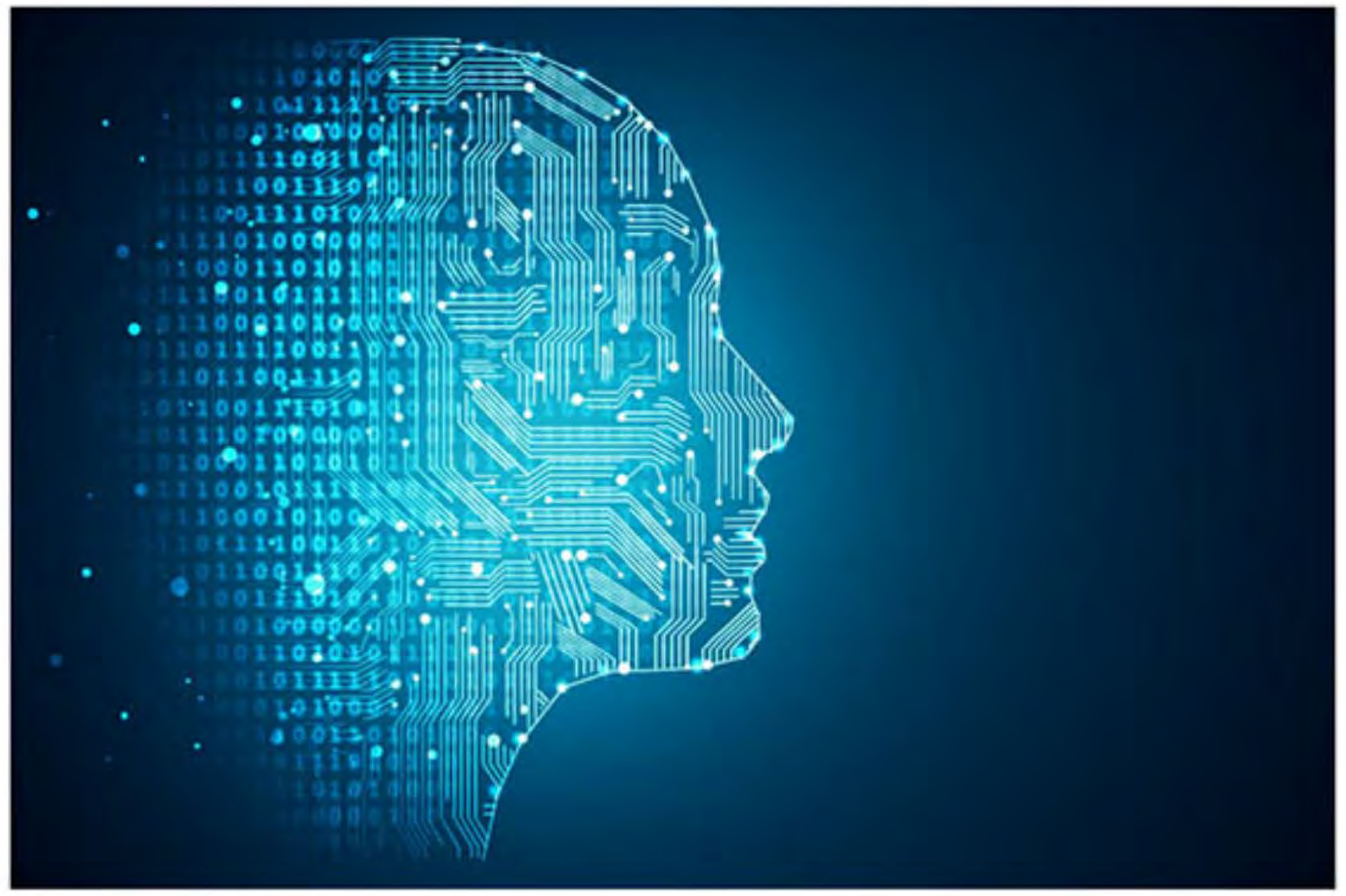
PRINCETON UNIVERSITY

CENTER FOR STATISTICS AND MACHINE LEARNING

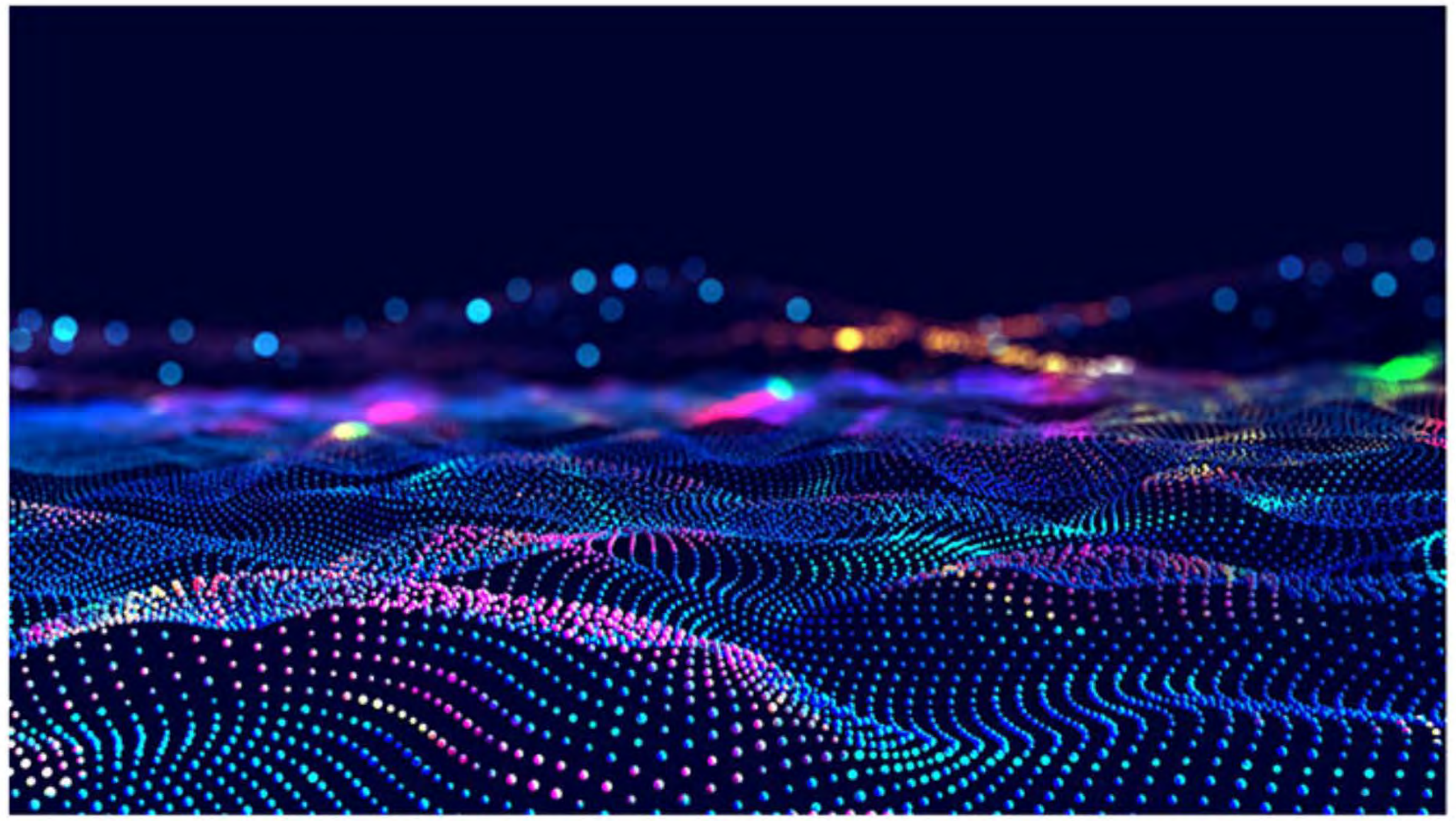
NATURAL & ARTIFICIAL MINDS

A Hands-On Introduction to Machine Learning



Julian Gold
Jonathan Hanke

January 13–16, 2026



With materials from:

Brian Arnold, Gage DeZoort, Julian Gold, Jonathan Halverson, Jonathan Hanke, Christina Peters
Jake Snell, Savannah Thias, Amy Winecoff

