies.
- This technology is particularly beneficial for vulnerable populations such as the elderly or individuals with disabilities, providing them with an added layer of safety and support.

**Research and Innovation in Human-Computer Interaction:**

- Human activity analysis using smartphone datasets fuels research and innovation in human-computer interaction (HCI) and pervasive computing.
- Understanding how humans interact with digital devices and environments can inspire the design of more intuitive interfaces and adaptive systems.
- By leveraging machine learning classification techniques, HCI researchers can develop smarter and more context-aware interfaces that anticipate users' actions and preferences

## 1.1 OBJECTIVE:

The objective of a project focusing on human activity analysis using machine learning algorithms could be multifaceted, depending on the specific goals and applications. Here are some possible objectives for such a project:

**Activity Recognition and Classification:** The primary objective could be to accurately recognize and classify human activities based on sensor data. This involves developing machine learning models capable of distinguishing between different activities such as walking, running, sitting, standing, and more.

**Real-Time Monitoring:** Another objective could be to enable real-time monitoring of human activities using sensors such as accelerometers, gyroscopes, and GPS. The project may aim to develop algorithms capable of processing sensor data in real-time and providing timely feedback or alerts based on detected activities.

**Healthcare and Fitness Tracking:** If the project is focused on healthcare or fitness applications, the objective could be to develop a system that monitors and tracks physical activities to provide insights into users' health and fitness levels. This could involve

analyzing activity patterns over time to detect changes or anomalies that may indicate health issues or improvements in fitness.

**Improving Accuracy and Robustness:** A key objective could be to improve the accuracy and robustness of activity recognition algorithms, especially in challenging real-world environments where sensor data may be noisy or incomplete. This could involve exploring advanced machine learning techniques, such as deep learning and sensor fusion, to enhance performance.

**User Experience and Interaction:** The project may also aim to enhance the user experience and interaction with the activity recognition system. This could involve developing intuitive user interfaces, designing feedback mechanisms, or incorporating personalized recommendations based on users' activity patterns and goals.

**Privacy and Security:** Another objective could be to address privacy and security concerns associated with collecting and analyzing human activity data. This may involve implementing privacy-preserving techniques such as federated learning, differential privacy, or data anonymization to protect users' sensitive information.

**Scalability and Deployment:** Finally, the project may have objectives related to scalability and deployment, aiming to develop a system that can handle large-scale data streams from multiple users and devices. This could involve optimizing algorithms for efficiency and scalability, as well as designing architectures that support distributed deployment in cloud or edge computing environments.

Overall, the objective of the project would likely be a combination of these factors, tailored to the specific goals, requirements, and applications of the human activity analysis system.

**ADVANTAGES:**

**Cost-Effective Solution**:

- Compared to traditional activity monitoring systems that require specialized hardware or sensors, smartphone-based solutions are cost-effective and accessible.
- Users already own smartphones, eliminating the need for additional investments in equipment or devices.

**Real-Time Feedback:**

- By analyzing sensor data in real-time, smartphone-based activity analysis systems can provide immediate feedback and insights to users.
- This real-time feedback can be valuable for guiding users' behavior, encouraging physical activity, or alerting them to potential health risks.

**Scalability and Accessibility:**

- Smartphone-based activity analysis systems are highly scalable and accessible, capable of reaching large numbers of users across different geographic locations.
- With the widespread adoption of smartphones globally, these systems can be deployed and accessed by users with minimal barriers to entry

## 1.2 Existing System

Analyzing human activity using machine learning algorithms has a wide range of applications, from healthcare and fitness tracking to surveillance and security. Let's discuss both an existing system and a proposed system for human activity analysis using machine learning:

Activity Recognition Using Smartphones Dataset

This dataset is a widely-used benchmark for activity recognition research.

It contains data collected from smartphone sensors (accelerometer and gyroscope) while subjects perform six different activities: walking, walking upstairs, walking downstairs, sitting, standing, and laying.

Researchers have applied various machine learning algorithms such as Support Vector Machines (SVM), Random Forest, and Neural Networks to classify activities based on sensor data.

The existing system involves preprocessing the sensor data, extracting features (e.g., mean, standard deviation), and training a machine learning model to recognize activities.

### 1.3 Proposed System:

Sensor Fusion for Enhanced Activity Recognition:

The proposed system integrates data from multiple sensors (e.g., accelerometer, gyroscope, GPS) to improve activity recognition accuracy.In addition to smartphone sensors, wearable devices such as smartwatches or fitness trackers could be used to gather additional data. Sensor fusion techniques, such as combining data from multiple sensors using techniques like Kalman filtering or Bayesian fusion, can provide more robust and accurate activity recognition. Machine learning algorithms such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) could be employed to learn complex patterns from the fused sensor data.

**Deep Learning for Sequential Activity Recognition:**

Instead of treating each activity recognition instance as an independent sample, the proposed system considers the sequential nature of human activities.

Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks can be trained to model temporal dependencies in human activity data.

By considering the sequence of sensor readings over time, the model can capture contextual information and make more accurate predictions.

Additionally, attention mechanisms can be incorporated into the model to focus on the most relevant parts of the sensor data sequence.

**Privacy-Preserving Activity Recognition:**

Privacy concerns are paramount when dealing with human activity data, especially if it involves personal information.

The proposed system includes techniques for privacy-preserving activity recognition, such as federated learning or differential privacy.

Federated learning allows the model to be trained on decentralized data sources without sharing raw data, preserving the privacy of individual users.

Differential privacy adds noise to the training data or model parameters to prevent the leakage of sensitive information.

In summary, while existing systems for human activity analysis using machine learning have made significant progress, there are still opportunities for improvement and innovation, especially in areas such as sensor fusion, deep learning, and privacy preservation

# CHAPTER – 2
# LITERATURE SURVEY

## 2.1 Introduction

In our survey, we have gone through previous year papers to know about the human activity data briefly from different concerns and to know about different technologies, methods as well as methodologies used in their own survey.

## 2.2 Survey on Human Activity Analysis data using various algorithms

Smartphone sensors like accelerometer and gyroscope have paved the way for more and more sensor based applications like Human Activity recognition for the welfare of people. Simple daily life activities are recognized and based on it more complex activities can be identified. In the paper, a complex activity like Muslim prayer is recognized by firstly collecting data, preprocessing the data collected, extracting important features, classifying into classes and finally aggregation . In another paper machine learning technique is introduced which uses ensemble manifold rank preserving (EMRP) algorithm to predict human activity on two datasets, namely, the SCUT NAA dataset and the NMHA .

Convolution Neural Networks technique is used for classification and the datasets used were Actitracker dataset along with Skoda and Opportunity datasets [4]. Automatic extraction of discriminating features is described in the paper [5].The data taken is recorded from mobile phone accelerometer which is placed in person's pocket and hand. In the paper [6], different classifier were analyzed and five of the best, high performance classifiers were selected. The results of the proposed model were up to 91.15% accurate. An Algorithm called LHUC (Learning Hidden Unit Contributions) which is based Neural Networks is applied in Smartphone sensor.

Other than mobile phone accelerometers, accelerometers are also sometimes embedded in other wearable devices with the benefit that it is not as sensitive to damage as smart phones and can be used by children and elderly people care. In a paper machine learning technique called the random forest is applied on data recorded from accelerometer sensors embedded in a wearable device [7]. The results of the experimentation were up to 94% accurate. In the paper, it is described how features selection mechanism is applied on statistically

derived features [8]. Human activity is recognize using a wearable device which is described in the paper.

Feature selection is a critical part of machine learning methodology. In the paper those features are extracted and presented that helps classify physical activities better [9]. 94% of accuracy is obtained using the Random Forest algorithm. Numerous activities are recognized using a wearable accelerometer and classified into classes of activities and sub-activities successfully. Using K-nearest neighbor classification six activities were correctly classified into the class of the type of activity being performed. In another [10] work deep learning technique is used to for activity recognition. Firstly useful features are extracted automatically and computational cost is decreased. A wearable device is worn on arm which has a tri-axial accelerometer embedded in it which recognized the movement of the arm. Different techniques of human activity recognition are studied and analyzed in various different works. In a work human activity recognition techniques are discussed which are used in both wearable as well as smartphones [11].

Different techniques for activity recognition, including temporal pattern mining and ANFIS are proposed. The techniques discussed in the paper are evaluate for both wearable and smartphone accelerometer data and gave accurate results.

Human motion analysis is receiving increasing attention from computer vision researchers. This interest is motivated by a wide spectrum of applications, such as athletic performance analysis, surveillance, man–machine interfaces, content-based image storage and retrieval, and video conferencing. This paper gives an overview of the various tasks involved in motion analysis of the human body. We focus on three major areas related to interpreting human motion: (1) motion analysis involving human body parts, (2) tracking a moving human from a single view or multiple camera perspectives, and (3) recognizing human activities from image sequences. Motion analysis of human body parts involves the low-level segmentation of the human body into segments connected by joints and recovers the 3D structure of the human body using its 2D projections over a sequence of images. Tracking human motion from a single view or multiple perspectives focuses on higher-level processing, in which moving humans are tracked without identifying their body parts. After successfully matching the moving human image from one frame to another in an

image sequence, understanding the human movements or activities comes naturally, which leads to our discussion of recognizing human activities.

Human activity recognition is an important area of computer vision research. Its applications include surveillance systems, patient monitoring systems, and a variety of systems that involve interactions between persons and electronic devices such as human-computer interfaces. Most of these applications require an automated recognition of high-level activities, composed of multiple simple (or atomic) actions of persons. This article provides a detailed overview of various state-of-the-art research papers on human activity recognition. We discuss both the methodologies developed for simple human actions and those for high-level activities. An approach-based taxonomy is chosen that compares the advantages and limitations of each approach. Recognition methodologies for an analysis of the simple actions of a single person are first presented in the article. Space-time volume approaches and sequential approaches that represent and recognize activities directly from input images are discussed. Next, hierarchical recognition methodologies for high-level activities are presented and compared. Statistical approaches, syntactic approaches, and description-based approaches for hierarchical recognition are discussed in the article. In addition, we further discuss the papers on the recognition of human-object interactions and group activities. Public datasets designed for the evaluation of the recognition methodologies are illustrated in our article as well, comparing the methodologies' performances. This review will provide the impetus for future research in moreproductive areas

Visual analysis of human motion is currently one of the most active research topics in computer vision. This strong interest is driven by a wide spectrum of promising applications in many areas such as virtual reality, smart surveillance, perceptual interface, etc. Human motion analysis concerns the detection, tracking and recognition of people, and more generally, the understanding of human behaviours , from image sequences involving humans. This paper provides a comprehensive survey of research on computer- vision-based human motion analysis. The emphasis is on three major issues involved in a general human motion analysis system, namely human detection, tracking and activity understanding. Various methods for each issue are discussed in order to examine the state of the art. Finally, some research challenges and future directions are discussed.

## Overview

In this paper a method for automatic detection of human activity, given a mobile phone accelerometer data, is presented. Target is to develop a classifier that gives best results using Python platform and Python Classification Learner App. The strategy is trained and tested on the data set, Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine. The data set consists of gyroscope and accelerometer data recorded for activities like, walking, climbing stairs, siting, standing and laying.

B. Pre-Processing of Training Data First the raw sensor train and test data were loaded from the data set folder. The data in training .txt file is converted into tables and saved as .mat files for later access. The raw sensor training data is analyzed by plotting the accelerometer x y and z axis on three different graphs. In pre-processing of data mean is taken for every 128 points as there were 128 readings per window in the raw training data.

C. Additional Feature Extraction For better learning some more features were extracted from the training data including, mean, standard deviation and principal component analysis. These features were used for learning

D. Using Classification Learner for Training a Model and Assessing its Performance python Classification Learner Application is used for automated training of different models which could be used to classifythe test data.

## 2.1 SYSTEM REQUIREMENT SPECIFICATION

**Software Interface**

OperatingSystem : WindowsXP/2007/8/10

Programming Language : Python

**Hardware Interface**

Processor : Pentium IV 24 GHZ or more

Hard disk : 40 – 100 GB

RAM : 512MB minimum

Key Board : Standard Windows Keyboard

Mouse : Two or Three Button Mouse

Monitor : SVGA

# CHAPTER-3

# FEASIBILITY STUDY

Feasibility study is conducted once the problem is clearly understood. The feasibility study which is a high level capsule version of the entire system analysis and design process. The objective is to determine whether the proposed system is feasible or not and it helps us to the minimum expense of how to solve the problem and to determine, if the Problem is worth solving. The following are the three important tests that have been carried out for feasibility study.

This study tells about how this package is useful to the users and its advantages and disadvantages, and also it tells whether this package is cost effective are not. There are three types of feasibility study, they are

Economic Feasibility.

Technical Feasibility.

Operational Feasibility.

## Software Environment

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, https://www.python.org/, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well.

For a description of standard objects and modules, see The Python Standard Library. The Python Language Reference gives a more formal definition of the language. To write extensions in C or C++, read Extending and Embedding the Python Interpreter and Python/C API Reference Manual. There are also several books covering Python in depth.

This tutorial does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. Instead, it introduces many of Python's most noteworthy features, and will give you a good idea of the language's flavor and style. After reading it, you will be able to read and write Python modules and programs, and you will be ready to learn more about the various Python library modules described in The Python Standard Library.

Python is a high level general purpose open source programming language. It is both object oriented and procedural. Python is an extremely powerful language. This language is very easy to learn and is a good choice for most of the professional programmers. Python is invented by Guido Van Rossum at CWI in Netherland in 1989. It is binding of C, C++, Java. It also provides a library for GUI.

**Python Features and Characteristics:**

- Python is a high level, open source, general purpose programming language.
- It is object oriented, procedural and functional.
- It has library to support GUI.
- It is extremely powerful and easy to learn.
- It is open source, so free to available for everyone.
- It supports on Windows, Linux and Mac OS.

❖ As code is directly compiled with byte code, python is suitable for use in scripting languages.

❖ Python enables us to write clear, logical applications for small and large tasks.

❖ It has high level built-in data types: string, lists, dictionaries etc.

❖ Python has a set of useful Libraries and Packages that minimize the use of code in our day to day life, like- Django, Tkinter, OpenCV, NextworkX, lxml.

❖ It plays well with java because of python. Python is a version of Python.

❖ It encourages us to write clear and well structured code.

❖ It is easy to interface with C, Objective C, java, etc.

❖ We can easily use Python with other languages. It has different varieties of languages like-

• CPython – Python implemented in C.

• Jython – Python implemented in java Environment. 38

• PYPY – Python with JIT compiler and stackless mode. 4.2.2

**WHY CHOOSE PYTHON**

If you're going to write programs, there are literally dozens of commonly used languages to choose from. Why choose Python? Here are some of the features that make Python an appealing choice. Python is popular:- Python has been growing in popularity over the last few years. The 2018 Stack Overflow Developer Survey ranked Python as the 7th most popular and the number one most wanted technology of the year. World-class software development countries around the globe use Python every single day. According to research by Dice Python is also one of the hottest skills to have and the most popular programming language in the world based on the Popularity of Programming Language Index. Due to the popularity and widespread use of Python as a programming language, Python developers are sought after and paid well. If you'd like to dig deeper into Python salary statistics and job opportunities, you can do so here. Python is interpreted:- Many languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer's processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly. This makes for a quicker development cycle because you just type in your code and run it, without the intermediate compilation step. One potential downside to

interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. For some applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting. 39 In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The expediency of coding in an interpreted language is typically worth it for most applications. Python is Free:- The Python interpreter is developed under an OSI-approved open-source license, making it free to install, use, and distribute, even for commercial purposes. A version of the interpreter is available for virtually any platform there is, including all flavours of Unix, Windows, MACOS, smart phones and tablets, and probably anything else you ever heard of. A version even exists for the half dozen people remaining who use OS/2. Python is Portable:- Because Python code is interpreted and not compiled into native machine instructions, code written for one platform will work on any other platform that has the Python interpreter installed. (This is true of any interpreted language, not just Python.) Python is Simple:- As programming languages go, Python is relatively uncluttered, and the developers have deliberately kept it that way. A rough estimate of the complexity of a language can be gleaned from the number of keywords or reserved words in the language. These are words that are reserved for special meaning by the compiler or interpreter because they designate specific built-in functionality of the language. Python 3 has 33 keywords, and Python 2 has 31. By contrast, C++ has 62, Java has 53, and Visual Basic has more than 120, though these latter examples probably vary somewhat by implementation or dialect. Python code has a simple and clean structure that is easy to learn and easy to read. In fact, as you will see, the language definition enforces code structure that is easy to read. 40 For all its syntactical simplicity, Python supports most constructs that would be expected in a very high-level language, including complex dynamic data types, structured and functional programming, and object-oriented programming. Additionally, a very extensive library of classes and functions is available that provides capability well beyond what is built into the language, such as database manipulation or GUI programming. Python accomplishes what many programming languages don’t: the language itself is simply designed, but it is very versatile in terms of what you can accomplish with it.

