

# PD1 - Tunowalność hiperparametrów

Zuzanna Kotłńska, Agata Kopyt

November 2023

## 1 Wstęp

W uczeniu maszynowym optymalizacja hiperparametrów modelu ma kluczowe znaczenie, jako że umożliwia kontrolowanie procesu trenowania tak, by uzyskać lepszą wydajność dla określonego problemu lub zbioru danych. Celem raportu jest przedstawienie wyników eksperymentów przeprowadzonych, by przeanalizować tunowalność hiperparametrów dla trzech różnych algorytmów uczenia maszynowego oraz czterech różnych zbiorów danych.

## 2 Eksperyment

### 2.1 Dane

Testy zostały przeprowadzone na zbiorach danych pochodzących z OpenML. Wszystkie służą do zadania klasyfikacji, co jest korzystniejsze dla tego eksperymentu, ponieważ daje to możliwość łatwiejszego porównywania wartości metryk między sobą. Wykorzystane zbiory to:

- kc2,
- fri-c2-500-25,
- breast-w,
- diabetes.

### 2.2 Modele

Na każdym zbiorze danych zostały wytrenowane trzy przedstawione niżej algorytmy uczenia maszynowego:

- Random Forest,
- Elastic Net,
- Decision Tree.

## 2.3 Losowanie hiperparametrów

Zestawy hiperparametrów zostały wybrane za pomocą dwóch różnych metod:

- Random Search - metoda oparta na wyborze punktów z rozkładu jednostajnego, w której wartości hiperparametrów są wybierane losowo z siatki,
- Bayes Optimization - metoda oparta na technice bayesowskiej, która wykorzystuje model probabilistyczny do przewidywania, które kombinacje hiperparametrów są najbardziej prawdopodobne do osiągnięcia optymalnej wydajności.

Zestawy hiperparametrów, wraz z zakresami ich wartości, zostały wybrane na podstawie przykładów przedstawionych w źródle. Poniższe tabele zawierają szczegółowe informacje na temat siatek hiperparametrów dla każdego z trzech testowanych modeli:

Hiperparametr	Typ	Wartości
n_estimators	integer	[1, 2000]
criterion	string	gini, entropy
max_depth	integer	[3, 10]
min_samples_split	integer	[2, 10]
min_samples_leaf	integer	[1, 10]
bootstrap	logical	True, False
max_samples	numeric	[0, 1]

Tabela 1: Zestaw hiperparametrów dla Random Forest

Hiperparametr	Typ	Wartości
alpha	numeric	$2^x$ dla $x \in [-10, 10]$
l1_ratio	numeric	[0, 1]

Tabela 2: Zestaw hiperparametrów dla Elastic Net

Hiperparametr	Typ	Wartości
max_depth	integer	[1, 30]
max_features	string	None, log2, sqrt
criterion	string	gini, entropy
splitter	string	best, random
min_samples_split	integer	[1, 60]
min_samples_leaf	integer	[1, 60]

Tabela 3: Zestaw hiperparametrów dla Decision Tree

## 2.4 Trening i ewaluacja

Dla każdego algorytmu uczenia maszynowego i każdego zbioru danych znalezione zostało 100 różnych zestawów hiperparametrów. Do ewaluacji wyników zastosowana została metryka ROC AUC. Poniżej przedstawiony jest wzór na tunowalność algorytmu, wykorzystany w eksperymencie.

$$d^{(j)} := |R^{(j)}(\theta^*) - R^{(j)}(\theta^{(j)*})|, \quad \text{for } j = 1, \dots, m. \quad (1)$$

gdzie:

$\theta^*$  - zestaw hiperparametrów, który daje średnio najlepszy wynik metryki ROC AUC na wszystkich zbiorach,

$\theta^{(j)*}$  - zestaw hiperparametrów, który daje najlepszy wynik metryki ROC AUC na  $j$ -tym zbiorze,

$R^{(j)}(\theta)$  - wartość metryki ROC AUC na  $j$ -tym zbiorze dla zestawu hiperparametrów  $\theta$ .

