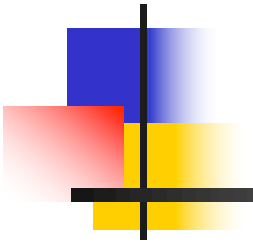
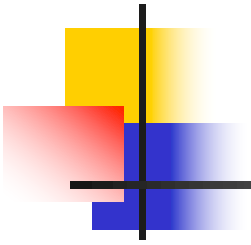


CSCI 5090/7090: Machine Learning

Spring 2018

Mehdi Allahyari
Georgia Southern University

(some slides from Pedro Domingos, University of Washington)



Mehdi Allahyari

- Call me Dr. Yari
- PhD from UGA
- Machine Learning, Semantic Web, Text/Data mining, Information Retrieval

Course Information

CSCI 5090/7090 - Machine Learning

Spring 2018

Instructor: Dr. Mehdi Allahyari <mallahyari@georgiasouthern.edu>
Office: CEIT 2321
Course Webpage: https://sci2lab.github.io/mehdi/teaching/x90
Piazza: http://piazza.com/georgiasouthern/spring2018/cscix090

Lecture Times

	Monday	Wednesday
CRN 18045/11783	09:05 AM – 10:20 AM CEIT 2212	09:05 AM – 10:20 AM CEIT 2212

Office Hours

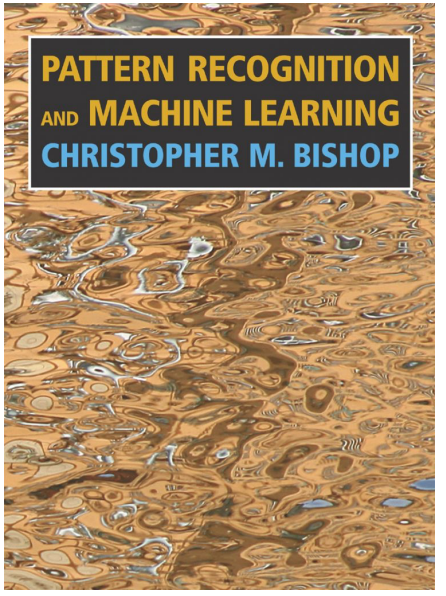
Monday	Wednesday
11:00 AM – 12:00 PM	11:00 AM – 12:00 PM

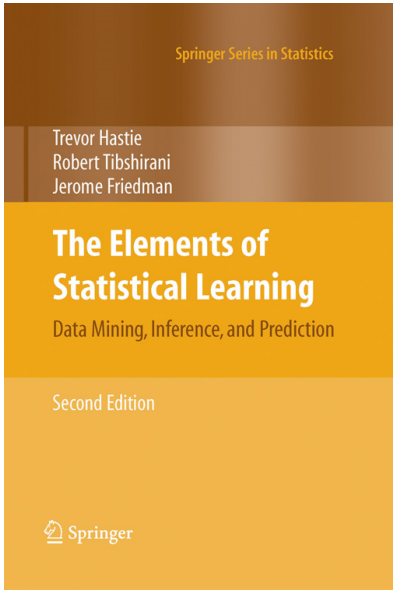
Tuesday	Thursday
03:30 PM – 04:30 PM	03:30 PM – 04:30 PM
CEIT 2321	CEIT 2321

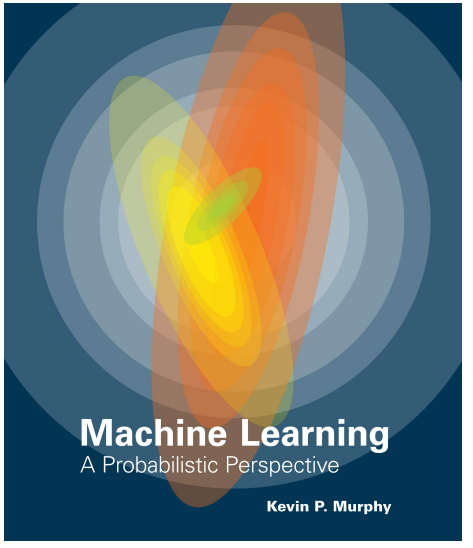
or by appointment

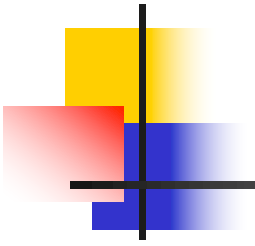
Text Books


MACHINE
LEARNING
TOM M. MITCHELL


PATTERN RECOGNITION
AND MACHINE LEARNING
CHRISTOPHER M. BISHOP


Springer Series in Statistics
Trevor Hastie
Robert Tibshirani
Jerome Friedman
The Elements of
Statistical Learning
Data Mining, Inference, and Prediction
Second Edition
Springer


Machine Learning
A Probabilistic Perspective
Kevin P. Murphy

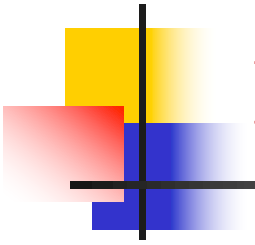


Resources

- Course page: https://sci2lab.github.io/mehdi/teaching/x90/
  - is where you get course content (syllabus, lectures, etc.)
- Folio:
  - Folio is the website where you will submit all assignments (if online), see your grades
- Piazza:
  - An online Q/A board
  - Ask questions here before emailing
  - Answering questions **could** = extra credit!
    - Don't post any code….

Course Description

- This introductory course is designed:
  - to give upper level undergraduate and graduate students a broad overview of many concepts and algorithms in ML.
  - covers the theory and practical algorithms for machine learning from a variety of perspectives.
- Topics:
  - Statistical and probabilistic methods, generative and discriminative models, linear and logistic regression, Bayesian networks, decision tree learning, unsupervised learning and clustering and dimensionality reduction.
  - In addition, the course covers fundamental concepts such as training, validation, overfitting, and error rates



Prerequisites

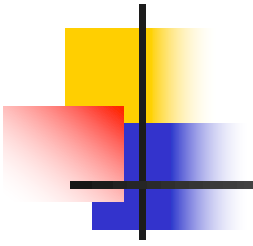
- Basics knowledge of probability, linear algebra and calculus.
  - For example, standard probability distributions and also how to calculate derivatives.

