2A. Porówanie metod klasyfikacji - pakiet R

Konrad Cendrowski

Jacek Giedronowicz

Spis treści

[Wstęp 2](#_Toc74083489)

[Krótki opis danych 2](#_Toc74083490)

[Iris 4](#_Toc74083491)

[Podsumowanie 6](#_Toc74083492)

[Adult 7](#_Toc74083493)

[Podsumowanie 10](#_Toc74083494)

[Dressify 11](#_Toc74083495)

[Podsumowanie 13](#_Toc74083496)

[Podsumowanie 13](#_Toc74083497)

[Źródła: 13](#_Toc74083498)

# Wstęp

Przedstawimy wyniki porównania czterech metod klasyfikacji:

* Decision Trees
* K-Nearest Neighbours
* Support Vector Machine
* Random Forest

Modele zostały zbudowane na danych z ogólnodostępnych baz tj:

* iris
* imports-85
* adults

# Krótki opis danych

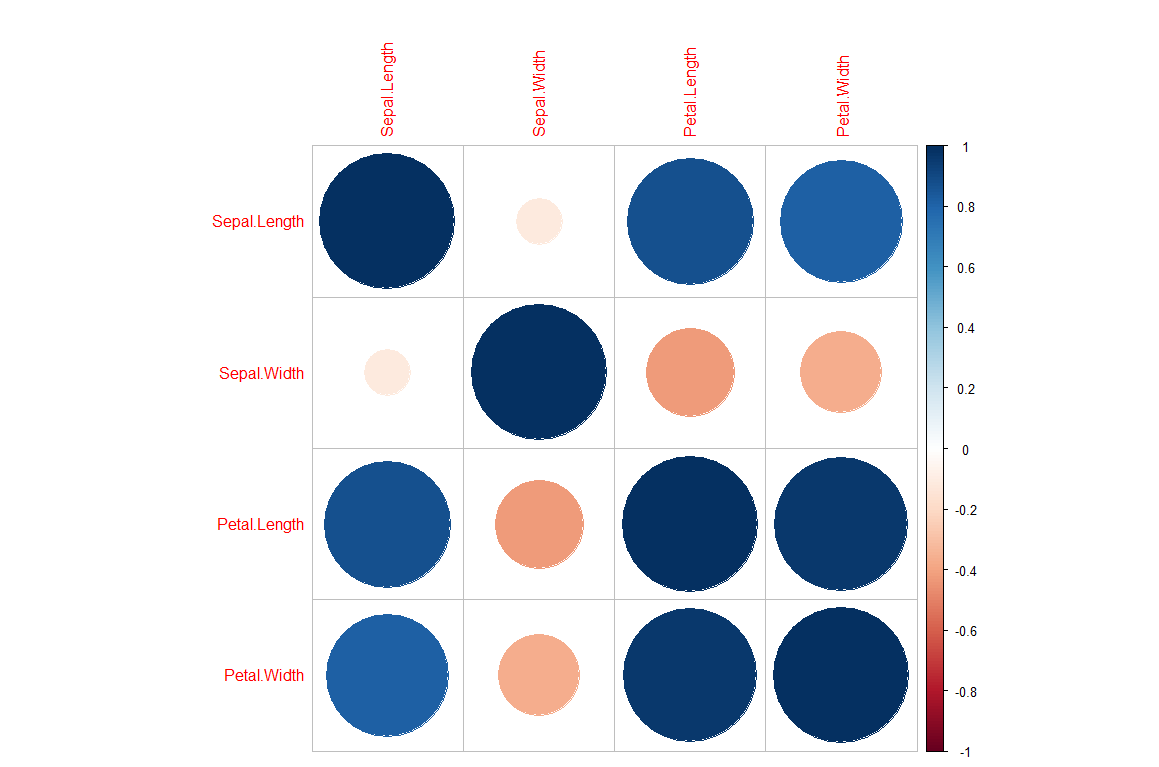
* iris – dane opisują pewne cechy kwiatów irysów tj:
  + Sepal.Length – długość działki kielicha przy kwiecie liczone w centymetrach,
  + Sepal.Width – szerokość działki kielicha przy kwiecie liczone w centymetrach,
  + Petal.Length – długość płatków kwiatu liczone w centymetrach,
  + Petal.Width – szerokość płatków kwiatu liczone w centymetrach,
  + Species – nazwa odmiany.
* adult – dane opisują pewne cechy dorosłych
  + age - wiek
  + workclass – klasa robocza
    - Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked
  + fnlwgt: continuous.
  + education – rodzaj wykształcenia
  + education-num: continuous.
  + marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
  + occupation - zawód
  + relationship - posiadane związki (Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried)
  + race - rasa
  + sex - płeć
  + capital-gain / capital-loss – zysk / strata kapitałowa
  + hours-per-week – przepracowane godziny tygodniowo
  + native-country – pochodzenie
  + income – zarobki (progowo)
* dressify – dane opisujące pewne cechy ubrań
  + Style – styl ubrania
    - Calual, work, Novelty, bohemian, party, Sexy, cute, Brief, vintage,
  + Price – cena ubrania
  + Rating – ocena, wartości od 0 do 5
  + Size – rozmiar
    - Free, S, M, L, XL
  + Season – rekomendowany sezon używania danego ubrania
    - Winter, Summer, Spring, Automn
  + NeckLine – typ dekoltu
    - V-neck, o-neck, beat-neck, Sweeatheart, turndowncollor, peterpan-collor, slash-neck, bowneck
  + SleeveLength – długość rękawów
    - Sleeveless, short, full, halfsleeve, thressqatar, capsleeves, Petal, butterfly
  + Waistline – talia
    - Empire, natural
  + Material - materiał
    - Cotton, polyester, chiffonfabric, microfiber, spandex, silk, rayon, lycra, milksilk, cashmere, knitting, acrylic, mix, sill, nylon
  + FabricType – tkanina
    - Chiffon, broadcloth, organza, tulle, worsted, other, jersey, poplin, knitting, terry
  + Decoration – dekoracje
    - Hollowout, lace, applique, bow, sashes, beading, ruched, ruffles, pockets, button, cascading, feathers, Tiered, embroidary, sequined, rivet, draped, pleat
  + PatternType – wzór
    - Solid, print, patchwork, character, animal, striped, geometric, leopard, floral
  + Area
    - A, B, C, D, E
  + Recommended – rekmendacja wartość 0 lub 1

