CS360 MACHINE LEARNING

Credit and Hours

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	**4**	**2**	**-**	**6**	5
Marks	**100**	**50**	**-**	**150**	

Few Questions ?

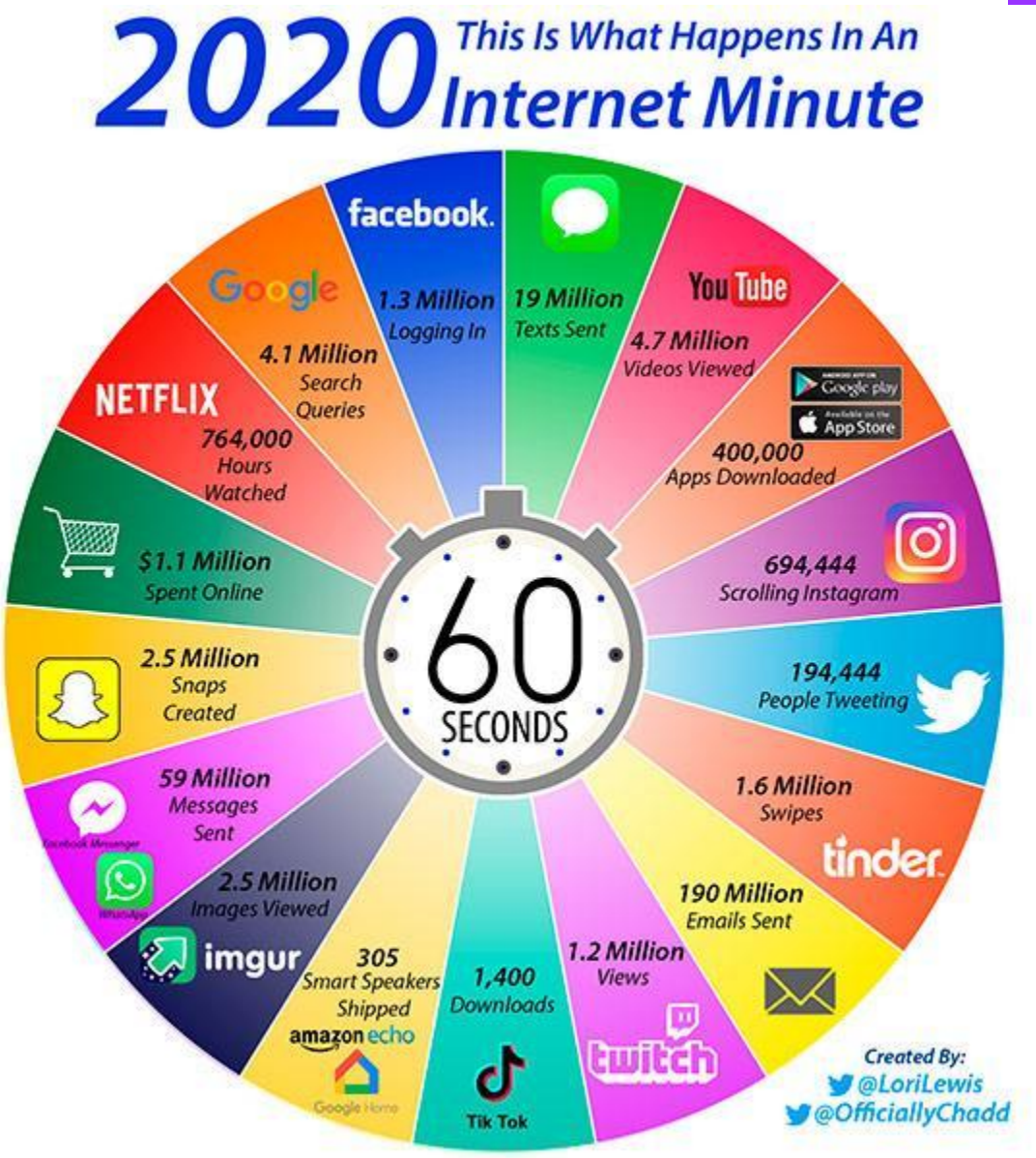
Questions	Timer
How many text messages in 60 Seconds ?	
How many you tube video viewed in 60 Seconds?	
How many Applications downloaded in 60 Seconds?	
How many Whatsapp Messages in 60 Seconds ?	
How Many Email send in 60 Seconds?	
How many Google Search Query in 60 Seconds?	
How many Face book Login in 60 Seconds ?	

An Internet One Minute : 2017

An Internet One Minute : 2018

An Internet One Minute : 2019

An Internet One Minute : 2020



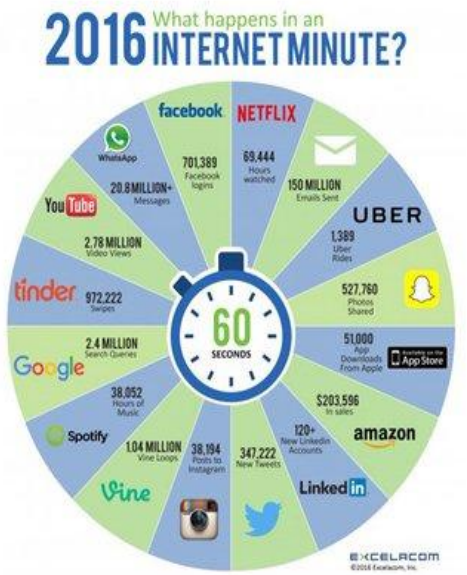
An Internet One Minute : 2020


2017 This Is What Happens In An Internet Minute
facebook
900,000 Logins
16 Million Text Messages
YouTube
4.1 Million Videos Viewed
342,000 Apps Downloaded
46,200 Posts Uploaded
Instagram
452,000 Tweets Sent
990,000 Swipes
tinder
156 Million Emails Sent
40,000 Hours Listened
Spotify
50 Voice-First Devices Shipped
amazon echo
120 New Accounts Created
LinkedIn
15,000 GIFs Sent via Messenger
1.8 Million Snaps Created
$751,522 Spent Online
NETFLIX
70,017 Hours Watched
Google
3.5 Million Search Queries
60 SECONDS
Created By:
@LoriLewis
@OfficiallyChadd

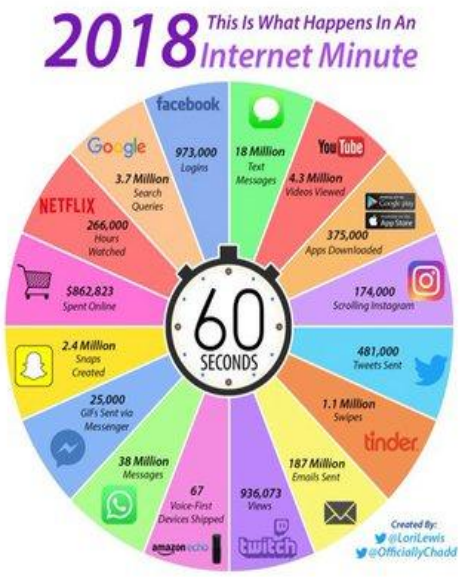

2018 This Is What Happens In An Internet Minute
facebook
973,000 Logins
18 Million Text Messages
YouTube
4.3 Million Videos Viewed
375,000 Apps Downloaded
174,000 Scrolling Instagram
481,000 Tweets Sent
1.1 Million Swipes
tinder
187 Million Emails Sent
936,073 Views
twitch
67 Voice-First Devices Shipped
amazon echo
38 Million Messages
25,000 GIFs Sent via Messenger
2.4 Million Snaps Created
$862,823 Spent Online
NETFLIX
266,000 Hours Watched
Google
3.7 Million Search Queries
60 SECONDS
Created By:
@LoriLewis
@OfficiallyChadd

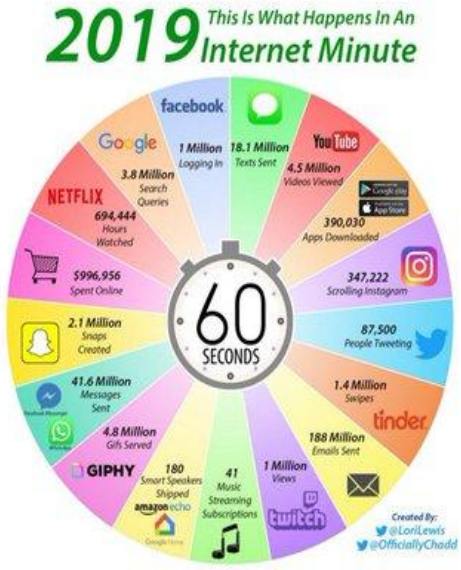

2019 This Is What Happens In An Internet Minute
facebook
1 Million Logging In
18.1 Million Texts Sent
YouTube
4.5 Million Videos Viewed
390,030 Apps Downloaded
347,222 Scrolling Instagram
87,500 People Tweeting
1.4 Million Swipes
tinder
188 Million Emails Sent
1 Million Views
twitch
41 Music Streaming Subscriptions
180 Smart Speakers Shipped
amazon echo
4.8 Million Gifs Served
GIPHY
41.6 Million Messages Sent
2.1 Million Snaps Created
$996,956 Spent Online
NETFLIX
694,444 Hours Watched
Google
3.8 Million Search Queries
60 SECONDS
Created By:
@LoriLewis
@OfficiallyChadd


2020 This Is What Happens In An Internet Minute
facebook
1.3 Million Logging In
19 Million Texts Sent
YouTube
4.7 Million Videos Viewed
400,000 Apps Downloaded
694,444 Scrolling Instagram
194,444 People Tweeting
1.6 Million Swipes
tinder
190 Million Emails Sent
1.2 Million Views
twitch
1,400 Downloads
Tik Tok
305 Smart Speakers Shipped
amazon echo
2.5 Million Images Viewed
imgur
59 Million Messages Sent
2.5 Million Snaps Created
$1.1 Million Spent Online
NETFLIX
764,000 Hours Watched
Google
4.1 Million Search Queries
60 SECONDS
Created By:
@LoriLewis
@OfficiallyChadd

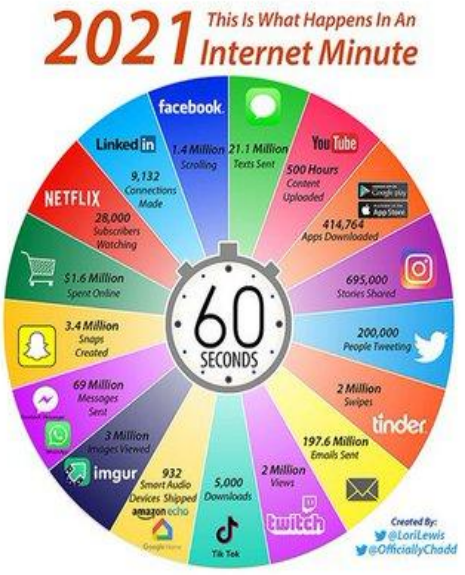

2016 What happens in an INTERNET MINUTE?
60 SECONDS
facebook
NETFLIX
UBER
YouTube
tinder
Google
Spotify
Vine
LinkedIn
amazon
EXCELACOM


2017 This Is What Happens In An Internet Minute
60 SECONDS
facebook
Google
NETFLIX
YouTube
tinder
Instagram
Spotify
LinkedIn
amazon echo
Created By:
@LoriLewis
@OfficiallyChadd


2018 This Is What Happens In An Internet Minute
60 SECONDS
facebook
Google
NETFLIX
YouTube
tinder
Instagram
twitch
amazon echo
Created By:
@LoriLewis
@OfficiallyChadd


2019 This Is What Happens In An Internet Minute
60 SECONDS
facebook
Google
NETFLIX
YouTube
tinder
GIPHY
twitch
amazon echo
Created By:
@LoriLewis
@OfficiallyChadd


2020 This Is What Happens In An Internet Minute
60 SECONDS
facebook
Google
NETFLIX
YouTube
tinder
imgur
twitch
amazon echo
Tik Tok
Created By:
@LoriLewis
@OfficiallyChadd


2021 This Is What Happens In An Internet Minute
60 SECONDS
facebook
LinkedIn
NETFLIX
YouTube
tinder
imgur
twitch
amazon echo
Tik Tok
Created By:
@LoriLewis
@OfficiallyChadd

The Information Age is Here!

- "Data doubles about every year, but useful information seems to be decreasing."
  - *Margaret Dunham, "Data Mining Techniques & Algorithms",*
- "There is a growing gap between the generation of data and our understanding of it."
  - *Witten & Frank, "Data Mining: Practical Machine Learning Tools",*

Evolution of Sciences

- Before 1600, **empirical science**
- 1600-1950s, **theoretical science**
  - Each discipline has grown a *theoretical* component. Theoretical models often motivate experiments and generalize our understanding.
- 1950s-1990s, **computational science**
  - Computational Science traditionally meant simulation. It grew out of our inability to find closed-form solutions for complex mathematical models.
- 1990-now, **data science**
  - The flood of data from new scientific instruments and simulations
  - The ability to economically store and manage petabytes of data online
  - The Internet and computing Grid that makes all these archives universally accessible
  - Scientific info. management, acquisition, organization, query, and visualization tasks scale almost linearly with data volumes.

Data science is a major new challenge!

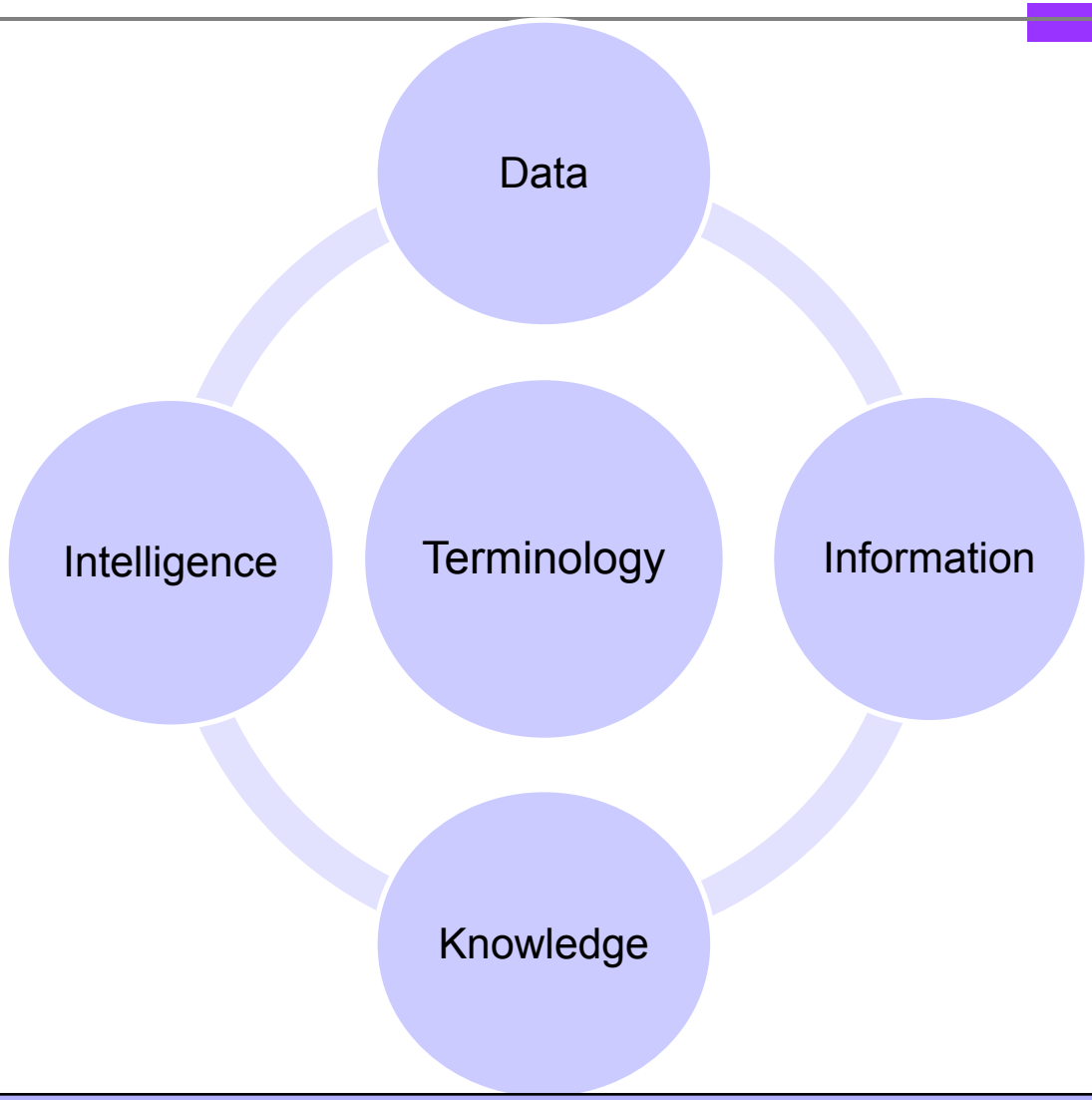
New Analytics Jobs By Industry

Analytics Services and Pharma lead the pack in the highest percentage of Analytics jobs being created in developing nations.

	US	INDIA	CHINA	UK	BRAZIL	JAPAN	SINGAPORE
ANALYTICS SERVICES	11%	54%	25%	9%	14%	6%	9%
PHARMA	14%	24%	32%	19%	30%	44%	26%
INSURANCE	39%	7%	8%	32%	11%	27%	24%
BANKING	20%	11%	22%	25%	19%	14%	25%
OIL & GAS	14%	3%	10%	13%	23%	8%	9%
COMMUNICATIONS TECHNOLOGIES	2%	1%	3%	2%	3%	1%	7%
TOTAL NUMBER OF JOBS	38,700	31,500	30,500	7,000	6,200	2,400	1,300

http://www.business-standard.com/article/technology/nasscom-boost-to-big-data-analytics-114062700723_1.html

Basic Terminology



Data □ Information □ Knowledge □ Understanding / Wisdom!

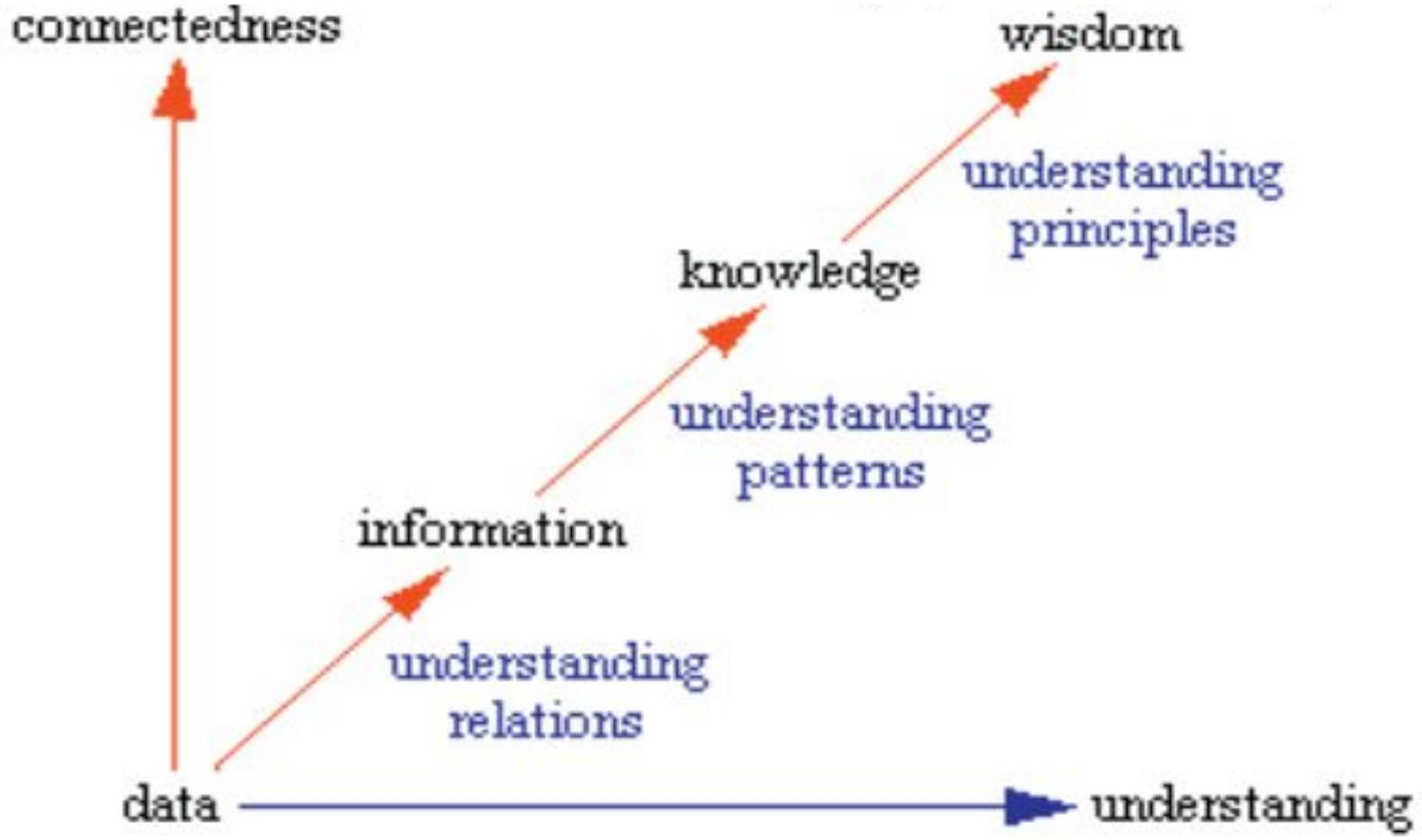
Data leads to Knowledge leads to Understanding

EXAMPLE

- Data = ages and heights of children (stored in database)
- Information = sorting as per age
- Knowledge = the older children tend to be taller
- Understanding = children's bones grow as they get older

Data □ Information □ Knowledge □ Understanding / Wisdom!

Relationship



Definition of Machine Learning

- Arthur Samuel, an early American leader in the field of computer gaming and artificial intelligence, coined the term "Machine Learning " in 1959 while at IBM.

- He defined machine learning as "the field of study that gives computers the ability to learn without being explicitly programmed"

Machine Learning

- Machine Learning is set of techniques to make computers better at doing things that humans(traditionally) can do better then machines.

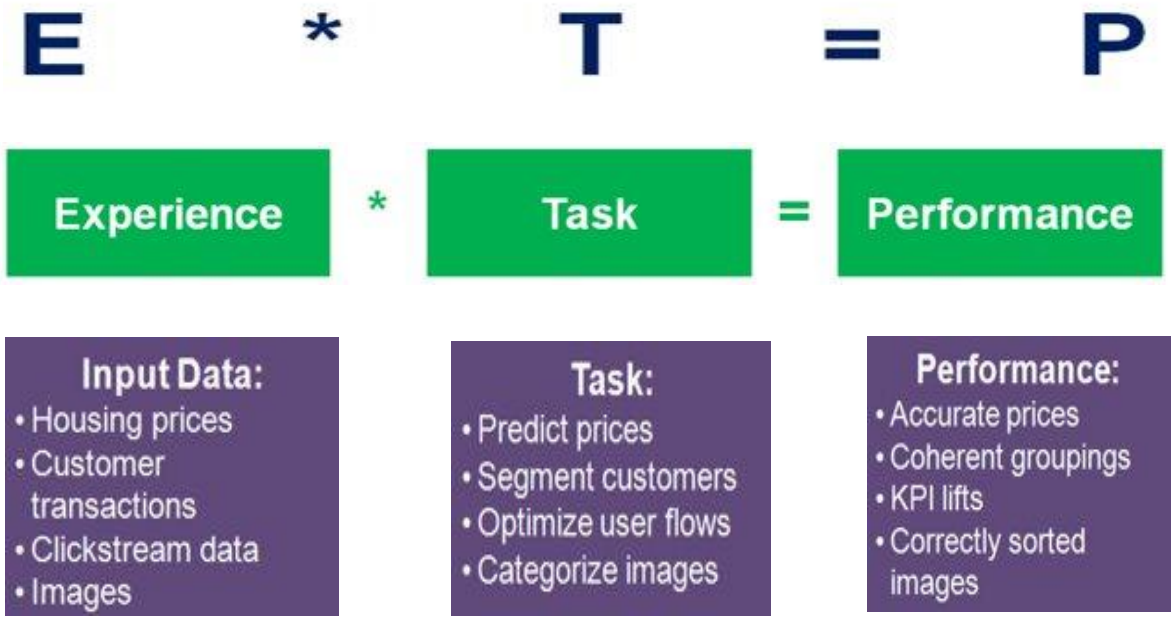
- Machine Learning involves making machines learn things like humans do

Machine Learning

Machine Learning is the science of getting computers to learn and act like humans do, and improve their learning over time in autonomous fashion, by feeding them data in the form of observations and real-world interactions.

Definition of learning

- 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 T, as measured by P , improves with experience E.



Examples

- Handwriting recognition learning problem
  - Task T : Recognizing and classifying handwritten words within images
  - Performance P : Percent of words correctly classified
  - Training experience E : A dataset of handwritten words with given classifications

Examples

- A robot driving learning problem
  - Task T : Driving on highways using vision sensors
  - Performance P : Average distance traveled before an error
  - Training experience E : A sequence of images and steering commands recorded while observing a human driver

What is Learning ?

- Data: Loan application data
- Task: Predict whether a loan should be approved or not.
- Performance measure: Accuracy.

Accuracy = 7/10 = 70%.

Classify all future applications (test data) to the majority class (i.e., No):

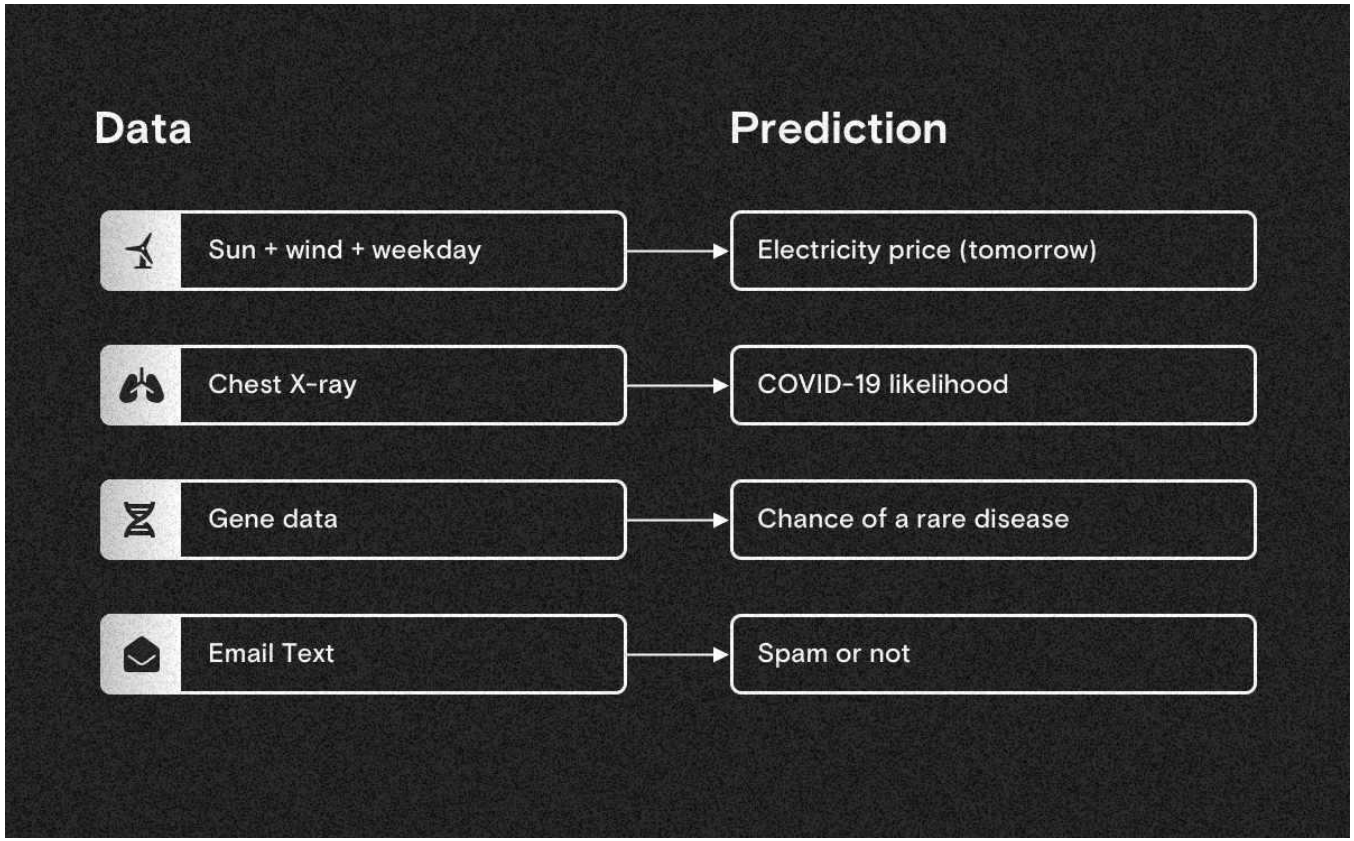
Tid	Refund	Marital Status	Taxable Income	Cheat Actual	Cheat predict
1	Yes	Single	125 Cr	No	No
2	No	Married	100 Cr	No	No
3	No	Single	70 Cr	No	No
4	Yes	Married	120 Cr	No	No
5	No	Divorced	95 Cr	Yes	No
6	No	Married	60 Cr	No	No
7	Yes	Divorced	220 Cr	No	No
8	No	Single	85 Cr	Yes	No
9	No	Married	75 Cr	No	No
10	No	Single	90 Cr	Yes	No

- We can do better than 70% with learning.

Machine Learning Program

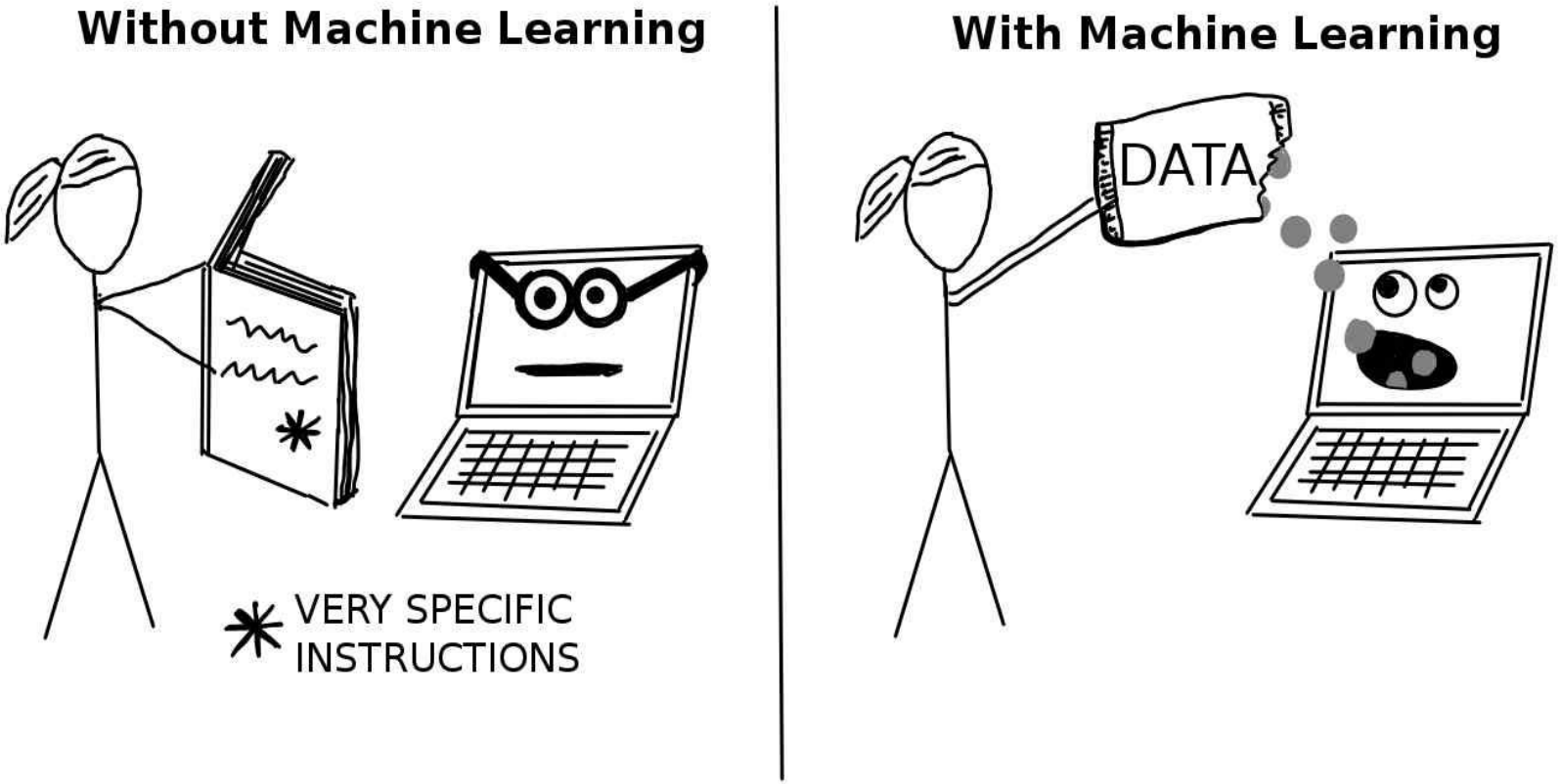
- **Definition:** A computer program which learns from experience is called a machine learning program or simply a learning program .
- Machine learning algorithms use historical data as input to predict new output values.
- Recommendation engines are a common use case for machine learning. Other popular uses include fraud detection, spam filtering, malware threat detection, business process automation (BPA) and Predictive maintenance.

Machine Learning Program



Machine Learning finds pattern in data and uses them to make predictions

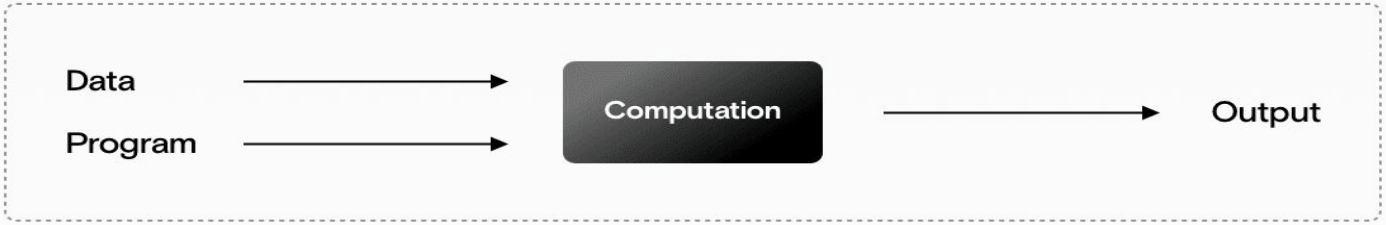
Traditional Programming Vs Machine Learning



Traditional Programming Vs Machine Learning

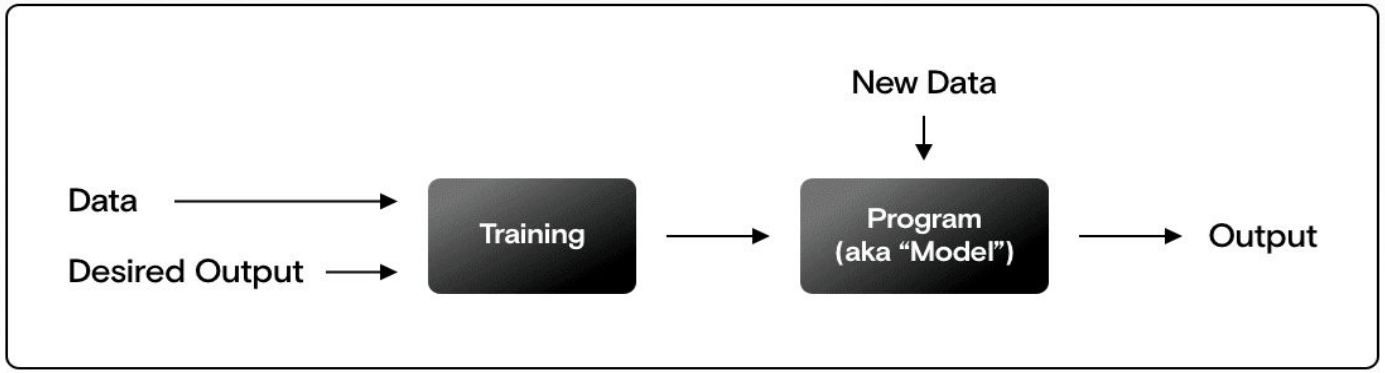
Traditional Programming

Developers write rules (program) that produce an output.

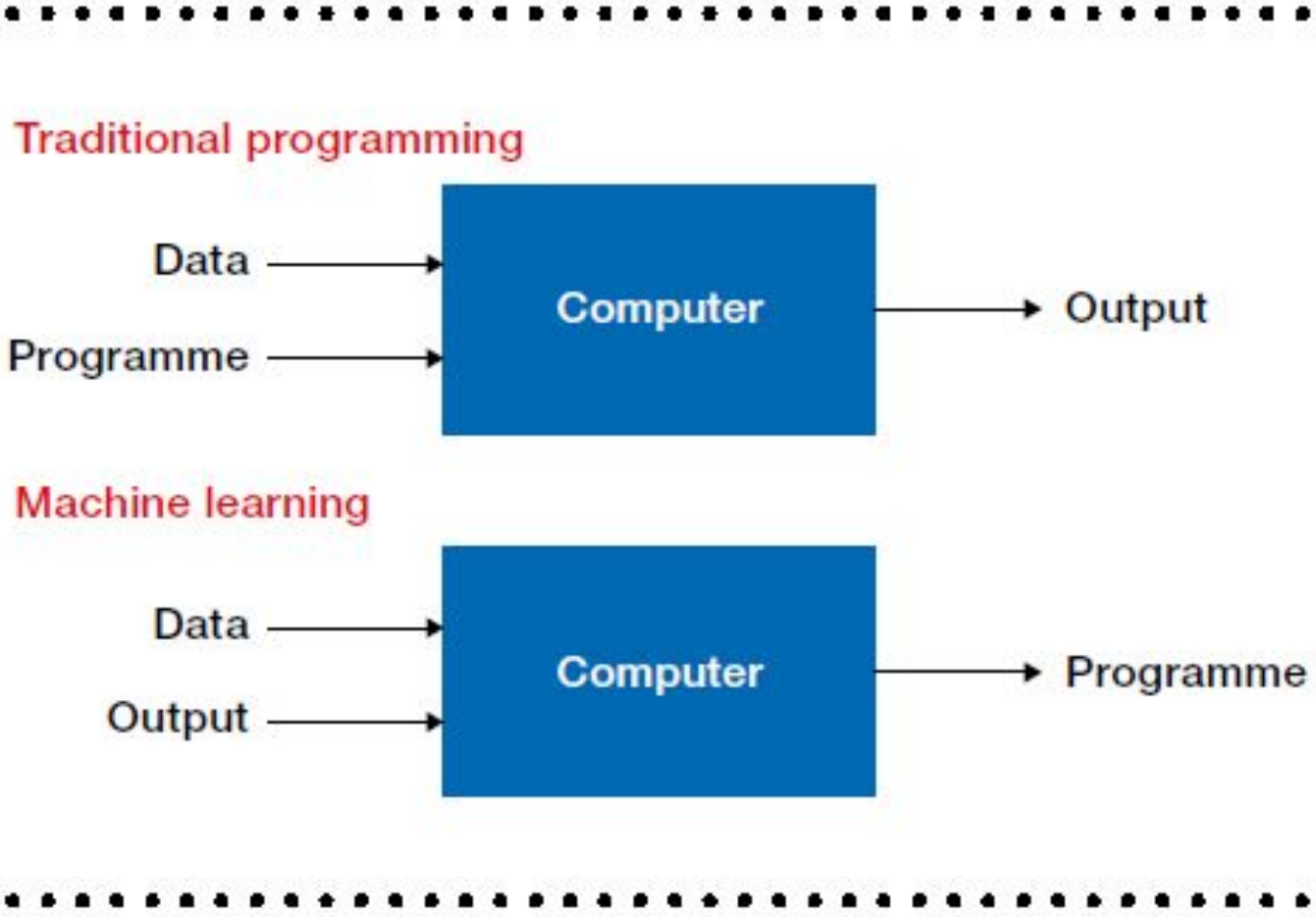


Machine Learning

Developers write a training algorithm, that finds rules, which produce the desired output.



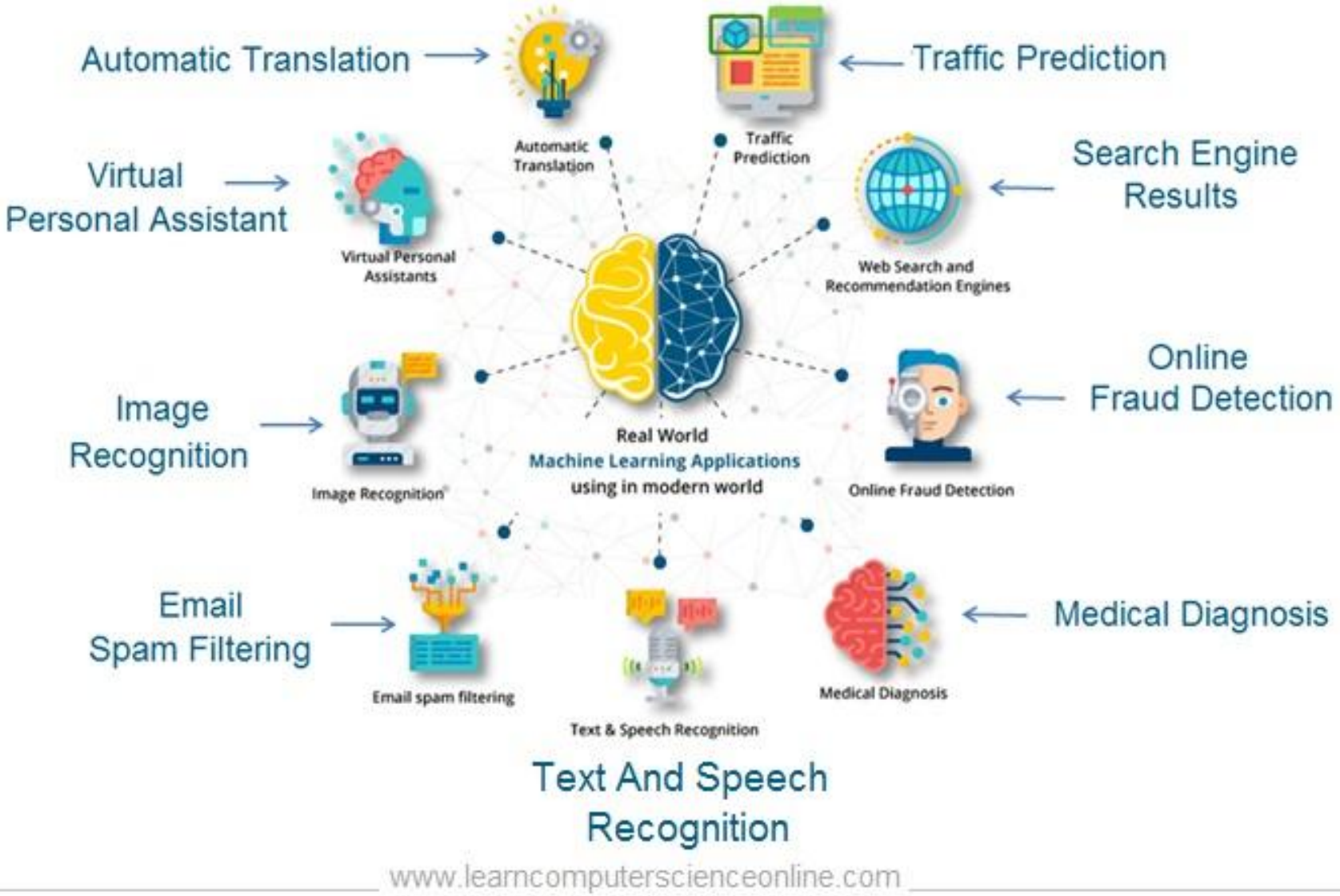
Traditional Programming Vs Machine Learning



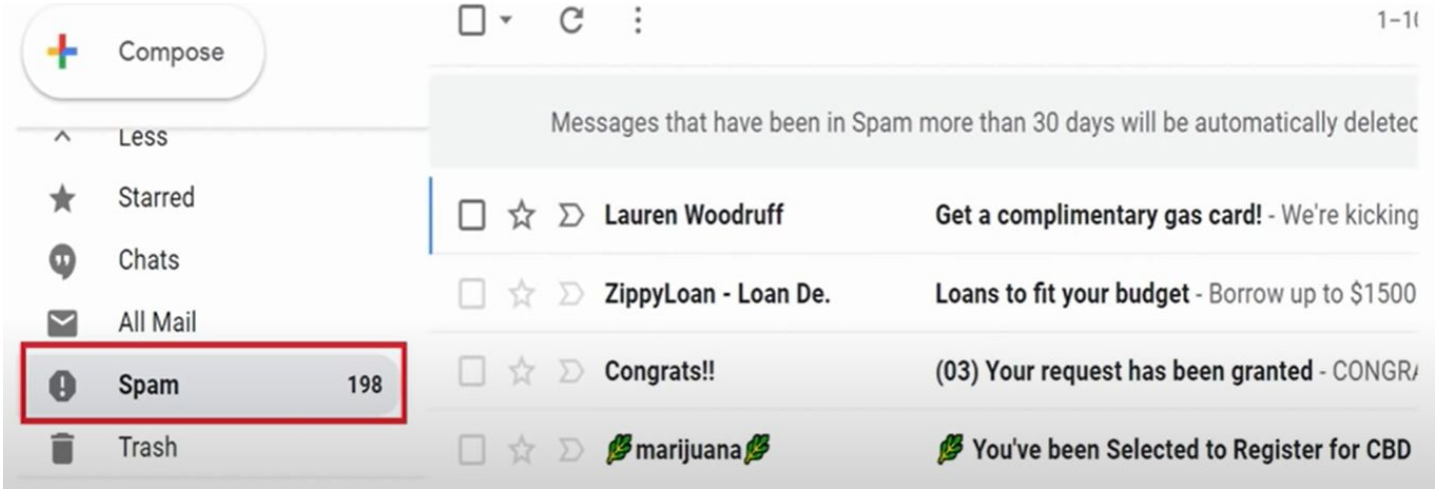
Some Task

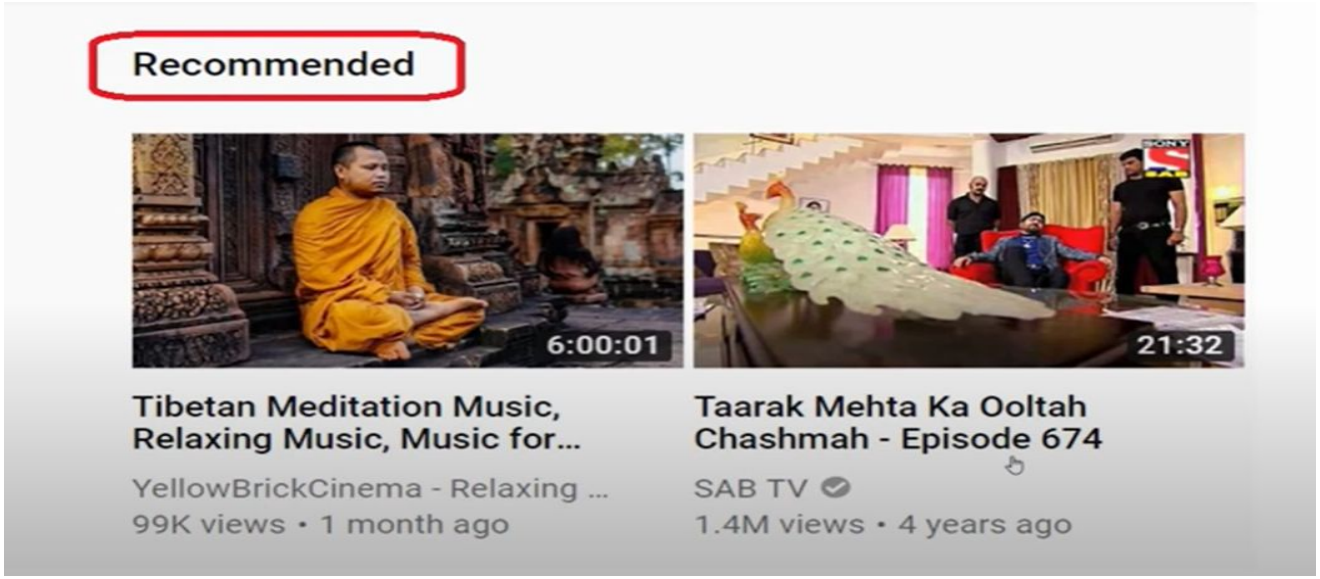
284029834098 * 845609845960

Applications

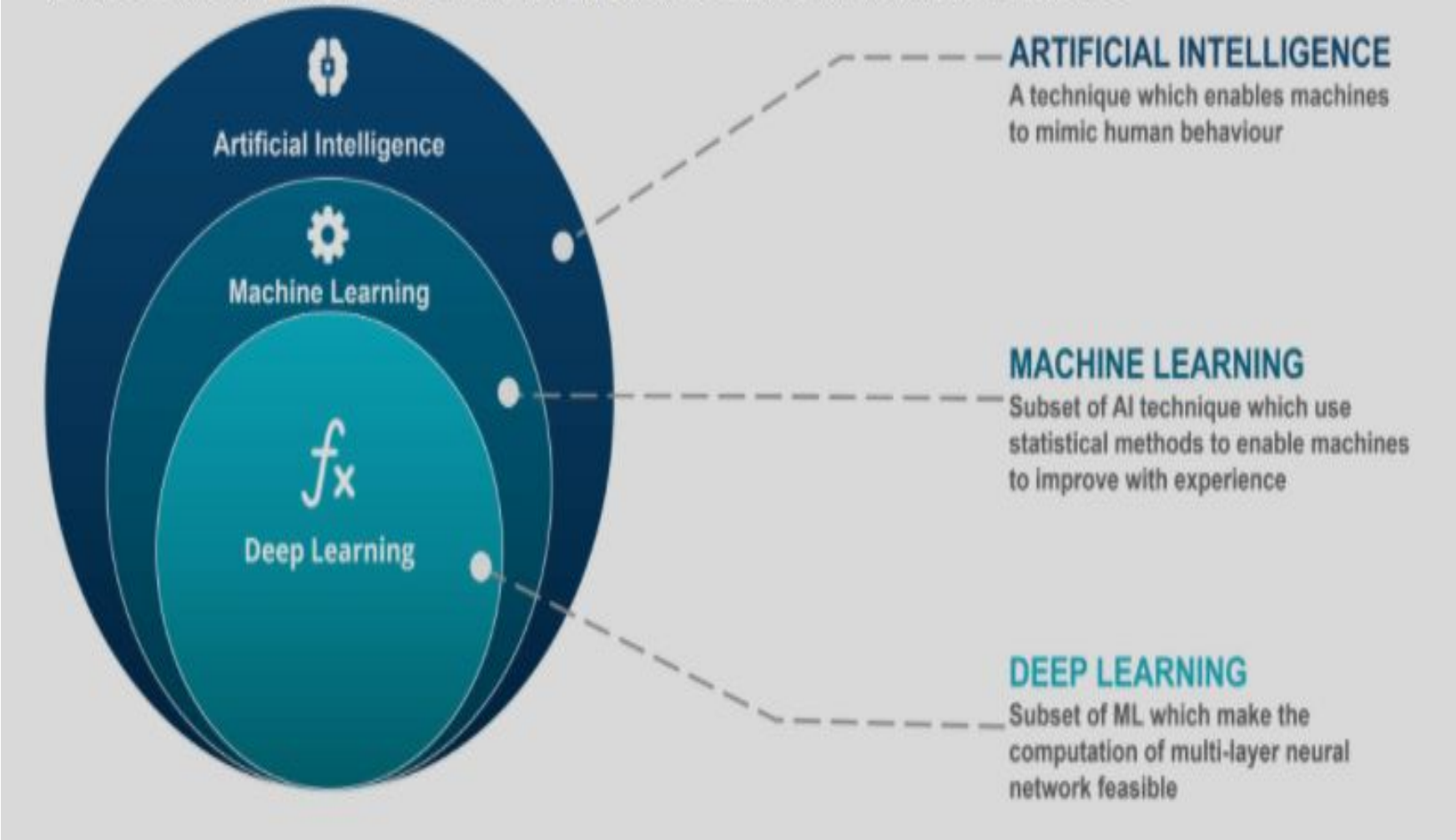


Applications





AI (superset) ML (superset) DL



Terminologies of Machine Learning

- **Model** A model is a **specific representation** learned from data by applying some machine learning algorithm. A model is also called **hypothesis**.

- **Feature** A feature is an individual measurable property of our data. A set of numeric features can be conveniently described by a **feature vector**. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, **etc.**

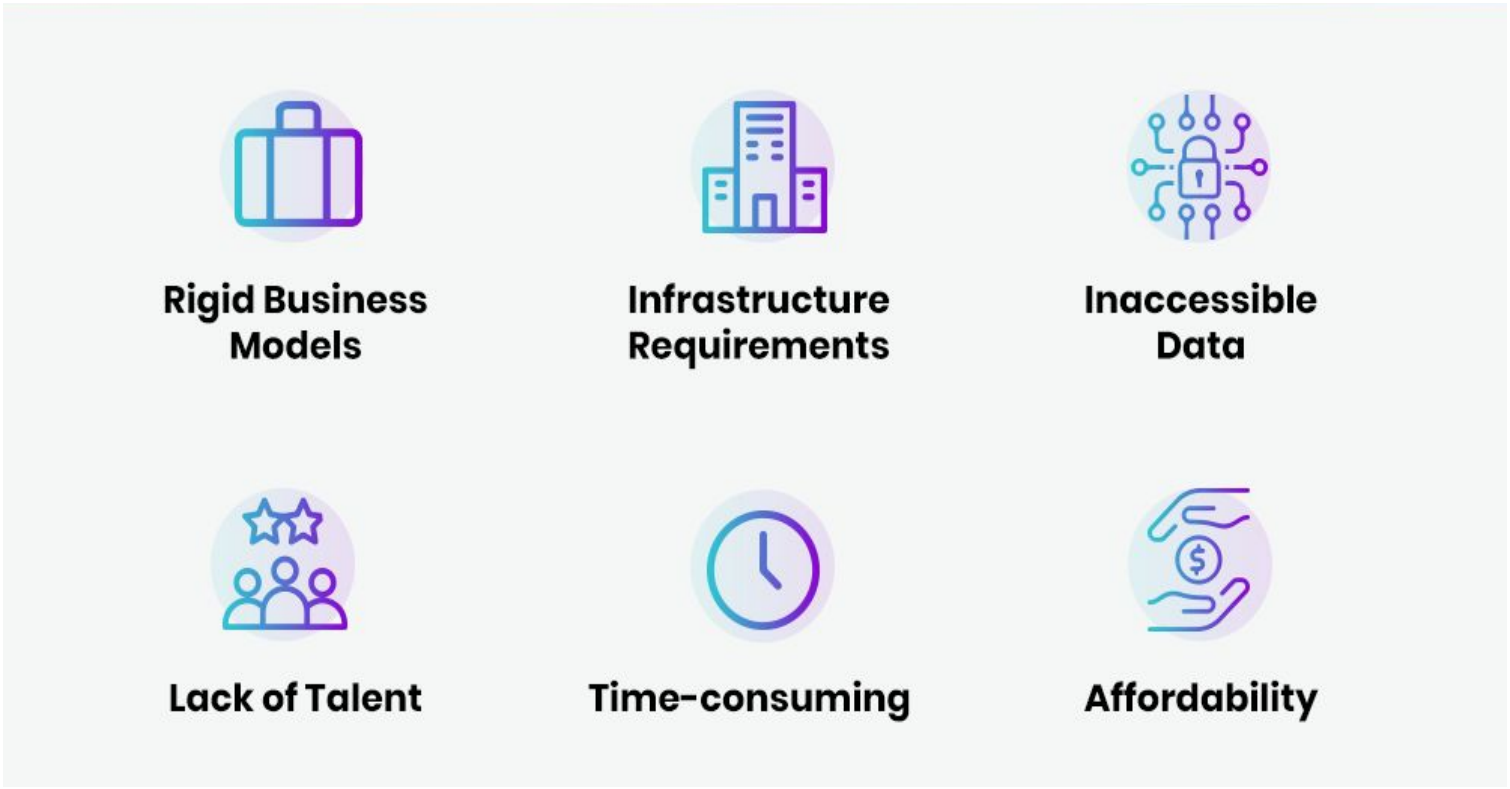
- **Target (Label)** A target variable or label is the value to be predicted by our model. For the fruit example discussed in the features section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.

Terminologies of Machine Learning

- **Training** The idea is to give a set of inputs(features) and it's expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.

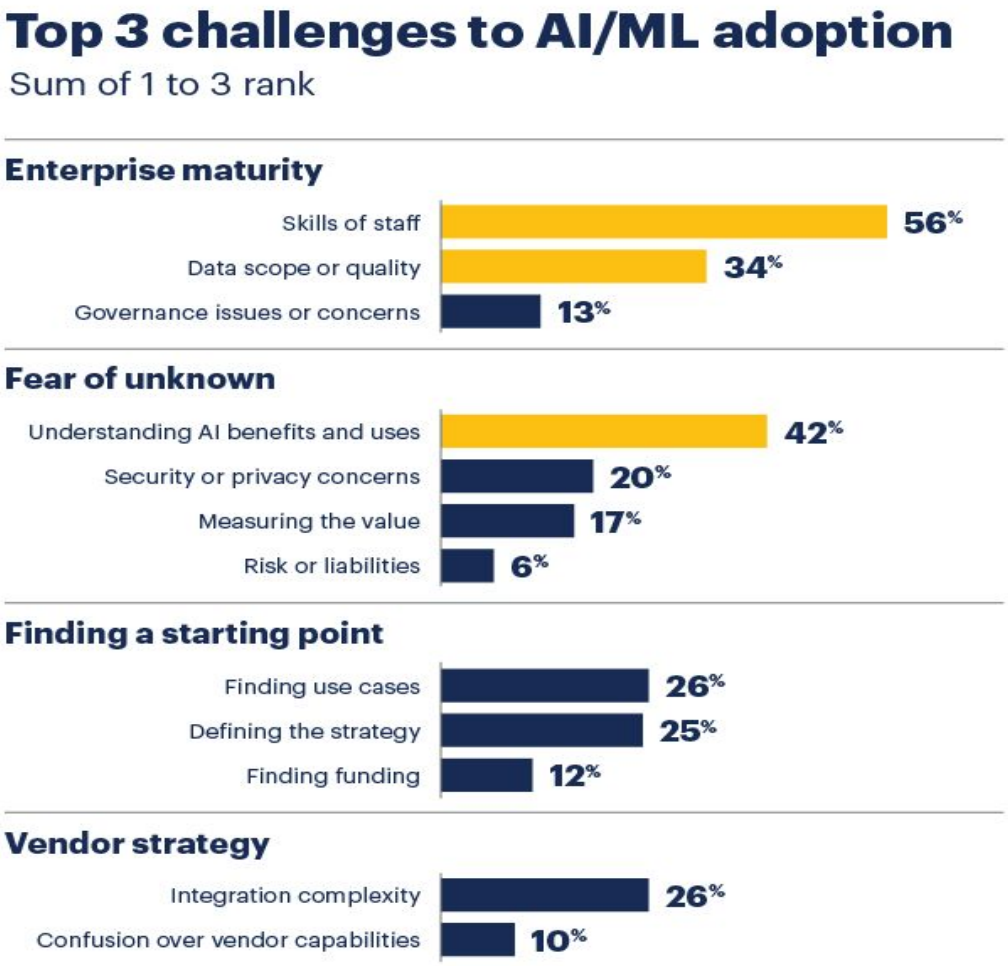
- **Prediction** Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label). But make sure if the machine performs well on unseen data, then only we can say the machine performs well.

