CPSC 330 Lecture 13: Feature Engineering and Selection

UBC Computer Science

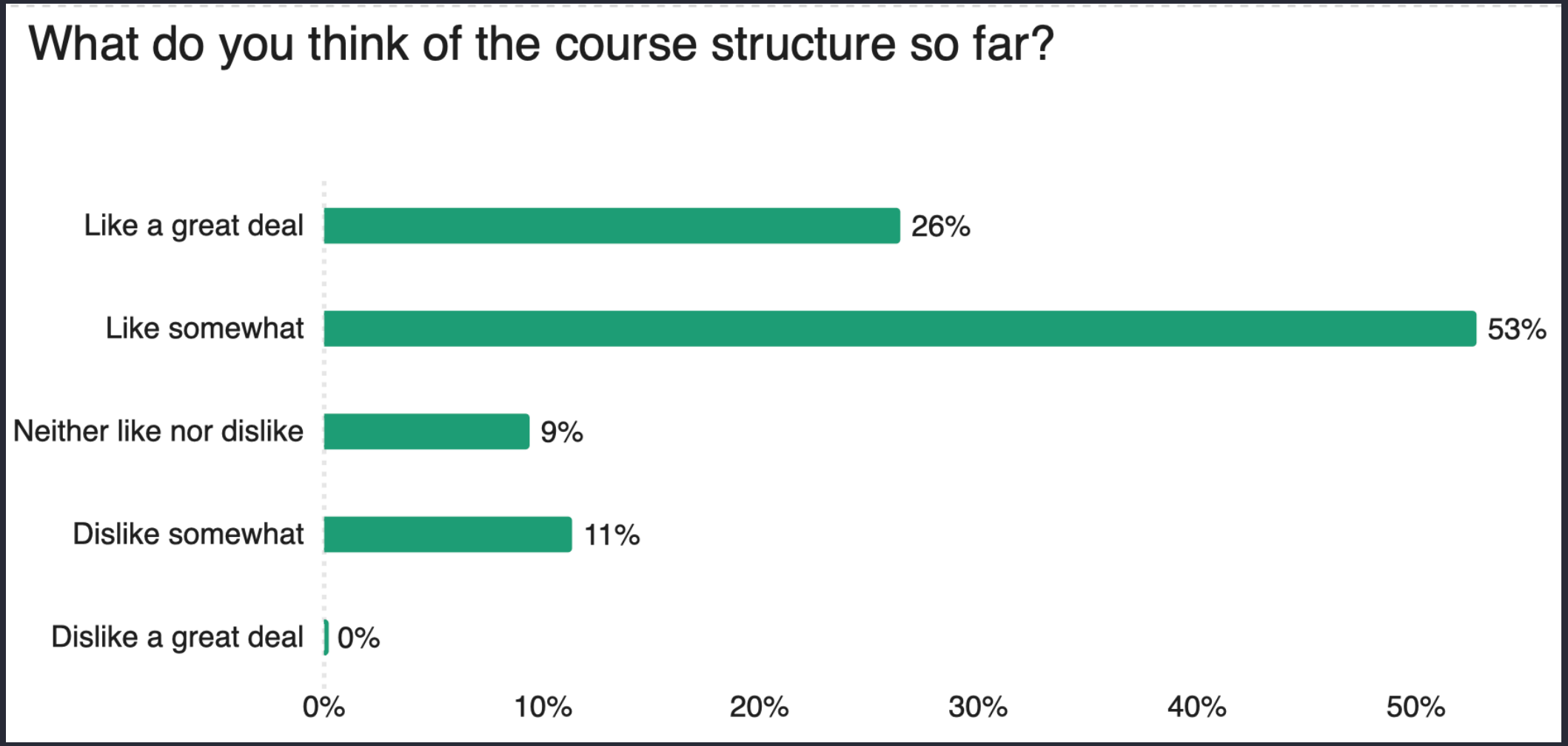
Announcements

- Midterm 1 scores are now out!
  - The class average was 80% - nice work everyone!
  - Viewing sessions (in the CBTF) will be next week.
  - Regrade requests must be submitted within the CBTF
- HW5 is due tomorrow!

