

# Course Overview

[Max \(Mahdi\) Roozbahani](#)

Senior Lecturer, College of Computing, SCI

Georgia Tech

Founder of [Filio](#)

# How to succeed in this course?

I will always start the class with “Good Morning” or “Good Afternoon” (of course, if the class starts before noon)

I need a response **full of energy** from **all** students.  
It gives me a good feeling and shows me you are here to learn new exciting things. That way I stay motivated.

If I stay motivated → I do my best in teaching → you learn better  
→ you will get an A

## Let's practice

Refer to:

# Class Website

For anything (updates, lectures, logistics, and so on) related to this class

# CLOS - Students

- “The course was great and helpful as a starter to learn machine learning. The project work is really valuable and can make a great addition to resumes if students put in a good effort and learn. The project requires self-learning quite a few things but I think that's a great way to learn in depth. For me the project was very helpful since lots of my interviewers in Intel asked about it and were interested in it. The cherry on the cake was that I landed the job at Intel.”
- “I went back and watched lectures before I took the weekly quizzes. The homework was tough, and they weren't kidding when they said it would take a lot of time.”
- “Metaphorically speaking, I felt like whenever I left lecture, I knew how to put up a fence in my backyard. but when the time came to do the homeworks, it felt like I was being asked to rebuild the great wall of china.”
- “This class expects a lot of effort and I had to expend a lot of effort. The concepts are very advanced and complicated, so it is important to pay attention to lectures and allocate plenty of time to complete the homework. There is a lot of learning as you go, so time management is key.”
- “The fact that I got to learn so much, complete some intense coding, do a group project and accomplish so much in one class in one semester is amazing.”
- “I think that the class inherently requires a significant amount of time and effort since the material is very complicated. For the complexity of what we are learning, the effort required is appropriate”

# I verbally ask questions in the class

Sometimes I ask questions about previous lectures or something that you have already learnt.

Answer the question even if you think you are wrong (nobody loses point answering the question wrong). It will help you and other students to understand concepts much better.

## Ed (the best source for Q/A)

Ask your questions in ed (make it public to other students), and also please see other questions on Ed, it might answer your question. (Please do not send me or TAs Emails regarding hw questions, exams, and other logistics – you can also ask “private question” on ed) =>Class participation

Bonus points: Undergrad and grad

# Some important notes:

- No Email please, just Ed (Chat and Threads).
- Please read the website carefully. Website will be the first thing that we update if there any changes.
- Add HWs dues and Quizzes to your desired calendar.
- Please come to the class (required); it will help a lot 😊
- ML is **NOT** a programming class.
- Lectures will be Math Heavy and HWs will be mostly programming. ML is all about Linear Algebra, Probability, Statistics and Optimization. You need to have both mathematical and programming skills to be successful in this area.

# Office hours:

Notes:

- 1) Each student may fill in their name and create a BlueJeans Meeting for the office hour
- 2) The signup policy is first come, first serve. **Please, DO NOT override others' sign-ups without their permissions.**
- 3) Because of the time limit, each student will have a maximum time of 10 minutes to work with the TA
- 4) Please come prepared with specific questions so we can help you more effectively

The slots were cleared on Nov 15th. Please add your name again if you had filled it for this week. Apologies for the inconvenience.

Time Slots	Student Name	Question of Interest (Please be more specific about the question you want to ask)	BlueJeans Address	Label
Jayanta's OH				
12:00--12:10				
12:10--12:20				
12:20--12:30				
12:30--12:40				
12:40--12:50				
12:50--13:00				
Waitlist				
	1			
	2			
	3			
Huili's OH				
14:30--14:40				
14:40--14:50				
14:50--15:00				
15:00--15:10				
15:10--15:20				
15:20--15:30				
Waitlist				
	1			
	2			

- You have just 10 minutes for each slot.
- Please mindful of other students (you can't block multiple office hours, just one and we have the waitlist!
- Make sure you pinpoint your problem exactly before joining the office hour.
- Office hour is not for general code debugging. You need to be specific about your question.