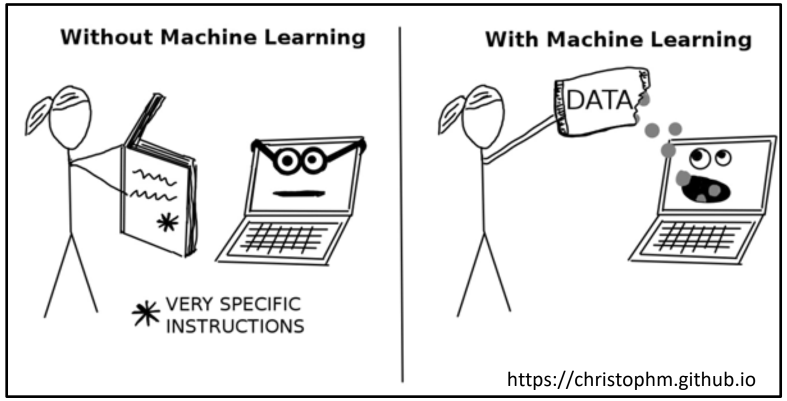
AI LAB
Princeton Laboratory
for **Artificial Intelligence**

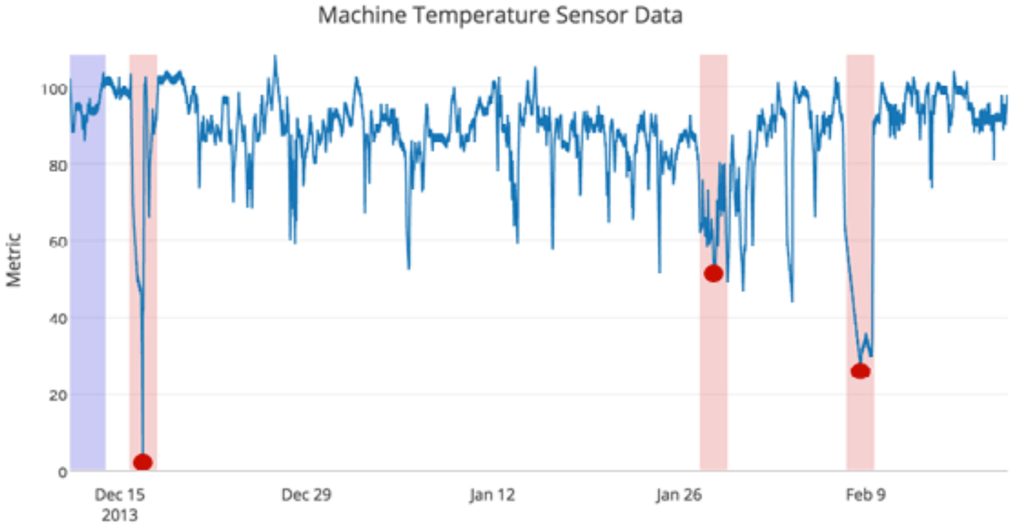
Mini-Course Outline

Date	Topic	Instructor
Tue. 1/13	Machine Learning Overview and Simple Models	Julian Gold
Wed. 1/14	Model Evaluation and Improving Performance	Julian Gold
Thu. 1/15	Introduction to Neural Networks	Jonathan Hanke
Fri. 1/16	Survey of Neural Network Architectures	Jonathan Hanke

