Reg. No.: 15BCE0329
Name - Harshit Kedia

Course Code - CSE4020
Slot - L19+L20

Experiment 1

CODE:

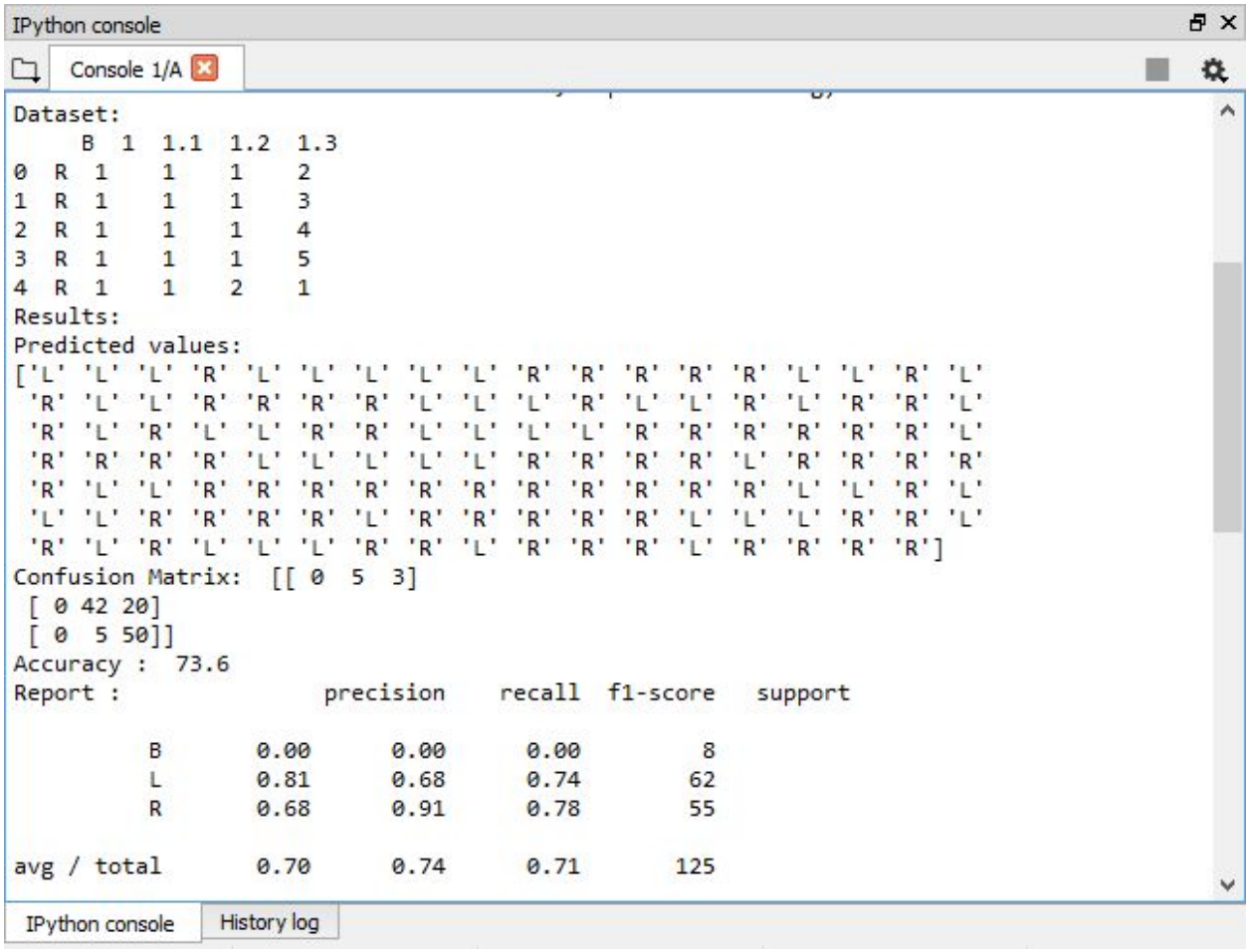
```
from sklearn.metrics import confusion_matrix
from sklearn.cross_validation import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
import numpy as np
import pandas as pd

def importdata():
    balance_data =
pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-'+'databases/balance-scale/balanc
e-scale.data')
    print ("Dataset:\n ",balance_data.head())
    return balance_data

def splitdataset(balance_data):

    X = balance_data.values[:, 1:5]
    Y = balance_data.values[:, 0]
    X_train, X_test, y_train, y_test = train_test_split(
    X, Y, test_size = 0.2, random_state = 0)
    return X, Y, X_train, X_test, y_train, y_test

def train(X_train, X_test, y_train):

    clf = DecisionTreeClassifier(random_state = 100,max_depth=3, min_samples_leaf=5)
    clf.fit(X_train, y_train)
    return clf


def prediction(X_test, clf_object):

    y_pred = clf_object.predict(X_test)
    print("Predicted values:")
```

```
    print(y_pred)
    return y_pred

def cal_accuracy(y_test, y_pred):

    print("Confusion Matrix: ", confusion_matrix(y_test, y_pred))
    print ("Accuracy : ", accuracy_score(y_test,y_pred)*100)
    print("Report : ", classification_report(y_test, y_pred))

# Driver function
def main():
    data = importdata()
    X, Y, X_train, X_test, y_train, y_test = splitdataset(data)
    clf= train(X_train, X_test, y_train)
    print("Results:")
    y_pred= prediction(X_test, clf)
    cal_accuracy(y_test, y_pred)

if __name__=="__main__":
    main()
```

