[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`

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:

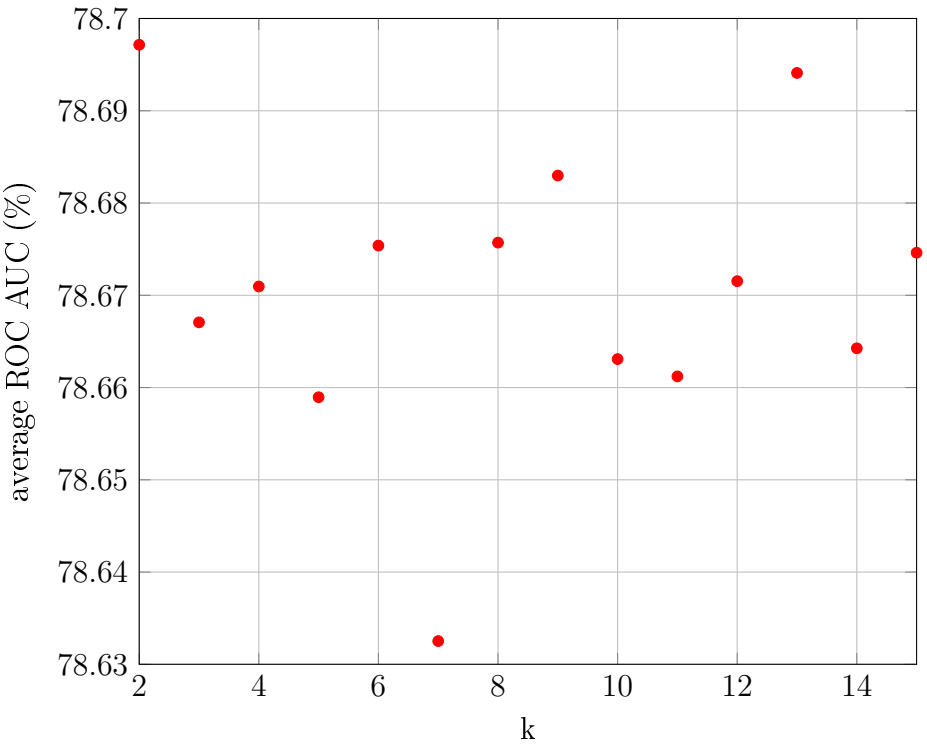
- tol - tolerance for stopping criteria
- C - inverse of regularization strength
- fit_intercept - specifies if a constant should be added to the decision function
- class_weight - weights associated with classes
- solver - algorithm to use in the optimization problem
- max_iter - maximum number of iterations taken for the solvers to converge
- warm_start

Dla XGBoost przeszukano następujące parametry:

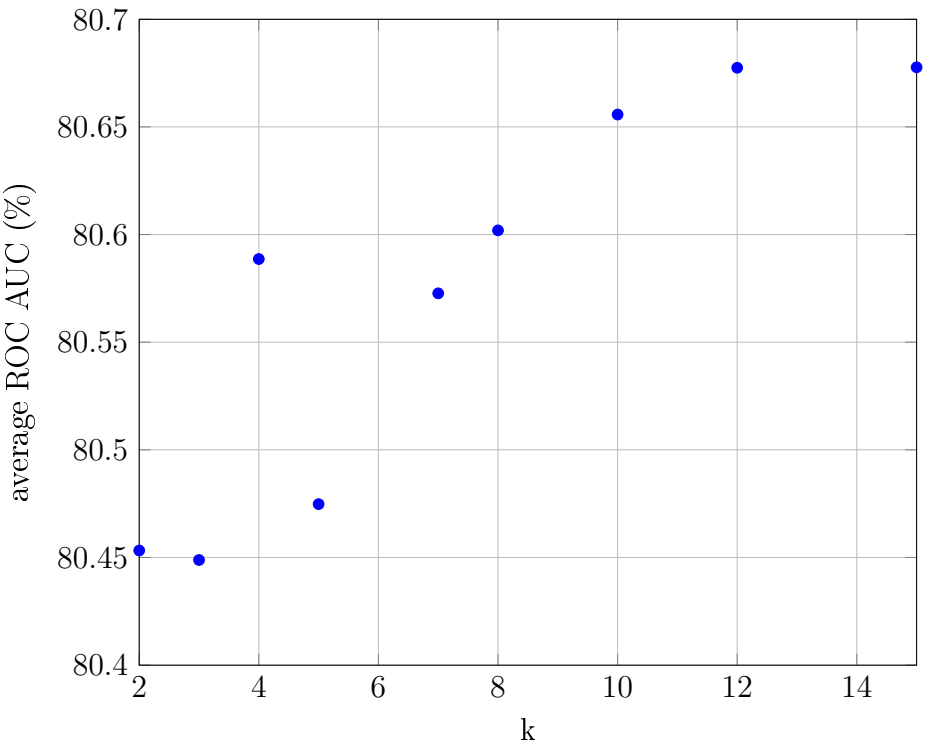
- n_estimators
- max_depth
- learning_rate

- booster
- tree_method
- gamma
- subsample
- colsample_bytree
- colsample_bylevel
- colsample_bynode
- reg_alpha
- reg_lambda
- scale_pos_weight

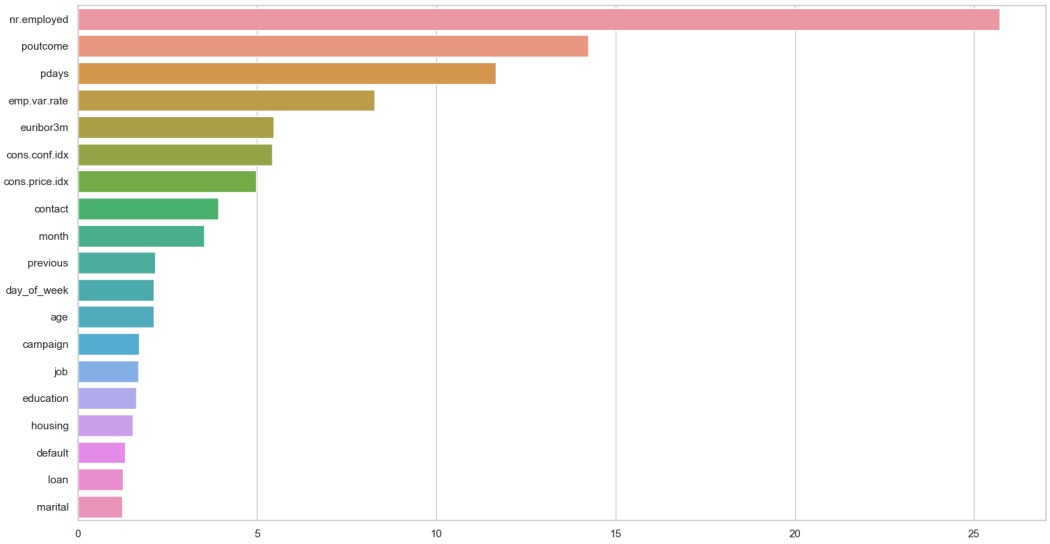
Wyniki



Rysunek 1: **Regresja logistyczna i k-krotna walidacja.** Wyższa ocena świadczy o efektywniejszej klasyfikacji.



Rysunek 2: **XGBoost i k-krotna walidacja.** Wyższa ocena świadczy o efektywniejszej klasyfikacji.



Rysunek 3: **Najważniejsze atrybuty wg XGBoost.**

Dyskusja

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.

Literatura

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

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