100 feet Ring Road, BSK 3rd Stage, Bengaluru

560085

UE17CS325

Artificial Intelligence

Project Report

PES1201700121 Rahul M Koushik

PES1201700099 Rishabh K Jain

PES1201701554 Abhaay S

Problem Statement:

Using a set of strategies to solve the given problem with high accuracy and in a given time frame, with the constraints being, not using the method of back-propagation to solve the same.

Our approach:

We tried a number of different approaches to attain the required result. In some eagle strategy was used. We also used decision trees to solve the same, but could attain the required result with higher accuracy in lesser time. Here are the results.

1)Mlrose-

In this we used simulated annealing algorithm from the imported mlrose library and ran it on the dataset. We got the following results -

Confusion Matrix:

[[10 0]

[ 0 40]]

Accuracy : 100.0

Report : precision recall f1-score support

0 1.00 1.00 1.00 10

1 1.00 1.00 1.00 40

accuracy 1.00 50

macro avg 1.00 1.00 1.00 50

weighted avg 1.00 1.00 1.00 50

Execution time in seconds = 0:00:02.286481

2)Decision trees-

Here we use Decision tree classifier and train our model using gini algorithm and entropy and test it on our dataset and get the following result-

Results Using Gini Index:

Confusion Matrix:

[[10 0]

[ 0 40]]

Accuracy : 100.0

Report : precision recall f1-score support

0 1.00 1.00 1.00 10

1 1.00 1.00 1.00 40

accuracy 1.00 50

macro avg 1.00 1.00 1.00 50

weighted avg 1.00 1.00 1.00 50

Results Using Entropy:

Confusion Matrix:

[[10 0]

[ 0 40]]

Accuracy : 100.0

Report : precision recall f1-score support

0 1.00 1.00 1.00 10

1 1.00 1.00 1.00 40

accuracy 1.00 50

macro avg 1.00 1.00 1.00 50

weighted avg 1.00 1.00 1.00 50

Execution time in seconds = 0:00:00.534451

3) Simulated Annealing

We have designed our own simulated annealing algorithm to update the weights

of our neural network. In order to update weights, we search for neighbors of the

current weight matrix and using the neighbor, we again run our neural network

through the dataset once to get the accuracy of the model with the new set of

weights. Based on the temperature and probability, we decide if we take the new

weights.

Confusion Matrix:

[[ 8 2]

[ 0 40]]

Accuracy of training : 0.9133333333333333

Accuracy of testing : 0.96

Execution time in seconds = 0:00:25.302744