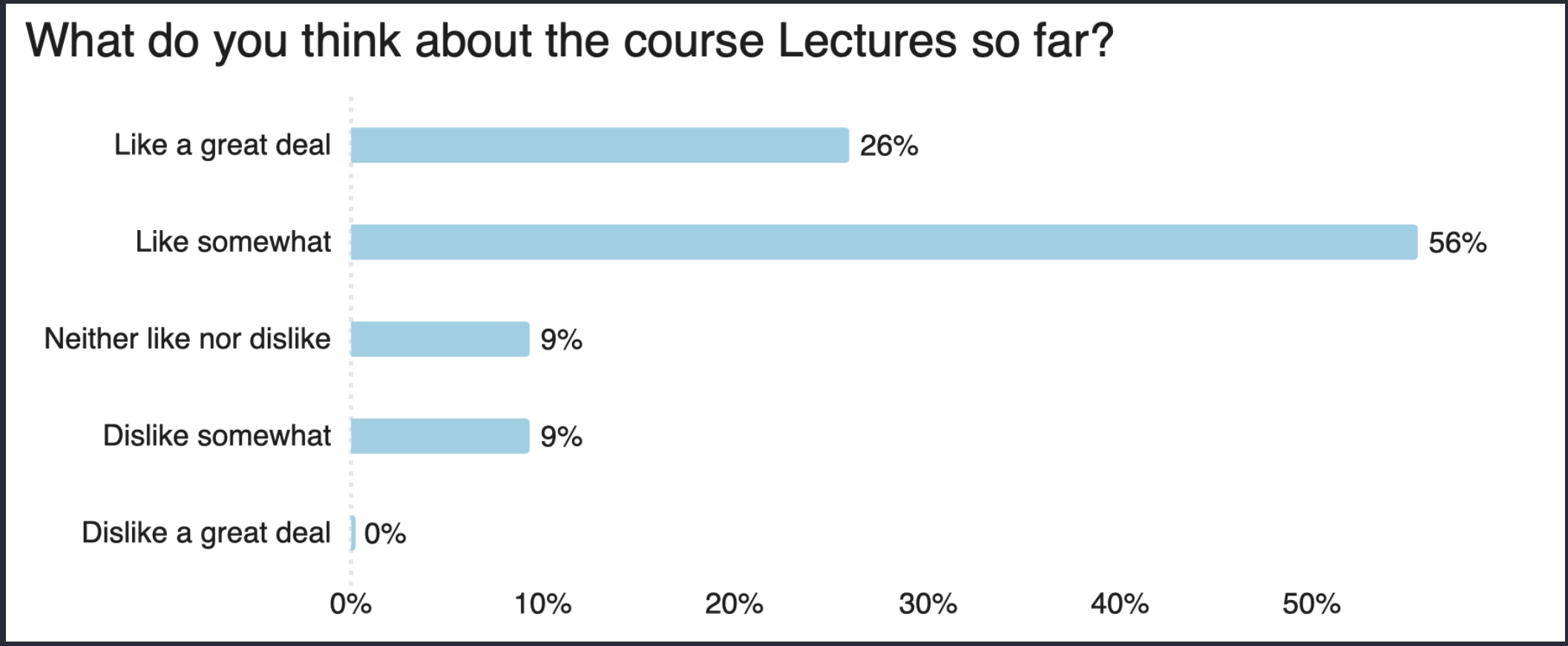
CPSC 330 Final Exam

- The final exam will be self-scheduled in the CBTF
- Exam window: Monday June 23 and Tuesday June 24th
  - Typically, sessions will be available at 9 AM, 12 PM, 3 PM and sometimes, 6 PM.
- I will announce details on booking final exams in the CBTF soon...

