

MODULE EVALUATION REPORT

Module	CS3244 - MACHINE LEARNING
Academic Year/Sem	2017/2018 - Sem 1
Department	COMPUTER SCIENCE
Faculty	SCHOOL OF COMPUTING

Note: Class Size = Invited; Response Size = Responded; Response Rate = Response Ratio

Raters	Student
Responded	48
Invited	99
Response Ratio	48%

1. Overall opinion of the module

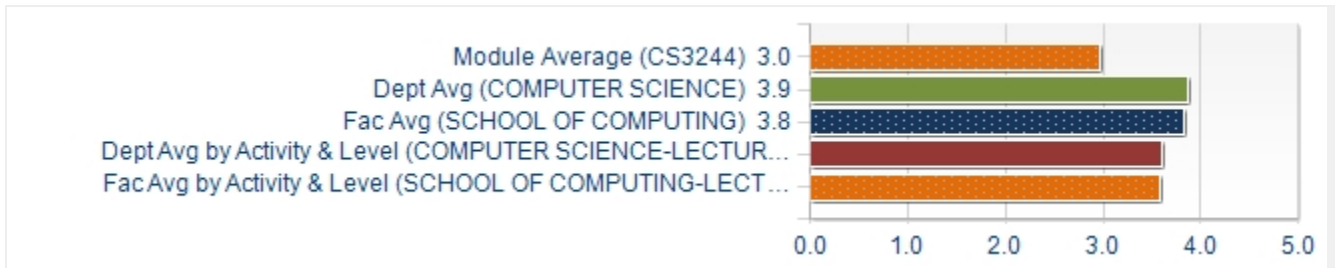
Frequency Analysis



Statistics	Value
Response Count	48
Mean	3.0
Standard Deviation	1.4

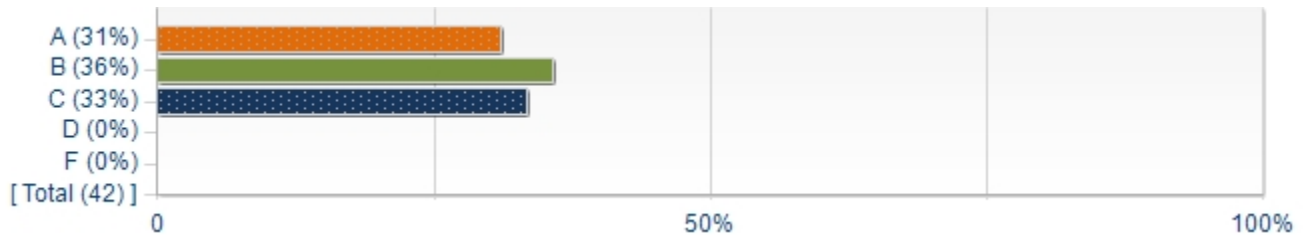
Normative Analysis

Question	Module Average (CS3244)		Dept Avg (COMPUTER SCIENCE)		Fac Avg (SCHOOL OF COMPUTING)		Dept Avg by Activity & Level (COMPUTER SCIENCE-LECTURE (Level 3000))		Fac Avg by Activity & Level (SCHOOL OF COMPUTING-LECTURE (Level 3000))	
	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
What is your overall opinion of the module?	3.0	1.4	3.9	0.9	3.8	0.9	3.6	1.1	3.6	1.1



