Assignment 1 Report
CSE 572: Data Mining
Spring 2020

Submitted to:

Dr. Ayan Banerjee

Ira. A Fulton Schools of Engineering

Arizona State University

Introduction

The ‘Meal Detection’ project is a part of the course requirement for Data Mining (CSE 572) for the session of Spring 2020 at Arizona State University. The goal of the project is to attempt to develop a computing system that can understand the variations in glucose levels corresponding to a meal intake and inject that much amount of insulin (mimicking the role of a pancreas) for maintaining blood sugar levels in a patient.

The glucose levels are monitored by a Continuous Glucose Monitoring (CGM) device which records glucose levels in the body every 5 minutes. We are given data for 5 patients in which monitoring begins 30 minutes before a meal and goes on for 2 more hours after the meal. The insulin infusions are categorized into 2 categories: Basal and Bolus. The Bolus infusion is the amount of insulin injected following a meal which directly counteracts the increase in glucose levels due to the absorption of Carbohydrates by the body. Whereas, the Basal infusion, also known as “Background Insulin,” releases insulin in the background and regulates the glucose levels throughout the day. Maintaining optimal glucose levels are extremely important in the case of diabetic patients, since either higher or lower than normal levels may lead to further medical complications.

The first phase of the project primarily has us deal with 2 streams of data, namely CGM Levels and the corresponding Time stamps.

Tasks

1. Feature Extraction

The first subtask has us take out 4 types of time series features using only the CGM data cell array and CGM timestamp cell array. Before applying our intuition and testing different techniques for feature extraction, the data has to be thoroughly preprocessed to account for the “real-world” data. Prior to feature extraction, we apply the following pre-processing techniques to better understand the data:

- For both CGM levels and corresponding timestamps, all cells after 30 columns were omitted.
- For both CGM levels and corresponding timestamps, the column ordering was reversed to make the data chronologically accurate from past to present.
- For both CGM levels and corresponding timestamps, all rows with NaN or missing values were synchronously deleted for more representative data.

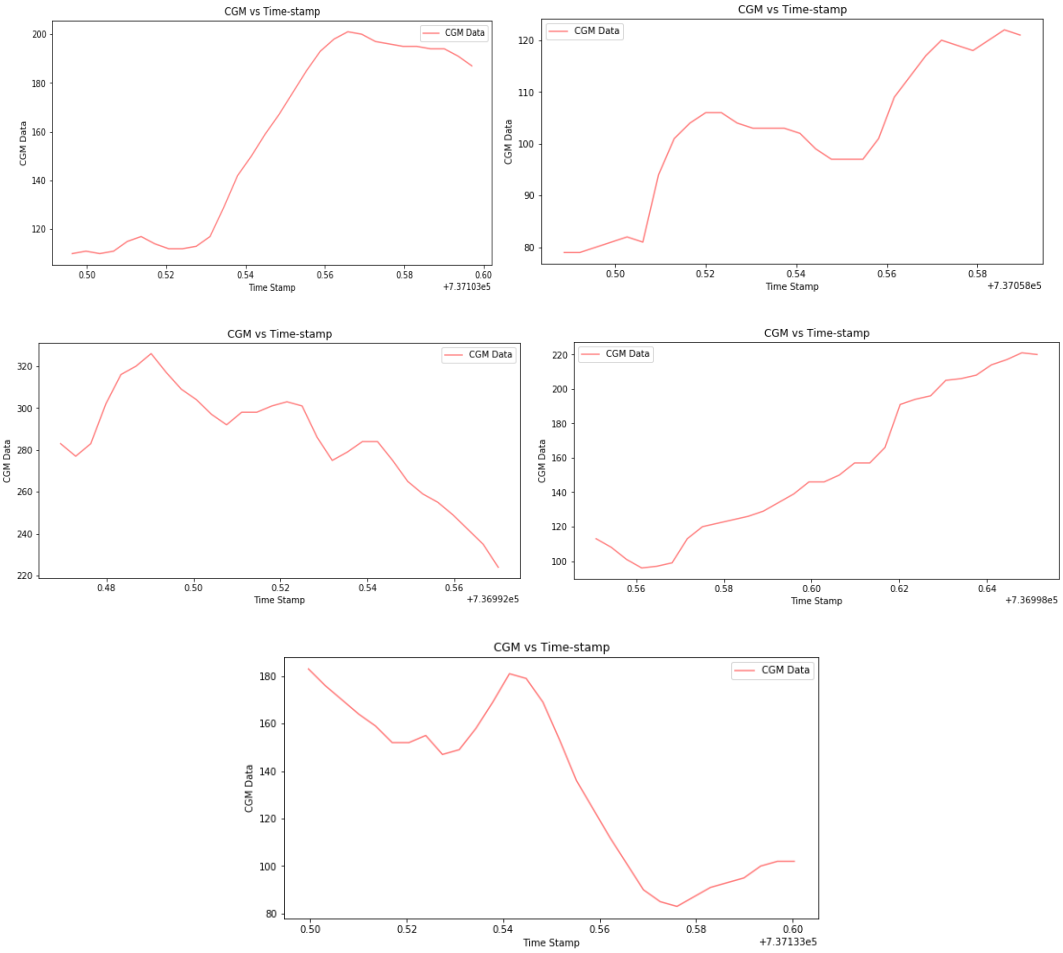
	cgmDatenum_1	cgmDatenum_2	cgmDatenum_3	cgmDatenum_4	cgmDatenum_5	cgmDatenum_6	cgmDatenum_7	cgmDatenum_8	cgmDatenum_9	cgmDatenum_10	cgmDatenum_1
0	737225.584155	737225.580683	737225.577211	737225.573738	737225.570266	737225.566794	737225.563322	737225.559850	737225.556377	737225.552905	737225.54943
1	737217.627778	737217.624306	737217.620833	737217.617361	737217.613889	737217.610417	737217.606944	737217.603472	737217.600000	737217.596528	737217.59305
2	737216.551319	737216.547847	737216.544375	737216.540903	737216.537431	737216.533958	737216.530486	737216.527014	737216.523542	737216.520069	737216.51659
3	737215.572095	737215.568623	737215.565150	737215.561678	737215.558206	737215.554734	737215.551262	737215.547789	737215.544317	737215.540845	737215.53737
4	737201.589410	737201.585938	737201.582465	737201.578993	737201.575521	737201.572049	737201.568576	737201.565104	737201.561632	737201.558160	737201.55468
5	737196.607431	737196.603958	737196.600486	737196.597014	737196.593542	737196.590046	737196.586574	737196.583102	737196.579630	737196.576157	737196.57268

