Data Mining - CSE 572

Fall 2018

Assignment 1 Report

Submitted to:
Professor Ayan Banerjee
Ira A. Fulton School of Engineering
Arizona State University

Group: 23
Aikya Shah(1214353568)
Chethan Kumar Kolar Ananda Kumar(1214234150)
Manjunath Darshan Shanthigrama Rangaswamy(1214392270)
Sanjay Narayana(1214137287)
Varun Singh(1214070493)

Table of Contents

Introduction

In this project, we aim to develop a system to understand human activities. We aim to classify different human activities like Cooking, Driving, Eating, Typing etc.,. This phase of the project focused on data collection. We could not collect data before the assignment submission due date. Hence, data already collected using the **Myo Gesture Control Armband.** Then there were two tasks which were needed to perform:-

1. **Feature Extraction:** Select and Implement five feature extraction methods to distinguish between different activities.
2. **Feature Selection:** This task is related to dimensionality reduction of feature space by keeping those feature that shows maximum distance between two classes, Using PCA(Principal Component Analysis).

Phase 2: Feature Extraction

The actions that were considered for comparison were: **Eating and Driving.** The five feature extraction methods that best discriminate the two actions are considered.
In this section, For each extracted feature,

1. The intuition behind selecting such feature is explained.
2. An explanation of feature extraction performed
3. Filename containing Matlab code for feature extraction
4. Plots to analyze each feature which have significant difference
5. A discussion that evaluates our intuition

Below are the five extraction methods used in this phase:

I. Fast Fourier Transform
II. Min
III. Variance
IV. Max
V. Mean

I. Fast Fourier Transform (FFT):

a. The intuition behind using FFT as a feature extraction method is that the frequency pattern for the amplitude of the signal varies for various muscle movements. Applying FFT on the EMG sensor should provide a clear distinction in the amplitude of sampled frequencies clearly indicating the contraction and relaxation of muscles associated for each gesture. Similarly, FFT on the Accelerometer and Gyroscope should distinguish the gestures based on the different linear accelerations and angular momentums associated with every gesture.

b. The raw data measured by each sensor in time domain is converted to frequency domain by using built in fft function in Matlab. The sample frequency for each sensor is calculated. Upon plotting and analyzing we observed that only EMG sensor has significant difference for the Eating and Driving actions. From EMG, three maximum amplitude peaks are identified as features.
c. The Matlab code for this feature extraction can be found in “feature_extraction.m’
d. Plot for FFT on EMG sensor for Eating and Driving action can be found below.
e. Other than EMG 3 and EMG 4, other EMG readings showed significant difference for the two actions. We noticed relatively higher EMG readings of FFT for driving action compared to eating action.

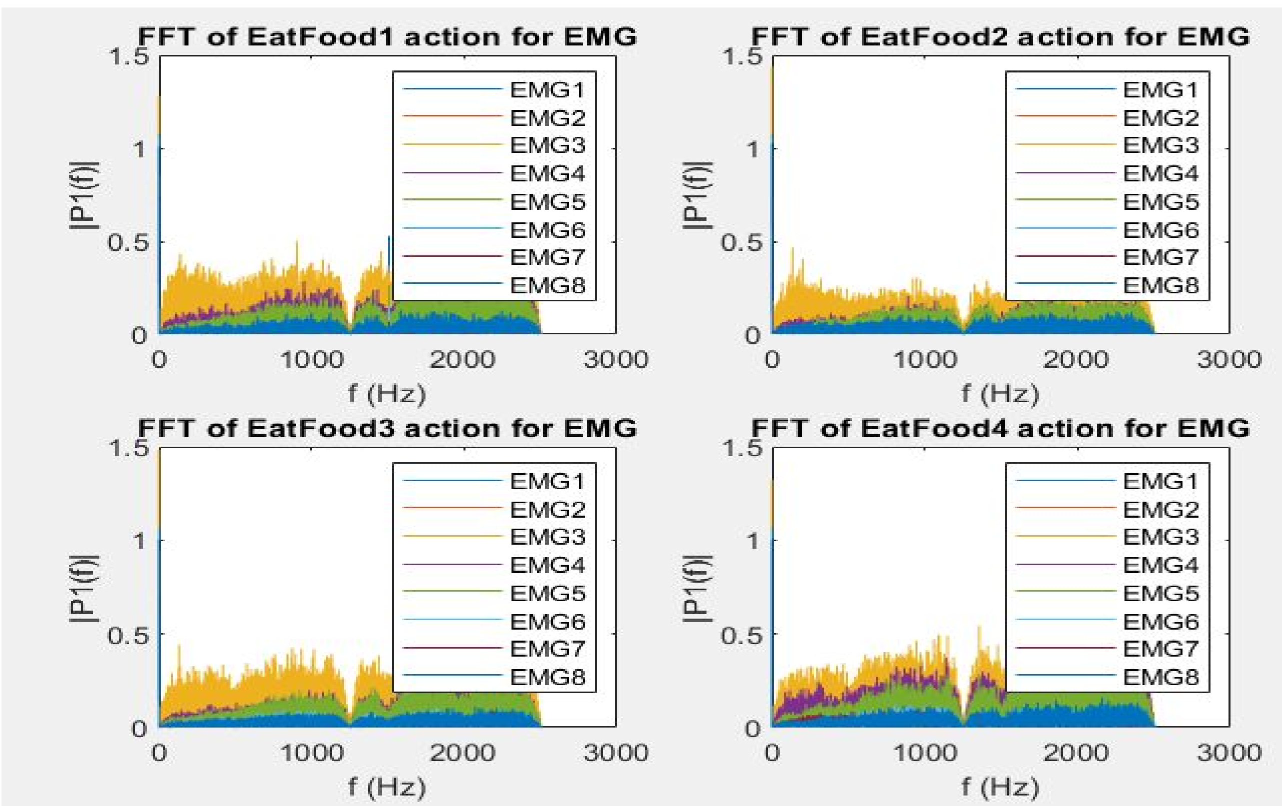


Figure: Plot of FFT signals for the EMG sensors for 4 instances of ‘Eat’ action.

