**Machine Learning in Medical Science**

Medical science is all about classification i.e. a person is suffering from a particular disease or not. This study tends to finds the way which can solve this problem in efficient way.

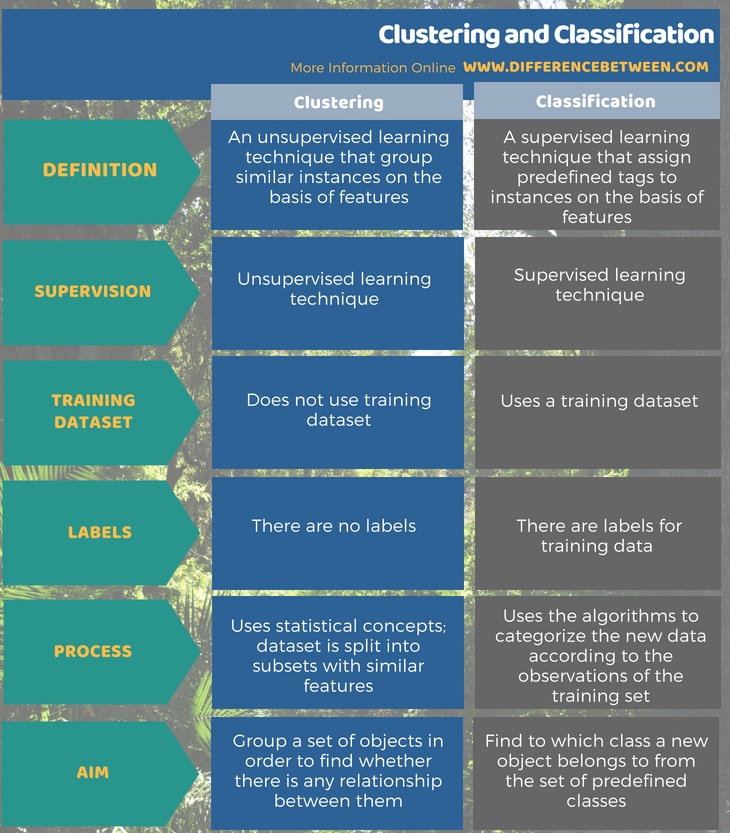
Machine learning comes in three flavours Supervised, Unsupervised and Reinforcement Learning. We are not concerned with Reinforcement Learning since it is simply an interaction of agent with its environment where we improve understanding or behaviour of the agent to the environment either by updating goal or guiding it in some other way. That leaves two main part of the machine learning, before going into detail of which I would like to bring your attention to the kinds of problems that we solve through these algorithms.

**Problems Solved With Machine Learning**

So the problems that are solved with machine learning can be thought of as **Regression**, Classification and Clustering. When there are infinitely possible outcomes of a given problem then you have regression problem. That is when output of a model is continuous. For example how much time would it take a business to reach a particular point of market capitalization. Now, we see that time can be seen in its smallest unit and which would be ‘second’ and yet we can divide it in infitely many smaller pieces like millisecond, microsecond, nanosecond etc. Similarly we can find salary of a person given the experience, which could be years, but then it could be months, or seconds, or milliseconds and so on. However it must be noted that time can be taken in chunks in that time it would not be continuous rather it would be non-continuous entity and chunks can be grouped then. In that way we can Classifytime into groups for example age can be classified into groups like baby, kid, adolescent, young adult, adult and so on.

And then we have **Classification** problems. Here we deal with groups and when data is presented we are supposed to identify which entry in dataset belongs to what category or class. We normally have labeled data and dataset has “features” and then we are given some features and we are to predict which features would lead to what classification. For example, we might be given colours and sizes of fruits and are to predict which fruit that is. In this we are given some labels, initially, and then when new data is presented it is not labeled and we are to assigned labels to it based on its features. And this falls in **Supervised Learning,** since data is already labeled and thus we supervise the computer for making predictions. Another problem that can be taken here, for easier understanding is we are given labeled data with ‘male’ and ‘female’ classes and features could probably be weight, height, skin or scale colour (in case it is a fish).

However, data labeling is not always possible that is when computer is supposed to learn from given features and put things with similar features at one place. This would be **Clustering** which is **Unsupervised Learning** type of problem. Consider the problem above. We are given data but it is not labeled and computer automatically sorts things by features and later we can easily put labels on the clusters. Consider this a little more complicated example; some super store asks its customers about what they do and estimates their income and then tries to correlate it with spending. We might see, solving this problem, different pattern in our data, one might be that some people shop more carefully when they have low income and some shop more when they have more to spend and some might just spend it all as soon as they get despite not being very high income. This kind of classification might help the store managers give the shoppers a few benefits which may encourage them to spend for the best of everyone i.e. store as well as themselves.



Reiterating all above is that, there are two main ways to solve classification problems one is the supervised way, in which computer learns from labels, and the other is unsupervised way, where computer learns features and puts them together.

**Input Data Types**

Data can be in form of text for example categorical data which is given in columns or in form of images. In case of text based data i.e. columns, computer tries to formulate what features describe data best i.e. what would produce best classification results, as well as inference of as to what relation do the features have with each other and to the class. Consider this example, height and weight may directly predict whether the observed entry belongs to male or female as well as they have relation with each other as more the height higher the weight as well, however, there may be anther feature which determines gender of the observed, say skin tone, and neither weight nor height can predict skin tone or complexion.

But there are times when data can’t just be presented in form of written values. For example, to determine which fruit is the observed case. If there are over a thousand classes of fruits then it would be almost impossible to write down all the columns catering all the features that are required to determine the class of the observed fruit. In such cases we take images and label them this way computer automatically make inferences about what the features are that determine the class. And it is important to give information like this in some cases, specially when there are a huge number of classes. Consider this example; if there are only two things to be distinguish then it is easy for example colour, contour and shape and size of object would determine whether it is mango or apple but if apple is replaced with oranges and we have only these features to determine then would wrongly classify oranges as apples because both of the fruit are round and almost have same size. That is why images must be provided in such cases.

If there are a huge number of images to be classified for example on Image Net the dataset has over 14 million of images so they can’t all be labeled, if there is such a huge data, not that image net is not labeled, that is labeled but doing that for another dataset or making a dataset that is as huge as that is, would require a huge number of man hours, if done by people, or huge compute would be required to do that if it id done by computer.

**Machine Learning Algorithms**

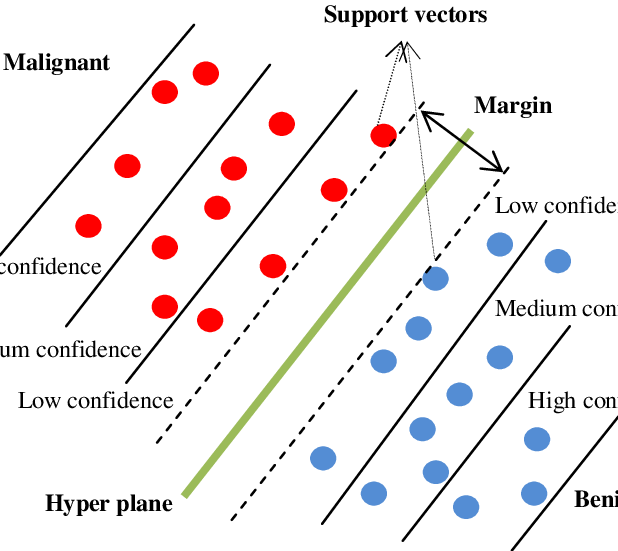
In terms of understandably of algorithms there are of two types: **Black Box** and **White Box**. Names suggesting that it is not easily possible to see what an algorithm is doing for black box algorithms and vise versa for the other.

