Assignment: multivariate analysis of a dataset in R

I0P16a: Applied Multivariate Statistical Analysis

Prof. E. Schrevens 2021-2022

Mianyong Ding

R0823572

Introduction

This report contains five parts, introduction, methodology, results, conclusion and annex. The all detailed R-code is included in the annex part.

1.1 Description of problem

As the final-year students of Bioinformatics, half of us will go to job markets after the graduation according to the past years' situations. However, the jobs related bioinformatics are very limited in the job search websites such as Indeed, LinkedIn. Only a few positions are offered and some of them ask for a PhD degree. If we want to find the job of other fields, many choices provided may waste our time to read job description one by one. And the users have to provide different keywords for searching, sometimes the users may miss some potential unrecognized fields with suitable jobs(and sometimes even have no idea what to enter). For the jobs published on the website, each of them has several paraphs with the contents of company description, job description, job requirements. The occurrence and frequency of some keywords in these contents may highly represent the types of jobs. Several questions are explored in this assignment. Can we extract some most frequent words from the these contents which can highly represent the job types ("bioinformatics","marketing"…)? Can we find some distinct clusters of different job types based on these highly frequent words? Can we build the discriminant model with high accuracy to find several jobs of other fields which are highly related to the bioinformatics?

1.2 Description of the data

The data was scraped from Indeed(www.indeed.com) (last accessed 29th Jan 2022). For each type of jobs, keywords were entered in the search panel and the jobs were searched globally. The data was extracted with the filters of "master's degree ", "entry-level" and "full-time". The data was firstly tried to be scraped by "rvest" package in R. After several attempts, the python was used to scrape the data. The data and python on my personal GitHub website (https://github.com/Raminmian/scraping-AMSA-assign). For the datasets, there were four columns ("number1", "number2", "job title", "job description", "category") and 480 rows(each row stands for a job posted on Indeed). The data contains four category (bioinformatics, teacher, marketing, IT) and each category has 120 observations.

Methodology

The text in the job description was process with "tm" package, "qdap" package. The whitespace, stopword, punctuation, lower case, stemmer was processed.

To analysis the observation related to the bioinformatics, the top 30 most frequent words were extracted as new variables and the value of each variable was calculated as count of worlds divided by the number of words of the processed of job description and multiplied by 100. The scales of each variable are close, no scaling was required, and the data was centred. The PCA was used to count the percentage of variances caught by first and second principal components.

The world clouds were extracted using the function from “worldcloud” package. The biplot function was used to see the correlation of variables.

In most analysis, the variables are the frequency of the common words. To select the good number of variables of observations, there is a trade-off between the variable and observation. With the increase of number of observations, the relatedness will decrease, as the jobs posted on the first page is highly related. If the number of observations is too low, the common variables without specific meaning will dominate the variables. After several attempts, for each category, 120 is chosen. The number of variables was chosen differently for each analysis described as below. When we choose the number of variables, too many variables will cause a lot of noise and too less variables will lose the specific features of each observation. The reason why the most common words were chosen separately according to different categories is to preserve the specific features of each category. If the variables were extracted together, there will be a lot of manfulness common words.

To firstly check if two categories of job can be separated, the category “marketing” and “bioinformatics” were used. The top 30 most common words of each category were chosen, merged and duplicated elements were removed. The processed words were extracted as new variables and the value of all observations were calculated as the same way as above. The four clustering methods were used and compared. The methods are hierarchical clustering methods with complete linkage distance, average linkage distance and ward methods and non-hierarchical methods k-means clustering. The accuracy of classification was calculated.

To cluster all four categories of jobs, the top 30 most common words of each category were chosen, processed as new variables. The observation with too many zeros and the variables with too many zeros were removed. Finally, the 240 observations with 86 variables were used for further analysis. The PCA was used after centring the data matrix. The non-metric analysis from vegan package were used, the MDA biplot was generated to see the association between clusters and variables. The ward method and k-means clustering method were used, and the accuracy was calculated.

To build the discriminant analysis model, the ida and qda function from vegan package are used as the covariance of the groups are not known. The biplot based on ida model was generated.

