

MACHINE LEARNING

SVM

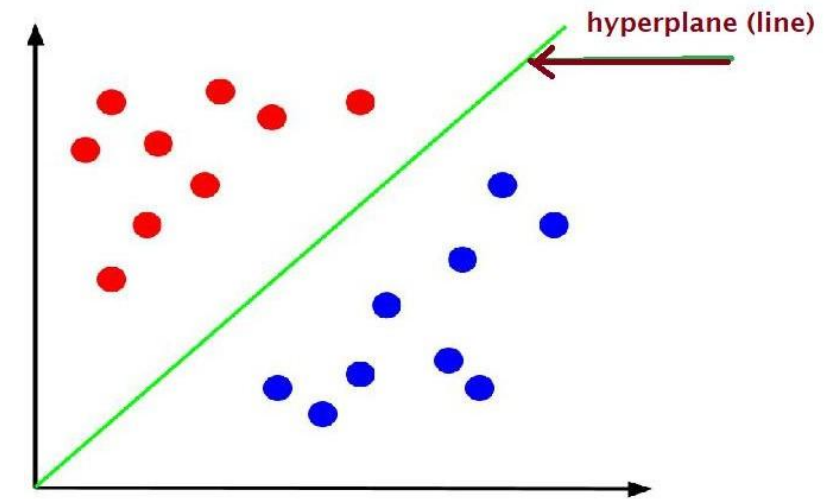
(Support Vector Machine)

Trainer : Sujata Mohite
Email: sujata.mohite@sunbeaminfo.com



Overview

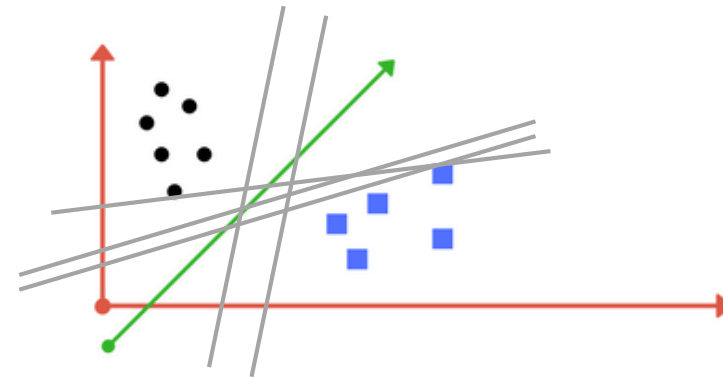
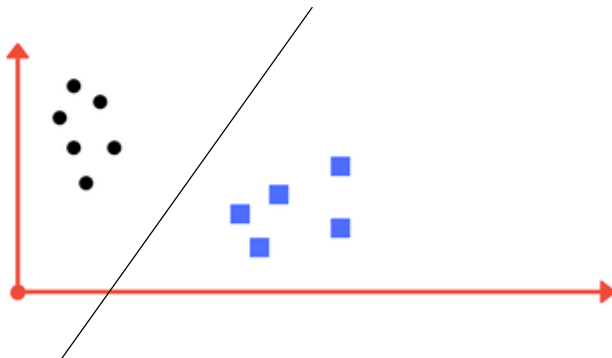
- It is a supervised machine learning algorithm that can be used for both **classification** and **regression**
- However, it is mostly used in **classification problems**
- The objective of the support vector machine algorithm is to find a **hyperplane** in an N-dimensional space (N — the number of features) that distinctly classifies the data points
- **Hyperplane**: A decision boundary separating different classes in feature space and is represented by the equation $wx + b = 0$ in linear classification.



How does it work ?