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:

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

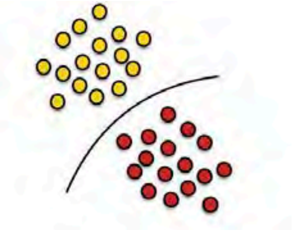
Goal is often to use data to create an algorithm/model that

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

ML tasks

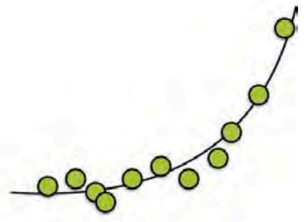
1. building and understanding methods that ‘**learn**’ by using **data** to improve performance on some set of tasks

classification



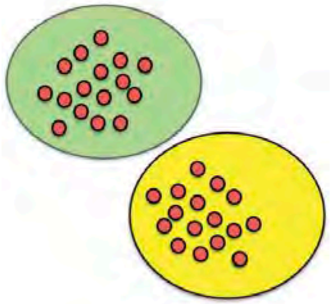
predict input data labels

regression

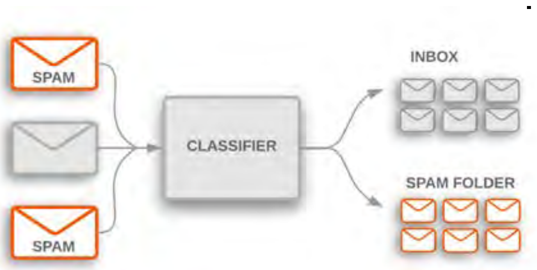


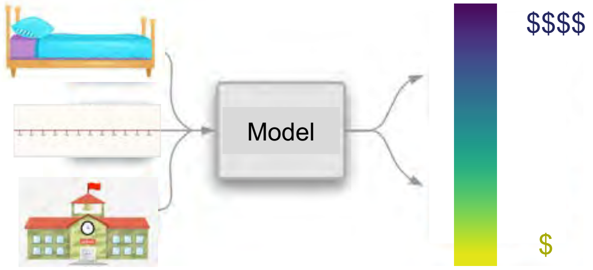
predict input data values

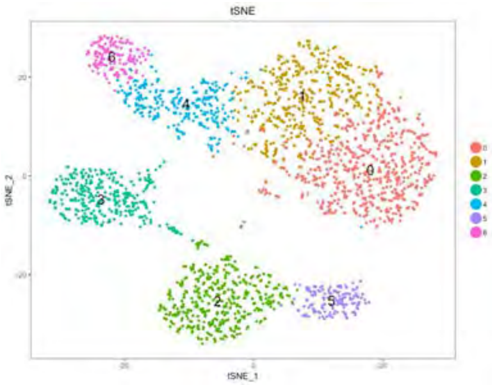
clustering



group input data



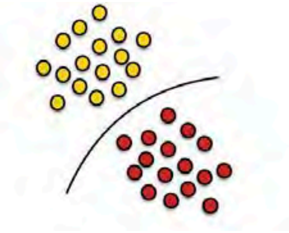




task images from from Carrasquilla 2020

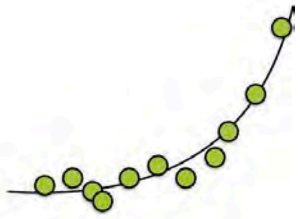
ML tasks

classification



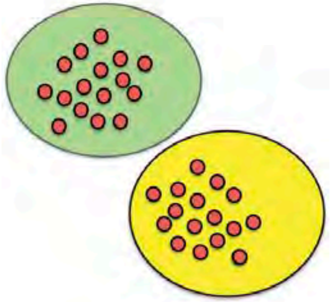
predict input data labels

regression



predict input data values

clustering



group input data

supervised

response (label/value) y_i

features $x_{i,1}, x_{i,2} \dots x_{i,n}$

sample i

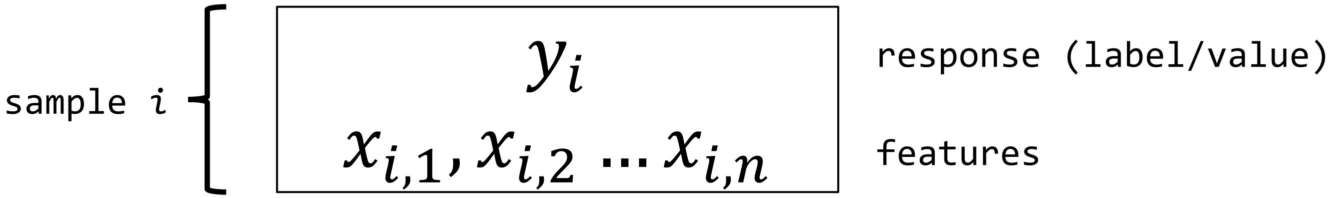
$x_{i,1}, x_{i,2} \dots x_{i,n}$

unsupervised

ML learns the relationship between the features and response

ML learns patterns/groupings

Terminology



sample i

y_i

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

response (label/value)

features

sample i

- sample
- data point
- observation

y_i

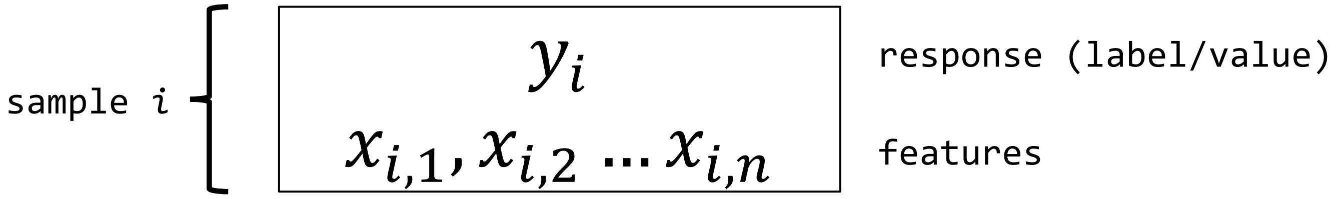
- response
- target
- class (if categorical)
- outcome
- dependent variable

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

- features
- predictors
- descriptors
- attributes
- covariates
- independent variables

many terms in English, but the **<u>math</u>** is always the same!

