Badanie tunowalności hiperparametrów – raport

Mikołaj Twardowski, Maciej Turczyński, Mateusz Kiszka

Listopad 2025

1 Wprowadzenie

Celem projektu było przetestowanie tunowalności 3 wybranych algorytmów uczenia maszynowego: *RandomForestClassifier*, *XGBoost* i *SGDClassifier*. W ramach eksperymentu wykorzystaliśmy dwie różne strategie próbkowania na czeterech zbiorach danych. Analiza ta miała na celu nie tylko wyznaczenie nowych, optymalnych konfiguracji domyślnych dla tych algorytmów, lecz także zbadanie wpływu metody próbkowania, i wielkości zbioru danych na tunowalność.

2 Metodyka

2.1 Zbiory danych

Prace nad tym projektem rozpoczęliśmy od losowego wybrania 4 zbiorów danych z platformy OpenML, i pobrania ich przy użyciu funkcji *get_datasets()*. Z uwagi na badane przez nas metody uczenia maszynowego, postawiliśmy na zbiory danych, w których zmienna celu jest zmienną binarną. Wybór kilku różnych zbiorów był celowy, ponieważ naszym zadaniem jest badanie domyślnych wartości hiperparametrów, takich które będą odpowiednie dla szerokiej klasy problemów.

2.2 Przestrzeń hiperparametrów

Po wybraniu algorytmów do analizy, dla każdego z nich przygotowaliśmy szeroką przestrzeń przeszukiwania hiperparametrów, która motywowana była stosowanymi w literaturze zakresami [1]. Celem było pokrycie regionów, w których spodziewaliśmy się znaleźć optymalne wartości. Rozważane zakresy hiperparametrów przedstawia tabela 1.

Algorytm	**Hiperparametr**	**Zakres**
Random Forest	`n_estimators`	randint(2, 1000)
	`max_depth`	randint(2, 32)
	`min_samples_split`	randint(2, 32)
XGBoost	`n_estimators`	randint(2, 2000)
	`max_depth`	randint(1, 15)
	`learning_rate`	loguniform(0.001, 1.0)
	`subsample`	uniform(0.1, 0.9)
SGD Classifier	`sgd_alpha`	loguniform(1e-6, 1e-1)
	`sgd_l1_ratio`	uniform(0.0, 1.0)
	`sgd_penalty`	['l2', 'l1', 'elasticnet']

Tabela 1: Przestrzeń hiperparametrów użyta w eksperymencie.

2.3 Opis eksperymentu

Główna procedura eksperymentalna została zaimplementowana w funkcji *run_experiment()*. Dla każdego z 4 zbiorów danych i 3 algorytmów zastosowaliśmy dwie techniki próbkowania: Random Search i Bayes Search. Przy ich pomocy przeszukiwaliśmy zdefiniowane przestrzenie hiperparametrów w poszukiwaniu ich optymalnej kombinacji.

Dla każdego uruchomienia procedury, wspomniane metody przeprowadziły 40 iteracji. W każdej z nich wydajność danej konfiguracji ewaluowaliśmy za pomocą miary AUC, szacowanej przy użyciu 2-krotnego stratified CV. Dodatkowo, w celu rozszerzenia badania, cała procedura była powtarzana dla podzbiorów danych o rozmiarach 25%, 50%, 75% i 100% oryginalnego zbioru.

