POLITECHNIKA WARSZAWSKA

WYDZIAŁ
MECHANICZNY ENERGETYKI I LOTNICTWA

ZAKŁAD **Silników Lotniczych**

PRACA PRZEJŚCIOWA INŻYNIERSKA

Piotr Walędzik

Porównanie metod analizy danych z użyciem pakietu R

Nr albumu: 259692

Opiekun: dr inż. Mateusz Żbikowski

Warszawa, czerwiec 2016

SPIS TREŚCI:

1. Cel pracy.

Celem pracy jest przedstawienie różnych sposobów analizy danych oraz sposobów ich efektywnego wykorzystania. W dzisiejszych czasach takie pojęcia jak Statystyka, Big Data, czy Data Scientist stają się coraz popularniejsze i technologie z nimi związane wkraczają w piorunującym tempie w nasze życie.

Termin Statystyka jako nauka, której przedmiotem zainteresowania są metody pozyskiwania i prezentacji, a przede wszystkim analizy danych opisujących zjawiska, w tym najczęściej masowe.

Big Data jako określenie dużych, zmiennych i różnorodnych zbiorów danych, których przetwarzanie i analiza jest trudna, ale jednocześnie wartościowa, ponieważ może prowadzić do zdobycia nowej wiedzy. Pojęcie dużego zbioru danych oznacza sytuację, kiedy danych nie da się przetwarzać przy użyciu metod trywialnych i powszechnie dostępnych metod.

Pojęcie Data Scientist jako termin określający profesję osoby zajmującej się analizą danych, w ostatnim czasie otrzymał miano najseksowniejszego zawodu XXI wieku.

Właśnie dlatego uważam, że temat którym zajmę się w poniższej pracy jest czymś rozwojowym i pozwoli mi zagłębić się w podstawy analizy danych.

Dzięki tej pracy, czytelnik będzie wiedział, do jakich danych można wykorzystać poszczególne metody oraz w jaki sposób dobierać sposób ich analizy.

2. *Informacje ogólne*

Praca będzie opierała się na języku R, czyli interpretowanym języku programowania do obliczeń statystycznych i wizualizacji wyników oraz RStudio, czyli darmowym środowisku open-source dedykowanym dla języka R.

- Co to jest R?
  R Project for Statistical Computing (w skrócie po prostu R) jest potężnym narzędziem analizy danych. To jednocześnie język programowania oraz środowisko obliczeniowe i graficzne.

  R jest darmowym produktem stworzonym na zasadzie otwartego oprogramowania na licencji GNU General Public License. Działa w systemach operacyjnych Mac, Windows i Unix.

- Co to jest RStudio?
  Jak możemy dowiedzieć się ze strony producenta, misją twórców RStudio jest chęć stania się najczęściej używanym oprogramowaniem Open Source dedykowanym dla języka R.

Dlaczego powinienem używać środowiska R?

Środowisko R pozwala kontrolować swoje dane oraz uzyskiwać z nich jak najwięcej informacji. R umożliwia projektowanie i uruchamianie złożonych analiz dopasowanych do potrzeb użytkownika, co nie jest możliwe w przypadku innych pakietów oprogramowania. Dzięki temu, środowisko R odpowiada potrzebom dużej grupy potencjalnych użytkowników.

3. *Model jednokrotnej regresji liniowej*

Model jednokrotnej regresji liniowej wygląda następująco:

$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i,\ i = 1, 2, \ldots, n$, gdzie ε_i to niezależne zmienne losowe o rozkładzie $\mathcal{N}(0, \sigma)$, natomiast β_0, β_1 i $\sigma > 0$ to pewne stałe rzeczywiste (tzw. parametry modelu).

Regresję liniową w R wykonuje się przy pomocy funkcji :

lm (zmienna.objaśniana ~ zmienna.objaśniająca + zmienna objaśniająca2)