**Conclusion:-** This section gave an overview of the Python programming language, including:

• A brief history of the development of Python

• Some reasons why you might select Python as your language of choice Python is a great option, whether you are a beginning programmer looking to learn the basics, an experienced programmer designing a large application, or anywhere in between. The basics of Python are easily grasped, and yet its capabilities are vast. Proceed to the next section to learn how to acquire and install Python on your computer. Python is an open source programming language that was made to be easy-to-read and powerful. A Dutch programmer named Guido van Rossum made Python in 1991. He named it after the television show Monty Python's Flying Circus. Many Python examples and tutorials include jokes from the show. Python is an interpreted language. Interpreted languages do not need to be compiled to run. A program called an interpreter runs Python code on almost any kind of computer. This means that a programmer can change the code and quickly see the results. This also means Python is slower than a compiled language like C, because it is not running machine code directly. Python is a good programming language for beginners. It is a high-level language, which means a programmer can focus on what to do instead of how to do it. Writing programs in Python takes less time than in some other languages. 41 Python drew inspiration from other programming languages like C, C++, Java, Perl, and Lisp. Python has a very easy-to-read syntax. Some of Python's syntax comes from C, because that is the language that Python was written in. But Python uses whitespace to delimit code: spaces or tabs are used to organize code into groups. This is different from C. In C, there is a semicolon at the end of each line and curly braces ({}) are used to group code. Using whitespace to delimit code makes Python a very easy-to-read language. Python use [change / change source]:- Python is used by hundreds of thousands of programmers and is used in many places. Sometimes only Python code is used for a program, but most of the time it is used to do simple jobs while another programming language is used to do more complicated tasks. Its standard library is made up of many functions that come with Python when it is installed. On the Internet there are many other libraries available that make it possible for the Python language to do more things. These libraries make it a powerful language; it can do many different things. Some things that Python is often used for are:

• Web development

• Scientific programming

• Desktop GUIs

• Network programming

• Game programming

**The Python Standard Library**

While The Python Language Reference describes the exact syntax and semantics of the Python language, this library reference manual describes the standard library that is distributed with Python. It also describes some of the optional components that are commonly included in Python distributions.

Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below. The library contains built-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. Some of these modules are explicitly designed to encourage and enhance the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs.

The Python installers for the Windows platform usually include the entire standard library and often also include many additional components. For Unix-like operating systems Python is normally provided as a collection of packages, so it may be necessary to use the packaging tools provided with the operating system to obtain some or all of the optional components

## Dealing with Bugs

Python is a mature programming language which has established a reputation for stability. In order to maintain this reputation, the developers would like to know of any deficiencies you find in Python.

It can be sometimes faster to fix bugs yourself and contribute patches to Python as it streamlines the process and involves less people. Learn how to contribute.

## Documentation bugs

If you find a bug in this documentation or would like to propose an improvement, please submit a bug report on the tracker. If you have a suggestion how to fix it, include that as well.

If you're short on time, you can also email documentation bug reports to docs@python.org (behavioral bugs can be sent to python-list@python.org). 'docs@' is a mailing list run by volunteers; your request will be noticed, though it may take a while to be processed.

**Documentation bugs on the Python issue tracker**

**Using the Python issue tracker**

Bug reports for Python itself should be submitted via the Python Bug Tracker (https://bugs.python.org/). The bug tracker offers a Web form which allows pertinent information to be entered and submitted to the developers.

The first step in filing a report is to determine whether the problem has already been reported. The advantage in doing so, aside from saving the developers time, is that you learn what has been done to fix it; it may be that the problem has already been fixed for the next release, or additional information is needed (in which case you are welcome to provide it if you can!). To do this, search the bug database using the search box on the top of the page.

If the problem you're reporting is not already in the bug tracker, go back to the Python Bug Tracker and log in. If you don't already have a tracker account, select the "Register" link or, if you use OpenID, one of the OpenID provider logos in the sidebar. It is not possible to submit a bug report anonymously.

Being now logged in, you can submit a bug. Select the "Create New" link in the sidebar to open the bug reporting form.

The submission form has a number of fields. For the "Title" field, enter a *very* short description of the problem; less than ten words is good. In the "Type" field, select the type of your problem; also select the "Component" and "Versions" to which the bug relates.

In the "Comment" field, describe the problem in detail, including what you expected to happen and what did happen. Be sure to include whether any extension modules were

involved, and what hardware and software platform you were using (including version information as appropriate).

Each bug report will be assigned to a developer who will determine what needs to be done to correct the problem. You will receive an update each time action is taken on the bug.

**Introduction to Data Mining**

Data mining integrates approaches and techniques from various disciplines such as machine learning, statistics, artificial intelligence, neural networks, database management, data warehousing, data visualization, spatial data analysis, probability graph theory etc. In short, data mining is a multi-disciplinary field.

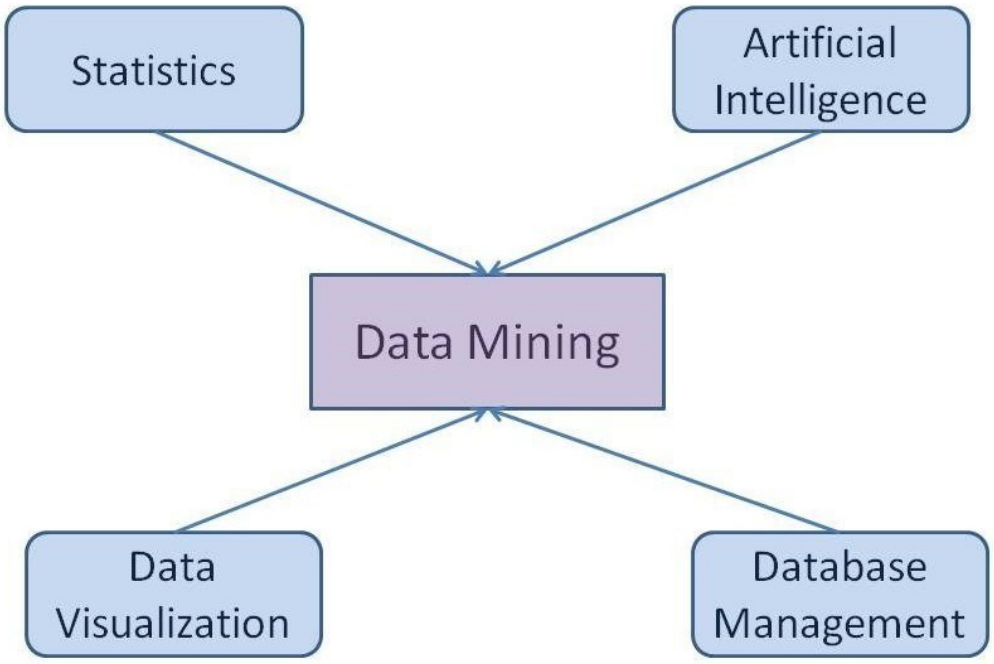



Figure 1

**Statistics**

Statistics includes a number of methods to analyze numerical data in large quantities. Different statistical tools used in data mining are regression analysis, cluster analysis, correlation analysis and Bayesian network. Statistical models are usually built from a training data set. Correlation analysis identifies the correlation of variables to each other. Bayesian network is a directed graph that represents casual relationship among data found out using the Bayesian probability theorem. Given below is a simple Bayesian network where the nodes represent variables whereas edges represent the relationship between the nodes.

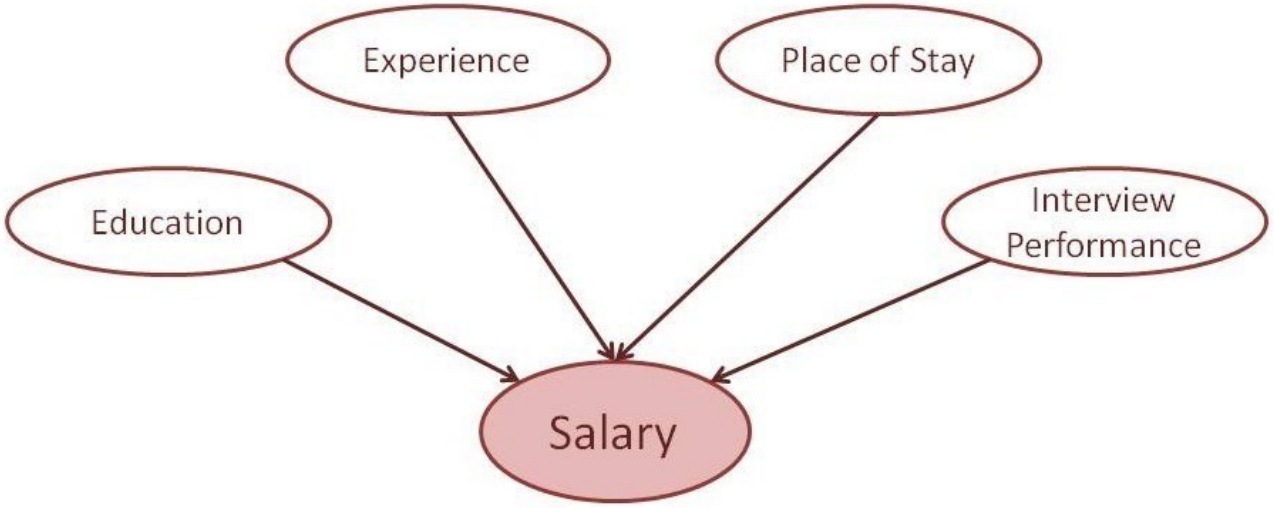



Figure 2

## Machine Learning

Machine learning is the collection of methods, principles and algorithms that enables learning and prediction on the basis of past data. Machine learning is used to build new models and to search for a best model matching the test data. Machine learning methods normally use heuristics while searching for the model. Data mining uses a number of machine learning methods including inductive concept learning, conceptual clustering and decision tree induction. A decision tree is a classification tree that decides the class of an object by following the path from the root to a leaf node. Given below is a simple decision tree that is used for weather forecasting.

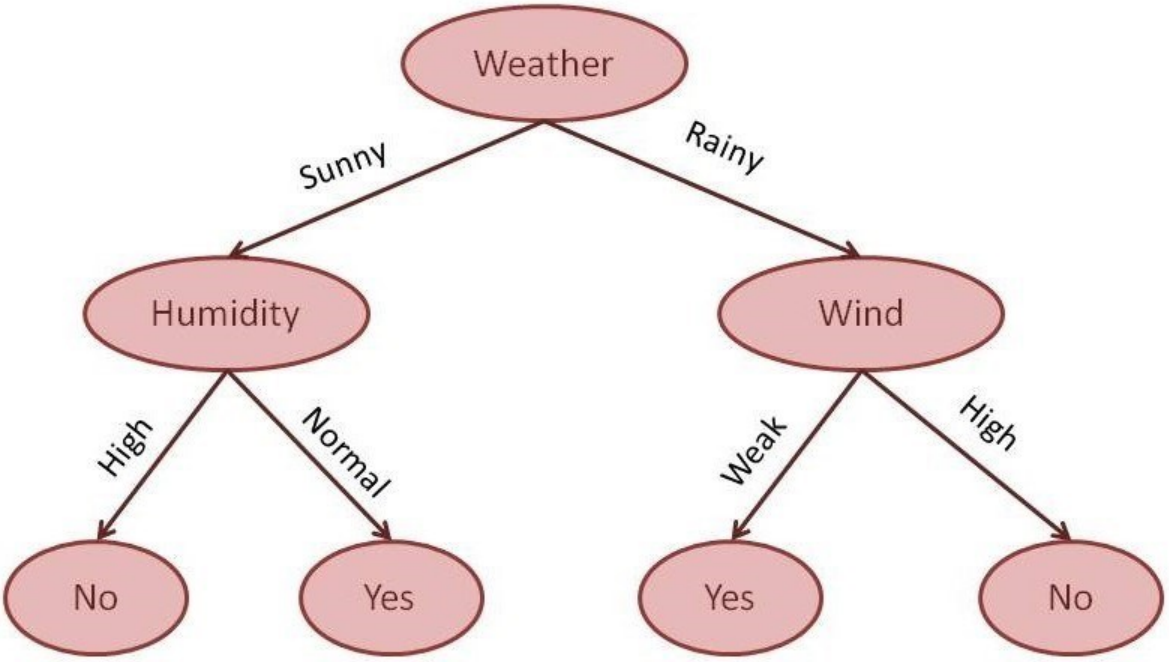



Figure 3

**Database Oriented Techniques**

Advancements in database and data warehouse implementation helps data mining in a number of ways. Database oriented techniques are used mainly to develop characteristics of the available data. Iterative database scanning for frequent item sets, attribute focusing, and attribute oriented induction are some of the database oriented techniques widely used in data mining. The iterative database scanning searches for frequent item sets in a database. Attribute oriented induction generalizes low level data into high level concepts using conceptual hierarchies.

## Neural Networks

A neural network is a set of connected nodes called neurons. A neuron is a computing device that computes some requirement of its inputs and the inputs can even be the outputs of other neurons. A neural network can be trained to find the relationship between input attributes and output attribute by adjusting the connections and the parameters of the nodes.

## Data Visualization

The information extracted from large volumes of data should be presented well to the end user and data visualization techniques make this possible. Data is transformed into different visual objects such as dots, lines, shapes etc and displayed in a two or three dimensional space. Data visualization is an effective way to identify trends, patterns, correlations and outliers from large amounts of data.

Summary

Data mining combines different techniques from various disciplines such as machine learning, statistics, database management, data visualization etc. These methods can be combined to deal with complex problems or to get alternative solutions. Normally data mining system employs one or more techniques to handle different kinds of data, different data mining tasks, different application areas and different data requirements.

## Patterns in Data Mining

**1. Association**

The items or objects in relational databases, transactional databases or any other information repositories are considered, while finding associations or correlations.

**2. Classification**

The goal of classification is to construct a model with the help of historical data that can accurately predict the value.

It maps the data into the predefined groups or classes and searches for the new patterns.

For example:
To predict weather on a particular day will be categorized into - sunny, rainy, or cloudy.

**3. Regression**

Regression creates predictive models. Regression analysis is used to make predictions based on existing data by applying formulas.

Regression is very useful for finding (or predicting) the information on the basis of previously known information.

**4. Cluster analysis**

It is a process of portioning a set of data into a set of meaningful subclass, called as cluster.

It is used to place the data elements into the related groups without advanced knowledge of the group definitions.

**5. Forecasting**

Forecasting is concerned with the discovery of knowledge or information patterns in data that can lead to reasonable predictions about the future.

Technologies used in data mining

Several techniques used in the development of data mining methods.
Some of them Are mentioned below:

**1. Statistics:**

It uses the mathematical analysis to express representations, model and summarize empirical data or real world observations.

Statistical analysis involves the collection of methods, applicable to large amount of data to conclude and report the trend.

**2. Machine learning**

Arthur Samuel defined machine learning as a field of study that gives computers the ability to learn without being programmed.

When the new data is entered in the computer, algorithms help the data to grow or change due to machine learning.

In machine learning, an algorithm is constructed to predict the data from the available database (Predictive analysis).

It is related to computational statistics.

The four types of machine learning are:

**1. Supervised learning**

It is based on the classification.It is also called as inductive learning. In this method, the desired outputs are included in the training dataset.

**2. Unsupervised learning**

Unsupervised learning is based on clustering. Clusters are formed on the basis of similarity measures and desired outputs are not included in the training dataset.

**3. Semi-supervised learning**

Semi-supervised learning includes some desired outputs to the training dataset to generate the appropriate functions. This method generally avoids the large number of labeled examples (i.e. desired outputs) .

**4. Active learning**

Active learning is a powerful approach in analyzing the data efficiently.

The algorithm is designed in such a way that, the desired output should be decided by the algorithm itself (the user plays important role in this type).

## 3. Information retrieval

Information deals with uncertain representations of the semantics of objects (text, images). For example: Finding relevant information from a large document.

## 4. Database systems and data warehouse

Databases are used for the purpose of recording the data as well as data warehousing.

Online Transactional Processing (OLTP) uses databases for day to day transaction purpose.

To remove the redundant data and save the storage space, data is normalized and stored in the form of tables.

Entity-Relational modeling techniques are used for relational database management system design.

Data warehouses are used to store historical data which helps to take strategical decision for business.

It is used for online analytical processing (OALP), which helps to analyze the data.

**5.** Decision support system

Decision support system is a category of information system. It is very useful in decision making for organizations.

It is an interactive software based system which helps decision makers to extract useful information from the data, documents to make the decision.

# CHAPTER-4

# SYSTEM ANALYSIS

## 4.SYSTEM ANALYSIS

## 4.1 THE STUDY OF THE SYSTEM

To conduct studies and analyses of an operational and technological nature, and

To promote the exchange and development of methods and tools for operational analysis as applied to defence problems.

