

A Crush Course for Machine Learning

Xie He

What is Machine Learning

“Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy.”



The diagram consists of three concentric circles. The outermost circle is light blue and contains the text for 'Artificial Intelligence:'. Inside it is a medium-sized light purple circle containing the text for 'Machine Learning:'. Inside the purple circle is a smaller light green circle containing the text for 'Deep Learning:'. This visualizes that Deep Learning is a subset of Machine Learning, which is a subset of Artificial Intelligence.

Artificial Intelligence:

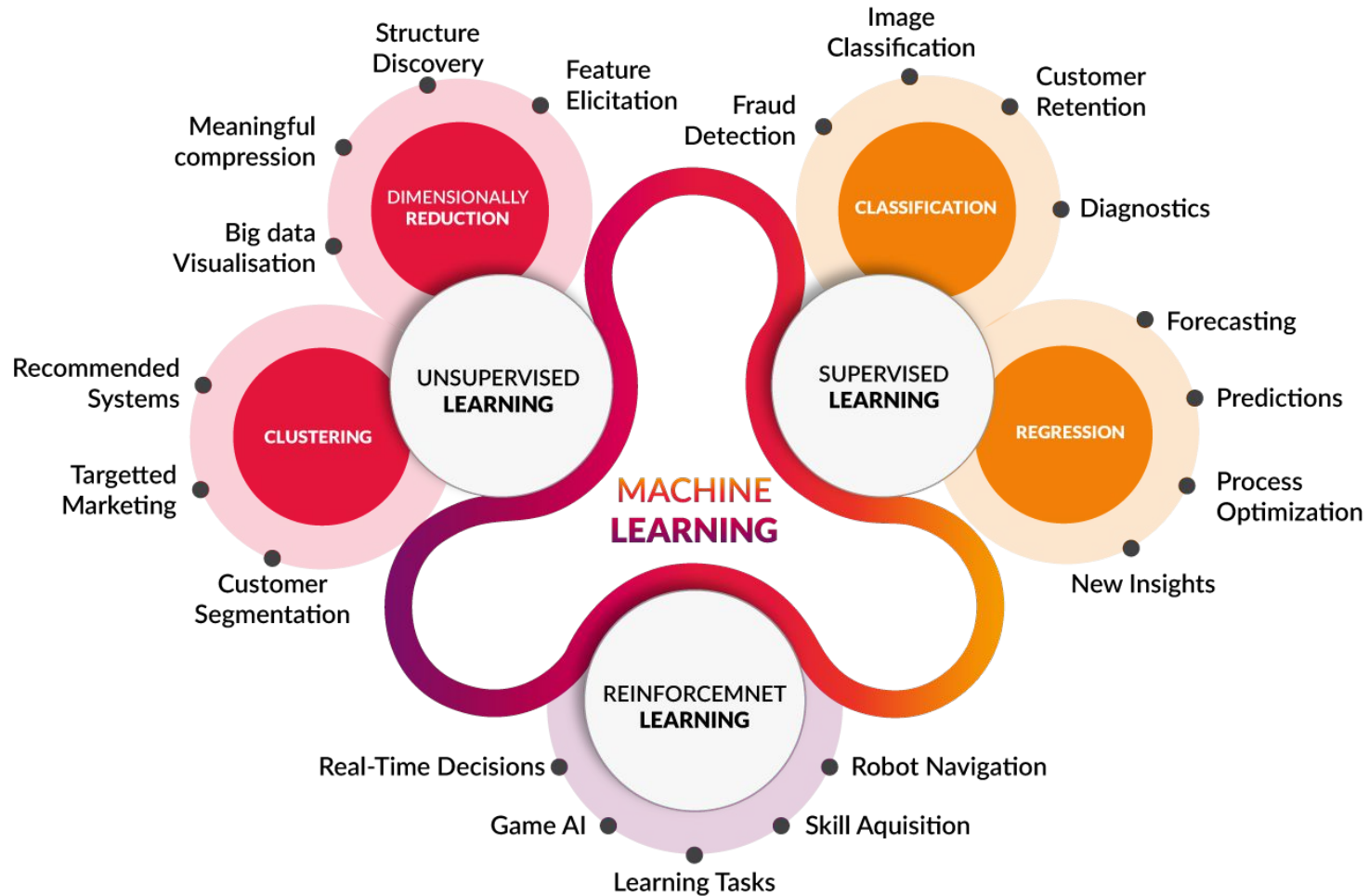
Mimicking the intelligence or behavioural pattern of humans or any other living entity.

