Lab Assignment : 02

SVM :

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report, confusion_matrix
iris = load_iris()
X = iris.data[:, :2]
y = (iris.target == 0).astype(int)
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Train SVM model
model = SVC(kernel='linear')
model.fit(x_train, y_train)

# Prediction
y_pred = model.predict(x_test)

# Evaluation
print("Classification Report:\n", classification_report(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))

# Visualization
def plot_hyperplane(X, y, model):
    plt.scatter(X[:, 0], X[:, 1], c=y, cmap='bwr', edgecolors='k')
    ax = plt.gca()
    xlim = ax.get_xlim()
```

```
    ylim = ax.get_ylim()
    xx = np.linspace(xlim[0], xlim[1])
    yy = np.linspace(ylim[0], ylim[1])
    YY, XX = np.meshgrid(yy, xx)
    xy = np.vstack([XX.ravel(), YY.ravel()]).T
    Z = model.decision_function(xy).reshape(XX.shape)
    ax.contour(XX, YY, Z, colors='k', levels=[-1, 0, 1], alpha=0.5,
    linestyles=['--', '-', '--'])
    support_vectors = model.support_vectors_
    ax.scatter(support_vectors[:, 0], support_vectors[:, 1], s=100,
    linewidth=1, facecolors='none', edgecolors='k')

plt.title('SVM Decision Boundary with Support Vectors')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.show()
plot_hyperplane(x_train, y_train, model)
```

Output :

```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        20
           1       1.00      1.00      1.00        10

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30

Confusion Matrix:
 [[20  0]
 [ 0 10]]
```