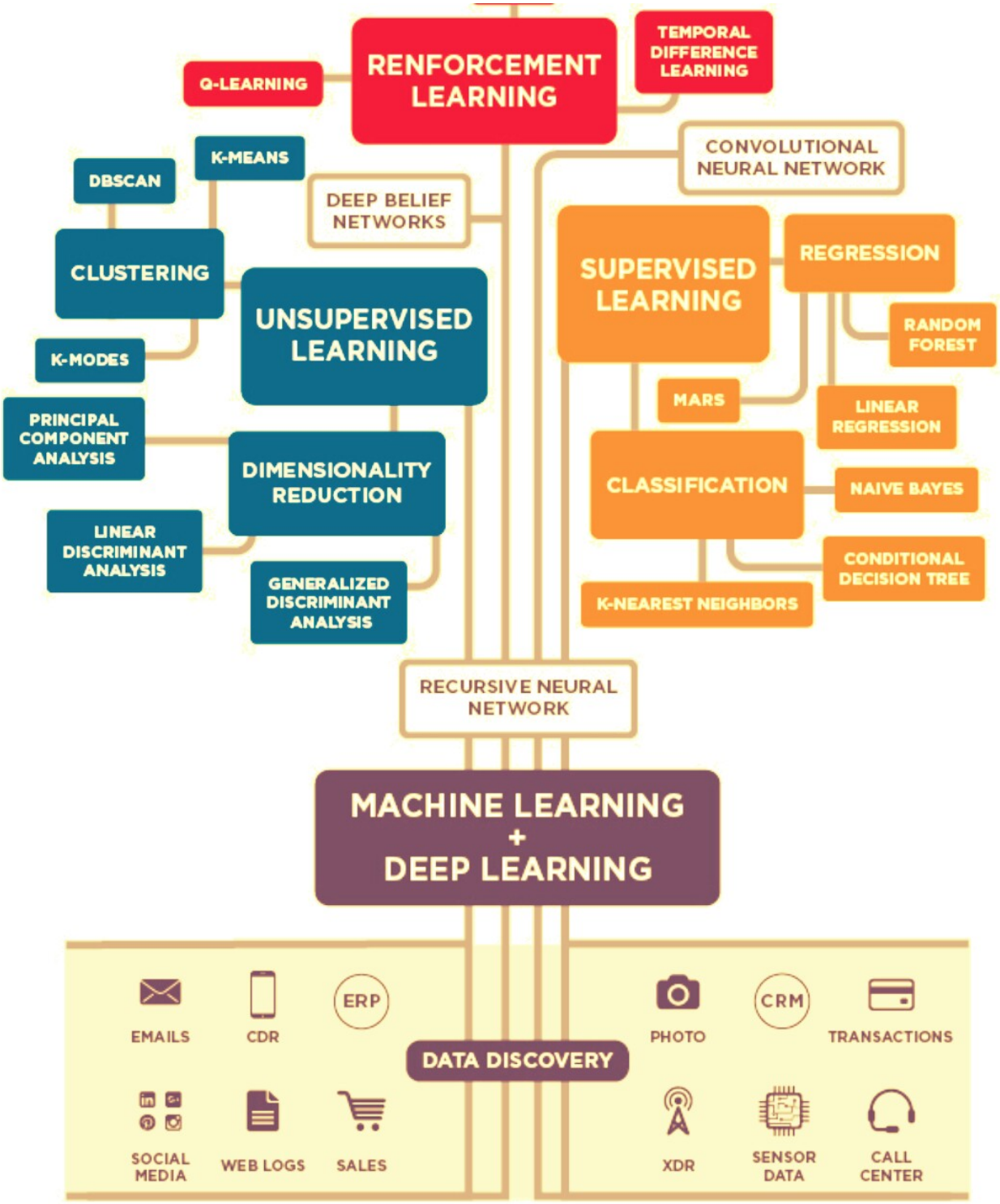
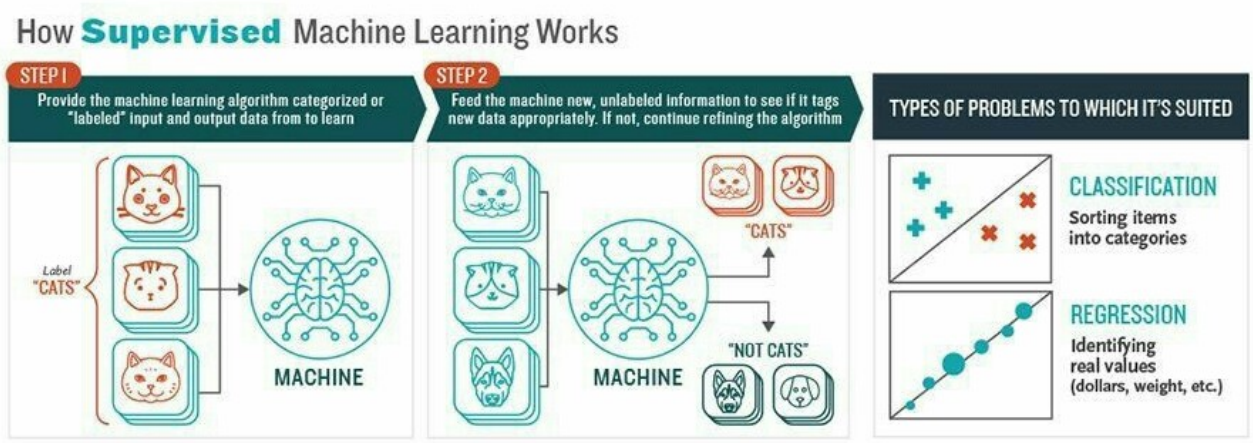
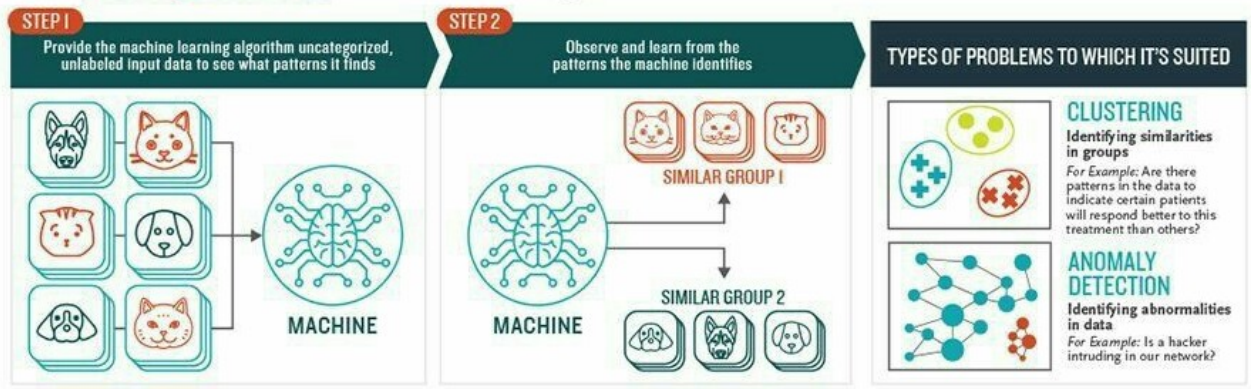
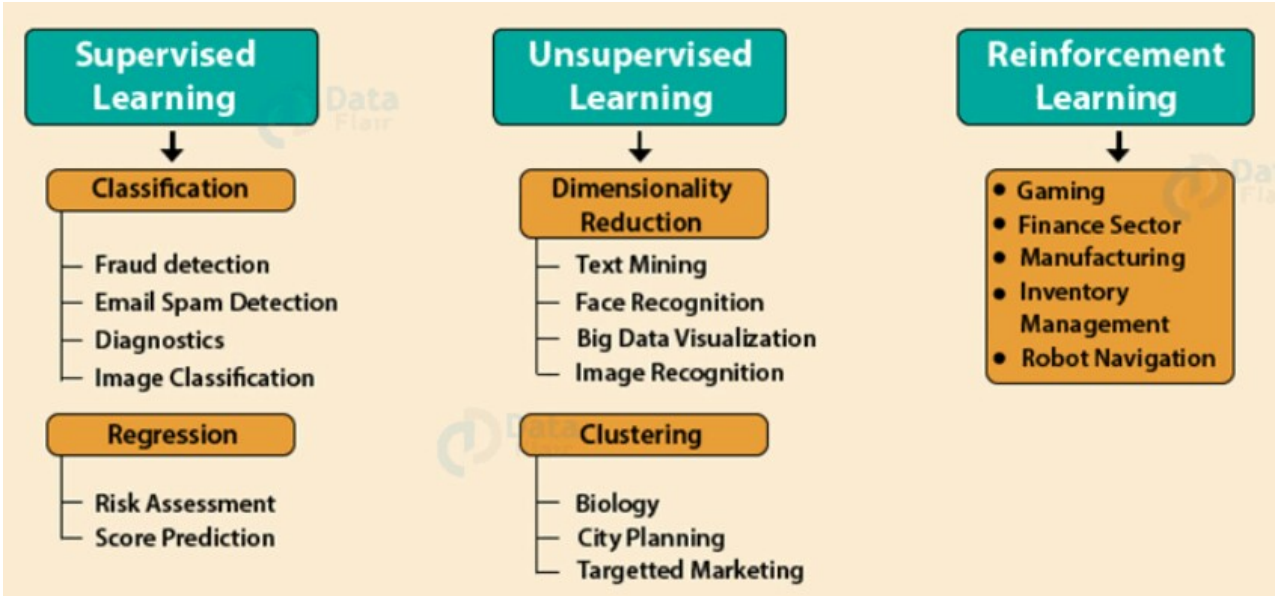
Getting Started

