# Task 4

for Advanced Methods for Regression and Classification

Teodor Chakarov

23.11.2022

## Contents



## Exercise 1

### Load data and preprocess

Let's load csv file

```
library(MASS)
library(cvTools)
```

```
## Loading required package: lattice
```

```
## Loading required package: robustbase
```

```
library(ggplot2)
library(ggfortify)
library(ipred)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(imbalance)
library(ROSE)
```

```
## Loaded ROSE 0.0-4
```

```
df <- read.csv("hcvdat1.csv")
str(df)
```

```
## 'data.frame':    587 obs. of  13 variables:
##  $ Category: chr  "BloodDonor" "BloodDonor" "BloodDonor" "BloodDonor" ...
##  $ Age     : int  32 32 32 32 32 32 32 32 32 32 ...
##  $ Sex     : chr  "m" "m" "m" "m" ...
##  $ ALB     : num  38.5 38.5 46.9 43.2 39.2 41.6 46.3 42.2 50.9 42.4 ...
##  $ ALP     : num  52.5 70.3 74.7 52 74.1 43.3 41.3 41.9 65.5 86.3 ...
##  $ ALT     : num  7.7 18 36.2 30.6 32.6 18.5 17.5 35.8 23.2 20.3 ...
##  $ AST     : num  22.1 24.7 52.6 22.6 24.8 19.7 17.8 31.1 21.2 20 ...
##  $ BIL     : num  7.5 3.9 6.1 18.9 9.6 12.3 8.5 16.1 6.9 35.2 ...
##  $ CHE     : num  6.93 11.17 8.84 7.33 9.15 ...
##  $ CHOL    : num  3.23 4.8 5.2 4.74 4.32 6.05 4.79 4.6 4.1 4.45 ...
##  $ CREA    : num  106 74 86 80 76 111 70 109 83 81 ...
##  $ GGT     : num  12.1 15.6 33.2 33.8 29.9 91 16.9 21.5 13.7 15.9 ...
##  $ PROT    : num  69 76.5 79.3 75.7 68.7 74 74.5 67.1 71.3 69.9 ...
```

We have 11 numeric attributes, and 2 categorical once. We are going to drop the NaN values and set the categorical to factors.

```
df <- na.omit(df)

df$Sex <- as.numeric(as.factor(df$Sex))
```

```
str(df)
```

```
## 'data.frame':    570 obs. of  13 variables:
##  $ Category: chr  "BloodDonor" "BloodDonor" "BloodDonor" "BloodDonor" ...
##  $ Age     : int  32 32 32 32 32 32 32 32 32 32 ...
##  $ Sex     : num  2 2 2 2 2 2 2 2 2 2 ...
##  $ ALB     : num  38.5 38.5 46.9 43.2 39.2 41.6 46.3 42.2 50.9 42.4 ...
##  $ ALP     : num  52.5 70.3 74.7 52 74.1 43.3 41.3 41.9 65.5 86.3 ...
##  $ ALT     : num  7.7 18 36.2 30.6 32.6 18.5 17.5 35.8 23.2 20.3 ...
##  $ AST     : num  22.1 24.7 52.6 22.6 24.8 19.7 17.8 31.1 21.2 20 ...
##  $ BIL     : num  7.5 3.9 6.1 18.9 9.6 12.3 8.5 16.1 6.9 35.2 ...
##  $ CHE     : num  6.93 11.17 8.84 7.33 9.15 ...
##  $ CHOL    : num  3.23 4.8 5.2 4.74 4.32 6.05 4.79 4.6 4.1 4.45 ...
```

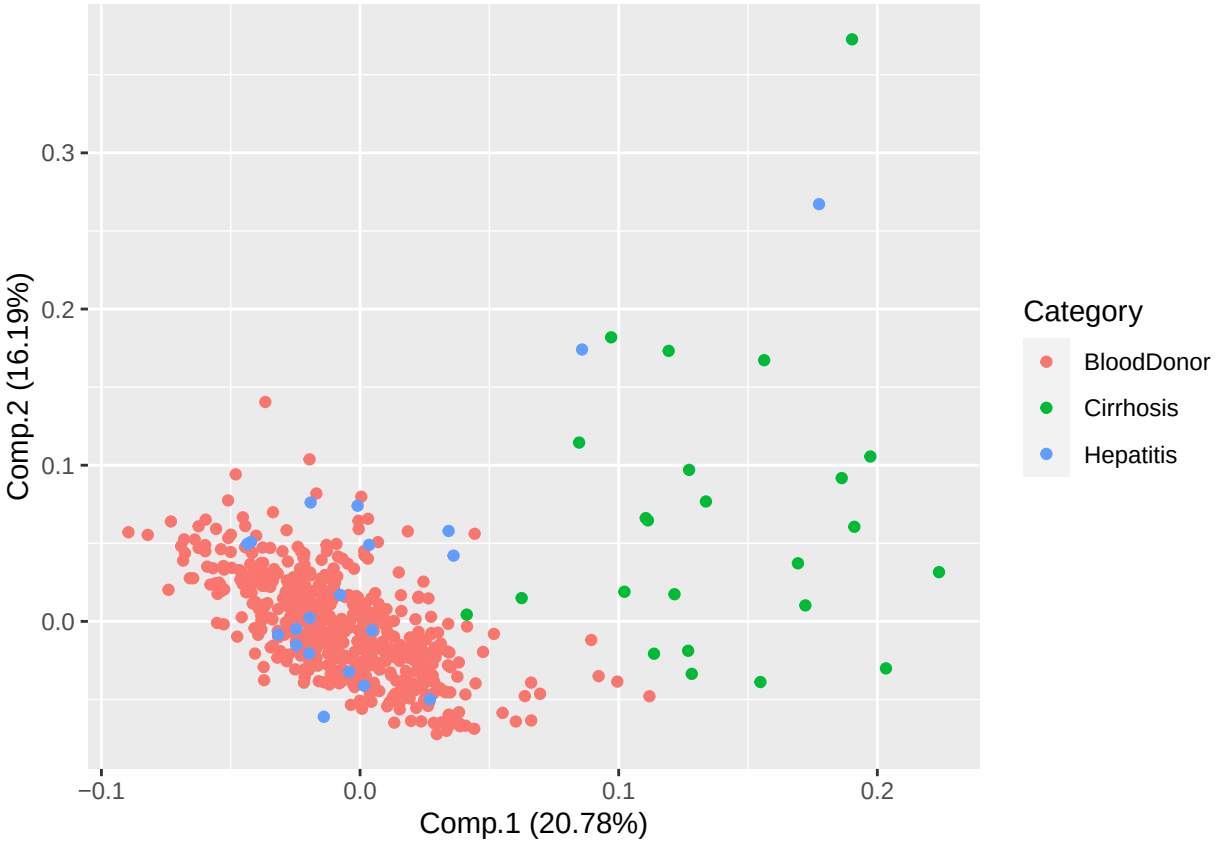
```
##  $ CREA    : num  106 74 86 80 76 111 70 109 83 81 ...
##  $ GGT     : num  12.1 15.6 33.2 33.8 29.9 91 16.9 21.5 13.7 15.9 ...
##  $ PROT    : num  69 76.5 79.3 75.7 68.7 74 74.5 67.1 71.3 69.9 ...
##  - attr(*, "na.action")= 'omit' Named int [1:17] 122 320 330 414 425 434 499 534 535 539 ...
##   ..- attr(*, "names")= chr [1:17] "122" "320" "330" "414" ...
```

## Compute PCA

```
x <- df[, -which(names(df) %in% c("Category"))]

mean_train <- apply(x, 2, mean)
sd_train <- apply(x, 2, sd)

x_scaled <- (data.frame(scale(x, center=mean_train, scale = sd_train)))

# Apply PCA
comps <- princomp(x_scaled, cor=TRUE, scores=TRUE)

comps
```

```
## Call:
## princomp(x = x_scaled, cor = TRUE, scores = TRUE)
##
## Standard deviations:
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7    Comp.8
## 1.5790699 1.3939464 1.1909117 1.0607443 0.9984910 0.9353799 0.8465485 0.8029830
##    Comp.9   Comp.10   Comp.11   Comp.12
## 0.7407590 0.6880233 0.6557861 0.5783609
##
##  12  variables and  570 observations.
```

```
autoplot(comps, df, colour = 'Category')
```




We can see here the clusters of the given classes. The red dots for Blood Donor are distinguishable with cirrhosis. The Hepatitis on the other hand are within the blood doner cluster. We would need here another dimension because we can't draw a hyper-plane which can separate the data as it is now. Our classification model can't calculate the separation point.

### Task 2: Fitting in models

Splitting the data to train and test.

```
set.seed(1234555)
row_Count <- floor(round(nrow(df)*2.0/3))
train_Data <- sample(seq_len(nrow(df)), size = row_Count)
```

```
df$Category <- as.factor(df$Category)

train <- df[train_Data, ]
test <- df[-train_Data, ]

y_train = train[ , which(names(train) %in% c("Category"))]
y_test = test[ , which(names(test) %in% c("Category"))]

x_train = train[ , -which(names(train) %in% c("Category"))]
x_test = test[ , -which(names(test) %in% c("Category"))]

summary(x_train)
```

```
##       Age             Sex             ALB             ALP
##  Min.   :23.00   Min.   :1.000   Min.   :23.00   Min.   : 19.10
##  1st Qu.:39.00   1st Qu.:1.000   1st Qu.:38.70   1st Qu.: 55.90
##  Median :47.00   Median :2.000   Median :42.00   Median : 67.25
##  Mean   :47.11   Mean   :1.624   Mean   :41.79   Mean   : 69.70
##  3rd Qu.:54.00   3rd Qu.:2.000   3rd Qu.:45.30   3rd Qu.: 80.28
##  Max.   :76.00   Max.   :2.000   Max.   :82.20   Max.   :416.60
##       ALT              AST              BIL              CHE
##  Min.   :  0.90   Min.   : 12.20   Min.   :  2.00   Min.   : 1.420
##  1st Qu.: 16.98   1st Qu.: 21.68   1st Qu.:  5.30   1st Qu.: 6.973
##  Median : 23.40   Median : 25.55   Median :  7.30   Median : 8.320
##  Mean   : 26.35   Mean   : 33.50   Mean   : 12.01   Mean   : 8.228
##  3rd Qu.: 33.10   3rd Qu.: 31.30   3rd Qu.: 10.82   3rd Qu.: 9.610
##  Max.   :118.10   Max.   :324.00   Max.   :209.00   Max.   :15.430
##       CHOL            CREA             GGT             PROT
##  Min.   :1.430   Min.   :  8.00   Min.   :  7.00   Min.   :51.0
##  1st Qu.:4.630   1st Qu.:  68.00   1st Qu.: 15.68   1st Qu.:69.3
##  Median :5.350   Median :  78.00   Median : 22.90   Median :72.2
##  Mean   :5.464   Mean   :  84.32   Mean   : 37.47   Mean   :72.1
##  3rd Qu.:6.240   3rd Qu.:  89.03   3rd Qu.: 36.20   3rd Qu.:75.2
##  Max.   :9.670   Max.   :1079.10   Max.   :650.90   Max.   :86.0
```

**Linear Discrimant Analysis**

```
mod.lda<-lda(y_train ~ .,data=x_train)
predict.lda <- predict(mod.lda, x_test)$class
(TAB<-table(y_test,predict.lda ))
```

```
##             predict.lda
## y_test       BloodDonor Cirrhosis Hepatitis
##   BloodDonor        179         0         0
##   Cirrhosis           1         3         2
##   Hepatitis           3         0         2
```

We can see when we train without CV, that on test data we have Cirrhosis which is the most miss classified. Let's check now the miss classification error.

```
lda.cv <- lda(y_train~.,data=x_train,CV=TRUE)
(TAB <- table(y_train,lda.cv$class))
```

```
##
## y_train      BloodDonor Cirrhosis Hepatitis
##   BloodDonor        346         0         1
##   Cirrhosis           1        14         3
##   Hepatitis           7         1         7
```

```
print(paste0("Misclassification rate of CV: ", 1-sum(diag(TAB))/sum(TAB)))
```

```
## [1] "Misclassification rate of CV: 0.0342105263157895"
```

We can see that we don't have big miss classification error on both training and testing data.

