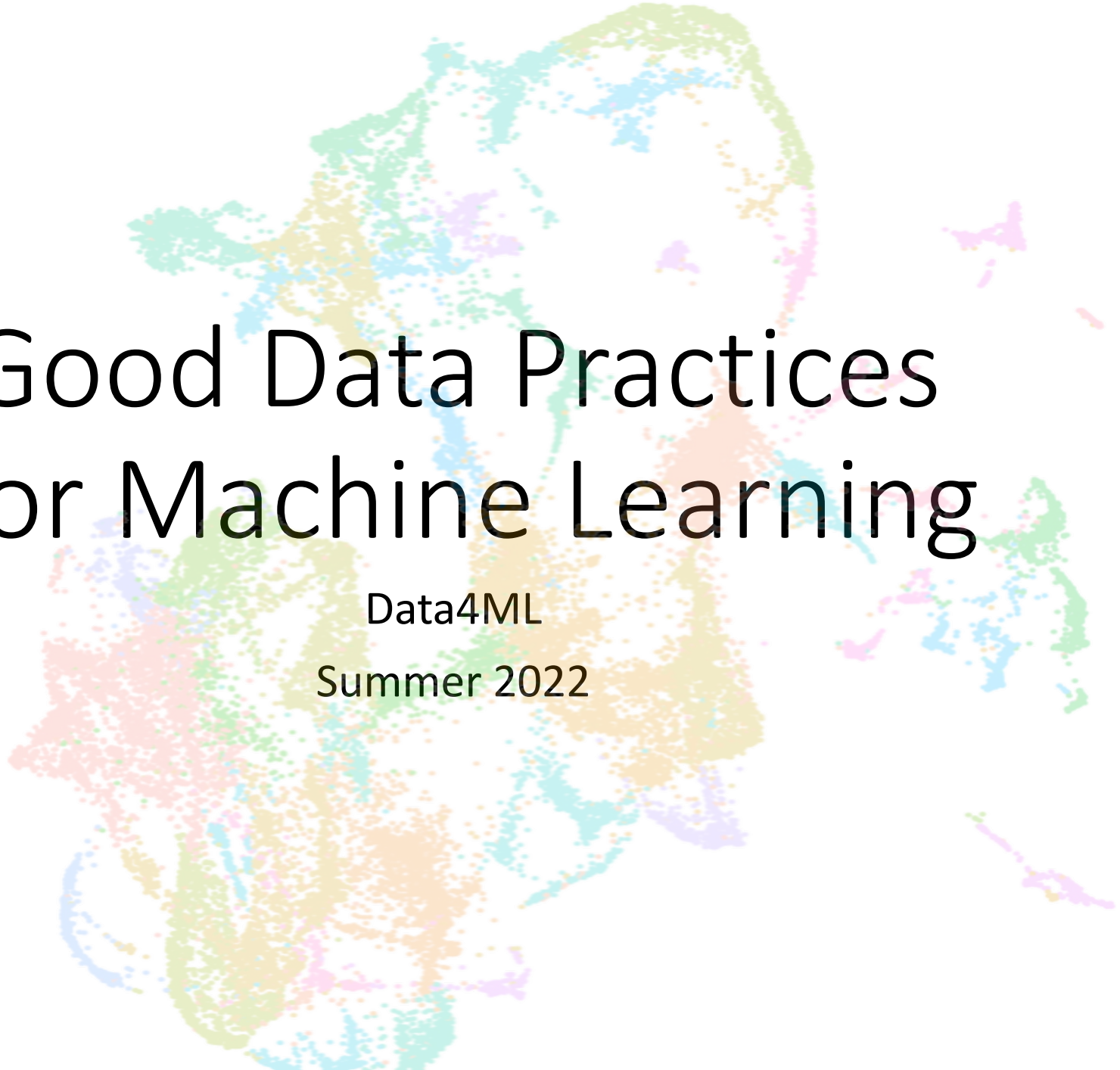


Good Data Practices For Machine Learning

Data4ML

Summer 2022

It’s All About Data

- Good Machine Learning starts with good datasets
- What makes a dataset ‘good’
  - Reproducible
  - ‘FAIR’ – Finable Accessible Interoperable and Reusable
- How to plan out your data taking
- What makes a data set ‘AI-Ready’

