10.12.2018

Błażej Kucman
ind. 238228
gr.1

Inteligencja Obliczeniowa – Projekt 2

Zgłębianie danych

Tematem zadania było przećwiczenie poznanych na laboratoriach technik zgłębiania danych na wybranym zbiorze danych (najlepiej z kolumną „class" z dwoma możliwymi wartościami).

1. Wstęp i omówienie bazy

Do zadania została wybrana baza z wyniki i danymi osób z problemami chorób serca. Na ich podstawie chcemy się dowiedzieć przy jakich wynikach oraz danych osoby jest możliwa choroba serca.

Baza składa się z kolumn:

Skrót ST – częstoskurcz zatokowy

age - wiek w latach,
sex – płeć(1=mężczyzna, 0= kobieta),
cp – rodzaj bólu w klatce piersiowej (wartości 1-4)
1.Typowa angina
2.Nietypowa angina
3.Nie anginowy ból
4.Bezobjawowy
trestbps - spoczynkowe ciśnienie krwi (w mm Hg przy przyjęciu do szpitala)
chol - poziom cholesterolu w surowicy w mg/dl
fbs – cukier we krwi(na czczo) > 120 mg/dl (1-tak, 0-nie)
restecg - spoczynkowe wyniki elektrokardiograficzne(wartości 0-3)
- Wartość 0: normalna
- Wartość 1: nieprawidłowość fali ST-T (odwrócenie załamka T i / lub uniesienie odcinka ST> 0,05 mV)
- Wartość 2: wykazanie prawdopodobnego lub określonego przerostu lewej komory według kryteriów Estesa.
thalach – osiągnięte maksymalne tętno
exang – angina wywołana wysiłkiem (1-tak , 0-nie)
oldpeak – spadek ST wywołany pod czas ćwiczeń względem stanu spoczynku
slope – nachylenie spadku ST pod czas ćwiczeń(0-3)
- Value 1: upsloping
- Value 2: flat
- Value 3: downsloping

ca - liczba głównych naczyń (wartości 0-3) zabarwionych za pomocą fluoroskopii
thal – 3 – normalny; 6 – wyleczony , 7- wada odwracalna
target – 0-zdrowy, (1-4) – chory

Jako klasa wybrana kolumna target została ona przekształcona tak aby przechowywała tylko wartości 0 i 1 gdzie 1 będzie znaczyło chory .

2. Przetwarzanie i obróbka danych

Przetworzona została kolumna target która przechowywała wartości od 0 do 4, na taką która będzie miała wartości 0 i 1. Kolumna ta będzie służyć jako kolumna klasy.

Wykorzystana została biblioteka deducorrect i funkcja editWithRules. Poniżej kod odpowiedzialny za to.

```
setSickCorrection <- correctionRules(expression(
 if(is.finite(target)){
  if(target != 0)
   {
     if (target == 1 | target == 2 | target == 3 | target == 4) {
       target <- 1
    } else
    {
      target <- NA
    }
  }
 }else
 {
  target <- NA
 }
))
```

```
heart.disase.SickSet <- (correctWithRules(setSickCorrection,heart.disase))$corrected
```

Po edycji zostaje wyciągnięta tylko część kolumn z danymi (corrected) pomijając informacje o tym które zostały zmienione.

Funkcja zmieniała także każdą wartość która nie pasuje do wzorca na i w następnym kroku zostaną usunięte wiersze które zawierają NA w target, ponieważ interesują nas tylko te które mają wartość poprawną

```
heart.disase.Better <- subset(heart.disase.SickSet,is.finite(target))
```

Za pomocą biblioteki editrules zostało sprawdzone czy w bazie znajdują się jakiekolwiek dane nie pasujące do wytycznych czy zakresów.

```
E <- editset(c(
        "age >0.0","sex %in% c(0,1)","cp %in% c(1,2,3,4)","trestbps > 0",
        "chol > 0", "fbs %in% c(1,0)","restecg %in% c(0,1,2,3)","thalach > 0",
        "exang %in% c(0,1)", "oldpeak >=0","slope %in% c(1,2,3)","ca %in% c(0,1,2,3)",
        "thal %in% c(3,6,7)"
        ))

ve <- violatedEdits(E,heart.disase.Better)
```

Dzięki temu wiadomo dla których kolumn należy pisać funkcje uzupełniające dane bądź usuwające wiersze.

