Human Activity Recognition Using Smartphone Data

`Description`

The Human Activity Recognition database was built from the recordings of 30 study participants performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors. **The objective is to classify activities into one of the six activities performed.**

`Description of experiment`

- 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, WALKINGUPSTAIRS, WALKINGDOWNSTAIRS, 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.

`Steps`

1. **Importing necessary libraries**

2. **Loading data**

3. **Data preprocessing**

   i. Checking for duplicates

   ii. Checking for missing values

   iii. Checking for class imbalance

4. **Exploratory Data Analysis**

i. Analysing tBodyAccMag-mean feature

ii. Analysing Angle between X-axis and gravityMean feature

iii. Analysing Angle between Y-axis and gravityMean feature

iv. Visualizing data using t-SNE

5. **Build: Training and Testing Model set**

6. **Model Prediction and Evaluation**

i. Logistic regression model with Hyperparameter tuning and cross validation

ii. Linear SVM model with Hyperparameter tuning and cross validation

iii. Kernel SVM model with Hyperparameter tuning and cross validation

iv. Decision tree model with Hyperparameter tuning and cross validation

v. Random forest model with Hyperparameter tuning and cross validation

7. **Result**

`Dataset Link` : https://www.kaggle.com/datasets/uciml/human-activity-recognition-with-smartphones

1. Importing Required libraries

In [4]:

```
# Basic Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

# Analysis
from collections import Counter
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE

# Model
from sklearn.model_selection import RandomizedSearchCV

# Machine Learning Model
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

# Metrics
from sklearn.metrics import confusion_matrix, accuracy_score, classification_rep
```

2. Loading Dataset

```
In [6]: train = pd.read_csv('train.csv')
        test = pd.read_csv('test.csv')
```

```
In [7]: train.head()
```

Out[7]:

	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	tBodyAcc-mad()-X	tB
0	0.288585	-0.020294	-0.132905	-0.995279	-0.983111	-0.913526	-0.995112	-(
1	0.278419	-0.016411	-0.123520	-0.998245	-0.975300	-0.960322	-0.998807	-(
2	0.279653	-0.019467	-0.113462	-0.995380	-0.967187	-0.978944	-0.996520	-(
3	0.279174	-0.026201	-0.123283	-0.996091	-0.983403	-0.990675	-0.997099	-(
4	0.276629	-0.016570	-0.115362	-0.998139	-0.980817	-0.990482	-0.998321	-(

5 rows × 563 columns

```
In [9]: test.head()
```

Out[9]:

	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	tBodyAcc-mad()-X	tB
0	0.257178	-0.023285	-0.014654	-0.938404	-0.920091	-0.667683	-0.952501	-(
1	0.286027	-0.013163	-0.119083	-0.975415	-0.967458	-0.944958	-0.986799	-(
2	0.275485	-0.026050	-0.118152	-0.993819	-0.969926	-0.962748	-0.994403	-(
3	0.270298	-0.032614	-0.117520	-0.994743	-0.973268	-0.967091	-0.995274	-(
4	0.274833	-0.027848	-0.129527	-0.993852	-0.967445	-0.978295	-0.994111	-(

5 rows × 563 columns

```
In [10]: train.subject.value_counts()
```

Out[10]:

```
subject
25    409
21    408
26    392
30    383
28    382
27    376
23    372
17    368
16    366
19    360
1     347
29    344
3     341
15    328
6     325
14    323
22    321
11    316
7     308
5     302
8     281
Name: count, dtype: int64
```

In [23]:

```
# Shape of Train Dataset
print('Number of rows in traing dataset: ',train.shape[0])
print('Number of columns in traing dataset: ',train.shape[1])
```

```
Number of rows in traing dataset:  7352
Number of columns in traing dataset:  563
```

3. Dara Wrangling / Pre-processing

i. Checking for the Duplicates

In [24]:

```
print('Number of duplicates in train : ', train.duplicated().sum())
print('Number of duplicates in test : ', test.duplicated().sum())
```

```
Number of duplicates in train :  0
Number of duplicates in test :  0
```

ii. Checking for Null Values

In [25]:

```
print('Total number of missing values in train : ', train.isna().values.sum())
print('Total number of missing values in train : ', test.isna().values.sum())
```

```
Total number of missing values in train :  0
Total number of missing values in train :  0
```

In [26]:

```
# Null value Percentage
def find_dirty_values(data):
    dtypes = pd.DataFrame(data.dtypes,columns=["Data Type"])
    dtypes["Unique Values"]=data.nunique().sort_values(ascending=True)
    dtypes["Null Values"]=data.isnull().sum()
    dtypes["% null Values"]=data.isnull().sum()/len(data)
    return dtypes.sort_values(by="Null Values" , ascending=False).style.backgrou
```

```
null_data = find_dirty_values(train)
null_data
```

Out[26]:

	Data Type	Unique Values	Null Values	% null Values
tBodyAcc-mean()-X	float64	7347	0	0.000000
fBodyAccJerk-kurtosis()-Y	float64	7351	0	0.000000
fBodyAccJerk-meanFreq()-X	float64	7351	0	0.000000
fBodyAccJerk-meanFreq()-Y	float64	7352	0	0.000000
fBodyAccJerk-meanFreq()-Z	float64	7352	0	0.000000
fBodyAccJerk-skewness()-X	float64	7352	0	0.000000
fBodyAccJerk-kurtosis()-X	float64	7352	0	0.000000
fBodyAccJerk-skewness()-Y	float64	7352	0	0.000000
fBodyAccJerk-skewness()-Z	float64	7351	0	0.000000
fBodyAccJerk-maxInds-Y	float64	48	0	0.000000
fBodyAccJerk-kurtosis()-Z	float64	7351	0	0.000000
fBodyAccJerk-bandsEnergy()-1,8	float64	6461	0	0.000000
fBodyAccJerk-bandsEnergy()-9,16	float64	6958	0	0.000000
fBodyAccJerk-bandsEnergy()-17,24	float64	7103	0	0.000000
fBodyAccJerk-bandsEnergy()-25,32	float64	7165	0	0.000000
fBodyAccJerk-bandsEnergy()-33,40	float64	7138	0	0.000000
fBodyAccJerk-maxInds-Z	float64	49	0	0.000000
fBodyAccJerk-maxInds-X	float64	48	0	0.000000
fBodyAccJerk-bandsEnergy()-25,48.2	float64	7189	0	0.000000
fBodyAccJerk-energy()-X	float64	7101	0	0.000000
fBodyAccJerk-max()-Y	float64	7351	0	0.000000
fBodyAccJerk-max()-Z	float64	7348	0	0.000000
fBodyAccJerk-min()-X	float64	7344	0	0.000000
fBodyAccJerk-min()-Y	float64	7349	0	0.000000
fBodyAccJerk-min()-Z	float64	7348	0	0.000000
fBodyAccJerk-sma()	float64	7350	0	0.000000
fBodyAccJerk-energy()-Y	float64	7224	0	0.000000
fBodyAccJerk-entropy()-Z	float64	3325	0	0.000000
fBodyAccJerk-energy()-Z	float64	7207	0	0.000000
fBodyAccJerk-iqr()-X	float64	7347	0	0.000000
fBodyAccJerk-iqr()-Y	float64	7351	0	0.000000
fBodyAccJerk-iqr()-Z	float64	7350	0	0.000000

	Data Type	Unique Values	Null Values	% null Values
fBodyAccJerk-entropy()-X	float64	3313	0	0.000000
fBodyAccJerk-entropy()-Y	float64	3359	0	0.000000
fBodyAccJerk-bandsEnergy()-41,48	float64	7150	0	0.000000
fBodyAccJerk-bandsEnergy()-49,56	float64	7000	0	0.000000
fBodyAccJerk-bandsEnergy()-57,64	float64	5865	0	0.000000
fBodyAccJerk-bandsEnergy()-33,40.2	float64	7212	0	0.000000
fBodyAccJerk-bandsEnergy()-1,24.1	float64	7206	0	0.000000
fBodyAccJerk-bandsEnergy()-25,48.1	float64	7240	0	0.000000
fBodyAccJerk-bandsEnergy()-1,8.2	float64	7252	0	0.000000
fBodyAccJerk-bandsEnergy()-9,16.2	float64	7240	0	0.000000
fBodyAccJerk-bandsEnergy()-17,24.2	float64	7203	0	0.000000
fBodyAccJerk-bandsEnergy()-25,32.2	float64	7177	0	0.000000
fBodyAccJerk-bandsEnergy()-41,48.2	float64	7275	0	0.000000
fBodyAccJerk-bandsEnergy()-1,16	float64	6849	0	0.000000
fBodyAccJerk-bandsEnergy()-49,56.2	float64	7300	0	0.000000
fBodyAccJerk-bandsEnergy()-57,64.2	float64	7045	0	0.000000
fBodyAccJerk-bandsEnergy()-1,16.2	float64	7260	0	0.000000
fBodyAccJerk-bandsEnergy()-17,32.2	float64	7175	0	0.000000
fBodyAccJerk-bandsEnergy()-33,48.2	float64	7244	0	0.000000
fBodyAccJerk-bandsEnergy()-49,64.2	float64	7299	0	0.000000
fBodyAccJerk-bandsEnergy()-49,64.1	float64	7233	0	0.000000
fBodyAccJerk-bandsEnergy()-33,48.1	float64	7271	0	0.000000
fBodyAccJerk-bandsEnergy()-17,32.1	float64	7225	0	0.000000
fBodyAccJerk-bandsEnergy()-1,16.1	float64	7178	0	0.000000
fBodyAccJerk-bandsEnergy()-57,64.1	float64	6745	0	0.000000
fBodyAccJerk-bandsEnergy()-49,56.1	float64	7256	0	0.000000
fBodyAccJerk-bandsEnergy()-41,48.1	float64	7285	0	0.000000
fBodyAccJerk-bandsEnergy()-33,40.1	float64	7236	0	0.000000
fBodyAccJerk-bandsEnergy()-25,32.1	float64	7220	0	0.000000
fBodyAccJerk-bandsEnergy()-17,24.1	float64	7226	0	0.000000
fBodyAccJerk-bandsEnergy()-9,16.1	float64	7169	0	0.000000
fBodyAccJerk-bandsEnergy()-1,8.1	float64	7218	0	0.000000

	Data Type	Unique Values	Null Values	% null Values
fBodyAccJerk-bandsEnergy()-25,48	float64	7202	0	0.000000
fBodyAccJerk-bandsEnergy()-1,24	float64	7032	0	0.000000
fBodyAccJerk-bandsEnergy()-49,64	float64	7031	0	0.000000
fBodyAccJerk-bandsEnergy()-33,48	float64	7123	0	0.000000
fBodyAccJerk-bandsEnergy()-17,32	float64	7161	0	0.000000
fBodyAccJerk-max()-X	float64	7348	0	0.000000
fBodyAccJerk-mad()-Z	float64	7349	0	0.000000
fBodyAccJerk-mad()-Y	float64	7349	0	0.000000
fBodyAcc-bandsEnergy()-41,48	float64	7114	0	0.000000
fBodyAcc-kurtosis()-Z	float64	7352	0	0.000000
fBodyAcc-bandsEnergy()-1,8	float64	6876	0	0.000000
fBodyAcc-bandsEnergy()-9,16	float64	6948	0	0.000000
fBodyAcc-bandsEnergy()-17,24	float64	7134	0	0.000000
fBodyAcc-bandsEnergy()-25,32	float64	7181	0	0.000000
fBodyAcc-bandsEnergy()-33,40	float64	7125	0	0.000000
fBodyAcc-bandsEnergy()-49,56	float64	6957	0	0.000000
fBodyAcc-bandsEnergy()-1,8.1	float64	7293	0	0.000000
fBodyAcc-bandsEnergy()-57,64	float64	6123	0	0.000000
fBodyAcc-bandsEnergy()-1,16	float64	6950	0	0.000000
fBodyAcc-bandsEnergy()-17,32	float64	7152	0	0.000000
fBodyAcc-bandsEnergy()-33,48	float64	7122	0	0.000000
fBodyAcc-bandsEnergy()-49,64	float64	6866	0	0.000000
fBodyAcc-bandsEnergy()-1,24	float64	6989	0	0.000000
fBodyAcc-skewness()-Z	float64	7351	0	0.000000
fBodyAcc-kurtosis()-Y	float64	7352	0	0.000000
fBodyAcc-skewness()-Y	float64	7352	0	0.000000
fBodyAcc-kurtosis()-X	float64	7351	0	0.000000
fBodyAcc-skewness()-X	float64	7350	0	0.000000
fBodyAcc-meanFreq()-Z	float64	7352	0	0.000000
fBodyAcc-meanFreq()-Y	float64	7352	0	0.000000
fBodyAcc-meanFreq()-X	float64	7352	0	0.000000
fBodyAcc-maxInds-Z	float64	26	0	0.000000

