ASU ID-1217319415
PRANJALI PATIL
CSE 572
Data Mining
Project 1

TASK 1

Task is to extract four different types of time series features from only the CGM data cell array and CGM timestamp cell array. Before starting using the data some pre-processing is required. Missing values are replaced by zeros. We have concatenated the data and performed feature extraction.

Four types of features extracted are:

1. Fast Fourier Transform
2. Rolling Mean and Rolling Deviation
3. Polynomial fit
4. Inter-Quartile Range

TASK 2

Feature 1: Fast Fourier Transform

Fast Fourier Transform (FFT) converts data in Time domain to frequency domain. The number of computations reduces significantly to 2NlogN for N points.

In Python, scipy.fft.fft computes the one-dimensional discrete Fourier transform. FFT of glucose series is plotted against frequency. Using this, glucose level can be depicted on the graph and the highest amplitude corresponding to it are observed.

Feature 2: Polynomial fit

In Polynomial fit, it fits a nonlinear relationship between the value of x and conditional mean of y. It employs the least-squares method to fit the data. However, it is considered as a special case of multiple linear regression. Glucose level pattern can be observed as polyfit regression is applied on the data. Using different degrees, we can observe different ways in which it fits the CGM data. In Python, “polyfit” function can be used for calculating polynomial fit.

Feature 3: Rolling Mean and Rolling Deviation

Rolling Mean is simply unweighted mean of the last n values. It analyses various data points by creating a series of averages of different subsets of the full data set. A moving average is commonly used with time series data to smooth out short-term fluctuations and highlight longer-term trends or cycles [1]. It is a good feature as compared to the mean. It will consider all the attributes. Using “rolling” function in Python and by specifying mean, rolling mean can be calculated.

Rolling Standard Deviation

Rolling deviation can be defined as standard deviation for a moving window. It is a measure of the amount of variation of the data points. The formula given below is used for calculation

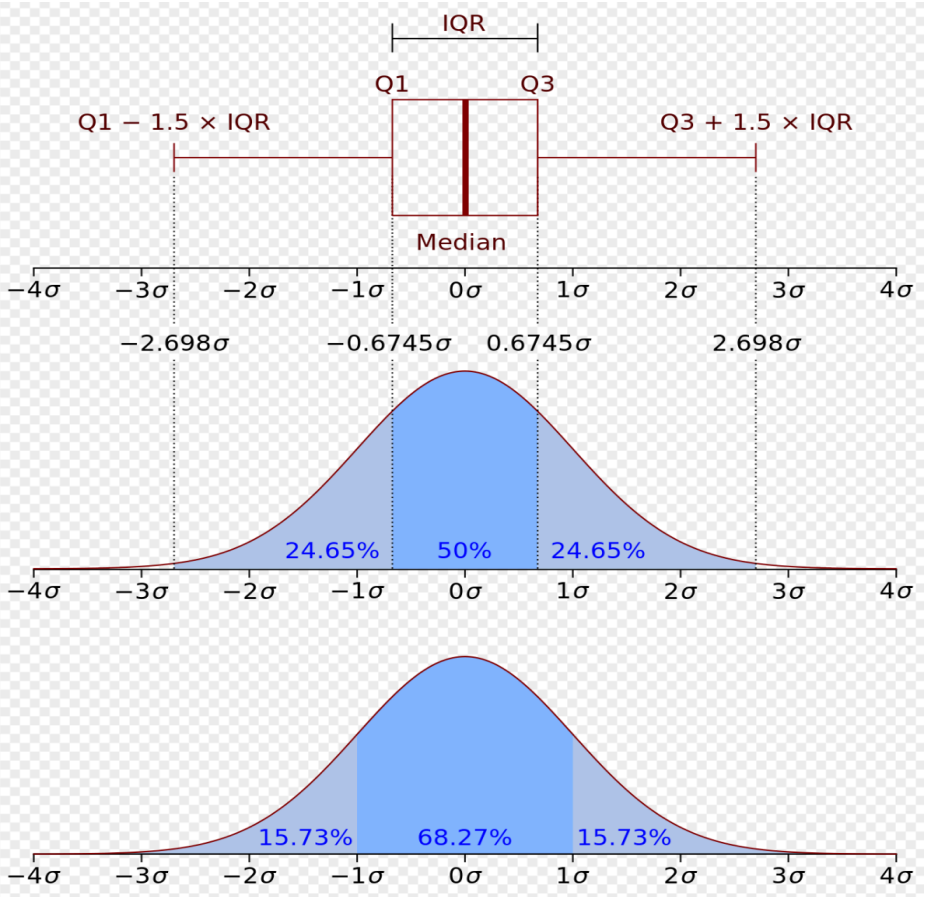
of standard deviation. Rolling Standard Deviation is a good feature as it will tell the amount of variance and it is easier for identification of values that not closely associated. . Using “rolling” function in Python and by specifying std, rolling mean can be calculated.

$$s^2 = \frac{\sum_{i=1}^{N} x_i^2 - N\bar{x}^2}{N-1}$$

Feature 4: Inter-Quartile Range

In case of Inter-Quartile Range, data is divided into two groups: High and Low. Statistical medians is found of the low and high groups: Quartile 1 (Q1) and Quartile 3 (Q3). Inter Quartile Range is defined as Q3-Q1

Variability is summarized by interquartile range when a data set has outliers.



TASK 3

Fast Fourier Transform

As depicted in Figure 1, FFT will gives a mirror image and top 5 peaks are computed from the graph. As FFT gives transformation in the frequency domain, it is easier to observe the peaks. It can be used in order to distinguish between no meal or meal. Intuition is correct as it can be clearly observed from the graph using the peaks. Peaks can tell us the information about the glucose level drastic increase which can be useful information.