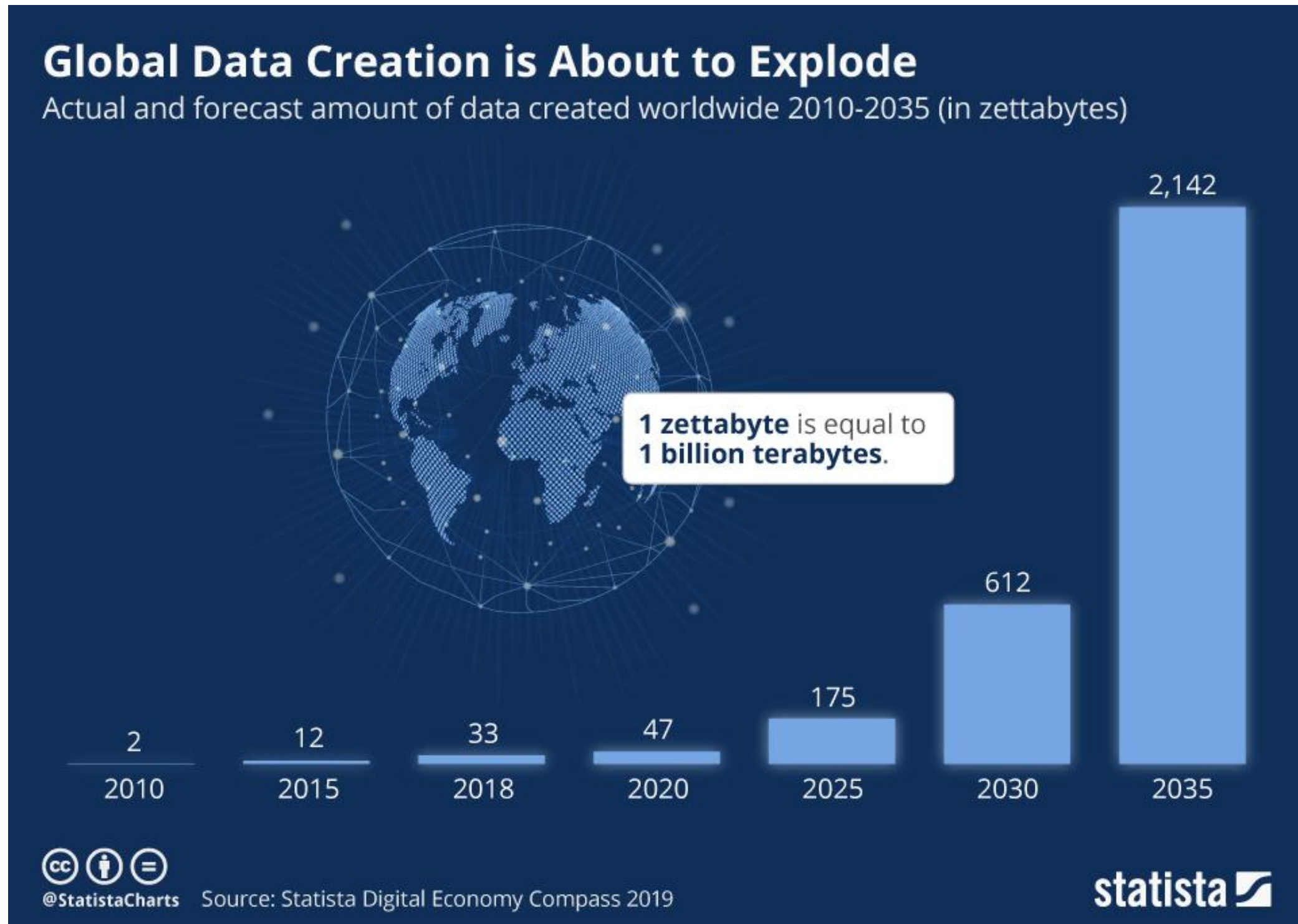
# Honest statements:

- As I mentioned, the class is math-heavy, and at times, it can get boring. That is why we all need to work together to make the class interactive and exciting. It might be tough to see the immediate benefits in the classroom but trust me, once you step into your career (or dive into research), you'll truly appreciate the skills you have earned here.
- I say many times that you need to start working on the HWs as soon as they are out, and sometimes I also say if you have not started them yet, you can't finish them. I know it may create pressure on our students to say that, but my intention is simply to help you avoid the immense stress of rushing at the last minute or, worse, not being able to submit your work on time. It's all about pacing ourselves and managing our time wisely, right?
- I take intermittent\ frequent attendance. I KNOW it could be annoying, particularly for our senior undergrad and grad students. I want you to know that this comes from a place of genuine care - our team deeply wants each of you to excel in machine learning. We believe in every single one of you and your potential to master this field.
- I know that Machine Learning isn't the only course on your plate this semester, and this class comes with major requirements and deliverables. It is not about us wanting to hog all your attention; it is just that there is so much we want to share in this class. We would hate for you to miss out on any of it, that is all.
- Please **DON'T** forget the statements above as we progress throughout the semester on why we do such things in our class. Let's keep learning together, shall we?



