NANYANG
THE INNOVATIVE POLYTECHNIC

Introduction to AI and Machine Learning

School of Engineering
Nanyang Polytechnic

Topics

1. Overview of AI, ML, and DL
2. Machine Learning Types and Techniques
3. Machine Learning Modeling Process
4. Practical: Regression Models in ML

2. Machine Learning Types and Techniques

2.1. Supervised, Unsupervised, and Reinforcement Learning
2.2. Classification
2.3. Regression
2.4. Clustering
2.5. Artificial Neural Networks

2.1. Supervised, Unsupervised, and Reinforcement Learning

Machine Learning Types

- Supervised learning
- Unsupervised learning
- Reinforcement learning

Supervised Learning

Regression

- Obtain an optimal model with required performance through training and learning based on the samples of known categories.
- Then, use the model to map all inputs to outputs and check the output for the purpose of classifying unknown data.

Types of Supervised Learning

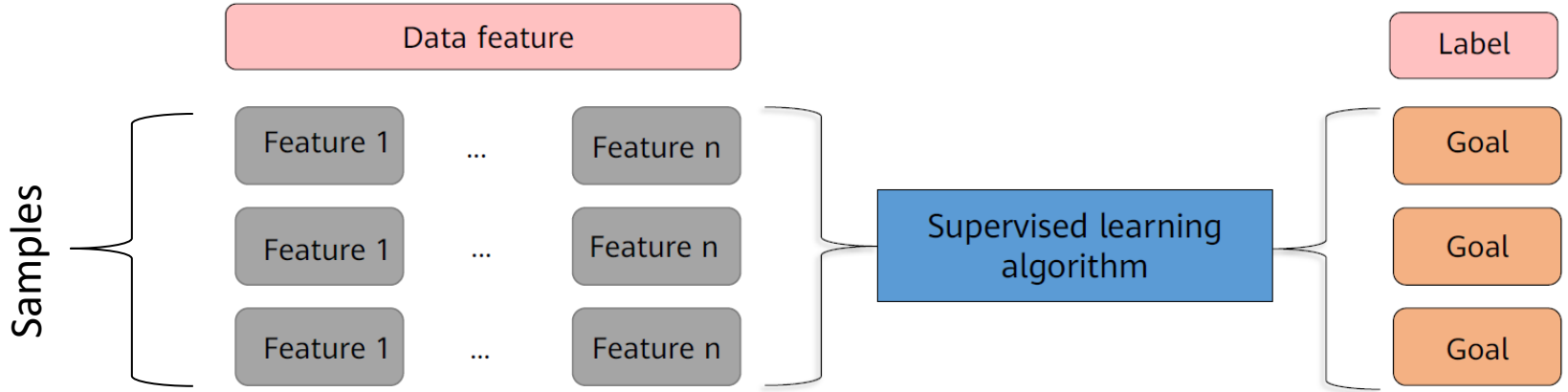
- Regression — Outcome is continuous (numerical)
- Classification — Outcome is a category
  - Binary
  - Multiclass

Supervised Learning Overview

Data with answers + Model → Fit → Model

Data without answers + Model → Predict → Predicted answers

Supervised Learning



Features →

Samples

Weather	Temperature	Wind Speed
Sunny	Warm	Strong
Rainy	Cold	Fair
Sunny	Cold	Weak

Supervised Learning Algorithm

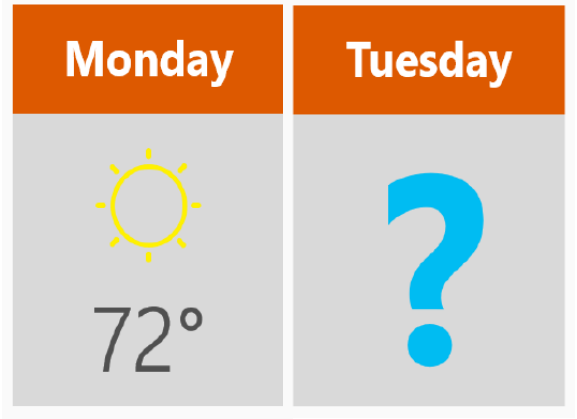
Enjoy Sports
Yes
No
Yes

← Class

Predictions

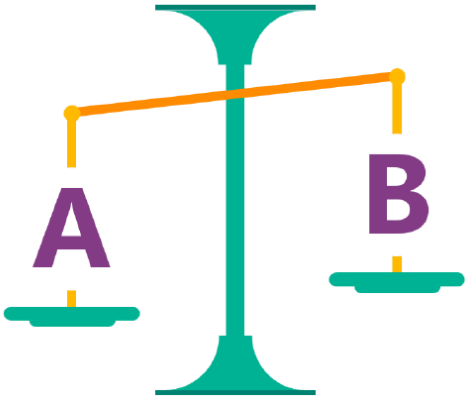
Supervised Learning - Regression

- Regression: Response to the features values of samples in a sample dataset to discover the dependencies/relationships between the feature values by mapping the relationships through functions.
- Regression Questions:
  - How much will I benefit from the stock next week?
  - What's the temperature on Tuesday?



Supervised Learning - Classification

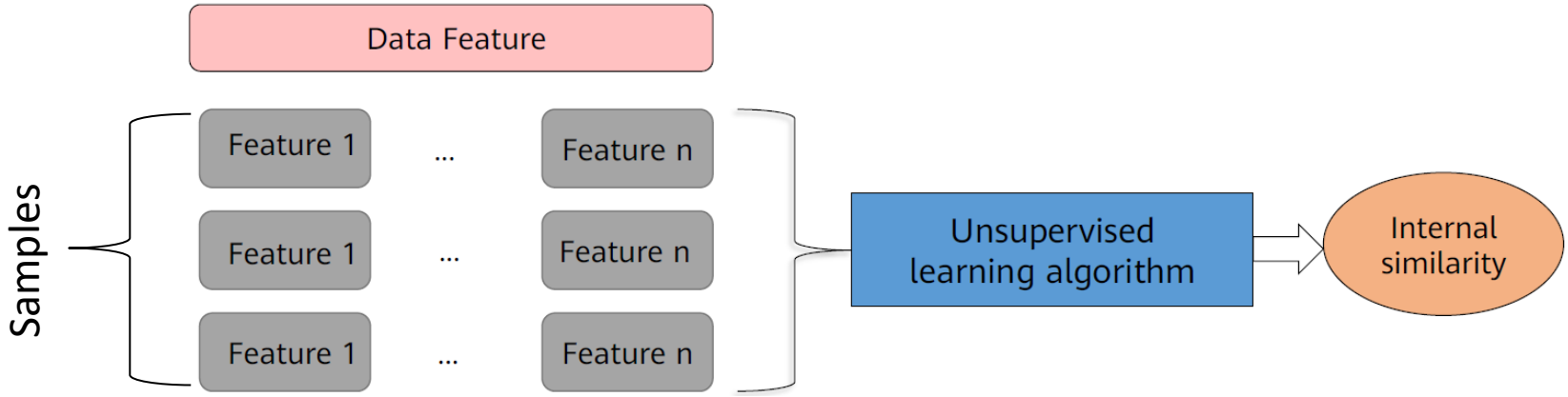
- Classification: maps samples in a sample dataset to a specified category by using a classification model.
- Classification Questions:
  - Will there be a traffic jam on XX road during the morning rush hour tomorrow?
  - Which method is more attractive to customers: $10 voucher or 25% off?



Unsupervised Learning

- For unlabeled samples, the learning algorithms directly model the input datasets.
- Clustering is a common form of unsupervised learning.
- We only need to put highly similar samples together, calculate the similarity between new samples and existing ones, and classify them by similarity.

Unsupervised Learning



Features

Samples

Monthly Consumption	Commodity	Consumption Time
1000–2000	Badminton racket	6:00–12:00
500–1000	Basketball	18:00–24:00
1000–2000	Game console	00:00–6:00

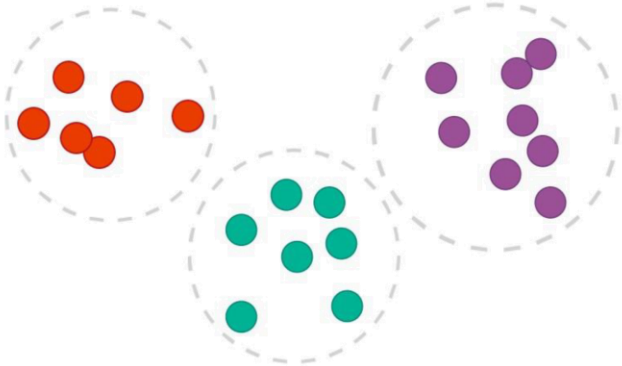
Category
Cluster 1
Cluster 2

12

Supervised Learning - Clustering

- Clustering: classifies samples in a sample dataset into several categories based on the clustering model.

- Clustering Questions:
  - Will the similarity of samples belonging to the same category be high?
  - Which audiences like to watch movies of the same subject?
  - Which of these components are damaged in a similar way?

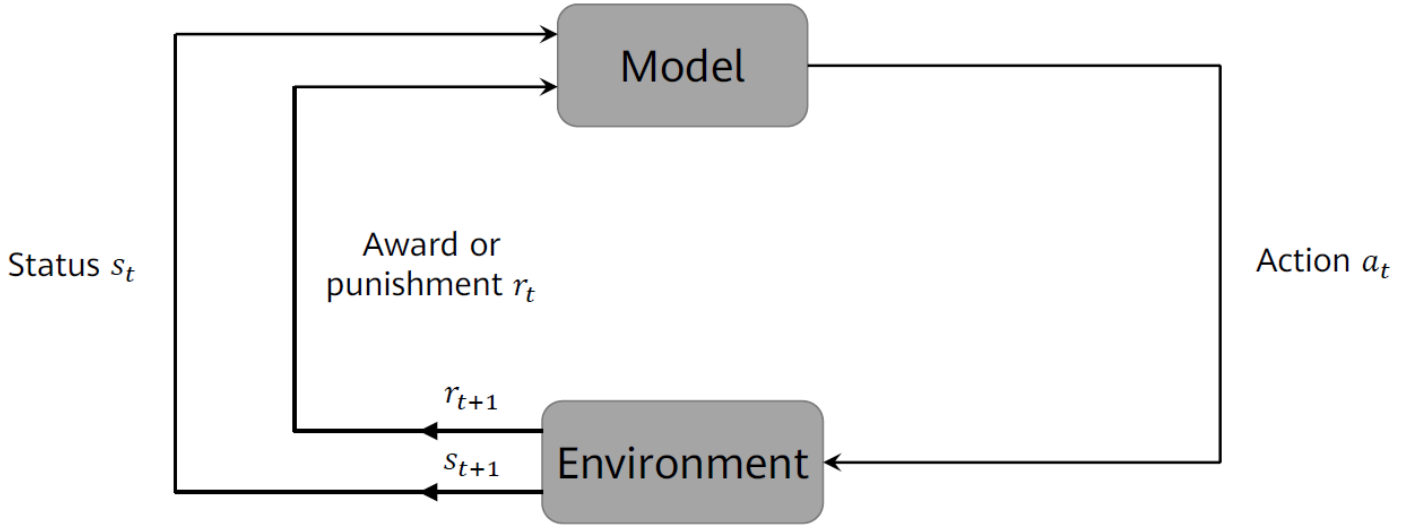


Reinforcement Learning

- The goal of RL is to train an agent to complete a task within an uncertain environment.
- It is an area of machine learning concerned with how agents ought to take actions in an environment to maximize some notion of cumulative reward.
- The difference between reinforcement learning and supervised learning is the teacher/reward/reinforcement signal.
- The reinforcement signal provided by the environment in RL is used to evaluate the action (scalar signal) rather than telling the learning system how to perform correct actions.

Reinforcement Learning

The model perceives the environment, takes actions, and makes adjustments and choices based on the status and award or punishment.

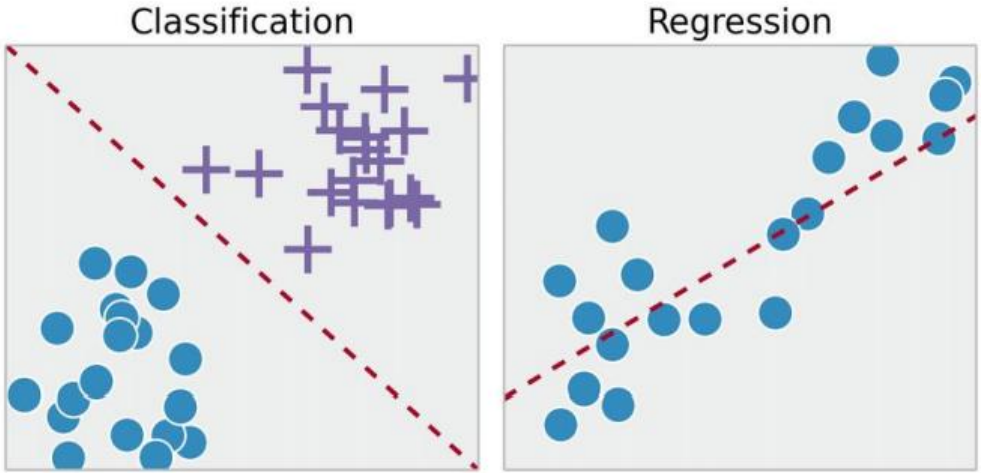


Reinforcement Learning - Best Behavior

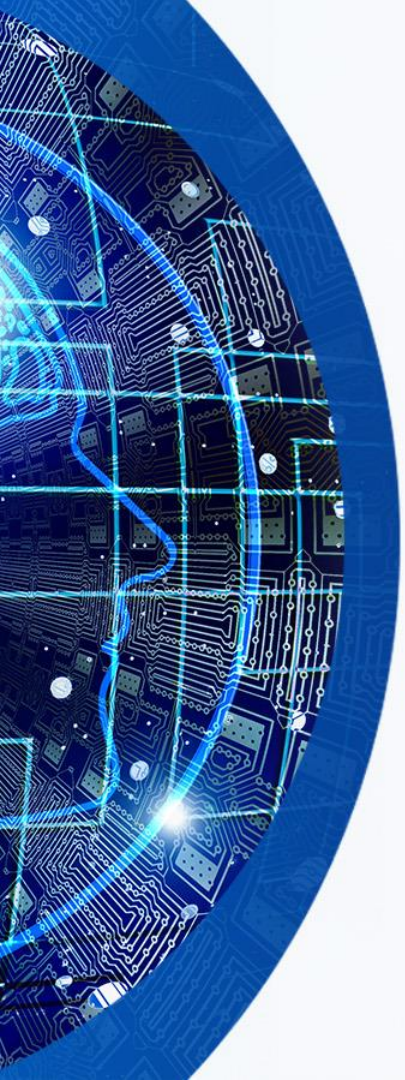
- Reinforcement learning: always looks for best behaviors. Reinforcement learning is targeted at machines or robots.
  - Autopilot: Should it brake or accelerate when the yellow light starts to flash?
  - Cleaning robot: Should it keep working or go back for charging?

