

Warszawa, 19 stycznia 2019

Metody ewolucyjne i uczenie się maszyn

Projekt 2

Dokumentacja końcowa

Na jednym z benchmarków dot. modelowania szeregów czasowych (np. NN3 lub M3) porównać metodę xgboost z modelem liniowym.

Prowadzący:
dr inż. Paweł Zawistowski

Wykonali:
Michał Kruszewski, Łukasz Błaszka

Spis treści

1.	Cel	3
2.	Założenia	3
3.	Krótki wstęp teoretyczny	3
3.1.	ARIMA	3
3.2.	Metoda xgboost	4
4.	Zbiór danych	4
5.	Implementacja	5
5.1.	ARIMA	6
5.2.	XGBoost	6
6.	Prezentacja wyników	7

Bibliografia	13
---------------------	-----------

1. Cel

Celem projektu było praktyczne poznanie dowolnie wybranego modelu liniowego oraz metody xgboost, stosowanych do modelowania szeregów czasowych. Jako model liniowy wybrany został popularny model ARIMA (*ang. AutoRegressive Integrated Moving Average*). Realizacja celu oparta została na porównaniu obu metod na benchmark'u NN3.

2. Założenia

W specyfikacji wstępnej projektu znalazły się następujące założenia:

- Zadanie rozwiązane miało zostać w języku *R*.
- Jako benchmark testowy wybrany został nowszy benchmark NN3. Zawiera on mniej szeregów czasowych niż benchmark M3 (111 vs 3003), dlatego lepiej nadaje się do celów dydaktycznych (subiektywna ocena osób realizujących projekt).
- Jako model liniowy wybrany miał zostać popularny model ARIMA (*ang. AutoRegressive Integrated Moving Average*), uznawany za jedną z opcji do modelowania szeregów niestacjonarnych.
- W celu dopasowania najlepszego modelu ARIMA wykorzystana miała zostać metoda *auto.arima* pochodząca z pakietu *forecast*.
- W celu predykcji metodą xgboost wykorzystana miała zostać biblioteka *xgboost* (<https://github.com/dmlc/xgboost>).
- Ze względu na brak jakiegokolwiek doświadczenia z metodą xgboost, oraz liczbę jej parametrów, sposób optymalizacji określony miał zostać na etapie implementacji.

3. Krótki wstęp teoretyczny

3.1. ARIMA

Model ARIMA jest rozszerzeniem modelu ARMA, który poddawany jest różnicowaniu w celu usunięcia trendu. Model ARMA(p, q) dla stacjonarnego szeregu czasowego z_t ma postać:

$$z_t = \phi_1 z_{t-1} + \dots + \phi_p z_{t-p} + w_t + \psi_1 w_{t-1} + \dots + \psi_q w_{t-q} \quad (1)$$

gdzie z_t jest szeregiem stacjonarnym o zerowej średniej, $\phi_i, \psi_i \in \mathbb{R} (\phi_p \neq 0, \psi_q \neq 0)$, a $w_i \sim N(0, \sigma^2)$.

Model ARMA jest połączeniem dwóch modeli:

- Model AR(p) - proces autoregresyjny p rzędu określa wpływ p poprzednich obserwacji na wartość obecną z_t . Przy wyliczaniu modelu AR poszukiwane są wartości (ϕ_1, \dots, ϕ_p) by spełnić równanie[4]:

$$z_t = c + \phi_1 z_{t-1} + \phi_2 z_{t-2} + \dots + \phi_p z_{t-p} + w \quad (2)$$

- Model MA(q) - średnia ruchoma q rzędu opisuje zależność obserwacji z_t od aktualnego zakłócenia w_t oraz q poprzednich. Poszukiwane są wartości (ψ_1, \dots, ψ_q) przy znanych wartościach zakłóceń (w_1, \dots, w_t) . Wzór modelu MA[4]:

$$z_t = c + \psi_1 w_{t-1} + \psi_2 w_{t-2} + \dots + \psi_q w_{t-q} + w_t \quad (3)$$

Model ARIMA do usuwania trendu wykonuje operację różnicowania szeregu gdzie d to opóźnienie (lag). Wzór różnicowania szeregu[4]:

$$z'_t = z_t - z_{t-d} \quad (4)$$

Ostatecznie model ARIMA(p,d,q) ma postać[4]:

$$z'_t = c + \phi_1 z'_{t-1} + \dots + \phi_p z'_{t-p} + w_t + \psi_1 w_{t-1} + \dots + \psi_q w_{t-q} \quad (5)$$

