Project Report

On

Human Activity Recognition Using Smartphones

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

Divya Gupta (11907192)

Section: KM055 – A06

(E-mail: divyagupta221100@gmail.com)

School of Computer Science and Engineering

Lovely Professional UniversityPhagwara, Punjab(India)

CERTIFICATE

This is to certify that **Divya Gupta** of course **Integrated Bachelor of Technology and Management in Business Administration** has successfully completed her project on topic **Human Activity Recognition Using Smartphones** as prescribed by **Dr. Sagar Pande** during the academic year **2021-22** as per the guidelines given by **School of Computer Science and Engineering**.

Signature
(Dr. Sagar Pande)

DECLARATION

I hereby solemnly declare that the report of the project work entitled **Human Activity Recognition Using Smartphones**, is prepared by me during the year 2021-22 under the guidance of **Dr. Sagar Pande**, **School of Computer Science and Engineering**, **Lovely Professional University** in the partial fulfillment of degree **Integrated Bachelor of Technology and Management in Business Administration**.

I assure that the report is based on my own work carried out during the study. I assert that the statements made, and conclusions drawn are an outcome of the project work.

Name: Divya Gupta
Registration Number: 11907192

ACKNOWLEDGEMENT

"It is not possible to prepare a project report without the assistance & encouragement of other people. This one is certainly no exception."

On the very outset of this report, I would like to extend my sincere & heartfelt obligation towards all the personages who have helped me in this endeavour. Without their active guidance, help, cooperation & encouragement, I would not have made headway in the project.

I am extremely thankful and pay my gratitude to my faculty Dr. Sagar Pande for his valuable guidance and support on completion of this project in its presently.

I extend my gratitude to Lovely Professional University for giving me this opportunity. I also acknowledge with a deep sense of reverence, my gratitude towards my parents and member of my family, who has always supported me morally as well as economically. At last, but not least gratitude goes to all my friends who directly or indirectly helped me to complete this project report.

Thanking You
Divya Gupta
(11907192)

INDEX

ABSTRACT

Human Activity Recognition database built from the recordings of 30 subjects performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors. Human activity recognition plays a significant role in medical field and in security field.

Human Activity Recognition(HAR) is classifying activity of a person using responsive sensors that are affected from human movement. Both users and capabilities(sensors) of smartphones increase and users usually carry their smartphone with them. A 3-Dimensional Smartphone sensor named accelerometer and gyroscope is used to collect time series signal, from which 26 features are generated in time and frequency domain. These facts make Human Activity Recognition more important and popular. This work focuses on recognition of human activity using smartphone sensors using different machine learning classification approaches. Data retrieved from smart phones' accelerometer and gyroscope sensors are classified to recognize human activity.

In this project I have designed a model which recognize a person's activity based on the data given by Smartphones. The activities are classified using 6 different classification model, i.e., Perceptron, Logistic Regression, Support Vector Classifier, K-Neighbors, Decision Tree, and Random Forest and the best one out of these is selected based on the accuracy and various performance measures.

We also applied various active machine learning algorithms visualizing and preprocessing of data, so to make the model more accurate and error free.

LIST OF FIGURES

LIST OF TABLES

LIST OF SCREENSHOTS

INTRODUCTION

HUMAN ACTIVITY RECOGNITION USING SMARTPHONES

Human activity recognition, or HAR, is a challenging time series classification task. It involves predicting the movement of a person based on sensor data and traditionally involves deep domain expertise and methods from signal processing to correctly engineer features from the raw data in order to fit a machine learning model.

The first work on human activity recognition dates to the late ’90s. During the last 30 years, the Human Activity Recognition (HAR) research community has been highly active proposing several methods and techniques. In recent years, significant research has been focused on experimenting with solutions that can recognize Activities of Daily Living (ADLs) from inertial signals. This is mainly due to two factors: the increasingly low cost of hardware and the wide spread of mobile devices equipped with inertial sensors. The use of smartphones to both acquire and process signals opens opportunities in a variety of application contexts such as surveillance, healthcare, and delivering.

In the context of Human Activity Recognition, most of the classification methods rely on the Activity Recognition Process (ARP) protocol. ARP consists of five steps, acquisition, pre-processing, segmentation, feature extraction, and classification.

The *data acquisition* step oversees acquiring data from sensors. Data generally originates from sensors such as accelerometers, compasses, and gyroscopes. Data acquired from sensors typically include artifacts and noise due to many reasons, such as electronic fluctuation, sensors calibration, and malfunctions. Thus, data must be processed.

The *pre-processing* step is responsible for removing artifacts and noise. Generally, pre-processing is based on filtering techniques. The output of the step is a set of filtered data that constitute the input for the next step.

The *data segmentation* step is responsible of splitting data into segments, also called windows. Data segmentation is a frequent practice which facilitates the next step.

The *feature extraction* step aims to extract the most significative portion of information from the data to be given to the classification algorithm while reducing data dimension.

The *classification* is the last step of the process. It consists in training and testing the algorithm. That is, the parameters of the classification model are estimated during the training procedure. Thereinafter, the classification performances of the model are tested in the testing procedure.

HUMAN ACTIVITY RECOGNITION USING SMARTPHONES DATASET

The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed six activities (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING) wearing a smartphone (Samsung Galaxy S II) on the waist. Using its embedded accelerometer and gyroscope, we captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually. The obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data.

The sensor signals (accelerometer and gyroscope) were pre-processed by applying noise filters and then sampled in fixed width sliding windows of 2.56 sec and 50% overlap (128 readings/window). The sensor acceleration signal, which has gravitational and body motion components, was separated using a Butterworth low-pass filter into body acceleration and gravity. The gravitational force is assumed to have only low frequency components, therefore a filter with 0.3 Hz cutoff frequency was used. From each window, a vector of features was obtained by calculating variables from the time and frequency domain.

ATTRIBUTES

The features selected for this database come from the accelerometer and gyroscope 3-axial raw signals tAcc-XYZ and tGyro-XYZ. These time domain signals (prefix 't' to denote time) were captured at a constant rate of 50 Hz. Then they were filtered using a median filter and a 3rd order low pass Butterworth filter with a corner frequency of 20 Hz to remove noise. Similarly, the acceleration signal was then separated into body and gravity acceleration signals (tBodyAcc-XYZ and tGravityAcc-XYZ) using another low pass Butterworth filter with a corner frequency of 0.3 Hz.

Subsequently, the body linear acceleration and angular velocity were derived in time to obtain Jerk signals (tBodyAccJerk-XYZ and tBodyGyroJerk-XYZ). Also, the magnitude of these three-dimensional signals was calculated using the Euclidean norm (tBodyAccMag, tGravityAccMag, tBodyAccJerkMag, tBodyGyroMag, tBodyGyroJerkMag).

Finally, a Fast Fourier Transform (FFT) was applied to some of these signals producing fBodyAcc-XYZ, fBodyAccJerk-XYZ, fBodyGyro-XYZ, fBodyAccJerkMag, fBodyGyroMag, fBodyGyroJerkMag. (Note the 'f' to indicate frequency domain signals).

These signals were used to estimate variables of the feature vector for each pattern: '-XYZ' is used to denote 3-axial signals in the X, Y and Z directions.

1. tBodyAcc-XYZ
2. tGravityAcc-XYZ
3. tBodyAccJerk-XYZ
4. tBodyGyro-XYZ
5. tBodyGyroJerk-XYZ
6. tBodyAccMag
7. tGravityAccMag
8. tBodyAccJerkMag
9. tBodyGyroMag
10. tBodyGyroJerkMag
11. fBodyAcc-XYZ
12. fBodyAccJerk-XYZ
13. fBodyGyro-XYZ
14. fBodyAccMag
15. fBodyAccJerkMag
16. fBodyGyroMag
17. fBodyGyroJerkMag

The set of variables that were estimated from these signals are:

1. mean(): Mean value
2. std(): Standard deviation
3. mad(): Median absolute deviation

4. max(): Largest value in array
5. min(): Smallest value in array
6. sma(): Signal magnitude area
7. energy(): Energy measure. Sum of the squares divided by the number of values.
8. iqr(): Interquartile range
9. entropy(): Signal entropy
10. arCoeff(): Autorregresion coefficients with Burg order equal to 4
11. correlation(): correlation coefficient between two signals
12. maxInds(): index of the frequency component with largest magnitude
13. meanFreq(): Weighted average of the frequency components to obtain a mean frequency
14. skewness(): skewness of the frequency domain signal
15. kurtosis(): kurtosis of the frequency domain signal
16. bandsEnergy(): Energy of a frequency interval within the 64 bins of the FFT of each window.
17. angle(): Angle between to vectors.

