Wyznaczanie tunowalności hiperparametrów regresji logistycznej, lasu losowego i modelu XGBoost oraz porównanie metod optymalizacji hiperparametrów

Autorzy:

Hanna Szczerbińska

Helena Wałachowska

Paula Wołkowska

18 listopada 2025

Spis treści

1 Wstęp

Celem naszego projektu było odtworzenie eksperymentu opisanego w artykule "Tunability: Importance of Hyperparameters of Machine Learning Algorithms"[1], dotyczącego wyznaczania najbardziej optymalnego zestawu hiperparametrów dla wybranych algorytmów uczenia maszynowego oraz badania tunowalności tych algorytmów oraz ich poszczególnych hiperparametrów. Ponadto naszym zadniem było porównanie wyników uzyskiwanych za pomocą różnych metod optymalizacji hiperparametrów.

2 Dane

Tuning parametrów i uczenie algorytmów przeprowadziłyśmy na czterech możliwie różnorodnych zbiorach danych z repozytorium OpenML. Wybrałyśmy dane dotyczące problemu klasyfikacji binarnej. Były to "adult"o wymiarach [48842, 14] (cel: przewidzenie, czy dochód osoby przekracza 50 000 USD/rok), „bank-marketing" o wymiarach [45211, 17] (cel: przewidzenie, czy klient zdecyduje się na depozyt terminowy), „eeg-eye-state" o wymiarach [14980, 15] (cel: klasyfikacja stanu oka (zamknięte vs otwarte) na podstawie tych sygnałów EEG), „spambase" o wymiarach [4601, 58] (cel: klasyfikacja wiadomości e-mail jako spam / nie-spam).

Przed rozpoczęciem procesu doboru parametrów poddałyśmy nasze zbiory danych preprocessingowi, który obejmował uzupełnienie potencjalnych braków danych za pomocą mediany (dla cech numerycznych) i najczęściej występującej kategorii (dla cech kategorycznych), normalizację danych numerycznych poprzez dzielenie przez odchylenie standardowe i zamianę cech kategorycznych na zmienne binarne za pomocą metody One-Hot Encoding.

3 Metodologia

W naszej wersji eksperymentu postanowiłyśmy zająć się optymalizacją hiperparametrów 3 algorytmów używanych do klasyfikacji binarnej, różniących się poziomem skomplikowania modelu: regresji logistycznej, klasycznego lasu losowego i algorytmu XGBoost. Postanowiłyśmy porównać proces optymalizacji i jej rezultaty uzyskiwane za pomocą dwóch metod: Random Search i Bayesian Search. Każda z zastosowanych przez nas metod działała przez ustaloną liczbę iteracji (50 - 100) przy ustawionym ziarnie generatora, aby umożliwić replikację wyników. W każdej iteracji następuje trening i testowanie modelu za pomocą cross-walidacji - algorytm z wybranymi w danej iteracji hiperparametrami jest trenowany i testowany na kolejnych podzbiorach pełnego zbioru danych, a ostateczny wynik - jakość predykcji modelu w danej iteracji - to uśredniona miara AUC ze wszystkich foldów w cross-walidacji.

Następnie zgodnie z metodologią zaprezentowaną w artykule za pomocą metody Random Search wyznaczyłyśmy "globalnie optymalny"zestaw wartości hiperparametrów dla każdego algorytmu, a zatem wartości hiperparametrów, które osiągają średnio najlepsze wyniki niezależnie od zbioru danych - ze wszystkich przetestowanych przez Random Search kombinacji parametrów danego algormytmu wybrałyśmy tę, która dawała najwyższą średnią wartość miary AUC na wszystkich zbiorach danych.

