MACHINE LEARNING NOTES

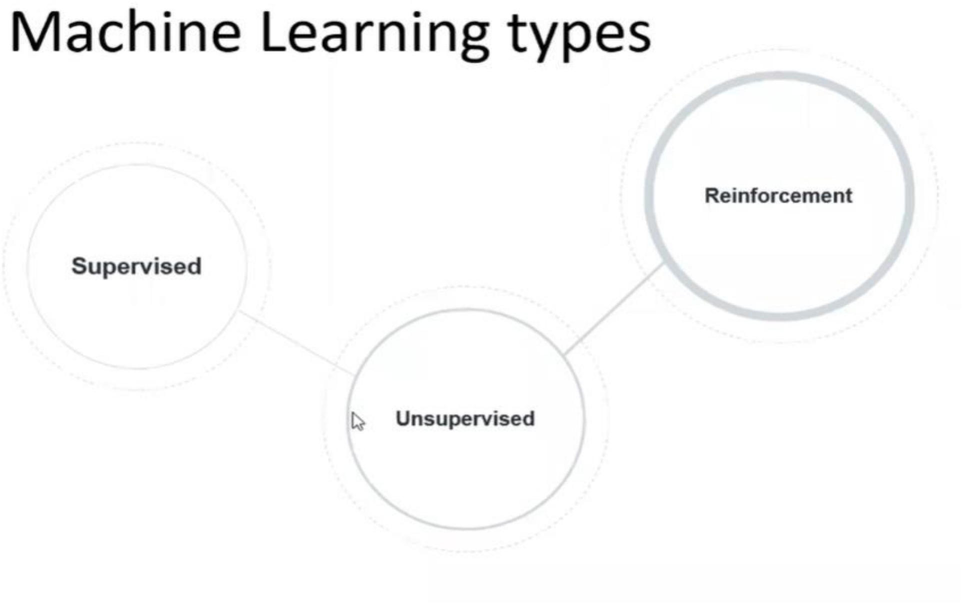
What is machine learning?

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed.

It deals with the algorithms which replicates the behaviour of decision-making process.

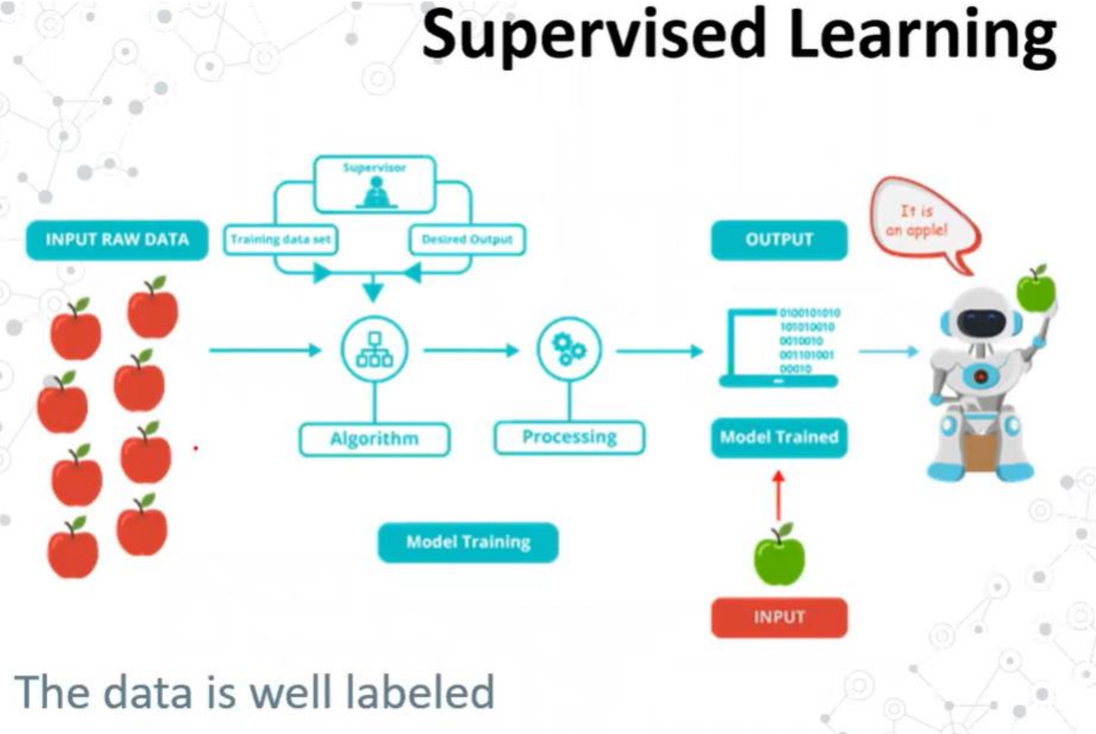
What are the Applications of Machine Learning?

- Image recognition
- Speech recognition
- Medical diagnosis
- Statistical Arbitrage
- Learning Associations
- Classification
- Prediction
- Extraction
- Regression
- Financial services



What are the types of Machine Learning?

1. Supervised Learning, in which the training data is labelled with the correct answers, e.g., “spam” or “ham.” The two most common types of supervised learning are classification (where the outputs are discrete labels, as in spam filtering) and regression (where the outputs are real-valued).



Supervised Learning

INPUT RAW DATA

Supervisor

Training data set

Desired Output

Algorithm

Processing

Model Trained

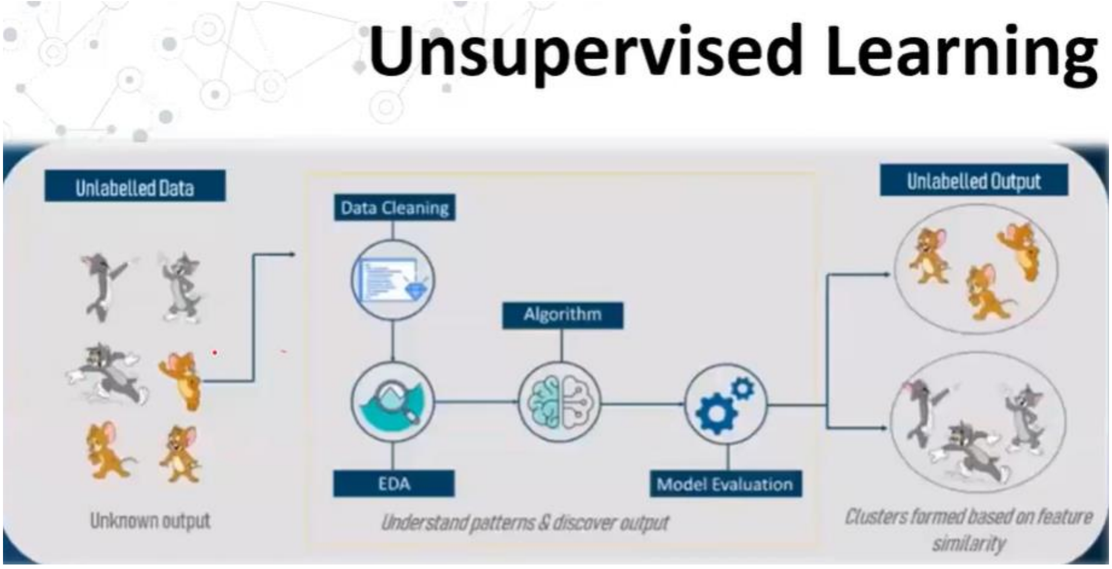
OUTPUT

Model Training

INPUT

The data is well labeled

2. Unsupervised learning, in which we are given a collection of unlabelled data, which we wish to analyse and discover patterns within. The two most important examples are dimension reduction and clustering.



