Experiment No. 10

1. **Aim**

   To understand the Support Vector Machine (SVM) algorithm, its principles, and its application in classification tasks by implementing the algorithm on a sample dataset.

2. **Objectives**

   1. To gain an understanding of the SVM algorithm and its theoretical foundations.
   2. To learn how to construct an SVM model for classification tasks.
   3. To implement an SVM algorithm and interpret its results.
   4. To evaluate the performance of the SVM model on classification problems.

3. **Course Outcomes**

   - **CO6:** Evaluate the performance of classification models and interpret the results.

4. **Hardware / Software Required**

   1. **Software:** Python with libraries such as scikit-learn, NumPy, pandas, Jupyter Notebook or any Python IDE.
   2. **Hardware:** Computer with at least 4GB RAM and sufficient storage to handle datasets.

5. **Theory**

   Support Vector Machine (SVM):

   SVM is a supervised learning algorithm used for classification and regression tasks. It aims to find a hyperplane that best separates the classes in the feature space. The main concepts in SVM include:

   1. **Hyperplane:** A decision boundary that separates different classes in the feature space.
   2. **Margin:** The distance between the hyperplane and the nearest data points from either class. SVM aims to maximize this margin.
   3. **Support Vectors:** Data points that are closest to the hyperplane and influence its position and orientation.

   Mathematical Formulation:

   1. **Linear SVM:**

      For a linearly separable dataset, the goal is to find a hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$ that maximizes the margin. The optimization problem can be formulated as:

      $$\text{Maximize } \frac{2}{\parallel \mathbf{w} \parallel^2}$$

      subject to:

      $$y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 \text{ for all } i$$

      where **w** is the weight vector, b is the bias, and y_i is the class label of $\mathbf{x_i}$.

2. **Non-linear SVM:**

For non-linearly separable data, SVM uses kernel functions to transform the data into a higher-dimensional space where a linear separator can be found. Common kernels include:

- **Polynomial Kernel:** $K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T\mathbf{x}' + c)^d$
- **Radial Basis Function (RBF) Kernel:** $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \parallel \mathbf{x} - \mathbf{x}' \parallel^2)$

6. **Algorithm / Design / Procedure / Flowchart / Analysis**

Algorithm:

1. **Start** with the dataset $\{(\mathbf{x_i}, y_i)\}$.
2. **Choose** a kernel function if the data is not linearly separable.
3. **Formulate** the optimization problem to maximize the margin.
4. **Solve** the optimization problem using methods like quadratic programming.
5. **Obtain** the optimal hyperplane parameters **w** and b.
6. **Classify** new data points using the obtained hyperplane.

Procedure:

1. Load and preprocess the dataset.
2. Select the appropriate kernel function (linear or non-linear).
3. Train the SVM model using the training dataset.
4. Evaluate the model's performance on the test dataset.
5. Visualize the decision boundary and support vectors.

Flowchart:

Start → Load Dataset → Choose Kernel Function → Formulate Optimization Problem → Solve Optimization Problem → Obtain Hyperplane Parameters → Classify Data → **End**

Example Dataset:

Feature 1	Feature 2	Class
2	3	1
3	3	1
6	6	-1
7	7	-1

1. **Construct the Hyperplane:**

   For simplicity, assume we use a linear kernel. The decision boundary is given by:

   $$\mathbf{w}^T\mathbf{x} + b = 0$$

   - Compute the parameters **w** and b using the optimization problem.
   - For this example, let's assume the computed hyperplane is $x_1 - x_2 + 1 = 0$.

2. **Classify New Data Point:**

   To classify a new point $\mathbf{x} = (4,5)$:

$$\mathbf{w}^T\mathbf{x} + b = 4 - 5 + 1 = 0$$

Since the result is zero, the point lies on the decision boundary. For this example, additional calculations are needed to determine the exact class, typically requiring solving the optimization problem.

Analysis:

Evaluate the SVM model using metrics such as accuracy, precision, recall, and F1 score. Discuss how well the SVM model separates the classes and the impact of different kernels.

7. **Results/Output Analysis**

   Present and analyze the results of the SVM classification. Include the accuracy of the model, the decision boundary, and the support vectors. Discuss how the choice of kernel function affects the results.

8. **Conclusions**

   Summarize the key findings of the experiment, including the effectiveness of the SVM algorithm in classifying the data. Discuss how well the experiment met its objectives and any insights gained from the SVM model's performance and decision boundary.

9. **Viva Questions**

   1. What is the main objective of the SVM algorithm in classification tasks?
   2. Explain the concept of margin and its importance in SVM.
   3. How does the SVM algorithm handle non-linearly separable data?
   4. Describe the role of support vectors in an SVM model.
   5. What are the differences between linear and non-linear SVMs?

10. **References**

   1. Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*, 273-297.
   2. Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.