W dalszych krokach eksperymentu wybrana kombinacja wartości hiperparametrów była traktowana jako defaultowa. Oznacza to, że podczas wyboru zestawu najbardziej optymalnych wartości parametrów danego algorytmu na konkretnym zbiorze danych tunowalność każdej kombinacji hiperparametrów była obliczna jako $AUC_{\text{Testowanej Kombinacji}} - AUC_{\text{Defaultowej Kombinacji}}$. Ostateczna wartość tunowalności danego algorytmu na wybranym zbiorze danych obliczana jest jako $max_{k \in K}(AUC_k - AUC_{default})$, gdzie K to zbiór wszystkich przetestowanych kombinacji hiperparametrów dla wybranego algorytmu. W analogiczny sposób wyznaczyliśmy tunowalność poszczególnych hiperparametrów - w tej sytuacji jedynie wartość optymalizowanego parametru była zmieniana przy pozostałych parametrach przyjmujących wartości defaultowe.

4 Skrócone wyniki na przykładzie algorytmu XGBoost

4.1 Wybór przestrzeni przeszukiwań

Zgodnie z tabelą zamieszczoną we wcześniej cytowanym artykule w poniższej tabeli zamieszczamy zakresy parametrów algorytmu XGBoost, dla których dokonałyśmy optymalizacji. W kolejnych kolumnach

[1] Probst, P., Boulesteix, A. L., Bernd Bischl, B. (2019). Tunability: Importance of Hyperparameters of Machine Learning Algorithms. Journal of Machine Learning Research, 20 1-32.

zamieszczamy również wyznaczone wartości parametrów defaultowych oraz wartości najbardziej optymalnych parametrów dla każdego zbioru z uwzględnieniem metody, za pomocą której zostały uzyskane (zestaw parametrów, który umożliwił uzyskanie najwyższego AUC na danym zbiorze).

Warto zwrócić uwagę, że parametry n_estimators i booster nie były optymalizowane - ich wartości były ustalone na cały czas trwania eksperymentu. Co ciekawe, dla zbiorów danych bank-marketing, eeg-eye-state i spambase ten sam zestaw wartości hiperparametrów umożliwił uzyskanie najwyższego AUC; zestaw tych wartości pochodzi z optymalizacji metodą Bayes Search.

Parametr	**Min**	**Max**	**Uwagi**	**default**	**adult (RS)**	**bank-marketing (BS)**	**eg-eye-state (BS)**	**spambase (BS)**
n_estimators	–	–	500	500	500	500	500	500
learning_rate	-10	0	2^x	0.0241	0.0205	0.0215	0.0215	0.0215
subsample	0.1	1	–	0.7184	0.4855	0.6506	0.6506	0.6506
booster	–	–	gbtree	gbtree	gbtree	gbtree	gbtree	gbtree
max_depth	1	15	–	13	7	14	14	14
min_child_weight	0	7	2^x	1.3532	3.3777	1.1678	1.1678	1.1678
colsample_bytree	0	1	–	0.5582	0.4896	0.7238	0.7238	0.7238
colsample_bylevel	0	1	–	0.7328	0.8566	0.4943	0.4943	0.4943
reg_lambda	-10	10	2^x	0.0062	0.003	0.002	0.002	0.002
reg_alpha	-10	10	2^x	0.0076	0.0126	0.0163	0.0163	0.0163

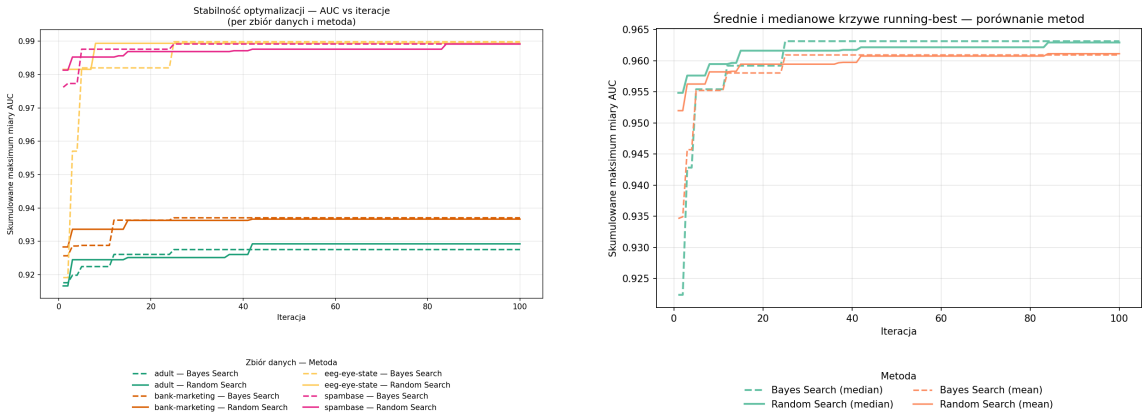
Tabela 1: Zakresy i najbardziej optymalne wartości parametrów modelu XGBoost.