Reminder

- Clear Data Splits
  - Designate some examples as training data
  - Designate some examples as testing data
- Structed Data is ‘Tidy’ - https://vita.had.co.nz/papers/tidy-data.pdf
  - Every column is a variable.
  - Every row is an observation.
  - Every cell is a single value
- The Dataset is Clean
  - No missing values or ‘Nans’
  - No ‘bad’ data
  - Consistent as Possible

ML Data Content

- ML models are programed with data
- Beyond technical requirements the content of the data is important
- What is in your data determines how your model works
  - Will it generalize
  - Will it be able to train
  - How accurate will it be

Confounders

- Beyond the format of the data good datasets have limited irrelevant confounders
- ML picks up on patterns, so you don’t want to add your own that might not be relevant to the question you’re asking

Columbia River

Riverbend

Confounders - Example

- Study interested in understand Phenotype differences in two populations
- Rulers placed in image for scale – Good for Consistency
- ID numbers include in image – Good for reproducibility

Columbia River

Riverbend

Confounders - Example

- Ruler color differs between collection site!
- ID numbers differ between collection site
- Both effects were learned by a classifier trying to predict the phenotype from the image!

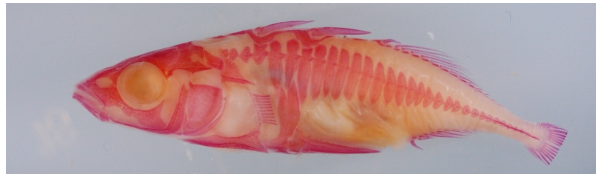
Columbia River

Riverbend

Confounders - Lessons

- You may need to clean your data before training an ML algorithm
- Sometimes what's best for reproducibility is bad for ML algorithms
  - If you inject unique features into your data for each subject or experiment it can help you keep organized
  - Expect ML models to learn these features too! You'll need to remove them before training.
- This happens in other kinds of data too not just images

Columbia River



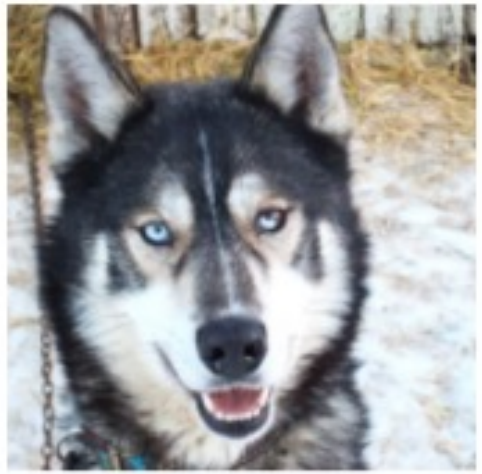
Riverbend

What's the Confounder? Husky or Wolf

https://arxiv.org/abs/1602.04938

Confounders

- Is it a problem?



(a) Husky classified as wolf

(b) Explanation

Figure 11: Raw data and explanation of a bad model's prediction in the "Husky vs Wolf" task.

	Before	After
Trusted the bad model	10 out of 27	3 out of 27
Snow as a potential feature	12 out of 27	25 out of 27

Table 2: "Husky vs Wolf" experiment results.

