Human Activity Recognition Using Smartphones

Introduction :

The human ability to recognize another person's activities is one of the main subjects of study of the scientific areas of computer vision and machine learning. Because of this research, many applications, including video surveillance systems, human-computer interaction, and robotics for human behavior characterization, require a multiple activity recognition system.

Problem Statement :

The problem in question is a multi-class Classification problem. The input data consists 3-axial linear acceleration and 3-axial angular velocity readings from the gyroscope of a Samsung Galaxy S II at a constant rate of 50Hz accelerometer. The input data consists of 561 which we need to analyze and our goal is to predict if a user is doing any of the following activities: WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING, LAYING from their smartphone activity.

I will be approaching this as any other multi-class classification problem and after analyzing the dataset and doing all the needful pre-processing on the dataset. I will apply Naive Bayes Algorithm to do an initial classification to check how the algorithm works on the dataset. After that I plan to use a Logistic Regression Classifier, a Support Vector Classifier followed by an ensemble using LightGBM and finally a Keras Neural Network to check which model performs best for the problem at hand. The solution will be classifying a user's activity to any of the following six activities: (WALKING, WALKING_UPSTAIRS, WALKING_DOWNSTAIRS, SITTING, STANDING or LAYING based on the input feature vector from his smartphone's sensor readings.

My approach is to find an optimal model which can solve the task in hand accurately and quickly. So, I will be using few classification algorithms on the dataset to see which one performs better and then use the one that provides the best output and work on that model further to fine tune its parameters to reach an optimum model.

DataSet :

The dataset I am going to use is an open source dataset which can be found in the UCI Machine Learning repository .(http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones)
.

How data was recorded:

The data set was collected from an experiment which was carried out with a group of 30 volunteers within an age bracket of 19-48 years. They performed a series of activities standing, sitting, lying, walking, walking downstairs and walking upstairs. The experiment also included postural transitions that occurred between the static postures which are: stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-tolie, and lie-to-stand. All the participants were wearing a smartphone (Samsung Galaxy S II) on the waist during the experiment execution. Data was captured with 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50Hz using the embedded accelerometer and gyroscope of the device. The experiments were video-recorded to label the data manually. The obtained dataset was randomly partitioned into two sets, where 70% of the volunteers was selected for generating the training data and 30% the test data.

Feature names :

These sensor signals are preprocessed by applying noise filters and then sampled in fixed-width windows(sliding windows) of 2.56 seconds each with a 50% overlap. ie., each window has 128 readings.

1. From Each window, a feature vector was obtained by calculating variables from the time and frequency domain.
In our dataset, each datapoint represents a window with different readings
2. The acceleration signal was separated into Body and Gravity acceleration signals(**tBodyAcc-XYZ** and **tGravityAcc-XYZ**) using some low pass filter with a corner frequency of 0.3Hz.
3. After that, the body linear acceleration and angular velocity were derived in time to obtain jerk signals (**tBodyAccJerk-XYZ**and **tBodyGyroJerk-XYZ**).
4. The magnitude of these 3-dimensional signals were calculated using the Euclidean norm. This magnitudes are represented as features with names like tBodyAccMag, tGravityAccMag, tBodyAccJerkMag, tBodyGyroMag and tBodyGyroJerkMag.
5. Finally, We've got frequency domain signals from some of the available signals by applying a FFT (Fast Fourier Transform). These signals obtained were labeled with **prefix 'f'** just like original signals with **prefix 't'**. These signals are labeled as **fBodyAcc-XYZ**, **fBodyGyroMag** etc.,.
7. These are the signals that we got so far.

- tBodyAcc-XYZ
- tGravityAcc-XYZ
- tBodyAccJerk-XYZ

- tBodyGyro-XYZ
- tBodyGyroJerk-XYZ
- tBodyAccMag
- tGravityAccMag
- tBodyAccJerkMag
- tBodyGyroMag
- tBodyGyroJerkMag
- fBodyAcc-XYZ
- fBodyAccJerk-XYZ
- fBodyGyro-XYZ
- fBodyAccMag
- fBodyAccJerkMag
- fBodyGyroMag
- fBodyGyroJerkMag

8. We can estimate some set of variables from the above signals. ie., We will estimate the following properties on each and every signal that we recorded so far.

- **mean()**: Mean value
- **std()**: Standard deviation
- **mad()**: Median absolute deviation
- **max()**: Largest value in array
- **min()**: Smallest value in array
- **sma()**: Signal magnitude area
- **energy()**: Energy measure. Sum of squares divided by the number of values.
- **iqr()**: Interquartile range
- **entropy()**: Signal entropy
- **arCoeff()**: Autoregression coefficients with Burg order equal to 4
- **correlation()**: correlation coefficient between two signals
- **maxInds()**: index of the frequency component with largest magnitude
- **meanFreq()**: Weighted average of the frequency components to obtain a mean frequency
- **skewness()**: skewness of the frequency domain signal

9. We can obtain some other vectors by taking the average of signals in a single window sample. These are used on the angle() variable'

- **gravityMean**
- **tBodyAccMean**

- **tBodyAccJerkMean**
- **tBodyGyroMean**
- **tBodyGyroJerkMean**
- **kurtosis()**: kurtosis of the frequency domain signal
- **bandsEnergy()**: Energy of a frequency interval within the 64 bins of FFT of each window.
- **angle()**: Angle between two vectors.