Wyznaczone w analogiczny sposób optymalne zestawy hiperparametrów dla pozostałych omawianych algorytmów są umieszczone w ostatniej sekcji raportu, w tabelach 3 i 4.

4.2 Stabilizacja procesu optymalizacji

Poniższe wykresy przedstawiają najwyższą do danej iteracji wartość AUC uzyskaną na podstwie dotychczas przetestowanych zestawów parametrów (maksymalna skumulowana wartość AUC). Na pierwszym wykresie widzimy krzywe obrazujące proces optymalizacji wykonany obiema metodami na każdym zbiorze; drugi wykres przedstawia krzywe średnich i median wyników z pierwszego wykresu po wszystkich zbiorach dla obu metod optymalizacji.

Możemy stwierdzić, że po ok. 40 iteracjach wartość AUC dla wszystkich metod na każdym zbiorze nie ulega znaczącej poprawie.



Rysunek 1: Wykresy najlepszego dotychczas uzyskanego wyniku AUC w kolejnych iteracjach w podziale na poszczególne zbiory oraz zagregowane dla Random i Bayes Search, dla algorytmu XGBoost.

Analogiczne wykresy uzyskane dla pozostałych omawianych algorytmów są umieszczone w ostatniej sekcji raportu, na wykresach 4 i 5.

4.3 Tunowalność algorytmu wybranymi metodami

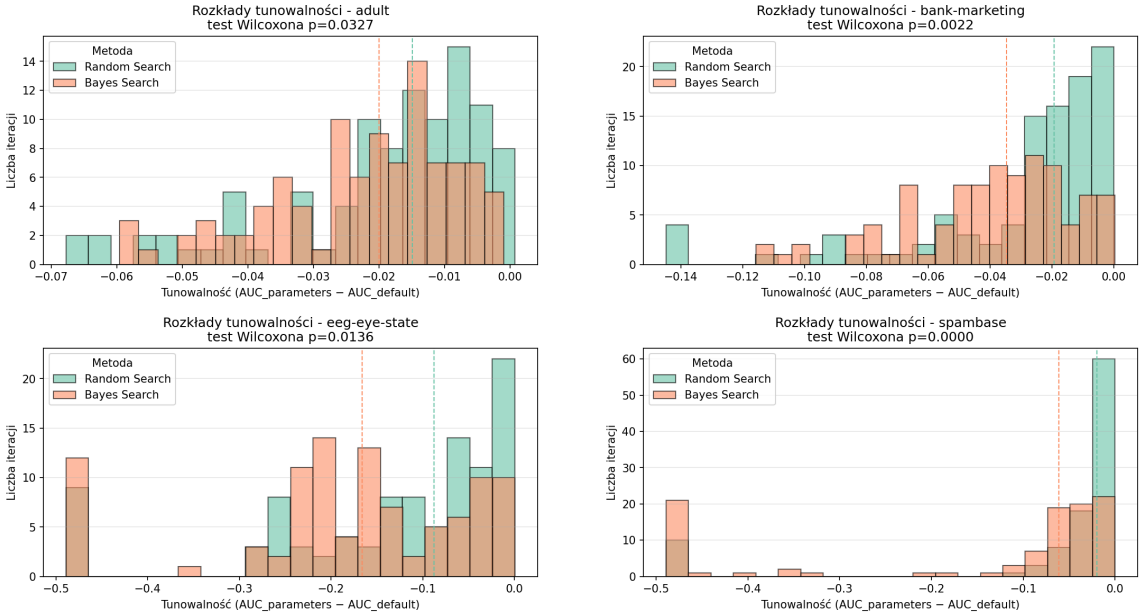
W poniższej tabeli przedstawiamy wartości AUC uzyskane dla każdego zbioru za pomocą parametrów defaultowych oraz tunowalność algorytmu XGBoost uzyskaną dla każdego zbioru danych za pomocą obu metod optymalizacji. Widzimy, że wartości AUC uzyskane przy ustawieniu defaultowych wartości hiperparametrów dla wybranych zbiorów danych są bardzo wysokie, co sprawia, że tunowalność algorytmu na tych zbiorach jest bardzo mała.

