

Raport
Automatyczne uczenie maszynowe – Projekt 1

Krzysztof Adamczyk
Katarzyna Bownik
Aliaksei Shaukunou

18 listopada 2025

1 Wstęp

Celem projektu jest przeanalizowanie tunowalności hiperparametrów 3 algorytmów klasyfikacji - Decision Tree, Extra Trees i Gradient Boosting, wykorzystując 2 różne techniki losowania punktów - jedna oparta na wyborze punktów z rozkładu jednostajnego, a druga opierająca się na technice bayesowskiej. Historia tuningu zostanie wykorzystana do wyznaczenia dla każdego algorytmu nowej domyślnej konfiguracji hiperparametrów oraz do zbadania, czy metoda wyboru punktów ma wpływ na tunowalność. Również techniki losowania zostaną porównane pod względem prędkości osiągania stabilnych wyników.

2 Opis eksperymentu

Do badań zostały wybrane 4 zbiory danych z OpenML - adult (ID: 45068), blood-transfusion-centre (ID: 1464), credit (ID: 31) i diabetes (ID: 37). Dla każdego algorytmu i każdego zbioru danych wylosowano 50 punktów za pomocą funkcji RandomSearchCV i BayesSearchCV oraz wyznaczono miary AUC przy użyciu 3-krotnej stratyfikowanej walidacji krzyżowej. Zakres przeszukiwanych hiperparametrów jest przedstawiony w poniższej tabeli:

Algorytm	Hiperparametr	Typ	Zakres
Gradient Boosting	n_estimators	integer	50 - 200
	learning_rate	numeric	0.01 - 0.2
	max_depth	integer	1 - 15
	min_samples_split	integer	2 - 20
	min_samples_leaf	integer	1 - 10
	subsample	numeric	0.5 - 1.0
Extra Trees	n_estimator	integer	50 - 200
	criterion	discrete	gini, entropy
	max_depth	integer	1 - 30
	min_samples_split	integer	2 - 40
	ccp_alpha	numeric	0.0 - 0.01
Decision Tree	criterion	discrete	gini, entropy
	splitter	discrete	best, random
	max_depth	integer	1 - 30
	min_samples_split	integer	2 - 60
	min_samples_leaf	integer	1 - 60
	ccp_alpha	numeric	0.0 - 0.01

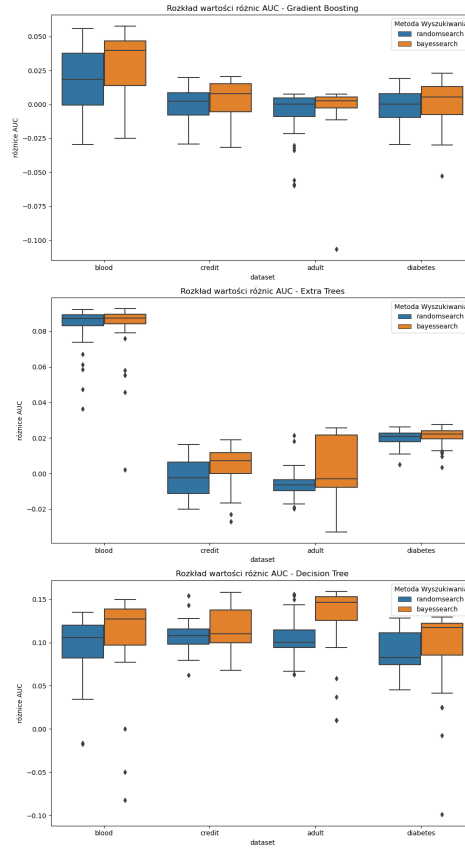
Tabela 1: Zakresy hiperparametrów

Poszczególne zakresy zostały wybrane na podstawie wartości podanych w pracy Tunability: Importance of Hyperparameters of Machine Learning Algorithms. W niektórych przypadkach te zakresy zostały zmniejszone, aby zwiększyć prawdopodobieństwo uzyskania dobrych wyników przy 50 losowaniach. Pozostałe zakresy wybrano na podstawie obserwacji, jakie wartości hiperparametrów występują przy dobrych wynikach AUC.

3 Wyniki

3.1 Tunowalność i rozkłady miar

Na poniższych wykresach przedstawione zostały rozkłady różnic pomiędzy uzyskanymi AUC a AUC dla domyślnej konfiguracji hiperparametrów. Jak możemy zauważyć generalnie dla każdego algorytmu i każdego zbioru danych mamy tendencję uzyskiwania wyższych wartości AUC przy użyciu bayesowskiej techniki losowania punktów.



Rysunek 1: Rozkłady różnic uzyskanych AUC

Tunowalności algorytmów dla obu technik losowania punktów są przedstawione w poniższej tabeli. Dla większej przejrzystości wyniki zostały zaokrąglone do szóstej liczby po przecinku.

