MINOR PROJECT REPORT

Topic Name : Recognizing Human Activity using Smartphone Sensors

NAME	ENROLLMENT NO.	BATCH	Personal Email ID	College Email Id
Rahul Singhal	18103316	B10	rahulsinghal1904@gmail.com	18103316@mail.jiit.ac.in
Himanshu Joshi	18103132	B4	himanshujiit2018@gmail.com	18103132@mail.jiit.ac.in
Pranjal Chandel	18103139	B4	pchandel2000@gmail.com	18103139@mail.jiit.ac.in

Project Supervisor: Dr. Satish Chandra

ACKNOWLEDGEMENT

First of all we would like to express our gratitude towards Almighty to enable us to complete this report on “Recognizing Human Activity using Smartphone Sensors”.

Successful completion of any project requires help from a number of persons.So we would like to thank our parents and friends for being our constant support during the course of the project.

Also, most importantly, we would like to extend our sincere thanks towards our Project Supervisor Dr. Satish Chandra. Without his kind direction and proper guidance this study would not have been a success. In every phase of this project his supervision and guidance shaped this report to be completed successfully. Working on this project has helped us immensely in increasing our knowledge and skills.

Introduction

Smartphones are an inevitable part of our lives, as we all know that smartphones use many types of sensors to enhance the experience of their users. Two of these are accelerometers and gyrometer. Accelerometer measures acceleration while the gyroscope measures angular velocity. These kinds of technological innovations are much needed these days as they are being used in day to day purposes like fitness gadgets, fitness bands or any mobile fitness apps.
So we chose this trending and upsurging topic to be the subject of our project assessment so as to learn the real world applications machine learning. For this, people wore a smartphone on their waists while performing different activities to get the required readings. So by using the features built using readings provided by accelerometer and gyroscope, we will try to predict one of the six activities which human beings performed. During the course of the project we have tried to use different models and compare them to get the model with adequate accuracy according to the features and data available to us.

Objective

We intend to apply the best suited machine learning model which fits well in recognizing activities like standing,walking,walking upstairs,walking downstairs etc. These days fitness apps and bands have become very popular and this project tries to use the sensors in smartphones which use the same mechanism as used in these applications.

Understanding human activities comes handy in the health sector, especially in rehabilitation, elder care support services and cognitive impairment. Sensors will record and monitor the patient's activities and report automatically when any abnormality is detected and hence it saves a variety of resources. Also other applications include location indicator and human survey system which are extremely useful these days.

Data Source and its Storage

Accelerometer and Gyroscope readings which are used in this project are taken from 30 volunteers while performing the following 6 Activities.

These activities are listed as follows:

WALKING
WALKING_UPSTAIRS

WALKING_DOWNSTAIRS
SITTING
STANDING
LYING

Each person performed six activities wearing a smartphone 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 obtained dataset has been randomly partitioned into two sets, where 70% of the volunteers were selected for generating the training data and 30% the test data.

Features:

1. These sensor signals are first processed by applying noise filters and then sampled in fixed-width windows of 2.56 seconds having 50% overlap. i.e., each window has 128 readings.
2. Each window from signal processing has engineered a feature vector of size 564 by calculating different variables from the time and frequency domain.
3. 564 features are stored in the file "features.docx".
4. The acceleration signal was separated into Body and Gravity acceleration signals(tBodyAcc-XYZ and tGravityAcc-XYZ)
5. After that, the body linear acceleration and angular velocity were derived in time to obtain jerk signals (tBodyAccJerk-XYZ and tBodyGyroJerk-XYZ).
6. The magnitude of these 3-dimensional signals were calculated using the Euclidean norm. These magnitudes are represented as features with names like tBodyAccMag, tGravityAccMag, tBodyAccJerkMag, tBodyGyroMag and tBodyGyroJerkMag.
7. Finally, We've got frequency domain signals from some of the available signals by applying a FFT (Fast Fourier Transform). These signals are labelled as fBodyAcc-XYZ, fBodyGyroMag etc.

These are the signals that we got so far. We have also estimated the following properties on each and every signal that we recorded so far.

Sr. No.	***Signals***	***Statistical properties estimated on every signal***
1	tBodyAcc-XYZ	mean(): Mean value
2	tGravityAcc-XYZ	std(): Standard deviation
3	tBodyAccJerk-XYZ	mad(): Median absolute deviation
4	BodyGyro-XYZ	max(): Largest value in array
5	tBodyGyroJerk-XYZ	min(): Smallest value in array
6	tBodyAccMag	sma(): Signal magnitude area
7	tGravityAccMag	energy(): Energy measure. Sum of the squares divided by the number of values.
8	tBodyAccJerkMag	iqr(): Interquartile range
9	tBodyGyroMag	entropy(): Signal entropy
10	tBodyGyroJerkMag	arCoeff(): Auto-regression coefficients with Burg order equal to 4
11	fBodyAcc-XYZ	correlation(): correlation coefficient between two signals
12	fBodyAccJerk-XYZ	maxInds(): index of the frequency component with largest magnitude
13	fBodyGyro-XYZ	meanFreq(): Weighted average of the frequency components to obtain a mean frequency
14	fBodyAccMag	skewness(): skewness of the frequency domain signal
15	fBodyAccJerkMag	kurtosis(): kurtosis of the frequency domain signal
16	fBodyGyroMag	bandsEnergy(): Energy of a frequency interval within the 64 bins of the FFT of each window.
17	fBodyGyroJerkMag	angle(): Angle between two vectors.

There are 564 features made out of 128 readings of raw data from which All of the Accelerometer and Gyroscope are tri-axial, which means that they measure acceleration and angular-velocity respectively in the X-axis, Y-axis and Z-axis. So, we have in total six time-series data. We can visualize this problem as a multi class classification problem in terms of Machine Learning where the main goal is to predict which of the six activities is highly likely that the human is performing. For this we are using four of the well known ML models: Logistic Regression, Linear SVM, Decision Tree and LSTM Model and then comparing them to get the best model.

10. We can obtain some other vectors by taking the average of signals in a single window sample which are used on the angle() variable.
gravityMean
tBodyAccMean
tBodyAccJerkMean
tBodyGyroMean
tBodyGyroJerkMean

Problem Statement

To identify and classify different human activities (walking simple,upstairs,downstairs, sitting,lying) using different machine learning models and then finding the most suitable one on the base of accuracy and features.

About Android Smartphone Sensors

<u>Accelerator:</u>

An accelerometer is an electromechanical device used to measure acceleration forces. Such forces may be static, like the continuous force of gravity or, as is the case with many mobile devices, dynamic to sense movement or vibrations.Acceleration is the measurement of the change in velocity, or speed divided by time. For example, a car accelerating from a standstill to 60 mph in six seconds is determined to have an acceleration of 10 mph per second (60 divided by 6).

<u>Gyrometer:</u>

A gyroscope is a device used for measuring or maintaining orientation and angular velocity.It is a spinning wheel or disc in which the axis of rotation (spin axis) is free to assume any orientation by itself. When rotating, the orientation of this axis is unaffected by tilting or rotation of the mounting, according to the conservation of angular momentum.

Gyroscopes based on other operating principles also exist, such as the microchip-packaged MEMS gyroscopes found in electronic devices (sometimes called gyrometers), solid-state ring lasers, fibre optic gyroscopes, and the extremely sensitive quantum gyroscope.

