Course Overview

CSci 5525: Advanced Machine Learning

Instructor: Nicholas Johnson

September 5, 2023

General Information

- Course Number: CSci 5525
- Class: Tue, Thu 4:00-5:15 PM
- Location: Mechanical Engineering 212

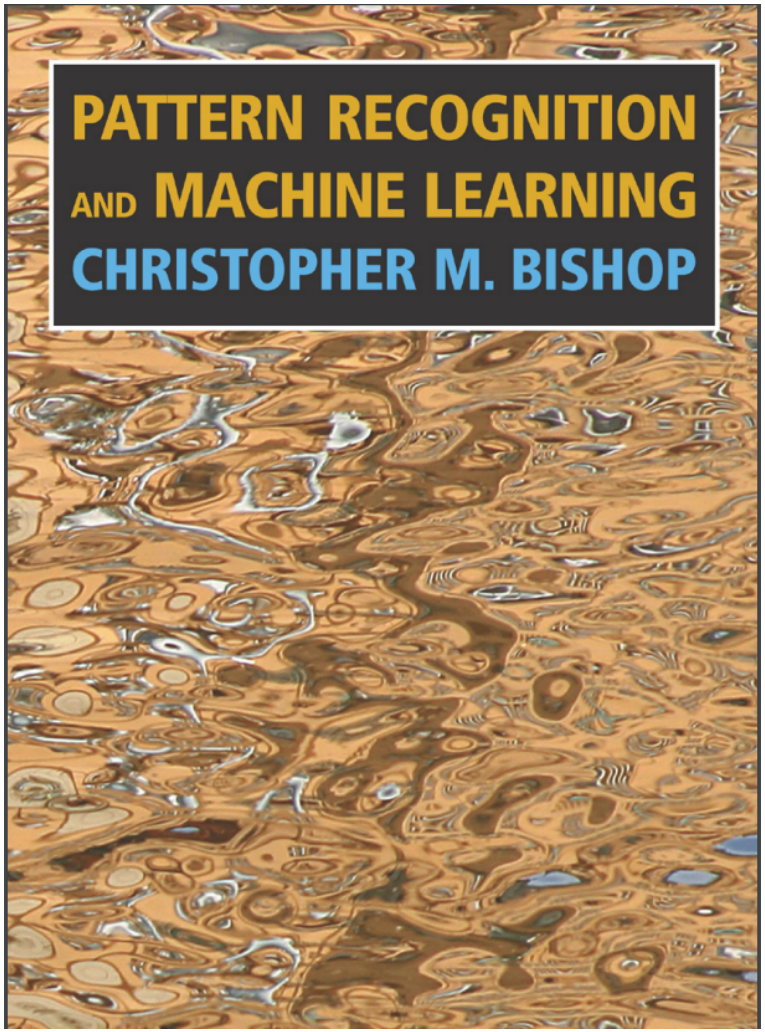
- Instructor: Nicholas "Nick" Johnson
- TAs: Harshavardhan "Harsha" Battula

