Inteligencja obliczeniowa

Zagłębianie danych

Julian Podleśny, 235333

Informatyka I rok, mgr

Grupa 3

1. Wstęp

Do wykonania projektu została wykorzystana baza danych znajdująca się na stronie: https://archive.ics.uci.edu/ml/datasets/Hepatitis. Baza ta zawiera informacje na temat wpływu choroby, jaką jest zapalenie wątroby, na śmiertelność ludzi.

Wyodrębnić można dwadzieścia kolumn:
Class – kolumna, która najbardziej nadaje się na klasę, ze względu na występujące tam dwie wartości – die oraz live,
Age – wiek osoby
Sex – płeć
Steroid – czy w organizmie były obecne steroidy
Antivirals – czy przyjmowane były leki przeciwwirusowe
Fitigue – czy występowało zmęczenie
Malaise – czy występowało złe samopoczucie
Anorexia – czy występowała anoreksja
Liver.big – czy wątroba była powiększona
Liver.firm – czy wątroba była twarda
Spleen.palpable – czy śledziona była powiększona
Spiders – czy występowały pajączki naczyniowe
Ascites – czy występowało wodobrzusze
Varices – czy występowały żylaki
Bilirubin - stężenie bilirubiny we krwi
Alk.phosphate – ilość fosfatazy alkalicznej we krwi
Sgot – ilość aminotransferazy asparaginianowej we krwi
Albumin – ilość albuminy we krwi
Protime – czas protrombinowy, mierzony w sekundach (czas w którym krew tworzy skrzep)
Histology – czy było robione badanie mikroskopowe budowy ciała

2. Obróbka bazy danych

Z racji tego, iż baza danych zawiera brakujące wartości (oznaczone jako „?”), należy je uzupełnić. W tym celu posłużę się techniką najbliższych sąsiadów (k-Nearest Neighbor). Wszystkie kroki zostały przedstawione poniżej:

1. Wczytanie bazy danych z pliku.

```
hepatitis <- read.csv("hepatitis.csv", header=TRUE, sep=",", stringsAsFactors = FALSE)
```

2. Przypisanie „brudnej” bazy do nowej zmiennej.

```
hepatitis.clear <-hepatitis
```

3. Zmiana wartości „?” na NA

```
hepatitis.clear[, c(4,6:19)][hepatitis.clear[, c(4,6:19)] == "?"] <- NA
```

4. Zainstalowanie/wczytanie bilioteki VIM

```
library("VIM”)
```

5. Zastosowanie kNN

```
hepatitis.clear <- kNN(hepatitis.clear)
```

6. Usunięcie zbędnych kolumn

```
hepatitis.clear[, c(21:40)] <- NULL
```

7. Zmiana cyfr na odpowiadającą im wartość tekstową

```
hepatitis.clear$Class[hepatitis.clear$Class == 1] <- "die"
hepatitis.clear$Class[hepatitis.clear$Class == 2] <- "live"
hepatitis.clear$SEX[hepatitis.clear$SEX == 1] <- "male"
hepatitis.clear$SEX[hepatitis.clear$SEX == 2] <- "female"
```

8. Zmiana typów kolumn oraz ich wartości

```
hepatitis.clear[,1] <- factor(hepatitis.clear [,1])
hepatitis.clear[, 4] <- as.numeric(hepatitis.clear[,4])
hepatitis.clear[, 4] [hepatitis.clear[, 4] == 2] <- 0
hepatitis.clear[, 4] [hepatitis.clear[, 4] == 3] <- 1

for (i in 6:14) {
  hepatitis.clear[, i] <- as.numeric(hepatitis.clear[,i])
        for(j in 2:nrow(hepatitis.clear)) {
              hepatitis.clear[, i] [hepatitis.clear[, i] == 2] <- 0
              hepatitis.clear[, i] [hepatitis.clear[, i] == 3] <- 1
        }
}
hepatitis.clear[, 20] [hepatitis.clear[, 20] == 1] <- 0
hepatitis.clear[, 20] [hepatitis.clear[, 20] == 2] <- 1
```

Fragment bazy danych przed zmianami:

	Class	AGE	SEX	STEROID
1	2	30	2	1
2	2	50	1	1
3	2	78	1	2
4	2	31	1	?
5	2	34	1	2

I po:

	Class	AGE	SEX	STEROID
1	live	30	female	0
2	live	50	male	0
3	live	78	male	1
4	live	31	male	1
5	live	34	male	1