# Iris

W zbiorze danych ‘Iris’ przewidujemy zmienną ‘Species’ czyli odmiane irysów.

Znormalizowane dane dzielimy na zbiory treningowy i testowy w stosunku 2:1.

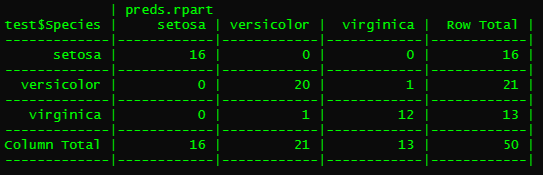
Wykres korelacji:



**Decision Trees:**

Czas wykonania: 0.03397989 secs

Dokładność predykcji: 96%



**k-Nearest Neighbour:**

Czas wykonania: 0.00800705 secs

Dokładność predykcji: 94% - 96%

Aby wyznaczyć klasyfikacje knn o największej dokładności trzeba dobrać odpowiedni parametr k do zbioru danych. W tym celu wykonujemy predykcje 50 razy dla różnych k i zapisujemy ich dokładność. Wyniki możemy przedstawić na wykresie:



W tym przypadku najmniejsze k dający największą dokładność jest k = 7.

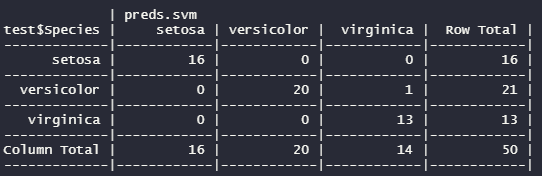
Porównanie wyniku algorytmu z rzeczywistymi danymi zmiennej celu możemy przedstawić w tabelce i obliczyć na jej podstawie dokładność jako suma na przekątnej podzielona przez liczba rekordów, razy 100 aby uzyskać wynik procentowy.