- Office Hours:
  - Nick: Keller Hall 6-196 Tue 5:30 - 7:30 PM
  - Harsha: TBD

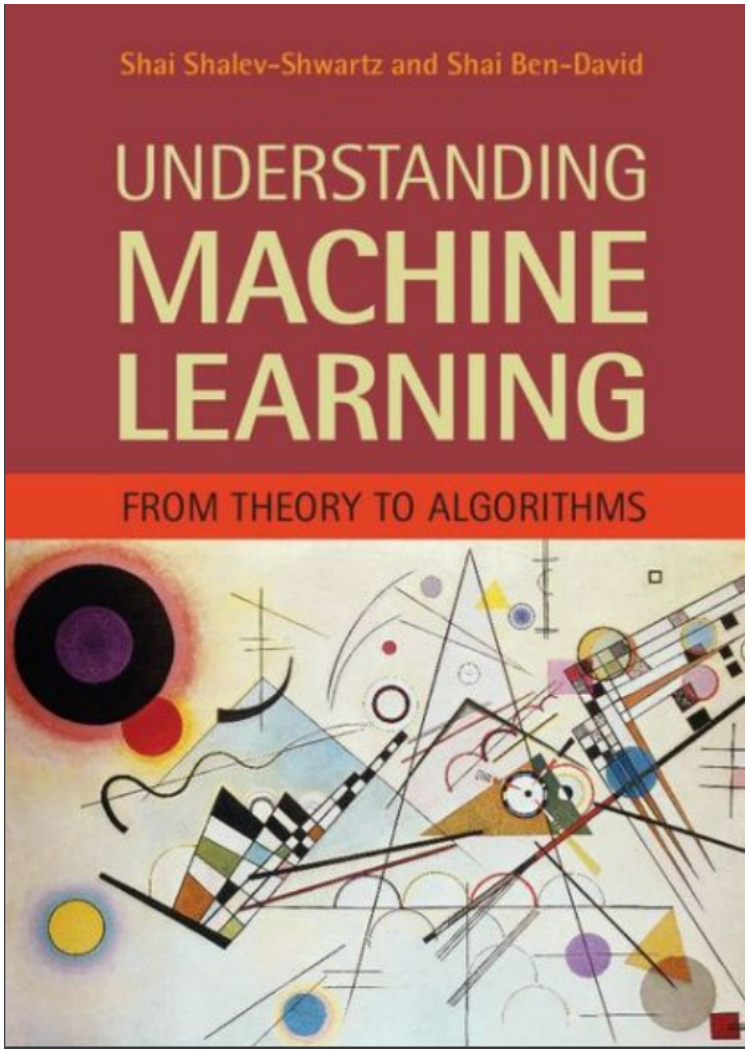
- Canvas page: https://canvas.umn.edu/courses/391265

