

Machine learning Internship Day_3

Presented by,
Anjana G



Contents

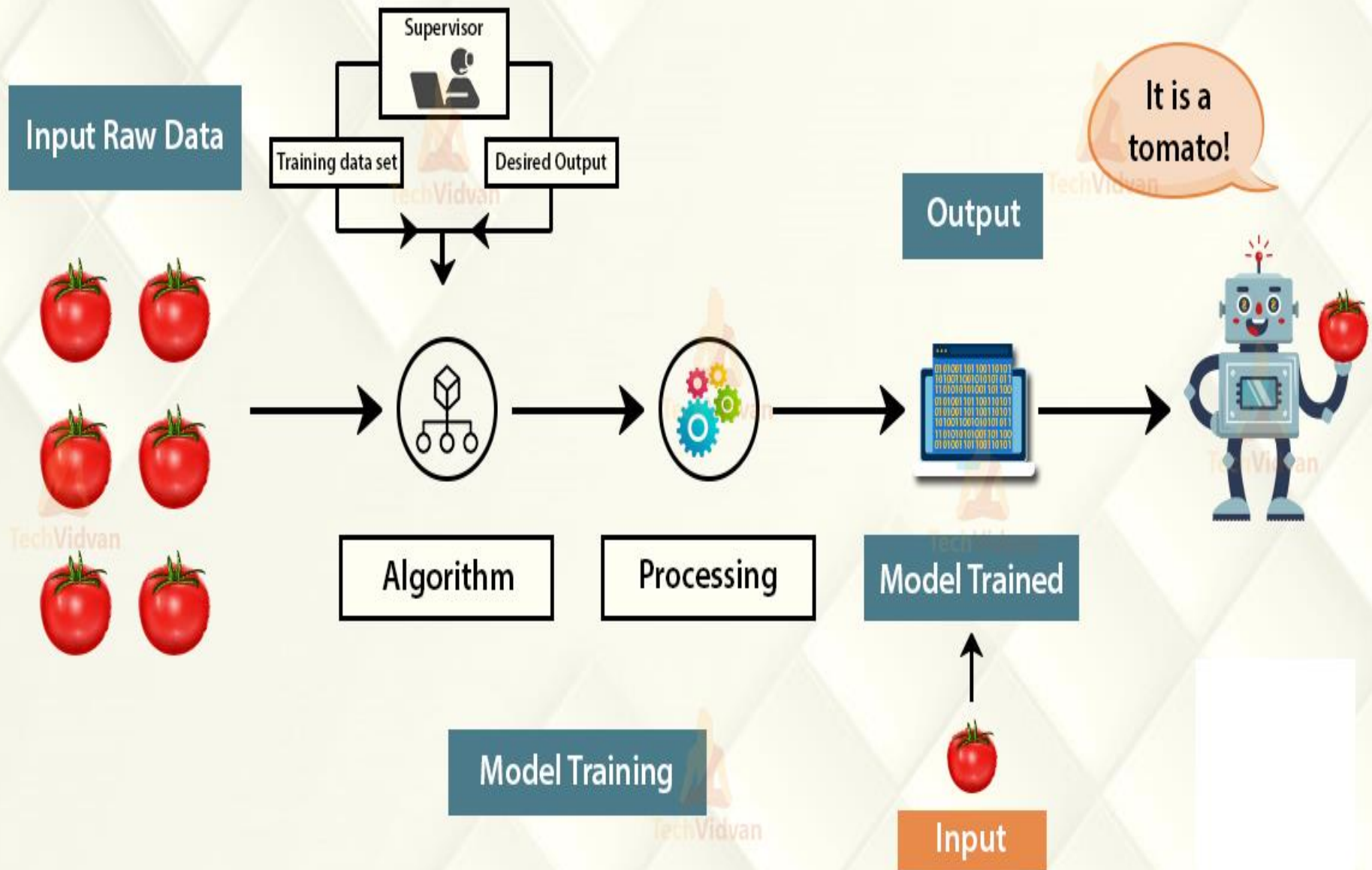
SUPERVISED LEARNING ALGORITHMS

- **Regression**
- **Classification**
- **Support Vector Machine**
- **Naïve Bayesian Model**



Supervised Learning...

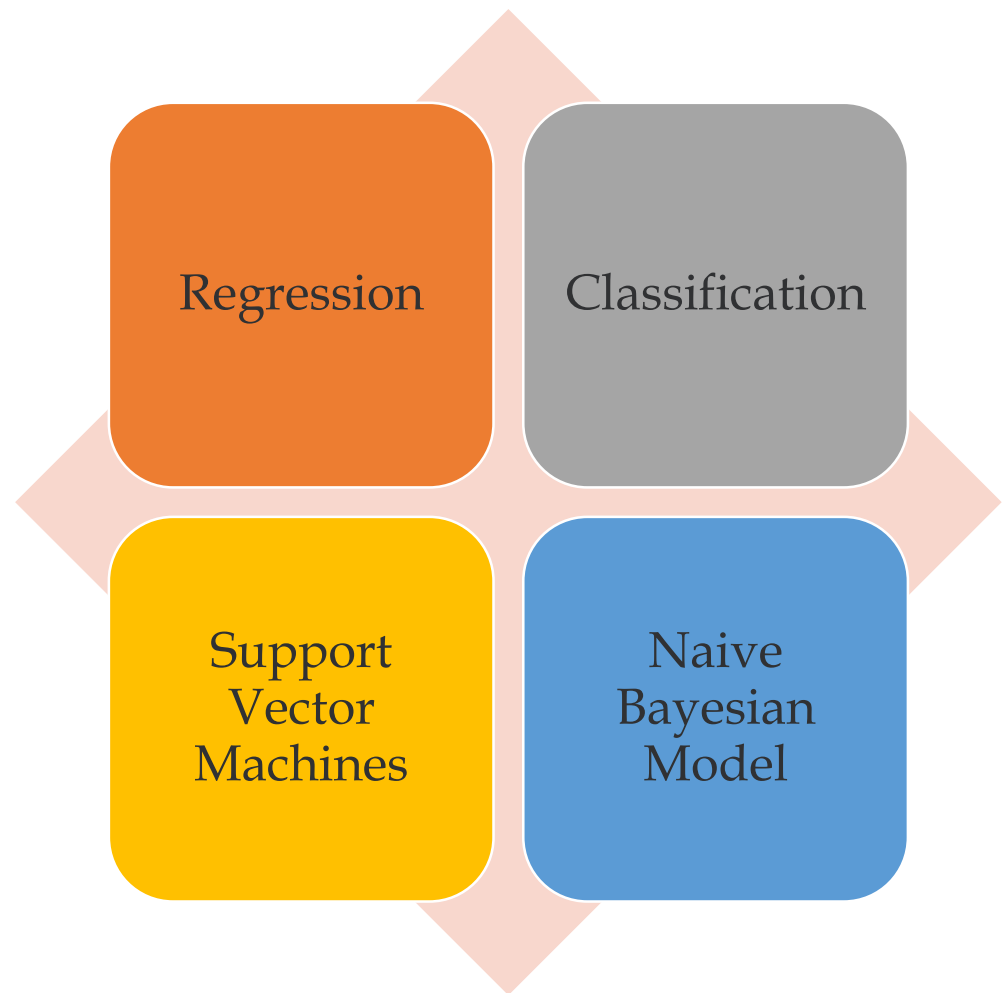
Supervised Learning in ML



What is Supervised Learning?

- In Supervised Learning, a machine is trained using 'labeled' data.
- Datasets are said to be labeled when they contain both input and output parameters.
- In other words, the data has already been tagged with the correct answer.
- So, the technique mimics a classroom environment where a student learns in the presence of a supervisor or teacher.
- Supervised machine learning is immensely helpful in solving real-world computational problems.
- The algorithm predicts outcomes for unforeseen data by learning from labeled training data.

Types of Supervised Learning



REGRESSION ANALYSIS

- Regression analysis is a form of predictive modelling technique which investigates the relationship between a dependent (target) and independent variable (s) (predictor).
- This technique is used for forecasting, time series modelling and finding the causal effect relationship between the variables.
- Regression analysis is an important tool for modelling and analyzing data.
- Here, we fit a curve / line to the data points, in such a manner that the differences between the distances of data points from the curve or line is minimized.

Why do we use Regression Analysis?

- Regression analysis estimates the relationship between two or more variables.
- Let's understand this with an easy example:
- Let's say, you want to estimate growth in sales of a company based on current economic conditions.
- You have the recent company data which indicates that the growth in sales is around two and a half times the growth in the economy.
- Using this insight, we can predict future sales of the company based on current & past information.

➤ There are multiple benefits of using regression analysis. They are as follows:

1.It indicates the **significant relationships** between dependent variable and independent variable.