Machine Learning:

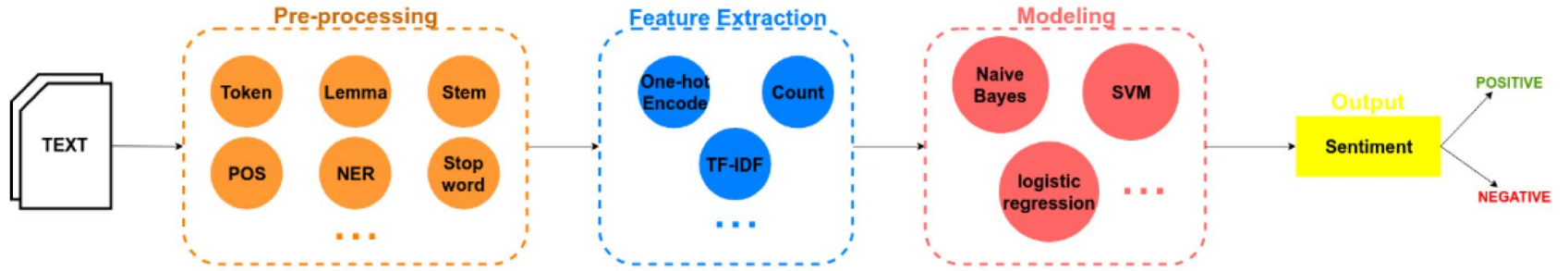
A technique by which a computer can "learn" from data, without using a complex set of different rules. This approach is mainly based on training a model from datasets.

Deep Learning:

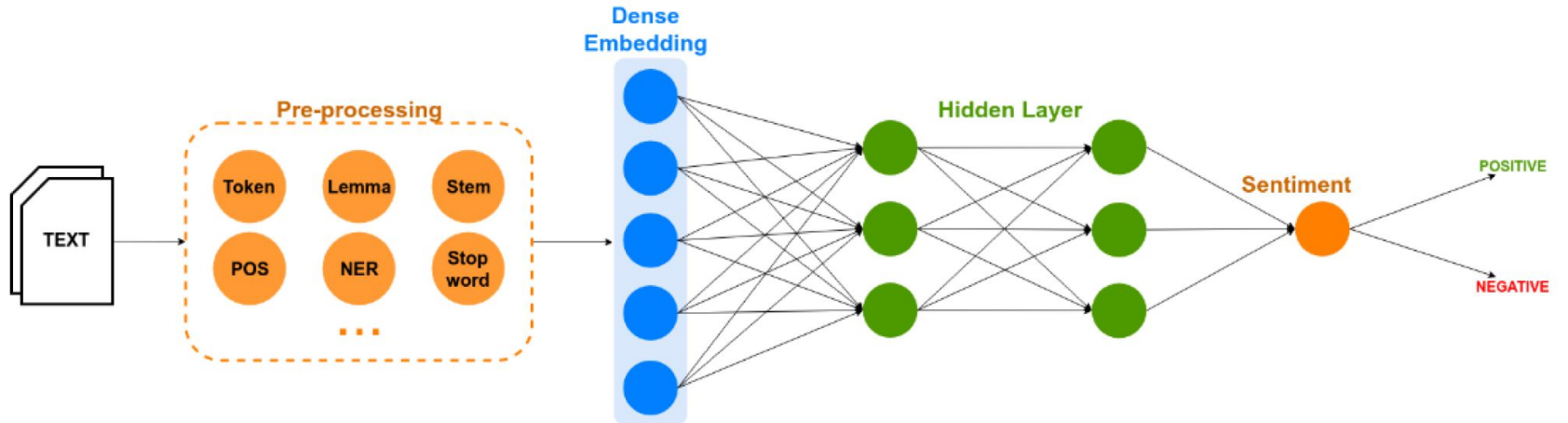
A technique to perform machine learning inspired by our brain's own network of neurons.



