Logo

Description automatically generated

Machine Learning

CSE475

**Lab Manual on**

Implementation of

Support Vector Machine using Python

**Md Mohsin Uddin**

Senior Lecturer

Department of Computer Science & Engineering

East West University

**Introduction**

Support Vector Machines (SVMs in short) are machine learning algorithms that are used for classification and regression purposes. SVMs are powerful machine learning algorithms for classification, regression, and outlier detection purposes. An SVM classifier builds a model that assigns new data points to one of the given categories. Thus, it can be viewed as a non-probabilistic binary linear classifier. SVMs can be used for linear classification purposes. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using the kernel trick. It enables us to implicitly map the inputs into high-dimensional feature spaces.

*Now, we should be familiar with some SVM terminology.*

**Hyperplane**

A hyperplane is a decision boundary that separates a given set of data points having different class labels. The SVM classifier separates data points using a hyperplane with the maximum amount of margin. This hyperplane is known as the maximum margin hyperplane and the linear classifier it defines is known as the maximum margin classifier.

**Support Vectors**

Support vectors are the sample data points, which are closest to the hyperplane. These data points will define the separating line or hyperplane better by calculating margins.

**Margin**

A margin is a separation gap between the two lines on the closest data points. It is calculated as the perpendicular distance from the line to support vectors or closest data points. In SVMs, we try to maximize this separation gap so that we get the maximum margin.

**SVM Under the hood**

In SVMs, our main objective is to select a hyperplane with the maximum possible margin between support vectors in the given dataset. SVM searches for the maximum margin hyperplane in the following 2 step process –

* Generate hyperplanes that segregate the classes in the best possible way. There are many hyperplanes that might classify the data. We should look for the best hyperplane that represents the largest separation, or margin, between the two classes.
* So, we choose the hyperplane so that distance from it to the support vectors on each side is maximized. If such a hyperplane exists, it is known as the maximum margin hyperplane and the linear classifier it defines is known as a maximum margin classifier.

**Problem with dispersed datasets**

Sometimes, the sample data points are so dispersed that it is not possible to separate them using a linear hyperplane. In such a situation, SVMs use a kernel trick to transform the input space to a higher dimensional space as shown in the diagram below. It uses a mapping function to transform the 2-D input space into the 3-D input space. Now, we can easily segregate the data points using linear separation.

* **Kernel trick**

In practice, the SVM algorithm is implemented using a kernel. It uses a technique called the kernel trick. In simple words, a kernel is just a function that maps the data to a higher dimension where data is separable. A kernel transforms a low-dimensional input data space into a higher dimensional space. So, it converts non-linear separable problems into linear separable problems by adding more dimensions to them. Thus, the kernel trick helps us to build a more accurate classifier. Hence, it is useful in non-linear separation problems.

***In the context of SVMs, there are 4 popular kernels – Linear kernel, Polynomial kernel, Radial Basis Function (RBF) kernel (also called Gaussian kernel), and Sigmoid kernel.***

**Objective**:

In this lab, we will learn about support vector machines (SVM). After that, we will do a hands-on implementation of SVM in Python using Jupyter notebook and other related libraries such as Pandas, sklearn, NumPy, and matplotlib. In the end, we will generate a report for our produced model and measure it from a different aspect.

**Tasks:**

1. Import necessary libraries
2. Load dataset from sklearn dataset
3. Split the data set into training and testing parts in a moderate ratio ( train\_test\_split )
4. Import SVM model from sklearn
5. Create a model and predict with the testing dataset
6. Create a model prediction report using the sklearn metrics library which will contain

(precision [2] , recall [3] , f1-score[4], accuracy etc.)

***[2] Precision*** *is the ratio of correctly predicted positive observations to the total predicted positive observations.*

***[3] Recall*** *is the ratio of correctly predicted positive observations to all observations in actual class*

***[4] F1 Score*** *is the weighted average of Precision and Recall.*