AutoML – Projekt 1 – raport

Sebastian David Botero Leonik, Bartosz Pokora, Franciszek Saliński

Listopad 2025

1 Wstęp

W naszym projekcie postanowiliśmy jako metody samplingu wybrać *RandomSearch* oraz *Bayes optimization*. Testowane przez nas modele to *LogisticRegression*, *RandomForest* oraz *XGBoost*, wszystkie wykorzystaliśmy do zadania klasyfikacji binarnej. Wybraliśmy 4 zbiory danych z OpenML: *higgs*, *sf-police-incidents*, *adult* i *credit-g*. W projekcie wykonaliśmy następujące punkty:

1. Obliczenie "optymalnych" defaultów według definicji w [1], korzystając z historii random search.
2. Wyznaczenie najlepszych kombinacji dla danego modelu i zbioru danych za pomocą zarówno random search jak i bayes optimization. Na wykresach porównaliśmy szybkość zbieżności obu metod.
3. Wykorzystanie obliczeń z poprzedniego podpunktu do określenia ogólnej tunowalności algorytmów.
4. Określenie tunowalności pojedynczych hiperparametrów za pomocą dodatkowego random search.

2 Przygotowanie zbiorów danych

W celu umożliwienia użycia wszystkich modeli, wprowadzeniu większej różnorodności w rozmiarach zbiorów danych i przyspieszenia obliczeń, poddaliśmy datasety samplingowi oraz prostemu pipeline'owi, który kodował zmienne kategoryczne, standaryzował zmienne i wybierał podzbiór zmiennych najbardziej istotnych zmiennych według modelu Random Forest. Rozmiary przekształconych datasetów to:

1. credit-g – (1000,12)
2. adult – (20000, 26)
3. sf-police-incidents – (40000, 1469)
4. higgs – (50000, 10)

Ogromna liczba kolumn w sf-police-incidents wynika z binarnego encodowania kolumny unikalnych adresów, gdzie popełnione zostały przestępstwa. Oczywiście coś takiego nie ma sensu, jeśli chcemy stworzyć faktycznie dobry model. Zdecydowanie lepiej byłoby na przykład odzyskać z adresów koordynaty. Stwiedziliśmy natomiast, że zobaczymy co się dzieje, jeżeli podejdziemy do zbiorów danych czysto mechanicznie/matematycznie, nie korzystając z żadnej dodatkowej wiedzy "eksperckiej".

3 Sprawdzane zakresy hiperparametrów

Zakresy sprawdzanych hiperparametrów zaczerpnęliśmy z [1]. Różnice wynikają z użycia innych implementacji rozważanych modeli.

3.1 Logistic Regression

- $C \sim \text{LogUniform}(2^{-10}, 2^{10})$
- $\text{l1\_ratio} \sim \text{Uniform}(0, 1)$

3.2 Random Forest

- $\text{n\_estimators} \sim \text{DiscreteUniform}(5, 300)$
- $\text{min\_samples\_leaf} \sim \text{DiscreteUniform}(1, 100)$
- $\text{max\_features} \in \{\text{sqrt}, \log_2\}$

3.3 XGBoost

- n_estimators $\sim$ DiscreteUniform$(1, 5000)$
- max_depth $\sim$ DiscreteUniform$(1, 15)$
- learning_rate $\sim$ LogUniform$(2^{-10}, 2^{0})$
- subsample $\sim$ Uniform$(0.1, 1.0)$
- colsample_bytree $\sim$ Uniform$(0, 1)$
- colsample_bylevel $\sim$ Uniform$(0, 1)$
- reg_lambda $\sim$ LogUniform$(2^{-10}, 2^{10})$
- reg_alpha $\sim$ LogUniform$(2^{-10}, 2^{10})$

4 Defaulty

Defaulty zostały wybrane według definicji w [1], czyli jako zestaw parametrów, który średnio najlepiej radzi sobie na datasetach. Otrzymane wyniki to:

4.1 Logistic Regression

- $C = 2.703833014029592$
- l1_ratio $= 0.4330742481104214$
- Średnie cv AUC $= 0.7420233693386228$