Zbiór danych	AUC default[2]	RS tunability	BS tunability
adult	0.9285	7.412e-04	-9.699e-04
bank-marketing	0.9366	4.864e-05	4.462e-04
eg-eye-state	0.9891	2.537e-04	7.314e-04
spambase	0.9892	0.0000	1.853e-05

Tabela 2: Wartość AUC dla parametrów defaultowych i tunowalność dla poszczególnych zbiorów danych uzyskane metodami Random i Bayes Search, dla algorytmu XGBoost.

Wyznaczone w analogiczny sposób wyniki dla pozostałych omawianych algorytmów są umieszczone w ostatniej sekcji raportu, w tabelach 5 i 6; na tej podstawie obliczono tunability tych modeli, wynoszące odpowiednio: dla regresji logistycznej i metody Random Search 3.0558e-3, dla regresji logistycznej i metody Bayes Search 3.1042e-3, dla lasu losowego i metody Random Search 2.2499e-2, dla lasu losowego i metody Bayes Search 3.3439e-2.

Na poniższych histogramach przestwiamy rozkłady tunowalności uzyskane dla za pomocą metod Random i Bayes Search dla każdego zbioru. W celu porównania, czy rozkłady tunowalności uzyskane za pomocą różnych metod dla danego zbioru różnią się istotnie wykonałyśmy nieparametryczny test Wilcoxona, który na poziomie istotności 0.05 wykazał, że dla każdego zbioru danych różnice między rozkładami tunowalności są istotne.



