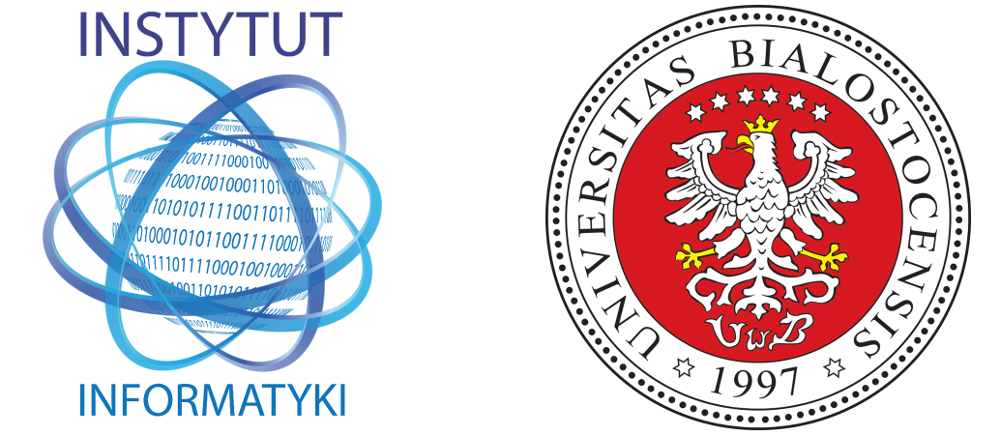
Uniwersytet w Białymstoku

Instytut informatyki



Projekt Sztuczna inteligencja

2021/2022

Wykonał: Dawid Metelski

Prowadzący: dr hab. Anna Gomolińska

**Podstawowe informacje**

Sprzęt:

* Procesor: Intel Core i5-9400F 2.9GHz
* Pamięć RAM: 16GB 3000MHz
* Karta graficzna: MSI GeForce GTX 1660
* System operacyjny: Windows 10 64bit

Wykorzystane środowisko:

* RStudio

Wykorzystane biblioteki:

* randomForest
* neuralnet
* kernlab
* naivebayes
* cluster
* ClusterR

**Wykorzystane zbiory danych**

1. Haberman's Survival Data Set

Źródło: <https://archive.ics.uci.edu/ml/datasets.php>

Ilość obiektów: 306

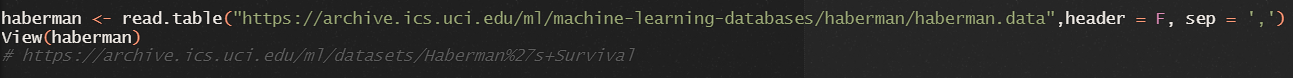
Ilość atrybutów: 4

Brakujące wartości: 0

Lista atrybutów:

* Age of patient at time of operation : numerical
* Patient's year of operation : numerical
* Number of positive axillary nodes detected : numerical
* Survival status - atrybut decyzyjny (2 klasy) : numerical (1 or 2)

Wczytanie zbioru i przypisanie nazwy:



1. South German Credit Data Set

Źródło: [https://archive.ics.uci.edu/ml/dataset](https://archive.ics.uci.edu/ml/datasets.php)

Ilość obiektów: 1000

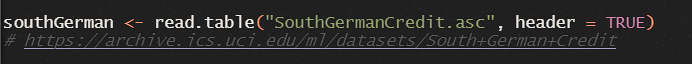
Ilość atrybutów: 21

Brakujące wartości: 0

Lista atrybutów:

* laufkont - atrybut decyzyjny (4 klasy) : numerical
* laufzeit : numerical
* moral : numerical
* verw : numerical
* hoehe : numerical
* sparkont : numerical
* beszeit : numerical
* rate : numerical
* famges : numerical
* buerge : numerical
* wohnzeit : numerical
* verm : numerical
* alter : numerical
* weitkred : numerical
* wohn : numerical
* bishkred : numerical
* beruf : numerical
* pers : numerical
* telef : numerical
* gastarb : numerical
* kredit : numerical

Wczytanie zbioru i przypisanie nazwy:



**Metody klasyfikacji**

Wybrane metody:

* Random Forest
* K najbliższych sąsiadów
* Naive Bayes’a
* Neuralnet

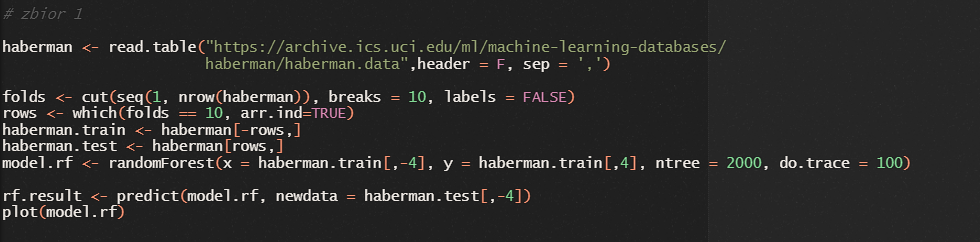
**Metoda Random Forest**

Las losowy, losowylas decyzyjny – metoda uczenia maszynowego dla klasyfikacji lub regresji, która polega na konstruowaniu wielu drzew decyzyjnych w czasie uczenia i generowaniu klasy, która jest dominantą klas (klasyfikacja) lub przewidywaną średnią (regresja) poszczególnych drzew.

Do zaimplementowania metody wykorzystano bibliotekę randomForest, ilość drzew (ntree) wynosiła 2000, a parametr do.trace został ustawiony na 100. Podział zbiorów: 80% zbiór treningowy, 20% zbiór testujący.

