Analiza tunowalności hiperparametrów

Martyna Sadowska, Zuzanna Zalewska, Aleksandra Zawadka

1 Wstęp

Celem projektu jest przeanalizowanie i porównanie różnych metod strojenia hiperparametrów oraz wyznaczenie nowej domyślnej konfiguracji dla każdego z rozważanych algorytmów. Wykorzystane zostały trzy modele uczenia maszynowego: Decision Tree, Random Forest oraz XGBoost. Każdy z nich został zoptymalizowany przy użyciu dwóch technik doboru punktów: losowego przeszukiwania (Random Search) oraz optymalizacji bayesowskiej (Bayesian Optimization), w celu zbadania możliwości dostrajania poszczególnych algorytmów. Do pomiaru jakości działania modeli wykorzystano metrykę ROC-AUC, umożliwiającą porównanie wydajności klasyfikatorów niezależnie od progu decyzyjnego. Dzięki temu zmiany w ROC-AUC są porównywalne między różnymi zbiorami danych i nie wymagają dodatkowego skalowania, co czyni ją odpowiednią miarą do oceny jakości klasyfikacji w tym projekcie. Podstawą naszych badań jest artykuł [1].

2 Zbiory danych

Do przeprowadzenia badań wykorzystano następującę zbiory danych klasyfikacyjnych:

1. Fitness Classification Dataset
2. Heart Failure Prediction Dataset
3. Placement Prediction Dataset
4. Travel Insurance Prediction Data
5. Customer Churn Dataset

Każdy ze zbiorów został przetworzony w jednolity sposób, czyli usunięto brakujące wartości, przeprowadzono kodowanie zmiennych kategorycznych oraz normalizację cech numerycznych.

3 Definicja i estymacja tunowalności hiperparametrów

W analizie tunowalności hiperparametrów rozważamy każdy algorytm uczenia maszynowego jako funkcję $\hat{f}(X, \theta)$, gdzie θ jest wektorem hiperparametrów należących do przestrzeni poszukiwań Θ. Aby ocenić jakość wybranego zestawu hiperparametrów, definiujemy funkcję straty $L(Y, \hat{f}(X, \theta))$ oraz odpowiadające jej oczekiwane ryzyko:

$$R(\theta) = E[L(Y, \hat{f}(X, \theta)) \mid P],$$

gdzie P jest nieznanym rozkładem danych.

Dla m różnych zbiorów danych lub rozkładów $P_1, \ldots, P_m$ otrzymujemy m funkcji ryzyka

$$R^{(j)}(\theta) = E[L(Y, \hat{f}(X, \theta)) \mid P_j], \quad j = 1, \ldots, m.$$

Najlepsza konfiguracja hiperparametrów dla zbioru danych j definiowana jest jako

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

Wartości domyślne, stosowane w oprogramowaniu, powinny dobrze sprawdzać się na różnych zbiorach danych. Możemy zdefiniować optymalne wartości domyślne na podstawie eksperymentów na wielu zbiorach danych jako

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

gdzie g jest funkcją agregującą (w tym przypadku średnią).

Tunowalność algorytmu na zbiorze danych j definiujemy jako różnicę między ryzykiem przy wartości domyślnej a ryzykiem przy najlepszej możliwej konfiguracji:

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

4 Siatka hiperparametrów

Dla każdego algorytmu wybrano zestaw kluczowych hiperparametrów, które mają istotny wpływ na wydajność modelu. Wartości hiperparametrów były losowane z określonych przedziałów (zostało to przedstawione w tabeli 1), co umożliwiało badanie tunowalności modeli. Zakresy hiperparametrów używane do strojenia powinny obejmować wartości bliskie optymalnym konfiguracjom z wysokim prawdopodobieństwem.