Discrete/categorical or continuous values!



sample i

y_i

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

response (label/value)

features

examples

response y_i

- a sample’s disease status (discrete)
- a sample’s height/length (continuous)
- a house’s market value (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)
- the age of a house (continuous)

Why use machine learning?

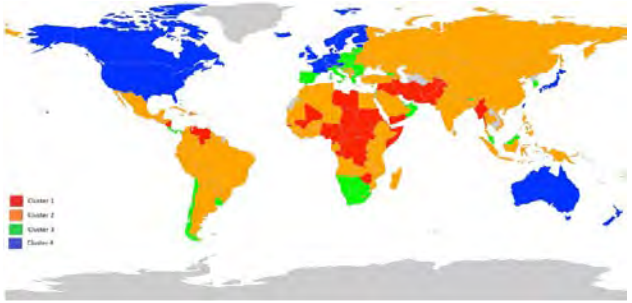
we want to

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

examples

- supervised
  - should I sell this stock?
  - 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?
  - 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?

<u>supervised</u>

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

<u>unsupervised</u>

- learn underlying structure of data

Overview of Machine Learning Process

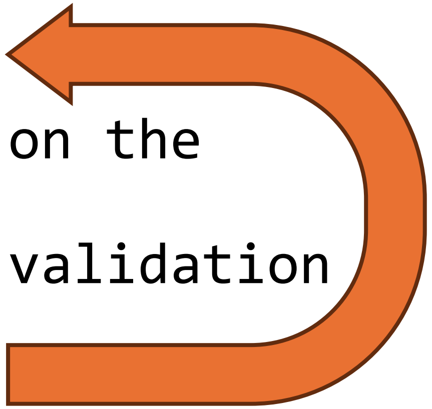
1. Define the problem to be solved.

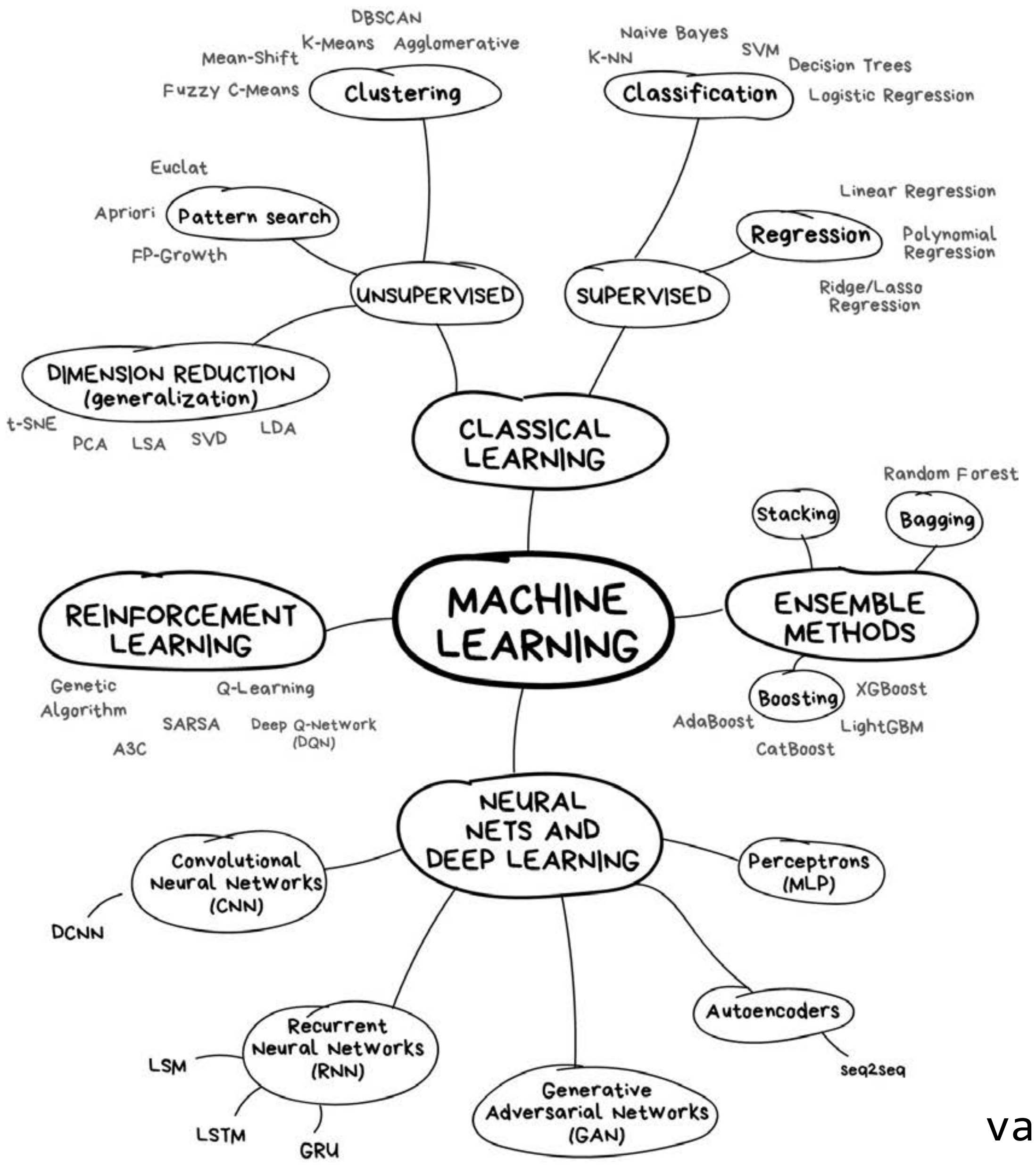
Datasets? Input features?
Targets? Evaluation metrics?

