**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 7: random forests and gradient boosting

# Outline

- TL;DR: **ensemble of trees**

- random forests

  → boostrapping + bagging

  → averaging/voting

  → out-of-bag samples

- gradient boosting

  → **boosting** as an optimisation problem

  → **gradient**-based approach for optimisation

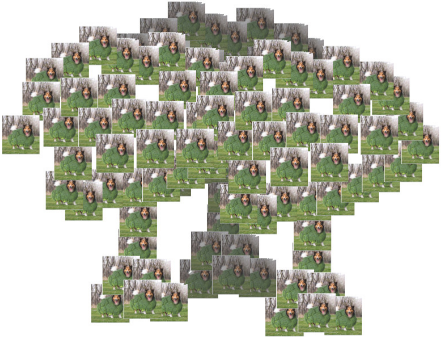


**Figure:** A forest made of trees.

# Oceanic application

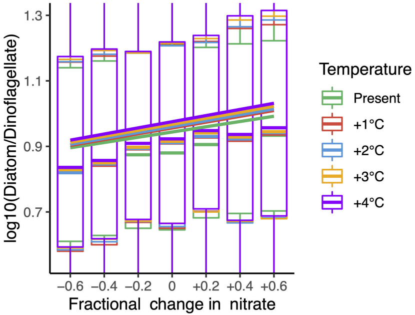


**Figure:** Predicted log of diatom-dinoflagellate ratio in HK coastal waters using random forests (which is a collection of decision trees) under projected warming scenarios. From Fig. 4 of Cheung *et al.* (2021).

- marine pollution

  → used station data to train up a regression model

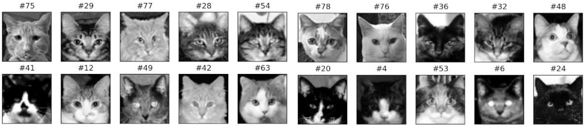
  → focused on diatom-dinoflagellate ratio

  → identified key features of importance

  → can do projection (!!!) and find pollution increases under warming

(see also Lee *et al.* (2025) from OCES)

# Schematic of random forests



full data

**Figure:** Schematic of random forests: full data.

# Schematic of random forests

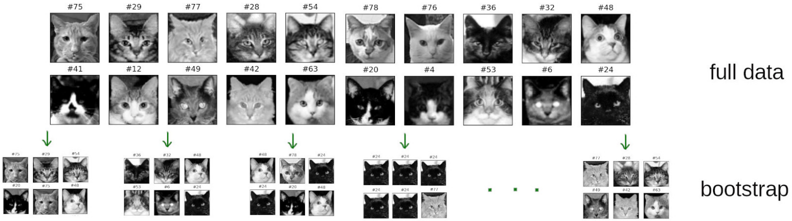
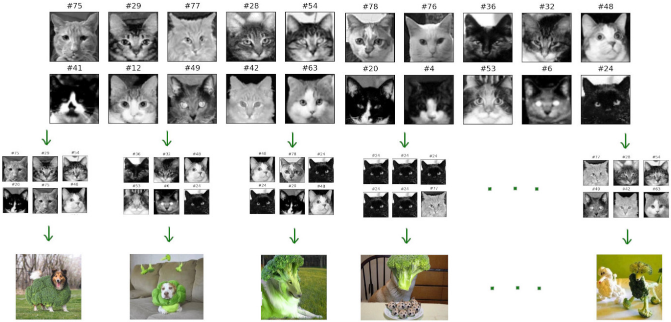


