Badanie bazy danych używek

Autor: Mikołaj Błaszczyk

Technologia: Język programowania R

Data: 16.01.2020

Spis treści

1. WSTĘP

Opisywany przeze mnie projekt polega na przeanalizowaniu bazy danych zawierającej informacje zebrane od różnych osób na temat używek jakich zażywały w swoim życiu.

Znajdziemy tutaj używki od alkoholu i papierosów do twardych narkotyków.

Chciałbym na tym zbiorze danych przeanalizować czy na podstawie zebranych parametrów, dałoby się przewidzieć prawdopodobieństwo zażycia danej używki przez ankietowanego.

Jako, że wszystkich używek jest bardzo dużo to postanowiłem skupić się na jednej – alkoholu. Dlaczego?

Powodów jest kilka. Jeden jest taki, że alkohol jest najbardziej dostępny ze względu na legalność w przeciwieństwie do narkotyków. Drugi powód jest taki, że alkoholizm jest bardzo częstą chorobą, o której się nie mówi za wiele, a być może taka analiza mogłaby nam pomóc w identyfikowani osób, które mogą być narażone na tę chorobę.

2. NARZĘDZIA

Przy tworzeniu tego projektu wykorzystałem wiele technologii i pakietów, które postaram się w celu ułatwienia wymienić w tym dziale.

Ważne do odnotowania jest to, że całość programu została stworzona w języku R przy pomocy programu R Studio.

```
1.      Wczytywanie danych i rysowanie wykresów:
2.      library(readr)
3.      library(plotrix)
4.
5.      Algorytmy:
6.    library(gbm)            Gradniet Boosting Machine
7.    library(party)          Drzewa decyzyjne
8.    library(partykit)       Drzewa decyzyjne (Nowsze)
9.    library(class)          KNN
10.   library(e1071)          Algorytm Naivebayes
11.   library(naivebayes)  Algorytm Naivebayes (Nowszy)
12.   library(randomForest) Algorytm Losowego Lasu
13.   library(neuralnet)    Sieć Neuronowa prosta
14.   library(editrules)    Algorytm K-średnich
15.   library(factoextra)   Algorytm K-meoid
16.   library(arules)        Reguły Asocjacyjne
17.   library(MASS)      Liniowa i Kwadratowa Analiza
18.   library(VGAM)           Regresja Liniowa
19.   library(mda)            Elastyczna Analiza
20.   library(caret)          Obliczanie macierzy błędu
21.   library(RWeka)          Drzewo C4.5
22.   library(ipred)          Drzewo Bagging
23.
```

Aby zainstalować każdy z algorytmów należy w języku R wprowadzić polecenie:

```
install.packages("Nazwa pakietu")
```

3. DANE

Baza danych została pobrana ze strony:

https://archive.ics.uci.edu/ml/datasets/Drug+consumption+%28quantified%29

Zawiera ona informacje ankietowanego o:

- Płci
- Wieku
- Wykształceniu
- Kraju
- Pochodzeniu

Jednakowo posiadając informacje o cechach psychologicznych:

- Neurotyczność
- Ekstrawersji
- Otwartości na nowe doświadczenia
- Ugodowości
- Sumienności
- Impulsywności
- Uczuciowości

Oprócz tego baza danych zawiera informacje o różnego rodzaju używkach:

- Alkoholu
- Papierosach
- Marihuanie
- Itp…

W moim projekcie skupiłem się głównie na użyciu alkoholu przez osoby ankietowane. Przeprowadziłem testy:

- Dzieląc użycie alkoholu na:
    - Wypity w przeciągu ostatniego roku
    - Nie pity w przez co najmniej ostatni rok

4. TEORIA

W programie spróbowałem użyć wielu różnych algorytmów i metod klasyfikujących dane aby wybrać najlepsze rozwiązanie do ustalonego przeze mnie wcześniej problemu.

Użyte algorytmy:

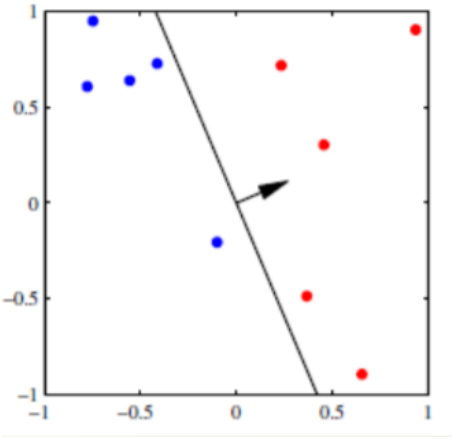
- Klasyfikacja Danych:
  - Klasyfikacja liniowa:
    - Regresja Logistyczna
    - Liniowa Analiza Dyskryminacyjna
  - Klasyfikacja nieliniowa:
    - Analizy Dyskryminacyjne:
      a) Kwadratowa Analiza Dyskryminacyjna
      b) Elastyczna Analiza Dyskryminacyjna
    - Sieć Neuronowa
    - K-najbliższych Sąsiadów
    - Maszyna Wektorów Nośnych
    - Naive Bayes Podstawowy
    - Naive Bayes Bernoulliego
    - Drzewa Decyzyjne:
      a) Zwykłe Drzewa Decyzyjne
      b) Maszyna Ucząca Gradienty
      c) Losowy Las
      d) Drzewa Decyzyjne 4.5
      e) Bagging Trees
- Grupowanie Danych:
  - Metoda K-średnich
  - Metoda K-medoid
  - Metoda DBSCAN
- Reguły Asocjacyjne:
  - Algorytm Apriori

Na czym polega klasyfikacja danych?

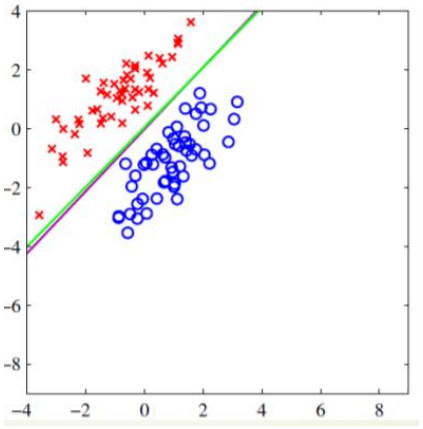
Klasyfikacja danych polega na użyciu i przeanalizowaniu charakterystyk obiektu testowego w celu zidentyfikowaniu do jakiej klasy/kategorii on należy. Na przykład w moim projekcie sprawdzam czy osoba piła lub nie piła alkoholu na podstawie zebranych informacji o niej.

Na czym polega klasyfikacja liniowa?

Klasyfikacja liniowa jako jeden ze sposobów badania danych, klasyfikuje je za pomocą wzoru funkcji liniowej. Aby łatwiej to sobie wyobrazić można zobaczyć poniższe wykresy:



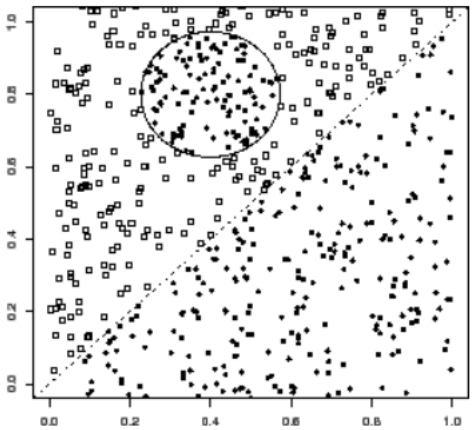
Rysunek 2: Wykres klasyfikacji liniowej 1



Rysunek 1: Wykres klasyfikacji liniowej 2

Na czym polega klasyfikacja nieliniowa?

Klasyfikacja nieliniowa klasyfikuje dane za pomocą funkcji nieliniowych czyli np. kwadratowych lub dopasowujących się do danych, a także za pomocą metod probabilistycznych np. NaiveBayes, maszyn lub drzew decyzyjnych.



Rysunek 3: Wykres klasyfikacji nieliniowej (Knn)

Wyjaśnienie działania algorytmów.

Regresja Logiczna (klasyfikacja liniowa) - opiera się na obliczaniu prawdopodobieństwa. Zamiast określać prawdopodobieństwo klasycznie, za pomocą stosunku liczby sukcesów do liczby wszystkich prób, oblicza się szansę, czyli stosunek prawdopodobieństwa sukcesu do prawdopodobieństwa porażki.

Wzór:

$$\frac{p}{1-p} = e^{\alpha} e^{\beta x},$$

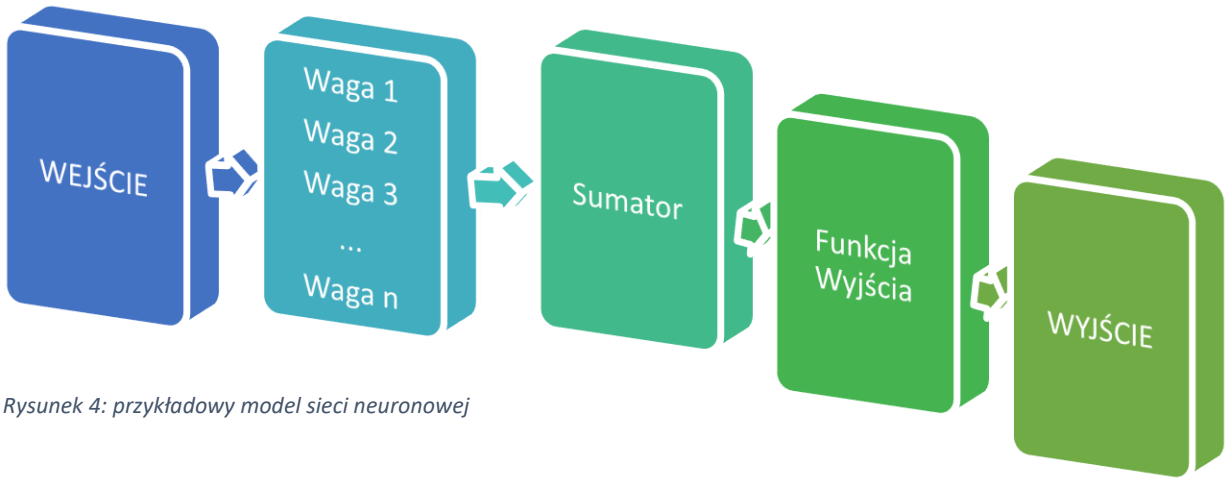
alpha – stała regresji dla regresji logistycznej,
beta – współczynnik regresji logistycznej dla i-tej zmiennej niezależnej,
x – zmienna niezależna (i-ta).

