CSE 572: Data Mining

Project Phase 1 Report

Submitted by Arun Raj, Deva (asu id: 1218410609)

Submitted to:

**Professor Ayan Banerjee**

Ira A. Fulton School of Engineering

Arizona State University

## 1. Introduction

The objective of this phase of project is to identify the patterns in glucose level and identify the meal intake. We are using CGM (Continuous Glucose Monitor) to monitor the glucose levels in a patient. In this phase of project we are using CGMDatenumLunchPat and CGMSeriesLunchPat data. CGMDatenumLunchPat is data for time stamps and CGMSeriesLunchPat is monitored glucose values in a patient for respective patient.

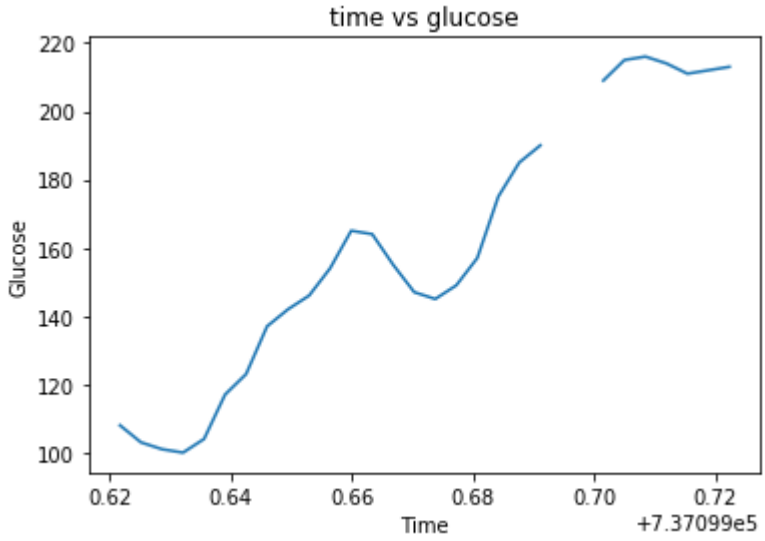
## 2. Project Phase 1

This phase of project has 6 tasks as data preprocessing, features extraction, justifying the selection of that particular feature, generating feature matrix, PCA (principal component analysis) implementation and plotting PCA matrix after dimension reduction. Each task is explained in detail in below sections.

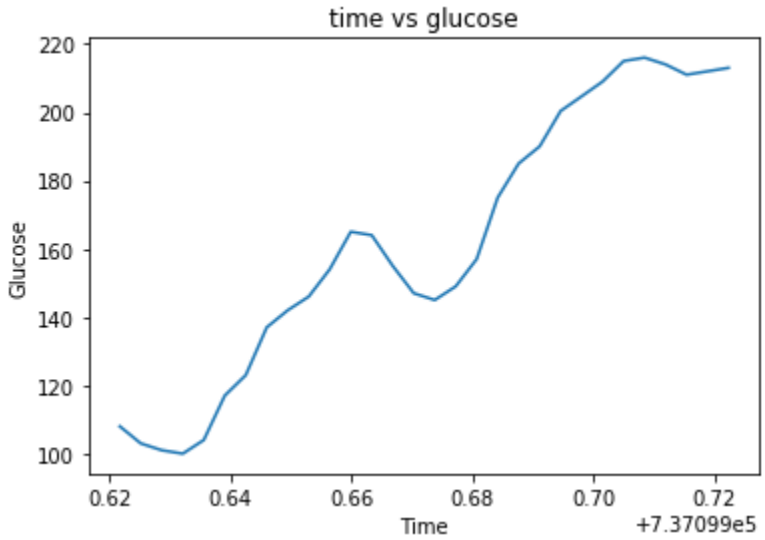
## 3. Data Pre-processing

Data preprocessing is an early task in the project, this task is mainly used to fill the NaN values from in the data set provided. Given data set is of 5 patients which has CGM values, continuous glucose monitoring values of the patient. This csv file consists of glucose levels at that particular time and these timestamps are provided in another csv file. The data provided is for every 5 minutes for 2.5 hours of continuous monitoring, which gives in total of 30 – 31 values for each patient meal. Few rows have very less number of data points and most of these data points are NaN. These data rows with most of predicted values will not give any useful information. So, if any meal has less than 26 data points I deleted the meal from the provided data set. Even after

removing these data rows, there are still a few data points where value is not recorded or the value provided at that timestamp is null. So, we have to predict the value at those timestamps and place it in those null values. To achieve this I used a local mean logic. For example: If a row 1 column 4 has null value, I took the average of column 2, column 3, column 5 and column 6 values. There are other ways to fill these NaN values in python like polyfit. I added few conditions to handle all exception cases.




