

Analiza tunowalności hiperparametrów wybranych algorytmów uczenia maszynowego

Automatyczne Uczenie Maszynowe, Praca Domowa 1

Marcel Witas, Wiktor Jakubowski

24 listopada 2023

1 Przygotowanie eksperymentu

1.1 Zbiory danych

Wybraliśmy 4 zbiory danych, pochodzące ze strony *OpenML*. We wszystkich zmienna objaśniana jest zmienną binarną. Dane mają w większości kolumny numeryczne i nie zawierają braków danych. Rozmiar wybranych przez nas zbiorów jest zróżnicowany, aby móc sprawdzić wszechstronność algorytmów.

- *diabetes*: Zbiór danych zawierający informację o osobach testowanych na obecność cukrzycy. Zbiór ma 768 wierszy i 9 kolumn
- *blood-transfusion-service-center*: Zmienna celu jest zmienną binarną reprezentującą, czy dana osoba oddała krew w marcu 2007 r. Ramka ma 748 wierszy, 5 kolumn.
- *phoneme*: Ramka danych do rozróżniania dźwięków nosowych i ustnych. Zbiór zawiera 5404 wierszy, 6 kolumn.
- *wilt* : Zestaw danych zawiera próbki z badania zdalnego wykrywania chorób drzew na obrazach, zawiera 4839 wierszy oraz 6 kolumn.

1.2 Algorytmy

Do eksperymentów wykorzystaliśmy trzy poniższe algorytmy uczenia maszynowego:

- Drzewo decyzyjne (DecisionTreeClassifier z biblioteki `scikit-learn`)
- Las losowy (RandomForestClassifier z biblioteki `scikit-learn`)
- Sieć elastyczna (ElasticNet z biblioteki `scikit-learn`)

Algorytmy te zostały przetestowane pod kątem tunowalności na wyżej wymienionych zbiorach danych. Eksperymenty, które zostały przeprowadzone, zostały opisane w następnej sekcji.

2 Eksperyment

Przeprowadzony eksperyment polegał na analizie tunowalności algorytmów poprzez wytrenowanie każdego z nich określoną liczbę razy na każdym zbiorze. Do każdej iteracji, dokonywano wyboru zestawu hiperparametrów według jednej z dwóch metod *samplingu*:

- Random Search - metoda opierająca się na wyborze punktów z rozkładu jednostajnego
- Bayes Search - metoda opierająca się na technice bayesowskiej wyboru punktów

Dla niektórych zbiorów danych i algorytmu drzewa decyzyjnego udało się dokonać większej liczby iteracji, natomiast przyrównamy w naszych rozważaniach wszystkie algorytmy na jednakowej liczbie prób, w celu uzyskania miarodajnego porównania.

Jako liczbę iteracji przyjęliśmy 100 wraz krosvalidacją równą 3.

W każdej z iteracji daną metodą (Random Search lub Bayes Search) losowany był zestaw hiperparametrów dla trenowanego algorytmu z ustalonej dla każdego zbioru takiej samej siatki hiperparametrów. Tabela 1 przedstawia siatkę hiperparametrów dla każdego algorytmu, poddawanego eksperymentom.

Tabela 1: Testowana siatka hiperparametrów w eksperymentach

Algorytm	Hiperparametr	Typ	Minimum	Maksimum	Wartości
DecisionTreeClassifier					
<i>(Drzewo decyzyjne)</i>	max_features	dyskretny	-	-	[None, 'log2', 'sqrt']
	criterion	dyskretny	-	-	['gini', 'entropy']
	splitter	dyskretny	-	-	['best', 'random']
	max_depth	całkowity	1	15	-
	min_samples_split	całkowity	2	19	-
	min_samples_leaf	całkowity	1	19	-
RandomForestClassifier					
<i>(Las losowy)</i>	n_estimators	całkowity	1	2000	-
	bootstrap	dyskretny	-	-	['True', 'False']
	max_features	numeryczny	0	1	co 0.01
	criterion	dyskretny	-	-	['gini', 'entropy']
	max_depth	całkowity	1	15	-
	min_samples_split	całkowity	2	19	-
	min_samples_leaf	całkowity	1	19	-
ElasticNet					
<i>(Sieć elastyczna)</i>	alpha	numeryczny	0	1	co 0.001
	l1_ratio	numeryczny	2^{-10}	2^0	potęga co 0.01