3. Klasyfikatory i ich ewaluacja

Wyczyszczoną i uzupełnioną bazę danych należy teraz podzielić na zbiór treningowy (na którym trenujemy model) oraz zbiór testowy (na którym ewaluujemy każdy klasyfikator). W tym celu do projektu trzeba podłączyć bibliotekę „party”.

```
library("party")

set.seed(1234)
ind <- sample(2, nrow(hepatitis.clear), replace=TRUE, prob=c(0.67, 0.33))
hepatitis.training <- hepatitis.clear[ind==1,1:20]
hepatitis.test <- hepatitis.clear[ind==2,1:20]
```

a) Klasyfikacja za pomocą drzewa

```
hepatitis.ctree <- ctree(Class ~ ., data=hepatitis.training)
predicted.tree <- predict(hepatitis.ctree, hepatitis.test[,2:20])
real.tree <- hepatitis.test[,1]
conf.matrix.tree <- table(predicted.tree,real.tree)
accuracy.tree <- sum(diag(conf.matrix.tree))/sum(conf.matrix.tree)
```

Macierz błędu:

```
> conf.matrix.tree
              real.tree
predicted.tree die live
          die    7    4
          live   6   34
```

Po przeanalizowaniu wyżej przedstawionej macierzy błędu można postawić następujące wnioski:

1. TP (ang. *true positive*) – prawdziwie pozytywna: wartość wynosi 7, co można tłumaczyć jako: 7 osób poprawnie uznano jako nieżyjące
2. FP (ang. *false positive*) – fałszywie pozytywna: wartość wynosi 4: 4 osoby żyjące uznano za nieżyjące
3. TN (ang. *true negative*) – prawdziwie negatywna: wartość wynosi 34: 34 osoby poprawnie uznano jako żyjące
4. FN (ang. *false negative*) – fałszywie negatywna: wartość wynosi 6: 6 osób nieżyjących uznano jako żyjące
5. TPR = TP/(TP + FN) = 0.53846153846

6. FPR = FP/(FP + TN) = 0.10526315789
7. FNR = 1 – TPR = 0.46153846154
8. TNR = 1 – FPR = 0.89473684211
9. Błąd pierwszego rodzaju: FP
10. Błąd drugiego rodzaju: FN
11. Im więcej błędów pierwszego rodzaju, tym mniejszy TNR, a tym większy FPR. Im więcej błędów drugiego rodzaju, tym mniejszy TPR, a większy FNR.
12. Uważam, że gorszym do popełnienia jest błąd pierwszego rodzaju. Na przykładzie wykorzystanej przeze mnie bazy: osoby, które są żyjące, zostaną zakwalifikowani jako nieżyjące, co może być tragiczne w skutkach, w przypadku załatwiania na przykład spraw urzędowych.

Dokładność:

```
> accuracy.tree
[1] 0.8039216
```

b) Naive Bayes

```
install.packages("e1071")
library("e1071")

hepatitis.nBayes <- naiveBayes(Class ~ ., data = hepatitis.training)
predicted.bayes <- predict(hepatitis.nBayes, hepatitis.test[,2:20])
real.bayes <- hepatitis.test[,1]
conf.matrix.bayes <- table(predicted.bayes,real.bayes)
accuracy.bayes <- sum(diag(conf.matrix.bayes))/sum(conf.matrix.bayes)
```

Macierz błędu:

```
> conf.matrix.bayes
                real.bayes
predicted.bayes die live
           die    8    6
           live   5   32
```

Po przeanalizowaniu wyżej przedstawionej macierzy błędu można postawić następujące wnioski:

1. TP (ang. *true positive*) – prawdziwie pozytywna: 8 osób poprawnie uznano za nieżyjące

2. FP (ang. *false positive*) – fałszywie pozytywna: 6 osoby żyjących uznano za nieżyjące
3. TN (ang. *true negative*) – prawdziwie negatywna: 32 osoby poprawnie uznano za żyjące
4. FN (ang. *false negative*) – fałszywie negatywna: 5 osób nieżyjących uznano za żyjące
5. TPR = TP/(TP + FN) = 0.61538461538
6. FPR = FP/(FP + TN) = 0.15789473684
7. FNR = 1 – TPR = 0.38461538462
8. TNR = 1 – FPR = 0.84210526316