Additional vectors obtained by averaging the signals in a signal window sample. These are used on the angle() variable:

1. gravityMean
2. tBodyAccMean
3. tBodyAccJerkMean
4. tBodyGyroMean
5. tBodyGyroJerkMean

OUTPUT VARIABLE

The output variable of the given dataset is “Activity”.

Activity Description:

1. WALKING
2. WALKING_UPSTAIRS
3. WALKING_DOWNSTAIRS

4. SITTING
5. STANDING
6. LAYING

ATTRIBUTE INFORMATION

For each record in the dataset, it is provided:

- Triaxial acceleration from the accelerometer (total acceleration) and the estimated body acceleration.
- Triaxial Angular velocity from the gyroscope.
- A 561-feature vector with time and frequency domain variables.
- Its activity label.
- An identifier of the subject who carried out the experiment.
- Features are normalized and bounded within [-1,1].
- Each feature vector is a row on the text file.
- The units used for the accelerations (total and body) are 'g's (gravity of earth -> 9.80665 m/seg2).
- The gyroscope units are rad/seg.

LITERATURE REVIEW/ RELATED WORKS

[1] Smartphones are now nearly ubiquitous; their numerous built-in sensors enable continuous measurement of activities of daily living, making them especially well-suited for health research. Researchers have proposed various human activity recognition (HAR) systems aimed at translating measurements from smartphones into various types of physical activity. In this review, we summarized the existing approaches to smartphone-based HAR. For this purpose, we systematically searched Scopus, PubMed, and Web of Science for peer-reviewed articles published up to December 2020 on the use of smartphones for HAR. We extracted information on smartphone body location, sensors, and physical activity types studied and the data transformation techniques and classification schemes used for activity recognition. Consequently, we identified 108 articles and described the various approaches used for data acquisition, data preprocessing, feature extraction, and activity classification, identifying the most common practices, and their alternatives. We conclude that smartphones are well-suited for HAR research in the health sciences. For population-level impact, future studies should focus on improving the quality of collected data, address missing data, incorporate more diverse participants and activities, relax requirements about phone placement, provide more complete documentation on study participants, and share the source code of the implemented methods and algorithms.

[2] The ubiquity of smartphones and their rich set of on-board sensors has created many exciting new opportunities, where smartphones are used as powerful computing platforms to sense and analyze pervasive data. One important application of mobile sensing is activity recognition based on smartphone inertial sensors, which is a fundamental building block for a variety of scenarios, such as indoor pedestrian tracking, mobile health care, and smart cities. Although many approaches have been proposed to address the human activity recognition problem, several challenges are still present: 1) people's motion modes are very different for different individuals; 2) there is only a very limited amount of training data; 3) human activities can be arbitrary and complex, thus handcrafted feature engineering often fails to work; and 4) the recognition accuracy tends to be limited due to confusing activities. To tackle those challenges, in this paper, we propose a human activity recognition framework based on convolutional neural networks (CNNs) with two convolutional layers using the smartphone-based accelerometer, gyroscope, and magnetometer. To solve the confusion between highly similar activities like going upstairs and walking, this paper presents a novel ensemble model of CNN to further improve the

identification accuracy. The extensive experiments have been conducted using 235 977 sensory samples from a total of 100 subjects. The results have shown that the classification accuracy of the proposed model can be up to 96.11%, which proves the effectiveness of the proposed model.

[3] This paper describes how to recognize certain types of human physical activities using acceleration data generated by a user's cell phone. We propose a recognition system in which a new digital low-pass filter is designed in order to isolate the component of gravity acceleration from that of body acceleration in the raw data. The system was trained and tested in an experiment with multiple human subjects in real-world conditions. Several classifiers were tested using various statistical features. High-frequency and low-frequency components of the data were taken into account. We selected five classifiers each offering good performance for recognizing our set of activities and investigated how to combine them into an optimal set of classifiers. We found that using the average of probabilities as the fusion method could reach an overall accuracy rate of 91.15%.

[4] Human Activity Recognition(HAR) is classifying activity of a person using responsive sensors that are affected from human movement. Both users and capabilities(sensors) of smartphones increase and users usually carry their smartphone with them. These facts makes HAR more important and popular. This work focuses on recognition of human activity using smartphone sensors using different machine learning classification approaches. Data retrieved from smart phones' accelerometer and gyroscope sensors are classified in order to recognise human activity. Results of the approaches used are compared in terms of efficiency and precision.

[5] Recognizing human activities and monitoring population behavior are fundamental needs of our society. Population security, crowd surveillance, healthcare support and living assistance, and lifestyle and behavior tracking are some of the main applications that require the recognition of human activities. Over the past few decades, researchers have investigated techniques that can automatically recognize human activities. This line of research is commonly known as Human Activity Recognition (HAR). HAR involves many tasks: from signals acquisition to activity classification. The tasks involved are not simple and often require dedicated hardware, sophisticated engineering, and computational and statistical techniques for data preprocessing and analysis. Over the years, different techniques have been tested and different solutions have been proposed to achieve a classification process that provides reliable results. This survey presents the most recent solutions proposed for each task in the human activity classification

process, that is, acquisition, preprocessing, data segmentation, feature extraction, and classification. Solutions are analyzed by emphasizing their strengths and weaknesses. For completeness, the survey also presents the metrics commonly used to evaluate the goodness of a classifier and the datasets of inertial signals from smartphones that are mostly used in the evaluation phase.

[6] Smartphones, smartwatches, fitness trackers, and ad-hoc wearable devices are being increasingly used to monitor human activities. Data acquired by the hosted sensors are usually processed by machine-learning-based algorithms to classify human activities. The success of those algorithms mostly depends on the availability of training (labeled) data that, if made publicly available, would allow researchers to make objective comparisons between techniques. Nowadays, there are only a few publicly available data sets, which often contain samples from subjects with too similar characteristics, and very often lack specific information so that is not possible to select subsets of samples according to specific criteria. In this article, we present a new dataset of acceleration samples acquired with an Android smartphone designed for human activity recognition and fall detection. The dataset includes 11,771 samples of both human activities and falls performed by 30 subjects of ages ranging from 18 to 60 years. Samples are divided in 17 fine grained classes grouped in two coarse grained classes: one containing samples of 9 types of activities of daily living (ADL) and the other containing samples of 8 types of falls. The dataset has been stored to include all the information useful to select samples according to different criteria, such as the type of ADL performed, the age, the gender, and so on. Finally, the dataset has been benchmarked with four different classifiers and with two different feature vectors. We evaluated four different classification tasks: fall vs. no fall, 9 activities, 8 falls, 17 activities and falls. For each classification task, we performed a 5-fold cross-validation (i.e., including samples from all the subjects in both the training and the test dataset) and a leave-one-subject-out cross-validation (i.e., the test data include the samples of a subject only, and the training data, the samples of all the other subjects). Regarding the classification tasks, the major findings can be summarized as follows: (i) it is quite easy to distinguish between falls and ADLs, regardless of the classifier and the feature vector selected. Indeed, these classes of activities present quite different acceleration shapes that simplify the recognition task; (ii) on average, it is more difficult to distinguish between types of falls than between types of activities, regardless of the classifier and the feature vector selected. This is due to the similarity between the acceleration shapes of different kinds of falls. On the contrary, ADLs acceleration shapes present differences except for a small group. Finally, the evaluation shows that the presence of samples

of the same subject both in the training and in the test datasets, increases the performance of the classifiers regardless of the feature vector used. This happens because each human subject differs from other subjects in performing activities even if she shares with them the same physical characteristics.

