Course Logistics and Introduction

CS771: Introduction to Machine Learning

Piyush Rai

Course Logistics

Course Logistics

- Course Name: Introduction to Machine Learning – CS771

Course Logistics

- Course Name: Introduction to Machine Learning – CS771
  - An introductory course – supposed to be your first intro to the subject

Course Logistics

- Course Name: Introduction to Machine Learning – CS771
  - An introductory course – supposed to be your first intro to the subject
- Usually 3 lectures every week in form of videos (hosted on mooKIT)

Course Logistics

- Course Name: Introduction to Machine Learning – CS771
  - An introductory course – supposed to be your first intro to the subject

- Usually 3 lectures every week in form of videos (hosted on mooKIT)
  - Think can these as Mon/Wed/Fri lectures in the usual classroom setting

Course Logistics

- Course Name: Introduction to Machine Learning – CS771
  - An introductory course – supposed to be your first intro to the subject

- Usually 3 lectures every week in form of videos (hosted on mooKIT)
  - Think can these as Mon/Wed/Fri lectures in the usual classroom setting
  - mooKIT URL: https://hello.iitk.ac.in/cs771a/ (CC id and password to be used for login)

Course Logistics

- Course Name: Introduction to Machine Learning – CS771
  - An introductory course – supposed to be your first intro to the subject

- Usually 3 lectures every week in form of videos (hosted on mooKIT)
  - Think can these as Mon/Wed/Fri lectures in the usual classroom setting
  - mooKIT URL: https://hello.iitk.ac.in/cs771a/ (CC id and password to be used for login)

- An additional discussion session every Monday, 6pm-7pm (via YouTube Live)

Course Logistics

- Course Name: Introduction to Machine Learning – CS771
  - An introductory course – supposed to be your first intro to the subject
- Usually 3 lectures every week in form of videos (hosted on mooKIT)
  - Think can these as Mon/Wed/Fri lectures in the usual classroom setting
  - mooKIT URL: https://hello.iitk.ac.in/cs771a/ (CC id and password to be used for login)
- An additional discussion session every Monday, 6pm-7pm (via YouTube Live)
- All material will be posted on the mooKIT page for the course

Course Logistics

- Course Name: Introduction to Machine Learning – CS771
  - An introductory course – supposed to be your first intro to the subject

- Usually 3 lectures every week in form of videos (hosted on mooKIT)
  - Think can these as Mon/Wed/Fri lectures in the usual classroom setting
  - mooKIT URL: https://hello.iitk.ac.in/cs771a/ (CC id and password to be used for login)

- An additional discussion session every Monday, 6pm-7pm (via YouTube Live)

- All material will be posted on the mooKIT page for the course

- Q/A and announcements on Piazza. Please sign up

Course Team

Soumya Banerjee
soumyab@cse.iitk.ac.in

Shivam Bansal
sbansal@cse.iitk.ac.in

Dhanajit Brahma
dhanajit@cse.iitk.ac.in

Amit Chandak
amitch@cse.iitk.ac.in

Neeraj Matiyali
neermat@cse.iitk.ac.in

Pratik Mazumder
pratikm@cse.iitk.ac.in

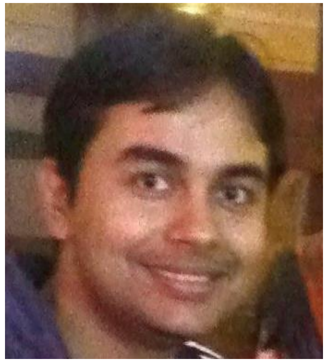
Course Team

Avik Pal
avikpal@cse.iitk.ac.in

Niravkumar Panchal
nirav@cse.iitk.ac.in

Hemant Sadana
soumyab@cse.iitk.ac.in

Rahul Sharma
rsharma@cse.iitk.ac.in



Piyush Rai
piyush@cse.iitk.ac.in

Workload and Grading Policy

Workload and Grading Policy

- 4 homework assignments (theory + programming) worth 50%

Workload and Grading Policy

- 4 homework assignments (theory + programming) worth 50%
  - Theory part: Derivations/analysis

Workload and Grading Policy

- 4 homework assignments (theory + programming) worth 50%
  - Theory part: Derivations/analysis
  - Programming part: Implement/use ML algos, analysis of results. Must be done in Python (learn if not already familiar)

Workload and Grading Policy

- 4 homework assignments (theory + programming) worth 50%
  - Theory part: Derivations/analysis
  - Programming part: Implement/use ML algos, analysis of results. Must be done in Python (learn if not already familiar)

Python: https://www.geeksforgeeks.org/python-programming-language/

Workload and Grading Policy

- 4 homework assignments (theory + programming) worth 50%
  - Theory part: Derivations/analysis
  - Programming part: Implement/use ML algos, analysis of results. Must be done in Python (learn if not already familiar)
  - Must be typeset in LaTeX (learn if not already familiar)

Python: https://www.geeksforgeeks.org/python-programming-language/

Workload and Grading Policy

- 4 homework assignments (theory + programming) worth 50%
  - Theory part: Derivations/analysis
  - Programming part: Implement/use ML algos, analysis of results. Must be done in Python (learn if not already familiar)
  - Must be typeset in LaTeX (learn if not already familiar)

Python: https://www.geeksforgeeks.org/python-programming-language/

LaTeX: www.sharelatex.com/blog/latex-guides/beginners-tutorial.html
www.overleaf.com/learn/latex/Tutorials

Workload and Grading Policy

- 4 homework assignments (theory + programming) worth 50%
  - Theory part: Derivations/analysis
  - Programming part: Implement/use ML algos, analysis of results. Must be done in Python (learn if not already familiar)
  - Must be typeset in LaTeX (learn if not already familiar)
  - To be submitted via Gradescope (login details will be provided)

Python: https://www.geeksforgeeks.org/python-programming-language/

LaTeX: www.sharelatex.com/blog/latex-guides/beginners-tutorial.html
www.overleaf.com/learn/latex/Tutorials

Workload and Grading Policy

- 4 homework assignments (theory + programming) worth 50%
  - Theory part: Derivations/analysis
  - Programming part: Implement/use ML algos, analysis of results. Must be done in Python (learn if not already familiar)
  - Must be typeset in LaTeX (learn if not already familiar)
  - To be submitted via Gradescope (login details will be provided)

- Quizzes and exams (mid-sem and end-sem) worth 50%

Python: https://www.geeksforgeeks.org/python-programming-language/

LaTeX: www.sharelatex.com/blog/latex-guides/beginners-tutorial.html
www.overleaf.com/learn/latex/Tutorials

Workload and Grading Policy

- 4 homework assignments (theory + programming) worth 50%
  - Theory part: Derivations/analysis
  - Programming part: Implement/use ML algos, analysis of results. Must be done in Python (learn if not already familiar)
  - Must be typeset in LaTeX (learn if not already familiar)
  - To be submitted via Gradescope (login details will be provided)

- Quizzes and exams (mid-sem and end-sem) worth 50%
  - Will be held online – details later

Python: https://www.geeksforgeeks.org/python-programming-language/

LaTeX: www.sharelatex.com/blog/latex-guides/beginners-tutorial.html
www.overleaf.com/learn/latex/Tutorials

Workload and Grading Policy

- 4 homework assignments (theory + programming) worth 50%
  - Theory part: Derivations/analysis
  - Programming part: Implement/use ML algos, analysis of results. Must be done in Python (learn if not already familiar)
  - Must be typeset in LaTeX (learn if not already familiar)
  - To be submitted via Gradescope (login details will be provided)

- Quizzes and exams (mid-sem and end-sem) worth 50%
  - Will be held online – details later
  - Exact break-up of individual components will be announced in a few days

Python: https://www.geeksforgeeks.org/python-programming-language/

LaTeX: www.sharelatex.com/blog/latex-guides/beginners-tutorial.html
www.overleaf.com/learn/latex/Tutorials