Before data pre - processing




After data preprocessing

## 4. Feature Extraction (Task a, b, c)

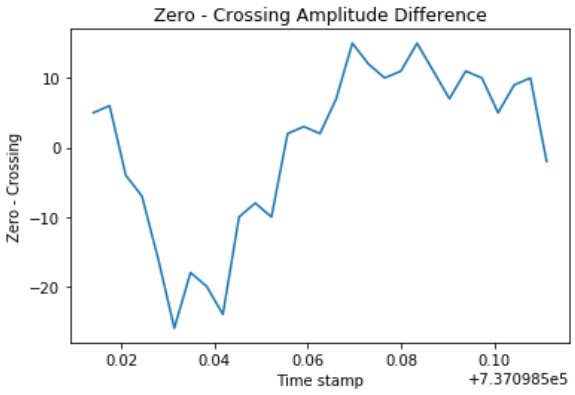
Feature extraction can be applied after the data pre-processing. Feature extraction is nothing but we try to make some meaningful assumptions from the given data set to infer something else which was not provided directly. The feature extraction techniques are applied to the given timestamp or glucose level data sets. For extracting these features and generate the feature matrix, I used 8 feature extraction techniques as listed below:

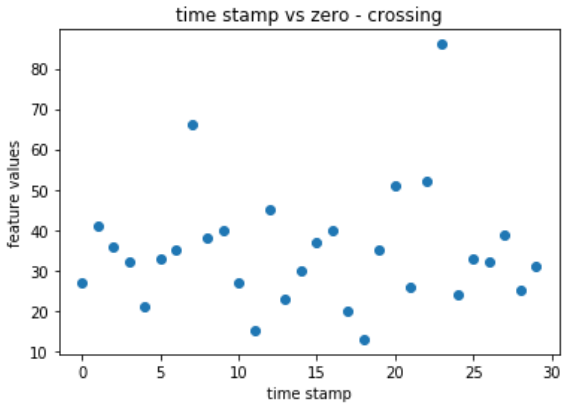
1. Zero – Crossing
2. Velocity
3. Minimum
4. Maximum
5. Mean
6. FFT - Fast Fourier Transform
7. Entropy
8. RMS – Root mean Square

Above listed techniques are used to identify the patterns. All the below features are extracted only for patient 2. So, 30 data rows are considered, other patients data can be generated similarly.

### 4.1 Zero – Crossing

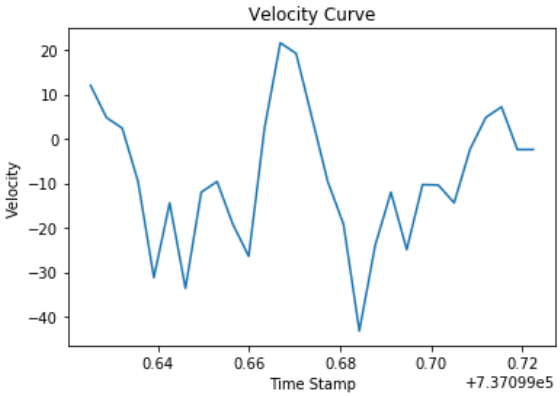
In this feature, I took the difference of two consecutive glucose values. If the glucose level increases from that timestamp to next timestamp then, the difference calculated will be a positive value. If the glucose value is decreasing then, the difference will be negative. In this I am taking the difference between the maximum and minimum value, which gives the relation between the minimum and maximum difference. This feature helps in finding the difference of glucose levels in with and without meal cases, so in my opinion this is one of the necessary feature.

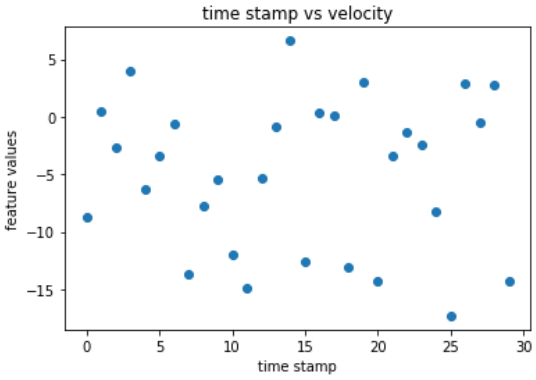


## 4.2 Velocity or Slope

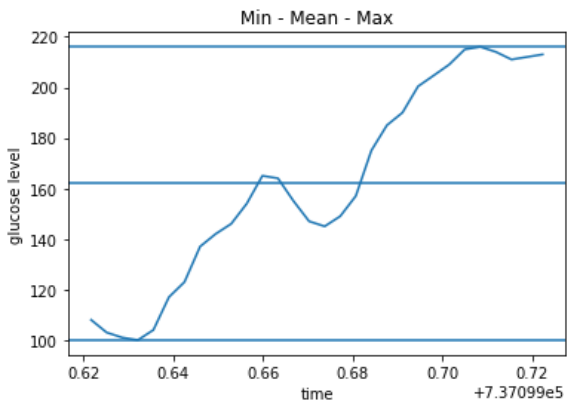
In this feature, we calculated, the velocity, means difference of glucose levels in the time period. For example: Let's consider two points 1 and 2 now the velocity will be G2 – G1 / T2 – T1, which is basically the slope formula, y axis difference upon the x axis difference. This feature will let me know if the glucose increase or decrease rate in the patient's body. We already know that as slope increases the y axis value also increases, in this case the glucose level. Just from the velocity value we can say the rate of change in glucose. While the velocity in positive side of the curve it means the glucose level is increasing and if the velocity is negative it means the glucose level is decreasing as shown below, this features give further insight on the patients glucose level rate, so I considered this feature as important.