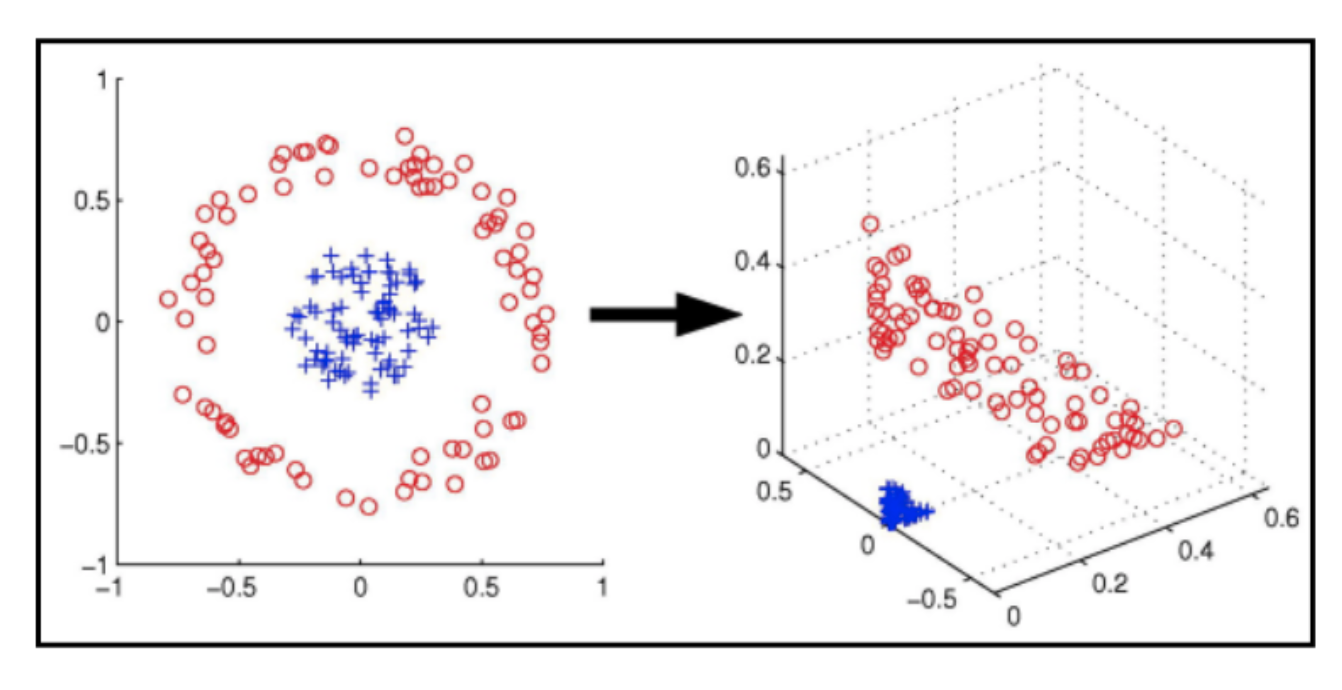
Medical science is all about image classification because now a days it is easier to train a model to classify images than to obtain and extract features by manually and carefully reading information of patients or of people who are potentially suffering from disease.

**Support Vector Machines**

One of the two most widely used algorithm in medical science, pre deep-learning era, is SVM. These algorithms are dichotomous in classification outcome i.e. they have only two categories, one is positive and other is negative. So they are the best for use when there are only two outcomes i.e. yes or no. Since in health care we are interested in knowing whether a person has a disease or not so they are most suitable. An other merit of this technique is that it does not require huge amount of data for classifier to work and learn from.

The way it works is that data is projected in a dimension higher than it was originally in then a spatial separator is used to distinguish the two classes. Idea is to find support vectors which put the opposite classes at maximum margin.

****

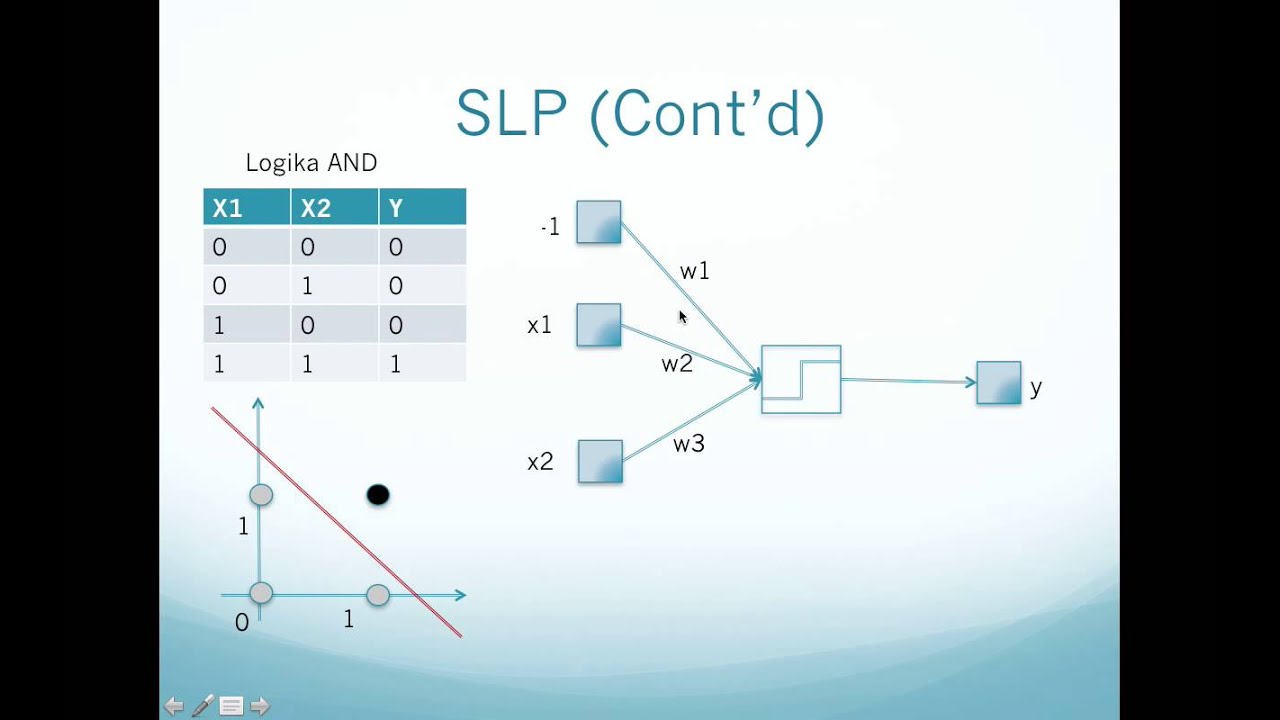
****

**Logistic Regression**

This is a white box technique used to classify objects. This algorithm assumes that there is a linear, or nonlinear, function that describes relationship between input features or independent variables and the dependent or output variables. This generally plots a line between boundaries of classes. This boundary is called line of best fit. The relationship is usually given by the function mx+b or something similar to it. It could be polynomial where ‘x’ has higher power than one, quadratic for two for example, and it can also be multi linear regression where there are many independent variables that determine outcome of the equation. And the function could be a mix of these as well. The more the variables or the higher the power of the highest power of a variable the more difficult it is to represent the equation on graph, and anything above degree of three is impossible since a hyper plain can’t be plotted. Also higher degree polynomials require more compute resources.

