Pattern Recognition

Pattern is everything around in this digital world. A pattern can either be seen physically or it can be observed mathematically by applying algorithms.
Example: Mathematics: AP, GP Series. Biology: DNA, RNA. Chemistry: Elements showing a set of specific properties (S, P, D, F Blocks). The colours on the clothes, speech pattern etc.
In computer science, a pattern is represented using vector features values.

What is Pattern Recognition ?

Pattern recognition is the process of recognizing patterns by using machine learning algorithm. Pattern recognition can be defined as the classification of **data based** on **knowledge** already gained or on **statistical information extracted** from **patterns** and/or their **representation**. One of the important aspects of the pattern recognition is its application potential.

Examples: Speech recognition, speaker identification, multimedia document recognition (MDR), automatic medical diagnosis.
In a typical pattern recognition application, the raw data is processed and converted into a form that is amenable for a machine to use. Pattern recognition involves classification and cluster of patterns.

- In classification, an appropriate class label is assigned to a pattern based on an abstraction that is generated using a set of training patterns or domain knowledge. Classification is used in supervised learning.
- **Classification** uses predefined classes in which objects are assigned
- Clustering generated a partition of the data which helps decision making, the specific decision making activity of interest to us. Clustering is used in an unsupervised learning.
- **Clustering** identifies similarities **between** objects, which it groups according to those characteristics in common and which **differentiate** them from other.

Features may be represented as continuous, discrete or discrete binary variables. A feature is a function of one or more measurements, computed so that it quantifies some significant characteristics of the object.
Example: consider our face then eyes, ears, nose etc are features of the face.
A set of features that are taken together, forms the **features vector**.

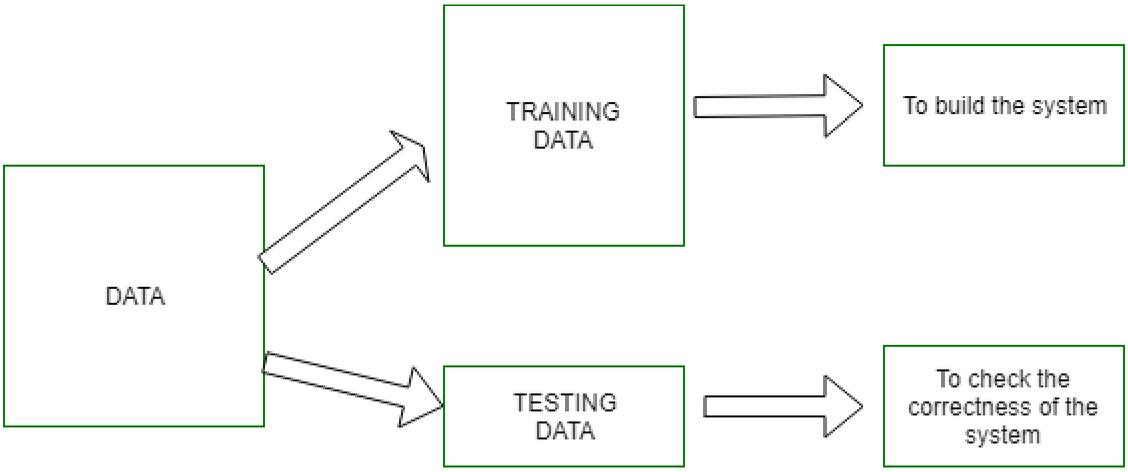
Pattern recognition possesses the following features:

- Pattern recognition system should recognise familiar pattern quickly and accurate
- Recognize and classify unfamiliar objects
- Accurately recognize shapes and objects from different angles
- Identify patterns and objects even when partly hidden
- Recognise patterns quickly with ease, and with automaticity.

Training and Learning in Pattern Recognition

Learning is a phenomena through which a system gets trained and becomes adaptable to give result in an accurate manner. Learning is the most important phase as how well the system performs on the data provided to the system depends on which algorithms used on the data. Entire dataset is divided into two categories, one which is used in training the model i.e. Training set and the other that is used in testing the model after training, i.e. Testing set.

- **Training set:**
  Training set is used to build a model. It consists of the set of images which are used to train the system. Training rules and algorithms used give relevant information on how to associate input data with output decision. The system is trained by applying these algorithms on the dataset, all the relevant information is extracted from the data and results are obtained. Generally, 80% of the data of the dataset is taken for training data.
- **Testing set:**
  Testing data is used to test the system. It is the set of data which is used to verify whether the system is producing the correct output after being trained or not. Generally, 20% of the data of the dataset is used for testing. Testing data is used to measure the accuracy of the system. Example: a system which identifies which category a particular flower belongs to, is able to identify seven category of flowers correctly out of ten and rest others wrong, then the accuracy is 70 %



A pattern is a physical object or an abstract notion. While talking about the classes of animals, a description of an animal would be a pattern. While talking about various types of balls, then a description of a ball is a pattern. In the case balls considered as pattern, the classes could be football, cricket ball, table tennis ball etc. Given a new pattern, the class of the pattern is to be determined. The choice of attributes and representation of patterns is a very important step in pattern classification. A good representation is one which makes use of discriminating attributes and also reduces the computational burden in pattern classification. An obvious representation of a pattern will be a **vector**. Each element of the vector can represent one attribute of the pattern. The first element of the vector will contain the value of the first attribute for the pattern being considered.

Advantages:

- Pattern recognition solves classification problems

- Pattern recognition solves the problem of fake bio metric detection.
- It is useful for cloth pattern recognition for visually impaired blind people.
- It helps in speaker diarization.
- We can recognise particular object from different angle.

Disadvantages:

- Syntactic Pattern recognition approach is complex to implement and it is very slow process.
- Sometime to get better accuracy, larger dataset is required.
- It cannot explain why a particular object is recognized.

Applications:

- **Image processing, segmentation and analysis**
  Pattern recognition is used to give human recognition intelligence to machine which is required in image processing.
- **Computer vision**
  Pattern recognition is used to extract meaningful features from given image/video samples and is used in computer vision for various applications like biological and biomedical imaging.
- **Seismic analysis**
  Pattern recognition approach is used for the discovery, imaging and interpretation of temporal patterns in seismic array recordings. Statistical pattern recognition is implemented and used in different types of seismic analysis models.
  (Analyzing Building structures during Earthquake).
- **Radar signal classification/analysis**
  Pattern recognition and Signal processing methods are used in various applications of radar signal classifications like AP mine detection and identification.
- **Speech recognition**
  The greatest success in speech recognition has been obtained using pattern recognition paradigms. It is used in various algorithms of speech recognition which tries to avoid the problems of using a phoneme level of description and treats larger units such as words as pattern
- **Finger print identification**
  The fingerprint recognition technique is a dominant technology in the biometric market. A number of recognition methods have been used to perform fingerprint matching out of which pattern recognition approaches is widely used.

Basics and Design Principles

Pattern Recognition System
Pattern is everything around in this digital world. A pattern can either be seen physically or it can be observed mathematically by applying algorithms.

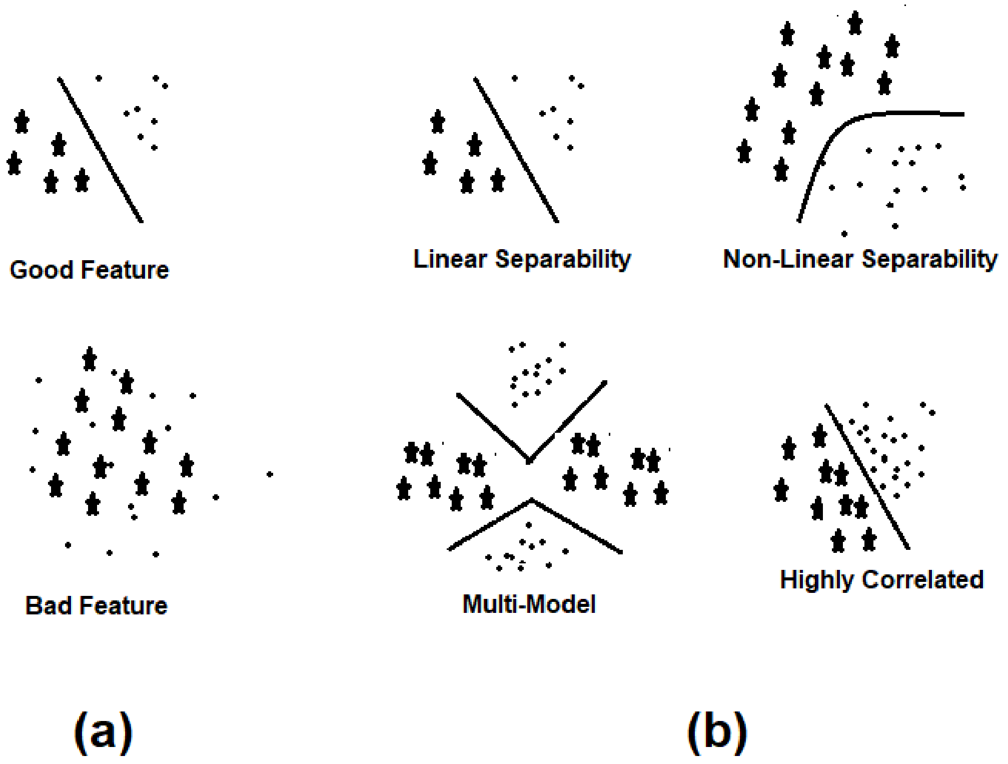
In **Pattern Recognition**, pattern is comprises of the following two fundamental things:

- Collection of observations
- The concept behind the observation

Feature Vector:

The collection of observations is also known as a feature vector. A feature is a distinctive characteristic of a good or service that sets it apart from similar items. **Feature vector** is the combination of n features in n-dimensional column vector. The different classes may have different features values but the same class always has the same features values.

Example:

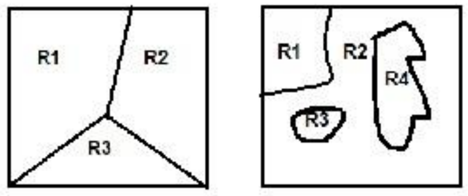


a. Differentiate between good and bad features.
b. Feature properties.

A metric space is a set where a distance(called a metric) is defined b/w elements of set.

Classifier and Decision Boundaries:

- In a statistical-classification problem, a **decision boundary** is a hypersurface that partitions the underlying vector space into two sets. A decision boundary is the region of a problem space in which the output label of a classifier is ambiguous. **Classifier** is a hypothesis or discrete-valued function that is used to assign (categorical) class labels to particular data points.
- **Classifier** is used to partition the feature space into class-labeled decision regions. While **Decision Boundaries** are the borders between decision regions.



