MIZAN TEPI UNIVERSITY

TEPI CAMPUS

SCHOOL OF COMPUTING AND INFORMATICS

DEPARTMENT OF: INFORMATION SYSTEMS

ASSIGNMENT OF: INTRODUCTION TO MACHINE LEARNING

COURSE CODE: INSY 4101

TITLE: SUPPORT VECTOR MACHINE

Group Members Name **ID No**

1. Amanuel Derese...NSR/0197/12
2. Behija Yussuf.. NSR/0435/12
3. Zekarias Karie.. NSR/2041/12
4. Misganaw Amanu.. NSR/1404/12
5. Yohannis Biwota.. NSR/2005/12

SUBMITTED TO: Mr. GEBREYES G.

SUBMMATION DATE: FEB 6 2023

Table of Contents

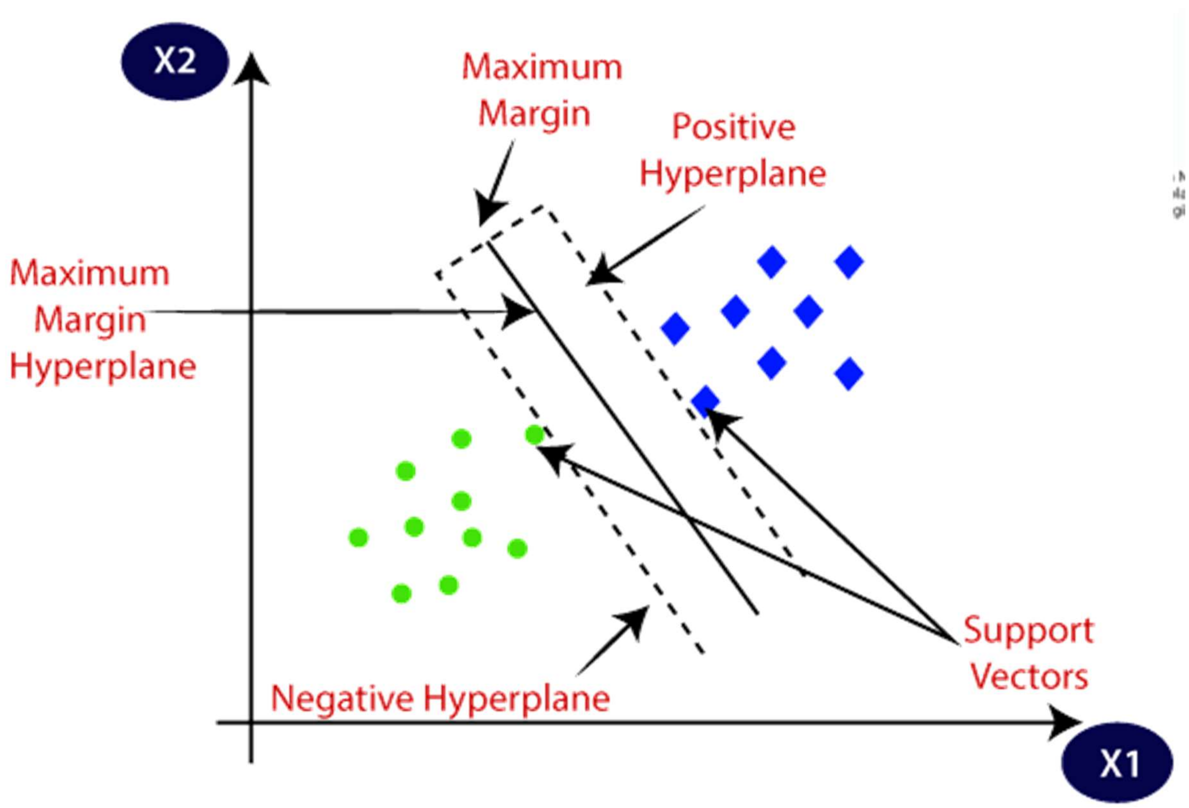
What is Support Vector Machine Algorithm