3.2. Metoda xgboost

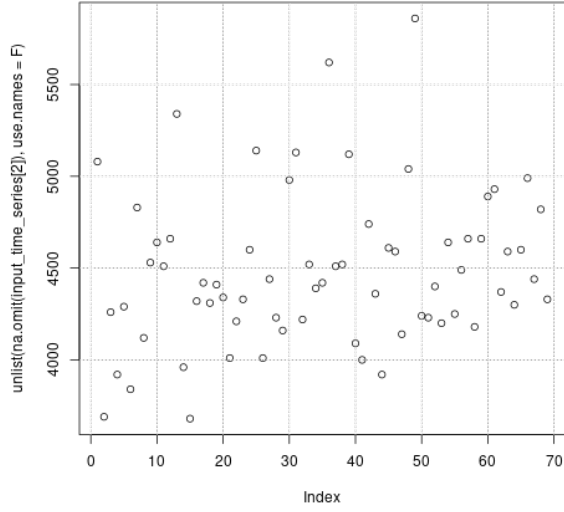
Metoda xgboost (*Extreme Gradient Boosting*) wywodzi się z metody GBM (*ang. Gradient Boosting Machine*). Obie te metody opierają się o gradientowe wzmacnianie drzew (*ang. gradient boosted trees*). U ich podstaw znajduje się model zespołu drzew decyzyjnych (*ang. decision tree ensembles*). W zespole takim pojedyncze drzewo zazwyczaj słabo radzi sobie z klasyfikacją/regresją, dlatego do predykcji wykorzystywany jest cały zespół drzew. Idea wzmacniania gradientowego polega na tworzeniu ciągu prostych drzew, z których każde kolejne jest zbudowane do predykcji reszt generowanych przez poprzednie.

Różnica w idei działania xgboost a GDM jest niewielka. xgboost używa bardziej regularnej formalizacji modelu w celu kontroli przeuczenia (*over-fitting*). Dodatkowo xgboost jest mocno zoptymalizowany w obszarach szybkości obliczeń oraz zużycia pamięci.

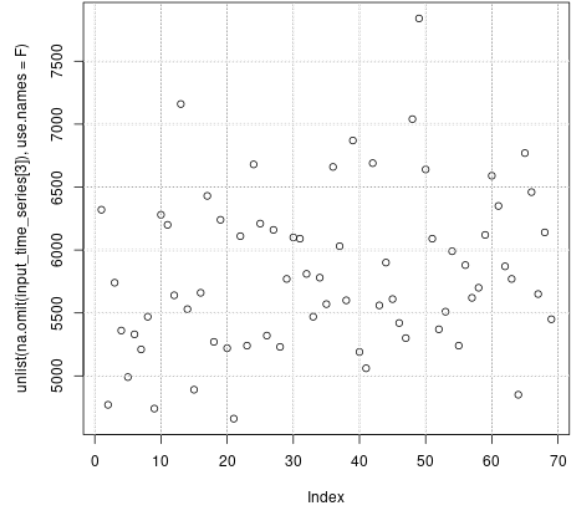
Trzeba dodać informację co jest x a co jest y podczas predykcji w XGBoost.

4. Zbiór danych

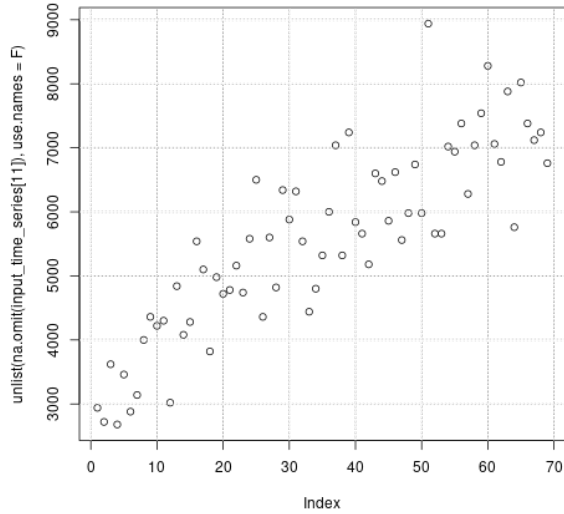
Liczba szeregów czasowych znajdujących się w benchmark'u NN3 wynosi 111. Długość wektora testowego dla każdego z szeregów wynosi 18. Zbiór danych jest zaczerpnięty z jednorodnej populacji empirycznych biznesowych szeregów czasowych. Wśród danych dominują szeregi, które wydają się być mocno zaszumione (rys. 1a, 1b). Znaleźć tam również można szeregi czasowe z wyraźnym trendem (rys. 1c) lub wahaniami okresowymi (rys. 1d).



