PRINCETON UNIVERSITY

DataX
Accelerating Scientific Discovery at Princeton

Introduction to machine learning

Brian Arnold
Christina Peters
Amy Winecoff
Gage DeZoort

CENTER FOR
STATISTICS AND
MACHINE LEARNING

Welcome!

About me

- **background**: evolutionary/statistical genetics
- work with faculty in CS and EEB dept's
- **approach**: data-driven problem solving, *sometimes* with ML

About the course

- overview of conceptual foundation, model building
- learning with real datasets (w/ python)
- light on math/linear algebra

Beyond the course

- ***many*** ML models exist, we will cover only a few
- fundamentals here will be useful for further study

Course outline

Day 1: Introduction

- what is machine learning (ML) and why use it?
- ML conceptual basics
- tour of full modeling process, data processing -> prediction

Day 2: Simple ML methods

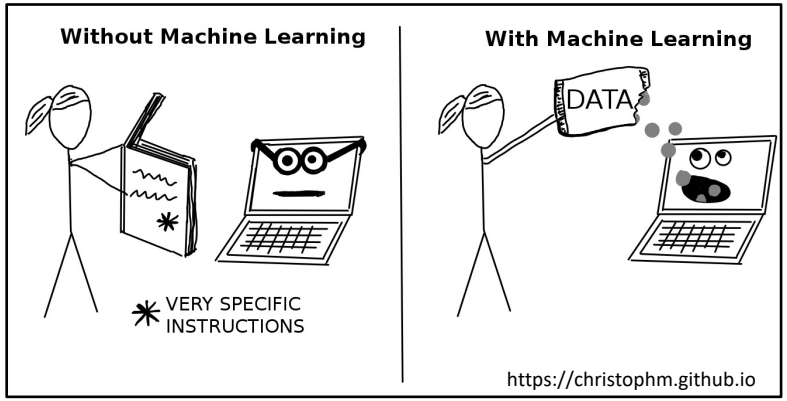
Day 3: Evaluating and improving models

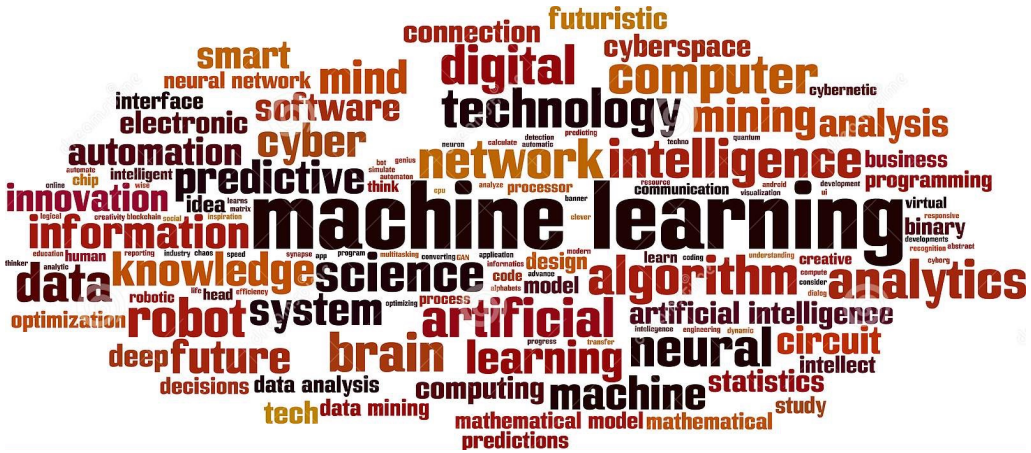
Day 4: Neural networks

Day 5: Hackathon

What is machine learning?

1. building and understanding methods that '**learn**' by using **data** to improve performance on some set of tasks
2. using and developing computer systems that can **learn** and adapt without following explicit instructions, by using algorithms and statistical models to analyze and draw inferences from patterns in **data**.





Also known as:

- artificial intelligence
- pattern recognition
- data mining
- predictive analytics

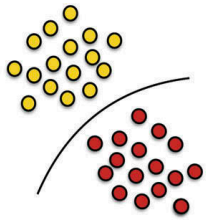
Goal is often to use data to create an algorithm/model that

- makes accurate predictions
- is interpretable, revealing (previously unknown) patterns in data

ML tasks

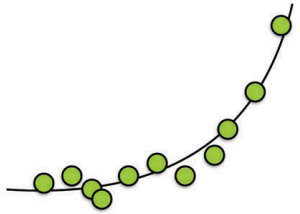
1. building and understanding methods that '**learn**' by using **data** to improve performance on some set of tasks