4.2 Random Forest

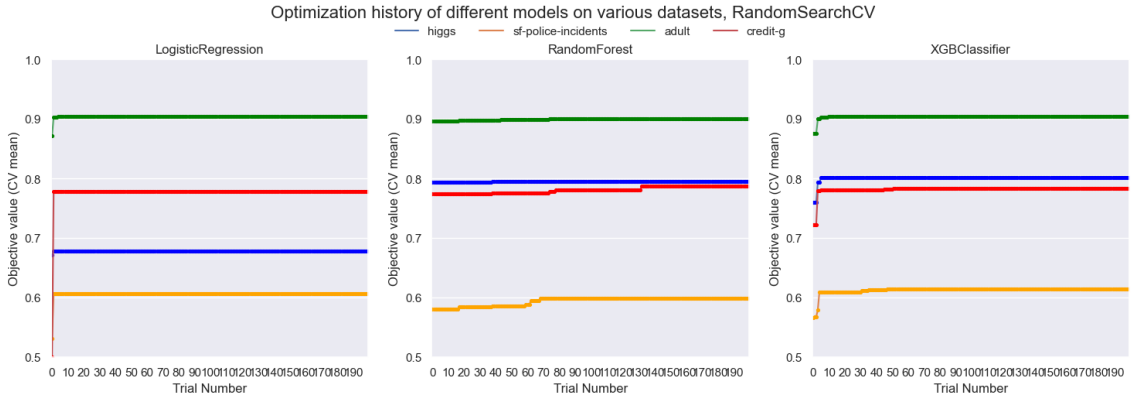
- n_estimators $= 132$
- min_samples_leaf $= 3$
- max_features = 'sqrt'
- Średnie cv AUC $= 0.7690873357968596$

4.3 XGBoost

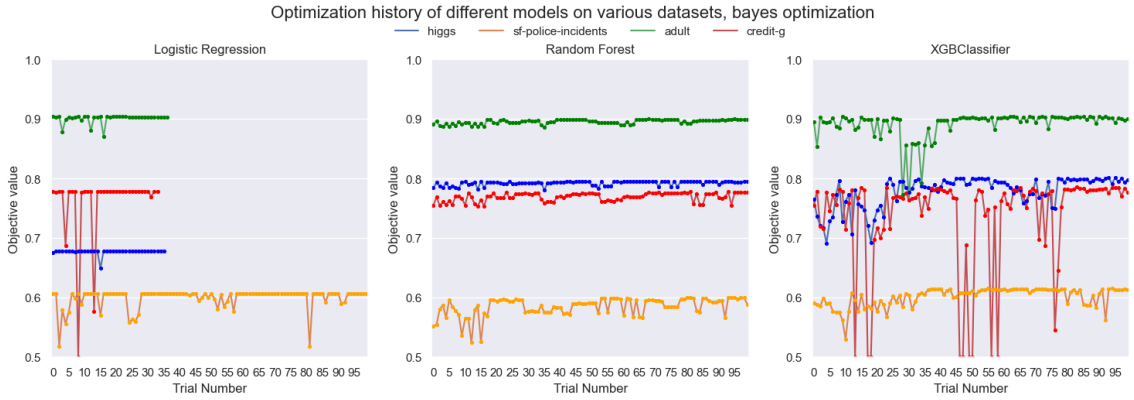
- n_estimators $= 4799$
- max_depth $= 10$
- learning_rate $= 0.0011584669622169$
- subsample $= 0.2316777631402585$
- colsample_bytree $= 0.7259027723025078$
- colsample_bylevel $= 0.6411364308215736$
- reg_lambda $= 0.0685457715422182$
- reg_alpha $= 0.0245844336635187$
- Średnie cv AUC $= 0.7714287119005276$

5 Szybkość zbieżności metod samplingu

Wykresy 1 i 2 ukazują szybkość i stabilność zbieżności obu metod samplingu do znajdowania optymalnych hiperparametrów. Możemy zaobserwować, że *RandomSearch* już po niewielkiej liczbie iteracji znajdował parametry dające wynik bliski maksimum. Podobnie możemy powiedzieć o *Bayes optimization*. Warto zwrócić uwagę, że w przypadku *Bayes optimization* wykres pokazuje wartości w kolejnych iteracjach, a w przypadku *RandomSearch* jest to maksimum wartości z dotychczasowych iteracji.