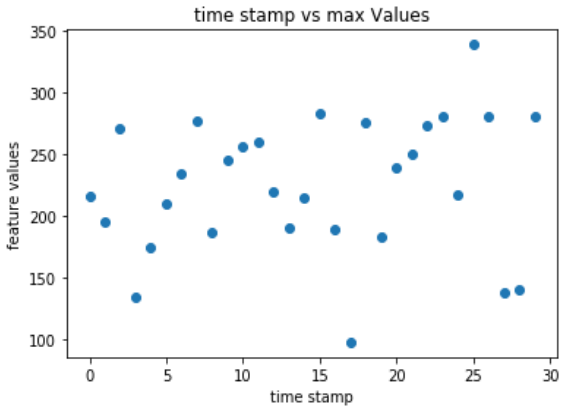


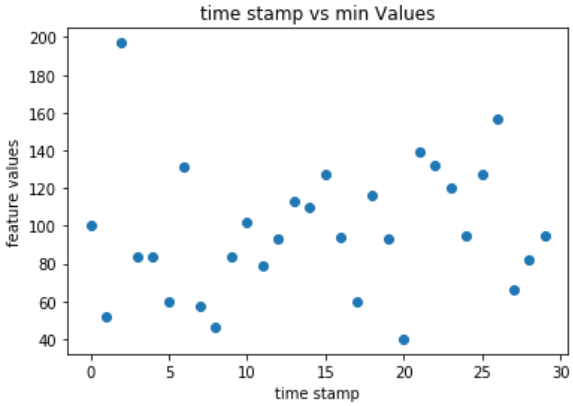



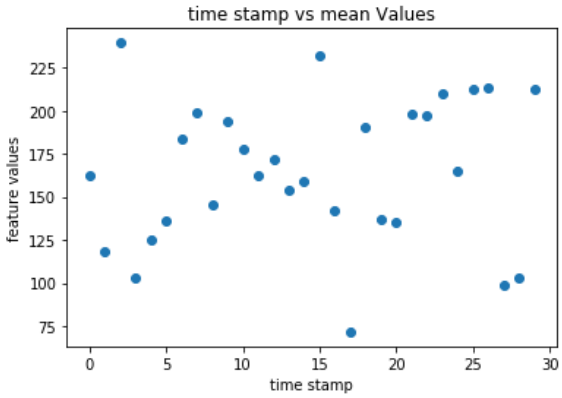

## 4.3 – 4.5 Minimum, Maximum and Mean

These features will give the minimum, maximum and mean values for the glucose level in the duration of 2.5 hours. These values will give an insight about what the minimum glucose value present in the patient's body along with the maximum and mean value. With these values we can state whether the patient had any meal or not. If the glucose level is above the mean and near to maximum value then we can say patient had meal little early and if the glucose level is less then we can say it's been a long time since a last meal. In this below, figure the three horizontal lines represent the minimum, maximum and mean value in total data row. This feature gives the values for minimum level, maximum level and average level of glucose in the patient's body. Depending on current level we can say the status of meal taken or not. So, I considered this feature to give overall view on glucose levels.



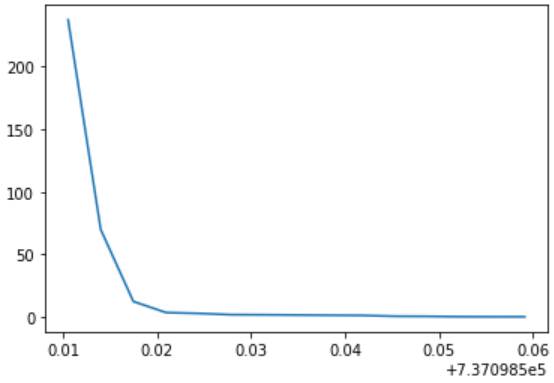



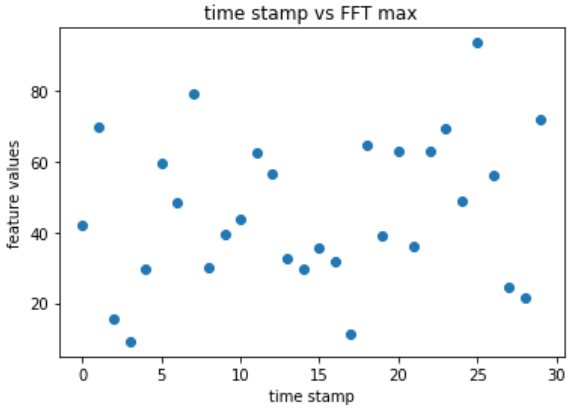



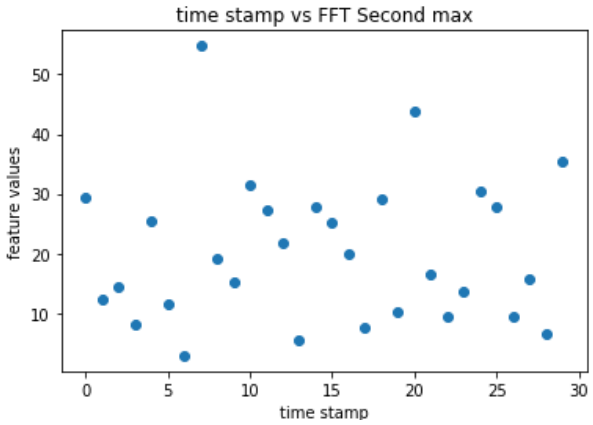




## 4.6 Fast Fourier Transform

Fast Fourier Transform converts the signal from one domain to frequency domain. We can generate significant components from the fourier transform. From these significant components we can reconstruct the important features of original data. In this feature I am considering maximum amplitude and second highest amplitude as the features. We can also consider the lower frequency as a new feature. We are not going to consider the first highest amplitude. One of main advantage of FFT is it converts discrete data to continuous data, which can help in understand the glucose level for long interval.