2. Expected Grade

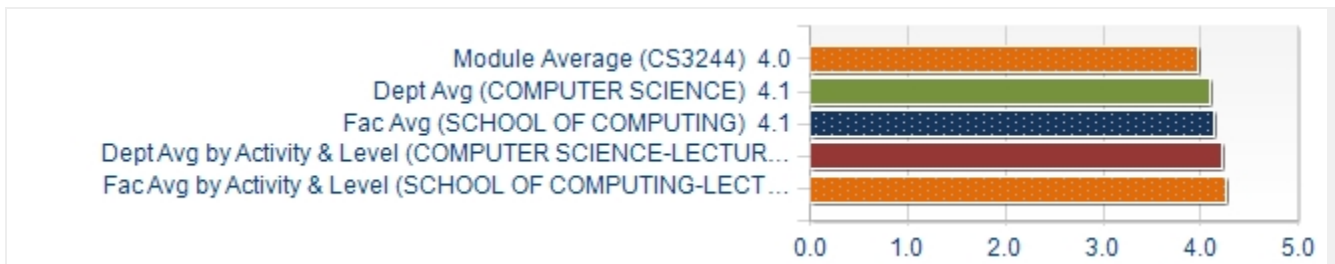
Frequency Analysis



Statistics	Value
Response Count	42
Mean	4.0
Standard Deviation	0.8

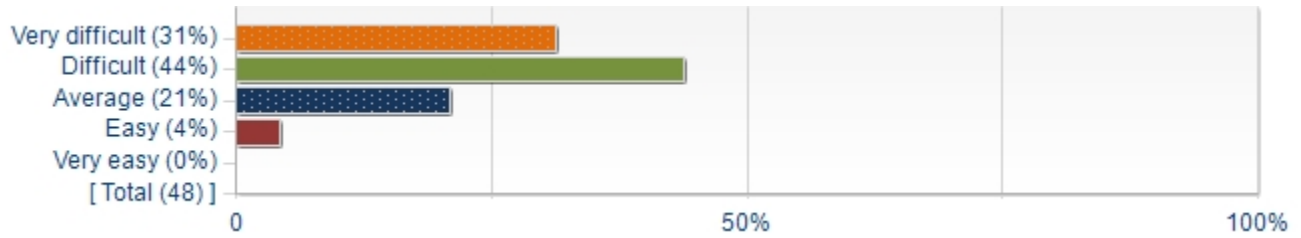
Normative Analysis

Question	Module Average (CS3244)		Dept Avg (COMPUTER SCIENCE)		Fac Avg (SCHOOL OF COMPUTING)		Dept Avg by Activity & Level (COMPUTER SCIENCE-LECTURE (Level 3000))		Fac Avg by Activity & Level (SCHOOL OF COMPUTING-LECTURE (Level 3000))	
	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
The grade that I am most likely to get in the module is:	4.0	0.8	4.1	0.8	4.1	0.7	4.2	0.7	4.3	0.7



3. Difficulty Level of the module

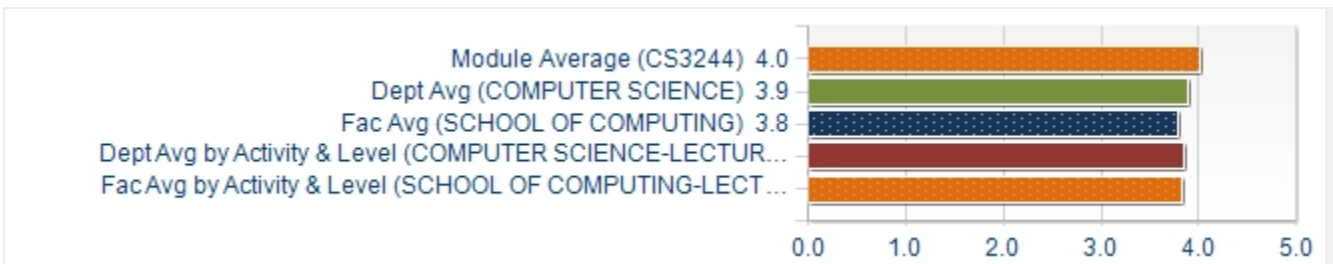
Frequency Analysis



Statistics	Value
Response Count	48
Mean	4.0
Standard Deviation	0.8

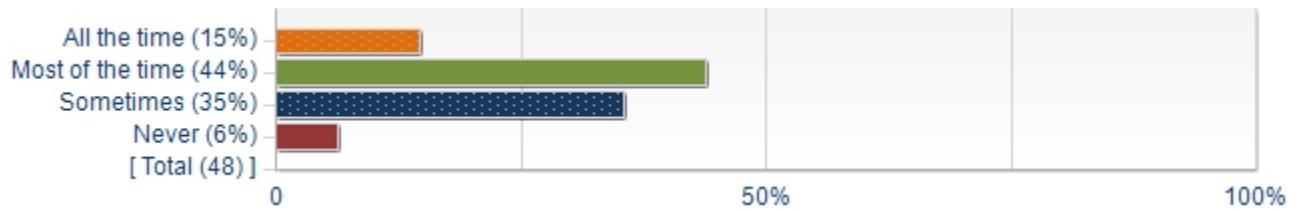
Normative Analysis

Question	Module Average (CS3244)		Dept Avg (COMPUTER SCIENCE)		Fac Avg (SCHOOL OF COMPUTING)		Dept Avg by Activity & Level (COMPUTER SCIENCE-LECTURE (Level 3000))		Fac Avg by Activity & Level (SCHOOL OF COMPUTING-LECTURE (Level 3000))	
	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
I rate this module as:	4.0	0.8	3.9	0.8	3.8	0.8	3.9	0.8	3.8	0.8