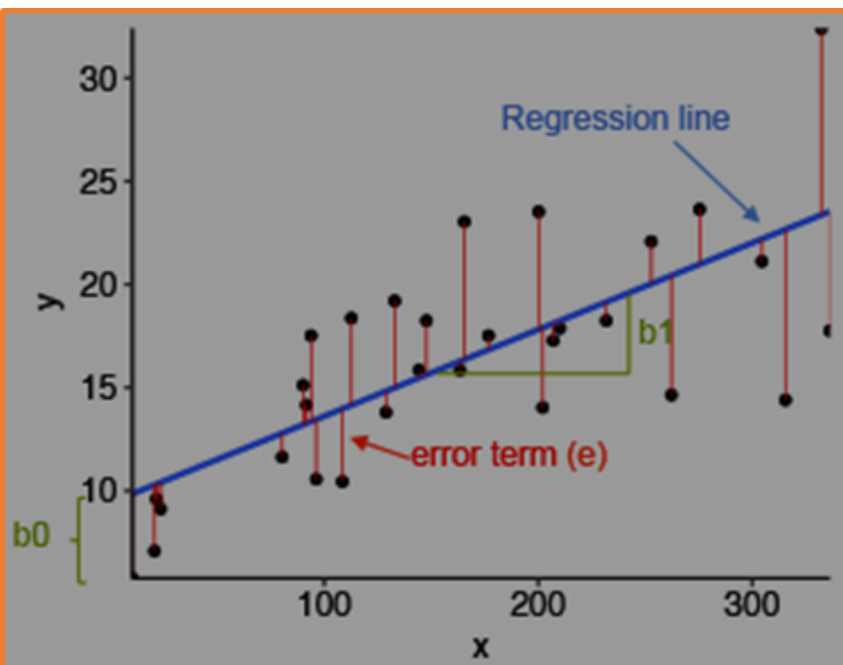
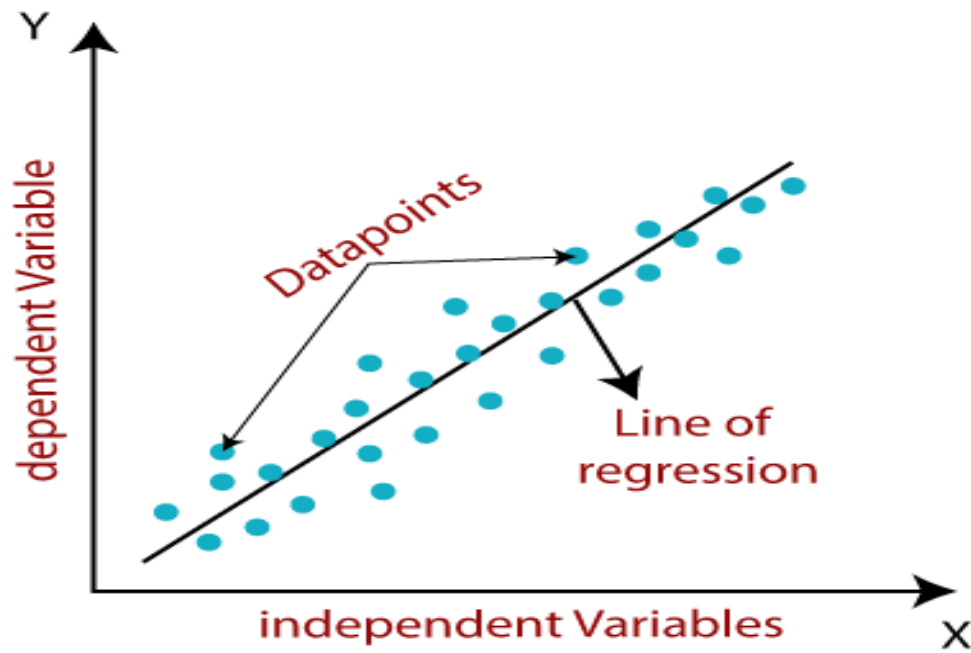
2.It indicates the **strength of impact** of multiple independent variables on a dependent variable.

3.Regression analysis also allows us to compare the effects of variables measured on different scales, such as the effect of price changes and the number of promotional activities.

➤ These benefits help market researchers / data analysts / data scientists to eliminate and evaluate the best set of variables to be used for building predictive models.

Linear Regression

- **Linear Regression** is a machine learning algorithm based on **supervised learning**.
- It performs a **regression task**.
- Regression models a target prediction value based on independent variables.
- It is mostly used for finding out the relationship between variables and forecasting.
- Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used.
- Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x).
- So, this regression technique finds out a linear relationship between x (features) and y (response).
- Hence, the name is Linear Regression



Estimated (or predicted) y value

Estimate of the regression intercept

Estimate of the regression slope

Independent variable

Error term

$$y_i = b_0 + b_1 x + e$$

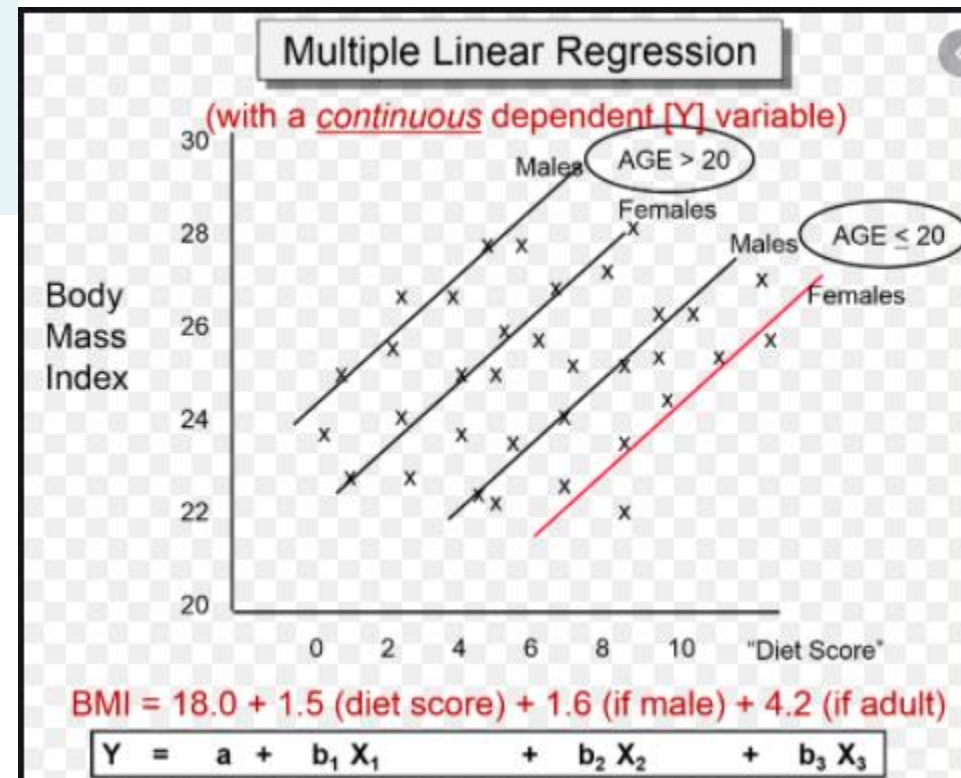
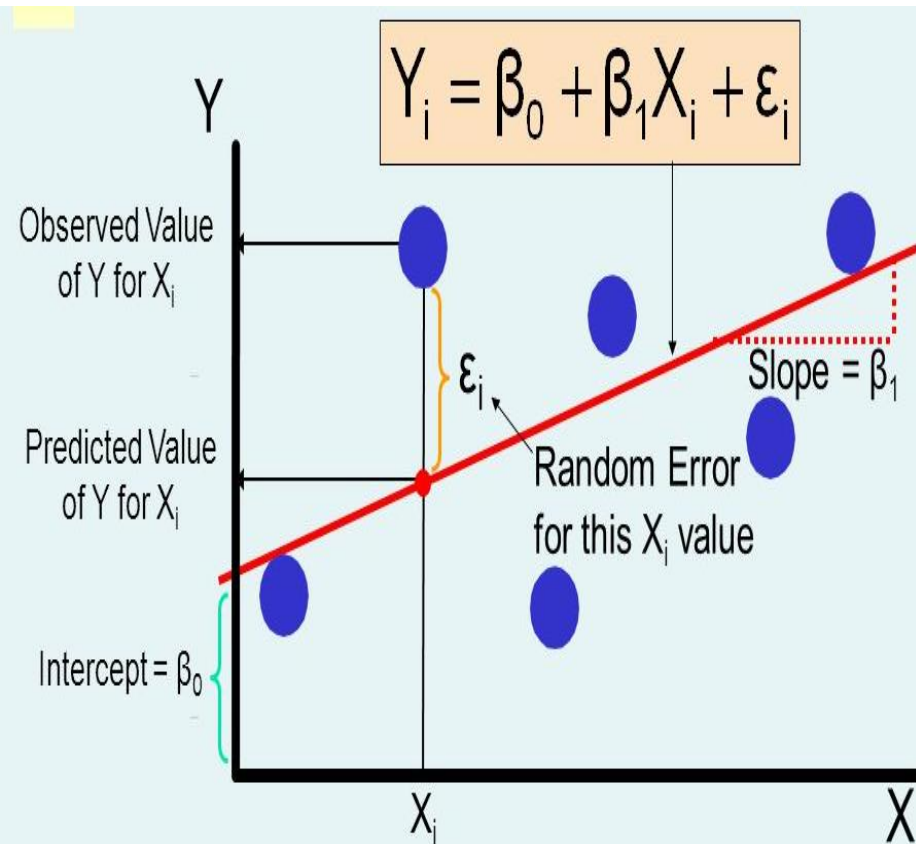
Linear Regression: Single Variable

$$\boxed{\hat{y}} = \underbrace{\beta_0 + \beta_1}_{\text{Coefficients}} \underbrace{x}_{\text{Input}} + \underbrace{\epsilon}_{\text{Error}}$$

Predicted output

Linear Regression: Multiple Variables

$$\boxed{\hat{y}} = \underbrace{\beta_0 + \beta_1 x_1}_{\text{Coefficients}} + \dots + \underbrace{\beta_p x_p}_{\text{Coefficients}} + \underbrace{\epsilon}_{\text{Error}}$$

