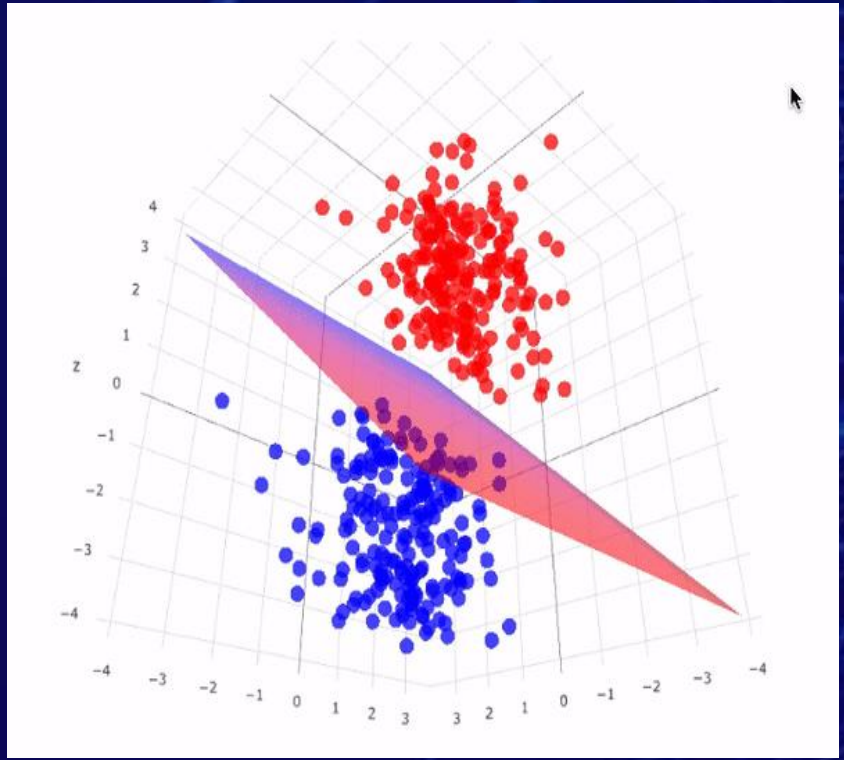
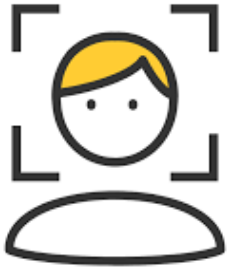
Unit 2.5
Support Vector Machine



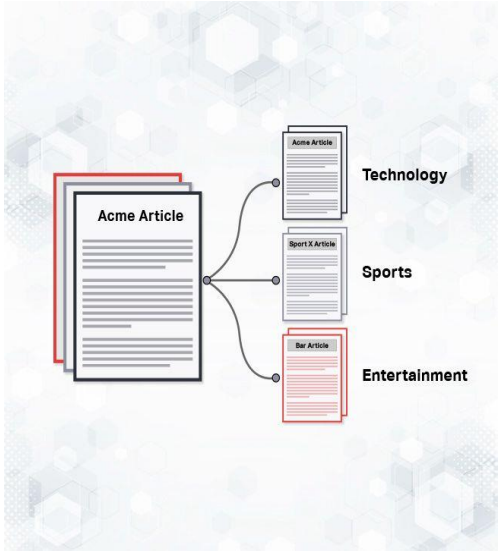
Disclaimer

The content is curated from online/offline resources and used for educational purpose only