**Logical design**

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modeling, using an over-abstract (and sometimes graphical) model of the actual system. In the context of systems design are included. Logical design includes ER Diagrams i.e. Entity Relationship Diagrams

**Physical design**

The physical design relates to the actual input and output processes of the system. This is laid down in terms of how data is input into a system, how it is verified / authenticated, how it is processed, and how it is displayed as output. In Physical design, following requirements about the system are decided.

Input requirement,

Output requirements,

Storage requirements,

Processing Requirements,

System control and backup or recovery.

Put another way, the physical portion of systems design can generally be broken down into three sub-tasks:

User Interface Design

Data Design

Process Design

User Interface Design is concerned with how users add information to the system and with how the system presents information back to them. Data Design is concerned with how the data is represented and stored within the system. Finally, Process Design is concerned with how data moves through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system. At the end of the systems design phase, documentation describing the three sub-tasks is produced and made available for use in the next phase.

Physical design, in this context, does not refer to the tangible physical design of an information system. To use an analogy, a personal computer's physical design involves input via a keyboard, processing within the CPU, and output via a monitor, printer, etc. It would not concern the actual layout of the tangible hardware, which for a PC would be a monitor, CPU, motherboard, hard drive, modems, video/graphics cards, USB slots, etc. It involves a detailed design of a user and a product database structure processor and a control processor. The H/S personal specification is developed for the proposed system.

## 4.2 INPUT & OUTPUT REPRESENTATION

**Input Design**

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

so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

What data should be given as input?

How the data should be arranged or coded?

The dialog to guide the operating personnel in providing input.

Methods for preparing input validations and steps to follow when error occur.

## Objectives

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

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

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

**Output Design**

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

Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that

people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

Select methods for presenting information.

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

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

Convey information about past activities, current status or projections of the future. Signal

important events, opportunities, problems, or warnings.

Trigger an action.

Confirm an action.

## 4.3 PROCESS MODEL USED WITH JUSTIFICATION

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

SDLC (Spiral Model):

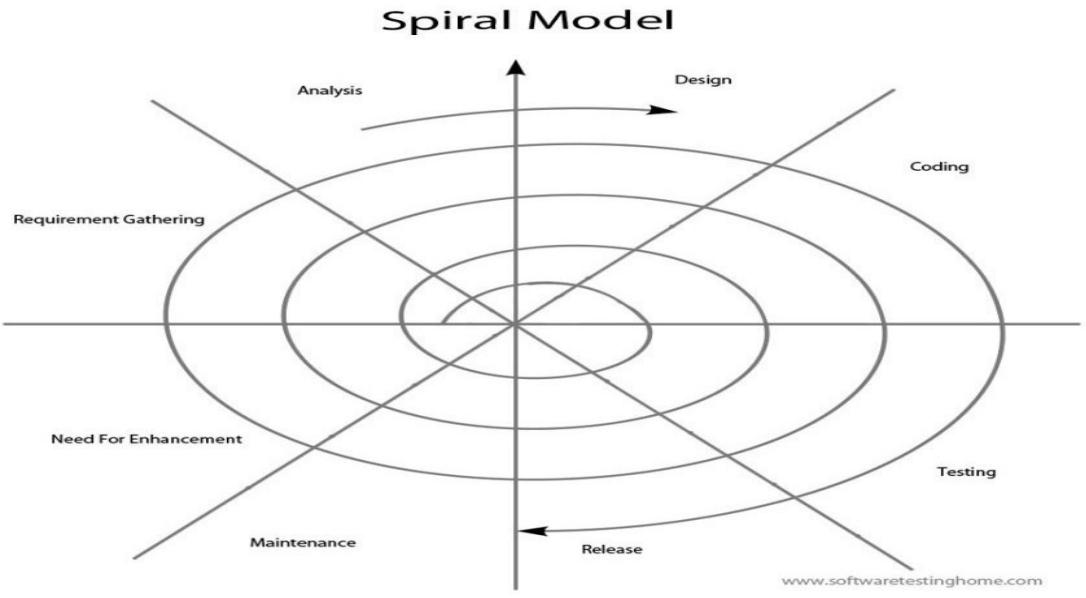



Figure 1 Fig 1: Spiral Model

Stages of SDLC:

Requirement Gathering and Analysis

Designing

Coding

Testing

Deployment

## Requirements Definition Stage and Analysis:

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.

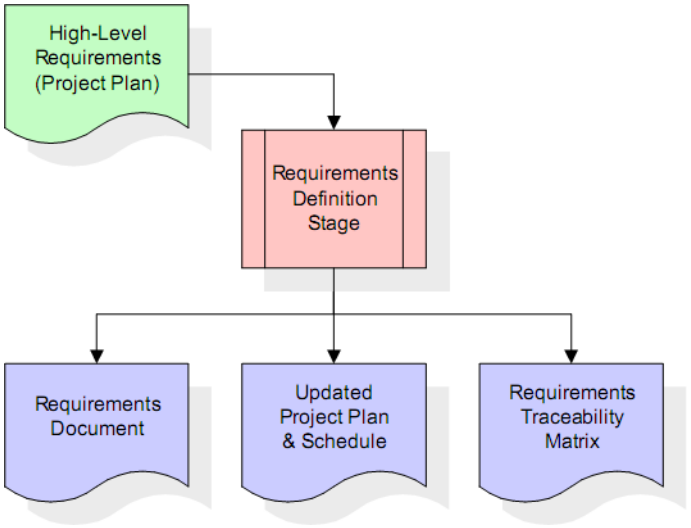



Figure 15

These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). the requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are not included in the requirements document. The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability. The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

## Design Stage:

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

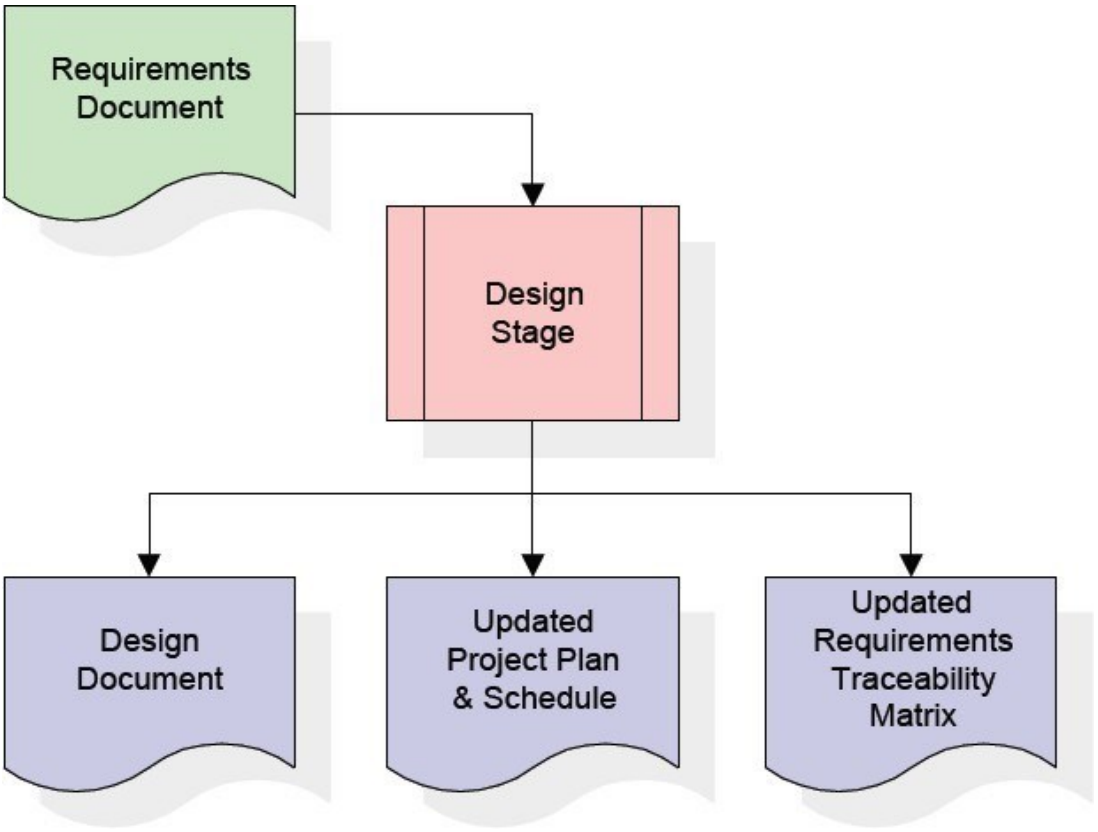



Figure 16

## Design Stage

When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

## Development Stage:

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.

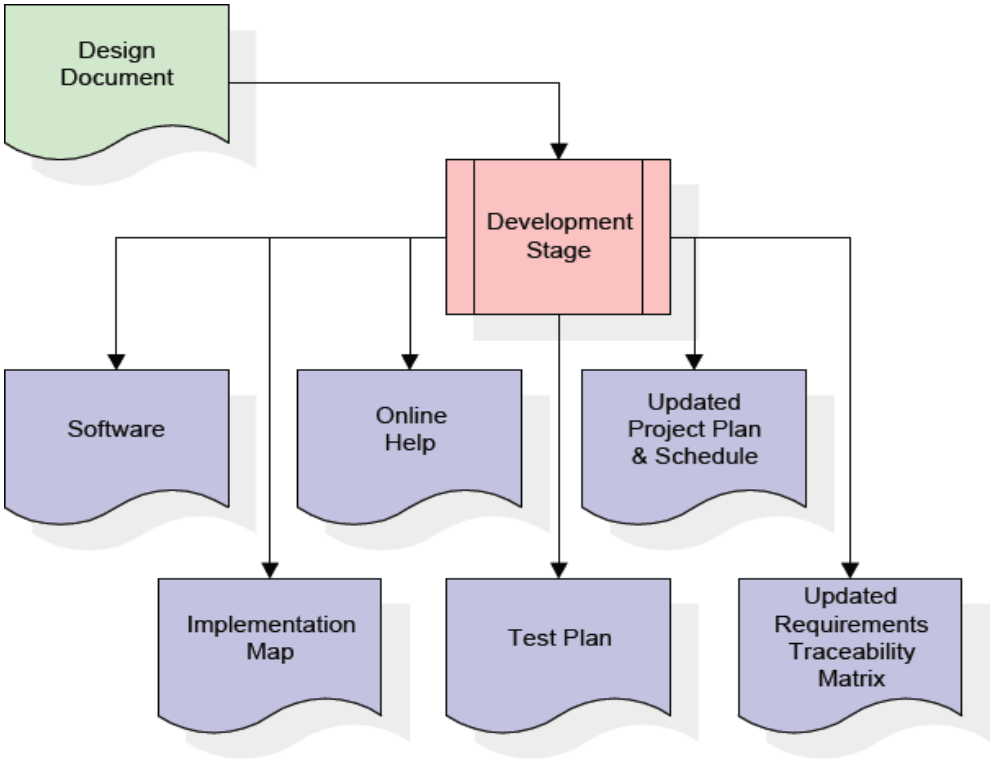



Figure 17

## Development Stage

The RTM will be updated to show that each developed artefact is linked to a specific design element, and that each developed artefact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

## Integration & Test Stage:

During the integration and test stage, the software artefacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability.

During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

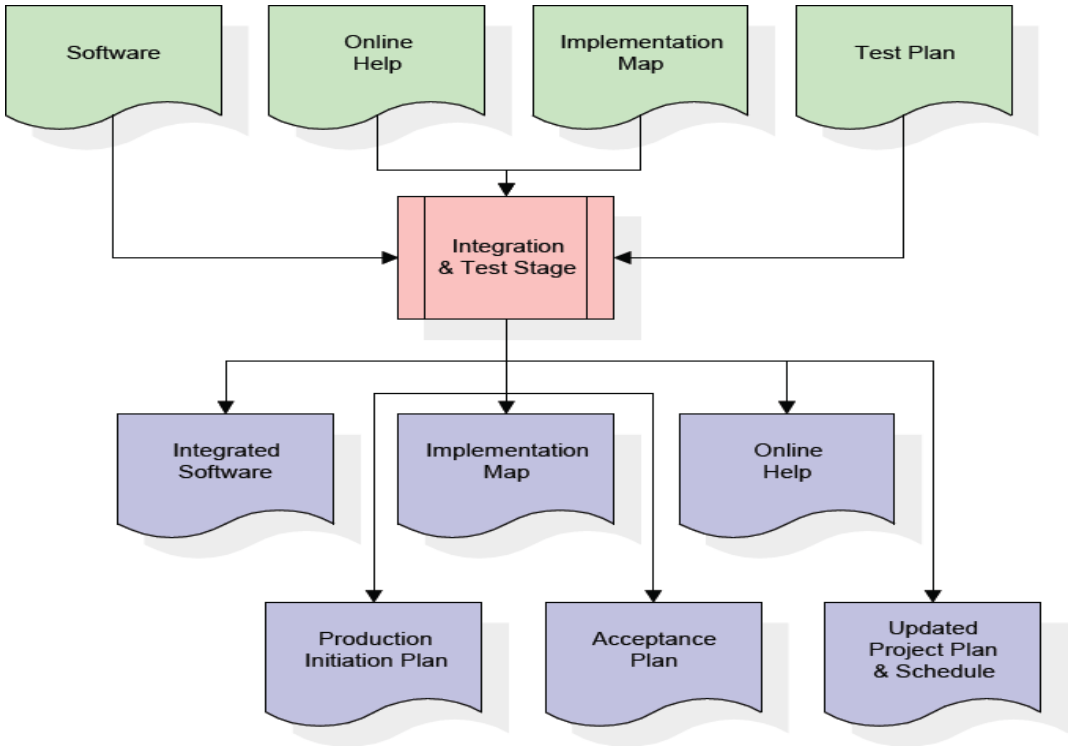



Figure 18

## Integration and Test stage

The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

## Installation & Acceptance Stage

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.

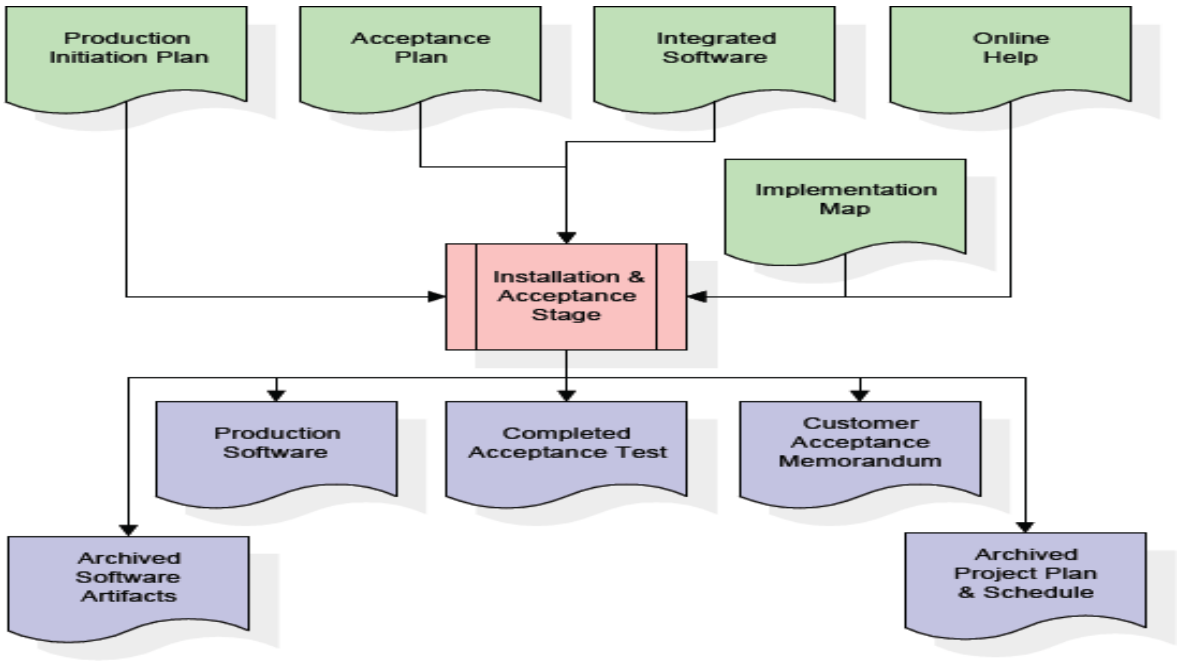



Figure 19

### Installation and Acceptance Stage

The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labour data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

## 4.4 SYSTEM ARCHITECTURE

**Architecture Flow:**

Below architecture diagram represents mainly flow of request from the users to database through servers. In this scenario overall system is designed in three tiers separately using three layers called presentation layer, business layer, data link layer. This project was developed using 3-tier architecture.

**3-Tier Architecture:**

The three-tier software architecture (three layer architecture) emerged in the 1990s to overcome the limitations of the two-tier architecture. The third tier (middle tier server) is between the user interface (client) and the data management (server) components. This middle tier provides process management where business logic and rules are executed and can accommodate hundreds of users (as compared to only 100 users with the two tier architecture) by providing functions such as queuing, application execution, and database staging.