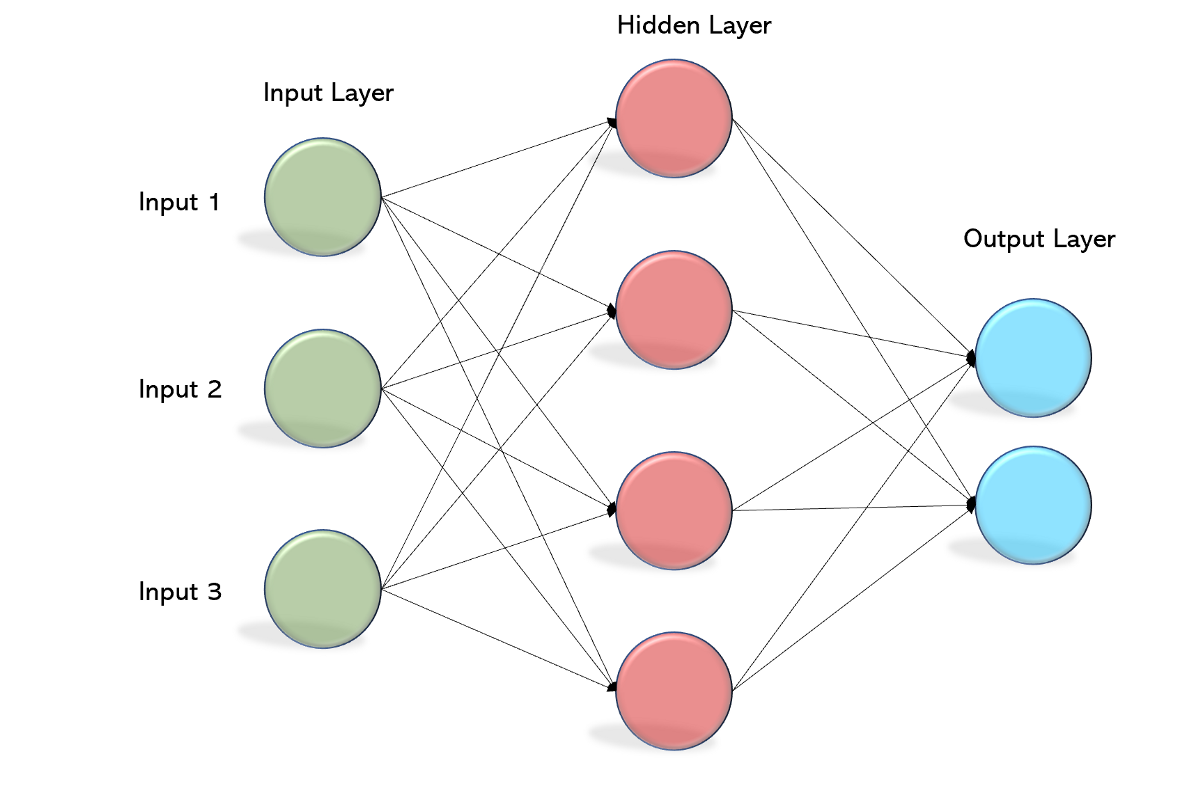
**Multi-Layer Perceptron**

A perceptron is basically just linear or logistic regression model, however, when multiple of them are put together in a way that output of one becomes input to next in multiple layer fashion then we have multi layered preceptron. They are basis of the deep learning which is described below.

****

Above is pictorial representation of single layer preceptron. It looks and functions similar to logistic and linear regression model.

Below is multi layer preceptron, it has three layers, at least three are required, one is input layer which is the basic part of the model, it takes input, one node or neuron or perceptron is used to get one input, though as opposed to single layer perceptron where one/single perceptron is used to take all the inputs, and then each node is connected, in densely fashion, to all the neuron in next layer. And we have one output layer, which consists of at least one neuron, if there are only two out comes possible then only one is required though multiple neurons can do the job, however, this will over complicate the things and more compute power will be required. And in the middle we have hidden layers, if there is only one layer then we have just “multi layered” perceptron, if however we have many layers then we have a fully connected neural network.