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

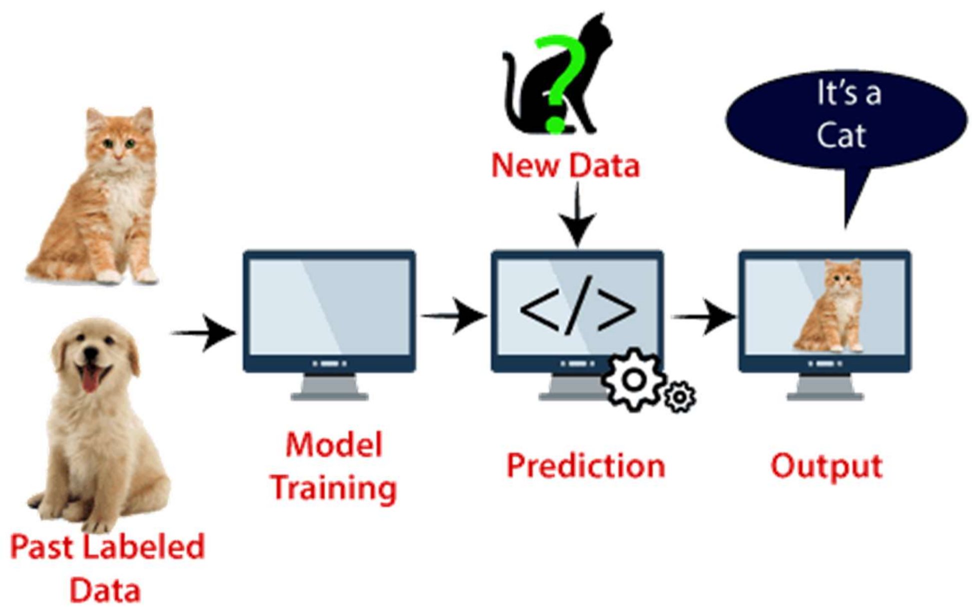
The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

:



Example: SVM can be understood with the example that we have used in the KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat. Consider the below diagram:



SVM algorithm can be used for **Face detection, image classification, text categorization,** etc.

Types of SVM

SVM can be of two types:

Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane and Support Vectors in the SVM algorithm:

Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

Support Vectors:

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

Examples of Support Vector Machines

SVMs rely on supervised learning methods to classify unknown data into known categories. These find applications in diverse fields.

Here, we'll look at some of the top real-world examples of SVMs:

1. Addressing the geo-sounding problem

The geo-sounding problem is one of the widespread use cases for SVMs, wherein the process is employed to track the planet's layered structure. This entails solving the inversion problems where the observations or results of the issues are used to factor in the variables or parameters that produced them.

In the process, linear function and support vector algorithmic models separate the electromagnetic data. Moreover, linear programming practices are employed while developing the supervised models in this case. As the problem size is considerably small, the dimension size is inevitably tiny, which accounts for mapping the planet's structure.

2. Assessing seismic liquefaction potential

Soil liquefaction is a significant concern when events such as earthquakes occur. Assessing its potential is crucial while designing any civil infrastructure. SVMs play a key role in determining the occurrence and non-occurrence of such liquefaction aspects. Technically, SVMs handle two tests: SPT (Standard Penetration Test) and CPT (Cone Penetration Test), which use field data to adjudicate the seismic status.

Moreover, SVMs are used to develop models that involve multiple variables, such as soil factors and liquefaction parameters, to determine the ground surface strength. It is believed that SVMs achieve an accuracy of close to 96-97% for such applications.

3. Protein remote homology detection

Protein remote homology is a field of computational biology where proteins are categorized into structural and functional parameters depending on the sequence of amino acids when sequence identification is seemingly difficult. SVMs play a key role in remote homology, with kernel functions determining the commonalities between protein sequences.

Thus, SVMs play a defining role in computational biology.

4. Data classification

SVMs are known to solve complex mathematical problems. However, smooth SVMs are preferred for data classification purposes, wherein smoothing techniques that reduce the data outliers and make the pattern identifiable are used.

Thus, for optimization problems, smooth SVMs use algorithms such as the Newton-Armijo algorithm to handle larger datasets that conventional SVMs cannot. Smooth SVM types typically explore math properties such as strong convexity for more straightforward data classification, even with non-linear data.

5. Facial detection & expression classification

SVMs classify facial structures vs. non-facial ones. The training data uses two classes of face entity (denoted by +1) and non-face entity (denoted as -1) and n*n pixels to distinguish between face and non-face structures. Further, each pixel is analyzed, and the features from each one are extracted that denote face and non-face characters. Finally, the process creates a square decision boundary around facial structures based on pixel intensity and classifies the resultant images.

Moreover, SVMs are also used for facial expression classification, which includes expressions denoted as happy, sad, angry, surprised, and so on.

6. Surface texture classification

In the current scenario, SVMs are used for the classification of images of surfaces. Implying that the images clicked of surfaces can be fed into SVMs to determine the texture of surfaces in those images and classify them as smooth or gritty surfaces.

