**Human Activity Recognition**

**Problem:**

smartphone---->>Sensors---->>accelerometer(acceleration) and gyroscope(angular velocity)

Predict Activities : Walking, Sitting,Walking Upstairs,Walking Downstairs,Laying down,standing..

Why???

Smart watches like fitbit , apple watch , we can predict how many hours we have slept , calories we have burned, monitor heart rate etc….

Diving deep:

Accelerometer(Triaxial x,y,z axis) measured over time

Gyroscope(Triaxial x,y,z axis) measured over time

This is a 6 time series data that we have

## **How data was recorded**

This dataset is collected from 30 persons(referred as subjects in this dataset), performing different activities with a smartphone to their waists. The data is recorded with the help of sensors (accelerometer and Gyroscope) in that smartphone. This experiment was video recorded to label the data manually.

By using the sensors(Gyroscope and accelerometer) in a smartphone, they have captured '3-axial linear acceleration'(*tAcc-XYZ*) from accelerometer and '3-axial angular velocity' (*tGyro-XYZ*) from Gyroscope with several variations.

prefix 't' in those metrics denotes time.

suffix 'XYZ' represents 3-axial signals in X , Y, and Z directions.

### **Feature names**

1. These sensor signals are preprocessed by applying noise filters and then sampled in fixed-width windows(sliding windows) of 2.56 seconds each with 50% overlap. ie., each window has 128 readings.
2. From Each window, a feature vector was obtianed by calculating variables from the time and frequency domain.  
   In our dataset, each datapoint represents a window with different readings
3. The accelertion signal was saperated into Body and Gravity acceleration signals(***tBodyAcc-XYZ*** and ***tGravityAcc-XYZ***) using some low pass filter with corner frequecy of 0.3Hz.
4. After that, the body linear acceleration and angular velocity were derived in time to obtian *jerk signals* (***tBodyAccJerk-XYZ*** and ***tBodyGyroJerk-XYZ***).
5. The magnitude of these 3-dimensional signals were calculated using the Euclidian norm. This magnitudes are represented as features with names like *tBodyAccMag*, *tGravityAccMag*, *tBodyAccJerkMag*, *tBodyGyroMag* and *tBodyGyroJerkMag*.
6. 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 esitmate some set of variables from the above signals. ie., We will estimate the following properties on each and every signal that we recoreded 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 the squares divided by the number of values.
   * ***iqr()***: Interquartile range
   * ***entropy()***: Signal entropy
   * ***arCoeff()***: Autorregresion 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
   * ***kurtosis()***: kurtosis of the frequency domain signal
   * ***bandsEnergy()***: Energy of a frequency interval within the 64 bins of the FFT of each window.
   * ***angle()***: Angle between to vectors.
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

### **Y\_Labels(Encoded)**

* In the dataset, Y\_labels are represented as numbers from 1 to 6 as their identifiers.
  + WALKING as **1**
  + WALKING\_UPSTAIRS as **2**
  + WALKING\_DOWNSTAIRS as **3**
  + SITTING as **4**
  + STANDING as **5**
  + LAYING as **6**

## **Train and test data were saperated**

* The readings from ***70%*** of the volunteers were taken as ***trianing data*** and remaining ***30%*** subjects recordings were taken for ***test data***

## **Data**

* All the data is present in 'UCI\_HAR\_dataset/' folder in present working directory.
  + Feature names are present in 'UCI\_HAR\_dataset/features.txt'
  + ***Train Data***
    - 'UCI\_HAR\_dataset/train/X\_train.txt'
    - 'UCI\_HAR\_dataset/train/subject\_train.txt'
    - 'UCI\_HAR\_dataset/train/y\_train.txt'
  + ***Test Data***
    - 'UCI\_HAR\_dataset/test/X\_test.txt'
    - 'UCI\_HAR\_dataset/test/subject\_test.txt'
    - 'UCI\_HAR\_dataset/test/y\_test.txt'