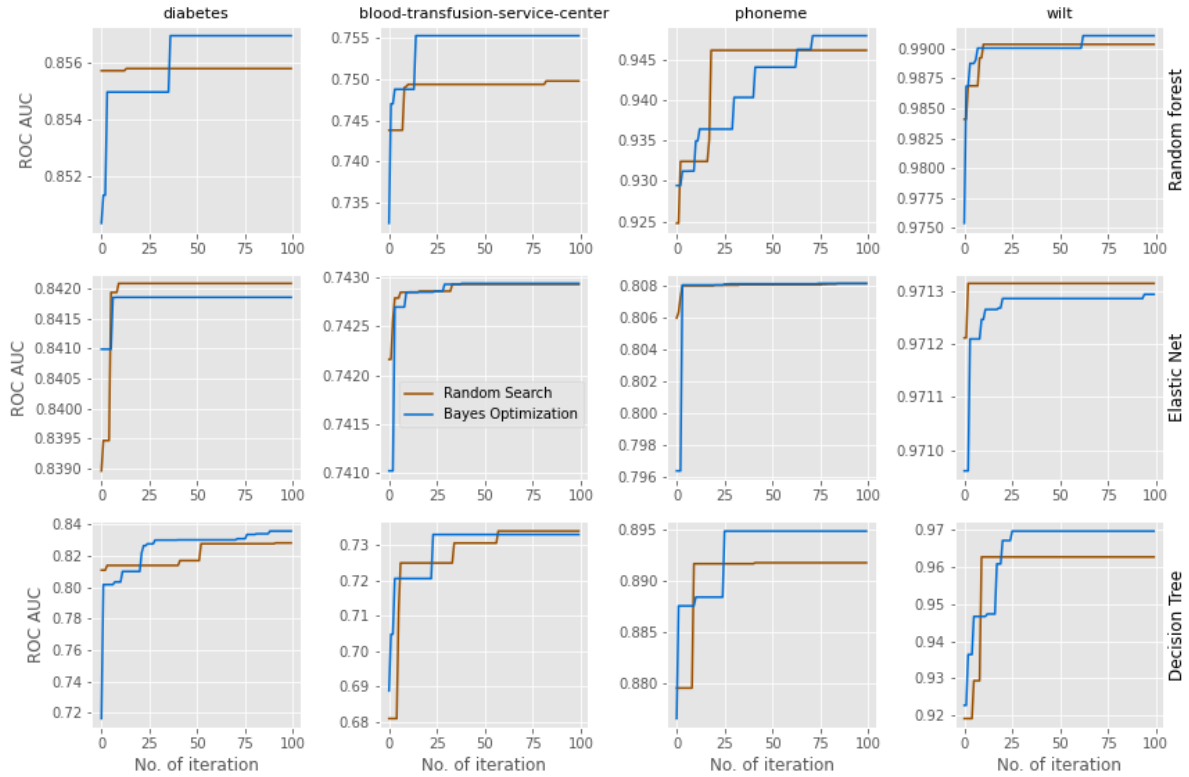
Wartości zostały dobrane, wzorując się na [1]. Niektóre wartości numeryczne poddawaliśmy tranformacji przez funkcję, co zostało odnotowane w kolumnie *Wartości*. W przypadku hiperparametrów dyskretnych, w tejże kolumnie umieściliśmy rozważane możliwe wartości hiperparametru.

Metryka, którą wybraliśmy do badania tunowalności, to ROC AUC.

3 Wyniki

3.1 Stabilność optymalizacji

Pierwszą badaną kwestią dotyczyła tego, ile iteracji każdej metody potrzebujemy, żeby uzyskać stabilne wyniki optymalizacji. Sprawdziliśmy to, zapisując w trakcie treningu, po każdej iteracji, dla każdego zbioru danych i każdego algorytmu najlepszy dotychczasowy wynik. Rysunek 1 przedstawia wyniki. Choć występowały różnice pomiędzy zbiorami danych, i algorytmami, w większości przypadków dla metody Random Search już od około 20 iteracji ROC AUC nie zwiększało się znacząco. Dla optymalizacji bayesowskiej stabilizacja następowała w granicach 25 iteracji. Wyjątkiem był model Random Forest na zbiorze *phoneme*, gdzie przez 70 iteracji metryka regularnie się poprawiała.



