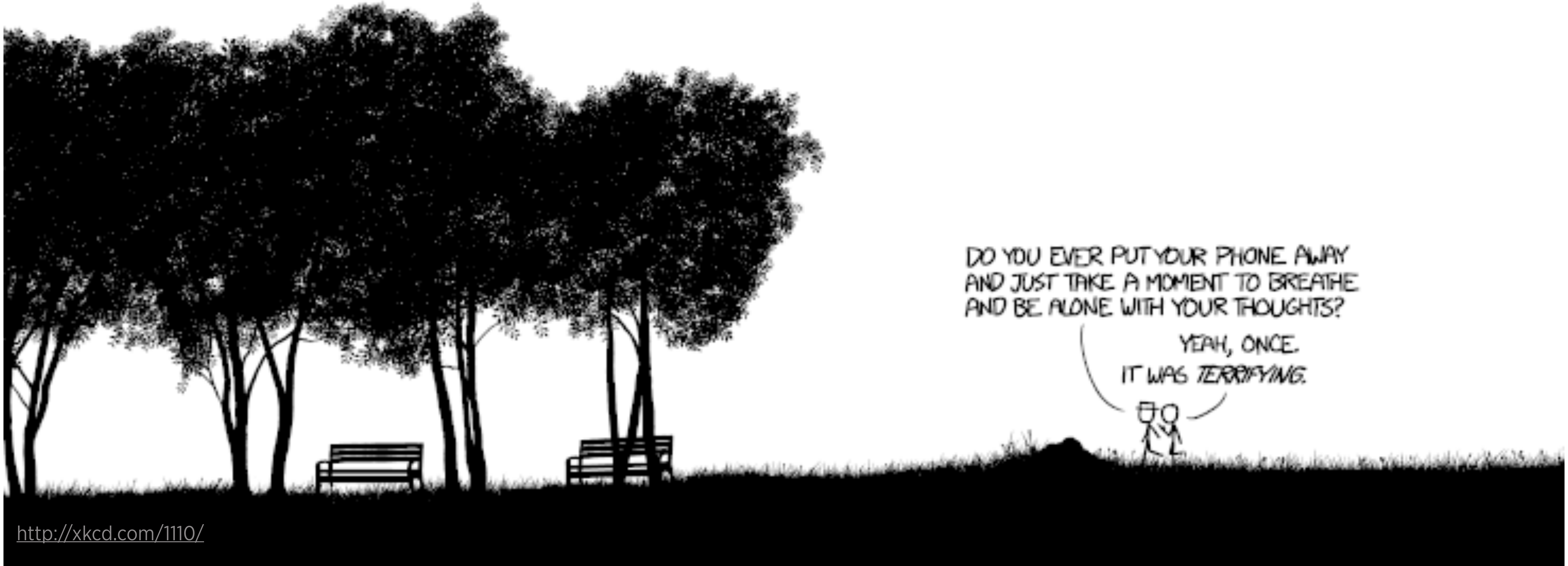
DO YOU EVER PUT YOUR PHONE AWAY AND JUST TAKE A MOMENT TO BREATHE AND BE ALONE WITH YOUR THOUGHTS?
YEAH, ONCE. IT WAS *TERRIFYING*.

http://xkcd.com/1110/

Predictive Analytics

Rosalind Wang

Spring 2024

What does “*predictive*” mean?

This class is an introduction to machine learning methods, with a focus on (relatively) simple methods that are useful to make predictions of sorts.

“Predictive” could mean about the future – but does not have to mean that. We say “predict” when we estimate outcomes for previously unobserved parts of a data set.

Some machine learning methods are useful for different (or additional) purposes, for example to provide insights about data, or to decide best possible actions.

What can you do with these methods?

Examples:

Predict (estimate) prices for items (e.g. houses, cars, …) based on properties of that item

Distinguish different kinds of objects in images

Estimate the age of a person in an image

Distinguish spam emails from useful emails

“Read” hand-written text

…

Understanding vs Applying methods

It is possible to apply machine learning methods without knowing how they work. There are many libraries and programming frameworks that make using these relatively simple.

There exist many different methods in machine learning, there is not one “best” method for all problems. If you understand how some of the methods work, this will help you to

- select a method that has a better chance of achieving good results
- find out what went wrong, and what to do if things don’t work as expected
- modify existing methods, implement new methods from scratch if you have to.

This class focuses on understanding (i.e., the theory) and (practically) applying the methods. There is less focus on implementing from scratch.

Class structure

Lectures:

- Monday 10:00 - 12:00, PS-EA.G.19

Tutorials:

- Wednesday 11:00-13:00, PS-EB.1.46
- Wednesday 15:00-17:00, PS-EB.1.46

There will be no recordings of the classes

To excel in this subject you should attempt the tutorials ***before*** *the Wednesday class.*

Subject *Philosophy*

“Understanding” assumes some knowledge that you may have from school or earlier classes, mostly linear algebra, calculus, stats.

“Applying” means that you will have to program (usually short pieces of code). We will use Python as programming language.

We will explain and revise some of that, but this is not a Maths or Programming class. If you don’t understand or don’t know how to do things at first, persist and try to figure things out.