Data Exploration :

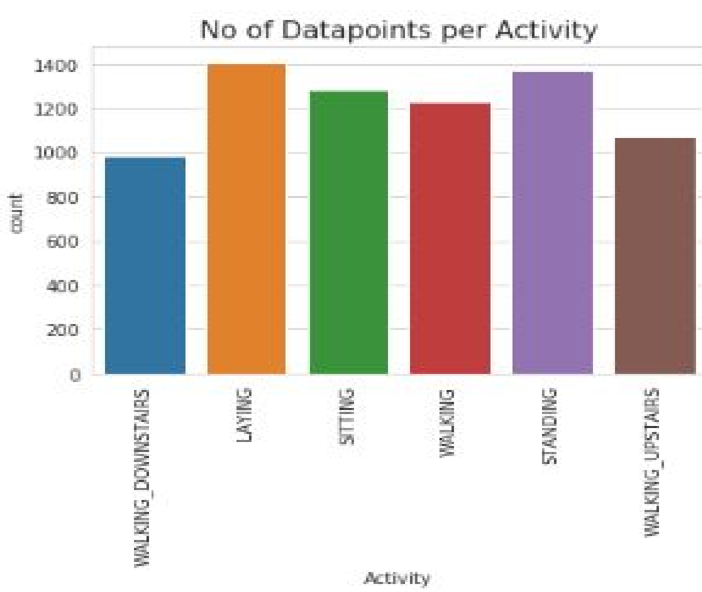
The data-set consists of the following features:

• A 561-feature vector with time and frequency domain variables.
• Its associated activity labels (WALKING, WALKING_UPSTAIRS, WALKING_ DOWNSTAIRS, SITTING, STANDING or LAYING).
• An identifier of the subject who carried out the experiment.

Data Cleaning:

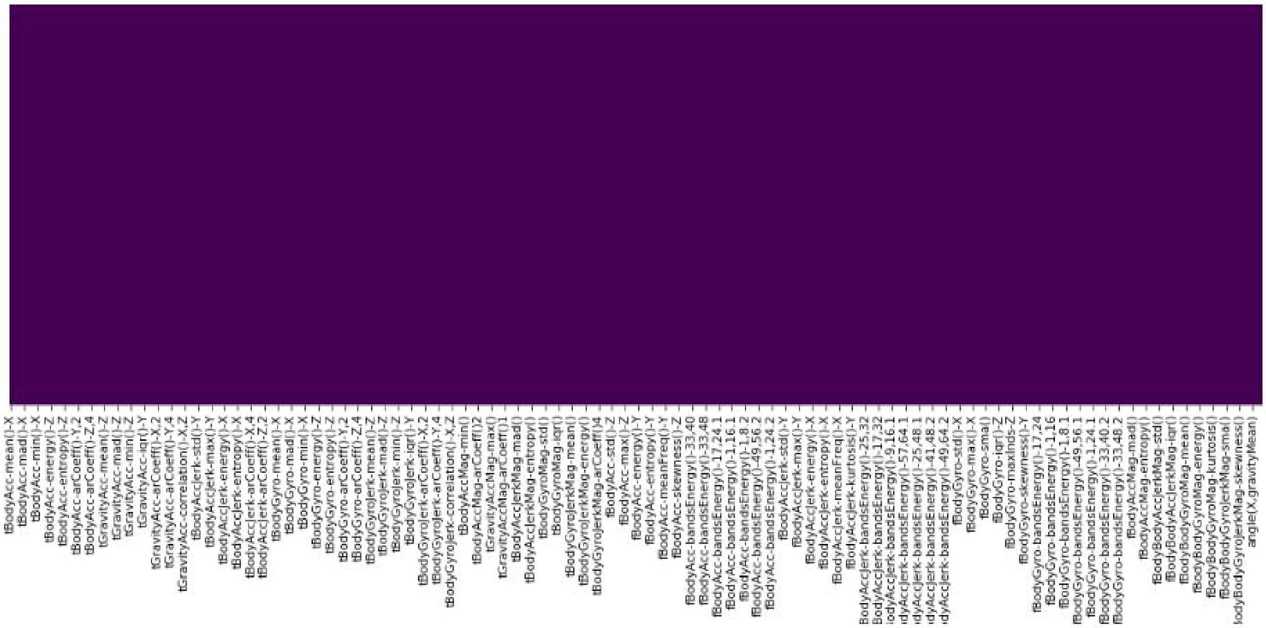
INITIAL EXPLORATION AND VISUALIZATION: In-order to make sure the dataset under consideration doesn't have any major data quality issues like class imbalance, missing data etc. that will hinder the performance of classification models to a huge extent we start with the basic analysis of our dataset using visualizations.

After importing the necessary libraries like NumPy, pandas, matplotlib and seaborn we start our analysis of data by checking if the data has class imbalance issue. The plot of the target variable 'Activity'. We observed that there isn't much variation in the class label counts.



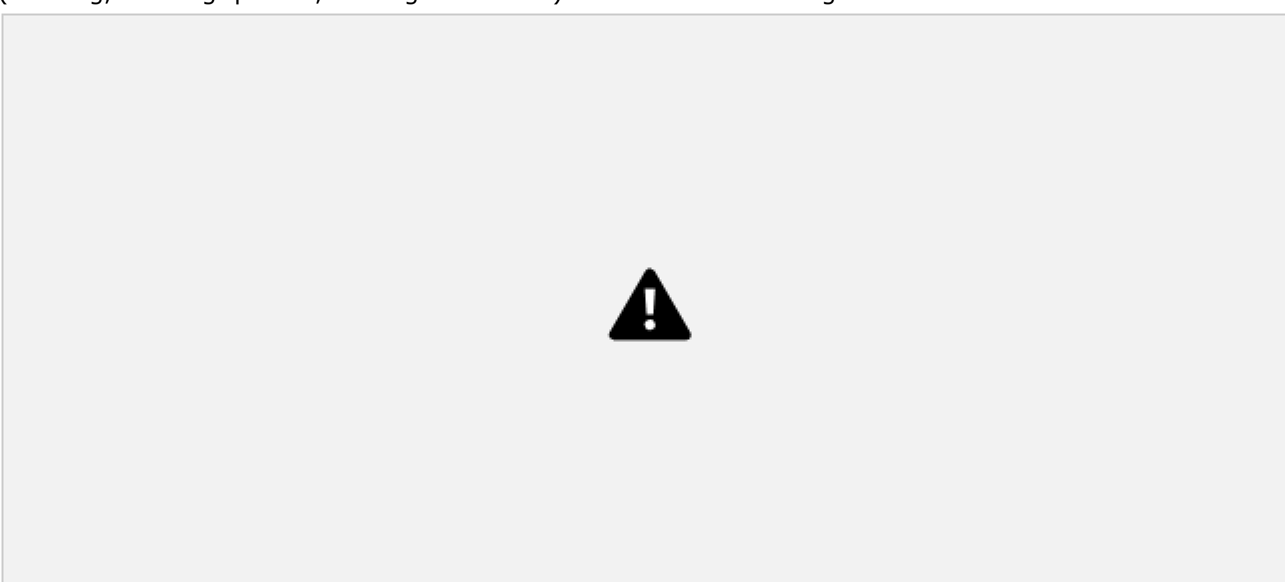
Missing Data:

To check for missing data points, we have generated a heatmap which would show us missing instances as colored dash lines in each feature. We observed that there are no missing

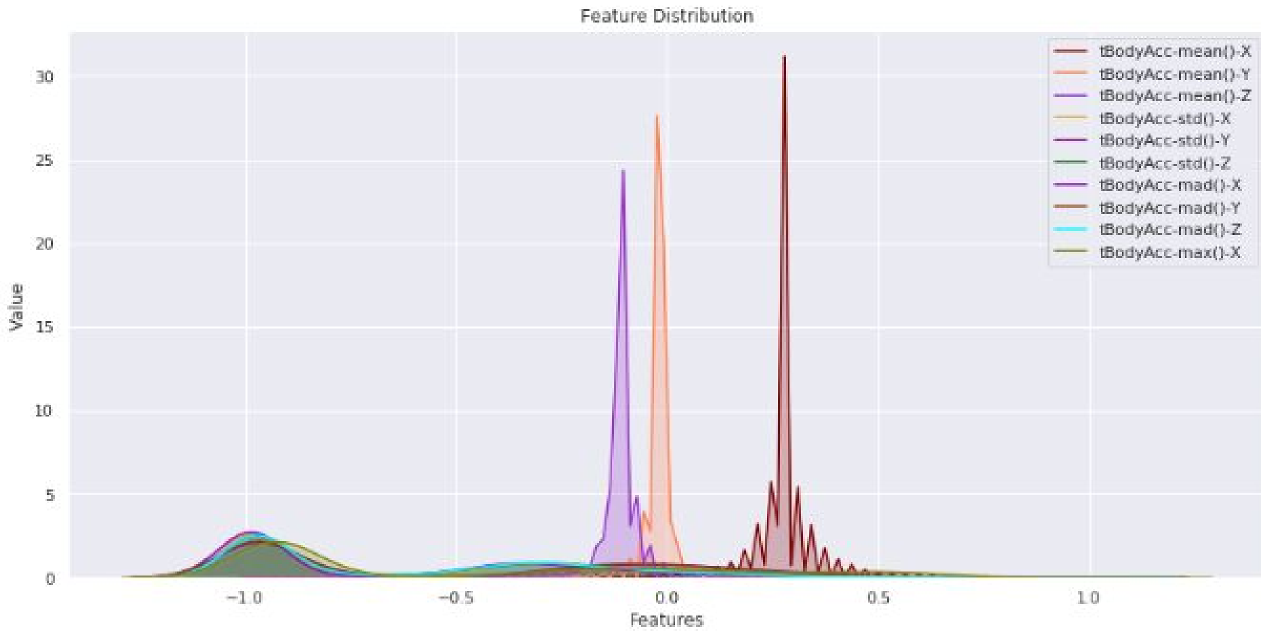


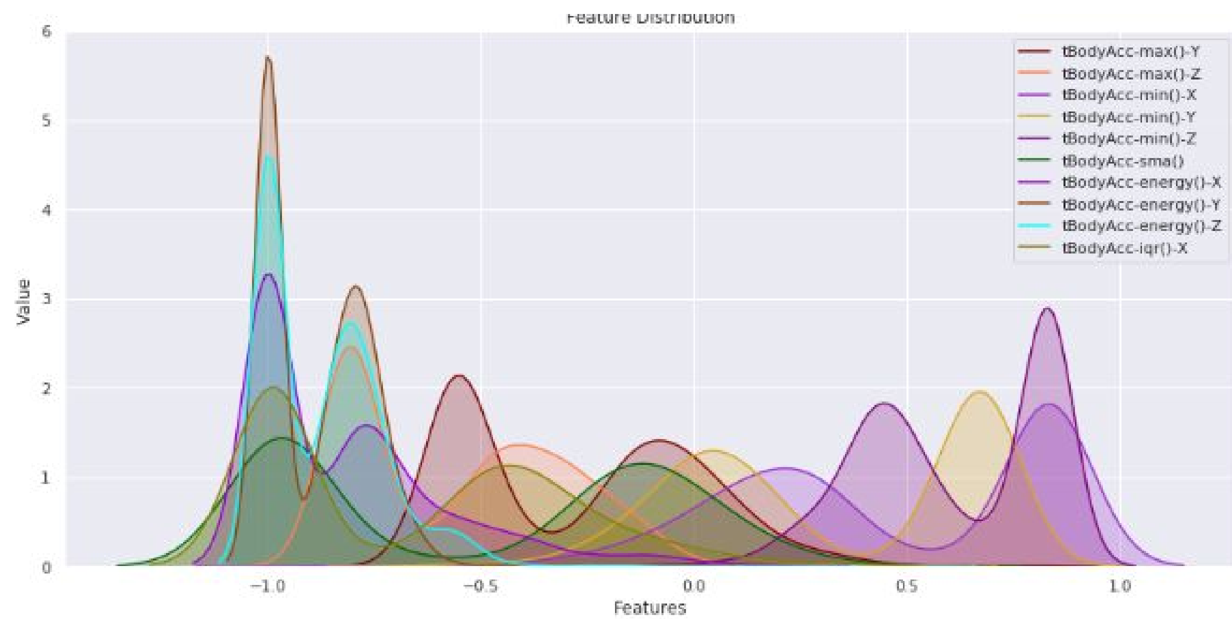
Static and Dynamic Activities :