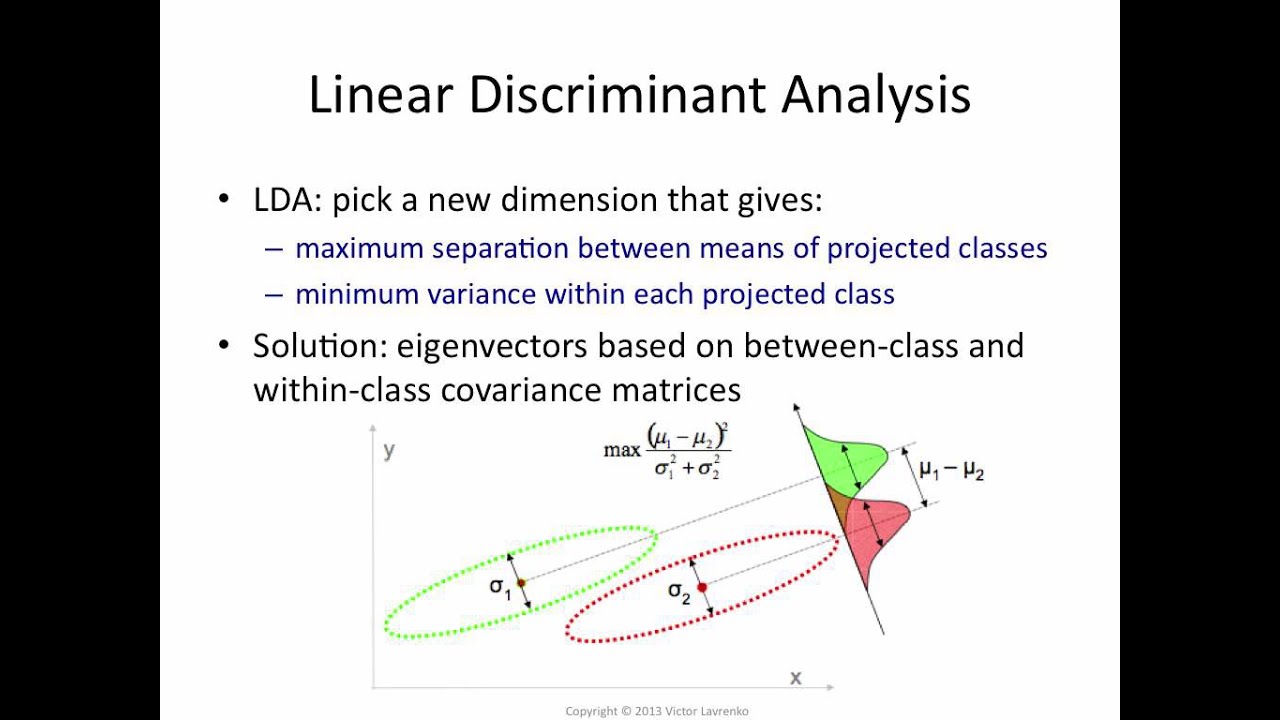
****

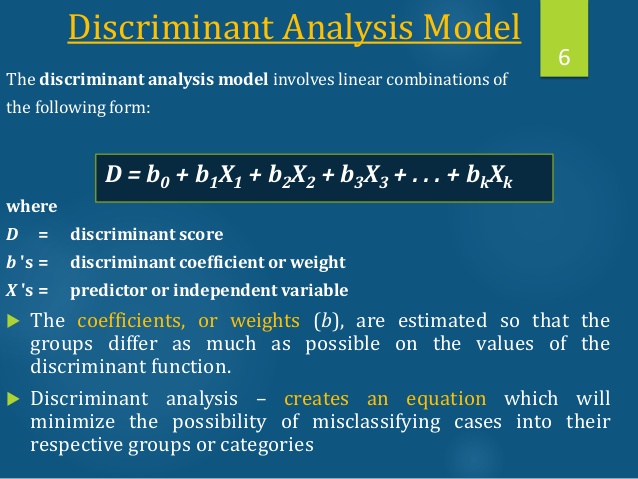
**Discriminant Analysis**

Linear

It is similar, in a way, to SVM and to logistic regression in another way. LDA is done by maximizing the distance between the mean points of two classes, in first step, and then minimizing the “scatter” or variation between the points of a class.

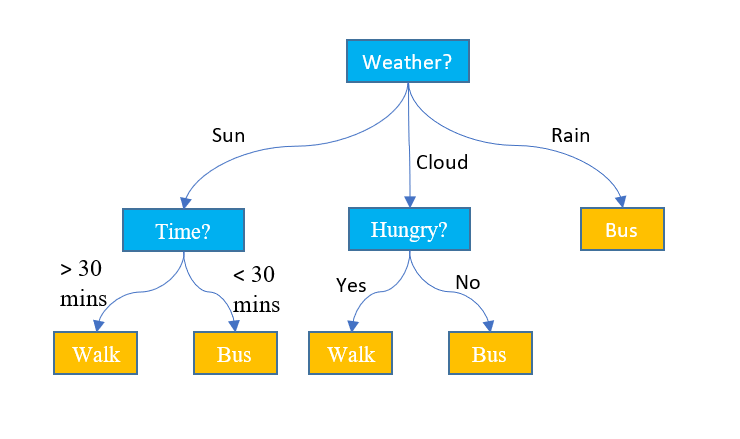
There are many other form of discriminant analysis, for example quadratic discriminant analysis and principal component analysis. They are based on the same idea. They are also used to extract feature or reduce dimensionality of the data. That is when there are many features to a data then curse of dimentionality ensues i.e. computation becomes very heavy and in most cases data can not be plotted on graph. However when dimensions are reduced. Some times it is possible to represent data in fewer dimensions that is when most of features do not contain as much valuable information on outcome of the independent variables.

****

****

**Trees**

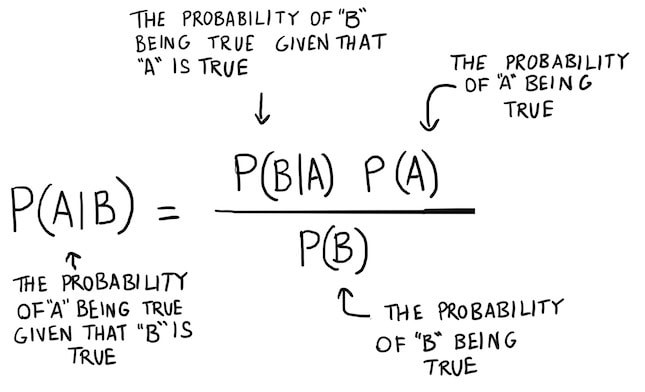
Now there are many algorithms that contain various architecture of vanilla **decision trees,** for example the most recent entry in tree family is **Catboost** which appeared a year after **XGBoost**, these are boosting techniques which aim to boost i.e. give higher weightage to a decision tree that predicted outcome correctly previously and vice versa to the other, these are based on **Adaboost** technique which underlays paths of the boosting technique. Other than boosting we have bagging in which we take different decision trees and aggregate results. So what is decision tree exactly? Well, they are algorithms that predict outcome based on rules. For example

When you have two choices to reach a destination i.e. either to walk or take bus and you have to ask yourself a few questions what should I do there is a systematic way this can be solved. 1> Look for rain, if it is raining then take bus, if it is not raining then check whether it is cloudy or is it sunny, if it is sunny and you do not have more than 30 minutes then take bus other wise if you are hungry then take lunch and digest it one the way.

However, decision trees have tendency of being over fitting so, a bunch of questions are taken in tandem and then checked if they present better decision on unseen data that is what **Random Forest** is. However now a days new bagging and boosting techniques are replacing use of RFs as well.

**Naive Bayes Algorithm**

What is probability of an even occurring while another event has already occurred. This is statistical model which deals with probabilities. To find probability of the desired event you need to find probability of both the evens independently. This makes it naive because it assumes that all features in dataset are independent of each other. Thus it also suffers from ‘zero-frequency problem’ which is if in training data a particular class does not exist then on test data zero probability will be assigned to it. However its advantages include that it does not require huge amount of data to make a prediction. This speeds up the computations a lot. It is also good for multi class prediction.