**Figure:** Schematic of random forests: bootstrap sampling, sub-sampling **with replacement** (!!!).

# Schematic of random forests



full data

bootstrap

bagging

**Figure:** Schematic of random forests: bagging (or bootstrap aggregation).

# Schematic of random forests

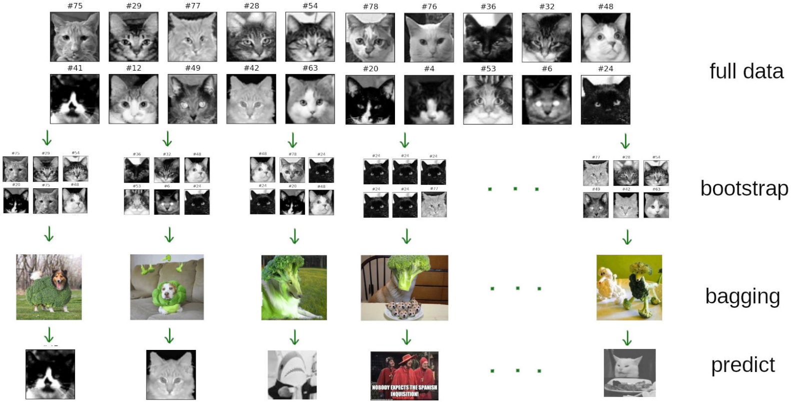
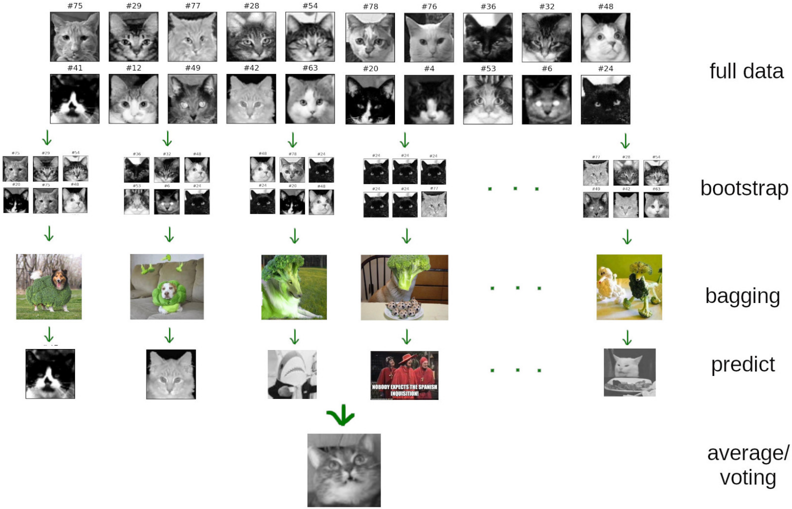