**Zbiór Haberman's Survival Data Set**

Użyty kod:



Otrzymane wyniki:

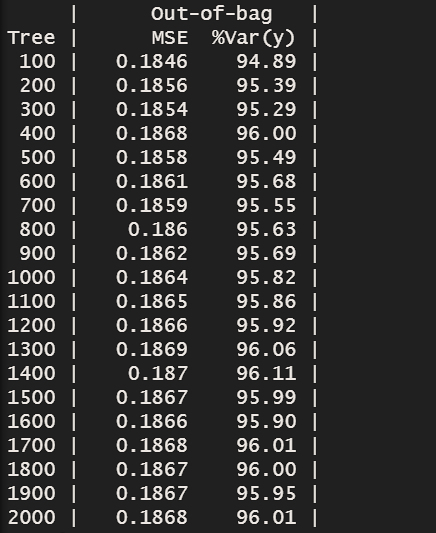
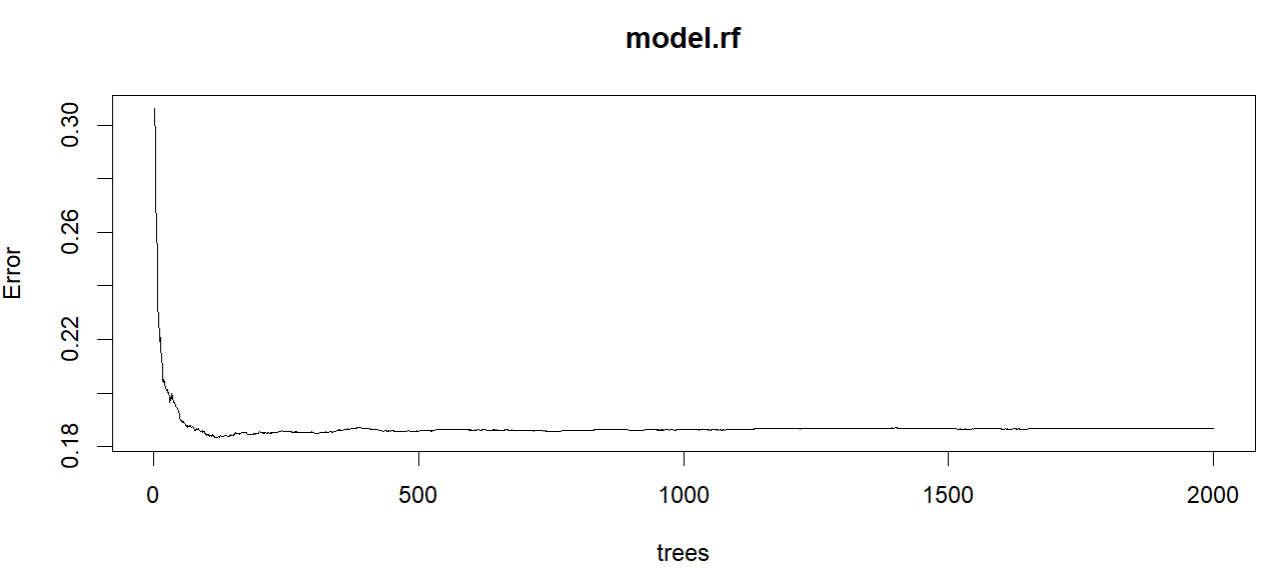


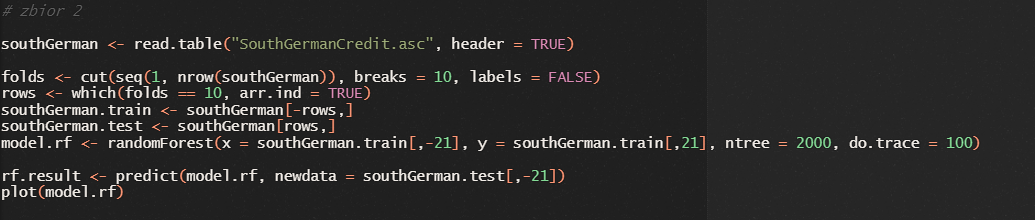
Tabela przedstawia wyniki MSE (czyli błąd średniokwadratowy) i % dokładności, dla poszczególnej ilości drzew. Średnia wartość błędu średniokwadratowego wyniosła 0,186305, a średnia dokładność 95,762%.



Wykres przedstawiający zestawienie wyników błędu (oś y) dla określonej ilości drzew (oś x).

**Zbiór South German Credit Data Set**

Użyty kod:



Otrzymane wyniki:

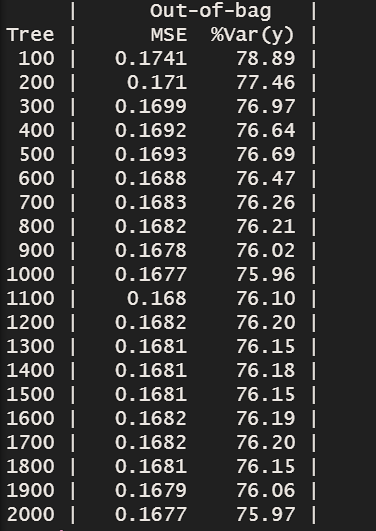
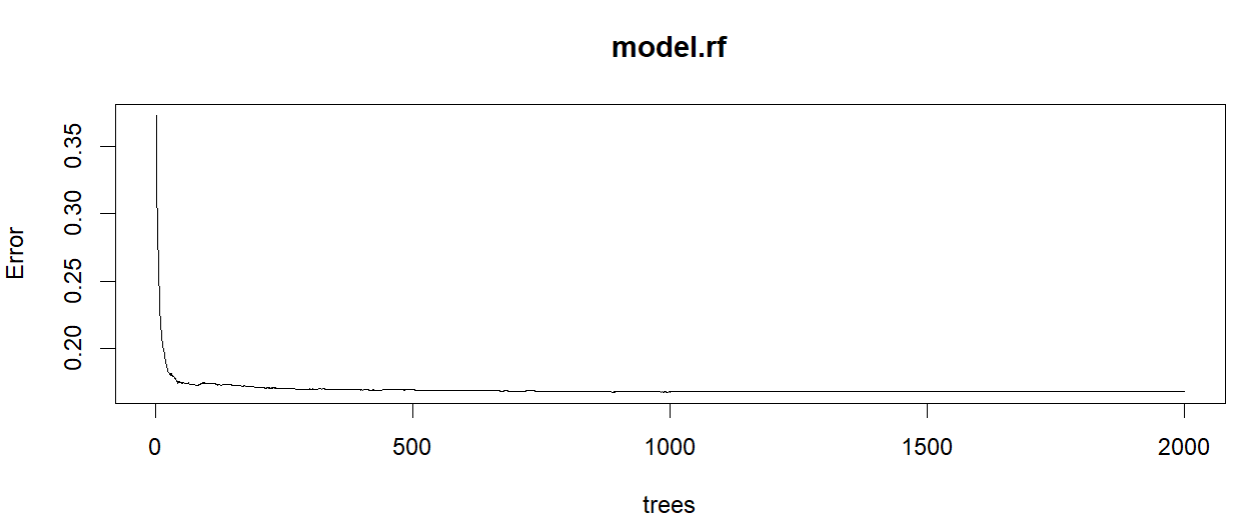


