Boring but important disclaimers:

- If you are not getting this from the GitHub repository or the associated Canvas page (e.g. CourseHero, Chegg etc.), you are probably getting the substandard version of these slides Don't pay money for those, because you can get the most updated version for free at

  `https://github.com/julianmak/OCES4303_ML_ocean`

  The repository principally contains the compiled products rather than the source for size reasons.
- Associated Python code (as Jupyter notebooks mostly) will be held on the same repository. The source data however might be big, so I am going to be naughty and possibly just refer you to where you might get the data if that is the case (e.g. JRA-55 data). I know I should make properly reproducible binders etc., but I didn't...
- I do not claim the compiled products and/or code are completely mistake free (e.g. I know I don't write Pythonic code). Use the material however you like, but use it at your own risk.
- As said on the repository, I have tried to honestly use content that is self made, open source or explicitly open for fair use, and citations should be there. If however you are the copyright holder and you want the material taken down, please flag up the issue accordingly and I will happily try and swap out the relevant material.

OCES 4303 :

an introduction to **data-driven and ML methods** in ocean sciences

Session 1: basics of ML and Python refresher

Outline

- admin things (canvas, GitHub, Colab)
- approach of the course
  $\rightarrow$ lecture, workshop, assessment
- demonstration

Content of course is somewhat ocean science motivated, but skills are entirely generic

Practicalities

Instructors:

- I Julian Mak (`jclmak@ust.hk`)
- TA Jonathan Lee (`hcleear@connect.ust.hk`)

Course grade breakdown:

method	
quizzes	$4\% \times 5 = 20\%$
assignments	$20\% \times 3 = 60\%$
interview style exam	20%

- pass mark is going to be 60%
- "A" boundary will be around 85-90%

(I don't grade to a curve)

- assignments with plagiarism will get a minimum of **zero**

Practicalities

- weekly **Fri 1330 to 1630**

  $\rightarrow$ first 40-60 mins is lecture / outline

  $\rightarrow$ rest of the time is computer workshop, which is the **main learning part**

- F2F in class

  $\rightarrow$ it's more for you: us trying to help you code/debug without control/access to your computer is usually frustrating for both parties

- **Prerequisite: OCES 3301**

  $\rightarrow$ will assume you are familiar with the majority of content there

Approach of course

What this course is:

- a hands-on intro to basics of data science and machine learning methods in Python

  → on understanding principles, sampling some methodologies, learn by tackling some problems (some contrived, some 'realistic')

  → transferable skills

- **you have to try do stuff**

  → the lectures are entirely supplementary to the computer workshops

- **focus on teaching you to teach yourself**

Approach of course

This course is not:

- a computer science and/or Python course as such

  $\rightarrow$ we use Python but the focus is the analysis tools
- a statistics/maths course

  $\rightarrow$ although you will be dealing with numbers quite a bit

(!!! will not deal with text, language processing and LLMs in this course !!!)

Approach of course

This course is <u>not</u>:

- a computer science and/or Python course as such

  → we use Python but the focus is the analysis tools
- a statistics/maths course

  → although you will be dealing with numbers quite a bit

(!!! will not deal with text, language processing and LLMs in this course !!!)

After this course you will <u>not</u> be:

- a computer/machine learning wizard

  → it will hopefully give you some mindset and tools to work towards being one (if you want to)
- know how to solve every data problem in ocean science

  → but it should teach you how you might get started on other problems

Material

- most of the material is in the Jupyter notebooks

  $\rightarrow$ most things in slides are actually taken from the notebooks

  $\rightarrow$ if you don't want to listen to me you could always go straight to the notebooks

Material

- most of the material is in the Jupyter notebooks

  → most things in slides are actually taken from the notebooks

  → if you don't want to listen to me you could always go straight to the notebooks

- for extra reading, try Google (search the topic)
- for Python syntax queries, try Google (stack overflow / stack exchange)
- for help with debugging, try Google (stack overflow / stack exchange)
- for help with science queries, try Google

- you could try me but I would probably tell you to try Google first (teaching you to teach yourself)

Syllabus

S01 Basics of ML, Python refresher, data handling

S02 Recap of regression and probability, data scaling, (cross-)validation

S03 Linear models + dimension reduction

S04 Clustering

S05 Classification tasks

S06 Decision Trees

S07 Random Forests + Gradient Boosting

S08 Neural Networks

S09 Convolutional Neural Networks

S10 Other architectures (e.g. autoencoders, RNNs, transformers)

- lecture material is **cumulative**
- quizzes every week, assignments at S04, S07 and S10

  → model assignment and marking criteria will be provided when assignment 1 is released, and applicable to all subsequent assignments
- material for up to three bonus lectures to be confirmed (e.g. PINNs, SINDy, TDA, ethics, GANs, diffusion models etc.)