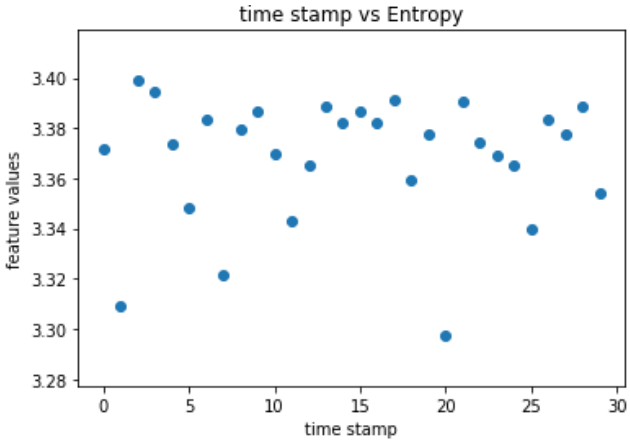




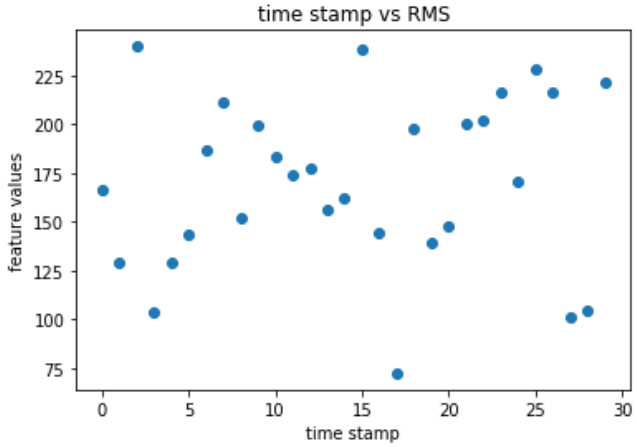

## 4.7 Entropy

Entropy often interpreted as the degree of disorder or randomness in the system is the dictionary definition of Entropy. This will help to infer the received glucose levels and judge the received values using entropy feature. Using this entropy we can construct the decision trees and make decisions based on the values. Decision tree is the important part in gaining the information. Decision trees are mostly used in classification technique.



## 4.8 Root Mean Square

Root mean square is square root of mean of squared values. This metric is used to calculate the magnitude of data. This magnitude will help in analyzing the data. So, I considered RMS as one of the important feature.




## Conclusion on Intuition (Task c):

Basically, my intuition was the features extracted in the previous section will help to make a better model or a feature matrix which in turn gives further insight on the data model where analyzing the data set becomes simple based on extraction of proper important features. Just as from what I expected my intuition is correct each of these features gave some other important data about the given data set like velocity feature helps to understand at what rate the glucose level is changing, entropy feature gives the order of disorderness which helps in making a better decision tree which can be used with classification technique, fast fourier transform was helpful in analyzing the data in continuous way and in frequency domain rather than time domain, minimum, maximum mean values helped to understand the current state of patients glucose level etc. So, each extracted feature helped in analyzing the data set further.

## 5. Feature Matrix (Task d)

Feature matrix is combination of all features extracted in previous section. These features will categorize the different data objects. As name suggests this will be a matrix that is generated based on number of features that are extracted. In my case, I extracted 9 different types of features and the features were extracted for patient 2 who had 30 valid data sets. So, the feature matrix created will have 30 rows of data and 9 columns which denotes the extracted features. This entire matrix formed is called as a feature matrix. To implement PCA we need to use feature matrix for dimensionality reduction. Each column in the matrix represents the respective feature values. Below, is the generated feature matrix in my execution of input data set.

