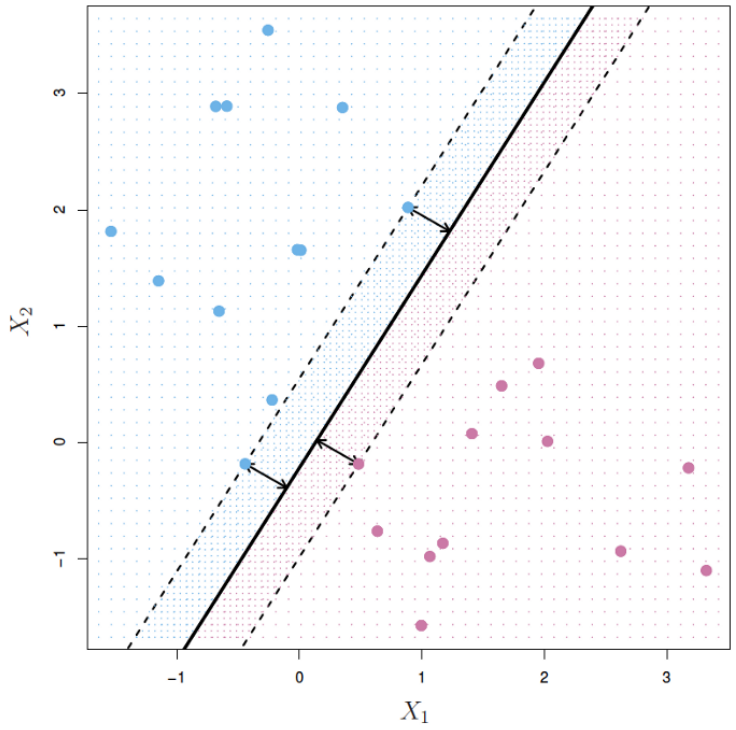
Question 1

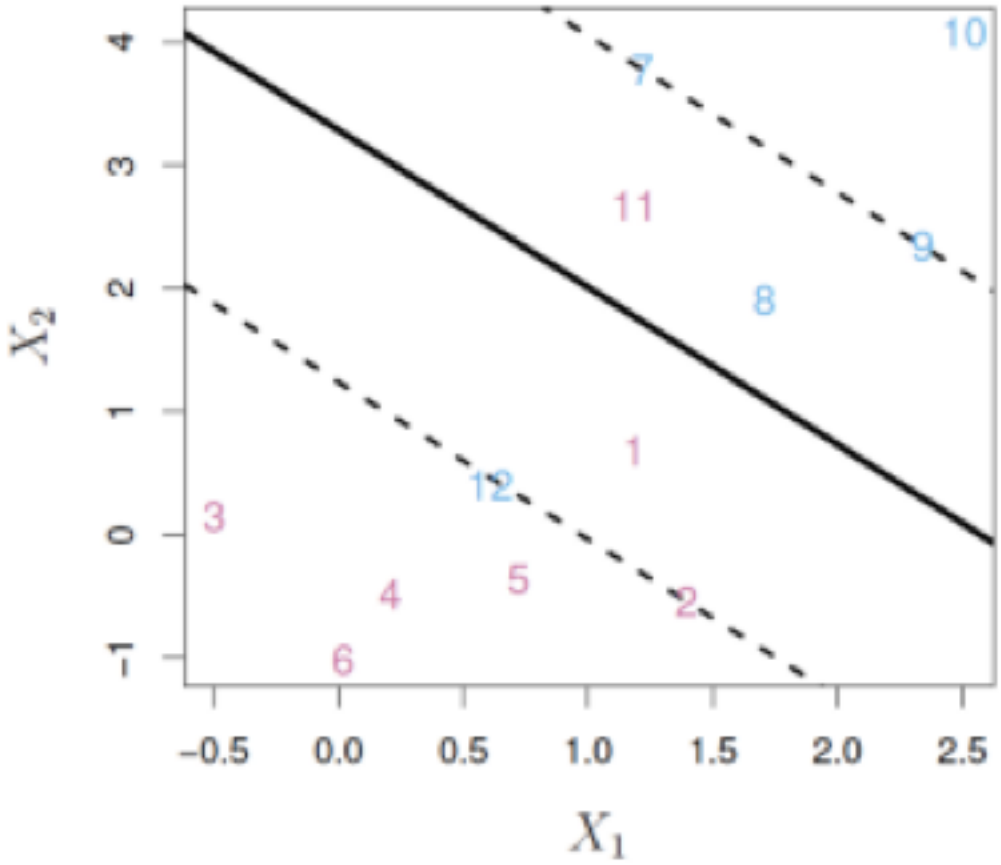
How is Soft Margin Classifier different from Maximum Margin Classifier?