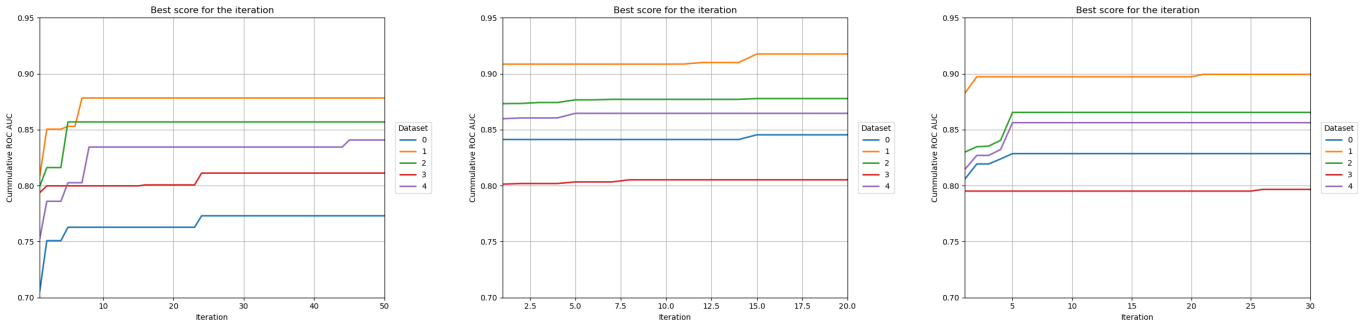
Algorytm	Hiperparametr	Typ	Dolna granica	Górna granica
Decision Tree	*max_depth*	całkowity	1	30
	min_samples_split	całkowity	2	20
	min_samples_leaf	całkowity	1	10
	max_features	liczbowy	0.1	0.9
Random Forest	*n_estimators*	całkowity	100	2000
	max_depth	całkowity	5	30
	min_samples_leaf	całkowity	1	10
	max_features	liczbowy	0.1	0.9
XGBoost	*n_estimators*	całkowity	100	5000
	learning_rate	liczbowy	0.01	0.3
	max_depth	całkowity	1	15
	subsample	liczbowy	0.1	0.9

Tabela 1: Hiperparametry algorytmów.

5 Random Search

Pierwszą metodą losowania punktów, która została użyta był Random Search. Dla każdego modelu zastosowałyśmy pięciokrotną walidację krzyżową i w zależności od wydajności modelu od 20 do 50 iteracji.

Na wykresie 1 widać, że dla Decision Tree istotne było wykonać przynajmniej 10 iteracji, natomiast później poprawa wyniku jest już nieznaczna. Dla Random Forest nowe iteracje nie wnosiły dużej poprawy wartości ROC-AUC w żadnej z kolejnych iteracji. Dla XGBoost pierwsze 5 iteracji znacząco poprawiło wynik, natomiast później nie ma już widocznej zmiany.



Decision Tree | Random Forest | XGBoost

Rysunek 1: Najlepsze wartości ROC-AUC otrzymane do danej iteracji dla trzech modeli po wyszukiwaniu losowym.