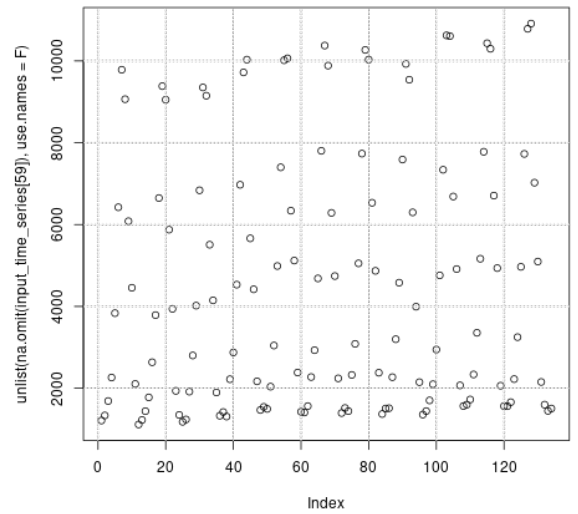
(a) Dane zaszumione



(b) Dane zaszumione



(c) Trend



(d) Wahania okresowe

Rysunek 1: Przykładowe szeregi czasowe z benchmark'u NN3.

Szeregi czasowe w benchmark'u NN3 są stosunkowo krótkie (min 68, max 144), a sam benchmark uznawany jest przez ekspertów z dziedziny modelowania szeregów czasowych jako dość „prosty”.

5. Implementacja

Szeregi czasowe z benchmark'u NN3 zostały znormalizowane do przedziału $< 1 : 2 >$ w celu umożliwienia porównywania błędów bezwzględnych. Wadą normalizacji może być przeskalowanie większości z wartości szeregu do wąskiego przedziału w przypadku, kiedy w danych znajduje się co najmniej 1 mocno odstający punkt [1]. Przesunięcie zakresu z przedziału $< 0 : 1 >$ do $< 1 : 2 >$ pozwala również na obliczanie błędów

względnych bez obawy dzielenia przez 0.

Predykcja wartości odbywa się z krokiem o długości 1. Oznacza to, iż przy predykcji wartości dla chwil czasowych $t + n$ wykorzystywane są rzeczywiste wartości szeregu z chwil czasowych wcześniejszych od $t + n$.

5.1. ARIMA

Predykcje z krokiem o długości 1 zrealizowano tworząc oddzielny model dla każdej wartości szeregu testowego. Szereg uczący w każdym kroku jest zwiększany o jedną wartość z szeregu testowego. Do uzyskania przewidywanej wartości z wyliczonego modelu wykorzystaniu funkcję `forecast`. Parametr "h" określa dla ilu kroków ma zostać wyliczona przewidywana wartość.

Poniżej kod dokonuje predykcji z krokiem o długości 1 na podstawie wyliczonego modelu ARIMA'y dla N-tego przebiegu czasowego benchmark'u NN3:

```
learn_data_length = length( tmp_ts_learn )
for (j in 1:TEST_DATA_LENGTH) {
  ts_end = learn_data_length + j
  ts <- ts(tmp_ts[1:ts_end])
  arima_model = auto.arima(ts)
  arima_forecast_all = forecast(arima_model, h = 1)
  arima_forecast = as.data.frame(arima_forecast_all)$'Point_Forecast'
  arima_forecast_oneAhead[j] = arima_forecast
}
```

5.2. XGBoost

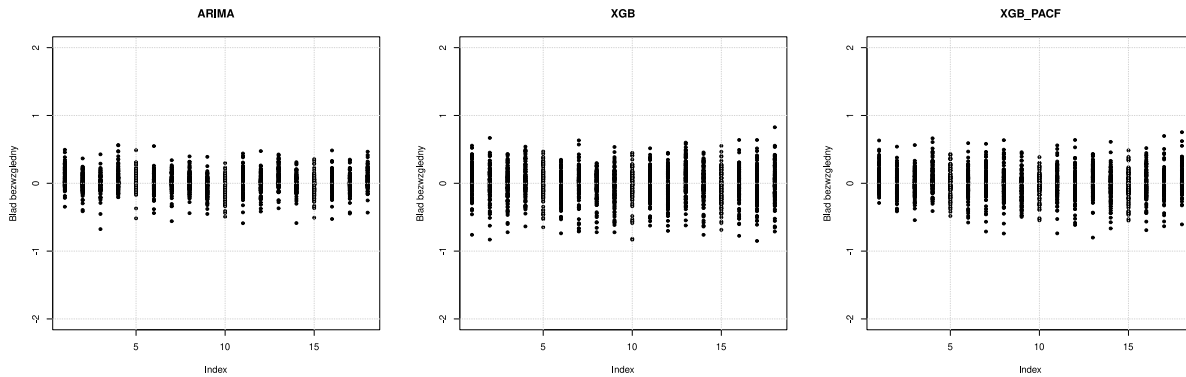
Predykcja za pomocą metody XGBoost zrealizowana została na dwa sposoby, różniące się podejściem do wyboru atrybutów/cech wykorzystywanych do uczenia i przewidywania.

