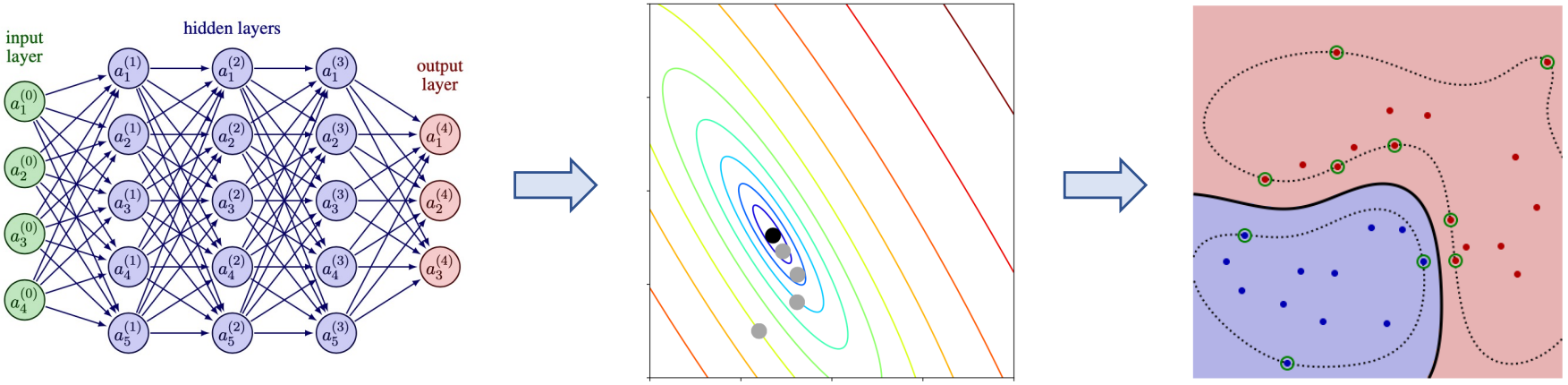
CS273A:
Introduction to Machine Learning



Prof. Alexander Ihler

Fall 2024

Today's Lecture

Introduction to Machine Learning

Course Organization

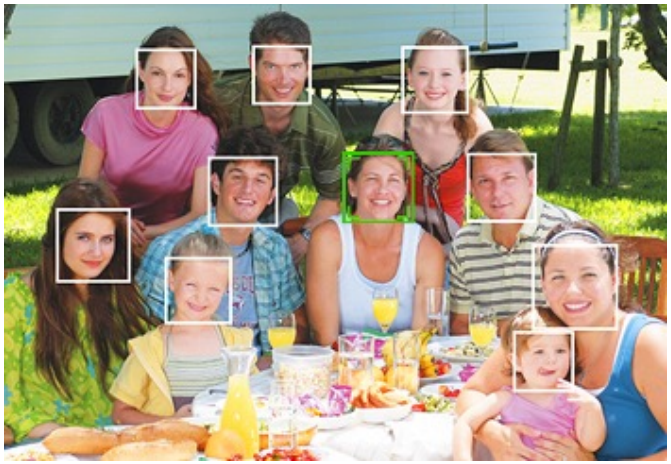
Supervised Learning

Data Exploration

Driving

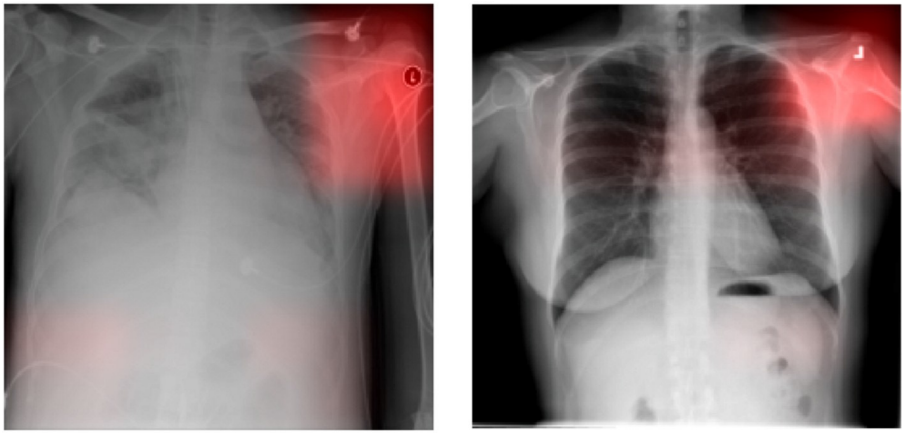
From shutterstock.com

Photos



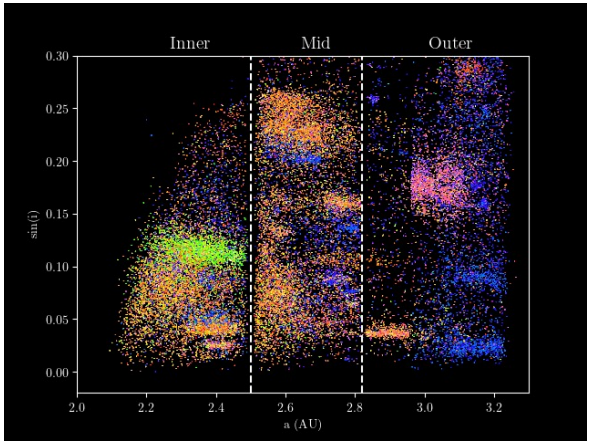
From medium.com

Medicine



From Zech et al, PLOS Medicine, 2018

Science



From astroml.org

Key Ideas in Machine Learning

- Machine learning models
  - Map inputs to outputs, e.g.,
    - Image pixels - > identification of object in image
    - Speech signal -> identification of a word
    - Text -> prediction about the text
- Models are learned rather than designed
  - Traditional AI, science, etc: hand-designed models
  - Machine learning: models are learned from data
- Ideal for problems with a lot of data and little theory

Machine Learning on your Phone

Applications driven by Machine Learning

Speech recognition
Speaker verification

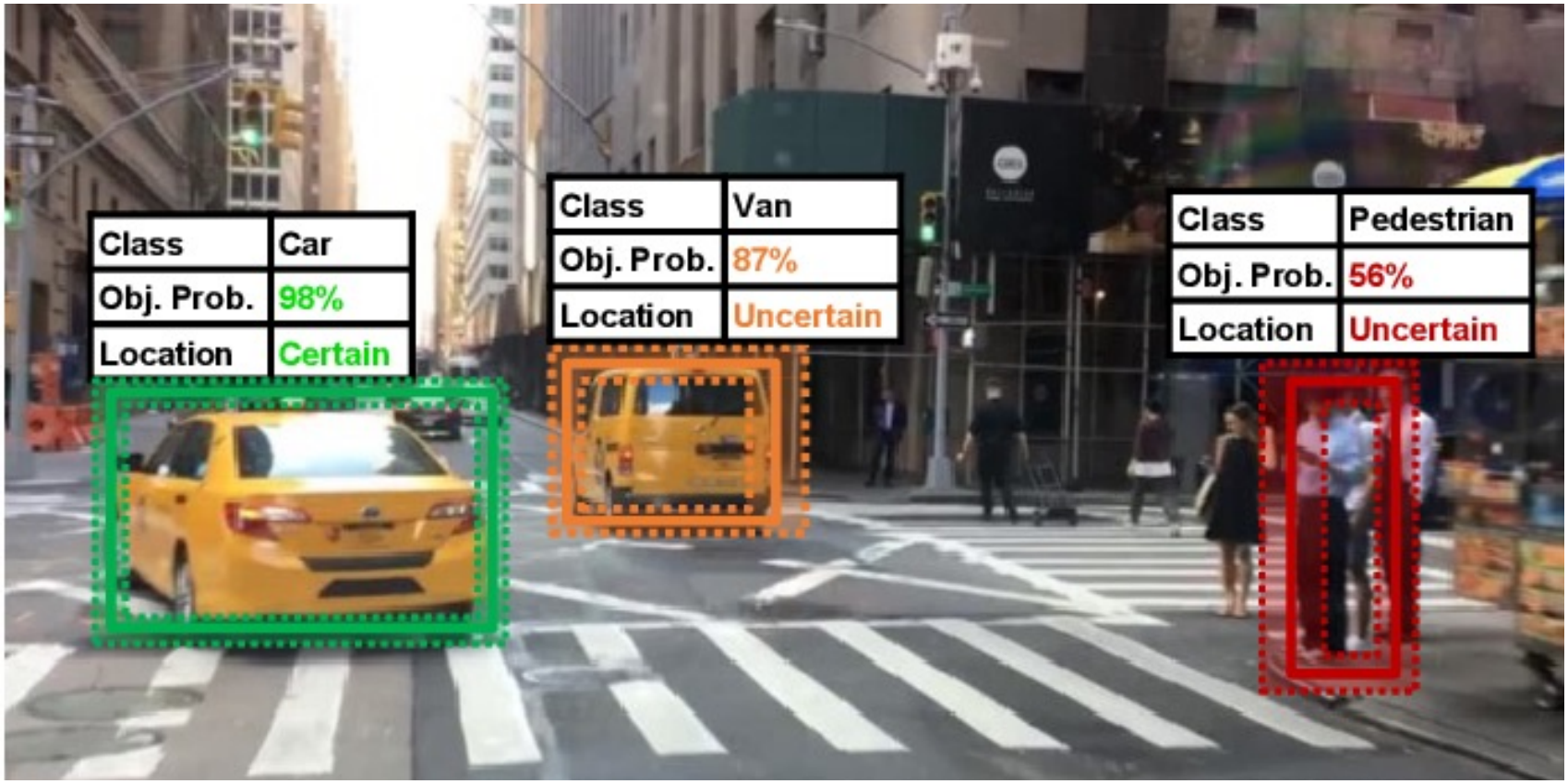
Fingerprint verification

Face recognition
Object recognition

Text autocomplete

…. and more

Machine Learning in Cars



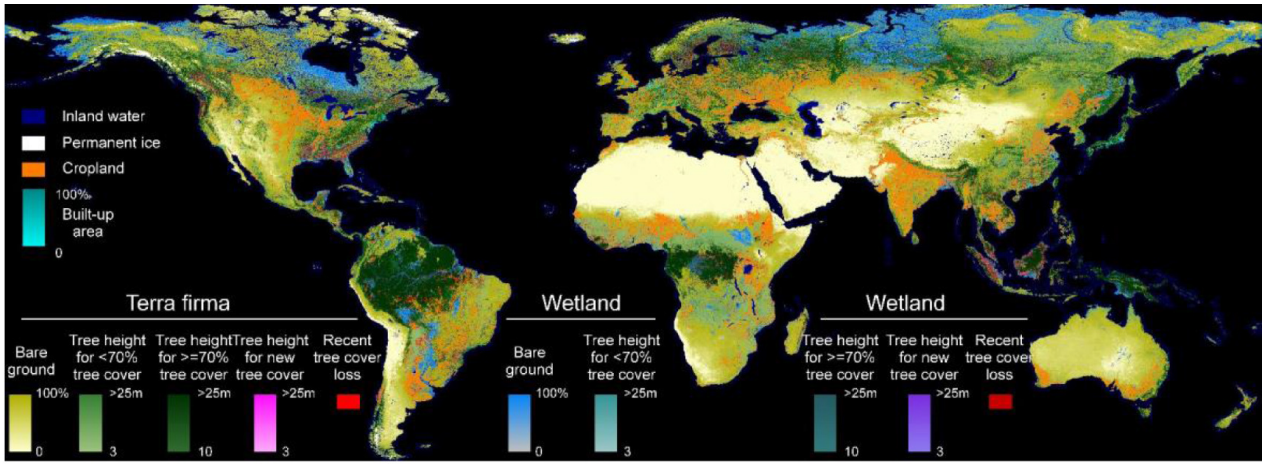
From Feng et al, IEEE Transactions on Intelligent Transportation Systems, 2020