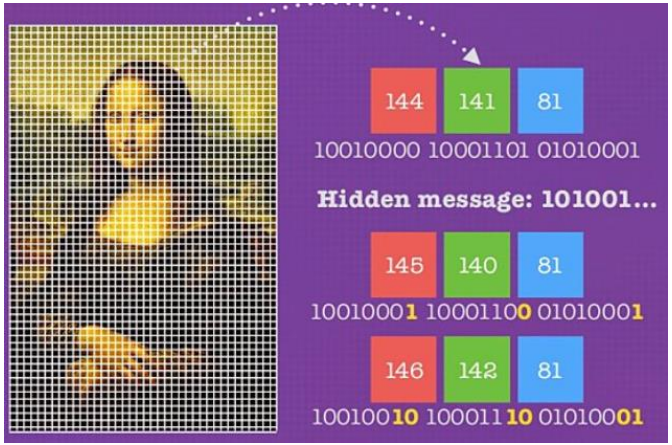
Support Vector Machine



Face Detection



Text Classification



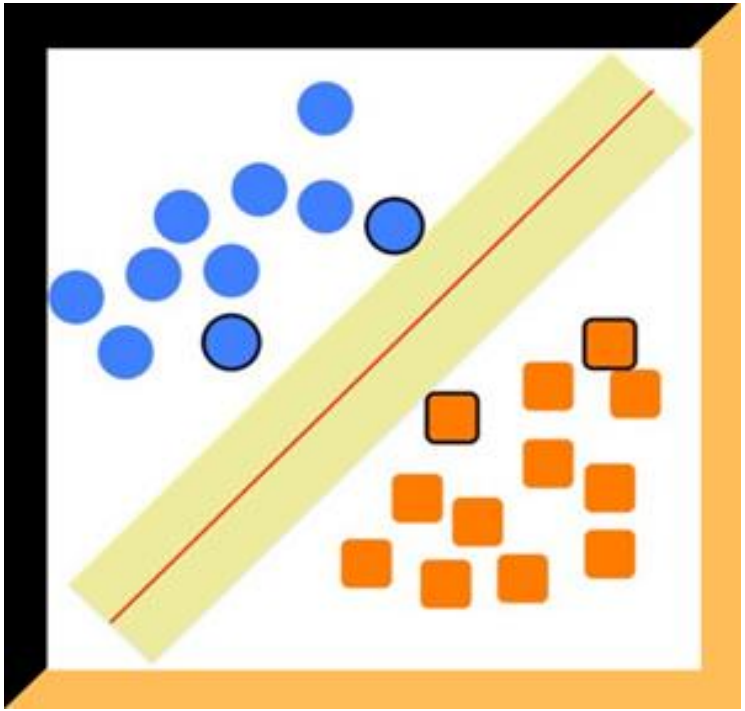
Steganography detection

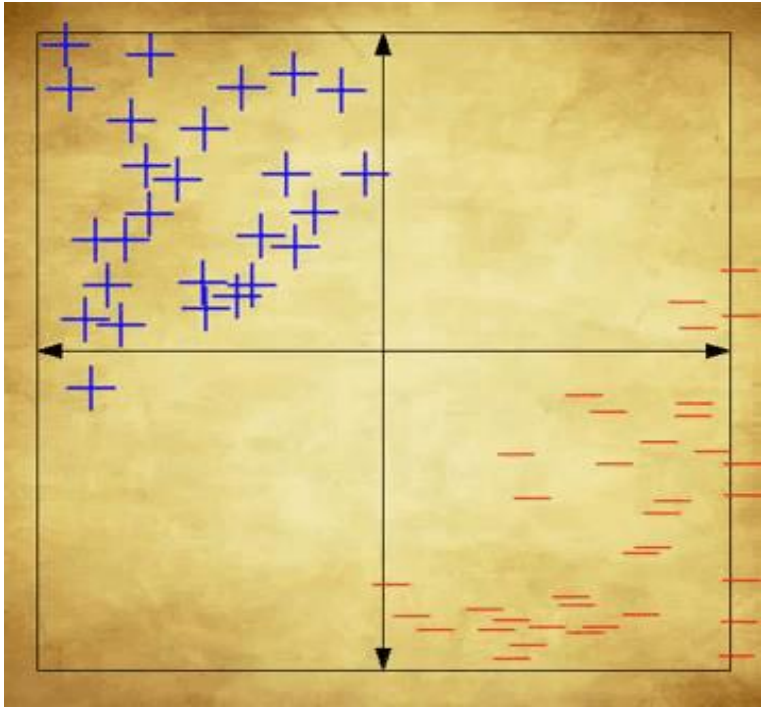
Learning Objectives

You will learn in this lesson:

- Support Vector Machine
- Features space
- Decision Boundary
- Dimension Expansion
- Hyperplane
- Transformation Approach

Support Vector Machine (SVM)





Reference

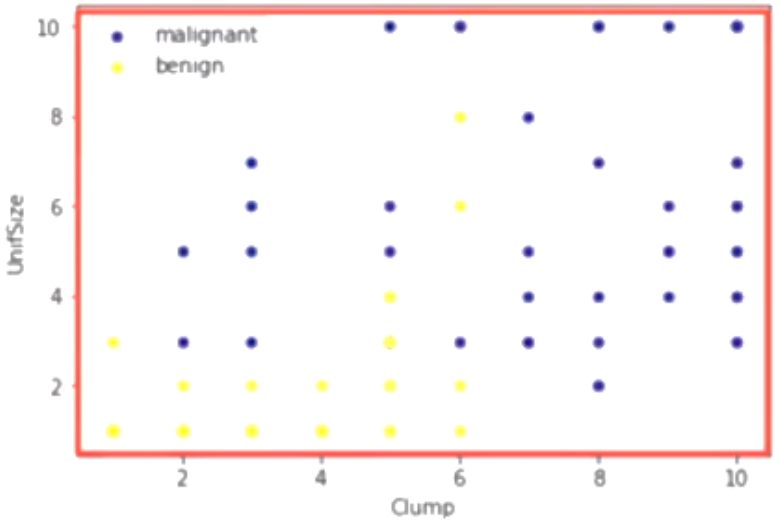
Introduction

- A Support Vector Machine is a supervised algorithm that can classify cases by **finding a separator**.
- SVM works by first, **mapping data to a high-dimensional feature space** so that data points can be categorized, even when the data are not otherwise linearly separable.
- Then, a separator is estimated for the data.
- The data should be transformed in such a way that a separator could be drawn as a hyperplane.

Feature Space

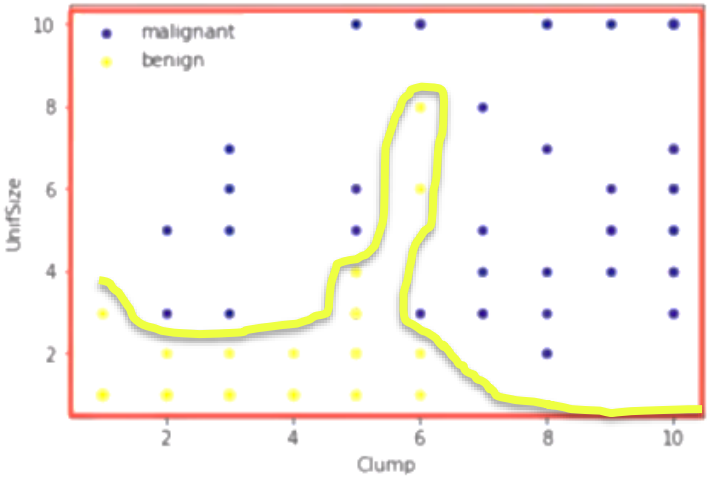
For example, consider the following figure, which shows the distribution of a small set of cells, only based on their Unit Size and Clump thickness.

It represents a linearly, non-separable, dataset.