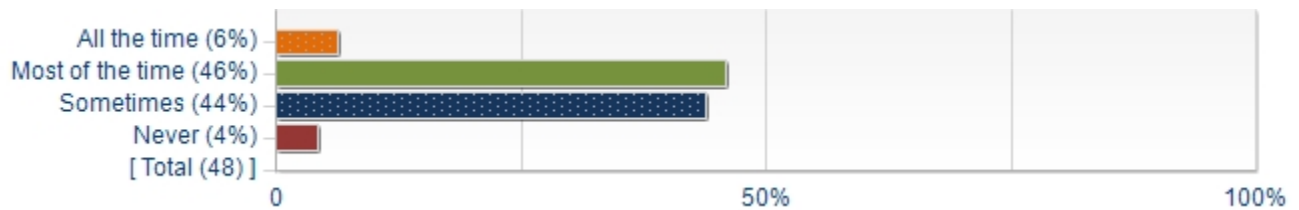
MODULE LEARNING OUTCOMES

1. Understand the basic concepts of machine learning.



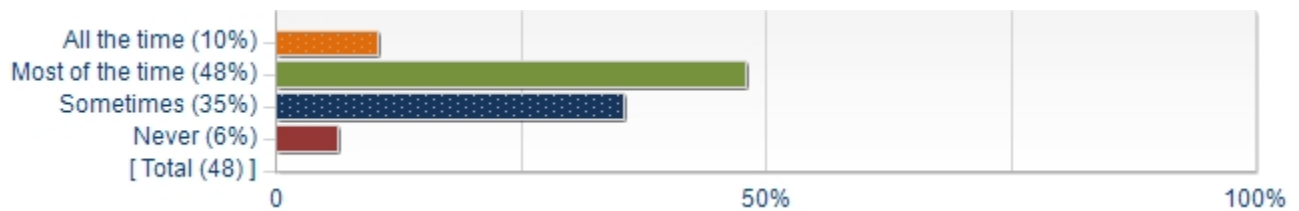
Statistics	Value
Response Count	48
Mean	2.7
Standard Deviation	0.8

2. Apply an appropriate machine learning algorithm for a given problem.



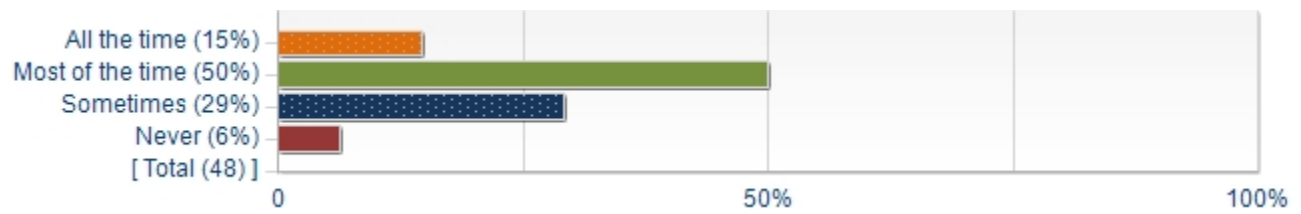
Statistics	Value
Response Count	48
Mean	2.5
Standard Deviation	0.7

3. Evaluate the performance of a machine learning solution.



Statistics	Value
Response Count	48
Mean	2.6
Standard Deviation	0.8

4. Use a machine learning tool to carry out machine learning experiments.



Statistics	Value
Response Count	48
Mean	2.7
Standard Deviation	0.8

WHAT I LIKE / DISLIKE ABOUT THE MODULE

What I liked about the module:

Comments
Homeworks which allow us to see the various machine learning algorithms being applied in a practical way.
I would like to add that for the last few tutorials of the course, Animesh took over and god I wondered why didn't he do that much earlier on. Animesh's teaching effectiveness is similar to Jay, except that Animesh clearly knew his stuff well, and is enthusiastic and passionate about transferring the knowledge to the students, and executes it well. Kudos to him, I didn't feel like I have completely wasted my time and money in this module.
I like that ML is such a popular topics, we can always find external materials to refer to. That means no matter how bad the lectures or the tutorials were it's not the end of the world.
–
–
–
It introduced me to a comprehensive set of concepts that are fundamental to machine learning. The syllabus is definitely great and follows a nice sequence.
The topic is popular and useful
Topics covered were very useful
Learned a lot about machine learning models
interesting
A brief overview of machine learning which is always interesting.
The homeworks.
Combination of Theory and application of Machine Learning.
Machine learning is a very practical subject.
N/A
interesting content, applicable to industry and future
The content
Don't know.
Interesting assignments & the presence of webcasts help, especially when the module is algorithm heavy. Multiple reviews help students really understand the material.
Covers a lot of different learning algorithms. Number and schedule of assignments is good.
The module covers basics topics in Machine Learning and gives me an overview of this subject.
This module exposes me to a lot of machine learning algorithms, and give me a broad overview of machine learning.
It exposed me to the power of machine learning and Python and Tensorflow.