Machine Learning in Science

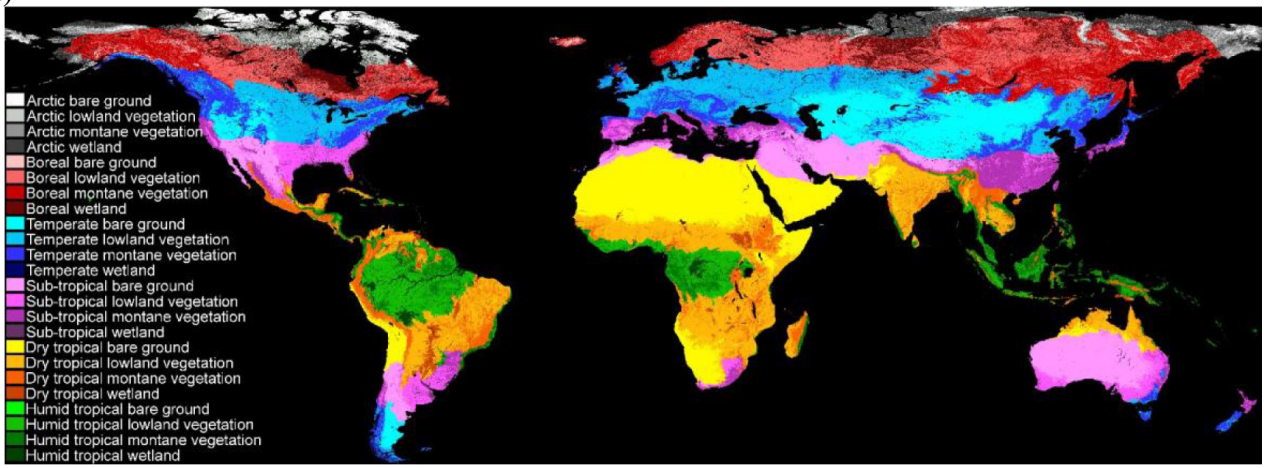
Global landcover maps
produced
by machine learning algorithms
from satellite images

Machine learning is used
by NASA and other agencies to
produce such maps for scientists

e.g, very useful for automated
tracking of changes over time



a)



b)

From Hansen et al, Environmental Research Letters, March 2022

Machine Learning in Government

Machine learning algorithm predicts building “footprints” from aerial images

Images from https://www.esriuk.com/en-gb/news/we-talk-tech/when-remote-sensing-meets-artificial-intelligence-and-gis

Machine Learning to Count Trees

From https://www.gislounge.com/using-remote-sensing-to-count-trees/

Machine Learning in Medicine

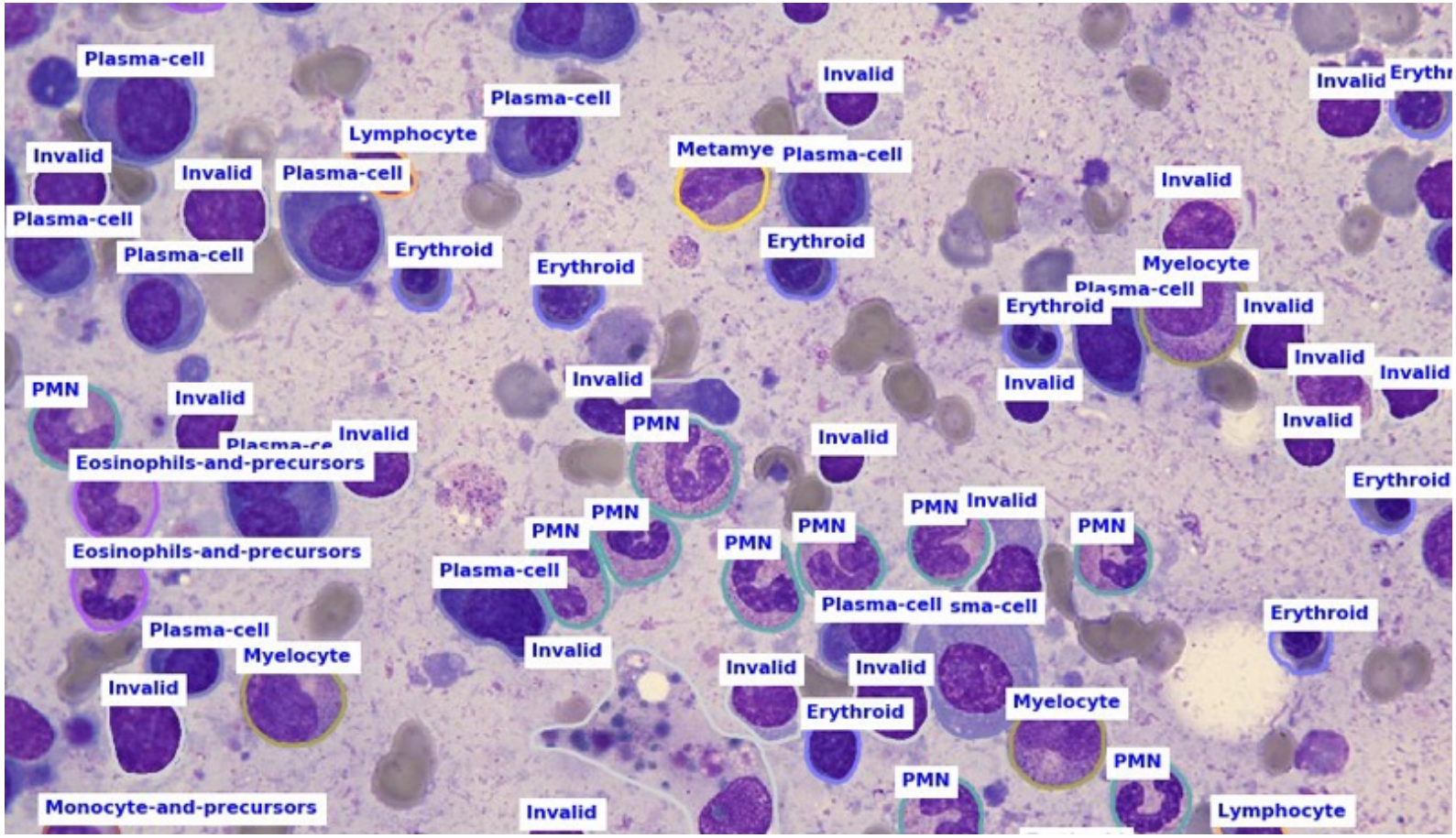
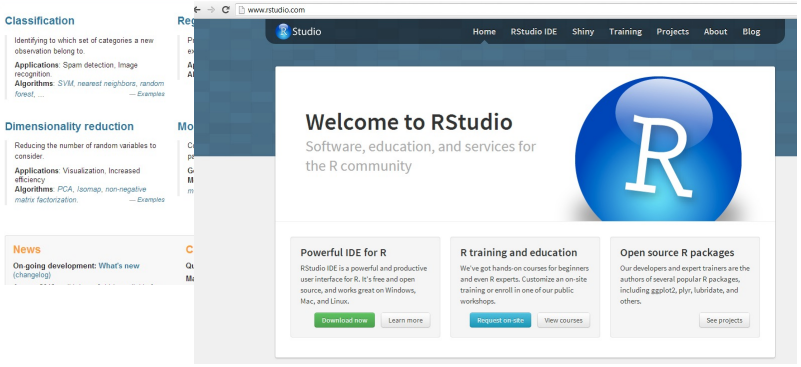
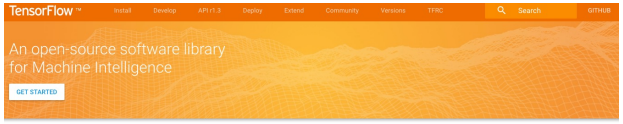


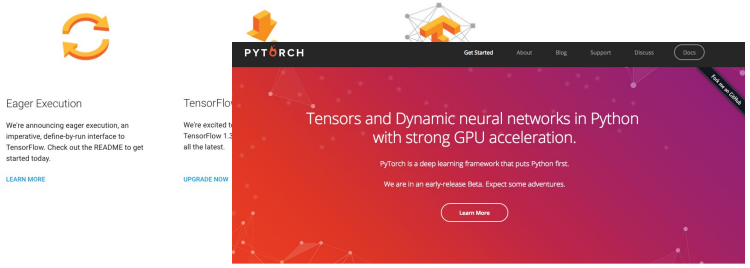
Image from https://www.theimagingsource.com/media/blog/archive/20210114/

General-purpose open-source software for machine learning

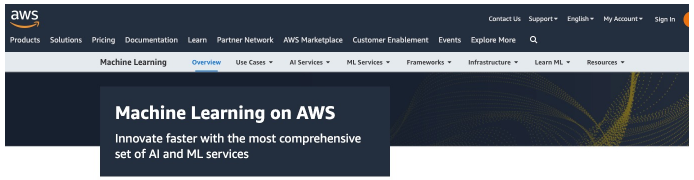


Open-source software from industry with a focus on deep learning





Cloud-based commercial systems for machine learning development



Machine Learning and AI?