Tabela przedstawia wyniki MSE (czyli błąd średniokwadratowy) i % dokładności, dla poszczególnej ilości drzew. Średnia wartość błędu średniokwadratowego wyniosła 0,168745, a średnia dokładność 76,446%.



Wykres przedstawiający zestawienie wyników błędu (oś y) dla określonej ilości drzew (oś x).

**Wnioski:**

Metoda las losowy uzyskała dużo lepsze wyniki dla zbioru Haberman’s Survival Data Set. Różnica średnich dokładności między zbiorami wyniosła 19,316%. Może być to spowodowane mniejszą ilością obserwacji lub atrybutów przeważającego zbioru.

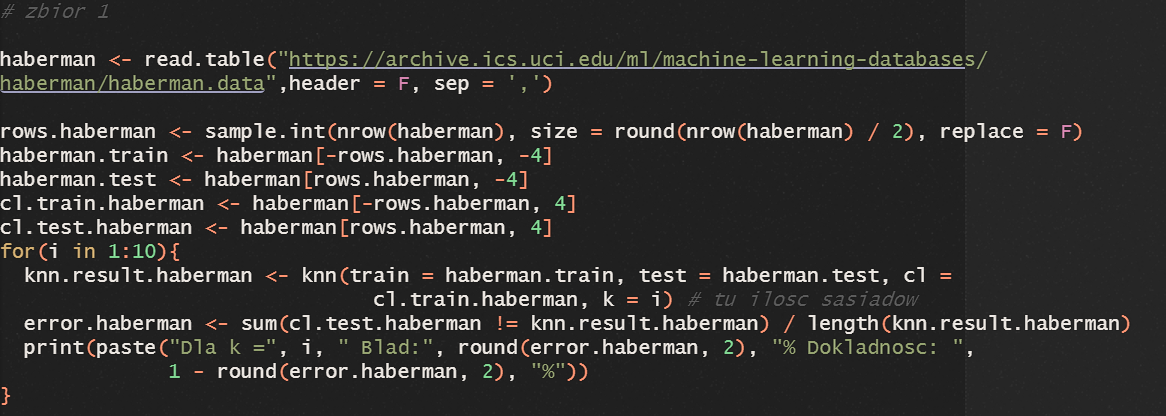
**Metoda k najbliższych sąsiadów**

Metoda K-najbliższych sąsiadów stosuje pamięciowy (memory-based) model zdefiniowany przez zestaw "przykładowych" przypadków o znanych wartościach zmiennej wyjściowej (i zmiennych wejściowych). Zarówno zmienne zależne jak i niezależne mogą być ciągłe i skategoryzowane.

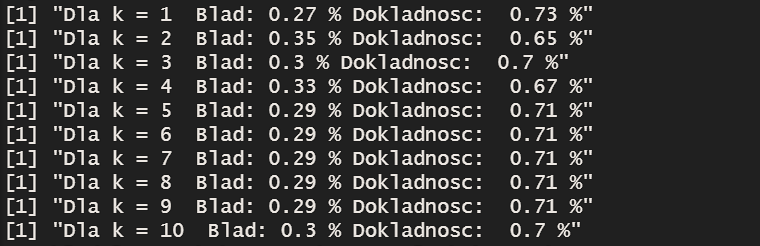
Do implementacji metody wykorzystano bibliotekę class. Metoda dla każdego zbioru była wywoływana dla 10 różnych wartości k (ilości sąsiadów). Podział zbiorów: 80% zbiór treningowy, 20% zbiór testujący.

**Zbiór Haberman's Survival Data Set**

Użyty kod:



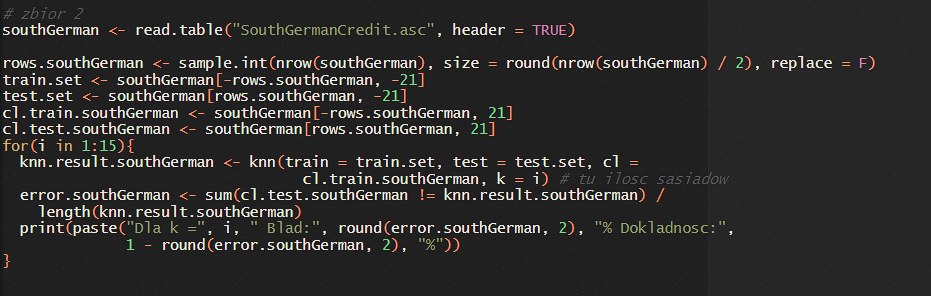
Otrzymane wyniki:



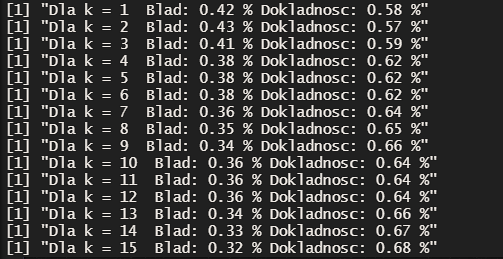
Wynik przedstawia wartość błędu i dokładności, dla każdego k. Średnia dokładność wyniosła 70%, a błąd 30%.