Applications of gyroscopes include inertial navigation systems, such as in the Hubble Telescope, or inside the steel hull of a submerged submarine. Gyroscopes can be used to construct gyrocompasses, which complement or replace magnetic compasses (in ships, aircraft and spacecraft, vehicles in general), to assist in stability (bicycles, motorcycles, and ships) or be used as part of an inertial guidance system.

Information about Different Models Used

Logistic Regression:

Logistic regression is one of the most fundamental and widely used Machine Learning Algorithms. Logistic regression is usually among the first few topics which people pick while learning predictive modeling. Logistic regression is not a regression algorithm but a probabilistic classification model.Classification in Machine Learning is a technique of learning, where an instance is mapped to one of many labels. The machine learns patterns from data in such a way that the learned representation successfully maps the original dimension to the suggested label/class without any intervention from a human expert.

Logistic regression has a sigmoidal curve. Multiclass classification with logistic regression can be done either through the one-vs-rest scheme in which for each class a binary classification problem of data belonging or not to that class is done, or changing the loss function to cross- entropy loss.In the multi class logistic regression python Logistic Regression class, multi-class classification can be enabled/disabled by passing values to the argument called ‘‘multi_class’ in the constructor of the algorithm. By default, multi_class is set to ’ovr’.

Linear SVM:

The objective of Linear SVM is to find a hyperplane in an n-dimensional space that separates the data points to their potential classes. The hyperplane should be positioned with the maximum distance to the data points. The data points with the minimum distance to the hyperplane are called Support Vectors. Due to their close position, their influence on the exact position of the hyperplane is bigger than other data points. It is a supervised machine learning algorithm that helps in classification or regression problems. It aims to find an optimal boundary between the possible outputs.

Simply put, SVM does complex data transformations depending on the selected kernel function and based on that transformations, it tries to maximize the separation boundaries between your data points depending on the labels or classes you've defined.In its most simple type, SVM doesn't support multiclass classification natively.For multiclass classification, the same principle is utilized after breaking down the multi classification problem into multiple binary classification problems.
The idea is to map data points to high dimensional space to gain mutual linear separation between every two classes. This is called a One-to-One approach, which breaks down the multiclass problem into multiple binary classification problems. A binary classifier per each pair of classes.Another approach one can use is One-to-Rest. In that approach, the breakdown is set to a binary classifier per each class.

Decision Trees:

A Decision Tree is a predictive model expressed as a recursive partition of the feature space to subspaces that constitute a basis for prediction. A Decision Tree is a rooted directed tree. In DTs, nodes with outgoing edges are the internal nodes. All other nodes are terminal nodes or leaves of the DT. DTs classify using a set of hierarchical decisions on the features. The decisions made at internal nodes are the split criterion. In DTs, each leaf is assigned to one class or its probability. Small variations in the training set results in different splits leading to a different DT. Thus, the error contribution due to variance is large for DTs. Ensemble learning, discussed in the next section, can help palliate the error due to variance.Decision trees are one of the most commonly used predictive modeling algorithms in practice. The reasons for this are many. Some of the distinct advantages of using decision trees in many classification and prediction applications are explained below along with some common pitfalls.

1. Easy to interpret and explain to nontechnical users
2. As we have seen in the few examples discussed so far, decision trees are very intuitive and easy to explain to nontechnical people, who are typically the consumers of analytics.
3. Decision trees require relatively little effort from users for data preparation

4. If we have a data set consisting of widely ranging attributes, for example, revenues recorded in millions and loan age recorded in years, many algorithms require scale normalization before model building and application. Such variable transformations are not required with decision trees because the tree structure will remain the same with or without the transformation.
5. When we fit a decision tree to a training data, the top few nodes on which the tree is divided are the most essential variables within the data set and feature selection is completed automatically.

LSTM Model:

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems.
This is a behavior required in complex problem domains like machine translation, speech recognition, and more.LSTMs are a complex area of deep learning. It can be hard to get your hands around what LSTMs are, and how terms like bidirectional and sequence-to-sequence relate to the field.Recurrent neural networks are different from traditional feed-forward neural networks.This difference in the addition of complexity comes with the promise of new behaviors that the traditional methods cannot achieve.Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition, speech recognition and anomaly detection in network traffic or IDSs (intrusion detection systems).

A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs.

Steps involved during the project

To achieve the desired outcome of this project,we first brainstormed and chalked out a complete plan on how to use relevant libraries and models. After researching,we laid the following roadmap:->

- So first we started with importing all the important libraries required for the project. These libraries were namely:-
  numpy,pandas,seaborn,matplotlib.pyplot,sklearn.manifold,warnings,datetime,GridSearchCV,confusion_matrix,accuracy_score,Dropout,Trials,STATUS_OK,tpe,optim,choice,uniform.
- Then we extracted features from **features.txt** file.
- Then we read the training and test data as training data is used to train the model to get accurate predictions and test data is used to test the model.
- Then we looked for removing the duplicate and null values in data(if present).Also we checked if there is any imbalance in data.
- We visualized the balance in data by plotting the graphs like **Count v/s Subject ID** and **Count v/s Different Human Activities.**
- Then we explored feature information from our previous knowledge. We categorised the motion into two categories - **Static and Dynamic**.
  Static includes Sitting,Standing,Lying and Dynamic includes Walking Upstairs,Walking Downstairs and normal walking.
- Then we plotted two graphs which observed that the magnitude of the mean of the body's acceleration in the time domain measured by the accelerometer is able to separate static activity from dynamic activity.
- So we applied the T-distributed stochastic neighbor embedding (TSNE) plot to data for better visualization and to get a better perplexity. So using this, we obtained different sets of data points corresponding to different human activities. From T-SNE plots, we can observe that except Standing and Sitting all other activities are separated fairly well.
- For better understanding, we applied following models and compared their accuracies accordingly:-
  - We first applied **Logistic Regression** and made three matrices namely Confusion Matrix, Precision matrix and Recall Matrix.
  - Then we applied **Linear SVM** and also made three matrices for this model also: namely Confusion Matrix, Precision matrix and Recall Matrix.
  - Also we applied **Decision Tree** and made three matrices Confusion,Precision and Recall Matrix.
  - Then we applied the deep learning **LSTM Model** where we used raw readings obtained from accelerometer and gyroscope signals and tuned Hyper-parameters using Hyperas and used the best Hyper Parameters.
- Then we compared the accuracies of all the models used and found that the LSTM Model gave the good predictions even when we didn't have domain expert engineered features.
  Also we noticed that when we used the best Hyper parameters, the accuracy of the model was improved.

Performance Metric Used

We use accuracy as one of the metrics. We use confusion-matrix to check that in which two activities our model is confused and predicting incorrect activity, for example, between standing up and standing down and between walking upstairs and walking downstairs.

