#Error de Prediccion Aparente

APE=function(datos,y)

{

datos=as.matrix(datos)

n=dim(datos)[1]

resi=lm(datos[,y]~datos[,-y])$res

APE=sum(resi^2)/n

return(APE)

}

APE(rock,1)

APE(women,2)

#PRESS

PRESS=function(datos,y)

{

datos=as.matrix(datos)

n=dim(datos)[1]

resi=rep(0,n)

for (i in 1:n)

{

estim=sum(lm(datos[-i,y]~datos[-i,-y])$coe\*c(1,datos[i,-y]))

resi[i]=datos[i,y]-estim

}

PRESS=sum(resi^2)/n

return(PRESS)

}

PRESS(rock,1)

PRESS(women,2)

#Validacion Cruzada

crossval=function(data,repet,K,y)

{

data=as.matrix(data)

n=dim(data)[1]

p=dim(data)[2]

EVC=rep(0, repet)

for (i in 1:repet)

{

resid=matrix(0,1,K)

indices=sample(1:n,n,replace=F)

azar=data[indices,]

subm=floor(n/K)

for (j in 1:K)

{

unid=((j-1)\*subm+1):(j\*subm)

if (j == K)

{

unid=((j-1)\*subm+1):n

}

datap=azar[unid,]

datae=azar[-unid,]

ye=datae[,y]

xe=datae[,-y]

betas=lm(ye~xe)$coef

unos=rep(1,dim(datap)[1])

data1=cbind(unos,datap[,-y])

predict=data1%\*%as.matrix(betas)

resid[j]=sum((predict-datap[,y])^2)

}

EVC[i]=sum(resid)/n

}

EVCP=mean(EVC)

return (list(EVC=EVC, EVCP=EVCP))

}

datos = read.table(file.choose(),header=T)

APE(datos,1)

PRESS(datos,1)

crossval(datos,20,4,1)

crossval(datos,20,30,1)

#Analizar el sesgo

P=APE(datos,1)

#Si K=4

crossval(datos,50,4,1)$EVCP-P

#Si K=30

crossval(datos,50,30,1)$EVCP-P

#Analizar la varianza de los errores de prediccion

#Si K=4

var(crossval(datos,50,4,1)$EVC)

#Si K=30

var(crossval(datos,50,30,1)$EVC)

#Analizar el tiempo de procesamiento

#Si K=4

system.time(crossval(datos,50,4,1))[3]

#Si K=30

system.time(crossval(datos,50,30,1))[3]

vcg=function(datos,y)

{

datos=as.matrix(datos)

n=dim(datos)[1]

p=dim(datos)[2]

resi=lm(datos[,y]~datos[,-y])$res

num=sum(resi^2)/n

den=(1-(p/n))^2

VCG=num/den

return(VCG)

}

APE(datos,1)

PRESS(datos,1)

crossval(datos,20,4,1)$EVCP

crossval(datos,20,30,1)$EVCP

vcg(datos,1)