Ans -1

A Maximum Margin classifier is the best hyperplane (plane or line) that perfectly separates the two different classes under observation in a way that it maintains the largest possible equal distance from the nearest point of both the classes. It is suited if both the classes are linearly separable and have clear separation and do not have any overlaps. It is rigid on the Margins and does not allow any misclassifications.



A soft margin classifier or support vector classifier allows SVM to make a certain number of mistakes and keeps the margin as wide as possible so that other points can be classified correctly. It deliberately allows certain points to be mis classified to allow classification of most of the points correctly in the unseen data and hence is more robust. It is a bit flexible on the margins and allows some observations to fall on the wrong side.



In the above case points 12 and 11 are misclassified by the Soft Margin Classifier. The amount of misclassification is dependent on the hyperparameter C. If C is large, the number of misclassifications allowed has been increased . The support vector classifier can fail if the data is not linearly separable

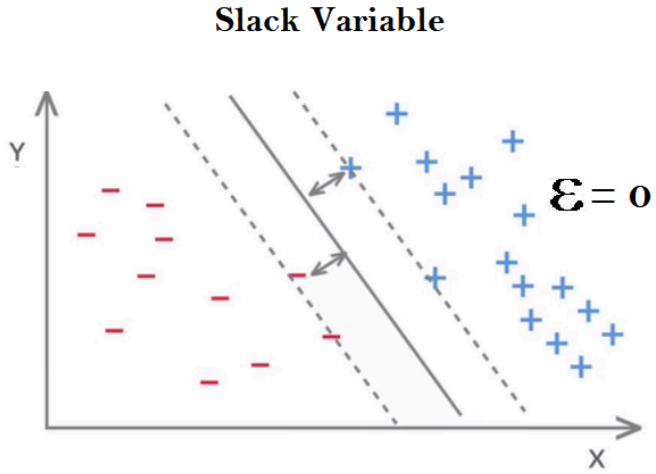
Question 2

What does the slack variable Epsilon (ε) represent?

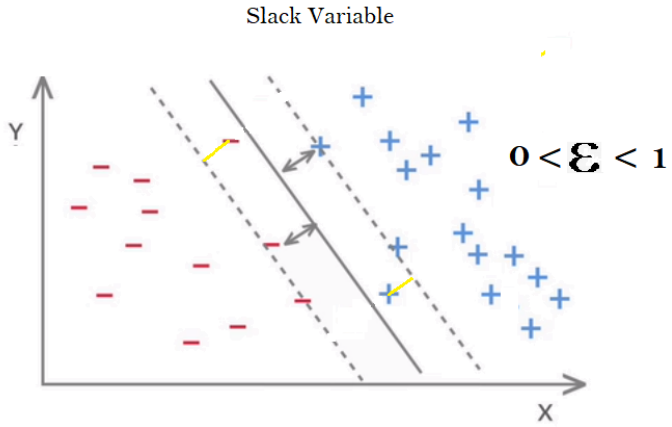
Ans -2

In a Support Vector Classifier or Soft Margin classifier we allow certain points to be deliberately classified. In order to control the misclassification we used the slack variable. Each data point has a slack value associated with it which ranges form 0 to +infinity.

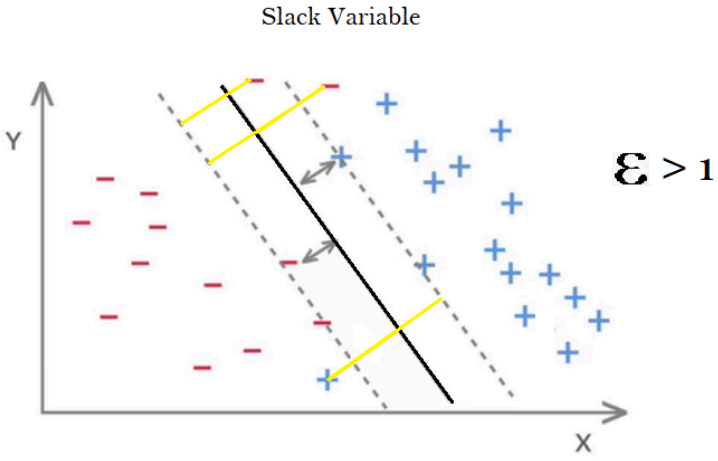
For all points that are classified correctly and are at a safe distance from hyperplane where safe distance >=M (M is the Margin), the value of slack variable is 0.



