

WYDZIAŁ MATEMATYKI I NAUK INFORMACYJNYCH  
POLITECHNIKA WARSZAWSKA

---

Badanie tunowalności wybranych algorytmów uczenia  
maszynowego

---

AUTOMATYCZNE UCZENIE MASZYNOWE  
PROJEKT 1

Maciej Momot, Mikołaj Rowicki, Krzysztof Tkaczyk

13 listopada 2024

# 1 Wstęp

W niniejszej pracy przedstawiamy rezultaty badania tunowalności trzech algorytmów uczenia maszynowego: Random Forest, XGBoost oraz Decision Tree. Wymienione algorytmy wykorzystaliśmy do rozwiązywania zadania klasyfikacji binarnej dla 4 różnych zbiorów danych. Szukaliśmy najlepszych wartości hiperparametrów za pomocą dwóch metod samplingu: *RandomSearch* i *BayesSearch*.

## 1.1 Zbiory danych

Modele budowaliśmy na podstawie poniższych zbiorów danych:

- Airline Passenger Satisfaction[1]
- Weather Dataset[6]
- Banking Dataset[2]
- Mushroom Dataset[3]

Liczbę obserwacji w poszczególnych zbiorach danych zredukowaliśmy do 5000 z uwzględnieniem rozkładu zmiennej celu (miał być taki sam jak w oryginalnych zbiorach).

## 1.2 Metryka

Jakość modeli ocenialiśmy za pomocą metryki *ROC-AUC*[5].

# 2 Stabilność rezultatów otrzymywanych za pomocą różnych metod samplingu

Z powodu wydajności, ustawiliśmy liczbę iteracji na 1000 dla *RandomSearch* oraz na 100 dla *BayesSearch*. Metoda *BayesSearch* potrzebowała znacznie mniej powtórzeń by osiągnąć wyniki nawet lepsze niż *RandomSearch*. Obie metody osiągały stabilne wyniki po przejściu ok. 60% iteracji. Rysunek 2 przedstawia opisane zależności.

# 3 Określenie zakresów hiperparametrów dla poszczególnych modeli

Do badania tunowalności naszych modeli musieliśmy wybrać zakresy hiperparametrów, które naszym zdaniem warto jest modyfikować podczas procesów: *RandomSearch* oraz *BayesSearch*. Motywowani artykułem "Tunability: Importance of Hyperparameters of Machine Learning Algorithms"[4] początkowo dostosowaliśmy dość szerokie zakresy dla naszych hiperparametrów, ale po ponownym ich zwalidowaniu oraz w oparciu o charakterystykę naszych zbiorów danych, ostatecznie zdecydowaliśmy się wybrać przedziały przedstawione w tabeli 1.

W przypadku modelu Decision Tree hiperparametr **criterion** wybieraliśmy spośród dwóch możliwych wartości - gini oraz entropy. Przedstawione zakresy zostały ustalone dla naszych

zbiorów danych dokładnie tak, aby uniknąć niepotrzebnie straconych iteracji podczas procesów optymalizacji. Generowałoby to bowiem znacząco niższe wyniki z powodu wykorzystywania nieadekwatnych wartości hiperparametrów do postawionego problemu.

## 4 Tunowalność poszczególnych algorytmów

Nasze badania rozpoczęliśmy od znalezienia defaultowych wartości hiperparametrów dla poszczególnych algorytmów optymalizacji. W tym celu wykorzystaliśmy metodę *RandomSearch* z ustalonym ziarnem losowości dla wszystkich modeli. Chcieliśmy zapewnić, że każdy zbiór hiperparametrów zostanie wykorzystany dla wszystkich zbiorów danych. Zastosowaliśmy pięciokrotną krosvalidację i ustawiliśmy 1000 iteracji poszukiwania optymalnych hiperparametrów dla każdego algorytmu. Po przejściu przez ten proces dla poszczególnych modeli, historię wyników zapisywaliśmy do plików csv, aby łatwiej operować nimi przy wyznaczaniu tunowalności. W efekcie otrzymaliśmy nasze optymalnie najlepsze zestawy hiperparametrów modeli dla wszystkich zbiorów razem wziętych. Do wyliczenia tych wartości stosowaliśmy średnią, ze względu na małą liczbę zbiorów danych. Metryką w naszym badaniu było *ROC-AUC*.