Dokładność:

```
> accuracy.bayes
[1] 0.7843137
```

c) k-Najbliższych Sąsiadów, k = 3

```
predicted.knn <- knn(hepatitis.training[,c(2,4:20)], hepatitis.test[,c(2,4:20)],
cl=hepatitis.training[,1], k=3, prob=FALSE)

real.knn <- hepatitis.test[,1]
conf.matrix.knn <- table(predicted.knn,real.knn)
accuracy.knn <- sum(diag(conf.matrix.knn))/sum(conf.matrix.knn)
```

Macierz błędu:

```
> conf.matrix.knn
             real.knn
predicted.knn die live
         die    5    1
         live   8   37
```

Po przeanalizowaniu wyżej przedstawionej macierzy błędu można postawić następujące wnioski:

1. TP (ang. *true positive*) – prawdziwie pozytywna: 5 osób poprawnie uznano za nieżyjące
2. FP (ang. *false positive*) – fałszywie pozytywna: 1 osobę żyjącą uznano za nieżyjącą
3. TN (ang. *true negative*) – prawdziwie negatywna: 37 osób poprawnie uznano za żyjące

4. FN (ang. *false negative*) – fałszywie negatywna: 8 osób nieżyjących uznano za żyjące
5. TPR = TP/(TP + FN) = 0.38461538461
6. FPR = FP/(FP + TN) = 0.02631578947
7. FNR = 1 – TPR = 0.61538461539
8. TNR = 1 – FPR = 0.97368421053

Dokładość:

```
> accuracy.knn
[1] 0.8235294
```

d) Metoda wektorów podpierających

```
install.packages("kernlab")
library("kernlab")

hepatitis.ksvm <- ksvm(Class ~ ., data = hepatitis.training)
predicted.ksvm <- predict(hepatitis.ksvm, hepatitis.test[,2:20])
real.ksvm <- hepatitis.test[,1]
conf.matrix.ksvm <- table(predicted.ksvm,real.ksvm)
accuracy.ksvm <- sum(diag(conf.matrix.ksvm))/sum(conf.matrix.ksvm)
```

Macierz błędu:

```
> conf.matrix.ksvm
                real.ksvm
predicted.ksvm die live
          die    5    4
          live   8   34
```

Po przeanalizowaniu wyżej przedstawionej macierzy błędu można postawić następujące wnioski:

1. TP (ang. *true positive*) – prawdziwie pozytywna: 5 osób poprawnie uznano za nieżyjące
2. FP (ang. *false positive*) – fałszywie pozytywna: 4 osoby żyjące uznano za nieżyjące
3. TN (ang. *true negative*) – prawdziwie negatywna: 34 osoby poprawnie uznano za żyjące
4. FN (ang. *false negative*) – fałszywie negatywna: 8 osób nieżyjących uznano za żyjące
5. TPR = TP/(TP + FN) = 0.38461538461

6. FPR = FP/(FP + TN) = 0.10526315789
7. FNR = 1 – TPR = 0.61538461539
8. TNR = 1 – FPR = 0.89473684211

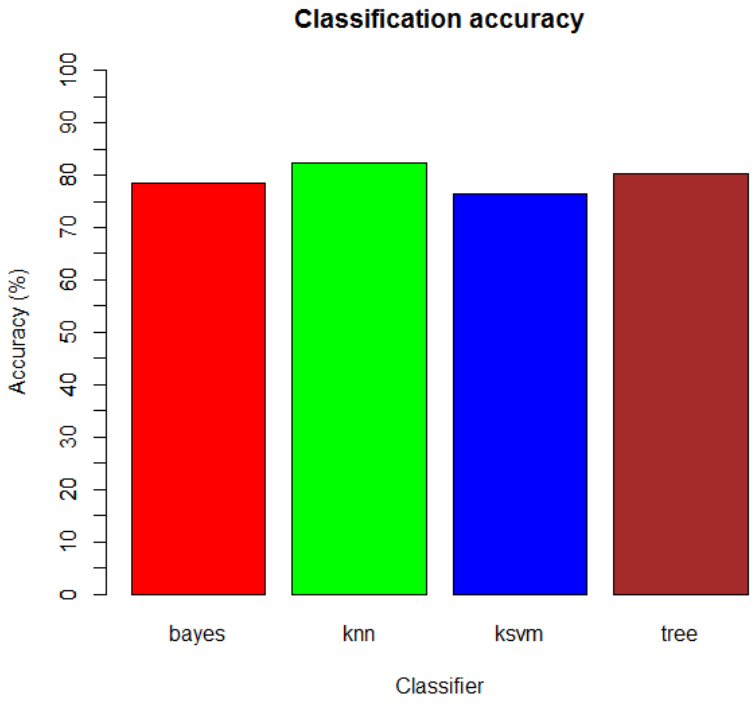
Dokładność:

```
> accuracy.ksvm
[1] 0.7647059
```

e) Podsumowanie klasyfikatorów

Zestawienie dokładności wszystkich klasyfikatorów powyżej na wykresie słupkowym.

