

# Metody Odkrywania Wiedzy

## Dokumentacja końcowa projektu

„Predykcja zużycia energii na podstawie danych czujnikowych”

Krzysztof Belewicz  
Paweł Pińczuk

26 stycznia 2020

# 1. Opis projektu

Celem projektu było wyznaczenie całkowitego zużycia energii dla zadanej chwili czasu, tzn. sumy poborów sprzętów AGD (kolumna *Appliances*) i oświetlenia (kolumna *lights*). Zbiór danych został pozyskany z archiwum dostępnego na stronie: <https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction>. Pojęciem docelowym jest wartość całkowitej pobieranej mocy przez gospodarstwo domowe. W ramach projektu zdecydowano się na oddzielne wykonania zadania regresji dla celu *Appliances* i celu *lights*, ze względu na hipotezę, że modele je wyznaczające mogą mieć inne właściwości.

Dokonano selekcji atrybutów za pomocą trzech algorytmów opisanych w rozdziale 3. Przeprowadzono procedurę oceny algorytmów liniowej regresji, drzew regresji oraz kawałkami liniowej regresji.

## 2. Opis danych

### 2.1. Charakterystyka danych

Dane wykorzystywane do eksperymentów zostały zebrane za pomocą sieci czujników w niewielkim domu w czasie 4.5 miesiąca. Składają się z:

- daty i godziny pomiaru,
- poboru energii sprzętów domowych [*Wh*],
- poboru energii oświetlenia [*Wh*],
- pomiarów temperatury i wilgotności dla 8 różnych pomieszczeń ( $^{\circ}\text{C}$ , [%]),
- pomiarów temperatury i wilgotności dla zewnętrznej, północnej strony budynku ( $^{\circ}\text{C}$ , [%]),
- danych z pobliskiej stacji pogodowej:
  - temperatura powietrza [ $^{\circ}\text{C}$ ],
  - temperatura punktu rosy [ $^{\circ}\text{C}$ ],
  - ciśnienie atmosferyczne [*mm Hg*],
  - wilgotność [%],
  - prędkość wiatru [*m/s*],
  - widoczność [*km*].

### 2.2. Przygotowanie danych

Każdy pomiar został uśredniony z 3 próbek wykonanych w równych odstępach co ok. 3,3 min. W ramach przygotowania danych, data i godzina pomiaru zostały rozdzielone na cztery oddzielne kolumny, zawierające miesiąc, dzień, godzinę i minutę pomiaru.

Liczba wszystkich obserwacji, zebranych w pliku *energydata\_complete.csv* wynosi 19735. Celem przyspieszenia obliczeń, algorytmy przedstawione w zadaniu zostały wykonane na danych zawierających 2000 pierwszych rekordów zmienna *testDataLength*. Wszystkie operacje dot. przygotowania danych są wykonane w funkcji *data\_org*.

### 3. Selekcja atrybutów

Aby zapobiec nadmiernemu dopasowaniu, stosuje się selekcję atrybutów, która wybiera kilka najważniejszych atrybutów do późniejszego stworzenia modeli. Po zastosowaniu selekcji, modele oparte o ograniczoną liczbę atrybutów zwykle są lepsze od opartych o wszystkie atrybuty. Istnieje wiele metod selekcji atrybutów; w ramach projektu zostało sprawdzone kilka metod (w nawiasach umieszczono opcję typu funkcji *feature\_selection*):

