Applied Deep Learning-ID6001W

Ganapathy Krishnamurthi

IIT-Madras

Applied Deep Learning - Syllabus

1. Introduction to ANNs: Linear Regression, Logistic Regression, Perceptron, MLP, Back propagation (4 Hours)
2. Deep neural networks: Universal Approximation Theorem (UAT), Feed Forward networks, Fully connected network, Layer-wise learning, SGD, RMSProp, Adam, dropout, batch normalization (6 Hours)
3. Convolutional neural networks (CNN): Convolution operation, CNN architecture, Object Detection, Semantic Segmentation(10 Hours)
4. Sequence learning: RNNs, BPTT, LSTM, GRUs (10 Hours)
5. Autoencoders: PCA and its nonlinear variant, Regularised AE, Sparse AE, manifold learning with AE, and applications (4 hours)
6. Advanced topics: Bayesian Networks, Graph Neural networks, Generative neural networks (6 hours)

Grading Scheme

1. Six Programming Assignments $\longrightarrow$ 40%
2. One Mid-term $\longrightarrow$ 20%
3. Final exam $\longrightarrow$ 40%

List of topics to be covered

Series of classes on Deep Learning foundations

List of topics to be covered

Series of classes on Deep Learning foundations

1. Linear Regression
    - Linear Model
    - Gradient Descent for Parameter estimation

List of topics to be covered

Series of classes on Deep Learning foundations

1. Linear Regression
   - Linear Model
   - Gradient Descent for Parameter estimation
2. Logistic Regression-Classification Problems

List of topics to be covered

Series of classes on Deep Learning foundations

1. Linear Regression
    - Linear Model
    - Gradient Descent for Parameter estimation
2. Logistic Regression-Classification Problems
3. Transition to MLPs

List of topics to be covered

Series of classes on Deep Learning foundations

1. Linear Regression
   - Linear Model
   - Gradient Descent for Parameter estimation
2. Logistic Regression-Classification Problems
3. Transition to MLPs
4. Artificial Neural Networks (ANNs)
   - Forward Pass
   - Backward pass or Backpropagation
   - Optimization for ANNs

Introductory Lecture

- An overview of Artificial Intelligence and Machine Learning

Introductory Lecture

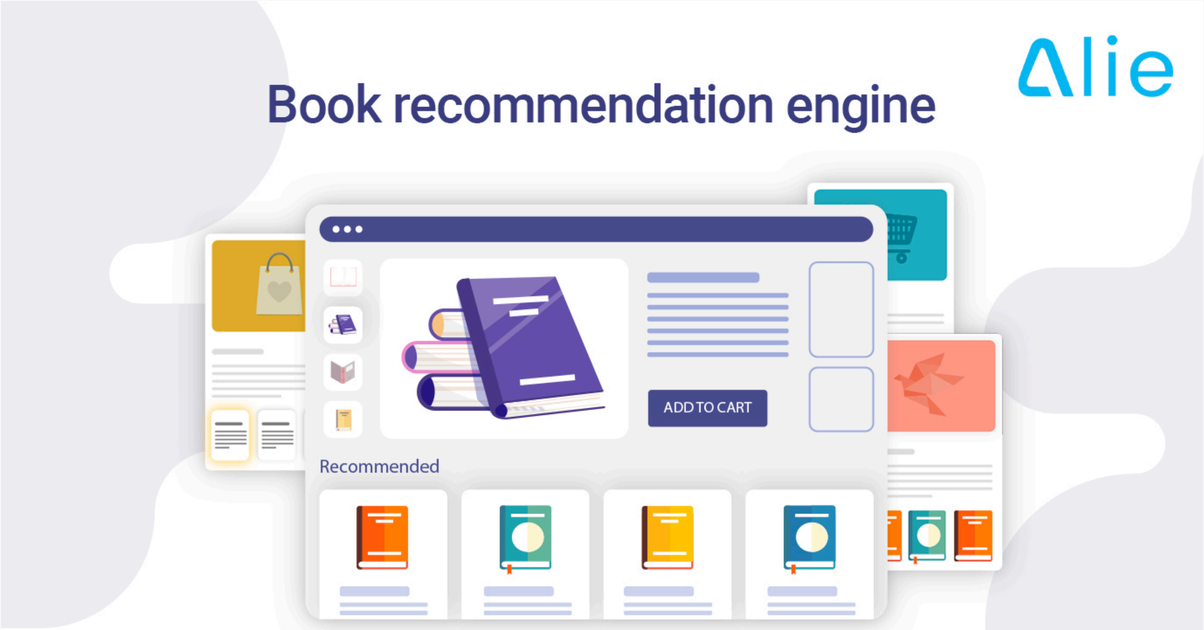
- An overview of Artificial Intelligence and Machine Learning
- What makes Machine Learning in general and Deep Learning in specific work?

Introductory Lecture

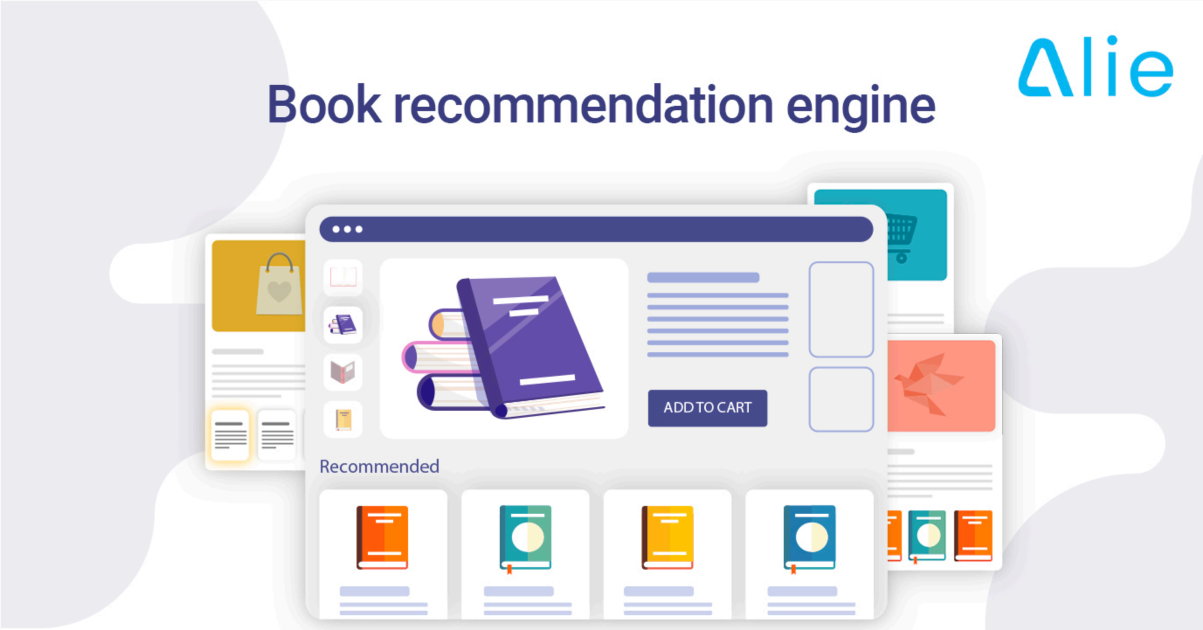
- An overview of Artificial Intelligence and Machine Learning
- What makes Machine Learning in general and Deep Learning in specific work?
- An introduction to Deep Learning
    - Artificial Neural Networks

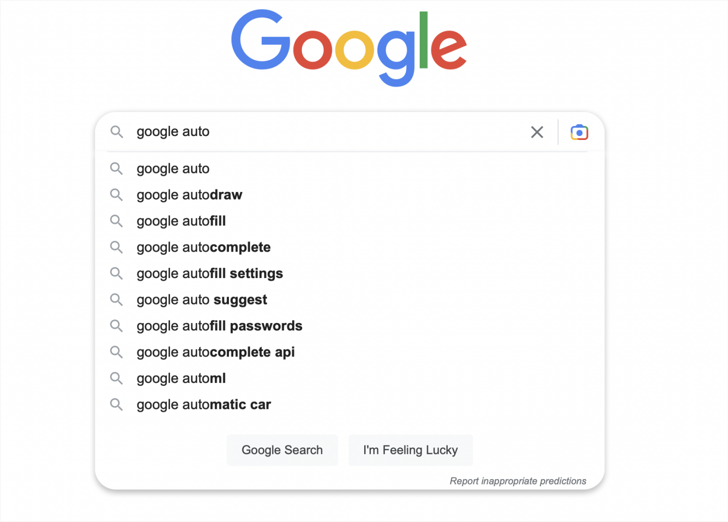
Overview of AI and ML

What makes these possible?

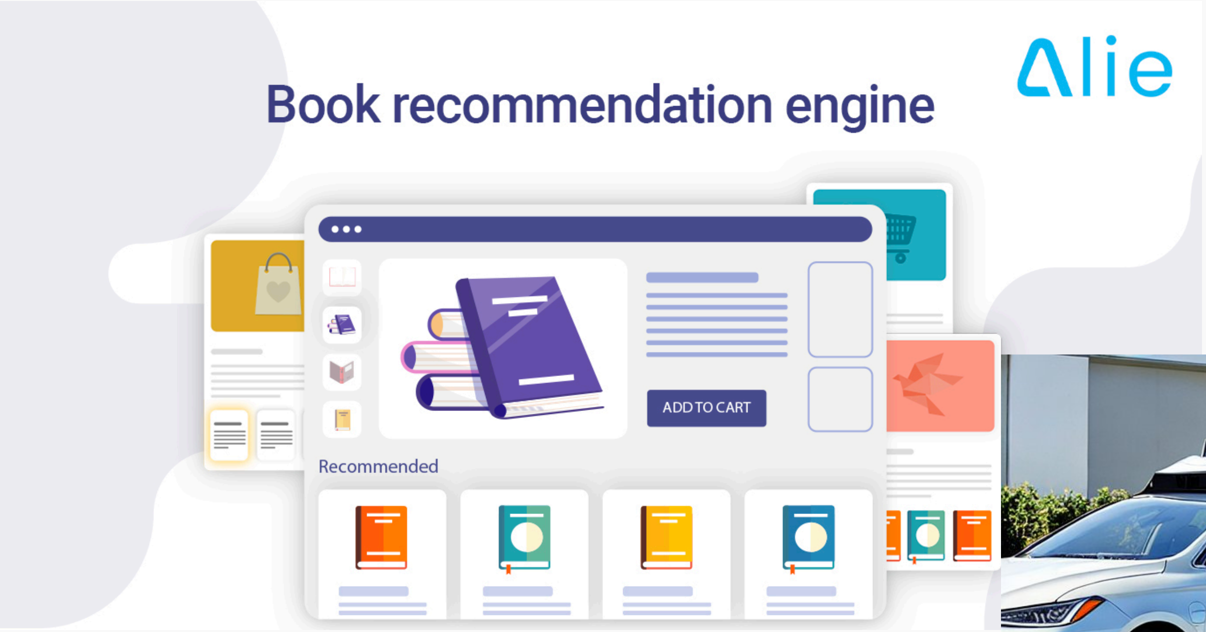


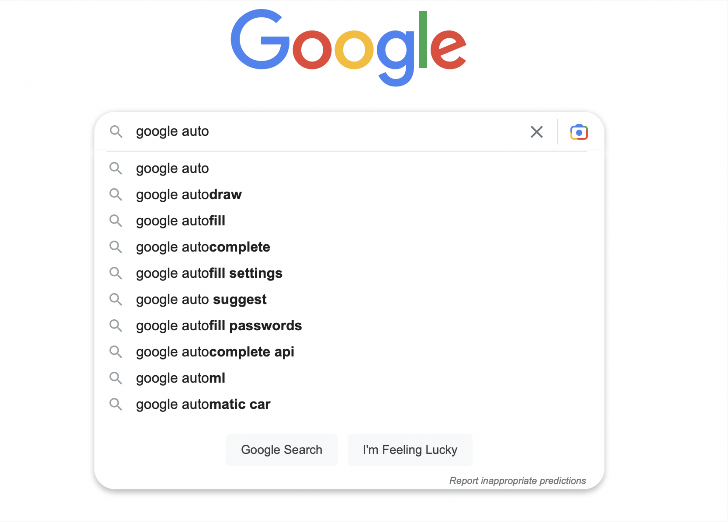
What makes these possible?

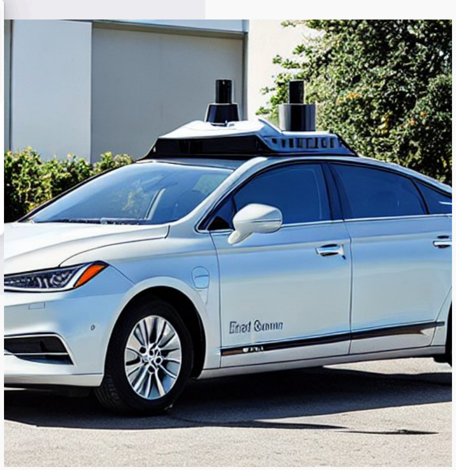




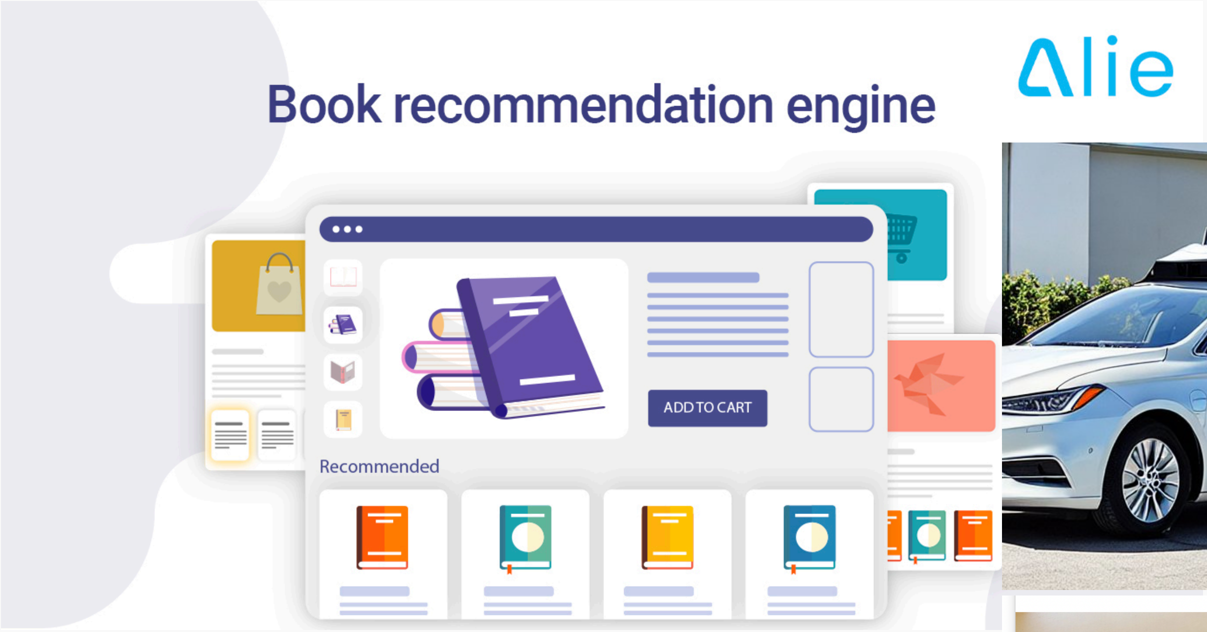
What makes these possible?

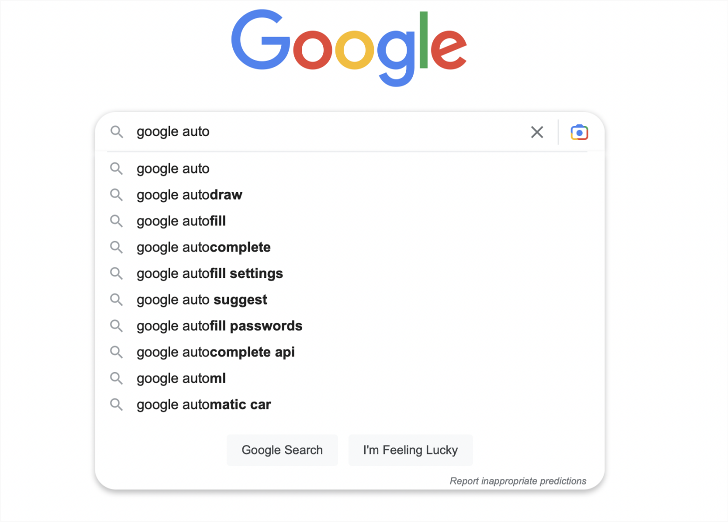




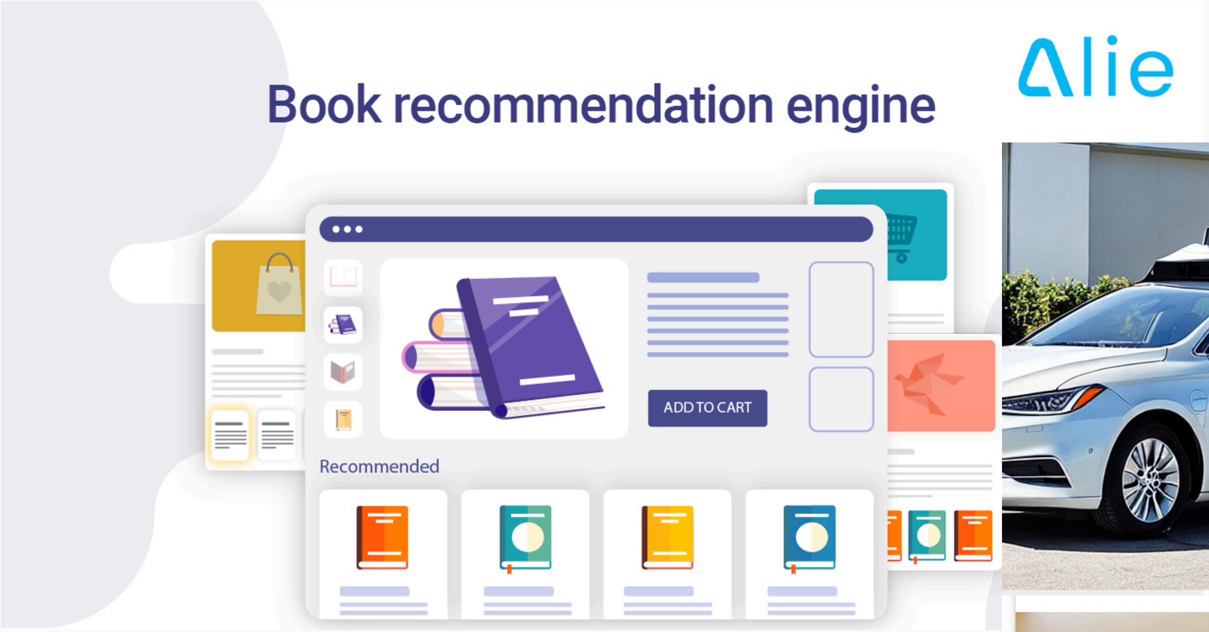


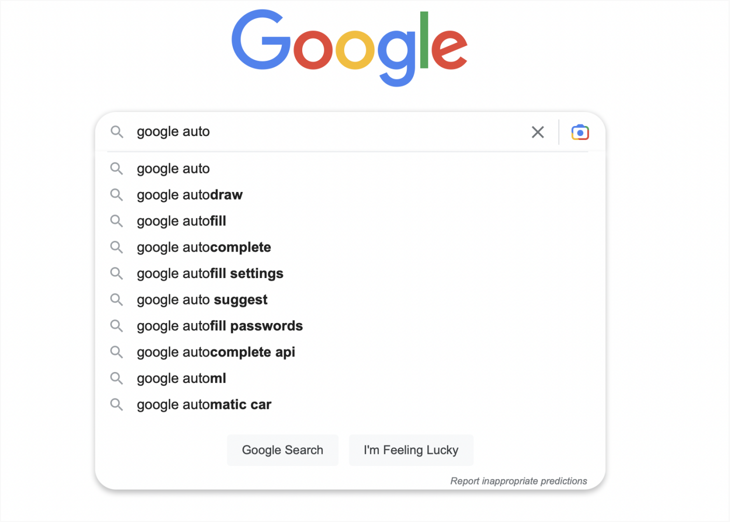
What makes these possible?





