# [PSZT] PZ.U17 - Marketing bankowy

Adrian Brodzik Jakub Górka

27 stycznia 2020

## Zadanie

Porównać algorytmy regresji logistycznej i XGBoost.

## Teza

**Efektywność regresji logistycznej i XGBoost jest równa dla problemu ustalenia potencjalnych klientów lokat bankowych pewnej portugalskiej instytucji finansowej.**

## Analiza danych

Źródło danych: `https://archive.ics.uci.edu/ml/datasets/Bank+Marketing`
Plik danych: `bank-additional-full.csv`
Opis danych: `bank-additional-names.txt`

Zbiór danych wykorzystywanych w projekcie pochodzi z kampanii marketingowej lokaty terminowej prowadzonej przez portugalską instytucję finansową. Owa kampania oparta była o kontakt telefoniczny. Każdy rekord przedstawia jednego klienta, z podziałem na poszczególne atrybuty.

Atrybuty wejściowe przedstawiają szczegóły danego klienta umożliwiające dokładniejszą klasyfikację, natomiast atrybutem wyjściowym jest informacja o decyzji przystąpieniu do lokaty terminowej. Podział względem poszczególnych atrybutów, pozwala stwierdzić iż dominującą grupą wiekową są osoby w wieku 31-40, niestety nie wiemy nic o podziale na płeć poszczególnych osób. Kolejną obserwacją jest większościowy udział osób po ślubie. Dominującym zawodem wśród klientów były osoby administracji publicznej. Pod

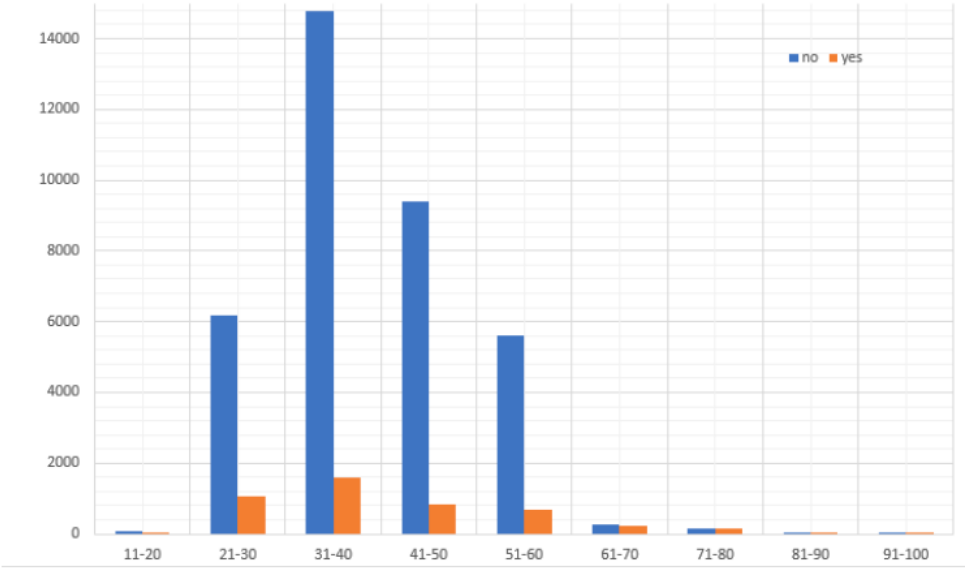
względem wykształcenia osoby które najczęściej decydowały się na lokatę były osobami, które uzyskały wykształcenie podstawowe oraz uniwersyteckie. Atrybut *duration* nie jest uwzględniany w analizie i nie jest używany w doświadczeniu tworzenia modelu predykcyjnego ze względu na niewiedzę zakończenia trwającej rozmowy.