For those points that are classified correctly but fall within the margin , the slack variable is between 0 and 1.



For points that are misclassified and lie on the other side of the hyperplane, the slack vriable is greater than 1 for them.



In order to compare two Support classifiers one can sum the epsilons of each data point of the two classifiers and judge the one which has least summation of epsilons as better one.

Question 3

How do you measure the cost function in SVM? What does the value of C signify?

Answer -3

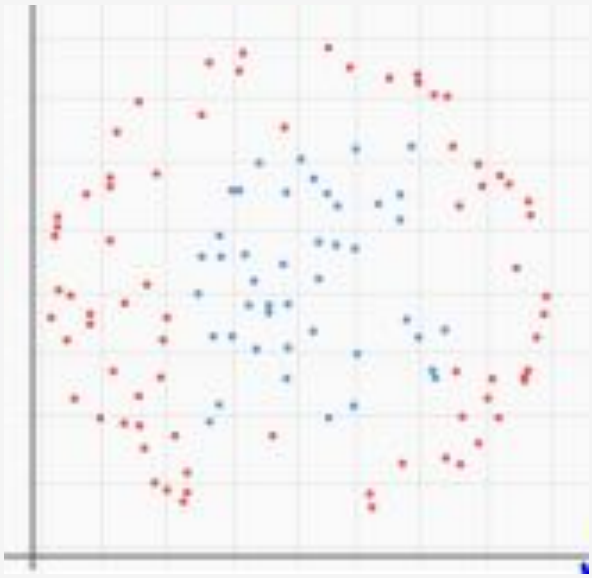
We all know that Support Vector Classifier or Soft Margin classifier allows misclassification of certain points and each point under classification has a slack value associated with it based on its location. The cost function can be calculated by taking summation of the slack values of each data point under classification. The lower is the summation of slack values or value of C ,the lower are the mis classification and better the classifier.

$$\sum \epsilon_i \leq C.$$

A higher value of C means model is flexible, more generalisable and less likely to over fit.. It has high bias.

A lower value of C signifies that the classifier allows very few misclassifications , so Margin is narrow and model is less flexible and more likely to overfit. It has high variance.

Question 4



Given the above dataset where red and blue points represent the two classes, how will you use SVM to classify the data?

Ans -4

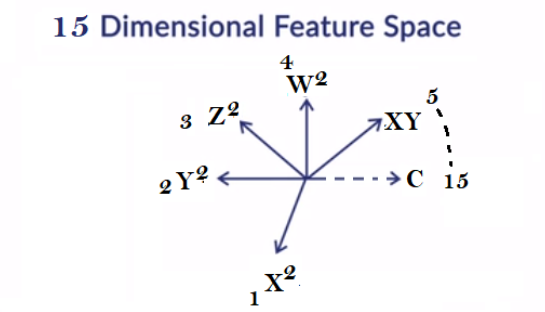
Looking at the given data the data is not linearly separable and the separator will not be a plane rather looks like a circle. So a non-linear SVM will be required to classify the data. A non-linear SVM with an rbf kernel will suit this data to be classified. Using “rbf” kernels, the Kernel Trick will supply the original non-linear data as linear data vis shortcut to SVM.

Question 5 What do you mean by feature transformation?

Ans -5 When you have non-linear data that needs to be classified, you transform the non-linear boundaries to linear boundaries by applying certain functions to original attributes. While doing so, the original attribute space gets transformed in to a new feature space. This process of transforming the original attributes to a new feature space is called Feature Transformation. As the no of original attributes increase there is an exponential rise in the number of dimensions in the transformed feature space.

A 4 D Feature space after transformation

$$a_1X^2 + a_2Y^2 + a_3Z^2 + a_4W^2 + a_5XY + a_6YZ + a_7ZW + a_8WX + a_9WY + a_{10}ZX + a_{11}X + a_{12}Y + a_{13}Z + a_{14}W + C = 0$$



This is computationally intensive if no of attributes in original space are more.