Unsupervised Learning

Unlabelled Data

Unknown output

Data Cleaning

EDA

Algorithm

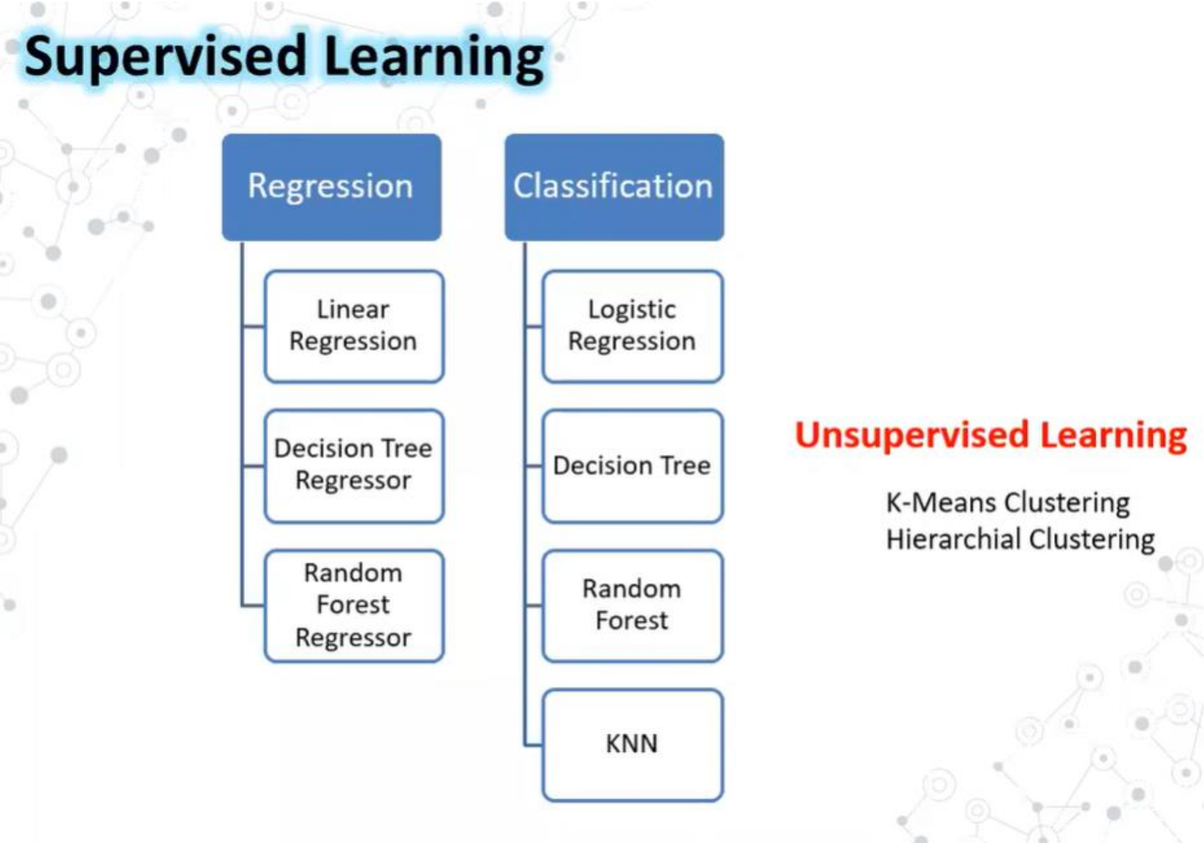
Model Evaluation

Understand patterns & discover output

Unlabelled Output

Clusters formed based on feature similarity

3. Reinforcement learning, in which an agent (e.g., a robot or controller) seeks to learn the optimal actions to take based on the outcomes of past actions.



Supervised Learning

Regression

- Linear Regression
- Decision Tree Regressor
- Random Forest Regressor

Classification

- Logistic Regression
- Decision Tree
- Random Forest
- KNN

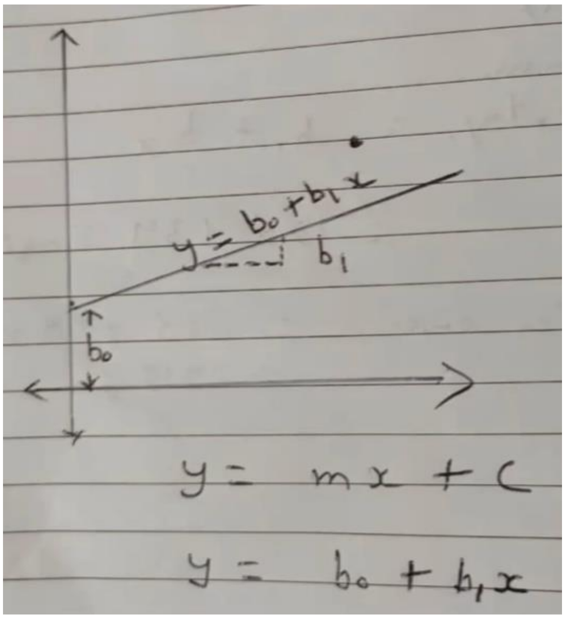
Unsupervised Learning

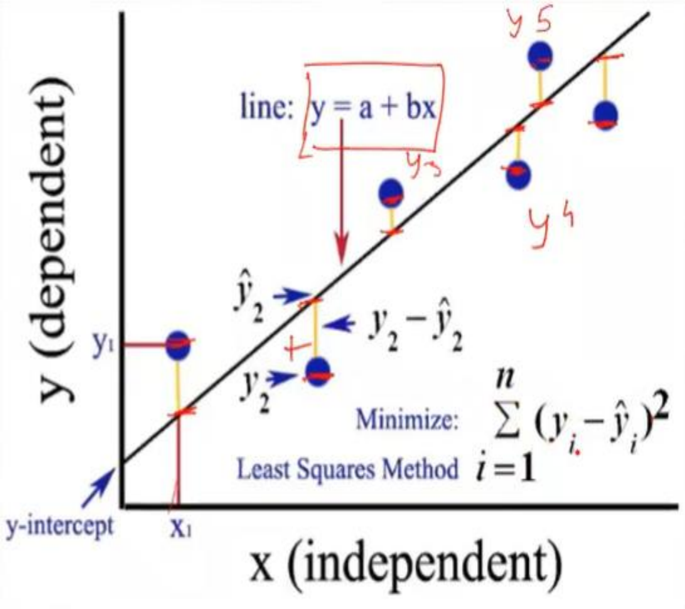
K-Means Clustering
Hierarchial Clustering