- Email:
  - Nick: njohnson@cs.umn.edu
  - Harsha: battu018@umn.edu

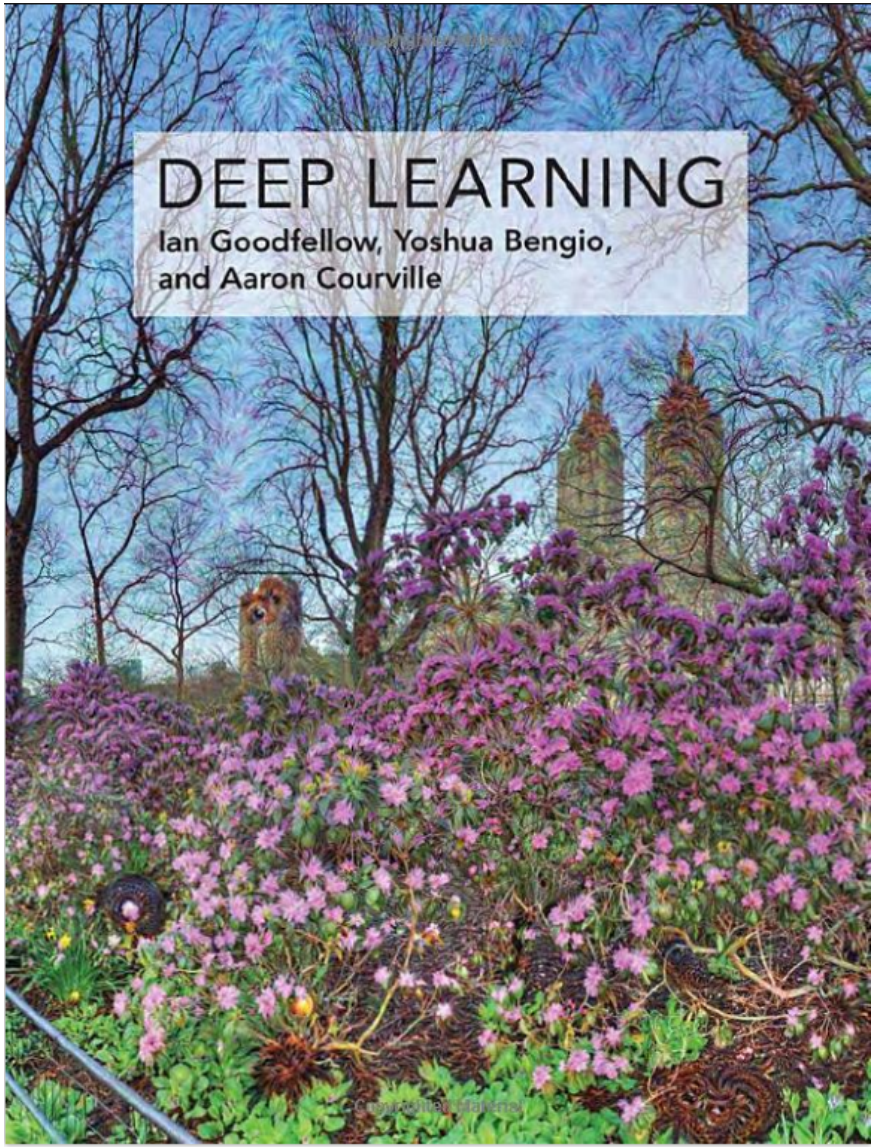
Books


PATTERN RECOGNITION
AND MACHINE LEARNING
CHRISTOPHER M. BISHOP

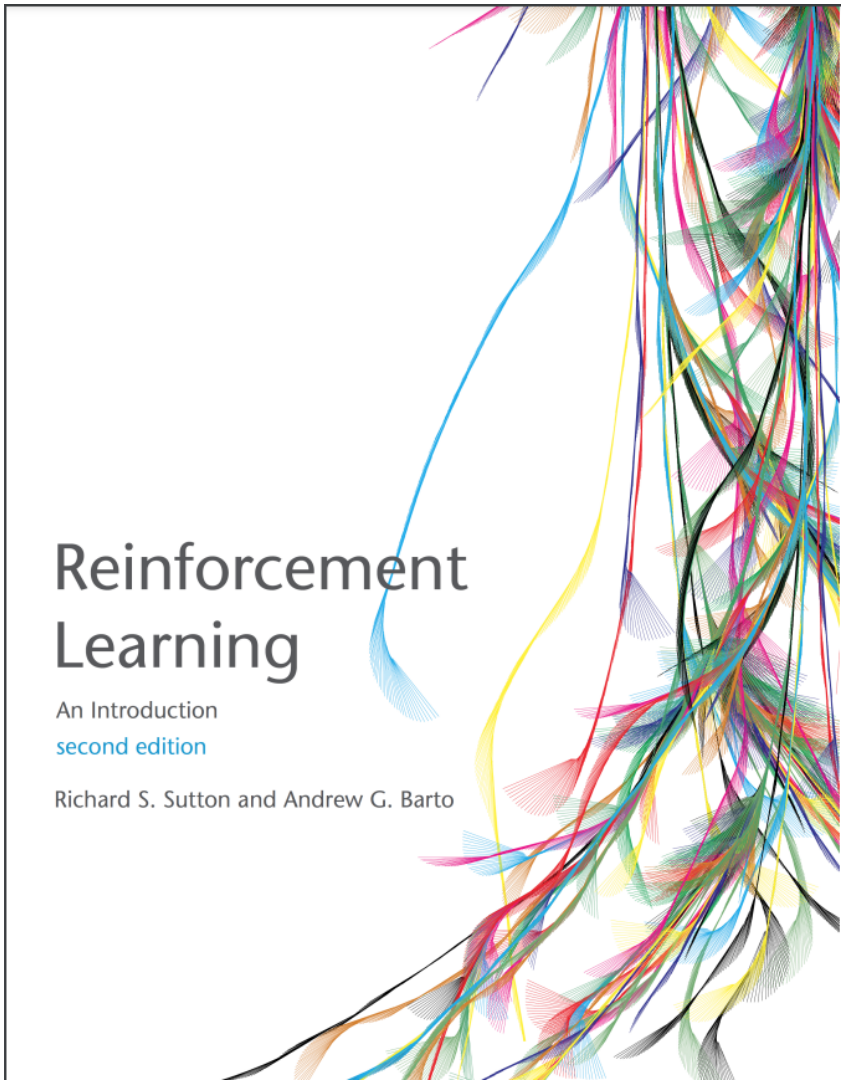
Books


Shai Shalev-Shwartz and Shai Ben-David
UNDERSTANDING
MACHINE
LEARNING
FROM THEORY TO ALGORITHMS

