PERFORMANCE ANALYSIS OF MACHINE LEARNING ALGORITHMS FOR MONITORING HUMAN ACTIVITY RECOGNITION

RICHA GOEL

Thesis Report

FEBRUARY 2020

ACKNOWLEDGEMENTS

First and foremost, praises and thanks to God, the Almighty, for His showers of blessings throughout my research work to complete the research successfully.

This thesis would not have been possible without the invaluable guidance and support from **Akshita Bhandari M.Sc., Machine Learning, Research Assistant IISC Bangalore &Prof. Manoj Jayabalan, Post Doctorate Fellow LJMU**. Feedback on each stage was very important and helped me a lot to complete my research project. Their dynamism, vision, sincerity, and motivation have deeply inspired me. They have taught me the methodology to carry out the research and to present the research works as clearly as possible. It was a great privilege and honor to work and study under their guidance.

I did like to thank my parents for their love, prayers, caring, and sacrifices for educating and preparing me for my future.

I sincerely appreciate all the support from **UPGRAD** and **LJMU (Liverpool John Moores University)** during my research work. I also thank all the staff of UPGRAD and LJMU for their kindness.

Finally, I Would like to thank everybody who supported & encouraged me during my research project.

ABSTRACT

Over the last few years, the average age of the world has been growing, and devices for health monitoring have received considerable attention as they improve the quality and span of people's lives. Throughout various areas such as health care, wellness, and elderly care, recognition of human motions is an important task. Mobile applications using wearable sensors can now achieve this goal. These applications can be helpful for elderly or sick people for any anomaly found in their normal activities. Like, suppose older adults fall or need medical emergency help, then an alert system would be triggered, which can be used to connect to their relatives. The thesis presented a comparison of multiple Machine learning algorithms to predict human activities based on wearable sensor data and Smartphone data. We classified human activity based upon six popular machine learning algorithm (Random Forest, Deep Neural Network, KNN, SVM, GBM, and Logistics Regression)

We performed this comparative study on two UCI Datasets,1 is Batteryless Wearable Sensor Dataset, and the other one is Smartphone data.

As per our result, Random Forest outperformed for Wearable sensor Dataset with an accuracy of 99% and DNN performed for smartphone dataset with an accuracy of 94%. We made the comparison based upon multiple metrics like precision, recall, F-score, specificity, ROC Area as well other than accuracy as it is crucial to identify all the activities correctly.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

KNN- K – Nearest Neighbor

HAR-Human Activity Recognition

AUC-Under the curve

EEG-Electroencephalogram Helmet

DNN-Deep Neural Network

PCA-Principal component analysis

SVM-Support Vector Machine

RF-Random Forest,

GBM-Gradient Boost Machine

DNN- Deep Neural Network

ANN-Artificial Neural Network

USC-HAD -University of Southern California-Human Activities Dataset.

Table of Contents

CHAPTER 1

1 INTRODUCTION

1.1 Background of the study

The growth rate of the U.S. population's elderly group has been much higher than the overall population growth rate. It is estimated that Americans 65 and older will almost double from 52 million in 2018 to 95 million by 2060 and that the share of the population in the 65-and-old age group will increase from 16 to 23 percent. There is an unprecedented scarcity of health workers as compared to the increasing numbers of the elderly. A large number of Falls Incidents have been observed while using traditional care methods. These have not only led to extensive post-care and surgery (costing up to 5 billion dollars) but are also associated with additional suffering and mortality. Falls are the top cause of accidents and serious injuries, causing accidental deaths in older people. In such circumstances, technology can greatly assist by offering automated solutions for the problem at hand. The proposed work is capable of generating a timely response to alert the healthcare workers and the elderly by analyzing the wireless data streams acquired through wearable sensors by using machine learning techniques. By using this application, we can minimize the high-risk activities of older people in hospitals or aged care facilities. Correct recognition of such high-risk events can lead to an intervention to mitigate an event that can potentially cause further physical injury and mental distress. Proper identification of such high-risk incidents can contribute to action to reduce an incident that could potentially cause more physical injury and mental distress.

In the case of high-risk activities, it is critical that real-time activities are correctly identified. Most falls occur in the process of change of activities like sit to stand or stand to sit etc.

Falls occur commonly in hospitals, especially in older people with dementia or delirium, where about 30% of falls result in some type of injury. Falls in hospitals have been reported in the literature to occur inside the patient's rooms (84%) and during ambulation (19%)(Hitcho et al., 2004). Moreover, the majority of falls occur around the bed and chair area. (Vassallo et al., 2000) .Falls are costly as patients have a longer length of stay in hospital wards and other related expenses. The estimated cost of fall-related hospitalization in the United States of America is US$50 534 per person.

Machine learning is a powerful research technique in which complex associations are managed in extensive data, hidden patterns are discovered, and functional predictions are made.

The Scope of this research project is to predict human activity from Wearable Sensor data and smartphone data by using six Machine Learning Algorithm (**k-nearest neighbors(KNN), Logistics Regression, Support Vector Machine(SVM), Random Forest(R.F.), Gradient Boosting Machine(GBM) and Deep Neural Network Algorithm (DNN))** and perform a comparison study between those algorithms performance.

1.2 Problem Statement

For many countries around the world, the elderly population is on the rise (Jobanputra et al., 2019). The elderly may have many difficulties, such as their body, environment, mind, etc. Elders who have to live alone while their children go to work when their ability to support themselves declines are at an increased risk of not receiving immediate assistance in the event of danger.

This problem can be mitigated by using technology to develop an activity monitoring system for the elderly. These systems use sensor data to analyze the model to recognize the elderly activity.

The research presents a comparison of older people's perceived activity with sensor data and smartphone data by using Machine Learning Algorithm (k-nearest neighbors(KNN), Logistics Regression, Support Vector Machine(SVM), Random Forest(R.F.), Gradient Boosting Machine(GBM) and Deep Neural Network Algorithm (DNN))**.** As a modeling platform, we have used the Python Platform and UCI Machine Learning Repository data set to build and evaluate models.(Sriwichian and Muangprathub, 2019)

This study will predict Human activity by using six algorithms and compare the results to classify the behaviors of six algorithms, which will help to develop a tracking and monitoring system for older adults.

1.3 Aims & Objectives

The objective of this study is to develop, validate, and do a comparative study of popular machine-learning algorithms to predict and identify human activity using Wearable sensor data and smartphone data by applying Machine Learning techniques. The research objectives are formulated based on the aim of this study which is as follows:

- To analyze the pattern and relationship between the predictors and the risk factors.
- To compare the different predictive models.
- To evaluate the performance of different machine learning algorithms.

We will use below six machine Algorithm to identify human activity:-

1) Logistics Regression
2) K Nearest Neighbors Algorithm
3) Support Vector Machine Algorithm
4) Random Forest Algorithm
5) Deep Neural Network
6) Gradient Boosting Machine Algorithm

1.4 Research Questions (IF ANY)

A comparison of the Machine Learning Algorithm used to classify Human Activity for older people by using Sensor data and smartphone Data from UCI Dataset(Wearable Sensor Data and smartphone Data).

1.5 Scope of the Study

The research presents a comparison of older people's perceived activity with Logistics Regression, k-nearest neighbors, Support Vector Machine, Random Forest, Gradient Boosting Machine, and Deep Neural Network Algorithm. As a modeling platform, we have used the Python Platform and UCI Machine Learning Repository data set(Wearable Sensor Data and smartphone Data) to build and evaluate models.

1.6 Significance of the Study

We will be able to find a model that will be able to identify all human activity with reasonable accuracy. This model can be used in future applications to predict human behavior based upon a wearable sensor dataset or smartphone dataset.

1.7 Structure of the Study

The remaining study is structured in the following way:-

Chapter 2:-This section discusses the relevant research work and all the current solutions to predict human activity by using Machine Algorithm.

Chapter 3:- In this section, the data collection method, data cleaning, and processing steps to build the model and all the testing parameters to compare the model are presented.

Chapter 4:- In this section, all the analysis work like Data Preparation, missing value check, Univariate, Bivariate analysis, Splitting of data set between test and train Data set are discussed.

Chapter 5:- In this section, the performance evaluation Matrix for all the Machine Learning algorithms are discussed and identifies the model with a higher degree of accuracy to predict Human activity for sensor and smartphone data.

Chapter 6:- Finally, the conclusion and future work are presented in this section.

Appendix Section:-Research proposal, plan are attached in this section.

CHAPTER 2

2 LITERATURE REVIEW

2.1 Introduction

There is a lot of research being done for Human Activity Recognition (HAR) Problem, and it is one of the famous research areas. Data required to solve this problem can be easily found by using a smartphone or a low-cost wearable sensor. The outcome of this research can be used to design Multiple systems like Activity tracker, Alert system for Older people, Smart home at low cost.(Jobanputra et al., 2019)

In Human Activity Recognition problem various activity data can be collected by a wearable sensor or Smartphone sensor and various human activity like Running, Sitting, Sleeping, Standing, Walking can be easily identified, and Activity prediction result can be used in many systems like Medical Diagnosis, Smart Phone, Alert System and Activity tracker.

This section explains sensors and different Machine Learning algorithms used to predict Human Activity so far.

2.2 Human Activity Recognition:-

Human Activity Recognition is being used in multiple everyday life problems like in Hospitals or Eldercare. It can detect if any patients need help in case of falls, or it can track a person's daily activity and detect any difference in behavior and trigger an Alarm or call Emergency help if required. Due to these use of HAR, It is a very extensively used technology is current days. Data can be collected easily from Wearable sensors or smartphone or smartwatch. Still, processing of this data and predicting human activity is a challenging task as every human behavior is very different. Multiple Machine Algorithm is used to solve this problem so far. The sensor is also a significant part of the HAR problem, as it is crucial to collect accurate data. This data can be obtained from two types of sensors: External Sensors or Wearable Sensor.

Wearable Sensors are the devices that can be easily integrated into various wearable accessories like smartwatches, Activity trackers, or can be attached to person clothes. The external sensor is like a smartphone which is not entirely connected to the human body but can still track human activity data.

Recognition of human behavior has evolved in recent years as a result of its applications in m any areas such as health, protection, tracking, entertainment, and smart settings (Hussain et al., 2019).

"Activity recognition can be defined as the ability to recognize/detect current activity on the basis of information received from different sensors" (Hussain et al., 2019)

2.3 Sensors for Human Activity Recognition:-

There are typically two kinds of devices used for data collection:- Vision-based and sensor-based. Based on this, we can categorize the approach to human activity recognition in a two-part, vision, and sensor approach.

In a Vision-based approach, A Camera tracks human activity and gathers the information about human behavior. This data can be processed in the computer to capture Human activity. The camera-based approach has some advantages and some disadvantages. It's an easy way to get precise data, but privacy for this kind of sensor is a significant issue. Another issue with the sensor type is that in the absence of light, these sensors can not function. There is a lot of research being done based upon Vision-based Sensors, but Our focus is the Sensor-based approach for this project. (Hussain et al., 2019) .

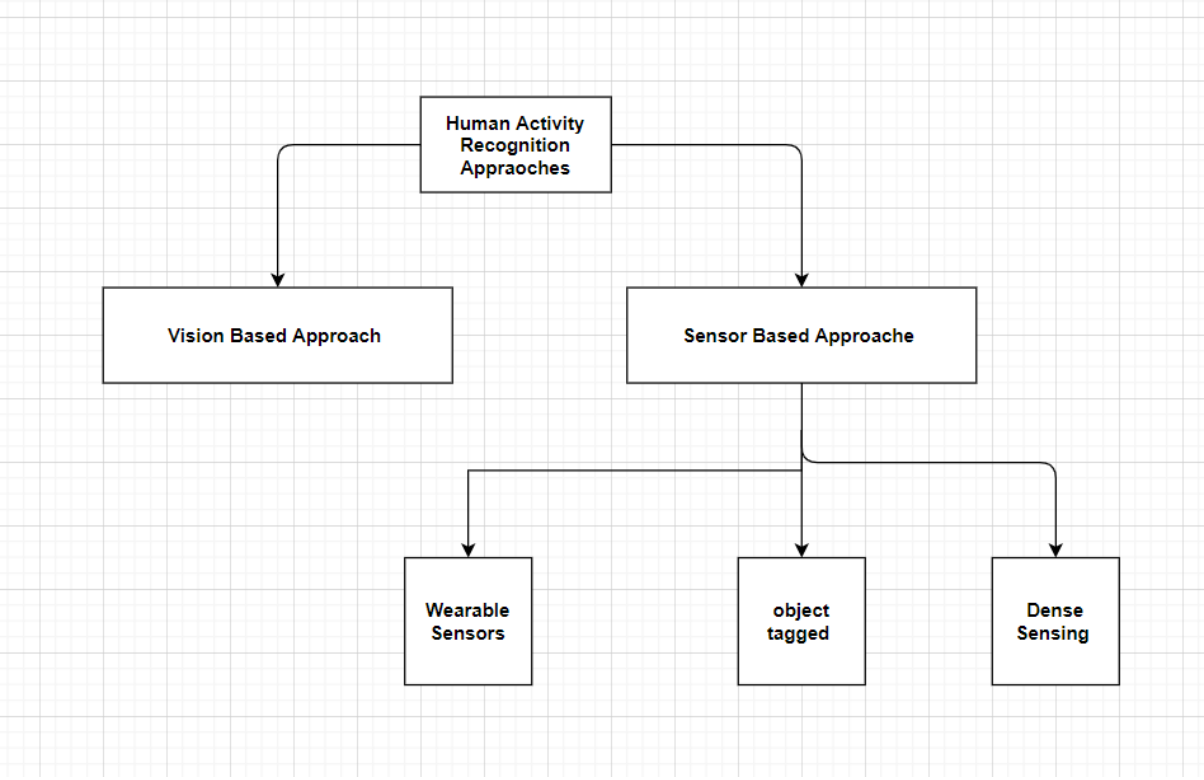


Fig 1-Human Activity Recognition Approach Classification

The sensor is fixed on the human body and collects human activity information in a sensor-based approach. There was a great deal of work on the detection of human behavior by using wearable sensor data. In this section, we will address multiple studies using different Machine algorithms to process data to classify human behavior.

2.4 Machine Learning Algorithm used for Human Activity Recognition:

2.4.1 Human Activity Recognition Using an Artificial Neural Network Algorithm

Min-Cheol Kwon & Sunwoong Choi(Kwon and Choi, 2018) presented an Artificial Neural network(ANN) approach to classify Human activity based upon smartwatch data. In this approach, data is collected from the smartwatch and sent to the server via Bluetooth. In this study,11 activity(Office work, Reading, Writing, Taking a rest, Playing a game, Eating, Cooking, Washing Dishes, Walking, Running, and Taking a transport) were identified by using 5 level layer ANN algorithm with an accuracy of 95%. This analysis was performed for window size 1,5,10,30,60 (1min), 120(2min) and 180(3 min), They also compared the performance of the most commonly used algorithms such as supervised learning: decision tree (D.T.), random forest (R.F.), and support vector machine (SVM) and found that ANN algorithm outperformed other algorithms for this dataset. According to the author's finding, the proposed system can be used for enhancing convenience and decrease wastage of energy.

2.4.2 Human Activity Recognition Using Random Forest Algorithm

Serkan Ball(Balli et al., 2019) recommended the use of smartwatch data and the Random Forest machine-learning algorithm to classify human gestures. Eight different daily activities (brushing teeth, walking, running, vacuuming, writing on the board, writing on the paper, using the keyboard and stationary state) data was collected from the smartwatch's accelerometer, gyroscope, step counter, and heart rate sensors. In this study, 8000 seconds of sensor data for eight different activities from five people are collected and transferred to a connected smartphone. They applied the PCA process to reduce features to 3 from 14 original features, and these 3 Priciniple component data were classified in 8 different human algorithms by using various machine algorithms like C4.5, K Neatest Neighbor, Random forest and SVM. These algorithms are compared based upon their performance matrix parameter like Area Under the curve (AUC), Classification Accuracy, F Measure.

As per this study, Random forest was the most successful approach to classify human activity with an accuracy of 97%. One of the critical advances in the research is the use of pedometer and heart rate sensors in addition to motion sensors (accelerometer and gyroscope). It is found that the use of step counter data improves the accuracy of the Human Activity Recognition Problem.

Human Activity Recognition can be used to identify and avoid risky acts such as abduction and the absence of older people and young children. The author suggested that in the future, more

behaviors such as Eating, Smoking, Cooking, Hand Shake, and Hand Waive can be monitored using the same method.

2.4.3 Human Activity Recognition Using Deep Learning Neural Algorithm:-

Ronao et al. (Ronao and Cho, 2016) collected data for 30 volunteers using a smartphone accelerator and Gyroscope sensor and processed data by using the Deep Neural Network algorithm and achieved 95.75% accuracy. Sensor data is sampled at 50 Hz, divided into 129-value windows with an overlap of 50 percent. The 128-value vector is an example of one Human activity.

In this research, the SVM algorithm is also studied, and it is found that SVM provides good accuracy in the classification of fixed activity like standing, lying. Still, Deep Learning Neural Network is more efficient in terms of moving tasks such as walking upstairs and walking downstairs. The author listed some of the future scopes that can be incorporated in this research, such as cross channel pooling instead of max pooling, the combination of convent and SVM, and incorporating frequency convolution together with time convolution.

2.4.4 Human Activity Recognition Using SVM Algorithm

A substantial amount of research work has been performed using the SVM algorithm in the Human Activity classification. The SVM algorithm demonstrates high precision in predicting static activity like standing, lying, sitting. Zubin & Soujanya(Sunkad and Soujanya, 2016) aimed to find the best subset of hyperparameter(Regularization parameter, SVM kernel) for SVM classifier for Human activity Recognition problem. In this research data is obtained from the University of Southern California-Human Activities Dataset (USC-HAD).

Grid Search Algorithm is used to find the optimized hyperparameter. The optimized hyperparameter that is obtained from this study was Kernal as RBF, Regularization parameter(C) as 100, and Υ as 0.001. The maximum accuracy achieved from this experiment was 99%. The future focus of this work can be the optimization of another classifier algorithm for high accuracy.

2.5 Related Research Publications:-

Feature selection is a crucial concept of machine learning that significantly impacts the efficiency of the model, and it is a process to select the most relevant feature out of a complete dataset that contributes more to output (predictive variable). There are multiple algorithms available to choose features. In this study, the author(Capela) used three feature selection algorithm (Relief F, Correlation-based Feature Selection(CFS), and Fast Correlation based Filter(FCBF) to reduce feature and classified these feature subset data by using the classifier

(Support Vector Machine, Naïve Bayes, and Decision Tree). Feature selection algorithm has a lot of benefits as data is reduced, such a computational burden and processing time are reduced. Capela performed this study on 76 signal features collected from 44 people for 18 different tasks. The author suggested applying this study for a bigger data set and people having more health problems like neurological disorders. (Capela et al., 2015)

One of the critical applications for Human activity recognition is fall detection for older people so that in case of any fall, an alert can be generated, and support can be provided to the person in minimal time to avoid any fatality. Villasenor presented a study on Fall detection by using multiple sensors (wearable sensor, ambient, and vision) and multiple classifier algorithm. In this study, four volunteer data (two male and two female) were collected by using sensors(four IMUs, one Electroencephalogram Helmet (EEG), and four Infrared sensors). By using this publicly available dataset, a comparison study has been performed, and it is concluded that better results can be achieved by hyper-tunning the parameter. (Martínez-Villaseñor et al., 2018)

Torres RL presented a study that can provide real-time tracking for older people. As per this study, the wireless wearable sensor gives a better result than the localized sensor on the bed and chair. The author developed a movement monitoring tracking system based upon wireless sensor data received from 14 healthy volunteers with an accuracy of 94%, precision of 78%, and F-score of 84%.

As a potential scope of this study author suggested that more work can be done by placing a sensor on the shoulder and clothing with integrated sensors. (Torres et al., 2016)

2.6 Discussion

Recognition of the Human activity is the origin of multiple application like Eldercare, Fitness tracker, Smart home, Health monitoring. In all of these applications, data is collected from various sensors, and then data is processed through multiple Machine algorithm to predict the output. Two types of Sensors are generally used to collect the data that is a wearable sensor and an external sensor. The camera is an example of an external sensor, and a smartwatch & smartphone is a type of wearable sensor. In some studies, PCA and feature selection algorithm are used to reduce features, and then data is processed through multiple classification algorithms like SVM, Random Forest, and Deep Neural Network. (Slim et al., 2019) . Recently a lot of studies have been done on the Deep Neural Network algorithm, but a machine with higher configuration is required for this. We will be doing a comparative study of multiple classifiers like Logistics regression, KNN, SVM, Random Forest, GBM, and DNN.

2.7 Summary

This section presented multiple studies that use various machine learning algorithms (traditional and deep learning) to predict the output based upon data collected from wearable and external sensors. As per these studies, Deep learning Neural network shows better accuracy for moving activities like running, walking upstairs, walking downstairs, and for non-movement events like lying, sitting, SVM can provide satisfying accuracy. Human activity recognition findings can be used in various applications like smart homes (where energy can be saved by predicting human behavior), elder care (where fall detection or other problem can be identified, and assistance can be provided to prevent fatality), fitness tracker, etc.

CHAPTER 3

3 RESEARCH METHODOLOGY

3.1 Introduction

This research will be conducted through modeling, data processing, and testing following the Data analytics lifecycle process. (Sriwichian and Muangprathub, 2019)

Below steps will be followed in this study:-

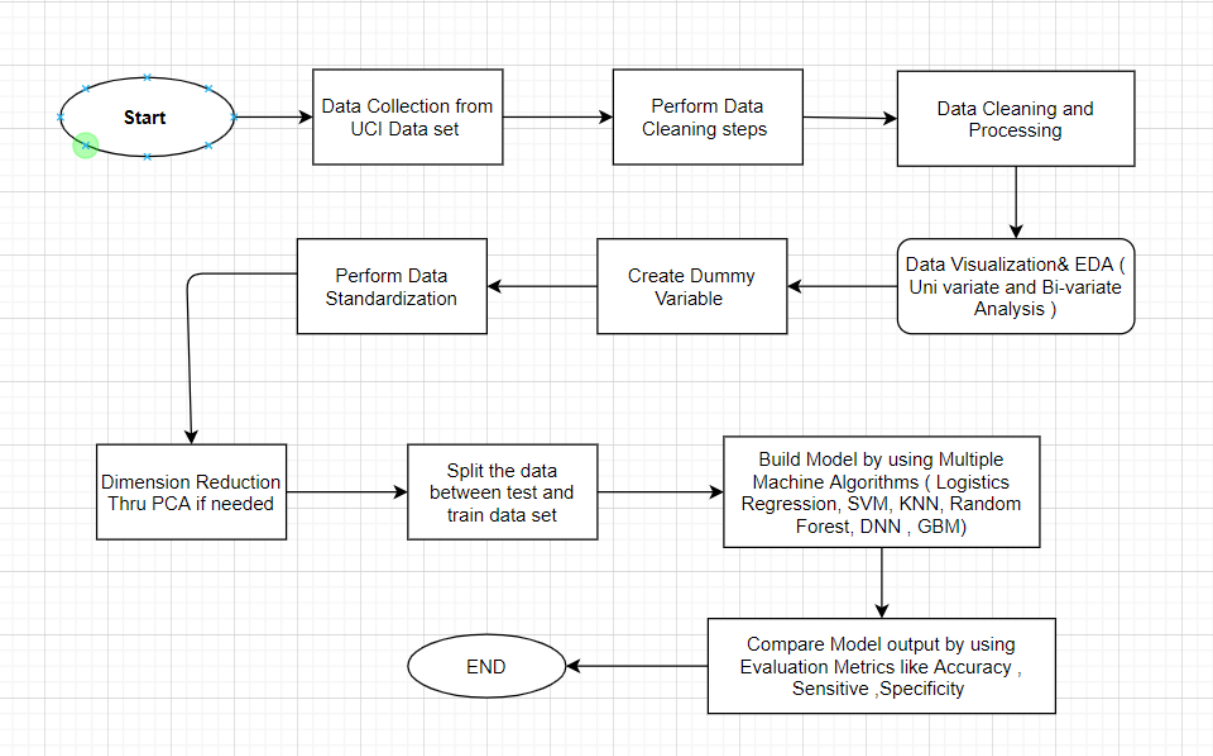


Fig 2:- Flow Diagram of RESEARCH METHODOLOGY

The problem of monitoring elderly people's behavior through the use of sensor information is an important question. We will be studying comparison between multiple machine learning algorithms (traditional and Deep Neural networks), and we hypothesize that the Random Forest Regression algorithm can perform better than other methods in predicting human activity.

3.2 Research Methodology

- In research methodology, we have outlined below steps used for this study
- Data Collection
- Data Check and Data Cleaning steps
- Exploratory Data Visualization(Univariate and Bivariate)
- Create dummy Variables if needed
- Perform Dimension Reduction thru PCA

- Model building by using various Machine Algorithm
- Model comparison based upon metrics like confusion Matrix, sensitive, specificity

3.2.1 Data Collection

We performed this study on two datasets 1, Batteryless wearable sensor dataset for healthy older people, and 2. Smartphone Sensor dataset.

Batteryless wearable sensor dataset for healthy older people

This dataset includes motion data of 14 stable older people between the ages of 66 and 86. They performed multiple scripted activities, and data is collected from a wearable sensor placed on the top of their clothing. Participants lived in two separate rooms (S1 and S2) during this study. S1 room has four RFID antennas .out of those, one RFID antenna is placed on the ceiling level, and three others are placed on the wall level. ("UCI Machine Learning Repository: Activity recognition with healthy older people using a batteryless wearable sensor Data Set," n.d.)

S2 room has three RFID antennas, and out of those, two are placed on ceiling level, and one is placed at the wall level.

During this study, participants performed below activity that is commonly performed by older people.

- Walking to the chair
- Getting out the chair
- Sitting on the chair
- Walking to the bed
- Lying on the bed
- Getting off the bed
- Walking to the door

This data set is in csv format. There are two folders. One is for Room S1 & Another folder is for Room2.

Based upon the above mentioned activity, sensor can predict four activity as below:-

1- Sit on bed
2- Sit on chair
3- Lying
4- Ambulating like Getting off the bed, Getting off the chair

Below fields are present in dataset:-

Column Name	Detail
Time	Time(Second)
AGF	Acceleration rating in G for frontal axis)
AGV	Acceleration rating in G for Vertical axis
AGL	Acceleration rating in G for Lateral axis
IDARS	ID of antenna Reading Sensors
RSSI	Received Signal Strength indicator
Phase	Phase
Frequency	Frequency
Activity	1-Sit on bed 2-Sit on chair 3:-Lying 4-Ambulating

Table1 –Columns of a wearable sensor-based dataset.

In the case of older people, a very sudden transition is not possible so four seconds window should be enough to identify human activity so any activity that is done at least for 4 seconds can be classified through a Machine learning algorithm.

Smartphone Dataset

For this dataset,30 volunteers(Age Range 19-48) performed some simple tasks with a waist-mounted smartphone with integrated sensors. Participants showed six primary activity like Lying, Sitting, Standing, Walking Upstairs, Walking downstairs and Walking

They also performed some transition activities like Stand to sit, Sit to lie, Lie to sit, and Stand to lie. By using a smartphone integrated accelerometer and gyroscope, three axial linear acceleration and three axial angular velocities at a constant rate of 50 Hz are obtained. The data set is randomly divided into two groups. One is 70% data that will be used as a training dataset to build the model & the other one 30% is test data that will be used as a testing dataset to test and validate the model.