Before Preprocessing

	cgmDatenum_30	cgmDatenum_29	cgmDatenum_28	cgmDatenum_27	cgmDatenum_26	cgmDatenum_25	cgmDatenum_24	cgmDatenum_23	cgmDatenum_22	cgmDatenum_21	cgmDatenum
0	737225.483461	737225.486933	737225.490405	737225.493877	737225.497350	737225.500822	737225.504294	737225.507766	737225.511238	737225.514711	737225.518
1	737217.527083	737217.530556	737217.534028	737217.537500	737217.540972	737217.544444	737217.547917	737217.551389	737217.554861	737217.558333	737217.561
2	737216.450625	737216.454097	737216.457569	737216.461042	737216.464514	737216.467986	737216.471458	737216.474931	737216.478403	737216.481875	737216.485
3	737215.471400	737215.474873	737215.478345	737215.481817	737215.485289	737215.488762	737215.492234	737215.495706	737215.499178	737215.502650	737215.506
4	737201.488715	737201.492188	737201.495660	737201.499132	737201.502604	737201.506076	737201.509549	737201.513021	737201.516493	737201.519965	737201.523
5	737196.506713	737196.510185	737196.513657	737196.517130	737196.520602	737196.524074	737196.527546	737196.531019	737196.534491	737196.537963	737196.541

After Preprocessing

Sample CGM vs Timestamp Data for 5 patients



After preprocessing, the following 5 types of features were extracted:

- Fast Fourier Transform
- Discrete Wavelet Transform
- Moving Average
- CGM Velocity
- Entropy

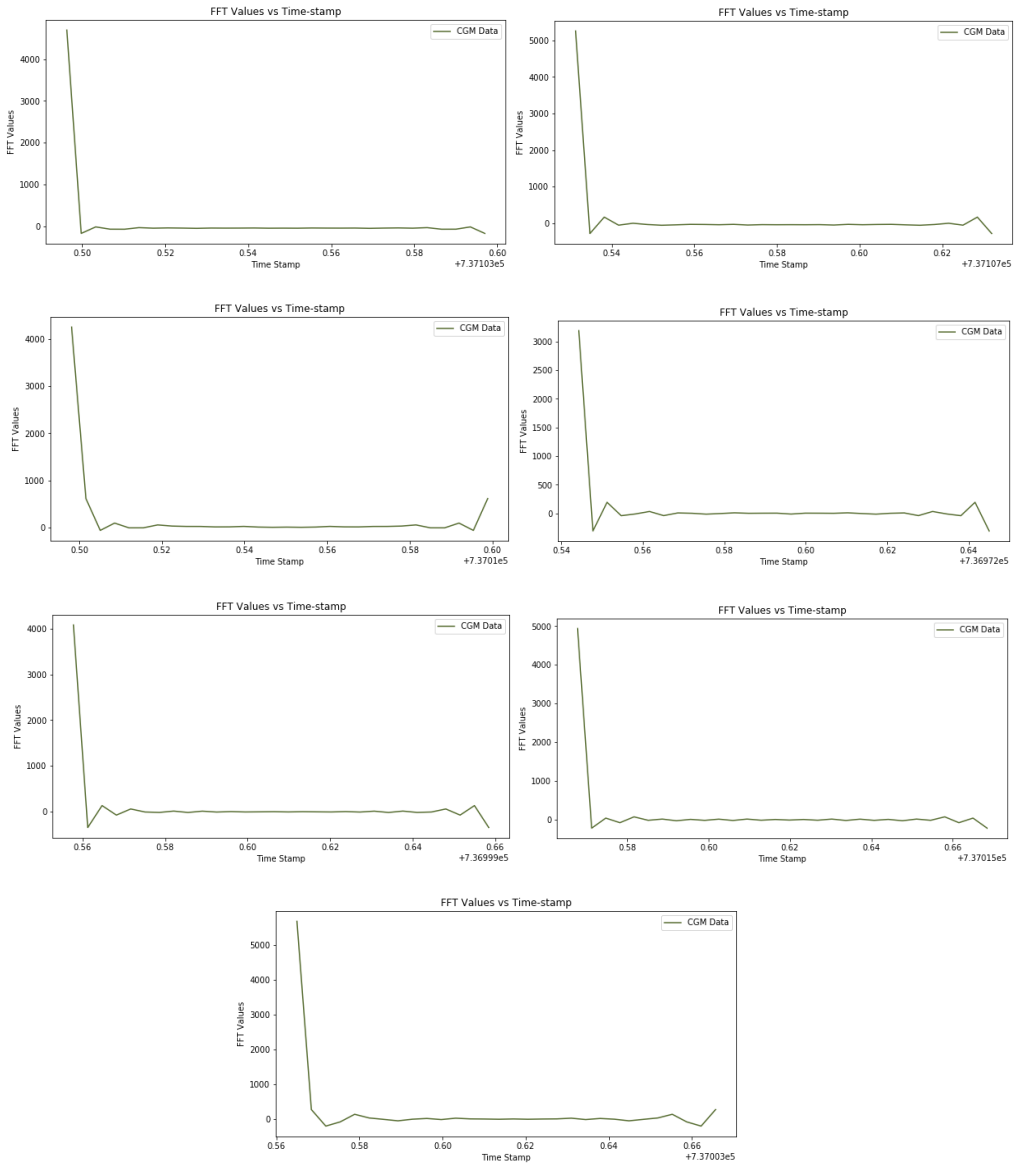
1.1 Fast Fourier Transform

Fast Fourier Transform (FFT) is an algorithm that efficiently computes discrete frequency components of a given signal over a time period. It computes the trigonometric series representing all the frequencies present in an input signal. We use the python library scipy.fftpack to compute FFT values for each row.

	cgmDatenum_30	cgmDatenum_29	cgmDatenum_28	cgmDatenum_27	cgmDatenum_26	cgmDatenum_25	cgmDatenum_24	cgmD
1	5159.000000-0.000000j	-289.606984+17.501297j	20.870977+164.218193j	-31.472136+78.117659j	-83.439155+73.210051j	24.000000+46.765372j	-48.562306+24.449955j	-25.680863+
2	5073.000000-0.000000j	-590.888819+56.098033j	25.415645+260.650017j	-2.746711+29.997292j	-12.959701+27.061316j	-18.000000+3.464102j	-59.989357+45.405954j	-31.784939+
3	5586.000000-0.000000j	-327.632440+46.320281j	-184.932657-69.383772j	-45.551663-5.567582j	40.242876-16.625575j	3.000000-10.392305j	11.690983-18.519102j	-20.661457-
4	4166.000000-0.000000j	-178.998456-7.950356j	54.837700+104.405420j	-7.572949-56.818633j	-22.490199+84.137439j	27.500000-38.971143j	-22.253289+9.286051j	6.193210+
6	5299.000000-0.000000j	231.790118-146.650689j	24.613542+51.399428j	-93.002512-24.654224j	56.145930-102.406282j	11.000000-17.320508j	47.579527-21.596785j	15.429571-