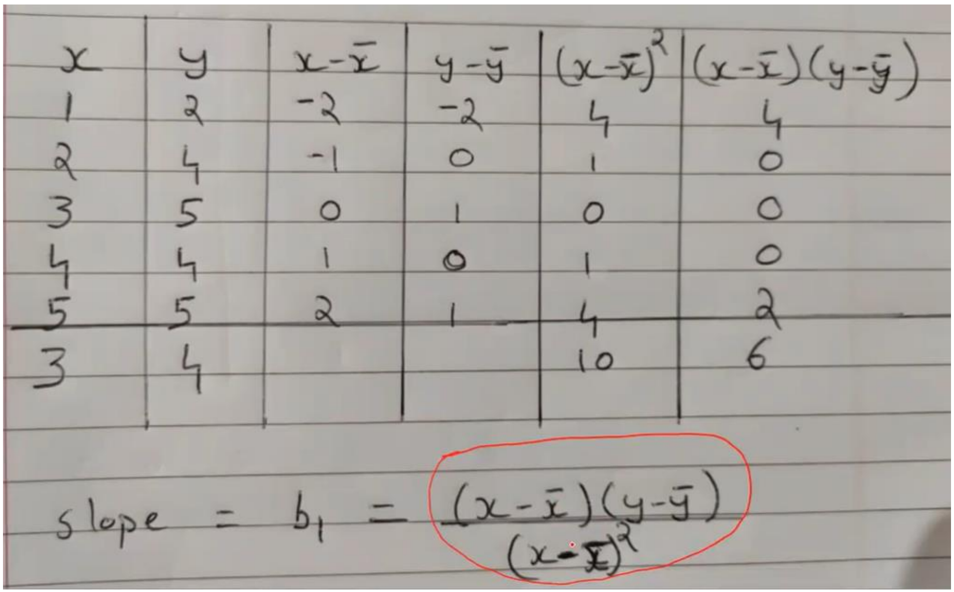
❖ LINEAR REGRESSION

Linear regression is a popular regression learning algorithm that learns a model which is a linear combination of features of the input example.

1. Least square method

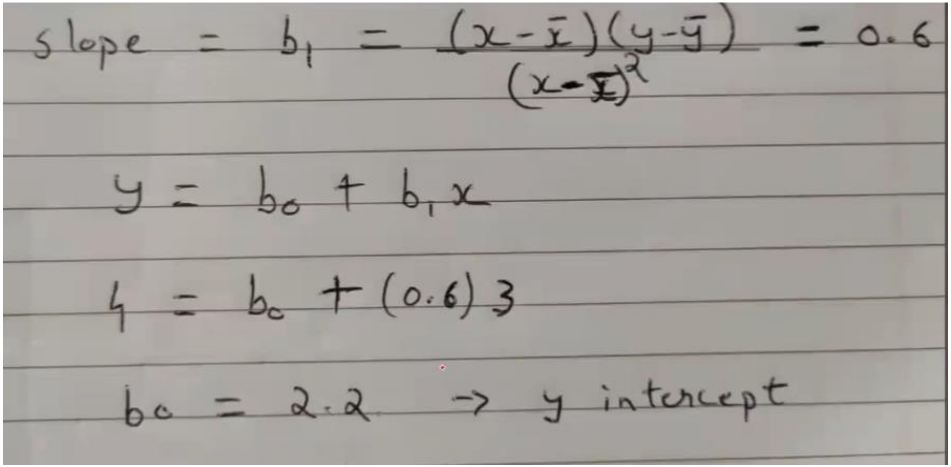






x	y	$x-\bar{x}$	$y-\bar{y}$	$(x-\bar{x})^2$	$(x-\bar{x})(y-\bar{y})$
1	2	-2	-2	4	4
2	4	-1	0	1	0
3	5	0	1	0	0
4	4	1	0	1	0
5	5	2	1	4	2
3	4			10	6

$$\text{slope} = b_1 = \boxed{\frac{(x-\bar{x})(y-\bar{y})}{(x-\bar{x})^2}}$$

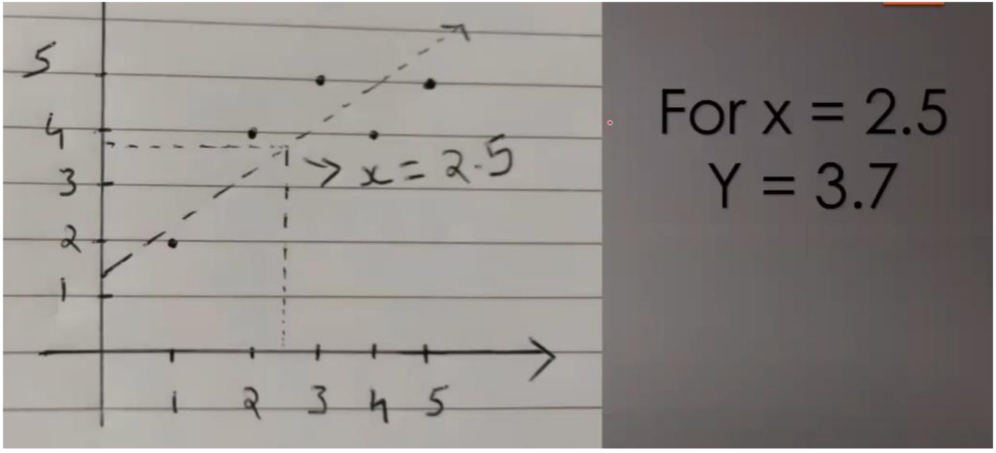


$$\text{slope} = b_1 = \frac{(x-\bar{x})(y-\bar{y})}{(x-\bar{x})^2} = 0.6$$

$$y = b_0 + b_1 x$$

$$4 = b_0 + (0.6)3$$

$$b_0 = 2.2 \rightarrow y \text{ intercept}$$



For x = 2.5
Y = 3.7

Sample data: https://colab.research.google.com/drive/1om1Ol7MuVwyEUi1ITB19KTCnmSGLf9X3#scrollTo=OLlpncchrdCm