**Support Vector Machine:**

Czas wykonania: 0.01498914 secs

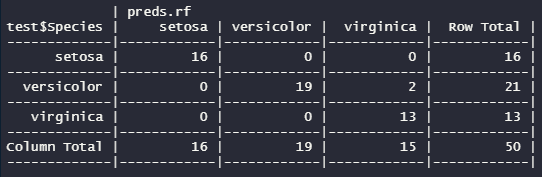
Dokładność predykcji: 98%



**Random Forest:**

Czas wykonania: 0.03198099 secs

Dokładność predykcji: 96%



### Podsumowanie

Pod względem dokładności predykcji, najlepiej wypadł algorytm Support Vector Machine, natomiast najszybszy okazał się algorytm k-Nearest Neighbours.

# Adult

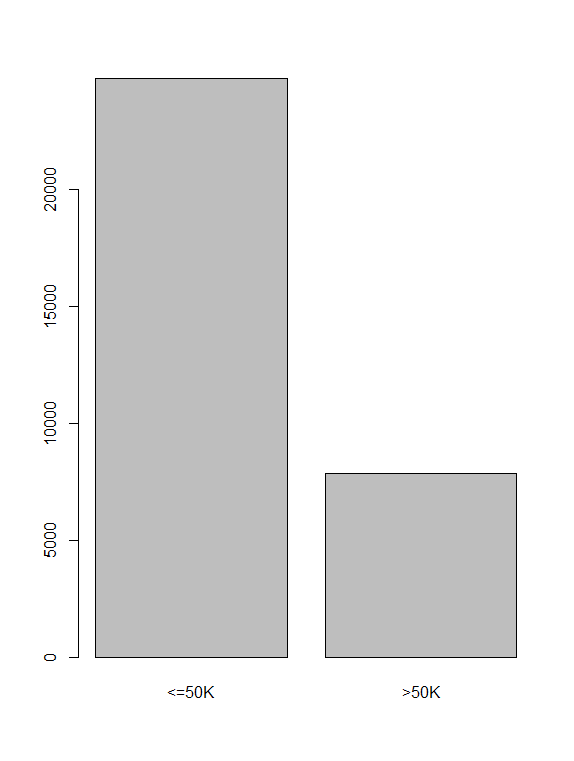
Baza ‘adult’ jest największą z opracowywanych przez nas bas. Zawiera 32 561 rekordów i 15 kolumn co sprawia, że budowanie modeli jest znaczenie dłuższe niż w przypadku innych baz.

Dane po czyszczeniu i łączeniu poziomów podzieliliśmy na zbiory treningowe 70% i testowe 30%. Przewidywana zmienna to ‘income’.

Wykres korelacji:



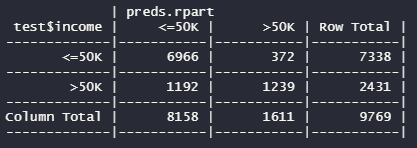
Zmienna income:



**Decision Trees:**

Czas wykonania: 0.8444691 secs

Dokładność predykcji: 84%



**k-Nearest Neighbour:**

Czas wykonania: 121.8649 secs / 2.253829 secs

Dokładność predykcji: 80.27%

Aby wyznaczyć klasyfikacje knn o największej dokładności trzeba dobrać odpowiedni parametr k do zbioru danych. W tym celu wykonujemy predykcje 50 razy dla różnych k i zapisujemy ich dokładność. Wyniki możemy przedstawić na wykresie:



W tym przypadku najmniejsze k dający największą dokładność jest k = 20.

Porównanie wyniku algorytmu z rzeczywistymi danymi zmiennej celu możemy przedstawić w tabelce i obliczyć na jej podstawie dokładność jako suma na przekątnej podzielona przez liczba rekordów, razy 100 aby uzyskać wynik procentowy.