Some propaganda to start with

ML algorithms are:

- algorithms + tools, and that's it

  $\rightarrow$ very powerful, but context dependent
- usually black box

  $\rightarrow$ it can work wonderfully / fail spectacularly, but you don't necessarily know why...

!!! prudent to do checks!

(see Lec 02, cross-validation)

Figure: Hermeowus Mora, disciple of Hermaeus Mora the Daedric prince of knowledge and memory

Some propaganda to start with

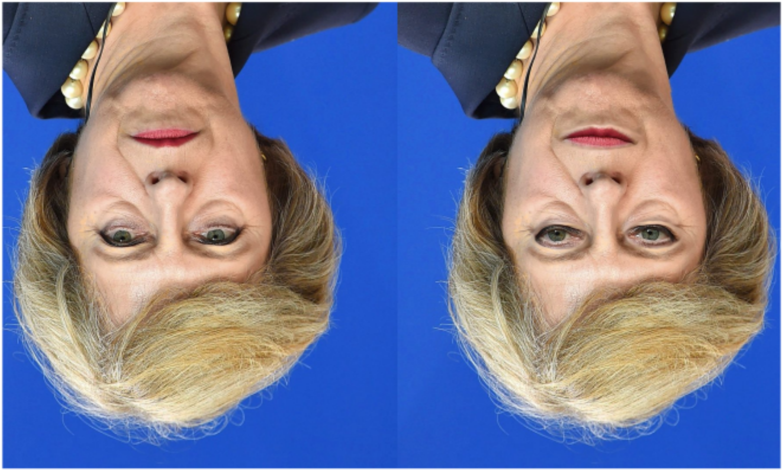


Figure: Cursed beast?

Some propaganda to start with

Figure: Cursed

Machine learning + regression

Recall regression is that, for X the input, y the output, f the model, we want

$$y = f(X)$$

- prediction or forward problem: have X and f, want y
- inference or inverse problem: have y and f, want X
- inference/regression/Machine Learning: have X and y, want f

Unsupervised vs. supervised

- unsupervised ML is where data is unlabelled, and algorithm picks out features by themselves

Figure: Eigenpets essentially. Figure adapted from Fig. 10 of Brunton, Brunton, Proctor & Kutz (2013); will do an easier case of this in Lec 03.

Unsupervised vs. supervised

- supervised ML is where data is labelled, and algorithm fits model between input and output
  $\rightarrow$ often want this for prediction purposes

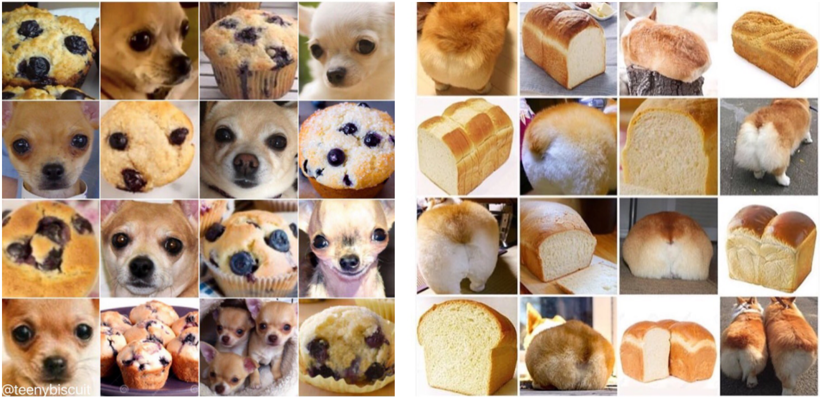


Figure: Various entries from the “animal or things” meme, as found on the internet.

Unsupervised vs. supervised

Figure: Various entries from the “animal or things” meme, as found on the internet.

Data handling: functions

- suppose I take

$$f(t) = \sin(t)$$

$\rightarrow$ then for $t = \{0, 1, 2 \ldots\}$, I store them as an array as

$$\begin{aligned} \texttt{f} &= [\sin(0), \sin(1), \sin(2), \ldots] \\ &= [0.000, 0.841, 0.909, \ldots] \end{aligned}$$

$\rightarrow$ plotted out as the blue curve

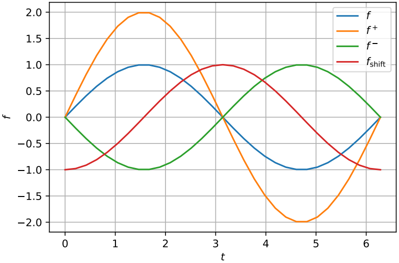
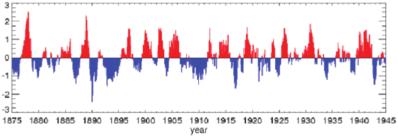


Figure: Some curves.