Data columns:-

We have used a pre-processed version of the UCI data dataset instead of the raw dataset in this study. Below pre-processing has been done on raw data

- Filter preprocess accelerometer and gyroscope data using noise filter.
- Divide data into 2.56 seconds (128 data points) fixed frames with a 50 percent overlap.

- Divide the accelerometer data into components of gravity (total) and body motion.

For each 2.56 window, multiple time and frequency variable were extracted that is used widely for Human activity recognition. The outcome of preprocessing was 561 variable.
561 signals are derived from the below time and frequency variables:-

tBodyAcc-XYZà This parameter is measured in three directions (X, Y& Z)

tGravityAcc-XYZ- This parameter is measured in three directions (X, Y& Z)

tBodyAccJerk-XYZ- This parameter is measured in three directions (X, Y& Z)

tBodyGyro-XYZ- This parameter is measured in three directions (X, Y& Z).

tBodyGyroJerk-XYZ- This parameter is measured in three directions (X, Y& Z).

tBodyAccMag-

tGravityAccMag

tBodyAccJerkMag

tBodyGyroMag

tBodyGyroJerkMag

fBodyAcc-XYZ- This parameter is measured in three directions (X, Y& Z).

fBodyAccJerk-XYZ- This parameter is measured in three directions (X, Y& Z).

fBodyGyro-XYZ- This parameter is measured in three directions (X, Y& Z)

fBodyAccMag

fBodyAccJerkMag

fBodyGyroMag

fBodyGyroJerkMag

Column type	No of Columns
mean(): Mean value	33
std(): Standard deviation	33
mad(): Median absolute deviation	33
max(): Largest value in array	33
min(): Smallest value in array	33
sma(): Signal magnitude area	17
energy(): Energy measure. Sum of the squares divided by the number of values.	33
iqr(): Interquartile range	33
entropy(): Signal entropy	33
arCoeff(): Autorregresion coefficients with Burg order equal to 4	80
correlation(): correlation coefficient between two signals	15
maxInds(): index of the frequency component with largest magnitude	13
meanFreq(): Weighted average of the frequency components to obtain a mean frequency	13
skewness(): skewness of the frequency domain signal	13
kurtosis(): kurtosis of the frequency domain signal	13
bandsEnergy(): Energy of a frequency interval within the 64 bins of the FFT of each window.	126
angle(): Angle between to vectors.	7

Table2 –Columns of smartphone sensor-based dataset

3.2.2 Data Cleaning & Pre Processing

Before conducting further processing, we will be performing the below steps to prepare data for model building.

3.2.2.1 Data check and missing value imputation

There can be various type of data quality issue that needs to be resolved before data analysis. Otherwise, it becomes difficult to analyze data, and due to those issues, we might get an error in the result or irrelevant result. So once we have data, it is important to check those quality issues and fix them if needed.

The data cleaning process can be different for each data type, but there is some basic check that needs to be done for almost every data set.

- Delete unnecessary Columns – Select the columns that are related to the analysis; other columns can be dropped.
- Missing value imputation- Check all the missing values in a dataset, sometimes these missing values can be like X, NA, Not found. We can impute missing value in the following way
    - Delete rows/columns: If there are a lot of missing value in any particular rows or column, for example, if more than 50% values are missing, then those columns or rows can be dropped.
    - Fill value based upon business knowledge: Some time it is possible to fill those values based upon business knowledge; for example, if a zip code is provided, then the state can be filled, or the country is provided, then time zone can be filled.
    - Replace with Mean or Mode value: Substitution with the median for continuous variables and mode for categorical can be done.

3.2.2.2 Dummy Variable Creation

If we have a non-numeric variable in the dataset, then these variables are known as categorical variables, which must also be converted in the form of 0 and 1 before processing the model. For example, a categorical variable has n level, then (n-1) dummy variable will be created. Those dummy variable will have value as 0 and 1 value for each level . for example lets a variable (Age Group) has three value Adult, Child and Teen then two dummy variable will be created as below:-

Age Group	Adult	Teen
Adult	1	0
Teen	0	1
Child	0	0

Table 3- Dummy variable creation logic

3.2.2.3 Standardization of the Dataset

There are two major methods to scale the variables, i.e. standardization and MinMax scaling. Standardization brings all of the data into a standard normal distribution with mean zero and standard deviation one. MinMax scaling, on the other hand, brings all of the data in the range of 0 and 1. The formulae in the background used for each of these methods are as given below Standardization:-

$$x = \frac{\big(x - mean(x)\big)}{sd(x)}$$

MinMax Scaling

$$x = \frac{(x - min(x))}{max(x) - min(x)}$$

3.2.3 Exploratory Data Analysis:-

Exploratory data analysis is an approach of analyzing data to summarise their main characteristics, often with visual methods. Exploratory data analysis is the very first and important step in analysis of any kind of data. EDA is not just a specific set of procedures, but it is an approach that seeks to explore the most important and often hidden patterns in a data set. EDA involves exploring the data and coming up with a hypothesis about it. Later on this hypothesis is tested using hypothesis testing methods. Often statistics or data scientists use EDA to look at the data and seeking to understand it. Below are some methods we use to perform Exploratory Data Analysis:

3.2.3.1 Univariate analysis:-

Univariate analysis is the simplest method of data analysis where only one variable is found in the data being analyzed. Since it is a single variable, it is not about causes or relationships. The primary aim of the Univariate analysis is to classify the data and to identify trends within it. It involves understanding Metadata Description, Data distribution plots, and summary metrics, which provides central tendency measures like mean median, mode. It also provides the spread of the data, also called dispersion, and the measure include maximum, minimum, range, variance, quartiles and standard deviations, etc. We also plot frequency distribution tables, pie-charts, histograms, bar charts, and frequency polygons, etc.

3.2.3.2 Bivariate analysis:-

Bivariate analysis is done for continuous variables and is used to find the relationship between two variables. It involves creating a Correlation matrix or creating a scatterplot etc.

Correlation is a metric to find the relationship between the variables. It is a number between -1 and 1, which quantifies the extent to which two variables 'correlate' with each other.

- If one increases as the other increases, the correlation is positive
- If one decreases as the other increases, the correlation is negative
- If one stays constant as the other varies, the correlation is zero

In general, a positive correlation means that two variables will increase and decrease together, e.g., an increase in rain is accompanied by an increase in humidity. If one variable increases while others decrease, then it is a negative correlation, e.g., in some cases, as the price of a commodity decreases, its demand increases.

A perfect positive correlation means that the correlation coefficient is exactly 1. This implies that as one variable move, either up or down, the other one moves in the same direction. A perfect negative correlation means that two variables move in opposite directions, while a zero correlation implies no relationship at all.

By using Scatterplot, we plot one variable against another on a Cartesian plane that is X and Y-axis. On the X-axis we plot the independent variable while we plot the dependent variable on the Y-axis. When we get a line or curve on the cartesian plot, then the data has a relationship or correlation.

3.2.3.3 Multivariate analysis:-

Multivariate analysis is similar to Bivariate analysis, but it involves analyzing three or more variables. We deal with more than one dependent variable and one independent variable. There are multiple ways to perform multivariate analysis, depending on our goals. Some of these

methods include Cluster Analysis, Additive Tree, Factor Analysis, Canonical Correlation Analysis, Correspondence Analysis / Multiple Correspondence Analysis, Partial Least Square Regression, Generalized Procrustean Analysis, MANOVA, Multidimensional Scaling, Principal Component Analysis / Regression / PARAFAC, Multiple Regression Analysis, and Redundancy Analysis, etc.

3.4 Proposed Method (Model Building)

First, we will define a cohort and will apply feature selection techniques PCA (Principle Component Analysis) and then divide it into training, testing, and validation samples. For Opioid overdose prediction, we will be developing, and testing algorithms using below commonly used machine learning approaches Principal Component Analysis (PCA), Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine(SVM), Random Forest, Gradient Boost Machine (GBM), Deep Neural Network (DNN).

3.4.1 Principal component analysis (PCA)

PCA is a pre-processing data method that helps in decreasing the complexity of the model by using a dimensionality reduction algorithm. Let's say we have a data X with m-dimension. Then PCA provides a data Y with n-dimension with minimum loss. This is achieved by determining the projection vectors in the direction of the maximum variance of data X. The resulted projection vectors have unique values in different spaces. Two vectors x and y are defined on the same plane, and the projection of x on vector y is given by the function: [30].

$$f=\langle x,y\rangle|x|\cdot y \tag{1}$$

This Projection vectors give us the covariance of the data. After that, eigenvalue – eigenvector Transformation is used on the obtained covariance of the data, and sorting in descending order of eigenvalues is completed. A matrix of nn eigenvectors with respect to ordered dd eigenvalues is obtained. This gives a W projection matrix, which explains the best covariance. As we multiply the projection matrix and Data X, we achieve the dimensionally reduced date for further processing. The equation is provided below:

$$Y=WTXY=WTX \tag{2}$$

3.4.2 Logistic Regression

Logistic regression is a binary classifier type and is a linear model. It analyses a set of data points with respect to one or more independent variables, also known as input variables. If these input variables vary, the outcomes can be changed. Finally, Logistic Regression gives us a best-fitting model by using the below equation:

$$Yi = \alpha + \beta X_i + \varepsilon^i \ where\ E(\varepsilon^i) = 0$$

Data:

Yi = Outcome variable (example response , dependent)

X_i = Predictor Variable (Independent Variable or explanatory also known as co-variate)

α = Intercept

β = Slope

A Logistic Regression is termed as a Multivariate logistic regression when two or more independent variables predict the value of a dependent variable.

Some assumptions for Logistic Regression:

- A binary logistic regression uses a binary dependent variable, while an ordinal or categorical logistic regression uses a categorical dependent variable.
- Observations need to be independent of each other, and there should not be duplicated, repeated, or matched observations.
- There should be least or zero multicollinearity among the independent variables meaning no higher correlation between them.
- Independent variables should be a linear combination of log odds.

3.4.3 K – Nearest Neighbor (KNN)

K-NN is a multiclass non-linear modeling technique that can be used for both classifications as well as regression problems. It classifies each observation or data point of the test set by analyzing the nearest neighbors from the training set. This data point is assigned to the same class, which is very common among its neighbors. K-NN algorithm does not make any assumptions on the underlying data, which means it is non-parametric. It uses all training data to do the classification and does not pre-train itself (uses lazy learning).

This is highly effective for many classification problems with a small number of features that is a classification problem with low dimensionality and is also very simple in implementation

and understanding. For classification to the K nearest neighbors, distance functions are used. Hamming distance function is used for Categorical variables while anyone out of Manhattan, Euclidean or Minkowski distance function is used for continuous variables.

3.4.4 Support Vector Machine (SVM)

SVM is a linear model but can also be used as a non-linear model by adding a kernel to it. Let us suppose we have n number of features. We plot these features in n-dimensional space as data items by using the values of each feature as a value of a particular coordinate. As an example, let us consider we have two features humidity and temperature of five cities. Now we will plot these two variables in a two-dimensional space such that each city has two coordinates. Here we have two coordinates, but similar n-coordinates corresponding to n-features are referred to as support vectors. Vectors are points in a two-dimensional space. In the real world, we have data sets of n-dimensions. So in an n-dimensional space, it is easier and sensible to do Vector and matrix manipulations and arithmetics. Thus these are referred to as vectors but not just a point.

Now coming back to our example of five cities and two features, we try to find a line which splits the data features into two differently classified groups such that the distance from the closest point of each feature of the two groups is maximum. As shown in the above example, the green line is farthest from the nearest point in each feature group. So this green line acts as our classifier.

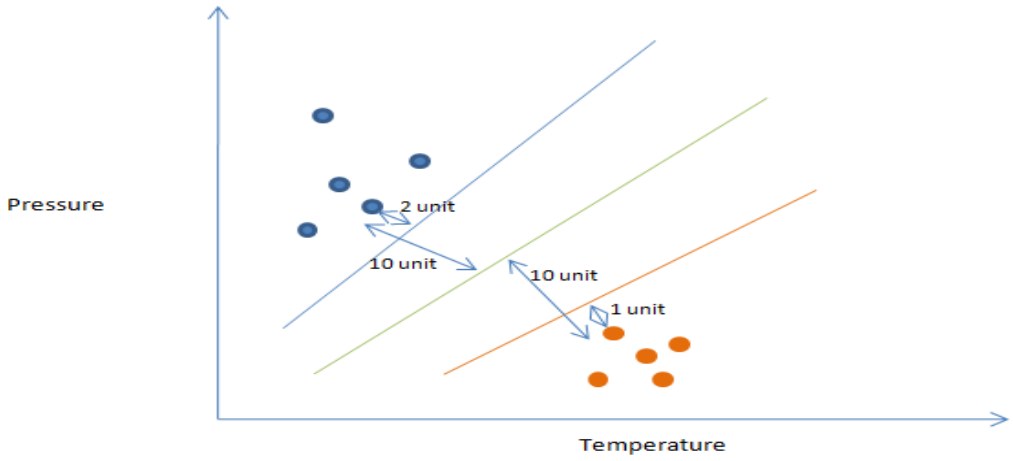


Fig 3-SVM Logic

3.4.5 Random Forest

Random forest is a supervised learning algorithm and is an advanced version of the decision tree algorithm. It is termed as "Forest" because it builds upon an ensemble of decision trees by using a training method called "bagging". Merging multiple learning models together can increase the overall result of the whole model combination, this is a general concept of "Bagging". Multiple decision trees are created and merged together to achieve a more accurate, reliable and stable prediction. Random forest can be used for both classification and regression problems. Random Forest gives this a huge benefit, as most of today's machine learning algorithms use either classification techniques or regression techniques. Let us look at the random forest in classification.

Figure below shows Random forest with two trees:

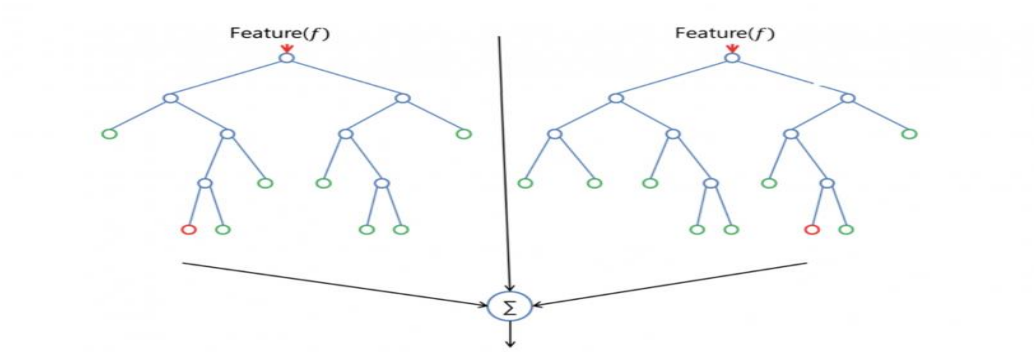


FIGURE 4- Random Forest Algorithm

The random forest has nearly identical hyperparameters to the decision tree or baggage classification. Luckily, there is no need to use any bagging classifier in random forest because we can simply use the random forest classifier. We can also perform regression tasks with random forest using a regressor algorithm. As the decision trees grow, Random forests add randomness to the model. It looks for the best feature in a random subset of features instead of looking for the most appropriate feature during node split. This results in more randomness and diversity, leading to a better model. We can also change the random thresholds for each feature, which also adds to the randomness. For splitting a node, unlike a decision tree where a set of fixed feature is utilized, a random subset of features is considered.

Discussing advantages and disadvantages:-

The flexibility and versatility of random forest is one of its biggest advantages. It can be employed for regression as well as for classification tasks, and the relative significance to the input features can also be easily seen. Random forests are often very convenient, as their predictive effect is often attributed to the default hyperparameters used.The output hyperparameters is very easy to understand due to less count. Overfitting is one of the main challenges in machine learning, random forest classifier overcomes this. The classifier does not overfit a model if there are enough decision trees in the forest. Random forest has one key drawback that if the decision trees are large in count, it will make the algorithm too sluggish and inefficient for real-time predictions.

Random Forest algorithms are usually easy to train but very slow to predict. As more and more decision trees are included for correct prediction, the model becomes slower. The random forest algorithm in most real-world applications is fast enough but it can certainly happen that run-time efficiency is very necessary and other approaches may be preferred over Random Forest. Random forest is a predictive modeling tool and not a descriptive tool. That means other methods will be easier if we are looking for a summary and explanation of the relations in the data.

If someone need to train in model development process, Random Forest is simple and great choice to see the complete modelling process. It has a good performance, flexibility and can handle murltiple typr of features such as binary, numerical and categorical etc. We sure have some models performing better than Random Forest but advantages of using Random Forest cannot be denied.

3.4.6 Gradient Boosting Machine (GBM)

In 2001, Friedman invented "Gradient Boosting Machine," also known as MART (Multiple Additive Regression Trees) . GBM model is an example of ensemble learning(many models to work together). These models are built-in series, and in each successive model, the weights are adjusted based on learning from the previous model. Gradient boosting is a special type of boosting technique where it works on reducing errors sequentially.

Mathematically it can be shown as below:-

GBM predicted target $= F0 + \beta1*T1(X) + \beta2*T2(X) + \ldots + \beta100*T100(X)$

Where F0 is the starting value for the tree series, X is a vector of pseudo-residual values remaining at each point in the series, T1(X), T2(X), T3(X),… are trees fitted to the pseudo-

residuals, and β1, β2,… are coefficients of the tree node predicted values that are computed by the GBM algorithm.

- **Hyperparameter for GBM**

    1. **Number of trees**:- The total number of the tree that are ensemble sequentially to provide output.
    2. **Learning rate**:-It determines the weightage of each treen on final output.
    3. **Tree depth**: It is the depth of the individual tree.
    4. **Minimum number of observations in terminal nodes**: It determines the complexity of each tree

3.4.6 Deep Learning or ANN

Artificial Neural network is a computational algorithm based on the mechanism of neural networks that function in a way identical to the human brain. The human brain is a chain of millions of Neurons, processing the information, and transmitting the signal to another part of the body.

Deep learning Neural network is a kind of Artificial Neural Network(ANN). The basic components of these algorithms are called artificial neurons. These neurons are a type of processing unit that inputs external data and generates an output. ANN includes multiple processing units that process the information and process the output. Each unit receives one or multiple inputs and produces the output. The output of one processing unit works as an input of the next processing unit and produces output. It processes each layer of neurons, and the final output is obtained from terminal neurons

The neural network can be used for classification as well as for the Regression of continuous Output.

A neural network can include three layers:-

- Input layer:- Input layer is used to receive raw input. Input layers and do not alter the data. Just feed each value to the input of the hidden layer. A number of nodes in the input layer are generally equal to the number of variables in the input dataset. For example, in the above figure, we have one input variable (x), so a number of nodes in the input layer will be one.

- Hidden Layer:- In one Deep learning network, there can be multiple hidden layers. The hidden layers take input from the output of the input layer and process it and provide one output then goes to the input of the 2nd hidden layer.
- Output layer:- finally, the hidden layer output is feed to the Output layer. Typically there is only one output node for classification problem. The output layer provides the final output value.

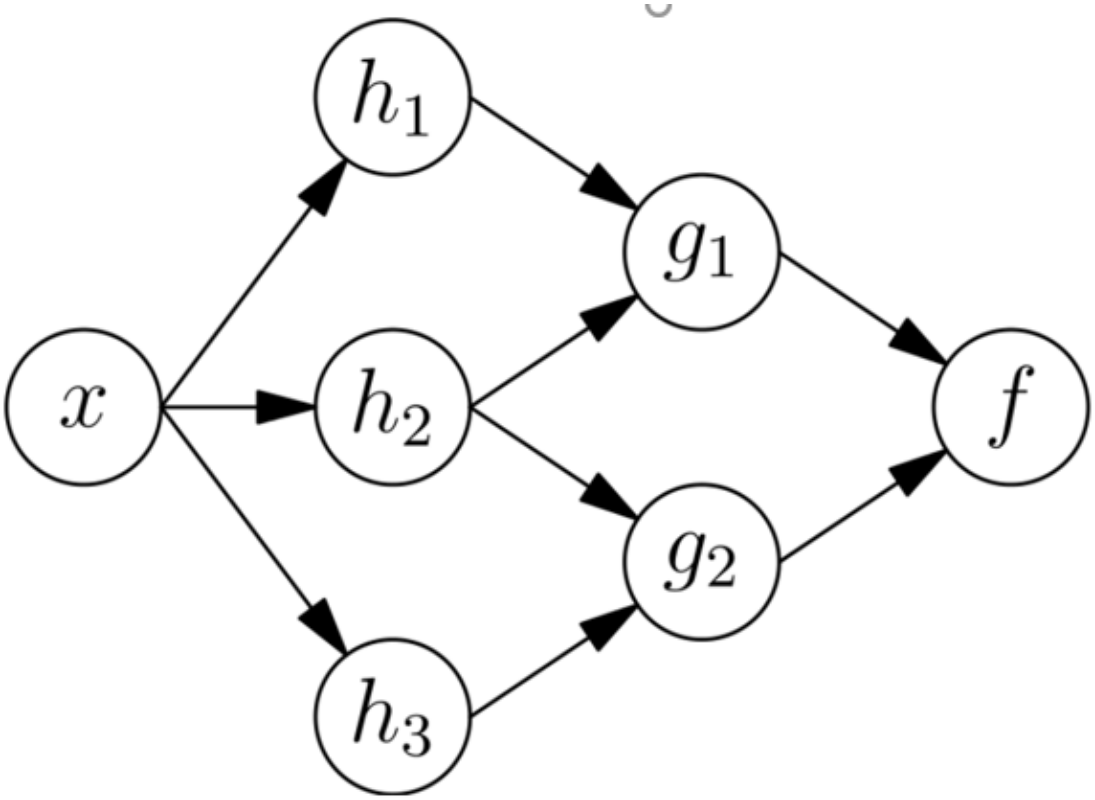


Figure 5 -Neural Network for Deep Learning Neural Network

The overfitting and Computational time are two specific problems of DNN. There are many applications of Deep learning neural network algorithms like Automatic Speech Recognition, Image recognition. Medical image analysis, Financial fraud detection.s

3.3 Model Evaluation

3.3.1 Confusion Matrix and Prediction performance Metrics

It is also known as an error matrix. Table form representation is easy for the confusion matrix. By this table, the percentage of correctly predicted value can be shown easily. As in our scenario, the final output has four activity labels, so four rows and four columns can show a comparison of each activity label with other activity labels.

Terminologies used are

1. **TP** = True positive
2. **TN** = True negative
3. **FP** = False positive
4. **FN** = False negative

Confusion Matrix	**Predicted Category**	
Actual Category	**Positive Labels**	**Negative Labels**
Positive Labels	True Positive (T.P.)	False Negative (F.N.)
Negative Labels	False Positive (F.P.)	True Negative (T.N.)

Sensitivity (Se) or recall (Rc):.The ratio of correctly predicted positive observations(T.P.) and all actual positive observations (TP+FN).

$$\text{Sensitivity } (S_e) = \frac{TP}{TP + FN}$$

Specificity (Sp) : The ratio of correctly predicted negative observations (T.N.) and all actual negative observations (FP+TN).

$$\text{Specificity } (S_p) = \frac{TN}{FP + TN}$$

Positive predictive value (PPV) or precision (Pr): The ratio of correctly predicted positive observations(T.P.)with all observations predicted as positive observations(TP+FP). PPV is directly proportional to the prevalence of the outcome of interest.

$$\text{Positive Predictive Value (PPV) or Precision } (P_r) = \frac{TP}{TP + FP}$$

$$= \frac{\text{Sensitivity} \times \text{Prevalence}}{\text{Sensitivity x Prevalence} + (1 - \text{Specificity}) \text{ x } (1 - \text{Prevalence})}$$

Negative predictive value (NPV): The ratio of correctly predicted negative observations(T.N.) and all observations predicted as negative observation(TN+FN). In the case of rare outcomes, NPV is typically high.

$$\text{Negative Predictive Value (NPV)} = \frac{TN}{FN + TN}$$

Positive likelihood ratio (PLR): The ratio of **t**he probability of an actual positive observation predicted as positive observation and the probability to not have a positive observation but predicted as a positive observation. The larger value of PLR (>1), suggests a good prediction performance for the algorithm.

$$\text{Positive likelihood ratio (PLR)} = \frac{\text{Sensitivity}}{1 - \text{Specificity}}$$

Negative likelihood ratio (NLR): The ratio of the probability of an actual positive observation predicted as negative observation and the probability to not have a positive observation but predicted as a negative observation. The smaller value of NLR (near to 0), suggests a good prediction performance for the algorithm.

$$\text{Negative likelihood ratio (NLR)} = \frac{\text{Specificity}}{1 - \text{Sensitivity}}$$

Overall misclassification rate: The ratio of correctly predicted and the total observations (i.e. the sum of all observations).

$$\text{Overall misclassification rate} = \frac{FP + FN}{TP + FN + FP + TN}$$

F1 score: The weighted harmonic mean or average of precision (also known as Positive Predictive Value (PPV)) and recall (also known as sensitivity). In case of an uneven class distribution, as in our case majority of the cohort is comprised of non-overdose individuals, the F1 Score is more useful than the Overall misclassification rate because it takes both false positives and negatives into consideration. F1score closer to 1 is desirable to have.

$$\text{F1 Score} = 2 \text{ X } \frac{Pr \text{ X } Rc}{Pr + Rc}$$

C-statistic: When we plot sensitivity against F.P. (False Positive or 1-specificity) for every potential cut-off probability threshold, we get a graph called ROC Curve of an algorithm. The area under the ROC curve (receiver operating characteristics curve) is known as C-statistics. For rare outcomes or imbalanced data, the comparisons of C-statistics can be misleading because C-statistics does not take care of prevalence; that is, it does not take care of information about the pre-test probability of the outcome.

Precision-recall curves: When a graph is plotted taking recall that is sensitivity on the x-axis and precision or PPV on the y-axis, the curve is known as a precision-recall curve. For an algorithm with better performance, the curve is closer to the upper right corner, which is corresponding to 100% precision and 100% recall.

Number needed to evaluate (NNE): The reciprocal of the PPV value is NNE. The number of observations required to assess or screen a positive result is called NNE. A PPV of 10% is the same as NNE of 10.

$$\text{Negative needed to evaluate (NNE)} = \frac{1}{\text{PPV}}$$

3.3.2 Different Hypothesis Testing

- **Normal Test or Z-Test:** For this test, the observation sample is assumed to be distributed normally. By using population parameters like population mean and population standard deviation, a z-score is calculated. This z-score is then used to validate a hypothesis that the test sample drawn belongs to the same population.

  Null and Alternate Hypothesis:

  Null: Null Hypothesis is a prevailing belief about the population. Example: Sample mean and the population mean are the same.

  Alternate: Alternate Hypothesis is a claim opposing Null Hypothesis. Example: Sample mean, and the population mean are not the same.

  z-statistics are calculated as below:

  $z = (x - \mu) / (\sigma / \sqrt{n})$, where

  x= sample mean

  μ = population mean

  $\sigma / \sqrt{n}$ = population standard deviation

  - If the test statistic (z-statistic) is lower than the critical value(Zc value), then we accept the hypothesis or else reject the hypothesis.

- **One-Tailed Hypothesis Tests (One – Tailed T tests):** This is often referred to as directional and unilateral experiments since results are only measured in one direction. The entire significant level percentage reaches the extreme end of one tail of the distribution when a One-tailed test is conducted. It can be a left-tailed or right-tailed.
- **2-Tailed T-Test:** For statistics, a two-tail test is a system where a distribution's critical area is dual-sided and checks whether a sample meets or is below a certain set of values. The approach is used to test a null-hypothesis and test for statistical significance. When the test sample falls in one of the critical areas, the alternative hypothesis rather than the null hypothesis is accepted. The two-tailed test is named after checking both tails of a normal distribution, although the test can be used on other non-normal distributions tool.
- **$\chi 2$ test:** A chi-square ($\chi 2$) is a test that measures how expectations compare with the actual data observed (or results of the model). The data used to measure a chi-square statistic must be arbitrary, incomplete, mutually exclusive, derived from independent variables, and taken from a sample that is large enough.