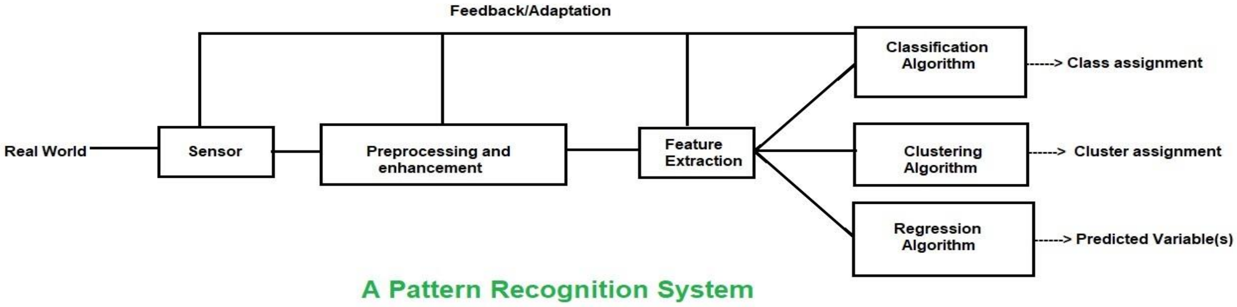
Classifier and decision boundaries

Positive and Negative Prediction

Components in Pattern Recognition System:

A pattern recognition systems can be partitioned into components. There are five typical components for various pattern recognition systems. These are as following:

- **A Sensor :** A sensor is a device used to measure a property, such as pressure, position, temperature, or acceleration, and respond with feedback.
- **A Preprocessing Mechanism :** Segmentation is used and it is the process of partitioning a data into multiple segments. It can also be defined as the technique of dividing or partitioning an data into parts called segments.
- **A Feature Extraction Mechanism :** feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. It can be manual or automated.
- **A Description Algorithm :** Pattern recognition algorithms generally aim to provide a reasonable answer for all possible inputs and to perform “most likely” matching of the inputs, taking into account their statistical variation
- **A Training Set :** Training data is a certain percentage of an overall dataset along with testing set. As a rule, the better the training data, the better the algorithm or classifier performs.



A Pattern Recognition System

Design Principles of Pattern Recognition

In pattern recognition system, for recognizing the pattern or structure two basic approaches are used which can be implemented in different techniques. These are –

- Statistical Approach and
- Structural Approach

Statistical Approach:

Statistical methods are mathematical formulas, models, and techniques that are used in the statistical analysis of raw research data. The application of statistical methods extracts information from research data and provides different ways to assess the robustness of research outputs.

Two main statistical methods are used :

1. **Descriptive Statistics:** It summarizes data from a sample using indexes such as the mean or standard deviation.
2. **Inferential Statistics:** It draw conclusions from data that are subject to random variation.

Structural Approach:

The Structural Approach is a technique wherein the learner masters the pattern of sentence. Structures are the different arrangements of words in one accepted style or the other.

Types of structures:

- Sentence Patterns
- Phrase Patterns
- Formulas
- Idioms

Difference Between Statistical Approach and Structural Approach:

Sr. No.	Statistical Approach	Structural Approach
1	Statistical decision theory.	Human perception and cognition.
2	Quantitative features.	Morphological primitives
3	Fixed number of features.	Variable number of primitives.
4	Ignores feature relationships.	Captures primitives relationships.
5	Semantics from feature position.	Semantics from primitives encoding.
6	Statistical classifiers.	Syntactic grammars.

Supervised and Unsupervised learning

Supervised learning

Supervised learning as the name indicates the presence of a supervisor as a teacher. Basically supervised learning is a learning in which we teach or train the machine using data which is well labeled that means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples(data) so that supervised learning algorithm analyses the training data(set of training examples) and produces a correct outcome from labeled data.

Supervised learning classified into two categories of algorithms:

- **Classification**: A classification problem is when the output variable is a category, such as “Red” or “blue” or “disease” and “no disease”.
- **Regression**: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

Supervised learning deals with or learns with “labeled” data. Which implies that some data is already tagged with the correct answer.

Types:-

- Regression
- Logistic Regression
- Classification
- Naïve Bayes Classifiers
- Decision Trees
- Support Vector Machine

Advantages:-

- ➢ Supervised learning allows collecting data and produce data output from the previous experiences.
- ➢ Helps to optimize performance criteria with the help of experience.
- ➢ Supervised machine learning helps to solve various types of real-world computation problems.

Disadvantages:-

- ➢ Classifying big data can be challenging.
- ➢ Training for supervised learning needs a lot of computation time. So, it requires a lot of time.

Unsupervised learning

Unsupervised learning is the training of machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. Here the task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of data.

Unlike supervised learning, no teacher is provided that means no training will be given to the machine. Therefore machine is restricted to find the hidden structure in unlabeled data by our-self.

Unsupervised learning classified into two categories of algorithms:

- **Clustering**: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.
- **Association**: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

Types of Unsupervised Learning:-

Clustering

1. Exclusive (partitioning)
2. Agglomerative
3. Overlapping
4. Probabilistic

Clustering Types:-

1. Hierarchical clustering
2. K-means clustering
3. K-NN (k nearest neighbors)
4. Principal Component Analysis
5. Singular Value Decomposition
6. Independent Component Analysis

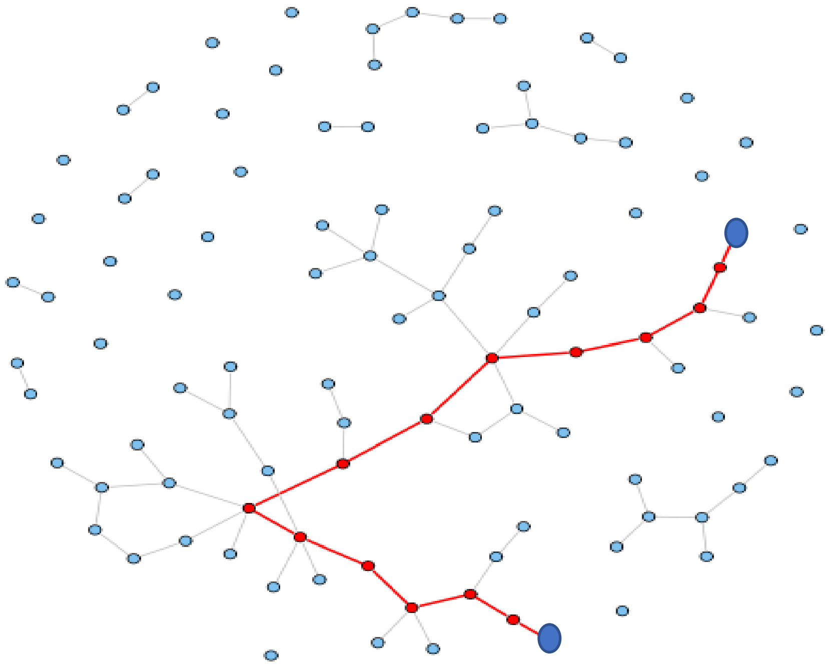
Metric Space Method

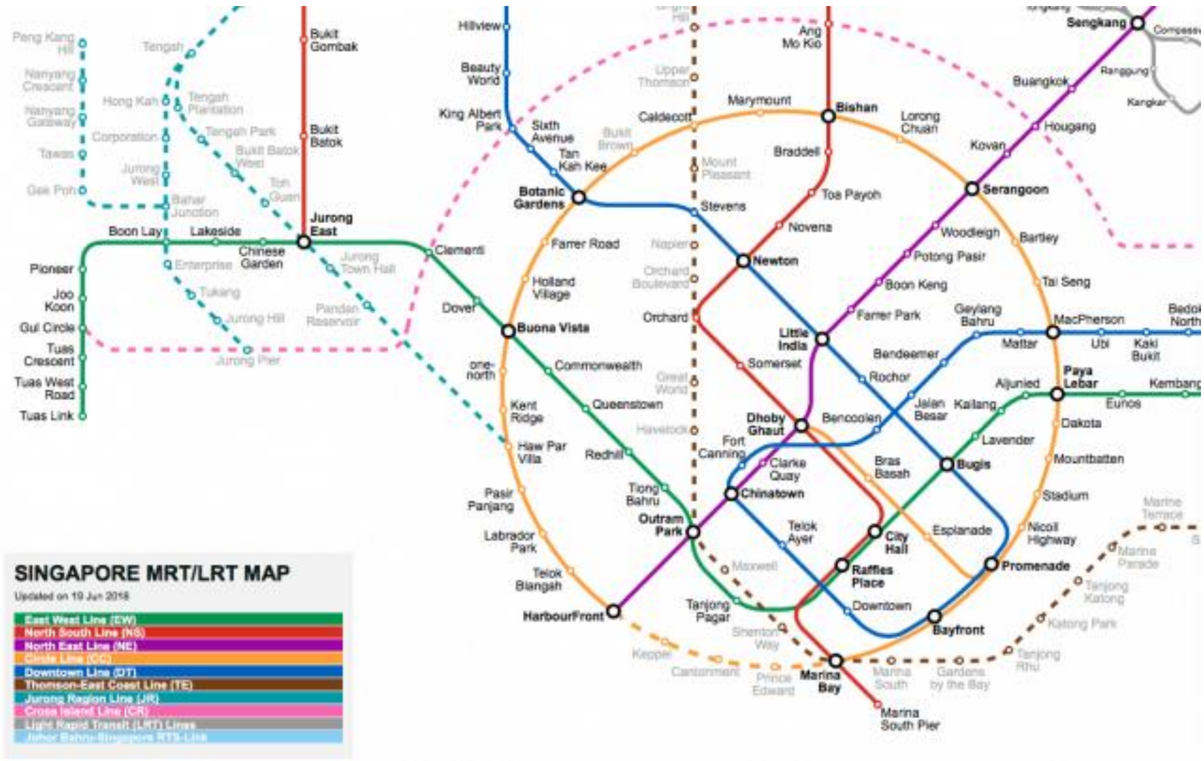
Mathematical Derivation and Problem has been already solved.

In mathematics, a metric space is a set where a distance (called a metric) is defined between elements of the set. Metric space methods have been employed for decades in various applications, for example in internet search engines, image classification etc.

What is a Distance?

The creation of a metric space requires the definition of a dissimilarity distance, which measures the dissimilarity between two different models. The distance measure has two main requirements. First, since the distance must be calculated between each model pair, it should be rapid to calculate for large ensembles of models. Second, the distance measure must be designed for the purpose of the study to be undertaken. No single distance measure is applicable to all situations. Finally, the distance measure must be easy to understand, in order to understand the results of the study.