classification



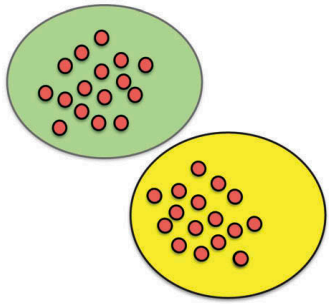
predict input data labels

regression

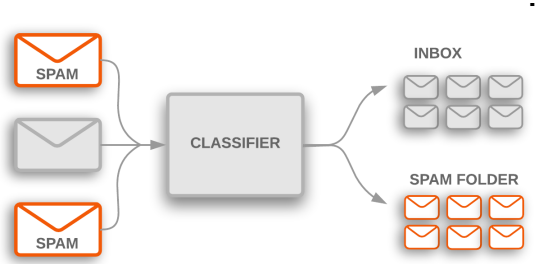


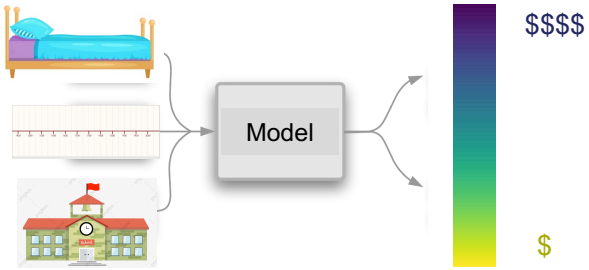
predict input data values

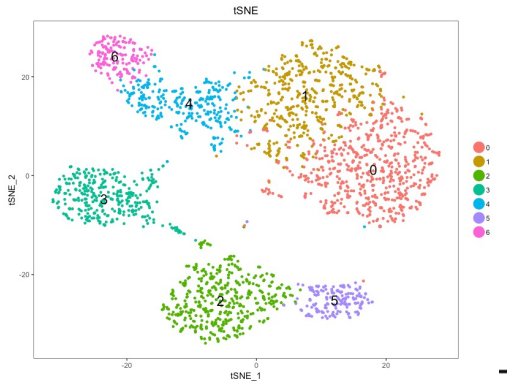
clustering



group input data



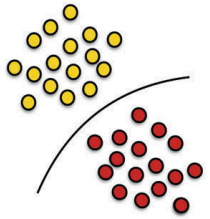




task images from from Carrasquilla 2020

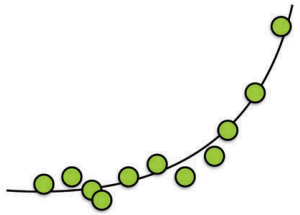
ML tasks

classification



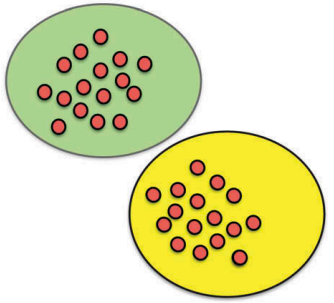
predict input data labels

regression



predict input data values

clustering



group input data

supervised

sample i

$$y_i$$

$$x_{i,1}, x_{i,2} \ldots x_{i,n}$$

response (label/value to model)

features

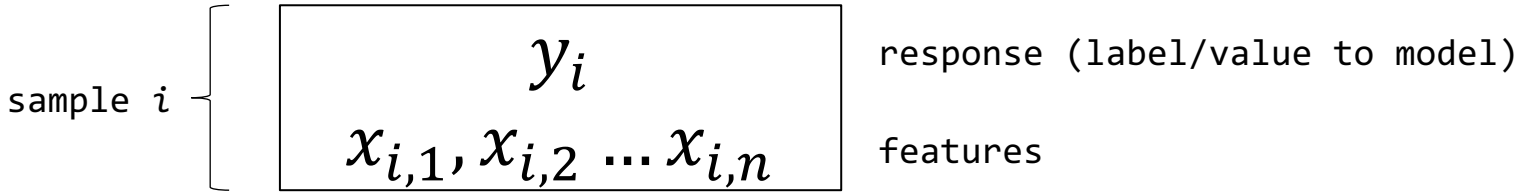
ML learns the relationship between the features and response

unsupervised

$$x_{i,1}, x_{i,2} \ldots x_{i,n}$$

ML learns patterns/groupings

Terminology



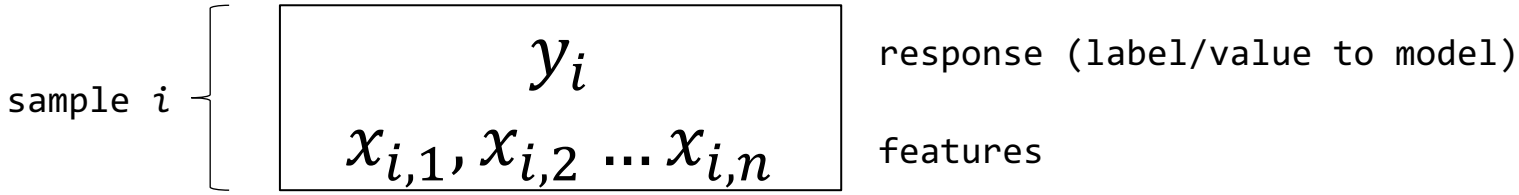
sample i
- sample
- data point
- observation

y_i
- response
- target
- class (if categorical)
- outcome
- dependent variable

$x_{i,1}, x_{i,2} \dots x_{i,n}$
- features
- predictors
- descriptors
- attributes
- covariates
- independent variables

many terms in English, but the **math** is always the same!