**Zbiór South German Credit Data Set**

Użyty kod:



Otrzymane wyniki:



Wynik przedstawia wartość błędu i dokładności, dla każdego k. Średnia dokładność wyniosła 63.2%, a błąd 36.8%.

**Wnioski:**

Metoda uzyskała dość słabe, zbliżone do siebie wyniki. Oznaczać to może, że różnica obserwacji i atrybutów zbiorów nie spowodowała dużej różnicy wyników.

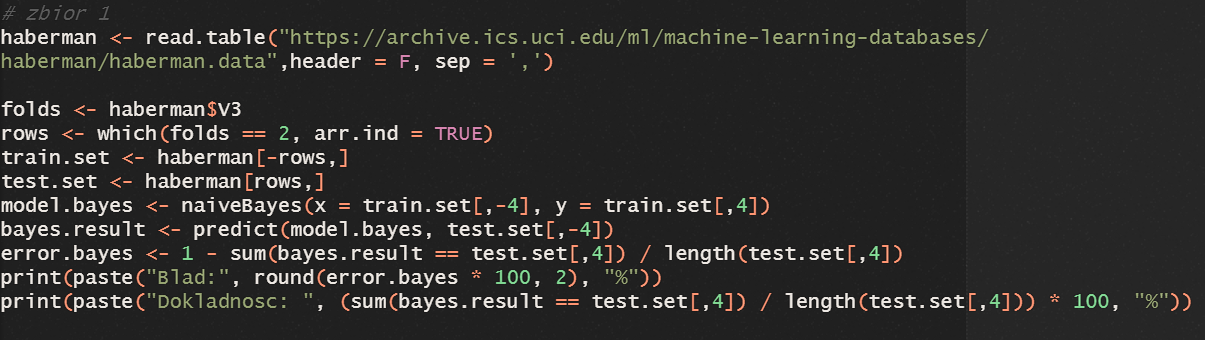
**Metoda Naive Bayes’a**

Naiwny klasyfikator bayesowski, naiwny klasyfikator Bayesa – prosty klasyfikator probabilistyczny. Naiwne klasyfikatory bayesowskie są oparte na założeniu o wzajemnej niezależności predyktorów (zmiennych niezależnych). Często nie mają one żadnego związku z rzeczywistością i właśnie z tego powodu nazywa się je naiwnymi. Bardziej opisowe jest określenie – „model cech niezależnych”. Ponadto model prawdopodobieństwa można wyprowadzić korzystając z twierdzenia Bayesa.

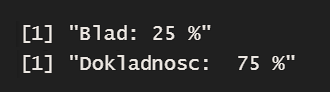
Do zaimplementowania metody wykorzystano bibliotekę naivebayes. Podział zbiorów: 80% zbiór treningowy, 20% zbiór testujący.

**Zbiór Haberman's Survival Data Set**

Użyty kod:

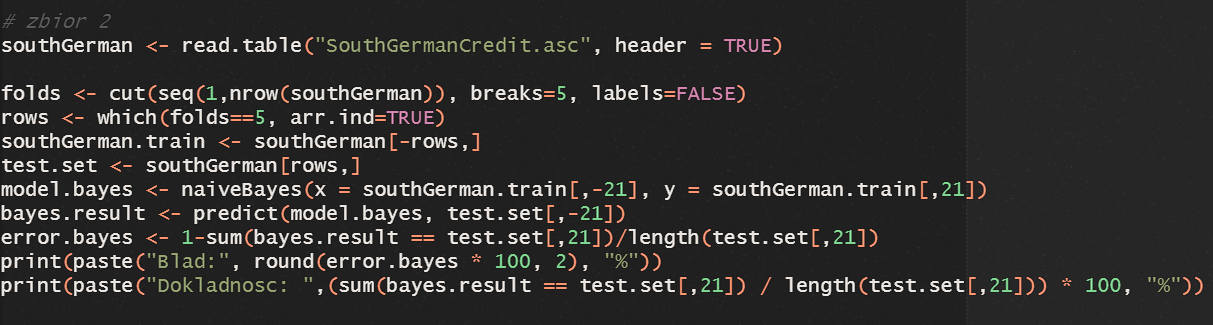


Otrzymane wyniki:

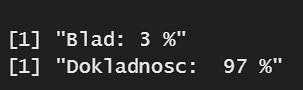


**Zbiór South German Credit Data Set**

Użyty kod:

****

Otrzymane wyniki:



**Wnioski:**

Metoda Bayes’a uzyskała dużo lepszy wynik dla zbioru South German Credit Data Set. Dokładność wyniosła 97% i jest o 22% wyższa od dokładności dla zbioru Haberman’s Survival Data Set (75%).