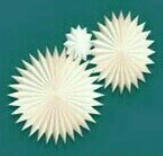
kaggle.com/getting-started/191390




How Supervised Machine Learning Works
STEP 1
Provide the machine learning algorithm categorized or "labeled" input and output data from to learn
Label "CATS"
MACHINE
STEP 2
Feed the machine new, unlabeled information to see if it tags new data appropriately. If not, continue refining the algorithm
MACHINE
"CATS"
"NOT CATS"
TYPES OF PROBLEMS TO WHICH IT'S SUITED
CLASSIFICATION
Sorting items into categories
REGRESSION
Identifying real values (dollars, weight, etc.)


How Unsupervised Machine Learning Works
STEP 1
Provide the machine learning algorithm uncategorized, unlabeled input data to see what patterns it finds
MACHINE
STEP 2
Observe and learn from the patterns the machine identifies
SIMILAR GROUP 1
SIMILAR GROUP 2
MACHINE
TYPES OF PROBLEMS TO WHICH IT'S SUITED
CLUSTERING
Identifying similarities in groups
For Example: Are there patterns in the data to indicate certain patients will respond better to this treatment than others?
ANOMALY DETECTION
Identifying abnormalities in data
For Example: Is a hacker intruding in our network?


Supervised Learning
Classification
Fraud detection
Email Spam Detection
Diagnostics
Image Classification
Regression
Risk Assessment
Score Prediction
Unsupervised Learning
Dimensionality Reduction
Text Mining
Face Recognition
Big Data Visualization
Image Recognition
Clustering
Biology
City Planning
Targetted Marketing
Reinforcement Learning
Gaming
Finance Sector
Manufacturing
Inventory Management
Robot Navigation

Component	Supervised machine Learning	Unsupervised machine Learning
 Concept comparison	Algorithms in Supervised Learning learn on a labelled dataset containing labelled keys from which evaluation process will be based on	Algorithms in Unsupervised Learning are based on unlabelled data, trying to make sense through extraction of features and patterns without supervision
 Model comparison	Supervised learning model training data includes both the input and desired results	Unsupervised learning model is not provided with correct result during training
 Training comparison	Supervised Learning uses training data to infer model and apply the model to test data . Therefore we can test the model	Unsupervised Learning has no training data ,model inference and application both rely on test data exclusively. We cannot test the model.
 Algorithm comparison	Supervised Learning Algorithms are • Classification (Classification of labelled data) • Regression (Predicting trends using previous labelled data)	Unsupervised Learning Algorithms use • Clustering(Finding patterns and groupings from unlabelled data)
 Input data comparison	Supervised Learning uses labelled input data	Unsupervised Learning uses unlabelled input data
 Model comparison	Supervised learning model is trained on how to do tasks	Unsupervised learning needs no training, but figures out patterns and meanings from data on it's own
 Computation comparison	Unsupervised learning does it computations in real time in a way in which input data is analysed and labelled in the presence of learners which minimises complications in mastering different methods of learning and raw data classification	Supervised machine learning uses offline analysis
 Accuracy comparison	Supervised machine learning results are accurate and reliable	Unsupervised machine learning results produce moderate accurate and reliable results

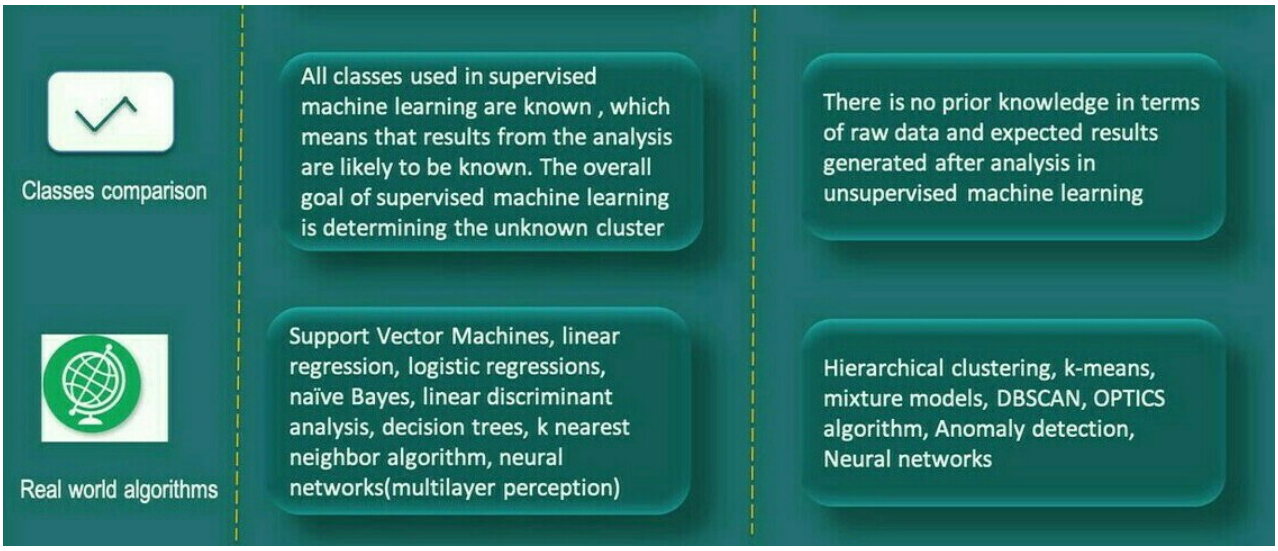
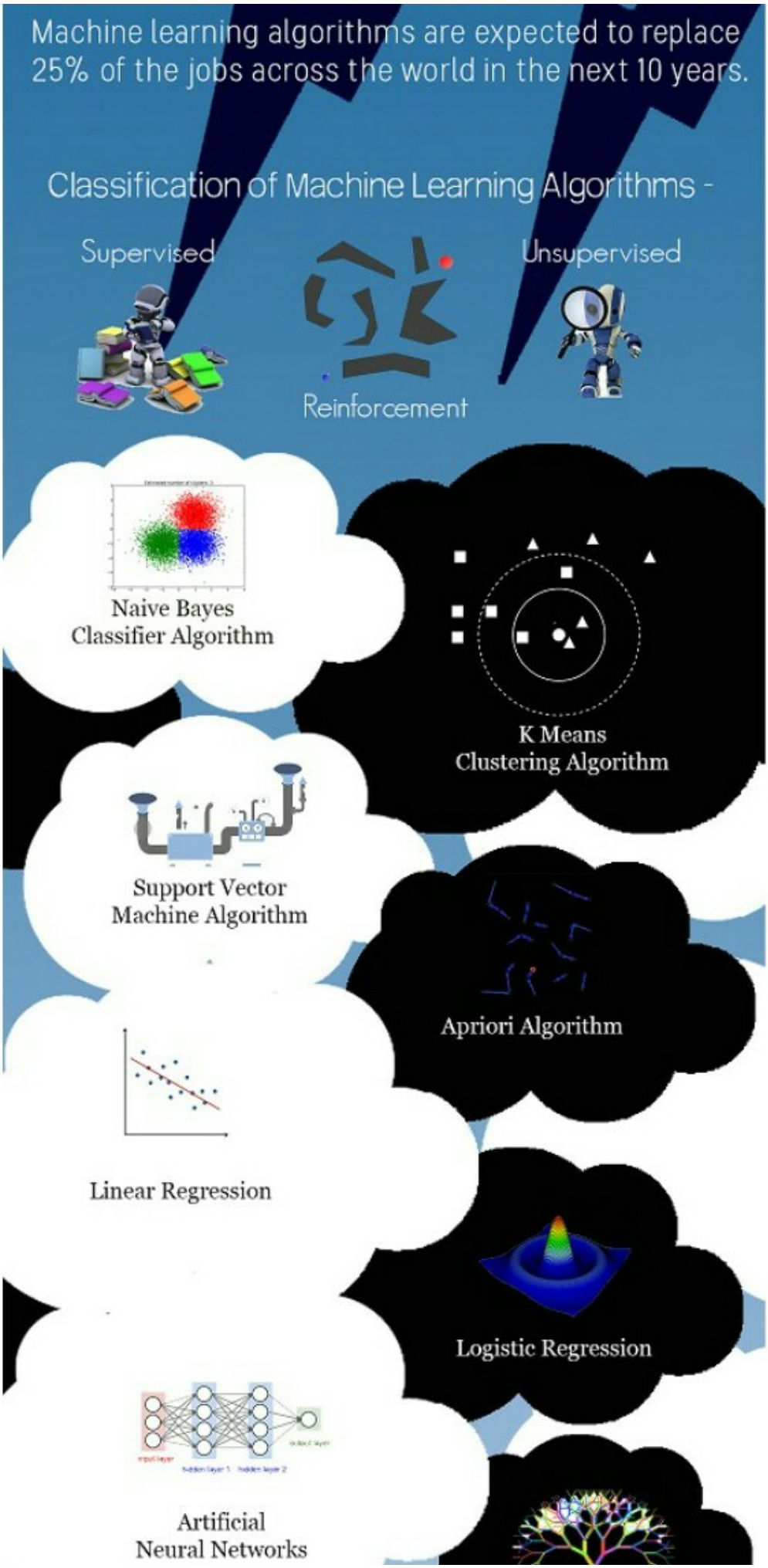