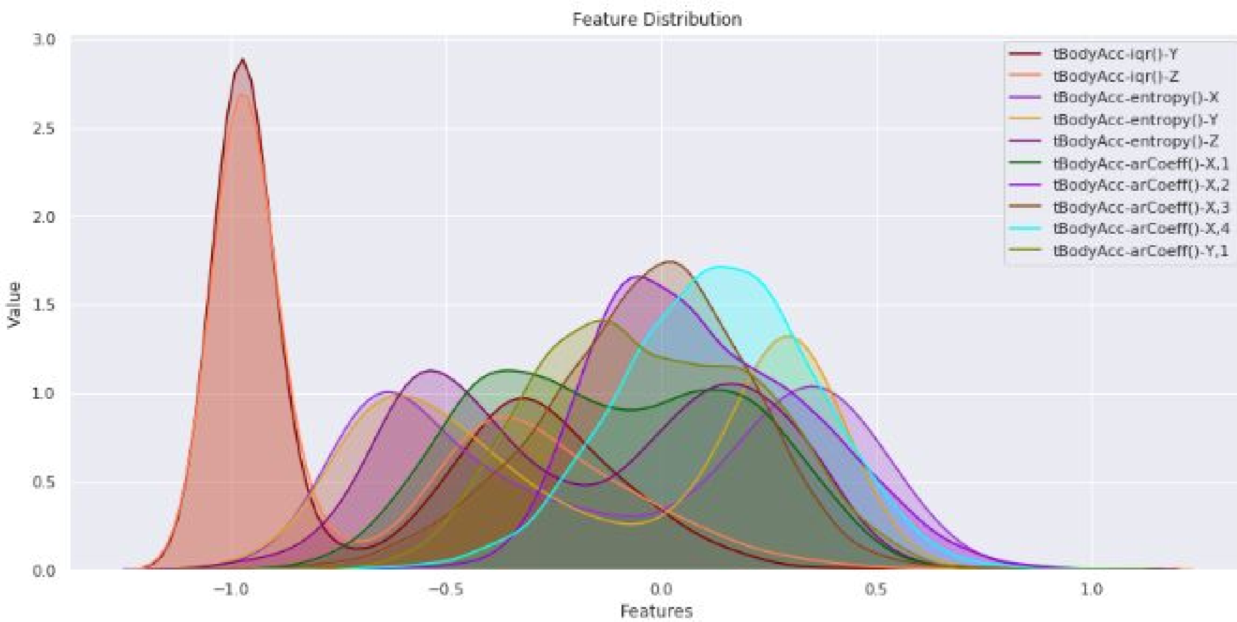
In static activities (sit, stand, lie down) motion information will not be very useful.In the dynamic activities (Walking, WalkingUpstairs,WalkingDownstairs) motion info will be significant.