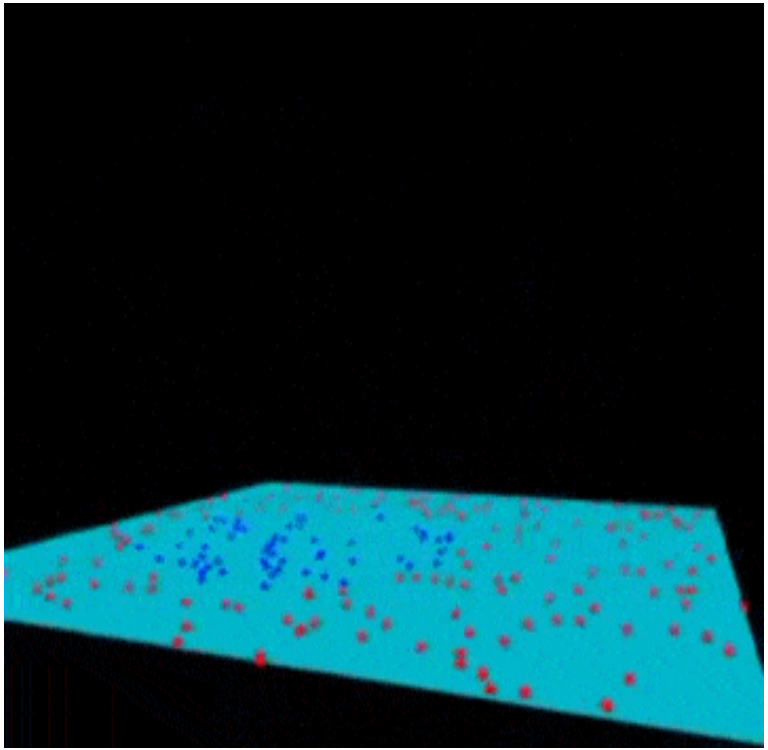
Decision Boundary..

- The two categories can be separated with a curve.
- Not a line that formulates most real world datasets.
- Twist and Turns on trajectory



Dimension Expansion

- We can transfer this data to a higher dimensional space.
- For example, mapping it to a 3-dimensional space.
- Separation boundary gets simplified



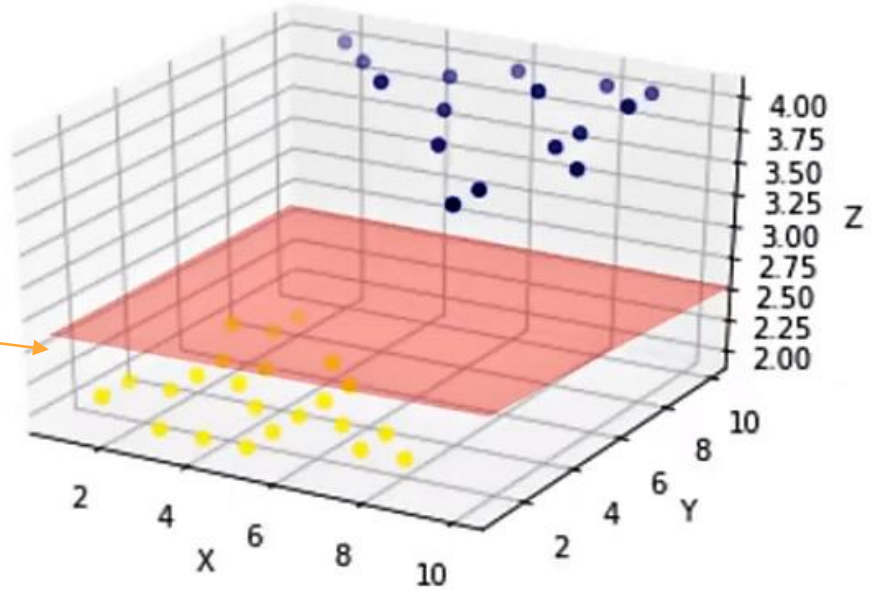
Reference

Hyperplane

After the transformation, the boundary between the two categories can be defined by a hyperplane.