Why SVMs are used in machine learning

SVMs are used in applications like handwriting recognition, intrusion detection, face detection, email classification, gene classification, and 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.

Another reason we use SVMs is because they can find complex relationships between your data without you needing to do a lot of transformations on your own. 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.

WE NEED BECAUSE OF:

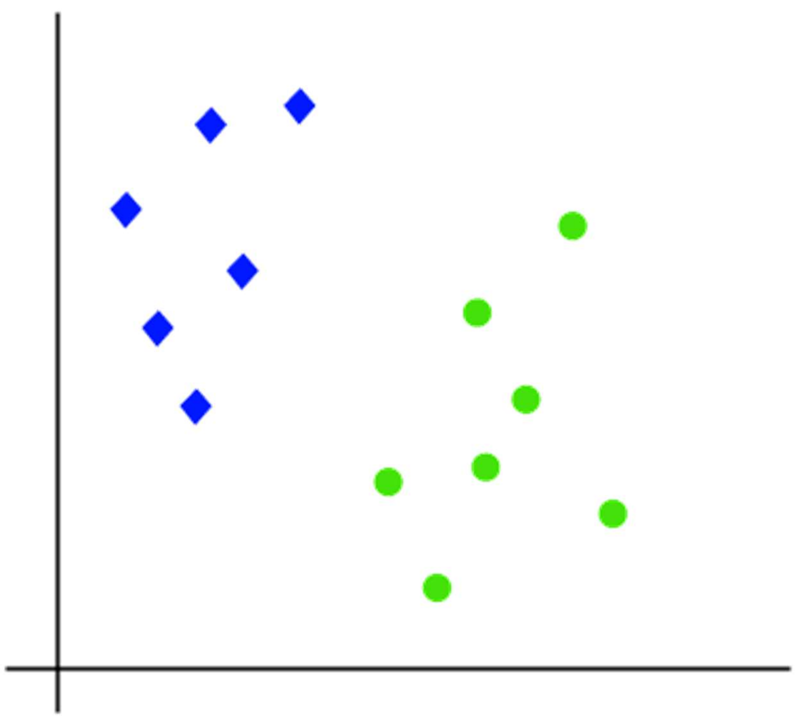
- Effective on datasets with multiple features, like financial or medical data.
- Effective in cases where number of features is greater than the number of data points.
- Uses a subset of training points in the decision function called support vectors which makes it memory efficient.

- Different kernel functions can be specified for the decision function. You can use common kernels, but it's also possible to specify custom kernels.

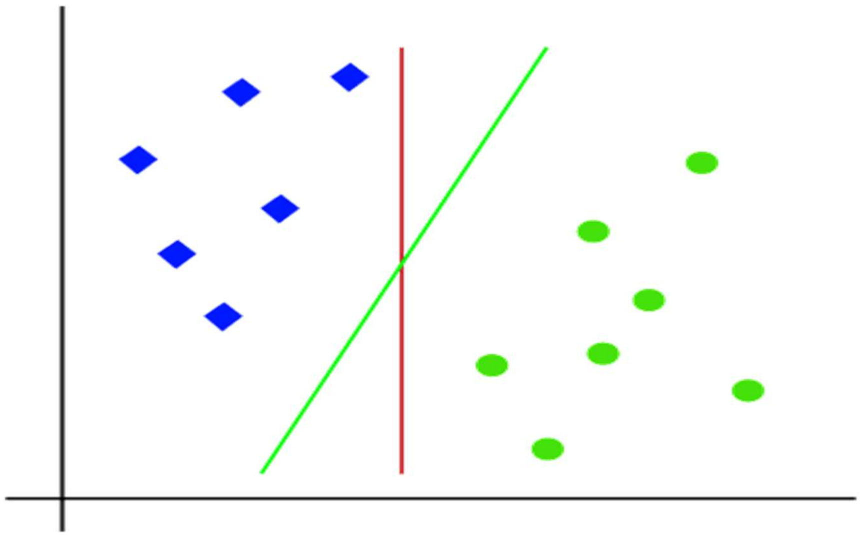
How does SVM works?