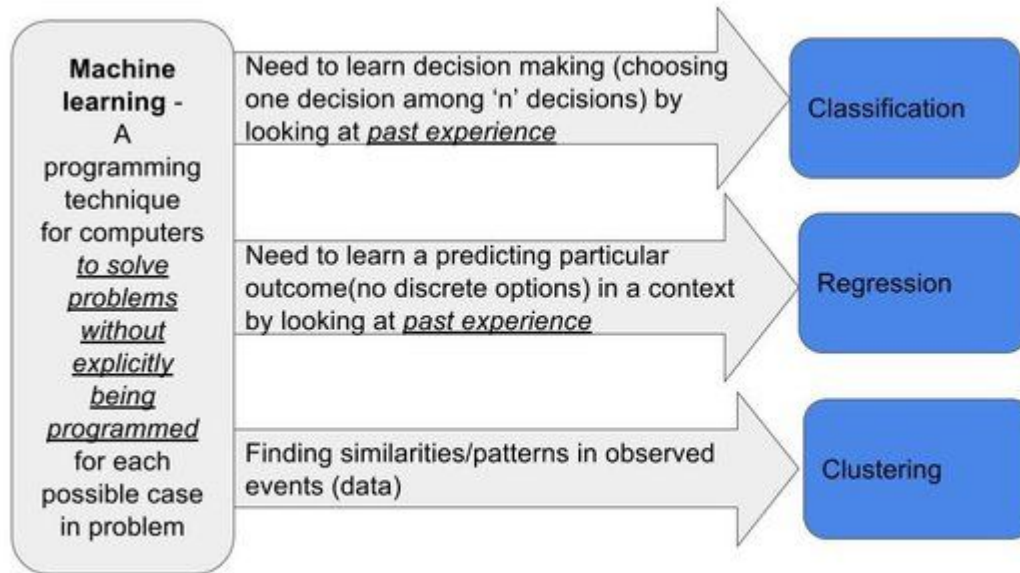
Machine Learning



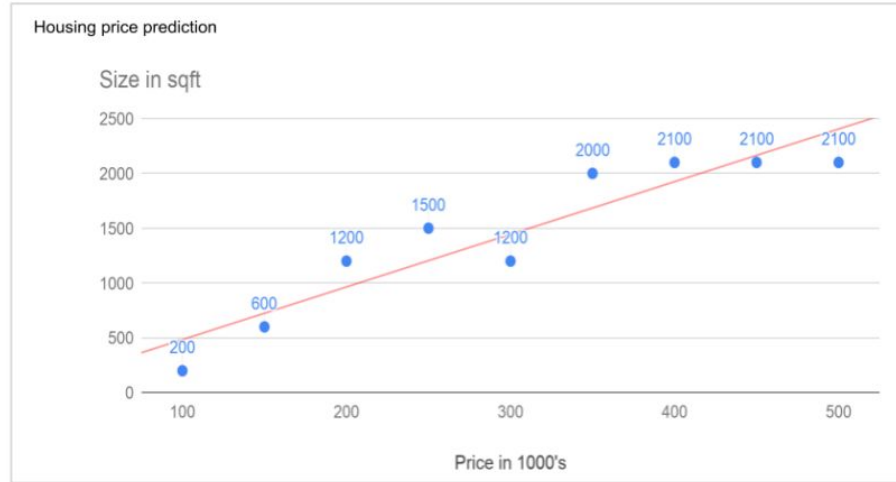
Deep Learning



Clustering, Classification, Regression



Consider the following figure that shows a plot of house prices versus its size in sq. ft.



After plotting various data points on the XY plot, we draw a best-fit line to do our predictions for any other house given its size. You will feed the known data to the machine and ask it to find the best fit line. Once the best fit line is found by the machine, you will test its suitability by feeding in a known house size, i.e. the Y-value in the above curve. The machine will now return the estimated X-value, i.e. the expected price of the house. The diagram can be extrapolated to find out the price of a house which is 3000 sq. ft. or even larger. This is called regression in statistics. Particularly, this kind of regression is called linear regression as the relationship between X & Y data points is linear.

Outline of this tutorial

The goal is not to teach you machine learning in 1 hour, which is not possible, but for you to learn the basic tools and understand how/when they would be useful

- Feature Engineering
- Regression: Linear and Logistic (Supervised, prediction)
- PCA (Unsupervised, classification)
- SVM (supervised, classification)
- KNN (supervised, classification and regression)
- K-means (unsupervised, clustering)
- Naive Bayes (supervised, classification)
- Decision Trees (supervised, classification)

Feature Engineering

- Imputation = Handle missing data
 - Use the mean value to fill the blank
 - Use the most often appeared thing to fill the blank
 - The above 2 is because of the principle of normal distribution (where the values in the distribution are more likely to occur closer to the mean rather than the edges). But this could cause trouble as there will be places where you could make huge mistakes and make the data quality worse, so be careful.
 - A few other ways to go about this include replacing missing values by picking the value from a normal distribution with the mean and standard deviation of the corresponding existing values or even replacing the missing value with an arbitrary value.

Feature Engineering

- Discretization = Grouping the same information
 - Examples like groups weekdays (Monday, Tuesday) and weekends (Saturday/Sunday), data that has the same frequency or interval.
 - This could help prevent data from overfitting but comes at the cost of loss of granularity of data.
- Categorical Encoding
 - One hot encoding: basically anything that does not necessarily have an order and you want it to be a number, one hot encode it. Problem is that you will have a lot of extra features to work with,
 - Ordinal encoding: assign number to each observation, anything that have order could be encoded this way.

Feature Engineering

- Handling Outliers

- Removal: The records containing outliers are removed from the distribution. However, the presence of outliers over multiple variables could result in losing out on a large portion of the datasheet with this method.
- Replacing values: The outliers could alternatively be treated as missing values and replaced by using appropriate imputation.
- Capping: Capping the maximum and minimum values and replacing them with an arbitrary value or a value from a variable distribution.

Feature Engineering

- Variable Transformations

- Variable transformation techniques could help with normalizing skewed data. One such popularly used transformation is the logarithmic transformation. Logarithmic transformations operate to compress the larger numbers and relatively expand the smaller numbers. This in turn results in less skewed values especially in the case of heavy-tailed distributions. Other variable transformations used include Square root transformation and Box cox transformation which is a generalization of the former two.

Feature Engineering

- Scaling

- Min-Max Scaling: This process involves the rescaling of all values in a feature in the range 0 to 1. In other words, the minimum value in the original range will take the value 0, the maximum value will take 1 and the rest of the values in between the two extremes will be appropriately scaled.
- Standardization/Variance scaling: All the data points are subtracted by their mean and the result is divided by the distribution's variance to arrive at a distribution with a 0 mean and variance of 1.

What is Linear Regression? What is it used for?

