**FIT5201-Assessment 2 – Report**

**Part A: Document Clustering**

1. Derive **Expectation** and **Maximization** steps of the hard-EM algorithm for Document Clustering.

In Expectation Minimization algorithm for Document Clustering, we need to find the estimation of clusters for each document and maximize the parameters in the process that best describe the cluster assignment for a document.

We treat each document as a set of words in that document irrespective of their order. Also, we assume that all the words present in a document are part of dictionary .

Model parameters for this model are mixing parameter for each cluster and set of word proportions representing word proportions for each cluster . We represent set of all the model parameters as

**Expectation Step:**

1. For each document and for each cluster calculate the responsibility factor. That is, the posterior probability ,of document belonging to cluster given parameters **.**
2. In this step we do hard assignment for the document to cluster having maximum posterior probability among all clusters . We set forto cluster having maximum posterior probability and setfor other clusters.

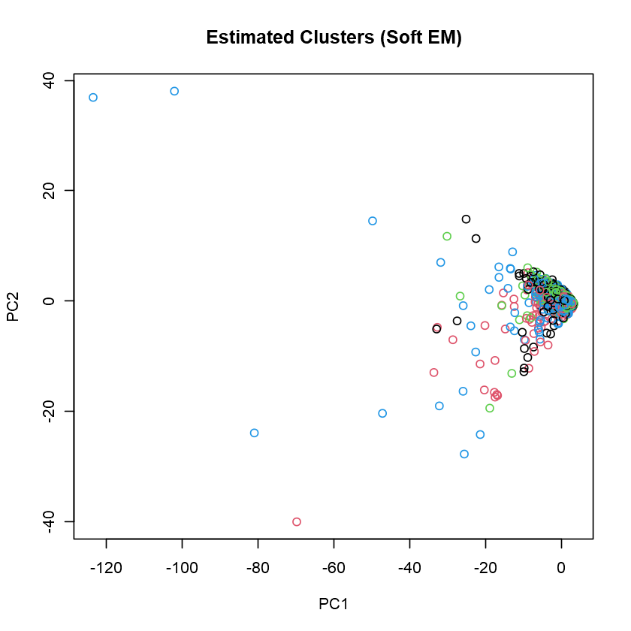
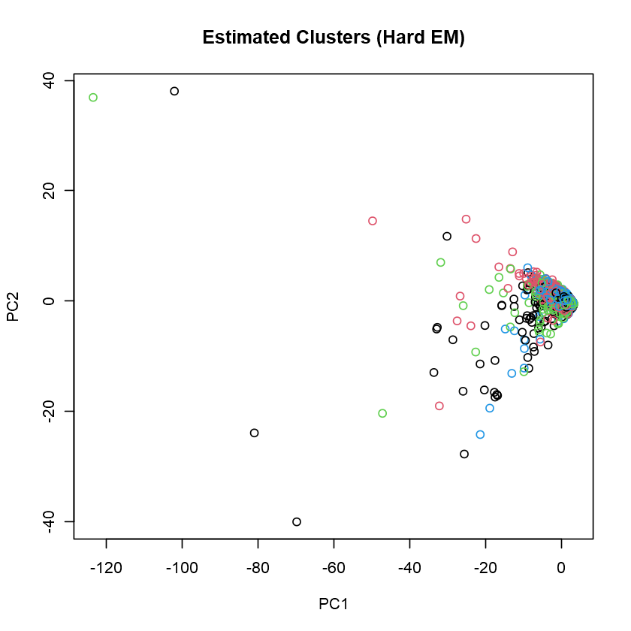
**Maximization Step:**

Based on the results of our expectation step we obtain posterior probabilities for each document and for each cluster. Now in Maximization step we will try to maximize the likelihood by updating our parameters