3.4 Summary

The chapter outlines the method used in the Human Activity Recognition classification problem for data collection, Data cleaning, Data preparation, Machine Learning Algorithm, and evaluation method to select the best classification algorithm for Human activity recognition problems for smartphone data and sensor data.

CHAPTER 4

4 ANALYSIS

4.1 Introduction

We will be using publicly available UCI data set for this study. This data is no derived from our own experiment, and we used a publicly accessible UCI smartphone-based and wearable sensor-based dataset. So it becomes essential to check the dataset for all the data quality issues. Otherwise, data quality issues can provide irrelevant results, so to prepare data for building models, we need to perform data cleaning. In this section, we will discuss all the data cleaning, data preparation steps for model building, and hyperparameter optimization of the machine learning algorithm.

We used two datasets for this study so that we will discuss all the steps for both the data set.

4.2 Datasets

4.2.1 Wearable Sensor-Based dataset

This data set is obtained from a wearable sensor-based experiment. In this experiment, 14 older people who were above 65 asked for performing some common activities like Walking, Getting off the chair, Lying, Sitting on the chair, Sitting on the bed. This wearable sensor was kept on their clothing. These participants stayed in two-room during this study. One room has four antennae, while the other has three. ("UCI Machine Learning Repository: Activity recognition with healthy older people using a batteryless wearable sensor Data Set," n.d.)

This dataset is available in two folder S1 and S2. Folder S1_dataset has 60 files, and Folder S2 has 30 files. These files represent sensor data for individual trials performed by a subject wearing a wireless, batteryless sensor.

In this dataset, as per the Activity label, below four activities were performed.

- Sit on bed
- Sit on chair
- Lying
- Ambulating like Getting off the bed, Getting off the chair

In these files, participants' details were not provided, but room and gender can be identified by the file name. The last character of the file name represents the gender, while the second character of the file name represents room number. For example:- filename d2p01f represents female, and room two and d1p43M represent room One and male.

Folder S1_Dataset and S2_dataset have multiple text files. We loaded all the records from all the files in python and checked total records in dataset. There are 52482 records in S1 folder

files and 22646 records in the S2 folder. There are 11 columns in this Dataset including Room and Gender.

4.2.2 Smartphone Sensor-Based dataset

This data is collected from an experiment where 30 peoples whose age was between 19 to 49 performed every day six tasks like Lying, Sitting, Walking, Standing, Walking Up Stairs, and Waking downstairs. and with a waist-mounted smartphone.("Index of /ml/machine-learning-databases/00427," n.d.)

When the user performed these activities and some other transition activities like Stand to sit. Sit to lie and Lie to Sit, three axial linear acceleration, and three axial angular velocities at a constant rate of 50 Hz with the help of smartphone integrated sensors are obtained. This dataset is divided into two parts train and test.

We have used a preprocessed version of the UCI data dataset instead of the raw dataset in this Both train and test data is loaded in one final data frame. This dataset has 10299 rows and 563 columns. Out of 562 columns, 561 columns were float type, and 1 was an activity label.

4.3 Data preparation & Data check for Wearable Sensor-Based Dataset

4.3.1 Dataset information

Below is the statistical data for Wearable Sensor based dataset.

Features	count	mean	std	min	25%	50%	75%	max
Time_Sec	75128	299.08	257.5	0	121.25	250.72	402.5	1739
Acc_Front_axis	75128	0.7142	0.4046	-0.748	0.3424	0.6824	1.1045	1.503
Acc_vert_axis	75128	0.3452	0.419	-0.554	-0.0023	0.2159	0.8589	2.03
Acc_Lat_axis	75128	-0.2175	0.4382	-1.336	-0.1847	-0.0707	0.0319	1.218
id_Antenna	75128	2.4096	1.1022	1	1	3	3	4
RSSI	75128	-58.277	5.1741	-72	-62	-58	-56	-38.5
Phase	75128	3.1573	2.1823	0	1.0324	2.7673	5.3586	6.282
Freq	75128	922.67	1.6791	920.25	921.25	922.75	924.25	925.8
Activity Label	75128	2.5284	0.8643	1	2	3	3	4

Table 4 -Statistical data for Wearable Sensor Dataset

4.3.2 Missing Value Check

We checked if there is any missing value in any of the columns.

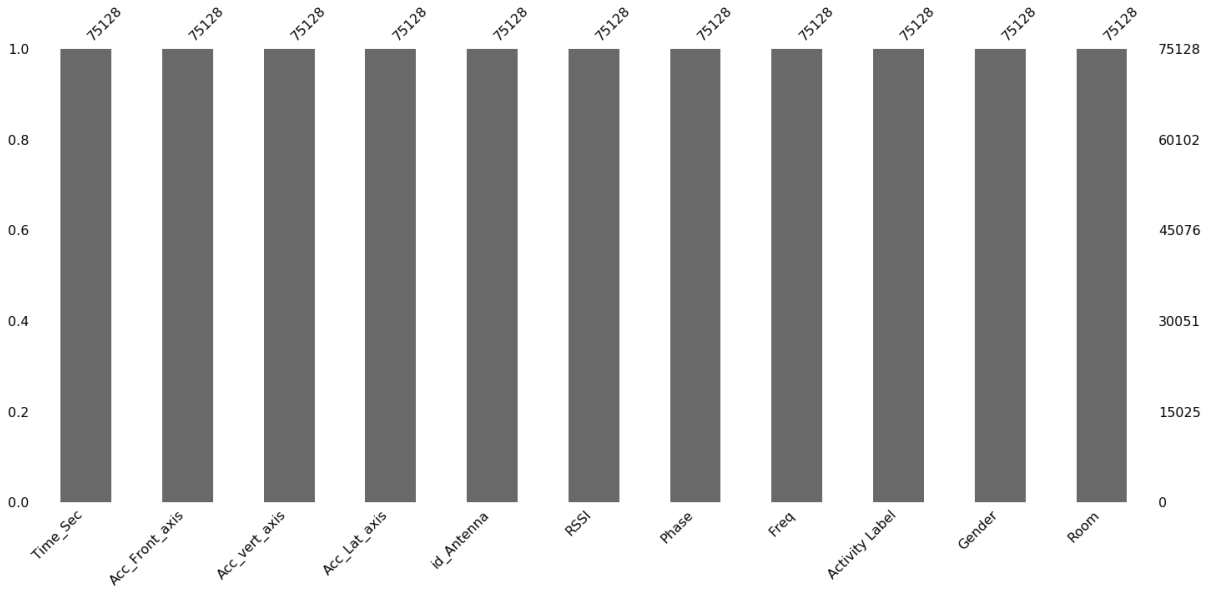


Figure 6-Missing Value check for Wearable Sensor Dataset

There is no missing value in any of these columns, so we don't need missing value imputation for this dataset.

4.3.3 Data Distribution Check

We checked data distribution for each activity, room, gender for our analysis purpose.

The total number of the record for each activity:- We checked the data distribution for each activity. Below are the details

Activity	**No of Record**	**Percentage of Record**
Lying	51520	68.6
Sit on Chair	4911	6.5
Sit on bed	16406	21.8
Ambulating	2291	3
Total	75128	100%

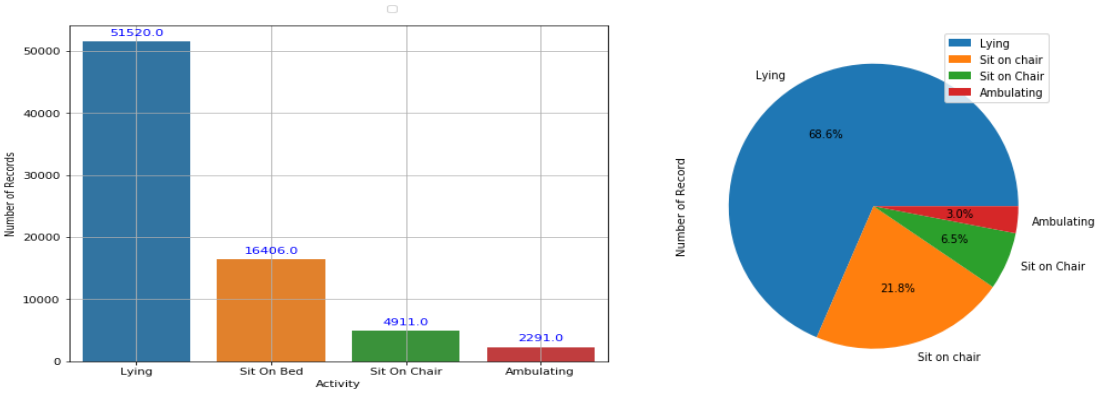


Figure 7 -Total number for the record for each activity

There are more records, a total of 51520 records for Lying activity than a total of the other three activities. Almost 68.6% of data is for Lying activity.

Total number of the record for gender:-

We checked the data distribution for each gender in the complete dataset.

Activity	No of Record	Percentage of Record
Male	43282	58.34
Female	31300	41.66
Total	74582	100%

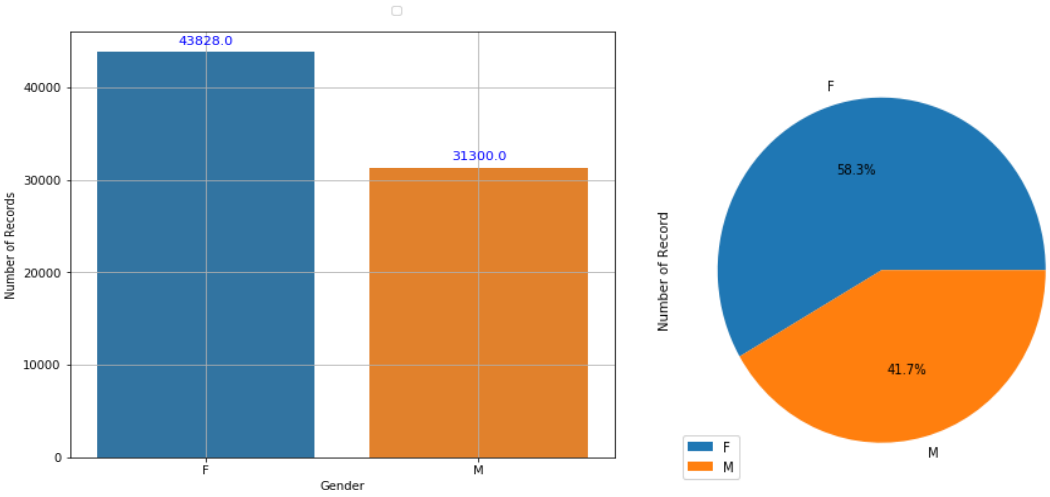


Figure 8 -Total number for the record for each gender

There is more record for females than males.

The total number of the record for each room

There were two rooms used in this experiment. So we checked the data distribution based upon the room.

Room	No of Record	Percentage of Record
Room1	52482	69.9
Room2	22646	30.1
Total	74582	100%

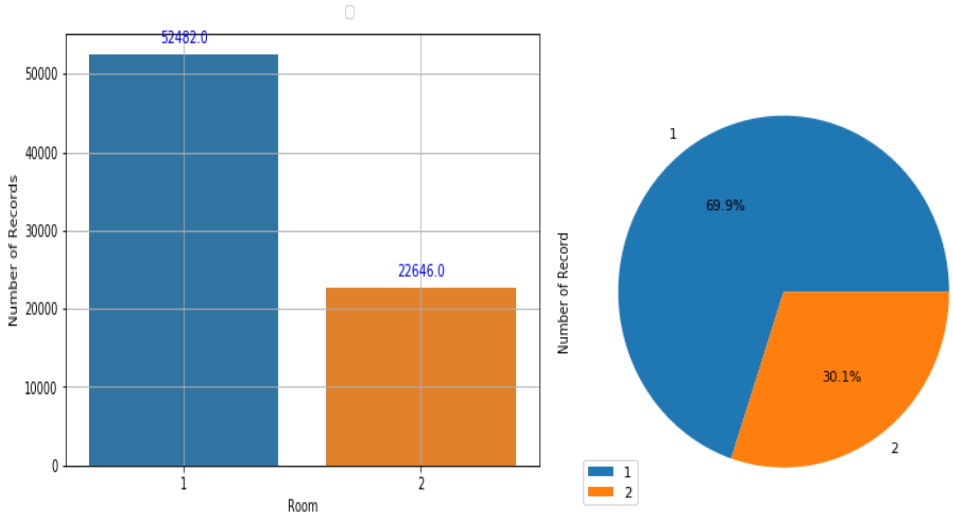
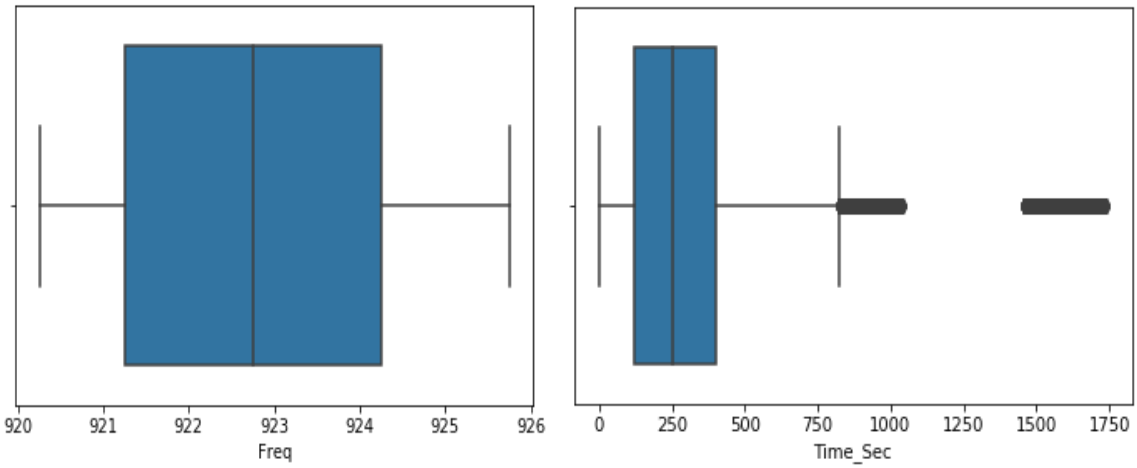


Figure 9 -Total number for the record for each room

There are approx. 70% of data from room1, and 30% from room2.

Distribution of all the columns and Outlier Check :-



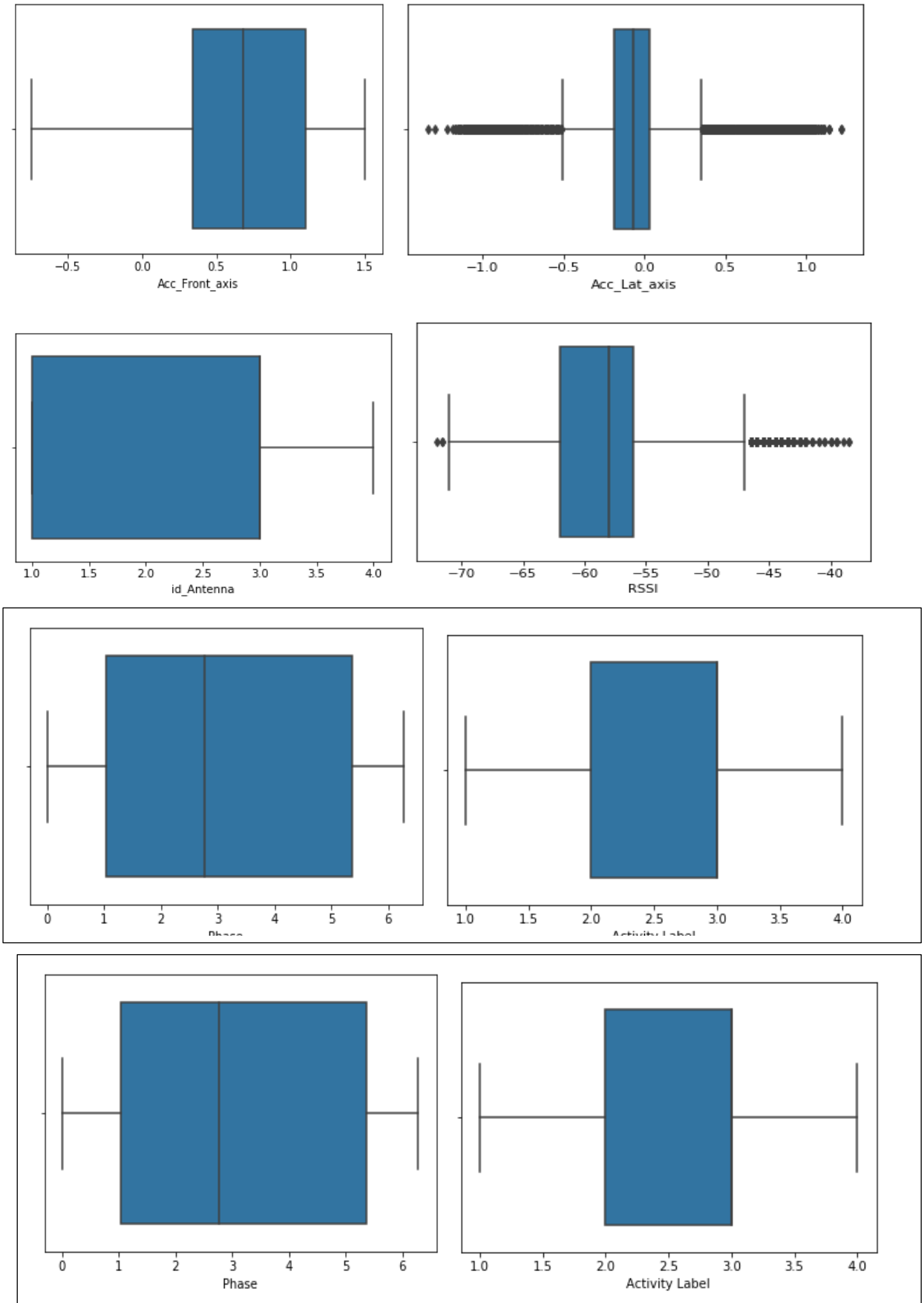


Figure 10 –Box plot for all Numeric Columns

We plotted the box plot for all the numeric columns to check if there is an outlier that needs to be removed as an outlier can impact results significantly. Still, there is no outlier treatment required for this dataset.

4.4 Data preparation & Data check for Smartphone Sensor-Based Dataset

4.4.1 Dataset information

In this dataset, there is a total of 10299 records for six activities. In this dataset, there are 561 numeric columns and three categorical data, which are Data, Subject, and Output.

4.4.2 Missing Value Check

We checked if there is any missing value in any of the columns. But no missing value found in any of the columns. So we don't need missing value imputation for this dataset.

4.4.3 Data Distribution Check

We checked data distribution for each activity, room, gender for our analysis purpose.

The total number of the record for each activity: We checked the data distribution for each activity. This dataset has a good balance of the number of records for each activity comparative to sensor-based data. In this dataset, each activity has more than 13.5 % record of the total dataset, whereas, in wearable sensor-based data, lying has more than 68% of the total record.

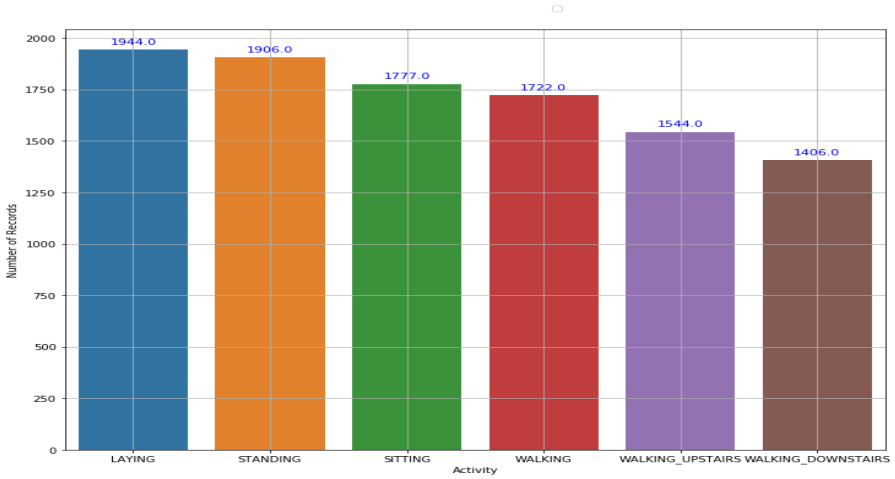


Figure 11 - Total Number of record for each activity

Activity	No of Record	Percentage of Record
Laying	1944	18.9
Standing	1906	18.5
Sitting	1777	17.3
Walking	1722	16.7
Walking Upstairs	1544	15
Walking Downstairs	1406	13.7
Total	10299	100%

Table 5 –Number of records for each activity in smartphone-based data

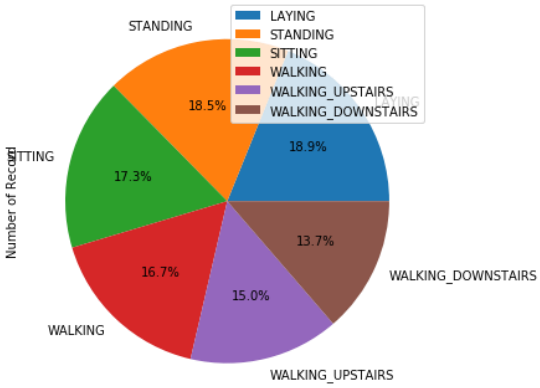


Figure 12 – Number of records for each activity in Smartphone-based sensor dataset

<u>Activity performed for each user</u>:

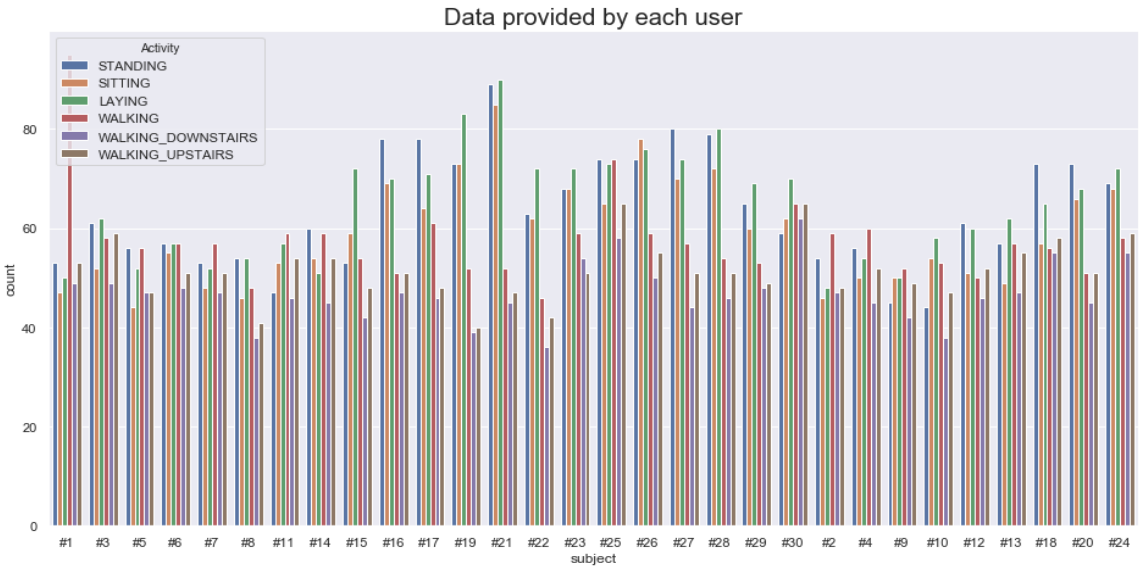


Figure 11 –Activity Performed by each user in smartphone dataset

4.5 Exploratory Data Analysis

Exploratory data analysis is done to understand the dataset. We performed below data analysis for wearable sensor-based datasets and smartphone-based dataset.

4.5.1 Wearable Sensor-Based Dataset

4.5.1.1 Correlation Between all the columns and output

Heatmap is plotted to understand if there is any correlation between all the input variables and which variable has the highest correlation with the output variable (Activity Label).

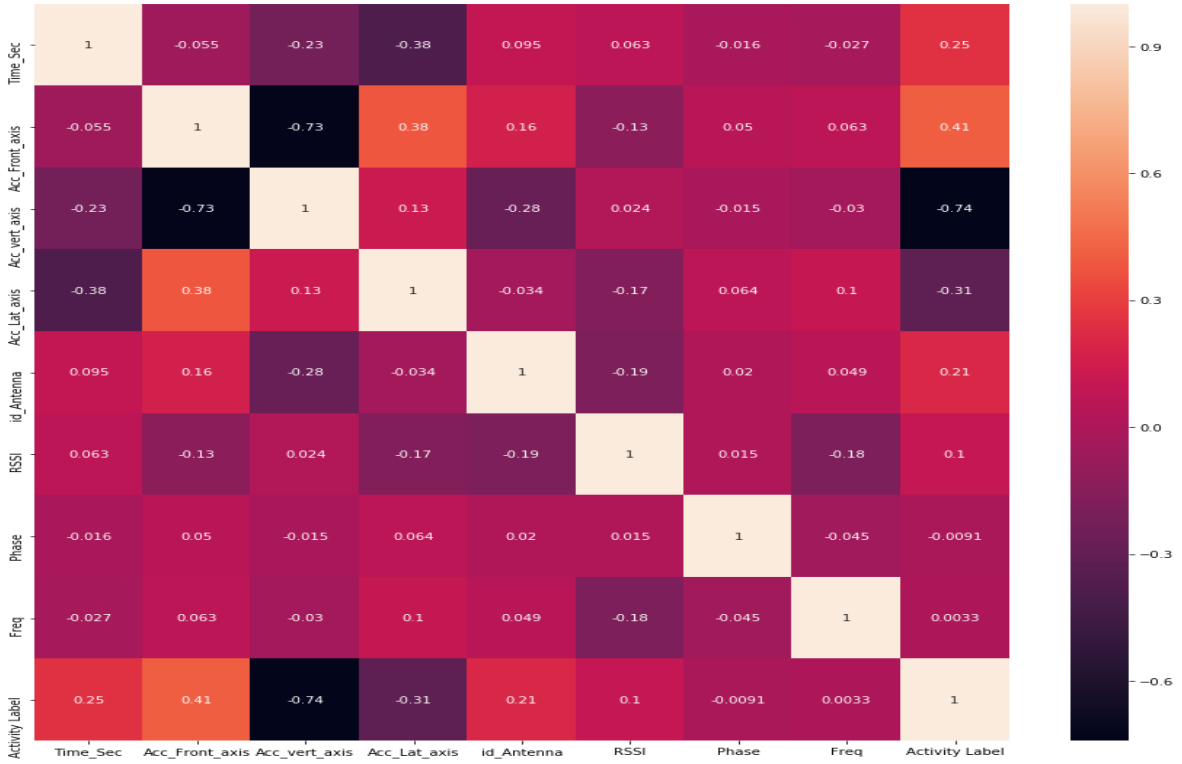


Figure 12–Correlation Graph between all the fields for Sensor-Based Dataset

As per the above correlation graph, the Activity label is highly correlated with acceleration in the vertical axis, and frequency and phase have the lowest correlation.