# Machine Learning

***“We are drowning in information but starved for knowledge.” — John Naisbitt***



33 zettabytes is equal to the storage capacity of 33 million human brains altogether

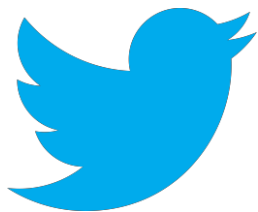
## Levels of AGI

Performance (rows) x Generality (columns)	Narrow <i>clearly scoped task or set of tasks</i>	General <i>wide range of non-physical tasks, including metacognitive abilities like learning new skills</i>
<b>Level 0: No AI</b>	<b>Narrow Non-AI</b> calculator software; compiler	<b>General Non-AI</b> human-in-the-loop computing, e.g., Amazon Mechanical Turk
<b>Level 1: Emerging</b> <i>equal to or somewhat better than an unskilled human</i>	<b>Emerging Narrow AI</b> GOF <sup>AI</sup> <sup>4</sup> ; simple rule-based sys- tems, e.g., SHRDLU (Winograd, 1971)	<b>Emerging AGI</b> ChatGPT (OpenAI, 2023), Bard (Anil et al., 2023), Llama 2 (Touvron et al., 2023)
<b>Level 2: Competent</b> <i>at least 50th percentile of skilled adults</i>	<b>Competent Narrow AI</b> toxicity detectors such as Jig- saw (Das et al., 2022); Smart Speakers such as Siri (Apple), Alexa (Amazon), or Google As- sistant (Google); VQA systems such as PaLI (Chen et al., 2023); Watson (IBM); SOTA LLMs for a subset of tasks (e.g., short essay writing, simple coding)	<b>Competent AGI</b> not yet achieved
<b>Level 3: Expert</b> <i>at least 90th percentile of skilled adults</i>	<b>Expert Narrow AI</b> spelling & grammar checkers such as Grammarly (Gram- marly, 2023); generative im- age models such as Imagen (Sa- haria et al., 2022) or Dall-E 2 (Ramesh et al., 2022)	<b>Expert AGI</b> not yet achieved
<b>Level 4: Virtuoso</b> <i>at least 99th percentile of skilled adults</i>	<b>Virtuoso Narrow AI</b> Deep Blue (Campbell et al., 2002), AlphaGo (Silver et al., 2016, 2017)	<b>Virtuoso AGI</b> not yet achieved
<b>Level 5: Superhuman</b> <i>outperforms 100% of humans</i>	<b>Superhuman Narrow AI</b> AlphaFold (Jumper et al., 2021; Varadi et al., 2021), AlphaZero (Silver et al., 2018), StockFish (Stockfish, 2023)	<b>Artificial Superintelligence (ASI)</b> not yet achieved

# The Booming Age of Data



30 trillion Web pages



500 million tweets per day



2.27 billion monthly active users



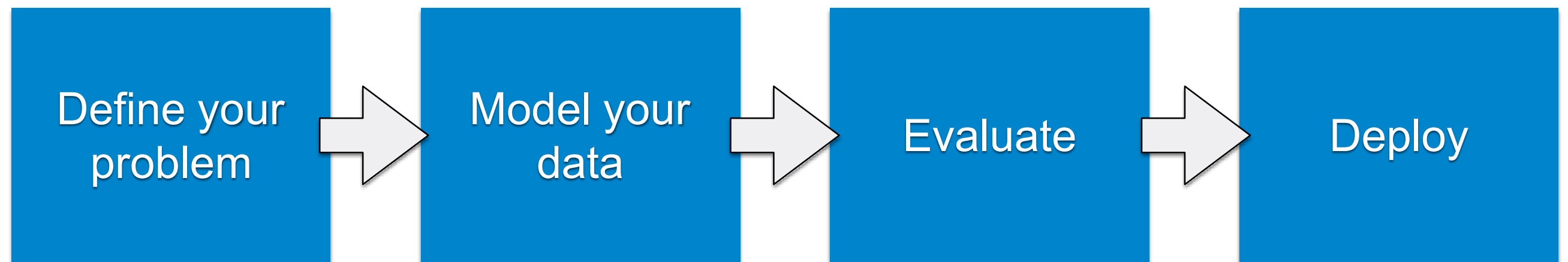
1.8 billion images uploaded to Internet per day



2.9 billion base pairs in human genome

# Machine Learning

Machine Learning is the process of **turning data into actionable knowledge** for **task support** and **decision making**.



# Course Objectives

- Introduce to you the **pipeline of Machine Learning**
- Help you understand **major machine learning algorithms**
- Help you learn to **apply tools** for **real data analysis problems**
- Encourage you to **do research** in data science and machine learning

# Brief History of Machine Learning

## 1950s

**Samuel's checker player**

Selfridge's Pandemonium

## 1960s:

**Neural networks: Perceptron**

Pattern recognition

Learning in the limit theory

Minsky and Papert prove limitations of Perceptron

## 1970s:

Symbolic concept induction

Winston's arch learner

Expert systems and the knowledge acquisition bottleneck

Quinlan's ID3

Michalski's AQ and soybean diagnosis

Scientific discovery with BACON

**Mathematical discovery with AM (Automated Mathematician)**



# Brief History of Machine Learning

## 1980s:

### **Advanced decision tree and rule learning**

Explanation-based Learning (EBL)