Po przeanalizowaniu kolumn wybrana została metoda kNN z biblioteki VIM aby na podstawie najbliższych sąsiadów uzupełnić luki w bazi. Kolumny te przechowują dane ściśle określone np.(1 lub 0), więc średnia nie była by oczekiwana w tym wypadku.

```
heart.knn <- kNN(heart.disase.Better)
```

3. Klasyfikatory i ich ewaluacja

Kolumną klasy jest target. Przechowuje ona wartość 0 i 1 czyli zdrowy i chory.

a)

Baza została podzielona na treningową i testową w stosunku 67/33.

Wylosowanie gdzie trafi który wiersz.

```
ind <- sample(2, nrow(heart.disase.ready), replace=TRUE, prob=c(0.67, 0.33))
```

i podzielenie bazy

```
heart.disase.train <- heart.disase.ready[ind==1, 1:14]
heart.disase.test <- heart.disase.ready[ind==2, 1:14]
```

b)

Użycie poszczególnych klasyfikatorów

Klasyfikator C4.5/ID3 znajduje się w bibliotece *party*

```
heart.disase.ctree <- ctree(target ~ age + sex + cp + trestbps + chol + fbs + restecg +
                            thalach + exang + oldpeak + slope + ca + thal, data=heart.disase.train.ctree)
```

jego użycie polega stworzeniu modelu drzewa powyższym poleceniem gdzie pierwszym argument jest wynik który oczekujemy elementy podane po ~ to już elementy wnioskowania, należy także podać bazę treningową.

Analiza danych następuje po przez następujące polecenia.

```
predicted.ctree <- predict(heart.disase.ctree, heart.disase.test[,1:13])
trueSickValue <- heart.disase.test[,14]
conf.matrix.ctree <- table(predicted.ctree,heart.disase.test[,14])
conf.matrix.ctree
accuracy.ctree <- sum(diag(conf.matrix.ctree))/sum(conf.matrix.ctree)
accuracy.ctree
```

Polecenim predict uruchamia się model dla danych testowych.

Klasyfikator kNN

Do normalizacji wybrane zostały kolumny które posiadały pomiary i duże liczby aby nie powodować dużych różnic w dopasowaniach.

```
compute.model.norm.knn <- knn(heart.disase.norm.train[,1:13], heart.disase.norm.test[,1:13],
cl=heart.disase.norm.train[,14], k = 3, prob=FALSE)
```

Tworzenie modelu który wyliczy wyniki na bazie testowej odbywa się po przez funkcję knn, w której podawana jest baza treningowa z kolumną target a testowa bez.
Analiza przebiega tak samo w C4.5/ID3

Klasyfikator Naive Bayes

```
modelHeart <- naiveBayes(target ~ ., data = heart.disase.train.NB)
```

Tworzenie modelu przebiega bardzo prosto podajemy kolumnę z docelowym wynikiem i nazwę bazy treningowej .