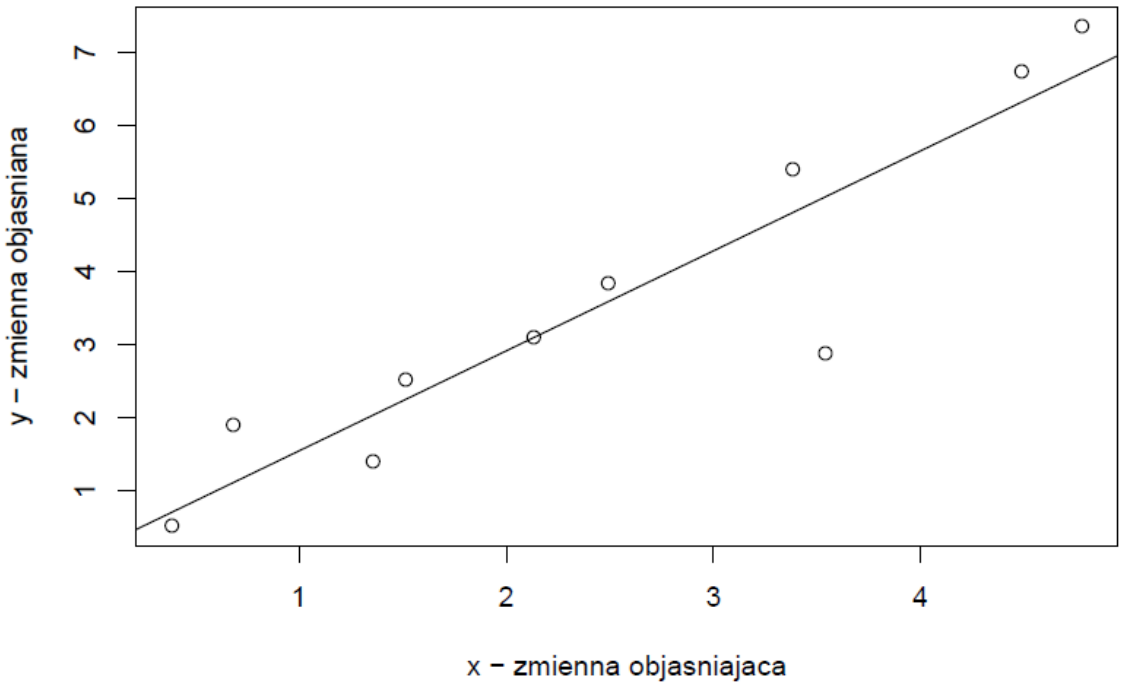
Na podstawie przykładu: $\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2}$

Zastosowanie:

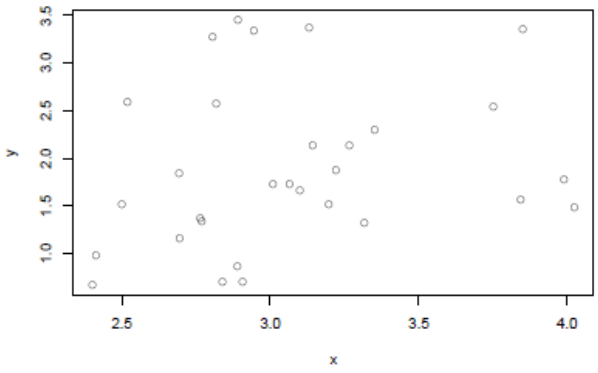
Trzeba zauważyć, że zależność wartości średniej zmiennej objaśnianej od zmiennej objaśniającej jest zależnością liniową tj. :

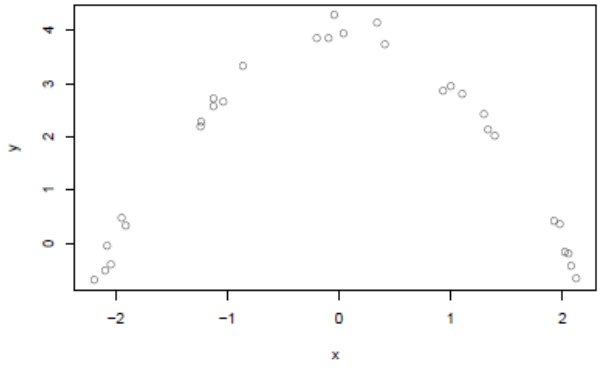
$E(Y_i) = \beta_0 + \beta_1 x_i$, *i=1,2,…,n*

Oznacza to zatem dopasowywanie modelu regresji liniowej do konkretnych danych, takich, gdy wykres rozproszenia, sporządzony dla tych danych, ma choć w przybliżeniu charakter liniowy. Przykład:

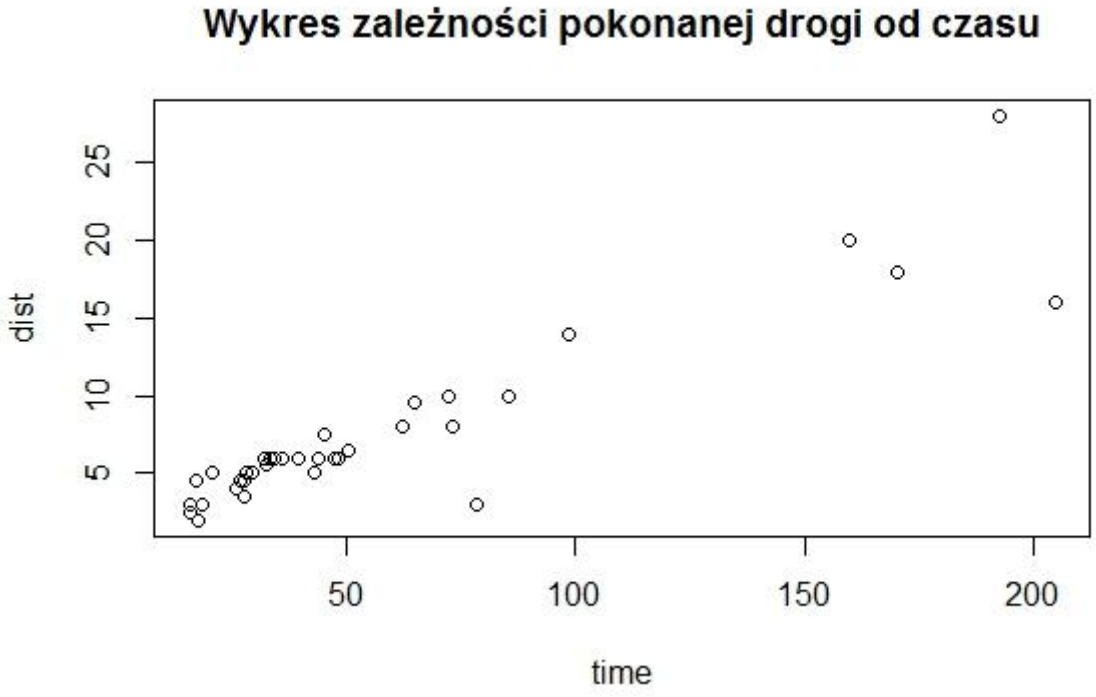


W przypadku, kiedy dysponujemy punktami rozproszonymi, nie wykazującymi tendencji liniowej, to dopasowanie prostej regresji mija się z celem. Sytuację taką można zobrazować na poniższych dwóch rysunkach:





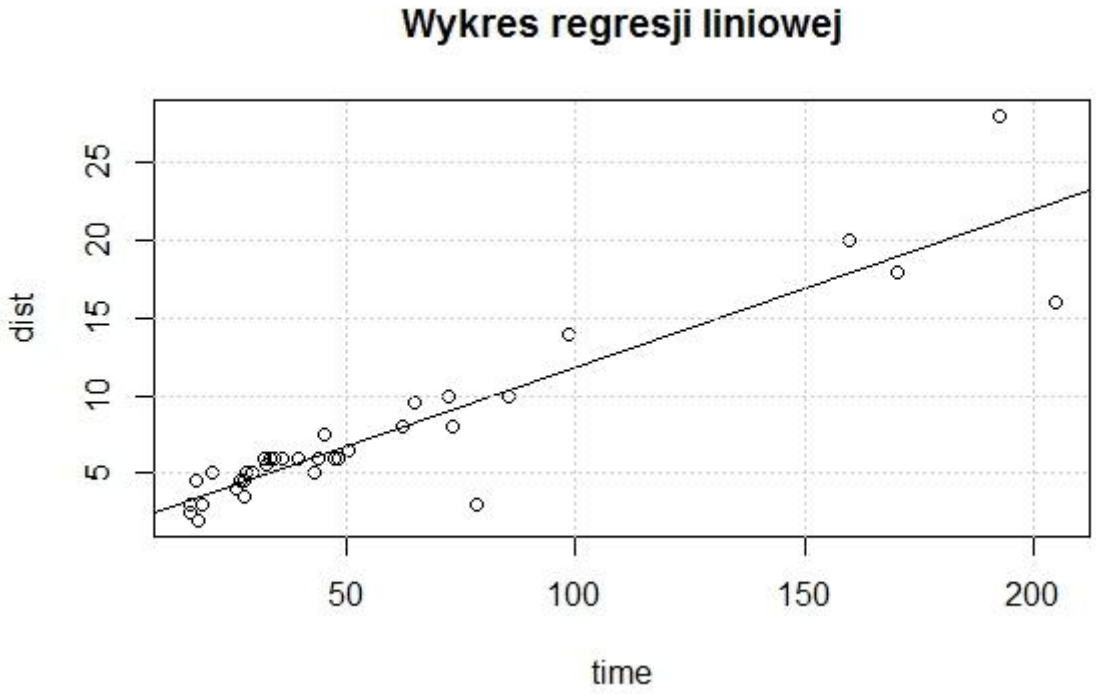
Na początku chciałbym przybliżyć najprostsze wykorzystanie modelu regresji liniowej na podstawie wbudowanego prostego zestawu danych „hills”, który przyjmuje podane zmienne dist(dystans w milach) zależną od time(czas na pokonanie góry w minutach). Jako zmienną objaśnianą przyjmę dist(dystans w milach), a jako zmienną objaśniającą przyjmę time(czas na pokonanie góry w minutach).