Warto zaznaczyć, że policzenie  $\theta^*$  ze względu na to, że wymaga tych samych kombinacji hiperparametrów dla każdego zbioru danych, jest możliwe tylko dla metody Random Search. Zatem w przypadku Bayes Optimization punktem odniesienia będzie także  $\theta^*$  wyznaczone metodą Random Search.

## 2.5 Wyniki

Poniższe tabele przedstawiają optymalne zestawy hiperparametrów i wartości metryki AUC w zależności od algorytmu uczenia maszynowego i zbioru danych.

		Random Forest	Elastic Net	Decision Tree
zbiór 1	theta	n_estimators=1124 min_samples_split=2 min_samples_leaf=2 max_samples=None max_depth=3 criterion=entropy bootstrap=True	alpha=512 l1_ratio=0.646	splitter=best min_samples_split=58 min_samples_leaf=57 max_features=None max_depth=19 criterion=gini
	AUC	0.855	0.863	0.849
zbiór 2	theta	n_estimators=259 min_samples_split=9 min_samples_leaf=1 max_samples=None max_depth=9 criterion=entropy bootstrap=True	alpha=2 l1_ratio=0.009	splitter=best min_samples_split=35 min_samples_leaf=19 max_features=None max_depth=23 criterion=gini
	AUC	0.928	0.717	0.878
zbiór 3	theta	n_estimators=1398 min_samples_split=6 min_samples_leaf=1 max_samples=None max_depth=9 criterion=gini bootstrap=False	alpha=2 l1_ratio=0.009	splitter=random min_samples_split=56 min_samples_leaf=12 max_features=None max_depth=5 criterion=entropy
	AUC	0.996	0.998	0.987
zbiór 4	theta	n_estimators=1784 min_samples_split=6 min_samples_leaf=8 max_samples=None max_depth=4 criterion=entropy bootstrap=True	alpha=0.001 l1_ratio=0.01	splitter=random min_samples_split=13 min_samples_leaf=17 max_features=None max_depth=29 criterion=gini
	AUC	0.834	0.829	0.801

Tabela 4: Hiperparametry i wartość AUC dla Random Search

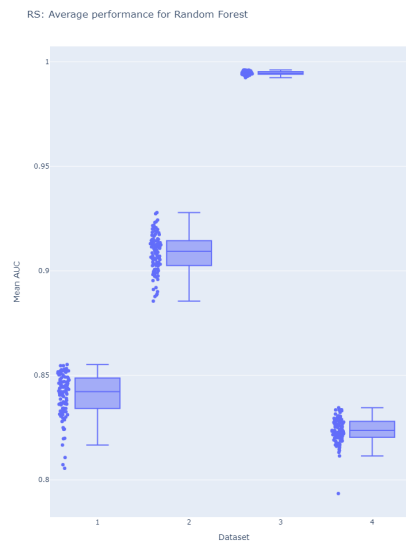
		Random Forest	Elastic Net	Decision Tree
zbiór 1	theta	n_estimators=1779 min_samples_split=8 min_samples_leaf=9 max_samples=None max_depth=3 criterion=entropy bootstrap=True	alpha=64 l1_ratio=0.999	splitter=best min_samples_split=21 min_samples_leaf=30 max_features=log2 max_depth=30 criterion=gini
	AUC	0.857	0.864	0.857
zbiór 2	theta	n_estimators=920 min_samples_split=3 min_samples_leaf=1 max_samples=None max_depth=10 criterion=gini bootstrap=False	alpha=0.031 l1_ratio=0.489	splitter=best min_samples_split=49 min_samples_leaf=19 max_features=None max_depth=22 criterion=gini
	AUC	0.930	0.714	0.886
zbiór 3	theta	n_estimators=697 min_samples_split=5 min_samples_leaf=1 max_samples=None max_depth=9 criterion=gini bootstrap=False	alpha=2 l1_ratio=0.0	splitter=best min_samples_split=17 min_samples_leaf=6 max_features=sqrt max_depth=19 criterion=gini
	AUC	0.996	0.998	0.988
zbiór 4	theta	n_estimators=577 min_samples_split=6 min_samples_leaf=7 max_samples=None max_depth=3 criterion=entropy bootstrap=True	alpha=0.015 l1_ratio=0.001	splitter=best min_samples_split=29 min_samples_leaf=43 max_features=None max_depth=4 criterion=entropy
	AUC	0.836	0.829	0.797