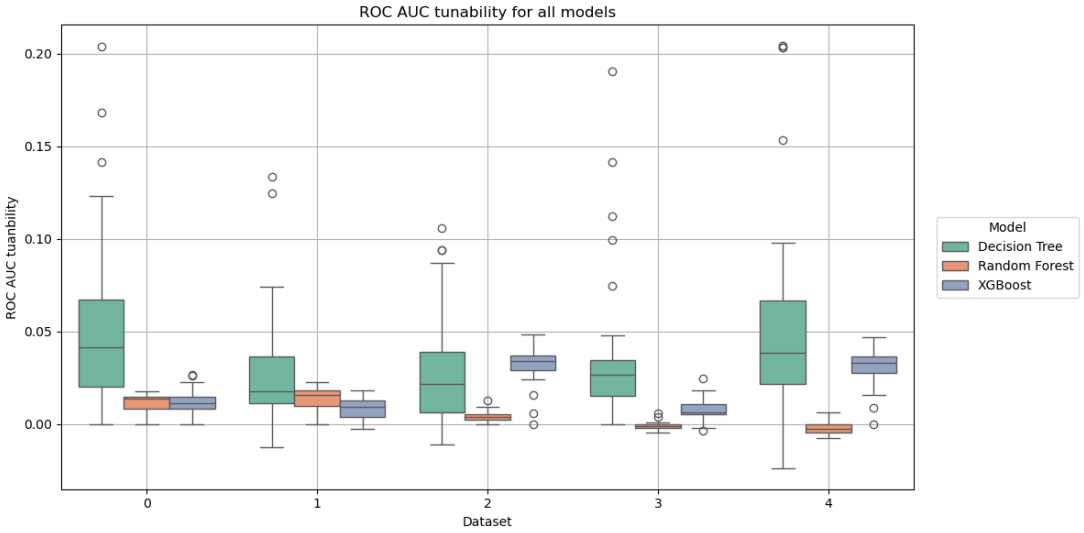
Po wykonaniu wyszukiwania losowego dla każdego z modeli uzyskałyśmy hiperparametry, które dawały średnio najlepsze wyniki na wszystkich zbiorach danych. Modele, które zostały naszymi nowymi defaultami są umieszczone w tabeli 2.

Decision Tree		**Random Forest**		**XGBoost**	
max_depth:	9.0	*max_depth:*	12.0	*learning_rate:*	0.1421
max_features:	0.5565	*max_features:*	0.1980	*max_depth:*	5.0
min_samples_leaf:	9.0	*min_samples_leaf:*	6.0	*n_estimators:*	139.0
min_samples_split:	15.0	*n_estimators:*	795.0	*subsample:*	0.7411
Mean Test Score:	0.8226	**Mean Test Score:**	0.8598	**Mean Test Score:**	0.8481

Rysunek 2: Najlepsze modele otrzymane po wyszukiwaniu losowym

Tutaj tunowalność jest liczona jako różnica między wynikiem optymalnym a uzyskanym w danej iteracji. Przedstawiłyśmy ją na wykresie 3 dla każdego zbioru danych i dla każdego modelu. Widać na nim, że wyniki dla Decision Tree

znacząco różniły się na przestrzeni iteracji i wybrany przez nas model ma lepsze wyniki niż modele o innych wylosowanych hiperparametrach. Tunowalność Random Foresta jest najmniejsza z naszych modeli. Wyniki w poszczególnych iteracjach tylko nieznacznie się od siebie róznią. XGBoost także nie daje dużych wyników tunowalności, ale jednak wyniki najlepszego modelu są trochę lepsze od reszty.

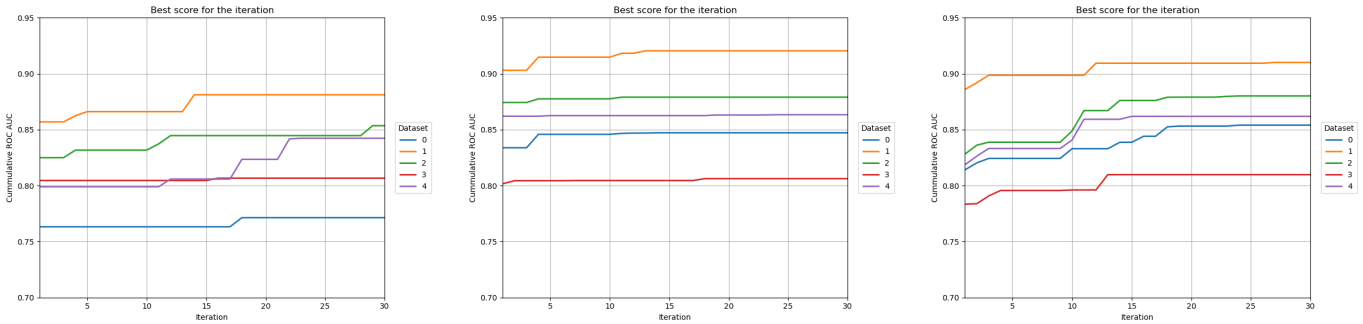


Rysunek 3: Tunowalność wszystkich testowanych modeli.

