**PROJECT CODE**

**-------START OF APRIORI CODE------**

p<-read.csv(file.choose())//Choose our dataset

df<-data.frame(p)//it is converted into dataframe

df$N<-cut(df$N,br=c(0,40,80,100,200,400),labels=c("low","bavg","avg","aavg","high"))//a

df$P<-cut(df$P,br=c(0,40,80,100,200,400),labels=c("low","bavg","avg","aavg","high"))//b

df$K<-cut(df$K,br=c(0,40,80,100,150,200),labels=c("low","bavg","avg","aavg","high"))//c

df$area<-cut(df$area,br=c(0,5,10),labels=c("less","more"))//d

df$rainfall<-cut(df$rainfall,br=c(0,50,100,150,200),labels=c("low","avg","aavg","high"))//e

df$temperature<-cut(df$temperature,br=c(0,20,25,30,35,50),labels=c("low","bavg","avg","aavg","high"))//f

df$production<-cut(df$production,br=c(0,200,400,600,800,1000),labels=c("low","bavg","avg","aavg","high"))//g

df$distance<-cut(df$distance,br=c(0,100,200,300,400,500),labels=c("low","bavg","avg","aavg","high"))//h

//in a,b,c,d,e,f,g,h we are discretizing all the numeric attributes.

Library(arules)

rules1<-apriori(df,parameter=list(minlen=2,maxlen=20,support=0.05,confidence=0.1),appearance=list(rhs=c("prediction.of.profit.loss=H"),default="lhs"))//we are using apriori function,here df is the datset,in appearance we are fixing the RHS so that we will get rules only having highest profits.

rules1

inspect(rules1)//for viewing the rules

inspect(subset(rules1,subset=lhs %pin% "crop="))//

from the rules obtained we are again taking into consideration which have crop attribute as a subset in them.

------END OF APRIORI CODE------

--------Start of clustering code-----------

data<-read.csv(file.choose())//choosing the dataset

df<-data[,c(3,4,5,8,12,13,14)]//consider only numerical attributes

result <- kmeans(df,4);//applying kmeans,df=dataset,4 clusters

result//viewing the row number with their cluster number

plot(df[c("area","production")],col=result$cluster)//

graphical representation of the clusters

result$cluster

legend("topleft",inset=0.01,legend=c("cluster1","cluster2","cluster3","cluster4"),col=(1:4),lty=1,horiz=F)//

keeping legend for the plot of the clusters obtained

library("ggpubr")

ggscatter(df, x = "area", y = "production",

add = "reg.line", conf.int = TRUE,

cor.coef = TRUE, cor.method = "pearson",

xlab = "area(in hectares)", ylab = "production")//scatter plot for area and production using pearson correlation method

data1<-data//taking the dataset into data1

data1<-myfunction(data1,"Rabi","Rice",90,70,50,"NULL","CFL",15000,"T","canal irrigation","black",150,35,10000,"M")//taking inputs

myfunction<-function(da,a,b,c,d,e,f,g,h,i,j,k,m,n,q,s){

l<-nrow(da)//l has no of rows of dataset

da[l+1,1]<-a

da[l+1,2]<-b

da[l+1,3]<-c

da[l+1,4]<-d

da[l+1,5]<-e

da[l+1,6]<-f

da[l+1,7]<-g

da[l+1,8]<-h

da[l+1,9]<-i

da[l+1,10]<-j

da[l+1,11]<-k

da[l+1,12]<-m

da[l+1,13]<-n

da[l+1,14]<-q

da[l+1,15]<-s//inserting those attributes

return(da)//return the dataset with the added tuples

}

g1<-data1[,c(3,4,5,8,12,13,14)]//taking numeric values

g2<-kmeans(g1,4)//apply kmeans again

g2//check the cluster into which the new tuple has fallen into

plot(g1[c("area","production")],col=g2$cluster)//plot the cluster

g2$cluster

legend("topleft",inset=0.01,legend=c("cluster1","cluster2","cluster3","cluster4"),col=(1:4),lty=1,horiz=F)