- How does machine learning relate to AI?
  - Historically AI is broadly concerned with "intelligence"
    - Knowledge representation, logic, rule-bases, reasoning, etc
    - …and learning from data was seen as just one aspect of AI
    - 30+ years ago AI research was only 10% about ML
  - But in the 1990's-2000's people found it hard to scale up AI
    - Building large knowledge-based systems manually is hard
    - Companies like Google, Microsoft, others, started using ML
  - By 2022, more than 90% of activity in AI was based on ML
    - Particularly on problems that have large datasets
  - Ongoing debate about whether "learning from data" is true AI

Machine Learning and Statistics?

- Isn't Statistics also about learning from data?
  - So how is Statistics different from Machine Learning (ML)?

- …..many similarities, e.g.,
  - Many models in ML (logistic models, trees, etc) came from statistics
  - Theoretical frameworks for learning are often based on statistics
  - Many statisticians also work on ML

- … but some systematic differences, e.g.,
  - ML typically focuses on problems with more inputs, large datasets
  - ML focuses on prediction; Statistics more about model interpretation
  - ML is often algorithm/software focused; statistics can be more mathematical
  - ……

Machine Learning Errors

original image

Model classifies this as “pig”

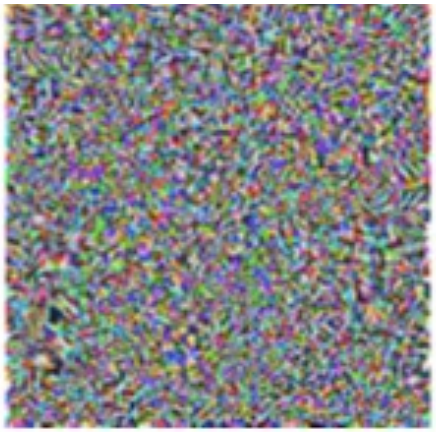
Figure from Engstrom et al, Proceedings of the International Conference on Machine Learning, 2019

Machine Learning Errors

original image

noise

+ 0.005 x



=

Model classifies this as “pig”

Model classifies this as “airplane”

Figure from Engstrom et al, Proceedings of the International Conference on Machine Learning, 2019

Pedestrian Detection Errors

vertical structures

traffic lights

car parts

tree leaves

A Lesson of Tesla Crashes? Computer Vision Can't Do It All Yet

By STEVE LOHR SEPT. 19, 2016

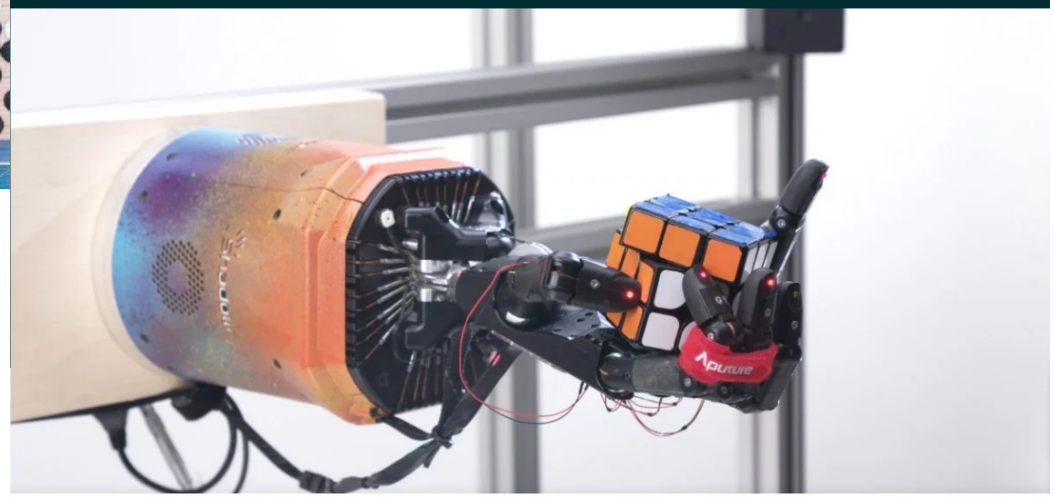
Daily Report: AlphaGo Shows How Far Artificial Intelligence Has Come

Bits

By PUI-WING TAM MAY 23, 2017

OpenAI disbands its robotics research team

Kyle Wiggers @Kyle_L_Wiggers July 16, 2021 11:24 AM



Ethics, Fairness, Bias in ML

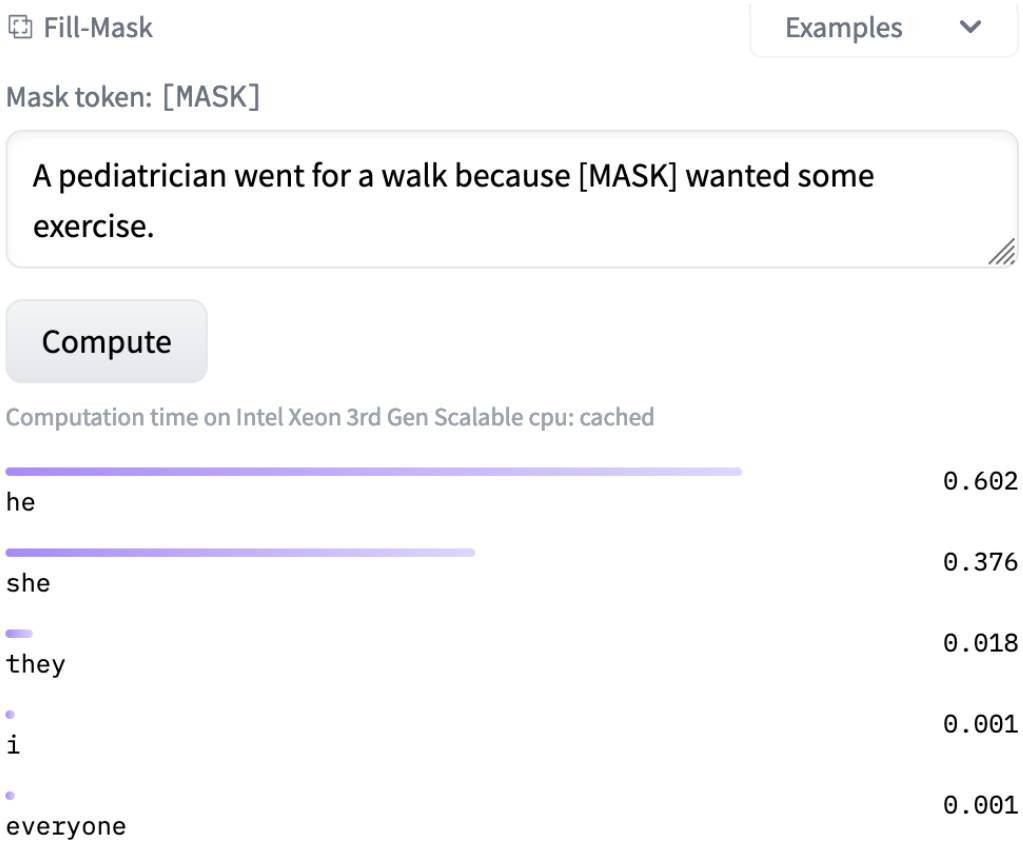
- Ethical use of machine learning
  - Machine learning is a powerful technology
  - Important to use it for public good

- Algorithmic Fairness and ML
  - Algorithmic fairness: is an algorithm fair to all individuals?
  - Examples of unfair algorithms:
    - ML model for loan applications, biased against zip-codes
    - ML model for text generation, biased against women
    - ML model for disease diagnosis, less accurate for some age groups
    - ML model for face recognition, less accurate for some racial groups

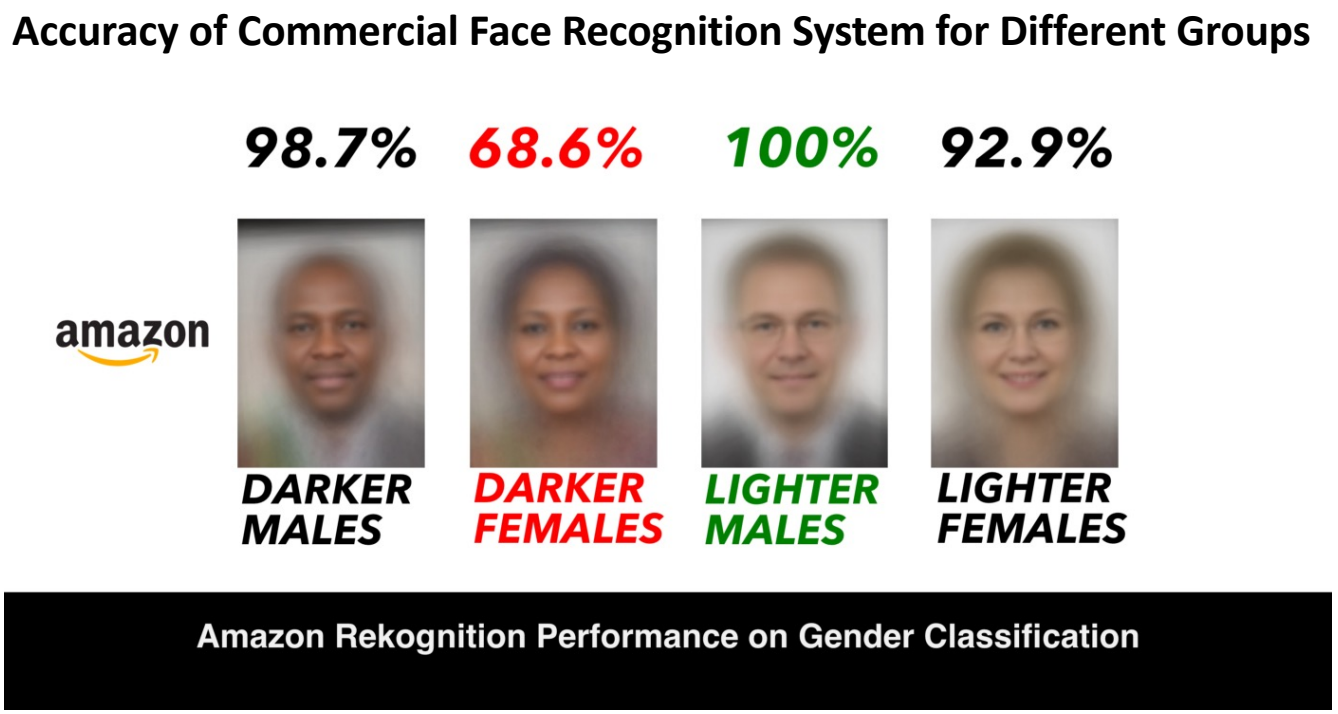
  - Bias/unfairness often inherent in data used to build models
    - Can be carried through into the ML model
    - May even be **amplified** by the ML model!