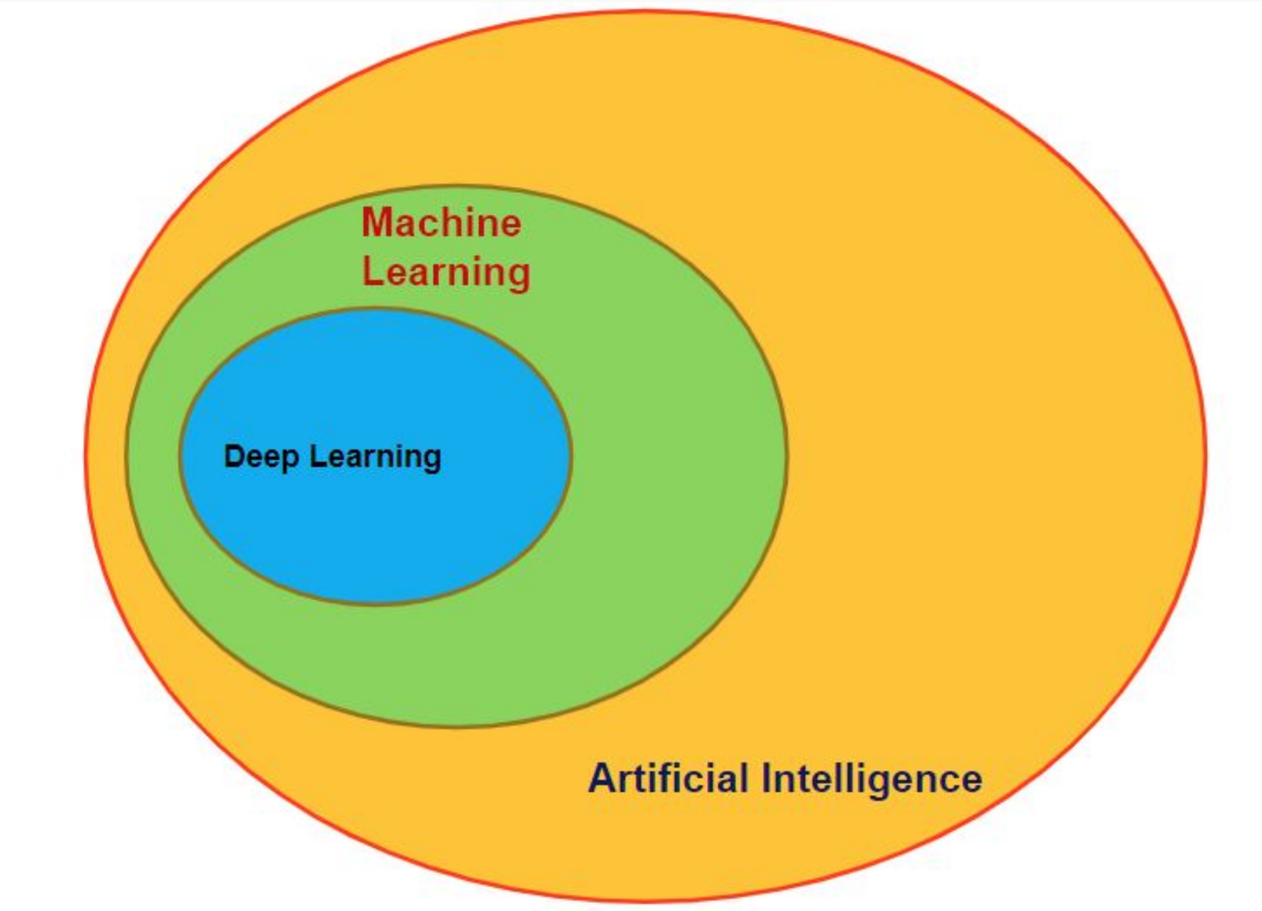
What makes these possible?





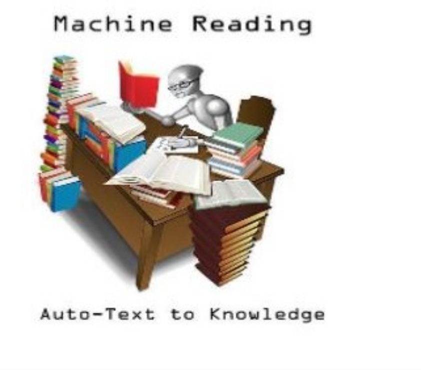
Simplistic Definition - Artificial Intelligence and Machine Learning aim to replicate activities requiring human cognition

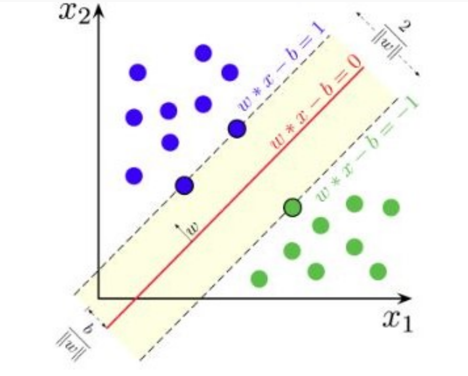
The AI Venn diagram



Adapted from Deep Learning, Goodfellow et al (2016)

What is Machine Learning?





- Simple Definition – Using Data to answer questions
- Study of computer algorithms
  - that improve automatically
  - through experience.
- Formally, A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.

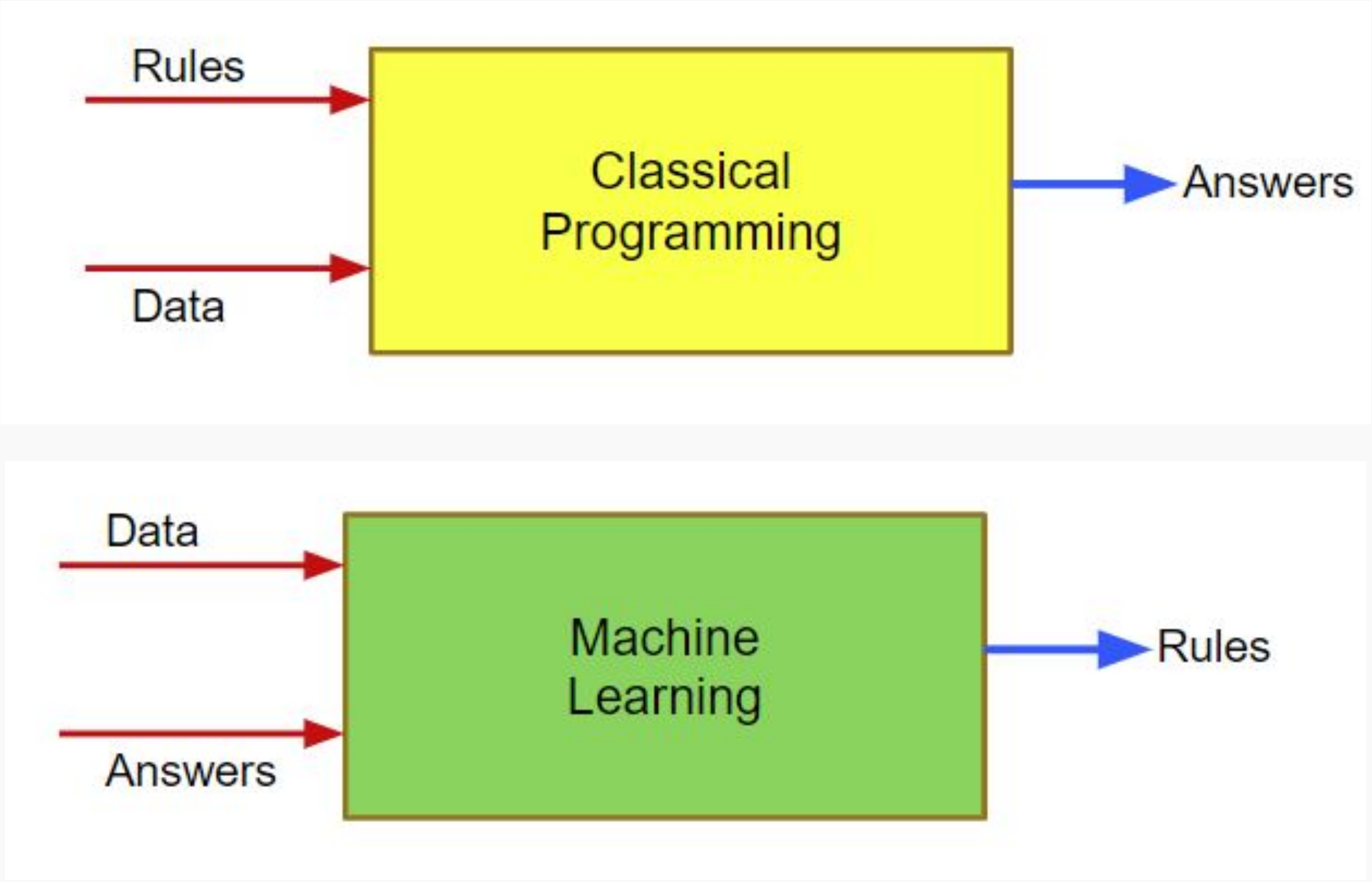
---

https://upload.wikimedia.org/wikipedia/commons/thumb/6/6a/Maching_Reading_Robot_Auto-Text_to_Knowledge.jpg/463px-Maching_Reading_Robot_Auto-Text_to_Knowledge.jpg

Yufeng Guo –https://www.youtube.com/watch?v=HcqpanDadyQ

https://upload.wikimedia.org/wikipedia/commons/thumb/7/72/SVM_margin.png/1280px-SVM_margin.png

The Machine Learning Paradigm



A spam example

A spam example

Supply from alfasigma

REPLY   REPLY ALL   FORWARD

Zofia Antoni <info@patlimited.com>

Sat 6/27/2020 5:42 PM

Mark as read

- To help protect your privacy, some content in this message has been blocked. To re-enable the blocked features, click here.
- To always show content from this sender, click here.

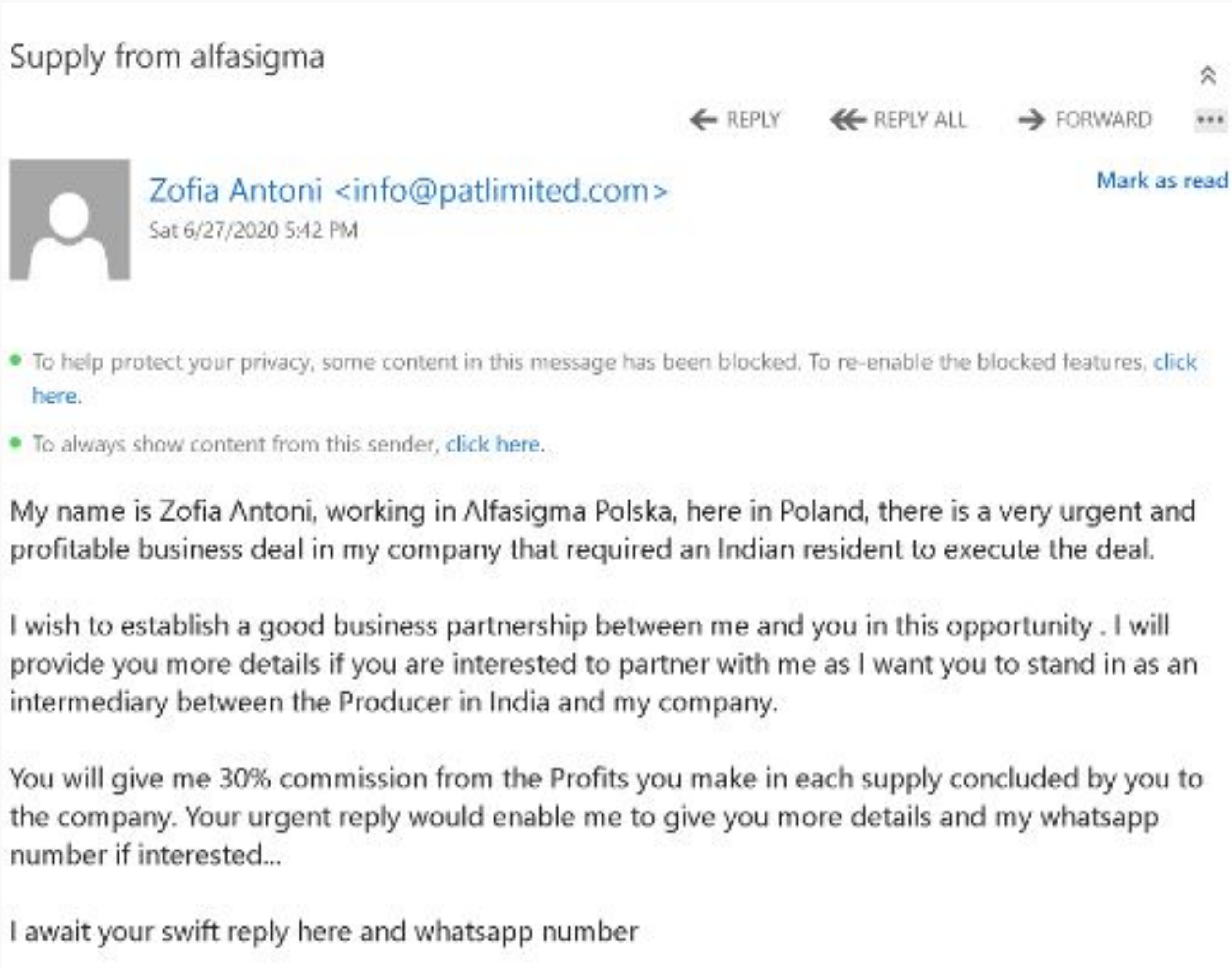
My name is Zofia Antoni, working in Alfasigma Polska, here in Poland, there is a very urgent and profitable business deal in my company that required an Indian resident to execute the deal.

I wish to establish a good business partnership between me and you in this opportunity . I will provide you more details if you are interested to partner with me as I want you to stand in as an intermediary between the Producer in India and my company.

You will give me 30% commission from the Profits you make in each supply concluded by you to the company. Your urgent reply would enable me to give you more details and my whatsapp number if interested...

I await your swift reply here and whatsapp number

A spam example



Supply from alfasigma

REPLY REPLY ALL FORWARD

Zofia Antoni <info@patlimited.com>

Sat 6/27/2020 5:42 PM

Mark as read

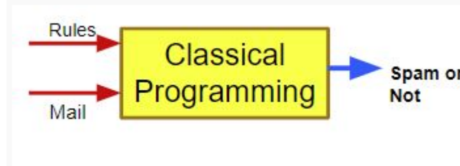
- To help protect your privacy, some content in this message has been blocked. To re-enable the blocked features, click here.
- To always show content from this sender, click here.

My name is Zofia Antoni, working in Alfasigma Polska, here in Poland, there is a very urgent and profitable business deal in my company that required an Indian resident to execute the deal.