	Data Type	Unique Values	Null Values	% null Values
fBodyAcc-maxInds-Y	float64	26	0	0.000000
fBodyAcc-maxInds-X	float64	29	0	0.000000
fBodyAcc-entropy()-Z	float64	3811	0	0.000000
fBodyAcc-entropy()-Y	float64	3801	0	0.000000
fBodyAcc-entropy()-X	float64	3602	0	0.000000
fBodyAcc-iqr()-Z	float64	7350	0	0.000000
fBodyAcc-iqr()-Y	float64	7347	0	0.000000
fBodyAcc-iqr()-X	float64	7349	0	0.000000
fBodyAcc-bandsEnergy()-25,48	float64	7180	0	0.000000
fBodyAcc-bandsEnergy()-9,16.1	float64	7179	0	0.000000
fBodyAccJerk-mad()-X	float64	7348	0	0.000000
fBodyAcc-bandsEnergy()-1,24.2	float64	7301	0	0.000000
fBodyAcc-bandsEnergy()-49,56.2	float64	7279	0	0.000000
fBodyAcc-bandsEnergy()-57,64.2	float64	7090	0	0.000000
fBodyAcc-bandsEnergy()-1,16.2	float64	7300	0	0.000000
fBodyAcc-bandsEnergy()-17,32.2	float64	7205	0	0.000000
fBodyAcc-bandsEnergy()-33,48.2	float64	7243	0	0.000000
fBodyAcc-bandsEnergy()-49,64.2	float64	7277	0	0.000000
fBodyAcc-bandsEnergy()-25,48.2	float64	7186	0	0.000000
fBodyAcc-bandsEnergy()-17,24.1	float64	7224	0	0.000000
fBodyAccJerk-mean()-X	float64	7348	0	0.000000
fBodyAccJerk-mean()-Y	float64	7350	0	0.000000
fBodyAccJerk-mean()-Z	float64	7349	0	0.000000
fBodyAccJerk-std()-X	float64	7349	0	0.000000
fBodyAccJerk-std()-Y	float64	7348	0	0.000000
fBodyAccJerk-std()-Z	float64	7347	0	0.000000
fBodyAcc-bandsEnergy()-41,48.2	float64	7289	0	0.000000
fBodyAcc-bandsEnergy()-33,40.2	float64	7235	0	0.000000
fBodyAcc-bandsEnergy()-25,32.2	float64	7181	0	0.000000
fBodyAcc-bandsEnergy()-17,24.2	float64	7237	0	0.000000
fBodyAcc-bandsEnergy()-9,16.2	float64	7262	0	0.000000
fBodyAcc-bandsEnergy()-1,8.2	float64	7316	0	0.000000

	Data Type	Unique Values	Null Values	% null Values
fBodyAcc-bandsEnergy()-25,48.1	float64	7256	0	0.000000
fBodyAcc-bandsEnergy()-1,24.1	float64	7307	0	0.000000
fBodyAcc-bandsEnergy()-49,64.1	float64	7231	0	0.000000
fBodyAcc-bandsEnergy()-33,48.1	float64	7280	0	0.000000
fBodyAcc-bandsEnergy()-17,32.1	float64	7206	0	0.000000
fBodyAcc-bandsEnergy()-1,16.1	float64	7287	0	0.000000
fBodyAcc-bandsEnergy()-57,64.1	float64	6904	0	0.000000
fBodyAcc-bandsEnergy()-49,56.1	float64	7260	0	0.000000
fBodyAcc-bandsEnergy()-41,48.1	float64	7279	0	0.000000
fBodyAcc-bandsEnergy()-33,40.1	float64	7288	0	0.000000
fBodyAcc-bandsEnergy()-25,32.1	float64	7251	0	0.000000
fBodyAccJerk-bandsEnergy()-1,24.2	float64	7239	0	0.000000
fBodyGyro-mean()-X	float64	7351	0	0.000000
fBodyAcc-energy()-Y	float64	7298	0	0.000000
fBodyBodyAccJerkMag-min()	float64	7346	0	0.000000
fBodyAccMag-skewness()	float64	7352	0	0.000000
fBodyAccMag-kurtosis()	float64	7352	0	0.000000
fBodyBodyAccJerkMag-mean()	float64	7345	0	0.000000
fBodyBodyAccJerkMag-std()	float64	7350	0	0.000000
fBodyBodyAccJerkMag-mad()	float64	7348	0	0.000000
fBodyBodyAccJerkMag-max()	float64	7348	0	0.000000
fBodyBodyAccJerkMag-sma()	float64	7345	0	0.000000
fBodyAccMag-maxInds	float64	29	0	0.000000
fBodyBodyAccJerkMag-energy()	float64	7195	0	0.000000
fBodyBodyAccJerkMag-iqr()	float64	7347	0	0.000000
fBodyBodyAccJerkMag-entropy()	float64	3396	0	0.000000
fBodyBodyAccJerkMag-maxInds	float64	57	0	0.000000
fBodyBodyAccJerkMag-meanFreq()	float64	7352	0	0.000000
fBodyBodyAccJerkMag-skewness()	float64	7352	0	0.000000
fBodyAccMag-meanFreq()	float64	7352	0	0.000000
fBodyAccMag-entropy()	float64	3828	0	0.000000
fBodyGyro-mean()-Y	float64	7349	0	0.000000

	Data Type	Unique Values	Null Values	% null Values
fBodyGyro-bandsEnergy()-25,48.2	float64	6912	0	0.000000
fBodyGyro-bandsEnergy()-57,64.2	float64	6560	0	0.000000
fBodyGyro-bandsEnergy()-1,16.2	float64	7183	0	0.000000
fBodyGyro-bandsEnergy()-17,32.2	float64	7025	0	0.000000
fBodyGyro-bandsEnergy()-33,48.2	float64	7027	0	0.000000
fBodyGyro-bandsEnergy()-49,64.2	float64	7101	0	0.000000
fBodyGyro-bandsEnergy()-1,24.2	float64	7201	0	0.000000
fBodyAccMag-mean()	float64	7351	0	0.000000
fBodyAccMag-iqr()	float64	7351	0	0.000000
fBodyAccMag-std()	float64	7352	0	0.000000
fBodyAccMag-mad()	float64	7349	0	0.000000
fBodyAccMag-max()	float64	7350	0	0.000000
fBodyAccMag-min()	float64	7348	0	0.000000
fBodyAccMag-sma()	float64	7351	0	0.000000
fBodyAccMag-energy()	float64	7291	0	0.000000
fBodyBodyAccJerkMag-kurtosis()	float64	7352	0	0.000000
fBodyBodyGyroMag-mean()	float64	7351	0	0.000000
fBodyBodyGyroMag-std()	float64	7350	0	0.000000
angle(tBodyAccMean,gravity)	float64	7352	0	0.000000
fBodyBodyGyroJerkMag-iqr()	float64	7347	0	0.000000
fBodyBodyGyroJerkMag-entropy()	float64	3706	0	0.000000
fBodyBodyGyroJerkMag-maxInds	float64	52	0	0.000000
fBodyBodyGyroJerkMag-meanFreq()	float64	7352	0	0.000000
fBodyBodyGyroJerkMag-skewness()	float64	7351	0	0.000000
fBodyBodyGyroJerkMag-kurtosis()	float64	7352	0	0.000000
angle(tBodyAccJerkMean),gravityMean)	float64	7352	0	0.000000
fBodyBodyGyroMag-mad()	float64	7350	0	0.000000
angle(tBodyGyroMean,gravityMean)	float64	7352	0	0.000000
angle(tBodyGyroJerkMean,gravityMean)	float64	7352	0	0.000000
angle(X,gravityMean)	float64	7352	0	0.000000
angle(Y,gravityMean)	float64	7352	0	0.000000
angle(Z,gravityMean)	float64	7352	0	0.000000

	Data Type	Unique Values	Null Values	% null Values
subject	int64	21	0	0.000000
fBodyBodyGyroJerkMag-energy()	float64	6907	0	0.000000
fBodyBodyGyroJerkMag-sma()	float64	7347	0	0.000000
fBodyBodyGyroJerkMag-min()	float64	7348	0	0.000000
fBodyBodyGyroJerkMag-max()	float64	7349	0	0.000000
fBodyBodyGyroJerkMag-mad()	float64	7349	0	0.000000
fBodyBodyGyroJerkMag-std()	float64	7349	0	0.000000
fBodyBodyGyroJerkMag-mean()	float64	7347	0	0.000000
fBodyBodyGyroMag-kurtosis()	float64	7352	0	0.000000
fBodyBodyGyroMag-skewness()	float64	7352	0	0.000000
fBodyBodyGyroMag-meanFreq()	float64	7352	0	0.000000
fBodyBodyGyroMag-maxInds	float64	27	0	0.000000
fBodyBodyGyroMag-entropy()	float64	4458	0	0.000000
fBodyBodyGyroMag-iqr()	float64	7348	0	0.000000
fBodyBodyGyroMag-energy()	float64	7255	0	0.000000
fBodyBodyGyroMag-sma()	float64	7351	0	0.000000
fBodyBodyGyroMag-min()	float64	7347	0	0.000000
fBodyBodyGyroMag-max()	float64	7350	0	0.000000
fBodyGyro-bandsEnergy()-49,56.2	float64	7182	0	0.000000
fBodyGyro-bandsEnergy()-41,48.2	float64	7141	0	0.000000
fBodyGyro-bandsEnergy()-33,40.2	float64	7033	0	0.000000
fBodyGyro-maxInds-X	float64	27	0	0.000000
fBodyGyro-iqr()-X	float64	7350	0	0.000000
fBodyGyro-iqr()-Y	float64	7350	0	0.000000
fBodyGyro-iqr()-Z	float64	7347	0	0.000000
fBodyGyro-entropy()-X	float64	4485	0	0.000000
fBodyGyro-entropy()-Y	float64	4495	0	0.000000
fBodyGyro-entropy()-Z	float64	4288	0	0.000000
fBodyGyro-maxInds-Y	float64	29	0	0.000000
fBodyGyro-kurtosis()-Y	float64	7352	0	0.000000
fBodyGyro-maxInds-Z	float64	25	0	0.000000
fBodyGyro-meanFreq()-X	float64	7352	0	0.000000

	Data Type	Unique Values	Null Values	% null Values
fBodyGyro-meanFreq()-Y	float64	7352	0	0.000000
fBodyGyro-meanFreq()-Z	float64	7352	0	0.000000
fBodyGyro-skewness()-X	float64	7352	0	0.000000
fBodyGyro-kurtosis()-X	float64	7352	0	0.000000
fBodyGyro-energy()-Z	float64	7220	0	0.000000
fBodyGyro-energy()-Y	float64	7223	0	0.000000
fBodyGyro-energy()-X	float64	7089	0	0.000000
fBodyGyro-sma()	float64	7351	0	0.000000
fBodyGyro-min()-Z	float64	7344	0	0.000000
fBodyGyro-min()-Y	float64	7347	0	0.000000
fBodyGyro-min()-X	float64	7342	0	0.000000
fBodyGyro-max()-Z	float64	7348	0	0.000000
fBodyGyro-max()-Y	float64	7352	0	0.000000
fBodyGyro-max()-X	float64	7348	0	0.000000
fBodyGyro-mad()-Z	float64	7352	0	0.000000
fBodyGyro-mad()-Y	float64	7351	0	0.000000
fBodyGyro-mad()-X	float64	7351	0	0.000000
fBodyGyro-std()-Z	float64	7352	0	0.000000
fBodyGyro-std()-Y	float64	7350	0	0.000000
fBodyGyro-std()-X	float64	7351	0	0.000000
fBodyGyro-mean()-Z	float64	7350	0	0.000000
fBodyGyro-skewness()-Y	float64	7352	0	0.000000
fBodyGyro-skewness()-Z	float64	7352	0	0.000000
fBodyGyro-bandsEnergy()-25,32.2	float64	6907	0	0.000000
fBodyGyro-bandsEnergy()-17,32.1	float64	6706	0	0.000000
fBodyGyro-bandsEnergy()-25,32.1	float64	6671	0	0.000000
fBodyGyro-bandsEnergy()-33,40.1	float64	6731	0	0.000000
fBodyGyro-bandsEnergy()-41,48.1	float64	7013	0	0.000000
fBodyGyro-bandsEnergy()-49,56.1	float64	7043	0	0.000000
fBodyGyro-bandsEnergy()-57,64.1	float64	6440	0	0.000000
fBodyGyro-bandsEnergy()-1,16.1	float64	7228	0	0.000000
fBodyGyro-bandsEnergy()-33,48.1	float64	6800	0	0.000000