Metoda samplingu	GradientBoosting	ExtraTree	DecisionTree
Random Search	0.025599	0.039077	0.143394
Bayes Search	0.027222	0.041258	0.149144

Tabela 2: Tunowalność algorytmów dla Random i Bayes Search

Gradient Boosting wykazuje bardzo niską tunowalność, co oznacza, że model działa blisko optimum już przy ustawieniach domyślnych, a jego przestrzeń hiperparametrów jest stosunkowo płaska i mało wrażliwa na tuning. Extra Trees osiąga ponad dwukrotnie wyższą tunowalność, ale warto zauważyć, że z rozkładów różnic możemy zauważyć, że dla zbioru danych blood-transfusion-centre uzyskujemy znacznie większe polepszenia niż dla pozostałych zbiorów danych. Średnia poprawa AUC bez tego zbioru wynosi około 0.021337 i 0.024117 dla RandomSearch i BayesSearch odpowiednio. Największą tunowalność uzyskano dla Decision Tree, co oznacza, że model jest silnie zależny od właściwej regularyzacji, a tuning znacząco ogranicza jego podatność na przeuczenie. W każdym przypadku, przy użyciu metody bayesowskiej, uzyskano wyższe wyniki.

Do zbadania wpływu techniki losowania punktów na uzyskane tunowalności użyto testu Wilcozona. Porównane zostały ze sobą najwyższe polepszenia miary AUC dla każdego algorytmu i każdego zbioru danych względem techniki losowania. Wartość statystyki Wilcozona dla naszych danych to 0, czyli w każdym przypadku uzyskaliśmy wyższe polepszenie przy użyciu techniki bayesowskiej. P-value wynosi 0.0005, czyli dla testu na poziomie istotności 0.05 mamy podstawy do odrzucenia hipotezy zerowej, czyli istnieją statystycznie istotne różnice pomiędzy tunowalnościami.

3.2 Nowe domyślne konfiguracje

Nowe domyślne konfiguracje zostały wybrane na podstawie wyników historii tunowalności przy użyciu techniki losowania punktów Random Search jako, że w tym przypadku przez ustawienie seedu dla każdego zbioru

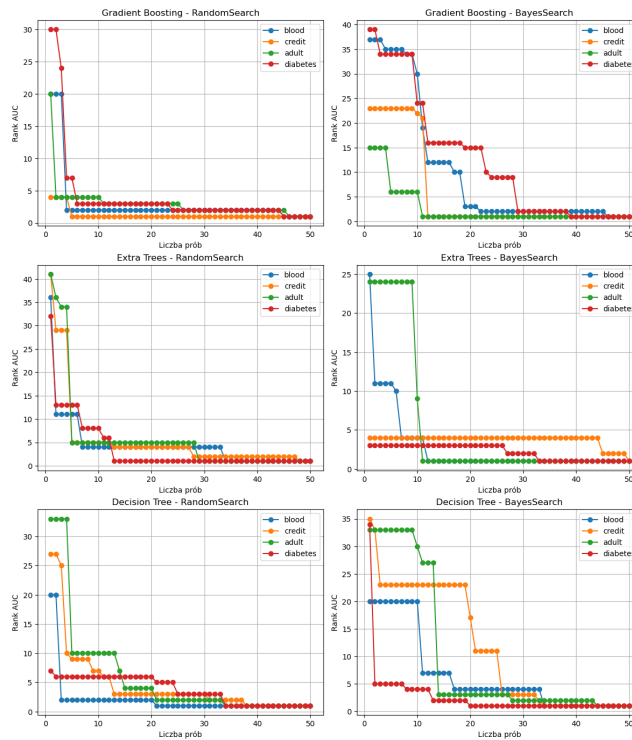
danych uzyskujemy tę samą siatkę hiperparametrów. Zostały wybrane te konfiguracje hiperparametrów, dla których AUC było średnio najwyższe na testowanych zbiorach danych.

Algorytm	Hiperparametr	Wartość domyślna	Nowa wartość domyślna
Gradient Boosting	n_estimators	100	59
	learning_rate	0.1	0.078934
	max_depth	3	2
	min_samples_split	2	6
	min_samples_leaf	1	1
	subsample	1.0	0.55
	mean AUC	0.800799	0.816085
Extra Trees	n_estimator	100	121
	criterion	gini	gini
	max_depth	None	18
	min_samples_split	2	23
	ccp_alpha	0.0	0.0
	mean AUC	0.777762	0.814422
Decision Tree	criterion	gini	gini
	splitter	best	best
	max_depth	None	10
	min_samples_split	2	33
	min_samples_leaf	1	38
	ccp_alpha	0.0	0.0
mean AUC		0.651015	0.783432

Tabela 3: Nowe wartości domyślne hiperparametrów

3.3 Prędkość zbieżności

Na poniższych wykresach przedstawiono, podczas której iteracji uzyskano najwyższy wynik AUC dla każdego z algorytmów przy danej technice losowania. Z wykresów wynika, że przy losowaniu przy użyciu Random-Search szybciej uzyskujemy stabilność wyników, ponieważ w większości przypadków już po 20 iteracjach nie otrzymujemy znacznych poprawień miary.



Rysunek 2: Zależność rang modeli od liczby iteracji

4 Podsumowanie

Z testowanych modeli najwyższą tunowalność uzyskaliśmy dla algorytmu Decision Tree, a najniższą dla algorytmu Gradient Boosting. Z wyników możemy wywnioskować, że używając bayesowskiej metody samplingu jesteśmy w stanie bardziej podwyższyć miarę AUC, ale wiąże się to z potrzebą przeprowadzenia większej liczby iteracji i większym kosztem obliczeniowym.