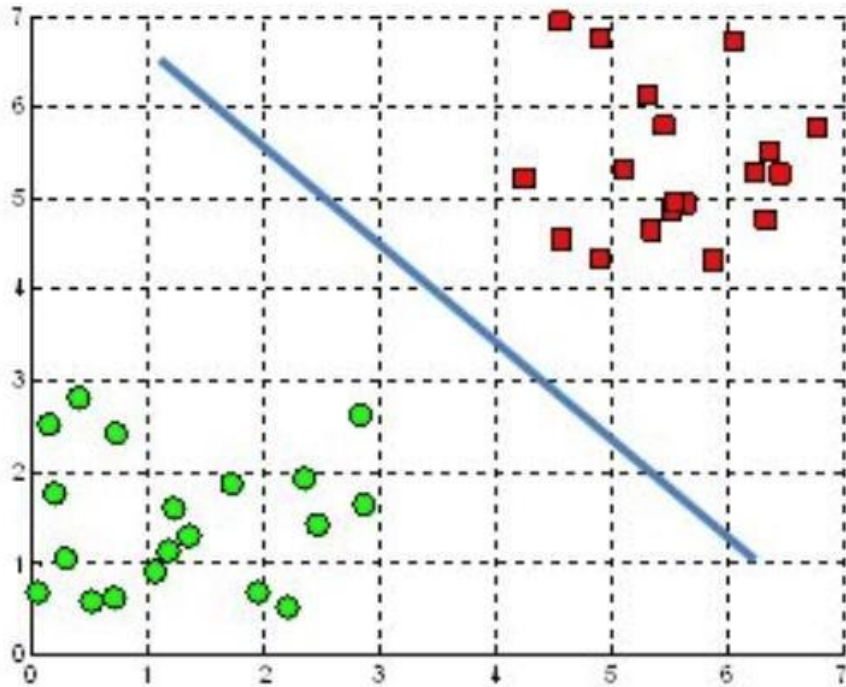
- SVM separates the classes using hyperplane
- In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side
- To separate the two classes of data points, there are many possible hyperplanes that could be chosen
- Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes

If hyperplane is chosen in this way

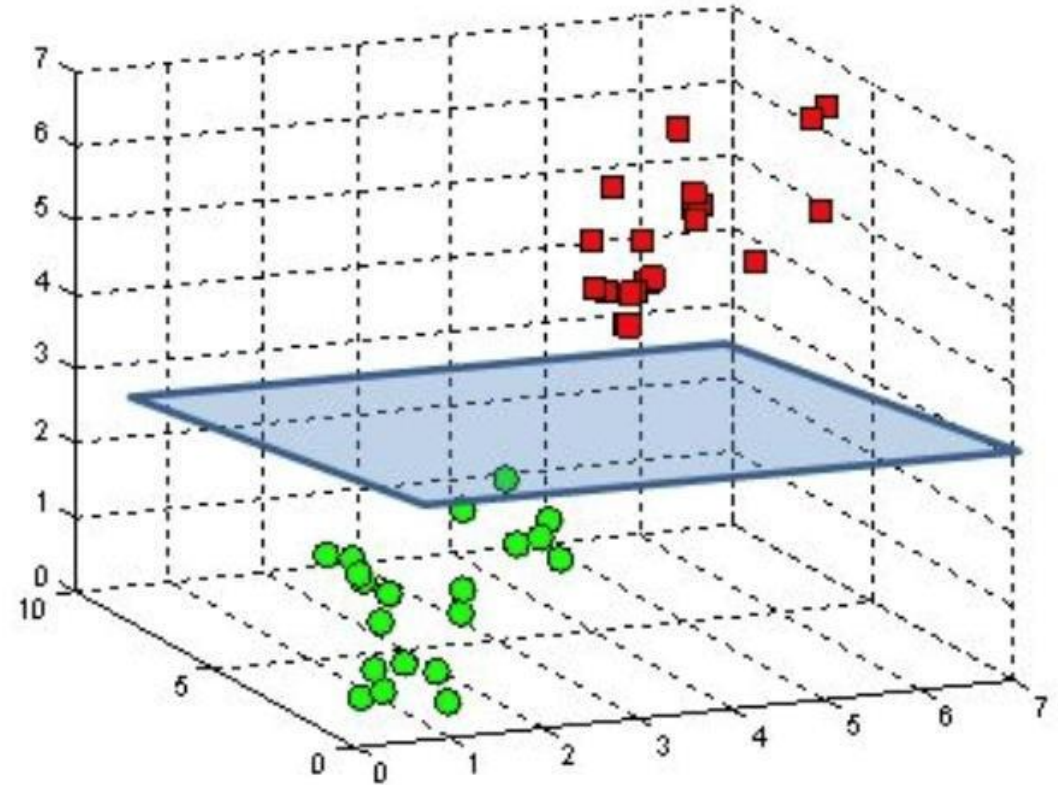


Hyperplane

A hyperplane in \mathbb{R}^2 is a line

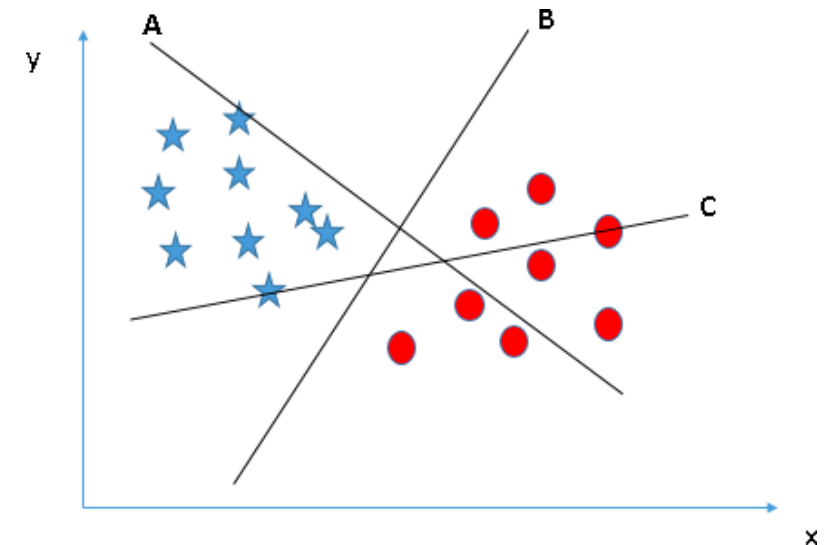


A hyperplane in \mathbb{R}^3 is a plane



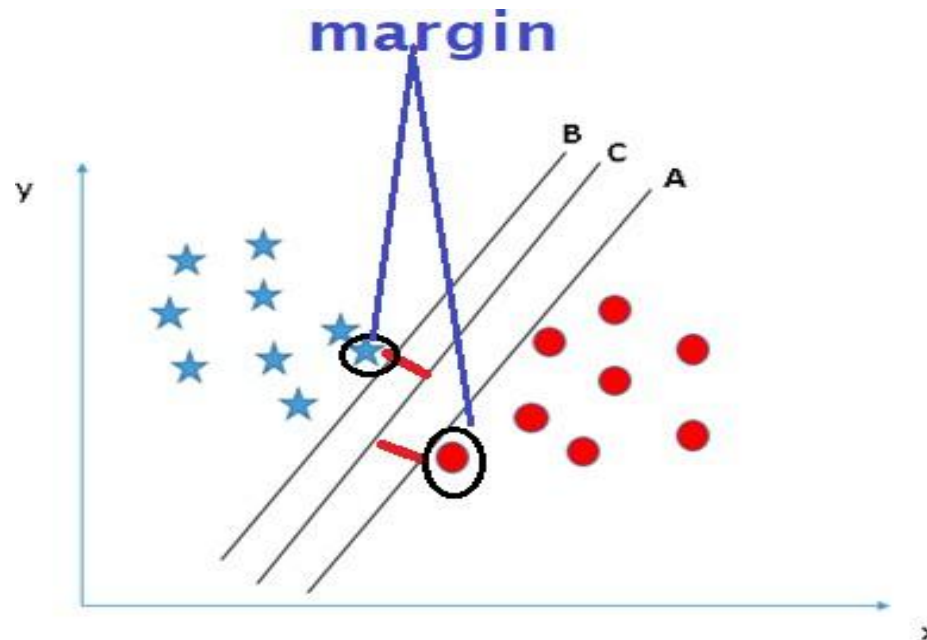
Scenario 1

- 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 (No mis-classification)**
- In this scenario, **hyper-plane “B”** has excellently performed this job