Te kilka linijek kodu pozwala nam zdefiniować funkcję regresji dla naszego zestawu danych Hills, a wynik prezentowany jest poniżej:

```r
##############################################
#           Model Regresji Liniowej          #
##############################################

#zaczytanie danych (tutaj skorzystam z wbudowanego zestawu "hills")

attach(hills)

#konstruujemy prosty model regresji liniowej
model.regresji = lm(dist ~ time, data = hills)

#dokladniejsze wypisanie uzyskanych wynikow
model.opis = summary(model.regresji)

#Wyraz wolny Bo
coef(model.regresji)[1]

#Współczynnik kierunkowy B1
coef(model.regresji)[2]

#Wartości prognozowane: Yi
fitted(model.regresji)

#typowy wykres regresji || jedna zmienna objasniająca
plot(time, dist, main="Wykres regresji liniowej")
abline(model.regresji)
grid()
```

16:24 (Top Level) R Script



Równanie funkcji regresji liniowej:

Y = 1.65347 + 0.1015124x

(dist = 1.65347 + 0.1015124 time)

Przydatne funkcje w pakiecie R:

model.regresji = lm(dist ~ time, data = hills)

1. Wyraz wolny $\hat{\beta}_0$: coef(model.regresji)[1]
2. Współczynnik kierunkowy $\hat{\beta}_1$: coef(model.regresji)[2]
3. Wartości prognozowane $\widehat{Y}_i$: fitted(model.regresji)
4. Residua (wartości resztowe) $e_i = Y_i - \hat{Y}_i$: residuals(model.regresji)

4. *Model drzewa decyzyjnego*

Co to jest drzewo decyzyjne?

Drzewo decyzyjne jest strukturą drzewiastą, w której:

- Węzły wewnętrzne posiadają testy na wartościach atrybutów
- Z każdego węzła wewnętrznego wychodzi tyle gałęzi, ile jest możliwych wyników testu w tym węźle
- Liście zawierają podjętą decyzję o klasyfikacji obiektu

Jak ocenia się jakość drzewa?

Jakość drzewa oceniana jest poprzez:

- Rozmiar: im drzewo jest mniejsze, tym lepsze
    1. Mała liczba węzłów
    2. Mała wysokość
    3. Mała liczba liści
- Dokładnością klasyfikacji na zbiorze testowym
- Dokładnością klasyfikacji na zbiorze treningowym

```
1  ##############################################
2  #           Model Drzewa Decyzyjnego         #
3  ##############################################
4
5  library("party")
6
7  #skorzystam z dobrze znanego data setu do nauki
8  #a mianowicie z danych dotyczących irysów
9  #jest to data frame ze 150 przypadkami i 5 zmiennymi
10
11 #zaczytanie danych
12 irysy = as.data.frame(iris)
13
14 #podstawowe informacje o danych
15 str(iris)
16 #Sepal length and width = długość i szerokość kielicha
17 #Petal length and width = długość i szerokość płatków
18 #Species = gatunek
19
20
21 #Zamieniam nazwy kolumn na polskie
22 library(plyr)
23 irysy = rename(irysy, c("Sepal.Length"="Kielich.Długość", "Sepal.Width"="Kielich.Szerokość",
24                 "Petal.Length"="Płatek.Długość", "Petal.Width"="Płatek.Szerokość",
25                 "Species"="Gatunek"))
26
27 #Dzielę dane na treningowe i testowe w skali 70% i 30%
28 set.seed(1234)
29 temp = sample(2, nrow(irysy), replace=TRUE, prob=c(0.7,0.3))
30 trainData = irysy[temp==1,]
31 testData = irysy[temp==2,]
```

```r

#drzewo decyzyjne, przewidywanie gatunku sądząc po wymiarach
irysy.drzewo1 = ctree(Gatunek ~ Kielich.Długość + Kielich.Szerokość +
                      Płatek.Długość + Płatek.Szerokość, data = trainData)
#Otrzymane drzewo decyzyjne:
plot(irysy.drzewo1)
#barplot, wykres dla każdego liścia węzła ukazuje prawdopodobieństwo
#danego przypadku, ze względu na jeden z trzech gatunków

plot(irysy.drzewo1, type="simple")
#tutaj na przykład node2 mówi że mamy 40 przypadków, z czego
#wszystkie należą do pierwszego gatunku czyli Setosa

#Przewidywania na danych treningowych
przewidywaniaTrain = predict(irysy.drzewo1)
przewidywania1 = table(przewidywaniaTrain,trainData$Gatunek)

#Przewidywania na danych testowych
przewidywaniaTest = predict(irysy.drzewo1, newdata = testData)
przewidywania2 = table(przewidywaniaTest, testData$Gatunek)
```

13:1 (Top Level) R Script

W powyższym przykładzie widzimy jak w prosty sposób możemy zbudować podstawowy model drzewa decyzyjnego.

Na początku zaczytuję dane oraz odczytuję informację o zbiorze danych, na których mam zamiar pracować.

Jak widać dowiedziałem się, co opisują kolumny oraz dzięki funkcji Rename zmieniłem ich nazwy na polskie.

Po czym przeszedłem do podziału danych na treningowe (70%) i testowe (30%).

- **Dane treningowe:**
  Zwykle około 70%. Są to dane, na podstawie których klasyfikator „uczy się” poprawnej klasyfikacji.
- **Dane testowe:**
  Zwykle około 30%. Są to dane, na podstawie których sprawdzana jest jakość generalizacji badanego klasyfikatora, tzn. jak dobrze, klasyfikator „nauczony” na zbiorze treningowym, radzi sobie z klasyfikacją danych ze zbioru testowego.

