Deadline 7th of May, 2023, midnight, submit your code and any plots to LMS.

For k-means, we will be working with three datasets in this assignment:

1. Glass: 214 instances, 9 features and 6 classes.

2. Ecoli: 327 instances, 7 features and 5 classes.

3. Yeast: 1479 instances, 8 features and 9 classes.

All the datasets are multi-class classification datasets i.e., have more than two classes.

**Evaluation of clustering asssignments**

One way to evaluate clustering is to calculate adjusted rand index. See: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted\_rand\_score.html

**K-means clustering**

You will use the k-means clustering algorithm to cluster all three datasets. In order to have a suitable clustering you would need to find the number of clusters (k) that produce the best clustering. Try different value of k starting from 2 to number of classes + 2 for each dataset. For instance, for dermatology try values between 2 and 8.

For each dataset provide the plot of the sum of squared error value (SSE) vs k (number of clusters). What do you observe?

Also, calculate the silhouette index for each k and choose k based on silhouette index. Report the SSE value with this k.

Set the number of clusters equal to the number of classes for each dataset and run the k-means algorithm. List the resulting SSE for each dataset in a table.

Compare the adjusted rand index values when k is chosen according to the silhouette index and when k is chosen based on the known number of classes. If the number of clusterns chosen by silhouette score is the same as the known number of classes, then you just need to report a single adjusted rand index value.

**Gaussian mixture models**

Read the simulatedx.csv which has 500 two dimensional data points. The corresponding class labels are in simulatedy.xsv.

Run Gaussian mixture models with scikit learn, try different covariance matrices and different number of components. Follow the procedure here: <https://scikit-learn.org/stable/auto_examples/mixture/plot_gmm_selection.html#sphx-glr-auto-examples-mixture-plot-gmm-selection-py>

Plot the best model at the end.