[7] Mobile phones are equipped with a rich set of sensors, such as accelerometers, magnetometers, gyroscopes, photometers, orientation sensors, and gravity sensors. These sensors can be used for human activity recognition in the ubiquitous computing domain. Most of reported studies consider acceleration signals that are collected from a known fixed device location and orientation. This paper describes how more accurate results of basic activity recognition can be achieved with transformed accelerometer data. Based on the rotation matrix (Euler Angle Conversion) derived from the orientation angles of gyroscopes and orientation sensors, we transform input signals into a reference coordinate system. The advantage of the transformation is that it allows activity classification and recognition to be carried out independent of the orientation of sensors. We consider five user activities: staying, walking, running, ascending stairs, and descending stairs, with a phone being placed in the subject's hand, or in pants pocket, or in a handbag. The results show that an overall orientation independent accuracy of 84.77% is achieved, which is a improvement of 17.26% over those classifications without input transformation.

[8] Activity classification in smartphones helps us to monitor and analyze the physical activities of the user in daily life and has potential applications in healthcare systems. This paper proposes a descriptor-based approach for activity classification using built-in sensors of smartphones. Accelerometer and gyroscope sensor signals are acquired to identify the activities performed by the user. In addition, time and frequency domain signals are derived using the collected signals. In the proposed approach, two descriptors, namely, histogram of gradient and centroid signature-based Fourier descriptor, are employed to extract feature sets from these signals. Feature and score level fusion are explored for information fusion. For classification, we have studied the performance of multiclass support vector machine and k-nearest neighbor classifiers. The proposed approach is evaluated on two publicly available data sets, namely, UCI HAR data set and physical activity sensor data. Our experimental results show that the feature level fusion provides better performance than the score level fusion. In addition, our approach provides considerable improvement in classifying different activities as compared with the existing works. The average activity classification accuracy achieved using the proposed method is 97.12% as

against the existing work, which provided 96.33% on UCI HAR data set. On the second data set, the proposed approach attained 96.83% classification accuracy, whereas the existing work achieved 90.2%.

[9] The proliferation of smartphones has significantly facilitated people's daily life, and diverse and powerful embedded sensors make smartphone a ubiquitous platform to acquire and analyze data, which may also provide great potential for efficient human activity recognition. This paper presents a systematic performance analysis of motion-sensor behavior for human activity recognition via smartphones. Sensory data sequences are collected via smartphones, when participants perform typical and daily human activities. A cycle detection algorithm is applied to segment the data sequence for obtaining the activity unit, which is then characterized by time-, frequency-, and wavelet-domain features. Then both personalized and generalized model using diverse classification algorithms are developed and implemented to perform activity recognition. Analyses are conducted using 27 681 sensory samples from 10 subjects, and the performance is measured in the form of F-score under various placement settings, and in terms of sensitivity to user space, stability to combination of motion sensors, and impact of data imbalance. Extensive results show that each individual has its own specific and discriminative movement patterns, and the F-score for personalized model and generalized model can reach 95.95% and 96.26%, respectively, which indicates our approach is accurate and efficient for practical implementation.

[10] Smartphone based human activity monitoring and recognition play an important role in several medical applications, such as eldercare, diabetic patient monitoring, post-trauma recovery after surgery. However, it is more important to recognize the activity sequences in terms of transitions. In this work, we have designed a detailed activity transition recognition framework that can identify a set of activity transitions and their sequence for a time window. This enables us to extract more meaningful insight about the subject’s physical and behavioral context. However, precise labeling of training data for detailed activity transitions at every time instance is required for this purpose. But, due to non uniformity of individual gait, the labeling tends to be error prone. Accordingly, our contribution in this work is to formulate the activity transition detection problem as a multiple instance learning problem to deal with imprecise labeling of data. The proposed human activity transition recognition framework forms an ensemble model based on different MIML-kNN distance metrics. The ensemble model helps to find both the activity sequence as well as multiple activity transition. The framework is

implemented for a real dataset collected from 8 users. It is found to be working adequately (average precision 0.94).

[11] Smartphones have emerged as a promising type of equipment for monitoring human activities in environmental health studies. However, degraded location accuracy and inconsistency of smartphone-measured GPS data have limited its effectiveness for classifying human activity patterns. This study proposes a fuzzy classification scheme for differentiating human activity patterns from smartphone-collected GPS data. Specifically, a fuzzy logic reasoning was adopted to overcome the influence of location uncertainty by estimating the probability of different activity types for single GPS points. Based on that approach, a segment aggregation method was developed to infer activity patterns, while adjusting for uncertainties of point attributes. Validations of the proposed methods were carried out based on a convenient sample of three subjects with different types of smartphones. The results indicate desirable accuracy (e.g., up to 96% in activity identification) with use of this method. Two examples were provided in the appendix to illustrate how the proposed methods could be applied in environmental health studies. Researchers could tailor this scheme to fit a variety of research topics.

[12] Human activity recognition (HAR) through sensors embedded in smartphones has allowed for the development of systems that are capable of detecting and monitoring human behavior. However, such systems have been affected by the high consumption of computational resources (e.g., memory and processing) needed to effectively recognize activities. In addition, existing HAR systems are mostly based on supervised classification techniques, in which the feature extraction process is done manually, and depends on the knowledge of a specialist. To overcome these limitations, this paper proposes a new method for recognizing human activities based on symbolic representation algorithms. The method, called “Multivariate Bag-Of-SFA-Symbols” (MBOSS), aims to increase the efficiency of HAR systems and maintain accuracy levels similar to those of conventional systems based on time and frequency domain features. The experiments conducted on three public datasets showed that MBOSS performed the best in terms of accuracy, processing time, and memory consumption.

[13] Recently, a significant amount of literature concerning machine learning techniques has focused on automatic recognition of activities performed by people. The main reason for this considerable interest is the increasing availability of devices able to acquire signals which, if properly processed, can provide information about human activities of daily living (ADL). The

recognition of human activities is generally performed by machine learning techniques that process signals from wearable sensors and/or cameras appropriately arranged in the environment. Whatever the type of sensor, activities performed by human beings have a strong subjective characteristic that is related to different factors, such as age, gender, weight, height, physical abilities, and lifestyle. Personalization models have been studied to take into account these subjective factors and it has been demonstrated that using these models, the accuracy of machine learning algorithms can be improved. In this work we focus on the recognition of human activities using signals acquired by the accelerometer embedded in a smartphone. The contributions of this research are mainly three. A first contribution is the definition of a clear validation model that takes into account the problem of personalization and which thus makes it possible to objectively evaluate the performances of machine learning algorithms. A second contribution is the evaluation, on three different public datasets, of a personalization model which considers two aspects: the similarity between people related to physical aspects (age, weight, and height) and similarity related to intrinsic characteristics of the signals produced by these people when performing activities. A third and last contribution is the development of a personalization model that considers both the physical and signal similarities. The experiments show that the employment of personalization models improves, on average, the accuracy, thus confirming the soundness of the approach and paving the way for future investigations on this topic.

OVERVIEW OF THE PROJECT

OBJECTIVE:

- The main objective of the project is to recognize six daily life activity i.e., Walking, Walking Upstairs, Walking Downstairs, Sitting, Standing and Laying.
- The secondary objective is to choose best suitable Machine Learning model among the selected one based on accuracy.

OPPORTUNITY:

- It can be implemented in medical science to recognize and track the daily activity of an individual.
- It can be used in human survey system by recording the tasks performed by the participants.

PROBLEM:

- The common problem is that the use of smartphones, makes it possible to follow large numbers of individuals over a long period of time, but at the same time there is need to consider approaches to missing sensor data.
- The other problem that arises is that collection of data using smartphones undoubtedly raises concerns about individual’s privacy.

GOAL:

- The goal is to improve Human Activity Recognition using Smartphones by adding more activities like running, riding, driving, falling, etc. and then choosing the best fitting model.
- Another goal is to perform the Human Activity Recognition using Smartphones with less computation time and more precisely.

CLASSIFIER MODELS:

- Perceptron

  It is a supervised learning algorithm of binary classifiers. A single neuron, the perceptron model detects whether any function is an input or not and classifies them in either of the classes.

- Logistic Regression

  It is a supervised algorithm that can be used to model the probability of a certain class or event for binary classification problem. It is used when the data is linearly separable, and the outcome is binary or dichotomous in nature.

