Assignment-3

Name: Shradha Agarwal

Enrolment number: MT20123

Data Preparation: -

Deleted the columns of ID, Gender, DOB, 10th board, 12th board, college ID, college city ID, college city tier and college state; as these attributes have little/no effect on deciding whether a person will get high income/low income salary.

Converted college tier, degree and specialisation to categorical codes.

Shuffled the dataset.

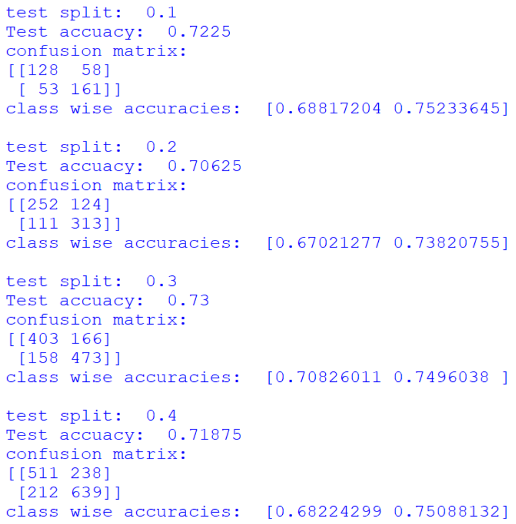
Experiments: -

Performed Logistic regression on the training set and prediction on test set.

Calculated and displayed accuracy, class wise accuracy and the confusion matrix.

Tried on several values of test set like 0.1, 0.2, 0.3, 0.4.

Result: -



```
test split:  0.1
Test accuacy:  0.7225
confusion matrix:
[[128  58]
 [ 53 161]]
class wise accuracies:  [0.68817204 0.75233645]

test split:  0.2
Test accuacy:  0.70625
confusion matrix:
[[252 124]
 [111 313]]
class wise accuracies:  [0.67021277 0.73820755]

test split:  0.3
Test accuacy:  0.73
confusion matrix:
[[403 166]
 [158 473]]
class wise accuracies:  [0.70826011 0.7496038 ]

test split:  0.4
Test accuacy:  0.71875
confusion matrix:
[[511 238]
 [212 639]]
class wise accuracies:  [0.68224299 0.75088132]
```

Result analysis: -

It can be observed that maximum test accuracy, 73%, is achieved with test size 30%. Also, for test size 30%, the class wise accuracies are similar for class 0 (70.826%) and class 1 (74.96%). Thus, this is a better split than 10%, 20% and 40%. The least test accuracy obtained is 70.625% for test size 20%.

Code: -

from sklearn.linear_model import LogisticRegression

from sklearn.utils import shuffle

from sklearn.metrics import confusion_matrix

```
from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score

import numpy as np

import pandas as pd

import joblib


def preprocess():

data=pd.read_csv('C:/Users/Admin/Desktop/Sem1/AI/Assignment_3_graded/ShradhaAgarwalDataAssignment3.csv.csv',skiprows=[0],header=None)

   label = data.iloc[:,33].copy()

   data.drop([0,1,2,4,7,8,13,14,15,33],axis=1,inplace=True)

   data[9] = data[9].astype('category').cat.codes

   data[10] = data[10].astype('category').cat.codes

   data[11] = data[11].astype('category').cat.codes

   #print(data.head())

   #print(data.dtypes)

   data = data.to_numpy()

   label = label.to_numpy()

   data, label = shuffle(data, label,random_state=5)

   return data,label


def data_split(data,label, amnt_test):

   x_train,x_test,y_train,y_test = train_test_split(data,label, test_size=amnt_test, random_state=5)

   return x_train,x_test,y_train,y_test


def logistic_reg(x_train,x_test,y_train,y_test,test_size):

   #result = LogisticRegression(max_iter=10000).fit(x_train,y_train)

   #joblib.dump(result, 'C:/Users/Admin/Desktop/Sem1/AI/Assignment_3_graded/'+'LogisticRegression'+str(test_size)+'.pkl')
```

```
    loaded_model = joblib.load('C:/Users/Admin/Desktop/Sem1/AI/Assignment_3_graded/'+'LogisticRegression'+str(test_size)+'.pkl')
    y_pred_test = loaded_model.predict(x_test)
    print("")
    print("test split: ",test_size)
    acc=accuracy_score(y_test,y_pred_test)
    print("Test accuacy: ",acc)
    confusion_mat = confusion_matrix(y_test, y_pred_test)
    print("confusion matrix: ")
    print(confusion_mat)
    confusion_mat=confusion_mat.astype('float')/confusion_mat.sum(axis=1)[:, np.newaxis]
    print("class wise accuracies: ",confusion_mat.diagonal())


data,label=preprocess()
test_size = [0.1,0.2,0.3,0.4]
for i in test_size:
    x_train,x_test,y_train,y_test= data_split(data,label,i)
    logistic_reg(x_train,x_test,y_train,y_test,i)
```