2. Split the data into train / validation / test.
3. Run the validation loop:
    a. Choose a set of models.
    b. Train each model by optimizing its parameters on the training set.
    c. Evaluate the performance of each model on the validation set.
    d. Repeat until performance is satisfactory.
4. Evaluate final performance on the test set.

Overview of Machine Learning Process

1. Define the problem to be solved.
2. Split the data into train / validation / test.
3. Run the validation loop:
    a. Choose a set of models.
    b. Train each model by optimizing its parameters on the training set.
    c. Evaluate the performance of each model on the validation set.
    d. Repeat until performance is satisfactory.
4. Evaluate final performance on the test set.




MACHINE LEARNING
CLASSICAL LEARNING
UNSUPERVISED
SUPERVISED
Clustering
DBSCAN
K-Means
Agglomerative
Mean-Shift
Fuzzy C-Means
Pattern search
Euclat
Apriori
FP-Growth
DIMENSION REDUCTION (generalization)
t-SNE
PCA
LSA
SVD
LDA
Classification
Naive Bayes
K-NN
SVM
Decision Trees
Logistic Regression
Regression
Linear Regression
Polynomial Regression
Ridge/Lasso Regression
ENSEMBLE METHODS
Stacking
Bagging
Random Forest
Boosting
XGBoost
AdaBoost
LightGBM
CatBoost
REINFORCEMENT LEARNING
Genetic Algorithm
Q-Learning
SARSA
Deep Q-Network (DQN)
A3C
NEURAL NETS AND DEEP LEARNING
Convolutional Neural Networks (CNN)
DCNN
Perceptrons (MLP)
Recurrent Neural Networks (RNN)
LSM
LSTM
GRU
Generative Adversarial Networks (GAN)
Autoencoders
seq2seq

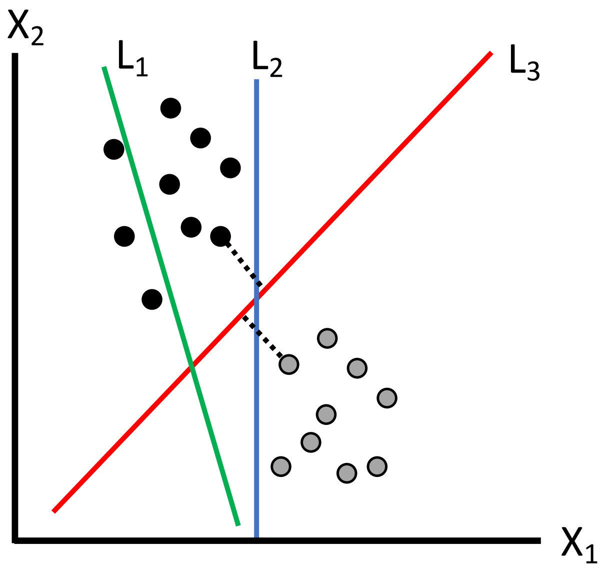
vas3k.com

Overview of Machine Learning Process

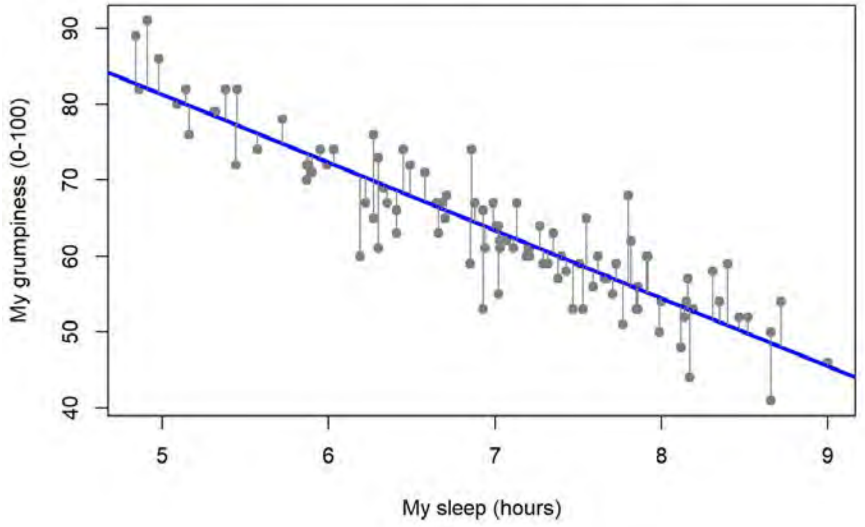
1. Define the problem to be solved.
2. Split the data into train / validation / test.
3. Run the validation loop:
    a. Choose a set of models.
    b. Train each model by optimizing its parameters on the training set.
    c. Evaluate the performance of each model on the validation set.
    d. Repeat until performance is satisfactory.
4. Evaluate final performance on the test set.