2.2. Classification

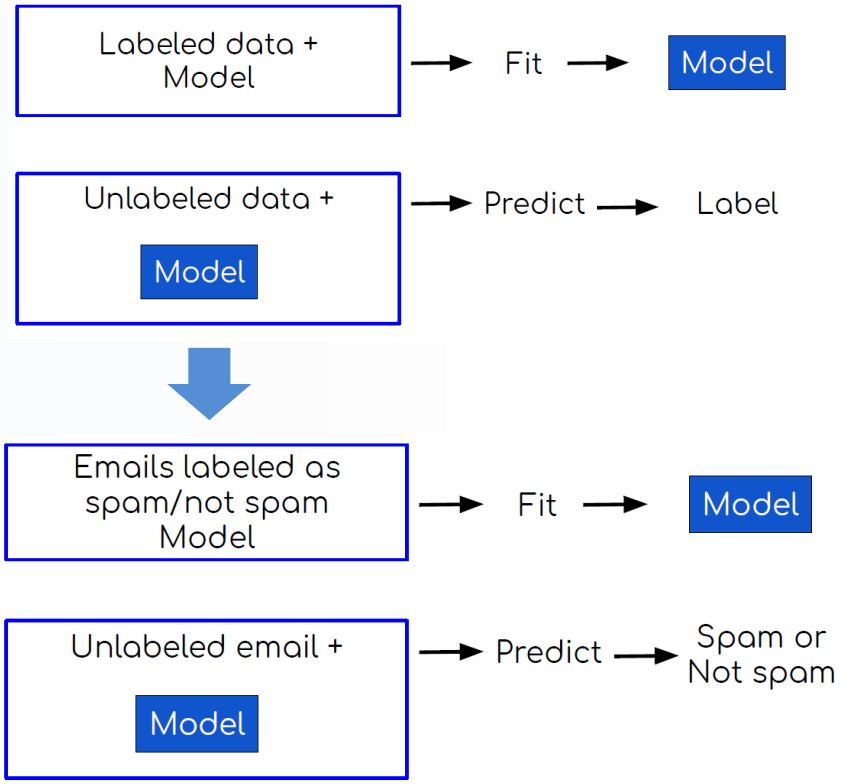
What is Classification?



- Classification is to identifying to which category an object belongs to.
- Classification is a supervised learning method. During training, labels/classes are provided.

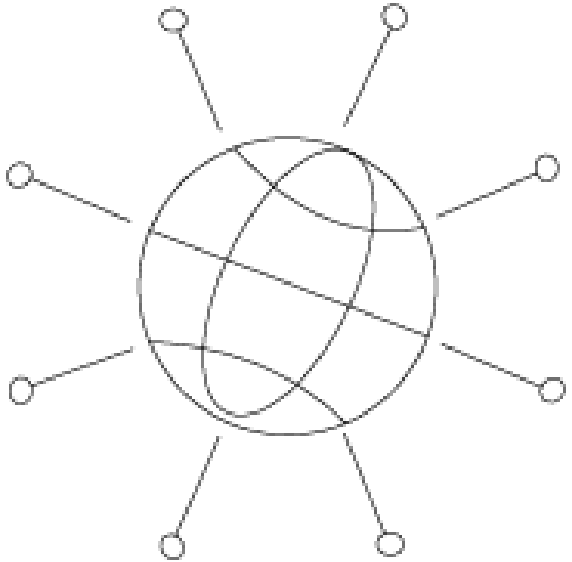


Classification: Categorial Answers

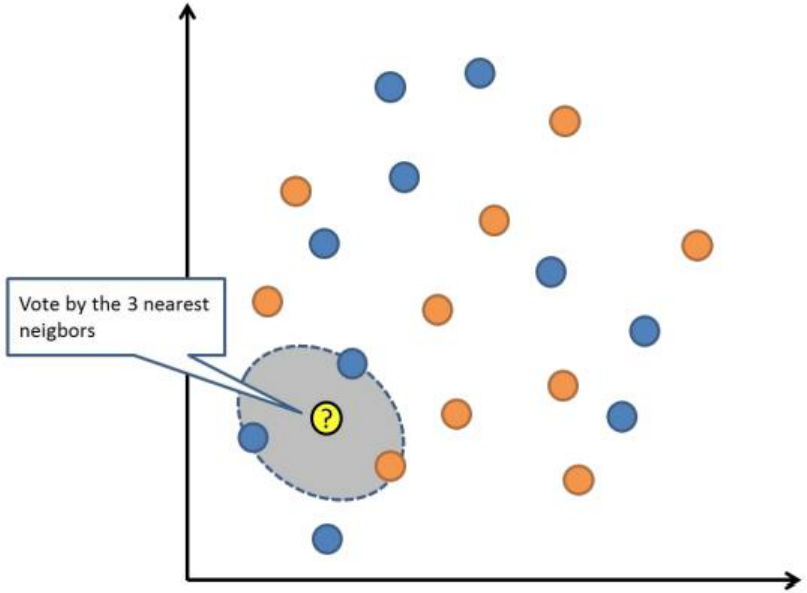


Classification Algorithms

- K Nearest Neighbors
- Logistic Regression
- Support Vector Machine
- Gaussian Naive Bayes
- Decision Tree
- Ensemble Methods



K Nearest Neighbors (KNN)

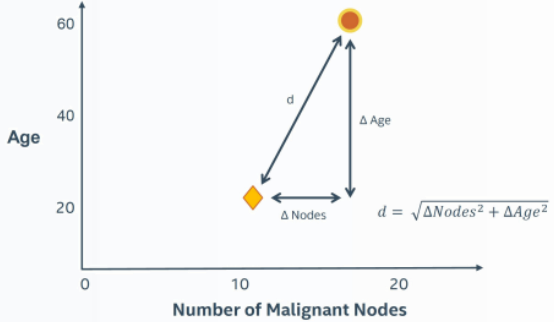


- A simple majority vote of the nearest neighbors of each point.
- Training data is the model. Fitting is fast just store data. However, Prediction can be slow because lots of distances to measure
- Decision boundary is flexible.

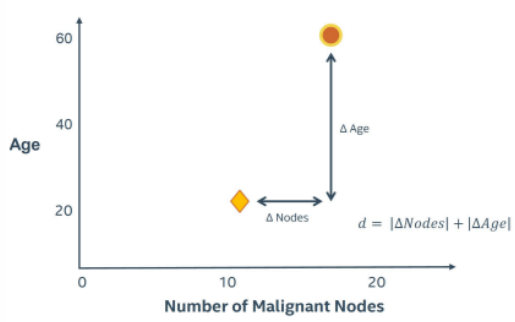
Distance Measure for KNN

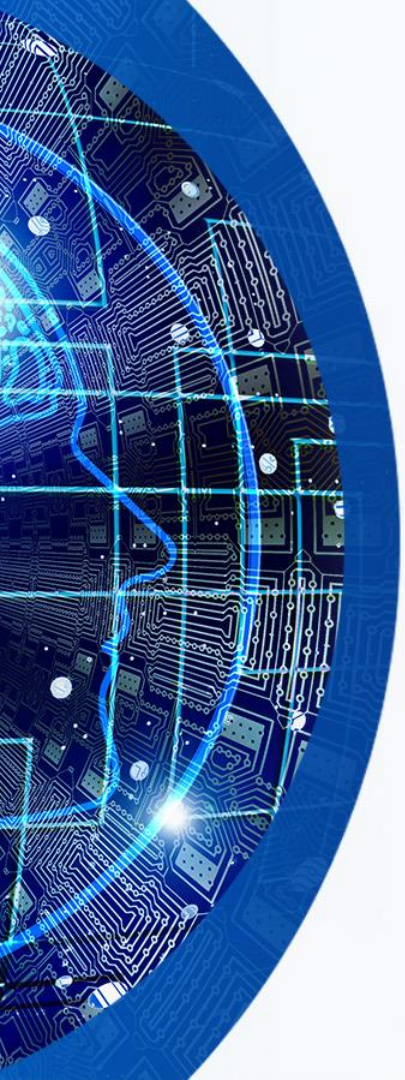
- KNN depends on the distance measure
- There are two major distance measures:
  - Euclidean (L2)
  - Manhattan (L1)

Euclidean (L2)



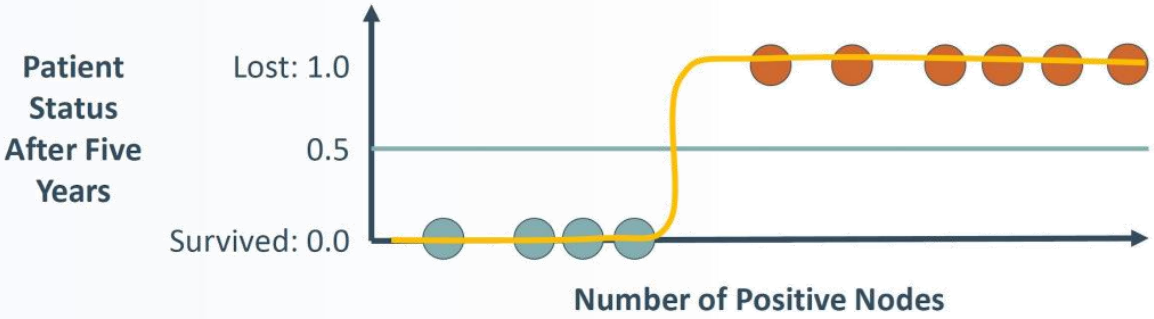
Manhattan (L1)





Logistic Regression

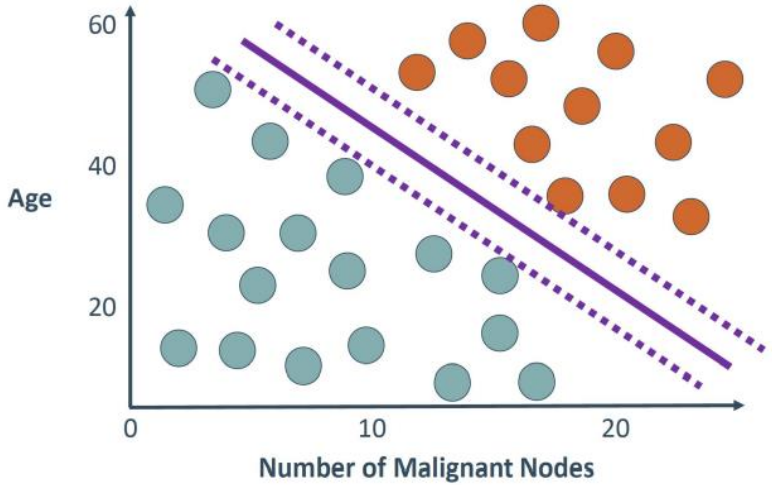
- Fit the logistic function to the data
- Fitting can be slow must find best parameters
- Prediction is fast to calculate expected value
- Decision boundary is simple, less flexible



$$y_\beta(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x+\varepsilon)}}$$

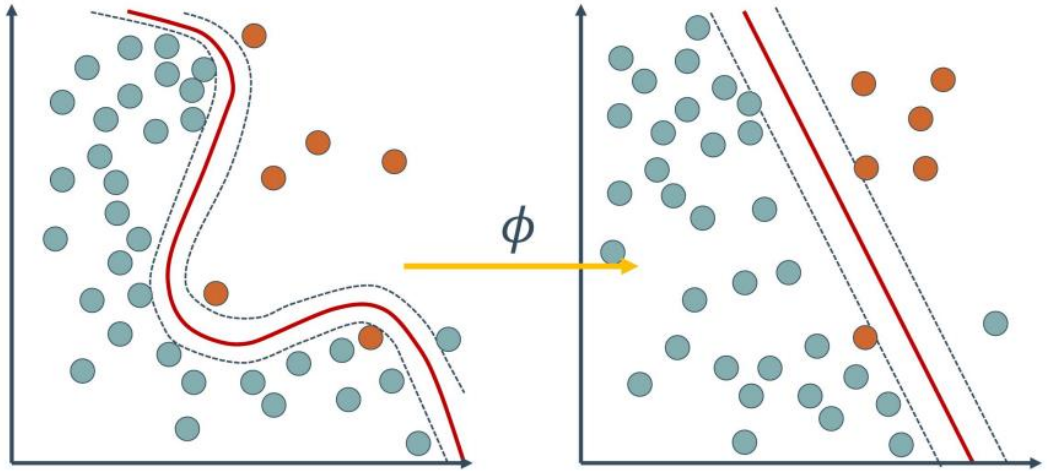
Support Vector Machine (SVM)

- Identify a linear hyperplane (boundary) with maximum distance apart



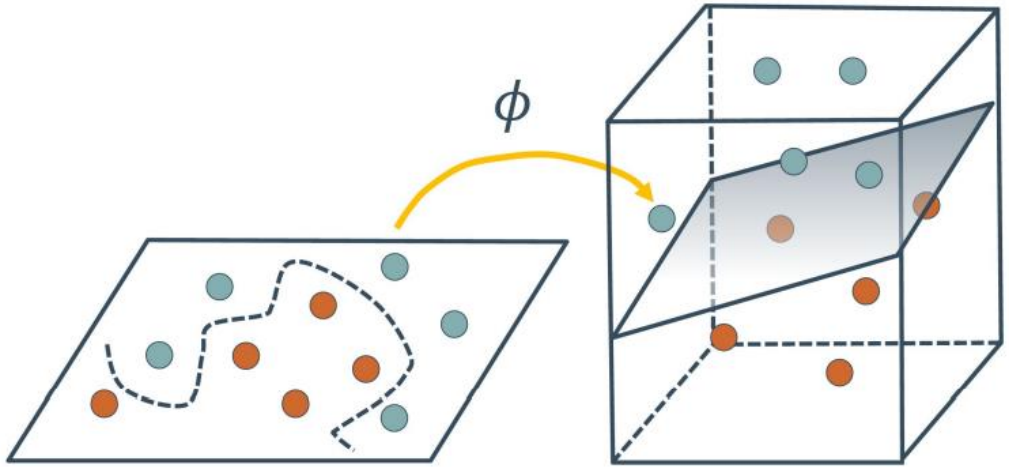
Non-Linear Boundary in SVM

- Non-linear data can be made linear with higher dimension using Kernel trick



Kernel Trick

- Transform data so that it is linearly separable



Gaussian Naïve Bayes (GNB)

- Naive Bayes is a conditional probability model.
- Given the feature vector x, it predict the probability for class C
- GNB assumes each feature is Gaussian distributed

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})} \qquad \text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

$$p(x = v \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}}\, e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

Example of GNB

Problem: Classify whether a given person is a male or a female based on the measured features. The features include height, weight, and foot size

Person	Height (feet)	Weight (lbs)	Foot size (inches)
Male	6	180	12
Male	5.92 (5'7")	190	11
Male	5.58 (5'7")	170	12
Male	5.92 (5'11")	165	10
Female	5	100	6
Female	5.5 (5'6")	150	8
Female	5.42 (5'5")	130	7
Female	5.75 (5'9")	150	9

Assuming Gaussian distributed, the feature means are computed as follows

Person	Mean (height)	Variance (Height)	Mean(Weight)	Variance (weight)	Mean (foot size)	Variance(foot size)
Male	5.885	$3{,}5033*10^{-2}$	176.25	$1{,}2292*10^{2}$	11.25	$9.1667*10^{-1}$
Female	5.4175	$9.7225*10^{-2}$	132.5	$5.5833*10^{2}$	7.5	1.6667

Example of GNB

- Below is a sample to be classified as male or female

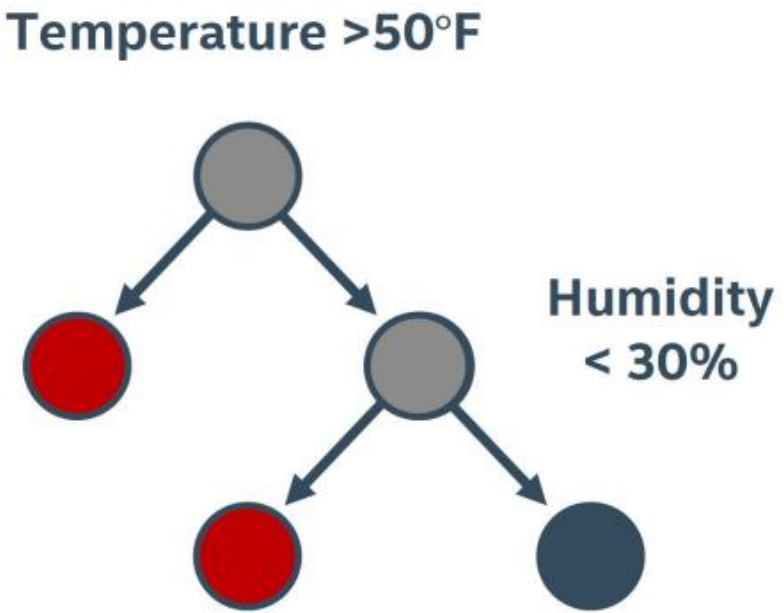
Person	Height(feet)	Weight(lbs)	Foot size(inches)
sample	6	130	8

posterior of male ~ $6.19\text{x}10^{-9}$
posterior of female ~ $5.37\text{x}10^{-4}$