The three tier architecture is used when an effective distributed client/server design is needed that provides (when compared to the two tier) increased performance, flexibility, maintainability, reusability, and scalability, while hiding the complexity of distributed processing from the user. These characteristics have made three layer architectures a popular choice for Internet applications and net-centric information systems..

**Advantages of Three-Tier:**

Separates functionality from presentation.

Clear separation - better understanding.

Changes limited to well define components.

# CHAPTER-5

# SYSTEM DESIGN

## 5.1 UML Diagrams

UML (Unified Modeling Language) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997. It was initially started to capture the behavior of complex software and non-software system and now it has become an OMG standard. This tutorial gives a complete understanding on UML.

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.

UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

OMG is continuously making efforts to create a truly industry standard. UML

stands for Unified Modeling Language.

UML is different from the other common programming languages such as C++, Java, COBOL, etc.

UML is a pictorial language used to make software blueprints.

UML can be described as a general-purpose visual modeling language to visualize, specify, construct, and document software system.

Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc.

UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object- oriented analysis and design. After some standardization, UML has become an OMG standard.

## Components of the UML

UML diagrams are the ultimate output of the entire discussion. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete.

UML includes the following nine diagrams, the details of which are described in the subsequent chapters.

Class diagram

Object diagram

Use case diagram

Sequence diagram

Collaboration diagram

Activity diagram

State chart diagram

Deployment diagram

Component diagram

The following are the main components of uml: -

Use-case Diagram

Class Diagram

Sequence Diagram

Activity Diagram

Collaboration Diagram

## 5.2 Use-Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

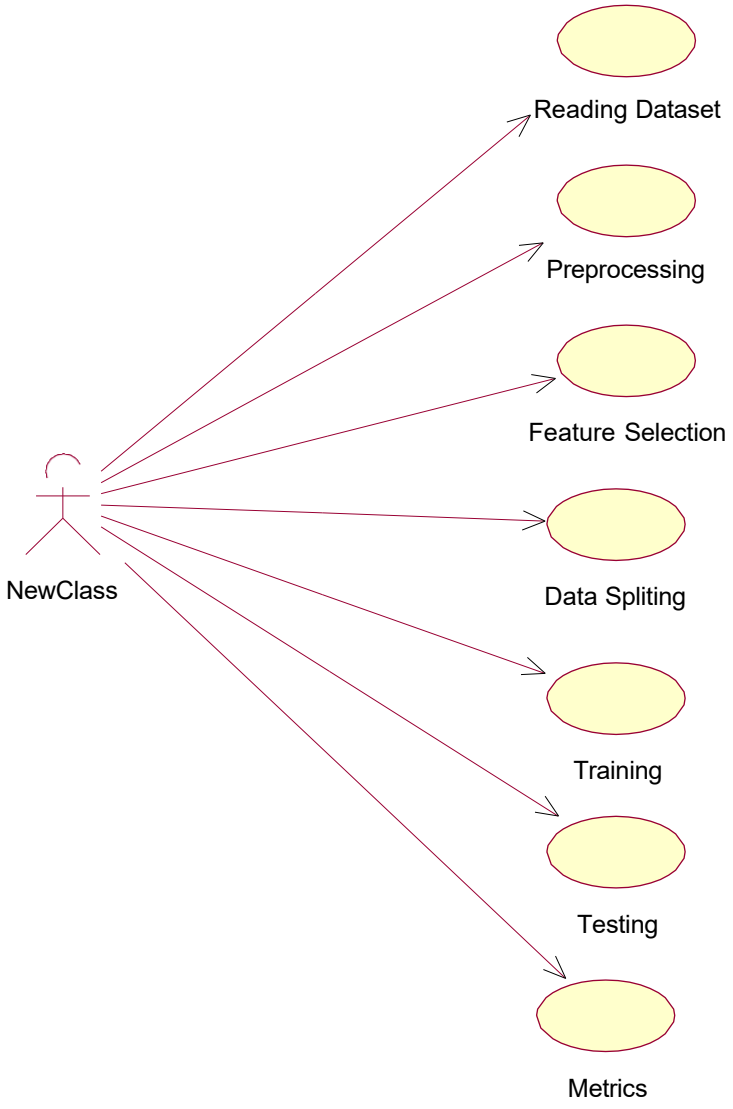


Figure 20

### 5.3 Class Diagram:

In software engineering, a class diagram in the Unified

Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

Class Diagram for Our System:

## Class Diagram

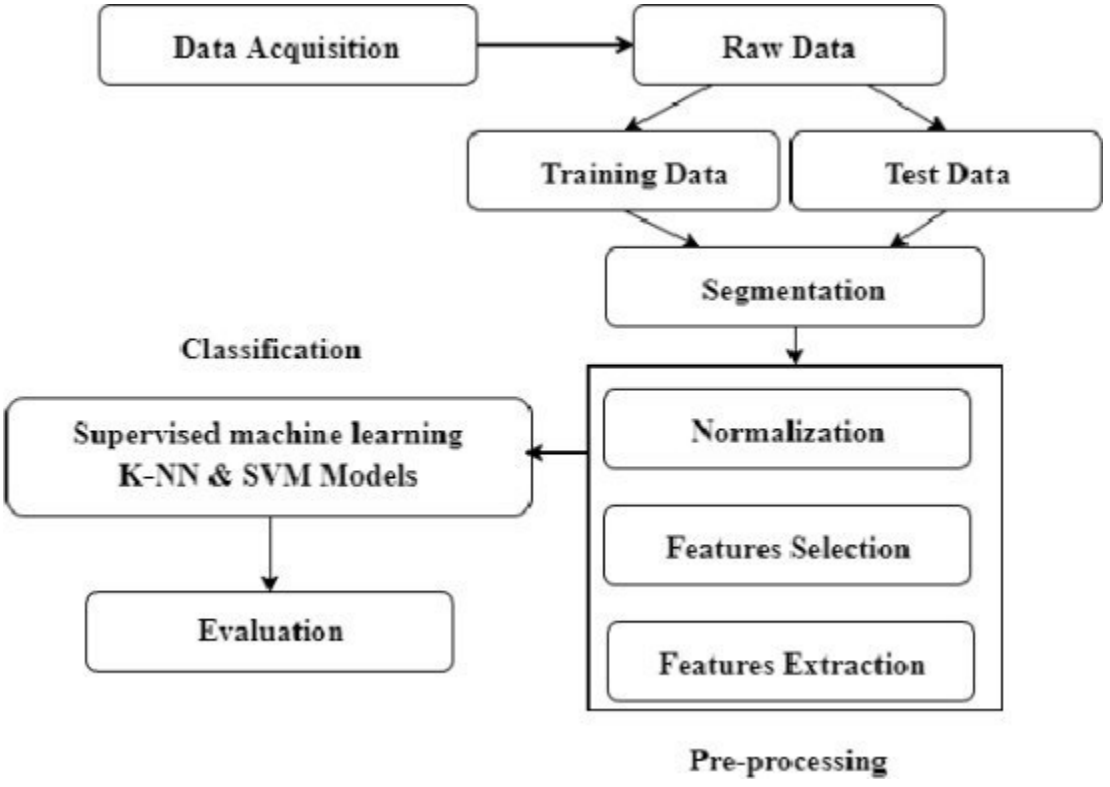



Figure 21

Class Diagram for Our System

## 5.4 Sequence Diagram:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

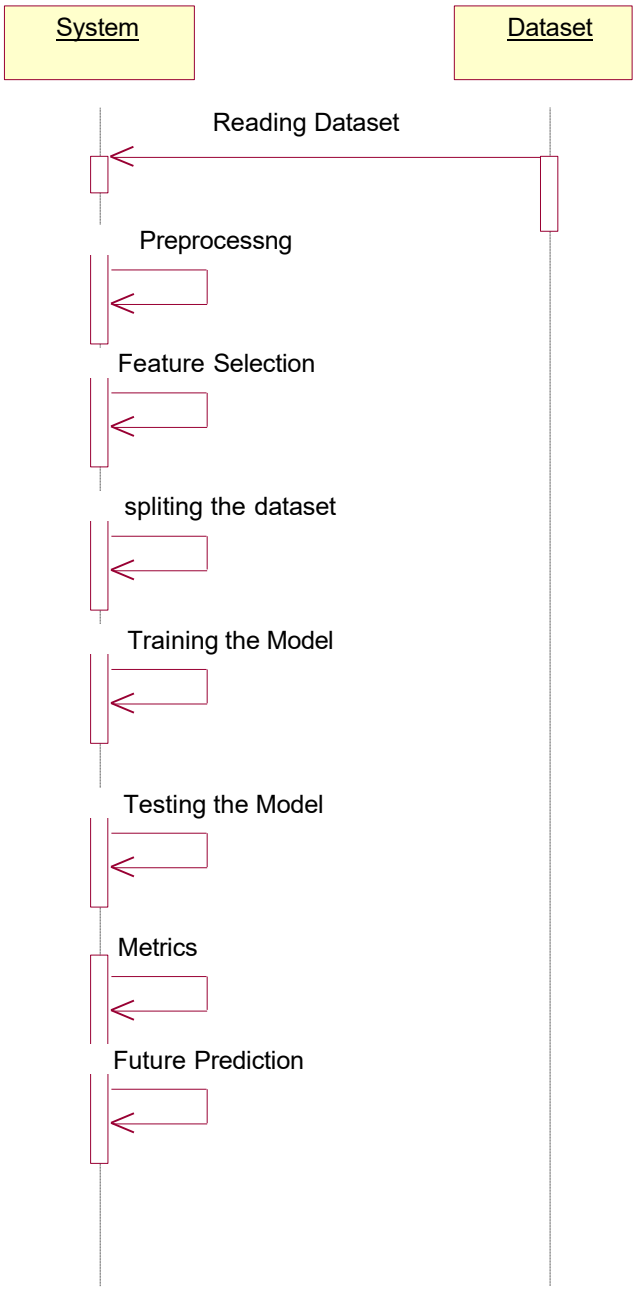



Figure 22: Sequence Diagram

Sequence Diagram for Our System

## 5.5 Collaboration Diagram:

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization where as the collaboration diagram shows the object organization.

## Collaboration Diagrams for Our System:

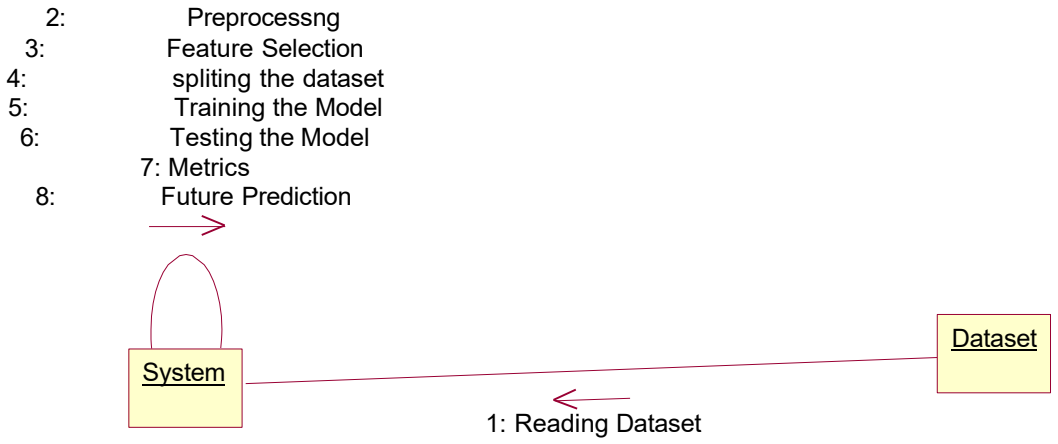
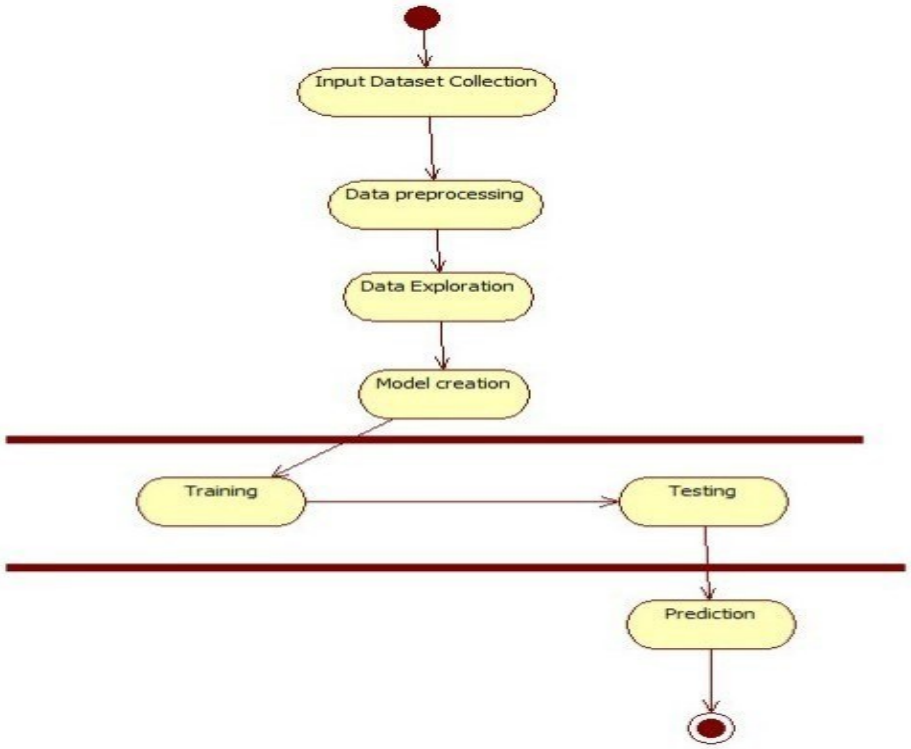
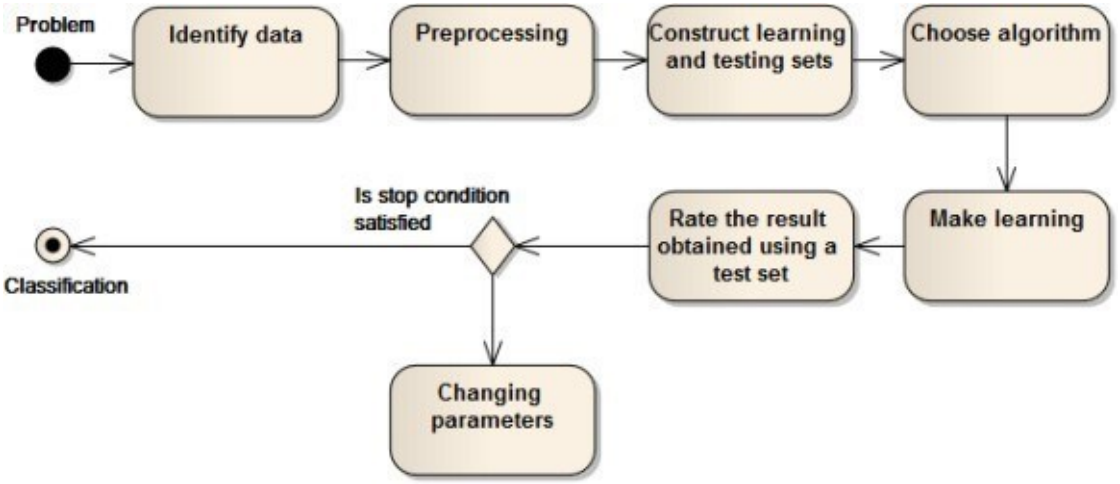



Figure 23

### 5.6 Activity diagram

Activity diagrams show the workflow from a start point to the finish point detailing the many decision paths that exist in the progression of events contained in the activity.



# CHAPTER-6

# IMPLEMENTATION

Here are some potential modules with descriptions for a project focusing on human activity analysis using machine learning algorithms:

**Data Collection Module:**

Description: This module is responsible for collecting raw sensor data from various sources such as smartphones, wearable devices, or IoT sensors. It may include components for data streaming, data preprocessing, and data storage.

**Data Preprocessing Module:**

Description: This module preprocesses the raw sensor data to make it suitable for input to the machine learning models. Preprocessing tasks may include filtering, noise removal, feature extraction, normalization, and segmentation of the data.