- Support Vector Classifier

  It is used to fit the data provided, returning a "best fit" hyperplane that divides, or categorizes, the data. From there, after getting the hyperplane, one can feed some features to classifier to see what is the "predicted" class.

- K- Nearest Neighbor

  It is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. It's easy to implement and understand but has a major drawback of becoming significantly slows as the size of that data in use grows. It classifies new data points based on the nearest data points.

- Decision Tree

  It is a type of supervised machine learning used to categorize or make predictions based on how a previous set of questions were answered. The model is trained and tested on a set of data that contains the desired categorization.

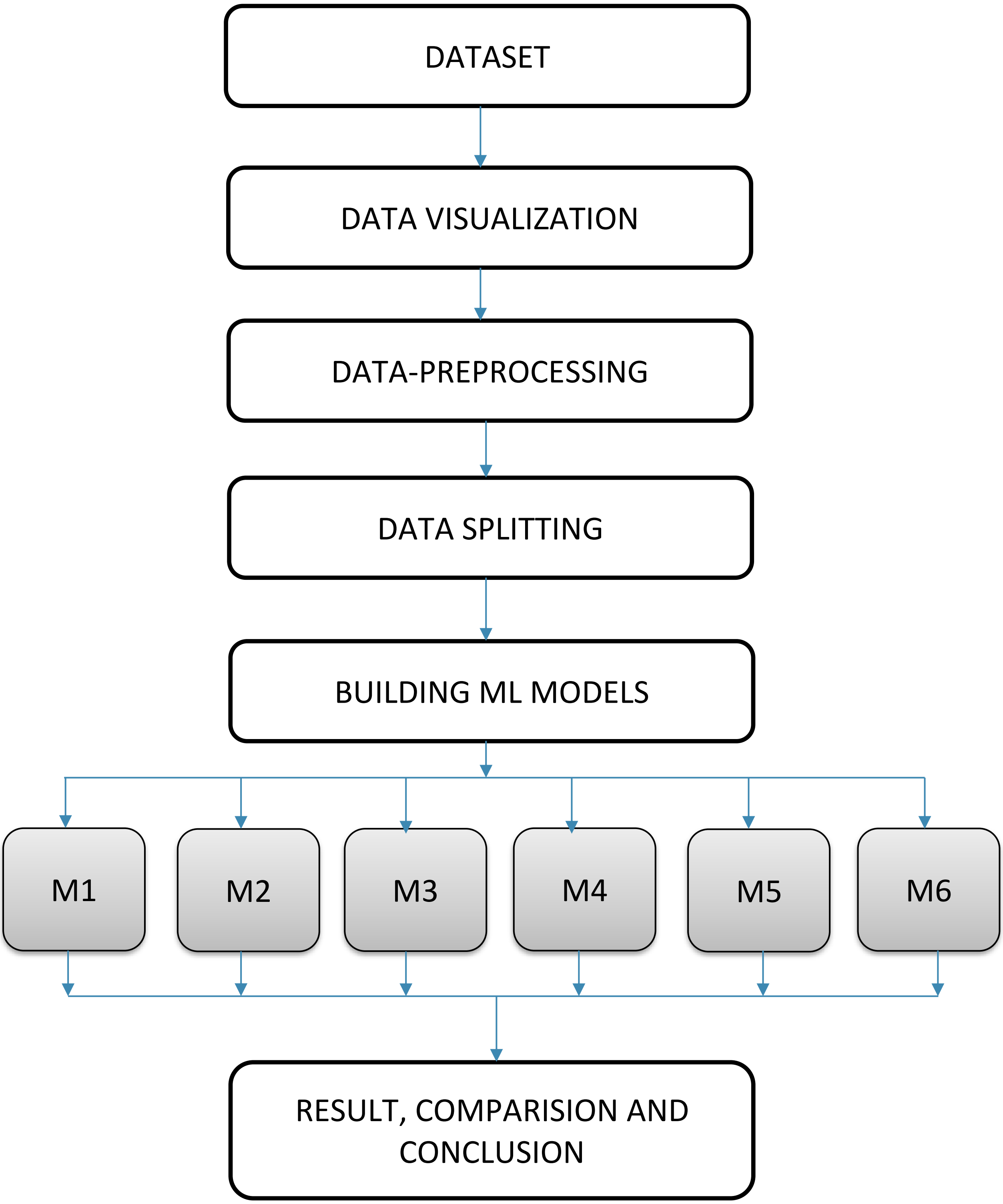
- Random Forest

  It is a supervised machine learning algorithm made up of decision trees. It is used for both classification and regression. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

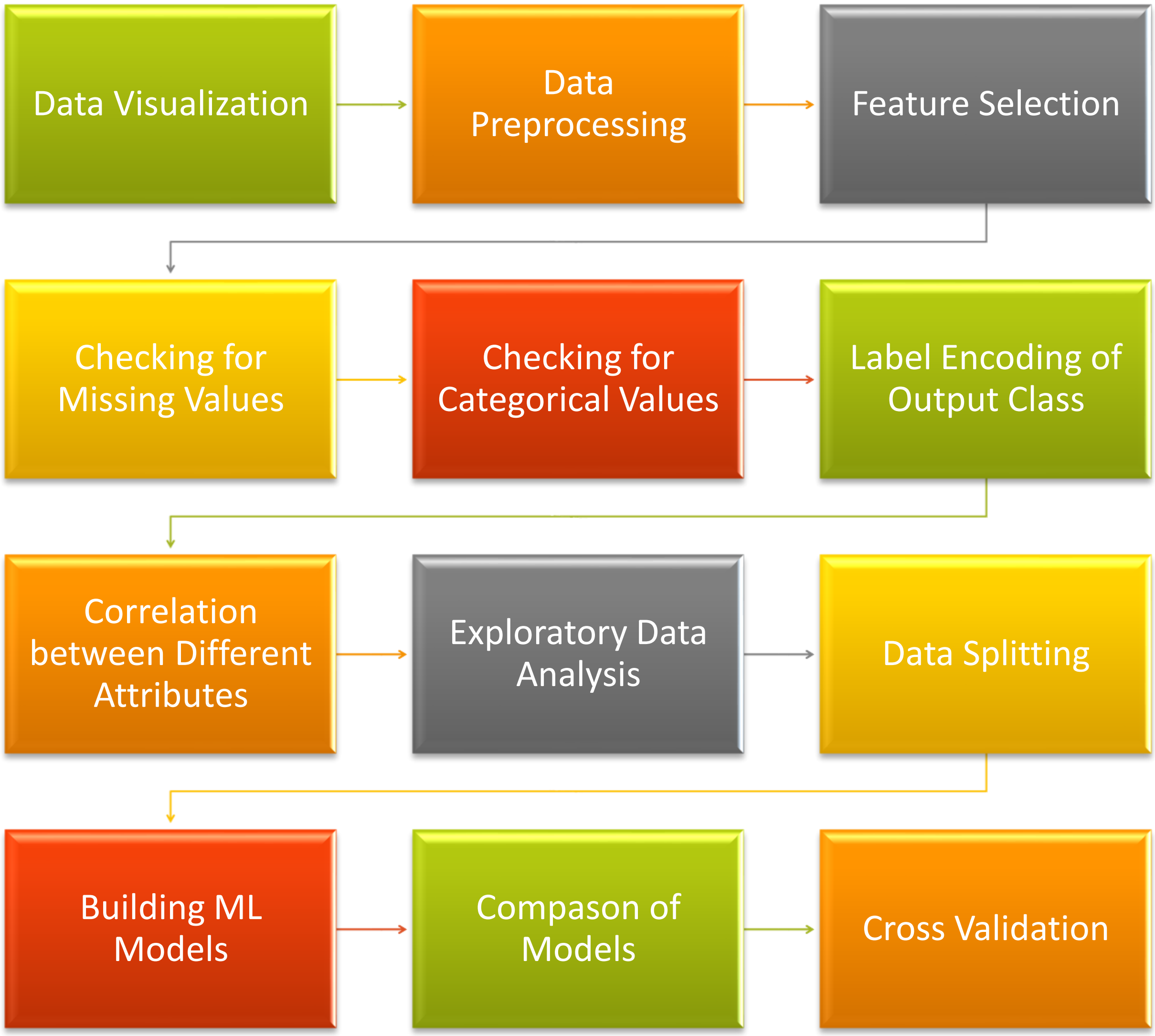
DATASET LINKS

- UCI – https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones#
- Kaggle – https://www.kaggle.com/datasets/devyansh30gupta/human-activity-recognition-using-smartphone-data