6 Optymalizacja Bayesowska

Druga metodą losowania punktów opierała się na technice bayesowskiej. Dla każdego modelu zastosowałyśmy pięciokrotną walidację krzyżową i 30 iteracji.

Na wykresie 4 widać, że dla Decision Tree najwięcej wnosiły iteracje 10-25, ale później też były iteracje poprawiające wynik. Dla Random Forest pierwsze kilka iteracji podnosiły wartość AUC-ROC, ale kolejne już nic nie zmieniały. XGBoost potrzebował ok. 20 iteracji, aby osiągnąć optymalny wynik.



Decision Tree Random Forest XGBoost

Rysunek 4: Najlepsze wartości ROC-AUC otrzymane do danej iteracji dla trzech modeli po optymalizacji bayesowskiej.

Po wykonaniu optymalizacji bayesowskiej dla każdego z modeli i dla każdego zbioru danych uzyskałyśmy hiperparametry, które dawały najlepsze wyniki. Modele, które zostały naszymi nowymi domyślnymi konfiguracjami są umieszczone w tabeli 6 w załączniku.

Analizując tunowalność modeli po optymalizacji bayesowskiej porównałyśmy wyniki ROC-AUC dla najlepszych znalezionych przez nas modeli z wynikami otrzymanymi w domyślnych modelach systemowych. W tabeli 2 umieszczone są otrzymane przez nas wyniki. Widać, że dla wszystkich zbiorów danych wyniki Decision Tree zostały poprawione. Dla XGBoosta niektóre zbiory uzyskały lepsze wyniki niż modele domyślne, ale niektóre poprawiły się nieznacznie. Namniejszą poprawę widać w modelu Random Forest. Poprawa wyniku dla każdego zbioru jest niewielka.