Calculated FFT Values

Sample FFT vs Timestamp Graphs for each patient



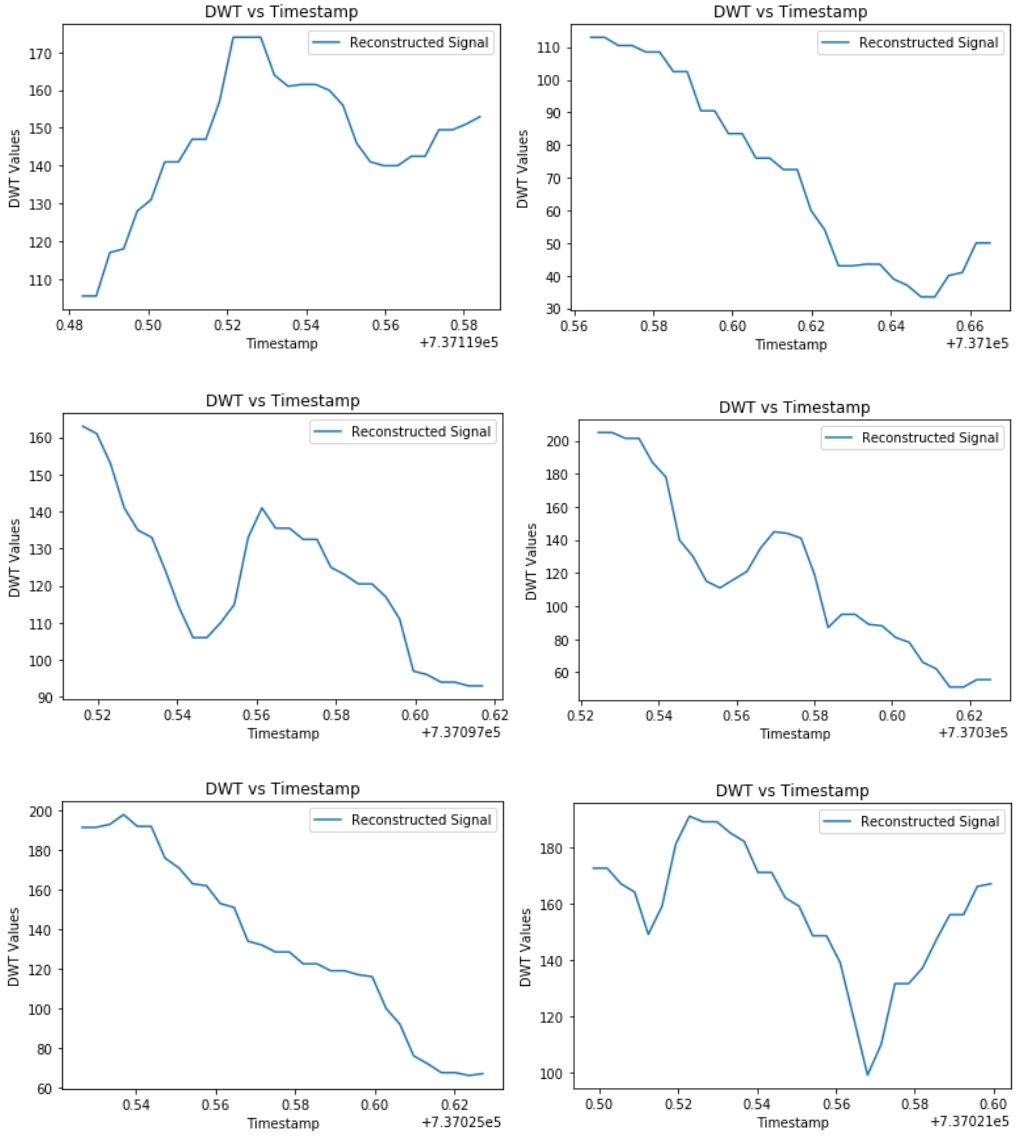
1.2 Discrete Wavelet Transform

Discrete Wavelet Transform (DWT) is an implementation of wavelet transform using a disjoint set of wavelet spaces. The transform decomposes the signal into mutually orthogonal set of wavelets. We use the python library pywt to calculate the DWT values of each row.

0	105.477943	105.477943	117.001615	117.954270	128.001615	130.954270	140.977943	140.977943	146.977943	146.977943	157.001615	173.954270	173.977943
1	86.001615	87.954270	97.977943	97.977943	96.977943	96.977943	121.001615	130.954270	152.001615	157.954270	172.001615	179.954270	189.001615
2	129.977943	129.977943	137.977943	137.977943	145.001615	158.954270	168.977943	168.977943	169.977943	169.977943	166.977943	166.977943	164.477943
3	100.477943	100.477943	99.977943	99.977943	101.477943	101.477943	106.977943	106.977943	108.977943	108.977943	115.001615	122.954270	144.977943
4	176.977943	176.977943	167.954270	165.001615	166.477943	166.477943	159.977943	159.977943	157.977943	157.977943	150.954270	148.001615	137.977943
5	152.977943	152.977943	150.977943	150.977943	148.977943	148.977943	154.477943	154.477943	145.954270	135.001615	123.477943	123.477943	138.001615

Calculated DWT Values

Sample DWT vs Timestamp graphs for each patient



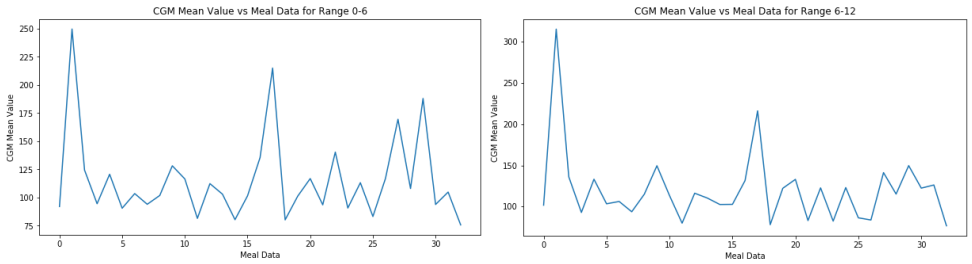
1.3 Moving Average [Discrete – 30-minute intervals]

Moving averages are calculated for a window size of 30 minutes (6 samples). The 30 given samples are divided into 5 intervals and the highest mean is calculated amongst the given interval.