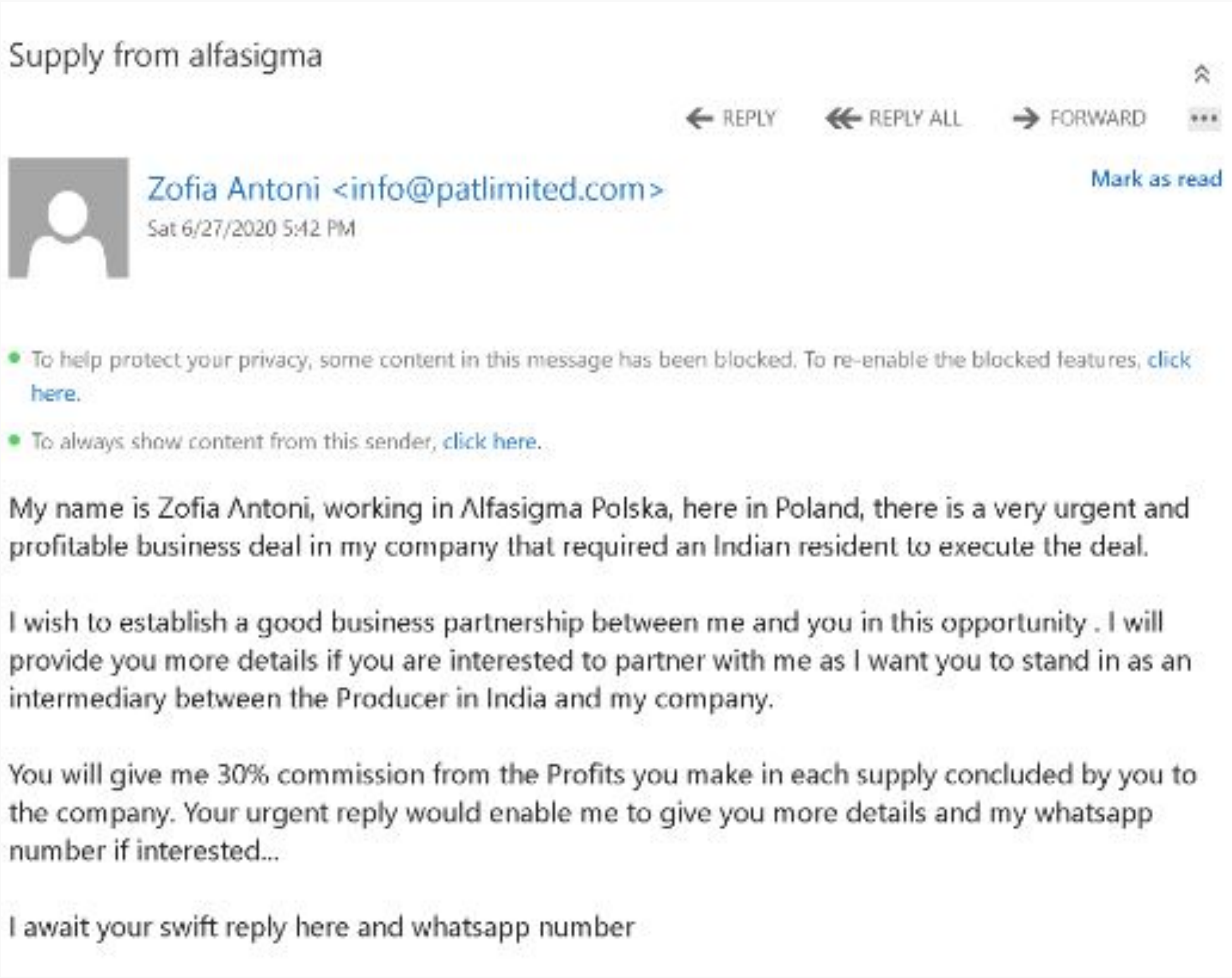
I wish to establish a good business partnership between me and you in this opportunity . I will provide you more details if you are interested to partner with me as I want you to stand in as an intermediary between the Producer in India and my company.

You will give me 30% commission from the Profits you make in each supply concluded by you to the company. Your urgent reply would enable me to give you more details and my whatsapp number if interested...

I await your swift reply here and whatsapp number



A spam example


Supply from alfasigma

REPLY REPLY ALL FORWARD

Zofia Antoni <info@patlimited.com>
Sat 6/27/2020 5:42 PM

Mark as read

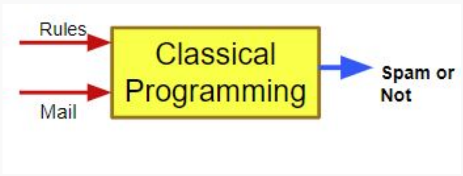
- To help protect your privacy, some content in this message has been blocked. To re-enable the blocked features, click here.
- To always show content from this sender, click here.

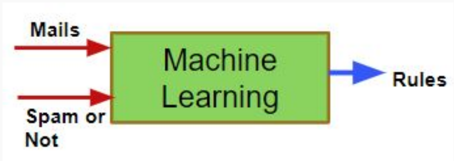
My name is Zofia Antoni, working in Alfasigma Polska, here in Poland, there is a very urgent and profitable business deal in my company that required an Indian resident to execute the deal.

I wish to establish a good business partnership between me and you in this opportunity . I will provide you more details if you are interested to partner with me as I want you to stand in as an intermediary between the Producer in India and my company.

You will give me 30% commission from the Profits you make in each supply concluded by you to the company. Your urgent reply would enable me to give you more details and my whatsapp number if interested...

I await your swift reply here and whatsapp number



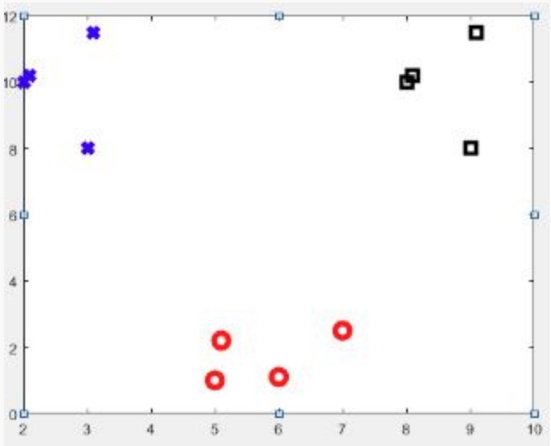


When is Machine Learning useful?

- When experts are unable to explain their expertise
    - Image recognition
    - Speech recognition
    - Driving a car
- When past Human expertise does not exist
    - Hazardous environments – Navigating on Mars
- Solution needs to be adapted to particular cases
    - User biometrics
    - Patient specific treatments
    - Customer specific advertisements or offers

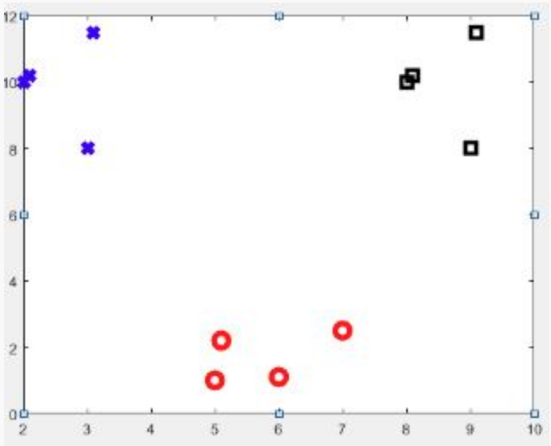
Types of learning approaches

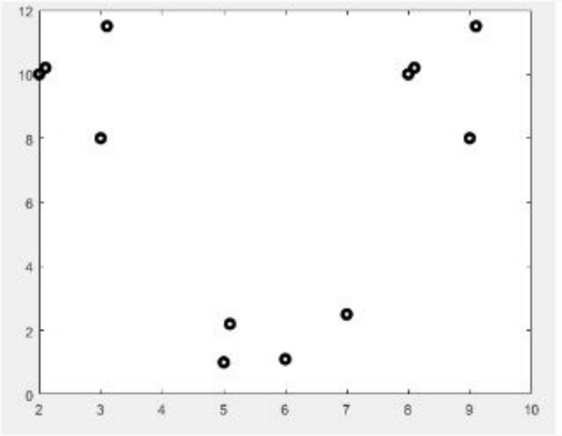
- Supervised Learning
  - Data labeled by human experts
  - Labeling images
  - Speech recognition
  - OCR



Types of learning approaches

- Supervised Learning
  - Data labeled by human experts
  - Labeling images
  - Speech recognition
  - OCR
- Unsupervised Learning
  - Unlabeled data
  - Grouping customers
  - Detecting new diseases
  - Anomaly detection





Seven Steps in Machine Learning

Seven Steps in Machine Learning

1. Gathering Data

Seven Steps in Machine Learning

1. Gathering Data
    - Deciding what “data” means is part of the problem

Seven Steps in Machine Learning

1. Gathering Data
    - Deciding what “data” means is part of the problem
2. Preparing Data

Seven Steps in Machine Learning

1. Gathering Data
    - Deciding what "data" means is part of the problem
2. Preparing Data
    - E.g : Cleaning data, Ensuring that there is no bias

Seven Steps in Machine Learning

1. Gathering Data
   - Deciding what “data” means is part of the problem
2. Preparing Data
   - E.g : Cleaning data, Ensuring that there is no bias
3. Choosing a Model/Algorithm

Seven Steps in Machine Learning

1. Gathering Data
   - Deciding what “data” means is part of the problem
2. Preparing Data
   - E.g : Cleaning data, Ensuring that there is no bias
3. Choosing a Model/Algorithm
   - Examples – Random Forest, ANNs, Hidden Markov Models,etc

Seven Steps in Machine Learning

1. Gathering Data
    - Deciding what “data” means is part of the problem
2. Preparing Data
    - E.g : Cleaning data, Ensuring that there is no bias
3. Choosing a Model/Algorithm
    - Examples – Random Forest, ANNs, Hidden Markov Models,etc
4. Training

Seven Steps in Machine Learning

1. Gathering Data
   - Deciding what “data” means is part of the problem
2. Preparing Data
   - E.g : Cleaning data, Ensuring that there is no bias
3. Choosing a Model/Algorithm
   - Examples – Random Forest, ANNs, Hidden Markov Models,etc
4. Training
   - Using data to determine model parameters

Seven Steps in Machine Learning

1. Gathering Data
   - Deciding what “data” means is part of the problem
2. Preparing Data
   - E.g : Cleaning data, Ensuring that there is no bias
3. Choosing a Model/Algorithm
   - Examples – Random Forest, ANNs, Hidden Markov Models,etc
4. Training
   - Using data to determine model parameters
5. Evaluation – How well did we do?

Seven Steps in Machine Learning

1. Gathering Data
    - Deciding what "data" means is part of the problem
2. Preparing Data
    - E.g : Cleaning data, Ensuring that there is no bias
3. Choosing a Model/Algorithm
    - Examples – Random Forest, ANNs, Hidden Markov Models,etc
4. Training
    - Using data to determine model parameters
5. Evaluation – How well did we do?
6. Hyperparameter Tuning

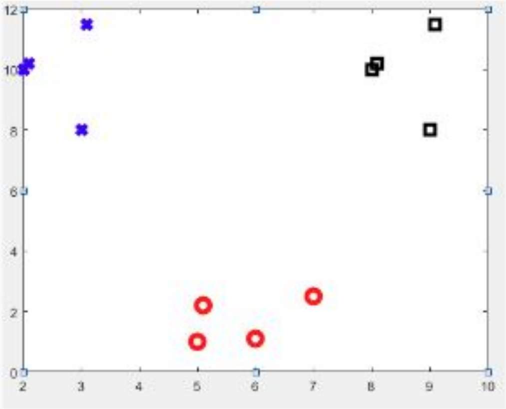
Seven Steps in Machine Learning

1. Gathering Data
    - Deciding what "data" means is part of the problem
2. Preparing Data
    - E.g : Cleaning data, Ensuring that there is no bias
3. Choosing a Model/Algorithm
    - Examples – Random Forest, ANNs, Hidden Markov Models,etc
4. Training
    - Using data to determine model parameters
5. Evaluation – How well did we do?
6. Hyperparameter Tuning
7. Prediction

Two problems in Supervised Learning

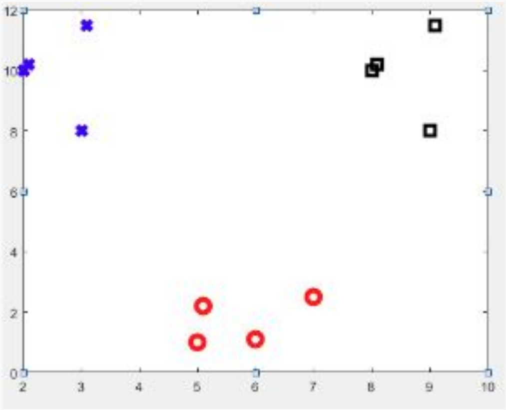
Classification	Regression
Split it	Fit it
Discrete or Categorical data.	Real number data
Has category associated	Has associated number
Example : Tumour classification	Example : Tumour classification

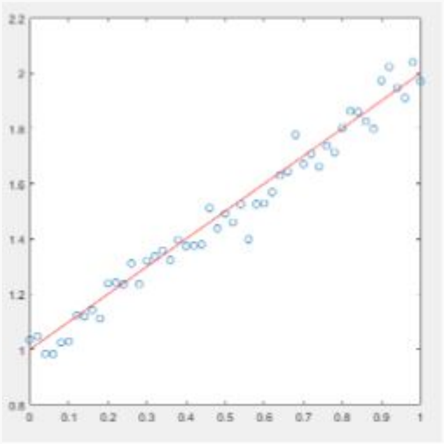
Two problems in Supervised Learning



Classification	Regression
Split it	Fit it
Discrete or Categorical data.	Real number data
Has category associated	Has associated number
Example : Tumour classification	Example : Tumour classification

Two problems in Supervised Learning





Classification	Regression
Split it	Fit it
Discrete or Categorical data.	Real number data
Has category associated	Has associated number
Example : Tumour classification	Example : Tumour classification

Other types of learning approaches

- Generative approaches
  - Creating new data that is "like" given data
  - Generally included in unsupervised learning
- Semi-supervised learning
  - Small amount of labeled data available along with unlabeled data
- Self-supervised learning
  - Implicit labels are extracted from data using heuristics
- Reinforcement learning
  - Action strategy chosen based on temporally delayed rewards.
  - Useful in strategic decisions . Example : Chess, Games, Investment, etc

---

The distinction between the various types of learning is often blurred

The Learning Paradigm

A fundamental “trick” in most of ML

- All problems are data, all solutions are functions/maps
- Cognitive tasks – Humans get sensory inputs as qualia

A fundamental "trick" in most of ML

- All problems are data, all solutions are functions/maps
- Cognitive tasks – Humans get sensory inputs as qualia
    - We must convert these qualitative inputs into numbers – Input Vectors
    - Similarly, outputs that humans give must also be converted into numbers – Output/Target vectors

A fundamental "trick" in most of ML

- All problems are data, all solutions are functions/maps
- Cognitive tasks – Humans get sensory inputs as qualia
  - We must convert these qualitative inputs into numbers – Input Vectors
  - Similarly, outputs that humans give must also be converted into numbers – Output/Target vectors
- Determining appropriate inputs and outputs for a machine learning task is an essential part of the process
- Often the "Learning Task" is learning the mapping from input to output.

Example – From image to vector

https://upload.wikimedia.org/wikipedia/commons/2/27/MnistExamples.png

Example – From image to vector

```
>> india_data=imread('india.png');
>> size(india_data)

ans =

   600    538      3
```

https://upload.wikimedia.org/wikipedia/commons/2/27/MnistExamples.png

https://upload.wikimedia.org/wikipedia/commons/thumb/0/05/India_geo_stub.svg/538px-India_geo_stub.svg.png

Example – From image to vector

```
>> india_data=imread('india.png');
>> size(india_data)

ans =

   600   538     3
```

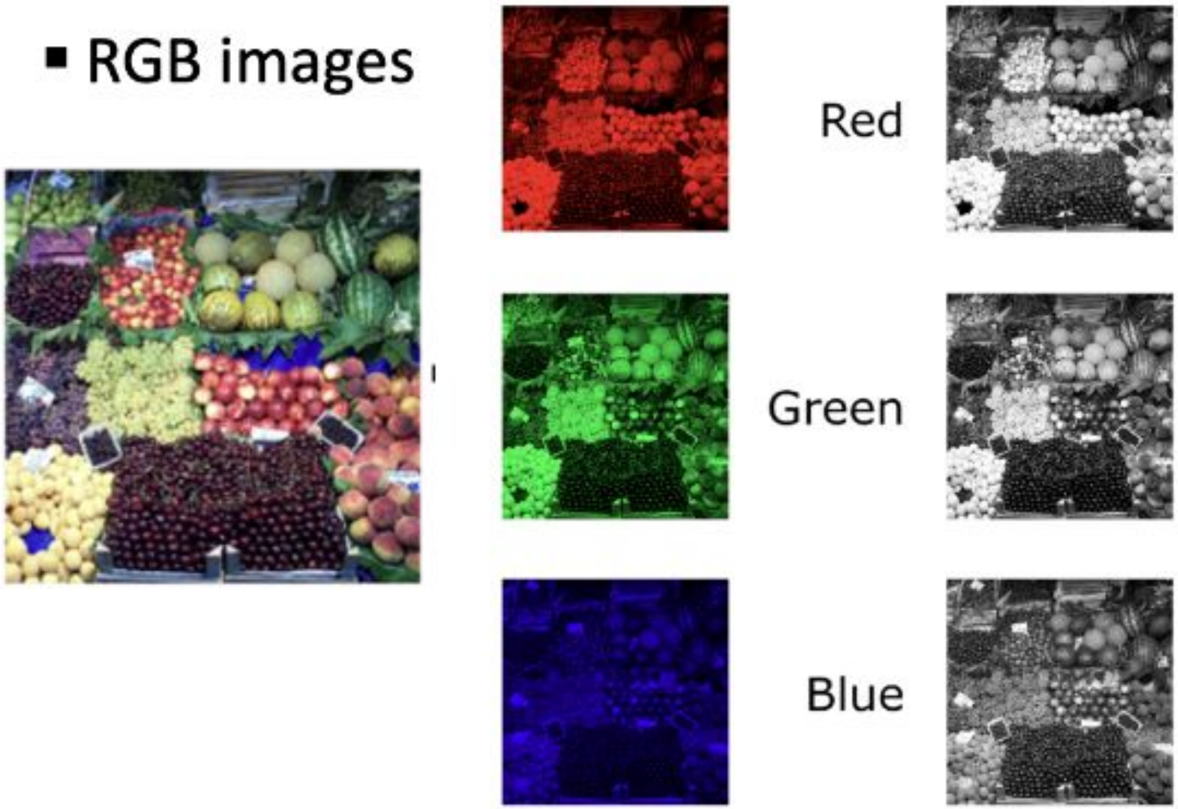
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	240
255	255	255	255	255	255	255	255	255	255	231	80
255	255	255	255	255	255	255	255	255	242	78	51
255	255	255	255	255	255	255	255	255	131	51	51
255	255	255	255	255	255	255	255	225	54	51	51
82	140	209	254	255	255	255	255	108	51	51	51
51	51	51	89	180	227	209	120	51	51	51	51
51	51	51	51	51	51	51	51	51	51	51	51
51	51	51	51	51	51	51	51	51	51	51	51
51	51	51	51	51	51	51	51	51	51	51	51
51	51	51	51	51	51	51	51	51	51	51	51
51	51	51	51	51	51	51	51	51	51	51	51
51	51	51	51	51	51	51	51	51	51	51	51

https://upload.wikimedia.org/wikipedia/commons/2/27/MnistExamples.png

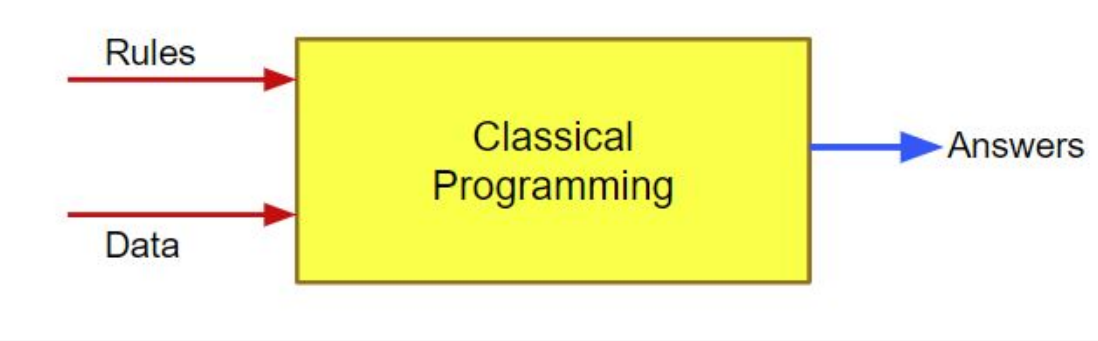
https://upload.wikimedia.org/wikipedia/commons/thumb/0/05/India_geo_stub.svg/538px-India_geo_stub.svg.png

