Course Logistics and Introduction

CS771: Introduction to Machine Learning

Piyush Rai

Course Logistics

- Course name: Introduction to Machine Learning CS771A)
- Timing and Venue: Mon/Thur 6:00-7:30pm, L-20
- Course website: https://tinyurl.com/cs771-a23 (slides, readings, etc)
- Online discussion/QA: Piazza (https://tinyurl.com/cs771-a23-piazzasignup)
- Instructor's contact email: piyush@cse.iitk.ac.in, office: RM-502 (CSE dept)
  - Prefix email subject with CS771, else might get ignored
  - Use of Piazza is encouraged for course-related matters (also has private messaging)
  - Office hours: Wed 6pm-7pm (or by appointment)
- Unofficial auditors are welcome. However, can't participate in exams/quizzes
  - Can attempt homeworks, quizzes, exams on their own. Won't be graded

Course Team (TAs)

- Aditya Dhaulakhandi (adityad@cse)
- Subhajit Panday (subhajitpanday@cse)
- Malay Pandey (malay@cse)
- Abhishek Jaiswal (abhijais@cse),
- Putrevu Venkata Sai Charan (pvcharan@cse)
- Pramit Bhattacharyya (pramitb@cse)
- Gargi Sarkar (gsarkar@cse)
- Virendra Nishad (viren@cse)
- Priyanka Maity (priyankamaity@cse)
- Ayush Pande (ayushp@cse)
- Debkanta Chakraborty (debkanta@cse)
- Saqib Sarwar (saqib@cse)

TA office locations and office hours announced soon

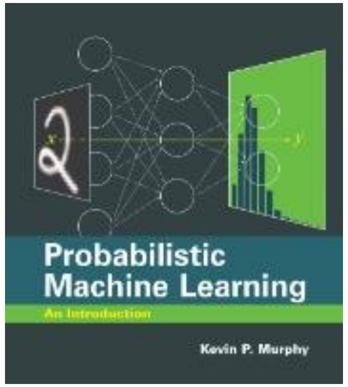
Workload and Grading Policy

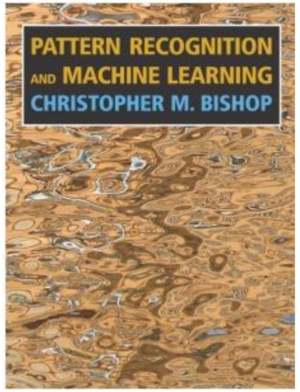
Quizzes will be closed-book.
For exams, one A4 size
cheat-sheet will be allowed

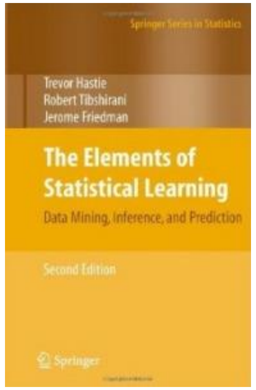
- 4 quizzes: 30%
- 2 homeworks/mini-projects: 20%
  - Writeups must be prepared in PDF using the provided LaTeX template
  - Knowledge of Python programming is assumed (will have a tutorial this weekend)
- Mid-sem exam: 20%
- End-sem exam: 30%
- Quiz dates (tentative): Aug 16, Sept 5, Oct 13, Nov 7
  - Quiz timing and venue: will be announced closed to the quiz date
- HW/mini-project dates (tentative): Aug 17, Oct 2 (roughly 3 work-weeks given)
- Mid-sem and end-sem exam dates: As per DOAA announcements

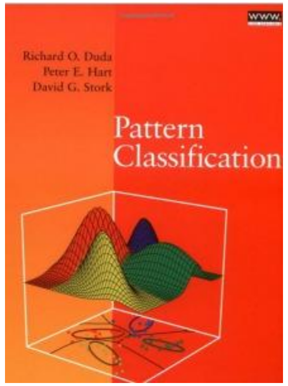
Textbook and References

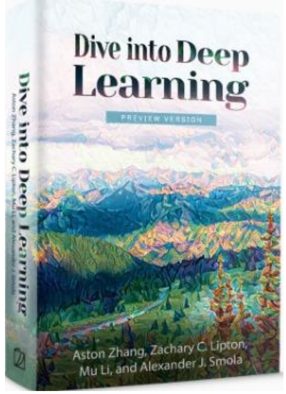
- Many excellent texts but none "required". Some include:

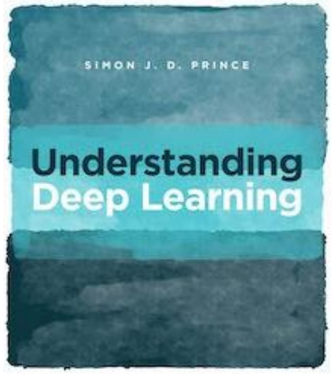












- See the course website for links and other relevant texts and references
- Different books might vary in terms of
  - Set of topics covered
  - Flavor (e.g., classical statistics, deep learning, probabilistic/Bayesian, theory)
  - Terminology and notation (beware of this especially)
- For each topic in the course, we will provide you recommended readings

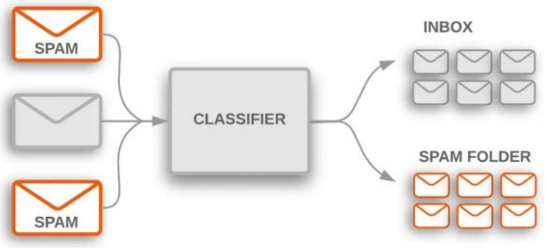
Course Goals

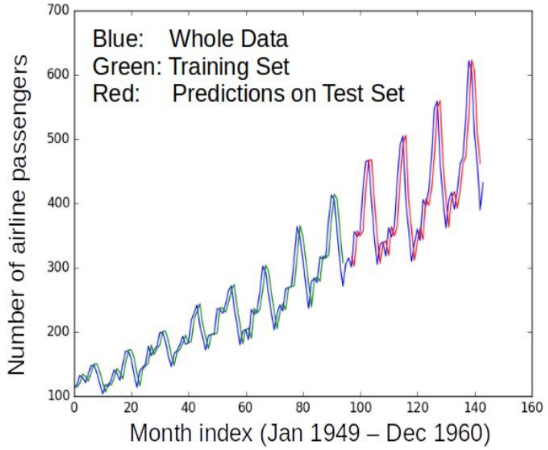
- Introduction to the foundations of machine learning (ML)
- Focus on developing the ability to
  - Understand the underlying principles (and maths ☺) behind ML models and algos
  - Understand how to implement and evaluate them
  - Understand/develop intuition on choosing the right ML model/algo for your problem
- (Hopefully) inspire you to work on and learn more about ML
- Not an intro to popular software frameworks and libraries, such as scikit-learn, PyTorch, Tensorflow, etc
  - However, you are encouraged to explore these as the course progresses

What Is Machine Learning?

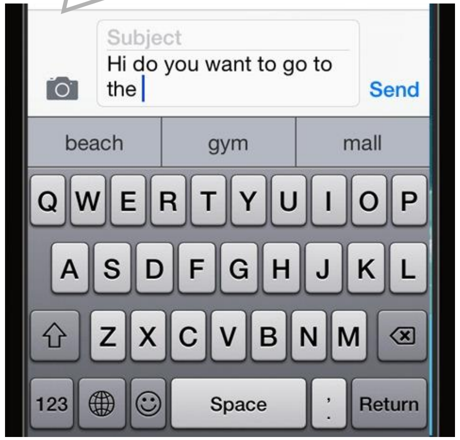
Machine Learning (ML)

- Designing algorithms that ingest data and learn a model of the data
- The learned model can be used to
  - Detect patterns/structures/themes/trends etc. in the data
  - Make predictions about future data and make decisions





Next word prediction (key task in large-language models like ChatGPT)



- Modern ML algorithms are heavily "data-driven"
  - No need to pre-define all the rules by humans (infeasible/impossible anyway)
  - The rules are not "static"; can adapt as the ML algo ingests more and more data

Where Should We Use ML?

Handwritten digit recognition: Not too complex but still reasonably complex that an ML approach is desirable

- When the learning problem is very complex, e.g.,
  - Enumerating all rules is infeasible or too time-consuming
  - Rules might evolve with time
- In such cases, hard-coding the rules in a computer program may not work
  - Difficult to define and code all possible rules
  - Difficult to update the program if rules evolve
- ML replaces the idea of humans writing code by humans supplying data
  - The ML algorithm automatically learns the model (the rules) from the supplied data
  - The model can evolve with more and more data

Artificial Intelligence (AI) vs Machine Learning

- Often the terms AI and ML are used synonymously but don't mean the same

- AI is about endowing machines with intelligence

- ML is a way to achieve AI by learning patterns/predictive models from data

- AI is a much broader term and covers various sub-fields such as
  - Machine Learning
  - Natural Language Processing
  - Computer Vision
  - .. and many others