SINGAPORE MRT/LRT MAP
Updated on 19 Jun 2018
East West Line (EW)
North South Line (NS)
North East Line (NE)
Circle Line (CC)
Downtown Line (DT)
Thomson-East Coast Line (TE)
Jurong Region Line (JR)
Cross Island Line (CR)
Light Rapid Transit (LRT) Lines
Johor Bahru-Singapore RTS Link
Peng Kang Hill
Nanyang Crescent
Nanyang Gateway
Tawas
Gek Poh
Tengah
Hong Kah
Tengah Plantation
Corporation
Tengah Park
Bukit Batok West
Jurong West
Toh Guan
Bahar Junction
Bukit Gombak
Bukit Batok
Jurong East
Boon Lay
Lakeside
Chinese Garden
Enterprise
Tukang
Jurong Hill
Jurong Pier
Jurong Town Hall
Pandan Reservoir
Pioneer
Joo Koon
Gul Circle
Tuas Crescent
Tuas West Road
Tuas Link
Hillview
Beauty World
King Albert Park
Sixth Avenue
Tan Kah Kee
Bukit Brown
Botanic Gardens
Clementi
Farrer Road
Holland Village
Dover
Buona Vista
one-north
Commonwealth
Kent Ridge
Queenstown
Haw Par Villa
Redhill
Pasir Panjang
Tiong Bahru
Labrador Park
Outram Park
Telok Blangah
HarbourFront
Keppel
Cantonment
Prince Edward
Upper Thomson
Caldecott
Mount Pleasant
Stevens
Napier
Orchard Boulevard
Orchard
Great World
Havelock
Maxwell
Shenton Way
Marymount
Ang Mo Kio
Bishan
Braddell
Toa Payoh
Novena
Newton
Little India
Somerset
Dhoby Ghaut
Fort Canning
Clarke Quay
Chinatown
Tanjong Pagar
Telok Ayer
Bencoolen
Bras Basah
City Hall
Raffles Place
Downtown
Marina Bay
Marina South
Marina South Pier
Gardens by the Bay
Esplanade
Bayfront
Promenade
Lorong Chuan
Serangoon
Woodleigh
Potong Pasir
Boon Keng
Farrer Park
Bendeemer
Rochor
Jalan Besar
Bugis
Kallang
Lavender
Aljunied
Geylang Bahru
Mattar
MacPherson
Ubi
Kaki Bukit
Bedok North
Paya Lebar
Eunos
Kembangan
Dakota
Mountbatten
Stadium
Nicoll Highway
Tai Seng
Bartley
Kovan
Hougang
Buangkok
Sengkang
Compassvale
Rangpung
Kangkar
Marine Terrace
Marine Parade
Tanjong Katong
Katong Park
Tanjong Rhu

UNIT: 2

What is Classification?

Classification is technique to categorize our data into a desired and distinct number of classes where we can assign label to each class.

In classification tasks, your job is to build a function that takes in a vector of **features X** (also called "inputs") and predicts a **label** Y (also called the "class" or "output"). Features are things you know, and the label is what your algorithm is trying to figure out; for example, the label might be a binary variable indicating whether an animal is a cat or a dog, and the features might be the length of the animal's whiskers, the animal's weight in pounds, and a binary variable indicating whether the animal's ears stick up or are droopy. Your algorithm needs to tell dogs and cats apart (Y) using only this information about weight, whiskers, and ears (**X**).

Applications of Classification are: speech recognition, handwriting recognition, biometric identification, document classification etc.

Classifiers can be:

Binary classifiers: Classification with only 2 distinct classes or with 2 possible outcomes

example: Male and Female

example: classification of spam email and non spam email

example: classification of author of book

example: positive and negative sentiment

Multi-Class classifiers: Classification with more than two distinct classes.

example: classification of types of soil

example: classification of types of crops

example: classification of mood/feelings in songs/music

1). Naive Bayes (Classifier):

Naive Bayes is a probabilistic classifier inspired by the Bayes theorem. Under a simple assumption which is the attributes are conditionally independent.

$$\underset{\text{Posterior Probability}}{P(c\mid x)} = \frac{\overset{\text{Likelihood}}{P(x\mid c)}\,\overset{\text{Class Prior Probability}}{P(c)}}{\underset{\text{Predictor Prior Probability}}{P(x)}}$$

$$P(c\mid \mathrm{X}) = P(x_1\mid c)\times P(x_2\mid c)\times\cdots\times P(x_n\mid c)\times P(c)$$

Fig: Naïve Bayes

The classification is conducted by deriving the maximum posterior which is the maximal P(Ci|**X**) with the above assumption applying to Bayes theorem. This assumption greatly reduces the computational cost by only counting the class distribution. Even though the assumption is not valid in most cases since the attributes are dependent, surprisingly Naive Bayes has able to perform impressively.

Naive Bayes is a very simple algorithm to implement and good results have obtained in most cases. It can be easily scalable to larger datasets since it takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

Naive Bayes can suffer from a problem called the zero probability problem. When the conditional probability is zero for a particular attribute, it fails to give a valid prediction. This needs to be fixed explicitly using a Laplacian estimator.

Advantages: This algorithm requires a small amount of training data to estimate the necessary parameters. Naive Bayes classifiers are extremely fast compared to more sophisticated methods.

Disadvantages: Naive Bayes is is known to be a bad estimator.

Steps for Implementation:

- Initialise the classifier to be used.
- Train the classifier: All classifiers in scikit-learn uses a fit(X, y) method to fit the model(training) for the given train data X and train label y.
- Predict the target: Given an non-label observation X, the predict(X) returns the predicted label y.
- Evaluate)* the classifier model

$$\text{Bayes Theorem: } P(A/B) = \frac{P(B/A)*P(A)}{P(B)}$$

Dataset: x

$x = \{x_1, x_2, x_3 \ldots\ldots\ldots\ldots\ldots x_n\} = \{y\}$

F_1	F_2	$F_3 \ldots\ldots\ldots$	Y
x_1	x_2	x_3	y_1

$$P(y/x_1,x_2,x_3,\ldots\ldots..x_n) = \frac{P(x_1/y)* P(x_2/y)*P(x_3/y)*\ldots\ldots\ldots.\ P(x_n/y) \quad * \quad P(y)}{P(x_1)* P(x_2)*P(x_3)*\ldots\ldots\ldots.\ P(x_n)}$$

$$P(y/x_1,x_2,x_3,\ldots\ldots..x_n) = \frac{P(y) \ * \ \pi_{i=1}^{n} P(x_i/y)}{P(x_1)* P(x_2)*P(x_3)*\ldots\ldots\ldots.\ P(x_n)}$$

$$P(x_1)* P(x_2)*P(x_3)*\ldots\ldots\ldots.\ P(x_n) \propto P(y) \ * \ \pi_{i=1}^{n} P(x_i/y)$$

$$y = \text{argmax } P(y) \ \pi_{i=1}^{n} P(x_i/y)$$

for yes = 0.7

no = 0.3

Example:
Finding probability that the player can play outside or not, Depending upon the weather.

Outlook:

	Yes	No	P(Y)	P(N)
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0/5
Rainy	3	2	1/3	2/5
Total	9	5	100%	100%

Temperature:

	Yes	No	P(Y)	P(N)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cold	3	1	1/3	1/5
Total	9	5	100%	100%

Using the above data we need to find that a player can play outside today or not?
Weather for today is sunny and hot.

		P(Y)
		/
		P(N)

Yes	9	9/14
No	5	5/14
Total	14	100%

$$P(Y/Today) = \frac{P(OC/Yes) * P(Cold/Yes) * P(Yes)}{P(Today)}$$

$$P(Y/Today) \propto P(OC/Yes) * P(Cold/Yes) * P(Yes)$$
$$= 4/9 * 1/3 * 9/14 = 0.0987$$

$$P(N/Today) \propto P(OC/No) * P(Cold/No) * P(No)$$
$$= 0/5 * 1/5 * 5/14 = 0$$

Normalization

$$P(Yes) = \frac{0.0987}{0.0987 + 0} = 1$$

$$P(No) = 1 - P(Yes) = 0$$

Random Forest Classifier:

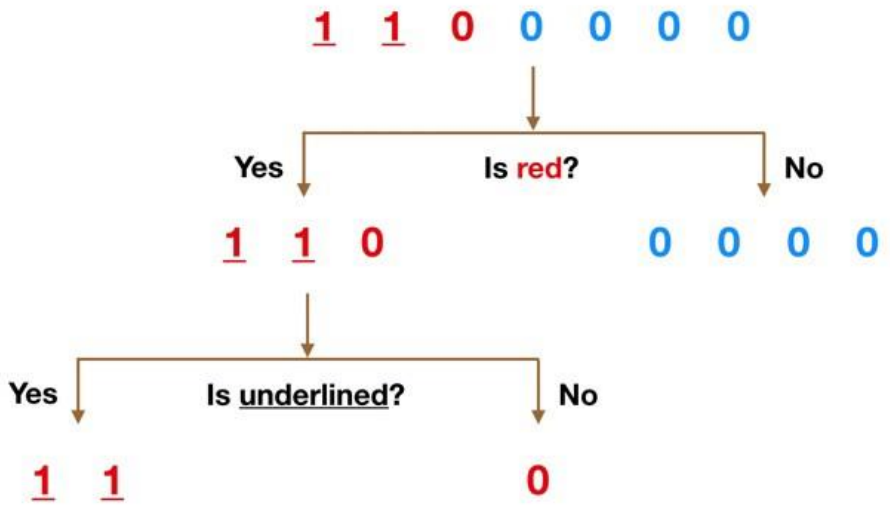
It is an ensemble tree-based learning algorithm. The Random Forest Classifier is a set of decision trees from randomly selected subset of training set. It **aggregates the votes from different decision trees** to decide the final class of the test object.

Ensemble Algorithm :

Ensemble algorithms are those which **combines more than one algorithms of same or different kind for classifying objects**. For example, running prediction over Naive Bayes, SVM and Decision Tree and then taking vote for final consideration of class for test object.

Decision Trees

Decision trees are the building blocks of the random forest model. It's probably much easier to understand how a decision tree works through an example.