Oczywiście, w celu oceny jakości klasyfikacji zbioru testowego, konieczna jest znajomość prawdziwej przynależności jego elementów do klas i porównanie jej z przyporządkowaniem elementów do klas zaproponowanych przez klasyfikator.

Idealną sytuacją byłoby, gdyby istniała możliwość oceny zbiorów, treningowego i testowego, pochodzących z różnych badań i zawierających odpowiednio duże ilości danych. Niestety jednak, zazwyczaj, z powodu różnych trudności praktycznych, do dyspozycji jest tylko jeden zbiór danych. Aby uporać się z tym problemem, stosuje się kilka metod podziału zbioru na treningowy i testowy.

1. **metoda resubstytucji** - uczenie klasyfikatora następuje z wykorzystaniem całego dostępnego zbioru danych, jako zbiór testowy występuje kolejno każdy z obiektów badanego zbioru danych. Zaletą tej metody jest wykorzystanie, w procesie uczenia klasyfikatora, całej dostępnej informacji (cały dostępny zbiór danych jest wykorzystany w etapie uczenia), jednak użycie zbioru danych najpierw do nauczenia klasyfikatora a potem do jego przetestowania, może dawać "nieobiektywne" oceny co do jego skuteczności

2. **metoda leave-one-out** - metoda podobna do resubstytucji, z tym tylko wyjątkiem, że zbiór treningowy zawiera wszystkie elementy, oprócz jednego, który w tym czasie tworzy (jednoelementowy) zbiór treningowy

3. **losowy podział na zbiory treningowy i testowy** - zbiór danych jest losowo dzielony na część treningową i testową. Uczenie klasyfikatora przebiega z wykorzystaniem zbioru treningowego, natomiast w celu sprawdzania jakości generalizacji klasyfikatora wykorzystywana jest testowa część danych. Metoda ta wydaje się być najbardziej zbliżona do rzeczywistości. Wadą w tym wypadku jest ograniczenie ilości danych treningowych, co w przypadku, gdy dyspozycji jest niewielka liczba próbek, prowadzi do pogorszenia skuteczności klasyfikatora

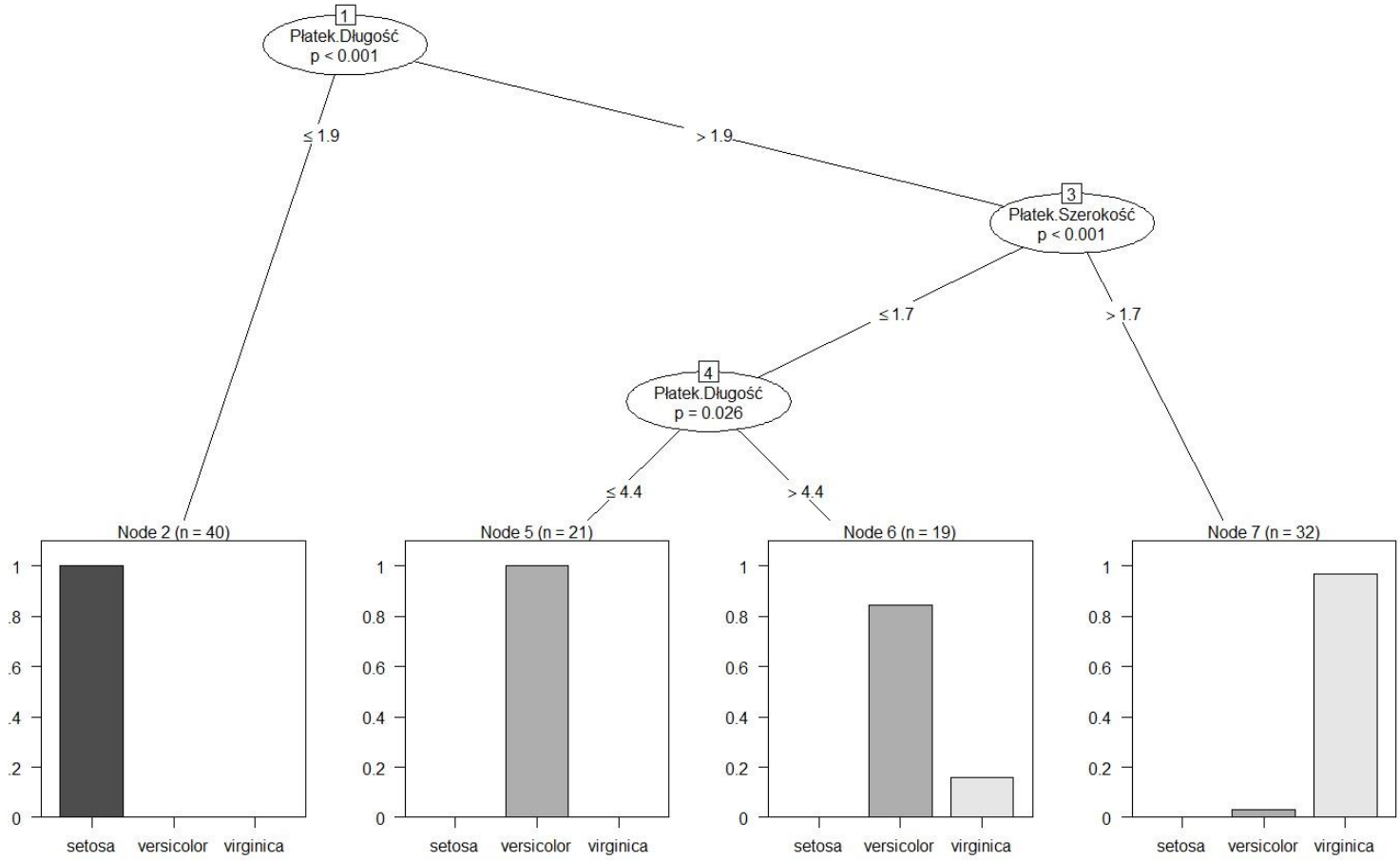
W moim przypadku skorzystałem z losowego podziału na zbiór treningowy i testowy.

