# **SUPPORT VECTOR MACHINE**

### **Aim**

To implement a Support Vector Machine with Linear Kernel for Classification.

### **Problem description**

This project involves implementing a Support Vector Machine Classifier with Linear Kernel using Iris Dataset.

### **Algorithm**

#### **1) Import the required libraries**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn import metrics

from sklearn.svm import SVC

#### **2) Load the iris dataset**

data = pd.read\_csv("iris.csv")

#### **3) Preprocessing the dataset**

Display the first 5 rows of the dataset and create a new column 'target' based on the species.

#### **4) Train-Test Split**

#### Split the dataset into training and testing sets using 80-20 split ratio, stratified by the 'species' column.

#### **5) Data Visualization:**

#### Create separate DataFrames (df0, df1, df2) for each species and plot scatter plots.

#### **6) Feature and Target Selection:**

#### Prepare the feature matrix (X) and target variable (y) by dropping the 'target' and 'species' columns.

**7)Train-Test Split for Modeling:**

Further split the data into training and testing sets for building and evaluating the SVM model.

**8)SVM Model Training and Evaluation:**

Create a linear SVM model, fit it to the training data, make predictions on the test data, and evaluate its accuracy.

### **Program code/ Pseudocode**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.model\_selection import train\_test\_split

from sklearn import metrics

from sklearn.svm import SVC

data = pd.read\_csv(r"C:\Users\91830\Desktop\DUK\AIML\svm\iris.csv")

data.head(5)

data['target'] = data['species'].map({'Setosa': 0, 'Versicolor': 1, 'Virginica': 2})

data.head(5)

data[data.target==1].head()

data[data.target==2].head()

data.groupby('species').size()

train, test = train\_test\_split(data, test\_size = 0.4, stratify = data['species'], random\_state = 42)

train.groupby('species').size()

df0 = data[:50]

df1 = data[50:100]

df2 = data[100:]

plt.xlabel('Sepal Length')

plt.ylabel('Sepal Width')

plt.scatter(df0['sepal.length'], df0['sepal.width'],color="green",marker='+')

plt.scatter(df1['sepal.length'], df1['sepal.width'],color="blue",marker='.')

plt.xlabel('Petal Length')

plt.ylabel('Petal Width')

plt.scatter(df0['petal.length'], df0['petal.width'],color="green",marker='+')

plt.scatter(df1['petal.length'], df1['petal.width'],color="blue",marker='.')

X = data.drop(['target','species'], axis='columns')

y = data.target

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2)

len(X\_train)

len(X\_test)

linear\_svc = SVC(kernel='linear').fit(X\_train, y\_train)

prediction=linear\_svc.predict(X\_test)

print('The accuracy of the linear SVC is',"{:.3f}".format(metrics.accuracy\_score(prediction,y\_test)\*100),"%")

### **Result**

### The SVM Linear model is successfully implemented with an accuracy of 96.667%.