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

$$\hat{\boldsymbol{\beta}} = \operatorname{argmin}_{\mathbf{b} \in \mathbb{R}^p} S(\mathbf{b}) = (X^T X)^{-1} X^T \mathbf{y} .$$

- Linear Regression is exactly what it sounds like: a linear model, that represent the relationship between the dependent variable (what you want to predict) and the independent variable (the features you have for prediction)
- Regularization: L1 and L2.
- Can use regularization to prevent overfitting: LASSO (L1 norm, used to select features) and Ridge (L2 norm)
- Cross-validation

What is Linear Regression? What is it used for?

- The goodness of the model is usually measured by the mean squared error, sometimes called MSE

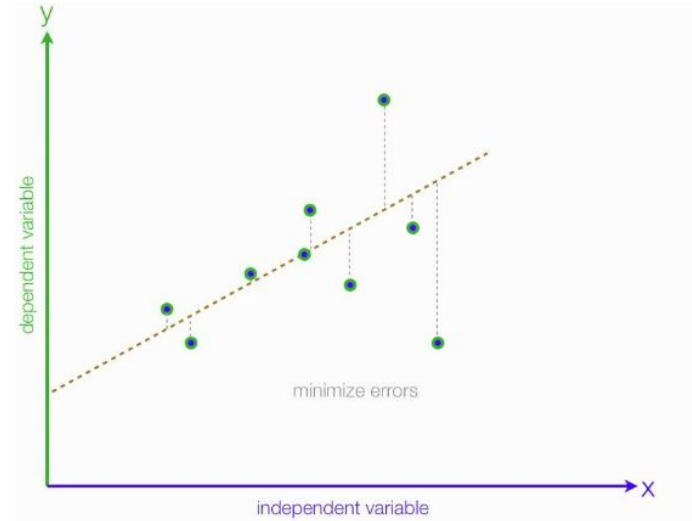
$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

MSE = mean squared error

n = number of data points

Y_i = observed values

\hat{Y}_i = predicted values



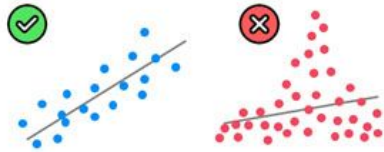
Assumptions of Linear Regression

Assumptions of Linear Regression



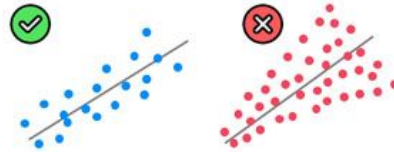
1. Linearity

(Linear relationship between Y and each X)



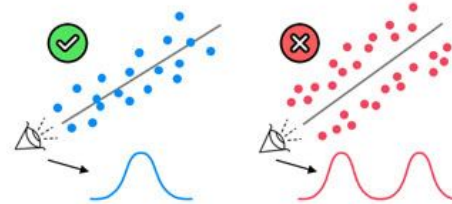
2. Homoscedasticity

(Equal variance)



3. Multivariate Normality

(Normality of error distribution)



4. Independence

(of observations. Includes "no autocorrelation")



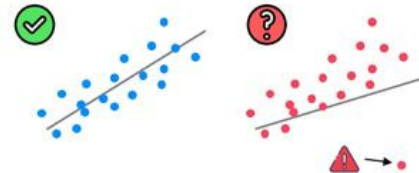
5. Lack of Multicollinearity

(Predictors are not correlated with each other)



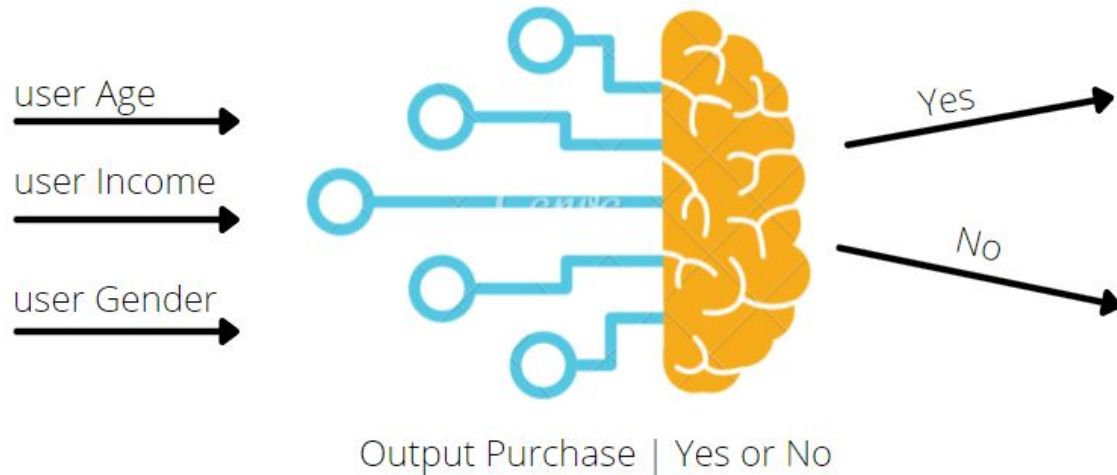
6. The Outlier Check

(This is not an assumption, but an "extra")