**K-Nearest Classifier**

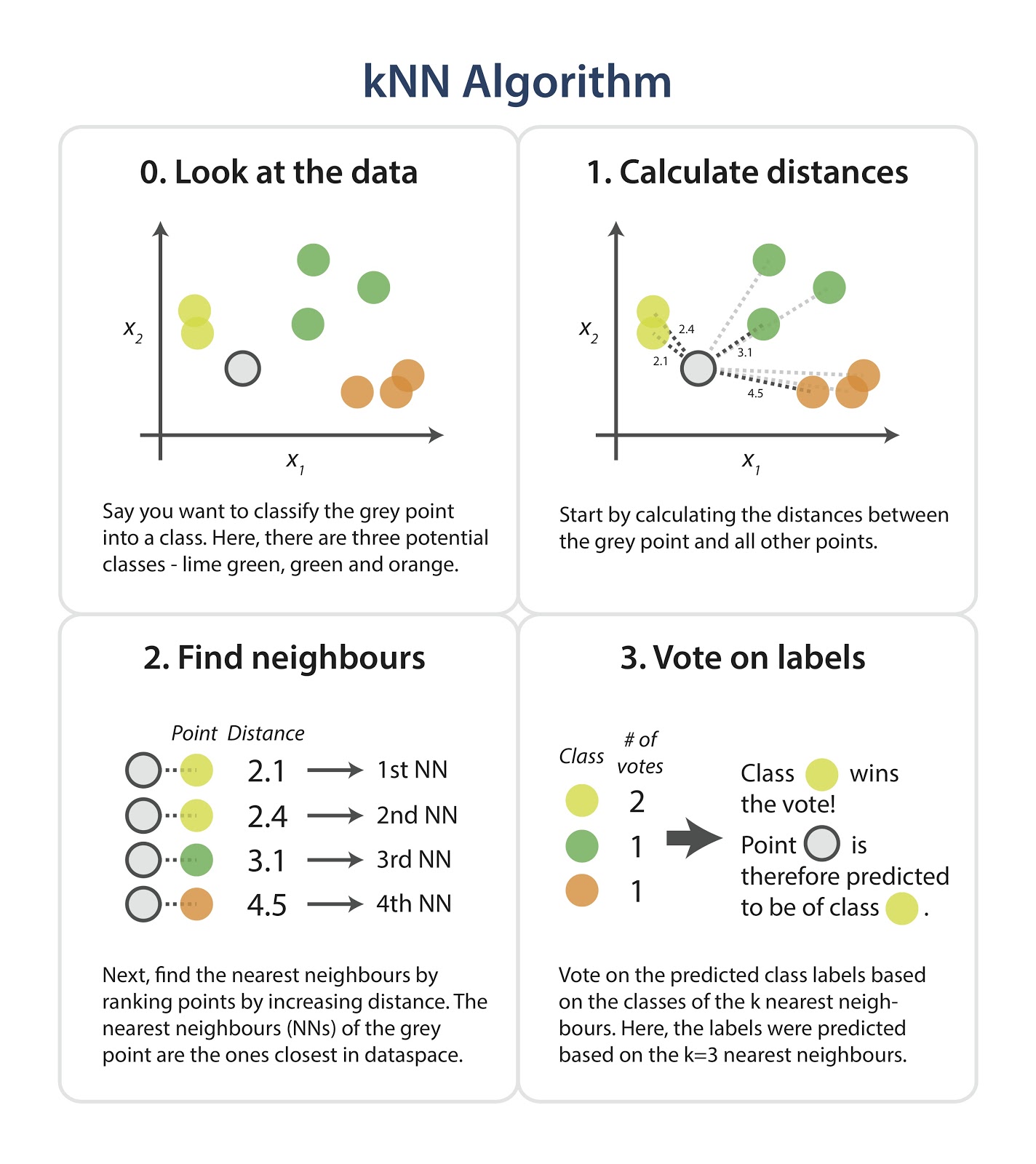
What is true for 2D is true for data that is in any other dimensions. Steps are

1> Consider this example of 2D data where all data points are plotted on graph. And a number is assumed that there are these many number of classes of the data.

2> Take the distance of the data point under observation from all other data points on graph.

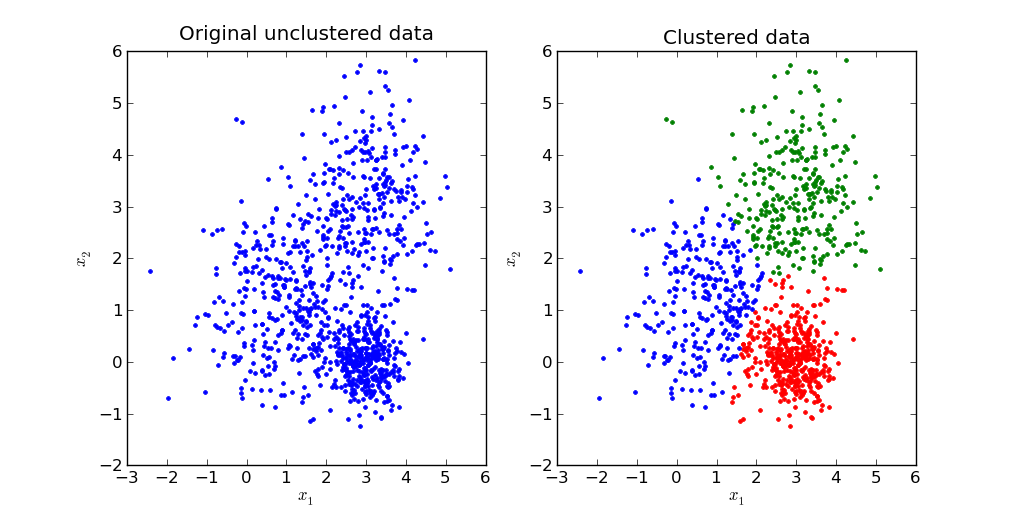
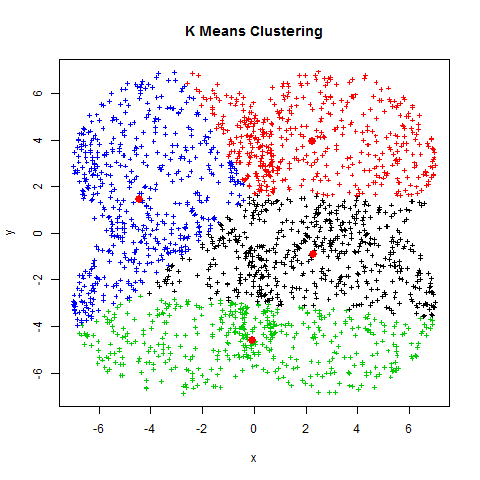
3> The nearest data points to the datapoint under observation would be used to classify the datapoint.

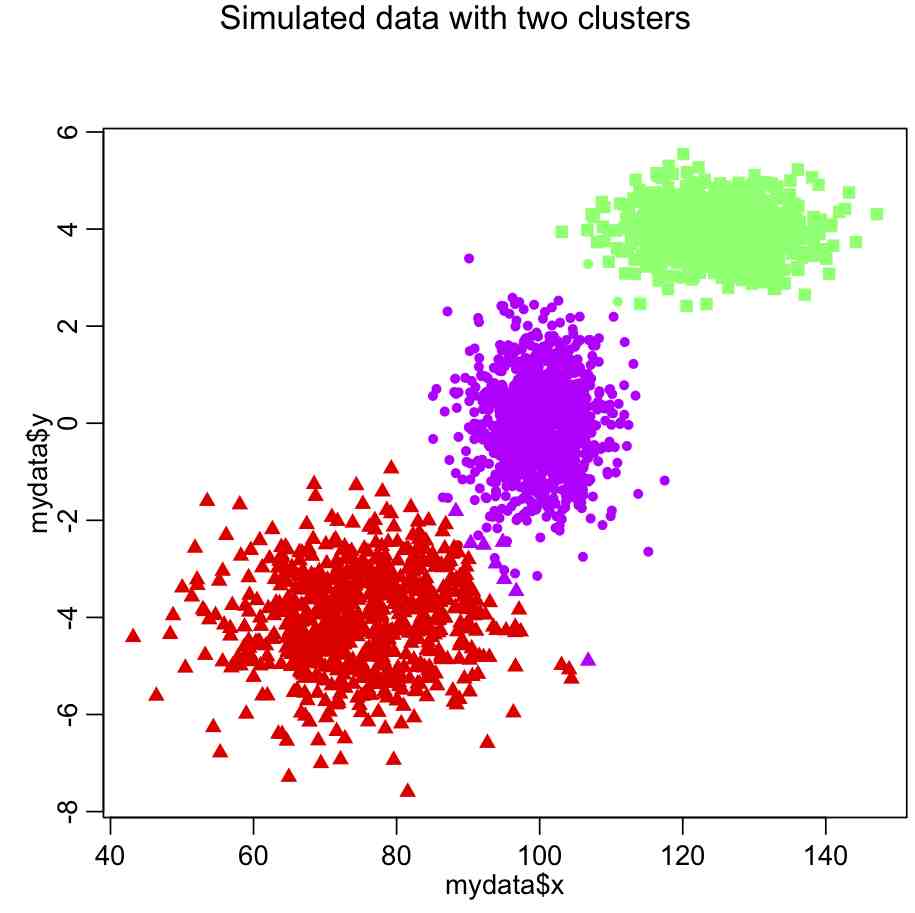
There can be as many classes as there are datapoint on the graph. The more the classes there are the harder it is for computer to compute the class for each datapoint. So it becomes slow when there is a huge data to be plotted and when there are too many classes possible.



