# Map 1-based optional input ports to variables

dataset1 <- maml.mapInputPort(1) # class: data.frame

dataset2 <- maml.mapInputPort(2) # class: data.frame

library(dplyr)

library(tidyr)

testdata<-as.vector(t(dataset2))

gdata\_coll<-dataset1

gdata\_coll<-gdata\_coll[,c(2,3)]

gdata\_coll<-unique(gdata\_coll)

##Assign 1s if particular user has purchased the particular product.

gdata\_coll$purchase<-1

g\_matrix<-spread(gdata\_coll,product\_name,purchase,fill=0)

g\_matrix<-g\_matrix[,-c(1)]

g\_matrix<-g\_matrix[,c(1:3)]

##Now to compute the cosine similarity between sets of items

##A function to calculate Cosine similarity- Source: http://www.salemmarafi.com/code/collaborative-filtering-r/

getcossim<-function(x,y){

cosim<-(x%\*%y)/(sqrt(sum(x))\*sqrt(sum(y)))

return(cosim)

}

##Create a dummy matrix that will extract the similar products from this product list

phmatrix<-matrix(NA,

nrow=ncol(g\_matrix),

ncol=ncol(g\_matrix),

dimnames=list(colnames(g\_matrix),colnames(g\_matrix)))

##Fill up this dummy matrix with the similarities of the user products

for (i in 1:ncol(phmatrix))

{

for (j in 1:ncol(phmatrix))

{

if (sum(as.vector(g\_matrix[,i]))==0|sum(as.vector(g\_matrix[,j]))==0){

phmatrix[i,j]<-0

} else {

phmatrix[i,j]<-getcossim(as.vector(g\_matrix[,i]),as.vector(g\_matrix[,j]))

}

}

}

##Convert Matrix output to Data Frame

sim\_matrix<-as.data.frame(phmatrix)

##Get the list of 5 closest items in this list

closem<-matrix(NA, nrow=nrow(sim\_matrix),ncol=3,dimnames=list(colnames(sim\_matrix),

c("Item0","Item1","Item2")))

for (i in 1:nrow(closem))

{

closem[i,]<-(t(head(n=3,rownames(sim\_matrix[order(sim\_matrix[,i],decreasing=TRUE),][i]))))

}

closem<-as.data.frame(closem)

closem$Item0<-rownames(closem)

cflist<-closem[which(rownames(closem)%in%testdata),]

data.set<-cflist

maml.mapOutputPort("data.set");