2.4 Badanie tunowalności

Analizę tunowalności oparliśmy na podejściu Probst i in. (2019) [1]. Najpierw wyznaczyliśmy nową optymalną konfigurację domyślną θ^* dla każdego algorytmu. W tym celu, aby zapewnić spójną bazę porównawczą, wykorzystano wyłącznie wyniki uzyskane metodą Random Search. Zapewniło to, że punkt odniesienia został wyznaczony na podstawie próbkowania z tej samej, ustalonej siatki hiperparametrów, natomiast konfiguracja domyślna została zdefiniowana jako ta z najwyższym średnim AUC, uśrednionym po wszystkich zbiorach danych. Następnie wykorzystaliśmy θ^* jako punkt odniesienia do zdefiniowania następujących dwóch miar:

- **Score Difference** będąca różnicą między wynikiem AUC konfiguracji domyślnej, a wynikiem AUC wszystkich innych konfiguracji. Ujemne wartości tej miary oznaczają konfiguracje lepsze niż domyślna.
- **Tunability** będąca różnicą między najlepszym wynikiem AUC osiągniętym na danym zbiorze, a wynikiem konfiguracji domyślnej. Pozwala ona zmierzyć potencjalny zysk wynikający ze strojenia hiperparametrów.

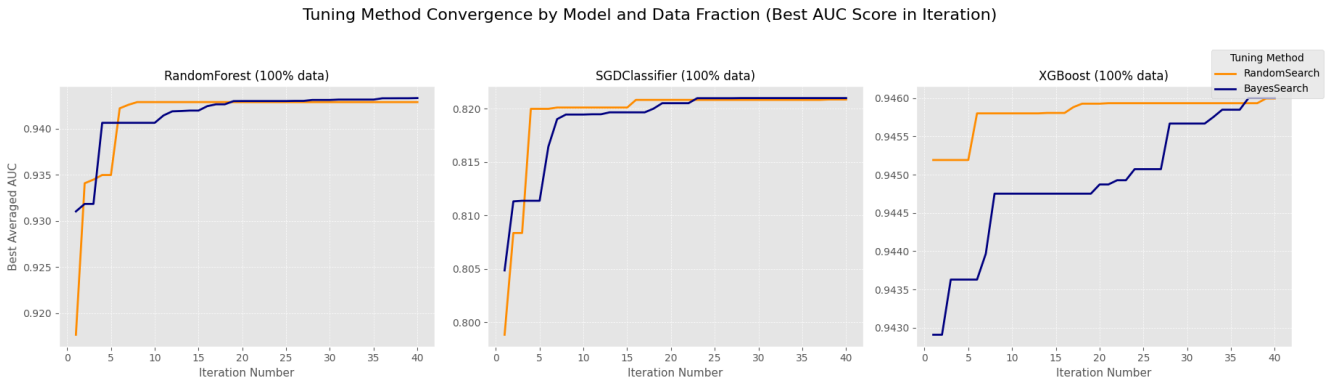
3 Wyniki

Analizę wyników naszego eksperymentu podzieliliśmy na cztery kluczowe obszary: zbieżność różnych metod optymalizacji, tunowalność algorytmów, wpływu użytej metody próbkowania na uzyskiwane wyniki oraz wielkości zbiorów danych użytych do trenowania modeli.

3.1 Analiza zbieżności

Pierwszym krokiem była próba odpowiedzi na pytanie, jak szybko każda z technik próbkowania osiąga stabilne wyniki. Na podstawie poniższego wykresu zbieżności (rysunek 1), można wyciągnąć następujące wnioski:

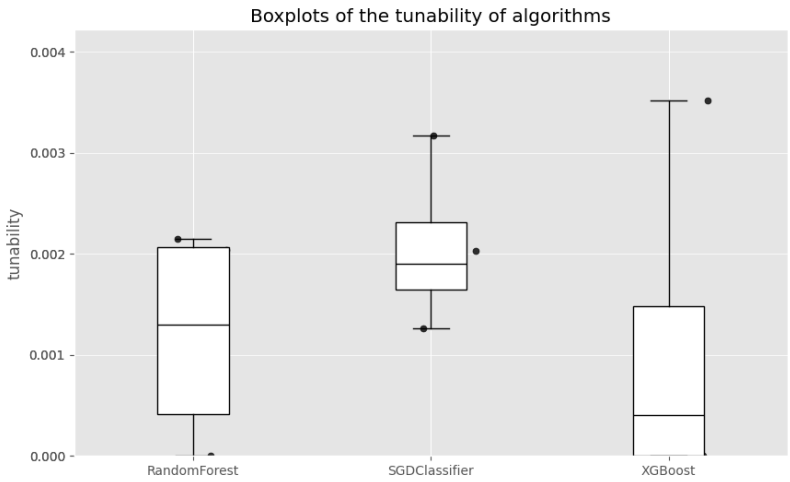
- dla modeli RandomForest oraz SGDClassifier optymalizacja bayesowska znajdowała lepszy wynik początkowy i stabilizowała się na wysokim poziomie AUC po około $15 - 20$ iteracjach, natomiast przeszukiwanie losowe szybko poprawiało początkowe gorsze wyniki i uzyskiwało stabilność nieco szybciej;
- dla modelu XGBoost sytuacja była zupełnie inna: RandomSearch niemal od początku znajdował konfiguracje o znakomitym wyniku i osiągał stabilność także po $15 - 20$ iteracjach, a BayesSearch startował z niższego poziomu i potrzebował koło 35 iteracji, by uzyskać zbliżone wyniki do tej pierwszej metody.



Rysunek 1: Zbieżność optymalizacji dla obydwu metod przeszukiwania

3.2 Tunowalność algorytmów

Zgodnie z metodyką przedstawioną w sekcji 2, tunowalność algorytmów została zbadana poprzez wyznaczenie nowej, domyślnej konfiguracji hiperparametrów. Pozwala to ocenić ogólną wrażliwość modelu na zmianę parametrów.



Rysunek 2: Tunowalność algorytmów dla pełnego zbioru danych

Powyższy wykres pudełkowy (rysunek 2) przedstawia uzyskaną tunowalność algorytmów. Na jego podstawie możemy zauważyć, że:

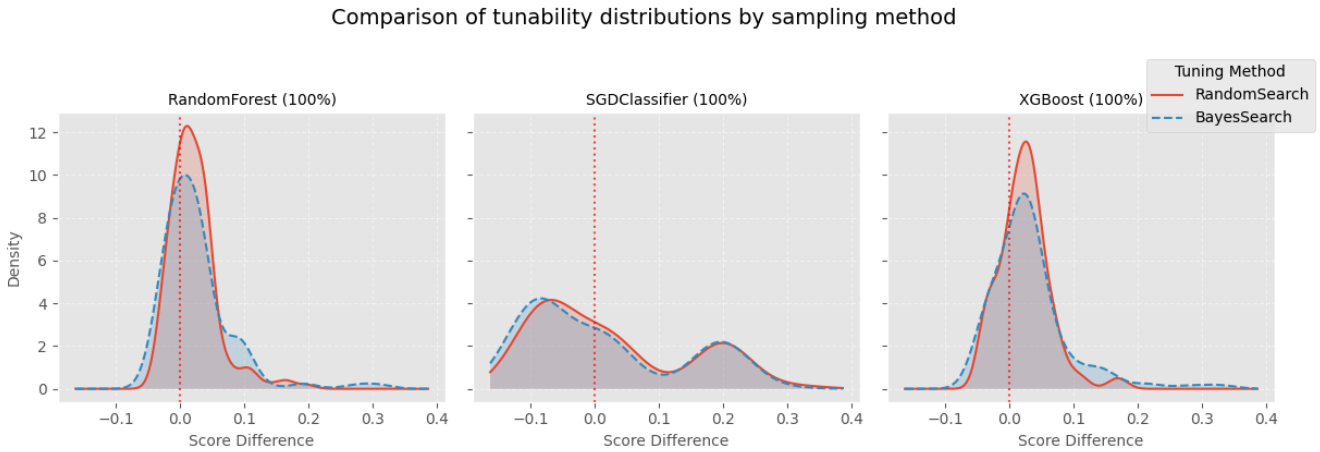
- SGDClassifier wykazuje najwyższą medianę tunowalności, co oznacza, że jego nowa konfiguracja domyślna jest przeciętnie najdalsza od optymalnej wartości na poszczególnych zbiorach danych;
- XGBoost charakteryzuje się najniższą medianą, lecz jednocześnie największym rozstępem międzykwartylowym. Świadczy to, że istnieją zbiory danych, na których dopasowanie tego modelu może przynieść największy zysk;
- RandomForest pod względem uzyskanych wyników plasuje się pomiędzy poprzednimi modelami w obydwu badanych cechach.