Discrete/categorical or continuous values!



examples

response y_i
- a sample's disease status (discrete)
- a sample's height/length (continuous)

features $(x_{i,1}, x_{i,2} \ldots x_{i,n})$
- the presence of a mutation in genome (discrete)
- cigarettes smoked per week (continuous)

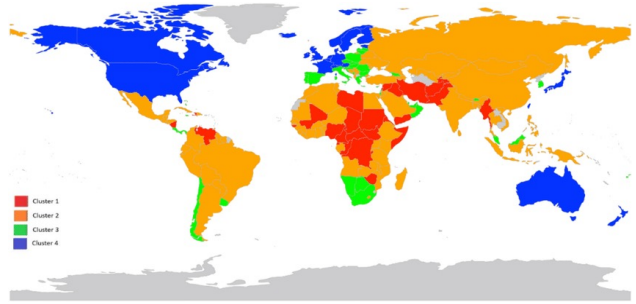
Why use machine learning?

we want to

- know a future event
- make decision based on information
- look for useful patterns in data

examples

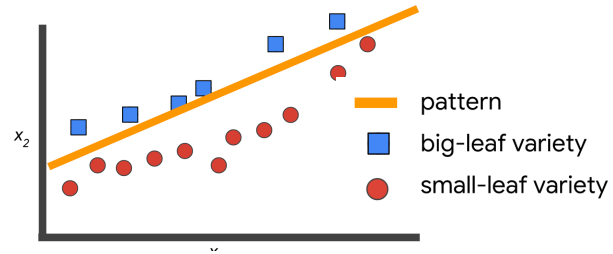
- supervised
  - how many copies will this book sell?
  - will this customer move their business to a different company?
  - how much will my house sell for in the current market?
  - does a patient have a specific disease?
  - based on past choices, which movies will interest this viewer?
  - should I sell this stock?
  - which people should we match in our online dating service?
  - will this patient respond to this therapy?
- unsupervised
  - how do customers differ from one another?
  - how are countries different in terms of socio-economic/health?
  - how many cell types are in my sample?

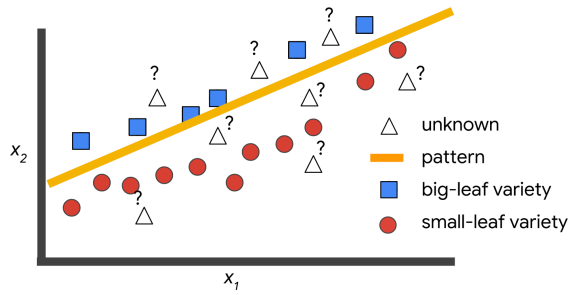


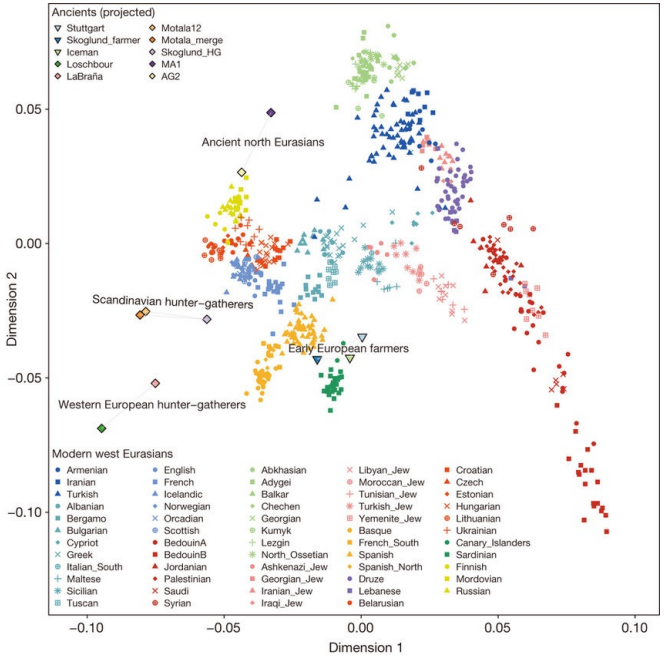
Why use machine learning?

supervised
- prediction
  - predict response value for new samples
- inference
  - understand *how* and *why* a model works

unsupervised
- learn underlying structure of data







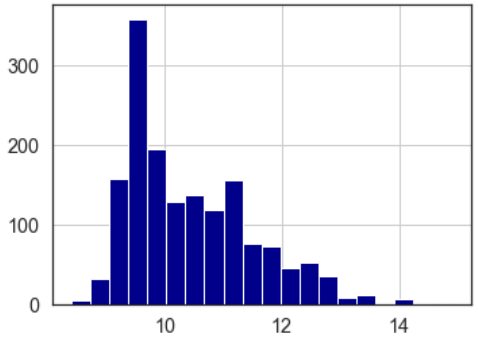
CENTER FOR STATISTICS AND MACHINE LEARNING