What I did not like about the module:

Comments
It kind of follows other online courses, but not very closely, which result in double the time spent learning about the same thing.
Also, learning would be much more easier if assumptions, introduction, some background and intuition are introduced first before diving into the meat. This experience repeats itself chapter after chapter, in which I actually understood some of the things in the lecture slides.
Some inconsistencies and errors in lecture notes and tutorials went unnoticed by the teaching staff.
As much as Min's effort in revamping and deliering the course is appreciated, I felt that Min do not fully grasp some of

Comments

the subject material. Often, I had to scour through the internet to find video lectures explaining the same concepts but made so much sense, which led me to wonder why there was such a big difference in the delivery of knowledge.

The TAs do not seem to care about the students and seem to be just going through the motion to fulfill their teaching hour quota. I suggest they are not allowed to be TA other modules, for the sanity and benefit of future students.

—

everything, regret

—

The course's administration is rather troubling. The entire Kaggle thing was an embarrassment. It seemed no one bother to test whether the competition bounds made sense? i.e. being able to merge teams, having clearer instructions (are 0 sales ignored or mapped to 1?) and submission limits. Tutorials aren't even numbered correctly. They also have quite a few typos to add to the confusion.

I also heard through the grapevine that the makeup midterm was the previous year's midterm recycled in verbatim. If this is true, it represents a serious breach in the sanctity of examination and places other students at a significant disadvantage. It is trivial to get one's hands on the previous midterms via past students. It is not even the fault of the students who took the makeup test - nobody with a reasonably sound mind expects the exact same paper to be recycled.

Tutors need to be more well-versed, exam questions should be more mathematical and less vague.

The content of lecture, tutorial and assignment seemed to me segregated, as I always need to search for additional information to understand any of the three components.

The arrangement is bad

Flipped classroom did not work -- material was uploaded too close to the class time. More practical application-based assignments might be good too (e.g. HW3)

It has a steep learning curve, my gradient ascent takes too long

very theoretical and needs very good explanation and good time management

This course is the worst I have taken in School of Computing, and I truly care about SoC being a top class institution and hence this review will be lengthy but sincere.

Course Curriculum and Lecture

This course is very much a combination of Stanford's CS229, CS231 and Caltech's Machine Learning course. On one hand it might be very tempting to believe that our students actually learn both in 1 module. On the other hand, it is really naive to think so. Most of our lectures are touch and go with extremely little explanation. It is like applying "wishful thinking" that our students can even have a strong mathematical underpinning on the materials. With this materials, we at best have a heuristic understanding. If we want a touch and go style lecture, we should give readings such as <http://cs231n.github.io/convolutional-networks/> BEFORE the lecture and make them compulsory. We can even cut off the entire part on neural networks, CNN and deep learning and focus on the portion in front. Teaching poorly and glossing over materials is as good as not teaching, we might as well focus on the portion in front.

Too much content. Maybe deep learning can be omitted to gain a better understanding of the other half.

It is nearing the end of the semester and I am still not able to pinpoint exactly what I learnt or took out of this module. This could speak to either the failure of the module or of myself, but in this case I believe it would be the former.

1. Lecturer takes on an interesting approach to math

One can either attempt to take on a rigorous stance to the mathematics behind machine learning, or focus on the intuitions without the math. The lecturer seems to not be able to decide between the two. It is not acceptable to dump mathematical monstrosities on the lecture slides with little to no explanation. It takes a while for me to even figure out the *notation*, let alone the proofs (often non-existent)!

I would like to note here that while many other lecturers do re-appropriate much of their lecture resources from elsewhere, it takes a special level of incompetency to re-appropriate an entire module from Yaser and yet still do so much drastically worse. I have taken to watching his lectures, instead of the official ones from this module.

2. Disconnect between lectures and tutorials

Comments

No one expects the lecture to spoonfeed all the answers such that the tutorials are an exercise in memory. It is good for tutorial questions to require some higher order thinking about the material covered in lecture. What is completely unacceptable is that prerequisite facts such as mathematical definitions required for solving the tutorial questions are not even provided anywhere. Going through the tutorial answers is not the time to pull prerequisite facts out of nowhere. It is due to this that many of my friends completely stopped bothering to attempt the tutorial questions after the third week, as it is an exercise in futility.

3. In fact, tutorials are a joke

A tutor can work through a question halfway and completely forget how the proof works, even after flashing the solutions on the screen. Even as a fellow undergraduate tutor myself, I cannot sympathize with this lack of preparation.