| | Feat_Zero( | Feat_Veloc | Feat_Max | Feat_Min | Feat_Mear | Feat_FFT1 | Feat_FFT2 | Feat_Entrc | Feat_RMS |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 27 | 14.15 | 216 | 100 | 162.07 | 42.09 | 29.37 | 3.371373 | 166.72 |
| 1 | 41 | 24.5 | 195 | 52 | 118.63 | 69.81 | 12.4 | 3.309034 | 128.86 |
| 2 | 36 | 12.75 | 271 | 197 | 239.4 | 15.4 | 14.49 | 3.398795 | 239.97 |
| 3 | 32 | 13.9 | 134 | 84 | 103.37 | 9.17 | 8 | 3.394232 | 104.09 |
| 4 | 21 | 11.92 | 175 | 84 | 125.47 | 29.68 | 25.51 | 3.373926 | 128.86 |
| 5 | 33 | 21.44 | 210 | 60 | 136.43 | 59.69 | 11.61 | 3.347925 | 143.28 |
| 6 | 35 | 16.88 | 235 | 131 | 183.57 | 48.54 | 2.91 | 3.383404 | 186.79 |
| 7 | 66 | 26.32 | 277 | 57 | 198.47 | 79.19 | 54.75 | 3.321518 | 211.07 |
| 8 | 38 | 16.3 | 187 | 46 | 145.52 | 30.24 | 19.27 | 3.37941 | 151.68 |
| 9 | 40 | 19.37 | 245 | 84 | 193.58 | 39.38 | 15.19 | 3.38698 | 199.87 |
| 10 | 27 | 14.98 | 256 | 102 | 177.97 | 43.78 | 31.46 | 3.369753 | 183.16 |
| 11 | 15 | 15.06 | 260 | 79 | 162.35 | 62.7 | 27.38 | 3.343302 | 174.14 |
| 12 | 45 | 17.05 | 220 | 93 | 171.6 | 56.43 | 21.76 | 3.365175 | 177.15 |
| 13 | 23 | 12.66 | 191 | 113 | 154.07 | 32.67 | 5.42 | 3.388881 | 155.91 |
| 14 | 30 | 17.66 | 215 | 110 | 159.21 | 29.51 | 27.87 | 3.382222 | 162.23 |
| 15 | 37 | 18.04 | 283 | 127 | 231.58 | 35.82 | 25.1 | 3.386601 | 238.18 |
| 16 | 40 | 17.88 | 189 | 94 | 141.83 | 31.77 | 19.84 | 3.382198 | 144.42 |
| 17 | 20 | 8.44 | 98 | 60 | 71.7 | 11.27 | 7.48 | 3.391113 | 72.45 |
| 18 | 13 | 13.41 | 276 | 116 | 190.03 | 64.61 | 29.12 | 3.359041 | 197.8 |
| 19 | 35 | 13.99 | 183 | 93 | 136.53 | 38.88 | 10.35 | 3.377245 | 139.72 |
| 20 | 51 | 22.76 | 239 | 40 | 135.03 | 63.13 | 43.78 | 3.297574 | 147.57 |
| 21 | 26 | 15.31 | 250 | 139 | 197.93 | 36.17 | 16.48 | 3.390573 | 200.02 |
| 22 | 52 | 25 | 274 | 132 | 196.97 | 63.06 | 9.44 | 3.374164 | 202.15 |
| 23 | 86 | 30.87 | 281 | 120 | 210.07 | 69.48 | 13.58 | 3.369347 | 216.39 |
| 24 | 24 | 14.48 | 217 | 95 | 165.1 | 48.92 | 30.37 | 3.365179 | 170.57 |
| 25 | 33 | 19.45 | 339 | 127 | 212.71 | 93.62 | 27.87 | 3.339687 | 228.23 |
| 26 | 32 | 17.79 | 281 | 157 | 212.93 | 56.19 | 9.38 | 3.383157 | 216.77 |
| 27 | 39 | 13.9 | 138 | 66 | 98.52 | 24.3 | 15.69 | 3.377474 | 101.07 |
| 28 | 25 | 9.35 | 141 | 82 | 103.1 | 21.33 | 6.67 | 3.388362 | 104.45 |
| 29 | 31 | 16.47 | 281 | 95 | 212.6 | 71.84 | 35.44 | 3.353914 | 221.53 |

## 6. PCA – Principal Component Analysis (Task e)

PCA is short form for Principal Component Analysis. It is used for dimension reduction. PCA is one of the solution of curse of dimensionality. With dimension reduction we can avoid curse of dimensionality, reduce time and memory requirements to execute an algorithm, reduces noise and mainly for easy human visualization. The PCA algorithm takes the feature matrix as an input and gives a new feature matrix after completion of the dimension reduction. To complete the PCA process following steps were performed.

- In the first step a re-scaling is applied on the feature matrix, this is not a necessary step but re-scaling is performed to normalize the values and attributes which rely on magnitude of the feature. This step was performed using sklearn.preprocessing package in python.
- After completion of the re-scaling we will find the co-variance matrix of the scaled matrix. This step is performed using numpy package in python.
- In this step, we will find the values for eigen vectors and eigen values of covariance matrix, this new eigen vectors will define the new space of the features and eigen values are constants. This step is executed using the same numpy package it has method called np.linalg.eig(matrix)
- With the generated eigen vectors and eigen values PCA matrix is formed with the reduction of dimensions as requirement. This step uses sklearn.decomposition package in python.