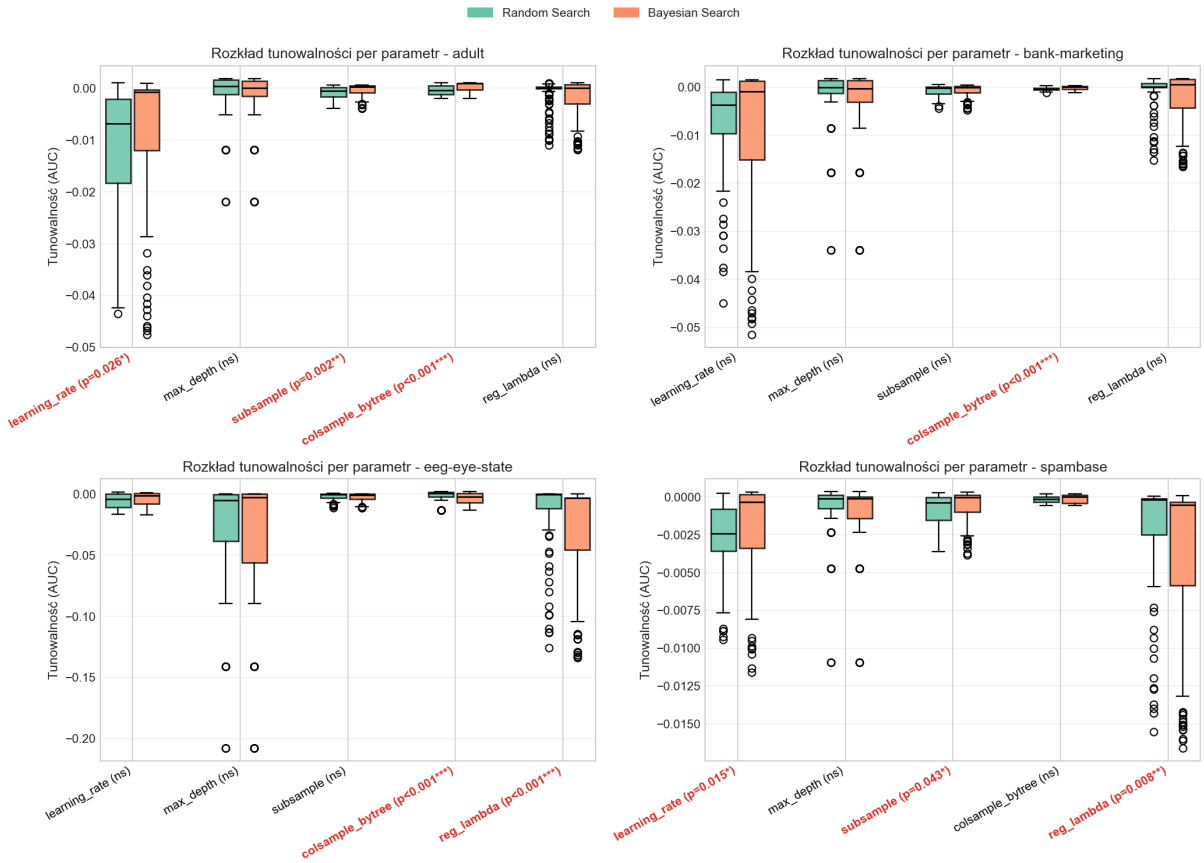
Rysunek 2: Rozkłady tunowalności uzyskane za pomocą metod Random i Bayes Search dla poszczególnych zbiorów, dla modelu XGBoost.

Analogiczne wykresy uzyskane dla pozostałych omawianych algorytmów są umieszczone w ostatniej sekcji raportu, na wykresach 6 i 7.

4.4 Tunowalność wybranych parametrów wybranymi metodami

Na poniższym wykresie przedstawiamy porównanie rozkładów tunowalności dla wybranych hiperparametrów (learning_rate, max_depth, subsample, colsample_bytree, reg_lambda) uzyskanych za pomocą dwóch metod optymalizacji na każdym zbiorze danych. Na czerwono zaznaczono nazwy hiperparametrów, których rozkłady tunowalności różniły się istotnie według testu Wilcoxona na poziomie istotności 0.05. Dla kilku zbiorów danych istotne różnice w rozkłądach tunowalności zostały wykryte dla prametrów colsample_bytree, learning_rate, reg_lambda.

[2]Wartości miar AUC są zbliżone na każdym zbiorze, dlatego nie normalizujemy ich podczas uśredniania w celu wyznaczenia defaultowej kombinacji hiperparametrów.



Rysunek 3: Rozkłady tunowalności uzyskane za pomocą metod Random i Bayes Search dla wybranych hiperparametrów na każdym zbiorze.

Wyznaczone w analogiczny sposób tunowalności poszczególnych hiperparametrów dla pozostałych omawianych algorytmów są umieszczone w ostatniej sekcji raportu, w tabelach 7 i 8.

5 Wnioski

Dla wszystkich przetestowanych algorytmów wyniki miary AUC podczas optymalizacji hiperparametrów stabilizują się po ok. 40 iteracjach. Pomimo różnych miar wartości AUC uzyskiwanych na poszczególnych zbiorach (0.55-0.98), tunowalność algorytmu wyznaczona na podstawie tych zbiorów danych jest pomijalnie mała. Różnice w tunowalności uzyskane za pomocą dwóch metod optymalizacji są przeważnie statystycznie istotne. Dla algorytmu XGBoost optymalizacja metodą Random Search pozwala uzyskiwać lepsze kombinacje hiperparametrów dla tego algorytmu niz Bayes Search; pozostałe algorytmy uzyskują lepsze wyniki przy zastosowaniu optymalizacji metodą Bayes Search.