Results and Discussion

Information about category of bioinformatics

From the word cloud of bioinformatics (figure 1), we can see some top common worlds can represent the features of bioinformatics very well such as “data”, “bioinformat”, “biology”, “develop” and “compute”. For PCA analysis, 21% variance is explained by PC 1 and 15% is explained by PC 2, the first two PCs cannot catch the majority of variance. The variables are not highly correlated. The two PCs cannot catch most of the variance, so there is a lot of missing information of variables in the pc biplot, however we can still find some interested information. Some common meaningfulness words are highly correlated, such as “expri”, “team”, “support”, “develop”. And “bioinforma” is highly correlated with some specific features such as “data”,

“program”, “comput” and “genom”. And these two clusters are close to be orthogonal, showing some independence. These common meaningfulness words can be removed to improve the performance.

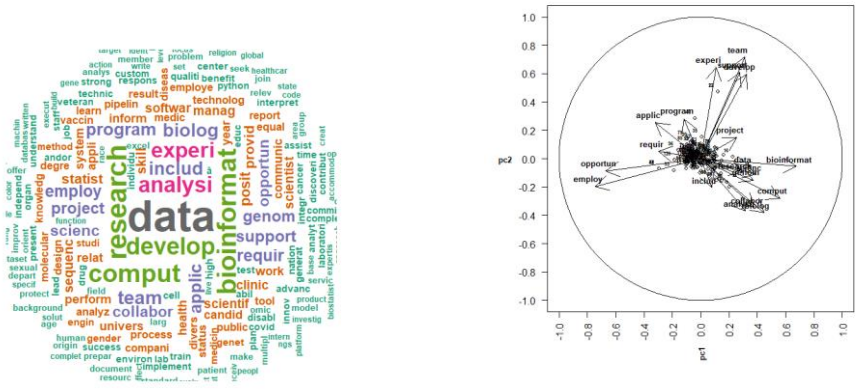

Figure 1. The world cloud(left) and PCA biplot(right) of bioinformatics jobs

Clustering between marketing and bioinformatics

From the cluster Dendrogram (figure 2), we can see the ward D methods are the best, two clear clusters with close size are displayed as left and right branches. For the complete linkage and average linkage methods, they are easily influenced by some outliers, no distinct two clusters are formed. And the accuracy of clustering of ward D methods and k- means clustering are 95.8% and 96.7%.

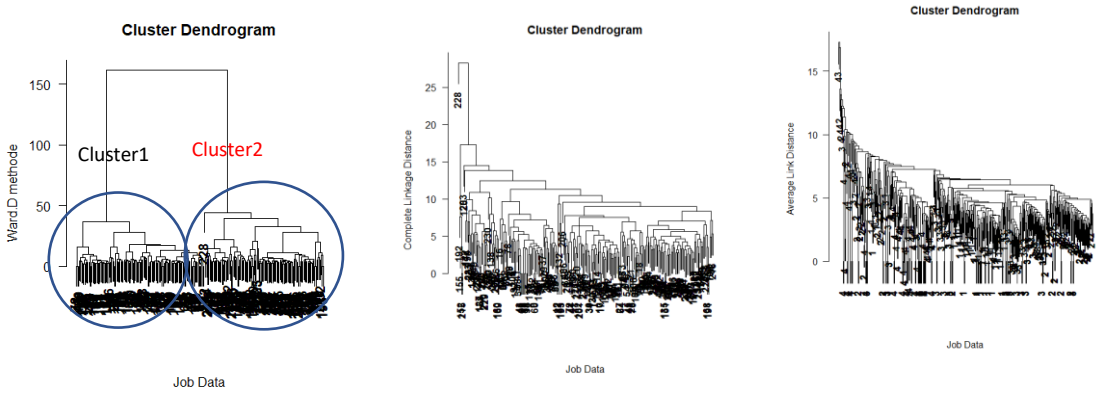

Figure 2. The cluster dendrogram of ward D(left), complete linkage distance(middle), average linkage distance methods (right).

The explained variance of first two PCs are 21% and 15%. The clustering plots projected on first two PCs are not informative enough, but still showing two close clusters. The two clusters are not very distinct.

