# Unit 2

# CLASSIFICATION

Supervised Machine Learning algorithm can be broadly classified into Regression and Classification Algorithms. In Regression algorithms, we have predicted the output for continuous values, but to predict the categorical values, we need Classification algorithms.

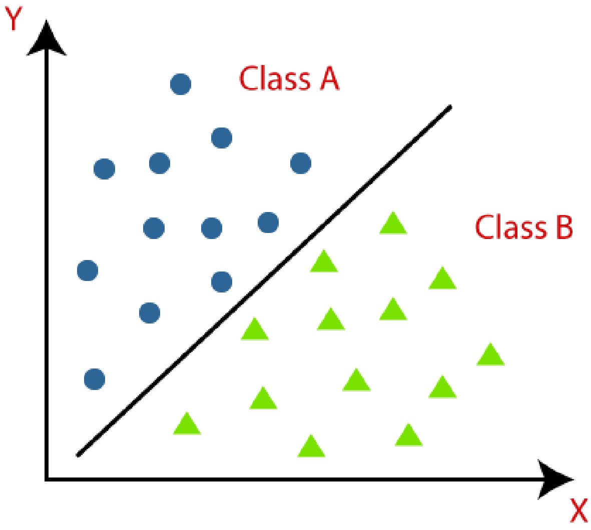
## 2.1 Classification Algorithm:

The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations based on training data. In Classification, a program learns from the given dataset or observations and then classifies new observation into several classes or groups. Such as, Yes or No, 0 or 1, Spam or Not Spam, cat, or dog, etc. Classes can be called as targets/labels or categories.

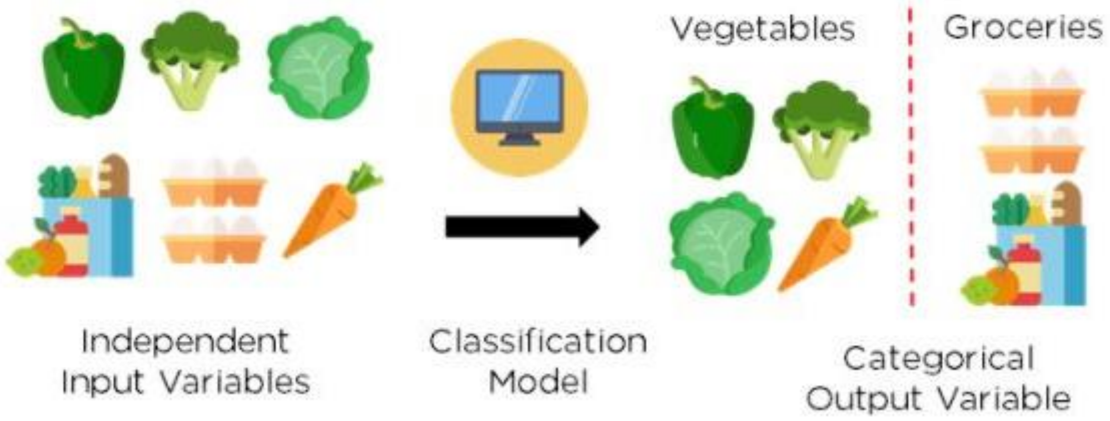
Unlike regression, the output variable of Classification is a category, not a value, such as "Green or Blue", "fruit or animal", etc. Since the Classification algorithm is a Supervised learning technique, hence it takes labelled input data, which means it contains input with the corresponding output.

In classification algorithm, a discrete output function(y) is mapped to input variable(x).

The main goal of the Classification algorithm is to identify the category of a given dataset, and these algorithms are mainly used to predict the output for the categorical data. Classification algorithms can be better understood using the below diagram. In the below diagram, there are two classes, class A and Class B. These classes have features that are like each other and dissimilar to other classes.



The algorithm which implements the classification on a dataset is known as a classifier.



There are two types of Classifications:

- Binary Classifier: If the classification problem has only two possible outcomes, then it is called as Binary Classifier.

Examples: YES or NO, MALE or FEMALE, SPAM or NOT SPAM, CAT, or DOG, etc.

- Multi-class Classifier: If a classification problem has more than two outcomes, then it is called as Multi-class Classifier.

Examples: Classifications of types of crops, Classification of types of music.

The best example of an ML classification algorithm is Email Spam Detector.

2.1.1 Learners in Classification Problems:

In the classification problems, there are two types of learners:

1. Lazy Learners: Lazy Learner firstly stores the training dataset and wait until it receives the test dataset. In Lazy learner case, classification is done based on the most related data stored in the training dataset. It takes less time in training but more time for predictions.

Example: K-NN algorithm, Case-based reasoning

2. Eager Learners: Eager Learners develop a classification model based on a training dataset before receiving a test dataset. Opposite to Lazy learners, Eager Learner takes more time in learning, and less time in prediction.

Example: Decision Trees, Naïve Bayes, ANN.

2.1.2 Types of ML Classification Algorithms:

Classification Algorithms can be further divided into the Mainly two category:

1. Linear Models
   - Logistic Regression
   - Support Vector Machines

2. Non-linear Models
    - K-Nearest Neighbours
    - Kernel SVM
    - Naïve Bayes
    - Decision Tree Classification
    - Random Forest Classification

Classification versus Prediction

Classification: Predicts categorical class labels.

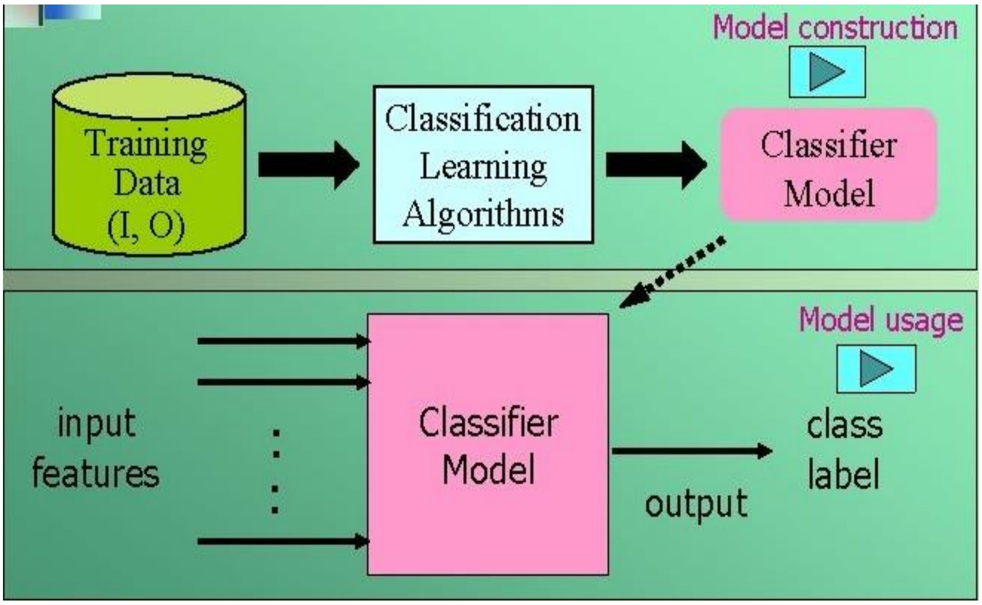
Classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data.

