FEATURE EXTRACTION AND PCA ON CGM TIMES SERIES DATA

---

1. Introduction

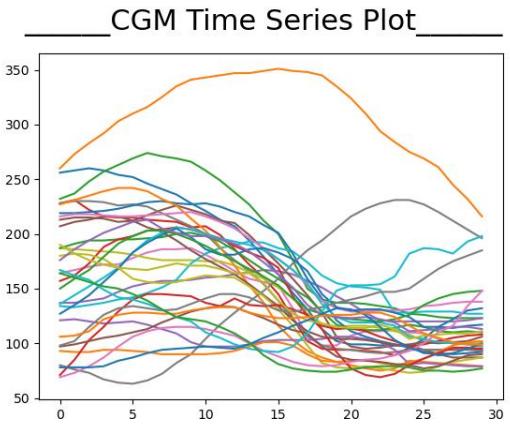
In this project, we study the techniques used to extract the features and analysis of the principal components of the data to get the features with higher importance amongst others. The data provided is of meals eaten by a person and the glucose CGM levels are analyzed according to a given 2-and-a-half-hour time frame. The time starts from 30 mins before having the meal and continues to another 2 hours after starting to have the meal. The approach towards this project has been threefold wherein I first preprocessed the data, extracted the features and found the principal component analysis components using the feature matrix. My findings in this project have been complimented with representative graphs and thus portray my findings while working on the project as well as the final results.

2. Project Phase 1: Pre-processing

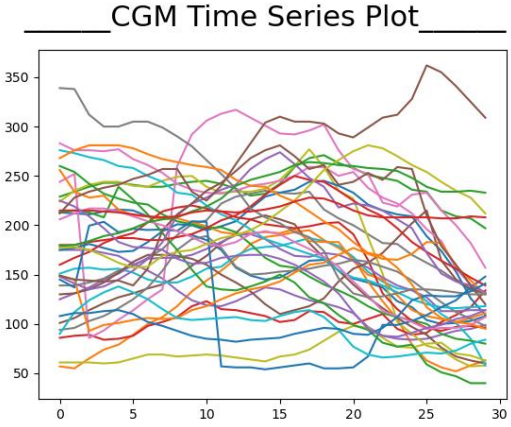
The first phase of the project comprises of the preprocessing of the data. The data collection is done by the CGM sensor. Initially, keeping into consideration that the data has a lot of NaN values and a lot of missing data which might not be very good for future analysis I move towards preprocessing the data. In order to preprocess the data, I first took a look at it and removed unnecessary additional rows which had falsified data. My approach towards cleaning of the data led me to removing the data if it had a majority of NaN values which might not provide appropriate knowledge to train the model in the future so I keep the passing mark of less than 40% NaN values. This removes those rows from consideration but now to tackle the remaining NaN values, I quadratically fit the data to find the missing values using interpolation techniques of order 2. Also, as 31st column data has maximum missing values I completely pruned that row from consideration and moved further towards feature extraction.

Now, the data has been fully preprocessed. It has 30 rows covering 2 and a half hours of time duration at 5-minute intervals. A graphical representation of the data for each patient:

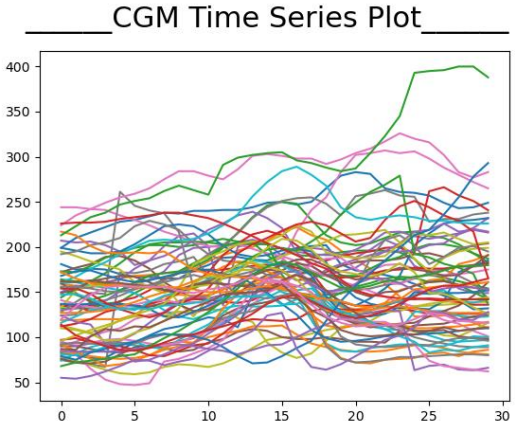
Anish Oswal (aaoswal@asu.edu)



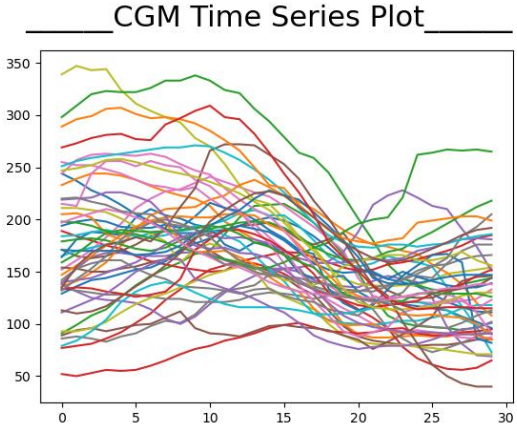
Patient 1



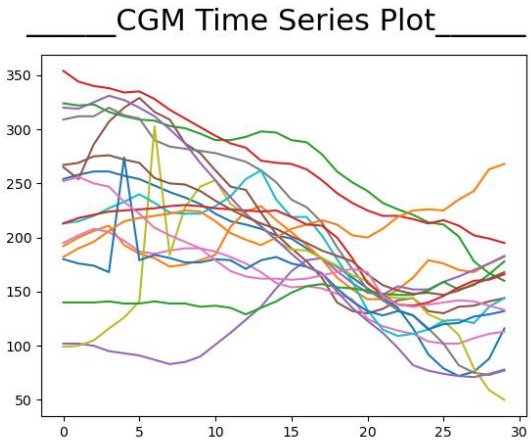
Patient 2



Patient 3



Patient 4



Patient 5

Anish Oswal (aaoswal@asu.edu)

3. Project Phase 2: Feature Extraction

In this phase, I have selected and implemented four feature extraction methods for the given data. I have carefully selected moving features with a window of 10 and with 40% overlapping as they provide more features which are distributed between different parts of data.