Z historii metody *RandomSearch* znaleźliśmy również najlepsze zestawy hiperparametrów dla poszczególnych zbiorów danych i mając tę wiedzę, mogliśmy wyznaczyć tunowalność algorytmów.

Przy optymalizacji bayesowskiej (*BayesSearch*) (z pięciokrotną krossvalidacją i ustalonym ziarnem losowości) trzymaliśmy się już wyznaczonych przez nas defaultów. Pozostało nam znaleźć najlepsze hiperparametry dla poszczególnych zbiorów danych. Ze względu na długie przetwarzanie tej metody, zdecydowaliśmy się na 100 iteracji dla każdego algorytmu i zbioru. Historię zapisywaliśmy analogicznie jak dla *RandomSearch* i znaleźliśmy interesujące nas najlepsze zestawy hiperparametrów dla każdego zbioru, por. tabela 2.

Tunowalność liczyliśmy ze wzoru:

$$tunability = \frac{best\_score - default\_score}{default\_score} \cdot 100\%$$

Wyniki porównujące tunowalność uzyskaną obiema metodami przedstawia tabela 3.

Zauważmy, że największy potencjał tunowalności zdaje się mieć algorytm Decision Tree, gdzie tunowalność potrafi przekroczyć 2% przy wykorzystaniu obydwu metod optymalizacji. To bardzo dobry wynik na tle innych algorytmów. XGBoost wykazał niską tunowalność na naszych zbiorach danych dla obydwu metod. Nawet na jednym zbiorze tunowalność wynosiła 0%, co oznacza, że defaultowe parametry okazały się dla tego zbioru najlepsze w *RandomSearch*. W przypadku Random Foresta wyniki nie przekraczają 0,3%.

## 5 Tunowalność poszczególnych hiperparametrów

W ramach naszego projektu przeprowadziliśmy również badanie tunowalności poszczególnych hiperparametrów stosowanych przez nas modeli. Przy obliczaniu tej wartości wykorzystaliśmy defaultowe konfiguracje hiperparametrów wyznaczone metodą *RandomSearch* dla 1000 iteracji (wartości zdefiniowane w tabeli 2). Następnie dla każdego modelu ustalaliśmy jeden z 4 hiperparametrów i zmienialiśmy jego wartość w obrębie zakresów zdefiniowanych w tabeli 1. W tym celu posłużyliśmy się metodą *RandomSearch* wykorzystującą 200 iteracji oraz

algorytmem *BayesSearch*, w którym zastosowaliśmy 50 iteracji. Ten proces powtarzaliśmy dla każdego z hiperparametrów, każdego modelu i każdego zbioru danych. Aby zapewnić czytelność wyników, w tym przypadku zdecydowaliśmy się na tunowalność zagregowaną ze względu na zbiory danych. Wyliczyliśmy ją jako średnią po wartościach dla wszystkich zbiorów danych ze wzoru:

$$\text{tunability} = \text{mean} \left( \max \left\{ \frac{\text{best\_score} - \text{default\_score}}{\text{default\_score}} \cdot 100\%, 0\% \right\} \right),$$

gdzie przez *best\_score* rozumiemy najlepszy wynik metryki *ROC-AUC* uzyskany daną metodą optymalizacji, przy założeniu, że optymalizowany był tylko jeden hiperparametr. Z kolei maksimum, zdefiniowane tak jak powyżej, działa równoważnie z dodaniem do historii tuningu defaultowej konfiguracji, to znaczy zabezpiecza nas przed ujemnymi wartościami tunowalności. Wyniki zostały przez nas zapisane w Tabeli 4.