Prediction: Models continuous-valued functions, i.e., predicts unknown or missing values.

Classification is a two-step process

Model Construction: Describing a set of predetermined classes. Each sample/record is assumed to belong to a predefined class, as determined by the class label attribute. The set of samples used for model construction is the training set. The model is represented as classification rules, decision trees, or mathematical formulae.

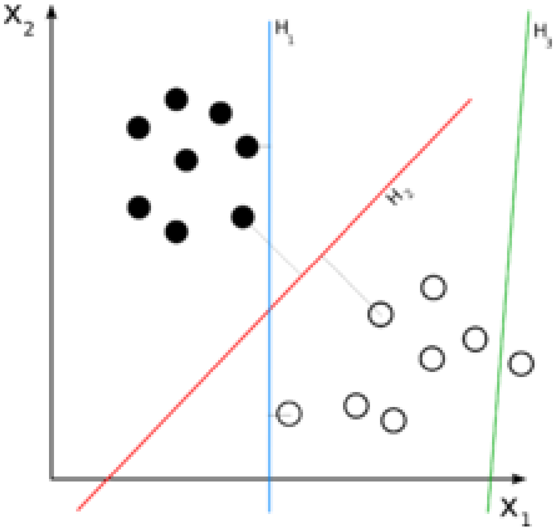
Model usage: For classifying future or unknown objects need to estimate the accuracy of the model by comparing known label of a test sample with the classified results from the model. Accuracy rate is the percentage of test set samples that are correctly classified by the model. The test set is independent of the training set, otherwise over-fitting will occur. If the accuracy is acceptable, use the model to classify data objects whose class labels are not known.



### 2.1.3 Linear Classification model:

A linear classifier achieves this by making a classification decision based on the value of a linear combination of the characteristics. An object's characteristics are also known as feature values and are typically presented to the machine in a vector called a feature vector. Such classifiers work well for practical problems such as document classification, and more generally for problems with many variables (features), reaching accuracy levels comparable to non-linear classifiers while taking less time to train and use. A linear classifier is a model that makes a decision to categories a set of data points to a discrete class based on a linear combination of its explanatory variables.

Ex: combining details about a dog such as weight, height, colour, and other features would be used by a model to decide its species. The effectiveness of these models lies in their ability to find this mathematical combination of features that groups data points together when they have the same class and separate them when they have different classes, providing us with clear boundaries for how to classify.

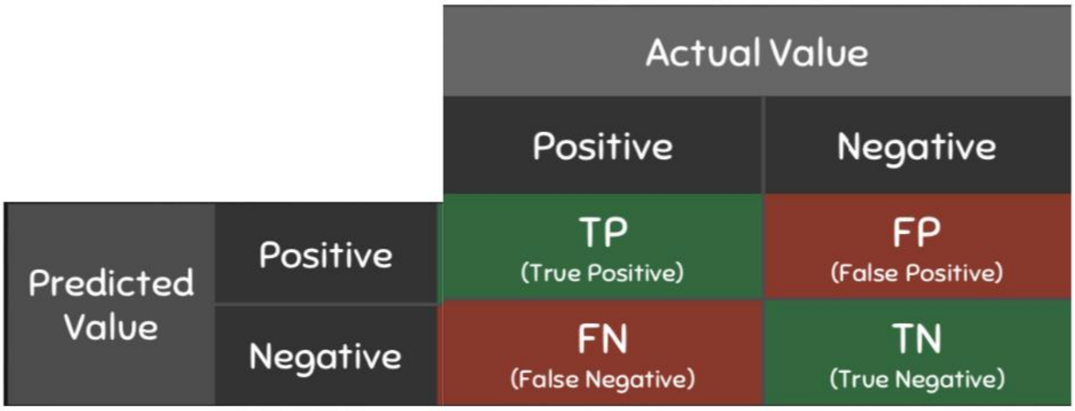


If the input feature vector to the classifier is a real vector to the classifier is a real vector , then the output score is

$$y = f(\vec{w} \cdot \vec{x}) = f\left(\sum_j w_j x_j\right)$$

## 2.2 Performance Evaluation:

- Confusion Matrix: As the target variable is not continuous, binary classification model predicts the probability of a target variable to be Yes/No. To evaluate such a model, a metric called the confusion matrix is used, also called the classification or co-incidence matrix. With the help of a confusion matrix, we can calculate important performance measures:



| | | Actual Value | |
|---|---|---|---|
| | | Positive | Negative |
| Predicted Value | Positive | TP<br>(True Positive) | FP<br>(False Positive) |
| | Negative | FN<br>(False Negative) | TN<br>(True Negative) |

  - **TP means True Positive:** It can be interpreted as the model predicted positive class and it is True.
  - **FP means False Positive:** It can be interpreted as the model predicted positive class, but it is False.
  - **FN means False Negative:** It can be interpreted as the model predicted negative class, but it is False.
  - **TN means True Negative:** It can be interpreted as the model predicted negative class and it is True.

**Use case:** Let's take an example of a patient who has gone to a doctor with certain symptoms. Since it's the season of Covid, let's assume that he went with fever, cough, throat ache, and cold. These are symptoms that can occur during any seasonal changes too. Hence, it is tricky for the doctor to do the right diagnosis.