Learning and planning and problem solving

Utility problem

Analogy

Cognitive architectures

Resurgence of neural networks (connectionism, backpropagation)

Valiant's PAC Learning Theory

Focus on experimental methodology

## 1990s

### **Data mining**

Adaptive software agents and web applications

Text learning

Reinforcement learning (RL)

Inductive Logic Programming (ILP)

### **Ensembles: Bagging, Boosting, and Stacking**

Bayes Net learning

# Brief History of Machine Learning

## 2000s:

**Support vector machines**

**Kernel methods**

Graphical models

Statistical relational learning

Transfer learning

Sequence labeling

Collective classification and structured outputs

Computer Systems Applications

Learning in robotics and vision

## 2010s:

**Deep learning**

**Reinforcement learning**

**Generative models**

Adversarial learning

Muti-task learning

**Learning in NLP, CV, Robotics, ...**

AI Winter 1: 1974 to 1980 over-promising and under-delivering (could not do even simple tasks of machine translation or general-purpose robotics)

AI Winter 2: 1987-2000 Moving from expensive Lisp machines to UNIX workstations



# Syllabus

## **Part I: Basic math** for computational data analysis

- Probability, statistics, linear algebra

## **Part II: Unsupervised learning** for data exploration

- Clustering analysis, dimensionality reduction, kernel density estimation

## **Part III: Supervised learning** for predictive analysis

- Tree-based models, linear classification/regression, neural networks

# Unsupervised and Supervised learning

	Weight(lb)	Height(cm)	Fur color	Eye color	Label
Point 1	10	20	<i>w</i>	<i>g</i>	<i>cat</i>
Point 2	50	100	<i>br</i>	<i>bl</i>	<i>dog</i>
Point 3	8	15	<i>bl</i>	<i>bl</i>	<i>dog</i>
Point 4	12	25	<i>w</i>	<i>bl</i>	<i>cat</i>
Point 5	14	10	<i>bl</i>	<i>g</i>	<i>dog</i>
$X_{n \times d}$					$= Y_{n \times 1}$

Unsupervised just focuses on  $X_{n \times d}$

Supervised focus on  $X_{n \times d}$  and  $Y_{n \times 1}$

# We can do better than Cat and Dog

	Weight(lb)	Height(cm)	Fur color	Eye color	Label
Point 1	10	20	$w$	$g$	Blob Fish
Point 2	50	100	$br$	$bl$	opossum
Point 3	8	15	$bl$	$bl$	opossum
Point 4	12	25	$w$	$bl$	Blob Fish
Point 5	14	10	$bl$	$g$	opossum
$X_{n \times d}$					$= Y_{n \times 1}$



# Syllabus: Unsupervised Learning

## Clustering Analysis

- K-means

- Gaussian mixture model

- Hierarchical clustering

- Density-based clustering

- Evaluation of clustering algorithms

## Dimension Reduction

- Principal component analysis

## Kernel Density Estimation

- Parametric density estimation

- Non-parametric density estimation



# Community Detection in Social Networks

What are the inputs  
and how to represent  
them?

What are the desired  
outputs?

What learning  
algorithms to choose?



# Examples on our class website

- **Sample Projects**

- Sample Project from previous semester [[Undergrad Canvas Access for previous ML projects](#)]; [[Grad Canvas Access for previous ML projects](#)]; [Stanford Project Examples](#);

- **General project guidance**

- Your project will be graded based on the following criteria:

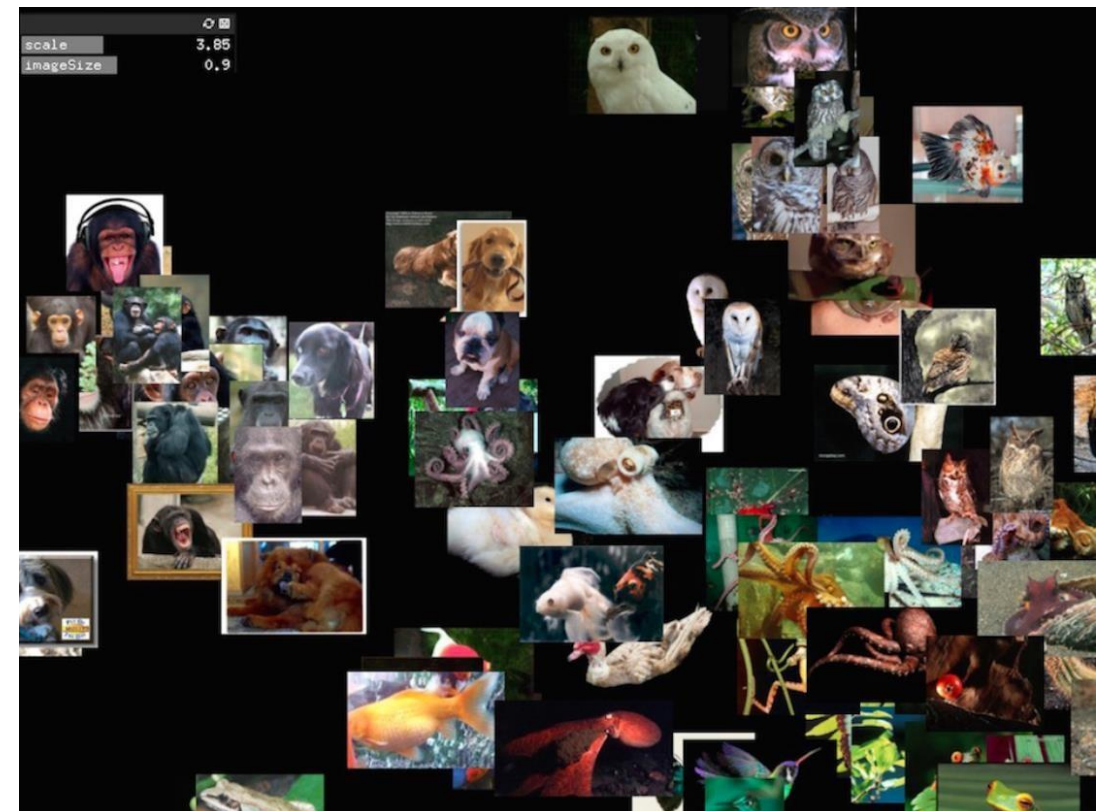
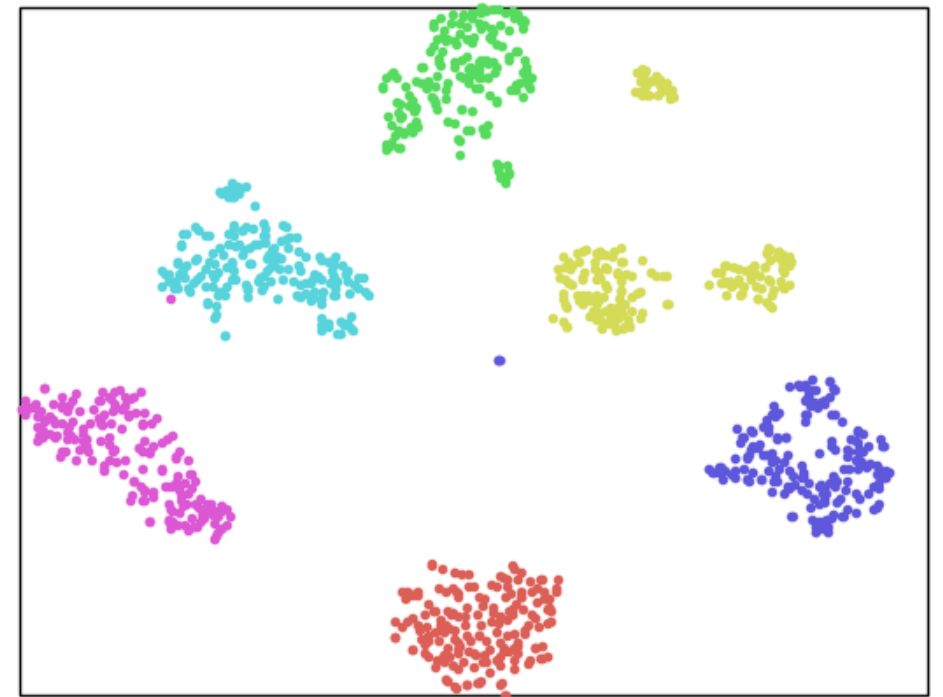