Model Training

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



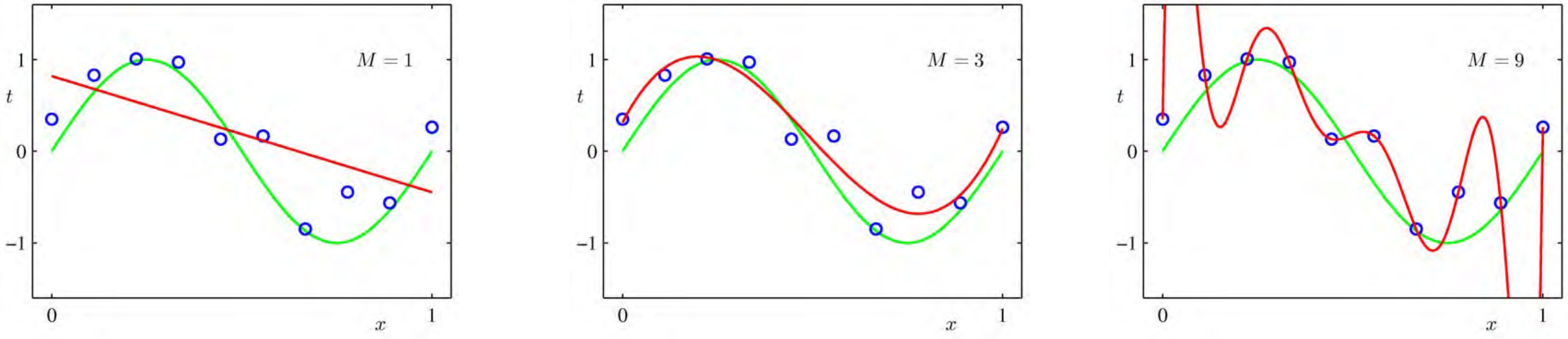
e.g. find slope of line that **best** predicts training values



Overview of Machine Learning Process

1. Define the problem to be solved.
2. Split the data into train / validation / test.
3. Run the validation loop:
    a. Choose a set of models.
    b. Train each model by optimizing its parameters on the training set.
    c. Evaluate the performance of each model on the validation set.
    d. Repeat until performance is satisfactory.
4. Evaluate final performance on the test set.

Model Complexity

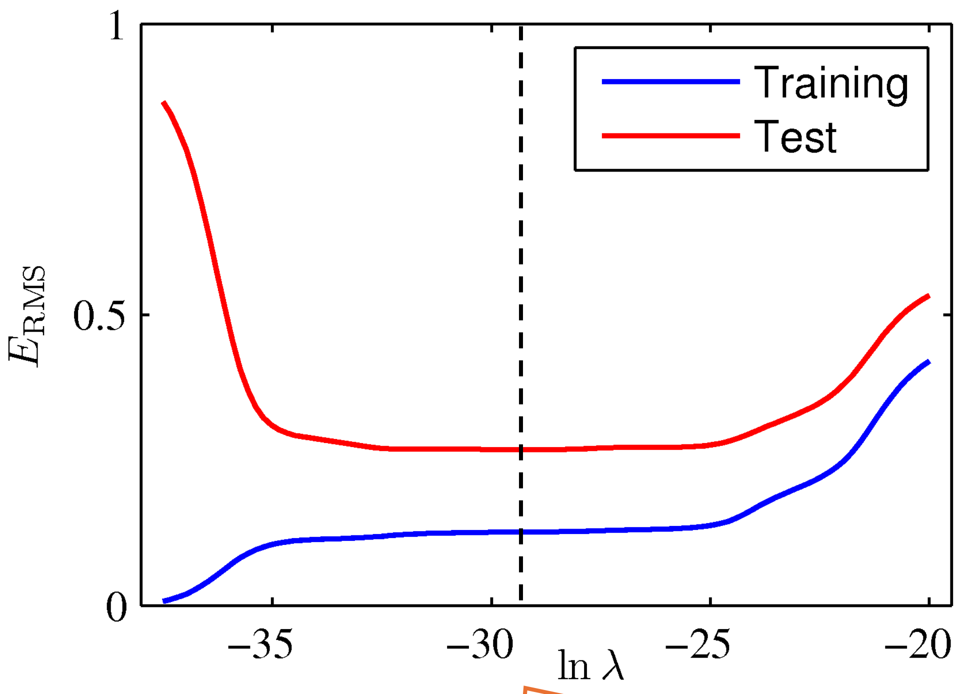


Underfitting
(misses general trend)

Overfitting
(captures noise)

Figure credit: Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning*.

