Tunowalnósc wybranych algorytmów uczenia maszynowego

Szymon Kiełtyka, Ryszard Czarnecki, Antoni Rakowski

18 listopada 2025

1 Algorytmy i zbiory danych

Wybrane algorytmy: DecisionTreeClassifier, KNeighborsClassifier, LogisticRegression z penalty='elasticnet' Wybrane zbiory danych: Credit, Blood, Jm1, Bank marketing, Diabetes, Heart, Pc4, Sonar, Schizo, Cmc, Eucalyptus

2 Wykorzystane metody

2.1 Zakresy hiperparametrów

Zakresy hiperparametrów zostały wybrane na podstawie dokumentacji oraz artykułu (Probs et al., 2019)

Algorytm	**Hiperparametr**	**Zakres**
DecisionTreeClassifier	max_depth	[1, 30]
	min_samples_split	[1, 60]
	min_samples_leaf	[1, 60]
KNeighborsClassifier	n_neighbors	[1, 30]
	weights	{'uniform', 'distance'}
LogisticRegression z penalty='elasticnet'	C	$[2^{-10}, 2^{10}]$
	l1_ratio	[0, 1]

Tabela 1: Zakresy hiperparametrów dla wybranych algorytmów.

2.2 Metody wyboru punktów

Do wyboru punktów zostały użyte dwie meteody:RandomizedSearchCV z pakietu scikit-learn oraz BayesSearchCV z pakietu scikit-optimize. Dla modeli DecisionTreeClassifier oraz LogisticRegression zrobiono 100 iteracji każdej z metod losowania punktów, a dla modelu KNeighborsClassifier 50 iteracji. W metodzie RandomizedSearchCV dla każdego zbioru danych przeszukiwana była taka sama siatka punktów dla danego modelu.

2.3 Definicje

Definicje optymalnej konfiguracji hiperparametrów dla danego zbioru, domyślnej konfiguracji hiperparametrów oraz tunowalności algorytmów zostały wyznaczona zgodnie z definicjami z artykułu (Probs et al., 2019). Jako funkcje straty R urzyto metryki AUC.

Definicja domyślnej konfiguracji hiperparametrów:

$$\theta^\star := \arg\min_{\theta\in\Theta} g\Big(R^{(1)}(\theta), \ldots, R^{(m)}(\theta)\Big),$$

gdzie g - funkcja agregująca (wykorzystana została średnia)

Definicja optymalnej konfiguracji hiperparametrów dla j-ego zbioru:

$$\theta^{(j)\star} := \arg\min_{\theta\in\Theta} R^{(j)}(\theta).$$

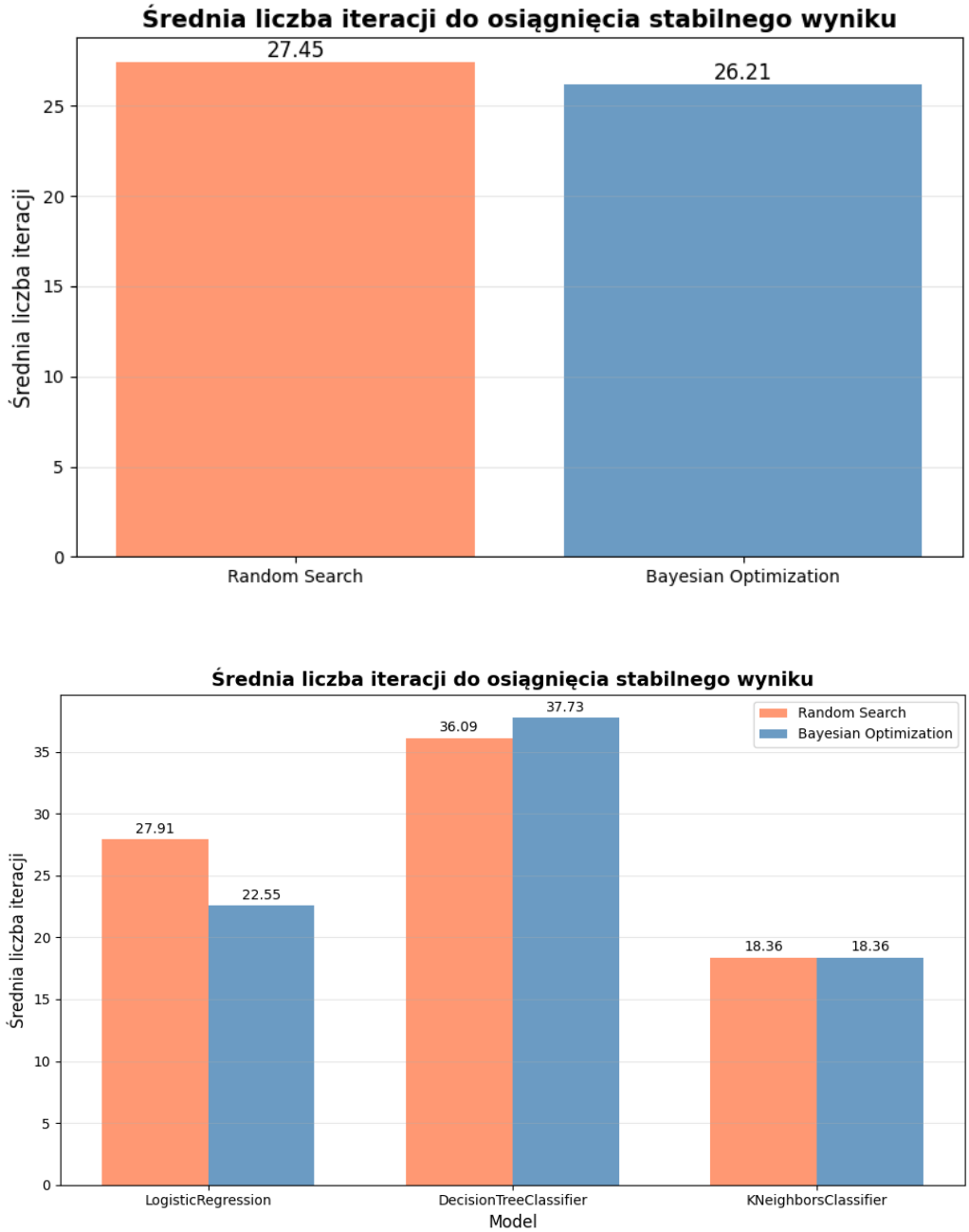
Definicja tunowalności algorytmu dla j-tego zbioru danych:

$$d^{(j)} := R^{(j)}(\theta^\star) - R^{(j)}(\theta^{(j)\star})$$

Ogólna tunowalność algorytmu została wyznaczona przez średnią tunowalność algorytmu dla wszystkich zbiorów danych.

3 Wyniki eksperymentów

3.1 Stabilne wyniki optymalizacji



Rysunek 1: Średnia liczba iteracji do osiągnięcia stabilnego wyniku (99,9% najlepszego wyniku)

Aby osiągnąć stabilne wyniki optymalizacji metoda BayesSearchCV potrzebuje średnio nieznacznie mniej kroków niż RandomizedSearchCV, aczkolwiek może się to różnić przy danym modelu