**Feature Extraction Module:**

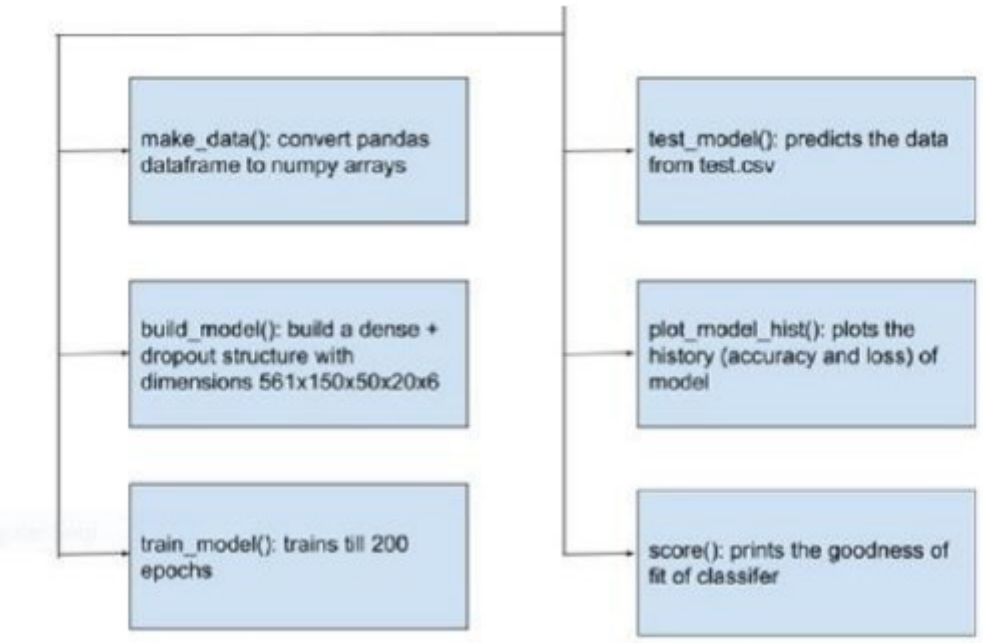
Description: Feature extraction is crucial for capturing relevant information from the raw sensor data. This module extracts meaningful features from the preprocessed data, such as statistical features (mean, standard deviation), frequency-domain features (FFT coefficients), time-domain features (temporal patterns), or domain-specific features (e.g., gait analysis features for motion data).

**Machine Learning Model Training Module:**

Description: This module trains machine learning models to recognize and classify human activities based on the extracted features. It involves selecting appropriate algorithms (e.g., SVM, Random Forest, Neural Networks) and optimizing their parameters using techniques such as cross-validation and hyperparameter tuning.

**Model Evaluation Module:**

Description: After training the machine learning models, this module evaluates their performance using evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrices. It may involve splitting the data into training and testing sets, performing cross-validation, and analyzing model performance on unseen data.




Random Forest works for the both classification and or regression tasks. Arbitrary forests forms a lot of choice trees. Each tree is created from a bootstrap test from the preparation information. Determine what number of choice trees will be incorporated into the woodland (Number of trees in the timberland), and what number of characteristics will be selfassertively drawn for thought at every hub. In the event that the last isn't indicated (choice Number of properties… left unchecked), this number is equivalent to the square base of the quantity of traits in the information. Unique Breiman's proposition is to develop the trees with no pre-pruning, however since pre-pruning regularly works great and is quicker, the client can set the profundity to which the trees will be developed. The kNNis one of the classification methods where no prior knowledge about the data distribution is required. Set the value for k which represents the number of nearest neighbors. The unknown datapoint will be classified as the class from the bunch of the labeled points among its nearest neighbor. Here the distance formula is used and it can be either Manhattan or Euclidean. KNN gadget utilizes the calculation that looks for k nearest preparing models in highlight space and uses their normal as forecast.kNN algorithm gives the accuracy of 96.6% in tracking human activity , when k is 5. The Stochastic Gradient Descent widget uses 35 stochastic gradient descent that minimizes chosen loss function with linear function. Decide the algorithm parameters for classification loss function, regression loss function and norms to prevent overfitting. Finally fix the learning parameters and result is obtained depends upon the learning rate. The initial learning rate is 0.0100 with constant and number of iterations is fixed as 1000, number of passes through

the training data. Naive Bayes is a probabilistic model and the predictions from the data in real time is made faster. After loading the dataset, the following functions is carried out the for the working of the model:

♣ make_data(): convert pandas dataframe to numpy arrays

♣ build_model(): build a dense + dropout structure with dimensions561x150x50x20x6

♣ train_model(): trains till 200 epochs

♣ test_model(): predicts the data from test.csv

♣ plot_model_hist(): plots the history (accuracy and loss) of model

♣ score(): prints the goodness of fit of classifier Logistic regression standard model was incorporated with a loss function and chosen the regularization type as either L1 or L2 and set the cost strength.

It gives the good result compared to Naïve Bayes in human activity analysis. It is one of the best performing models and provided improved accuracy compared to other machine learning techniques. It can handle any nonlinear effects and reduce the noise.

**Real-Time Monitoring Module:**

Description: This module enables real-time monitoring of human activities using the trained machine learning models. It processes incoming sensor data streams, applies the trained models for activity recognition, and provides feedback or alerts based on detected activities. It may also include components for visualizing activity data in real-time.

**User Interface Module:**

Description: The user interface module provides an interactive interface for users to interact with the activity analysis system. It may include graphical user interfaces (GUIs), web-based dashboards, or mobile applications that display activity summaries, trends, and insights. It allows users to visualize their activity data, set goals, and receive personalized recommendations.

**Privacy and Security Module:**

Description: This module addresses privacy and security concerns associated with collecting and analyzing human activity data. It implements privacy-preserving techniques

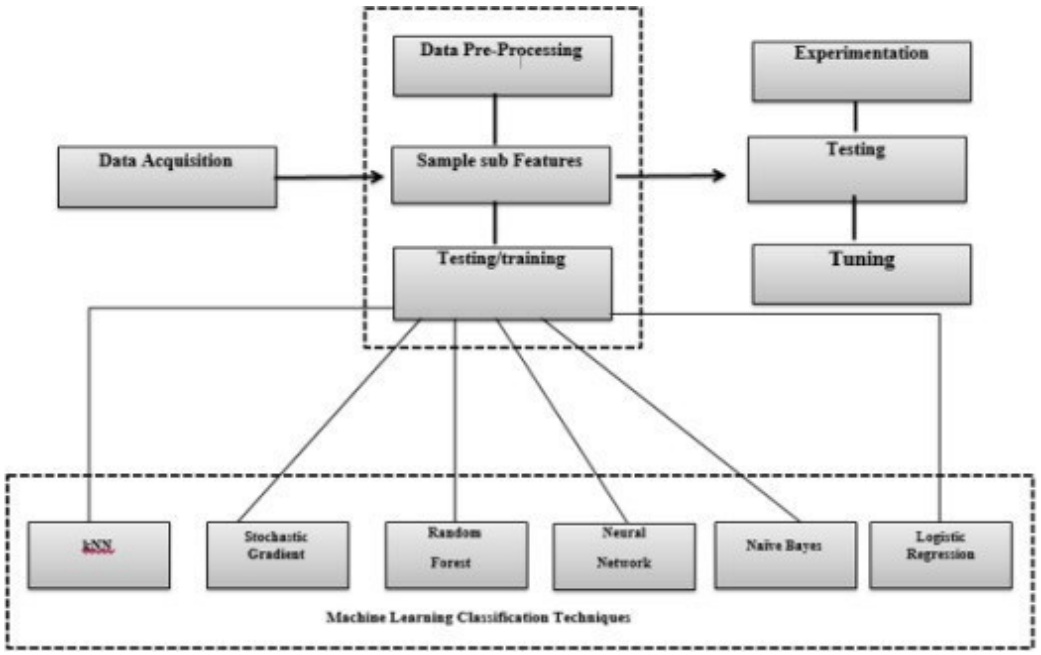
such as data encryption, anonymization, or differential privacy to protect users' sensitive information. It may also include components for user authentication, access control, and data encryption.

**Deployment and Integration Module:**

Description: This module focuses on deploying the activity analysis system in production environments and integrating it with existing systems or platforms. It may involve packaging the system components into deployable containers, configuring cloud infrastructure, and ensuring interoperability with other software systems or APIs.

**Performance Optimization Module:**

Description: This module optimizes the performance of the activity analysis system in terms of speed, scalability, and resource efficiency. It may involve optimizing algorithm implementations, parallelizing computations, and leveraging hardware acceleration (e.g., GPU computing) to enhance system performance.



### 6.1 Data Set Description:

Artificial Intelligence(AI) models are developed to recognize the activity of the human from the provided dataset UCI online storehouse. Necessary Packages:

1. Numpy
2. Pandas
3. Matplotlib
4. pyplot
5. Scikit-learn
6. Tensorflow
7. Jupyter Sample dataset for Human activity analysis

| tBodyAcc- | tBodyAcc- | tBodyAcc- | tBodyAcc- | tBodyAcc- | tBodyAcc- | tBodyAcc- | tBodyAcc- | tBodyAcc- | tBodyAcc- | tBodyAcc- | tBodyAcc- | tBodyAcc- | tBodyAcc- | tBodyAcc- | tBodyAcc- | tBodyAcc- | tBodyAcc- | tBodyAcc- | tBodyAcc- |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.288585 | -0.02029 | -0.13291 | -0.99528 | -0.98311 | -0.91353 | -0.99511 | -0.98318 | -0.92353 | -0.93472 | -0.56738 | -0.74441 | 0.852947 | 0.685845 | 0.814263 | -0.96552 | -0.99994 | -0.99986 | -0.99461 | -0.99423 |
| 0.278419 | -0.01641 | -0.12352 | -0.99825 | -0.9753 | -0.96032 | -0.99881 | -0.97491 | -0.95769 | -0.94307 | -0.55785 | -0.81841 | 0.849308 | 0.685845 | 0.822637 | -0.98193 | -0.99999 | -0.99979 | -0.99841 | -0.99915 |
| 0.279653 | -0.01947 | -0.11346 | -0.99538 | -0.96719 | -0.97894 | -0.99652 | -0.96367 | -0.97747 | -0.93869 | -0.55785 | -0.81841 | 0.843609 | 0.682401 | 0.839344 | -0.98348 | -0.99997 | -0.99966 | -0.99947 | -0.99713 |
| 0.279174 | -0.0262 | -0.12328 | -0.99609 | -0.9834 | -0.99068 | -0.9971 | -0.98275 | -0.9893 | -0.93869 | -0.57616 | -0.82971 | 0.843609 | 0.682401 | 0.837869 | -0.98609 | -0.99998 | -0.99974 | -0.9995 | -0.99718 |
| 0.276629 | -0.01657 | -0.11536 | -0.99814 | -0.98082 | -0.99048 | -0.99832 | -0.97967 | -0.99044 | -0.94247 | -0.56917 | -0.82471 | 0.849095 | 0.68325 | 0.837869 | -0.99265 | -0.99999 | -0.99986 | -0.99976 | -0.998 |
| 0.277199 | -0.0101 | -0.10514 | -0.99733 | -0.99049 | -0.99542 | -0.99763 | -0.99022 | -0.99555 | -0.94247 | -0.56568 | -0.82277 | 0.849095 | 0.695586 | 0.845922 | -0.99393 | -0.99999 | -0.99986 | -0.99992 | -0.99758 |
| 0.279454 | -0.01964 | -0.11002 | -0.99692 | -0.96719 | -0.98312 | -0.997 | -0.9661 | -0.98312 | -0.94099 | -0.56568 | -0.81719 | 0.85104 | 0.674347 | 0.833591 | -0.98684 | -0.99998 | -0.99966 | -0.99963 | -0.99686 |
| 0.277432 | -0.03049 | -0.12536 | -0.99656 | -0.96673 | -0.98159 | -0.99649 | -0.96631 | -0.98298 | -0.94099 | -0.57264 | -0.81719 | 0.850328 | 0.67041 | 0.832383 | -0.97685 | -0.99998 | -0.99934 | -0.99916 | -0.99577 |
| 0.277293 | -0.02175 | -0.12075 | -0.99733 | -0.96125 | -0.98367 | -0.9976 | -0.95724 | -0.98438 | -0.9406 | -0.56418 | -0.82353 | 0.850328 | 0.67041 | 0.832383 | -0.9833 | -0.99999 | -0.99953 | -0.99943 | -0.99735 |
| 0.280586 | -0.00996 | -0.10607 | -0.9948 | -0.97276 | -0.98624 | -0.9954 | -0.97366 | -0.98564 | -0.94003 | -0.55459 | -0.81585 | 0.845442 | 0.684757 | 0.838455 | -0.98654 | -0.99996 | -0.99966 | -0.99972 | -0.99616 |
| 0.27688 | -0.01272 | -0.10344 | -0.99482 | -0.97308 | -0.98536 | -0.99551 | -0.97395 | -0.98517 | -0.94003 | -0.55459 | -0.81585 | 0.845442 | 0.684757 | 0.838455 | -0.98788 | -0.99996 | -0.99972 | -0.99967 | -0.99587 |
| 0.276228 | -0.02144 | -0.1082 | -0.99825 | -0.98721 | -0.99273 | -0.99825 | -0.986 | -0.99318 | -0.94391 | -0.57142 | -0.82069 | 0.850532 | 0.686528 | 0.846013 | -0.99462 | -0.99999 | -0.99989 | -0.99988 | -0.99747 |
| 0.278457 | -0.02041 | -0.11273 | -0.99913 | -0.98468 | -0.99627 | -0.99908 | -0.98294 | -0.99641 | -0.94391 | -0.5697 | -0.82503 | 0.852069 | 0.686528 | 0.846013 | -0.99512 | -1 | -0.99988 | -0.99992 | -0.99838 |
| 0.277175 | -0.01471 | -0.10676 | -0.99919 | -0.99053 | -0.99337 | -0.99921 | -0.99069 | -0.99217 | -0.94332 | -0.56936 | -0.82251 | 0.852035 | 0.692647 | 0.84635 | -0.99656 | -1 | -0.99994 | -0.99989 | -0.99808 |
| 0.297946 | 0.027094 | -0.06167 | -0.98864 | -0.8167 | -0.90191 | -0.98896 | -0.79428 | -0.88801 | -0.92598 | -0.44847 | -0.73058 | 0.848666 | 0.68065 | 0.838205 | -0.90381 | -0.9998 | -0.9894 | -0.9912 | -0.98937 |
| 0.279203 | -0.02302 | -0.12208 | -0.99684 | -0.97485 | -0.98339 | -0.99709 | -0.97333 | -0.98407 | -0.94172 | -0.57094 | -0.81763 | 0.848666 | 0.68065 | 0.838205 | -0.9844 | -0.99998 | -0.99972 | -0.99937 | -0.99668 |
| 0.279038 | -0.0148 | -0.11685 | -0.99694 | -0.98187 | -0.98258 | -0.99722 | -0.98162 | -0.98134 | -0.94172 | -0.5635 | -0.82421 | 0.852573 | 0.688895 | 0.838398 | -0.9893 | -0.99998 | -0.99986 | -0.99951 | -0.99668 |
| 0.280135 | -0.01392 | -0.10637 | -0.99769 | -0.98752 | -0.99041 | -0.99801 | -0.98795 | -0.99219 | -0.94208 | -0.5635 | -0.81589 | 0.849851 | 0.693876 | 0.838398 | -0.99491 | -0.99999 | -0.9999 | -0.99982 | -0.99791 |
| 0.277731 | -0.01821 | -0.10919 | -0.99749 | -0.99322 | -0.99613 | -0.9979 | -0.99271 | -0.99649 | -0.94487 | -0.57509 | -0.81589 | 0.847154 | 0.692591 | 0.847226 | -0.99814 | -0.99999 | -0.99996 | -0.99994 | -0.99748 |
| 0.275568 | -0.01698 | -0.11143 | -0.99781 | -0.99052 | -0.99762 | -0.99821 | -0.98947 | -0.99719 | -0.94566 | -0.57334 | -0.82658 | 0.847154 | 0.692591 | 0.847509 | -0.99742 | -0.99999 | -0.99995 | -0.99996 | -0.99843 |

