ARTIFICIAL INTELLIGENCE

(UE17CS325)

Guided By:

Prof. K. S. Srinivas

Compiled By:

1. Saioni Chatterjee (PES1201700118)
2. Shivangi Raj (PES1201700098)
3. Bhargavi Priyasha Kumar(PES1201701802)

**INTRODUCTION**

The program comprises three algorithmic functions: Simulated Annealing for global search optimization, and Hill Climbing and Genetic Algorithm for local search optimization.

**DESCRIPTION**

The program begins by passing the dataset to all the algorithmic functions. The algorithmic function which returns a higher accuracy will be chosen by the program and used henceforward.

The pseudo-code of the implementation of the algorithms and how the three have been compared on the basis of accuracy is hereinafter.

df\_results = []

acc1 = random\_hill\_climb(iters)

acc2 = simulated\_annealing(iters)

acc3 = genetic\_alg(iters)

df\_results.append(['random\_hill\_climb',acc1[1]])

df\_results.append(['simulated\_annealing',acc2[1]])

df\_results.append(['genetic\_alg',acc3[1]])

df\_results=sorted(df\_results,key=lambda x:x[1])

max\_acc\_algo=df\_results[0][0]

max\_acc\_value=df\_results[0][1]

if max\_acc\_algo == 'random\_hill\_climb':

    model = 1

elif max\_acc\_algo == 'simulated\_annealing':

    model = 2

else:

    model = 3

acc\_val = max\_acc\_value

The program aspires to achieve a minimum accuracy of 92% before reaching a maximum iterations of 200.

After every iteration, the maximum iteration limit per algorithm run is increased by 50 in order to attain a greater scope of improving accuracy.

i = 1

while (i < 2 and acc\_val < 95 ):

    iters = iters + 150

    if model == 1:

        y\_test\_accuracy = random\_hill\_climb(iters)

    elif model ==2:

        y\_test\_accuracy = simulated\_annealing(iters)

    else:

        y\_test\_accuracy = genetic\_alg(iters)

    if(y\_test\_accuracy[1] \* 100 == acc\_val):

        i+=1

    else:

        i=0

    acc\_val = y\_test\_accuracy[1] \* 100

Eventually, the code prints the final accuracy achieved, confusion matrix and the time taken to achieve this accuracy. With random state value of 156, the final accuracy achieved is 92 per cent in a timespan of 3.38 seconds.

print("The final accuracy is:",round(acc\_val,4))

print("Total iterations:",iters)

print("Execution time in seconds = ",datetime.now()-startTime,"\n")

**CONCLUSION**

The code is implemented in order to always give the best and most optimal accuracy that can be calculated from the implemented algorithms as output. Once the better of the three algorithms is established, the code ensures that this specific algorithm is implemented.