	Data Type	Unique Values	Null Values	% null Values
fBodyGyro-kurtosis()-Z	float64	7352	0	0.000000
fBodyGyro-bandsEnergy()-49,64.1	float64	7018	0	0.000000
fBodyGyro-bandsEnergy()-1,24.1	float64	7231	0	0.000000
fBodyGyro-bandsEnergy()-25,48.1	float64	6769	0	0.000000
fBodyGyro-bandsEnergy()-1,8.2	float64	7186	0	0.000000
fBodyGyro-bandsEnergy()-9,16.2	float64	6957	0	0.000000
fBodyGyro-bandsEnergy()-17,24.2	float64	6962	0	0.000000
fBodyGyro-bandsEnergy()-17,24.1	float64	6560	0	0.000000
fBodyGyro-bandsEnergy()-9,16.1	float64	6735	0	0.000000
fBodyGyro-bandsEnergy()-1,8.1	float64	7257	0	0.000000
fBodyGyro-bandsEnergy()-25,48	float64	6961	0	0.000000
fBodyGyro-bandsEnergy()-1,24	float64	7060	0	0.000000
fBodyGyro-bandsEnergy()-49,64	float64	6800	0	0.000000
fBodyGyro-bandsEnergy()-33,48	float64	7094	0	0.000000
fBodyGyro-bandsEnergy()-17,32	float64	7007	0	0.000000
fBodyGyro-bandsEnergy()-1,16	float64	7073	0	0.000000
fBodyGyro-bandsEnergy()-57,64	float64	6184	0	0.000000
fBodyGyro-bandsEnergy()-49,56	float64	6941	0	0.000000
fBodyGyro-bandsEnergy()-41,48	float64	7048	0	0.000000
fBodyGyro-bandsEnergy()-33,40	float64	7036	0	0.000000
fBodyGyro-bandsEnergy()-25,32	float64	6900	0	0.000000
fBodyGyro-bandsEnergy()-17,24	float64	6969	0	0.000000
fBodyGyro-bandsEnergy()-9,16	float64	7020	0	0.000000
fBodyGyro-bandsEnergy()-1,8	float64	7021	0	0.000000
fBodyAcc-energy()-Z	float64	7306	0	0.000000
fBodyAcc-energy()-X	float64	7034	0	0.000000
tBodyAcc-mean()-Y	float64	7352	0	0.000000
tBodyAccJerk-energy()-X	float64	7109	0	0.000000
tBodyAccJerk-max()-Y	float64	5249	0	0.000000
tBodyAccJerk-max()-Z	float64	5210	0	0.000000
tBodyAccJerk-min()-X	float64	5282	0	0.000000
tBodyAccJerk-min()-Y	float64	5236	0	0.000000

	Data Type	Unique Values	Null Values	% null Values
tBodyAccJerk-min()-Z	float64	5221	0	0.000000
tBodyAccJerk-sma()	float64	7351	0	0.000000
tBodyAccJerk-energy()-Y	float64	7229	0	0.000000
tBodyAccJerk-mad()-Z	float64	7349	0	0.000000
tBodyAccJerk-energy()-Z	float64	7196	0	0.000000
tBodyAccJerk-iqr()-X	float64	7347	0	0.000000
tBodyAccJerk-iqr()-Y	float64	7350	0	0.000000
tBodyAccJerk-iqr()-Z	float64	7344	0	0.000000
tBodyAccJerk-entropy()-X	float64	4130	0	0.000000
tBodyAccJerk-entropy()-Y	float64	4485	0	0.000000
tBodyAccJerk-max()-X	float64	5272	0	0.000000
tBodyAccJerk-mad()-Y	float64	7351	0	0.000000
tBodyGyro-iqr()-Y	float64	7351	0	0.000000
tGravityAcc-correlation()-X,Z	float64	7352	0	0.000000
tGravityAcc-arCoeff()-Y,4	float64	7352	0	0.000000
tGravityAcc-arCoeff()-Z,1	float64	7352	0	0.000000
tGravityAcc-arCoeff()-Z,2	float64	7352	0	0.000000
tGravityAcc-arCoeff()-Z,3	float64	7351	0	0.000000
tGravityAcc-arCoeff()-Z,4	float64	7352	0	0.000000
tGravityAcc-correlation()-X,Y	float64	7352	0	0.000000
tGravityAcc-correlation()-Y,Z	float64	7351	0	0.000000
tBodyAccJerk-mad()-X	float64	7348	0	0.000000
tBodyAccJerk-mean()-X	float64	7352	0	0.000000
tBodyAccJerk-mean()-Y	float64	7352	0	0.000000
tBodyAccJerk-mean()-Z	float64	7352	0	0.000000
tBodyAccJerk-std()-X	float64	7347	0	0.000000
tBodyAccJerk-std()-Y	float64	7351	0	0.000000
tBodyAccJerk-std()-Z	float64	7350	0	0.000000
tBodyAccJerk-entropy()-Z	float64	4973	0	0.000000
tBodyAccJerk-arCoeff()-X,1	float64	7352	0	0.000000
tBodyAccJerk-arCoeff()-X,2	float64	7352	0	0.000000
tBodyGyro-max()-Z	float64	5414	0	0.000000

	Data Type	Unique Values	Null Values	% null Values
tBodyGyro-std()-Z	float64	7351	0	0.000000
tBodyGyro-mad()-X	float64	7349	0	0.000000
tBodyGyro-mad()-Y	float64	7351	0	0.000000
tBodyGyro-mad()-Z	float64	7348	0	0.000000
tBodyGyro-max()-X	float64	5439	0	0.000000
tBodyGyro-max()-Y	float64	5303	0	0.000000
tBodyGyro-min()-X	float64	5399	0	0.000000
tBodyAccJerk-arCoeff()-X,3	float64	7352	0	0.000000
tBodyGyro-min()-Y	float64	5325	0	0.000000
tBodyGyro-min()-Z	float64	5416	0	0.000000
tBodyGyro-sma()	float64	7349	0	0.000000
tBodyGyro-energy()-X	float64	7119	0	0.000000
tBodyGyro-energy()-Y	float64	7246	0	0.000000
tBodyGyro-energy()-Z	float64	7233	0	0.000000
tBodyGyro-std()-Y	float64	7352	0	0.000000
tBodyGyro-std()-X	float64	7346	0	0.000000
tBodyGyro-mean()-Z	float64	7351	0	0.000000
tBodyGyro-mean()-Y	float64	7352	0	0.000000
tBodyGyro-mean()-X	float64	7352	0	0.000000
tBodyAccJerk-correlation()-Y,Z	float64	7350	0	0.000000
tBodyAccJerk-correlation()-X,Z	float64	7352	0	0.000000
tBodyAccJerk-correlation()-X,Y	float64	7352	0	0.000000
tBodyAccJerk-arCoeff()-Z,4	float64	7352	0	0.000000
tBodyAccJerk-arCoeff()-Z,3	float64	7352	0	0.000000
tBodyAccJerk-arCoeff()-Z,2	float64	7352	0	0.000000
tBodyAccJerk-arCoeff()-Z,1	float64	7352	0	0.000000
tBodyAccJerk-arCoeff()-Y,4	float64	7352	0	0.000000
tBodyAccJerk-arCoeff()-Y,3	float64	7351	0	0.000000
tBodyAccJerk-arCoeff()-Y,2	float64	7352	0	0.000000
tBodyAccJerk-arCoeff()-Y,1	float64	7352	0	0.000000
tBodyAccJerk-arCoeff()-X,4	float64	7352	0	0.000000
tGravityAcc-arCoeff()-Y,3	float64	7351	0	0.000000

	Data Type	Unique Values	Null Values	% null Values
tGravityAcc-arCoeff()-Y,2	float64	7352	0	0.000000
tGravityAcc-arCoeff()-Y,1	float64	7352	0	0.000000
tBodyAcc-arCoeff()-X,1	float64	7352	0	0.000000
tBodyAcc-iqr()-X	float64	7349	0	0.000000
tBodyAcc-iqr()-Y	float64	7348	0	0.000000
tBodyAcc-iqr()-Z	float64	7347	0	0.000000
tBodyAcc-entropy()-X	float64	3860	0	0.000000
tBodyAcc-entropy()-Y	float64	5848	0	0.000000
tBodyAcc-entropy()-Z	float64	6462	0	0.000000
tBodyAcc-arCoeff()-X,2	float64	7352	0	0.000000
tBodyAcc-arCoeff()-Z,2	float64	7352	0	0.000000
tBodyAcc-arCoeff()-X,3	float64	7351	0	0.000000
tBodyAcc-arCoeff()-X,4	float64	7352	0	0.000000
tBodyAcc-arCoeff()-Y,1	float64	7352	0	0.000000
tBodyAcc-arCoeff()-Y,2	float64	7352	0	0.000000
tBodyAcc-arCoeff()-Y,3	float64	7352	0	0.000000
tBodyAcc-arCoeff()-Y,4	float64	7352	0	0.000000
tBodyAcc-energy()-Z	float64	7317	0	0.000000
tBodyAcc-energy()-Y	float64	7239	0	0.000000
tBodyAcc-energy()-X	float64	7054	0	0.000000
tBodyAcc-sma()	float64	7351	0	0.000000
tBodyAcc-min()-Z	float64	5160	0	0.000000
tBodyAcc-min()-Y	float64	5243	0	0.000000
tBodyAcc-min()-X	float64	5207	0	0.000000
tBodyAcc-max()-Z	float64	5216	0	0.000000
tBodyAcc-max()-Y	float64	5204	0	0.000000
tBodyAcc-max()-X	float64	5219	0	0.000000
tBodyAcc-mad()-Z	float64	7351	0	0.000000
tBodyAcc-mad()-Y	float64	7352	0	0.000000
tBodyAcc-mad()-X	float64	7347	0	0.000000
tBodyAcc-std()-Z	float64	7350	0	0.000000
tBodyAcc-std()-Y	float64	7351	0	0.000000

	Data Type	Unique Values	Null Values	% null Values
tBodyAcc-std()-X	float64	7349	0	0.000000
tBodyAcc-mean()-Z	float64	7349	0	0.000000
tBodyAcc-arCoeff()-Z,1	float64	7352	0	0.000000
tBodyAcc-arCoeff()-Z,3	float64	7351	0	0.000000
tGravityAcc-arCoeff()-X,4	float64	7352	0	0.000000
tGravityAcc-iqr()-Y	float64	7348	0	0.000000
tGravityAcc-min()-Z	float64	5726	0	0.000000
tGravityAcc-sma()	float64	7352	0	0.000000
tGravityAcc-energy()-X	float64	7350	0	0.000000
tGravityAcc-energy()-Y	float64	7348	0	0.000000
tGravityAcc-energy()-Z	float64	7349	0	0.000000
tGravityAcc-iqr()-X	float64	7338	0	0.000000
tGravityAcc-iqr()-Z	float64	7350	0	0.000000
tBodyAcc-arCoeff()-Z,4	float64	7352	0	0.000000
tGravityAcc-entropy()-X	float64	3168	0	0.000000
tGravityAcc-entropy()-Y	float64	1179	0	0.000000
tGravityAcc-entropy()-Z	float64	2710	0	0.000000
tGravityAcc-arCoeff()-X,1	float64	7352	0	0.000000
tGravityAcc-arCoeff()-X,2	float64	7351	0	0.000000
tGravityAcc-arCoeff()-X,3	float64	7350	0	0.000000
tGravityAcc-min()-Y	float64	5681	0	0.000000
tGravityAcc-min()-X	float64	5617	0	0.000000
tGravityAcc-max()-Z	float64	5689	0	0.000000
tGravityAcc-max()-Y	float64	5768	0	0.000000
tGravityAcc-max()-X	float64	5703	0	0.000000
tGravityAcc-mad()-Z	float64	7349	0	0.000000
tGravityAcc-mad()-Y	float64	7347	0	0.000000
tGravityAcc-mad()-X	float64	7347	0	0.000000
tGravityAcc-std()-Z	float64	7350	0	0.000000
tGravityAcc-std()-Y	float64	7349	0	0.000000
tGravityAcc-std()-X	float64	7346	0	0.000000
tGravityAcc-mean()-Z	float64	7352	0	0.000000

	Data Type	Unique Values	Null Values	% null Values
tGravityAcc-mean()-Y	float64	7352	0	0.000000
tGravityAcc-mean()-X	float64	7351	0	0.000000
tBodyAcc-correlation()-Y,Z	float64	7352	0	0.000000
tBodyAcc-correlation()-X,Z	float64	7352	0	0.000000
tBodyAcc-correlation()-X,Y	float64	7352	0	0.000000
tBodyGyro-iqr()-X	float64	7351	0	0.000000
tBodyGyro-iqr()-Z	float64	7347	0	0.000000
fBodyAcc-sma()	float64	7348	0	0.000000
tBodyAccJerkMag-arCoeff()3	float64	7352	0	0.000000
tBodyAccJerkMag-sma()	float64	7350	0	0.000000
tBodyAccJerkMag-energy()	float64	7195	0	0.000000
tBodyAccJerkMag-iqr()	float64	7347	0	0.000000
tBodyAccJerkMag-entropy()	float64	5605	0	0.000000
tBodyAccJerkMag-arCoeff()1	float64	7350	0	0.000000
tBodyAccJerkMag-arCoeff()2	float64	7352	0	0.000000
tBodyAccJerkMag-arCoeff()4	float64	7352	0	0.000000
tBodyAccJerkMag-max()	float64	5284	0	0.000000
tBodyGyroMag-mean()	float64	7352	0	0.000000
tBodyGyroMag-std()	float64	7352	0	0.000000
tBodyGyroMag-mad()	float64	7349	0	0.000000
tBodyGyroMag-max()	float64	5524	0	0.000000
tBodyGyroMag-min()	float64	5296	0	0.000000
tBodyGyroMag-sma()	float64	7352	0	0.000000
tBodyAccJerkMag-min()	float64	5143	0	0.000000
tBodyAccJerkMag-mad()	float64	7347	0	0.000000
tBodyGyro-entropy()-X	float64	5961	0	0.000000
tGravityAccMag-energy()	float64	7286	0	0.000000
tGravityAccMag-mean()	float64	7350	0	0.000000
tGravityAccMag-std()	float64	7350	0	0.000000
tGravityAccMag-mad()	float64	7350	0	0.000000
tGravityAccMag-max()	float64	5421	0	0.000000
tGravityAccMag-min()	float64	5171	0	0.000000