Predykcja na danych treningowych przedstawia się następująco:

```
przewidywaniaTrain setosa versicolor virginica
         setosa         40          0         0
         versicolor      0         37         3
         virginica       0          1        31
>
```

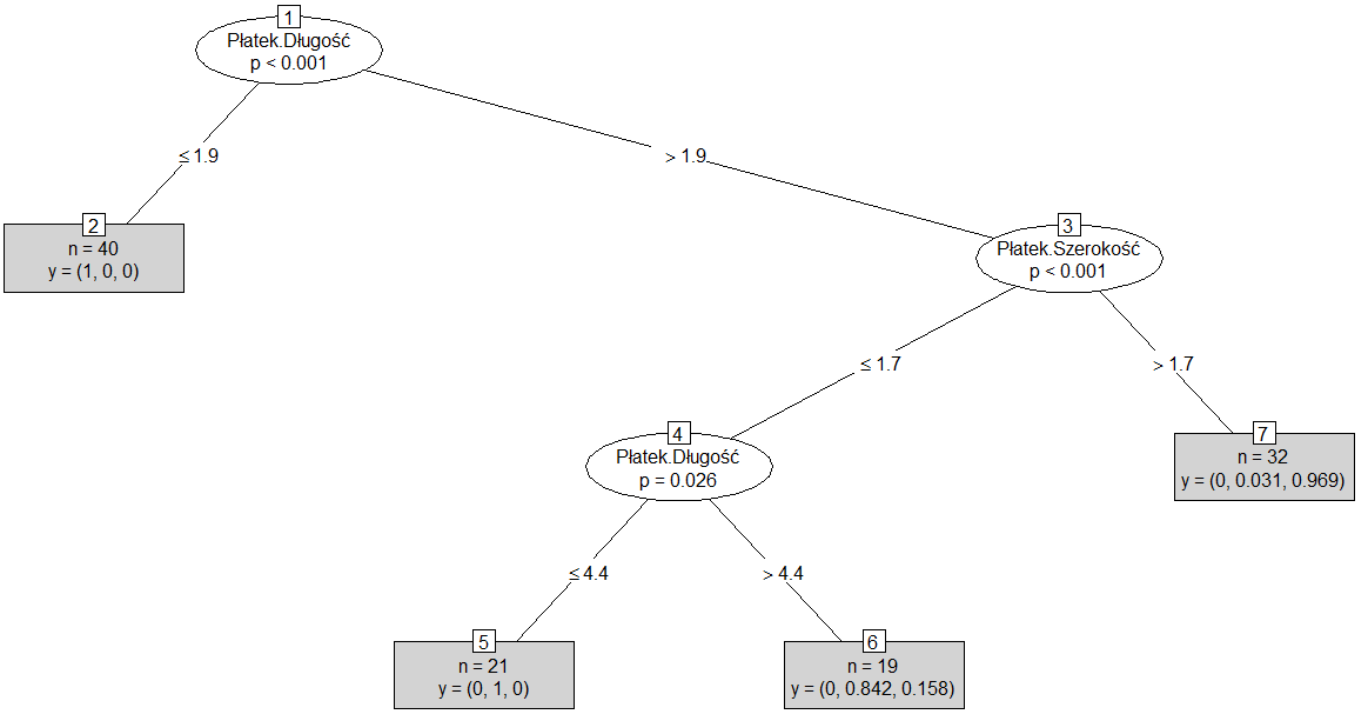
Zobrazowanie drzewa, przy pomocy funkcji:

plot(irysy.drzewo1)



Zobrazowanie drzewa, przy pomocy funkcji:

plot(irysy.drzewo1, type="simple")



Z powyższych rzeczy wynika, że w pierwszym z zaprezentowanych grafów, dla każdego węzła liścia ukazuje on prawdopodobieństwo danego przypadku, ze względu na jeden z trzech gatunków.