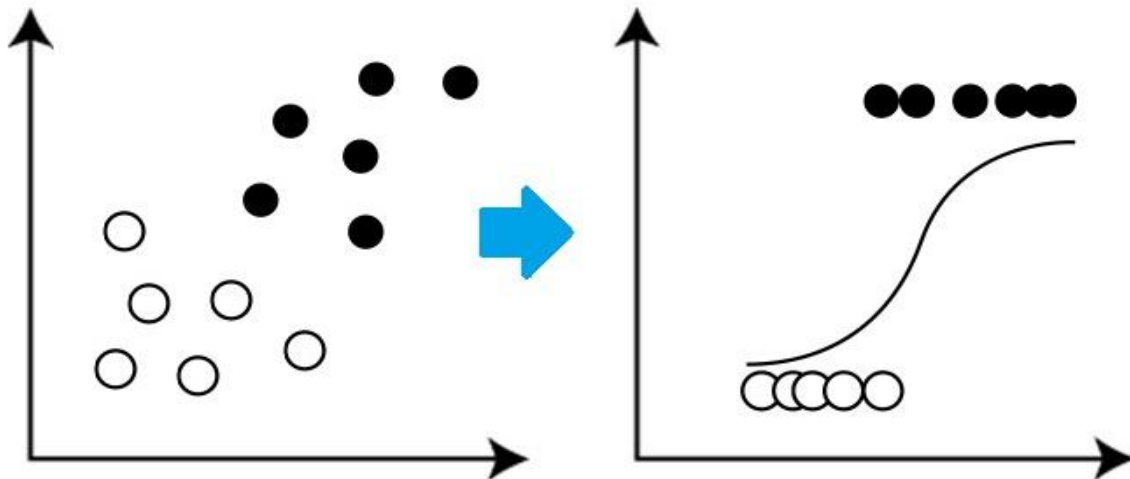




Logistic Regression

- Logistic regression is basically a supervised classification algorithm.
- In a classification problem, the target variable(or output), y , can take only discrete values for given set of features(or inputs), X .
- The model builds a regression model to predict the probability that a given data entry belongs to the category numbered as “1”.
- Logistic regression models the data using the sigmoid function.

$$g(z) = \frac{1}{1+e^{-z}}$$



Logistic regression equation :

Linear regression

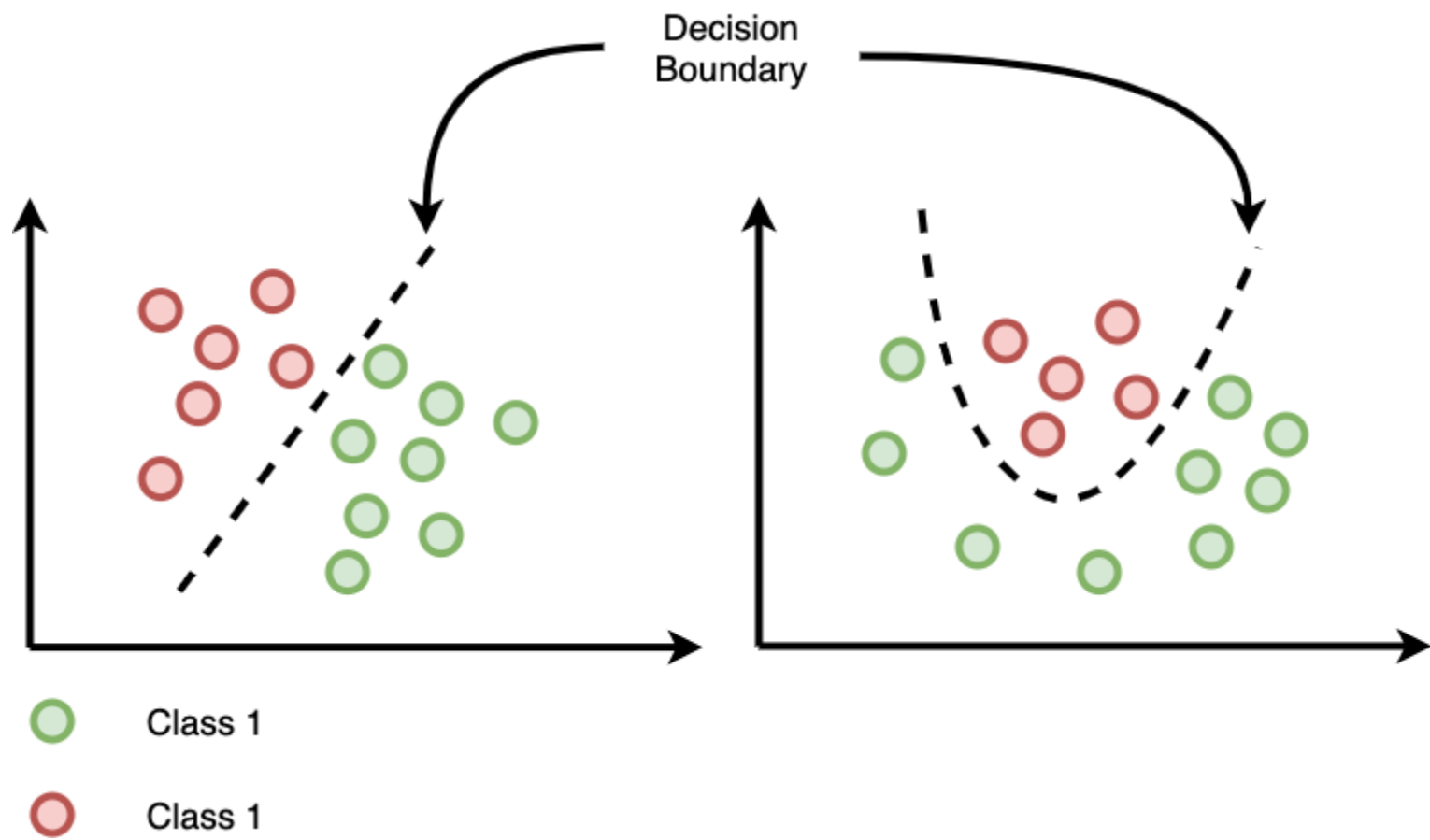
$$Y = b_0 + b_1 \times X_1 + b_2 \times X_2 + \dots + b_K \times X_K$$

Sigmoid Function

$$P = \frac{1}{1 + e^{-Y}}$$

By putting Y in Sigmoid function, we get the following result.

$$\ln \left(\frac{P}{1 - P} \right) = b_0 + b_1 \times X_1 + b_2 \times X_2 + \dots + b_K \times X_K$$