XGBoost z subiektywnie wybranymi cechami

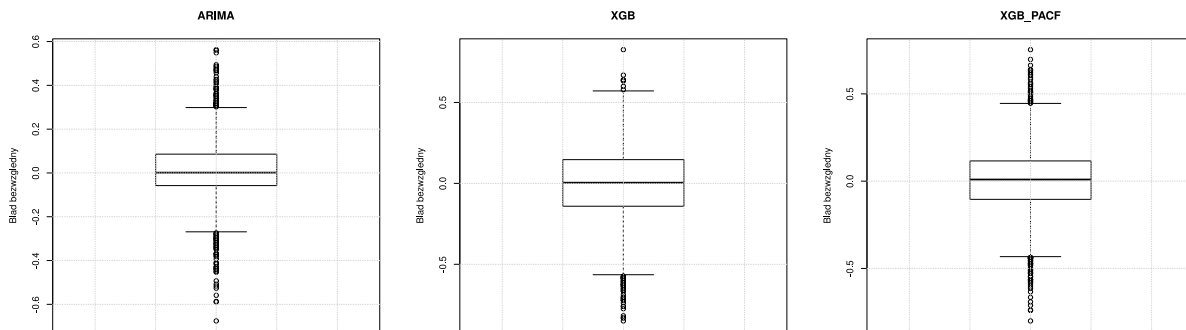
Osoby realizujące projekt dokonały subiektywnego wyboru dwóch atrybutów. Pierwszy z nich to miesiąc, z którego pochodzi dana wartość szeregu czasowego. Drugi to tzw. wskaźnik siły względnej (*ang. relative strength index*), określający siłę trendu. Oparty jest on o średnią ruchomą [2]. W projekcie obliczany jest z wykorzystaniem ważonej średniej ruchomej (*ang. weighted moving average*).

XGBoost na podstawie autokorelacji częściowej

Metoda ta polegała na obliczaniu funkcji autokorelacji częściowej i wyznaczaniu na jej podstawie tzw. lagów, czyli wartości szeregu przesuniętych w czasie. Wybrana została funkcja autokorelacji częściowej, a nie funkcja autokorelacji, ponieważ wydaje się być ona bardziej odpowiednia dla metod nieliniowych. Autokorelacja częściowa dla lagu k jest korelacją, która usuwa efekt jakichkolwiek korelacji dla lagów krótszych niż k [3]. Ze względu na fakt, iż wszystkie szeregi czasowe z benchmark'u NN3 są danymi biznesowymi, a okres ich próbkowania wynosił 1 miesiąc, maksymalne opóźnienie (lag) dla którego jest wyznaczana funkcja autokorelacji częściowej wynosi 12. Szeregi czasowe w benchmark'u NN3 są stosunkowo krótkie, w związku z tym należy brać pod uwagę fakt, iż wykorzystywanie lagów jako atrybutów zmniejsza liczbę elementów zbioru uczącego. Do zbioru atrybutów wybierane były wszystkie lags, dla których wartość absolutna funkcji autokorelacji częściowej była większa lub równa 0.1.