Wyniki badania pozwalają zauważyć konkretne hiperparametry, których optymalizacja może przynieść nam szczególne korzyści. Są to przede wszystkim parametry *max\_depth* i *min\_samples\_leaf*. Niezależnie od metody tunowania parametry te osiągnęły dość wysokie wartości względnej tunowalności. Istnieją również hiperparametry, które niekoniecznie warto optymalizować, na przykład parametry *max\_depth* i *min\_child\_weight* w modelu XGBoost. Ponadto, na podstawie eksperymentu można zauważyć, że hiperparametry modeli takich jak Random Forest oraz XGBoost są bardziej podatne na optymalizację wykorzystującą *RandomSearch*, natomiast hiperparametry modelu Decision Tree (zdecydowanie najprostszego spośród rozważanych przez nas) osiągają największą tunowalność za pomocą optymalizacji *BayesSearch*. Trudno na tej podstawie wyciągać ogólne wnioski na temat tego, która z metod optymalizacji jest lepsza. Zauważalne jest natomiast, że w obrębie każdego modelu istnieje szczególnie preferowana metoda optymalizacji.

## 6 Wpływ techniki losowania na tunowalności algorytmów i hiperparametrów

Na podstawie wyników zawartych w Tabeli 3 możemy stwierdzić, że spośród 12 rozważanych par zbiór - model w pięciu przypadkach różnica względnej tunowalności między metodami optymalizacji jest większa niż 0,1 %. W dwóch przypadkach te wartości są równe (z dokładnością do trzech miejsc po przecinku). Co jednak szczególnie ważne, niezależnie od wybranej metody, najbardziej tunowalnym modelem okazuje się Decision Tree, zaś najmniej XGBoost (dla dwóch zbiorów danych) i Random Forest (dla jednego zbioru danych). W przypadku jednego zbioru danych (Weather [6]), o tym, który model jest najmniej tunowalny, faktycznie decyduje wybór metody optymalizacji.

Aby dokładniej przyjrzeć się tym różnicom, postanowiliśmy przeprowadzić test statystyczny Wilcozona dla wyników metryki *ROC-AUC* uzyskanych w poszczególnych iteracjach. Porównaliśmy w ten sposób sto pierwszych iteracji metody *RandomSearch* ze stoma iteracjami optymalizacji bayesowskiej. Przyjęliśmy poziom istotności testu  $\alpha = 0.01$ . W dziewięciu na 12 przypadków okazało się, że istnieją istotne różnice między próbkami. Co istotne, wszystkie z pozostałych trzech par zbiór - model, dla których nie wykazano istotnych różnic, dotyczyły modelu XGBoost.

W przypadku tunowania hiperparametrów tabela 4 pokazuje, że największe różnice tunowalności między metodami występują dla algortmu XGBoost, zaś najmniejsze dla Decision Tree. Spośród 12 rozważanych par zbiorów - model w pięciu przypadkach różnica względnej tunowalności między metodami optymalizacji jest mniejsza niż 0,01 %. Jeżeli natomiast chodzi o trzy najbardziej podatne na tuning hiperparametry (`max_depth`, `min_samples_split` i `min_samples_leaf` w Decision Tree), to są one takie same dla obu metod optymalizacji. Podobnie jest z dwoma najmniej podatnymi na tuning hiperparametrami (`max_depth` i `min_child_weight` w XGBoost).

Zwróciliśmy również uwagę na sposób w jaki wybierane są kolejne zestawy parametrów przez obie metody samplingu. Wyniki przedstawione na rysunku 1 wskazują, że gdy chcemy przebadąć większy zakres przestrzeni parametrów powinniśmy wybrać metodę *RandomSearch*. Metoda *BayesSearch* zazwyczaj wybierała parametry w taki sposób, że kolejne wyniki tunowalności miały dość mały rozrzut oraz ale niekoniecznie były skupione wokół wyników optymalnych (defaultowych).