Liniowa Analiza Dyskryminacyjna (klasyfikacja liniowa) - Zakłada, że wektory obserwacji są wektorami w przestrzeni **p** - wymiarowej **X ⊂ R^p** i prowadzi do otrzymania reguły dyskryminacyjnej opartej na funkcji liniowej. Reguła ta dla **g = 2** stara się znaleźć kierunek **a** w **X**, który najlepiej rozdziela obydwie podpróby uczące.

Kwadratowa Analiza Dyskryminacyjna (klasyfikacja nieliniowa) – jest rozszerzeniem Liniowej Analizy Dyskryminacyjnej. Czasem wyników nie da się sklasyfikować za pomocą funkcji liniowej i wtedy z pomocą przychodzi funkcja kwadratowa. Różnica pomiędzy Liniową, a Kwadratową jest taka, że w Kwadratowej nie zakładamy już, że macierz kowariancji jest wspólna dla wszystkich poziomów zmiennej g=2. Założenie to powoduje, że wartości dyskryminacyjne nie są już liniowe tylko kwadratowe.

Elastyczna Analiza Dyskryminacyjna (klasyfikacja nieliniowa) – opiera się na tworzeniu funkcji bardziej dopasowanej do zbioru przedstawionych informacji, a więc nie ograniczamy się już tylko analizą zwykła liniową lub kwadratową.

Sieć Neuronowa – Polega on na przetwarzaniu danych przez neurony pogrupowane w warstwy. Odpowiednie wyniki uzyskuje się dzięki procesowi uczenia, który polega na modyfikowaniu wag tych neuronów, które są odpowiedzialne za błąd.



Rysunek 4: przykładowy model sieci neuronowej

K-najbliższych Sąsiadów – prosty algorytm klasyfikujący geometrycznie wartość na podstawie **K** najbliższych sąsiadów. Przykładowo mamy niesklasyfikowaną zmienną **x** i szukamy jaką klasę jej dobrać na układzie współrzędnych. Ustalamy **K** np. równe 3, w takim wypadku szukamy 3 najbliższych sąsiadów są to sąsiedzi **a**, **a** i **b**. W takim wypadku mamy więcej sąsiadów klasy a, więc klasyfikujemy nasze **x** do klasy **a**.

Maszyna Wektorów Nośnych – koncept maszyny abstrakcyjnej, która wyznacza hiperpłaszczyznę rozdzielającą przykłady dla dwóch klas. Maszyna ta ma za zadanie wyznaczyć najszerszą granicę dyskryminacji spośród wszystkich możliwych. Szerokość tej granicy wyznaczamy za pomocą **iloczynu kartezjańskiego wektora wag** oraz **różnicy wektorów skrajnych** należących do obu klas.

Naive Bayes Podstawowy – opiera się na założeniu wzajemnej niezależności predyktorów zmiennych niezależnych. Jest on nazywany naiwnym ponieważ często te zmienne nie wpływają w ogóle na klasyfikacje w świecie realnym. Klasyfikacja odbywa się licząc naiwny model probabilistyczny Bayesa. Jest on dosyć długi do wypisania ale prosty do obliczeń dlatego można go zobaczyć np. tutaj:

https://www.machinelearningplus.com/predictive-modeling/how-naive-bayes-algorithm-works-with-example-and-full-code/#5naivebayesexamplebyhand

Naive Bayes Bernoulliego – jest to algorytm Bayesowy z tą różnicą, że korzysta z modelu dystrybucji cech stworzonego przez Bernoulliego. Szczególnie przydatny przy klasyfikowani dokumentów tekstowych.

Bagging Trees (drzewa decyzyjne) - jest to model stworzony z wielu różnych drzew decyzyjnych. Algorytm ten wybiera w każdym podziale różny zakres danych do wytrenowania drzew decyzyjnych. Kończymy w ten sposób z wieloma modelami drzew i wybieramy średni model drzewa decyzyjnego.

Maszyna Ucząca Gradienty (drzewa decyzyjne) – jest to maszyna mająca za zadanie uczenie drzewa decyzyjnego w taki sposób by było ono jak najbardziej dokładne. Po stworzeniu wag prototypowego drzewa algorytm zwiększa wagi tych obserwacji, które są trudne do sklasyfikowania gdziekolwiek, oraz zmniejsza wagi tych łatwych do klasyfikacji. Algorytm powtarza te operacje tyle razy ile mu rozkażemy i na koniec otrzymamy drzewo będące sumą warzoną przewidywań poprzednich modeli drzew.

Las Losowy nazywany także lasem losowych decyzji (drzewa decyzyjne) – jest to model stworzony z wielu różnych drzew decyzyjnych. Algorytm ten wybiera w każdym podziale kandydata w procesie uczenia się losowego podzbioru cech. Powodem tego jest korelacja drzew w zwykłej próbce: jeśli jedna lub kilka cech jest bardzo silnymi predyktorami dla zmiennej odpowiedzi, funkcje te zostaną wybrane w wielu drzewach, powodując ich skorelowanie. Prostszymi słowami: każde drzewo podejmuje decyzje na podstawie tylko tych cech, które zna, gdy na końcu przeanalizujemy wszystkie drzewa możemy otrzymać wynik lepszy niż tylko z jednego drzewa decyzyjnego, ponieważ nie będzie on bazował na niektórych atrybutach, które mogą nie mieć nic wspólnego z klasyfikacją.

PAR Trees (drzewa decyzyjne) - Jest to algorytm tworzenia drzewa decyzyjnego jak sama nazwa wskazuje. Drzewo decyzyjne podejmuje decyzje patrząc na atrybuty i porównując ich wartość w ten sposób przydziela odpowiednią ścieżkę, która prowadzi do klasyfikacji czyli rozwiązania.

Drzewa Decyzyjne 4.5 (drzewa decyzyjne) – rozwinięcie pomysłu drzewa decyzyjnego. Drzewo decyzyjne 4.5 ma kilka przewag nad drzewem ID3:

- Budując drzewo decyzyjne możemy rozwiązać przypadek, gdy rekordy mają nieznaną wartość dla pewnych atrybutów.
  Gain jest wtedy wyliczane na podstawie tylko tych rekordów, gdzie dana wartość jest zdefiniowana.
- Posiada mechanizm przeciwdziałający zjawisku *overfittingowi* (doprowadzającego do wysokiego poziomu błędu dla rzeczywistych danych)

Drzewa Decyzyjne CIT (drzewa decyzyjne) - Drzewa wnioskowania warunkowego szacują relację regresji przez binarne partycjonowanie rekurencyjne w ramach wnioskowania warunkowego. Z grubsza algorytm działa w następujący sposób:

1) Przetestuj globalną zerową hipotezę niezależności między dowolną zmienną wejściową, a odpowiedzią (która może być również wielowymiarowa). Przestań, jeśli tej hipotezy nie można odrzucić. W przeciwnym razie wybierz zmienną wejściową o największym powiązaniu z odpowiedzią. To powiązanie jest mierzone wartością **p** odpowiadającą testowi częściowej hipotezy zerowej pojedynczej zmiennej wejściowej i odpowiedzi.

2) Zaimplementuj podział binarny w wybranej zmiennej wejściowej.

3) Rekurencyjnie powtarzaj kroki 1) i 2).

5. OPIS EKSPERYMENTÓW

Aby rozpocząć pracę należy najpierw przygotować sobie zbiór danych.

```
library(readr)
file <- "C:/ProgramyZainstalowane/R_Workspace/drug_consumption.data"
drugs <- read_csv(file, col_names = FALSE)

names <- c("id", "age", "gender", "education", "country", "ethnicity",
           "nscore", "escore", "oscore", "ascore", "cscore", "impulsive",
           "ss", "alcohol", "amphet", "amyl", "benzos", "caff", "cannabis",
           "choc", "coke", "crack", "ecstasy", "heroin", "ketamine", "legalh",
           "lsd", "meth", "mushrooms", "nicotine", "semer", "vsa")
drugs <- setNames(drugs, names)
```