Therefore the person is likely to be female

Decision Tree

- Decision tree is easy to interpret and implement
- Heterogeneous input data allowed, preprocessing required
- However, decision trees tend to overfit
- Pruning helps reduce variance to a point. Often not significant for model to generalize well



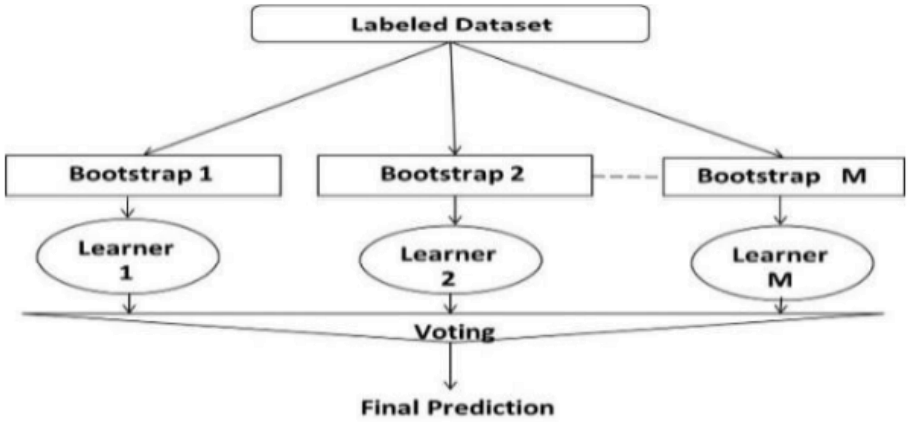
Ensemble Methods

The main principle behind Ensemble methods is that a group of “weak learners” can come together to form a “strong learner”.

There are 3 types of Ensemble methods

1. Bagging (Bootstrap Aggregating)
2. Boosting
3. Stacking

Bagging

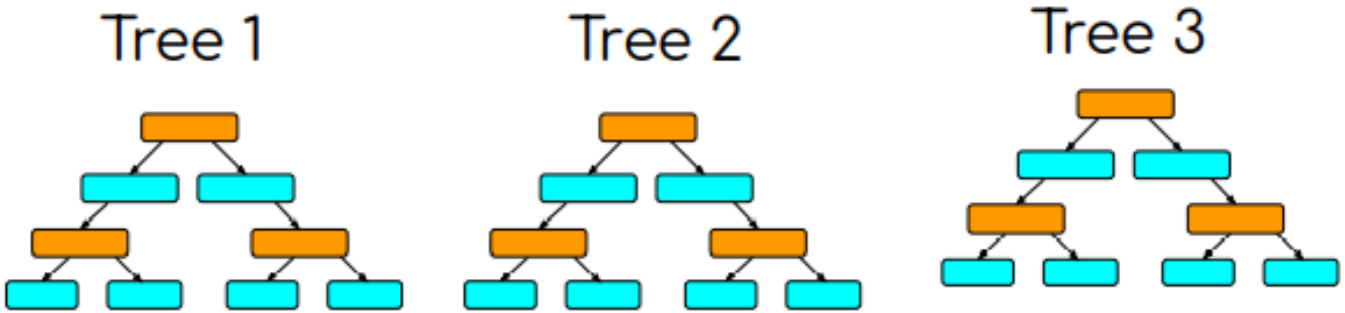


Bagging (Bootstrap Aggregating) creates separate samples of the training dataset and creates a classifier for each sample.

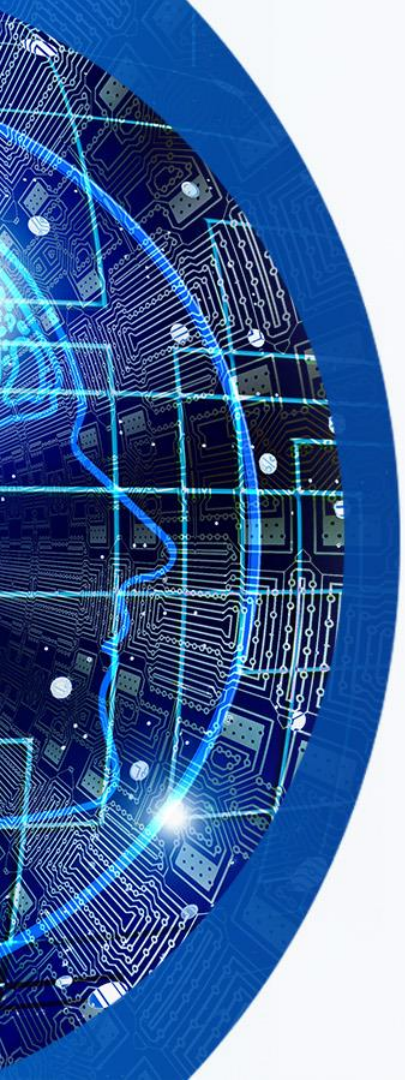
- The results of these multiple classifiers are then combined (such as averaged or majority voting).

Example: Random Forest

Random Forest classifier is a bagging ensemble method based on lots of decision trees with random selection of subsets of training samples.

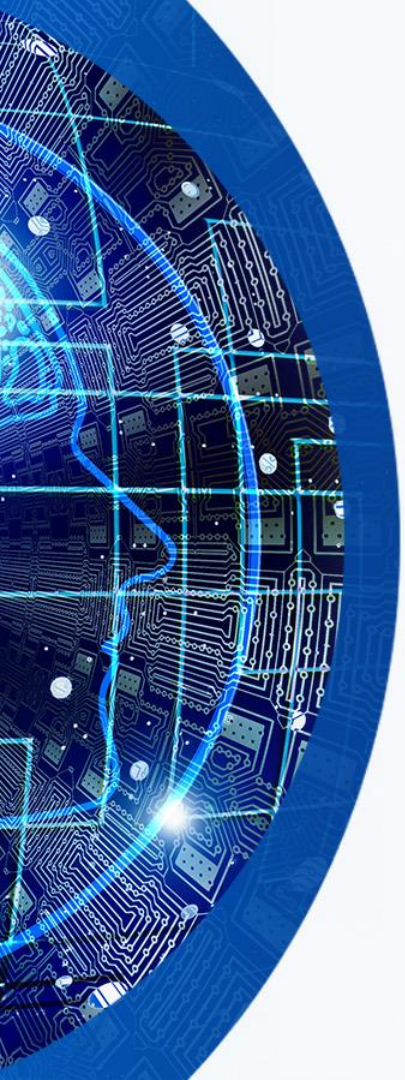


The result is based on the majority votes from all the decision trees



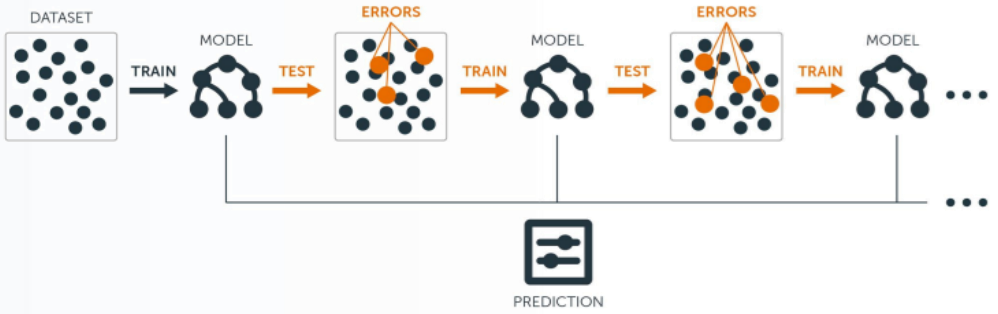
Boosting

- **Boosting** starts out with a base classifier that is prepared on the training data.
- A second classifier is then created behind it to focus on the instances in the training data that the first classifier got wrong.
- The process continues to add classifiers until a limit is reached in the number of models or accuracy



Example: Gradient Boosting

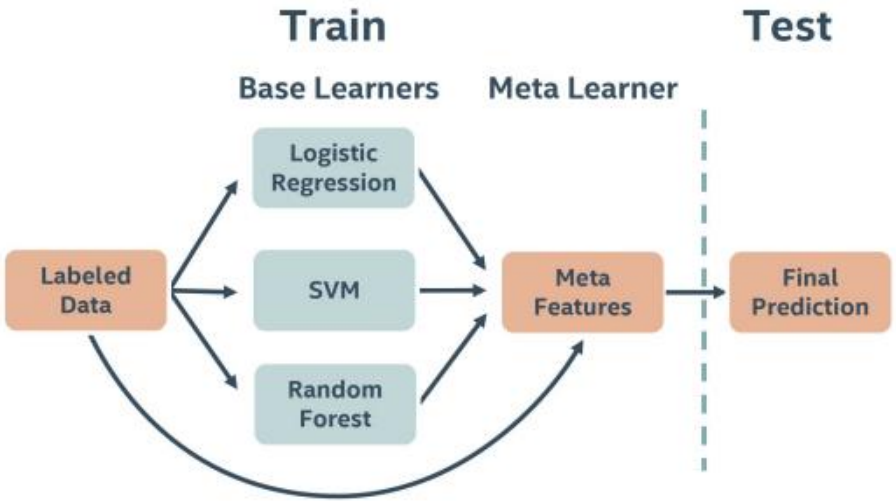
- Gradient Boosting is a ensemble boosting method that "boosting" many weak predictive models into a strong one, in the form of ensemble of weak models.
- Boosting utilizes different loss functions At each stage, the margin is determined for each point. Margin is positive for correctly classified points and negative for misclassifications. Value of loss function is calculated from margin



Stacking

- **Stacking** starts out with a set of base-level classifiers and train a meta-level classifier to combine the outputs of the base-level classifiers

  - Models of any kind combined to create stacked model
  - Output of base learners can be combined via majority vote or weighted
  - Additional hold out data needed if meta learner parameters are used
  - Be aware of increasing model complexity



Binary Classification

- In many predictive analysis, we are interested in YES/NO analysis such as spam/not-spam, health/not-healthy etc.
- In this case, we can form a confusion matrix of 2 columns and 2 rows

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Type II Error

Type I Error

Accuracy

Accuracy is the percentage of correct hits

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}$$

Recall or Sensitivity

Recall or Sensitivity is to identify all the positive instances

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

$$\text{Recall or Sensitivity} = \frac{TP}{TP + FN}$$

Precision

Precision is to identify only the positive instances

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

$$\text{Precision} = \frac{TP}{TP + FP}$$

Specificity

Specificity is to Identify only all the negative instances to detect false alarm

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

F1 Score

- For error measurement of a particular model, need to check all the four metrics
- However, it is more convenient to check a single parameter using F1 score

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

$$\textbf{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}$$

$$\textbf{Precision} = \frac{TP}{TP + FP}$$

$$\textbf{Recall or Sensitivity} = \frac{TP}{TP + FN}$$

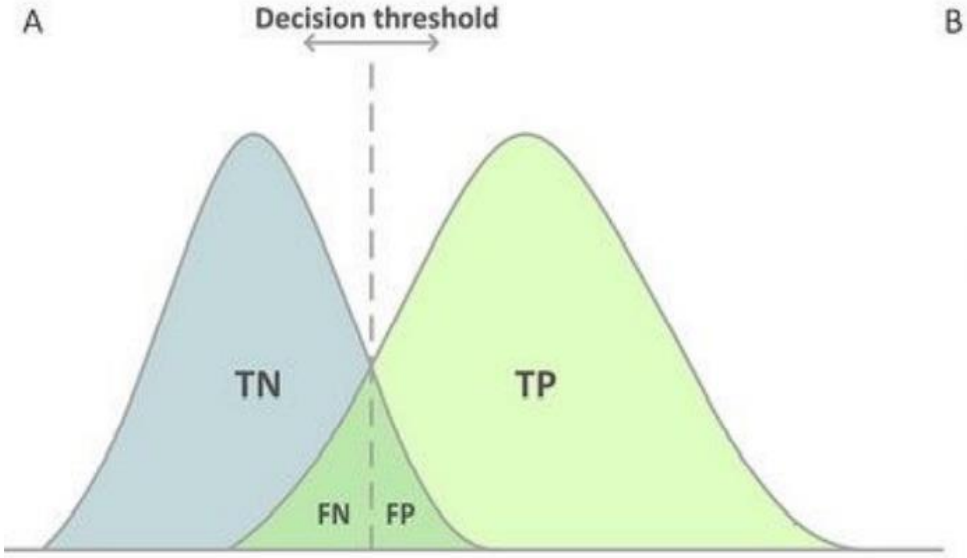
$$\textbf{Specificity} = \frac{TN}{TN + FP}$$

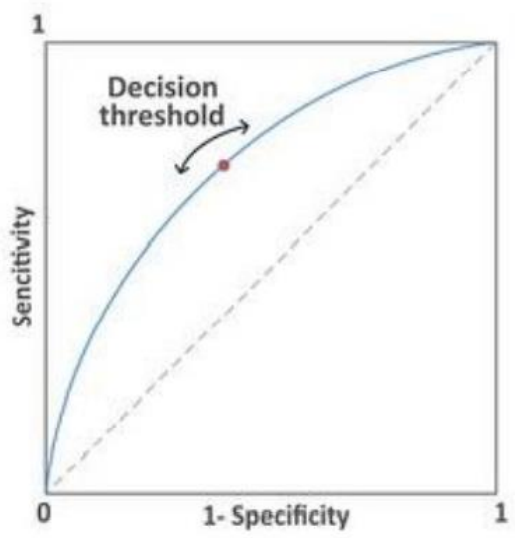
$$F1 = 2.\left[\frac{Precision * Recall}{Precision + Recall}\right]$$

$$= \frac{TP}{TP + \frac{1}{2}(FP+FN)}$$

Fine Tuning Binary Classification

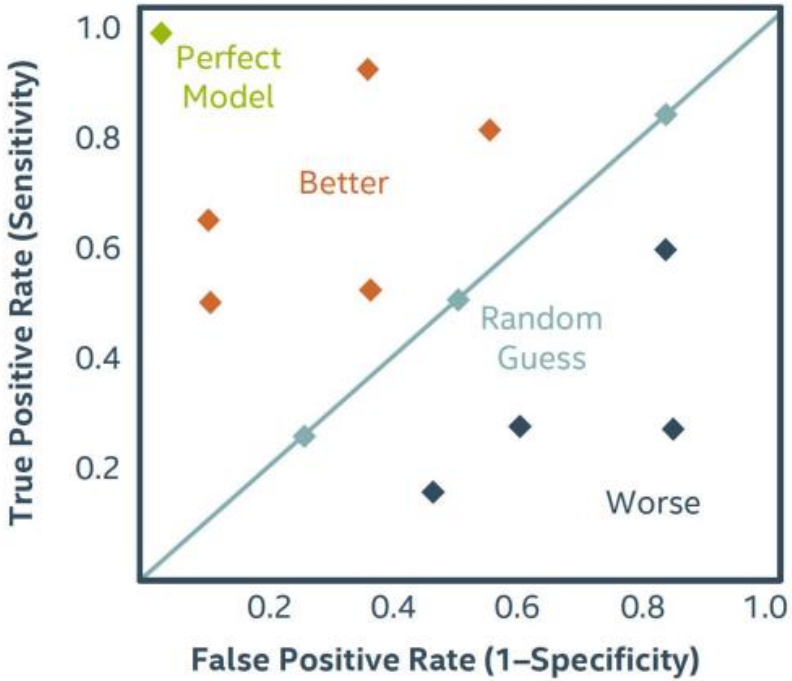
It is tempting to assume that the classification threshold (decision threshold) should always be 0.5, but thresholds are problem-dependent, and are therefore values that you must tune.