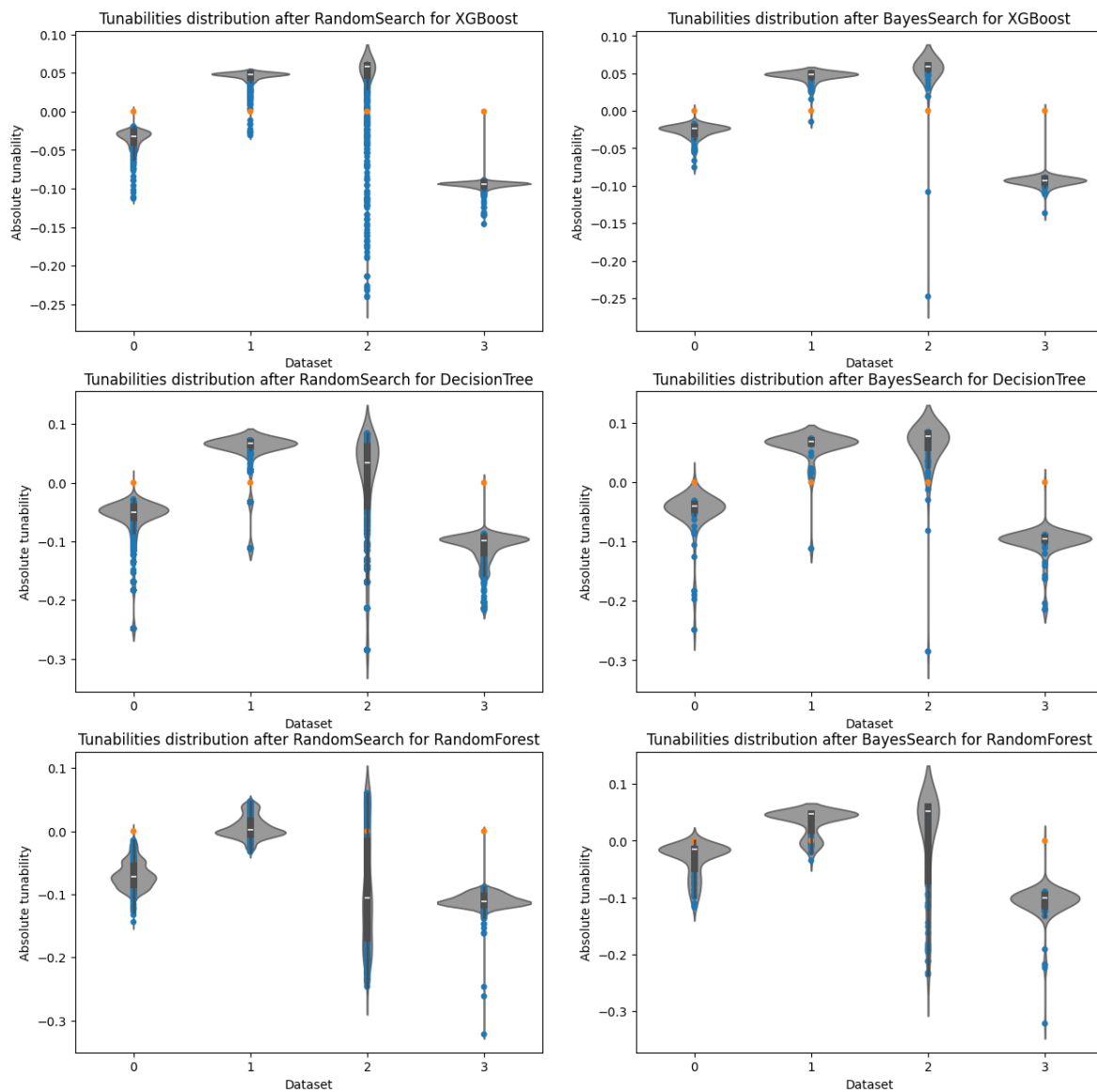
Podsumowując, między wynikami metryki *ROC-AUC* w poszczególnych iteracjach da się zauważyć istotne różnice między poszczególnymi metodami optymalizacji. W rezultacie nie przekładają się one jednak na kluczowe wnioski. Najbardziej tuningowalne algorytmy i hiperparametry pozostają takie same, niezależnie od sposobu przeszukiwania przestrzeni parametrów. Podobna zasada, z jednym wyjątkiem, dotyczy tych najmniej podatnych na tuning algorytmów i hiperparametrów.