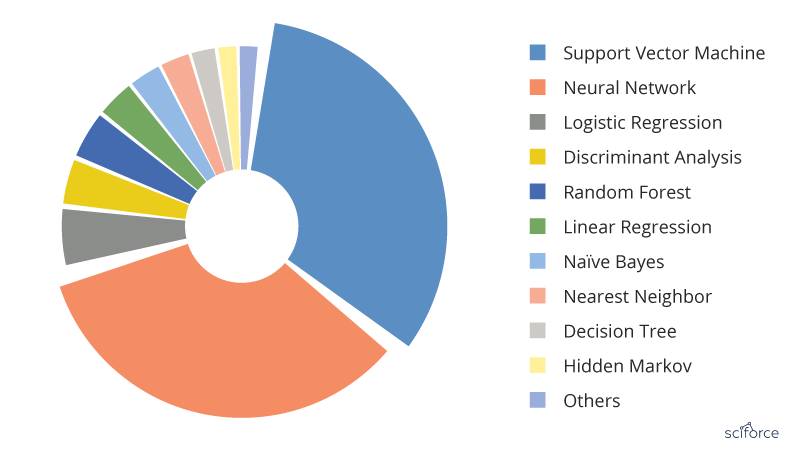
**K Means Clustering**

You must have some heuristics about number of clusters at start, mostly it is easily decipherable from plotting on graph. Then you find centroids or central points of each cluster. Find distance of observation to the centroids. Group data based on minimum distance. Recentre the centroid, if you have reached desired outcome then end the process otherwise start from second step that is find distance of observations from centroids.





Plot of usage of various algorithms as currently they are used in medical science is as follows.



There are other clustering techniques but I would like to move on to deep learning now.

**Deep Learning**

Deep learning is done in two ways, one is features are given in texual form and the other is pictures. When there are features given as text then you can use simple, vanilla, dense neural network which is multilayered perceptron to classify things. Otherwise there are two main techniques that I consider worth mentioning. Attention based and pattern based.

When we have data in form of images there are various things we can do to obtain relevant information, for example in case we did not get clean images, then we can de-noise images or increase resolution. Though GANs, generative adversarial networks, may generate data that may lead to wrong classification, so we only use GANs to interpret images ourselves i.e. do not pass it as “original” image to computer to classify, specially in situation where correct predictions are critical.

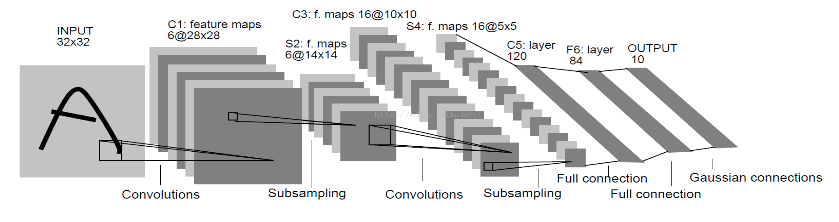
Prior to 2010 SVMs ruled the field of machine learning, when computers were not strong enough, and there were only single core computers. Then advent of stronger GPUs and recently TPUs and IPUs has seen many new trends. Even in 1990s perceptrons were considered potential replacement for SVMs but there were not many case studies back then which supported that claim since deep learning algorithms require a huge amount of data and huge amount of compute power to process it, which was not available back then. But recently, 13 Jan 2021 a nature article claims that there was substantial improvement in performance for MRI image classification when done on deep learning models as compared to standard machine learning models, as discussed above.

***Attention based techniques***

***Local Attention: CNNs***

Yan LeCunn came up with idea of convolutions. That map the features of image which, maps, are flattened and put in form of array and are fed to a fully connected layer to a dense network. His idea was brand new and he wanted to read hand written alphabets and numbers. This was a genius idea but of course could have not been used much because of its data hungry nature and requirement of computational resources.

The idea is that if a convolutional layer can map features of image then it does not matter whether the object in image has moved position or angel because the feature will always be captured by the convolutional layer.