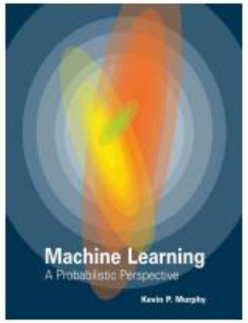
Textbook and References

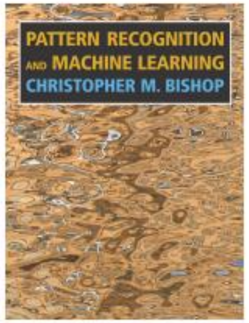
Textbook and References

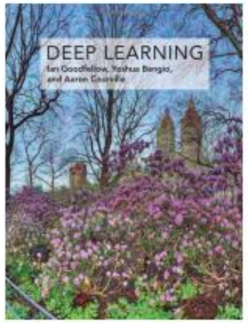
- Many excellent texts but none "required". Some include:

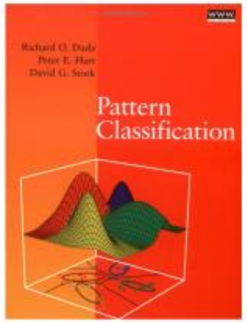
Textbook and References

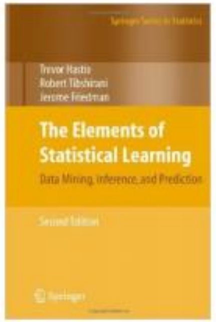
- Many excellent texts but none "required". Some include:

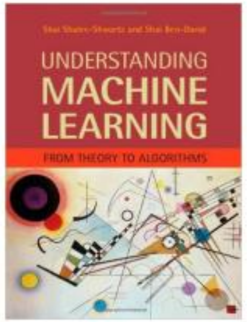


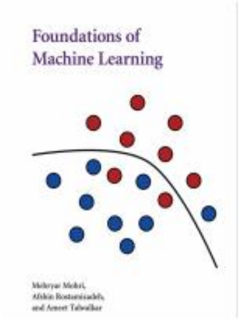






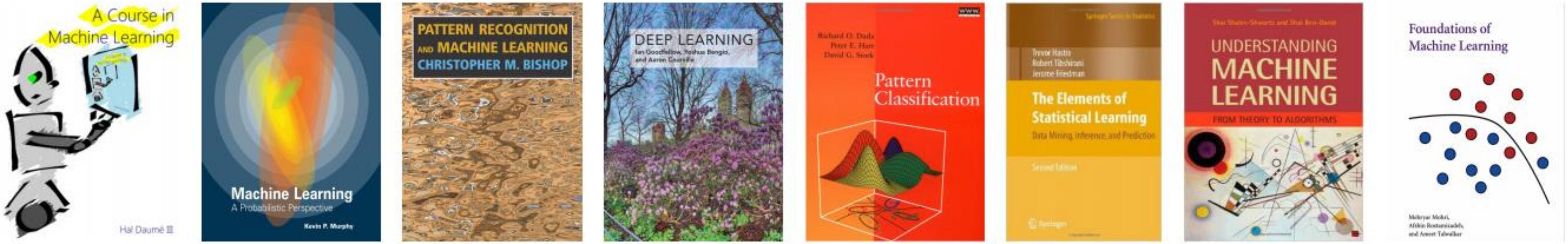






Textbook and References

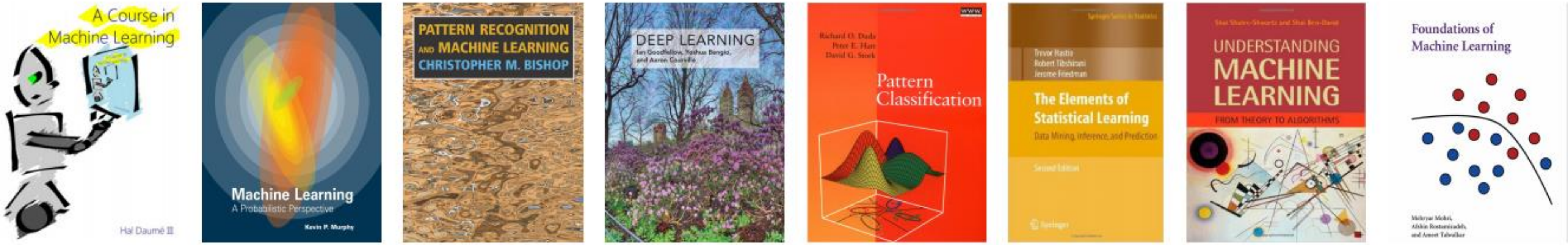
- Many excellent texts but none “required”. Some include:



- Different books might vary in terms of

Textbook and References

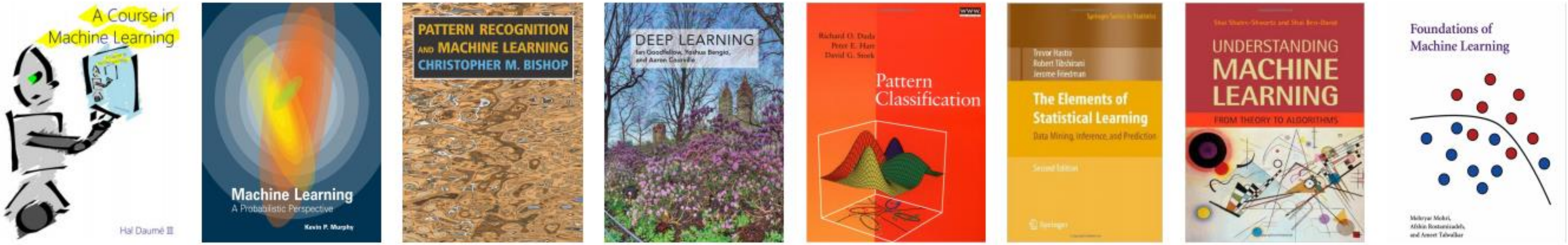
- Many excellent texts but none “required”. Some include:



- Different books might vary in terms of
  - Set of topics covered

Textbook and References

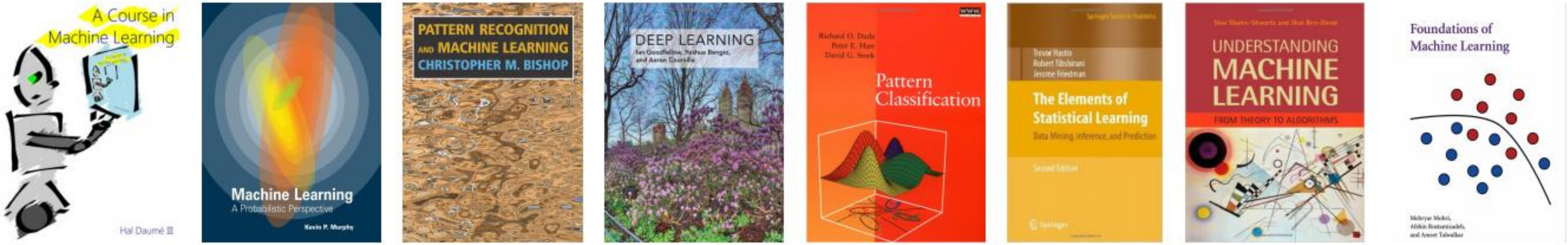
- Many excellent texts but none “required”. Some include:



- Different books might vary in terms of
  - Set of topics covered
  - Flavor (e.g., classical statistics, deep learning, probabilistic/Bayesian, theory)