Na razie dodaliśmy wszystkie potrzebne nazwy kolumn do surowej bazy danych.

```
for (i in 1:nrow(drugs)) {
  if (drugs["age"][i,] == -0.95197) drugs["age"][i,] <- "18-24"
  else if (drugs["age"][i,] == -0.07854) drugs["age"][i,] <- "25-34"
  else if (drugs["age"][i,] == 0.49788) drugs["age"][i,] <- "35-44"
  else if (drugs["age"][i,] == 1.09449) drugs["age"][i,] <- "45-54"
  else if (drugs["age"][i,] == 1.82213) drugs["age"][i,] <- "55-64"
  else if (drugs["age"][i,] == 2.59171) drugs["age"][i,] <- "65+"

  if (drugs["gender"][i,] == 0.48246) drugs["gender"][i,] <- "Female"
  else if (drugs["gender"][i,] == -0.48246) drugs["gender"][i,] <- "Male"

  if (drugs["education"][i,] == -
2.43591) drugs["education"][i,] <- "Left school before 16 years"
  else if (drugs["education"][i,] == -
1.73790) drugs["education"][i,] <- "Left school at 16 years"
  else if (drugs["education"][i,] == -
1.43719) drugs["education"][i,] <- "Left school at 17 years"
  else if (drugs["education"][i,] == -
1.22751) drugs["education"][i,] <- "Left school at 18 years"
  else if (drugs["education"][i,] == -
0.61113) drugs["education"][i,] <- "Some college or university, no certificate or de
gree"
  else if (drugs["education"][i,] == -
0.05921) drugs["education"][i,] <- "Professional certificate/ diploma"
  else if (drugs["education"][i,] == 0.45468) drugs["education"][i,] <- "University
degree"
  else if (drugs["education"][i,] == 1.16365) drugs["education"][i,] <- "Masters deg
ree"
  else if (drugs["education"][i,] == 1.98437) drugs["education"][i,] <- "Doctorate d
egree"

  if (drugs["country"][i,] == -0.09765) drugs["country"][i,] <- "Australia"
  else if (drugs["country"][i,] == 0.24923) drugs["country"][i,] <- "Canada"
  else if (drugs["country"][i,] == -0.46841) drugs["country"][i,] <- "New Zealand"
  else if (drugs["country"][i,] == -0.28519) drugs["country"][i,] <- "Other"
  else if (drugs["country"][i,] == 0.21128) drugs["country"][i,] <- "Republic of Ire
land"
```

```
  else if (drugs["country"][i,] == 0.96082) drugs["country"][i,] <- "UK"
  else if (drugs["country"][i,] == -0.57009) drugs["country"][i,] <- "USA"

  if (drugs["ethnicity"][i,] == -0.50212) drugs["ethnicity"][i,] <- "Asian"
  else if (drugs["ethnicity"][i,] == -1.10702) drugs["ethnicity"][i,] <- "Black"
  else if (drugs["ethnicity"][i,] == 1.90725) drugs["ethnicity"][i,] <- "Mixed-
Black/Asian"
  else if (drugs["ethnicity"][i,] == 0.12600) drugs["ethnicity"][i,] <- "Mixed-
White/Asian"
  else if (drugs["ethnicity"][i,] == -0.22166) drugs["ethnicity"][i,] <- "Mixed-
White/Black"
  else if (drugs["ethnicity"][i,] == 0.11440) drugs["ethnicity"][i,] <- "Other"
  else if (drugs["ethnicity"][i,] == -0.31685) drugs["ethnicity"][i,] <- "White"

  if (drugs["nicotine"][i,] == "CL0") drugs["nicotine"][i,] <- "Never Used"
  else if (drugs["nicotine"][i,] == "CL1") drugs["nicotine"][i,] <- "Used over a Dec
ade Ago"
  else if (drugs["nicotine"][i,] == "CL2") drugs["nicotine"][i,] <- "Used in Last De
cade"
  else if (drugs["nicotine"][i,] == "CL3") drugs["nicotine"][i,] <- "Used in Last Ye
ar"
  else if (drugs["nicotine"][i,] == "CL4") drugs["nicotine"][i,] <- "Used in Last Mo
nth"
  else if (drugs["nicotine"][i,] == "CL5") drugs["nicotine"][i,] <- "Used in Last We
ek"
  else if (drugs["nicotine"][i,] == "CL6") drugs["nicotine"][i,] <- "Used in Last Da
y"
}
```

Teraz zamieniliśmy wartości liczbowe na takie, które będą łatwiejsze do zrozumienia dla człowieka. Wartości zostały zmienione według opisu na stronie autora bazy danych.

```
drugs.alcohol <- drugs2[c(2:13,14,30)]

for (i in 1:nrow(drugs.alcohol)) {
  nonuserSet <- c("CL0", "CL1","CL2", "CL3")
  userSet <- c("CL4", "CL5", "CL6")
  if (drugs.alcohol["alcohol"][i,] %in% nonuserSet) {
    drugs.alcohol["alcohol"][i,] <- "Did not drunk for at least 1 year"
  } else if (drugs.alcohol["alcohol"][i,] %in% userSet) {
    drugs.alcohol["alcohol"][i,] <- "Drunk alcohol for at least 1 year"
  }
}
```

Bierzemy do naszej bazy roboczej tylko te wartości które nas interesują i zamieniamy pole alkoholu w taki sposób aby można było je klasyfikować dwoma klasami.

```
norm <- function(x) {
  (x-min(x))/(max(x)-min(x))
}

drugs.norm <- data.frame(norm(drugs.alcohol[1]), norm(drugs.alcohol[2]),
                         norm(drugs.alcohol[3]), norm(drugs.alcohol[4]),
                         norm(drugs.alcohol[5]), norm(drugs.alcohol[6]),
                         norm(drugs.alcohol[7]), norm(drugs.alcohol[8]),
                         norm(drugs.alcohol[9]), norm(drugs.alcohol[10]),
                         norm(drugs.alcohol[11]), norm(drugs.alcohol[12]),
                         drugs.alcohol[13])

set.seed(1234)
ind <- sample(2, nrow(drugs.norm), replace=TRUE, prob=c(0.67, 0.33))
drugs.train <- drugs.norm[ind==1,1:13]
drugs.test <- drugs.norm[ind==2,1:13]

drugsOld.train <- drugs.norm[ind==1,1:13]
drugsOld.test <- drugs.norm[ind==2,1:13]
```

Normalizujemy naszą bazę danych i dzielimy ją na dwie pary zbiorów treningowych i testowych.

Teraz można przejść do implementacji algorytmów liniowych

Warto dodać że każdy algorytm jest u mnie w ten sposób wywoływany:

1. przypisanie wykonanego modelu danego algorytmu
2. przewidzenie tego modelu względem bazy testowej
3. wypisanie macierzy błędu
4. obliczenie dokładności w procentach na podstawie macierzy błędu

```
#Logistic Regression
drugs.vgam<-vglm(factor(alcohol)~age+gender+education+country+
                   ethnicity+nscore+escore+oscore+ascore+cscore+impulsive+ss,
                 family = "multinomial",drugsOld.train)

summary(drugs.vgam)
drugs.vgam.predicted<-predict(drugs.vgam,drugsOld.test[,1:12],type="response")
drugs.vgam.tab1<-table(apply(drugs.vgam.predicted,1,which.max),drugs.real)
drugs.vgam.tab<-drugs.mda.tab
drugs.vgam.tab[1]=drugs.vgam.tab1[1]
drugs.vgam.tab[2]=drugs.vgam.tab1[2]
drugs.vgam.tab[2][1]=drugs.vgam.tab1[2][1]
drugs.vgam.conf<-confusionMatrix(drugs.vgam.tab)
cf$overall[1][1]*100
drugs.vgam.accuracy=drugs.vgam.conf$overall[1][1]*100

#Linear Discriminant Analysis
drugs.lda<-MASS::lda(factor(alcohol)~age+gender+education+country+
              ethnicity+nscore+escore+oscore+ascore+cscore+impulsive+ss,drugsOld.tra
in)
summary(drugs.lda)
drugs.lda.predicted<-predict(drugs.lda,drugsOld.test[,1:12])
drugs.lda.predicted[1]
drugs.lda.tab1<-table(as.matrix(as.data.frame(drugs.lda.predicted[1])),drugs.real)
drugs.lda.conf<-confusionMatrix(drugs.lda.tab1)
drugs.lda.accuracy=drugs.lda.conf$overall[1][1]*100
drugs.lda.accuracy
```

Teraz możemy zaimplementować analizy dyskryminacyjne nieliniowe:

```
#Quadratic Discriminant Analysis
drugs.mass<-MASS::qda(factor(alcohol)~age+gender+education+country+
                    ethnicity+nscore+escore+oscore+ascore+cscore+impulsive+ss,drugsOld
.train)

summary(drugs.mass)
drugs.mass.predicted<-predict(drugs.mass,drugsOld.test[,1:12])
drugs.mass.predicted[1]
drugs.mass.tab1<-
table(as.matrix(as.data.frame(drugs.mass.predicted[1])),drugs.real)
drugs.mass.conf<-confusionMatrix(drugs.mass.tab1)
drugs.mass.accuracy=drugs.mass.conf$overall[1][1]*100

#------Flexible Discriminant Analysis-----
drugs.mda<-fda(factor(alcohol)~age+gender+education+country+
                  ethnicity+nscore+escore+oscore+ascore+cscore+impulsive+ss,data=drug
sOld.train)
summary(drugs.mda)
drugs.mda.predicted<-predict(drugs.mda,drugsOld.test[,1:12],type="class")
drugs.mda.tab<-table(drugs.mda.predicted,drugs.real)
cf<-confusionMatrix(drugs.mda.tab)
cf$overall[1][1]*100
drugs.mda.accuracy=cf$overall[1][1]*100
```

Tak samo możemy policzyć algorytm 3 najbliższych sąsiadów, Naive Bayes, Bernoulli Naive Bayes oraz Maszynę Wektorów Nośnych:

```
# -----KNN-----------------------------------------------------
knn.3 <- knn(drugs.train[,1:12], drugs.test[,1:12], cl=drugs.train[,13], k=3, prob=F
ALSE)
knn.predicted <- knn.3
knn.real <- drugs.test[,13]
knn.conf.matrix <- table(knn.predicted, knn.real)
knn.accuracy <- sum(diag(knn.conf.matrix))/sum(knn.conf.matrix)
# -----NaiveBayes Porbability----------------------------------------
#Method from June 3/2019 Year
nbayesNew <- bernoulli_naive_bayes(as.matrix(drugs.train[,1:12]), drugs.train[,13],
laplace = 0.5)
nbayesNew.predicted <- predict(nbayesNew, as.matrix(drugs.test[,1:12]))
nbayesNew.real <- drugs.test[,13]
nbayesNew.conf.matrix <- table(nbayesNew.predicted, nbayesNew.real)
nbayesNew.accuracy <- sum(diag(nbayesNew.conf.matrix))/sum(nbayesNew.conf.matrix)
#Older method
nbayes <- naiveBayes(drugs.train[,1:12], drugs.train[,13])
nbayes$levels <- c("Did not drunk for at least 1 year", "Drunk alcohol for at least
1 year")
nbayes.predicted <- predict(nbayes, drugs.test[,1:12])
nbayes.real <- drugs.test[,13]
nbayes.conf.matrix <- table(nbayes.predicted, nbayes.real)
nbayes.accuracy <- sum(diag(nbayes.conf.matrix))/sum(nbayes.conf.matrix)
# -----SVM-----------------------------------------------
library("e1071")
data <- drugs.train
model_svm <- svm(alcohol ~ ., data=data)
summary(model_svm)
pred <- predict(model_svm,dataValid)
table(pred, data$alcohol)
accuracySVM = mean(pred == dataValid$alcohol)
```