Rysunek 2: Błąd bezwzględny



Rysunek 3: Wykres pudełkowy

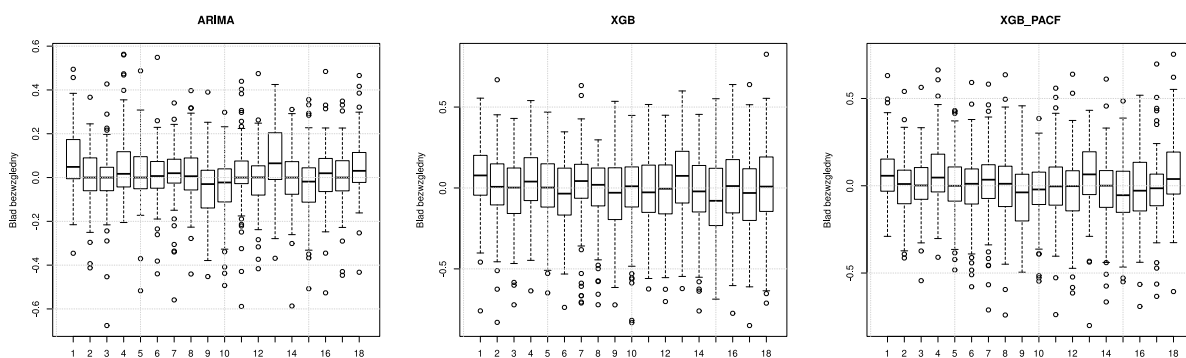
Optymalizacja hiperparametrów

6. Prezentacja wyników

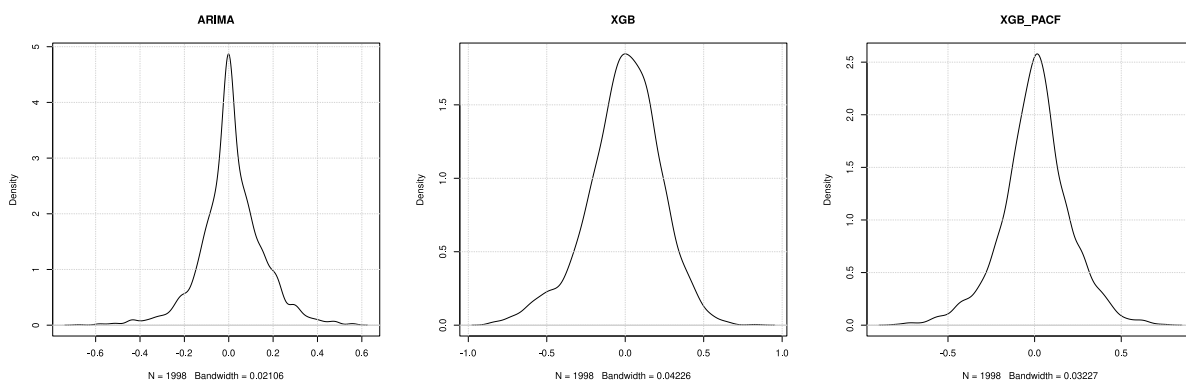
Porównanie modelu ARIMA z metodą xgboost dokonane zostanie na podstawie przebiegu funkcji gęstości prawdopodobieństwa względnych błędów. Wartości błędów nie będą uśredniane dla poszczególnych szeregów, co oznacza, że długość wektora błędów dla którego zostanie przedstawiony przebieg funkcji gęstości prawdopodobieństwa wynosić będzie $111 \times 18 = 1998$.

Zagregowana funkcja gęstości prawdopodobieństwa błędów może ukrywać niektóre właściwości stosowanych modeli, dlatego zostanie ukazany również, na wspólnym wykresie, przebieg błędów dla wszystkich szeregów czasowych.