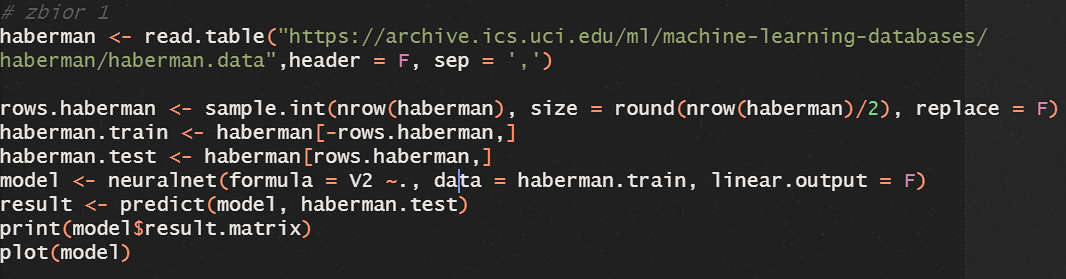
**Metoda Neuralnet**

Metoda sieci neuronowej wewnętrznie rozdziela rekordy między podzbiór budowania modelu oraz zbiór zabezpieczający przed “przeuczeniem”, który jest niezależnym zbiorem rekordów danych używanym do śledzenia błędów podczas uczenia i zapobiegania modelowaniu przez metodę zmienności prawdopodobieństwa w danych.

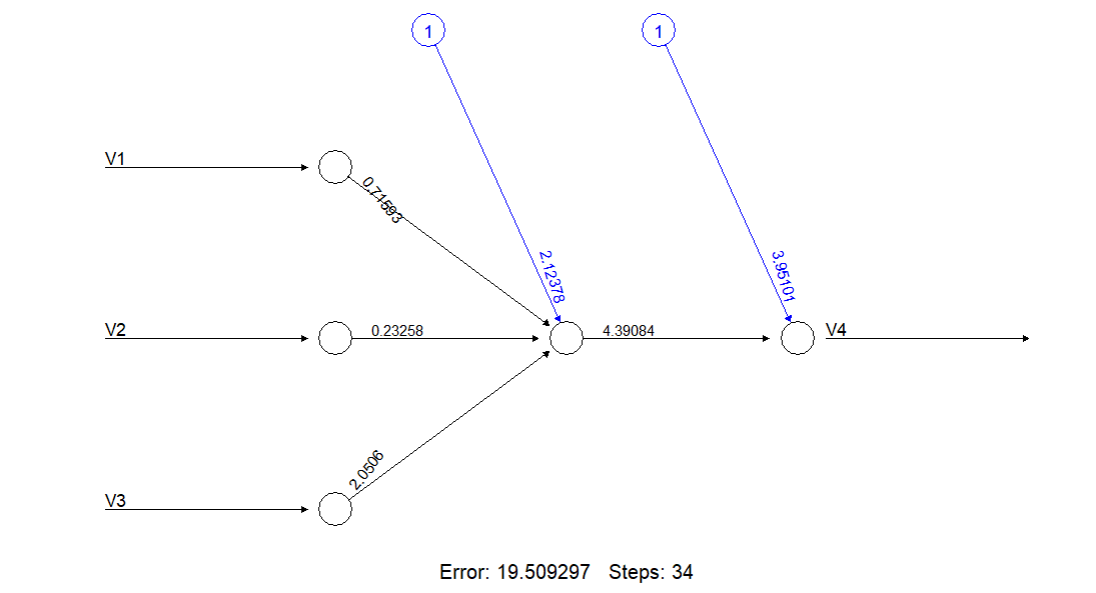
Do zaimplementowania metody wykorzystano biblioteki neuralnet i kernlab. Podział zbiorów: 80% zbiór treningowy, 20% zbiór testujący.

**Zbiór Haberman's Survival Data Set**

Użyty kod:



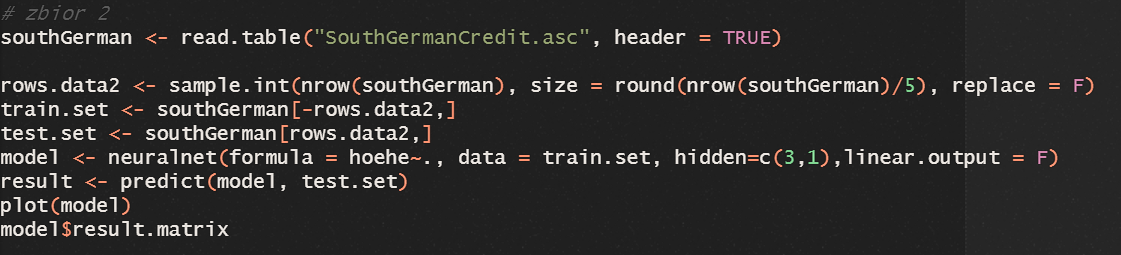
Otrzymane wyniki:

****

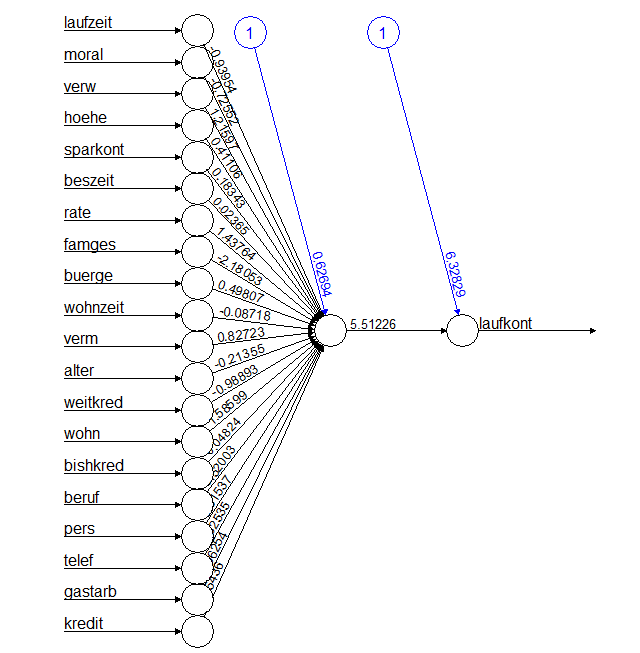
Wykres ukazuje trzy neurony w warstwie ukrytej, z wagami: 0.71593, 0.23258, 2.0506 i jeden w warstwie wejściowej o wadze 4.39084. Błąd wyniósł w przybliżeniu 19.51%.