Zostały nam do zrobienia drzewa decyzyjne:

```
# -----Decision Tree-----------------------------------
#New algorythm 18 July 2019
drugsNew.ctree <- partykit::ctree(factor(alcohol) ~ age + gender + education + country + ethnicity + nscore + escore + oscore + ascore + cscore + impulsive + ss,
                              data=drugsOld.train)
print(drugs.ctree)
plot(drugs.ctree)
plot(drugs.ctree, type="simple")

treeNew.predicted <- predict(drugsNew.ctree, drugsOld.test[,1:12])
treeNew.real <- drugsOld.test[,13]
treeNew.conf.matrix <- table(treeNew.predicted, treeNew.real)
treeNew.accuracy <- sum(diag(treeNew.conf.matrix))/sum(treeNew.conf.matrix)
treeNew.accuracy
# -----Bagging CART-----------------------------------
drugs.bagg<-
bagging(factor(alcohol) ~ age + gender + education + country + ethnicity
                    + nscore + escore + oscore + ascore + cscore + impulsive + ss,
                    data=drugsOld.train)
summary(drugs.bagg)
drugs.bagg.predicted <- predict(drugs.bagg, drugsOld.test[,1:12])
drugs.bagg.real <- drugsOld.test[,13]
drugs.bagg.conf.matrix <- table(drugs.bagg.predicted, drugs.bagg.real)
drugs.bagg.accuracy <- sum(diag(drugs.bagg.conf.matrix))/sum(drugs.bagg.conf.matrix) *100
drugs.bagg.accuracy
# -----C4.5-----------------------------------
drugs.c45<-J48(factor(alcohol) ~ age + gender + education + country + ethnicity
               + nscore + escore + oscore + ascore + cscore + impulsive + ss,
               data=drugsOld.train)
summary(drugs.c45)
drugs.c45.predicted <- predict(drugs.c45, drugsOld.test[,1:12])
drugs.c45.real <- drugsOld.test[,13]
drugs.c45.conf.matrix <- table(drugs.c45.predicted, drugs.c45.real)
drugs.c45.accuracy <- sum(diag(drugs.c45.conf.matrix))/sum(drugs.c45.conf.matrix) *100
drugs.c45.accuracy
#------GBM-------------------------------------------------
library(gbm)
drugs.gbm<-gbm(factor(alcohol)~age+gender+education+country+
             ethnicity+nscore+escore+oscore+ascore+cscore+impulsive+ss,data=drugsOld.train,
             distribution="multinomial")
summary(drugs.gbm)
drugs.gbm.predicted<-predict(drugs.gbm,drugsOld.test[,1:12],n.trees=1)
drugs.gbm.eval<-
colnames(drugs.gbm.predicted)[apply(drugs.gbm.predicted,1,which.max)]
drugs.real <- drugsOld.test[,13]
drugs.gbm.tab<-table(drugs.gbm.eval,drugs.real)
drugs.gbm.tab
confusionMatrix(drugs.gbm.tab)
drugs.gbm.eval.acc=491/590 *100
```

I na koniec sieć neuronowa:

```
# -----Neural Net-----------------------------------------------
drugs.train$tested_positive <- 0
drugs.train$tested_negative <- 0
for (row in 1:nrow(drugs.train)) {
  if (drugs.train[row,]["alcohol"] == "Drunk alcohol for at least 1 year") drugs.train[row,]["tested_positive"] = 1
  if (drugs.train[row,]["alcohol"] == "Did not drunk for at least 1 year") drugs.train[row,]["tested_negative"] = 1
}

drugs.train <- subset(drugs.train, select = -c(alcohol))
drugs.neuralnet <- neuralnet(tested_positive + tested_negative ~ age + gender + education + country + ethnicity + nscore + escore + oscore + ascore + cscore + impulsive + ss,
                             drugs.train, hidden=4)
drugs.pred <- neuralnet::compute(drugs.neuralnet, drugs.test[,1:12])
plot(drugs.neuralnet)
drugs.pred_species <- c()
for (row in 1:nrow(drugs.pred$net.result)) {
  column <- match(max(drugs.pred$net.result[row,]), drugs.pred$net.result[row,])
  if (column == 1) drugs.pred_species <- c(drugs.pred_species, "tested_positive")
  if (column == 2) drugs.pred_species <- c(drugs.pred_species, "tested_negative")
}

drugs.comparison <- cbind("real" = as.character(drugs.test["alcohol"][,1]), "predicted" = drugs.pred_species)
drugs.result <- c()
for (row in 1:nrow(drugs.comparison)) {
  if (drugs.comparison[,1][row] == "Drunk alcohol for at least 1 year" && drugs.comparison[,2][row] == "tested_positive"
    || drugs.comparison[,1][row] == "Did not drunk for at least 1 year" && drugs.comparison[,2][row] == "tested_negative" ) drugs.result <- c(drugs.result, TRUE)
  else drugs.result <- c(drugs.result, FALSE)
}
sum(drugs.result, na.rm = TRUE)
NeuronNetAccuracy <- as.numeric(table(drugs.result)["TRUE"])/as.numeric(table(drugs.result)["TRUE"]+table(drugs.result)["FALSE"])*100
NeuronNetAccuracy
```

Naturalnie przy samej sieci neuronowej jest trochę więcej roboty. Trzeba do skopiowanego znormalizowanego zbioru dodać pola binarne mówiące nam czy dana osoba piła alkohol czy też nie i usunąć pole klasy alkohol w którym zawieraliśmy te dane w sposób tekstowy.

Po wygenerowaniu sieci neuronowej, wykorzystujemy ją do wygenerowania przewidywanych wartości dla zbioru testowego.

Teraz by porównać zbiór testowy realny z przewidzianym przez sieć musimy sprawdzić czy wynik przewidziany jest taki sam jak realny i stworzyć w ten sposób tabelkę wartości boolowskich **TRUE** i **FALSE**. Na koniec tak jak w każdym algorytmie, tworzymy macierz błędu i obliczamy dokładność procentową.

Grupowanie danych 3 sposobami: k-średnich, DBSCANem i K-meoid:

```
# -----PRZYGOTOWANIE DANYCH
drugs.log <- log(drugs.norm[,1:12])
drugs.log <- drugs.log[is.finite(rowSums(drugs.log)),]
drugs.log$gender <- NULL
drugs.scale <- scale(drugs.log, center=TRUE)
drugs.pca <- prcomp(drugs.scale)
drugs.final <- predict(drugs.pca)
drugs.final <- drugs.final[,1:11]

# ----- GRUPOWANIE METOD K-ŚREDNICH
drugs.kmeans <- kmeans(drugs.final, 2)
table(drugs.kmeans[["cluster"]])
dev.off()
plot(drugs.final, col = drugs.kmeans[["cluster"]],
     main="Algorytm grupowania metod¹ k-œrednich")
points(drugs.kmeans[["centers"]], col = 1:2, pch = 16, cex=1.5)

#----------------DBSCAN
library("factoextra")
set.seed(123)
dbscan::kNNdistplot(drugs.final, k =  2)
res.fpc <- fpc::dbscan(drugs.final, eps = 3.3, MinPts = 2)
plot(res.fpc, drugs.final, main = "DBSCAN", frame = FALSE)
fviz_cluster(drugs.kmeans, drugs.final, stand = FALSE, frame = FALSE, geom = "point"
)

#------------K-medoids
library(cluster)
library(factoextra)
pam.res <- pam(drugs.final, 2)
print(pam.res)
fviz_cluster(pam.res, drugs.final, stand = FALSE, frame = FALSE, geom = "point")
```

No na sam koniec programu warto wyodrębnić reguły asocjacyjne:

```
#-----Associastion rulsets-------------------------------------------------
raDrugs <- drugs[,2:6]
raDrugs$alcohol <- drugs.alcohol$alcohol

raDrugs$age <- factor(raDrugs$age)
raDrugs$gender <- factor(raDrugs$gender)
raDrugs$education <- factor(raDrugs$education)
raDrugs$country <- factor(raDrugs$country)
raDrugs$ethnicity <- factor(raDrugs$ethnicity)
raDrugs$alcohol <- factor(raDrugs$alcohol)

rules <- apriori(raDrugs)
inspect(rules)
# -----interesting rulesets-----------------------------------
rules2 <- apriori(raDrugs,
                  appearance = list(rhs=c("alcohol=Drunk alcohol for at least 1 year
"), default="lhs"), control = list(verbose=F))
rules2 <- sort(rules2, by="lift")
inspect(rules2)
inspect(rules2[2])
```

W ten sposób wypisaliśmy sobie wszystkie reguły i na koniec wyodrębniliśmy te najciekawsze w naszych badaniach czyli dotyczące alkoholu.

6. WYNIKI

Eksperymenty nie byłyby dopełnione bez odpowiednich wyników i diagramów.