4.5.1.2 Mean Distribution of Acceleration in Vertical, Lateral and Front axis

We plotted the mean value of acceleration in the vertical, lateral, and frontal axis for each activity.

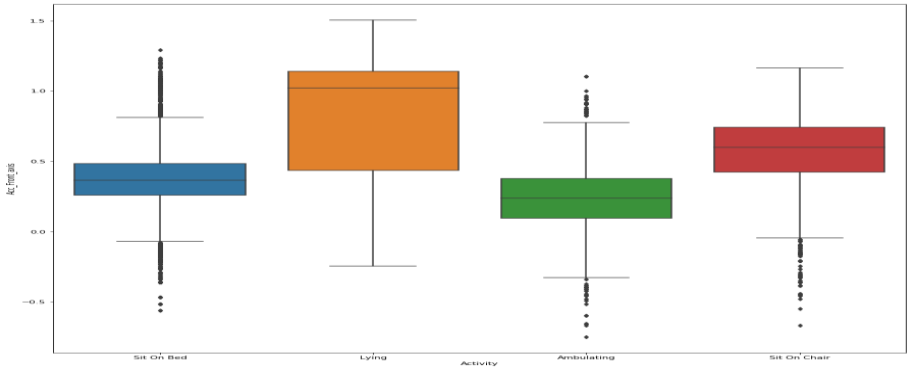


Figure 13–Distributions of Acc_frontal_axis for each activity

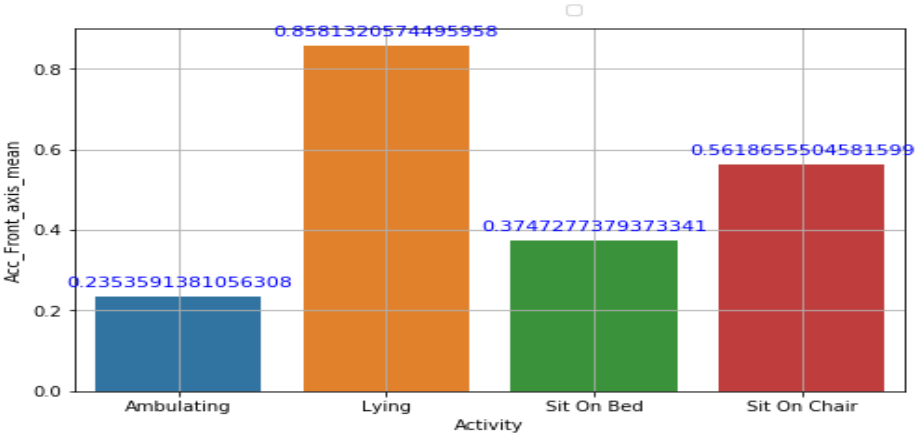


Figure 14 –Acc_frontal_axis Mean for each activity

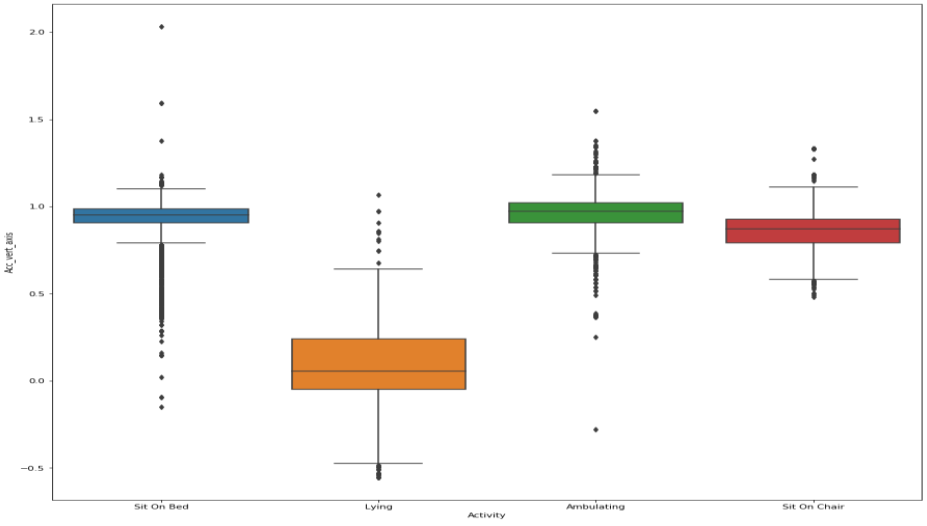


Figure 15 –Distributions of Acc_Vertical_axis for each activity

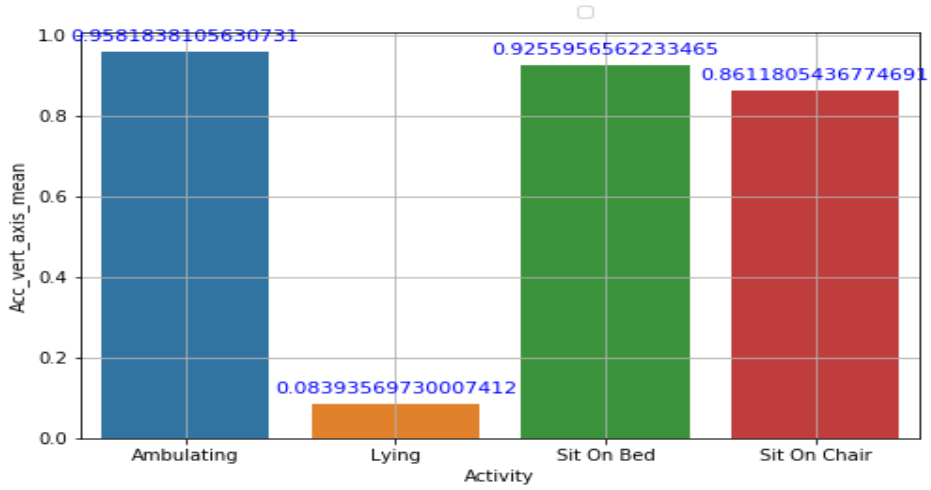


Figure 16 –Acc_frontal_Vertical Mean for each activity

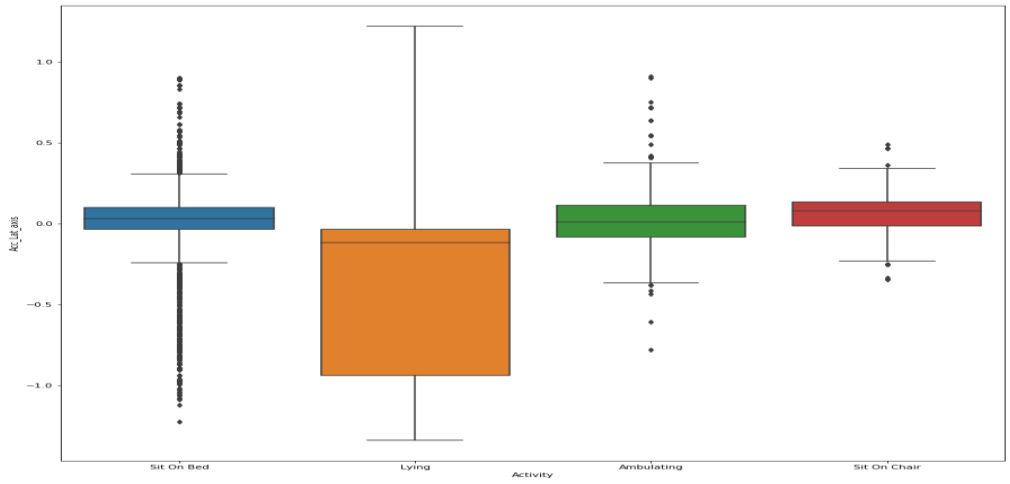


Figure 17 –Distributions of Acc_Lateral_axis for each activity

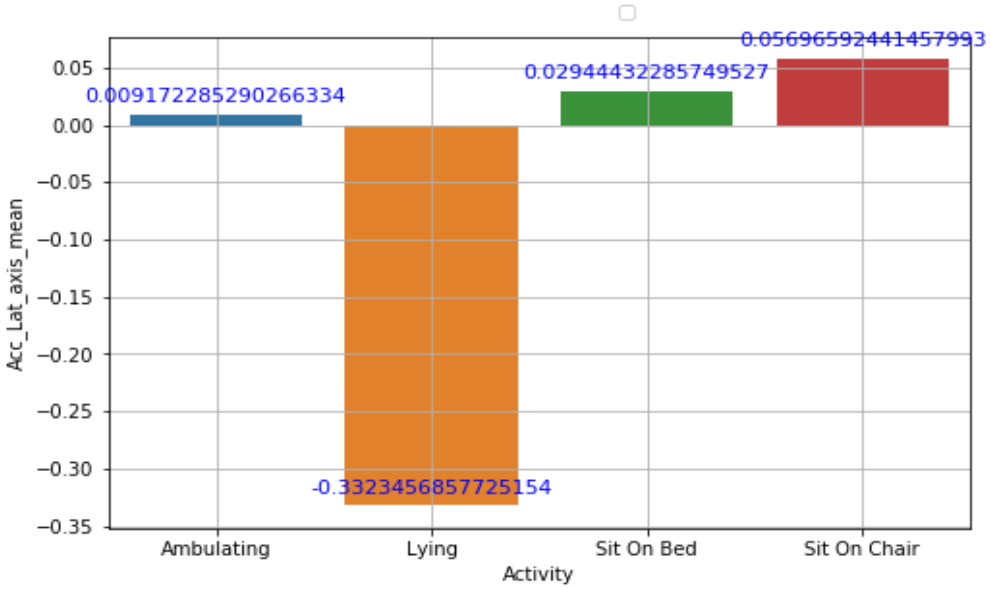


Figure 18 –Acc_Lateral_axis Mean for each activity

Observations:-

From all the above graph it is concluded that:

- Acceleration in the Lateral axis is negative for activity Lying but positive for other activity.
- Acceleration in the vertical axis is the lowest for activity lying if Acc_vert_axis is more than 0.5 only for Sitting and Ambulating.
- Acceleration in the frontal axis is highest for activity lying.

4.5.1.3 Distribution of Record as per Activity and Room

As per this graph, record count is very high for room1 and activity Lying.

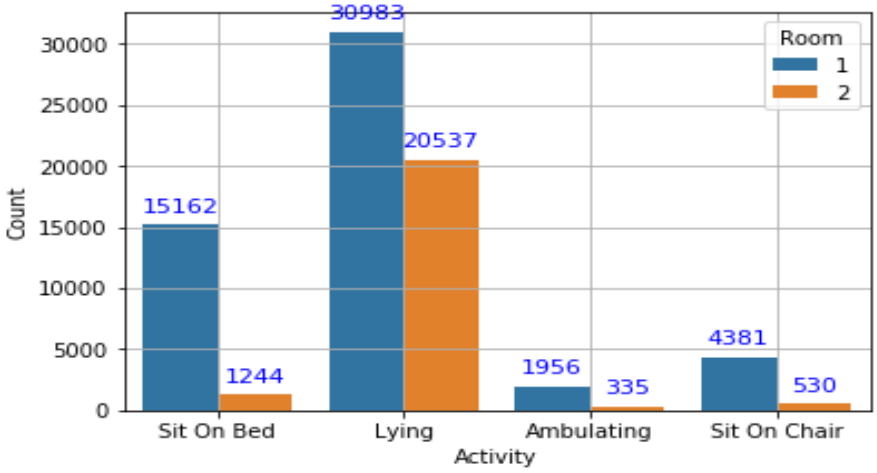


Figure 19 –Record count for each activity and each room

Activity	Room	No of Record
Sit on Bed	Room 1	15162
Sit on Bed	Room 2	1244
Lying	Room 1	30983
Lying	Room 2	20537
Ambulating	Room 1	1956
Ambulating	Room 2	335
Sit on Chair	Room 1	4381
Sit on Chair	Room 2	530
	Total Record	75128

Table 6 - Record count for each activity and each room

4.5.1.4 Distribution of Record as per Activity and gender

As per this graph, record count is highest for gender and activity Lying

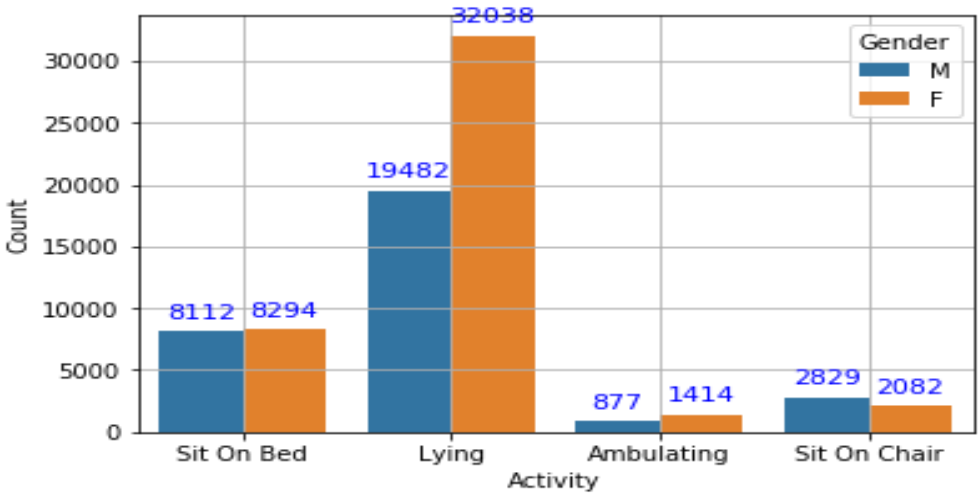


Figure 20 –Record count for each activity and each gender

Gender	Activity	No of Record
Male	Sit on Bed	8112
	Lying	19482
	Sit on Chair	2829
	Ambulating	877
Female	Sit on Bed	8294
	Lying	32038
	Sit on Chair	2082
	Ambulating	1414
	Total Record	75128

Table 7 –Record count for each activity and each gender

4.5.1.4 Relationship between all the variables

It shows the relationship between all the columns of sensor-based dataset for all the activity.

Figure 21–Relationship between all the variables for sensor-based dataset.

4.5.2 Smartphone Sensor-Based Dataset

4.5.2.1 Feature Distribution of all the columns in Dataset:

- **Box plot of Angle between X-Axis and gravity mean**

We plotted distributions of Angle between X-axis and gravity mean for each activity and concluded below points:-

- The angle between X-axis and gravity mean is higher than 0 only in case of Activity Laying
- Activity Laying can be easily predicted only based upon angular between X axis and gravity mean in this data set.

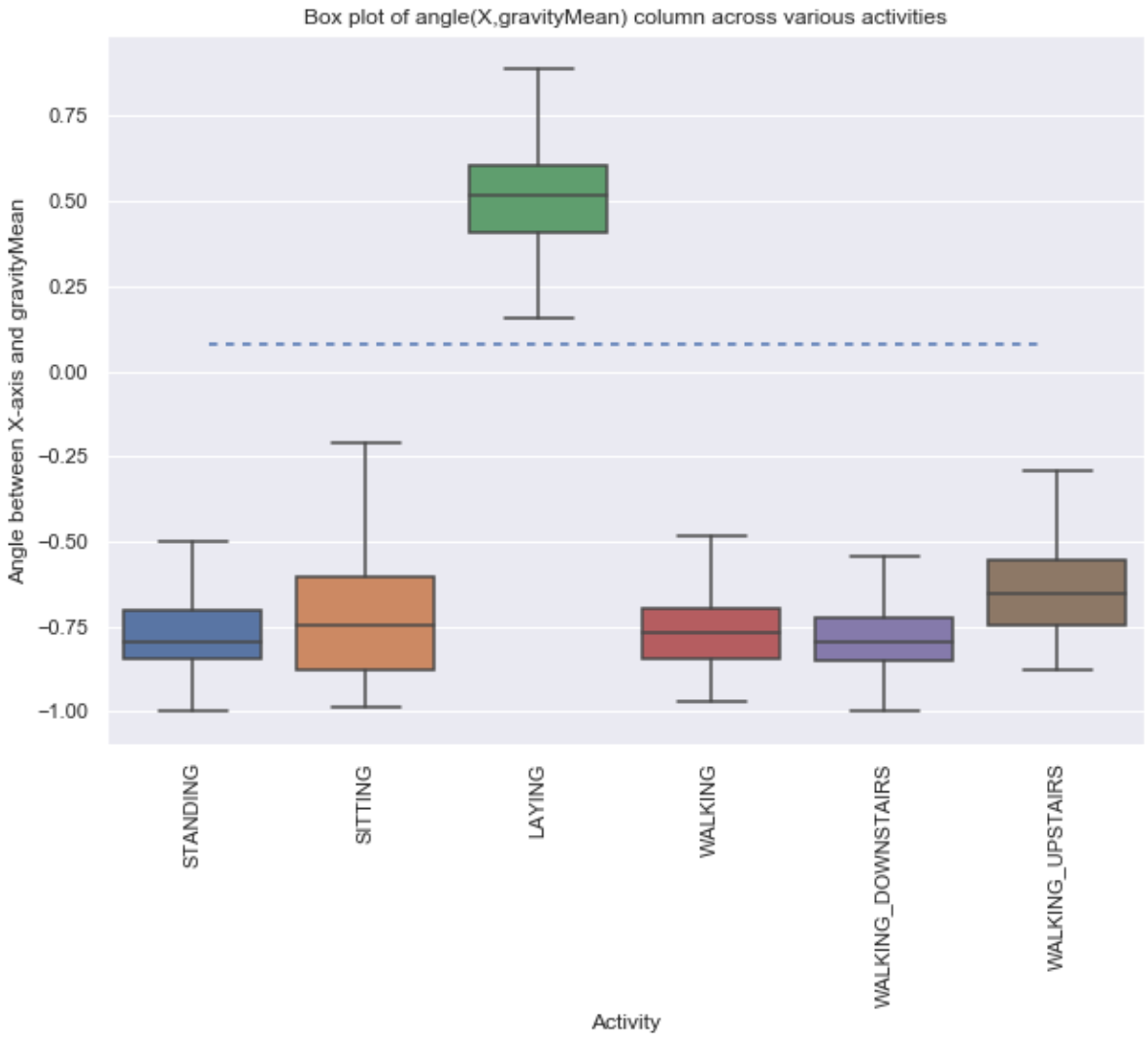


Figure 22 –Box plot of Angle between X-axis and gravity mean for all the activity

- **Box plot of Body Acceleration Magnitude Mean-**

We plotted distributions of body acceleration magnitude mean for each activity and concluded below points:-

- For a static activity like Standing, Sitting and lying acceleration magnitude mean is negative and less than -0.8
- Acc Mean is highest for Activity Walking downstairs and almost the same as walking up staircase and down staircase.

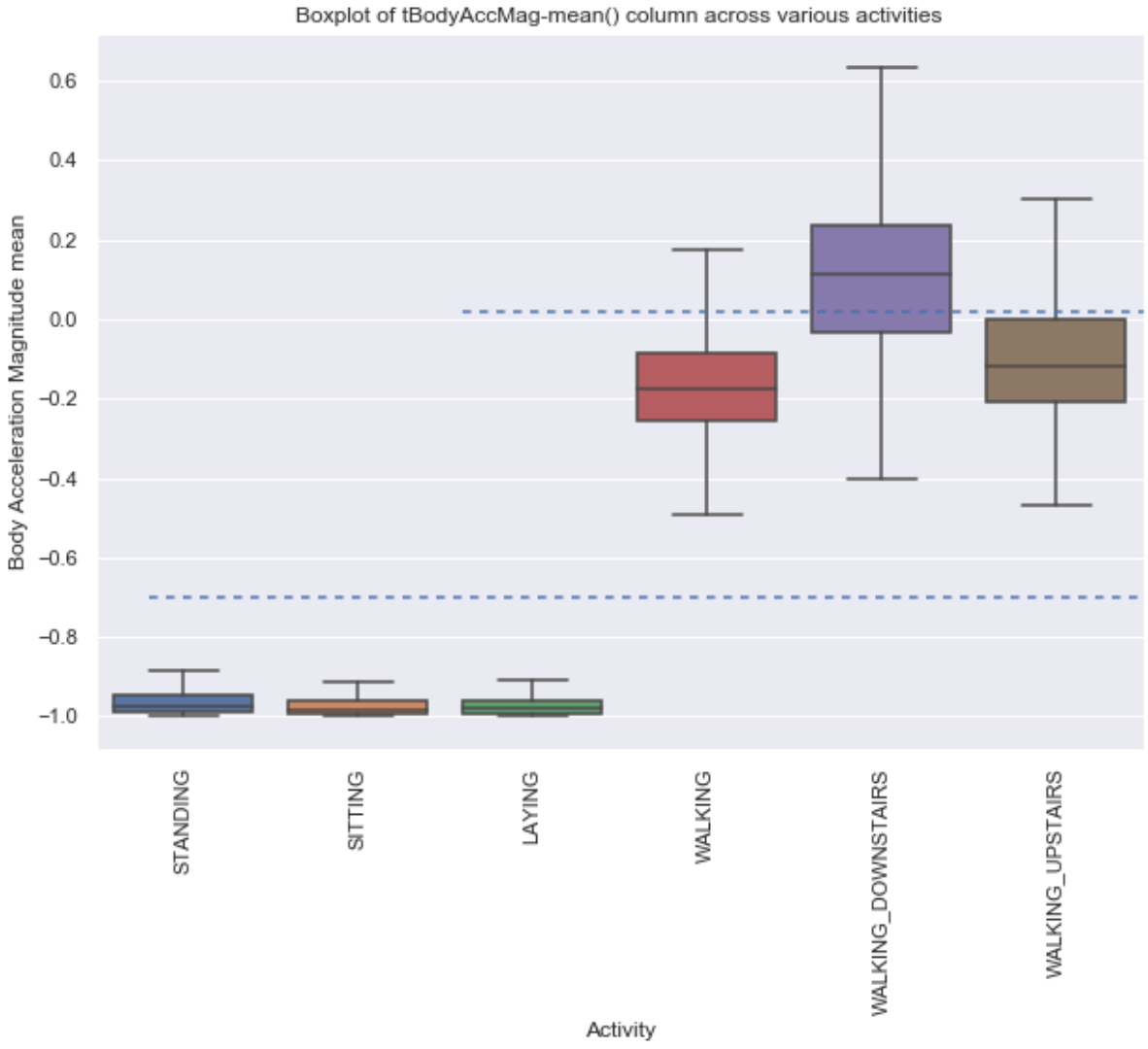


Figure 23 –Box plot of Body Acceleration Magnitude Mean for each activity

.

- **Box plot of Angle Y and gravity mean for activity**

We plotted distributions of angle y and gravity means for each activity and concluded below points:-

- Angle between Y axis and gravity means is lowest for Activity Laying and high for dynamic acidity like walking, walking_downstairs, and upstairs.

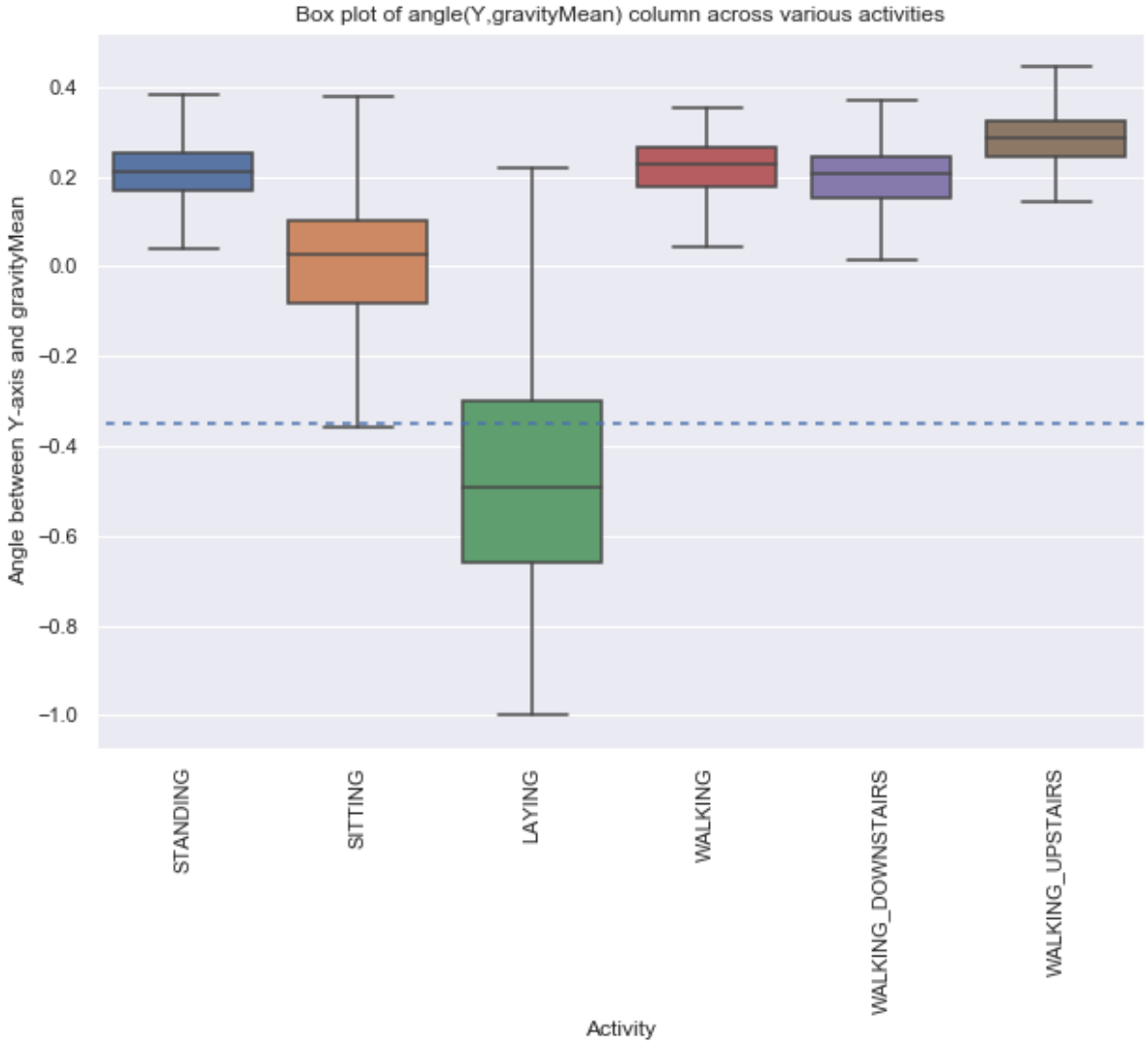


Figure 24–Box plot of Angle between Y-axis and gravity mean for each Activity

- The time duration for walking Upstairs and walking downstairs

  It is concluded that almost all the participants took less time in walking downstairs comparative to upstairs. This is the total duration spent by each participant in walking upstairs and walking downstairs. Almost every person took more time in walking upstairs than downstairs. Some person took more time in comparison to others but we are assuming that can be related to their physical condition and health, for example, a healthy and young person walk on upstairs fast then an older person.