As we are now in 3-dimensional space, the separator is shown as a plane.

This plane can be used to classify new or unknown cases.



Reference

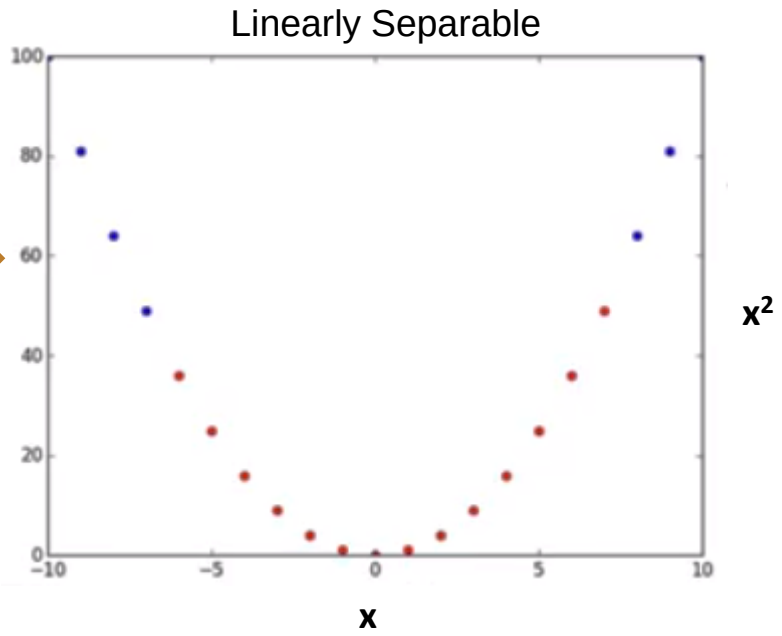
Transformation Approach

For example, your can increase the dimension of data by

mapping x into a new space using a function, with outputs x and x^2.

$$\emptyset(x) = [x, x^2]$$

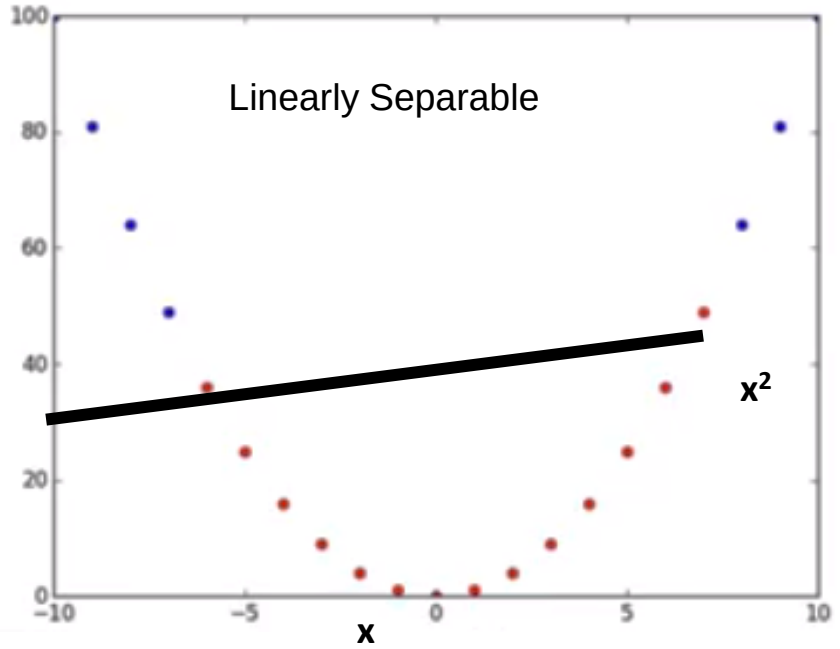
Now, the data is linearly separable!



Transformation Approach

Notice that, as we are in a two-dimensional space, the hyperplane is a line dividing a plane into two parts where each class lays on either side.