Challenges to ML adoption



https://marutitech.com/challenges-machine-learning/

Challenges to ML adoption



https://www.gartner.com/smarterwithgartner/3-barriers-to-ai-adoption

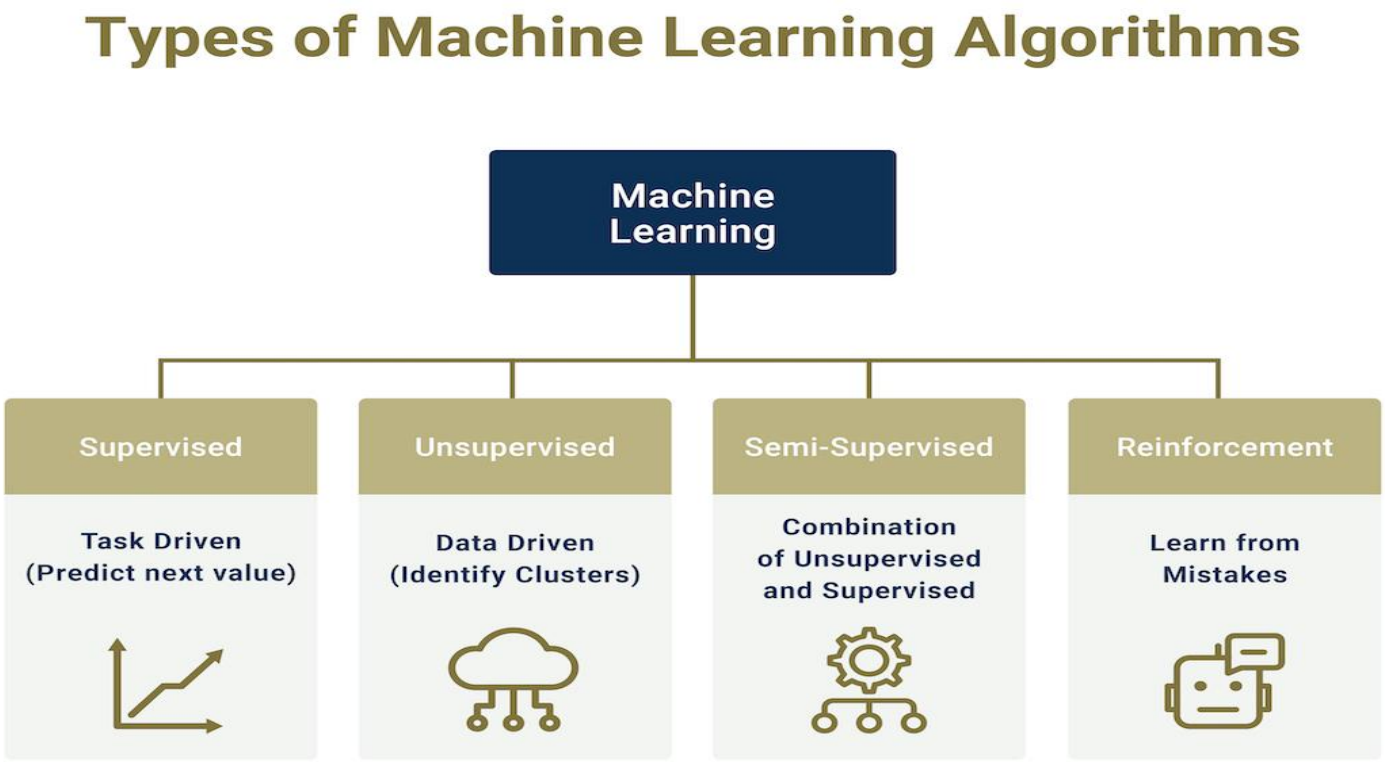
Types of Machine Learning

Supervised Learning

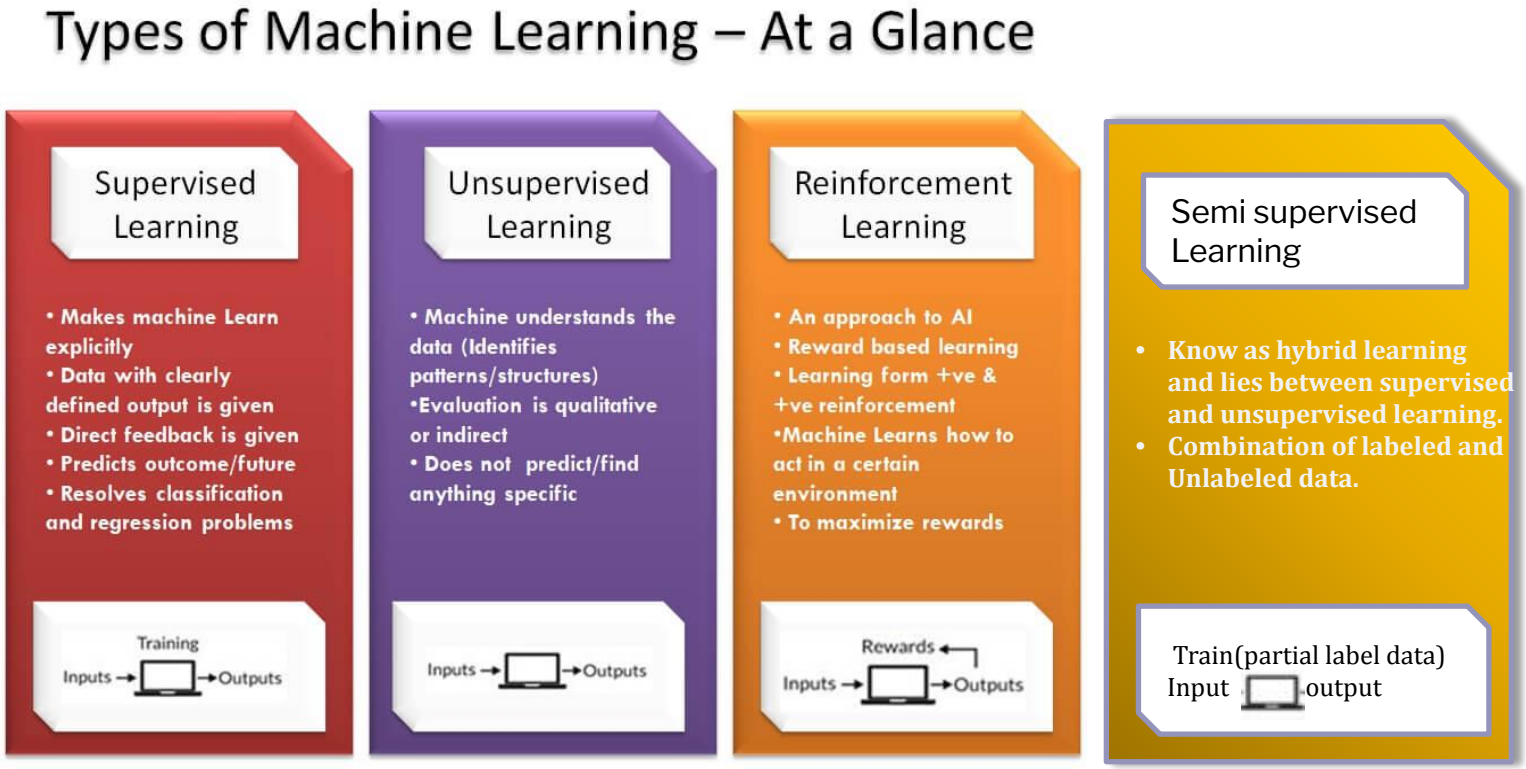
Unsupervised Learning

Reinforcement Learning

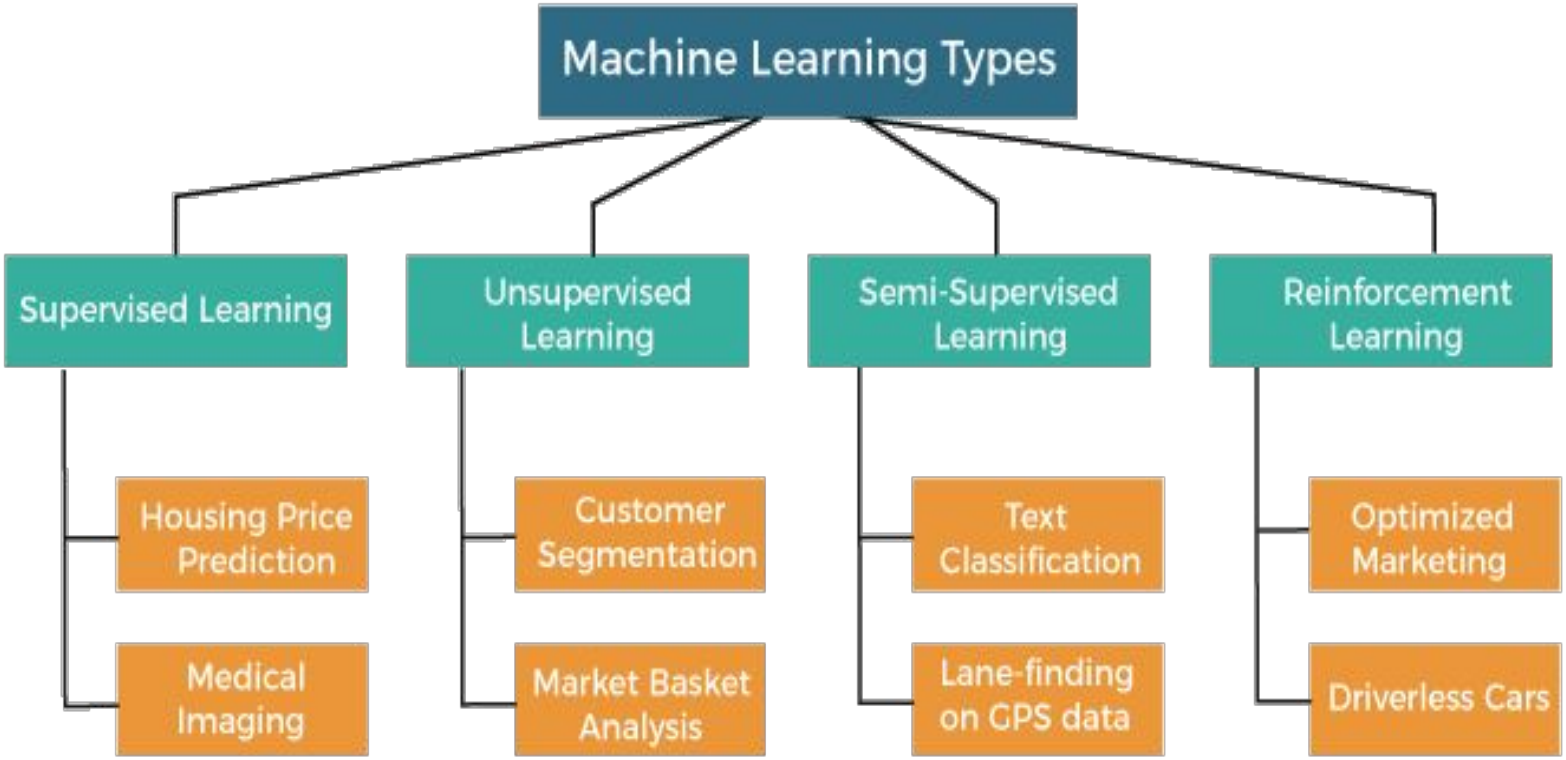
Types of Machine Learning



Types of Machine Learning

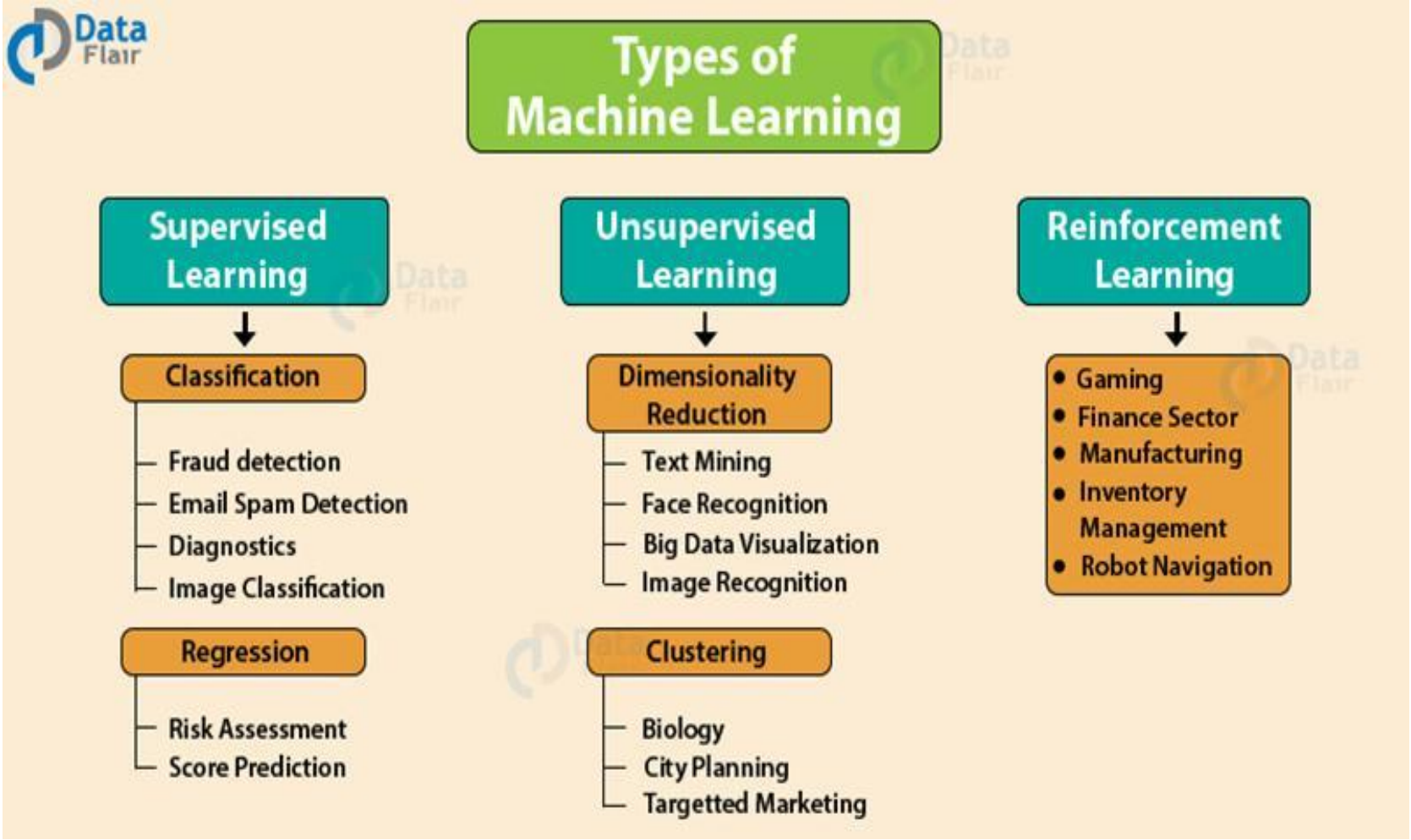


Types of Machine Learning



https://www.javatpoint.com/types-of-machine-learning

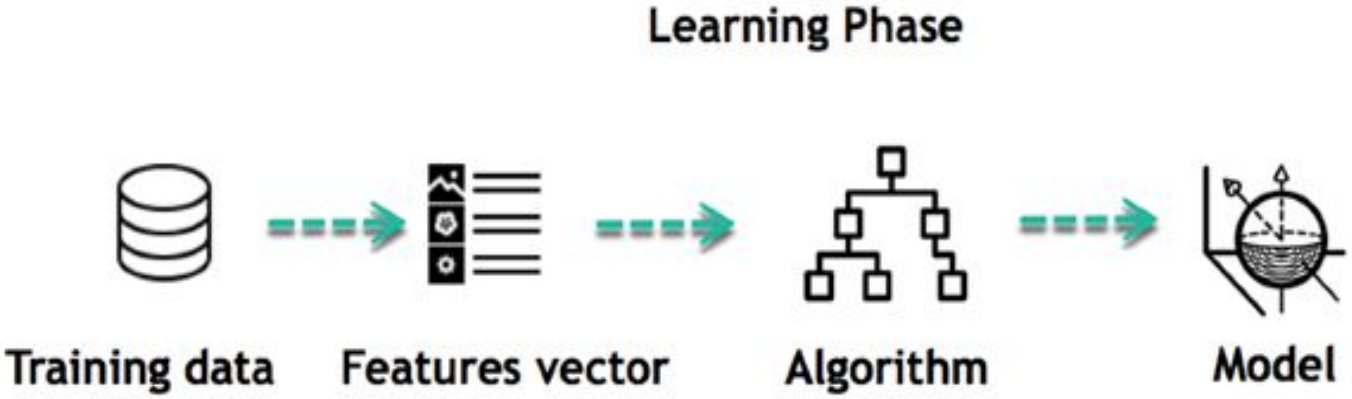
Types of Machine Learning



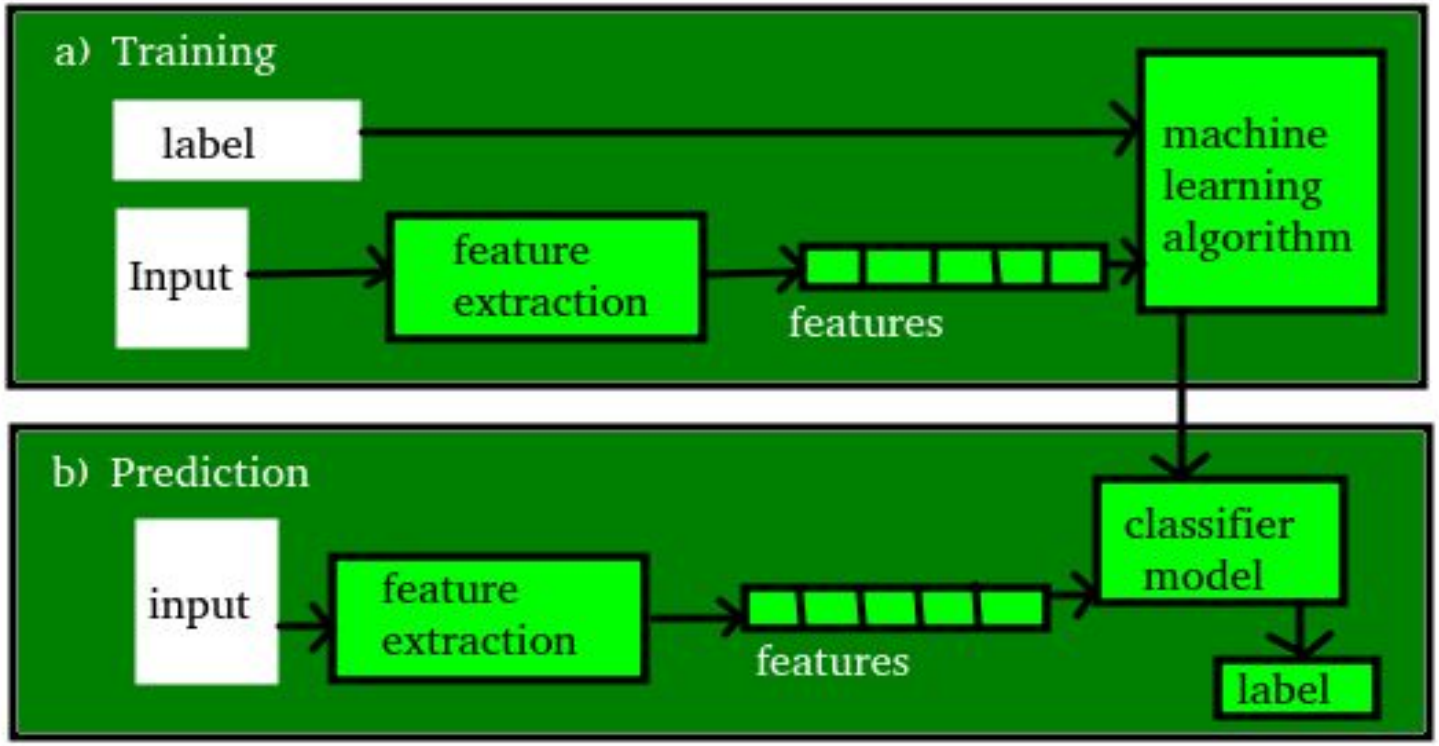
https://www.analytixlabs.co.in/blog/types-of-machine-learning/

Supervised Learning

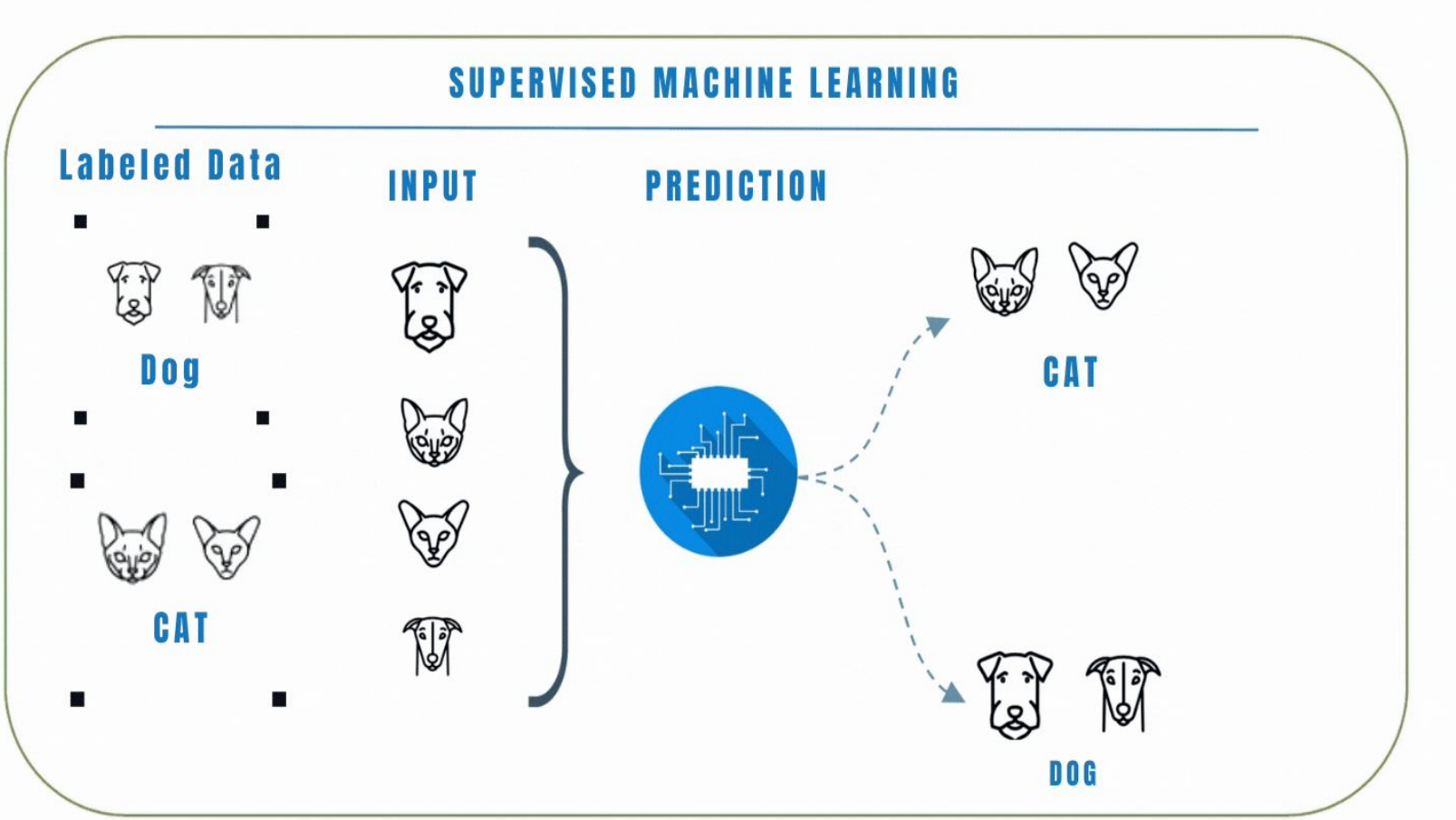
- Supervised Machine Learning is an algorithm that learns from labeled training data to help you predict outcomes for unseen data.
- In supervised learning, we train the machine using data that is well labeled.



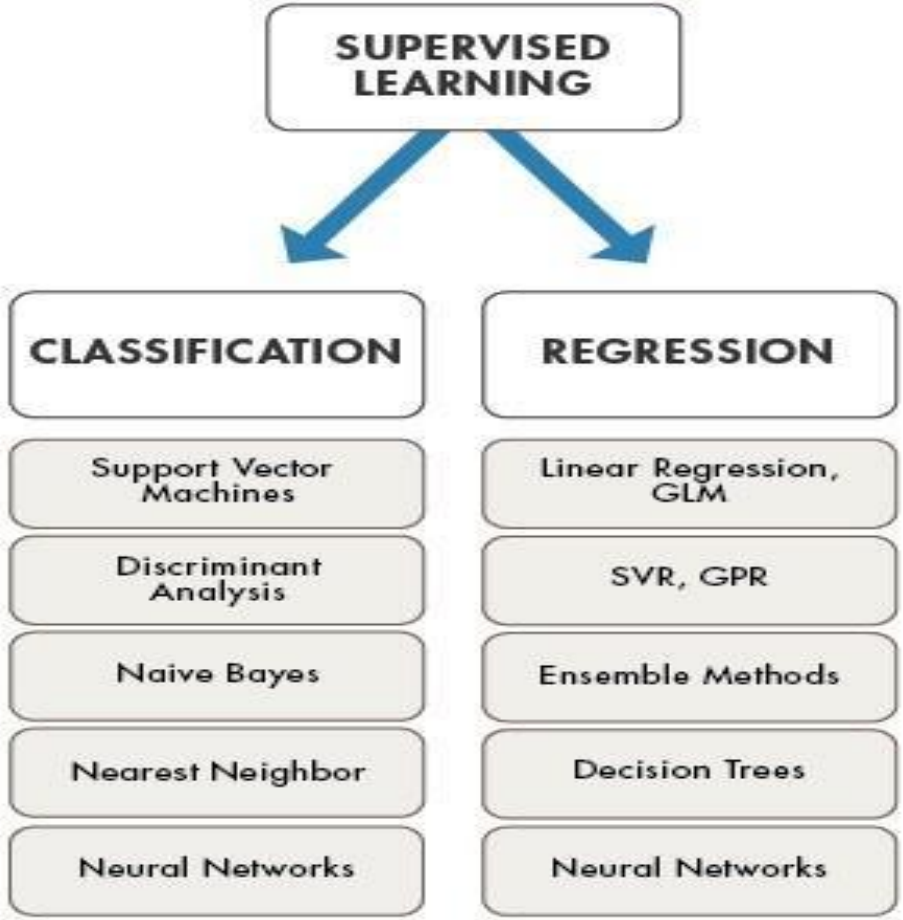
Supervised Machine Learning



Supervised Learning



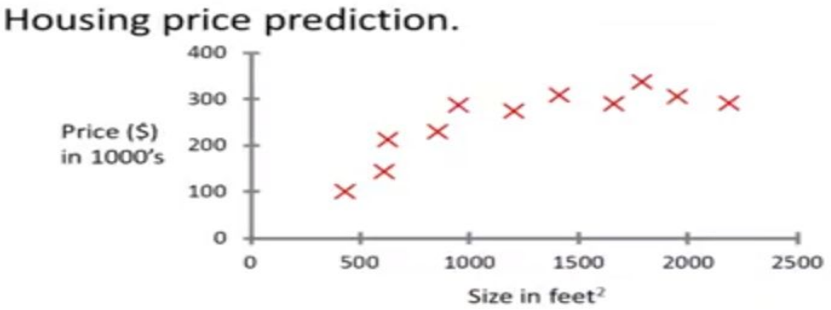
Types of Supervised Learning



Category of Supervised Learning Problems

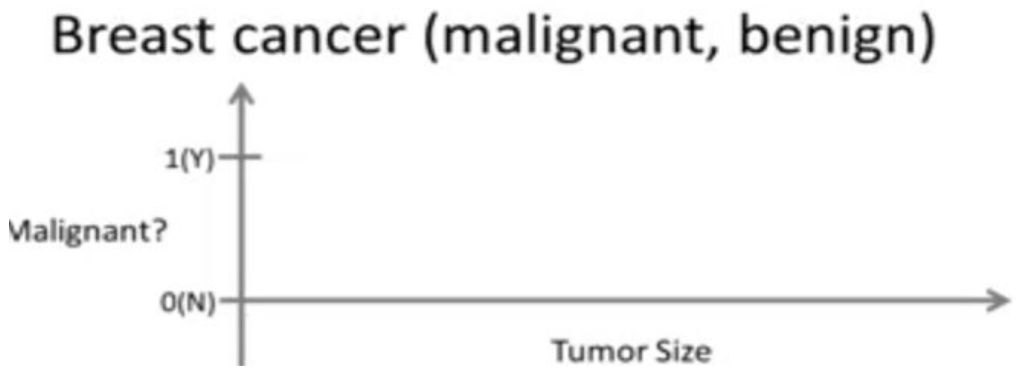
- **Regression:**

  Learning algorithms map input variables to continuous output



- **Classification**

  Learning algorithms map input variables to discrete output

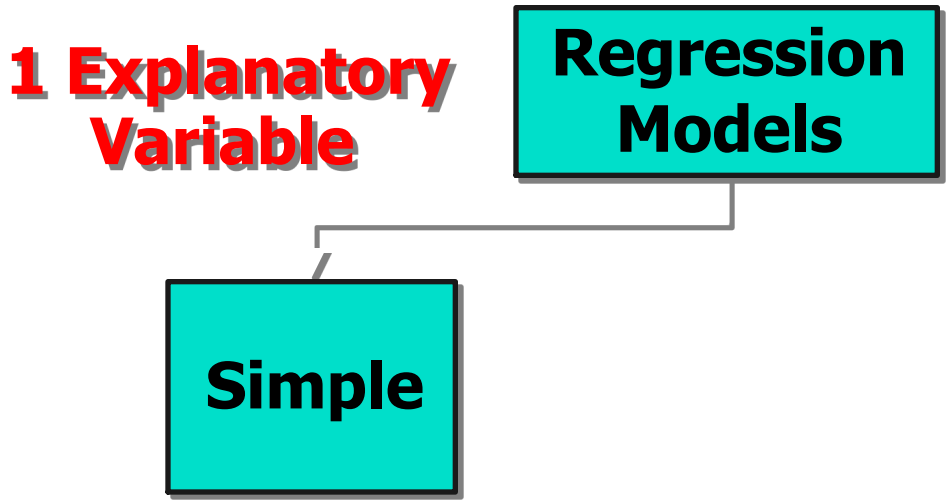


Example

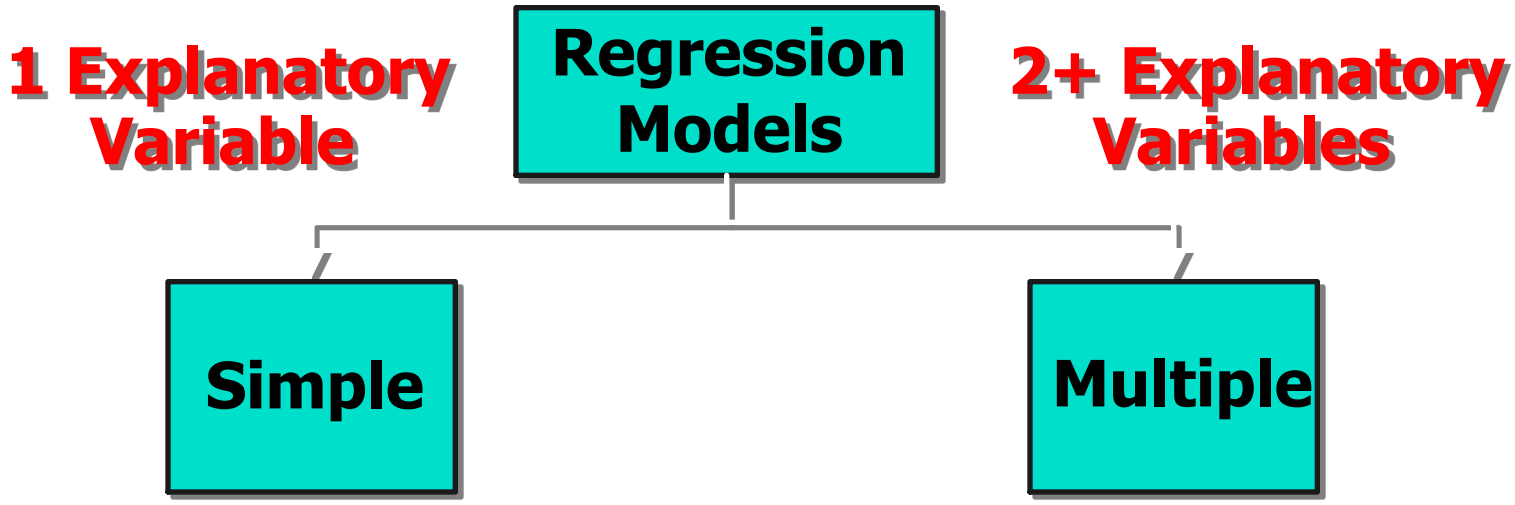

Regression
What will house prices be like in my town next year?
$208K
Price in $
100K 120K 140K 160K 180K 200K 220K 240K 260K 280K 300K
Classification
Will houses be affordable in my town next year?
AFFORDABLE
EXPENSIVE
Price in $
100K 120K 140K 160K 180K 200K 220K 240K 260K 280K 300K

Types of Regression Models

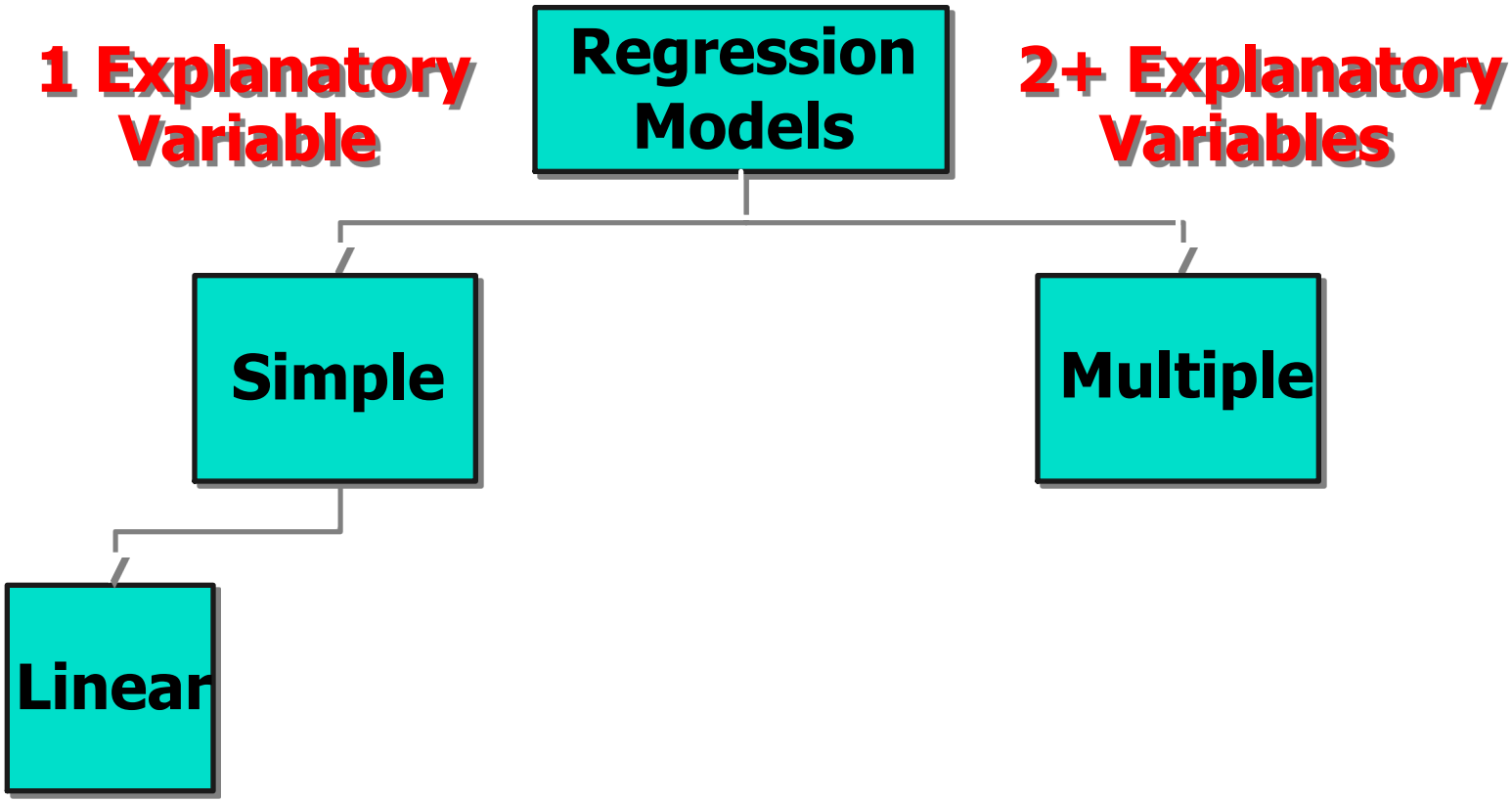
Types of Regression Models



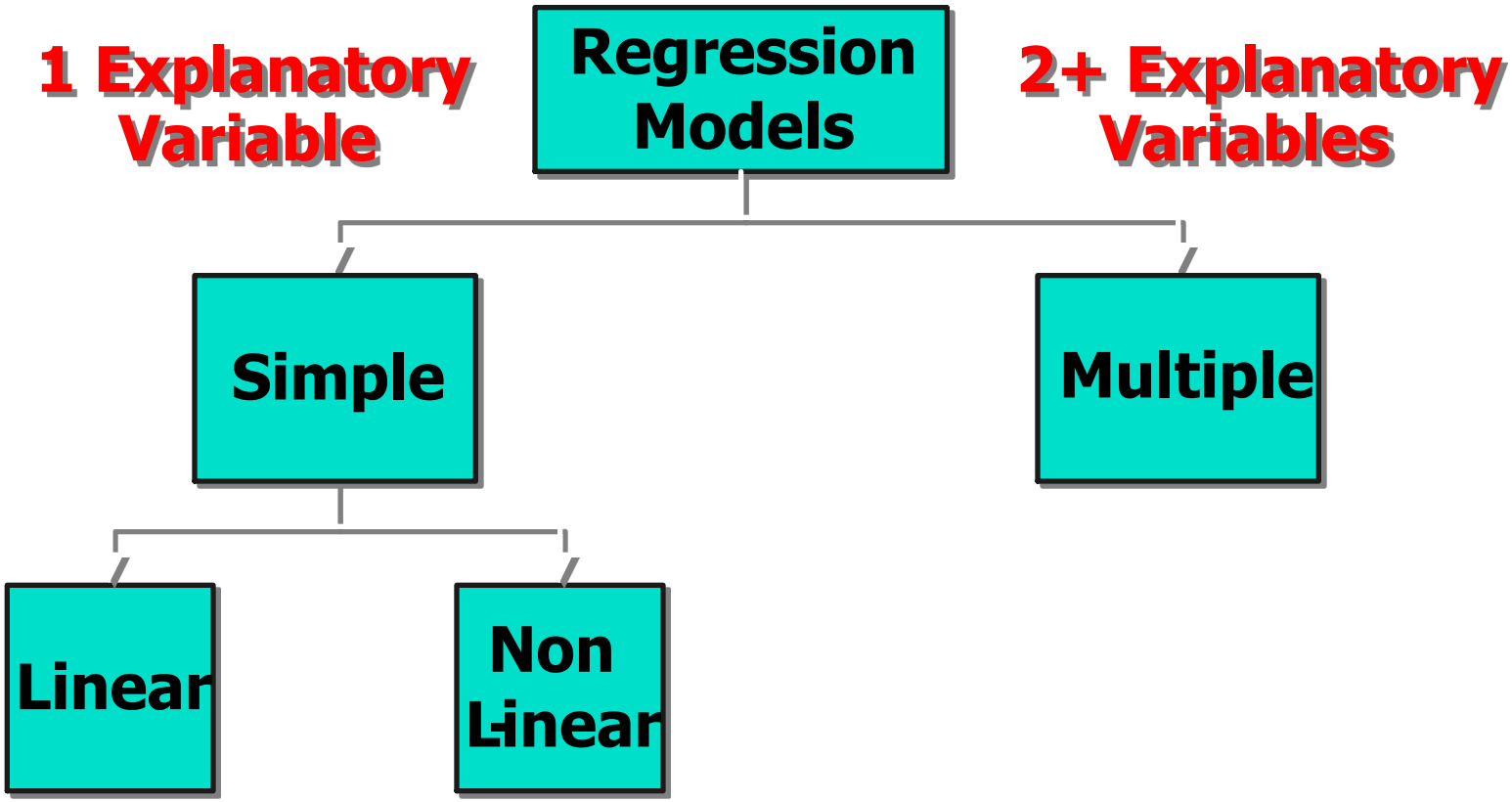
Types of Regression Models



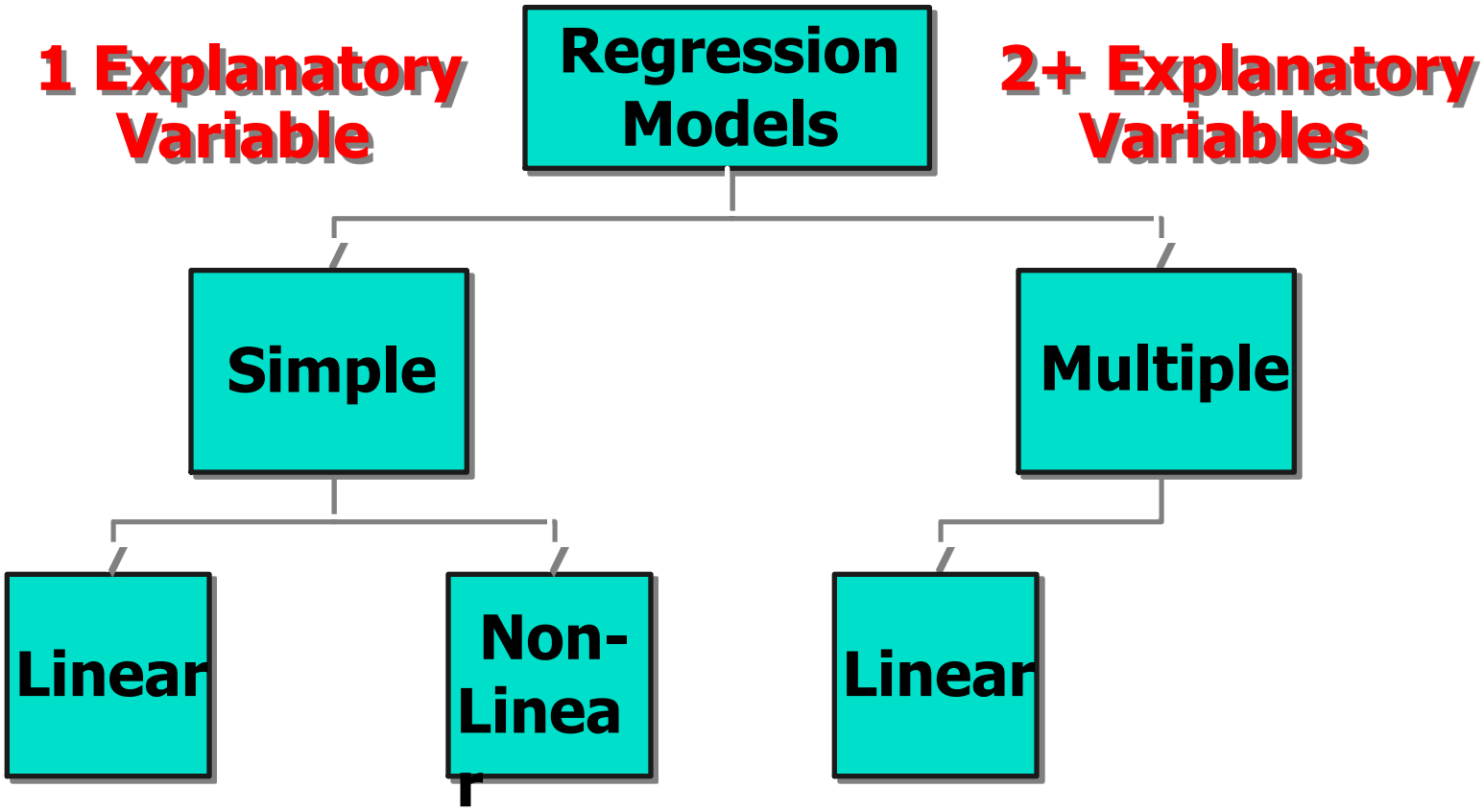
Types of Regression Models



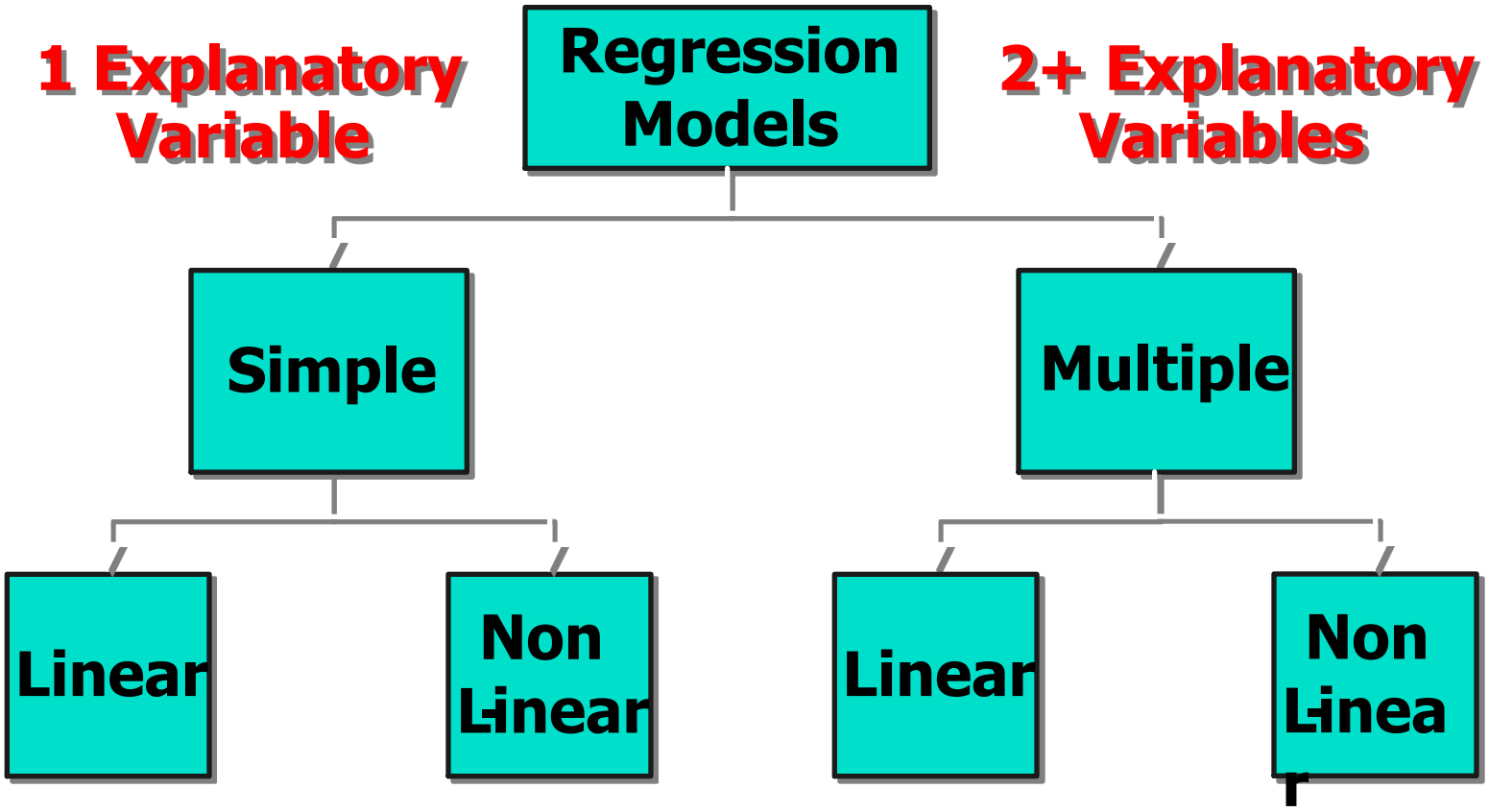
Types of Regression Models



Types of Regression Models



Types of Regression Models

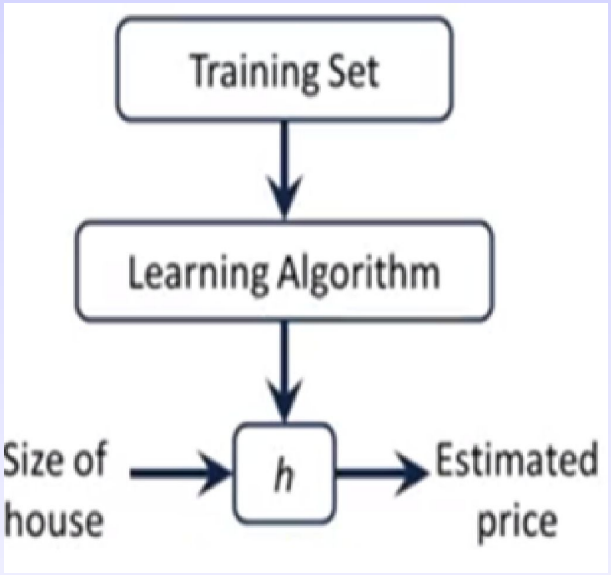


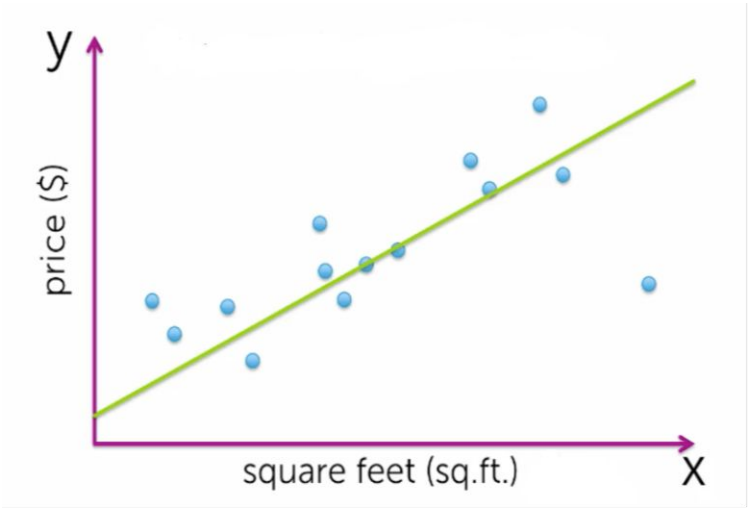
Linear Regression

- In statistical modeling, regression analysis is a set of statistical processes for estimating the relationships between a dependent variable (often called the 'outcome variable') and one or more independent variables (often called 'predictors', 'covariates', or 'features').

- The most common form of regression analysis is linear regression, in which a researcher finds the line (or a more complex linear combination) that most closely fits the data according to a specific mathematical criterion.

Supervised Learning (Linear Regression with one variable)

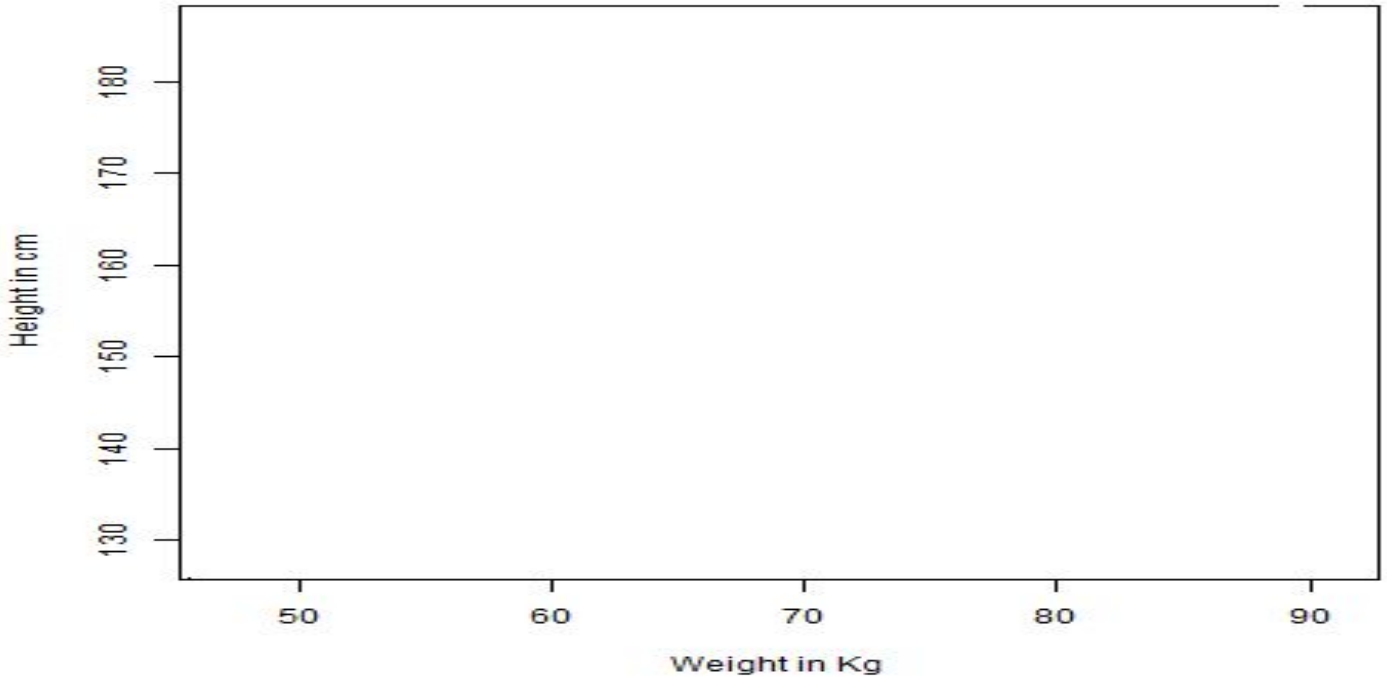




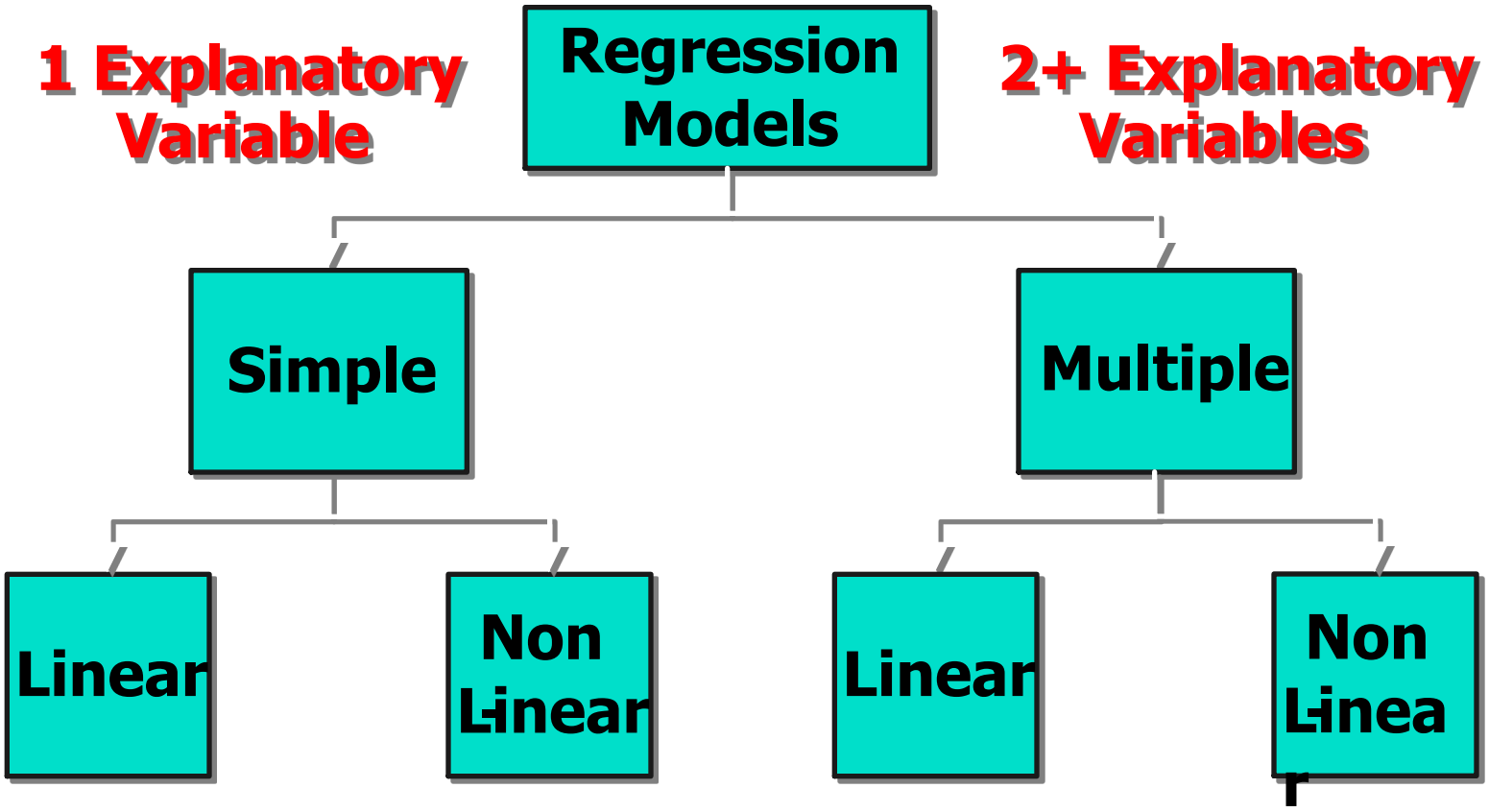
Hypothesis maps x to y

Supervised Learning (Linear Regression)

Values of height: 151, 174, 138, 186, 128, 136, 179, 163, 152, 131
Values of weight: 63, 81, 56, 91, 47, 57, 76, 72, 62, 48



Types of Regression Models

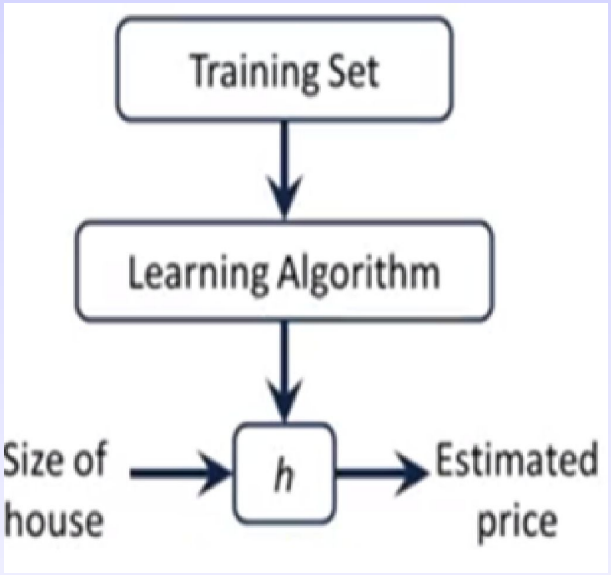


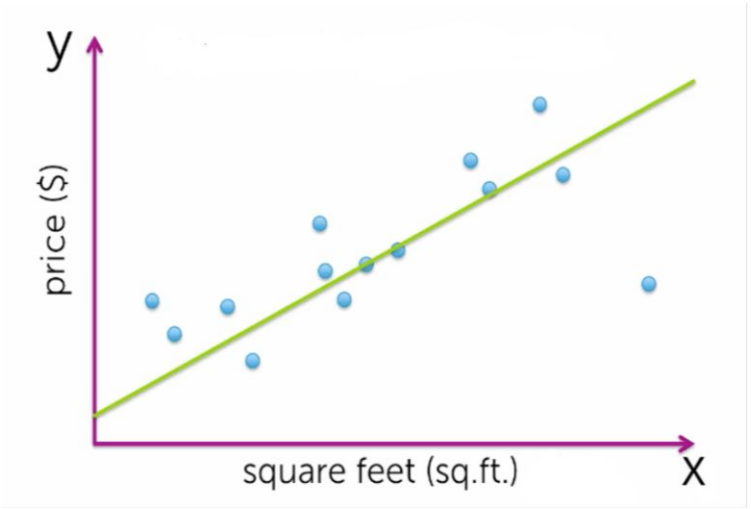
Linear Regression

- In statistical modeling, regression analysis is a set of statistical processes for estimating the relationships between a dependent variable (often called the 'outcome variable') and one or more independent variables (often called 'predictors', 'covariates', or 'features').