Opis atrybutów

| Atrybut | typ | Opis |
|---|---|---|
| age | Int | |
| job | Kategoryjny | Rodzaj zawodu: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown' |
| marital | Kategoryjny | Stan cywilny: 'divorced','married','single','unknown'; uwaga: jako 'divorced' również oznaczona osobę owdowiałą |
| education | Kategoryjny | Poziom wykształcenia: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown' |
| default | Kategoryjny | Ma nieuiszczone należności: 'no','yes','unknown' |
| housing | Kategoryjny | Ma kredyt hipoteczny: 'no','yes','unknown' |
| loan | Kategoryjny | Ma kredyty inny: 'no','yes','unknown' |
| contact | Kategoryjny | Rodzaj kontaktu: 'cellular','telephone' |
| month | Kategoryjny | Miesiąc w którym wystąpił ostatni kontakt: 'jan', 'feb', 'mar', ..., 'nov', 'dec' |
| day_of_week | Kategoryjny | Dzień tygodnia ostatniego kontaktu : 'mon','tue','wed','thu','fri' |
| duration | Int | Długość trwania ostatniej rozmowy. |
| campaign | Int | Liczba prób kontaktu z klientem. |
| pdays | Int | Liczba dni, które upłynęły od ostatniego kontaktu z klientem w poprzedniej kampanii. 999 oznacza, że próba kontaktu z klientem nie wystąpiła. |
| previous | Int | Liczba kontaktów wykonana przed obecną kampania do danego klienta |
| poutcome | Kategoryjny | Wynik poprzedniej kampanii marketingowej |
| emp.var.rate | Float | Wskaźnik zatrudnienia |
| cons.price.idx | Float | Wskaźnik cen towarów i usług konsumpcyjnych |
| cons.conf.idx | Float | Wskaźnik ufności konsumenckiej |
| euribor3m | Float | Referencyjna wysokość oprocentowania- 3-miesięczna |
| nr.employed | Float | Liczba pracowników |
| y | Binarny | Czy klient założył lokatę: 'yes','no'. |

Udział poszczególnych grup wiekowych z podziałem na decyzję o założeniu lokaty

| AGE | 11-20 | 21-30 | 31-40 | 41-50 | 51-60 | 61-70 | 71-80 | 81-90 | 91-100 |
|---|---|---|---|---|---|---|---|---|---|
| DECISION = 'yes | 57 | 1067 | 1597 | 837 | 668 | 212 | 143 | 54 | 5 |
| DECISION = 'no' | 83 | 6176 | 14788 | 9403 | 5602 | 276 | 160 | 55 | 5 |




Udział osób w poszczególnym stanie cywilnym

| MARITAL | divorced | married | single | unknown |
|---|---|---|---|---|
| DECISION = 'yes' | 476 | 2532 | 1620 | 12 |
| DECISION = 'no' | 4136 | 22396 | 9948 | 68 |

Udział poszczególnych zawodów

| JOB | admin. | blue-collar | entrepreneur | housemaid | management | retired | self-employed | services | student | technician | unemployed | unknown |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DECISION = 'yes' | 1352 | 638 | 124 | 106 | 328 | 434 | 149 | 323 | 275 | 730 | 144 | 37 |
| DECISION = 'no' | 9070 | 8616 | 1332 | 954 | 2596 | 1286 | 1272 | 3646 | 600 | 6013 | 870 | 293 |

Podział klientów ze względu na poziom wykształcenia

| EDUCATION | basic.4y | basic.6y | basic.9y | high.school | illiterate | professional.course | university.degree | unknown |
|---|---|---|---|---|---|---|---|---|
| DECISION = 'yes' | 428 | 188 | 473 | 1031 | 4 | 595 | 1670 | 251 |
| DECISION = 'no' | 3748 | 2104 | 5572 | 8484 | 14 | 4648 | 10498 | 1480 |

Podział liczby poszczególnych decyzji na miesiące

| MONTH | mar | apr | may | jun | jul | aug | sep | oct | nov | dec |
|---|---|---|---|---|---|---|---|---|---|---|
| DECISION = 'yes' | 276 | 539 | 886 | 559 | 649 | 655 | 256 | 315 | 416 | 89 |
| DECISION = 'no' | 270 | 2093 | 12883 | 4759 | 6525 | 5523 | 314 | 403 | 3685 | 93 |