Precision

Out of all the positive predicted, what percentage is truly positive.The precision value lies between 0 and 1.

$$Precision = \frac{TP}{TP + FP}$$

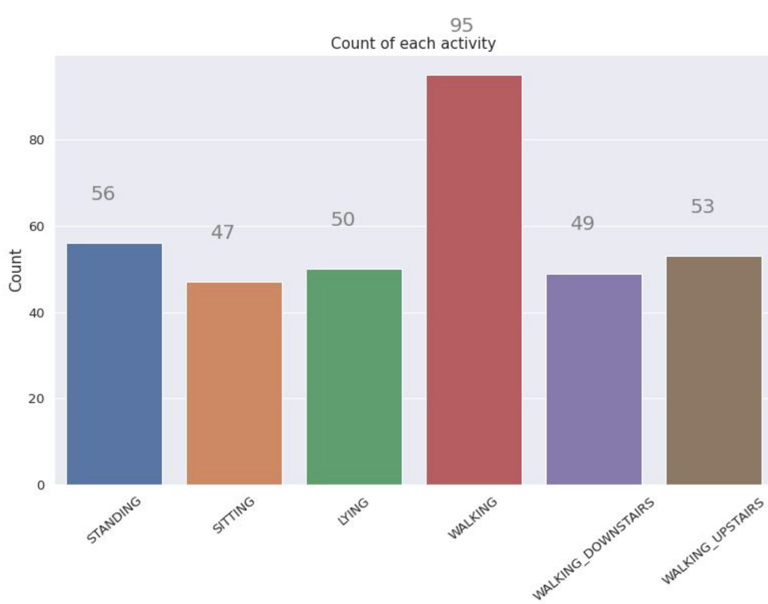
Recall

Out of the total positive, what percentage are predicted positive. It is the same as TPR (true positive rate).

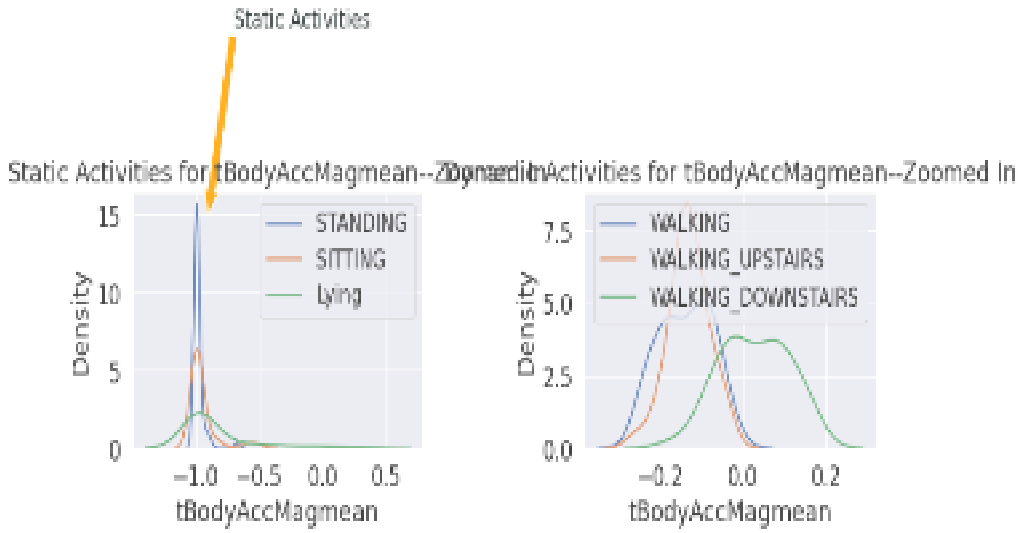
$$Recall = \frac{TP}{TP + FN}$$

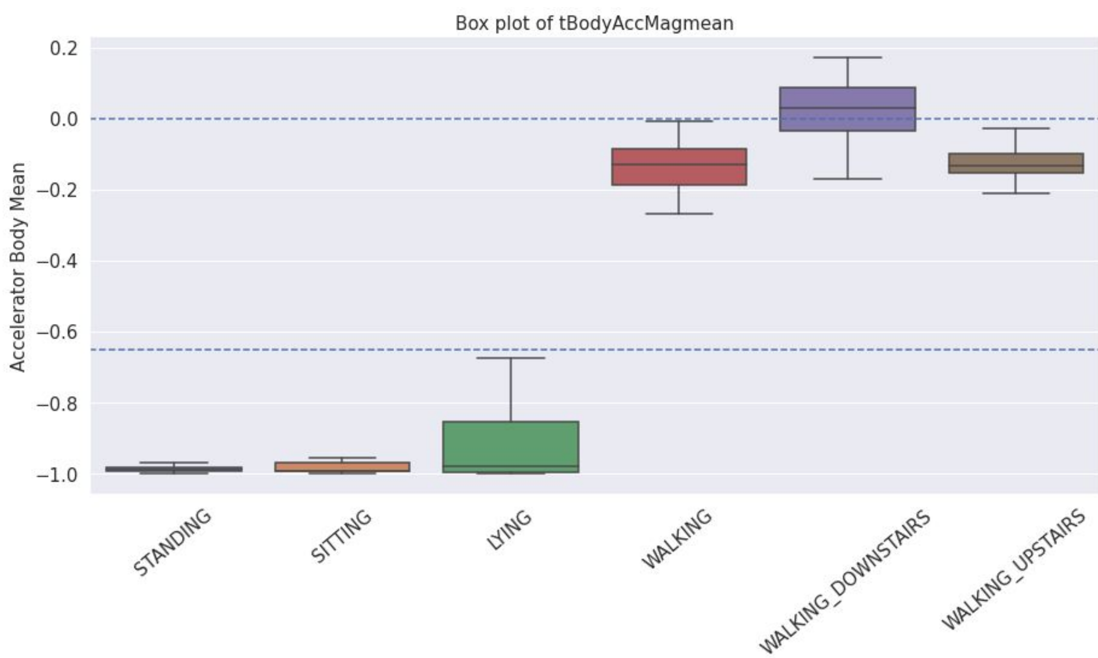
Observations derived during the course of the project