### 6.2 SAMPLE CODE:

```
# import necessary libraries
import pandas as pd
import random
import numpy as np
from sklearn.linear_model import LogisticRegression, LinearRegression
from sklearn.ensemble import RandomForestClassifier,
RandomForestRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing  import  LabelEncoder
from sklearn.model_selection import cross_val_score
from sklearn.metrics  import  f1_score
random.seed(0)
# Now you can fetch the data
hapt_train = pd.read_csv('train.csv')
```

```
hapt_test = pd.read_csv('test.csv')


hapt_train
```

| | tBodyAcc-mean()-X | tBodyAcc-mean()-Y | tBodyAcc-mean()-Z | tBodyAcc-std()-X | tBodyAcc-std()-Y | tBodyAcc-std()-Z | tBodyAcc-mad()-X | tBodyAcc-mad()-Y | tBodyAcc-mad()-Z | tBodyAcc-max()-X | ... | fBodyBodyGyroJerkMag-kurtosis() | angle(tBodyAccMean,gravity) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.288585 | -0.020294 | -0.132905 | -0.995279 | -0.983111 | -0.913526 | -0.995112 | -0.983185 | -0.923527 | -0.934724 | ... | -0.710304 | -0.112754 |
| 1 | 0.278419 | -0.016411 | -0.123520 | -0.998245 | -0.975300 | -0.960322 | -0.998807 | -0.974914 | -0.957686 | -0.943068 | ... | -0.861499 | 0.053477 |
| 2 | 0.279653 | -0.019467 | -0.113462 | -0.995380 | -0.967187 | -0.978944 | -0.996520 | -0.963668 | -0.977469 | -0.938692 | ... | -0.760104 | -0.118559 |
| 3 | 0.279174 | -0.026201 | -0.123283 | -0.996091 | -0.983403 | -0.990675 | -0.997099 | -0.982750 | -0.989302 | -0.938692 | ... | -0.482845 | -0.036788 |
| 4 | 0.276629 | -0.016570 | -0.115362 | -0.998139 | -0.980817 | -0.990482 | -0.998321 | -0.979672 | -0.990441 | -0.942469 | ... | -0.699205 | 0.123320 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 7347 | 0.299665 | -0.057193 | -0.181233 | -0.195387 | 0.039905 | 0.077078 | -0.282301 | 0.043616 | 0.060410 | 0.210795 | ... | -0.880324 | -0.190437 |
| 7348 | 0.273853 | -0.007749 | -0.147468 | -0.235309 | 0.004816 | 0.059280 | -0.322552 | -0.029456 | 0.080585 | 0.117440 | ... | -0.680744 | 0.064907 |
| 7349 | 0.273387 | -0.017011 | -0.045022 | -0.218218 | -0.103822 | 0.274533 | -0.304515 | -0.098913 | 0.332584 | 0.043999 | ... | -0.304029 | 0.052806 |
| 7350 | 0.289654 | -0.018843 | -0.158281 | -0.219139 | -0.111412 | 0.268893 | -0.310487 | -0.068200 | 0.319473 | 0.101702 | ... | -0.344314 | -0.101360 |
| 7351 | 0.351503 | -0.012423 | -0.203867 | -0.269270 | -0.087212 | 0.177404 | -0.377404 | -0.038678 | 0.229430 | 0.269013 | ... | -0.740738 | -0.280088 |

7352 rows × 563 columns

```
hapt_train.columns
```

Index(['tBodyAcc-mean()-X', 'tBodyAcc-mean()-Y', 'tBodyAcc-mean()-Z',
'tBodyAcc-std()-X', 'tBodyAcc-std()-Y', 'tBodyAcc-std()-Z',
'tBodyAcc-mad()-X', 'tBodyAcc-mad()-Y', 'tBodyAcc-mad()-Z',
'tBodyAcc-max()-X',
...
'fBodyBodyGyroJerkMag-kurtosis()', 'angle(tBodyAccMean,gravity)',
'angle(tBodyAccJerkMean),gravityMean)',
'angle(tBodyGyroMean,gravityMean)',
'angle(tBodyGyroJerkMean,gravityMean)', 'angle(X,gravityMean)',
'angle(Y,gravityMean)', 'angle(Z,gravityMean)', 'subject', 'Activity'],
dtype='object', length=563)

```
Xtrain, Xtest, ytrain, ytest = hapt_train.drop('Activity', axis=1),
hapt_test.drop('Activity', axis=1), hapt_train['Activity'],
hapt_test['Activity']
```

```
label_encoder = LabelEncoder()
ytrain = label_encoder.fit_transform(ytrain)
ytest = label_encoder.transform(ytest)
```

```
Xtrain.shape, Xtest.shape, ytrain.shape, ytest.shape
```

((7352, 562), (2947, 562), (7352,), (2947,))

# Implement Genetic Algoíithm Functions

```
population_size = 20     # number of chromosomes in the population
c = Xtrain.shape[1]      # length of the chromosome
selected_features = 20  # number of selected features
mutation_rate = 0.05     # probability of mutation
num generation = 11      # Set the number of Generation
```

## Initial Population

```
def
init_population(population_size=population_size,c=c,top_number=selected
_features):
    population = []
    features = list(range(c))
    for i in range(population_size):
        # individual = []
        # j = 0
        # while(j<top_number):
        #     p = random.uniform(0,1)
        #     feature_idx = random.sample(features,1)[0]
        #     if(p>=0.5 and feature_idx not in individual):
        #         individual.append(feature_idx)
        #         j+=1
        # population.append(individual)


        population.append(np.random.permutation(features).tolist()[0:to
p_number])


    print('population is ')
    print(population)
    print('----------------------------------------')
    print('Each chromosome is ', population)
    print('----------------------------------------')
    return population
```

```
init_pop = init_population()
```

```
population is
[[435, 348, 247, 120, 227, 405, 104, 23, 391, 68, 273, 301, 289, 257,
477, 460, 162, 220, 238, 0], [184, 396, 548, 15, 12, 328, 265, 258,
107, 157, 358, 164, 343, 58, 99, 84, 127, 422, 207, 304], [100, 555,
244, 334, 240, 281, 552, 521, 189, 519, 63, 437, 265, 390, 476, 424,
307, 416, 89, 81], [375, 476, 442, 250, 14, 302, 358, 408, 175, 177,
466, 312, 163, 387, 272, 313, 348, 176, 528, 87], [205, 109, 483, 435,
364, 470, 520, 288, 332, 193, 118, 313, 505, 151, 254, 270, 211, 98,
139, 265], [320, 525, 536, 257, 486, 80, 516, 471, 275, 44, 49, 161,
198, 102, 148, 351, 433, 290, 76, 83], [96, 241, 97, 351, 286, 497,
238, 396, 374, 225, 73, 462, 186, 382, 78, 235, 172, 504, 146, 350],
[533, 358, 395, 408, 455, 205, 400, 141, 144, 435, 260, 235, 367, 255,
414, 498, 553, 232, 325, 381], [296, 252, 157, 466, 23, 222, 116, 57,
95, 426, 304, 293, 554, 338, 259, 532, 528, 280, 59, 283], [299, 463,
148, 450, 538, 98, 473, 100, 6, 248, 424, 365, 336, 280, 234, 153, 553,
291, 201, 310], [386, 72, 551, 247, 89, 130, 59, 506, 215, 395, 447,
96, 204, 380, 16, 544, 17, 454, 383, 155], [502, 155, 164, 80, 196, 4,
```

```
251, 559, 292, 294, 424, 17, 326, 313, 350, 39, 186, 460, 314, 261],
[378, 11, 501, 320, 137, 93, 274, 294, 342, 513, 470, 225, 370, 97,
154, 170, 280, 164, 241, 25], [547, 270, 360, 464, 227, 480, 412, 13,
543, 374, 141, 210, 204, 517, 202, 322, 434, 157, 281, 528], [230, 58,
501, 421, 9, 56, 220, 165, 534, 232, 285, 176, 115, 561, 387, 274, 366,
370, 395, 290], [487, 268, 100, 515, 91, 138, 430, 374, 273, 463, 551,
93, 80, 192, 85, 328, 390, 126, 164, 276], [148, 342, 347, 76, 12, 211,
454, 435, 75, 43, 248, 44, 343, 551, 120, 496, 264, 538, 382, 204],
[321, 516, 523, 456, 313, 163, 394, 310, 500, 130, 432, 269, 195, 150,
488, 57, 258, 139, 119, 28], [346, 488, 178, 460, 55, 259, 515, 279,
529, 72, 379, 422, 412, 487, 554, 333, 255, 434, 400, 418], [108, 506,
542, 177, 252, 66, 32, 399, 229, 367, 7, 491, 199, 86, 366, 464, 387,
469, 12, 418]]
-----------------------------------------------
Each chromosome is  [[435, 348, 247, 120, 227, 405, 104, 23, 391, 68,
273, 301, 289, 257, 477, 460, 162, 220, 238, 0], [184, 396, 548, 15,
12, 328, 265, 258, 107, 157, 358, 164, 343, 58, 99, 84, 127, 422, 207,
304], [100, 555, 244, 334, 240, 281, 552, 521, 189, 519, 63, 437, 265,
390, 476, 424, 307, 416, 89, 81], [375, 476, 442, 250, 14, 302, 358,
408, 175, 177, 466, 312, 163, 387, 272, 313, 348, 176, 528, 87], [205,
109, 483, 435, 364, 470, 520, 288, 332, 193, 118, 313, 505, 151, 254,
270, 211, 98, 139, 265], [320, 525, 536, 257, 486, 80, 516, 471, 275,
44, 49, 161, 198, 102, 148, 351, 433, 290, 76, 83], [96, 241, 97, 351,
286, 497, 238, 396, 374, 225, 73, 462, 186, 382, 78, 235, 172, 504,
146, 350], [533, 358, 395, 408, 455, 205, 400, 141, 144, 435, 260, 235,
367, 255, 414, 498, 553, 232, 325, 381], [296, 252, 157, 466, 23, 222,
116, 57, 95, 426, 304, 293, 554, 338, 259, 532, 528, 280, 59, 283],
[299, 463, 148, 450, 538, 98, 473, 100, 6, 248, 424, 365, 336, 280,
234, 153, 553, 291, 201, 310], [386, 72, 551, 247, 89, 130, 59, 506,
215, 395, 447, 96, 204, 380, 16, 544, 17, 454, 383, 155], [502, 155,
164, 80, 196, 4, 251, 559, 292, 294, 424, 17, 326, 313, 350, 39, 186,
460, 314, 261], [378, 11, 501, 320, 137, 93, 274, 294, 342, 513, 470,
225, 370, 97, 154, 170, 280, 164, 241, 25], [547, 270, 360, 464, 227,
480, 412, 13, 543, 374, 141, 210, 204, 517, 202, 322, 434, 157, 281,
528], [230, 58, 501, 421, 9, 56, 220, 165, 534, 232, 285, 176, 115,
561, 387, 274, 366, 370, 395, 290], [487, 268, 100, 515, 91, 138, 430,
374, 273, 463, 551, 93, 80, 192, 85, 328, 390, 126, 164, 276], [148,
342, 347, 76, 12, 211, 454, 435, 75, 43, 248, 44, 343, 551, 120, 496,
264, 538, 382, 204], [321, 516, 523, 456, 313, 163, 394, 310, 500, 130,
432, 269, 195, 150, 488, 57, 258, 139, 119, 28], [346, 488, 178, 460,
55, 259, 515, 279, 529, 72, 379, 422, 412, 487, 554, 333, 255, 434,
400, 418], [108, 506, 542, 177, 252, 66, 32, 399, 229, 367, 7, 491,
199, 86, 366, 464, 387, 469, 12, 418]]
```

## Fitness Function

```
# ------------------------------------------------------------------------ Classification ------------------------------------------
----------------------------

def fitness_fun(features, target=ytrain, cv=10, scoring='f1_macro',
n_jobs=-1):
```

```
    sc = StandardScaler()
    features = sc.fit_transform(features)

    # Dimensionality reduction using PCA
    # PCA(n_components=20).fit_transform(features)

    classifier = LogisticRegression(max_iter=10000)
#RandomForestClassifier()
    scores = cross_val_score( classifier, features, target,
scoring=scoring, n_jobs=n_jobs, cv=cv)
    return scores.mean()
```

```
# Val accuracy of the model on the training data
fitness_fun(Xtrain, ytrain)
```

0.9414125133548931

```
def get_fitness(population,data):
    fitness_values = []
    for individual in population:
      df = data

      # First Way
      # i=0
      # for column in data:
      #   if(individual[i]==0):
      #     df = df.drop(column,axis=1)
      #   i=i+1

      # ----------------------------------------------------------------------------------------------------------------------------------
      # Second Way
      # df = df[df.columns[[False if item == 0 else True for item in
individual]]]
      df = df.iloc[:,individual]
      features = df
      individual_fitness = fitness_fun(features)
      fitness_values.append(individual_fitness)

    return fitness_values
```

```
fitnes_val = get_fitness(population=init_pop,data=Xtrain)
fitnes_val
```

[0.675775199322714,
 0.7396056328740525,

0.683789027801135,
0.6516049876224352,
0.702845338495355,
0.8089182150379093,
0.7341285959477328,
0.6116419533023432,
0.815772849493035,
0.7178573279926911,
0.7060612092736054,
0.8566655742034699,
0.6574016418492918,
0.6619484109605434,
0.7761704945087998,
0.7425865128535556,
0.6823620527505041,
0.7906839521552638,
0.6298121432179926,
0.6690457956554694]

## Paíent Selection

```
def roulette_wheel_select(population, fitness_values,
num_parents=population_size):
  parents = []
  total = sum(fitness_values)
  norm_fitness_values = [x / total for x in fitness_values]
  print('norm_fitness_values is ', norm_fitness_values)
  cumulative_fitness = [sum(norm_fitness_values[:i+1]) for i in
range(len(norm_fitness_values))]
  print('cumulative_fitness is ', cumulative_fitness)


  for _ in range(num_parents):
    random_number = random.uniform(0, 1)
    individual_number = next(i for i, score in
enumerate(cumulative_fitness) if random_number <= score)
    parents.append(population[individual_number])

  return parents




def tournament_select(population, fitness_values, tournament_size = 3,
num_parents=population_size):
    selected = []
    for _ in range(num_parents):
        participants = random.sample(range(len(population)),
tournament_size)
        winner = max(participants, key=lambda x: fitness_values[x])
        selected.append(population[winner])
```

```
    return selected
```

```
 #Predict the target variable for the test data
def linear_rank_selection(population, fitness_values,
num_parents=population_size):
    sorted_indices = np.argsort(fitness_values)
    ranks = np.arange(1, len(population) + 1)
    selection_probabilities = ranks / np.sum(ranks)
    selected_indices = np.random.choice(sorted_indices,
size=num_parents, replace=False, p=selection_probabilities)
    selected_parents = [population[i] for i in selected_indices]
    return selected parents
```

```
high_individual =
roulette_wheel_select(population=init_pop,fitness_values=fitnes_val)
get_fitness(population=high_individual,data=Xtrain)
```

norm_fitness_values is [0.0472085540819701, 0.05166764414239853, 0.04776838707895835, 0.045520062483922596, 0.04909962988708824, 0.05650970817038135, 0.05128502726891124, 0.042728310035962815, 0.05698856176209536, 0.05014834300762483, 0.049324285378600535, 0.059845260865564215, 0.04592500730113001, 0.046242637181157974, 0.05422200578472242, 0.05187588355968593, 0.04766870092752982, 0.055235892282637956, 0.0439976498906546, 0.04673844890900293]
cumulative_fitness is [0.0472085540819701, 0.09887619822436863, 0.14664458530332697, 0.19216464778724957, 0.2412642776743378, 0.29777398584471915, 0.3490590131136304, 0.39178732314959325, 0.4487758849116886, 0.49892422791931346, 0.548248513297914, 0.6080937741634782, 0.6540187814646082, 0.7002614186457662, 0.7544834244304887, 0.8063593079901746, 0.8540280089177045, 0.9092639012003424, 0.953261551090997, 0.9999999999999999]
[0.6823620527505041,
 0.7425865128535556,
 0.815772849493035,
 0.8089182150379093,
 0.7060612092736054,
 0.815772849493035,
 0.7425865128535556,
 0.7341285959477328,
 0.7178573279926911,
 0.8566655742034699,
 0.7906839521552638,
 0.7060612092736054,
 0.8089182150379093,
 0.7425865128535556,
 0.6574016418492918,
 0.8089182150379093,
 0.6298121432179926,
 0.6690457956554694,
 0.6823620527505041,
 0.7906839521552638]

```
high_individual =
tournament_select(population=init_pop,fitness_values=fitnes_val)
get fitness(population=high individual,data=Xtrain)
```

[0.7178573279926911,
0.7906839521552638,
0.8566655742034699,
0.7906839521552638,
0.7761704945087998,
0.815772849493035,
0.683789027801135,
0.7425865128535556,
0.7060612092736054,
0.7341285959477328,
0.7906839521552638,
0.7761704945087998,
0.7425865128535556,
0.7906839521552638,
0.7906839521552638,
0.7906839521552638,
0.7906839521552638,
0.7761704945087998,
0.7060612092736054,
0.8089182150379093]

```
high_individual =
linear_rank_selection(population=init_pop,fitness_values=fitnes_val,num
_parents=population_size)
get_fitness(population=high_individual,data=Xtrain)
```

[0.675775199322714,
0.7761704945087998,
0.8566655742034699,
0.8089182150379093,
0.7906839521552638,
0.7396056328740525,
0.6116419533023432,
0.6516049876224352,
0.7425865128535556,
0.702845338495355,
0.7341285959477328,
0.815772849493035,
0.7060612092736054,
0.6298121432179926,
0.6574016418492918,
0.6823620527505041,
0.6619484109605434,
0.7178573279926911,
0.683789027801135,

0.6690457956554694]