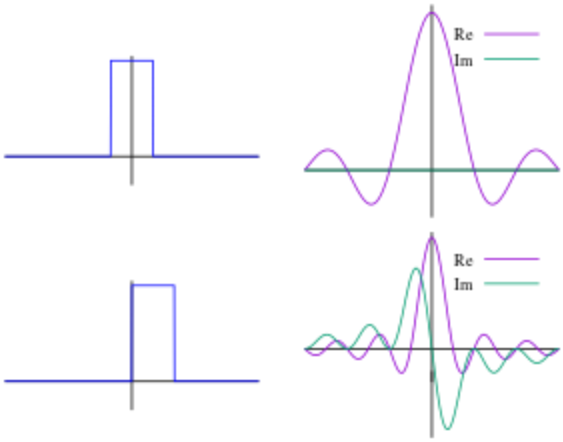
The four feature extraction techniques that I have used are:

1. Fast Fourier Transform for real input
2. Moving Kurtosis
3. Moving Standard Error of Mean
4. Moving Average

3.1 Fast Fourier Transform for Real Input (rfft)

Fast Fourier transform is an algorithm that computes the discrete Fourier transform of the given series. It converts a one-dimensional transform into a valued array as an output which portrays the most dominating frequencies. As the given series of data is time based, we can use FFT as a feature extraction technique which will be an optimum choice for consideration. It gives us 5 features.

Intuition:
As the time series-based data continuously has movement, a frequency-based feature metric is optimal to use. FFT works well with the data as it divides the data into frequency distributions of the features. This helps us get better data distributions and specially as CGM is based off time movement, it aligns perfectly with the properties of FFT and hereby can help in future predictions on such data.



3.2 Moving Kurtosis

Kurtosis is a measure of the date being either heavy-tailed or lightly tailed and is a relative type of distribution. On getting a high kurtosis values we tend to assume that the data has more outliers whereas if the value of kurtosis is low then we say that the data has little to no outliers at all. One of the extreme cases for kurtosis would be a normal distribution. For the project, I have considered a moving kurtosis of a window size of 10 where 40% values can be overlapped. This enables us to get multiple features for a particular data, specifically 5 features from the given data per patient observed. It gives us 5 features.

Intuition:
Kurtosis being a relative distribution helps us identify large values present in kurtosis which portray the ends of the distribution. Comparatively, a normal distribution fails to portray such values vividly. If the kurtosis value is nearing to 0 then the distribution can be considered normalized in nature with noise. A moving kurtosis window enables to handle unnecessary noise efficiently. All of the features of kurtosis with regards to the tail feature will help us estimate the food consumption patterns of different patients and help us get the peaks which help us understand the food consumption time.

3.3 Moving Standard Error of Mean

Standard error of mean represents the error in the mean of the data that is being referred. It calculates the error and the smaller the error, the more it is representative of the actual data. A higher value for the standard error of mean would represent the occurrence of more random variables, thus providing the measurement of the spread. It reflects the accuracy of the data and identifies if the data has a lot of outliers and noise. It can be summarized as the standard deviation of the mean data. It gives us 5 features.

Intuition:
The standard error of mean is an important statistical feature for inference of the data. It portrays us the spread of the data. It helps us find noise present in the data. Considering a moving standard error, it helps us identify the spread among different parts of the data and better tackles any behavior of the different patients. It can tell us about the mean intake of the meals and identify the patterns optimally on how the CGM levels are affected after and before the meal appropriately. Lower error would tell us that a patient has eaten the meal already and higher error would tell us how the patient might be having a meal soon to predict and formalize the meal timings in the data.

3.4 Moving Average

Moving average helps us refine the variation in-between the time stamps which removes unnecessary noise and formulate the underlying information in the data optimally. They are very commonly used in time series datasets due to their properties. The window size of is 10 and with 40% overlapping

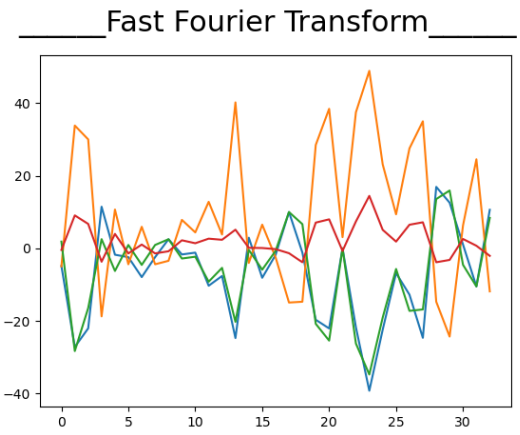
threshold. The moving average generates a new soothed feature representing the data and can also help in training the model for future predicting similar values related to the same data domain.

Intuition:
Moving average soothes out unnecessary noise or error in data as any noise or outlier would have less of an impact on the results because of the average value. Hence, I consider this feature method to be useful. With the help of the soothing feature of the moving average I am able to see the time stamps where the glucose levels are going up from the data. This is especially useful to know and analyze if the person has had a meal or not given that we know that the glucose levels shoot up when and after taking a meal.

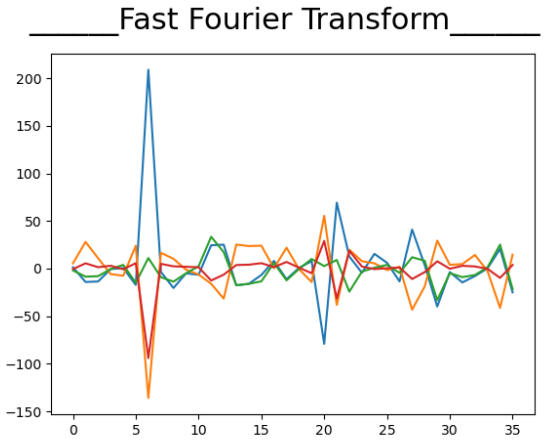
4. Validation of Features

4.1 Fast Fourier Transform

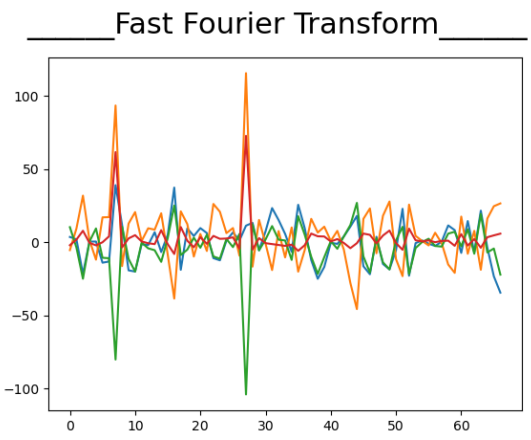
FFT can be considered an optimal feature as it does not change the data itself but just translates the time series data into frequency terms making it easier to observe trends in the data. The graphs represent peaks which are distinct for each patient. These peaks show us the highest CGM values present in the graph which can be interpreted as the times of the meal intake. Defining the meal intake time is necessary as it helps us map the time and the CGM level together for further study. The other smaller peaks represent noise in the data most of the time until we consider a sudden increase in the CGM levels due to an unexpected event or an external factor. Thus, looking at the information that can be extracted from FFT we can say that it is an optimal feature helping predict mealtimes.



