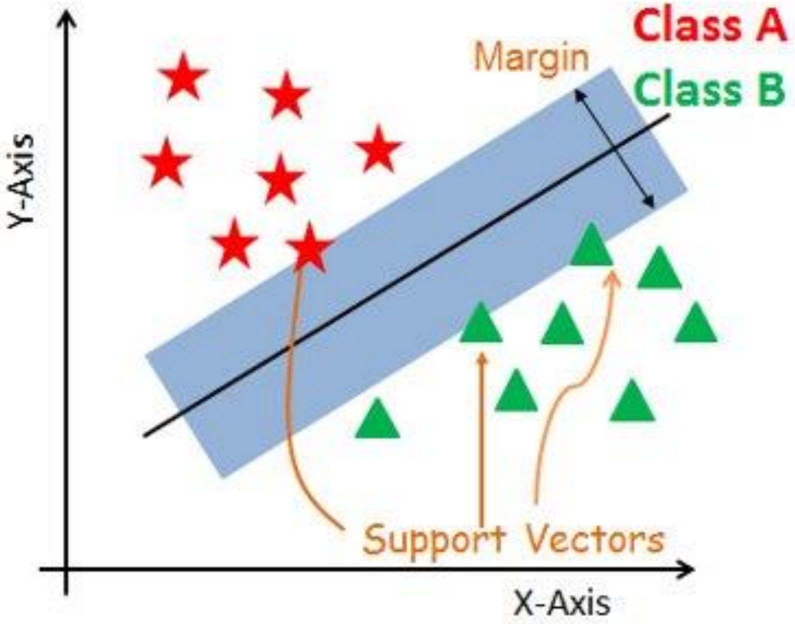
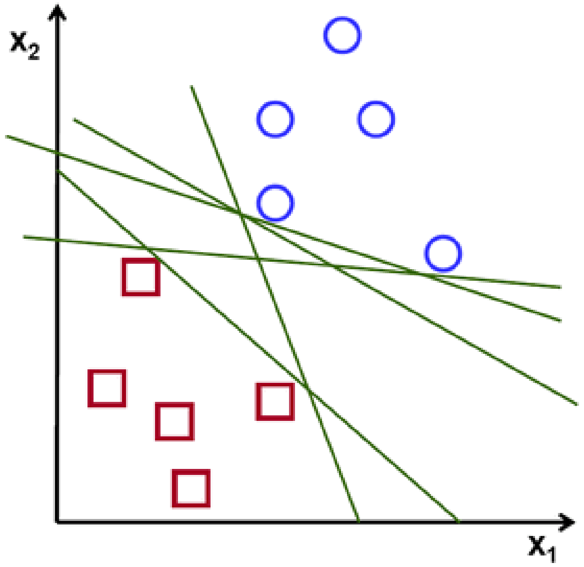
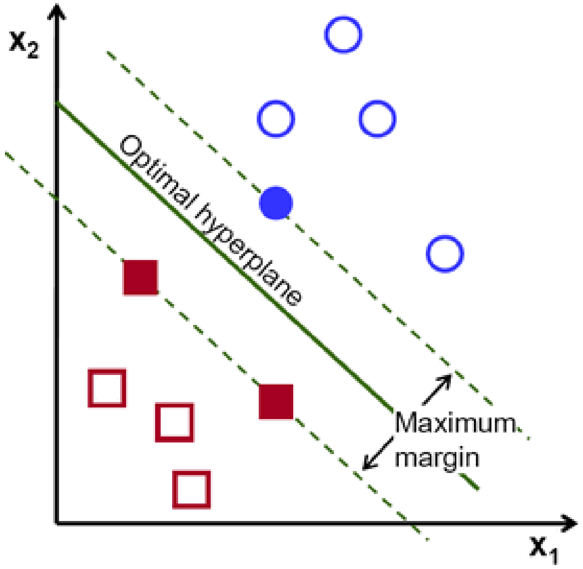


Support Vector Machines

What is Support Vector Machine?

- 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.





How does SVM work?

The main objective is to segregate the given dataset in the best possible way.

The distance between the either nearest points is known as the margin.

The objective is to select a hyperplane with the maximum possible margin between support vectors in the given dataset.