Tabela 5: Hiperparametry i wartość AUC dla Bayesian Optimization

	Random Forest	Elastic Net	Decision Tree
theta *	n_estimators=790 min_samples_split=6 min_samples_leaf=3 max_samples=None max_depth=5 criterion=entropy bootstrap=True	alpha=0.950 l1_ratio=0.031	splitter=best min_samples_split=34 min_samples_leaf=43 max_features=None max_depth=21 criterion=gini

Tabela 6: Najlepszy zestaw parametrów w zależności od algorytmu.

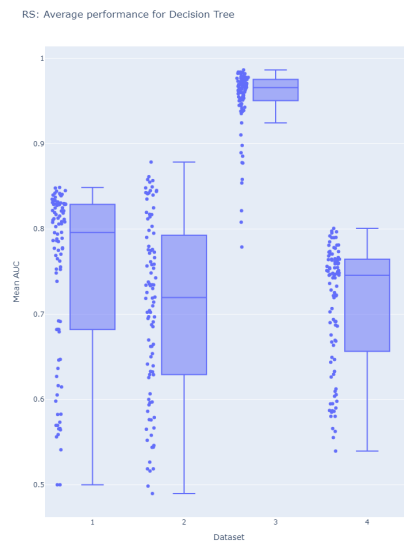
Poniższe boxploty przedstawiają, jak rozkładały się wartości AUC w zależności od zbioru danych, dla każdego z trzech algorytmów uczenia maszynowego i dwóch metod samplingu.



Rysunek 1: Średnie AUC dla Random Forest - Random Search



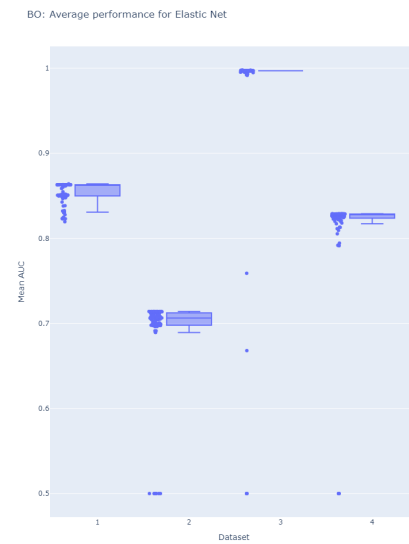
Rysunek 2: Średnie AUC dla Elastic Net - Random Search