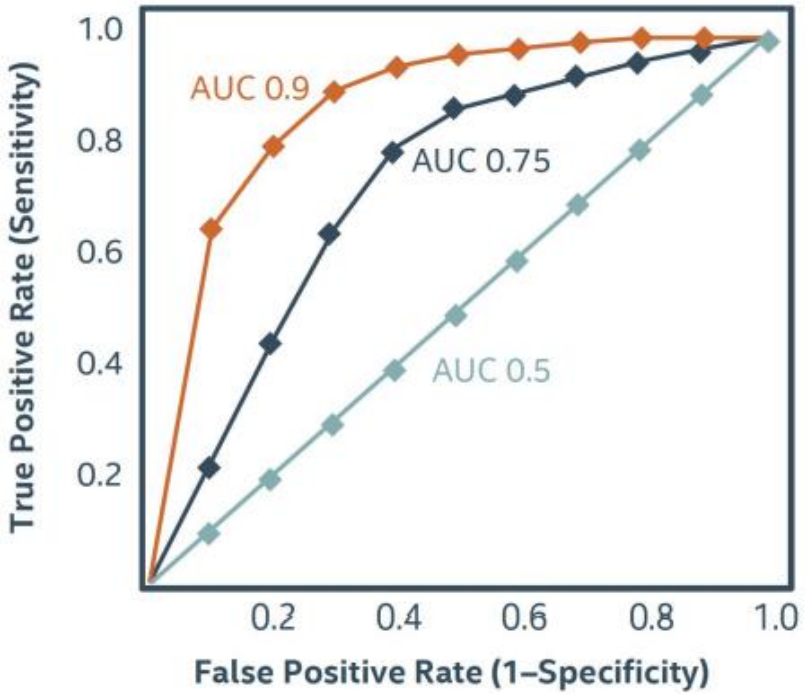
Receiver Operation Characteristics (ROC)

- Receiver Operating Characteristics (ROC) can be used to evaluate all the possible thresholds
- ROC is a plot of True vs False Positive Rates.



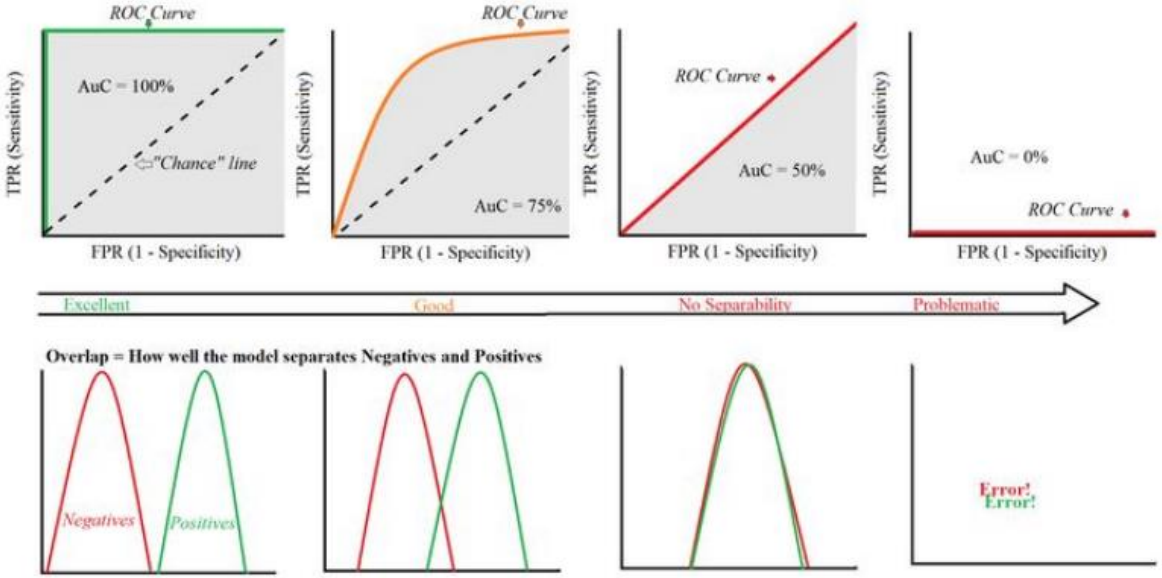
Area Under Curve (AUC)

- AUC measures the area under the ROC curve.
- It is a measure of how good is the binary classification model.



ROC and AUC

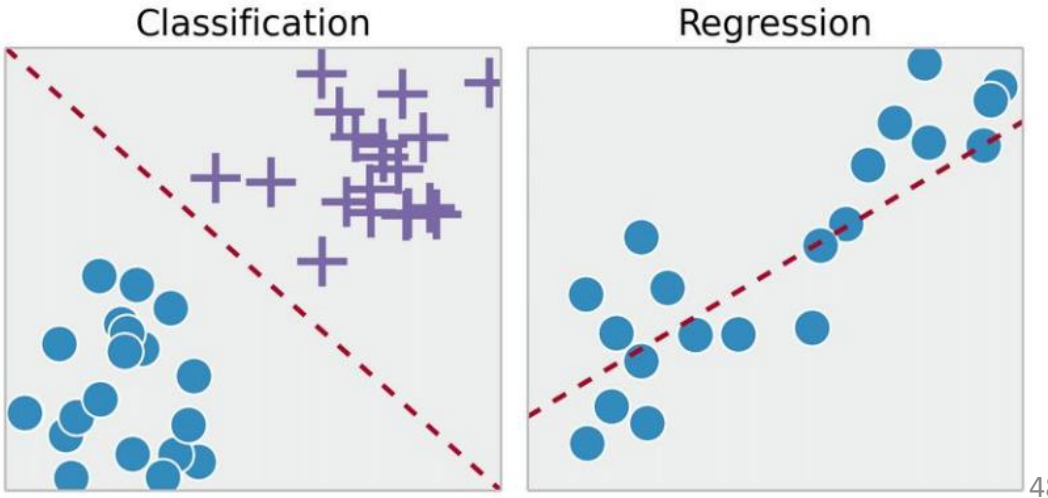
- With a ROC curve, you're trying to find a good model that optimizes the trade off between the False Positive Rate (FPR) and True Positive Rate (TPR).
- What counts here is how much area is under the curve (Area under the Curve = AuC).

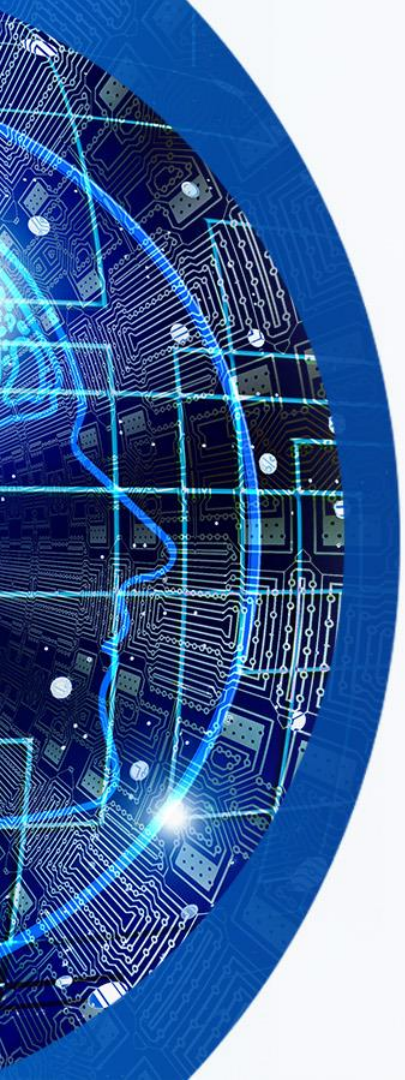


2.3 Regression

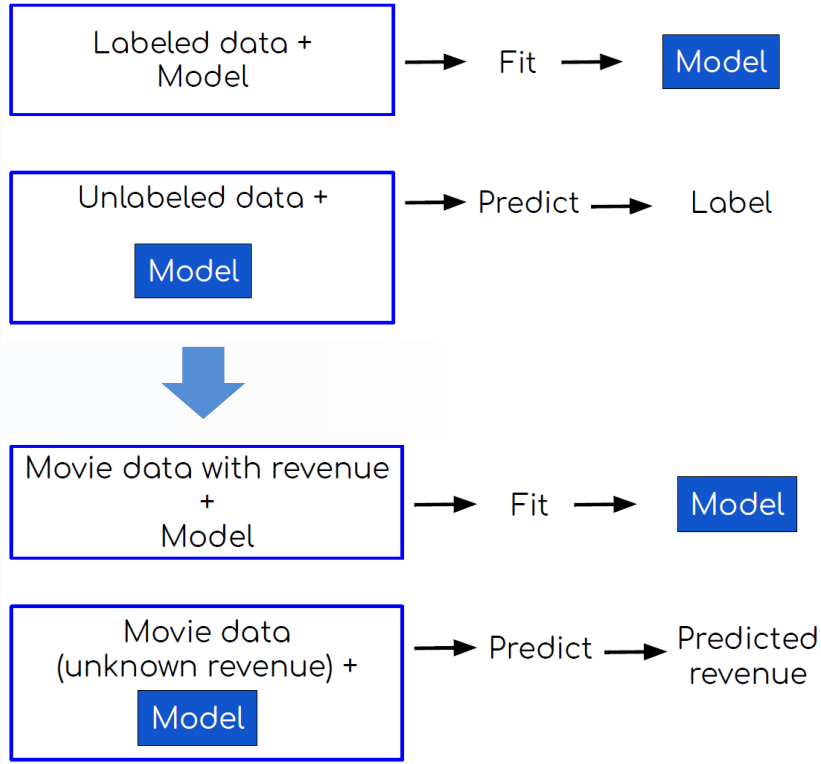
What is Regression

- Regression it to predicting a continuous-valued attribute associated with an object.
- Regression is a supervised learning method. During training, actual values are provided.

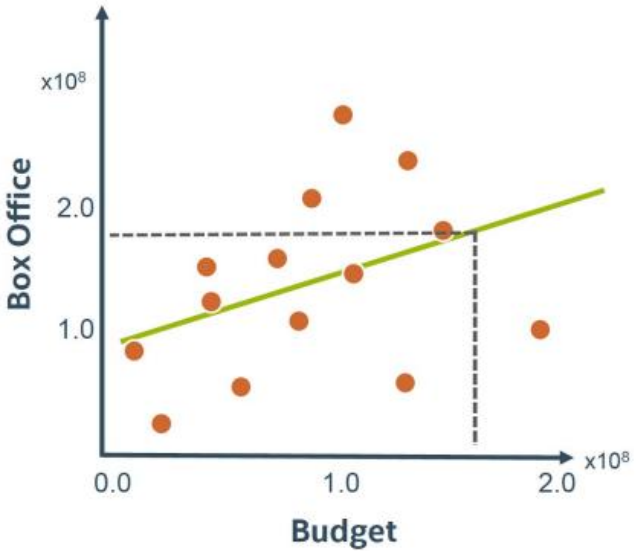




Classification: Numeric Answers



Regression Applications

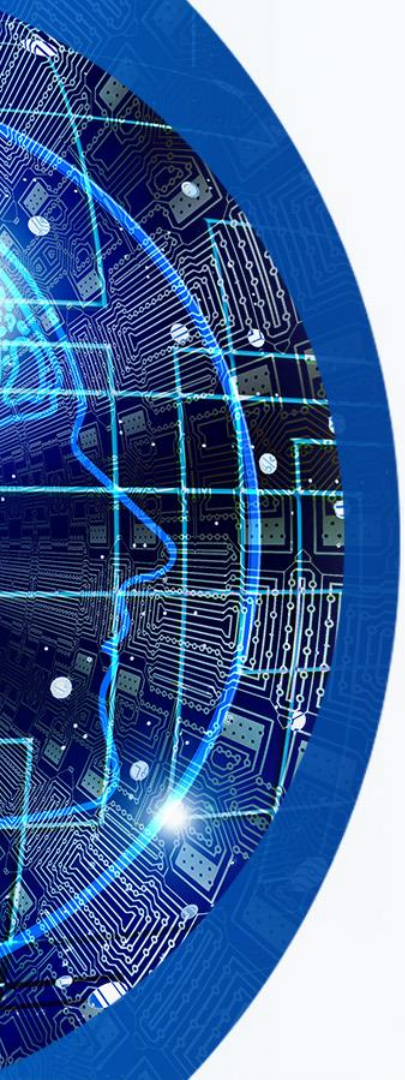


- Linear regression is the most common regression model. Many predictive models use linear regression models
- You can use a linear regression model to predict the box office from the budget.

$$y_\beta(x) = \beta_0 + \beta_1 x$$

β_0= 80 million, β_1= 0.6

Predict 175 Million Gross for 160 Million Budget

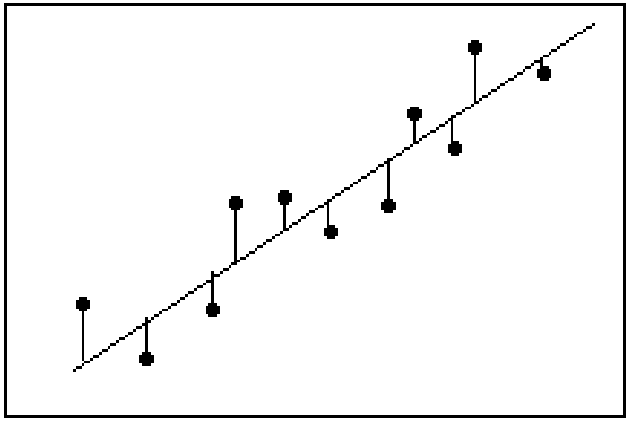


Regression Algorithms

- Linear Regression (Most Common)
- Ridge Regression
- Lasso Regression
- Elastic Net Regression