All these above mentioned steps can also be performed directly using sklearn package with pca.fit_transform and sklearn.decomposition.PCA methods.

covarianceMatrix - NumPy array

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1.03448 | 0.866113 | 0.28086 | -0.0599666 | 0.27278 | 0.369314 | 0.135259 | -0.28272 | 0.287942 |
| 1 | 0.866113 | 1.03448 | 0.535153 | -0.0842864 | 0.390826 | 0.699377 | 0.252303 | -0.56379 | 0.430603 |
| 2 | 0.28086 | 0.535153 | 1.03448 | 0.534659 | 0.949762 | 0.748677 | 0.437578 | -0.314113 | 0.976614 |
| 3 | -0.0599666 | -0.0842864 | 0.534659 | 1.03448 | 0.729684 | -0.0444403 | -0.271097 | 0.521514 | 0.679644 |
| 4 | 0.27278 | 0.390826 | 0.949762 | 0.729684 | 1.03448 | 0.477959 | 0.274113 | 0.0248548 | 1.03075 |
| 5 | 0.369314 | 0.699377 | 0.748677 | -0.0444403 | 0.477959 | 1.03448 | 0.47482 | -0.77517 | 0.542839 |
| 6 | 0.135259 | 0.252303 | 0.437578 | -0.271097 | 0.274113 | 0.47482 | 1.03448 | -0.624116 | 0.325378 |
| 7 | -0.28272 | -0.56379 | -0.314113 | 0.521514 | 0.0248548 | -0.77517 | -0.624116 | 1.03448 | -0.0548742 |
| 8 | 0.287942 | 0.430603 | 0.976614 | 0.679644 | 1.03075 | 0.542839 | 0.325378 | -0.0548742 | 1.03448 |

Co-Variance matrix

eigenValues - NumPy array

| | 0 |
|---|---|
| 0 | 4.58162 |
| 1 | 2.61472 |
| 2 | 1.2493 |
| 3 | 0.551237 |
| 4 | 0.145352 |
| 5 | 0.0875971 |
| 6 | 0.061487 |
| 7 | 0.0188226 |
| 8 | 0.000205169 |

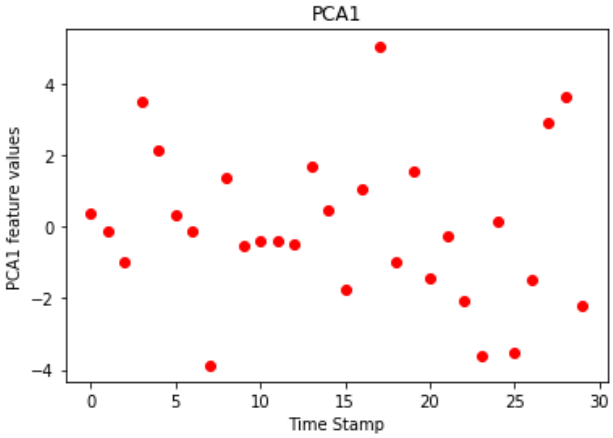
eigenVectors - NumPy array

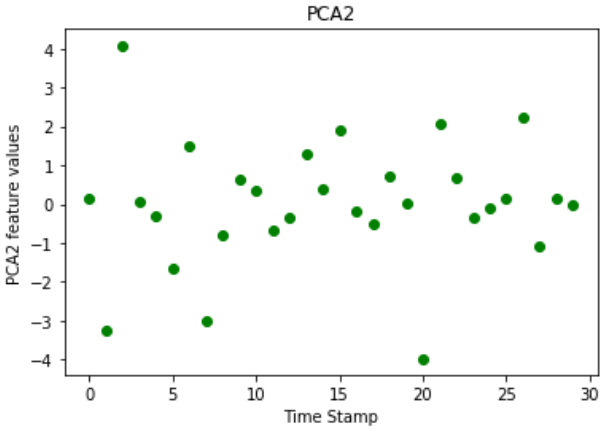
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | -0.249193 | -0.197255 | -0.659772 | -0.378342 | 0.180953 | 0.403629 | -0.311241 | -0.167198 | -0.00804595 |
| 1 | -0.347564 | -0.257796 | -0.458716 | 0.0506129 | -0.0762916 | -0.515164 | 0.54532 | 0.175467 | 0.0195309 |
| 2 | -0.449245 | 0.147858 | 0.155221 | 0.110382 | -0.0864051 | -0.117564 | 0.0546092 | -0.842352 | -0.0788183 |
| 3 | -0.162521 | 0.558511 | -0.0587619 | 0.112689 | 0.750946 | 0.0827234 | 0.24621 | 0.112445 | 0.0383988 |
| 4 | -0.401298 | 0.321964 | 0.0472435 | -0.120103 | -0.222875 | -0.101607 | -0.267082 | 0.347159 | -0.683223 |
| 5 | -0.385198 | -0.237543 | 0.120746 | 0.515347 | -0.145258 | 0.639612 | 0.217681 | 0.193493 | -0.00244723 |
| 6 | -0.236934 | -0.267937 | 0.506938 | -0.682895 | 0.160417 | 0.117612 | 0.322695 | 0.07057 | 0.00789268 |
| 7 | 0.220513 | 0.499839 | -0.218712 | -0.272078 | -0.484588 | 0.32699 | 0.481789 | -0.0726632 | 0.0229183 |
| 8 | -0.417955 | 0.281281 | 0.071424 | -0.0950416 | -0.242317 | -0.104128 | -0.295269 | 0.225468 | 0.72421 |