- prosty filtr statystyczny („*simple*”) - pomiędzy każdym z atrybutów a celem regresji stosuje się miarę statystyczną, która określa zależność celu od danego atrybutu (dalej „miara zależności”). Następnie wybiera się kilka atrybutów o największej „mierze zależności”. W ramach regresji pomiędzy atrybutami ciągłymi zastosowano współczynnik korelacji (Pearsona);
- bazująca na drzewach losowych („*rf*”) - w tym celu wykorzystano pakiet *randomForest* i jego wbudowaną opcję zwracającą parametr *IMPORTANCE* (bazujący na mierze MSE), o możliwości konfiguracji ilości drzew, w badaniu wykorzystano generowanie 500 drzew;
- metoda *RRELIEF* („*relief*”) - wersja algorytmu *RELIEF* do zastosowań w zadaniu regresji. Algorytm *RELIEF*, początkowo zaprojektowany dla zadania klasyfikacji binarnej, polega na losowym wybraniu obserwacji (jednego rekordu klasy+atrybuty). Następnie wyszukuje się  $k$  najbardziej podobnych obserwacji tej samej klasy, oraz  $k$  klasy przeciwnej. Dla każdego atrybutu oblicza się wagę istotności. Po wykonaniu  $K$  operacji, wykonuje się średnią wag istotności. Atrybuty segreguje się według wag istotności. W zadaniu regresji stosuje się inne funkcje obliczające wagę np. funkcję rozkładu. W projekcie  $k=3$ ,  $K=50$ .

W ramach projektu stosuje się następujące podejście: dla każdej wymienionej metody wykonuje się selekcję połowy atrybutów ( $part=0.5$ ) atrybutów, następnie wyznaczoną formułę aplikuje się do stworzenia modelu *rpart()*, i procedurze oceny (10-krotnej walidacji krzyżowej *model\_eval()*)<sup>1</sup>. Następnie największy współczynnik korelacji Pearsona wyznacza najlepszą metodę selekcji atrybutów oraz formułę do stworzenia modelu.

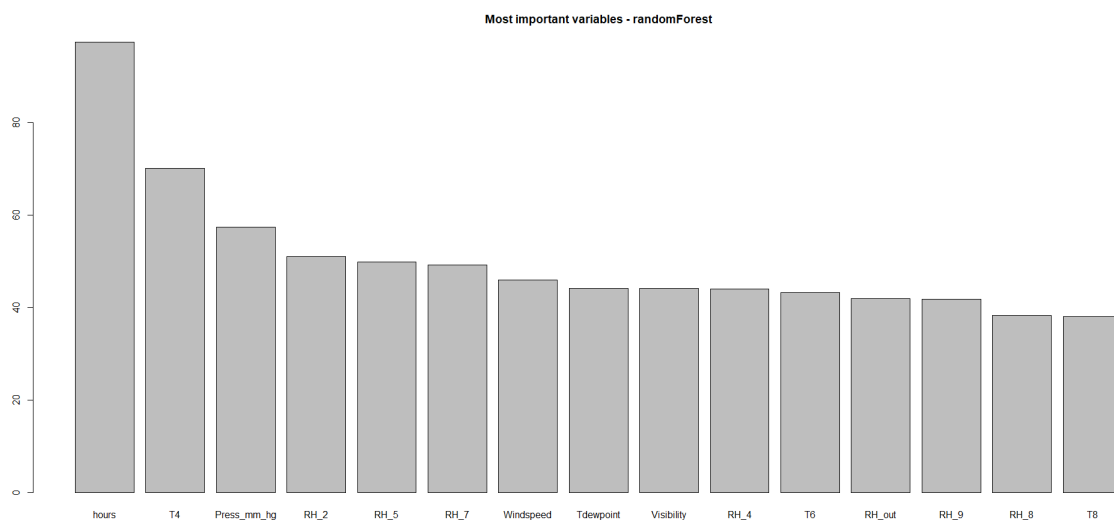
#### 3.1. Wyniki selekcji atrybutów

Na rysunkach 3.1, 3.2 i 3.3 przedstawiono wyniki każdej z selekcji. Dla *randomForest* i *simple.filter* atrybut *hours* dominuje nad pozostałymi. W tabeli 3.1 przedstawiono porównanie wyników każdej z selekcji. Wynika z tego, że atrybuty wyznaczone funkcją *simple.filter* pozwalają na najlepsze wyznaczenie modelu. Dla porównania przedstawiono też wynik walidacji krzyżowej dla modelu opartego o wszystkie atrybuty. Tylko selekcja atrybutów za pomocą *simple.filter* pozwala na poprawę dla obecnych warunków testowych (dostępne dane, ilość selekcionowanych argumentów, algorytm do walidacji krzyżowej). W związku z wynikami, atrybuty wyznaczono za pomocą prostego filtra statystycznego posłużą w dalszej konstrukcji modeli.