Books


DEEP LEARNING
Ian Goodfellow, Yoshua Bengio,
and Aaron Courville

Books


Reinforcement
Learning
An Introduction
second edition
Richard S. Sutton and Andrew G. Barto

Assignments

- **Please** read the syllabus carefully
- 6 Homeworks
- Project: proposal, progress, final
- Paper review
- Each assignment due at 11:59 PM central time on due date
- Each assignment must be done individually (high-level discussions are OK)
- You must complete all assignments with a score > 0 on each to pass the class

Homeworks

- There will be 6 homeworks
  - HW0 is on background/preparation; must be completed/submitted to remain enrolled
  - All submissions must be in PDF format
  - All programming must be in Python 3.6+

Homeworks: Late Submission Policy

- You have a total of 5 grace days
- You can choose to use them as convenient to delay one/more homework submissions
- Cannot use grace days for project or paper review assignments
- Delay gets rounded up to a day
  - Late by 45 mins $\equiv$ late by a (grace) day
  - Late by 23 hr 45 mins $\equiv$ late by a (grace) day
- Delays beyond the grace days:
  - Late by 0-24 hrs: 50% of actual score
  - Late by 24-48 hrs: 25% of actual score
  - Late by more than 48 hrs: Will receive a zero

Project

- 3 components
  - Proposal
  - Progress
  - Final
- Proposal and progress components have a written and peer-review part
- Final component has a written, pre-recorded video presentation, and peer-review part
- Allowed to use programming tools/open-source code
- Must implement at least 1 algorithm yourself (or significantly modify existing code)

Project Proposal

- Written proposal
  - At most 1 page
  - Describe motivation, problem to solve, why use ML, initial idea on how to solve it, data needed, etc.
- Proposal peer review
  - Review and provide feedback on at most 2 other written proposal reports
  - Rubric will be given

Project Progress Report

- Written progress report
  - At most 2 pages
  - Describe work done thus far (e.g., solutions attempted, initial results, challenges faced, next steps, expected outcome, how you used peer-review feedback, etc.)
  - Make sure to also submit and describe the code you've written yourself
- Progress report peer-review
  - Review and provide feedback on at most 2 other written progress reports
  - Rubric will be given

Project Final Report

- Written final report
  - At most 5 pages + references (must be self-contained)
  - Make sure to also submit and describe the code you've written yourself
- Final report pre-recorded video presentation
  - At most 10 mins
  - Should supplement written report; a demo would be great
- Final report peer review
  - Review and provide feedback on at most 2 other written final reports

Paper Review

- Review 1 state-of-the-art paper from AI/ML conference
  - ICML, NeurIPS, AAAI, ICLR, CVPR, IJCAI, NAACL, etc.
- Assignment consists of written review and pre-recorded video presentation
- Written review:
  - At most 2 pages
  - Overview including: problem, motivation, technique/algorithm introduced, key results
  - About half of review must be a discussion of the paper
- Pre-recorded video presentation:
  - At most 10 mins
  - Should supplement written report
  - We will post the videos to the class for extra credit opportunities

Grading

- Homeworks: 55 %
- Project: 35 %
- Paper review: 10 %

- Grading is absolute (i.e. not on a curve)
- For S/N grading, a satisfactory grade (S) requires at least a C-