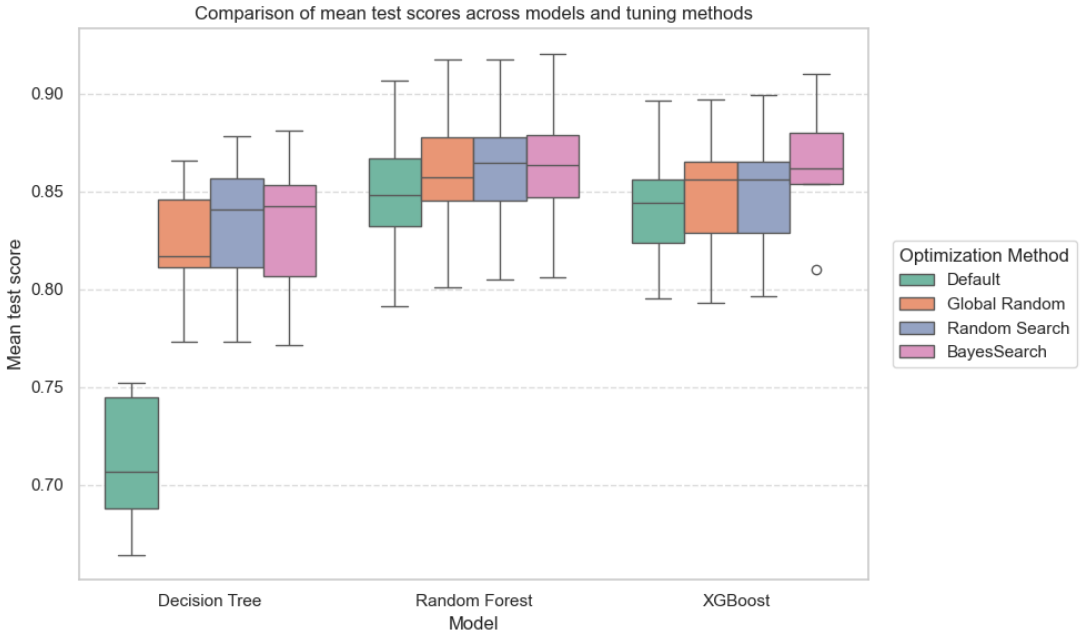
7 Porównanie metod samplingu

Na wykresie 5 widać, że największa poprawa została osiągnięta dla Decision Tree. BayesSearch okazał się trochę skuteczniejszy od RandomSearch dla modeli Random Forest i XGBoost. W ogólności oba sposoby samplingu popra-

Zbiór danych	Model	Wynik ROC-AUC po optymalizacji bayesowskiej	Poprawa wyniku domyślnego
0	Decision Tree	0.7714	0.1073
1	Decision Tree	0.8812	0.1290
2	Decision Tree	0.8536	0.1468
3	Decision Tree	0.8067	0.0618
4	Decision Tree	0.8423	0.1543
0	Random Forest	0.8472	0.0152
1	Random Forest	0.9204	0.0140
2	Random Forest	0.8791	0.0121
3	Random Forest	0.8063	0.0148
4	Random Forest	0.8634	0.0150
0	XGBoost	0.8540	0.0304
1	XGBoost	0.9101	0.0137
2	XGBoost	0.8801	0.0243
3	XGBoost	0.8098	0.0148
4	XGBoost	0.8619	0.0180

Tabela 2: Porównanie wyników modeli po optymalizacji bayesowskiej i poprawy względem wartości domyślnych.

wiają wyniki w porównaniu do Default. Dokładne wyniki różnic w tych metodach zostały przedstawione w tabeli 3 w załączniku.



Rysunek 5: Porównanie metod samplingu: Default - model z domyślnymi hiperparametrami; Global Random - najlepszy model globalnie według RandomSearch; Random Search - najlepszy model dla danego zbioru; BayesSearch - najlepszy model dla danego zbioru danych.

8 Podsumowanie

Celem projektu było zbadanie tunowalności poszczególnych algorytmów. Porównano skuteczność dwóch metod strojenia hiperparametrów - Random Search i optymalizacji bayesowskiej - dla modeli Decision Tree, Random Forest oraz XGBoost. Obie techniki poprawiły wyniki względem konfiguracji domyślnych, przy czym największą tunowalność zaobserwowano dla Decision Tree. Random Forest i XGBoost okazały się mniej wrażliwe na zmianę hiperparametrów, co potwierdza dobre dopasowanie ich ustawień domyślnych. Optymalizacja bayesowska osiągała nieco lepsze rezultaty niż wyszukiwanie losowe, zwłaszcza dla bardziej złożonych modeli.

Literatura

[1] Philipp Probst, Anne-Laure Boulesteix & Bernd Bischl. *Tunability: Importance of Hyperparameters of Machine Learning Algorithms. Journal of Machine Learning Research*, 20 (2019), 1–32.

