****

**Hamdard University Islamabad**

**Assignment # 03**

**Artificial Intelligence**

**Members Name**

Hassan Farid(1240-2020)

Aqsa Farooq (1468-2020)

Ali Najam(1843-2020)

Arbab Hussain(1837-2020)

**Submitted To:** Dr. Shaheer Khan

## **Supervised vs Unsupervised learning**

Two of the most commonly used strategies in machine learning include supervised learning and unsupervised learning.

### What is supervised learning?

Supervised learning is when you train a machine learning model using labelled data. It means that you have data that already have the right classification associated with them. One common use of supervised learning is to help you predict values for new data.

With supervised learning, you'll need to rebuild your models as you get new data to make sure that the predictions returned are still accurate. An example of supervised learning would be labeling pictures of food. You could have a dataset dedicated to just images of pizza to teach your model what pizza is.

### What is unsupervised learning?

Unsupervised learning is when you train a model with unlabeled data. This means that the model will have to find its own features and make predictions based on how it classifies the data.

An example of unsupervised learning would be giving your model pictures of multiple kinds of food with no labels. The dataset would have images of pizza, fries, and other foods and you could use different algorithms to get the model to identify just the images of pizza without any labels.

**Support Vector Machine (SVM)**

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. Developed at AT&T Bell Laboratories by Vladimir Vapnik with colleagues (Boser et al., 1992, Guyon et al., 1993, Cortes and Vapnik, 1995, Vapnik et al).

The main objective of the SVM algorithm is to find the optimal hyperplane in an N-dimensional space that can separate the data points in different classes in the feature space. The hyperplane tries that the margin between the closest points of different classes should be as maximum as possible. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

## **Types of Support Vector Machines**

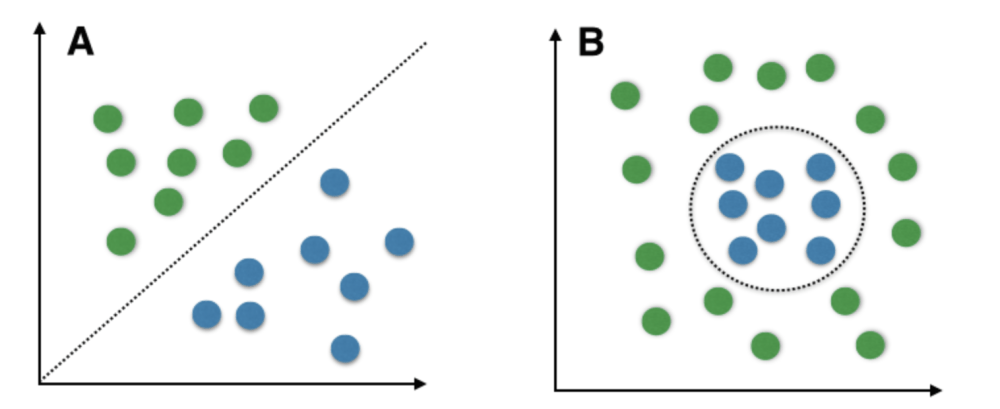
Support vector machines are broadly classified into two types: simple or linear SVM and kernel or non-linear SVM.

#### **1. Linear SVM**

When the data is perfectly linearly separable only then we can use Linear SVM. Perfectly linearly separable means that the data points can be classified into 2 classes by using a single straight line(if 2D).

**2. Non-Linear SVM**

When the data is not linearly separable then we can use Non-Linear SVM, which means when the data points cannot be separated into 2 classes by using a straight line (if 2D) then we use some advanced techniques like kernel tricks to classify them. In most real-world applications we do not find linearly separable datapoints hence we use kernel trick to solve them.

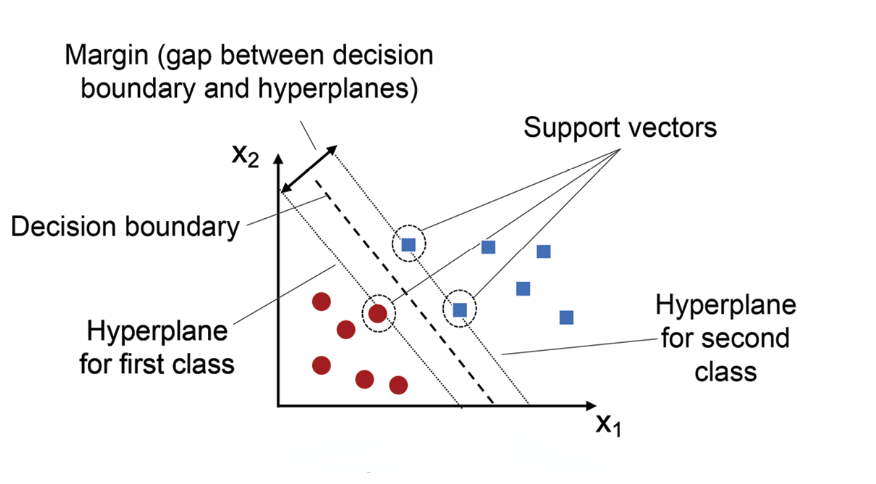


Linear SVM Non-Linear SVM

**Important Terms**

Now let’s define two main terms which will be repeated again and again in this article:

* **Support Vectors:**These are the points that are closest to the hyperplane. A separating line will be defined with the help of these data points.
* **Margin:** it is the distance between the hyperplane and the observations closest to the hyperplane (support vectors). In SVM large margin is considered a good margin. There are two types of margins **hard margin** and **soft margin.**