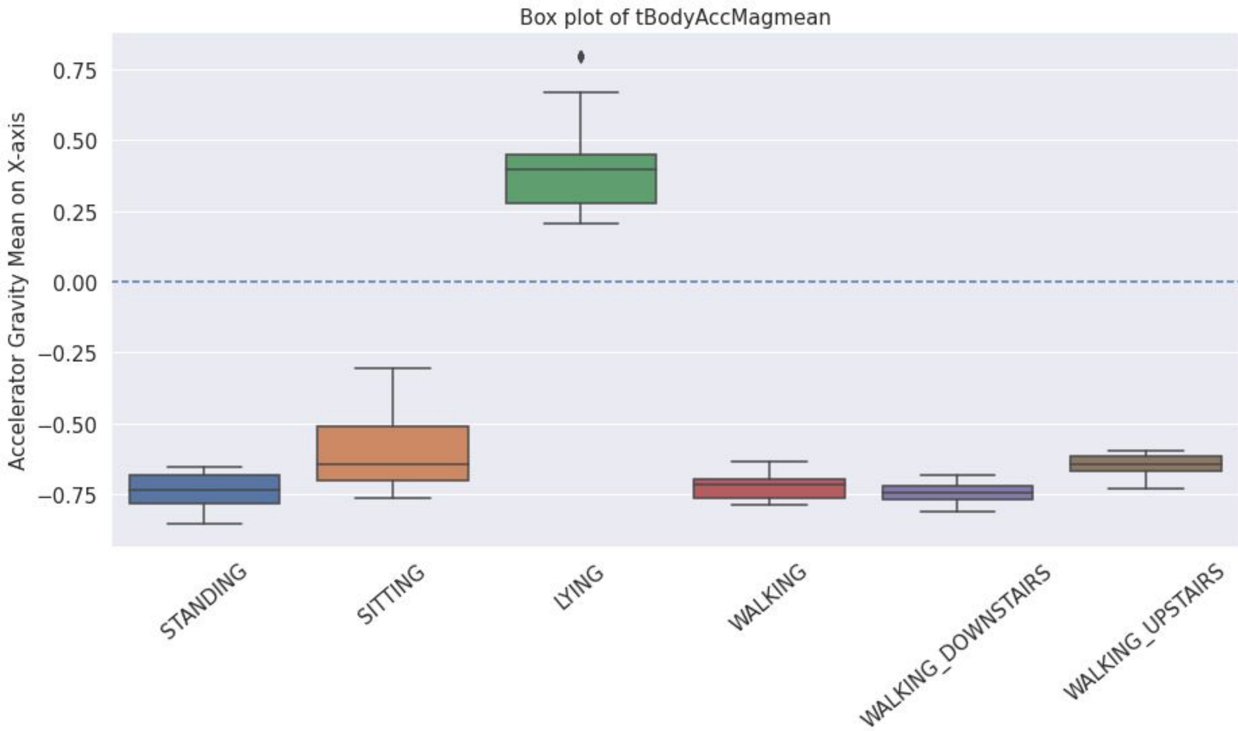
- Count v/s Count of each activity graph