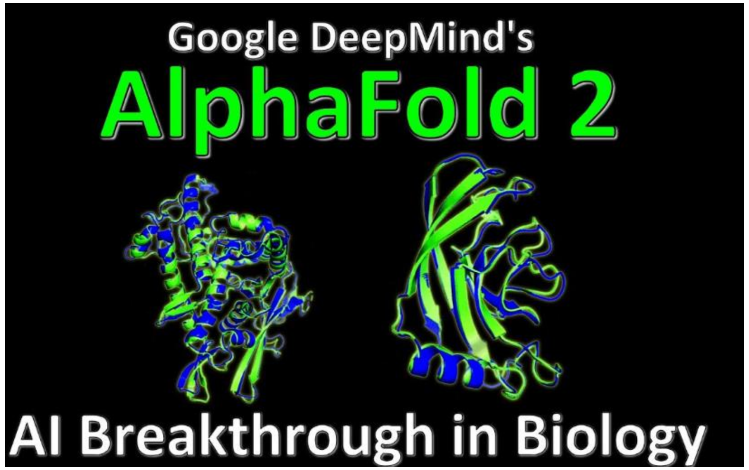
Can think of "deep learning" as a way of doing machine learning

And what about "deep learning"

Kind of a way to make machine learning more "automatic" (will discuss more later)

ML: Some Success Stories

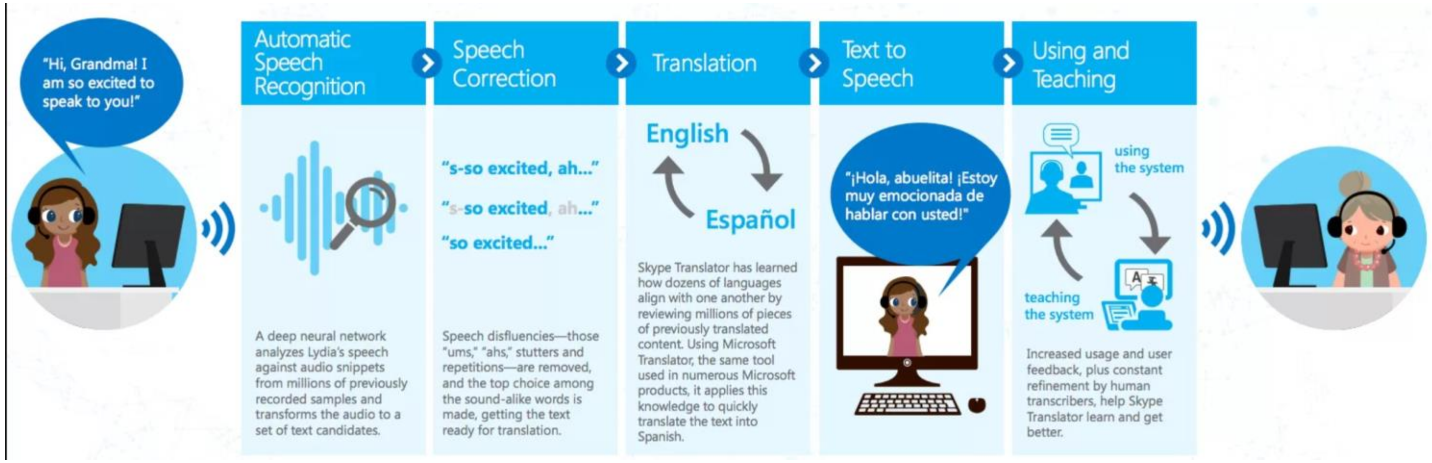
Protein Structure Prediction



Autonomous Driving

AI Generated Digital Art (Dall E 2)