Overview of the full ML process

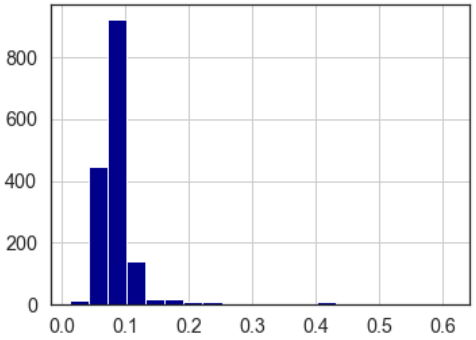
machine learning basics

1. feature scaling/normalization

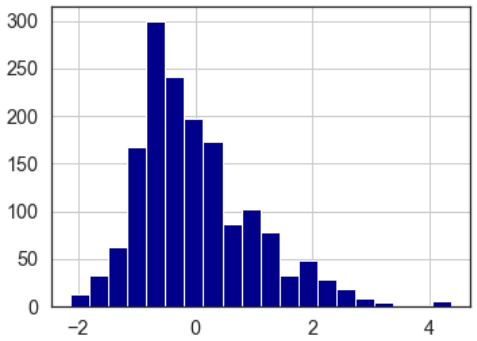
feature 1

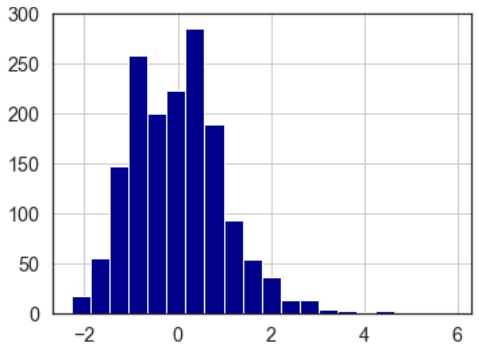


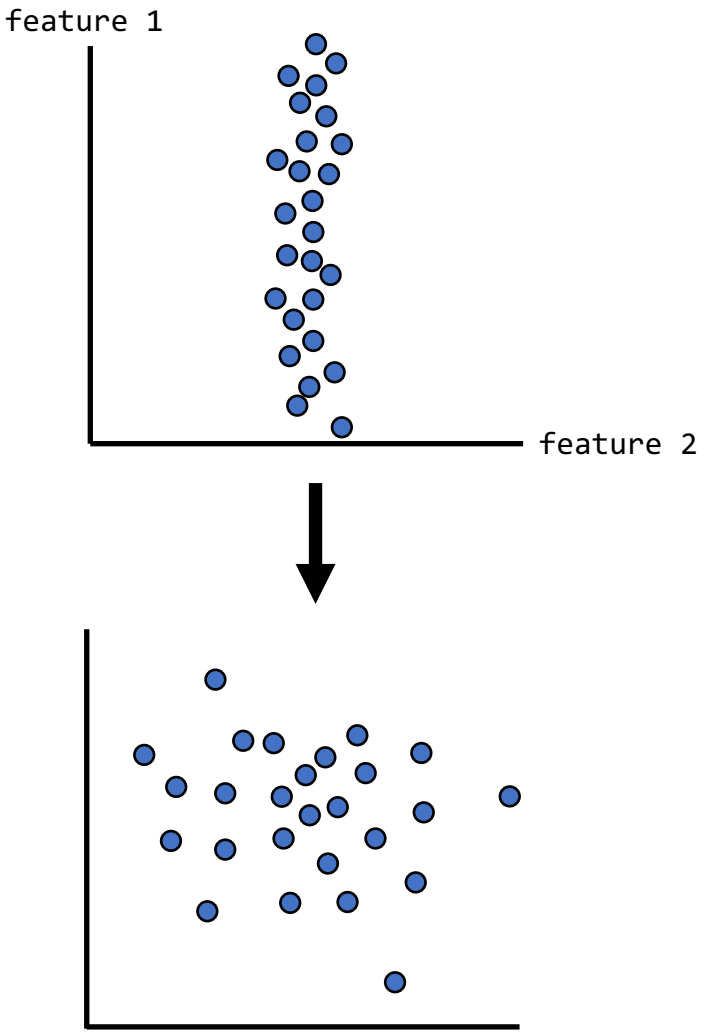
feature 2



normalization







min-max method

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

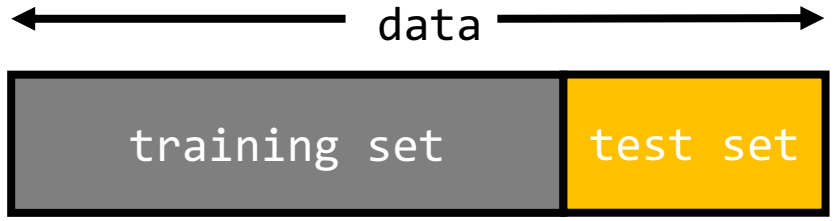
Z-score method

