AINT351 MACHINE LEARNING 2016
P1.3: KMEANS CLUSTERING

IMPORTANT NOTICE

The final lab journal should contain all your individual lab reports
Submit to DLE: 12th January 2017 (4pm)
You will receive feedback within 20 working days

Please only submit your overall report in PDF format. You **must** include your student number in a header on every page of the report and **also include** your student number **in the document filename**!

Each laboratory report should contain a few pages of explanation (although it can be more than this is necessary) and a set of images showing the plots requested by the individual practical exercises, as well as embedded Matlab code that you developed to implement your solutions.

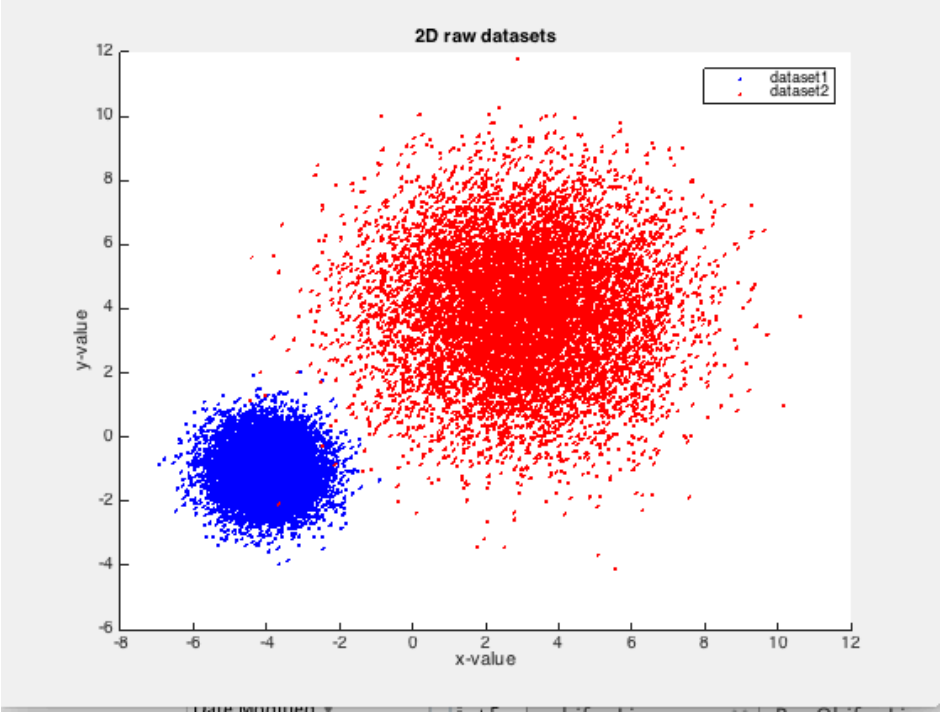
The report must be a **stand-alone document**. Please embed images in-line in the report. In order to show video demonstrations if you have any, please use Internet links to videos that you have been uploaded to YouTube. Please do not submit more than the single PDF file.

1. Generate dataset

- Generate a 2D uncorrelated dataset with slightly overlapping clusters of 10000 points per cluster (20000 points for dataset in total).
  For the first 10000 points use a mean of [-4 -1] and std .deviation 0.75
  For the remaining 10000 points use a mean of [3 4] and std. deviation 2.0
- Plot all the points in the datasets in 2 dimensions using distinct colours for the different sets of points.

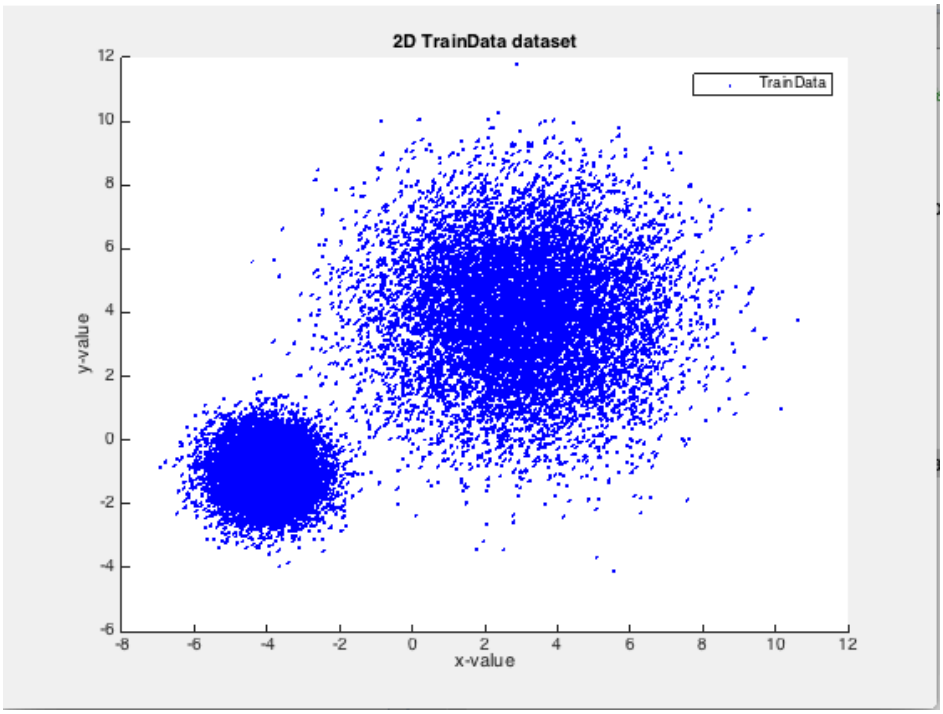
- Include your commented Matlab solution code embedded in the report document