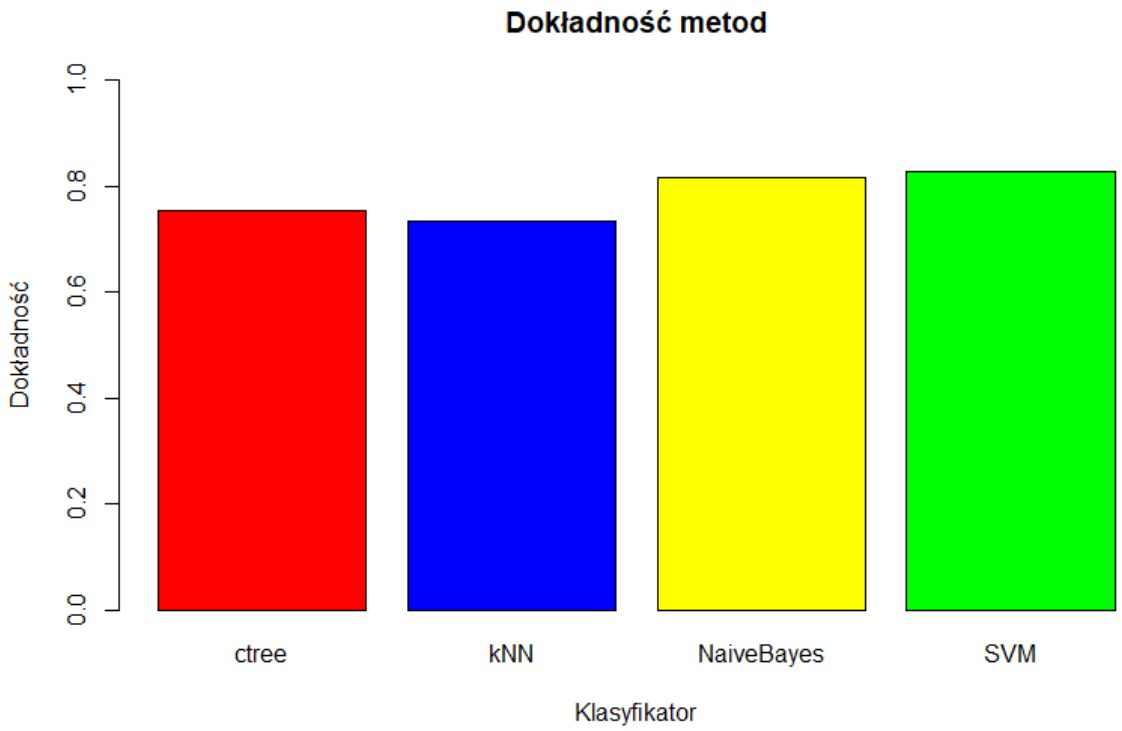
Analiza podobnie jak w pierwszym.

Dodatkowym klasyfikatorem jest SVM Support Vector Machines.

```
model.SVM <- svm(target ~ ., data = heart.disase.norm.train, type='C-classification',kernel='linear',
        scale=FALSE)
```

Porównanie wyników

Klasyfikiator	Macierz błędu	Dokładność	TPR	FPR
Ctree	Predicted 0 1 Real 0 49 7 1 17 25	0.755102	0.875	0.4047619
kNN	Predicted 0 1 Real 0 40 16 1 10 32	0.7346939	0.7142857	0.2380952
naiveByes	Predicted 0 1 Real 0 46 10 1 8 34	0.8163265	0.8214286	0.1904762
SVM	Predicted 0 1 Real 0 48 8 1 9 33	0.8265306	0.8571429	0.2142857



c)

TP – oznacza liczbę osób, którym zgadł że są zdrowe

FP – oznacza liczbę pomylonych osób, które oznaczył jako chore a były zdrowe

FN – oznacza liczbę pomylonych osób, które oznaczył jako zdrowe a były chore

TN – oznacza liczbę osób, którym zgadł że są chore

TPR przedstawia odsetek poprawnie zdiagnozowanych(zdrowych) a TNR poprawnie zdiagnozowanych(chorych). Natomiast wartości FPR i FNR przedstawiają odsetki źle zdiagnozowanych. FNR jest dopełnieniem TPR. FPR dopełnieniem TNR

Czułość, TPR $\frac{\sum TP}{\sum TP+\sum FN}$	FNR $\frac{\sum FN}{\sum TP+\sum FN}$
FPR $\frac{\sum FP}{\sum FP+\sum TN}$	Swoistość, SPC, TNR $\frac{\sum TN}{\sum FP+\sum TN}$