Hyperplane

A hyperplane is a decision plane which separates between a set of objects having different class memberships.

Support Vectors

- Support vectors are the data points, which are closest to the hyperplane. These points will define the separating line better by calculating margins. These points are more relevant to the construction of the classifier.

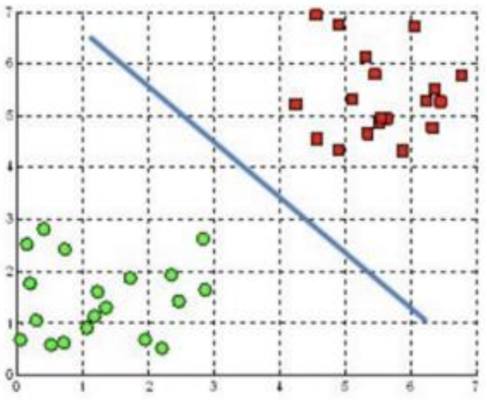
Margin

- A margin is a gap between the two lines on the closest class points. This is calculated as the perpendicular distance from the line to support vectors or closest points. If the margin is larger in between the classes, then it is considered a good margin, a smaller margin is a bad margin.

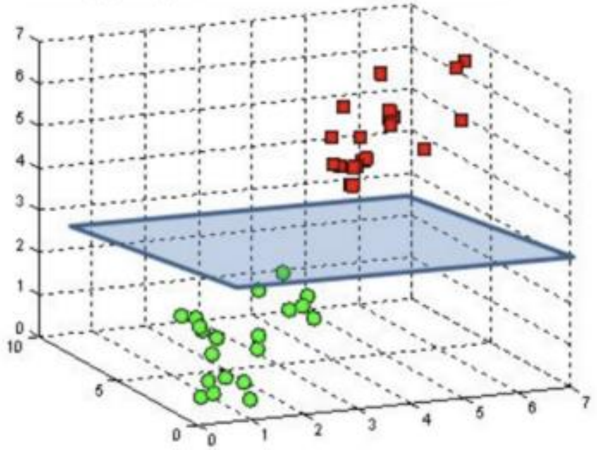
Hyperplane

- 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.