- The results should look something like this:



2. Concatenate the training datasets and plot

- Concatenate the two datasets into a single data let
- HINT: In Matlab you can write
- TrainData = [dataNormal1; dataNormal2;];
- Plot the results. Note all information as to the distribution identity is now no longer directly available so we can only show the points in the same colour.
- Include your commented Matlab solution code embedded in the report document

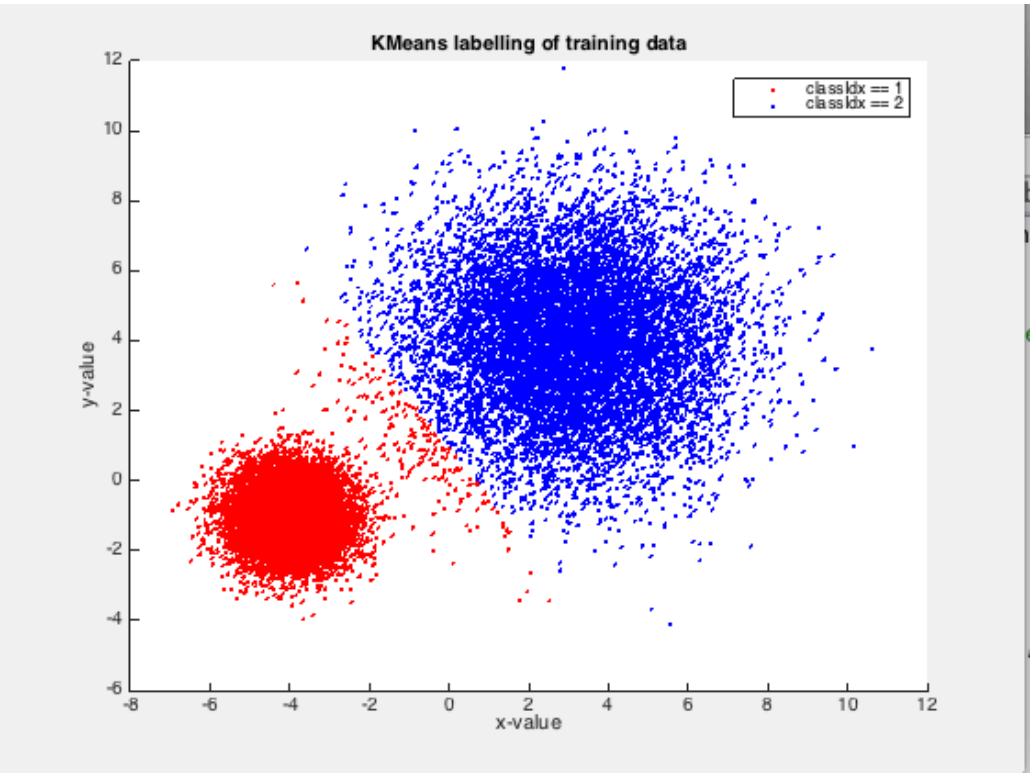


3. Implement KMeans from first principles

- Implement a KMeans algorithm in Matlab yourself from first principles
- DO NOT use a Matlab kmeans function to do this!
- Include your commented Matlab solution code embedded in the report document

4. Run your KMeans function on the data and plot results

- Run your algorithm and label the training data
- How many clusters should you be looking for?
- Plot the results - they should look something like that shown below:
- Include your commented Matlab solution code embedded in the report document



- Interpret the results
- What is wrong with testing on the training data?
- Is there a better way to test the algorithm?
- If you can think of one, run the test again and plot that output too.
- Include your commented Matlab solution code embedded in the report document