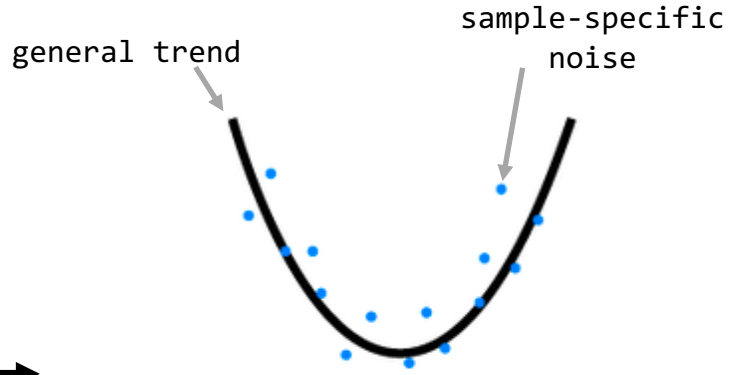
$$x' = \frac{x - \bar{x}}{\sigma}$$

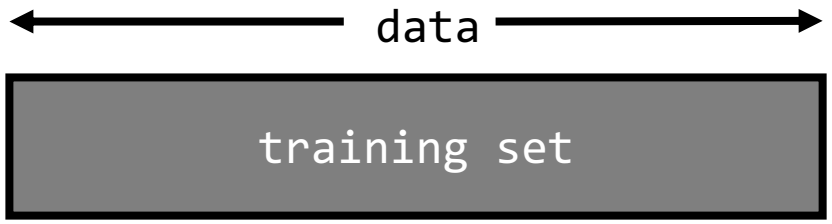
ignoring this can break your model!

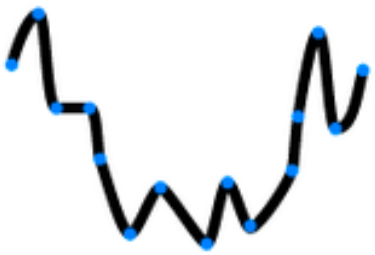
machine learning basics

1. feature scaling/normalization
2. **split data into training set and test set**

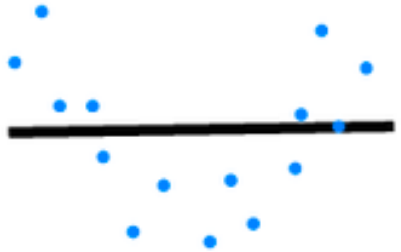








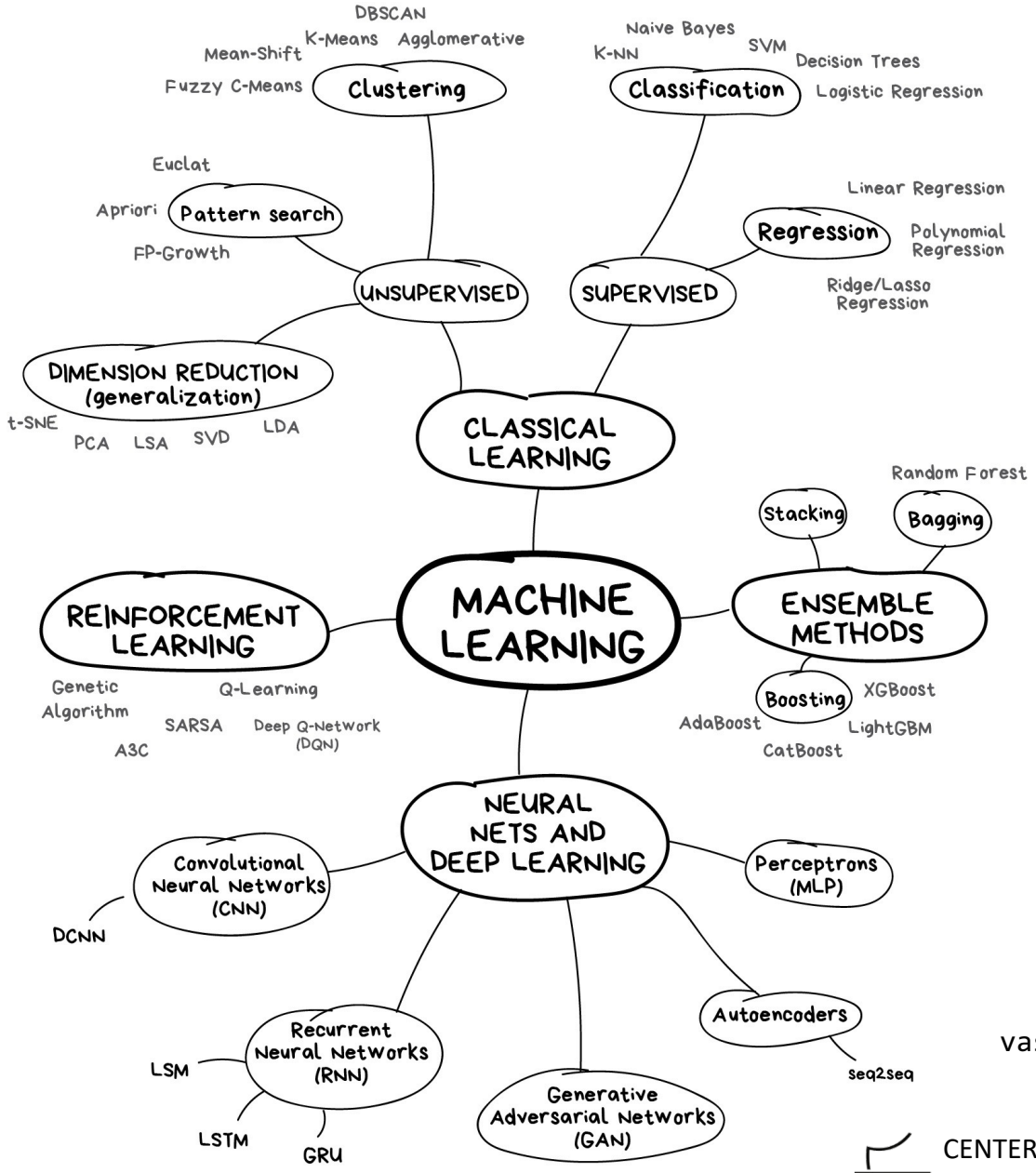
<u>overfitting</u>
(model captures noise)