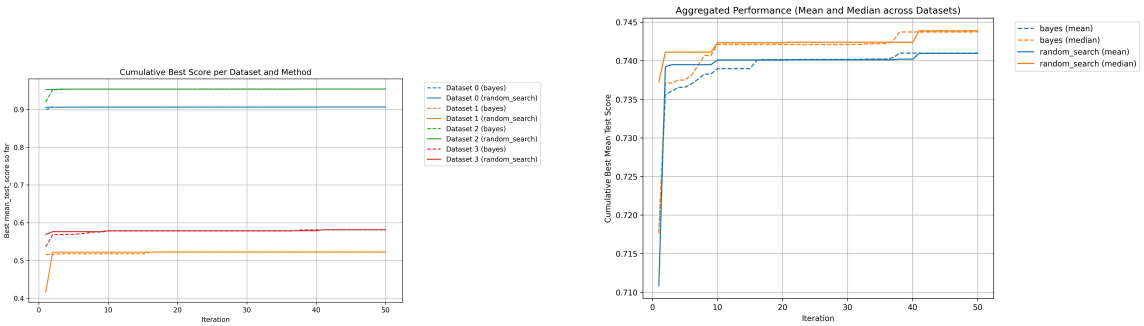
6 Dodatek: dokładne wyniki dla wszystkich testowanych algorytmów

Parametr	Min	Max	Uwagi	default	adult (RS)	bank-marketing (BS)	eg-eye-state (BS)	spambase (BS)
l1_ratio	0	1	–	0.5	1.00	0.25	0.50	1.00
C	−4	4	log	2.7826	0.3594	2.7826	2.1544	0.3594
penalty	–	–	{l2, elasticnet}	elasticnet	elasticnet	elasticnet	elasticnet	elasticnet
solver	–	–	saga	saga	saga	saga	saga	saga
max_iter	–	–	5000	5000	5000	5000	5000	5000

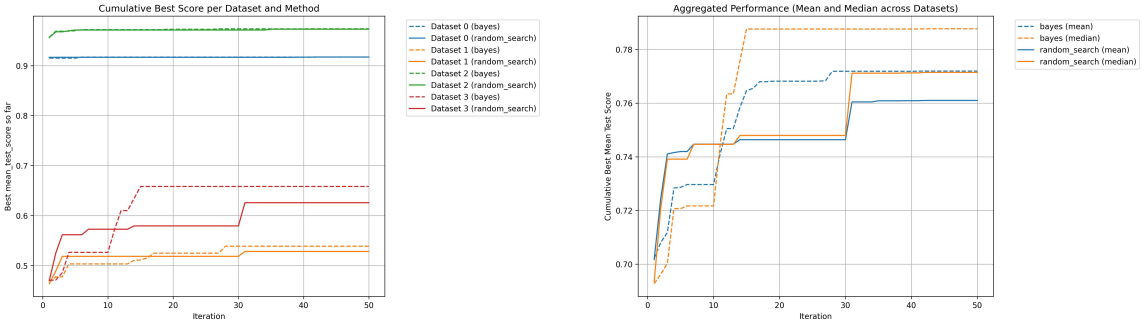
Tabela 3: Zakresy i najbardziej optymalne wartości parametrów w modelu regresji logistycznej.

Parametr	Min	Max	Uwagi	default	adult (RS)	bank-marketing (BS)	eg-eye-state (BS)	spambase (BS)
bootstrap	–	–	{True, False}	False	True	True	False	True
max_depth	3	20	–	3	12	4	17	4
max_features	–	–	{sqrt, log2, None}	log2	sqrt	log2	log2	log2
min_samples_leaf	1	20	–	4	6	7	2	7
n_estimators	10	250	–	77	171	50	60	50

Tabela 4: Zakresy i najbardziej optymalne wartości parametrów w modelu Random Forest.