- The most common form of regression analysis is linear regression, in which a researcher finds the line (or a more complex linear combination) that most closely fits the data according to a specific mathematical criterion.

Supervised Learning (Linear Regression with one variable)

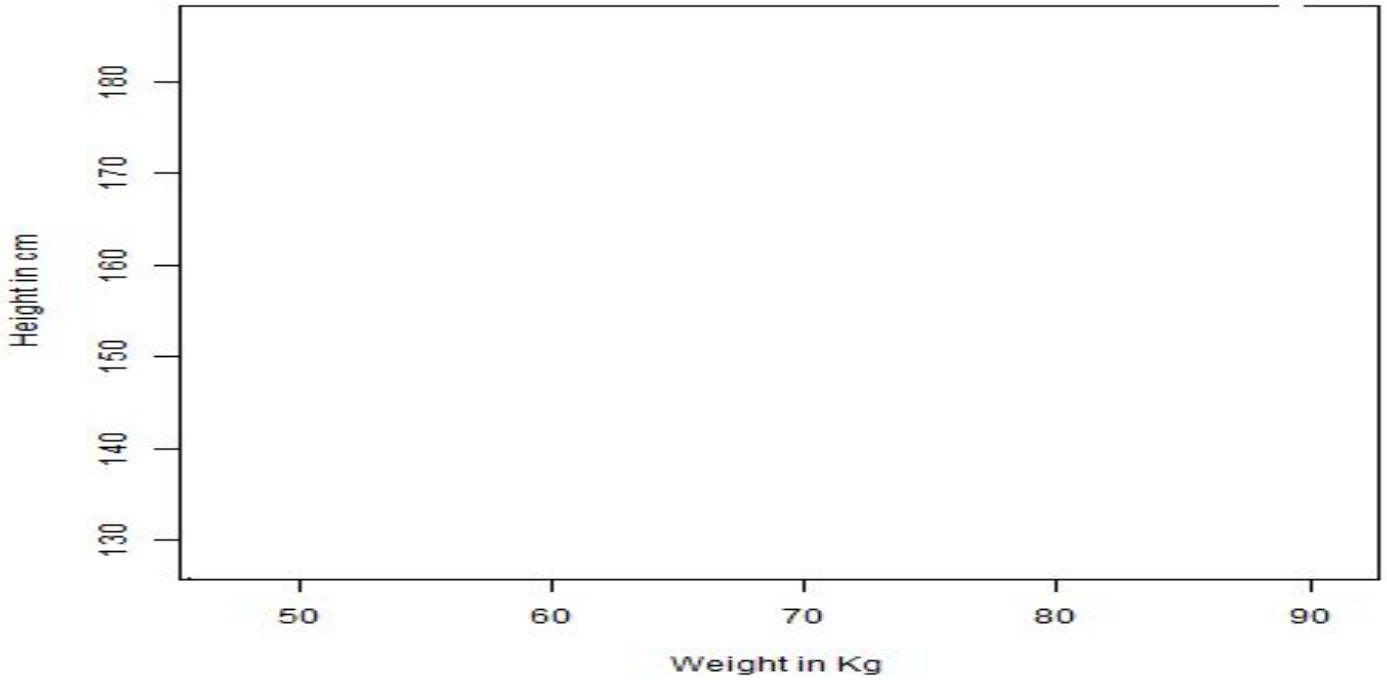




Hypothesis maps x to y

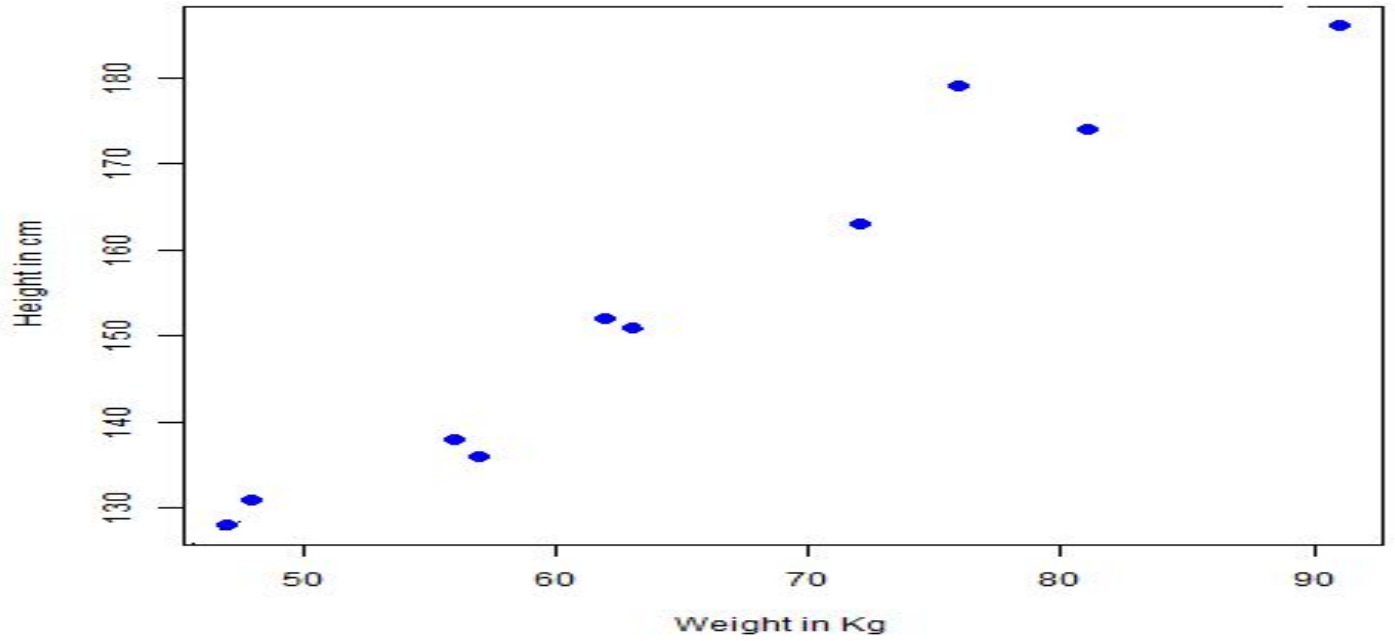
Supervised Learning (Linear Regression)

Values of height: 151, 174, 138, 186, 128, 136, 179, 163, 152, 131
Values of weight: 63, 81, 56, 91, 47, 57, 76, 72, 62, 48



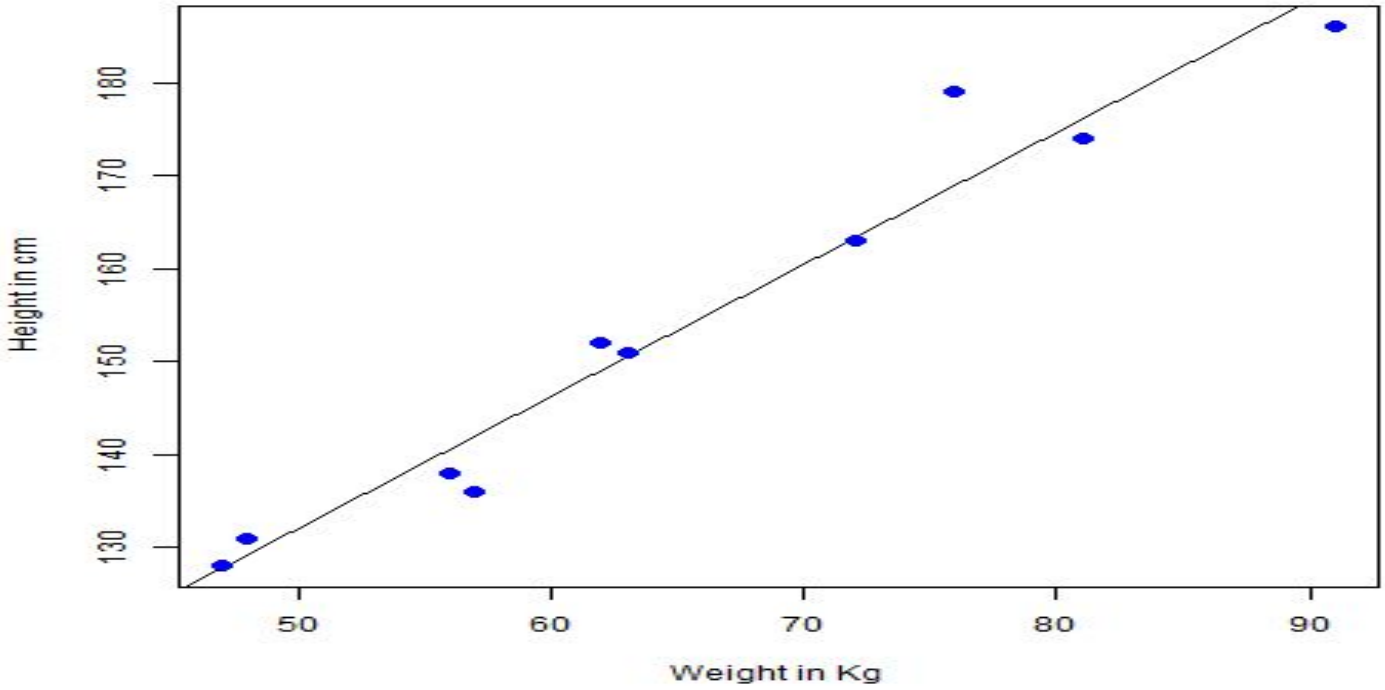
Supervised Learning (Linear Regression)

 # Values of height: 151, 174, 138, 186, 128, 136, 179, 163, 152, 131
 # Values of weight: 63, 81, 56, 91, 47, 57, 76, 72, 62, 48



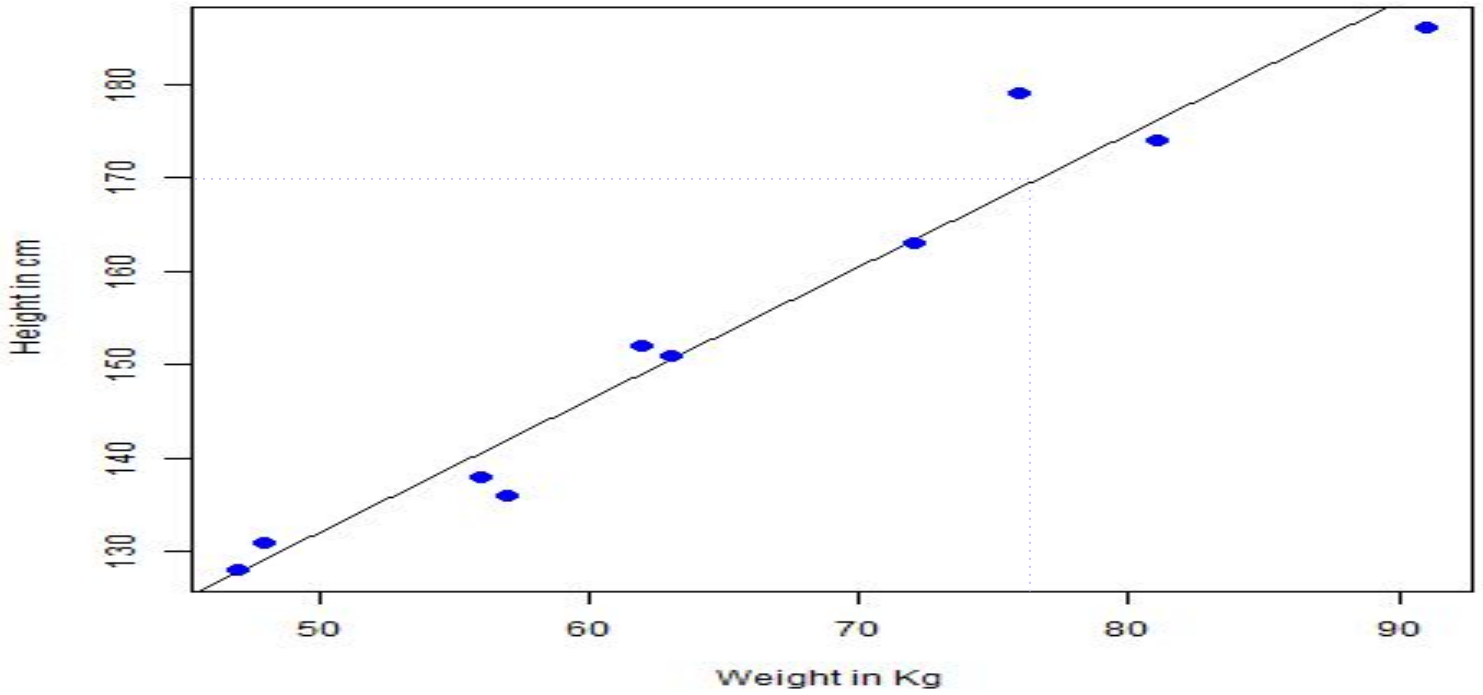
Supervised Learning (Linear Regression)

Values of height: 151, 174, 138, 186, 128, 136, 179, 163, 152, 131
Values of weight: 63, 81, 56, 91, 47, 57, 76, 72, 62, 48

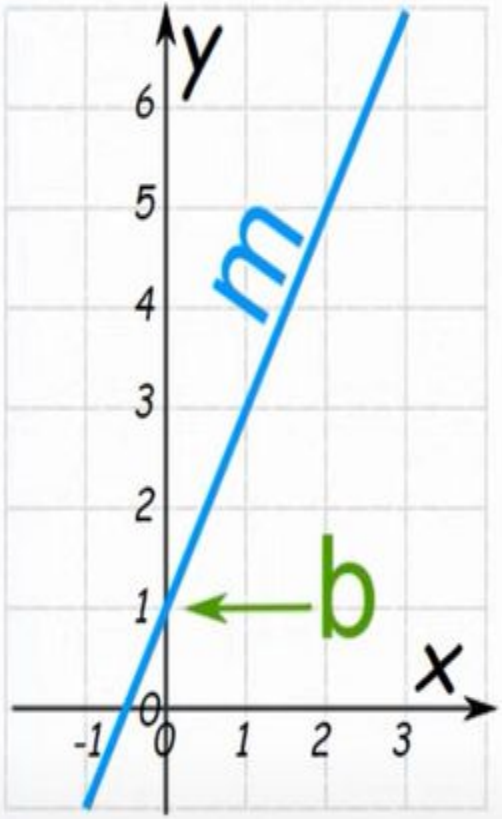


Supervised Learning (Linear Regression)

Values of height: 170
Values of weight: ??



Linear Regression



$$price = m * area + b$$

$$y = mx + b$$

Slope (or Gradient) Y Intercept

What is Regression?

- Regression is similar to classification
  - First, construct a model
  - Second, use model to predict unknown value
    - Major method for prediction is regression
      - Linear and multiple regression
      - Non-linear regression
- Prediction is different from classification
  - Classification refers to predict categorical class label
  - Prediction models continuous-valued functions

Linear regression

- Linear dependence: constant rate of increase of one variable with respect to another (as opposed to, e.g., diminishing returns).
- Regression analysis describes the relationship between two (or more) variables.
- Examples:
  - Income and educational level
  - Demand for electricity and the weather
  - Home sales and interest rates

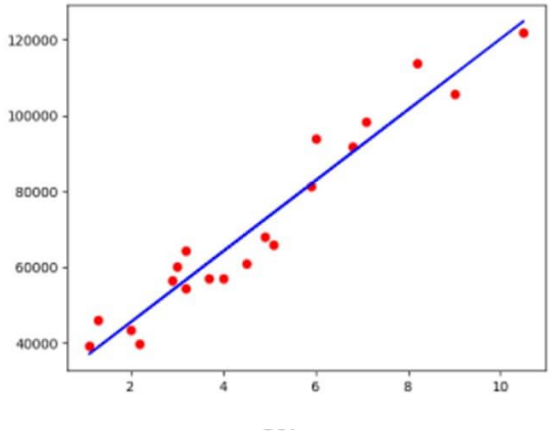
Simple Linear Regression

- Linear regression is the most basic form of regression algorithms in machine learning. The model consists of a single parameter and a dependent variable has a linear relationship.

- When the number of independent variables increases, it is called the multiple linear regression models.

- We denote simple linear regression by the following equation given below.

$$y = mx + c$$



Example

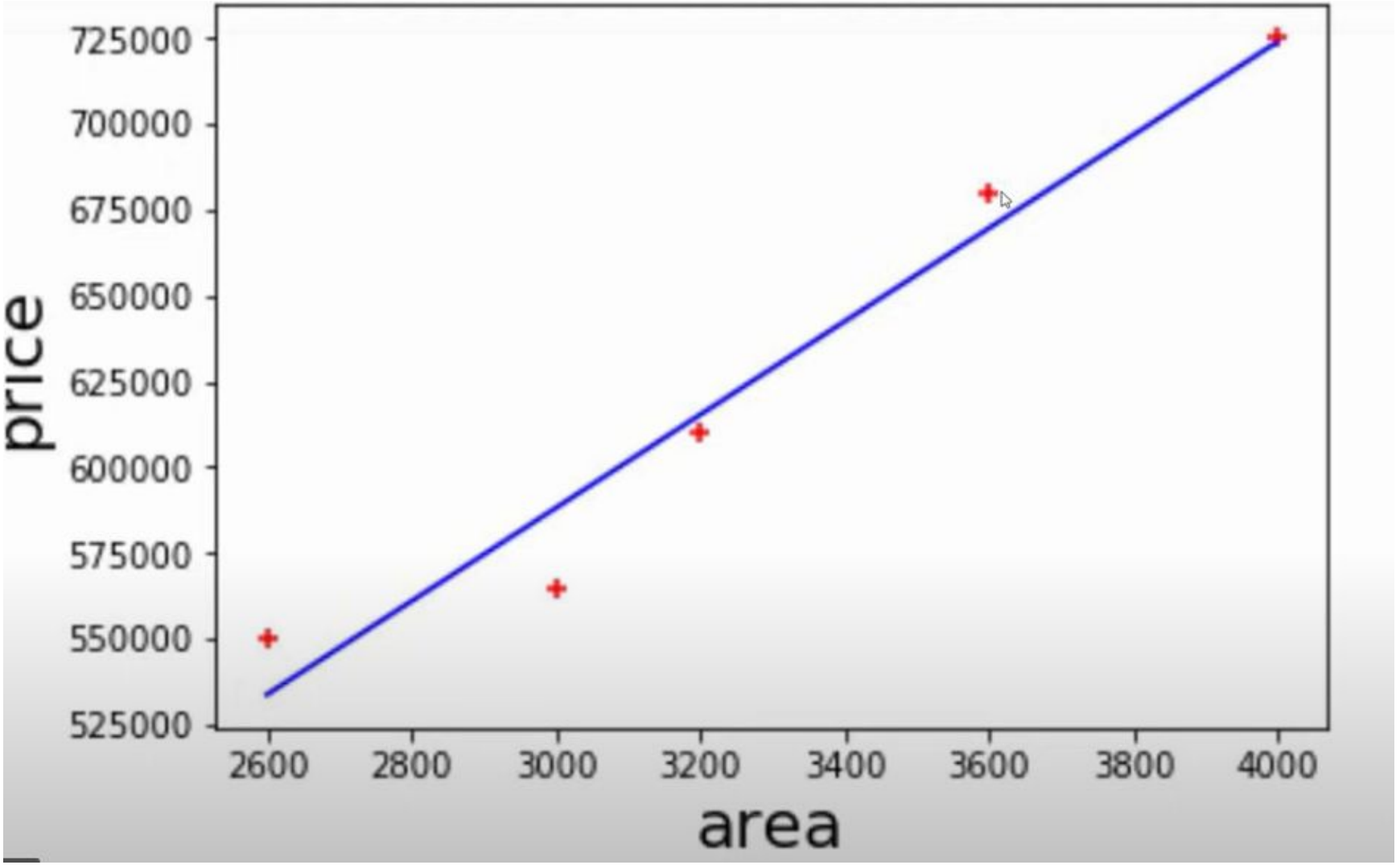
Home prices in Monroe Twp, NJ (USA)

area	price
2600	550000
3000	565000
3200	610000
3600	680000
4000	725000

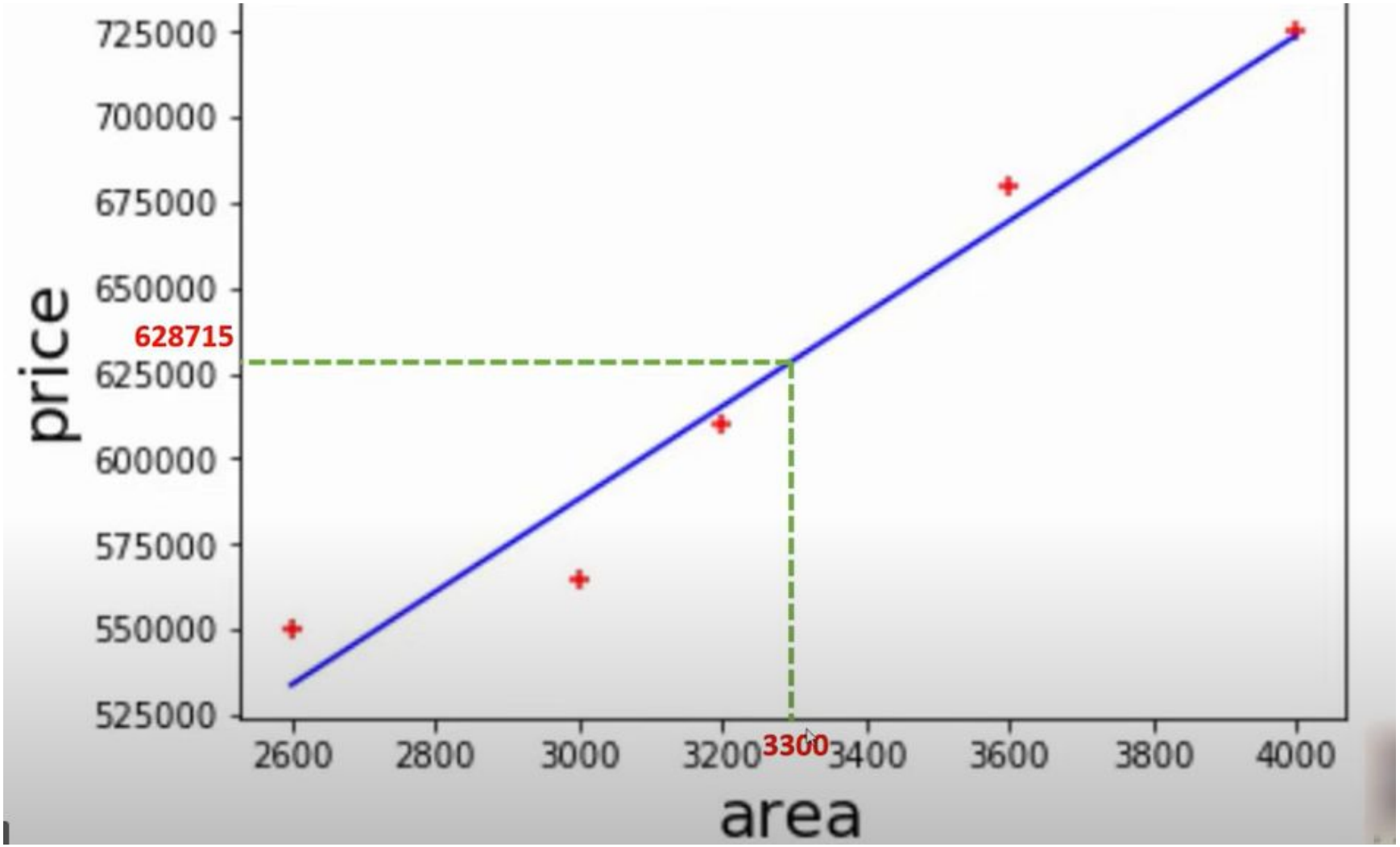
Given these home prices find out prices of homes whose area is,

3300 square feet
5000 square feet

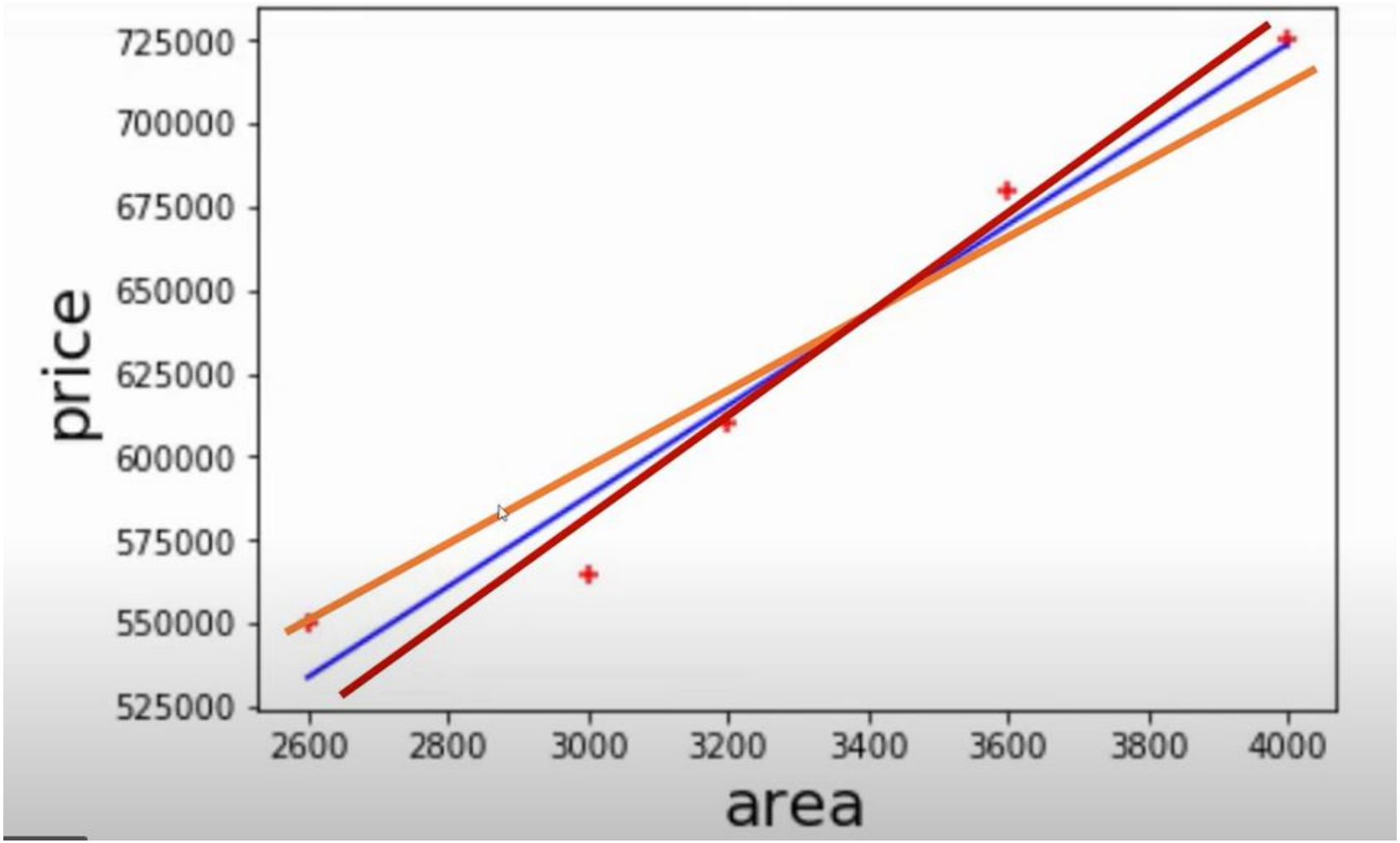
Example


725000
700000
675000
650000
625000
600000
575000
550000
525000
price
2600
2800
3000
3200
3400
3600
3800
4000
area

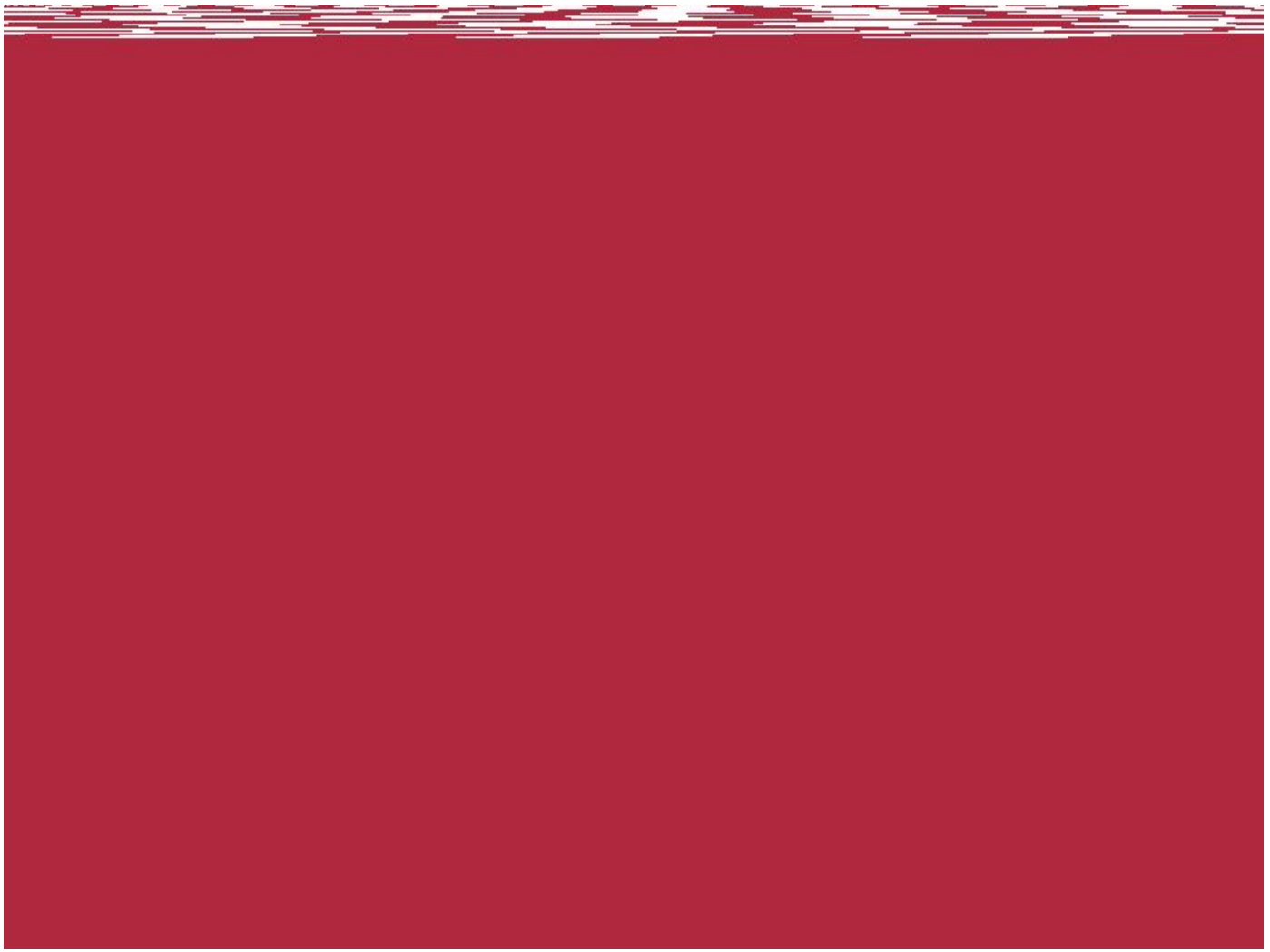
Example


725000
700000
675000
650000
628715
625000
600000
575000
550000
525000
price
2600
2800
3000
3200
3300
3400
3600
3800
4000
area

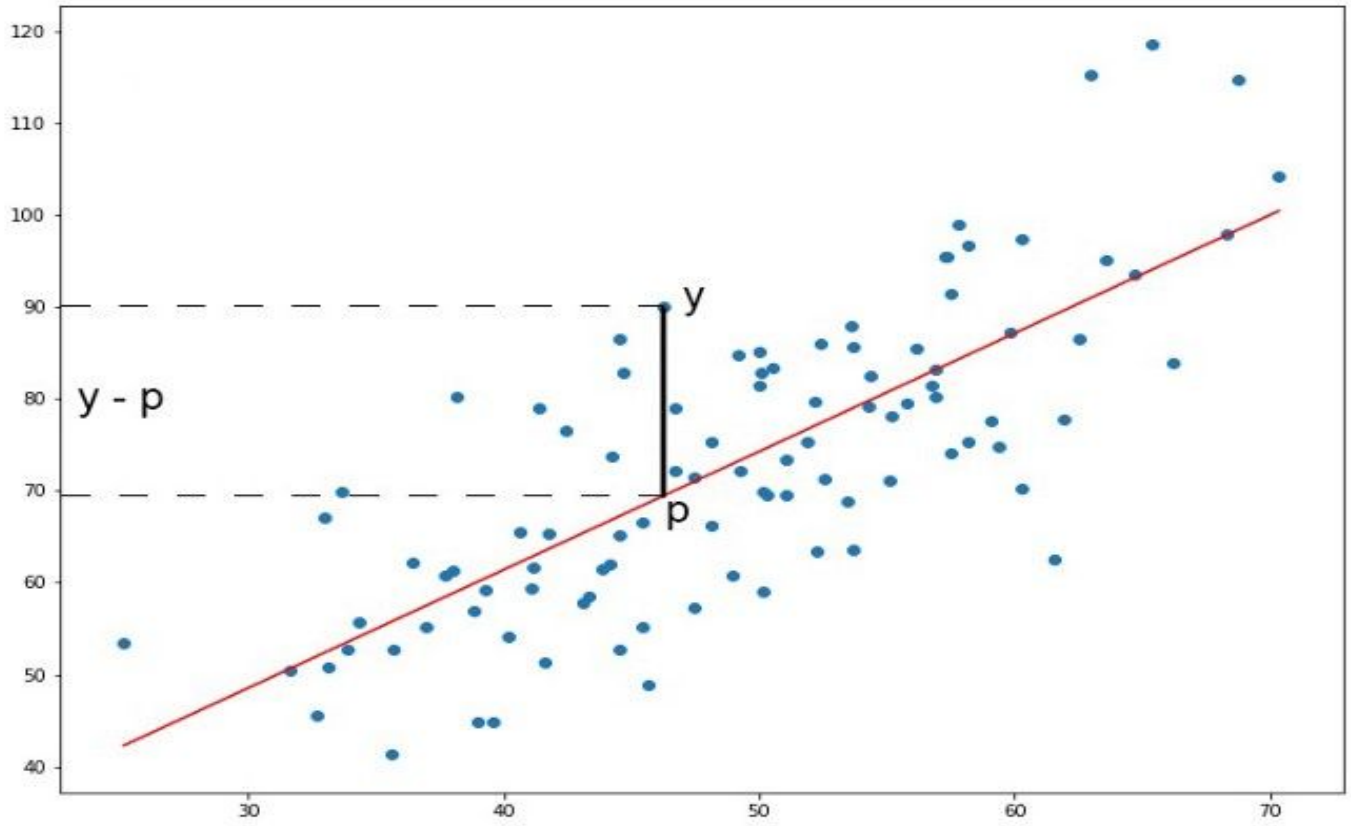
Example


725000
700000
675000
650000
625000
600000
575000
550000
525000
price
2600
2800
3000
3200
3400
3600
3800
4000
area

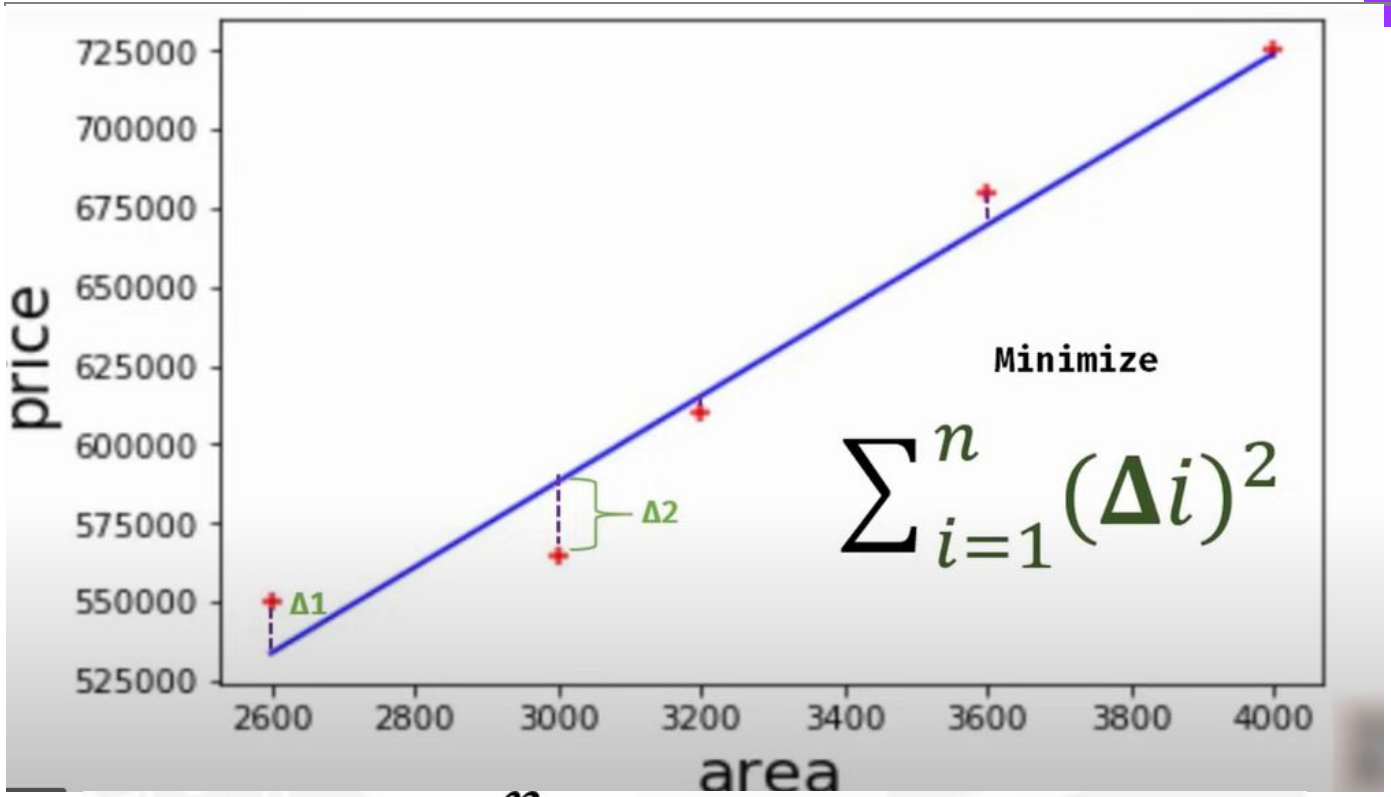
Best Fit Line



Error



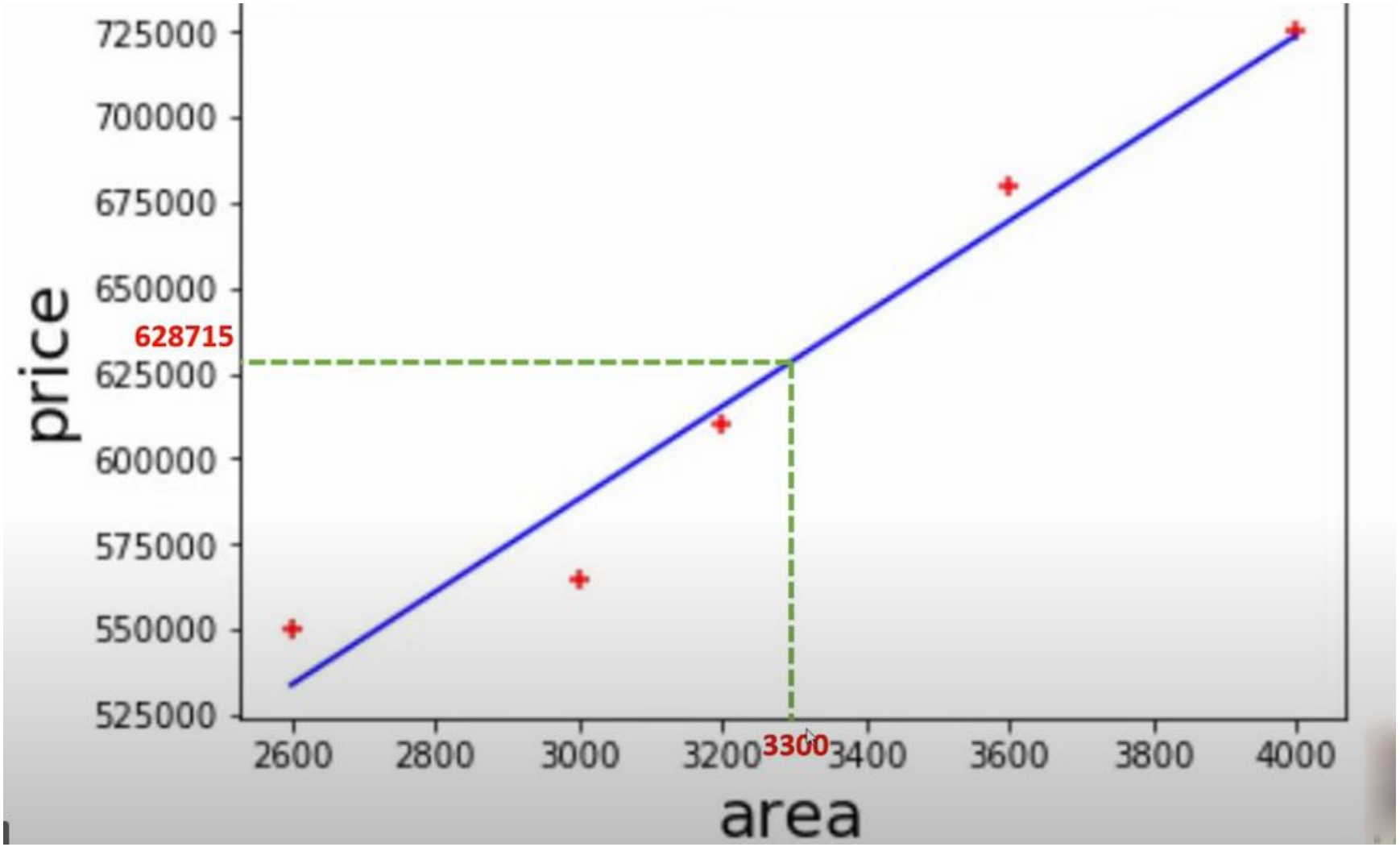
Error



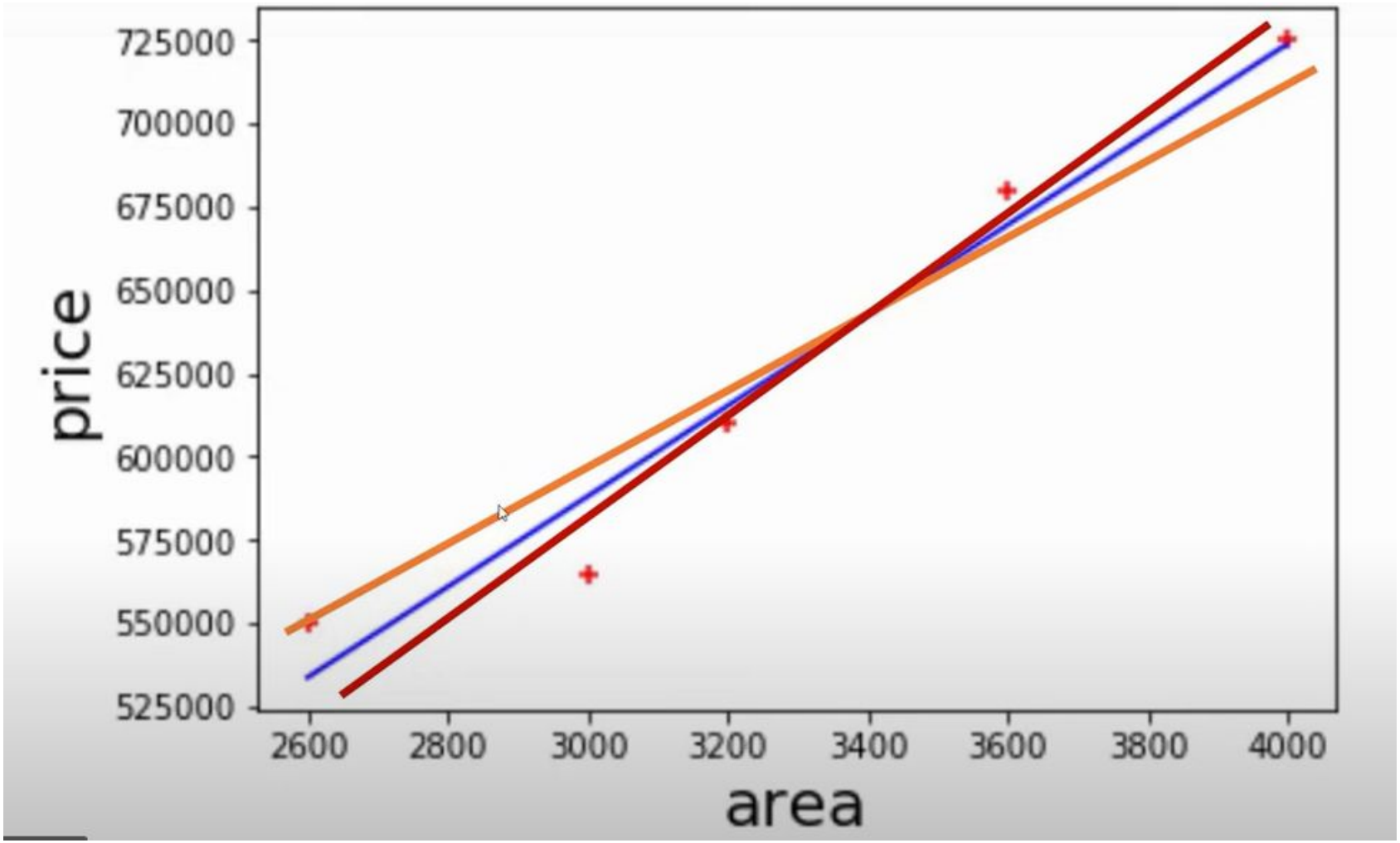
$$mse = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_{predicted}\right)^2$$

Cost Function

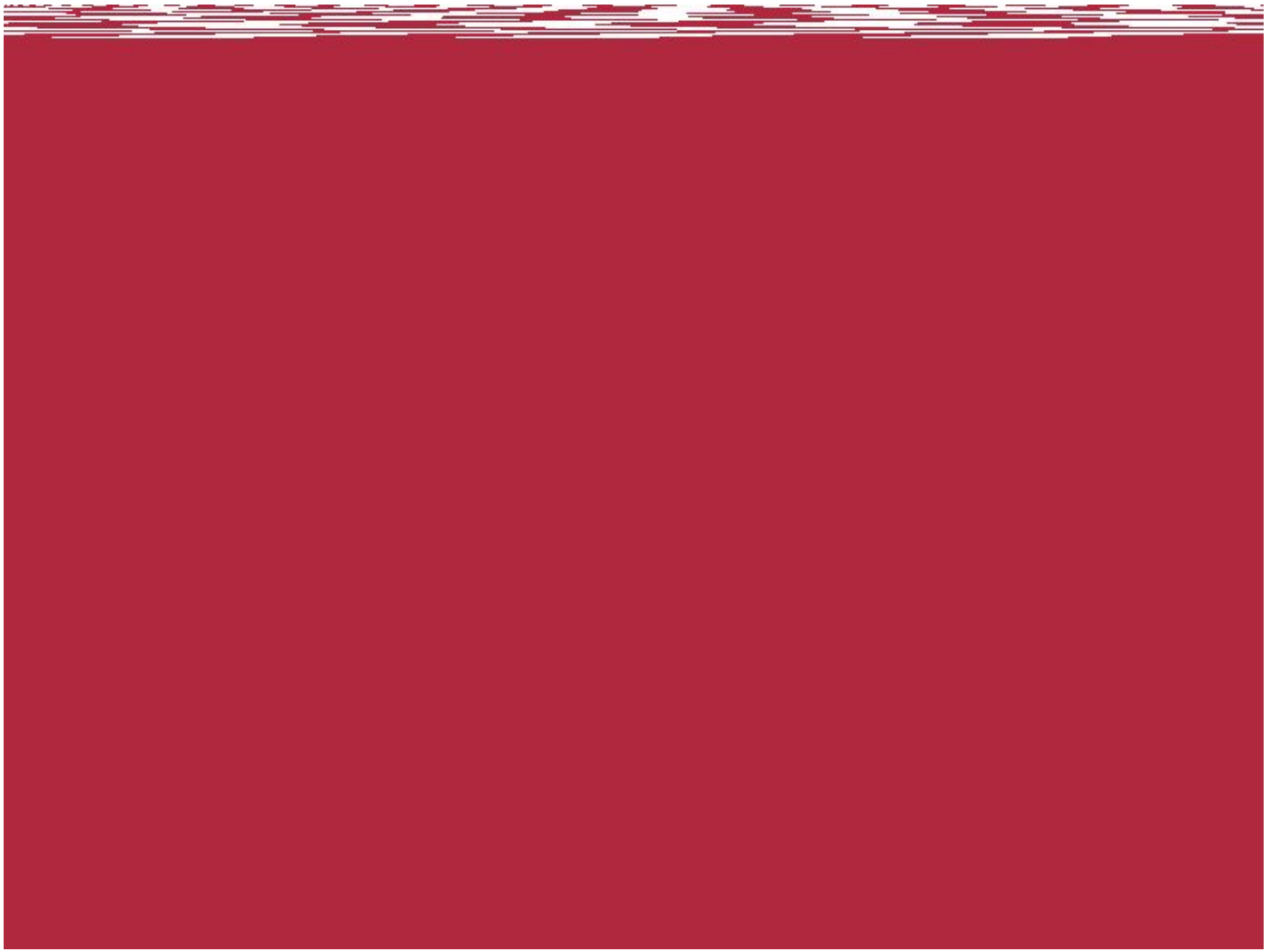
Example


725000
700000
675000
650000
628715
625000
600000
575000
550000
525000
price
2600
2800
3000
3200
3300
3400
3600
3800
4000
area

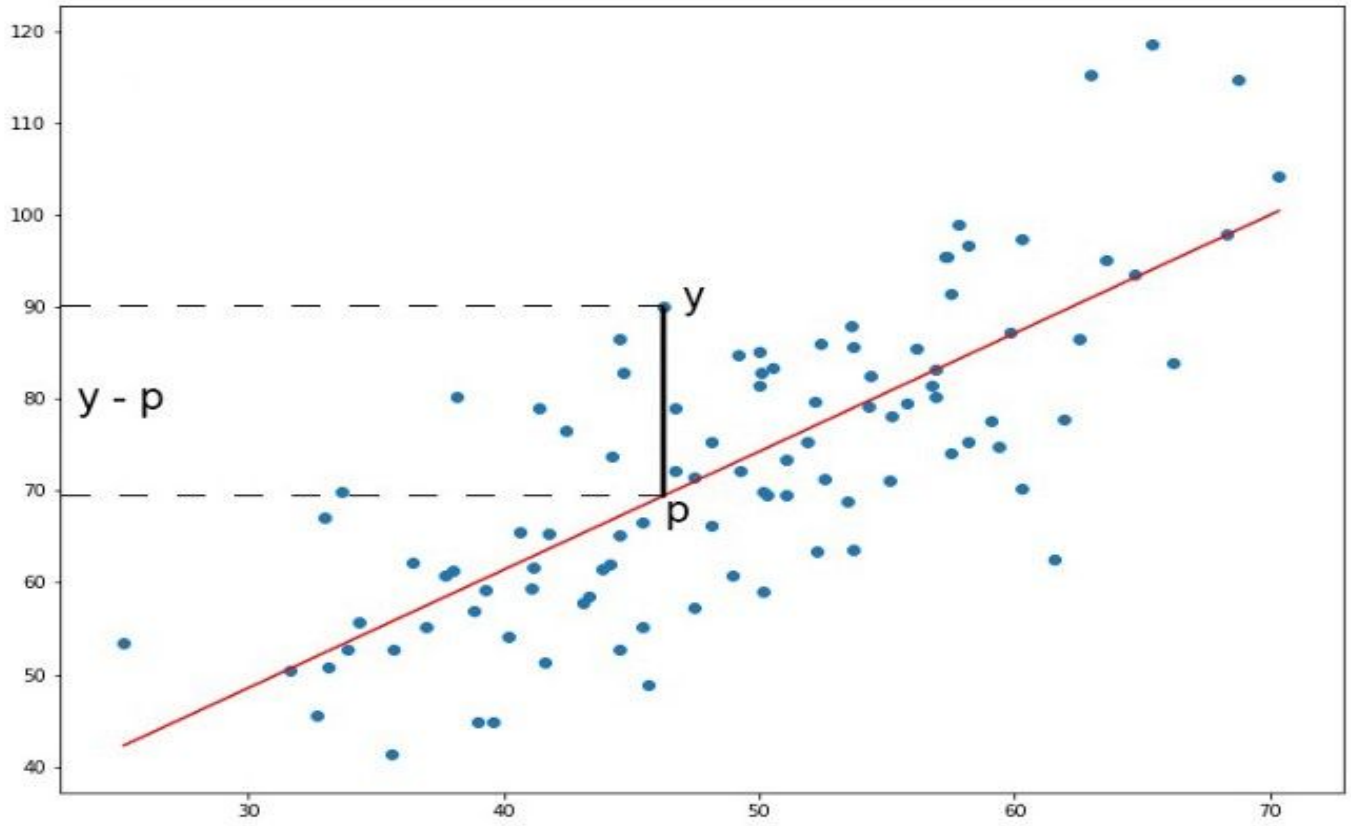
Example


725000
700000
675000
650000
625000
600000
575000
550000
525000
price
2600
2800
3000
3200
3400
3600
3800
4000
area

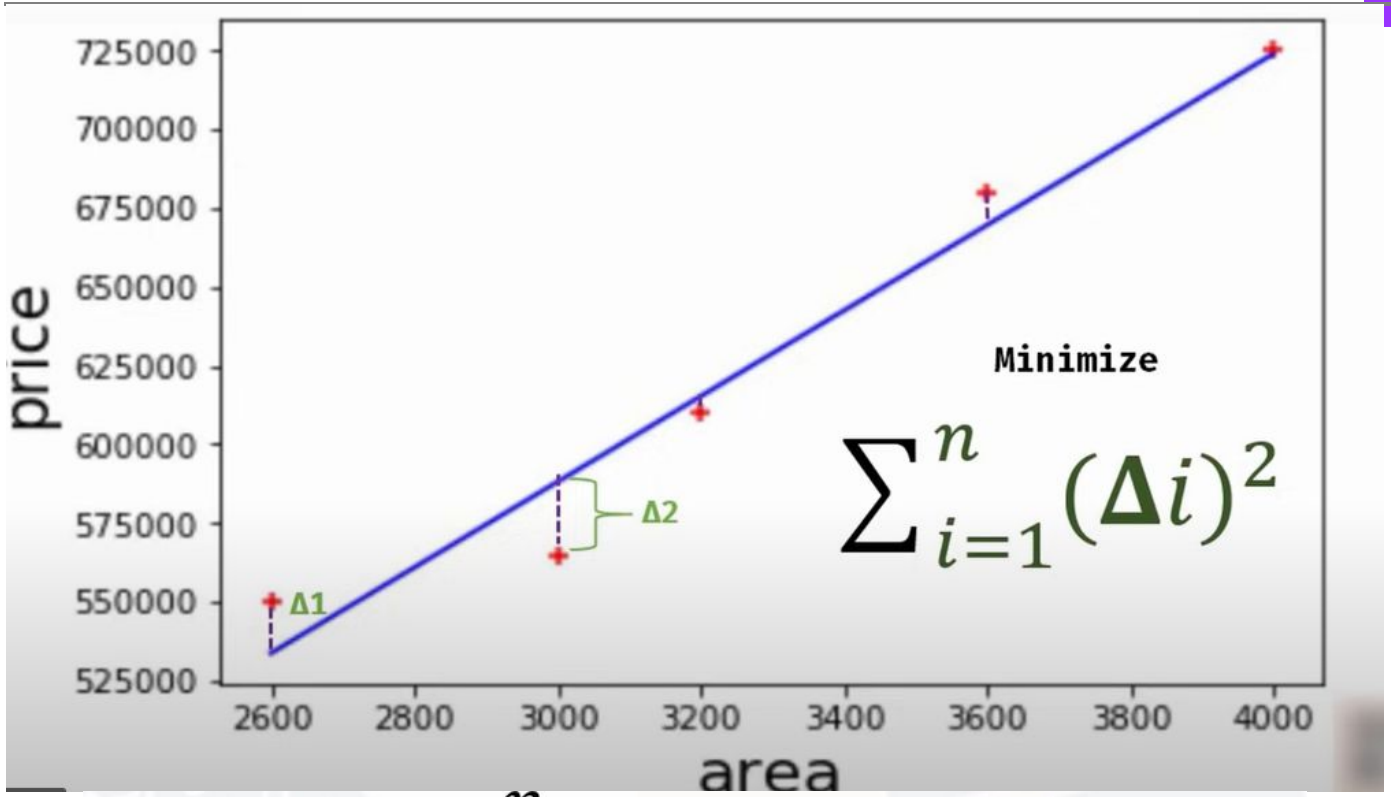
Best Fit Line



Error



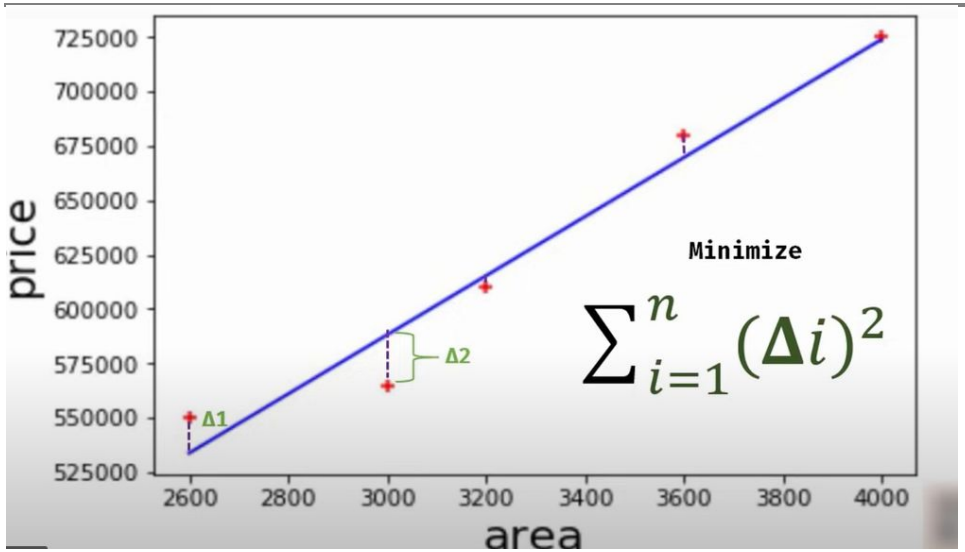
Error



$$mse = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_{predicted}\right)^2$$

Cost Function

Error



$$mse = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_{predicted}\right)^2$$

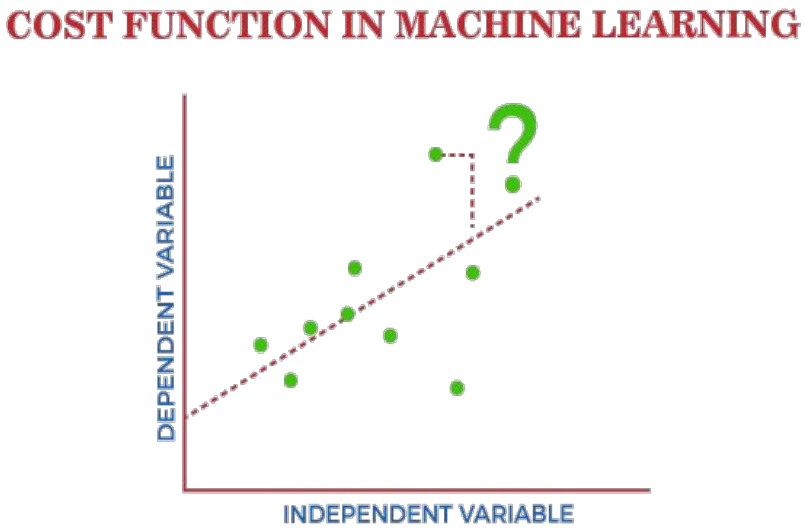
$$mse = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (mx_i + b)\right)^2$$

Cost Function

Cost Function

- A Machine Learning model should have a very high level of accuracy in order to perform well with real-world applications..