Effect of Complexity on Test Performance



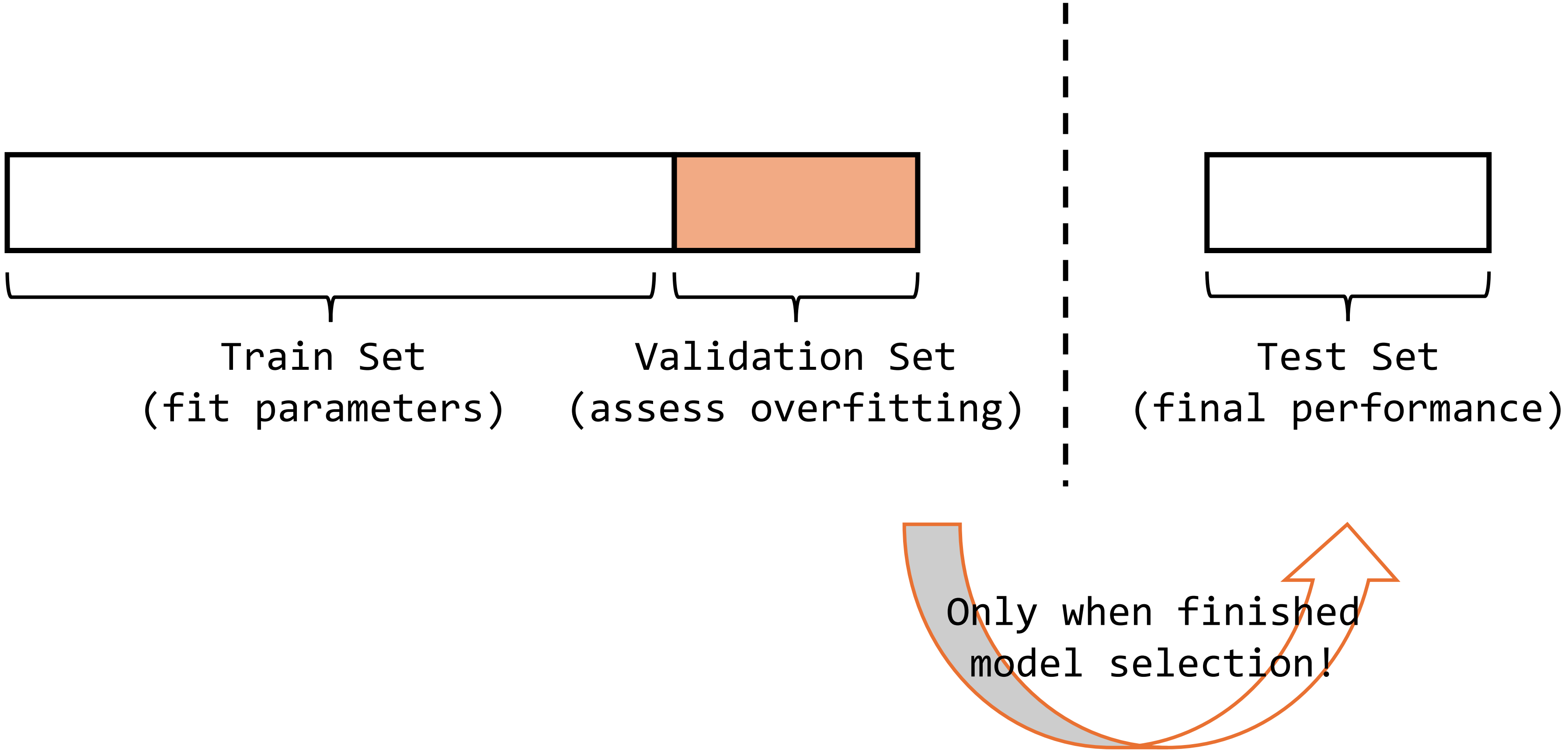
Best model has this value!

Figure credit: Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning*.

Overview of Machine Learning Process

1. Define the problem to be solved.
2. Split the data into train / validation / test.
3. Run the validation loop:
    a. Choose a set of models.
    b. Train each model by optimizing its parameters on the training set.
    c. Evaluate the performance of each model on the validation set.
    d. Repeat until performance is satisfactory.
4. Evaluate final performance on the test set.

Role of the Validation Set



Train Set
(fit parameters)

Validation Set
(assess overfitting)

Test Set
(final performance)

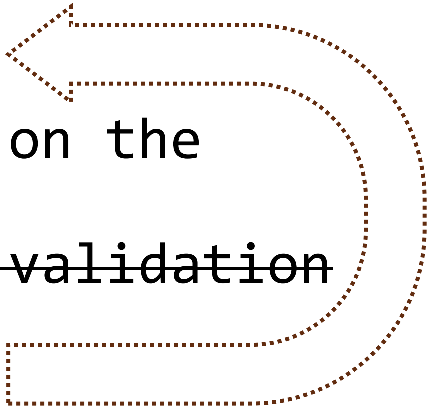
Only when finished model selection!

Overview of Machine Learning Process

1. Define the problem to be solved.
2. Split the data into train / validation / test.
3. Run the validation loop:
   a. Choose a set of models.
   b. Train each model by optimizing its parameters on the training set.
   c. Evaluate the performance of each model on the validation set.
   d. Repeat until performance is satisfactory.
4. Evaluate final performance on the test set.

Simplified Machine Learning Process

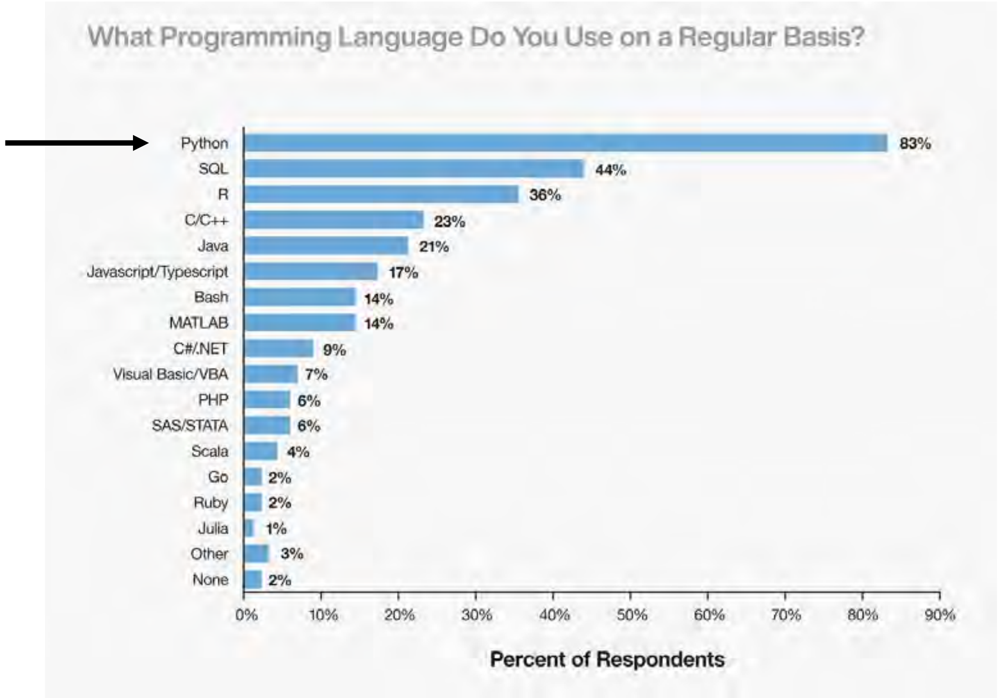
1. Download a dataset from the internet.
2. Use the predefined train / test split.
3. ~~Run the validation loop:~~
    a. Choose a ~~set of~~ model~~s~~.
    b. Train ~~each~~ model by optimizing its parameters on the training set.
    ~~c. Evaluate the performance of each model on the validation set.~~
    ~~d. Repeat until performance is satisfactory.~~
4. Evaluate final performance on the test set.

