MSC-BDT5002, Fall 2021
Knowledge Discovery and Data Mining
Assignment 2
Deadline: October 22th, 2021 11:59pm

Submission Guidelines

1. Attachments should be named in the format of: **A2_itsc_stuid.zip** which includes
   - A2_itsc_stuid_report.pdf/.docx: Please put all your reports in this file. (Attachments should be original .pdf or .docx, NOT compressed)
   - A2_itsc_stuid_code.zip: The zip file contains all your source codes for the assignment.
   - A2_itsc_stuid_Q1_code: this is a folder that should contain all your source code for Q1.
   - A2_itsc_stuid_Q2_code: same as above.
2. TA will check your source code carefully, so your code MUST be executable, your result MUST be reproducible.
3. For programming language, in principle, python is preferred.
4. Your grade will be based on the correctness, efficiency and clarity.
5. Please check carefully before submitting to avoid multiple submissions.
6. Submissions after the deadline or not following the rules above are **NOT** accepted.
7. The email for Q&A: hxinaa@connect.ust.hk.
8. **Plagiarism will lead to zero points**.

(Please read the guidelines carefully)

1 Comparison of Classifiers (60 marks)

We utilize different classifiers for classification in Assignment 2.

1.1 Data Description

We use the **Wine Quality Data Set** . The statistics of dataset is shown in Table 1.

	Size	***Features***
Train	3918	12
Test	980	12

Table 1: Data Statistics

1.2 Comparison of Classifiers

You are required to implement the following classifiers and compare the performance achieved by different classifiers.

- **Decision Tree**
  You should build decision trees on the dataset in terms of *entropy* and *gini* criterions. For each criterion, you should set the depth as {5,10,15,20} respectively. You need to compare the performance (*accuracy, precision, recall, f1 score and training time*) and give a brief discussion. (**30 marks**)
- **KNN, Random Forest**
  Apply three different classifiers KNN and Random Forest on the dataset. For each classifier, evaluate the performance (*accuracy, precision, recall, f1 score and training time*). You are required to compare the performance of different classifiers and give a brief discussion. (**30 marks**)

In your report, for each sub-question, you are required to provide a table (just like Table 2 as shown below) followed by a brief discussion.

Classifier	**Accuracy**	**Precision**	**Recall**	**F1**	**Training Time**

Table 2: Example of summary table

1.3 Note

- Note that all the codes should be compilable and well-commented (provide enough comments for each key line of code), otherwise you may lose some marks if the code is very difficult to understand.
- In this question, you are **FREE** to use different Python libraries for implementation.

- The problem in this assignment is a multi-class classification. All the metrics (*accuracy, precision, recall, f1 score and training time*)should be the average results over the entire test set.
- For classifiers without specified parameters (like KNN, Random Forest), you are free to adjust parameters by yourself.

2 Implementation of Adaboost (40 marks)

2.1 Data Description

Table 3 shows the training dataset. It consists of 10 data and 2 labels.

#	1	2	3	4	5	6	7	8	9	10
$\boldsymbol{x}$	0	1	2	3	4	5	6	7	8	9
$\boldsymbol{y}$	1	1	-1	-1	-1	1	1	-1	-1	1

Table 3: Training Dataset

2.2 Implementation

We assume the weak classifier is produced by $x < v$ or $x > v$ where v is the threshold and makes the classifier get the best accuracy on the dataset. You should implement the **AdaBoost** algorithm to learn **a strong classifier**.

2.3 Note

- Adaboost library is **NOT** allowed to be used. You need to implement it manually and submit your code.
- You should also report the final expression of the strong classifier, such as $C^*(x) = sign[\alpha_1 C_1(x) + \alpha_2 C_2(x) + \alpha_3 C_3(x) + \ldots]$, where $C_i(x)$ is the base classifier and α_i is the weight of base classifier. You are also required to describe each basic classifier in detail.
- For simplicity, the threshold v should be the multiple of 0.5, i.e., v%0.5==0. For example, you can set v as 2, 2.5, or 3, but you cannot set v as 2.1.
- *sign* function: https://en.wikipedia.org/wiki/Sign_function