- Significance programming experience
  - **Python** is the language we'll use

Grading

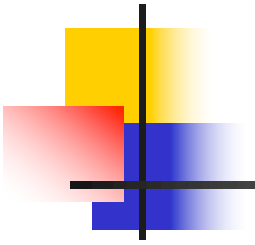
- Attendance and participation: 5%
- Homeworks: 30%
  - Four homeworks: submitted hardcopy *and* on Folio
  - Some exercises, significant programming
- Midterm 03/07/2017
- Final exam o5/02/2018
- Final project:
  - Teams of 2
  - Define a task, create/acquire data for the task, train ML algorithm(s), evaluate & report
  - ***Check the course website for details***

$A \geq 85$	$85 > B \geq 80$	$80 > C \geq 70$	$70 > D \geq 60$	$F < 60$



Homeworks Deadlines

- Assignments are usually due every 2 weeks (see course schedule for deadlines)
- Late submissions for labs/projects are penalized
  - 0-24 hours late -10 points
  - 24-48 hours late -20 points
  - >48 hours late 0
- Forgetting to submit is not a valid excuse
  - Double check your submission
  - Save your confirmation email



Collaboration Policy & Academic Honesty

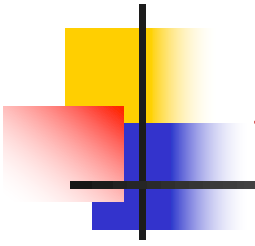
- Homeworks to be done individually
  - Your own answer and code
- can collaborate in figuring out answers and helping each other solve the problems at ***high level***
- No sharing of pseudocode or code or simulation results
- Academic Honesty:
  - abide by the University's academic honesty policy and the Student Honor Code
  - Lack of knowledge of the academic honesty policy is not a reasonable explanation for a violation

Tentative Schedule

Week	Topic	Note
Jan 08	Course overview	
Jan 15	Decision Trees	
Jan 22	Review of probability	
Jan 29	Probability and Estimation	
Feb 05	Naive Bayes	
Feb 12	Gaussian Naive Bayes	
Feb 19	Linear Regression	
Feb 26	Logistic Regression	
Mar 05	Review and Midterm	
Mar 12	Spring Break	No Classes
Mar 19	Ensemble Methods	
Mar 26	Graphical Models	
Apr 02	Hidden Markov Models	
Apr 09	Clustering	
Apr 16	Dimensinality Reduction	
Apr 23	Neural Networks	

Syllabus Questions?

- Familiarize yourself with the syllabus
- This PowerPoint highlights key information from the syllabus but does not cover every subject in detail
  - You are responsible for the entirety of the syllabus
- Your HW is to go home & read the syllabus thoroughly

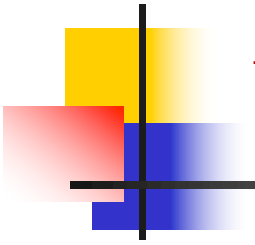


A Few Quotes

- “A breakthrough in machine learning would be worth ten Microsofts” (Bill Gates, Founder, Microsoft)
- “Machine learning is the next Internet” (Tony Tether, Director, DARPA)
- Machine learning is the hot new thing” (John Hennessy, President, Stanford)
- “Machine learning is Google’s top priority” (Eric Schmidt, Chairman, Alphabet)
- “Machine learning is Microsoft Research’s largest investment area” (Peter Lee, Head, Microsoft Research)
- “‘Data scientist’ is the hottest job title in Silicon Valley” (Tim O’Reilly, Founder, O’Reilly Media)

What is Machine Learning?

- How can we solve a specific problem?
- As computer scientists we write a program that encodes a set of rules that are useful to solve the problem
- In many cases is very difficult to specify those rules, e.g., given a picture determine whether there is a cat in the image



What is Machine Learning?

- Learning systems are not directly programmed to solve a problem, instead develop own program based on:
  - Examples of how they should behave
  - From trial-and-error experience trying to solve the problem

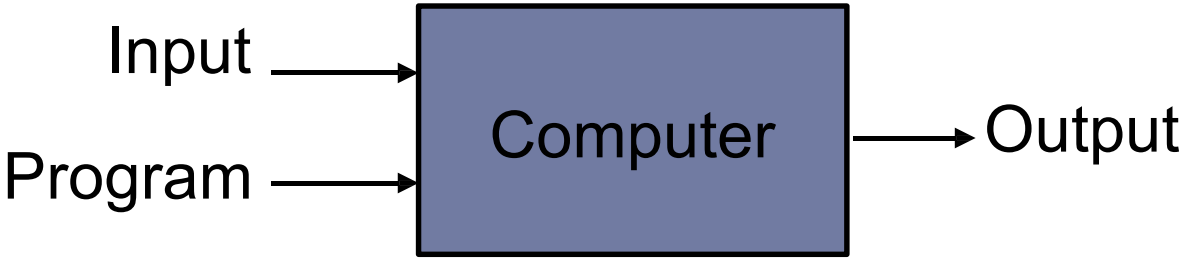
- Different than standard CS
- Want to implement unknown function, only have access e.g., to sample input-output pairs (training examples)
- Learning simply means incorporating information from the training examples into the system

What is Machine Learning?

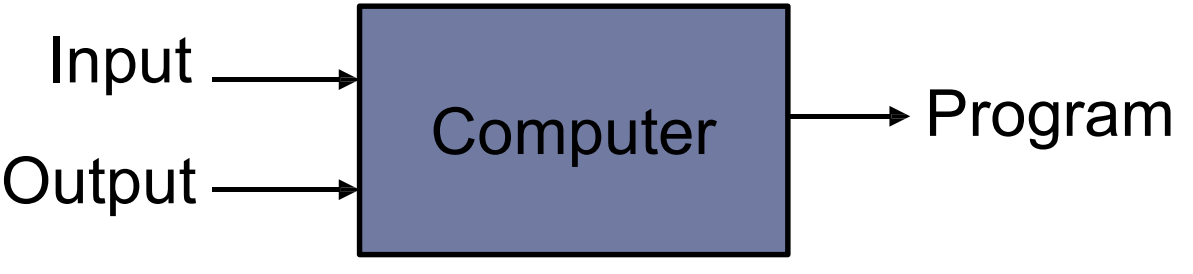
- "The study of computer programs that improve automatically with experience"
  - T. Mitchell *Machine Learning*

- Getting computers to program themselves
- Writing software is the bottleneck, Let the data do the work instead!

Traditional Programming



Machine Learning



Magic?

