**Inteligencja Obliczeniowa**

Praca domowa nr 2 – Zgłębianie danych

Krzysztof Kulewski, 238149, grupa 1, 08.12.2018

## Opis bazy danych

Do realizacji zadania wybrano bazę ***Pima Indians Diabetes Database***, pochodzącą ze zbiorów *National Instutite of Diabetes and Digestive and Kidney Diseases*. Jest ona wynikiem badań przeprowadzonych na żeńskiej części populacji Indian z plemienia Pima, zamieszkujących stan Phoenix w USA. Plemię te jest pod ciągłą obserwacją z powodu wysokiego wskaźnika zachorowań na cukrzycę. Występowanie cukrzycy było określane na podstawie standardowych kryteriów *WHO*.   
  
Badana populacja składała się z **768** osób, które opisano za pomocą **9 atrybutów**:  
**- Pregnancies** – (liczba naturalna) – przebyte ciąże,  
**- Glucose** – (liczba naturalna) – poziom glukozy we krwi po 2 godzinach od zażycia dawki testowej,  
**- BloodPressure** – (liczba naturalna) – rozkurczowe ciśnienie krwi (mm/Hg),  
**- SkinThickness** – (liczba naturalna) – grubość fałdu skórnego zmierzonego na tricepsie (mm),  
**- Insulin** – (liczba naturalna) – poziom insuliny po 2 godzinach od zażycia dawki testowej,  
**- BMI** – (liczba zmiennoprzecinkowa) – indeks masy ciała, pośrednio opisuje zawartość tkanki tłuszczowej w organizmie na podstawie proporcji masy ciała mierzonej w kilogramach do wzrostu w metrach,  
**- DiabetesPedigreeFunction** (liczba zmiennoprzecinkowa) – funkcja wpływu genetycznego na występowanie cukrzycy (wyliczona na podstawie historia występowania cukrzycy u krewnych i stopień ich pokrewieństwa),  
**- Age** – (liczba całkowita) – wiek badanej osoby,  
**- Outcome** – (liczba binarna) –wynik badania, określający występowanie (1) cukrzycy lub jej brak (0).  
Atrybut ten wydaje się być naturalnym kandydatem na klasę zbioru danych.  
  
Celem projektu będzie stworzenie modelu, który na podstawie 8 atrybutów będzie w stanie trafinie przewidzieć wartość kolumny „Outcome”, czyli stwierdzić, czy dana osoba choruje na cukrzycę.

## Obróbka danych

Bardzo szybko możemy dostrzec, że w bazie są pewne niespójności. O ile w przypadku liczby ciąży wartość 0 jest jak najbardziej prawdopodobna, tak zdecydowanie nie jest to dobra wartość dla kolumn *Glucose*, *BloodPressure*, *SkinThickness* itd.

W takim przypadku możemy albo zignorować rekordy, które zawierają nieprawidłowe wartości, albo uzupełnić je wartościami średnimi. Baza stanowiąca przedmiot zadania jest relatywnie mała, stąd dużo lepsze będzie rozwiązanie drugie.  
  
Wyliczanie średnich wartości dla poszczególnych kolumn przebiegało według następującego schematu:

# baza jest załadowana do zmiennej ‘db’  
# wybieramy wartości większe od zera i tworzymy wektor  
non.zero.glucose = db$Glucose[which(db$Glucose>0)]  
# obliczamy średnią wartość  
glucose.mean = mean(non.zero.glucose)

Powyższą operację zastosowano dla kolumn: Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction i Age.  
  
Obliczone średnie zostały następnie zaokrąglone do odpowiedniej (wynikającej z danych) liczby miejsc po przecinku i podstawione w miejsca wystąpienia liczby 0:

# kopiujemy bazę do zmiennej db.nz (non-zero)   
db.nz = db  
# iterujemy po wszystkich wierszach  
for (i in 1:nrow(db))  
{  
 # jeśli wartość wynosi 0 – podstawiamy zaokrągloną średnią  
 if (db$Glucose[i] == 0) db.nz$Glucose[i] = round(glucose.mean)  
 if (db$BloodPressure[i] == 0) db.nz$BloodPressure[i] = round(bloodPressure.mean)  
 if (db$SkinThickness[i] == 0) db.nz$SkinThickness[i] = round(skinThickness.mean)  
 if (db$Insulin[i] == 0) db.nz$Insulin[i] = round(insulin.mean)  
 if (db$BMI[i] == 0) db.nz$BMI[i] = round(bmi.mean, digits = 1)  
 if (db$DiabetesPedigreeFunction[i] == 0) db.nz$DiabetesPedigreeFunction[i] =  
round(dpf.mean, digits = 3)  
 if (db$Age[i] == 0) db.nz$Age[i] = round(age.mean)  
}

Kolejnym krokiem jest normalizacja kolumn oraz zamiana wartości ‘0’ i ‘1’ na ‘healthy’ i ‘sick’:

# zmieniamy wartości kolumny Outcome z ‘0’ i ‘1’ na ‘healthy’ i ‘sick’  
db.nz$Outcome = factor(db.nz$Outcome, level=0:1, labels=c("healthy", "sick"))  
# normalizujemy dane numeryczne  
fnorm = function(x) { return ((x-min(x))/(max(x)-min(x))) }  
db.nz$Pregnancies = fnorm(db.nz$Pregnancies)  
db.nz$Glucose = fnorm(db.nz$Glucose)  
db.nz$BloodPressure = fnorm(db.nz$BloodPressure)  
db.nz$SkinThickness = fnorm(db.nz$SkinThickness)  
db.nz$Insulin = fnorm(db.nz$Insulin)  
db.nz$BMI = fnorm(db.nz$BMI)  
db.nz$DiabetesPedigreeFunction = fnorm(db.nz$DiabetesPedigreeFunction)  
db.nz$Age = fnorm(db.nz$Age)

## Klasyfikacja danych

## Źródła

1. Dane benchmarkowe  
<https://www.cs.ubc.ca/~hoos/SATLIB/benchm.html>  
  
2. Dokumentacja biblioteki GenAlg  
<https://cran.r-project.org/web/packages/genalg/genalg.pdf>

## Załączniki

1. Pliki źródłowe w języku R: fitness.r, parser.r, program.r  
2. Formuły CNF w formacie DIMACS  
3. Treść zadania (PDF)