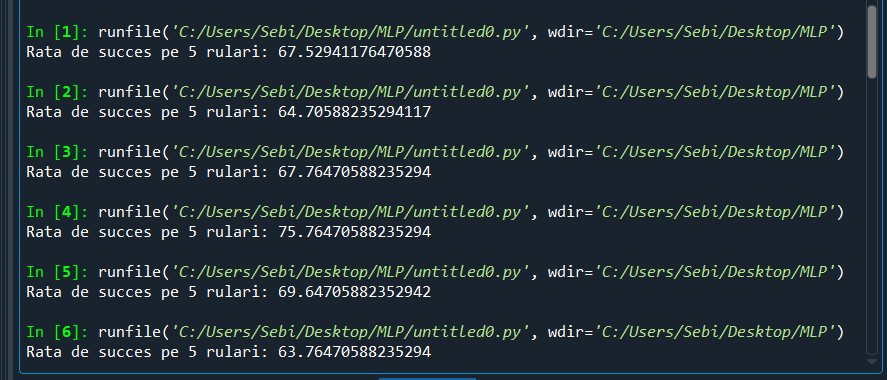
Cândea Sebastian Constantin -424A

Raport Proiect

Problema care mi-a fost atribuită este una de clasificare de tip MLP cu variabile lipsa. Baza de date este formată din 336 de exemple, cu 18 atribute si 22 de clasificări numerotate de la 1-22 in data.txt. Baza de date a fost puțin modificată in program și anume am șters 2 coloane si am rămas in 16 atribute deoarece prea multe exemple aveau date lipsă. Pentru a avea un amestec cât mai omogen am folosit funcția ,, random.shuffle(data)” din biblioteca random, apoi am împărțit in 75% -date training și 25% date test. Pentru aplicarea algoritmului MPL am folosit biblioteca from sklearn import neural\_network.

Rezultate:

* Learing rate de 0.1 cu un strat de 16 generează o medie pe 5 rulări de 75.7647% acuratețe
* Learing rate de 0.1 cu un strat de 8 generează o medie pe 5 rulări de 65.4117% acuratețe
* Learing rate de 0.01 cu un strat de 16 generează o medie pe 5 rulări de 70.11764% acuratețe
* Learing rate de 0.01 cu un strat de 8 generează o medie pe 5 rulări de 69.64705% acuratețe
* Learing rate de 0.1 cu două straturi de 16 generează o medie pe 5 rulări de 57.88235% acuratețe
* Learing rate de 0.01 cu două straturi de 16 generează o medie pe 5 rulări de 68.0% acuratețe
* Learing rate de 0.1 cu un strat de 16 și unul de 8 generează o medie pe 5 rulări de 58.82352% acuratețe
* Learing rate de 0.01 cu un strat de 16 și unul de 8 generează o medie pe 5 rulări de 69.8823 % acuratețe
* Learing rate de 0.1 cu două straturi de 8 generează o medie pe 5 rulări de 61.411764% acuratețe
* Learing rate de 0.01 cu două straturi de 8 generează o medie pe 5 rulări de 67.05882% acuratețe



Codul:

from sklearn import neural\_network

import numpy as np

import random

data = np.loadtxt("data.txt", dtype='i', delimiter=',')

random.shuffle(data)

a=np.array(data[:,1:])

b=np.array(data[:,0])

np.delete(a,2,1)

np.delete(a,3,1)

date\_test=a[110:195]

etichete\_test=b[110:195]

date\_train=np.vstack((a[:109],a[196:]))

etichete\_train=np.append(b[:109],b[196:])

#print(etichete\_test)

#75% din 336=254

def procent(x,y):

nr=0

for i in range (len(etichete\_test)): #84

if x[i]==y[i]:

nr=nr+1;

return 100\*nr/len(etichete\_test)#85

s=0

for i in range(5):

clf=neural\_network.MLPClassifier(hidden\_layer\_sizes=(16),learning\_rate\_init=0.01,max\_iter=1000)

#antrenare

clf.fit(date\_train,etichete\_train)

#predictii

predictii=clf.predict(date\_test)

#rata de succes

a=procent(predictii,etichete\_test)

#print("Rata",i+1,":",a)

s=s+a

print("Rata de succes pe 5 rulari:",s/5)