**Zbiór South German Credit Data Set**

Użyty kod:



Otrzymane wyniki:

****

Wykres przedstawia 20 neuronów w warstwie ukrytej i jeden w warstwie wyjściowej o wadze 5.51226. Błąd wyniósł w przybliżeniu 15%.

**Wnioski:**

Metoda uzyskała gorszą dokładność dla zbioru Haberman’s Survival Data Set, wyniosła ona 80,49%. Dokładność dla drugiego zbioru była równa 85%.

**Metody grupowania**

Wybrana metoda:

* k średnich (k-means)

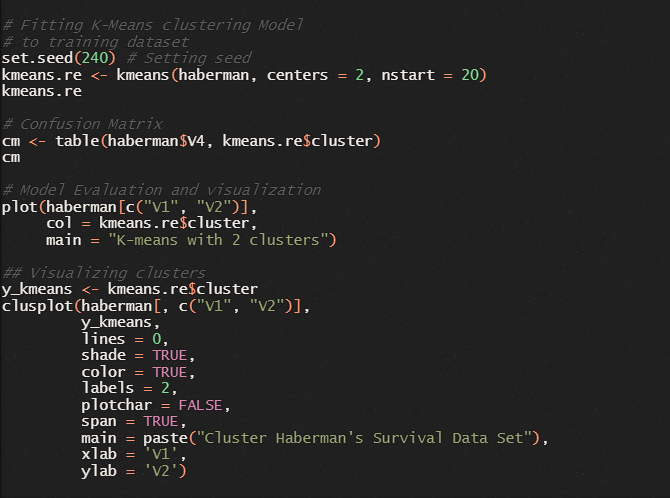
**Metoda k - średnich**

Algorytm k-średnich (z ang. k-means) inaczej zwany również algorytmem centroidów, służy do podziału danych wejściowych na z góry założoną liczbę klas. Jest to jeden z algorytmów stosowany w klasteryzacji (grupowaniu) .

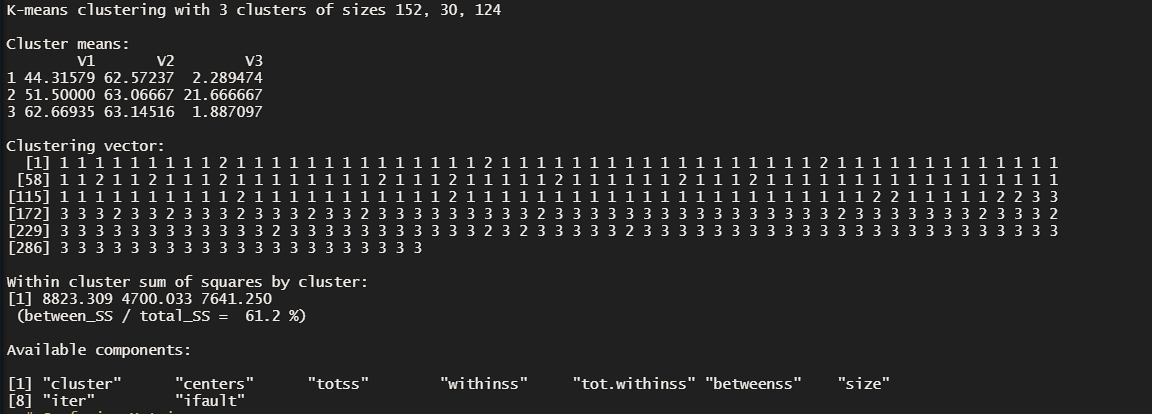
Do zaimplementowania metody wykorzystano biblioteki cluster i ClusteR. Ilość grup, na jakie podzielone zostały zbiory, zależna była od ilości klas atrybutu decyzyjnego.