- A cost function is an important parameter that determines how well a machine learning model performs for a given dataset. It calculates the difference between the expected value and predicted value and represents it as a single real number.

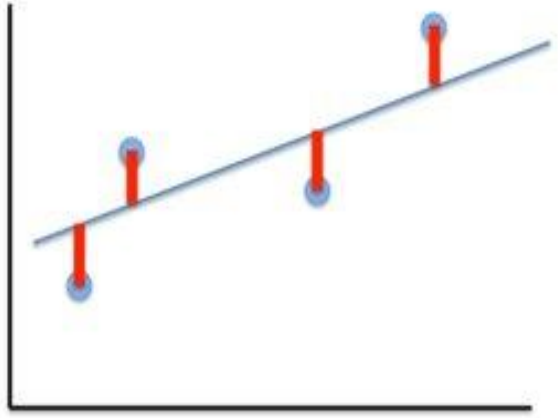


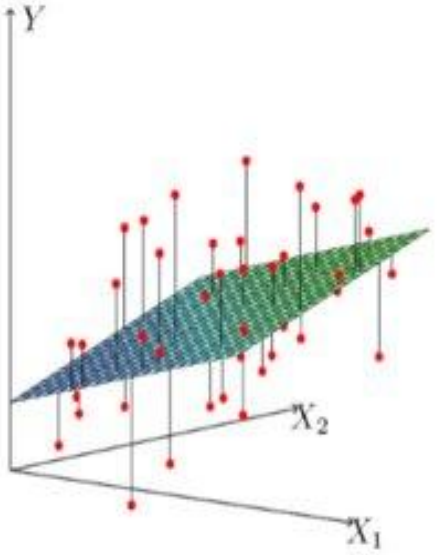
Cost Function

- The goal of the cost function is to help us measuring the accuracy of the prediction.

$$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2$$

Fit by solving $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$





Idea: Choose $\boldsymbol{\Theta}_0$ and $\boldsymbol{\Theta}_1$, so that $h\boldsymbol{\Theta}(x)$ is close to y for our training example (x,y).

$$mse = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_{predicted}\right)^2$$

Cost Function

- Used to find θ_0 and θ_1 which fits the best possible straight line to training data
- Idea: Choose θ_0 and θ_1 so that $h_\theta(x)$ is close to y for training example (x,y)

$$\text{minimize}(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta\left((x^{(i)}\right) - y^{(i)})^2$$

$$\text{Cost function } J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta\left((x^{(i)}\right) - y^{(i)})^2$$

Cost Function: Intuition

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

Parameters: θ_0, θ_1

Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Cost Function: Intuition

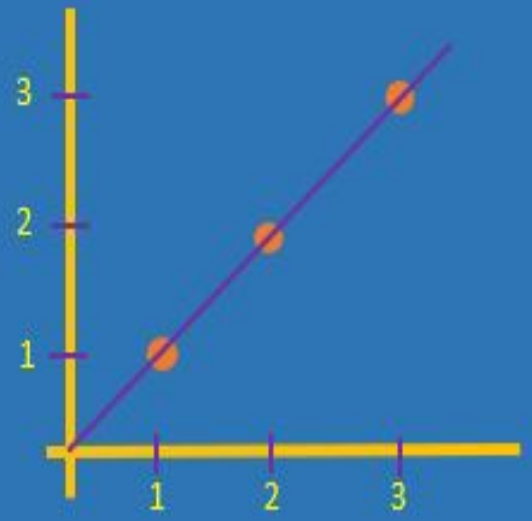
$h_\theta(x) = \theta_1 x$ $(\theta_0=0)$

(for fixed θ_1, this is a function of x)

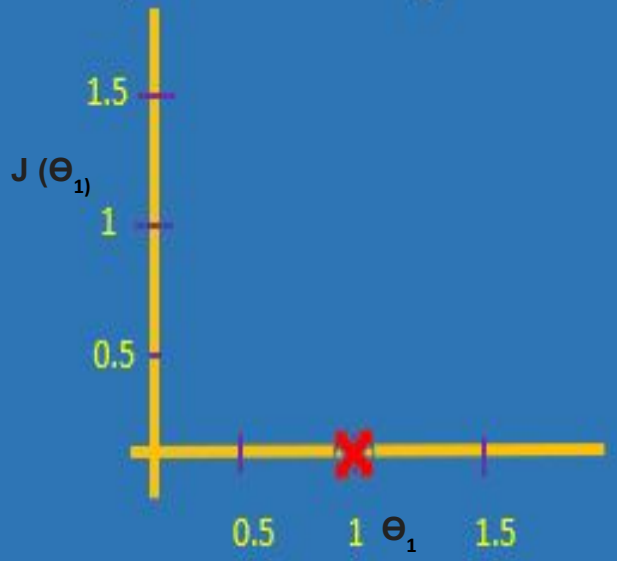
$h_\theta(x) = \theta_1 x + \theta_0$

$J(\theta_1)$

(function of the parameter θ_1)



X	Y
1	1
2	2
3	3



Example 1: $\theta_1 = 1$ and $\theta_0=0$

$h_\theta(1) = \theta_1 x = 1x1 = 1$

$h_\theta(2) = \theta_1 x = 1x2 = 2$

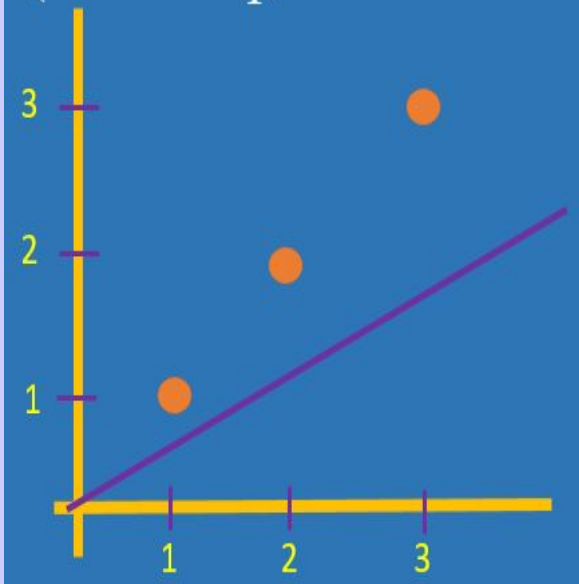
$$J(1) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta\left((x^{(i)}\right) - y^{(i)})^2$$

$$= \frac{1}{6}[(1-1)^2 + (2-2)^2 + (3-3)^2] = 0$$

Cost Function: Intuition

$h_\theta(x) = \theta_1 x \quad (\theta_0=0)$

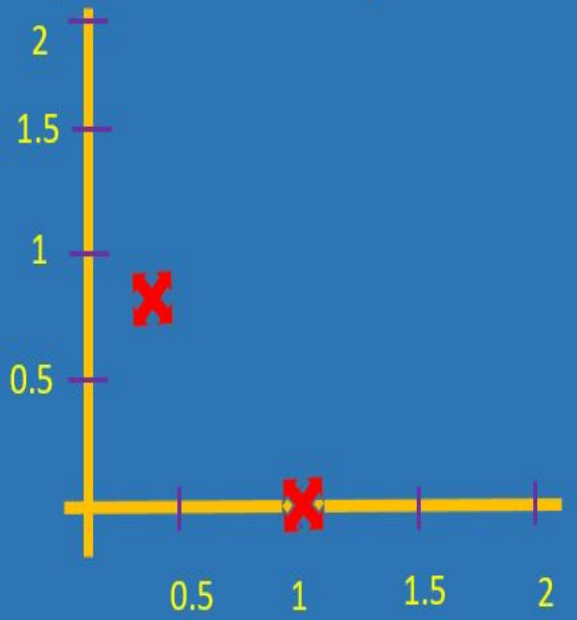
(for fixed θ_1, this is a function of x)



X	Y
1	1
2	2
3	3

$J(\theta_1)$

(function of the parameter θ_1)



Example 1: $\theta_1 = 0.5$ and $\theta_0=0$

$h_\theta(1) = \theta_1 x = 0.5x1 = 0.5$

$h_\theta(2) = \theta_1 x = 0.5x2 = 1.0$

$h_\theta(3) = \theta_1 x = 0.5x3 = 1.5$

$$J(1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta\left(x^{(i)}\right) - y^{(i)})^2$$

$$= \frac{1}{6} [(0.5-1)^2 + (1-2)^2 + (1.5-3)^2]$$

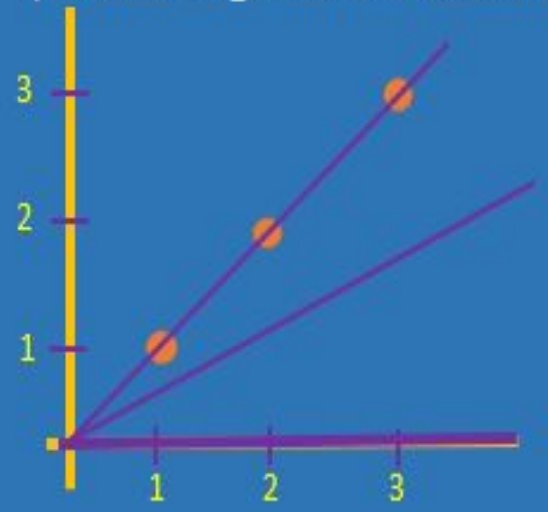
$$= 0.58$$

Cost Function: Intuition

$h_\theta(x) = \theta_1 x + \theta_0$

$h_\theta(x) = \theta_1 x \quad (\theta_0 = 0)$

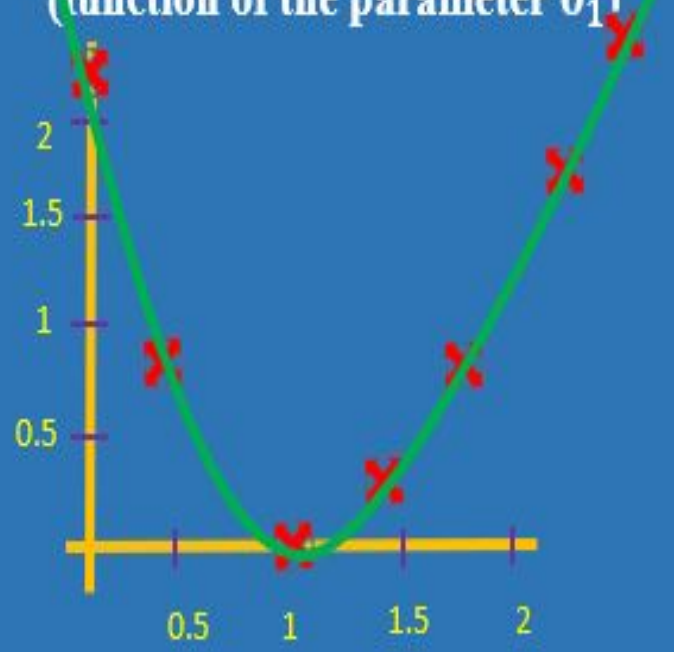
(for fixed θ_1, this is a function of x)



X	Y
1	1
2	2
3	3

$J(\theta_1)$

(function of the parameter θ_1)



Example 1: $\theta_1 = 0$ and $\theta_0 = 0$

$h_\theta(1) = \theta_1 x = 0x1 = 0$

$h_\theta(2) = \theta_1 x = 0x2 = 0$

$h_\theta(3) = \theta_1 x = 0x3 = 0$

$$J(1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

$$= \frac{1}{6} [(0-1)^2 + (0-2)^2 + (0-3)^2]$$

$$= 2.3$$

Cost Function: Intuition

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

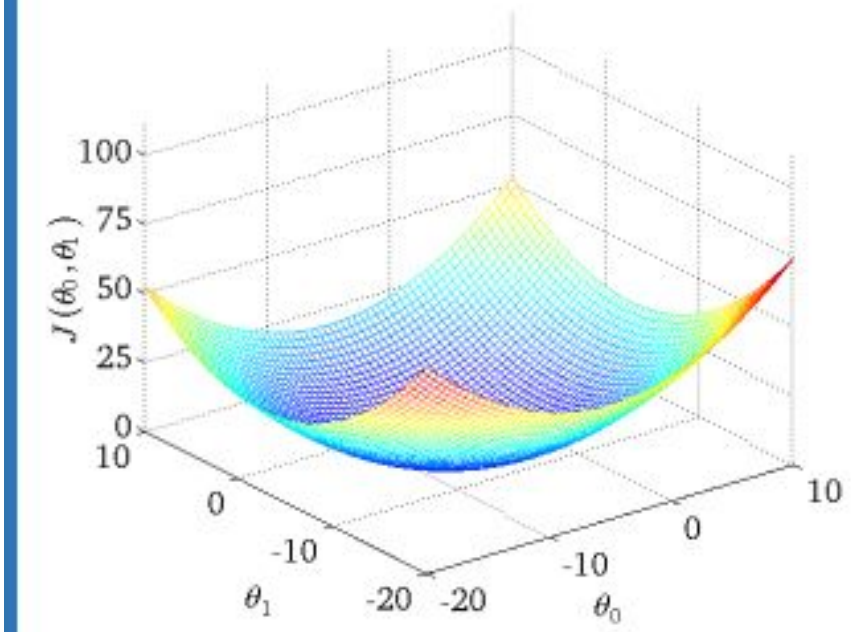
Parameters: θ_0, θ_1

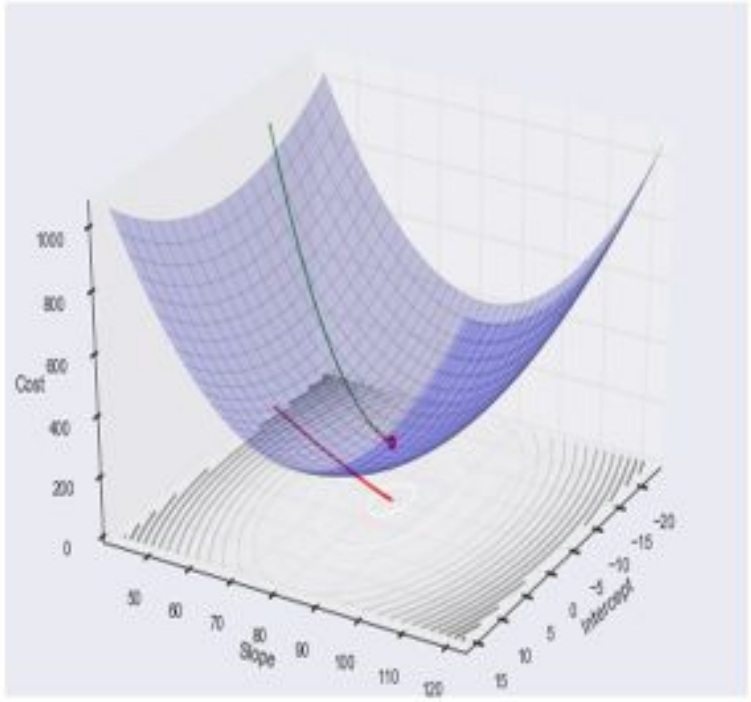
Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Cost Function

Cost Function: Intuition





Based on example
by Andrew Ng

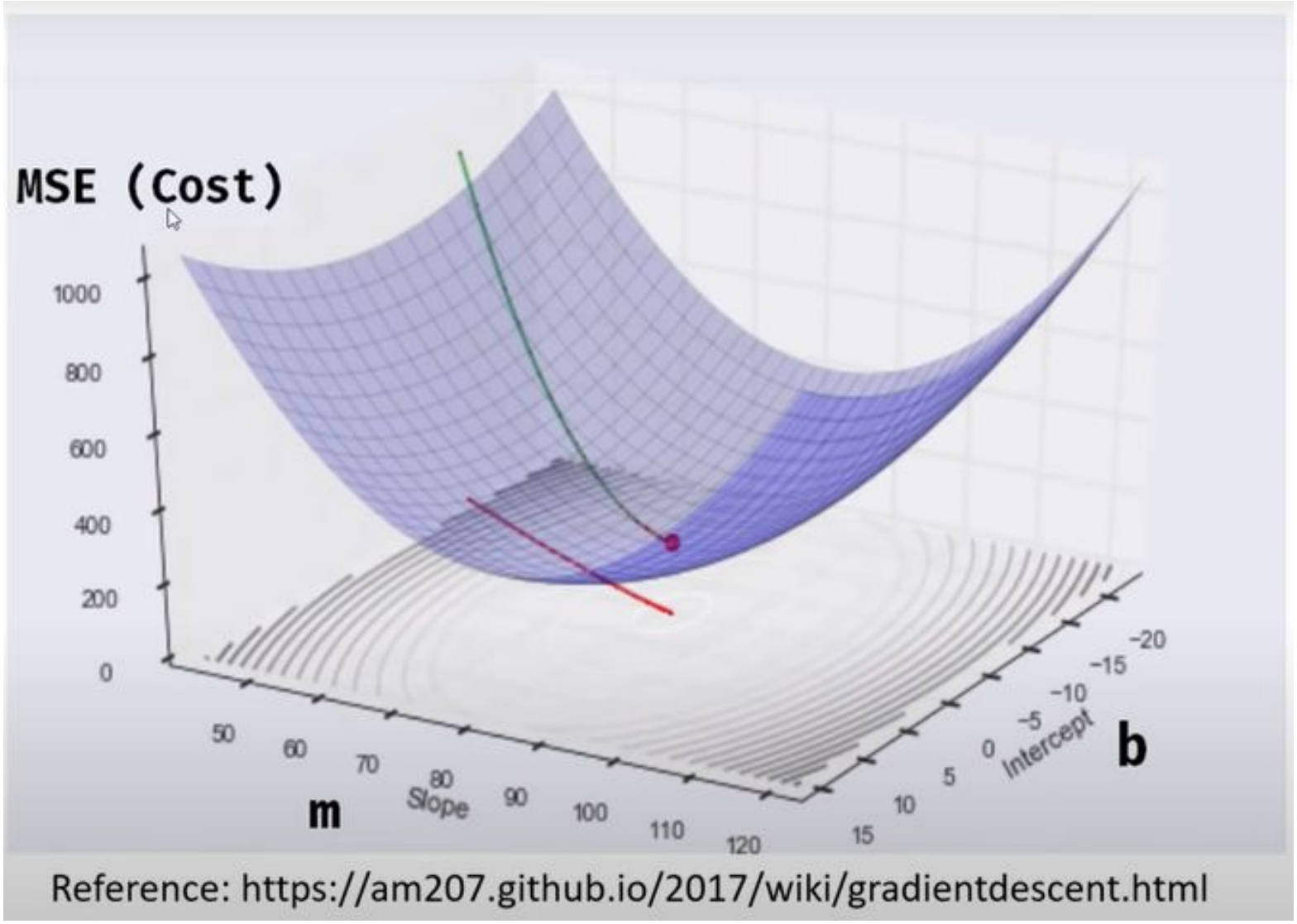
What is Gradient Descent?

"A gradient measures how much the output of a function changes if you change the inputs a little bit."
— Lex Fridman (MIT)

Gradient descent is simply used to find the values of a function's parameters (coefficients) that minimize a cost function as far as possible.

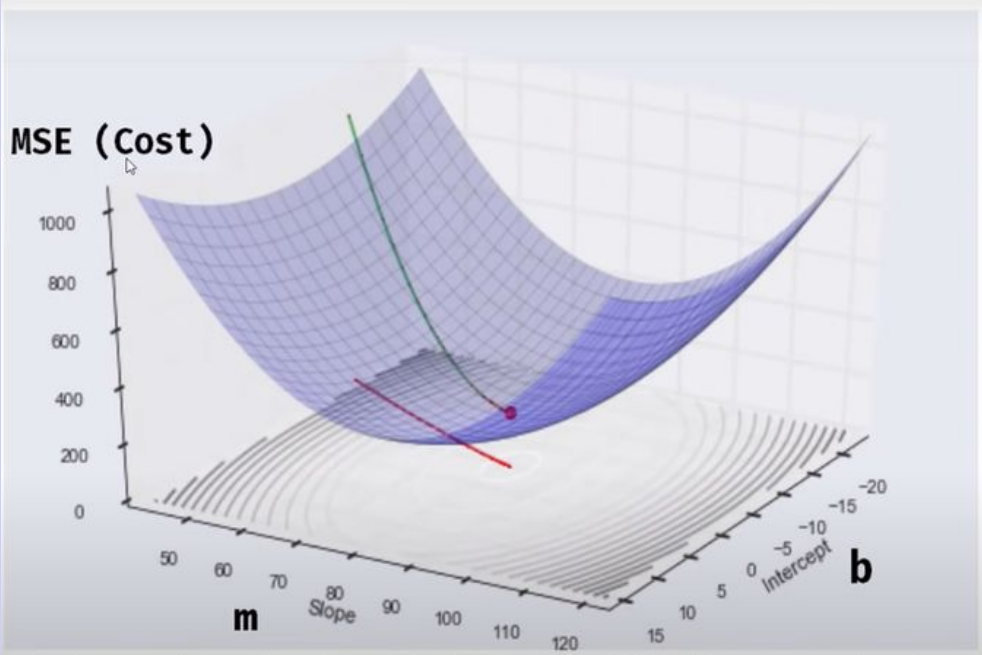
Gradient Descent

Gradient Descent

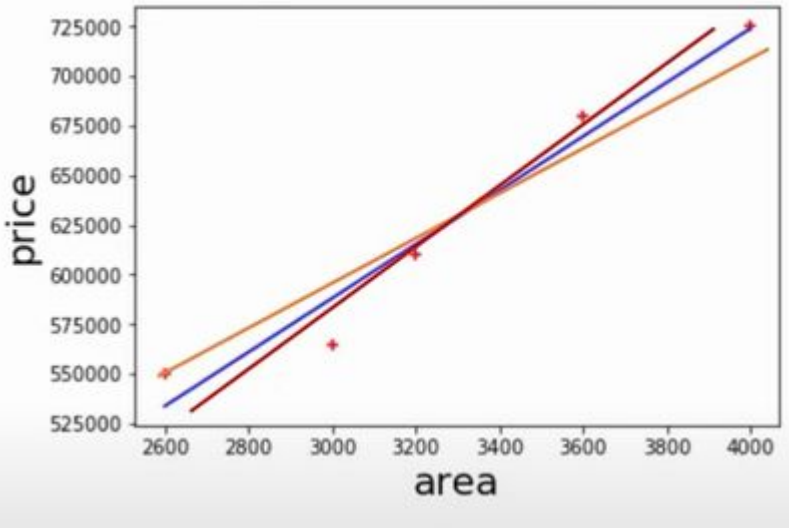


Reference: https://am207.github.io/2017/wiki/gradientdescent.html

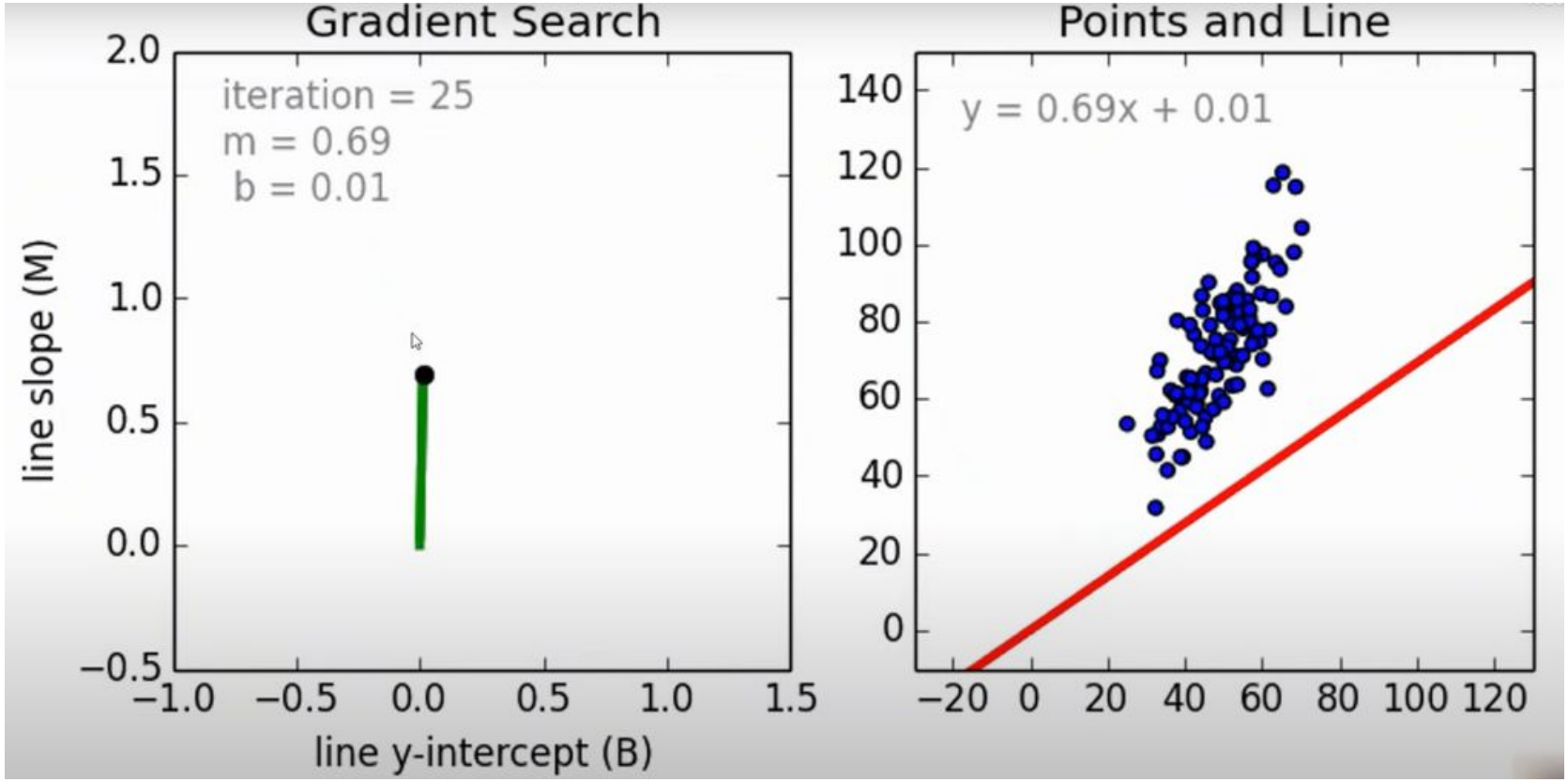
Gradient Descent



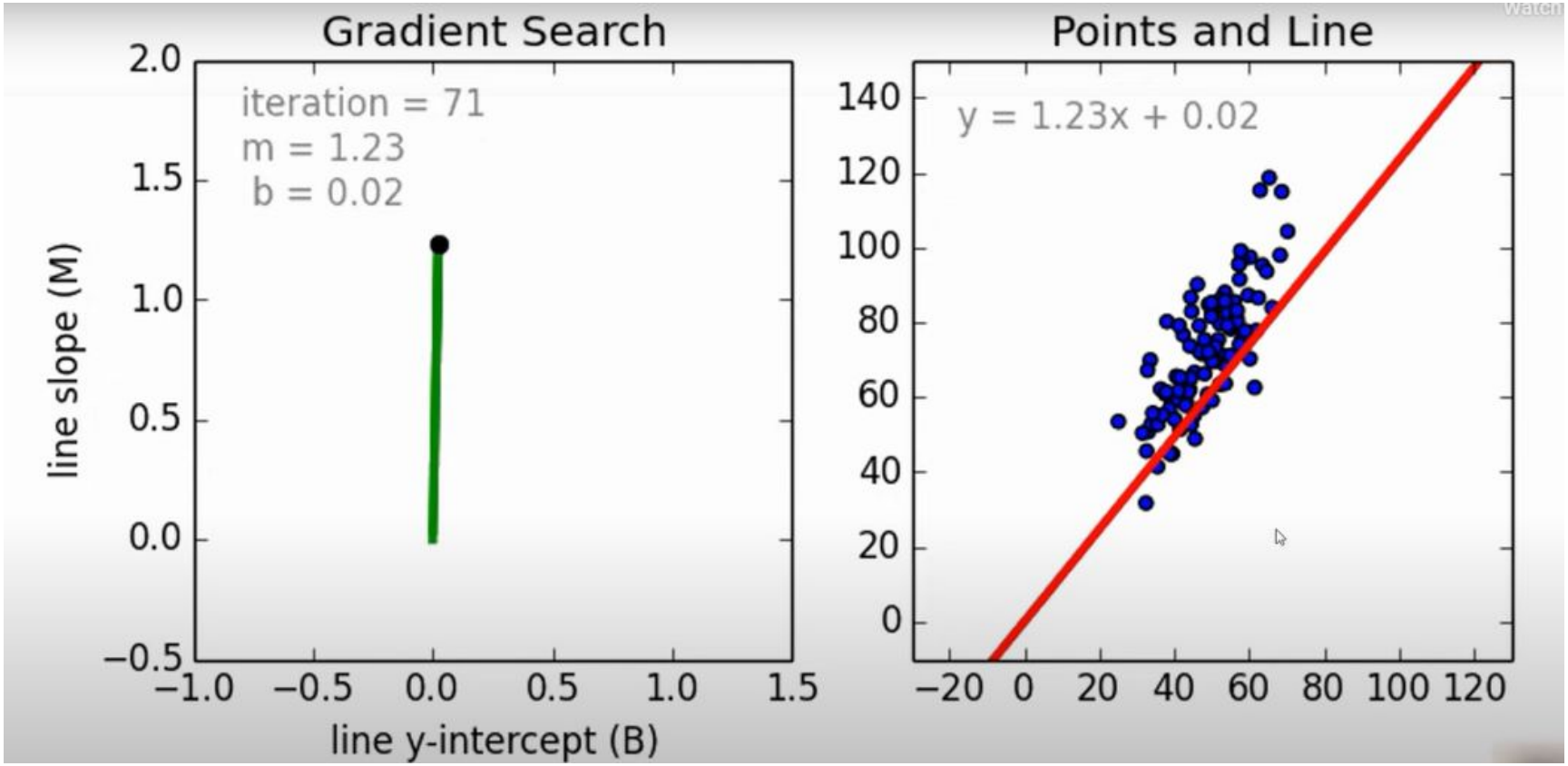
Reference: https://am207.github.io/2017/wiki/gradientdescent.html



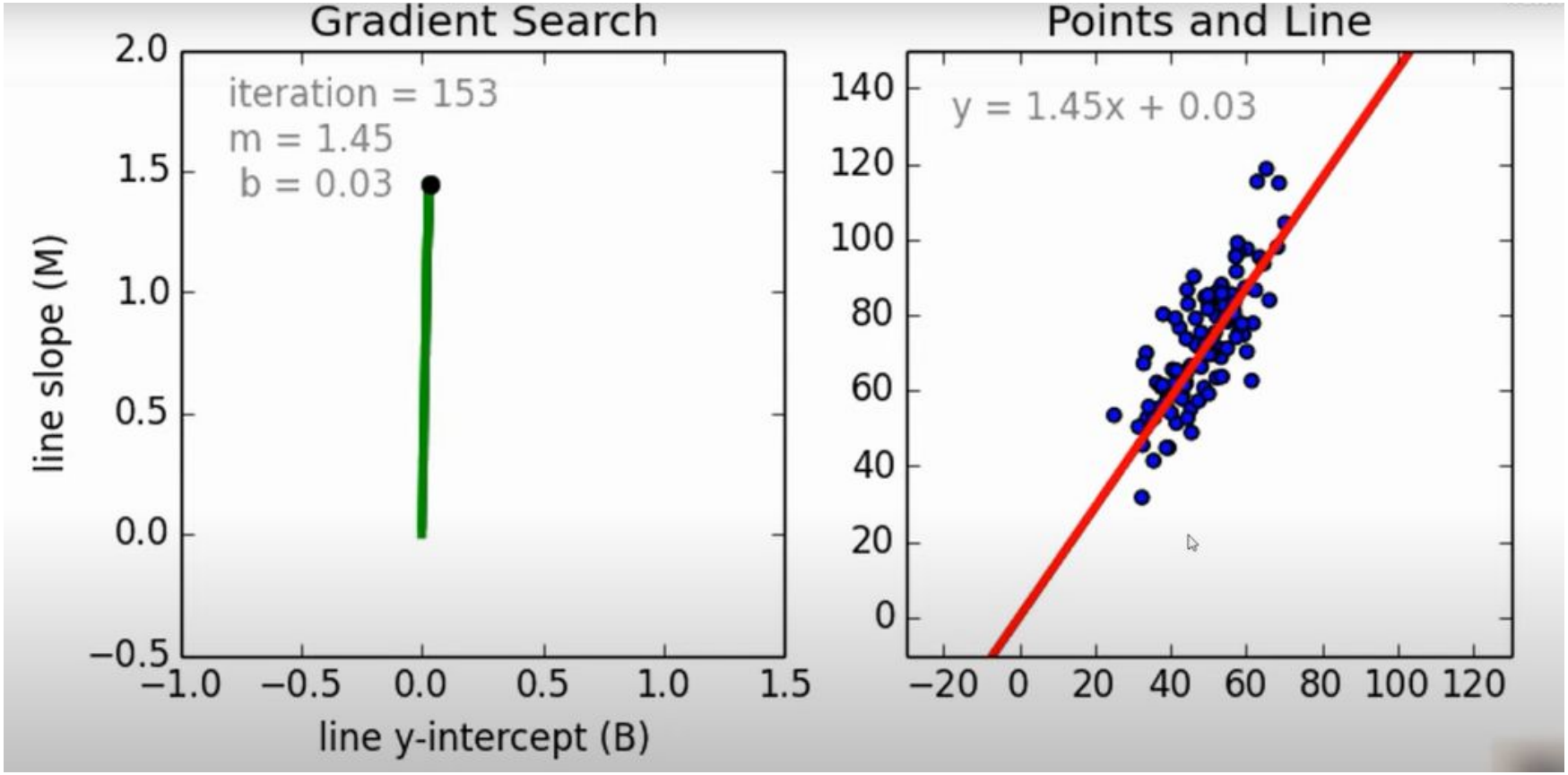
Gradient Descent



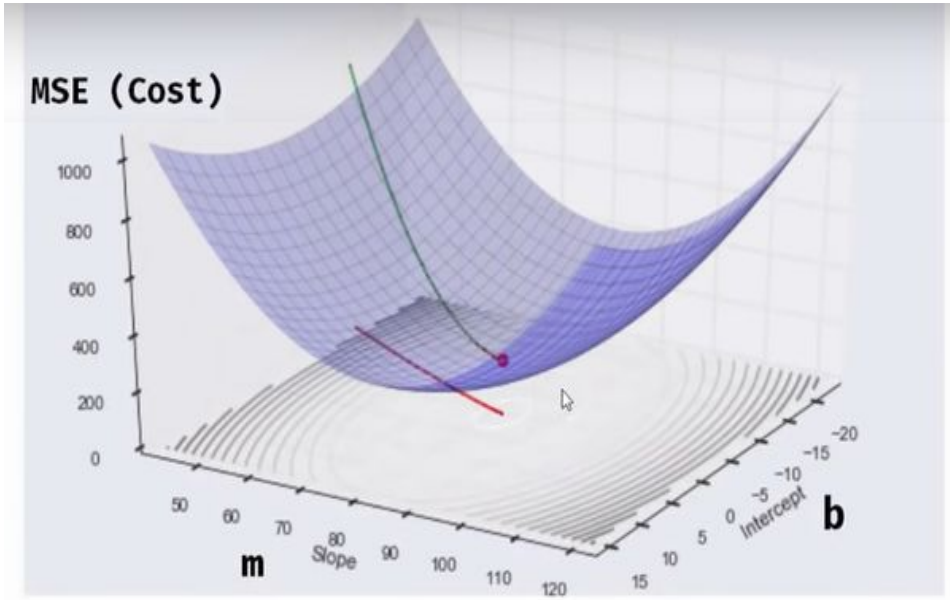
Gradient Descent

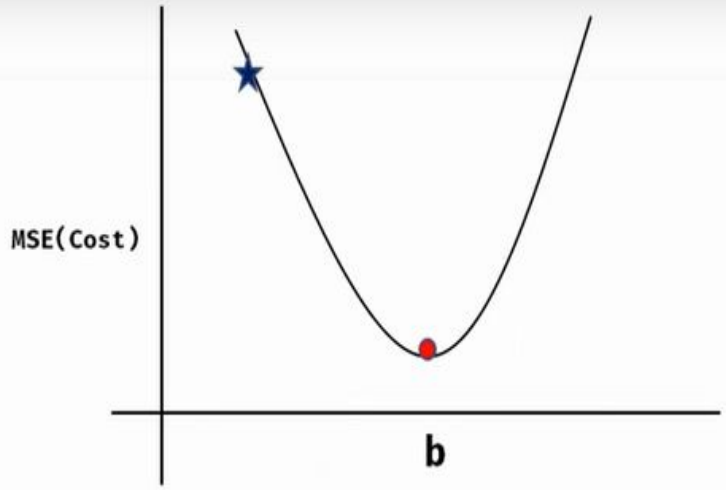


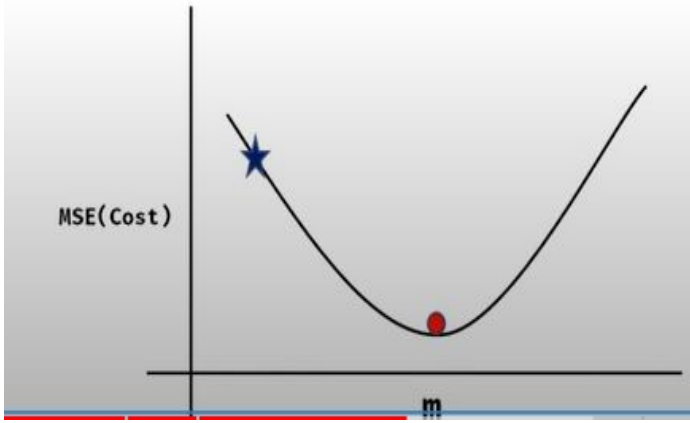
Gradient Descent



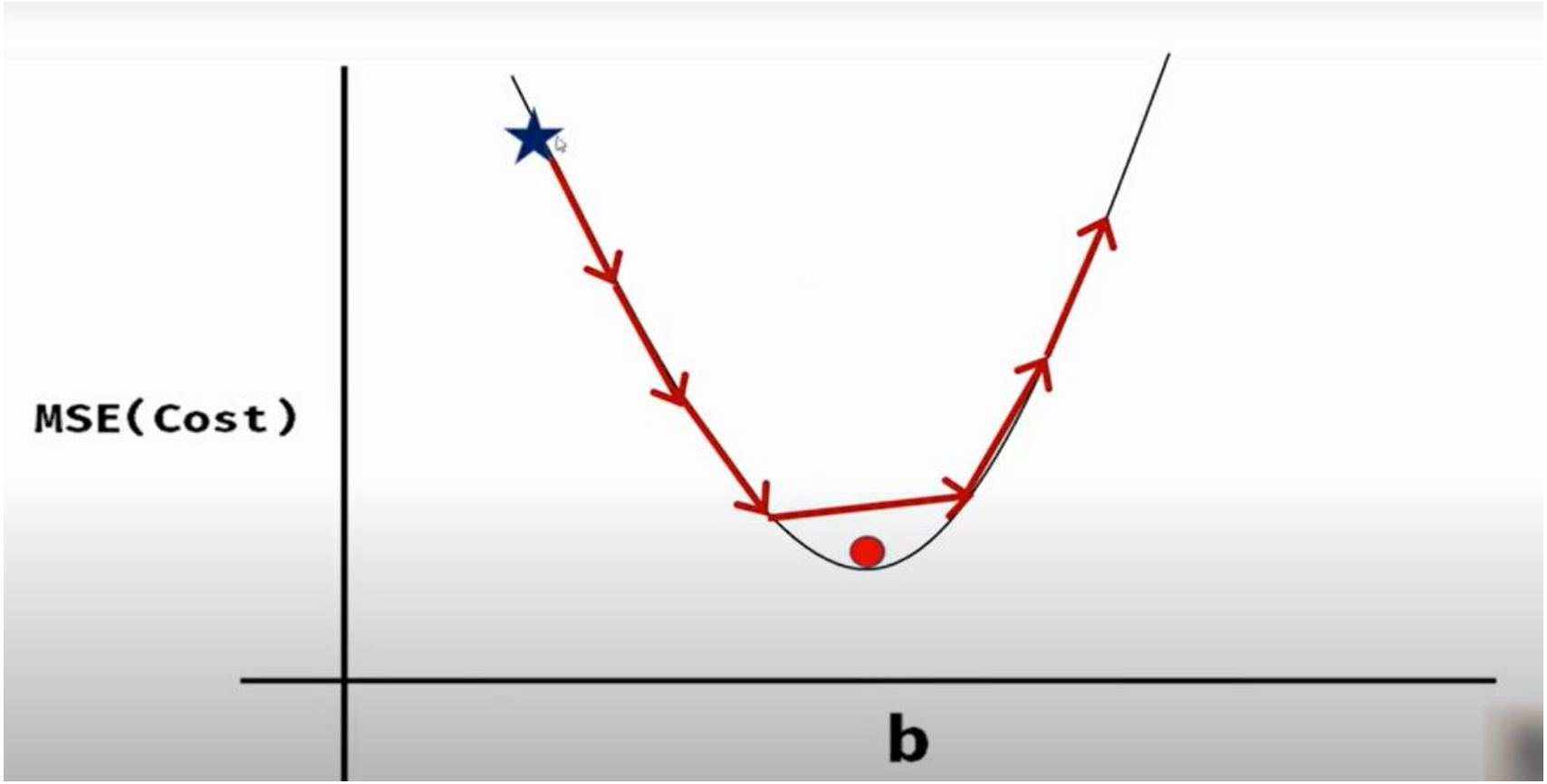
Gradient Descent



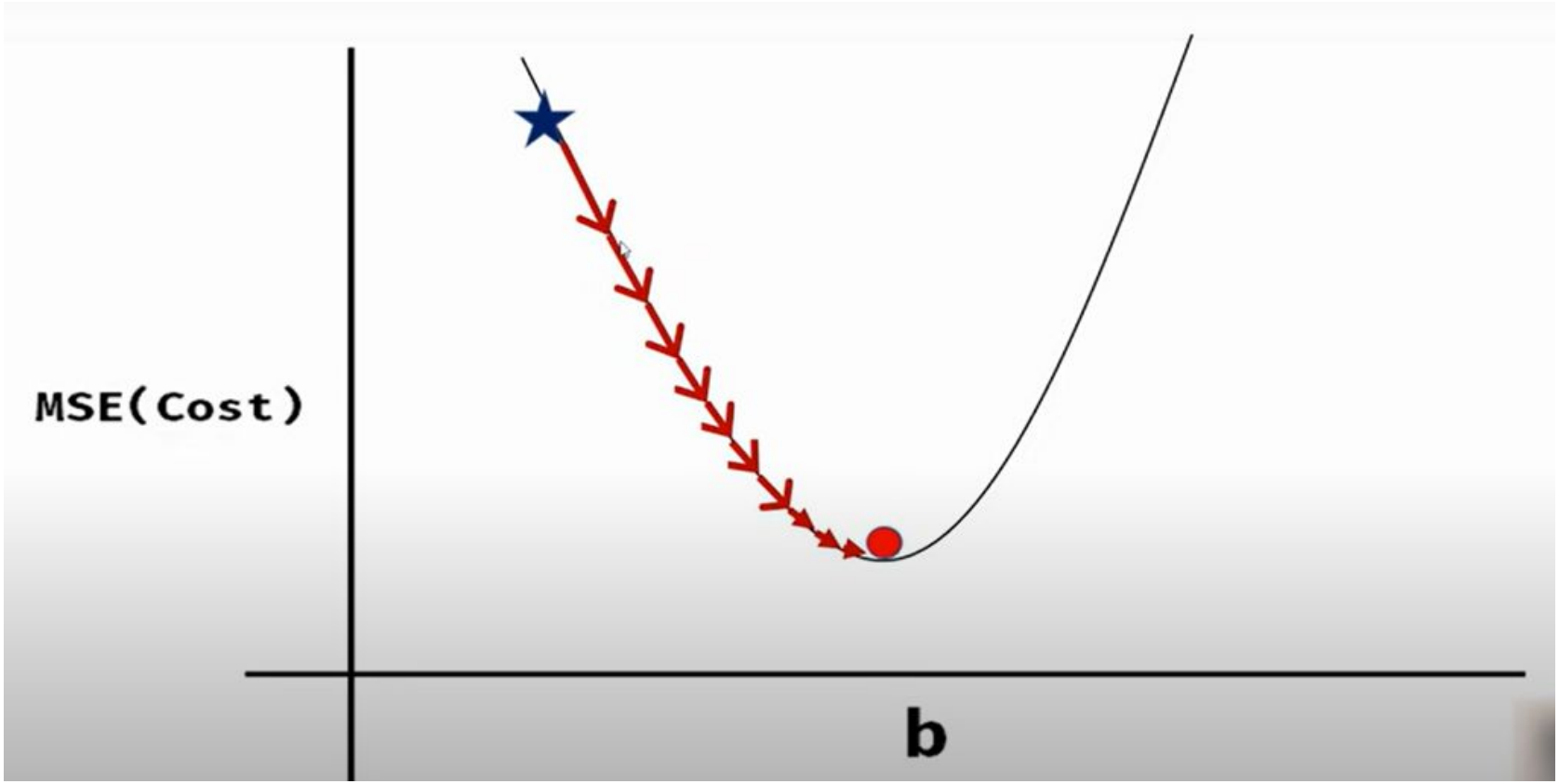




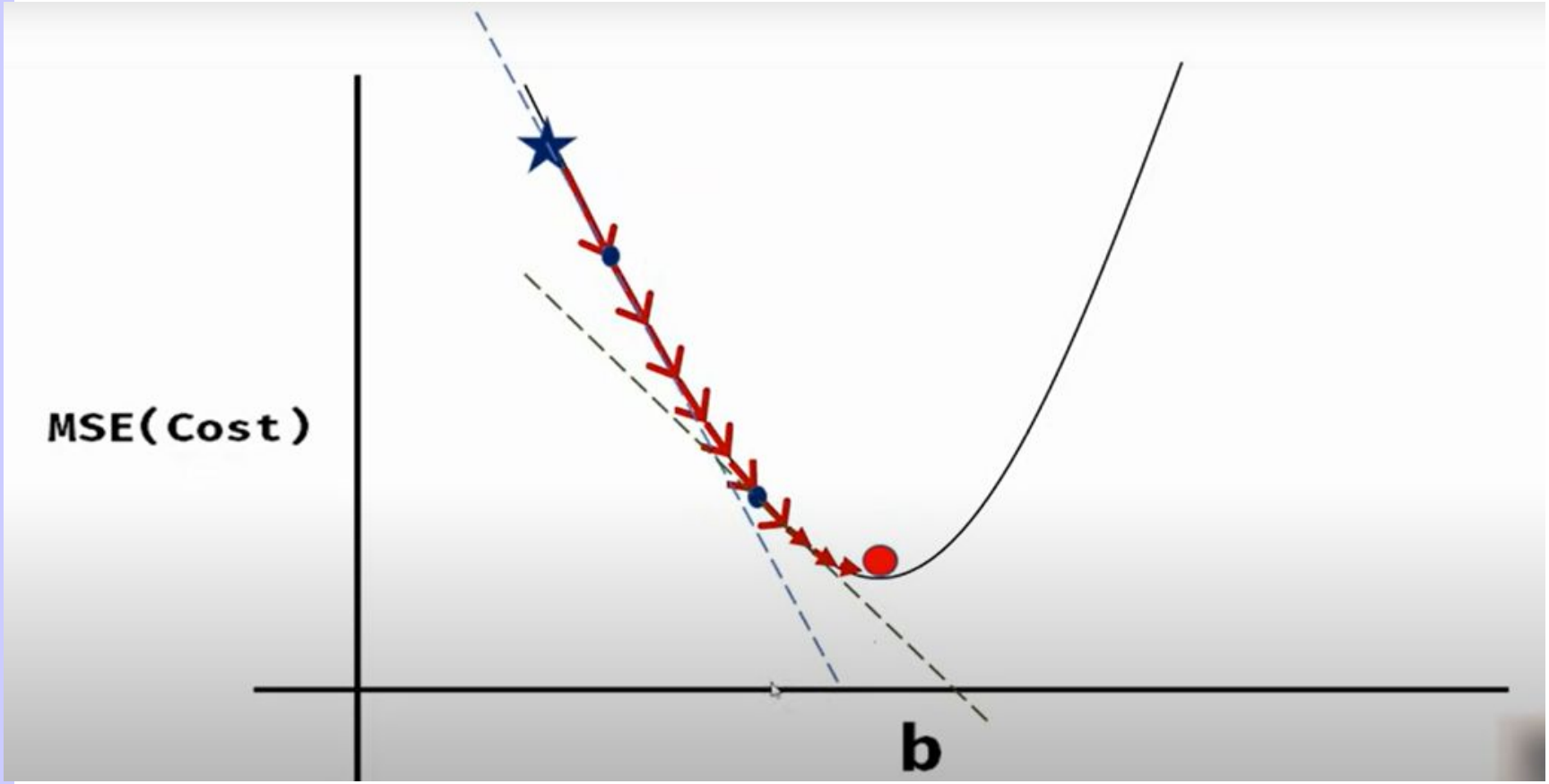
Gradient Descent



Gradient Descent



Gradient Descent



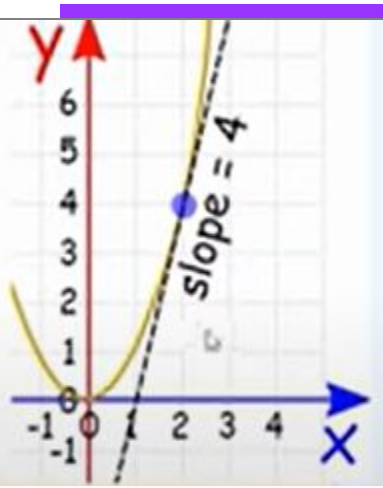
Gradient Descent

What does $\frac{d}{dx}x^2 = 2x$ mean?

It means that, for the function x^2, the slope or "rate of change" at any point is **2x**.

So when **x=2** the slope is **2x = 4**, as shown here:

Or when **x=5** the slope is **2x = 10**, and so on.



$$f(x) = x^3$$

$$\frac{\mathrm{d}}{\mathrm{d}x}x^3 = 3x^2$$

$$f(x,y) = x^3 + y^2$$

$$\partial f/\partial x = 3x^2 + 0 = 3x^2$$

$$\partial f/\partial y = 0 + 2y = 2y$$

Gradient Descent

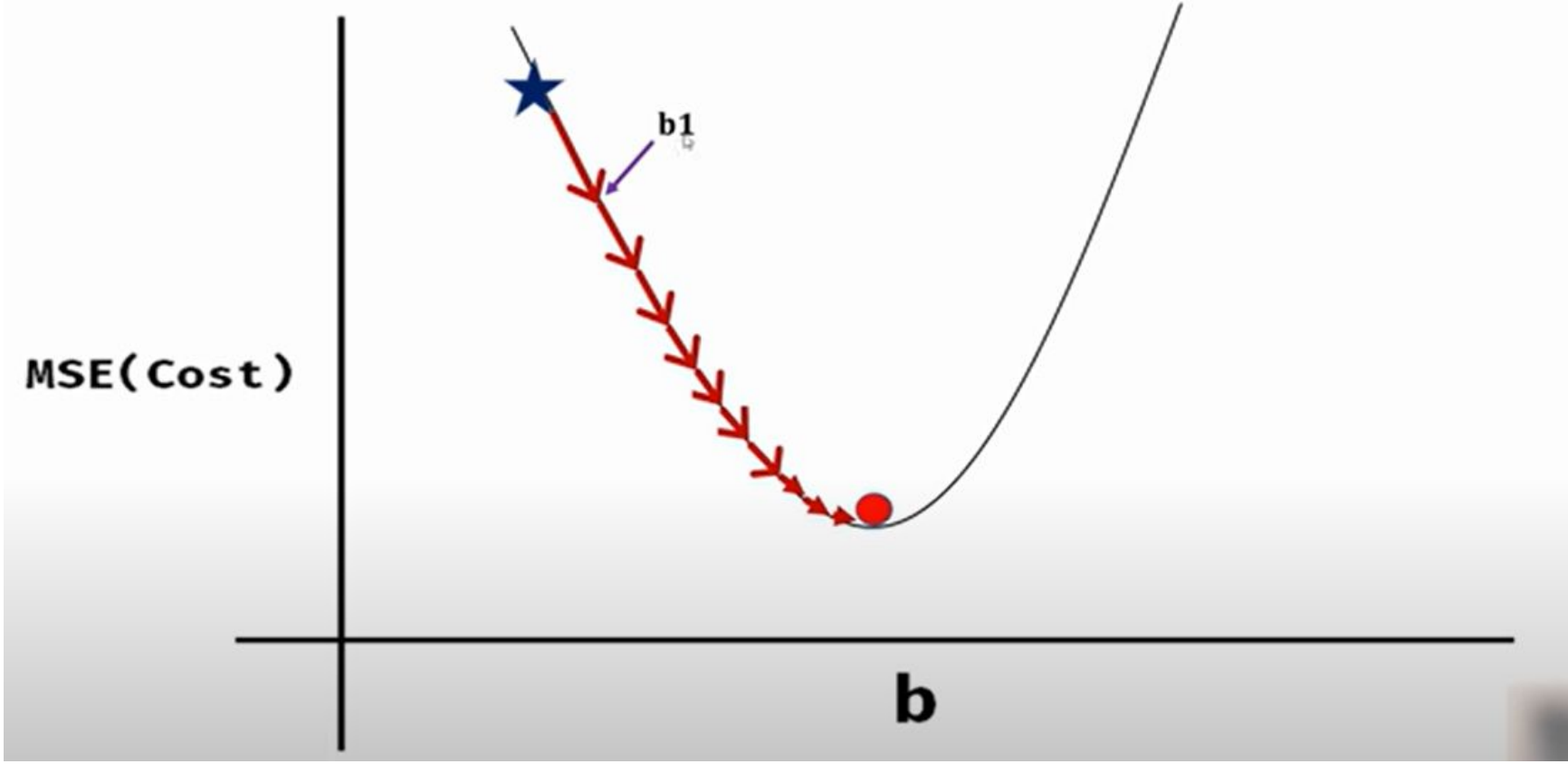
$$mse = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (mx_i + b)\right)^2$$

$$\partial/\partial m = \frac{2}{n}\sum_{i=1}^{n} -x_i\left(y_i - (mx_i + b)\right)$$

$$\partial/\partial b = \frac{2}{n}\sum_{i=1}^{n} -\left(y_i - (mx_i + b)\right)$$

Learning Rate

This parameter **determines how fast or slow we will move towards the optimal weights**.

