

Que1.

Script File: go_knn.py

Approach: Divided the outer for loop of function go_knn.py into separate chunks and calculated the accuracy based on that.

Time taken by single and multiprocessor and with accuracy obtained:

Multicore Accuracy 0.794000 Time 49.834857 secs.

Single Core Accuracy 0.794000 Time 189.445583 secs.

Que2.

Script File: Part2Que2.py

Approach: Divide the training data into 4 chunks and then train the model on the subset data and add and take mean of all the accuracies obtained.

Time taken by single and multiprocessor and with accuracy obtained:

For Single processor

('Prediction Accuracy is', 75.3)

('time passed', 1.3690872192382812, 'Seconds')

Multicore Accuracy 75.300000 Time 0.493723 secs.

Discussion:

Why did we use to compute the testing time on Q1 but training time on Q2

Solution:

As in go_knn algorithm we are computing the distance of the points in test set to all the nearest point in training set and just taking into consideration the points nearest to the the test set points. So if in case we decrease the points in test set where we need to compute distance with all training points we can end up in reducing the computation time of the entire program.

As in case of logistic regression algorithm we are computing the gradient on the training data to just training data requires multiprocessing to decrease the computation time of gradient.