Na grafie drugim, ta sama cecha jest pokazana jako „y” w węzłach liścia.

Na przykład:

Węzeł nr 2, posiadający n=40, y=(1,0,0), oznacza, że zawiera 40 jednostek ze zbioru treningowego i wszystkie z nich przynależą do pierwszego gatunku, czyli „Setosa”.

Węzeł nr 5, posiadający n=21, y=(0,1,0), oznacza, że zawiera 21 jednostek ze zbioru treningowego i wszystkie z nich przynależą do pierwszego gatunku, czyli „Versicolor”.

Węzeł nr 6, posiadający n=19, y=(0,0.842,0.158), oznacza, że zawiera 19 jednostek ze zbioru treningowego, z czego 16 przynależy do „Versicolor” oraz 3 przynależą do „Virginica”.

Węzeł nr 7, posiadający n=32, y=(0,0.031,0.969), oznacza, że zawiera 32 jednostki ze zbioru treningowego, z czego 1 przynależy do zbioru „Versicolor” oraz 31 przynależą do „Virginica”.

Poniżej prezentuję zestawienie predykcji danego drzewa na danych testowych:

przewidywaniaTest	setosa	versicolor	virginica
setosa	10	0	0
versicolor	0	12	2
virginica	0	0	14

>

Interpretacja jak wyżej.

Jak widać, mechanizm drzewa decyzyjnego jest ważnym narzędziem w uczeniu maszynowym i eksploracji danych. Drzewa decyzyjne są wykorzystywane do rozwiązywania problemu klasyfikacji.

5. *Model Random Forest – Las Losowy*

Lasy losowe (Random Forests) są rozszerzeniem modułu poświęconego drzewom klasyfikacyjnym. Znajomość tego ostatniego jest potrzebna do zrozumienia modułu nt. lasów. Lasy losowe wyróżniają się wśród metod klasyfikacji dużą mocą predykcyjną oraz wydajnością estymacji i zastosowania nawet na wielkich zbiorach danych z tysiącami zmiennych.

Podobnie jak drzewa klasyfikacyjne, lasy losowe mają dwa główne zastosowania: predykcję i deskrypcję. Do ich innych funkcji zalicza się m.in. analiza istotności zmiennych.

Ważną zaletą lasów losowych jest to, że w trakcie budowy modelu estymator błędu klasyfikacji jest automatycznie otrzymywany jako produkt uboczny algorytmu wyboru prób uczących dla każdego drzewa.

Lasy losowe zapewniają również rezultaty bardziej "wygładzone" niż te otrzymywane przy użyciu pojedynczego drzewa klasyfikacyjnego. Różnica jest widoczna szczególnie na dużych zbiorach danych. Jest to spowodowane tym, że w przypadku lasów losowych końcowa decyzja jest uśrednieniem wielu decyzji cząstkowych pochodzących z różnych, wchodzących w skład lasu, drzew klasyfikacyjnych.

Podobnie jak drzewa klasyfikacyjne, lasy losowe nie mają specyficznych wymagań odnośnie danych.