Classes comparison
All classes used in supervised machine learning are known , which means that results from the analysis are likely to be known. The overall goal of supervised machine learning is determining the unknown cluster
There is no prior knowledge in terms of raw data and expected results generated after analysis in unsupervised machine learning
Real world algorithms
Support Vector Machines, linear regression, logistic regressions, naïve Bayes, linear discriminant analysis, decision trees, k nearest neighbor algorithm, neural networks(multilayer perception)
Hierarchical clustering, k-means, mixture models, DBSCAN, OPTICS algorithm, Anomaly detection, Neural networks

CHART TYPE	CATEGORICAL COMPARISON ordered/unordered, share of total, distribution	VALUES OVER TIME Showing values against a time series	CORRELATION Comparing two or more values to show a relation	MEASURES ONLY/KPI Displaying single values that may/may not be comparable	GEOGRAPHICAL Showing values that have specific locations or areas
horizontal bars 	**Use** for ordered/ranked and share of total data **Example:** top 10 customers by sales **Note:** better than vertical when longer label names			**Only use** if it's valid to compare between the different measures **Example:** % delivered in full vs % delivered on time	**Use** for data that's geographical, but when analysis isn't geographically important **Example:** sales by region when location of the regions is not important to the analysis
vertical bars 	**Use** for unordered and distribution **Examples:** •sales by product category •purchases by age range (distribution)	**Use** when comparing discrete time data (e.g. years, months), and when gaps in data exist (e.g. missing weekend data)		**Only use** if it's valid to compare between the different measures	**Use** for data that's geographical, but when the analysis isn't geographically important **Example:** sales by state
table 	**Use** for ordered or unordered when[1]: •there is a hierarchy of categorical data on rows •subtotals are used •there are extra values/ properties per item to show **Example:** regions and customers	**Use** when the values over time are significant, as opposed to the trend over time (see Line). Same rules as for categorical comparison **Example:** sales by year on columns, customers by region on rows	**Use** when finding correlation between different categories (on rows) **Example:** correlation between different items purchased at the same time; heat map of product categories A& B	**Only use** if wanting a tabular format. Otherwise consider using text (below) for more readable formats	**Use** for data that's geographical, but when the analysis isn't geographically important **Example:** •regions and customers •showing contact details and market category
text 				**Use** for measures without categories **Note:** use text formatting and size proportionately to show significance; the number should be the largest text	
line 	**Use** when comparing multiple categorical series that have a clear order **Example:** Outstanding invoices by age	**Use** when plotting data over time[2] Good for: •detailed levels of time (e.g. date) •trends over time **Example:** amount of cases by date for different priorities	**Use** for showing correlation of measures: •instead of points/ bubbles when more than three measures •good for showing a trend that is correlated **Example:** temp, precipitation and ice cream sales by day		
area 	**Use** for showing a distribution when there's only a single series or a trend/shape **Example:** case load by time of day	**Use** as an alternative to line when[2]: •there's only one series, or the series are part of a whole and are shown as a stacked area •the values can be accumulated **Example:** total sales by week			
points & bubbles 	**Only use** when axes don't start at zero and bar markers are in use **Example:** quantity sold by store (axis starts at 5M because variation between stores is small compared to total)		**Use** when showing the correlation of 2-3 different measures **Example:** opportunity closed amount (y), opportunity estimated amount (x), estimated closed probability (bubble size) by opportunity		See Map Points
bar marker 	**Use** in conjunction with horizontal or vertical bars for a comparison or target value **Example:** sales by quarter, with bars to compare vs quarter in prior year	**Use** in conjunction with vertical bars for a comparison with a target value OR instead of a line when data for time periods is missing **Example:** budget vs actual, or hours worked per week (missing weekends)			
pie, donut, gauge 	**Only use** pie/donut when accurate comparison isn't required and it is important to signal that segments make up a whole value **Example:** budget by department, project and team			**Only** use a circular gauge when the value shown is a percentage or has a clear target value which represents the complete circle **Example:** Project percent complete	
treemap	**Use** when the categorical data contains multiple				

	levels of categories simultaneously **Example:** budget by department, project and team				
map shapes 					**Use** when data represents geographical areas and when it's common to look at this by shapes (e.g. US states), OR location is important for the analysis (e.g. where to place distribution centers by looking at delivery times by postal code)
map points 					**Use** when data has a specific location (latitude and longitude) and location is important for the analysis
word cloud 	**Only use** when the categorical data does not require accurate comparison, and when the categorical information is written language **Example:** show sentiment and amount of use of terms in relation to a brand				


Machine learning algorithms are expected to replace 25% of the jobs across the world in the next 10 years.
Classification of Machine Learning Algorithms -
Supervised
Unsupervised
Reinforcement
Naive Bayes Classifier Algorithm
K Means Clustering Algorithm
Support Vector Machine Algorithm
Apriori Algorithm
Linear Regression
Logistic Regression
Artificial Neural Networks