Systematic Biases in Language Models

Examples generated from https://huggingface.co/bert-base-uncased



Systematic Bias in Face Recognition



From Joy Boulamwini, MIT, medium.com, Jan 2019

A growing issue…

193 countries adopt first-ever global agreement on the Ethics of Artificial Intelligence

Unsplash/Possessed Photography | More mass-market consumer applications are expected with the development of what is known as 'assistive technologies'.

25 November 2021 | Culture and Education

Questions?

Machine Learning

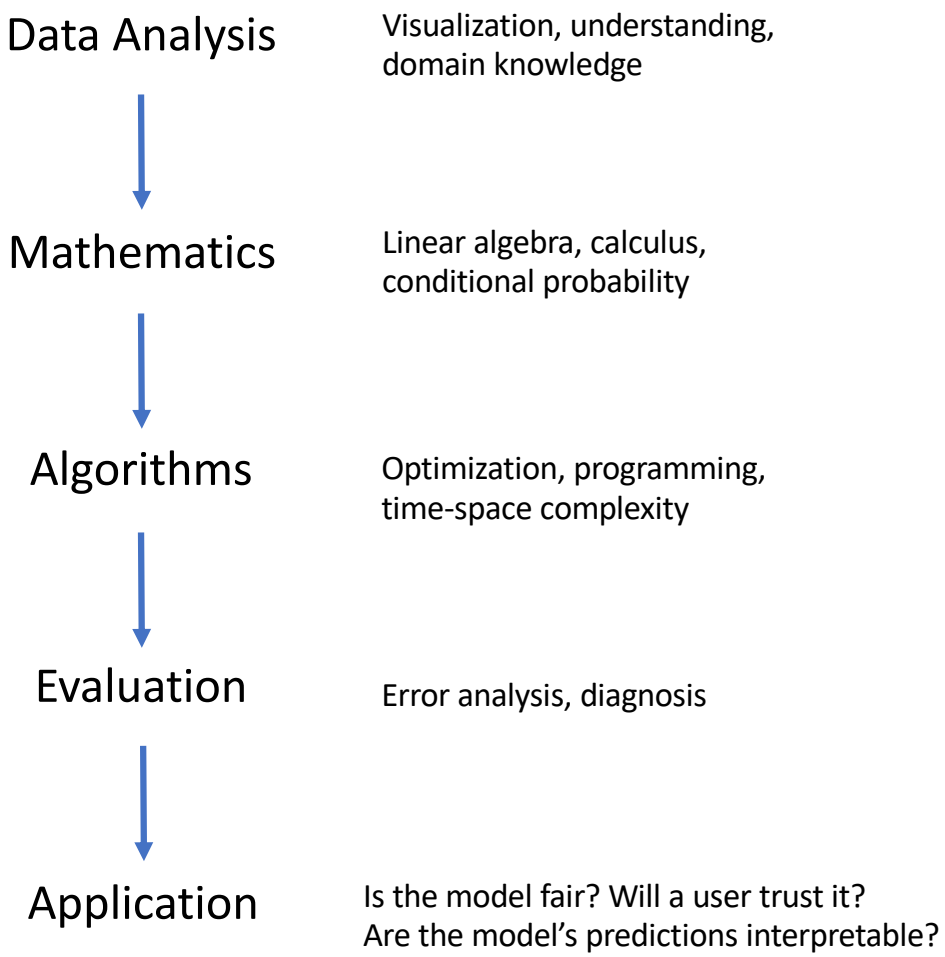
Introduction to Machine Learning

Course Organization

Supervised Learning

Data Exploration

Important Aspects of Machine Learning



Course Structure

Course Staff	
Instructor	Prof. Alex Ihler
TAs	Ali Derakhshan Soyeon Kwon

Teaching		
Lecture	MWF 3:00-4:00	RH 101
Office Hours	See Canvas	

Reading Resources

- No required textbook

- **Course notes** on Canvas (in "files": check for updates)
- Lecture recordings available (next day or so)

- Supplemental books, free online:
  - Daume, A Course in Machine Learning
  - Barber, Bayesian Reasoning and Machine Learning.
  - Hastie, Tibshirani, and Friedman, The Elements of Statistical Learning.
  - MacKay, Information Theory, Inference, and Learning Algorithms.

- Additional, more advanced suggested books as well
  - But, not required!

How to Contact Us

- Use (**only!**) the EdD discussion board for questions
  - Broadcast your question to "all" if it might be useful to other students
  - Feel free to answer other student's questions
  - We will try to answer your questions as soon as we can
  - Can send individual messages to / tag TAs or Instructor if needed

Please **don't use email or canvas messages** unless there is no other option….

Grading

- Homeworks 35%
  - Homeworks will mostly involve Python
    - Application of concepts learned in class
  - 6 homeworks
  - Late homeworks: 10% penalty per day, 3 day maximum
  - Lowest-scoring homework will be dropped

- Exams: 50%
  - Midterm 20%
  - Final 30%
  - Exams will be in-person, in lecture hall

- Project: 15%
  - 3-person teams

Projects

Project Teams

- Team size = 3
- Weeks 4 to 10 (approx)

- Two deadlines:
  - Team Formation and Dataset Selection (20% Credit)
  - Project Submission (80% Credit)

Additional details and deadlines will be posted on Canvas

Academic Integrity

- Please read the guidelines on academic integrity on the Canvas Website.
  - It is the responsibility of each student to be familiar with UCI's Academic Integrity Policies and UCI's definitions and examples of academic misconduct.
  - Violating these policies can result in a student receiving a failing grade in the class.

- For assignments
  - You can discuss the assignments verbally with other class members
  - You cannot look at or to copy anyone else's written material: notes, solutions, code, etc.
  - All problem solutions and code submitted must be material you have personally written

- For class projects
  - All text/figures in reports submitted must be written by members of your project team.
  - Code used in class projects can be both your code and publicly-available code.

Questions?

Machine Learning

Introduction to Machine Learning

Course Organization

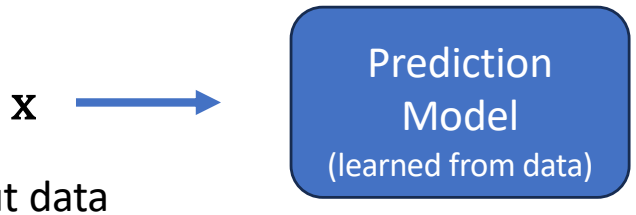
Supervised Learning

Data and Visualization

Types of Machine Learning

- Supervised Learning
  - Classification
  - Regression
- Unsupervised Learning
  - Clustering
  - Dimension reduction
- Active Learning
- Reinforcement Learning
- Other variants
  - Semi-supervised, online learning, ….

Supervised Learning



Input data
(typically real valued numbers)

Referred to as a "feature vector"

Output
(what the model predicts)

Classification:

y can take a finite set of K values, e.g.,

$K = 2, \quad y \in \{0, 1\}$

$K = 10, \quad y \in \{1, 2, \ldots, 10\}$

Regression:

