

# Training Day-14 Report:

## Support Vector Machine (SVM) Algorithm

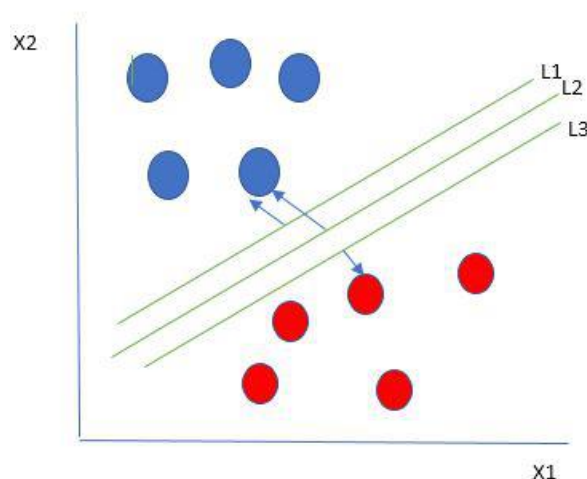
- 

Support Vector Machine (SVM) is a powerful machine learning algorithm used for linear or nonlinear classification, regression, and even outlier detection tasks. SVMs can be used for a variety of tasks, such as text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection. SVMs are adaptable and efficient in a variety of applications because they can manage high-dimensional data and nonlinear relationships.

SVM algorithms are very effective as we try to find the maximum separating hyperplane between the different classes available in the target feature.

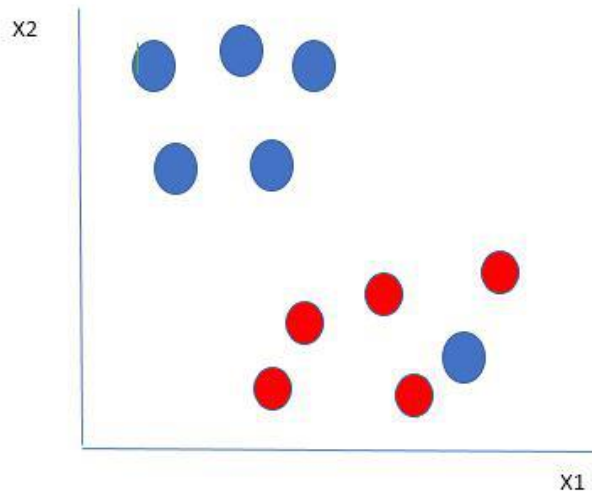
### How does SVM work?

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

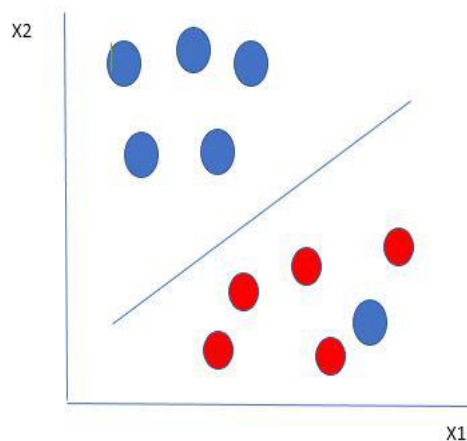


*Multiple hyperplanes separate the data from two classes*

So we choose the hyperplane whose distance from it to the nearest data point on each side is maximized. If such a hyperplane exists it is known as the **maximum-margin hyperplane/hard margin**. So from the above figure, we choose L2. Let's consider a scenario like shown below

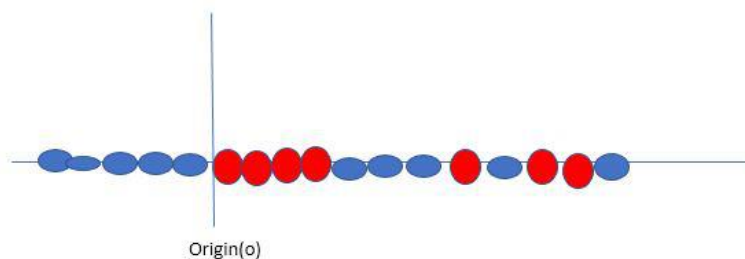


Here we have one blue ball in the boundary of the red ball. So how does SVM classify the data? It's simple! The blue ball in the boundary of red ones is an outlier of blue balls. The SVM algorithm has the characteristics to ignore the outlier and finds the best hyperplane that maximizes the margin. SVM is robust to outliers.