Podział liczbę kontaktów

| CAMPAIGN | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | other |
|---|---|---|---|---|---|---|---|---|---|---|---|
| DECISION = 'yes' | 2300 | 1211 | 574 | 249 | 120 | 75 | 38 | 17 | 17 | 12 | 27 |
| DECISION = 'no' | 15342 | 9359 | 4767 | 2402 | 1479 | 904 | 591 | 383 | 266 | 213 | 842 |

# Eksperyment

Dane są ładowane z pliku `bank-additional-full.csv` do pamięci za pomocą modułu `pandas`.

Wartości kolumny celu $y$ są mapowane na 0 dla wartości *no*, albo 1 dla wartości *yes*. Następnie wartości kolumn kategorycznych są jednoznacznie mapowane na liczby całkowite dodatnie. Gdyby model był zapisany i uruchomiony na nowych danych, to wartości danych powinny być dokładnie tak zakodowane, jak podczas procesu uczenia się modelu. Natomiast dane, które nigdy wcześniej się nie pojawiły, są kodowane na wartość wyrażenia `__UNKNOWN__`. Na koniec usuwane są kolumny niepotrzebne (w zależności od wykonywanego eksperymentu).

Do podziału danych wykorzystany jest proces k-krotnej walidacji z modułu `scikit-learn`. Ponadto dane są rozłożone równomiernie ze względu na wartość kolumny celu $y$ (ang. *stratified k-fold*).

W każdej z $k$ iteracji generowane są dwa zbiory danych: treningowy oraz walidacyjny. Zbiór treningowy służy do uczenia modelu `LogisticRegression` z modułu `scikit-learn` albo `XGBRegressor` z modułu `XGBoost`. Zbiór walidacyjny służy do oceny nauczonego modelu, np. za pomocą `roc_auc_score` z modułu `scikit-learn`. ROC-AUC określa w jakim stopniu nauczony model jest w stanie rozpoznać daną klasę.

# Hiperparametry

Do automatycznego strojenia parametrów wykorzystano moduł `hyperopt`, a w szczególności estymator jądrowy gęstości (ang. *Tree-structured Parzen Estimator*).

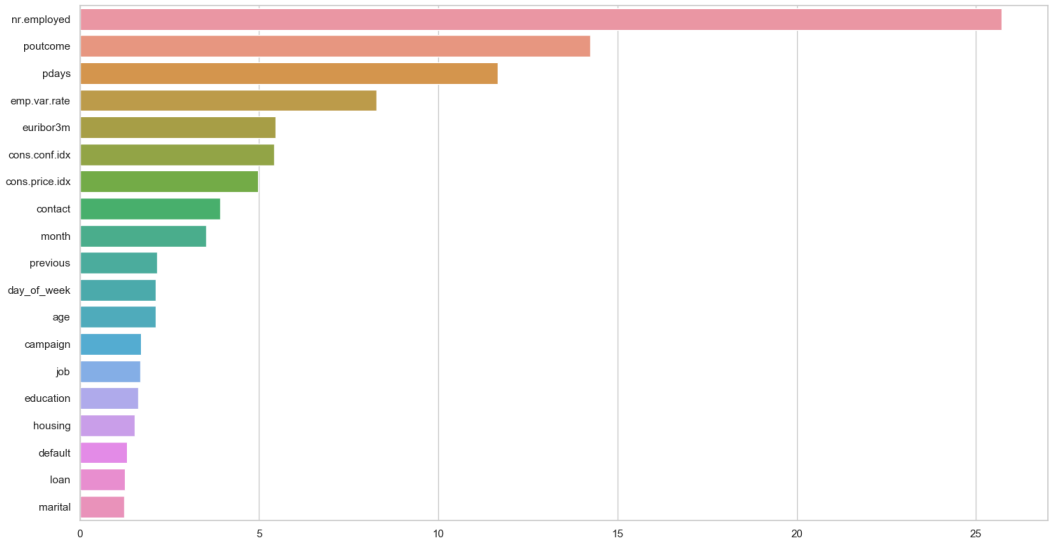
Dla regresji logistycznej przeszukano następujące parametry:

- tol - tolerancja błędu, aby doszło do wczesnego zatrzymania
- C - odwrotność siły regularyzacji
- fit_intercept - czy powinna być dodana stała, np. bias
- class_weight - wagi klas
- solver - algorytm optymalizacyjny
- max_iter - maksymalna liczba iteracji

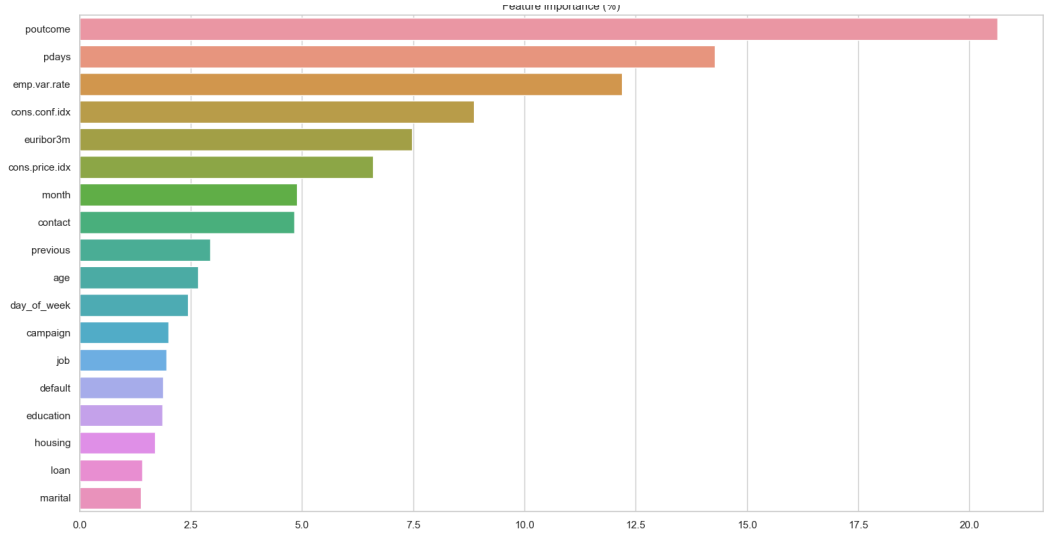
Dla XGBoost przeszukano następujące parametry:

- n_estimators - liczba drzew
- max_depth - maksymalna głębokość drzew
- learning_rate - współczynnik nauczania
- booster - algorytm wzmocnienia gradientowego
- tree_method - algorytm konstruowania drzew
- gamma - minimalna redukcja błędu potrzebna, aby doszło do podziału liścia
- subsample - współczynnik instancji treningowej
- colsample_bytree - współczynnik kolumn podczas budowania drzewa
- colsample_bylevel - współczynnik kolumn dla każdego poziomu w drzewie
- colsample_bynode - współczynnik kolumn do podziału
- reg_alpha - regularyzacja L1
- reg_lambda - regularyzacja L2
- scale_pos_weight - równoważenie dodatnich i ujemnych wag

# Wyniki




Rysunek 1: **Najważniejsze kolumny wg XGBoost** (bez *duration*)



