CODE FOR FEATURE SELECTION

```
function AAA = clas_PCA_KNN(yyy,A,prop,Knn,sigma,NOI,claf,VII)
% A=data (matrix consisting of 9 variables and the class in the 10th
column)
% prop=proportion
% Knn: number of nearest neighbors

tic
[na,ma]=size(A);


tt=0;
while tt<NOI

tt=tt+1;

[train,test] = crossvalind('HoldOut',A(:,ma),(1-prop));
tr = A(train,1:(ma-1));
ta = A(train,ma);
ts = A(test,1:(ma-1));
correct = A(test,ma);



[mtr,ntr]=size(tr);
[mts,nts]=size(ts);
tr1=zscore(tr);
ts1=zscore(ts);

if VII==1
[coefs,scores,variances,t2] = princomp(tr1);
coefs;
P=[variances(1)*ones(ma-1,1),variances(2)*ones(ma-1,1)];
[PP]=abs(coefs(:,1:2)).*P;
WWW1=[sum(PP')',(1:ntr)'];
%WWW1=[sum(abs(coefs(:,1:3))')',(1:ntr)'];
WWW=flipud(sortrows(WWW1,1));
elseif VII==2
% Batacharryya - inicio

[observacoes,variaveis]=size(tr1);
x=1;
y=1;
d=zeros(variaveis,1);
for n=1:observacoes
    if ta(n)~=0;
        C1(x,:)=tr1(n,:);
        x=x+1;
    else
        C0(y,:)=tr1(n,:);
        y=y+1;
    end
end
for nn=1:variaveis
```

```
    vA=var(C1(:,nn));
    vB=var(C0(:,nn));
    mA=mean(C1(:,nn));
    mB=mean(C0(:,nn));
    d(nn,1)=0.25*(log(0.25*((vA/vB)+(vB/vA)+2)))+0.25*(((mA-
mB)^2)/(vA+vB));
end
a=1:variaveis;
S=[d,a'];
WWW=sortrows(S,-1);

% Batacharryya - fim

else VII==3;


[coefs,scores,variances,t2] = princomp(tr1,'econ');
coefs;
P=[variances(1)*ones(ma-1,1),variances(2)*ones(ma-1,1)];
[PP]=abs(coefs(:,1:2)).*P;
WWW1=[sum(PP')',(1:ntr)'];
%WWW1=[sum(abs(coefs(:,1:3))')',(1:ntr)'];
WWW=flipud(sortrows(WWW1,1));



[observacoes,variaveis]=size(tr1);
x=1;
y=1;
d=zeros(variaveis,1);
for n=1:observacoes  ;
    if ta(n)~=0;
        C1(x,:)=tr1(n,:);
        x=x+1;
    else
        C0(y,:)=tr1(n,:);
        y=y+1;
    end
end
for nn=1:variaveis;
    vA=var(C1(:,nn));
    vB=var(C0(:,nn));
    mA=mean(C1(:,nn));
    mB=mean(C0(:,nn));
    d(nn,1)=0.25*(log(0.25*((vA/vB)+(vB/vA)+2)))+0.25*(((mA-
mB)^2)/(vA+vB));
end
a=1:variaveis;
S=[d,a'];
WWW=sortrows(S,-1);
WWWcomb=[WWW(:,1).*S(:,1),a'];
WWW=sortrows(WWWcomb,-1);

end
```

```
tr1=tr1(:,WWW(:,2));
ts1=ts1(:,WWW(:,2));
WWW2=[WWW(:,2)];
for mm=0:ntr-2
    tr2=tr1(:,1:ntr-mm);
    ts2=ts1(:,1:ntr-mm);
    WWW22=WWW2(1:ntr-mm);

train_patterns1=tr2';
x1x=size (train_patterns1);
test_patterns1=ts2';
train_targets=ta;

    if yyy==5
        limit=2;
        [COEFF,SCOREtr,latent] = princomp(tr2);
        [COEFF,SCOREts,latent] = princomp(ts2);
        train_patterns1=SCOREtr(:,1:limit-1)';
        test_patterns1=SCOREts(:,1:limit-1)';
    end

if claf==1;
test_targets=knnclassify(ts2,tr2,ta)';
% L         = length(train_targets);
% Uc          = unique(train_targets);
%
% if (L < Knn),
%    error('You specified more neighbors than there are points.')