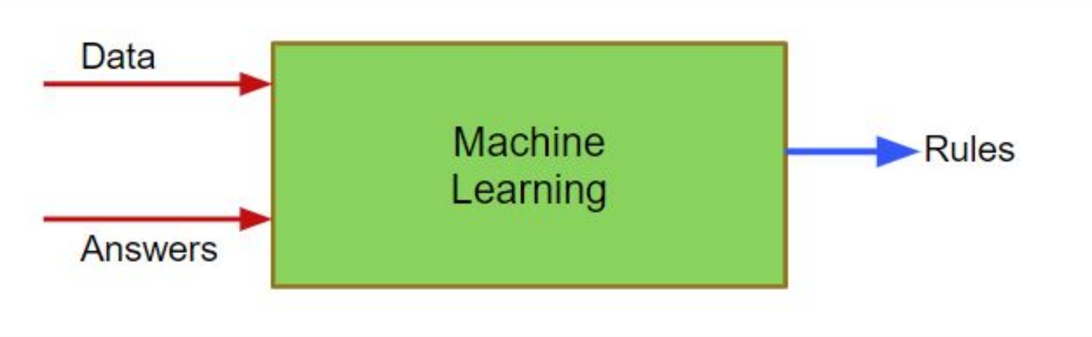
Image parametrization

- RGB images

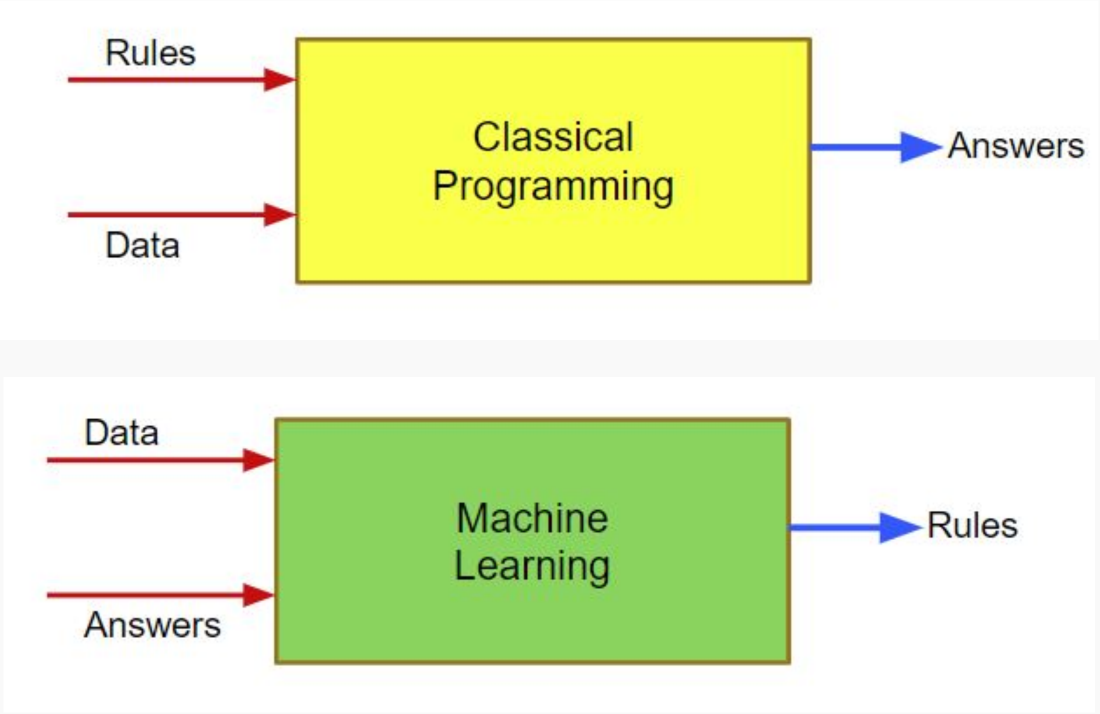


Recall – The Machine Learning Paradigm





Recall – The Machine Learning Paradigm



Let us now look at the Machine Learning Paradigm in some more detail

Forward Modeling

Forward Modeling

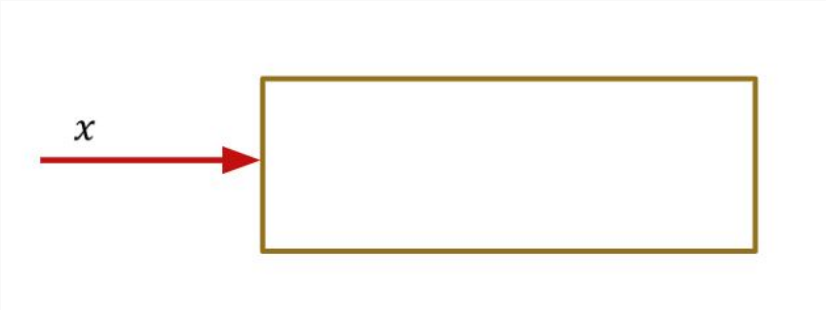
- A model or hypothesis is simply an educated guess at what the relationship between input and output is.
- As mentioned before, it has two pieces
  - Form of the function - Linear, Quadratic, Exponential, etc
  - Parameters of the function
- We sometimes use the notation $y = f(x; w)$
  - Given x and a choice of w, we can find a corresponding y
- The function f going from x to y is called the forward model
  - The process is called the forward pass or inference $->$ Given x, w finding y

Forward Modeling

- A model or hypothesis is simply an educated guess at what the relationship between input and output is.
- As mentioned before, it has two pieces
  - Form of the function - Linear, Quadratic, Exponential, etc
  - Parameters of the function
- We sometimes use the notation y = f(x; w)
  - Given x and a choice of w, we can find a corresponding y
- The function f going from x to y is called the forward model
  - The process is called the forward pass or inference $->$ Given x, w finding y

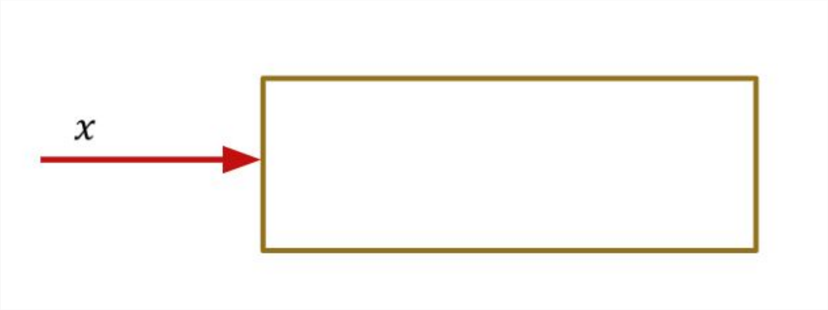
Learning the parameters via feedback

Learning the parameters via feedback



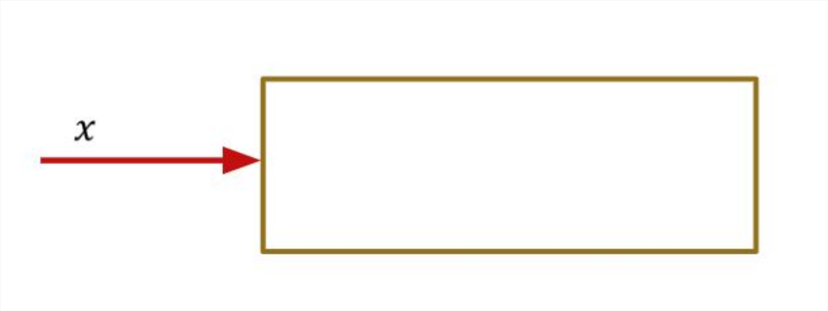
- To learn the parameters (given x, y find w), we follow this paradigm

Learning the parameters via feedback



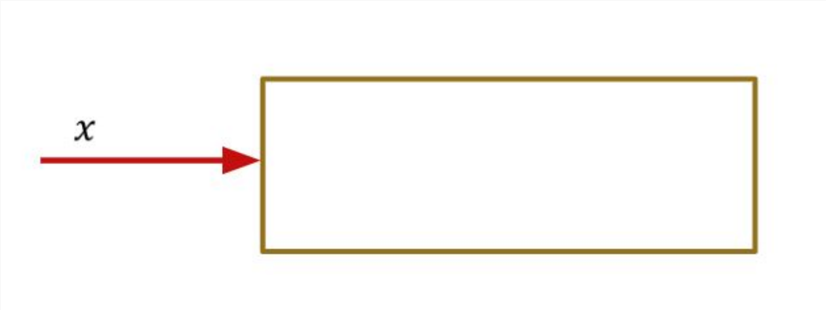
- To learn the parameters (given x, y find w), we follow this paradigm
- Collect lots of data pairs (Input Vector, Output Vector) = (x, y)

Learning the parameters via feedback



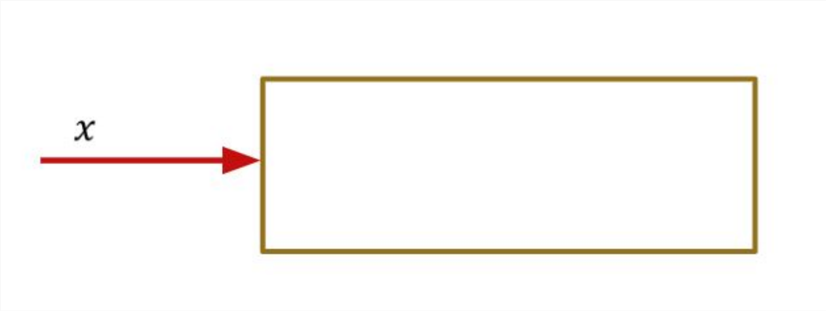
- To learn the parameters (given x, y find w), we follow this paradigm
- Collect lots of data pairs (Input Vector, Output Vector) = (x, y)
- Guess for the form of the hypothesis function h(x; w)
    - Example: $h(x; w) = w_0 + w_1 x_1 + w_2 x_2$

Learning the parameters via feedback



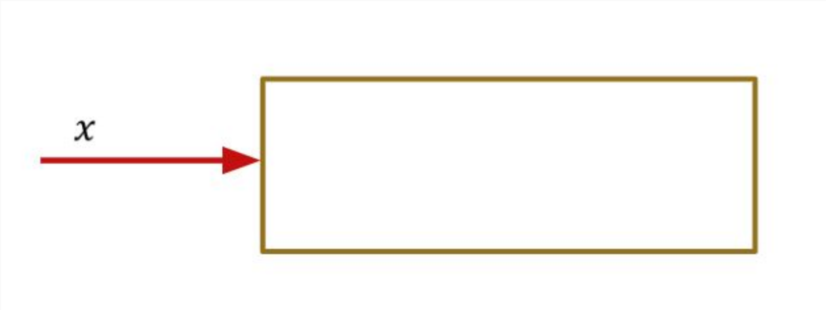
- To learn the parameters (given x, y find w), we follow this paradigm
- Collect lots of data pairs (Input Vector, Output Vector) = (x, y)
- Guess for the form of the hypothesis function h(x; w)
  - Example: $h(x; w) = w_0 + w_1 x_1 + w_2 x_2$
- For an arbitrary guess for w
  - We will get some $\hat{y} = h(x; w)$ <u>which will not match the ground truth y</u>.

Learning the parameters via feedback



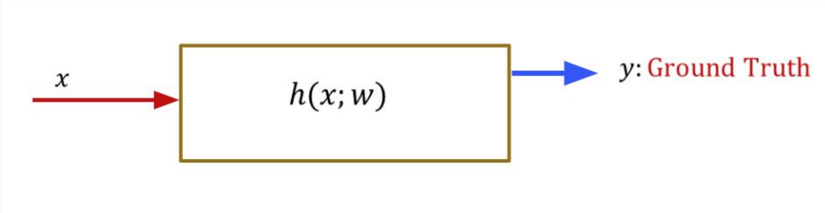
- To learn the parameters (given x, y find w), we follow this paradigm
- Collect lots of data pairs (Input Vector, Output Vector) = (x, y)
- Guess for the form of the hypothesis function h(x; w)
  - Example: $h(x; w) = w_0 + w_1 x_1 + w_2 x_2$
- For an arbitrary guess for w
  - We will get some $\hat{y} = h(x; w)$ which will not match the ground truth y.
- Define a cost function $J(y, \hat{y}(w))$ depending on the difference

Learning the parameters via feedback



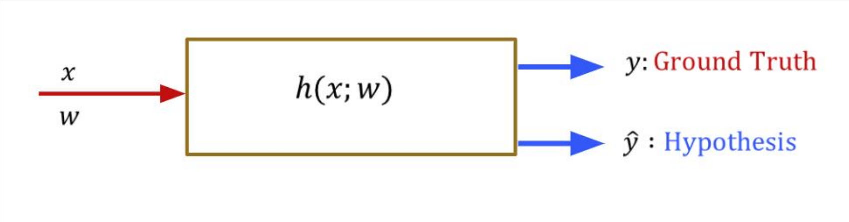
- To learn the parameters (given x, y find w), we follow this paradigm
- Collect lots of data pairs (Input Vector, Output Vector) = (x, y)
- Guess for the form of the hypothesis function h(x; w)
  - Example: $h(x; w) = w_0 + w_1 x_1 + w_2 x_2$
- For an arbitrary guess for w
  - We will get some $\hat{y} = h(x; w)$ which will not match the ground truth y.
- Define a cost function $J(y, \hat{y}(w))$ depending on the difference
- Find optimal w by minimizing $J(w)$
  - By using some optimization procedure such as Gradient Descent

Learning the parameters via feedback



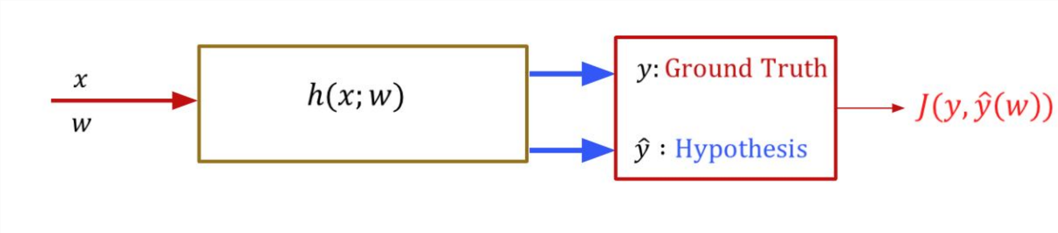
- To learn the parameters (given x, y find w), we follow this paradigm
- Collect lots of data pairs (Input Vector, Output Vector) $= (x, y)$
- Guess for the form of the hypothesis function h(x; w)
  - Example: $h(x; w) = w_0 + w_1 x_1 + w_2 x_2$
- For an arbitrary guess for w
  - We will get some $\hat{y} = h(x; w)$ which will not match the ground truth y.
- Define a cost function $J(y, \hat{y}(w))$ depending on the difference
- Find optimal w by minimizing $J(w)$
  - By using some optimization procedure such as Gradient Descent

Learning the parameters via feedback



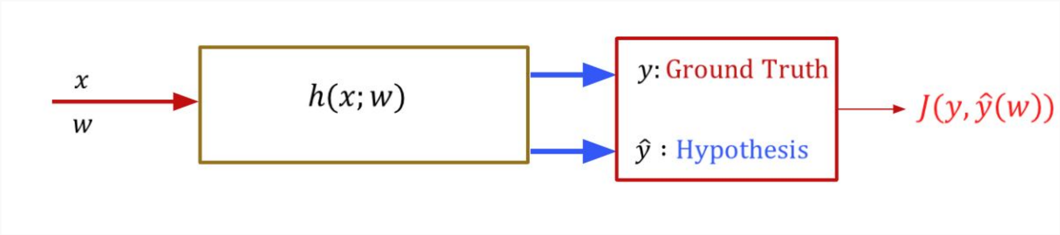
- To learn the parameters (given x, y find w), we follow this paradigm
- Collect lots of data pairs (Input Vector, Output Vector) $= (x, y)$
- Guess for the form of the hypothesis function h(x; w)
    - Example: $h(x; w) = w_0 + w_1 x_1 + w_2 x_2$
- For an arbitrary guess for w
    - We will get some $\hat{y} = h(x; w)$ <u>which will not match the ground truth y</u>.
- Define a cost function $J(y, \hat{y}(w))$ depending on the difference
- Find optimal w by minimizing $J(w)$
    - By using some optimization procedure such as Gradient Descent

Learning the parameters via feedback



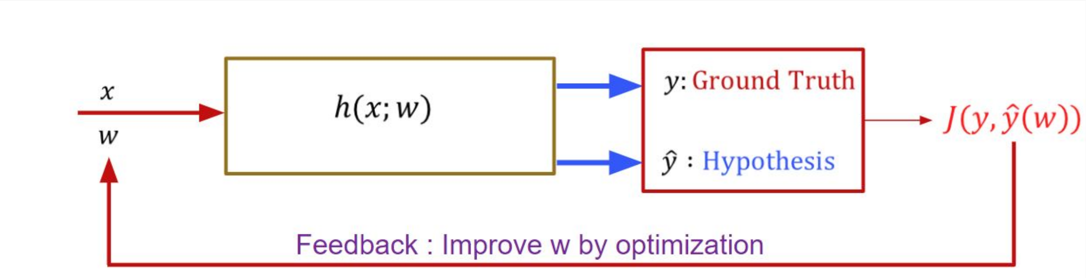
- To learn the parameters (given x, y find w), we follow this paradigm
- Collect lots of data pairs (Input Vector, Output Vector) $= (x, y)$
- Guess for the form of the hypothesis function h(x; w)
  - Example: $h(x; w) = w_0 + w_1 x_1 + w_2 x_2$
- For an arbitrary guess for w
  - We will get some $\hat{y} = h(x; w)$ <u>which will not match the ground truth y</u>.
- Define a cost function $J(y, \hat{y}(w))$ depending on the difference
- Find optimal w by minimizing $J(w)$
  - By using some optimization procedure such as Gradient Descent

Learning the parameters via feedback



- To learn the parameters (given x, y find w), we follow this paradigm
- Collect lots of data pairs (Input Vector, Output Vector) $= (x, y)$
- Guess for the form of the hypothesis function h(x; w)
    - Example: $h(x; w) = w_0 + w_1 x_1 + w_2 x_2$
- For an arbitrary guess for w
    - We will get some $\hat{y} = h(x; w)$ which will not match the ground truth y.
- Define a cost function $J(y, \hat{y}(w))$ depending on the difference
- Find optimal w by minimizing $J(w) \rightarrow$ Feedback Process
    - By using some optimization procedure such as Gradient Descent

Learning the parameters via feedback



- To learn the parameters (given x, y find w), we follow this paradigm
- Collect lots of data pairs (Input Vector, Output Vector) $= (x, y)$
- Guess for the form of the hypothesis function h(x; w)
    - Example: $h(x; w) = w_0 + w_1 x_1 + w_2 x_2$
- For an arbitrary guess for w
    - We will get some $\hat{y} = h(x; w)$ which will not match the ground truth y.
- Define a cost function $J(y, \hat{y}(w))$ depending on the difference
- Find optimal w by minimizing $J(w)$ $\rightarrow$ Feedback Process
    - By using some optimization procedure such as Gradient Descent

Summary

Two main ideas

1. What enables breadth of application?

   By converting even qualitative problems into data problems and treating every rule which we wish to learn as a mathematical function.