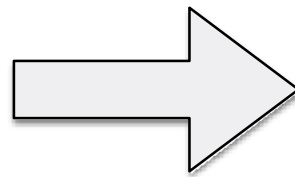
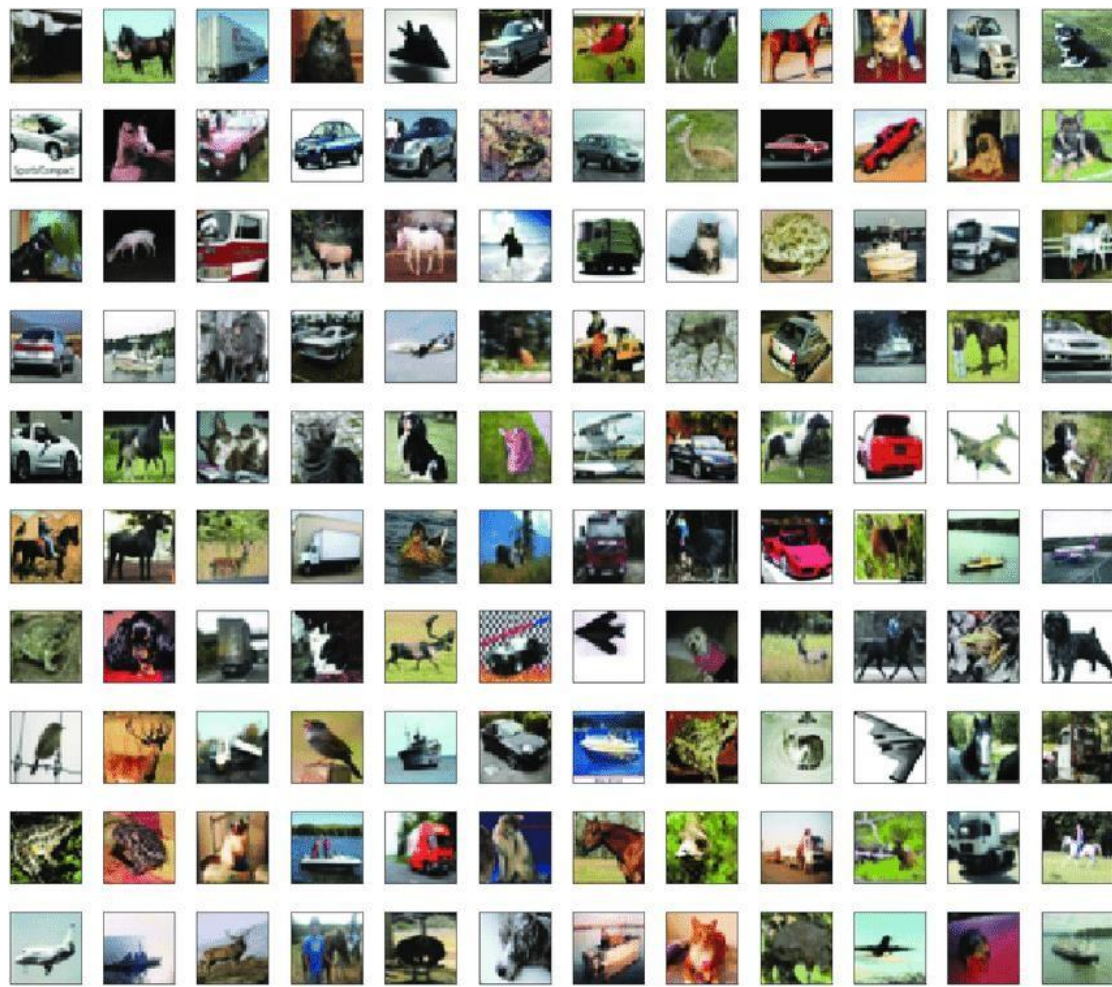
**Was the motivation clear?**