## **Data Size :**

27 MB

**Data Cleaning:**

**Check for Duplicates:**

No of duplicates in train: 0

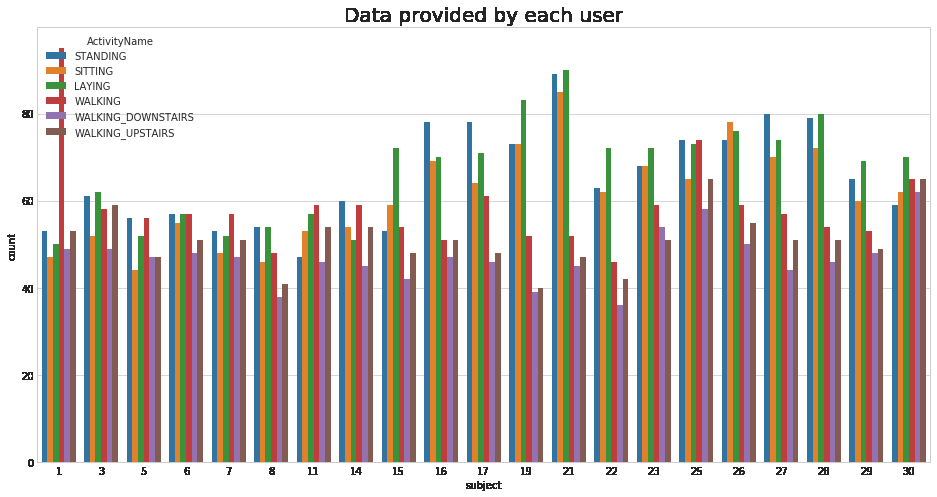
No of duplicates in test : 0

**Check for NAN/NULL values:**

We have 0 NaN/Null values in train

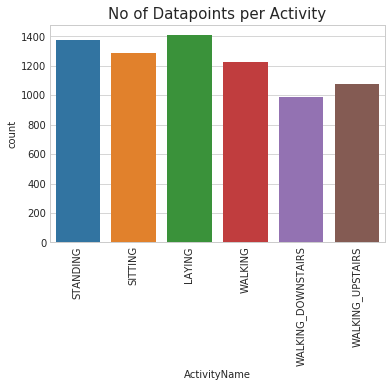
We have 0 NaN/Null values in test

**Check for Data Imbalance:**

****

**observation:**

**We have got almost same number of reading from all the subjects**

****

**Observation**

**Our data is well balanced (almost)**

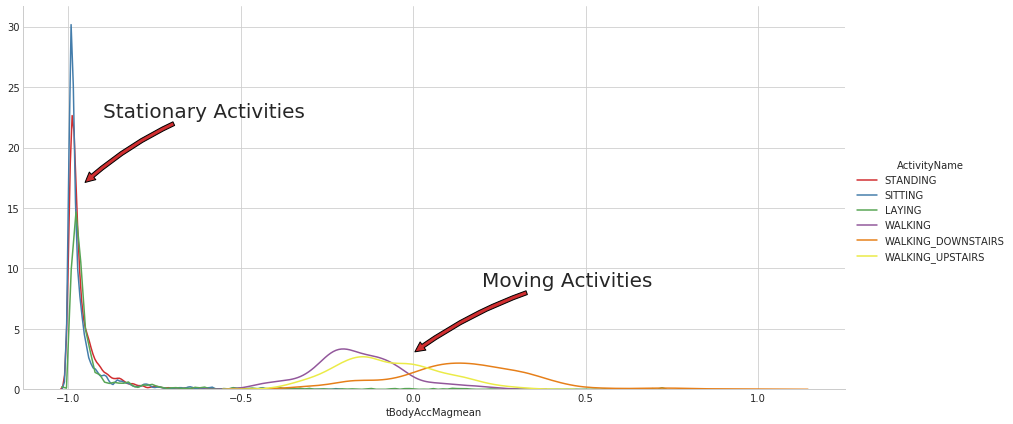
**Exploratory Data Analysis:**

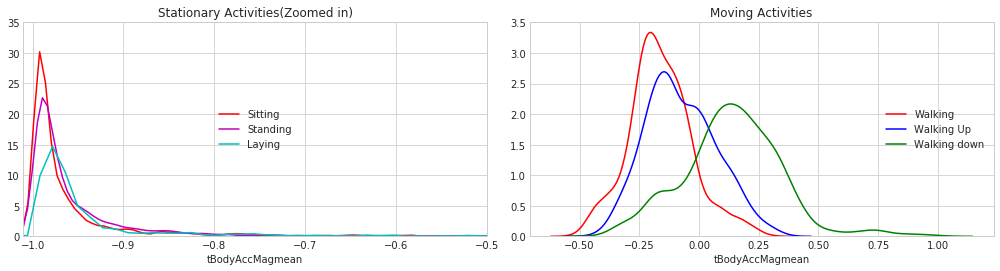
### **1. Featuring Engineering from Domain Knowledge**

* 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.

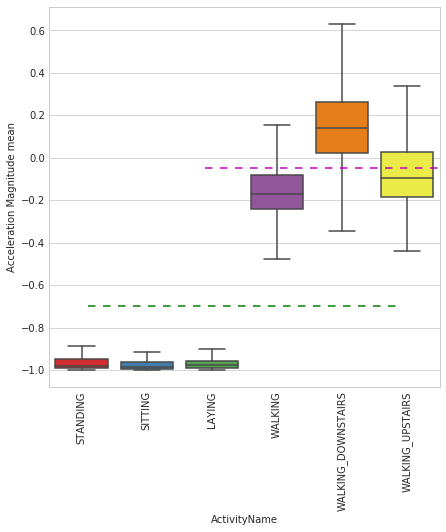
### **2. Stationary and Moving activities are completely different**

Doing Univariate Analysis on some features out of 561 features we have.