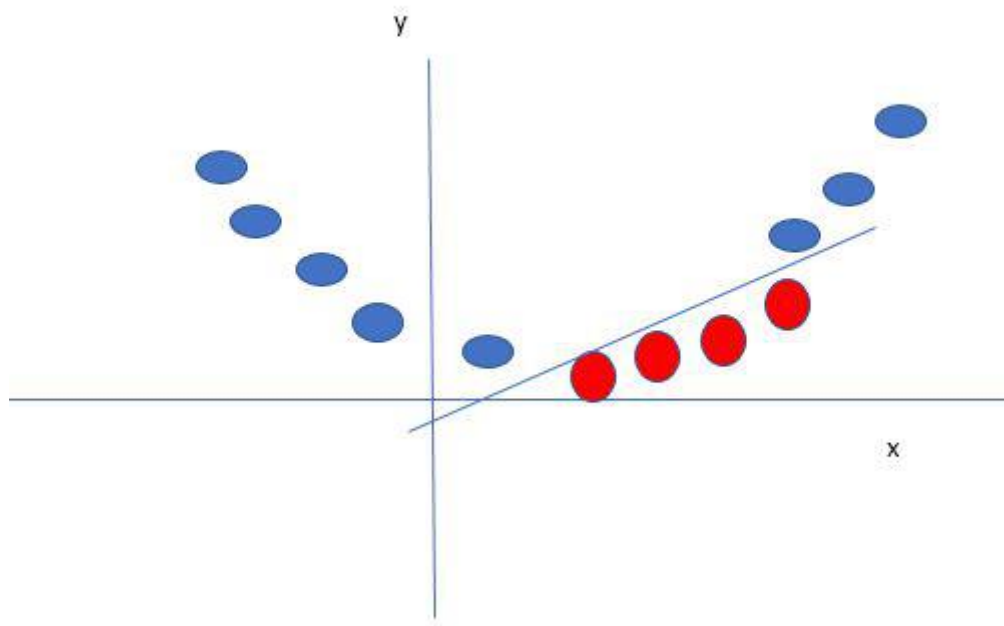


So in this type of data point what SVM does is, finds the maximum margin as done with previous data sets along with that it adds a penalty each time a point crosses the margin. So the margins in these types of cases are called **soft margins**. When there is a soft margin to the data set, the SVM tries to minimize  $(1/\text{margin} + \lambda(\sum \text{penalty}))$ . Hinge loss is a commonly used penalty. If no violations no hinge loss. If violations hinge loss proportional to the distance of violation.

Till now, we were talking about linearly separable data (the group of blue balls and red balls are separable by a straight line/linear line). What to do if data are not linearly separable?



Say, our data is shown in the figure above. SVM solves this by creating a new variable using a **kernel**. We call a point  $x_i$  on the line and we create a new variable  $y_i$  as a function of distance from origin. So if we plot this we get something like as shown below



*Mapping 1D data to 2D to become able to separate the two classes*

In this case, the new variable  $y$  is created as a function of distance from the origin. A non-linear function that creates a new variable is referred to as a kernel.