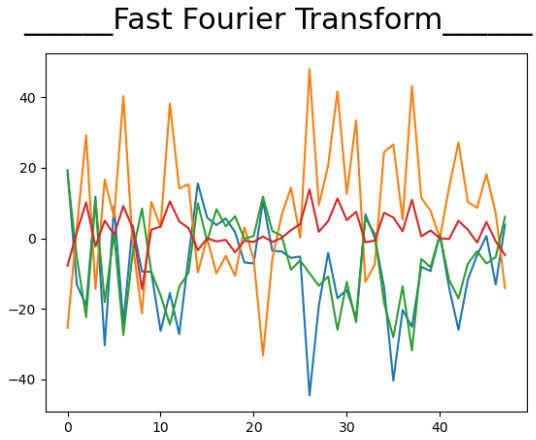
Patient 1



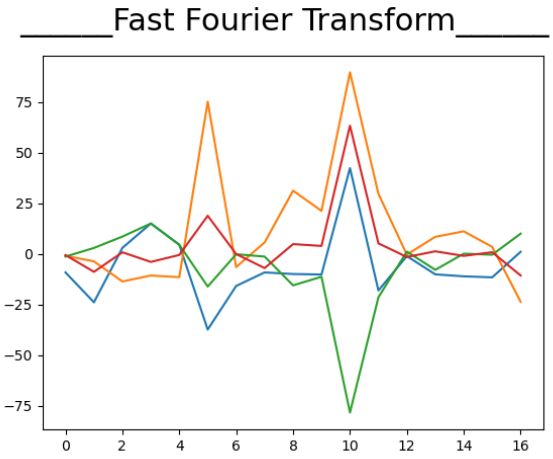
Patient 2



Patient 3



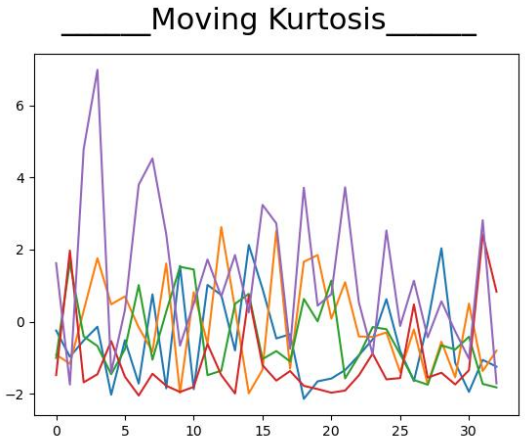
Patient 4



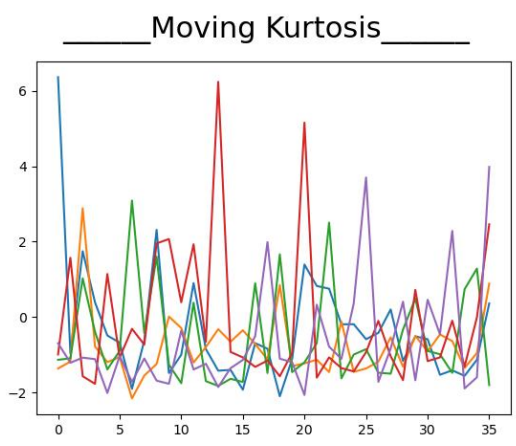
Patient 5

4.2 Moving Kurtosis

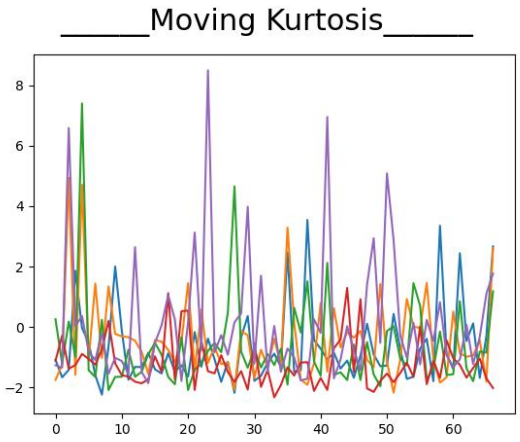
Kurtosis in the graphs portrays the peaks that are represented by the data. These peaks are not absolute but relative to the rest of the data for the patient. Considering the tail property of kurtosis, we can understand the different changes in CGM values and compare that data with other patients but that does not help in the analysis. But since, the peaks generated by the kurtosis function cannot find a single absolute peak point in the distribution we cannot consider it as a valuable feature to find the high CGM values and understand the meal time of a person with referencing it to the CGM level distribution for standardization of patterns for future analysis. Hence, we cannot consider this feature to be an optimal feature for extraction.