True Positive (TP):

Let's say the patient was actually suffering from Covid and on doing the required assessment, the doctor classified him as a Covid patient. This is called TP or True Positive. This is because the case is positive in real and at the same time the case was classified correctly. Now, the patient can be given appropriate treatment which means, the decision made by the doctor will have a positive effect on the patient and society.

False Positive (FP):

Let's say the patient was not suffering from Covid and he was only showing symptoms of seasonal flu, but the doctor diagnosed him with Covid. This is called FP or False Positive. This is because the case was actually negative but was falsely classified as positive. Now, the patient will end up getting admitted to the hospital or home and will be given treatment for Covid. This is an unnecessary inconvenience for him and others as he will get unwanted treatment and quarantine. This is called Type I Error.

True Negative (TN):

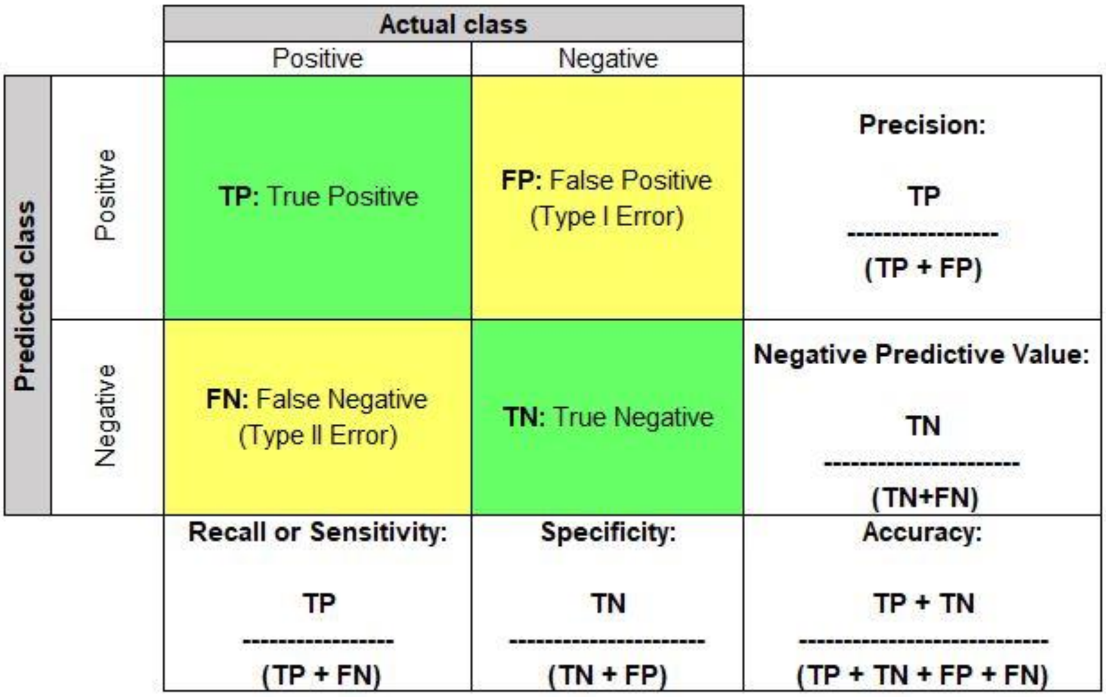
Let’s say the patient was not suffering from Covid and the doctor also gave him a clean chit. This is called TN or True Negative. This is because the case was actually negative and was also classified as negative which is the right thing to do. Now the patient will get treatment for his actual illness instead of taking Covid treatment.

False Negative (FN):

Let’s say the patient was suffering from Covid and the doctor did not diagnose him with Covid. This is called FN or False Negative as the case was actually positive but was falsely classified as negative. Now the patient will not get the right treatment and also, he will spread the disease to others. This is a highly dangerous situation in this example. This is also called Type II Error.

Summary: In this particular example, both FN and FP are dangerous and the classification model which has the lowest FN and FP values needs to be chosen for implementation. But in case there is a tie between few models which score very similar when it comes to FP and FN, in this scenario the model with the least FN needs to be chosen. This is because we simply cannot afford to have FNs! The goal of the hospital would be to not let even one patient go undiagnosed (no FNs) even if some patients get diagnosed wrongly (FPs) and asked to go under quarantine and special care.

Here is how a confusion matrix looks like:



| | | **Actual class** | | |
|---|---|---|---|---|
| | | Positive | Negative | |
| **Predicted class** | Positive | **TP:** True Positive | **FP:** False Positive<br>(Type I Error) | **Precision:**<br>**TP**<br>**-----------------**<br>**(TP + FP)** |
| | Negative | **FN:** False Negative<br>(Type II Error) | **TN:** True Negative | **Negative Predictive Value:**<br>**TN**<br>**----------------------**<br>**(TN+FN)** |
| | | **Recall or Sensitivity:**<br>**TP**<br>**-----------------**<br>**(TP + FN)** | **Specificity:**<br>**TN**<br>**----------------------**<br>**(TN + FP)** | **Accuracy:**<br>**TP + TN**<br>**----------------------------**<br>**(TP + TN + FP + FN)** |

- Accuracy: Accuracy is the simple ratio between the number of correctly classified points to the total number of points.

Accuracy = (TP + TN) / (TP + FP +TN + FN)

This term tells us how many right classifications were made out of all the classifications. In other words, how many TPs and TNs were done out of TP + TN + FP + FNs. It tells the ratio of “True”s to the sum of “True”s and “False”s.

Use case: Out of all the patients who visited the doctor, how many were correctly diagnosed as Covid positive and Covid negative.

- Precision: Precision is the fraction of the correctly classified instances from the total classified instances. Precision is the fraction of true positive examples among the examples that the model classified as positive. In other words, the number of true positives divided by the number of false positives plus true positives.

$$Precision = \frac{TP}{TP+FP}$$

Low precision: the more False positives the model predicts, the lower the precision.

Use case: Let’s take another example of a classification algorithm that marks emails as spam or not. Here, if emails that are of importance get marked as positive, then useful emails will end up going to the “Spam” folder, which is dangerous. Hence, the classification model which has the least FP value needs to be selected. In other words, a model that has the highest precision needs to be selected among all the models.

- Recall or Sensitivity: Recall is the fraction of the correctly classified instances from the total classified instances. The number of true positives divided by the number of true positives plus false negatives.