### Support Vector Machine Terminology

1. **Hyperplane:** Hyperplane is the decision boundary that is used to separate the data points of different classes in a feature space. In the case of linear classifications, it will be a linear equation i.e.  $wx+b=0$ .
2. **Support Vectors:** Support vectors are the closest data points to the hyperplane, which makes a critical role in deciding the hyperplane and margin.
3. **Margin:** Margin is the distance between the support vector and hyperplane. The main objective of the support vector machine algorithm is to maximize the margin. The wider margin indicates better classification performance.
4. **Kernel:** Kernel is the mathematical function, which is used in SVM to map the original input data points into high-dimensional feature spaces, so, that the hyperplane can be easily found out even if the data points are not linearly separable in the original input space. Some of the common kernel functions are linear, polynomial, radial basis function(RBF), and sigmoid.
5. **Hard Margin:** The maximum-margin hyperplane or the hard margin hyperplane is a hyperplane that properly separates the data points of different categories without any misclassifications.
6. **Soft Margin:** When the data is not perfectly separable or contains outliers, SVM permits a soft margin technique. Each data point has a slack variable introduced by the soft-margin SVM formulation, which softens the strict margin requirement and permits certain misclassifications or violations. It discovers a compromise between increasing the margin and reducing violations.
7. **C:** Margin maximisation and misclassification fines are balanced by the regularisation parameter  $C$  in SVM. The penalty for going over the margin or misclassifying data items is decided by it. A

stricter penalty is imposed with a greater value of  $C$ , which results in a smaller margin and perhaps fewer misclassifications.

8. **Hinge Loss:** A typical loss function in SVMs is hinge loss. It punishes incorrect classifications or margin violations. The objective function in SVM is frequently formed by combining it with the regularisation term.
9. **Dual Problem:** A dual Problem of the optimisation problem that requires locating the Lagrange multipliers related to the support vectors can be used to solve SVM. The dual formulation enables the use of kernel tricks and more effective computing.

Optimization:

- **For Hard margin linear SVM classifier:**

The target variable or label for the  $i^{\text{th}}$  training instance is denoted by the symbol  $t_i$  in this statement. And  $t_i = -1$  for negative occurrences (when  $y_i = 0$ ) and  $t_i = 1$  for positive instances (when  $y_i = 1$ ) respectively.

Because we require the decision boundary that satisfy the constraint:

- **For Soft margin linear SVM classifier:**

**Dual Problem:** A dual Problem of the optimisation problem that requires locating the Lagrange multipliers related to the support vectors can be used to solve SVM. The optimal Lagrange multipliers  $\alpha(i)$  that maximize the following dual objective function

where,

- $\alpha_i$  is the Lagrange multiplier associated with the  $i^{\text{th}}$  training sample.
- $K(x_i, x_j)$  is the kernel function that computes the similarity between two samples  $x_i$  and  $x_j$ . It allows SVM to handle nonlinear classification problems by implicitly mapping the samples into a higher-dimensional feature space.
- The term  $\sum \alpha_i$  represents the sum of all Lagrange multipliers.

The SVM decision boundary can be described in terms of these optimal Lagrange multipliers and the support vectors once the dual issue has been solved and the optimal Lagrange multipliers have been discovered. The training samples that have  $\alpha_i > 0$  are the support vectors, while the decision boundary is supplied by:

### Types of Support Vector Machine

Based on the nature of the decision boundary, Support Vector Machines (SVM) can be divided into two main parts:

- **Linear SVM:** Linear SVMs use a linear decision boundary to separate the data points of different classes. When the data can be precisely linearly separated, linear SVMs are very suitable. This means that a single straight line (in 2D) or a hyperplane (in higher dimensions) can entirely divide the data points into their respective classes. A hyperplane that maximizes the margin between the classes is the decision boundary.
- **Non-Linear SVM:** Non-Linear SVM can be used to classify data when it cannot be separated into two classes by a straight line (in the case of 2D). By using kernel functions, nonlinear SVMs can handle nonlinearly separable data. The original input data is transformed by these kernel functions into a higher-dimensional feature space, where the data points can be linearly separated. A linear SVM is used to locate a nonlinear decision boundary in this modified space.

Advantages of SVM

- Effective in high-dimensional cases.
- Its memory is efficient as it uses a subset of training points in the decision function called support vectors.
- Different kernel functions can be specified for the decision functions and its possible to specify custom kernels.

