Third obligatory assignment

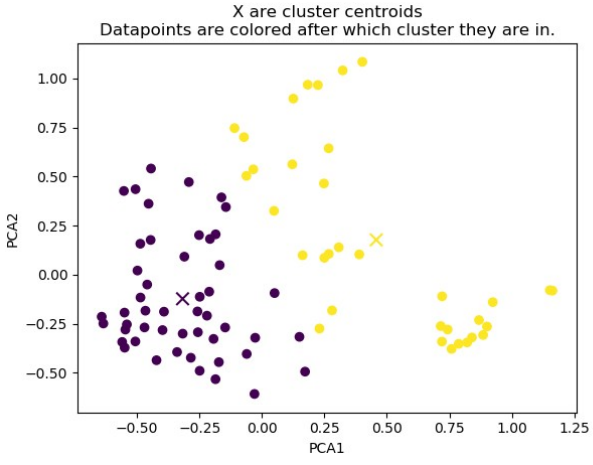
I began with modifying the data set “Engelsberger_short” slightly. I shifted everything up a row so that the column headers would load correctly into an pandas dataframe object. I removed the header “relative peptide intensity levels (time)” and pushed up “0-3-5-10-30” to be the column names. I assume this is refering to minutes that have passed.

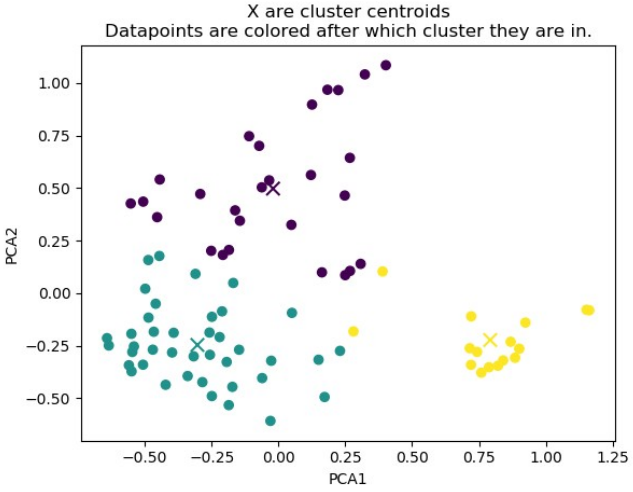
K-means and Gaussian Mixture will not work with missing values. The dataset has both NaN and just pure empty slots. Therefore the next step is to either remove missing values or impute some “dummy” values. I will try out both strategies and see how it turns out. I have commented the code heavily, so please see source code for more information on this.

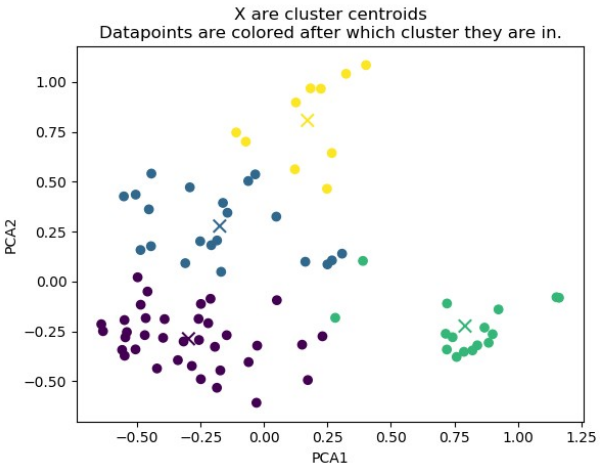
I have now implemented Principal component analysis on the data in order to visualize it in 2 dimensions.