Real-time Speech Translation

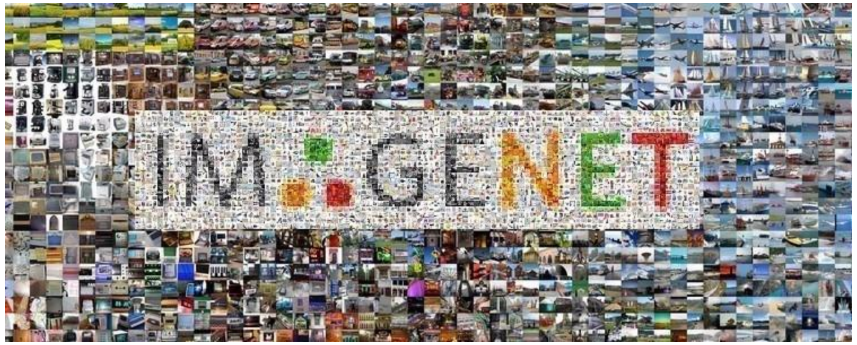


Conversational Systems

Digital art pic credit: David Schnurr

Key Enablers for Modern ML

- Availability of large amounts of data to train ML models

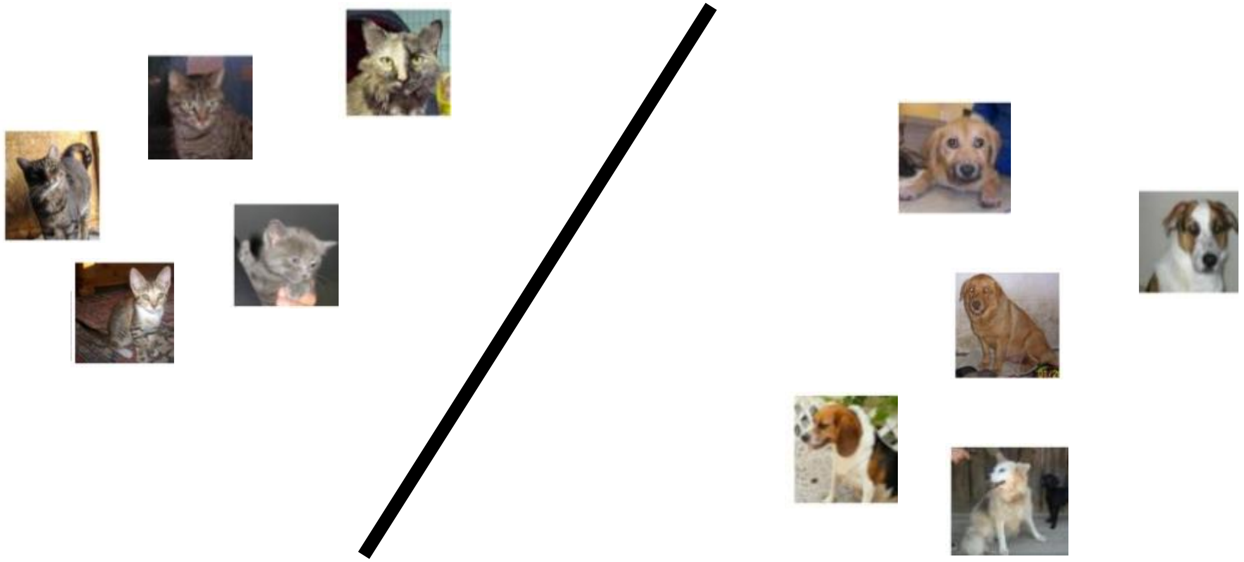


- Increased computing power (e.g., GPUs)

ML: A Simple Illustration

- ML enables intelligent systems to be data-driven rather than rule-driven
- How: By supplying training data and building statistical models of data
- Pictorial illustration of an ML model for binary classification:

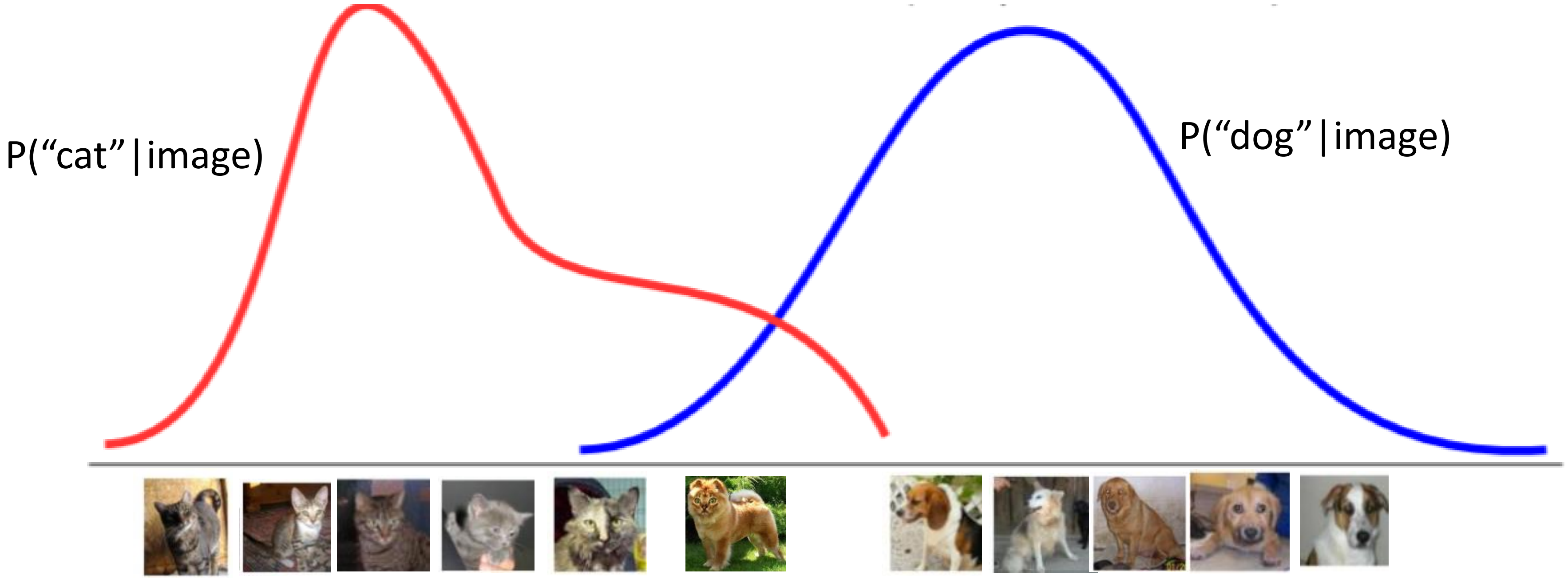
A Linear Classifier (the statistical model)



ML: A Simple Illustration

- ML enables intelligent systems to be data-driven rather than rule-driven
- How: By supplying training data and building statistical models of data
- Pictorial illustration of an ML model for binary classification:

A Probabilistic Classifier (the statistical model)



ML: The Exam Anology

- It's the performance on the D-day which matters

- In an exam, our success is measured based on how well we did on the questions in the test (not on the questions we practiced on)

- Likewise, in ML, success of the learned model is measured based on how well it predicts/fits the future test data (not the training data)

In Machine Learning, generalization performance on the test data matters (we should not "overfit" on training data)

Coming Up Next..

- Types of ML problems
- Typical workflow of ML problems
- Various perspectives of ML problems

A Loose Taxonomy of ML

Learning using labeled data

Learning using unlabeled data

"Labeled" means, during training, for each input, the corresponding output is available (i.e., the machine learner is explicitly told that a cat image is of a cat)

Some examples of supervised learning problems

- Classification
- Regression
- Ranking

Supervised Learning

Machine Learning

Unsupervised Learning

Some examples of unsupervised learning problems

- Clustering
- Dimensionality Reduction
- Unsupervised Probability Density Estimation

Reinforcement Learning

RL doesn't use "labeled" or "unlabeled" data in the traditional sense! In RL, an agent learns via its interactions with an environment

Many other specialized flavors of ML also exist, some of which include

- Semi-supervised Learning
- Self-supervised Learning (very popular these days)
- Active Learning
- Transfer Learning
- Multitask Learning
- Zero-Shot/Few-Shot Learning
- Continual learning

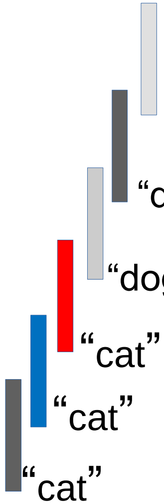
A Typical Supervised Learning Workflow

Note: This example is for the problem of binary classification, a supervised learning problem

Labeled Training Data

"dog"
"dog"
"dog"
"cat"
"cat"
"cat"

"Feature" Extraction



"dog"
"dog"
"dog"
"cat"
"cat"
"cat"

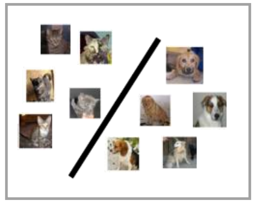
ML Algorithm (outputs a "model")

Is feature extraction done "manually" as a pre-processing step before the ML algo starts working? Can't we "automate" this part? Can't we "learn" good features directly from raw inputs?

Feature extraction converts raw inputs to a numeric representation that the ML algo can understand and work with. More on feature extraction later.