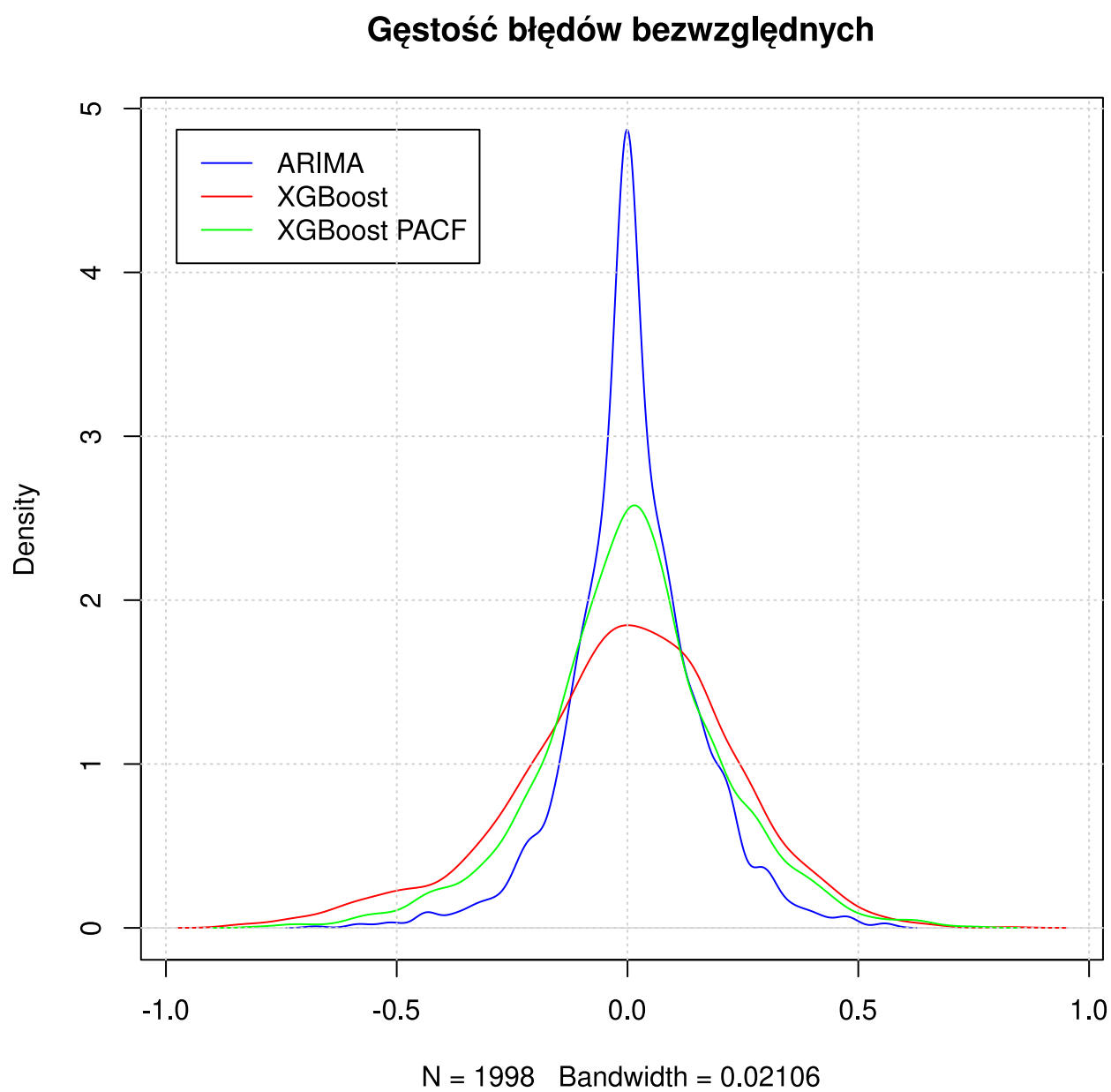
Wyniki strojenia parametrów metody xgboost zaprezentowane zostaną w analogiczny sposób. Dodatkowo zamieszczone mogą zostać fragmenty wydruków stosowanych funkcji, niosące wartościowe informacje.



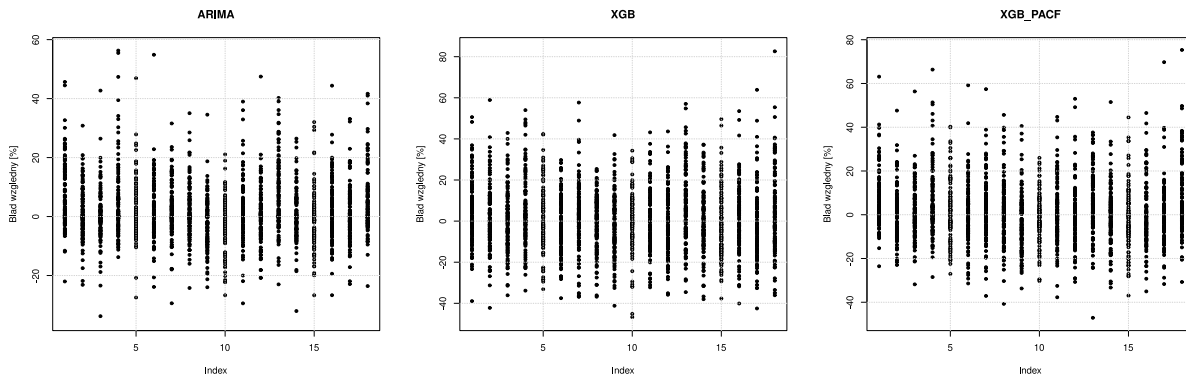
Rysunek 4: Wykres pudełkowy dla poszczególnych obserwacji szeregu testowego