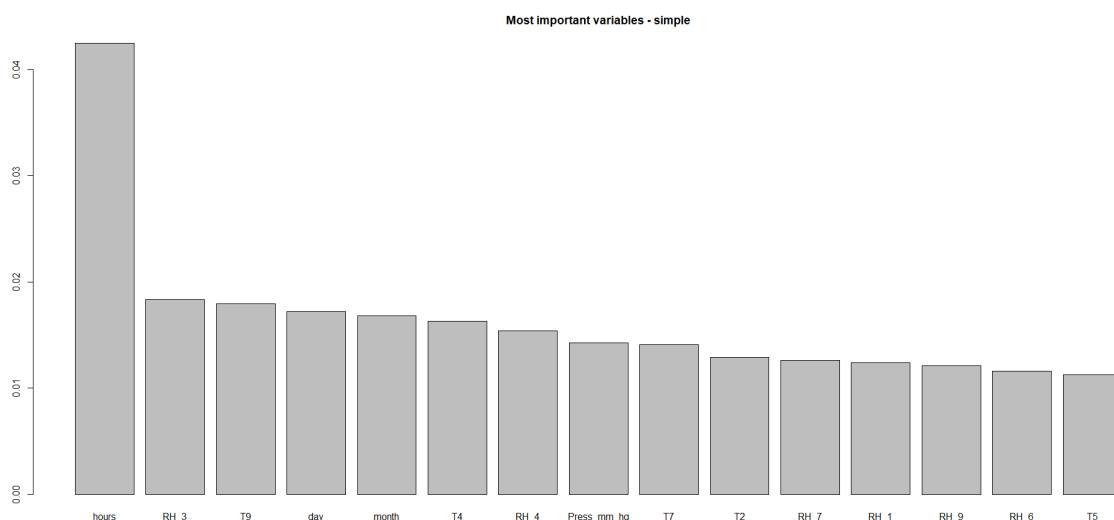
Tablica 3.1: Wyniki selekcji atrybutów — współczynniki korelacji

| Parametr   | <i>randomForest</i> | <i>simple</i> | <i>RELIEF</i> | bez selekcji |
|------------|---------------------|---------------|---------------|--------------|
| Appliances | 0,555               | 0,616         | 0,553         | 0,585        |
| lights     | 0,364               | 0,457         | 0,334         | 0,447        |

<sup>1</sup> Więcej nt. procedury walidacji krzyżowej w rozdziale 4.3



Rysunek 3.1: Wyniki selekcji z wykorzystaniem pakietu *randomForest*; cel: lights

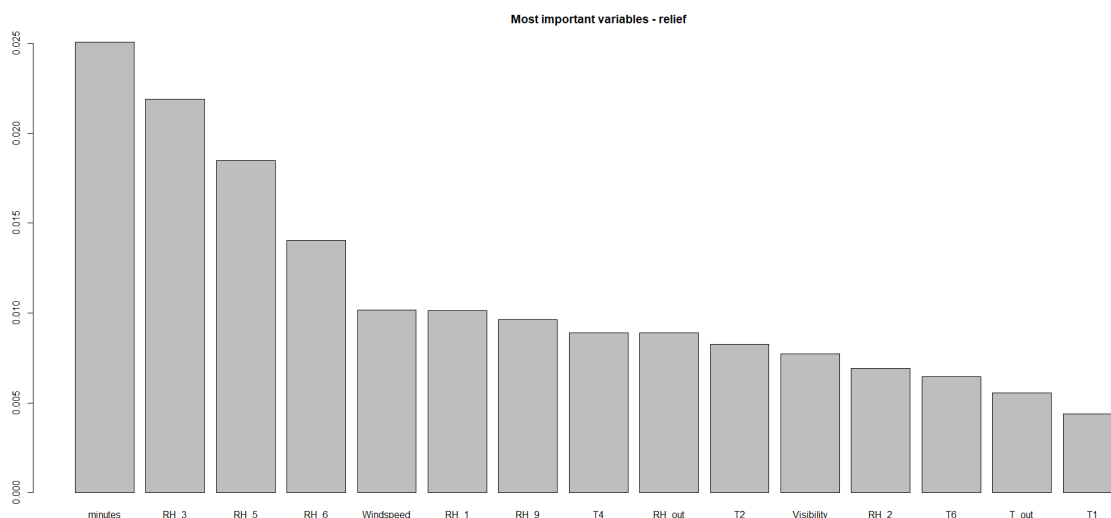


Rysunek 3.2: Wyniki selekcji z wykorzystaniem funkcji *simple.filter*; cel: lights