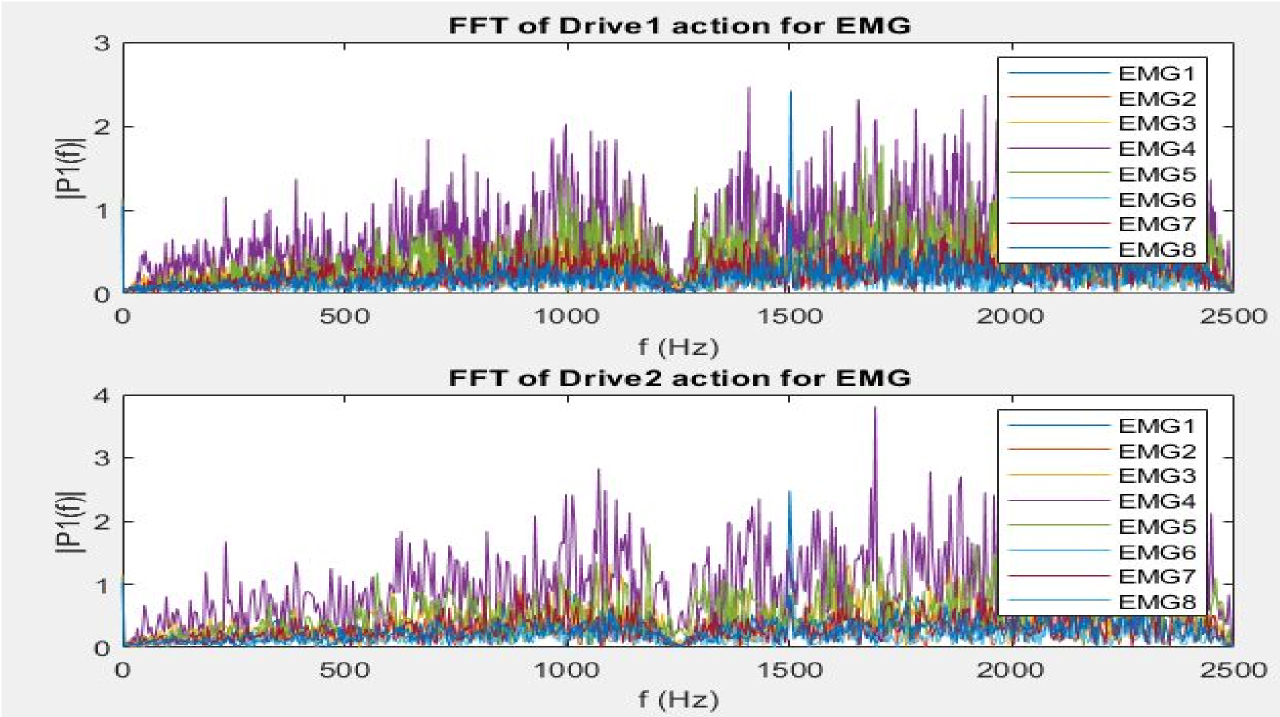


Figure: Plot of FFT signals for the EMG sensors for 2 instances of ‘Drive’ action.

Variance:

a. The intuition for using variance as a feature extraction method is because we expect to see high variance with respect to certain sensors for one action or the other. For example, we expect to see high variance for linear acceleration along Y-axis for Eating compared to Driving.
b. We anticipate variance on the gyroscope sensors to clearly show the difference in variations in angular velocity between the two different actions.
c. The Matlab code for this feature extraction can be found in “feature_extraction.m’
d. The plots for variance feature method can be found below.
e. As expected we noticed high variation between the two actions for the gyroscope sensors. In addition, we also noticed high variation for the Orientation-Euler and Orientation sensors. We attribute this to the different kind of orientations while performing both actions.

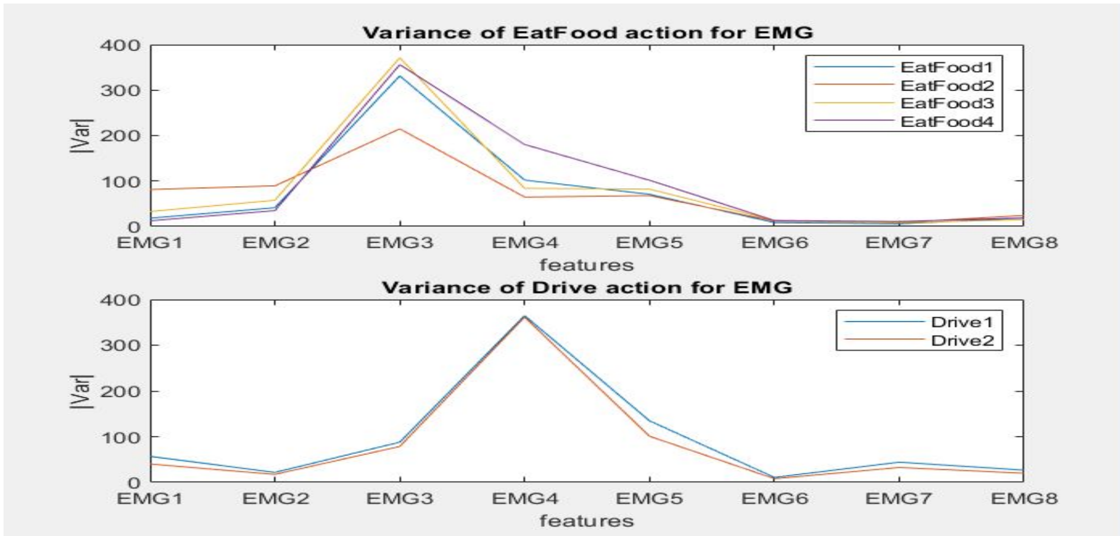


Figure: Plot of Variance for the EMG sensors for both actions.