Textbook and References

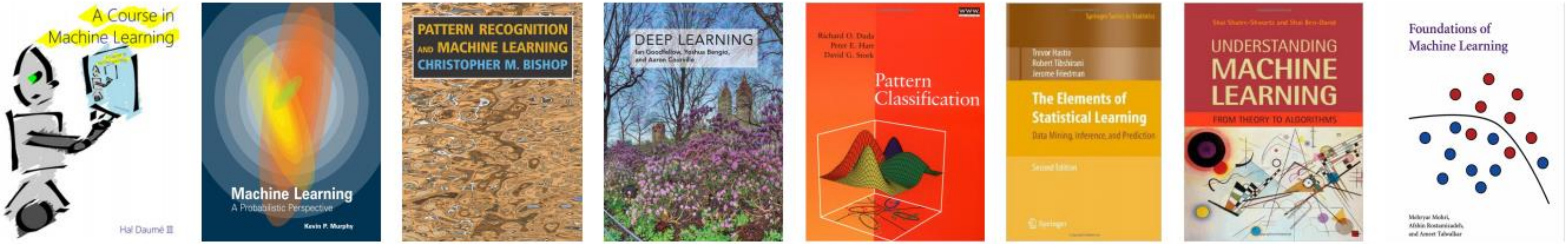
- Many excellent texts but none “required”. Some include:



- Different books might vary in terms of
  - Set of topics covered
  - Flavor (e.g., classical statistics, deep learning, probabilistic/Bayesian, theory)
  - Terminology and notation (beware of this especially)

Textbook and References

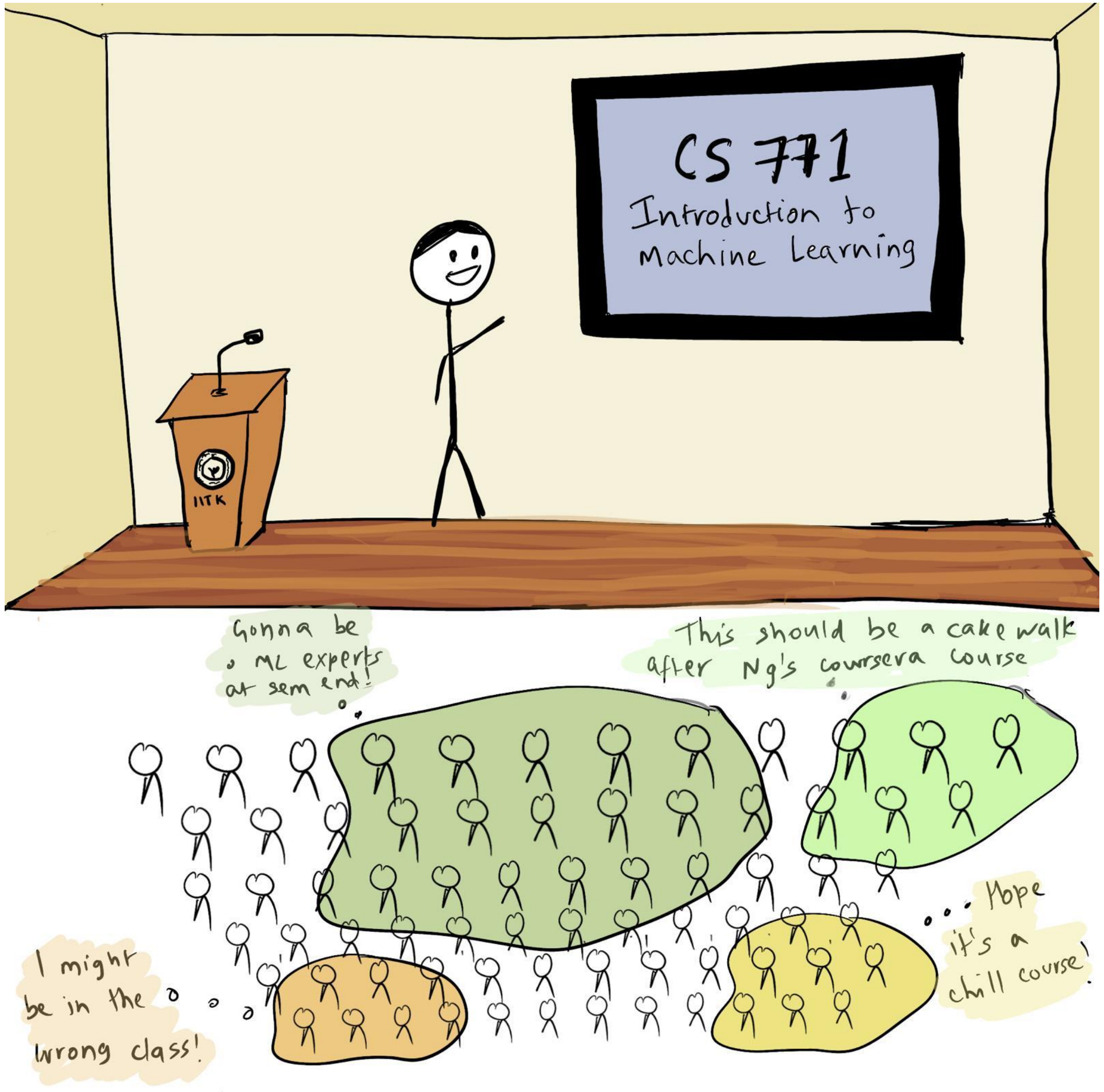
- Many excellent texts but none “required”. Some include:



- Different books might vary in terms of
  - Set of topics covered
  - Flavor (e.g., classical statistics, deep learning, probabilistic/Bayesian, theory)
  - Terminology and notation (beware of this especially)

- We will provide you the reading material from the relevant sources

Course Goals



Credit: Rishika Agarwal (EE, graduated 2017)

Course Real Goals..

Course Real Goals..

- Introduction to the foundations of machine learning models and algos

Course Real Goals..

- Introduction to the foundations of machine learning models and algos

- Focus on developing the ability to

Course Real Goals..

- Introduction to the foundations of machine learning models and algos
- Focus on developing the ability to
  - Understand the underlying principles behind ML models and algos

Course Real Goals..

- Introduction to the foundations of machine learning models and algos

- Focus on developing the ability to
  - Understand the underlying principles behind ML models and algos
  - Understand how to implement and evaluate them

Course Real Goals..

- Introduction to the foundations of machine learning models and algos

- Focus on developing the ability to
  - Understand the underlying principles behind ML models and algos
  - Understand how to implement and evaluate them
  - Understand/develop intuition on choosing the right ML model/algo for your problem

Course <u>Real</u> Goals..

- Introduction to the foundations of machine learning models and algos

- Focus on developing the ability to
  - Understand the underlying principles behind ML models and algos
  - Understand how to implement and evaluate them
  - Understand/develop intuition on choosing the right ML model/algo for your problem

- (Hopefully) inspire you to work on and learn more about ML

Course Real Goals..

- Introduction to the foundations of machine learning models and algos
- Focus on developing the ability to
  - Understand the underlying principles behind ML models and algos
  - Understand how to implement and evaluate them
  - Understand/develop intuition on choosing the right ML model/algo for your problem
- (Hopefully) inspire you to work on and learn more about ML
- Not an introduction to popular software frameworks and libraries, such as scikit-learn, PyTorch, Tensorflow, etc

Course Real Goals..