1. Mixing parameter as,

1. Word proportion parameters for each clusteras,

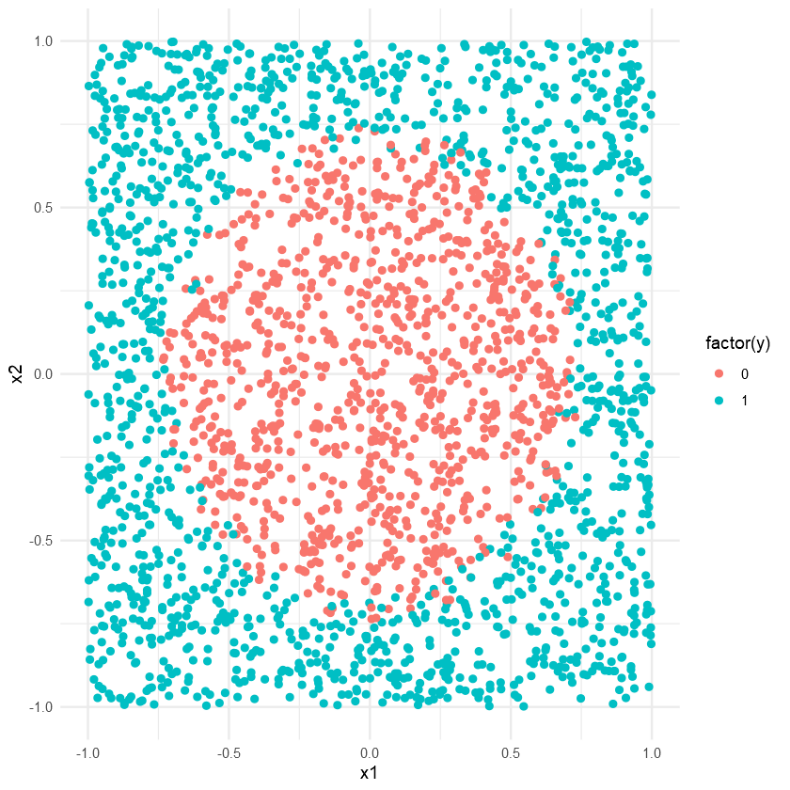
1. Perform a PCA on the clustering that you get based on the hard-EM and soft-EM algorithms. Report how and why the hard and soft-EM are different.



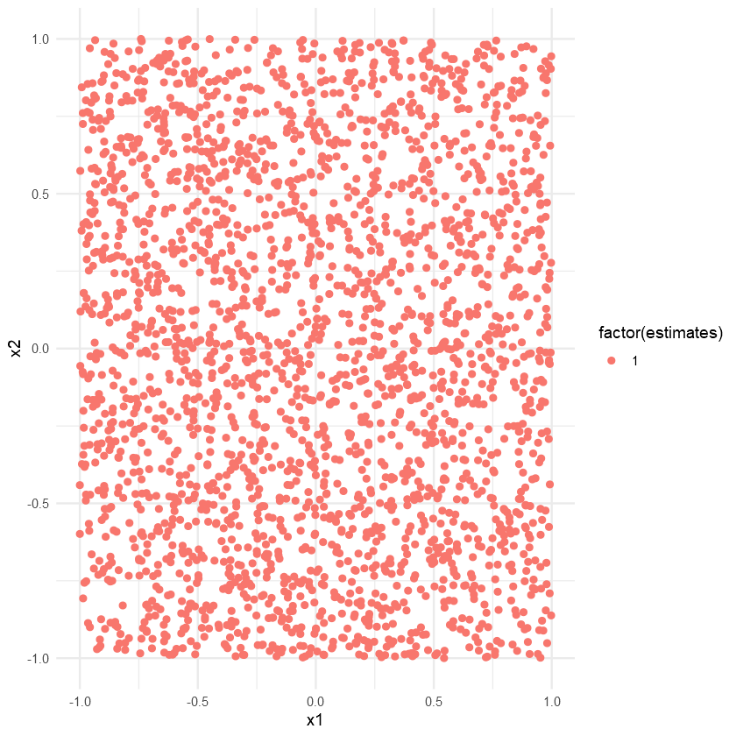
* Hard EM assigns fix cluster to each document, setting for the cluster. While Soft EM provides posterior probability of assignment of a datapoint to a cluster. Therefore, Soft EM best describes the data when there’s not clear decision for which cluster should data point belong. In such cases. Soft EM provides the partial assignment of a datapoints. Based on this knowledge and graphs given above, we can conclude that boundaries of cluster created with Hard EM has well defined boundaries, whereas clusters created with Soft EM algorithm does not.