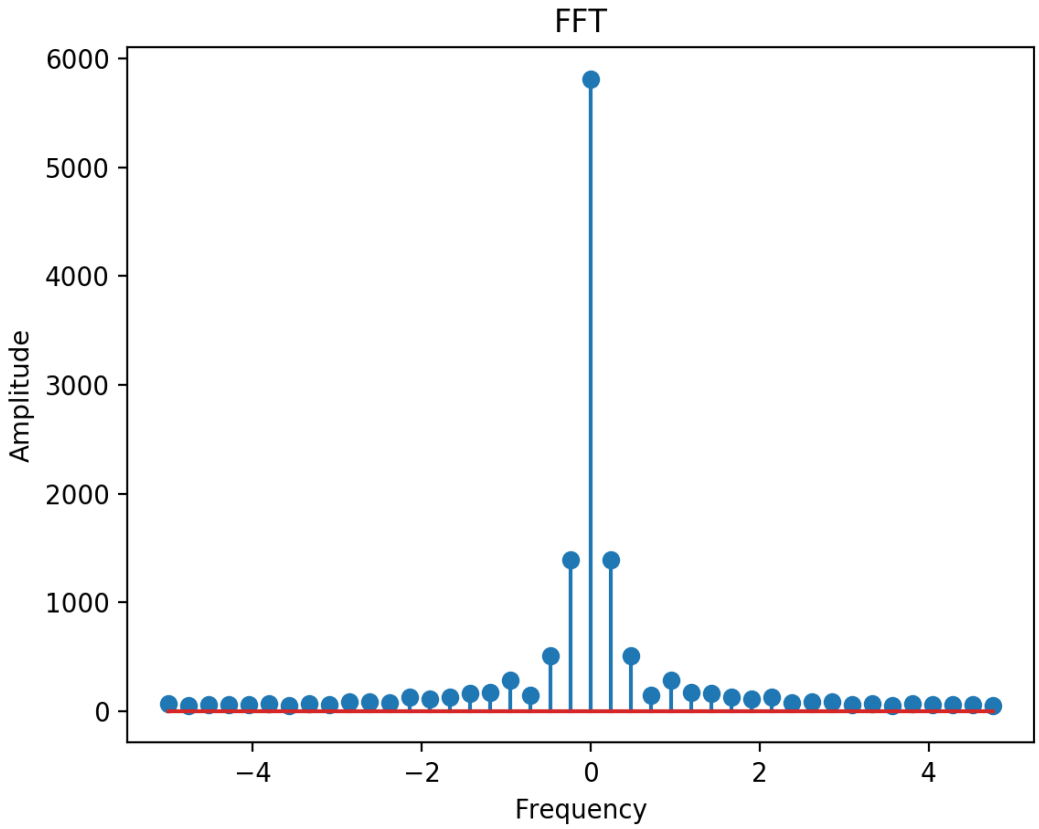


Figure 1: FFT of CGM Signal

Polynomial Fit

As depicted in Figure 2, polynomial fit function is fitting the CGM data. This curve has also be used for reference. I have chosen degree 4 for curve fitting. It can be tried out with different polynomial degrees. Intuition is correct regarding this feature as it can be used for generating different patterns.

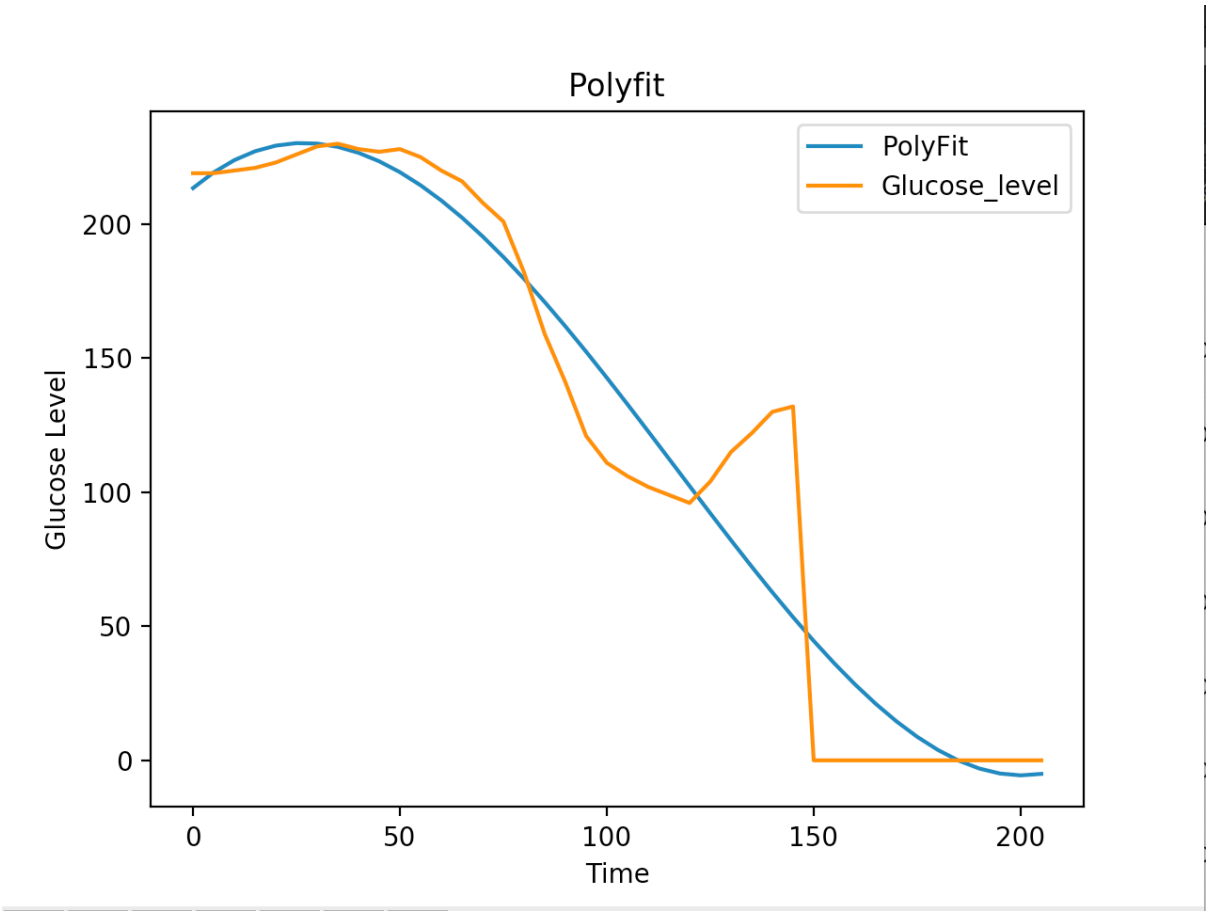


Figure 2: Polynomial fit representation of Order 4

Rolling Mean and Rolling Standard Deviation

Rolling Mean depicts mean in different subsets. Even if there is a spike in a data, it will not affect the rolling mean in a major way. It is a good feature to depict the mean of the data without being affected by one particular value.
Rolling Standard Deviation will give us the variance in the glucose data. As calculated in mean, values calculated in subsets is what makes it unique. Using this variance between data can be calculated and glucose values can be observed. We can observe using this data whether there are major changes.

