1. How the classifier was implemented?

Support Vector Machines is considered to be a classification approach but can be employed in both types of classification and regression problems. It can easily handle multiple continuous and categorical variables. SVM constructs a hyperplane in multidimensional space to separate different classes.

1. Generate hyperplanes that segregate 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.
2. Select the right hyperplane with the maximum segregation from either nearest data points.

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. Here, I have used **Radial Basis Function Kernel** and **Polynomial Kernel**

- **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.

2. Results comparison among the classifiers tested upto now.

Comparing results with KNN

3. How SVM can be used for multi-class classification?

For multiclass classification, the same principle is utilized after breaking down the multiclassification problem into multiple binary classification problems.

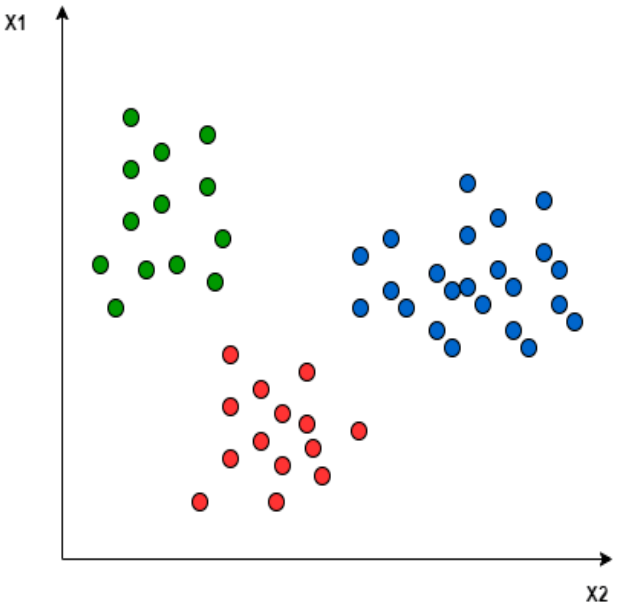
The idea is to map data points to high dimensional space to gain mutual linear separation between every two classes. This is called a *One-to-One* approach, which breaks down the multiclass problem into multiple binary classification problems. A binary classifier per each pair of classes.

Another approach one can use is *One-to-Rest*. In that approach, the breakdown is set to a binary classifier per each class.

A single SVM does binary classification and can differentiate between two classes. So that, according to the two breakdown approaches, to classify data points from **m** classes data set:

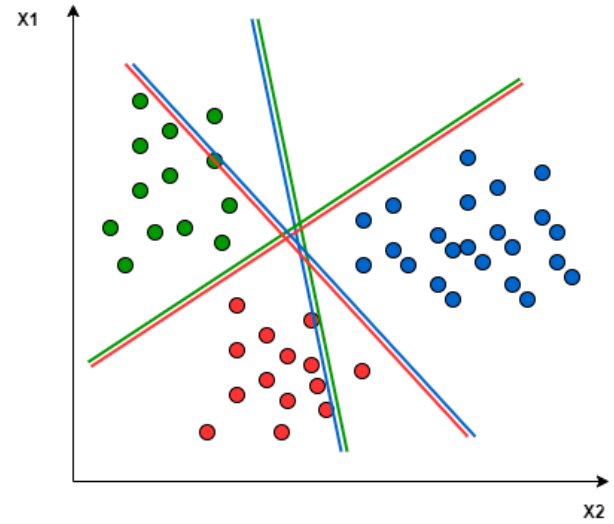
- In the *One-to-Rest* approach, the classifier can use **m** SVMs. Each SVM would predict membership in one of the **m** classes.
- In the *One-to-One* approach, the classifier can use **m(m-1)/2** SVMs.

Let's take an example of 3 classes classification problem; green, red, and blue, as the following image:

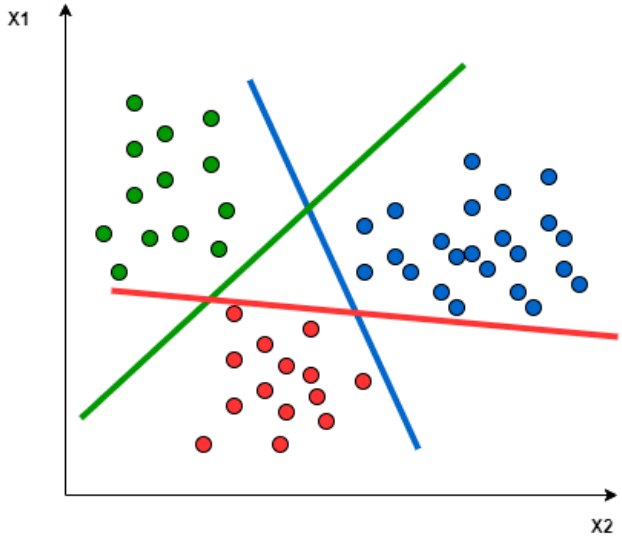


Applying the two approaches to this data set results in the followings:

In the *One-to-One* approach, we need a hyperplane to separate between every two classes, neglecting the points of the third class. This means the separation takes into account only the points of the two classes in the current split. For example, the red-blue line tries to maximize the separation only between blue and red points. It has nothing to do with green points:



In the *One-to-Rest* approach, we need a hyperplane to separate between a class and all others at once. This means the separation takes all points into account, dividing them into two groups; a group for the class points and a group for all other points. For example, the green line tries to maximize the separation between green points and all other points at once:



One of the most common real-world problems for multiclass classification using SVM is text classification. For example, classifying news articles, tweets, or scientific papers.