Test Image

"Feature" Extraction



Cat vs Dog Prediction model

Predicted Label (cat/dog)

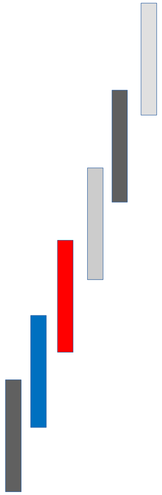
Indeed. Deep Learning algos do precisely that! (feature + model learning). More on Deep Learning later

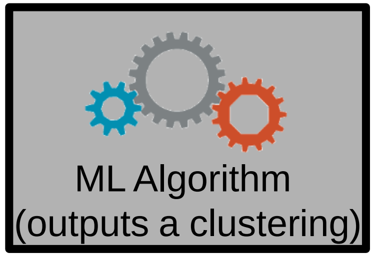
A Typical Unsupervised Learning Workflow

Note: This example is for the problem of data clustering, an unsupervised learning problem

Unlabeled Data

"Feature" Extraction





Yes. In this example, given a new "test" cat/dog image, we can assign it to the cluster with closer centroid

Does unsupervised learning also have a test phase? That is, can we also predict the cluster of a new test input?

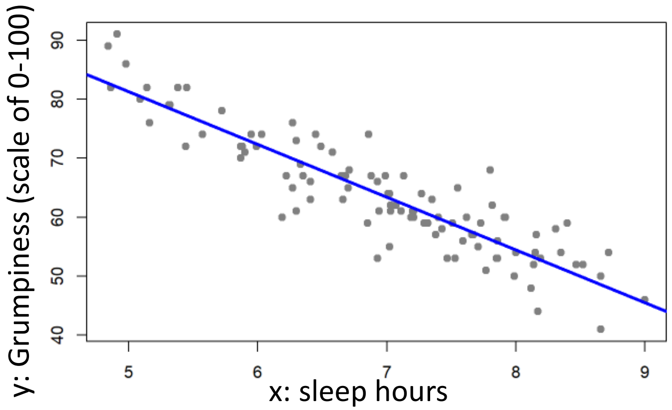
https://www.pinclipart.com/, http://www.pngtree.com

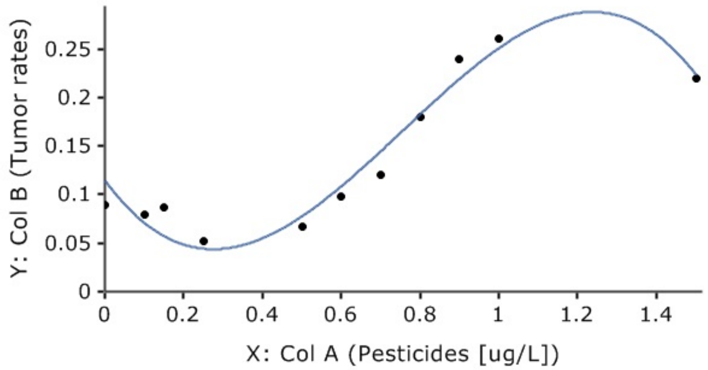
ML from Geometric Perspective

Recall that feature extraction converts inputs into a numeric representation

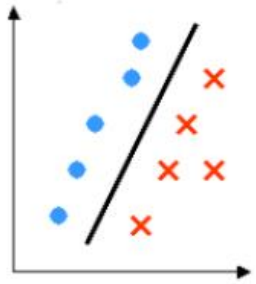
- Basic fact: Inputs in ML problems can often be represented as points or vectors in some vector space
- Doing ML on such data can thus be seen from a geometric view

Regression: A supervised learning problem. Goal is to model the relationship between input (x) and real-valued output (y). This is akin to a line or curve fitting problem

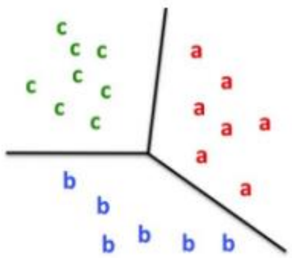




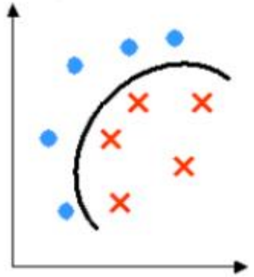
Classification: A supervised learning problem. Goal is to learn a to predict which of the two or more classes an input belongs to. Akin to learning linear/nonlinear separator for the inputs



