**LAB EXERCISE – 11**

**Logistic Regression**

**Aim of the Experiment**

The main aim of this experiment is to explore logistic regression model of scikit-learn. The objectives of this experiment are:

1. Explore random dataset generation for logistic regression.
2. Explore logistic regression model in python for randomly generated dataset

Reference to the Textbook

All the fundamentals are given in Chapter 5 and Appendix 2.

Random dataset for classification model can be as follows:

X, y = make\_blobs(n\_samples=200, centers=3, n\_features=3)

The n\_samples and n\_features can be changed. This has to be imported using the command,

from sklearn.datasets import make\_blobs

Logistic regression model can be created by scikit-learn as

model = LogisticRegression()

The algorithm can be applied to the given data as

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

The predictions of the constructed model can be done as

predicted = model.predict(X\_test)

The classification report can be generated as follows:

report\_lr = classification\_report(y\_test,predicted)

This classification report must be imported as

from sklearn.metrics import classification\_report

**WARNING – Random dataset is used for Listing 1. So, the dataset would be generated at every run. As dataset is generated again, the results would vary every time the program is run.**

**Listing - 1**

import pandas as pd

from sklearn.datasets import make\_blobs

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

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import classification\_report

X, y = make\_blobs(n\_samples=200, centers=3, n\_features=3)

df = pd.DataFrame(dict(x=X[:,0], y=X[:,1], label=y))

# Print the sample top five records

print("Top five Records\n\n")

df\_top = df.head(5)

print(df\_top)

# Condition the input

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,test\_size=0.40,random\_state=0)

# Construct the logistic regression model

model = LogisticRegression()

# Fit the model

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

#Prediction for the test sample

predicted = model.predict(X\_test)

# Print the classification report

print("\n\nClassification Report")

report\_lr = classification\_report(y\_test,predicted)

print(report\_lr)

**Output**

**The top five records of 200 samples is displayed as follows:**

A screen shot of a computer

Description automatically generated

The Classification report generated for this problem is shown as follows:

Screen of a cell phone

Description automatically generated