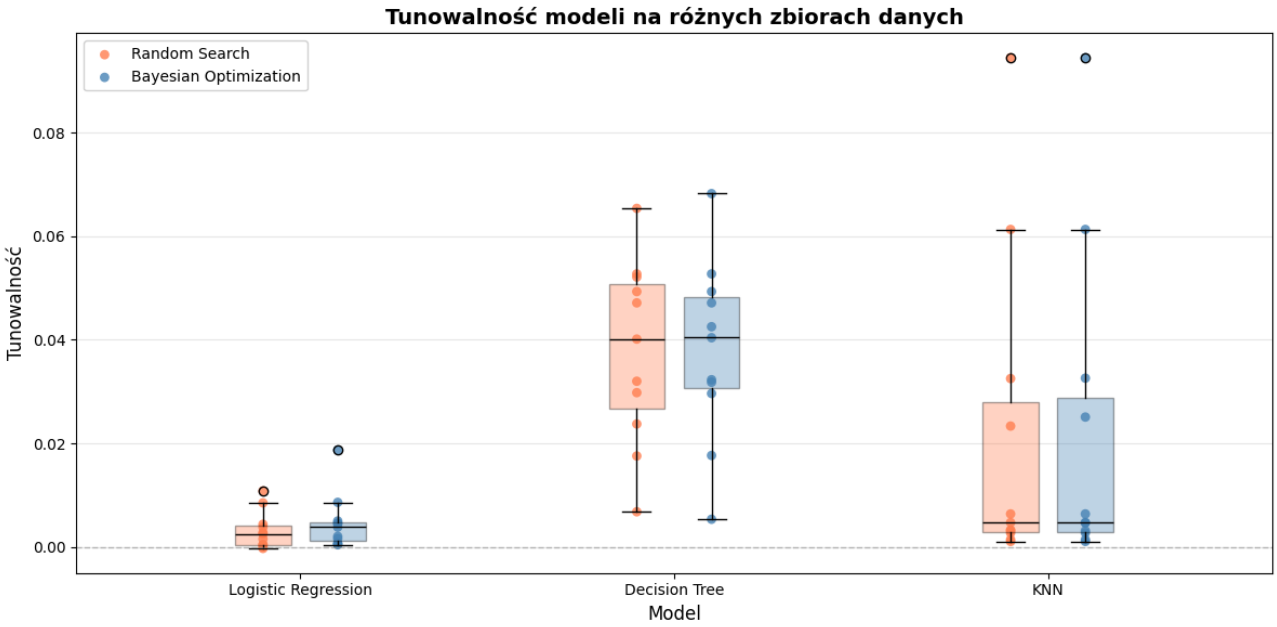
3.2 Domyślne konfiguracje

Algorytm	**Hiperparametr**	**Wartość**
DecisionTreeClassifier	max_depth	7
	min_samples_split	9
	min_samples_leaf	10
KNeighborsClassifier	n_neighbors	22
	weights	distance
LogisticRegression z penalty='elasticnet'	C	0.0802
	l1_ratio	0.1101

Tabela 2: Najlepsze znalezione wartości hiperparametrów dla każdego klasyfikatora

Domyślne konfiguracje dla danego modelu uzyskano przez wybór najlepszego średniego wyniku AUC dla wszystkich zbiorów danych