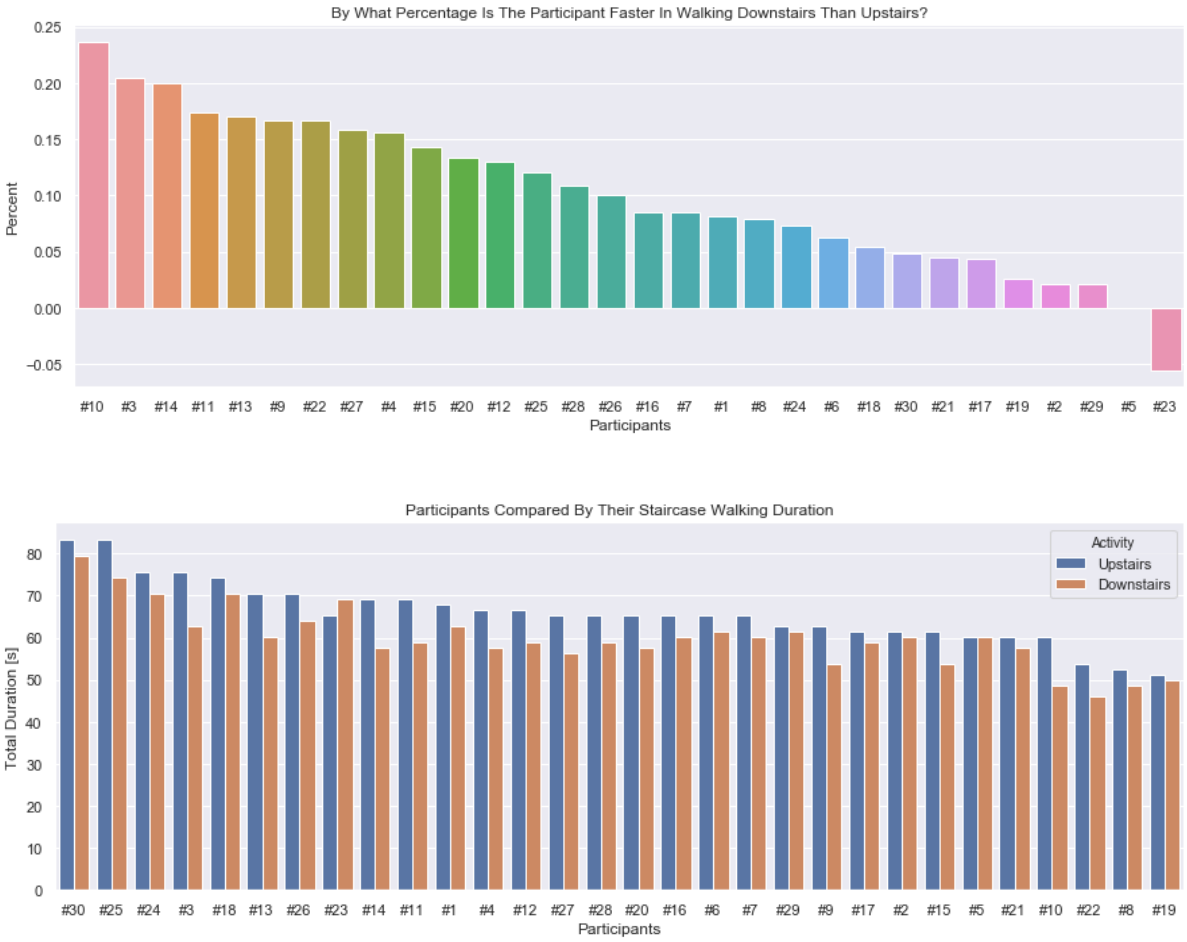


Figure 25 -Total duration spent by each user in walking upstairs and walking downstairs.

4.6 Data Preprocessing

4.6.1 Dummy Variable Creation

If there is any categorical value, that needs to be changed in the form of 0 and 1 before model building. In this dataset, there are two variable rooms and gender that are converted into the dummy variable before model processing. There were no categorical columns in Smartphone-based Dataset, so there is no dummy variable created for it.

4.6.2 Standardization of Dataset

Normalization converts data between 0 and 1. In this dataset, all the numeric variables are normalized before splitting between test and train. We used the below method to normalize the data.

$$x' = \frac{X - Xmin}{Xmax - Xmin}$$

Xmax is the maximum value of that column, and Xmin is the minimum value of those columns. So the maximum possible value in this scenario is one, and the minimum is 0.

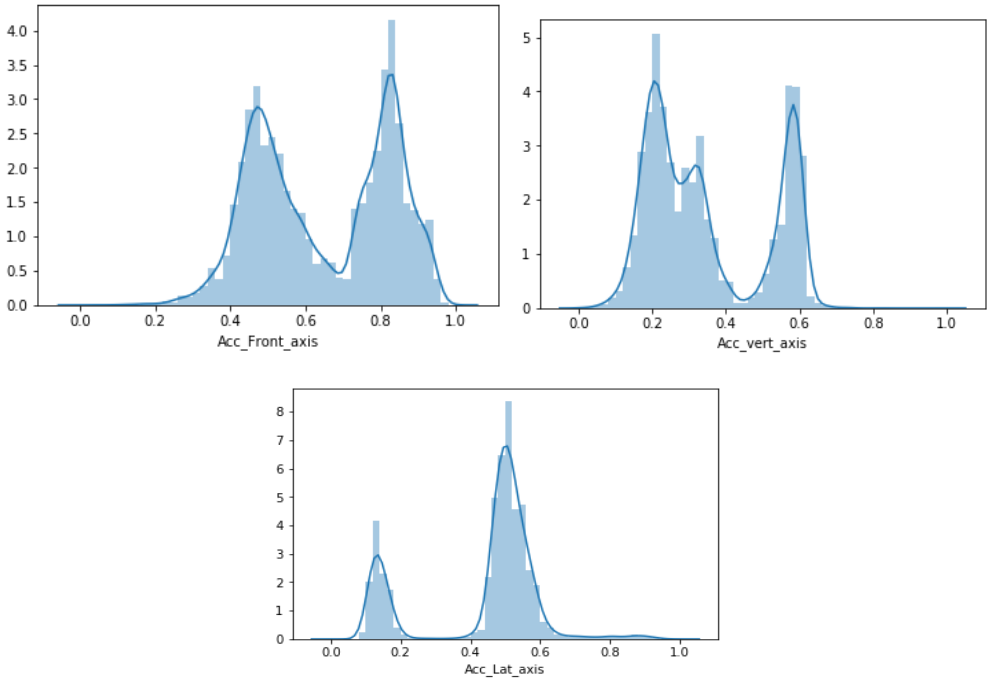


Figure 26-Standardization of the dataset

4.6.3 Splitting of original Dataset between test and train

For this study, the Complete Dataset is divided between two datasets. 70% of data is used as training data for the model, and the remaining 30% of data is used to test the model.

The training dataset has 52589 records, and the test dataset has 22539 rows.

Training Data set (52589)	Test Data (22359)

Smartphone-based data was already divided into two files, test and train, so we used the same dataset for training and testing. We did not split the dataset.

4.6.4 PCA –Principal component Analysis

We plotted the correlation graph for all the fields for the smartphone sensor-based dataset and found that some of the variables have multicollinearity. To remove multicollinearity and reduce no of features, we used PCA approach for this dataset. We applied principal component for test and train data.

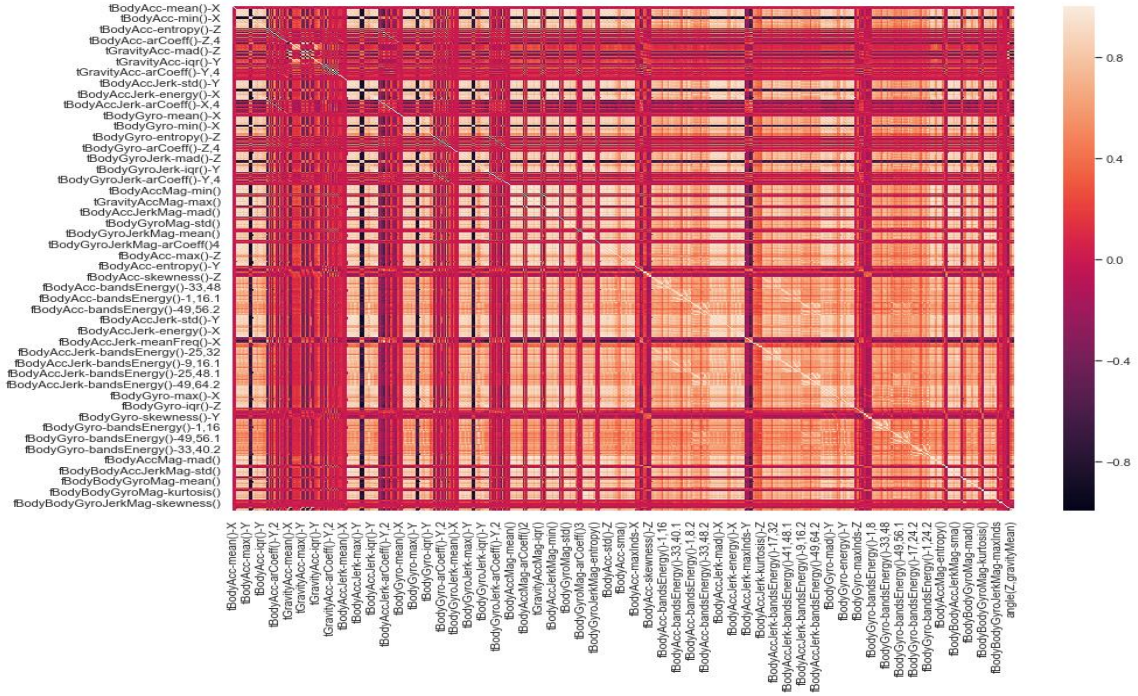


Figure 27 -Correlation Graph between all the columns.

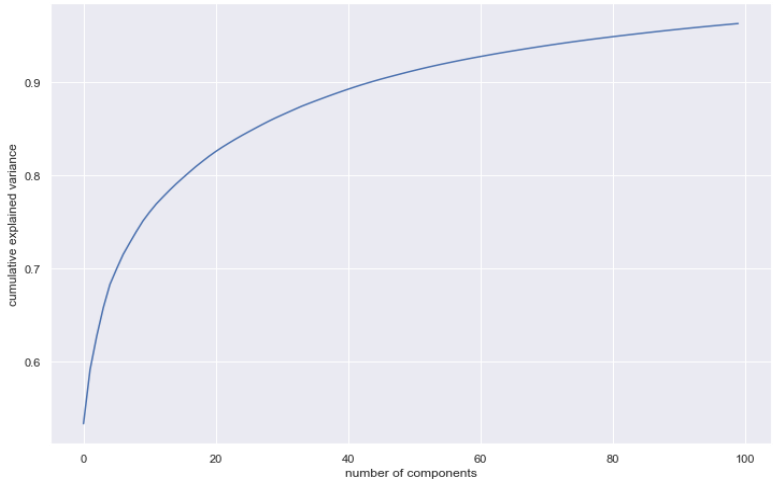


Figure 28 -No of PCAComponents and Cumulative Variance.

We plotted the graph between no of components and Cumulative variance and found that PCA Components 102 is able to show more than 95% variance of data, so took PCA components as 102 for model building.

We checked the correlation graph between all PCA components and found that those components have zero correlation.

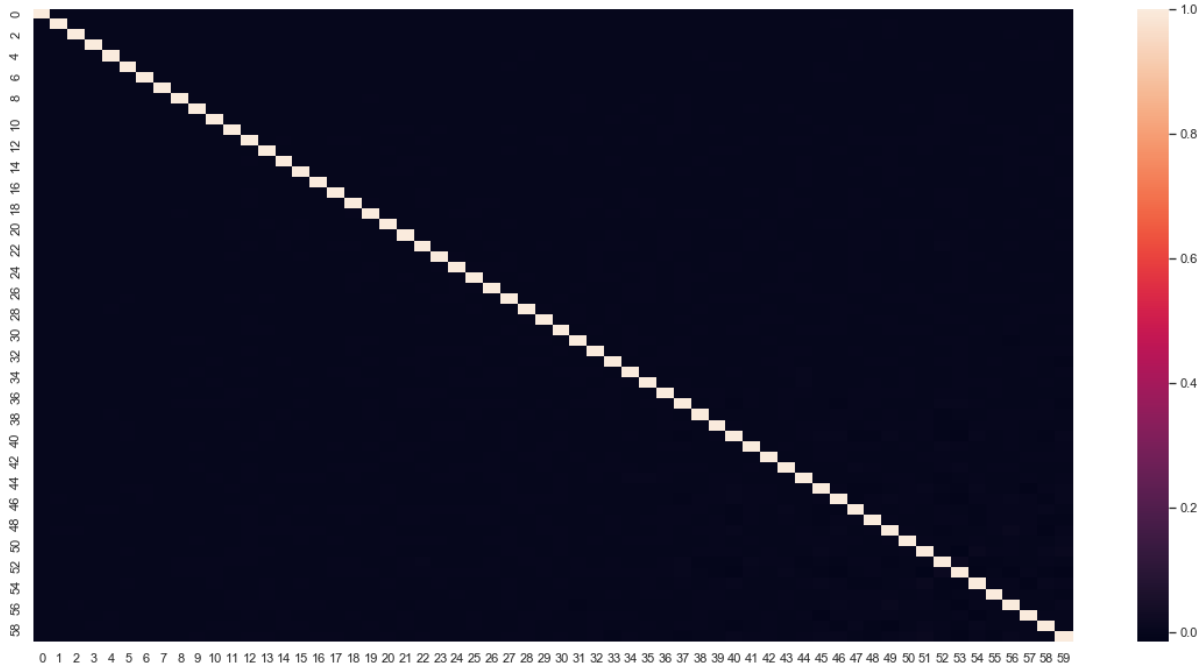


Figure 29-Correlation between PCA Variables.

Train_pca and test_pca is created after applying PCA for the train and test dataset.

4.7 Model Building and Hyper-Parameter Optimization- Wearable Sensor Data

We created a classification model by using six Machine learning algorithm (KNN, Logistics regression, Random Forest, SVM, GBM, and DNN) for smartphone sensor-based dataset and Wearable sensor-based dataset.

4.7.1 GridSearchCV - Hyper-Parameter Tuning

GridSearchCV is provided by scikit-learn. It is the process of tuning the hyperparameter so that optimal values for a given model can be defined. The output of the whole model depends on the defined hyperparameter values. It exhaustively generates predictor candidates from a dictionary of parameter values provided in the param_grid parameter.

GridSearchCV(estimator, param_grid, scoring=None, n_jobs=None, iid='deprecated', refit=True, cv=None, verbose=0, pre_dispatch='2*n_jobs', error_score=nan, return_train_score=False)

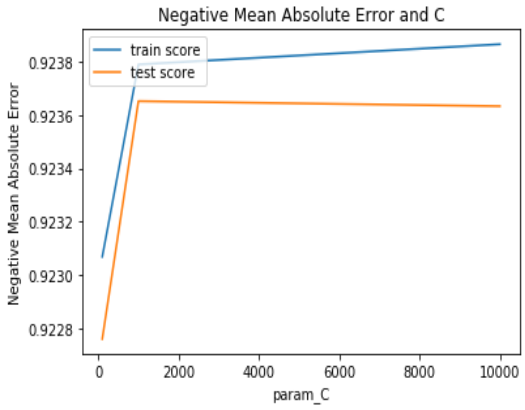
'estimator': It has the type of regressor we are using. Example: LogisticRegression() etc.
'param_grid': dictionary or list of dictionaries of parameter including penalty type and C values. L1 and L2 are two types of penalty parameters. L1 penalty uses the Lasso technique and produces sparse models. It causes many co-efficient to become zero and thus removing those features. L2 regularization adds an L2 penalty equal to the coefficient square. L2 does not generate sparse models, so all coefficients are reduced by the same factor. This approach is used by Ridge regression and SVMs. C values are provided to take full advantage of the randomization.
'cv' : cv is used to provide n-fold cross-validation
'scoring': We provide this value to evaluate the prediction on the test set.

4.7.2 Logistics Regression

Logistics regression is used to predict the activity label for smartphone data and sensor-based dataset. We used gridsearch cv to optimize hyperparameter.

Optimization of Hyper-Parameter: Penalty Parameter and Param C is optimized for the below range. Penalty=[11,12] ; C =[0.001,0.01,0.1,1,10,100,1000,10000] and cv= 5
Below accuracy is achieved for sensor-based datasets.

param_C	param_penalty	mean_test_score	mean_train_score
0.001	l1	0.6863	0.6863
0.001	l2	0.6863	0.6863
0.01	l1	0.6985	0.6984
0.01	l2	0.6863	0.6863
0.1	l1	0.6867	0.6865
0.1	l2	0.6896	0.6898
1	l1	0.8921	0.8921
1	l2	0.6879	0.6878
10	l1	0.9113	0.9119
10	l2	0.7045	0.7044
100	l1	0.9228	0.9230
100	l2	0.7798	0.7798
1000	l1	0.9234	0.9236
1000	l2	0.8750	0.8750
10000	l1	0.9234	0.9238
10000	l2	0.8984	0.8987



At C = 1000, the best score is 0.9234 at penalty L1.

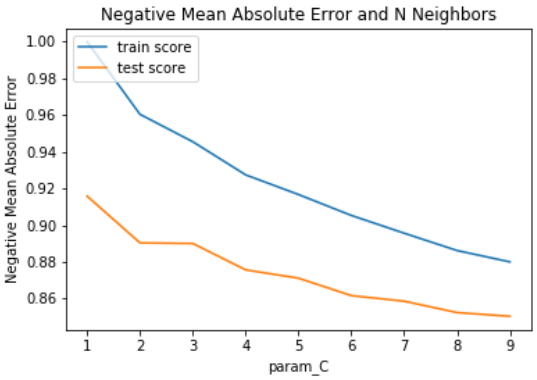
Figure 31-Hyper-Tunning Parameter of Logistics Regression

4.7.3 K-Nearest Neighbors (KNN)

For hyper-tunning KNN, we have again used GridSearchCV.

Optimization of Hyper-Parameter : We set the param_grid for Hyper-Parameter (n_neighbors = list(range(1,10))) and cv = 5, sets the K-Fold as 5-Fold Cross validation. 'estimator' = KNeighborsClassifier( )

param_n_neighbors	mean_test_score	mean_train_score
1	0.9158	1.0000
2	0.8904	0.9604
3	0.8901	0.9455
4	0.8757	0.9275
5	0.8711	0.9167
6	0.8616	0.9053
7	0.8586	0.8957
8	0.8524	0.8862



We have got best score 0.9158 at best 'n_neighbors' = 1.

Figure 32-Hyper-Tunning Parameter of KNN Regression

4.7.4 Support Vector Machine (SVM)

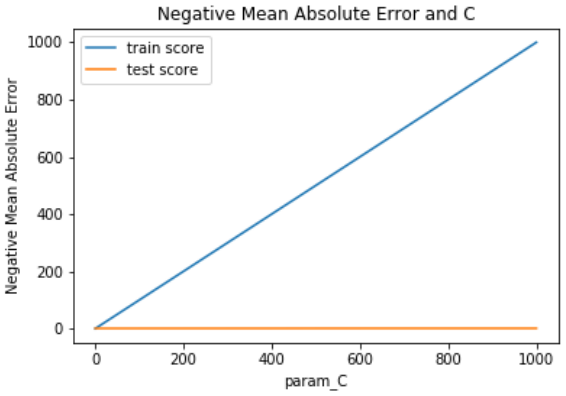
For hyper tuning SVM, we have used GridSearchCV.

For param_grid , we have ,C = [0.1, 1, 10, 100, 200, 300, 400, 500, 600 , 1000]

cv = 5, which sets the K-Fold as 5-Fold Cross-Validation.

estimator = SVC( ).

param_C	mean_test_score	mean_train_score
0.1	0.6863	0.6863
1	0.6863	0.6863
10	0.6863	0.6863
100	0.7028	0.7028
200	0.7260	0.7262
300	0.7493	0.7494
400	0.7704	0.7707
500	0.8211	0.8209
600	0.8564	0.8565
1000	0.8818	0.8818



We have best score as 0.8818 at best C = 1000

Figure 33-Hyper-Tunning Parameter of SVM Classifier

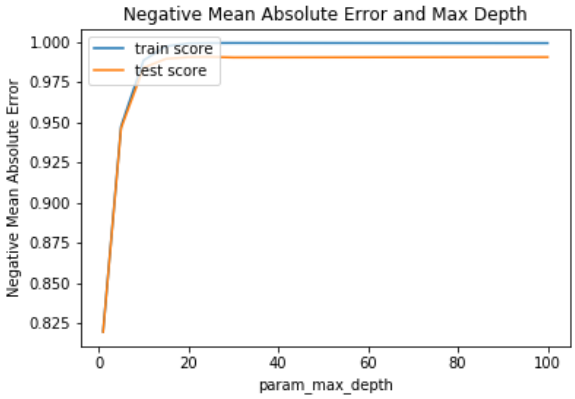
4.7.5 Random Forest

For hyper-tuning Random Forest, we have used GridSearchCV.

To set the param_grid, for hyper tuning

'bootstrap' = [True] ,'max_depth' = [1 , 5 , 10 , 15 , 20 , 25 , 30 , 100] and cv = 5, which sets the K-Fold as 5-Fold. estimator = RandomForestClassifier()

param_max_depth	mean_test_score	mean_train_score
1	0.8196	0.8199
5	0.9463	0.9475
10	0.9839	0.9885
15	0.9894	0.9972
20	0.9905	0.9991
25	0.9906	0.9993
30	0.9903	0.9993
100	0.9905	0.9992



We have best score as 0.9906 at max_depth = 15

Figure 34-Hyper-Tunning Parameter of Random Forest Classification

4.7.6 Gradient Boosting Machine (GBM)

For hypertunning GBM, we have used GridSearchCV with param_grid as below:

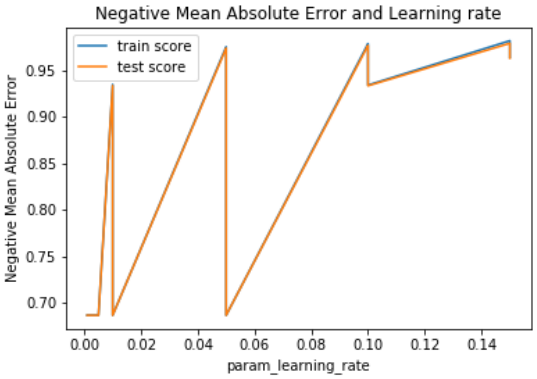
param_grid = {'learning_rate':[0.15,0.1,0.05,0.01,0.005,0.001], 'n_estimators':[2,5,10,20]}

And cv = 5,.

estimator = GradientBoostingClassifier(max_depth=4,min_samples_split=2,min_samples_leaf=1,random_state=10).

The below table shows max_depth values and mean, test, and train scores.

param_learning_rate	param_n_estimators	mean_test_score	mean_train_score
0.15	2	0.9633	0.9644
0.15	5	0.9748	0.9762
0.15	10	0.9763	0.9778
0.15	20	0.9793	0.9821
0.1	2	0.9336	0.9343
0.1	5	0.9732	0.9742
0.1	10	0.9740	0.9753
0.1	20	0.9770	0.9789
0.05	2	0.6863	0.6863
0.05	5	0.9580	0.9589
0.05	10	0.9732	0.9744
0.05	20	0.9741	0.9755
0.01	2	0.6863	0.6863
0.01	5	0.6863	0.6863
0.01	10	0.6863	0.6863
0.01	20	0.9337	0.9350
0.005	2	0.6863	0.6863
0.005	5	0.6863	0.6863
0.005	10	0.6863	0.6863
0.005	20	0.6863	0.6863
0.001	2	0.6863	0.6863
0.001	5	0.6863	0.6863
0.001	10	0.6863	0.6863
0.001	20	0.6863	0.6863



We have best score as 0.9793 at learning_rate = 0.15 and n_estimators = 20.

Figure 35-Hyper-Tunning Parameter of GBM Classification

4.7.7 Deep Neural Networks (DNN)

In DNN we added an input layer, two hidden layers, and one output layer. The input layer is defined with 64 input nodes or neurons as we have 64 features and activation type as 'relu.'

```
Model1 = Sequential()
## Add input layer
Model1.add(Dense(64, input_dim=X_train.shape[1] , activation='relu'))
```

Then we add our first hidden layer with 128 input nodes and activation type as ‘relu’

```
## Add hidden layer
Model1.add(Dense(128, activation='relu'))
```

The second hidden layer is added with input nodes as 196.

```
##Add 2nd hidden layer
```

Model1.add(Dense(196, activation='relu'))

Finally output layer is added with outpot nodes as 4 as we have categorical output

Add output layer

Model1.add(Dense(4, activation='sigmoid'))

The model is compiled with loss type as ‘categorical_crossentropy.

Compile the model

Model1.compile(optimizer='adam',

loss= categorical_crossentropy,

metrics=['accuracy'])

```
Using TensorFlow backend.

Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 64)                704
_________________________________________________________________
dense_2 (Dense)              (None, 128)               8320
_________________________________________________________________
dense_3 (Dense)              (None, 196)               25284
_________________________________________________________________
dense_4 (Dense)              (None, 4)                 788
=================================================================
Total params: 35,096
Trainable params: 35,096
Non-trainable params: 0
_________________________________________________________________
None
```

While training the model we have used epochs = 20 and found the best accuracy at least loss.

Fil model on Training data

Model1.fit(X_train, Y_train, epochs=20)

At loss= 0.145 , accuracy = 0.922

4.8 Model Building and Hyper-Parameter Optimization- Smartphone-based Sensor Data

4.8.1 Logistics Regression

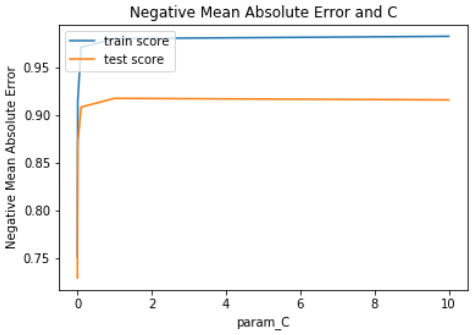
Logistics regression is used to predict the activity label for smartphone data and sensor-based dataset.

Optimization of Hyper-Parameter: Penalty Parameter and Param C is optimized for the below range.

Penalty=[11,12] ; C =[0.001,0.01,0.1,1,10,100,1000,10000] and cv= 5

Below accuracy is achieved for sensor-based datasets.

param_C	param_penalty	mean_test_score	mean_train_score
0.001	l1	0.7296	0.7518
0.001	l2	0.8881	0.9314
0.01	l1	0.8716	0.9159
0.01	l2	0.9037	0.9585
0.1	l1	0.9090	0.9718
0.1	l2	0.9129	0.9731
1	l1	0.9183	0.9810
1	l2	0.9144	0.9796
10	l1	0.9166	0.9834
10	l2	0.9135	0.9825
100	l1	0.9129	0.9834
100	l2	0.9140	0.9836
1000	l1	0.9113	0.9836
1000	l2	0.9120	0.9836
10000	l1	0.9108	0.9836
10000	l2	0.9094	0.9838



At C = 1, the best score is 0.9183 at penalty L1.

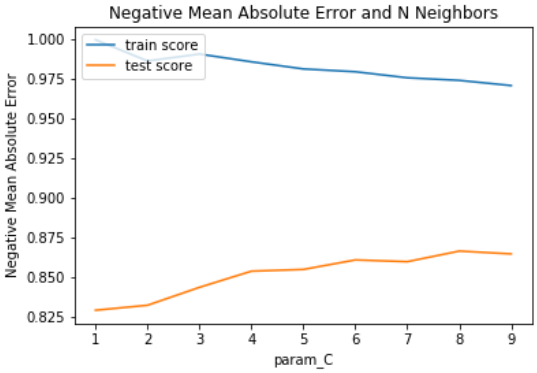
Figure 36-Hyper-Tunning Parameter of Logistics Regression

4.8.2 K-Nearest Neighbors (KNN)

For hyper-tunning KNN, we have again used GridSearchCV.