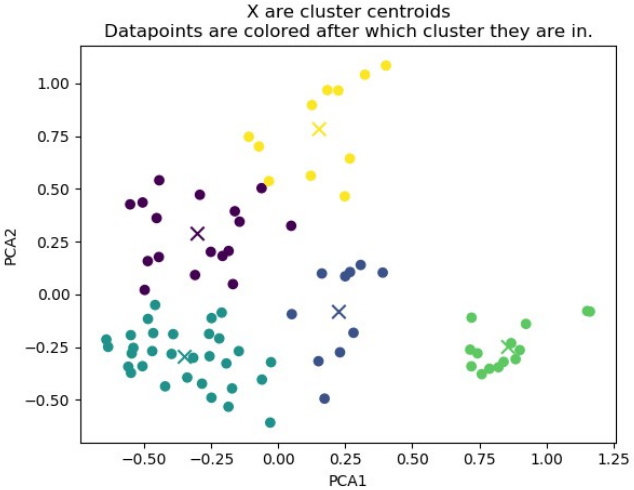
K-Means

I will show a series of plots where the cluster number is set from 2 til 5. Dataset is EdRemovedNaN. OBS: I just realized that I did things in the wrong order. The plots down below are the result of first doing a PCA and then doing the K-Means on only those 2 dimensions. I will leave them in though because they gave a very clean clustering.