Salary Data:

https://colab.research.google.com/drive/1WFhMCqxApoxVXuUmEf7xtoJVtYKUfSV-#scrollTo=CkSY4ZUOH2yZ

Advertising data: https://colab.research.google.com/drive/1-WzpBp5SND-i1drSdUTzxtwDniGYr0A6

2. Gradient Descent

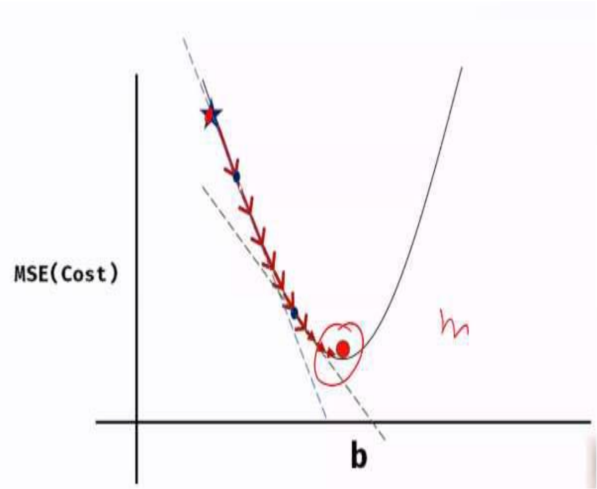
Gradient descent can be used to find optimal parameters for linear and logistic regression, SVM and also neural networks. However, the optimization criterion will have two parameters: *w* and *b*. The extension to multi-dimensional training data is straightforward: you have variables *w*(1), *w*(2), and *b* for two-dimensional data, *w*(1), *w*(2), *w*(3), and *b* for three-dimensional data and so on.
The linear regression model looks like this: *f*(*x*) = *wx* + *b*. We don't know what the optimal values for *w* and *b* are and we want to learn them from data. To do that, we look for such values for *w* and *b* that minimize the mean squared error:

$$l = \frac{1}{N}\sum_{i=1}^{N}(y_i - (wx_i + b))^2.$$

Gradient descent starts with calculating the partial derivative for every parameter

$$\frac{\partial l}{\partial w} = \frac{1}{N}\sum_{i=1}^{N} -2x_i(y_i - (wx_i + b));$$

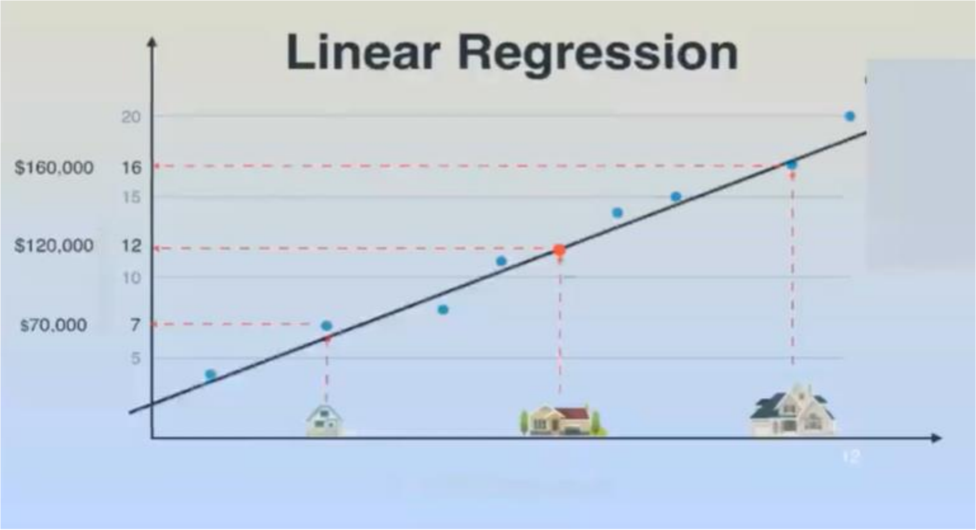
$$\frac{\partial l}{\partial b} = \frac{1}{N}\sum_{i=1}^{N} -2(y_i - (wx_i + b)).$$



BASIC CODE:https://colab.research.google.com/drive/10y3G7KuUBEmhuXOssL6R69b5jOqtUz-z#scrollTo=OeFSXgByEeKL

- Multiple linear regression

  House prices prediction: https://colab.research.google.com/drive/1sv6iKZY5dZSkWxZIPkXiq7DfIhvrz834#scrollTo=xPxypBZHL3-o

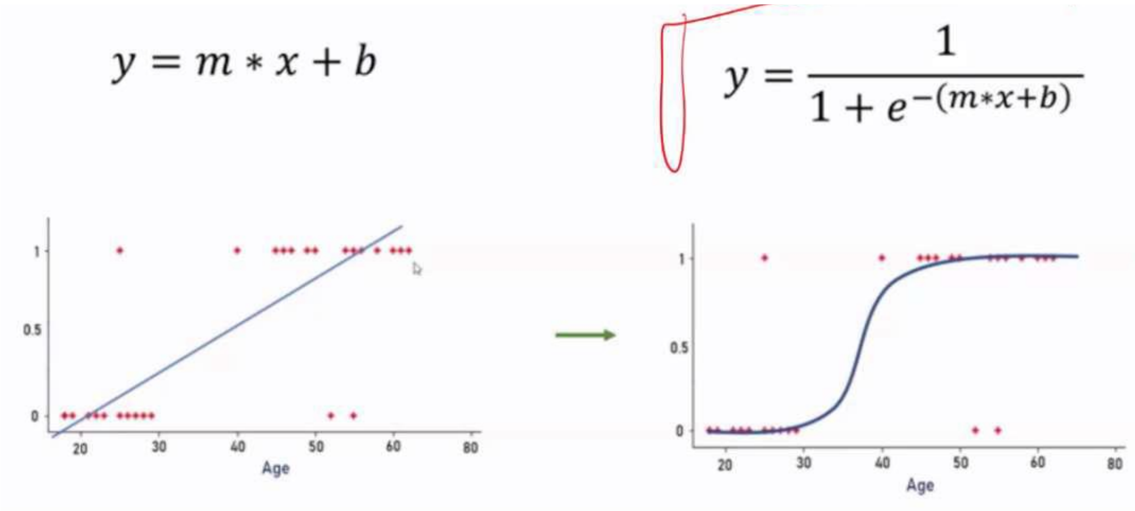


- ❖ **LOGISTIC REGRESSION**

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic **sigmoid** function to return a probability value which can then be mapped to two or more discrete classes