Rysunek 3: Średnie AUC dla Decision Tree - Random Search

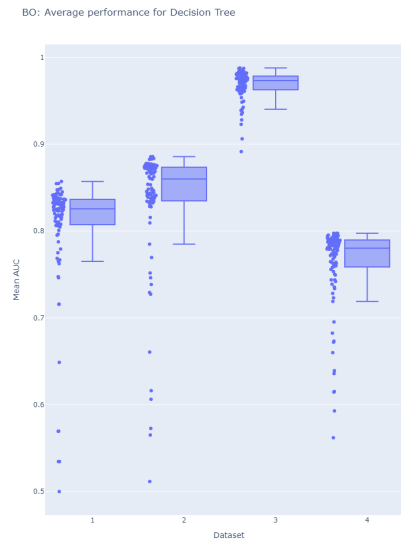


Rysunek 4: Średnie AUC dla Random Forest - Bayesian Optimization

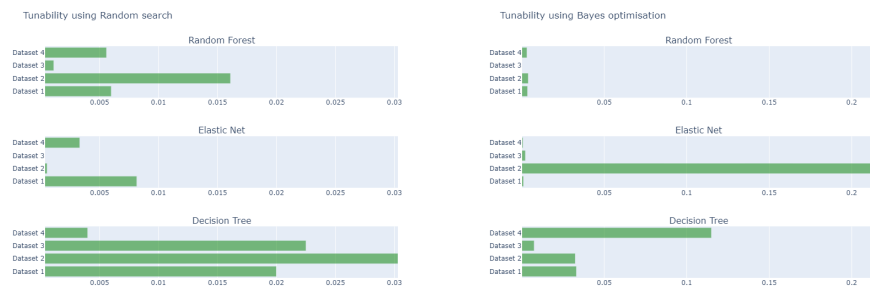


Rysunek 5: Średnie AUC dla Elastic Net - Bayesian Optimization



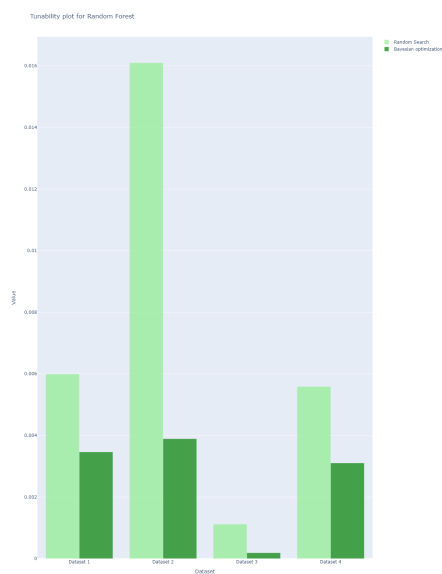


Rysunek 6: Średnie AUC dla Decision Tree - Bayesian Optimization

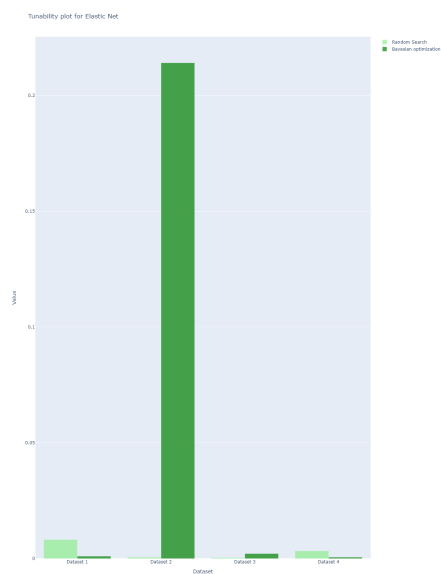


(a) Tunowalność dla Random Search      (b) Tunowalność dla Bayes Optimization

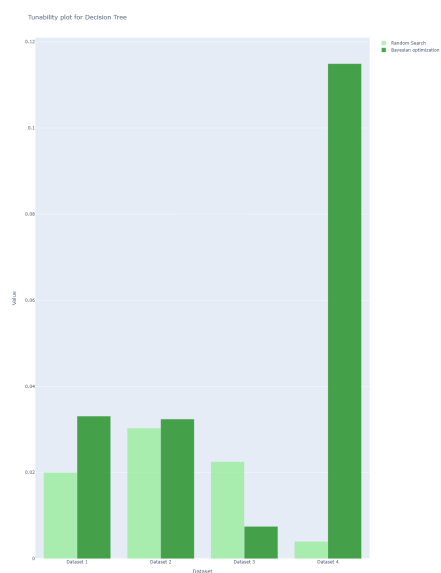
Rysunek 7: Tunowalność algorytmu w zależności od zbioru danych



