Practical No.:- 02

**Aim**:-Implement Logistic Regression (Iris Dataset).

# Theory

Linear Regression is a linear approach to modelling the relationship between a scalar response (y dependent variables) and one or more explanatory variables (X — independent variables). I will be exemplifying the use of Linear Regression to predict the sepal length (cm) of a genus of flower called Iris.

# Material

* Numpy
* Sklearn
* Logisticregression

# Program

import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear\_model import LogisticRegression from sklearn import datasets

# import some data to play with iris = datasets.load\_iris()

X = iris.data[:, :2] # we only take the first two features.

Y = iris.target

# Create an instance of Logistic Regression Classifier and fit the data. logreg = LogisticRegression(C=1e5)

logreg.fit(X, Y)

# Plot the decision boundary. For that, we will assign a color to each

# point in the mesh [x\_min, x\_max]x[y\_min, y\_max]. x\_min, x\_max = X[:, 0].min() - 0.5, X[:, 0].max() + 0.5

y\_min, y\_max = X[:, 1].min() - 0.5, X[:, 1].max() + 0.5 h = 0.02 # step size in the mesh

xx, yy = np.meshgrid(np.arange(x\_min, x\_max, h), np.arange(y\_min, y\_max, h)) Z = logreg.predict(np.c\_[xx.ravel(), yy.ravel()])

# Put the result into a color plot Z = Z.reshape(xx.shape) plt.figure(1, figsize=(4, 3))

plt.pcolormesh(xx, yy, Z, cmap=plt.cm.Paired) # Plot also the training points

plt.scatter(X[:, 0], X[:, 1], c=Y, edgecolors="k", cmap=plt.cm.Paired) plt.xlabel("Sepal length")

plt.ylabel("Sepal width") plt.xlim(xx.min(), xx.max())

plt.ylim(yy.min(), yy.max()) plt.xticks(())

plt.yticks(()) plt.show()

# Output