### Quadratic Discrimant Analysis

```
qda.cv <- qda(y_train~.,data=x_train,CV=TRUE)
(TAB <- table(y_train,qda.cv$class))
```

```
##
## y_train      BloodDonor Cirrhosis Hepatitis
##   BloodDonor        346         1         0
##   Cirrhosis           1        16         1
##   Hepatitis           9         3         3
```

```
print(paste0("Misclassification rate of CV: ", 1-sum(diag(TAB))/sum(TAB)))
```

```
## [1] "Misclassification rate of CV: 0.0394736842105263"
```

I am getting bigger miss classification error with the qda model.

```
table(y_train)
```

```
## y_train
## BloodDonor  Cirrhosis  Hepatitis
##        347         18         15
```

```
table(y_test)
```

```
## y_test
## BloodDonor  Cirrhosis  Hepatitis
##        179          6          5
```

Within couple of times qda returned an error which was "some group is too small for 'qda' " In order to make it work i need to stratify split the data.

## Impoving

### Oversample

Since we have heavily imbalanced data in regard of the classes, we can split the date with stratify method, which is diving the train and test data based on proportionally separation of the classes in the sets.

Another technique can be under or over-sampling. In this case we can make the unbalanced data more fair when we train and give equal amount of examples on the training model.

```
df_no_hepa <- df[df$Category != "Hepatitis",]
df_no_cirr <- df[df$Category != "Cirrhosis",]
```

```
data_no_Hepa <- df_no_hepa[df_no_hepa$Category != "Hepatitis",]
data_cirrsample <- ROSE(Category ~ ., data=data_no_Hepa, seed=1234)$data

data_no_Cirr <- df_no_cirr[df_no_cirr$Category != "Cirrhosis",]
data_heppasample <- ROSE(Category ~ ., data=data_no_Cirr, seed=1234)$data

data_cirr <- data_cirrsample[data_cirrsample$Category == "Cirrhosis",]
data_rose <- rbind(data_heppasample, data_cirr)
table(data_rose$Category)
```

```
##
## BloodDonor  Hepatitis  Cirrhosis
##        260        286        290
```

Now we have equal amount of classes and lets fit it in lda model:

```
lda_cv_rose <- lda(Category~.,data=data_rose,CV=TRUE)
(TAB <- table(data_rose$Category,lda_cv_rose$class))
```

```
##
##              BloodDonor Hepatitis Cirrhosis
##   BloodDonor        224        36         0
##   Hepatitis          62       211        13
##   Cirrhosis           8        16       266
```

```
print(paste0("Misclassification rate of CV with rose: ", 1-sum(diag(TAB))/sum(TAB)))
```

```
## [1] "Misclassification rate of CV with rose: 0.161483253588517"
```

Our miss classification error raised at 18% but that's with almost 94 times more generated samples.

### Dimensionallity reduction

```
prin_comps <- princomp(x_train)
data_pca <- prin_comps$scores[,1:9]
data_pca <- as.data.frame(data_pca)
```

```
lda_cv_pca <- lda(y_train~.,data=data_pca,CV=TRUE)
(TAB <- table(lda_cv_pca$class,y_train))
```

```
##             y_train
##              BloodDonor Cirrhosis Hepatitis
##   BloodDonor        346         1         7
##   Cirrhosis           0        14         1
##   Hepatitis           1         3         7
```

```
print(paste0("Misclassification rate of CV for QDA with principal componants: ", 1-sum(diag(TAB))/sum(TA
```

```
## [1] "Misclassification rate of CV for QDA with principal componants: 0.0342105263157895"
```

After the PCA we have little bit less of an error and also more True positives of the Hepatitis.