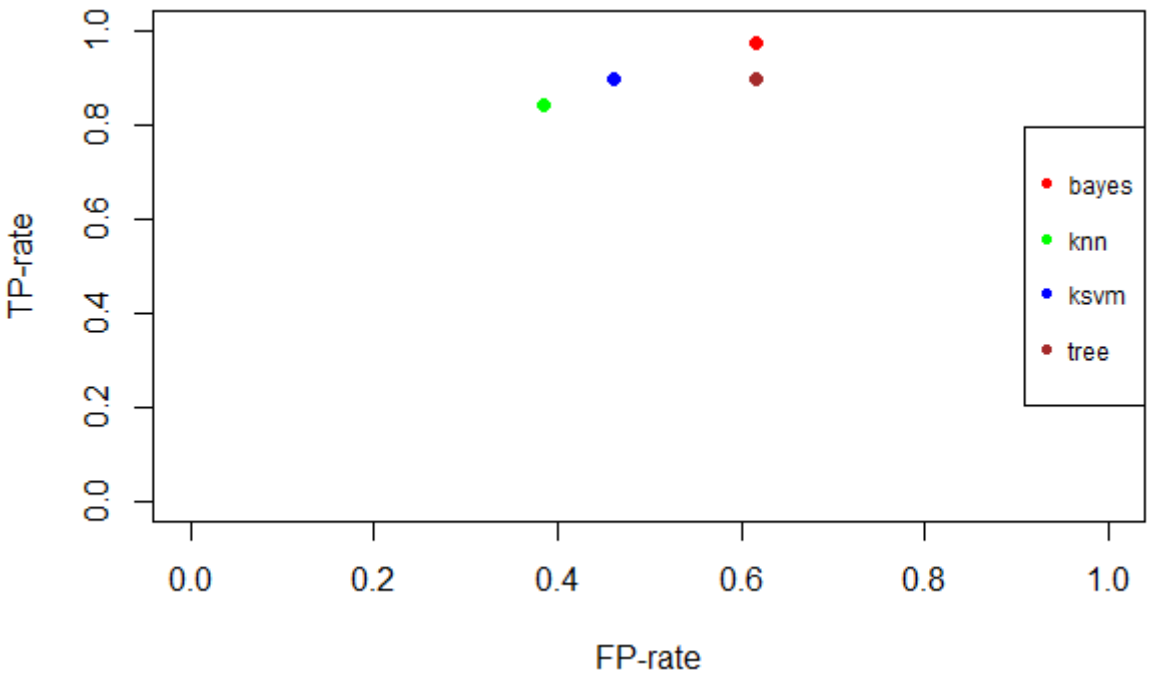
```
colors = c("red", "green", "blue", "brown")

barplotData <-
c(accuracy.bayes*100,accuracy.knn*100,accuracy.ksvm*100,accuracy.tree*100)

barplot(barplotData, names.arg = c("bayes","knn","ksvm","tree"),  xlab =
"Classifier", ylab = "Accuracy (%)", col = colors, main = "Classification accuracy",
ylim = c(0,100)) axis(2, seq(0, 100, by = 5))
```



Najlepszy rezultat dla naszej bazy daje klasyfikator kNN (82% dokładności), zaraz za nim drzewo (80%), Naive Bayes (78%) oraz ksvm (76%).