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

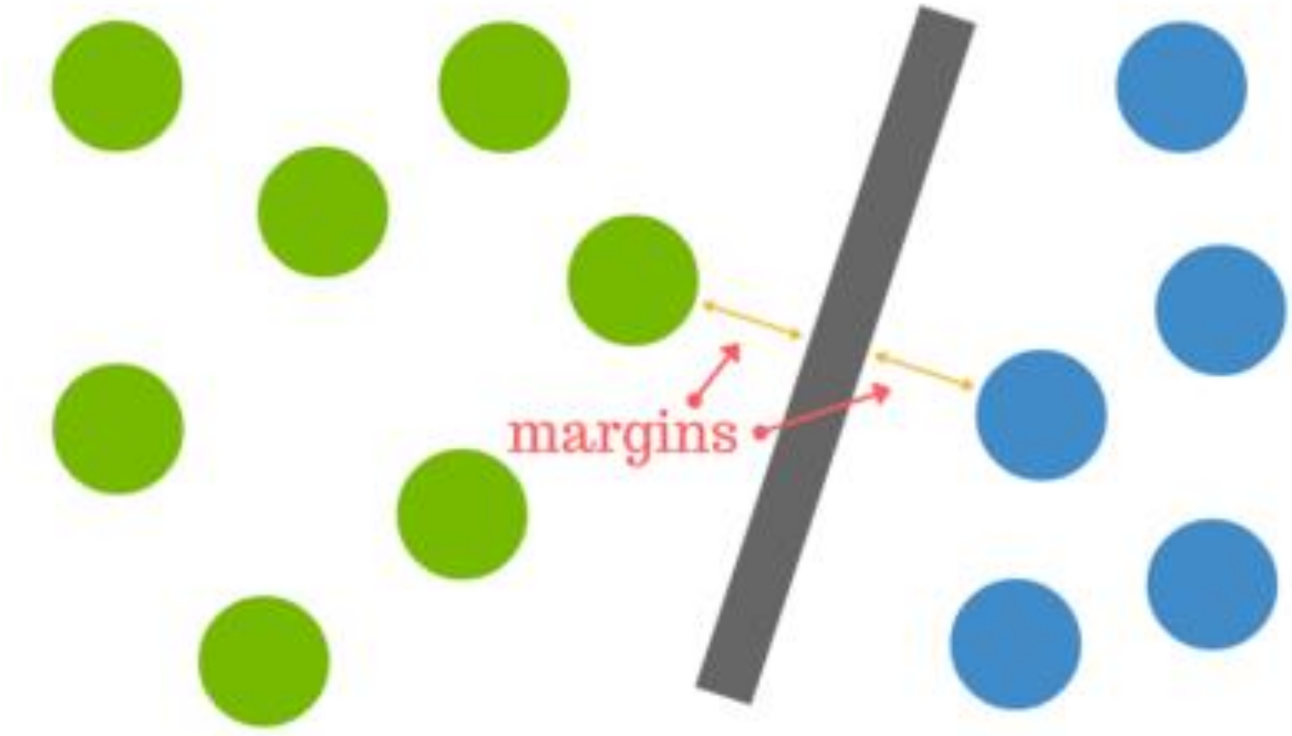


A hyperplane in $\mathbb{R}^3$ is a plane



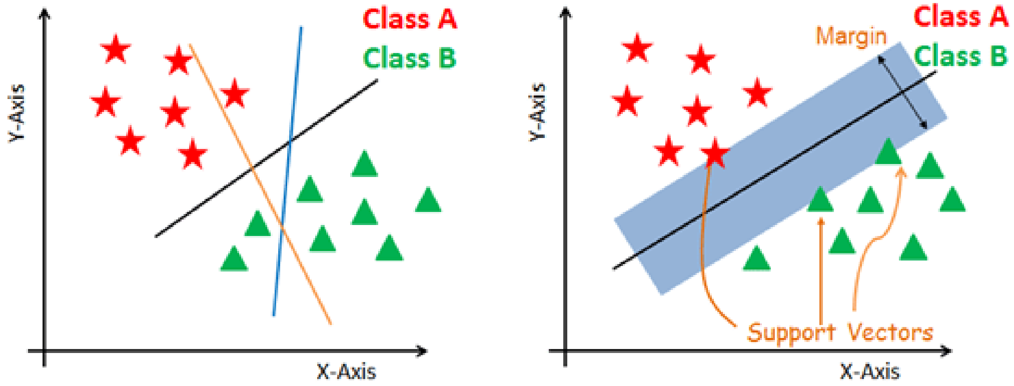
How do we find the right hyperplane?

- Or, in other words, how do we best segregate the two classes within the data?
- The distance between the hyperplane and the nearest data point from either set is known as the margin.
- The goal is to choose a hyperplane with the greatest possible margin between the hyperplane and any point within the training set, giving a greater chance of new data being classified correctly.



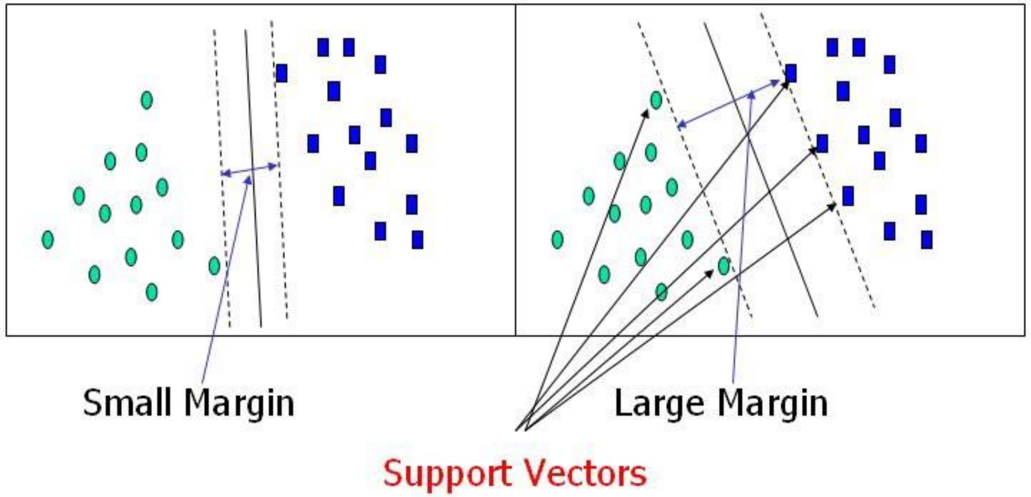
SVM searches for the maximum marginal hyperplane in the following steps:

- Generate hyperplanes which segregates the classes in the best way.
- Left-hand side figure showing three hyperplanes black, blue and orange. Here, the blue and orange have higher classification error, but the black is separating the two classes correctly.
- Select the right hyperplane with the maximum segregation from the either nearest data points as shown in the right-hand side figure.



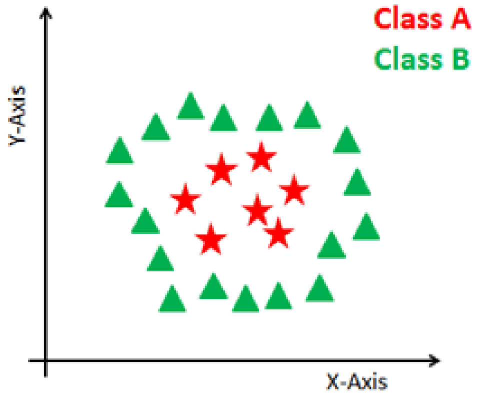
Margin

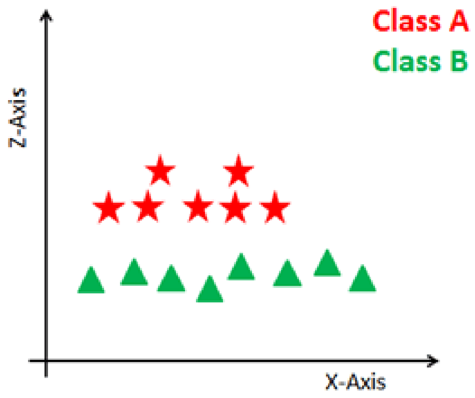
- Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.
- Large Margin Intuition:
- In logistic regression, we take the output of the linear function and squash the value within the range of [0,1] using the sigmoid function. If the squashed value is greater than a threshold value(0.5) we assign it a label 1, else we assign it a label 0. In SVM, we take the output of the linear function and if that output is greater than 1, we identify it with one class and if the output is -1, we identify is with another class. Since the threshold values are changed to 1 and -1 in SVM, we obtain this reinforcement range of values([-1,1]) which acts as margin.

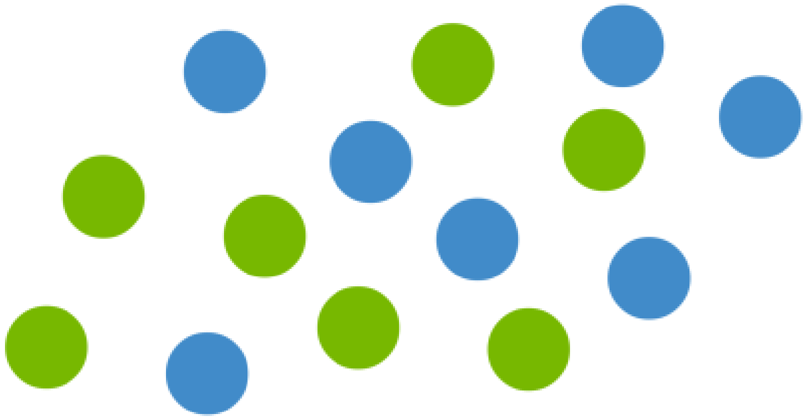


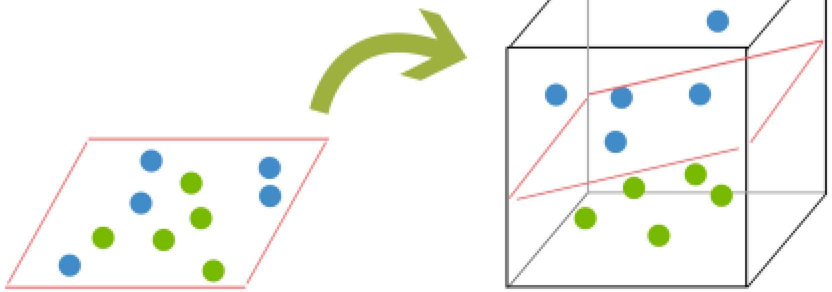
Dealing with non-linear and inseparable planes