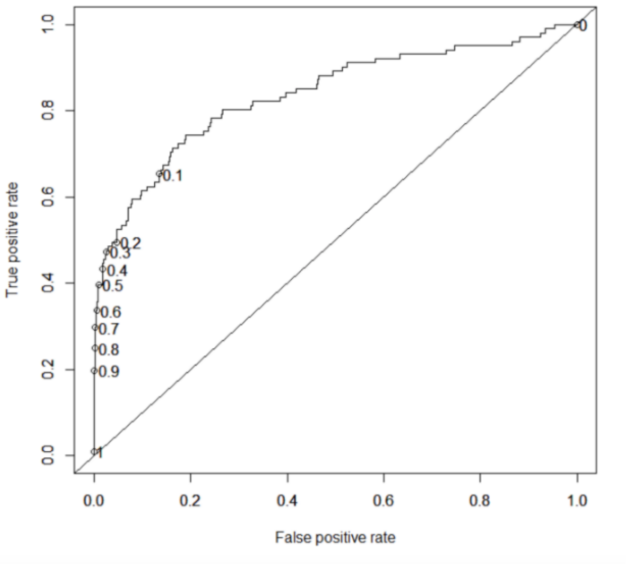
Low recall: the more False Negatives the model predicts, the lower the recall.

$$Recall = \frac{TP}{TP+FN}$$

Use case: Out of all the actual Covid patients who visited the doctor, how many were actually diagnosed as Covid positive. Hence, the classification model which has the least FN value needs to be selected. In other words, a model that has the highest recall value needs to be selected among all the models.

Precision helps us understand how useful the results are. Recall helps us understand how complete the results are.

- ROC Curves: A Receiver Operating Characteristic curve or ROC curve is created by plotting the True Positive (TP) against the False Positive (FP) at various threshold settings. The ROC curve is generated by plotting the cumulative distribution function of the True Positive in the y-axis versus the cumulative distribution function of the False Positive on the x-axis.



F-Measure: Once precision and recall have been calculated for a binary classification problem, the two scores can be combined into the calculation of the F-Measure.

The traditional F measure is calculated as follows:

F-Measure = (2 * Precision * Recall) / (Precision + Recall)

$$
\begin{aligned}
F_1 &= \frac{2}{\frac{1}{\mathrm{recall}} \times \frac{1}{\mathrm{precision}}} = 2 \times \frac{\mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \\
&= \frac{\mathrm{tp}}{\mathrm{tp} + \frac{1}{2}(\mathrm{fp} + \mathrm{fn})}
\end{aligned}
$$

This is the harmonic mean of the two fractions. This is sometimes called the F-Score or the F1-Score and might be the most common metric used on imbalanced classification problems.

2.3 Multi-class Classification: Multiclass classification is a classification task with more than two classes. Each sample can only be labelled as one class. Each training point belongs to one of N different classes. The goal is to construct a function which, given a new data point, will correctly predict the class to which the new point belongs. There are many scenarios in which there are multiple categories to which points belong, but a given point can belong to multiple categories. In its most basic form, this problem decomposes trivially into a set of unlinked binary problems, which can be solved naturally using our techniques for binary classification.

For example, classification using features extracted from a set of images of fruit, where each image may either be of an orange, an apple, or a pear. Each image is one sample and is labelled as one of the 3 possible classes. Multiclass classification assumes that each sample is assigned to one and only one label - one sample cannot, for example, be both a pear and an apple

2.3.1 Binary vs Multiclass Classification:

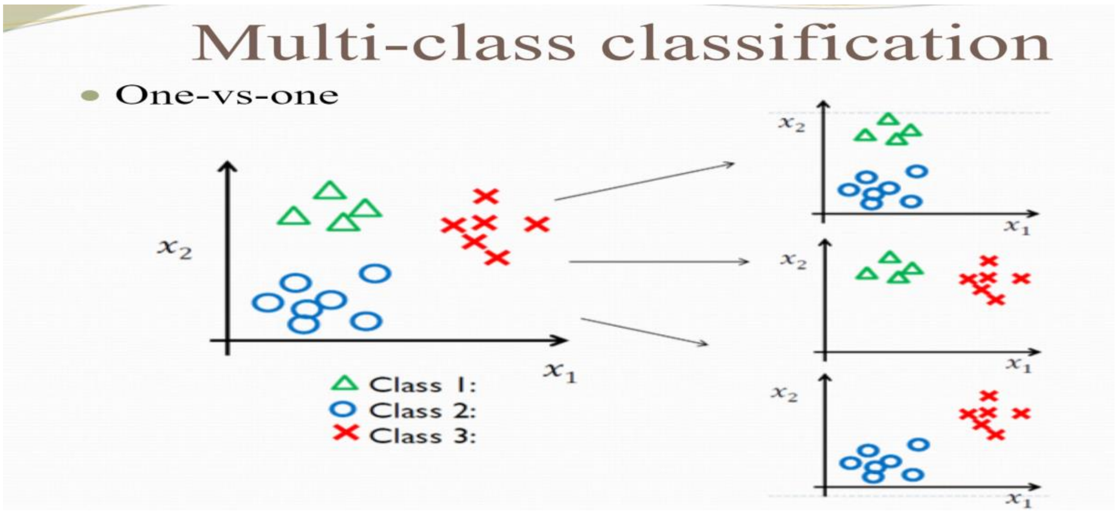
| Parameters | Binary classification | Multi-class classification |
|---|---|---|
| No. of classes | It is a classification of two groups, i.e., classifies objects in at most two classes. | There can be any number of classes in it, i.e., classifies the object into more than two classes. |
| Algorithms used | The most popular algorithms used by the binary classification are-<br>Logistic Regression<br>k-Nearest Neighbours<br>Decision Trees<br>Support Vector Machine<br>Naive Bayes | Popular algorithms that can be used for multi-class classification include:<br>k-Nearest Neighbours<br>Decision Trees<br>Naive Bayes<br>Random Forest.<br>Gradient Boosting |
| Examples | Email spam detection (spam or not).<br>Churn prediction (churn or not).<br>Conversion prediction (buy or not). | Face classification.<br>Plant species classification.<br>Optical character recognition. |