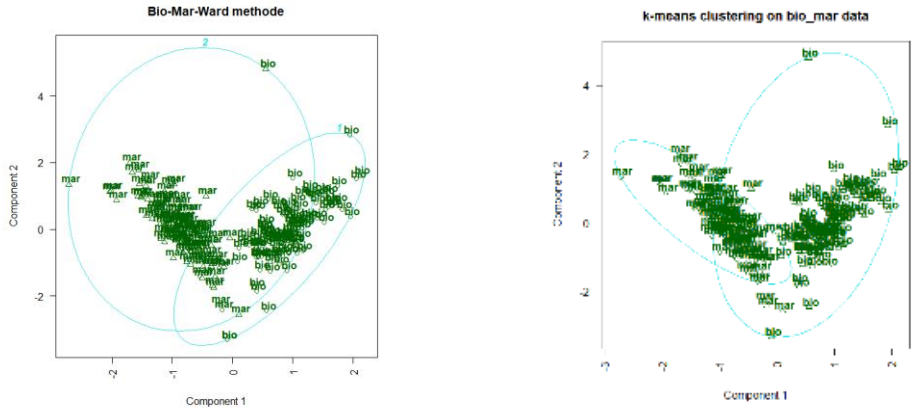

Figure 3. The cluster plots of ward D method (left) and k-means(right). The bio stands for bioinformatics and the mar stands for the marketing.

Clustering for four categories

For k-mean(left) and ward D(right), the tables to represent the classification is shown as table 1

From	Classified into 1	2	3	4
1	109	8	0	0
2	3	113	0	1
3	0	117	0	0
4	0	4	94	2

From	Classified into 1	2	3	4
1	105	11	1	0
2	0	112	4	1
3	3	8	106	0
4	0	9	0	91

Table 1. Contingency Tables show the classification of k-mean(left) and ward methods(right).

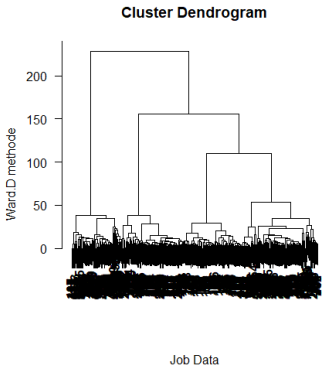

Figure 4. The cluster dendrogram of four clusters

The fourth cluster of k-mean is not good, there are two sub-cluster in cluster 2 which are not classified by k-mean. The ward method is much better, four clusters are formed with small misclassification error (with accuracy of 91.7%). The dendrogram is shown as figure 4.

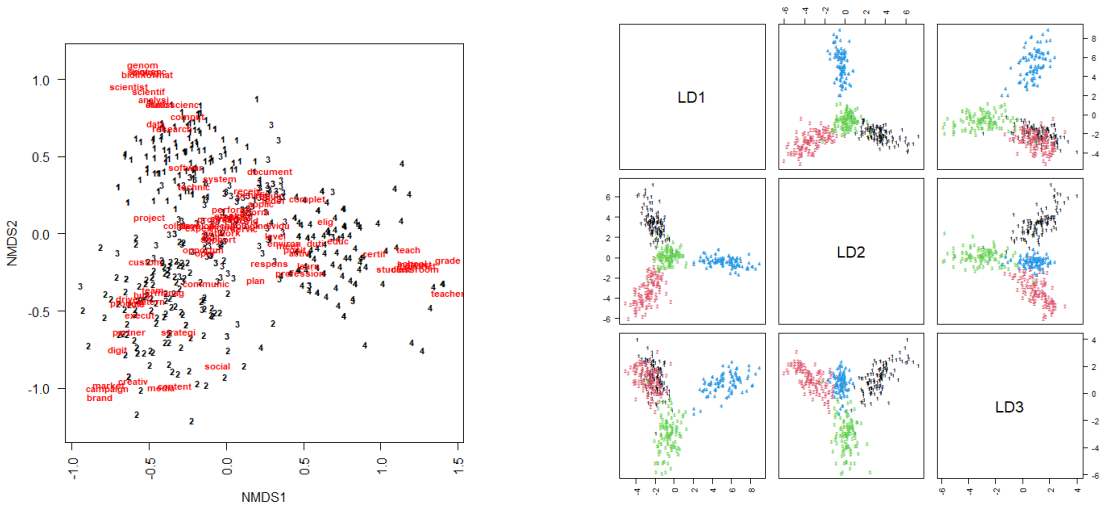

Figure 5 The MDS biplot(left) and biplot on LDA biplot (right). 1 is “bioinformatics”, 2 is “marketing”, 3 is "teacher” and 4 is “IT”

With the MDS biplot, we can see some clusters with associated variables. For example, the bioinformatic is highly associated with “data”, “science”, “bioinfor” and the teacher is associated the keyword “student”, “grade”, “teach” and marketing is associated with “digital”. The explained variance of first two PCs are 17.61% which is quite low.