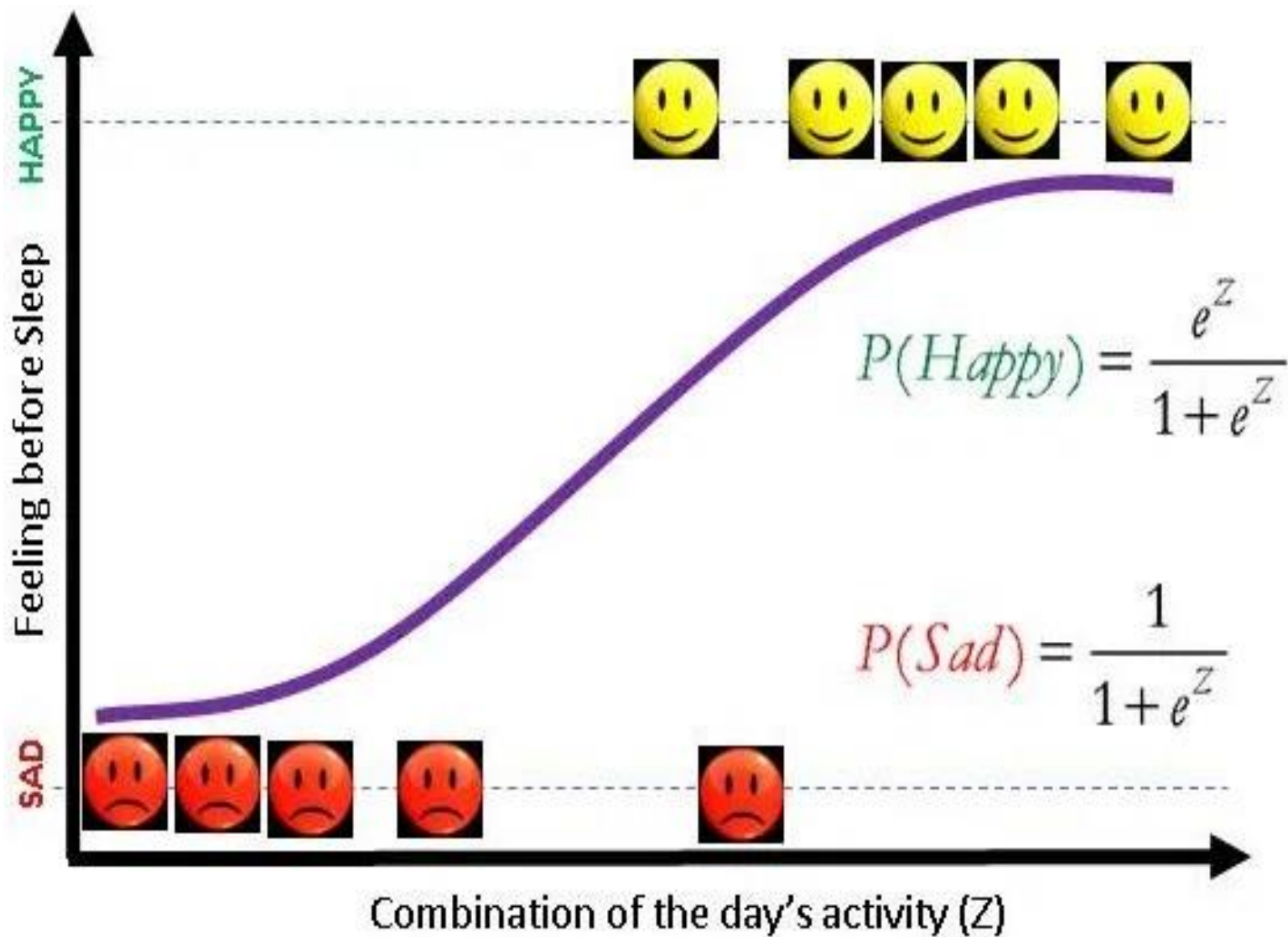


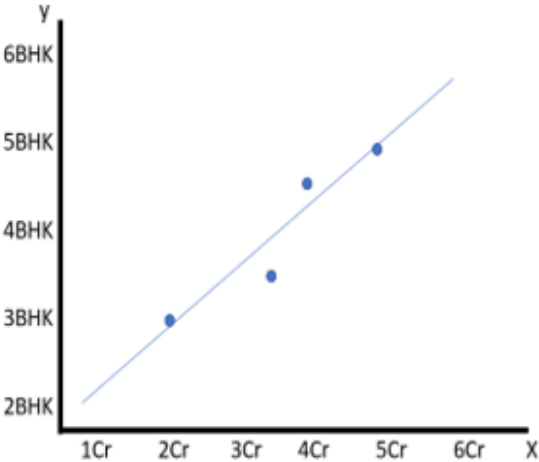
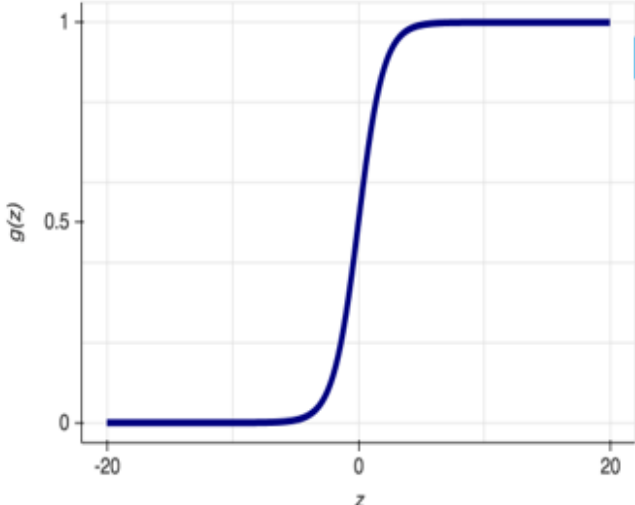
Based on the number of categories, Logistic regression can be classified as:

1.Binomial: target variable can have only 2 possible types: “0” or “1” which may represent “win” vs “loss”, “pass” vs “fail”, “dead” vs “alive”, etc.

2.Multinomial: target variable can have 3 or more possible types which are not ordered(i.e. types have no quantitative significance) like “disease A” vs “disease B” vs “disease C”.

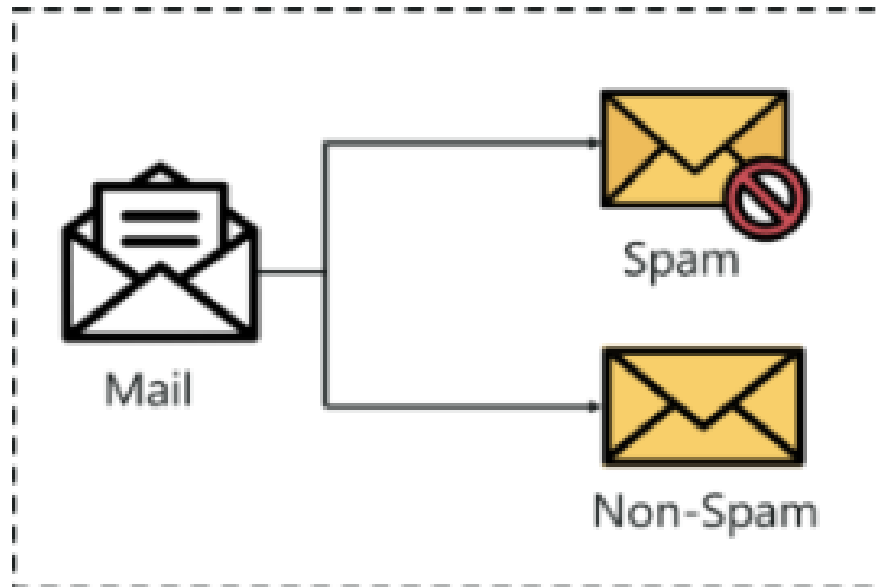
3.Ordinal: it deals with target variables with ordered categories. For example, a test score can be categorized as:“very poor”, “poor”, “good”, “very good”. Here, each category can be given a score like 0, 1, 2, 3.



Linear Regression	Logistic Regression
Target is an interval variable	Target is discrete (binary or ordinal) variable
Predicted values are the mean of the target variable at the given values of the input variable	Predicted values are the probability of the particular levels of the given values of the input variable
Solve regression problems	Solve classification problems
Example : What is the Temperature?	Example : Will it rain or not?
Graph is straight line	Graph is S-curve
 <p>A scatter plot illustrating Linear Regression. The horizontal axis (X) represents the number of bedrooms in Cr (1Cr to 6Cr), and the vertical axis (Y) represents the number of bedrooms in BHK (2BHK to 6BHK). Five data points are plotted, showing a positive linear trend. A straight blue line of best fit is drawn through the points, indicating that as the number of bedrooms in Cr increases, the number of bedrooms in BHK also increases linearly.</p>	 <p>A graph illustrating the S-curve (sigmoid function) used in Logistic Regression. The horizontal axis is labeled z and ranges from -20 to 20. The vertical axis is labeled $g(z)$ and ranges from 0 to 1. The curve is an S-shape, starting near 0 for negative z, passing through 0.5 at $z=0$, and approaching 1 for positive z. This function maps any real-valued number into the range (0, 1), which can be interpreted as probabilities.</p>

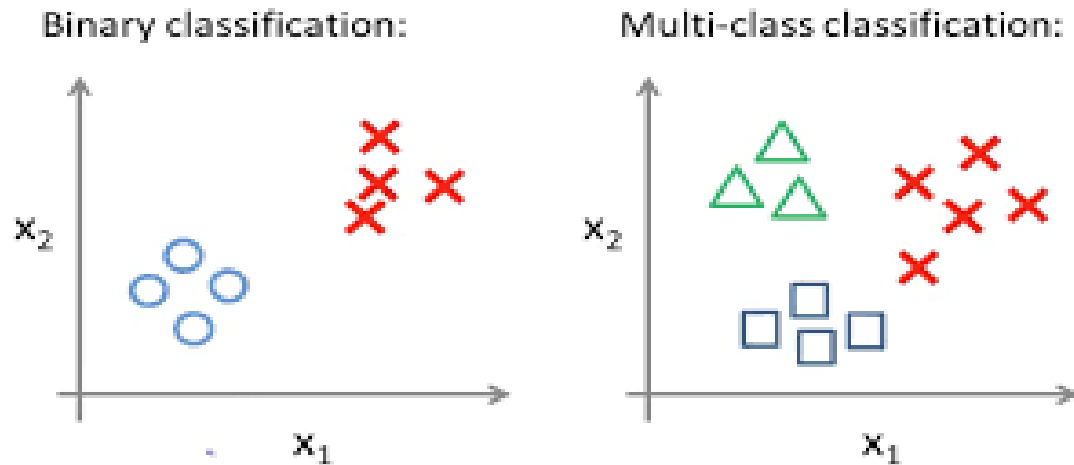
CLASSIFICATION

- Classification is a process of categorizing a given set of data into classes
- It can be performed on both structured or unstructured data
- The process starts with predicting the class of given data points
- The classes are often referred to as target, label or categories.
- The classification predictive modeling is the task of approximating the mapping function from input variables to discrete output variables
- The main goal is to identify which class/category the new data will fall into.



Example:

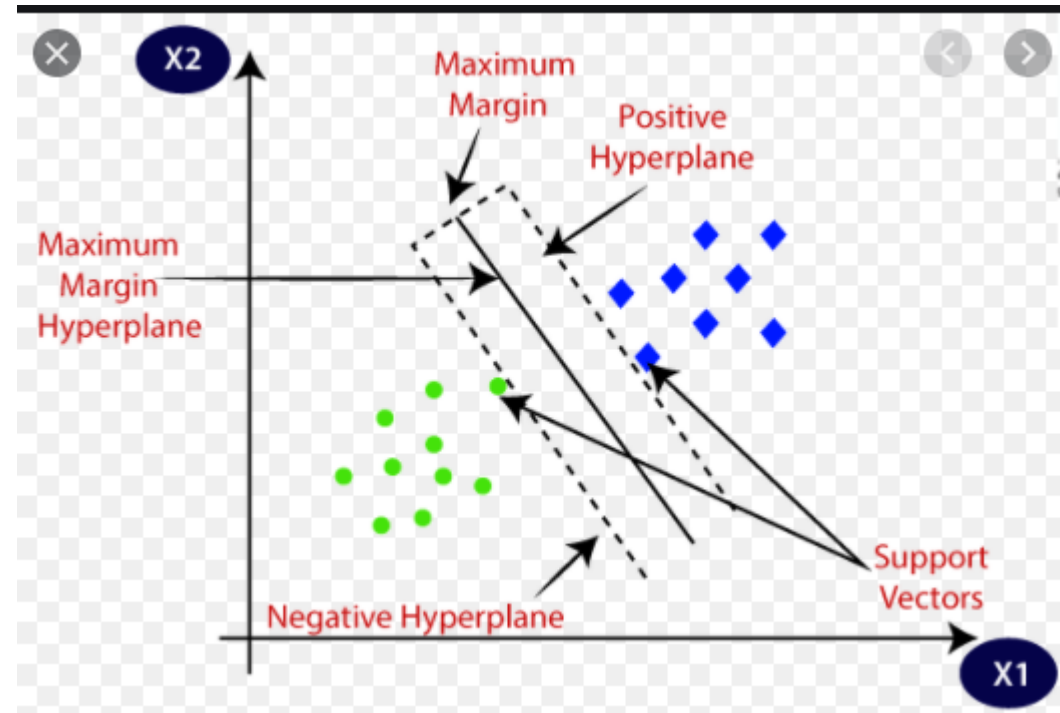
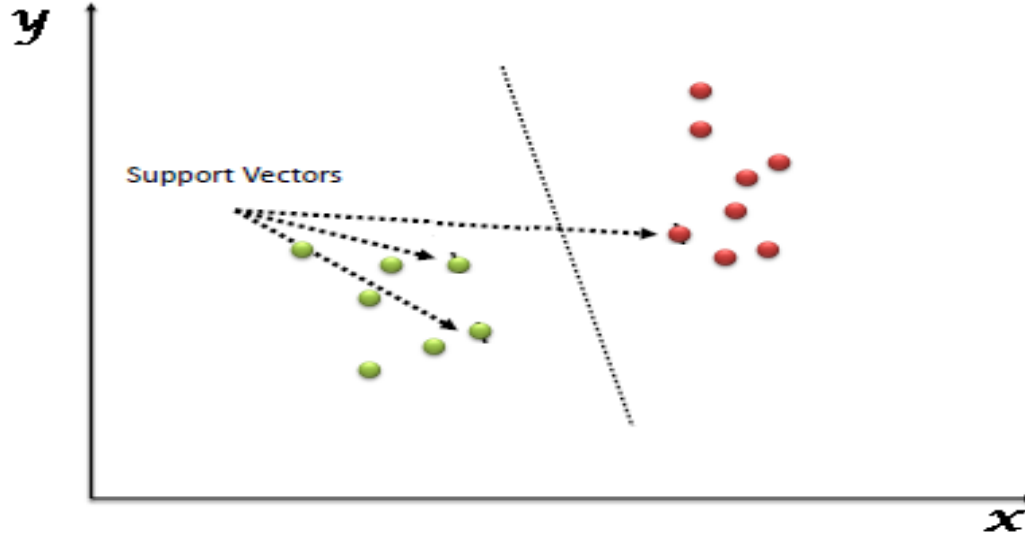
- Heart disease detection can be identified as a classification problem, this is a binary classification since there can be only two classes i.e has heart disease or does not have heart disease.
- The classifier, in this case, needs training data to understand how the given input variables are related to the class.
- And once the classifier is trained accurately, it can be used to detect whether heart disease is there or not for a particular patient.