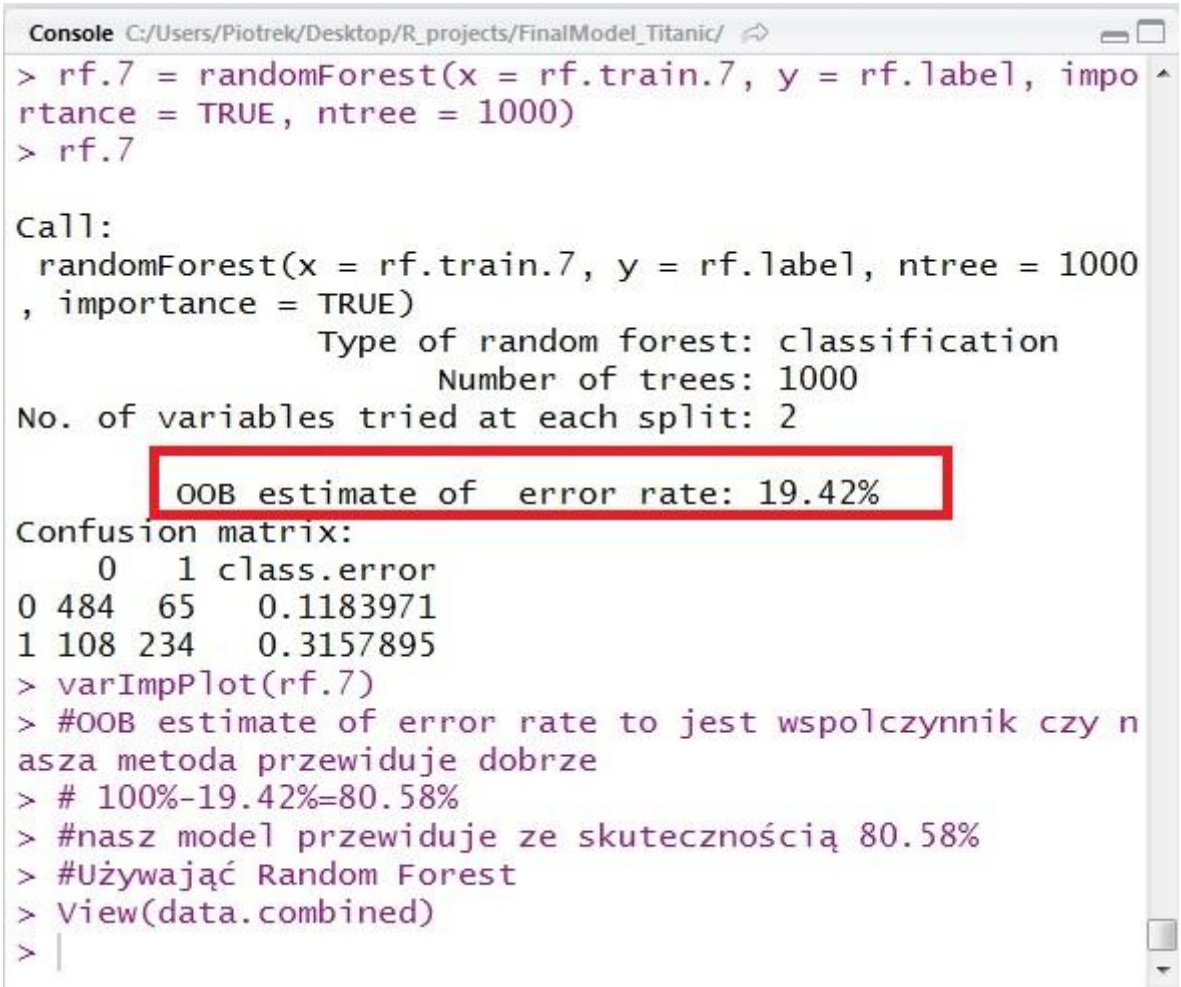
Wadą klasyfikacji za pomocą lasów losowych jest jej względna nieprzejrzystość. Las losowy w pewnym stopniu przypomina czarną skrzynkę: użytkownik nie ma wglądu w przebieg procesu decyzyjnego, a struktura modelu jest trudna do zinterpretowania z powodu jego rozmiaru i skomplikowania. Ponieważ końcowa decyzja jest średnią z wielu niezależnych decyzji cząstkowych, zazwyczaj nie ma możliwości wyjaśnienia, dlaczego decyzja modelu jest właśnie taka.

Finalnym i najbardziej obszernym modelem jaki stworzyłem, jest model predykcyjny dotyczący przewidywania kto przetrwa, a kto zginie w katastrofie statku RMS Titanic.

Kod, jako model końcowy i najbardziej rozbudowany, ze względów praktycznych został on opisany we wnętrzu poprzez komentarze.

Zarówno zbiór danych, jak i kod zostaje dołączony do pracy.

Stworzony model wytrenowany jest na zbiorze 891 jednostek treningowych i przewiduje ze skutecznością 100%-19.42%=80.58%, operując na danych pochodzących z kolumn Pclass, title, Parch, family.size, które oceniłem jako najbardziej wartościowe.



Cały proces tworzenia modelu jest skrupulatnie komentowany w kodzie.

6. *Podsumowanie*

W mojej pracy przedstawiłem wstęp do obszernej dziedziny, jaką jest analiza danych oraz podstawowe jej wykorzystanie za pomocą pakietu R.

Omówione metody oraz przedstawione przykłady są podstawami do rozpoczęcia swojej przygody na ścieżce Data Science.

Powyższa praca była dla mnie mobilizacją do rozwoju moich umiejętności związanych z językiem R oraz podstawowymi metodami analizy danych.

7. *Źródło informacji*

1. Przemysław Biecek, „*Analiza danych z programem R*”, Warszawa 2011,2013, Wydawnictwo Naukowe PWN SA

2. John M. Quick, „*Analiza statystyczna w środowisku R dla początkujących*”, 2012, Wydawnictwo HELION

3. Yanchang Zhao, „*R and Data Mining: Examples and Case Studies*”, Elsevier, 2012-2015, yanchang@rdatamining.com

4. UCI – Machine Learning Repository
   http://mlr.cs.umass.edu/ml/index.html

5. http://algolytics.pl/wp-content/uploads/docs/pl/bk01pt04ch32.html
   Rozdział 32. Las Losowy, Część IV.Moduły

6. WIKIPEDIA:
   https://pl.wikipedia.org/wiki/Data_scientist
   https://pl.wikipedia.org/wiki/Statystyka
   https://pl.wikipedia.org/wiki/Big_data
   https://pl.wikipedia.org/wiki/Analiza_danych

7. https://www.rstudio.com/about/

8. https://www.kaggle.com/

9. http://theta.edu.pl/wp-content/uploads/2011/02/regresja-liniowa.pdf