Rysunek 4: Wykresy najlepszego dotychczas uzyskanego wyniku AUC w kolejnych iteracjach w podziale na poszczególne zbiory oraz zagregowane dla Random i Bayes Search, dla algorytmu regresji logistycznej.



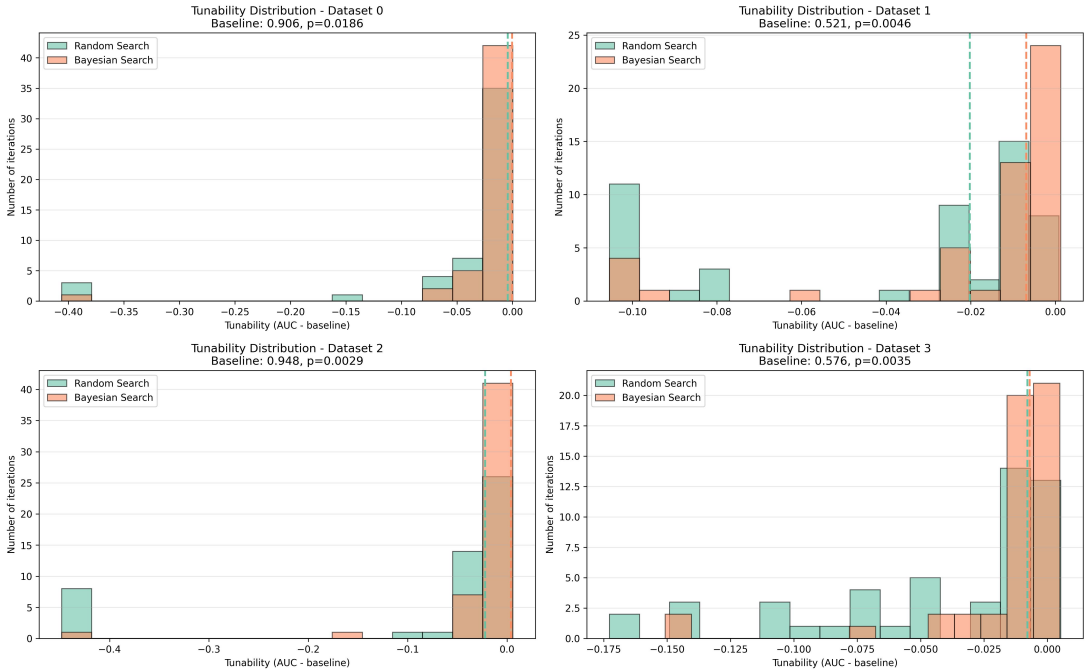
Rysunek 5: Wykresy najlepszego dotychczas uzyskanego wyniku AUC w kolejnych iteracjach w podziale na poszczególne zbiory oraz zagregowane dla Random i Bayes Search, dla algorytmu Random Forest.

Zbiór danych	default score	RS best score	BS best score
adult	0.906104	0.906407	0.906487
bank-marketing	0.521086	0.521964	0.522348
eg-eye-state	0.948269	0.954050	0.954125
spambase	0.576088	0.581350	0.581004