TYPE	NAME	DESCRIPTION	ADVANTAGES	DISADVANTAGES
Linear	Linear regression	**The "best fit" line** through all data points. Predictions are numerical.	Easy to understand -- you clearly see what the biggest drivers of the model are.	X Sometimes too simple to capture complex relationships between variables. X Tendency for the model to "overfit".
Linear	Logistic regression	The adaptation of **linear regression** to problems of classification (e.g., yes/no questions, groups, etc.)	Also easy to understand.	X Sometimes too simple to capture complex relationships between variables. X Tendency for the model to "overfit".
Tree-based	Decision tree	A graph that uses a **branching method** to match all possible outcomes of a decision.	Easy to understand and implement.	X Not often used on its own for prediction because it's also often too simple and not powerful enough for complex data.
Tree-based	Random Forest	Takes the average of many decision trees, each of which is made with a sample of the data. Each tree is weaker than a full decision tree, but **by combining them we get better overall performance.**	A sort of "wisdom of the crowd". Tends to result in very high quality models. Fast to train.	X Can be slow to output predictions relative to other algorithms. X Not easy to understand predictions.
Tree-based	Gradient Boosting	Uses even weaker decision trees, that are increasingly **focused on "hard" examples.**	High-performing.	X A small change in the feature set or training set can create radical changes in the model. X Not easy to understand predictions.
Neural networks	Neural networks	Mimics the behavior of the brain. Neural networks are interconnected neurons that pass messages to each other. Deep learning uses several **layers of neural networks put one after the other.**	Can handle extremely complex tasks - no other algorithm comes close in image recognition.	X Very, very slow to train, because they have so many layers. Require a lot of power. X Almost impossible to understand predictions.

DATA PREPROCESSING

Getting Started with Machine Learning

python™

Step 1: Importing the required Libraries

These Two are essential libraries which we will import every time.
NumPy is a Library which contains Mathematical functions.
Pandas is the library used to import and manage the data sets.

Step 2: Importing the Data Set

Data sets are generally available in .csv format. A CSV file stores tabular data in plain text. Each line of the file is a data record. We use the read_csv method of the pandas library to read a local CSV file as a dataframe. Then we make separate Matrix and Vector of independent and dependent variables from the dataframe.

Step 3: Handling the Missing Data

The data we get is rarely homogeneous. Data can be missing due to various reasons and needs to be handled so that it does not reduce the performance of our machine learning model. We can replace the missing data by the Mean or Median of the entire column. We use Imputer class of sklearn.preprocessing for this task.

Step 4: Encoding Categorical Data

Categorical data are variables that contain label values rather than numeric values. The number of possible values is often limited to a fixed set. Example values such as "Yes" and "No" cannot be used in mathematical equations of the model so we need to encode these variables into numbers. To achieve this we import

LabelEncoder class from sklearn.preprocessing library.

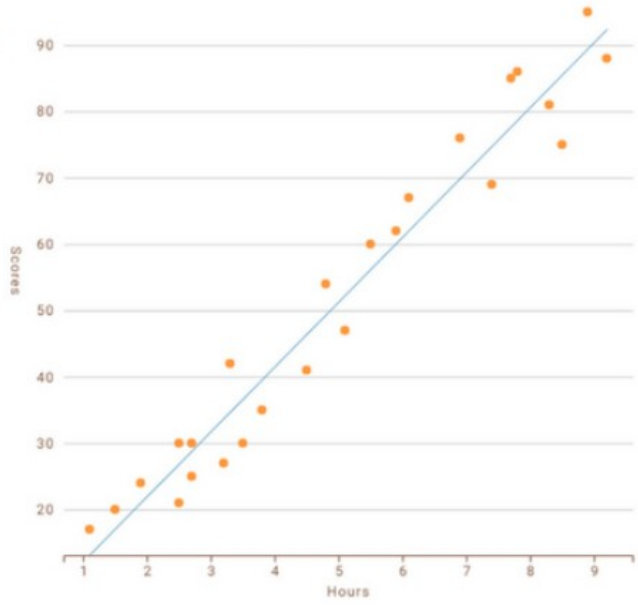
Step 5: Splitting the dataset into test set and training set

We make two partitions of dataset one for training the model called training set and other for testing the performance of the trained model called test set. The split is generally 80/20. We import train_test_split() method of sklearn.crossvalidation library.

Step 6: Feature Scaling

Most of the machine learning algorithms use the Euclidean distance between two data points in their computations, features highly varying in magnitudes, units and range pose problems. High magnitudes features will weigh more in the distance calculations than features with low magnitudes. Done by Feature standardization or Z-score normalization. StandardScalar of sklearn.preprocessing is imported.

SIMPLE LINEAR REGRESSION



Predicting a response using a single feature.

It is a method to predict dependent variable (Y) based on values of independent variables (X). It is assumed that the two variables are linearly related. Hence, we try to find a linear function that predicts the response value(y) as accurately as possible as a function of the feature or independent variable(x).

How to find the best fit line?

In this regression model, we are trying to minimize the errors in prediction by finding the "line of best fit" – the regression line from the errors would be minimal. We are trying to minimize the length between the observed value (Yi) and the predicted value from our model (Yp).

Predicted Value

Observed Value

$$y_i \quad y_p$$

$$\min \{\mathrm{SUM}(y_i - y_p)^2\}$$

Dependent Variable

$$y = b_\circ + b_1 x_1$$

Independent Variable

In this regression task, we will predict the percentage of marks that a student is expected to score based upon the number of hours they studied.

Slope

$$Score = b_\circ + b_1 * hours$$

Y - intercept

STEP 1: PREPROCESS THE DATA

We will follow the same steps as in my previous infographic of Data Preprocessing.

- Import the Libraries.
- Import the DataSet.
- Check for Missing Data.
- Split the DataSet.
- Feature Scaling will be taken care by the Library we will use for Simple Linear Regression Model.

STEP 2: FITTING SIMPLE LINEAR REGRESSION MODEL TO THE TRAINING SET

To fit the dataset into the model we will use LinearRegression class from sklearn.linear_model

library. Then we make an object regressor of LinearRegression Class. Now we will fit the regressor object into our dataset using fit() method of LinearRegression Class.

STEP 3: PREDICTING THE RESULT

Now we will predict the observations from our test set. We will save the output in a vector Y_pred. To predict the result we use predict method of LinearRegression Class on the regressor we trained in the previous step.

STEP 4: VISUALIZATION

The final step is to visualize our results. We will use matplotlib.pyplot library to make Scatter Plots of our Training set results and Test set results to see how close our model predicted the Values

MULTIPLE LINEAR REGRESSION

Multiple linear regression attempts to model the relationship between two or more features and a response by fitting a linear equation to observed data. The steps to perform multiple linear regression are almost similar to that of simple linear regression. The difference lies in the evaluation. You can use it to find out which factor has the highest impact on the predicted output and how different variables relate to each other.

Dependent Variable

multiple independent Variables

$$y = b_\circ + b_1 x_1 + b_2 x_2 \ldots\ldots b_n x_n$$

ASSUMPTIONS

FOR A SUCCESSFUL REGRESSION ANALYSIS, IT'S ESSENTIAL TO VALIDATE THESE ASSUMPTIONS.

1. Linearity: The relationship between dependent and independent variables should be Linear.
2. Homoscedasticity (constant variance) of the errors should be maintained.
3. Multivariate Normality: Multiple regression assumes that the residuals are normally distributed.
4. Lack of Multicollinearity: It is assumed that there is little or no multicollinearity in the data. Multicollinearity occurs when the features (or independent variables) are not independent of each other.

NOTE

Having too many variables could potentially cause our model to become less accurate, especially if certain variables have no effect on the outcome or have a significant effect on other variables. There are various methods to select the appropriate variable like -

1. Forward Selection
2. Backward Elimination
3. Bi-directional Comparision

DUMMY VARIABLES

Gender
Female
Female
Male
Female
Male
Male
Male

Male	Female
0	1
0	1
1	0
0	1
1	0
1	0
1	0

Using categorical data in Multiple Regression Models is a powerful method to include non-numeric data types into a regression model.