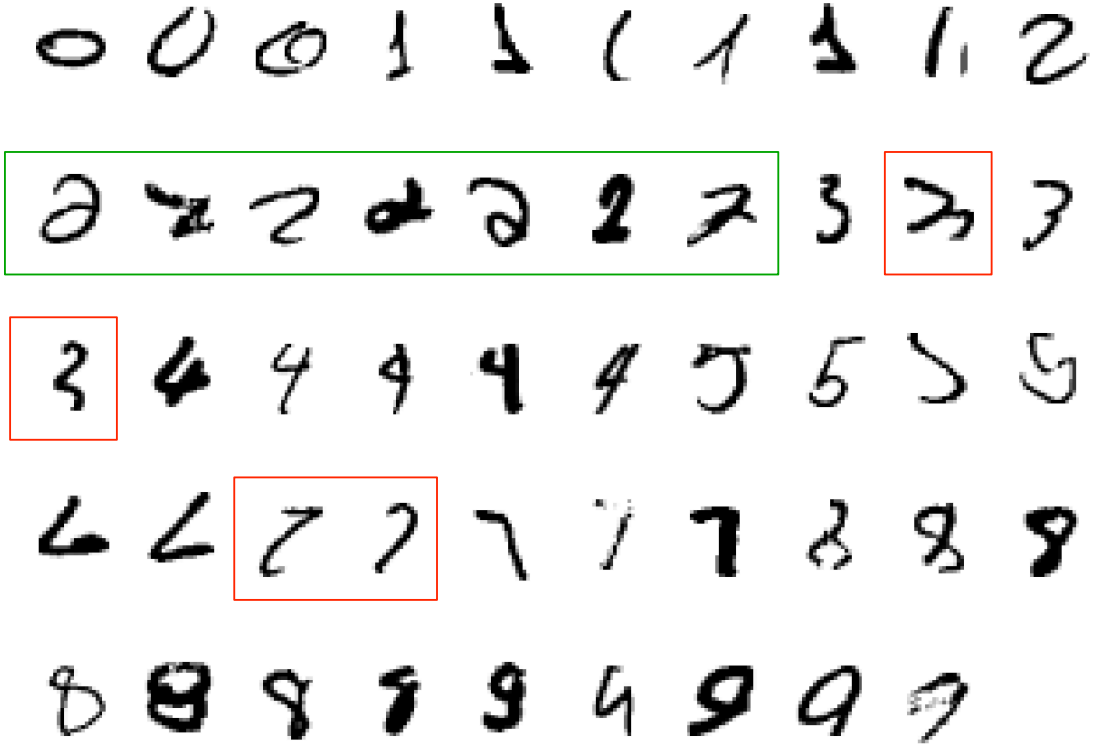
No, more like gardening

- **Seeds** = Algorithms
- **Nutrients** = Data
- **Gardener** = You
- **Plants** = Programs

Sample Applications

- Web search
- Computational biology
- Finance
- E-commerce
- Space exploration
- Robotics
- Information extraction
- Social networks
- Debugging
- [Your favorite area]

Tasks that requires machine learning: What makes a 2?



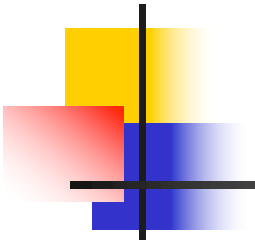
Why use machine learning?

- It is very hard to write programs that solve problems like recognizing a handwritten digit
  - What distinguishes a 2 from a 7?
  - How does our brain do it?

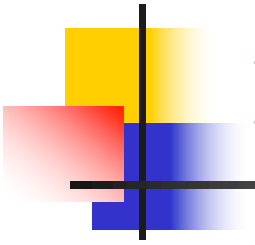
- Instead of writing a program by hand, we collect examples that specify the correct output for a given input

- A machine learning algorithm then takes these examples and produces a program that does the job
  - The program produced by the learning algorithm may look very different from a typical hand-written program. It may contain millions of numbers.
  - If we do it right, the program works for new cases as well as the ones we trained it on.

Relationship of Machine Learning to…

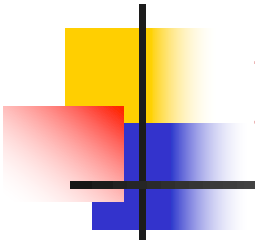


- Statistics
- Analytics / Data Science
- Data Mining
- Artificial Intelligence



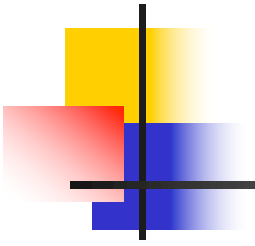
ML in Practice

- Understanding domain, prior knowledge, and goals
- Data integration, selection, cleaning
- pre-processing, etc.
- Learning models
- Interpreting results
- Consolidating and deploying discovered knowledge
- Loop



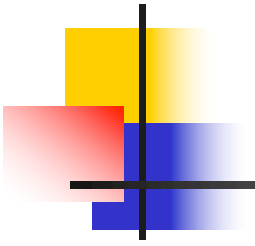
ML in Nutshell

- Tens of thousands of machine learning algorithms
- Hundreds new every year
- Every machine learning algorithm has three components:
  - **Representation**
  - **Evaluation**
  - **Optimization**



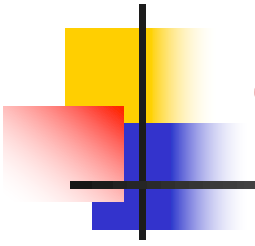
Representation

- *How do we represent the function from input to output?*
- Decision trees
- Sets of rules / Logic programs
- Graphical models (Bayes/Markov nets)
- Neural networks
- Support vector machines
- Model ensembles
- Etc.



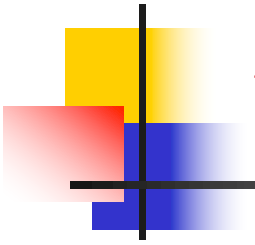
Evaluation

- *Given some data, how can we tell if a function is "good"?*
- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence
- Etc.



Optimization

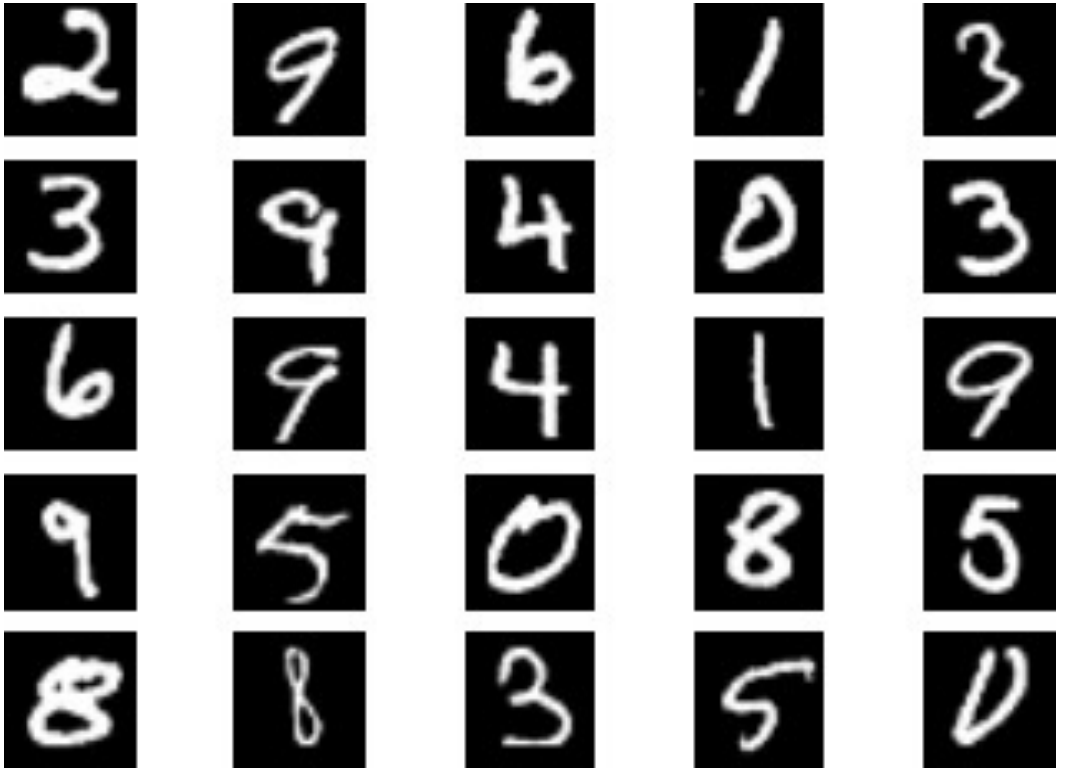
- *Given some data, how do we **find** the "best" function?*
- Combinatorial optimization
  - E.g.: Greedy search
- Convex optimization
  - E.g.: Gradient descent
- Constrained optimization
  - E.g.: Linear programming



Learning algorithms are useful in many tasks

1. Classification: Determine which discrete category the example is

Examples of Classification



What digit is this?