### SVM implementation in Python

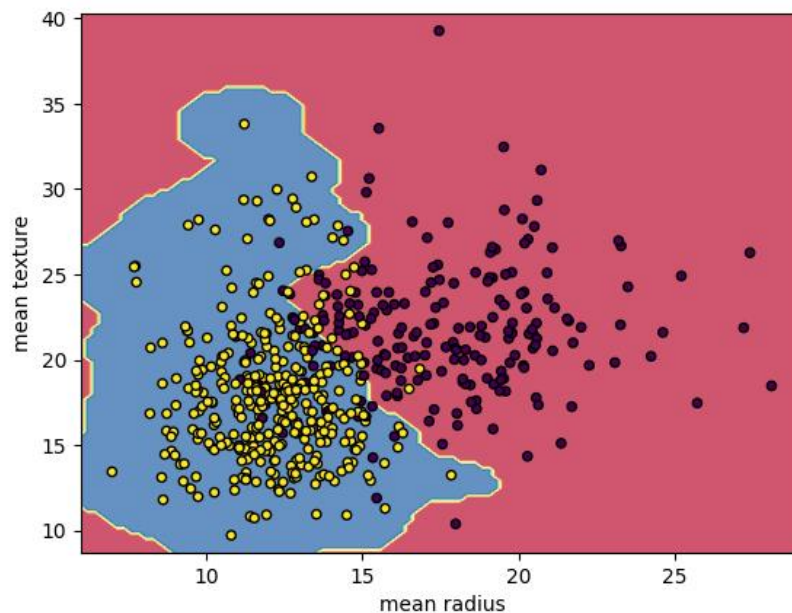
Predict if cancer is Benign or malignant. Using historical data about patients diagnosed with cancer enables doctors to differentiate malignant cases and benign ones are given independent attributes.

### Steps

- Load the breast cancer dataset from sklearn.datasets
- Separate input features and target variables.
- Build and train the SVM classifiers using RBF kernel.
- Plot the scatter plot of the input features.
- Plot the decision boundary.
- Plot the decision boundary

```
# Load the important packages
from sklearn.datasets import load_breast_cancer
import matplotlib.pyplot as plt
from sklearn.inspection import DecisionBoundaryDisplay
from sklearn.svm import SVC
# Load the datasets
cancer = load_breast_cancer()
X = cancer.data[:, :2]
y = cancer.target
#Build the model
svm = SVC(kernel="rbf", gamma=0.5, C=1.0)
Trained the model
svm.fit(X, y)
# Plot Decision Boundary
DecisionBoundaryDisplay.from_estimator(
    svm,
    X,
    response_method="predict",
    cmap=plt.cm.Spectral,
    alpha=0.8,
```

### Output:



*Breast Cancer Classifications with SVM RBF kernel*

### K-Nearest Neighbor(KNN) Algorithm:-

The • K-Nearest Neighbors (KNN) algorithm is a supervised machine learning method employed to tackle classification and regression problems. Evelyn Fix and Joseph Hodges developed this algorithm

in 1951, which was subsequently expanded by Thomas Cover. What is the K-Nearest Neighbors Algorithm? KNN is one of the most basic yet essential classification algorithms in machine learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining, and intrusion detection. It is widely disposable in real-life scenarios since it is non-parametric, meaning it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data). We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute.

Working of KNN algorithm:- The K-Nearest Neighbors (KNN) algorithm operates on the principle of similarity, where it predicts the label or value of a new data point by considering the labels or values of its K nearest neighbors in the training dataset. Step-by-Step explanation of how KNN works is discussed below:

Step 1: Selecting the optimal value of K • K represents the number of nearest neighbors that needs to be considered while making prediction.

Step 2: Calculating distance • To measure the similarity between target and training data points, Euclidean distance is used. Distance is calculated between each of the data points in the dataset and target point.

Step 3: Finding Nearest Neighbors • The k data points with the smallest distances to the target point are the nearest neighbors.

Step 4: Voting for Classification or Taking Average for Regression • In the classification problem, the class labels of are determined by performing majority voting.