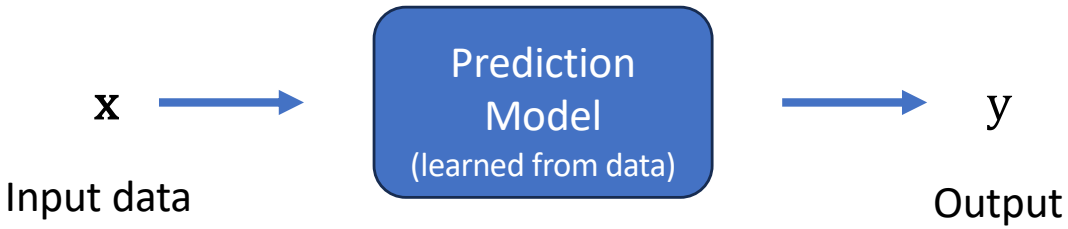
y is real-valued, e.g,

y is any value on the real-line, or

$y > 0$

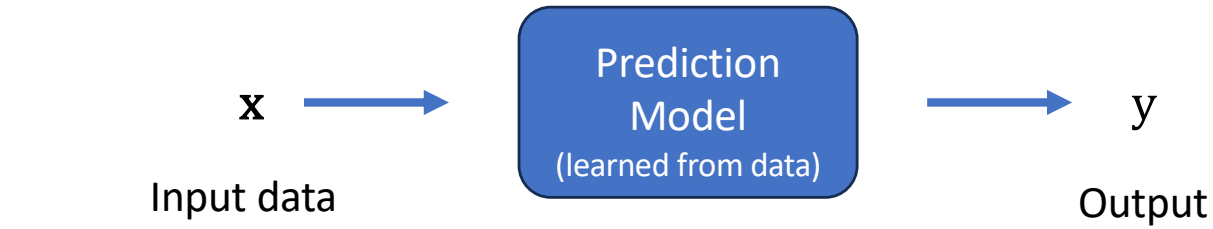
$y \in [a,b]$

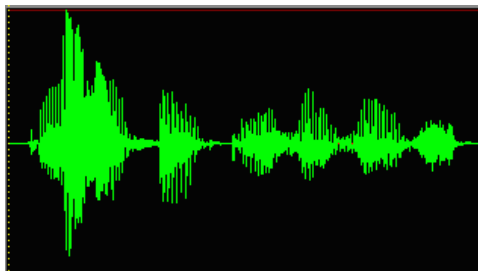
Examples of Classification



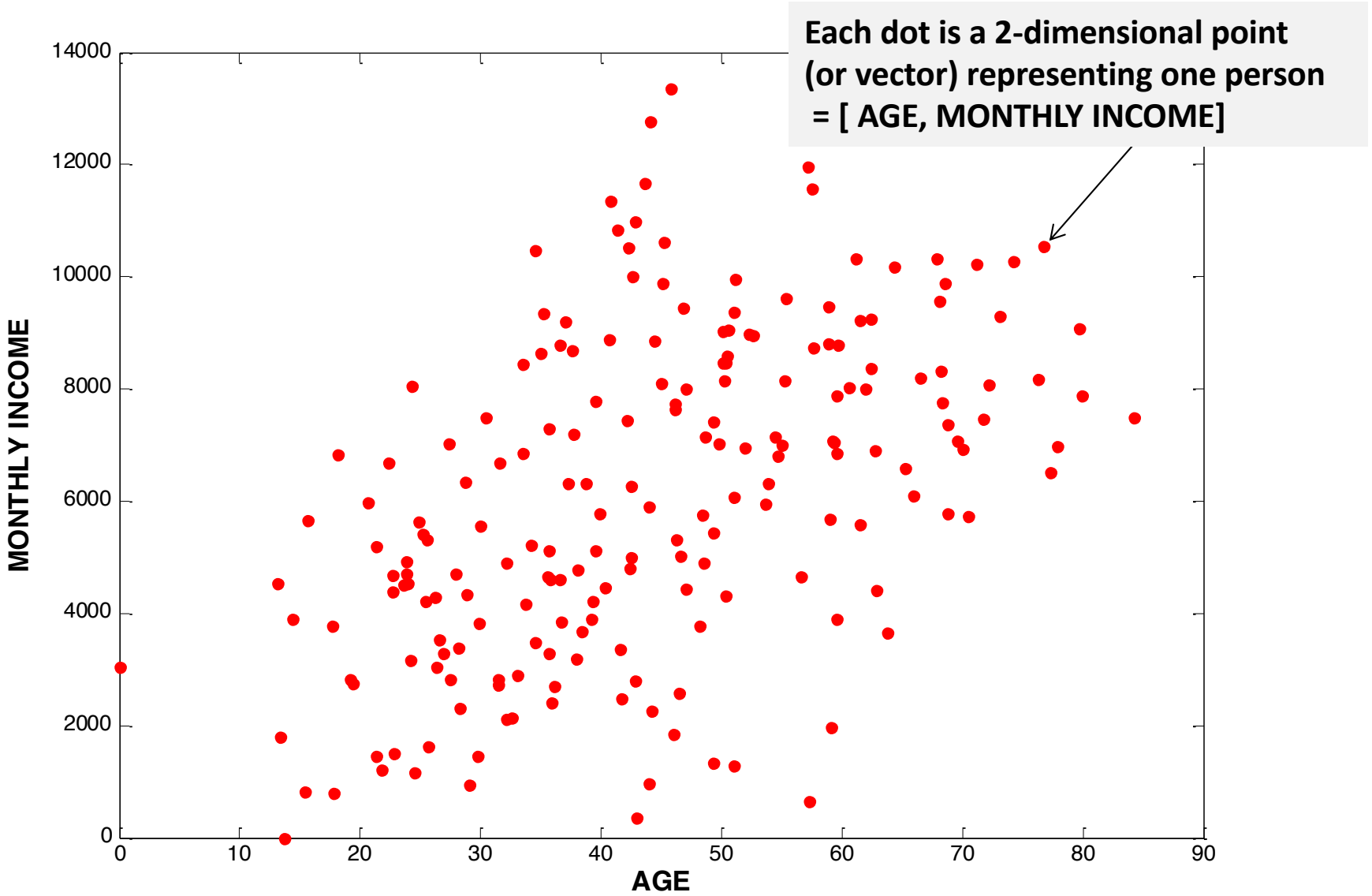
Text of email	→	Spam or not (K = 2)
Pixels in image	→	Type of animal (K = 500 or more)
Speech signal	→	English word (K = 100,000 or more)
Loan application	→	Grant loan or not (K = 2)
Medical data	→	Diagnosis (K = 2 or more)

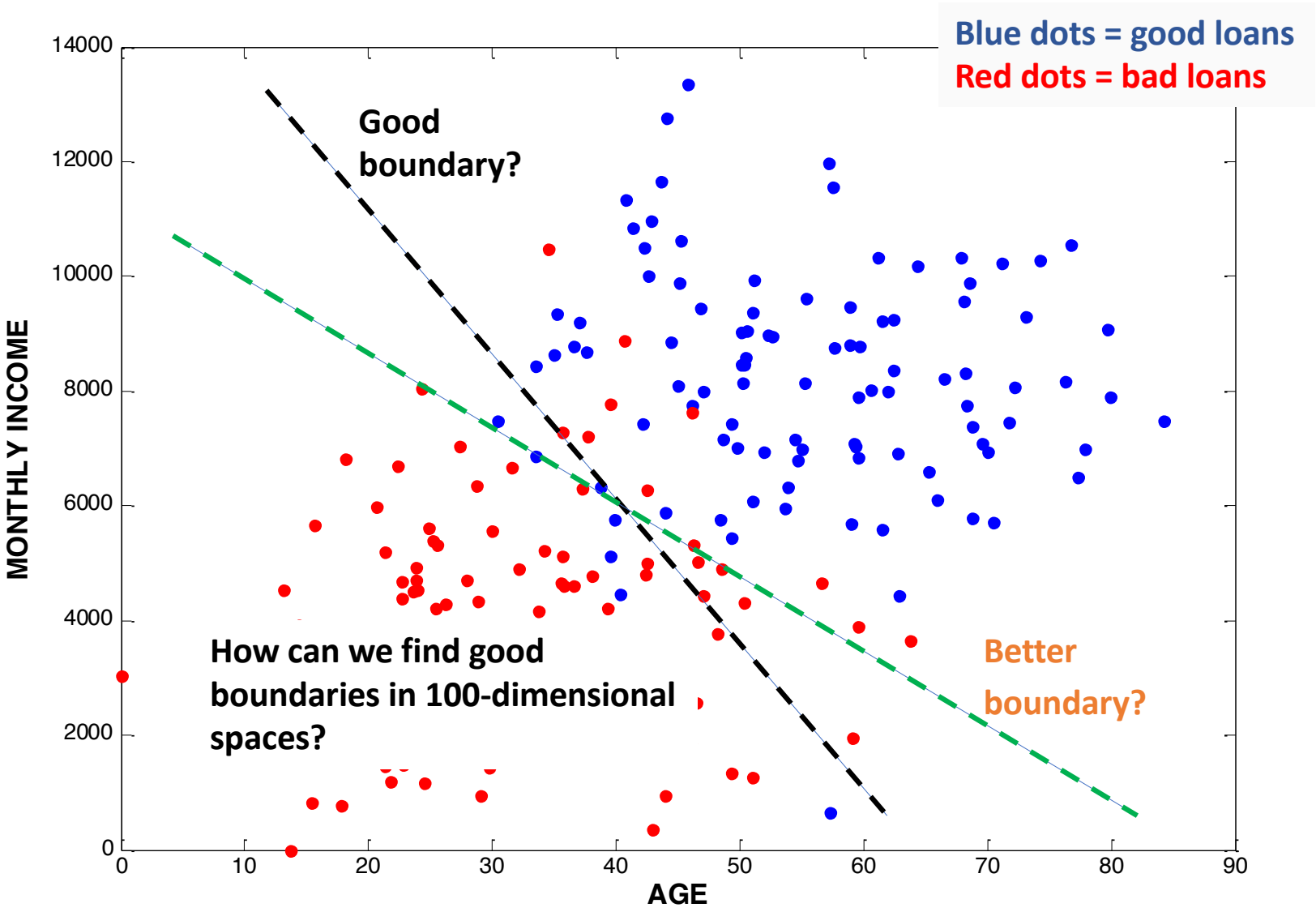
Speech Recognition





"Dog"
"Cat"
"Red"
"Two"
..
..
..
..


Each dot is a 2-dimensional point
(or vector) representing one person
= [AGE, MONTHLY INCOME]
MONTHLY INCOME
AGE
14000
12000
10000
8000
6000
4000
2000
0
0
10
20
30
40
50
60
70
80
90


Blue dots = good loans
Red dots = bad loans
Good boundary?
How can we find good boundaries in 100-dimensional spaces?
Better boundary?
MONTHLY INCOME
AGE
14000
12000
10000
8000
6000
4000
2000
0
0
10
20
30
40
50
60
70
80
90

Two Phases of Supervised Learning

1. Learn the model
   - Machine learning algorithm is provided with "training data"
   - Training data = dataset of <x, y> pairs, i.e., input-output pairs
   - Algorithm then learns a model from training data
     (this is what we will spend much of the quarter discussing)

2. Make predictions
   - Given a new x, where y is known….
     …. Use the learned model to make a prediction for y

In practice there may be other intermediate phases, e.g.,

- Using validation data to find good "hyperparameters" for our learning algorithm
- Testing the model on test data