**Figure:** Schematic of random forests: bagging (or bootstrap aggregation).

# Schematic of random forests



full data

bootstrap

bagging

predict

average/ voting

**Figure:** Schematic of random forests: averaging/voting (depending on regressor or classifier).

# Example: penguins data

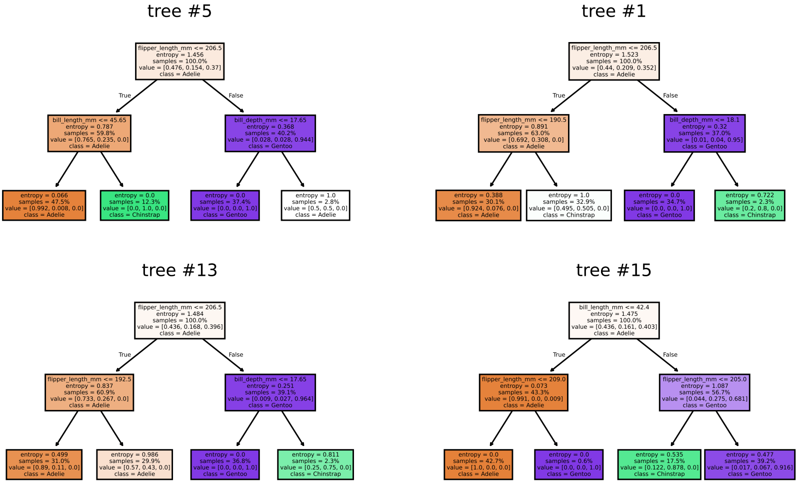


**Figure:** Sample of trees within the ensemble. Max depth two and information criterion.

# Example: penguins data

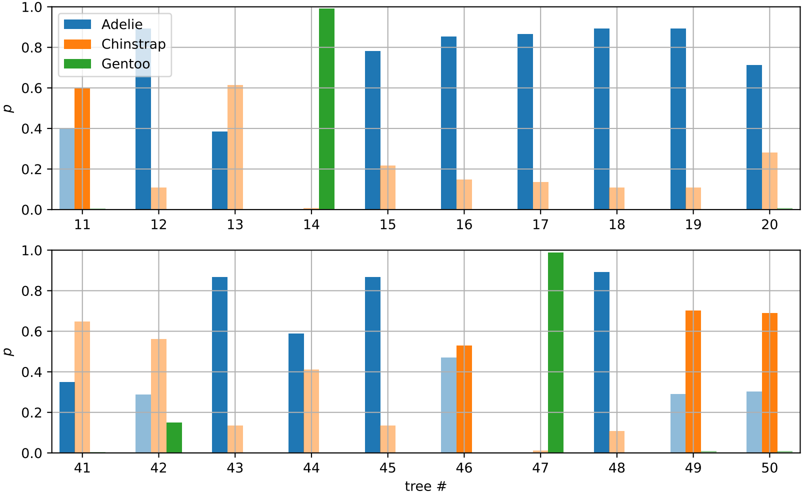
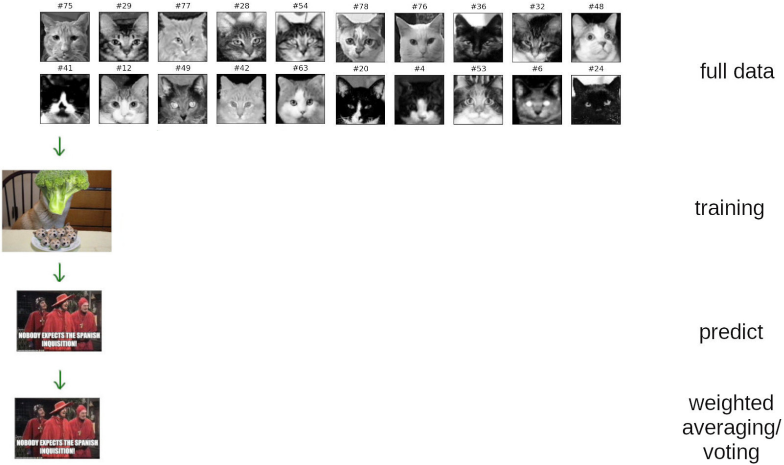


**Figure:** Probabilities associated with the predictions on test set with random forest. Highlighted bar denotes truth, and note highest bar is not always highlighted bar (e.g. index 13, 41, 44).

# Schematic of boosting


#75
#29
#77
#28
#54
#78
#76
#36
#32
#48
#41
#12
#49
#42
#63
#20
#4
#53
#6
#24
NOBODY EXPECTS THE SPANISH INQUISITION!
NOBODY EXPECTS THE SPANISH INQUISITION!

full data

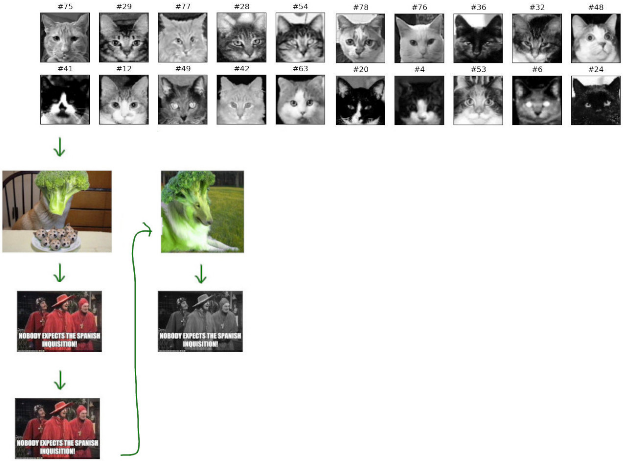
training

predict

weighted averaging/ voting

**Figure:** Schematic of boosting: train a weak model, and compute mismatches (e.g. wrong colour).

# Schematic of boosting


#75
#29
#77
#28
#54
#78
#76
#36
#32
#48
#41
#12
#49
#42
#63
#20
#4
#53
#6
#24
NOBODY EXPECTS THE SPANISH INQUISITION!
NOBODY EXPECTS THE SPANISH INQUISITION!
NOBODY EXPECTS THE SPANISH INQUISITION!

full data

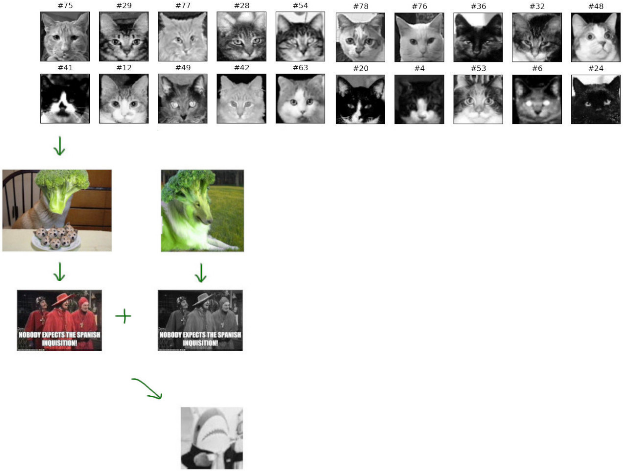
training

predict

weighted averaging/ voting

**Figure:** Schematic of boosting: train new model to reduce previous mismatches (e.g. predicting a grayscale image).

# Schematic of boosting


#75
#29
#77
#28
#54
#78
#76
#36
#32
#48
#41
#12
#49
#42
#63
#20
#4
#53
#6
#24
+
NOBODY EXPECTS THE SPANISH INQUISITION!
NOBODY EXPECTS THE SPANISH INQUISITION!