Examples of Classification

Is this a dog?

Examples of Classification

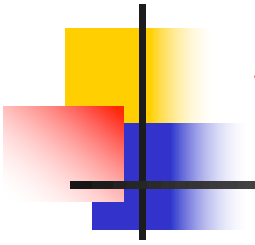
What about this one?

Examples of Classification

Am I going to pass the exam?

Examples of Classification

Do I have diabetes?



Learning algorithms are useful in many tasks

1. Classification: Determine which discrete category the example is
2. Recognizing patterns: Speech Recognition, facial identity, etc

Examples of Recognizing patterns

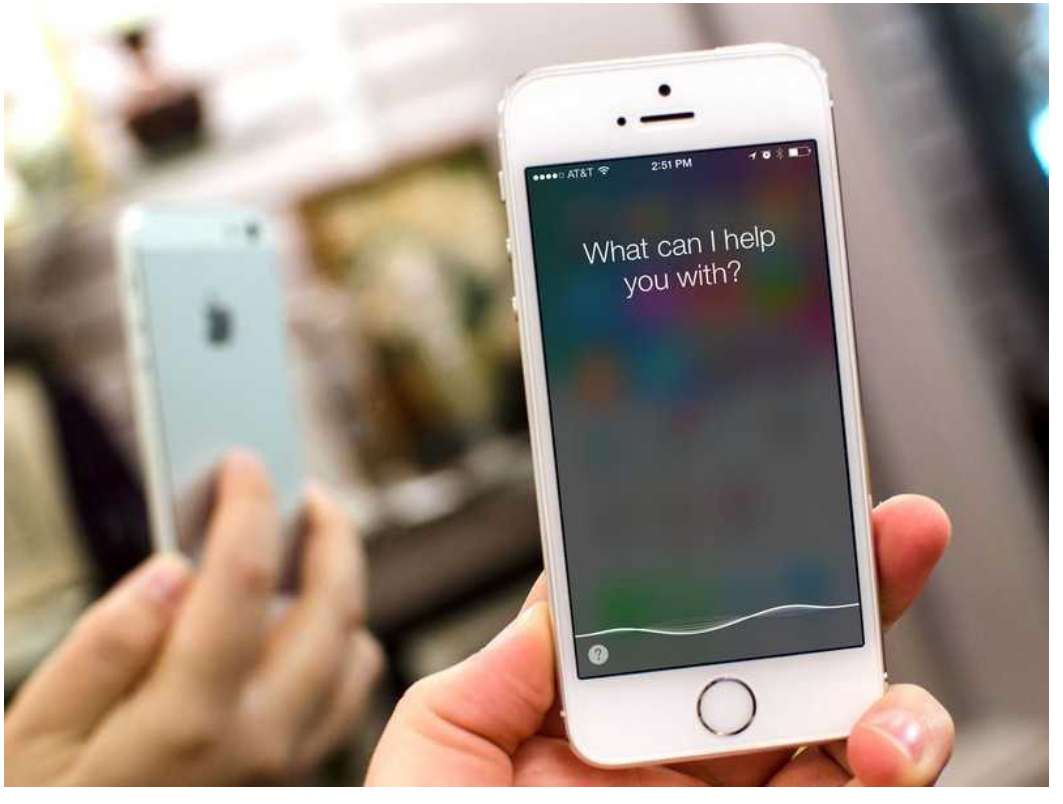


Figure: Siri: https://www.youtube.com/watch?v=8ciagGASro0

Examples of Recognizing patterns

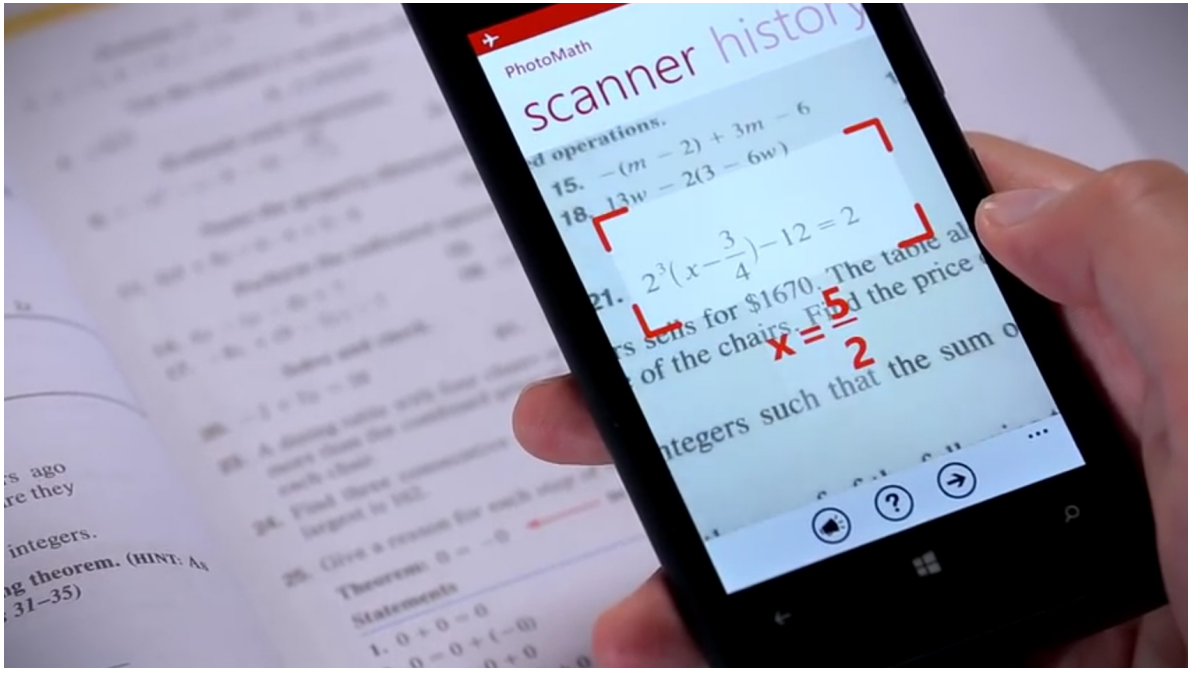
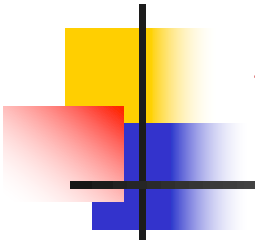