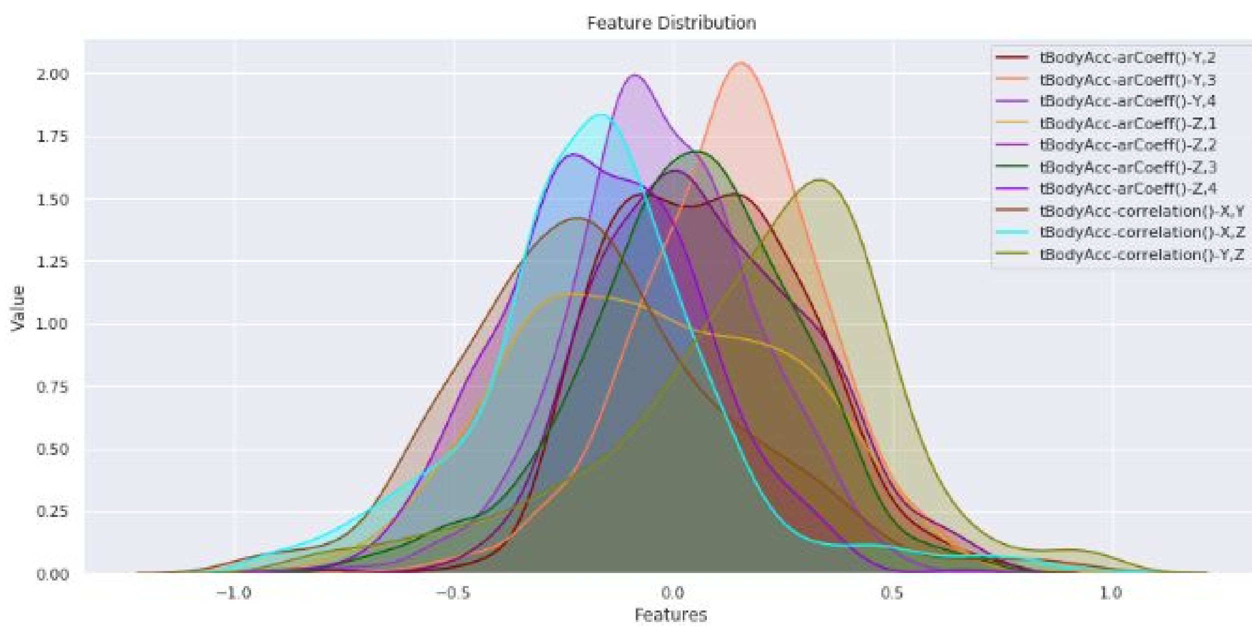


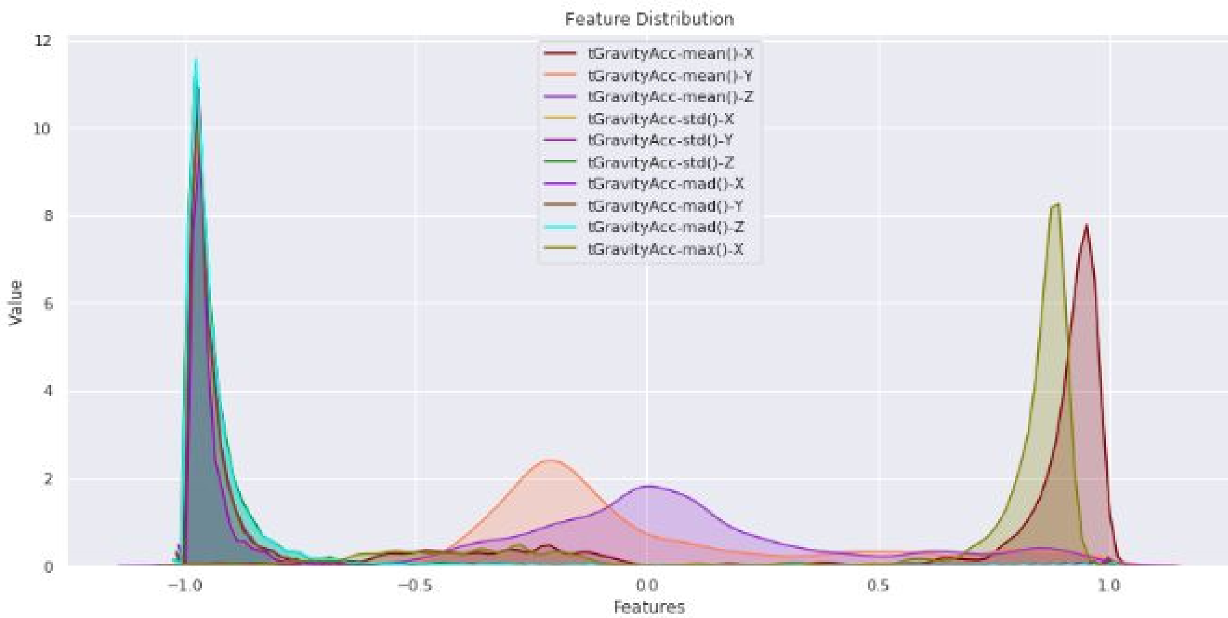
Feature visualization :


Feature Distribution
tBodyAcc-mean()-X
tBodyAcc-mean()-Y
tBodyAcc-mean()-Z
tBodyAcc-std()-X
tBodyAcc-std()-Y
tBodyAcc-std()-Z
tBodyAcc-mad()-X
tBodyAcc-mad()-Y
tBodyAcc-mad()-Z
tBodyAcc-max()-X
Value
Features


Feature Distribution
tBodyAcc-max()-Y
tBodyAcc-max()-Z
tBodyAcc-min()-X
tBodyAcc-min()-Y
tBodyAcc-min()-Z
tBodyAcc-sma()
tBodyAcc-energy()-X
tBodyAcc-energy()-Y
tBodyAcc-energy()-Z
tBodyAcc-iqr()-X
Value
Features

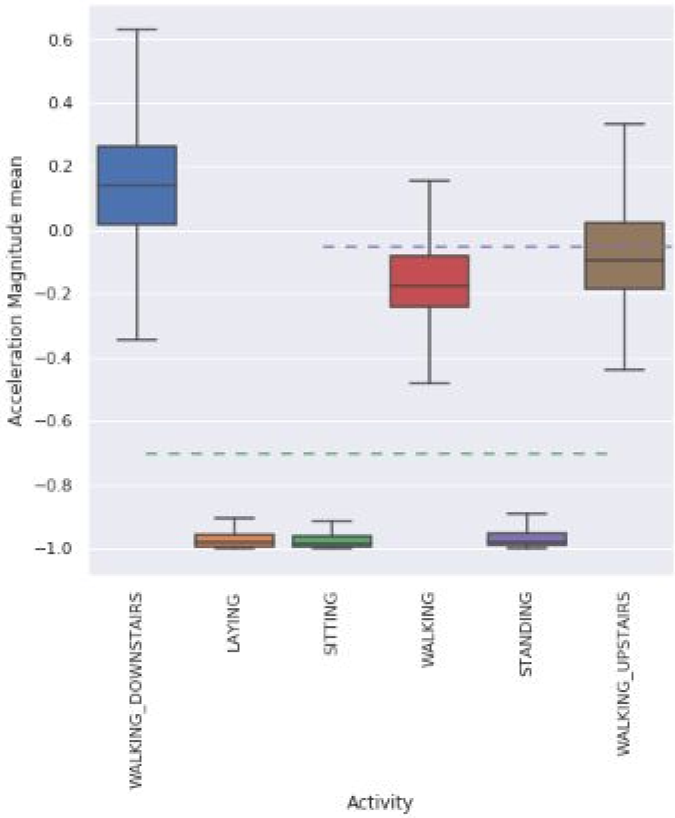

Feature Distribution
tBodyAcc-iqr()-Y
tBodyAcc-iqr()-Z
tBodyAcc-entropy()-X
tBodyAcc-entropy()-Y
tBodyAcc-entropy()-Z
tBodyAcc-arCoeff()-X,1
tBodyAcc-arCoeff()-X,2
tBodyAcc-arCoeff()-X,3
tBodyAcc-arCoeff()-X,4
tBodyAcc-arCoeff()-Y,1
Value
Features


Feature Distribution
tBodyAcc-arCoeff()-Y,2
tBodyAcc-arCoeff()-Y,3
tBodyAcc-arCoeff()-Y,4
tBodyAcc-arCoeff()-Z,1
tBodyAcc-arCoeff()-Z,2
tBodyAcc-arCoeff()-Z,3
tBodyAcc-arCoeff()-Z,4
tBodyAcc-correlation()-X,Y
tBodyAcc-correlation()-X,Z
tBodyAcc-correlation()-Y,Z
Value
Features



Relationship between Magnitude of an acceleration and Activity

If tBodyAccMag-mean is < -0.8 then the Activities are either Standing or Sitting or Laying.
If tBodyAccMag-mean is > -0.6 then the Activities are either Walking or WalkingDownstairs or WalkingUpstairs.
If tBodyAccMag-mean > 0.0 then the Activity is WalkingDownstairs.
We can classify 75% the Acitivity labels with some errors.