Linear SVM:

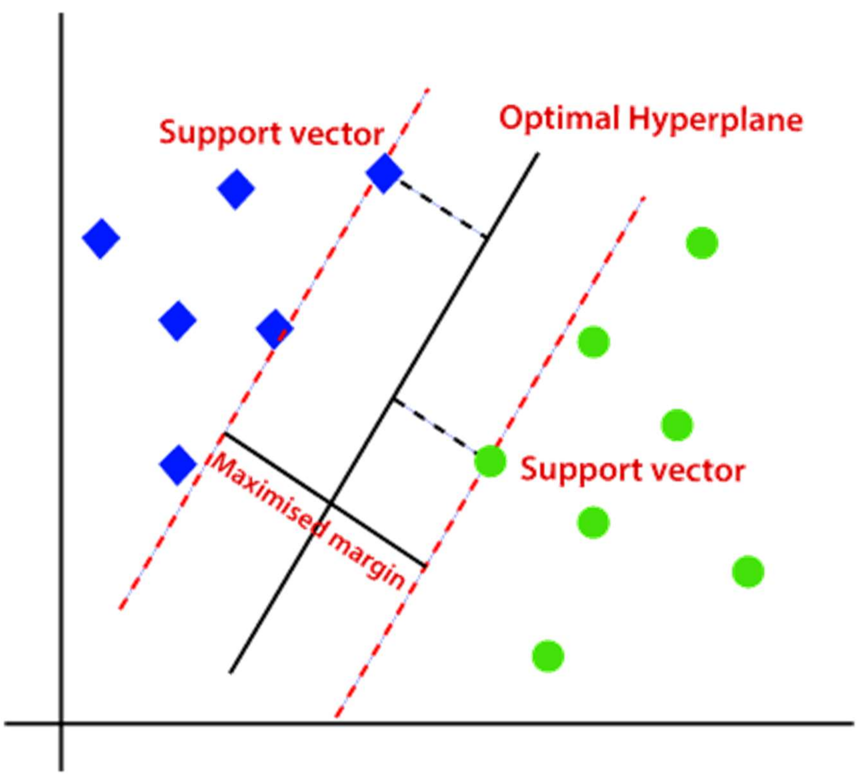
The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the below image:



So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:

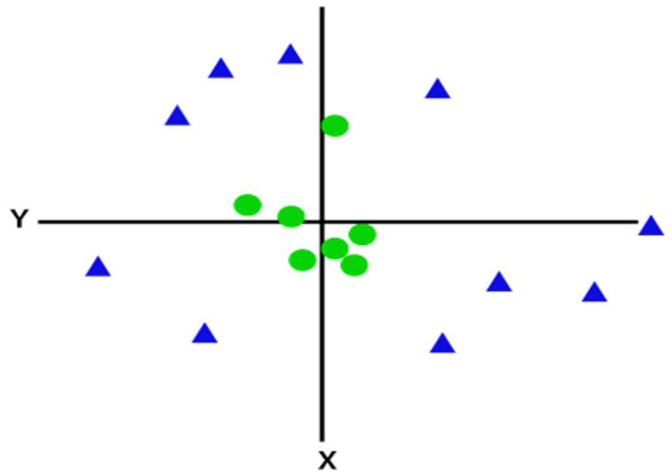


Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as **margin**. And the goal of SVM is to maximize this margin. The **hyperplane** with maximum margin is called the **optimal hyperplane**.



Non-Linear SVM:

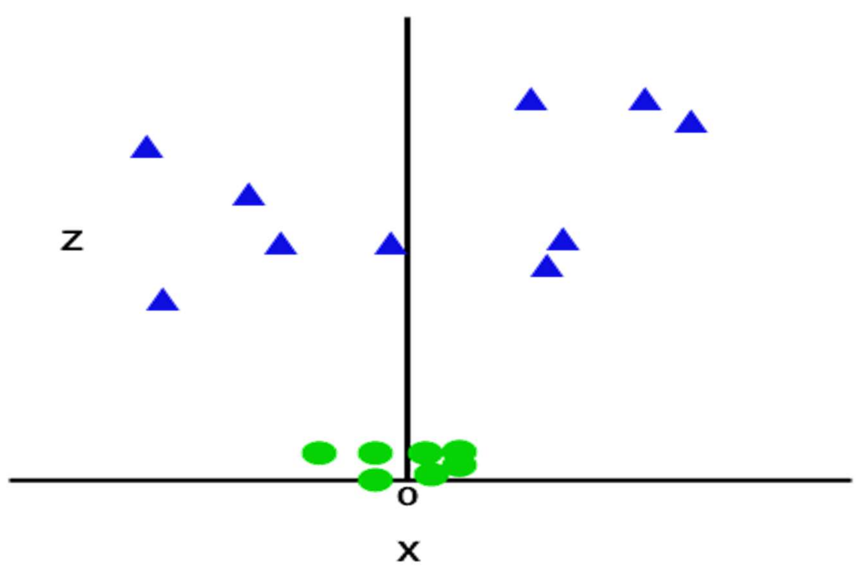
If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image:



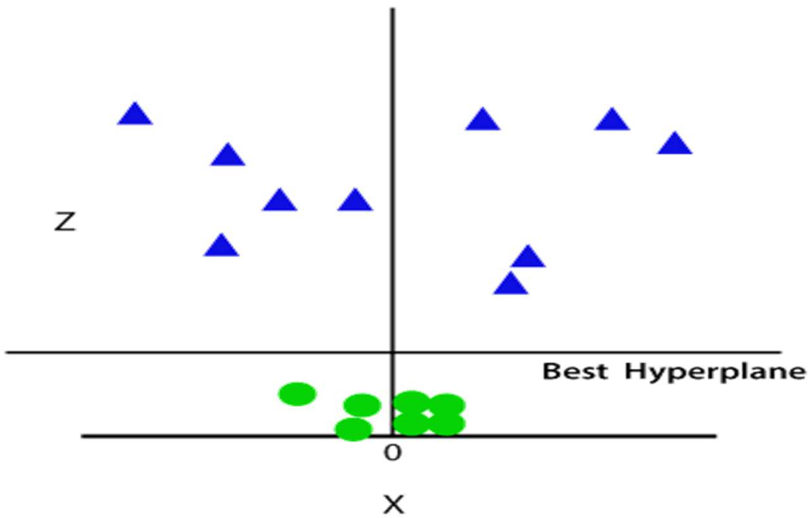
So to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions' x and y, so for non-linear data, we will add a third dimension z. It can be calculated as:

$z=x^2 +y^2$

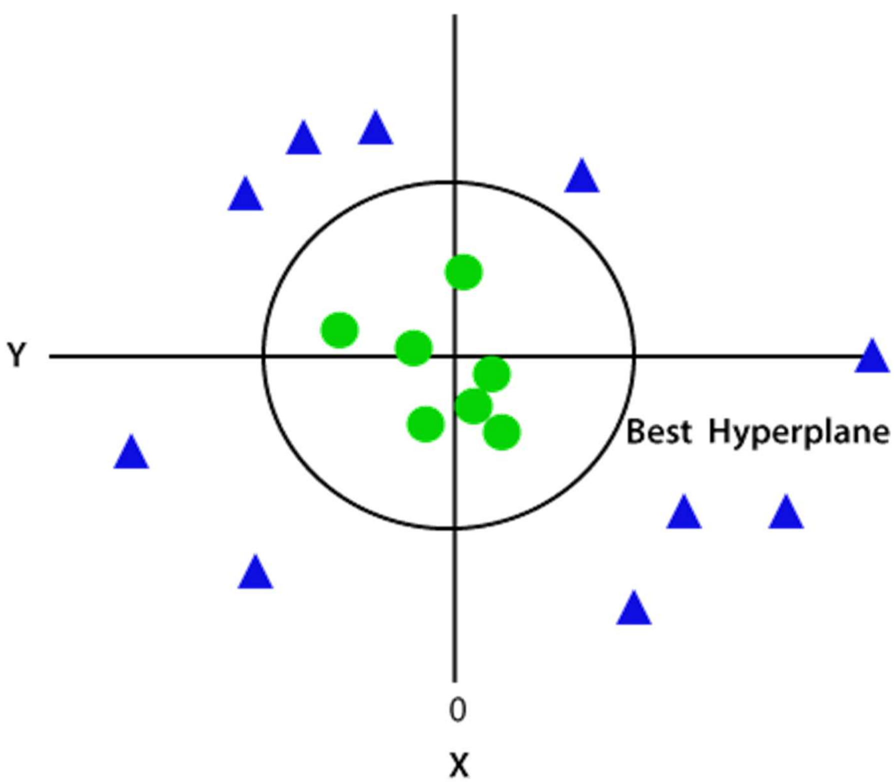
By adding the third dimension, the sample space will become as below image:



So now, SVM will divide the datasets into classes in the following way. Consider the below image:



Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1, then it will become as:



Hence we get a circumference of radius 1 in case of non-linear data.

Reference

- https://towardsdatascience.com
- https://www.javatpoint.com
- https://www.spiceworks.com