Summary

Two main ideas

1. What enables breadth of application?

   By converting even qualitative problems into data problems and treating every rule which we wish to learn as a mathematical function.

$$\textbf{Input} \xrightarrow{\text{Function}} \textbf{Output}$$

Summary

Two main ideas

1. What enables breadth of application?

   By converting even qualitative problems into data problems and treating every rule which we wish to learn as a mathematical function.

   $$\textbf{Input} \xrightarrow{\text{Function}} \textbf{Output}$$

2. How does (supervised) learning happen?

   1. We collect data as (Input, Output) pairs.
   2. We provide a hypothesis as a form of the function connecting the inputs to the outputs
   3. Learn (improve) the parameters based on mathematical feedback from data via loss function.

Machine Learning Models

Main ideas

Two main ideas in this session

1. Demonstration of the learning paradigm.

   We will see how to apply the learning paradigm on a very simple model – the linear model. This will give us the chance to see some details of the training process.

Main ideas

Two main ideas in this session

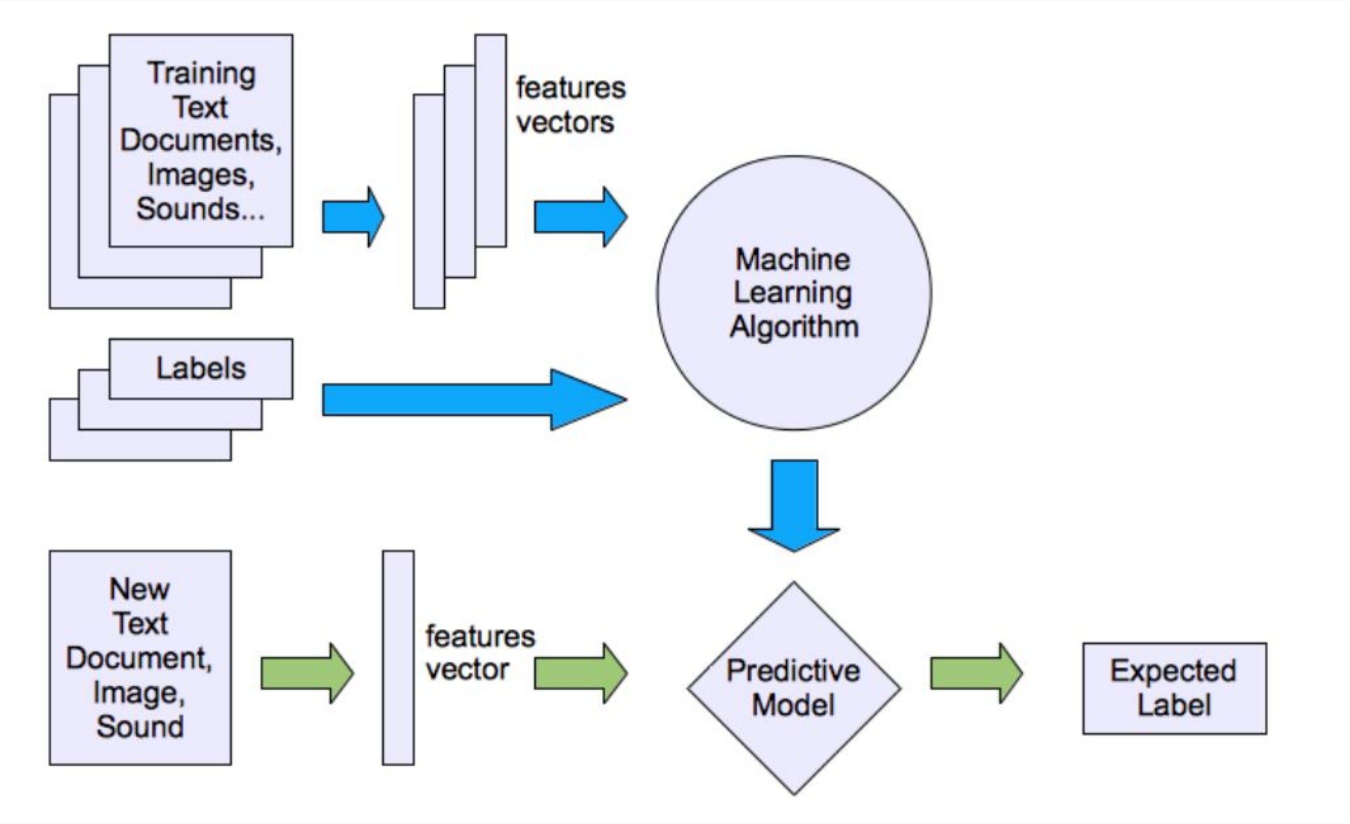
1. Demonstration of the learning paradigm.

   We will see how to apply the learning paradigm on a very simple model – the linear model. This will give us the chance to see some details of the training process.

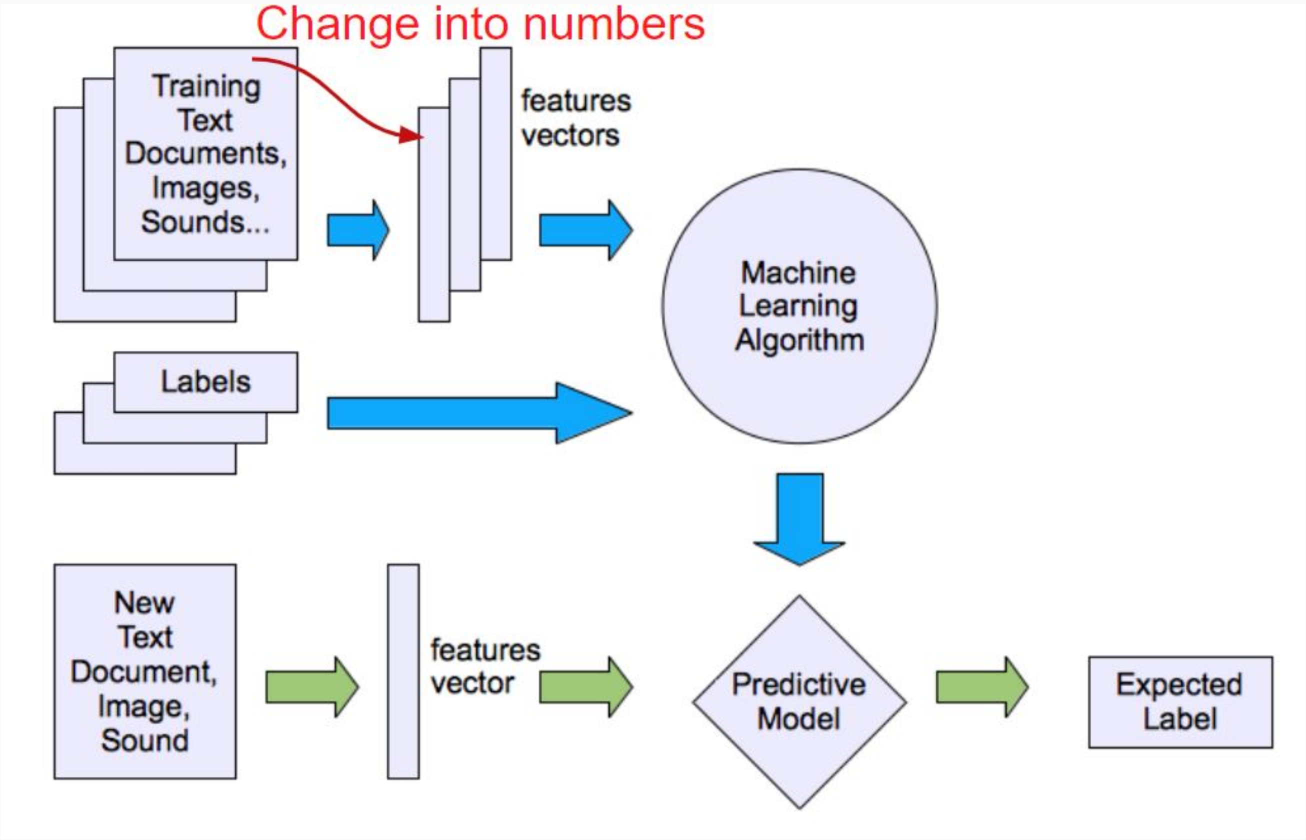
2. Neural Networks

   We will see how, with a small twist on linear models, you can obtain a very sophisticated model – the neural network. We will also discuss why Neural Networks are so popular.

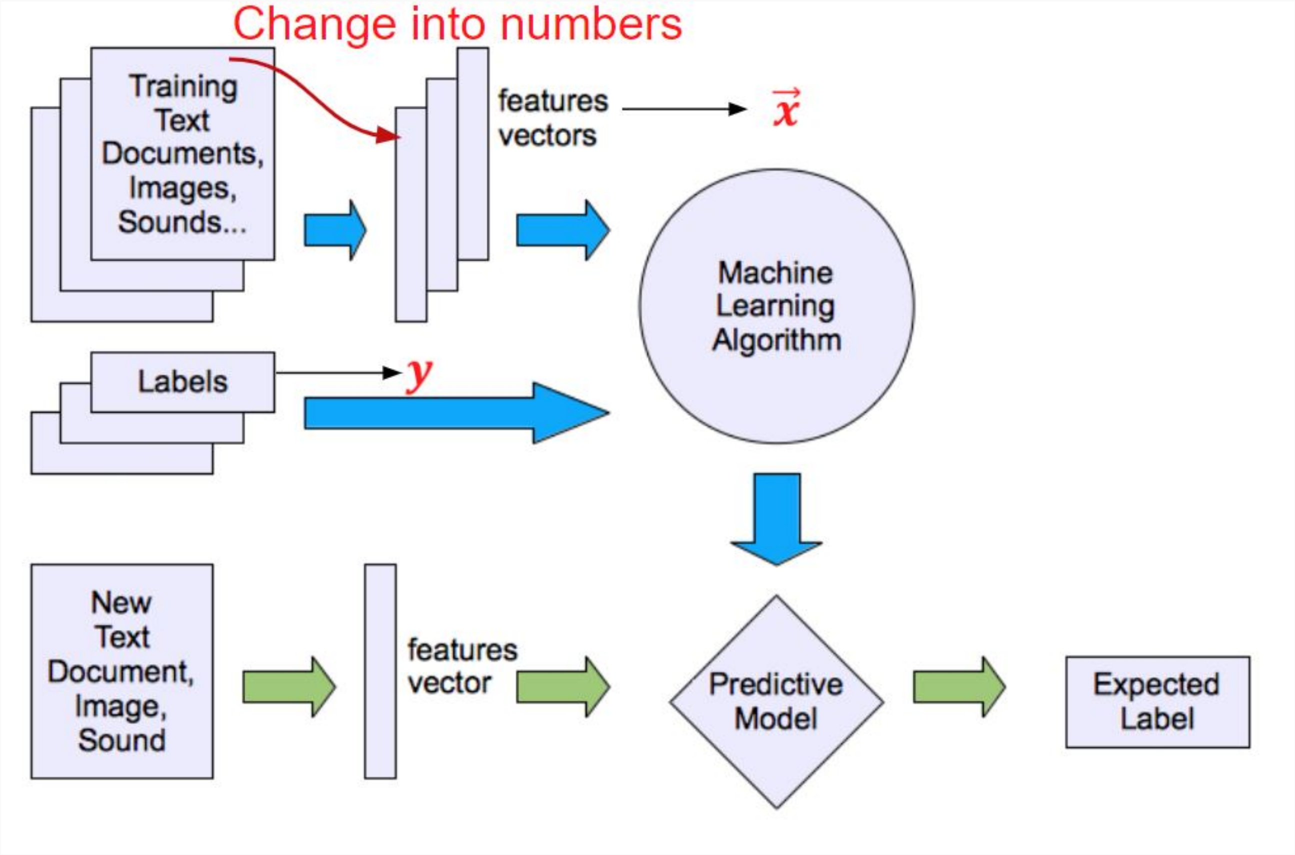
General structure of supervised learning



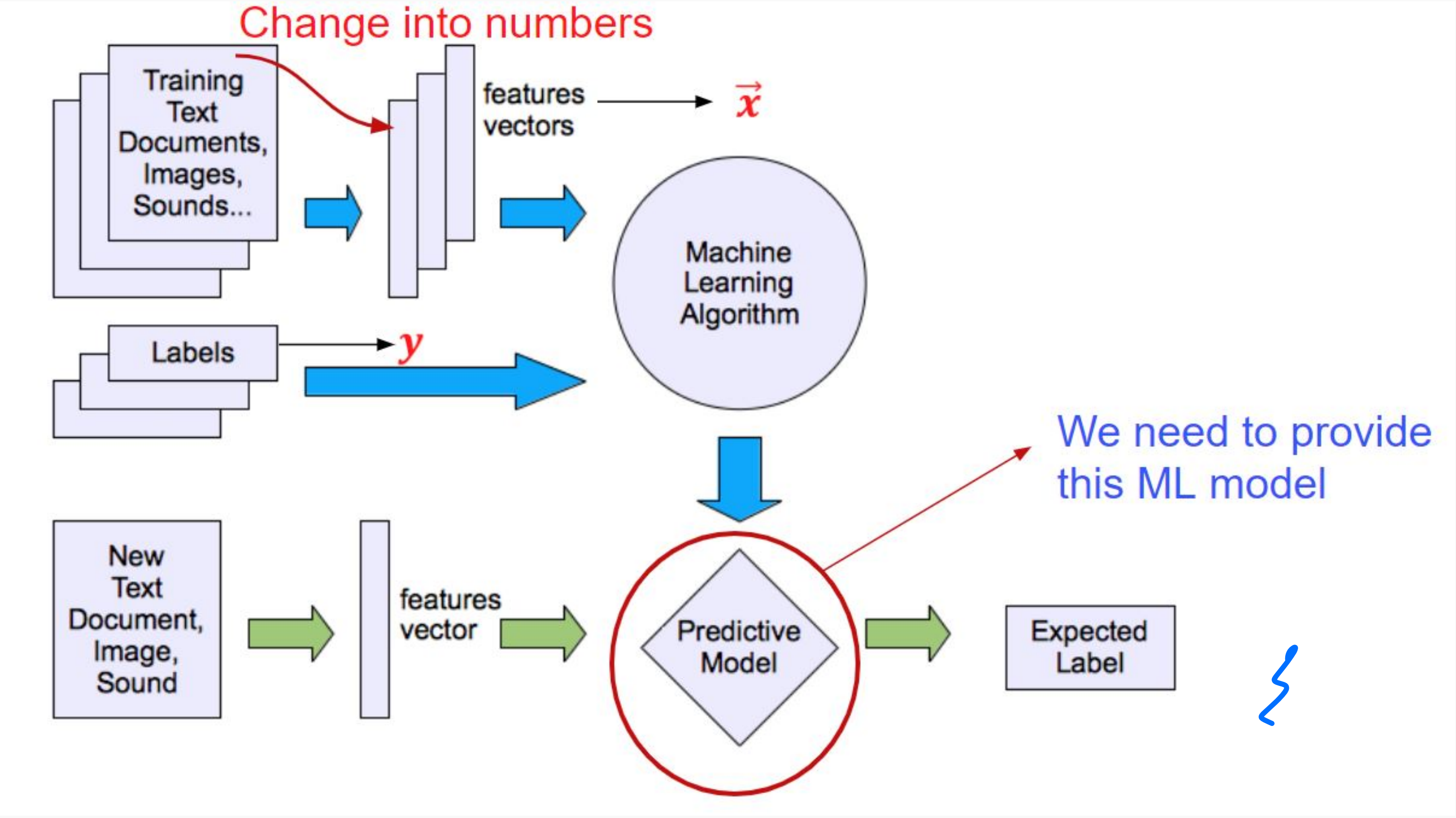
General structure of supervised learning