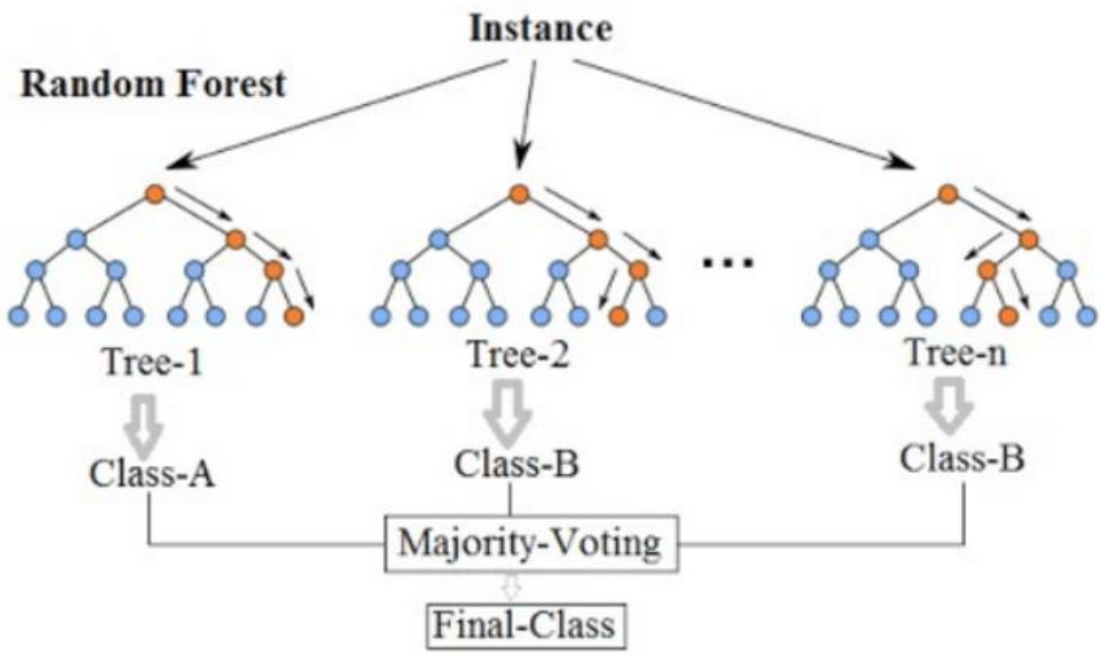
Simple Decision Tree Example

Imagine that our dataset consists of the numbers at the top of the figure to the left. We have two 1s and five 0s (1s and 0s are our classes) and desire to separate the classes using their features. The features are color (red vs. blue) and whether the observation is underlined or not. So how can we do this?

Color seems like a pretty obvious feature to split by as all but one of the 0s are blue. So we can use the question, "Is it red?" to split our first node. You can think of a node in a tree as the point where the path splits into two — observations that meet the criteria go down the Yes branch and ones that don't go down the No branch.

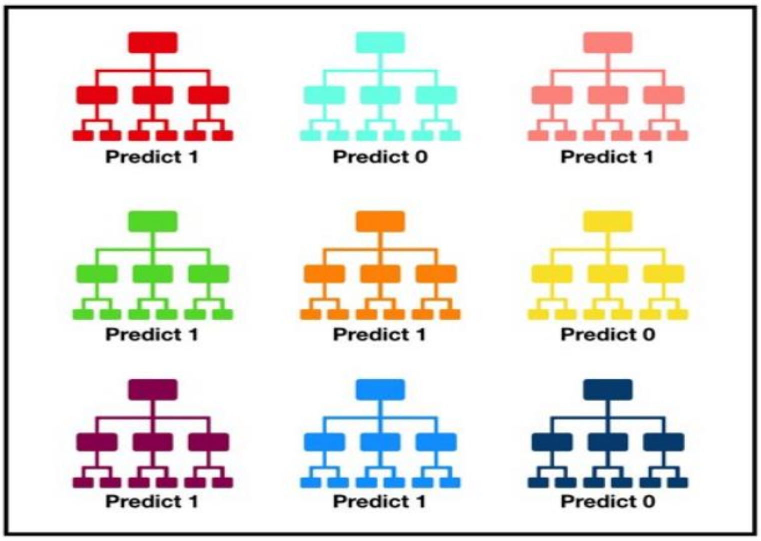
The No branch (the blues) is all 0s now so we are done there, but our Yes branch can still be split further. Now we can use the second feature and ask, "Is it underlined?" to make a second split.

The two 1s that are underlined go down the Yes subbranch and the 0 that is not underlined goes down the right subbranch and we are all done. Our decision tree was able to use the two features to split up the data perfectly.



Structure of Random Forest Classification

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction



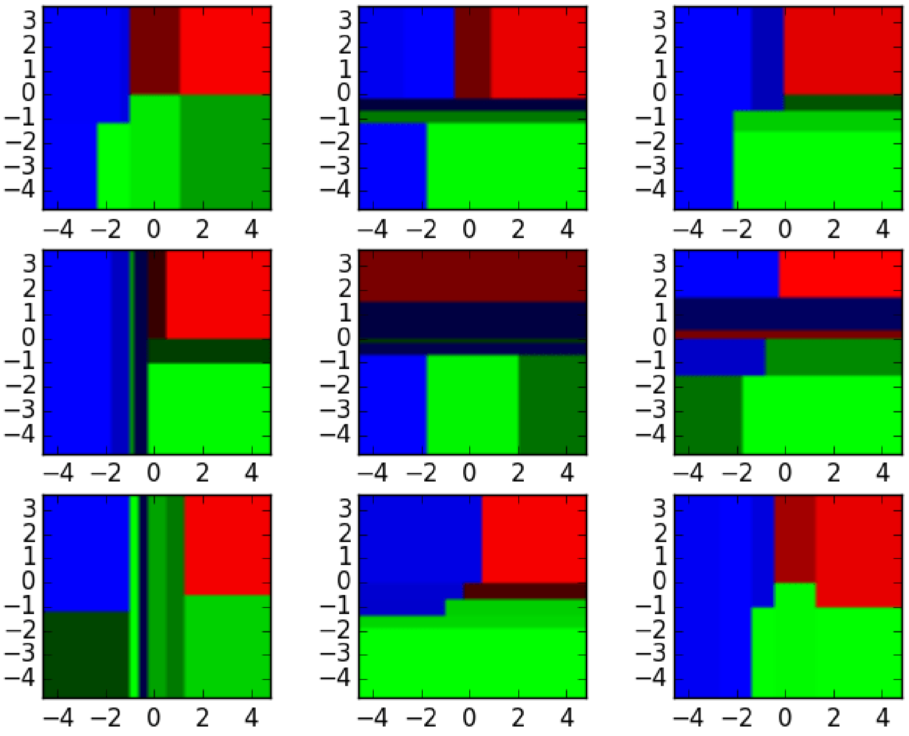
Types of Random Forest models:

1. Random Forest Prediction for a **classification problem**:

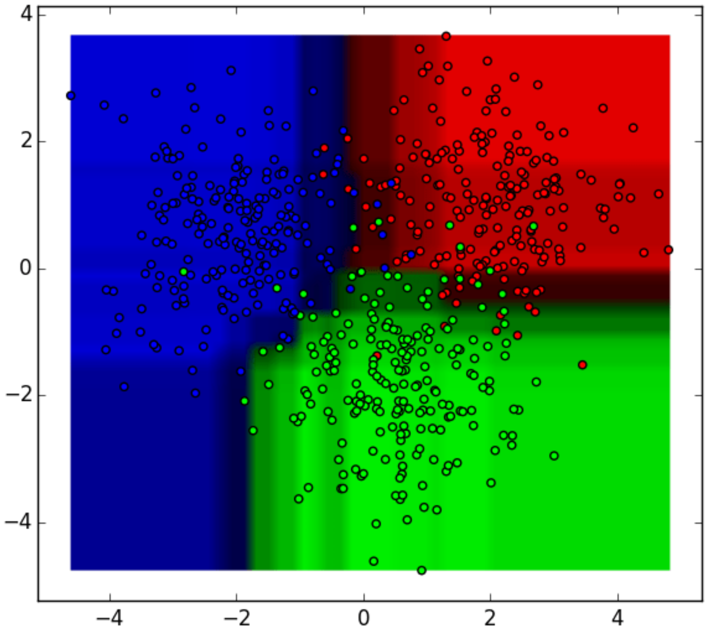
f(x) = majority vote of all predicted classes over B trees

2. Random Forest Prediction for a **regression problem**:

f(x) = sum of all sub-tree predictions divided over B trees



Nine Different Decision Tree Classifier



Aggregated Result based on above classifier

The 9 decision tree classifiers shown above can be aggregated into a random forest ensemble which **combines their input**. The horizontal and vertical axes of the above decision tree outputs can be thought of as features x1 and x2. At certain values of each feature, the decision tree outputs a classification of “blue”, “green”, “red”, etc.

These above **results are aggregated**, through **model votes or averaging**, into a single ensemble model that ends up outperforming any individual decision tree’s output.

Features and Advantages of Random Forest :

1. It is one of the most accurate learning algorithms available. For many data sets, it produces a **highly accurate classifier**.
2. It runs efficiently on large databases.
3. It can **handle thousands of input variables** without variable deletion.
4. It gives estimates of what variables that are important in the classification.
5. It generates an internal **unbiased estimate of the generalization error** as the forest building progresses.
6. It has an **effective method for estimating missing data** and maintains accuracy when a large proportion of the data are missing.

Disadvantages of Random Forest:

1. Random forests have been observed to **overfit for some datasets** with noisy classification/regression tasks.
2. For data including categorical variables with different number of levels, **random forests are biased in favor of those attributes with more levels**. Therefore, the variable importance scores from random forest are not reliable for this type of data.

Nearest Neighbour

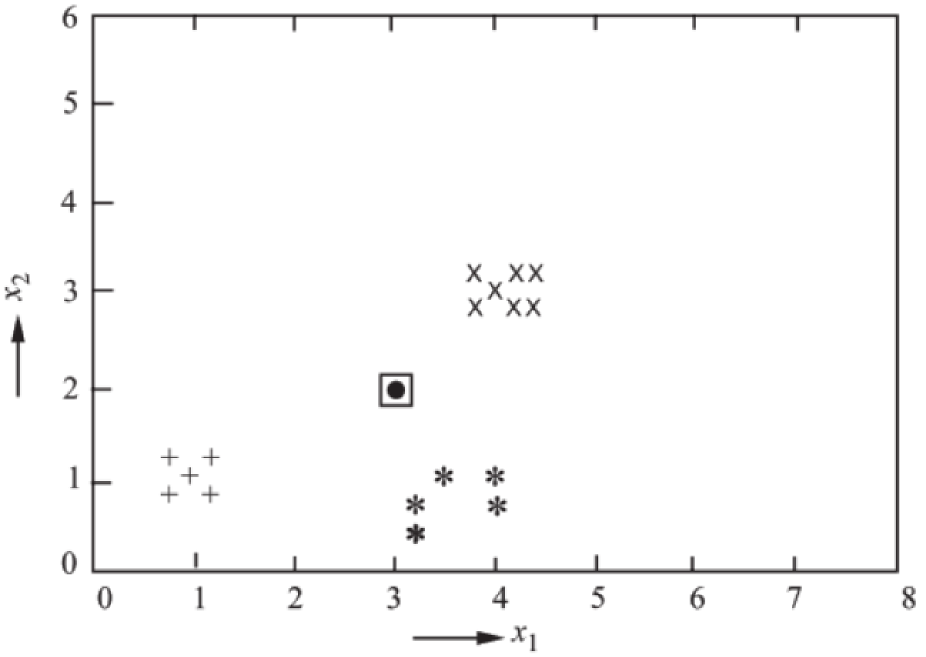
One of the simplest decision procedures that can be used for classification is the nearest neighbour (NN) rule. It classifies a sample based on the category of its nearest neighbour. When large samples are involved, it can be shown that this rule has a probability of error which is less than twice the optimum error—hence there is less than twice the probability of error compared to any other decision rule. The nearest neighbour based classifiers use some or all the patterns available in the training set to classify a test pattern. These classifiers essentially involve finding the similarity between the test pattern and every pattern in the training set.