//use legend the for the clusters again

---------END OF CLUSTER CODE--------

---------START OF NOVELITY1---------

d<-read.csv(file.choose())//Choosing dataset

data1<-d//storing it in data1

str(data1)

m<-lm(prediction.of.profit.loss~crop+area+N+P+K+rainfall+temperature,data=data1)//create a learned model

summary(m)//obtain which attribute mostly affects the attribute taken on LHS

s<-readline(prompt="enter season:")

c<-readline(prompt="enter crop:")

n<-readline(prompt="enter N:")

n<-as.integer(n)

p<-readline(prompt="enter P:")

p<-as.integer(p)

k<-readline(prompt="enter K:")

k<-as.integer(k)

f<-readline(prompt="enter other fertilizers:")

b<-readline(prompt="enter brand:")

a<-readline(prompt="enter area:")

a<-as.integer(a)

ma<-readline(prompt="enter automation:")

i<-readline(prompt="enter irri\_type:")

so<-readline(prompt="enter soiltype:")

r<-readline(prompt="enter rainfall:")

r<-as.integer(r)

t<-readline(prompt="enter temperature:")

t<-as.integer(t)

pr<-readline(prompt="enter production:")

pr<-as.integer(pr)

d1<-readline(prompt="enter distance:")//ask input from user

d1<-as.integer(d1)//input stored into d1

typeof(s)

x<-predict(m,data.frame("crop"=c,"N"=n,"P"=p,"K"=k,"rainfall"=r,"area"=a,"temperature"=t))//by taking inputs from user we will predict the attribute in lhs in learned model

x//class label predicted

if((0<x)&&(x<=1.5)){ //dicretising the class label means the class label must have only four values either 1 or 2 or 3 or 4

x<-1

}else if((1.5<x)&&(x<=2.5)){

x<-2

}else if((2.5<x)&&(x<=3.5)){

x<-3

}else

x<-4

x

tail(data1)

library(pracma)//to include strcmpi function

thres<-20//taking some threshold distance

count<-0//setting count to zero

near<-3//threshold count

n1<-nrow(data1)//taking number of rows of data1 to n1

for(i in 1:n1)//comparing the input crop with other crops in the dataset

{

result<-strcmpi(as.character(data1[i,2]),c)

if(result)

{

d2<-abs(data1[i,16]-d1)

if(d2<=thres)

count<-count+1

}

}

if(count>=near)//if more than threshold count

{

if(x==4)

x<-3 //decreasing the classlabel

else if(x==3)

x<-2

else if(x==2)

x<-1

else

x<-1

}

x//print the classlabel

**---------END OF NOVELITY1------------**

--------**START OF NOVELITY2**------------

t1<-read.csv(file.choose())//choosing dataset

t2<-read.csv(file.choose())//choosing dataset

install.packages("sqldf")

library(sqldf)

data<-sqldf("SELECT crop,t1.rawmaterial,AVG(punjab) as p,AVG(madhyapradesh) as m,AVG(himachalpradesh) as h,AVG(mizoram) as z,AVG(ABS(d-distance1)) as pd,AVG(ABS(d-distance2)) as md,AVG(ABS(d-distance3)) as hd,AVG(ABS(d-distance4)) as zd FROM t1,t2 WHERE t1.rawmaterial=t2.rawmaterial GROUP BY crop,t1.rawmaterial

")//storing crop with its rawmaterial,average production in different states,average distance from the state where the crop is grown to different states into data

Data//print data

d<-sqldf("SELECT crop,t1.rawmaterial,MAX(AVG(punjab),AVG(madhyapradesh),AVG(mizoram),AVG(himachalpradesh)) AS MAXPRO,MIN(AVG(ABS(d-distance1)),AVG(ABS(d-distance2)),AVG(ABS(d-distance3)),AVG(ABS(d-distance4))) AS MINDIS FROM t1,t2 WHERE t1.rawmaterial=t2.rawmaterial GROUP BY crop,t1.rawmaterial

")//storing crop with its rawmaterial,max production of rawmaterial in particular state,min distance from state where crop is grown to different state into d

d//printing d

c<-readline(prompt="enter crop")

Banana

x<-readline(prompt="enter rawmaterial1 for crop")

zinc

y<-readline(prompt="enter rawmaterial2 for crop")

sulphur

if(sqldf("SELECT MAXPRO from d where crop='Banana' and rawmaterial='zinc'")==sqldf("SELECT AVG(punjab) FROM t1,t2 where t1.rawmaterial=t2.rawmaterial and crop='Banana' and t1.rawmaterial='zinc'"))

{

print("for growing banana crop recommendable state for importing zinc rawmaterial is punjab w.r.t production")

}else if(sqldf("SELECT MAXPRO from d where crop='Banana' and rawmaterial='zinc'")==sqldf("SELECT AVG(madhyapradesh) FROM t1,t2 where t1.rawmaterial=t2.rawmaterial and crop='Banana' and t1.rawmaterial='zinc'"))

{

print("for growing banana crop recommendable state for importing zinc rawmaterial is madhyapradesh w.r.t production")