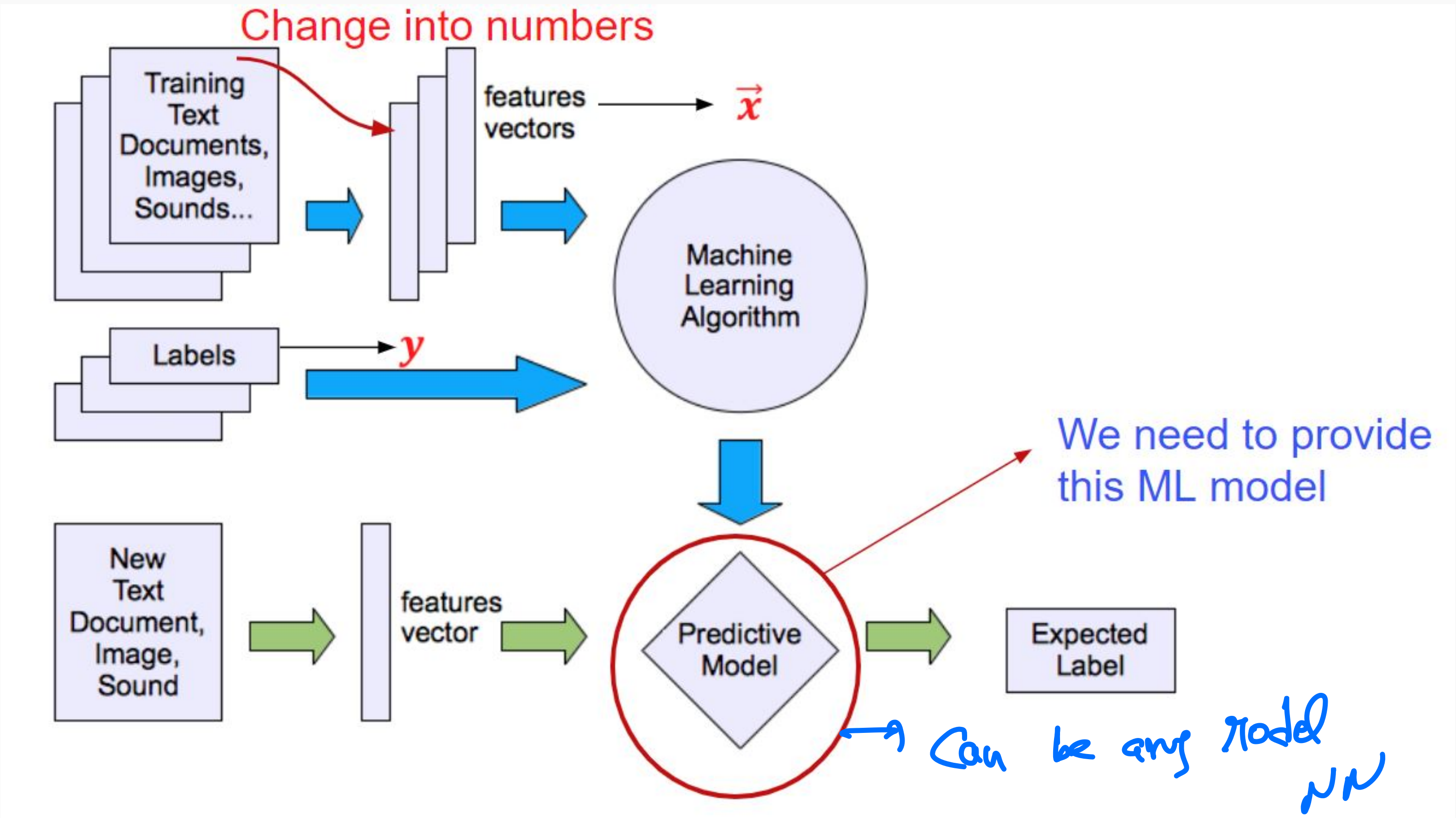
General structure of supervised learning



General structure of supervised learning



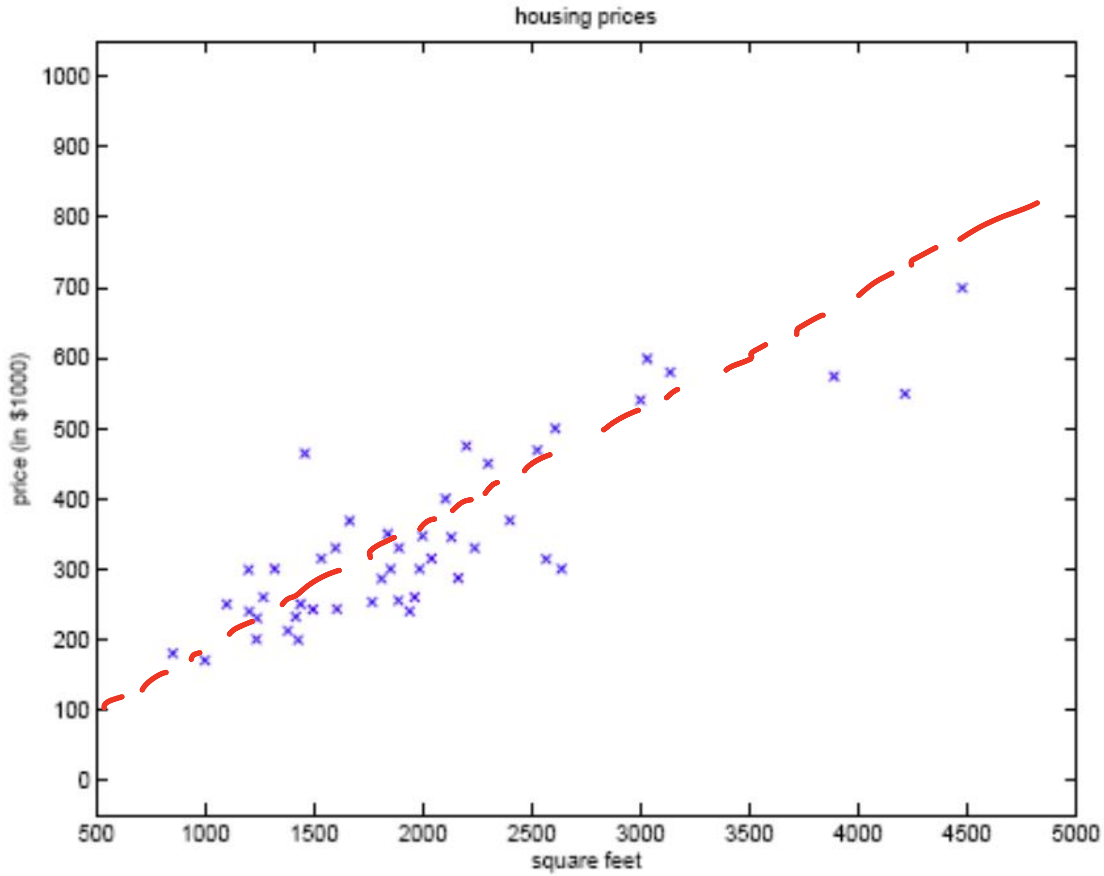
General structure of supervised learning



Philosophical underpinning of ML/NN: All problems are data, all solutions are functions

Regression Example – House price prediction

kaggle.com – housing price prediction

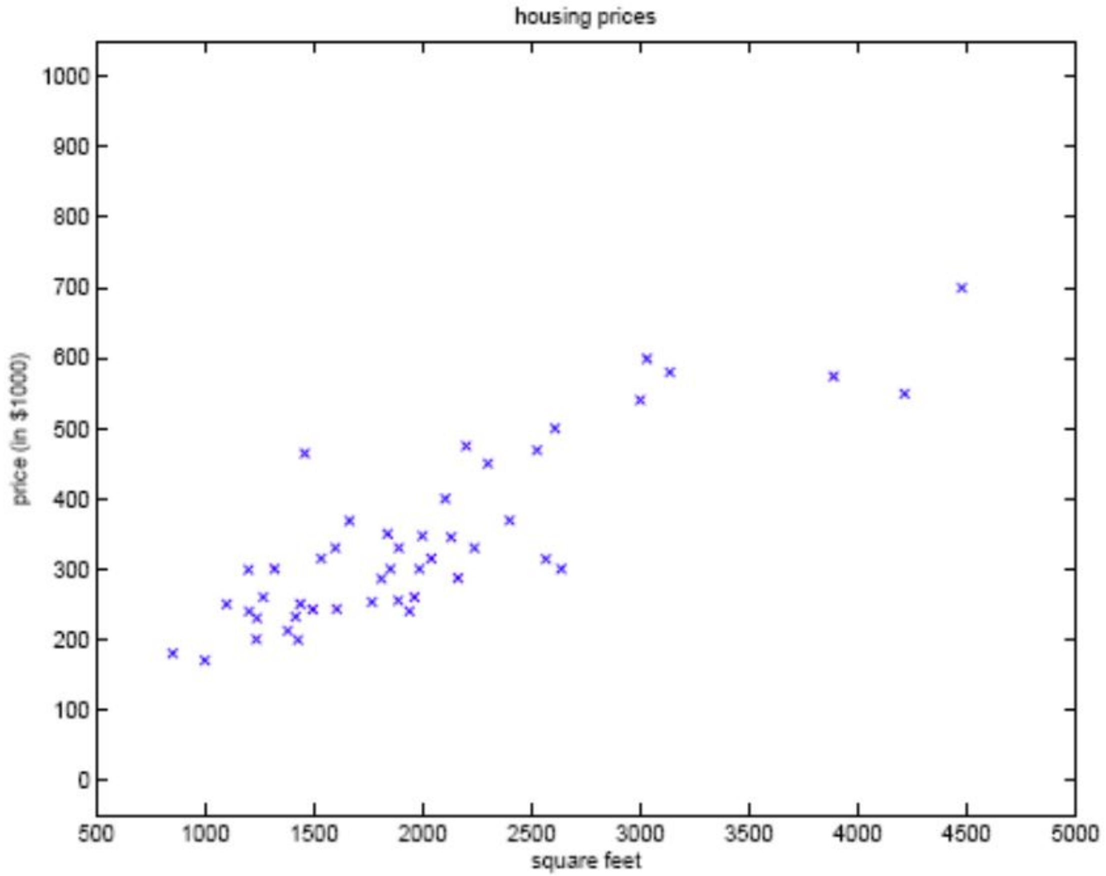


Regression problem

Living area ($feet^2$)	Price (1000$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

y - scalar value

Regression Example – House price prediction



Living area (feet^2)	Price (1000$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

Regression question : Given the living area, can you predict the price?

Learning a model or map for house price

- Machine learning may be viewed as learning an optimal map between
  - Input Vector $x \rightarrow$ Area, Image, Email, Geometry
  - Output Vector $y \rightarrow$ Price, Class, Spam label, Heat Transfer
  - by training over many example sets $\rightarrow$ Given data points

Learning a model or map for house price

- Machine learning may be viewed as learning an optimal map between
  - Input Vector x $\rightarrow$ Area, Image, Email, Geometry
  - Output Vector y $\rightarrow$ Price, Class, Spam label, Heat Transfer
  - by training over many example sets $\rightarrow$ Given data points

- The map/function consists of two parts
  - Form of the functions
    - E.g., Linear:

      Bias

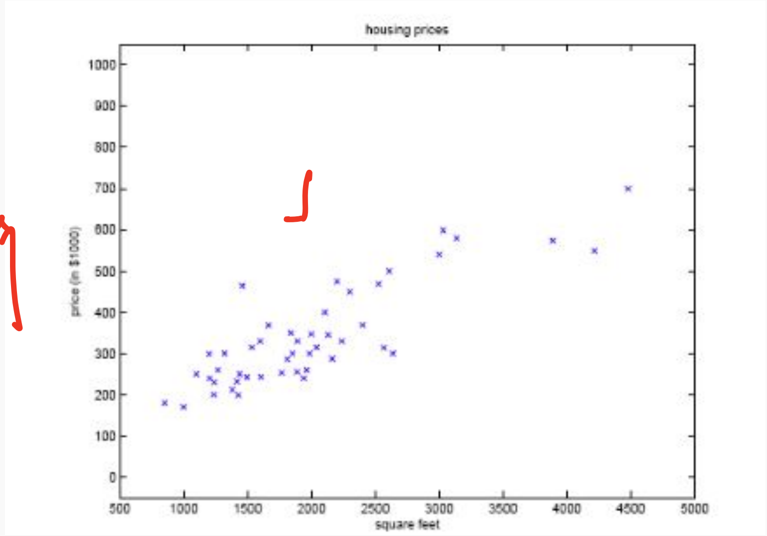
      $$\hat{y} = h^{lin}(x) = w_0 + w_1 x$$

      We provide this by ourselves. The machine doesn't know to do it.
  - Parameters of the map
    - E.g Find w_0, w_1 through "training" by minimizing least squares

Learning a model or map for house price

- Machine learning may be viewed as learning an optimal map between
  - Input Vector x $\rightarrow$ Area, Image, Email, Geometry
  - Output Vector y $\rightarrow$ Price, Class, Spam label, Heat Transfer
  - by training over many example sets $\rightarrow$ Given data points

y



Area → x

- The map/function consists of two parts
  - Form of the functions
    - E.g., Linear:

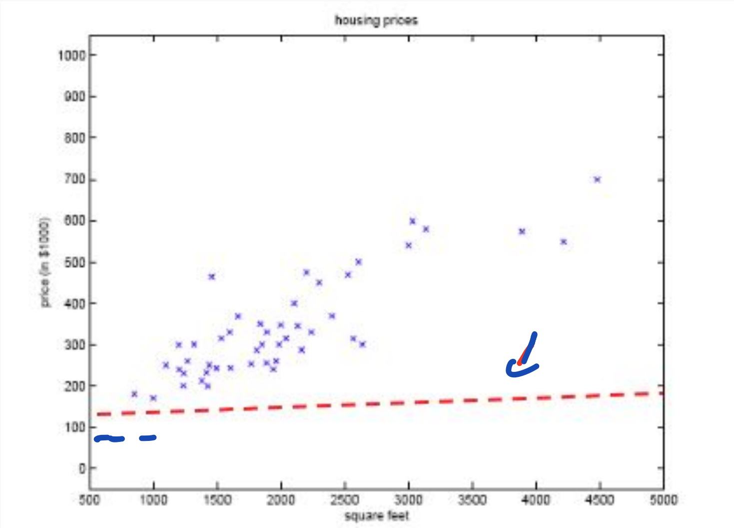
      $\hat{y} = h^{lin}(x) = w_0 + w_1 x$

      We provide this by ourselves.

      The machine doesn't know to do it.
  - Parameters of the map
    - E.g Find w_0, w_1 through "training" by minimizing least squares

Learning a model or map for house price

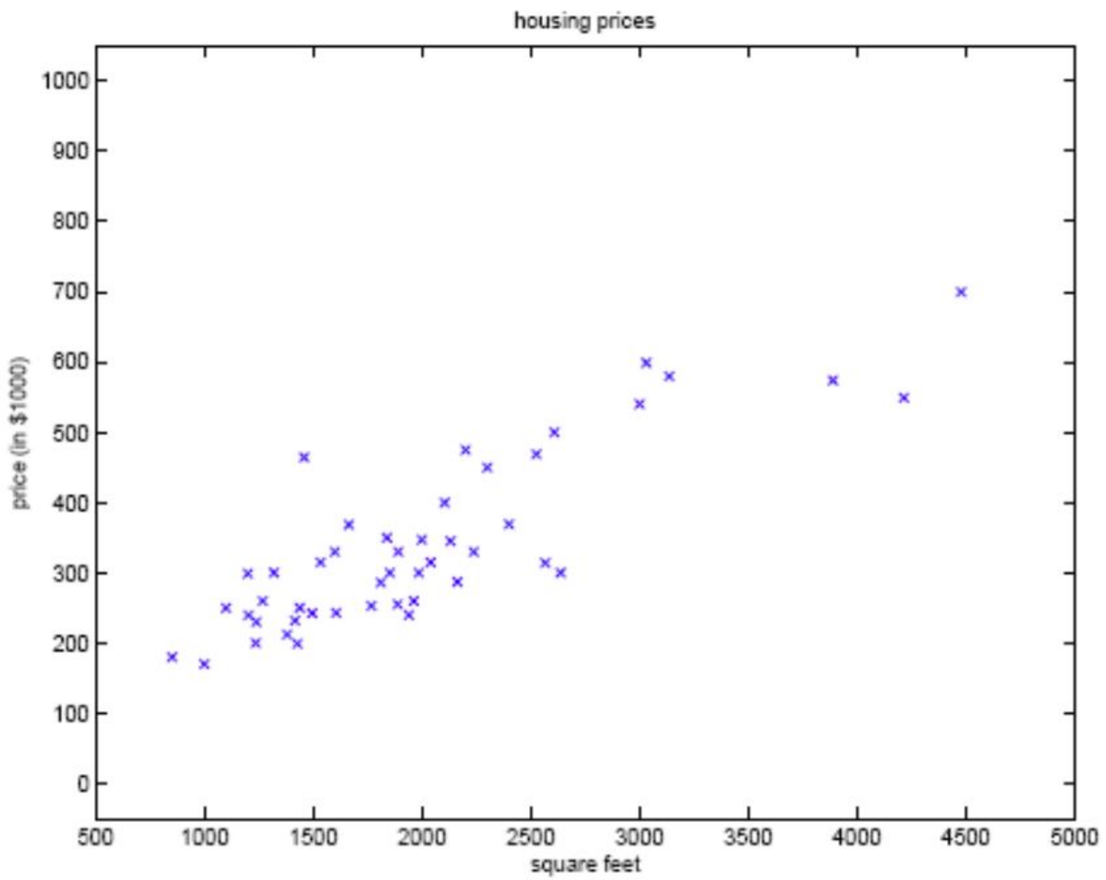
- Machine learning may be viewed as learning an optimal map between
  - Input Vector x $\rightarrow$ Area, Image, Email, Geometry
  - Output Vector y $\rightarrow$ Price, Class, Spam label, Heat Transfer
  - by training over many example sets $\rightarrow$ Given data points



- The map/function consists of two parts
  - Form of the functions
    - E.g., Linear:
      $\hat{y} = h^{lin}(x) = \boxed{w_0 + w_1 x}$
      We provide this by ourselves. The machine doesn't know to do it.
  - Parameters of the map

    w_0, w_1 → randomly initialized
    - E.g Find w_0, w_1 through "training" by minimizing least squares