**Zbiór Haberman's Survival Data Set**

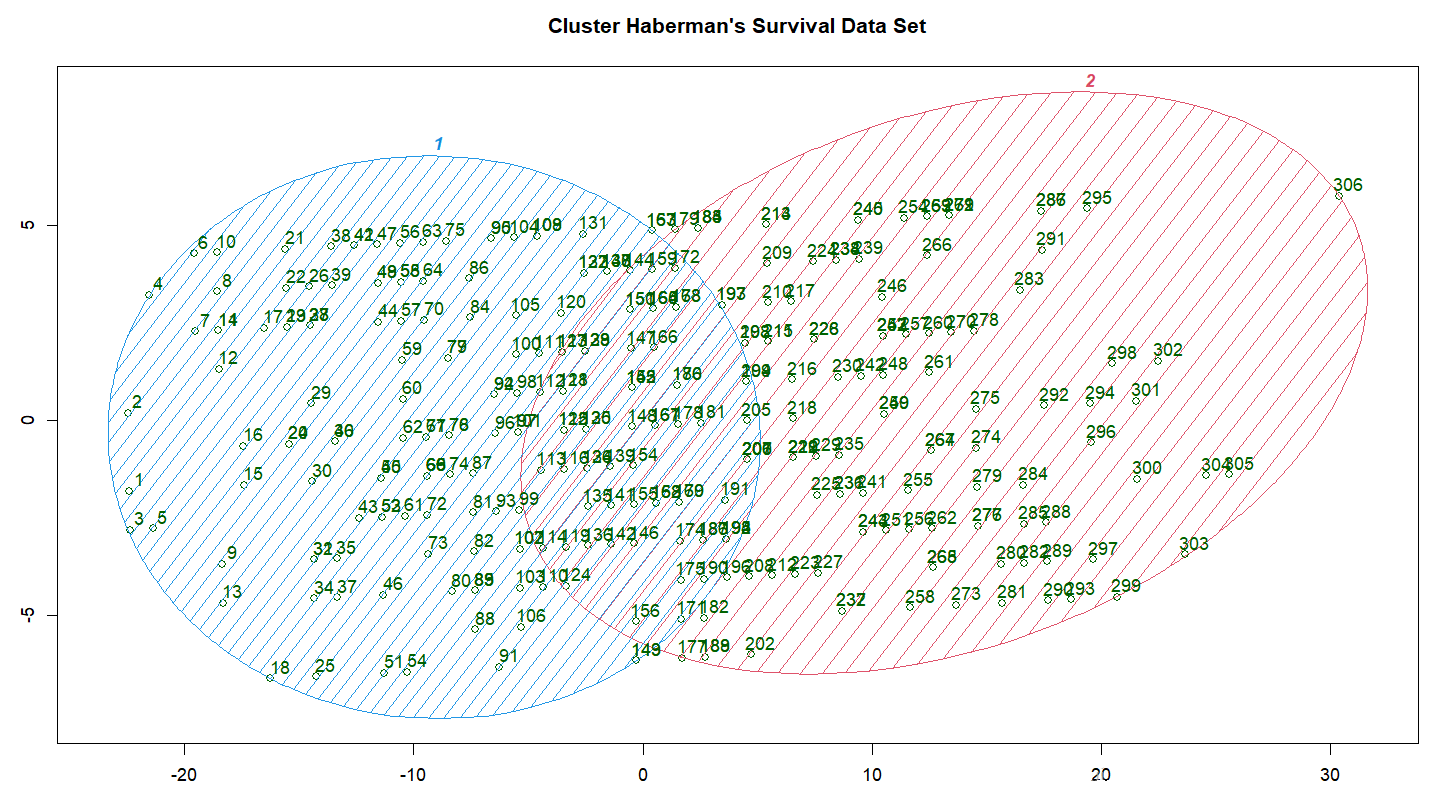
Użyty kod:



Otrzymane wyniki:

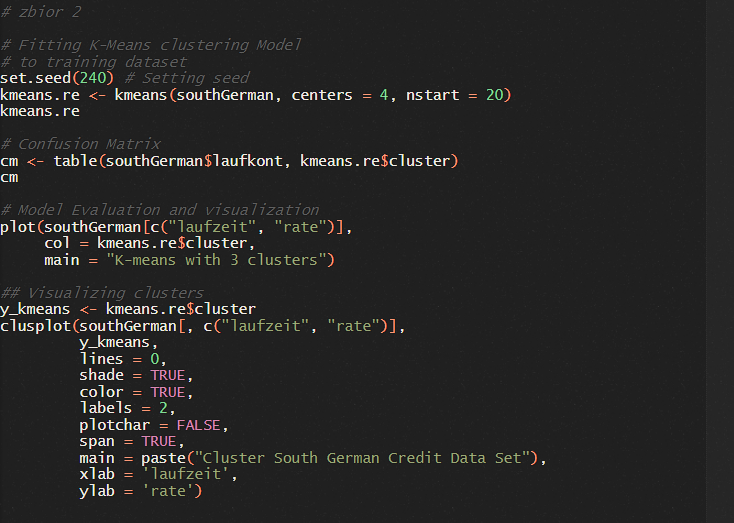
****

Wynik przedstawia model kmeans. Widzimy, że suma kwadratów reszt wyniosła 61.2%.