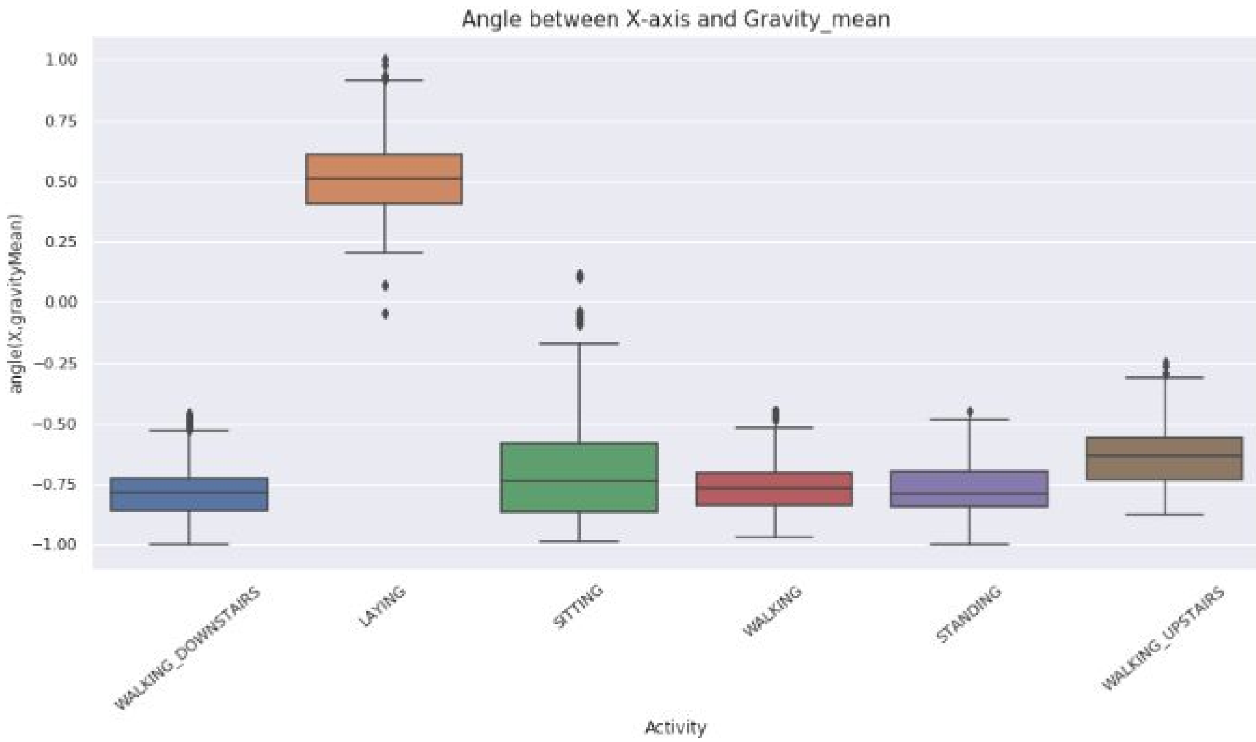
If tBodyAccMag-mean is < -0.8 then the Activities are either Standing or Sitting or Laying.
If tBodyAccMag-mean is > -0.6 then the Activities are either Walking or WalkingDownstairs or WalkingUpstairs.

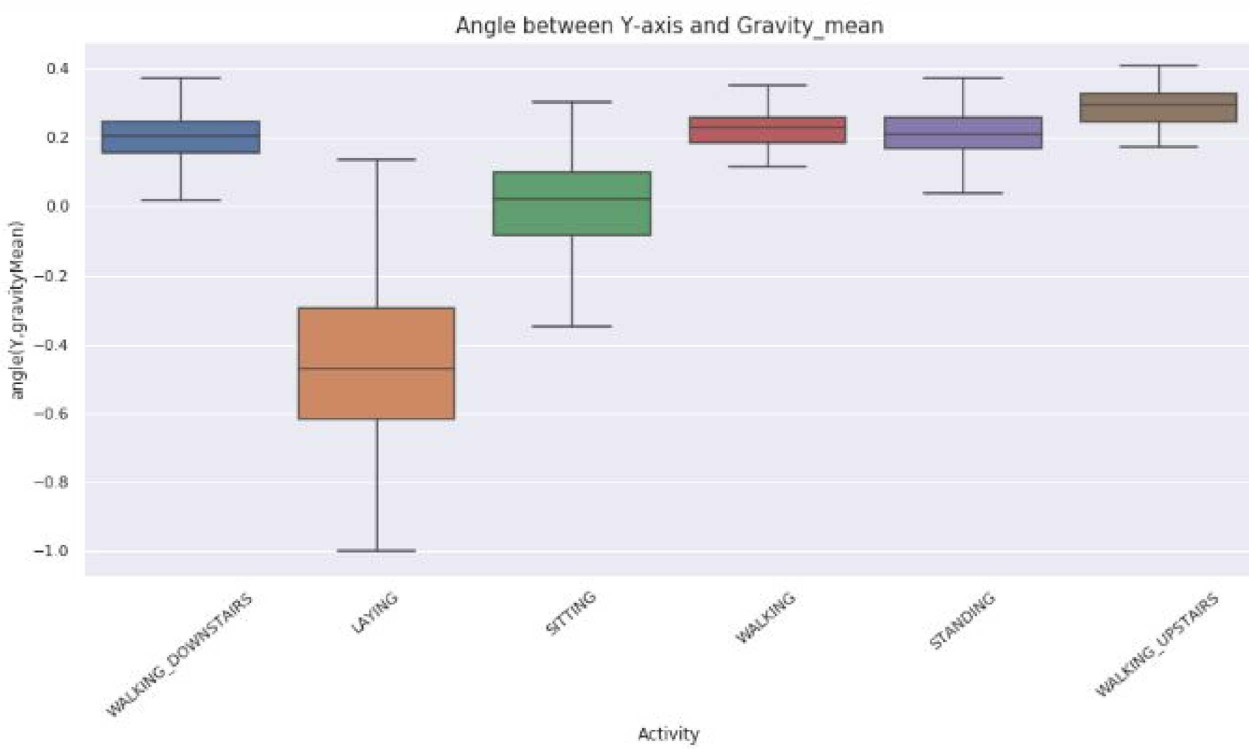
If tBodyAccMag-mean > 0.0 then the Activity is WalkingDownstairs.