Nearest Neighbour Algorithm

The nearest neighbour algorithm assigns to a test pattern the class label of its closest neighbour. Let there be n training patterns, (X_1,θ_1), (X_2,θ_2), ..., (X_n, θ_n), where Xi is of dimension d and θi is the class label of the ith pattern. If P is the test pattern, then if
$d(P, X_k) = \min\{ d(P, X_i)\}$
where i = 1...n.
Pattern P is assigned to the class θ_k associated with X_k.

Example

Let the training set consist of the following three dimensional patterns:
X1 = (0.8, 0.8, 1), X2 = (1.0, 1.0, 1), X3 = (1.2, 0.8, 1)
X4 = (0.8, 1.2, 1), X5 = (1.2, 1.2, 1), X6 = (4.0, 3.0, 2)
X7 = (3.8, 2.8, 2), X8 = (4.2, 2.8, 2), X9 = (3.8, 3.2, 2)
X10 = (4.2, 3.2, 2), X11 = (4.4, 2.8, 2), X12 = (4.4, 3.2, 2)
X13 = (3.2, 0.4, 3), X14 = (3.2, 0.7, 3), X15 = (3.8, 0.5, 3)
X16 = (3.5, 1.0, 3), X17 = (4.0, 1.0, 3), X18 = (4.0, 0.7, 3)



Data Set

For each pattern, the first two numbers in the triplets gives the first and second features, and the third number gives the class label of the pattern. This can be seen plotted in figure above. Here ‘‘+’’ corresponds to Class 1, ‘‘X’’ corresponds to Class 2 and ‘‘*’’ corresponds to Class 3. Now if there is a test pattern P = (3.0, 2.0), it is necessary to find the distance from P to all the training patterns.
Let the distance between X and P be the Euclidean distance

$$d(\mathrm{X}, \mathrm{P}) = \sqrt{(X[1] - P[1])^2 + (X[2] - P[2])^2}$$

The distance from a point P to every point in the set can be computed using the above formula. For P = (3.0, 2.0), the distance to X1 is

$$d(\mathrm{X}_1, \mathrm{P}) = \sqrt{(0.8 - 3.0)^2 + (0.8 - 2.0)^2} = 2.51$$

We find, after calculating the distance from all the training points to P, that the closest neighbour of P is X16, which has a distance of 1.12 from P and belongs to Class 3.
Hence P is classified as belonging to Class 3.

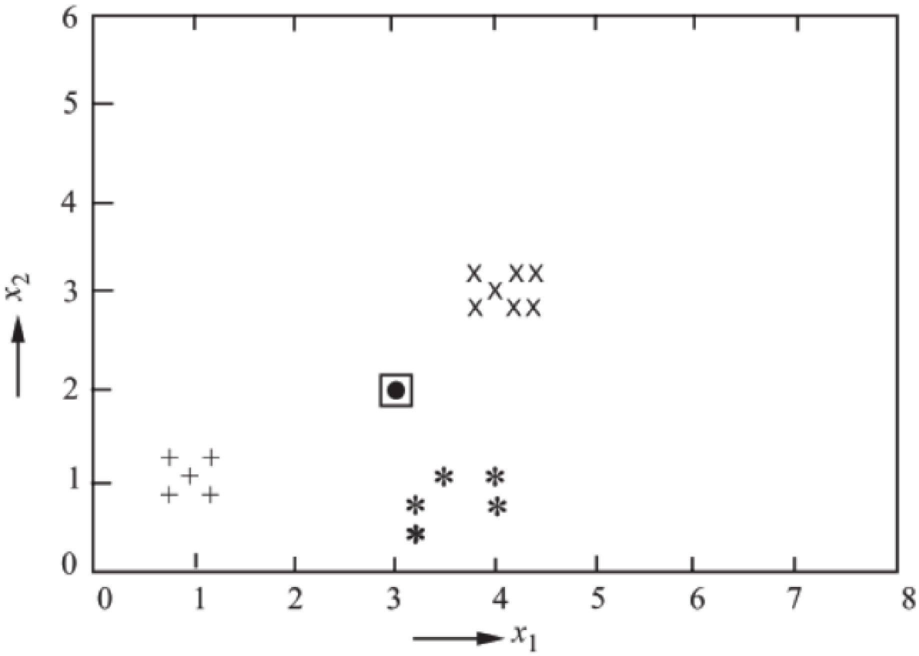
k-Nearest Neighbour (kNN) Algorithm

In this algorithm, instead of finding just one nearest neighbour as in the NN algorithm, k neighbours are found. The majority class of these k nearest neighbours is the class label assigned to the new pattern. The value chosen for k is crucial. With the right value of k, the classification accuracy will be better than that got by using the nearest neighbour algorithm.

"In KNN, 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. Data scientists usually choose as an odd number if the number of classes is 2 and another simple approach to select k is set k=sqrt(n)."

The optimal K value usually found is the square root of N, where N is the total number of samples.

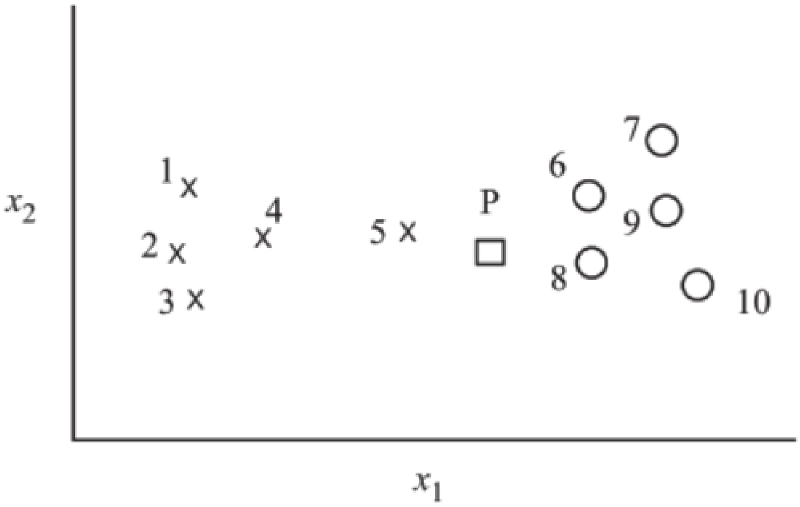
Example



Data Set

In the example shown in Figure, if k is taken to be 5, the five nearest neighbours of P are X16, X7, X14, X6 and X17. The majority class of these five patterns is class 3. This method will reduce the error in classification when training patterns are noisy. The closest pattern of the test pattern may belong to another class, but when a number of neighbours are obtained and the majority class label is considered, the pattern is more likely to be classified correctly.

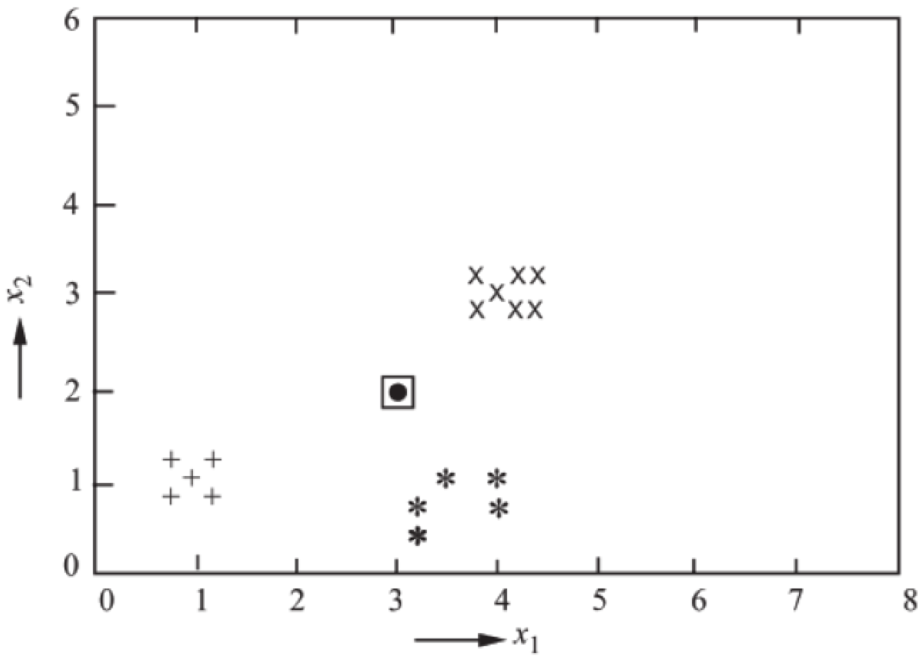
Example



P can be correctly classified using the kNN algorithm

It can be seen from the above Figure that the test point P is closest to point 5 which is an outlier in Class 1 (represented as a cross). If kNN algorithm is used, the point P will be classified as belonging to Class 2 represented by circles. Choosing k is crucial to the working of this algorithm. For large data sets, k can be larger to reduce the error. The value of k can be determined by experimentation, where a number of patterns taken out from the training set(validation set) can be classified using the remaining training patterns for different values of k. It can be chosen as the value which gives the least error in classification.

Example



In above Figure, if P is the pattern (4.2, 1.8), its nearest neighbour is X17 and it would be classified as belonging to Class 3 if the nearest neighbour algorithm is used. If the 5 nearest neighbours are taken, it can be seen that they are X17 and X16, both belonging to Class 3 and X8, X7 and X11, belonging to Class 2. Following the majority class rule, the pattern would be classified as belonging to Class 2.