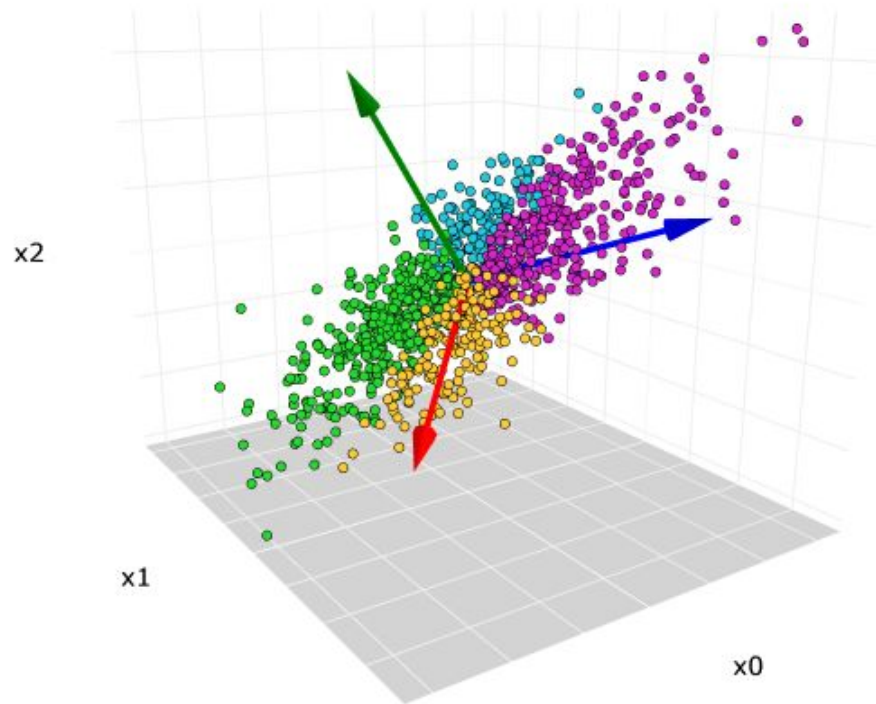


What is logistic Regression

Logistic Regression

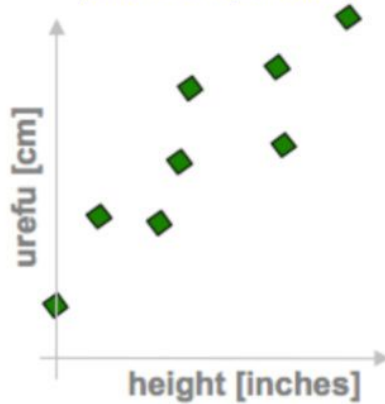


PCA: Principal Component Analysis

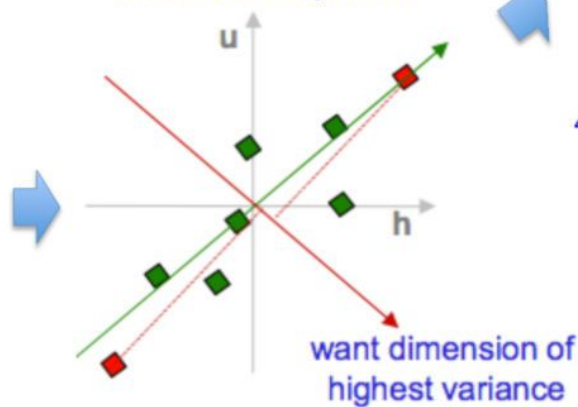


PCA in a nutshell

1. correlated hi-d data
("urefu" means "height" in Swahili)



2. center the points



3. compute covariance matrix

$$\begin{matrix} & h & u \\ h & 2.0 & 0.8 \\ u & 0.8 & 0.6 \end{matrix} \rightarrow \text{cov}(h, u) = \frac{1}{n} \sum_{i=1}^n h_i u_i$$

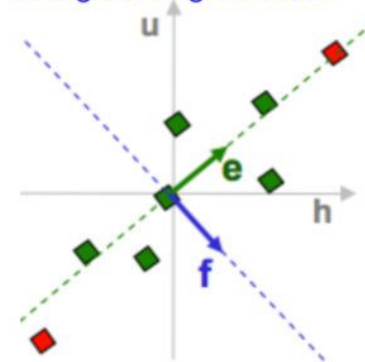
4. eigenvectors + eigenvalues

$$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \begin{pmatrix} e_h \\ e_u \end{pmatrix} = \lambda_e \begin{pmatrix} e_h \\ e_u \end{pmatrix}$$

$$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \begin{pmatrix} f_h \\ f_u \end{pmatrix} = \lambda_f \begin{pmatrix} f_h \\ f_u \end{pmatrix}$$

$\text{eig}(\text{cov}(\text{data}))$

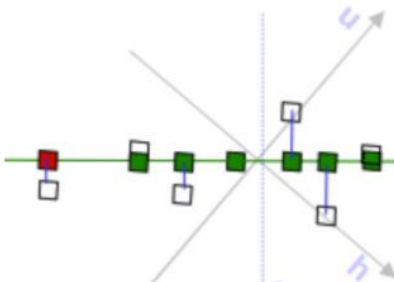
5. pick $m < d$ eigenvectors
w. highest eigenvalues