2.4 Multiclass Classification techniques: multi-class classification techniques can be categorized into

1. One vs One
2. One vs Rest

1. One vs One: One-vs-One (OvO) is heuristic method for using binary classification algorithms for multi-class classification. One-vs-One splits a multi-class classification dataset into binary classification problems. The One-vs-One approach splits the dataset into one dataset for each class versus every other class. Each classifier trained on a small subset of data (only those labelled with those two classes would be involved), which can result in high variance. This component implements the one-versus-one method, in which a binary model is created per class pair. At prediction time, the class which received the most votes are selected. Since it requires to fit n* (n - 1) / 2 classifiers, this method is usually slower than one-versus-all, due to its O(n^2) complexity. However, this method may be

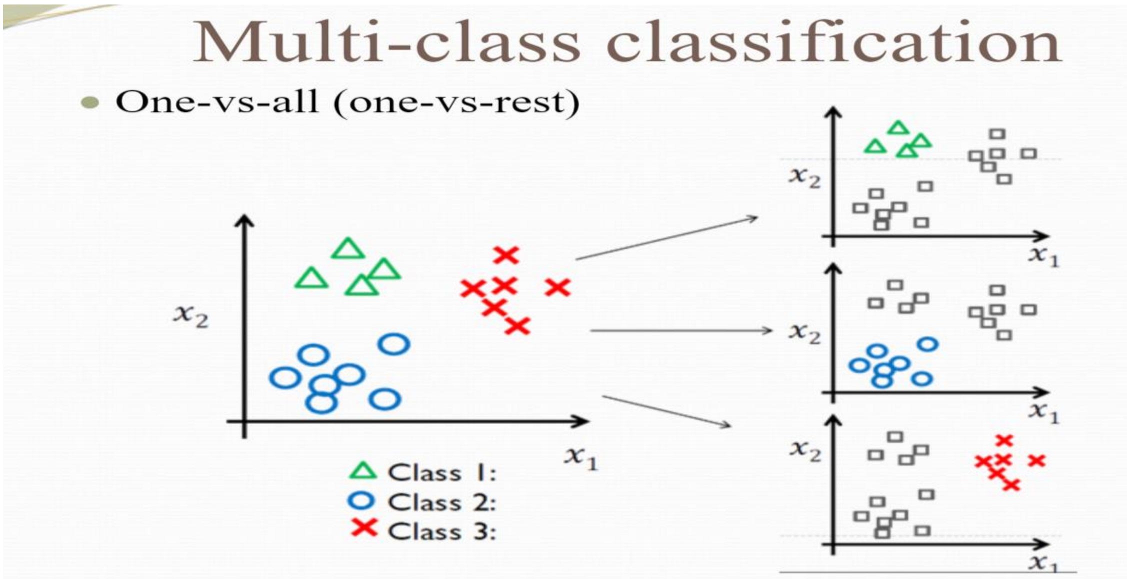
advantageous for algorithms such as kernel algorithms which don’t scale well with n_samples. This is because each individual learning problem only involves a small subset of the data whereas, with one-versus-all, the complete dataset is used n_classes times.



Multi-class classification

- One-vs-one

2. One vs Rest: One-vs-rest (OvR for short, also referred to as One-vs-All or OvA) is a heuristic method for using binary classification algorithms for multi-class classification. It involves splitting the multi-class dataset into multiple binary classification problems. A binary classifier is then trained on each binary classification problem and predictions are made using the model that is the most confident. The obvious approach is to use a one-versus-the-rest approach (also called one-vs-all), in which we train C binary classifiers, fc(x), where the data from class c is treated as positive, and the data from all the other classes is treated as negative.



Multi-class classification

- One-vs-all (one-vs-rest)

## 2.4 Linear Models:

Linear modelling in a classification context consists of regression followed by a transformation to return a categorical output and thereby producing a decision boundary. The two most used linear classification algorithms are logistic regression and linear support vector machines. In the field of machine learning, the goal of statistical classification is to use an object's characteristics to identify which class (or group) it belongs to. A linear classifier achieves this by making a classification decision based on the value of a linear combination of the characteristics.

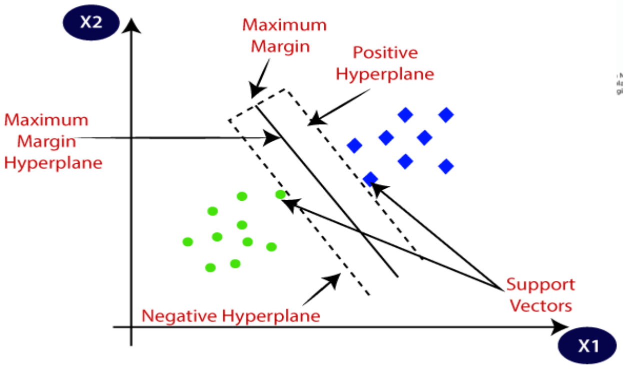
The mathematical formula for binary classification to make prediction is given below —

$$\hat{y} = x[0] * z[0] + x[1] * z[1] + \ldots + x[p] * z[p] + b > 0$$

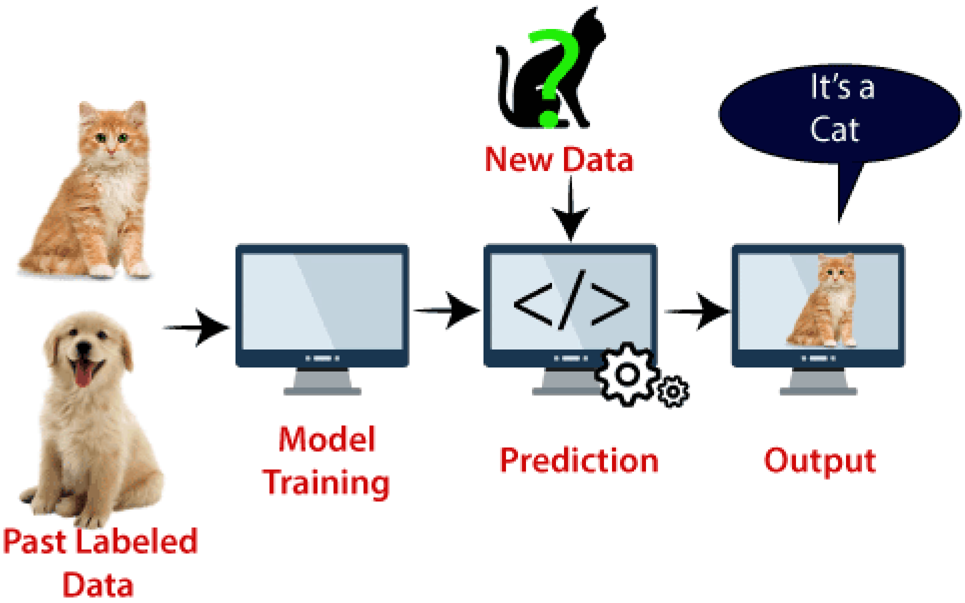
The formula is quite like the one used in linear regression, but here the weighted sum of the features is just returned. The threshold of the predicted value is zero in binary classification. If the function is less than zero, the class is predicted as -1 and if it is greater than zero, the class is predicted as +1. This common rule is used in case of all linear models for classification.