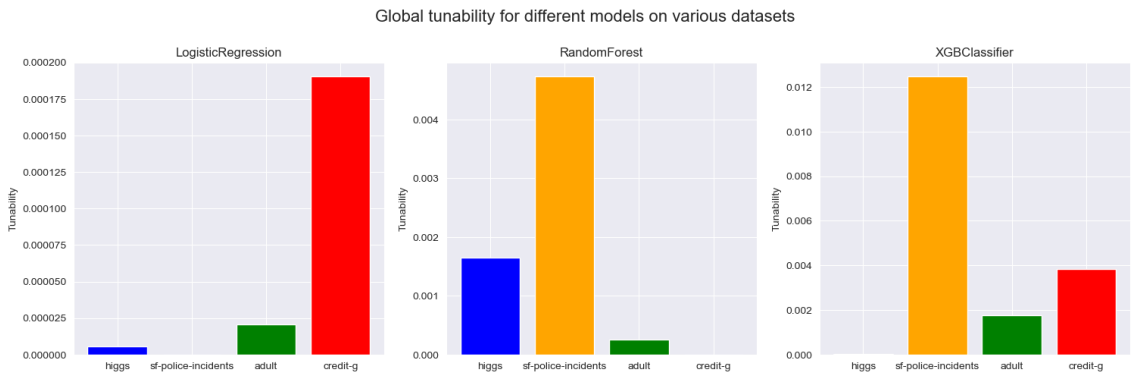
Rysunek 1: Historia optymalizacji RandomSearch



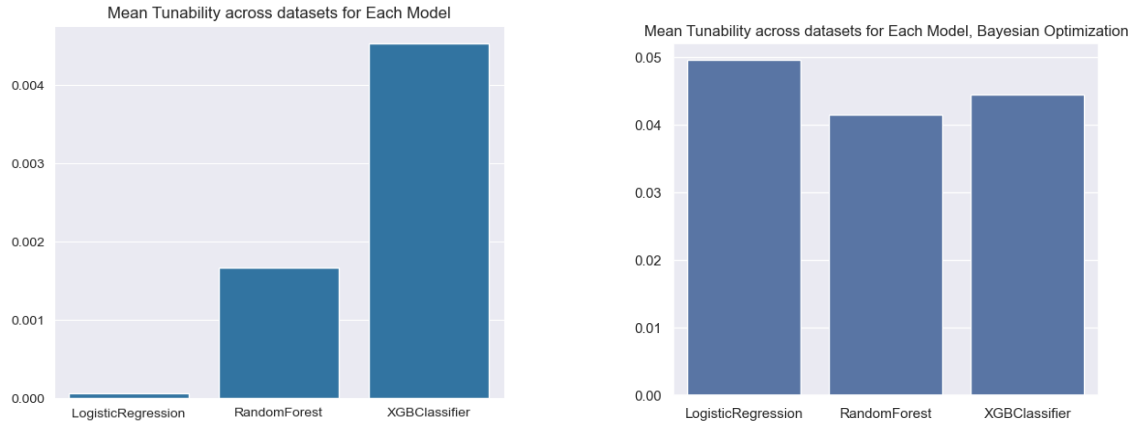
Rysunek 2: Historia optymalizacji bayesowskiej

6 Tunowalność algorytmów

Znalezione tunowalności na badanych przez nas datasetach okazały się dość małe (chociaż bliskie średnim tunowalnościom z eksperymentu w [1], co wskazuje na znikomy zysk z tuningu pod konkretny dataset.



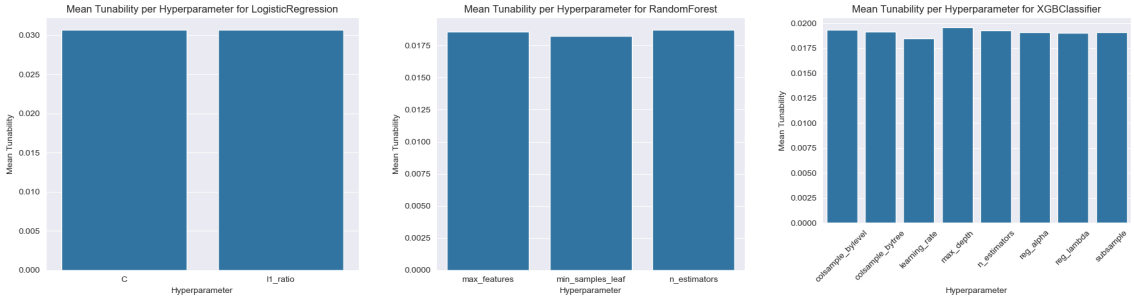
Rysunek 3: Tunowalność modeli



Rysunek 4: Średnia tunowalność modeli dla różnych metod samplingu

7 Tunowalność hiperparametrów

Średnie tunowalności pojedynczych hiperparametrów prawie nie różnią się od siebie, co wskazuje na ich równe znaczenie (lub być może raczej brak znaczenia, przynajmniej na czterech rozważanych datasetach).



Rysunek 5: Średnia tunowalność pojedynczych hiperparametrów

Literatura

[1] Philipp Probst, Anne-Laure Boulesteix, Bernd Bischl, *Tunability: Importance of Hyperparameters of Machine Learning Algorithms*