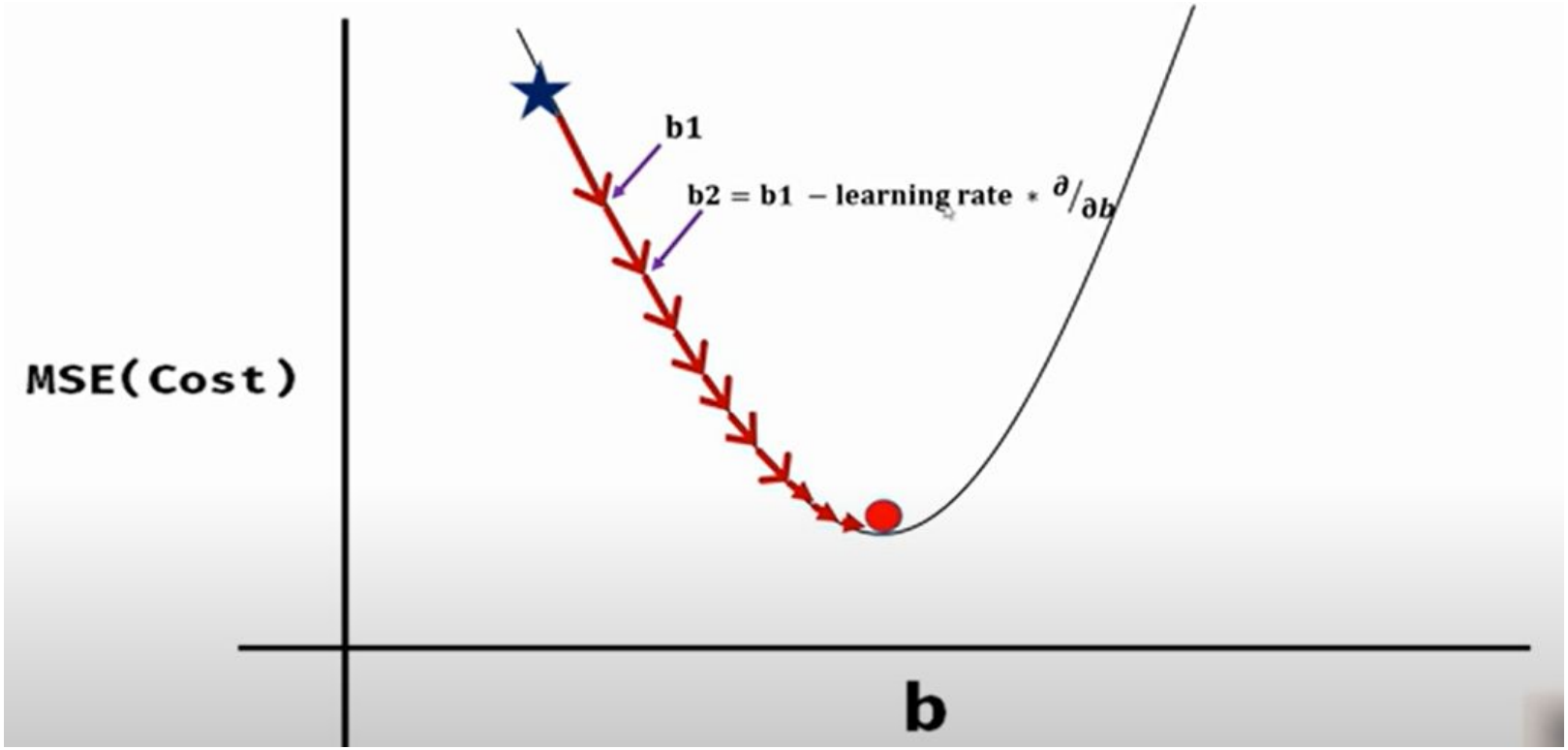


Learning Rate

$$m = m - \text{learning rate} * {}^{\partial}/_{\partial m}$$

$$b = b - \text{learning rate} * {}^{\partial}/_{\partial b}$$

Learning Rate



Algorithm

1. Initially let m = 0 and c = 0. Let L be our learning rate. This controls how much the value of **m** changes with each step. L could be a small value like 0.0001 for good accuracy.

Algorithm

2. Calculate the partial derivative of the loss function with respect to m, and plug in the current values of x, y, m and c in it to obtain the derivative value **D**.

$$D_m = \frac{\partial(Cost\ Function)}{\partial m} = \frac{\partial}{\partial m}\left(\frac{1}{n}\sum_{i=0}^{n}(y_i - y_{i\,pred})^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial m}\left(\sum_{i=0}^{n}(y_i - (mx_i + c))^2\right)$$

$$= \frac{-2}{n}\sum_{i=0}^{n} x_i(y_i - (mx_i + c))$$

$$= \frac{-2}{n}\sum_{i=0}^{n} x_i(y_i - y_{i\,pred})$$

D□ is the value of the partial derivative with respect to **m**.

Algorithm

2. Calculate the partial derivative of the loss function with respect to m, and plug in the current values of x, y, m and c in it to obtain the derivative value **D**.

$$D_m = \frac{\partial(Cost\ Function)}{\partial m} = \frac{\partial}{\partial m}\left(\frac{1}{n}\sum_{i=0}^{n}(y_i - y_{i\,pred})^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial m}\left(\sum_{i=0}^{n}(y_i - (mx_i + c))^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial m}\left(\sum_{i=0}^{n}(y_i^2 + m^2x_i^2 + c^2 + 2mx_ic - 2y_imx_i - 2y_ic)\right)$$

$$= \frac{-2}{n}\sum_{i=0}^{n}x_i(y_i - (mx_i + c))$$

$$= \frac{-2}{n}\sum_{i=0}^{n}x_i(y_i - y_{i\,pred})$$

D□ is the value of the partial derivative with respect to **m**.

Algorithm

3. Calculate the partial derivative of the loss function with respect to c.

$$= \frac{1}{n}\frac{\partial}{\partial c}\left(\sum_{i=0}^{n}(y_i - (mx_i + c))^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial c}\left(\sum_{i=0}^{n}(y_i - (mx_i + c))^2\right)$$

$$= \frac{-2}{n}\sum_{i=0}^{n}(y_i - (mx_i + c))$$

$$\frac{-2}{n}\sum_{i=0}^{n}(y_i - y_{i\,pred})$$

Dc is the value of the partial derivative with respect to **c**.

Algorithm

3. Calculate the partial derivative of the loss function with respect to c.

$$D_c = \frac{\partial(Cost\ Function)}{\partial c} = \frac{\partial}{\partial c}\left(\frac{1}{n}\sum_{i=0}^{n}(y_i - y_{i\,pred})^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial c}\left(\sum_{i=0}^{n}(y_i - (mx_i + c))^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial c}\left(\sum_{i=0}^{n}(y_i^2 + m^2x_i^2 + c^2 + 2mx_ic - 2y_imx_i - 2y_ic)\right)$$

$$= \frac{-2}{n}\sum_{i=0}^{n}(y_i - (mx_i + c))$$

$$\frac{-2}{n}\sum_{i=0}^{n}(y_i - y_{i\,pred})$$

Dc is the value of the partial derivative with respect to **c**.

Algorithm

4. Now we update the current value of **m** and **c** using the following equation:

$$m = m - L \times D_m$$

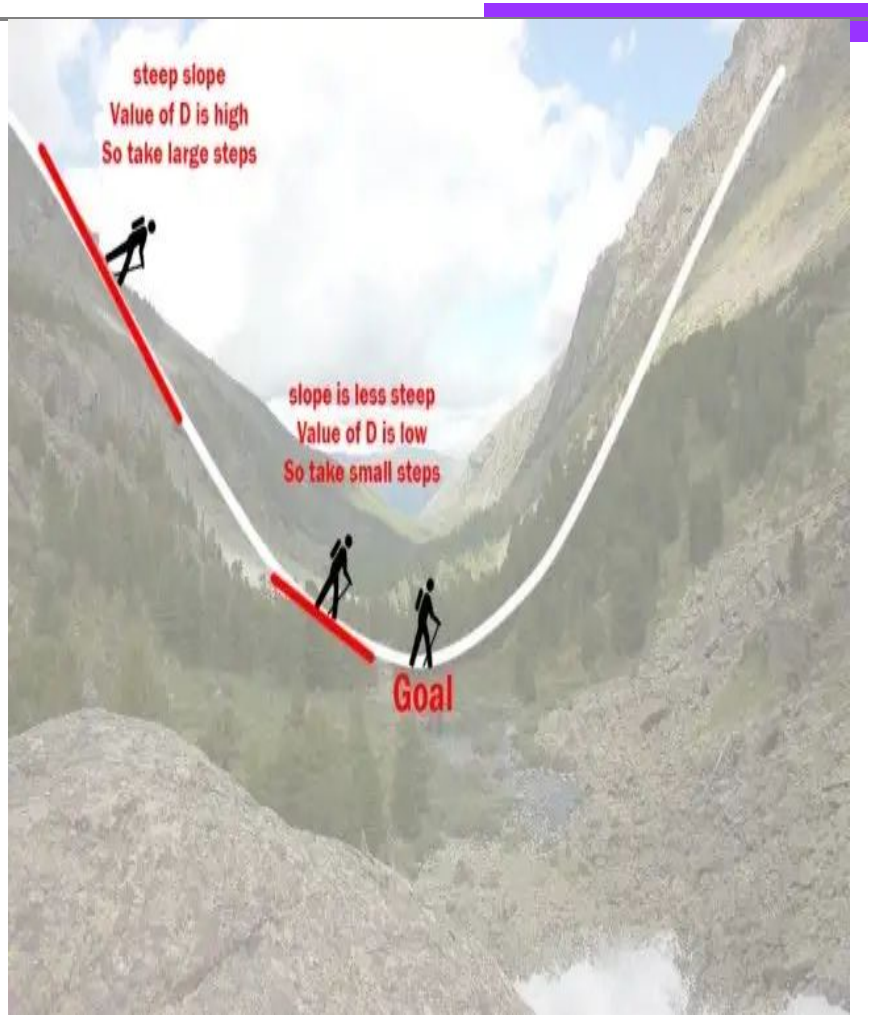
$$c = c - L \times D_c$$

5. We repeat this process until our loss function is a very small value or ideally 0 (which means 0 error or 100% accuracy). The value of **m** and **c** that we are left with now will be the optimum values.

Example

Now going back to our analogy, **m** can be considered the current position of the person. **D** is equivalent to the steepness of the slope and **L** can be the speed with which he moves. Now the new value of **m** that we calculate using the above equation will be his next position, and **L×D** will be the size of the steps he will take. When the slope is more steep (**D** is more) he takes longer steps and when it is less steep (**D** is less), he takes smaller steps. Finally he arrives at the bottom of the valley which corresponds to our loss = 0.

Now with the optimum value of **m** and **c** our model is ready to make predictions !



Finding Error

- So to minimize the error we need a way to calculate the error in the first place.
-
- A **loss function** in machine learning is simply a measure of how different the predicted value is from the actual value.

- **Quadratic Loss Function is used** to calculate the loss or error in the model. It can be defined as:

$$L(x) = \sum_{i=1}^{n} (y_i - p_i)^2$$

Linear Regression

- Straight-line regression analysis involves a response variable, *y*, and a single predictor variable, *x*. It is the simplest form of regression, and models *y* as a linear function of *x*.
- That is, $y = w_0 + w_1 x$:
- These coefficients can be solved for by the method of least squares, which estimates the best-fitting straight line as the one that minimizes the error between the actual data and the estimate of the line.
- Let *D* be a training set consisting of values of predictor variable.
- The regression coefficients can be estimated using this method with the following equations:

$$w_1 = \frac{\sum_{i=1}^{|D|}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|}(x_i - \bar{x})^2}$$

$$w_0 = \bar{y} - w_1 \bar{x}$$

Least Square Method

Now that we have determined the loss function, the only thing left to do is minimize it.

This is done by finding the partial derivative of **L**, equating it to 0 and then finding an expression for **m** and **c**.

After we do the math, we are left with these equations:

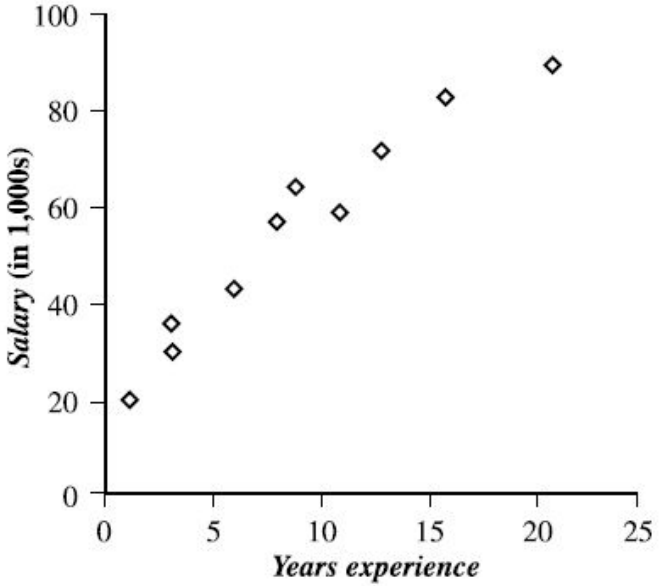
$$m = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$c = \bar{y} - m\bar{x}$$

Example

Salary data.

x years experience	*y* salary (in $1000s)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83



Given the above data, we compute Mean(*x*) = 9.1 and Mean(*y*) = 55.4.

$$w_1 = \frac{(3-9.1)(30-55.4)+(8-9.1)(57-55.4)+\cdots+(16-9.1)(83-55.4)}{(3-9.1)^2+(8-9.1)^2+\cdots+(16-9.1)^2} = 3.5$$

$$w_0 = 55.4-(3.5)(9.1) = 23.6$$

Thus, the equation of the least squares line is estimated by $y = 23.6+3.5x$.

Example

X: Independent variable

Y: Dependent variable

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

Continuous Values

Parameters of Linear Regression : Example

$$\hat{y} = \theta_0 + \theta_1 x_1$$

$$\theta_1 = \frac{\sum_{i=1}^{s}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{s}(x_i - \bar{x})^2}$$

$\bar{x} = (2.0 + 2.4 + 1.5 + \dots)/9 = 3.34$

$\bar{y} = (196 + 221 + 136 + \dots)/9 = 256$

$$\theta_1 = \frac{(2.0 - 3.34)(196 - 256) + (2.4 - 3.34)(221 - 256) + \dots}{(2.0 - 3.34)^2 + (2.4 - 3.34)^2 + \dots}$$

$\theta_1 = 39$

$$\theta_0 = \bar{y} - \theta_1 \bar{x}$$

$\theta_0 = 256 - 39 * 3.34$

$\theta_0 = 125.74$

$$\hat{y} = 125.74 + 39x_1$$

7

Multiple Linear Regression

X: Independent variable

Y: Dependent variable

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

$$Co2\ Em = \theta_0 + \theta_1 Engine\ size + \theta_2 Cylinders + \ \ldots$$

Multiple Vs Linear Regression

- Simple Linear Regression:
  - Predict co2emission vs EngineSize of all cars
    - Independent variable (x): EngineSize
    - Dependent variable (y): co2emission
- Multiple Linear Regression:
  - Predict co2emission vs EngineSize and Cylinders of all cars
    - Independent variable (x): EngineSize, Cylinders, etc
    - Dependent variable (y): co2emission

Example

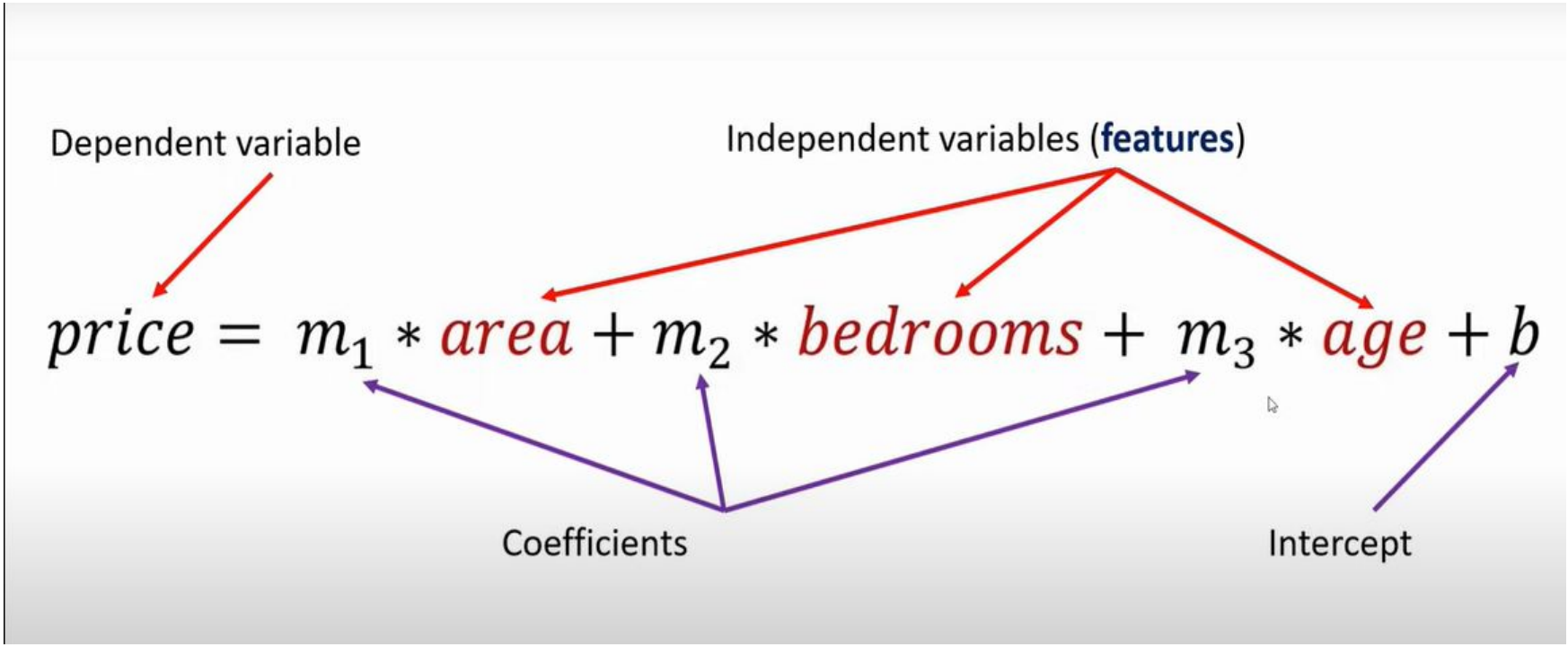
Home prices in Monroe Township, NJ (USA)

area	bedrooms	age	price
2600	3	20	550000
3000	4	15	565000
3200		18	610000
3600	3	30	595000
4000	5	8	760000

Given these home prices find out price of a home that has,

3000 sqr ft area, 3 bedrooms, 40 year old
2500 sqr ft area, 4 bedrooms, 5 year old

Example


Dependent variable
Independent variables (features)
$price = m_1 * area + m_2 * bedrooms + m_3 * age + b$
Coefficients
Intercept

Multiple Linear Regression

- Multiple linear regression is a statistical technique used to analyze a dataset with various independent variables affecting the dependent variable.

- The technique allows researchers to predict a dependent variable outcome based on certain variables' values. It also will enable researchers to assess whether or not there are any interactions between independent variables, which can help them understand more about how they affect each other.

Y = b0 + b1 * x1 + b2 * x2 + b3 * x3 + …… bn * xn

Y = Dependent variable and x1, x2, x3, …… xn = multiple independent variables

Applications of Multiple Regression

- Independent variables effectiveness on prediction
  - Does revision time, test anxiety, lecture attendance and gender have any effect on the exam performance of students?
- Predicting impacts of changes
  - How much does blood pressure go up (or down) for every unit increase (or decrease) in the BMI of a patient?

Example

X: Independent variable

Y: Dependent variable

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

$$Co2\ Em = \theta_0 + \theta_1 Engine\ size + \theta_2 Cylinders + \ \ldots$$

Parameters

$$Co2\ Em = \theta_0 + \theta_1 Engine\ size + \theta_2 Cylinders + \ \ldots$$

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ \ldots + \theta_n x_n$$

$$\hat{y} = \theta^T X$$

$$\theta^T = [\theta_0, \theta_1, \theta_2, \ldots.] \qquad X = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \ldots \end{bmatrix}$$

Error Calculation

$\hat{y} = \theta^T X$

$\hat{y}_i = 140$ the predicted emission of x_i

$y_i = 196$ actual value of x_i

$y_i - \hat{y}_i = 196 - 140 = 56$ residual error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Best Parameter

- How to estimate θ?
  - Ordinary Least Squares
    - Linear algebra operations
    - Takes a long time for large datasets (10K+ rows)
  - An optimization algorithm
    - Gradient Descent
    - Proper approach if you have a very large dataset

Example

	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_COMB	CO2EMISSIONS
0	2.0	4	8.5	196
1	2.4	4	9.6	221
2	1.5	4	5.9	136
3	3.5	6	11.1	255
4	3.5	6	10.6	244
5	3.5	6	10.0	230
6	3.5	6	10.1	232
7	3.7	6	11.1	255
8	3.7	6	11.6	267
9	2.4	4	9.2	?

Example

$$\hat{y} = \theta^T X$$

$$\theta^T = [125, 6.2, 14, \ldots]$$

$$\hat{y} = 125 + 6.2x_1 + 14x_2 +$$

$$Co2Em = 125 + 6.2EngSize + 14\ \text{Cylinders} + \ldots$$

$$Co2Em = 125 + 6.2 \times 2.4 + 14 \times 4 + \ldots$$

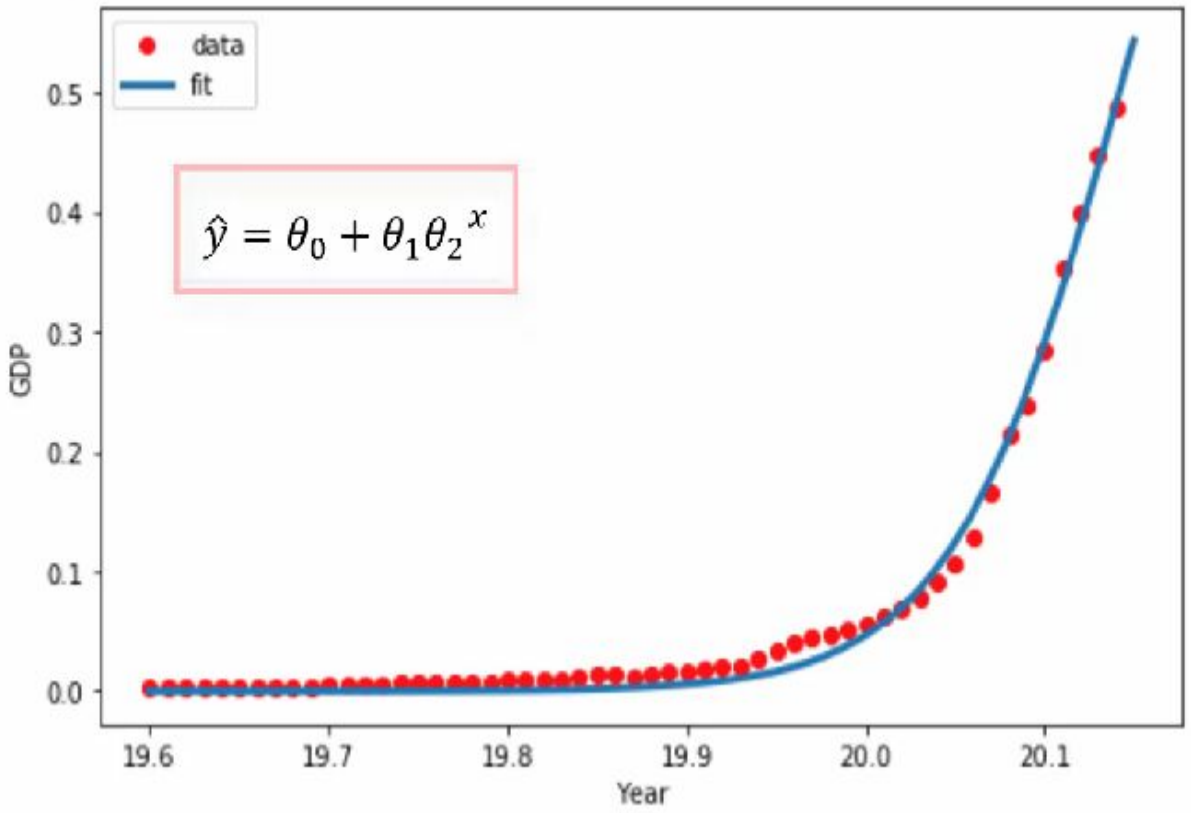
$$Co2Em = 214.1$$

Questions

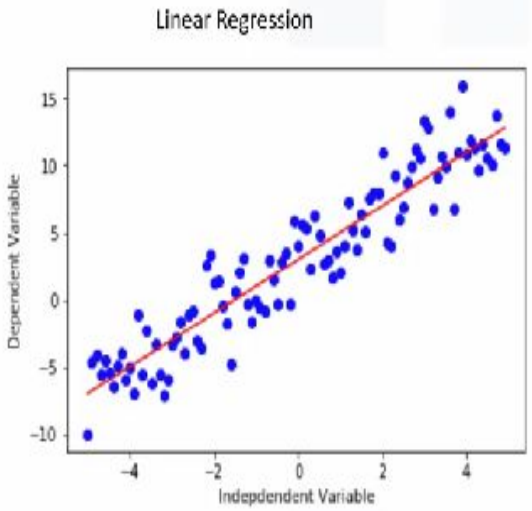
- How to determine whether to use simple or multiple linear regression?
- How many independent variables should you use?
- Should the independent variable be continuous?
- What are the linear relationships between the dependent variable and the independent variables?

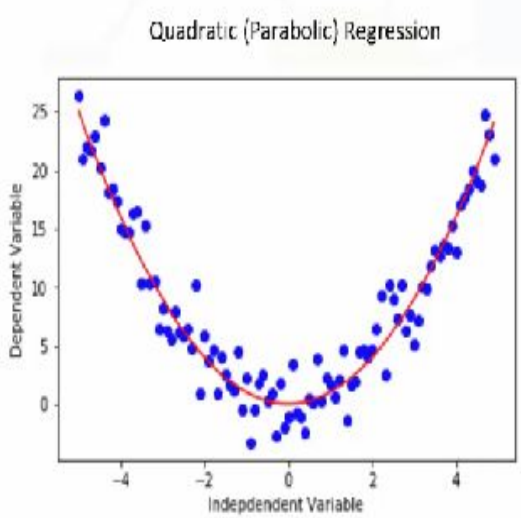
Non Linear Regression

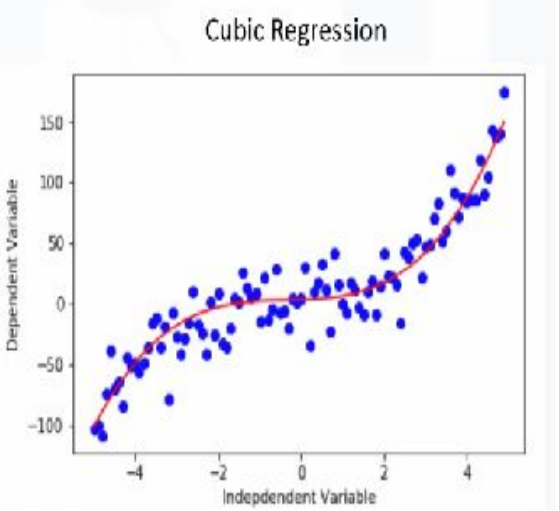
	Year	Value
0	1960	5.918412e+10
1	1961	4.955705e+10
2	1962	4.668518e+10
3	1963	5.009730e+10
4	1964	5.906225e+10
5	1965	6.970915e+10
6	1966	7.587943e+10
7	1967	7.205703e+10
8	1968	6.999350e+10
9	1969	7.871882e+10
...	.....	.....



Types of Regressions







Non Linear Regression

- To model non-linear relationship between the dependent variable and a set of independent variables
- $\hat{y}$ must be a non-linear function of the parameters θ, not necessarily the features x

$$\hat{y} = \theta_0 + \theta_2{}^2 \mathrm{x}$$

$$\hat{y} = \theta_0 + \theta_1 \theta_2{}^x$$

$$\hat{y} = \log(\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3)$$

$$\hat{y} = \frac{\theta_0}{1 + \theta_1{}^{(x-\theta_2)}}$$

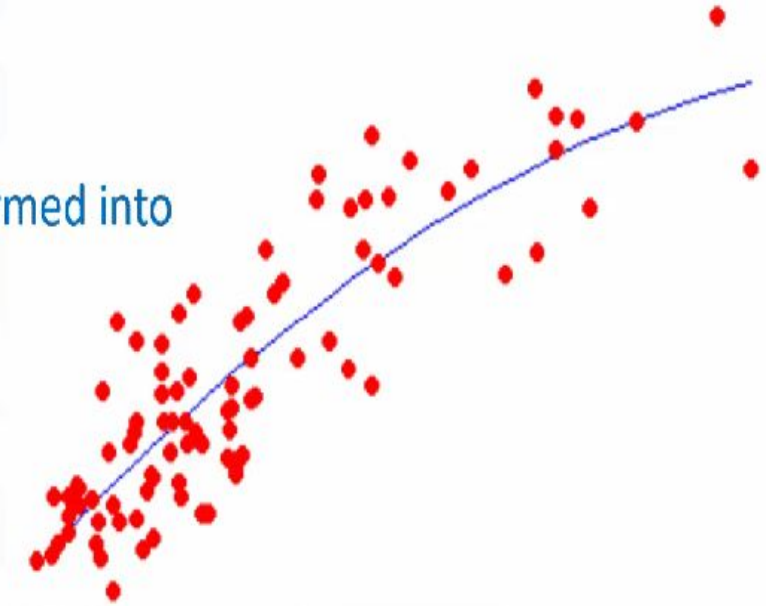
Polynomial Regression

- Some curvy data can be modeled by a polynomial regression
- For example:

$$\hat{y} = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

- A polynomial regression model can be transformed into linear regression model.

$$x_1 = x$$
$$x_2 = x^2$$
$$x_3 = x^3$$



$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$ → Multiple linear regression → Least Squares

Parameter Tuning

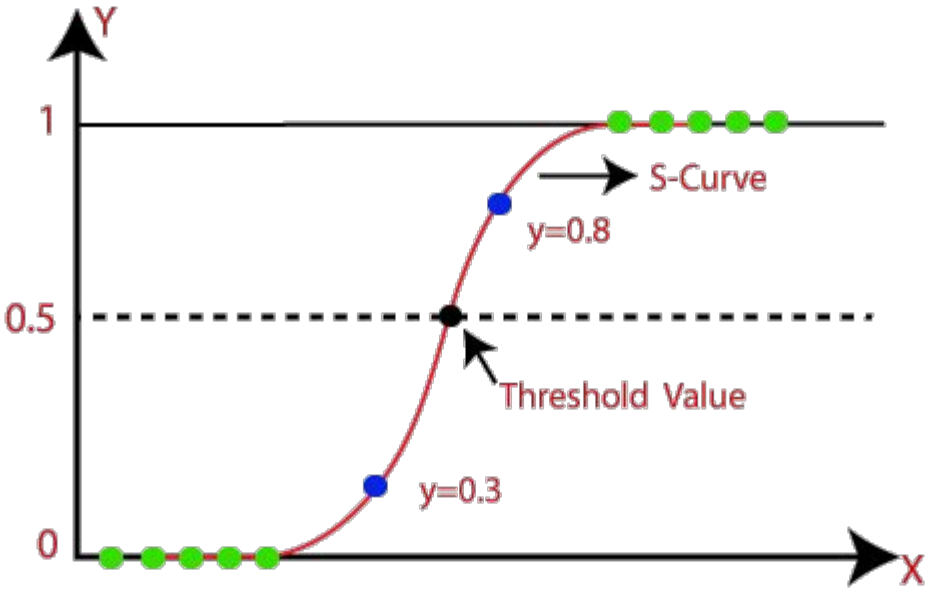
- How to estimate θ?
  - Ordinary Least Squares
    - Linear algebra operations
    - Takes a long time for large datasets (10K+ rows)
  - An optimization algorithm
    - Gradient Descent
    - Proper approach if you have a very large dataset

How to Choose?

- How can I know if a problem is linear or non-linear in an easy way?
  - Inspect visually
  - Based on accuracy
- How should I model my data, if it displays non-linear on a scatter plot?
  - Polynomial regression
  - Non-linear regression model
  - Transform your data

Logistic Regression

- It is used for predicting the categorical dependent variable using a given set of independent variables.Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).



Logistic Regression

Linear Regression

1. Home prices
2. Weather
3. Stock price

Predicted value is continuous

Classification

1. Email is spam or not
2. Will customer buy life insurance?
3. Which party a person is going to vote for?
   1. Democratic
   2. Republican
   3. Independent

Predicted value is categorical

Classification & It's Types

- Logistic regression is one of the techniques used for Classification.

Example

Will customer buy life insurance?

1. Yes
2. No

Binary Classification

Example

Which party a person is going to vote for?

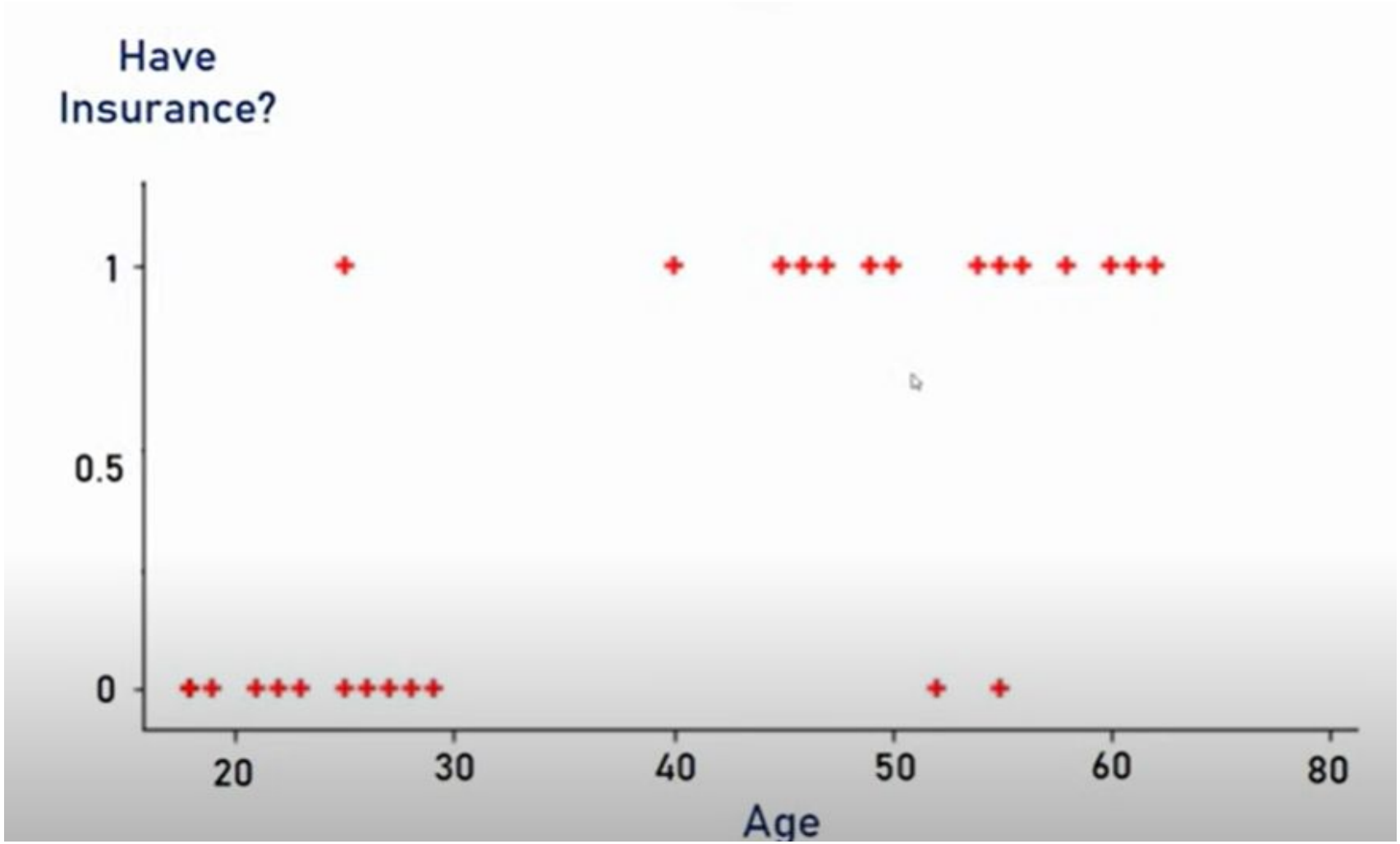
1. Democratic
2. Republican
3. Independent

Multiclass Classification

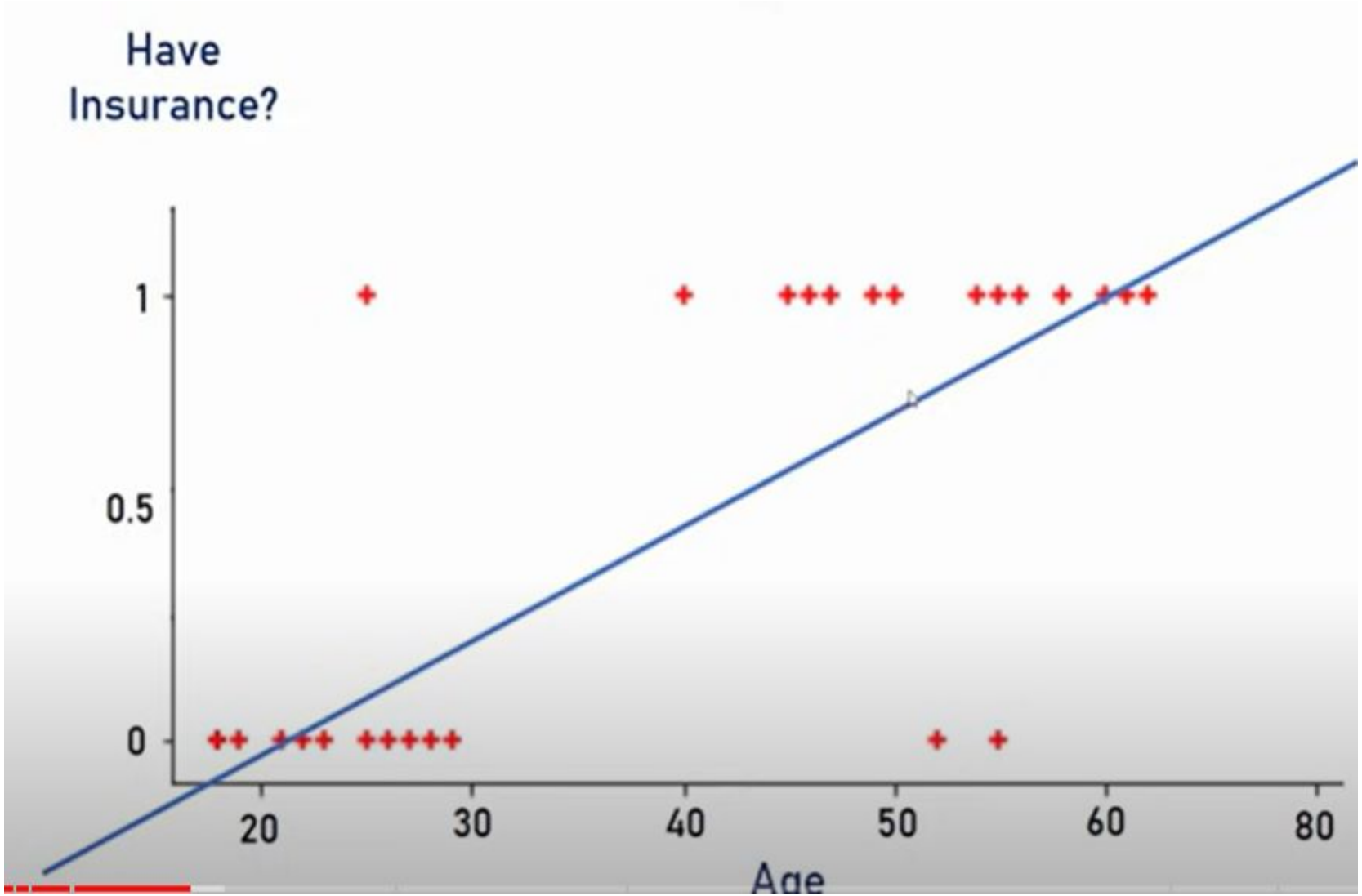
Example

age	have_insurance
22	0
25	0
47	1
52	0
46	1
56	1
55	0
60	1
62	1
61	1
18	0
28	0
27	0
29	0

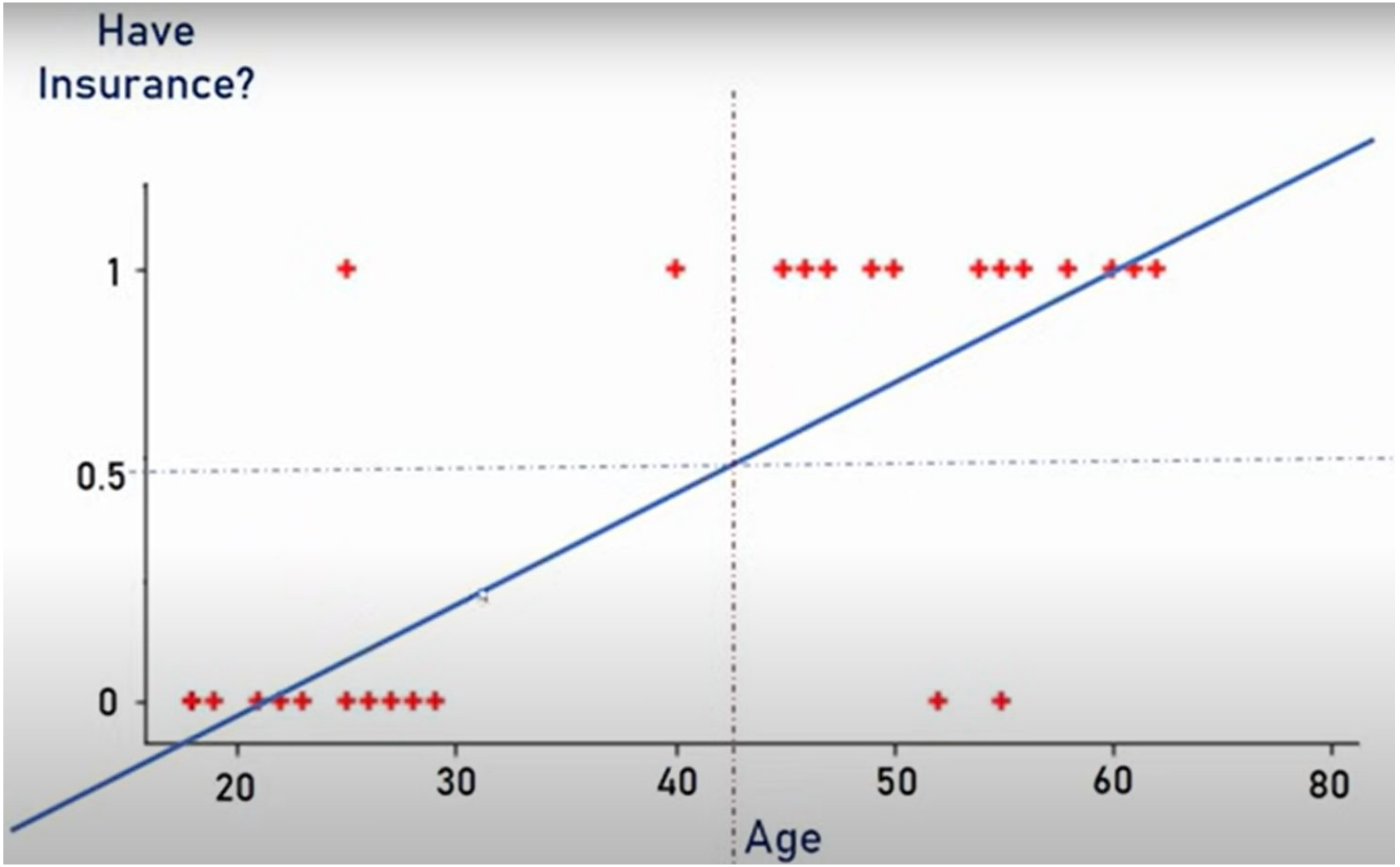
Graph Plot

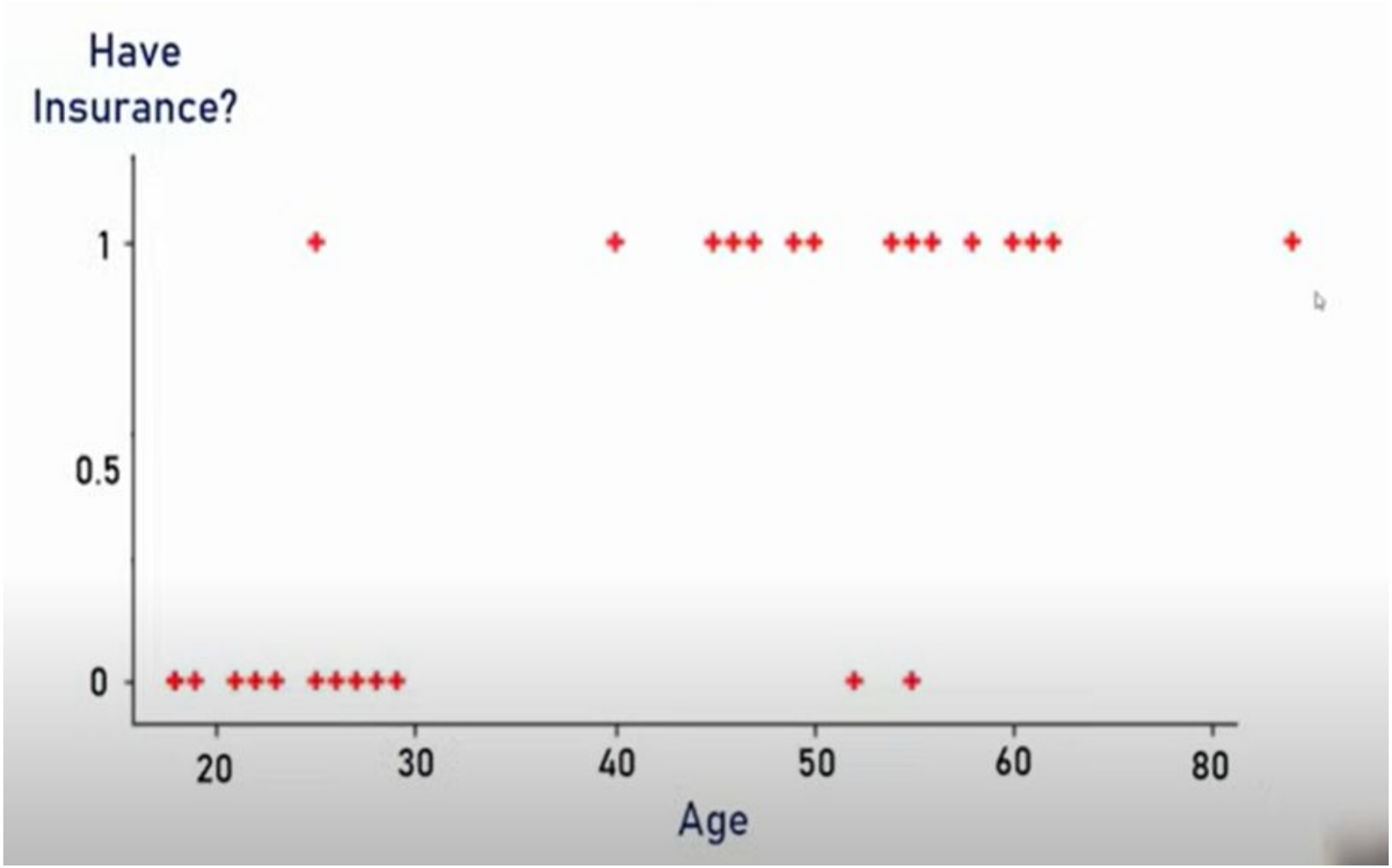


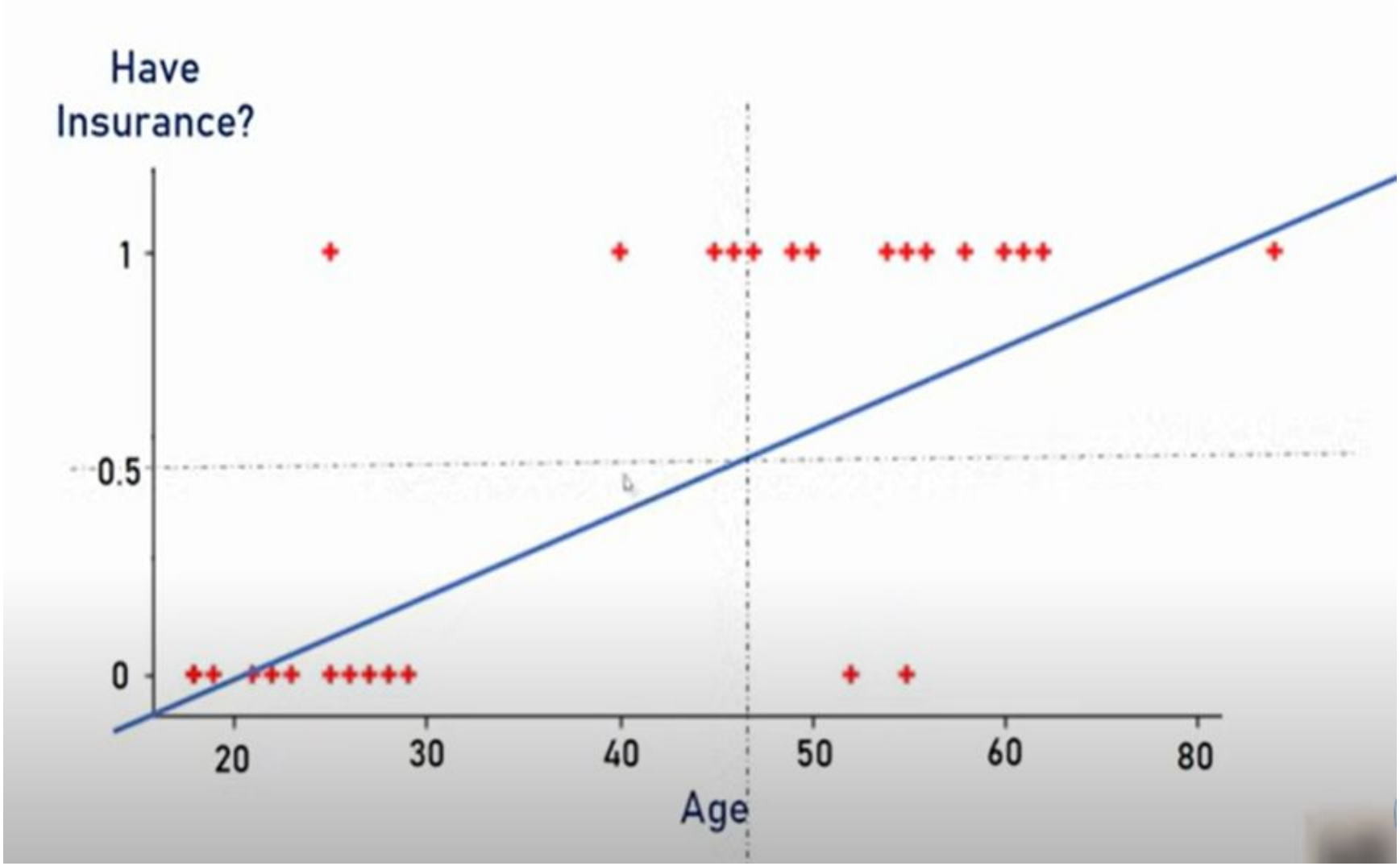
Linear line



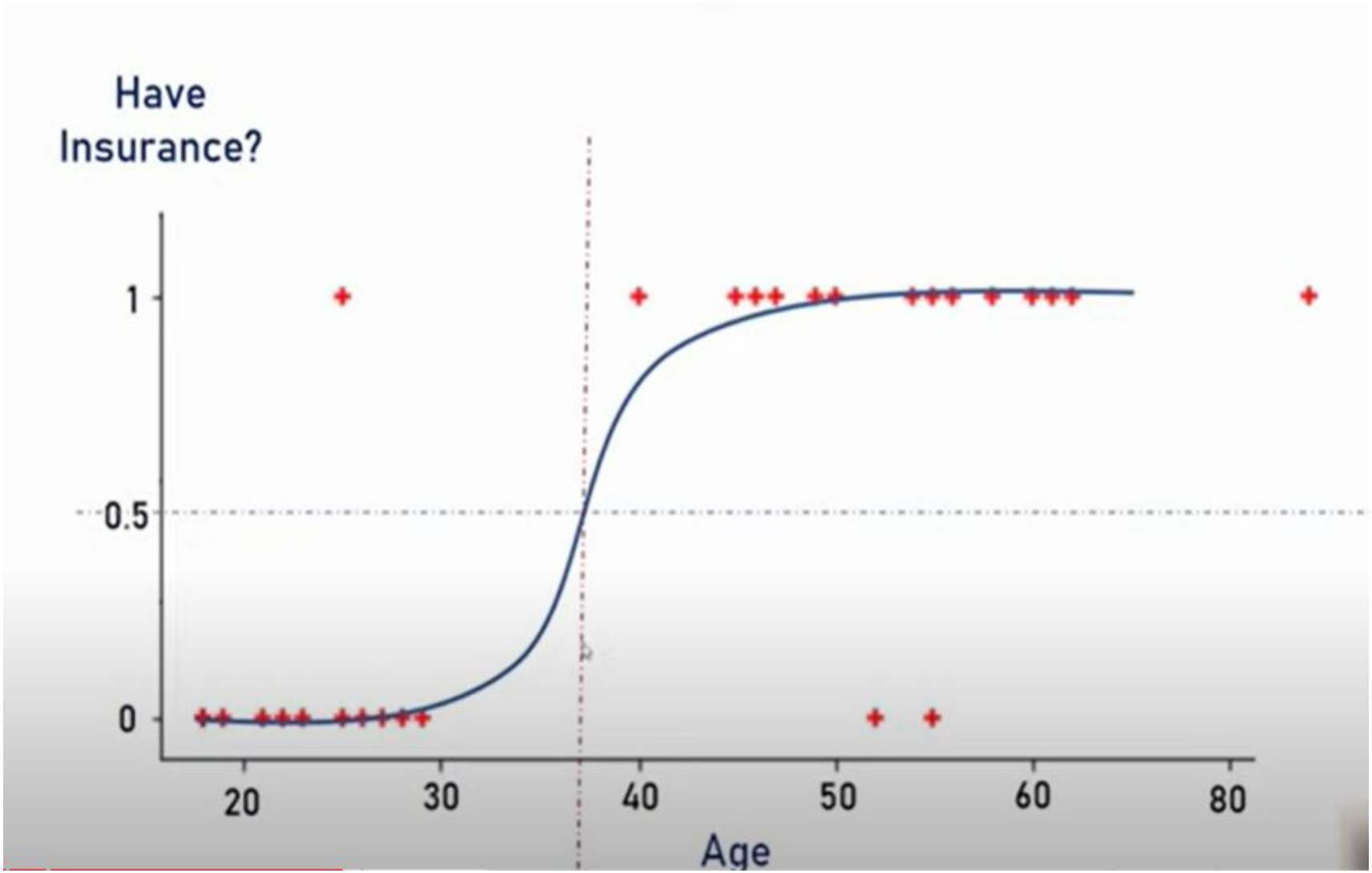
Create a Threshold

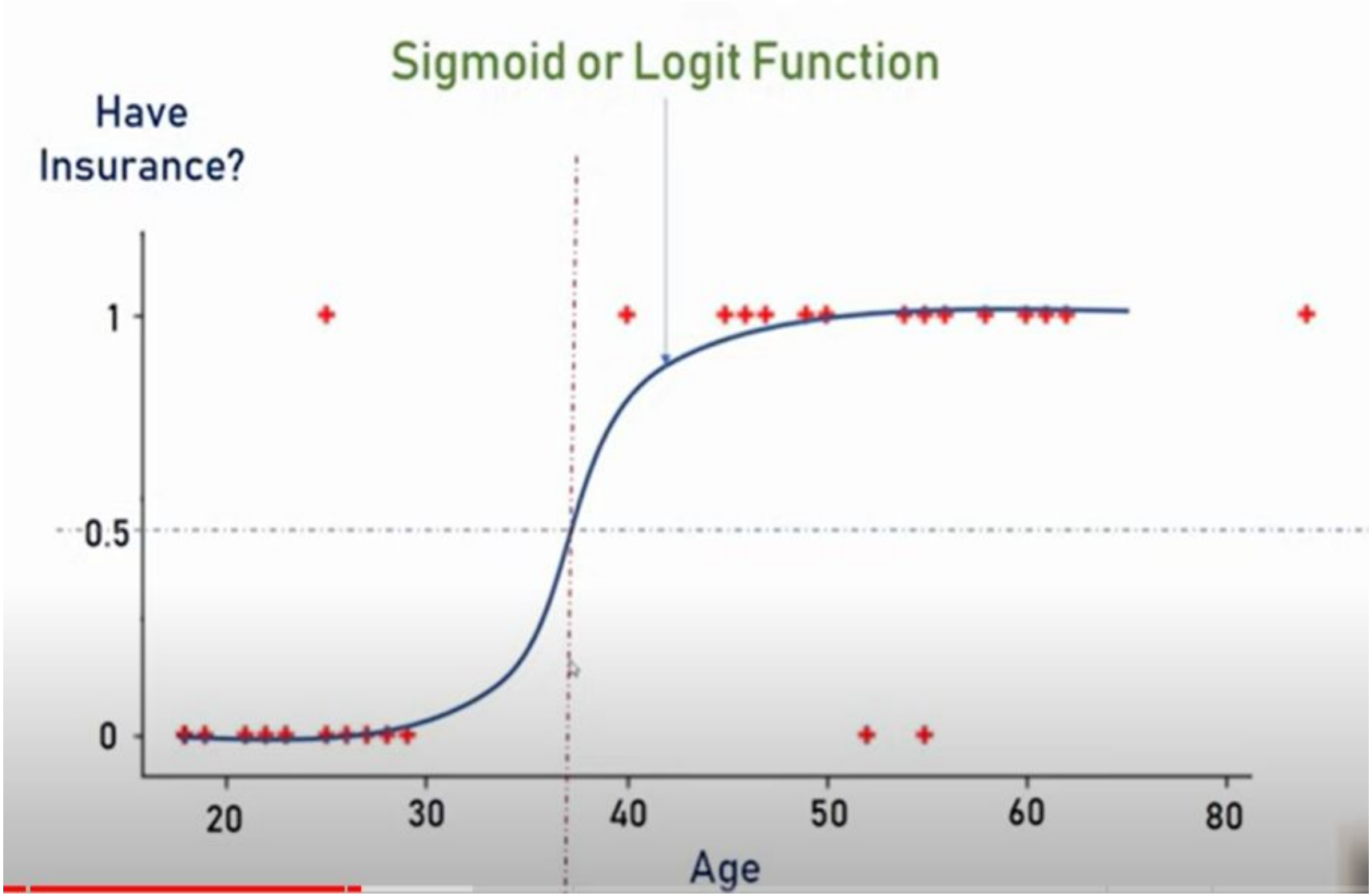



Have
Insurance?
1
0.5
0
20
30
40
50
60
80
Age


Have
Insurance?
1
0.5
0
20
30
40
50
60
80
Age

Sigmoid Curve or Logistic




Sigmoid or Logit Function
Have Insurance?
1
0.5
0
20
30
40
50
60
80
Age

Sigmoid Function

$$sigmoid(z) = \frac{1}{1 + e^{-z}}$$

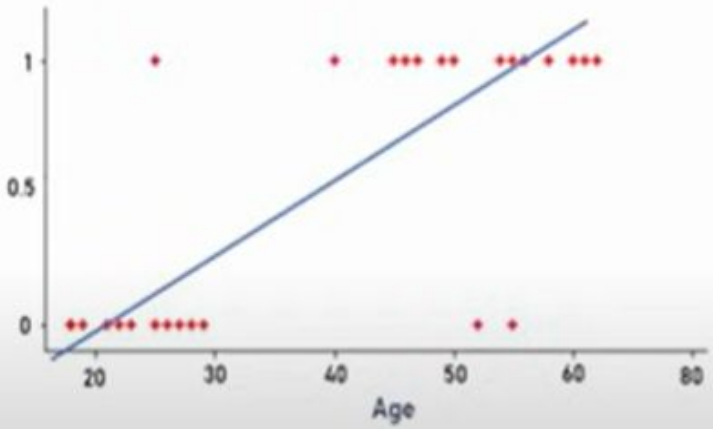
e = Euler's number ~ 2.71828

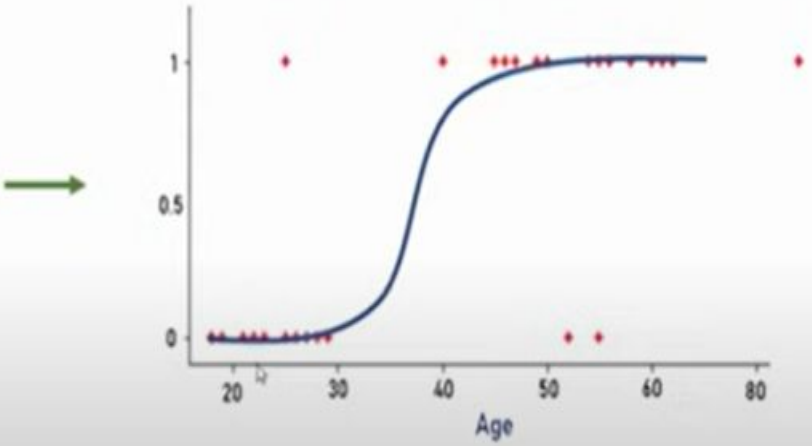
Sigmoid function converts input into range 0 to 1