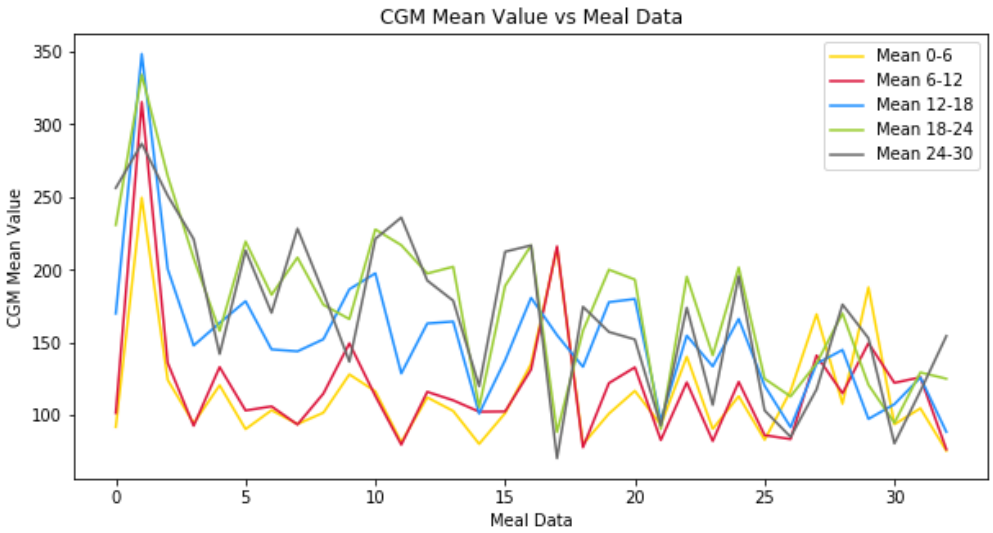
	Mean 0-6	Mean 6-12	Mean 12-18	Mean 18-24	Mean 24-30
1	143.500000	177.166667	192.000000	173.166667	174.000000
2	120.000000	178.333333	209.166667	177.000000	161.000000
3	166.000000	194.666667	198.000000	204.500000	167.833333
4	126.666667	137.666667	155.666667	138.833333	135.500000
6	196.000000	181.833333	167.833333	155.166667	182.333333
7	177.000000	165.500000	172.000000	130.000000	103.833333
10	136.666667	118.166667	94.500000	67.500000	67.333333
11	125.833333	118.666667	144.500000	99.833333	87.000000
12	250.333333	272.333333	246.833333	238.333333	217.500000

Calculated windowed discrete average values

Individual CGM Mean vs Number of meals graph



Combined CGM Mean for all intervals vs Number of meals graph



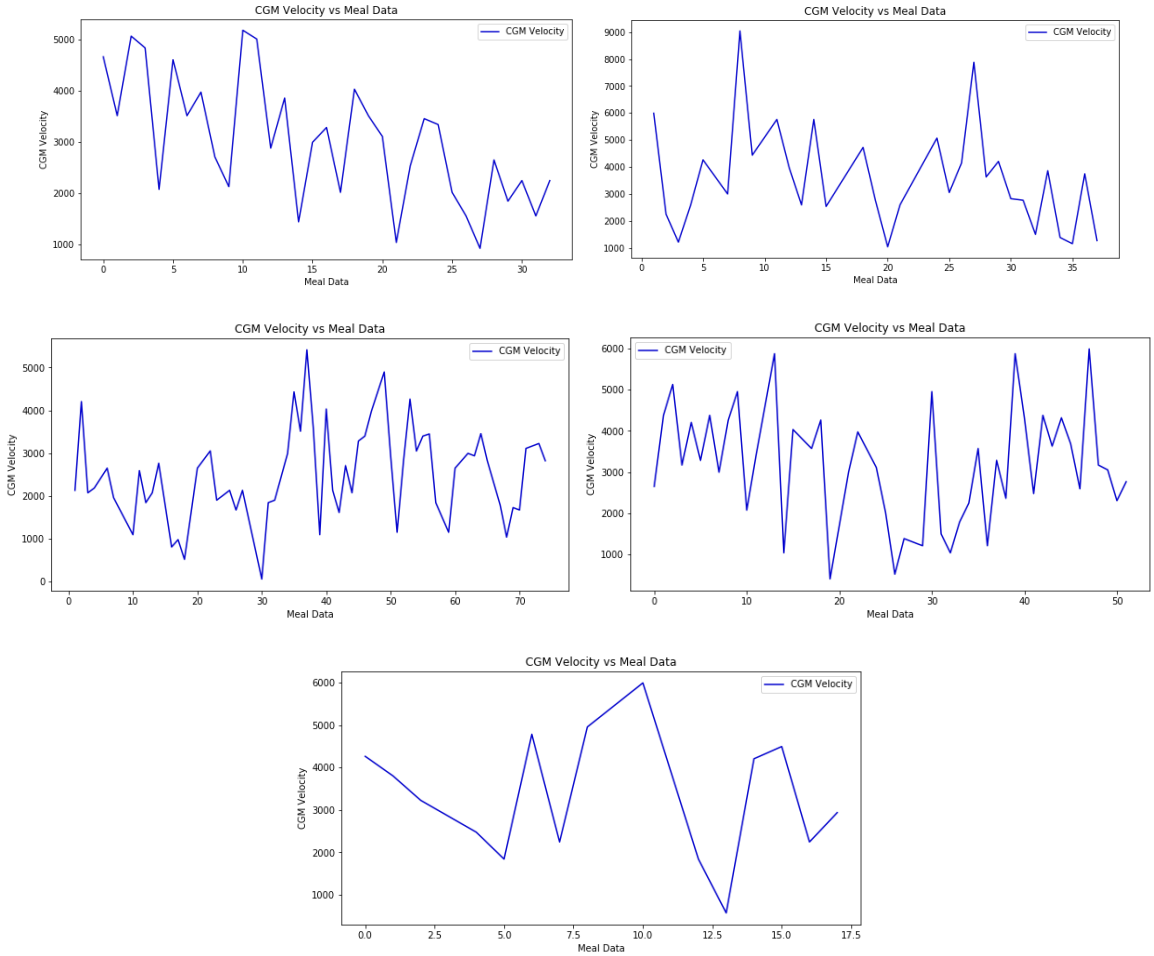
1.4 CGM Velocity [Rolling – 30-minute intervals]

The velocity of CGM values are computed by calculating the rate of change of glucose levels with respect to the time period. We use the two-point coordinate formula for calculating the slope of a graph. The maximum velocity calculated from all the rolling intervals is taken as the velocity of the row.