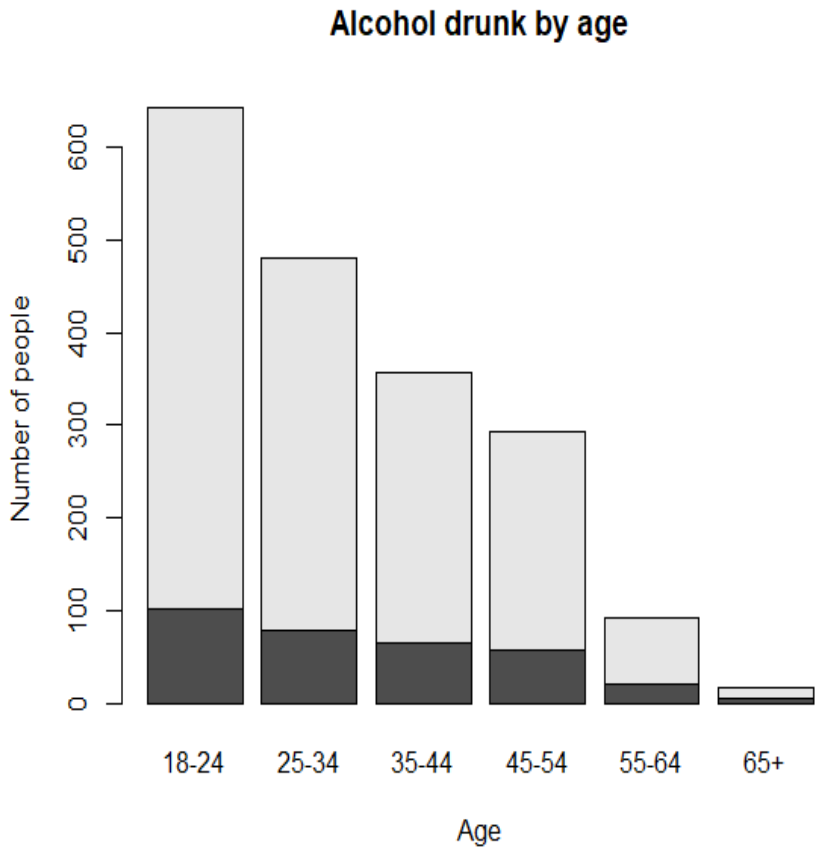
Zacznijmy od statystyk tekstowych:

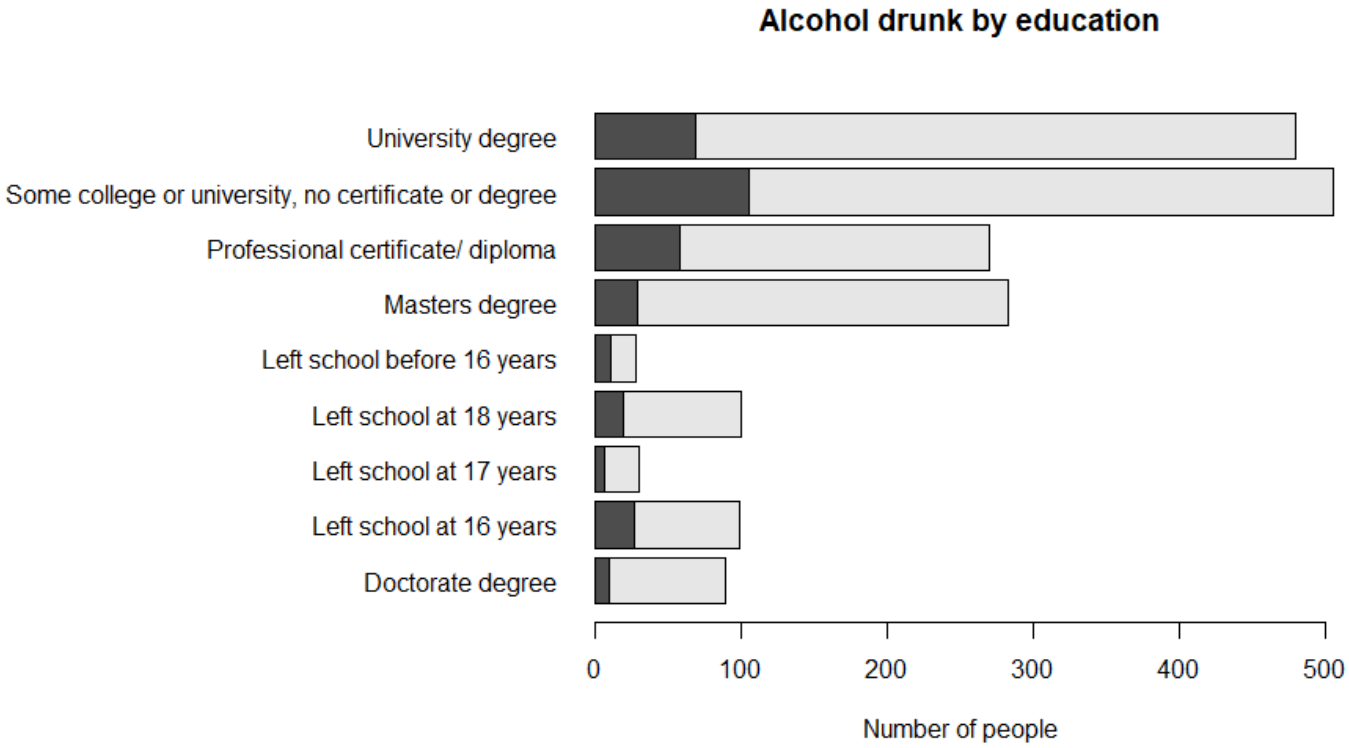
```
statistics <- list("ages" = table(drugs["age"]),
                   "genders" = table(drugs["gender"]),
                   "education" = table(drugs["education"]),
                   "country" = table(drugs["country"]),
                   "ethnicity" = table(drugs["ethnicity"]),
                   "alcohol" = table(drugs.alcohol["alcohol"]))

statistics$ages
statistics$genders
statistics$education
statistics$country
statistics$ethnicity
statistics$alcohol
```

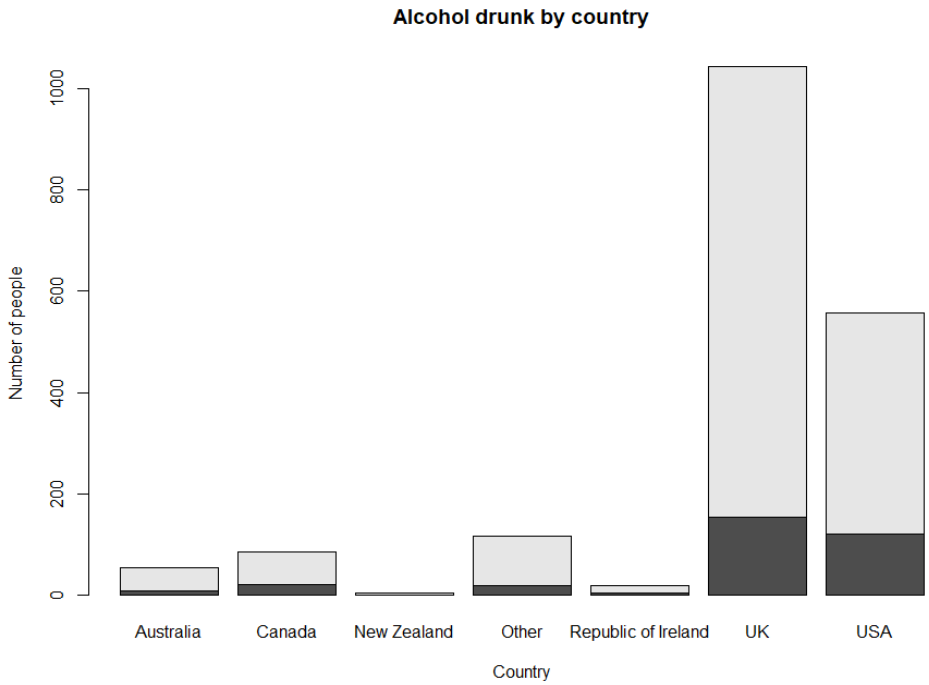
```
> statistics$ages

18-24 25-34 35-44 45-54 55-64   65+
  643   481   356   294    93    18
> statistics$genders

Female   Male
   942    943
> statistics$education

                                  Doctorate degree                              Left school at 16 years
                                                89                                                   99
                         Left school at 17 years                              Left school at 18 years
                                                30                                                  100
                       Left school before 16 years                                      Masters degree
                                                28                                                  283
                  Professional certificate/ diploma Some college or university, no certificate or degree
                                               270                                                  506
                                 University degree
                                               480
> statistics$country

         Australia             Canada        New Zealand              Other Republic of Ireland                  UK
                54                 87                  5                     118                  20            1044
               USA
               557
> statistics$ethnicity

           Asian             Black Mixed-Black/Asian Mixed-White/Asian Mixed-White/Black             Other
              26                33                 3                20                20                63
           White
            1720
> statistics$alcohol

Did not drunk for at least 1 year Drunk alcohol for at least 1 year
                              334                               1551


>
```

Wynika z nich, że większość ankietowanych to młodzi ludzie do 24 roku życia, przeważnie studenci. Zarówno kobiet i mężczyzn jest prawie tyle samo. Głównie badania były przeprowadzone w Wielkiej Brytanii wśród ludności białej i statystycznie tylko 17% osób nie piło w ciągu ostatniego roku alkoholu.

Wynik pokazuje że obie płcie piją bardzo podobnie różnica w danych wynosi jedynie, że 1% kobiet więcej wypiło alkohol w ciągu ostatniego roku od mężczyzn.