**Part B: Neural Networks**

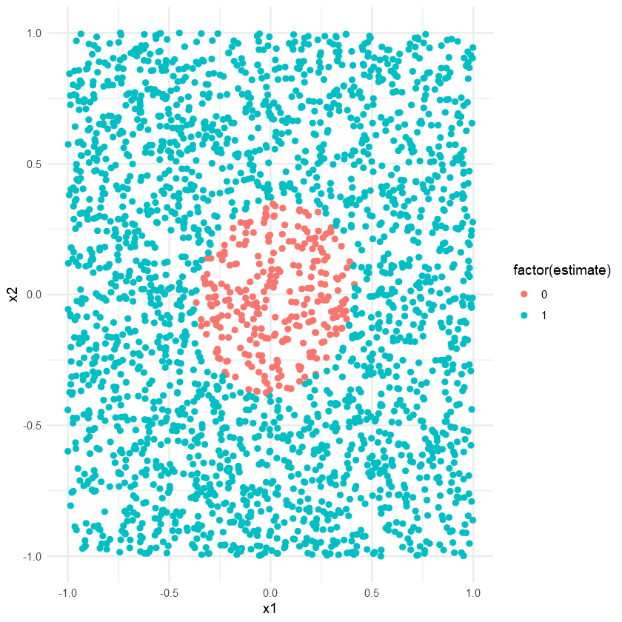
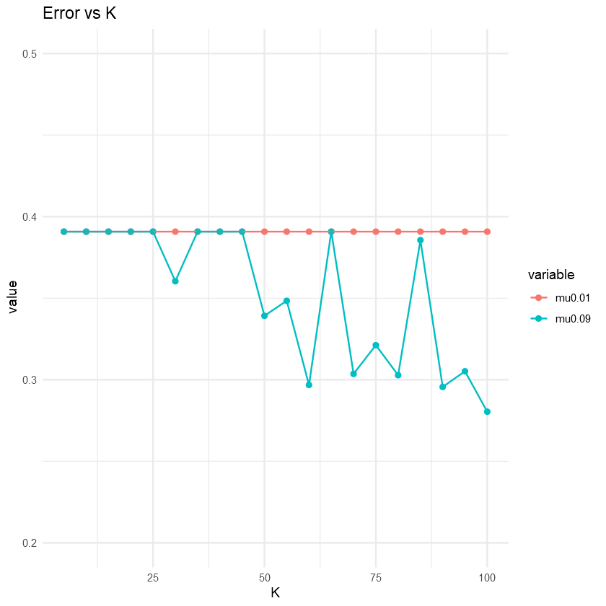
1. **Task2B\_train.csv** and **Task2B\_test.csv** sets, plot the training data with classes are marked with different colours.

****

1. Train two perceptron models on the loaded training data and plot the test data while the points are coloured with their estimated class labels using the best model that you have selected.



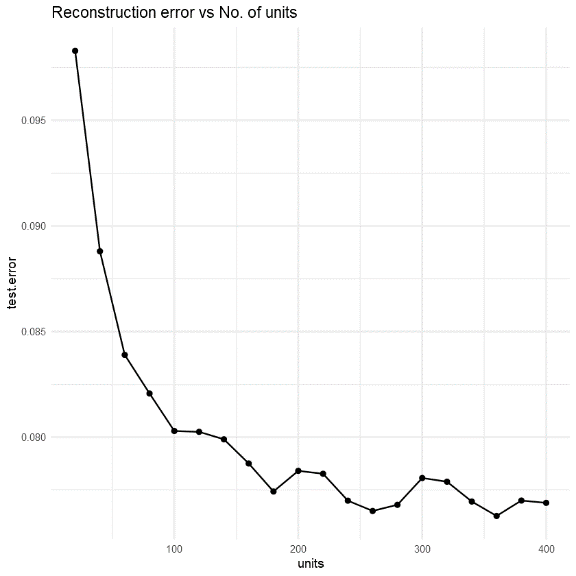
1. For each combination of K {20, 40… 400}, and μ (learning rate) in {0.01, 0.09} run the 3-layer Neural Network. Plot the error for μ 0.01 and 0.09 vs K. Based on this plot, find the best combination of K and μ and the corresponding model, then plot the test data while the points are coloured with their estimated class labels using the best model that you have selected. Explain the reason responsible for such difference between perceptron and a 3-layer NN by comparing the plots.