## 4. Konstrukcja i ocena modeli

### 4.1. Metody konstrukcji modeli

W pierwszym kroku dokonano modelowania za pomocą liniowej regresji. Realizuje ją algorytm *lm()*. Argumentami algorytmu są tylko: formuła (cel+atrybuty) oraz zestaw danych w formie *data.frame*. Algorytm na podstawie średniej lub mediany wyznacza współczynniki funkcji liniowej tj. współczynnik kierunkowy i wyraz wolny. Następnie od każdego atrybutu wyznacza wagę współczynnika kierunkowego oraz jeden wyraz wolny. Zaletą tego algorytmu jest jego prostota i szybkość



Rysunek 3.3: Wyniki selekcji z wykorzystaniem funkcji *rrelief.filter*; cel: lights

działania, niewątpliwą wadą jest fakt, że większość procesów zachodzących w świecie nie da się opisać za pomocą liniowych zależności.

Następnie dokonano modelowania za pomocą drzew regresji (algorytm *rpart()*). Funkcja buduje model rekurencyjnie dzieląc zbiór na mniejsze i wyznaczając dla nich średnią. Dla metody „anova” dedykowanej do zadania regresji kryterium podziału wyznaczone jest na podstawie sum kwadratów dla danego węzła i dla jego potomnych. Algorytm może przyjąć także argumenty na minimalną liczbę podziałów *minsplit* oraz maksymalną głębokość drzewa *maxdepth* (czyli długość pomiędzy korzeniem a liśćmi).

Pojedyncze modele drzew regresji zazwyczaj cierpią z powodu wysokiej wariancji – jedną z metod jej redukcji jest tzw. **Bagging (Bootstrap aggregating)**. W ramach tej metody tworzonych jest pewna liczba zbiorów "bootstrapowych". Dla każdego z tych zbiorów tworzy się nieprzycięte drzewo regresji. Następnie uśrednia się każdy z tych modeli, zmniejszając wariancję i redukując zbytnie dopasowanie. Bagging może zostać zrealizowany za pomocą pakietu *ipred* lub *caret*. Zasada działania modelowania przy pomocy tych pakietów jest podobna, pakiet *caret* pozwala natomiast na łatwą analizę istotności atrybutów. Celem porównania wyników realizowanych przez oba algorytmy, zdecydowano się na użycie ich obu.

Modele oparte o drzewo regresji cierpią z powodu faktu, że w liściach, które reprezentują pewny podzbiór przestrzeni (na której buduje się model), wynikiem jest pojedyncza liczba. Przykładowo dla zależności celu od jednego atrybutu można to funkcja reprezentująca model może być nieciągła i stworzona z odcinków o zerowym współczynniku kierunkowym. Dużo lepiej byłoby aproksymować tę zależność funkcją kawałkami liniową. W tym celu wykorzystuje się regresję kawałkami liniową (*grow.modtree* z pakietu *dmr.regtree*). Za pomocą listy *plr\_args* ograniczono głębokość drzewa do 10 oraz wymuszono co najmniej 2 podziały.

Modele opisane w tym podrozdziale zostały ocenione za pomocą procedury k-krotnej walidacji krzyżowej z wykorzystaniem miar jakości opisanych dalej.