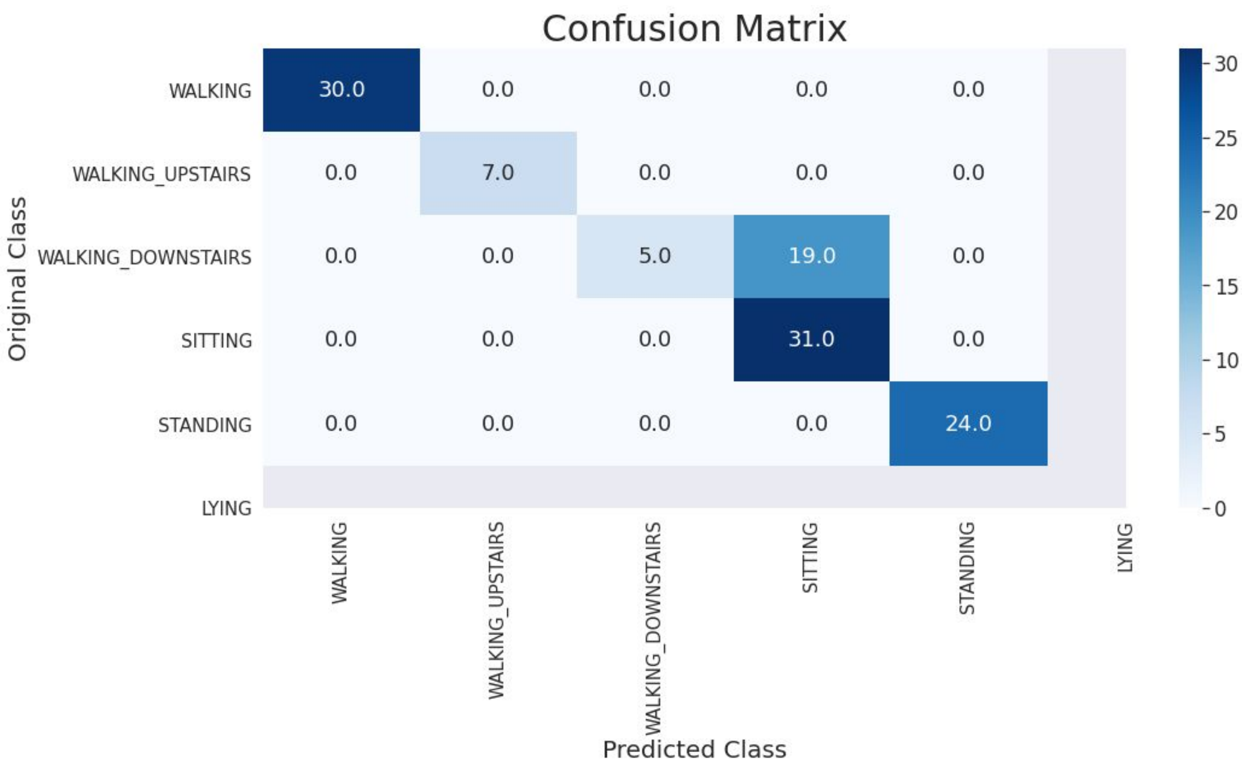


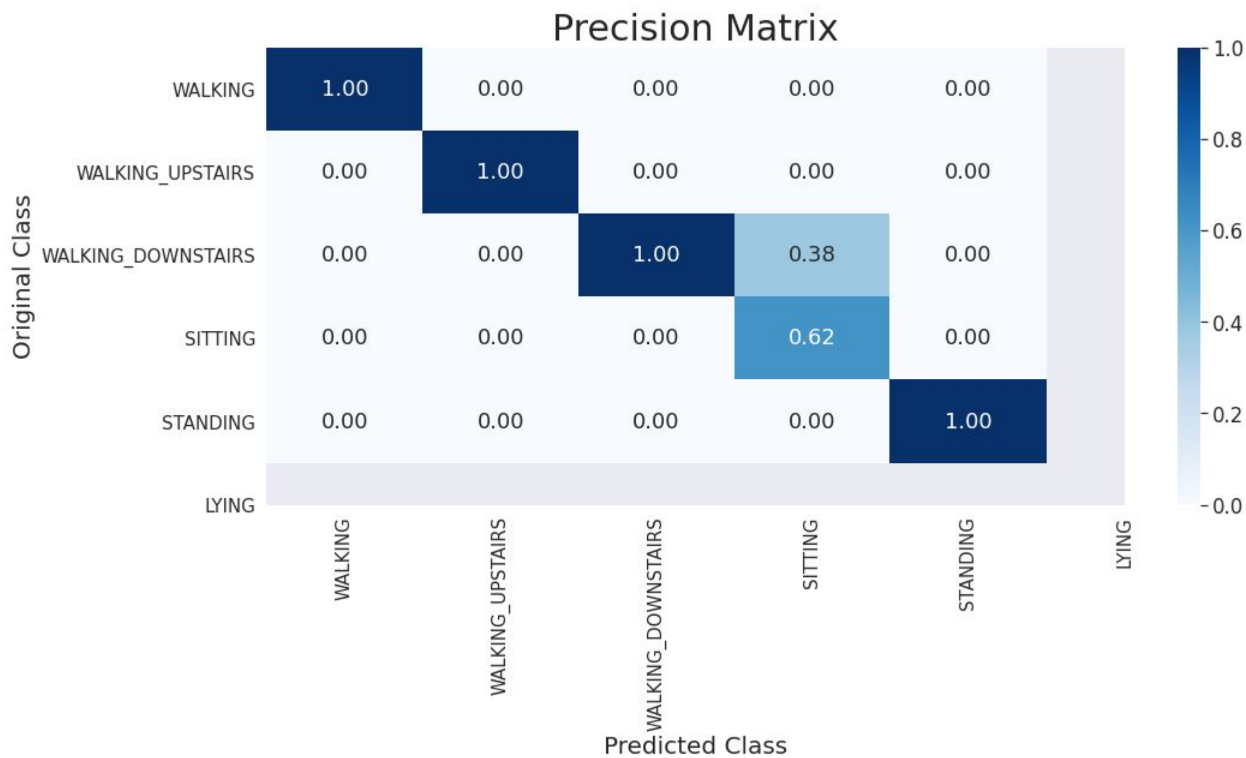
- Density v/s tBodyAccMagmean graphs



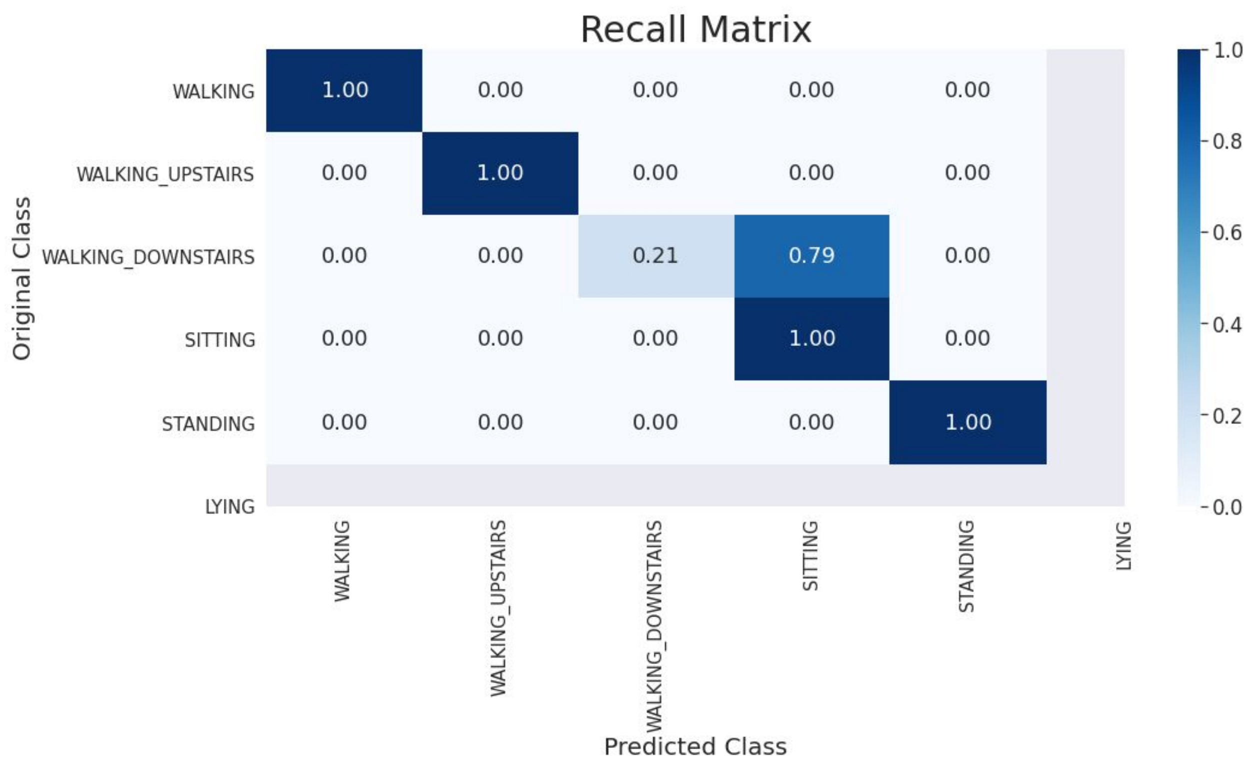



Box plot of tBodyAccMagmean
Accelerator Gravity Mean on X-axis
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
STANDING
SITTING
LYING
WALKING
WALKING_DOWNSTAIRS
WALKING_UPSTAIRS

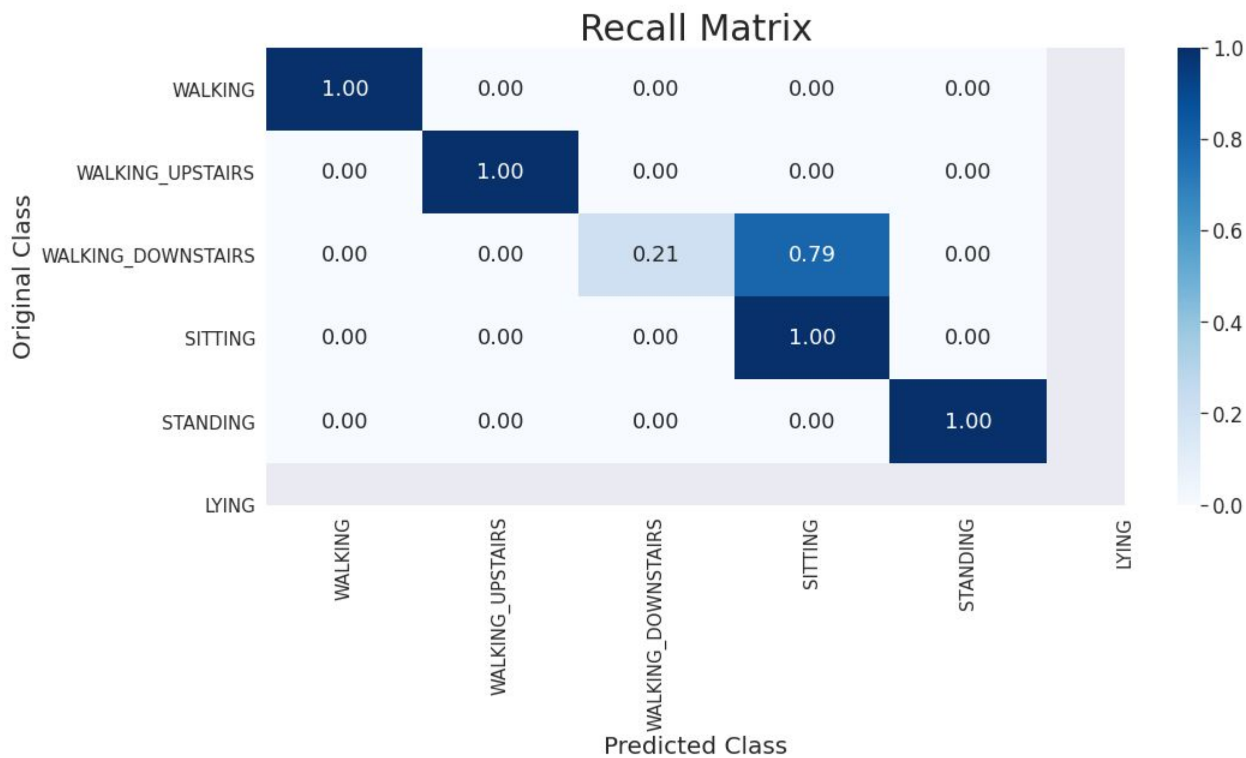

Confusion Matrix
Original Class
Predicted Class
WALKING
WALKING_UPSTAIRS
WALKING_DOWNSTAIRS
SITTING
STANDING
LYING
30.0
7.0
5.0
19.0
31.0
24.0
0.0