Cíossoveí

```
# we have used uniform crossover, other option one point and multiple
point crossovers
def uniform_crossover(p1, p2):
    c1, c2 = [], []
    for i in range(len(p1)):
        if random.random() < 0.5:
            c1.append(p1[i]); c2.append(p2[i])
        else:
            c1.append(p2[i]); c2.append(p1[i])
    return np.array(c1), np.array(c2)
```

```
uniform_crossover([1,0,1,0,1,0,1,0,1,0],[0,1,0,1,0,1,0,1,0,1])
```
(array([1, 0, 0, 0, 1, 1, 0, 0, 1, 1]), array([0, 1, 1, 1, 0, 0, 1, 1, 0, 0]))

```
def single_point_crossover(parent1, parent2):
    # Generate a random crossover point
    crossover_point = random.randint(0, len(parent1) - 1)
    print(crossover_point)
    # Perform crossover
    child1 = parent1[:crossover_point] + parent2[crossover_point:]
    child2 = parent2[:crossover_point] + parent1[crossover_point:]

    return child1, child2
```

```
single point crossover([1,0,1,0,1,0,1,0,1,0],[0,1,0,1,1,0,1,0,0,1])
```
8
([1, 0, 1, 0, 1, 0, 1, 0, 0, 1], [0, 1, 0, 1, 1, 0, 1, 0, 1, 0])

```
def multipoint_random_crossover(parent1, parent2, num_points=2):
    # Get the length of the parents
    length = len(parent1)

    # Generate random crossover points
    crossover_points = sorted(random.sample(range(1, length),
num_points))
    #  print(crossover_points)
    # Initialize the offspring
    offspring1 = parent1.copy()
    offspring2 = parent2.copy()

    # Perform crossover at the selected points
```

```
    for i in range(crossover_points[0], crossover_points[1]):
        offspring1[i], offspring2[i] = offspring2[i], offspring1[i]

    return offspring1, offspring2
```

```
multipoint_random_crossover([1,0,1,0,1,0,1,0,1,0],[0,1,0,1,1,0,1,0,0,1]
)
```

([1, 0, 1, 0, 1, 0, 1, 0, 1, 0], [0, 1, 0, 1, 1, 0, 1, 0, 0, 1])

```
def get_crossover(parents, crossover_fun):
    offspring = []
    for i in range(0, len(parents), 2):
        parent1 = parents[i]
        parent2 = parents[i+1]
        offspring.extend(crossover_fun(parent1, parent2))
    return offspring
```

```
offspring = get_crossover(parents=high_individual,
crossover_fun=multipoint_random_crossover)
get fitness(population=offspring,data=Xtrain)
```

[0.6551743699490548,
0.7955039960270771,
0.8299805593712367,
0.8200825723429601,
0.7568590546733999,
0.7571368690849902,
0.6162662089079728,
0.6509641901101404,
0.7191186913104842,
0.7046137995652388,
0.8404518477650103,
0.6941353209209277,
0.6958202976977164,
0.6704958357364046,
0.6330241426257986,
0.6811150942711113,
0.7250546551983676,
0.6793014479750478,
0.7175979545268206,
0.6723892283663508]

**Mutation**

```
def mutation(offspring, mutation_rate=mutation_rate):
    for i in range(len(offspring)):
        for j in range(len(offspring[i])):
            if random.uniform(0,1) < mutation_rate:
                offspring[i][j] = max(offspring[i]) - offspring[i][j]
```

```
    return offspring
```

```
print(init_pop[0], '\n', init_pop[1])
x = mutation([init_pop[0],init_pop[1]])
print("="*150)
print(x[1], '\n', x[0])
```

```
[435, 348, 247, 120, 227, 405, 104, 23, 391, 68, 273, 301, 289, 257,
477, 460, 162, 220, 238, 0]
 [184, 396, 548, 15, 12, 328, 265, 258, 107, 157, 358, 164, 343, 58,
99, 84, 127, 422, 207, 304]
==========================================================================
==========================================================================
========
[184, 396, 548, 15, 12, 328, 265, 258, 107, 157, 358, 164, 343, 58, 99,
84, 127, 422, 207, 304]
 [435, 348, 247, 120, 227, 405, 104, 23, 391, 68, 273, 301, 289, 257,
477, 460, 162, 220, 238, 0]
```

### Genetic Algorithm

```
# Initialize the population
population = init_population(population_size, c)

best_fitness_over_all_iterations = [0]


# Iterate through the generations
for iteration in range(num_generation):
    # Evaluate the fitness of each chromosome in the population
    fitness_values = get_fitness(population, Xtrain)

    # Select parents for reproduction
    high_individual = linear_rank_selection(population, fitness_values)

    # Perform crossover to create offspring
    offspring = get_crossover(parents=high_individual,
crossover_fun=uniform_crossover)


    # Perform mutation on the offspring
    mutated_offspring = mutation(offspring, mutation_rate)

    # Replace the population with the offspring
    population = mutated_offspring

    # Print the best fitness value in each iteration
    best_fitness = max(fitness_values)
    if iteration % 100 == 0:
```

```
        print(f"Generation {iteration+1} : Best Fitness =
{best_fitness_over_all_iterations[-1]}")
    if best_fitness > best_fitness_over_all_iterations[-1]:
        best_fitness_over_all_iterations.append(best_fitness)
        print(f"Iteration {iteration+1}: Best Fitness =
{best_fitness}")

# Get the best chromosome from the final population
best_chromosome = population[np.argmax(fitness_values)]

# Print the selected features from the best chromosome
selected_features = best_chromosome
print("Selected Features:", selected_features)
```

```
population is
[[248, 173, 495, 520, 234, 377, 76, 353, 11, 245, 189, 180, 529, 387,
124, 512, 140, 398, 467, 47], [179, 415, 316, 266, 98, 339, 460, 244,
215, 330, 538, 183, 137, 510, 25, 81, 218, 20, 469, 243], [408, 512,
233, 270, 524, 292, 485, 278, 159, 141, 475, 468, 52, 537, 46, 115,
202, 252, 196, 495], [353, 128, 18, 325, 155, 12, 249, 413, 480, 383,
377, 114, 136, 505, 246, 15, 123, 549, 58, 478], [51, 548, 121, 166,
246, 432, 329, 487, 298, 408, 460, 118, 466, 349, 520, 468, 452, 538,
140, 258], [40, 14, 132, 77, 360, 434, 541, 346, 477, 520, 34, 141,
449, 103, 517, 31, 83, 536, 448, 392], [173, 490, 323, 400, 545, 452,
55, 156, 453, 132, 302, 94, 362, 188, 93, 352, 451, 191, 68, 320],
[465, 366, 497, 268, 349, 324, 110, 69, 538, 120, 222, 203, 138, 370,
198, 252, 337, 509, 471, 77], [138, 193, 314, 443, 171, 265, 408, 403,
526, 14, 153, 54, 2, 455, 544, 411, 492, 451, 144, 467], [391, 92, 231,
162, 412, 487, 256, 106, 104, 402, 371, 349, 484, 480, 400, 273, 389,
120, 22, 281], [277, 472, 103, 120, 313, 492, 243, 463, 81, 361, 400,
558, 374, 70, 207, 328, 363, 214, 85, 454], [50, 262, 148, 347, 165,
19, 26, 362, 312, 277, 273, 322, 486, 212, 542, 361, 342, 493, 320,
266], [551, 378, 303, 159, 547, 226, 228, 163, 531, 461, 345, 85, 360,
27, 511, 78, 129, 528, 257, 155], [2, 99, 57, 228, 498, 443, 36, 196,
407, 435, 529, 327, 396, 284, 58, 135, 268, 141, 416, 220], [175, 59,
337, 528, 178, 248, 517, 17, 132, 53, 456, 439, 561, 170, 554, 252,
413, 406, 247, 395], [454, 165, 513, 385, 191, 237, 85, 527, 485, 144,
328, 534, 413, 30, 53, 188, 82, 272, 438, 94], [127, 134, 91, 413, 417,
397, 25, 510, 509, 534, 539, 161, 2, 493, 147, 255, 538, 5, 143, 290],
[143, 499, 99, 242, 526, 321, 438, 316, 131, 116, 340, 290, 160, 520,
514, 41, 125, 327, 370, 69], [45, 366, 332, 397, 299, 13, 409, 43, 451,
109, 464, 417, 46, 377, 509, 180, 31, 62, 481, 362], [171, 328, 124,
527, 152, 58, 117, 191, 524, 73, 228, 32, 467, 370, 182, 131, 261, 513,
543, 185]]
```

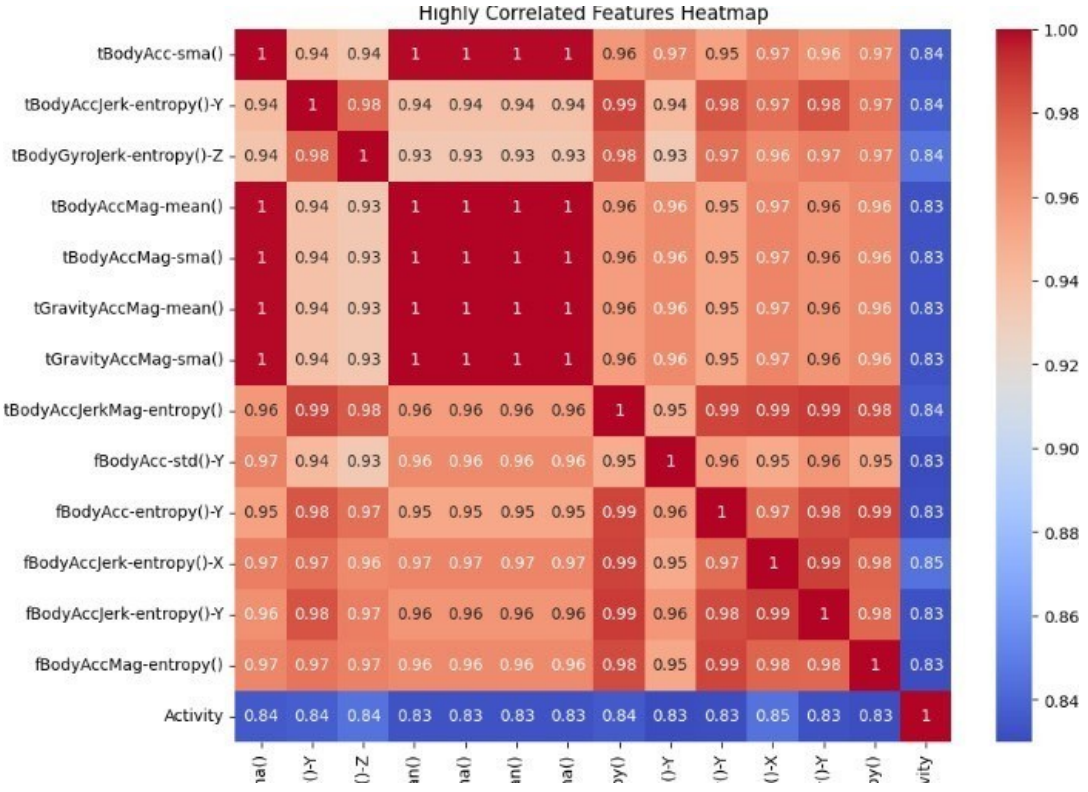
---

```
Each chromosome is  [[248, 173, 495, 520, 234, 377, 76, 353, 11, 245,
189, 180, 529, 387, 124, 512, 140, 398, 467, 47], [179, 415, 316, 266,
98, 339, 460, 244, 215, 330, 538, 183, 137, 510, 25, 81, 218, 20, 469,
243], [408, 512, 233, 270, 524, 292, 485, 278, 159, 141, 475, 468, 52,
537, 46, 115, 202, 252, 196, 495], [353, 128, 18, 325, 155, 12, 249,
413, 480, 383, 377, 114, 136, 505, 246, 15, 123, 549, 58, 478], [51,
548, 121, 166, 246, 432, 329, 487, 298, 408, 460, 118, 466, 349, 520,
468, 452, 538, 140, 258], [40, 14, 132, 77, 360, 434, 541, 346, 477,
```

```
520, 34, 141, 449, 103, 517, 31, 83, 536, 448, 392], [173, 490, 323,
400, 545, 452, 55, 156, 453, 132, 302, 94, 362, 188, 93, 352, 451, 191,
68, 320], [465, 366, 497, 268, 349, 324, 110, 69, 538, 120, 222, 203,
138, 370, 198, 252, 337, 509, 471, 77], [138, 193, 314, 443, 171, 265,
408, 403, 526, 14, 153, 54, 2, 455, 544, 411, 492, 451, 144, 467],
[391, 92, 231, 162, 412, 487, 256, 106, 104, 402, 371, 349, 484, 480,
400, 273, 389, 120, 22, 281], [277, 472, 103, 120, 313, 492, 243, 463,
81, 361, 400, 558, 374, 70, 207, 328, 363, 214, 85, 454], [50, 262,
148, 347, 165, 19, 26, 362, 312, 277, 273, 322, 486, 212, 542, 361,
342, 493, 320, 266], [551, 378, 303, 159, 547, 226, 228, 163, 531, 461,
345, 85, 360, 27, 511, 78, 129, 528, 257, 155], [2, 99, 57, 228, 498,
443, 36, 196, 407, 435, 529, 327, 396, 284, 58, 135, 268, 141, 416,
220], [175, 59, 337, 528, 178, 248, 517, 17, 132, 53, 456, 439, 561,
170, 554, 252, 413, 406, 247, 395], [454, 165, 513, 385, 191, 237, 85,
527, 485, 144, 328, 534, 413, 30, 53, 188, 82, 272, 438, 94], [127,
134, 91, 413, 417, 397, 25, 510, 509, 534, 539, 161, 2, 493, 147, 255,
538, 5, 143, 290], [143, 499, 99, 242, 526, 321, 438, 316, 131, 116,
340, 290, 160, 520, 514, 41, 125, 327, 370, 69], [45, 366, 332, 397,
299, 13, 409, 43, 451, 109, 464, 417, 46, 377, 509, 180, 31, 62, 481,
362], [171, 328, 124, 527, 152, 58, 117, 191, 524, 73, 228, 32, 467,
370, 182, 131, 261, 513, 543, 185]]
--------------------------------------------
Generation 1 : Best Fitness = 0
Iteration 1: Best Fitness = 0.8293629302405134
Iteration 2: Best Fitness = 0.8555614624091097
Iteration 3: Best Fitness = 0.8583006543677237
Iteration 4: Best Fitness = 0.8729892113740858
Iteration 5: Best Fitness = 0.8763104078092221
Iteration 6: Best Fitness = 0.8767540337512688
Iteration 7: Best Fitness = 0.8927486699213727
Iteration 8: Best Fitness = 0.9035479250636905
Selected Features: [353 548  57 239 234  13  55 527 298 149 169  32 138
528 350  15 451   7
 257 258]
```

```
import seaborn as sns
import matplotlib.pyplot as plt
# Calculate the correlation matrix
correlation_matrix =
pd.concat((Xtrain,pd.DataFrame(ytrain,columns=["Activity"])),
axis=1).corr()

# Set the threshold for correlation
threshold = 0.83

# Get the highly correlated features with "Activity"
highly_correlated_features =
correlation_matrix[abs(correlation_matrix.iloc[:,-1]) >=
threshold].index.tolist()

# Print the highly correlated features
print(highly_correlated_features)
```

```
# Filter the correlation matrix to include only highly correlated
features
filtered_corr_matrix =
correlation_matrix.loc[highly_correlated_features,
highly_correlated_features]

# Plot the heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(filtered_corr_matrix, annot=True, cmap='coolwarm')
plt.title('Highly Correlated Features Heatmap')
plt.show()
```




```
common_features = set([feature for i, feature in
enumerate(filtered_corr_matrix.columns) if i in selected_features]) &
set(highly_correlated_features)
print(common features)
```

```
{'Activity', 'tBodyAccJerkMag-entropy()'}
```

**RESULTS &DISCUSSION:**

| Method | AUC | CA | F1 | Precision | Recall |
|---|---|---|---|---|---|
| kNN | 0.998 | 0.966 | 0.966 | 0.966 | 0.966 |
| SGD | 0.988 | 0.981 | 0.981 | 0.981 | 0.981 |
| Random Forest | 1.000 | 0.981 | 0.981 | 0.981 | 0.981 |
| Neural Network | 0.999 | 0.986 | 0.986 | 0.986 | 0.986 |
| Naive Bayes | 0.925 | 0.727 | 0.732 | 0.752 | 0.727 |
| Logistic Regression | 0.999 | 0.986 | 0.986 | 0.986 | 0.986 |

**KNN (K- NEAREST NEIGHBOURHOOD)**

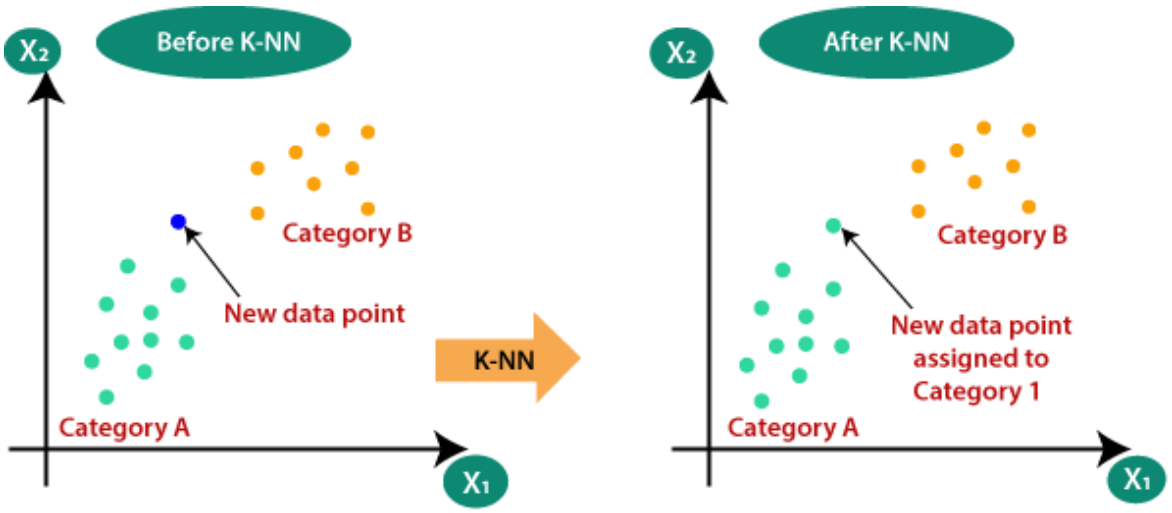
K-nearest neighbours (KNN) algorithm uses 'feature similarity' to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. We can understand its working with the help of following steps –

Step 1 − they want data frame for enforcing some method. And we must pack its instruction and perhaps even the relevant data within the first phase in KNN.