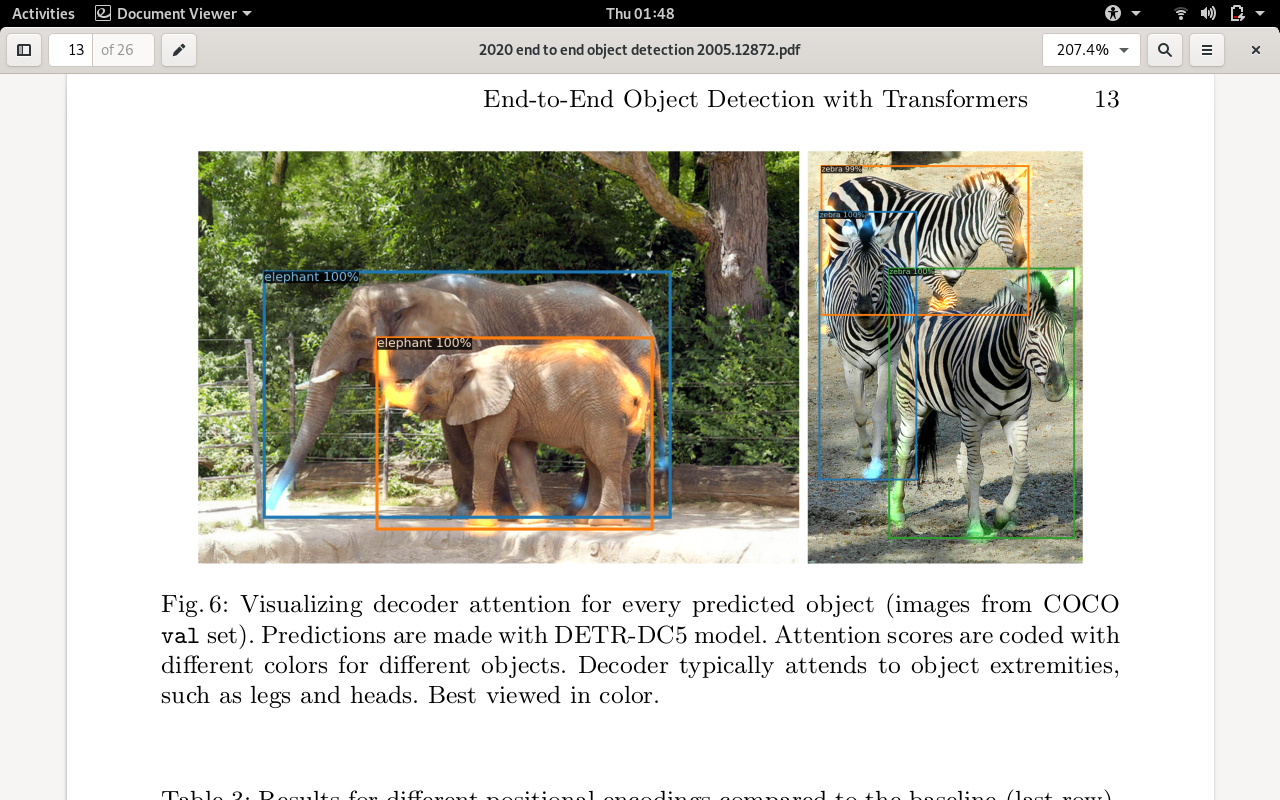


There have been many advancements in this regards and most of the models that locally pay attention through a small window of convolutional layer in form feature map have been out performing conventional ways of machine leaning. I would like to mention two important models in this regard. The first one which won image net 2014 competition with 92.7% accuracy. And the latest SOTA model is, efficient net family, best of the family members is L2 with 480M parameters and this achieves 90.2% accuracy on Top one accuracy and 98.8% on top 5 accuracy using **Meta Pseudo Labels** according to paper in 5 Jan 2021**. LambdaNetworks 2020** however must not be forgotten which achieve efficient net level of accuracy with 4.5 times increase in time and has almost ten times less parameters with LambdaResNet152 standing on 35M and LambdaResNet200 on 42M and biggest of them all, LambdaResNet-270, uses only 34 (G) FLOPS to compute. Lambda networks are based on lambda calculus.