Difference Between Linear line & Logistic

$$y = m * x + b$$

$$y = \frac{1}{1 + e^{-(m*x+b)}}$$





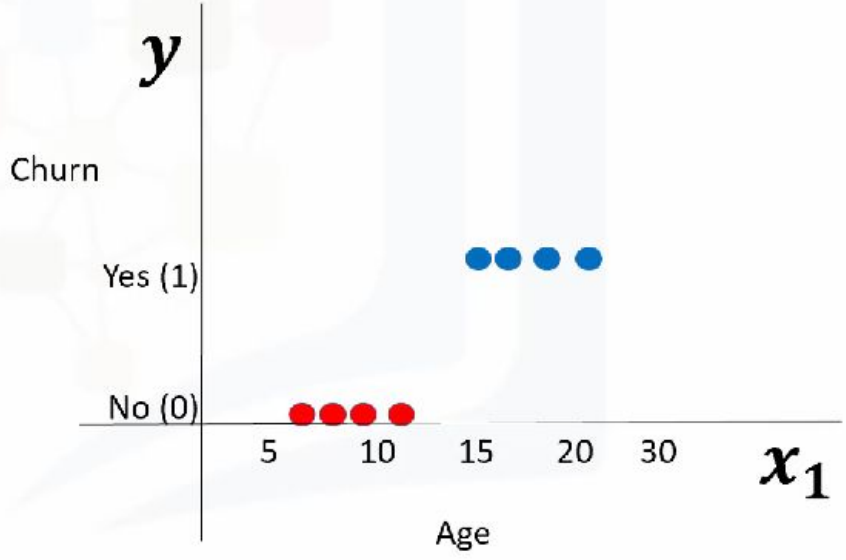
Linear Vs Logistic Regression

	tenure	age	address	income	ed	employ	equip	callcard	wireless	churn
0	11.0	33.0	7.0	136.0	5.0	5.0	0.0	1.0	1.0	Yes
1	33.0	33.0	12.0	33.0	2.0	0.0	0.0	0.0	0.0	Yes
2	23.0	30.0	9.0	30.0	1.0	2.0	0.0	0.0	0.0	No
3	38.0	35.0	5.0	76.0	2.0	10.0	1.0	1.0	1.0	No
4	7.0	35.0	14.0	80.0	2.0	15.0	0.0	1.0	0.0	No

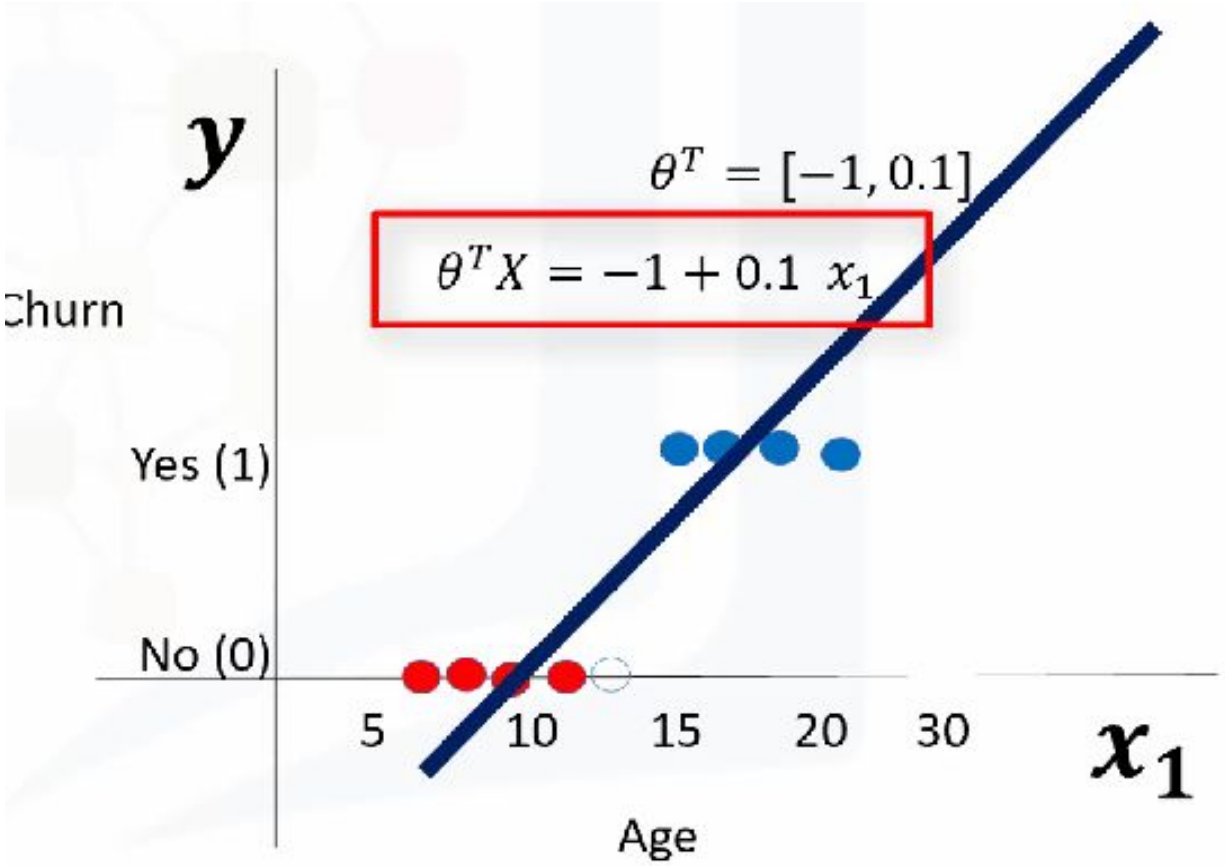
$$\hat{y} = P(y=1|x)$$

Predicting churn using linear regression

	tenure	age	address	income	ed	employ	equip	callcard	wireless	churn
0	11.0	33.0	7.0	136.0	5.0	5.0	0.0	1.0	1.0	1
1	33.0	33.0	12.0	33.0	2.0	0.0	0.0	0.0	0.0	1
2	23.0	30.0	9.0	30.0	1.0	2.0	0.0	0.0	0.0	0
3	38.0	35.0	5.0	76.0	2.0	10.0	1.0	1.0	1.0	0
4	7.0	35.0	14.0	80.0	2.0	15.0	0.0	1.0	0.0	0



Predicting churn using linear regression



Predicting churn using linear regression

$$\theta^T X = \theta_0 + \theta_1 x_1$$

$p_1 = [13]$ → $\theta^T X = -1 + 0.1 \,.\, x_1$

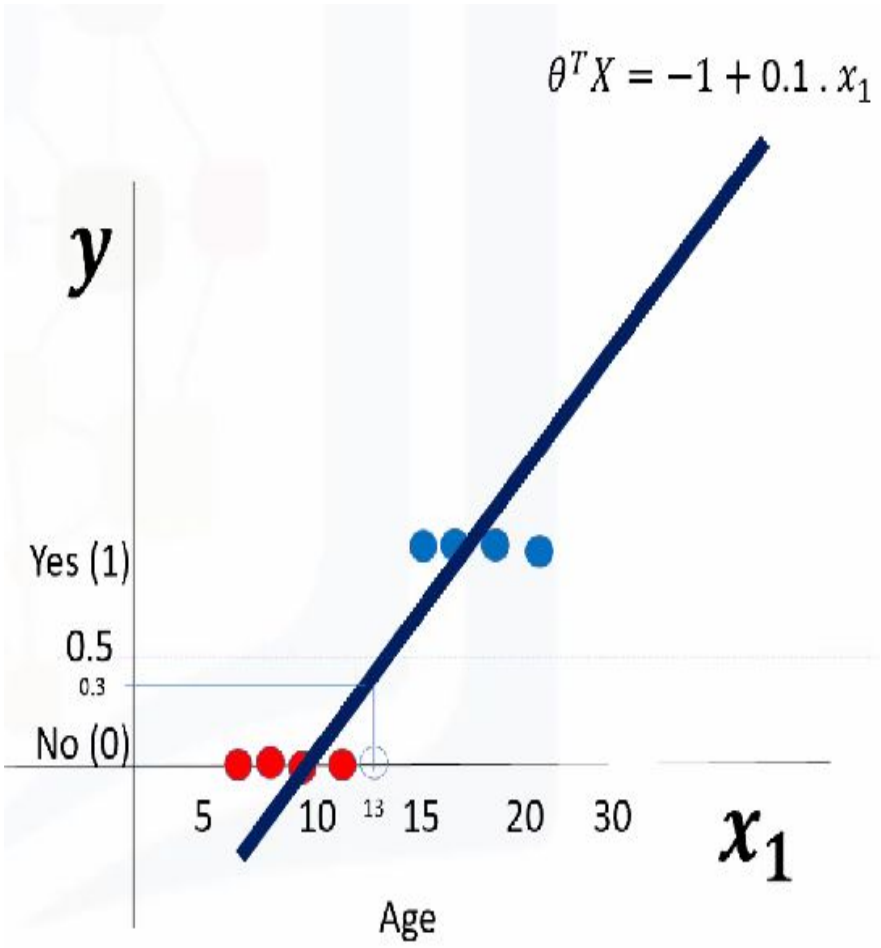
= -1 + 0.1 x 13

= 0.3

$$\hat{y} = \begin{cases} 0 & \text{if } \theta^T X < 0.5 \\ 1 & \text{if } \theta^T X \geq 0.5 \end{cases}$$

$\theta^T X = 0.3$

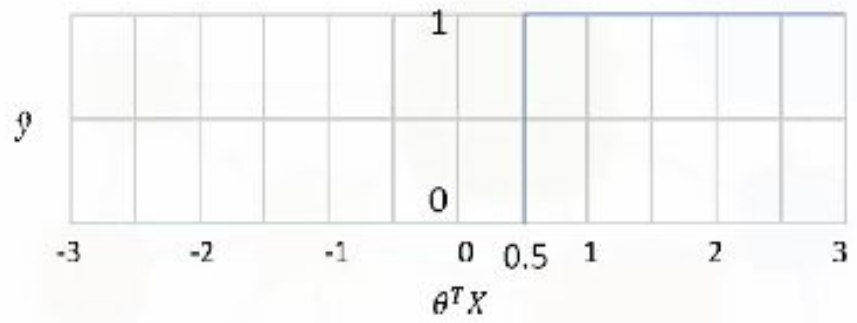
$\theta^T X < 0.5$ → Class 0

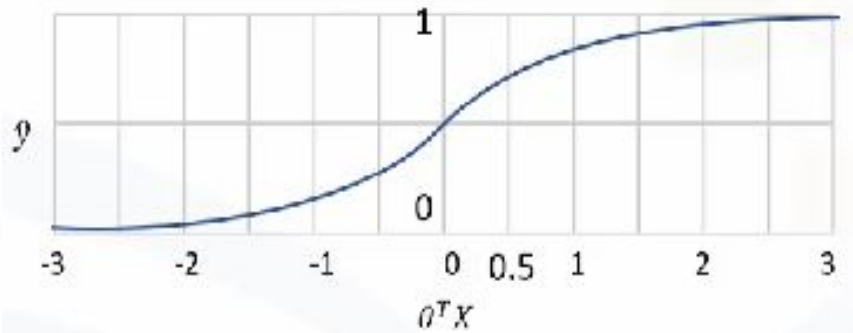


Problems with Linear Regression

$$\theta^T X = \theta_0 + \theta_1 x_1 + \cdots$$

$$\sigma(\theta^T X) = \sigma(\theta_0 + \theta_1 x_1 + \cdots)$$



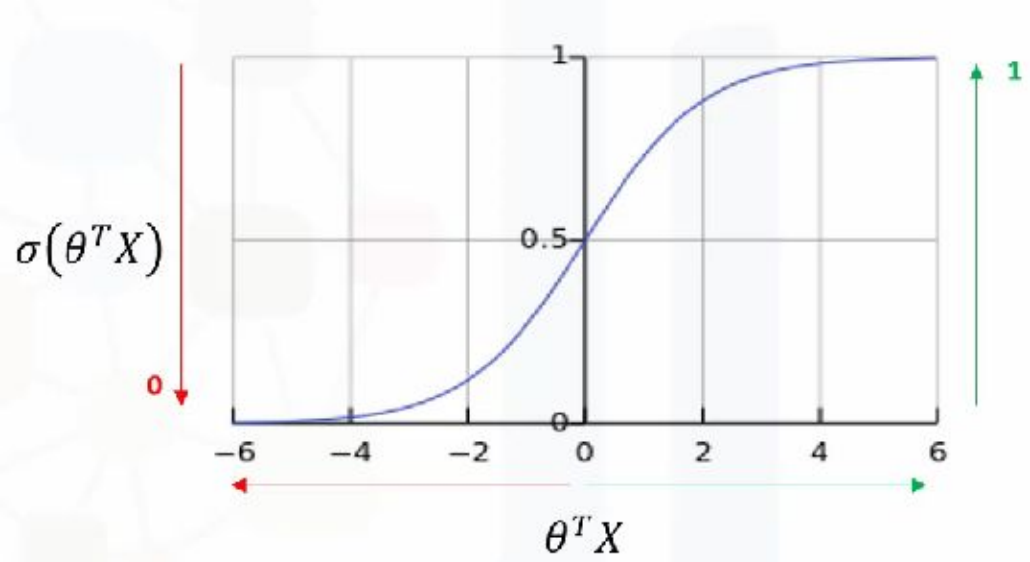


$$\hat{y} = \begin{cases} 0 & \text{if } \theta^T X < 0.5 \\ 1 & \text{if } \theta^T X \geq 0.5 \end{cases}$$

$$\hat{y} = \sigma(\theta^T X)$$

Sigmoid Function in Logistic Regression

$$\sigma(\theta^T X) = \frac{1}{1 + e^{-\theta^T X}}$$



$$\sigma(\theta^T X) = \frac{1}{1 + e^{-\theta^T X}}$$

$\sigma(\theta^T X) = 1$

$\sigma(\theta^T X) = 0$

P(y=1|x)

P(y=1|x)

Classification of churn using Logistic Regression

What is the output of our model?

- P(Y=1|X)
- P(y=0|X) = 1 - P(y=1|x)

- P(Churn=1|income,age) = 0.8
- P(Churn=0|income,age) = 1 - 0.8 = 0.2

$\sigma(\theta^T X)$ ⟶ P(y=1|x)

$1 - \sigma(\theta^T X)$ ⟶ P(y=0|x)

Training Process

$\sigma(\theta^T X) \rightarrow P(y=1|x)$

1. Initialize θ. $\theta = [-1,2]$
2. Calculate $\hat{y} = \sigma(\theta^T X)$ for a customer. $\hat{y} = \sigma([-1,2] \times [2,5]) = 0.7$
3. Compare the output of $\hat{y}$ with actual output of customer, y, and record it as error. Error = 1-0.7 = 0.3
4. Calculate the error for all customers. $Cost = J(\theta)$
5. Change the θ to reduce the cost. θ_{new}
6. Go back to step 2.

General Cost Function

$\sigma(\theta^T X) \longrightarrow$ P(y=1|x)

- Change the weight -> Reduce the cost
- Cost function $Cost(\hat{y}, y) = \frac{1}{2}(\sigma(\theta^T X) - y)^2$

$$J(\theta) = \frac{1}{m}\sum_{i-1}^{m} Cost(\hat{y}, y)$$

Minimizing Cost function of the model

- How to find the best parameters for our model?
  - Minimize the cost function
- How to minimize the cost function?
  - Using Gradient Descent
- What is gradient descent?
  - A technique to use the derivative of a cost function to change the parameter values, in order to minimize the cost

Plotting the cost function of the model

- Model $\hat{y}$
- Actual Value y=1 or 0

- If Y=1, and $\hat{y}$=1 → cost = 0

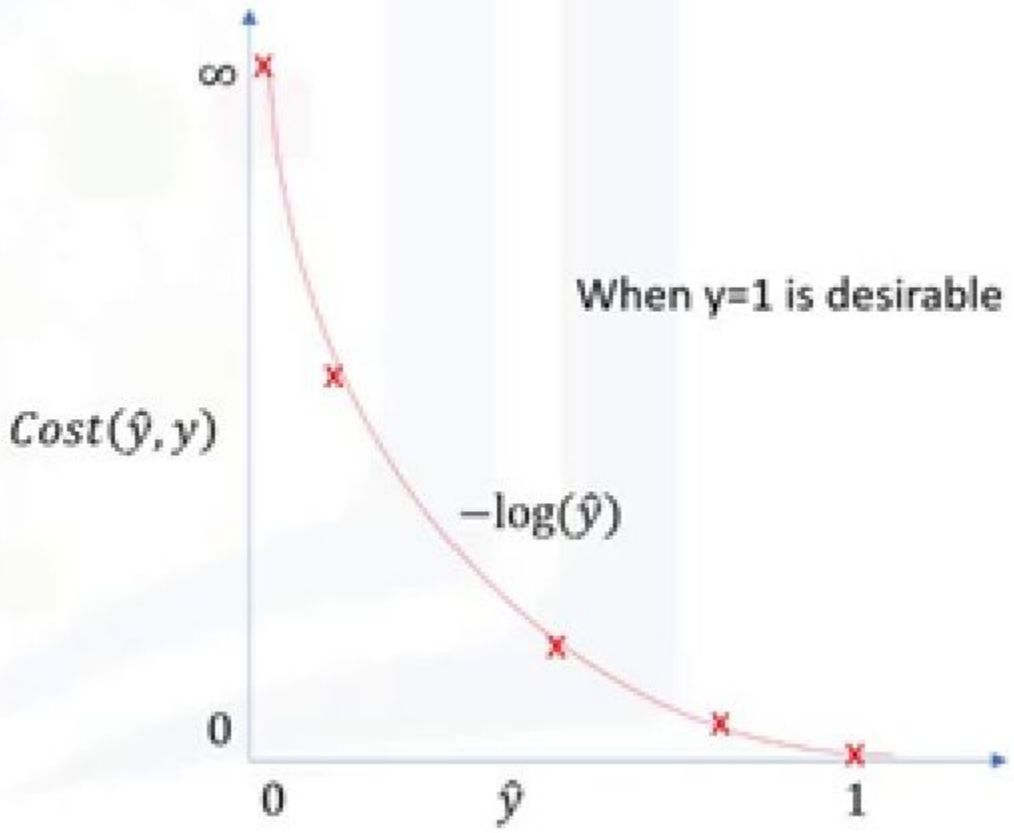
- If Y=1, and $\hat{y}$=0 → cost = large

Plotting the cost function of the model

- Model $\hat{y}$
- Actual Value y=1 or 0

- If Y=1, and $\hat{y}$=1 → cost = 0

- If Y=1, and $\hat{y}$=0 → cost = large



General Cost Function

$\sigma(\theta^T X)$ → P(y=1|x)

- Change the weight -> Reduce the cost
- Cost function $Cost(\hat{y}, y) = \frac{1}{2}(\sigma(\theta^T X) - y)^2$

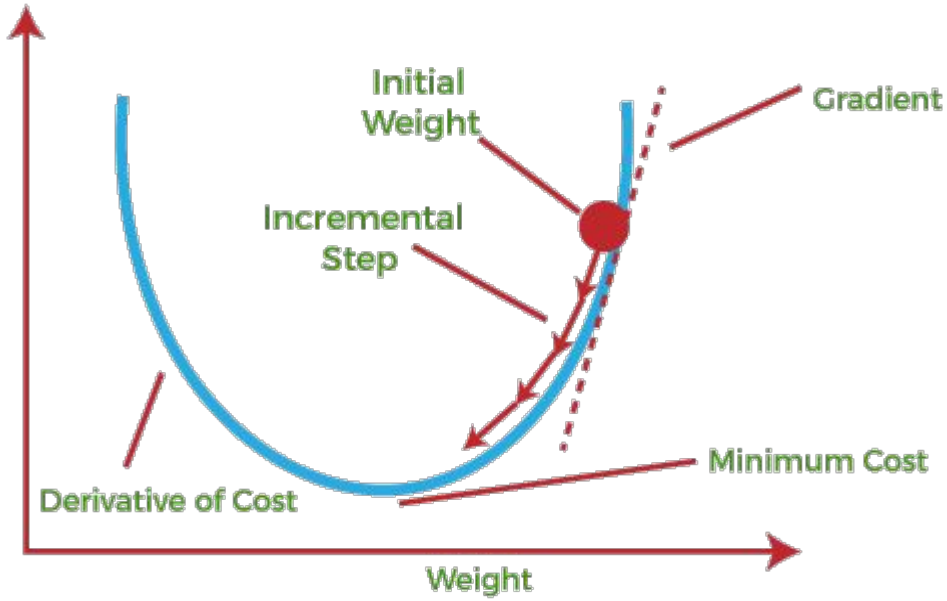
$$J(\theta) = \frac{1}{m}\sum_{i-1}^{m} Cost(\hat{y}, y)$$

$$Cost(\hat{y}, y) = \begin{cases} -\log(\hat{y}) & if\ y = 1 \\ -\log(1-\hat{y}) & if\ y = 0 \end{cases}$$

$$J(\theta) = -\frac{1}{m}\sum_{i-1}^{m} y^i \log(\hat{y}^i) + (1-y^i)\log(1-\hat{y}^i)$$

Gradient Descent

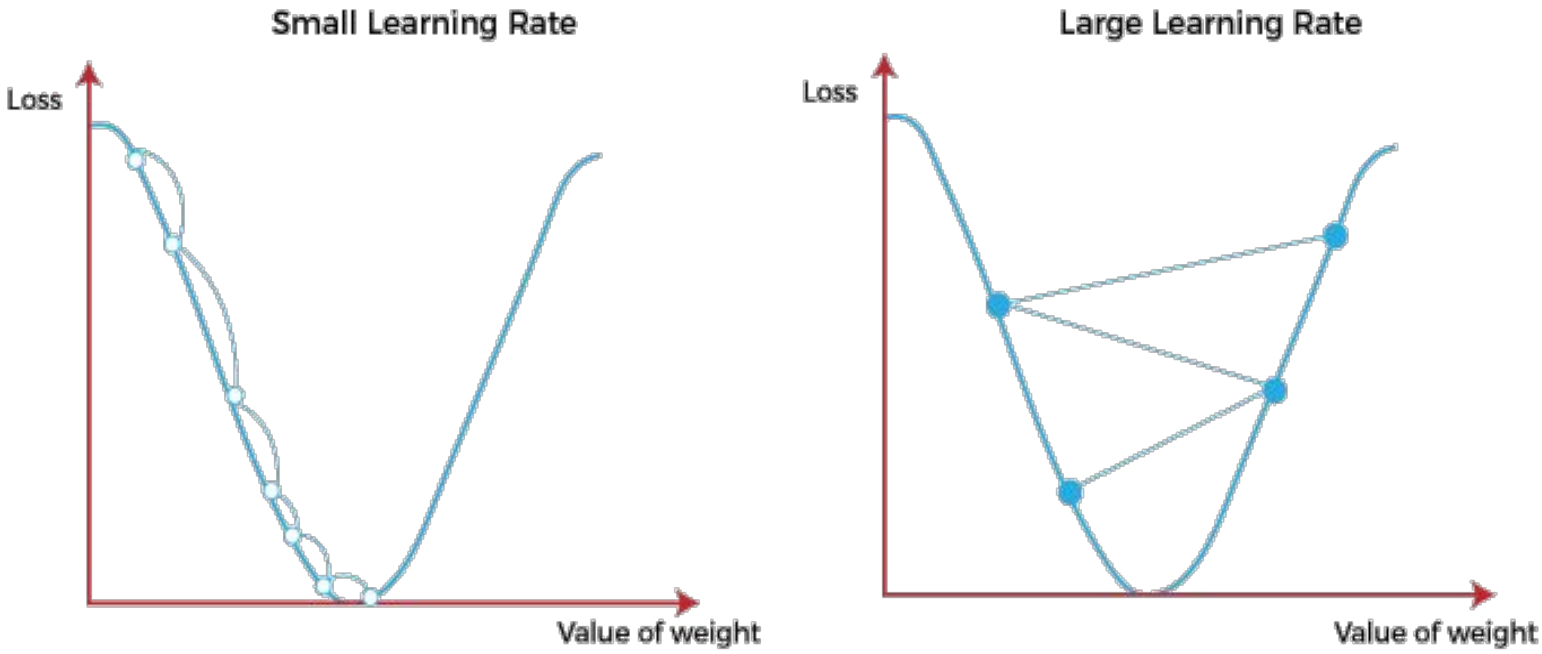
- Gradient Descent is known as one of the most commonly used optimization algorithms to train machine learning models by means of minimizing errors between actual and expected results. Further, gradient descent is also used to train Neural Networks.
- Gradient Descent is defined as one of the most commonly used iterative optimization algorithms of machine learning to train the machine learning and deep learning models. It helps in finding the local minimum of a function.



Learning Rate

- The main objective of gradient descent is to minimize the cost function or the error between expected and actual. To minimize the cost function, two data points are required:

  1)Direction 2)Learning Rate

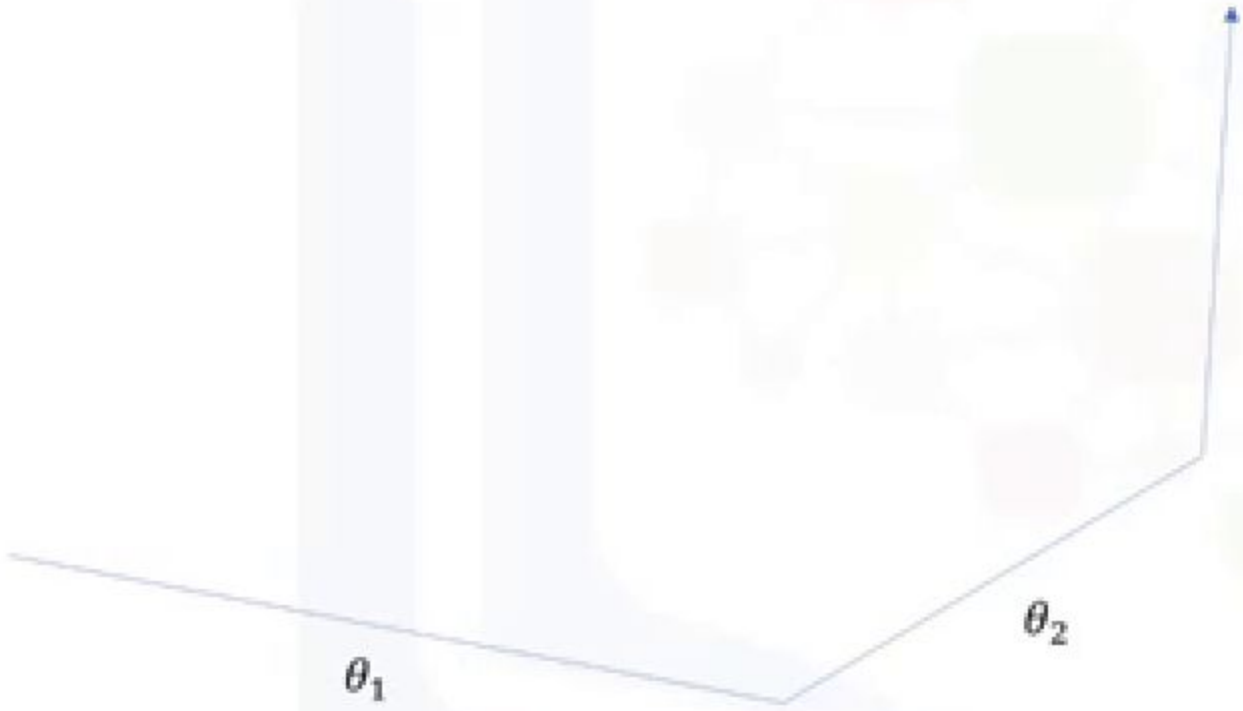


Using Gradient Descent to Minimize the cost

$$\hat{y} = \sigma(\theta_1 x_1 + \theta_2 x_2)$$

$$J(\theta) = -\frac{1}{m}\sum_{i-1}^{m} y^i \log(\hat{y}^i) + (1 - y^i)\log(1 - \hat{y}^i)$$

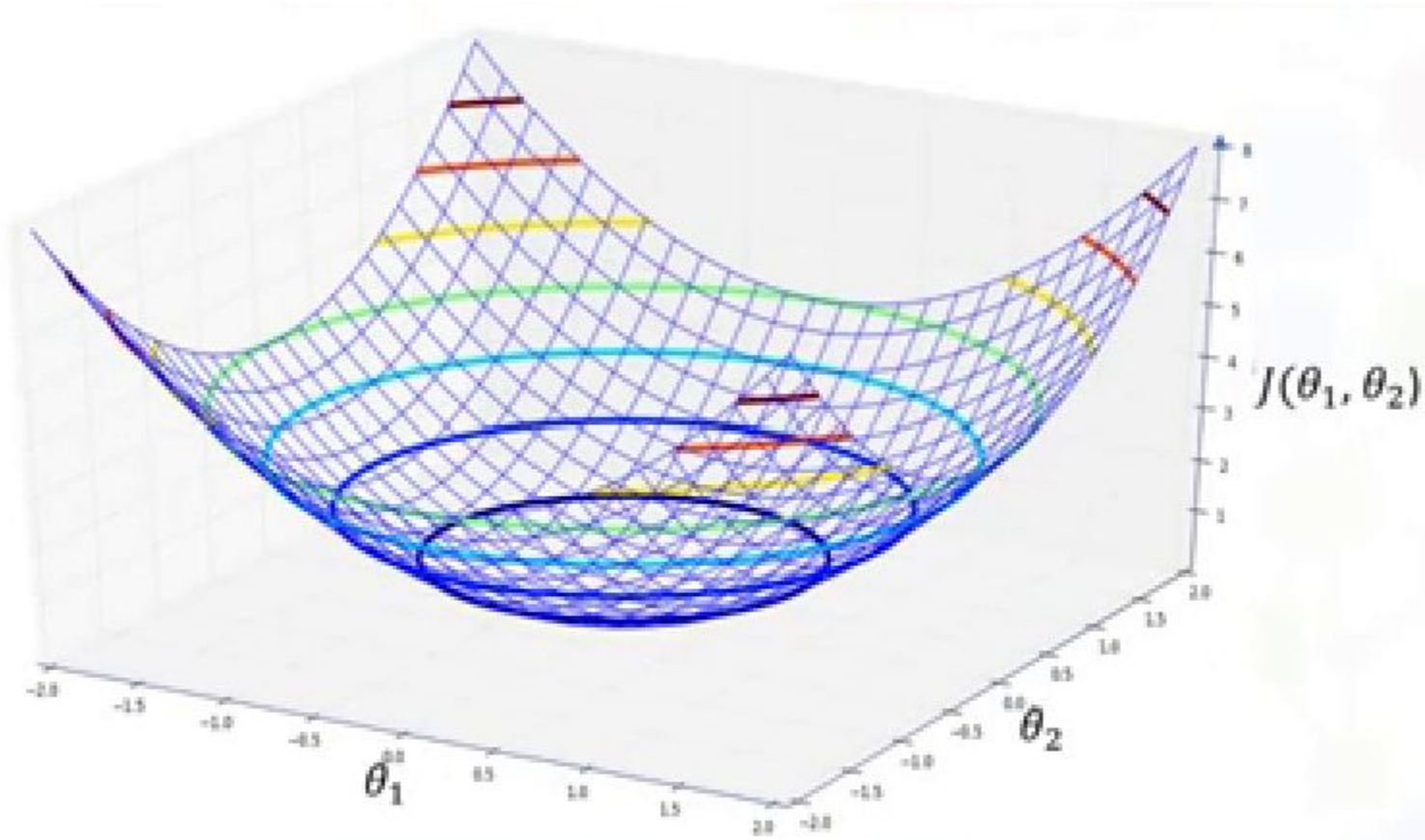
Using Gradient Descent to Minimize the cost



$$\hat{y} = \sigma(\theta_1 x_1 + \theta_2 x_2)$$

$$J(\theta) = -\frac{1}{m}\sum_{i-1}^{m} y^i \log(\hat{y}^i) + (1 - y^i)\log(1 - \hat{y}^i)$$

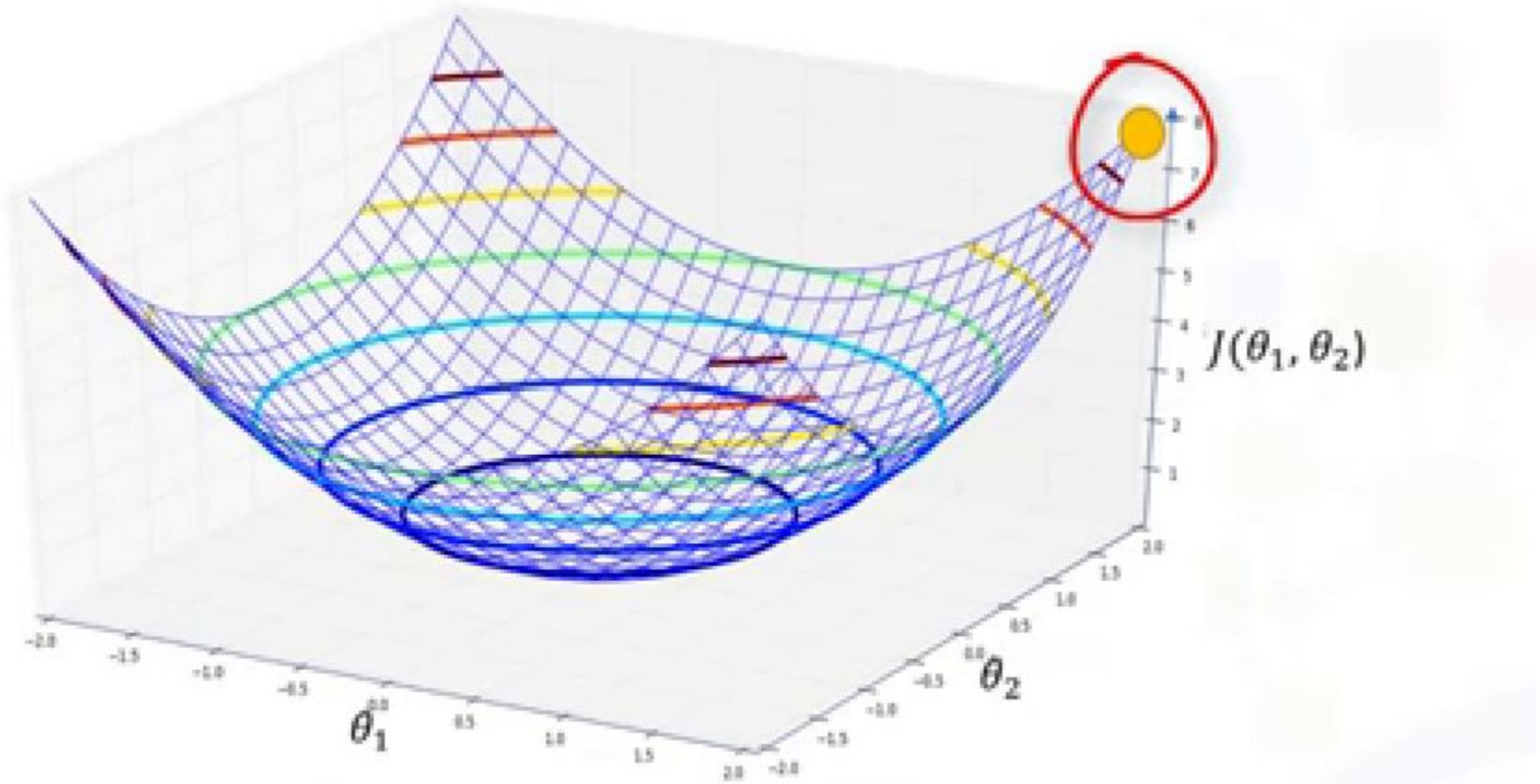
Using Gradient Descent to Minimize the cost



$$\hat{y} = \sigma(\theta_1 x_1 + \theta_2 x_2)$$

$$J(\theta) = -\frac{1}{m}\sum_{i-1}^{m} y^i \log(\hat{y}^i) + (1 - y^i)\log(1 - \hat{y}^i)$$

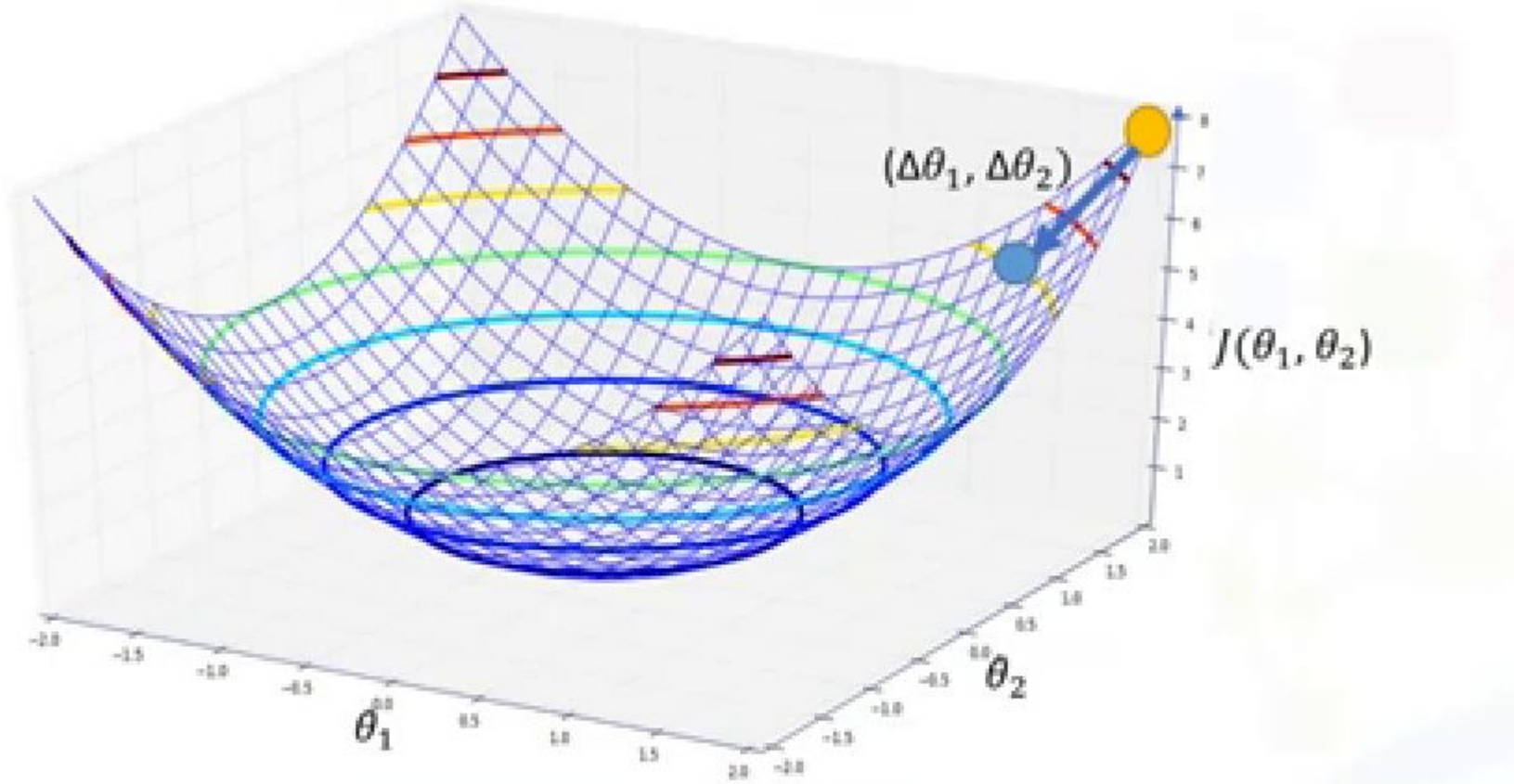
Using Gradient Descent to Minimize the cost



$$\hat{y} = \sigma(\theta_1 x_1 + \theta_2 x_2)$$

$$J(\theta) = -\frac{1}{m}\sum_{i-1}^{m} y^i \log(\hat{y}^i) + (1 - y^i)\log(1 - \hat{y}^i)$$

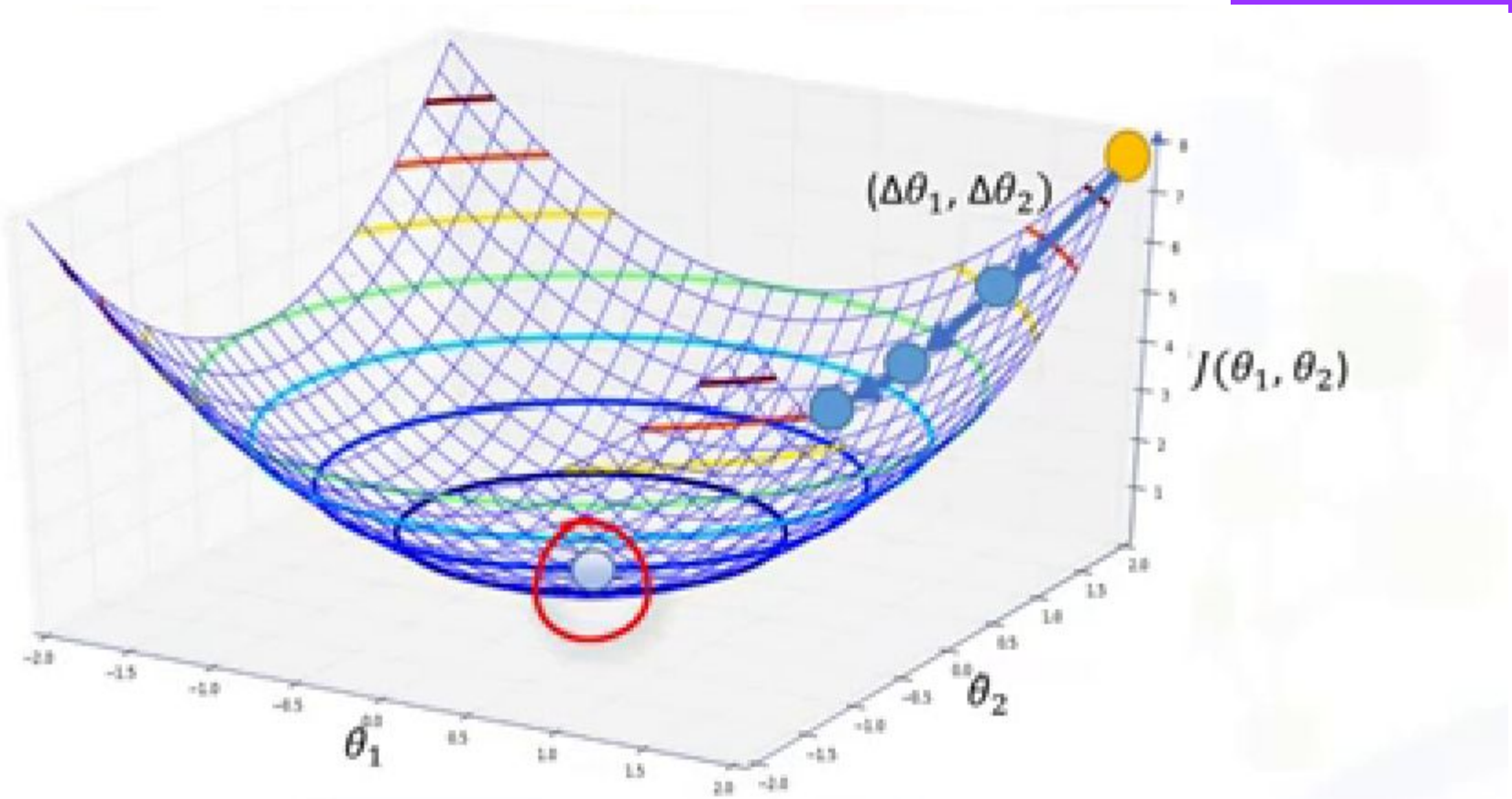
Using Gradient Descent to Minimize the cost



$$\hat{y} = \sigma(\theta_1 x_1 + \theta_2 x_2)$$

$$J(\theta) = -\frac{1}{m}\sum_{i-1}^{m} y^i \log(\hat{y}^i) + (1 - y^i)\log(1 - \hat{y}^i)$$

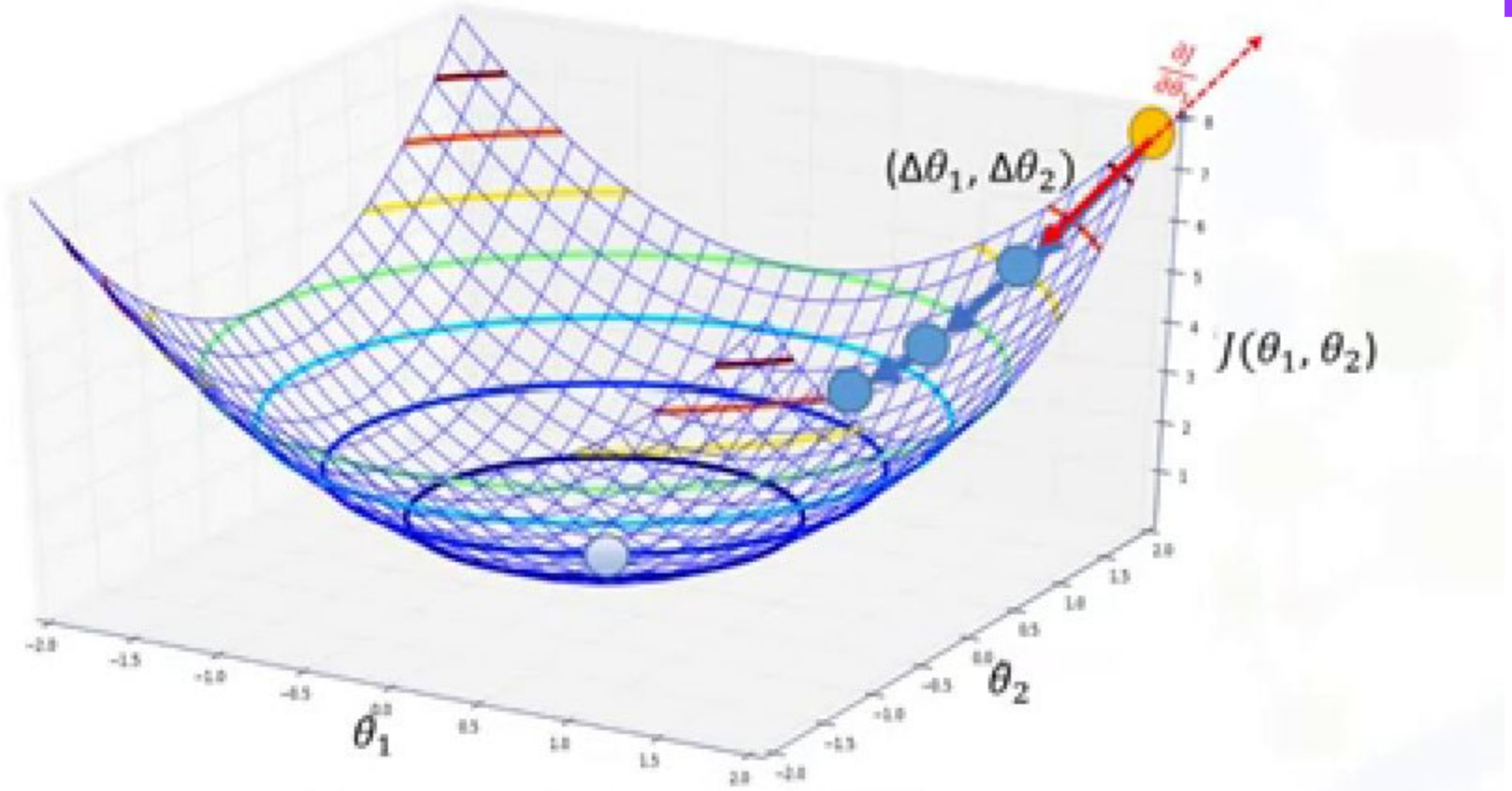
Using Gradient Descent to Minimize the cost



$$\hat{y} = \sigma(\theta_1 x_1 + \theta_2 x_2)$$

$$J(\theta) = -\frac{1}{m}\sum_{i-1}^{m} y^i \log(\hat{y}^i) + (1 - y^i)\log(1 - \hat{y}^i)$$

Using Gradient Descent to Minimize the cost



$$\hat{y} = \sigma(\theta_1 x_1 + \theta_2 x_2)$$

$$J(\theta) = -\frac{1}{m}\sum_{i-1}^{m} y^i \log(\hat{y}^i) + (1 - y^i)\log(1 - \hat{y}^i)$$

$$\frac{\partial J}{\partial \theta_1} = -\frac{1}{m}\sum_{i=1}^{m}(y^i - \hat{y}^i)\,\hat{y}^i(1-\hat{y}^i)\,x_1$$

Using Gradient Descent to Minimize the cost

$$\frac{\partial J}{\partial \theta_1} = -\frac{1}{m}\sum_{i=1}^{m}(y^i - \hat{y}^i)\,\hat{y}^i(1-\hat{y}^i)\,x_1$$

$$\nabla J = \begin{bmatrix} \frac{\partial J}{\partial \theta_1} \\ \frac{\partial J}{\partial \theta_2} \\ \frac{\partial J}{\partial \theta_3} \\ \cdots \\ \cdots \\ \frac{\partial J}{\partial \theta_k} \end{bmatrix}$$

$$New\theta = old\theta - \nabla J$$

$$New\theta = old\theta - \eta \nabla J$$

How Fast we move?
Learning rate

Training using Gradient Descent

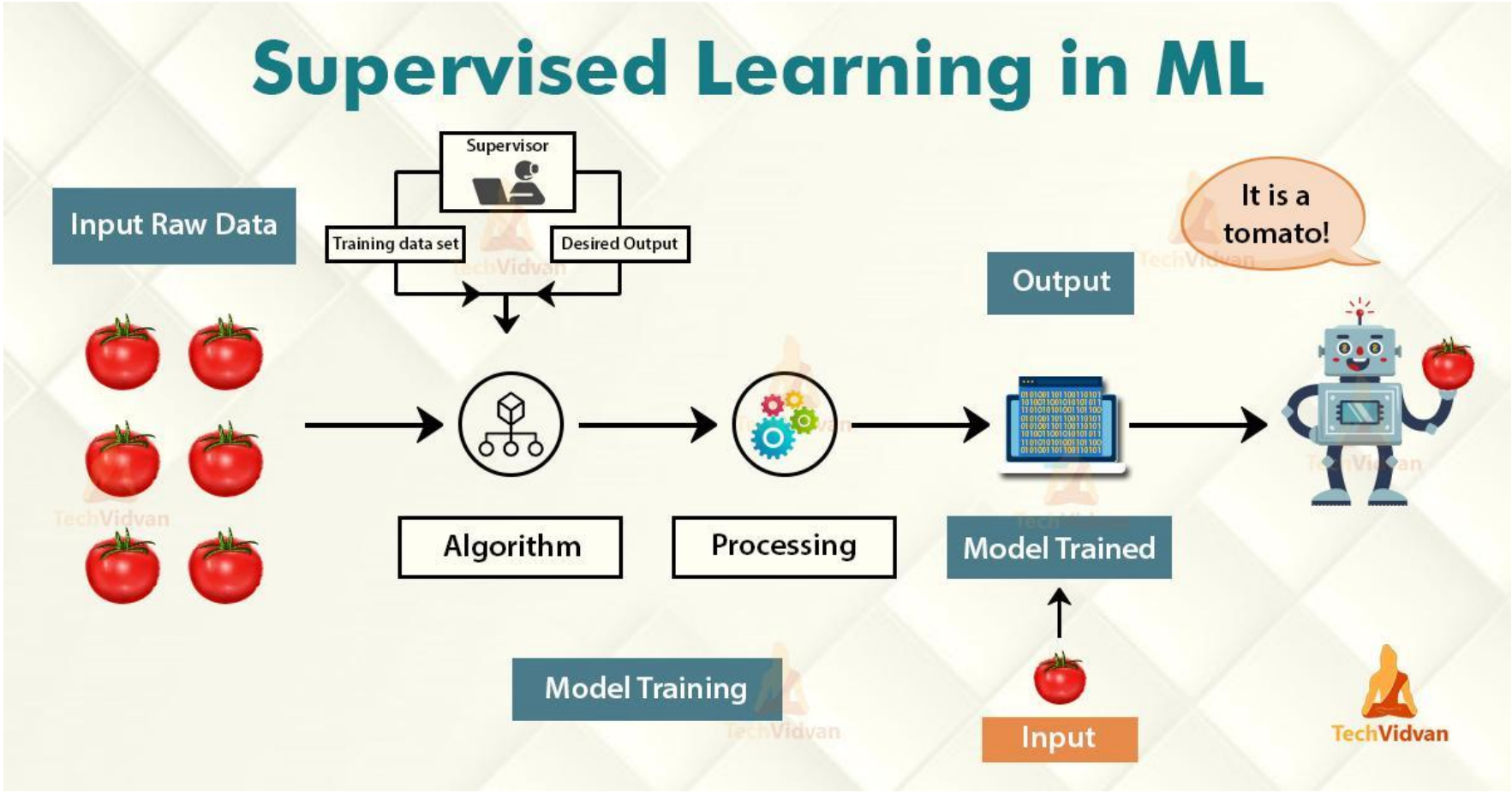
1. initialize the parameters randomly. $\theta^T = [\theta_0, \theta_1, \theta_2, \ldots]$
2. Feed the cost function with training set, and calculate the error.