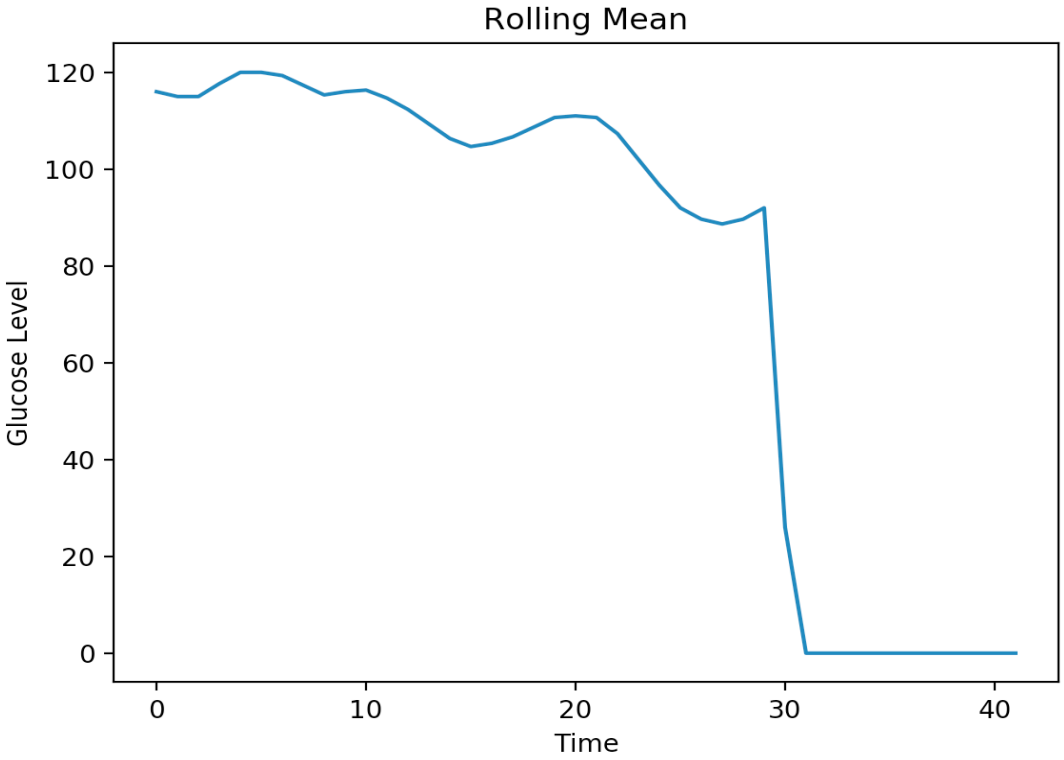


Figure 3

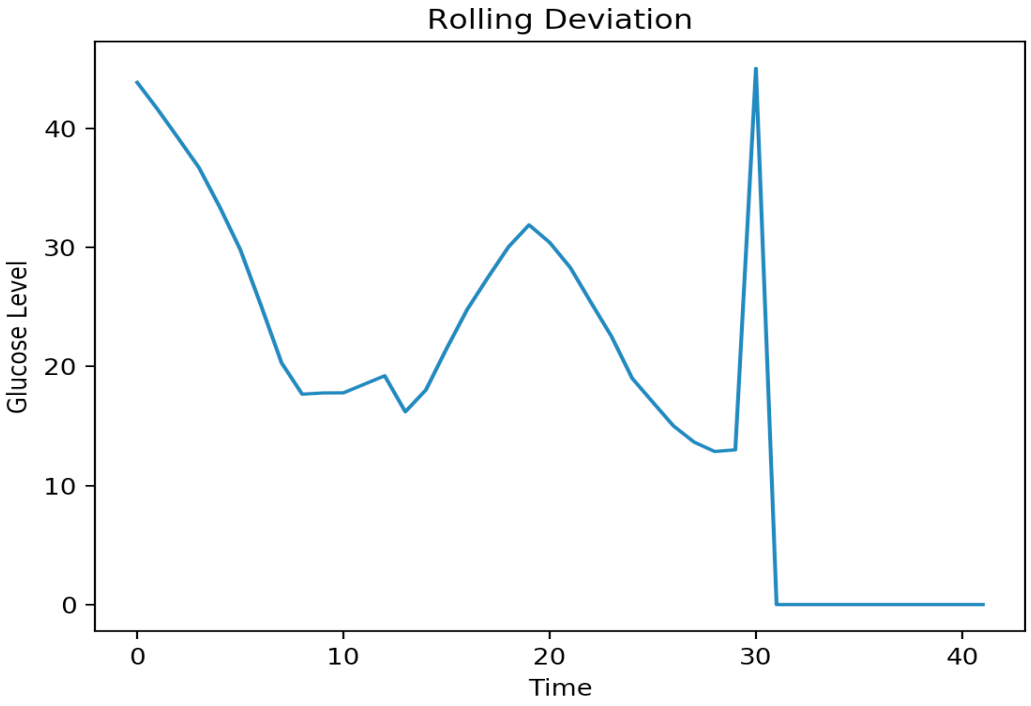


Figure 4

InterQuartile Range
It is an important feature in case of outliers. Using this feature, we can conclude if a patient's glucose level is extremely high or low. This can also be an indicator as to meal being taken or not.

TASK 4

FFT Feature matrix
Feature Matrix
(216, 101)

Rolling_Mean

	cgmSeries_ 1	cgmSeries_ 2	...	cgmSeries_41	cgmSeries_42
0	NaN	NaN	...	NaN	NaN
1	NaN	NaN	...	NaN	NaN
2	249.333333	256.000000	...	0.0	0.0
3	240.000000	247.000000	...	0.0	0.0
4	199.000000	201.666667	...	0.0	0.0
..	...	...	...	...	...
13	167.333333	168.666667	...	0.0	0.0
14	178.333333	182.666667	...	0.0	0.0
15	224.333333	225.666667	...	0.0	0.0
16	266.666667	268.666667	...	0.0	0.0
17	279.666667	281.333333	...	0.0	0.0

Rolling Mean

[216 rows x 42 columns]
Rolling_Deviation

	cgmSeries_ 1	cgmSeries_ 2	...	cgmSeries_41	cgmSeries_42
0	NaN	NaN	...	NaN	NaN
1	NaN	NaN	...	NaN	NaN
2	15.143756	18.083141	...	0.0	0.0
3	17.435596	22.715633	...	0.0	0.0
4	53.730811	56.083271	...	0.0	0.0
..	...	...	...	...	...
13	23.692474	25.794056	...	0.0	0.0
14	36.637867	39.513711	...	0.0	0.0
15	90.533603	89.745938	...	0.0	0.0
16	53.500779	50.500825	...	0.0	0.0