Step 2 −First; we will pick the K amount i.e. its closest information values. Some number could be K.

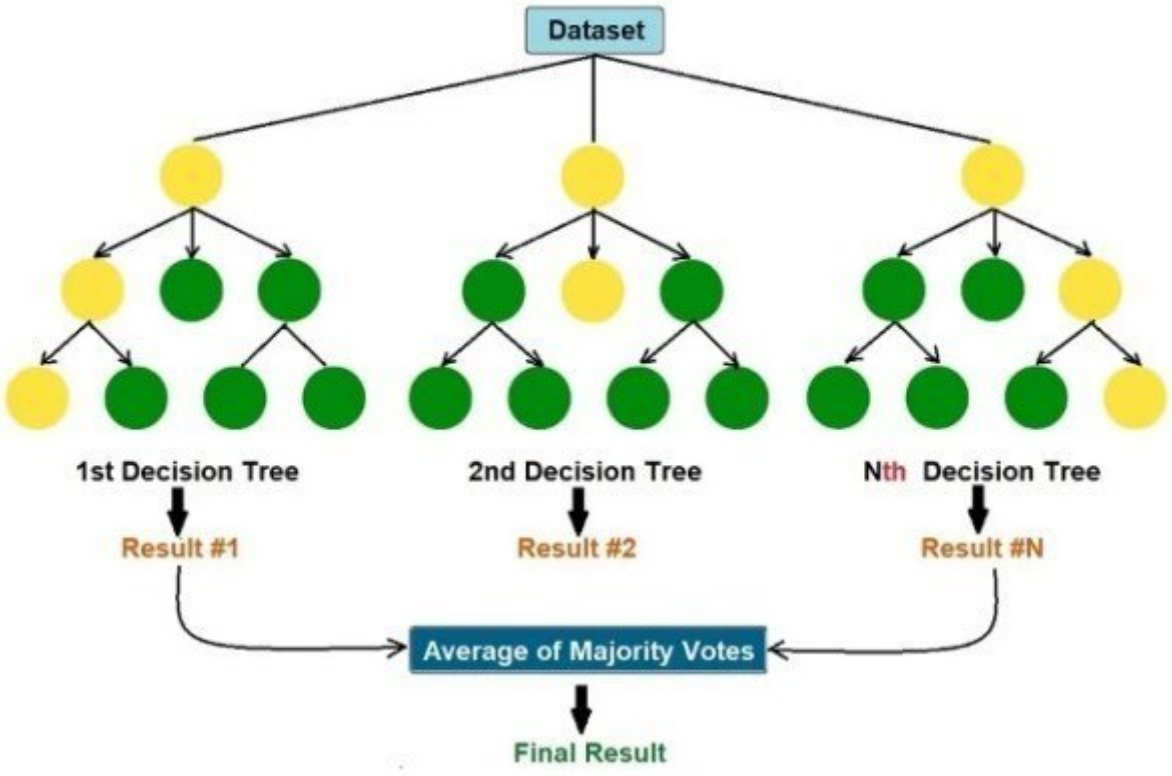
Step 3 − Does the preceding to every level well into the information − 58 3.1 − using any of the methods notably: Manhattan, Euclidean or Hamming distance measure the distance among testing data so each line of sample data. Its most frequently utilized form for range calculation is Euclidean. 3.2 − Now, based on the distance value, sort them in ascending order. 3.3 − Next, list the top K lines from both the list you have ordered. 3.4 −Now, the category would be allocated to both the check points based on its most common classes of those lines.

Step 4 – End




**RANDOM FOREST**

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. In our project for random forest algorithm we used 21 estimators that is nothing




**LOGISTIC REGRESSION**

Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given data set of independent variables.

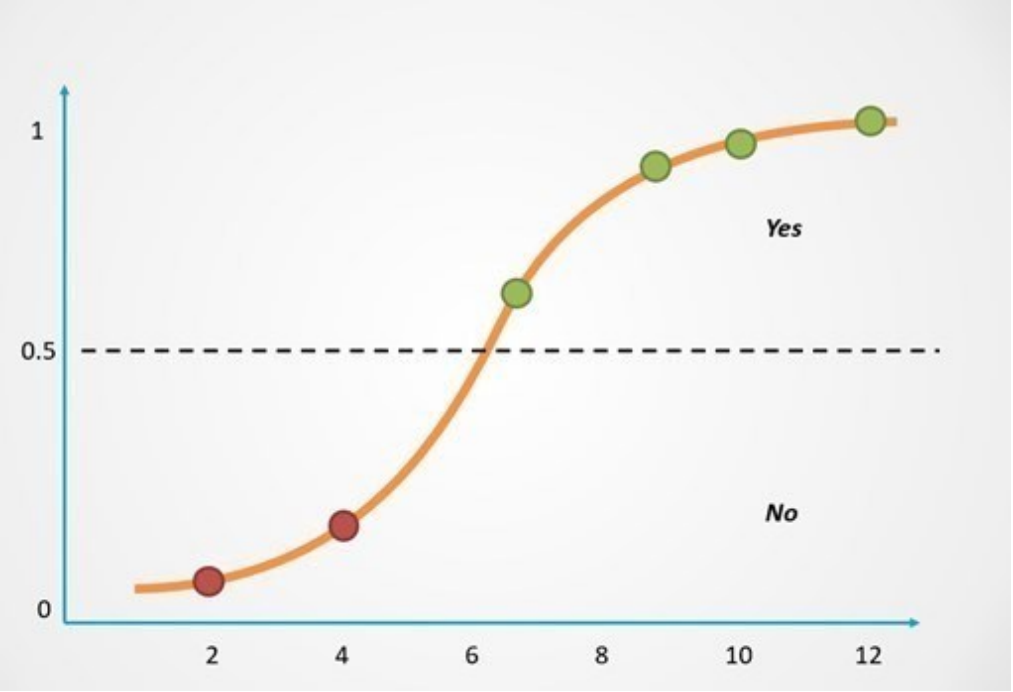
This type of statistical model (also known as *logit model*) is often used for classification and predictive analytics. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure. This is also commonly known as the log odds, or the natural logarithm of odds, and this logistic function is represented by the following formulas:

Logit(pi) = 1/(1+ exp(-pi))

ln(pi/(1-pi)) = Beta_0 + Beta_1*X_1 + … + B_k*K_k

In this logistic regression equation, logit(pi) is the dependent or response variable and x is the independent variable. The beta parameter, or coefficient, in this model is commonly estimated via maximum likelihood estimation (MLE). This method tests different values of

beta through multiple iterations to optimize for the best fit of log odds. All of these iterations produce the log likelihood function, and logistic regression seeks to maximize this function to find the best parameter estimate. Once the optimal coefficient (or coefficients if there is more than one independent variable) is found, the conditional probabilities for each observation can be calculated, logged, and summed together to yield a predicted probability. For binary classification, a probability less than .5 will predict 0 while a probability greater than 0 will predict 1. After the model has been computed, it's best practice to evaluate the how well the model predicts the dependent variable, which is called goodness of fit. The Hosmer–Lemeshow test is a popular method to assess model fit.



# CHAPTER-7

# CONCLUSION & REFERENCES

Increasing amount of work is being done to correctly identify Human Activity as it has many important applications. Using accelerometer data to recognize human activity is very convenient compared to classic ways recognition because it is in-expensive and mobile. Accelerometers are installed in all modern smartphones and is available everywhere. An accelerometer can also be embedded in a device other than smartphones. Recognizing activity in real time is great challenge which can be achieved efficiently only through using machine learning techniques. In this paper some important machine learning techniques have been analyzed and their results compared using python Classification Learner and Weka tool. In python Classification Learner the best model is Bagged Tree with 96.2% accuracy results is identified and it also gives good results in testing phase and on linear regression gives most accurate results compared to logistic, support vector and neural networks for this data set but with slight difference. Logistic regression id better because it is faster hence, should be used where real time recognition is required.

**Future Scope**

The future scope for a human activity analysis system using machine learning algorithms is promising, with opportunities for further advancements and applications in various domains. Here are some potential future directions:

**Advanced Machine Learning Techniques:** Continued research and development in machine learning techniques, such as deep learning, reinforcement learning, and transfer learning, can further enhance the accuracy and robustness of activity recognition models. These techniques can capture more complex patterns and dependencies in human activity data, leading to improved performance.

**Multimodal Sensor Fusion:** Integrating data from multiple sensors, including accelerometers, gyroscopes, GPS, and biometric sensors, can provide a more comprehensive understanding of human activities. Future systems could leverage multimodal sensor fusion techniques to combine information from diverse sensor modalities, leading to more accurate and context-aware activity recognition.

**Context-aware Activity Recognition:** Incorporating contextual information, such as time of day, location, environmental conditions, and user context, can improve the relevance and accuracy of activity recognition. Future systems could leverage contextual cues to adaptively adjust activity recognition models and provide more personalized insights and recommendations to users.

**Edge Computing and IoT Integration**: With the proliferation of IoT devices and edge computing capabilities, future activity analysis systems can leverage distributed computing architectures to process sensor data closer to the source. This can reduce latency, conserve bandwidth, and enhance privacy by processing sensitive data locally.

**Healthcare and Well-being Applications:** In the healthcare domain, activity analysis systems can play a significant role in remote patient monitoring, disease management, and rehabilitation. Future systems could incorporate physiological signals, such as heart rate variability and oxygen saturation, to provide holistic insights into users' health and well-being.

**Privacy-preserving Techniques:** Addressing privacy concerns remains a critical aspect of activity analysis systems. Future research could focus on developing advanced privacy-preserving techniques, such as federated learning, homomorphic encryption, and secure multiparty computation, to protect users' sensitive data while still enabling accurate activity recognition.

**Personalized Feedback and Intervention:** Future systems can leverage machine learning algorithms to provide personalized feedback and interventions based on users' activity patterns, goals, and preferences. This can empower users to make informed decisions about their health, fitness, and lifestyle choices, leading to improved outcomes.

**Integration with Wearable Devices and Smart Environments:** Integrating activity analysis systems with wearable devices, smart home environments, and ambient sensors can provide a seamless and pervasive monitoring experience. Future systems could leverage data from these sources to infer users' activities more accurately and provide proactive support in daily life activities.

**Longitudinal Studies and Population Health Monitoring:** Conducting longitudinal studies and population-level analyses using aggregated activity data can yield valuable insights into trends, patterns, and correlations related to human behavior and health. Future

research could focus on leveraging large-scale datasets to identify risk factors, predict health outcomes, and inform public health interventions.

In summary, the future scope for human activity analysis systems is vast and multifaceted, encompassing advancements in machine learning techniques, sensor technologies, privacy-preserving methods, and applications across healthcare, fitness, smart environments, and beyond. By addressing these challenges and opportunities, future systems have the potential to revolutionize how we understand, monitor, and promote human well-being.

**Future Enhancements:**

**Context-Aware Computing:** Further enhance the system's context-awareness by incorporating additional contextual factors such as weather conditions, social interactions, and user routines. Continuous Learning: Implement mechanisms for continuous model learning and adaptation to user feedback, improving the system's accuracy and relevance over time. **Gamification:** Introduce gamification elements to incentivize and motivate users to engage with the system and achieve their activity goals.

**Health Monitoring:** Extend the system to include health monitoring features, such as detecting anomalies or trends in activity patterns indicative of health conditions or changes. In summary, the proposed system offers a comprehensive solution for human activity analysis using smartphones and machine learning classification techniques. By leveraging the ubiquity of smartphones and advanced machine learning algorithms, the system aims to empower users to better understand and manage their daily activities, leading to improved health and well-being Top of Form

**REFERENCES**

[1] K. Eskaf, W. M. Aly and A. Aly, "Aggregated Activity Recognition Using Smart Devices," 3rd International Conference on Soft Computing & Machine Intelligence (ISCMI), Dubai, 2016, pp. 214-218.

[2] Y. Tian and W. Chen, "MEMS-based human activity recognition using smartphone," 35th Chinese Control Conference (CCC), Chengdu, 2016, pp. 3984-3989.

[3] D. Tao, L. Jin, Y. Yuan and Y. Xue, "Ensemble Manifold Rank Preserving for Acceleration-Based Human Activity Recognition," in IEEE Transactions on Neural Networks and Learning Systems, vol. 27, no. 6, pp. 1392-1404, June 2016.

[4] M. Zeng et al., "Convolutional Neural Networks for human activity recognition using mobile sensors," 6th International Conference on Mobile Computing, Applications and Services, Austin, TX, 2014, pp. 197-205.

[5] P.Casale, O.Pujol, and P.Radeva, "Human Activity Recognition from Accelerometer Data Using a Wearable Device," 5th Iberian Conference, IbPRIA, Las Palmas de Gran Canaria, Spain, 2011, pp. 289-296.

[6] S. Matsui, N. Inoue, Y. Akagi, G. Nagino and K. Shinoda, "User adaptation of convolutional neural network for human activity recognition," 25th European Signal Processing Conference (EUSIPCO), Kos, Greece, 2017, pp. 753-757. [7] M. Zubair, K. Song and C. Yoon, "Human activity recognition using wearable accelerometer sensors," 2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Seoul, 2016, pp. 1-5.

[8] P.Casale, sO.Pujol, and P.Radeva, "Human Activity Recognition from Accelerometer Data Using a Wearable Device," 5th Iberian Conference, IbPRIA, Las Palmas de Gran Canaria, Spain, 2011, pp. 289-296.

[9] E. Fullerton, B. Heller and M. Munoz-Organero, "Recognizing Human Activity in Free-Living Using Multiple Body-Worn Accelerometers," in IEEE Sensors Journal, vol. 17, no. 16, pp. 5290-5297, Aug.15, 2017.

[10] M. Panwar et al., "CNN based approach for activity recognition using a wrist-worn accelerometer," 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Seogwipo, 2017, pp. 2438-2441.

[11] M. Poorani, V. Vaidehi and P. Varalakshmi, "Performance analysis of triaxial accelerometer for activity recognition," Eighth International Conference on Advanced Computing (ICoAC), Chennai, 2017, pp. 170-175.

[12] S. Kalpana. "Classification of IRIS Dataset using Weka." International Journal of Computer Applications & Information Technology, 2020, 12(1), pp.287-29

**SRI SIVANI COLLEGE OF ENGINEERING**
**(affiliated to JNTU-GURAJADA VIZIANAGARAM)**
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

Program outcomes attained after completion of the project work with attainment levels

| Program Outcomes(POs) and Program Specific Outcomes(PSOs) | Attainment level | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| **PO 1.Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems. | | | |
| **PO 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. | | | |
| **PO 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. | | | |
| **PO 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. | | | |
| **PO 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. | | | |
| **PO 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 | | | |
| **PO 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 | | | |
| **PO 8. Ethics**: Apply ethical principles and commit to professional ethics andresponsibilities and norms of the engineering practice | | | |
| **PO 9. Individual and team work**: Function effectively as an individual, and as amember or leader in diverse teams, and in multidisciplinary settings | | | |
| **PO10.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 design documentation, make effective presentations, and give and receive clear instructions. | | | |
| **PO11.Project management and finance**: 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 manageprojects and in multidisciplinary environments. | | | |
| **PO12.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 technologicalchange | | | |