Topics

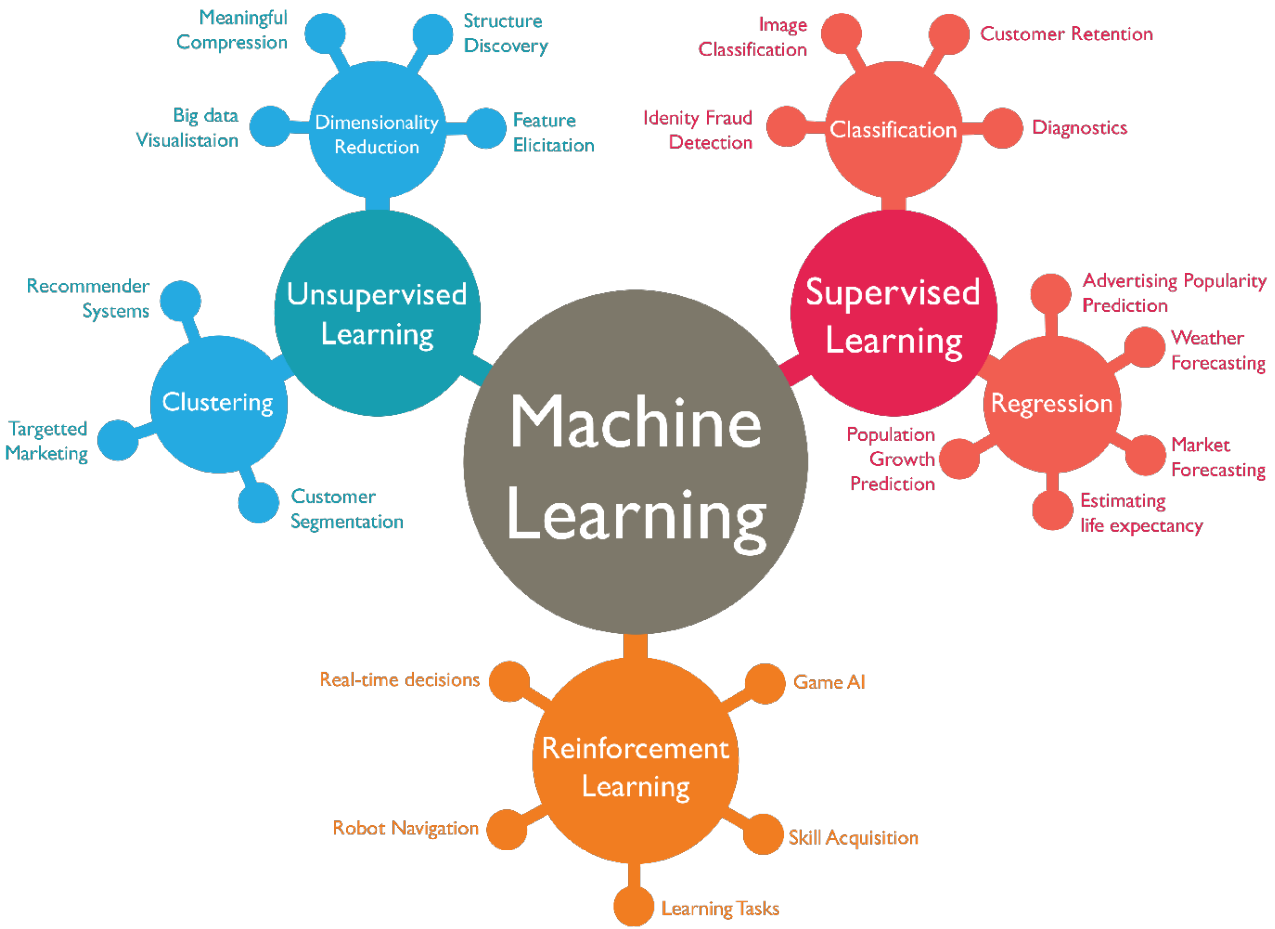
- Linear regression, linear discriminants
- Models: Generative (naive Bayes), Discriminative (logistic regression)
- Support Vector Machines
- Optimization: (Stochastic) Gradient Descent
- Boosting
- Nonlinear methods: Kernels
- Deep Learning
- Dimensionality Reduction: Linear, Nonlinear
- Generative models: autoencoders, GANs
- Learning Theory
- Online Learning, Online Optimization
- Reinforcement learning

What we will not cover

- Semi-supervised learning, cost sensitive learning
- Structured prediction, ranking, preference learning
- Graphical models, nonparametric Bayes, latent variable models
- Transfer and multi-task learning
- Active learning, noisy training
- Kernel learning
- Applications: Vision, Speech, NLP, IR, Bioinformatics, etc.
- Matrix factorization and recommendation systems
- ... and many other topics

What is Machine Learning?