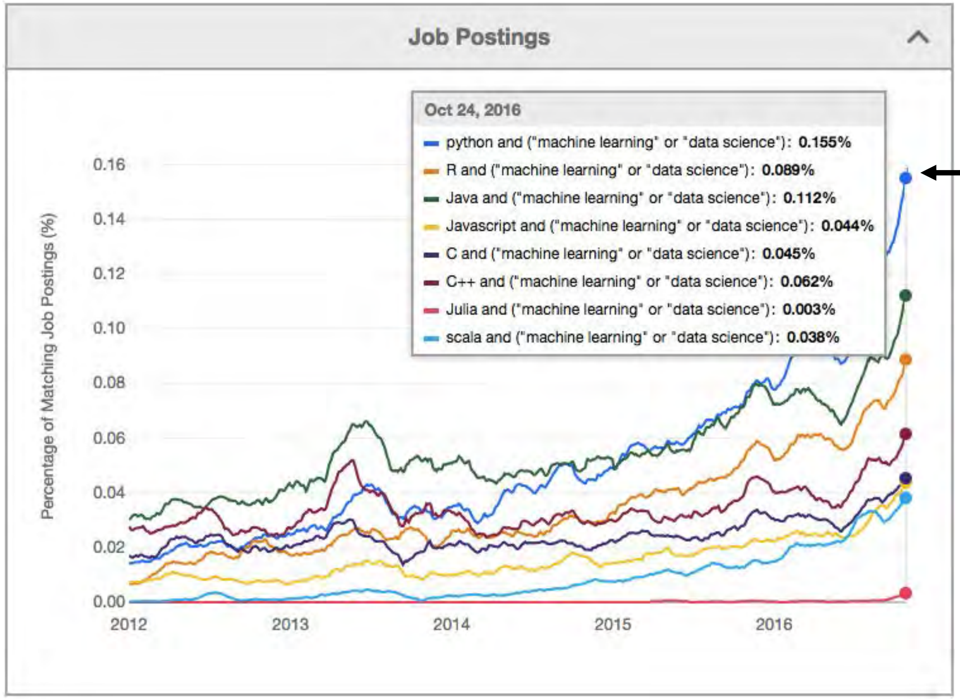


Machine learning in python™

- 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

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!

Open the iPython
notebook from this link!

https://github.com/PrincetonUniversity/intro_machine_learning/tree/main/day1

https://qrco.de/bgYIMl

Intro to *K*-nearest neighbors (KNN)

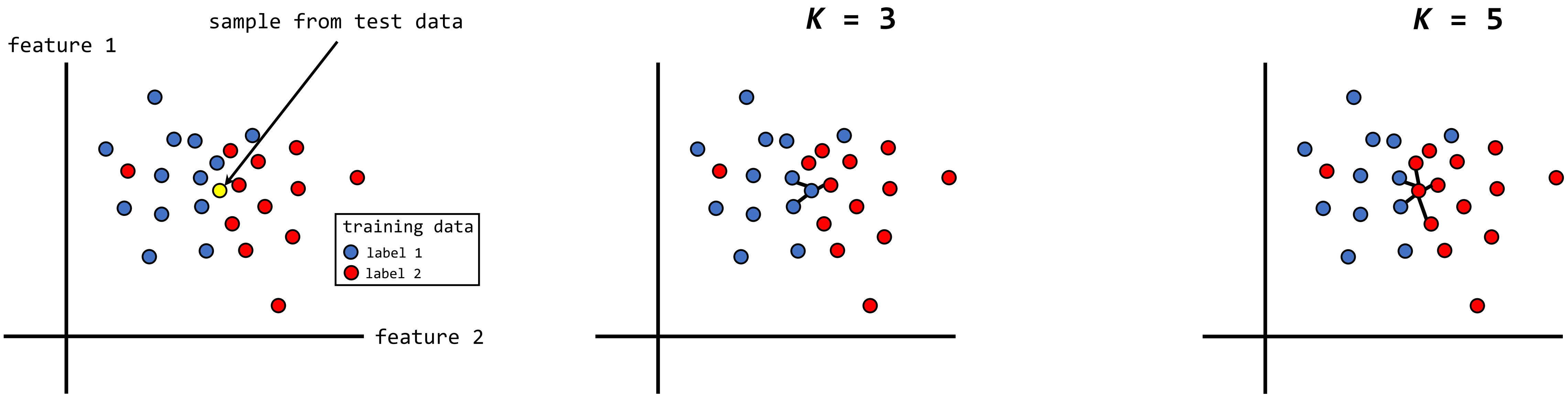
- simple but powerful
- can be used for classification *or* regression!
- algorithm
  1. for a given test sample (yellow dot), find the *K* nearest training samples in feature space

  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


sample from test data
feature 1
feature 2
training data
● label 1
● label 2
K = 3
K = 5