METHODOLOGY



CIRCUIT DESCRIPTION



DATASET

The data being used contains total Ten Thousand Two Hundred and Ninety-Nine (10299) samples/instances.
The dataset is described under Five Hundred and Sixty-Two (562) columns as follows:

1. tBodyAcc-XYZ
2. tGravityAcc-XYZ
3. tBodyAccJerk-XYZ
4. tBodyGyro-XYZ
5. tBodyGyroJerk-XYZ
6. tBodyAccMag
7. tGravityAccMag
8. tBodyAccJerkMag
9. tBodyGyroMag
10. tBodyGyroJerkMag
11. fBodyAcc-XYZ
12. fBodyAccJerk-XYZ
13. fBodyGyro-XYZ
14. fBodyAccMag
15. fBodyAccJerkMag
16. fBodyGyroMag
17. fBodyGyroJerkMag

The set of variables that were estimated from these signals are:

1. mean(): Mean value
2. std(): Standard deviation
3. mad(): Median absolute deviation
4. max(): Largest value in array
5. min(): Smallest value in array
6. sma(): Signal magnitude area
7. energy(): Energy measure. Sum of the squares divided by the number of values.
8. iqr(): Interquartile range
9. entropy(): Signal entropy

10. arCoeff(): Autorregresion coefficients with Burg order equal to 4
11. correlation(): correlation coefficient between two signals
12. maxInds(): index of the frequency component with largest magnitude
13. meanFreq(): Weighted average of the frequency components to obtain a mean frequency
14. skewness(): skewness of the frequency domain signal
15. kurtosis(): kurtosis of the frequency domain signal
16. bandsEnergy(): Energy of a frequency interval within the 64 bins of the FFT of each window.
17. angle(): Angle between to vectors.

Additional vectors obtained by averaging the signals in a signal window sample. These are used on the angle() variable:

1. gravityMean
2. tBodyAccMean
3. tBodyAccJerkMean
4. tBodyGyroMean
5. tBodyGyroJerkMean

CODE DESCRIPTION

Data Visualization and Data Preprocessing

"All the results below are drawn using python code and same can be traced from the code file as well."

A brief study using different data visualization techniques and python to understand the nature of the data samples and attributes which will further help us in data preprocessing and model selection.

- Shape of the dataset (10299, 562)
- 10299 Rows
- 562 Columns (Including 561 Features and 1 Output)

Since the number of features is 561, so we will apply feature selection to reduce the number of features.

```
[ ]  #Feature Selection
     from sklearn.feature_selection import SelectFromModel
     from sklearn.ensemble import RandomForestClassifier
     model = SelectFromModel(RandomForestClassifier(n_estimators=100,max_depth=5),
                             prefit=False,max_features=10)
     model.fit(x,y)
     s = model.transform(x)
     print(x.shape)
     print(s.shape)

     (10299, 561)
     (10299, 10)
```

After feature selection the shape of dataset is (10299,11).

- 10299 Rows
- 11 Columns (Including 10 Features and 1 Output)

Checking For Missing Values

Now I have checked if there is any missing data, if 'YES' then update it using different methods, if 'NO' then move to next stage.

```
[ ]  #Checking for the Missing Values
     print(df_new.isna().sum())

     F1          0
     F2          0
     F3          0
     F4          0
     F5          0
     F6          0
     F7          0
     F8          0
     F9          0
     F10         0
     Activity    0
     dtype: int64
```

There are no missing values present in our dataset. So, we are good to move forward.

Checking For Categorical Values

We have to check the nature of the attributes present. Whether they are Categorical or Numerical. If there is any categorical data, we need to convert it into numerical form with the help of different encoding techniques present, since we cannot perform mathematical operations on the categorical data. If all the values are numerical then we are good to go.

```
[ ]  #Checking for Categorical Data
     df_new.info()

     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 10299 entries, 0 to 10298
     Data columns (total 11 columns):
      #   Column    Non-Null Count  Dtype
     ---  ------    --------------  -----
      0   F1        10299 non-null  float64
      1   F2        10299 non-null  float64
      2   F3        10299 non-null  float64
      3   F4        10299 non-null  float64
      4   F5        10299 non-null  float64
      5   F6        10299 non-null  float64
      6   F7        10299 non-null  float64
      7   F8        10299 non-null  float64
      8   F9        10299 non-null  float64
      9   F10       10299 non-null  float64
      10  Activity  10299 non-null  object
     dtypes: float64(10), object(1)
     memory usage: 885.2+ KB
```

The above illustration is generated with the help of python, and it describes the datatype of each attribute.

We can say that there is no categorical data in our dataset except for the “Activity” column, so we need to convert it into numerical data.

```
[ ]  #Label Encoding for Activity
     from sklearn.preprocessing import LabelEncoder
     le = LabelEncoder()
     df_new['Activity'] = le.fit_transform(df_new[["Activity"]])
```

```
Output        According to dataset description
  0                 Laying
  2                 Standing
  1                 Sitting
  3                 Walking
  5                 Walking_Upstairs
  4                 Walking_Downstairs
```

After applying Label Encoder to the “Activity” the categorical data has been converted into numerical data.

Since now all the features are in numerical form, so we are good to move forward.

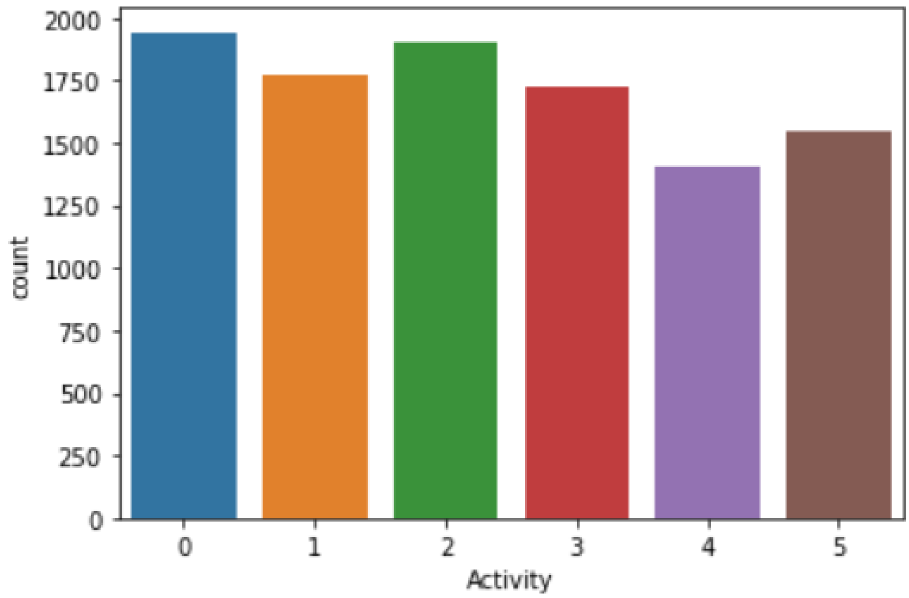
Data Visualization of the Output Class (Activity)

- Unique values in the output class
  a. 0
  b. 1
  c. 2
  d. 3
  e. 4
  f. 5

- Unique Value Counts in the dataset

```
[ ]  #Value Count for Activity
     df_new['Activity'].value_counts()

     0    1944
     2    1906
     1    1777
     3    1722
     5    1544
     4    1406
     Name: Activity, dtype: int64
```

- Graphical Representation of the above data



From the above I can conclude that the different data samples present in the dataset are being mapped to six different output classes [0,1,2,3,4,5]. Hence, we can conclude it to be a classification problem.

