Machine Learning, Course Project 2019

Important – Read before starting

- The deadline for completing and submitting your assignment is strictly Friday 10th January 2020 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.

Using Support Vector Machines to Classify Hand-Written Digits

- As part of your report, write a technical section about how Support Vector Machines work. Discuss why they are especially useful in dealing with classification problems, and how they differ from other methods like ANNs and kNN.
- Do not implement the SVM algorithm yourself (use a library to do this for you). Python may be a good choice for this project.
- You are required to implement an SVM to recognise (classify) hand-written digits 0-9.
- It is up to you to decide how to split the data set into training/validation/testing sets. You might want to experiment with these parameters.
- I highly recommend the MNIST data set http://yann.lecun.com/exdb/mnist/
- You are expected to experiment with different SVM parameters (e.g. kernels) to find out which work best.
- Make sure that your report has a good evaluation section for any artifacts you develop. Your evaluation has a significant impact on the final grade you will be awarded.
- Provide good instructions about how to run your program.

Statement of completion – MUST be included in your report

Item	**Completed (Yes/No/Partial)**
SVM technical discussion	
A good comparison to alternative methods	
Artifact	
Experimentation with different SVM params	
Experiments and their evaluation	
Overall conclusions	