"Machine learning is programming computers to optimize a performance criterion using example data or past experience" - Ethem Alpaydin



https://www.wordstream.com/blog/ws/2017/07/28/machine-learning-applications

Supervised Learning

- Learn mapping from input to output
- Utilizes labelled dataset
  - Input-output (features, labels or target values) pairs
  - Supervisor tells algorithm correct answer
- Classification, regression, anomaly detection, etc.
- Example applications:
  - Movie predictions
  - Stock price prediction
  - Face recognition

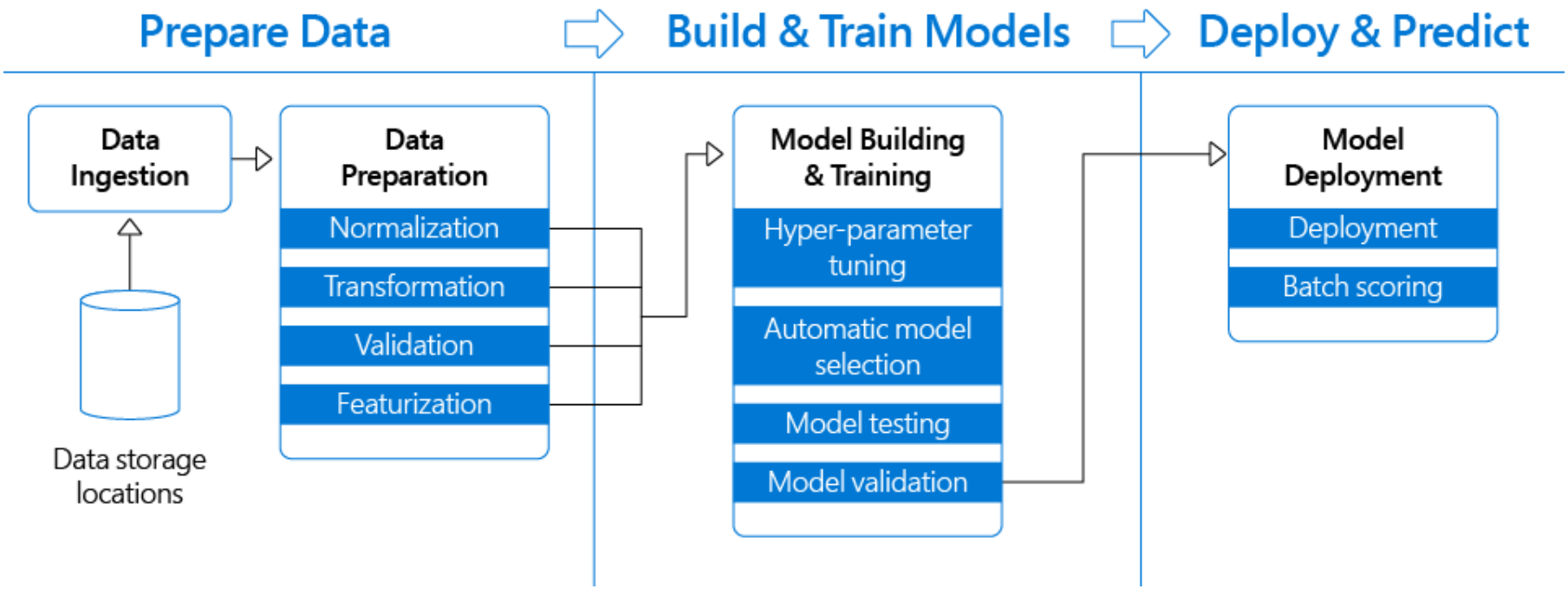
Unsupervised Learning

- Learn "what normally happens"
- No output (labels)
- Clustering: Grouping similar instances
- Example applications:
  - Customer segmentation
  - Image compression: Color quantization
  - Bioinformatics: Learning motifs