Difference between Regression and Classification

Regression Algorithm	Classification Algorithm
In Regression, the output variable must be of continuous nature or real value.	In Classification, the output variable must be a discrete value.
The task of the regression algorithm is to map the input value (x) with the continuous output variable(y).	The task of the classification algorithm is to map the input value(x) with the discrete output variable(y).
Regression Algorithms are used with continuous data.	Classification Algorithms are used with discrete data.
In Regression, we try to find the best fit line, which can predict the output more accurately.	In Classification, we try to find the decision boundary, which can divide the dataset into different classes.
Regression algorithms can be used to solve the regression problems such as Weather Prediction, House price prediction, etc.	Classification Algorithms can be used to solve classification problems such as Identification of spam emails, Speech Recognition, Identification of cancer cells, etc.
The regression Algorithm can be further divided into Linear and Non-linear Regression.	The Classification algorithms can be divided into Binary Classifier and Multi-class Classifier.

SUPPORT VECTOR MACHINE (SVM)



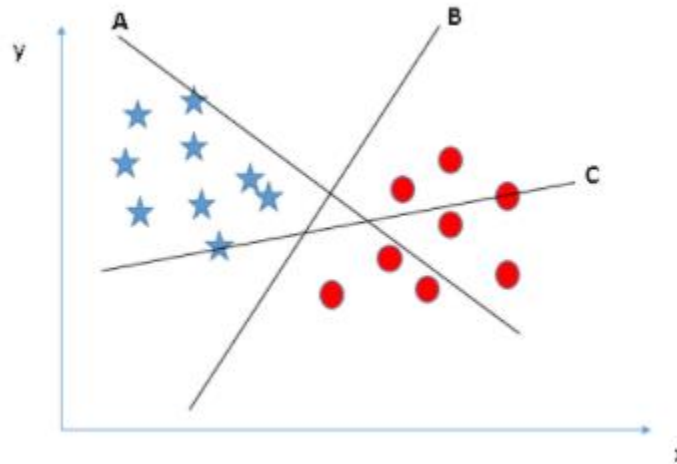
- Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges.
- However, it is mostly used in classification problems.
- In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate.
- Then, we perform classification by finding the hyper-plane that differentiates the two classes very well
- Support Vectors are simply the co-ordinates of individual observation. The SVM classifier is a frontier which best segregates the two classes (hyper-plane/ line).

How does it work?

- It is the process of segregating the two classes with a hyper-plane.
- Now the burning question is “How can we identify the right hyper-plane?”.
Don't worry, it's not as hard as you think!

Let's understand:

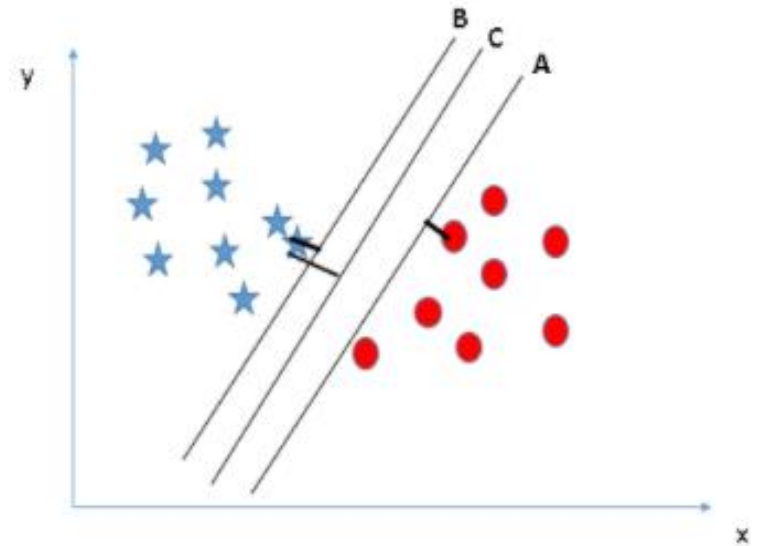
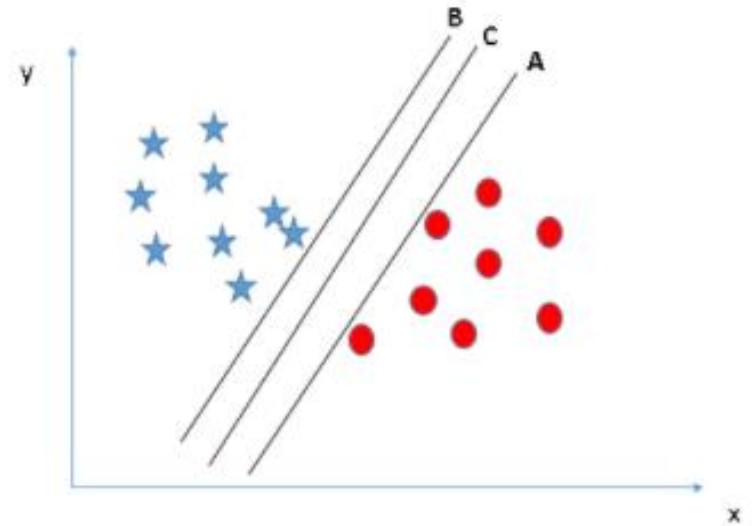
Identify the right hyper-plane (Scenario-1): Here, we have three hyper-planes (A, B and C). Now, identify the right hyper-plane to classify star and circle.



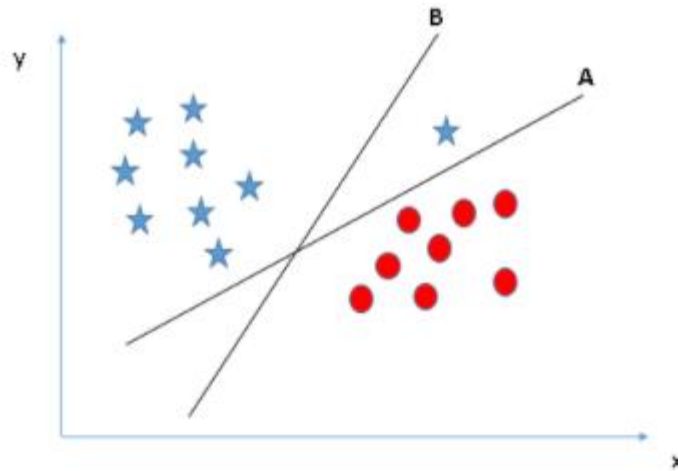
- You need to remember a thumb rule to identify the right hyper-plane: “Select the hyper-plane which segregates the two classes better”.
- In this scenario, hyper-plane “B” has excellently performed this job.

Identify the right hyper-plane (Scenario-2): Here, we have three hyper-planes (A, B and C) and all are segregating the classes well. Now, How can we identify the right hyper-plane?

- Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as **Margin**.
- Here you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.



•Identify the right hyper-plane (Scenario-3):Hint: Use the rules as discussed in previous section to identify the right hyper-plane

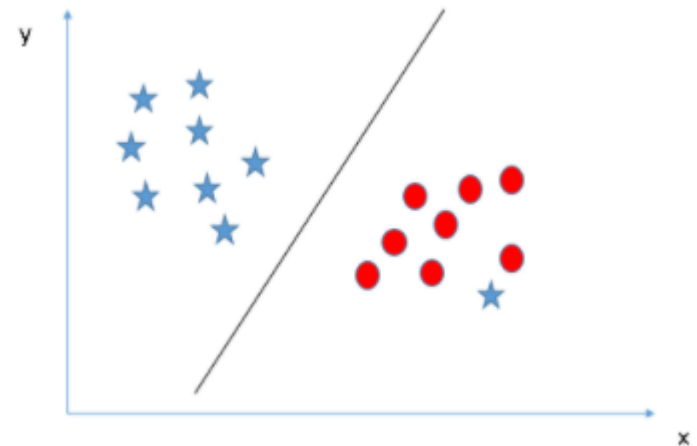


- Some of you may have selected the hyper-plane **B** as it has higher margin compared to **A**.
- But, here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin.
- Here, hyper-plane B has a classification error and A has classified all correctly.
- Therefore, the right hyper-plane is **A**.

Can we classify two classes (Scenario-4)?: Below, I am unable to segregate the two classes using a straight line, as one of the stars lies in the territory of other(circle) class as an outlier.