The linear discriminant analysis model for the prediction is built with low classification of error with or without 10-fold cross validation (table 2). The print of trace of each LD represents the percentage separation achieved by each discriminant function. The LD1 is 50.8% and the LD2 is 31.87%. The projection of data on LD1 and LD2 produce good clusters (figure 5).

	Classified into			
From	1	2	3	4
1	115	0	2	0
2	0	116	1	0
3	0	1	116	0
4	0	0	2	98

	Classified into			
From	1	2	3	4
1	110	0	7	0
2	1	112	4	0
3	1	3	112	1
4	0	0	3	97

Table 2. Contingency tables of misclassification of lda model with(right) and without (left) 10-fold cv.

Conclusion

From the analysis, the hierarchical methods with ward. D means can make good classification with this type of dataset. The most common words can represent the type of jobs very well. The most common words can be used to make the clusters between different categories. The model of prediction with linear discriminant analysis with low error can be built. If we extract scape more information about more categories of jobs, we can build the model which can make more classification. For the user who have no idea what to enter in the job search panel, we can just scrape a large number of jobs of random fields and use the model to make the prediction. The jobs have the same cluster as bioinformatics jobs or very close to the cluster of bioinformatics can be extracted for further check. This procedure can narrow the selection range and save time.

Annex

R code

```
library(qdap)
library(tm)
library("stats")
library("wordcloud")
library(cluster)
library(MASS)
library(plotrix)
library(vegan)
extractword<-function(data){ #function to extract the common words
  text=data
  text=tolower(text)  #to lower case
  text=removePunctuation(text)   # Remove punctuation
  text=removeNumbers(text)   # Remove numbers
  text=stripWhitespace(text)  #Remove whitespace
  text=bracketX(text)  # Remove text within brackets
  text=removeWords(text, stopwords("en"))   # remove  text with standard stop words
  text=removeWords(text, stopwords("SMART"))
  all_stops <- c("work", "job", stopwords("en"))
  text=removeWords(text, "work")   # stemming the words
  text=text_tokens(text, stemmer = "en")
  return(text)
}
# function to return the top frequent words with provided num
vari<-function(text,num){
  fre <- freq_terms(text, num)
}
#remove the variables with two many zero , fdd is the dataframe, thres is the percentage of not zero
remove1<-function(fdd,thres){
  #process the data
  rl=c()
  for (i in 1:ncol(fdd)){if(sum(fdd[,i]!=0)/nrow(fdd)>=thres){ rl=c(rl,i) } }
```

```
  dan<-fdd[,c(rl)]
  return(dan)
}
#remove the observation with two many zero of variables, fdd is the dataframe, thres is the percentage of not zero
remover<-function(fdd,thres){
 rl=c()
  for (i in 1:nrow(fdd)){
    if(sum(fdd[i,]!=0)/ncol(fdd)>=thres){ rl=c(rl,i) } }
  dan<-fdd[c(rl),]
  return(dan)
}
PCA.biplot<- function(x) {
  xm<-apply(x,2,mean)
  y<-sweep(x,2,xm)
  ss<-(t(y)%*%y)
  s<-ss/(nrow(x)-1)
  d<-(diag(ss))^(-1/2)
  e<-diag(d,nrow=ncol(x),ncol=ncol(x))
  z<-y%*%e
  r<-t(z)%*%z
  q<-svd(z)
  gfd<-((q$d[1])+(q$d[2]))/sum(q$d)
  gfz<-(((q$d[1])^2)+((q$d[2])^2))/sum((q$d)^2)