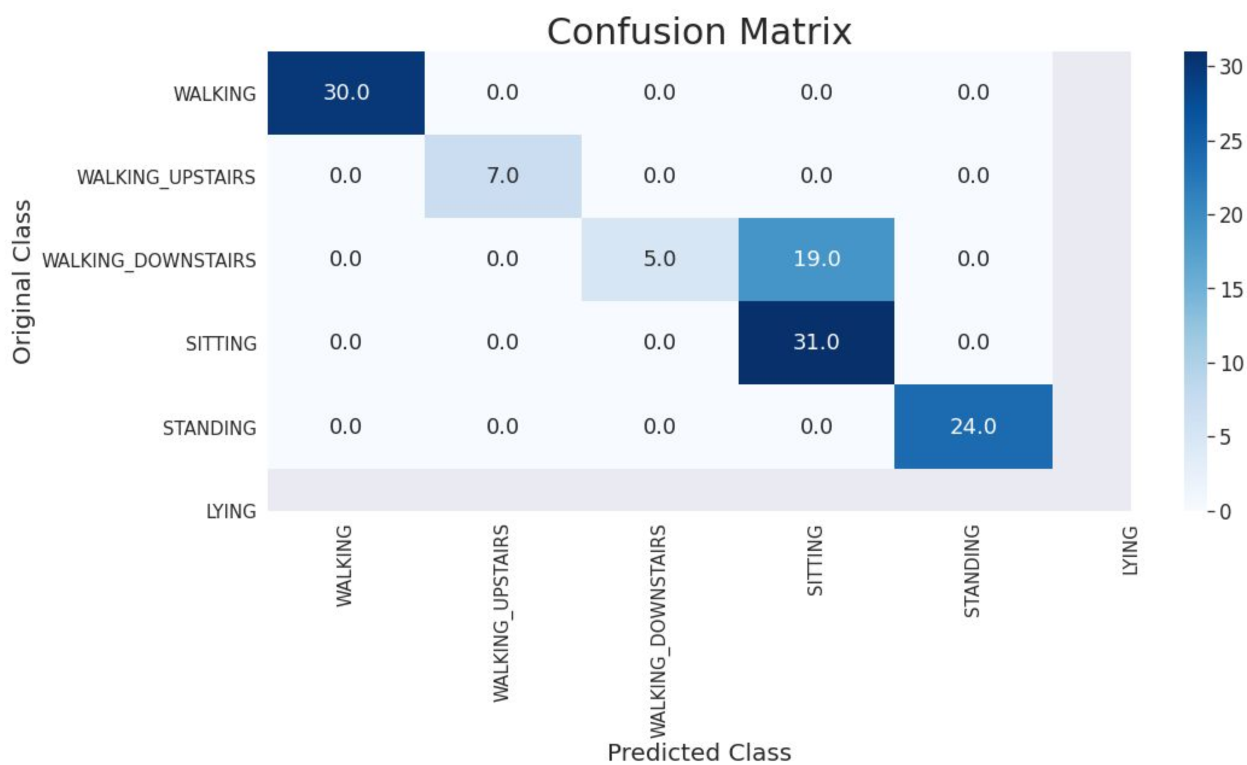


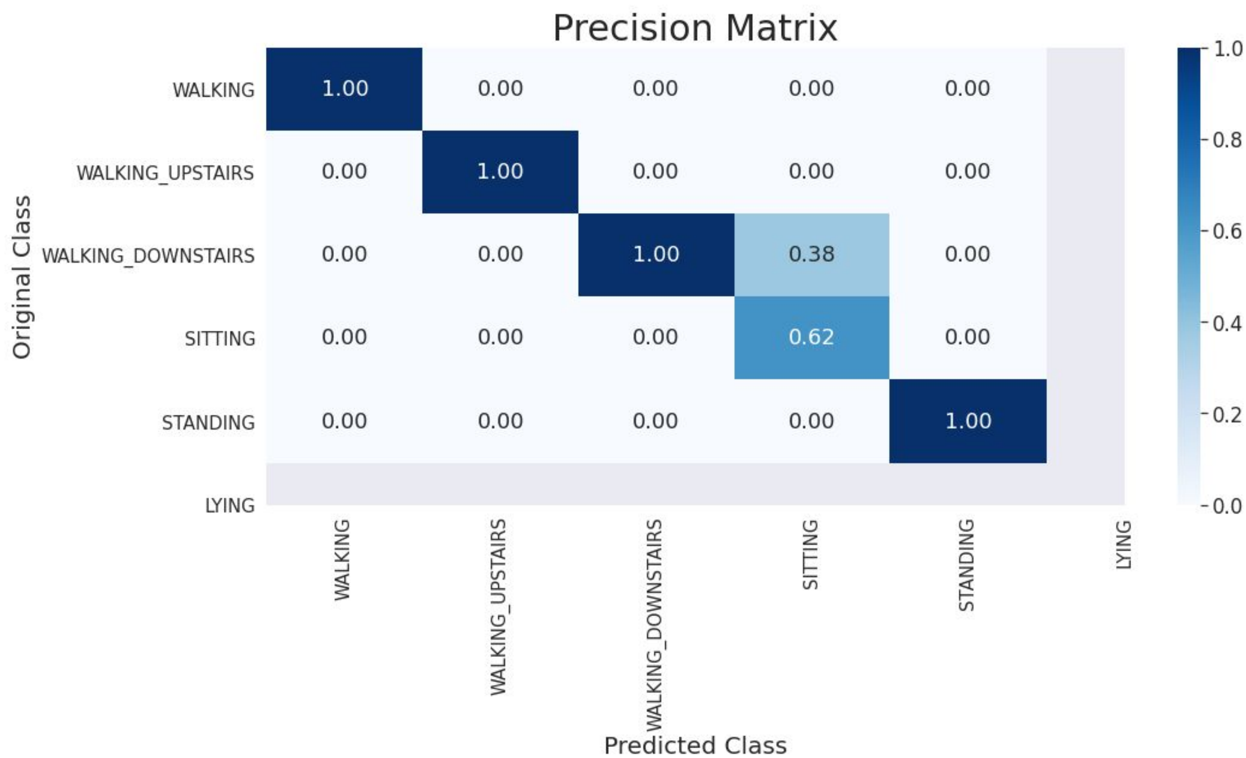
Precision Matrix
WALKING
WALKING_UPSTAIRS
WALKING_DOWNSTAIRS
SITTING
STANDING
LYING
1.00
0.00
0.38
0.62
1.0
0.8
0.6
0.4
0.2
0.0
Original Class
Predicted Class



Recall Matrix
WALKING
WALKING_UPSTAIRS
WALKING_DOWNSTAIRS
SITTING
STANDING
LYING
1.00
0.00
0.21
0.79
1.0
0.8
0.6
0.4
0.2
0.0
Original Class
Predicted Class