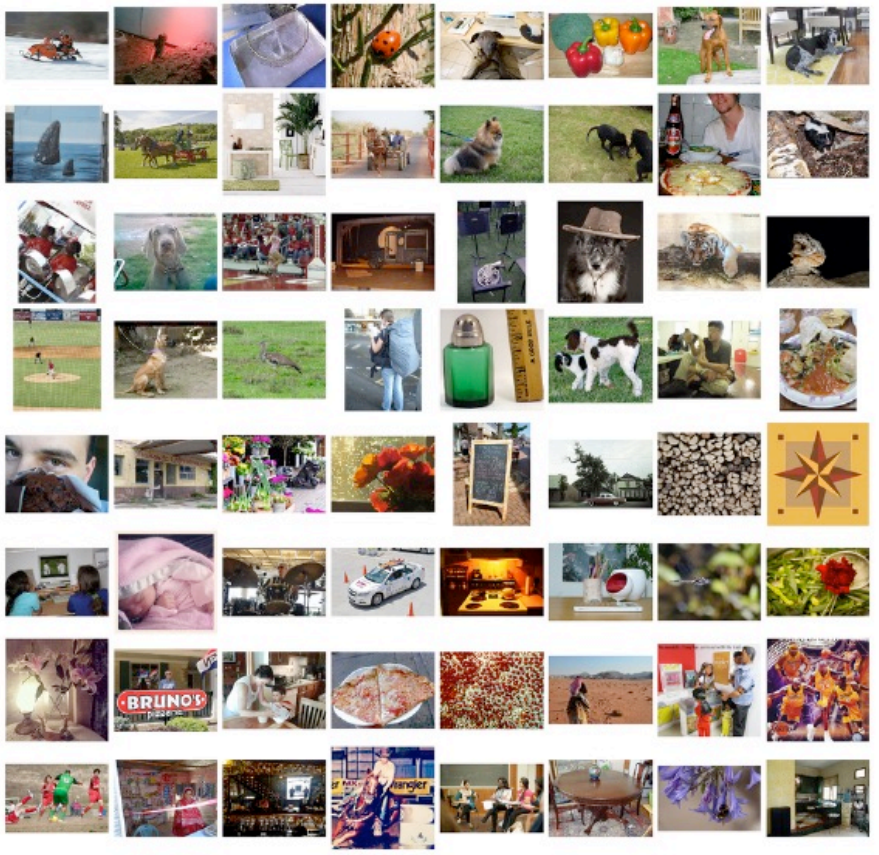
Image Classification

ImageNet

A testbed for evaluating
image classification algorithms

1000 different classes

14 million images



From Russakovsky et al, ImageNet Large Scale Visual Recognition Challenge, 2015

Error Rates over Time

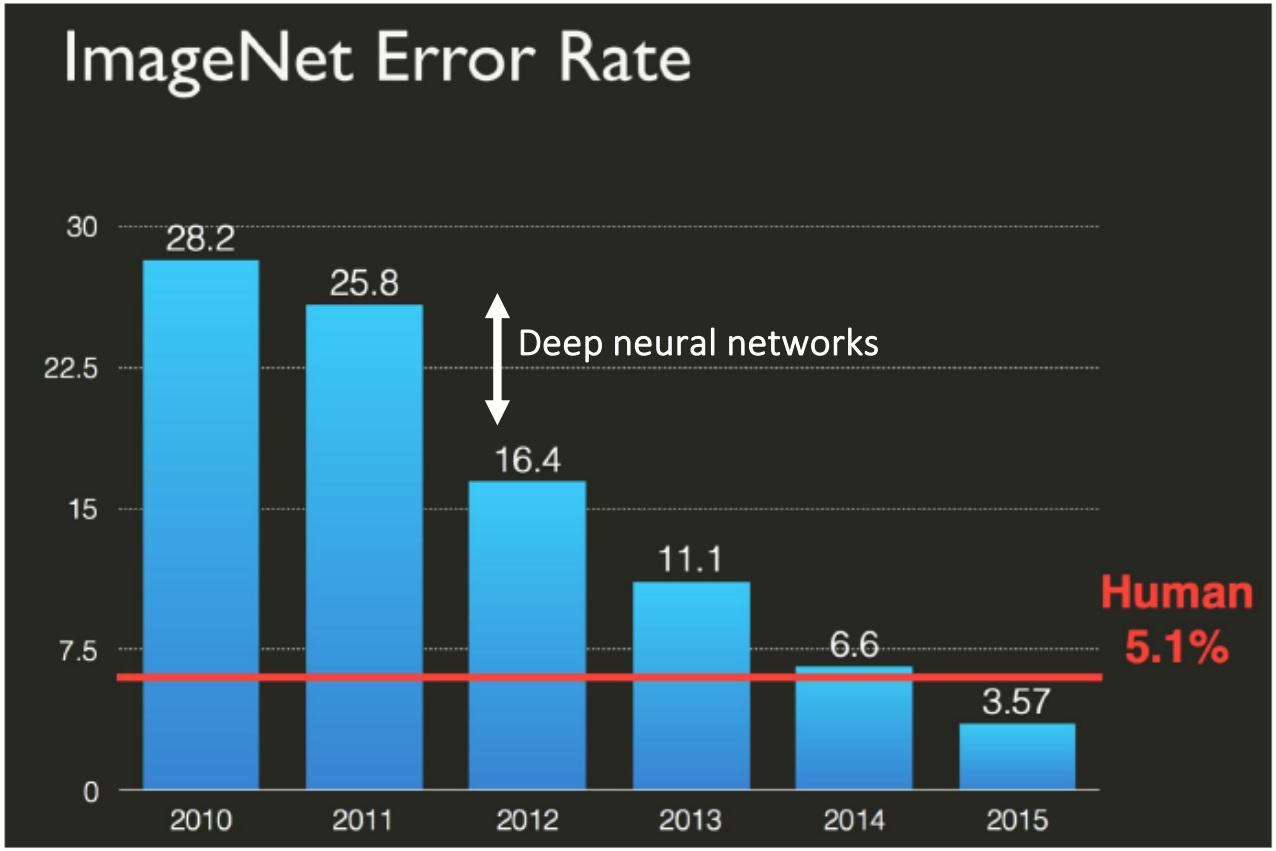
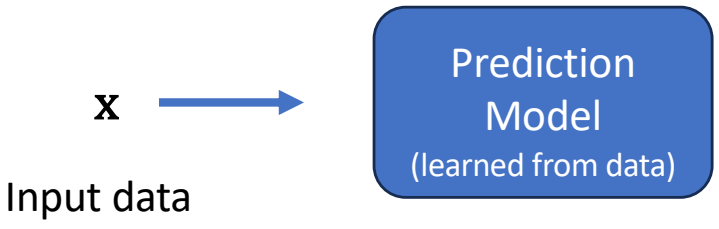


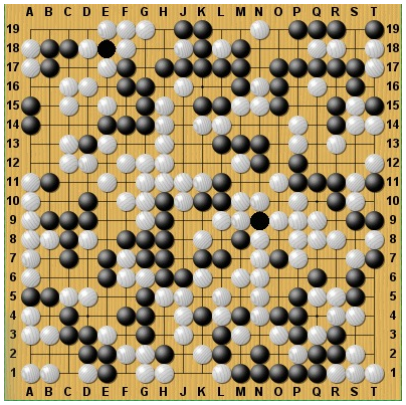
Figure from Kevin Murphy, Google, 2016

Examples of Regression (real-valued y)



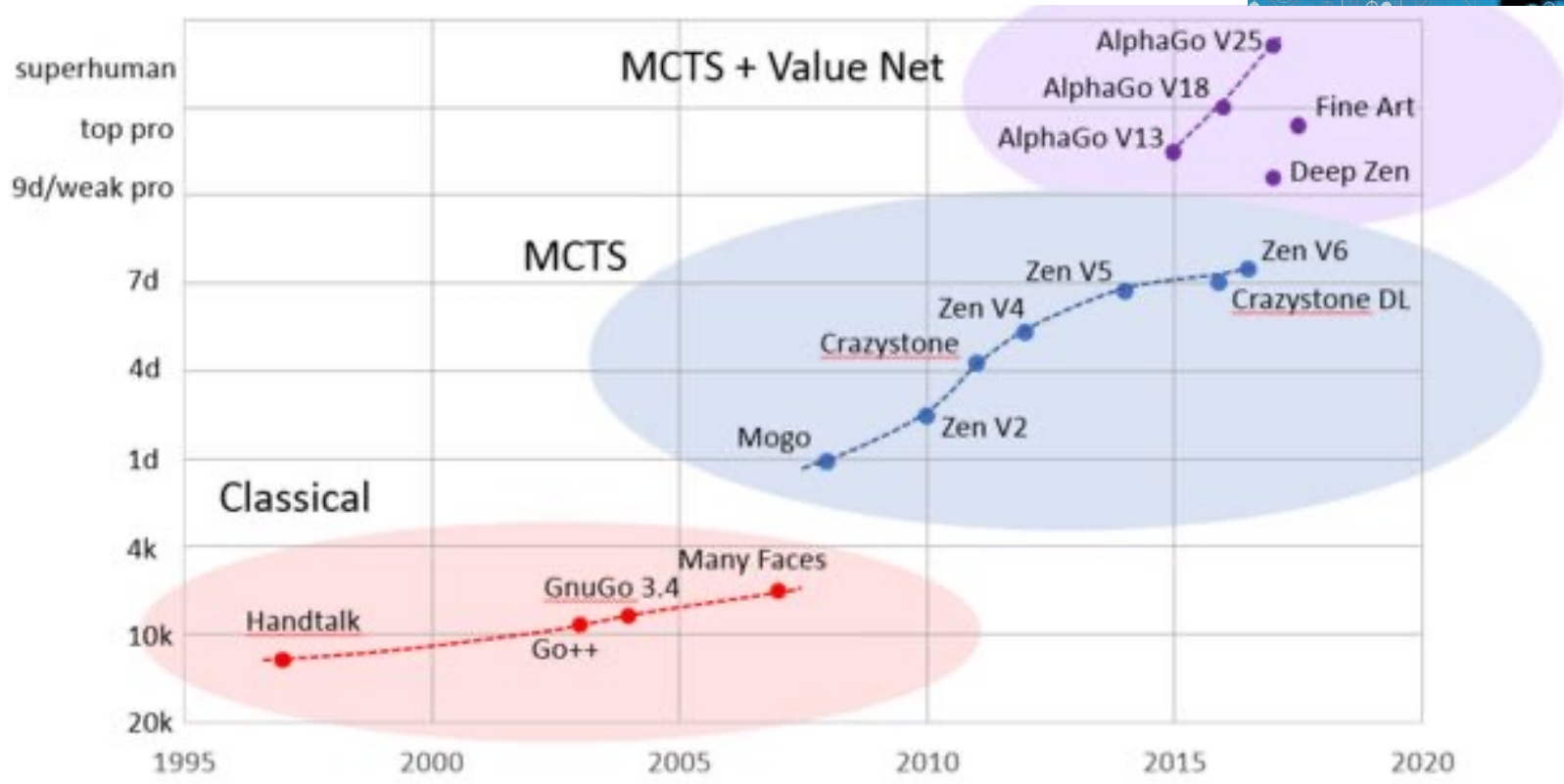
Financial data	→	Stock market value tomorrow (y > 0)
Laser signals	→	Distance of object to vehicle (y > 0)
Medical data	→	Time to survival (y > 0)
Customer click data	→	Profit/loss from customer

Learning to Play Go



Positions on a Go board

AI Systems for Playing Go



https://www.reddit.com/r/baduk/comments/6ttyyz/better_graph_of_go_ai_strength_over_time/

Summary of Types of Machine Learning

- Supervised Learning
  - <x, y> pairs are provided during training
    - Classification (y is categorical)
    - Regression (y is real-valued)
- Unsupervised Learning
  - Only x's provided, no y's
  - Examples: clustering, dimension reduction
- Other variants:
  - Semi-supervised: some target y's, some with only x values
  - Active: algorithm can actively select which x's to get y values for
- Reinforcement Learning
  - No "best answer", just feedback (better/worse); often sequential
- …

Questions?