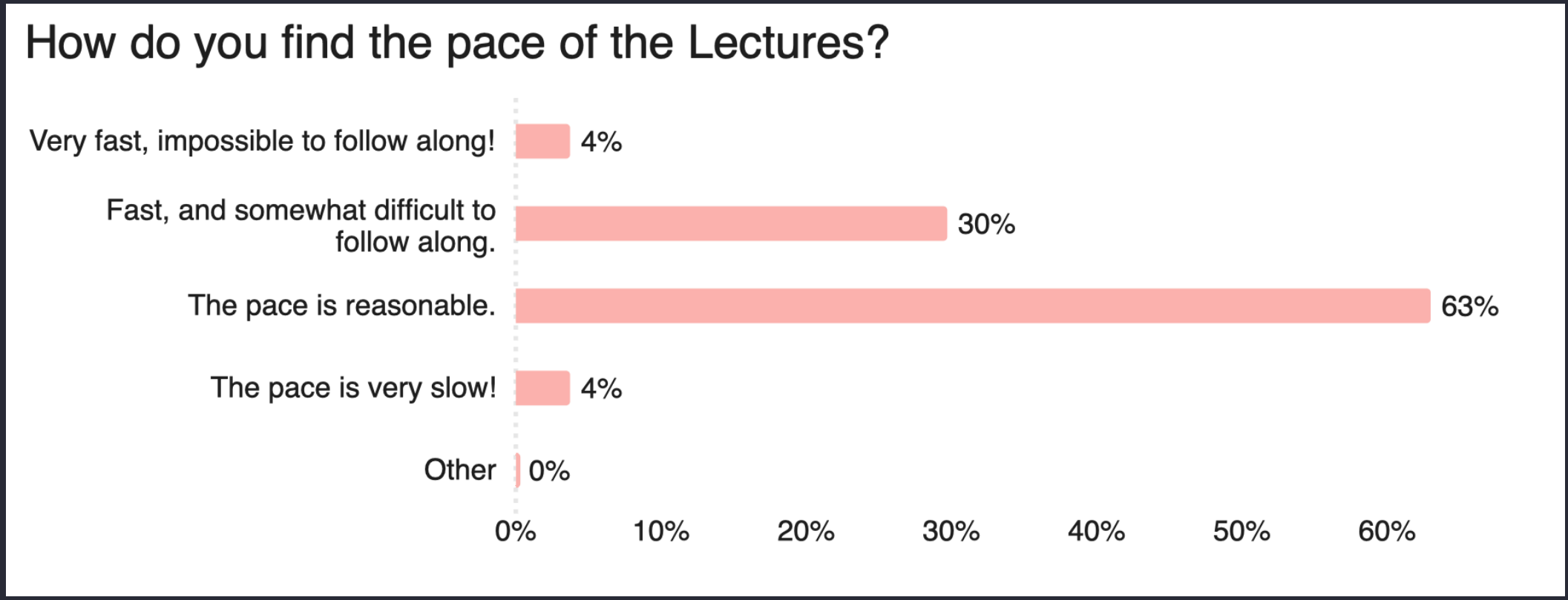
Course Feedback 1



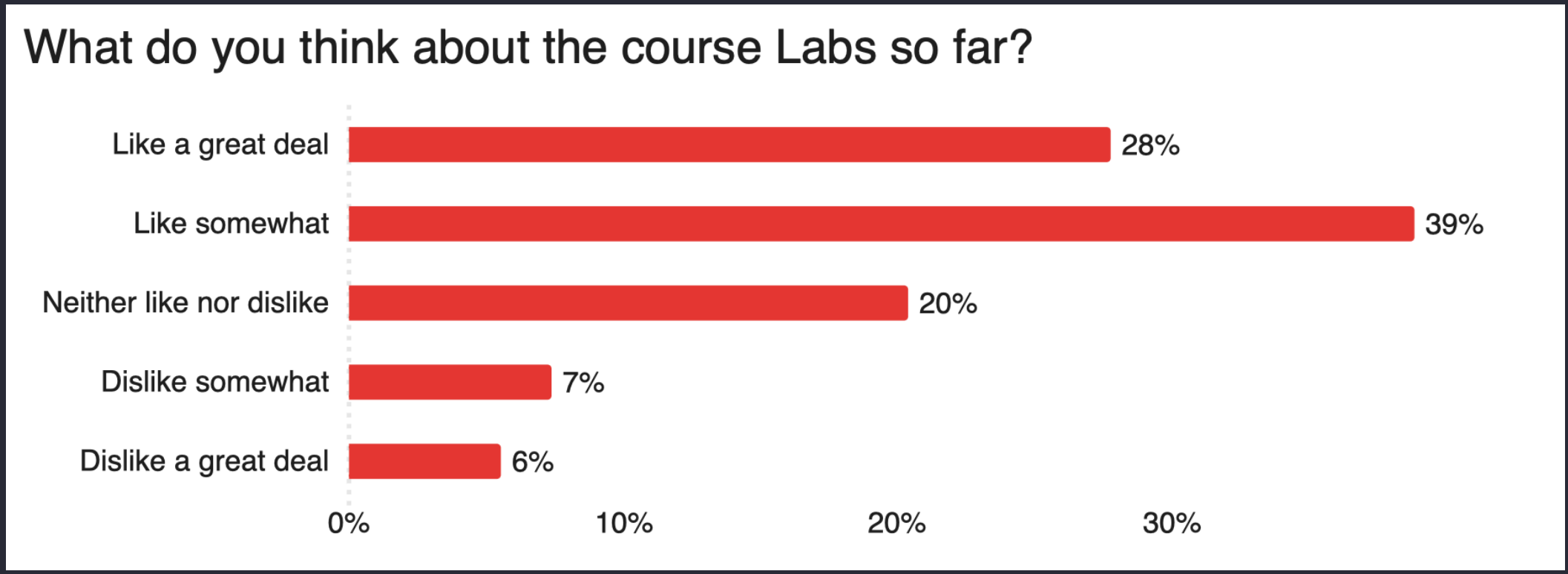
Course Feedback 2



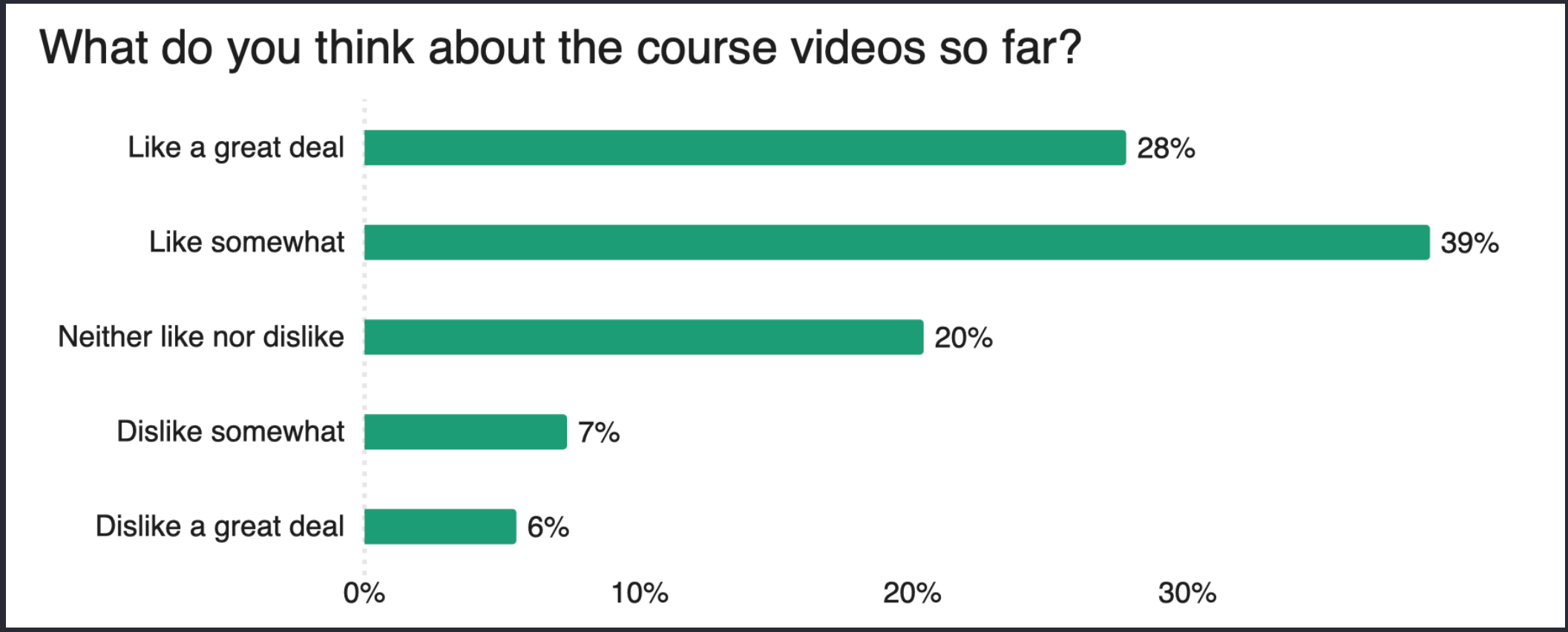