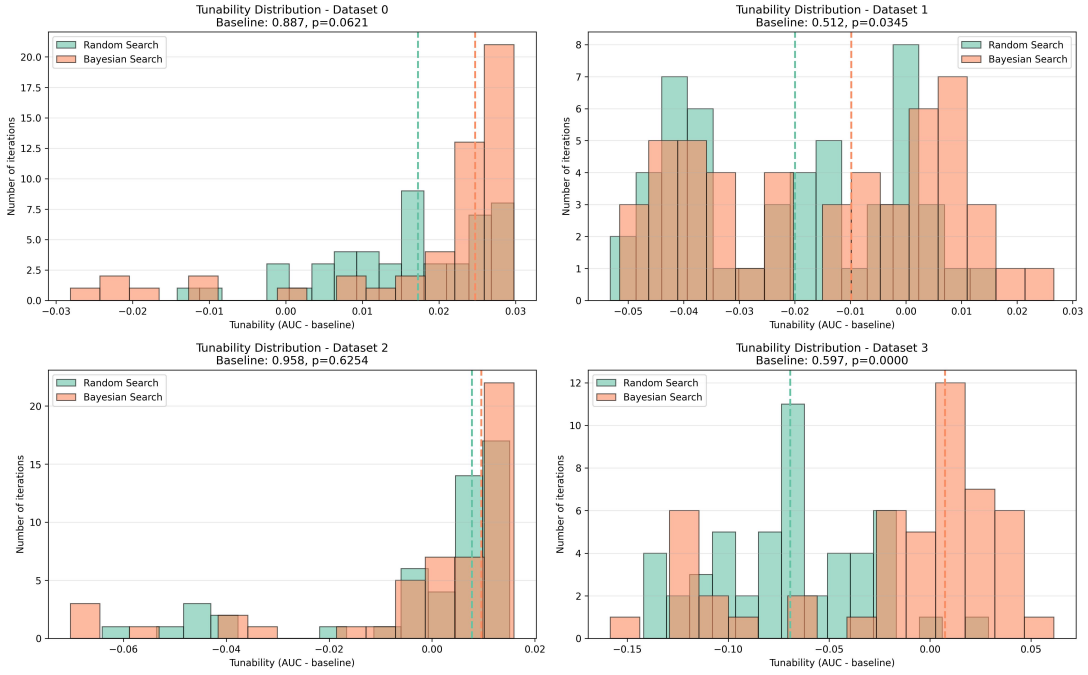
Tabela 5: Wartość AUC dla parametrów defaultowych i tunowalność dla poszczególnych zbiorów danych uzyskane metodami Random i Bayes Search, dla algorytmu regresji logistycznej.

Zbiór danych	default score	RS best score	BS best score
adult	0.887411	0.917187	0.917209
bank-marketing	0.512048	0.528225	0.538664
eg-eye-state	0.957753	0.972869	0.973686
spambase	0.596791	0.625719	0.658202

Tabela 6: Wartość AUC dla parametrów defaultowych i tunowalność dla poszczególnych zbiorów danych uzyskane metodami Random i Bayes Search, dla algorytmu Random Forest.



Rysunek 6: Rozkłady tunowalności uzyskane za pomocą metod Random i Bayes Search dla poszczególnych zbiorów, dla modelu regresji logistycznej.



Rysunek 7: Rozkłady tunowalności uzyskane za pomocą metod Random i Bayes Search dla poszczególnych zbiorów, dla modelu Random Forest.

nazwa hiperparametru	**tunability**	**std**	**min AUC diff**	**max AUC diff**
l1_ratio	1.754313e-03	2.589666e-03	1.114345e-04	5.564684e-03
C	1.441337e-03	2.892216e-03	-1.409477e-05	5.779649e-03
penalty	3.004212e-04	5.961278e-04	-2.969632e-06	1.194577e-03
solver	3.623742e-07	2.470015e-06	-1.782163e-06	3.893113e-06
max_iter	-1.098195e-07	6.015240e-07	-8.382956e-07	5.934155e-07

Tabela 7: Tunowalność poszczególnych hiperparametrów algorytmu regresji logistycznej.

nazwa hiperparametru	**tunability**	**std**	**min AUC diff**	**max AUC diff**
n_estimators	0.029113	0.041487	0.002979	0.090456
min_samples_leaf	0.020409	0.022584	0.003340	0.051935
max_depth	0.015912	0.009145	0.003816	0.025844
max_features	0.013881	0.017258	0.001101	0.038927
bootstrap	0.006789	0.008934	-0.001487	0.018709

Tabela 8: Tunowalność poszczególnych hiperparametrów algorytmu Random Forest.