6. project data points to those eigenvectors

$$x'_e = x^T e = \sum_{j=1}^a x_{ij} e_j$$

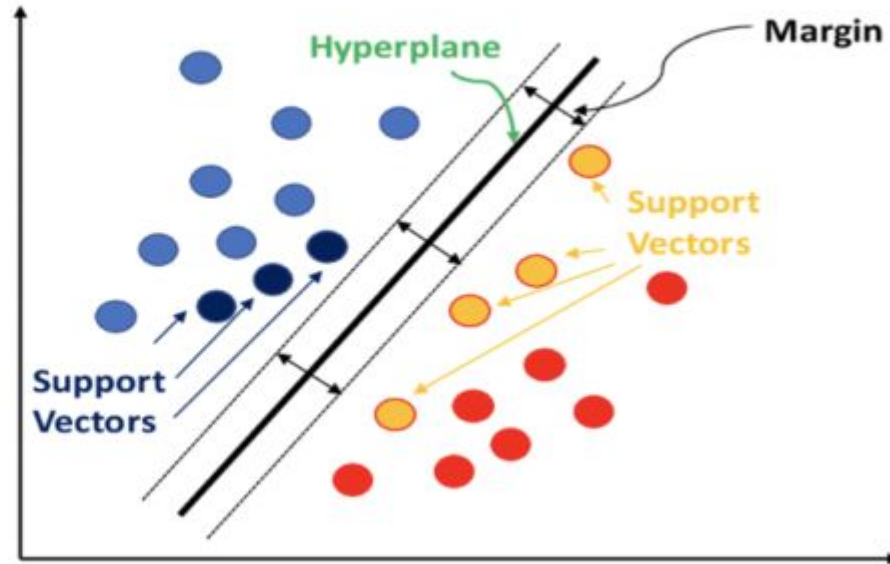
7. uncorrelated low-d data



SVM: Support Vector Machine

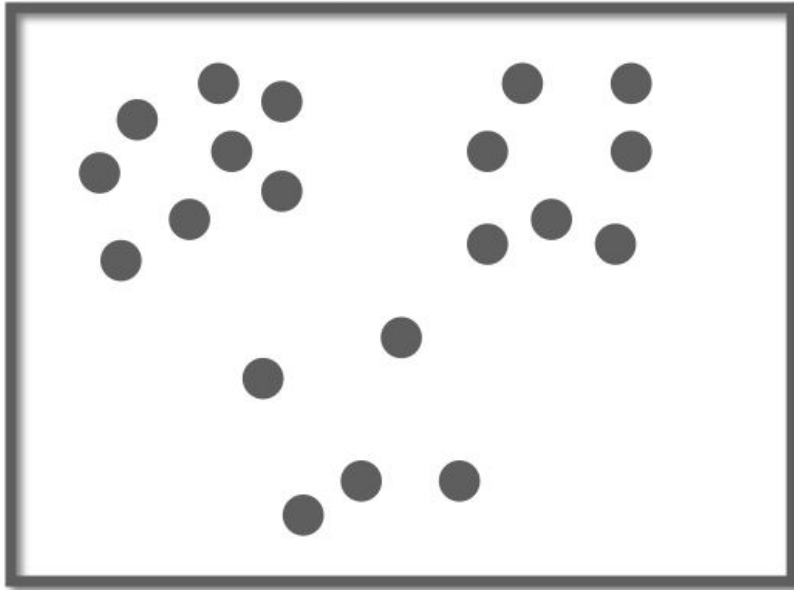
WHAT IS A

**SUPPORT
VECTOR
MACHINE?**



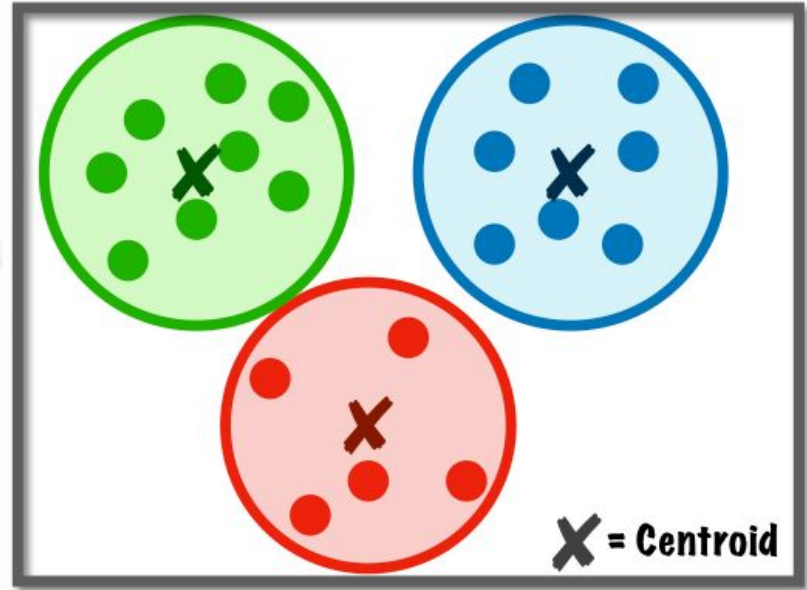
K-means (Unsupervised)