Data handling: numerical data

1		1948		2019									
2	1948	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99
3	1949	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99	-99.99
4	1950	24.55	25.06	25.87	26.28	26.18	26.46	26.29	25.88	25.74	25.69	25.47	25.29
5	1951	25.24	25.71	26.90	27.58	27.92	27.73	27.60	27.02	27.23	27.20	27.25	26.91
6	1952	26.67	26.74	27.17	27.80	27.79	27.18	26.53	26.30	26.36	26.26	25.92	26.21
7	1953	26.74	27.00	27.57	28.04	28.28	28.12	27.43	26.94	27.01	26.87	26.88	27.00
8	1954	26.98	27.03	26.90	26.64	27.12	26.80	26.11	25.43	25.12	25.23	25.57	25.26
9	1955	25.61	25.81	26.22	26.60	26.66	26.55	26.15	25.51	25.28	24.41	24.25	24.57
10	1956	25.34	25.76	26.46	26.85	27.13	26.81	26.23	25.68	25.73	25.75	25.56	25.71
11	1957	26.04	26.54	27.46	28.23	28.55	28.36	28.17	27.69	27.44	27.42	27.62	27.90
12	1958	28.33	28.24	28.27	28.27	28.31	27.99	27.32	26.85	26.40	26.45	26.75	26.62
13	1959	27.07	27.18	27.47	27.88	27.70	27.37	26.44	26.09	25.92	26.24	26.04	26.18
14	1960	26.27	26.29	26.98	27.49	27.68	27.24	26.88	26.70	26.44	26.22	26.26	26.22
15	1961	26.23	26.56	26.94	27.36	27.75	27.67	26.89	26.19	25.78	25.71	26.07	25.97
16	1962	25.96	26.19	26.80	27.13	27.05	27.08	26.76	26.33	25.94	25.97	25.75	25.67
17	1963	25.77	26.22	27.18	27.78	27.63	27.62	27.78	27.48	27.40	27.36	27.47	27.62
18	1964	27.34	27.13	27.02	26.95	26.82	26.59	26.33	25.60	25.32	25.37	25.26	25.23
19	1965	25.66	26.19	26.94	27.38	27.99	28.09	27.90	27.97	28.01	28.17	28.12	27.96
20	1966	27.67	27.55	28.21	28.16	27.55	27.64	27.33	26.48	26.27	26.22	26.23	26.03
21	1967	25.88	26.11	26.50	26.74	27.35	27.47	26.97	26.44	25.86	25.97	26.08	25.95
22	1968	25.69	25.68	26.33	27.10	27.19	27.88	27.58	27.01	26.72	26.75	27.20	27.27
23	1969	27.50	27.86	27.82	28.13	28.29	27.69	27.08	27.02	27.15	27.34	27.10	26.98



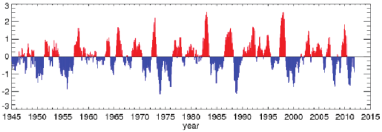


Figure: El-Niño 3.4 data (with masking values and averaged accordingly), and plotting the anomalies (relative to the de-trended signal).

Data handling: numerical data

- multi-dimensional arrays can be visualised as images
  → simulation data, satellite data, etc.

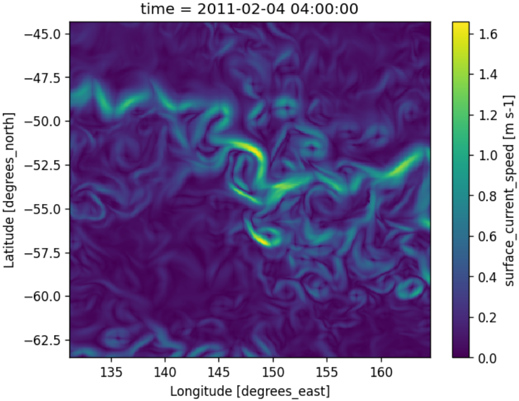


Figure: Surface current speed from simulation data (NEMO ORCA0083-N01), somewhere in the Southern Ocean.

Data handling: images

Figure: Pictures of the other three ad hoc TAs in the course, with special appearance from (in my opinion) the OG AI assistants. These guys will show up throughout the course.

- (64, 64, 1) arrays

  → entry denotes how 'light' it is (255 = white, 0 = black, gray in between)

  → converted from RGB, i.e. (64, 64, 3) arrays

Data handling: text + sound

Figure: The legend himself.

- text and sounds can be digitised and manipulated also (e.g., Shazam)

  $\rightarrow$ not going to focus on text (e.g. LLMs) or acoustic data

Demonstration

- GitHub repository
- Jupyter notebooks
- Python (+ Anaconda and/or Google Colab)
- data types
- xarray, pandas etc.
- data manipulation, plotting etc.

Moving to a Jupyter notebook $\rightarrow$