Obraz zawierający tekst

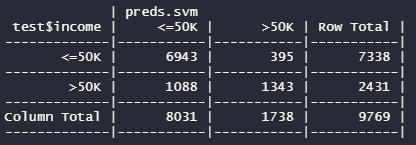
Opis wygenerowany automatycznie

Z powodu bardzo dużej liczby danych algorytm bardzo długo szukał najlepszego parametru k – 121 sekund. Obliczenie pojedynczej iteracji algorytmu knn z parametrem k = 20 zajęło 2.253829 sekundy.

**Support Vector Machine:**

Czas wykonania: 1.358648 mins

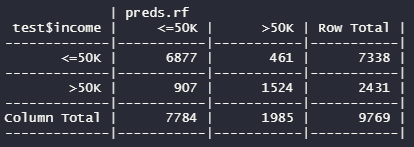
Dokładność predykcji: 85%



**Random Forest:**

Czas wykonania: 21.98921 secs

Dokładność predykcji: 85%



### Podsumowanie

W przypadku zbioru adult najdokładniejsze okazały się algorytmy SVM oraz Random Forest, natomiast pod względem szybkości predykcji najlepszy okazał się algorytm Decision Trees.

# Dressify

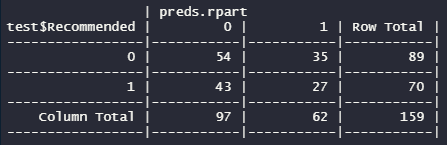
Przewidywaną zmienną zbioru ‘dressify’ jest ‘Recommended’. Po czyszczeniu i łączeniu poziomów wynikających z licznych pomyłek przy nazwach kolumn, podzieliliśmy zbiór na treningowy (200 rekordów) i testowy (159 rekordów).



**Decision Trees:**

Czas wykonania: 0.03097987 secs

Dokładność predykcji: 51%



**k-Nearest Neighbour:**

Czas wykonania: 0.1698921

Dokładność predykcji: 100%

Aby wyznaczyć klasyfikacje knn o największej dokładności trzeba dobrać odpowiedni parametr k do zbioru danych. W tym celu wykonujemy predykcje 50 razy dla różnych k i zapisujemy ich dokładność. Wyniki możemy przedstawić na wykresie:



W tym przypadku najmniejsze k dający największą dokładność jest k = 4.

Porównanie wyniku algorytmu z rzeczywistymi danymi zmiennej celu możemy przedstawić w tabelce i obliczyć na jej podstawie dokładność jako suma na przekątnej podzielona przez liczba rekordów, razy 100 aby uzyskać wynik procentowy.

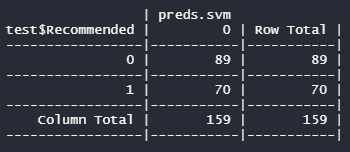
Obraz zawierający stół

Opis wygenerowany automatycznie

**Support Vector Machine:**

Czas wykonania: 0.06195903 secs

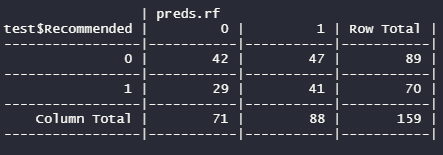
Dokładność predykcji: 56%



**Random Forest:**

Czas wykonania: 0.116925 secs

Dokładność predykcji: 52%



### Podsumowanie

W przypadku zbioru dressify największą dokładnością odznaczył się algorytm kNN natomiast najszybciej wykonał się algorytm Decision Trees.

# Podsumowanie

Na podstawie przeprowadzonych eksperymentów, można stwierdzić, że najlepszą dokładnością predykcji wyróżnia się algorytm Support Vector Machine, natomiast najszybciej wykonuje się algorytm Decision Trees. Przed wyborem algorytmu należy zwrócić uwagę na rozmiar bazy danych. Przy zbiorach takich jak ‘adult’ algorytm DecisionTree okazał się znacznie szybszy oraz jedynie nieznacznie mniej dokładny.

# Źródła:

<https://rpubs.com/>