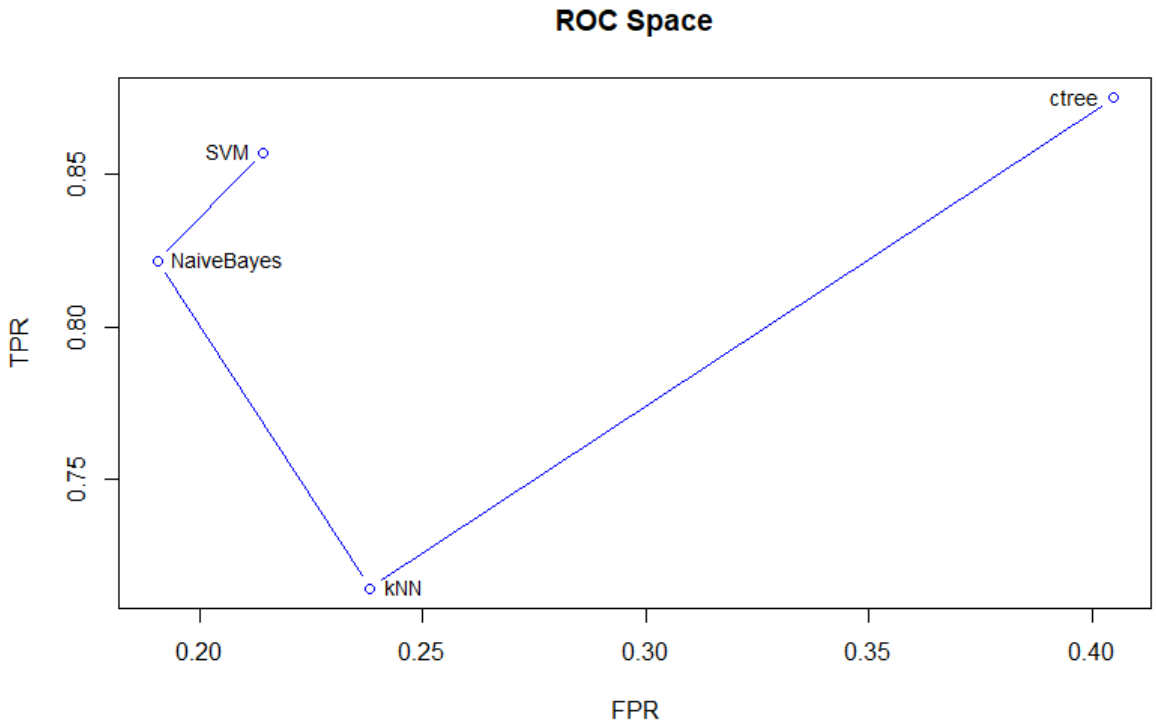
Błąd pierwszego stopnia u mnie to liczba osób źle sklasyfikowanych jako zdrowi a są chorzy.

Błąd drugiego stopnia u mnie to liczba osób zdrowych sklasyfikowanych jako chore.

Im więcej błędów drugiego stopnia to więcej osób mogło by być niepotrzebnie leczonych.

TNR i FPR przedstawiają skuteczność testu. A FNR i FPR nieskuteczność.

W mojej bazie i sposobie modelowania gorszy był by błąd pierwszego stopnia, ponieważ osoby chore nie były by leczone.



4. Grupowanie metodą k-średnich

Metoda k-średnich nie działa idealnie. Żadna z prób podzielenia na 2,3,4 czy 5 klastrów nie dała grup gdzie znajdziemy tylko chorych lub zdrowych. Przy podziale na dwie grupy skuteczność wyniosła ~ 0.835

Real	Grupy 1	2
0	151	13
1	37	102

Czyli w jednej grupie 1 algorytm umieścił (zdrowi - wywnioskowane) było 151 osób zdrowych i 37 chorych, a w grupie 2 (chorzy) 102 chorych i 13 zdrowych. Dokładność wydaję się lepsza niż wśród Klasyfikatorów, ale trzeba wziąć pod uwagę że algorytm miał do dyspozycji całą bazę choć bez target.

Podsumowując można być zadowolonym wynikiem, ponieważ skuteczność jest porównywalna z najlepszy klasyfikatorem.

Dla przykładu wyniki dla 3,4 i 5 klastrów.