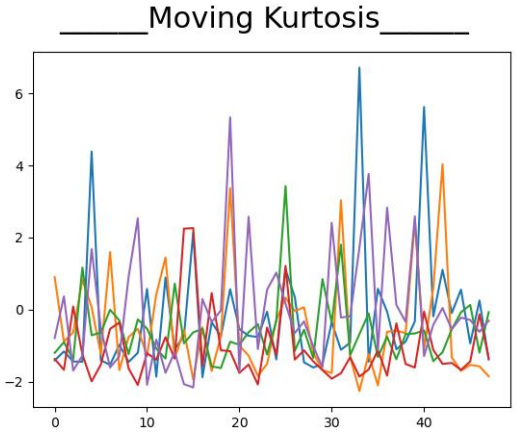
Patient 1



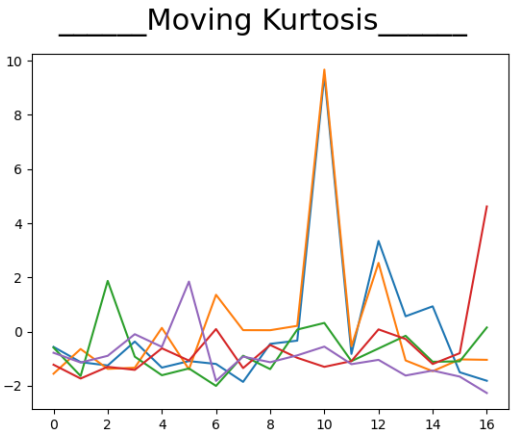
Patient 2



Patient 3



Patient 4

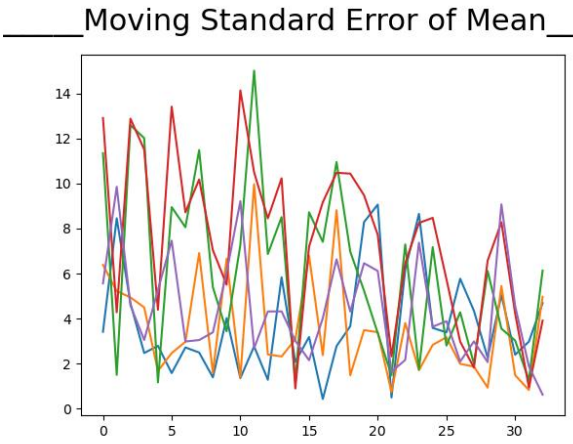


Patient 5

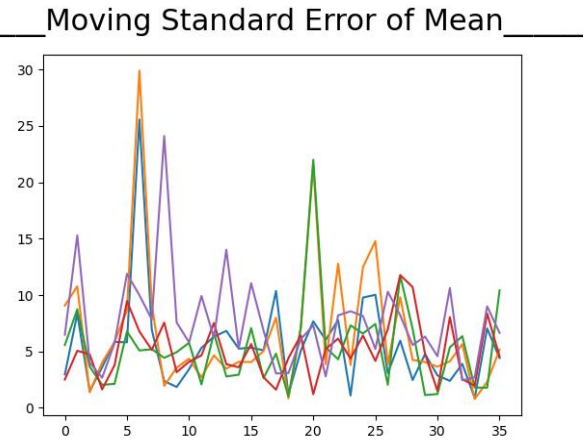
Anish Oswal (aaoswal@asu.edu)

4.3 Moving Standard Error of Mean

The moving standard error of mean gives us the standard deviation of the mean of the data in the graphs. The CGM values in the graphs show peaks at times which represent the patient's meal intake time. The rise in CGM values at peaks specifically point out the time when the patient started having the meal due to such a sudden rise in the values. Hence, this feature can be useful and optimal too because it tells us when a person is starting to have a meal.

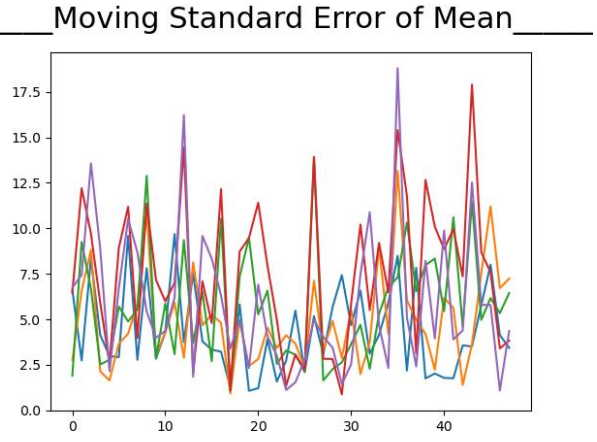


Patient 1



Patient 2

Patient 3



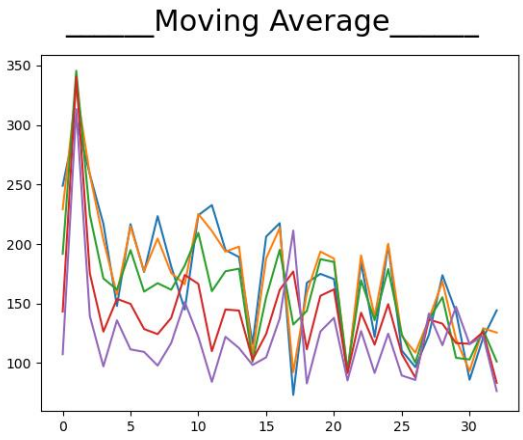
Patient 4

Anish Oswal (aaoswal@asu.edu)

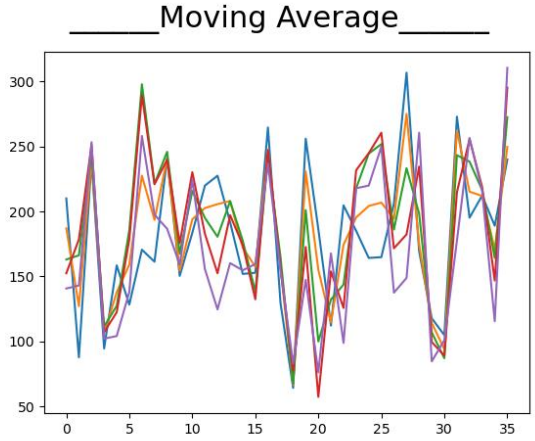
Patient 5

4.4 Moving Average

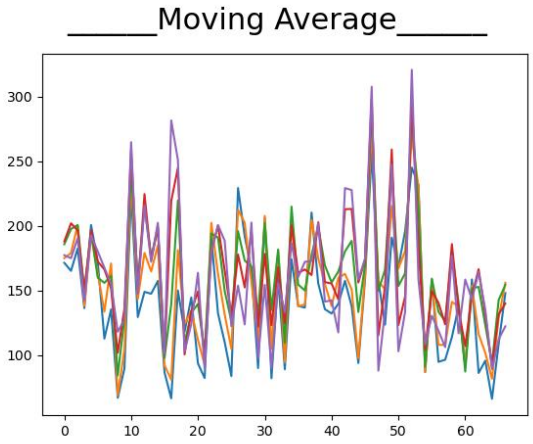
The moving average is plotted for 5 patients. Observing the data, there are peaks where the CGM values are high implying that the particular regions having high CGM values show that the person has had a meal. This helps us understand the time when the person has had a meal which is an important feature. Hence, this feature is useful and can be used to come up with different patterns from the data and mealtime.