- Some problems can’t be solved using linear hyperplane, as shown in the figure below (left-hand side).
- In such situation, SVM uses a kernel trick to transform the input space to a higher dimensional space as shown on the right. The data points are plotted on the x-axis and z-axis (Z is the squared sum of both x and y: z=x^2=y^2).
- Now you can easily segregate these points using linear separation.









SVM Kernels

The SVM algorithm is implemented in practice using a kernel.

A kernel transforms an input data space into the required form.

SVM uses a technique called the kernel trick. Here, the kernel takes a low-dimensional input space and transforms it into a higher dimensional space.

In other words, you can say that it converts nonseparable problem to separable problems by adding more dimension to it. It is most useful in non-linear separation problem.

Kernel trick helps you to build a more accurate classifier.

SVM Kernels

- **Linear Kernel** A linear kernel can be used as normal dot product any two given observations. The product between two vectors is the sum of the multiplication of each pair of input values.
- K(x, xi) = sum(x * xi)
- **Polynomial Kernel** A polynomial kernel is a more generalized form of the linear kernel. The polynomial kernel can distinguish curved or nonlinear input space.
- K(x,xi) = 1 + sum(x * xi)^d
- Where d is the degree of the polynomial. d=1 is similar to the linear transformation. The degree needs to be manually specified in the learning algorithm.
- **Radial Basis Function Kernel** The Radial basis function kernel is a popular kernel function commonly used in support vector machine classification. RBF can map an input space in infinite dimensional space.
- K(x,xi) = exp(-gamma * sum((x – xi)^2)
- Here gamma is a parameter, which ranges from 0 to 1. A higher value of gamma will perfectly fit the training dataset, which causes over-fitting. Gamma=0.1 is considered to be a good default value. The value of gamma needs to be manually specified in the learning algorithm.

Tuning Hyperparameters

- **Kernel**: The main function of the kernel is to transform the given dataset input data into the required form. There are various types of functions such as **linear, polynomial, and radial basis function (RBF)**.
- Polynomial and RBF are useful for non-linear hyperplane. Polynomial and RBF kernels compute the separation line in the higher dimension.
- In some of the applications, it is suggested to use a more complex kernel to separate the classes that are curved or nonlinear. This transformation can lead to more accurate classifiers.

- **Regularization**: Regularization parameter in python's Scikit-learn C parameter used to maintain regularization.
- Here C is the penalty parameter, which represents misclassification or error term.
- The misclassification or error term tells the SVM optimization how much error is bearable.
- This is how you can control the trade-off between decision boundary and misclassification term.
- A smaller value of C creates a small-margin hyperplane and a larger value of C creates a larger-margin hyperplane.

- **Gamma**: A lower value of Gamma will loosely fit the training dataset, whereas a higher value of gamma will exactly fit the training dataset, which causes over-fitting.
- In other words, you can say a low value of gamma considers only nearby points in calculating the separation line, while the a value of gamma considers all the data points in the calculation of the separation line.

Advantages

SVM Classifiers offer good accuracy and perform faster prediction compared to Naïve Bayes algorithm.

They also use less memory because they use a subset of training points in the decision phase.

SVM works well with a clear margin of separation and with high dimensional space.

Disadvantages

SVM is not suitable for large datasets because of its high training time and it also takes more time in training compared to Naïve Bayes.

It works poorly with overlapping classes and is also sensitive to the type of kernel used.

Cost Function and Gradient Updates

- In the SVM algorithm, we are looking to maximize the margin between the data points and the hyperplane. The loss function that helps maximize the margin is hinge loss.
- The cost is 0 if the predicted value and the actual value are of the same sign. If they are not, we then calculate the loss value. We also add a regularization parameter the cost function. The objective of the regularization parameter is to balance the margin maximization and loss. After adding the regularization parameter, the cost functions looks as below.
- https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47