Obliczenie TPR i FPR dla każdego klasyfikatora powyżej oraz zestawienie wyników na wykresie.

```
knnPos <- c(table(hepatitis.knn, hepatitis.test[,1])[2,1]/(table(hepatitis.knn,
hepatitis.test[,1])[2,1]+table(hepatitis.knn, hepatitis.test[,1])[1,1]),
        table(hepatitis.knn, hepatitis.test[,1])[2,2]/(table(hepatitis.knn,
hepatitis.test[,1])[2,2]+table(hepatitis.knn, hepatitis.test[,1])[1,2]))

bayesPos <- c(table(predicted.bayes,
hepatitis.test[,1])[2,1]/(table(predicted.bayes,
hepatitis.test[,1])[2,1]+table(predicted.bayes, hepatitis.test[,1])[1,1]),
        table(predicted.bayes, hepatitis.test[,1])[2,2]/(table(predicted.bayes,
hepatitis.test[,1])[2,2]+table(predicted.bayes, hepatitis.test[,1])[1,2]))

treePos <- c(table(predicted.tree, hepatitis.test[,1])[2,1]/(table(predicted.tree,
hepatitis.test[,1])[2,1]+table(predicted.tree, hepatitis.test[,1])[1,1]),
        table(predicted.tree, hepatitis.test[,1])[2,2]/(table(predicted.tree,
hepatitis.test[,1])[2,2]+table(predicted.tree, hepatitis.test[,1])[1,2]))

ksvmPos <- c(table(predicted.ksvm, hepatitis.test[,1])[2,1]/(table(predicted.ksvm,
hepatitis.test[,1])[2,1]+table(predicted.ksvm, hepatitis.test[,1])[1,1]),
        table(predicted.ksvm, hepatitis.test[,1])[2,2]/(table(predicted.ksvm,
hepatitis.test[,1])[2,2]+table(predicted.ksvm, hepatitis.test[,1])[1,2]))

posList <- c(bayesPos, knnPos, ksvmPos, treePos)

plotData <- matrix(posList, nrow=4, ncol=2, byrow = TRUE)

plot(-1,-1, xlim = c(0,1), ylim = c(0,1), xlab = "FP-rate", ylab = "TP-rate")

for (i in 1:nrow(plotData)) {
  points(plotData[i,1], plotData[i,2], col = colors[i], pch = 16)
}

legend("right","top", legend = c("bayes","knn","ksvm","tree"), cex=.75, col =
colors, pch = 16, lty = 0)
```



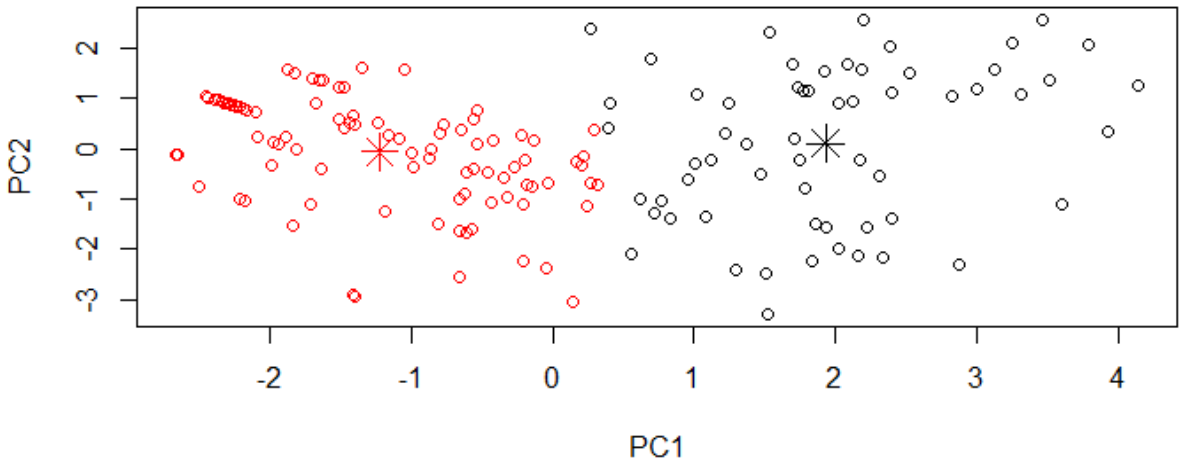
Punkt dla idealnego klasyfikatora leżałby w lewym górnym rogu. Myślę, że najbliższe ideałowi byłoby kNN, z tego względu, iż jego współczynnik FPR wynosi najmniej, co za tym idzie mniej błędów pierwszego rodzaju.