	Velocity 1-5	Velocity 2-6	Velocity 3-7	Velocity 4-8	Velocity 5-9	Velocity 6-10	Velocity 7-11	Velocity 8-12	Velocity 9-13	Velocity 10-14
0	288.000000	460.800006	460.800000	460.800003	460.800003	288.000004	806.399956	1900.800024	2707.200016	3686.399997
1	3398.400020	2995.199998	2822.399998	2822.400017	3168.000019	3455.999835	3513.600045	3110.400039	2246.400013	1555.200009
2	230.400003	518.400007	633.599974	691.200009	633.600004	576.000000	460.800000	921.599999	2303.999998	3628.800046
3	115.200001	-345.600004	-115.199994	-57.600001	0.000000	-115.200001	345.600000	576.000007	1727.999999	2937.600018
4	-172.800000	-57.600001	57.600000	460.800003	921.600005	1324.800017	1785.599999	2073.599901	2016.000012	1727.999999
5	633.600000	633.600008	518.400007	288.000002	172.800001	460.800006	1727.999999	2764.799850	3686.400022	4377.599997

Calculated windowed rolling CGM velocity

CGM Velocity vs Meal Data for each patient



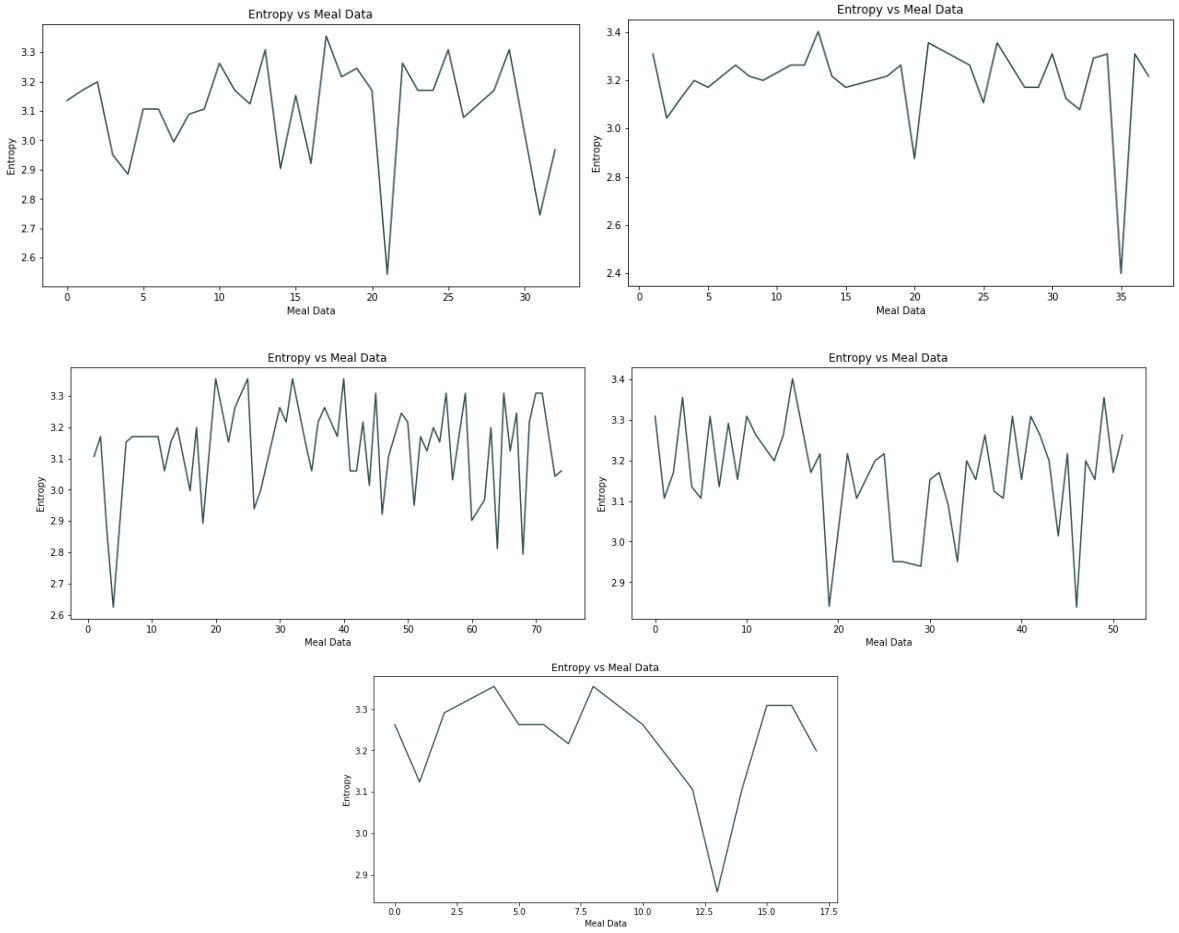
1.5 Entropy

Entropy is used to quantify the measure of randomness and unpredictability of the data over a given time period[1]. We use the python library scipy.stats for calculating entropy values for each individual cell and take the maximum value of each of the rows.

	Entropy
0	3.135265
1	3.170148
2	3.198917
3	2.950426
4	2.884467
5	3.106497
6	3.106497
7	2.994328
8	3.089055

Calculating entropy values for each row

Entropy vs Meal Data for each patient



2. Intuition

The intuition behind selecting the following features is to observe the general shape of the curve. The meal intake reported at 30-minutes, would ideally signal a sharp rise in the CGM values due to carbohydrate absorption and eventually plateau. Afterwards, due to the insulin injection the graph would fall which would keep the glucose levels in check. We wish to learn this pattern to classify whether a meal was taken or not.

2.1 Fast Fourier Transform (FFT)

Fast Fourier Transform (FFT) is an algorithm that efficiently computes discrete frequency components of a given signal over a time period. It computes the trigonometric series representing all the frequencies present in an input signal.

The intuition behind selecting FFT as a feature is that it removes noise from the data by filtering it by moving back and forth the time period of the given wave. FFT values discerns meaningful patterns from the data by displaying amplitudes of the highest frequencies, which in turn, correspond to a meal intake in the given CGM time series.