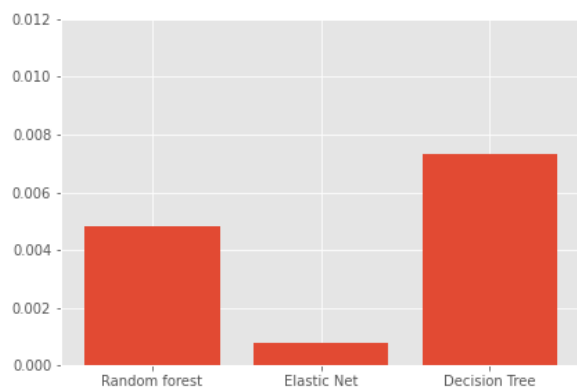
Rysunek 1: Najlepsze dotychczasowe ROC AUC dla poszczególnych modeli i zbiorów danych. Brązowe linie przedstawiają wyniki dla metody Random Search, niebieskie dla Bayes Search.

3.2 Tunowalność poszczególnych algorytmów

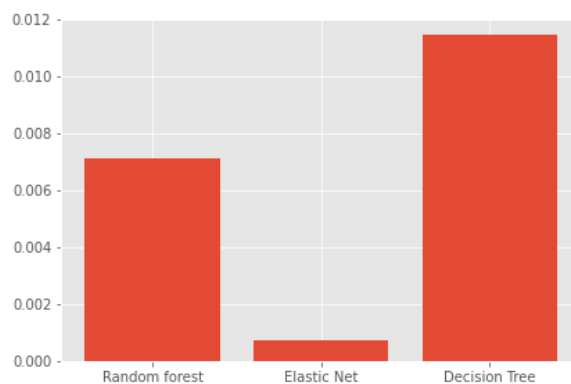
Kolejnym obszarem była tunowalność wybranych przez nas algorytmów. Dysponując wynikami historii treningu, dla każdego algorytmu znaleźliśmy zestaw hiperparametrów, który dawał najlepszą średnią wartość metryki na rozważanych zbiorach danych, nazywany przez autorów artykułu [1] optymalnym defaultem. Następnie porównaliśmy najlepszą uzyskaną wartość ROC AUC dla danego zestawu, z wynikiem jaki uzyskał model o optymalnych parametrach. Na rysunku 2 została przedstawiona średnia tunowalność poszczególnych algorytmów na testowanych przez nas zbiorach danych. Są to bardzo niskie wartości, najwyższa, uzyskana przez Decision Tree przy Bayes Searchu, ledwo przekracza 0.01.

Tunowalność tego algorytmu, przynajmniej na badanych przez nas zbiorach danych jest, wyższa niż pozostałych. Najniższa tunowalność (bliska 0) wyszła dla modelu ElasticNet. Najprawdopodobniej miała na to wpływ niewielka liczba optymalizowanych parametrów (2). Natomiast w przypadku Decision Tree i Random Forest liczba hiperparametrów była zbliżona, a średnia tunowalność okazała się wyższa dla modelu Decision Tree. Można zauważyć, że metoda optymalizacji bayesowskiej pozwoliła lepiej poprawić jakość algorytmów.

Zbadaliśmy również, czy wystąpiły znaczące różnice pomiędzy zbiorami danych. Wykresy zostały przedstawione na rysunkach 3 i 4. Widać, że algorytm Decision Tree zdecydowanie najwyższą tunowalność uzyskał na zbiorze *diabetes* (id= 37), a dla dwóch datasetów uzyskał gorszy wynik, niż Random Forest.