	Data Type	Unique Values	Null Values	% null Values
tGravityAccMag-sma()	float64	7350	0	0.000000
tGravityAccMag-iqr()	float64	7350	0	0.000000
tBodyAccJerkMag-std()	float64	7349	0	0.000000
tGravityAccMag-entropy()	float64	5329	0	0.000000
tGravityAccMag-arCoeff()1	float64	7352	0	0.000000
tGravityAccMag-arCoeff()2	float64	7352	0	0.000000
tGravityAccMag-arCoeff()3	float64	7352	0	0.000000
tGravityAccMag-arCoeff()4	float64	7352	0	0.000000
tBodyAccJerkMag-mean()	float64	7350	0	0.000000
tBodyGyroMag-energy()	float64	7288	0	0.000000
tBodyGyroMag-iqr()	float64	7348	0	0.000000
tBodyGyroMag-entropy()	float64	6243	0	0.000000
fBodyAcc-mad()-Y	float64	7352	0	0.000000
fBodyAcc-mean()-Y	float64	7350	0	0.000000
fBodyAcc-mean()-Z	float64	7351	0	0.000000
fBodyAcc-std()-X	float64	7347	0	0.000000
fBodyAcc-std()-Y	float64	7352	0	0.000000
fBodyAcc-std()-Z	float64	7351	0	0.000000
fBodyAcc-mad()-X	float64	7350	0	0.000000
fBodyAcc-mad()-Z	float64	7352	0	0.000000
tBodyGyroMag-arCoeff()1	float64	7352	0	0.000000
fBodyAcc-max()-X	float64	7349	0	0.000000
fBodyAcc-max()-Y	float64	7352	0	0.000000
fBodyAcc-max()-Z	float64	7352	0	0.000000
fBodyAcc-min()-X	float64	7347	0	0.000000
fBodyAcc-min()-Y	float64	7348	0	0.000000
fBodyAcc-min()-Z	float64	7349	0	0.000000
fBodyAcc-mean()-X	float64	7350	0	0.000000
tBodyGyroJerkMag-arCoeff()4	float64	7352	0	0.000000
tBodyGyroJerkMag-arCoeff()3	float64	7352	0	0.000000
tBodyGyroJerkMag-arCoeff()2	float64	7351	0	0.000000
tBodyGyroJerkMag-arCoeff()1	float64	7352	0	0.000000

	Data Type	Unique Values	Null Values	% null Values
tBodyGyroJerkMag-entropy()	float64	5465	0	0.000000
tBodyGyroJerkMag-iqr()	float64	7348	0	0.000000
tBodyGyroJerkMag-energy()	float64	6966	0	0.000000
tBodyGyroJerkMag-sma()	float64	7350	0	0.000000
tBodyGyroJerkMag-min()	float64	5163	0	0.000000
tBodyGyroJerkMag-max()	float64	5367	0	0.000000
tBodyGyroJerkMag-mad()	float64	7349	0	0.000000
tBodyGyroJerkMag-std()	float64	7352	0	0.000000
tBodyGyroJerkMag-mean()	float64	7350	0	0.000000
tBodyGyroMag-arCoeff()4	float64	7351	0	0.000000
tBodyGyroMag-arCoeff()3	float64	7351	0	0.000000
tBodyGyroMag-arCoeff()2	float64	7351	0	0.000000
tBodyAccMag-arCoeff()4	float64	7352	0	0.000000
tBodyAccMag-arCoeff()3	float64	7352	0	0.000000
tBodyAccMag-arCoeff()2	float64	7352	0	0.000000
tBodyGyroJerk-mad()-X	float64	7348	0	0.000000
tBodyGyroJerk-mean()-X	float64	7352	0	0.000000
tBodyGyroJerk-mean()-Y	float64	7352	0	0.000000
tBodyGyroJerk-mean()-Z	float64	7352	0	0.000000
tBodyGyroJerk-std()-X	float64	7347	0	0.000000
tBodyGyroJerk-std()-Y	float64	7349	0	0.000000
tBodyGyroJerk-std()-Z	float64	7347	0	0.000000
tBodyGyroJerk-mad()-Y	float64	7349	0	0.000000
tBodyGyroJerk-sma()	float64	7347	0	0.000000
tBodyGyroJerk-mad()-Z	float64	7344	0	0.000000
tBodyGyroJerk-max()-X	float64	5238	0	0.000000
tBodyGyroJerk-max()-Y	float64	5273	0	0.000000
tBodyGyroJerk-max()-Z	float64	5309	0	0.000000
tBodyGyroJerk-min()-X	float64	5272	0	0.000000
tBodyGyroJerk-min()-Y	float64	5300	0	0.000000
tBodyGyro-correlation()-Y,Z	float64	7352	0	0.000000
tBodyGyro-correlation()-X,Z	float64	7352	0	0.000000

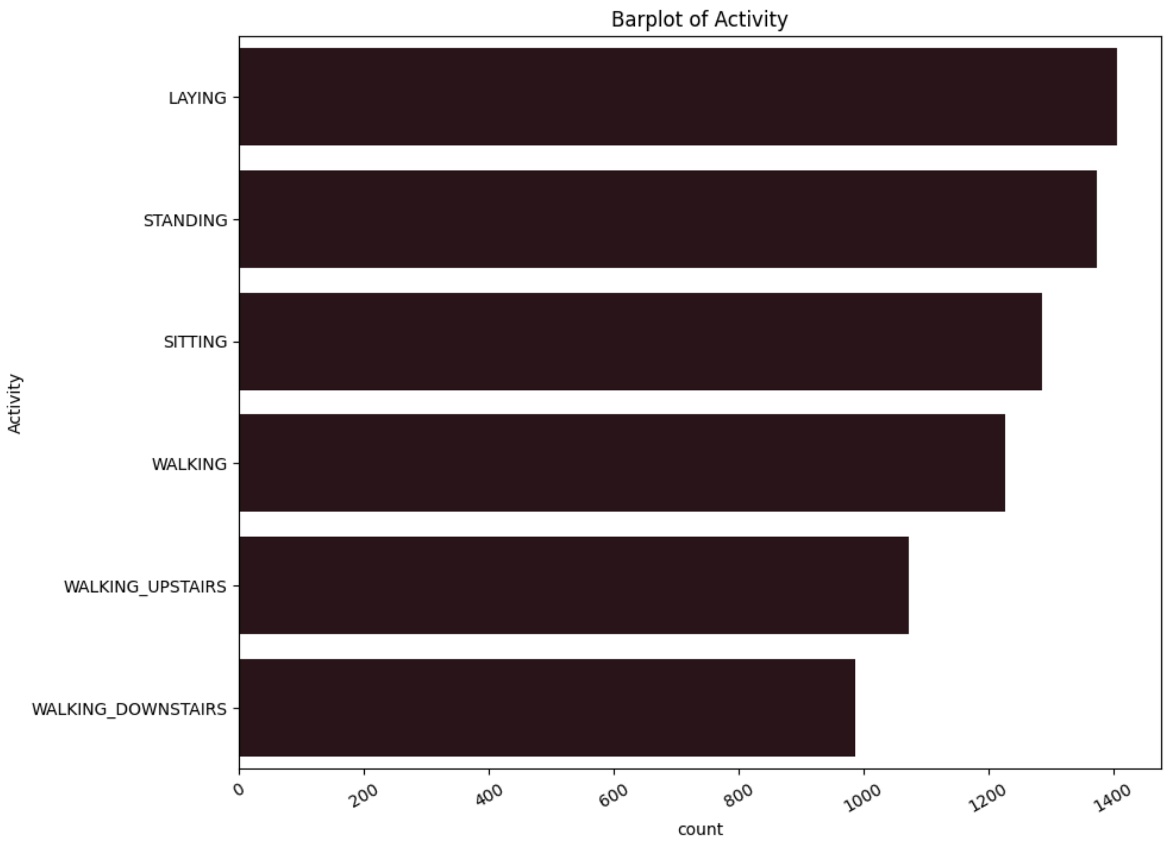
	Data Type	Unique Values	Null Values	% null Values
tBodyGyro-correlation()-X,Y	float64	7351	0	0.000000
tBodyGyro-arCoeff()-Z,4	float64	7352	0	0.000000
tBodyGyro-arCoeff()-Z,3	float64	7352	0	0.000000
tBodyGyro-arCoeff()-Z,2	float64	7352	0	0.000000
tBodyGyro-arCoeff()-Z,1	float64	7352	0	0.000000
tBodyGyro-arCoeff()-Y,4	float64	7351	0	0.000000
tBodyGyro-arCoeff()-Y,3	float64	7351	0	0.000000
tBodyGyro-arCoeff()-Y,2	float64	7352	0	0.000000
tBodyGyro-arCoeff()-Y,1	float64	7351	0	0.000000
tBodyGyro-arCoeff()-X,4	float64	7350	0	0.000000
tBodyGyro-arCoeff()-X,3	float64	7351	0	0.000000
tBodyGyro-arCoeff()-X,2	float64	7352	0	0.000000
tBodyGyro-arCoeff()-X,1	float64	7352	0	0.000000
tBodyGyro-entropy()-Z	float64	5452	0	0.000000
tBodyGyro-entropy()-Y	float64	6007	0	0.000000
tBodyGyroJerk-min()-Z	float64	5276	0	0.000000
tBodyGyroJerk-energy()-X	float64	7049	0	0.000000
tBodyAccMag-arCoeff()1	float64	7352	0	0.000000
tBodyAccMag-std()	float64	7350	0	0.000000
tBodyGyroJerk-arCoeff()-Z,3	float64	7352	0	0.000000
tBodyGyroJerk-arCoeff()-Z,4	float64	7352	0	0.000000
tBodyGyroJerk-correlation()-X,Y	float64	7351	0	0.000000
tBodyGyroJerk-correlation()-X,Z	float64	7352	0	0.000000
tBodyGyroJerk-correlation()-Y,Z	float64	7352	0	0.000000
tBodyAccMag-mean()	float64	7350	0	0.000000
tBodyAccMag-mad()	float64	7350	0	0.000000
tBodyGyroJerk-energy()-Y	float64	6903	0	0.000000
tBodyAccMag-max()	float64	5421	0	0.000000
tBodyAccMag-min()	float64	5171	0	0.000000
tBodyAccMag-sma()	float64	7350	0	0.000000
tBodyAccMag-energy()	float64	7286	0	0.000000
tBodyAccMag-iqr()	float64	7350	0	0.000000

	Data Type	Unique Values	Null Values	% null Values
tBodyAccMag-entropy()	float64	5329	0	0.000000
tBodyGyroJerk-arCoeff()-Z,2	float64	7352	0	0.000000
tBodyGyroJerk-arCoeff()-Z,1	float64	7352	0	0.000000
tBodyGyroJerk-arCoeff()-Y,4	float64	7352	0	0.000000
tBodyGyroJerk-arCoeff()-Y,3	float64	7352	0	0.000000
tBodyGyroJerk-arCoeff()-Y,2	float64	7352	0	0.000000
tBodyGyroJerk-arCoeff()-Y,1	float64	7351	0	0.000000
tBodyGyroJerk-arCoeff()-X,4	float64	7352	0	0.000000
tBodyGyroJerk-arCoeff()-X,3	float64	7352	0	0.000000
tBodyGyroJerk-arCoeff()-X,2	float64	7352	0	0.000000
tBodyGyroJerk-arCoeff()-X,1	float64	7351	0	0.000000
tBodyGyroJerk-entropy()-Z	float64	4599	0	0.000000
tBodyGyroJerk-entropy()-Y	float64	5181	0	0.000000
tBodyGyroJerk-entropy()-X	float64	4703	0	0.000000
tBodyGyroJerk-iqr()-Z	float64	7338	0	0.000000
tBodyGyroJerk-iqr()-Y	float64	7344	0	0.000000
tBodyGyroJerk-iqr()-X	float64	7350	0	0.000000
tBodyGyroJerk-energy()-Z	float64	6992	0	0.000000
Activity	object	6	0	0.000000

iii. Checking for ImBalance

In [32]:

```
plt.figure(figsize=(10,8))
plt.title('Barplot of Activity')
sns.countplot(train.Activity, order = train.Activity.value_counts().index, color
plt.xticks(rotation = 30)
plt.show()
```



- From the above imbalnce graph, there is almost same number of observations across all the six activities so this data does not have class imbalance problem.