As I have already mentioned, one star at other end is like an outlier for star class. The SVM algorithm has a feature to ignore outliers and find the hyper-plane that has the maximum margin. Hence, we can say, SVM classification is robust to outliers.

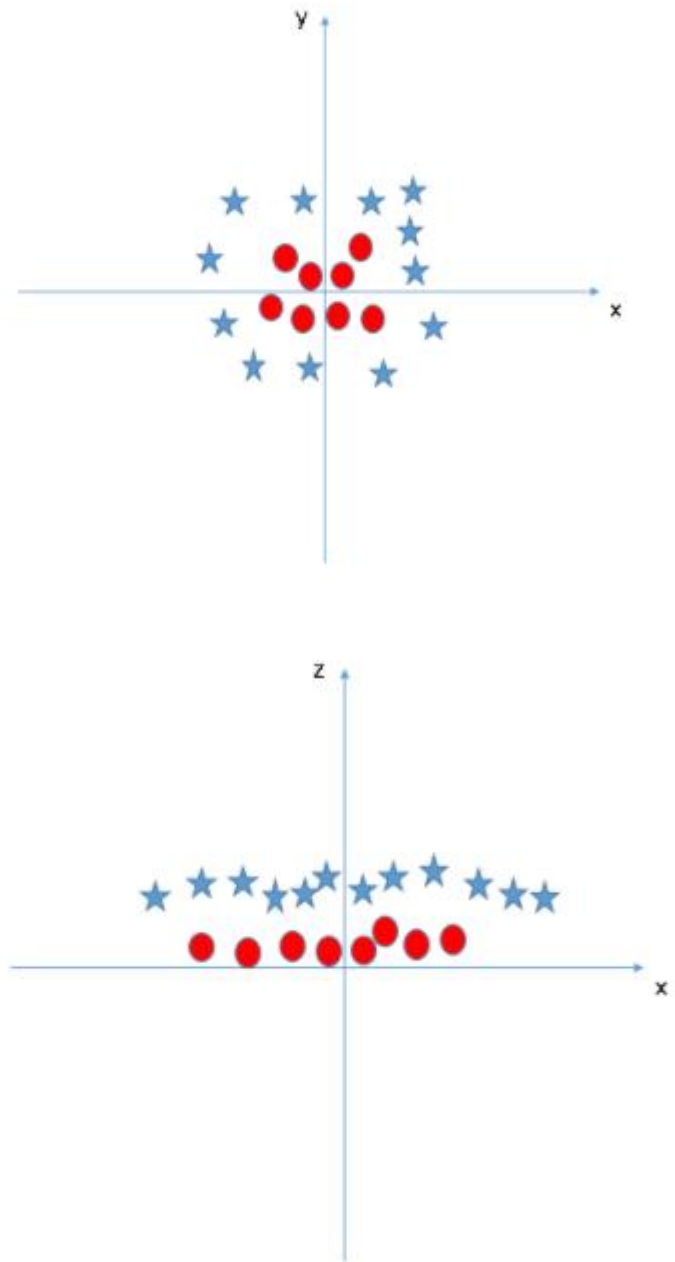


Find the hyper-plane to segregate to classes (Scenario-5): In the scenario below, we can't have linear hyper-plane between the two classes, so how does SVM classify these two classes? Till now, we have only looked at the linear hyper-plane.

- SVM can solve this problem. Easily! It solves this problem by introducing additional feature. Here, we will add a new feature $z=x^2+y^2$. Now, let's plot the data points on axis x and z:

In above plot, points to consider are:

- All values for z would be positive always because z is the squared sum of both x and y
- In the original plot, red circles appear close to the origin of x and y axes, leading to lower value of z and star relatively away from the origin result to higher value of z.



Linear SVM Mathematically

The linearly separable case

- Assume that all data is at least distance 1 from the hyperplane, then the following two constraints follow for a training set $\{(\mathbf{x}_i, y_i)\}$

$$\mathbf{w}^T \mathbf{x}_i + b \geq 1 \quad \text{if } y_i = 1$$

$$\mathbf{w}^T \mathbf{x}_i + b \leq -1 \quad \text{if } y_i = -1$$

- For support vectors, the inequality becomes an equality
- Then, since each example's distance from the hyperplane is

$$r = y \frac{\mathbf{w}^T \mathbf{x} + b}{\|\mathbf{w}\|}$$

- The margin is:

$$r = \frac{2}{\|\mathbf{w}\|}$$

Linear Support Vector Machine (SVM)

- **Hyperplane**

$$\mathbf{w}^T \mathbf{x} + b = 0$$

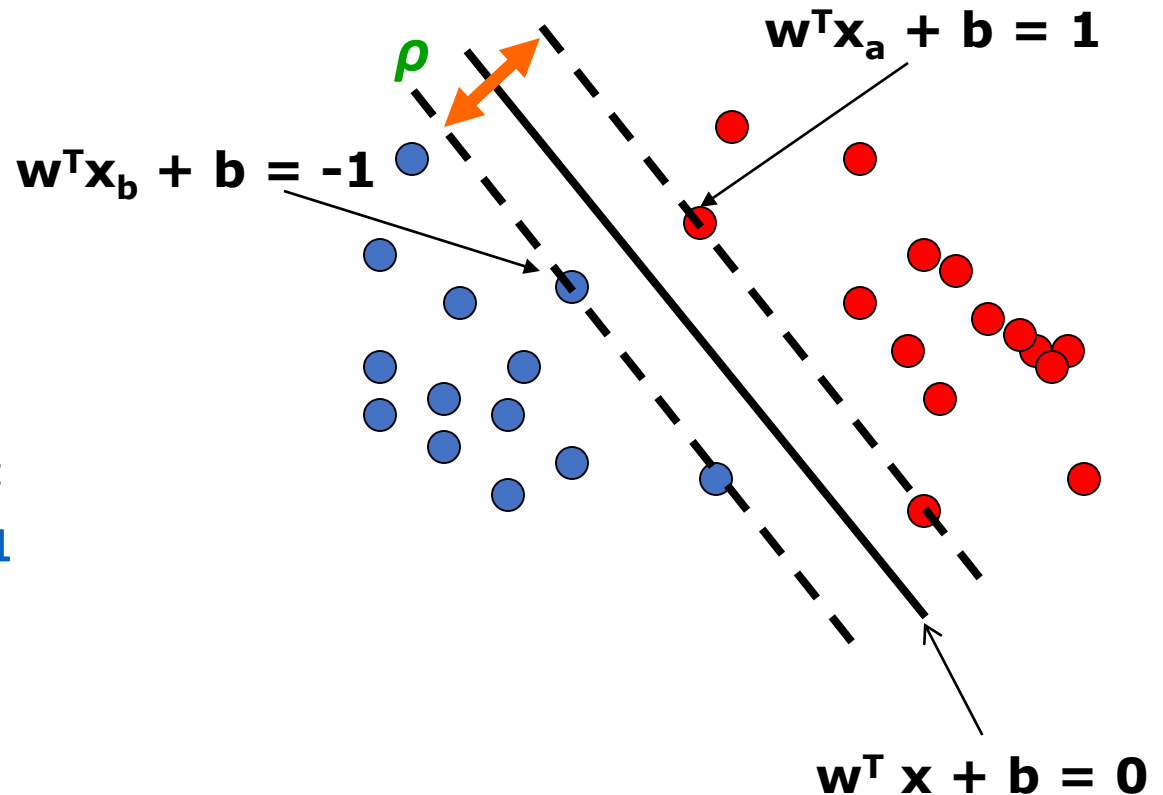
- **Extra scale constraint:**

$$\min_{i=1,\dots,n} |\mathbf{w}^T \mathbf{x}_i + b| = 1$$

- This implies:

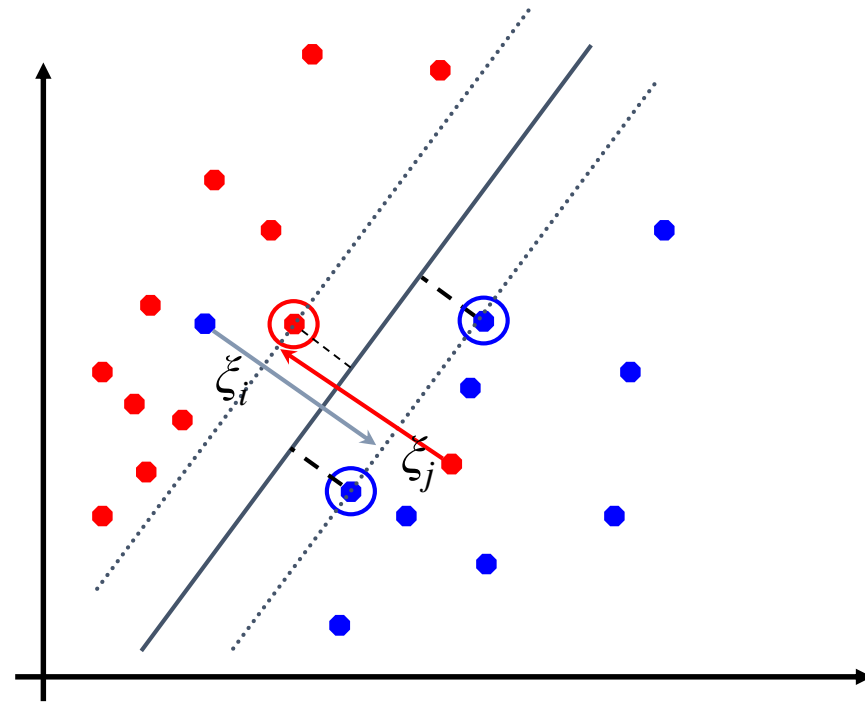
$$\mathbf{w}^T (\mathbf{x}_a - \mathbf{x}_b) = 2$$

$$\rho = \|\mathbf{x}_a - \mathbf{x}_b\|_2 = 2 / \|\mathbf{w}\|_2$$



Soft Margin Classification

- If the training data is not linearly separable, *slack variables* ξ_i can be added to allow misclassification of difficult or noisy examples.
- Allow some errors
 - Let some points be moved to where they belong, at a cost
- Still, try to minimize training set errors, and to place hyperplane “far” from each class (large margin)