Optimization of Hyper-Parameter : We set the param_grid for Hyper-Parameter (n_neighbors = list(range(1,10))) and cv = 5, sets the K-Fold as 5-Fold Cross validation. 'estimator' = KNeighborsClassifier( )

param_n_neighbors	mean_test_score	mean_train_score
1	0.8296	1.0000
2	0.8327	0.9867
3	0.8440	0.9910
4	0.8542	0.9861
5	0.8553	0.9816
6	0.8613	0.9799
7	0.8602	0.9761
8	0.8668	0.9744
9	0.8651	0.9712



We have got best score 0.8668 at best 'n_neighbors' = 8.

Figure 37-Hyper-Tunning Parameter of KNN Regression

4.8.3 Support Vector Machine (SVM)

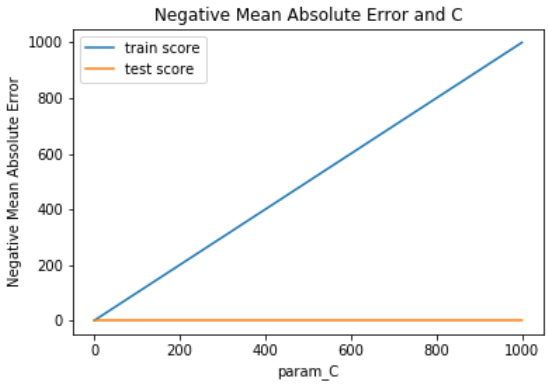
For hyper tuning SVM, we have used GridSearchCV.

For param_grid , we have ,C = [0.1, 1, 10, 100, 200, 300, 400, 500, 600 , 1000]

cv = 5, which sets the K-Fold as 5-Fold Cross-Validation.

estimator = SVC( ).

param_C	mean_test_score	mean_train_score
0.1	0.1914	0.1914
1	0.5495	0.5622
10	0.8807	0.9044
100	0.9120	0.9559
200	0.9193	0.9682
300	0.9225	0.9714
400	0.9241	0.9735
500	0.9251	0.9754
600	0.9268	0.9765



We have best score as 0.9282 at best C = 1000

Figure 38-Hyper-Tunning Parameter of SVM Classifier

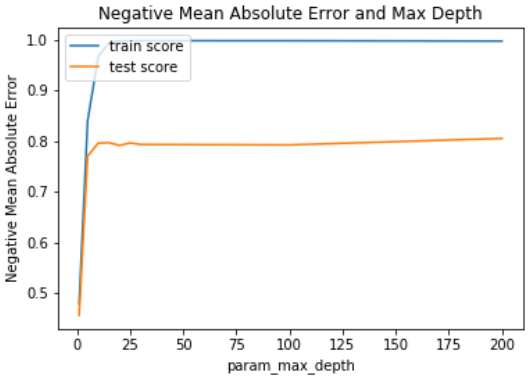
4.8.4 Random Forest

For hyper-tuning Random Forest, we have used GridSearchCV.

To set the param_grid, for hyper-tuning

'bootstrap' = [True] ,'max_depth' = [1 , 5 , 10 , 15 , 20 , 25 , 30 , 100] and cv = 5, which sets the K-Fold as 5-Fold. estimator = RandomForestClassifier()

param_max_depth	mean_test_score	mean_train_score
1	0.4570	0.1914
5	0.7709	0.5622
10	0.7965	0.9044
15	0.7973	0.9559
20	0.7919	0.9682
25	0.7968	0.9714
30	0.7938	0.9735
100	0.7930	0.9754



We have best score as 0.7930 at max_depth = 100

Figure 39-Hyper-Tunning Parameter of Random Forest Classification

4.8.5 Gradient Boosting Machine (GBM)

For hypertunning GBM, we have used GridSearchCV with param_grid as below:

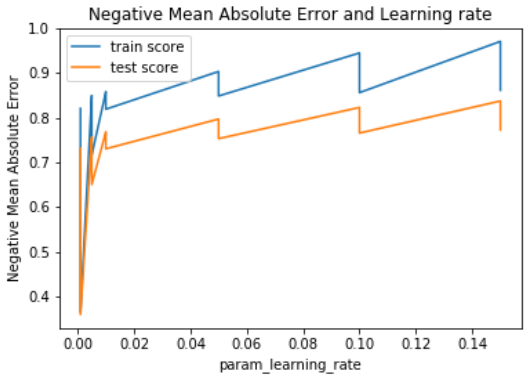
param_grid = {'learning_rate':[0.15,0.1,0.05,0.01,0.005,0.001], 'n_estimators':[2,5,10,20]}

And cv = 5,.

estimator = GradientBoostingClassifier(max_depth=4,min_samples_split=2,min_samples_leaf=1,random_state=10).

The below table shows max_depth values and mean, test, and train scores.

param_learning_rate	param_n_estimators	mean_test_score	mean_train_score
0.15	2	0.7727	0.8615
0.15	5	0.7870	0.8873
0.15	10	0.8101	0.9240
0.15	20	0.8373	0.9706
0.1	2	0.7656	0.8561
0.1	5	0.7791	0.8754
0.1	10	0.7975	0.9034
0.1	20	0.8229	0.9447
0.05	2	0.7531	0.8485
0.05	5	0.7697	0.8612
0.05	10	0.7822	0.8749
0.05	20	0.7971	0.9033
0.01	2	0.7304	0.8190
0.01	5	0.7484	0.8445
0.01	10	0.7568	0.8500
0.01	20	0.7684	0.8581
0.005	2	0.6504	0.7160
0.005	5	0.7410	0.8294
0.005	10	0.7493	0.8447
0.005	20	0.7565	0.8496
0.001	2	0.3598	0.3658
0.001	5	0.4905	0.5271
0.001	10	0.6285	0.6953
0.001	20	0.7318	0.8207



We have best score as 0.8373 at learning_rate = 0.15 and n_estimators = 20.

Figure 40-Hyper-Tunning Parameter of GBM Classification

4.8.6 Deep Neural Networks (DNN)

In DNN we add multiple layers, starting with an input layer, two hidden layers, and one output layer. The input layer is defined with 64 input nodes or neurons as we have 64 features and activation type as 'relu'.

Model1 = Sequential()

Add input layer

Model1.add(Dense(64, input_dim=X_train.shape[1] , activation='relu'))

Then we add our first hidden layer with 128 input nodes and activation type as ‘relu’

Add hidden layer

Model1.add(Dense(128, activation='relu'))

The second hidden layer is added with input nodes as 196.

##Add 2nd hidden layer

Model1.add(Dense(196, activation='relu'))

Finally output layer is added with outpot nodes as 6 as in smartphone dataset we have 6 activity labels

Add output layer

Model1.add(Dense6, activation='sigmoid'))

The model is compiled with loss type as ‘categorical_crossentropy’.

Compile the model

Model1.compile(optimizer='adam',

loss= ‘categorical_crossentropy’,s

metrics=['accuracy'])

```
29 print(Model1.summary())
```

```
Model: "sequential_8"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_29 (Dense)             (None, 64)                6592
_________________________________________________________________
dense_30 (Dense)             (None, 128)               8320
_________________________________________________________________
dense_31 (Dense)             (None, 196)               25284
_________________________________________________________________
dense_32 (Dense)             (None, 6)                 1182
=================================================================
Total params: 41,378
Trainable params: 41,378
Non-trainable params: 0
_________________________________________________________________
None
```

While training the model, we have used epochs = 20 and found the best accuracy at least loss.

Fil model on Training data

Model1.fit(X_train, Y_train, epochs=20)

At loss= 0.145 , accuracy = 0.922

4.9 Summary

In this section, first, we loaded both the dataset in python and then checked the data for any missing value or impurities. Both the dataset are good and do not need any data cleaning. For the sensor-based data set, we created two new field Rooms and Gender after reading data in python. As Room and Gender field was categorical, so created a dummy variable for that field. For smart Phone dataset, the input data contain 561 fields, and those columns have multicollinearity, so we performed PCA to reduce feature. Once we have data ready, we build the model and optimized hyper-parameter by using GridsearchCV method. Once we have a model ready, we will compare results for these models and find a good model for human activity recognition.

CHAPTER 5

5 RESULTS AND EVALUATION

5.1 Introduction

This chapter deals with the detailed execution of model building and evaluating the results and performance.

5.2 Model Output – Wearable Sensor based dataset

As we are dealing with the multiclass problem and class-in balanced dataset so accuracy will not be able to tell the full story. So we will be comparing all the performance evaluation metrics like precision, recall, sensitivity, specificity, F-score to find the best model for the smartphone dataset and sensor-based dataset.

5.2.1 Logistic Regression

We created a logistics regression model with an optimized hyperparameter. (C=1000, penalty='l1',)

We got below a confusion matrix from Logistics regression model:-

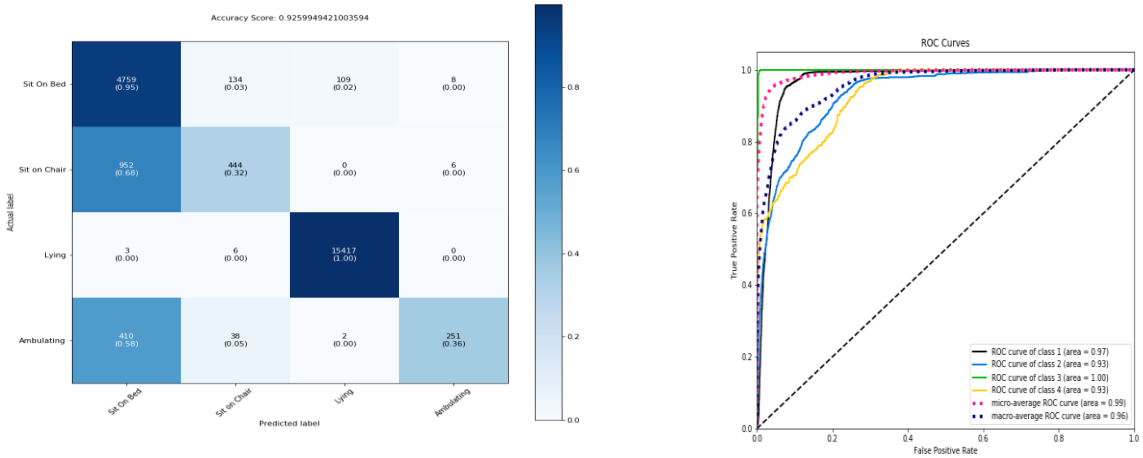


Figure-41 ROC and Confusion Matrix for Logistics Regression(Wearable Sensor based Dataset)

Results:-We get 92% accuracy on test data. Still, when we see the confusion matrix, we can see that for lying activity, almost all the records are correctly identified. But for 'Sit on Chair' activity, approximately 66% of records are identified as Sit on bed. Similarly, 58% of Ambulating activity records are identified as Sit on Bed. So we can say that even when logistics regression has 92% accuracy, but it is still not able to identify all the activity correctly.

To check all the parameters of the model, we created the below table of the performance metrics for all the activity, and it also signifies that F1 score & Recall is very less for activity Sit on Chair and Ambulating.

Performance Matrix	Sit On Bed	Sit on Chair	Lying	Ambulating
f1-score	0.8549	**0.4387**	0.9961	0.5197
precision	0.7771	**0.7138**	0.9929	0.9472
recall	0.9499	**0.3167**	0.9994	**0.3581**
support	5010.0000	1402.0000	15426.0000	701.0000
Training Accuracy	0.9242	0.9242	0.9242	0.9242
Test Accuracy	0.9260	0.9260	0.9260	0.9260
Model_Time	1.2065	1.2065	1.2065	1.2065
Specificity	0.9221	0.9916	0.9844	0.9994
Negative predictive value	0.9847	0.9563	0.9987	0.9798
Positive Likelihood Ratio	12.1984	37.6061	64.0437	558.5223
Negative likelihood ratio	18.4058	1.4511	1687.2526	1.5568
Overall Misclassification Ratio	0.0717	0.0504	0.0053	0.0206
ROC curve_Avergae	0.9700	0.9300	1.0000	0.9300

Table 8–Performance Matrix for Logistics Regression(Wearable Sensor dataset)

5.2.2 K-Nearest Neighbor (KNN)

We create KNN Classifier model with the optimized parameter(n_neibhours=1). Model accuracy was 98% with test data, so we checked other parameters of the model.

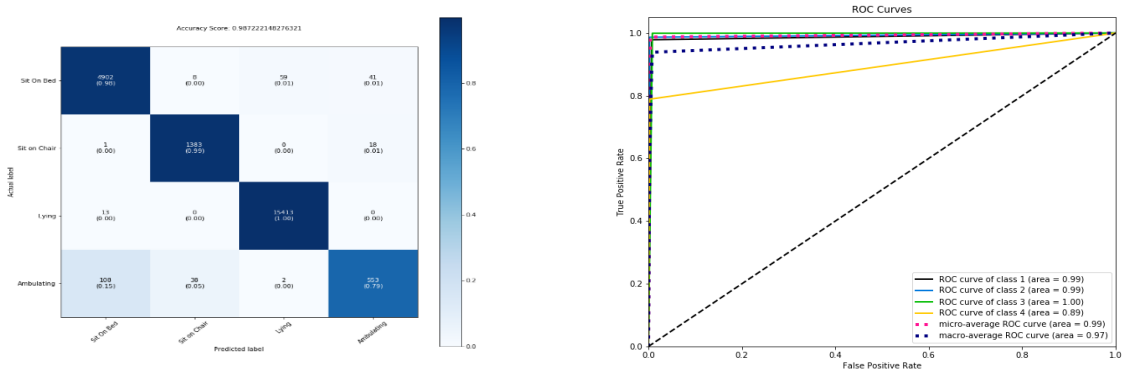


Figure-42-ROC and Confusion Matrix for KNN Classifier (Wearable Sensor based Dataset)

The below table shows the different performance matrix for the KNN model.

Performance Matrix	Sit On Bed	Sit on Chair	Lying	Ambulating
f1-score	0.9771	0.9770	0.9976	0.8423
precision	0.9757	0.9678	0.9961	0.9036
Recall	0.9784	0.9864	0.9992	**0.7889**
support	5010.0000	1402.0000	15426.0000	701.0000
Training Accuracy	1.0000	1.0000	1.0000	1.0000
Test Accuracy	0.9872	0.9872	0.9872	0.9872
Model_Time	0.0987	0.0987	0.0987	0.0987
Specificity	0.9930	0.9978	0.9914	0.9973
Negative predictive value	0.9938	0.9991	0.9982	0.9933
Positive Likelihood Ratio	140.5830	453.2728	116.5083	291.9900
Negative likelihood ratio	46.0660	73.6289	1176.4392	4.7237
Overall Misclassification Ratio	0.0102	0.0029	0.0033	0.0092
ROC curve_Avergae	0.9900	0.9900	1.0000	0.8900

Table 9 –Performance Matrix for KNN (Wearable Sensor dataset)

This model confusion matrix shows better results than logistics regression. Recall and F1 score improved to 97% compared to logistics regression, but still, Ambulating recall is 70%. We are still not able to identify 15% Ambulating activity record correctly.

5.2.3 Support Vector Machine (SVM)

SVM model is created with an optimized hyperparameter(C=1000). We used kernel as rbf, which is a complex SVM and takes a lot of fo time.SVM is also able to identify all the activities correctly, but almost 21% of Ambulating records are identified as Sit on bed.

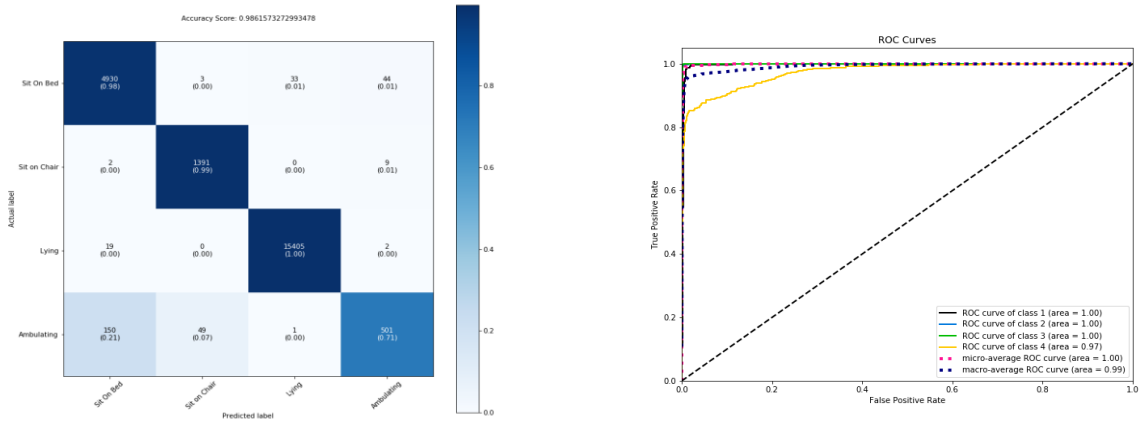


Figure-43 ROC and Confusion Matrix for SVM(Wearable Sensor based Dataset)

The below table shows the different performance matrix for the SVM model.

Performance Matrix	**Sit On Bed**	**Sit on Chair**	**Lying**	**Ambulating**
f1-score	0.9752	0.9779	0.9982	0.7971
precision	0.9665	0.9640	0.9978	0.9011
recall	0.9840	0.9922	0.9986	**0.7147**
support	5010.0000	1402.0000	15426.0000	701.0000
Training Accuracy	0.9956	0.9956	0.9956	0.9956
Test Accuracy	0.9862	0.9862	0.9862	0.9862
Model_Time	66.0916	66.0916	66.0916	66.0916
Specificity	0.9902	0.9975	0.9952	0.9975
Negative predictive value	0.9954	0.9995	0.9970	0.9909
Positive Likelihood Ratio	100.8719	403.2915	208.9211	283.7722
Negative likelihood ratio	62.0141	127.1410	731.0602	3.4962
Overall Misclassification Ratio	0.0111	0.0028	0.0024	0.0113
ROC curve_Avergae	1.0000	1.0000	1.0000	0.9700

Table 10–Performance Matrix for SVM(Wearable Sensor dataset)

5.2.4 Random Forest

Random and Forest model is created with optimized hyperparameter.

The below graph shows the confusion matrix and the ROC curve of the model.

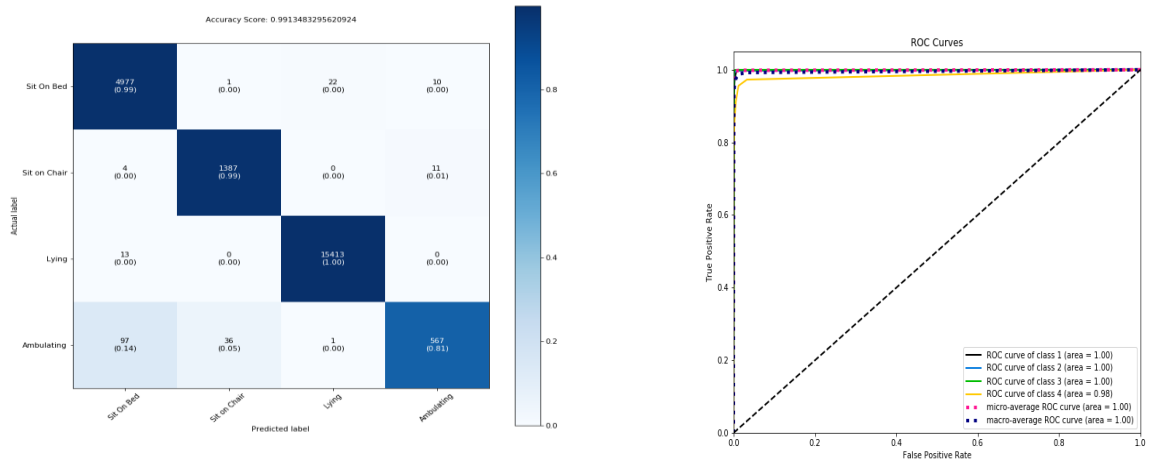


Figure-44 ROC and Confusion Matrix for Random Forest Algorithm (Wearable Sensor dataset)

The below table shows different performance matrix for the Random Forest model.

Performance Matrix	**Sit On Bed**	**Sit on Chair**	**Lying**	**Ambulating**
f1-score	0.9854	0.9816	0.9988	0.8798
precision	0.9776	0.9740	0.9985	0.9643
Recall	0.9934	0.9893	0.9992	**0.8088**
support	5010.0000	1402.0000	15426.0000	701.0000
Training Accuracy	0.9994	0.9994	0.9994	0.9994
Test Accuracy	0.9913	0.9913	0.9913	0.9913
Model_Time	0.5046	0.5046	0.5046	0.5046
Specificity	0.9935	0.9982	0.9968	0.9990
Negative predictive value	0.9981	0.9993	0.9982	0.9939
Positive Likelihood Ratio	152.7503	565.1582	309.0002	841.1213
Negative likelihood ratio	150.8308	93.3031	1182.7784	5.2263
Overall Misclassification Ratio	0.0065	0.0023	0.0016	0.0069
ROC curve_Avergae	1.0000	1.0000	1.0000	0.9800

Table 11 –Performance Matrix for Random Forest(Wearable Sensor dataset)

As per the confusion matrix,14% of Ambulating activity records are not correctly identified. Those records are identified as "Sit on Bed". All other activity records are correctly identified. We were able to achieve 80% recall with the Random forest model with 99% accuracy on test data.

5.2.5 Gradient Boosting Machine (GBM)

The GBM model is created with hyper tune parameter(Learning rate =0.15 and estimator =20)

The below graph shows the confusion matrix and ROC curve of the final outcome.

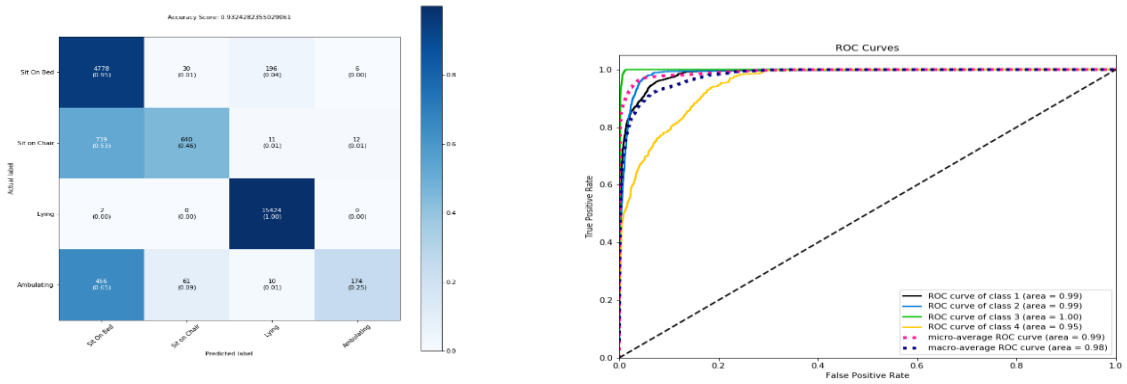


Figure-45 ROC and Confusion Matrix for GBM(Wearable Sensor)

The below table shows the different performance matrix for the GBM model.

Performance Matrix	**Sit On Bed**	**Sit on Chair**	**Lying**	**Ambulating**
f1-score	0.8699	**0.6001**	0.9930	**0.3897**
precision	0.7997	0.8755	0.9861	0.9063
recall	0.9537	**0.4565**	0.9999	**0.2482**
support	5010.0000	1402.0000	15426.0000	701.0000
Training Accuracy	0.9320	0.9320	0.9320	0.9320
Test Accuracy	0.9324	0.9324	0.9324	0.9324
Model_Time	1.2653	1.2653	1.2653	1.2653
Specificity	0.9317	0.9957	0.9695	0.9992
Negative predictive value	0.9860	0.9651	0.9997	0.9764
Positive Likelihood Ratio	13.9660	106.0313	32.7746	301.1422
Negative likelihood ratio	20.1202	1.8320	7477.6955	1.3291
Overall Misclassification Ratio	0.0634	0.0378	0.0097	0.0242
ROC curve_Avergae	0.9900	0.9900	1.0000	0.9500

Table 11 –Performance Matrix for GBM(Wearable Sensor dataset)

Recall, F1-score is very less for "Sit on chair "activity and Ambulating activity for the GBM model in comparison to the Random forest model.

5.2.6 Deep Neural Network (DNN)

DNN model is created with one input layer, two hidden layers, and one output layer. As per DNN Confusion matrix, we are not able to identify 30% of the Amulating record correctly.

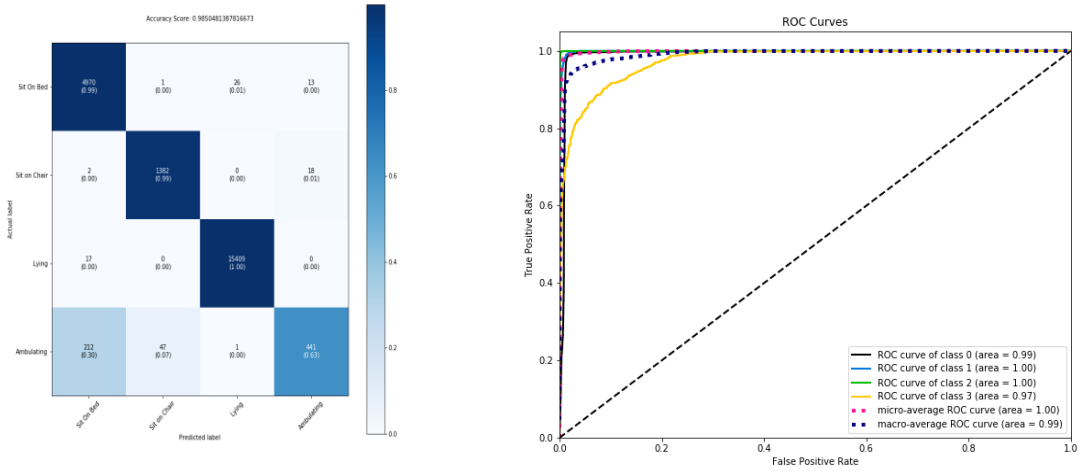


Figure-46 ROC and Confusion Matrix for DNN(Wearable Sensor based dataset)

The below table shows the different performance matrix for the DNN model.

Performance Matrix	Sit On Bed	Sit on Chair	Lying	Ambulating
f1-score	0.9735	0.9760	0.9986	0.7519
precision	0.9556	0.9664	0.9983	0.9343
recall	0.9920	0.9857	0.9989	0.6291
support	5010.0000	1402.0000	15426.0000	701.0000
Training Accuracy	0.9877	0.9877	0.9877	0.9877
Test Accuracy	0.9850	0.9850	0.9850	0.9850
Model_Time	45.1843	45.1843	45.1843	45.1843
Specificity	0.9868	0.9977	0.9962	0.9986
Negative predictive value	0.9977	0.9991	0.9976	0.9882
Positive Likelihood Ratio	75.2773	434.0724	263.1541	443.1714
Negative likelihood ratio	123.5994	69.9408	903.9674	2.6923
Overall Misclassification Ratio	0.0120	0.0030	0.0020	0.0129
ROC curve_Avergae	0.9900	1.0000	1.0000	0.9700

Table 12 –Performance Matrix for DNN (Wearable Sensor dataset)

5.3 Model Output – Smartphone Sensor-based dataset

We created all the six machine learning algorithms with an optimized parameter, and below is the performance matrix and confusion matrix for all algorithms.