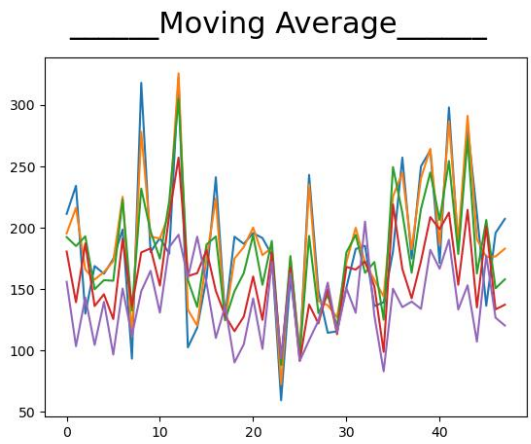
Patient 1



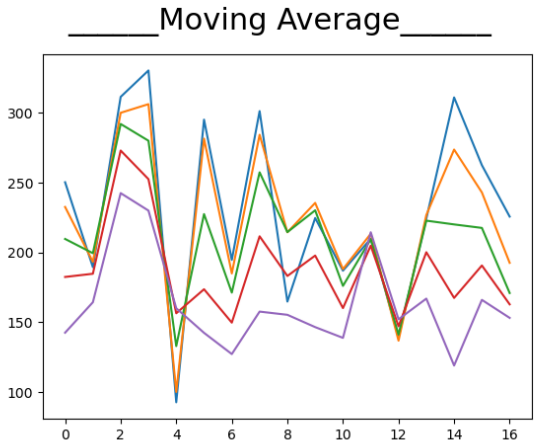
Patient 2



Patient 3



Patient 4



Patient 5

5. Project Phase 3: Feature Selection

5.1. Subtask 1: Arranging the feature matrix

Principal Component Analysis (PCA) takes only one matrix, so, I merged the results obtained in Phase 2 in a single matrix. Hence, the feature matrix will have (No. of patient meals) x 20 features corresponding to each action and the rows corresponding to the timestamps in Phase 2. I was able to find few useful features having higher discrimination power by plotting graphs against every feature for every patient. Hence, I further decided to proceed with selected features to perform PCA on the resulting matrix to find best latent semantics which have the highest discrimination power, even among the ones selected during Phase 2 feature selection process.

5.2. Subtask 2: Execution of PCA

PCA decomposes a correlation matrix into a matrix with Principal Components and the resulting matrix contains the Principal Components in decreasing order of their variance.

We pass the feature matrix obtained in the earlier step to the PCA function of sklearn.decomoposition in Python.

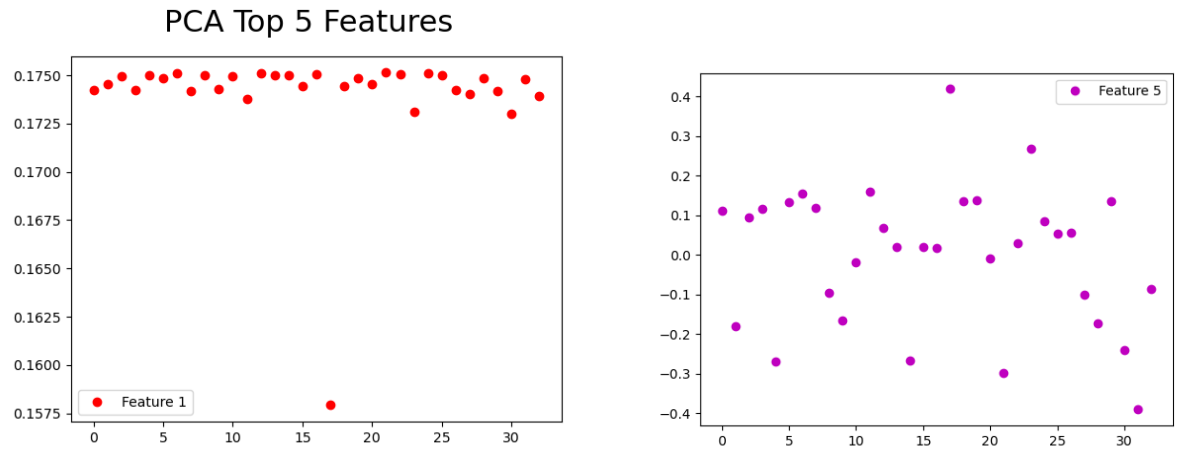
The PCA then tells us the following:
Top Features – Gives us the top features which have one of the highest discrimination powers amongst other features.
Time Series Data – It also gives us the time series data matric which can be fed to a model to perform analysis.

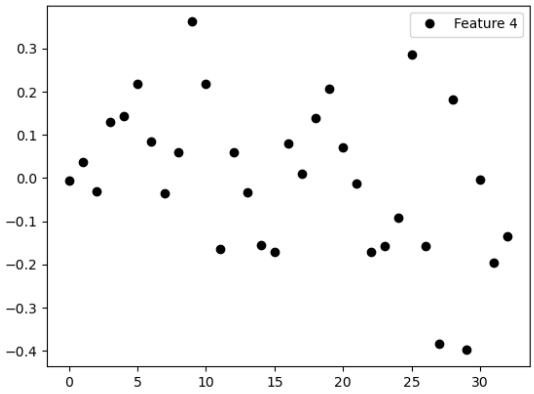
The code for PCA has been included in DM 1.py file.
Now we have the top features with the highest variance having higher discrimination power.

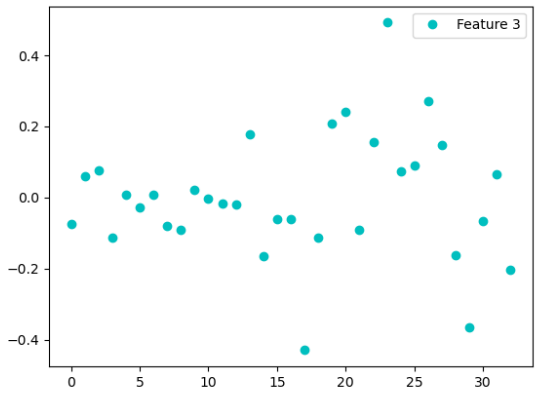
5.3. Subtask 3: Results of PCA

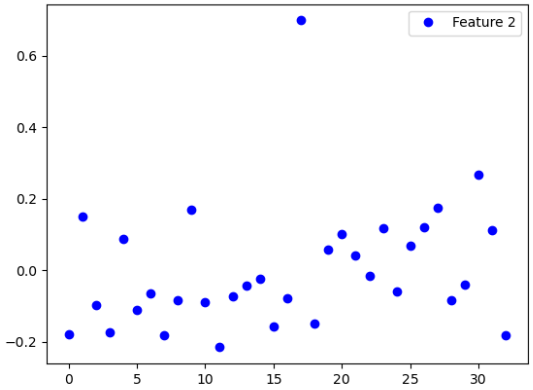
The plotted PCA features for each of the patients can be found below. The below graphs represent each principal component vectors for the data along with which data can be projected individually.