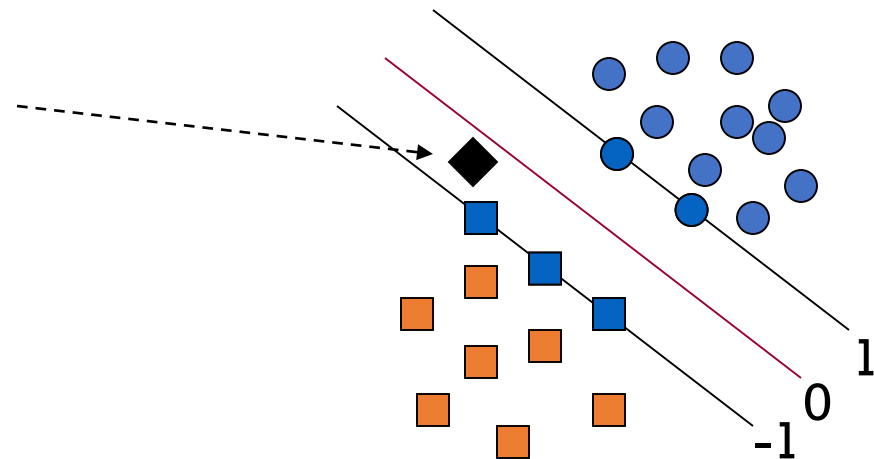
Classification with SVMs

- Given a new point \mathbf{x} , we can score its projection onto the hyperplane normal:
 - I.e., compute score: $\mathbf{w}^T \mathbf{x} + b = \sum \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b$
 - Decide class based on whether $<$ or > 0
- Can set confidence threshold t .

Score $> t$: yes

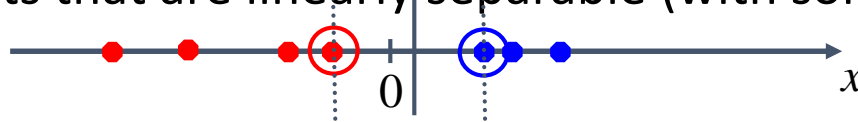
Score $< -t$: no

Else: don't know



Non-linear SVMs

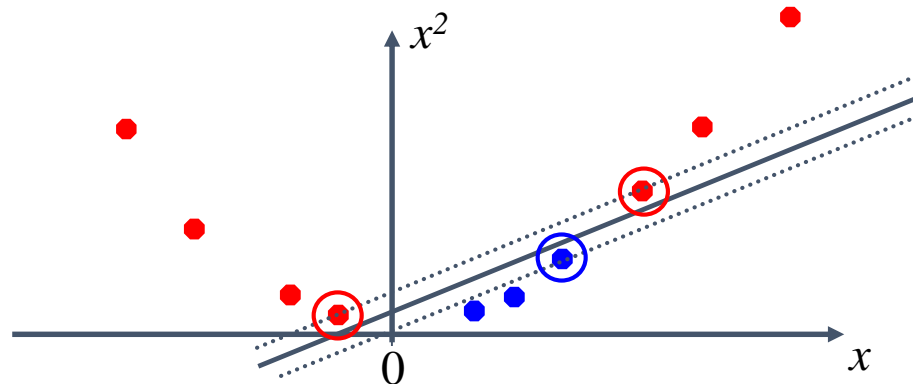
- Datasets that are linearly separable (with some noise) work out great:



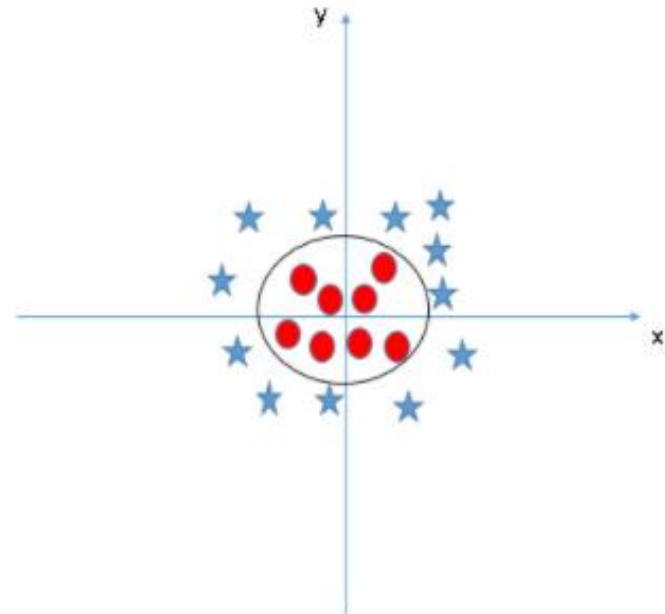
- But what are we going to do if the dataset is just too hard?



- How about ... mapping data to a higher-dimensional space:

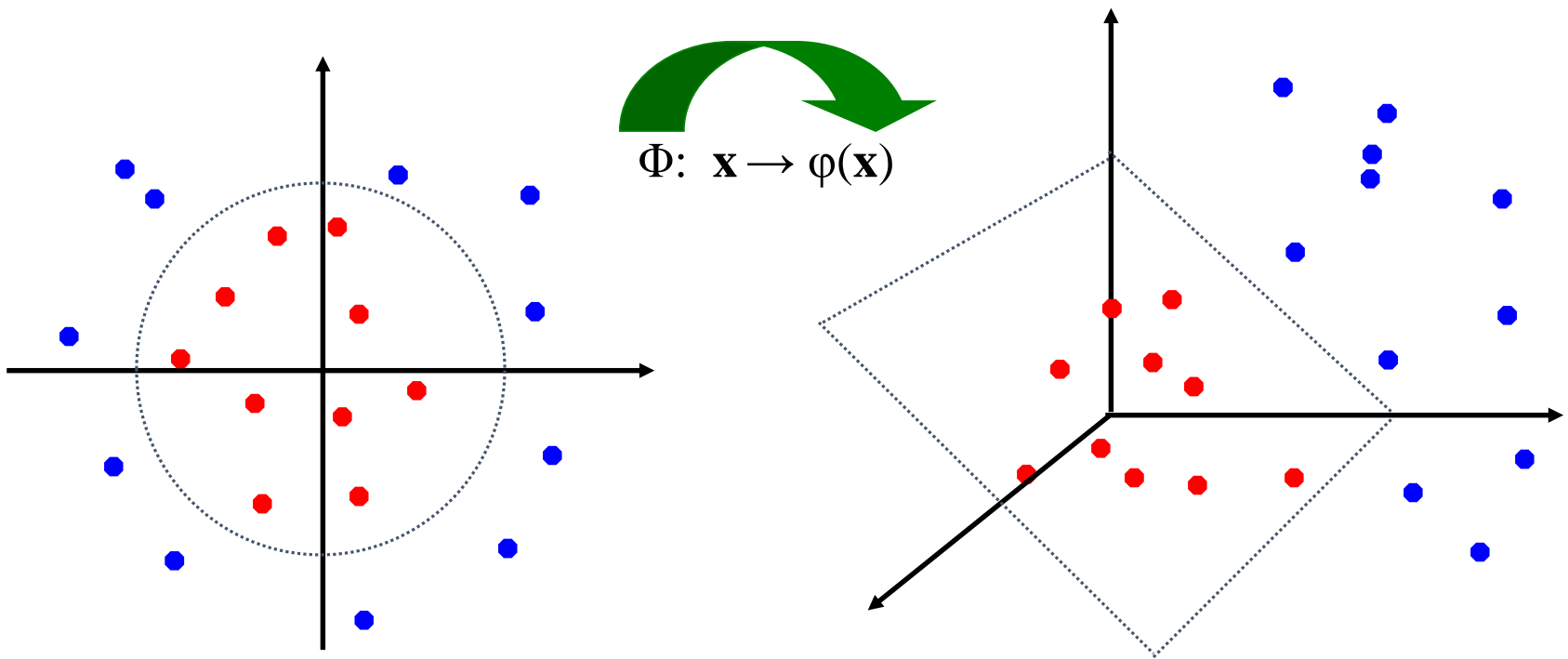


- In the SVM classifier, it is easy to have a linear hyper-plane between these two classes.
- But, another burning question which arises is, should we need to add this feature manually to have a hyper-plane.
- No, the SVM algorithm has a technique called the **kernel trick**.
- The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space i.e. it converts not separable problem to separable problem.
- It is mostly useful in non-linear separation problem.
- Simply put, it does some extremely complex data transformations, then finds out the process to separate the data based on the labels or outputs you've defined.
- When we look at the hyper-plane in original input space it looks like a circle:



Non-linear SVMs: Feature spaces

- General idea: the original feature space can always be mapped to some higher-dimensional feature space where the training set is separable:



Naive Bayes

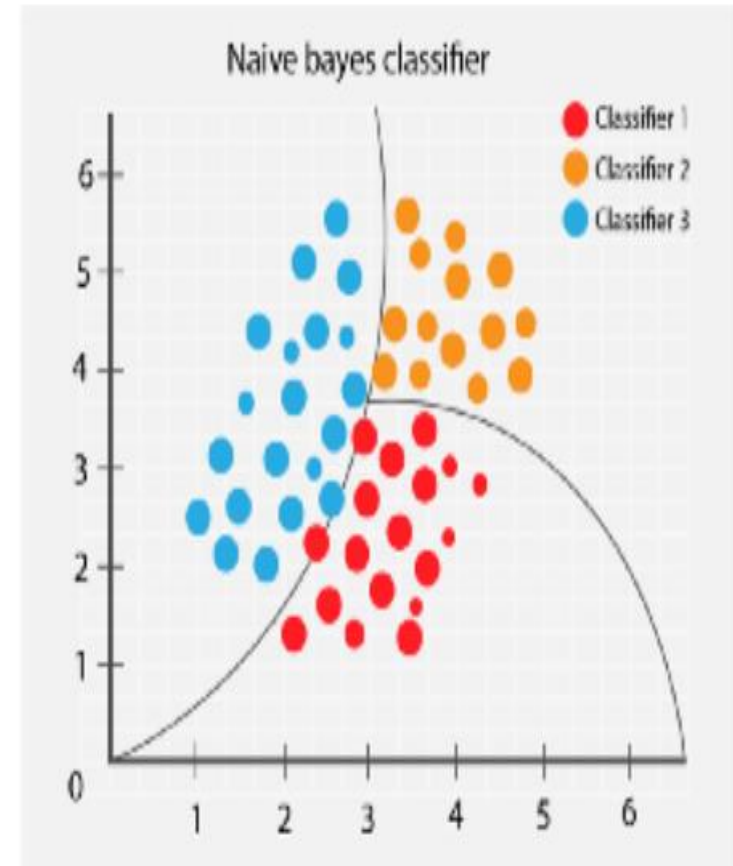
- Naive Bayes is among one of the most simple and powerful algorithms for **classification** based on Bayes' Theorem with an assumption of independence among predictors.
- Naive Bayes model is easy to build and particularly useful for very large data sets.
- There are two parts to this algorithm:
 - Naive
 - Bayes
- The Naive Bayes classifier assumes that the presence of a feature in a class is unrelated to any other feature.
- Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that a particular fruit is an apple or an orange or a banana and that is why it is known as “Naive”.

