|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | **Hope Foundation’s**  **Finolex Academy of Management and Technology, Ratnagiri** | | | | | |
| **Department of Computer Science and Engineering (AIML)** | | | | | |
| Subject name: Machine Learning | | | | | | Subject Code: CSL604 | |
| Class | | TE CSE | | | Semester –VI (CBCGS) | Academic year: 2024-25 | |
| Name of Student | |  | | | | **QUIZ Score :6** | |
| Roll No | |  | | Experiment No. | | 03 | |
| Title: **Implementation of Support Vector Machine.** | | | | | | | |
|  | | | | | | | |
| **1. Lab objectives applicable:**  **LOB1:**To introduce platforms such as Anaconda, COLAB suitable to Machine Learning.  **LOB2:**To implement various Regression techniques. | | | | | | | |
| **2. Lab outcomes applicable:**  **LO1:** Implement various machine learning models. | | | | | | | |
| **3. Learning Objectives:**   1. To perform classification through Support Vector Regression Techinique.. | | | | | | | |
| **4. Practical applications of the assignment/experiment:**  To obtain hyperplane separating two classes. | | | | | | | |
| **5. Prerequisites**:   1. Python language | | | | | | | |
| **6. Minimum Hardware Requirements**:-  I series processor, RAM 4GB,  **7. Software Requirements:-**  Colab or Visual Studio or Jupyter notebook (Anaconda) | | | | | | | |
| **8. Quiz Questions :** [**https://docs.google.com/forms/d/e/1FAIpQLScprjpoNHgeVinJaDCUOr91cUHcrlqLYTNJays2oWLL5uLP7w/viewform?usp=dialog**](https://docs.google.com/forms/d/e/1FAIpQLScprjpoNHgeVinJaDCUOr91cUHcrlqLYTNJays2oWLL5uLP7w/viewform?usp=dialog) | | | | | | | |
| **9. Experiment/Assignment Evaluation:** | | | | | | | |
| **Sr. No.** | **Parameters** | | | | | **Marks obtained** | **Out of** |
| **1** | Technical Understanding (Assessment may be done based on Q & A **or** any other relevant method.) Teacher should mention the other method used - | | | | | 6 | 6 |
| **2** | Lab Performance | | | | |  | 2 |
| **3** | Punctuality | | | | |  | 2 |
| **Date of performance (DOP)** | | |  | | **Total marks obtained** |  | **10** |

**Signature of Faculty**

**10. Theory:**

Support Vector Machine(**SVM)** is a supervised machine learning algorithm used for classification and regression tasks. It works by finding the optimal **hyperplane** that best separates data points of different classes in a feature space.

**HyperPlane:** A decision boundary that separates data points of different classes.

**Support Vectors**: Data points closest to the hyperplane that influence its position and orientation.

**Margin**: The distance between the hyperplane and the nearest support vectors. SVM maximizes this margin.

The Code attached below shows how a SVM is implemented in python for a synthetic dataset created using **make\_blobs()** function in python which creates two clusters of points (two classes) with slight overlap using “cluster\_std=1.5.”

After creating the required dataset, we train the SVM model.(Here we use linear SVM model).The C parameter balances maximizing margin and minimizing misclassification.

The hyperplane is computed using the weight vector w and intercept b from “*svm.coef\_*” and “*svm.intercept\_*”.

We know that from equation of linear regression,

For visualizing the hyperplane we used:

*plt.scatter(svm.support\_vectors\_[:, 0], svm.support\_vectors\_[:, 1],facecolors='none', edgecolors='k', s=100, label='Support Vectors')* which highlight the support vectors shown in the output.

**11. Installation Steps / Performance Steps and Results –**

import numpy as np

import matplotlib.pyplot as plt

from sklearn.svm import SVC

from sklearn.datasets import make\_blobs

# Generate synthetic data

X, y = make\_blobs(n\_samples=100, centers=2, random\_state =6, cluster\_std=1.5)

# Fit the SVM model

svm = SVC(kernel='linear', C=1)

svm.fit(X, y)

# Get the separating hyperplane

w = svm.coef\_[0]

b = svm.intercept\_[0]

# Calculate slope and intercept for the hyperplane

x0 = np.linspace(min(X[:, 0]), max(X[:, 0]), 100)

x1 = -(w[0] / w[1]) \* x0 - b / w[1]

# Plot the data points and the hyperplane

plt.scatter(X[:, 0], X[:, 1], c=y, cmap='coolwarm', s=50)

plt.plot(x0, x1, 'k-', label='Hyperplane')

# Plot the support vectors

plt.scatter(svm.support\_vectors\_[:, 0], svm.support\_vectors\_[:, 1],facecolors='none', edgecolors='k', s=100, label='Support Vectors')

plt.title("SVM Classification with Hyperplane")

plt.xlabel("Feature 1")

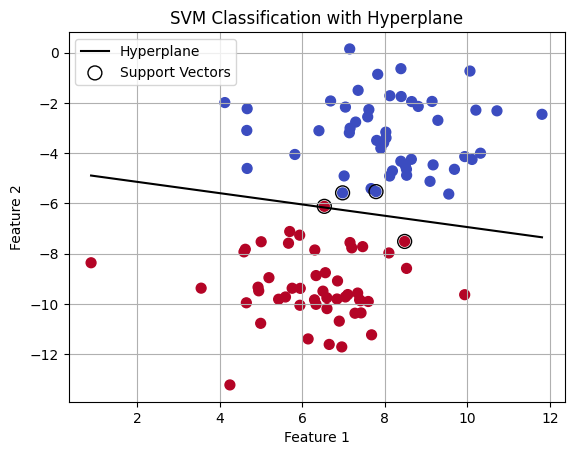
plt.ylabel("Feature 2")

plt.legend()

plt.grid(True)

plt.show()

**OUTPUT:**



**12. Learning Outcomes Achieved**

1. Students are able to perform classification through Support Vector Regression technique.

**13. Conclusion:**

**1. Applications of the Studied Technique in Industry**

SVM is used for classification and regression tasks, offering solutions for disease diagnosis, fraud detection, customer segmentation, and predictive maintenance. Its ability to handle high-dimensional data and deliver accurate results even with limited datasets makes it valuable in both traditional and emerging fields like artificial intelligence, image recognition, and natural language processing.

**2. Engineering Relevance**

SVM holds significant relevance in engineering as it provides an effective tool for pattern recognition, anomaly detection, and optimization problems. In fields such as robotics, control systems, and signal processing, SVM is used to enhance decision-making systems, improve system accuracy, and optimize performance under complex conditions.

**3. Skills Developed**

Through the study and implementation of SVM, individuals develop key skills in machine learning, data analysis, and problem-solving. They gain a deeper understanding of statistical methods, kernel functions, and optimization techniques.

**14. References**:

1. Nathalie Japkowicz & Mohak Shah, ―Evaluating Learning Algorithms: A Classification Perspective‖, Cambridge.
2. Marc Peter Deisenroth, Aldo Faisal, Cheng Soon Ong, ―Mathematics for machine learning‖
3. Samir Roy and Chakraborty, ―Introduction to soft computing‖, Pearson Edition.
4. Ethem Alpaydın, ―Introduction to Machine Learning‖, MIT Press McGraw-Hill Higher

Education

1. Peter Flach, ―Machine Learning‖, Cambridge University Press
2. Tom M. Mitchell, ―Machine Learning‖, McGraw Hill
3. Kevin P. Murphy, ―Machine Learning ― A Probabilistic Perspective‖, MIT Press