Modified k-Nearest Neighbour (MkNN) Algorithm

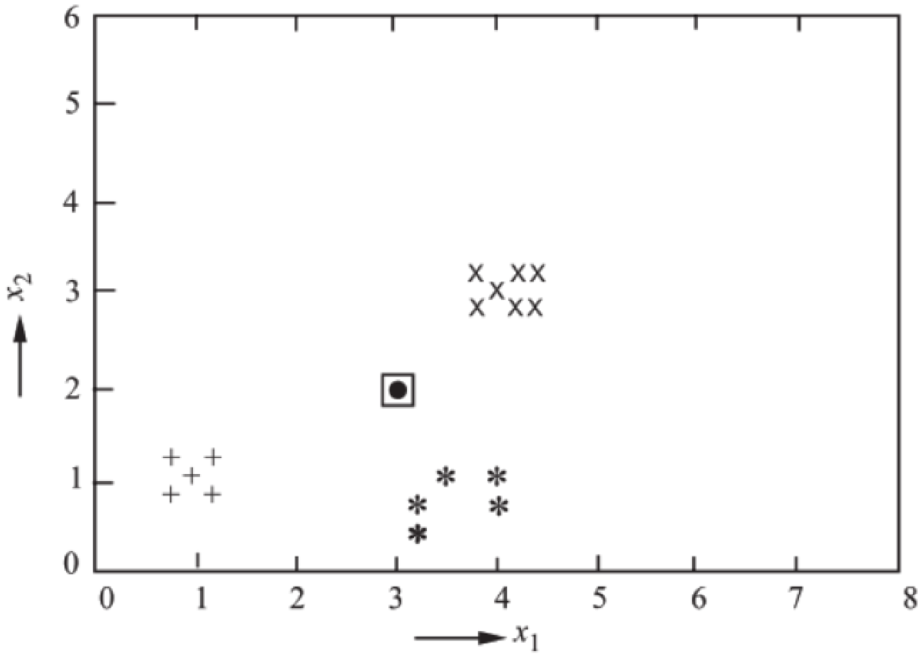
This algorithm is similar to the kNN algorithm, inasmuch as it takes the k nearest neighbours into consideration. The only difference is that these k nearest neighbours are weighted according to their distance from the test point. It is also called the distance-weighted k-nearest neighbour algorithm. Each of the neighbours is associated with the weight w which is defined as:

$$w_j = \begin{cases} \frac{d_k - d_j}{d_k - d_1} & \text{if } d_k \neq d_1 \\ 1 & \text{if } d_k = d_1 \end{cases}$$

where j = 1, .., k. Thevalueofwj varies from a maximum of 1 for the nearest neighbour down to a minimum of zero for the most distant. Having computed the weights wj, the MkNN algorithm assigns the test pattern P to that class for which the weights of the representatives among the k nearest neighbours sums to the greatest value.

Instead of using the simple majority rule, it can be observed that MkNN employs a weighted majority rule. This would mean that outlier patterns have lesser effect on classification.

Example



Consider P = (3.0, 2.0) in Figure. For the five nearest points, the distances from P are:
d(P, X_{16}) =1.2; d(P, X_7)=1.13; d(P, X_{14})=1.32; d(P, X_6) =1.41; d(P, X_{17}) =1.41;

The values of w will be

$$w_{16} = 1$$

$$w_7 = \frac{(1.41 - 1.13)}{(1.41 - 1.12)} = 0.97$$

$$w_{14} = \frac{(1.41 - 1.32)}{(1.41 - 1.12)} = 0.31$$

$$w_6 = 0$$

$$w_{17} = 0$$

Summing up for each class, Class 1 sums to 0, Class 2 to which X7 and X6 belong sums to 0.97 and Class 3 to which X16,X14 and X17 belong sums to 1.31. Therefore, the point P belongs to Class 3.
It is possible that kNN and MkNN algorithms assign the same pattern a different class label. This can be displayed with the help of next example.

Example 6
In above Figure, when P = (4.2, 1.8), the five nearest patterns are X_{17}, X_8, X_{11}, X_{16} and X_7. The distances from P to these patterns are:
d(P, X17)=0.83; d(P, X8) ==1.0; d(P, X11)=1.02
d(P, X16)=1.06; d(P, X7) ==1.08
The value of w will be:

$$w_{17} = 1$$

$$w_8 = \frac{(1.08 - 1.0)}{(1.08 - 0.83)} = 0.32$$

$$w_{11} = \frac{(1.08 - 1.02)}{(1.08 - 0.83)} = 0.24$$

$$w_{16} = \frac{(1.08 - 1.06)}{(1.08 - 0.83)} = 0.08$$

$$w_7 = 0$$

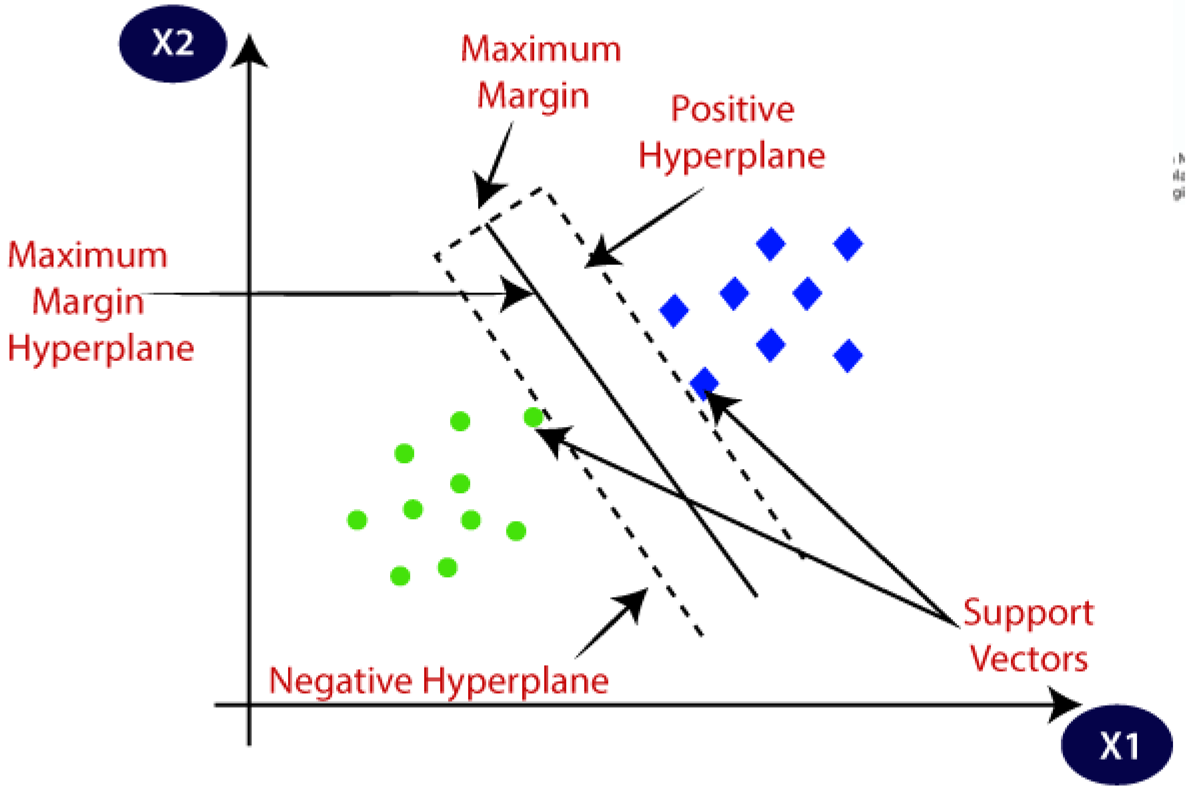
Summing up for each class, Class 1 sums to 0, Class 2 to which X_8, X_{11} and X_7 belong sums to 0.56 and Class 3 to which X_{17} and X_{16} belong sums to 1.08 and therefore, P is classified as belonging to Class 3. Note that the same pattern is classified as belonging to Class 2 when we used the k nearest neighbour algorithm with k =5.

SUPPORT VECTOR MACHINES

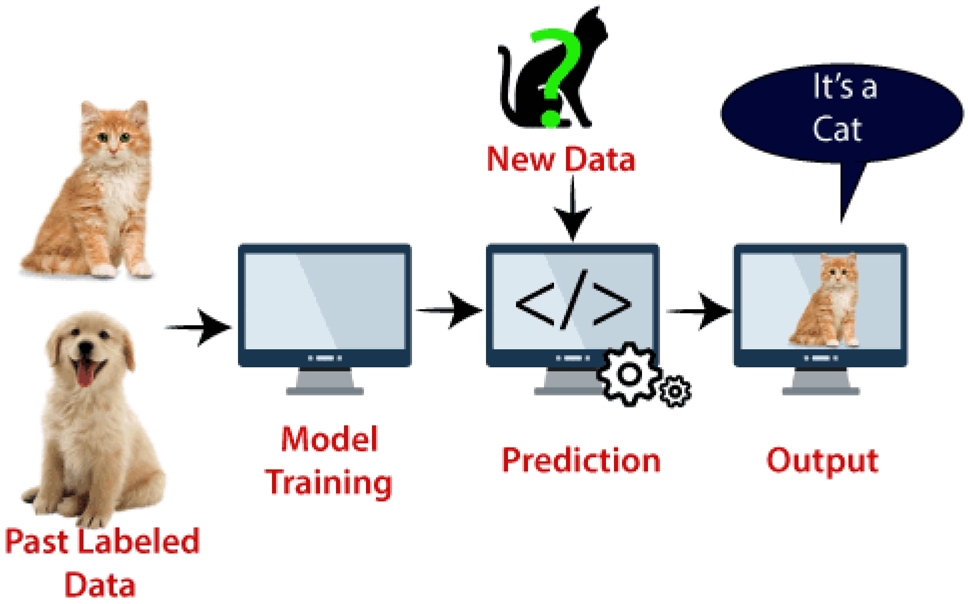
Support Vector Machine or SVM is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



SVM can be understood with the example that we have used in the KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat.



SVM algorithm can be used for **Face detection, image classification, text categorization,** etc.

SVM can be of two types:

- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane and Support Vectors in the SVM algorithm:

Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features, then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane. We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

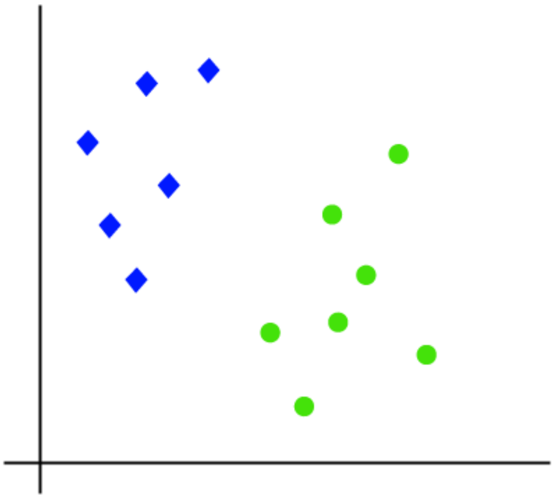
Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.

Support Vectors:

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

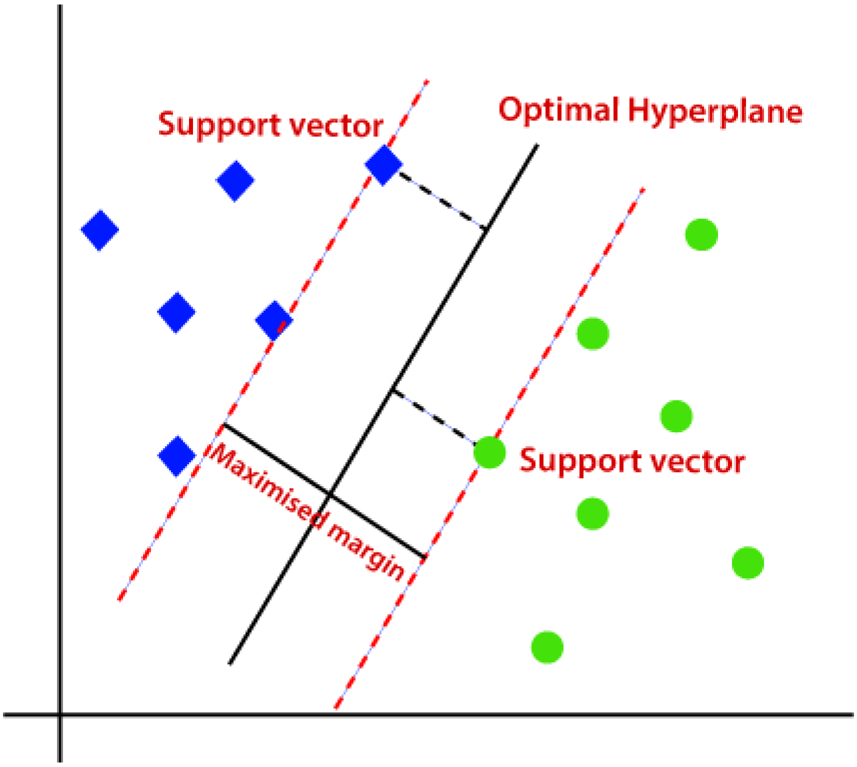
Linear SVM:

The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue.



