



Computer Science and Creative Technologies

Coursework or Assessment Specification

Module Details

Module Code	UFCFD3-30-1
Module Title	Introduction to Artificial Intelligence
Module Leader	Prof. Jim Smith
Module Tutors	Jim Smith, Chris Simons, Nathan Duran, Emmanuel Sinclair-smith
Year	2019-20
Component/Element Number	CW1
Total number of assessments for this module	3
Weighting	25%
Element Description	Coursework - solutions to a series of search problems

Dates

Date issued to students	23/09/2019
Date to be returned to students	
Submission Date	23/04/2020
Submission Place	Online via Blackboard
Submission Time	2pm
Submission Notes	Deliverable is a single file of C source code. Submission and marking is automated, accessed through a link in the Assignments=>Second Coursework folder on Blackboard

Feedback

Feedback provision will be	Provided automatically by the marking system at the time of submission.
-----------------------------------	---

Contents

Module Details.....	1
Dates.....	1
Feedback.....	1
Contents.....	2
Section 1: Overview of Assessment.....	3
Section 2: Task Specification.....	3
Section 3: Deliverables.....	3
Section 4: Marking Criteria.....	3
Section 5: Feedback mechanisms.....	4
Appendices and Additional Information.....	5
Marking Criteria.....	6



Section 1: Overview of Assessment

This assignment assesses the following module learning outcomes:

Design and implement intelligent solutions using at least two different architectures.

Design and implement a range of different search methods.

The assignment is worth **25%** of the overall mark for the module.

The assignment is described in more detail in section 2.

This is an individual assignment.

Working on this assignment will help you to:

- Understand how a search-based framework can be used to solve different problems;
- Gain experience of the practical issues around creating machine learning systems;
- Demonstrate that you can implement a simple Machine Learning algorithm.

If you have questions about this assignment, please post them to the discussion board *Search and Machine Learning Coursework Discussion Forum* on Blackboard.

Section 2: Task Specification

In the lectures you will have been introduced to various paradigms for representing models. In the tutorials you will have been introduced to an example C code framework for creating search-based solutions to problems. The task is to use that framework to create a Machine Learning algorithm that can learn predictive Rule-Based models, using a greedy hill-climbing approach.

The system will present your code using a number of datasets. Each has 150 examples, split 100:50 between a 'training set' and 'test' set. Each example contains integer values for 6 features, and a single integer giving the class label between 0 and 5. **Your code should:**

- implement a *train()* function that 'learn' a set of rules from the training set. Some of the marks will be for implementing the right learning algorithm. For simplicity, your code should not generate solutions with more than 5 rules.
- implement a *predictLabel()* function that provide a predictions for a new case. Some of the marks will be for the effectiveness of your solution, i.e., its accuracy.

The code framework, supporting videos, and html documentation of the provided code-base will be provided in the "Assignments->Second Coursework" folder on Blackboard.

Section 3: Deliverables

The deliverable consists of a single file of C source code called *TrainAndTest.c* that implements at least two functions, one called *train()* and another called *predictLabel()*.

- The exact syntax to be implemented will be supplied in the online documentation.
- The submitted files must make use of the supporting code-base, and produce results in the required format, otherwise they will lose marks.
- All submissions will be screened for security reasons – for example making “system()” calls or excessive memory use in C code will not be permitted.
- All code should be prepared using an platform-independent standard e.g. ANSI standard C rather than proprietary.

Section 4: Marking Criteria

This coursework is being marked by an automated system. You will be allowed six submission attempts and the highest mark will be taken.

- Your code will be compiled and tested automatically on four data sets with different characteristics. **At least one of these will be different for every student.** The total mark awarded for a submission will be the average of the marks on the three datasets
- The datasets have been designed to have different ‘target’ accuracies. The marks for the **effectiveness** of your Machine Learning system will depend on how close you get to that target.
- The version of the supporting code framework used in the marking system will be instrumented to provide outputs about how it is being used. This will be used to mark your **implementation** of the search algorithm. To emphasize the concept of search as a generic process, marks will be deducted for failure to use the framework.

The two marking criteria are each worth up to 50 marks (so 100 marks in total), as follows:

Criteria	How Tested	0	10 marks	25	40	50
Design and implement an effective rule-based architecture for learning predictive models from training sets	<i>Solution will compute the difference D between the target and measured accuracy (in %) on test set.</i>	<i>Solution predicts class label not present in training data</i> <i>OR</i> <i>Solution fails to compile,</i>	$10 < D$	$5 < D \leq 10$	$2.5 < D \leq 5$	$D \leq 2.5$
Implement the specified search based algorithm for Machine Learning within the chosen architecture.	<i>Marking system will automatically detect calls to functions provided within code framework.</i>	<i>OR</i> <i>Solution fails to complete within given time limit on system</i>	<i>System does not implement specified search algorithm or model paradigm. For example, implements different algorithm such as kNN.</i>	<i>System appears to generate rule-based classifier but makes no use of framework</i>	<i>Solution appears to work correctly but only makes partial use of framework.</i>	<i>Algorithm correctly implemented making full use of the supporting framework. Solution generates and tests successively more complex rulesets. Solution stops correctly.</i>

Section 5: Feedback mechanisms

The automated marking system will provide your score and instant feedback.

Clicking on the 'feedback' button will provide you more information about why you have got the score given, and how you can improve.

If the automatic marking system awards your marks for the effectiveness component, but zero for making appropriate use of the code framework, please contact your tutors via email for further feedback.