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} y^i \log(\hat{y}^i) + (1 - y^i)\log(1 - \hat{y}^i)$$

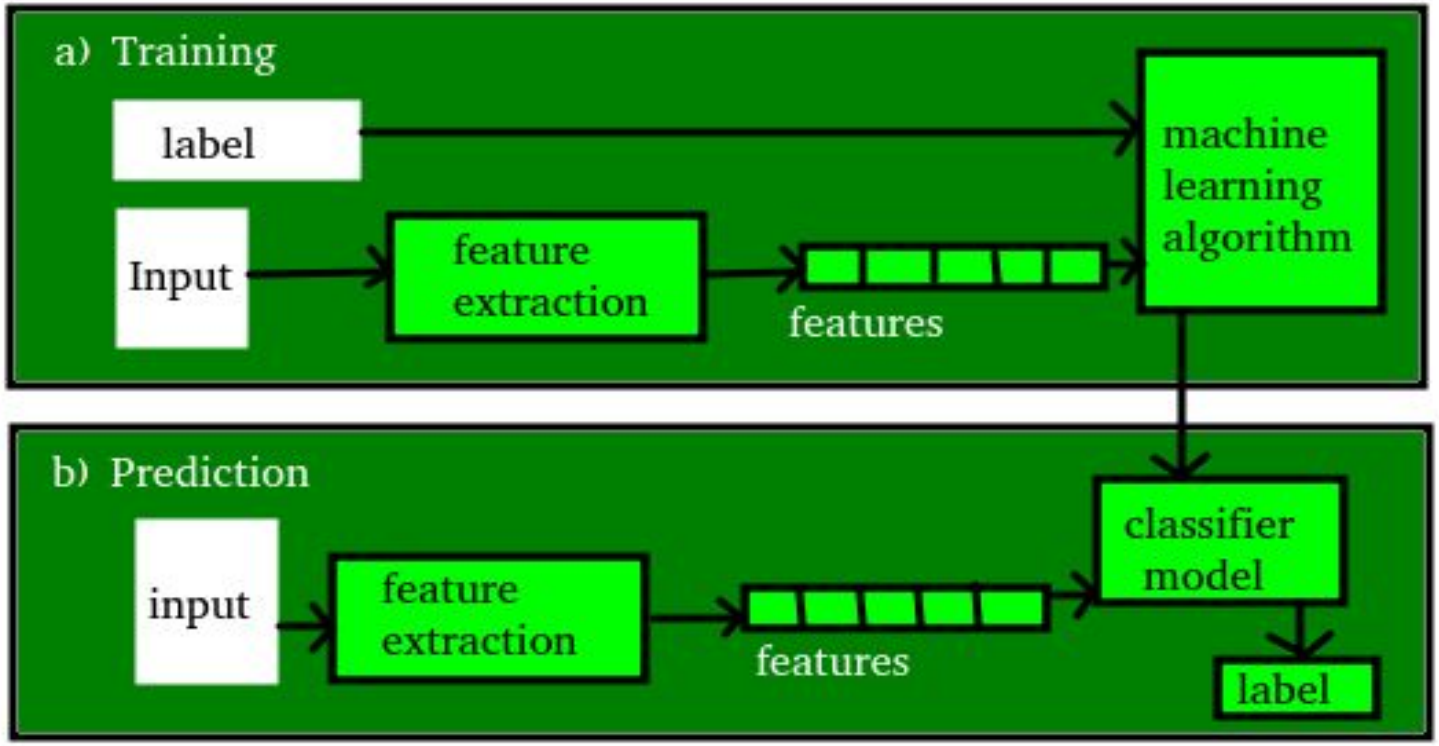
3. Calculate the gradient of cost function. $\nabla J = [\frac{\partial J}{\partial \theta_1}, \frac{\partial J}{\partial \theta_2}, \frac{\partial J}{\partial \theta_3}, \cdots, \frac{\partial J}{\partial \theta_k}]$
4. Update weights with new values. $\theta_{new} = \theta_{prv} - \eta \nabla J$
5. Go to step 2 until cost is small enough.
6. Predict the new customer X.

$$P(y=1|x) = \sigma(\theta^T X)$$

Supervised Learning



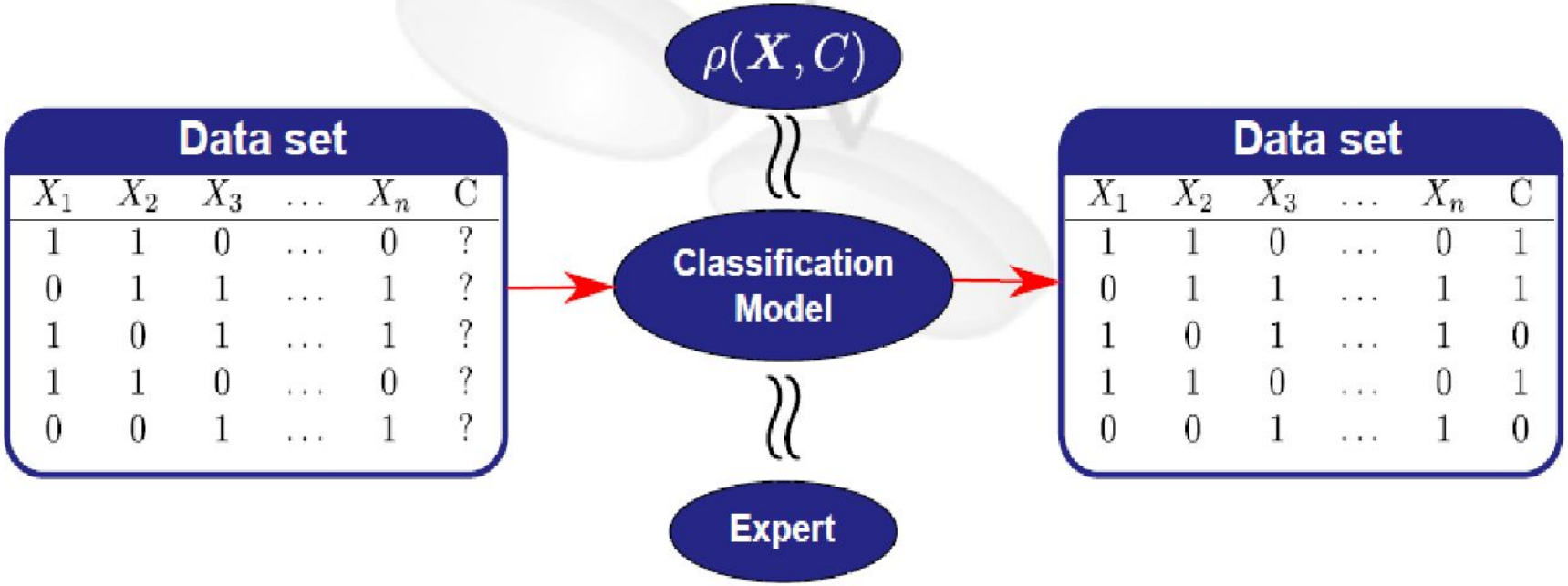
Supervised Machine Learning



Supervised Learning

Classification Model

- Classifier labels new data (unknown class value)



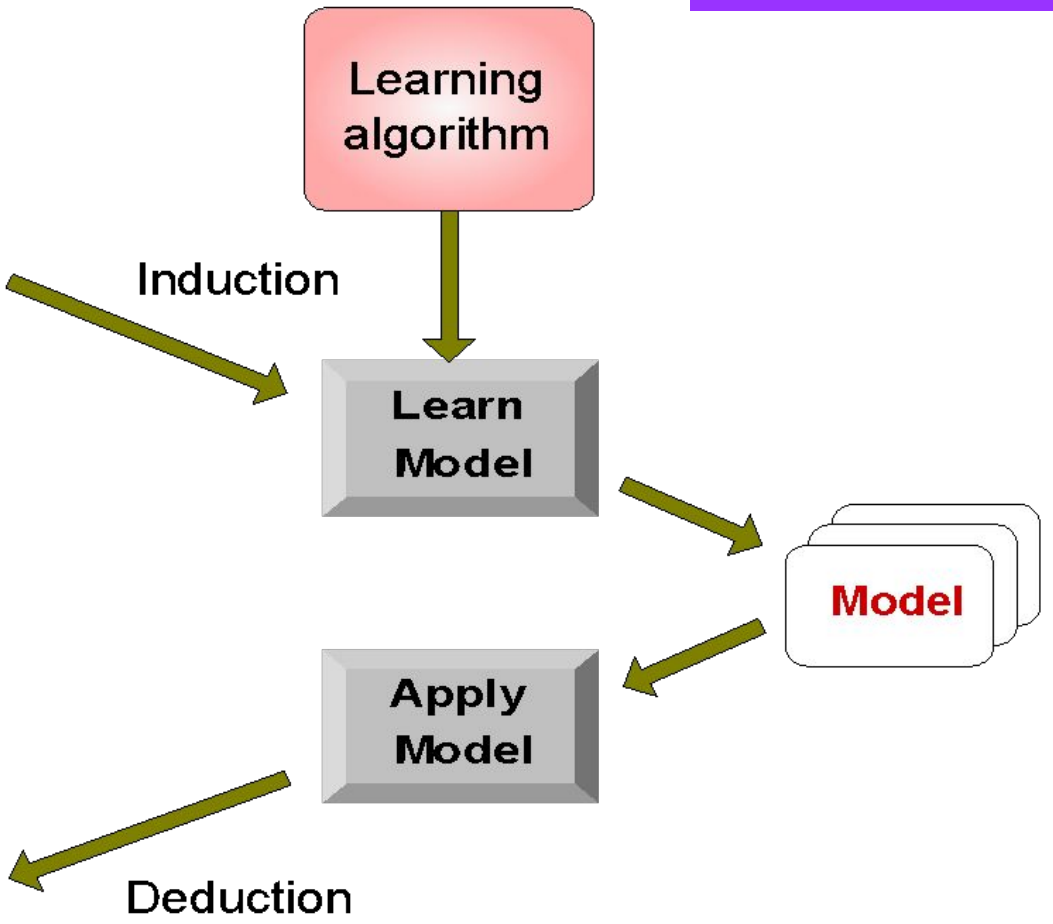
Illustrating Classification Task

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Classification: 3 Step Process

- **1. Model construction (Learning):**
  - Each record (instance) is assumed to belong to a predefined class, as determined by one of the attributes, called the class label
  - The set of all records used for construction of the model is called training set
  - The model is usually represented in the form of classification rules, (IF-THEN statements) or decision trees
- 2. **Model Evaluation (Accuracy):**
  - Estimate accuracy rate of the model based on a test set
  - The known label of test sample is compared with the classified result from model
  - Accuracy rate: percentage of test set samples correctly classified by the model
  - Test set is independent of training set
- **3. Model Use (Classification):**
  - The model is used to classify unseen instances (assigning class labels)
  - Predict the value of an actual attribute

What is Learning ?

- Data: Loan application data
- Task: Predict whether a loan should be approved or not.
- Performance measure: Accuracy.

Accuracy = 7/10 = 70%.

Classify all future applications (test data) to the majority class (i.e., No):

Tid	Refund	Marital Status	Taxable Income	Cheat Actual	Cheat predict
1	Yes	Single	125 Cr	No	No
2	No	Married	100 Cr	No	No
3	No	Single	70 Cr	No	No
4	Yes	Married	120 Cr	No	No
5	No	Divorced	95 Cr	Yes	No
6	No	Married	60 Cr	No	No
7	Yes	Divorced	220 Cr	No	No
8	No	Single	85 Cr	Yes	No
9	No	Married	75 Cr	No	No
10	No	Single	90 Cr	Yes	No

- We can do better than 70% with learning.

Classification Algorithms

Basic Principle (Inductive Learning Hypothesis):

Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples.

Typical Algorithms:

- **K Nearest Neighbour**
- **Neural networks**
- **Bayesian networks**

Eagar Learners

- Eager learners, when given a set of training tuples, will construct a generalization (i.e., classification) model before receiving new (e.g., test) tuples to classify.

- We can think of the learned model as being ready and eager to classify previously unseen tuples.

Eager Learners	Lazy Learners
• Do lot of work on training data	• Do less work on training data
• Do less work when test tuples are presented	• Do more work when test tuples are presented

Lazy Learners

- when given a training tuple, a lazy learner simply stores it (or does only a little minor processing) and waits until it is given a test tuple.
- Only when it sees the test tuple it performs generalization in order to classify the tuple based on its similarity to the stored training tuples.
- Unlike eager learning methods, lazy learners do less work when a training tuple is presented and more work when making a classification or prediction.
- Because lazy learners store the training tuples or "instances," they are also referred to as instance based learners.

Lazy Learners VS Eagar Learners

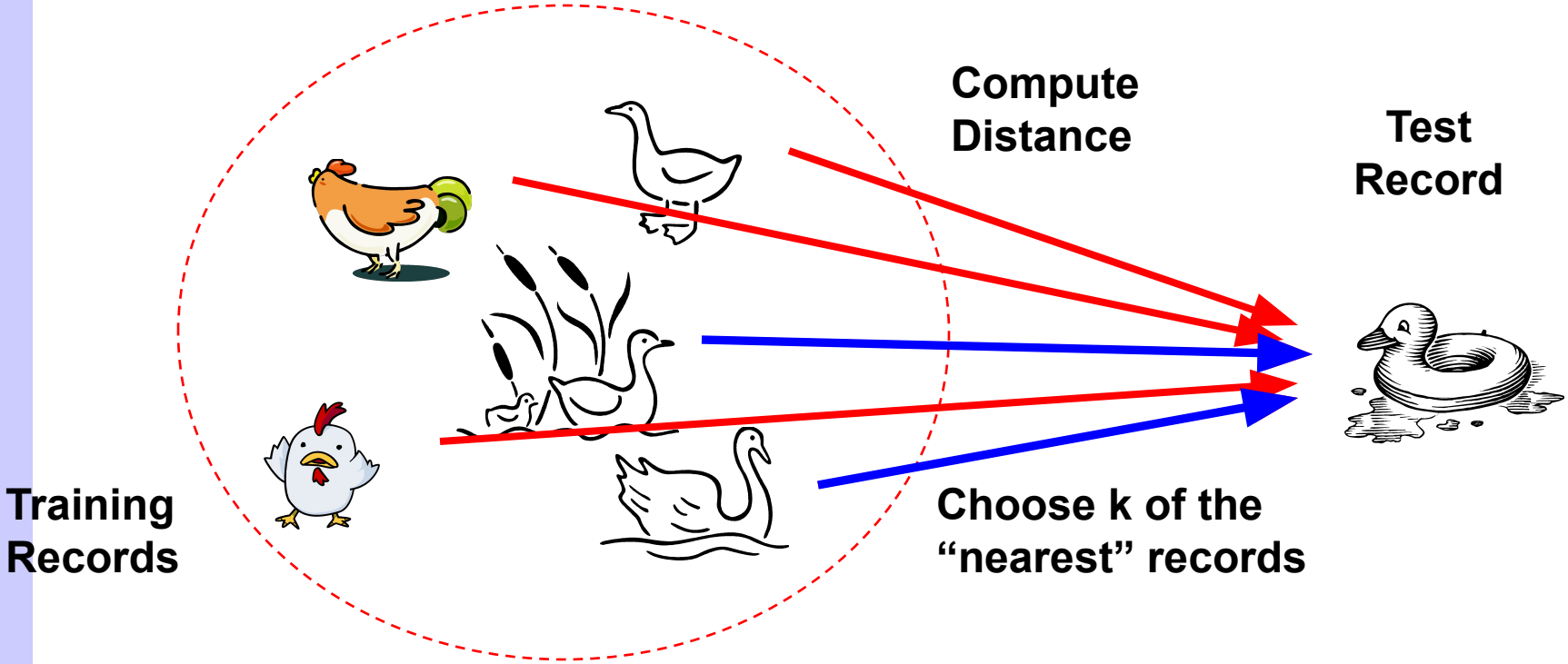
- When making a classification or prediction, lazy learners can be computationally expensive.
- Lazy Learners require efficient storage techniques and are well-suited to implementation on parallel hardware.
- Lazy Learners offer little explanation or insight into the structure of the data.
- Lazy learners, however, naturally support incremental learning.
- Lazy Learners are able to model complex decision spaces having hyperpolygonal shapes that may not be as easily describable by other learning algorithms (such as hyper-rectangular shapes modeled by decision trees).

KNN Basic Idea

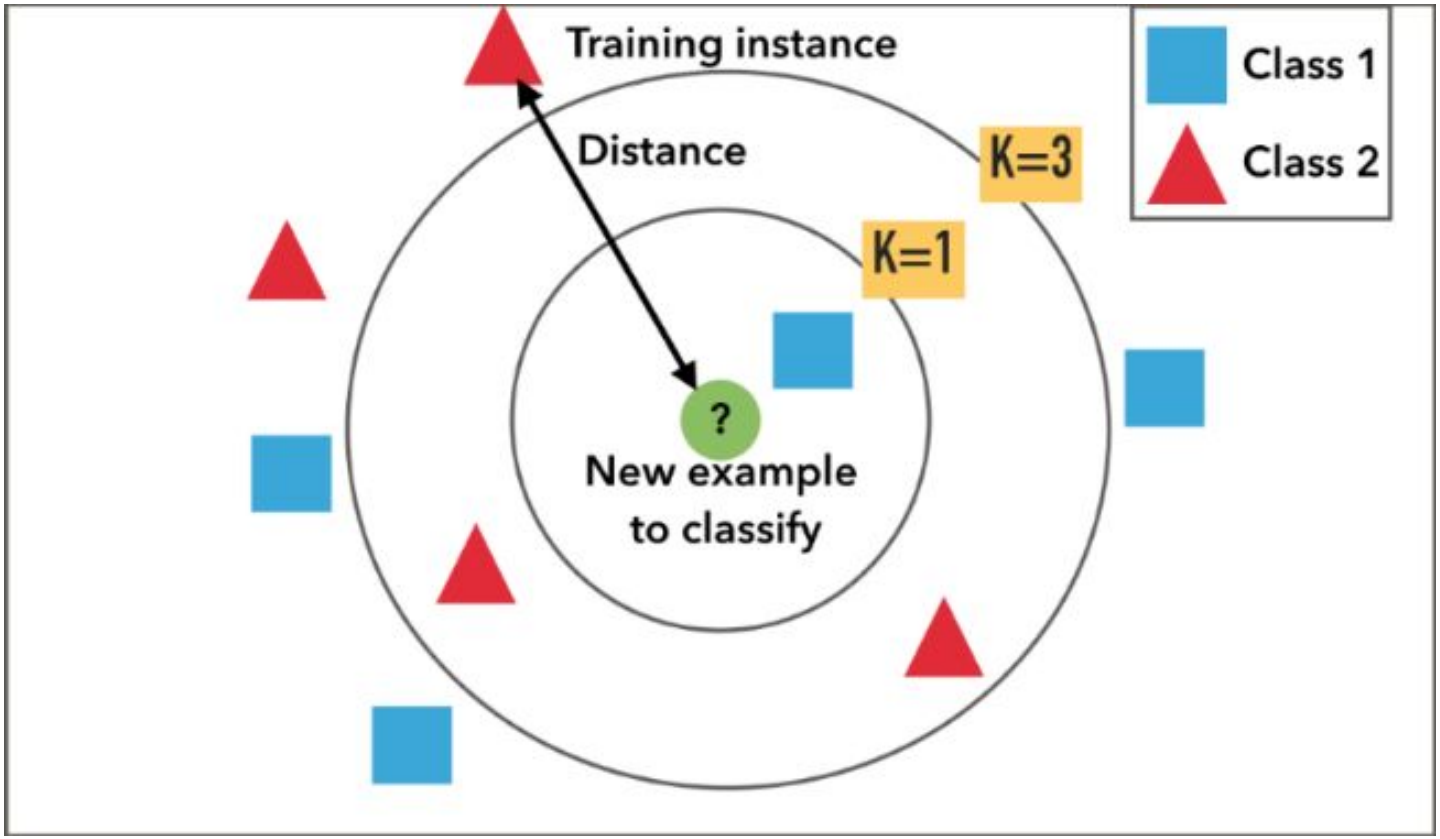
Tell me about your friends(*who your neighbors are*) and *I will tell you who you are*.

Nearest Neighbor Classifiers

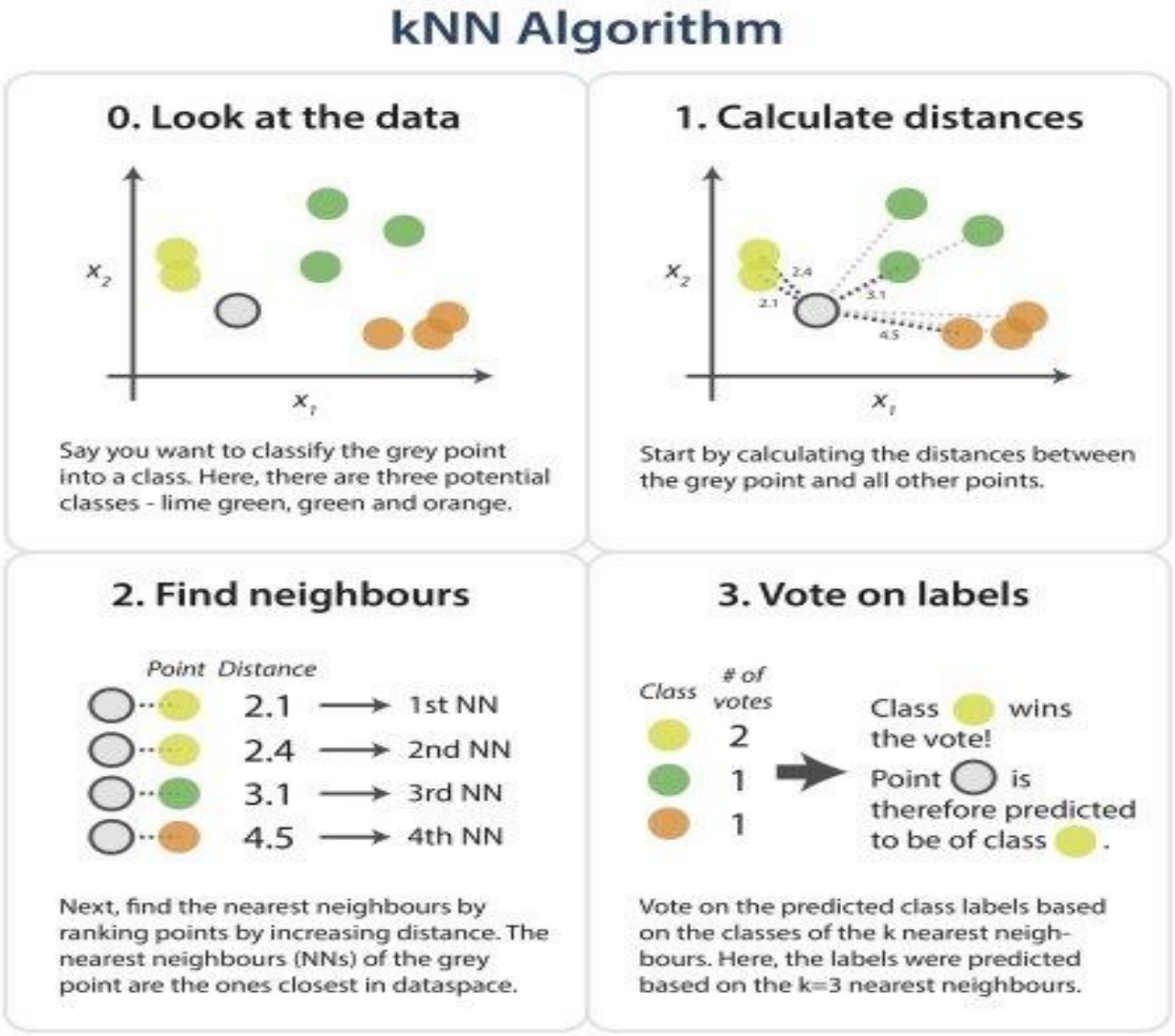
- Basic idea:
  - If it walks like a duck, quacks like a duck, then it's probably a duck



KNN Basic Idea



KNN Basic Idea



1. Look at Data

Customer	Age	Income	No. credit cards	Response
John	35	35K	3	No
Richa	22	50K	2	Yes
Harish	63	200K	1	No
Tom	59	170K	1	No
Neel	25	40K	4	Yes
David	37	50K	2	?

2. Calculate Distance

Distance functions

Euclidean

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Manhattan

$$\sum_{i=1}^{k}|x_i - y_i|$$

Minkowski

$$\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$$

2. Calculate Distance

- "Closeness" is defined in terms of a distance metric, such as Euclidean distance.

- The Euclidean distance between two points or tuples, say, $\boldsymbol{X}1 = (x11, x12, : : : , x1n)$ and $\boldsymbol{X}2 = (x21, x22, : : : , x2n)$, is

$$dist(\boldsymbol{X_1}, \boldsymbol{X_2}) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}.$$

2. Calculate Distance

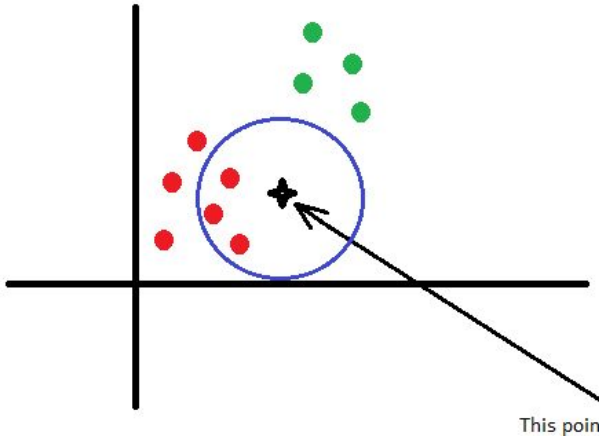
Customer	Age	Income	No. credit cards	Response
John	35	35K	3	No
Richa	22	50K	2	Yes
Harish	63	200K	1	No
Tom	59	170K	1	No
Neel	25	40K	4	Yes
David	37	50K	2	?

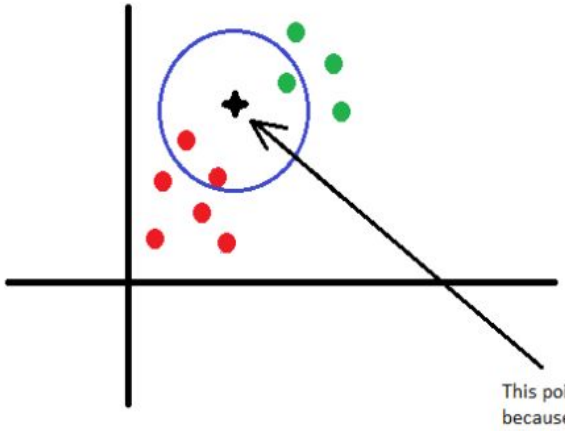
2. Calculate Distance

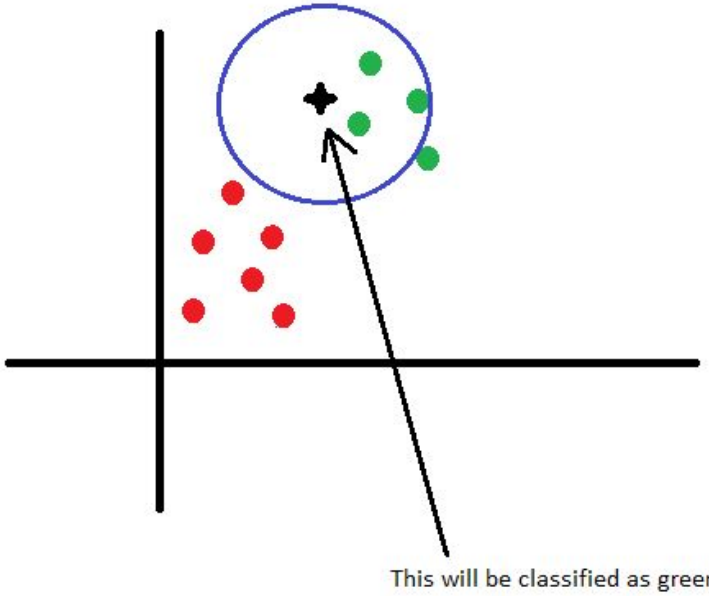
Example

Customer	Age	Income (K)	No. cards	Response	Distance from David
John	35	35	3	No	sqrt $[(35-37)^2+(35-50)^2+(3-2)^2]$=**15.16**
Richa	22	50	2	Yes	sqrt $[(22-37)^2+(50-50)^2+(2-2)^2]$=**15**
Harish	63	200	1	No	sqrt $[(63-37)^2+(200-50)^2+(1-2)^2]$=**152.23**
Tom	59	170	1	No	sqrt $[(59-37)^2+(170-50)^2+(1-2)^2]$=**122**
Neel	25	40	4	Yes	sqrt $[(25-37)^2+(40-50)^2+(4-2)^2]$=**15.74**
David	37	50	2		

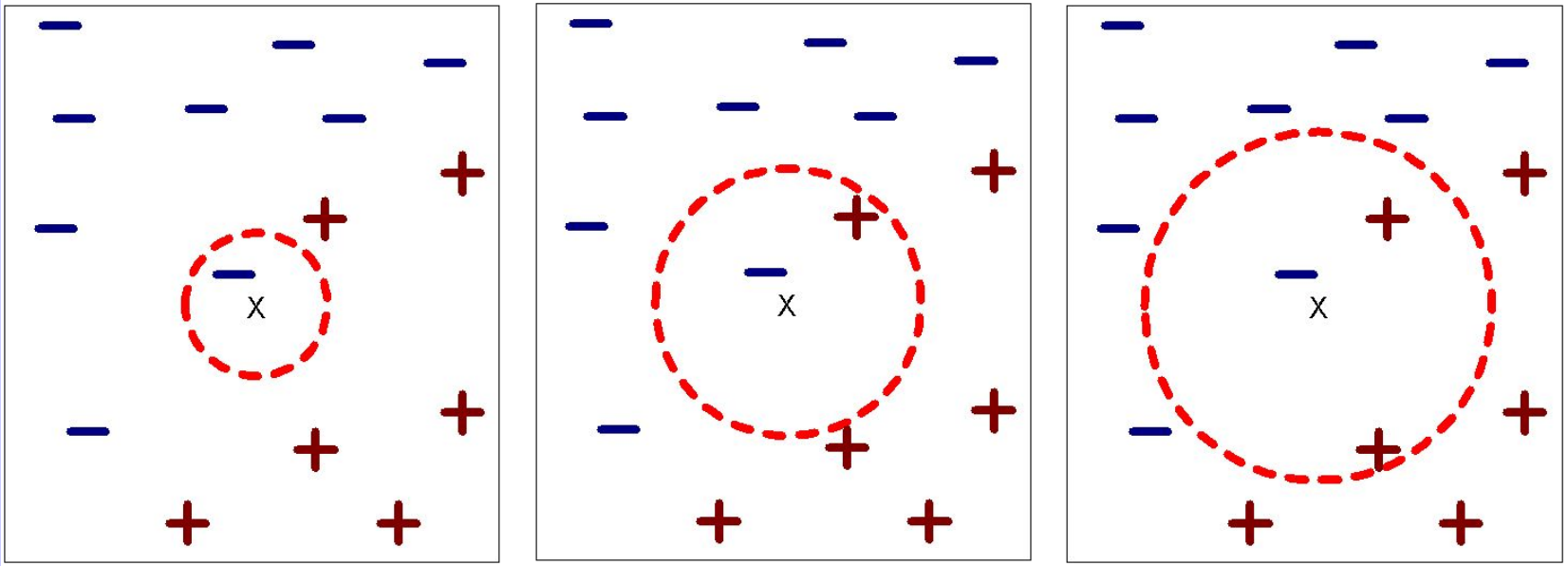
3. Find Neighbors (Value of K= ?)







Definition of Nearest Neighbor



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

What is K in KNN?

k = Number of nearest neighbor

If k=1, then test examples are given the same label as the closest example in the training set.

If k=3, the labels of the three closest classes are checked and the most common (i.e., occurring at least twice) label is assigned, and so on for larger k's.

3. Find Neighbors

Example

Customer	Age	Income (K)	No. cards	Response	Distance from David
John	35	35	3	No	sqrt $[(35-37)^2+(35-50)^2+(3-2)^2]$=**15.16**
Richa	22	50	2	Yes	sqrt $[(22-37)^2+(50-50)^2+(2-2)^2]$=**15**
Harish	63	200	1	No	sqrt $[(63-37)^2+(200-50)^2+(1-2)^2]$=**152.23**
Tom	59	170	1	No	sqrt $[(59-37)^2+(170-50)^2+(1-2)^2]$=**122**
Neel	25	40	4	Yes	sqrt $[(25-37)^2+(40-50)^2+(4-2)^2]$=**15.74**
David	37	50	2		

4. Vote on Labels

Example

Customer	Age	Income (K)	No. cards	Response	Distance from David
John	35	35	3	No	sqrt $[(35-37)^2+(35-50)^2+(3-2)^2]$=**15.16**
Richa	22	50	2	Yes	sqrt $[(22-37)^2+(50-50)^2+(2-2)^2]$=**15**
Harish	63	200	1	No	sqrt $[(63-37)^2+(200-50)^2+(1-2)^2]$=**152.23**
Tom	59	170	1	No	sqrt $[(59-37)^2+(170-50)^2+(1-2)^2]$=**122**
Neel	25	40	4	Yes	sqrt $[(25-37)^2+(40-50)^2+(4-2)^2]$=**15.74**
David	37	50	2	**Yes**	

Formal pseudocode

The steps of the KNN algorithm are (**formal pseudocode**):

1. Initialize ***selectedi = 0*** for all ***i*** data points from the **training set**

1. Select a **distance metric** (let's say we use Euclidean Distance)

1. For each training set data point ***i*** calculate the ***distancei*** = distance between the new data point and training point ***i.***

1. Choose the **K** parameter of the algorithm (**K = number of neighbors considered**), usually it's an odd number, this way avoiding ties in majority voting

1. For ***j = 1 to K*** loop through **all the training set data points** and in each step select the point **with minimum distance** to the new observation (minimum ***distancei***)

1. For **each existing class** count how many of the K selected data points are part of that class (**voting**)

1. Assign to the new observation the class with the **maximum** count (highest vote) — this is **majority voting**.

Algorithm

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.

Distance Function for Categorical Data

- *But how can distance be computed for attributes that not numeric, but categorical, such as color?"*

- For categorical attributes, a simple method is to compare the corresponding value of the attributes.

- If the two are identical (e.g., tuples ***X1*** and ***X2*** both have the color blue), then the difference between the two is taken as 0.

- If the two are different (e.g., tuple ***X1*** is blue but tuple ***X2*** is red), then the difference is considered to be 1.

Distance function for Categorical Data

Hamming Distance

$$D_H = \sum_{i=1}^{k} |x_i - y_i|$$

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

X	Y	Distance
Male	Male	0
Male	Female	1

Distance functions

- $D_{sum}(A,B) = D_{gender}(A,B) + D_{age}(A,B) + D_{salary}(A,B)$
- $D_{euclid}(A,B) = sqrt(D_{gender}(A,B)^2 + D_{age}(A,B)^2 + D_{salary}(A,B)^2$
- $D_{norm}(A,B) = D_{sum}(A,B)/max(D_{sum})$
- $A_{norm} = V(A)/max(A)$

Social Networks Ads

we will use the Social_Networks_Ads.csv file which contains information about the users like Gender, Age, Salary.

The **Purchased column** contains the **labels** for the users. This is a **binary classification** (we have two classes).

If the **label is 1** it means that the user **has bought product X** and **0** means the users **hasn't bought** that specific product.

User ID	Gender	Age	EstimatedSalary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	0
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	0
15728773	Male	27	58000	0
15598044	Female	27	84000	0
15694829	Female	32	150000	1
15600575	Male	25	33000	0
15727311	Female	35	65000	0
15570769	Female	26	80000	0
15606274	Female	26	52000	0
15746139	Male	20	86000	0
15704987	Male	32	18000	0

Import Dataset

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd


# Importing the dataset
dataset = pd.read_csv('Social_Network_Ads.csv')
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, 4].values
```

The pandas library contains the **read_csv** method which reads our data and saves it in a data structure called DataFrame.

Most of the algorithms from the **sklearn** library **requires** that the **attributes and the labels are in separate variables**, so we have to parse our data.

Split data in two different chunks

```
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.25, random_state = 0)
```

train_test_split, which will split our data set returning 4 values:

train attributes (X_train),
test attributes (X_test),
train labels (y_train)
the test labels (y_test).

A usual setup is to use **25% of the data set for test and 75% for train**.

Scaling / Normalization

- You can observe that the values from the Salary column are much higher than in the Age column.
- This can be a problem, because the impact of the Salary column will be much higher.
- Just think about it, if you have two very close salaries like 10000 and 9000, calculating the distance between them will result in 10000–9000 = 1000.
- Now if you take the Age column with values like 10 and 9, the difference is 10–9=1, which has lower impact

User ID	Gender	Age	EstimatedSalary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	0
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	0
15728773	Male	27	58000	0
15598044	Female	27	84000	0
15694829	Female	32	150000	1
15600575	Male	25	33000	0
15727311	Female	35	65000	0
15570769	Female	26	80000	0
15606274	Female	26	52000	0
15746139	Male	20	86000	0
15704987	Male	32	18000	0

Scaling / Normalization

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Train the model:

```
# Fitting classifier to the Training set
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 2)
classifier.fit(X_train, y_train)
```

1. **n_neighbors**: the value of k, the number of neighbors considered
2. **weights**: if you want to use weighted attributes, here you can configure the weights. This takes values like uniform, distance (inverse distance to the new point) or callable which should be defined by the user. The default value is uniform.
3. **algorithm**: if you want a different representation of the data, here you can use values like ball_tree, kd_tree or brute, default is auto which tries to automatically select the best representation for the current data set.
4. **metric**: the distance metric (Euclidean, Manhattan, etc), default is Euclidean.

Predict the class:

```
# Predicting the Test set results
y_pred = classifier.predict(X_test)
```

```
# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

```
[43,  7]
[ 5,  8]
```

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd


# Importing the dataset
dataset = pd.read_csv('Social_Network_Ads.csv')
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, 4].values
```

```
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.25, random_state = 0)
```

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

```
# Fitting classifier to the Training set
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 2)
classifier.fit(X_train, y_train)
```