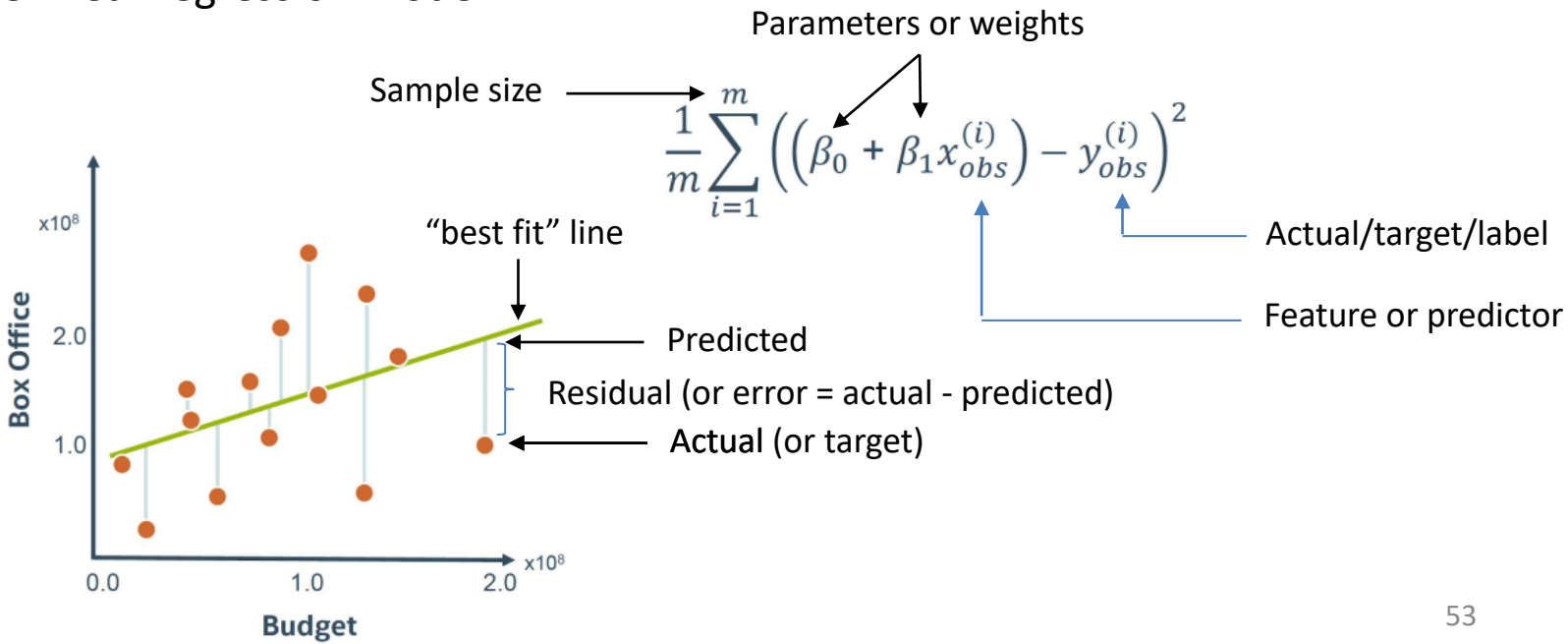
Assessing the Goodness-of-Fit

- After you have fit a linear model using regression analysis, you need to determine how well the model fits the data. There are several common functions used:
  - Mean squared error
  - Mean absolute error
  - R-squared (or Coefficient of Determination)



Mean Square Error

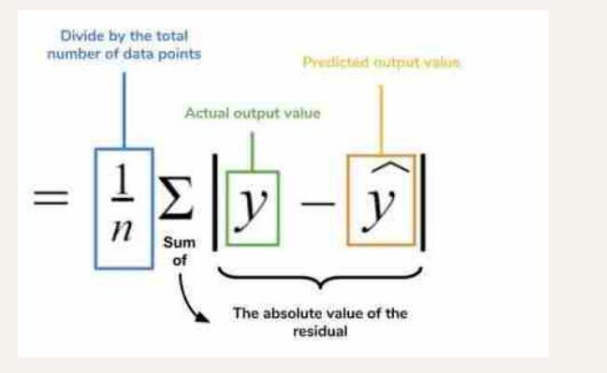
- Mean Square Error (MSE) is the common loss function to measure how good is the linear regression model.

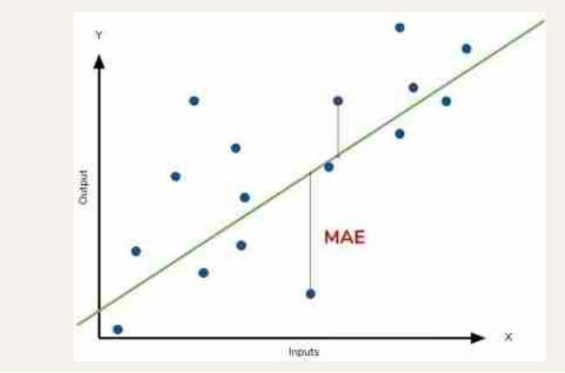


Mean Absolute Error

Mean absolute error (MAE)

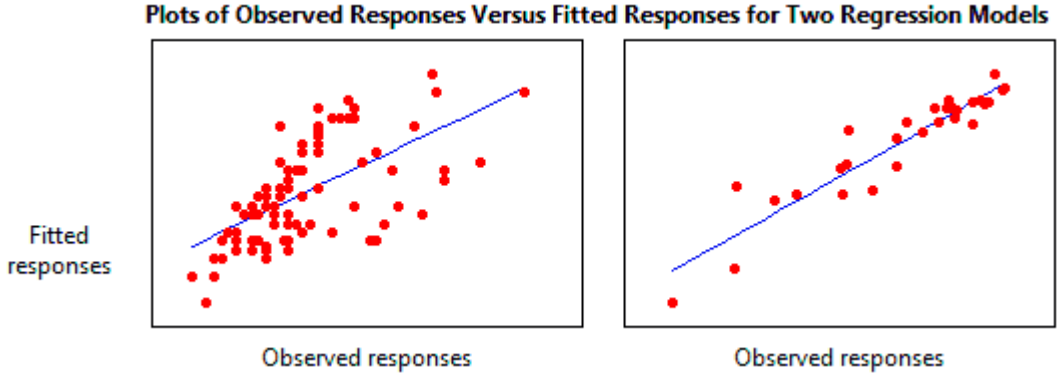
- Mean Absolute Error (MAE) is another the common error function used to measure how good is the linear regression model.





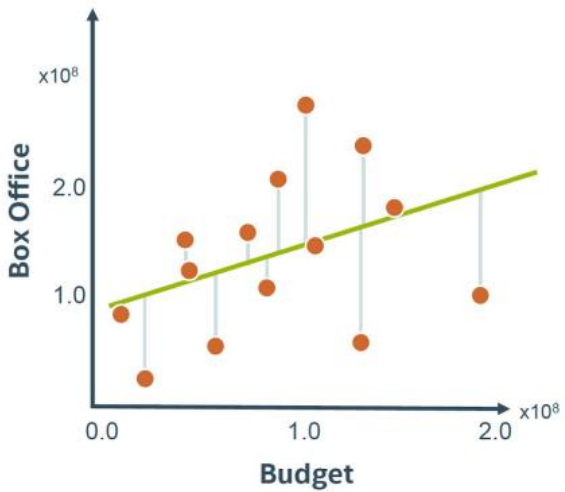
R-Squared

- R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.
- R-squared = Explained variation / Total variation
- R-squared is always between 0 and 100%
  - 0% indicates that the model explains none of the variability of the response data around its mean.
  - 100% indicates that the model explains all the variability of the response data around its mean.
  - In general, the higher the R-squared, the better the model fits your data.



Minimize the Mean Square Error

- Machine Learning aims to minimize the MSE to find the best linear regression model.



$$\min_{\beta_0,\beta_1} \frac{1}{m} \sum_{i=1}^{m} \left(\left(\beta_0 + \beta_1 x_{obs}^{(i)} \right) - y_{obs}^{(i)} \right)^2$$

Training a Linear Regression Model

- The most common technique used to train the linear regression equation from data is called **Ordinary Least Squares** or just Least Squares Regression.
- The Ordinary Least Squares procedure seeks to minimize the sum of the squared residuals.
- The approach uses linear algebra operations to estimate the optimal values for the coefficients of the linear equation.

Ordinary Least Square

Step 1: Calculate the mean of the x -values and the mean of the y -values.

$$\overline{X} = \frac{\sum_{i=1}^{n} x_i}{n}$$

$$\overline{Y} = \frac{\sum_{i=1}^{n} y_i}{n}$$

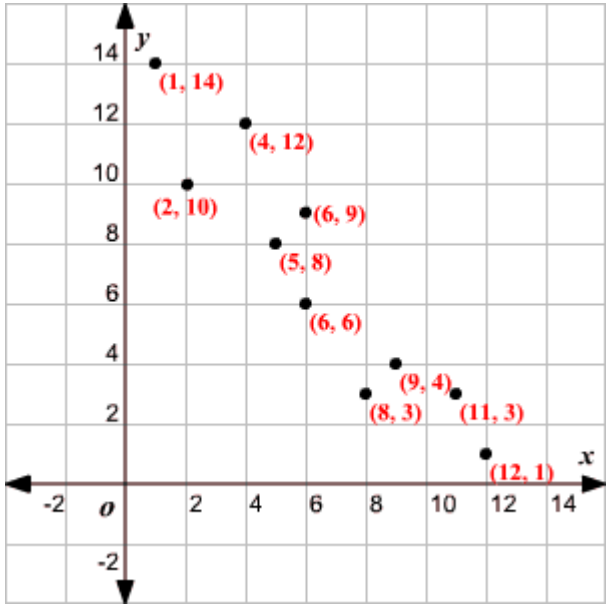
Step 2: The following formula gives the slope of the line of best fit:

$$m = \frac{\sum_{i=1}^{n} (x_i - \overline{X})(y_i - \overline{Y})}{\sum_{i=1}^{n} (x_i - \overline{X})^2}$$

Step 3: Compute the y -intercept of the line by using the formula:

$$b = \overline{Y} - m\overline{X}$$

Ordinary Least Square - Example



x	8	2	11	6	5	4	12	9	6	1
y	3	10	3	6	8	12	1	4	9	14

Calculate the means of the x -values and the y -values.

$$\overline{X} = \frac{8+2+11+6+5+4+12+9+6+1}{10} = 6.4$$

$$\overline{Y} = \frac{3+10+3+6+8+12+1+4+9+14}{10} = 7$$

Now calculate $x_i - \overline{X}$, $y_i - \overline{Y}$, $\left(x_i - \overline{X}\right)\left(y_i - \overline{Y}\right)$ and $\left(x_i - \overline{X}\right)^2$ for each i

Ordinary Least Square – Example (Cont'd)

i	x_i	y_i	$x_i - \overline{X}$	$y_i - \overline{Y}$	$(x_i - \overline{X})(y_i - \overline{Y})$	$(x_i - \overline{X})^2$
1	8	3	1.6	−4	−6.4	2.56
2	2	10	−4.4	3	−13.2	19.36
3	11	3	4.6	−4	−18.4	21.16
4	6	6	−0.4	−1	0.4	0.16
5	5	8	−1.4	1	−1.4	1.96
6	4	12	−2.4	5	−12	5.76
7	12	1	5.6	−6	−33.6	31.36
8	9	4	2.6	−3	−7.8	6.76
9	6	9	−0.4	2	−0.8	0.16
10	1	14	−5.4	7	−37.8	29.16
					$\sum_{i=1}^{n}(x_i - \overline{X})(y_i - \overline{Y}) = -131$	$\sum_{i=1}^{n}(x_i - \overline{X})^2 = 118.4$

Calculate the slope,

$$m = \frac{\sum_{i=1}^{n}(x_i - \overline{X})(y_i - \overline{Y})}{\sum_{i=1}^{n}(x_i - \overline{X})^2} = \frac{-131}{118.4} \approx -1.1$$

Calculate the y -intercept.

$$\begin{aligned} b &= \overline{Y} - m\overline{X} \\ &= 7 - (-1.1 \times 6.4) \\ &= 7 + 7.04 \\ &\approx 14.0 \end{aligned}$$

Use the slope and y -intercept to form the equation of the line of best fit.

The equation is y=−1.1x+14.0

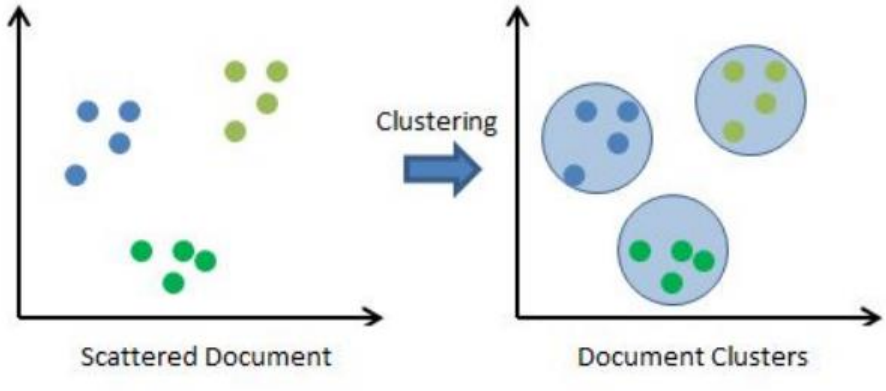
Many Types of Linear Regression

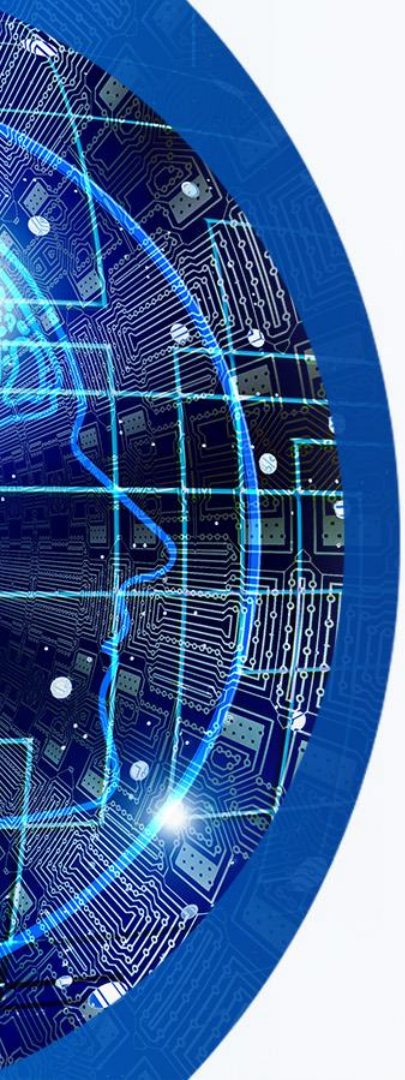
- **Linear regression** is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x).
- When there is a single input variable (x), the method is referred to as **simple linear regression**. When there are multiple input variables, it often refers to the method as **multiple linear regression**.

2.4 Clustering

What is Clustering?

- Clustering is the task of grouping a set of objects in such a way that objects in the same group (cluster) are more similar to each other than to those in other groups (clusters).
- Clustering is an unsupervised learning since no labels (targets) are needed for training.