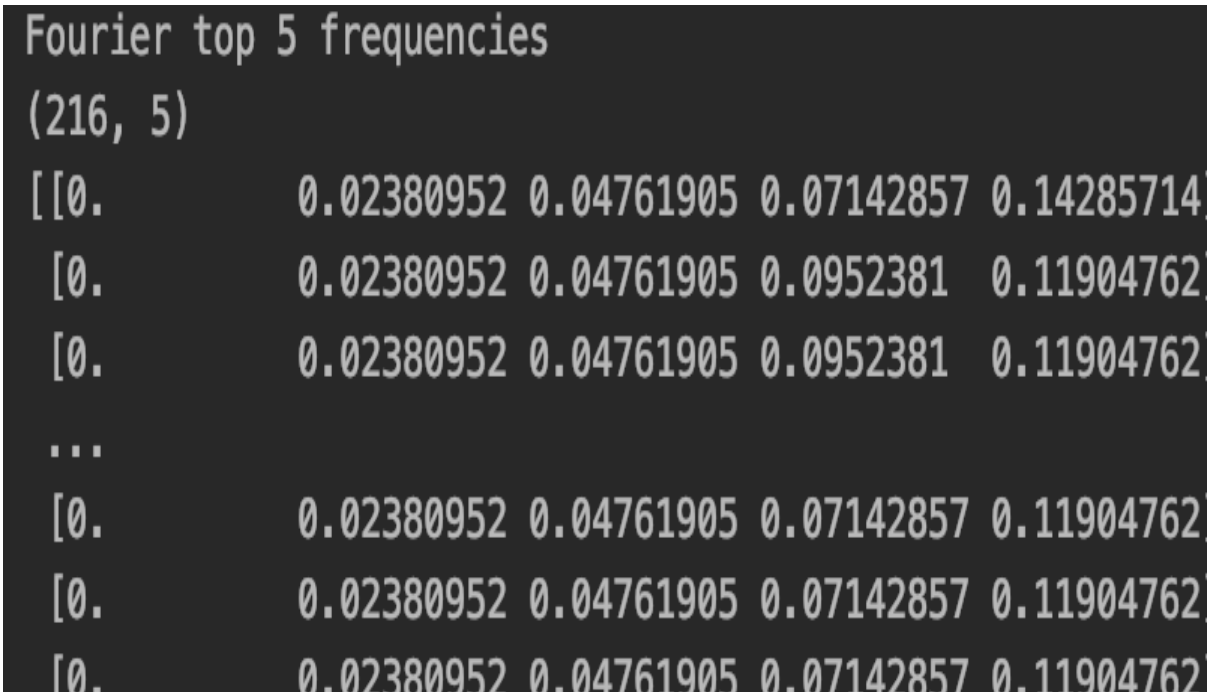
Rolling Deviation

```
Inter_Quartile_Range
(216, 1)
[[172. ]
 [335. ]
 [237.5]
 [158. ]
 [156. ]
 [168. ]
 [166.5]
 [193.5]
 [125. ]
 [156.5]
 [219.5]
 [195.5]
 [189. ]
 [180.5]
 [105. ]
 [124.5]
 [213.5]
 [195. ]
 [156. ]
```

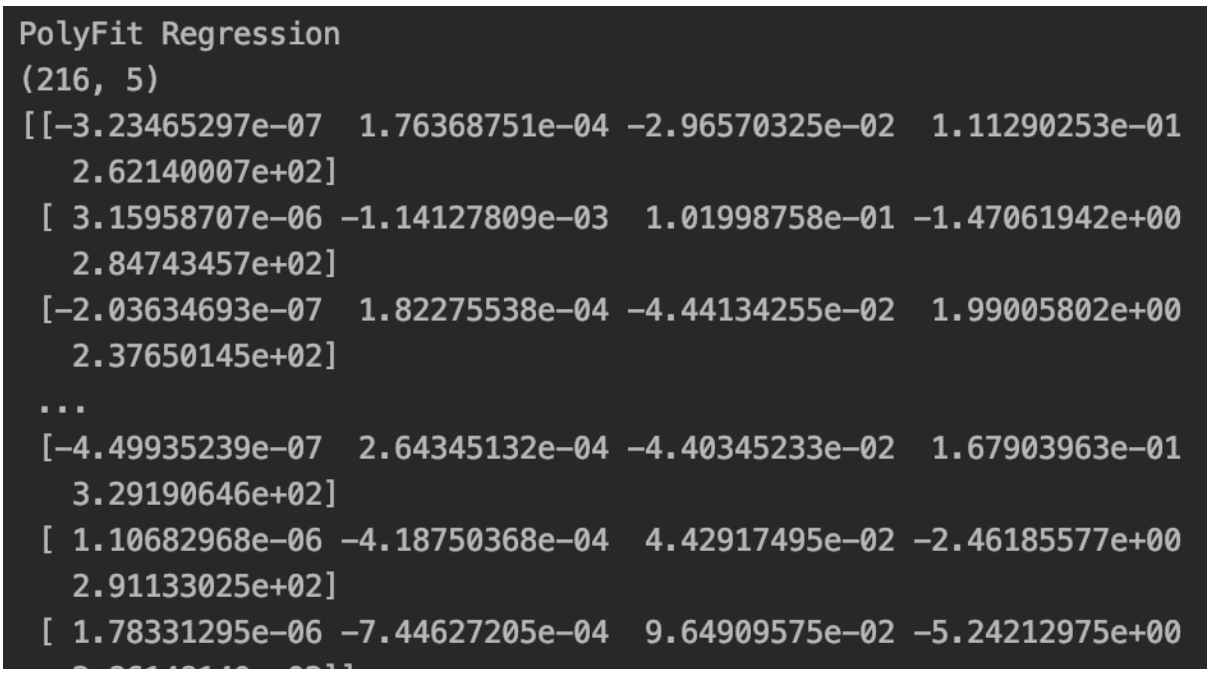
Inter-Quartile Range

```
Fourier Top 5 Peaks
(216, 5)
[[ 414.63262976  550.14124633 1247.65074728 2246.98320471 5196.        ]
 [ 394.57553624  678.02866188 1426.89644702 3786.42494359 9207.        ]
 [ 370.65812496  417.64415904 1310.8955883  2539.29058953 5858.        ]
 ...
 [ 530.2741941   748.90290423 1371.3487975  2998.5420782  5901.        ]
 [ 570.49205469  611.46038709 1417.35975584 2349.49249071 6225.        ]
 [ 553.30612694  606.51119717 1351.32667236 1859.46305902 5339.        ]]
```

Fourier top 5 Peaks



Fourier transform top 5 frequencies



Polynomial fit Regression

TASK 5

```
Feature Matrix
(216, 101)
[[ 0.00000000e+00  0.00000000e+00  0.00000000e+00 ... -1.51476321e-01
   3.51438016e+00  2.42779839e+02]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00 ... -3.34217712e-02
   2.31242874e+00  2.63221707e+02]
 [ 2.49333333e+02  2.56000000e+02  2.63666667e+02 ... -2.01192276e-01
   6.36976282e+00  2.12734018e+02]
 ...
 [ 2.24333333e+02  2.25666667e+02  2.28666667e+02 ... -1.29164510e-01
   2.54605771e+00  3.15661337e+02]
 [ 2.66666667e+02  2.68666667e+02  2.73666667e+02 ... -1.32103379e-01
   2.46584072e+00  2.63099375e+02]
 [ 2.79666667e+02  2.81333333e+02  2.83333333e+02 ... -6.49015246e-02
  -7.33540578e-01  2.60498789e+02]]
```

Feature Matrix

```
After PCA
(216, 5)
[[ 4.20654082e+01 -7.33450566e+02 -1.01495401e+03 -9.78539950e+01
  -1.64229833e+03]
 [ 8.65289405e+01 -1.21975869e+03 -1.62866517e+03 -1.51913790e+02
  -2.57032170e+03]
 [-8.25851188e+01 -1.10236866e+03 -2.28192552e+03 -2.09413374e+02
  -1.44428223e+03]
 ...
 [-4.56321778e+01 -1.13917070e+03 -2.15497794e+03 -3.33834825e+02
  -1.74570238e+03]
 [ 1.75834461e+01 -1.27676884e+03 -2.13685201e+03 -2.99596499e+02
  -1.64784858e+03]
 [-1.00868117e+00 -1.12461198e+03 -1.95908243e+03 -2.95509708e+02
  -1.43966826e+03]]
```