4. Exploratory Data Analysis

In [35]:
```
train.columns.value_counts().sum()
```

Out[35]:
```
563
```

`Subject` = Numbers from 1 to 30 reprsents the 30 volunteers

In [40]:
```
train['subject'].value_counts()
```

Out[40]:

```
subject
25    409
21    408
26    392
30    383
28    382
27    376
23    372
17    368
16    366
19    360
1     347
29    344
3     341
15    328
6     325
14    323
22    321
11    316
7     308
5     302
8     281
Name: count, dtype: int64
```

In [36]:

```
train.head()
```

Out[36]:

	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	tBodyAcc-mad()-X	tB
0	0.288585	-0.020294	-0.132905	-0.995279	-0.983111	-0.913526	-0.995112	-(
1	0.278419	-0.016411	-0.123520	-0.998245	-0.975300	-0.960322	-0.998807	-(
2	0.279653	-0.019467	-0.113462	-0.995380	-0.967187	-0.978944	-0.996520	-(
3	0.279174	-0.026201	-0.123283	-0.996091	-0.983403	-0.990675	-0.997099	-(
4	0.276629	-0.016570	-0.115362	-0.998139	-0.980817	-0.990482	-0.998321	-(

5 rows × 563 columns

In [44]:

```
"""
Here, tBodyAcc-mean()-X, etc. gives all the acceleration poition, so just split
"""
Counter([col.split('-')[0].split('(')[0] for col in train.columns])
Out[44]: Counter({'fBodyAcc': 79,
                  'fBodyAccJerk': 79,
                  'fBodyGyro': 79,
                  'tBodyAcc': 40,
                  'tGravityAcc': 40,
                  'tBodyAccJerk': 40,
                  'tBodyGyro': 40,
                  'tBodyGyroJerk': 40,
                  'tBodyAccMag': 13,
                  'tGravityAccMag': 13,
                  'tBodyAccJerkMag': 13,
                  'tBodyGyroMag': 13,
                  'tBodyGyroJerkMag': 13,
                  'fBodyAccMag': 13,
                  'fBodyBodyAccJerkMag': 13,
                  'fBodyBodyGyroMag': 13,
                  'fBodyBodyGyroJerkMag': 13,
                  'angle': 7,
                  'subject': 1,
                  'Activity': 1})
```

```
In [45]: # count: gives the parameters
         pd.DataFrame.from_dict(Counter([col.split('-')[0].split('(')[0] for col in trair
                                orient = "index").rename(columns = {0:'count'}).sort_valu
```

Out[45]:

	count
fBodyAcc	79
fBodyGyro	79
fBodyAccJerk	79
tGravityAcc	40
tBodyAcc	40
tBodyGyroJerk	40
tBodyGyro	40
tBodyAccJerk	40
tBodyAccMag	13
tGravityAccMag	13
tBodyAccJerkMag	13
tBodyGyroMag	13
tBodyGyroJerkMag	13
fBodyAccMag	13
fBodyBodyAccJerkMag	13
fBodyBodyGyroMag	13
fBodyBodyGyroJerkMag	13
angle	7
subject	1
Activity	1

Mainly there are 'acceleration' and 'gyroscope' features. A few 'gravity' features are there as well. Impressive how many features there are in regard of the limited number of sensors used.

Based on the common nature of activities we can broadly put them in two categories.

`Static and dynamic activities :`

- SITTING, STANDING, LAYING can be considered as static activities with no motion involved
- WALKING, WALKING_DOWNSTAIRS, WALKING_UPSTAIRS can be considered as dynamic activities with significant amount of motion involved

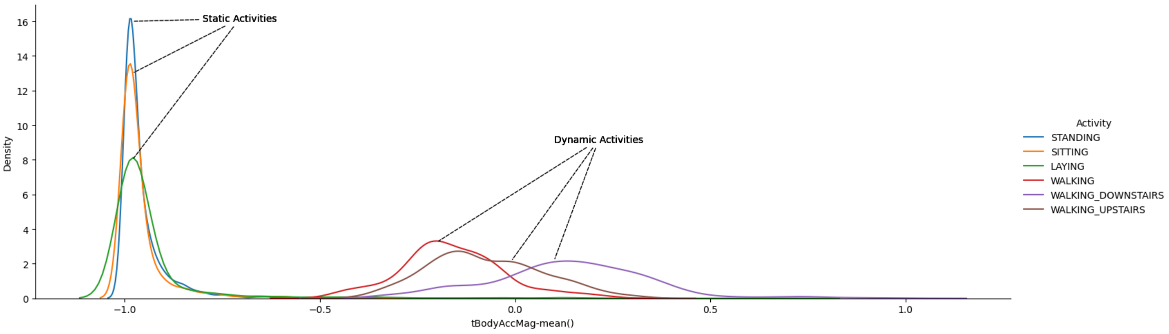
Let's consider tBodyAccMag-mean() feature to differentiate among these two broader set of activities.

If we try to build a simple classification model to classify the activity using one variable at a time then probability density function(PDF) is very helpful to assess importance of a continuous variable.s variable.s variable.variable.

i. Mean Feature- Analysis tBodyAccMag-mean

In [48]:

```
# Standing, Sitting, Laying are the Static activities and Walking, Walking_Downs
facetgrid = sns.FacetGrid(train, hue = 'Activity', height = 5, aspect = 3)
facetgrid.map(sns.distplot, 'tBodyAccMag-mean()', hist = False).add_legend()

#Annotions of Static Activities
plt.annotate("Static Activities", xy = (-.98, 8), xytext = (-.8, 16), arrowprops
plt.annotate("Static Activities", xy = (-.98, 13), xytext = (-.8, 16), arrowprop
plt.annotate("Static Activities", xy = (-.98, 16), xytext = (-.8, 16), arrowprop

#Annotions of Dynamic Actvities
plt.annotate("Dynamic Activities", xy=(-0.2,3.25), xytext=(0.1, 9),arrowprops={'
plt.annotate("Dynamic Activities", xy=(0.1,2.18), xytext=(0.1, 9),arrowprops={'a
plt.annotate("Dynamic Activities", xy=(-0.01,2.15), xytext=(0.1, 9),arrowprops={

plt.show()
```



Using the above density plot we can easily come with a condition to seperate static activities from dynamic activities.

if(tBodyAccMag-mean()<=-0.5):

```
    Activity = "static"
```

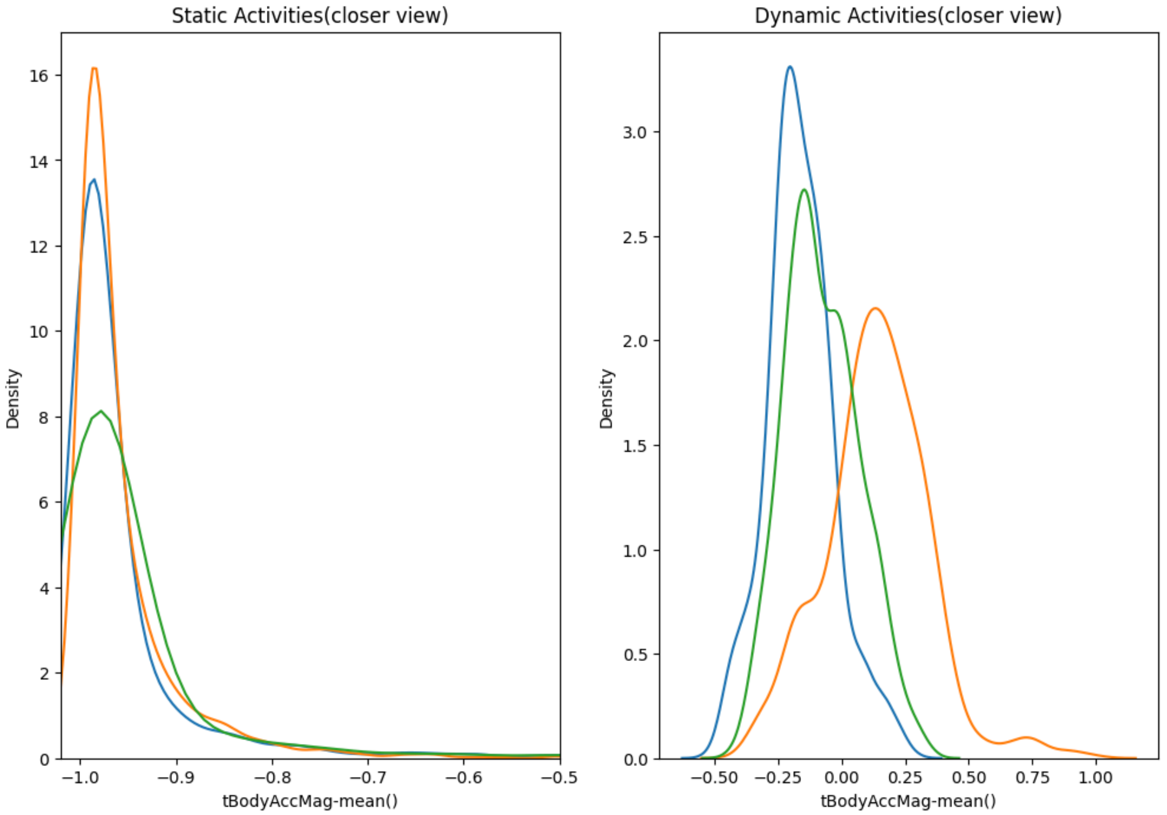
else:

```
    Activity = "dynamic"
```

Let's have a more closer view on the PDFs of each activity under static and dynamic categorization.

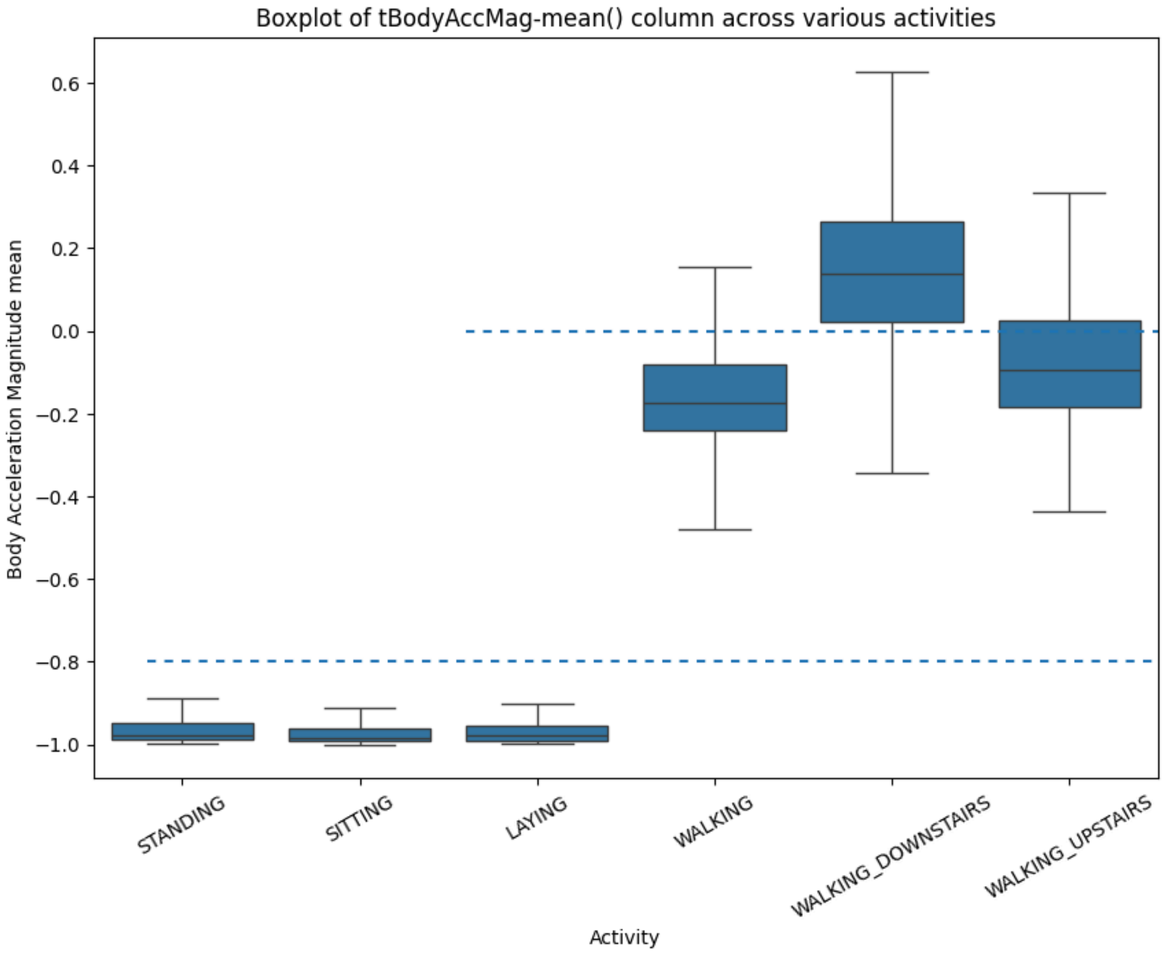
In [49]:

```
plt.figure(figsize=(12,8))
plt.subplot(1,2,1)
plt.title("Static Activities(closer view)")
sns.distplot(train[train["Activity"]=="SITTING"]['tBodyAccMag-mean()'],hist = Fa
sns.distplot(train[train["Activity"]=="STANDING"]['tBodyAccMag-mean()'],hist = F
sns.distplot(train[train["Activity"]=="LAYING"]['tBodyAccMag-mean()'],hist = Fal
plt.axis([-1.02, -0.5, 0, 17])
plt.subplot(1,2,2)
plt.title("Dynamic Activities(closer view)")
sns.distplot(train[train["Activity"]=="WALKING"]['tBodyAccMag-mean()'],hist = Fa
sns.distplot(train[train["Activity"]=="WALKING_DOWNSTAIRS"]['tBodyAccMag-mean()'
sns.distplot(train[train["Activity"]=="WALKING_UPSTAIRS"]['tBodyAccMag-mean()'],
plt.show()
```



The insights obtained through density plots can also be represented using Box plots. Let's plot the boxplot of Body Accelartion Magnitude mean(tBodyAccMag-mean()) across all the six categories.