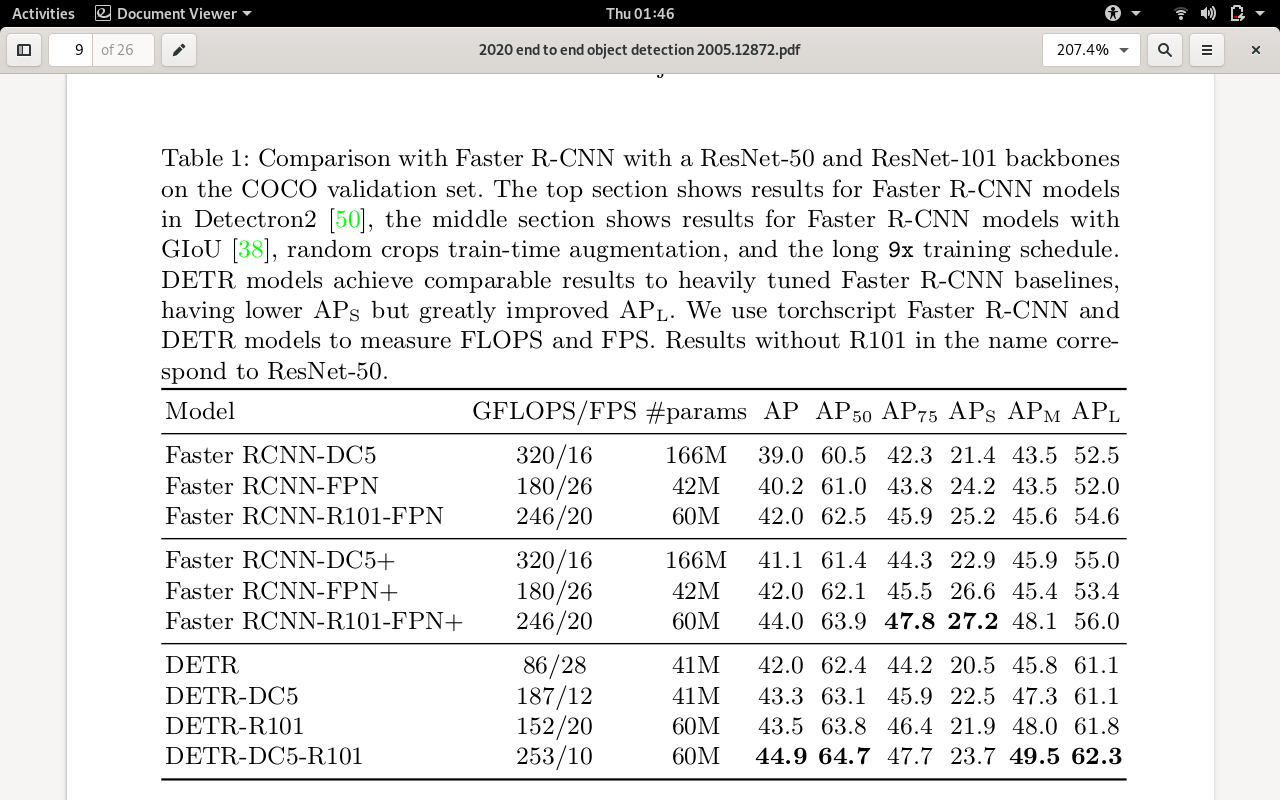
***Global Attention: Transformers***

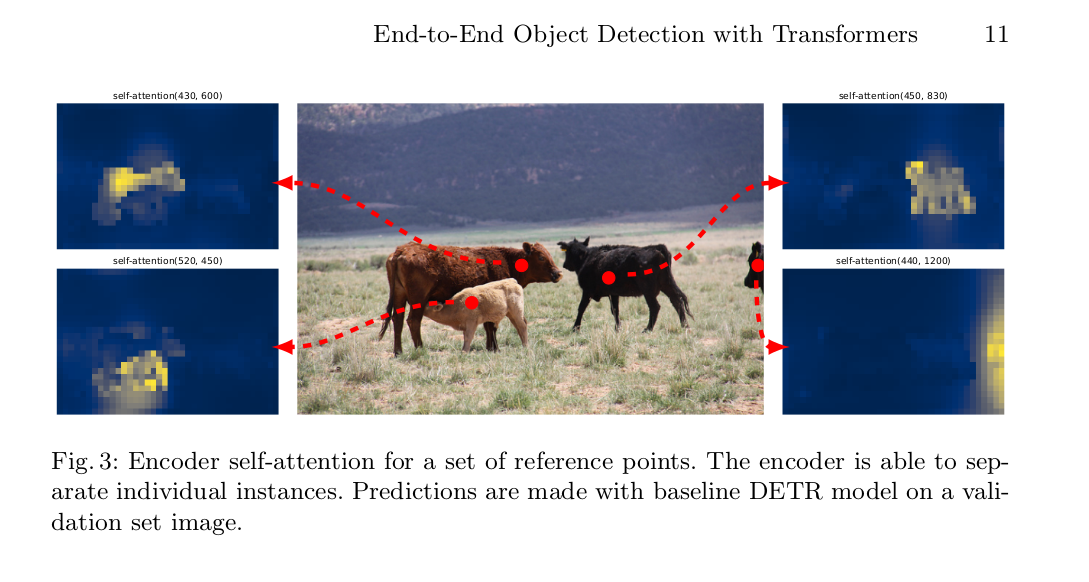
***ViT***

“Attention is all you need” was a game changer paper that came in 2017. And it took deep learning community by storm. It was initially used in NLP tasks because it is good at modeling sequence to sequence data. However its power was realised in computer vision as well. With papers like “**An Image is worth 16X 16 Words**” which came out in 2020 there was a new trend in CV. Now we have models that can generate bounding boxes with 100% certainty and speed of about 15-20 images per second. An other great paper in 2020 is “**End-to-End Object Detection with Transformer**” the amazing thing is that this paper can capture and identify things which were considered impossible by computer. And some times objects are so hard to identify that even human can’t see them correctly. The following image sows the power of transformers when used in computer vision for object detection. It not only has hundred percent recall and precision rate but also it can identify that the elephant surrounded with blue nebula is different from environment as well as the elephant from yellow nebula, even though there is only small part of tail and foot is visible to the screen.

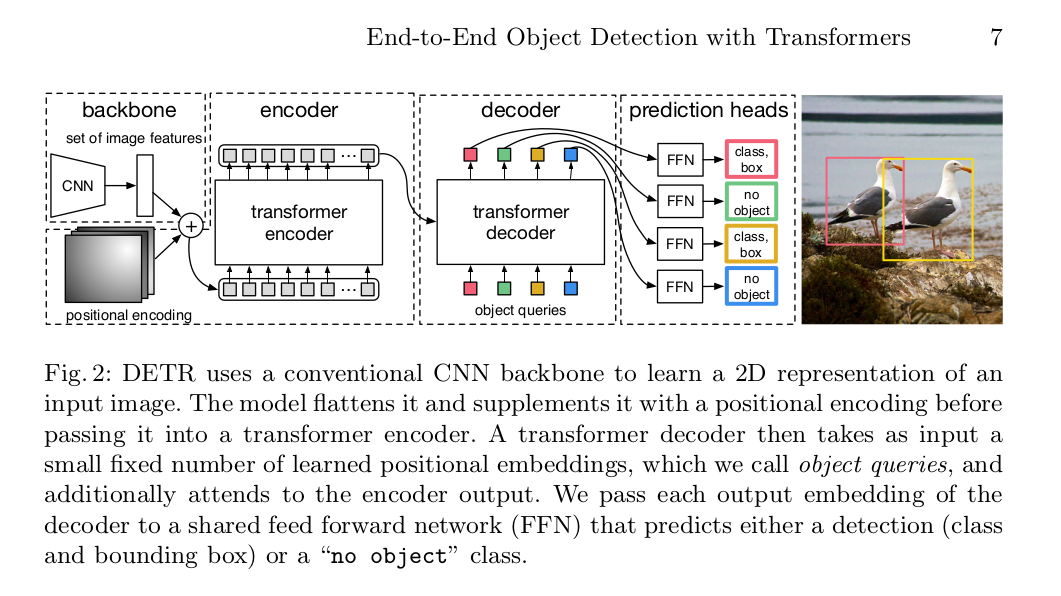


The image below is another screenshot of the paper showing that how detection through tranformers outperforms resnets (faster in this case) in almost all goals as well as it uses fewer parameters and thus has capacity to increase in parameters, but also mean that it can run way faster than faster RCNNs.

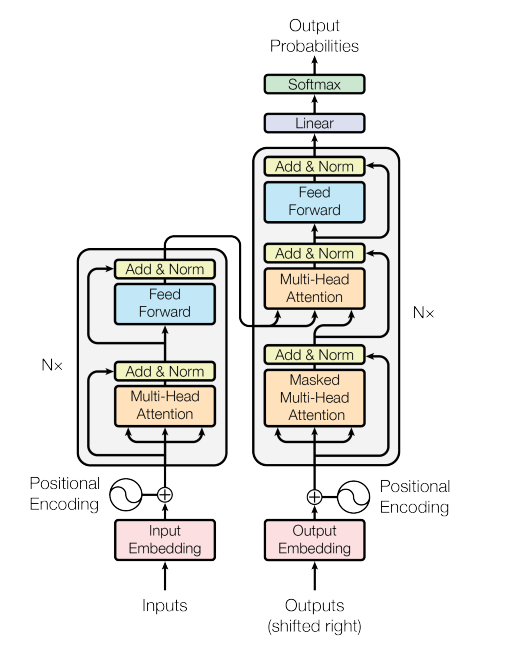


This image below shows that where attention is being paid, as attention is the main goal of this paper..

***Architecture of DETR***

******

In the end I would like to mention the architure of the mother of all attention papers which started this all with the paper **Attention is all you need**



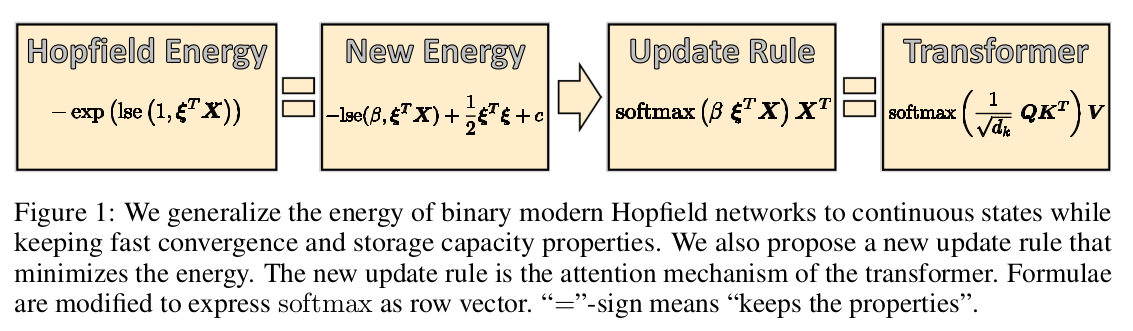
***Patterns based techniques***

**Hopfield Networks**

They minimize the energy of the system hence stabilizing them. They can store the patterns which can be matched and hence used in image classification and object detection in images. The paper “**HOPFIELD NETWORKS IS A LL YOU NEED**” 2020 suggest that the network qualifies as attention mechanism hence is almost as promising as transformers. More research is needed in this however to prove this claim. “The new Hopfield network can store exponentially (with the

dimension of the associative space) many patterns, retrieves the pattern with one

update, and has exponentially small retrieval errors.” And “**Modern Hopfield Networks and Attention for Immune Repertoire Classification”** claims that “attention mechanism of transformer architectures is actually the update rule of modern Hop-field networks that can store exponentially many patterns”. Following is the Hopfield energy is compared to that of trnasformer.



Below is the architecture of Hopfield Networks

