Libraries in project1 package:

genetic

IOfunctions

library

regression

spherical

structure

Other packages:

numpy

matplotlib.pyplot

os

re

datetime

shutil

glob

sys

random

time

copy

pandas

sklearn

pickle

scipy

statsmodels

Short description of searching algorithm (works after genetic)

Main program

while number of features > 1:

chromosome = BestFit2 # best fit for each number of features

append chromosome to list

RemoveWorstGene from chromosome

BestFit2

best\_nodes = A\_star # returns list of improved nodes with constant number of features

best\_node = best from best\_nodes

**Greedy Tree Traversal Algorithm**

NodeRoot = GA.Evolve()

BestNode = NodeRoot

WaitingList.InsertAndSort(Node=NodeRoot, Level=0)

until WaitingList.IsEmpty() {

CurrentNode = WaitingList.PullTheBestNode()

NodeLevel = GetLevel(CurrentNode)

if History.IsAlreadyVisited(CurrentNode) {

Cycle 🡪 ZZZ go to the next iteration of the loop

}

History.StoreVisited(CurrentNode)

if IsTimeToExit() {

return BestNode

}

BestRMSE = min(BestRMSE,CurrentNode.RMSE)

BestRMSE[NodeLevel]= min(BestRMSE[NodeLevel],CurrentNode.RMSE)

if ( CurrentNode.RMSE > BestRMSE[NodeLevel] ) {

continue loop ignoring this node

}

for (I in CurrentNode.Features) {

for (J in SortedCorrelatedList(I)) {

ChildNode = CurrentNode.ReplaceFeature(FeatureToReplace = I, NewFeature = J )

if History.IsAlreadyVisited(ChildNode) {

Cycle 🡪 ZZZ go to the next iteration of the J loop

}

If (BestRMSE[NodeLevel+1] > ChildNode.RMSE)){

WaitingList.InsertAndSort(Node=ChildNode, Level=NodeLevel+1)

}

}

}

}