In [53]:
```
plt.figure(figsize=(10,7))
sns.boxplot(x = "Activity", y="tBodyAccMag-mean()", data = train, showfliers = F
plt.ylabel('Body Acceleration Magnitude mean')
plt.title("Boxplot of tBodyAccMag-mean() column across various activities")
plt.axhline(y = -0.8, xmin = 0.05, dashes = (3,3))
plt.axhline(y= 0.0, xmin = 0.35, dashes=(3,3))
plt.xticks(rotation = 30)
plt.show()
```



Using boxplot again we can come with conditions to seperate static activities from dynamic activities.

if(tBodyAccMag-mean()<=-0.8):

```
    Activity = "static"
```

if(tBodyAccMag-mean()>=-0.6):

```
    Activity = "dynamic"
```

Also, we can easily seperate WALKING_DOWNSTAIRS activity from others using boxplot.

if(tBodyAccMag-mean()>0.02):

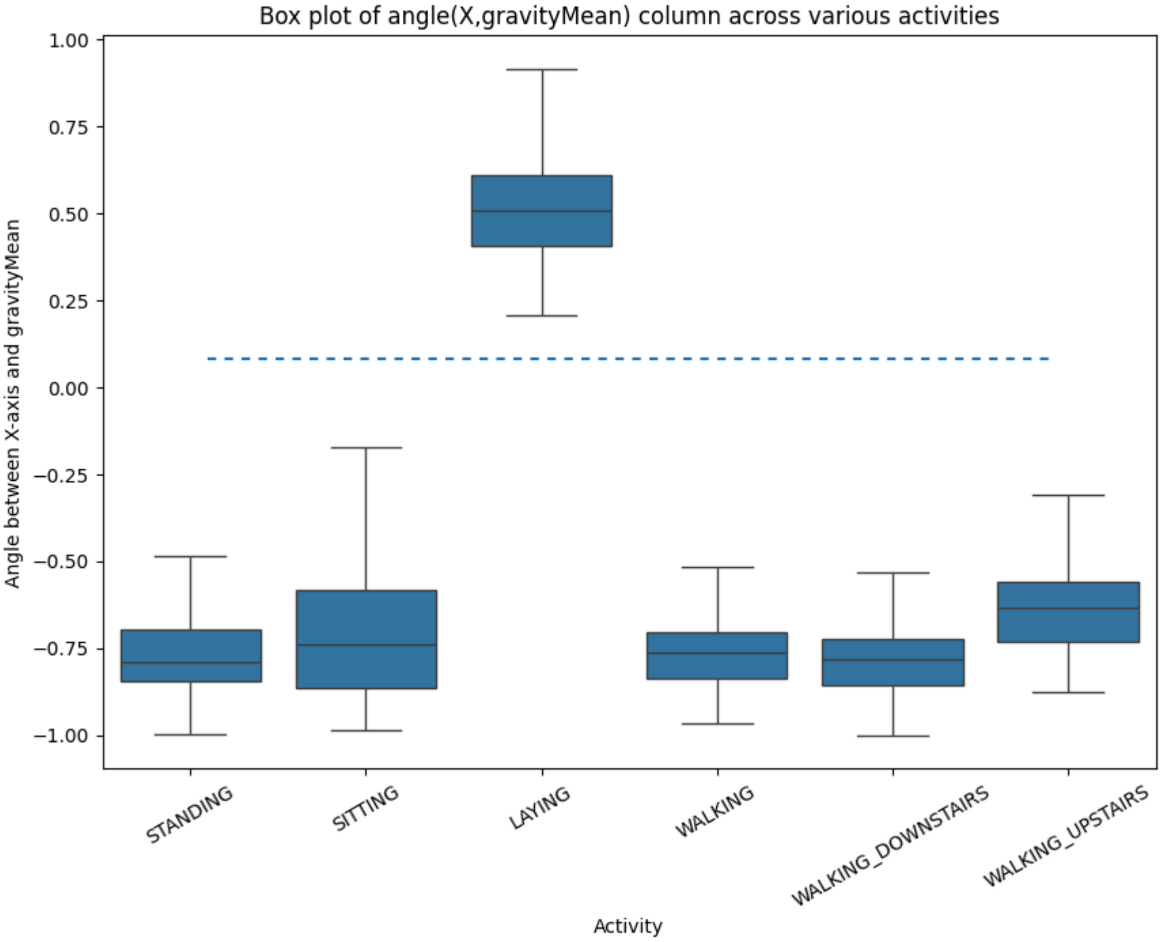
```
    Activity = "WALKING_DOWNSTAIRS"
```

else: Activity = "others"

But still 25% of WALKING_DOWNSTAIRS observations are below 0.02 which are misclassified as others so this condition makes an error of 25% in classification.

ii. Analysing Angle between X-axis and gravityMean feature

In [54]:
```
plt.figure(figsize=(10,7))
sns.boxplot(x='Activity', y='angle(X,gravityMean)', data=train, showfliers=False
plt.axhline(y=0.08, xmin=0.1, xmax=0.9,dashes=(3,3))
plt.ylabel("Angle between X-axis and gravityMean")
plt.title('Box plot of angle(X,gravityMean) column across various activities')
plt.xticks(rotation = 30)
plt.show()
```



From the boxplot we can observe that angle(X,gravityMean) perfectly seperates LAYING from other activities.

if(angle(X,gravityMean)>0.01):

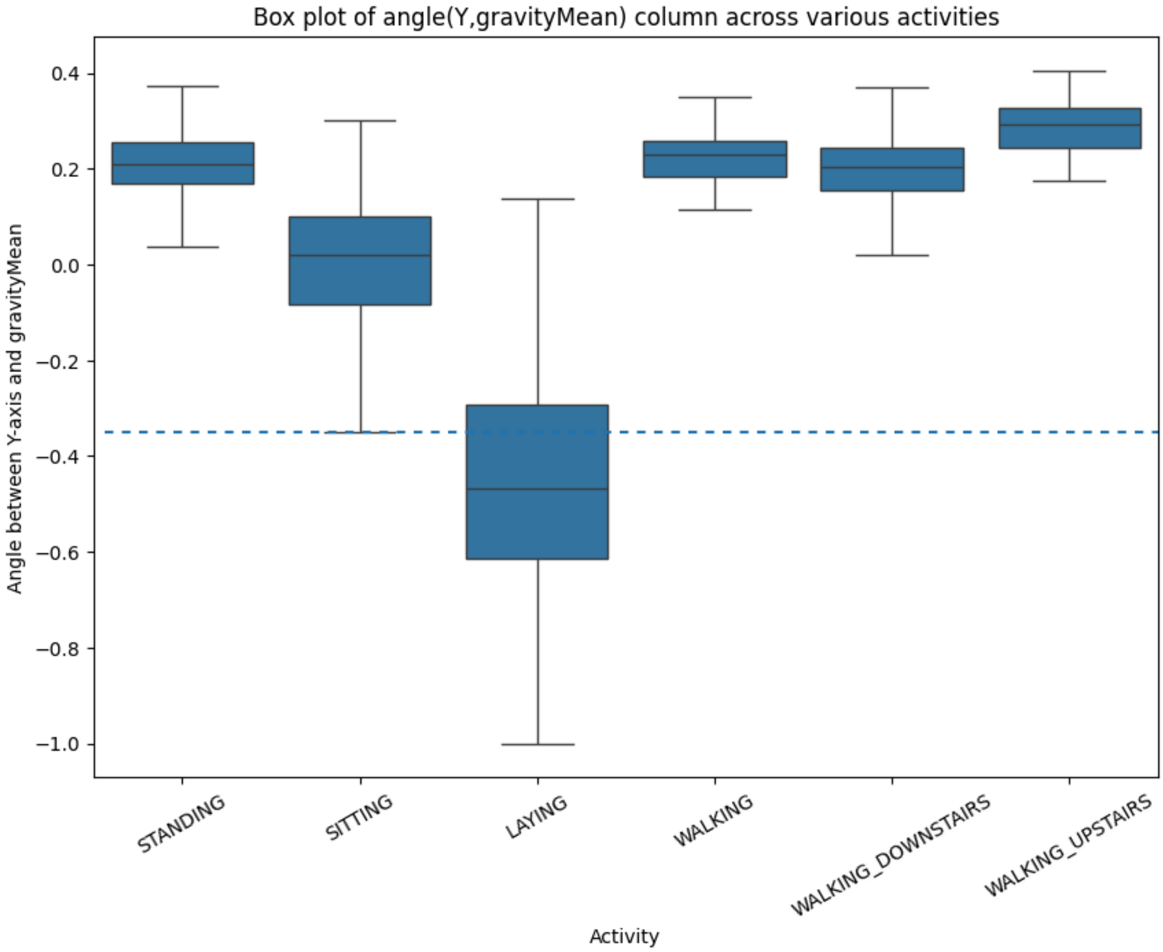
```
Activity = "LAYING"
```

else:

```
Activity = "others"
```

iii. Analysing Angle between Y-axis and gravityMean feature

In [56]:
```
plt.figure(figsize=(10,7))
sns.boxplot(x='Activity', y='angle(Y,gravityMean)', data = train, showfliers=Fal
plt.ylabel("Angle between Y-axis and gravityMean")
plt.title('Box plot of angle(Y,gravityMean) column across various activities')
plt.xticks(rotation = 30)
plt.axhline(y=-0.35, xmin=0.01, dashes=(3,3))
plt.show()
```



Similarly, using Angle between Y-axis and gravityMean we can seperate LAYING from other activities but again it leads to some misclassification error.

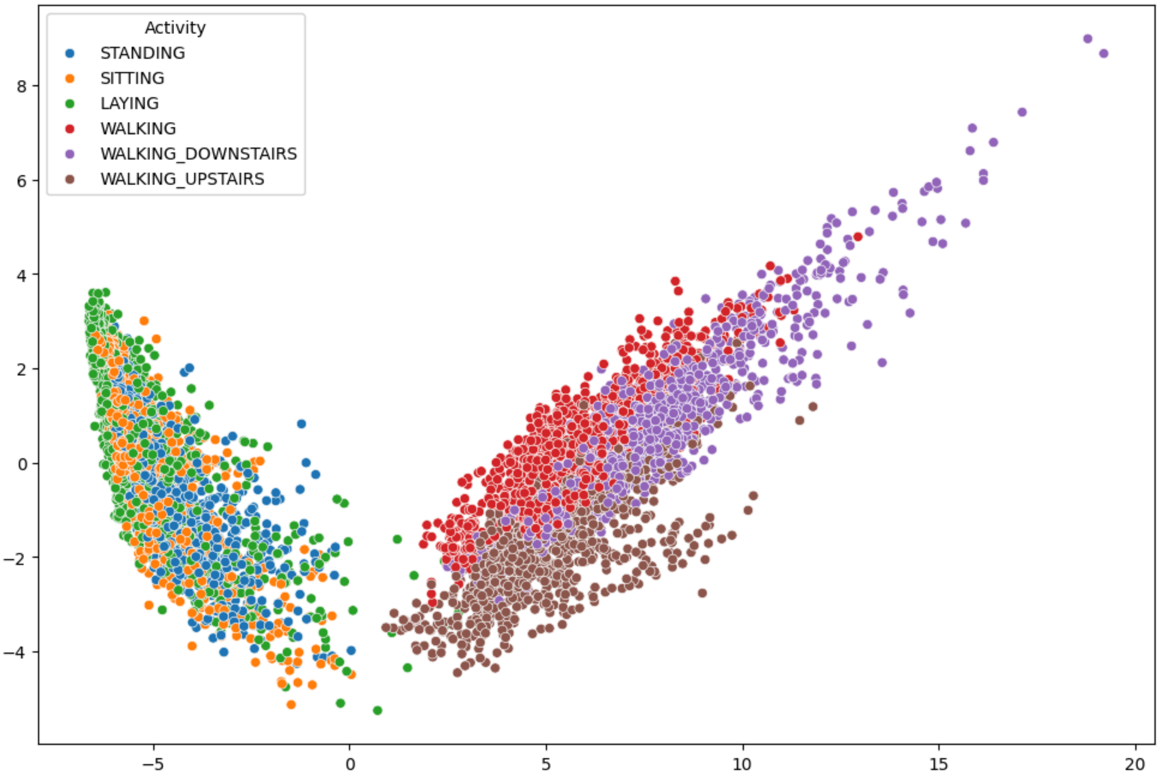
iv. Visualizing data using PCA (Principal Component Analysis)

Using PCA data can be visualized from a extremely high dimensional space to a low dimensional space and still it retains lots of actual information. Given training data has 561 unqiue features, using PCA let's visualize it to a 2D space.

In [58]:
```
x_for_pca = train.drop(['subject', 'Activity'], axis = 1)
pca = PCA(n_components=2, random_state=0).fit_transform(x_for_pca)
pca
```

Out[58]:
```
array([[-5.5202803 , -0.29027701],
       [-5.53534954, -0.08253011],
       [-5.47498801,  0.28738703],
       ...,
       [ 5.85750527, -3.08184312],
       [ 5.42109482, -3.42643002],
       [ 5.49797027, -2.78992867]])
```

In [59]:
```
plt.figure(figsize=(12,8))
sns.scatterplot(x = pca[:, 0], y = pca[:, 1], hue = train['Activity'])
plt.show()
```



- From the above graph, Using the two new components obtained through PCA we can visualize and seperate all the six activities in a 2D space.