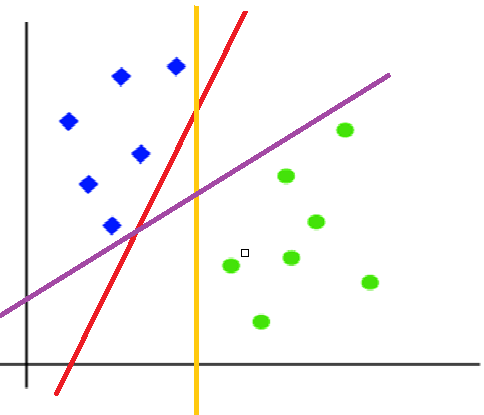
**How Does Support Vector Machine Work?**

SVM is defined such that it is defined in terms of the support vectors only, we don’t have to worry about other observations since the margin is made using the points which are closest to the hyperplane (support vectors), whereas in logistic regression the classifier is defined over all the points. Hence SVM enjoys some natural speed-ups.

Let’s understand the working of SVM using an example. Suppose we have a dataset that has two classes (green and blue). We want to classify that the new data point as either blue or green.



To classify these points, we can have many decision boundaries, but the question is which is the best and how do we find it? **NOTE:**Since we are plotting the data points in a 2-dimensional graph we call this decision boundary a **straight line** but if we have more dimensions, we call this decision boundary a **“hyperplane”**



The best hyperplane is that plane that has the maximum distance from both the classes, and this is the main aim of SVM. This is done by finding different hyperplanes which classify the labels in the best way then it will choose the one which is farthest from the data points or the one which has a maximum margin.



## **Why SVMs are used in machine learning**

SVMs are used in applications like

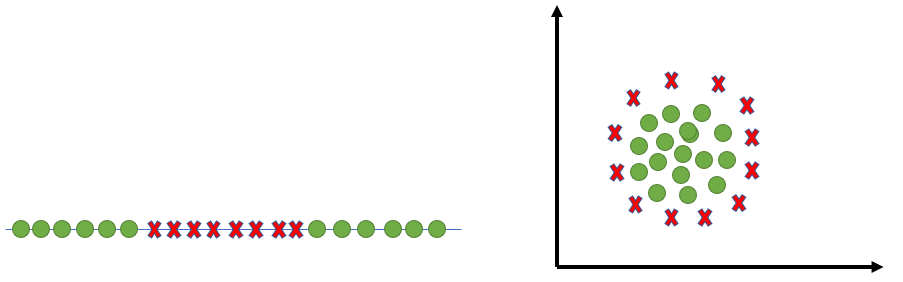
* Handwriting Recognition
* Intrusion Detection
* Face Detection
* Email Classification
* Gene Classification
* In Web Pages.

This is one of the reasons we use SVMs in machine learning. It can handle both classification and regression on linear and non-linear data.

It's a great option when you are working with smaller datasets that have tens to hundreds of thousands of features. They typically find more accurate results when compared to other algorithms because of their ability to handle small, complex datasets.

**Kernels in Support Vector Machine**

The most interesting feature of SVM is that it can even work with a non-linear dataset and for this, we use “Kernel Trick” which makes it easier to classifies the points. Suppose we have a dataset like this:



Here we see we cannot draw a single line or say hyperplane which can classify the points correctly. So what we do is try converting this lower dimension space to a higher dimension space using some quadratic functions which will allow us to find a decision boundary that clearly divides the data points. These functions which help us do this are called Kernels and which kernel to use is purely determined by hyperparameter tuning.



**Different Kernel Functions**

Some kernel functions which you can use in SVM are given below:

**1. Linear**

These are commonly recommended for text classification because most of these types of classification problems are linearly separable.

The linear kernel works really well when there are a lot of features, and text classification problems have a lot of features. Linear kernel functions are faster than most of the others and you have fewer parameters to optimize.

**Here's the function that defines the linear kernel:**

**f(X) = w^T \* X + b**

In this equation, **w**is the weight vector that you want to minimize, **X**is the data that you're trying to classify, and **b**is the linear coefficient estimated from the training data. This equation defines the decision boundary that the SVM returns.

**2. Polynomial**

The polynomial kernel isn't used in practice very often because it isn't as computationally efficient as other kernels and its predictions aren't as accurate.

**Here's the function for a polynomial kernel:**

**f(X1, X2) = (a + X1^T \* X2) ^ b**

This is one of the simpler polynomial kernel equations you can use. **f(X1, X2)**represents the polynomial decision boundary that will separate your data. **X1**and **X2** represent your data.

**3. Gaussian Radial Basis Function (RBF)**

One of the most powerful and commonly used kernels in SVMs. Usually the choice for non-linear data.

**Here's the equation for an RBF kernel:**

**f(X1, X2) = exp(-gamma \* ||X1 - X2||^2)**

In this equation, **gamma** specifies how much a single training point has on the other data points around it. **||X1 - X2||**is the dot product between your features.

**4. Sigmoid**

More useful in neural networks than in support vector machines, but there are occasional specific use cases.

**Here's the function for a sigmoid kernel:**

**f(X, y) = tanh(alpha \* X^T \* y + C)**

In this function, **alpha**is a weight vector and **C**is an offset value to account for some mis-classification of data that can happen.

**Others**

There are plenty of other kernels you can use for your project. This might be a decision to make when you need to meet certain error constraints, you want to try and speed up the training time, or you want to super tune parameters.

[Some other kernels include](https://data-flair.training/blogs/svm-kernel-functions/): ANOVA radial basis, hyperbolic tangent, and Laplace RBF.