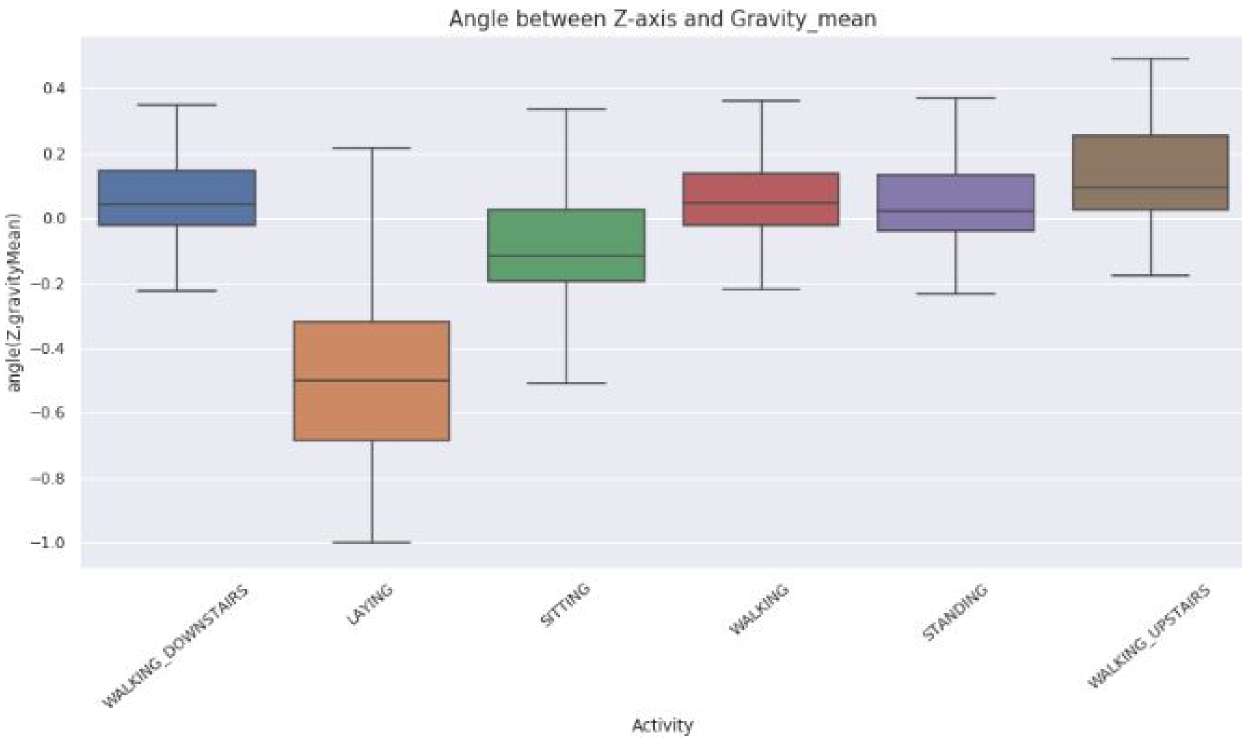
We can classify 75% the Acitivity labels with some errors.

Relationship between Position of Gravity Acceleration Components and the Target (activity) :

If angleX,gravityMean > 0 then Activity is Laying.We can classify all data points belonging to Laying activity with just a single if else statement.

