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

****

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:

```
Top five Records


          x         y  label
0  0.624773 -4.846126      1
1  7.688025 -4.351219      0
2  4.140132  6.958442      2
3  5.603908  5.247835      2
4  4.720044  6.363697      2
```

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

```
Classification Report
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        26
           1       1.00      1.00      1.00        28
           2       1.00      1.00      1.00        26

    accuracy                           1.00        80
   macro avg       1.00      1.00      1.00        80
weighted avg       1.00      1.00      1.00        80
```