### 2.4.1 Linear Support Vector Machines (SVM):

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which 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:



Example: 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. Based on the support vectors, it will classify it as a cat. Consider the below diagram:

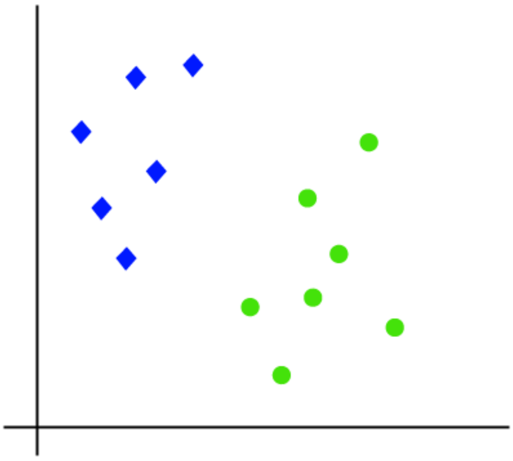


Types of SVM can be of two types:

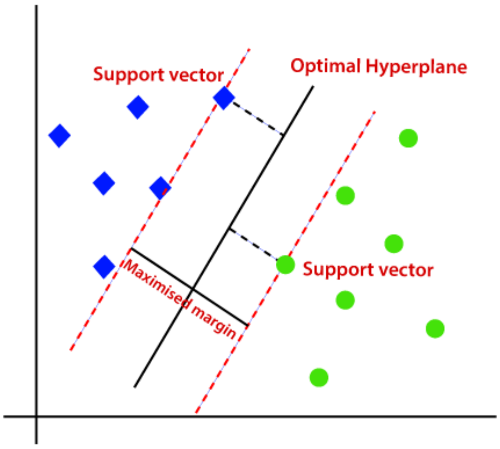
- Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- Non-linear SVM: Non-Linear SVM is used 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.

1. 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. Suppose we have a dataset that has two tags (green and blue), and

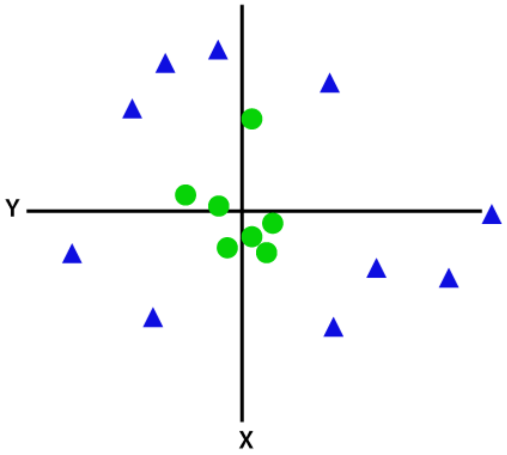
the dataset has two features x1 and x2. We want a classifier that can classify the pair (x1, x2) of coordinates in either green or blue. Consider the below image:



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.



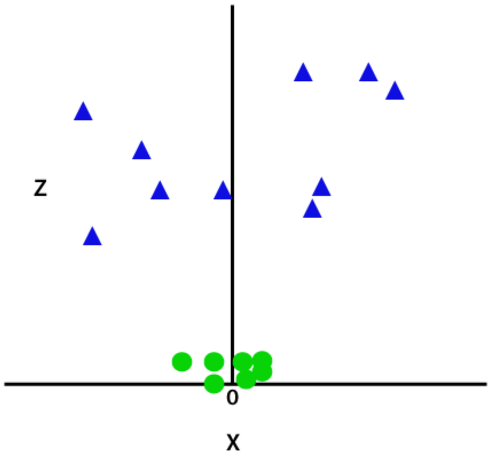
2. Non-Linear SVM: If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image:



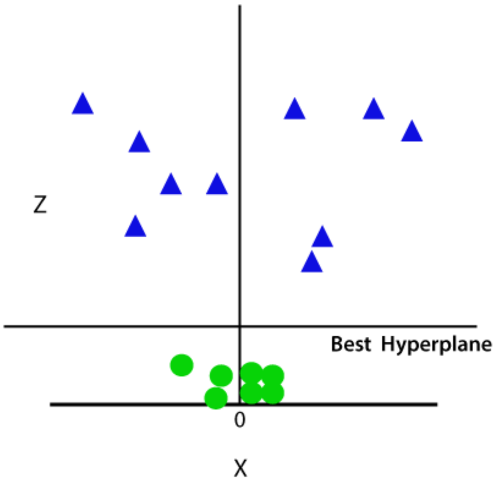
So, to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third-dimension z. It can be calculated as:

z=x2 +y2

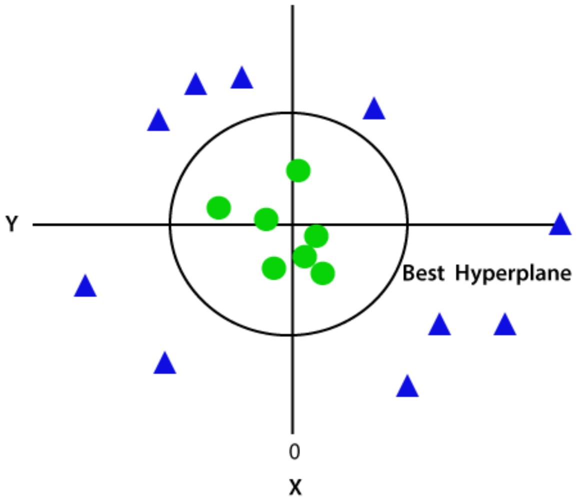
By adding the third dimension, the sample space will become as below image:



So now, SVM will divide the datasets into classes in the following way. Consider the below image:



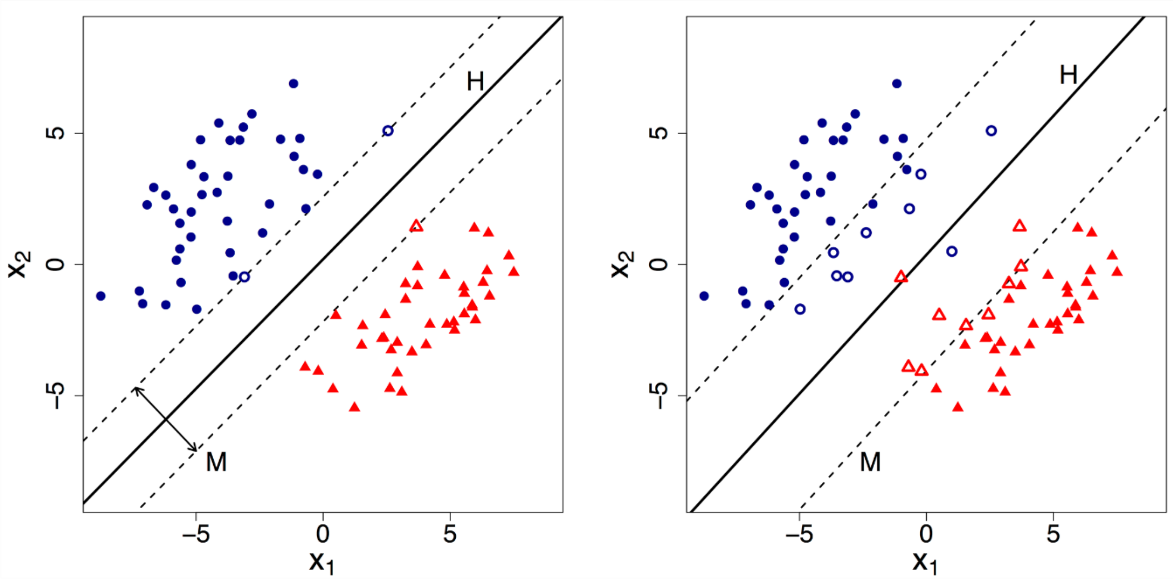
Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1, then it will become as:



Hence, we get a circumference of radius 1 in case of non-linear data.

2.5 Soft Margin SVM:

Choosing a correct classifier is really important. Let us understand this with an example.



Suppose we are given 2 Hyperplane one with 100% accuracy (HP1) on the left side and another with >90% accuracy (HP2) on the right side. Which one would you think is the correct classifier? Most of us would pick the HP2 thinking that it because of the maximum margin. But it is the wrong answer. But Support Vector Machine would choose the HP1 though it has a narrow margin. Because though HP2 has maximum margin but it is going against the constrain that: each data point must lie on the correct side of the margin and there should be no misclassification. This constrain is the hard constrain that Support Vector Machine follows throughout.

HP1 is a Hard SVM (left side) while HP2 is a Soft SVM (right side). By default, Support Vector Machine implements Hard margin SVM. It works well only if our data is linearly separable. Hard margin SVM does not allow any misclassification to happen. In case our data is non-separable/ nonlinear then the Hard margin SVM will not return any hyperplane as it will not be able to separate the data. Hence this is where Soft Margin SVM comes to the rescue. Soft margin SVM allows some misclassification to happen by relaxing the hard constraints of Support Vector Machine. Soft margin SVM is implemented with the help of the Regularization parameter (C): It tells us how much misclassification we want to avoid.

– Hard margin SVM generally has large values of C.

– Soft margin SVM generally has small values of C.

2.6 SVM Kernel to handle non-linear data:

SVM can be extended to solve nonlinear classification tasks when the set of samples cannot be separated linearly. By applying kernel functions, the samples are mapped onto a high-dimensional feature space, in which the linear classification is possible.

- Gaussian Radial Basis Function (RBF): It is one of the most preferred and used kernel functions in SVM. It is usually chosen for non-linear data. It helps to make proper separation when there is no prior knowledge of data.

  $$\mathbf{F\ (x,\ x_j) = exp\ (-\ \gamma\ *\ ||x\ -\ x_j||{}^\wedge 2)}$$

  The value of $\gamma$ (gamma) varies from 0 to 1. We must manually provide the value of $\gamma$ in the code. The most preferred value for gamma is 0.1.

- Gaussian Kernel: The Gaussian kernel is a very popular kernel function used in many machine learning algorithms, especially in support vector machines (SVMs). It is more often used than polynomial kernels when learning from nonlinear datasets and is usually employed in formulating the classical SVM for nonlinear problems. The Gaussian kernel function allows the separation of nonlinearly separable data by mapping the input vector to Hilbert space. The Gaussian kernel is an exponential function including norm and real constant.