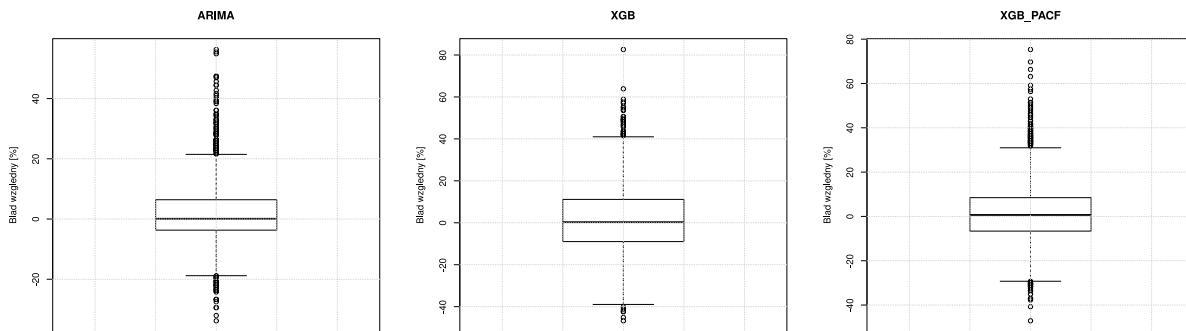
Rysunek 5: Gęstość błędów



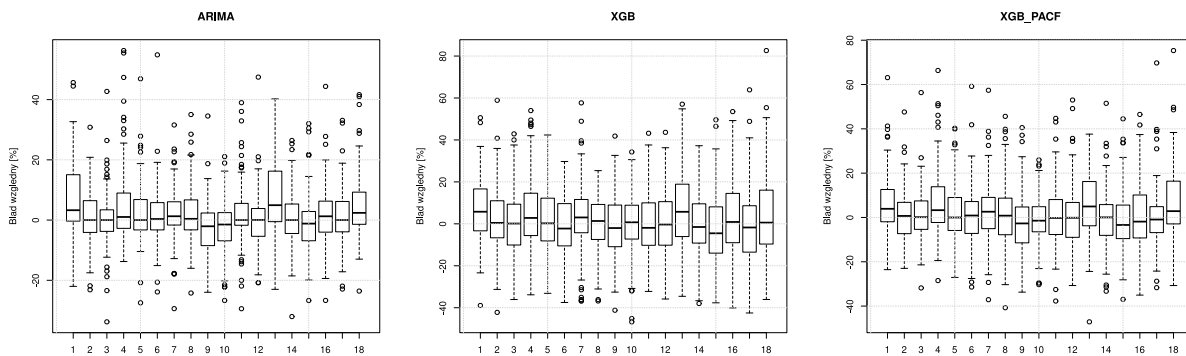
Rysunek 6: Gęstość błędu bezwzględnego (wspólny wykres)