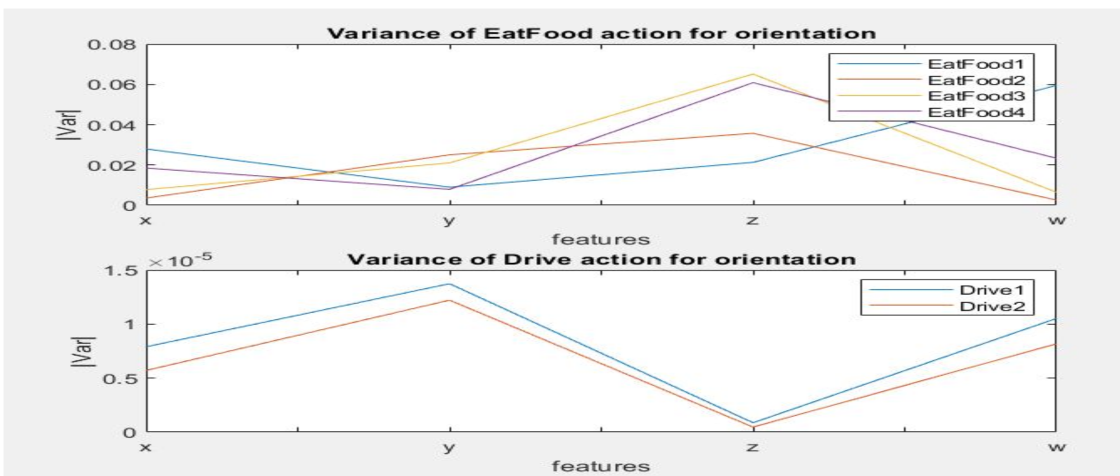


Figure: Plot of Variance for the orientations sensors for both actions.

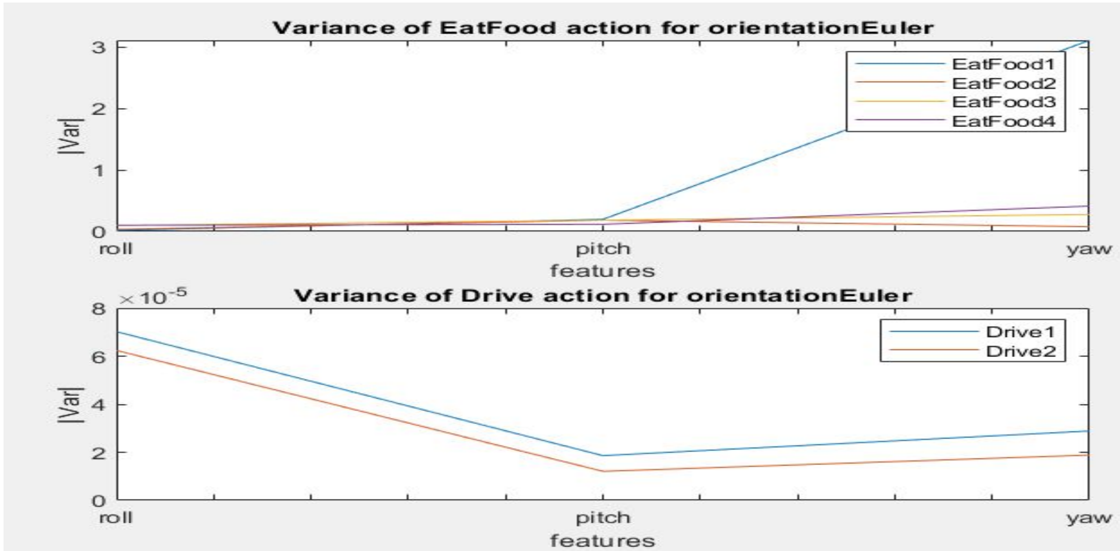


Figure: Plot of Variance for the Euler orientation sensor for both actions.

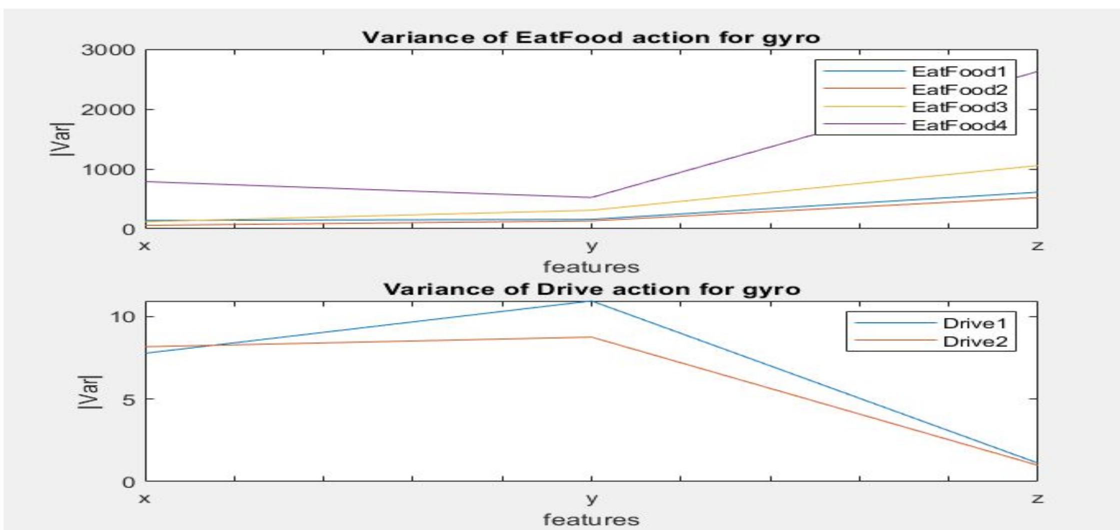


Figure: Plot of Variance for the Gyro sensors for both actions.