$$K(X_1, X_2) = exponent(-\gamma \|X_1 - X_2\|^2)$$

- Polynomial: In general, the polynomial kernel is defined as

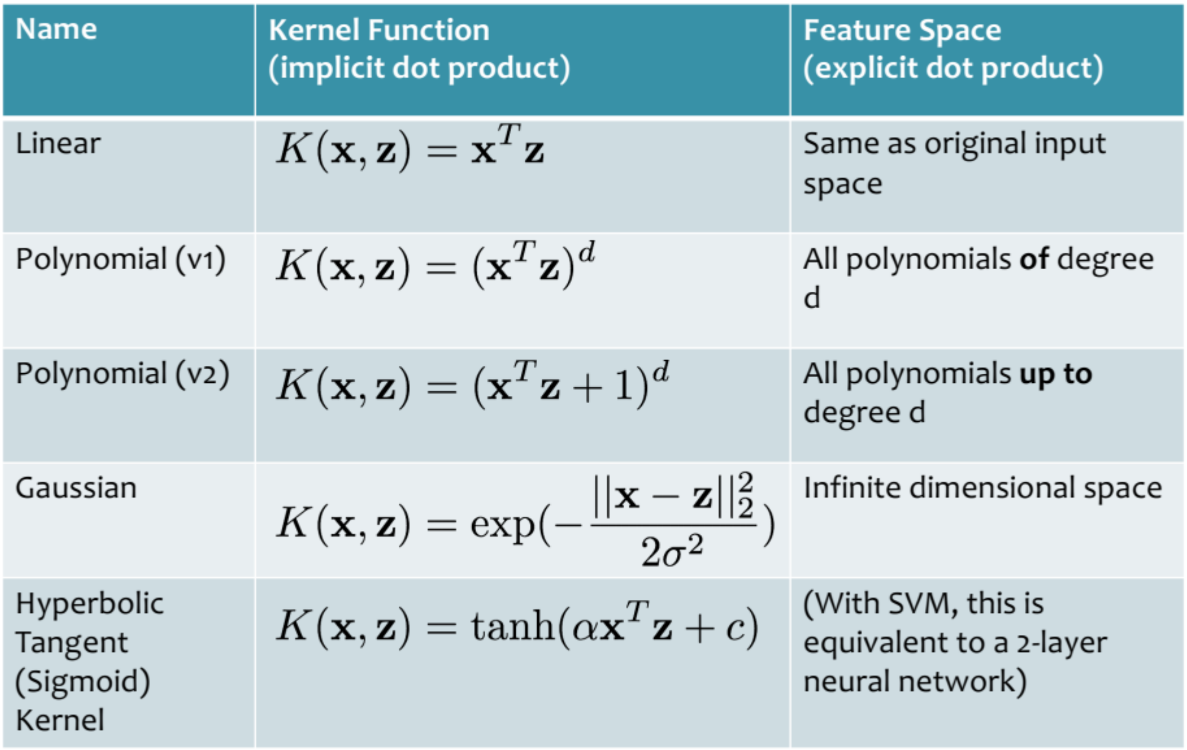
$$K(X_1, X_2) = (a + X_1^T X_2)^b$$

b = degree of kernel & a = constant term.

in the polynomial kernel, we simply calculate the dot product by increasing the power of the kernel.

- Sigmoid: This function is equivalent to a two-layer, perceptron model of the neural network, which is used as an activation function for artificial neurons. The sigmoid kernel was quite popular for support vector machines due to its origin from neural networks.

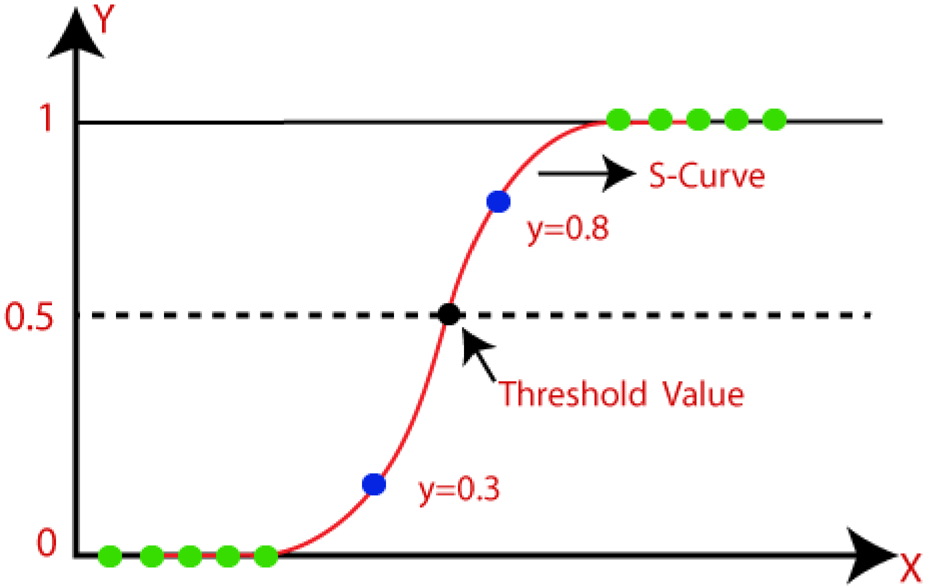
$$K(\mathbf{x}, \mathbf{z}) = \tanh(\alpha \mathbf{x}^T \mathbf{z} + c)$$



| Name | Kernel Function (implicit dot product) | Feature Space (explicit dot product) |
|---|---|---|
| Linear | $K(\mathbf{x}, \mathbf{z}) = \mathbf{x}^T \mathbf{z}$ | Same as original input space |
| Polynomial (v1) | $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T \mathbf{z})^d$ | All polynomials **of** degree d |
| Polynomial (v2) | $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T \mathbf{z} + 1)^d$ | All polynomials **up to** degree d |
| Gaussian | $K(\mathbf{x}, \mathbf{z}) = \exp(-\frac{\lvert\lvert \mathbf{x} - \mathbf{z} \rvert\rvert_2^2}{2\sigma^2})$ | Infinite dimensional space |
| Hyperbolic Tangent (Sigmoid) Kernel | $K(\mathbf{x}, \mathbf{z}) = \tanh(\alpha \mathbf{x}^T \mathbf{z} + c)$ | (With SVM, this is equivalent to a 2-layer neural network) |

2.7 Logistic Regression: Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or

False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. Logistic Regression is much like the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems. In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc. Logistic Regression is a significant machine learning algorithm because it can provide probabilities and classify new data using continuous and discrete datasets. Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. This type of statistical model (also known as logit model) is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure.



Logistic regression uses the concept of predictive modelling as regression; therefore, it is called logistic regression, but is used to classify samples; Therefore, it falls under the classification algorithm.

The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:

We know the equation of the straight line can be written as:

$$y= b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):

$$\frac{y}{1-y}\text{; 0 for y= 0, and infinity for y=1}$$

But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

$$log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

The above equation is the final equation for Logistic Regression.

2.7.1 Type of Logistic Regression:

On the basis of the categories, Logistic Regression can be classified into three types:

- Binomial: In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.
- Multinomial: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep".
- Ordinal: In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

2.7.2 Steps in Logistic Regression: To implement the Logistic Regression using Python, we will use the same steps as we have done in previous topics of Regression. Below are the steps:

- Data Pre-processing step
- Fitting Logistic Regression to the Training set
- Predicting the test result
- Test accuracy of the result (Creation of Confusion matrix)
- Visualizing the test set result.