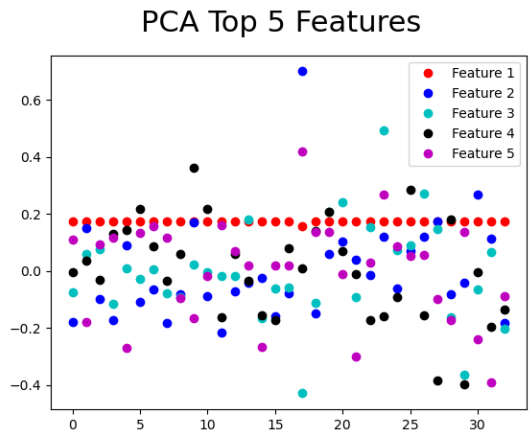
PCA results for the Patient 1:



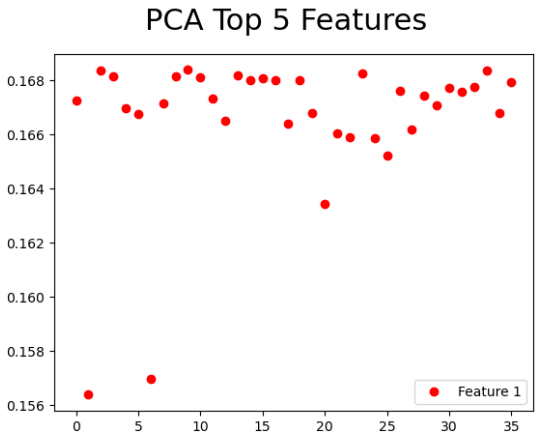


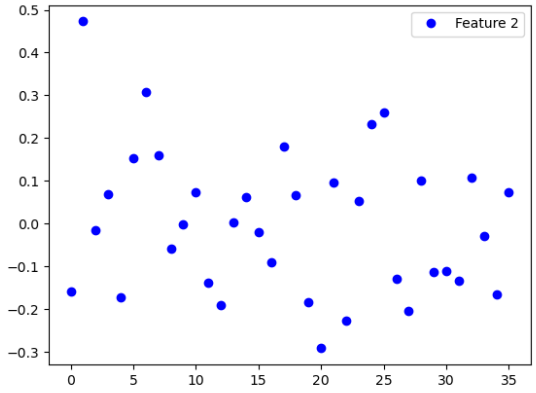




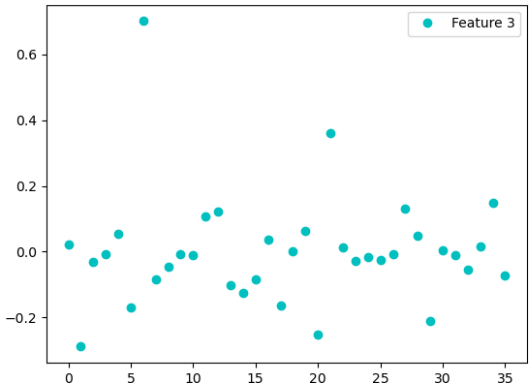


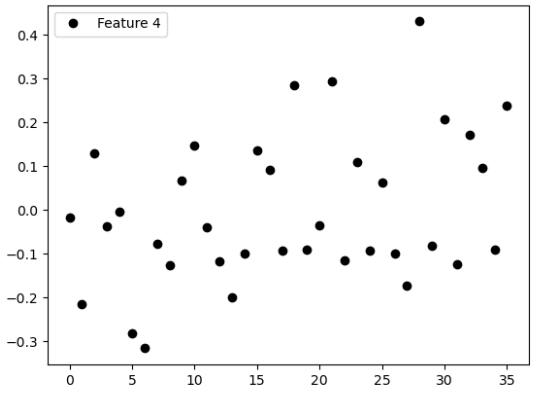
PCA results for the Patient 2:

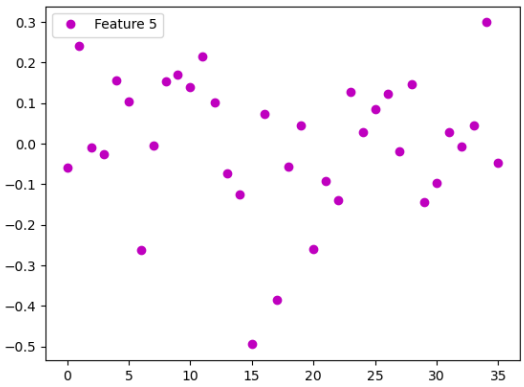


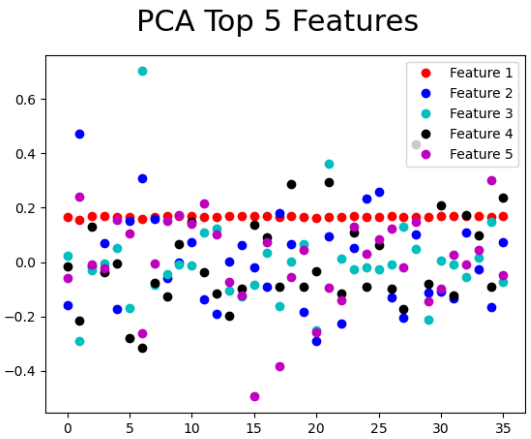


Anish Oswal (aaoswal@asu.edu)