We aim to not be overly prescriptive: there are usually multiple ways to solve problems, often there are various tools that you could use. If you find a better way to solve a problem than what we showed you in class – this is great. We prefer you to think for yourself and not just copy/paste. :)

Ask questions

Academic misconduct

https://www.westernsydney.edu.au/studysmart/home/study_with_integrity/academic_integrity_module

What we will be doing

Week	Lecture	Assessment due	Non-assessment
Week 1 22-07-2024	Introduction to the unit. Using Python		
Week 2 29-07-2024	Maths revision		
Week 3 05-08-2024	Gradient Descent	In-class quiz 1	
Week 4 12-08-2024	SGD and variants, linear and log regression		online quiz 1
Week 5 19-08-2024	Regularisation and motivation, Ridge Regression.		
Week 6 26-08-2024	Feature selection and regularisation. Lasso regression.	Assignment part 1	online quiz 2
Week 7 02-09-2024	Neural Networks (I)	In-class quiz 2	
Week 8 09-09-2024			
Week 9 16-09-2024	Neural Networks (II)		online quiz 3
Week 10 23-09-2024	Support Vector Machines		
Week 11 30-09-2024	Naive bayes		online quiz 4
Week 12 7-10-2023	*Labour Day*		
Week 13 17-10-2023	Semi-supervised learning	In-class quiz 3, Assignment part 2.	online quiz 5
Week 14 24-10-2023	Practical machine learning / applications	Assignment presentation	

Assessment Items

In-class quizzes (3 quizzes)	20%
Assignment part 1	10%
Assignment part 2	50%
Oral exam	20%

To pass, you will need to

- Achieve at least 50% overall
- Achieve at least 50% of the practical exam
- Attempt all assessment items

This unit is 10 credit points. In general, a student is expected to study an hour per credit point a week. That means, this unit will require 10 hours of study per week. This time includes the time spent within classes during lectures, tutorials or practicals (it also means that you shouldn’t take more than 4 units at the same time).

Teaching Team

Lectures:

- Rosalind Wang (rosalind.wang@westernsydney.edu.au)

Tutorials:

- Vishal Patel

Communications

vUWS announcements — ***important*** information

Discord — questions about the subject, discussion, small announcement from me

Email — questions that are more personal, e.g. you will miss a quiz

University’s guideline for email communications.

The tutorials

Will be in CDMS computer labs, where we could directly help with practical questions (also where everyone could help each other).

We will give you some tasks for the tutorial.

The idea is you then try solve the tasks on your own or in a group

In the tutorial, you can ask questions that you could not solve.

Discussing alternative solutions.

The usual questions

Will lecture X / topic Y / something we do in a practical be part of the exam?
Yes, possibly.

Will the exam be open book?
Yes, you can use your notes and code, any book.

Will we have to program, can we use our own computer?
Yes, and most MacOs/Windows/Linux computers should work.

Can we have a practice exam?
The exam will be similar to the quizzes and tutorial questions, there will be no additional practice exam.

What we will be doing

From 10000 miles high:

- We will look at basic Data Science (machine learning) methods
- We will write some code in Python to find out how to use these methods
- We will run experiments to see how and how well these methods work

- We recommend Python 3 / anaconda / Jupyter / Spyder. We will teach ***some*** things programming, but as this not a programming class, you will ***also*** need to teach yourself.
- We will use GitHub Classroom for group work

What should I do with my time?

(not a usual question, but glad that you asked)

- You will get much better at this with practice, so ..
- There are heaps of resources (tutorials, data sets, competitions) online
- Find practice projects for you to solve
- Good resources: If you don't know what to practice at any time - go there.
  - kaggle.com (competitions, forums, intro material)
  - edx.org self-paced online classes (maths, programming, ...)
  - https://towardsdatascience.com Data Science blog

  Also read this short article: http://norvig.com/21-days.html by Peter Norvig (Google): Teach Yourself Programming in Ten Years
- I will suggest online material from time to time

What books should I read / buy?

RTFLG (Read the fantastic learning guide :) for a list, but … you don’t have to buy a book.

“The Elements of Statistical Learning: Data Mining, Inference, and Prediction” (written by Trevor Hastie and his colleagues) is a great book and listed in the learning guide. This book is also useful as a reference for later. You can also find the PDF here: https://web.stanford.edu/~hastie/Papers/ESLII.pdf

You may also want or need to refresh and improve your maths, the book "Mathematics for Machine Learning" is very new and written for that purpose. There is much more in the book than we need for this class, but there is also a PDF online (https://mml-book.github.io/book/mml-book.pdf).

We have now transitioned the class to Python (we have been using R until last year). Learning programming languages is an ongoing project for every serious data scientist: If you don't know how to to get started: you could try the "Python Data Science Handbook" (https://jakevdp.github.io/PythonDataScienceHandbook/).

Python and programming environment

https://docs.python-guide.org/starting/installation/

Programming

This class will require you to write (short) programs in Python.

Any recent python (version 3) should work.

We recommend Jupyter Notebook if you have no other preference.

You don’t have to use Jupyter, you could also use any editor you like, and/or the command line. Extra geek cred if you end up using Emacs, and we will be impressed.

Anaconda

https://www.anaconda.com/products/individual

jupyter
SciPy
SPYDER
NumPy
DASK
bokeh
scikit learn
TensorFlow
Numba
pandas
$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$
Panel
matplotlib
PYTORCH
HoloViews
CONDA