Categorical data refers to data values which represent categories - data values with a fixed and unordered number of values, for instance, gender (male/female). In a regression model, these values can be represented by dummy variables - variables containing values such as 1 or 0 representing the presence or absence of the categorical value.

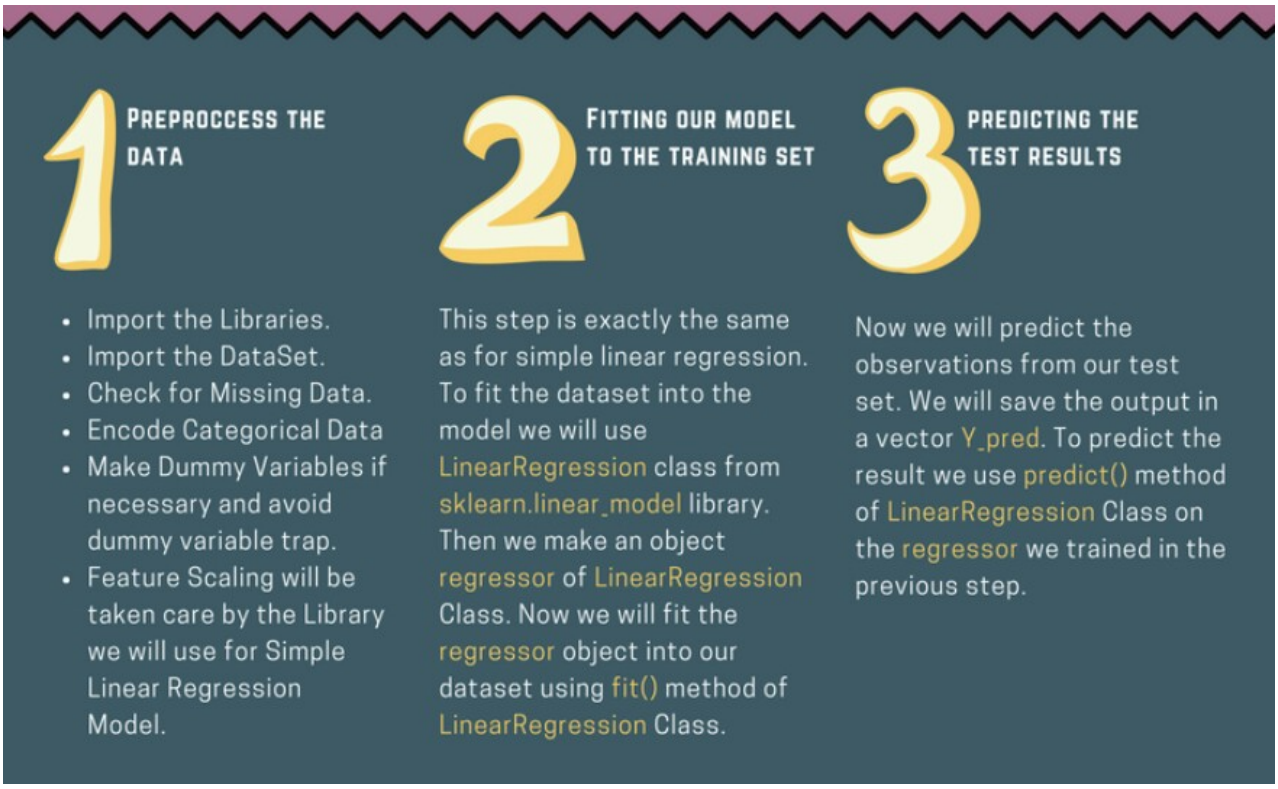
DUMMY VARIABLE TRAP

The Dummy Variable trap is a scenario in which two or more variables are highly correlated; in simple terms, one variable can be predicted from the others. Intuitively, there is a duplicate category: if we dropped the male category it is inherently defined in the female category (zero female value indicate male, and vice-versa).

The solution to the dummy variable trap is to drop one of the categorical variables - if there are m number of categories, use m-1 in the model, the value left out can be thought of as the reference value.

Dummy variable $D_2 = 1 - D_1$ Dummy Variable

$$y = b_\circ + b_1 x_1 + b_2 x_2 + b_3 D_1$$


1
PREPROCCESS THE DATA
Import the Libraries.
Import the DataSet.
Check for Missing Data.
Encode Categorical Data
Make Dummy Variables if necessary and avoid dummy variable trap.
Feature Scaling will be taken care by the Library we will use for Simple Linear Regression Model.
2
FITTING OUR MODEL TO THE TRAINING SET
This step is exactly the same as for simple linear regression. To fit the dataset into the model we will use LinearRegression class from sklearn.linear_model library. Then we make an object regressor of LinearRegression Class. Now we will fit the regressor object into our dataset using fit() method of LinearRegression Class.
3
PREDICTING THE TEST RESULTS
Now we will predict the observations from our test set. We will save the output in a vector Y_pred. To predict the result we use predict() method of LinearRegression Class on the regressor we trained in the previous step.

LOGISTIC REGRESSION

WHAT IS LOGISTIC REGRESSION

Logistic regression is used for a different class of problems known as classification problems. Here the aim is to predict the group to which the current object under observation belongs to. It gives you a discrete binary outcome between 0 and 1. A simple example would be whether a person will vote or not in upcoming elections.

How Does It Work?

Logistic Regression measures the relationship between the dependent variable (our label, what we want to predict) and the one or more independent variables (our features), by estimating probabilities using it's underlying logistic function.

Making Predictions

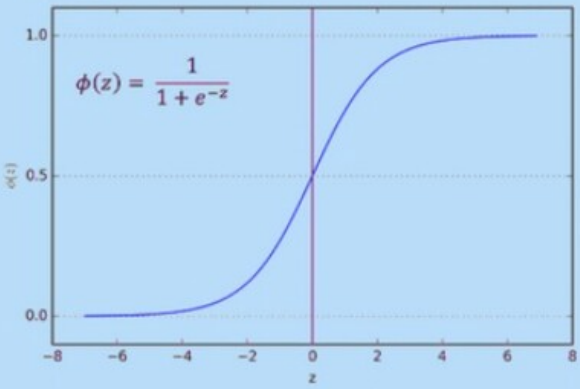
These probabilities must then be transformed into binary values in order to actually make a prediction. This is the task of the logistic function, also called the sigmoid function. This values between 0 and 1 will then be transformed into either 0 or 1 using a threshold classifier.

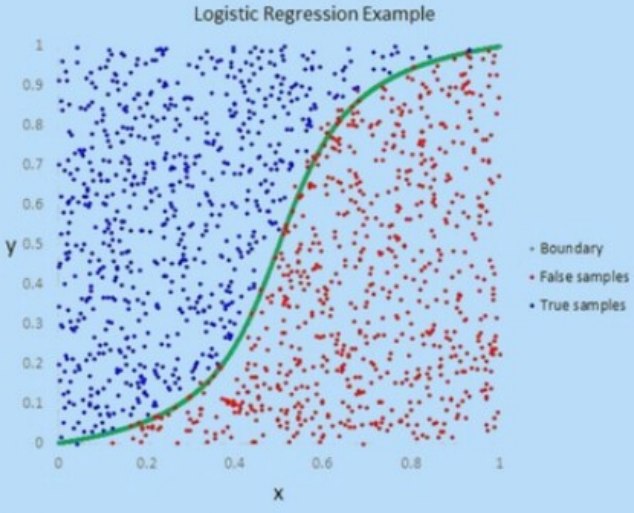
Logistic vs Linear

Logistic regression gives you a discrete outcome but linear regression gives a continuous outcome.

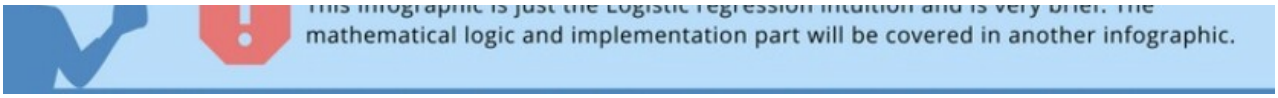
Sigmoid Function

The Sigmoid-Function is an S-shaped curve that can take any real-valued number and map it into a value between the range of 0 and 1, but never exactly at those limits.