****

****

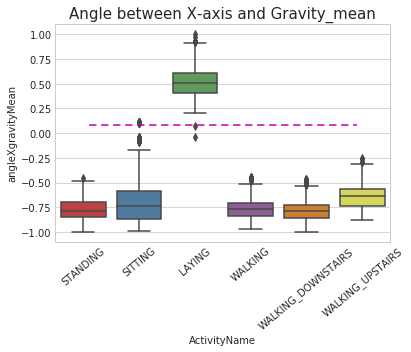
### **3. Magnitude of an acceleration can saperate it well**

****

**Observations:**

* 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 WalkingDownstairs or WalkingUpstairs.
* If tAccMean > 0.0 then the Activity is WalkingDownstairs.
* We can classify 75% the Acitivity labels with some errors**.**

### **4. Position of GravityAccelerationComponants also matters**

****

**Observations:**

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

# 

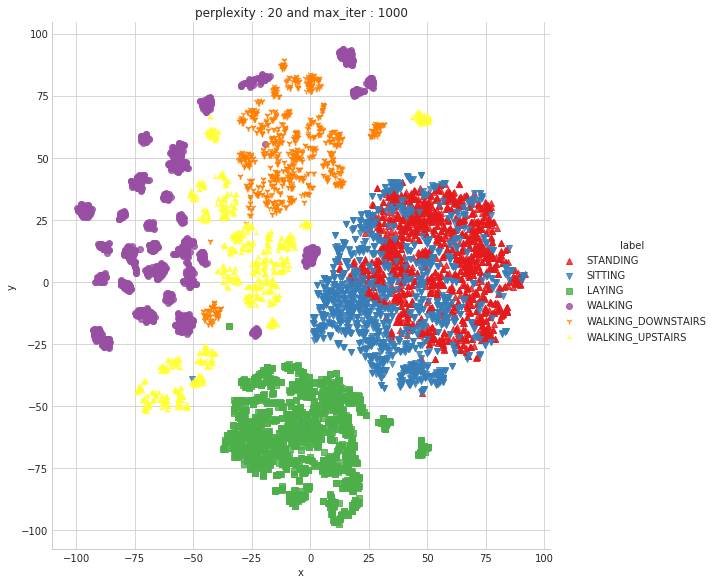
# 

# 

# 

# **Apply t-sne on the data**

T-sne is a dimensionality reduction algorithm for visualizatoion of Data with high Dimensions.

****

**Predicting on Engineered Data using Traditional ML tools:**

Accuracy Error

---------- --------

Logistic Regression : 96.27% 3.733%

Linear SVC : 96.61% 3.393%

rbf SVM classifier : 96.27% 3.733%

DecisionTree : 86.43% 13.57%

Random Forest : 91.31% 8.687%

GradientBoosting DT : 91.31% 8.687%

**Now Predicting using Raw Data:**

Training LSTM based Deep Learning model on raw Time Series data that is not engineered

We have 9 time series data:

3 time series accelerometer body,3 time series for accelerometer total and 3 for gyroscope body.

"body\_acc\_x",

"body\_acc\_y",

"body\_acc\_z",

"body\_gyro\_x",

"body\_gyro\_y",

"body\_gyro\_z",

"total\_acc\_x",

"total\_acc\_y",

"total\_acc\_z"

We have 9 time series of 128 time stamps which we are going to pass through our LSTM network.

Input Dimension = 9

Time stamp = 128

Total Train = 7K(approx)

**Defining Architecture:**

Layer (type) Output Shape Param #

=================================================================

lstm\_3 (LSTM) (None, 32) 5376

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dropout\_3 (Dropout) (None, 32) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_3 (Dense) (None, 6) 198

=================================================================

Total params: 5,574

Trainable params: 5,574

Non-trainable params: 0

**Confusion Matrix:**

**Pred LAYING SITTING STANDING WALKING WALKING\_DOWNSTAIRS \**

**True**

**LAYING 512 0 25 0 0**

**SITTING 3 410 75 0 0**

**STANDING 0 87 445 0 0**

**WALKING 0 0 0 481 2**

**WALKING\_DOWNSTAIRS 0 0 0 0 382**

**WALKING\_UPSTAIRS 0 0 0 2 18**

**Pred WALKING\_UPSTAIRS**

**True**

**LAYING 0**

**SITTING 3**

**STANDING 0**

**WALKING 13**

**WALKING\_DOWNSTAIRS 38**

**WALKING\_UPSTAIRS 451**

**Score:**

**[0.3087582236972612, 0.9097387173396675]**

* **With a simple 2 layer architecture we got 90.09% accuracy and a loss of 0.30**
* **We can further imporve the performace with Hyperparameter tuning**

**We can improve our model by adding more layers and more data this is a very basic LSTM architecture that i have designed**