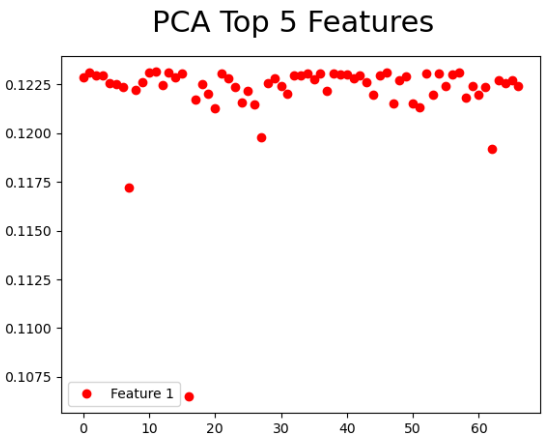


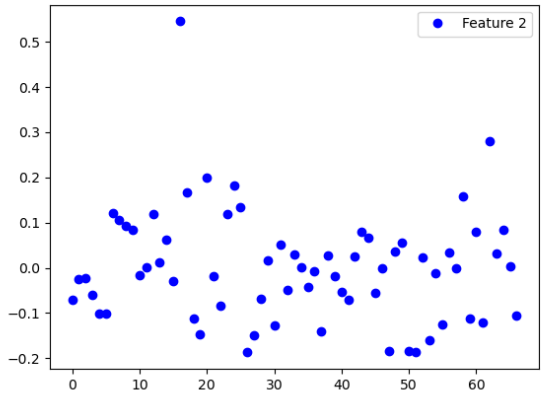


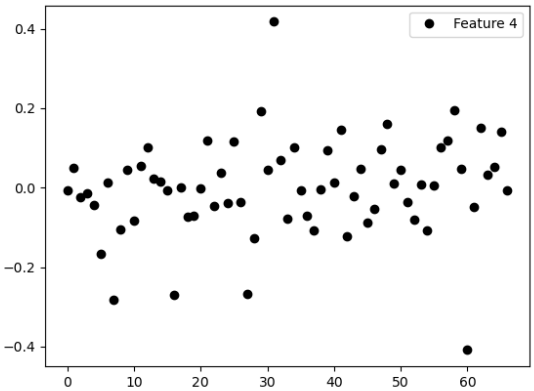


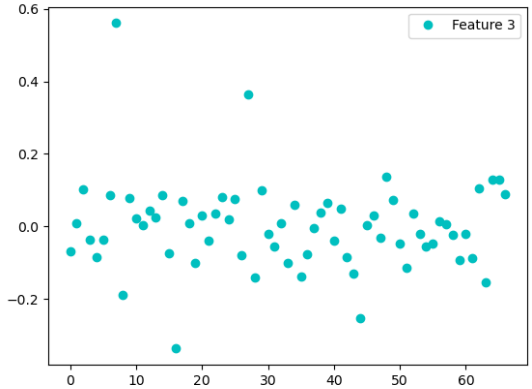
PCA results for the Patient 3:

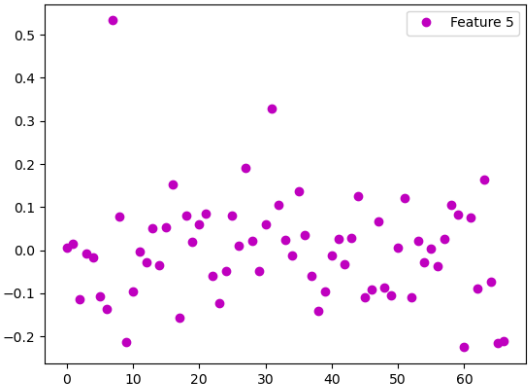
Anish Oswal (aaoswal@asu.edu)

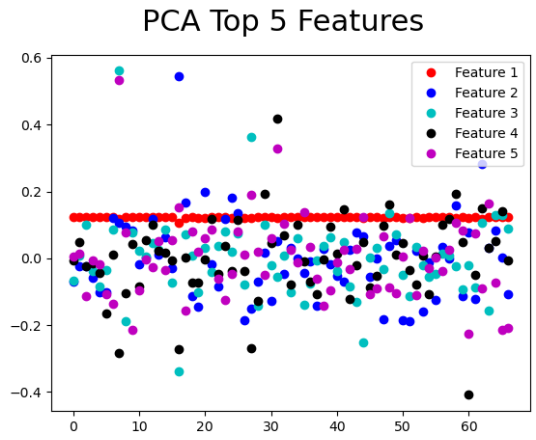






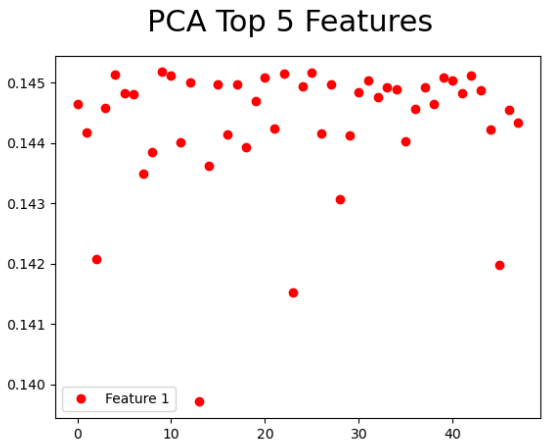


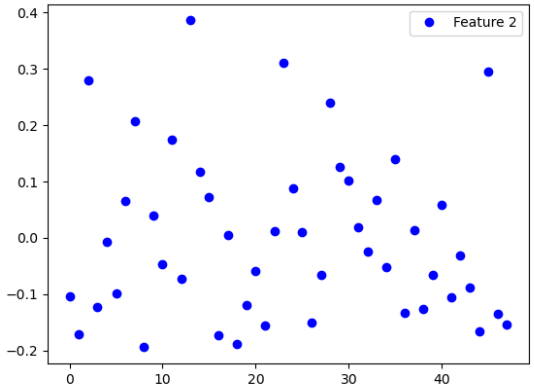


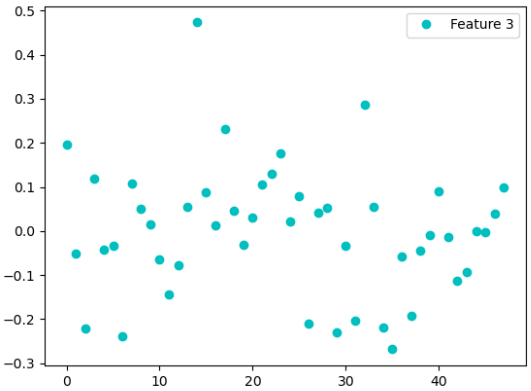


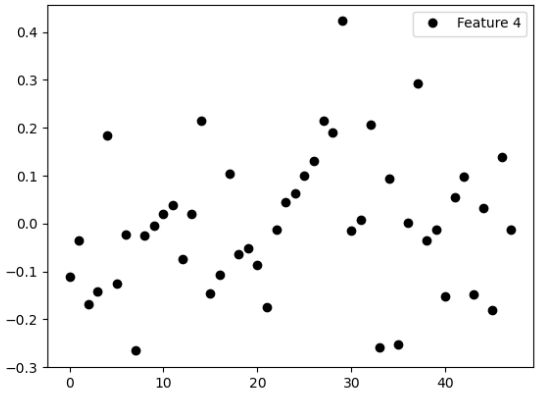
PCA results for the Patient 4:

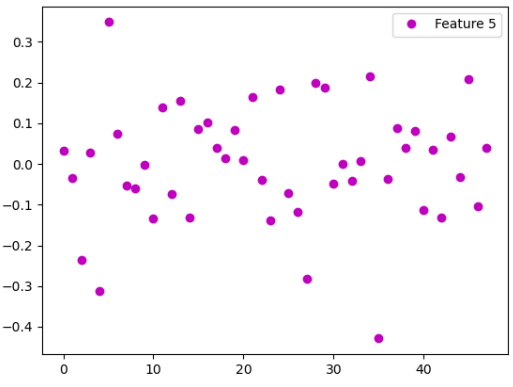
Anish Oswal (aaoswal@asu.edu)

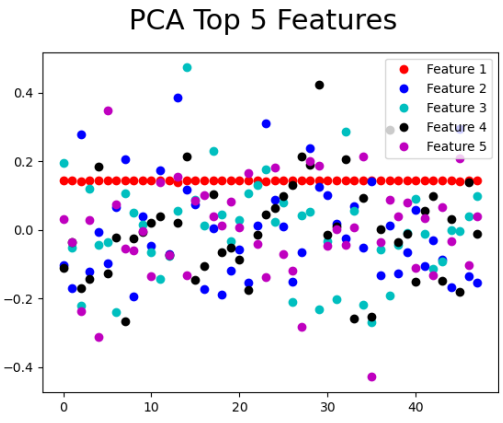






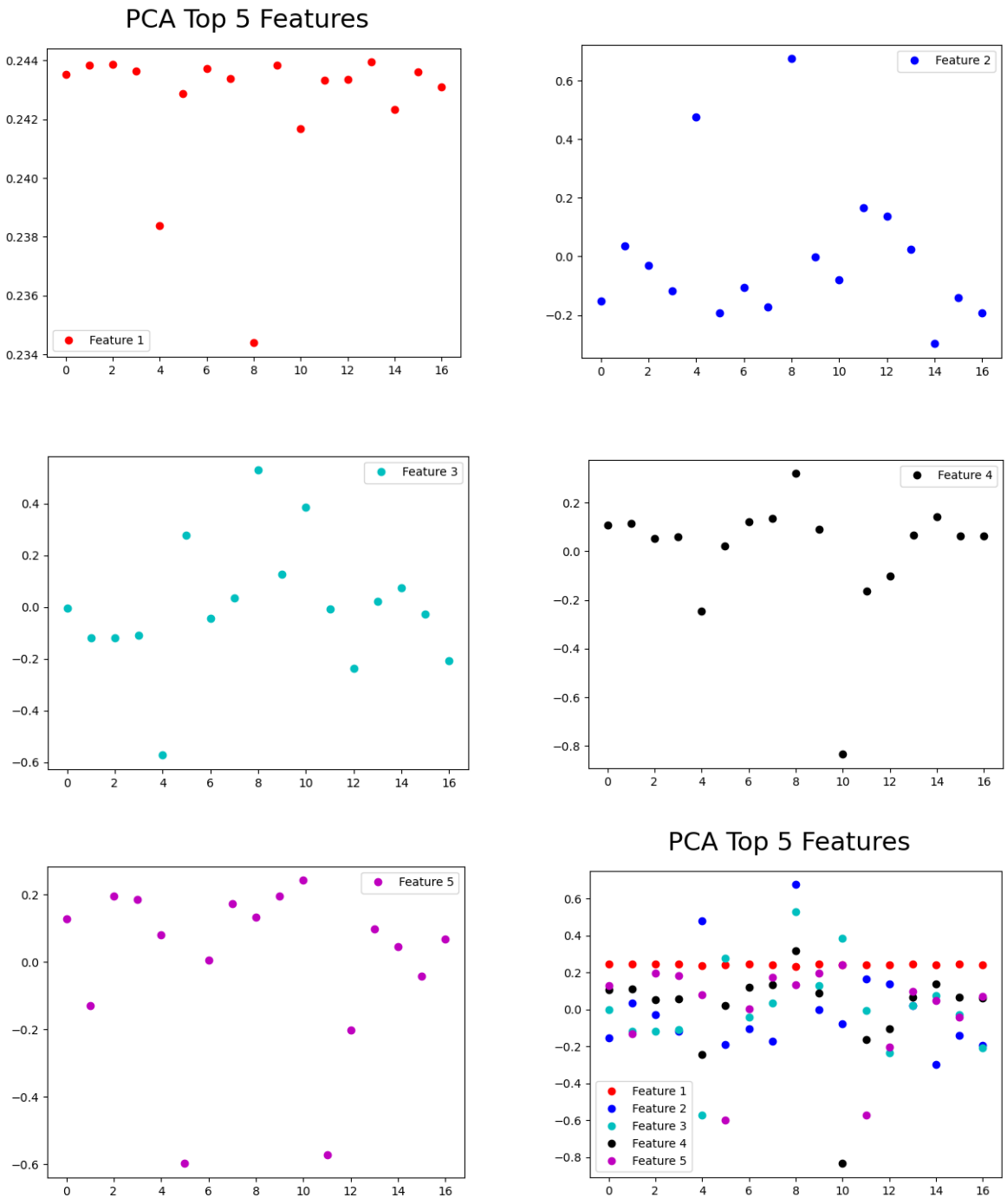






PCA results for the Patient 5:

Anish Oswal (aaoswal@asu.edu)



We can thus conclude that first five eigenvectors can represent more than 90% of the information. Hence, reducing the feature space to 5 or 6 latent features would help in representing data in this transformed space.

5.4. Subtask 5: Reason for the Features in PCA

For each of the above selected features chosen to be the top features among the others is because of the variance which is the measure of how far away each value from the mean of the data is. On observing the graphs, we can see that the information is well represented in the following graphs and hence the dimensionality of the overall information can be reduced to 5 latent features in representing the data. Thus, each of the feature marked as top five has the highest variances compared to others. Hence, the reason for choosing the features in top 5 in PCA are:
Feature 1 – Highest Variance among others.
Feature 2 – Second Highest Variance among others.
Feature 3 – Third Highest Variance among other features.
Feature 4 – Fourth Highest Variance among other features.
Feature 5 – Fifth Highest Variance among other features.

Therefore, as these features show maximum variance, they are selected by the features as top features.

References:

1. https://machinelearningmastery.com/moving-average-smoothing-for-time-series-forecasting-python/
2. https://stats.stackexchange.com/questions/27300/using-principal-component-analysis-pca-for-feature-selection
3. https://machinelearningmastery.com/basic-feature-engineering-time-series-data-python/
4. https://docs.scipy.org/doc/scipy/reference/stats.html