3.3 Tunowalność



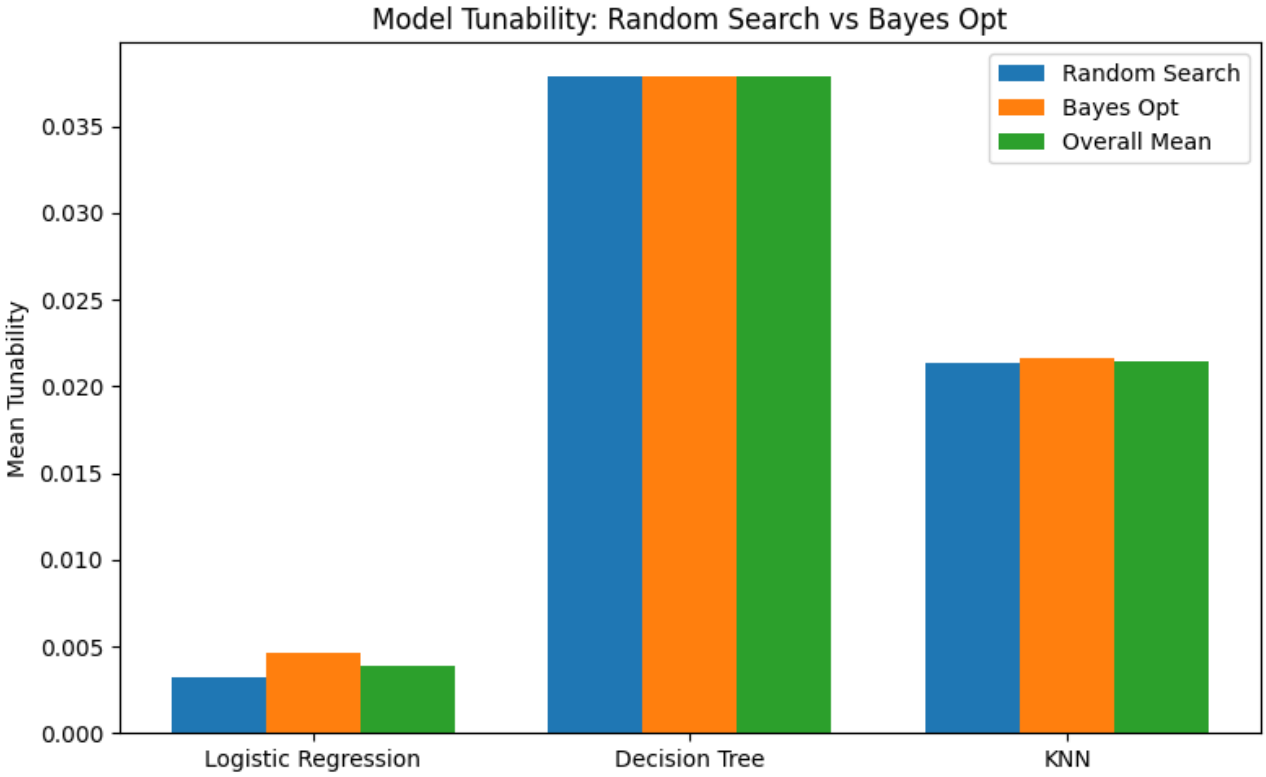
Rysunek 2: Tunowalnośc danych modeli

Obie techniki samplingu uzyskały lepsze wyniki niż domyślne wartości parametrów. Najbardziej tunowalnym algorytmem jest DecisionTreeClassifier (średnia tunowalność 0.03740), następnie KNeighborsClassifier (średnia tunowalność 0.021446), a najmniej tunowalneym algorytmem jest LogisticRegression z penalty='elasticnet' (średnia tunowalność 0.00388)

Zbiór danych	LogisticRegression		DecisionTreeClassifier		KNeighborsClassifier	
	Random Search	Bayes Opt	Random Search	Bayes Opt	Random Search	Bayes Opt
credit	0.0005	0.0007	0.0176	0.0177	0.0014	0.0014
jm1	0.0038	0.0038	0.0320	0.0323	0.0027	0.0027
blood	0.0000	0.0005	0.0471	0.0471	0.0325	0.0326
bank marketing	0.0004	0.0004	0.0401	0.0404	0.0064	0.0064
diabetes	0.0044	0.0046	0.0493	0.0493	0.0031	0.0031
heart	0.0028	0.0050	0.0654	0.0425	0.0033	0.0047
pc4	0.0085	0.0086	0.0527	0.0527	0.0047	0.0047
sonar	0.0003	0.0020	0.0521	0.0682	0.0613	0.0613
schizo	0.0108	0.0187	0.0237	0.0296	0.0945	0.0945
cmc	0.0015	0.0016	0.0298	0.0318	0.0233	0.0251
eucalyptus	0.0025	0.0045	0.0068	0.0053	0.0010	0.0010

Tabela 3: Tunowalność dla modelu i metody wyboru punktów

3.4 Bias sampling



Rysunek 3: Średnia tunowalność dla danej metody samplingu

Wyniki dla Random Search i Bayesian Optimization są bardzo zbliżone, więc nie można stwierdzić, że występuje bias sampling.

4 Literatura

[1] Probst, P., Boulesteix, A.-L., Bischl, B. (2019). Tunability: Importance of Hyperparameters of Machine Learning Algorithms. Journal of Machine Learning Research, 20, 1-32. Dostęp online: https://jmlr.org/papers/volume20/18-444/18-444.pdf