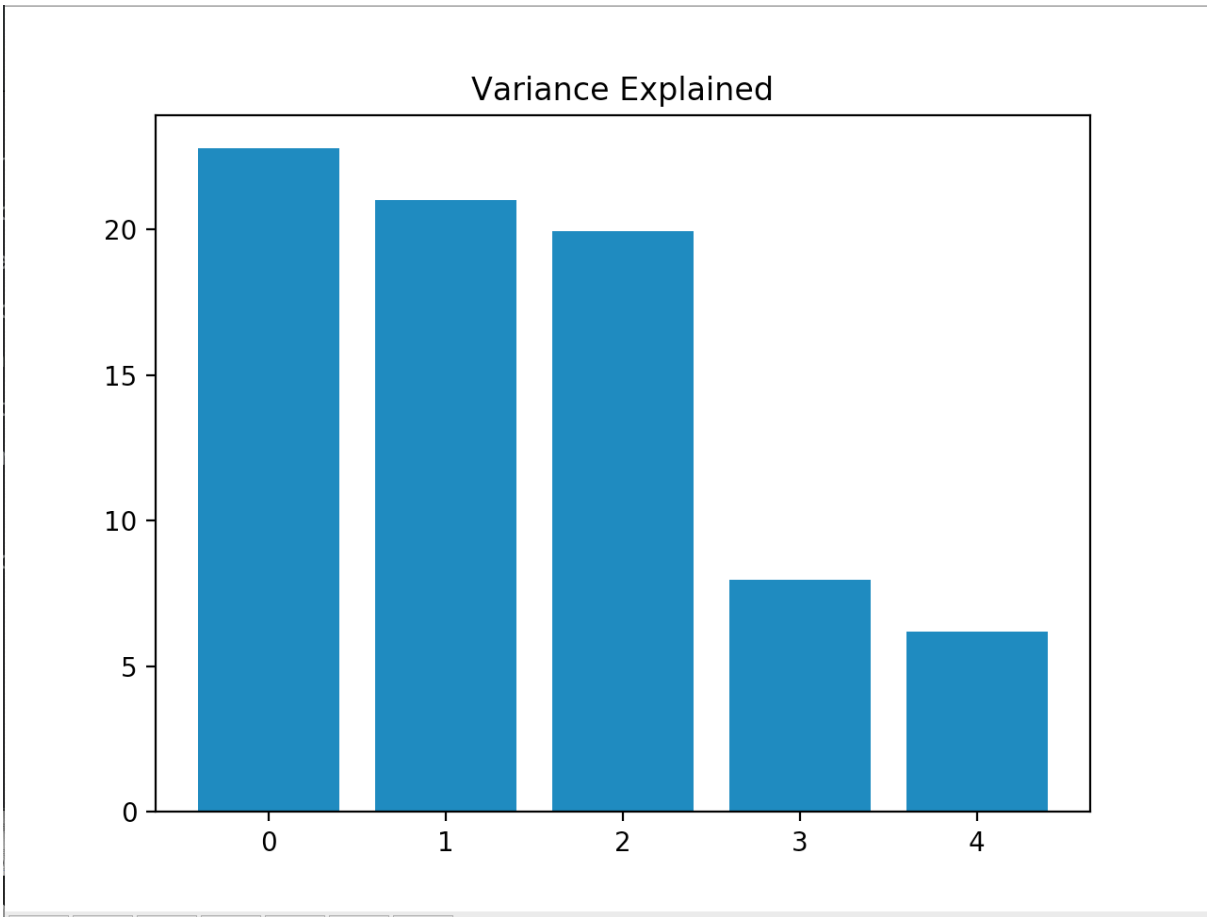
After PCA Feature Matrix

```
Eigen Values
[ 2.27856575e+01+0.00000000e+00j  2.10281616e+01+0.00000000e+00j
  1.99594726e+01+0.00000000e+00j  7.97569064e+00+0.00000000e+00j
  6.20889286e+00+0.00000000e+00j  5.24302489e+00+0.00000000e+00j
  4.61385351e+00+0.00000000e+00j  2.08421293e+00+0.00000000e+00j
  2.04938689e+00+0.00000000e+00j  1.70124361e+00+0.00000000e+00j
  1.47425700e+00+0.00000000e+00j  1.09454891e+00+0.00000000e+00j
  8.36072974e-01+0.00000000e+00j  6.55673919e-01+0.00000000e+00j
  5.85027394e-01+0.00000000e+00j  4.16573289e-01+0.00000000e+00j
  2.90208137e-01+0.00000000e+00j  2.67912472e-01+0.00000000e+00j
  2.17365635e-01+0.00000000e+00j  2.02759285e-01+0.00000000e+00j
  1.77622095e-01+0.00000000e+00j  1.63686234e-01+0.00000000e+00j
  1.34745832e-01+0.00000000e+00j  1.24823481e-01+0.00000000e+00j
  1.18963474e-01+0.00000000e+00j  1.03208867e-01+0.00000000e+00j
  9.22971557e-02+0.00000000e+00j  8.62162048e-02+0.00000000e+00j
  8.55475231e-02+0.00000000e+00j  6.66104715e-02+0.00000000e+00j
  6.44234946e-02+0.00000000e+00j  5.82530226e-02+0.00000000e+00j
  5.59087872e-02+0.00000000e+00j  4.46825512e-02+0.00000000e+00j
  4.40990462e-02+0.00000000e+00j  3.69839354e-02+0.00000000e+00j
  3.42893103e-02+0.00000000e+00j  3.31856394e-02+0.00000000e+00j
  2.87878371e-02+0.00000000e+00j  2.65896866e-02+0.00000000e+00j
  2.45305942e-02+0.00000000e+00j  2.13884952e-02+0.00000000e+00j
  1.98200483e-02+0.00000000e+00j  1.77984062e-02+0.00000000e+00j
  1.59443221e-02+0.00000000e+00j  1.13411810e-02+0.00000000e+00j
  1.03554137e-02+0.00000000e+00j  9.34593603e-03+0.00000000e+00j
  7.90930843e-03+0.00000000e+00j  6.53095870e-03+0.00000000e+00j
  6.19149665e-03+0.00000000e+00j  4.95285313e-03+0.00000000e+00j
```

Eigen Values

```
Eigen Vectors
[[ 2.62587205e-02+0.j -9.37264015e-02+0.j -1.44261130e-01+0.j ...
   3.77691411e-16+0.j -4.11862427e-20+0.j  4.14757615e-20+0.j]
 [ 2.69517079e-02+0.j -9.75575445e-02+0.j -1.43904812e-01+0.j ...
  -3.68132500e-16+0.j -3.47281817e-18+0.j  2.67807580e-18+0.j]
 [ 3.92318704e-02+0.j -1.19525296e-01+0.j -1.30530537e-01+0.j ...
  -1.37275088e-15+0.j -6.43117602e-17+0.j  5.91468413e-17+0.j]
 ...
 [-2.08886970e-02+0.j  2.46975994e-02+0.j  3.58894823e-03+0.j ...
   7.26469715e-06+0.j -7.92196040e-10+0.j  7.97764769e-10+0.j]
 [ 2.96018429e-02+0.j -3.13971117e-02+0.j  7.45571490e-03+0.j ...
   1.53169957e-06+0.j -1.67027801e-10+0.j  1.68201917e-10+0.j]
 [ 2.11783706e-02+0.j -7.41880760e-02+0.j -8.52810231e-02+0.j ...
  -2.64245666e-07+0.j -2.88152926e-11+0.j  2.90178478e-11+0.j]]
```