<u>underfitting</u>
model missed general trend

machine learning basics

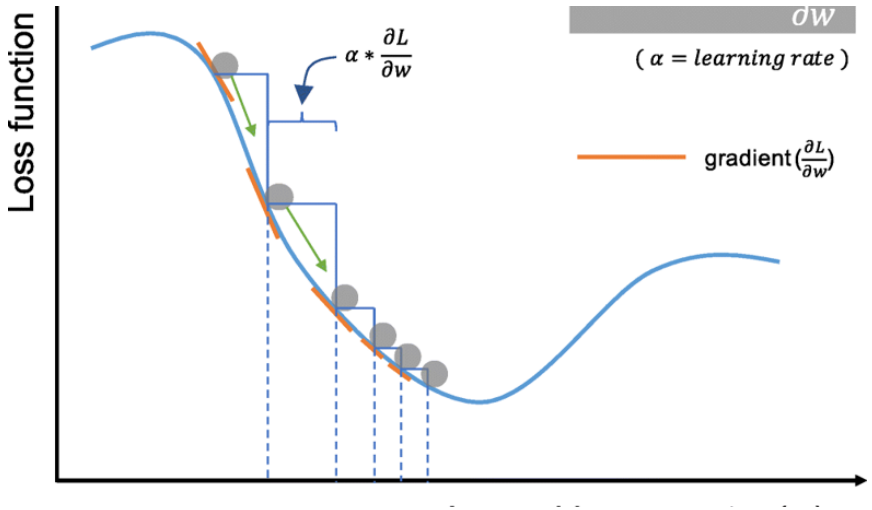
1. feature scaling/normalization
2. split data into training set and test set
3. **choose a ML model**



vas3k.com

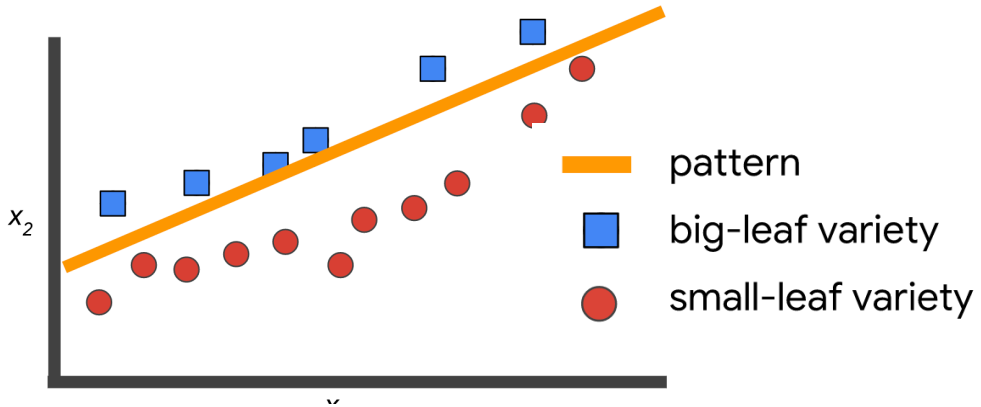
machine learning basics

1. feature scaling/normalization
2. split data into training set and test set
3. choose a ML model
4. **train model, optimizing parameters**