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

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

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

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



- Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred.
- Basically, we are trying to find probability of event A, given the event B is true. Event B is also termed as **evidence**.
- P(A) is the **priori** of A (the prior probability, i.e. Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance(here, it is event B).
- P(A|B) is a posteriori probability of B, i.e. probability of event after evidence is seen.

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	No
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	Yes
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	High	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

Just to clear, an example of a feature vector and corresponding class variable can be: (refer 1st row of dataset)

```
X = (Rainy, Hot, High, False)
y = No
```

with regards to our dataset, we can apply Bayes' theorem in following way:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

where, y is class variable and X is a dependent feature vector (of size n) where:

$$X = (x_1, x_2, x_3, \dots, x_n)$$

- So basically, $P(y|X)$ here means, the probability of “Not playing golf” given that the weather conditions are “Rainy outlook”, “Temperature is hot”, “high humidity” and “no wind”.
- Now, its time to put a naive assumption to the Bayes' theorem, which is, **independence** among the features. So now, we split **evidence** into the independent parts.
- Now, if any two events A and B are independent, then,

$$P(A,B) = P(A)P(B)$$

Hence, we reach to the result:

$$P(y|x_1, \dots, x_n) = \frac{P(x_1|y)P(x_2|y)\dots P(x_n|y)P(y)}{P(x_1)P(x_2)\dots P(x_n)}$$

which can be expressed as:

$$P(y|x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i|y)}{P(x_1)P(x_2)\dots P(x_n)}$$

Now, as the denominator remains constant for a given input, we can remove that term:

$$P(y|x_1, \dots, x_n) \propto P(y) \prod_{i=1}^n P(x_i|y)$$

- So, finally, we are left with the task of calculating $P(y)$ and $P(x_i | y)$.
- Please note that $P(y)$ is also called **class probability** and $P(x_i | y)$ is called **conditional probability**.
- The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of $P(x_i | y)$.
- Let us try to apply the above formula manually on our weather dataset.
- For this, we need to do some precomputations on our dataset. We need to find $P(x_i | y_j)$ for each x_i in X and y_j in y .

- All these calculations have been demonstrated in the tables below:

Outlook

	Yes	No	P(yes)	P(no)
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0/5
Rainy	3	2	3/9	2/5
Total	9	5	100%	100%

Temperature

	Yes	No	P(yes)	P(no)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
Total	9	5	100%	100%

Humidity

	Yes	No	P(yes)	P(no)
High	3	4	3/9	4/5
Normal	6	1	6/9	1/5
Total	9	5	100%	100%

Wind

	Yes	No	P(yes)	P(no)
False	6	2	6/9	2/5
True	3	3	3/9	3/5
Total	9	5	100%	100%

Play		P(Yes)/P(No)
Yes	9	9/14
No	5	5/14
Total	14	100%

- So, in the figure above, we have calculated $P(x_i | y_j)$ for each x_i in X and y_j in y manually in the tables 1-4.
- For example, probability of playing golf given that the temperature is cool, i.e $P(\text{temp.} = \text{cool} | \text{play golf} = \text{Yes}) = 3/9$.
- Also, we need to find class probabilities ($P(y)$) which has been calculated in the table 5. For example, $P(\text{play golf} = \text{Yes}) = 9/14$.
- So now, we are done with our pre-computations and the classifier is ready!

Let us test it on a new set of features (let us call it today):

```
today = (Sunny, Hot, Normal, False)
```

So, probability of playing golf is given by:

$$P(\text{Yes}|\text{today}) = \frac{P(\text{SunnyOutlook}|\text{Yes})P(\text{HotTemperature}|\text{Yes})P(\text{NormalHumidity}|\text{Yes})P(\text{NoWind}|\text{Yes})P(\text{Yes})}{P(\text{today})}$$

and probability to not play golf is given by:

$$P(\text{No}|\text{today}) = \frac{P(\text{SunnyOutlook}|\text{No})P(\text{HotTemperature}|\text{No})P(\text{NormalHumidity}|\text{No})P(\text{NoWind}|\text{No})P(\text{No})}{P(\text{today})}$$

Since, $P(\text{today})$ is common in both probabilities, we can ignore $P(\text{today})$ and find proportional probabilities as:

$$P(\text{Yes}|\text{today}) \propto \frac{2}{9} \cdot \frac{2}{9} \cdot \frac{6}{9} \cdot \frac{6}{9} \cdot \frac{9}{14} \approx 0.0141$$

and

$$P(\text{No}|\text{today}) \propto \frac{3}{5} \cdot \frac{2}{5} \cdot \frac{1}{5} \cdot \frac{2}{5} \cdot \frac{5}{14} \approx 0.0068$$

Now, since

$$P(\text{Yes}|\text{today}) + P(\text{No}|\text{today}) = 1$$

These numbers can be converted into a probability by making the sum equal to 1 (normalization):

$$P(\text{Yes}|\text{today}) = \frac{0.0141}{0.0141+0.0068} = 0.67$$

and

$$P(\text{No}|\text{today}) = \frac{0.0068}{0.0141+0.0068} = 0.33$$

Since

$$P(\text{Yes}|\text{today}) > P(\text{No}|\text{today})$$

So, prediction that golf would be played is 'Yes'.

- The method that we discussed above is applicable for discrete data.
- In case of continuous data, we need to make some assumptions regarding the distribution of values of each feature.
- The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of $P(x_i | y)$.

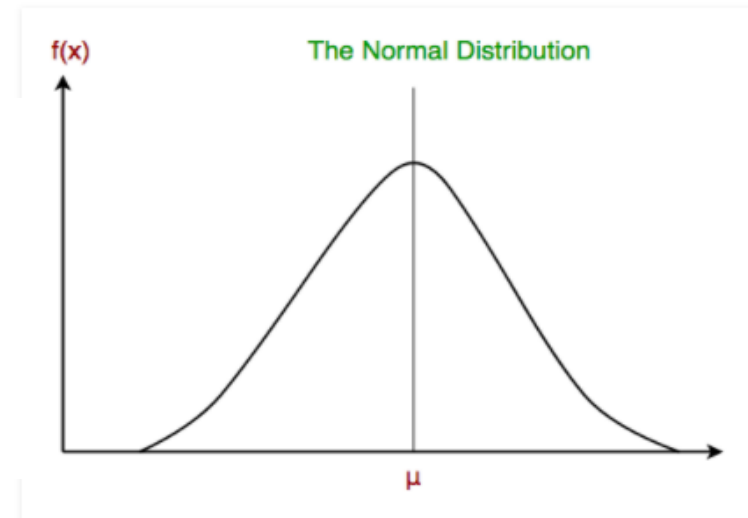
Now, we discuss one of such classifiers here.

Gaussian Naive Bayes classifier

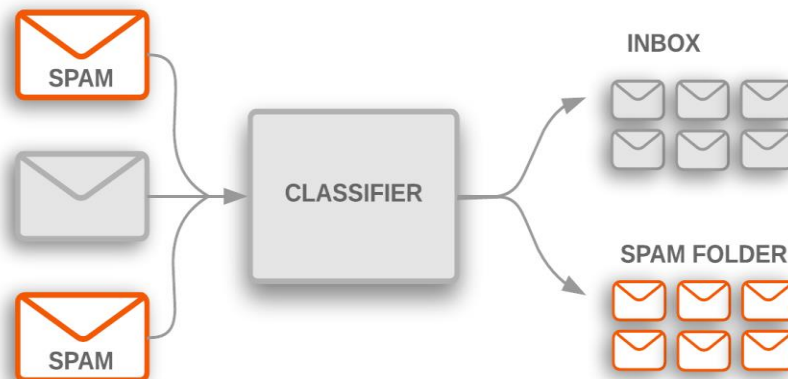
- In Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a **Gaussian distribution**.
- A Gaussian distribution is also called [Normal distribution](#).
- When plotted, it gives a bell shaped curve which is symmetric about the mean of the feature values as shown below:

The likelihood of the features is assumed to be Gaussian, hence, conditional probability is given by:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$



- The Naïve Bayes Classifier belongs to the family of probability classifier, using Bayesian theorem.
- The reason why it is called 'Naïve' because it requires rigid independence assumption between input variables.
- Therefore, it is more proper to call Simple Bayes or Independence Bayes.
- This algorithm has been studied extensively since 1960s.
- Simple though it is, Naïve Bayes Classifier remains one of popular methods to solve text categorization problem, the problem of judging documents as belonging to one category or the other, such as email spam detection.



THANK YOU

for listening...