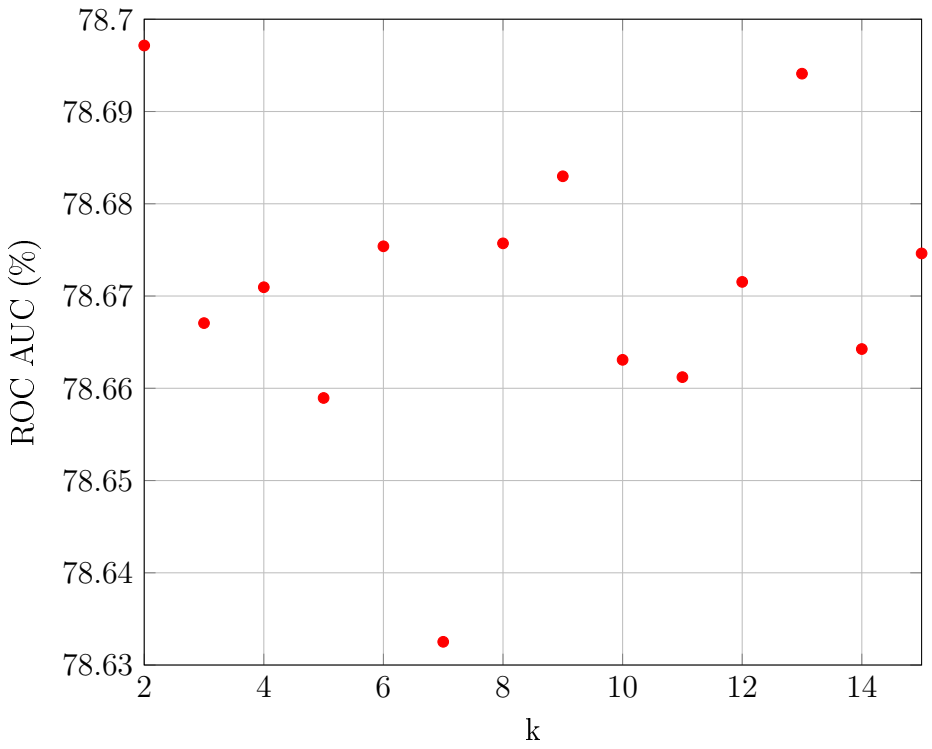

Rysunek 2: **Najważniejsze kolumny wg XGBoost** (bez *duration*, *nr.employed*)

| seed | regresja logistyczna ROC AUC (%) | XGBoost ROC AUC (%) |
|---|---|---|
| 42 | 78.60 | 80.54 |
| 64 | 78.63 | 80.50 |
| 100 | 78.69 | 80.70 |
| 123 | 78.65 | 80.47 |
| 234 | 78.63 | 80.48 |
| 345 | 78.63 | 80.65 |
| 456 | 78.59 | 80.53 |
| 567 | 78.56 | 80.53 |
| 678 | 78.69 | 80.60 |
| 789 | 78.69 | 80.60 |

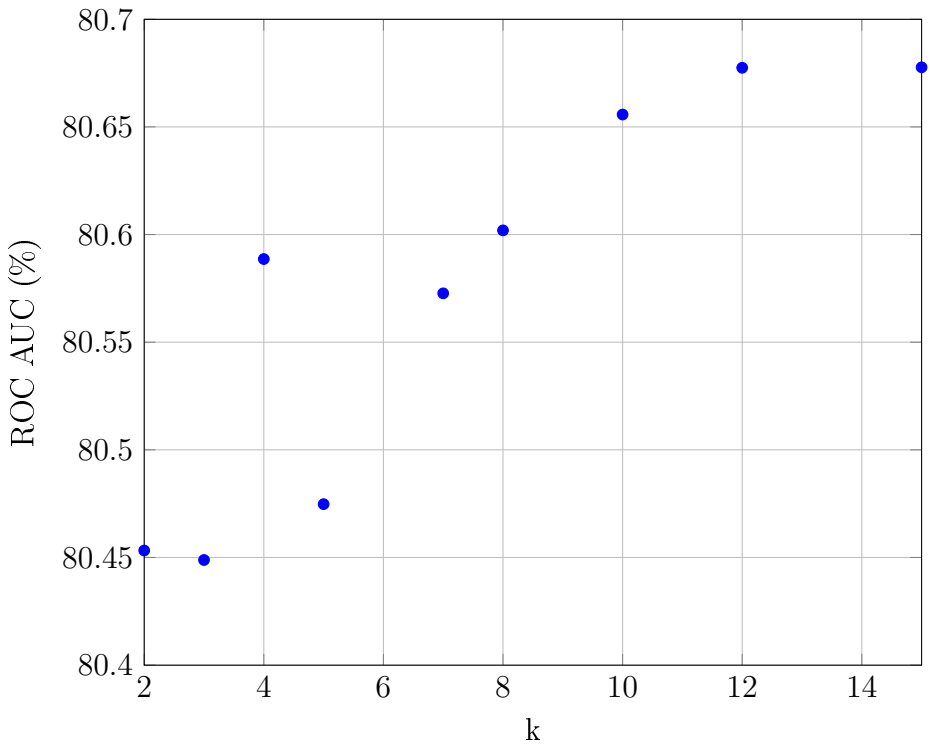
Tablica 1: Wyniki (bez kolumny *duration*)