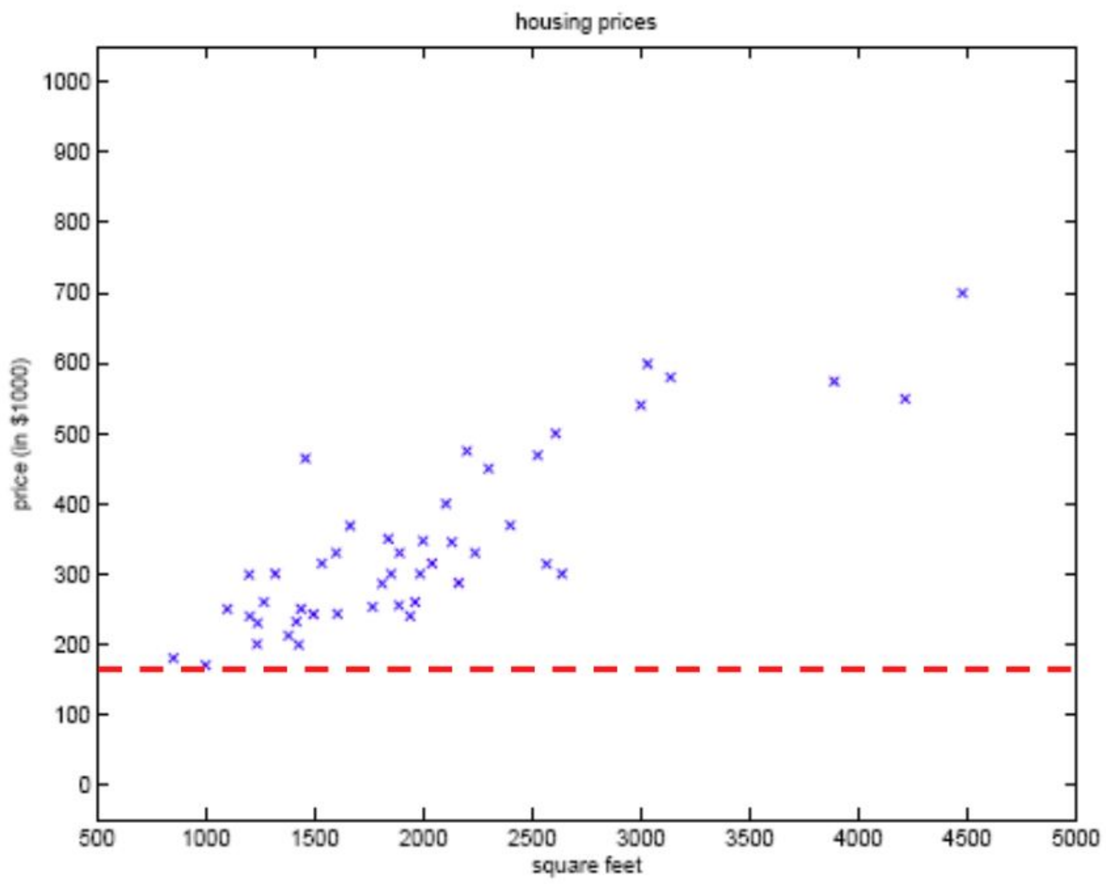
Regression Example – Housing price prediction



Living area (feet^2)	Price (1000\$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

$w_0 = 150, w_1 = 0$

Regression Example – Housing price prediction

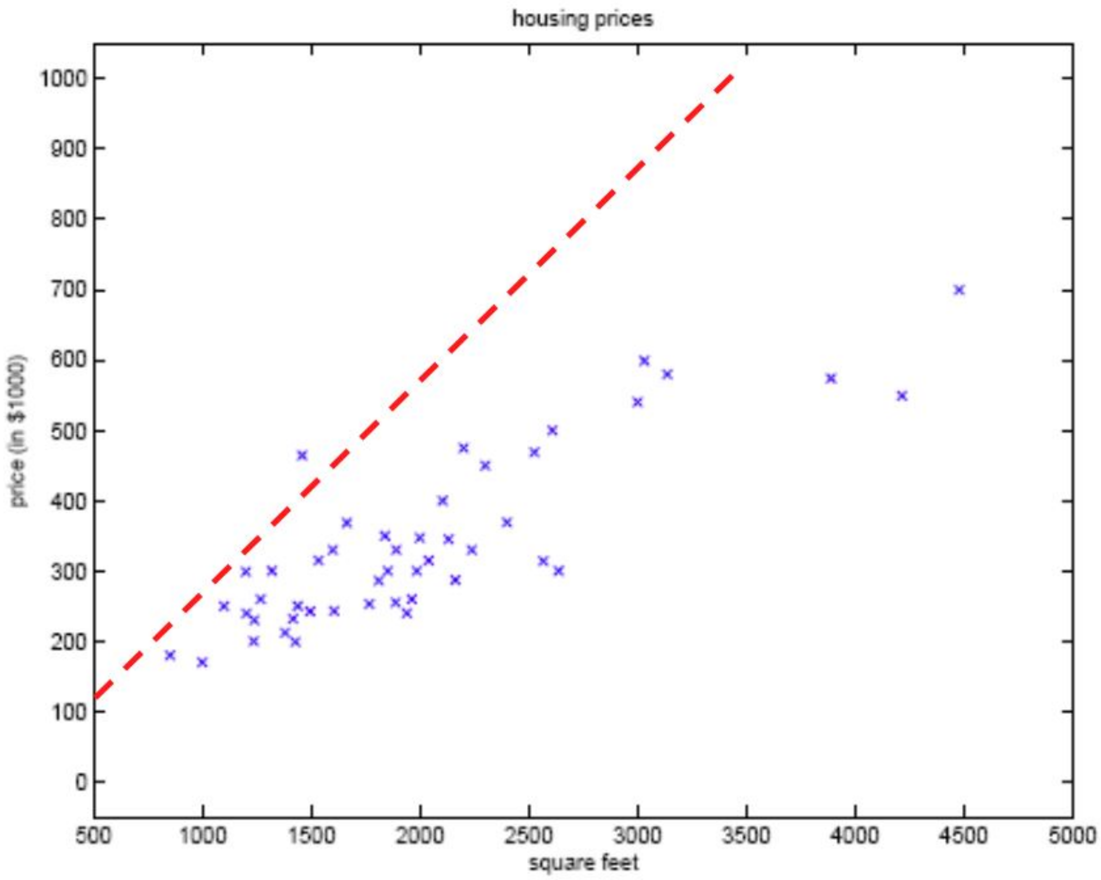


$\hat{y} = w_0 + w_1 x$

Living area (feet^2)	Price (1000\$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

$w_0 = 150,\ w_1 = 0$

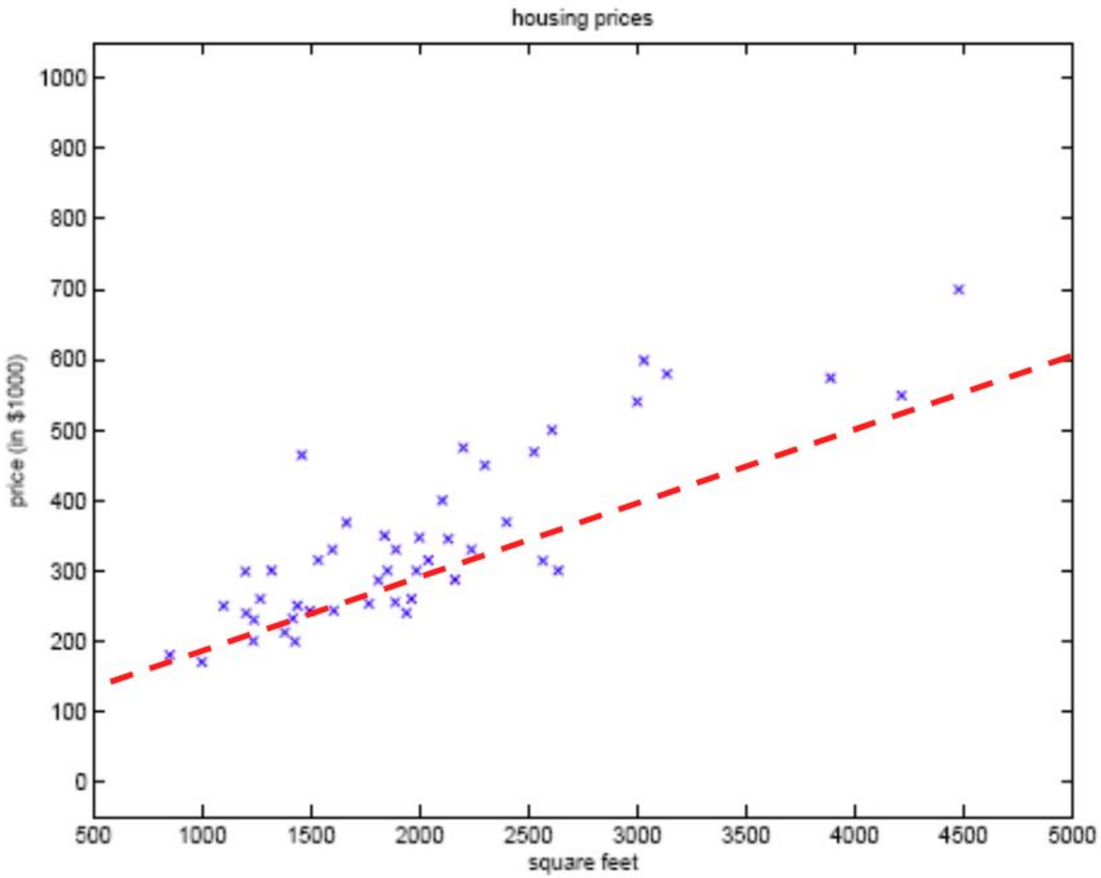
Regression Example – Housing price prediction



Living area (feet^2)	Price (1000\$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

Let us guess for parameters and improve

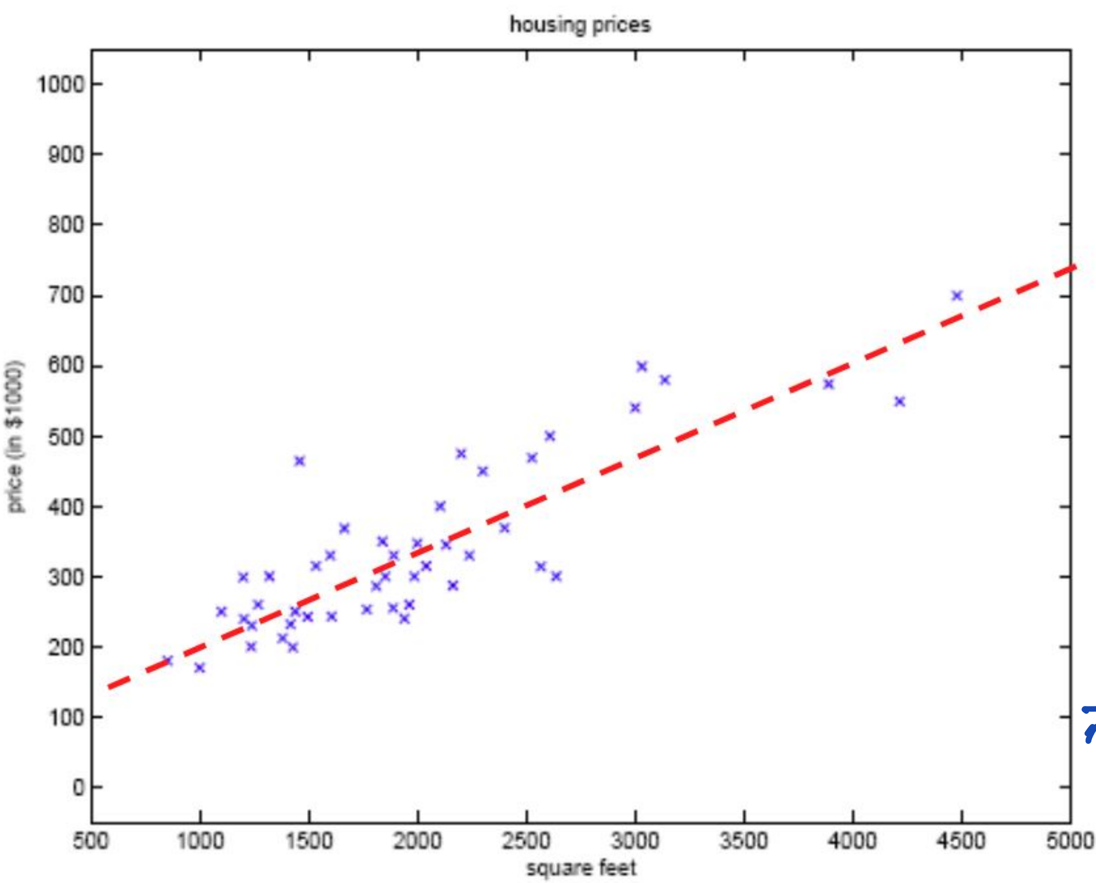
Regression Example – Housing price prediction



Living area (feet^2)	Price (1000$s)
2104	400
1600	330
2400	369
1416	232
3000	540
$\vdots$	$\vdots$

Let us guess for parameters and improve

Regression Example – Housing price prediction



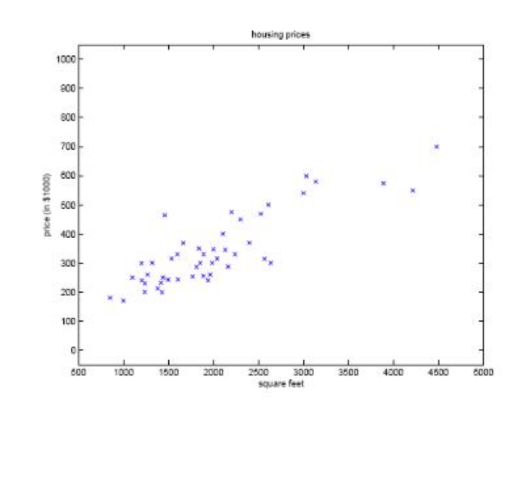
Living area (feet^2)	Price (1000\$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

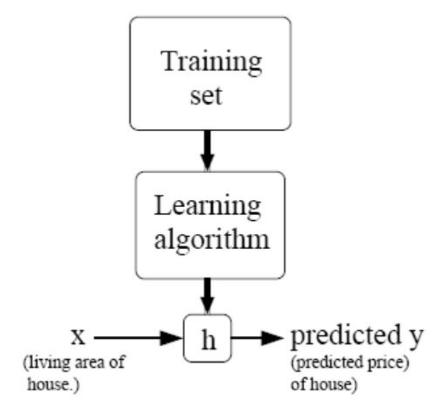
$\frac{1}{m}\sum_{m} (y-\hat{y})^2 \rightarrow$ minimize
w_0, w_1

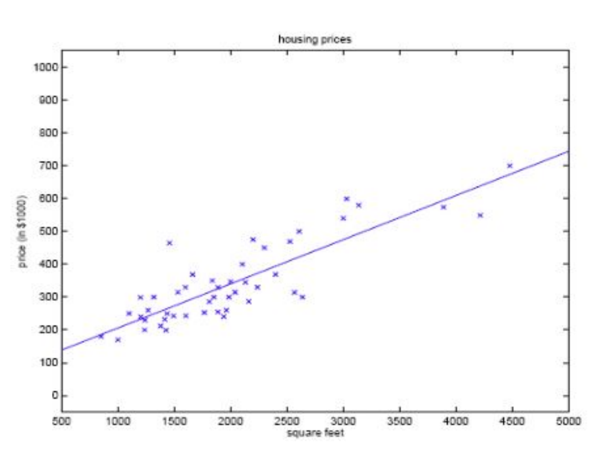
$\frac{1}{m} \sum (y - (w_0 + w_1 x))^2 = J$