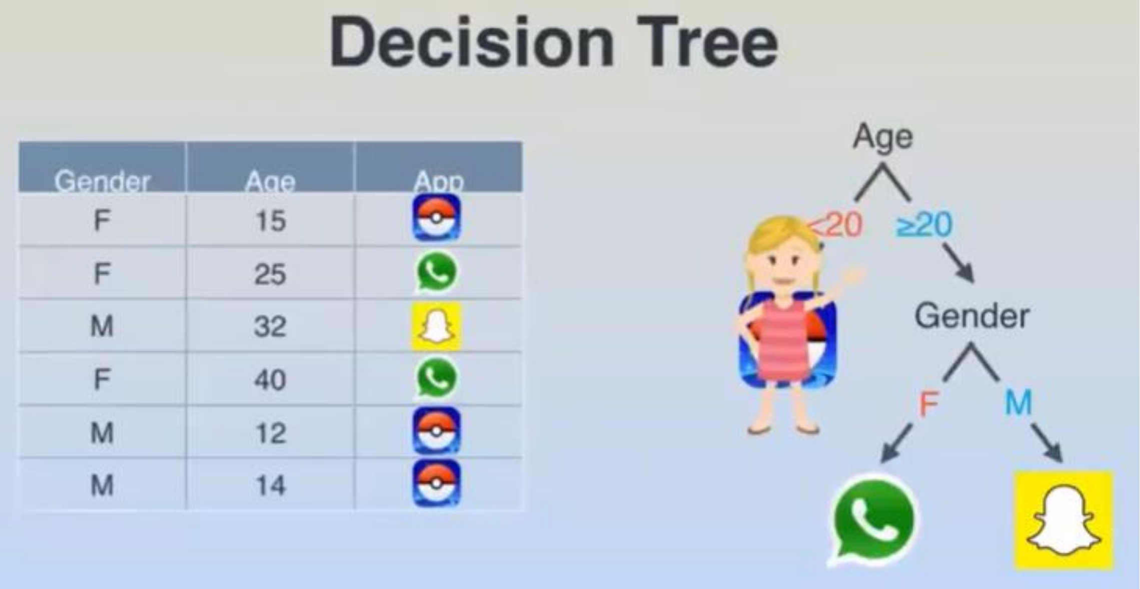
- **Linear Regression** could help us predict the student's test score on a scale of 0 - 100. Linear regression predictions are continuous (numbers in a range).
- **Logistic Regression** could help use predict whether the student passed or failed. Logistic regression predictions are discrete (only specific values or categories are allowed). We can also view probability scores underlying the model's classifications

Loan data:https://colab.research.google.com/drive/1BXV3G-ihTpxzSWIhrQUJg6S3MgtodqC-#scrollTo=EkCMdKVqUxqC



❖ DECISION TREE

A decision tree is an acyclic graph that can be used to make decisions. In each branching node of the graph, a specific feature *j* of the feature vector is examined. If the value of the feature is below a specific threshold, then the left branch is followed; otherwise, the right branch is followed. As the leaf node is reached, the decision is made about the class to which the example belongs.
Example:



Decision Tree

Gender	Age	App
F	15	◓
F	25	☎
M	32	👻
F	40	☎
M	12	◓
M	14	◓

Decision trees are very sensitive to even small changes in the data.
We use mean for regression trees and mode for classification trees.
This is where the bagging and Random Forest comes into play. Even with Bagging, the decision trees can have a lot of structural similarities and in turn have high correlation in their predictions.
Random forest changes the algorithm for the way that the sub-trees are learned so that the resulting predictions from all of the subtrees have less correlation.

Income class data:
https://colab.research.google.com/drive/1AxK7da70ez_Iv9_e1sGUKMA366ZpUQqa?authuser=2#scrollTo=hWDXDdZnqiIk

Heart stroke prediction:
https://colab.research.google.com/drive/12yXeMXRlJKUKzaVQrnCAET-TIUjoolu3?authuser=2#scrollTo=8Ag1VUQ85LrU

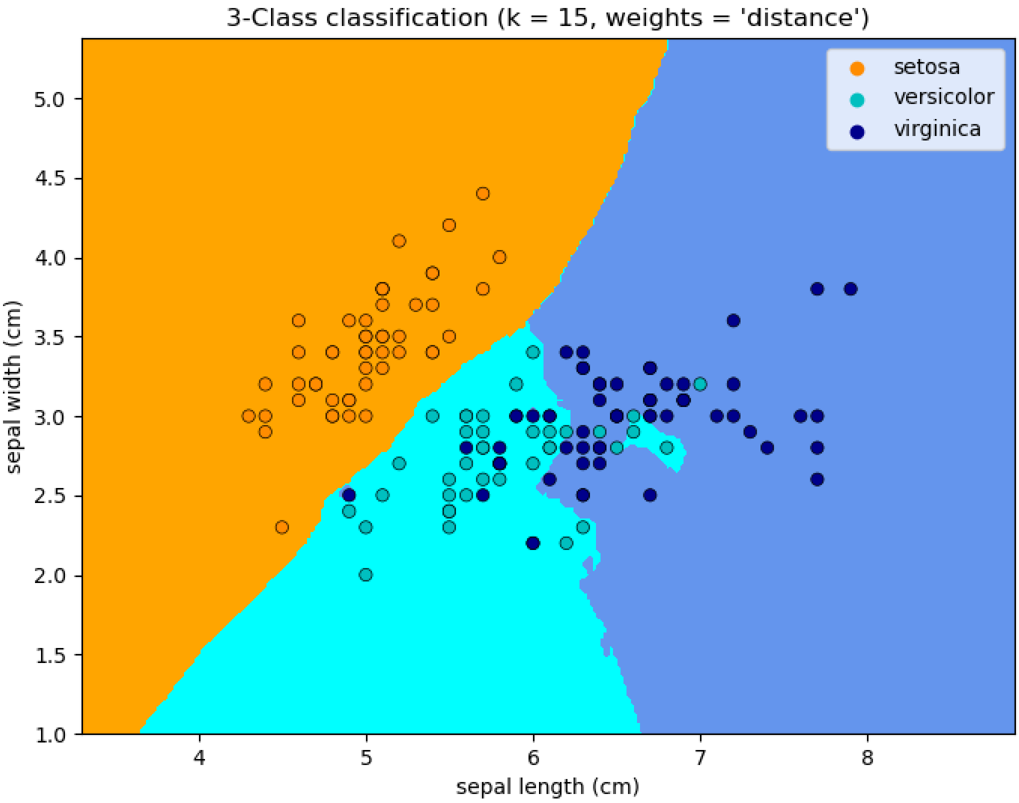
❖ K-NEAREST NEIGHBOR

1) Computes the distance between the new data point with every training example.