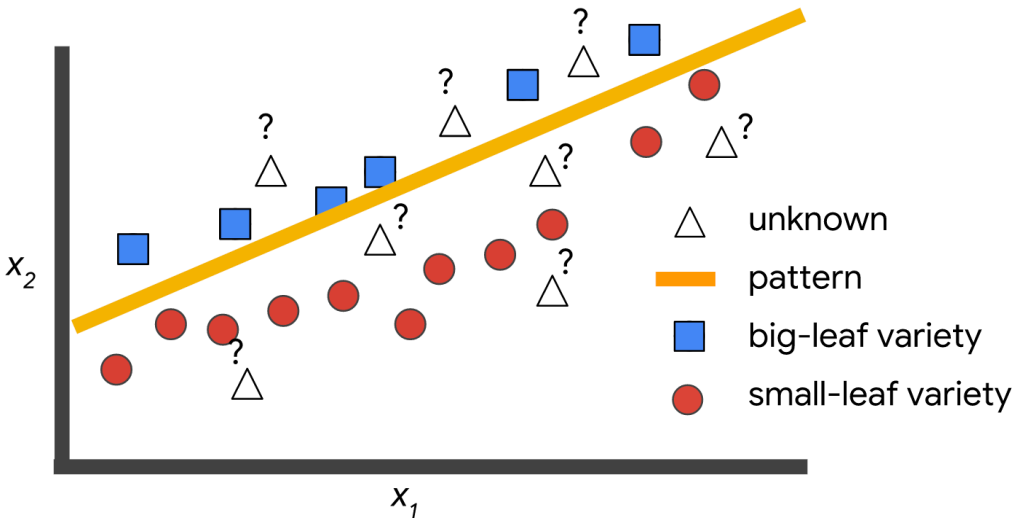
Loss function: L

e.g. find slope of line that **best** separates training labels

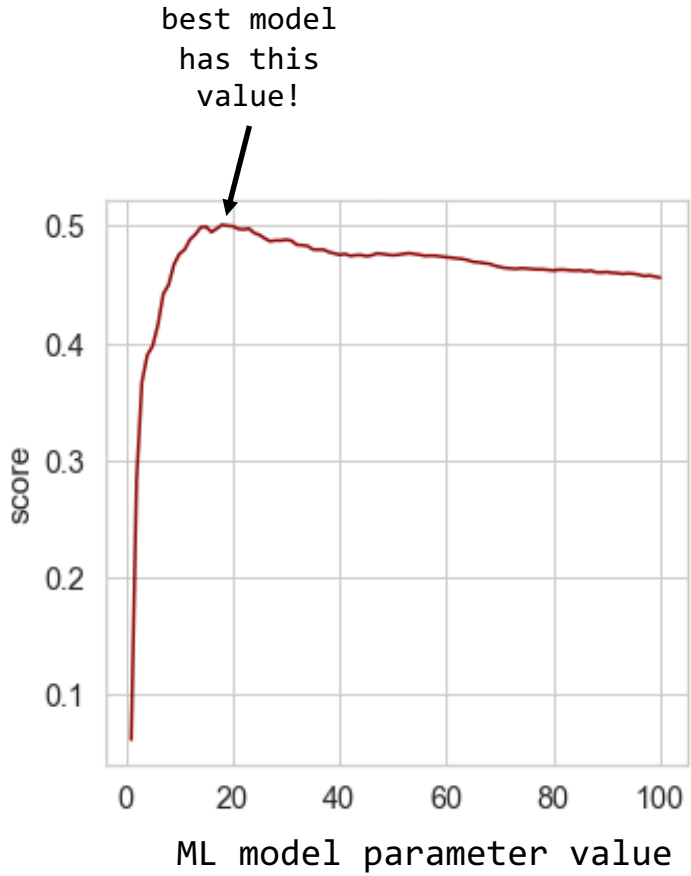


machine learning basics

1. feature scaling/normalization
2. split data into training set and test set
3. choose a ML model
4. train model, optimizing parameters
5. **use the test set to assess performance**



each ML model has it's own unique parameters to tune

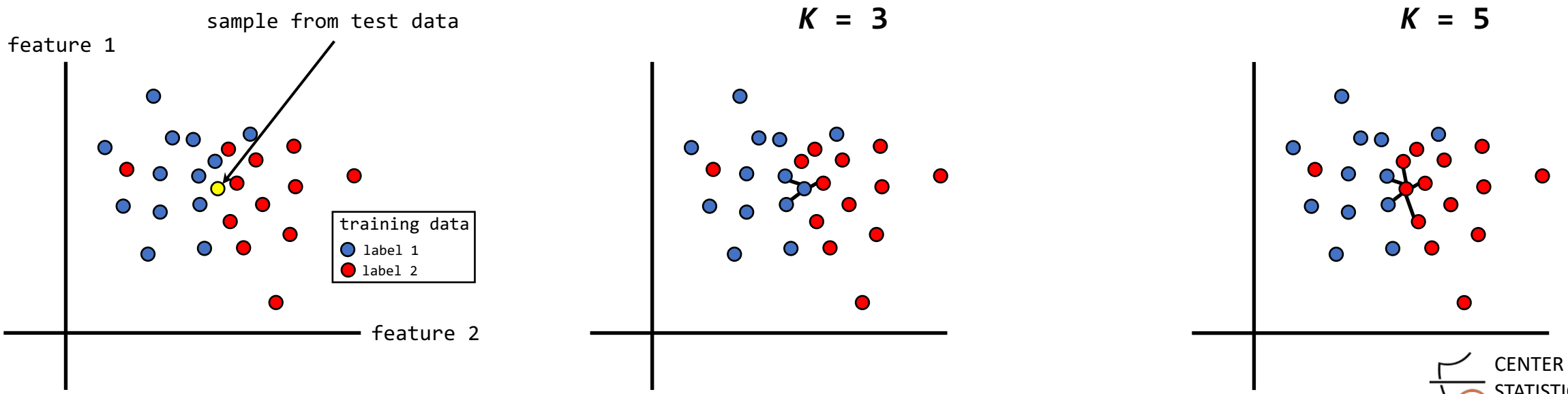


*** select parameter values with greatest predictive accuracy ***

Intro to *K*-nearest neighbors (KNN)

- simple but powerful
- can be used for classification *or* regression!
- algorithm
  1. for a given test sample (yellow dot), find the *K* nearest training samples in feature space
  2a. for **classification**, assign label by majority vote
  2b. for **regression**, assign value by mean of neighbors