Course Feedback 3



Course Feedback 4

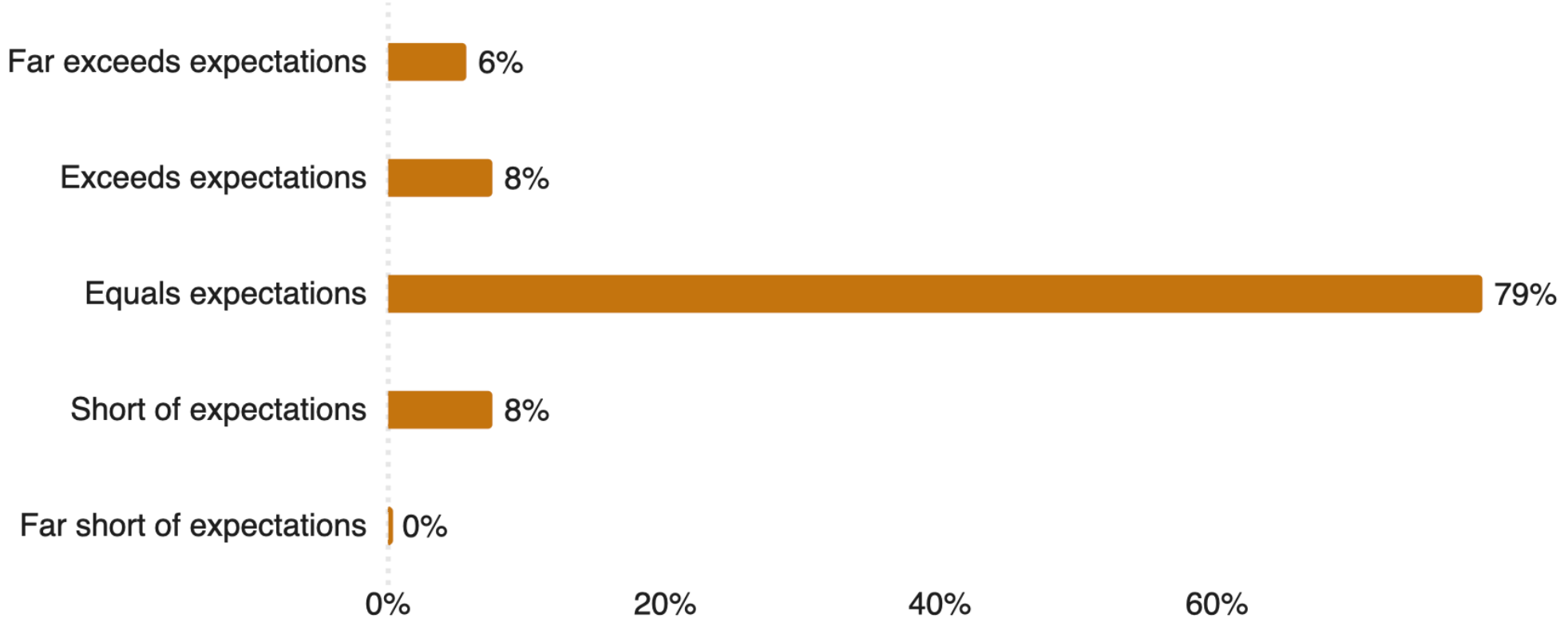


Course Feedback 5

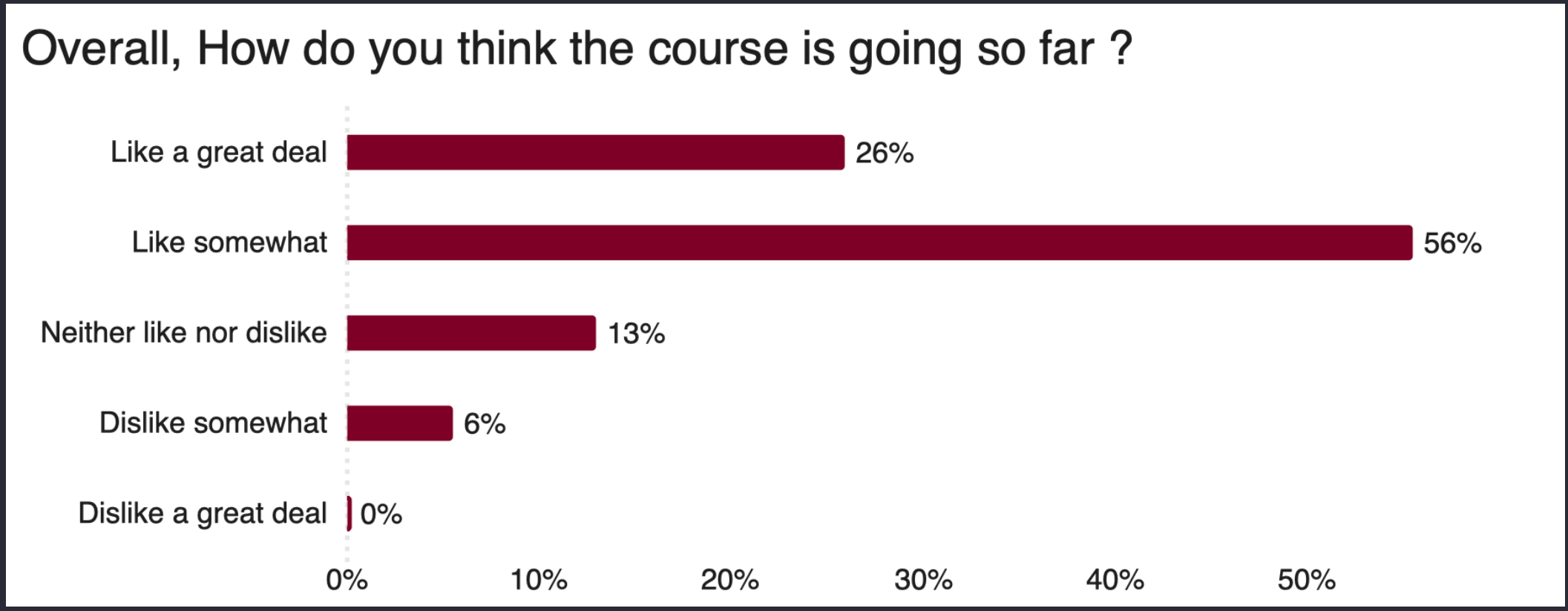


Course Feedback 6

Do you feel that you have sufficient help from the course team when you need it through the Instructor and TA office hours? The availability of the course team for office hours...