## Bibliografia

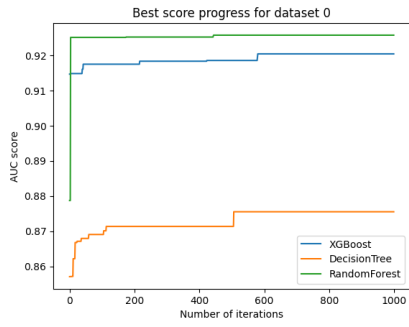
- [1] *Airline Passenger Satisfaction*. URL: <https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction>.
- [2] *Banking Dataset*. URL: <https://www.kaggle.com/datasets/prakharrathi25/banking-dataset-marketing-targets?select=train.csv>.
- [3] *Mushroom Dataset*. URL: <https://archive.ics.uci.edu/dataset/848/secondary+mushroom+dataset>.
- [4] Philipp Probst, Anne-Laure Boulesteix i Bernd Bischl. *Tunability: Importance of Hyperparameters of Machine Learning Algorithms*. URL: <https://jmlr.org/papers/volume20/18-444/18-444.pdf>.
- [5] *ROC-AUC score*. URL: [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc\\_auc\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html).
- [6] *Weather Dataset*. URL: [https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package?fbclid=IwZXh0bgNhZW0CMTEAAR2E70Y5b1gYNshFu6eIRsMpyNlpaBBKfvfZ4PIiaem\\_9FhTids3FY060V-bDXqPxQ](https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package?fbclid=IwZXh0bgNhZW0CMTEAAR2E70Y5b1gYNshFu6eIRsMpyNlpaBBKfvfZ4PIiaem_9FhTids3FY060V-bDXqPxQ).

# Aneks

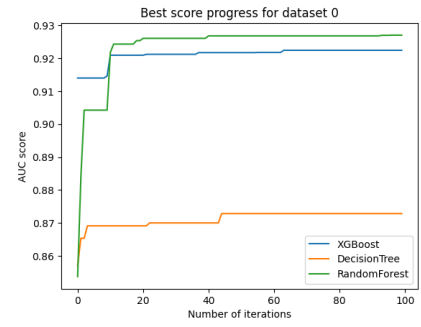
## Wykresy



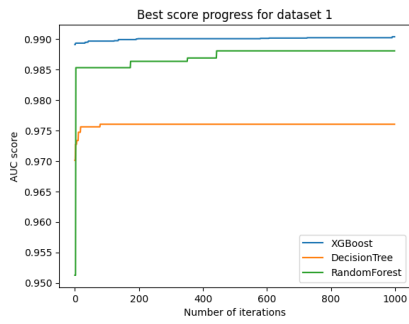
Rysunek 1: Wykresy przedstawiające rozkład tunowalności dla zestawów parametrów zbadanych w poszczególnych iteracjach wykonanych przez wybraną metodę z uwzględnieniem podziału na zbiory danych.



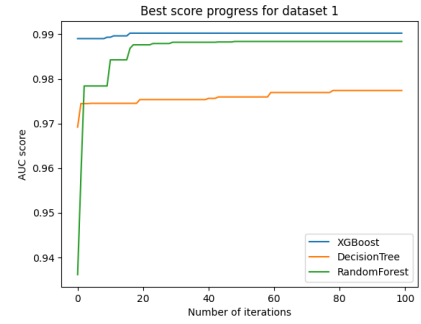
(a)



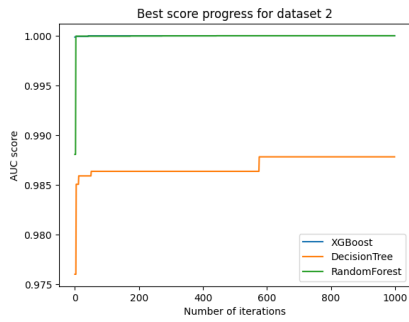
(b)



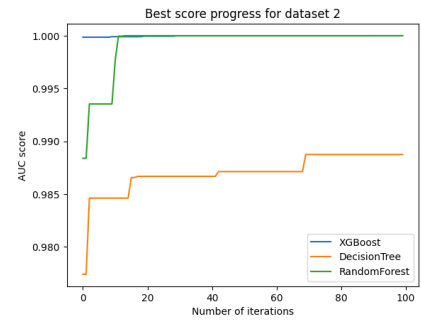
(c)