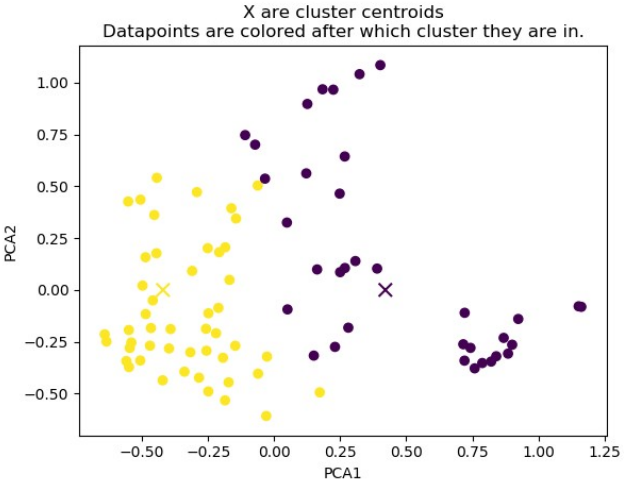


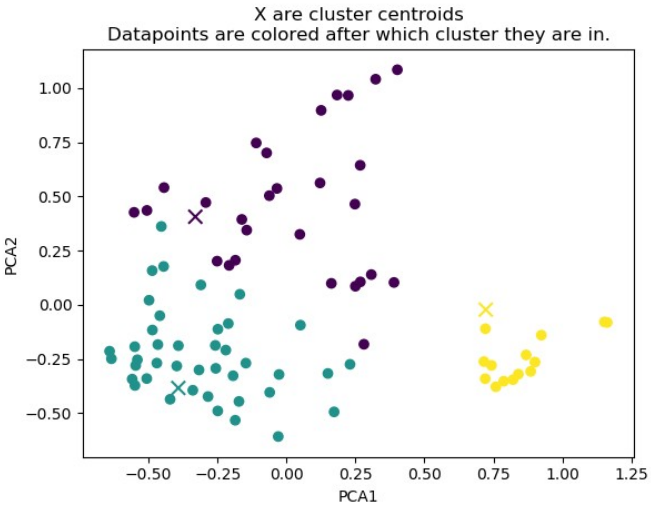


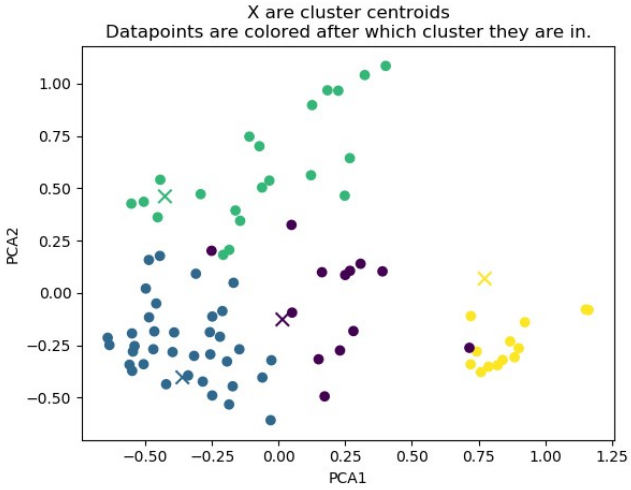


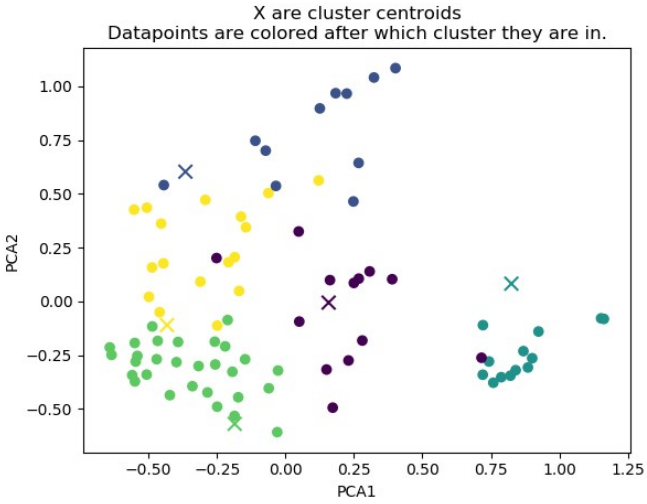


Here are the same plots but with K-means done first(with all 5 dimensions) and then PCA to visualize in 2d:

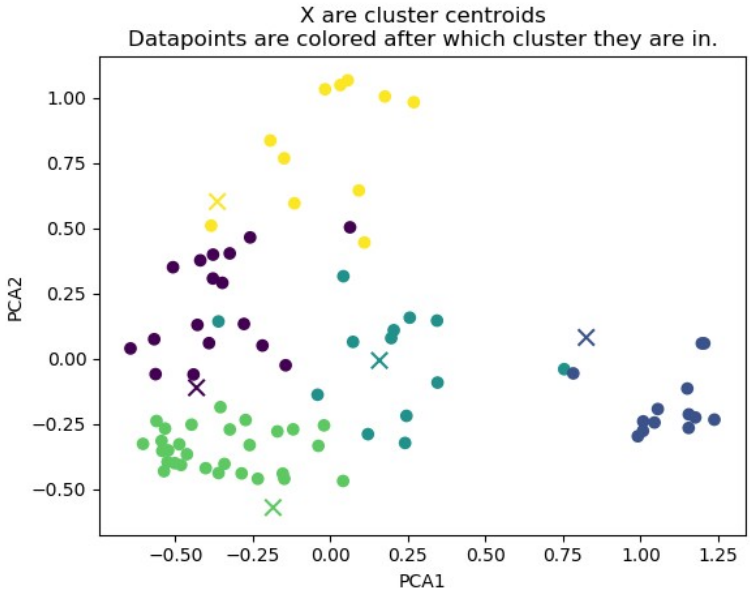






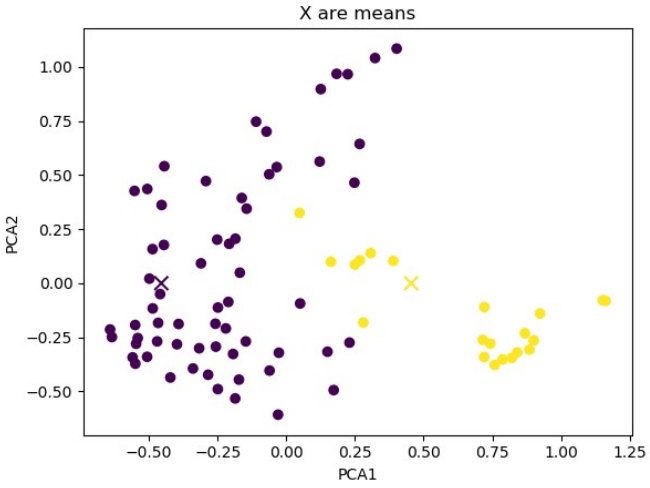


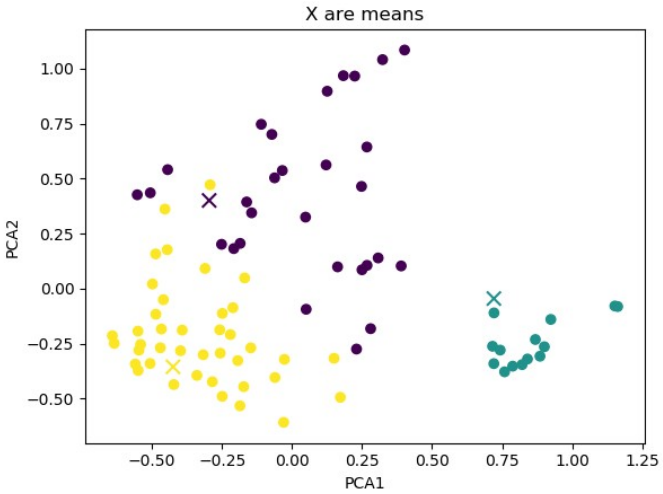
I am very unsure about if I should pass the original data(dataframe) into PCA or if I am supposed to do: newData = km.transform(dataframe), and then pass newData into PCA and then visualize it? I made the decision to stick to original dataframe. But here is an plot demonstrating what happens if I use newData instead(Same as last picture in last series of plots, 5 clusters):

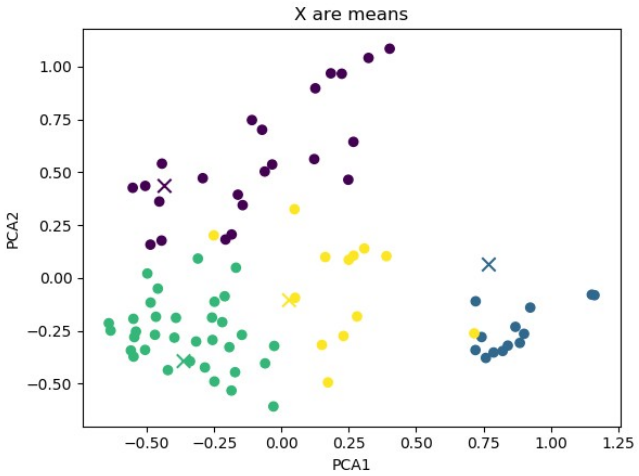


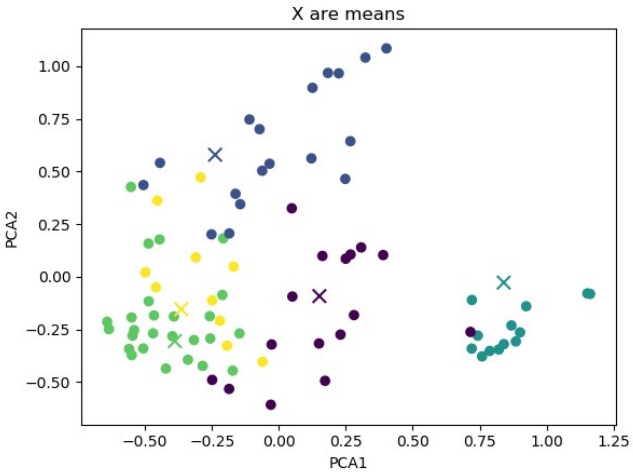
Gaussian mixture

Here I will follow the same order as in K-means section up above. Dataset used is EdRemovedNaN and covariance type is “full”.