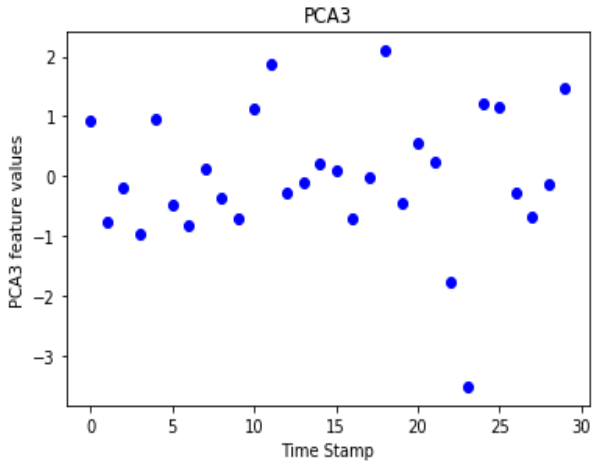
Eigen values and Eigen vectors

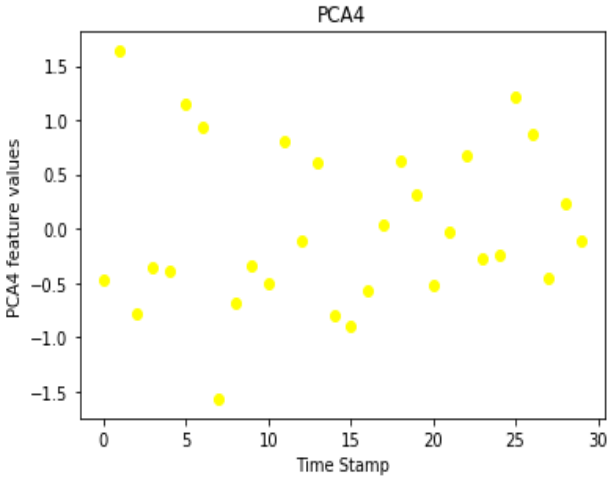
## PCA Matrix with 5 features:

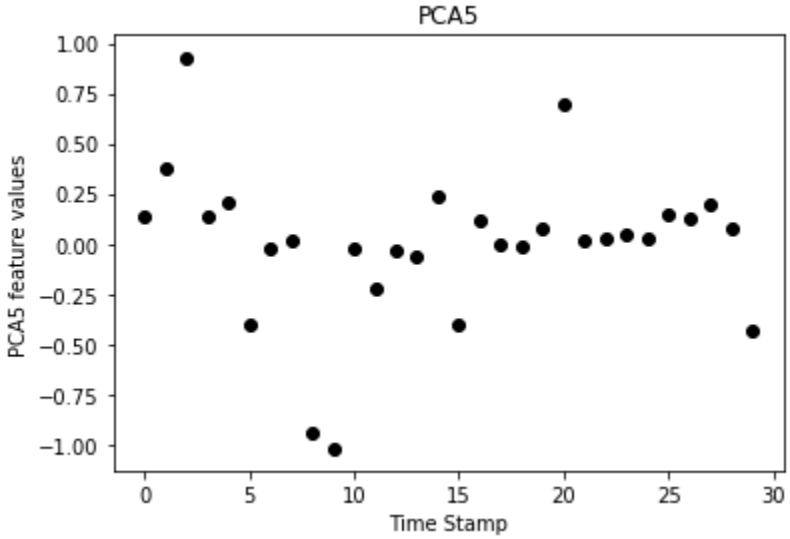
| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 0.393126 | 0.125547 | 0.942112 | -0.46705 | 0.134411 |
| 1 | -0.12439 | -3.23029 | -0.7612 | 1.63789 | 0.377581 |
| 2 | -0.99754 | 4.08498 | -0.17345 | -0.78024 | 0.928955 |
| 3 | 3.50087 | 0.046926 | -0.95076 | -0.35386 | 0.139978 |
| 4 | 2.12208 | -0.29794 | 0.968683 | -0.395 | 0.20212 |
| 5 | 0.30754 | -1.63444 | -0.4759 | 1.15353 | -0.40481 |
| 6 | -0.12782 | 1.4971 | -0.81935 | 0.938913 | -0.02629 |
| 7 | -3.86655 | -3.01825 | 0.138043 | -1.57068 | 0.012589 |
| 8 | 1.34931 | -0.79614 | -0.36704 | -0.69076 | -0.93783 |
| 9 | -0.51615 | 0.622034 | -0.69671 | -0.33386 | -1.01633 |
| 10 | -0.40798 | 0.354997 | 1.13526 | -0.51289 | -0.02788 |
| 11 | -0.4002 | -0.69113 | 1.86451 | 0.798 | -0.22368 |
| 12 | -0.4852 | -0.35155 | -0.28464 | -0.10951 | -0.03666 |
| 13 | 1.70635 | 1.27427 | -0.09282 | 0.613354 | -0.06624 |
| 14 | 0.487192 | 0.408426 | 0.213766 | -0.80721 | 0.233802 |
| 15 | -1.76412 | 1.89136 | 0.112606 | -0.89969 | -0.40322 |
| 16 | 1.05006 | -0.16734 | -0.68976 | -0.56363 | 0.115339 |
| 17 | 5.06185 | -0.521 | -0.02769 | 0.02956 | -0.00741 |
| 18 | -0.98394 | 0.696639 | 2.10835 | 0.629266 | -0.01789 |
| 19 | 1.57501 | 0.032451 | -0.44929 | 0.316162 | 0.074491 |
| 20 | -1.42819 | -3.96749 | 0.54627 | -0.52838 | 0.700134 |
| 21 | -0.2683 | 2.05172 | 0.249906 | -0.02472 | 0.01508 |
| 22 | -2.08445 | 0.665402 | -1.75405 | 0.675134 | 0.03041 |
| 23 | -3.59582 | -0.35302 | -3.51578 | -0.27438 | 0.046113 |
| 24 | 0.167774 | -0.10776 | 1.20725 | -0.23588 | 0.029996 |
| 25 | -3.5331 | 0.136265 | 1.15916 | 1.2085 | 0.146892 |
| 26 | -1.49415 | 2.22423 | -0.27296 | 0.875823 | 0.131702 |
| 27 | 2.92298 | -1.07621 | -0.67229 | -0.44874 | 0.196816 |
| 28 | 3.64055 | 0.13862 | -0.11768 | 0.234704 | 0.07924 |
| 29 | -2.20682 | -0.03841 | 1.47546 | -0.11435 | -0.42741 |