| seed | regresja logistyczna ROC AUC (%) | XGBoost ROC AUC (%) |
|---|---|---|
| 42 | 93.23 | 95.07 |
| 64 | 93.24 | 95.00 |
| 100 | 93.24 | 95.07 |
| 123 | 93.23 | 95.06 |
| 234 | 93.23 | 95.05 |
| 345 | 93.21 | 95.03 |
| 456 | 93.24 | 95.05 |
| 567 | 93.23 | 94.99 |
| 678 | 93.23 | 95.01 |
| 789 | 93.23 | 95.09 |

Tablica 2: Wyniki (wszystkie kolumny)




Rysunek 3: **Regresja logistyczna i k-krotna walidacja.** Wyższa ocena świadczy o efektywniejszej klasyfikacji.




Rysunek 4: **XGBoost i k-krotna walidacja.** Wyższa ocena świadczy o efektywniejszej klasyfikacji.

# Dyskusja

W przypadku użycia wszystkich kolumn oba algorytmy otrzymały wyższe wyniki. Stało się tak, ponieważ kolumna *duration* ma duży wpływ na kolumnę docelową *y*. W pozostałych testach nie użyto *duration*.

Usunięcie kolumny *nr.employed* nie wpłynęło negatywnie na efektywność modeli mimo tego, że była to najważniejsza kolumna według XGBoost. Najważniejszą kolumną stała się wówczas *poutcome*, tzn. obecny wynik głównie zależy od wyniku poprzedniej rozmowy z klientem. Ponadto ważne są kolumny dotyczące koniunktury rynku, np. *euribor3m*.

Wyższa wartość $k$ w procesie k-krotnej walidacji nie oznacza, że klasyfikator będzie efektywniejszy. Zbiór treningowy jest proporcjonalnie większy, ale zbiór walidacyjny jest proporcjonalnie mniejszy. Ciężko jest wykryć przeuczenie modelu, mając zbyt mały zbiór walidacyjny.

# Wnioski

Teza została potwierdzona w granicy błędu $2\% - 3\%$ ROC AUC. Regresja logistyczna i XGBoost są skutecznymi algorytmami do klasyfikacji obiektów z zadanej przestrzeni i detekcji nieliniowych relacji między nimi.

XGBoost osiągnął lepszy wynik, ale jego złożoność i czas obliczeniowy był znacznie gorszy w porównaniu do prostszego algorytmu regresji logistycznej. Model regresji logistycznej, który właściwie jest uogólnionym modelem liniowym, jest prostszy w konstrukcji i zrozumieniu niż losowo generowane drzewa decyzyjne XGBoost. Dlatego też większość instytucji finansowych preferuje, albo nawet musi, używać tych prostszych modeli.

XGBoost i podobne biblioteki wzmocnień gradientowych mają szerokie zastosowanie w bardziej skomplikowanych problemach, w szczególności konkursach sztucznej inteligencji, gdzie oprócz strojenia parametrów, konieczna jest inżyniera danych.

# Literatura


[1] Josh Starmer. Statquest: Logistic regression, 2018.

[2] Saishruthi Swaminathan. Logistic regression — detailed overview, 2018.

[3] Wai. An example of hyperparameter optimization on xgboost, lightgbm and catboost using hyperopt, 2019.