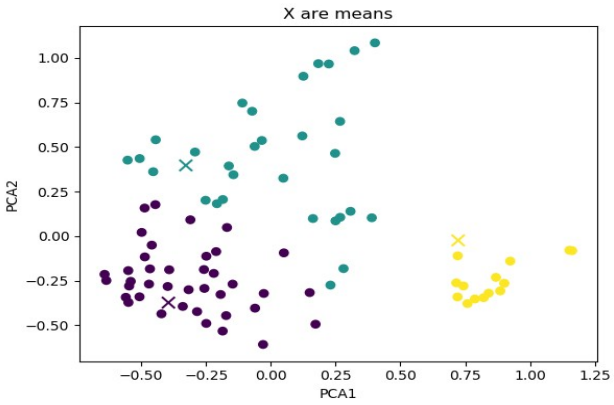




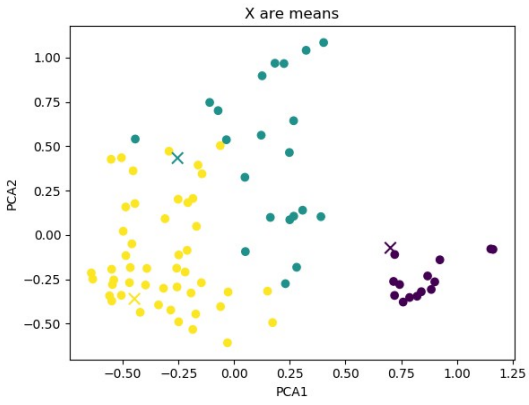




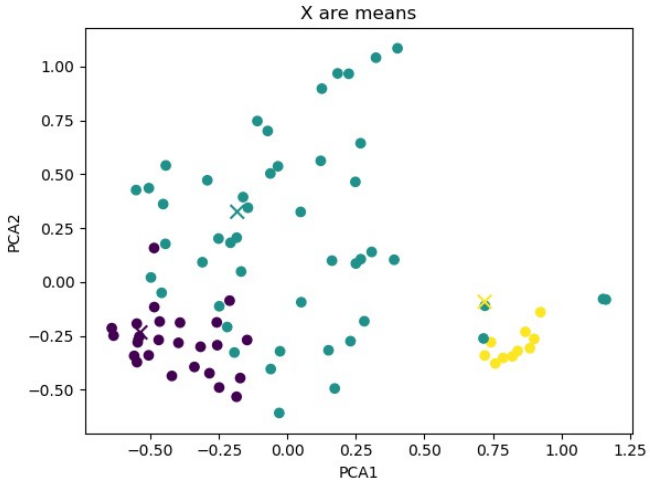
When the components/cluster gets to size 5 it gets really messy. It seems to me that 3 clusters definitely is the best in both K-means and Gaussian mixture. Lets try out different covariance types with 3 components:



Diagonal

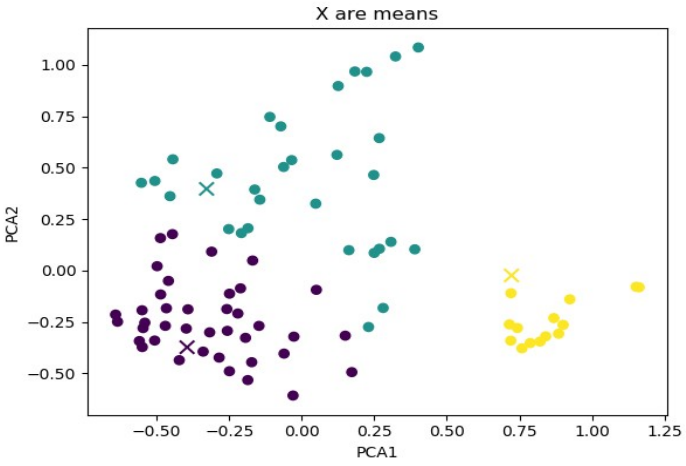


Tied

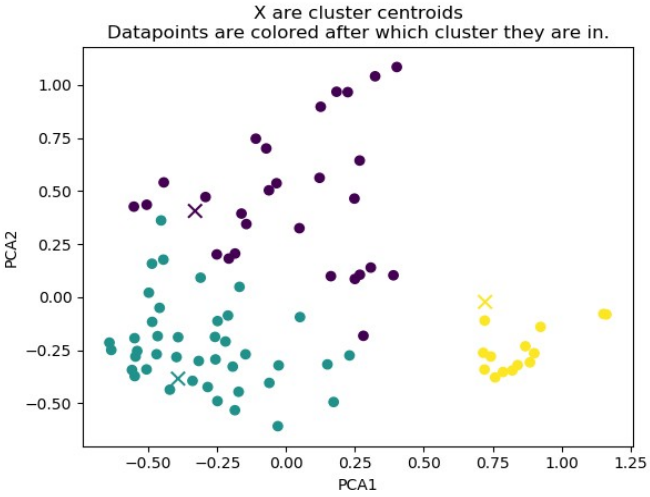


Spherical

Of these 4 types, diagonal seems to be the best one. To end this I am going to compare the best type of Gaussian mixture with the comparable K-means plot side-by-side:



Gaussian mixture(diagonal)



\- *K-means*

They seem to be almost identical. Although it seems that Gaussian mixture has a bit more defined borders.