Let us guess for parameters and improve

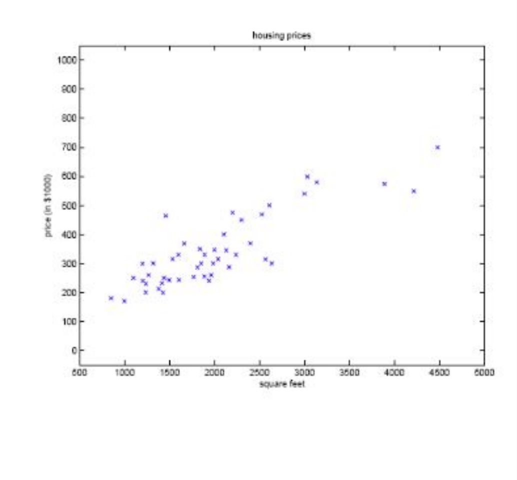
Workflow

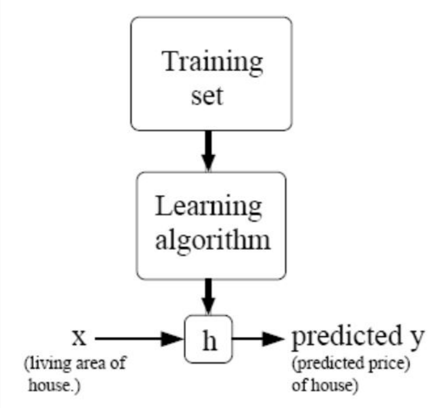

housing prices
price (in $1000)
square feet

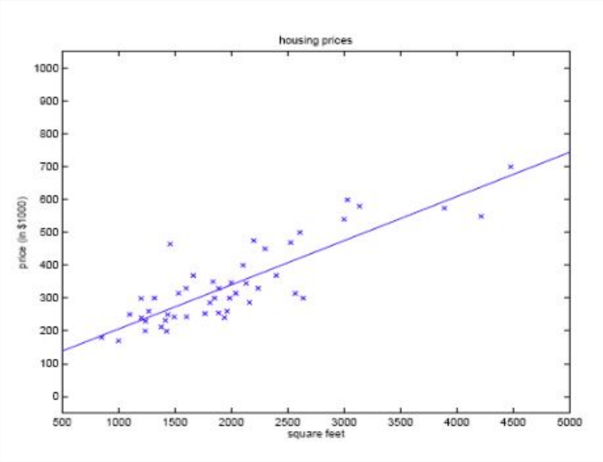

Training set
Learning algorithm
x
(living area of house.)
h
predicted y
(predicted price) of house)


housing prices
price (in $1000)
square feet

Workflow

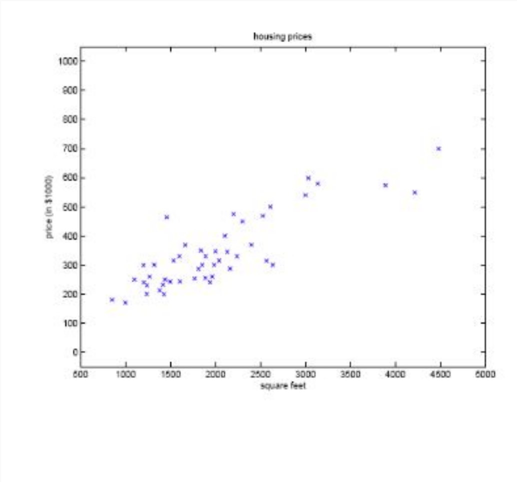


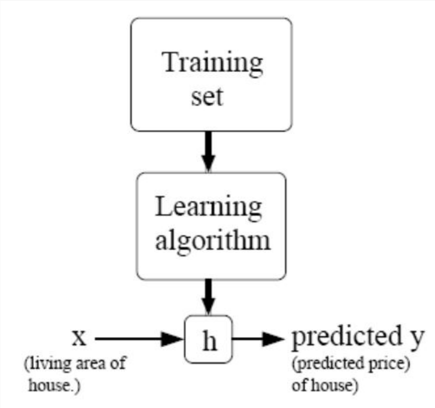


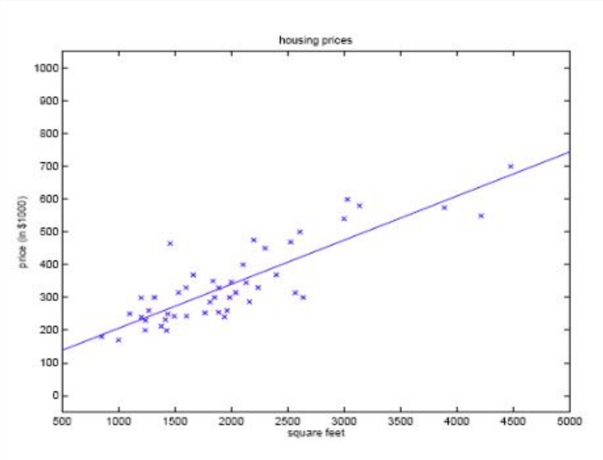


- Determine relevant input, output vectors and features

Workflow







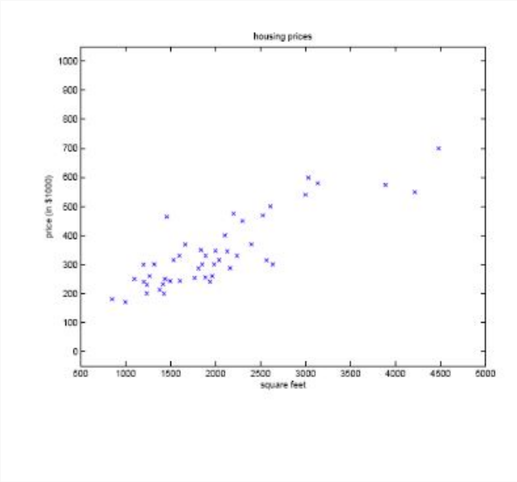
- Determine relevant input, output vectors and features
- Determine form of model (say, linear)

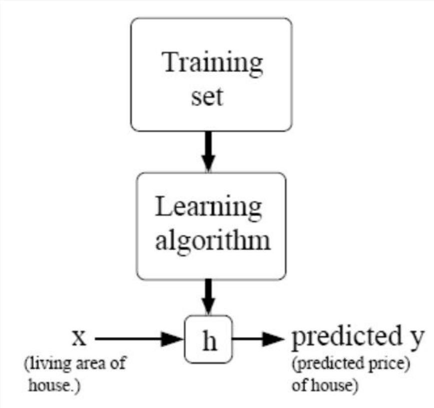
$\left\{ \begin{array}{l} \text{Input vector} \\ \text{Input features} \end{array} \right\}$ $\vec{x} = \{$Area, no of Bedrooms, Swimming pool area$\}$

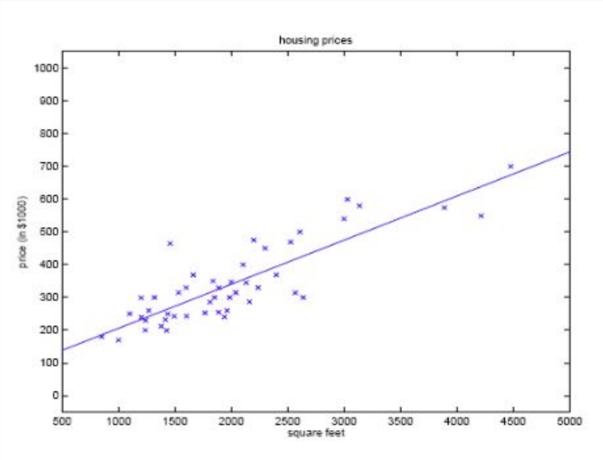
$$\vec{x} = \{1000, 2, 300\}$$

$$y = \{w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)} + w_3 x_3^{(i)}\}$$

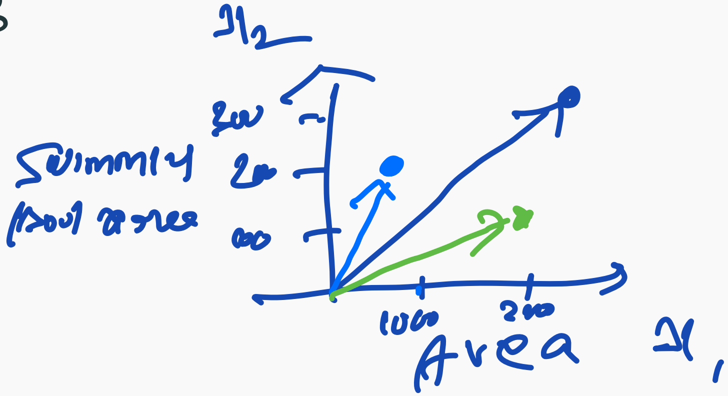
Workflow



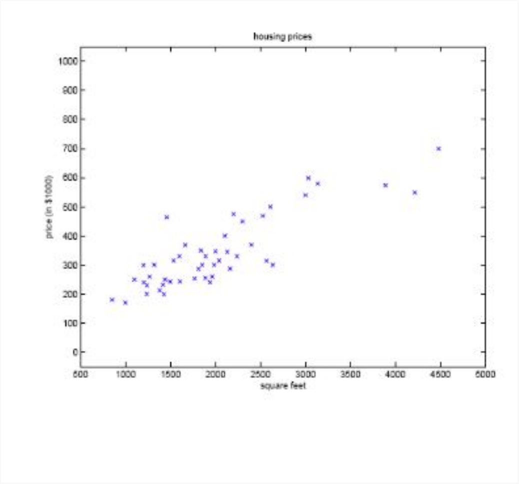


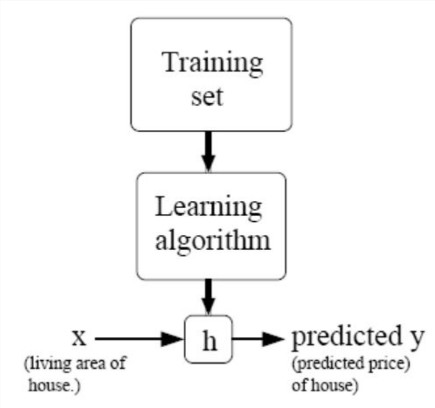


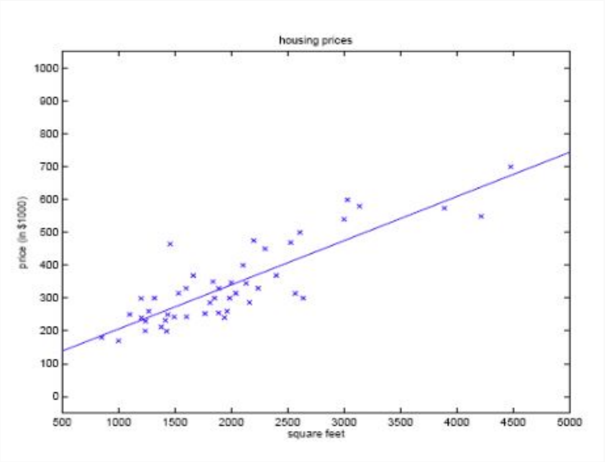
- Determine relevant input, output vectors and features
- Determine form of model (say, linear)
- Decide on cost function – e.g



Workflow

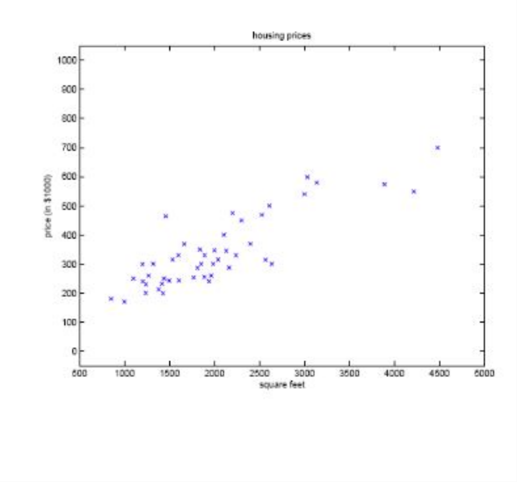


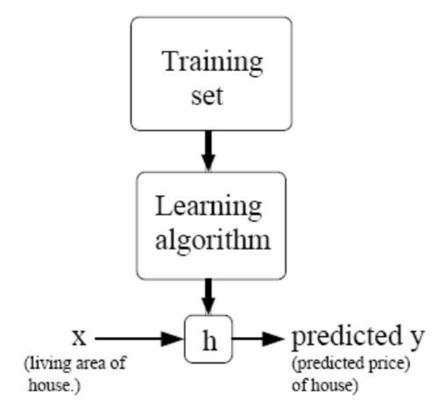


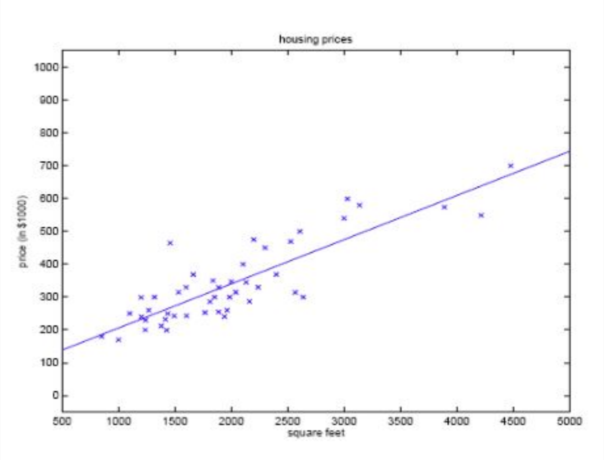


- Determine relevant input, output vectors and features
- Determine form of model (say, linear)
- Decide on cost function – e.g $\sum [y - \hat{y}]^2 = \sum (y_i \, \hat{y}_i)$
- Find the parameters that minimize the cost function

Workflow

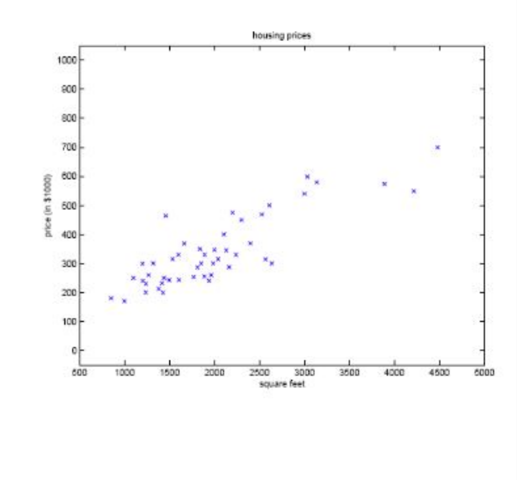


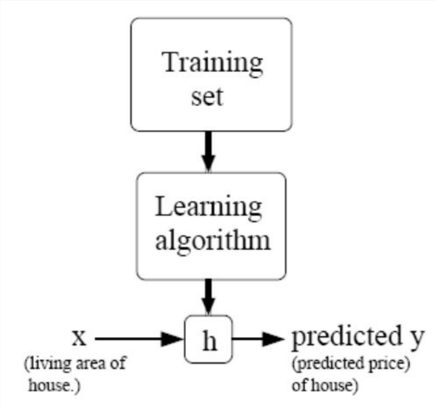


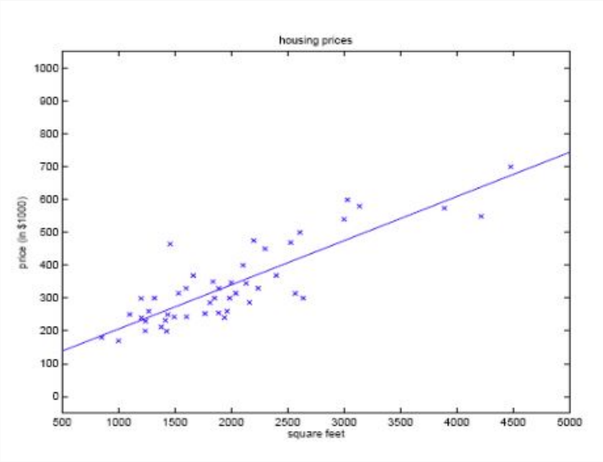


- Determine relevant input, output vectors and features
- Determine form of model (say, linear)
- Decide on cost function – e.g
- Find the parameters that minimize the cost function
    - Can either solve this problem directly or iteratively

Workflow

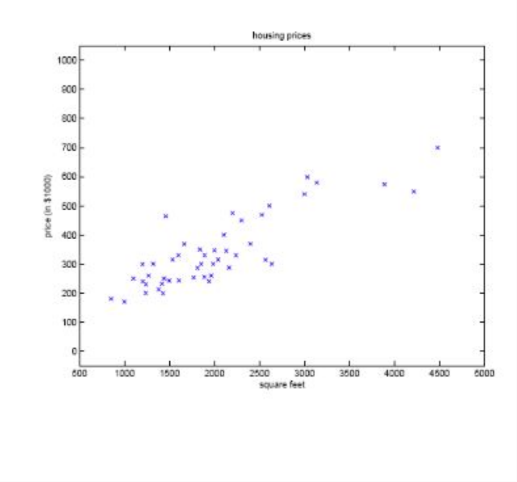


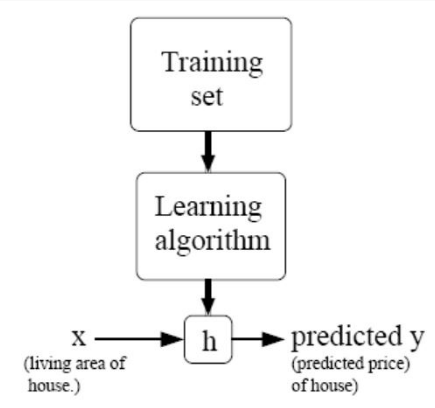


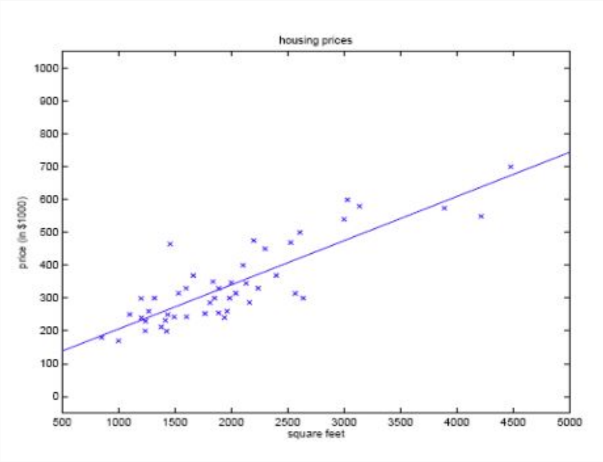


- Determine relevant input, output vectors and features
- Determine form of model (say, linear)
- Decide on cost function – e.g $J(y, \hat{y}) = (y - \hat{y})^2$
- Find the parameters that minimize the cost function
    - Can either solve this problem directly or iteratively
    - Simplest iterative algorithm – Gradient Descent

Workflow







- Determine relevant input, output vectors and features
- Determine form of model (say, linear)
- Decide on cost function – e.g $J(y, \hat{y}) = (y - \hat{y})^2$
- Find the parameters that minimize the cost function
  - Can either solve this problem directly or iteratively
  - Simplest iterative algorithm – Gradient Descent

Is there ONE general model which can approximate any function?

Summary

- Learning algorithms involve various types of hypothesis functions $\hat{y} = h(x; w)$.

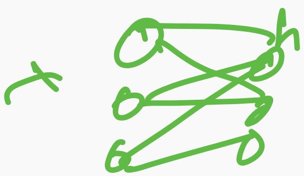
Summary

- Learning algorithms involve various types of hypothesis functions $\hat{y} = h(x; w)$.

  $w_0 + w_1 x$

- Each of these have their own purpose and domain where they work well
  - Linear Regression - For simple polynomial regression problems
  - Logistic Regression - For two-class (binary) classification problems

    f 1 of k classification
  - Deep Neural Networks - For any general, non-linear problem
    - There is a theorem that assures us that sufficiently large neural network will approximate any function
  - Convolutional Neural Networks - For vision/image based problems
  - Recurrent Neural Networks - For sequential/time-series problems



Summary

- Learning algorithms involve various types of hypothesis functions $\hat{y} = h(x; w)$. (x_1, x_2) $\to w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 x_2 + \cdots$
- Each of these have their own purpose and domain where they work well
  - Linear Regression - For simple polynomial regression problems
  - Logistic Regression - For two-class (binary) classification problems
  - Deep Neural Networks - For any general, non-linear problem
    - There is a theorem that assures us that sufficiently large neural network will approximate any function
  - Convolutional Neural Networks - For vision/image based problems
  - Recurrent Neural Networks - For sequential/time-series problems
- There are also appropriate loss functions for each.
- It is possible that you might have a better model than these for your problem. The general procedure outlined here remains the same.