Correlation between different Attributes

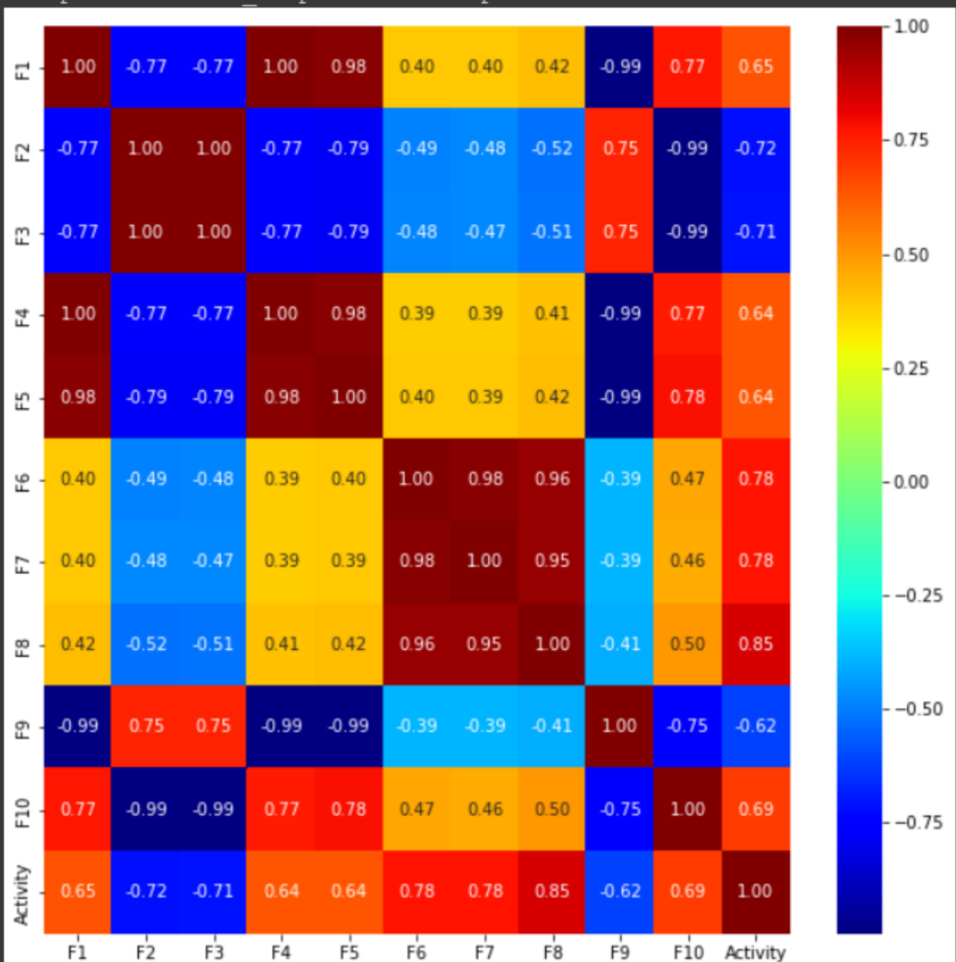
```
#Correlation between Attributes
cor=df_new.corr()
cor
```

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Activity
F1	1.000000	-0.770334	-0.773641	0.999021	0.983816	0.400207	0.396284	0.421894	-0.989280	0.768681	0.646631
F2	-0.770334	1.000000	0.997824	-0.767588	-0.785638	-0.490322	-0.478870	-0.523040	0.747495	-0.993425	-0.716847
F3	-0.773641	0.997824	1.000000	-0.772431	-0.787474	-0.483983	-0.472388	-0.514275	0.750624	-0.991255	-0.710595
F4	0.999021	-0.767588	-0.772431	1.000000	0.983074	0.391809	0.387936	0.411766	-0.988757	0.766217	0.639321
F5	0.983816	-0.785638	-0.787474	0.983074	1.000000	0.395942	0.393757	0.418393	-0.990372	0.784947	0.636015
F6	0.400207	-0.490322	-0.483983	0.391809	0.395942	1.000000	0.982802	0.960428	-0.387936	0.468721	0.775629
F7	0.396284	-0.478870	-0.472388	0.387936	0.393757	0.982802	1.000000	0.954178	-0.388229	0.458425	0.777133
F8	0.421894	-0.523040	-0.514275	0.411766	0.418393	0.960428	0.954178	1.000000	-0.406650	0.499311	0.845715
F9	-0.989280	0.747495	0.750624	-0.988757	-0.990372	-0.387936	-0.388229	-0.406650	1.000000	-0.748249	-0.615424
F10	0.768681	-0.993425	-0.991255	0.766217	0.784947	0.468721	0.458425	0.499311	-0.748249	1.000000	0.692325
Activity	0.646631	-0.716847	-0.710595	0.639321	0.636015	0.775629	0.777133	0.845715	-0.615424	0.692325	1.000000

- The above result is found using python can be traced same from the code file.
- Positive correlation is a relationship between two variables in which both variables move in tandem—that is, in the same direction. A positive correlation exists when one variable decreases as the other variable decreases, or one variable increase while the other increases.
- While a negative correlation describes the relationship between two variables which change in opposing directions. A negative correlation exists when one variable decreases as the other variable increases, or one variable increase while the other decreases.
- To better understand the correlation between different attributes, a heat map using python. Same result can be traced from the code file also.

```
#Heatmap for better visualizing the correlation
plt.figure(figsize=(10,10))
sns.heatmap(cor,annot=True,fmt='.2f',cmap='jet',annot_kws={'size':10},cbar=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fde67b09f10>
```

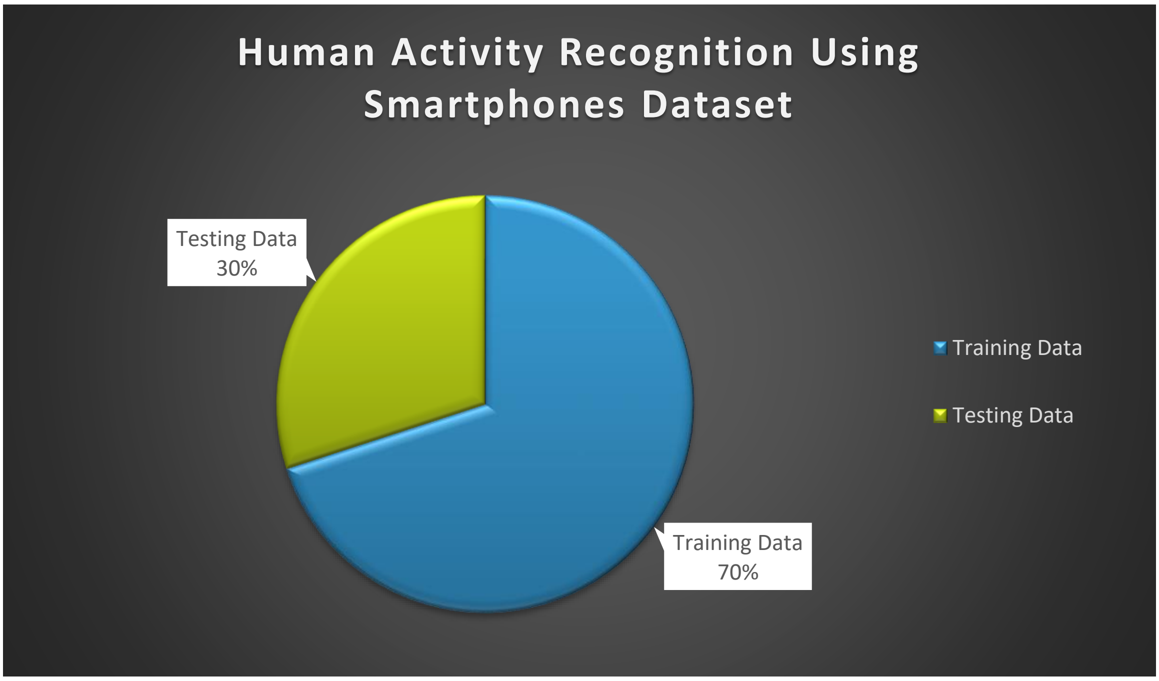


- The color bar in the above illustration demonstrates a color range to correlation coefficient.
- The more the intensity of the color the more is the correlation coefficient.

Data Splitting

We need to split the dataset into two sets as training data and testing data.
The dataset is divided into two sets as

- 70% Training Data -- will be used for training the model.
- 30% Testing Data -- will be used for testing the trained model.