  gfr<-(((q$d[1])^4)+((q$d[2])^4))/sum((q$d)^4)
  l<-diag(q$d,nrow=ncol(x),ncol=ncol(x))
  R.B<-q$u
  C.B<-q$v%*%l
  par(mar=c(4,4,4,4),pty='s',oma=c(5,0,0,0),font=2)
  plot(R.B[ ,1],R.B[ ,2],axes=F,xlim=c(-1,1),ylim=c(-1,1),xlab=' ',ylab=' ',cex=.8)   #Define axes systems between -1
and +1
  mtext('pc1',side=1,line=3,cex=.8)
  mtext('pc2',side=2,line=3,cex=.8)
  axis(1,at=c(-1,-.8,-.6,-.4,-.2,0,.2,.4,.6,.8,1),cex=.8)
  axis(2,at=c(-1,-.8,-.6,-.4,-.2,0,.2,.4,.6,.8,1),cex=.8)
```

```
  box( )
  text(R.B[,1]-.05,R.B[,2]+.05,as.character(dimnames(x)[[1]]),cex=0.5)
  points(R.B[,1],R.B[,2],pch=".")
  points(C.B[,1],C.B[,2],pch=".")
  text(C.B[,1]-.05,C.B[,2]+.05,as.character(dimnames(x)[[2]]),cex=0.8)          #Plot the variable names, dimnames(x)[[2]] necessary to label the variables
  for (i in seq(1,nrow(C.B),by=1))
    arrows(0,0,C.B[i,1],C.B[i,2])
  draw.circle(0,0,1,border='black')
  results<-list('correlation matrix'=r,'column effects'=C.B,'row effects'=R.B)
  cat('The goodness of fit for the correlation matrix is',gfr,'for the centered, standardized design matrix',gfz,'and for the Mahalanobis distances is',gfd,' '
  results
}
#count the frequency of each words in one observation,data1 is orgnial data, var1 is the name of variable and cat is catorgry
#allcha is the process description
countword<-function(data1,var1,cat,allcha){
  ww=t(as.data.frame(rep(NA,length(var1))))
  for (j in 1:(nrow(data1))){
    occ=rep(NA,length(var1))
    for (i in 1:length(var1)){
      occ[i]=(sum(allcha[[j]]==var1[i])/length(allcha[[j]]))*100  }
    ww=rbind(ww,occ)}
  ww=ww[-1,]
  #combine the category
  ww=cbind(cat,ww)
  #change the rowname and column names
  colnames(ww)[-1]<-var1
  rownames(ww)<-1:nrow(ww)
  return(ww)
}
#read the data
data1 <- read.csv(file = 'newdata.csv', header = TRUE)
```

```
#############################################################
#extrac different types of data
nn=120
a1<-data1[1:nn,4]
a2<-data1[121:240,4]
a3<-data1[241:360,4]
a4<-data1[361:480,4]
#extract all
order=30
d1=extractword(a1)
v1=vari(d1,order)
d2=extractword(a2)
v2=vari(d2,order)
d3=extractword(a3)
v3=vari(d3,order)
d4=extractword(a4)
v4=vari(d4,order)

#merge all common variables
daa=c(v1[,1],v2[,1],v3[,1],v4[,1])
#uniqe the variable,the var1 is the new varibale
var1=unique(daa)
length(var1)

#all processed words
allcha<-c(d1,d2,d3,d4)
#change the category to the number
cat=data1[,5]
cat[cat=="bioinformatics"]=1
cat[cat=="marketing"]=2
cat[cat=="it"]=3
cat[cat=="teacher"]=4
cat=as.numeric(cat)
```

```
ww<-countword(data1,var1,cat,allcha)
##the most frequent words in bioinformatics
vbio=vari(d1,200)

#wordcloud
set.seed(111)
wordcloud(words = vbio$WORD, freq = vbio$FREQ, min.freq = 1,
        max.words=200, random.order=FALSE, rot.per=0.35,
        colors=brewer.pal(8, "Dark2"))

#foucus on the bioin
bioin<-ww[,v1[1:20,1]][1:120,]
PCA.biplot(bioin)
x.pca <- princomp(bioin)
y<-(x.pca$sdev)^2
p1<-y[1]/sum(y)
p2<-y[2]/sum(y)
################## cluster between marketing and bioinformatics,comparing different methods
datbio<-data1[1:120,]
datma<-data1[121:240,]
order=30
teb=extractword(te120)
bvt=vari(teb,order)[,1]
bio120<-datbio[2:nrow(datbio),4]
bbio=extractword(bio120)
