DATA MINING-Assignment 01

1. **Jagdish Patil(2141)**
2. **Vipul Dhamale(2112)**

QUE.1) Variable Selection

```
Data=read.delim("C:\\Users\\jagdish\\OneDrive\\Documents\\Data mining\\dmassign..txt",header=T)
dim(Data)
```

```
## [1]  200 3001
```

Given data in matrix form

```
D=as.matrix(Data)
n=nrow(D);n# number of rows
```

```
## [1] 200
```

```
p=ncol(D);p#number of columns
```

```
## [1] 3001
```

```
X=D[,-p]#predictor variables
y=D[,p]#response variable
dim(X)
```

```
## [1]  200 3000
```

***we can see that given data is sparse matrix .**

(sparse Matrix:- matrix in which most of element are zero)

***we are going to remove that varibles which have all values zero.**

```r
a=colSums(X)
b=which(a==0)
b
```

```
##   X47  X140  X254  X335  X358  X911  X914  X975 X1475
X1509 X1846 X2290 X2476
##    47   140   254   335   358   911   914   975  1475
1509  1846  2290  2476
## X2591 X2612
##  2591  2612
```

```r
length(b)
```

```
## [1] 15
```

***There are 15 variable which are not significant as all observation in that variables are zero.**

```r
X=X[,-b]
D=D[,-b]
dim(D)
```

```
## [1]  200 2986
```

***we have two responses as healthy patients and cancer patients. Now, in order to separate the observations in both classes.**

```r
HP=which(y==0)#Healthy patients
CP=which(y==1)#cancer patients
length(HP)
```

```
## [1] 112
```

```r
length(CP)
```

```
## [1] 88
```

***There are 112 healthy and 88 cancer patients.**

```
H=X[HP,]
C=X[CP,]
D1=rbind(H,C)
dim(D1)
```

```
## [1]  200 2985
```

As we have given Sparse matrix.

```
f=function(C)
{
  n=length(C)
  P=c()
  P[1]=length(which(C==0))/n
return(P)
}
f1=c()
for(i in1:ncol(D1))
{
  f1[i]=max(f(C[,i]),f(H[,i]))
}
D2=D1[,which(f1<=0.5)]
dim(D2)
```

```
## [1]  200 1496
```

*Now, we are finding which variables are normal and which are non-normal

✠Normality Of Variables using Shapiro-Wilk normality test

✠Healthy Persons

```
alpha=0.01
pval=c()
for(i in1:ncol(D2[1:88,]))
{
  pval[i]=shapiro.test(D2[1:88,i])$p.value
}
length(which(pval>alpha))
```

```
## [1] 77
```

Cancer Person

```
pval1=c()
for(i in1:ncol(D2[89:200,]))
{
  pval1[i]=shapiro.test(D2[89:200,i])$p.value
}
length(which(pval1>alpha))

## [1] 33
```

P-Values

```
p1=matrix(pval,ncol=1)
dim(p1)

## [1] 1496    1

p2=matrix(pval1,ncol=1)
p=cbind(p1,p2)
dim(p)

## [1] 1496    2

normal=c()
for(i in1:nrow(p))
{
  pv=min(p[i,])
  normal[i]=ifelse(pv>=alpha,1,0)
}
table(normal)

## normal
##    0    1
## 1480   16
```

(Null hypothesis : Population is normally distributed)

*The variable for which p-value is greater than alpha(l.o.s) are normal and for which p-value is less

than alpha are non-normal. we get 16 Normal while 1480 Non normal variables.

```
N=D2[,which(normal==1)]  #Normal variable

NonN=D2[,which(normal==0)]  #Non-Normal variable

dim(N)

## [1] 200  16
```

- **T-test:We need to check equality of variances**

```
alpha=0.01
t=c()
for(i in1:16)
{
  t[i]=t.test(N[(1:88),i],N[(89:200),i],var.equal=T)$p.value
}
length(which(t>alpha))

## [1] 9
```

- **Out of 16 normal variable, there are 9 having equal variance from 16 ,there are 7 significant variables**

- **For Non Normal variables we will use Wilcoxon Test**

```
w=c()
for(i in1:1480)
{
  w[i]=wilcox.test(NonN[1:88,i],NonN[89:200,i])$p.value
}
length(which(w<alpha))

## [1] 243
```

- **243 Non-Normal Variables are signigicant.**

```
NEW=cbind(N[,which(t<alpha)],NonN[,which(w<alpha)])
dim(NEW)

## [1] 200 250
```

***Now there are 250 significant variables**

QUE.2) Models

1) KNN k-nearest neighbours

```
library(class)

h=colnames(NEW)
d1=D[,h]#main data
d=cbind(d1,y) #response included
p=ncol(d)
train=d[1:100,]
test=d[101:200,]
p1=knn(train[,-p],test[,-p],train[,p],k=2)
table=table(p1,test[,p])

table
```

P1	0	1
0	46	8
1	10	36

```
accuracy=sum(diag(table))/sum(table)
accuracy

## [1] 0.81
```

***Accuracy is 81 %**

***we can see that 8 healthy patient are misclassified as cancer patient and 10 cancer patient are misclassified as healthy patient**

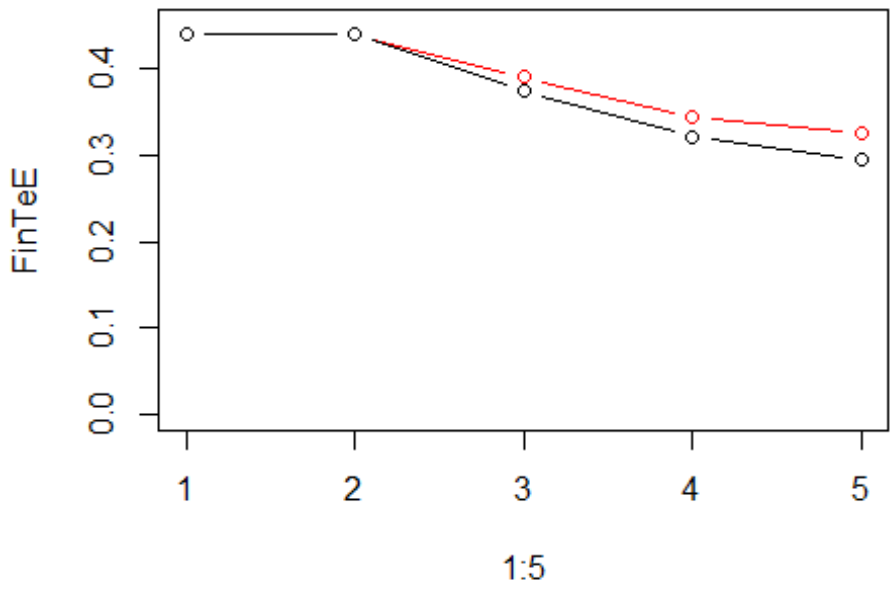
```
h=colnames(NEW)
d1=D[,h]
d=cbind(d1,y)
```

2)k fold cross validation

```
samp=sample(1:n,200,replace = F)
TrainD=d[samp,]
```

```r
newD=d[-samp,]

kf=5
n=nrow(d)
p=ncol(d)
knnclass=function(k,TrainD,newD)
{
  p=ncol(TrainD)
  YTrain=TrainD[,p]
  YnewD=newD[,p]
  n=nrow(newD)
  sc=c()
for (i in1:n) {
    D2=rbind(newD[i,-p],TrainD[,-p])
    DM=as.matrix(dist(D2))
    knndist=sort(DM[1,-1])[k]
    w=which(DM[1,-1]<=knndist)
if (length(w)>k)
      w=w[1:k]
    Ypoll=YTrain[w]
    sc[i]=which.max(table(Ypoll))[1]
  }
return(sc)
}

FinTrE=FinTeE=c()

for(k in1:5)
{
  o=sample(1:n,n,replace = F)

  TrE=TeE=c()

for (j in1:kf)
  {
    TestD=d[o[((j-1)*(n/kf)+1):(j*(n/kf))],]
    TrainD=d[-o[((j-1)*(n/kf)+1):(j*(n/kf))],]
    YPredTrain=knnclass(k,TrainD,TrainD)
```

```
    YPredTest=knnclass(k,TrainD,TestD)
    T1=table(YPredTrain,TrainD[,p])
    TrE[j]=(sum(T1)-sum(diag(T1)))/sum(T1)
    T2=table(YPredTest,TestD[,p])
    TeE[j]=(sum(T2)-sum(diag(T2)))/sum(T2)
  }
  FinTrE[k]=mean(TrE)
  FinTeE[k]=mean(TeE)
}

M=max(c(FinTeE,FinTrE))+0.01
plot(1:5,FinTeE,type ="b",col="red",ylim =c(0,M))
lines(1:5,FinTrE,type ="b",col="black")
```