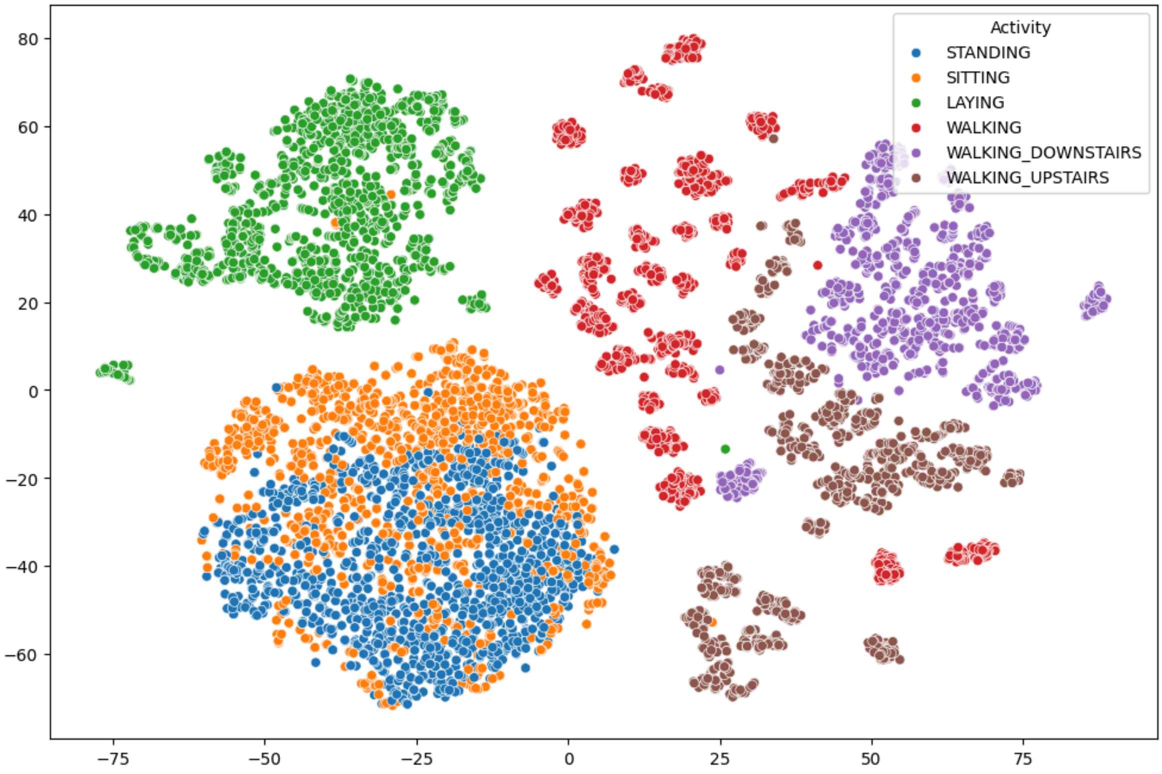
v. Visualizing data using t-SNE (TSNE: t-distributed Stochastic Neighbor Embedding)

Using t-SNE data can be visualized from a extremely high dimensional space to a low dimensional space and still it retains lots of actual information. Given training data has 561 unqiue features, using t-SNE let's visualize it to a 2D space.

In [60]:
```
x_for_tsne = train.drop(['subject', 'Activity'], axis = 1)
tsne = TSNE(n_components=2, random_state=0, n_iter=1000).fit_transform(x_for_tsn
tsne
```

Out[60]:
```
array([[ -6.9932804, -63.188377 ],
       [-25.07103  , -17.728373 ],
       [-27.199108 , -20.449244 ],
       ...,
       [ 23.136957 , -60.145756 ],
       [ 23.117773 , -59.890156 ],
       [ 23.605974 , -59.11817  ]], dtype=float32)
```

In [61]:
```
plt.figure(figsize=(12,8))
sns.scatterplot(x = tsne[:, 0], y = tsne[:, 1], hue = train['Activity'])
plt.show()
```



5. Build: Training and Testing Models

In [62]:

```
X_train = train.drop(['subject', 'Activity'], axis = 1)
y_train = train.Activity

X_test = test.drop(['subject', 'Activity'], axis = 1)
y_test = test.Activity
```

In [64]:

```
print(f"Shape Of Training Data Set : ",X_train.shape)
print(f"Shape Of Testing Data Set :",X_test.shape)
print(f"Shape Of Train Label :",y_train.shape)
print(f"Shape Of Test Label :",y_test.shape)
```

```
Shape Of Training Data Set :  (7352, 561)
Shape Of Testing Data Set : (999, 561)
Shape Of Train Label : (7352,)
Shape Of Test Label : (999,)
```

In [66]:

```
# lets define a function to plot a confusion matrix
def plot_confusion_matrix(cm,labels):
    fig, ax = plt.subplots(figsize=(12,8)) # for plotting confusion matrix as im
    im = ax.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
    ax.figure.colorbar(im, ax=ax)
    ax.set(xticks=np.arange(cm.shape[1]),
    yticks=np.arange(cm.shape[0]),
    xticklabels=labels, yticklabels=labels,
    ylabel='True label',
    xlabel='Predicted label')
    plt.xticks(rotation = 90)
    thresh = cm.max() / 2.
    for i in range(cm.shape[0]):
        for j in range(cm.shape[1]):
            ax.text(j, i, int(cm[i, j]),ha="center", va="center",color="white" i
    fig.tight_layout()
```

In [76]:

```
#function to get best random search attributes
def get_best_randomsearch_results(model):
    print("Best estimator : ", model.best_estimator_)
    print("Best set of parameters : ", model.best_params_)
    print("Best score : ", model.best_score_*100)
```

6. Fitting Machine Learning Algorithms to Model

i. Logistic Regression with Hyperparameter Tunning and Cross_Validation

In [77]:

```
parameters = {'max_iter': [100, 200, 500]}

# lr: Logistic Regression
lr_classifier = LogisticRegression()
lr_classifier_rs = RandomizedSearchCV(lr_classifier, param_distributions= parame
lr_classifier_rs.fit(X_train, y_train)
y_pred_lr = lr_classifier_rs.predict(X_test)

#Accuracy
lr_accuracy = accuracy_score(y_true=y_test, y_pred=y_pred_lr) * 100
print("Accuracy using Logistic Regression : ", lr_accuracy)
```

```
Accuracy using Logistic Regression :  95.4954954954955
```

In [78]:

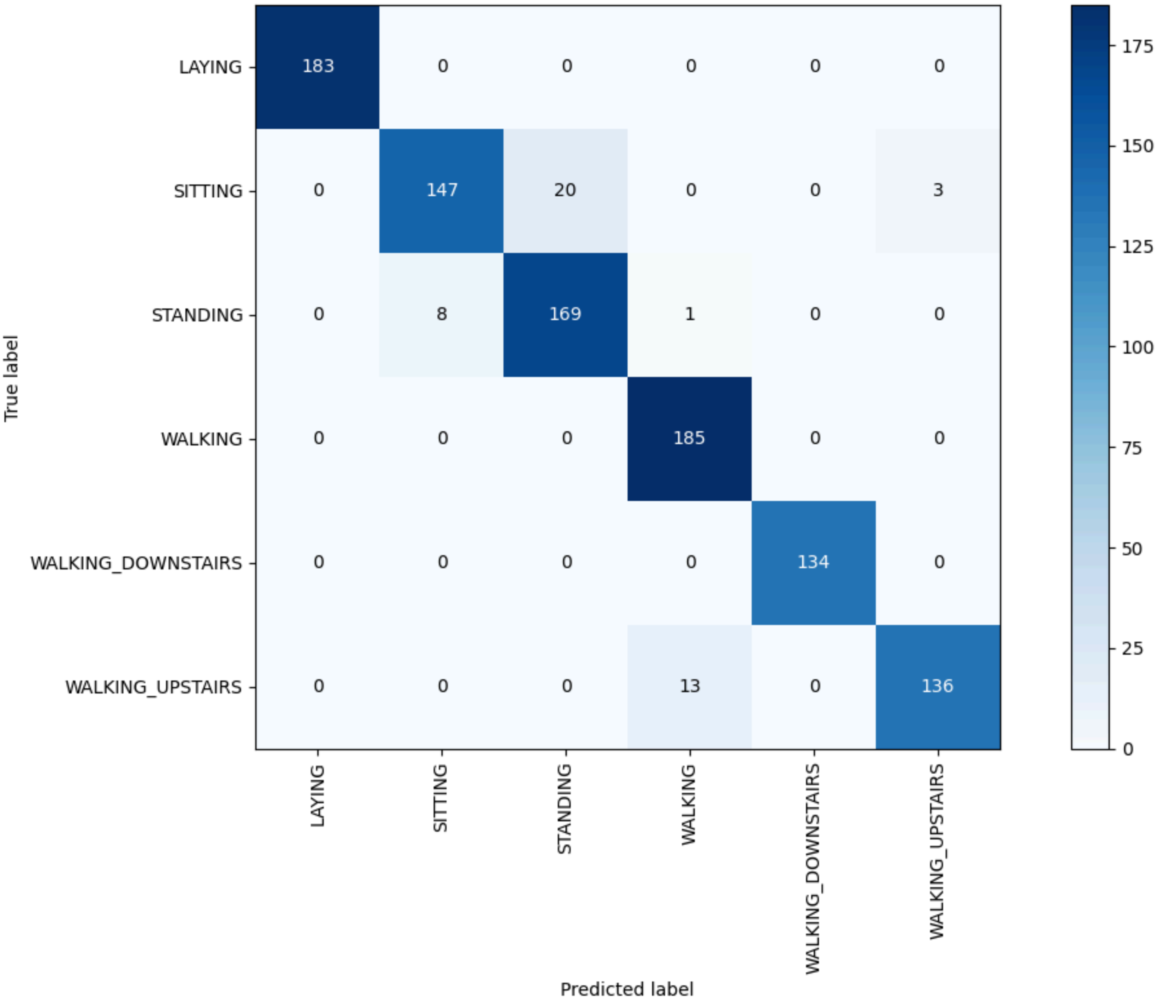
```
cm_lr = confusion_matrix(y_test.values,y_pred_lr)
cm_lr
```

Out[78]:

```
array([[183,   0,   0,   0,   0,   0],
       [  0, 147,  20,   0,   0,   3],
       [  0,   8, 169,   1,   0,   0],
       [  0,   0,   0, 185,   0,   0],
       [  0,   0,   0,   0, 134,   0],
       [  0,   0,   0,  13,   0, 136]], dtype=int64)
```

In [79]:

```
cm_lr = confusion_matrix(y_test.values,y_pred_lr)
plot_confusion_matrix(cm_lr, np.unique(y_pred_lr))
```



In [80]:
```
# getting best random search attributes
get_best_randomsearch_results(lr_classifier_rs)
```

```
Best estimator :  LogisticRegression(max_iter=500)
Best set of parameters :  {'max_iter': 500}
Best score :  93.73035141996976
```

ii. Kernel SVM model with Hyperparameter Tunning and Cross Validation

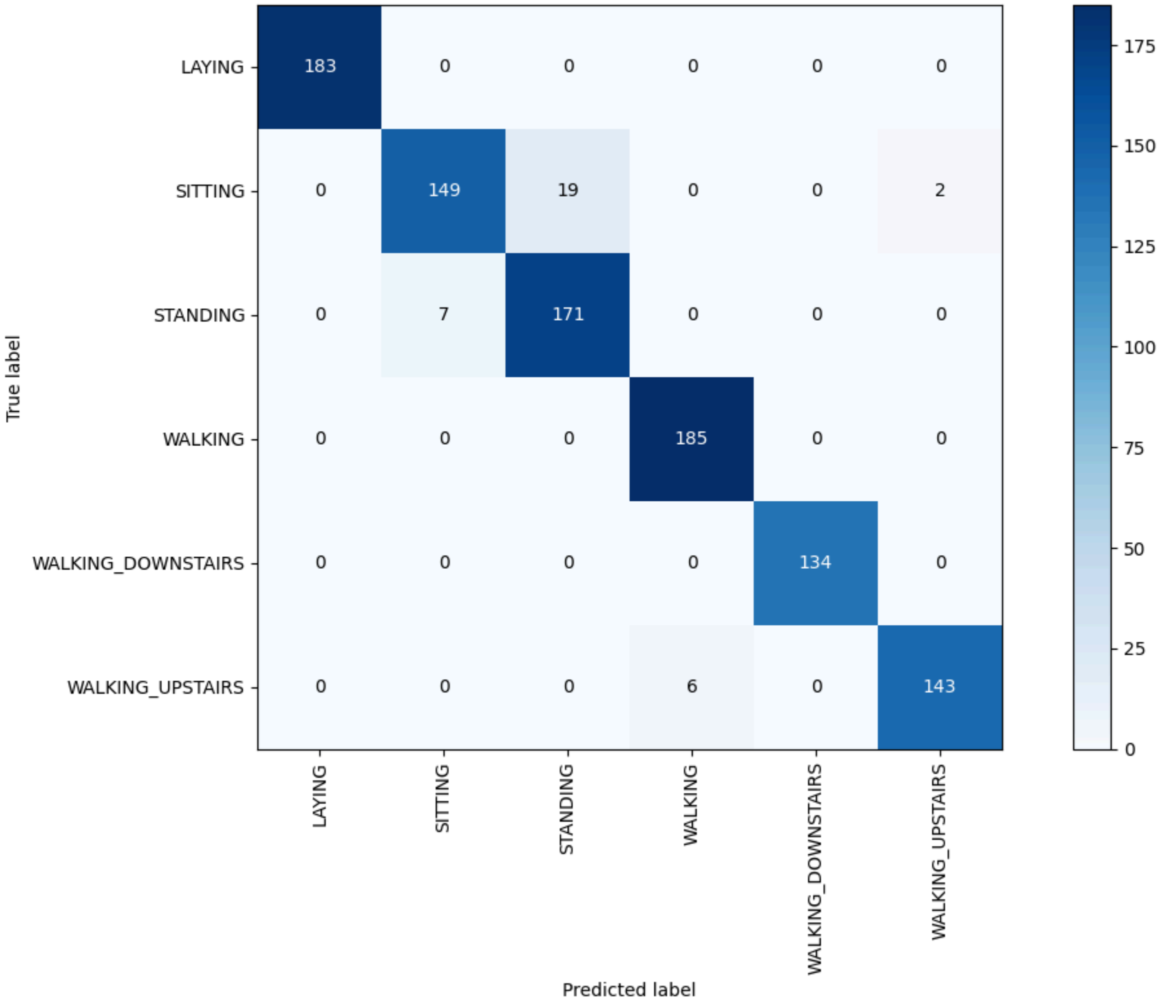
In [81]:
```
parameters = {
    'kernel': ['linear', 'rbf', 'poly', 'sigmoid'],
    'C': [100, 50]
}

svm_rs = RandomizedSearchCV(SVC(), param_distributions=parameters, cv = 3, randc
svm_rs.fit(X_train, y_train)
```