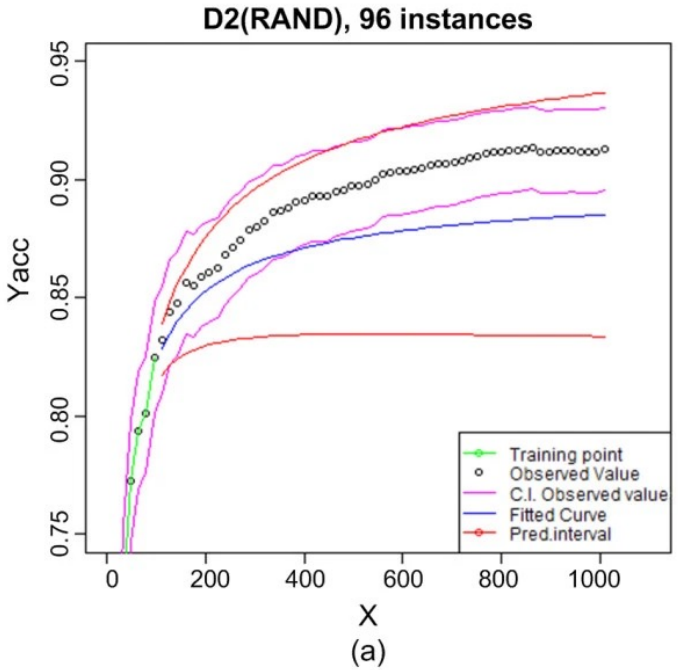
https://arxiv.org/abs/1602.04938

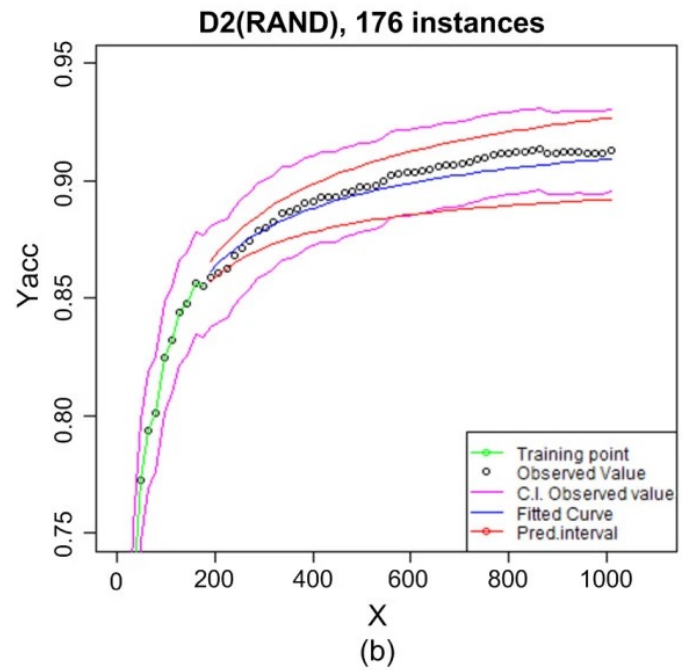
How much data do you need for ML

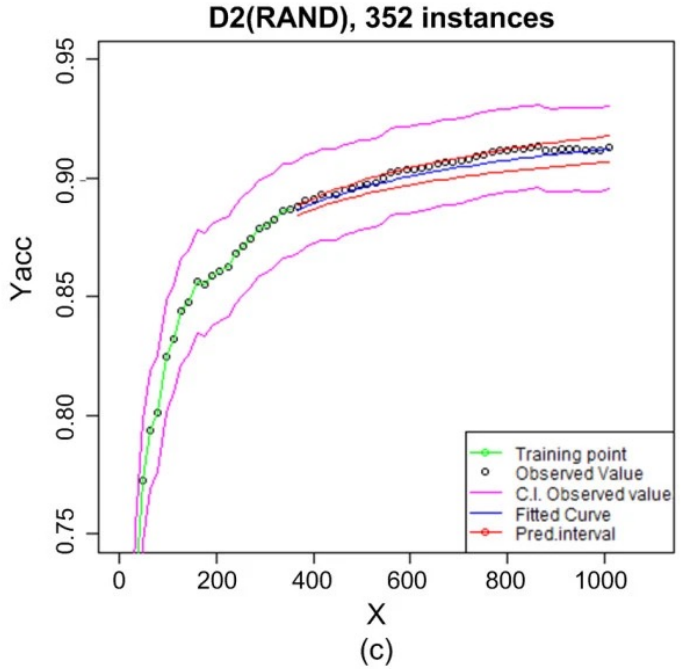
- A random answer on Quroa
  - At a bare minimum, collect around 1000 examples. For most "average" problems, you should have 10,000 - 100,000 examples.
- Statements like this are common, and completely incorrect
- **Problem:** There is no correct answer
  - ML finds patterns, the more obvious the pattern the less data you need
  - How good do you need your algorithm to work?
    - Better than people at something people are really good at like object recognition
      - A lot of data
    - Better than nothing
      - A little data
  - Do patterns exist?
    - If not then no amount of data will help