## 4.2. Miary jakości

Dla zbudowanych modeli oblicza się następujące miary jakości:

1. CC - współczynnik korelacji liniowej Pearsona

$$CC = \frac{cov(P, A)}{var(P) \cdot var(A)}$$

## 2. MSE - błąd średniokwadratowy

$$MSE = \frac{(p_1 - a_1)^2 + \dots + (p_n - a_n)^2}{n}$$

## 3. RMSE - pierwiastek z błędu średniokwadratowego

$$RMSE = \sqrt{\frac{(p_1 - a_1)^2 + \dots + (p_n - a_n)^2}{n}}$$

## 4. MAE - średni błąd względny

$$MAE = \frac{|p_1 - a_1| + \dots + |p_n - a_n|}{n}$$

## 5. RSE - względny błąd kwadratowy

$$RSE = \frac{(p_1 - a_1)^2 + \dots + (p_n - a_n)^2}{(a_1 - \bar{a})^2 + \dots + (a_n - \bar{a})^2}$$

## 6. RRSE - pierwiastek ze względnego błędu kwadratowego

$$RRSE = \sqrt{\frac{(p_1 - a_1)^2 + \dots + (p_n - a_n)^2}{(a_1 - \bar{a})^2 + \dots + (a_n - \bar{a})^2}}$$

## 7. RAE - błąd względny

$$RAE = \frac{|p_1 - a_1| + \dots + |p_n - a_n|}{|a_1 - \bar{a}| + \dots + |a_n - \bar{a}|}$$

TODO tutaj ładnie się wpasuje co po kolei w kodzie poszło

### 4.3. Procedury oceny

Aby móc ocenić model pod względem przydatności zastosowano metodę k-krotnej walidacji krzyżowej. Kod zawarto w funkcji `model_eval()`, która stanowi . Zbiór testowy jest dzielony losowo na k podzbiorów równej wielkości. W kolejnych iteracjach każdy ze zbiorów jest traktowany jako zbiór testowy, podczas gdy na reszcie danych buduje się model. Następnie modele są uśredniane i następuje predykcja. Po predykcji modelu na zbiorze testowym wyznacza się miary jakości opisane w 4.2.

### 4.4. Procedury oceny

Wyniki 10-krotnej walidacji krzyżowej zostały przedstawione w tabeli 4.1. Przedstawia ona miary jakości wyznaczone za pomocą tej procedury dla każdego algorytmu.

## 5. Wnioski

TODO

Tablica 4.1: Walidacja krzyżowa — porównanie parametrów

| <b>Appliances</b> | <b>MSE</b> | <b>RRSE</b> | <b>MAE</b> | <b>RMSE</b> | <b>RAE</b> | <b>CC</b> | <b>RSE</b> |
|-------------------|------------|-------------|------------|-------------|------------|-----------|------------|
| <i>lm()</i>       | 14914.800  | 2.5702027   | 75.20398   | 122.12616   | 1.9553172  | 0.3425775 | 6.6059421  |
| <i>rpart()</i>    | 11646.595  | 1.2051757   | 58.13124   | 107.91939   | 1.0320860  | 0.5698647 | 1.4524484  |
| <i>ipred</i>      | 9016.412   | 1.3234627   | 53.02221   | 94.95479    | 1.0308015  | 0.6983742 | 1.7515534  |
| <i>PLR</i>        | 266453.137 | 0.9929399   | 78.68274   | 516.19099   | 0.8137909  | 0.1532704 | 0.9859296  |
| <i>caret</i>      | 8958.720   | 1.2985949   | 52.82693   | 94.65051    | 1.0113670  | 0.6991588 | 1.6863487  |
| <b>lights</b>     | <b>MSE</b> | <b>RRSE</b> | <b>MAE</b> | <b>RMSE</b> | <b>RAE</b> | <b>CC</b> | <b>RSE</b> |
| <i>lm()</i>       | 0,259      | 0,404       | 0,185      | 0,404       | 0,404      | 0,404     | 0,404      |
| <i>rpart()</i>    | 0,259      | 0,404       | 0,185      | 0,404       | 0,404      | 0,404     | 0,404      |
| <i>ipred</i>      | 0,259      | 0,404       | 0,185      | 0,404       | 0,404      | 0,404     | 0,404      |
| <i>PLR</i>        | 0,259      | 0,404       | 0,185      | 0,404       | 0,404      | 0,404     | 0,404      |
| <i>caret</i>      | 0,259      | 0,404       | 0,185      | 0,404       | 0,404      | 0,404     | 0,404      |