Min/Max:

a. The Intuition behind using Min/Max is to find the values that can help to distinguish different activities how the min and max values differ for them.
b. we noticed a huge difference between the Min values of Gyro across all the axises between cooking and driving. Along with EMG values. Same we found for the max values. So, this helped us to deduce that angular velocity and orientation were more in cooking rather driving.
c. The matlab code for finding min and max can be found in feature_extraction.m

d. Plots for the min/max can be found below.
e. Form the above two graphs for min values on Accelerometer sensor for cooking and driving we don’t see any significant difference. But, For Gyro sensor(Plot below) we can clearly see the difference between min values of cooking and driving. Thus, Validating our intuition for using min for gyro sensor
   Similarly, for Max values:
   The max values of accelerometer and orientation sensor on cooking and driving did not showed any significant difference. But, for Gyro and Emg data the variation between cooking and driving was significant as shown below:

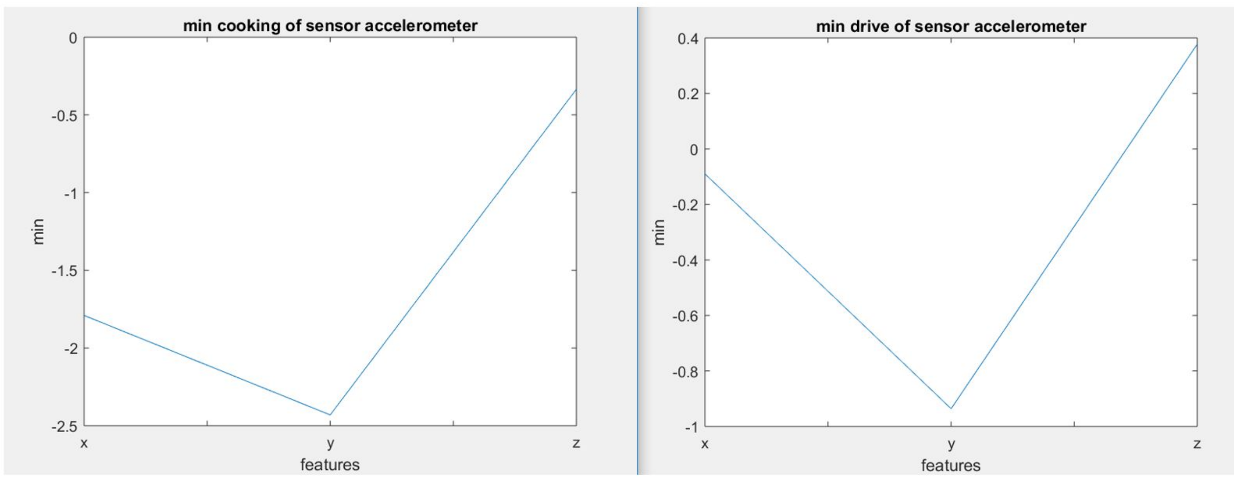


Figure: Plot of Minimum for the accelerometer sensors for both actions.

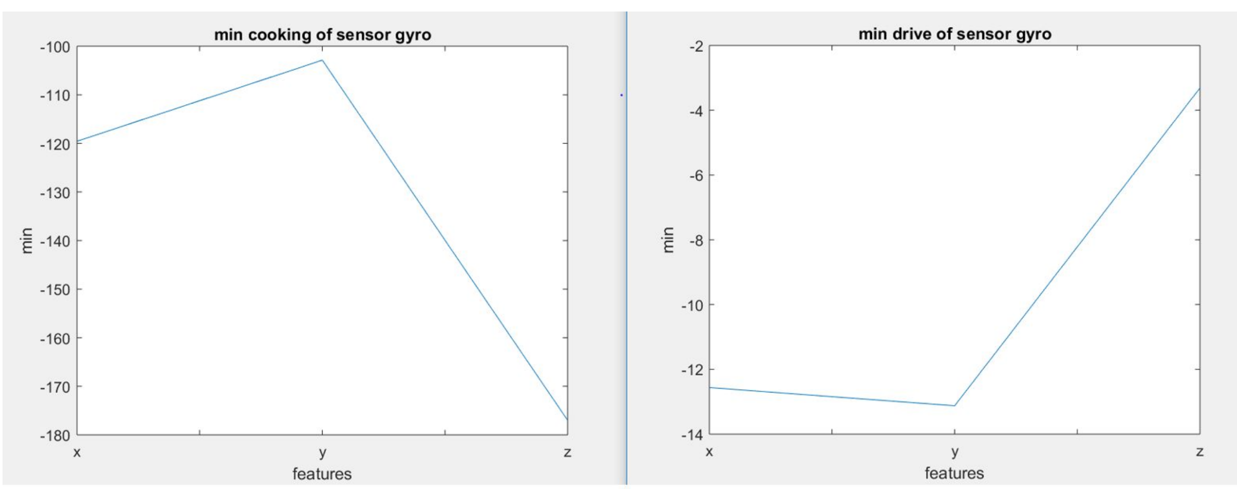


Figure: Plot of Min for the gyro sensors for both actions.