Reinforcement Learning

- Learn via interacting with environment (i.e., trial and error)
- Consists of observing environment state, taking action, receiving reward
- No supervisor to tell you correct answer
- Only relative rewards give clue of action quality
- Example applications:
  - Recommender systems / personalization
  - Manufacturing optimization
  - Game playing (e.g., Mario)
  - Robot search and rescue
  - Path planning
  - Drug discovery

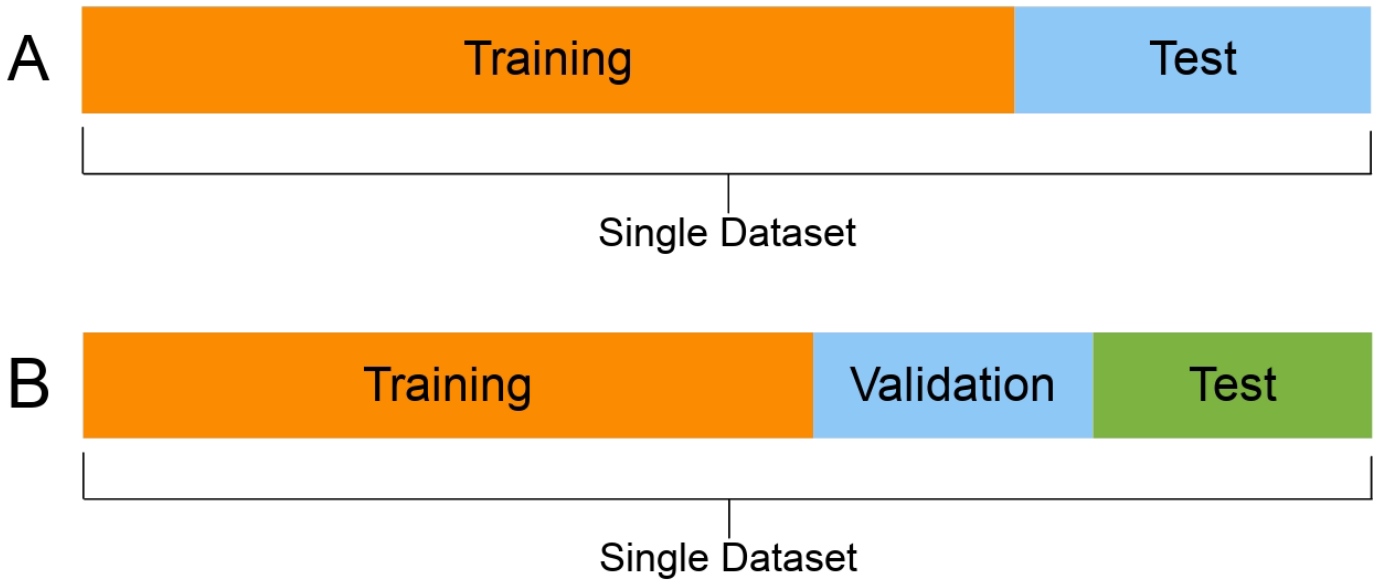
Machine Learning Pipeline



https://medium.com/microsoftazure/how-to-accelerate-devops-with-machine-learning-lifecycle-management-2ca4c86387a0