Learning Curves

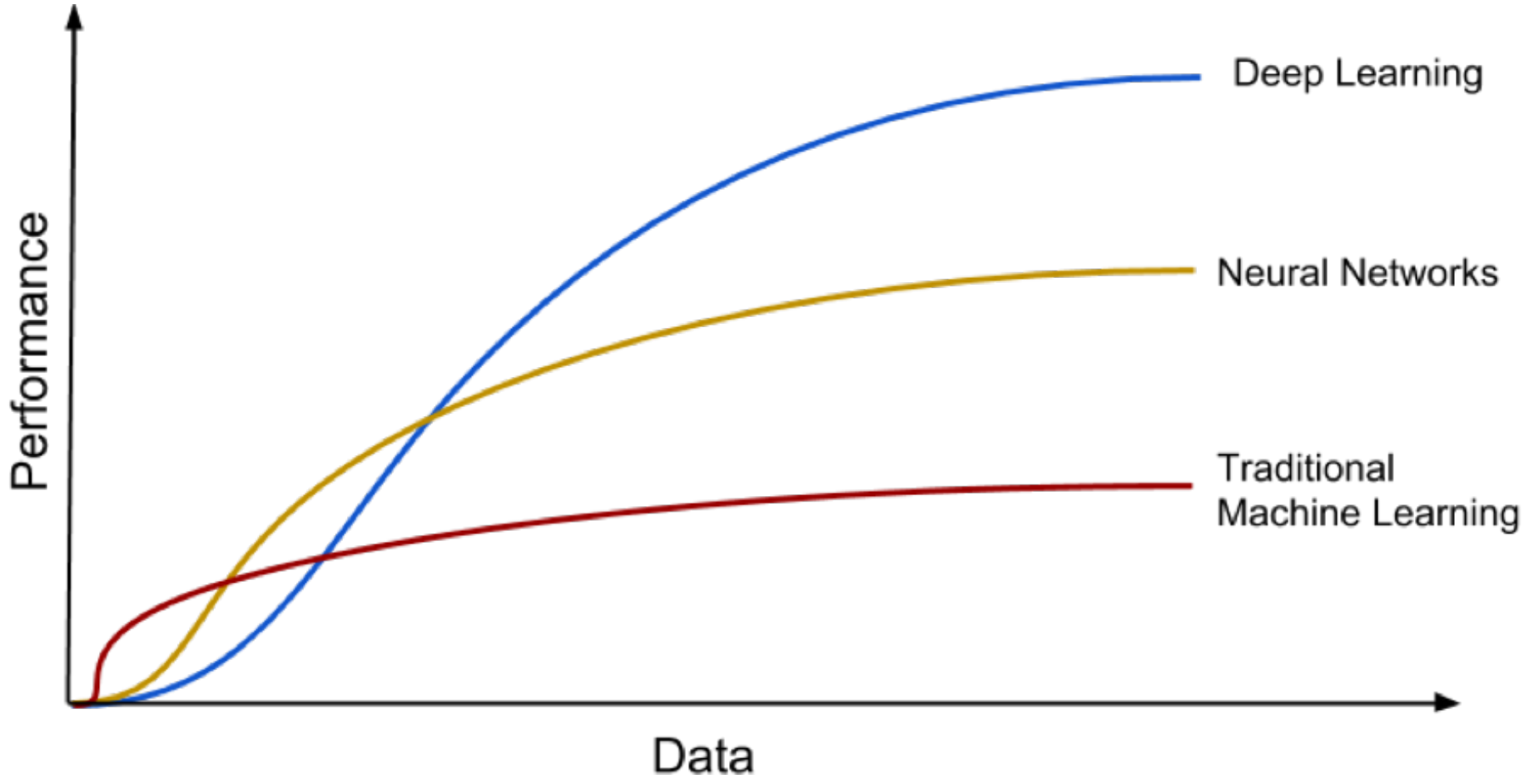
- **Predicting sample size required for classification performance** https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/1472-6947-12-8
- You'll need some data to estimate how much more you'll need to reach a given accuracy
- ML algorithms improve with data until they reach a saturation point
- Can be useful to train models early in data taking to get an idea of how much more you'll need to reach a certain goal