(a) Random Search



(b) Bayes Search

Rysunek 2: Tunowalność testowanych algorytmów w porównaniu do znalezionych optymalnych defaultów

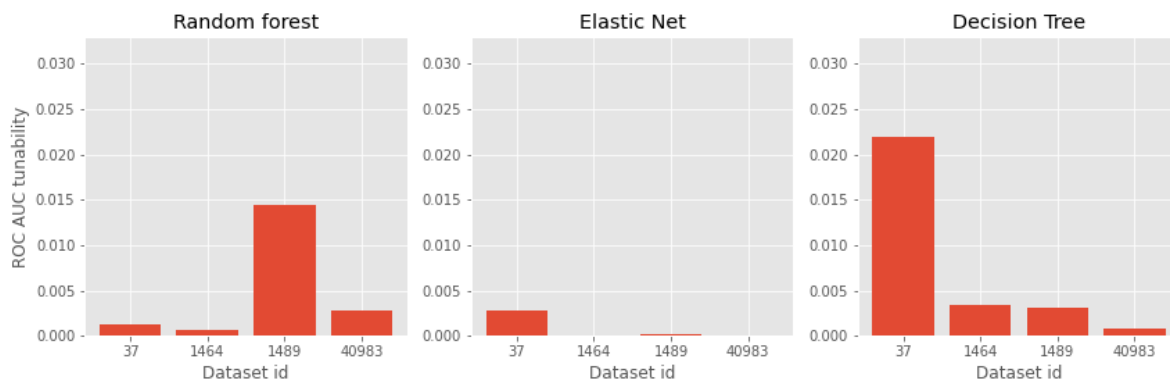
4 Podsumowanie

Przeprowadzony przez nas eksperyment pozwolił ustalić, że rozważane metody samplingu mogą dawać stabilne rezultaty po dość niewielkiej liczbie iteracji, odpowiednio 20 dla Random Search i 25 dla Bayes Search. Wyniki sugerują również, że algorytmy Decision Tree i Random Forest są bardziej tunowalne niż Elastic Net. Należy jednak pamiętać, że optymalizowaliśmy dla nich więcej hiperparametrów, a także zbadaliśmy to na jedynie 4 zbiorach danych. Wykonanie większej liczby iteracji, czy zdefiniowanie innej przestrzeni poszukiwań dla algorytmów również mogłoby zmienić rezultaty.

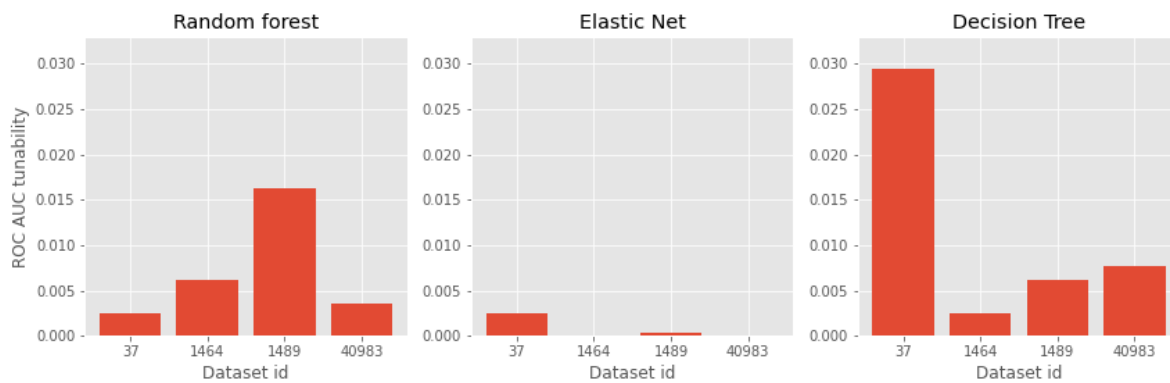
Literatura

- [1] Philipp Probst, Bernd Bischl, and Anne-Laure Boulesteix. Tunability: Importance of hyperparameters of machine learning algorithms, 2018.

A Dodatek



Rysunek 3: Tunowalność testowanych algorytmów dla metody Random Search z podziałem na zbiór danych.



Rysunek 4: Tunowalność testowanych algorytmów dla metody Bayes Search z podziałem na zbiór danych.