Unlabelled Data



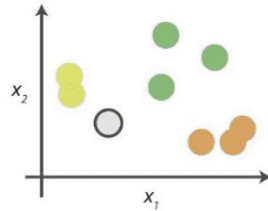
K-means

Labelled Clusters



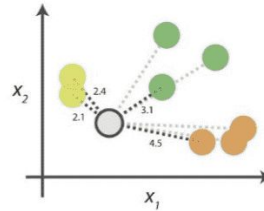
KNN: K-nearest-neighbors (Supervised)

0. Look at the data











Say you want to classify the grey point into a class. Here, there are three potential classes - lime green, green and orange.

1. Calculate distances






Start by calculating the distances between the grey point and all other points.

2. Find neighbours




Point Distance	
 ...  2.1	→ 1st NN
 ...  2.4	→ 2nd NN
 ...  3.1	→ 3rd NN
 ...  4.5	→ 4th NN

Next, find the nearest neighbours by ranking points by increasing distance. The nearest neighbours (NNs) of the grey point are the ones closest in dataspace.

3. Vote on labels

Class	# of votes
	2
	1
	1

→

Class  wins the vote!
Point  is therefore predicted to be of class .

Vote on the predicted class labels based on the classes of the k nearest neighbours. Here, the labels were predicted based on the k=3 nearest neighbours.

What is KNN

Step 1: Determine the value for K

Step 2: Calculate the distances between the new input (test data) and all the training data. The most commonly used metrics for calculating distance are Euclidean, Manhattan and Minkowski

Step 3: Sort the distance and determine k nearest neighbors based on minimum distance values

Step 4: Analyze the category of those neighbors and assign the category for the test data based on majority vote

Step 5: Return the predicted class

Naive Bayes

Naive Bayes

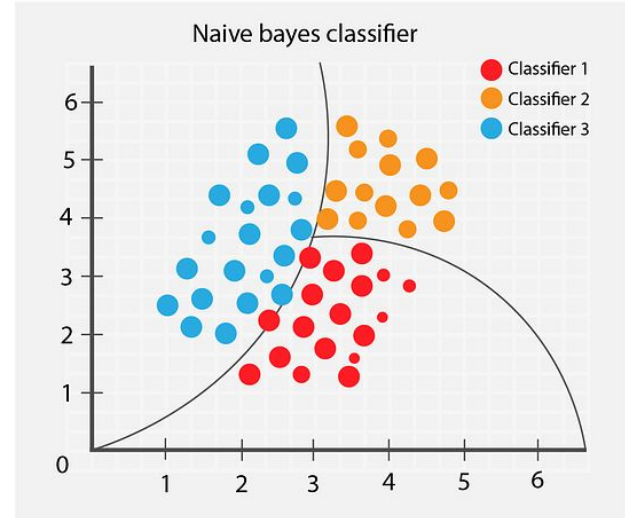


In machine learning, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

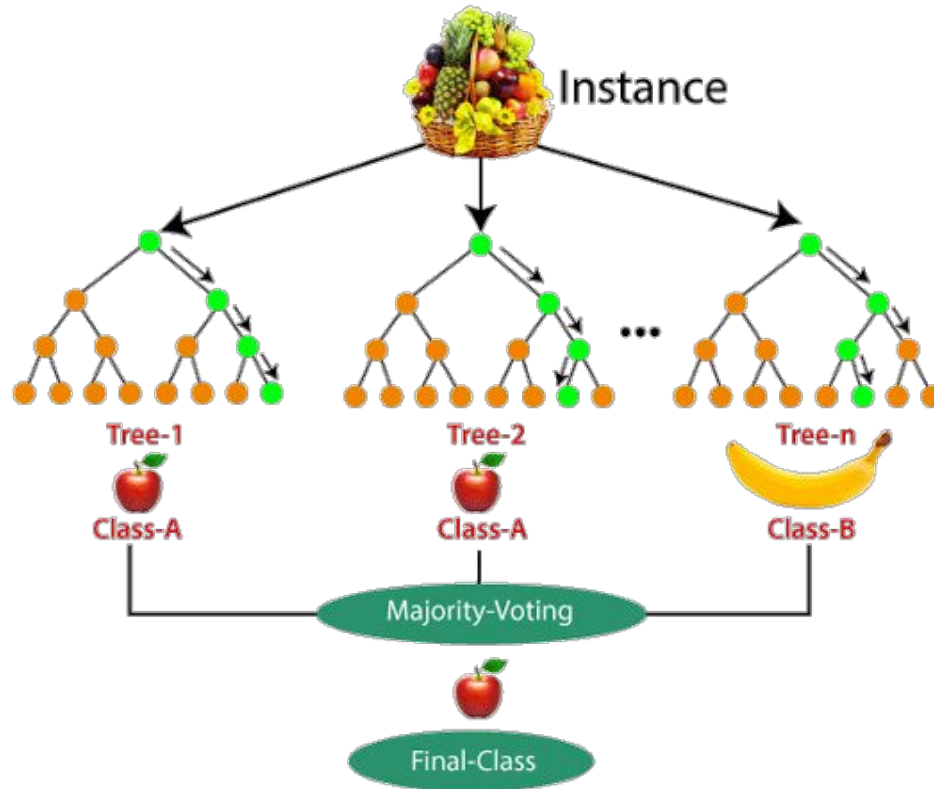
$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$