<table>
<tr><th>Ilość klastrów</th><th>Tabel</th><th>Skuteczność</th></tr>
<tr><td>3</td><td>Grupy
Real
<table><tr><th></th><th>1</th><th>2</th><th>3</th></tr><tr><td>0</td><td>10</td><td>86</td><td>68</td></tr><tr><td>1</td><td>95</td><td>34</td><td>10</td></tr></table></td><td>~0.8218</td></tr>
<tr><td>4</td><td>Grupy
Real
<table><tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th></tr><tr><td>0</td><td>8</td><td>23</td><td>66</td><td>67</td></tr><tr><td>1</td><td>76</td><td>20</td><td>7</td><td>36</td></tr></table></td><td>~0.76567</td></tr>
<tr><td>5</td><td>Grupy
Real
<table><tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr><tr><td>0</td><td>5</td><td>13</td><td>63</td><td>63</td><td>20</td></tr><tr><td>1</td><td>56</td><td>28</td><td>27</td><td>7</td><td>21</td></tr></table></td><td>~0.76237</td></tr>
</table>

Można dostrzec, że zwiększanie ilości klastrów pogarsza jakość grup, i bardzo złe grupy przeplatają się dobrymi grupami. Dla naszej bazy wydaję się najkorzystniejszy podział na dwie grupy, lecz niektóre grupy z większej ilości klastrów mogą być korzystne.

Kod odpowiedzialny za grupowanie:

```
heart.disase.logPrep <- heart.disase.norm[,1:13]
heart.disase.log <- log(heart.disase.logPrep)
heart.disase.stand <- scale(heart.disase.logPrep, center=TRUE)
heart.disase.pca <- prcomp(heart.disase.stand)
heart.disase.final <- predict(heart.disase.pca)[,1:13]
```

Użycie algorytmu grupowania

```
k <- kmeans(heart.disase.final, 5)
```

5. . Reguły asocjacyjne

Użyta została biblioteka arules. Baza jest dość nieprzyjazna w tej metodzie poszukiwań ponieważ zawiera część danych pomiarowych, dla tego badania wykonywane są tylko na części kolumn.

Do badania wybrano kolumny : sex,cp,restecg,exang,thal,target a jako wynikowa wybrana sex,

Jedna asocjacja połączyła 4 informacje dla części osób
{restecg=0,exang=1,thal=3,target=1} => {sex=0} co znaczy że były kobiety u których powiązane były aspekty choroby . Co ciekawe oznacza to że osoby te zachorowały mając normalne wyniki poza zachorowaniem na anginę.

Kolejną ciekawą asocjacją w tym badaniu było {cp=1,restecg=0,thal=3,target=0} => {sex=0} gdzie przy normalnych wynikach i zachorowaniu na anginę nie wystąpiła choroba serca.

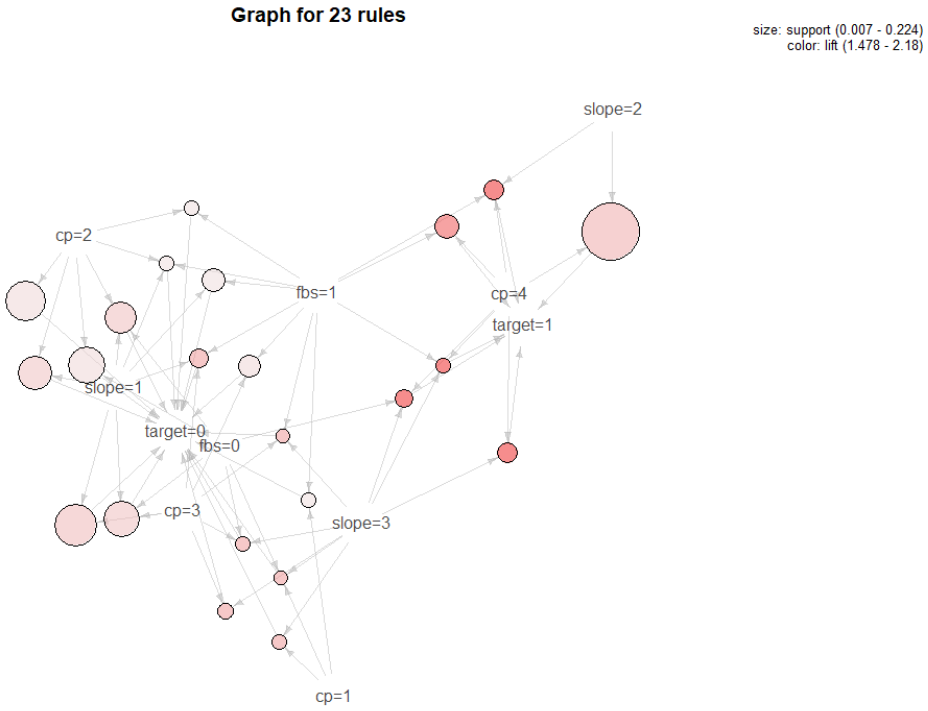
Do drugiego badania wybrano kolumny: cp, fbs, slop, target gdzie oczekiwaną jest target.