- What is the problem?
- Why is it important and why we should care?

**Were the dataset and approach used effectively?**



# Dimensionality Reduction



What are the inputs and how to represent them?

What are the desired outputs?

What learning algorithms to choose?

# Syllabus: Supervised Learning

## Tree-based models

- Decision tree

- Ensemble learning/Random forest

## Linear classification/regression models

- Linear regression

- Naive Bayes

- Logistic regression

- Support vector machine

## Neural networks

- Feedforward neural networks and backpropagation analysis
  - CNN



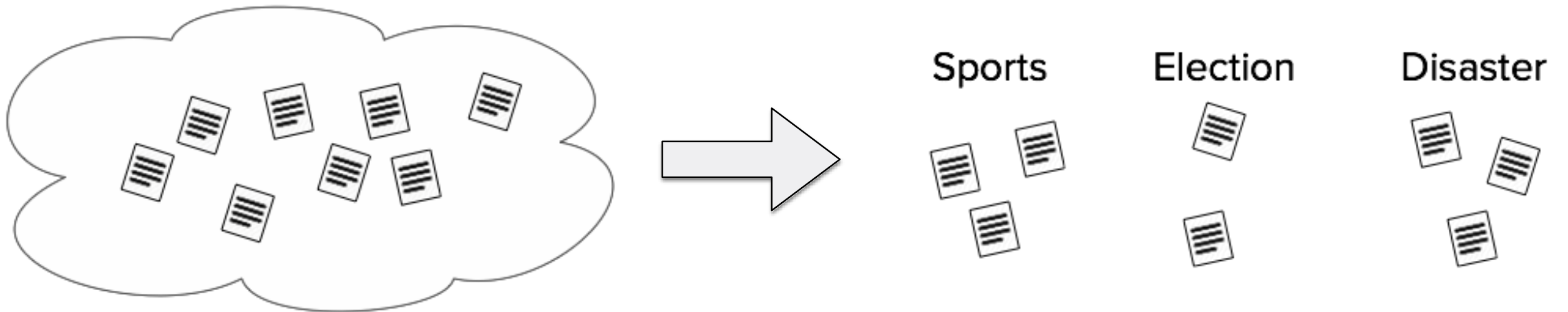
# News Classification



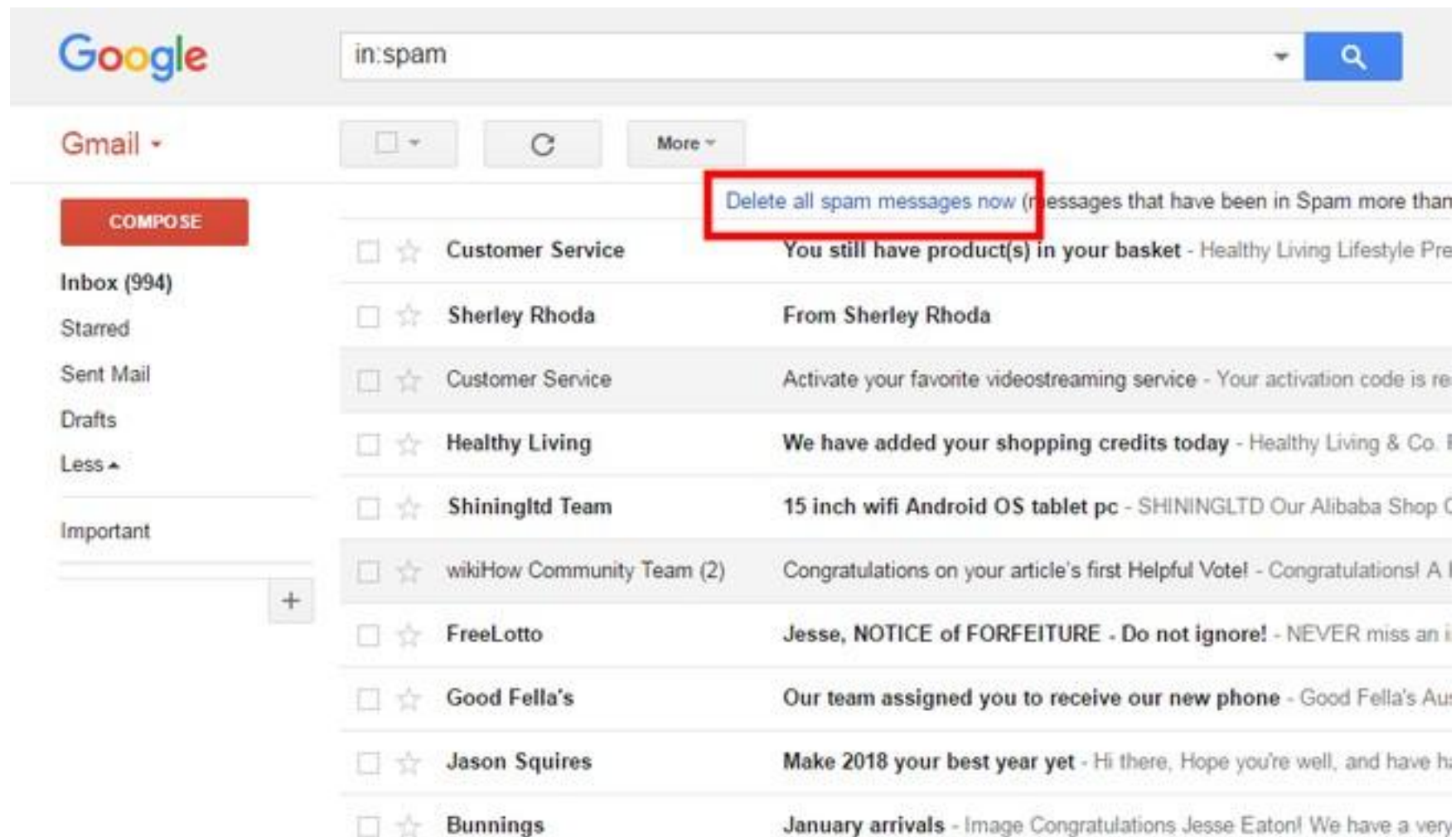
What are the inputs and how to represent them?

What are the desired outputs?