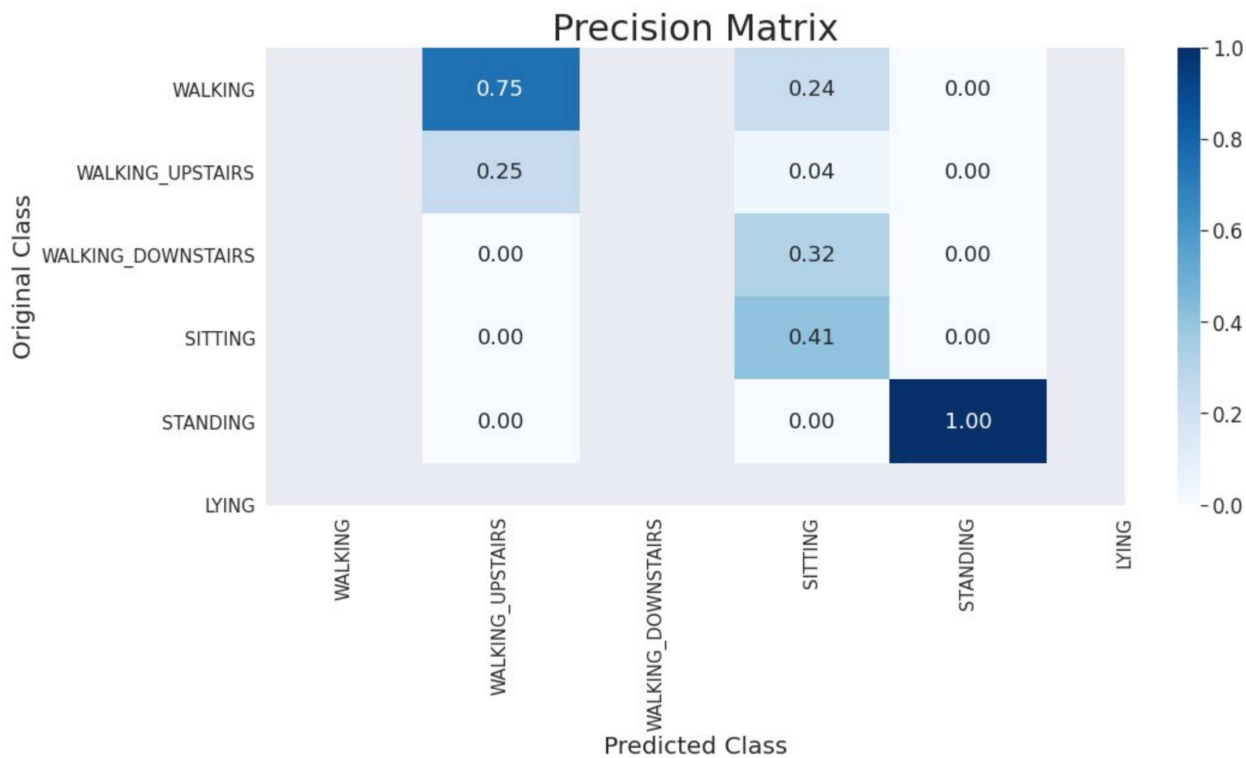

Recall Matrix
WALKING
WALKING_UPSTAIRS
WALKING_DOWNSTAIRS
SITTING
STANDING
LYING
Original Class
Predicted Class
1.00 0.00 0.00 0.00 0.00
0.00 1.00 0.00 0.00 0.00
0.00 0.00 0.21 0.79 0.00
0.00 0.00 0.00 1.00 0.00
0.00 0.00 0.00 0.00 1.00
1.0
0.8
0.6
0.4
0.2
0.0


Confusion Matrix
WALKING
WALKING_UPSTAIRS
WALKING_DOWNSTAIRS
SITTING
STANDING
LYING
Original Class
Predicted Class
30.0 0.0 0.0 0.0 0.0
0.0 7.0 0.0 0.0 0.0
0.0 0.0 5.0 19.0 0.0
0.0 0.0 0.0 31.0 0.0
0.0 0.0 0.0 0.0 24.0
30
25
20
15
10
5
0



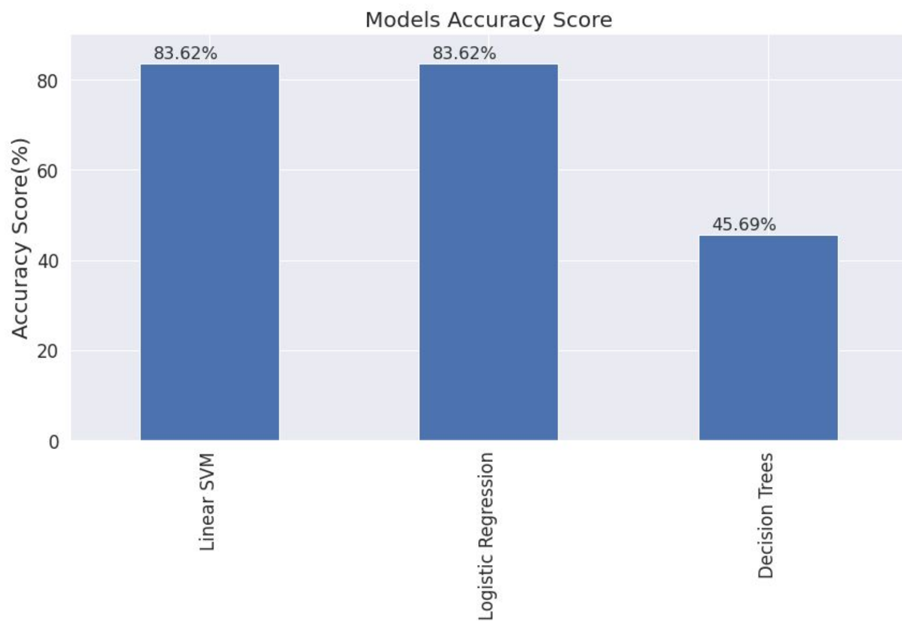
Precision Matrix
Original Class
Predicted Class
WALKING
WALKING_UPSTAIRS
WALKING_DOWNSTAIRS
SITTING
STANDING
LYING
1.00 0.00 0.00 0.00 0.00
0.00 1.00 0.00 0.00 0.00
0.00 0.00 1.00 0.38 0.00
0.00 0.00 0.00 0.62 0.00
0.00 0.00 0.00 0.00 1.00
0.0 0.2 0.4 0.6 0.8 1.0

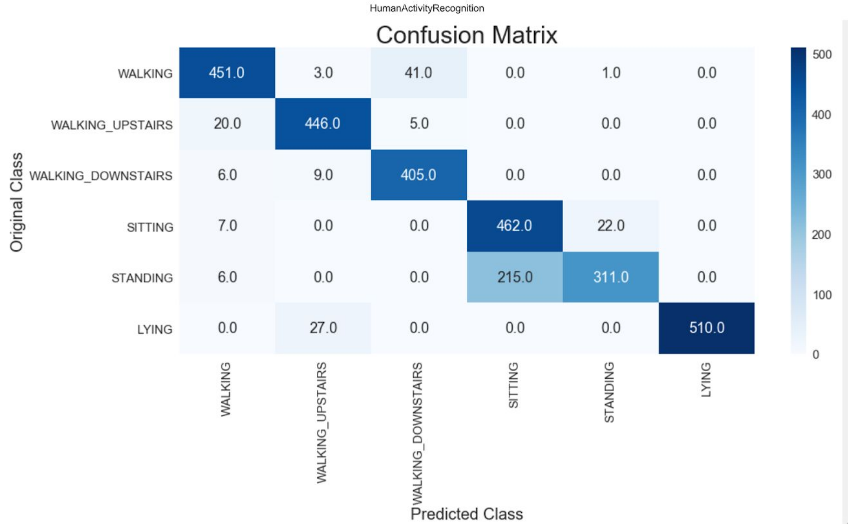
Confusion Matrix
Original Class
Predicted Class
WALKING
WALKING_UPSTAIRS
WALKING_DOWNSTAIRS
SITTING
STANDING
LYING
0.0 12.0 0.0 18.0 0.0
0.0 4.0 0.0 3.0 0.0
0.0 0.0 0.0 24.0 0.0
0.0 0.0 0.0 31.0 0.0
0.0 0.0 0.0 0.0 24.0
0 5 10 15 20 25 30