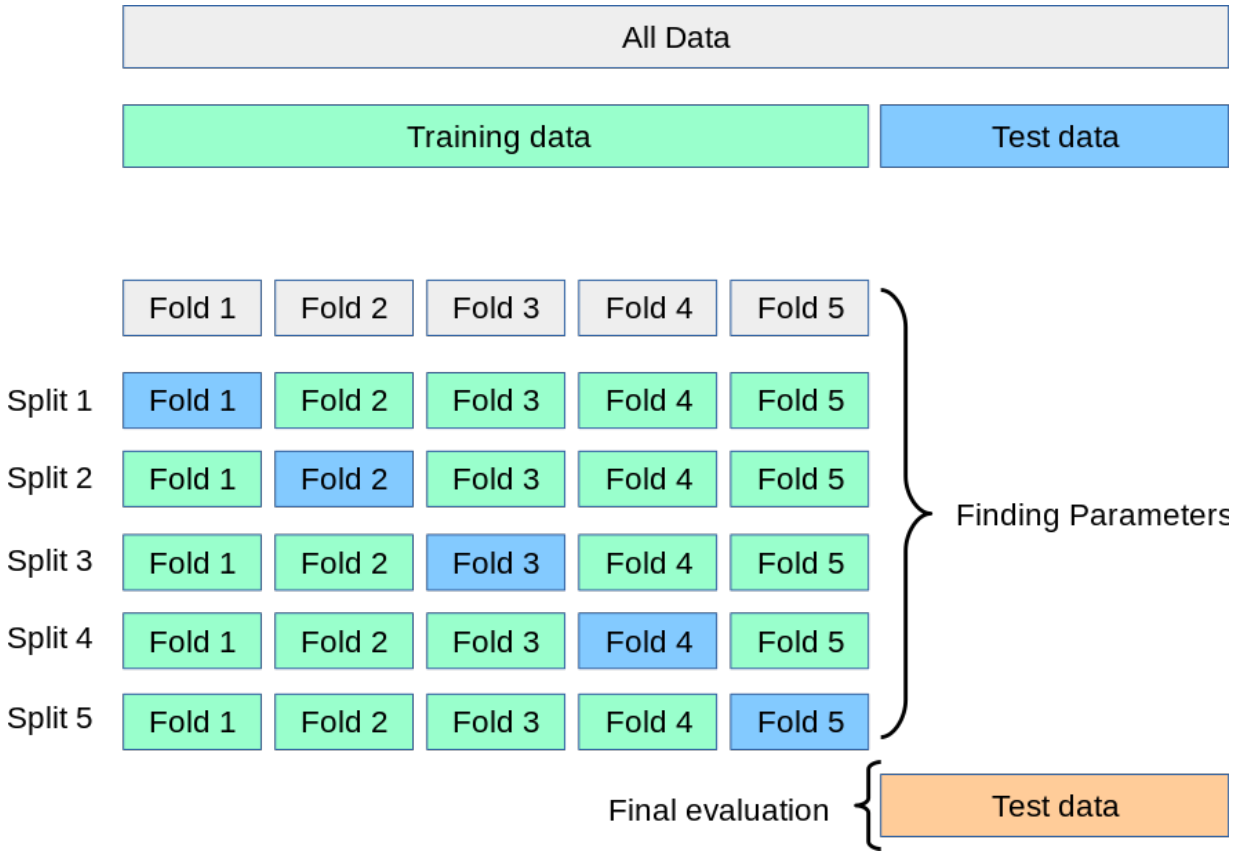
Datasets

- Dataset: collection of data (numbers, images, words, etc.)
  - Training data: data used to train a machine learning algorithm (i.e., learn a model)
  - Validation data: data used to measure performance of a model (for different hyperparameters)
  - Test data: unseen data used to measure performance of final model



https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets

Cross Validation



https://scikit-learn.org/stable/modules/cross_validation.html

Key Terms

- Hyperparameters: algorithm parameters user chooses before training (they control learning)
- Loss/cost function: function to measure performance of algorithm
- Generalization: how well a model predicts on unseen data
- Features: attribute (characteristic) of the data
- Labels/target values: class, output (dependent variable) value
- Representation: how data is encoded (images, numbers, graphs, etc.)
- Error rate: probability predicted class is incorrect
- Supervised: learning from a labeled dataset
- Unsupervised: learning structure of data (no labels/target values)
- Reinforcement: learning via interacting with environment (trial and error)

ML Examples

https://thispersondoesnotexist.com/

https://openai.com/dall-e-2/

Mario Kart with Q-learning:
https://www.youtube.com/watch?v=Tnu4O_xEmVk

RL Locomotion:
https://www.youtube.com/watch?v=hx_bgoTF7bs