(a) Tunowalność dla Random Forest



(b) Tunowalność dla Elastic Net



(c) Tunowalność dla Decision Tree

Rysunek 8: Tunowalność algorytmu w zależności od zbioru danych i metody

## 3 Podsumowanie

### 3.1 Random Search

W tej sekcji omówimy wyniki otrzymane przy użyciu metody Random Search. Patrząc na Rysunek 1, stwierdzamy, że wyniki dla każdego ze zbiorów są wysokie. Wybór hiperparametrów, w przypadku Random Forest ma największy wpływ na zbiór 2, wciąż znaczący dla zbiorów 1. i 4. Natomiast wyniki na zbiorze 3. są bardzo bliskie 1 i o małym odchyleniu standardowym, zatem poprawa względem defaultowych parametrów będzie niewielka. Dla Random Forest największa różnica na danym zbiorze to ok.0.05, a najmniejsza to ok. 0.001.

Dla algorytmu Elastic Net największą poprawę można uzyskać dla zbioru 1. i 2., których mediany wyników różnią się od maksymalnych o ok. 0.02 (Rysunek 2). Zbiory 3. oraz 4. mają bardzo bliskie wyniki mediany i czwartego kwartyla. Co ciekawe dla 2.,3. i 4. zbioru wyniku mogą diametralnie się pogorszyć dla pewnej grupy zestawów hiperparametrów, chociaż niewielkiej.

Algorytm Decision Tree ma najbardziej zróżnicowane wyniki. W porównaniu z poprzednimi algorytmami, różnice między medianą a 4. kwartylem są największe (w przypadku zbioru 2. to aż ok. 0.16), chociaż wciąż wartość mediany jest bliższa czwartemu kwartyłowi niż pierwszemu. Zdecydowanie dobór hiper parametrów dla zbioru 1. i 2. ma znaczący wpływ na wyniki.

Patrząc na rysunek 7a, możemy skonfrontować hipotezy dotyczące tunowalności. Dla dobranego defaultowego zestawu parametrów najwyższą tunowalność osiągnął Decision Tree, z wykorzystaniem najlepszych hiperparametrów dla danego zbioru danych, można uzyskać poprawę o ok. 0.02-0.03, chociaż dla 4. zbioru poprawa jest mniejsza niż 0.005. Najmniej poprawiły się rezultaty Elastic Net, gdzie w najlepszym przypadku poprawa jest mniejsza niż jedna setna, a dla dwóch zbiorów jest ona rzędu 0.0001. W przypadku Random Forest wyniki poprawiły się o w dwóch przypadkach o ok. 0.005, w najlepszym przypadku o ok. 0.015, a w najgorszym o ok. 0.001. Przypuszczenia co do stopnia wielkości poprawy wyników dla danego algorytmu potwierdziły się. Co ciekawe, pomimo dużych odchyłeń wyników dla Decision Tree ( największa różnica między medianą a najwyższym wynikiem wynosiła ok.0.16), najwyższa tunowalność wyniosła zaledwie 0.01. Jako, że tunowalność dla zbiorów jest relatywnie niewielka, korzystanie z ustalonego defaulta dla algorytmów może być sensowna, jeśli uzyskanie jak najwyższej skuteczności nie jest kluczowa.

### 3.2 Optymalizacja bayesowska

W tej sekcji omówimy wyniki otrzymane przy użyciu optymalizacji bayesowskiej.

Dla optymalizacji bayesowskiej, dla klasyfikatora Random Forest, rozkład wyników jest podobny jak w przypadku Random Search (Rysunek 4). Dla każdego ze zbiorów, większość wyników znajduje się w bliżej górnej granicy. Największa różnica między medianą, a najlepszym wynikiem to ok. 0.01 dla zbioru 2.. Prawdopodobnie tunowalność będzie niewielka.