2) For computing the distance measures such as Euclidean distance, Hamming distance or Manhattan distance will be used.

3) Model picks K entries in the database which are closest to the new data point.

4) Then it does the majority vote i.e., the most common class/label among those K entries will be the class of the new data point.



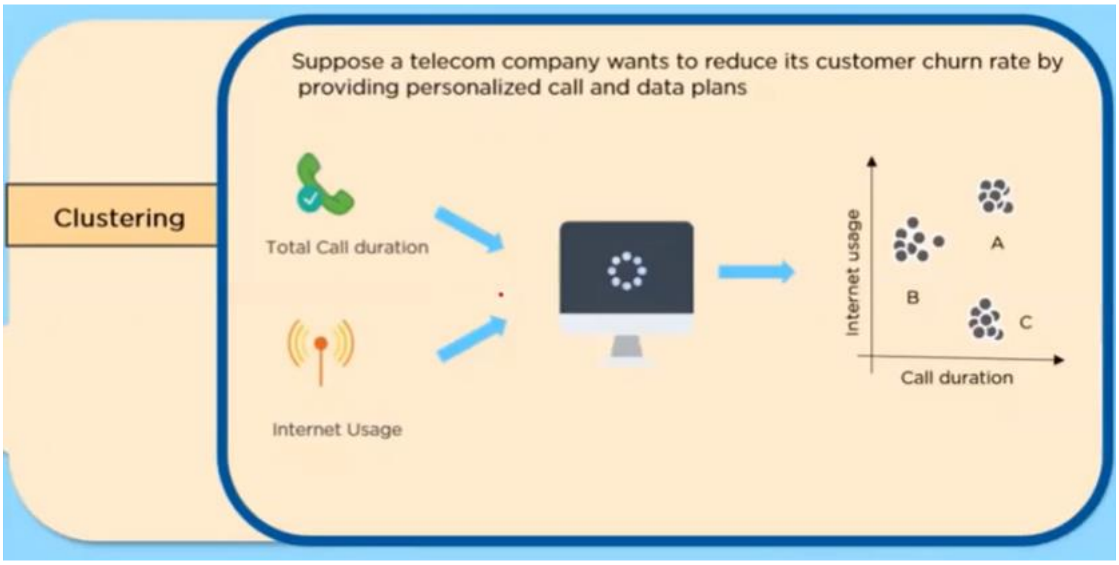
Iris dataset:https://colab.research.google.com/drive/1rqUIA_LaZLhwhpgk8-t23VmpcUrsOdzH?authuser=2#scrollTo=TgTkMeE7pUCA

❖ K-MEANS

The K-means algorithm aims to choose centroids that minimise the inertia, or within cluster sum of squares criterion:

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2)$$

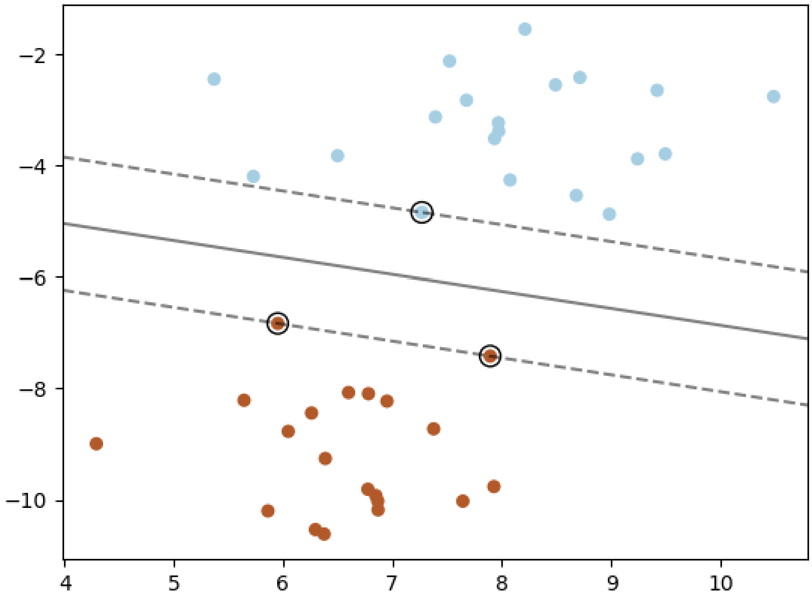
Kmeans1:https://colab.research.google.com/drive/1wZQ8f2jiu5g3bR8pQl-h6SJtFlkqErGj?authuser=2#scrollTo=rippNLAJeGNZ


Clustering

Suppose a telecom company wants to reduce its customer churn rate by providing personalized call and data plans

❖ SVM – SUPPORT VECTOR MACHINES

A support vector machine constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.



Hand written digits recognition:

https://colab.research.google.com/drive/114OB9BLTcj0XMrJi8rMcgkB5HpkWJGVy

LinearSVC:

$$\min_{w,b} \frac{1}{2} w^T w + C \sum_{i=1} \max(0, y_i(w^T \phi(x_i) + b)),$$

LinearSVR:

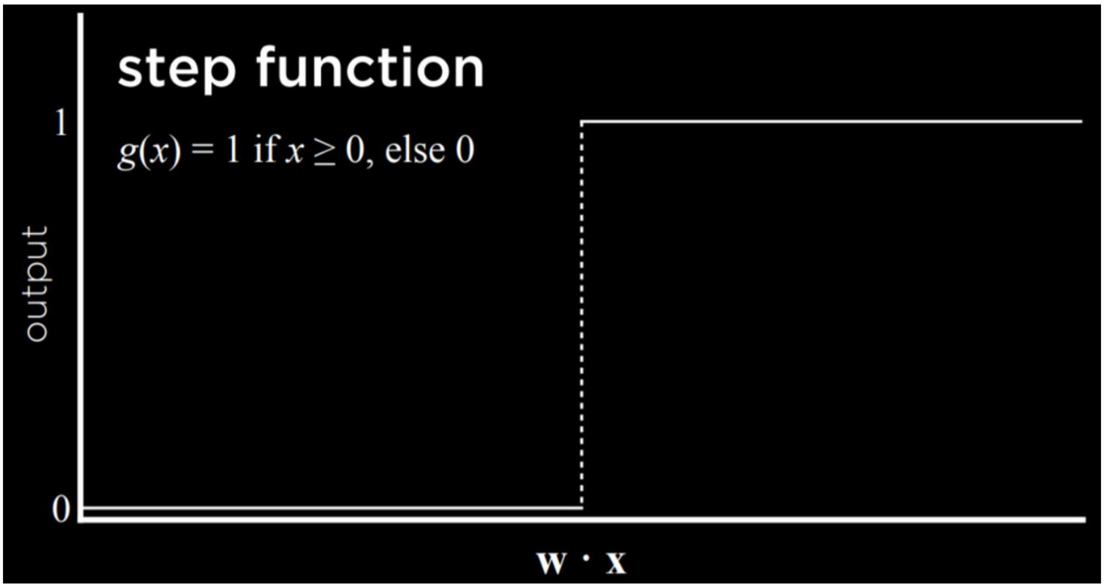
$$\min_{w,b} \frac{1}{2} w^T w + C \sum_{i=1} \max(0, |y_i - (w^T \phi(x_i) + b)| - \varepsilon),$$