 # Training Data

- To train the model we need to separate out our train data into two sets
- x_train = contains the features only
- y_train = contains the output only
- Both x_train and y_train are respective to each other
- In order to bring all the features at the same scale we need to apply feature scaling on x_train.

```
#Inorder to bring all the features at the same scale we need to apply feature scaling on x_train
from sklearn.preprocessing import StandardScaler
ss = StandardScaler()
x_train = pd.DataFrame(ss.fit_transform(x_train),columns=x_train.columns)
print("After Feature Scaling \n",x_train.head())
```

 # Testing Data

- To test the model, we need to separate out our test data into two sets.
- x_test = contains the features only.
- y_test = contains the output only.
- Both x_test and y_test is respective to each other.
- In order to bring all the features at the same scale we need to apply feature scaling on x_test

```
#Inorder to bring all the features at the same scale we need to apply feature scaling on x_test
from sklearn.preprocessing import StandardScaler
ss = StandardScaler()
x_test = pd.DataFrame(ss.fit_transform(x_test),columns=x_test.columns)
print("After Feature Scaling \n",x_test.head())
```

Now, we are done with Data Splitting, so we are good to move forward to Building ML Models.

Building ML Models

Model 1 – Perceptron

1. Imported Perceptron Classifier from sklearn.linear_model.
2. Created a new instance of Perceptron Classifier.
3. Trained the new instance using the x_train.
4. Predicted output class on x_train and x_test to calculate the accuracy and other performance measuring factors, like confusion matrix.

Model 2 – Logistic Regression Classifier

1. Imported Logistic Regression Classifier from sklearn.linear_model.
2. Created a new instance of Logistic Regression Classifier.
3. Trained the new instance using the x_train.
4. Predicted output class on x_train and x_test to calculate the accuracy and other performance measuring factors, like confusion matrix.

Model 3 – SVC

1. Imported Logistic Regression Classifier from sklearn.svm.
2. Created a new instance of SVC.
3. Trained the new instance using the x_train.
4. Predicted output class on x_train and x_test to calculate the accuracy and other performance measuring factors, like confusion matrix.

Model 4 – K-Neighbors Classifier

1. Imported Logistic Regression Classifier from sklearn.neighbors.
2. Created a new instance of K-Neighbors Classifier.
3. Trained the new instance using the x_train.
4. Predicted output class on x_train and x_test to calculate the accuracy and other performance measuring factors, like confusion matrix.

Model 5 – Decision Tree Classifier

1. Imported Decision Tree Classifier from sklearn.tree.
2. Created a new instance of Decision Tree Classifier.

3. Trained the new instance using the x_train.
4. Predicted output class on x_train and x_test to calculate the accuracy and other performance measuring factors, like confusion matrix.

Model 6 – Random Forest Classifier

1. Imported Random Forest Classifier from sklearn.ensemble.
2. Created a new instance of Random Forest Classifier.
3. Trained the new instance using the x_train.
4. Predicted output class on x_train and x_test to calculate the accuracy and other performance measuring factors, like confusion matrix.

RESULT

Model 1 – Perceptron

- Training Perceptron Model -> Time taken : 43.5 milliseconds
- Perceptron Model Training Accuracy : 0.800943265362741
- Perceptron Model Testing Accuracy : 0.7980582524271844

Model 2 – Logistic Regression Classifier

- Training Logistic Regression Classifier Model -> Time taken : 1440.4 milliseconds
- Logistic Regression Classifier Model Training Accuracy : 0.824663614925787
- Logistic Regression Classifier Model Testing Accuracy : 0.8200647249190939

Model 3 – SVC

- Training SVC Model -> Time taken : 707.1 milliseconds
- SVC Model Training Accuracy : 0.8199472881120822
- SVC Model Testing Accuracy : 0.8158576051779936

Model 4 – K-Neighbors Classifier

- Training K-Neighbors Classifier Model -> Time taken : 14.6 milliseconds
- K-Neighbors Classifier Model Training Accuracy : 0.922735469551949
- K-Neighbors Classifier Model Testing Accuracy : 0.8566343042071197

Model 5 – Decision Tree Classifier

- Training Decision Tree Classifier Model -> Time taken : 85.4 milliseconds
- Decision Tree Classifier Model Training Accuracy : 0.9993064225273963
- Decision Tree Classifier Model Testing Accuracy : 0.8724919093851132

Model 6 – Random Forest Classifier

- Training Random Forest Classifier -> Time taken : 1769.6 milliseconds
- Random Forest Classifier Model Training Accuracy : 1.0
- Random Forest Classifier Model Testing Accuracy : 0.9103559870550162

COMPARISON

Now, I have compared the accuracy between selected Classifier Models so that the result can be evaluated, hence reaching the conclusion.

```
from sklearn.linear_model import Perceptron
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

clf1 = Perceptron()
clf2 = LogisticRegression(max_iter=20000)
clf3 = SVC()
clf4 = KNeighborsClassifier()
clf5 = DecisionTreeClassifier()
clf6 = RandomForestClassifier(n_estimators=100,max_depth=21)

clf=[clf1,clf2,clf3,clf4,clf5,clf6]
clf_names=['Perc','LR','SVM','KNN','DT','RF']

test={}
T={}
for model,names in zip(clf,clf_names):
  st=time.time()
  model.fit(x_train,y_train)
  pred=model.predict(x_test)
  et=time.time()
  acc=accuracy_score(pred,y_test)
  test[names]=np.round(acc*100,1)
  T[names]=np.round((et-st)*1000,1)
for i in test.keys():
  print(i,"-",test[i],"-",T[i])
```

For the purpose of ease, I have constructed a table mentioning the output of ML Model for reviewing the result.

Classifier Model	**Training Time (milliseconds)**	**Training Accuracy**	**Testing Accuracy**
Perceptron	43.5	0.800943265362741	0.7980582524271844
Logistic Regression	1440.4	0.824663614925787	0.8200647249190939
SVC	707.1	0.8199472881120822	0.8158576051779936
K-Neighbors	14.6	0.922735469551949	0.8566343042071197
Decision Tree	85.4	0.9993064225273963	0.8724919093851132
Random Forest	1769.6	1.0	0.9103559870550162

- K-Neighbors Classifier took least training time.
- Random Forest Classifier took the highest training time.
- Perceptron Classifier results least training and testing accuracy.
- Random Forest Classifier results highest training and testing accuracy.

OUTPUT

Since Random Forest Classifier is giving us the highest testing accuracy, we will take it as the optimal classifier, for our human activity recognition using smartphone dataset.

Random Forest Classifier : Training : Confusion Matrix

```
Random Forest Classifier Training Confusion Matrix
 [[1367    0    0    0    0    0]
 [   0 1220    0    0    0    0]
 [   0    0 1323    0    0    0]
 [   0    0    0 1228    0    0]
 [   0    0    0    0  995    0]
 [   0    0    0    0    0 1076]]
```

Random Forest Classifier : Training : Classification Report

```
Random Forest Classifier Training Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1367
           1       1.00      1.00      1.00      1220
           2       1.00      1.00      1.00      1323
           3       1.00      1.00      1.00      1228
           4       1.00      1.00      1.00       995
           5       1.00      1.00      1.00      1076

    accuracy                           1.00      7209
   macro avg       1.00      1.00      1.00      7209
weighted avg       1.00      1.00      1.00      7209
```

Random Forest Classifier : Testing : Confusion Matrix

```
Random Forest Classifier Testing Confusion Matrix
 [[577   0   0   0   0   0]
 [  0 493  64   0   0   0]
 [  0  45 538   0   0   0]
 [  0   0   0 432  33  29]
 [  0   0   0  43 344  24]
 [  0   0   0  18  21 429]]
```