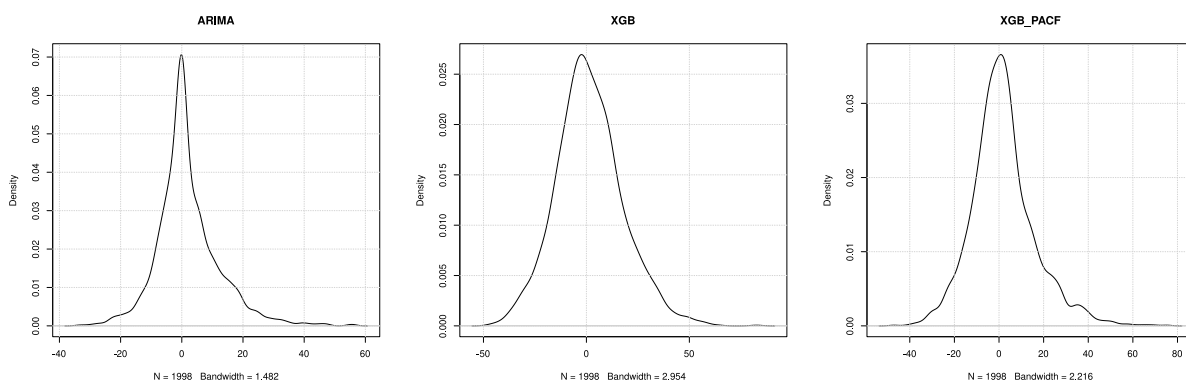
Rysunek 7: Błąd względny



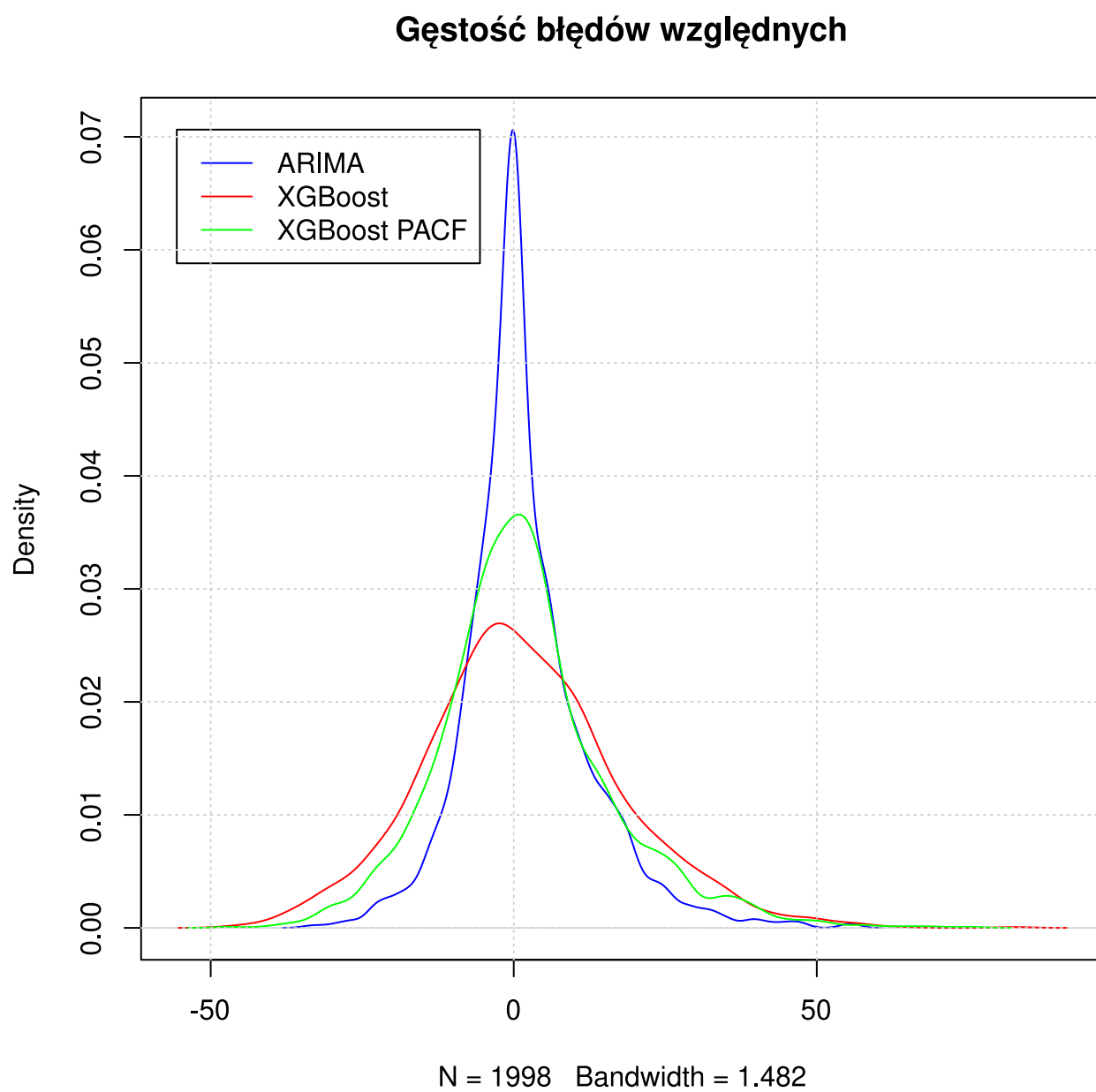
Rysunek 8: Wykres pudełkowy dla względnych wartości



Rysunek 9: Wykres pudełkowy względnych wartości błędów dla poszczególnych obserwacji szeregu testowego



Rysunek 10: Gęstość błędu względne



Rysunek 11: Gęstość błędu względnego (wspólny wykres)

Bibliografia

- [1] Standardization vs. normalization | Data Mining Blog - [www.dataminingblog.com](http://www.dataminingblog.com/standardization-vs-normalization/). <http://www.dataminingblog.com/standardization-vs-normalization/>.
- [2] Relative strength index, August 2018. https://en.wikipedia.org/w/index.php?title=Relative_strength_index&oldid=856226469.
- [3] Paul S. P. Cowpertwait and Andrew V. Metcalfe. *Introductory Time Series with R*. Springer, Dordrecht ; New York, 2009 edition edition, June 2009.
- [4] Sebastian Zając. Modelowanie szeregów czasowych za pomocą procesów arma i arima. <http://sebastianzajac.pl/work/BSc.pdf>.