4. Grupowanie metodą k-średnich

```
hepatitis.log <- log(hepatitis.clear[,c(2,4:14,20)])
hepatitis.log  <- do.call(data.frame, lapply(hepatitis.log, function(x) {
        replace(x, is.infinite(x) | is.na(x), -1)
}))

hepatitis.log.scale <- scale(hepatitis.log)
hepatitis.pca <- prcomp(hepatitis.log.scale)

hepatitis.pca$x
new.dat <- data.frame("PC1" = hepatitis.pca$x[,1], "PC2" = hepatitis.pca$x[,2])
```

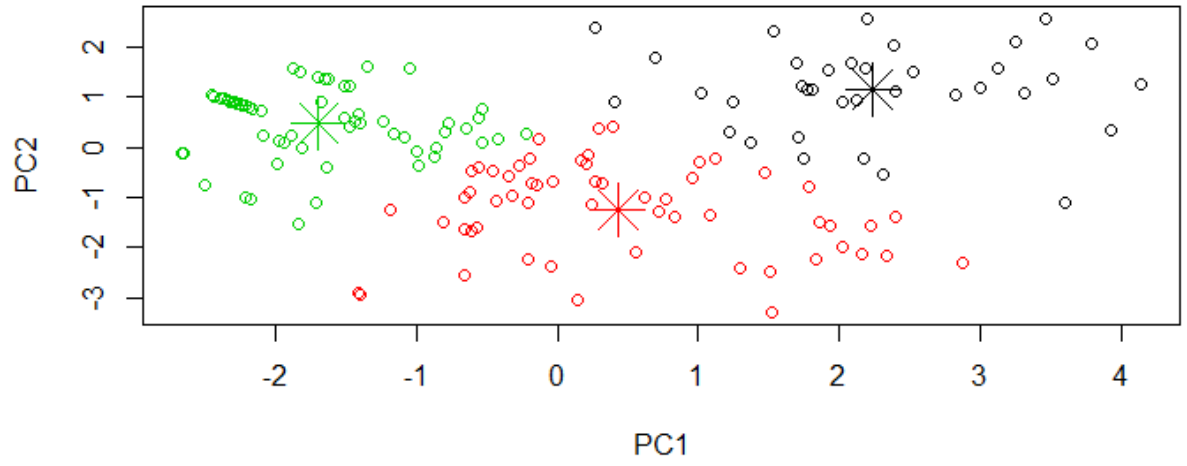
a) Dzielenie na 2 klastry

```
cl2 <- kmeans(new.dat, 2)
plot(new.dat, col = cl2$cluster)
points(cl2$centers, col = 1:2, pch = 8, cex = 2)
```



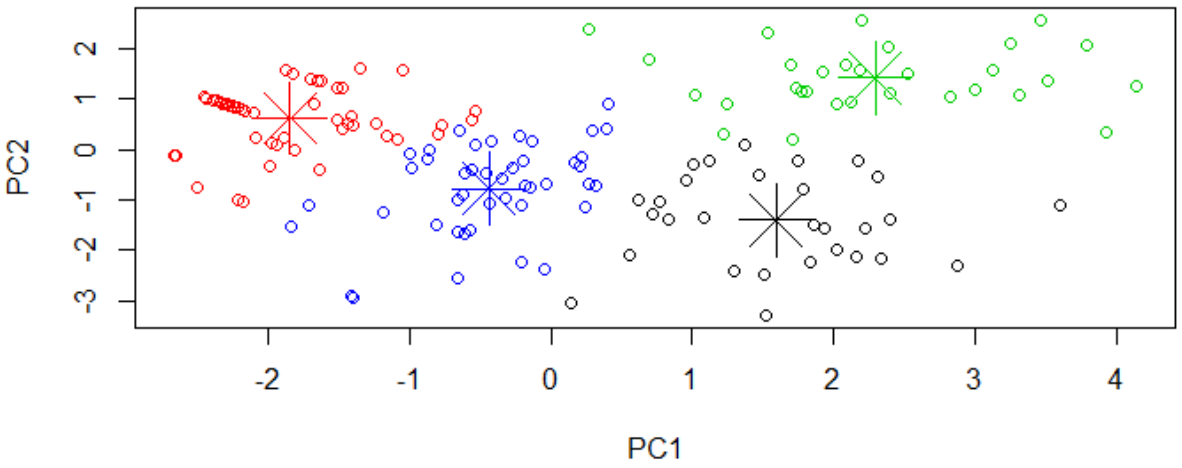
b) Dzielenie na 3 klastry

```
cl3 <- kmeans(new.dat, 3)
plot(new.dat, col = cl3$cluster)
points(cl3$centers, col = 1:3, pch = 8, cex = 3)
```



c) Dzielenie na 4 klastry

```
cl4 <- kmeans(new.dat, 4)
plot(new.dat, col = cl4$cluster)
points(cl4$centers, col = 1:4, pch = 8, cex = 4)
```