Course Feedback 7



Finishing up Feature importances

Why do we care about feature importances so much?

- Identify features that are not useful and maybe remove them.
- Get guidance on what new data to collect.
  - New features related to useful features -> better results.
  - Don’t bother collecting useless features -> save resources.

Finishing up Feature importances

- Help explain why the model is making certain predictions.
  - Debugging, if the model is behaving strangely.
  - Regulatory requirements.
  - Fairness / bias. See this.
  - Keep in mind this can be used on deployment predictions!

Extending SHAP

- Can also be used to explain text classification and image classification!

Extending SHAP

Source

Extending SHAP

- Example: In the picture below, red pixels represent positive SHAP values that increase the probability of the class, while blue pixels represent negative SHAP values the reduce the probability of the class.

iClicker Exercise 14.0

iClicker cloud join link: https://join.iclicker.com/YJHS

Suppose you are working on a machine learning project. If you have to prioritize one of the following in your project which of the following would it be?

a. The quality and size of the data

b. Most recent deep neural network model

c. Most recent optimization algorithm

Feature engineering motivation

Discussion question

- Suppose we want to predict whether a flight will arrive on time or be delayed. We have a dataset with the following information about flights:
  - Departure Time
  - Expected Duration of Flight (in minutes)

Upon analyzing the data, you notice a pattern: flights tend to be delayed more often during the evening rush hours. What feature could be valuable to add for this prediction task?

Garbage in, garbage out.

- Model building is interesting. But in your machine learning projects, you'll be spending more than half of your time on data preparation, feature engineering, and transformations.
- The *quality* of the data is important. Your model is only as good as your data.

Activity: Measuring quality of the data

- What are some properties of “good” or “bad” data?
- Along what possible dimensions could we “measure” goodness of data?

Dimension	Good Data	Bad Data	metric to measure

What is feature engineering?

> **Feature engineering** is the process of transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data.
> - Jason Brownlee

- Better features: more flexibility, higher score, we can get by with simple and more interpretable models.
- If your features, i.e., representation is bad, whatever fancier model you build is not going to help.

Some quotes on feature engineering

A quote by Pedro Domingos A Few Useful Things to Know About Machine Learning

> ... At the end of the day, some machine learning projects succeed and some fail. What makes the difference? Easily the most important factor is the features used.

Some quotes on feature engineering

A quote by Andrew Ng, Machine Learning and AI via Brain simulations

> Coming up with features is difficult, time-consuming, requires expert knowledge. “Applied machine learning” is basically feature engineering.

Better features usually help more than a better model

- Good features would ideally:
  - capture most important aspects of the problem
  - allow learning with few examples
  - generalize to new scenarios.
- There is a trade-off between simple and expressive features:
  - With simple features overfitting risk is low, but scores might be low.
  - With complicated features scores can be high, but so is overfitting risk.

The best features may be dependent on the model you use

- Examples:
  - For counting-based methods like decision trees separate relevant groups of variable values
    - Discretization makes sense
  - For distance-based methods like KNN, we want different class labels to be “far”.
    - Standardization
  - For regression-based methods like linear regression, we want targets to have a linear dependency on features.