5.3.1 Logistics Regression

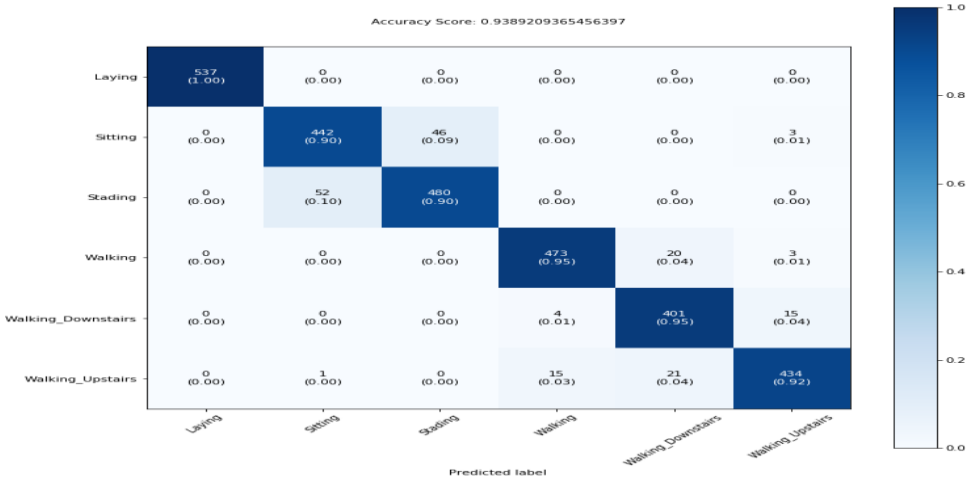


Figure-47 Confusion Matrix of the Logistics Regression model (Smartphone Sensor-based Dataset)

The below table shows the different performance matrix for the logistic regression model.

Performance Matrix	Laying	Sitting	Standing	Walking	Walking Downstairs	Walking Upstairs
f1-score	1	0.9	0.91	0.96	0.93	0.94
precision	1	0.89	0.91	0.96	0.91	0.95
recall	1	0.9	0.9	0.95	0.95	0.92
support	537	491	532	496	420	471
Training Accuracy	0.98	0.98	0.98	0.98	0.98	0.98
Test Accuracy	0.94	0.94	0.94	0.94	0.94	0.94
Model_Time	23.6	23.6	23.6	23.6	23.6	23.6
Specificity	1	0.98	0.98	0.99	0.98	0.99
Negative predictive value	1	0.98	0.98	0.99	0.99	0.99

Positive Likelihood Ratio	inf	41.72	47.37	123.02	58.85	108.64
Negative likelihood ratio	inf	9.8	10.04	21.4	21.75	12.62
Overall Misclassification Ratio	0	0.03	0.03	0.01	0.02	0.02
ROC curve_Avergae	1	0.99	0.99	1	1	1

Table 13–Performance Matrix for Logistics Regression(Smartphone Sensor Dataset)

As per the logistics regression model, we can achieve 94% of test accuracy as well as more than 90% precision and f1-score.

5.3.2 K-Nearest Neighbor (KNN)

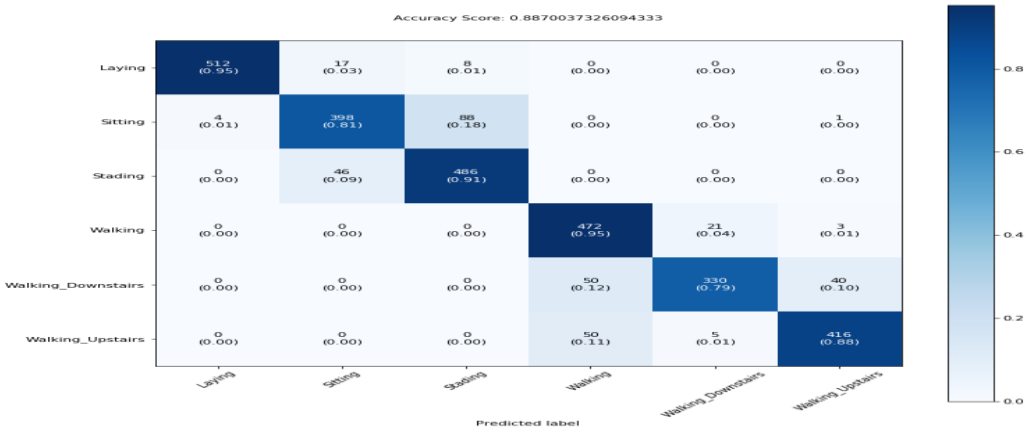


Figure-48 Confusion Matrix of the KNN (Smartphone Sensor-based Dataset)

The below table shows the different performance matrix for the KNN model.

Performance Matrix	**Laying**	**Sitting**	**Standing**	**Walking**	**Walking Downstairs**	**Walking Upstairs**
f1-score	**0.97**	**0.84**	**0.87**	**0.88**	**0.85**	**0.89**
precision	**0.99**	**0.86**	**0.84**	**0.83**	**0.93**	**0.9**
recall	**0.95**	**0.81**	**0.91**	**0.95**	**0.79**	**0.88**

support	537	491	532	496	420	471
Training Accuracy	0.97	0.97	0.97	0.97	0.97	0.97
Test Accuracy	0.89	0.89	0.89	0.89	0.89	**0.89**
Model_Time	0.07	0.07	0.07	0.07	0.07	0.07
Specificity	1	0.97	0.96	0.96	0.99	0.98
Negative predictive value	0.99	0.96	0.98	0.99	0.97	0.98
Positive Likelihood Ratio	574.45	31.6	22.98	23.32	76.37	49.7
Negative likelihood ratio	21.44	5.14	11.11	19.82	4.62	8.41
Overall Misclassification Ratio	0.01	0.05	0.05	0.04	0.04	0.03
ROC curve_Avergae	0.99	0.97	0.99	0.99	0.97	0.99

As per the KNN model, we can achieve 89% of test accuracy as well as more than 83% precision and f1-score.

5.3.3 Support Vector Machine (SVM)

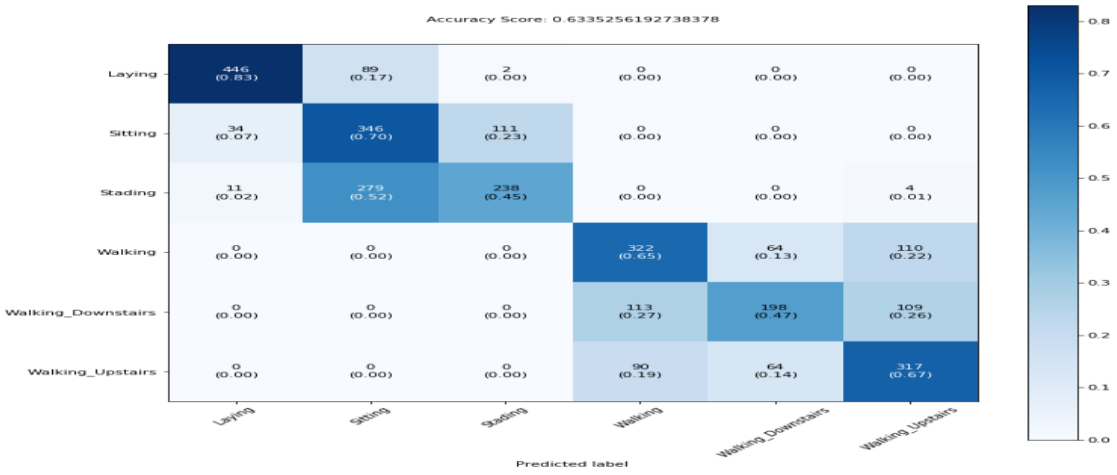


Figure-49 Confusion Matrix of SVM (Smartphone Sensor-based Dataset)

The below table shows the different performance matrix for the SVM model.

Performance Matrix	**Laying**	**Sitting**	**Standing**	**Walking**	**Walking Downstairs**	**Walking Upstairs**
f1-score	0.87	**0.57**	**0.54**	**0.63**	**0.53**	**0.63**
precision	0.91	**0.48**	**0.68**	**0.61**	**0.61**	**0.59**
recall	0.83	**0.7**	**0.45**	**0.65**	**0.47**	**0.67**
support	537	491	532	496	420	471
Training Accuracy	0.98	0.98	0.98	0.98	0.98	**0.98**
Test Accuracy	0.63	0.63	0.63	0.63	0.63	**0.63**
Model_Time	6.56	6.56	6.56	6.56	6.56	6.56
Specificity	0.98	0.85	0.95	0.92	0.95	0.91
Negative predictive value	0.96	0.94	0.89	0.93	0.92	0.94
Positive Likelihood Ratio	44.48	4.7	9.56	7.84	9.31	7.47
Negative likelihood ratio	5.79	2.88	1.72	2.61	1.8	2.78

Overall Misclassification Ratio	0.05	0.17	0.14	0.13	0.12	0.13
ROC curve_Avergae	0.99	0.88	0.87	0.85	0.8	0.87

SVM is not able to classify dynamic activity correctly, and that's why its accuracy, precision, recall is very less.

5.3.4 Random Forest

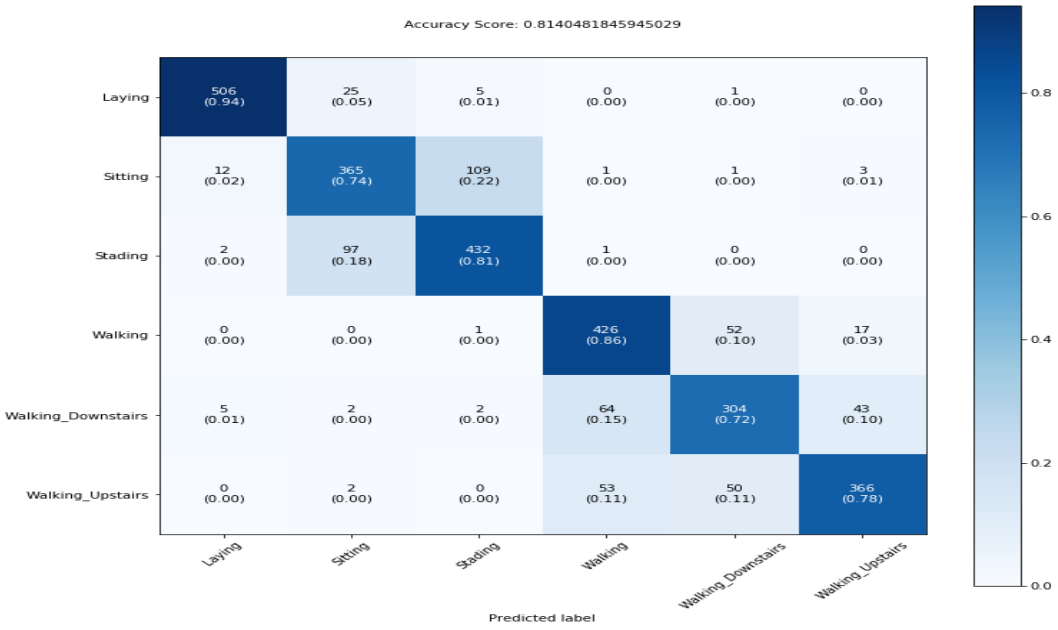


Figure-50 Confusion Matrix of the Random Forest (Smartphone Sensor-based Dataset)

The below table shows different performance matrix for the Random Forest model.

Performance Matrix	**Laying**	**Sitting**	**Standing**	**Walking**	**Walking Downstairs**	**Walking Upstairs**
f1-score	0.95	0.74	0.8	**0.82**	**0.73**	0.81
precision	0.96	0.74	0.79	**0.78**	**0.75**	0.85
recall	0.94	0.74	0.81	**0.86**	**0.72**	0.78
support	537	491	532	496	420	471
Training Accuracy	1	1	1	1	1	1
Test Accuracy	0.81	0.81	0.81	0.81	0.81	0.81
Model_Time	1.24	1.24	1.24	1.24	1.24	1.24
Specificity	0.99	0.95	0.95	0.95	0.96	0.97
Negative predictive value	0.99	0.95	0.96	0.97	0.95	0.96
Positive Likelihood Ratio	119.52	14.49	16.76	17.69	17.59	30.54

Negative likelihood ratio	17.19	3.7	5.06	6.74	3.47	4.37
Overall Misclassification Ratio	0.02	0.09	0.07	0.06	0.07	0.06
ROC curve_Avergae	1	0.95	0.97	0.97	0.95	0.98

We are able to achieve test accuracy as 81% for Random forest, but Random forest also could not correctly identify walking and siting activity accurately.

5.3.5 Gradient Boosting Machine (GBM)

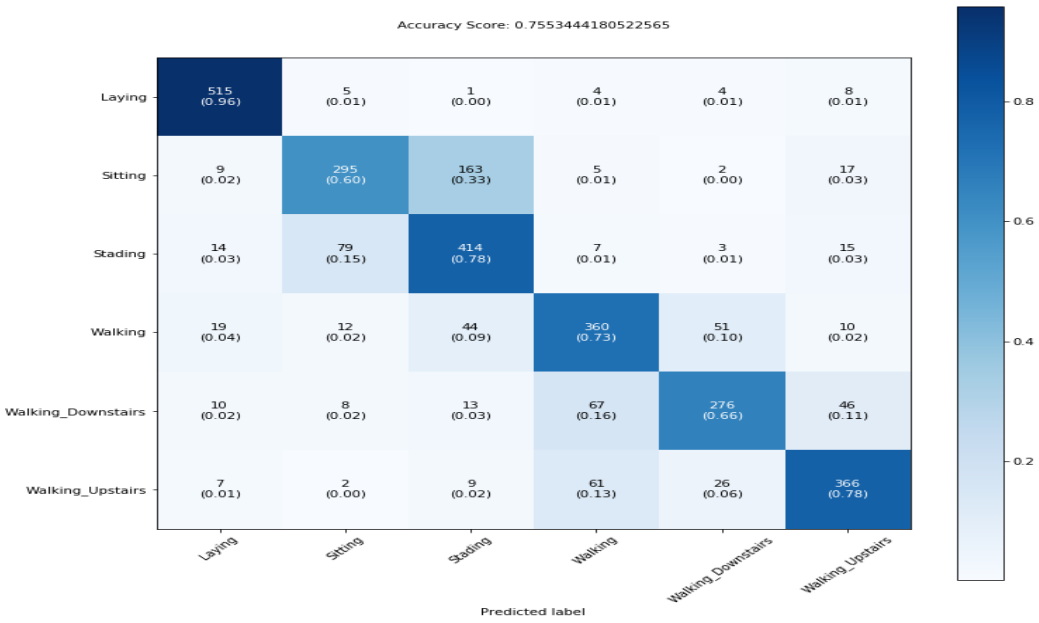


Figure 51 -Confusion Matrix of the GBM (Smartphone Sensor-based Dataset)

The below table shows the different performance matrix for the GBM model.

Performance Matrix	**Laying**	**Sitting**	**Standing**	**Walking**	**Walking Downstairs**	**Walking Upstairs**
f1-score	0.93	**0.66**	**0.7**	**0.72**	**0.71**	**0.78**
precision	0.9	**0.74**	**0.64**	**0.71**	**0.76**	**0.79**

recall	0.96	**0.6**	**0.78**	**0.73**	**0.66**	**0.78**
support	537	491	532	496	420	471
Training Accuracy	0.81	0.81	0.81	0.81	0.81	0.81
Test Accuracy	0.76	0.76	0.76	0.76	0.76	**0.76**
Model_Time	0.54	0.54	0.54	0.54	0.54	0.54
Specificity	0.98	0.96	0.9	0.94	0.97	0.96
Negative predictive value	0.99	0.92	0.95	0.94	0.94	0.96
Positive Likelihood Ratio	39.17	13.92	8.17	12.35	19.31	20.04
Negative likelihood ratio	23.81	2.4	4.08	3.43	2.82	4.31
Overall Misclassification Ratio	0.03	0.1	0.12	0.1	0.08	0.07
ROC curve_Avergae	0.99	0.92	0.92	0.93	0.93	0.97

We are able to achieve 75% accuracy with GBM.

5.3.6 Deep Neural Network (DNN)

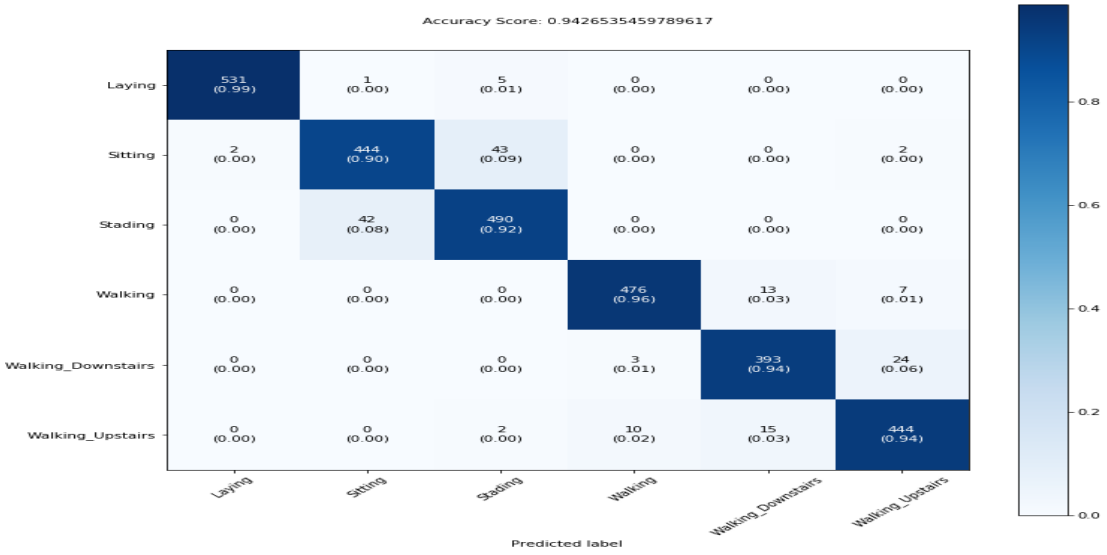


Figure-52 Confusion Matrix of the DNN (Smartphone Sensor-based Dataset)

The below table shows the different performance matrix for the DNN model.

Performance Matrix	**Laying**	**Sitting**	**Standing**	**Walking**	**Walking Downstairs**	**Walking Upstairs**
f1-score	**0.99**	**0.91**	**0.91**	**0.97**	**0.93**	**0.94**
precision	**1**	**0.91**	**0.91**	**0.97**	**0.93**	**0.93**
recall	**0.99**	**0.9**	**0.92**	**0.96**	**0.94**	**0.94**
support	537	491	532	496	420	471
Training Accuracy	1	1	1	1	1	1
Test Accuracy	**0.94**	**0.94**	**0.94**	**0.94**	**0.94**	**0.94**
Model_Time	10.97	10.97	10.97	10.97	10.97	10.97
Specificity	1	0.98	0.98	0.99	0.99	0.99
Negative predictive value	1	0.98	0.98	0.99	0.99	0.99
Positive Likelihood Ratio	1191.54	51.65	44.49	180.94	84.45	70.73

Negative likelihood ratio	89.43	10.26	12.4	24.67	15.38	17.21
Overall Misclassification Ratio	0	0.03	0.03	0.01	0.02	0.02
ROC curve_Avergae	1	0.99	0.99	1	1	1

DNN Model outperformed for the smartphone dataset. F1 score, precision & recall metrics for DNN algorithm is 90%.

5.4 Model Comparision

We will be comparing the models based on different performance matrix, e.g., Accuracy, Precision, Recall(Sensitivity), Specificity, ROC Area(AUC), Overall Misclassification Ratio, NPV, NLR, PLR, F-Score, and Model execution elapsed time.

5.4.1 Accuracy for Wearable Sensor based Dataset

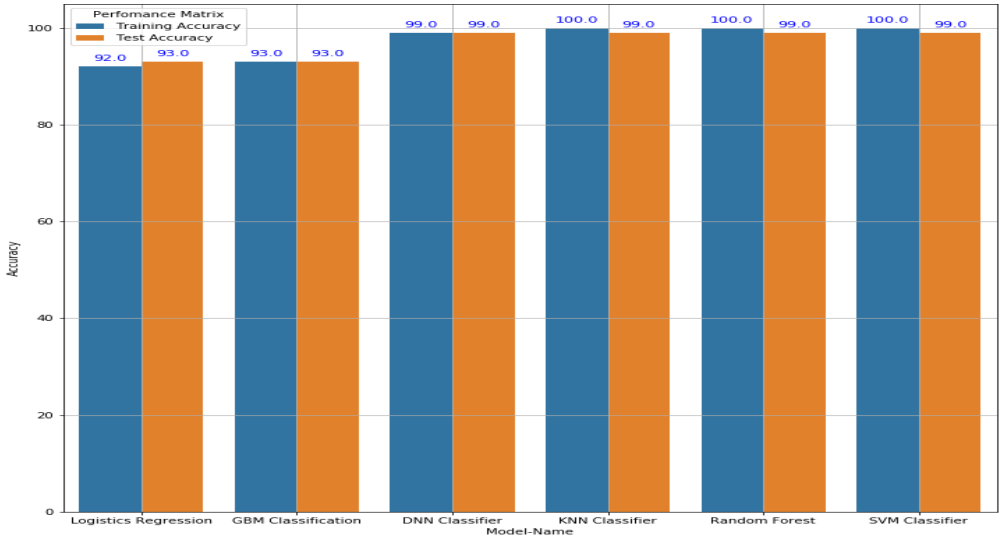


Figure 53-Test and Training accuracy for all the models for the Sensor-based dataset.

The test accuracy is 99% for DNN, KNN, Random Forest, SVM, and 93% for GBM and Logistics regression, but as it is an imbalanced class dataset so accuracy will not be able to clear us a clear picture. We will check other matrics parameter as well.

5.4.2 Accuracy for Smart Phone based dataset.

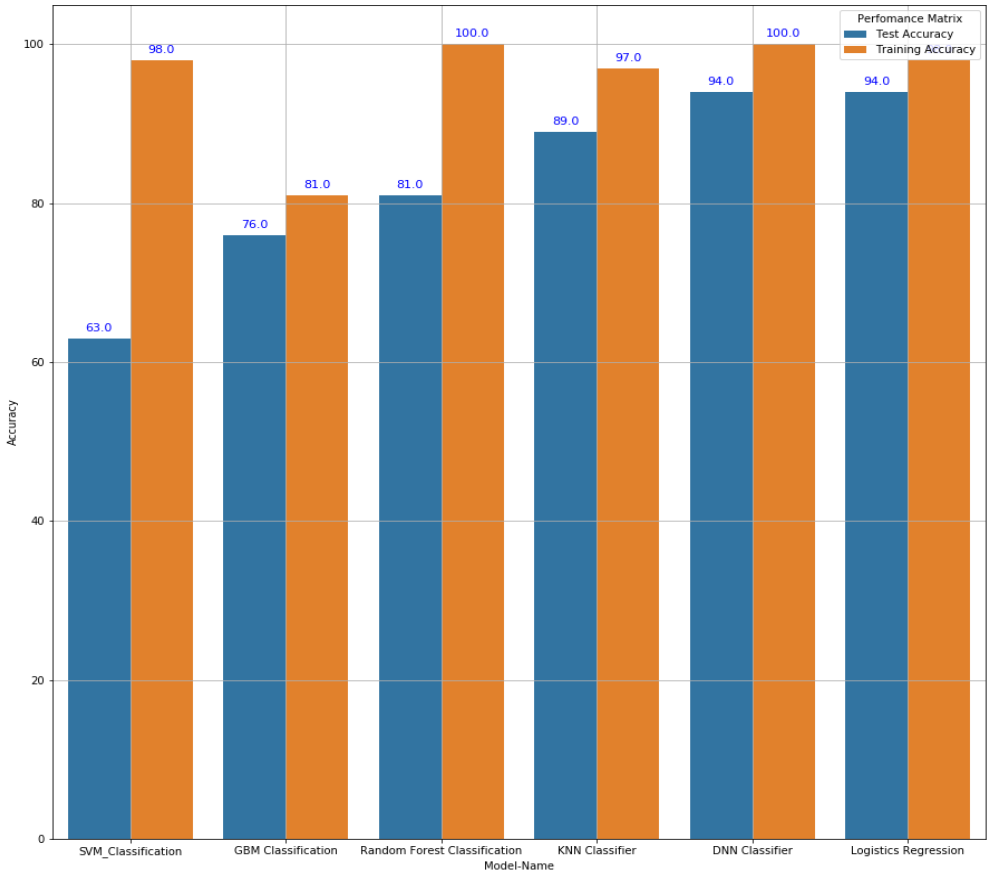


Figure 54 -Test and Training accuracy for all the models for Smart Phone-based dataset.

The test accuracy is 94% for DNN and logistics regression and 89% for KNN Classifier.

5.4.3 Precision vs Recall (Sensitivity) for Sensor based dataset

Precision gives us the proportion of positive identification, which is correct. While Recall provides us with the proportion of actual positives that were identified correctly. There is always a tug of war between Precision and Recall as one comes at the cost of others. So both the parameters need to be checked together for a balanced value.

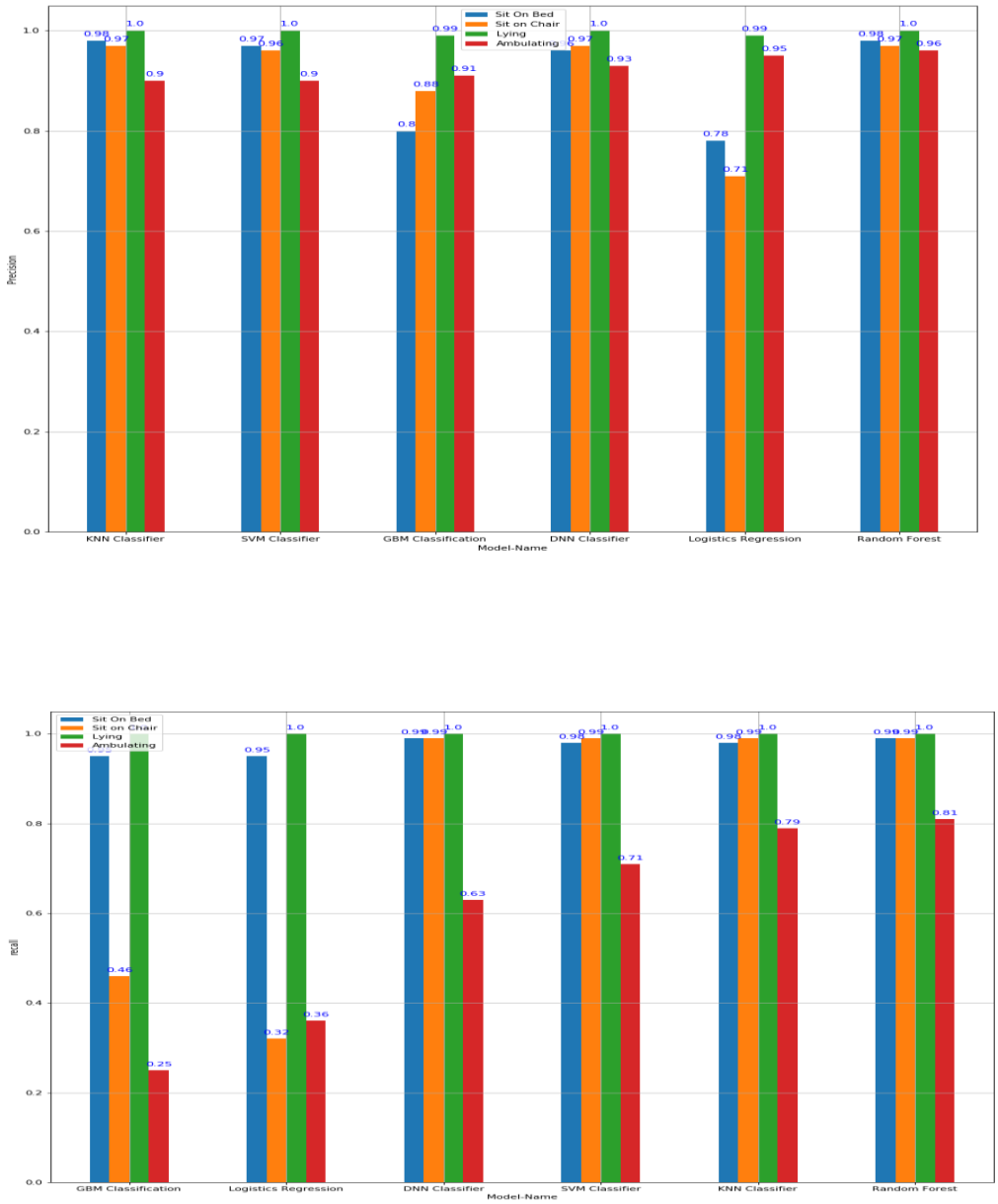


Figure 55-Precision and Recall Parameter for Sensor-based dataset.