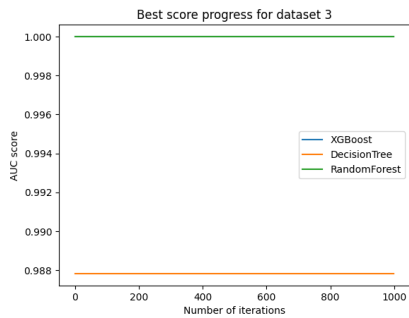
(d)



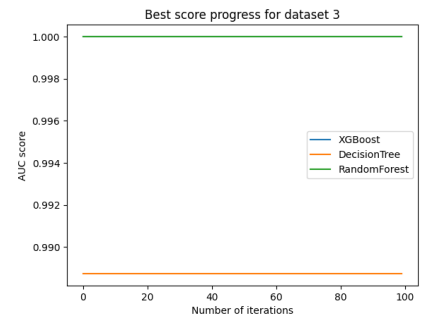
(e)



(f)



(g)



(h)

Rysunek 2: Wykresy a), c), e), g) przedstawiają zależność najlepszego wyniku  $ROC-AUC$  od numeru iteracji dla metody *RandomSearch*, a wykresy b), d), f), h) dla metody *BayesSearch*



## Tabele

Model	Parametr	Typ danych	Dolny zakres	Górny zakres
Decision Tree	max_depth	integer	1	30
	min_samples_split	integer	2	60
	criterion	discrete	-	-
	min_samples_leaf	integer	1	60
Random Forest	n_estimators	integer	100	500
	min_samples_leaf	integer	1	250
	max_samples	numeric	0.5	1
	max_features	numeric	$10^{-6}$	1
XGBoost	max_depth	integer	1	19
	min_child_weight	integer	0	19
	eta	numeric	0.01	0.11
	alpha	lognumeric	$10^{-4}$	10

Tabela 1: Zakresy hiperparametrów dla poszczególnych modeli.

Model	Parametry	Score
DecisionTree	criterion: gini max_depth: 17 min_samples_leaf: 10 min_samples_split: 58	0,904
RandomForest	max_features: 0,498 max_samples: 0,738 min_samples_leaf: 3 n_estimators: 478	0,94
XGBoost	alpha: 1,248 eta: 0,098 max_depth: 16 min_child_weight: 0	0,94

Tabela 2: Defaultowe wartości parametrów dla poszczególnych algorytmów optymalizacji wraz z wynikami (wartości zaokrąglone do 3 miejsca po przecinku).

Zbiór	Decision Tree		Random Forest		XGBoost	
	RS	BO	RS	BO	RS	BO
weather	<b>0,649%</b>	0,372%	0,058%	<b>0,197%</b>	0%	<b>0,219%</b>
flights	0,491%	<b>0,632%</b>	0,174%	<b>0,206%</b>	<b>0,027%</b>	0,012%
banking	1,166%	<b>1,260%</b>	0,003%	0,003%	0,001%	0,001%
mushrooms	<b>2,490%</b>	2,307%	<b>0,160%</b>	0,095%	<b>0,301%</b>	0,297%

Tabela 3: Porównanie tunowalności algorytmów metodą Bayesowskiej optymalizacji i Random Searchu (wartości zaokrąglone do 3 miejsca po przecinku).

Model	Parametr	Random Search	Bayes Search
Decision Tree	max_depth	<b>0.523%</b>	0.522%
	min_samples_split	0.215%	<b>0.217%</b>
	criterion	0.178%	<b>0.179%</b>
	min_samples_leaf	0.831%	<b>0.841%</b>
Random Forest	n_estimators	<b>0.073%</b>	0.066%
	min_samples_leaf	<b>0.101%</b>	0.069%
	max_samples	<b>0.077%</b>	0.065%
	max_features	<b>0.078%</b>	0.077%
XGBoost	max_depth	<b>0.046%</b>	0.022%
	min_child_weight	<b>0.046%</b>	0.001%
	eta	<b>0.120%</b>	0.052%
	alpha	0.092%	<b>0.182%</b>

Tabela 4: Średnie wyniki tunowalności dla poszczególnych hiperparametrów badanych modeli