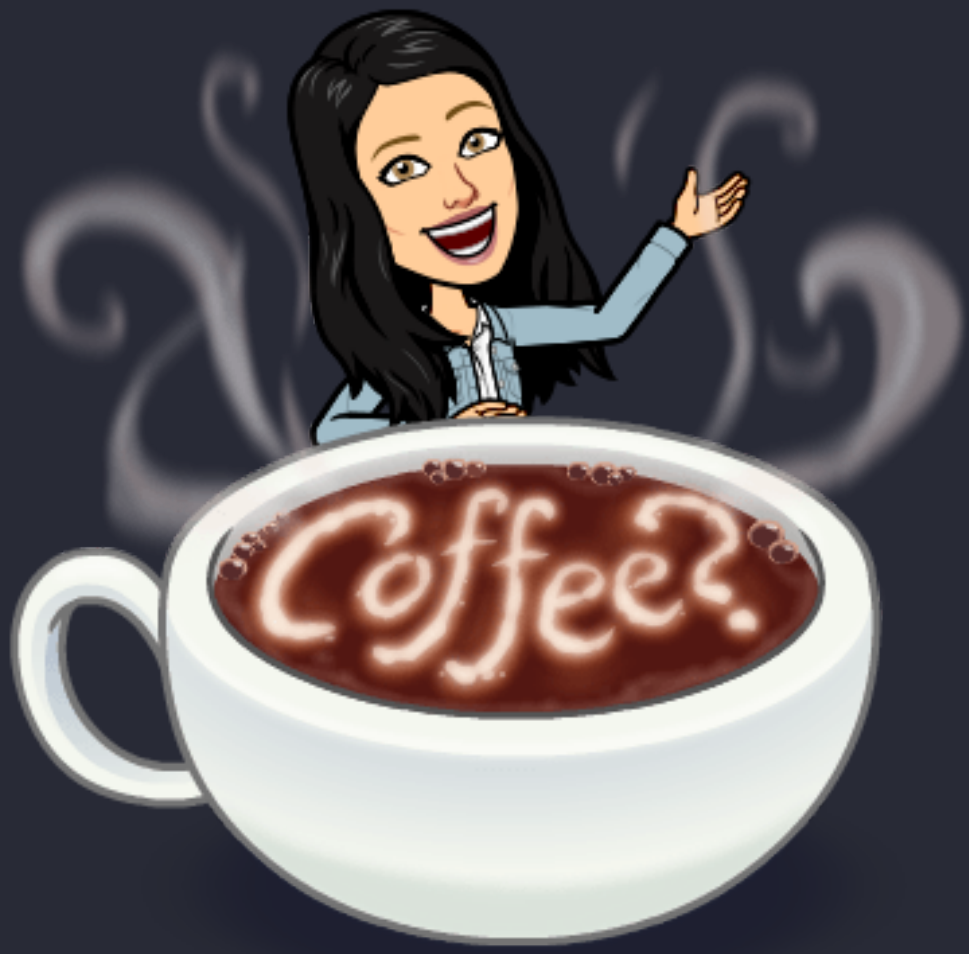
Motivating Feature Engineering

Questions:

- What are two possible ways we could “engineer” features?
  - Think broadly and philosophically rather than an implementation...

Break

Let’s take a break!

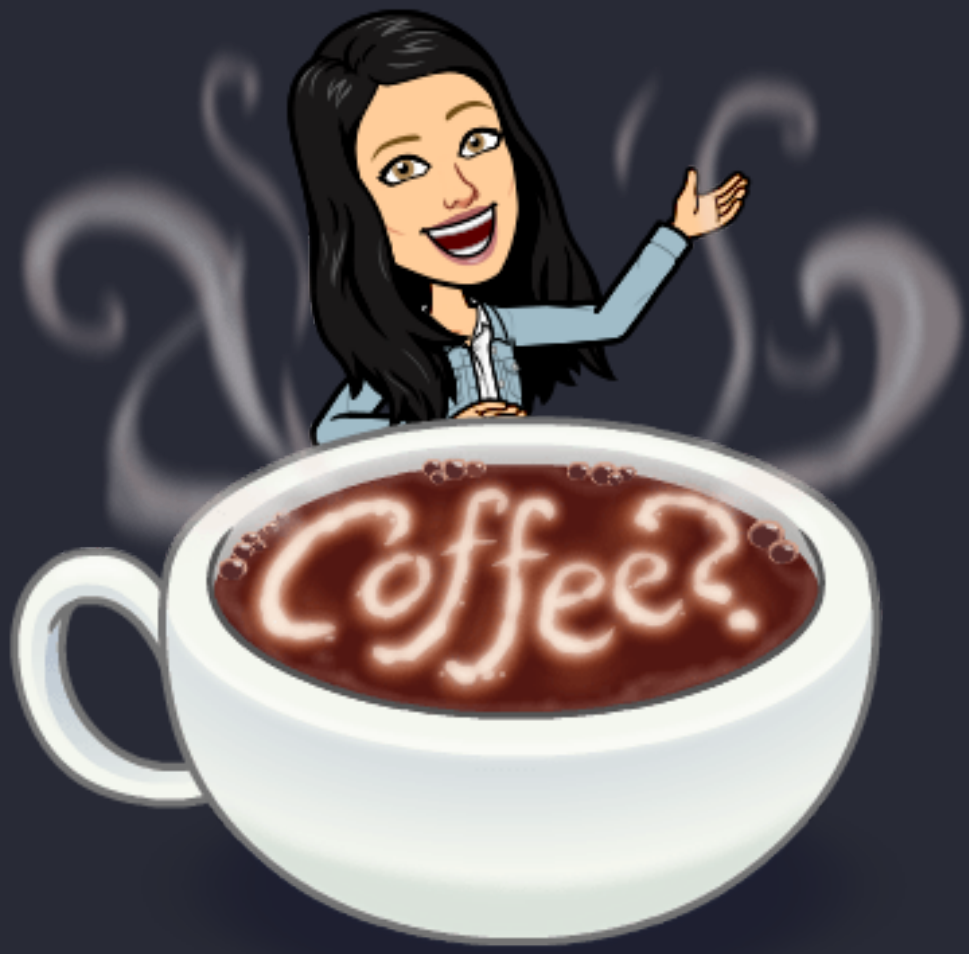


Group Work: Class Demo & Live Coding

For this demo, each student should click this link to create a new repo in their accounts, then clone that repo locally to follow along with the demo from today.