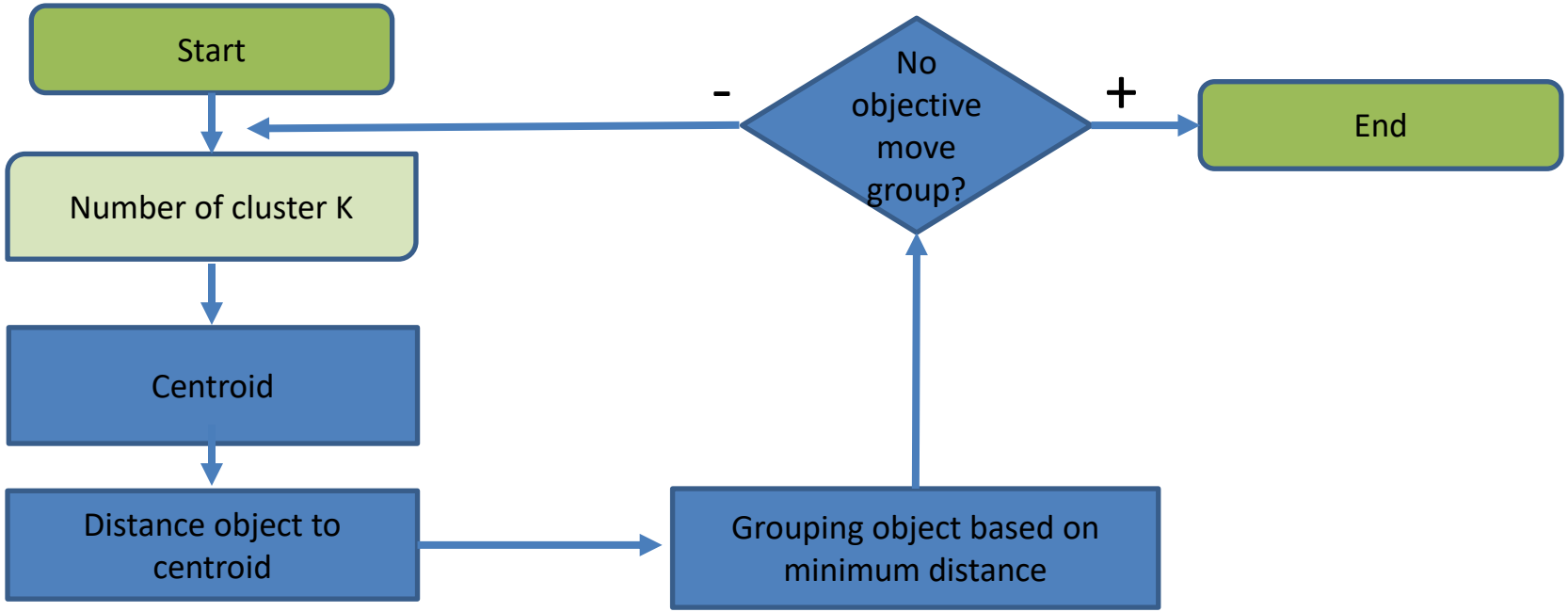




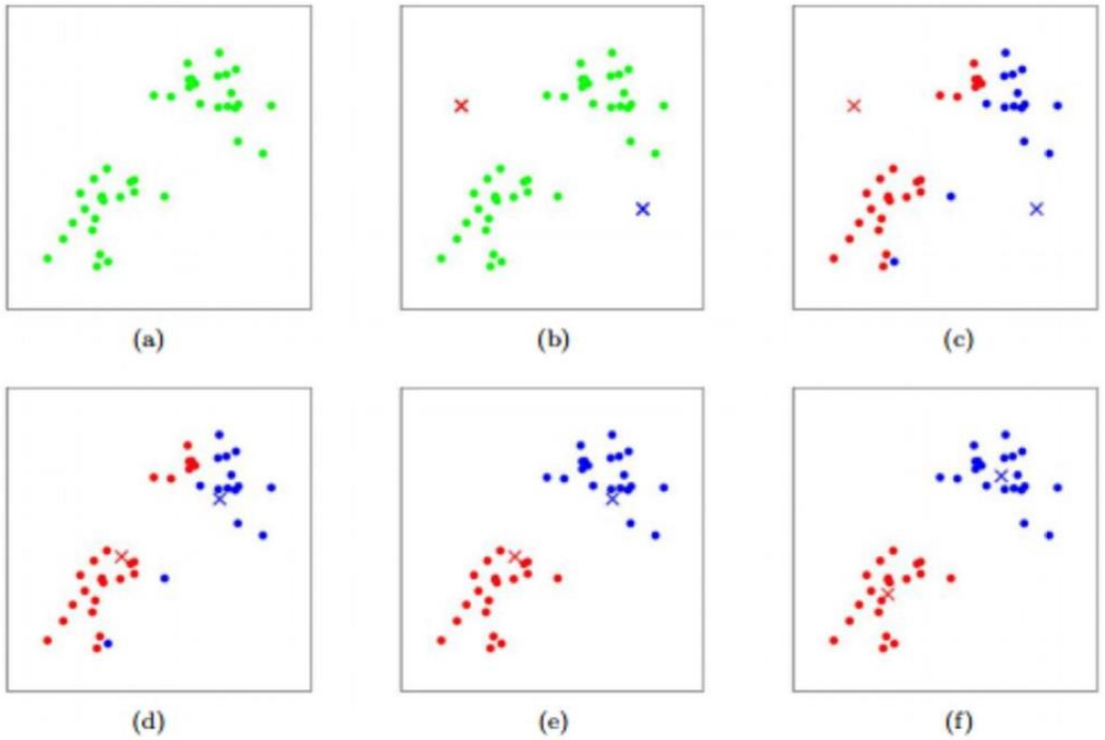
Key Clustering Algorithms

- K-Means Clustering
- Hierarchical Agglomerative Clustering

K-Mean Algorithm

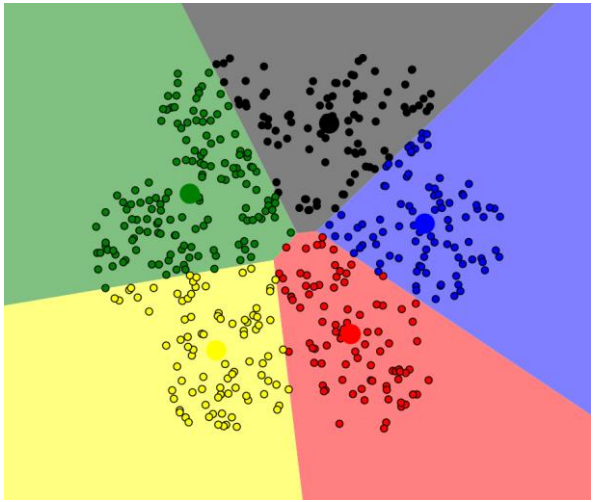


K-Mean Algorithm

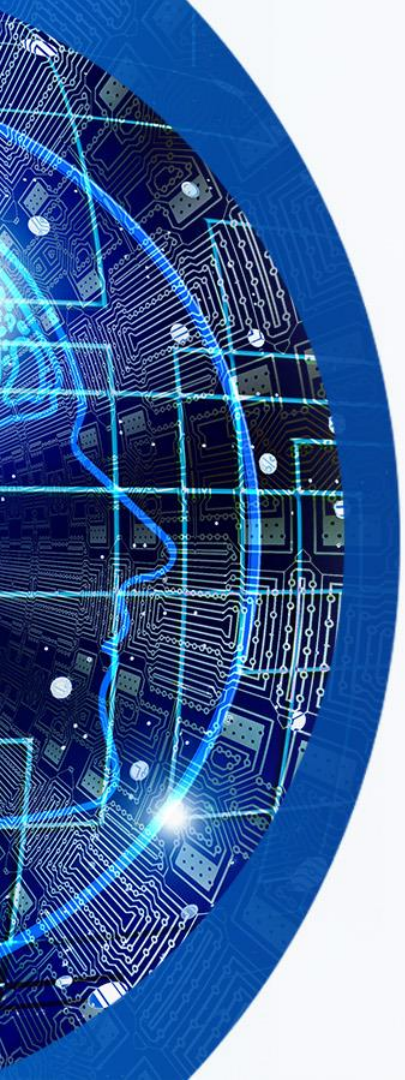


K-Mean Algorithm Simulation

Click on the images below to start k-mean clustering simulation



https://www.naftaliharris.com/blog/visualizing-k-means-clustering/



K-Mean Pros & Cons

Pros:

- Simple and fast, Easy to implement

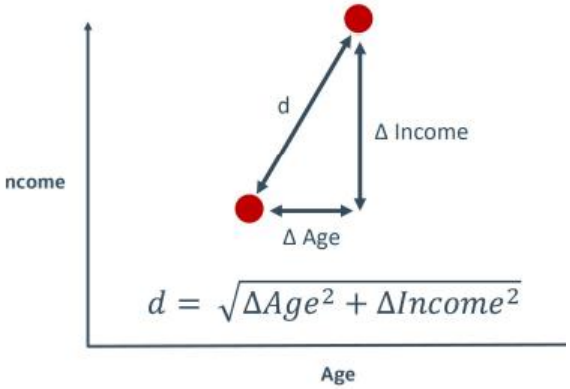
Cons:

- Needs to choose K
- Sensitive to outliers
- Prone to local minima.
- Extremely sensitive to initialization.
- Bad initialization can lead to:
  - poor convergence speed
  - bad overall clustering

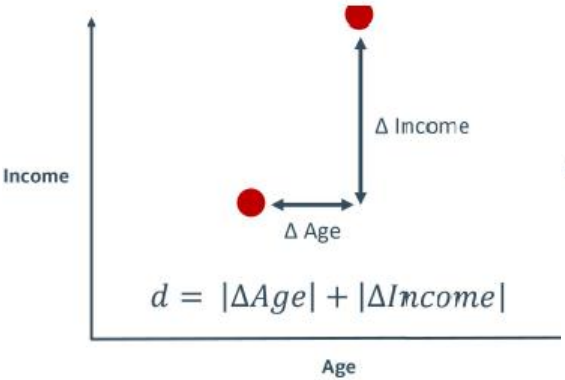
Distance Metric Choice

Choice of distance will significantly impact the Clustering algorithms

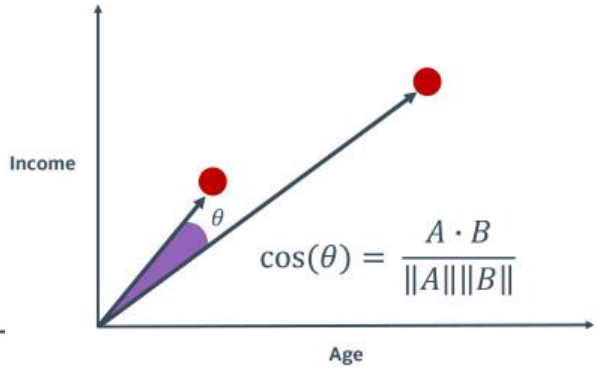
Enclidean (L2)



Mantattab (L1)

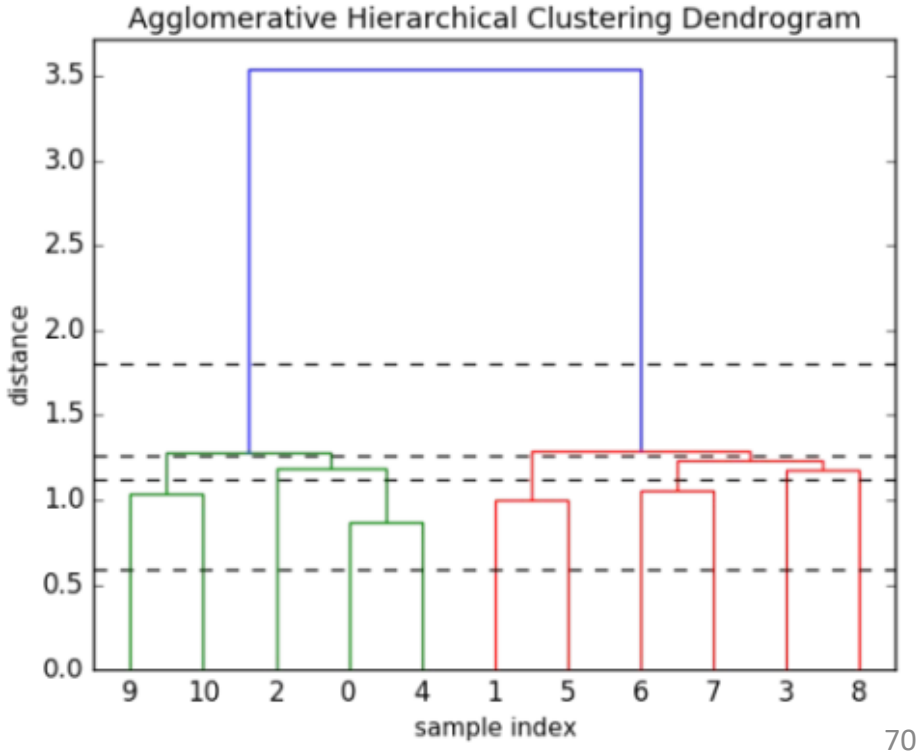


Cosine



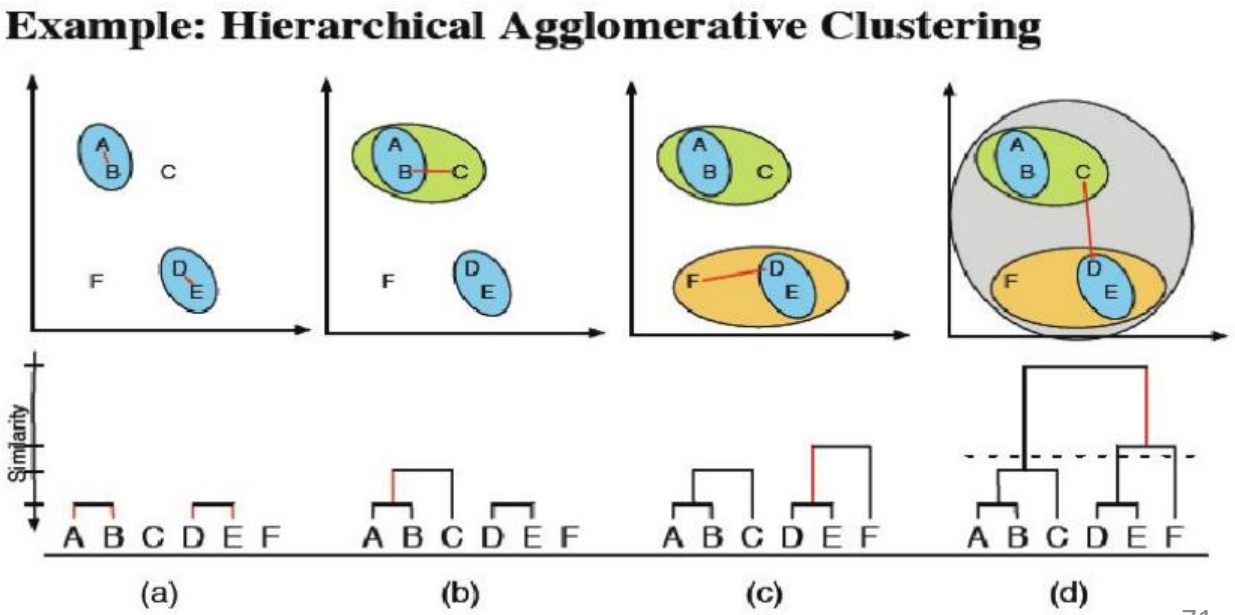
Hierarchical Clustering

- Hierarchical clustering is where you build a cluster tree (a dendrogram) to represent data, where each group (or “node”) links to two or more successor groups.
- The groups are nested and organized as a tree, which ideally ends up as a meaningful classification scheme.