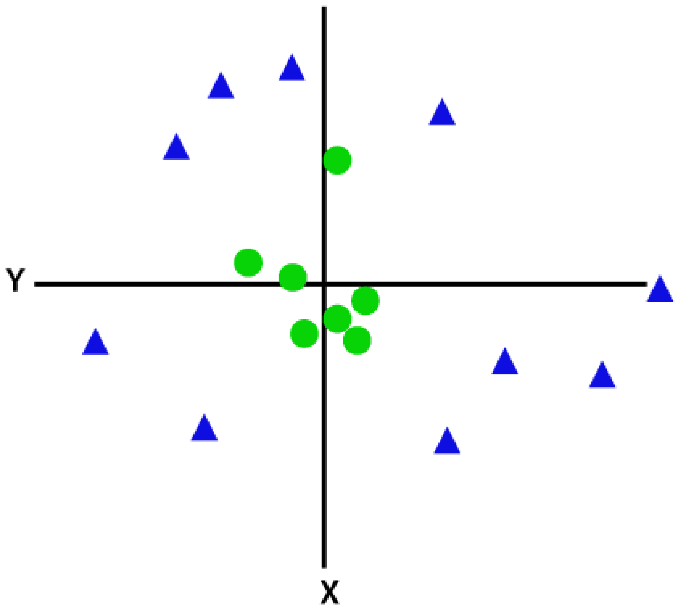
So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes.

Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as **margin**. And the goal of SVM is to maximize this margin. The **hyperplane** with maximum margin is called the **optimal hyperplane**.



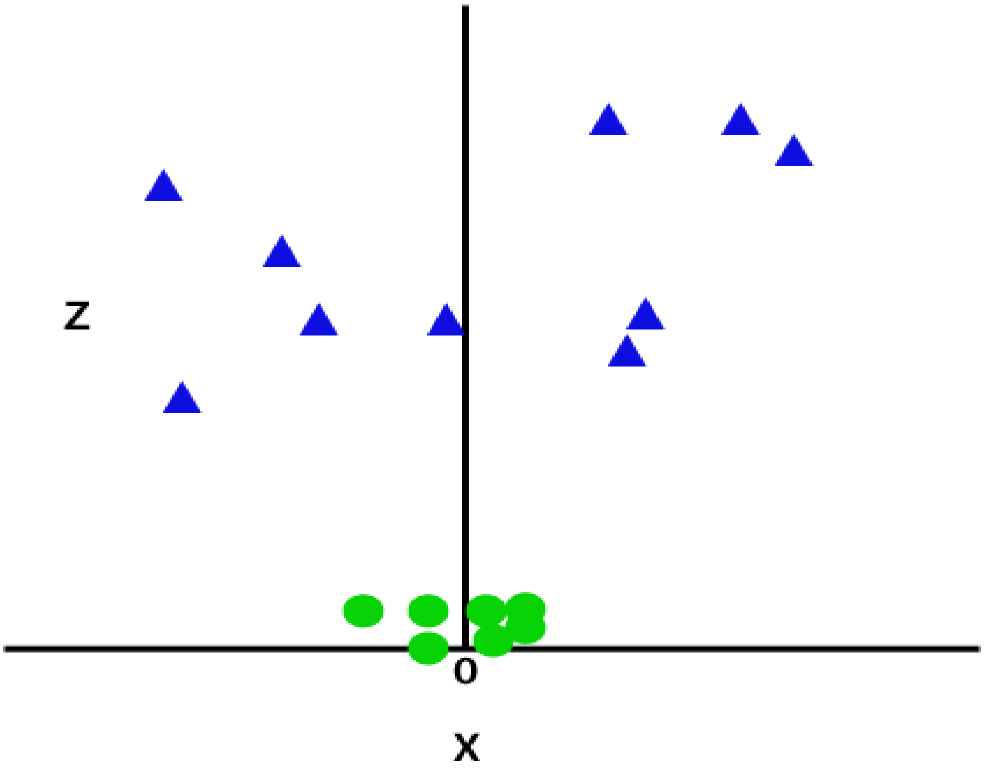
Non-Linear SVM:

If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line.

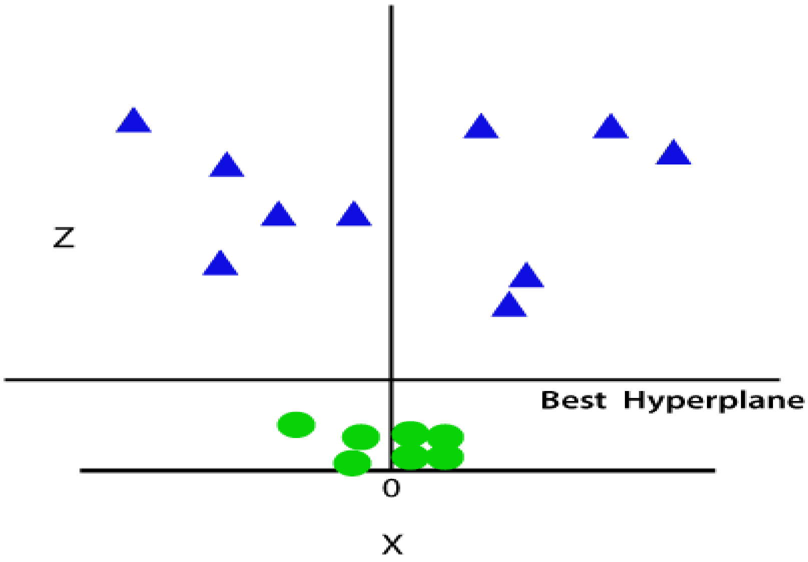


So to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third dimension z. It can be calculated as: $z = x^2 + y^2$

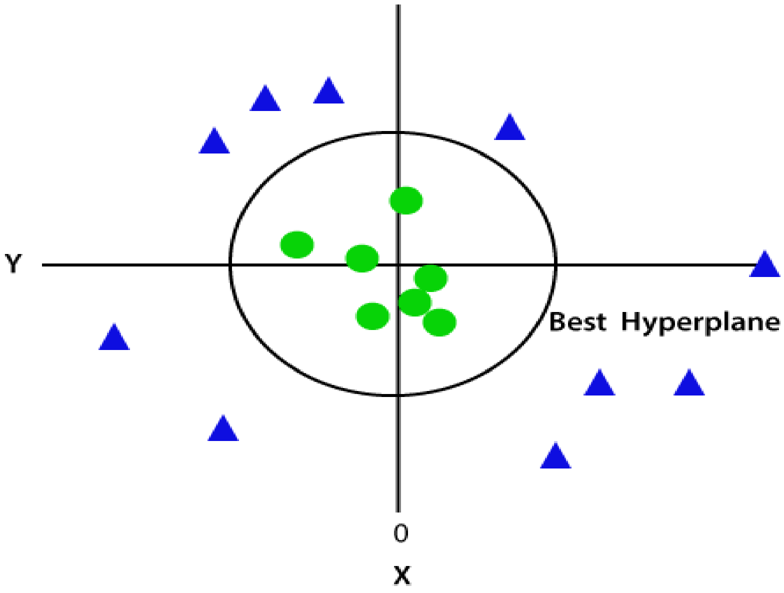
By adding the third dimension, the sample space will become as below image:



So now, SVM will divide the datasets into classes in the following way.



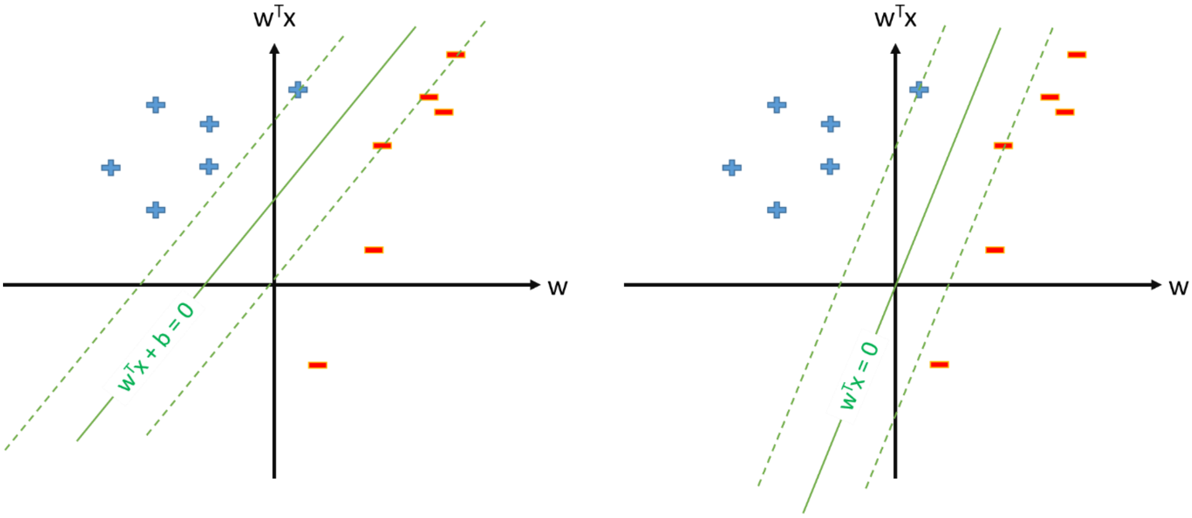
Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1, then it will become as:



Why bias is important?

The bias term b is a special parameter in SVM. Without it, the classifier will always go through the origin. So, SVM does not give you the separating hyperplane with the maximum margin if it does not happen to pass through the origin, unless you have a bias term.

Below is a visualization of the bias issue. An SVM trained with (without) a bias term is shown on the left (right). Even though both SVMs are trained on the **same data**, however, they look very different.



Why should the bias be treated separately?

As Ben DAI pointed out, the bias term b should be treated separately because of regularization. SVM maximizes the margin size, which is $1/||w||^2$ (or $2/||w||^2$ depending on how you define it).