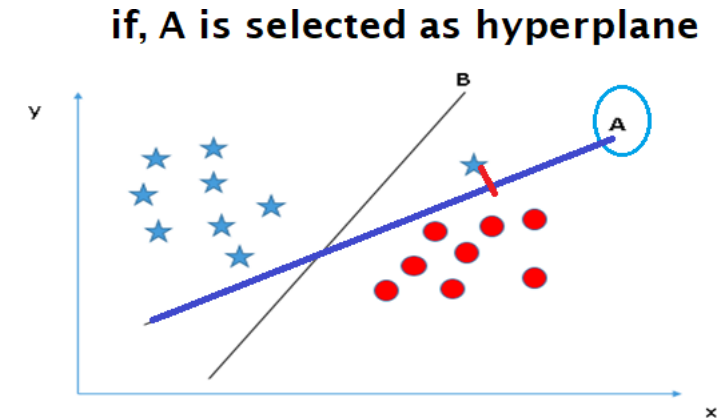
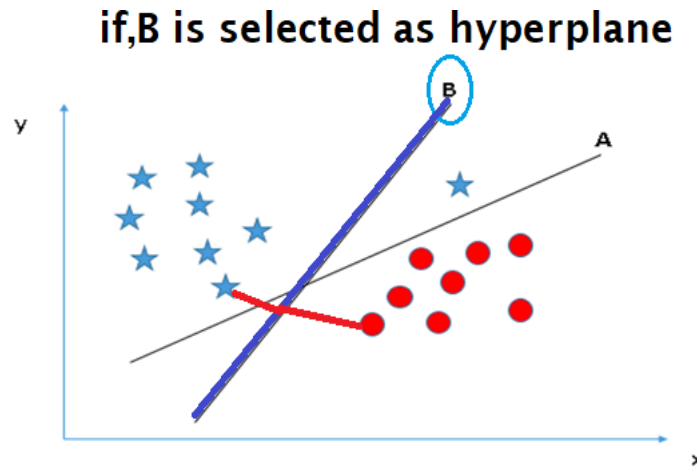
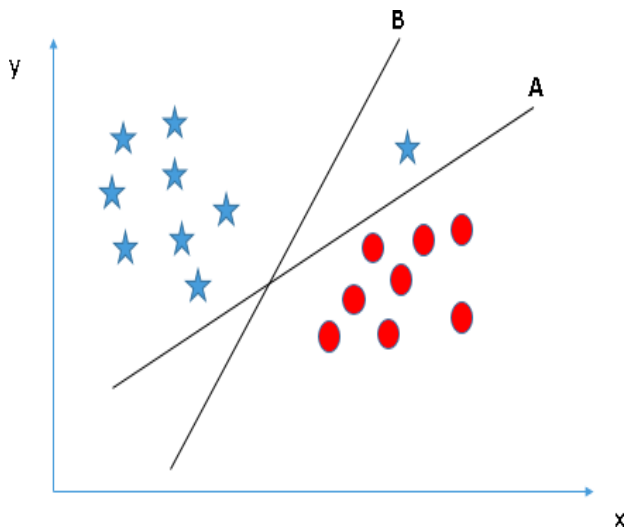
Scenario 2

- 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**.