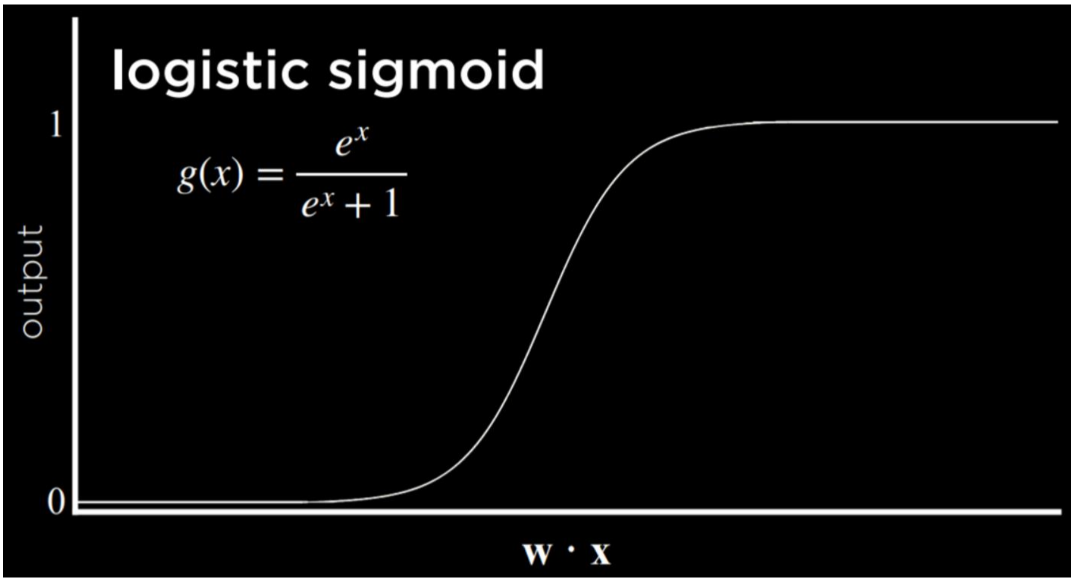
NEURAL NETWORKS

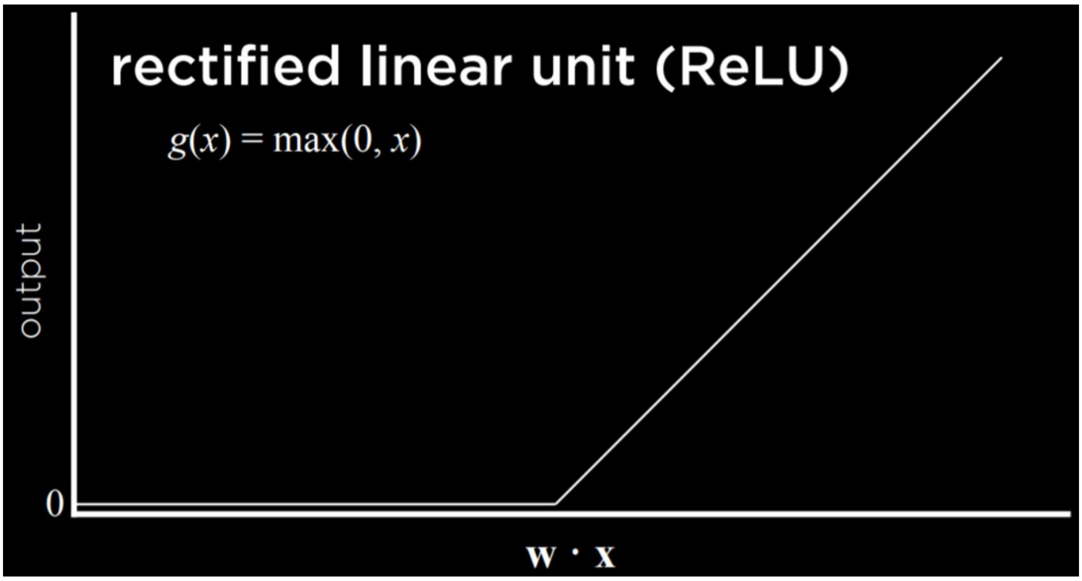
ANN:

An Artificial Neural Network is a mathematical model for learning inspired by biological neural networks. Artificial neural networks model mathematical functions that map inputs to outputs based on the structure and parameters of the network. In artificial neural networks, the structure of the network is shaped through training on data.

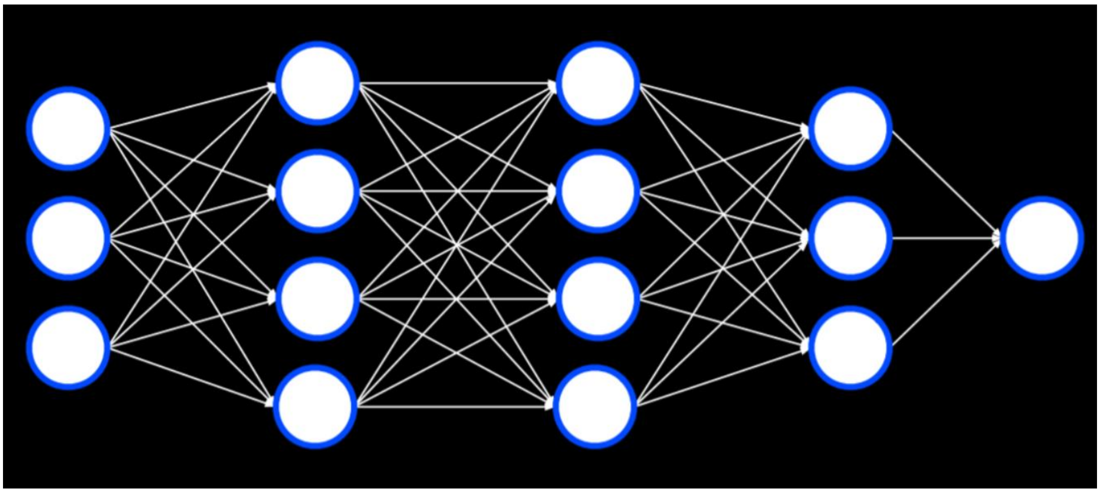
ACTIVATION FUNCTIONS:







Deep neural networks are multilayer neural network that have more than one hidden layer.