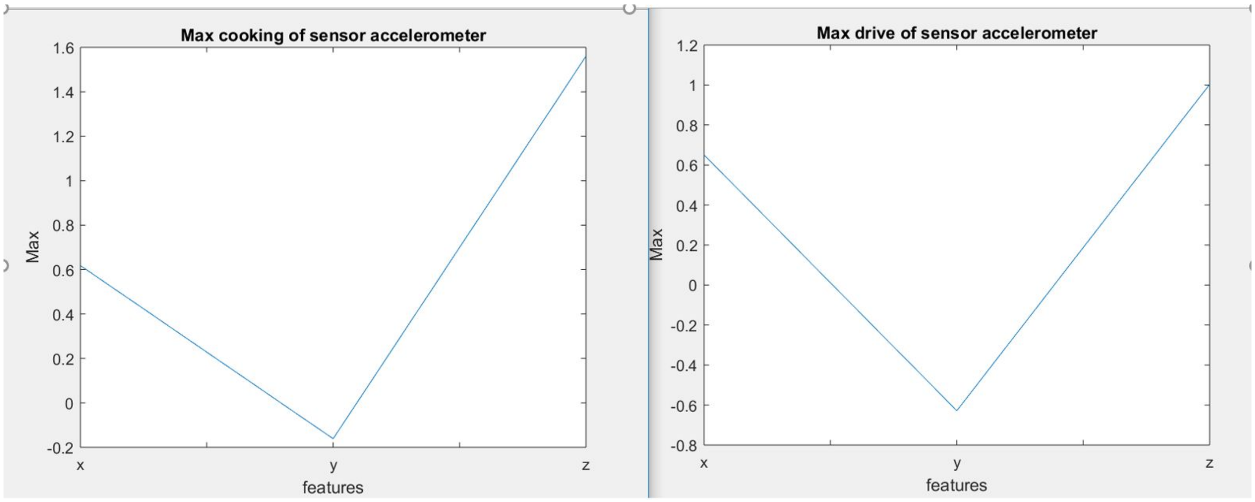


Figure: Plot of Max for the accelerometer sensor for both actions.