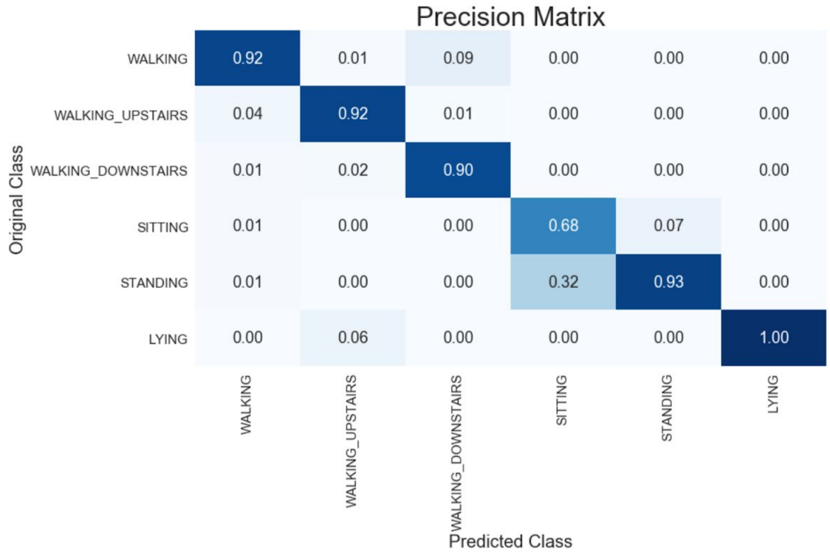

Precision Matrix
Original Class
Predicted Class
WALKING
WALKING_UPSTAIRS
WALKING_DOWNSTAIRS
SITTING
STANDING
LYING
0.75
0.24
0.00
0.25
0.04
0.00
0.00
0.32
0.00
0.00
0.41
0.00
0.00
0.00
1.00
1.0
0.8
0.6
0.4
0.2
0.0


Recall Matrix
Original Class
Predicted Class
WALKING
WALKING_UPSTAIRS
WALKING_DOWNSTAIRS
SITTING
STANDING
LYING
0.00 0.40 0.00 0.60 0.00
0.00 0.57 0.00 0.43 0.00
0.00 0.00 0.00 1.00 0.00
0.00 0.00 0.00 1.00 0.00
0.00 0.00 0.00 0.00 1.00
1.0
0.8
0.6
0.4
0.2
0.0

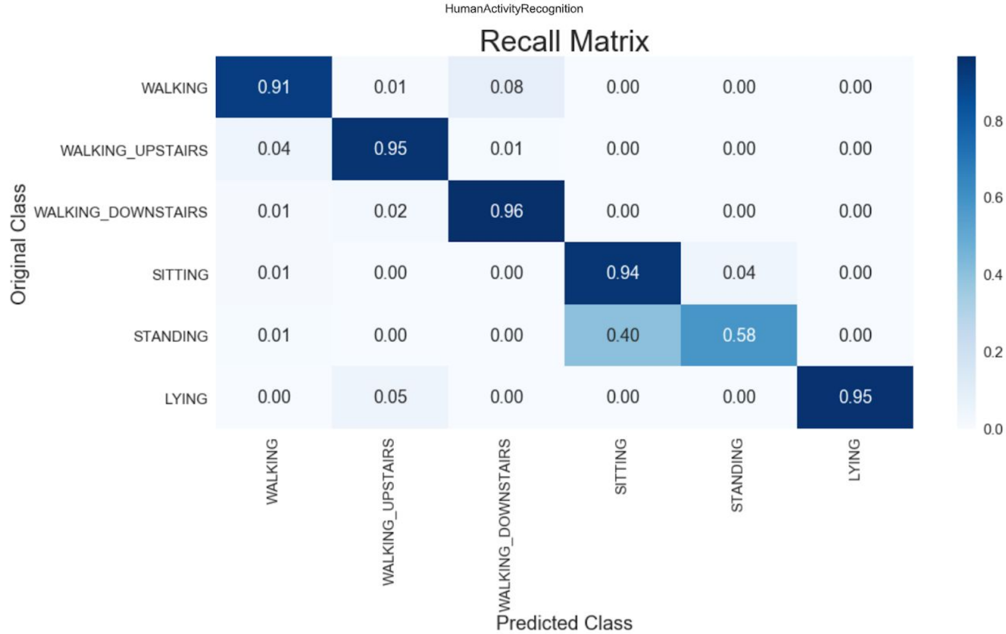
- Model Accuracy Score for Logistic Regression, Linear SVM and Decision Trees








```
------------------------------------------------------------------------------------------------------------------------
------
```

Final Comments

- By Simple two layered LSTM, we got a good accuracy of 87.72%. In short, DeeP Learning help us to built models even when we don't have domain expert engineered features.
- LSTM model can be further improved by running it for more epochs and more evaluations while tuning hyper-parameter.

Conclusion based on observations

From the above T-SNE plots, we can observe that except STANDING and SITTING, all other activities are separated fairly well.

- If Acc Gravity Mean > 0, we can infer that the activity will most likely be Lying.
- If Acc Gravity Mean < 0, we can infer that the activity can be anything but Lying.
- If tAccMean is < -0.8 then the Activities are either Standing or Sitting or Laying.
- If tAccMean is > -0.6 then the Activities are either Walking or Walking Downstairs or Walking Upstairs.
- If tAccMean > 0.0 then the Activity is Walking Downstairs.
- We can classify 75% of the Activity labels with some errors.

From the tBodyAccMagmean v/s Accelerator Gravity Mean plots we can clearly observe that how well "tBodyAccMagmean "--which is the magnitude of the mean of body acceleration in time-domain measured by accelerometer--is able to separate static activity from dynamic activity.

From the Count v/s Count of each activity plot, we can infer that our classes are almost balanced.

Also we saw that:

- Models: Logistic Regression, Linear SVM give accuracy 83.62%,decision tree is showing an accuracy of 45.69 % and LSTM is showing an accuracy of 87.72%. In short, deep learning helps us to build models even when we don't have domain expert engineered features.
- In the real world, having domain knowledge is one of the most important aspects of machine learning Modelling. Here, we got pretty good accuracy of 83.62%. This is very much due to the fact that features are very well engineered by domain experts in signal processing.
- In a nutshell, feature engineering is one of the most important aspects of machine learning.