Backpropagation:

It is the main algorithm used for training neural networks with hidden layers. It does so by starting with the errors in the output units, calculating the gradient descent for the weights of the of the previous layer, and repeating the process until the input layer is reached. In pseudocode, we can describe the algorithm as follows:

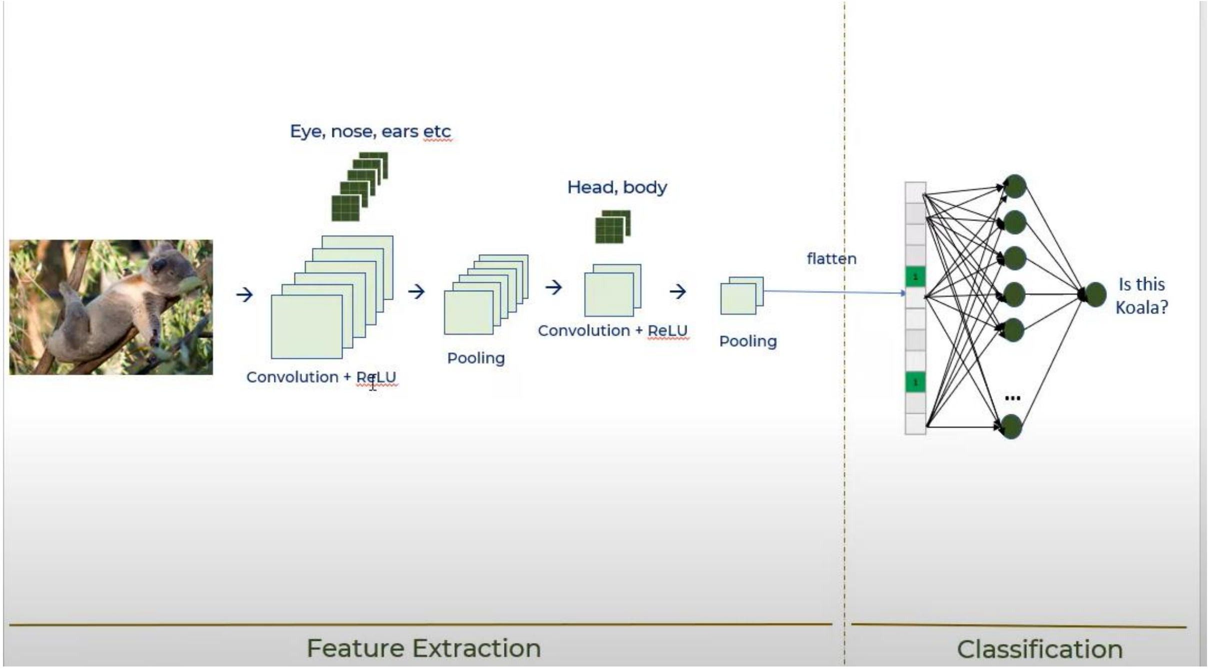
- Calculate error for output layer
- For each layer, starting with output layer and moving inwards towards earliest hidden layer:
  - Propagate error back one layer. In other words, the current layer that's being considered sends the errors to the preceding layer.
  - Update weights.

Cloths Images classification using ANN:

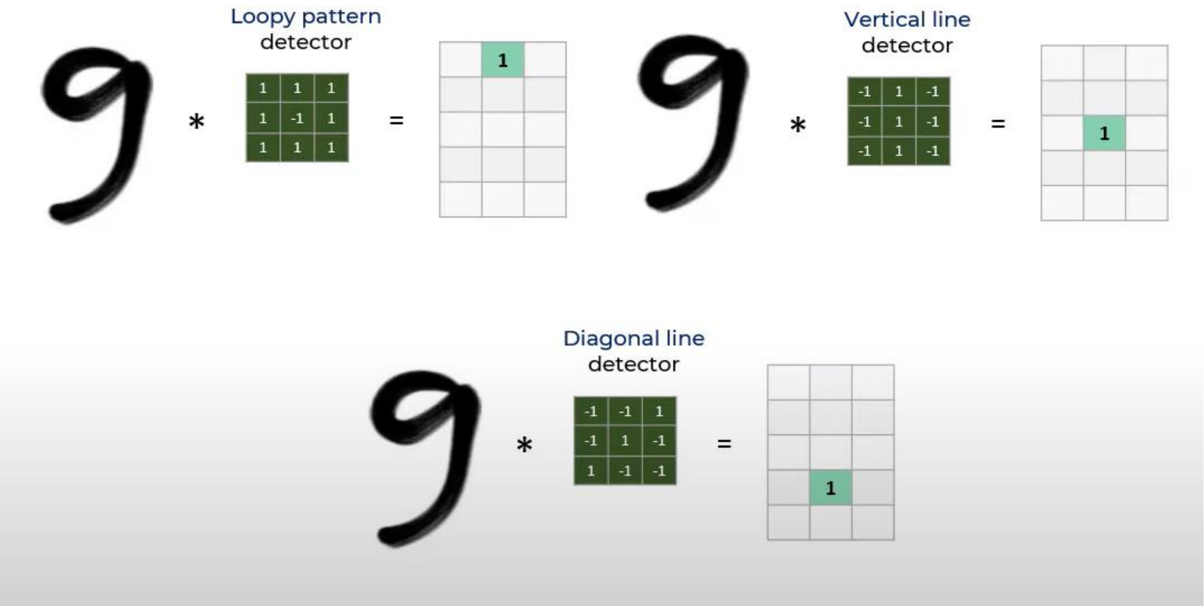
https://colab.research.google.com/drive/1z6KmATAHyHmsXWDCcyU5c5aXJDHb9XNh?authuser=2#scrollTo=vfuDfeuAVnXy

CNN

A convolutional neural network is a neural network that uses convolution, usually for analysing images. It starts by applying filters that can help distil some features of the image using different kernels. These filters can be improved in the same way as other weights in the neural network, by adjusting their kernels based on the error of the output. Then, the resulting images are pooled, after which the pixels are fed to a traditional neural network as inputs (a process called flattening).



The convolution and pooling steps can be repeated multiple times to extract additional features and reduce the size of the input to the neural network. One of the benefits of these processes is that, by convoluting and pooling, the neural network becomes less sensitive to variation. That is, if the same picture is taken from slightly different angles, the input for convolutional neural network will be similar, whereas, without convolution and pooling, the input from each image would be vastly different.



Small image classification:

https://colab.research.google.com/drive/1v8vzZI7cZfpA9Zm259yDqwTkVFWC02w0?authuser=2#scrollTo=ENY2jufys8D-