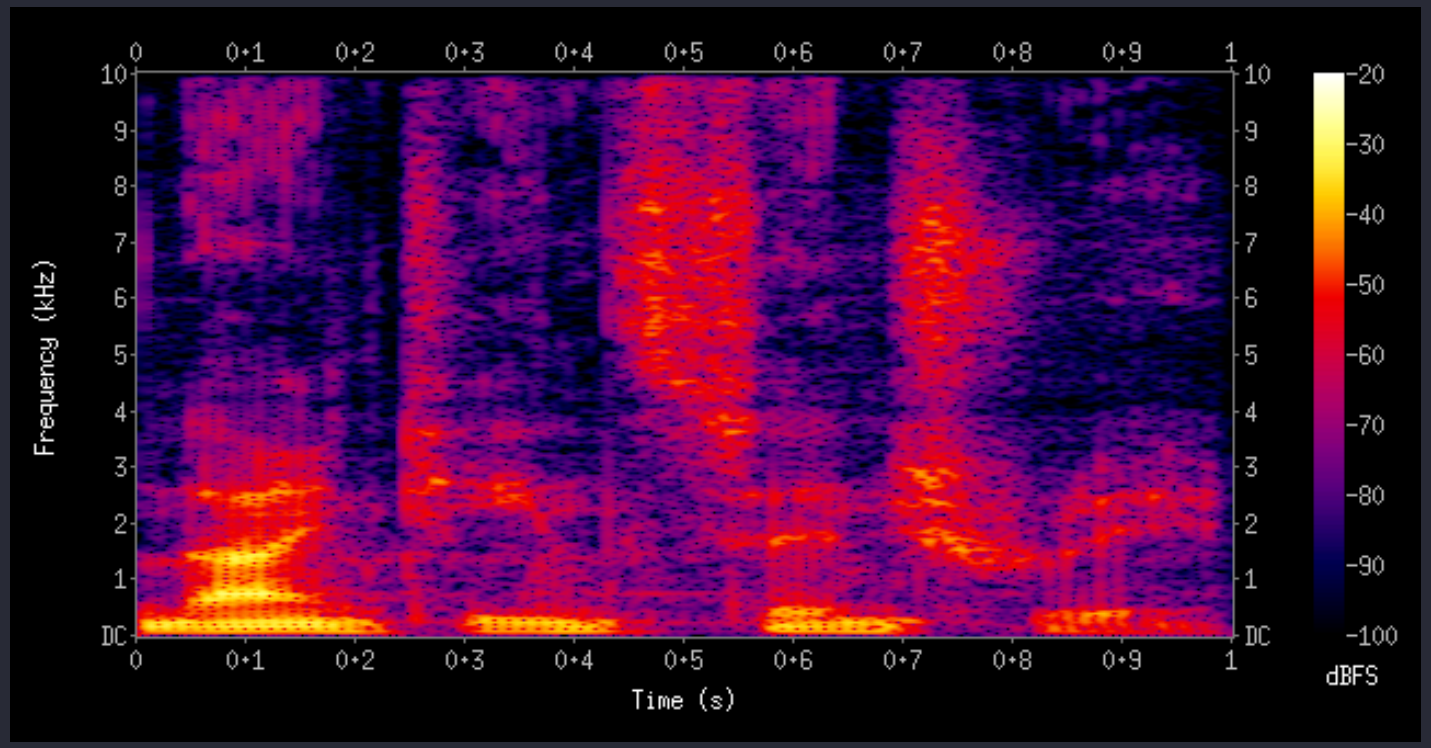
Break

Let’s take a break!



Domain-specific transformations

In some domains there are natural transformations to do: - Spectrograms (sound data) - Wavelets (image data) - Convolutions



Source

Developing intuition about Feature Engineering

...

(iClicker) Exercise 14.1

iClicker cloud join link: https://join.iclicker.com/YJHS

Select all of the following statements which are TRUE.

- a. Simple association-based feature selection approaches do not take into account the interaction between features.
- b. You can carry out feature selection using linear models by pruning the features which have very small weights (i.e., coefficients less than a threshold).
- c. The order of features removed given by `rfe.ranking_` is the same as the order of original feature importances given by the model.