Ciekawym jest zestawienie cp=4 czyli braku bólu w klatce piersiowej i jednocześnie slope=4 czyli obniżenie częstoskurczu naczyniowego: `{cp=4,slope=3} target=1`

```
Porównie dwych podobnych asocjacji
{cp=4,fbs=1,slope=2} => {target=1}
{cp=2,fbs=0,slope=1} => {target=0}
```

W pierwszym brak bólu jest powiązany z za dużym cukrem i płaskim częstoskurczem i daje nam chorobę.
W drugim mammy nietypową anginę i brak podwyższonego cukru oraz wzrosty częstoskurczu i brak choroby. Gdzie można by pomyśleć że wynik powinien być odwrotny.

Przykładowy graf asocjacji.



Kod odpowiedzialny za tworzenie reguł

```
rules <- apriori(mat1,
        parameter = list(minlen=2, supp=0.005, conf=0.8),
        appearance = list(rhs=c("sex=0","sex=1"), default="lhs"),
        control = list(verbose=F)
)
```

Oraz przetworzenie ich.

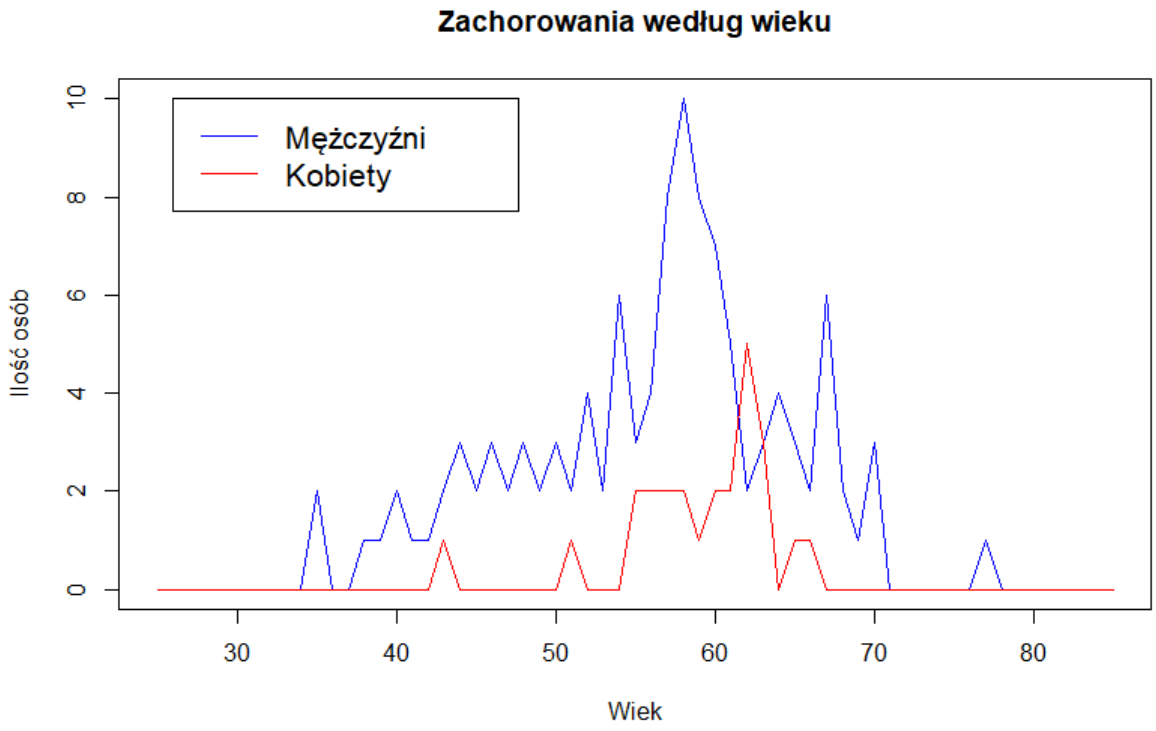
```
rules.sorted <- sort(rules, by="lift")
subset.matrix <- is.subset(rules.sorted, rules.sorted)
subset.matrix[lower.tri(subset.matrix, diag=T)] <- FALSE
redundant <- colSums(subset.matrix, na.rm=T) >= 1
rules.pruned <- rules.sorted[!redundant]
```