Załącznik

Decision Tree		Random Forest		XGBoost	
Dataset 0:	0.7714	**Dataset 0:**	0.8472	**Dataset 0:**	0.8540
max_ depth	4.0	*max_ depth*	14.0	*learning_ rate*	0.01
max_ features	0.8463	*max_ features*	0.1045	*max_ depth*	1.0
min_ samples_ leaf	10.0	*min_ samples_ leaf*	9.0	*n_ estimators*	4142.0
min_ samples_ split	19.0	*n_ estimators*	1987.0	*subsample*	0.4153
Dataset 1:	0.8812	**Dataset 1:**	0.9204	**Dataset 1:**	0.9101
max_ depth	5.0	*max_ depth*	6.0	*learning_ rate*	0.01
max_ features	0.1258	*max_ features*	0.1162	*max_ depth*	15.0
min_ samples_ leaf	4.0	*min_ samples_ leaf*	1.0	*n_ estimators*	564.0
min_ samples_ split	9.0	*n_ estimators*	1228.0	*subsample*	0.1000
Dataset 2:	0.8536	**Dataset 2:**	0.8791	**Dataset 2:**	0.8801
max_ depth	8.0	*max_ depth*	8.0	*learning_ rate*	0.01
max_ features	0.2860	*max_ features*	0.1000	*max_ depth*	1.0
min_ samples_ leaf	7.0	*min_ samples_ leaf*	10.0	*n_ estimators*	5000.0
min_ samples_ split	5.0	*n_ estimators*	2000.0	*subsample*	0.1000
Dataset 3:	0.8067	**Dataset 3:**	0.8063	**Dataset 3:**	0.8098
max_ depth	12.0	*max_ depth*	30.0	*learning_ rate*	0.01
max_ features	0.9000	*max_ features*	0.8982	*max_ depth*	10.0
min_ samples_ leaf	9.0	*min_ samples_ leaf*	5.0	*n_ estimators*	100.0
min_ samples_ split	20.0	*n_ estimators*	143.0	*subsample*	0.8337
Dataset 4:	0.8423	**Dataset 4:**	0.8634	**Dataset 4:**	0.8619
max_ depth	7.0	*max_ depth*	21.0	*learning_ rate*	0.01
max_ features	0.6300	*max_ features*	0.4371	*max_ depth*	15.0
min_ samples_ leaf	10.0	*min_ samples_ leaf*	8.0	*n_ estimators*	100.0
min_ samples_ split	2.0	*n_ estimators*	2000.0	*subsample*	0.2992

Rysunek 6: Najlepsze modele otrzymane po optymalizacji bayesowskiej dla poszczególnych zbiorów danych.

Dataset	Mean Test Score Default	Mean Test Score Global Random	Mean Test Score Random	Mean Test Score Bayes	Diff Default vs Global	Diff Default vs Random	Diff Default vs Bayes	Model
0	0.6641	0.7731	0.7731	0.7714	0.1090	0.1090	0.1073	Decision Tree
1	0.7522	0.8659	0.8783	0.8812	0.1137	0.1261	0.1290	Decision Tree
2	0.7067	0.8459	0.8569	0.8536	0.1391	0.1502	0.1468	Decision Tree
3	0.7448	0.8112	0.8112	0.8067	0.0664	0.0664	0.0618	Decision Tree
4	0.6880	0.8171	0.8408	0.8423	0.1291	0.1527	0.1543	Decision Tree
0	0.8320	0.8454	0.8454	0.8472	0.0134	0.0134	0.0152	Random Forest
1	0.9065	0.9176	0.9176	0.9204	0.0111	0.0111	0.0140	Random Forest
2	0.8670	0.8778	0.8778	0.8791	0.0109	0.0109	0.0121	Random Forest
3	0.7915	0.8008	0.8052	0.8063	0.0094	0.0137	0.0148	Random Forest
4	0.8483	0.8573	0.8646	0.8634	0.0090	0.0163	0.0150	Random Forest
0	0.8236	0.8286	0.8286	0.8540	0.0050	0.0050	0.0304	XGBoost
1	0.8964	0.8968	0.8994	0.9101	0.0004	0.0031	0.0137	XGBoost
2	0.8559	0.8654	0.8654	0.8801	0.0096	0.0096	0.0243	XGBoost
3	0.7950	0.7932	0.7966	0.8098	-0.0018	0.0016	0.0148	XGBoost
4	0.8439	0.8563	0.8563	0.8619	0.0123	0.0123	0.0180	XGBoost

Tabela 3: Porównanie wyników modeli w zależności od metody optymalizacji hiperparametrów. Default - model z domyślnymi hiperparametrami; Global Random - najlepszy model globalnie według RandomSearch; Random Search - najlepszy model dla danego zbioru; BayesSearch - najlepszy model dla danego zbioru danych według BayesSearch