Machine Learning

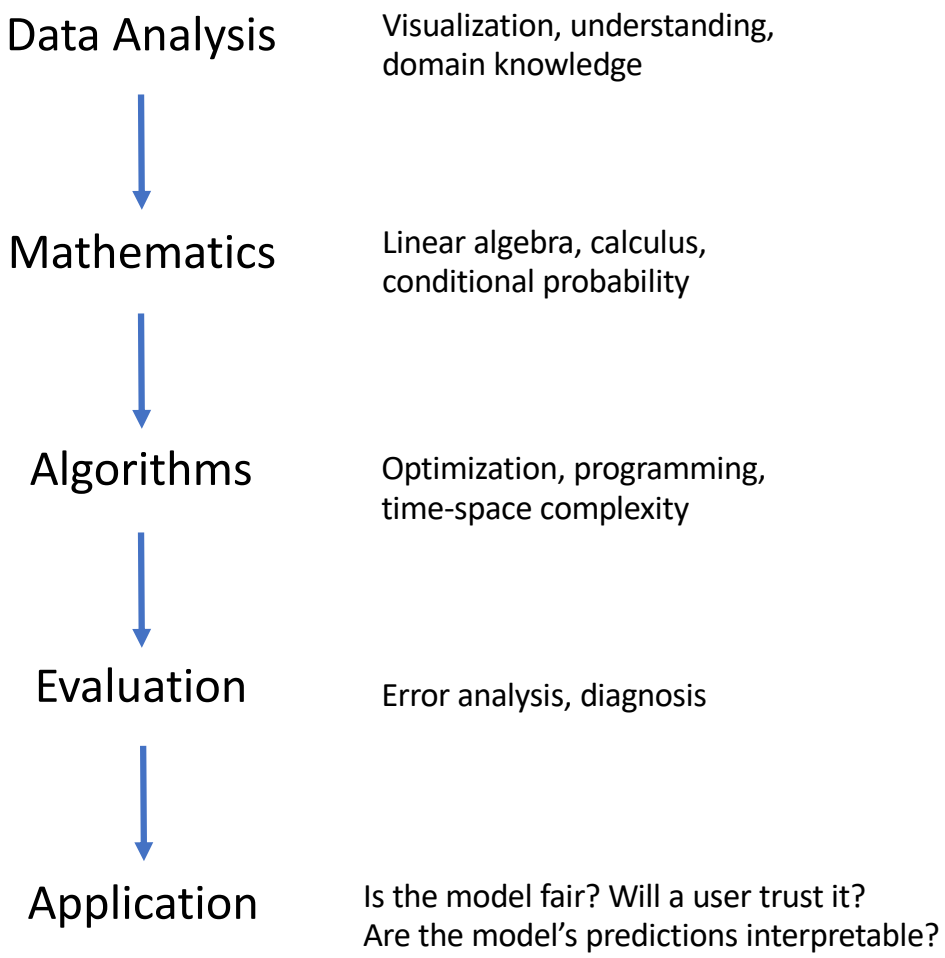
Introduction to Machine Learning

Course Organization

Supervised Learning

Data Exploration

Key Aspects of Machine Learning



Computational Programming Environments

Python

- Numpy, MatPlotLib, SciPy…
- Tensorflow, PyTorch for Deep Learning

Matlab (commercial)

R

- Used mainly in statistics

C/C++

- For fast performance (can be called by Python)

+ other, more specialized languages for visualization and modeling…

Python Libraries for Computation

- NumPy and SciPy
  - NumPy: Basic (fast) array operations, linear algebra
  - SciPy: broader scope of numerical algorithms (e.g., optimization)
- Pandas
  - Useful for data exploration, particularly for tabular data
  - Built on top of NumPy, more high-level data manipulation
  - Often uses DataFrames
- Plotting/visualization
  - Matplotlib, pyplot, seaborn, + others
  - Flexible plotting environments for interactive data analysis
  - Homeworks will use pyplot

Python Libraries for Machine Learning

- Scikit-learn (aka sklearn)
  - The standard machine learning library for Python
  - Will be the basis for most of our homeworks
  - Easy to use, flexible. Does not include deep learning

- Deep learning libraries
  - Tensorflow: broad framework for deep learning
  - PyTorch: More recent alternative to Tensorflow
  - Both broadly used in research and industry

- All are open-source

Data Exploration and Visualization

- Supervised learning: interested in inputs x and outputs y
- Often useful to try to understand basic properties of the x and y data in a machine learning problem
  - e.g.: by visualization; by computing summary statistics
- This is often referred to as "exploratory data analysis"
  - Can help us to better understand how x and y are related
  - Can potentially help detect problems (e.g., missing values)
- …but note that for very large high-dimensional datasets, exploratory data analysis may not always be practical

Fisher "Iris" Dataset

- Historical statistical data set

- Data Summary
  - 150 flowers
  - Each described by 4 real-valued features ("attributes")
    - petal length & width, sepal length & width
  - Each flower is in 1 of 3 classes: *setosa, virginica, versicolor*

  - Natural to put data in a 150 x 4 table
    - 150 rows (flowers) by 4 columns (features)

  - ..with additional 150 x 1 column of class labels

For more information see: http://en.wikipedia.org/wiki/Iris_flower_data_set

Representing data in Python

- Have m observations (data points)

$$\left\{x^{(1)} \ldots, x^{(m)}\right\}$$

superscript in parentheses: data point index

- Each observation is a vector consisting of n features

$$x^{(j)} = [x_1^{(j)} x_2^{(j)} \ldots x_n^{(j)}]$$

subscript: vector index

- Often, represent this as a "data matrix"

$$\underline{X} = \begin{bmatrix} x_1^{(1)} & \ldots & x_n^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(m)} & \ldots & x_n^{(m)} \end{bmatrix}$$

For us: rows are data points; each column is a feature. (Not always standardized! But normal in Python.)

```
import numpy as np    # import numpy
iris = np.genfromtxt("data/iris.txt",delimiter=None)
X = iris[:,0:-1]      # load data and split into features, targets
Y = iris[:,-1]
print X.shape         # 150 data points; 4 features each
   (150, 4)
```

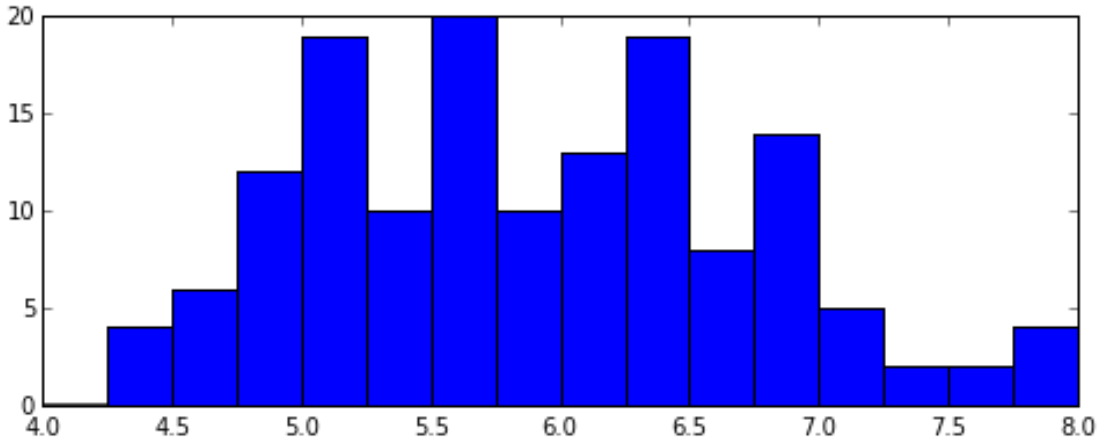
Basic data statistics

- Look at basic information about features
  - Average value? (mean, median, etc.)
  - "Spread"? (standard deviation, etc.)
  - Maximum / Minimum values?

```
print np.mean(X, axis=0)          # compute mean of each feature
 [ 5.8433   3.0573   3.7580   1.1993 ]
print np.std(X, axis=0)           # compute standard deviation of each feature
 [ 0.8281   0.4359   1.7653   0.7622 ]
print np.max(X, axis=0)           # largest value per feature
 [ 7.9411   4.3632   6.8606   2.5236 ]
print np.min(X, axis=0)           # smallest value per feature
 [ 4.2985   1.9708   1.0331   0.0536 ]
```

Visualizing data: histograms

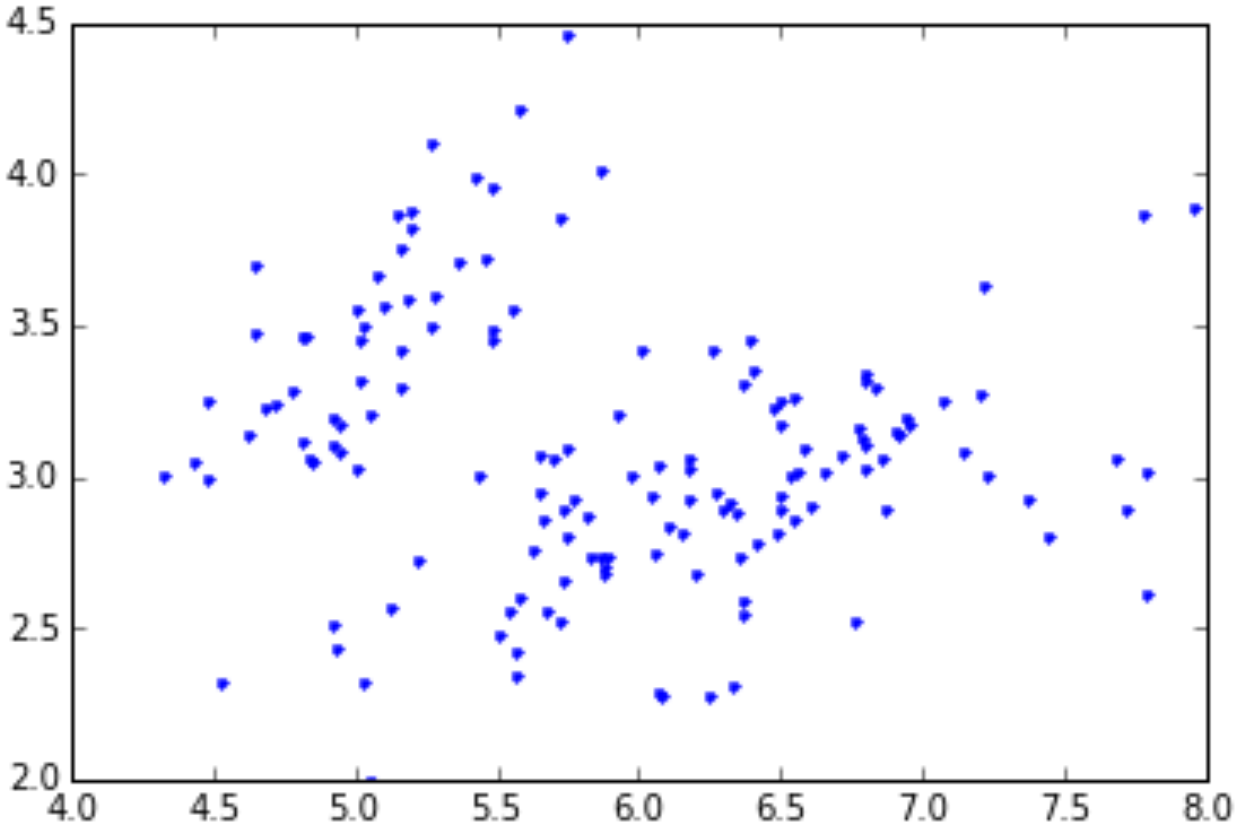
- Count the data falling in each of K bins
  - "Summarize" data as a length-K vector of counts (& plot)
  - Value of K determines "summarization"; depends on # of data
  - K too big: every data point falls in its own bin; just "memorizes"
  - K too small: all data in one or two bins; oversimplifies




```
# Histograms in MatPlotLib
import matplotlib.pyplot as plt
X1 = X[:,0]                 # extract first feature
Bins = np.linspace(4,8,17)  # use explicit bin locations
plt.hist( X1, bins=Bins )   # generate the plot
```