full data

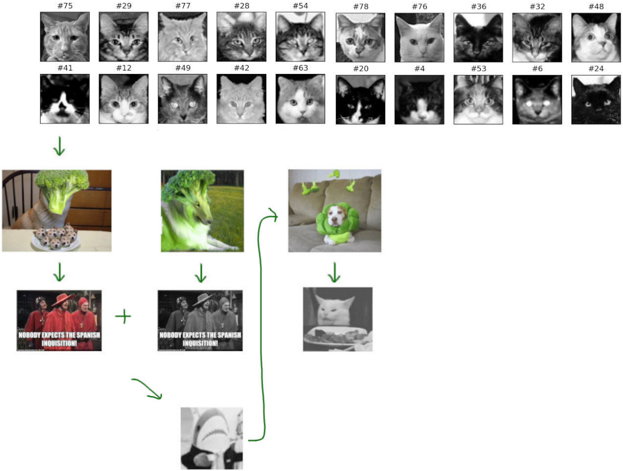
training

predict

weighted averaging/ voting

**Figure:** Schematic of boosting: re-weigh, predict, compute mismatches again (e.g. wrong animal).

# Schematic of boosting


#75
#29
#77
#28
#54
#78
#76
#36
#32
#48
#41
#12
#49
#42
#63
#20
#4
#53
#6
#24
NOBODY EXPECTS THE SPANISH INQUISITION!
+
NOBODY EXPECTS THE SPANISH INQUISITION!

full data

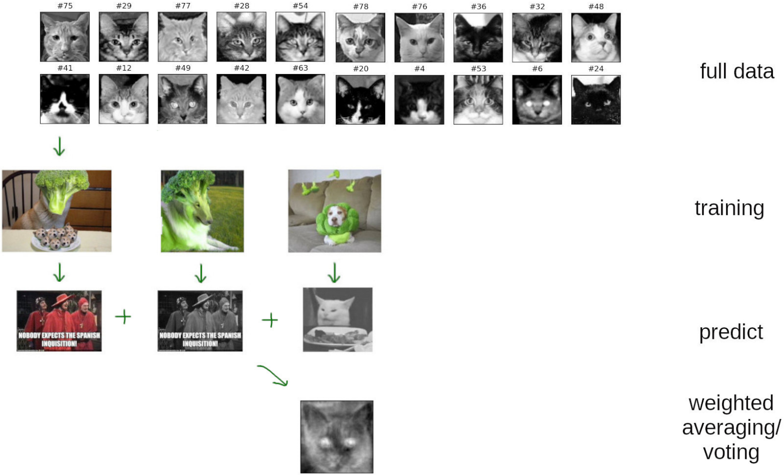
training

predict

weighted averaging/ voting

**Figure:** Schematic of boosting: train new model aiming to further reduce mismatches (e.g. predicting a cat).

# Schematic of boosting


#75 #29 #77 #28 #54 #78 #76 #36 #32 #48
#41 #12 #49 #42 #63 #20 #4 #53 #6 #24
NOBODY EXPECTS THE SPANISH INQUISITION!
+
NOBODY EXPECTS THE SPANISH INQUISITION!
+

full data

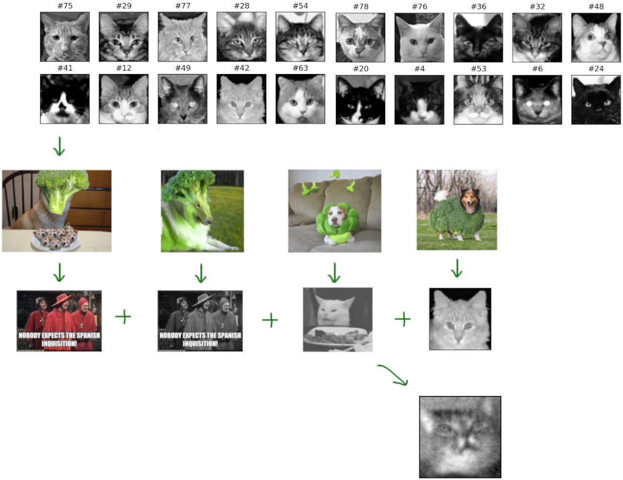
training

predict

weighted averaging/ voting

**Figure:** Schematic of boosting: adjust weight and continue...(e.g. cursed cat)

# Schematic of boosting


#75
#29
#77
#28
#54
#78
#76
#36
#32
#48
#41
#12
#49
#42
#63
#20
#4
#53
#6
#24
NOBODY EXPECTS THE SPANISH INQUISITION!
NOBODY EXPECTS THE SPANISH INQUISITION!
+
+
+

