UCCD2063 Artificial Intelligence Techniques
Practical Assignment

May 2020

1. General Instructions

The general guideline for this assignment is as follows:

1. The total marks for the assignment is 100 and contributes **30%** to the total grade.
2. This is a **group assignment.** Each group should have **2-3 students** (no need to be in the same practical group).
3. Each group has to submit ONE **Zip file** containing the **report (.pdf)**, **code (.ipynb)**, **dataset (.csv or other)** and other related materials to WBLE before the deadline.
4. The deadline for the submission of reports is on **September 7 (Monday) 5pm**. Late submissions may be subjected to a penalty of up to 30% of the total assignment mark.
5. Evidence of plagiarism will be taken seriously and University regulations will be applied fully to such cases, in addition to ZERO marks being awarded to all parties involved.

2. Introduction

In this assignment, you are required to implement a system that performs **classification** for a particular application domain. You are allowed to work on any real-world problems involving classification. The problem and the dataset selected should contain certain complexity that requires data preprocessing, model selection and tuning of learning algorithm in order to achieve better classification accuracy.

3. Requirements

Followings are requirements as well as suggestions for things to consider in your assignment.

3.1 Dataset

You are free to use existing dataset or to collect your own data. One valuable resource to look for datasets is the **UCI machine learning repository** (https://archive.ics.uci.edu/ml/index.php) which contains many dataset covering a wide range of classification application.

Remember, your dataset must contain certain **complexity** that requires data preprocessing, model selection and fine-tuning the classification models to achieve better results.

3.2 Experiment

Since the objective of this assignment is for you to learn, your experiment should not only run on one classification algorithm with default setting, but it should also include analysis of your learning algorithm by testing different kind of settings. For example, you can show how changing hyperparameters or training methods affects your classification results, or show the

effect of regularization or learning rate (if any) to the final performance by varying the parameters. You can also show the effect of number of input features to your prediction accuracy and apply appropriate validation and testing steps to verify your results.

3.3 Coding

You must use **Python** and related packages in your implementation. You may implement any learning algorithm of your choice, not limited to those you have learned in class as long as you understand how it works. Higher marks would be allocated for assignments that:

a. Implement learning algorithms not taught in class AND you understand the algorithm that you are using.
b. Implement two or more learning algorithms and compare the performance of the algorithms.
c. Analyse the results and optimize the performance of your models for the application.

Your program must be structured into several parts (e.g. visualizing, preprocessing, training and validation, tuning and testing) and well commented.

Python code must be submitted in the form of Notebook file (.ipynb). Must provide clear instruction on how to run your Python code and get the desired results as indicated in the report. Make sure there is no runtime error in your code.

3.4 Report

Your report should include the followings:

- A **title page** stating the title of the assignment, student names and student IDs, and a list of the effort and contribution of each team member in terms of percentage as follow. **Marks will be allocated based on a team member's contribution**.

Student 1	Student 2	Student 3	Total
30%	30%	40%	100%

- **Chapter structure**:
  1. Introduction
     - Motivation: describe about the problem you want to solve, why it is important, etc.
     - Objectives: the purposes of your project and contributions.
  2. Method and Experiments
     - Dataset: describe data collection, data exploration and visualization.
     - Overall framework: a flowchart or high-level overview of the classification pipeline of your implementation.
     - Implementation details: describe each of the processing steps (e.g. preprocessing, model training and validation, tuning and testing) in your framework and provide justification for your selection and implementation.
  3. Results and Discussion
     - Experimental results: summary of data preprocessing, training and validation results.

- Performance analysis: in-depth analysis of the classification performance of your model.
- Model comparisons: compare performance between different models or with other published results.

4. Conclusions

4. Marking Criteria

The grading of your assignment will be based on the following criteria:

Marking Criteria	
Criteria	**Weight**
Introduction	5%
Data exploration and problem understanding	5%
Data analysis and preprocessing	20%
Model training and validation	20%
Performance and error analysis	20%
Results presentation	10%
Discussion & conclusion	5%
Effort (e.g. In-depth analysis, self-learnt learning algorithm)	10%
Coding style (variable naming, comments, formatting, etc)	5%
Total:	100%

.