Visualizing data: scatterplots

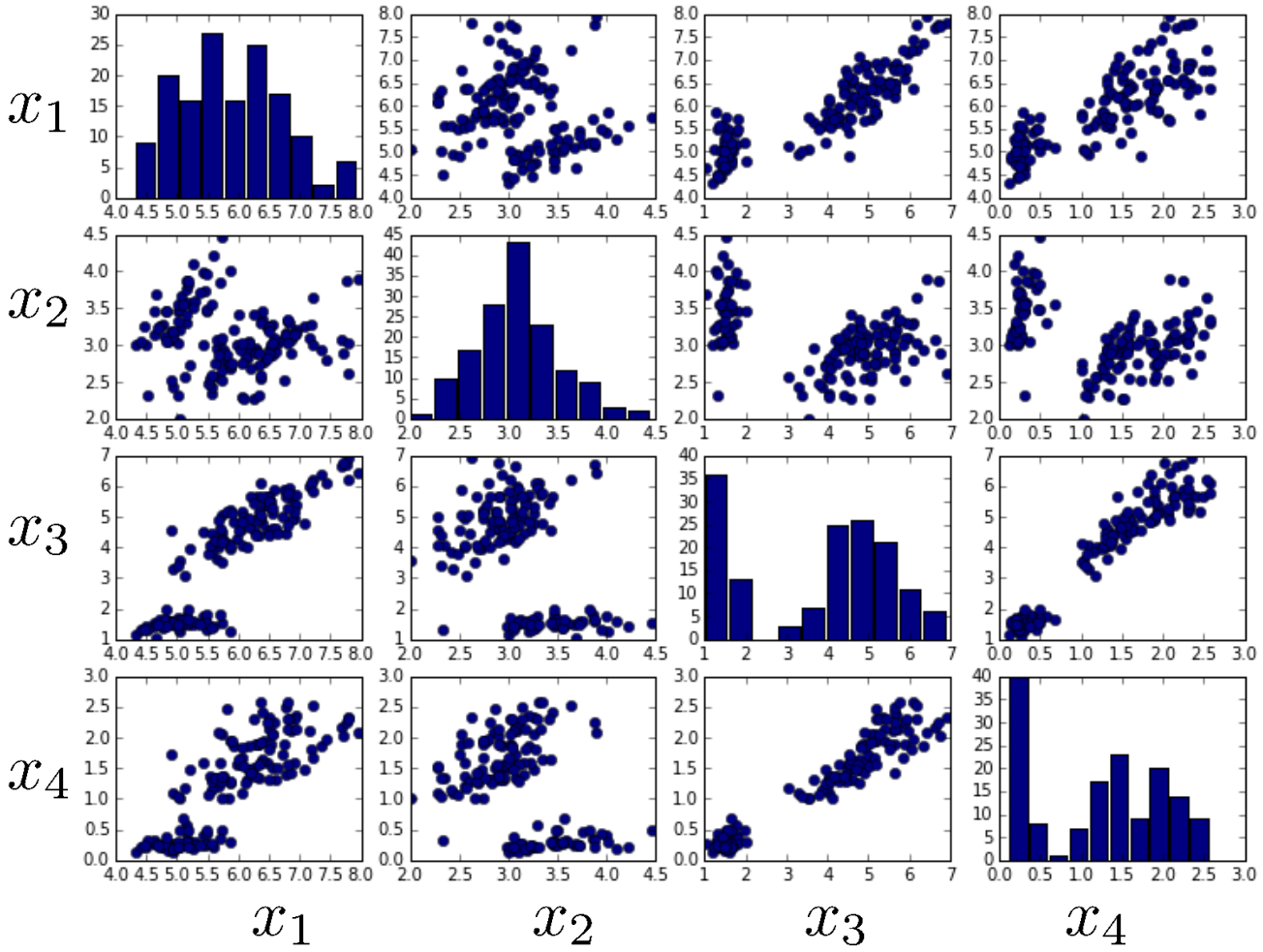
- Illustrate the relationship between two features




```
# Plotting in MatPlotLib
plt.plot(X[:,0], X[:,1], 'b.');   # plot data points as blue dots
plt.scatter(X[:,0], X[:,1], c='b');     # another plot function
```

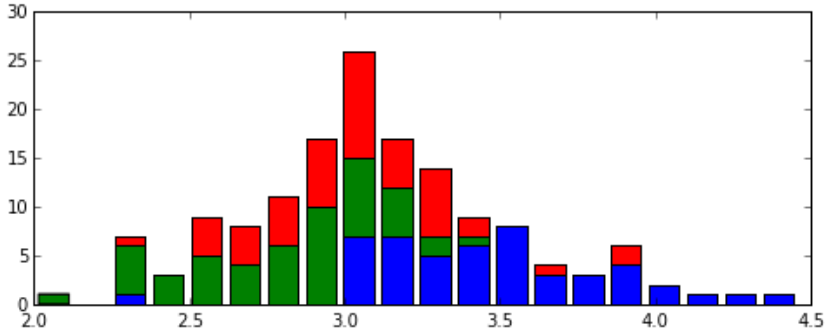
Scatterplots

- For more than two features we can use a “pair plot”:

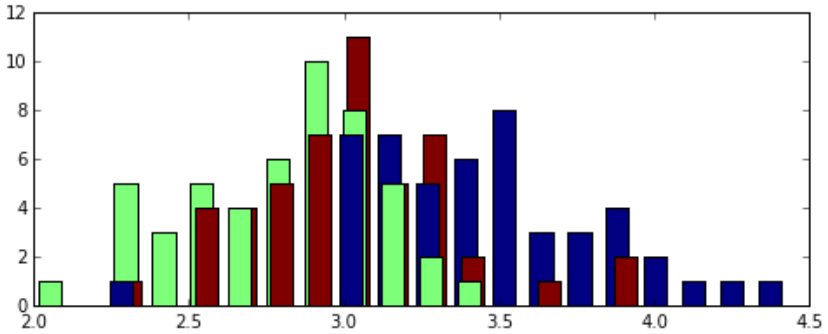


Supervised learning and targets

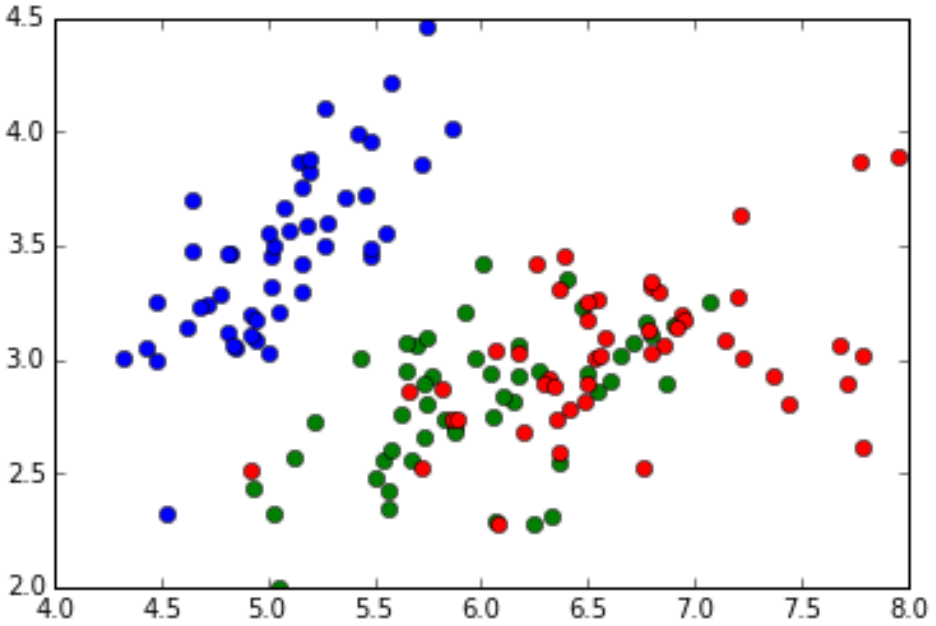
- Supervised learning: predict target values
- For discrete targets, often visualize with color




```
plt.hist( [X[Y==c,1] for c in np.unique(Y)] ,
          bins=20, histtype='barstacked')
```




```
ml.histy(X[:,1], Y, bins=20)
```




```
colors = ['b','g','r']
for c in np.unique(Y):
    plt.plot( X[Y==c,0], X[Y==c,1], 'o',
              color=colors[int(c)] )
```

Wrapup

Next Lecture

- How does ML work?
  - Learner, parameters, loss, training algorithm
  - Visualizing predictors (regression vs classification)
- A simple classifier: nearest centroid
- Decision theory
  - Optimal decisions; Bayes error rate
  - Estimating models from training data
- Evaluating classifier performance

Questions?