OUTPUT

```
DeprecationWarning: This module was deprecated in version 0.18 in favor of the
model_selection module into which all the refactored classes and functions are moved.
Also note that the interface of the new CV iterators are different from that of this
module. This module will be removed in 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)
Dataset:
     B  1  1.1  1.2  1.3
0  R  1    1    1    2
1  R  1    1    1    3
2  R  1    1    1    4
3  R  1    1    1    5
4  R  1    1    2    1
Results:
Predicted values:
['L' 'L' 'L' 'R' 'L' 'L' 'L' 'L' 'L' 'R' 'R' 'R' 'R' 'R' 'L' 'L' 'R' 'L'
 'R' 'L' 'L' 'R' 'R' 'R' 'R' 'L' 'L' 'L' 'R' 'L' 'L' 'R' 'L' 'R' 'R' 'L'
 'R' 'L' 'R' 'L' 'L' 'R' 'R' 'L' 'L' 'L' 'L' 'R' 'R' 'R' 'R' 'R' 'R' 'L'
 'R' 'R' 'R' 'R' 'L' 'L' 'L' 'L' 'L' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'R'
 'R' 'L' 'L' 'R' 'R' 'R' 'R' 'R' 'R' 'R' 'R' 'R' 'R' 'R' 'L' 'L' 'R' 'L'
 'L' 'L' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'R' 'R' 'L' 'L' 'L' 'R' 'R' 'L'
 'R' 'L' 'R' 'L' 'L' 'L' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'R']
Confusion Matrix:  [[ 0  5  3]
 [ 0 42 20]
 [ 0  5 50]]
Accuracy :  73.6
```

IPython console | History log


IPython console
Console 1/A
Dataset:
B 1 1.1 1.2 1.3
0 R 1 1 1 2
1 R 1 1 1 3
2 R 1 1 1 4
3 R 1 1 1 5
4 R 1 1 2 1
Results:
Predicted values:
['L' 'L' 'L' 'R' 'L' 'L' 'L' 'L' 'L' 'R' 'R' 'R' 'R' 'R' 'L' 'L' 'R' 'L'
'R' 'L' 'L' 'R' 'R' 'R' 'R' 'L' 'L' 'L' 'R' 'L' 'L' 'R' 'L' 'R' 'R' 'L'
'R' 'L' 'R' 'L' 'L' 'R' 'R' 'L' 'L' 'L' 'L' 'R' 'R' 'R' 'R' 'R' 'R' 'L'
'R' 'R' 'R' 'R' 'L' 'L' 'L' 'L' 'L' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'R'
'R' 'L' 'L' 'R' 'R' 'R' 'R' 'R' 'R' 'R' 'R' 'R' 'R' 'R' 'L' 'L' 'R' 'L'
'L' 'L' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'R' 'R' 'L' 'L' 'L' 'R' 'R' 'L'
'R' 'L' 'R' 'L' 'L' 'L' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'R']
Confusion Matrix: [[0 5 3]
[0 42 20]
[0 5 50]]
Accuracy : 73.6
Report : precision recall f1-score support
B 0.00 0.00 0.00 8
L 0.81 0.68 0.74 62
R 0.68 0.91 0.78 55
avg / total 0.70 0.74 0.71 125
IPython console
History log

\