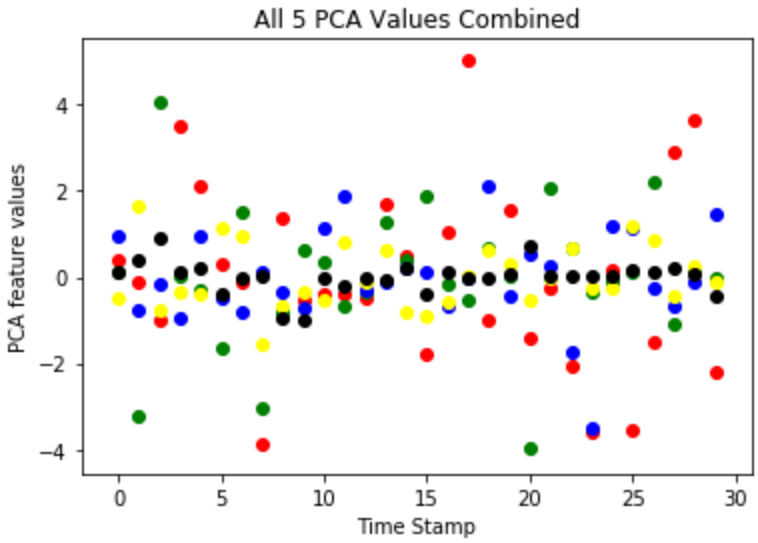

PCA1
PCA1 feature values
Time Stamp


PCA2
PCA2 feature values
Time Stamp


PCA3
PCA3 feature values
Time Stamp


PCA4
PCA4 feature values
Time Stamp


PCA5
PCA5 feature values
Time Stamp




The above plots represent the particular feature values after application of PCA, the last plot is combination of all the 5 features. These all features are calculated based on the eigen values, higher the eigen value of particular feature represents the higher important feature from all the features extracted. All above plots are done considering only the patient 2.

| Index | PCA | Eigen value | Variance | Feature |
|---|---|---|---|---|
| 0 | PCA1 | 4.58 | 49.19 | 8 |
| 1 | PCA2 | 2.61 | 28.03 | 4 |
| 2 | PCA3 | 1.25 | 13.42 | 7 |
| 3 | PCA4 | 0.55 | 5.90 | 6 |
| 4 | PCA5 | 0.15 | 1.4 | 1 |

## 7. Conclusion (Task f)

The top 5 features are given in the above tabular column. We took these 5 features as top 5 features based on the eigen values and variance. We implemented PCA on the feature matrix which was created in above sections and PCA is a dimension reduction mechanism where the important parts of data is not lost.

Entropy – 49.19%, Minimum – 28.03 %, FFT 2 – 13.42%, FFT 1 – 5.90%, Zero crossing – 1.4% are the variance ratio percentages which gave the top 5 features from the feature matrix. More than 97% of important data is from these 5 features. So, we can say that these are important features.

## 8. Bibliography

Most of the definitions of the features, concepts or steps involved in project and python code implementation was referred from many different blogs and websites, few of the references are listed below:


https://www.techopedia.com/definition/14650/data-preprocessing
https://en.wikipedia.org/wiki/Feature_extraction
https://docs.oracle.com/cd/B28359_01/datamine.111/b28129/feature_extr.htm#i1005920
http://ataspinar.com/2018/04/04/machine-learning-with-signal-processing-techniques/
https://en.wikipedia.org/wiki/Entropy
https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8
http://www.analytictech.com/mb313/rootmean.htm
https://numpy.org/doc/1.18/reference/routines.fft.html
https://numpy.org/doc/1.18/reference/generated/numpy.fft.fft.html#numpy.fft.fft
https://stackoverflow.com/questions/25735153/plotting-a-fast-fourier-transform-in-python
https://stackoverflow.com/questions/15450192/fastest-way-to-compute-entropy-in-python
https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.entropy.html
https://github.com/minsuk-heo/python_tutorial/blob/master/data_science/pca/PCA.ipynb