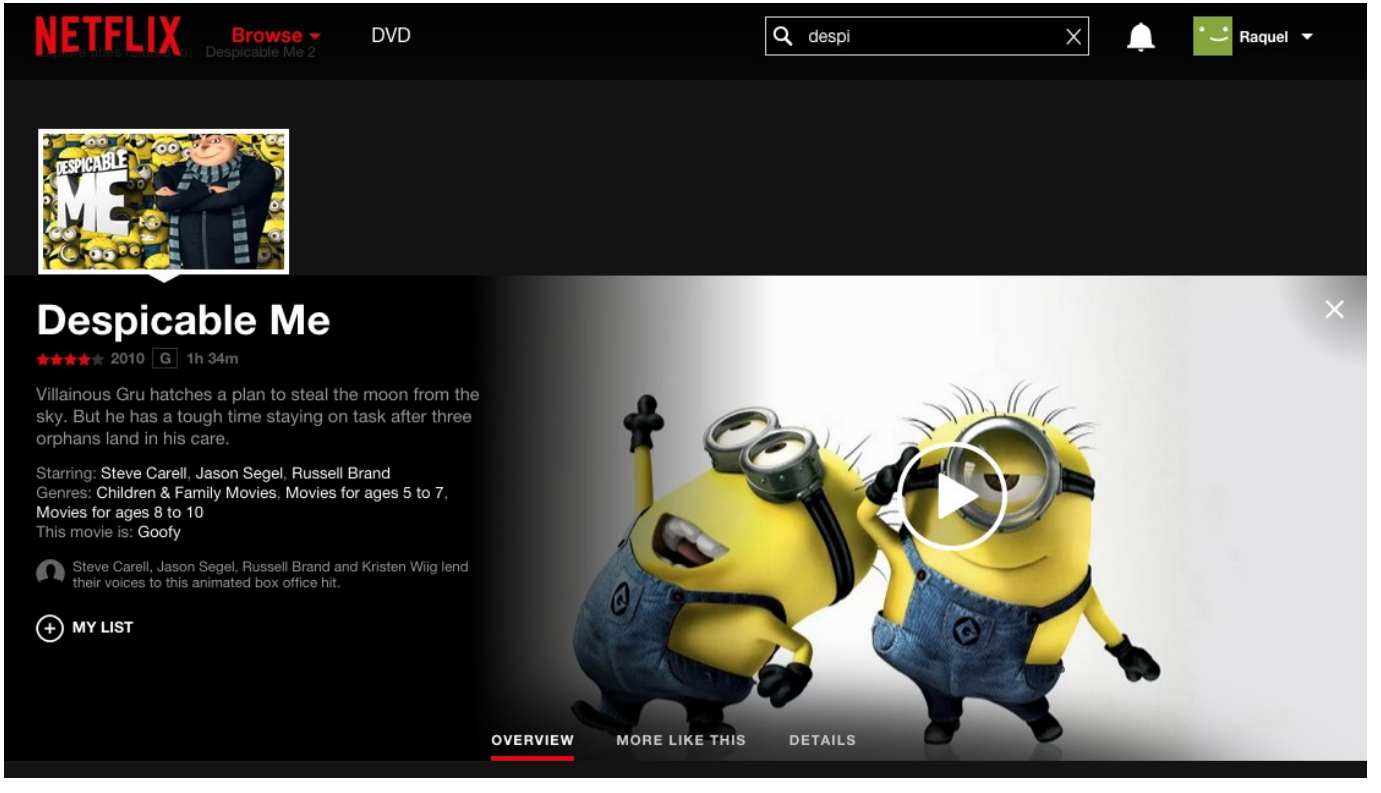
Figure: Photomath: https://photomath.net/



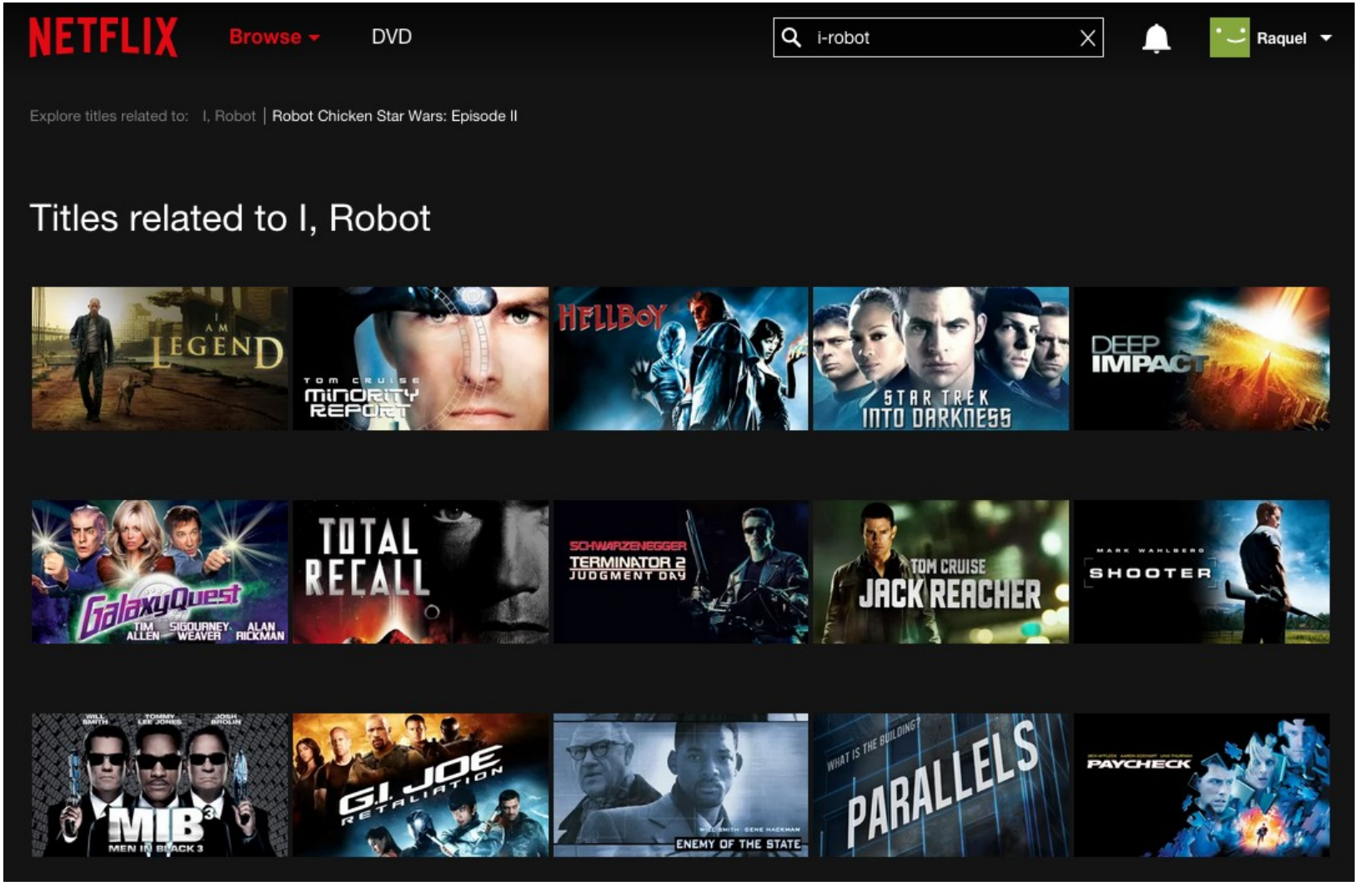
Learning algorithms are useful in many tasks

1. Classification: Determine which discrete category the example is
2. Recognizing patterns: Speech Recognition, facial identity, etc
3. Recommender Systems: Noisy data, commercial pay-off (e.g., Amazon, Netflix).

