

ДОДАТОК А
Класифікатор

Загрузка датасету:

```
import requests
target_csv_path = "news.csv"
response = requests.get(download_url)
response.raise_for_status() #Check that the request was successful
with open(target_csv_path, "wb") as f:
    f.write(response.content)
print("Download ready.")
```

Тренування класифікатора:

```
def main_trains(wt,alpha,data,epoch,epochnow):
    cl_A = 0
    cl_B = 0
    cl_C = 0
    for g in range (len(wt)):
        for w in range(len(wt[g])-1):
            cl_A += (data[w]-wt[0][w])
            cl_B += (data[w]-wt[1][w])
            cl_C += (data[w]-wt[2][w])
            w+=1
        g+=1
    xw1= mt.sqrt(cl_A**2)
    xw2 = mt.sqrt(cl_B**2)
    xw3 = mt.sqrt(cl_C**2)
    res = np.array([xw1,xw2,xw3])
    val = np.amin(res)
    if (val == xw1):
        win = "w0"
    winner = wt[0]
```

```

elif (val == xw2):
win = "w1"
winner = wt[1]
elif (val == xw3):
win = "w2"
winner = wt[2]

runnerup = []
for g in range(len(res)):
if (res[g] != val):
runnerup.append(res[g])
g += 1
runnerup2 = np.asarray(runnerup)
rval = np.amin(runnerup2)
if (rval == xw1):
rn = "w0"
runner = wt[0]
elif (rval == xw2):
rn = "w1"
runner = wt[1]
elif (rval == xw3):
rn = "w2"
runner = wt[2]
res2 = np.array([(val/rval),(rval/val)])
class_data = data[4]
win_class = winner[4]
runner_up_class = runner[4]
if (np.amin(res2) > ((1 - 0.2)*(1 + 0.2))):#LVQ3.0
print("CHANGE!")
if ((class_data == win_class) or (class_data == runner_up_class)):

```

```

for w in range(len(winner) - 1):
winner[w]=winner[w] + alpha*(data[w] - winner[w])
runner[w]=runner[w] - alpha*(data[w] - runner[w])
elif ((class_data == win_class) and (class_data ==
runner_up_class)):
m = 0.3
beta = m * alpha
for w in range(len(winner) - 1):
winner[w]=winner[w] + beta*(data[w] - winner[w])
runner[w]=runner[w] + beta*(data[w] - runner[w])
elif(np.amin(res2)>(1-0.2)and np.amax(res2)<1+0.2):#LVQ2.1
for w in range(len(winner) - 1):
winner[w] = winner[w] + alpha * (data[w] - winner[w])
runner[w] = runner[w] - alpha * (data[w] - runner[w])
elif(winner[4]!=runner[4] and data[4]==runner[4] and (val-
rval<=0.01)):#LVQ2
for w in range(len(winner) - 1):
winner[w] = winner[w] + alpha * (data[w] - winner[w])
runner[w] = runner[w] - alpha * (data[w] - runner[w])

else:#LVQ1
for w in range(len(winner) - 1):
winner[w] = winner[w] + alpha * (data[w] - winner[w])
if (win == "w0"):
wt[0] = winner
elif ( win == "w1"):
wt[1] = winner
elif (win == "w2"):
wt[2] = winner

```

```

if (rn == "w0"):
wt[0]=runner
elif ( rn == "w1"):
wt[1] = runner
elif (rn == "w2"):
wt[2] = runner

alpha = alpha * (1 - (epochnow / epoch))
return wt,alpha

```

```

def test_data(wt,data):
    comparison_mat = []
    cl_A = 0
    cl_B = 0
    cl_C = 0
    for q in range(len(data)):
        for g in range(len(wt)):
            for w in range(len(wt[g]) - 1):
                cl_A += (data[q][w] - wt[0][w])
                cl_B += (data[q][w] - wt[1][w])
                cl_C += (data[q][w] - wt[2][w])
            w += 1
        g += 1
    plant_class = data[q][4]
    xw1 = mt.sqrt(cl_A ** 2)
    xw2 = mt.sqrt(cl_B ** 2)
    xw3 = mt.sqrt(cl_C ** 2)
    res = np.array([xw1, xw2, xw3])
    val = np.amin(res)
    if(val == xw1):

```

```

tanaman_type = "IRIS SETOSA"
winner = wt[0]
elif (val == xw2):
tanaman_type = "IRIS VESICOLOR"
winner = wt[1]
elif (val == xw3):
tanaman_type = "IRIS VIRGINICA"
winner = wt[2]
print("Plant Type " + str(q + 1)+" = "+tanaman_type+".")
result = [plant_class,winner[4]]
comparison_mat.append(result)
q++1
right=0
wrong=0
for x in range(len(comparison_mat)):
if(comparison_mat[x][0]==comparison_mat[x][1]):
right+=1
else:
wrong+=1
print("Models Accuracy = "+ str((float(right)/(right+wrong))*100)
+"%")

```

```

def main_train():
alpha = 0.1
epoch = 10
for g in range(epoch):
if (alpha > 0.00001):
print("training at epoch - " + str(g + 1))
for d in range(len(data)):
wa=main_trains(weight,alpha, data[d] , epoch ,(g+1))

```

```
alpha = wa[1]
d ++
else:
break
g ++
weight_new= wa[0]
print("Last updated weight = \n" + str(weight[0]))
print("Last updated alpha = "+str(alpha))
return weight_new
weight=main_train()
test_data(weight, data_uji)
```

ДОДАТОК Б

Відомість атестаційної роботи магістра