2.2 Discrete Wavelet Transform (DWT)

Discrete Wavelet Transform (DWT) is an implementation of wavelet transform using a discrete set of wavelet spaces. The transform decomposes the signal into mutually orthogonal set of wavelets.

The advantage behind selecting DWT as a feature is the ability of the wavelet transform to capture both frequency and location information. This is beneficial in analyzing CGM data, where peak values occur at different timestamps (due to possibly missing a meal/earlier consumption of the meal).

2.3 Moving Average [Discrete – 30-minute intervals]

Moving averages are calculated for a window size of 30 minutes (6 samples). The 30 given samples were divided into 5 intervals and the highest mean was calculated amongst the given interval.

The intuition behind selecting moving average as a feature is straightforward. A higher mean would correspond to high collection of CGM values which in turn would signal that the meal has been consumed. The window where the most significant peak is observed would correspond to the state where the carbohydrates have been absorbed by the body, and insulin needs to be injected.

2.4 CGM Velocity [Rolling – 30-minute intervals]

The velocity of CGM values are computed by calculating the rate of change of glucose levels with respect to the time period.

The intuition behind selecting CGM velocity as a feature is to observe the sharp rise in CGM values during different phases of the meal. We use rolling 30-minute intervals to learn the significant window where the increase would be highest which would, in turn, correspond to the meal intake.

2.5 Entropy

Entropy is used to quantify the measure of randomness and unpredictability of the data over a given time period[1].

The intuition behind selecting entropy impurity as a feature is to symbolize the time series. The maximum value of randomness would correspond to the period of meal intake.

3. Feature Matrix

The feature matrix for Patient 1, after preprocessing has 32 timeseries and 13 features. Hence, our final feature matrix has the shape 32x13

	FFT1	FFT2	FFT3	FFT4	FFT5	DWT	mean0-6	mean6-12	mean12-18	mean18-24	mean24-30	maximumVelocity	Entropy
0	5105.0	315.231637	315.231637	210.702384	210.702384	231.745084	92.000000	101.666667	170.000000	230.833333	256.333333	4665.599997	3.135265
1	9207.0	19.203178	19.203178	23.430749	23.430749	322.745084	249.666667	315.333333	348.500000	334.166667	286.833333	3513.600045	3.170148
2	5858.0	112.537069	112.537069	52.832186	52.832186	245.245084	124.333333	136.000000	200.500000	264.500000	251.000000	5068.800064	3.198917
3	4587.0	149.395871	149.395871	1090.843661	1090.843661	202.245084	94.500000	92.833333	148.000000	208.000000	221.166667	4838.400061	2.950426
4	4305.0	79.034704	79.034704	12.974602	12.974602	139.245084	120.666667	133.166667	163.500000	158.000000	142.166667	2073.599901	2.884467
5	4831.0	171.107745	171.107745	133.551488	133.551488	197.081030	90.500000	103.333333	178.500000	219.500000	213.333333	4608.000058	3.106497
6	4250.0	117.752258	117.752258	27.639121	27.639121	159.245084	103.500000	106.166667	145.333333	182.833333	170.500000	3513.600045	3.106497
7	4611.0	1126.411906	1126.411906	159.053810	159.053810	202.245084	94.000000	93.666667	144.000000	208.500000	228.333333	3974.399997	2.994328
8	4377.0	149.284235	149.284235	104.404490	104.404490	159.245084	101.833333	115.000000	152.333333	176.000000	184.333333	2707.200016	3.089055
9	4604.0	110.315508	110.315508	13.312085	13.312085	164.245084	128.166667	149.500000	186.666667	166.166667	136.833333	2126.946030	3.106497
10	5260.0	232.662365	232.662365	101.542999	101.542999	202.245084	116.500000	113.333333	197.666667	227.833333	221.333333	5184.000031	3.262568
11	4459.0	1340.743114	1340.743114	121.445570	121.445570	214.745084	81.500000	79.833333	128.833333	217.000000	236.000000	5011.199996	3.170148
12	4691.0	90.142332	90.142332	64.784257	64.784257	173.245084	112.333333	116.166667	163.166667	197.500000	192.666667	2879.999998	3.123939
13	4553.0	32.592176	32.592176	42.917864	42.917864	177.745084	103.000000	110.333333	164.500000	202.166667	178.833333	3859.200023	3.308778
14	3058.0	41.474595	41.474595	13.228757	13.228757	94.245084	80.333333	102.500000	101.166667	105.666667	120.000000	1439.999999	2.904216

Final 32x13 shape of the feature matrix

4. Principal Component Analysis

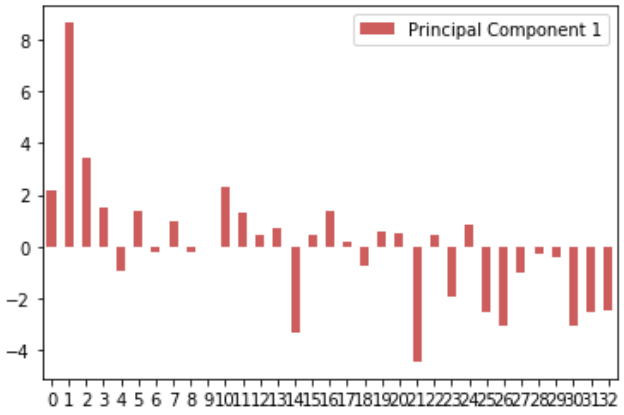
Providing the following feature matrix to PCA, we obtain a new feature matrix given below:

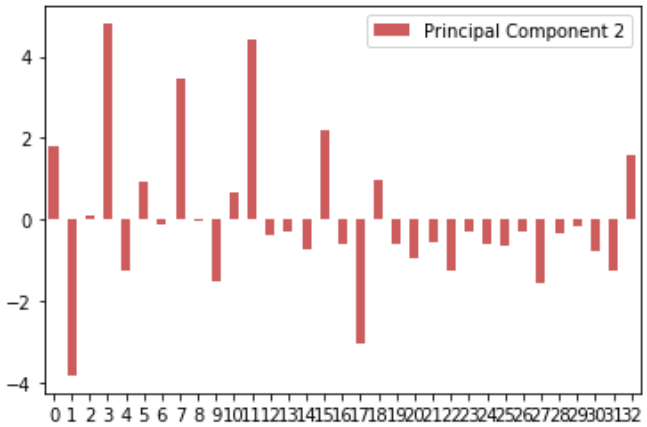
```
[[ 0.4122384   0.03590644  0.03590644  0.06047205  0.06048108  0.41363361
   0.21717158  0.26562948  0.38054369  0.39164989  0.35486125  0.27689099
   0.17479299]
 [-0.10679671  0.38752925  0.38752925  0.34320571  0.34318949  0.05154199
  -0.34407282 -0.36482738 -0.18514934  0.09049249  0.23424917  0.31310547
  -0.05611476]
 [ 0.00455832 -0.45802593 -0.45802593  0.51475139  0.51476387 -0.04502114
  -0.02177293  0.009937    0.06978156  0.00611705 -0.04180182  0.03695166
  -0.19958326]
 [ 0.03313254  0.31480888  0.31480888  0.29832657  0.2983173   0.08964245
   0.51067383  0.34486963 -0.12208157 -0.2542246  -0.17071507 -0.3525927
  -0.04408078]
 [ 0.13516552  0.08460058  0.08460058 -0.10153186 -0.10159681 -0.06001688
  -0.07551192  0.08398256  0.1395032   0.14237373  0.19567975 -0.18314123
  -0.90491176]]
```

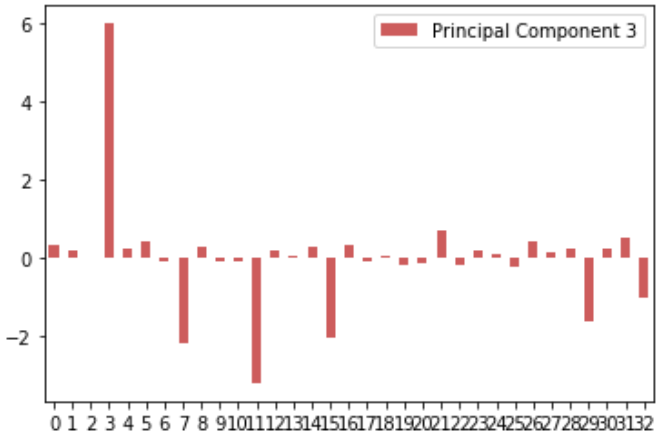
4.1 Plotting Top 5 features vs time series

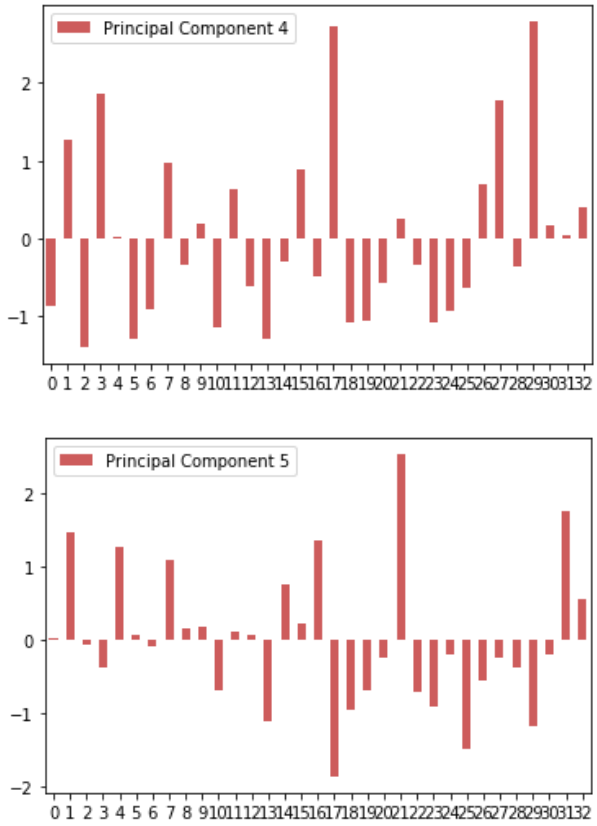
In principal component 1, we observe that columns corresponding to FFT features have higher values which means it significantly contributes to the data along the given component.

The following plots show the top 5 features vs time series:



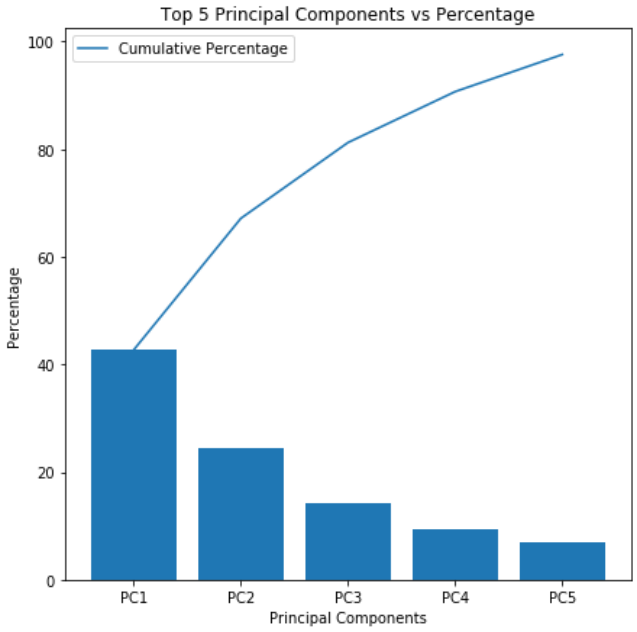






Principal components vs time series

Selecting the Top 5 principal components, we observe that 97% of the original data can be represented using the top 5 features instead of the original 13 with the first principal component accounting for 42% of the data.