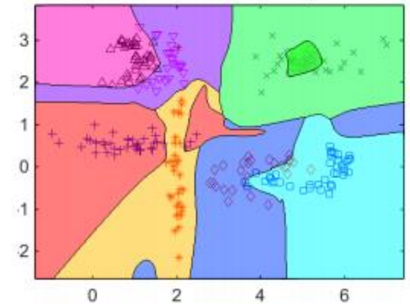
Two-Class (binary) Linear Classification



Multi-Class Linear Classification



Two-Class (binary) Nonlinear Classification

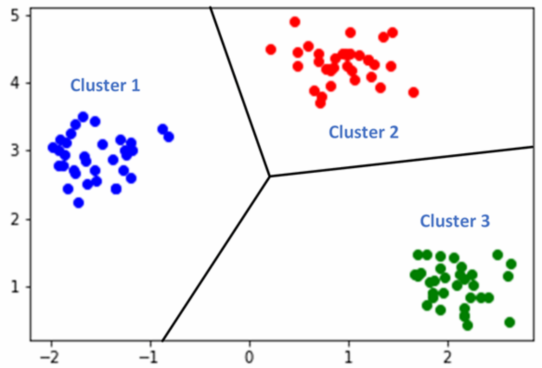


Multi-Class Nonlinear Classification

Pic from: https://learningstatisticswithr.com/book/regression.html, https://maxstat.de/

ML from Geometric Perspective

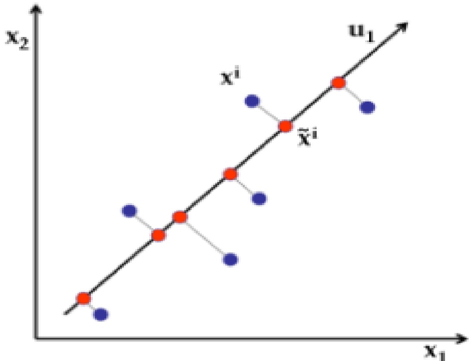
Clustering: An unsupervised learning problem. Goal is to group inputs in a few clusters based on their similarities with each other



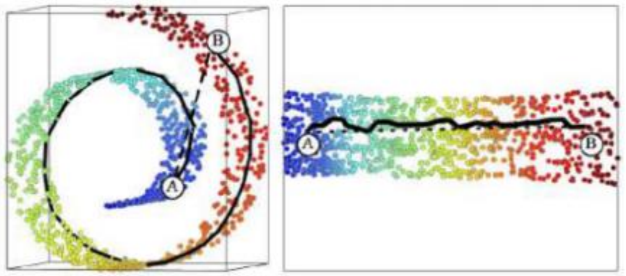
Clustering looks like classification to me. Is there any difference?

Yes. In clustering, we don't know the labels. Goal is to separate them without any labeled "supervision"

Dimensionality Reduction: An unsupervised learning problem. Goal is to compress the size of each input without losing much information present in the data



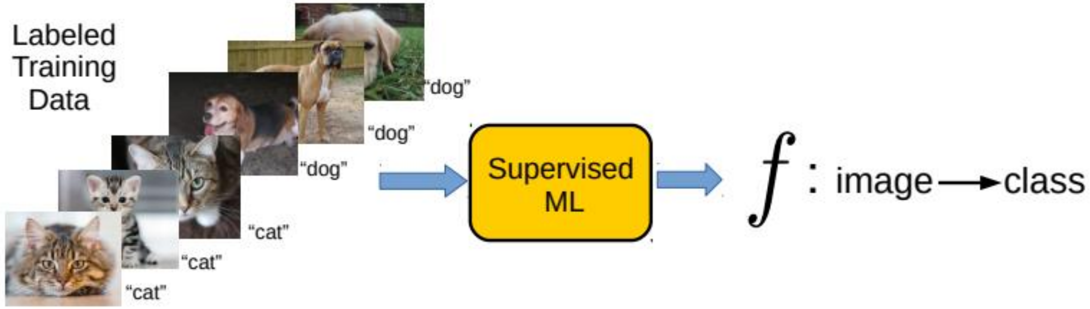
Two-dim to one-dim linear projection



Three-dim to two-dim nonlinear projection (a.k.a. manifold learning)