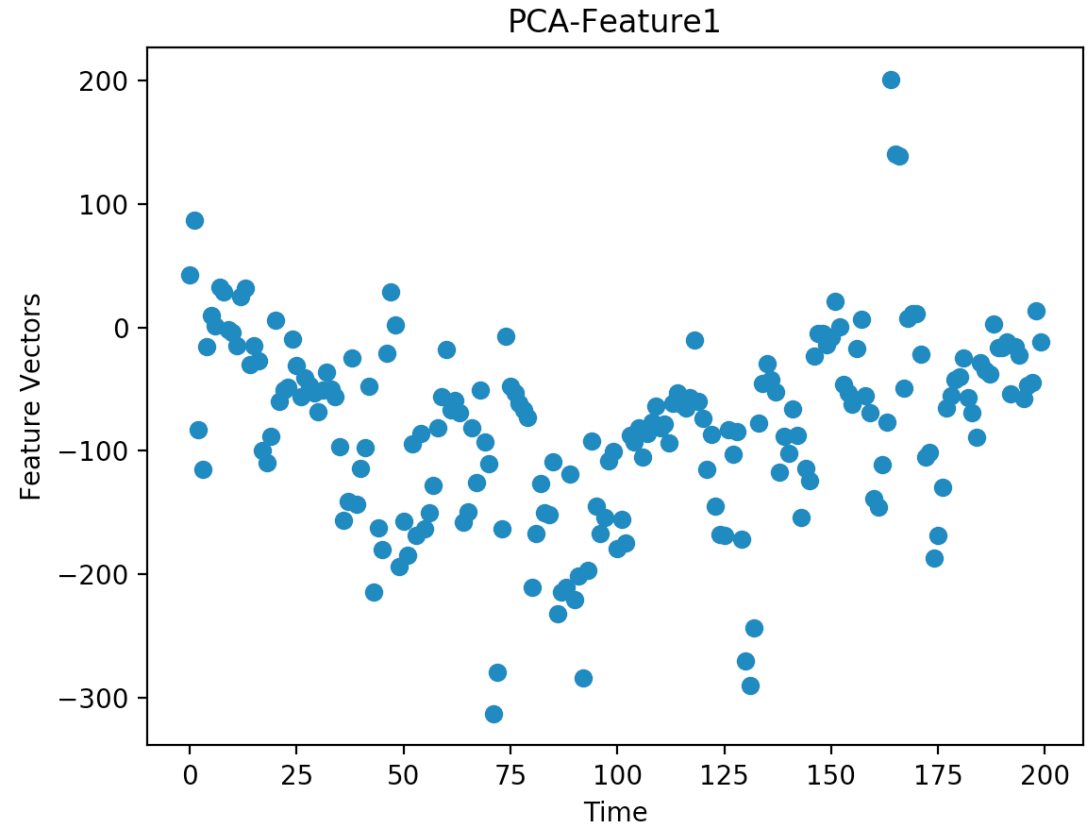
Eigen Vectors

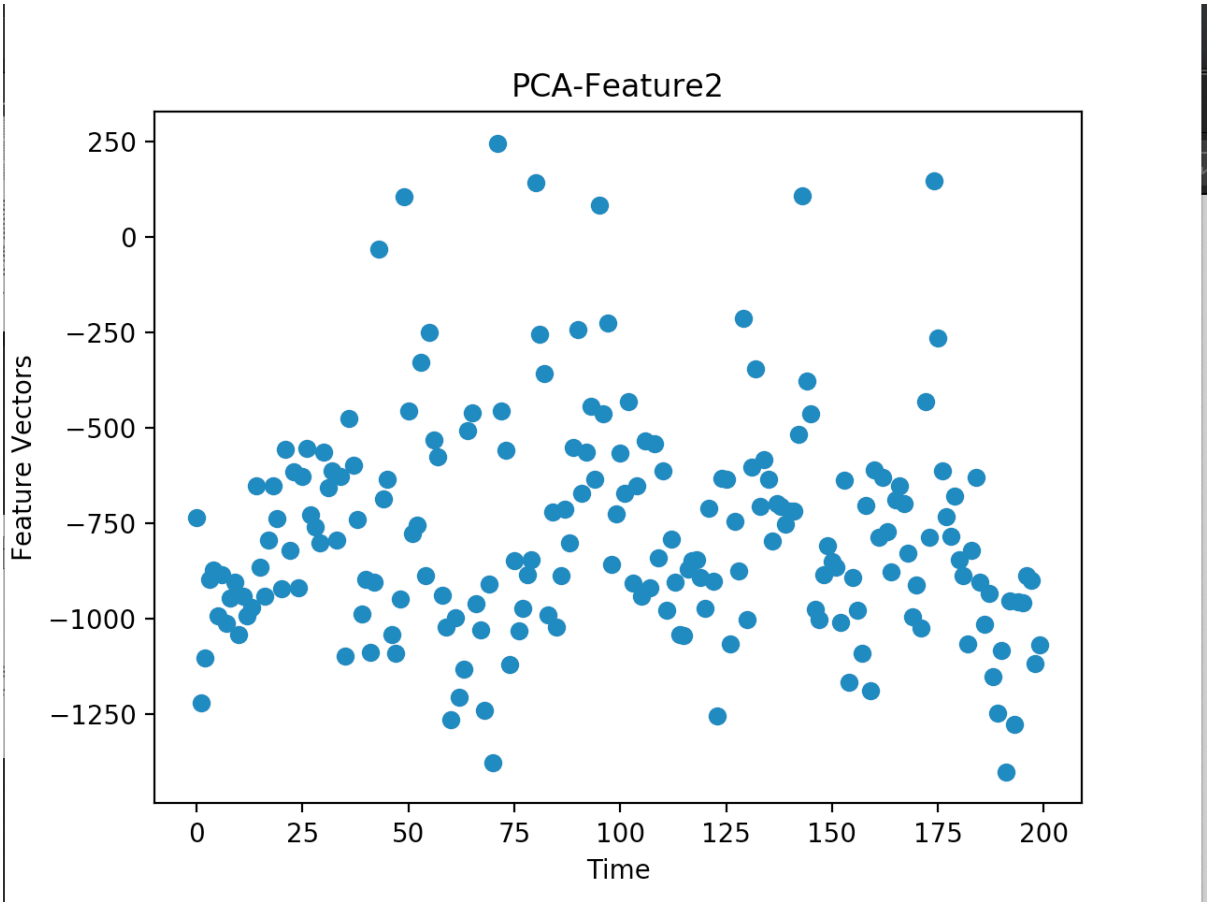


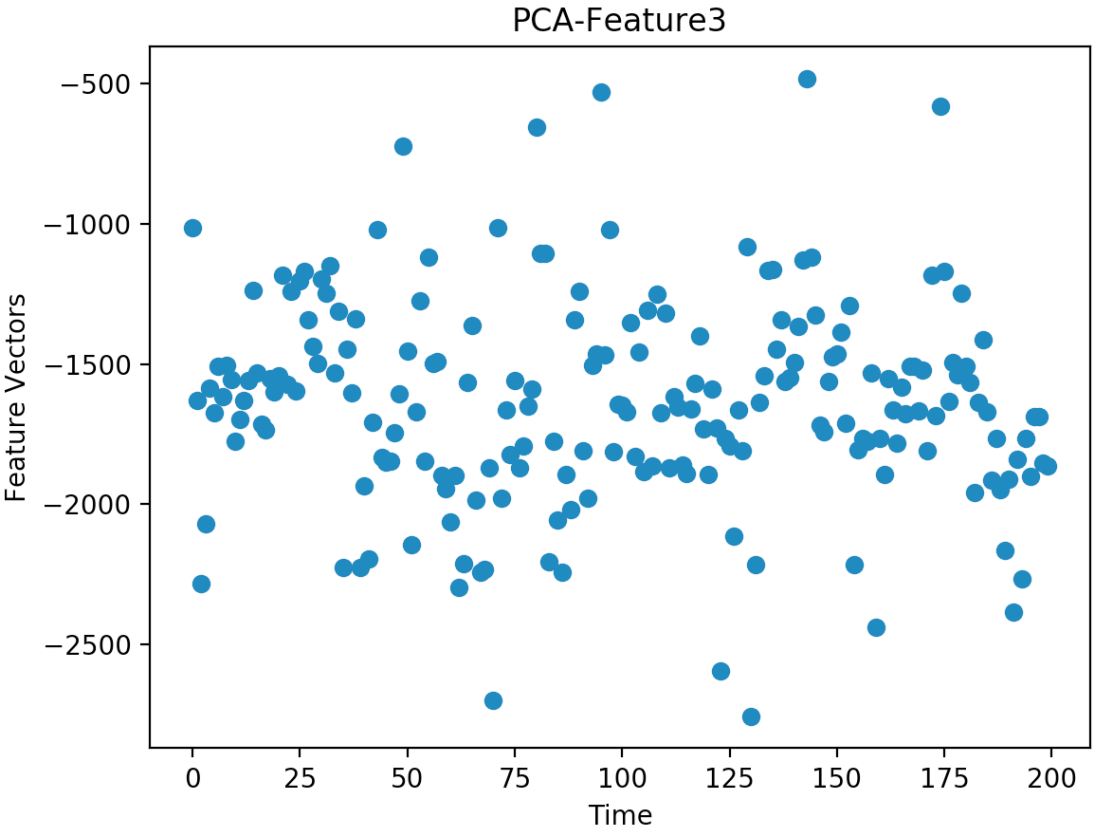