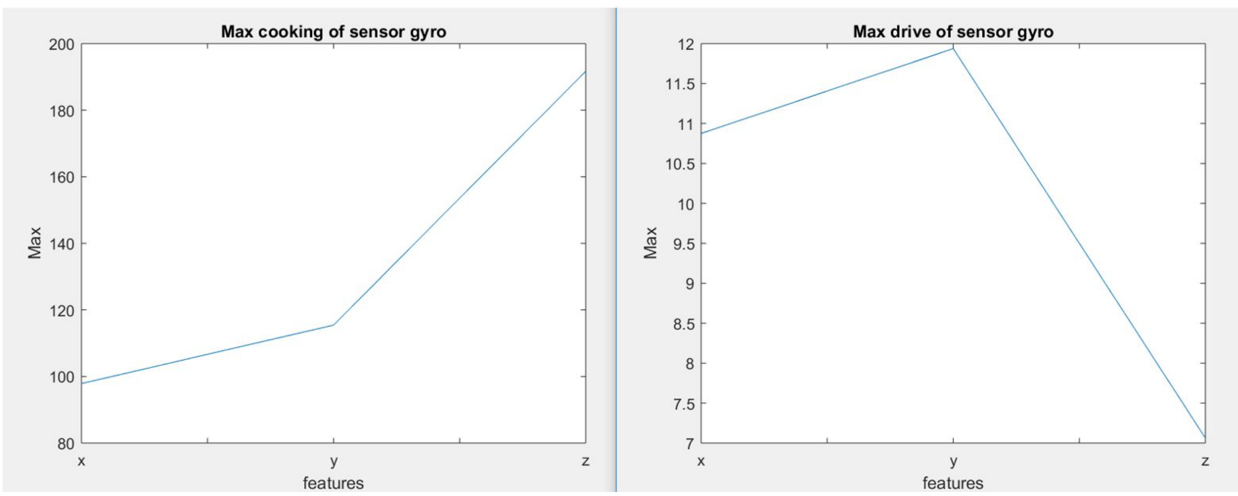


Figure: Plot of Max for the gyro sensor for both actions

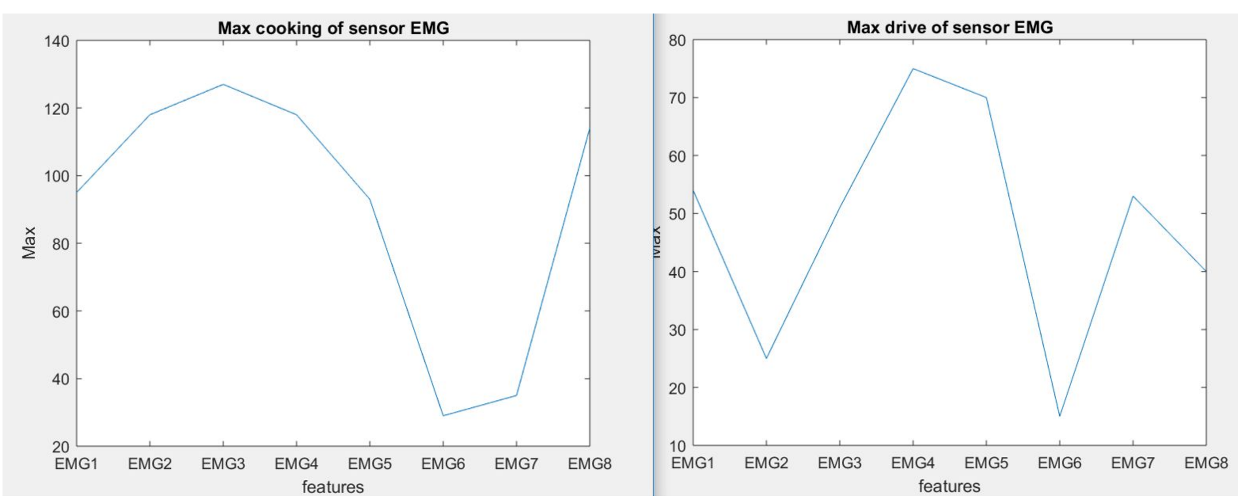


Figure: Plot of Max for the EMG sensors for both actions

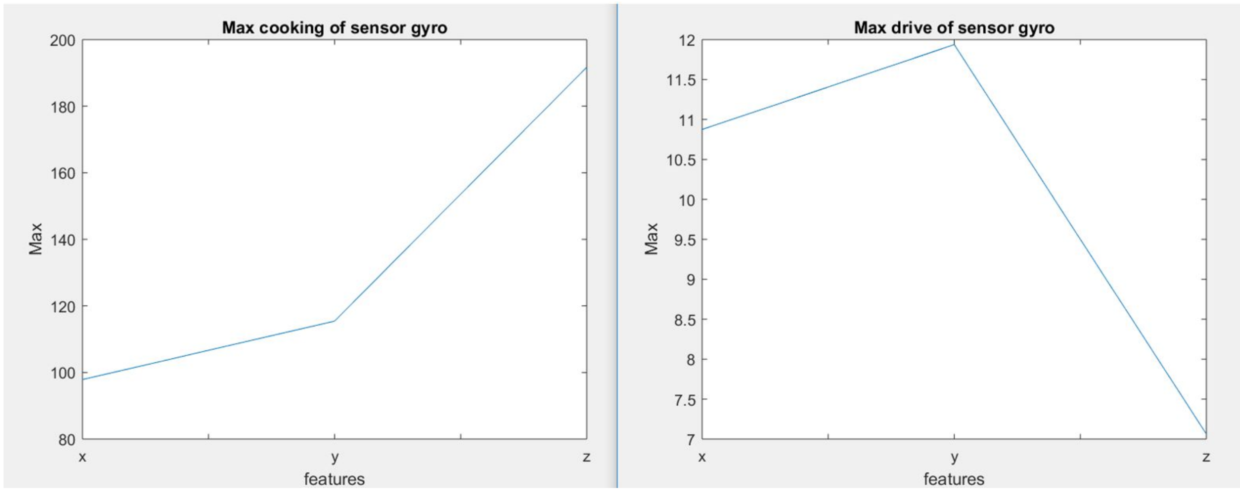


Figure: Plot of Max for gyro sensors for both actions

Mean:

a. The intuition for using Mean is that there is usually a noticeable difference between the mean values for Eating and Driving. For certain sensors like Orientation data we anticipate that the sensor values for the two actions will be skewed differently with respect to the medians.
b. The mean value of a set of data gives the idea of the range the values lie in.
c. The matlab code for finding mean can be found in ‘feature_extraction.m’
d. Plots for the mean can be found below.
e. As we can see in the plot for mean of Eating and Driving data using the Orientation Sensor, the actions show different mean values for each of the orientation sensors. So, we can select the Orientation X,W and Z components for the Feature Matrix construction.

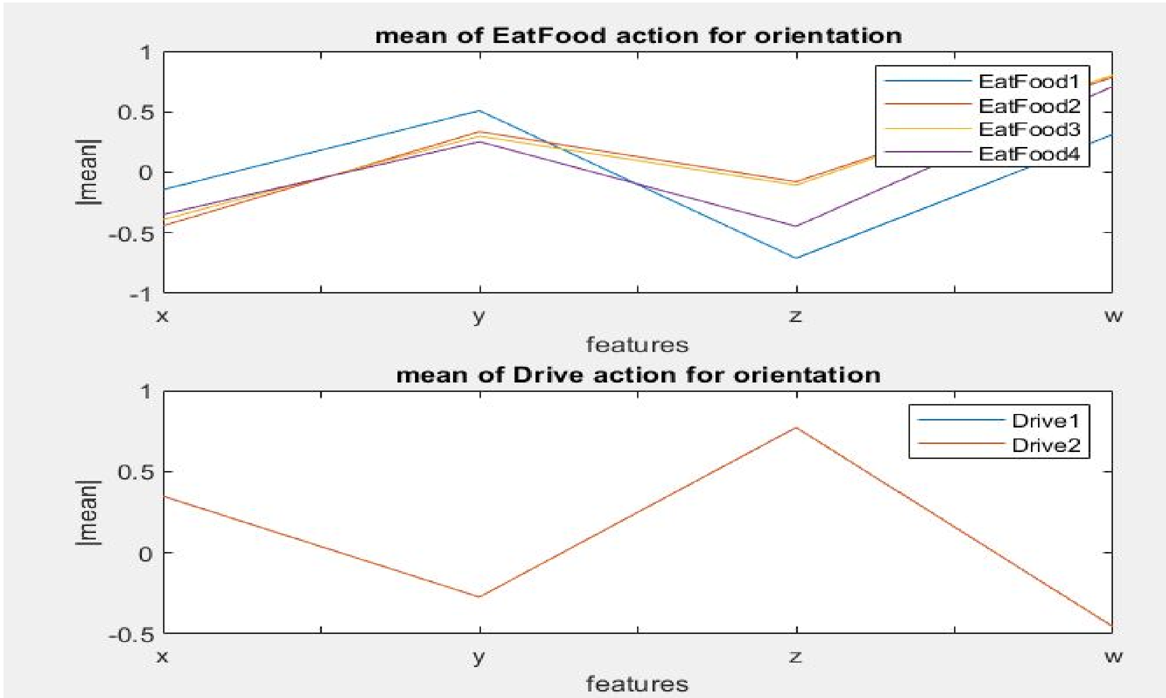


Figure: Plot of Mean for the Orientation sensor for both actions

Phase 3

Subtask 1:

The logic for arranging the feature matrix is coded using MATLAB and is attached in the file - 'FeatureMatrix.m'. Two feature matrices were created, one each for Eating instances and Drive instances.