This infographic is just the Logistic regression intuition and is very brief. The


This infographic is just the Logistic regression intuition and is very brief. The
mathematical logic and implementation part will be covered in another infographic.

K NEAREST NEIGHBOURS

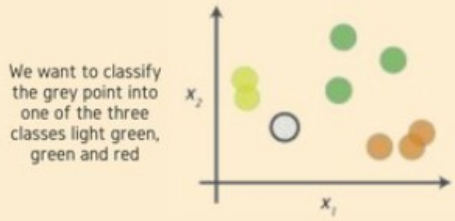
An Intuition to K-NN Classification Algorithm

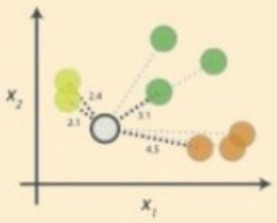
What is k-NN?

K-Nearest Neighbor algorithm is a simple yet most used classification algorithm. It can also be used for regression.

KNN is non-parametric (means that it does not make any assumptions on the underlying data distribution), instance-based (means that our algorithm doesnt explicitly learn a model. Instead, it chooses to memorize the training instances.) and used in a supervised learning setting.

k-NN is also called a lazy algorithm because it is instance based.





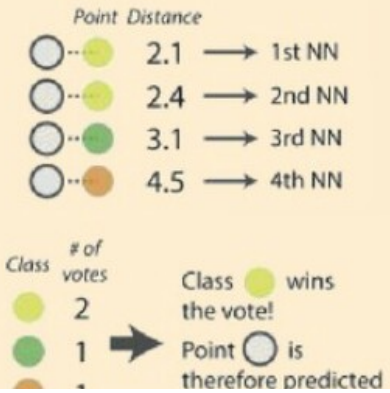
Start by calculating the distance between the grey point and k -nearest points

How Does k-NN Algorithm work?

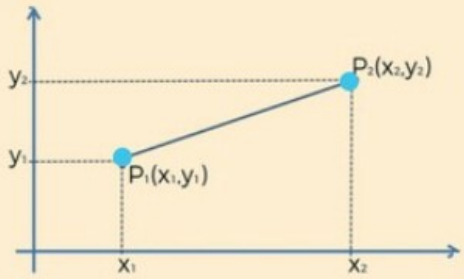
k-NN when used used for classification — the output is a class membership (predicts a class — a discrete value).
There are three key elements of this approach: a set of labeled objects, e.g., a set of stored records, a distance between objects, and the value of k, the number of nearest neighbors.

Making Predections

To classify an unlabeled object, the distance of this object to the labeled objects is computed, its k-nearest neighbors are identified, and the class label of the majority of nearest neighbors is then used to determine the class label of the object. For real-valued input variables, the most



popular distance measure is Euclidean distance.



Euclidean Distance between P_1 and $P_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Value of k

Finding the value of k is not easy. A small value of k means that noise will have a higher influence on the result and a large value make it computationally expensive. It depend a lot on your individual cases, sometimes it is best to run through each possible value for k and decide for yourself.

to be of class .

The Distance

Euclidean distance is calculated as the square root of the sum of the squared differences between a new point and an existing point across all input attributes .

Other popular distance measures include:

- Hamming Distance
- Manhattan Distance
- Minkowski Distance

SUPPORT VECTOR MACHINES

What is SVM?

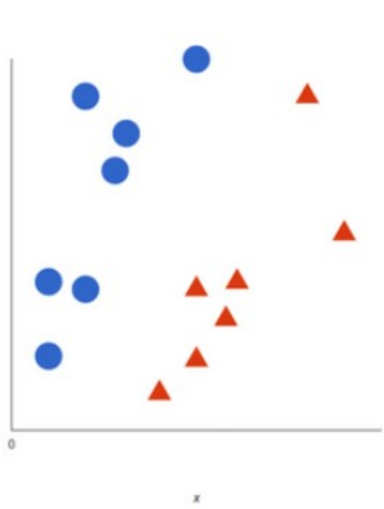
Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression. However, it is mostly used in classification problems.
In this algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features) with the value of each feature being the value of a particular coordinate.

How is the data classified?

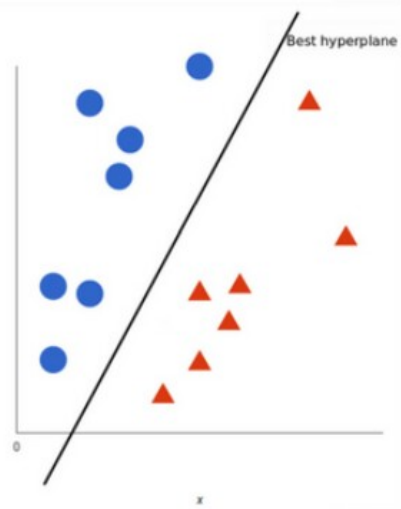
We perform classification by finding the hyperplane that differentiates the two classes very well. In other words the algorithm outputs an optimal hyperplane which categorizes new examples.

What is a optimal Hyper-Plane?

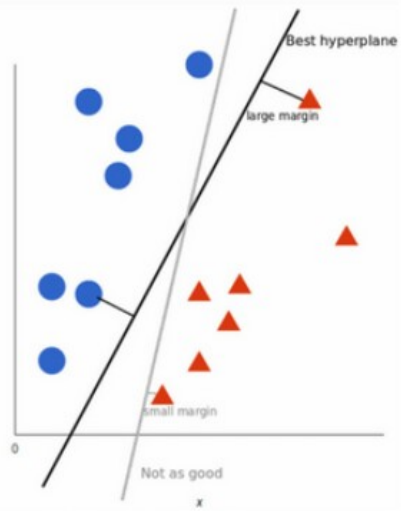
For SVM, it's the one that maximizes the margins from both tags. In other words: the hyperplane whose distance to the nearest element of each tag is the largest.



Our labeled data

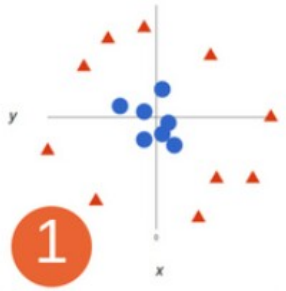


In 2D, the best hyper-plane is simply a line



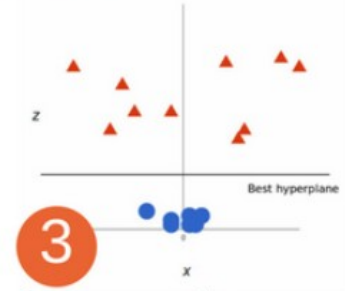
Not all hyper-planes are created equal

Nonlinear data

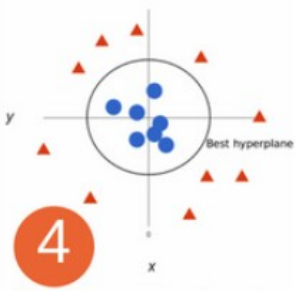


In above case we cannot draw a linear boundary.We will now add a third dimension. We create a new z dimension, and we rule that it be calculated a certain way that is convenient for us: $z = x^2 + y^2$ (equation for a circle)

This will give us a three-dimensional space. From a different perspective, the data is now in two linearly separated groups. All values for z would be always positive because z is the squared sum of both x and y.



Since we are in three dimensions now, the hyper-plane is a plane parallel to the x axis at a certain z. We choose the hyperplane, one that maximizes the margins from both of the Classes.



Now we map back to 2 dimensions.Our decision boundary is a circumference which separates both tags using SVM. We got a circle as a hyper-plane.

y (equation for a circle).

TUNING PARAMETERS

KERNEL

The learning of the hyperplane in linear SVM is done by transforming the problem using some linear algebra. This is where the kernel plays role. Polynomial and exponential kernels calculates separation line in higher dimension. This is called kernel trick.