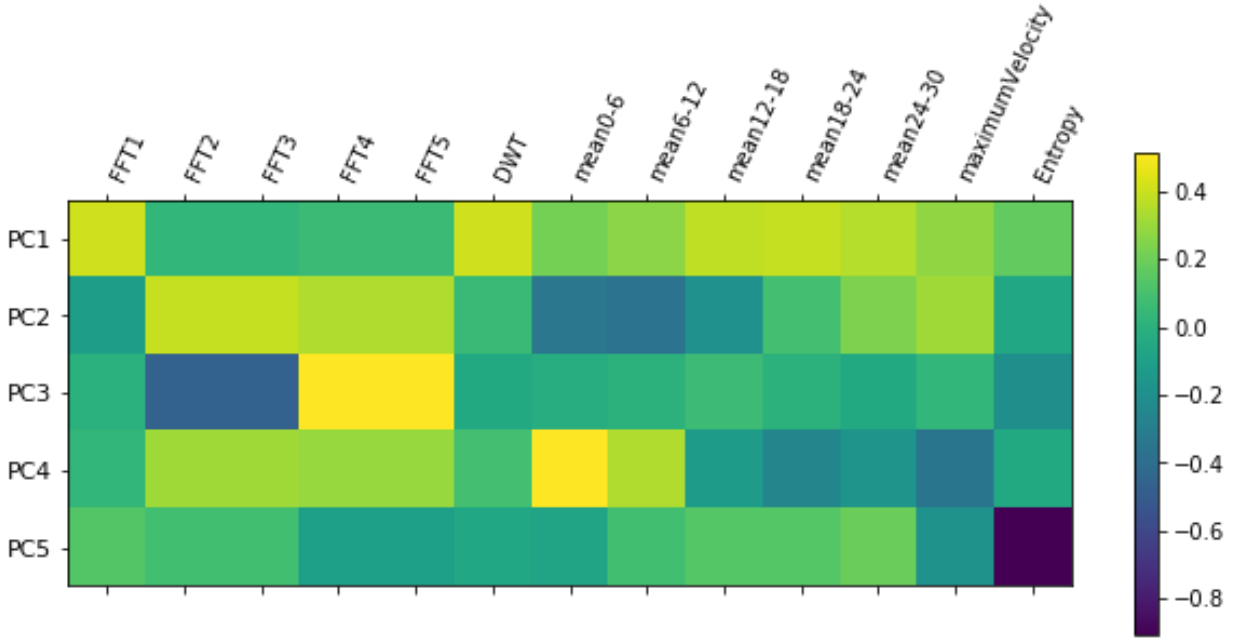


Principal components as percentages of variance

5. Explanation for selecting Top 5 features

5.1 Visualizing through a heat plot

We visualize the top features by using a heatmap between the features and the principal components.



Mapping important features with top 5 principal components

In the first principal component, we observe that the first feature of FFT and DWT has the maximum value and can be shown as significant features for meal detection.

In the second principal component, we observe that most of the FFT features have the highest value, signifying that FFT can be used as an important feature for meal detection.

In the third principal component, we observe that the third and fourth FFT features have the highest value, signifying that FFT can be used as an important feature for meal detection.

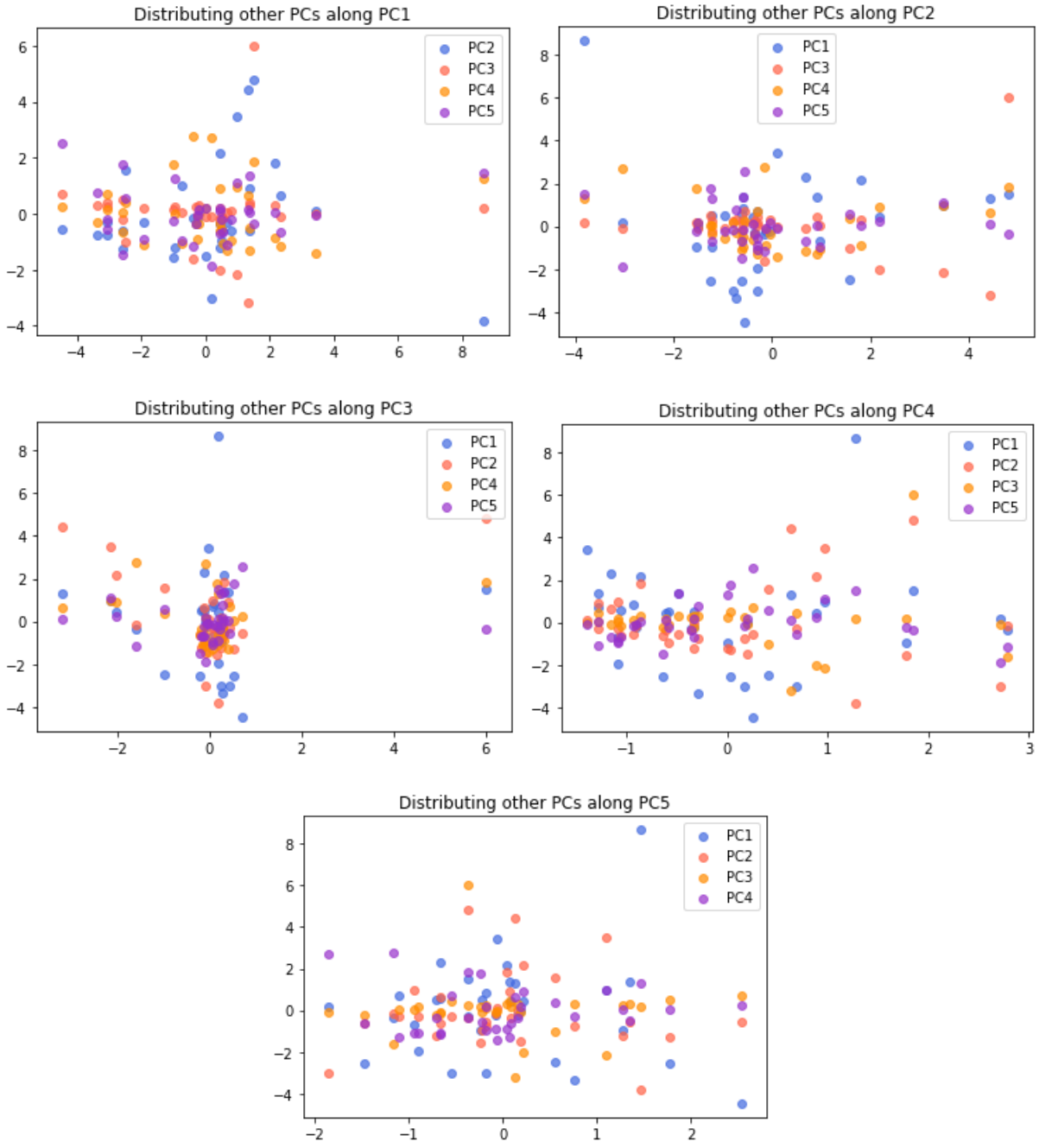
In the fourth principal component, we observe that the first mean interval has the highest value, signifying that mean can be used as an important feature for meal detection.

In the fifth principal component, we observe that most of the FFT and mean interval features have the highest value, signifying that both FFT and mean can be used as an important feature for meal detection.

5.2 Visualizing through spread of principal components

As we observed, the top 5 principal components contribute to most of the variance in the dataset.

We visualize this by observing the spread of data along a given principal component. The principal component 1 (indicated in blue), shows the maximum spread in the data signifying high variance and classifying power.



6. References

[1] Sam T, “Entropy, How Decision Trees Make Decisions”, *Medium*, Accessed 10. Feb. 2020.