Wszystkie klastry, przedstawione powyżej w podziałach, zawierają w sobie takie same rekordy. Algorytm zdołał podzielić na trzy i cztery klastry. Bliżej lewego górnego rogu znajduje się skupisko punktów. Po sprawdzeniu, jakie osoby się tam znajdują można stwierdzić, że są to żyjący mężczyźni, którzy mieli obecne steroidy w organizmie, przyjmowali leki przeciwwirusowe, występowało u nich zmęczenie, anoreksja oraz reszta objawów zawarta w bazie. Co więcej każdy z tych mężczyzn nie był badany mikroskopowo. Punkt, który znajduje się najbardziej po prawej stronie to nieżyjący mężczyzna. Skupisko punktów znajdujące się po lewej stronie od zielonego centroida to również nieżyjący mężczyźni.

5. Reguły asocjacyjne

```
library(arules)
```

Z racji iż funkcja apriori wyświetla komunikat błędu, należy przerobić bazę hepatitis.clear, zmieniając typy kolumn 2, 4-14 oraz 20.

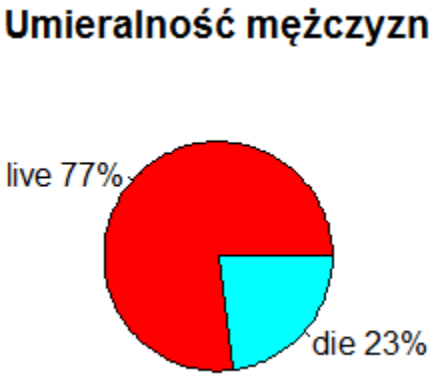
```
rules <- apriori(hepatitis.clear, parameter = list(minlen = 2, supp=0.1, conf=0.8),
appearance = list(rhs=c("Class=live", "Class=die"), default="lhs"),
control=list(verbose=F))

rules.sorted <- sort(rules, by="lift")
subset.matrix <- is.subset(rules.sorted, rules.sorted)
subset.matrix[lower.tri(subset.matrix, diag=T)] <- FALSE
redundant <- colSums(subset.matrix, na.rm) >= 1
rules.pruned <- rules.sorted[!redundant]
```

```
      lhs                    rhs            support confidence     lift count
[1]   {SEX=female}        => {Class=live} 0.1032258  1.0000000 1.260163    16

[34]  {ANOREXIA,
       LIVER.FIRM,
       SPIDERS,
       ASCITES,
       VARICES,
       HISTOLOGY}         => {Class=live} 0.1032258  1.0000000 1.260163    16
[148] {ANTIVIRALS,
       ANOREXIA,
       LIVER.BIG,
       ASCITES,
       VARICES}           => {Class=live} 0.4258065  0.9166667 1.155149    66
```

[1] Dużą szansę na przeżycie zapalenia wątroby mają kobiety. Z bazy danych użytej w tym projekcie wynika, że każda osoba, która jest kobietą – żyje. [34] Osoba, która ma stwierdzoną anoreksję, powiększoną wątrobę, pajączki naczyniowe, wodobrzusze, żylaki i miała robione badanie mikroskopowe budowy ciała będzie żyła. [148] Jeśli ktoś przyjmuje leki przeciwwirusowe, ma anoreksję, powiększoną wątrobę, wodobrzusze oraz żylaki, jest duża szansa, że przeżyje.

6. Diagramy


Ilość kobiet i mężczyzn
male 90%
female 10%


Umieralność mężczyzn
live 77%
die 23%


Umieralność kobiet
live 100%
die 0%

7. Podsumowanie

Z klasyfikatorów największą dokładnością wykazało się kNN. Jednakże dokładność na poziomie 82% nie jest do końca zadowalająca. Najgorszym klasyfikatorem okazał się ksvm z dokładnością 76%. Przy pracy z danymi medycznymi na pewno ciekawym rozwiązaniem są reguły asocjacyjne. Dzięki nim, można dowiedzieć się jakie są szanse przeżycia osoby mającej dane objawy. Z ciekawostek można dodać, że w użytej bazie danych wszystkie kobiety przeżyły zapalenie wątroby.