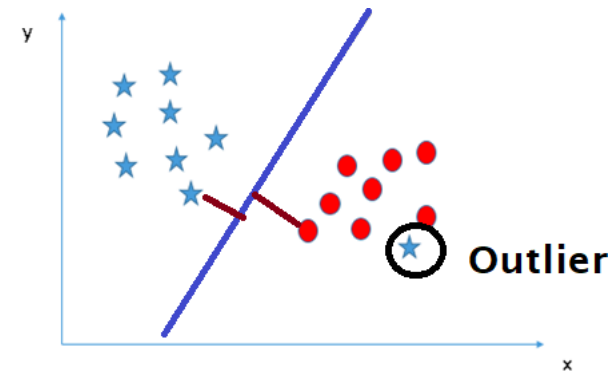
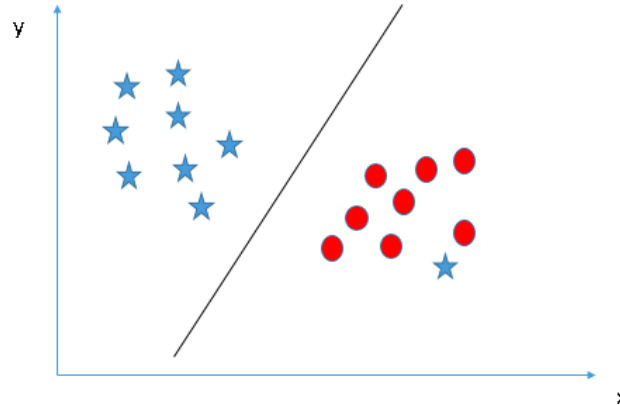
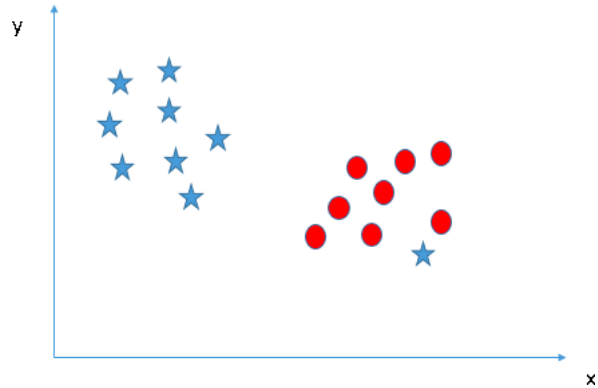
Scenario 3

- 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, 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**.



Scenario 4

- Unable to segregate the two classes using a straight line, as one of star lies in the territory of other(circle) class as an outlier
- SVM has a feature to ignore outliers and find the hyper-plane that has maximum margin.
- Hence, we can say, SVM is robust to **outliers**.

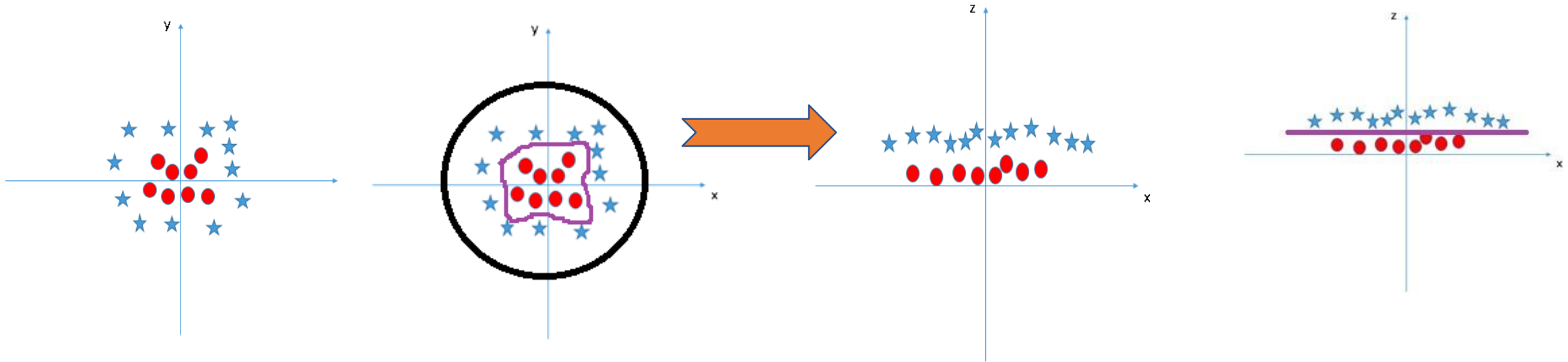


Scenario 5

- In the scenario below, we can't have linear hyper-plane between the two classes

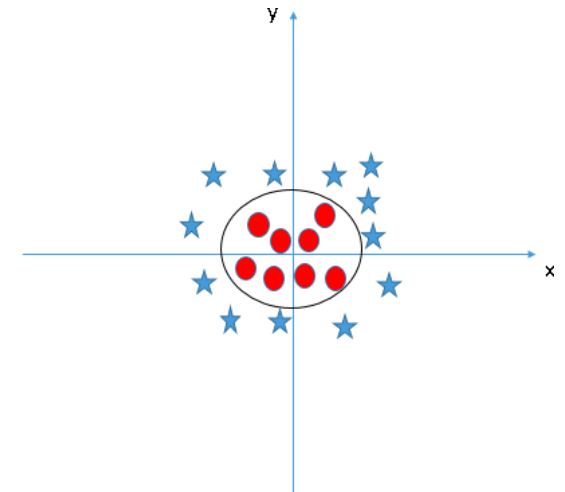
- SVM solves this problem by introducing **kernel**