full data

training

predict

weighted averaging/ voting

**Figure:** Schematic of boosting: adjust weight and continue...(e.g. blurry Miffy)

# Schematic of boosting

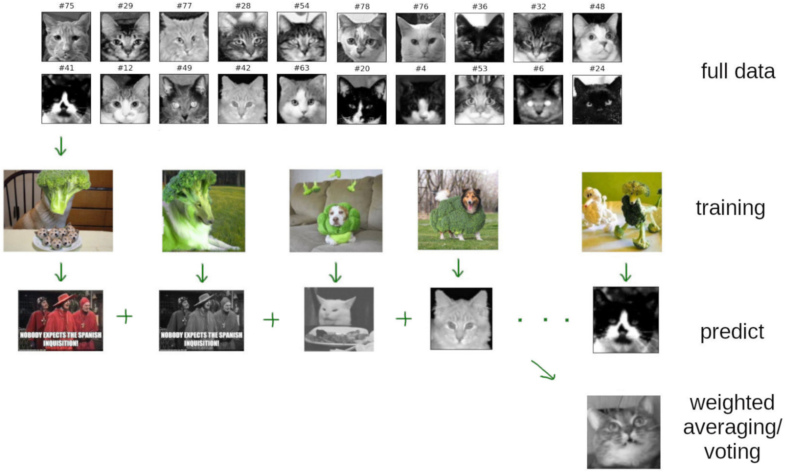

#75
#29
#77
#28
#54
#78
#76
#36
#32
#48
#41
#12
#49
#42
#63
#20
#4
#53
#6
#24
full data
training
NOBODY EXPECTS THE SPANISH INQUISITION!
+
NOBODY EXPECTS THE SPANISH INQUISITION!
+
+
. . .
predict
weighted averaging/ voting

**Figure:** Schematic of boosting: stop at some point (e.g. recovered a Miffy).

# Gradient boosting

- boosting is sequential: builds tree to build on previous issues, then take a sum, aims to reduce **bias**

  → bagging is is done in **parallel**, and averaging reduces **variance**

- aims to minimise errors in predictions by targeting the bad ones

  → i.e. an optimisation problem (again!)

  → the **gradient** part is then it uses gradient-based methods to find minimiser (e.g. SGD-based methods)

# Example: penguins data

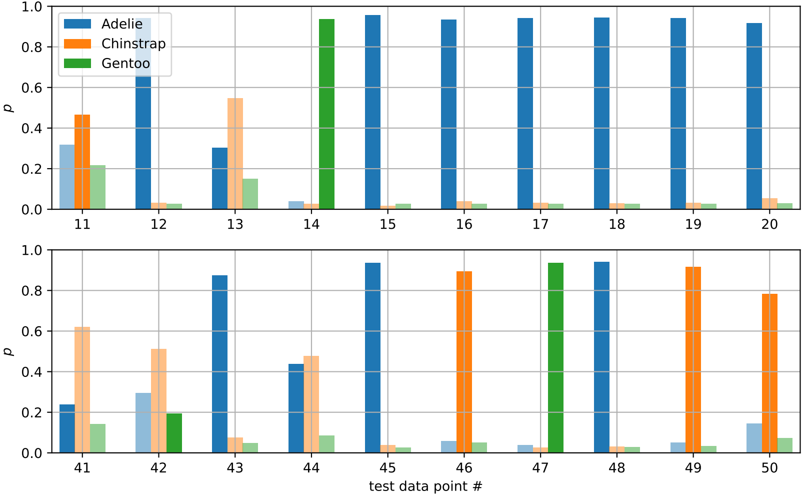


**Figure:** Probabilities associated with the predictions on test set with gradient boosting. Highlighted bar denotes truth, and note highest bar (which are higher than in random forests) is not always highlighted bar (e.g. index 13, 41, 44 as in random forests actually...)

# Demonstration

**Figure:** A forest of broc-collies.

- ideas behind ensemble methods

  → method applicable to non-trees in principle
- need to cross-validate and tune hyper-parameters accordingly!

Moving to a Jupyter notebook (e.g. do regression problems) →