- Introduction to the foundations of machine learning models and algos
- Focus on developing the ability to
  - Understand the underlying principles behind ML models and algos
  - Understand how to implement and evaluate them
  - Understand/develop intuition on choosing the right ML model/algo for your problem
- (Hopefully) inspire you to work on and learn more about ML
- Not an introduction to popular software frameworks and libraries, such as scikit-learn, PyTorch, Tensorflow, etc
  - Can explore once you have some understanding of various ML techniques

Introduction to Machine Learning

Machine Learning (ML)

Machine Learning (ML)

- Designing algorithms that ingest data and learn a model of the data

Machine Learning (ML)

- Designing algorithms that ingest data and learn a model of the data
- The learned model can be used to

Machine Learning (ML)

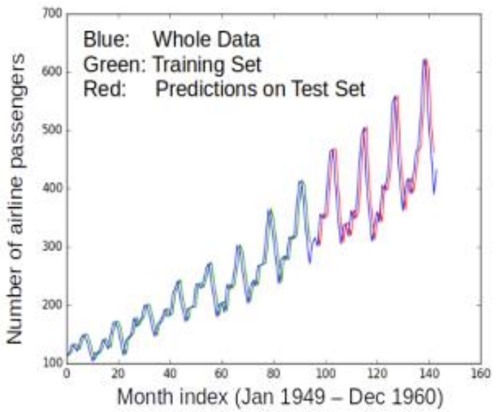
- Designing algorithms that ingest data and learn a model of the data
- The learned model can be used to
  - Detect patterns/structures/themes/trends etc. in the data

Machine Learning (ML)

- Designing algorithms that ingest data and learn a model of the data
- The learned model can be used to
  - Detect patterns/structures/themes/trends etc. in the data
  - Make predictions about future data and make decisions

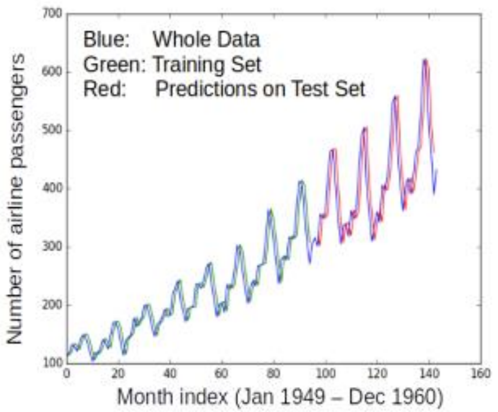
Machine Learning (ML)

- Designing algorithms that ingest data and learn a model of the data
- The learned model can be used to
  - Detect patterns/structures/themes/trends etc. in the data
  - Make predictions about future data and make decisions



Machine Learning (ML)

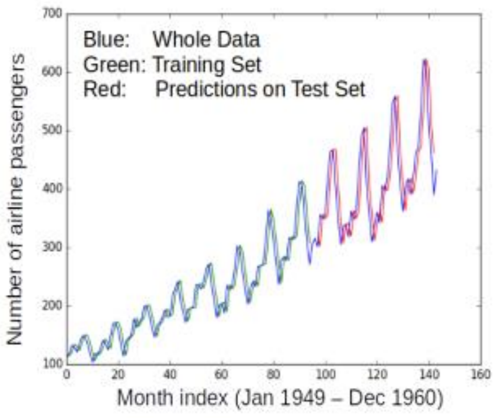
- Designing algorithms that ingest data and learn a model of the data
- The learned model can be used to
  - Detect patterns/structures/themes/trends etc. in the data
  - Make predictions about future data and make decisions



- Modern ML algorithms are heavily "data-driven"

Machine Learning (ML)

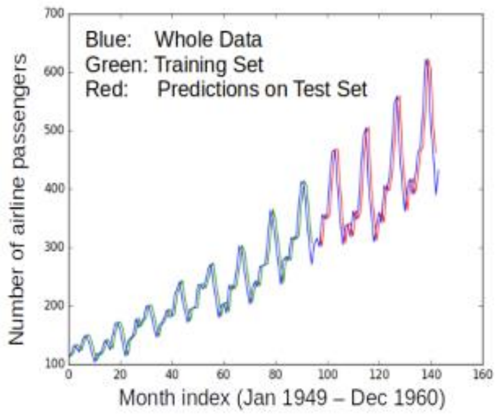
- Designing algorithms that ingest data and learn a model of the data
- The learned model can be used to
  - Detect patterns/structures/themes/trends etc. in the data
  - Make predictions about future data and make decisions



- Modern ML algorithms are heavily “data-driven”
  - No need to pre-define and hard-code all the rules (infeasible/impossible anyway)

Machine Learning (ML)

- Designing algorithms that ingest data and learn a model of the data
- The learned model can be used to
  - Detect patterns/structures/themes/trends etc. in the data
  - Make predictions about future data and make decisions



- Modern ML algorithms are heavily "data-driven"
  - No need to pre-define and hard-code all the rules (infeasible/impossible anyway)
  - The rules are not "static"; can adapt as the ML algo ingests more and more data

ML: From What It Does to How It Does It?

ML: From What It Does to How It Does It?

- ML enables intelligent systems to be data-driven rather than rule-driven

ML: From What It Does to How It Does It?

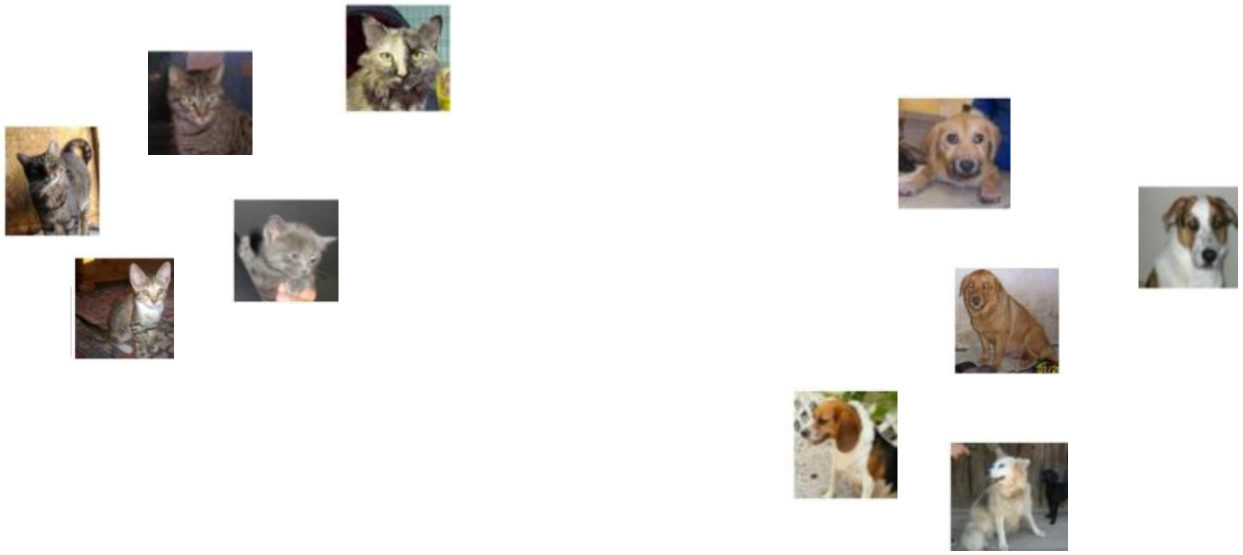
- ML enables intelligent systems to be data-driven rather than rule-driven
- How: By supplying training data and building statistical models of data

ML: From What It Does to How It Does It?

- ML enables intelligent systems to be data-driven rather than rule-driven
- How: By supplying training data and building statistical models of data
- Pictorial illustration of an ML model for binary classification:

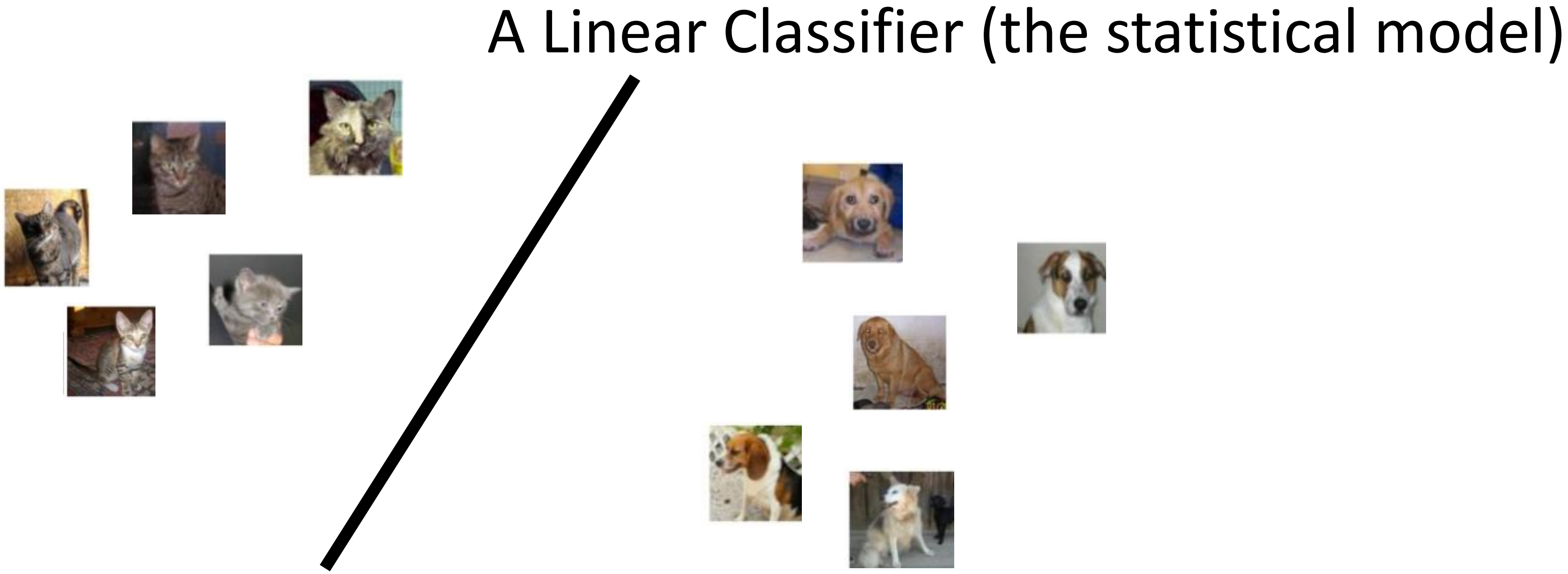
ML: From What It Does to How It Does It?

- ML enables intelligent systems to be data-driven rather than rule-driven
- How: By supplying training data and building statistical models of data
- Pictorial illustration of an ML model for binary classification:



ML: From What It Does to How It Does It?

- ML enables intelligent systems to be data-driven rather than rule-driven
- How: By supplying training data and building statistical models of data
- Pictorial illustration of an ML model for binary classification:



ML: From What It Does to How It Does It?

- ML enables intelligent systems to be data-driven rather than rule-driven
- How: By supplying training data and building statistical models of data
- Pictorial illustration of an ML model for binary classification:

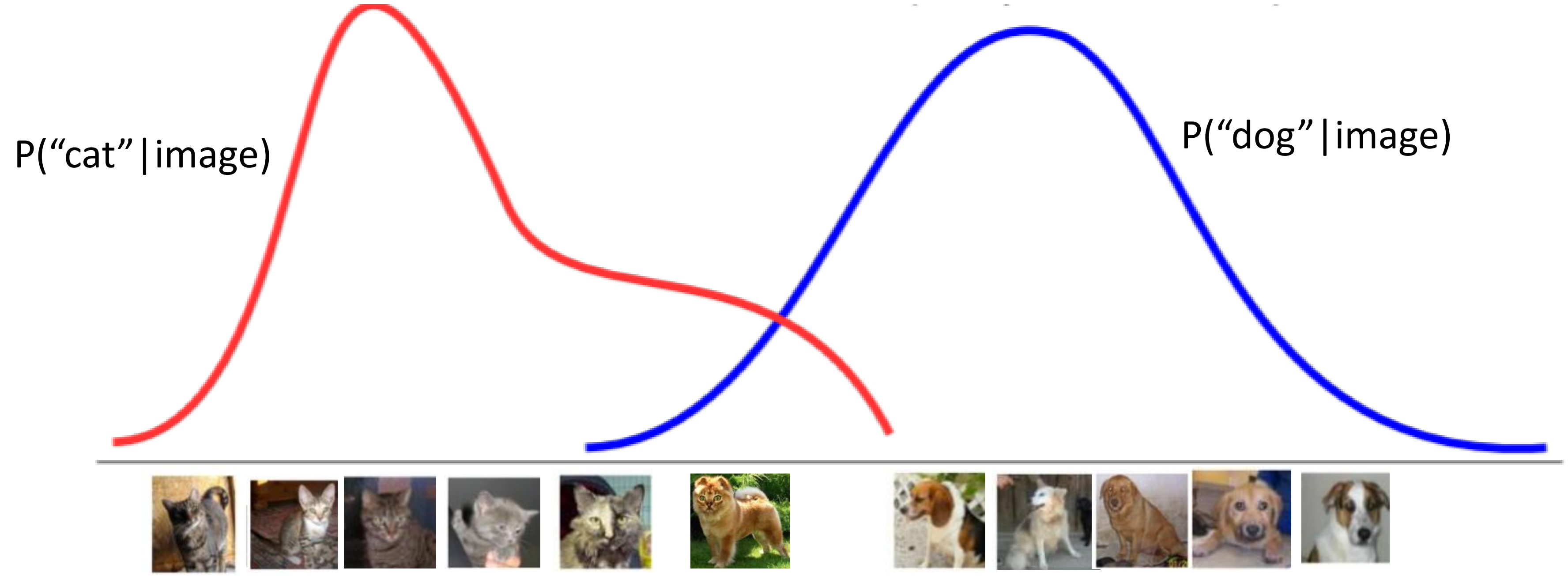
ML: From What It Does to How It Does It?

- ML enables intelligent systems to be data-driven rather than rule-driven
- How: By supplying training data and building statistical models of data
- Pictorial illustration of an ML model for binary classification:

ML: From What It Does to How It Does It?

- ML enables intelligent systems to be data-driven rather than rule-driven
- How: By supplying training data and building statistical models of data
- Pictorial illustration of an ML model for binary classification:

A Probabilistic Classifier (the statistical model)