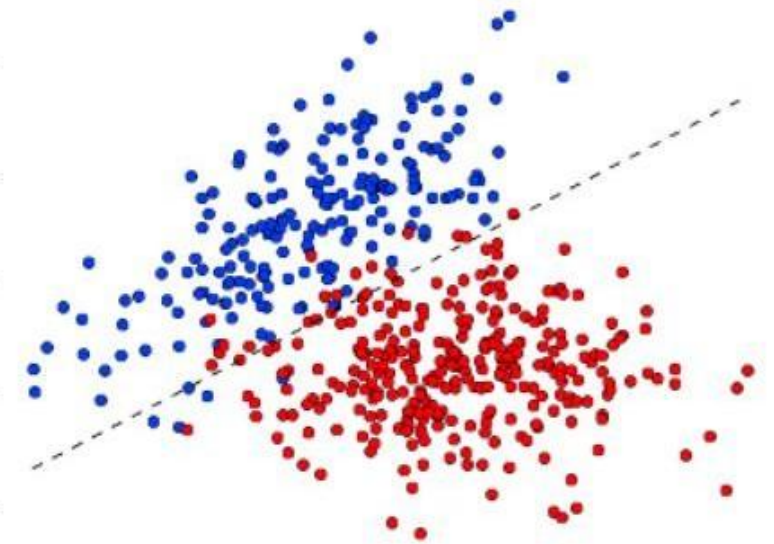
What learning algorithms to choose?



# Spam Detection



**NOT SPAM**



**SPAM**

What are the inputs and how to represent them?

What are the desired outputs?

What learning algorithms to choose?

# Examples on our class website

- **Sample Projects**

- Sample Project from previous semester [[Undergrad Canvas Access for previous ML projects](#)]; [[Grad Canvas Access for previous ML projects](#)]; [Stanford Project Examples](#);

- **General project guidance**

- Your project will be graded based on the following criteria:

**Was the motivation clear?**

- What is the problem?
- Why is it important and why we should care?

**Were the dataset and approach used effectively?**

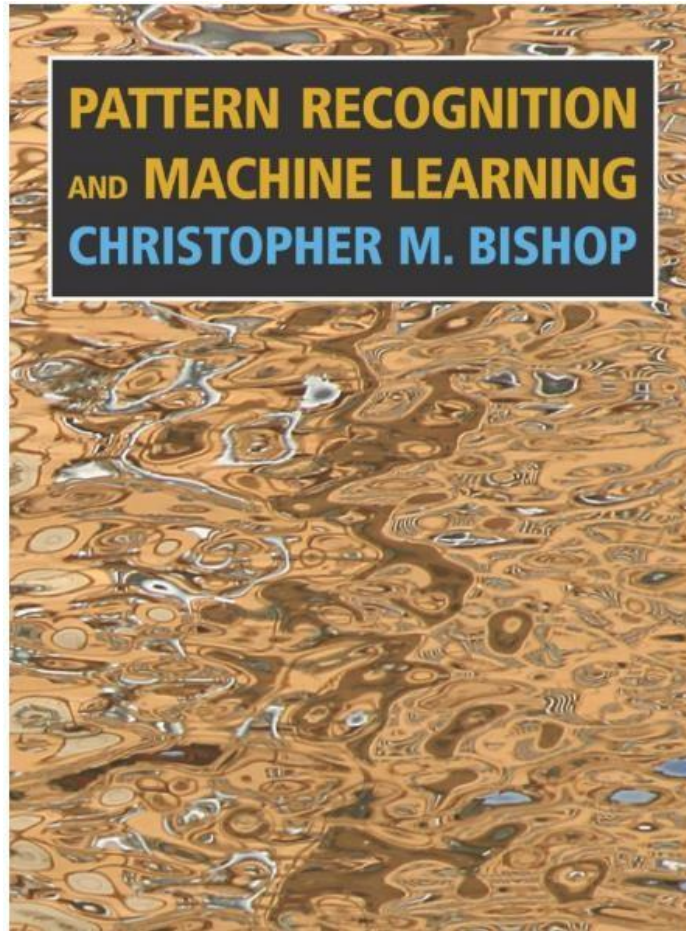
# Prerequisites

Basic knowledge in probability, statistics, and linear algebra

Basic programming skills in Python (Jupyter Notebook)

*No background in machine learning is required*

# Text Books



## [Pattern Recognition and Machine Learning](#), by Chris Bishop

Other recommended books:

[Learning from data](#), by Yaser S. Abu-Mostafa

[Machine learning](#), by Tom Mitchell

[Deep Learning](#), by Ian Goodfellow, Yoshua Bengio, and Aaron Courville

# Assignments

Four assignments (GradeScope)

Each can include written analysis or programming

**Start Early as soon as they are out**

**Read late policy on the class website**

**Don't copy**

Because of the large size of our class, if we observe any (even small) similarity\plagiarisms detected by GradeScope or our TAs, WE WILL DIRECTLY REPORT ALL CASES TO OSI, which may unfortunately lead to a very harsh outcome.



# Projects

- Work on a real-life Machine Learning problem
  - What is the problem? What is your method? How do you evaluate it?
- Exactly 5 people in a team (Grad and undergrad can't be mixed in a group)
- GitHub Pages (index.html)
- Start your projects early
- Ask for comments and feedbacks from the teaching staff
- You need to have a group meeting with your team at least once in a week (either virtual or in-person). Otherwise, team conflicts will most probably happen.