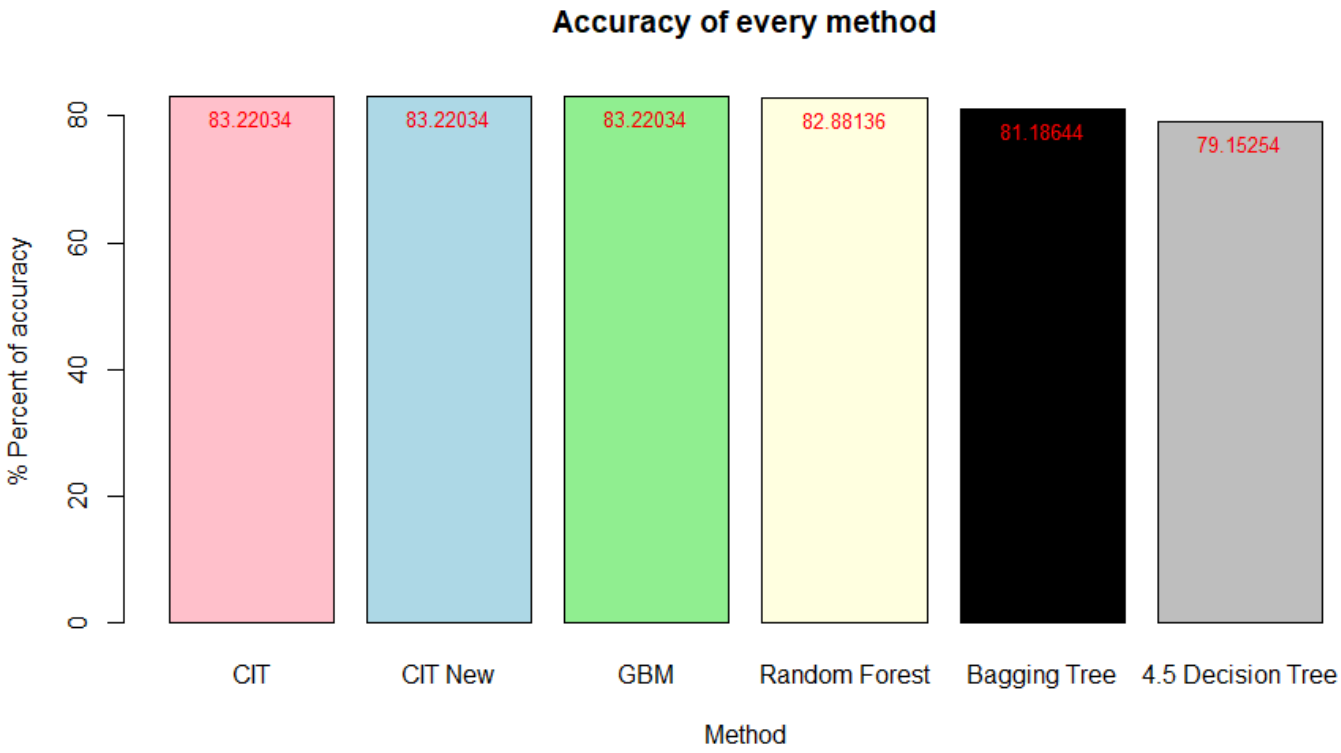



Alcohol drunk by education
University degree
Some college or university, no certificate or degree
Professional certificate/ diploma
Masters degree
Left school before 16 years
Left school at 18 years
Left school at 17 years
Left school at 16 years
Doctorate degree
0
100
200
300
400
500
Number of people

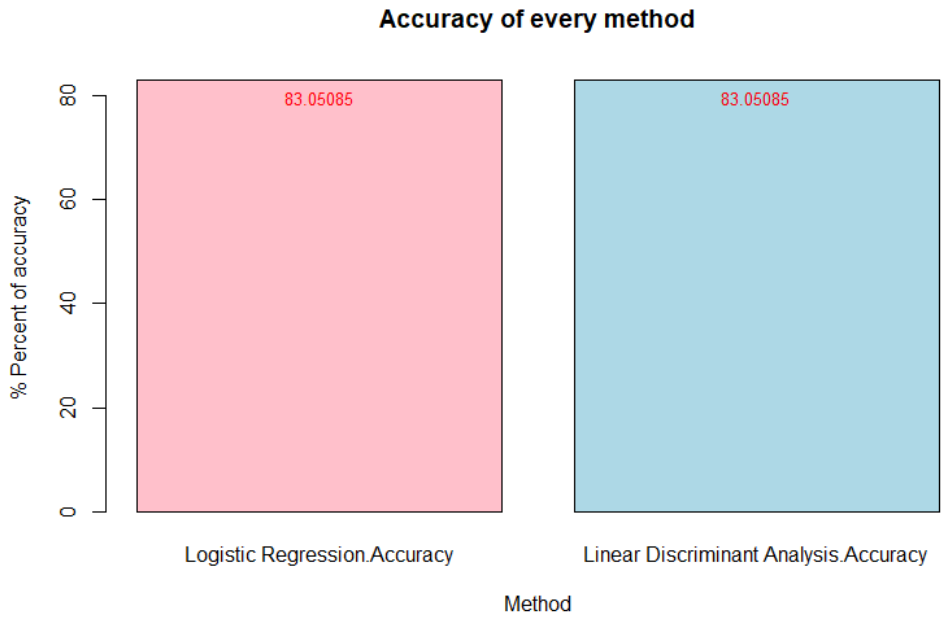

Alcohol drunk by country
Number of people
0
200
400
600
800
1000
Australia
Canada
New Zealand
Other
Republic of Ireland
UK
USA
Country

Pora teraz na przeanalizowanie algorytmów:

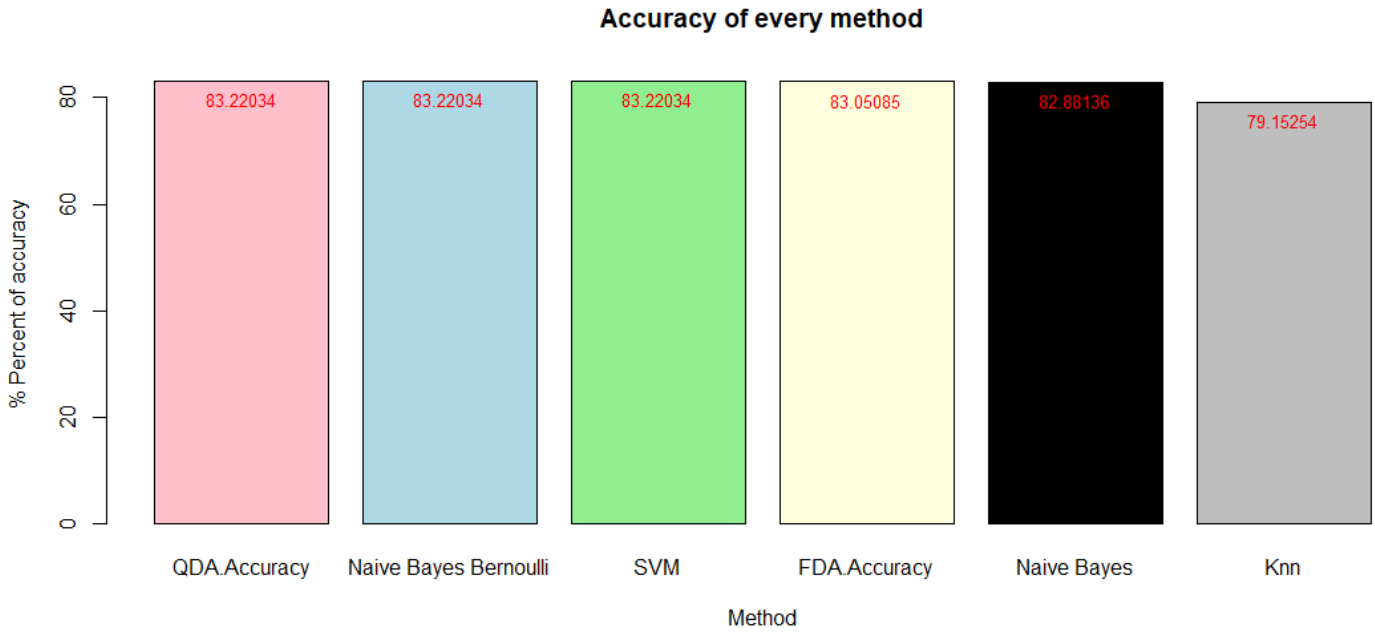
- Drzewa decyzyjne:



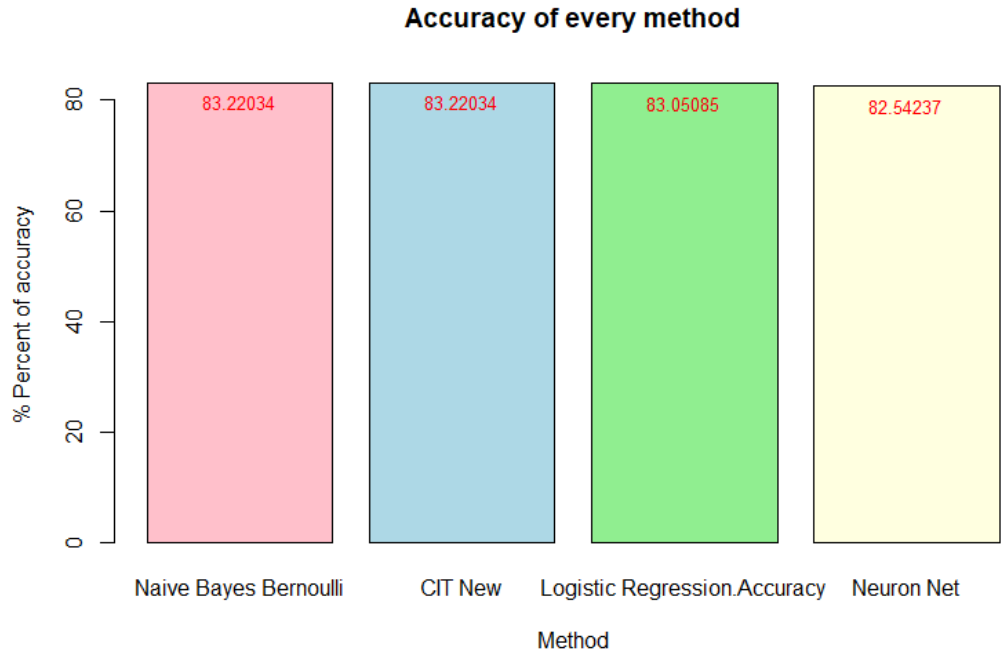
- Klasyfikacja liniowa:



- Klasyfikacja nieliniowa:

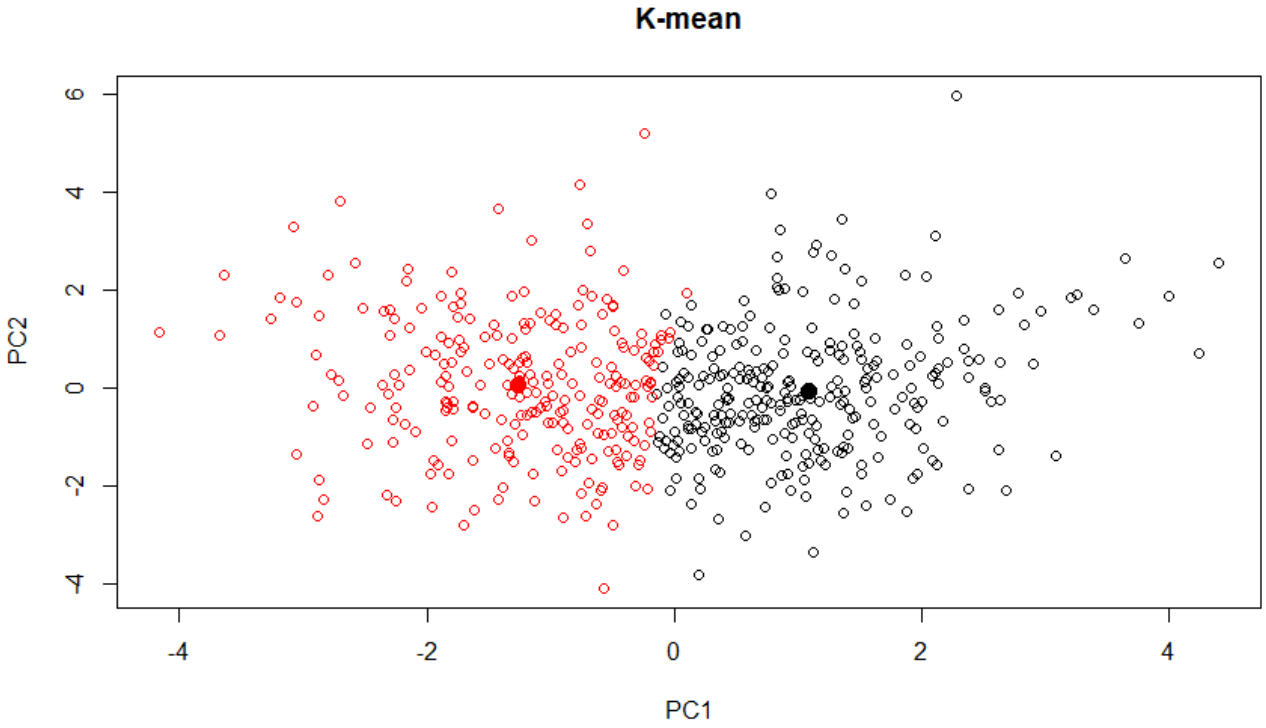


- Najbardziej dokładne sposoby:

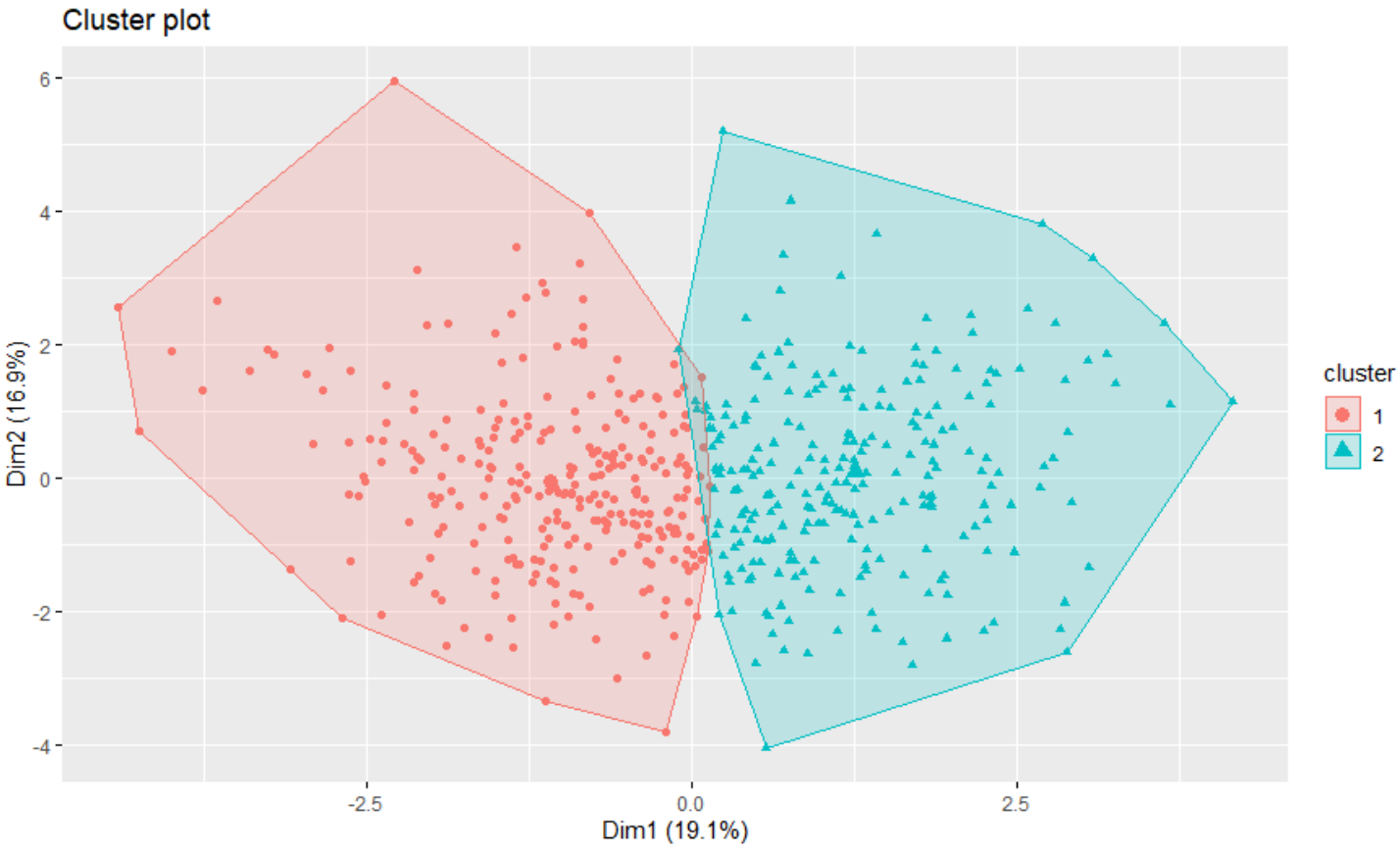


Grupowanie:

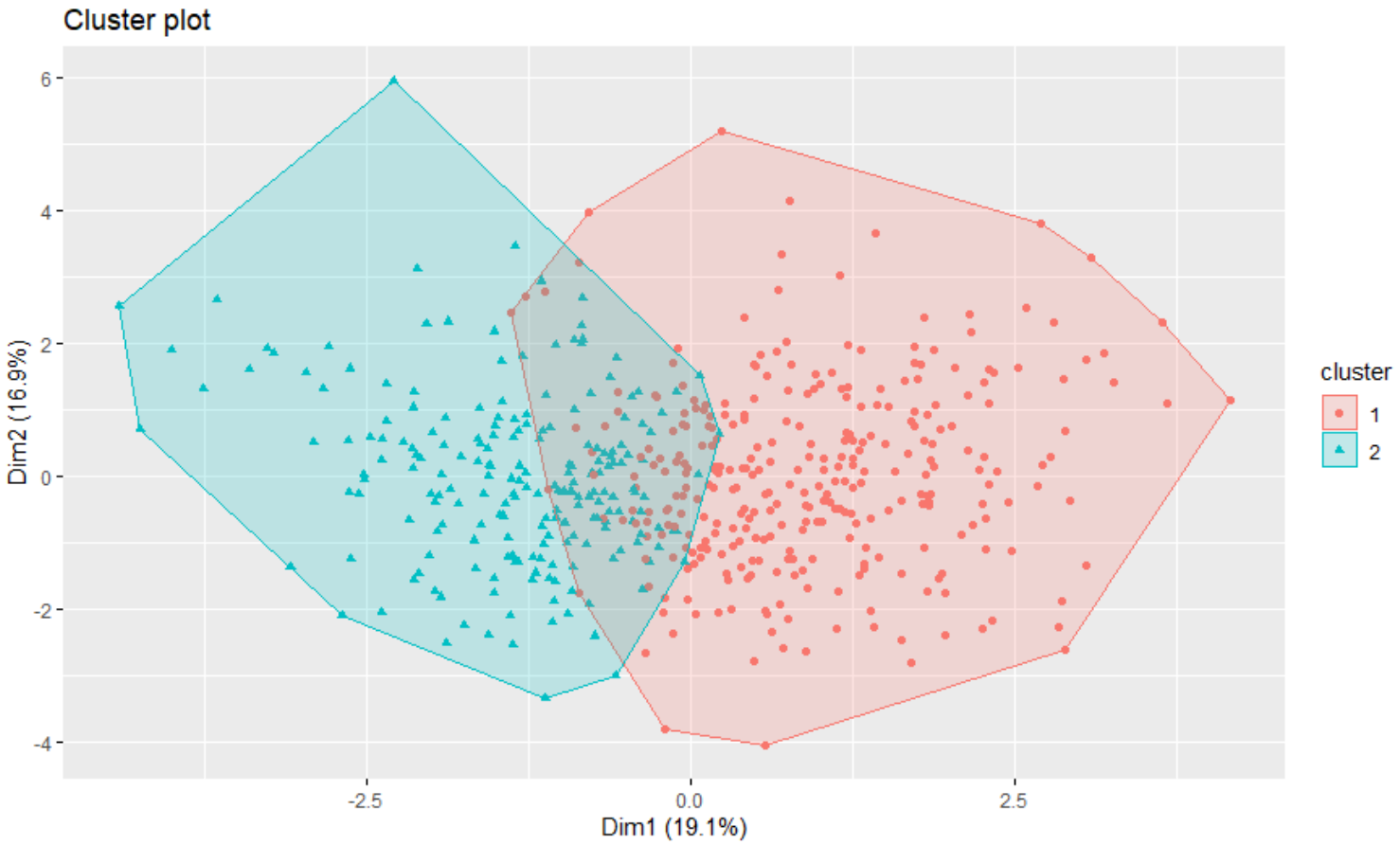
- K-średnich:



- DBSCAN:

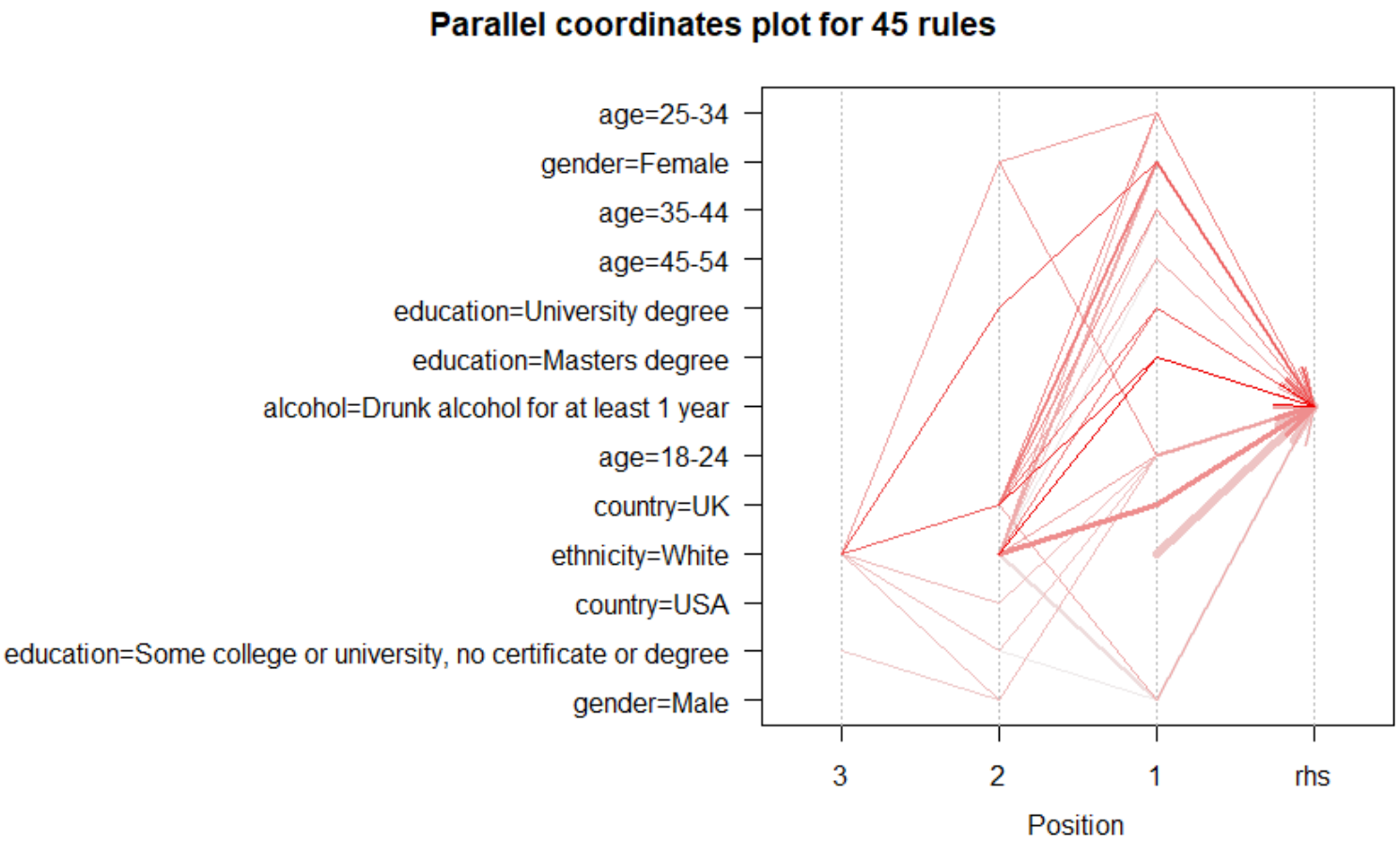


- KK-meoid:



-

Ciekawe zestawy zasad:



10 najważniejszych reguł:

```
[1]  {education=Masters degree, ethnicity=White} => { drunk }
Confidence = 0.1209549 | Support = 0.9156627 | lift = 1.1128460

[2]  {education=Masters degree, country=UK} => { drunk }
Confidence = 0.1039788 | Support = 0.9032258 | lift = 1.0977309

[3]  {education=Masters degree} => { drunk }
Confidence = 0.1347480 | Support = 0.8975265 | lift = 1.0908043

[4]  {gender=Female, education=University degree, ethnicity=White} => { drunk }
Confidence = 0.1246684 | Support = 0.8867925 | lift = 1.0777587

[5]  {education=University degree,country=UK, ethnicity=White} => { drunk }
Confidence = 0.1363395 | Support = 0.8801370 | lift = 1.0696700
```

7. INTERPRETACJA WYNIKÓW

Wyniki przeprowadzone eksperymentu są bardzo ciekawe, ponieważ niektóre metody nie były tak dokładne jak na początku programowania przypuszczałem.

Wśród drzew nie spodziewałem się tak wysokiego wyniku dla drzew **CIT**, a na pewno nie wyższego od algorytmu **RandomForest**, którego precyzja była o około 0.4% mniejsza. Oczywiście możemy założyć, że jest to niska różnica, która może być spowodowana struktura drzewa CIT, które za każdym razem testuje pewne informacja sprawdzając ich przydatność w przeciwieństwie do algorytmu Losowego Lasu.

Ciekawą sprawą jest także to że **Analiza za pomocą funkcji kwadratowej** dawała takie same wyniki jak ulepszona wersja algorytmu **Naive Bayes**. Okazuje się, że model probabilistyczny jeśli dobrze użyty potrafi być naprawdę przydatny w niektórych przypadkach pracy na danych.

Wykresy także pokazują, że **Model Bernoulliego** lepiej się sprawdził przy metodzie **Naive Bayes** co nie było dużym zaskoczeniem, bo nie tylko była to nowsza wersja biblioteki ale także sam model był dokładniejszy.

Najgorzej wypadł algorytm **najbliższych sąsiadów**, ale było to jednak także spodziewane. Algorytm ten staje się tym mniej dokładny im więcej mamy atrybutów do sklasyfikowania.

Sieć neuronowa nie poradziła sobie najlepiej jednak nie szukałbym w tym rezultacie złych przesłanek o samym sposobie implementacji, a bardziej o ilości danych. Około 2 tysiące danych to zdecydowanie za mało dla takiej sieci by potrafiła ona poprawnie przy tylu argumentach wykrywać prawidłową klasę. Szczerze mówiąc 82.5% to i tak dobry wynik jak na tak mało informacji. Prawdopodobnie gdybyśmy zebrali więcej informacji (tak powyżej 10 tysięcy) to nasza sieć neuronowa poradziłaby sobie znacznie lepiej.

Kończąc już porównanie algorytmów to najlepsze algorytmy do przeszukiwania danej bazy danych używek to: **Drzewa CIT i GBM, Funkcja Kwadratowa, SVM i Naive Bayes Bernoulliego**, których dokładność wynosiła 83.22034%.

Ciekawa sprawa jest taka, że gdy zmieniłem klasyfikacje alkoholu, dzieląc go na dwie klasy:

- Nie pity w ogóle
- Pity kiedykolwiek w życiu

To dokładność wszystkich algorytmów zwiększyła się o co najmniej 10%. Jednak należałoby zgasić ten entuzjazm prostym wnioskiem – przy takiej klasyfikacji osób które nigdy nie piły jest bardzo mało, więc nawet strzelając, że wszyscy ludzie ze zbiory testowego pili alkohol miałbym 82% poprawności wynikowej.

To także wyjaśnia dlaczego na wybrana na początku klasyfikacja alkoholu jest o wiele lepsza w celu sprawdzenia różnych algorytmów.

Warto też przeanalizować reguły asocjacyjne. Możemy dowiedzieć się z nich paru ciekawych informacji na przykład:

- Najwięcej alkoholu piją ludzie podczas studiów lub po ich ukończeniu
- W Wielkiej Brytanii bardzo dużo ludzi pije alkohol w porównaniu do osób nie pijących
- Mimo wszystko biali ludzie dominują w spożywaniu alkoholu