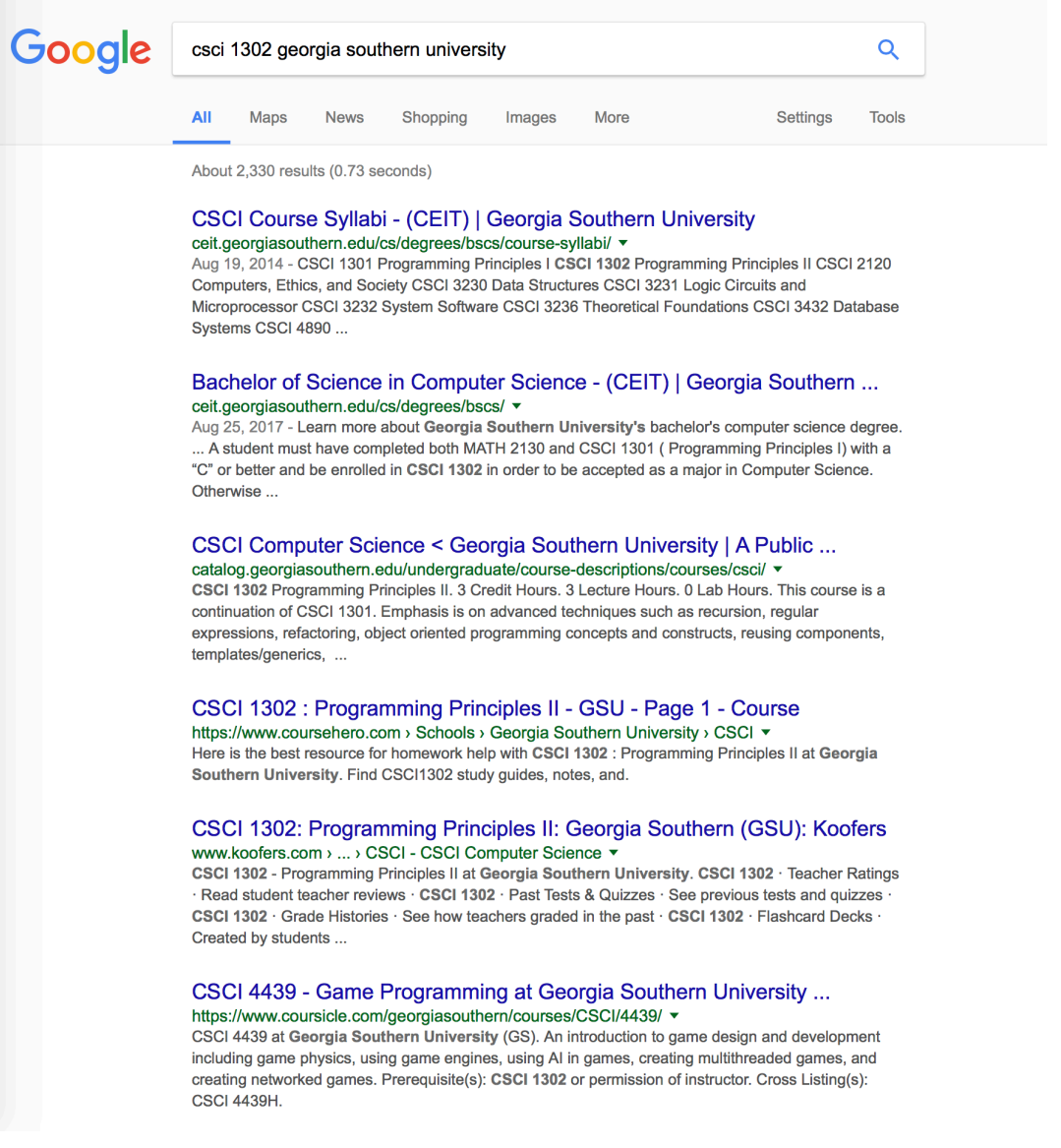
Examples of Recommendation systems



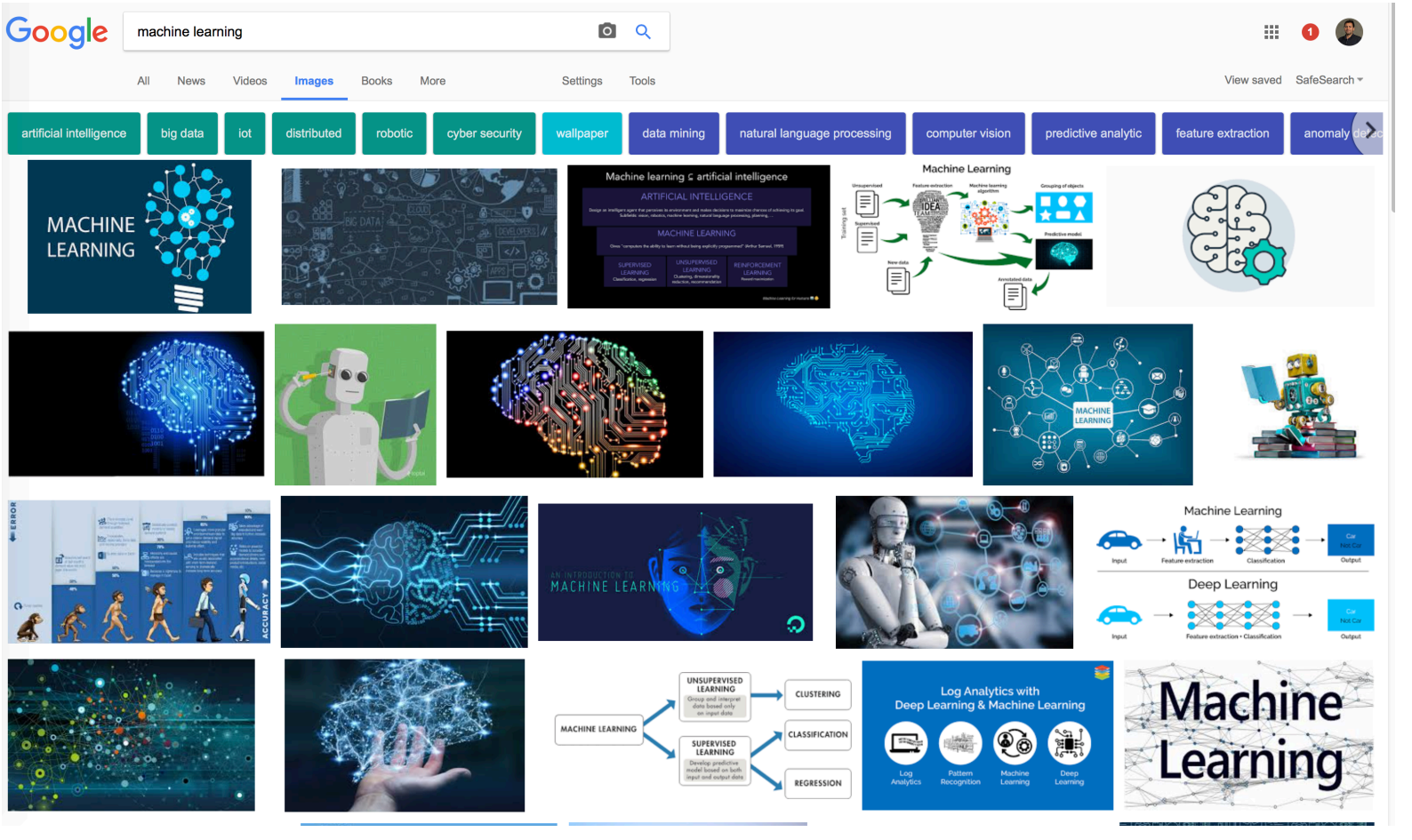
Examples of Recommendation systems



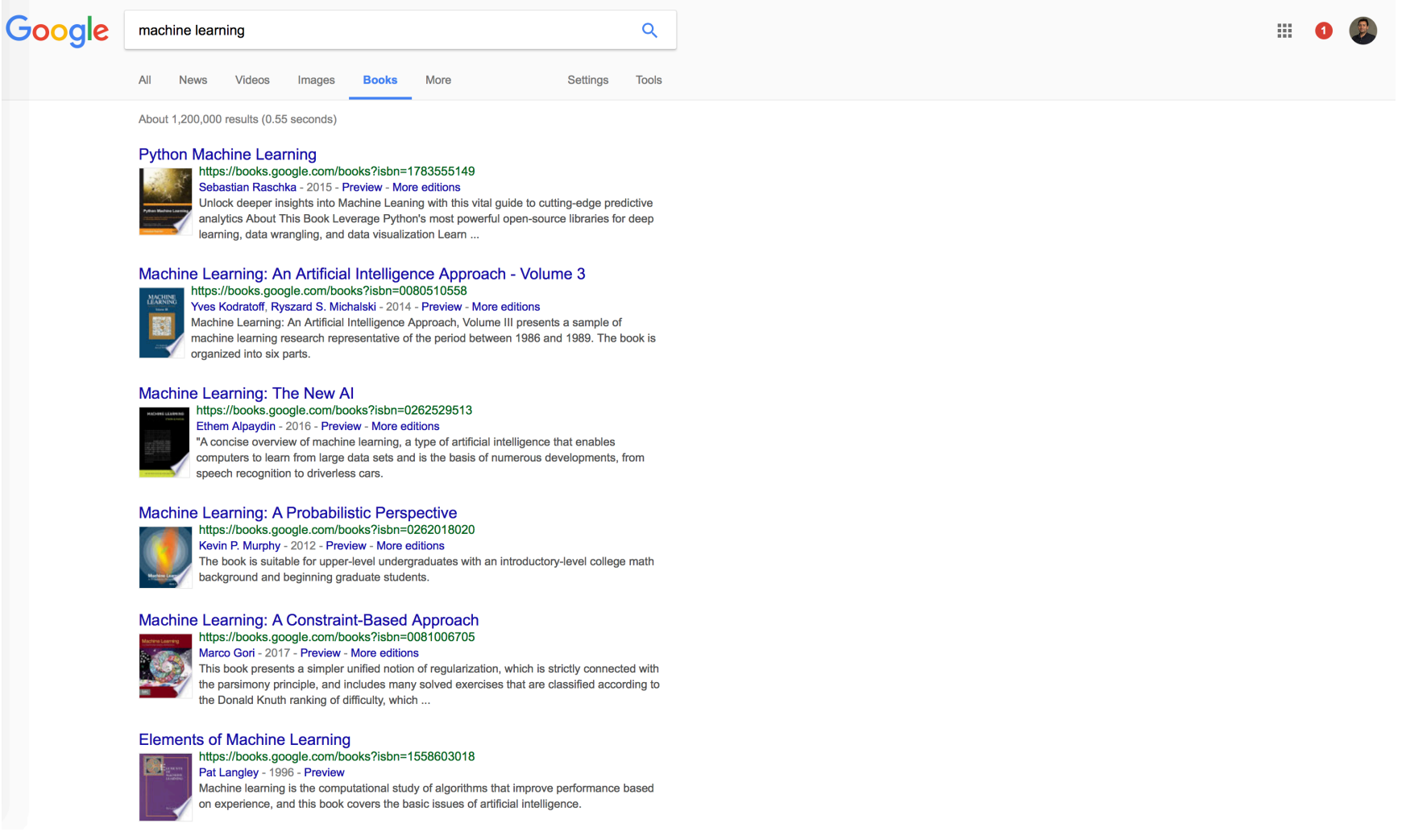
Examples of Information Retrieval



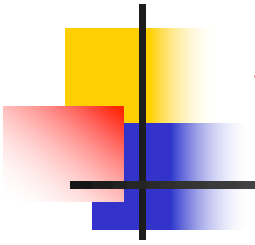
Examples of Information Retrieval



Examples of Information Retrieval



Learning algorithms are useful in many tasks



1. Classification: Determine which discrete category the example is
2. Recognizing patterns: Speech Recognition, facial identity, etc
3. Recommender Systems: Noisy data, commercial pay-off (e.g., Amazon, Netflix).
4. Information retrieval: Find documents or images with similar content
5. Computer vision: detection, segmentation, depth estimation, optical flow, etc