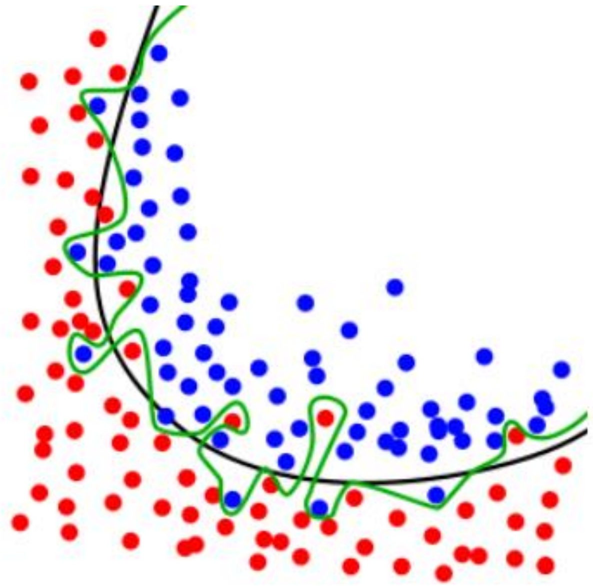
Overfitting = Bad ML

Overfitting = Bad ML

- Doing perfectly on training data is not good enough

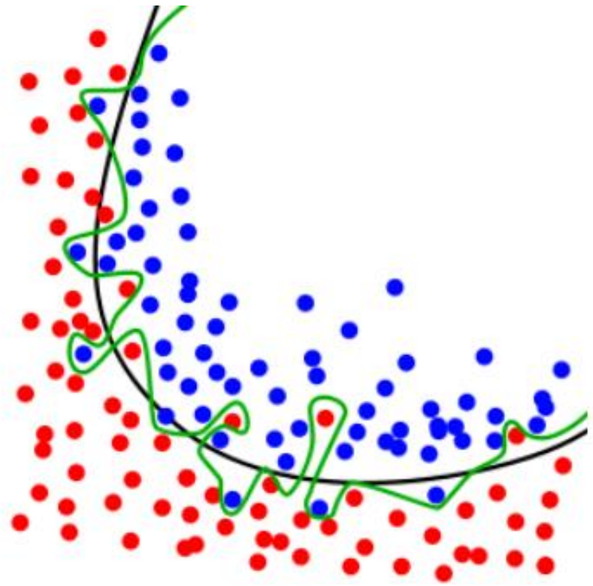
Overfitting = Bad ML

- Doing perfectly on training data is not good enough



Overfitting = Bad ML

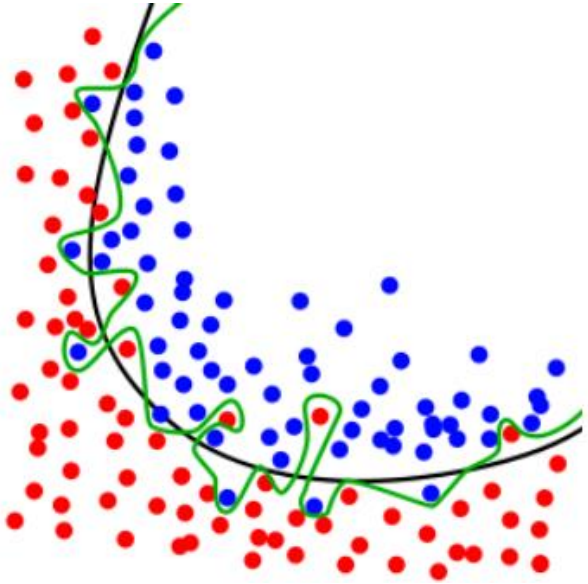
- Doing perfectly on training data is not good enough



- A good ML model must generalize well on unseen (test data)

Overfitting = Bad ML

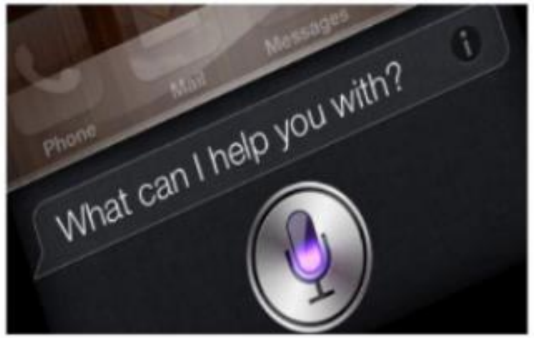
- Doing perfectly on training data is not good enough

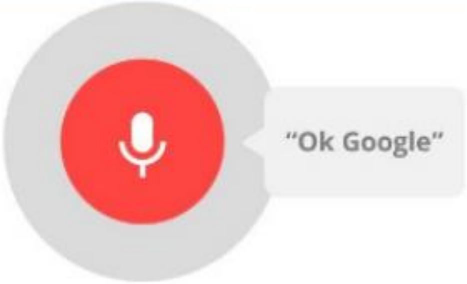


- A good ML model must generalize well on unseen (test data)

- Simpler models should be preferred over more complex ones!

ML Applications Abound..





Predictive Policing

Online Fraud Detection

Picture courtesy: gizmodo.com,rcdronearena.com,www.wiseyak.com,www.charlesdong.com

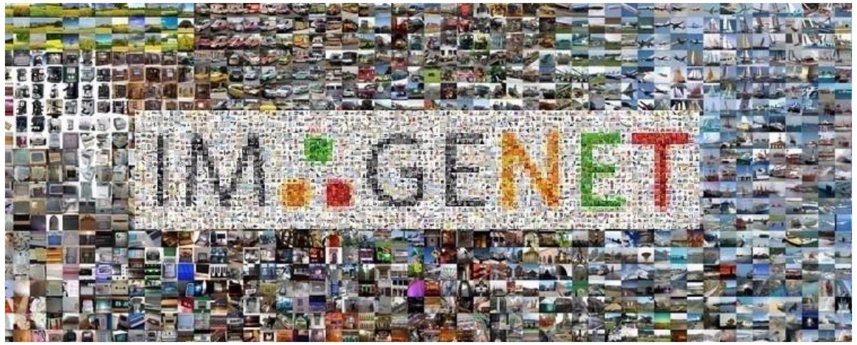
Key Enablers for Modern ML

Key Enablers for Modern ML

- Availability of large amounts of data to train ML models

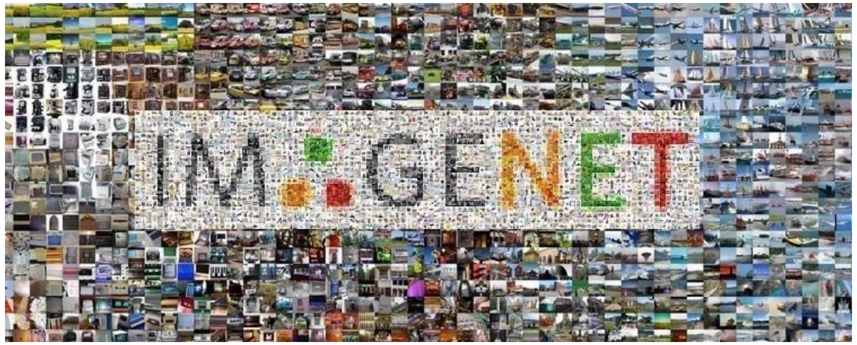
Key Enablers for Modern ML

- Availability of large amounts of data to train ML models



Key Enablers for Modern ML

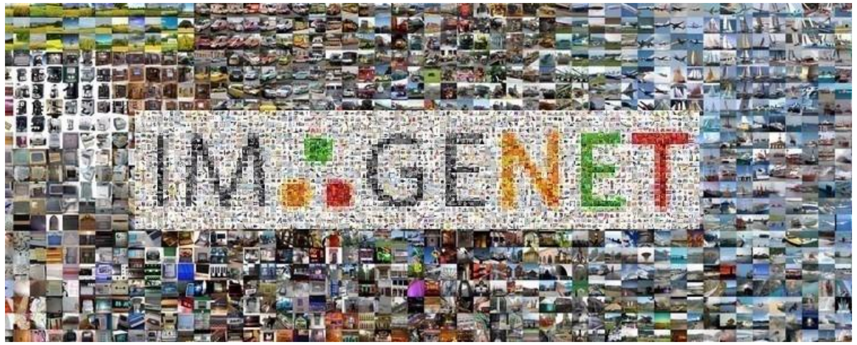
- Availability of large amounts of data to train ML models



- Increased computing power (e.g., GPUs)