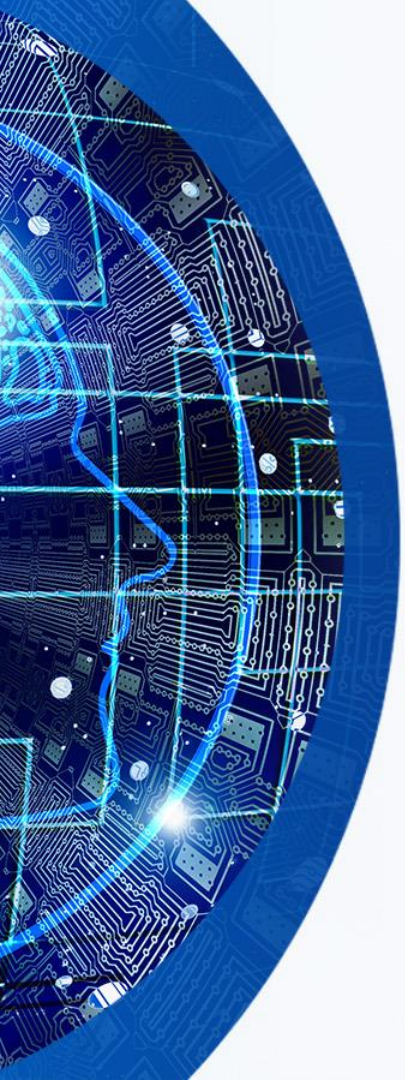


Hierarchical Clustering

- Clusters are consecutively merged with the most nearby clusters. The length of the vertical dendrogram lines (linkage) reflect the nearness

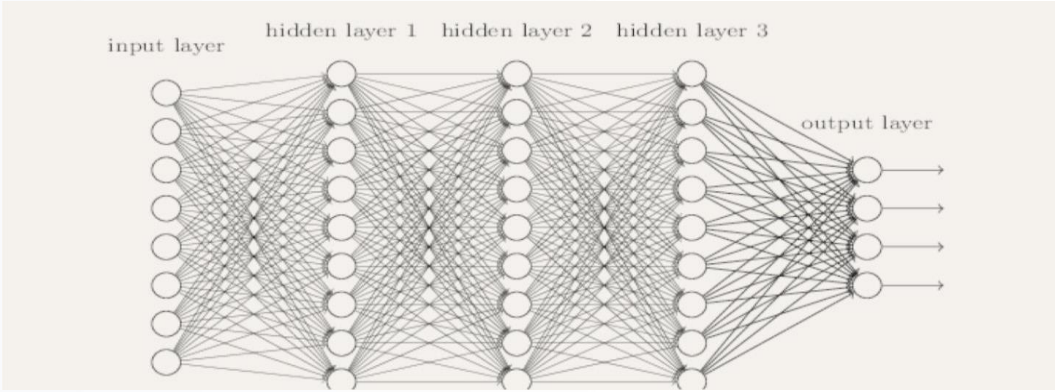


2.5 Artificial Neural networks



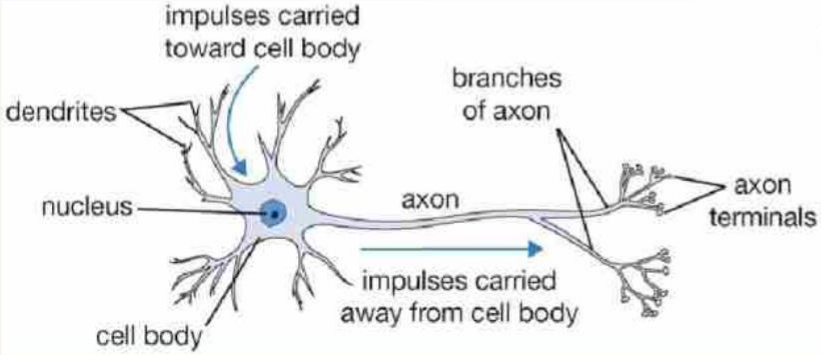
Neural Network (NN)

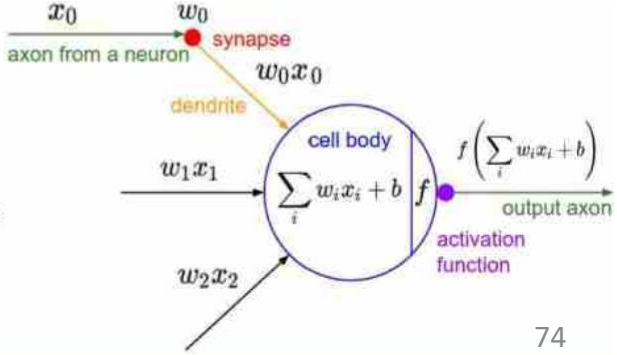
- Neural Networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns
- NN is made up of neurons. It consists of input layer, multiple hidden layers and output layers. The simplest NN is one hidden layer



What is a Neuron?

- A biological neuron has dendrites to receive signals, a cell body to process them, and an axon to send signals out to other neurons
- An artificial neuron has a number of input channels, a processing stage, and one output that can fan out to multiple other artificial neurons.
- The weighted sum of inputs + bias is feed through a non-linear activation function to output a value to next neurons.



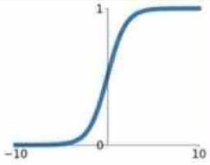


Activation Functions

Activation functions allow Neural Networks to learn complicated and Non-linear complex functional mappings between the inputs and outputs

Sigmoid

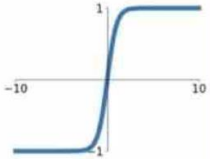
$\sigma(x) = \frac{1}{1+e^{-x}}$



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

tanh

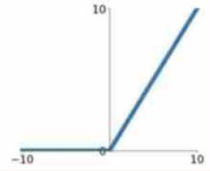
$\tanh(x)$



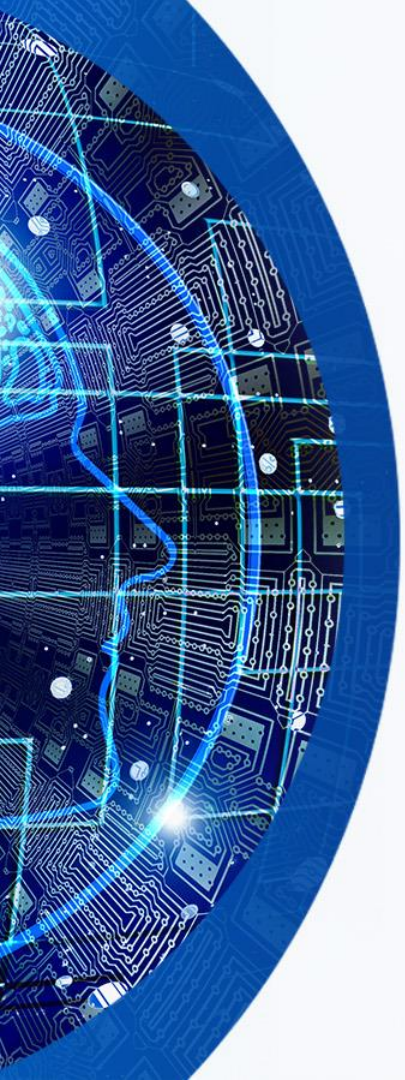
$$\sigma(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$$

ReLU

$\max(0, x)$



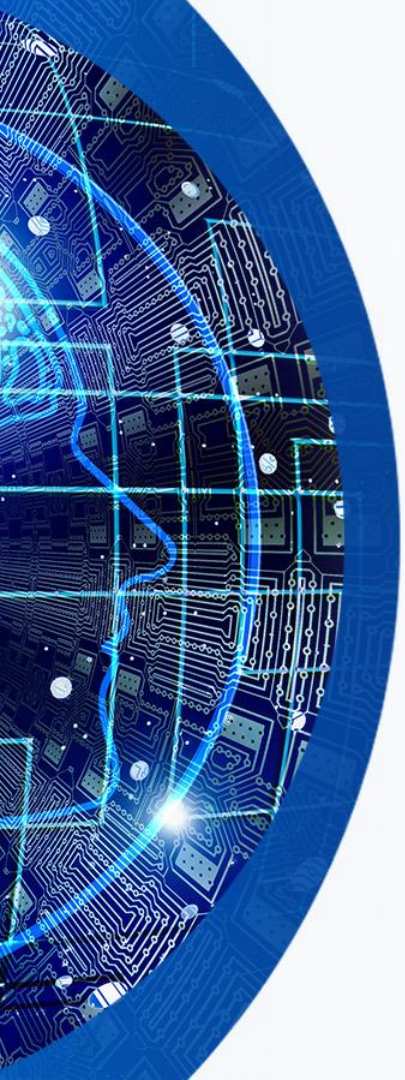
$$\sigma(z) = max(0, z)$$



Why Activation?

- Introduce non-linear properties to our Network: Make sense of something really complicated and Non-linear complex functional mappings between the inputs and response variable.

- Be differentiable: To perform backpropagation optimization strategy while propagating backwards in the network to compute gradients of Error(loss) with respect to Weights and then accordingly optimize weights using Gradient descend or any other Optimization technique to reduce Error.



Why not just Linear?

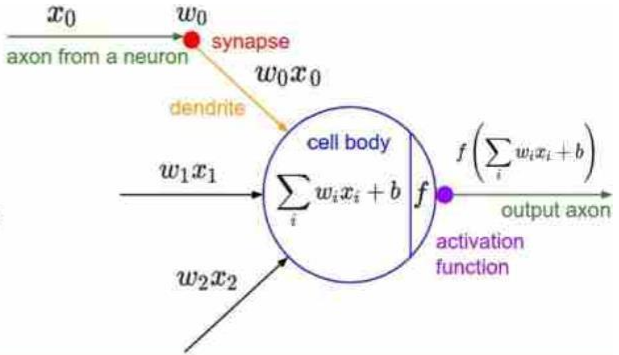
- If we do not apply a Activation function then the output signal would simply be a simple linear function.
- A linear function is just a polynomial of one degree.
- A linear equation is easy to solve but they are limited in their complexity and have less power to learn complex functional mappings from data.
- A Neural Network without Activation function would simply be a Linear regression Model, which has limited power and does not performs good most of the times

How is a Neural Network Trained?

- A neural network is trained in three steps:
  - Feed forward algorithm
  - Back propagation algorithm
  - Optimization algorithm (Optimizer)

Feed forward algorithm

- Feed forward algorithm represents the aspect of how input signals travel through different neurons present in different layers in form of weighted sums and activations, and, result in output / prediction. The key aspect in feed forward algorithm is activation function.



https://vitalflux.com/different-types-activation-functions-neural-networks/

Back propagation algorithm

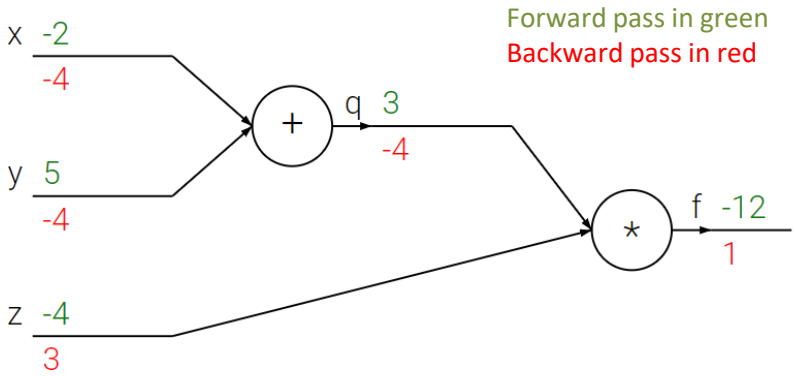
- Back propagation algorithm represents the manner in which gradients are calculated on output of each neuron going backwards (using chain rule)
- The goal is to determine changes which need to be made in weights in order to achieve the neural network output closer to actual output.
- Note that back propagation algorithm is only used to calculate the gradients.

Intuitive Understanding of Backpropagation

Let's say we have a function,

$$f(x, y, z) = (x + y)z$$

expressed in a circuit below,



f(x,y,z) is "happy" when the output is as high as possible. *How do we "teach" it to produce a high output? Answer: is to look at the gradient at each one of the gates.*

$$f = qz, where\ q = x + y$$

Let's compute the local gradient on *f:*

$$\frac{df}{dq} = z \qquad \frac{df}{dz} = q$$

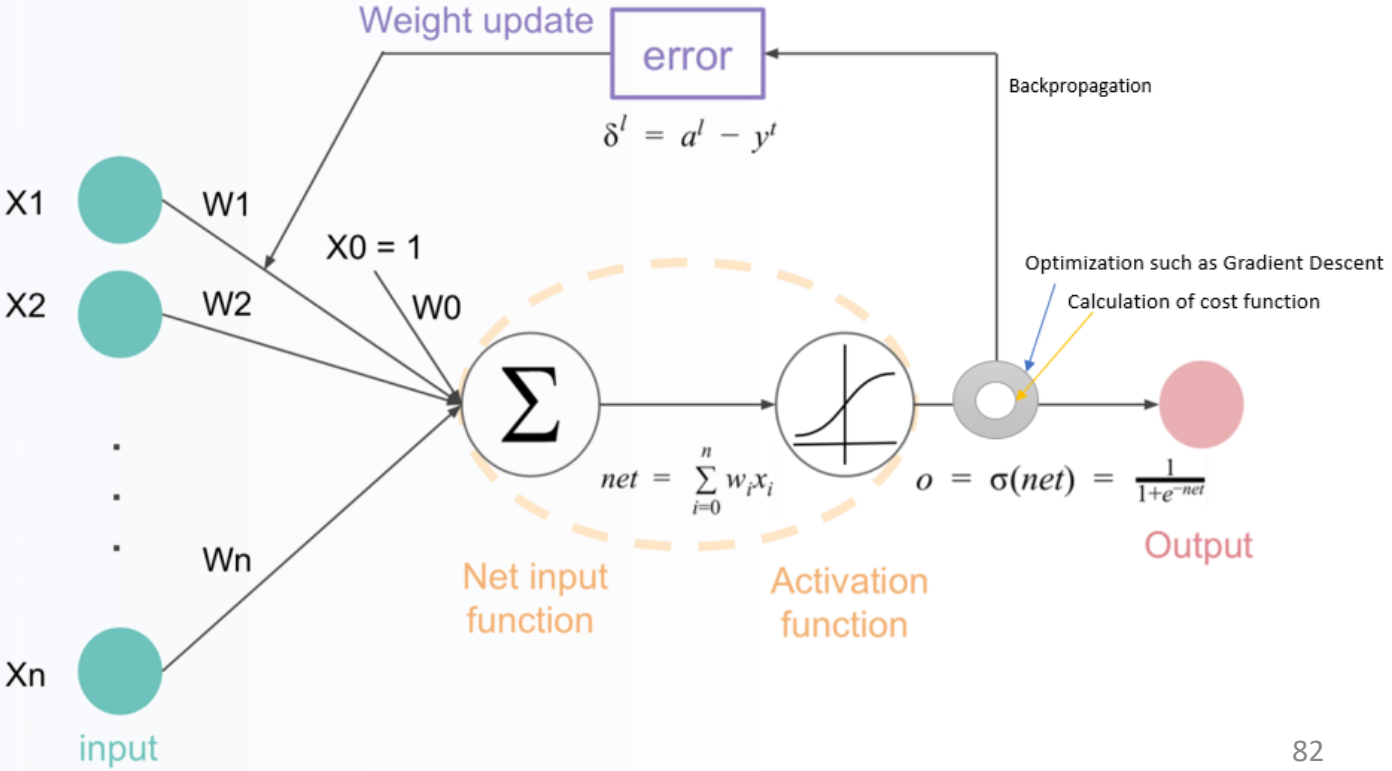
Partial derivatives of *f* with respect to *q* and with respect to *z*

Using the chain rule,

$$\frac{df}{dx} = \frac{df}{dq} \cdot \frac{dq}{dx}$$

the gradient of the output w.r.t. the input can be calculated by multiplying a series of local gradient as opposed to calculating the gradient of *f* with respect to its inputs *x,y,z,* hence the word, "chain".

Backpropagation in Neural Network



Optimization algorithm (Optimizer)

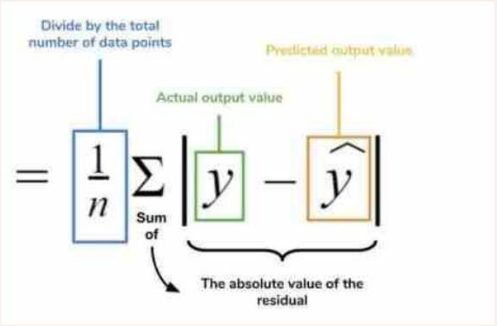
- Once the gradients are determined, the final step is to use appropriate optimization algorithm to update the weights using the gradients calculated by the back propagation algorithm.
- The fraction or amount (learning rate) of the weight adjustment, is how much the weight's gradient to be subtracted from the weight.
- To guide the optimization algorithm (learning) towards the right direction, a loss function is used.
- Note: Optimization is hard

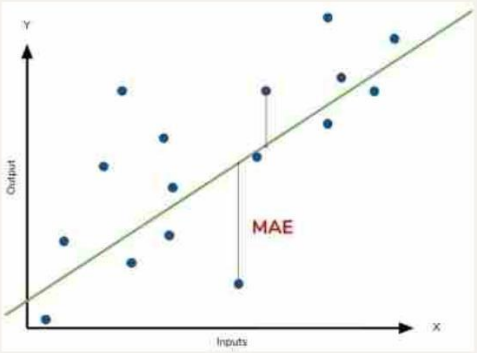
Loss Function (aka cost, objective) - MAE

Mean absolute error (MAE)

While the MAE is easily interpretable, using the absolute value of the residual often is not as desirable as squaring this difference.