Decision Trees



Measurement Metrics

True negative

Predicted negative
Actual negative

False positive

Predicted positive
Actual negative

False negative

Predicted negative
Actual positive

True positive

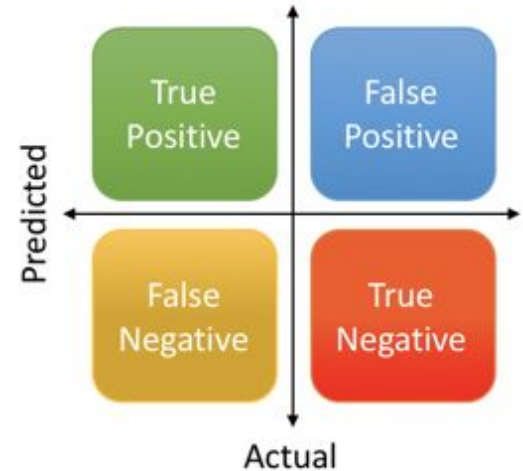
Predicted positive
Actual positive

Terms

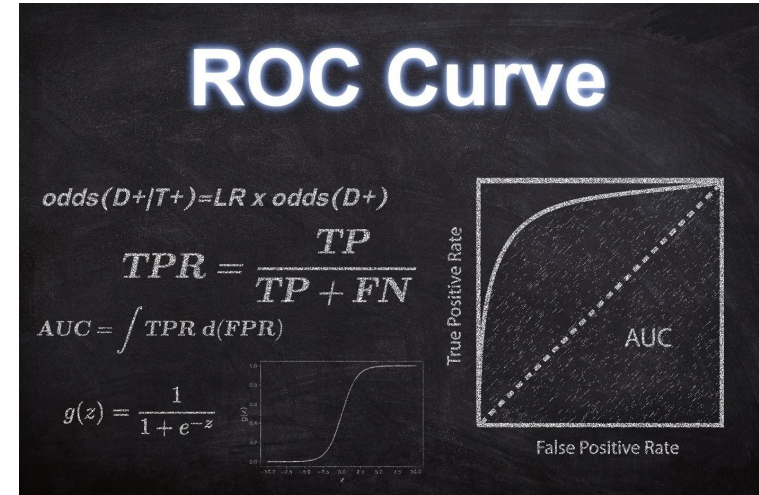
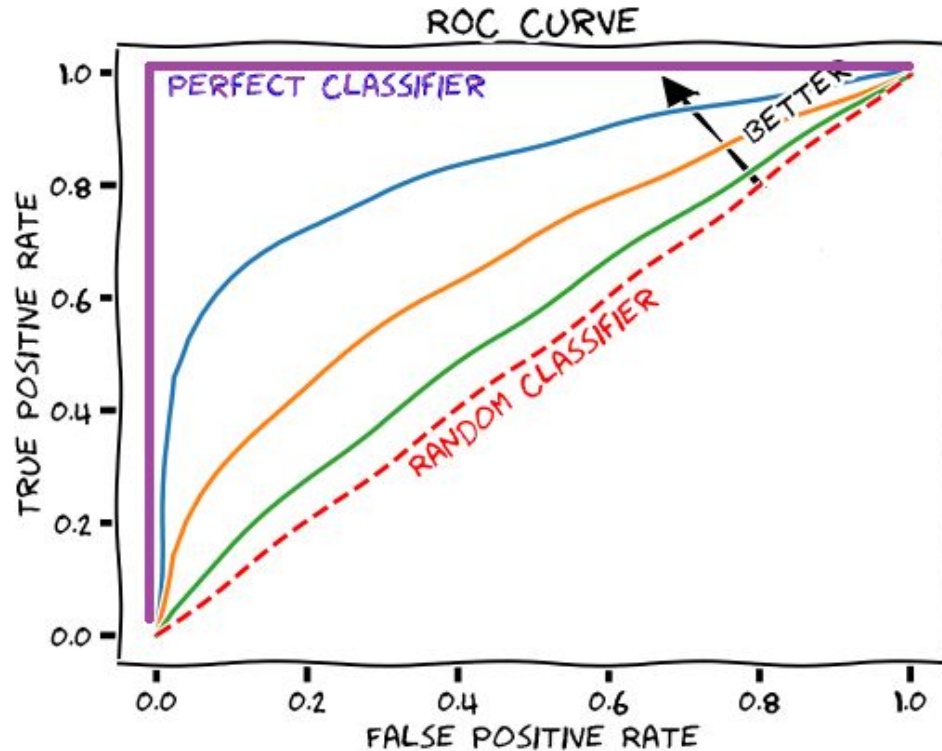
$$\text{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total}}$$



AUC: Area under the Roc Curve



Precision Recall Curve