```
# Predicting the Test set results
y_pred = classifier.predict(X_test)
```

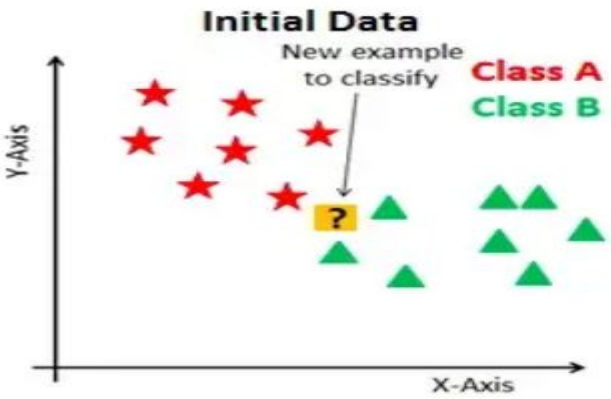
Applications

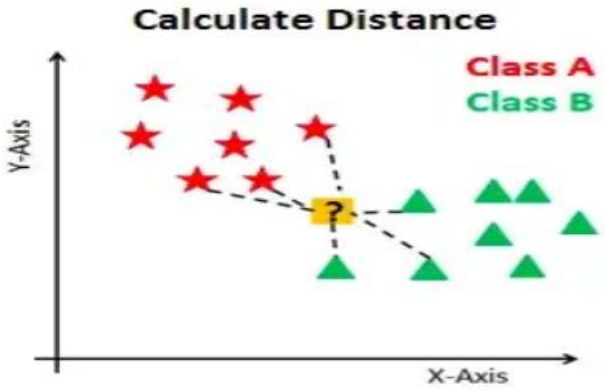
- KNN can be used for Recommendation Systems. Although in the real world, more sophisticated algorithms are used for the recommendation system. ...
- KNN can search for semantically similar documents. Each document is considered as a vector. ...
- KNN can be effectively used in detecting outliers.

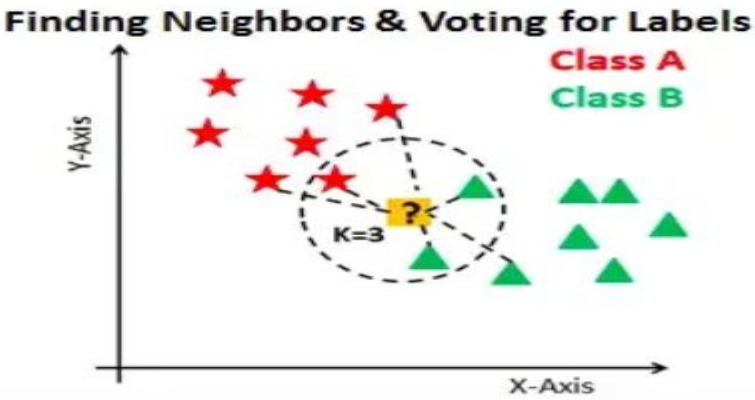
Steps for KNN

There are only 3 steps for KNN:

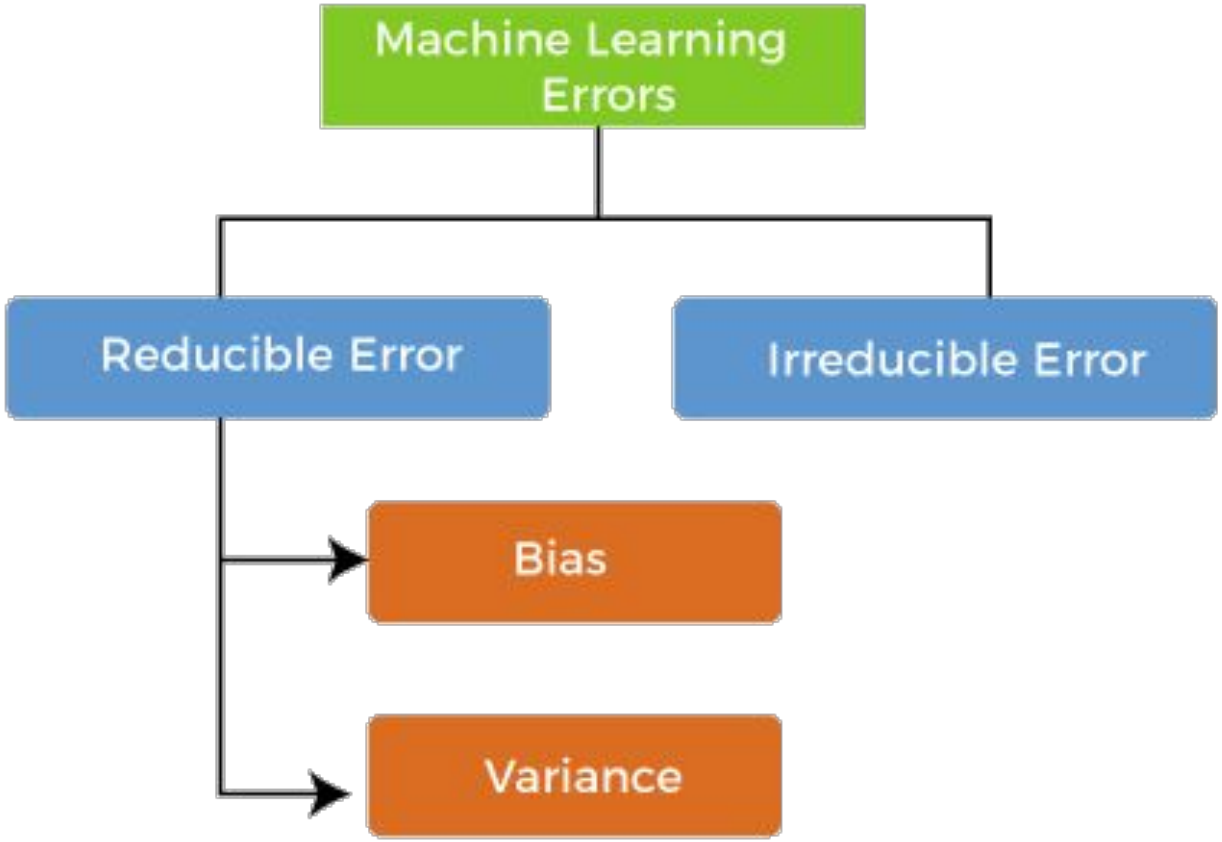
1. Calculate distance (e.g. Euclidean distance, Hamming distance, etc.)
2. Find k closest neighbors
3. Vote for labels or calculate the mean







Errors in Machine Learning?



Irreducible errors: These errors will always be present in the model

Reducible errors: These errors can be reduced to improve the model accuracy. Such errors can further be classified into bias and Variance.

Bias

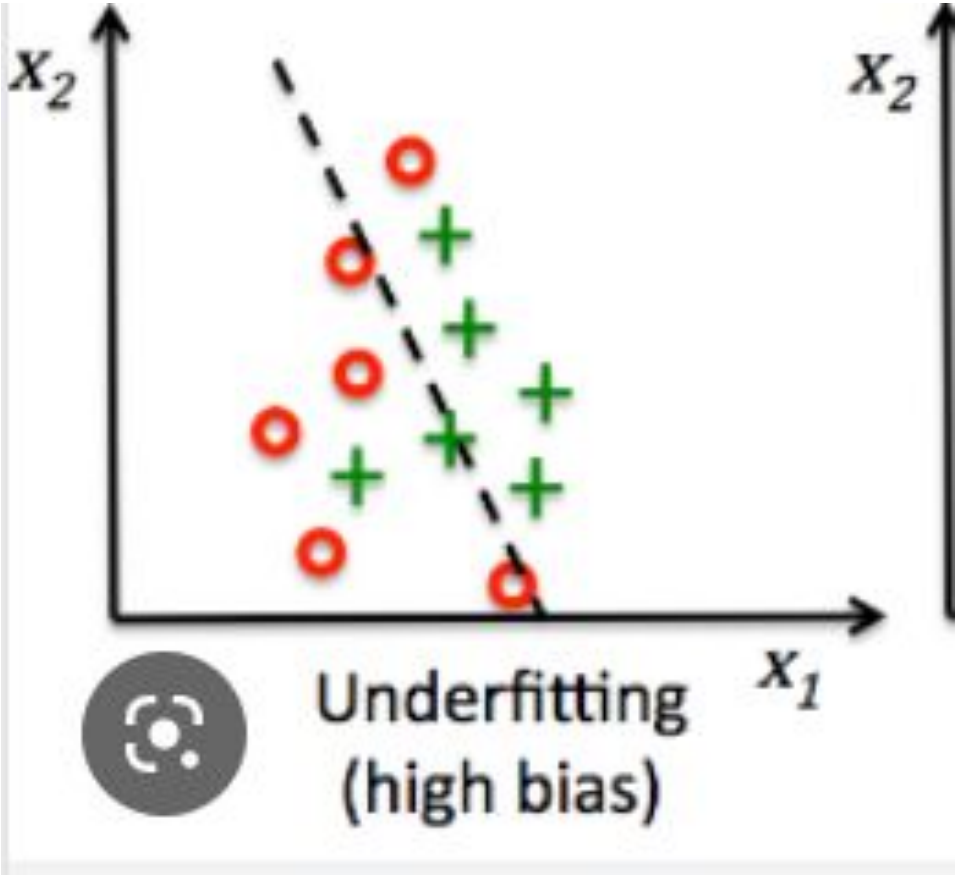
- *While making predictions, a difference occurs between prediction values made by the model and actual values/expected values, and this difference is known as bias errors or Errors due to bias.*
- Bias is the error between average model prediction and the ground truth. Moreover, it describes how well the model matches the training data set:

- A model with a higher bias would not match the data set closely.
- A low bias model will closely match the training data set.

Characteristics of a high bias model include:

(1) Failure to capture proper data trends
(2) Potential towards underfitting
(3) More generalized/overly simplified
(4) High error rate

High Bias



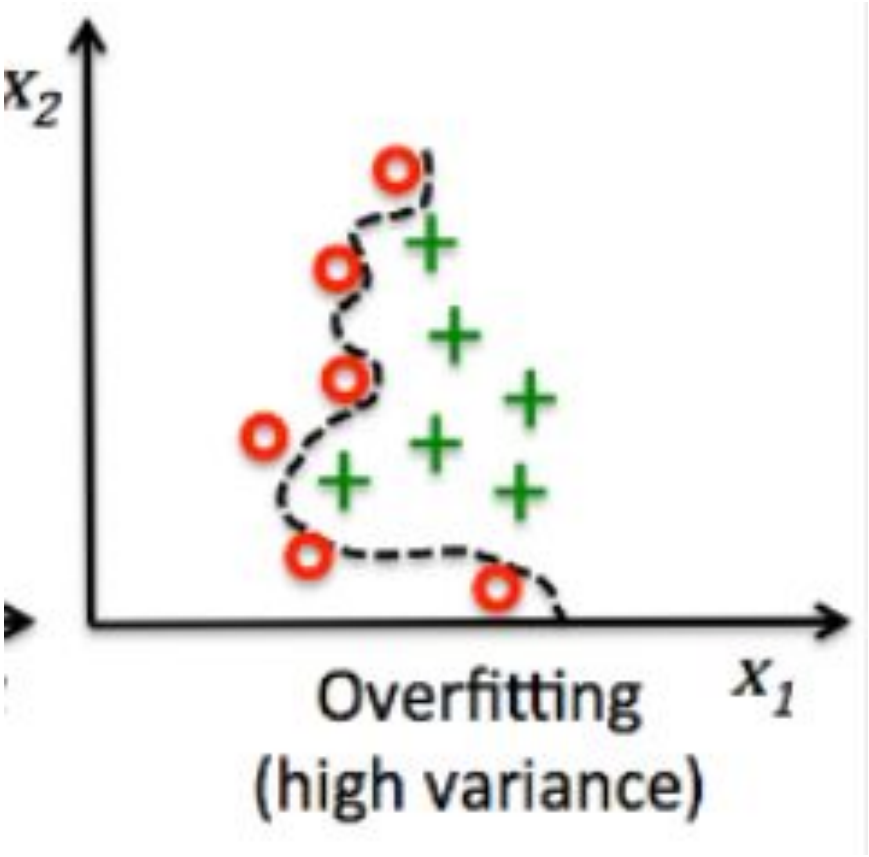
Variance

- The variance would specify the amount of variation in the prediction if the different training data was use.
- Variance is the variability in the model prediction—how much the ML function can adjust depending on the given data set.
- **Low variance** means there is a small variation in the prediction of the target function with changes in the training data set.
- **High variance** shows a large variation in the prediction of the target function with changes in the training dataset.

Characteristics of a high variance model include:

(1) Noise in the data set
(2) Potential towards overfitting
(3) Complex models
(4) Trying to put all data points as close as possible

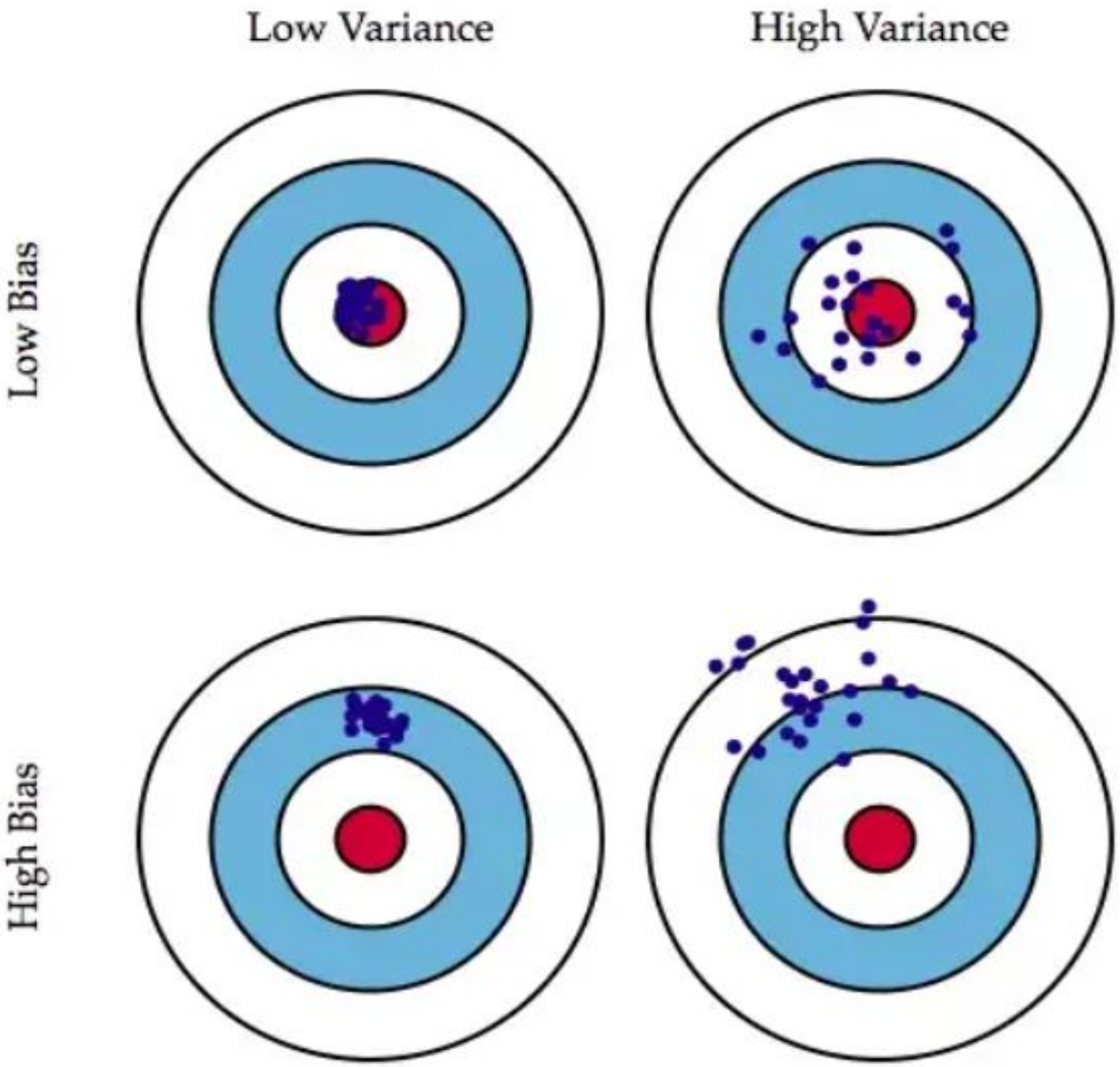
High Variance



Bias-Variance Tradeoff

- High bias can cause an algorithm to miss the relevant relations between features and target outputs.
- In other words, model with high bias pays very little attention to the training data and oversimplifies the model.
- High variance can cause an algorithm to model the random noise in the training data, rather than the intended outputs.
- In other words, model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn't seen before.

Bias and Variance



Overfitting and Underfitting

- There are other two terms related to bias and variance, **underfitting** and **overfitting**.
- Underfitting means the model does not fit, in other words, does not predict, the (training) data very well.
- Overfitting means that the model predict the (training) data too well. It is too good to be true. If the new data point comes in, the prediction may be wrong.
- Normally, underfitting implies _____ bias and _____ variance, and overfitting implies _____ bias but _____ variance.
- Models with _____ bias will have _____ variance.
- Models with _____ variance will have a _____ bias.

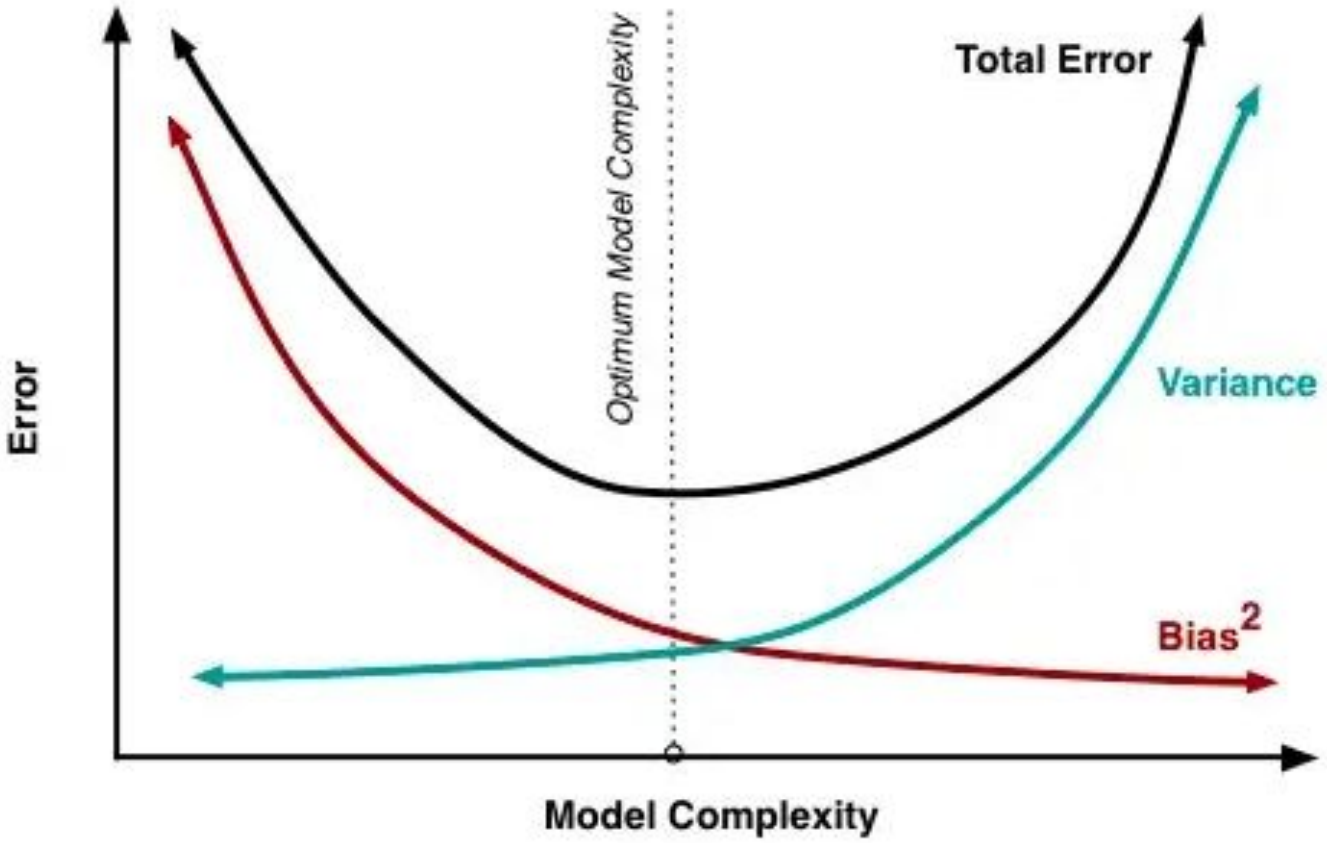
Overfitting and Underfitting

- Normally, underfitting implies high bias and low variance, and overfitting implies low bias but high variance.

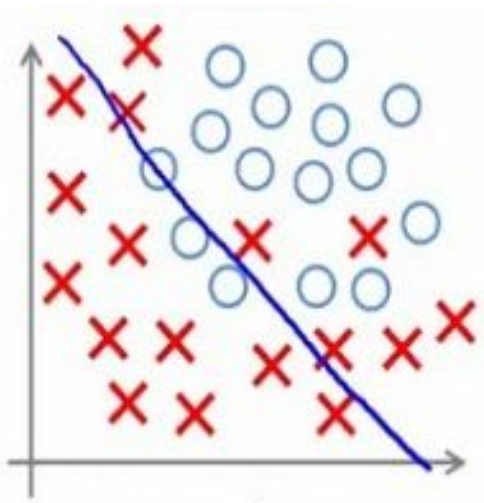
- Dealing with bias-variance problem is about dealing with over- and under-fitting. Bias is reduced and variance is increased in relation to model complexity.

- If the model is more complex, it means it has more power to capture the distribution of the data, which fits in the training set perfectly, in other word, overfitting.

Relationship between bias-variance & model complexity

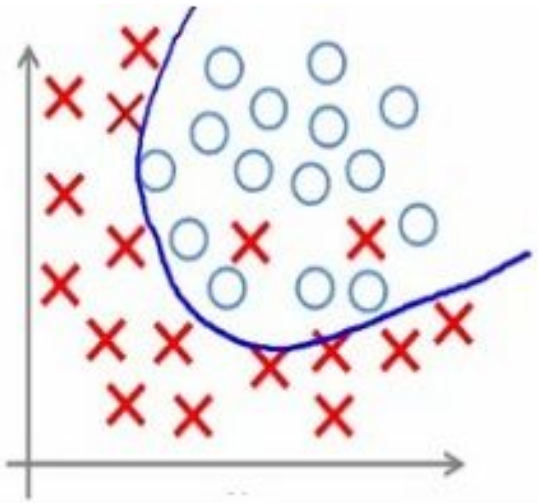


Overfitting and Underfitting

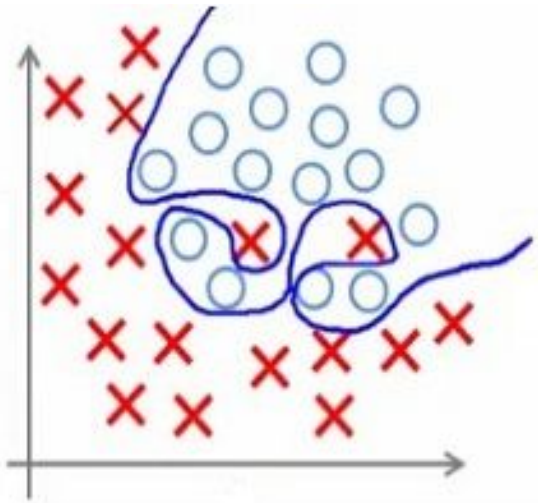


Under-fitting

(too simple to explain the variance)



Appropriate-fitting

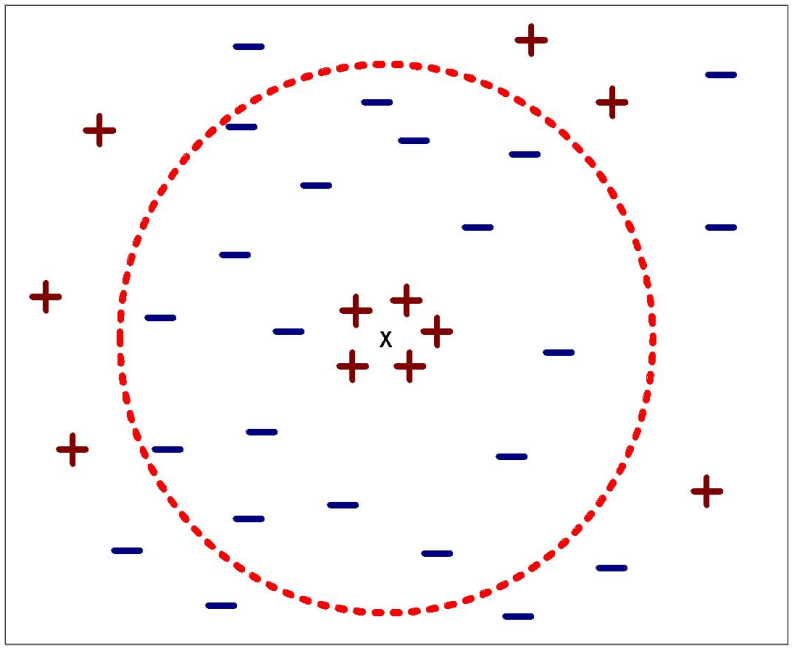


Over-fitting

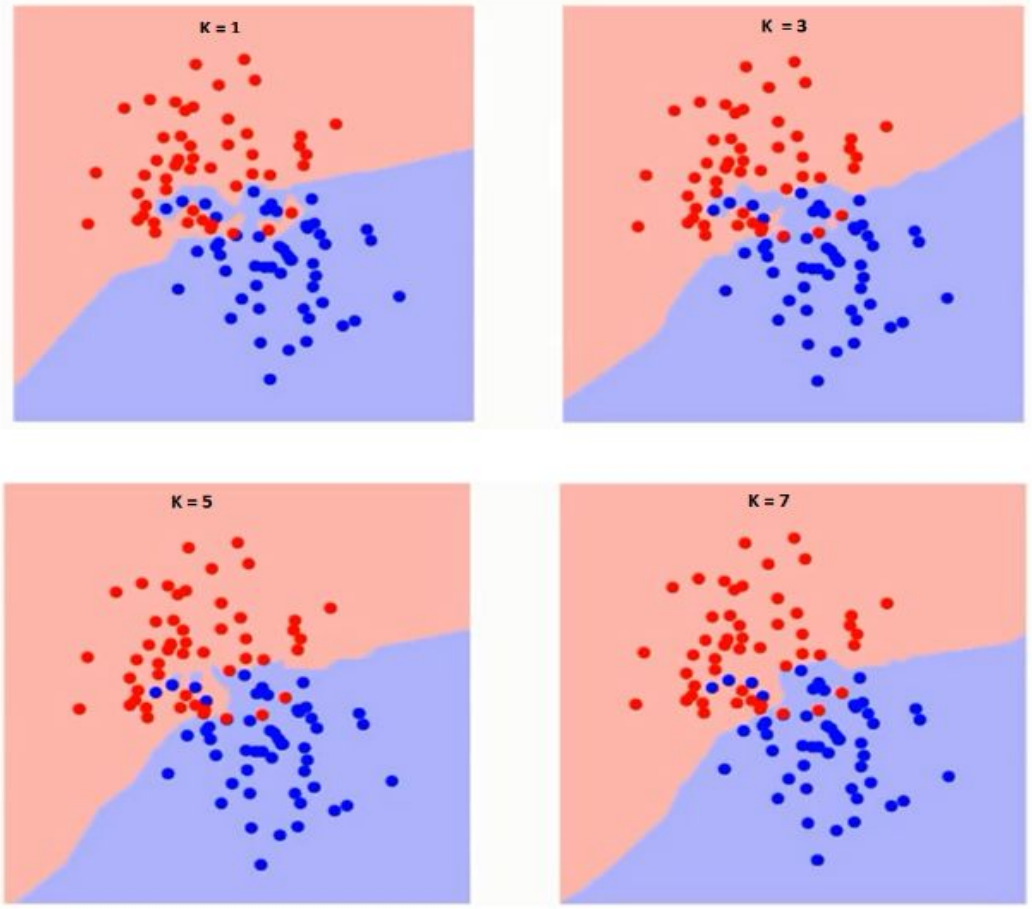
(forcefitting -- too good to be true)

Nearest Neighbor Classification…

- Choosing the value of k:
  - If k is too small, sensitive to noise points
  - If k is too large, neighborhood may include points from other classes

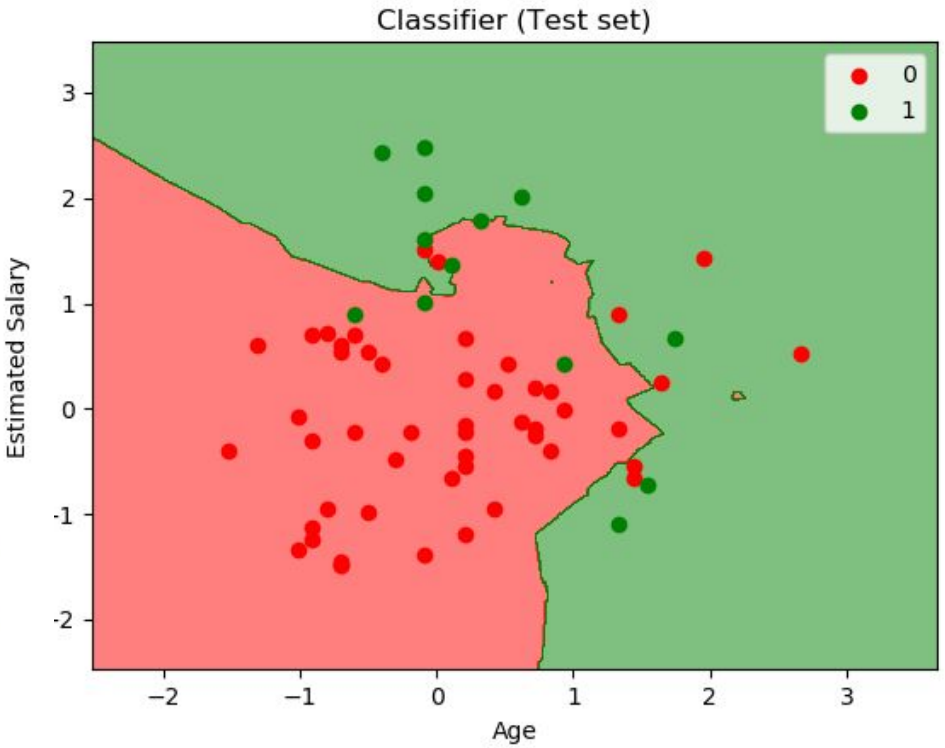


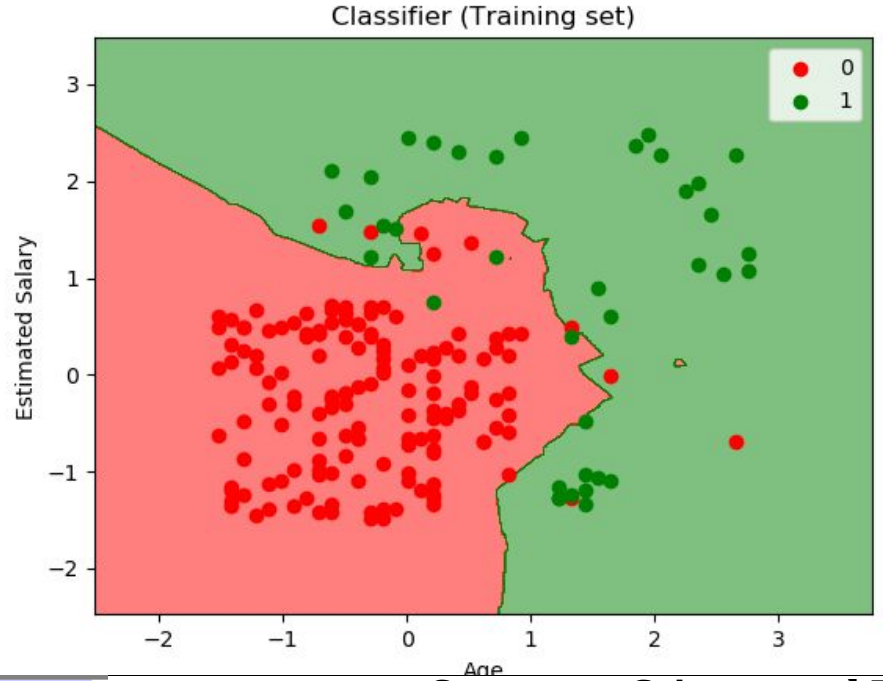
How do we choose the factor K?



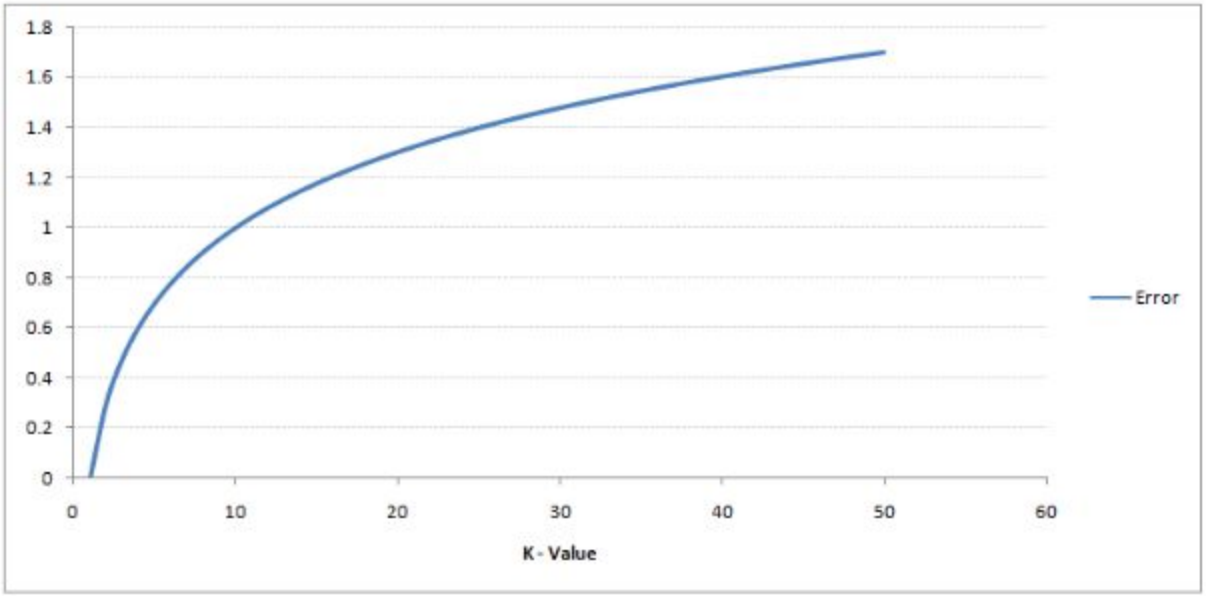
If you watch carefully, you can see that the boundary becomes smoother with increasing value of K. With K increasing to infinity it finally becomes all blue or all red depending on the total majority. The training error rate and the validation error rate are two parameters we need to access on different K-value.

Training Vs Testing



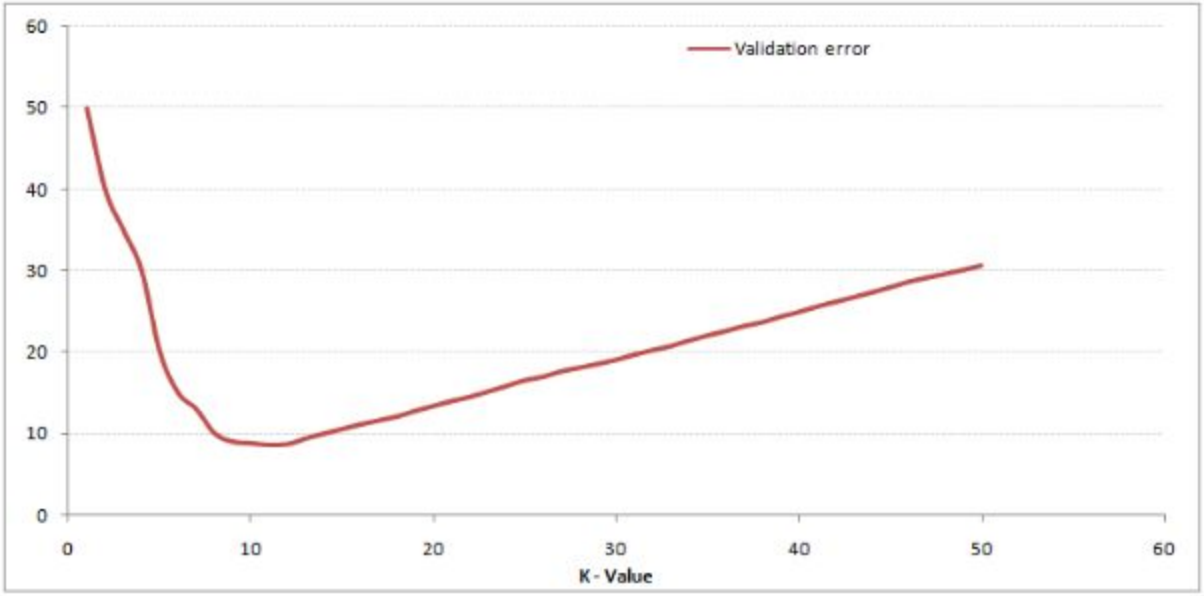


How do we choose the factor K?



- As you can see, the error rate at K=1 is always zero for the training sample. This is because the closest point to any training data point is itself. Hence the prediction is always accurate with K=1. If validation error curve would have been similar, our choice of K would have been 1.

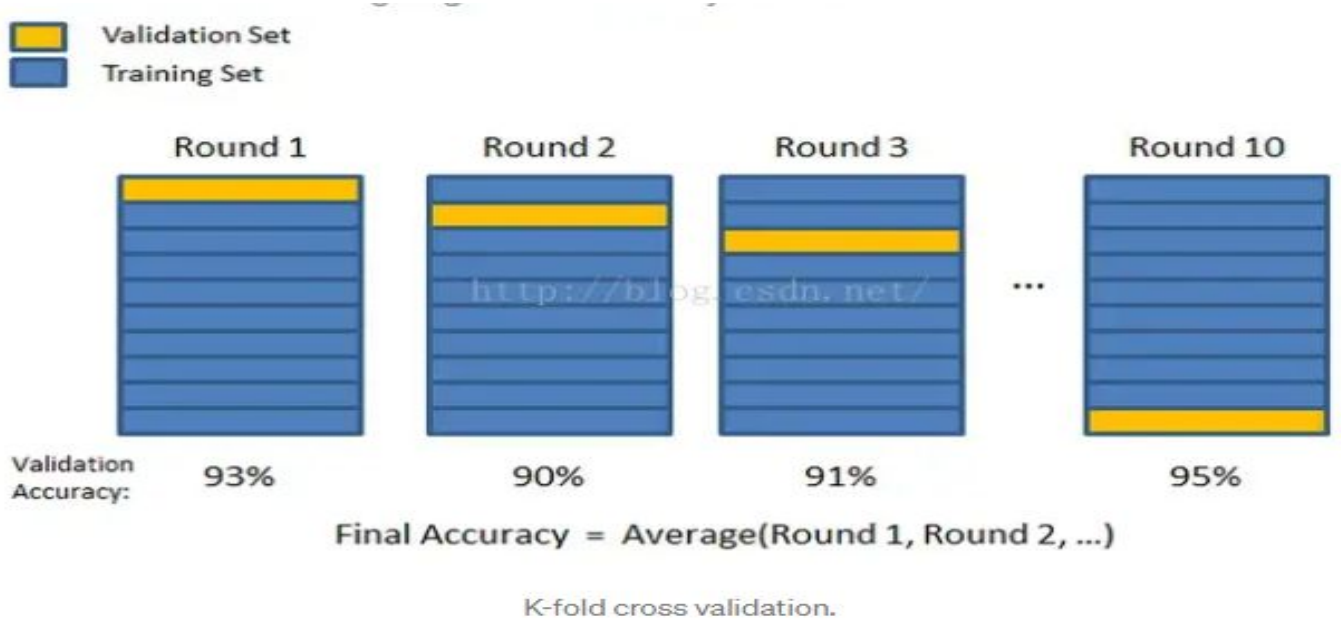
How do we choose the factor K?



- At K=1, we were overfitting the boundaries. Hence, error rate initially decreases and reaches a minima. After the minima point, it then increase with increasing K. To get the optimal value of K, you can segregate the training and validation from the initial dataset. Now plot the validation error curve to get the optimal value of K.

Cross-validation

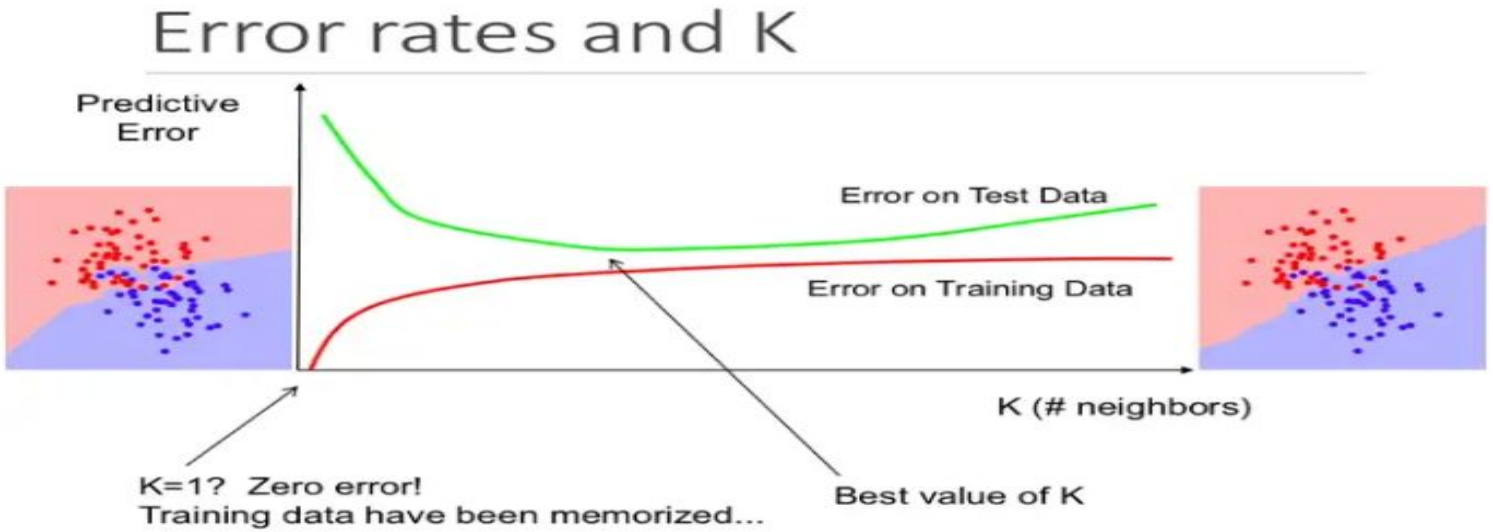
- Split the training set to A (80% of data) and B (20% of data).We then train our model based on A and test the model on B, since we know the ground truth of B now.
- B is called the **validation set**. We could use the validation set to tune our parameters, such as K in KNN.
- Rotate our validation set and use the rest of the data to train, say, split the data into K fold, and train on (K–1) folds and test on 1 fold as validation set, which is called K-fold cross validation.



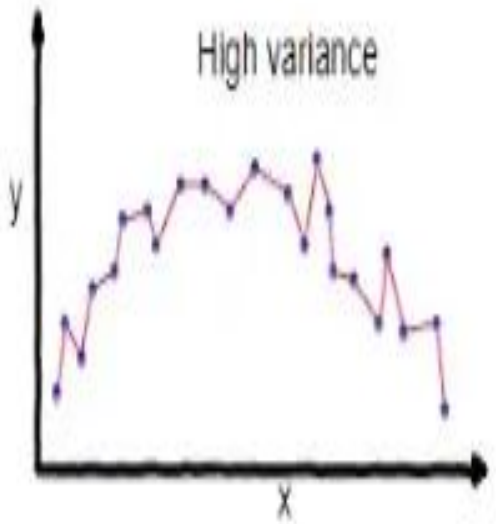
K-fold cross validation.

Error rate and K

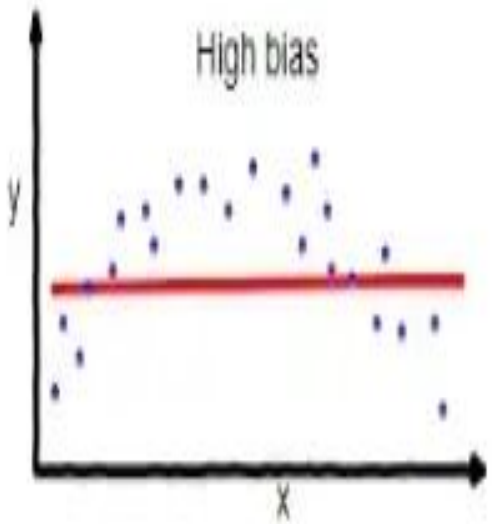
- The bias will be 0 when K=1, however, when it comes to new data (in test set), it has higher chance to be an error, which causes high variance.
- When we increase K, the training error will increase (increase bias), but the test error may decrease at the same time (decrease variance).
- when K becomes larger, since it has to consider more neighbors, its model is more complex.
- Now we can split the data into training and validation set and decide what K should be like.



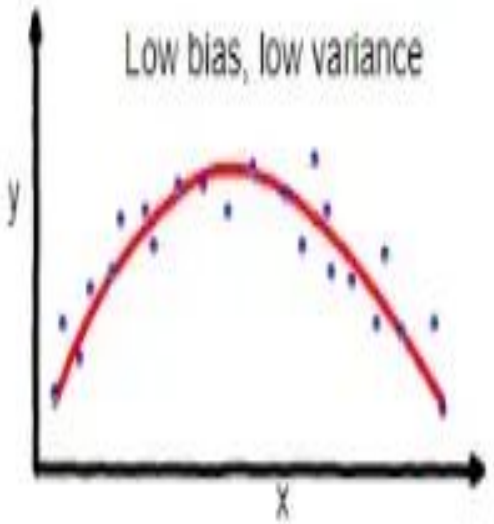
How to choose K?



overfitting



underfitting



Good balance

Example

Recnum	Gender	Age	Salary	Attriter
1	female	27	$19,000	no
2	male	51	$64,000	yes
3	male	52	$105,000	yes
4	female	33	$55,000	yes
5	male	45	$45,000	no
new	female	45	$100,000	?

	Neighbors	Neighbor Attrition	k = 1	k = 2	k = 3	k = 4	k = 5
d_{sum}							
d_{Euclid}							

	Neighbors	Neighbor Attrition	k = 1	k = 2	k = 3	k = 4	k = 5
d_{sum}	4,3,5,2,1	Y,Y,N,Y,N	yes	yes	yes	yes	yes
d_{Euclid}	4,1,5,2,3	Y,N,N,Y,Y	yes	?	no	?	yes

	k = 1	k = 2	k = 3	k = 4	k = 5
d_{sum}	yes, 100%	yes, 100%	yes, 67%	yes, 75%	yes, 60%
d_{Euclid}	yes, 100%	yes, 50%	no, 67%	yes, 50%	yes, 60%

Example

We have data from the questionnaires survey (to ask people opinion) and objective testing with two attributes (acid durability and strength) to classify whether a special paper tissue is good or not. Here is four training samples

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	Y = Classification
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

Now the factory produces a new paper tissue that pass laboratory test with X1 = 3 and X2 = 7. Without another expensive survey, can we guess what the classification of this new tissue is?

Example

- Suppose we have height, weight and T-shirt size of some customers and we need to predict the T-shirt size of a new customer given only height and weight information we have. Data including height, weight and T-shirt size information.

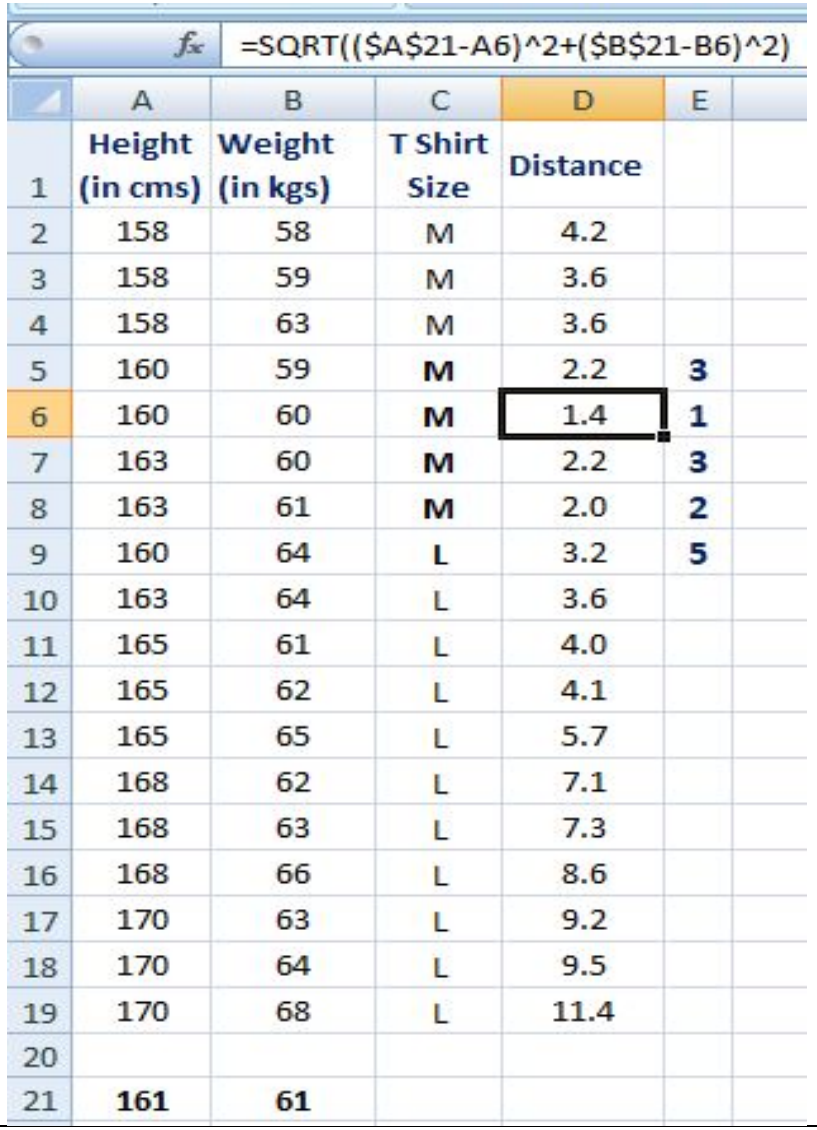
Height (in cms)	Weight (in kgs)	T Shirt Size
158	58	M
158	59	M
158	63	M
160	59	M
160	60	M
163	60	M
163	61	M
160	64	L
163	64	L
165	61	L
165	62	L
165	65	L
168	62	L
168	63	L
168	66	L
170	63	L
170	64	L
170	68	L

New customer named ‘Dhaval' has height 161 cm and weight 61kg.

Example

New customer named 'Dhaval' has height 161 cm and weight 61kg.

Height (in cms)	Weight (in kgs)	T Shirt Size
158	58	M
158	59	M
158	63	M
160	59	M
160	60	M
163	60	M
163	61	M
160	64	L
163	64	L
165	61	L
165	62	L
165	65	L
168	62	L
168	63	L
168	66	L
170	63	L
170	64	L
170	68	L



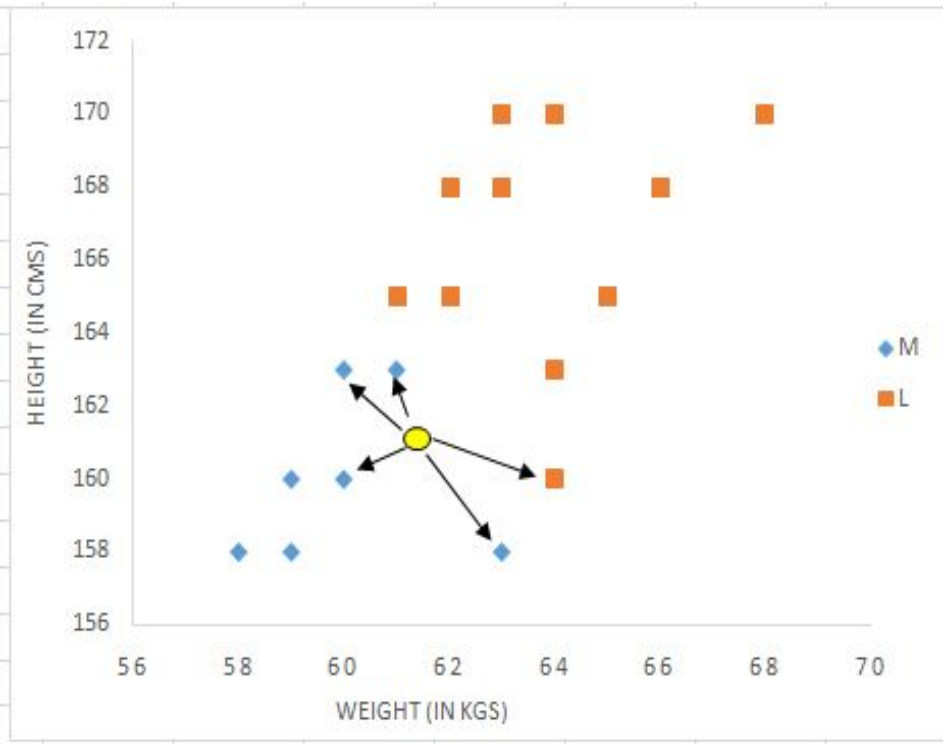
	A	B	C	D	E
1	Height (in cms)	Weight (in kgs)	T Shirt Size	Distance	
2	158	58	M	4.2	
3	158	59	M	3.6	
4	158	63	M	3.6	
5	160	59	M	2.2	3
6	160	60	M	1.4	1
7	163	60	M	2.2	3
8	163	61	M	2.0	2
9	160	64	L	3.2	5
10	163	64	L	3.6	
11	165	61	L	4.0	
12	165	62	L	4.1	
13	165	65	L	5.7	
14	168	62	L	7.1	
15	168	63	L	7.3	
16	168	66	L	8.6	
17	170	63	L	9.2	
18	170	64	L	9.5	
19	170	68	L	11.4	
20					
21	161	61			

Example

New customer named 'Dhaval' has height 161 cm and weight 61kg.

=SQRT((A21-A6)^2+(B21-B6)^2)

	A	B	C	D	E
1	Height (in cms)	Weight (in kgs)	T Shirt Size	Distance	
2	158	58	M	4.2	
3	158	59	M	3.6	
4	158	63	M	3.6	
5	160	59	M	2.2	3
6	160	60	M	1.4	1
7	163	60	M	2.2	3
8	163	61	M	2.0	2
9	160	64	L	3.2	5
10	163	64	L	3.6	
11	165	61	L	4.0	
12	165	62	L	4.1	
13	165	65	L	5.7	
14	168	62	L	7.1	
15	168	63	L	7.3	
16	168	66	L	8.6	
17	170	63	L	9.2	
18	170	64	L	9.5	
19	170	68	L	11.4	
20					
21	161	61			



Remarks on Lazy vs. Eager Learning

- Instance-based learning: lazy evaluation
- Decision-tree and Bayesian classification: eager evaluation
- Key differences
  - Lazy method may consider query instance *xq* when deciding how to generalize beyond the training data *D*
  - Eager method cannot since they have already chosen global approximation when seeing the query
- Efficiency: Lazy - less time training but more time predicting
- Accuracy
  - Lazy method effectively uses a richer hypothesis space since it uses many local linear functions to form its implicit global approximation to the target function
  - Eager: must commit to a single hypothesis that covers the entire instance space

Lazy vs. Eager Learning

Eagar	Lazy
Given a set of training set, constructs a classification model before receiving new (e.g., test) data to classify.	Simply stores training data (or only minor processing) and waits until it is given a test tuple.
Model Based	Instance Based
More time in training but less time in predicting	less time in training but more time in predicting
Must commit to a single hypothesis that covers the entire instance space	uses a richer hypothesis space since it uses many local linear functions to form its implicit global approximation to the target function
e.g. Naive Bayes Classifier	e.g. K-Nearest Neighbour

Classifier Output

- The type of information produced by the individual classifiers is shown below.

Classifier Outputs

Abstract Level

D_1	D_2	D_3	D_4	D_5	D_6
ω_1	ω_2	ω_1	ω_1	ω_3	ω_1

a classifier only outputs a unique label for each input pattern.

Measurement Level

	ω_1	ω_2	ω_3	ω_4
D_1	0.1	0.4	0.3	0.2
D_2	0.3	0.3	0.3	0.1
D_3	0.2	0.1	0.7	0.0

each classifier outputs class "confidence" levels for each input pattern.

The confidence of a classifier in a particular candidate class usually provides useful information that a simple Abstract Level cannot reflect.

Does patient have cancer or not?

- *A patient takes a lab test and the result comes back positive.*

  The test returns a correct positive result (+) in only 98% of the cases in which the disease is actually present.

  The test returns correct negative result (-) in only 97% of the cases in which the disease is not present.

Furthermore, 0.008 of the entire population have this cancer

Example

- Suppose data samples consists of fruits, described by their color and shape.
- Suppose that X is red and round, and that Y is the hypothesis that X is an apple.
- Then P(Y|X) reflects our confidence that X is an apple given that we have seen that X is red and round.

Bayes Theorem :

$$\Pr(Y \mid X) = \frac{\Pr(X \mid Y)\Pr(Y)}{\Pr(X)}$$

Naïve Bayes Classifier Algorithm

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems.
- It is mainly used in *text classification* that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- **It is a probabilistic classifier, which means it predicts on the basis of the probability of an object**.
- Some popular examples of Naïve Bayes Algorithm are **spam filtration, Sentimental analysis, and classifying articles**.

Why is it called Naïve Bayes?

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes,

Which can be described as:

- **Naïve**: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- **Bayes**: It is called Bayes because it depends on the principle of Bayes' Theorem

Bayes theorem

- Let X be a data sample whose class label is unknown.
- Let Y be some hypothesis, such as that the data sample X belongs to a specified class C.
- For classification problems, we want to determine P(Y | X), the probability that the hypothesis Y holds given the observed data sample X.
- P(Y | X) is the posterior probability, or a posteriori probability, of Y conditioned on X.

Bayes Theorem :

$$\Pr(Y \mid X) = \frac{\Pr(X \mid Y)\Pr(Y)}{\Pr(X)}$$

Naive Bayes Theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

The Bayes Classifier

- Use Bayes Rule!

$$P(Y|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y)P(Y)}{P(X_1,\ldots,X_n)}$$

Likelihood — Prior — Normalization Constant

- Why did this help? Well, we think that we might be able to specify how features are "generated" by the class label

The Naïve Bayes Model

- The *Naïve Bayes Assumption*: Assume that all features are independent given the class label Y
- Equationally speaking:

$$P(X_1, \ldots, X_n|Y) = \prod_{i=1}^{n} P(X_i|Y)$$

Building the Classifier

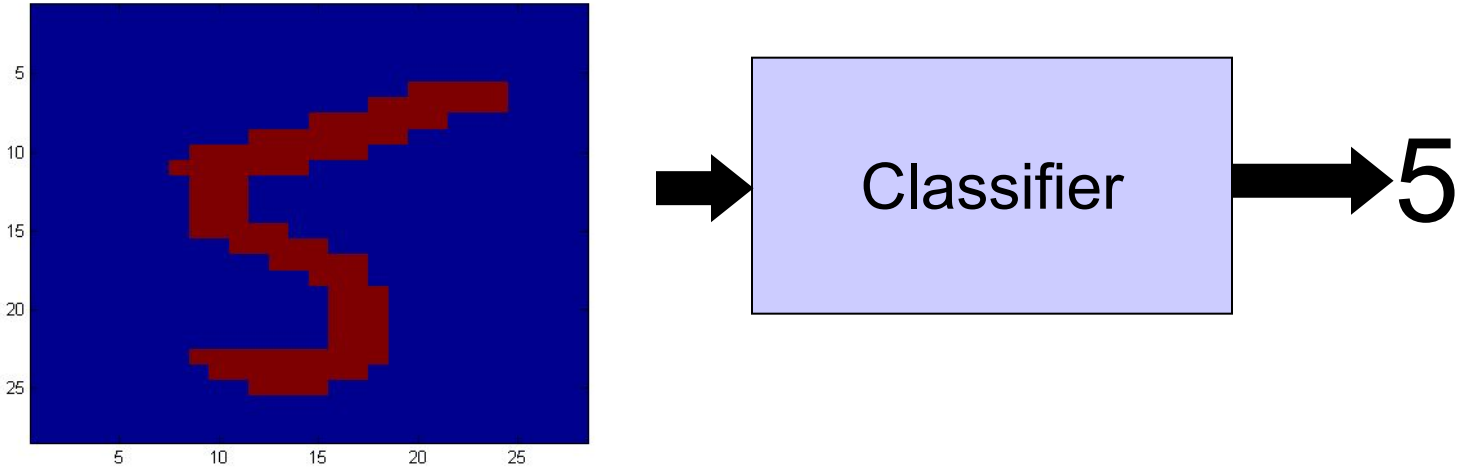
- Training phase:
  - Learning the posterior probabilities **Pr(Y|X)** for every combination of **X** and **Y** based on training data

- Test phase:
  - For test record **X'**, compute the class **Y'** that *maximizes the posterior probability*

    Pr(Y'|X')

Application

Digit Recognition



- $X_1,\ldots,X_n \in \{0,1\}$ (Black vs. White pixels)
- $Y \in \{5,6\}$ (predict whether a digit is a 5 or a 6)

The Bayes Classifier

- A good strategy is to predict:

$$\arg\max_Y P(Y|X_1,\ldots,X_n)$$

  - (for example: what is the probability that the image represents a 5 given its pixels?)

- So … How do we compute that?

The Bayes Classifier

- Let's expand this for our digit recognition task:

$$P(Y=5|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y=5)P(Y=5)}{P(X_1,\ldots,X_n|Y=5)P(Y=5)+P(X_1,\ldots,X_n|Y=6)P(Y=6)}$$

$$P(Y=6|X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n|Y=6)P(Y=6)}{P(X_1,\ldots,X_n|Y=5)P(Y=5)+P(X_1,\ldots,X_n|Y=6)P(Y=6)}$$

- To classify, we'll simply compute these two probabilities and predict based on which one is greater

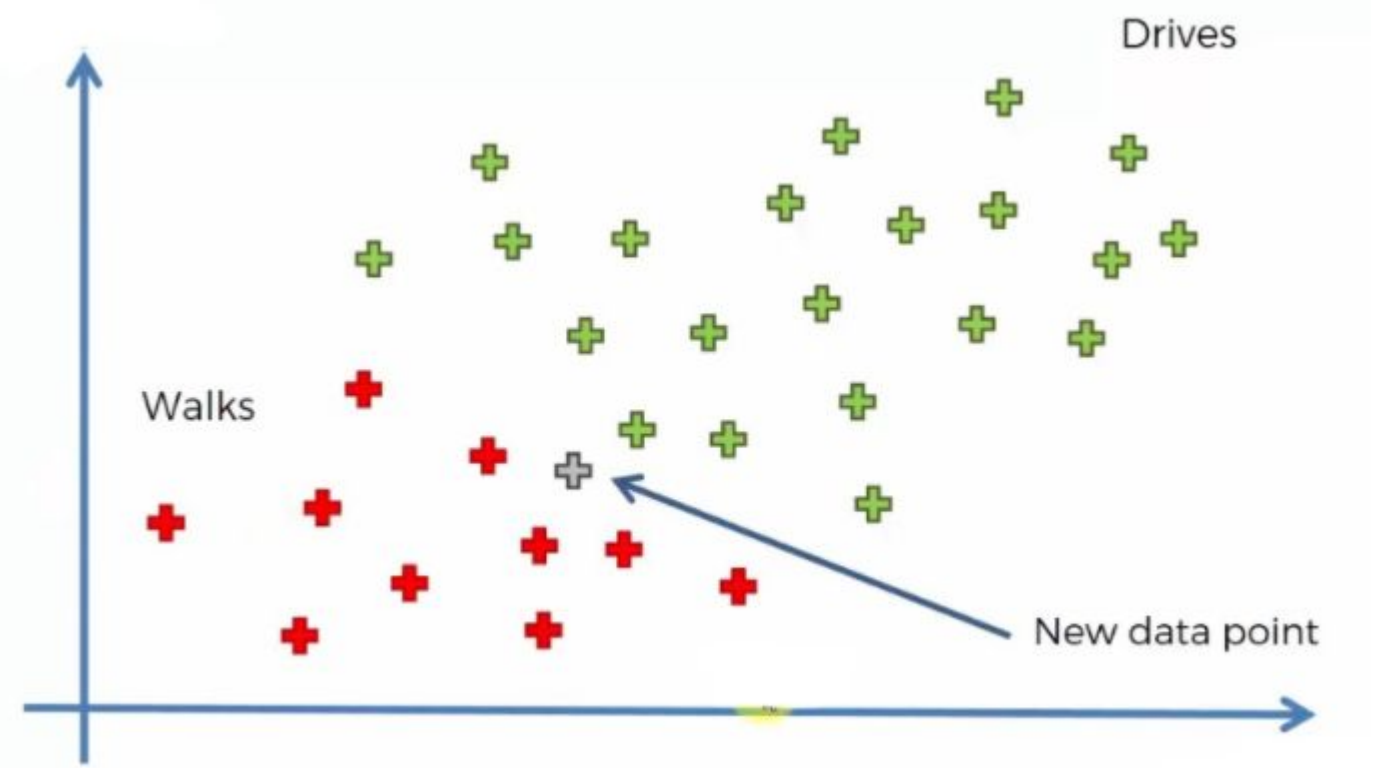
The Bayes Classifier

$$P(Y=5|X_1,\ldots,X_n) = P(X_1,\ldots,X_n|Y=5)P(Y=5)$$
$$P(Y=6|X_1,\ldots,X_n) = P(X_1,\ldots,X_n|Y=6)P(Y=6)$$

The naive Bayesian classifier assigns an unknown sample X to the class Ci if and only if :

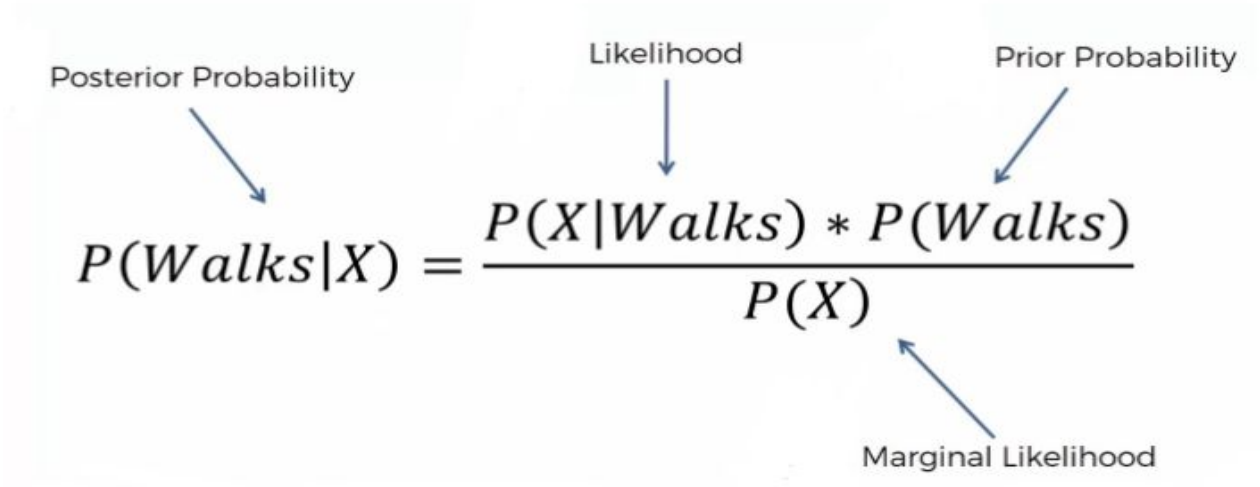
$$P(C_i|X) > P(C_j|X) \text{ for } 1 \leq j \leq m,\ j \neq i.$$

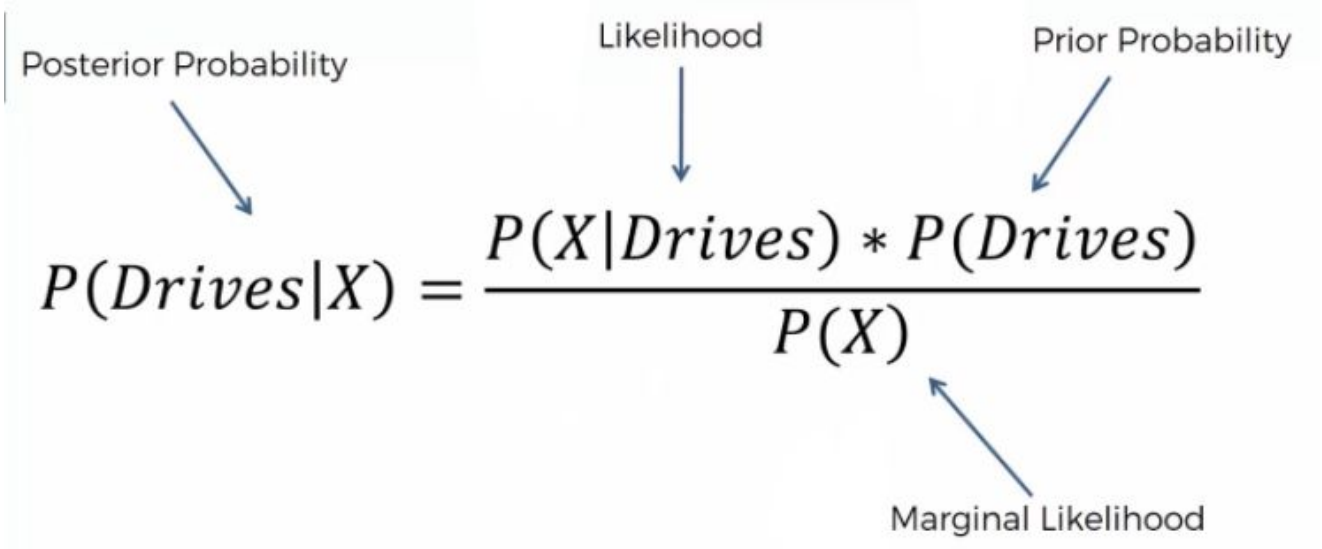
Example



create a model to find whether a person is going to the office by driving or walking using salary and age of the person.

Example


Posterior Probability
Likelihood
Prior Probability
$P(Walks|X) = \frac{P(X|Walks) * P(Walks)}{P(X)}$
Marginal Likelihood


Posterior Probability
Likelihood
Prior Probability
$P(Drives|X) = \frac{P(X|Drives) * P(Drives)}{P(X)}$
Marginal Likelihood

Example

	Outlook	Play
0	Rainy	Yes
1	Sunny	Yes
2	Overcast	Yes
3	Overcast	Yes
4	Sunny	No
5	Rainy	Yes
6	Sunny	Yes
7	Overcast	Yes
8	Rainy	No
9	Sunny	No
10	Sunny	Yes
11	Rainy	No
12	Overcast	Yes
13	Overcast	Yes

Problem: If the weather is sunny, then the Player should play or not?

Frequency table for the Weather Conditions:

Weather	Yes	No
Overcast	5	0
Rainy	2	2
Sunny	3	2
Total	10	5

Likelihood table weather condition:

Weather	No	Yes	
Overcast	0	5	5/14= 0.35
Rainy	2	2	4/14=0.29
Sunny	2	3	5/14=0.35
All	4/14=0.29	10/14=0.71	

P(Yes|Sunny)= P(Sunny|Yes)*P(Yes)/P(Sunny)

P(Sunny|Yes)= 3/10= 0.3

P(Sunny)= 0.35

P(Yes)=0.71

So P(Yes|Sunny) = 0.3*0.71/0.35= **0.60**

P(No|Sunny)= P(Sunny|No)*P(No)/P(Sunny)

P(Sunny|NO)= 2/4=0.5

P(No)= 0.29

P(Sunny)= 0.35

So P(No|Sunny)= 0.5*0.29/0.35 = **0.41**

Hence on a Sunny day, Player can play the game.

Play-tennis example: estimating $P(x_i|C)$

Each data sample is represented by an n-dimensional feature vector, X = (x1; x2; : : :; xn), depicting n measurements made on the sample from n attributes, respectively A1;A2; ::;An.

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

An unseen sample X = <rain, hot, high, false>

P(p) = 9/14

P(n) = 5/14

Example

An unseen sample X = <rain, hot, high, false>

P(Play | X) P(NotPlay | X)

P(Play | X) = P(X| Play) . P(Play)

= P(rain|p)·P(hot|p)·P(high|p)·P(false|p)·P(play)

P(NotPlay | X) = P(X| Not Play) . P(Not Play)

= P(rain|n)·P(hot|n)·P(high|n)·P(false|n)·P(Not Play)

$$P(C_i|X) > P(C_j|X) \text{ for } 1 \leq j \leq m, j \neq i.$$

Example

MLA Assi : 6 IT 1

outlook	
P(sunny|p) =	P(sunny|n) =
P(overcast|p) =	P(overcast|n) =
P(rain|p) =	P(rain|n) =
temperature	
P(hot|p) =	P(hot|n) =
P(mild|p) =	P(mild|n) =
P(cool|p) =	P(cool|n) =
humidity	
P(high|p) =	P(high|n) =
P(normal|p) =	P(normal|n) =
windy	
P(true|p) =	P(true|n) =
P(false|p) =	P(false|n) =

Example

outlook	
P(sunny|p) = 2/9	P(sunny|n) = 3/5
P(overcast|p) = 4/9	P(overcast|n) = 0
P(rain|p) = 3/9	P(rain|n) = 2/5
temperature	
P(hot|p) = 2/9	P(hot|n) = 2/5
P(mild|p) = 4/9	P(mild|n) = 2/5
P(cool|p) = 3/9	P(cool|n) = 1/5
humidity	
P(high|p) = 3/9	P(high|n) = 4/5
P(normal|p) = 6/9	P(normal|n) = 2/5
windy	
P(true|p) = 3/9	P(true|n) = 3/5
P(false|p) = 6/9	P(false|n) = 2/5

Example

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

$P(X|p)\cdot P(p) = P(rain|p)\cdot P(hot|p)\cdot P(high|p)\cdot P(false|p)\cdot P(p)$

$= 3/9\cdot 2/9\cdot 3/9\cdot 6/9\cdot 9/14$

$= 0.010582$

$P(X|n)\cdot P(n) = P(rain|n)\cdot P(hot|n)\cdot P(high|n)\cdot P(false|n)\cdot P(n)$

$= 2/5\cdot 2/5\cdot 4/5\cdot 2/5\cdot 5/14$

$= 0.018286$

Sample X is classified in class n (don't play)

The naive Bayesian classier

rid	age	income	student	credit_rating	Class: buys_computer
1	<30	high	no	fair	no
2	<30	high	no	excellent	no
3	30-40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	30-40	low	yes	excellent	yes
8	<30	medium	no	fair	no
9	<30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<30	medium	yes	excellent	yes
12	30-40	medium	no	excellent	yes
13	30-40	high	yes	fair	yes
14	>40	medium	no	excellent	no

Test tuple ={“<30”,”medium”,”yes”,”fair”}

Example

	Spam = *yes*	Spam = *no*
TimeZone = *US*	10	5
TimeZone = *EU*	0	0

P(TimeZone=US|Spam=yes)=10/10=1

P(TimeZone=EU|Spam=yes)=0/10=0

	Spam = *yes*	Spam = *no*
TimeZone = *US*	11	6
TimeZone = *EU*	1	1

P(TimeZone=US|Spam=yes)=11/12

P(TimeZone=EU|Spam=yes)=1/12

An approach to overcome this 'zero-frequency problem' in a Bayesian environment is to add one to the count for every attribute value-class combination when an attribute value doesn't occur with every class value.

What is Laplace Correction?

- When you have a model with a lot of attributes, it is possible that the entire probability might become zero because one of the feature values is zero. To overcome this situation, you can increase the count of the variable with zero to a small value like in the numerator so that the overall probability doesn't come as zero.
- This type of correction is called the Laplace Correction. Usually, all naive Bayes models use this implementation as a parameter.

Example

A patient takes a X-Ray lab test and the result comes back positive.

The test returns a correct positive result (+) in only 98% of the cases in which the disease is actually present, and a correct negative result (-) in only 97% of the cases in which the disease is not present. Furthermore, 0.008 of the entire population have this cancer

Does patient have cancer or not?

Suppose a positive result (+) is returned...

0.008 of the entire population have this cancer

$$P(cancer) = 0.008 \qquad P(\neg cancer) = 0.992$$

The test returns a correct positive result (+) in only 98% of the cases in which the disease is actually present

$$P(+|cancer) = 0.98 \qquad P(-|cancer) = 0.02$$

The test returns a correct negative result (-) in only 97% of the cases in which the disease is not present.

$$P(+|\neg cancer) = 0.03 \qquad P(-|\neg cancer) = 0.97$$

A patient takes a X-Ray lab test and the result comes back positive.

$$P(+|cancer) \cdot P(cancer) = 0.98 \cdot 0.008 = 0.0078$$

$$P(+|\neg cancer) \cdot P(\neg cancer) = 0.03 \cdot 0.992 = 0.0298$$

$$h_{MAP} = \neg cancer$$

Implementation of Naive Bayes

https://www.analyticsvidhya.com/blog/2021/01/a-guide-to-the-naive-bayes-algorithm/

Social Networks Ads

we will use the Social_Networks_Ads.csv file which contains information about the users like Gender, Age, Salary.

The **Purchased column** contains the **labels** for the users. This is a **binary classification** (we have two classes).

If the **label is 1** it means that the user **has bought product X** and **0** means the users **hasn't bought** that specific product.

User ID	Gender	Age	EstimatedSalary	Purchased
15624510	Male	19	19000	0
15810944	Male	35	20000	0
15668575	Female	26	43000	0
15603246	Female	27	57000	0
15804002	Male	19	76000	0
15728773	Male	27	58000	0
15598044	Female	27	84000	0
15694829	Female	32	150000	1
15600575	Male	25	33000	0
15727311	Female	35	65000	0
15570769	Female	26	80000	0
15606274	Female	26	52000	0
15746139	Male	20	86000	0
15704987	Male	32	18000	0

Code

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import sklearn
```

Importing of dataset

```
dataset = pd.read_csv('Social_Network_Ads.csv')
X = dataset.iloc[:, [1, 2, 3]].values
y = dataset.iloc[:, -1].values
```

Since our dataset containing character variables we have to encode it using LabelEncoder

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
X[:,0] = le.fit_transform(X[:,0])
```

Train test Split

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)
```

Feature scaling

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Training the Naive Bayes model on the training set

```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

Let's predict the test results

```
y_pred  =  classifier.predict(X_test)
```

Predicted and actual value

```
from sklearn.metrics import confusion_matrix,accuracy_score
cm = confusion_matrix(y_test, y_pred)
ac = accuracy_score(y_test,y_pred)
```

confusion matrix –

ac – 0.9125

	0	1
0	55	3
1	4	18

The Complete Code

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Social_Network_Ads.csv')
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, -1].values

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

# Training the Naive Bayes model on the Training set
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)

# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix, accuracy_score
ac = accuracy_score(y_test,y_pred)
cm = confusion_matrix(y_test, y_pred)
```

Types of NB

1. Multinomial Naïve Bayes Classifier:

Feature vectors represent the frequencies with which certain events have been generated by a **multinomial distribution**. This is the event model typically used for document classification.

2. Bernoulli Naïve Bayes Classifier:

In the multivariate Bernoulli event model, features are independent booleans (binary variables) describing inputs. Like the multinomial model, this model is popular for document classification tasks, where binary term occurrence (i.e. a word occurs in a document or not) features are used rather than term frequencies (i.e. frequency of a word in the document).

3. Gaussian Naïve Bayes Classifier:

In Gaussian Naïve Bayes, continuous values associated with each feature are assumed to be distributed according to a **Gaussian distribution** (Normal distribution). When plotted, it gives a bell-shaped curve which is symmetric about the mean of the feature values as shown below:

Exercise : Naïve Bayes

Use wine dataset from sklearn.datasets to classify wines into 3 categories. Load the dataset and split it into test and train. After that train the model using Gaussian and Multinominal classifier and post which model performs better. Use the trained model to perform some predictions on test data.

Reference Link

1)https://medium.com/analytics-vidhya/ordinary-least-square-ols-method-for-linear-regression-ef8ca10aadfc

2)https://medium.com/30-days-of-machine-learning/day-3-k-nearest-neighbors-and-bias-variance-tradeoff-75f84d515bdb

3)https://towardsdatascience.com/why-do-we-use-gradient-descent-for-optimization-in-machine-learning-19428760a7ed