COMP219 - 2020 - First CA Assignment
Individual coursework
Simple Machine Learning Model

Assessment Information

Assignment Number	1 (of 2)
Weighting	10%
Assignment Circulated	Friday 9 October 2020
Deadline	Friday November 20 2020, 15:00
Submission Mode	Electronic
Learning outcome assessed	2. Ability to choose, compare, and apply suitable basic learning algorithms to simple applications;
Purpose of assessment	To implement machine learning algorithms on a dataset
Marking criteria	The marking scheme can be found in Section 3
Submission necessary in order to satisfy Module requirements?	No
Late Submission Penalty	Standard UoL Policy.

1 Objectives

This assignment requires you to *implement* and *evaluate* one simple machine learning models on two datasets.

2 Requirement and Description

Language and Platform Python (version 3.5 or above). You can use some other libraries available on Python platform, including numpy, scipy, scikit-learn, and matplotlib. If you intend to use libraries other than these, please consult the demonstrator or the lecturer.

Dataset Please use one of the following two datasets, whose information can be found in `https://scikit-learn.org/stable/datasets/index.html`

- Optical recognition of handwritten digits dataset
- RCV1 dataset

Learning Task Classification on the dataset you select.

Learning Model/Algorithm You may choose one learning algorithm from the following list:

- decision tree learning
- naive Bayes
- k nearest neighbor

Assignment Tasks Once you have selected a learning algorithm, you need to implement the following *functionalities* for the Learning Task:

f1 provide the details of the dataset, including the number of instances, the number of features of each instance, and the value range of each feature.

f2 train a machine learning model by calling an algorithm from the machine learning libraries such as scikit-learn, and save the model so that it can be called later. You can save a model with scikit-learn built-in functionality such as `https://scikit-learn.org/stable/modules/model_persistence.html`.

f3 implement a machine learning algorithm by yourself, train a model with the algorithm, and save the model so that it can be called later;

f4 compare the train error and test error of the two models;

f5 enable the user to query the saved models by providing e.g., an index of the test dataset.

Additional Requirements We have *additional requirements* that,

1. the marker can run your code directly, i.e., see the results of functionalities **f1**, **f4**, and **f5** by loading the saved models, without calling the training functionalities **f2** and **f3**, and
2. you need to provide clear instructions on how to train the two models, i.e., run functionalities **f2** and **f3**. The instructions may be e.g., a different command or an easy way of adapting the source code.

Documentation You need to write a proper document

1. detailing how to run your program, including the software dependencies,
2. explaining how the functionalities and additional requirements are implemented, and
3. providing the details of your implementation, including e.g., the meaning of parameters and variables, the idea of your algorithm, etc.

Also, the document needs to follow the guidelines in Note 1 of Section 3.

Submission files Your submission should include the following files:

- a file for source code,
- two files for saved models, and
- a document.

Please see Section 4 for instructions on how to package your submission files.

3 Marking Criteria

The assignment is split in a number of steps. Every step gives you some marks.

Note 1 *At the beginning of the document, please include a check list indicating whether the below marking points have been implemented successfully. The length of the submitted document needs to be within 4 pages (A4 paper, 11pt font size).*

Note 2 *The marking of a functionality will also consider the quality of coding and the quality of documentation. A run-able implementation alone will have up to 50% of the marks.*

functionality f1: 20%

Successfully load the dataset and display the dataset information, including the number of data entries, the number of classes, the number of data entries for each classes, the minimum and maximum values for each feature, and the train dataset and test dataset split.

functionality f2: 20%

Successfully call library functions to train and save a model. There needs to be a corresponding saved model in your submission.

functionality f3: 40%

You have an implementation of an algorithm that is able to train a model. There needs to be a corresponding saved model in your submission. You cannot call any library which has direct implementation of a machine learning algorithm.

functionality f4: 10%

Please output the train and test errors for both models. Each model have 5%.

functionality f5: 10%

Allow users to query the models by changing the input. For example, you can use a variable to represent the index of the test dataset. Each model have 5%.

4 Deadline and Submission Instructions

- Deadline for submitting the first assignment is given at the beginning of this document.
- Please submit all the files in a single compressed file with the filename

  $$"\langle studentnumber\rangle.tar" \quad or \quad "\langle studentnumber\rangle.zip"$$

  For example, "201191838.tar" or "201191838.zip" if your student number is 201191838. Submissions with other filename will not be accepted. Also, in the submission files, please do not include your name.
- Submission is via CANVAS system.

5 Q&A

Q: What if I choose to implement k-NN which does not have a model?

A: You do not have to save a model, but you need to make sure that your program can run in less than 4 minutes (for two algorithms). You can for example take a subset of data to train. Also, you need to explain this situation (i.e., you have less files in your submission package), including how long it takes for the program to run on your own machine.

Q: Shall I download the dataset from the original place e.g., UCI repository, or sklearn package ?

A: Please use sklearn. Although the original repository include more data samples, we are focused on the algorithm and would like to take an easy, and consistent, way of loading data.

Q: Will I be penalised if the accuracy of my algorithm is not good?

A: Accuracy is not our major concern, and our marking will not be affected by the accuracy. However, if your algorithm is not correct, you will get less marks.

Q: Can I use a previous version of sklearn ?

A: We recommend everyone to use the up-to-date version of sklearn (version 0.21 as at October, 2019). However, if you have difficulty using this version, please clearly describe in your document which version you are working with and the reason why you cannot use the new version.

Q: I do not know how to load dataset. Can you help?

A: sklearn has built-in loading function for you to call directly. Please Google to learn this. :)