Depending on how you want your model to treat outliers, or extreme values, in your data, you may want to bring more attention to these outliers or downplay them. The issue of outliers can play a major role in which error metric you use.





Loss Function (aka cost, objective) – MSE

Mean square error (MSE)

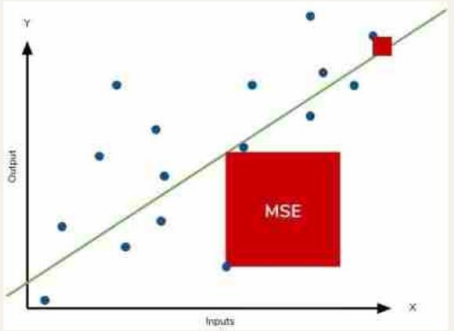
The effect of the square term in the MSE equation is most apparent with the presence of outliers in our data.

While each residual in MAE contributes proportionally to the total error, the error grows quadratically in MSE.

This means that outliers in our data will contribute to much higher total error in the MSE than they would the MAE.

Similarly, our model will be penalized more for making predictions that differ greatly from the corresponding actual value.

$$= \frac{1}{n}\Sigma\underbrace{\left(y - \hat{y}\right)^2}_{\text{The square of the difference between actual and predicted}}$$

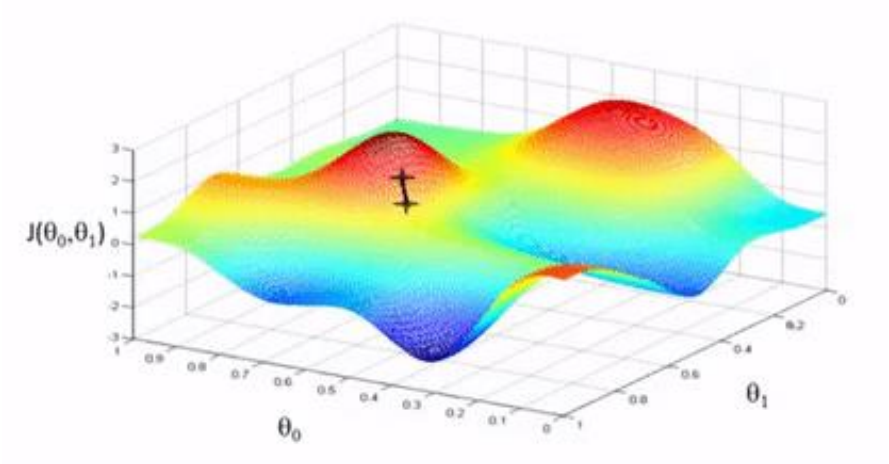


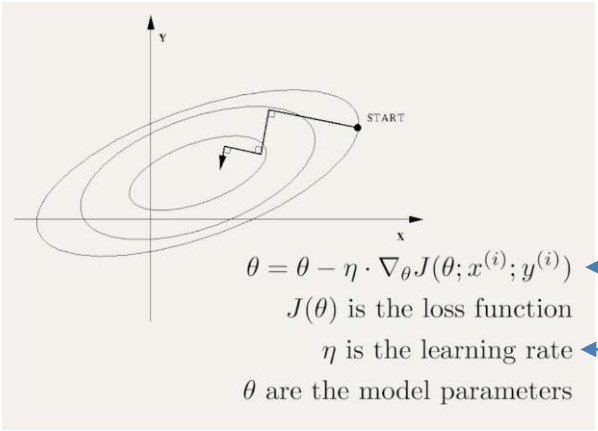
Loss Function (aka cost, objective) - RMSE

- Root mean squared error (RMSE) is the square root of the MSE. Because the MSE is squared, its units do not match that of the original output. Researchers will often use RMSE to convert the error metric back into similar units, making interpretation easier.

- Since the MSE and RMSE both square the residual, they are similarly affected by outliers. The RMSE is analogous to the standard deviation (MSE to variance) and is a measure of how large your residuals are spread out.

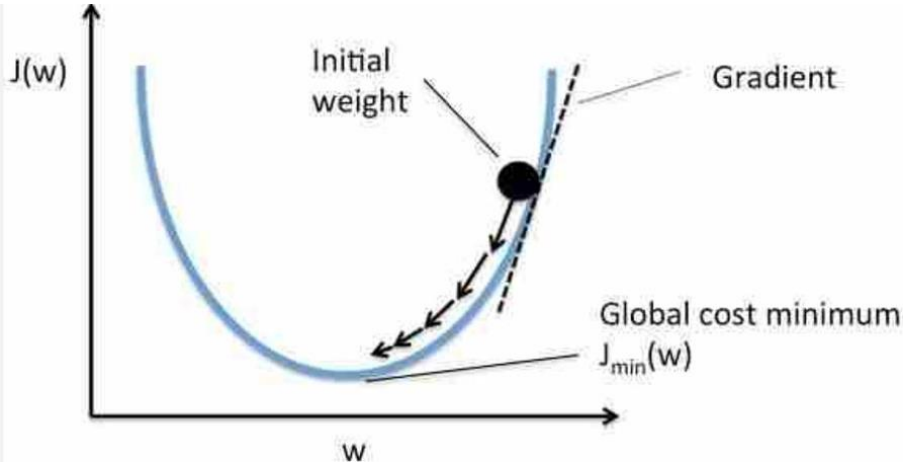
Gradient Descent

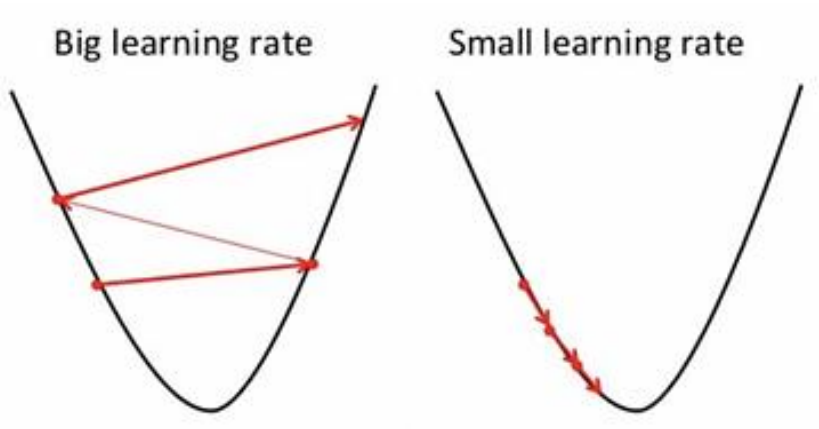




- What is a gradient? A gradient measures how much the output of a function changes if you change the inputs a little bit – Lex Fridman (MIT). In gradient descent, the aim is to get to the bottom of the graph to a point where it can no longer move downhill – a local minimum.
- Gradient descent computes the gradient of the cost function w.r.t. to the parameters θ for the entire training dataset: $\theta = \theta - \eta \cdot \nabla_\theta J(\theta)$. It is a way to minimize an objective function $J(\theta)$ parameterized by a model's parameters $\theta \in \mathbb{R}^d$ by updating the parameters in the opposite direction of the gradient of the objective function $\nabla_\theta J(\theta)$ w.r.t. to the parameters.

Learning Rate





- **Learning Rate**: How big the steps are gradient descent takes into the direction of the local minimum, by changing (fast or slow) the weights.
- For gradient descent to reach the local minimum we must set the learning rate to an appropriate value, which is neither too low nor too high.

Gradient Descent Optimizers

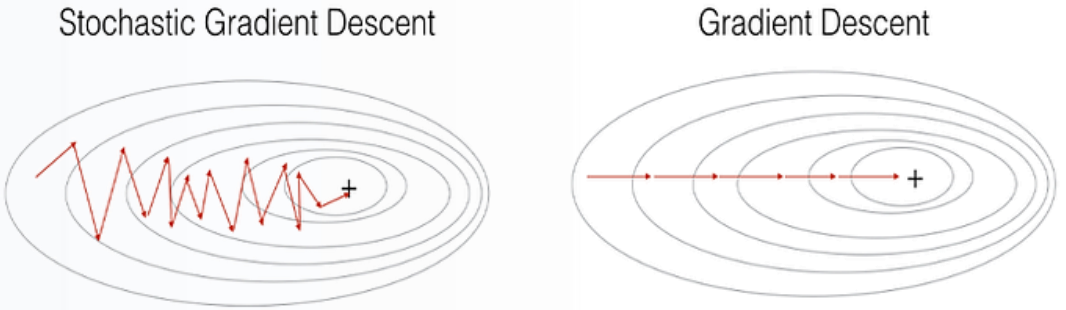
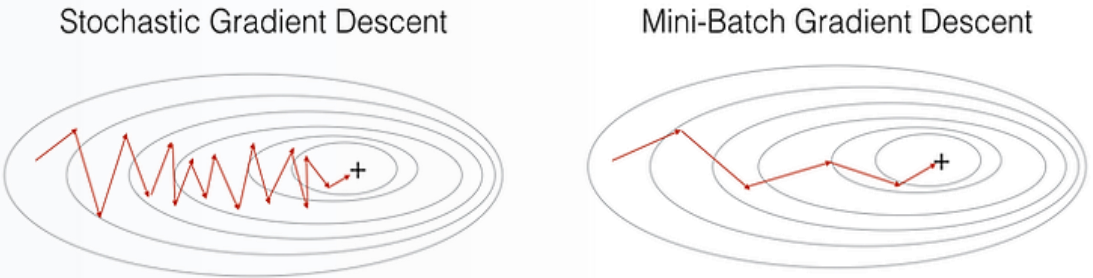
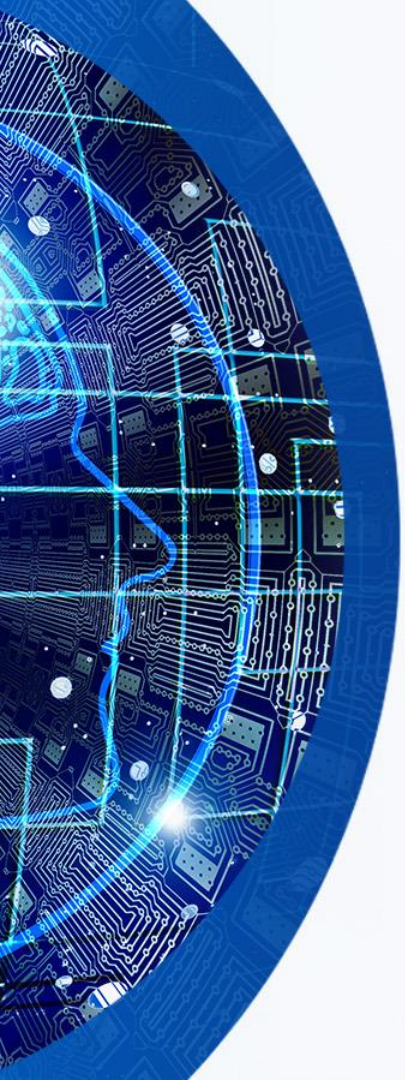


Figure 1: SGD vs GD

"+" denotes a minimum of the cost. SGD leads to many oscillations to reach convergence. But each step is a lot faster to compute for SGD than for GD, as it uses only one training example (vs. the whole batch for GD).

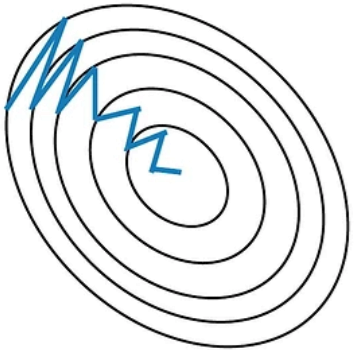


MB-SGD algorithm is an extension of the SGD algorithm and it overcomes the problem of large time complexity in the case of the SGD algorithm. MB-SGD algorithm takes a batch of points or subset of points from the dataset to compute derivate.



Stochastic Gradient Descent with Momentum

Stochastic Gradient Descent **without** Momentum

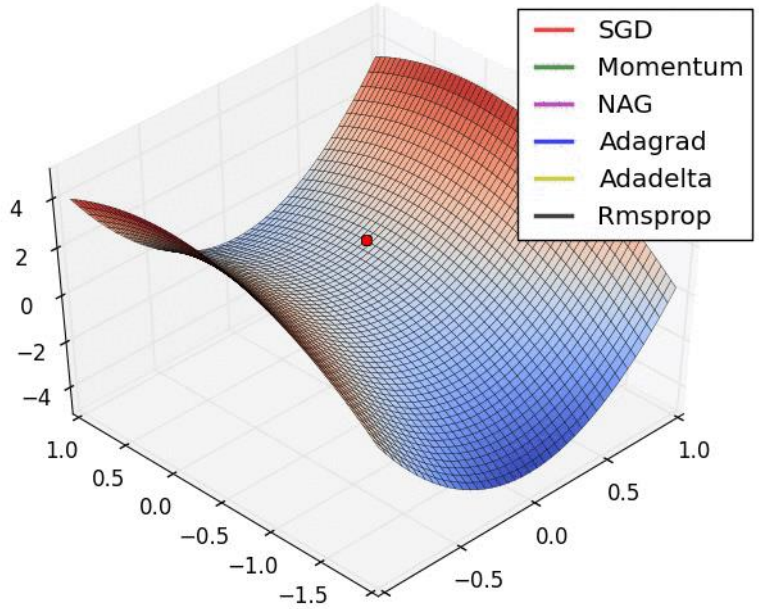


Stochastic Gradient Descent **with** Momentum

Momentum at time ‘t’ is computed using all previous updates giving more weightage to recent updates compared to the previous update. This leads to speed up the convergence (reduce oscillation).

Other Popular Optimizers

- Gradient Descent
- SGD (stochastic gradient descent)
- Mini-Batch SGD (MB-SGD)
- Adam
- RMSProp (Root Mean Squared Propagation)
- Adagrad (Adaptive Gradient)
- Adadelta
- Momentum (SGD with momentum)
- NAG (Nesterov accelerated gradient)



Saddle-point oscillation

Which optimizer should we use?

The question was to choose the best optimizer for our Neural Network Model in order to converge fast and to learn properly and tune the internal parameters so as to minimize the Loss function.

Adam works well in practice and outperforms other Adaptive techniques. Sparse data sets → adaptive learning-rate methods. (no need to adjust the learning rate but likely achieve the best results with the default value.)

Wants fast convergence and train a deep Neural Network Model or a highly complex Neural Network → Adam or any other Adaptive learning rate techniques (they outperforms every other optimization algorithms.)

Summary

- Three types of machine learning algorithms
- Two categories of supervised ML problems
- What type of problem might be solved with ML solution

Quiz III

1. When updating neural network weights using the loss function, what dictates how much change the weights should have?

A. Batch size
B. Learning rate
C. Initial weights
D. Bias term

Quiz III

2. A real estate company is building a linear regression model to predict housing prices Which of the following is NOT a good metric to measure performance of their regression model?

A. R-Squared value
B. F1 score
C. Mean-squared error
D. Mean absolute error

Quiz III

3. A movie streaming company is trying to tag each movie as it's added into different genres. They have historical information about the description of the movie along with other reviewed articles. What type of ML can they use to for this problem?

A. Binary classification
B. Multi-class classification
C. Regression
D. Reinforcement learning

End of Chapter 2

Q&A