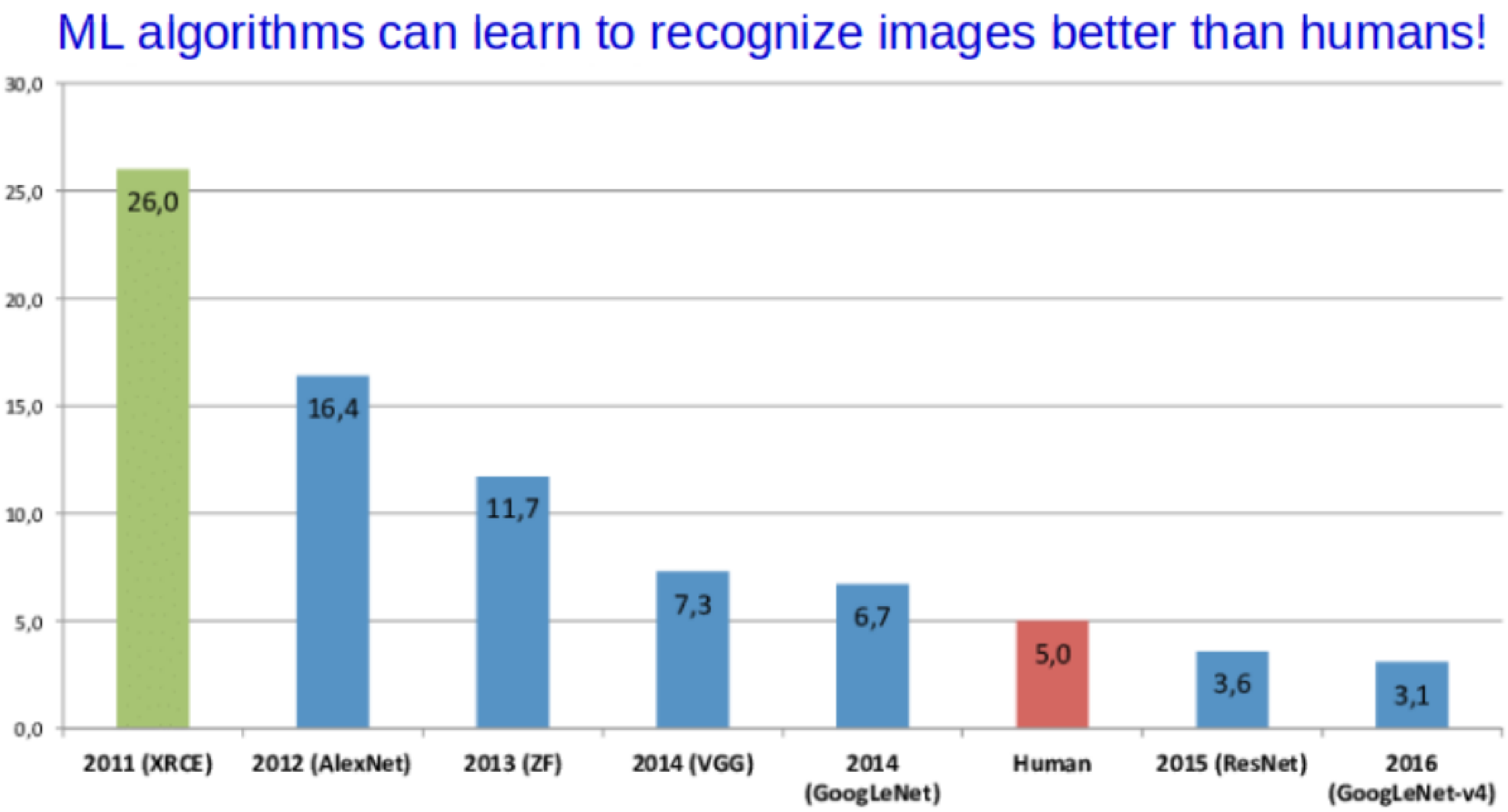
Key Enablers for Modern ML

- Availability of large amounts of data to train ML models



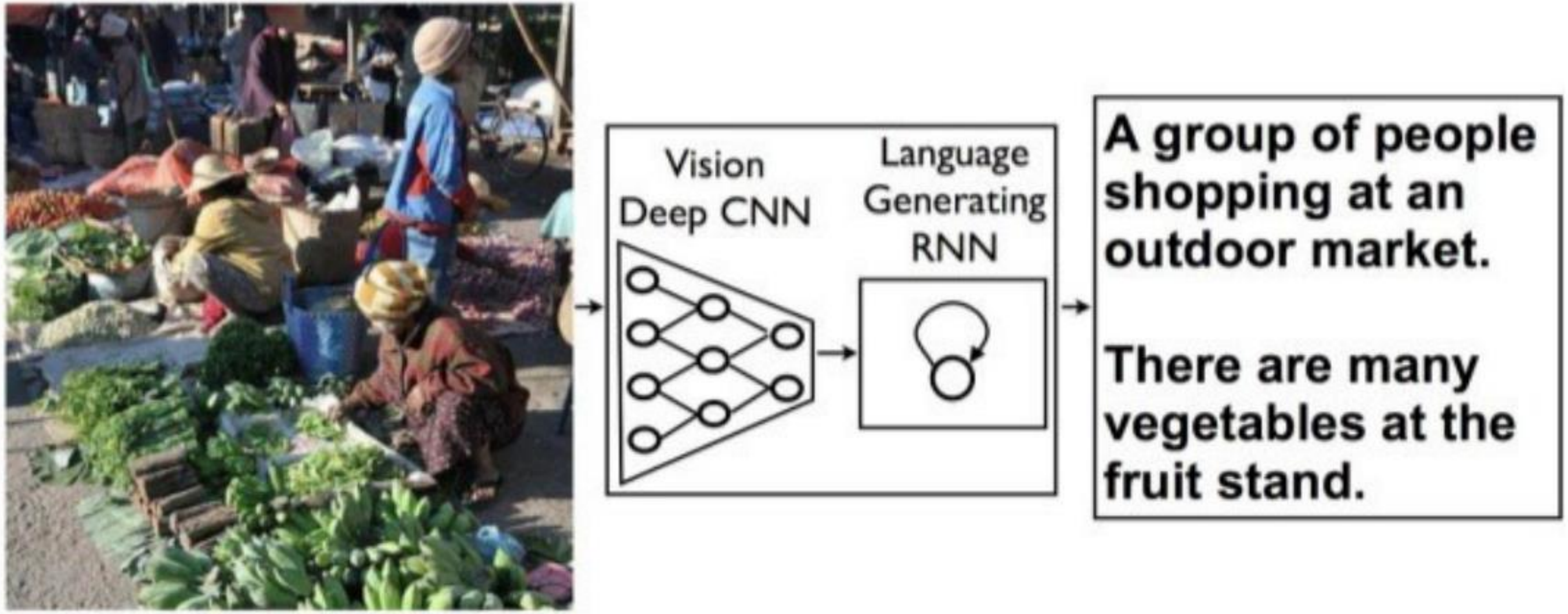
- Increased computing power (e.g., GPUs)

ML: Some Success Stories



ML: Some Success Stories

ML algorithms can learn to generate captions for images



http://arxiv.org/abs/1411.4555 "Show and Tell: A Neural Image Caption Generator"

ML: Some Success Stories

ML algorithms can learn to translate speech in **real time**

Picture courtesy: https://news.microsoft.com/

ML: Some Success Stories

- Automatic Program Correction

```
#include<stdio.h>
int main(){
 int a;
 scanf("%d", a );
 printf("ans=%d",
  a+10);
 return 0;
}
```

```
#include<stdio.h>
int main(){
 int a;
 scanf("%d", &a );
 printf("ans=%d",
  a+10);
 return 0;
}
```

Figure 1: Left: erroneous program, Right: fix by TRACER. The compiler message read: *Line-4, Column-9: warning: format '%d' expects argument of type 'int *', but argument 2 has type 'int'.*

```
#include<stdio.h>
int main(){
    int x,x1,d;
    // ...
    d=(x-x1)(x-x1);
    return d;
}
```

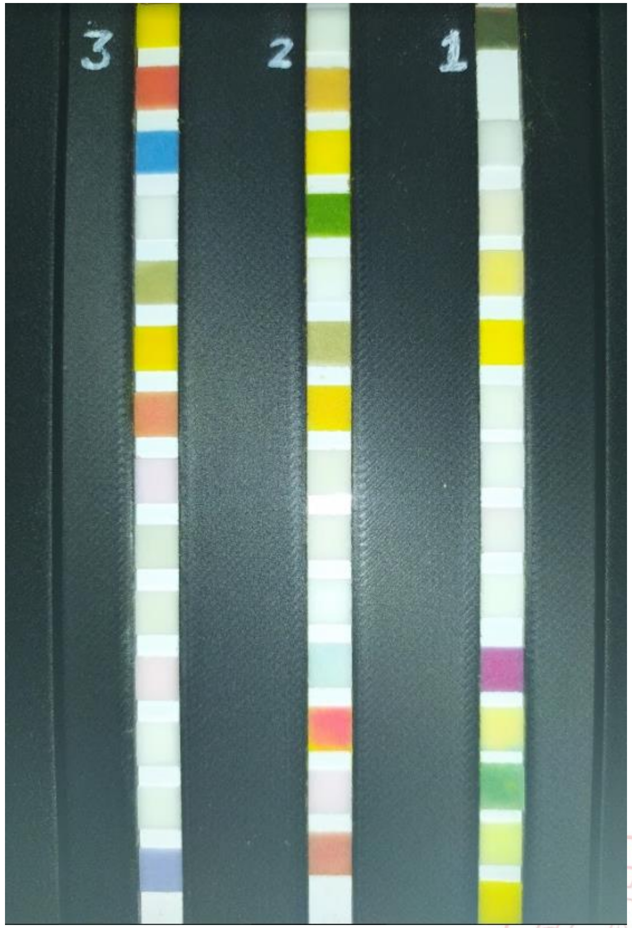
```
#include<stdio.h>
int main(){
    int x,x1,d;
    // ...
    d=(x-x1)*(x-x1);
    return d;
}
```