Amongst the 147 features obtained using Feature Extraction methods such as FFT, Mean. Min, Max and Variance, 48 features were selected to construct the Feature Matrix. Three peaks were considered from each selected FFT component of the sensors. The selected features are :

Feature Extraction Method	Features Selected
Mean	Orientation {X, W, Z}
Min	Gyro {X ,Y, Z} , EMG {1,2,8}
Max	Gyro {X,Y,Z}, EMG {2,3,8}
Variance	Orientation Euler { X }, Gyro {X, Y, Z} Orientation { X, Z}, EMG {3,4,7}

FFT	3 peaks selected each for EMG 1 to 8

The dimension of the Feature Matrix is:
Rows : Number of Action Instances
Columns : Number of Features Selected

For example, In the sample data for the action EatFood, we have 4 instances. The feature matrix will be a 4x48 matrix. For the action Drive, we have 2 instances and hence a 2x48 feature matrix.

	1	2	3	4	5	6	7	8
1	-0.1432	-0.7105	0.3127	-99	-128	-128	-119.5630	-102.8750
2	-0.4421	-0.0776	0.7867	-128	-120	-128	-130.5000	-124.5000
3	-0.3907	-0.1056	0.8042	-128	-128	-105	-145.5630	-151.6880
4	-0.3490	-0.4463	0.7103	-82	-126	-109	-233.4380	-181.6880

Figure: Screenshot of the first 8 features of the Feature Matrix created for 4 instances of EatFood Action

Subtask 2:
MATLAB's inbuilt function 'pca' was used to extract the principal components of the two feature matrices.