W przypadku Elastic Net wyniki mają podobny rozkład co Random Forest, chociaż częściej występują anomalie wśród wyników (Rysunek 5). Dla każdego ze zbiorów, większość wyników znajduje się w bliżej górnej granicy. Różnice między największą i najmniejszą wartością (poza outlierami) są niewielkie, co najwyżej ok 0.05 (zbiór 1.). Jednak największa różnica między medianą a największą wartością to ok. 0.01 dla zbioru 2.. Dla pozostałych zbiorów ta różnica jest rzędu 0.001. Rozkład wyników jest podobny jak dla Random Search. Prawdopodobnie tunowalność będzie niewielka.

Dla klasyfikatora Decision Tree wyniki z użyciem metody bayesowskiej są różne od wyników z użyciem metody Random Search (Rysunek 6). Rozkłady wyników są bardziej skupione. Prawdopodobnie ma to związek z tym, że w przypadku optymalizacji bayesowskiej wybór parametrów jest zależny od zbioru. Nie mniej jednak największa różnica między wynikiem dla najlepszych parametrów, a medianą to ok.0.03 dla zbioru 1. i ok 0.025 dla zbioru 2., a najmniejsza różnica to ok.0.015 dla zbioru 3.. Zatem dobór parametrów ma największe znaczenie dla algorytmu Decision Tree.

Patrząc na Rysunek 7b, konfrontujemy hipotezy dotyczące tunowalności algorytmów. Dla dobranego defaultowego zestawu parametrów najwyższą tunowalność osiągnął Decision Tree, z wykorzystaniem najlepszych hiperparametrów dla zbioru 4., można uzyskać poprawę o ok. 0.1, chociaż dla 3. zbioru poprawa jest mniejsza niż 0.001. Najmniej poprawiły się rezultaty Random Forest, gdzie dla 3 zbiorów tunowalność wyniosła ok. 0.003. , a dla zbioru 3. ok.0.0001. Tunowalność Elastic Net, wyniosła 0.0006 dla zbioru 4., 0.001 dla zbioru 1., natomiast dla zbioru 3. to aż ok. 0.21. Zgodnie z przewidywaniami, Decision Tree okazał się najbardziej tunowalnym algorytmem. Random Forest i Elastic Net korzystanie z defaultu wydają się być satysfakcjonujące. Jednak wśród wyników Elastic Net występują optymalizacja bayesowska dla zbioru 3. uzyskała znacznie lepszy zestaw parametrów niż default uzyskany przez Random Search.

### 3.3 Porównanie metod strojenia i wnioski

Porównując użycie metod strojenia parametrów na Rysunku 8, w przypadku Random Forest użycie metody Random Search skutkuje wyższą tunowalnością algorytmu. Poza przypadkiem dla Random Search dla zbioru 2., tunowalność Random Forest była niewielka - mniejsza niż 0.006. Dla klasyfikatora Elastic Net, metoda dająca wyższą tunowalność zależała od zbioru, chociaż w przypadku zbioru 2. optymalizacja bayesowska ma znacząco wyższą tunowalność. Oprócz tego jednego przypadku, tunowalność Elastic Net była bardzo niewielka. Dla Decision Tree tunowalność jest najwyższa porównując z innymi algorytmami, a w większości przypadków optymalizacja bayesowska otrzymała najlepsze rezultaty, ale nie można uznać tego za ogólną zasadę. Poza przypadkiem dla zbioru 4., gdzie optymalizacja bayesowska otrzymała znacząco wyższą tunowalność, tunowalność Decision Tree oscylowała wokół wartości 0.02.

Niezależnie od metody strojenia parametrów, algorytm Decision Tree okazał się najbardziej tunowalnym. Tunowalność Random Forest i Elastic Net jest niewielka, nie mniej jednak korzystając z defaultu dla Elastic Net trzeba brać

pod uwagę możliwość wystąpienia szczególnych przypadków, w których strojenie parametrów do danego zbioru ma duże znaczenie. Korzystanie z defaultu dla Random Forest, może być sensowne, bo nawet dla szczególnego przypadku tunowalność wyniosła ok. 0.016. Nie można jednoznacznie stwierdzić wpływu metody strojenia parametrów na tunowalność algorytmu.