$$z = x^2 + y^2$$



Scenario 5

- 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 .
- In SVM, it is easy to have a linear hyper-plane between these two classes, but for such scenarios, SVM uses a trick called as **Kernel**.
- These are functions which takes low dimensional input space and transform it to a higher dimensional space, i.e. it converts non separable problem to separable problem
- It is mostly useful in non-linear separation problem
- Simply put, it does some extremely complex data transformations, then find out the process to separate the data based on the labels or outputs you've defined.



Tuning Parameters - Kernels

- The learning of the hyperplane in linear SVM is done by transforming the problem using some linear algebra.
- This is where the kernel plays role.
- **Kernel:** A function that maps data to a higher-dimensional space enabling SVM to handle non-linearly separable data.

Hard Margin: A maximum-margin hyperplane that perfectly separates the data without misclassifications.

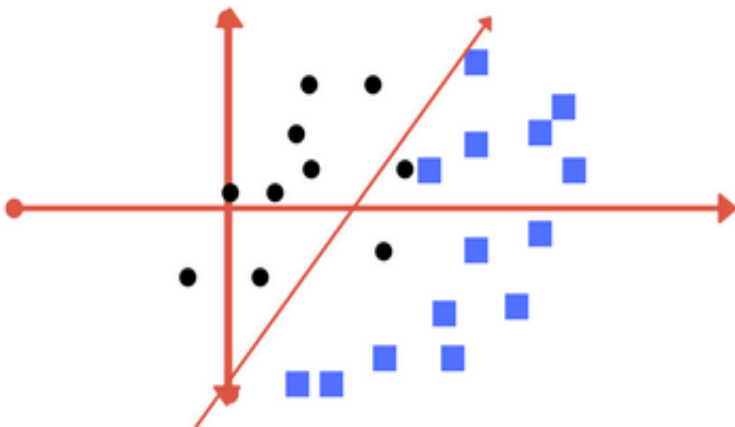
Soft Margin: Allows some misclassifications by introducing slack variables, balancing margin maximization and misclassification penalties when data is not perfectly separable.



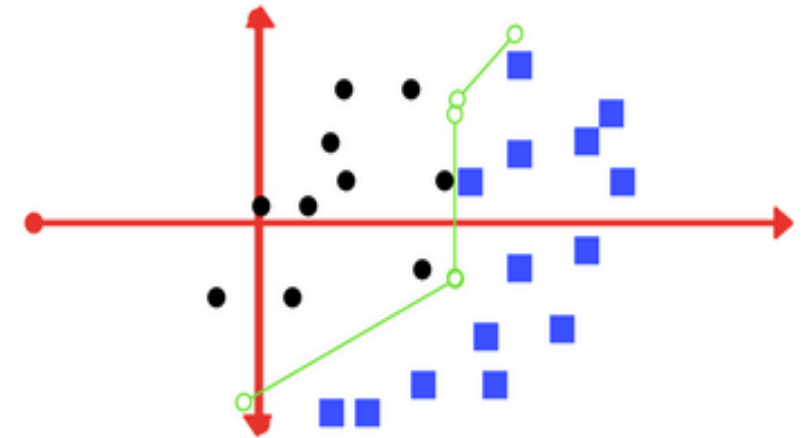
Tuning Parameters - Regularization

- The Regularization parameter tells the SVM optimization how much you want to avoid misclassifying each training example
- For large values of C , the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly
- Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points

