**Practical Number 2**

**Aim:** Logistic Regression using scikit-learn built-in dataset, Iris Flowers.

**Theory:**

This type of statistical model (also known as logit model) is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn’t vote, based on a given dataset of independent variables.

**Code:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn import datasets

from sklearn.linear\_model import LogisticRegression

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

from sklearn import metrics

# Load the iris dataset

iris = datasets.load\_iris()

# Use only two classes and two features for visualization purposes

X = iris["data"][:, (2, 3)]

y = iris["target"] == 2

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Train the logistic regression model

logreg = LogisticRegression(solver="lbfgs")

logreg.fit(X\_train, y\_train)

# Predict the test set

y\_pred = logreg.predict(X\_test)

# Compute the accuracy of the model

accuracy = metrics.accuracy\_score(y\_test, y\_pred)

print("Accuracy:", accuracy)

# Plot the accuracy, actual values, and predicted outputs

fig, ax = plt.subplots(1, 2, figsize=(12, 4))

# Plot the accuracy

ax[0].scatter(X\_test[:, 0], X\_test[:, 1], c=y\_test, cmap=plt.cm.Paired, edgecolor="black")

ax[0].set\_xlabel("Petal length")

ax[0].set\_ylabel("Petal width")

ax[0].set\_title("Actual values")

# Plot the predicted outputs

ax[1].scatter(X\_test[:, 0], X\_test[:, 1], c=y\_pred, cmap=plt.cm.Paired, edgecolor="black")

ax[1].set\_xlabel("Petal length")

ax[1].set\_ylabel("Petal width")

ax[1].set\_title("Predicted outputs (Accuracy: {:.2f})".format(accuracy))

plt.show()

**Results:**