Examples of Computer Vision

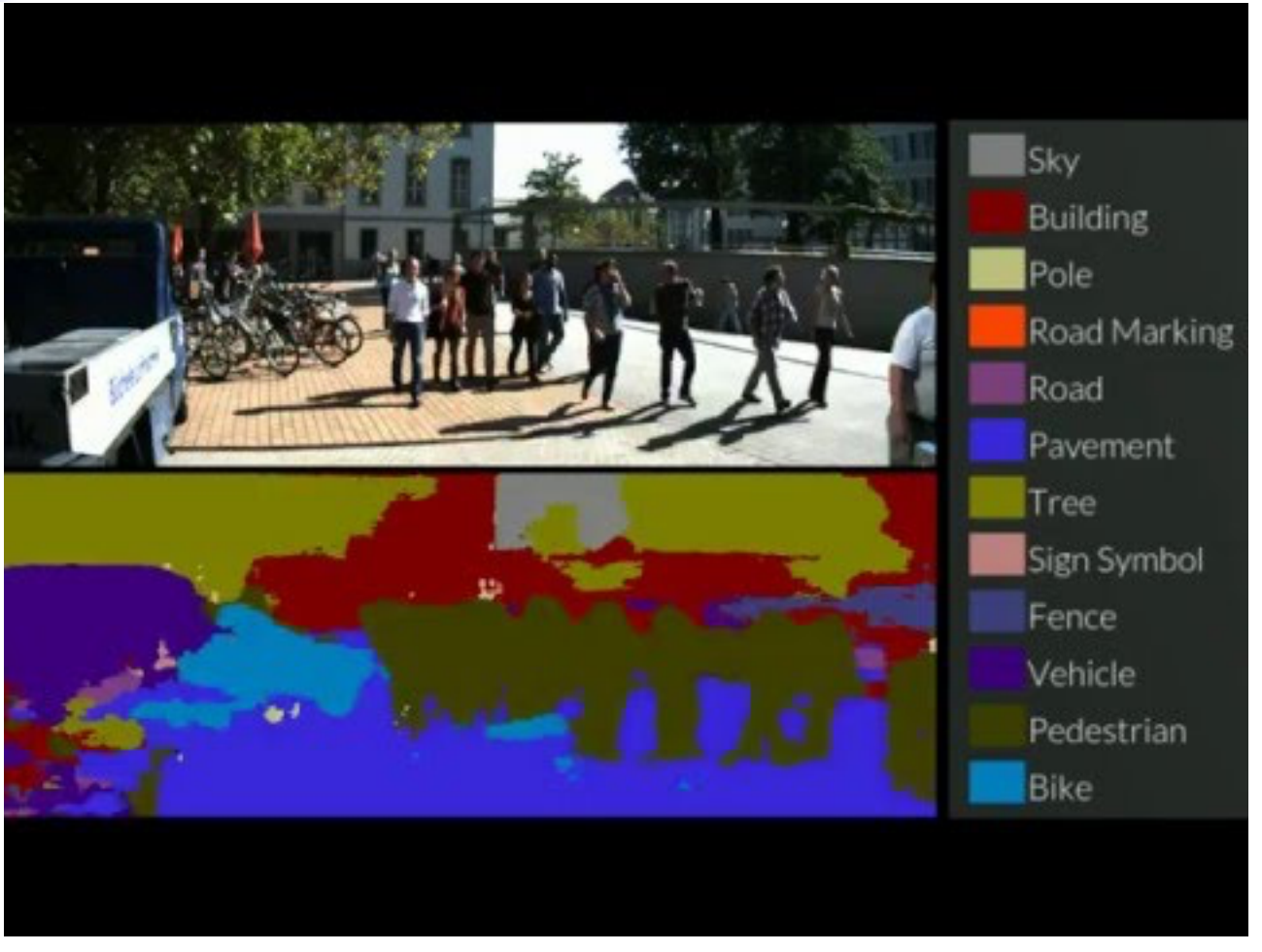
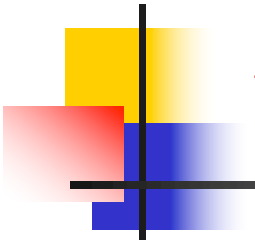


Fig: https://i.ytimg.com/vi/CrnLINlbfFA/hqdefault.jpg

https://www.youtube.com/watch?v=CrnLINlbfFA

Examples of Computer Vision

[Gatys, Ecker, Bethge. A Neural Algorithm of Artistic Style. Arxiv'15.]



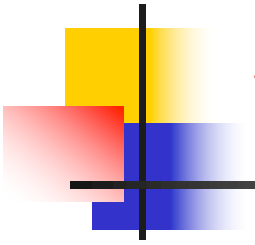
Learning algorithms are useful in many tasks

1. Classification: Determine which discrete category the example is
2. Recognizing patterns: Speech Recognition, facial identity, etc
3. Recommender Systems: Noisy data, commercial pay-off (e.g., Amazon, Netflix).
4. Information retrieval: Find documents or images with similar content
5. Computer vision: detection, segmentation, depth estimation, optical flow, etc
6. Robotics: perception, planning, etc

Autonomous Driving

Flying Robots

Figure: Video: https://www.youtube.com/watch?v=YQIMGV5vtd4



Learning algorithms are useful in many tasks

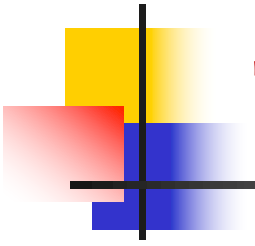
1. Classification: Determine which discrete category the example is
2. Recognizing patterns: Speech Recognition, facial identity, etc
3. Recommender Systems: Noisy data, commercial pay-off (e.g., Amazon, Netflix).
4. Information retrieval: Find documents or images with similar content
5. Computer vision: detection, segmentation, depth estimation, optical flow, etc
6. Robotics: perception, planning, etc
7. Learning to play games

Playing Games: Atari

Figure: Video: https://www.youtube.com/watch?v=V1eYniJ0Rnk

Learning algorithms are useful in many tasks

1. Classification: Determine which discrete category the example is
2. Recognizing patterns: Speech Recognition, facial identity, etc
3. Recommender Systems: Noisy data, commercial pay-off (e.g., Amazon, Netflix).
4. Information retrieval: Find documents or images with similar content
5. Computer vision: detection, segmentation, depth estimation, optical flow, etc
6. Robotics: perception, planning, etc
7. Learning to play games
8. Recognizing anomalies: Unusual sequences of credit card transactions, panic situation at an airport
9. Spam filtering, fraud detection: The enemy adapts so we must adapt too
10. Many more!

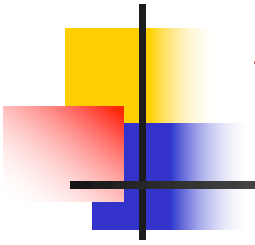


Types of Learning

- **Supervised (inductive) learning**
  - Training data includes desired outputs
- **Unsupervised learning**
  - Training data does not include desired outputs
- **Semi-supervised learning**
  - Training data includes a few desired outputs
- **Reinforcement learning**
  - Rewards from sequence of actions

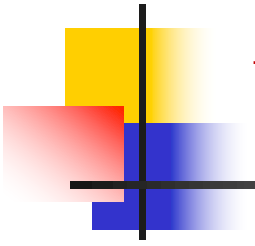
Supervised Learning

- **Given** examples of a function *(X, F(X))*
- **Predict** function *F(X)* for new examples *X*
  - **Discrete *F(X)*:** Classification (1-of-N output (speech recognition, object recognition, medical diagnosis)
  - **Continuous *F(X)*:** Regression (real-valued output (predicting market prices, customer rating)
  - *F(X)* = Probability(*X*): Probability estimation



Unsupervised Learning

- Create an internal representation of the input, capturing regularities/structure in data
- Example:
  - Clustering: Discover groups of similar inputs (documents, images, etc)



What We'll Cover

- **Supervised learning**
  - Decision tree induction
  - Instance-based learning
  - Bayesian learning
  - Logistic Regression
  - Neural networks
  - Support vector machines
- **Unsupervised learning**
  - Clustering
  - Dimensionality reduction