Now we can use this line to classify new cases.



Reference

Kernel in Support Vector Machine

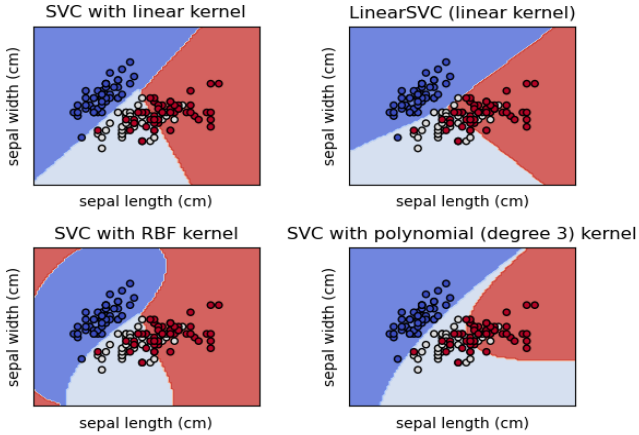
Mapping data into a higher dimensional space is called kernelling.

The mathematical function used for the transformation is known as the **kernel function**, and can be of different types, such as:

- Linear,
- Polynomial,
- Radial basis function (or RBF), and
- Sigmoid.

Already implemented in form of machine learning libraries.

Choose different functions in turn and compare the results.

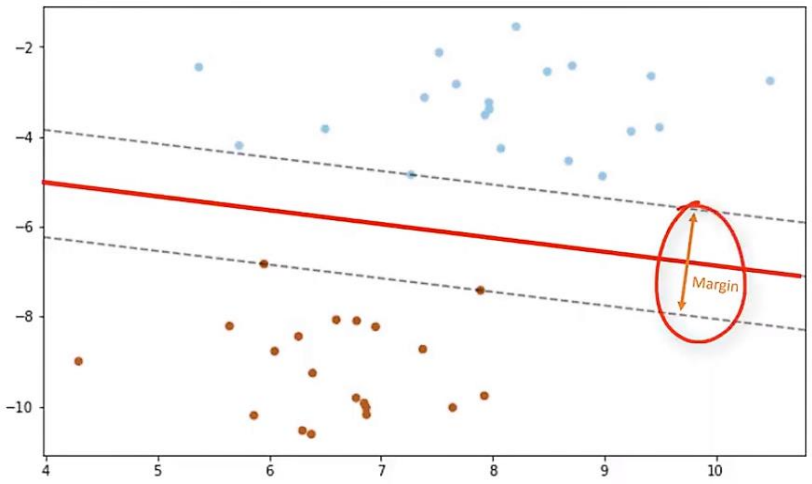


Reference

Finding Optimized Separator after Transformation

One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes.

So, the goal is to choose a hyperplane with as big a margin as possible.