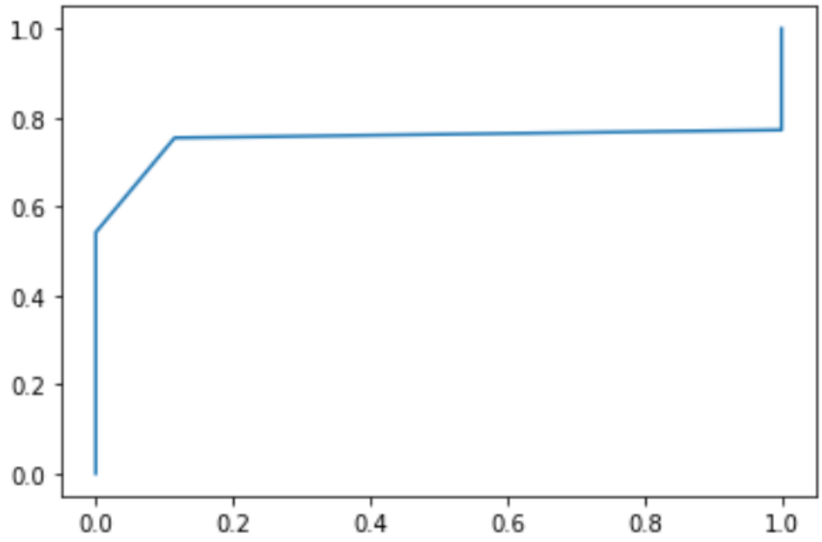
__Random Forest Classifier : Testing : Classification Report__

```
Random Forest Classifier Testing Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       577
           1       0.89      0.92      0.90       538
           2       0.92      0.89      0.91       602
           3       0.87      0.88      0.88       493
           4       0.84      0.86      0.85       398
           5       0.92      0.89      0.90       482

    accuracy                           0.91      3090
   macro avg       0.91      0.91      0.91      3090
weighted avg       0.91      0.91      0.91      3090
```

Random Forest Classifier : ROC & AUC

ROC (Receiver Operator Characteristics Curve) :

Plot between True Positive (TP) as y-axis and False Positive (FP) as x-axis.’



AUC (Area Under Curve) :

AUC= 0.7501011779164932

HARDWARE AND SOFTWARE SETUP

- To perform this experiment, I am using a 64-bit windows laptop, intel i7 6th generation.
- The software I am using is Google Colab. It is a software service provided by Google.

  Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

  a. Zero configuration required
  b. Free access to GPUs
  c. Easy sharing

  Whether you're a student, a data scientist, or an AI researcher, Colab can make our work easier.

CONCLUSION

Human activity recognition has broad application in medical research and human survey system. Through this project we can recognize overall 6 activities (Standing, Sitting, Walking, Walking Upstairs, Walking Downstairs, and Laying). The system collected time series signals using a built-in accelerometer, generated 26 features in both time and frequency domain. The activity data is trained and tested with six classifiers which are: Perceptron, Logistic Regression, SVC, K-Neighbors, Decision Tree, and Random Forest.

The data for accuracy for each classifier is as follows:

Model	**Accuracy (%)**
Perceptron	79.8%
Logistic Regression	82.0%
SVC	81.6%
K-Neighbors	85.7%
Decision Tree	80.4%
Random Forest	91.5%

The best classification rate in our experiment is 91.5% attained by Random Forest (Optimal classifier for the experiment).

The second-best classification rate in our experiment is 85.7% which is attained by K-Neighbors.

Perceptron is having the least classification rate, i.e., 79.8%.

Hence, it can be concluded that “Random Forest Classifier” is the optimal classification model for training the “Human Activity Recognition Using Smartphone” experiment.

FUTURE SCOPE

New and improved features that can better represent the peculiar traits of the human activities are needed: ideally, they would combine the domain knowledge of the experts given in the hand-crafted features and the lack of bias provided by the automatically generated features. Finally, regarding the Classification phase, we highlighted how Model-Driven approaches are being replaced by Data-Driven approaches as they are usually better performing.

Among the Data-Driven approaches, we find that both traditional ML approaches and more modern DL techniques can be applied to Human Activity Recognition problems. Specifically, we learned that while DL methods outperform traditional ML most of the time and can automatically extract the features, they require significant amounts of computational power and data more than traditional ML techniques, which makes the latter still a good fit for many use cases.

Regardless of the classification method, we discussed how Population Diversity may impact the performances of Human Activity Recognition applications. To alleviate this problem, we mentioned some contemporary trends regarding the personalization of the models. Personalizing a classification model means identifying only a portion of the population that is like the current subject under some perspective and then only use this subset to train the classifier. The resulting model should be better fitting to the end user. This, however, may exacerbate the issue of data scarcity, since only small portions of the full datasets may be used to train the model for that specific user.

For solving this issue, more large-scale data collection campaigns are needed, as well as further studies in the field of dataset combination and pre-processing pipelines to effectively combine and reduce differences among data acquired from various sources.

We can identify activities, classify, or group participants to activities, get additional insights of activity durations and patterns of individuals involved in those activities. One can exploit the rich scope such insights have to offer in developing real time human asset monitoring in highly secured installations, tracking Elderly or population with movement disability or illness for any emergencies based on movement patterns, determining if a person id under fatigue or not and so on and so forth. The application domains are as broad as from healthcare to security services and fitness monitoring.

REFERENCES

[1] Straczkiewicz, M., James, P. & Onnela, JP. A systematic review of smartphone-based human activity recognition methods for health research. *npj Digit. Med.* **4,** 148 (2021). https://doi.org/10.1038/s41746-021-00514-4.

[2] R. Zhu et al., "Efficient Human Activity Recognition Solving the Confusing Activities Via Deep Ensemble Learning," in IEEE Access, vol. 7, pp. 75490-75499, 2019, doi: 10.1109/ACCESS.2019.2922104.

[3] Akram Bayat, Marc Pomplun, Duc A. Tran, A Study on Human Activity Recognition Using Accelerometer Data from Smartphones, Procedia Computer Science, Volume 34, 2014, Pages 450-457, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2014.07.009.

[4] E. Bulbul, A. Cetin and I. A. Dogru, "Human Activity Recognition Using Smartphones," 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 2018, pp. 1-6, doi: 10.1109/ISMSIT.2018.8567275.

[5] Ferrari, A., Micucci, D., Mobilio, M. *et al.* Trends in human activity recognition using smartphones. *J Reliable Intell Environ* **7,** 189–213 (2021). https://doi.org/10.1007/s40860-021-00147-0.

[6] D. Micucci, M. Mobilio, and P. Napoletano, “UniMiB SHAR: A Dataset for Human Activity Recognition Using Acceleration Data from Smartphones,” *Applied Sciences*, vol. 7, no. 10, p. 1101, Oct. 2017, doi: 10.3390/app7101101. [Online]. Available: http://dx.doi.org/10.3390/app7101101.

[7] Heng, X., Wang, Z., and Wang, J. (2016) Human activity recognition based on transformed accelerometer data from a mobile phone. *Int. J. Commun. Syst.*, 29: 1981– 1991. doi: 10.1002/dac.2888.

[8] A. Jain and V. Kanhangad, "Human Activity Classification in Smartphones Using Accelerometer and Gyroscope Sensors," in IEEE Sensors Journal, vol. 18, no. 3, pp. 1169-1177, 1 Feb.1, 2018, doi: 10.1109/JSEN.2017.2782492.

[9] Y. Chen and C. Shen, "Performance Analysis of Smartphone-Sensor Behavior for Human Activity Recognition," in IEEE Access, vol. 5, pp. 3095-3110, 2017, doi: 10.1109/ACCESS.2017.2676168.

[10] Saha, J., Chowdhury, C., Ghosh, D. *et al.* A detailed human activity transition recognition framework for grossly labeled data from smartphone accelerometer. *Multimed Tools Appl* **80,** 9895–9916 (2021). https://doi.org/10.1007/s11042-020-10046-w

[11] Wan, N., & Lin, G. (2016). Classifying Human Activity Patterns from Smartphone Collected GPS data: a Fuzzy Classification and Aggregation Approach. *Transactions in GIS : TG*, *20*(6), 869–886. https://doi.org/10.1111/tgis.12181

[12] K. Montero Quispe, W. Sousa Lima, D. Macêdo Batista, and E. Souto, “MBOSS: A Symbolic Representation of Human Activity Recognition Using Mobile Sensors,” *Sensors*, vol. 18, no. 12, p. 4354, Dec. 2018, doi: 10.3390/s18124354. [Online]. Available: http://dx.doi.org/10.3390/s18124354

[13] A. Ferrari, D. Micucci, M. Mobilio and P. Napoletano, "On the Personalization of Classification Models for Human Activity Recognition," in IEEE Access, vol. 8, pp. 32066-32079, 2020, doi: 10.1109/ACCESS.2020.2973425.