bv1=vari(bbio,order)[,1]

allchab<-c(bbio,teb)
catb<-rep(1,length(allchab))
bff<-countword(rbind(datbio,datea[-1,]),unique(c(bv1,bvt)),catb,allchab)

bmat<-scale(bff[,2:ncol(bff)],center=T,scale=F)
xx=bmat
```

```
x.pca <- princomp(bioin)
y<-(x.pca$sdev)^2
p1<-y[1]/sum(y)
p2<-y[2]/sum(y)

bmclusA <- hclust(dist(xx), method="average")  # Averge linkage
class(bmclusA)
plot(bmclusA,xlab="Job Data",ylab="Average Link Distance", sub="")
bmgpA <- cutree(bmclusA,k=11)
table(bmgpA)
bmgpA

#ncompelet linkage
bmclusC <- hclust(dist(xx), method="complete")
plot(bmclusC,xlab="Job Data",ylab="Complete Linkage Distance", sub="")
bmgpC <-cutree(bmclusC,k=10)
table(bmgpC)



#ward metthod
bmclusD <- hclust(dist(xx), method="ward.D")
plot(bmclusD,xlab="Job Data",ylab="Ward.D methode",sub="")
bmgpD <- cutree(bmclusD,k=2)
table(bmgpD)

#calculate acucuracy
acc<-(sum(z[1:120]==1)+sum(z[121:240]==2))/240

rownames(xx)<-c(rep("bio",120),rep("mar",120))
clusplot(xx,bmgpD,stand=TRUE,labels=2,main="Bio-Mar-Ward methode".col=1)

#k means cluster
bmkcl <- kmeans(xx, 2, 20)
```

```
table(bmkcl$cluster)

#calculate acucuracy
z=bmkcl$cluster
acc<-(sum(z[1:120]==2)+sum(z[121:240]==1))/240
clusplot(xx,bmkcl$cluster,stand=TRUE,labels=3,main="k-means clustering on bio_mar data")

###################### more jobs categories
fdd<-ww
fid<-remove1(fdd,0.1)
fid<-remover(fid,0.1)
dim(fid)
#############analysis
x.mat=fid[,-1]
x.pca <- princomp(x.mat)
y<-(x.pca$sdev)^2
p1<-y[1]/sum(y)
p2<-y[2]/sum(y)
#do the mds plot
ad<-fid
rownames(ad)<-ad[,1]
allMDS <- metaMDS(ad[,-1],distance="jaccard",k=4)
plot(allMDS,type="t")
#########clutsering
xx=x.mat
#ward metthod
bmclusD <- hclust(dist(xx), method="ward.D")
plot(bmclusD,xlab="Job Data",ylab="Ward.D methode",sub="")
bmgpD <- cutree(bmclusD,k=4)
table(bmgpD)
z=bmgpD
rownames(xx)<-fid[,1]
clusplot(xx,bmgpD,stand=TRUE,labels=2,main="all-Ward methode")
```

```
#to change the labels
ac[ac==2]=5
ac[ac==3]=6
ac[ac==5]=3
ac[ac==6]=2
table(ac,z,dnn=c("From","Classified into"))
#calculate acucuracy
ac<-as.vector((fid[,1]))
acc<-(sum(as.vector((z))==ac))/nrow(xx)
#k means cluster
bmkcl <- kmeans(xx, 4, 20)
table(bmkcl$cluster)
#calculate acucuracy
clusplot(xx,bmkcl$cluster,stand=TRUE,labels=3,main="k-means clustering on all data")
z=bmkcl$cluster
table(ac,z,dnn=c("From","Classified into"))
#move to 5 cluster and remove one cluster
bmkcl <- kmeans(xx, 5, 20)
table(bmkcl$cluster)

acn=ac[bmkcl$cluster!=4]
#remove the cluster 4 and make the cluster to comapre ,no improvement
x2=xx[bmkcl$cluster!=4,]
bmkcl <- kmeans(x2, 4, 20)
table(bmkcl$cluster)
z=bmkcl$cluster
table(acn,z,dnn=c("From","Classified into"))


###finla make the discriminant analysis,build the model to predict the new observation
fid=as.data.frame(fid)
fid$cat<-as.factor(fid$cat)
lda1<-lda(cat~-cat+.,fid)
```

```
# Model overview, descriptive statistics and coefficients of the linear discriminant function
lda1
plot(lda1)
plot(lda1,col=as.numeric(fid$cat))

kernel.pred<-predict(kernel.lda)
kernel.pred
table(fid$cat,kernel.pred$class,dnn=c("From","Classified into"))
cvlda<-lda(cat~-cat+.,fid,CV=TRUE)
table(fid$cat,cvlda$class,dnn=c("From","Classified into"))
qda1<-qda(cat~-cat+.,fid) #too little data, cannt work
```