Models have different learning curves



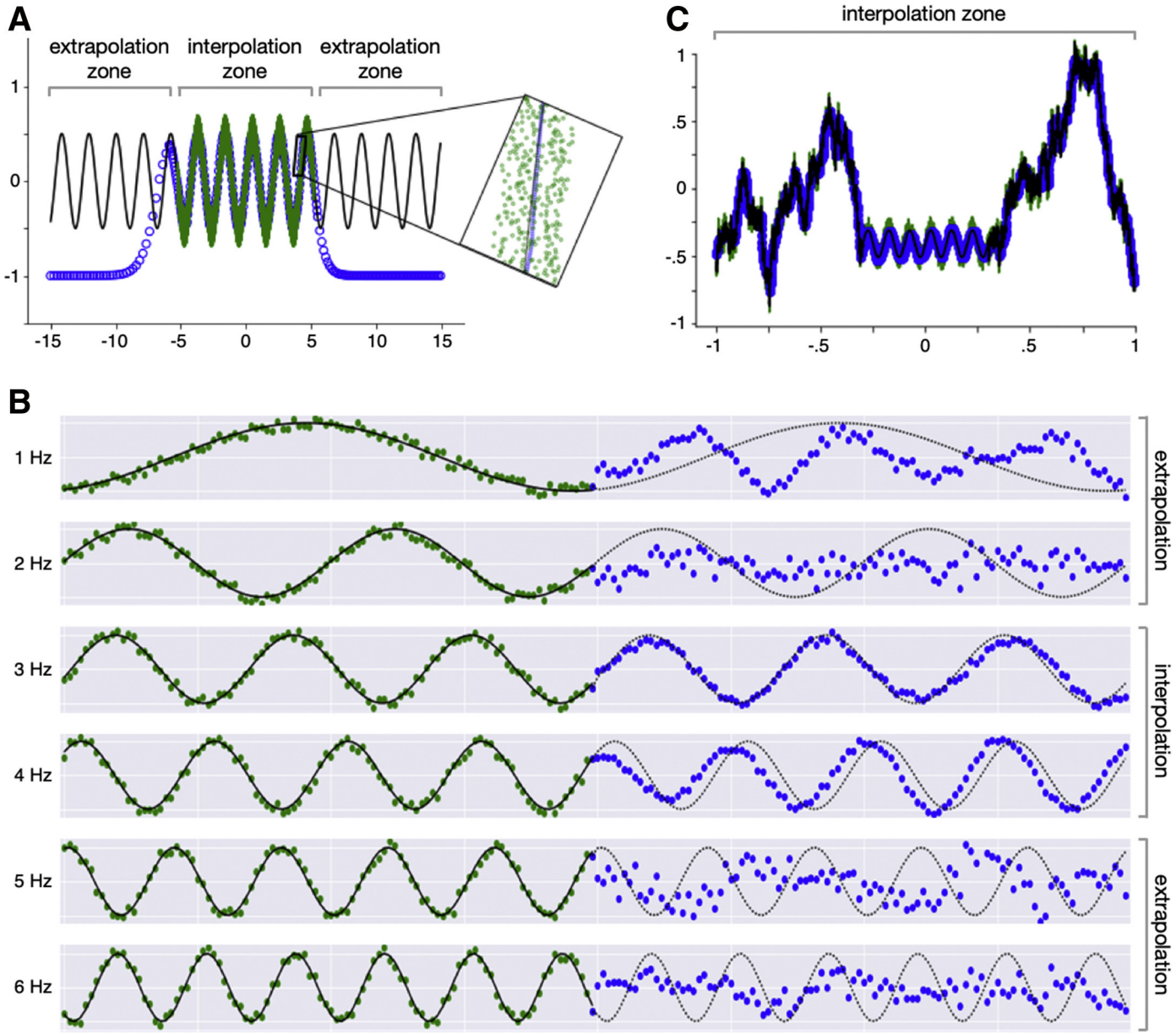
Data Splits – How Much

- Recommendation split data into 80% training 20% testing
- What's the tradeoff?
  - More testing data means better estimates of how well your model is working
  - More training data means the better your model works.

- Another Recommendation – if this split doesn't give you enough testing data consider a different validation strategy
  - Cross-validation – K-Folds or leave one out
  - Both the above give you estimates of how well some models

Extrapolation

- ML does a lot of neat things, but it isn't magic
- Unless specified treat ML algorithms as universal function approximators
  - i.e. 'Linear' regression is not a universal function approximator
- If you're collecting data make sure you collect 'representative' samples



Dataset Bias

- ML classifiers and algorithms will do there best to learn what they see in there training data.
- Example
  - Inputs: Room Temperature
  - Target: A random draw from a bag with 5 red balls and 5 green balls
  - This Input is completely un-related a well trained ML model will predict probabilities of ~50% red ~50% green
- Example 2 – Same except
  - Target: A random draw from a bag with 10 red balls and 5 green balls
  - A well-trained ML model will predict probabilities of ~66% red ~33% green
- If we turn our probabilities into decisions by using the >0.5 rule
  - Case 1: >50% is probably still random equal prediction of red and green
  - Case2: Red is always > 50% prediction is red all the time
- When we train a model we get to decided what data we include
  - We can balance the classes, leave them as is, or over-sample a class