If We compare the precision and recall parameter for each activity, then three model KNN, Random, and SVM model can be considered.

Model1	Performance Matrix	Sit On Bed	Sit on Chair	Lying	Ambulating
DNN Classifier	precision	0.96	0.97	1.00	0.93
DNN Classifier	recall	0.99	0.99	1.00	0.63
GBM Classification	precision	0.80	0.88	0.99	0.91
GBM Classification	recall	0.95	0.46	1.00	0.25
KNN Classifier	precision	**0.98**	**0.97**	**1.00**	**0.90**
KNN Classifier	recall	**0.98**	**0.99**	**1.00**	**0.79**
Logistics Regression	precision	0.78	0.71	0.99	0.95
Logistics Regression	recall	0.95	0.32	1.00	0.36
Random Forest	precision	**0.98**	**0.97**	**1.00**	**0.96**
Random Forest	recall	**0.99**	**0.99**	**1.00**	**0.81**
SVM Classifier	precision	**0.97**	**0.96**	**1.00**	**0.90**
SVM Classifier	recall	**0.98**	**0.99**	**1.00**	**0.71**

5.4.4 Precision vs Recall (Sensitivity) for Smartphone sensor-based dataset.

.

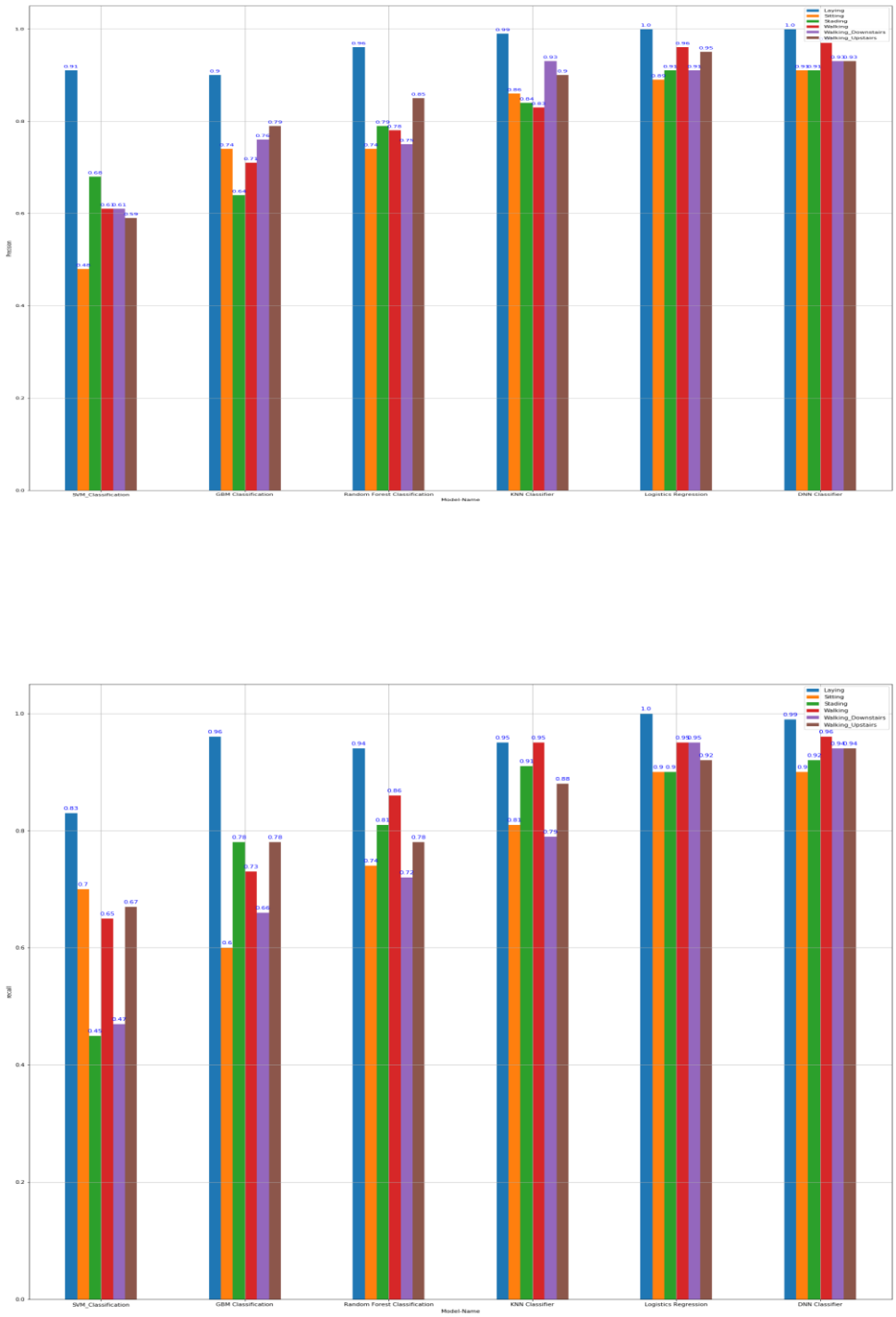


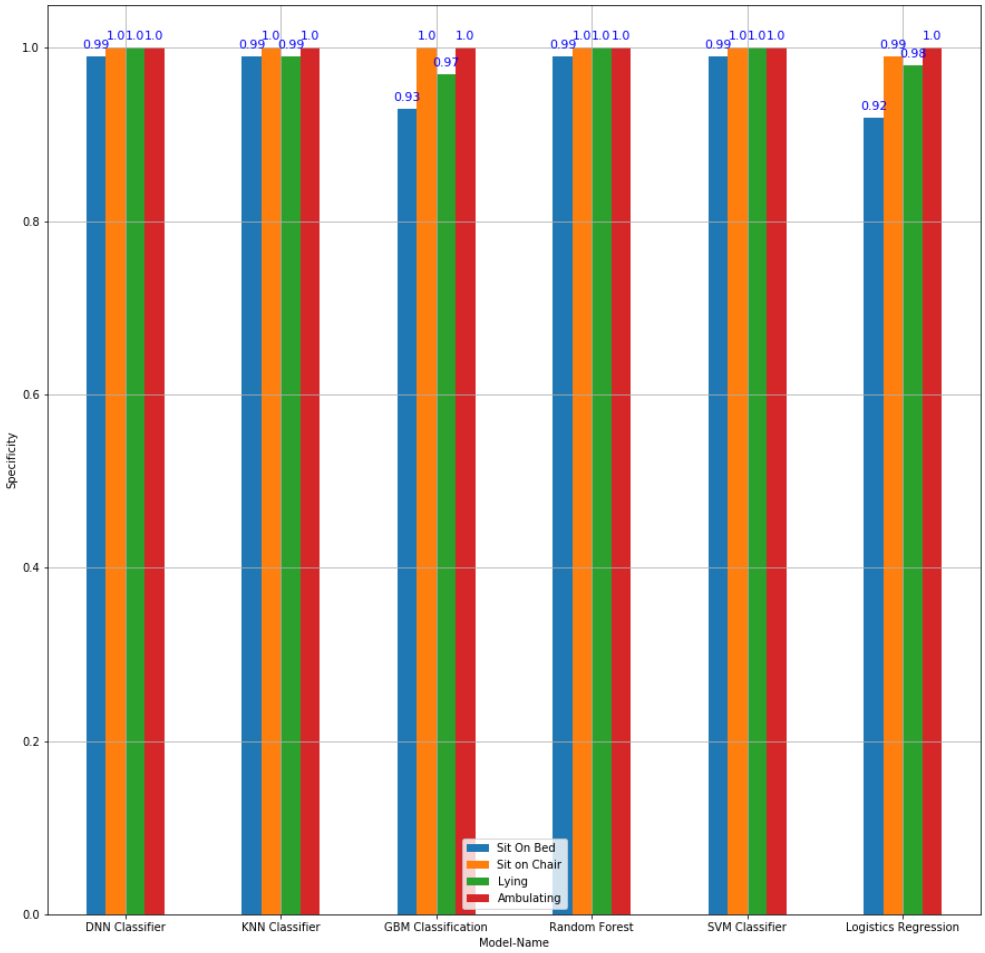
Figure 56-Precision and Recall Parameter for Smartphone-based sensor dataset.

If We compare the precision and recall parameter for each activity, then three model DNN Classifier, KNN, and Logistics regression performed better than others model.

Model1	Performance Matrix	Laying	Sitting	Standing	Walking	Walking_Downstairs	Walking_Upstairs
DNN Classifier	precision	**1**	**0.91**	**0.91**	**0.97**	**0.93**	**0.93**
DNN Classifier	recall	**0.99**	**0.9**	**0.92**	**0.96**	**0.94**	**0.94**
GBM Classification	precision	0.9	0.74	0.64	0.71	0.76	0.79
GBM Classification	recall	0.96	0.6	0.78	0.73	0.66	0.78
KNN Classifier	precision	**0.99**	**0.86**	**0.84**	**0.83**	**0.93**	**0.9**
KNN Classifier	recall	**0.95**	**0.81**	**0.91**	**0.95**	**0.79**	**0.88**
Logistics Regression	precision	**1**	**0.89**	**0.91**	**0.96**	**0.91**	**0.95**
Logistics Regression	recall	**1**	**0.9**	**0.9**	**0.95**	**0.95**	**0.92**
Random Forest Classification	precision	0.96	0.74	0.79	0.78	0.75	0.85
Random Forest Classification	recall	0.94	0.74	0.81	0.86	0.72	0.78
SVM_Classification	precision	0.91	0.48	0.68	0.61	0.61	0.59
SVM_Classification	recall	0.83	0.7	0.45	0.65	0.47	0.67

5.4.5 Specificity for Wearable Sensor based dataset

Specificity is the ability of a model to identify the true negatives. We have a Random forest, DNN, SVM, and KNN as top Specificity. Since Random Forest and KNN have better Sensitivity and precision as well we can consider them as top models.



5.4.5 Time elapsed in Model Execution:

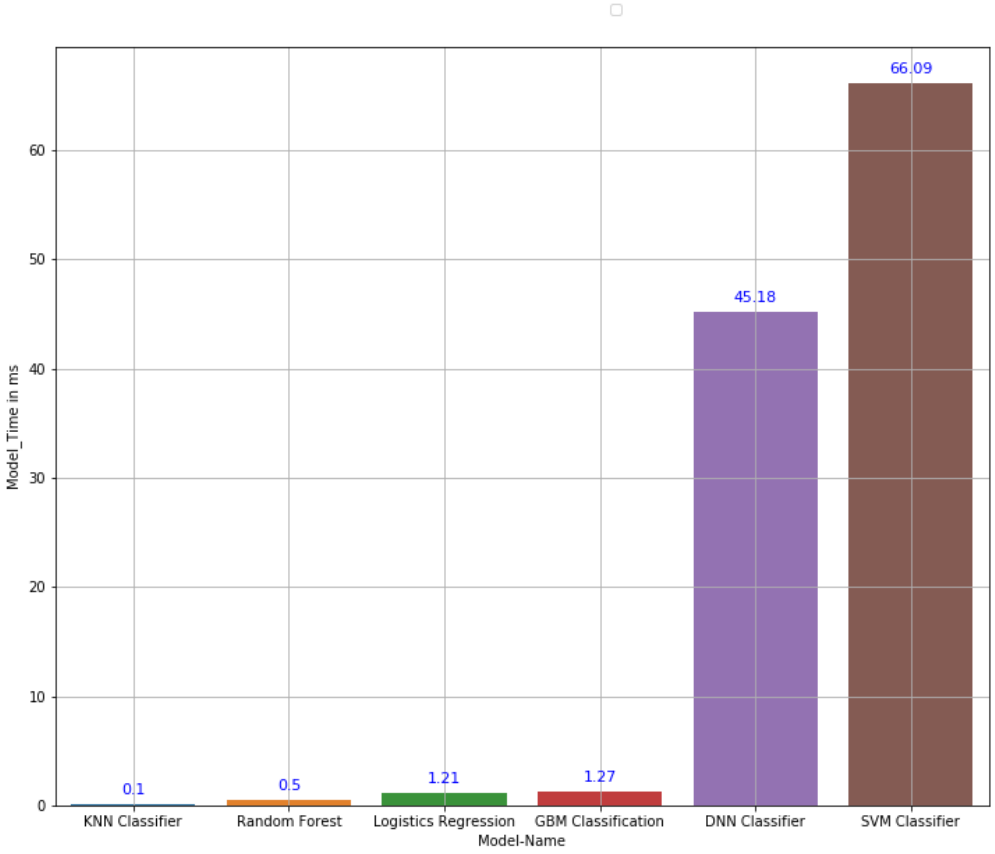


Figure 57 -Elapsed time for model execution for a wearable sensor-based dataset.

When comparing the total time taken for model execution, we have found KNN and Random Forest as the fastest. SVM is slowest, and between Random Forest and DNN Classifier, Random Forest is way faster.

5.6 Summary

We discussed the detailed execution of each model, starting from Logistic Regression, KNN, SVM, Random Forest, GBM, and DNN. We compared all the models based upon performance metrics like accuracy, sensitivity, specificity, recall, precision, Overall misclassification ratio, Roc Area. We found that Random Forest and SVM are the top two models for wearable sensor-based datasets, while DNN,KNN, and logistics regression are the top three models for smartphone-based sensor datasets.

Model Name	Performance Matrix	Laying	Sitting	Standing	Walking	Walking_Downstairs	Walking_U
DNN Classifier	precision	1	0.91	0.91	0.97	0.93	
DNN Classifier	recall	0.99	0.9	0.92	0.96	0.94	
DNN Classifier	Test Accuracy	0.94	0.94	0.94	0.94	0.94	
KNN Classifier	precision	0.99	0.86	0.84	0.83	0.93	
KNN Classifier	recall	0.95	0.81	0.91	0.95	0.79	
KNN Classifier	Test Accuracy	0.89	0.89	0.89	0.89	0.89	
Logistics Regression	precision	1	0.89	0.91	0.96	0.91	
Logistics Regression	recall	1	0.9	0.9	0.95	0.95	
Logistics Regression	Test Accuracy	0.94	0.94	0.94	0.94	0.94	

CHAPTER 6

6 CONCLUSIONS AND RECOMMENDATIONS

6.1 Introduction

In this study, we presented the modeling of the machine learning algorithms & compared the results for six machine learning classification algorithms to see which algorithm will perform better for Human Activity Recognition based upon the wearable sensor dataset and smartphone dataset.

6.2 Discussion and Conclusion

It is very important to identify all the human activity correctly for any application like fitness watch, smart home, or fall detection alert. Usually, it is challenging to identify similar activities like walking, walking upstairs, or walking downstairs or sit on a bed or sit on a chair. In this study, we used two data. Both the datasets are taken from UCI Dataset. One dataset has data for older people measured from the wearable wireless sensor, but the other one has data for young people measure from smartphone-based sensors. In both cases, we build the model using six classification algorithms (Logistics Regression, SVM, KNN, Random Forest, DNN & GBM). As per wearable sensor-based data test accuracy, all the models were able to provide more than 90% accuracy, and Random forest outperformed with an accuracy of 99% for wearable sensor data. But as we know, accuracy can not be the only criterion to judge the classification algorithm, so we check other metrics like Precision, Recall, and F-Score. When we compared recall metrics for all four activities, It is being observed that all the algorithms were able to identify Sit on bed and Lying Activity correctly. But, only KNN and Random Forest were able to provide 80% recall for activity sit on the bed and ambulating activity.

If we compare the time taken by each model, then SVM and DNN took maximum time while KNN and random forest were quick. If we check the overall misclassification ratio, then it is lowest for random forest and highest for logistics regression. As per performance metrics, we can suggest that Random Forest and KNN were two top models for a wearable sensor-based dataset.

Similarly, for Smartphone data where six human activity needs to be identified, As per test accuracy, DNN and Logistics Regression accuracy was 94% even recall parameter is good for DNN and Logistics regression, it is more than 0.9 for all the activity. Overall Miscalssification Ratio was also very low for Logistics Regression and DNN model, so for smartphone data Logistics Regression and DNN outperformed with an accuracy of 94%.

In 2019 (Sriwichian and Muangprathub, 2019), a study was performed on the UCI Wearable sensor dataset, and they were able to achieve 96.5% accuracy with ANN model and 96.41 with SVM. In contrast, we were able to achieve 99% accuracy with DNN & Random forest. They have not considered any other metrics for comparison, but we are making a comparison based upon Recall, Precision, F-score, Sensitivity, Speficifcty, and other parameters.

6.3 Contributions

As we identified the best model for human activity recognition for sensor-based datasets and smartphone-based dataset. These results can be used in multiple human activity monitoring applications like smartphones, fitness tracking, fall detection in nursing homes, or hospitals.

6.3 Future Works

As in this study, we included only six everyday activities. For future work, more activities like running, Jogging, exercise can be added. We used only accelerometer and gyroscope sensor data from the smartphone. Heart sensor data can also be included that can increase our accuracy. As everybody has a smartphone these days, so many applications are being developed, which can identify human behavior and provide recommendations. As of now, we included the recognition of only one human behavior. In future studies can be extended to recognize group behavior.

We performed this activity for 15 or 30 people. If data for a bigger group can be available, then better performance can be provided with a better model.

7 REFERENCES

1. Posted by Sheetal Sharma on August 8, 2017 at 7:00pm, Blog, V., n.d. Artificial Neural Network (ANN) in Machine Learning [WWW Document]. URL https://www.datasciencecentral.com/profiles/blogs/artificial-neural-network-ann-in-machine-learning (accessed 5.5.20).

2. A complete guide to the random forest algorithm [WWW Document], n.d. . Built In. URL https://builtin.com/data-science/random-forest-algorithm (accessed 2.23.20).

3. Ali, H.M., Muslim, A.M., 2016a. Human Activity Recognition Using Smartphone and Smartwatch. ijcert 3, 568. https://doi.org/10.22362/ijcert/2016/v3/i10/48906

4. Balli, S., Sağbaş, E.A., Peker, M., 2019. Human activity recognition from smart watch sensor data using a hybrid of principal component analysis and random forest algorithm. Measurement and Control 52, 37–45. https://doi.org/10.1177/0020294018813692

5. Beniwal, H., 2018. Handwritten Digit Recognition using Machine Learning [WWW Document]. Medium. URL https://medium.com/@himanshubeniwal/handwritten-digit-recognition-using-machine-learning-ad30562a9b64 (accessed 5.5.20).

6. Capela, N.A., Lemaire, E.D., Baddour, N., 2015b. Feature Selection for Wearable Smartphone-Based Human Activity Recognition with Able bodied, Elderly, and Stroke Patients. PLoS One 10. https://doi.org/10.1371/journal.pone.0124414

7. Chetty, G., White, M., 2016. Body sensor networks for human activity recognition, in: 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN). Presented at the 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN), pp. 660–665. https://doi.org

8. Dabbura, I., 2019. K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks [WWW Document]. Medium. URL

https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a (accessed 2.23.20).

9. Do hospital fall prevention programs work? A systematic review. - PubMed - NCBI [WWW Document], n.d. URL https://www.ncbi.nlm.nih.gov/pubmed/11129762 (accessed 4.29.20).

10. Drug Overdose Deaths | Drug Overdose | CDC Injury Center [WWW Document], 2019. URL https://www.cdc.gov/drugoverdose/data/statedeaths.html (accessed 12.12.19).

11. Hitcho, E.B., Krauss, M.J., Birge, S., Claiborne Dunagan, W., Fischer, I., Johnson, S., Nast, P.A., Costantinou, E., Fraser, V.J., 2004. Characteristics and circumstances of falls in a hospital setting: a prospective analysis. J Gen Intern Med 19, 732–739. https://doi.org/10.1111/j.1525-1497.2004.30387.x

12. Hussain, Z., Sheng, M., Zhang, W.E., 2019b. Different Approaches for Human Activity Recognition: A Survey.

13. Index of /ml/machine-learning-databases/00364 [WWW Document], n.d. URL https://archive.ics.uci.edu/ml/machine-learning-databases/00364/ (accessed 5.12.20).

14. Index of /ml/machine-learning-databases/00427 [WWW Document], n.d. URL https://archive.ics.uci.edu/ml/machine-learning-databases/00427/ (accessed 5.12.20)

15. Jobanputra, C., Bavishi, J., Doshi, N., 2019a. Human Activity Recognition: A Survey. Procedia Computer Science 155, 698–703. https://doi.org/10.1016/j.procs.2019.08.100

16. Kwon, M.-C., Choi, S., 2018c. Recognition of Daily Human Activity Using an Artificial Neural Network and Smartwatch [WWW Document]. Wireless Communications and Mobile Computing. https://doi.org/10.1155/2018/2618045

17. Manosha Chathuramali, K.G., Rodrigo, R., 2012. Faster human activity recognition with SVM, in: International Conference on Advances in ICT for Emerging Regions (ICTer2012). Presented at the International Conference on

Advances in ICT for Emerging Regions (ICTer2012), pp. 197–203. https://doi.org/10.1109/IC

18. Martínez-Villaseñor, L., Ponce, H., Espinosa-Loera, R.A., 2018c. Multimodal Database for Human Activity Recognition and Fall Detection. Proceedings 2, 1237. https://doi.org/10.3390/proceedings2191237

19. Medication Overuse Headache [WWW Document], n.d. . American Migraine Foundation. URL https://americanmigrainefoundation.org/resource-library/medication-overuse-headache/ (accessed 12.12.19).

20. Papagiannaki, A., Zacharaki, E., Kalouris, G., Kalogiannis, S., Deltouzos, K., Ellul, J., Megalooikonomou, V., 2019. Recognizing Physical Activity of Older People from Wearable Sensors and Inconsistent Data. Sensors 19, 880. https://doi.org/10.3390/s19040880

21. Posted by Sheetal Sharma on August 8, 2017 at 7:00pm, Blog, V., n.d. Artificial Neural Network (ANN) in Machine Learning [WWW Document]. URL https://www.datasciencecentral.com/profiles/blogs/artificial-neural-network-ann-in-machine-learning (accessed 5.5.20).

22. Reyes-Ortiz, J.-L., Oneto, L., Samà, A., Parra, X., Anguita, D., 2016. Transition-Aware Human Activity Recognition Using Smartphones. Neurocomputing 171, 754–767. https://doi.org/10.1016/j.neucom.2015.07.085

23. Ronao, C.A., Cho, S.-B., 2016a. Human activity recognition with smartphone sensors using deep learning neural networks. Expert Systems with Applications 59, 235–244. https://doi.org/10.1016/j.eswa.2016.04.032

24. SauceCat, 2017. Boosting algorithm: GBM [WWW Document]. Medium. URL https://towardsdatascience.com/boosting-algorithm-gbm-97737c63daa3 (accessed 5.5.20).

25. Shinmoto Torres, R.L., Visvanathan, R., Hoskins, S., van den Hengel, A., Ranasinghe, D.C., 2016. Effectiveness of a Batteryless and Wireless Wearable Sensor System for Identifying Bed and Chair Exits in Healthy Older People. Sensors (Basel) 16. https://doi.org/10.3390/s16040546

26. Slim, S.O., Atia, A., M.A., M., M.Mostafa, M.-S., 2019. Survey on Human Activity Recognition based on Acceleration Data. ijacsa 10. https://doi.org/10.14569/IJACSA.2019.0100311

27. Sriwichian, A., Muangprathub, J., 2019. Comparison of Algorithm Selection to Analyze Elderly Activity Recognition Based on Sensor Data Using R Program, in: 2019 16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). Presented

28. Sunkad, Z.A., Soujanya, 2016. Feature Selection and Hyperparameter Optimization of SVM for Human Activity Recognition, in: 2016 3rd International Conference on Soft Computing Machine Intelligence (ISCMI). Presented at the 2016 3rd International Conference on Soft Computing Machine Intelligence (ISCMI), pp.

29. Torres, R.L.S., Visvanathan, R., Hoskins, S., van den Hengel, A., Ranasinghe, D.C., 2016. Effectiveness of a Batteryless and Wireless Wearable Sensor System for Identifying Bed and Chair Exits in Healthy Older People. Sensors (Basel) 16. https://doi.org/10.3390/s16040546

30. UCI Machine Learning Repository: Activity recognition with healthy older people using a batteryless wearable sensor Data Set [WWW Document], n.d. URL https://archive.ics.uci.edu/ml/datasets/Activity+recognition+with+healthy+older+people+using+a+batteryless+wearable+sensor (accessed 2.23.20).

31. Vassallo, M., Amersey, R.A., Sharma, J.C., Allen, S.C., 2000. Falls on integrated medical wards. Gerontology 46, 158–162. https://doi.org/10.1159/000022152

32. Wang, J., Chen, Y., Hao, S., Peng, X., Hu, L., 2019. Deep Learning for Sensor-based Activity Recognition: A Survey. Pattern Recognition Letters 119, 3–11. https://doi.org/10.1016/j.patrec.2018.02.010

33. Xu, W., Pang, Y., Yang, Y., Liu, Y., 2018. Human Activity Recognition Based On Convolutional Neural Network, in: 2018 24th International Conference on Pattern Recognition (ICPR). Presented at the 2018 24th International Conference on Pattern Recognition (ICPR), pp. 165–170. https://doi.org/10.1109/ICPR.201

APPENDIX A: RESEARCH PLAN

Gantt Chart

Start Date	End Date	Task	Subtask	Duration(Week)
9/30/2019	10/14/2019	Background Studies	Literature Search	2
10/14/2019	11/4/2019		Literature Review	3
11/4/2019	11/18/2019	Data Set Collection	Data Set Collection	2
11/18/2019	12/2/2019	Data Preparations	Determine Scope of the Project	2
12/2/2019	12/9/2019		Upload and understanding the data set	1
12/9/2019	12/16/2019		Perform Data Cleaning Steps	1
12/16/2019	12/23/2019		Perform Exploratory data analysis	1

12/23/2019	1/6/2020		Perform Feature Selection	2
1/6/2020	1/27/2020	Model Building	Model Building using different ML Algorithm	3
1/27/2020	2/10/2020		Testing and Validating the result	2
2/10/2020	2/24/2020		Analyze the result	2
2/24/2020	3/9/2020		Comparing The Result with Different algorithm	2
3/9/2020	3/23/2020	Analyze and Evaluate the Model Performance	Identify the strongest feature based upon different model metrics	2
3/23/2020	4/6/2020	Documentations	Documentation	2
4/6/2020	4/20/2020	Buffer	Buffer	2

APPENDIX B: RESAERCH PROPOSAL

Richa_Goel_Reserach_
Proposal_Final_12_16_