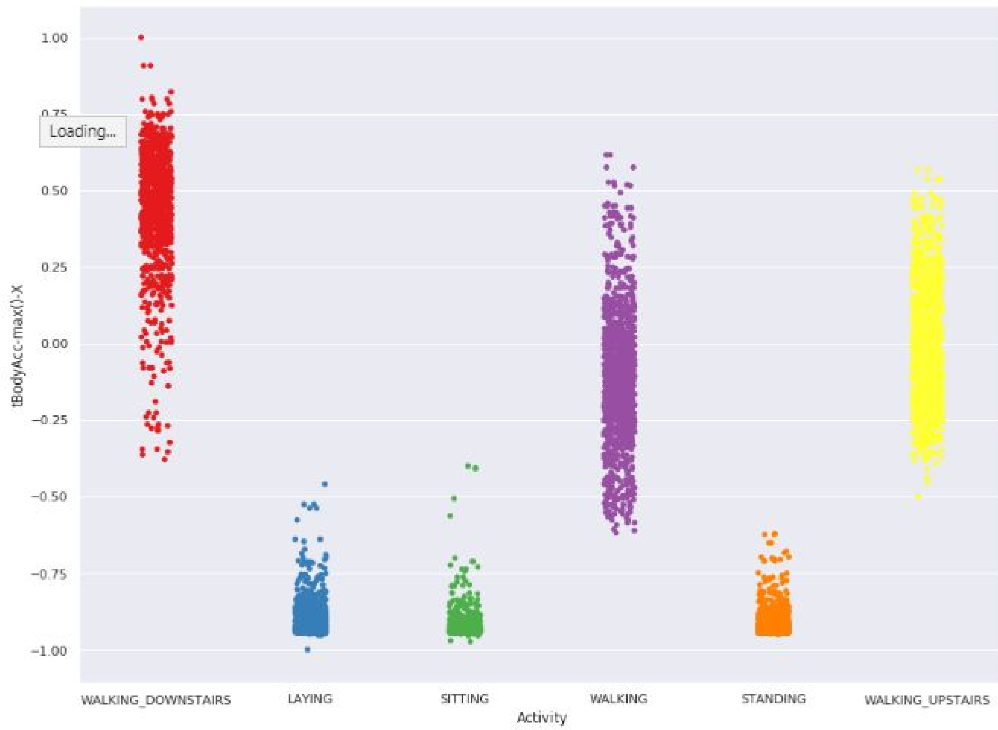

Angle between X-axis and Gravity_mean
angle(X,gravityMean)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
WALKING_DOWNSTAIRS
LAYING
SITTING
WALKING
STANDING
WALKING_UPSTAIRS
Activity

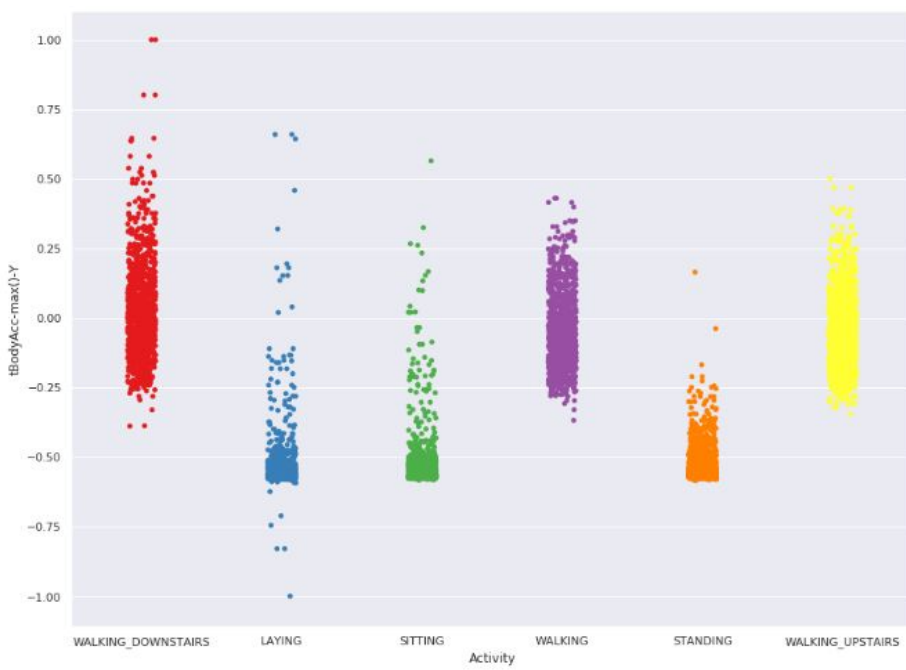

Angle between Y-axis and Gravity_mean
angle(Y,gravityMean)
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
WALKING_DOWNSTAIRS
LAYING
SITTING
WALKING
STANDING
WALKING_UPSTAIRS
Activity

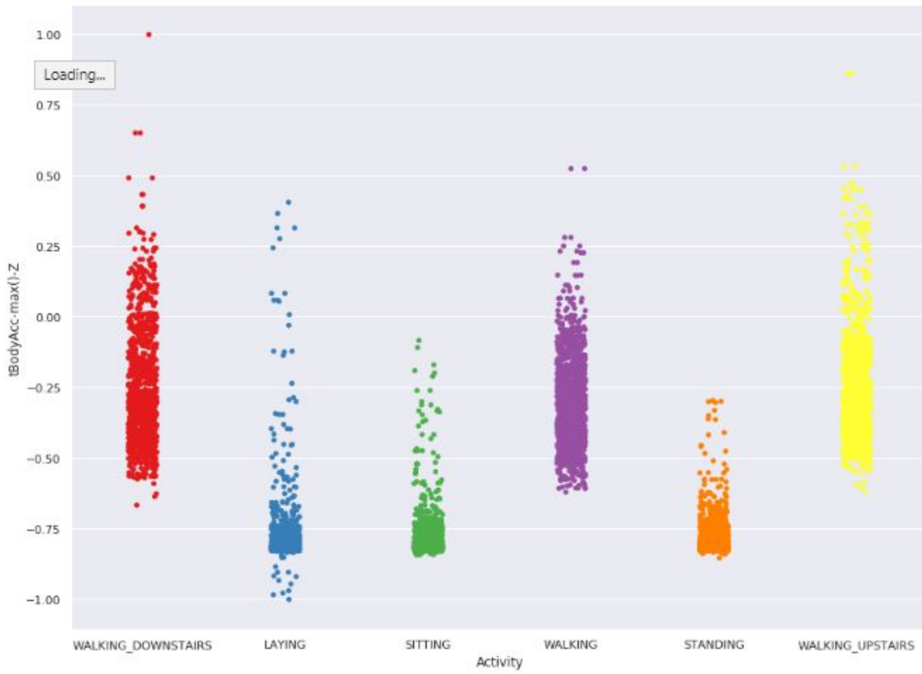


From the above visualization we can see that the Gravity Acceleration Component in X direction is more variable than in Y and Z direction. If we observe closely, for the active activities: WALKING, WALKING_UPSTAIRS and WALKING_DOWNSTAIRS the variation in X-axis more compared the passive activities: STANDING, SITTING and LAYING.

Comparing MaxBodyAcc in X Y Z direction







Comparing the above Strip-plot, we can clearly see that for Max Acc in all directions, the Passive activities are way below the active ones and for the active one the variation is also huge. If we notice in the first of the two plots, WALKING_DOWNSTAIRS has a greater value of Max body acceleration. This can be explained as when a person walks down the stairs, his body moves comparatively faster compared to when he is steadily walking or Walking_upstairs.