Out[81]:
```
▸ RandomizedSearchCV
  ▸ estimator: SVC
      ▸ SVC
```

In [82]:
```
y_pred = svm_rs.predict(X_test)
kernel_svm_accuracy = accuracy_score(y_true=y_test, y_pred=y_pred) * 100
print("Accuracy using Kernel SVM : ", kernel_svm_accuracy)
```

```
Accuracy using Kernel SVM :  96.5965965965966
```

In [83]:
```
cm_svm = confusion_matrix(y_test.values,y_pred)
plot_confusion_matrix(cm_svm, np.unique(y_pred))
```



In [84]:
```
get_best_randomsearch_results(svm_rs)
```

```
Best estimator :  SVC(C=50)
Best set of parameters :  {'kernel': 'rbf', 'C': 50}
Best score :  94.64109332023303
```

iii. Decision tree model with Hyperparameter tuning and cross validation

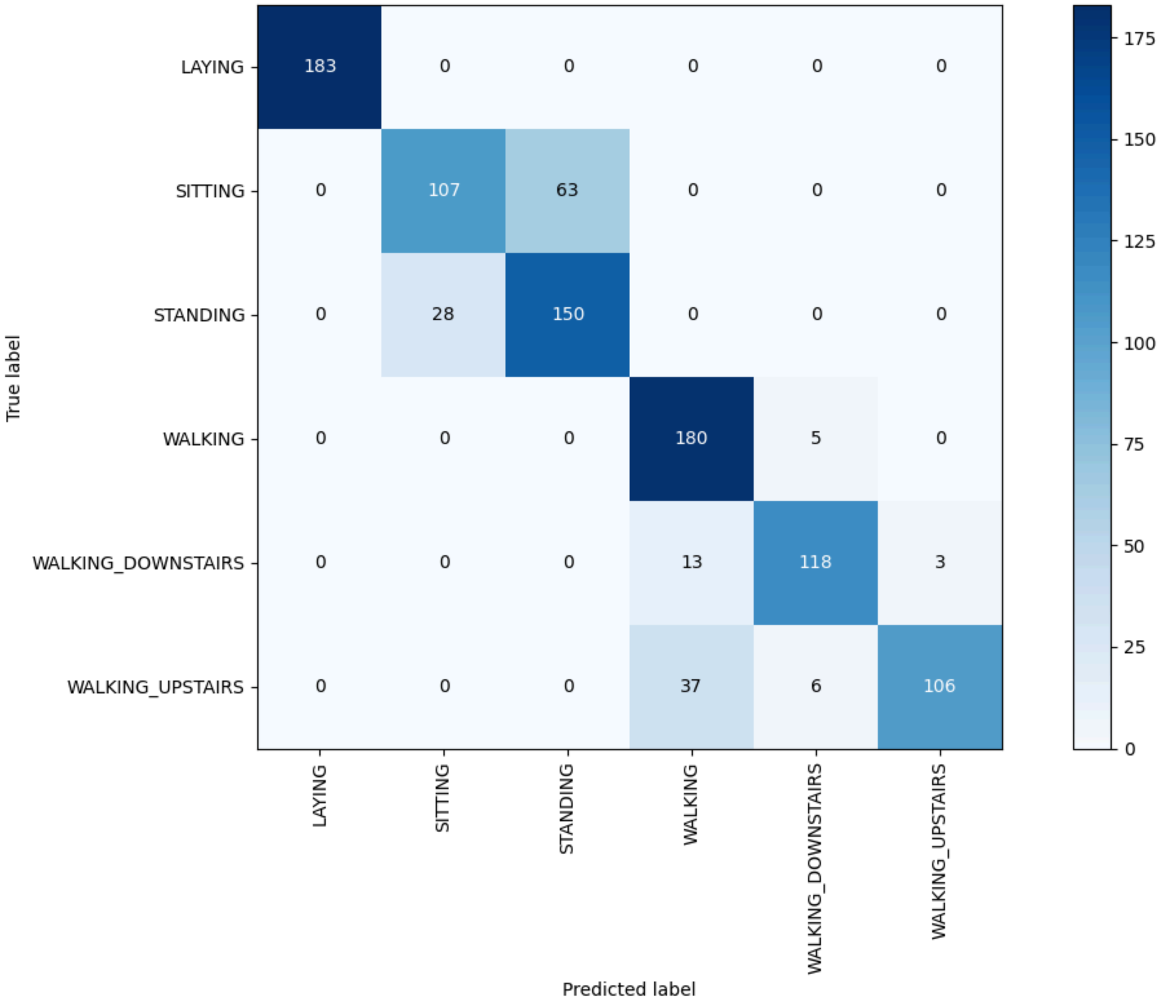
In [85]:
```
parameters = {'max_depth': np.arange(2, 10, 2)}

dt_classifier = DecisionTreeClassifier()
dt_classifier_rs = RandomizedSearchCV(dt_classifier, param_distributions = param
dt_classifier_rs.fit(X_train, y_train)
```

```
Out[85]:  ▸ RandomizedSearchCV
          ▸ estimator: DecisionTreeClassifier
              ▸ DecisionTreeClassifier
```

```
In [86]: y_pred = dt_classifier_rs.predict(X_test)
         dt_accuracy = accuracy_score(y_true=y_test, y_pred=y_pred) * 100
         print("Accuracy using Decision tree : ", dt_accuracy)
```

```
Accuracy using Decision tree :  84.48448448448448
```

```
In [87]: cm_dt = confusion_matrix(y_test.values,y_pred)
         plot_confusion_matrix(cm_dt, np.unique(y_pred)) # plotting confusion matrix
```



	LAYING	SITTING	STANDING	WALKING	WALKING_DOWNSTAIRS	WALKING_UPSTAIRS
LAYING	183	0	0	0	0	0
SITTING	0	107	63	0	0	0
STANDING	0	28	150	0	0	0
WALKING	0	0	0	180	5	0
WALKING_DOWNSTAIRS	0	0	0	13	118	3
WALKING_UPSTAIRS	0	0	0	37	6	106

```
In [88]: # getting best random search attributes
         get_best_randomsearch_results(dt_classifier_rs)
```

```
Best estimator :  DecisionTreeClassifier(max_depth=8)
Best set of parameters :  {'max_depth': 8}
Best score :  84.97097166534866
```

iv. Random forest model with Hyperparameter tuning and cross validation

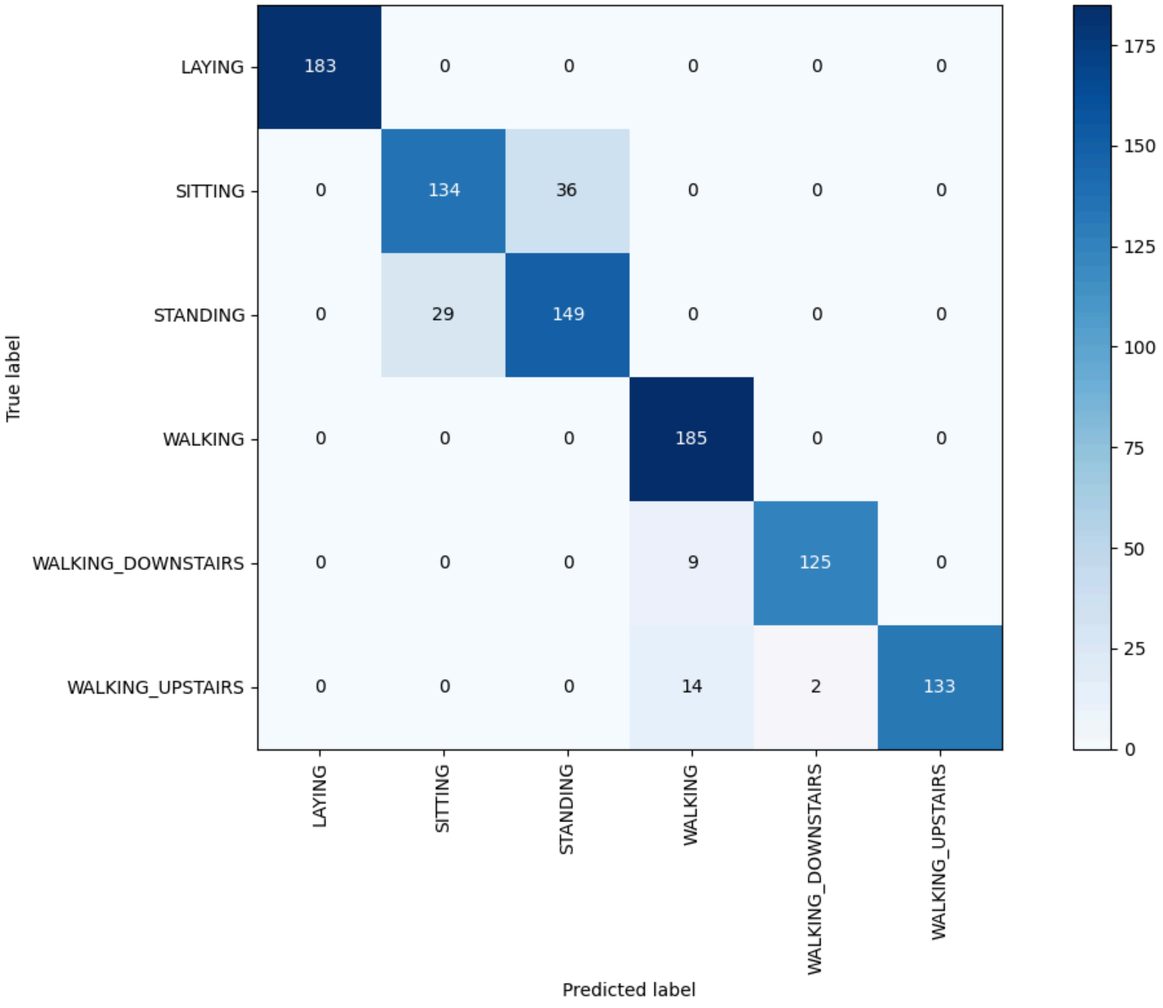
In [89]:
```
parameters = {
    'n_estimators':np.arange(20, 101, 10),
    'max_depth': np.arange(2, 17, 2)
}
rf_classifier = RandomForestClassifier()
rf_classifier_rs = RandomizedSearchCV(rf_classifier, param_distributions=paramet
rf_classifier_rs.fit(X_train, y_train)
```

Out[89]:
```
▸        RandomizedSearchCV
▸ estimator: RandomForestClassifier
      ▸ RandomForestClassifier
```

In [91]:
```
y_pred = rf_classifier_rs.predict(X_test)
rf_accuracy = accuracy_score(y_true=y_test, y_pred=y_pred) * 100
print("Accuracy using Random forest : ", rf_accuracy)
```

```
Accuracy using Random forest :  90.990990990991
```

In [97]:
```
cm_rf = confusion_matrix(y_test.values,y_pred)
plot_confusion_matrix(cm_rf, np.unique(y_pred))
```



In [98]:
```
get_best_randomsearch_results(rf_classifier_rs)
```

```
Best estimator :  RandomForestClassifier(max_depth=12, n_estimators=70)
Best set of parameters :  {'n_estimators': 70, 'max_depth': 12}
Best score :  92.31547792468449
```

In [99]:

```
cols = [
    ["Logistic Regression",dt_accuracy, lr_classifier_rs.best_score_*100],
    ["Kernel SVC", kernel_svm_accuracy, svm_rs.best_score_*100],
    ["Decision Trees", dt_accuracy, dt_classifier_rs.best_score_*100],
    ["Random Forest", rf_accuracy, rf_classifier_rs.best_score_*100]
    ]
results = pd.DataFrame( cols,
                       columns = ["Model","Accuracy %", "Best Score Accuracy %"]
                        by="Accuracy %",ascending=False)
results.style.background_gradient(cmap='Set1')
```

Out[99]:

	Model	Accuracy %	Best Score Accuracy %
1	Kernel SVC	96.596597	94.641093
3	Random Forest	90.990991	92.315478
0	Logistic Regression	84.484484	93.730351
2	Decision Trees	84.484484	84.970972

7. Result:

Kernel SVC shows the Best Score Accuracy.

In [ ]: