DSC 483 Practicum

Human Activity Recognition

Surya Rajaraman Iyer

Introduction

I aim to identify the actions performed by an individual based on the signals collected using an accelerometer and a gyroscope in a smartphone. Being able to identify the actions just from a smartphone can be immensely helpful for a large number of tasks. It can be possible for medics to track patients' activity in a remote manner. It can be used for improving the user experience in Virtual reality where the person can perform actions rather than only visualize environments.

Data

The Data set has been collected from experiments which was carried out by Smartlabs. The experiment was carried out with a group of 30 volunteers aged between 19-48 years.

The data is collected using an embedded accelerometer and gyroscope in a Samsung Galaxy Smartphone. The 3-axial linear acceleration and 3-axial angular velocity was captured at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually by the Smartlabs team.

The data contains 563 columns and 10299 records with the activity column that needs to be predicted. The subject column indicates the volunteer and can be dropped as it will not have an impact on the model predictions. The remaining 561 features are the features which are used to predict the activity of the volunteer. These features were generated by pre-processing and applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window).

There are two major components of the acceleration sensor readings: -

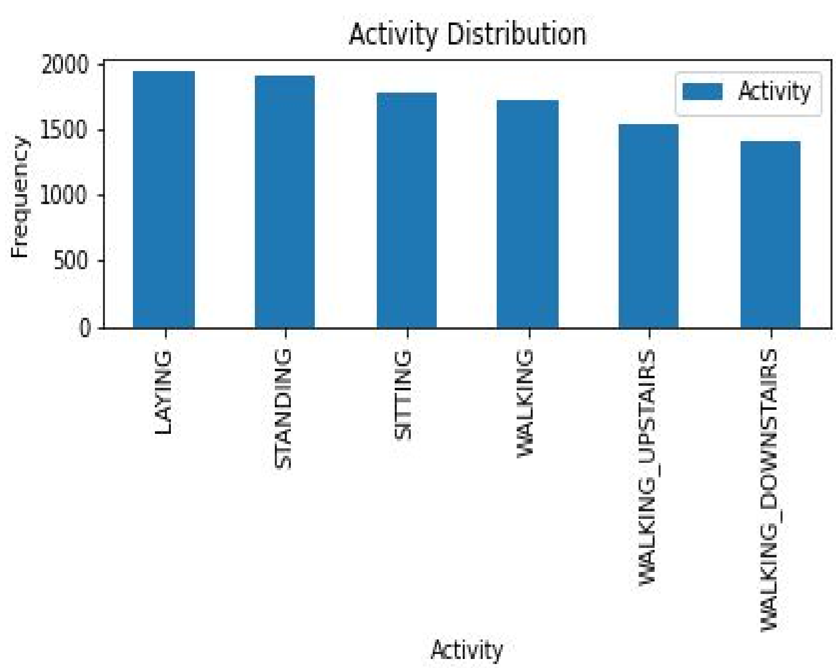
1. Gravitational - Low Frequency component of acceleration signal
2. Body motion - The other components of the acceleration signal

The obtained data set has been randomly partitioned into two sets, where 70% of the data as the training data and the remaining as the test data.

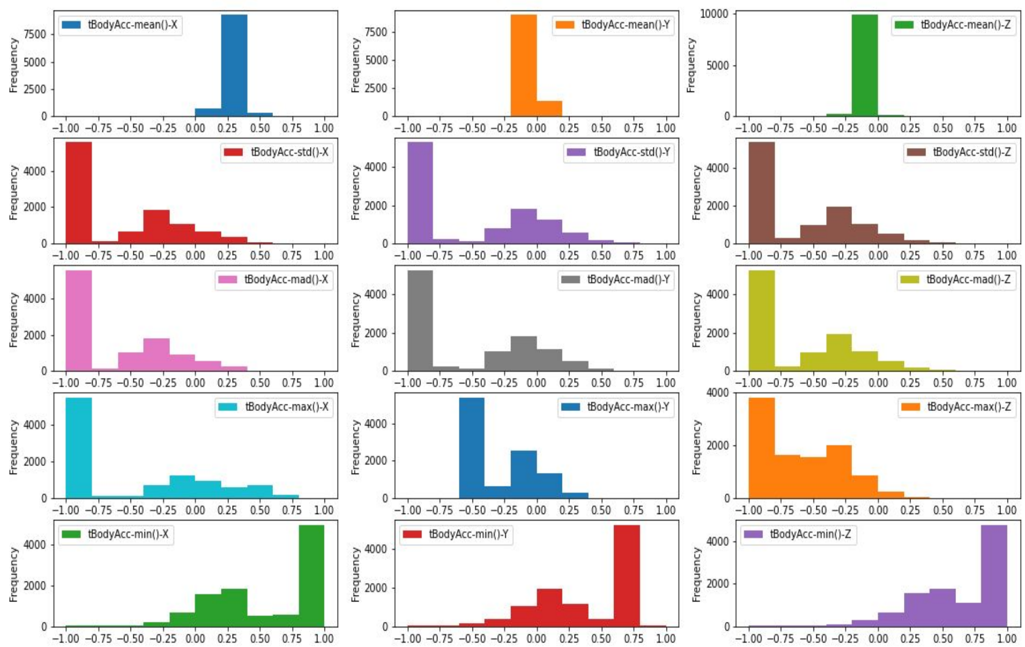
Exploratory Data Analysis

The dataset provided is a clean dataset where all the features (Triaxial acceleration and Triaxial Angular velocity) have no missing values and have been normalized to be in the range [-1,1].

The distributions of the Activity classes are mostly uniform which is good for the prediction model. So we do not need to perform any re-sampling to make it uniform.



While viewing the distributions of the acceleration plots, we can see that the mean is mostly constant whereas the variance has a bi-modal distribution and is spread out between through the range [-1, 1]. That is reasonable because, while performing either of the walking activities we expect the signals to fluctuate and hence have some deviation.

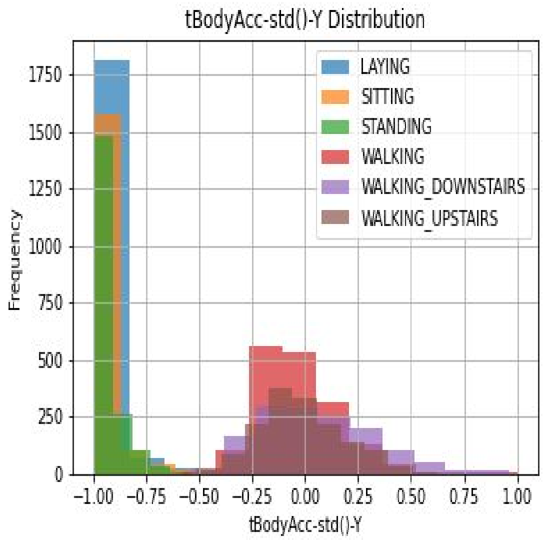


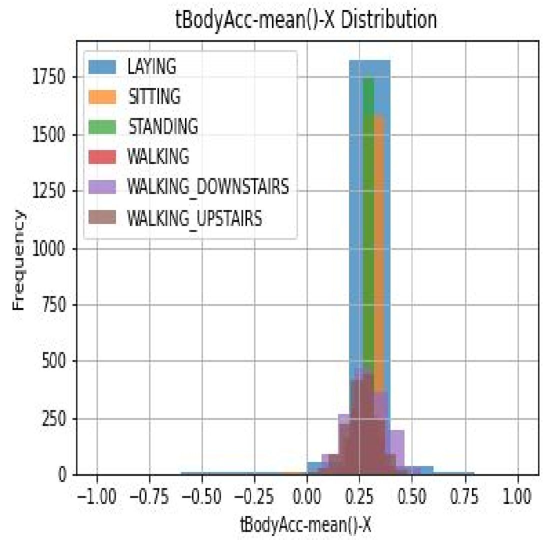
Histograms of the statistic of Body Acceleration

We can therefore expect the mean variables to have lesser impact on the model as compared to the standard deviations. The Mean absolute deviation is similar to that of the standard deviation, so we can expect that only one of the features may be used over the other.

The same observations are seen when we view the mean and standard deviation variables of the Body Angular velocity as well when measured by the gyroscope. Hence, we can conclude similarly that the means will not play an active role in prediction.

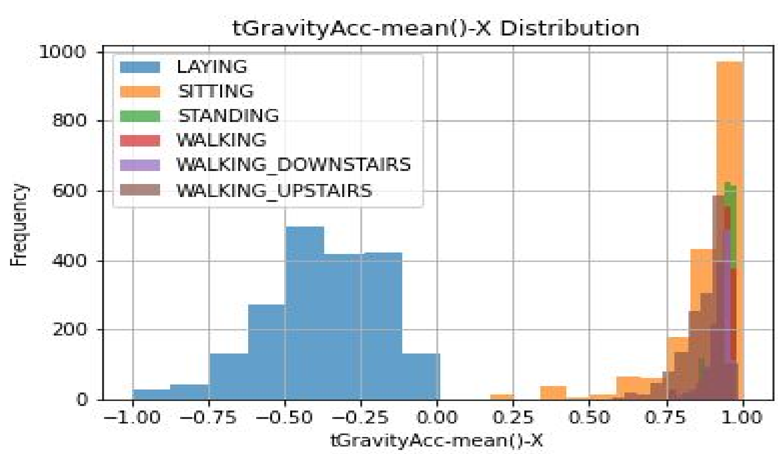
To further solidify that, we can see from the class-wise plots of the means that there is no significant deviation for each class. We can easily distinguish walking from the stationary classes just by using the standard deviation features.



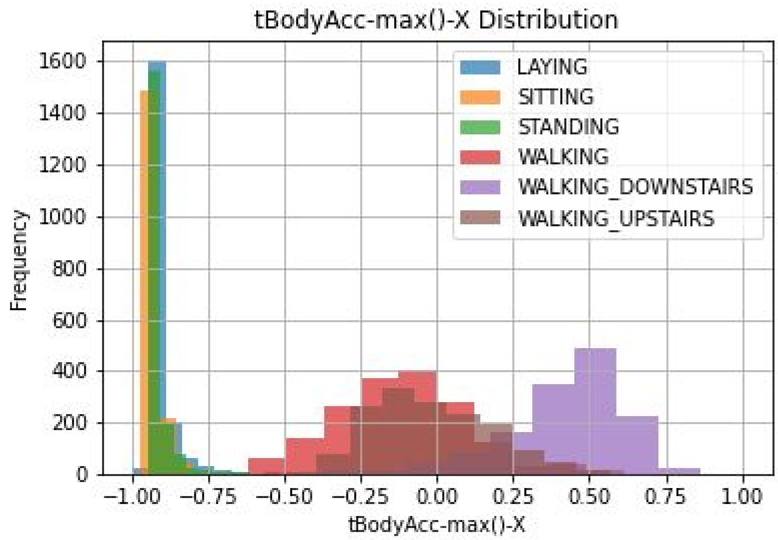


Similar distributions can be seen for the mean and standard deviation of the angular velocity as well.

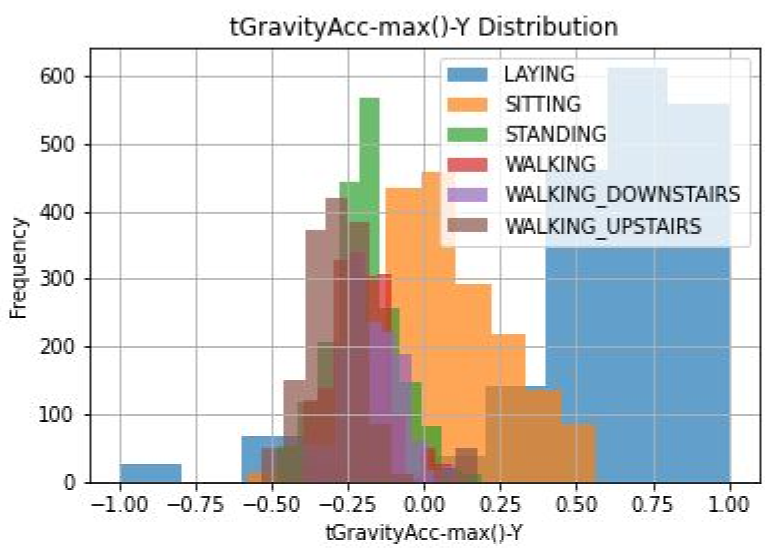
tGravityAcc-mean()-X, The Gravity Acceleration mean distribution shows that it can easily identify the class LAYING easily as there is a clear distinction of the values as compared to other classes.



tBodyAcc-max()-X seems to distinguish the class WALKING_DOWNSTAIRS although there is an overlap with WALKING and WALKING_UPSTAIRS. It can be separated out probably by using a combination with other features in the set.



tGravityAcc-max()-Y can be used to classify SITTING position. There is significant overlap with the LAYING class. But by using it in combination with tGravityAcc-mean()-X, we can separate the features out and reasonably classify the class.



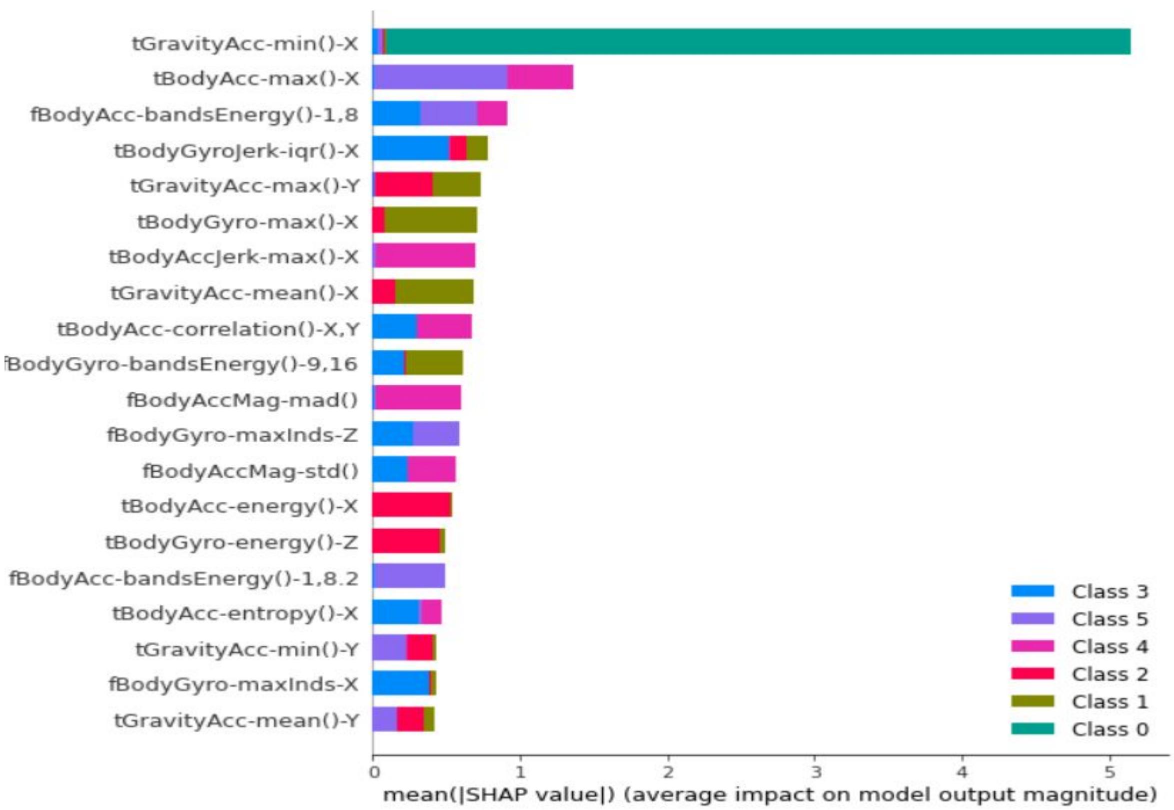
There are multiple features which reasonably separates the classes out. The corresponding plots can be viewed in the following link.

Methods

As we can see from the EDA, the classes can easily be predicted by using a rule based approach. Due to this reason I decided to use XGBoost and a Multi Layer Perceptron classifier for predicting the activity performed by the subject.

XGBoost is a tree based gradient boosting algorithm which can work really well as a rule based model. A Multilayer Perceptron Classifier can easily identify feature interactions to predict the class through multiple iterations of the network.

My initial approach was to use all the features for both of the models and to then compare the accuracy of the models. It is possible to then evaluate the feature importance of the XGBoost model using Shapley Values.



The Shapley value is the average marginal contribution of a feature value across all possible coalitions. Using Shapley values we can assign feature importance to each of the features and remove the non important features from the dataset. The feature importance can be seen above. The class labels for the same are

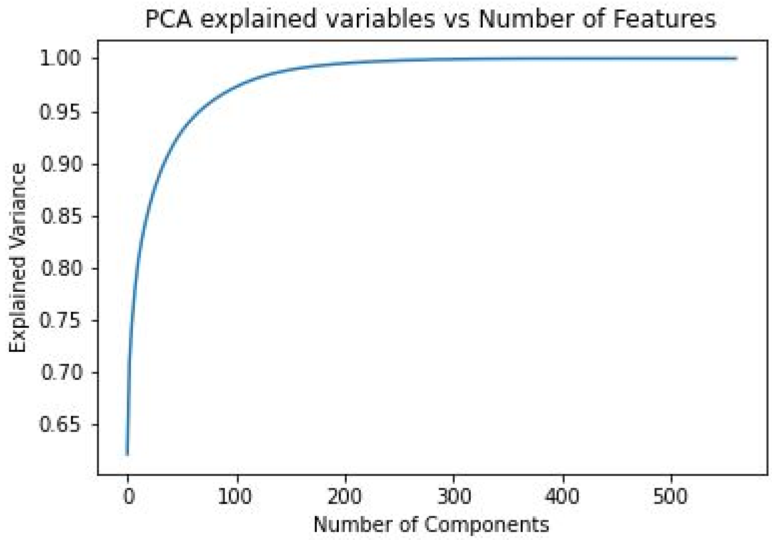
CLASS	LABEL
LAYING	Class 0
SITTING	Class 1
STANDING	Class 2
WALKING	Class 3
WALKING_DOWNSTAIRS	Class 4
WALKING_UPSTAIRS	Class 5

As mentioned in the EDA section, tGravityAcc-mean()-X is marked to be the most important and is used to classify the LAYING class. tBodyAcc-max()- X , the second most important feature plays an important role in identifying WALKING_DOWNSTAIRS and WALKING_UPSTAIRS.

The next step involved feature pruning. Given the list of order feature importance, I added features till I was able to get an accuracy of 90% and pruned out the remaining features. Having a lot of non important features in the model not only makes the training of the model slow due to the huge dimensionality of the data and also may cause issues due to additional data of no importance.

Another method to reduce dimensionality of the dataset is using PCA. PCA transforms the feature space by changing the basis of the data. After transforming the featurespace, we can then choose the components which can explain most of the variance in the data and then use the reduced dimensionality data set for training the model.

The below graph shows the ratio variance explained vs the number of PCA components chosen when the PCA is fit on the training data.



We can see that 90% of the variance is explained when we choose about 35 of the principal components and 95% of the variance is explained when we choose 69 principal components. We can thus reduce the dimensionality without losing much information by using PCA.

I have then retrained the XGBoost model using this reduced dataset by limiting the number of principal components until I am able to achieve 90% training accuracy.

Results

While using the entire feature space, I got the following results for the Xgboost and MLP Classifier: -

Accuracy	XgBoost	MLP
Training	99.94	99.93
Testing	98.28	98.34

The confusion Matrix and metrics for Xgboost Model is as follows. We can see that the LAYING class has been perfectly predicted, however there are very few mismatches among the other classes

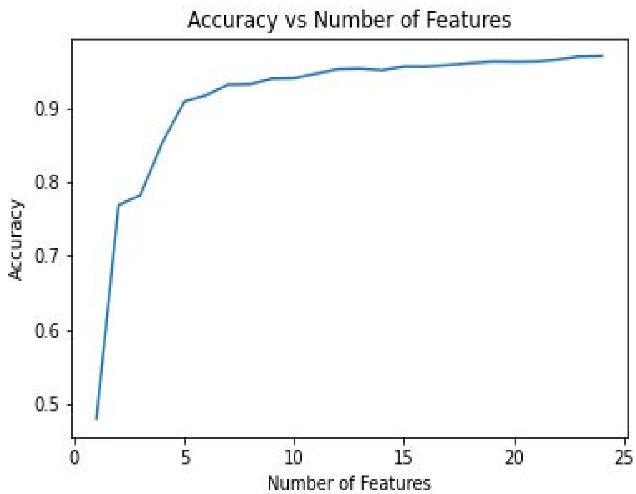
CLASS	LAY	SIT	STAND	WALK	WALK DOWN	WALK UP	Precision	Recall	f1-score
LAY	543	0	0	0	0	0	1	1	1
SIT	0	564	23	0	0	0	0.96	0.98	0.97
STAND	0	14	544	0	0	0	0.97	0.96	0.97
WALK	0	0	0	503	0	3	0.99	0.99	0.99
WALK DOWN	0	0	0	1	438	9	0.98	1	0.99
WALK UP	0	0	0	2	1	445	0.98	0.98	0.98

The confusion Matrix and metrics for MLP Model is as follows

CLASS	LAY	SIT	STAND	WALK	WALK DOWN	WALK UP	Precision	Recall	f1-score
LAY	543	0	0	0	0	0	1	1	1
SIT	0	551	36	0	0	0	0.94	0.97	0.96
STAND	0	15	543	0	0	0	0.97	0.94	0.96
WALK	0	0	0	506	0	0	1	1	1
WALK DOWN	0	0	0	0	448	0	1	1	1
WALK UP	0	0	0	0	0	448	1	1	1

For the MLP classifier, we see that all of the walking classes and the LAYING class has been correctly classified. There are a few mismatches between the STANDING and SITTING CLASS

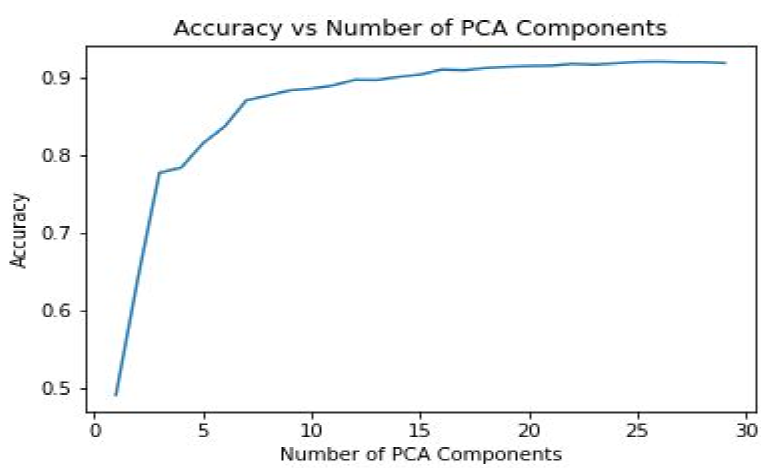
We can see the number of columns to achieve a certain accuracy in the below table and the plot



Number of Columns	Accuracy %
1	48
2	77
4	85
5	91
10	94

We are able to achieve 90% accuracy by only using 5 features and eliminating the remaining features.

After applying PCA, we see that we need at least 14 columns to have 90 % Accuracy. THe graph can be seen as below:-



All of the details of the project can be viewed at Github

REFERENCES

1. UCI Machine Learning Repository
2. Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz. A Public Domain Dataset for Human Activity Recognition Using Smartphones, Computational Intelligence and Machine Learning, ESANN 2013. Bruges, Belgium 24-26 April 2013.