3.3 Wpływ metody próbkowania

Oprócz wcześniejszych analiz, sprawdziliśmy jaki wpływ na wnioski dotyczące tunowalności ma wybór metody próbkowania. W tym celu porównaliśmy rozkłady różnic wyników uzyskane przez obydwie badane metody zwizualizowane za pomocą wykresów KDE (Kernel Density Estimation). Analizując taką wizualizację (rysunek 3) można wyciągnąć następujące wnioski:

- główne garby gęstości dla modelów RandomForest oraz XGBoost są zlokalizowane w pobliżu 0 i wyglądają podobnie dla RandomSearch i BayesSearch. Oznacza to, że ciężko było znaleźć konfiguracje znacznie lepsze czy gorsze;
- dla SGDClassifier możemy zauważyć, że funkcja gęstości ma rozkład bimodalny z modami po obu stronach zera. Świadczy to o fakcie, że istniały grupy konfiguracji, które były zarówno dużo gorsze, jak i dużo lepsze od defaulta.

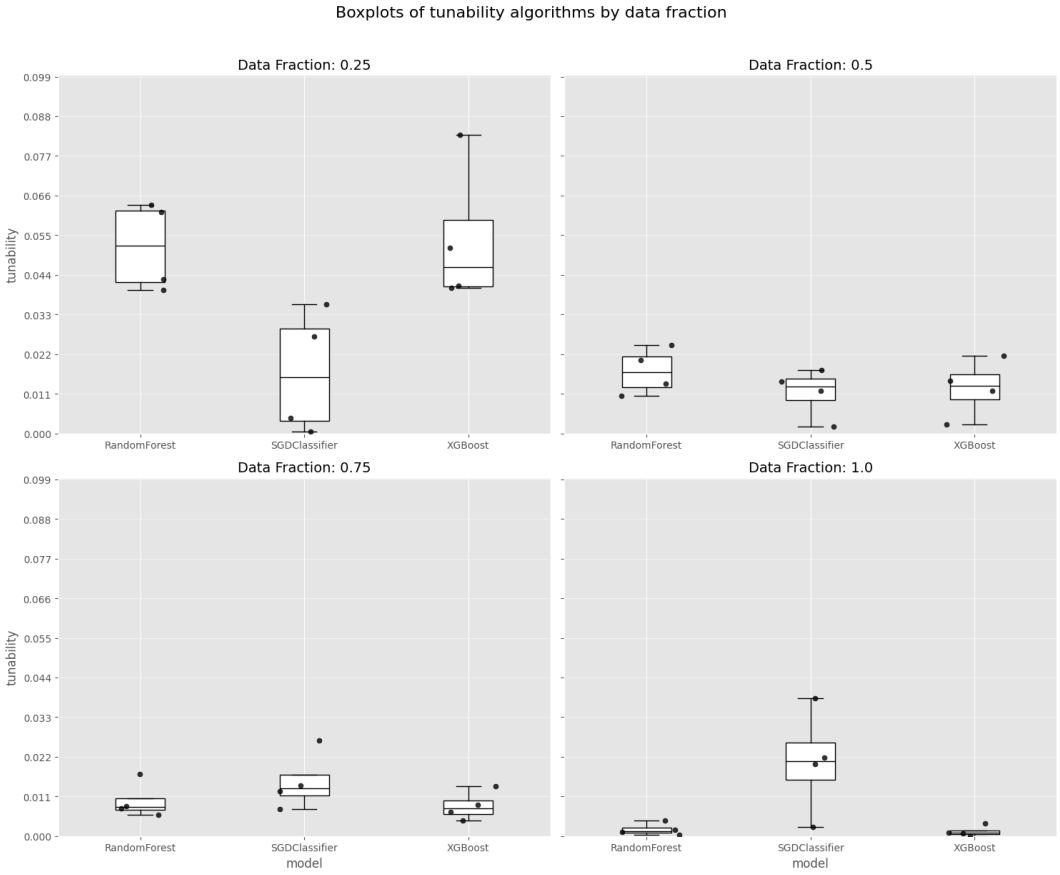
Uzyskane wyniki nie wskazują na to, że któraś z metod próbkowania jest lepsza, natomiast utwierdzają w tezie, że SGDClassifier jest algorytmem o największej możliwości tunowalności.