}else if(sqldf("SELECT MAXPRO from d where crop='Banana' and rawmaterial='zinc'")==sqldf("SELECT AVG(himachalpradesh) FROM t1,t2 where t1.rawmaterial=t2.rawmaterial and crop='Banana' and t1.rawmaterial='zinc'"))

{

print("for growing banana crop recommendable state for importing zinc rawmaterial is himachalpradesh w.r.t production")

}else{

print("for growing banana crop recommendable state for importing zinc rawmaterial is mizoram w.r.t production")

}//printing recommendable state for particular crop for importing rawmaterial w.r.t max production

if(sqldf("SELECT MINDIS from d where crop='Banana' and rawmaterial='zinc'")==sqldf("SELECT AVG(ABS(d-distance1)) FROM t1,t2 where t1.rawmaterial=t2.rawmaterial and crop='Banana' and t1.rawmaterial='zinc'"))

{

print("for growing banana crop recommendable state for importing zinc rawmaterial is punjab w.r.t distance")

}else if(sqldf("SELECT MINDIS from d where crop='Banana' and rawmaterial='zinc'")==sqldf("SELECT AVG(ABS(d-distance2)) FROM t1,t2 where t1.rawmaterial=t2.rawmaterial and crop='Banana' and t1.rawmaterial='zinc'"))

{

print("for growing banana crop recommendable state for importing zinc rawmaterial is madhyapradesh w.r.t production")

}else if(sqldf("SELECT MINDIS from d where crop='Banana' and rawmaterial='zinc'")==sqldf("SELECT AVG(ABS(d-distance3)) FROM t1,t2 where t1.rawmaterial=t2.rawmaterial and crop='Banana' and t1.rawmaterial='zinc'"))

{

print("for growing banana crop recommendable state for importing zinc rawmaterial is himachalpradesh w.r.t distance")

}else{

print("for growing banana crop recommendable state for importing zinc rawmaterial is mizoram w.r.t distance")

}//printing recommendable state for particular crop for importing rawmaterial w.r.t min distance

if(sqldf("SELECT MAXPRO from d where crop='Banana' and rawmaterial='sulphur'")==sqldf("SELECT AVG(punjab) FROM t1,t2 where t1.rawmaterial=t2.rawmaterial and crop='Banana' and t1.rawmaterial='sulphur'"))

{

print("for growing banana crop recommendable state for importing sulphur rawmaterial is punjab w.r.t production")

}else if(sqldf("SELECT MAXPRO from d where crop='Banana' and rawmaterial='sulphur'")==sqldf("SELECT AVG(madhyapradesh) FROM t1,t2 where t1.rawmaterial=t2.rawmaterial and crop='Banana' and t1.rawmaterial='sulphur'"))

{

print("for growing banana crop recommendable state for importing sulphur rawmaterial is madhyapradesh w.r.t production")

}else if(sqldf("SELECT MAXPRO from d where crop='Banana' and rawmaterial='sulphur'")==sqldf("SELECT AVG(himachalpradesh) FROM t1,t2 where t1.rawmaterial=t2.rawmaterial and crop='Banana' and t1.rawmaterial='sulphur'")){

print("for growing banana crop recommendable state for importing sulphur rawmaterial is himachalpradesh w.r.t production")

}else{

print("for growing banana crop recommendable state for importing sulphur rawmaterial is mizoram w.r.t production")

}

if(sqldf("SELECT MINDIS from d where crop='Banana' and rawmaterial='sulphur'")==sqldf("SELECT AVG(ABS(d-distance1)) FROM t1,t2 where t1.rawmaterial=t2.rawmaterial and crop='Banana' and t1.rawmaterial='sulphur'"))

{

print("for growing banana crop recommendable state for importing sulphur rawmaterial is punjab w.r.t distance")

}else if(sqldf("SELECT MINDIS from d where crop='Banana' and rawmaterial='sulphur'")==sqldf("SELECT AVG(ABS(d-distance2)) FROM t1,t2 where t1.rawmaterial=t2.rawmaterial and crop='Banana' and t1.rawmaterial='sulphur'"))

{

print("for growing banana crop recommendable state for importing sulphur rawmaterial is madhyapradesh w.r.t production")

}else if(sqldf("SELECT MINDIS from d where crop='Banana' and rawmaterial='sulphur'")==sqldf("SELECT AVG(ABS(d-distance3)) FROM t1,t2 where t1.rawmaterial=t2.rawmaterial and crop='Banana' and t1.rawmaterial='sulphur'"))

{

print("for growing banana crop recommendable state for importing sulphur rawmaterial is himachalpradesh w.r.t distance")

}else{

print("for growing banana crop recommendable state for importing sulphur rawmaterial is mizoram w.r.t distance")

}

------------END OF NOVELITY2-----------