6. Dodatkowe wykresy

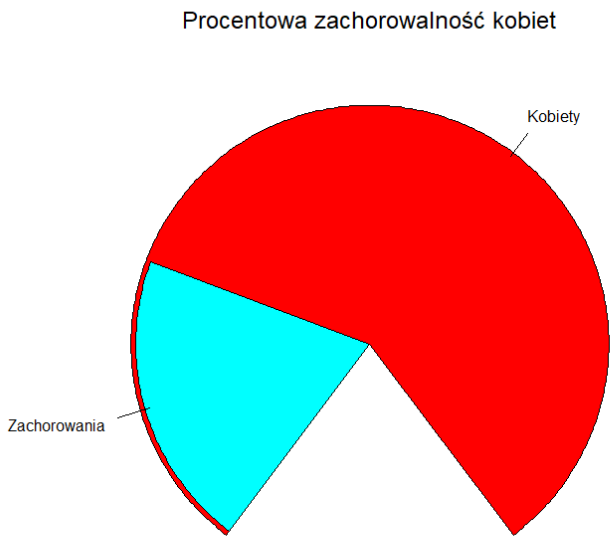
Liczba badanych mężczyzn 206 a kobiet 97.

Wiek minimalny kobiet 34, maksymalny 76.

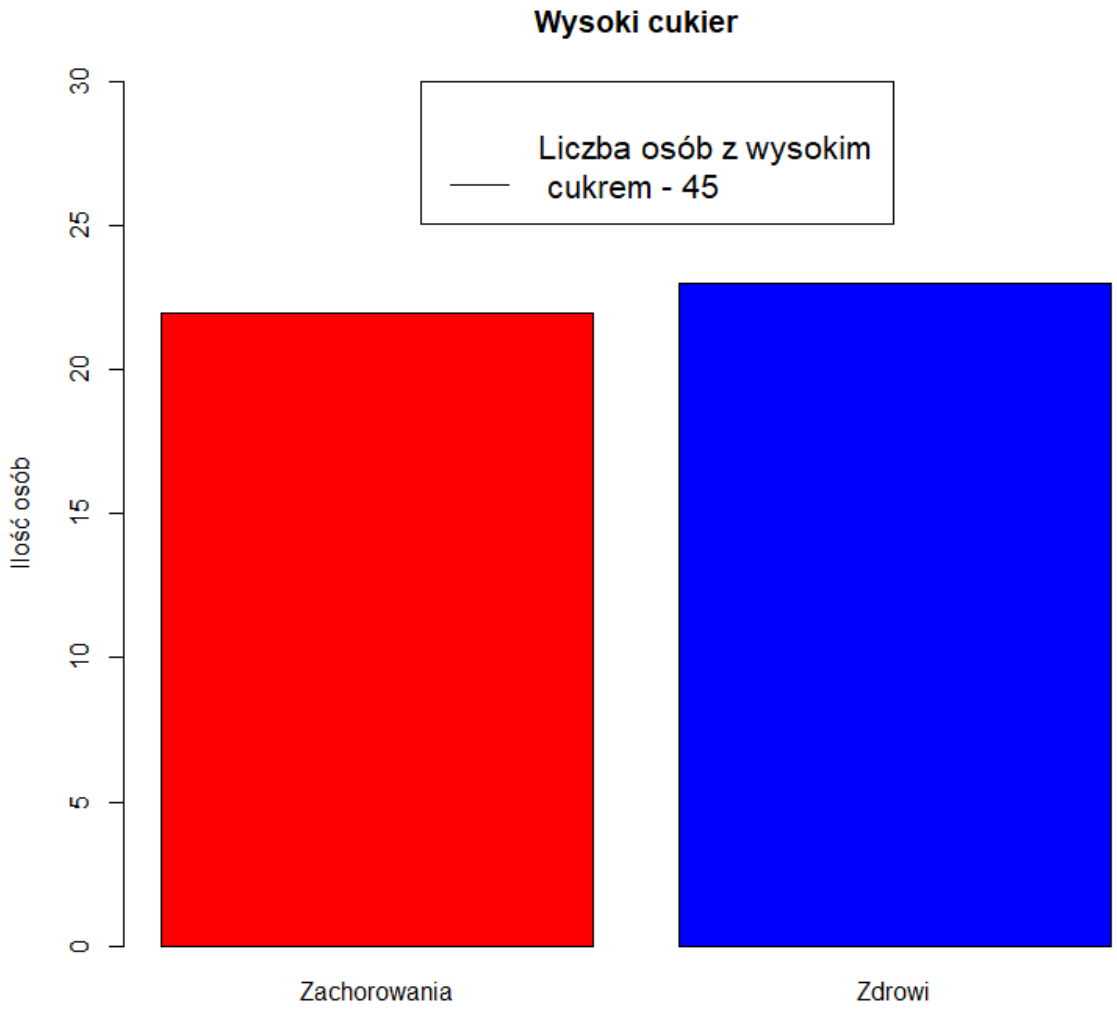
Wiek minimalny mężczyzn 29, maksymalny 767