% end
% N               = size(test_patterns1, 2);
% test_targets    = zeros(1,N);
% for i = 1:N,
%     dist            = sum(((train_patterns1 -
test_patterns1(:,i)*ones(1,L))).^2);
%
%     [m, indices]    = sort(dist);
%
%     n               = hist(train_targets(indices(1:Knn)), Uc);
%
%     [m, best]       = max(n);
%
%     test_targets(i) = Uc(best);

%end

elseif claf==2;
       test_targets=classify(ts2,tr2,ta,'diaglinear')';


else claf==3;


    [Dim, Nf]       = size(train_patterns1);
Dim             = Dim + 1;
train_patterns1(Dim,:) = ones(1,Nf);
```

```
u_targets       = unique(train_targets);

%Build the classifier
x               = train_patterns1;
W               = x ./ (ones(Dim,1)*sqrt(sum(x.^2)));  %x_jk <- x_jk /
sqrt(sum(x_ji^2)), w_jk <- x_jk

%if x in w_i then a_ji <- 1
a = zeros(Nf, length(u_targets));
for i = 1:length(u_targets),
    a(find(train_targets == u_targets(i)),i) = 1;
end

%Test it and classify the test patterns
test_patterns1 = [test_patterns1; ones(1, size(test_patterns1,2))];
test_patterns1 = test_patterns1 ./
(ones(Dim,1)*sqrt(sum(test_patterns1.^2)));

%net_k <- W'_t*x
net         = W' * test_patterns1;

%if a_ki=1 then g_i <- g_i + exp((net-1)/sigma^2)
arguments   = zeros(length(u_targets),size(test_patterns1,2));
for i = 1:length(u_targets),
    mask            = a(:,i) * ones(1,size(test_patterns1,2));
    arguments(i,:)  = sum(exp((net-1)/sigma^2) .* mask);
end

%class <- argmax g(x)
[m, indices] = max(arguments);
test_targets = zeros(1,size(test_patterns1,2));
for i = 1:length(u_targets),
    test_targets(find(indices == i)) = u_targets(i);
end
end

G=test_targets;
size(G);
H=G'==correct;
ACC_mm=sum(H)/mts;
ACC(mm+1,1)=ACC_mm;
TP_mm=sum(G'==1 & correct==1);
SENS_mm=sum(G'==1 & correct==1)/sum(correct');
SENS(mm+1,1)=SENS_mm;
TN_mm=sum(G'==0 & correct==0);
SPEC_mm=sum(G'==0 & correct==0)/(mts-sum(correct'));
SPEC(mm+1,1)=SPEC_mm;
FP_mm=sum(G'==1 & correct==0);
FN_mm=sum(G'==0 & correct==1);
PosPredValue_mm=TP_mm/(TP_mm+FP_mm);
PosPredValue(mm+1,1)=PosPredValue_mm;
NegPredValue_mm=TN_mm/(TN_mm+FN_mm);
NegPredValue(mm+1,1)=NegPredValue_mm;
end
```

```
mts;
uns_tt=sum(correct');
UNS(tt,1)=uns_tt;
MeanUNS=mean(UNS);
zeros_tt=mts-sum(correct');
ZEROOS(tt,1)=zeros_tt;
MeanZEROS=mean(ZEROOS);
ACC;
ty=flipud([2:ntr]');
%ACC2=flipud([ACC,ty]);

%maxACC(tt,1)=maxACC_tt;
%dd(tt,1)=dd_tt;
%MACC=mean(maxACC);
%MVR=mean(dd);


%sinal=ACC>=0.9951
Acc_sens_espec_PosPredValue_NegPredValue=flipud([ACC,SENS,SPEC,PosPredVal
ue,NegPredValue,ty/ntr]);

[maxACC_tt,dd_tt]=max(Acc_sens_espec_PosPredValue_NegPredValue(:,1));
AAA_tt=Acc_sens_espec_PosPredValue_NegPredValue(dd_tt,:);
AAA(tt,1:6)=AAA_tt;
MeanAAA=mean(AAA);
DesvAAA=std(AAA);
VARS=WWW(1:dd_tt+1,2);
[nvar,mvar]=size(VARS);
VARS2_tt=[VARS;zeros(ma-1-nvar,1)]';
VARS2(tt,1:ma-1)=VARS2_tt;


end

MeanAAA
DesvAAA

for jj=1:ma-1
[P1]=VARS2==jj;
P2_jj=sum(sum(P1));
P2(jj,1)=P2_jj/NOI;
end
uu=[1:ma-1]';
ID_incidences=[uu,P2];
bar(ID_incidences(:,2))%),ID_incidences(:,1))
xlabel('Feature ID')
ylabel('Retention frequency')
ID_incidences=sortrows(ID_incidences,-2);
%plot(ACC2(:,2),ACC2(:,1))
%xlabel('Number of retained attributes')
%ylabel('Classification accuracy')
toc
```