4. Course admin is a mess

I can sympathize with the tutors having 50% more students than originally planned for. I cannot sympathize with not even having the discipline to upload the tutorial pdfs with a proper naming convention -- there are like 3 different versions of the pdf uploaded with name "tutorial-9.pdf", "totorial-9.pdf" [sic], "T9.pdf", etc. (And yet our homework submissions are so heavily penalized for small things like extra nested zip structure?)

In addition, the Kaggle competition is a disaster. Any other module would have its tutors at least beta-test the platform to ensure that it *works**. Yet, there are so many errors that a very basic, 2-min check, would have surfaced -- For the first week of the competition, no one can even submit any predictions! Thereafter, the judging criteria was different from that as specified on the homework specifications (specs says drop rows with 0 sales, judging criteria coerces them to 1). Yet, the TAs did not so much as create an announcement to highlight this, instead hiding these facts in some obscure forum post. Furthermore, team merging wasn't supported.

I find it hard to believe that the tutors being overworked with additional students can serve as an excuse for not having the time merely to copy and paste a forum post into an IVLE announcement. I would suggest that some responsibility be taken here, as opposed to blaming everything on the extra 50% student intake.

5. Those who live in glass houses shouldn't throw stones

The teaching staff penalizes entirely arbitrarily in terms of coding style. There is an inherent subjectivity in coding style, yes, but if the teaching staff is unwilling to document their coding style, then penalties in this domain should be kept to the most obvious of cases, i.e. completely unreadable code. Yet, things like missing comments for functions that are so short as to be obvious at a glance incur a penalty.

In addition, I argue that the solutions to the coding questions in tutorial, as well as the template for the homework assignments, indeed fit this bill of unreadability. PEP8 seems to have been completely thrown out of the window. There are missing spaces between operators. The naming of variables seems to be inspired by the alphabet song. In fact, visually looking at the code, I am amazed that some of the code even *works**.

To put things in perspective, missing comments for small, easy-to-understand functions are at most a small transgression (I argue that it is not a transgression at all), in light of the code monstrosities that your tutors are uploading as tutorial solutions.

6. Reusing a midterm paper from last year as re-midterm? Seriously?

The lecturer has repeatedly refused to release any midterm paper from previous semesters, citing excuses such as the curriculum being vastly different, and practicing on the previous semester's midterm not likely to be useful in preparing for the midterm. Yet, I have it on good authority that the lecturer re-purposed in full (i.e. not a single change) the previous semester's midterm as a make-up midterm paper this semester. Clearly, the lecturer himself cannot possibly believe his own excuses, if he can set last semester's midterm as a make-up midterm paper this semester.

In addition, last semester's midterm is publicly available, from the previous batch of students. Indeed, the PDF was being circulated around freely.

I note, for the record, that the top scorer for the midterm took the make-up midterm paper and got full marks for the paper. I would like to point out that if I took the make-up midterm paper (which is last semester's paper), I also would

Comments

have gotten full marks, as would countless others, everyone having done it as a practice paper before.

In closing, this is possibly the worst module I have ever taken in my entire NUS life. Part of the reason for this could indeed be circumstantial (i.e. the student intake), but I would suggest that a large bit of it arises from the lack of accountability of the teaching staff.

Totally kills my interests in machine learning. Lecture materials are confusing with so many mistakes. Learning from other online lectures are even more effective than participating in lectures
So-called flipped class room video normally released one day before the lecture. Totally meaningless.
Until now homework 3 grade is not released. Very ineffective team

very math intensive, but thats the nature of the subject. oh well.

The prof could not teach well but expect us to learn well

Thrown to watch tons of videos outside lectures because the lectures is unclear (skipping a lot of contents). Don't really have time for this.

Assignment requirements are not always spelt out clearly, and can sometimes be interpreted wrongly if a single phrase is missed. A table specifying deliverables would be really helpful. Losing marks due to a silly reason like the above in assignments is quite a painful thing.

The second half of the module had content that was very difficult, feels like content overload. There isn't much time to get in-depth explanations of everything.

Many concepts are too instensive in Math, which I decided that I only tried to get a ground understanding and ignore the Math proof.

It was too difficult to handle. Tutorials were very difficult to follow as well although they were meant to clarify concepts taught in lecture.

It didn't feel like there was much thought put into helping students understand the content, although some effort was there.

I think the main problem was that it was assumed that everyone taking this module had a good grasp of linear algebra / statistics / even machine learning.

everything. I hate everything about this module. From the god awful slides to the god awful TAs who clearly have no understanding of the material