Rysunek 3: Porównanie rozkładów różnic wyników według metody próbkowania

3.4 Analiza wpływu wielkości zbioru danych

W ramach rozszerzenia eksperymentu zbadaliśmy również, jak na tunowalność wpływa wielkość rozważanych zbiorów danych. W tym celu wykorzystaliśmy podzbiory stanowiące 25%, 50%, 75% oraz 100% oryginalnych danych. Poniższy wykres skrzynkowy (rysunek 4) wskazuje na pewną tendencję, że tunability raczej maleje wraz ze wzrostem wielkości danych. Na mniejszych frakcjach mediana miary tunability jest największa, a rozkład najszerszy. Oznacza to, że przy mniejszej ilości danych modele są bardziej wrażliwe na dobór hiperparametrów, a znalezienie konfiguracji optymalnej przynosi największy zysk. Od tego wniosku odbiegają nieco wyniki dla modelu SGDClassifier, ale jest to spójne z jego charakterystyką opisaną już w poprzednich punktach.



Rysunek 4: Rozkład wpływu wielkości zbioru danych na tunowalność

4 Wnioski

Przeprowadzony eksperyment, mający na celu zbadanie tunowalności algorytmów *RandomForestClassifier*, *XGBoost* i *SGDClassifier*, pozwala na sformułowanie następujących wniosków:

- algorytmem o najwyższej tunowalności okazał się SGDClassifier. Wykazał on najwyższą medianę różnicy między wynikiem optymalnym a domyślnym (rysunek 2), co oznacza, że jego wydajność jest najbardziej wrażliwa na dobór hiperparametrów. RandomForest i XGBoost charakteryzują się niższą tunowalnością, co świadczy o solidności wyznaczonych dla nich konfiguracji domyślnych;
- Analiza wpływu wielkości zbioru danych wykazała wyraźną tendencję: tunowalność maleje wraz ze wzrostem liczby obserwacji (rysunek 4). Na mniejszych frakcjach danych (między innymi 25%) modele są bardziej wrażliwe na strojenie, a potencjalny zysk z optymalizacji jest największy.
- W kwestii metod próbkowania, optymalizacja Bayesowska była zbieżna szybciej dla RandomForest i SGDClassifier. Niespodziewanie, dla XGBoost to Random Search szybciej zidentyfikował region bliski optimum (rysunek 1). Mimo tych różnic w zbieżności, obie metody prowadziły do spójnych wniosków dotyczących ogólnej charakterystyki tunowalności każdego z algorytmów (rysunek 3).

Podsumowując, eksperyment potwierdził, że tunowalność jest silnie zależna od algorytmu. Wybór odpowiedniej konfiguracji jest krytyczny dla SGDClassifier, ale mniej istotny dla RandomForest i XGBoost, szczególnie gdy dysponujemy dużymi zbiorami danych.

Literatura

[1] P. Probst, B. Bischl, A-L. Boulesteix, (2019), *Tunability: Importance of Hyperparameters of Machine Learning Algorithms*, `https://jmlr.org/papers/volume20/18-444/18-444.pdf`