  - In our random example above this lets us pick an ML output anywhere from 100% red to 100% green
- **Whenever there is uncertainty your class balance will change your result!**
- **Be-careful about think of these results as the truth. Think of it as the truth of the training data you selected**

ML Experiment

1) Define your problem
2) Gather preliminary or existing data
3) Fit a test model
4) Use a learning curve fit to estimate how much more data you might need
5) Use importance or attribution to look for confounders that might affect your data
6) Gather data and experiment with new models
7) Deploy/Publish – Return to 6

Case Study: Why Amazons Automated Hiring Tool Discriminated Against Women

- 2014 Amazon built a model to read resumes and predict who will be successful
  - Inputs: Resume Terms
  - Outputs: Hired/Not Hired
- Uncertainty in hiring is large
- Dataset was historical and very Male – Gender was excluded so in theory the model shouldn't know
- Confounders – "downgraded resumes that included the word "women's" — as in "women's rugby team.""
- In search of objectivity the model reproduced the training data's gender bias
- The model was scraped before put into production

https://www.aclu.org/blog/womens-rights/womens-rights-workplace/why-amazons-automated-hiring-tool-discriminated-against

Conclusions and Homework

- Conclusions
  - It’s important to think about:
    - Possible confounders
    - Dataset bias and balance
    - Are you going to use your model on new data that isn’t representative in your training data
- Homework:
  - 1. Use penguins.csv to make a histogram of body mass by species
  - 2. Train a Random Forest Classifier from penguins.csv to predict species
    - Split dataset
    - Select numeric predictors
    - Fit a random forest.