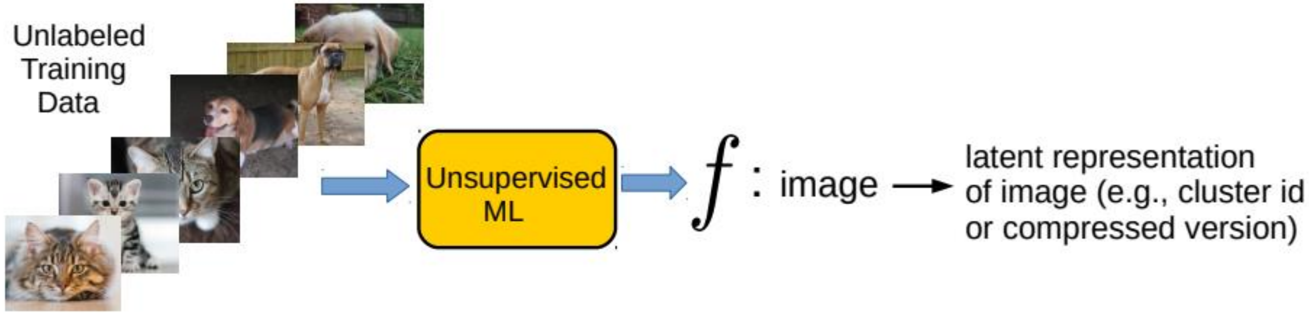
ML from Function Approximation Perspective

- Supervised Learning ("predict output given input") can be usually thought of as learning a function f that maps each input to the corresponding output



- Unsupervised Learning ("model/compress inputs") can also be usually thought of as learning a function f that maps each input to a compact representation

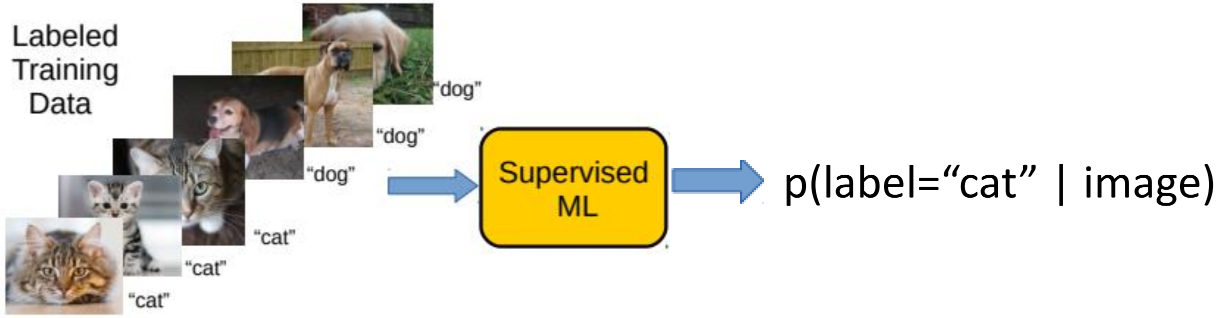
Harder since we don't know the labels in this case



- Reinforcement Learning can also be seen as doing function approximation

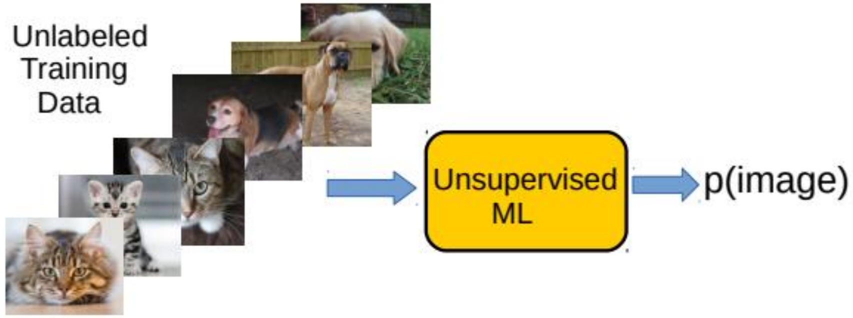
ML from Probability Estimation Perspective

- Supervised Learning ("predict output given input") can be thought of as estimating the conditional probability of each possible output given an input



- Unsupervised Learning ("model/compress inputs") can be thought of as estimating the probability density of the inputs

Harder since we don't know the labels in this case



Don't worry if this doesn't make much sense as of now ☺ But the basic idea is to learn the underlying data distribution using the unlabeled inputs; many ways to do this as we will see later

- Reinforcement Learning can also be seen as estimating probability densities

Next Class

- Data and features

- Some common machine learning paradigms

- Simple supervised learners