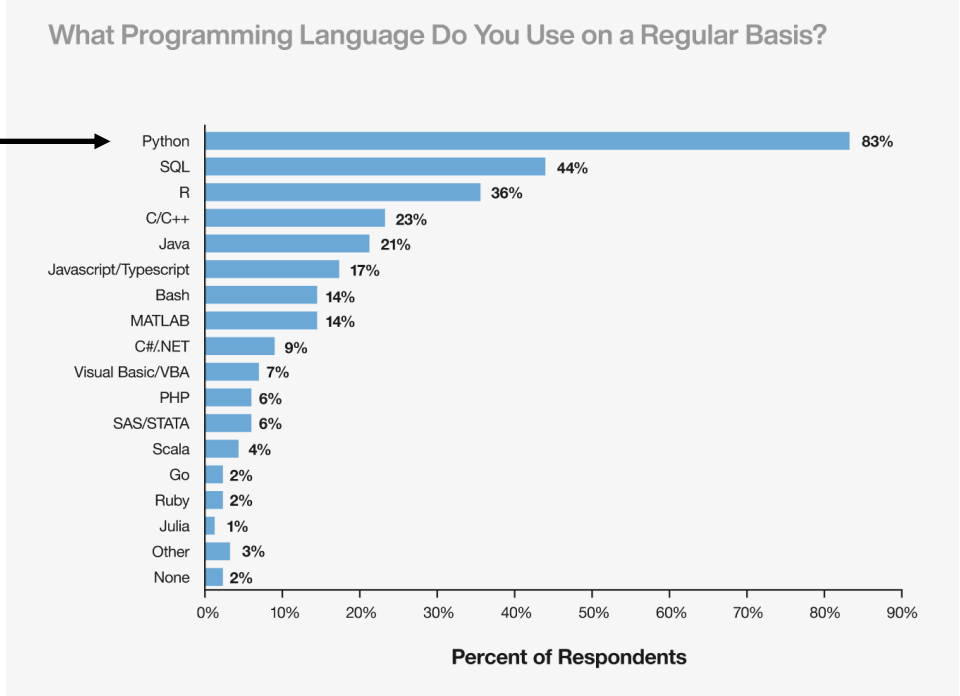
K is a tunable parameter!
- choose value that gives better predictions on test data

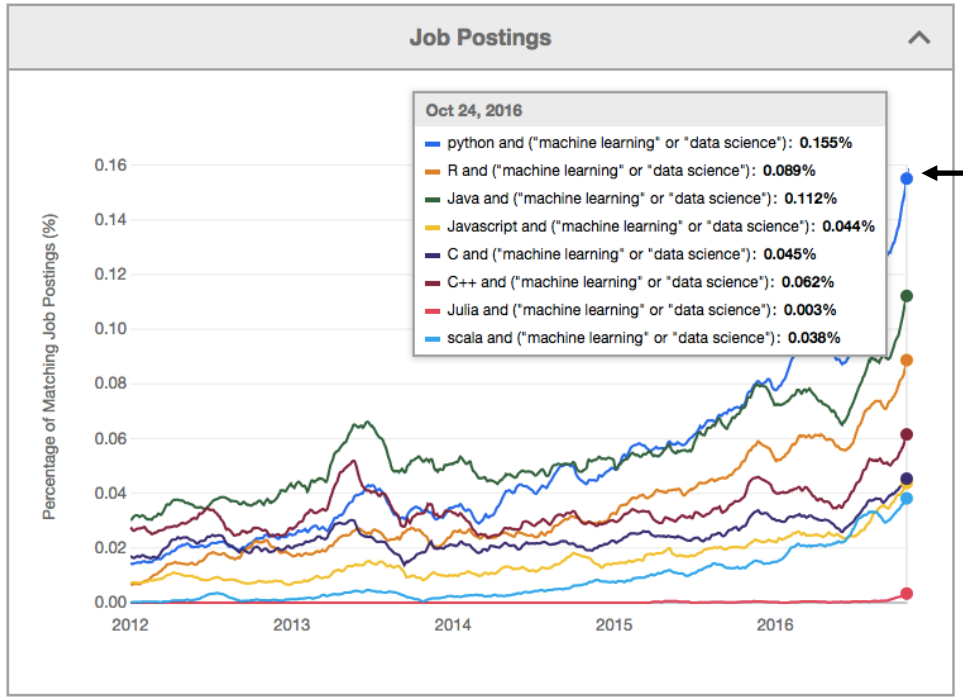


Machine learning in python™

- several options for building ML models
- Python most popular and most in demand (job postings)
- R also popular in statistics and biology communities

python →





← python

CENTER FOR STATISTICS AND MACHINE LEARNING

Fantastic Python libraries

- data analysis
  - Pandas: great for analyzing and manipulating data tables
  - Seaborn: simple functions -> detailed visualization, integrated with Pandas
  - Matplotlib: visualization

- machine learning
  - numpy: fast, powerful data structures for matrices
  - scikit-learn: simple, efficient, accessible tools for ML
  - Keras: neural networks
  - TensorFlow
  - PyTorch
  - …

today we will use **numpy** and **scikit-learn**!

ML coding tour in Python!

Questions?