Machine Learning, Course Project 2017

Important – Read before starting

- The deadline for completing and submitting your assignment is strictly Wednesday 17th January 2017 at 18:00.
- VLE will be set up to not accept late submissions meaning that you will get zero marks if late. Please plan ahead (it is recommended that you try and 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 just in case. It is your responsibility to ensure that your upload is complete, valid, and not corrupted. You can reupload the assignment as may 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.

Spam filtering using Support Vector Machines (SVMs)

- Dataset: a public spam filtering dataset may be obtained from http://spamassassin.apache.org/old/publiccorpus/
- Transform instances (text) into features (feature extraction) using the bag of words model. You are not required to implement a text parser – feel free to use a library.
- Learn about SVMs, and train an SVM on a subset of the dataset above.
  - Choose your training and testing sets wisely.
  - Strategy: train and classify based on the features you extracted.
  - You are not required to implement an SVM from scratch. You may use an SVM library.
- Learn about cross validation, and use it to tune SVM parameters (e.g. the parameters of the SVM kernel(s) you choose).
- Experiment with different SVM kernels.
- Your report should include: an introduction, description of bag of words model, SVMs, your setup and implementation, your experiments and results, and your conclusions.
  - Your experiments should take on the typical: experiment name, setup/parameters, expected results, actual results, and conclusions format.
  - Interesting experiments include: examining the performance w.r.t. different training/validation set sizes, the performance w.r.t. different SVM kernels, etc…

Statement of completion – MUST be included in your report

Item	**Completed (Yes/No/Partial)**
Implementation of bag of words model	
Implemention of SVM	
Used cross validation for parameter tuning	
Experiments and their evaluation	
Overall conclusions	