Low regularization

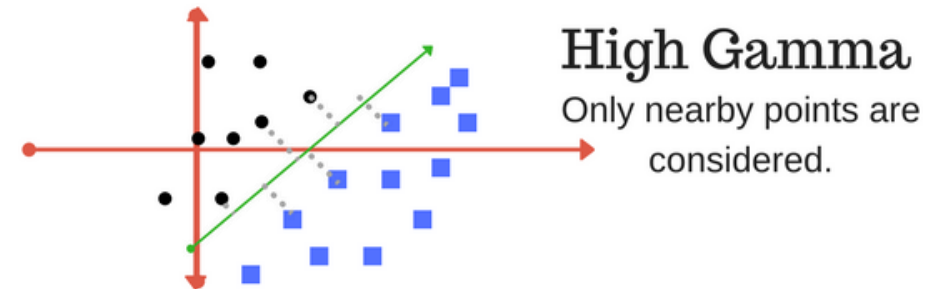
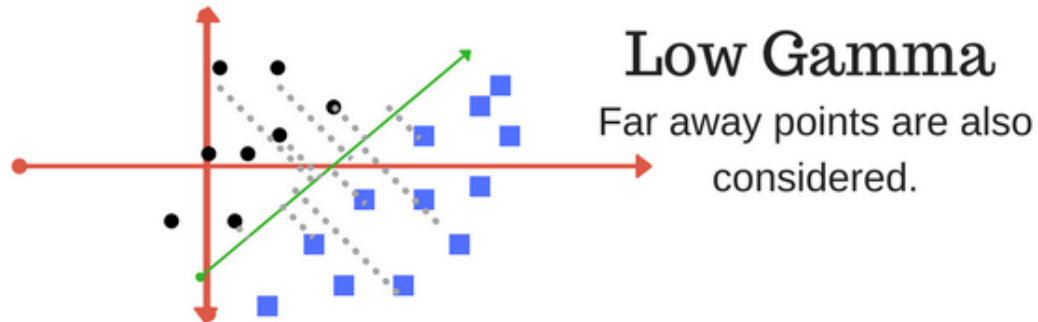


High regularization



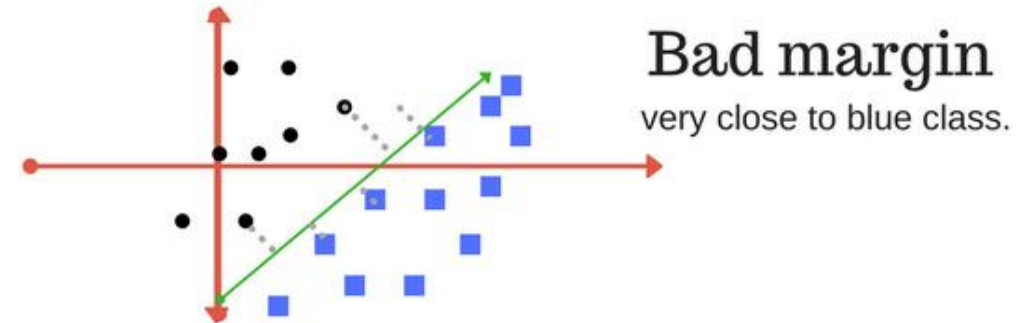
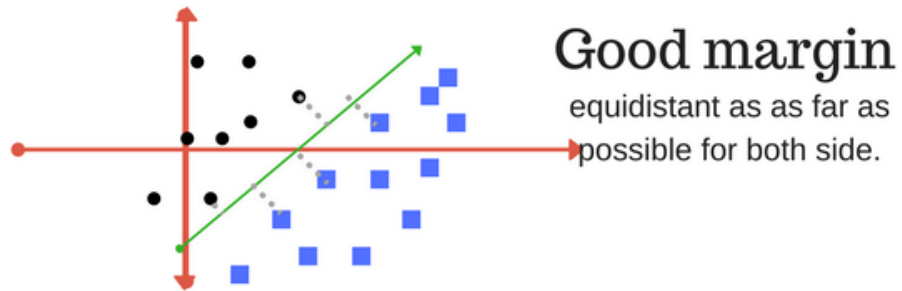
Tuning Parameters - Gamma

- The gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'
- In other words
 - With low gamma, points far away from plausible separation line are considered in calculation for the separation line
 - Whereas high gamma means the points close to plausible line are considered in calculation.



Tuning Parameters - Margin

- A margin is a separation of line to the closest class points
- A good margin is one where this separation is larger for both the classes



Thank You!!