Wykres przedstawia zachorowania według wieku podzielone na płcie. Można zauważyć trend wzrostowy pomiędzy 50 i 60 rokiem życie dla kobiet i mężczyzn, który utrzymuje się do 70 roku życia. Celem wykresu było zauważenie trendów.



W grupie badanych gdy prześledzić procentowy udział zachorowań według płci, wychodzi że mężczyźni mają znacznie większą tendencję do zachorowania.



Powyższy wykres pokazuje stosunek zachorowań osób z wysoki cukrem >120mg/dl. Przewidywałem że ma on duży wpływ na choroby serce lecz w tej grupie badanych okazało się, połowa osób był chora a połowa nie. Więc to ma tylko częściowy wpływ. Rozkład 22 do 23 osób.

Dodatkowo przeprowadziłem badanie jak 3 klasyfikatory poradziły sobie bez normalizacji.

Klasyfikator	Macierz błędu (norm)	Macierz błędu (bez norm)	Dokładność
kNN	Predicted / Real: 0 → 0: 40, 1: 16; 1 → 0: 10, 1: 32	Predicted / Real: 0 → 0: 35, 1: 21; 1 → 0: 15, 1: 27	Z norm. 0.7346939 Bez norm. 0.6326531
naiveByes	Predicted / Real: 0 → 0: 46, 1: 10; 1 → 0: 8, 1: 34	Predicted / Real: 0 → 0: 46, 1: 10; 1 → 0: 8, 1: 34	Identyczna 0.8163265
SVM	Predicted / Real: 0 → 0: 48, 1: 8; 1 → 0: 9, 1: 33	Predicted / Real: 0 → 0: 46, 1: 10; 1 → 0: 9, 1: 33	Z norm. 0.8265306 Bez norm. 0.8061224

7. Podsumowanie

Klasyfikatory radzą sobie w miarę sprawnie biorąc nie tak wielką próbkę danych 303 rekordy. Głównym problem było przewidywanie choroby i na 80% można to przewidzieć w tym wypadku.

Najskuteczniejszy z klasyfikatorów był SVM, ale dobrze poradziła sobie też metoda k-średnich. Zaskakujące były asocjacje, które był niekiedy sprzeczne z tym co można by myśleć, i również to że ciężko na tej bazie operować asocjacjami . Najlepszą drogą było zmniejszenie zakresu większości kolumn do kilku wartości sztucznie nazwanych, tylko wkładając wyniki pomiarów do szuflad o określonym zakresie można by zakłamać nieco wyniki.