GAMMA

The gamma parameter defines how far the influence of a single training set reaches. With low gamma, points far away from the possible separation line are considered in calculation for the separation line. Where as high gamma means the points close to possible line are considered in calculation.

REGULARIZATION

For large values of this parameter, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of it will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points.

MARGIN

A margin is a separation of line to the closest class points.
A good margin is one where this separation is larger for both the classes. A good margin allows the points to be in their respective classes without crossing to other class.

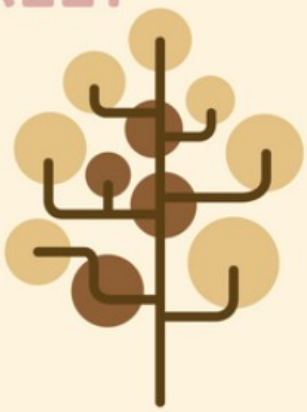
DECISION TREE

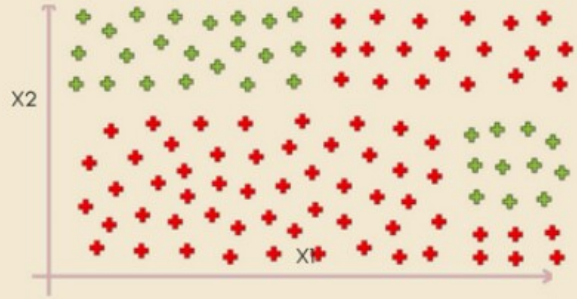
AN INTUITION ON DECISION TREE FOR CLASSIFICATION

1 WHAT IS A DECISION TREE?

It is a type of supervised learning algorithm that is mostly used in classification problems and works for both categorical and continuous input and output variables.

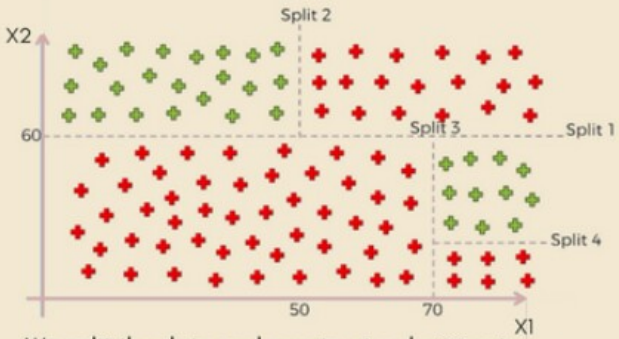
A decision tree is a tree in which each branch node represents a choice between a number of alternatives and each leaf node represents a decision.



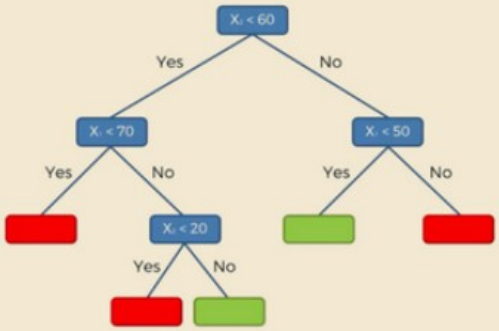


Here we've got an example with lots of points on our two dimensional scatter plot.
Now how does a decision tree work.

So what it is going to do is cut it up into slices in several iterations.



We split the data and construct a decision tree side by side which we will use later. This very task is achieved by using various algorithms. It builds a decision tree from a fixed set of examples and the resulting tree is used to classify future samples.



The resulting Tree (obtained by applying algorithms like CART, ID3) which will be later used to predict the outcomes

3 DECISION TREE ALGORITHM: ID3

ID3 stands for Iterative Dichotomizer 3. The basic idea is to construct the decision tree by employing a top-down, greedy search through the given sets to test each attribute at every tree node.

Loop:
A -> Best Attribute
Assign A as decision attribute for node.
For each value of A, create a descendant of node.

at every tree node.
Sounds simple — but which node should we select to build the correct and most precise decision tree? How would we decide that? Well, we have some measures that can help us in selecting the best choice!

Sort training examples to leaves.
If examples perfectly classified:
STOP
Else:
Iterate over leaves

INFORMATION GAIN

The best attribute is the one which gives us maximum Information Gain. Broadly speaking, it is a mathematical way to capture the amount of information we want by picking a particular attribute. But what it really speaks about us the reduction in the randomness, over the tables that we have with the set of data, based upon knowing the value of a particular attribute. Information gain is defined by:

$$Gain(S, A) = Entropy(S) - \sum_{v} \frac{|S_v|}{|S|} Entropy(S_v)$$

S = Collection of training examples
A = Particular attribute
|Sv| = Number of elements in Sv
|S| = Number of elements in S
v = All the possible values of the attribute

ENTROPY

Entropy in machine learning also carries almost the same meaning as it does in Thermodynamics. It is a measure of randomness.

$$Entropy = -\sum_{v} p(v) \log_2 p(v)$$

Where v = possible values for the attribute.

Steps :
1.compute the entropy for data-set
2.for every attribute/feature:
    1. calculate entropy for all categorical values
    2. take average information entropy for the current attribute
    3. calculate gain for the current attribute
3. pick the highest gain attribute.
4. Repeat until we get the tree we desired

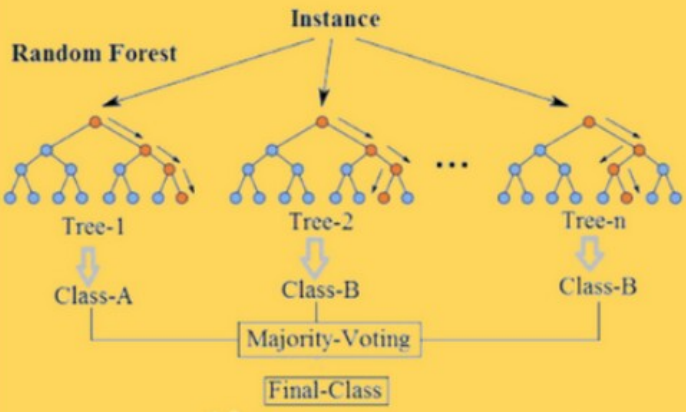
RANDOM FOREST

AN INTUITION TO RANDOM FOREST

RANDOM FORESTS ARE SUPERVISED ENSEMBLE-LEARNING MODELS USED FOR CLASSIFICATION AND REGRESSION.

Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

WHAT IS THE RANDOM FOREST ALGORITHM?



Ensemble learning models aggregate multiple machine learning models, allowing for overall better performance.

The logic behind this is that each of the models used is weak when employed on its own, but strong when put together in an ensemble. In the case of Random Forests, a large number of Decision Trees, acting as the "weak" factors, are used and their outputs are aggregated, with the result representing the "strong" ensemble.

HOW DOES IT WORK?

There are two steps in the Random Forest algorithm, one is random forest creation, the other is to make a prediction from the random forest classifier created in the first step.

THE DIFFERENCE BETWEEN THE RANDOM FOREST ALGORITHM AND THE DECISION TREE ALGORITHM IS THAT IN RANDOM FOREST, THE PROCESSES OF FINDING THE ROOT NODE AND SPLITTING THE FEATURE NODES WILL RUN RANDOMLY.

CREATION

Each tree is grown as follows:

1. If the number of cases in the training set is N, sample N cases at random - but with replacement, from the original data. This sample will be the training set for growing the tree.
2. If there are M input variables, a number is specified such that at each node, m variables are selected at random out of the M and the best split on this m is used to split the node.

PREDECTION

The random forest prediction is broken down in the below steps :

1. Takes the test features and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome (target)
2. Calculate the votes for each predicted target
3. Consider the high voted predicted target as the final prediction from the random forest algorithm

K-Means Clustering

STARTING WITH UNSUPERVISED LEARNING

1.) WHAT IS UNSUPERVISED LEARNING

Unsupervised learning allows us to approach problems with little or no idea what our results should look like. Unsupervised algorithms find patterns based only on input data. This technique is useful when we're not quite sure what to look for.