Variance Explained

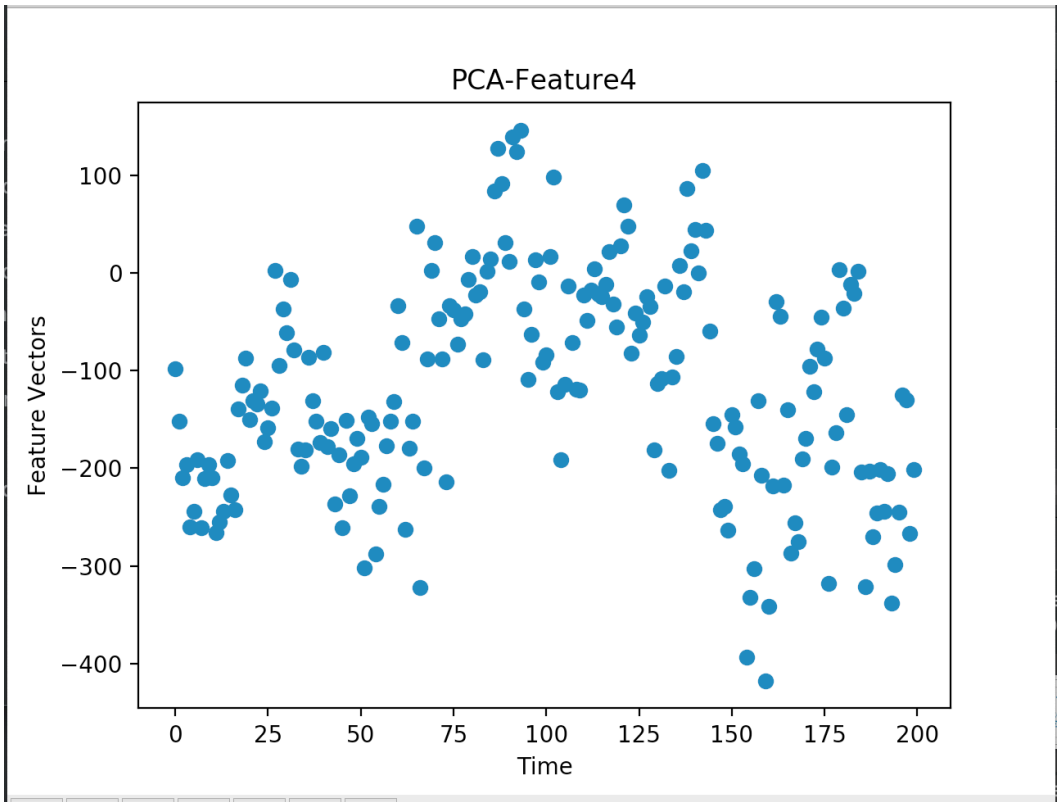
Explained Variance = [22.78565747 21.02816163 19.9594726 7.97569064 6.20889286]

