Machine Learning, Course Project 2018

Important – Read before starting

- The deadline for completing and submitting your assignment is strictly Wednesday 23rd January 2019 at 18:00.
- VLE will be set up to not accept late submissions meaning that you will get zero marks if your submission if late. Please plan ahead (it is recommended that you upload and verify your work a day before).
- You must complete the project completion form (shown later) and include it in your report. Submissions without the statement of completion will not be considered.
- You must complete a plagiarism declaration form and include it in your report. Submissions without the form will not be considered.
- Projects must be submitted using VLE only. Physical copies or projects (including parts of) sent by email will not be considered.
- For your convenience, a draft and final submission area will be set up in VLE. Only projects submitted in the *final* submission area will be graded. Projects submitted to the draft area are not considered.
- It is suggested that after submitting your project, you redownload it and check it again. It is your responsibility to ensure that your upload is complete, valid, and not corrupted. You can reupload the assignment as many times as you wish within the deadline.
- Your project must be submitted in ZIP format without passwords or encryption. Project submitted in any other archiving format will not be considered.
- The total size of your ZIP file should not exceed 38 megabytes.
- Your submission should include your report in PDF format, your source code, and executable file(s).
- It is expected that you submit a quality report with a proper introduction, discussion, evaluation of your work, and conclusions. Also, make sure you properly cite other people's work that you include in yours (e.g. diagrams, algorithms, etc…).
- In general, I am not concerned with which programming language you use to implement this project. However, unless you develop your artifact in BASIC, C, C++, Objective C, Swift, Go, Pascal, Java, C#, Matlab, or Python, please consult with me to make sure that I can correct it properly.
- This is not a group project.
- Plagiarism will not be tolerated.

Clustering in the Iris Dataset

- Obtain the 'Iris Dataset' from http://archive.ics.uci.edu/ml/datasets/Iris. Instances are defined by four features and each belongs to one of three possible classes.
- You are required to implement the following three artifacts:
    1. Write a program which loads the dataset, asks you to select either one, two, or three features and plots them as a graph. Each instance in the graph (a point) must be coloured/labelled either red, green, or blue depending on which class the instance belongs to. Use this to visually inspect the data and determine if there are any 'obvious' clusters.
    2. Implement the k-Means clustering algorithm (for k=3) to cluster the dataset into its three possible classes. Similar to what you did in (1), allow the user to select one, two, or three features and plot all the instances colouring each according to the cluster it belongs to (red, green, or blue).
    3. Implement the k-NN algorithm. 'Train' using a random % of the data and use the remaining % for evaluation. Identify the optimal splits and which k to use.
- The k-Means clustering, and k-NN implementations must be your own – don't use a library.
- You are required to write a report describing the techniques you use.
- Make sure that your report has a good evaluation section for each of the artifacts you develop.

Statement of completion – MUST be included in your report

Item	Completed (Yes/No/Partial)
Data visualisation (Artifact 1)	
k-Means (Artifact 2)	
k-NN (Artifact 3)	
Evaluation (Artifact 1)	
Evaluation (Artifact 2)	
Evaluation (Artifact 3)	
Overall conclusions	