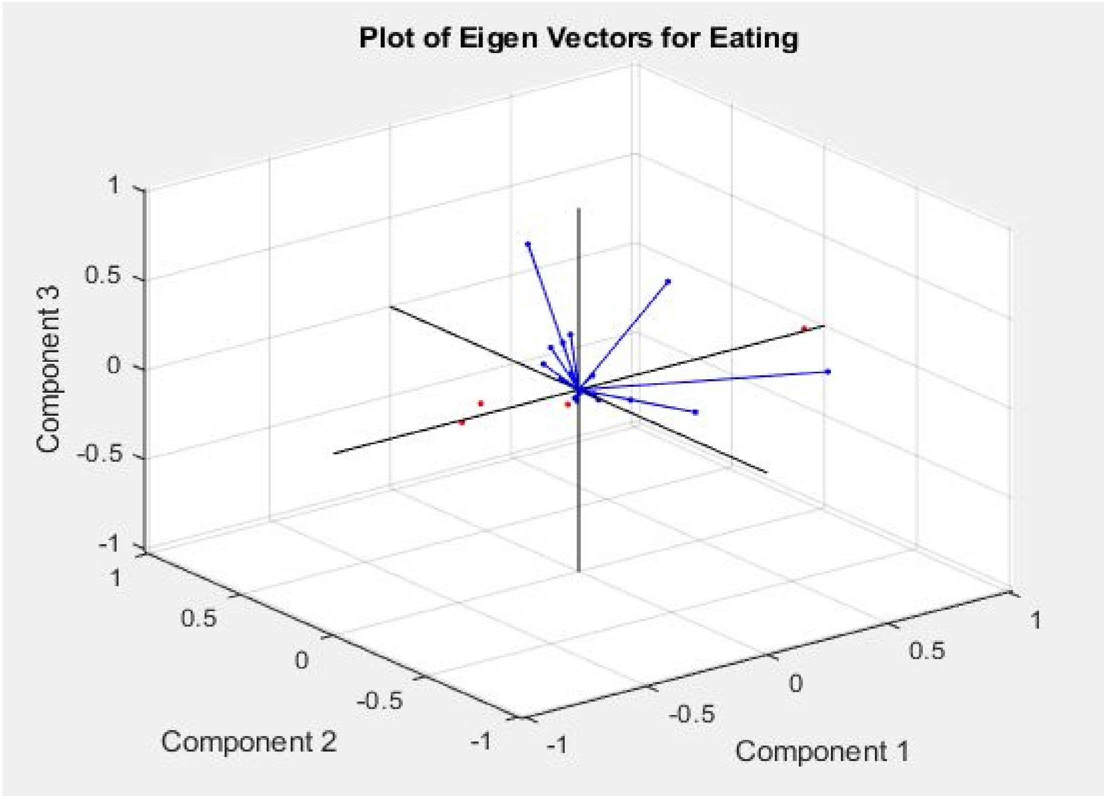
Principal Component Analysis

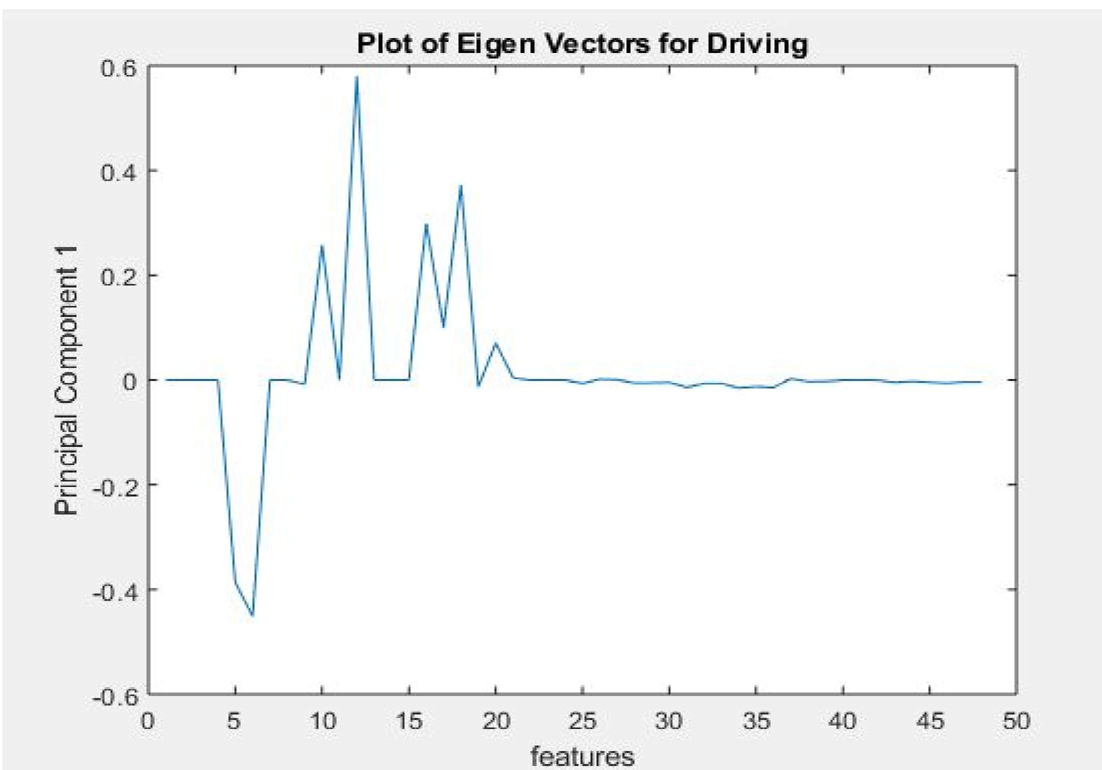
Function Call :

```
[coeff_drive,score_drive] = pca(Feature_Matrix_drive);
[coeff_EatFood,score_EatFood] = pca(Feature_Matrix_EatFood);
```

The pca call returns Coefficient Matrix and the Score Matrix. Each row of the Coefficient matrix matrix corresponds to a Eigenvector which is nothing but the projection of each feature on the Principal Component. When we apply pca in Matlab, the number of principal components returned = **min(number of continuous variables - 1, number of variables)**. For our project, number of continuous variables are the number of instances and the number of variables is the number of features selected after feature extraction. We can also specify the number of Principal Components required, but in our case we don't do that as we have very few instances of actions.

EigenVector Plots of Eating and Driving





The plot of Eigenvectors for Driving is One Dimensional as there is only one Principal Component for Driving.

Subtask 3:

Eigenvectors

PCA returns a set of principal components, which is a set of mutually uncorrelated orthogonal vectors. These components represent variability of the data. Each component representing high variance and each orthogonal to the previous component.

The application of pca on the feature matrix of EatFood returns 48 eigenvectors of size 3 as we have 3 principal components. Similarly for Drive we get 48 Eigenvectors of size 1 as we have only one principal component. Each Eigenvector corresponds to the transformed variable values of a certain action instance.

Subtask 4:
New Feature Matrix

The new feature matrix is created by multiplying the original feature matrix say X with the eigenvectors 'V' obtained using pca. For Eating, the new feature matrix has the dimension 4 x 3. The new feature matrix corresponds to the projection of the original data points onto the new dimension formed by the principal components. The code for creating the new feature matrix is in the file - 'FeatureMatrix.m'. The new feature matrices obtained for EatFood and Drive are shown in the figures below.

The process for obtaining new feature matrix for EatFood:
Original Feature Matrix(X) -----PCA----> Coefficient Matrix----X---> New Feature Matrix
4x48 48x3 4x3

The process for obtaining new feature matrix for Drive:
Original Feature Matrix(X) -----PCA----> Coefficient Matrix----X---> New Feature Matrix
2x48 48x1 2x1

	1	2	3
1	-621.4017	49.2306	70.7092
2	-739.4172	60.6309	-64.3418
3	-181.7658	-142.0211	-3.5872
4	1.5426e+03	32.1596	-2.7802

Fig: New Feature Matrix for EatFood

	1
1	15.5210
2	-15.5210

Fig: New Feature Matrix for Drive

Subtask 5: PCA is not very useful in our case. This is because the features considered far outnumber the number of observations. In our case, PCA reduces the number of observations rather than the number of features. In general, the number of principal components returned is given by the expression : **min(n-1,p).** Where n denotes the number of observations and p denotes the number of features. Since we haven't performed the data collection phase before the submission due date, we considered the sample data for our experiments. p is 48 for our considerations and n is 4 for ‘driving’ and 2 for ‘eating’.

REFERENCES

https://en.wikipedia.org/wiki/Accelerometer

https://imotions.com/blog/electromyography-101/

https://www.livescience.com/40103-accelerometer-vs-gyroscope.html

https://dev.fitbit.com/build/guides/sensors

https://www.youtube.com/watch?v=GZ3KUPqA1JM