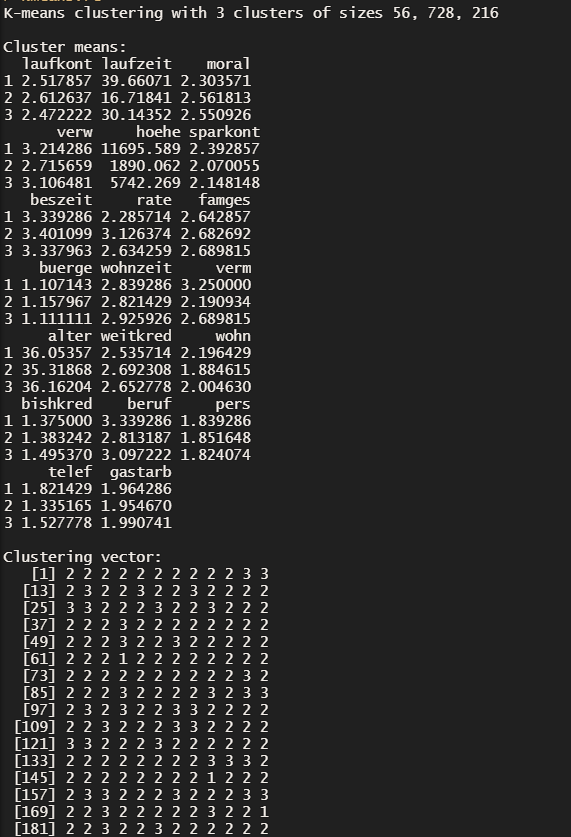
****

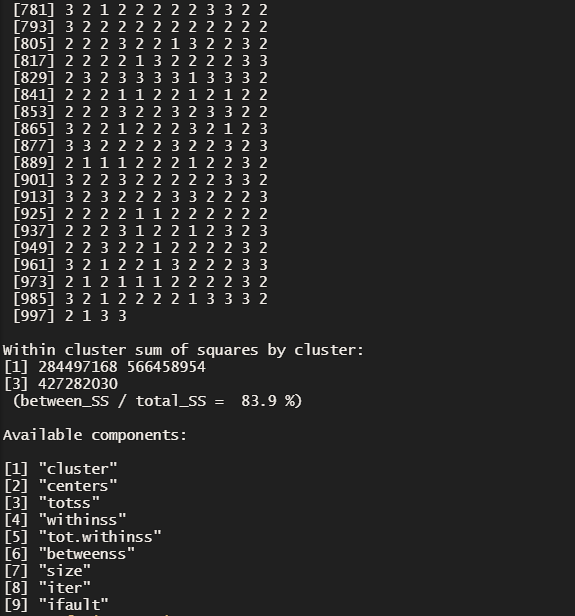
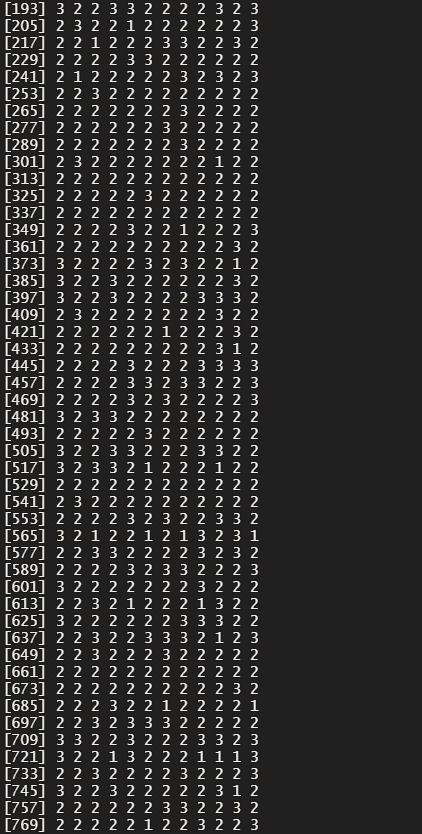
Obraz przedstawia dane zbioru Haberman’s Survival Data Set podzielone na dwie grupy. Grupowanie zostało przeprowadzone dla dwóch grup, ponieważ atrybut decyzyjny zawiera dwie klasy.

**Zbiór South German Credit Data Set**

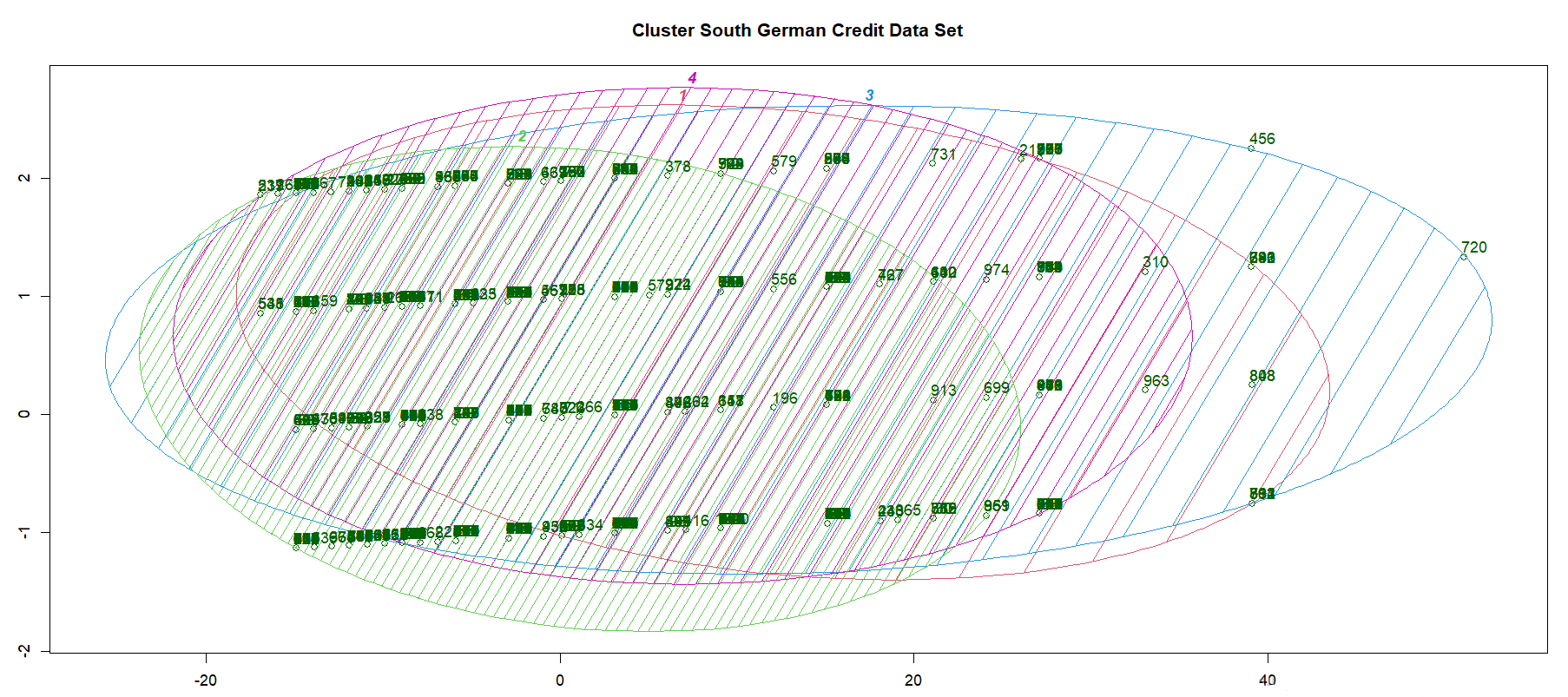
Użyty kod:  


Otrzymane wyniki:





Wynik przedstawia model kmeans. Widzimy, że suma kwadratów reszt wyniosła 83.9%.



Obraz przedstawia dane zbioru South German Credit Data Set podzielone na cztery grupy. Grupowanie zostało przeprowadzone dla czterech grup, ponieważ atrybut decyzyjny zawiera cztery klasy.