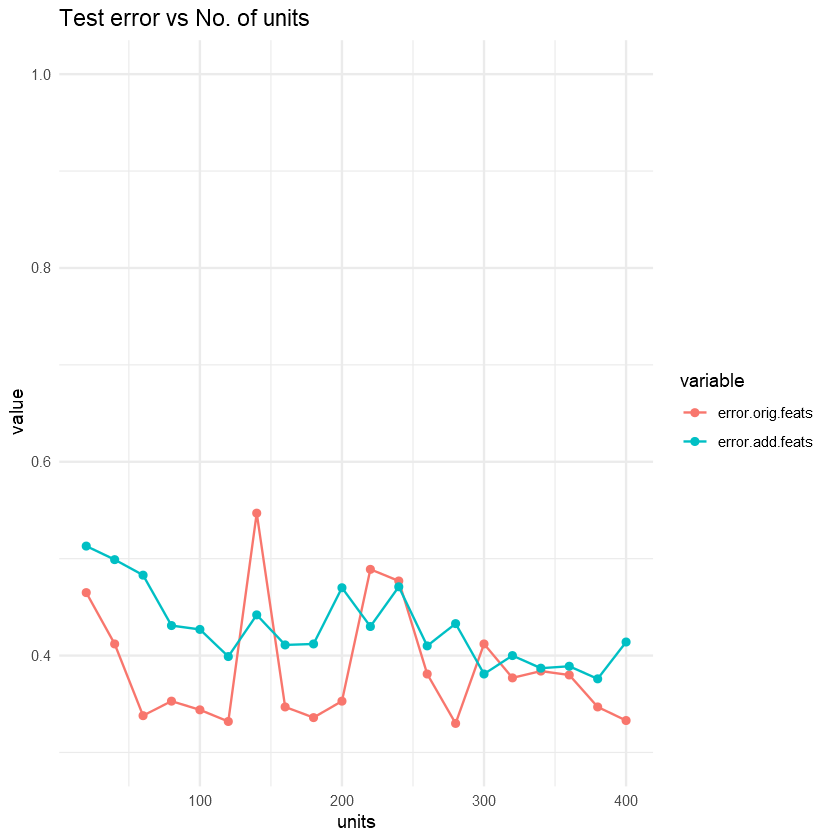
* Perceptron model uses linear equation as linear boundary to separate and classify datapoints. Therefore, it only works well for linearly separable data. As we can see in the plot for Q1, the data is not linearly separable therefore, perceptron model fails to define a boundary between data points. The resultant model is more biased model.
* On the other hand, neural network contains network of interconnected neurons where each neuron represents a non-linear equation. Therefore, neural network can fit well even on complex problems and has ability to learn data where datapoints are separated by non-linear boundary. The increasing number of hidden layers and/or neurons define its ability to learn complex problems.

**Part C: Self-Taught learning**

1. For each model in calculate and record the reconstruction error. Plot these values where the x-axis is the number of units in the middle layer and the y-axis is the reconstruction error.



1. Plot the error rates for the 3-layer neural networks and augmented self-taught networks, while the x-axis is the number of hidden neurons and y-axis is the classification error. Explain how the performance of the 3-layer neural networks and the augmented self-taught networks is different and why they are different or why they are not different, based on the plot.



* Autoencoders, given an unlabelled data learns the features of the data well and can reconstruct the original data well enough. It is assumed that, if you give more data to complex model then it can learn well and provide better prediction. Therefore, we can use adding new additional features learned by the autoencoders with to our original labelled data to train our classifier.
* In the Step 4, we trained our 3-layer NN with original labelled data. As the complexity of model increased, it was able to learn more features and given better predictions. Although increasing model complexity on simple data may lead to overfitting.
* In Step 5, we added features from the autoencoder, to train our model expecting a better performance. But as the training data features increased, our neural 3-layer NN model was not large or deep enough to learn additional complex features and remained underfitted.
* Comparing plots from Step 4 and 5, 3-layer NN learned all the possible features with given complexity, whereas augmented self-taught network given additional complex features didn’t learn well as expected than the simple 3-layer NN and ended up giving similar performance.