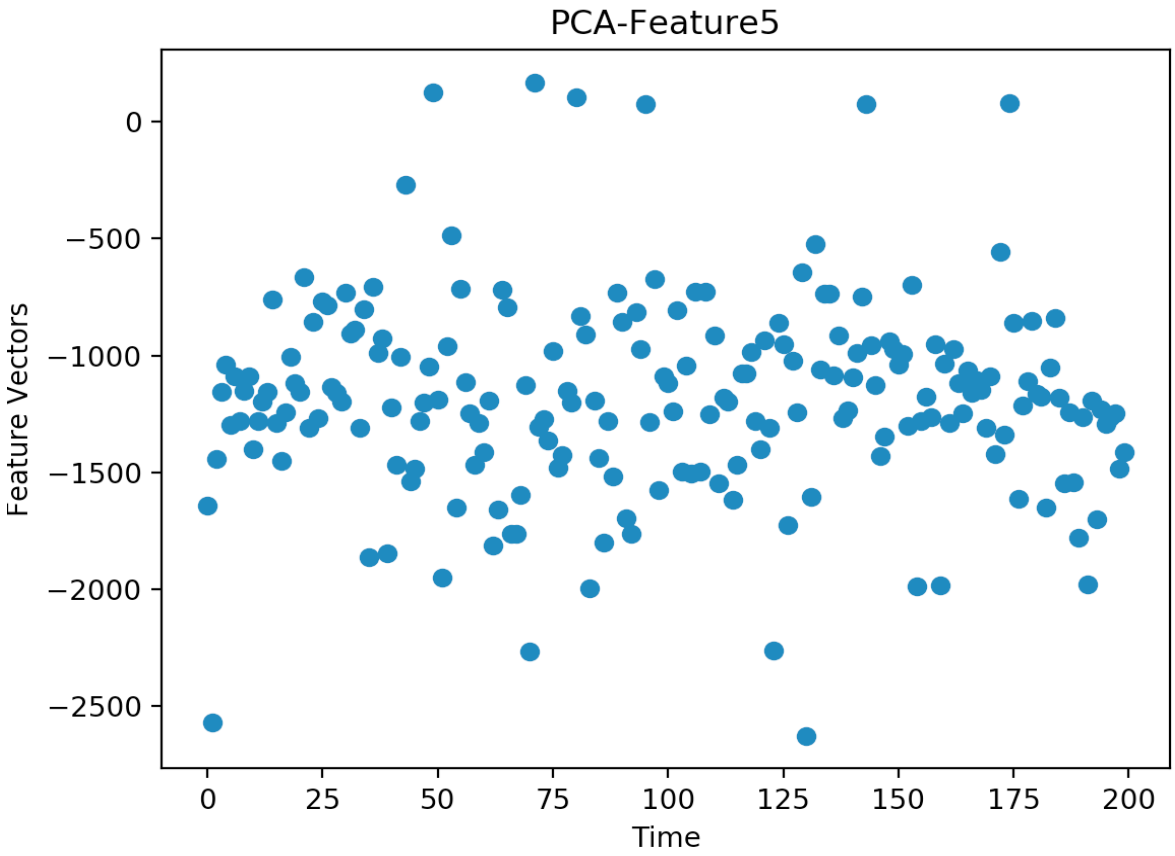
The above bar chart displays the PCA Explained Variance stored in the ‘Explained Variance’ array.

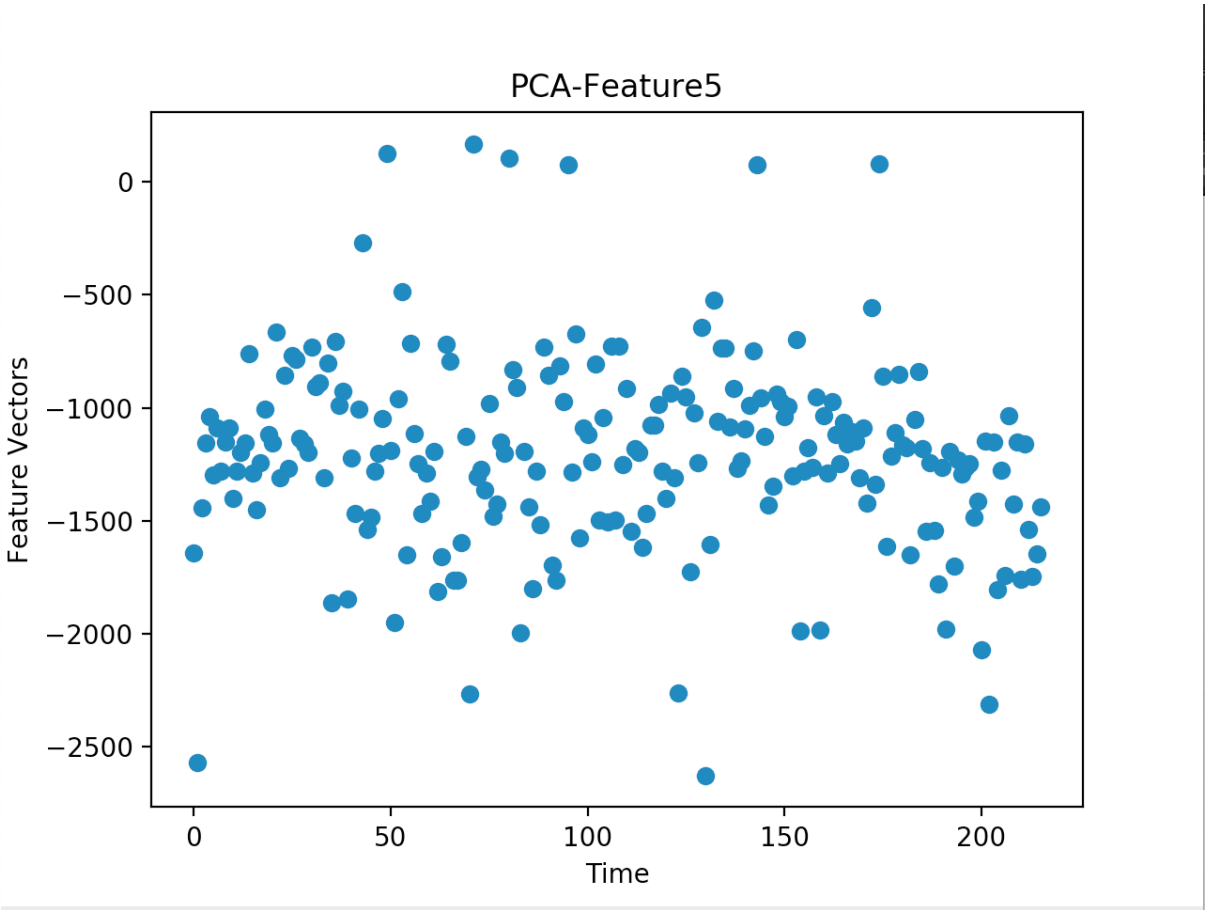

PCA-Feature1
Feature Vectors
Time
200
100
0
−100
−200
−300
0
25
50
75
100
125
150
175
200


PCA-Feature2
Feature Vectors
Time
250
0
−250
−500
−750
−1000
−1250
0
25
50
75
100
125
150
175
200


PCA-Feature3
Feature Vectors
−500
−1000
−1500
−2000
−2500
0
25
50
75
100
125
150
175
200
Time


PCA-Feature4
Feature Vectors
100
0
−100
−200
−300
−400
0
25
50
75
100
125
150
175
200
Time


PCA-Feature5
Feature Vectors
Time
0
−500
−1000
−1500
−2000
−2500
0
25
50
75
100
125
150
175
200



TASK 6

From the bar chart, the variance of different principal component can be clearly seen. First feature corresponds to about 22% followed by second best feature chosen as 21% and so on. PCA picks top k elements as these are the most important features as these new features have more variance than others. PCA picks the features in such a way that the variance in these features will be the largest and the features will be independent of each other. As eigenvectors are perpendicular to each other. The Eigen vectors are used in the principal component analysis because the features extracted from PCA are independent of each other. It is used to reduce higher dimensionality features into lower dimensionality features.
The above graph corresponds to the eigen vectors formed using eigen values and are displayed from the most significant to least significant (PCA 1 being the most significant and PCA5 being the least)

REFERENCES:
1. https://docs.scipy.org/doc/scipy/reference/generated/scipy.fftpack.fft.html
2. https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/interquartile-range/
3. https://docs.scipy.org/doc/numpy/reference/generated/numpy.polyfit.html
4. https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html