Maximizing the margin is the same as minimizing $||w||^2$ .This is also called the *regularization term* and can be interpreted as a measure of the complexity of the classifier. However, you do not want to regularize the bias term because, the bias shifts the classification scores up or down **by the same amount for all data points**. In particular, the bias does not change the **shape** of the classifier or its margin size.

Regularization perspectives on support-vector machines provide a way of interpreting support-vector machines (SVMs) in the context of other machine-learning algorithms. SVM algorithms categorize multidimensional data, with the goal of fitting the training set data well, but also avoiding overfitting, so that the solution generalizes to new data points. Regularization algorithms also aim to fit training set data and avoid overfitting. They do this by choosing a fitting function that has low error on the training set, but also is not too complicated, where complicated functions are functions with high norms in some function space.

UNIT – 3

Unsupervised Learning:

Unsupervised Learning is a machine learning technique in which the users do not need to supervise the model. Instead, it allows the model to work on its own to discover patterns and information that was previously undetected. It mainly deals with the unlabelled data.

Unsupervised Learning Algorithms

Unsupervised Learning Algorithms allow users to perform more complex processing tasks compared to supervised learning. Although, unsupervised learning can be more unpredictable compared with other natural learning methods. Unsupervised learning algorithms include clustering, anomaly detection, neural networks, etc.

Why Unsupervised Learning?

Here, are prime reasons for using Unsupervised Learning:

- Unsupervised machine learning finds all kind of unknown patterns in data.
- Unsupervised methods help you to find features which can be useful for categorization.
- It is taken place in real time, so all the input data to be analyzed and labeled in the presence of learners.
- It is easier to get unlabeled data from a computer than labeled data, which needs manual intervention.

Types of Unsupervised Learning

Unsupervised learning problems further grouped into clustering and association problems.

Clustering

Clustering is an important concept when it comes to unsupervised learning. It mainly deals with finding a structure or pattern in a collection of uncategorized data. Clustering algorithms will process your data and find natural clusters(groups) if they exist in the data. You can also modify how many clusters your algorithms should identify. It allows you to adjust the granularity of these groups.

Why use Clustering?

Grouping similar entities together help profile the attributes of different groups. In other words, this will give us insight into underlying patterns of different groups. There are many applications of grouping unlabeled data, for example, you can identify different groups/segments of customers and market each group in a different way to maximize the revenue. Another example is grouping documents together which belong to the similar topics etc.

There are different types of clustering you can utilize:

Exclusive (partitioning)

In this clustering method, Data are grouped in such a way that one data can belong to one cluster only.

Example: K-means

Agglomerative

In this clustering technique, every data is a cluster. The iterative unions between the two nearest clusters reduce the number of clusters.

Example: Hierarchical clustering

Overlapping

In this technique, fuzzy sets is used to cluster data. Each point may belong to two or more clusters with separate degrees of membership.

Here, data will be associated with an appropriate membership value. Example: Fuzzy C-Means

Probabilistic

This technique uses probability distribution to create the clusters

Example: Following keywords

- "man's shoe."
- "women's shoe."
- "women's glove."
- "man's glove."

can be clustered into two categories "shoe" and "glove" or "man" and "women."

Clustering Types

- Hierarchical clustering
- K-means clustering
- K-NN (k nearest neighbors)
- Principal Component Analysis
- Singular Value Decomposition
- Independent Component Analysis

Hierarchical Clustering:

Hierarchical clustering is an algorithm which builds a hierarchy of clusters. It begins with all the data which is assigned to a cluster of their own. Here, two close cluster are going to be in the same cluster. This algorithm ends when there is only one cluster left.

K-means Clustering

K means it is an iterative clustering algorithm which helps you to find the highest value for every iteration. Initially, the desired number of clusters are selected. In this clustering method, you need to cluster the data points into k groups. A larger k means smaller groups with more granularity in the same way. A lower k means larger groups with less granularity.

The output of the algorithm is a group of "labels." It assigns data point to one of the k groups. In k-means clustering, each group is defined by creating a centroid for each group. The centroids are like the heart of the cluster, which captures the points closest to them and adds them to the cluster.

K-mean clustering further defines two subgroups:

- Agglomerative clustering
- Dendrogram

Agglomerative clustering:

This type of K-means clustering starts with a fixed number of clusters. It allocates all data into the exact number of clusters. This clustering method does not require the number of clusters K as an input. Agglomeration process starts by forming each data as a single cluster.

This method uses some distance measure, reduces the number of clusters (one in each iteration) by merging process. Lastly, we have one big cluster that contains all the objects.

Dendrogram:

In the Dendrogram clustering method, each level will represent a possible cluster. The height of dendrogram shows the level of similarity between two join clusters. The closer to the bottom of the process they are more similar cluster which is finding of the group from dendrogram which is not natural and mostly subjective.

K- Nearest neighbors

K- nearest neighbour is the simplest of all machine learning classifiers. It differs from other machine learning techniques, in that it doesn't produce a model. It is a simple algorithm which stores all available cases and classifies new instances based on a similarity measure.

It works very well when there is a distance between examples. The learning speed is slow when the training set is large, and the distance calculation is nontrivial.

Principal Components Analysis:

In case you want a higher-dimensional space. You need to select a basis for that space and only the 200 most important scores of that basis. This base is known as a principal component. The subset you select constitute is a new space which is small in size compared to original space. It maintains as much of the complexity of data as possible.

Association

Association rules allow you to establish associations amongst data objects inside large databases. This unsupervised technique is about discovering interesting relationships between variables in large databases. For example, people that buy a new home most likely to buy new furniture.

Other Examples:

- A subgroup of cancer patients grouped by their gene expression measurements
- Groups of shopper based on their browsing and purchasing histories
- Movie group by the rating given by movies viewers.

Applications of unsupervised machine learning

Some applications of unsupervised machine learning techniques are:

- Clustering automatically split the dataset into groups base on their similarities
- Anomaly detection can discover unusual data points in your dataset. It is useful for finding fraudulent transactions
- Association mining identifies sets of items which often occur together in your dataset
- Latent variable models are widely used for data pre-processing. Like reducing the number of features in a dataset or decomposing the dataset into multiple components

Disadvantages of Unsupervised Learning

- You cannot get precise information regarding data sorting, and the output as data used in unsupervised learning is labelled and not known
- Less accuracy of the results is because the input data is not known and not labelled by people in advance. This means that the machine requires to do this itself.
- The spectral classes do not always correspond to informational classes.
- The user needs to spend time interpreting and label the classes which follow that classification.
- Spectral properties of classes can also change over time so you can't have the same class information while moving from one image to another.

CLUSTER VALIDATION:

The term **cluster validation** is used to design the procedure of evaluating the goodness of clustering algorithm results. This is important to avoid finding patterns in a random data, as well as, in the situation where you want to compare two clustering algorithms.

Generally, clustering validation statistics can be categorized into 3 classes:

1. **Internal cluster validation**, which uses the internal information of the clustering process to evaluate the goodness of a clustering structure without reference to external information. It can be also used for estimating the number of clusters and the appropriate clustering algorithm without any external data.
2. **External cluster validation**, which consists in comparing the results of a cluster analysis to an externally known result, such as externally provided class labels. It measures the extent to which cluster labels match externally supplied class labels. Since we know the "true" cluster number in advance, this approach is mainly used for selecting the right clustering algorithm for a specific data set.
3. **Relative cluster validation**, which evaluates the clustering structure by varying different parameter values for the same algorithm (e.g.,: varying the number of clusters k). It's generally used for determining the optimal number of clusters.

Internal measures for cluster validation

Recall that the goal of partitioning clustering algorithms is to split the data set into clusters of objects, such that:

- the objects in the same cluster are similar as much as possible,
- and the objects in different clusters are highly distinct

That is, we want the average distance within cluster to be as small as possible; and the average distance between clusters to be as large as possible.

Internal validation measures reflect often the **compactness**, the **connectedness** and the **separation** of the cluster partitions.

1. **Compactness** or cluster cohesion: Measures how close are the objects within the same cluster. A lower **within-cluster variation** is an indicator of a good compactness (i.e., a good clustering). The different indices for evaluating the compactness of clusters are based on distance measures such as the cluster-wise within average/median distances between observations.
2. **Separation**: Measures how well-separated a cluster is from other clusters. The indices used as separation measures include:
   - distances between cluster centers
   - the pairwise minimum distances between objects in different clusters
3. **Connectivity**: corresponds to what extent items are placed in the same cluster as their nearest neighbors in the data space. The connectivity has a value between 0 and infinity and should be minimized.

Generally most of the indices used for internal clustering validation combine compactness and separation measures as follow:

$$\text{Index} = \frac{\boldsymbol{\alpha \times Separation}}{\boldsymbol{\beta \times Compactness}}$$

Where α and β are weights.

Silhouette coefficient

The silhouette analysis measures how well an observation is clustered and it estimates the **average distance between clusters**. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters.

For each observation *I* , the silhouette width *si* is calculated as follows:

1. For each observation *i*, calculate the average dissimilarity a_i between *i* and all other points of the cluster to which i belongs.
2. For all other clusters *C*, to which i does not belong, calculate the average dissimilarity *d*(*i*,*C*) of *i* to all observations of C. The smallest of these *d*(*i*,*C*) is defined as $b_i=\min_C$ *d*(*i*,*C*). The value of b_i can be seen as the dissimilarity between *I* and its "neighbor" cluster, i.e., the nearest one to which it does not belong.
3. Finally the silhouette width of the observation *i* is defined by the formula: *Si*=(*bi*−*ai*)/*max*(*ai*,*bi*).

Silhouette width can be interpreted as follow:

- Observations with a large S_i (almost 1) are very well clustered.
- A small S_i (around 0) means that the observation lies between two clusters.
- Observations with a negative S_i are probably placed in the wrong cluster.

Dunn index

The **Dunn index** is another internal clustering validation measure which can be computed as follow:

1. For each cluster, compute the distance between each of the objects in the cluster and the objects in the other clusters
2. Use the minimum of this pairwise distance as the inter-cluster separation (*min.separation*)
3. For each cluster, compute the distance between the objects in the same cluster.
4. Use the maximal intra-cluster distance (i.e maximum diameter) as the intra-cluster compactness
5. Calculate the *Dunn index* (D) as follow:

$$D=\frac{\boldsymbol{min\ Separation}}{\boldsymbol{max\ Diameter}}$$

External measures for clustering validation

The aim is to compare the identified clusters (by k-means, pam or hierarchical clustering) to an external reference.

It's possible to quantify the agreement between partitioning clusters and external reference using either the corrected *Rand index* and *Meila's variation index VI*, which are implemented in the R function *cluster.stats*()[*fpc* package].

The corrected *Rand index* varies from -1 (no agreement) to 1 (perfect agreement).

External clustering validation, can be used to select suitable clustering algorithm for a given data set.