Figure 2: Left: erroneous program, Right: fix by TRACER. The compiler message read: *Line-5, Column-11: error: called object type 'int' is not a function or function pointer.*

Example from "Compilation error repair: for the student programs, from the student programs", Ahmed et al (2018)

ML: Some Success Stories

- ML based colorimetry for water quality assessment
- Take uncontaminated water sample
- Spike it with known concentration of various compounds (e.g., lead, iron, fluoride, etc)
- Dip a test strip (one square to measure each compound) in the contaminated water for some time.
- Take a picture of the strip using a phone camera to capture how the colors have changed
- Train an ML model to predict concentration levels of various compounds based on color levels in the images



(work being done at IITK in collaboration with two startups - Earthface Analytics Pvt Ltd and Kritsnam Technologies Pvt Ltd)

Good ML Systems Should be Fair and Unbiased

Good ML Systems Should be Fair and Unbiased

- Good ML should not just be about getting high accuracies

Good ML Systems Should be Fair and Unbiased

- Good ML should not just be about getting high accuracies

- Should also ensure that the ML models are fair and unbiased

Good ML Systems Should be Fair and Unbiased

- Good ML should not just be about getting high accuracies

- Should also ensure that the ML models are fair and unbiased

An image captioning system should not always assume a specific gender in examples like the above

Good ML Systems Should be Fair and Unbiased

- Good ML should not just be about getting high accuracies

- Should also ensure that the ML models are fair and unbiased

An image captioning system should not always assume a specific gender in examples like the above

Don't want a self-driving car that is more likely to hit black people than white people

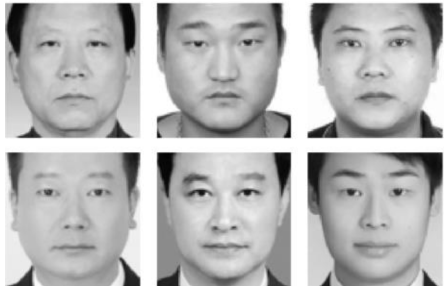
Good ML Systems Should be Fair and Unbiased

- Good ML should not just be about getting high accuracies

- Should also ensure that the ML models are fair and unbiased

An image captioning system should not always assume a specific gender in examples like the above

Don’t want a self-driving car that is more likely to hit black people than white people



Criminals?

Not Criminals?

Don’t want a predictive policing system that predicts criminality using facial features

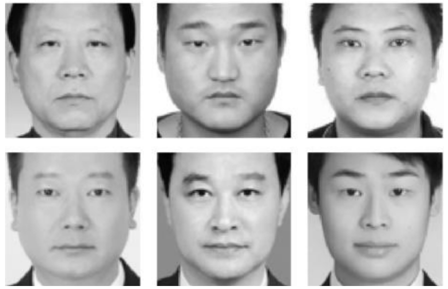
Good ML Systems Should be Fair and Unbiased

- Good ML should not just be about getting high accuracies

- Should also ensure that the ML models are fair and unbiased

An image captioning system should not always assume a specific gender in examples like the above

Don't want a self-driving car that is more likely to hit black people than white people



Criminals?

Not Criminals?

Don't want a predictive policing system that predicts criminality using facial features

- A lot of recent focus on Fairness and Transparency of ML systems

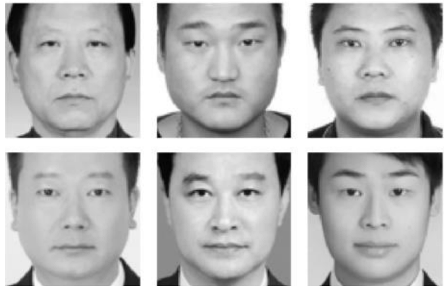
Good ML Systems Should be Fair and Unbiased

- Good ML should not just be about getting high accuracies

- Should also ensure that the ML models are fair and unbiased

An image captioning system should not always assume a specific gender in examples like the above

Don't want a self-driving car that is more likely to hit black people than white people



Criminals?

Not Criminals?

Don't want a predictive policing system that predicts criminality using facial features

- A lot of recent focus on Fairness and Transparency of ML systems

Picture courtesy: Bhargava and Forsyth (2019), https://www.thestranger.com/, Xiaolin Wu and Xi Zhang, "Automated Inference on Criminality Using Face Images"

Looking Back Before We Start: History of ML

1750-1900
- Origins of Bayes Theorem: Thomas Bayes and Pierre-Simon Laplace (~1800)
- Least Squares method: Legendre (1805)

1900-1950
- Early works on PCA (Peason), Factor Analysis (Spearman), CCA (by Hotelling), for exploratory data analysis
- Early works on Discriminant Analysis methods (Fisher) for classification
- McCulloch-Pitts model of the artificial neuron (1943)
- Turing predicted that Machine Learning is possible (1947)
- Design of the "Turing Test" (1950)

1950-1960
- Minsky & Edmonds' neural net machine (SNARC)
- Arthur Samuels' Checkers Player based on Machine Learning
- Rosenblatt's Perceptron algo
- Widrow-Hoff's ADALINE algo
- K-means clustering algo (Lloyd)
- Early origins of Reinforcement Learning (Bellman and others)

1960-1970
- Multi-layer ADALINE (called MADALINE)
- Nearest Neighbors algorithm
- Early works on Genetic Algorithms inspired by natural evolution (John Holland)

1970-1980
- Minsky-Papert showed that Perceptron can't learn XOR function (and nonlinear functions in general). Led to the so-called "AI Winter"
- Automatic Differentiation (later rediscovered as Backprop)

1980-1990
- Fukushima's Neurocognitron model for visual recognition
- Multi-layer Perceptrons (can learn nonlinear functions). Sunny days of AI are back!
- The Backpropagation algorithm to train deep neural nets
- Decision Trees, ID3 (Quinlan)
- NetTalk: Neural nets that can learn to pronounce English words
- Modern Reinforcement Learning

1990-2000
- Neural nets slumber due to lack of compute power
- Support Vector Machines (SVM), Kernel methods, Bayesian methods
- Random Forests, Boosting
- Continued work on neural nets for images, sequences (CNN, LSTM, etc)
- Deep Blue beats Kasparov

2000-2010
- Continued focus on classical statistical and probabilistic models, connections b/w learning & cognition
- Areas like vision, NLP, speech embrace data-driven ML methods as opposed to rule-based methods
- ML based algorithm wins the Netflix Challenge
- Neural nets re-emerged and rebranded as Deep Learning (Hinton, Bengio, LeCun, Ng, and others), thanks to improved training, GPUs

2010-2020
- Watson beats Humans in Jeopardy!
- Advances in ML/deep learning result in breakthroughs in areas like vision, NLP, speech, etc.
- Software frameworks (e.g., Tensorflow, PyTorch, ease implementing ML algos)
- Drones, self-driving cars, etc.
- **Human-like text generators (GPT-3)**
- A lot of industry/media focus, excitement, and hype
- Focus on Fairness, Accountability, and Transparency in ML algorithms

Road Ahead

Another **AI Winter** ?

Next Class

Next Class

- Various Flavors of ML problems

Next Class

- Various Flavors of ML problems

- Data and features

Next Class

- Various Flavors of ML problems
- Data and features
- Basic mathematical operations on data and features