3) logistic regression

```
d1=D[,h]
d=cbind(d1,y)
p=ncol(D)
y=D[,p]#response variable
alpha=0.05
n=ncol(d)
```

```
for (i in1:n)
{
  lr=glm(y~d[,i],binomial(link="logit"))
  p[i]=summary(lr)$coefficients[2,4]
}

X1=d[,which(p<alpha)]
dim(X1)

## [1] 200 249
```

*out of 250 ,1 variable has p-value is greater than alpha hence it is not significant

4)support vector machine

```
library(e1071)

## Warning: package 'e1071' was built under R version 4.1.3

ts=sample(1:nrow(d),nrow(d)*0.7)
train=d[ts,]
test=d[-ts,]
svm1=svm(y~.,data=train)
summary(svm1)

##
## Call:
## svm(formula = y ~ ., data = train)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.004
##     epsilon:  0.1
##
##
## Number of Support Vectors:  116

yhat=round(predict(svm1,train))
table=table(yhat,train[,p])
table

##
## yhat  0  1
```

```
##    0 73 10
##    1  9 48

accuracy=sum(diag(table))/sum(table)
accuracy

## [1] 0.8642857
```

From table we can see that 10 healthy patient are misclassified as cancer patient and 9 cancer patient are misclassified as healthy patient

***Accuracy is 86.42%**

5) linear discriminant analysis

```
da=D1[,h]
q=c(rep(0,112),rep(1,88))
da1=cbind(da,q)



n=nrow(da1)
p=ncol(da1)
hp=d[1:112,] #healthy
cp=d[113:n,] #cancer
xb=apply(hp,2,mean)
yb=apply(cp,2,mean)
n1=nrow(hp)
n2=nrow(cp)
s1=(n1-1)*cov(hp)
s2=(n2-1)*cov(cp)
sigh=(s1+s2)/(n1+n2)   #sigma hat
#dis=t(xb-yb)%*%solve(sigh)%*%(xb-yb)
round(det(sigh))

## [1] 0
```

***As determinant of sigma matrix is zero i.e matrix is singular we can not do further analysis.**

REPORT

- First We have 3000 variables and 200 observations. It has too many variables with zero entries .
- First we remove the columns with most of the zero entries.So we get 15 variables which are not significant as all observations in that variables are zero. We have two responses one as healthy patients and another as cancer patients. We separate them and we get 112 healthy patients and 88 cancer patients.
- For further analysis, we want to classify the data into normal and non-normal data. To classify main data into normal and non-normal data we used Shapiro-Wilk Normality Test and we get 1480 non-normal variables and 16 normal variables.
- we want to check which variables are significant to explain the healthiness of patients. So to check it for normal variables we used T-test and for non-normal variables, we used the Wilcoxon test. From the testing we did, we got 9 significant variables out of 16 normal variables and 243 non-normal variables

which are significant. So from all the stuff we did, we got 250 variables.

- After doing some analysis of variables selection we finally get 250 variables from 3000 variables.
- On the variables that we have, we wish to fit some supervised learning models.
- Firstly we fit the KNN model for training and testing data divide as 50-50 per cent. From the KNN model, we saw that 10 healthy patients are misclassified as cancer patients and 9 cancer patients are misclassified as healthy patients.Hence the accuracy is 81%.
- We did a Logistic regression model after logistic regression out of 250 variables only one variable had a p-value greater than alpha (level of significance) hence only one variable is not significant.
- From support vector machine we got 86.42% accuracy.
- After doing the SVM we further did a linear discriminant analysis. For doing LDA we wish to check if the sigma matrix is zero or non zero and we got determinant

of sigma matrix is zero so we can not do further analysis.