2.) CLUSTERING ALGORITHMS

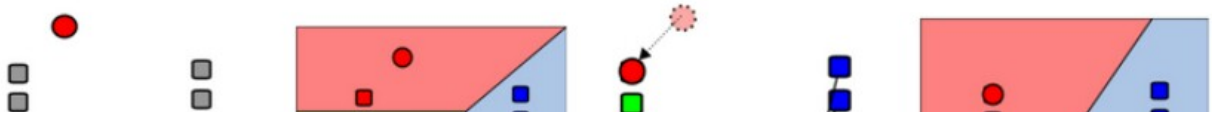
Clustering Algorithms do the task of dividing the population or data points into a variety of groups such that data points within the same cluster are similar to other data points within the same cluster than those in other groups. Basically, the aim is to separate groups with similar traits and assign them into clusters.

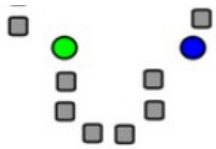
3.) K MEANS CLUSTERING

In this algorithm, we group the items into k clusters such that all items in the same cluster are as similar to each other as possible. And items not in the same cluster are as different as possible.

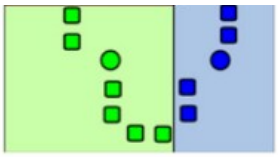
Distance measures(like Euclidean distance) are used to calculate similarity and dissimilarity between the data points. Each cluster has a centroid. Centroid can be thought as the point that is most representative of the cluster.

4.) HOW K-MEANS CLUSTERING WORKS

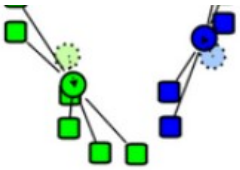




1. k initial "means" (in this case k=3) are randomly generated within the data domain.



2. k clusters are created by associating every observation with the nearest mean.



3. The centroid of each of the k clusters becomes the new mean.

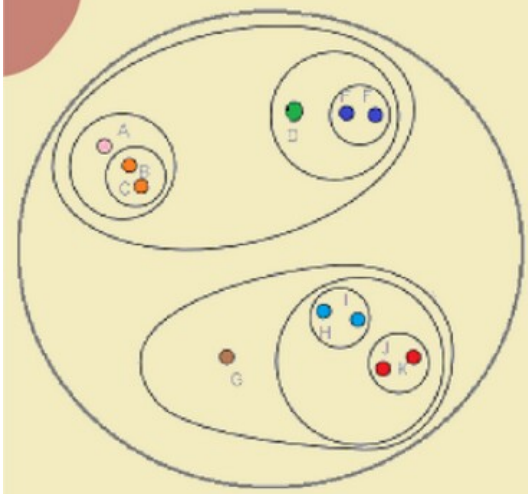
4. Steps 2 and 3 are repeated until convergence has been reached.

The objective of K-Means clustering is to minimize total intra-cluster variance, or, the squared error function:

number of clusters · number of cases · centroid for cluster j · case i

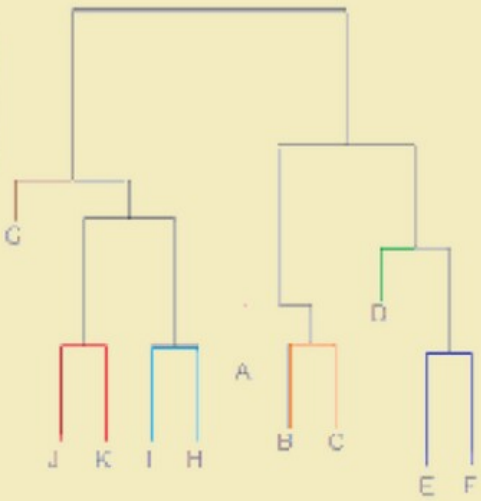
objective function ← $J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$

HIERARCHICAL CLUSTERING



HIERARCHICAL CLUSTERING

Hierarchical clustering, as the name suggests is an algorithm that builds a hierarchy of clusters. This algorithm starts with all the data points assigned to a cluster of their own. Then two nearest clusters are merged into the same cluster. In the end, this algorithm terminates when there is only a single cluster left. There are two types of hierarchical clustering, Divisive and Agglomerative.

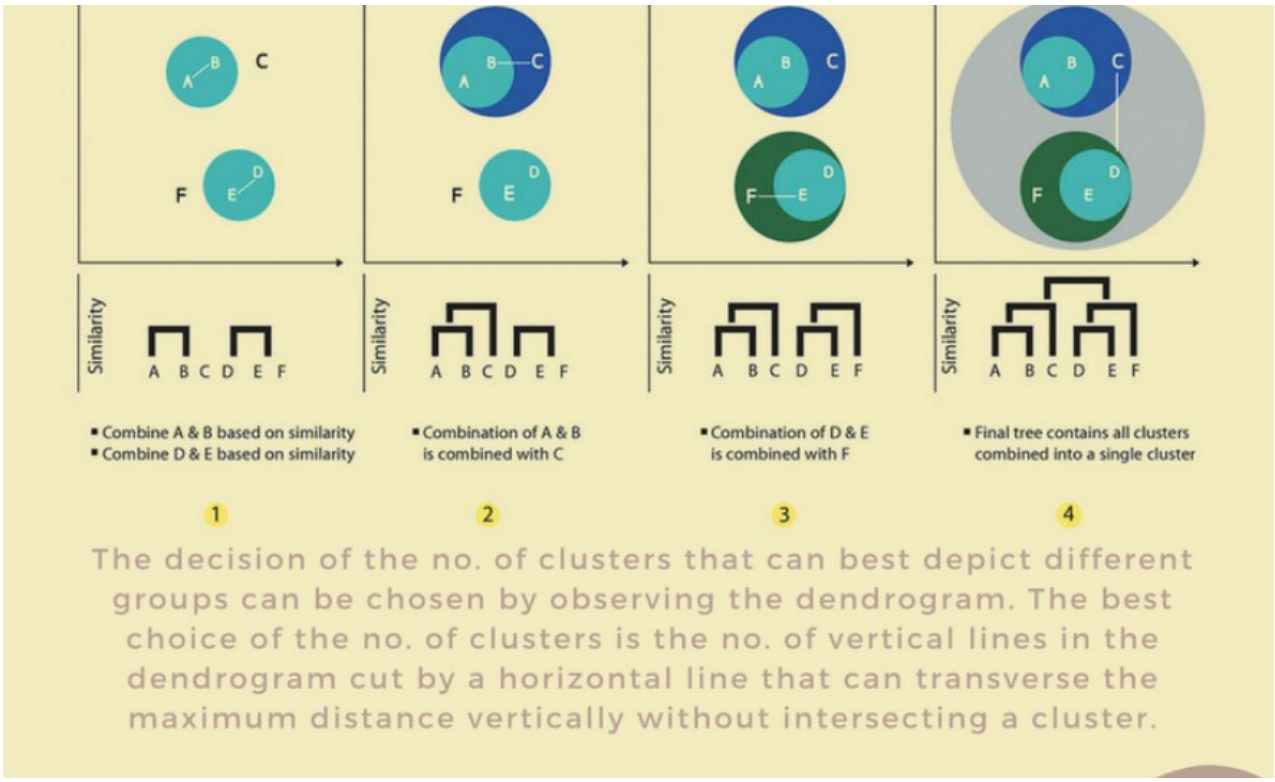


AGGLOMERATIVE HIERARCHICAL CLUSTERING

Here, each observation is initially considered as a cluster of its own (leaf). Then, the most similar clusters are successively merged until there is just one single big cluster (root). This hierarchy of clusters is represented as a tree (or dendrogram).

DENDROGRAM

The root of the tree is the unique cluster that gathers all the samples, the leaves being the clusters with only one sample.



The decision of the no. of clusters that can best depict different groups can be chosen by observing the dendrogram. The best choice of the no. of clusters is the no. of vertical lines in the dendrogram cut by a horizontal line that can transverse the maximum distance vertically without intersecting a cluster.