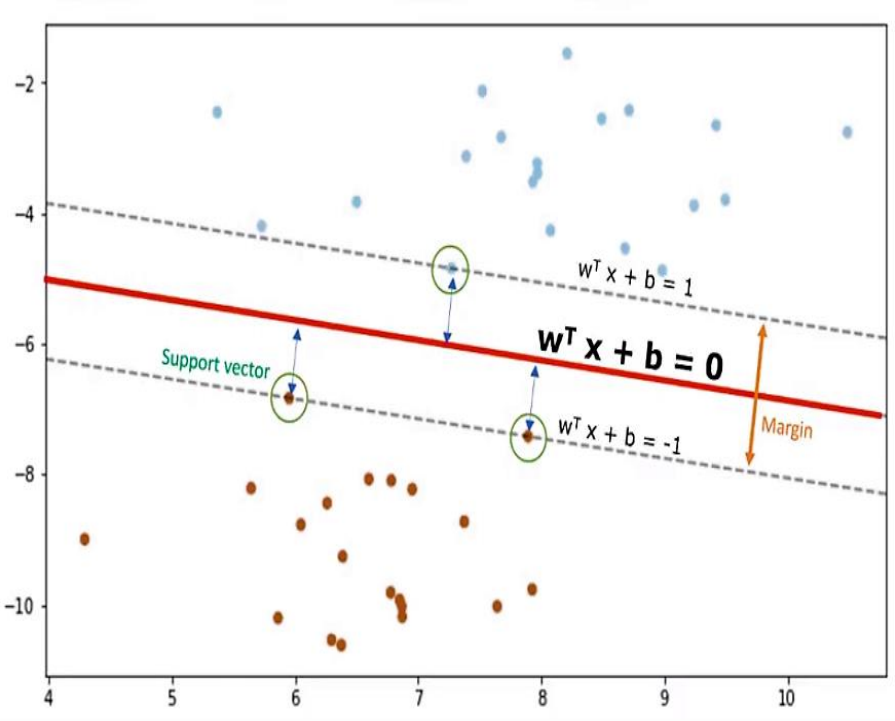
Margins

Only support vectors matter for achieving our goal; and thus, other training examples can be ignored.

We try to find the hyperplane in such a way that it has the maximum distance to support vectors, called optimal Hyperplane.

Hyperplane is learned from training data using an optimization procedure that maximizes the margin.

This optimization problem can be solved by Gradient descent.



Reference

SVM Outcomes

- The output of the algorithm is the values **‘w’ and ‘b’** for the line.
- You can make classifications using this estimated line.
- It is enough to plug in input values into the line equation, then, you can calculate whether an unknown point is above or below the line.
- If the equation returns a value greater than 0, then the point belongs to the first class, which is above the line, and vice versa.

Evaluation Metrics in Classification

- **Evaluation metrics explain the performance of a model.**
- Imagine that we have an historical dataset which shows the customer churn for a telecommunication company.
- We have trained the model, and now we want to calculate its accuracy using the test set.
- We pass the test set to our model, and we find the predicted labels.
- Now the question is, “**How accurate is this model?”**
- Basically, we compare the actual values in the test set with the values predicted by the model, to calculate the accuracy of the model.

Lab 1: Demonstrating Support Vector Machine

SVM Applications

- Image Analysis such as image classification and digit recognition
- Text mining
- Detecting spam
- Text categorization
- Sentiment analysis
- Gene Expression data classification

Summary

- Support Vector Machine (SVM) is a Supervised Machine Learning algorithm that can be used for both regression and classification problem.
- SVM algorithm is based on the concept of ‘decision planes’, where hyperplanes are used to classify a set of given objects.
- SVM kernel projects the non-linearly separable datasets of lower dimensions to linearly separable data of higher dimensions.
- SVM effective in high dimensional spaces.

Quiz

Q1. SVM can be used to solve _____ problems.

a) Classification

b) Regression

c) Both classification and Regression

d) Clustering

Answer: c

Quiz

Q2. Closest Point to the hyper plane are support vectors.

a) True

b) False

c) None

d) Unpredictable

Answer: a

Quiz

Q3. In SVM, if the number of input features is 2, then the hyperplane is?

a) Plane

b) Line

c) Circle

d) None

Answer: b

Quiz

Q4. In SVM, the dimension of the hyperplane depends upon which one?

a) the number of features

b) the number of samples

c) the number of target variables

d) None

Answer: a

Reference

- https://www.spiceworks.com/tech/big-data/articles/what-is-support-vector-machine/
- https://www.analytixlabs.co.in/blog/introduction-support-vector-machine-algorithm/
- https://scikit-learn.org/stable/modules/svm.html
- https://jakevdp.github.io/PythonDataScienceHandbook/05.07-support-vector-machines.html
- https://en.wikipedia.org/wiki/Support_vector_machine
- https://www.baeldung.com/cs/svm-multiclass-classification

Thank you...!