Raport - Projekt 1

Bartosz Szymański, Aleksandra Uznańska, Igor Lechoszest

18 listopada 2025

1 Wstęp

Wydajność modeli uczenia maszynowego jest silnie uzależniona od odpowiedniego doboru ich hiperparametrów. Głównym celem projektu była optymalizacja hiperparametrów, zbadanie, jak dużą poprawę jakości modelu można uzyskać dla każdego z algorytmów oraz analiza tunowalności dla trzech modeli klasyfikacyjnych: regresji logistycznej z regularyzacją Elastic Net, Extra Trees oraz K-nearest Neighbors.

2 Analizowane Przestrzenie Hiperparametrów

Kluczowym elementem eksperymentu był dobór szerokich, ale obliczeniowo zasadnych przestrzeni hiperparametrów dla każdego z badanych algorytmów. W Tabeli 1 przedstawiliśmy szczegółowy zakres analizowanych przez nas wartości.

3 Selekcja Hiperparametrów

Jednym z zadań projektu było znalezienie dla każdego z trzech badanych algorytmów (ElasticNet, Extra Trees Classifier, K-Nearest Neighbors) nowej, uogólnionej domyślnej kombinacji hiperparametrów. Taka konfiguracja powinna osiągać średnio najlepsze wyniki na wszystkich czterech analizowanych przez nas zbiorach danych. W tym celu wykorzystaliśmy algorytm RandomizedSearchCV, który dzięki parametrom $random\_state = 123$, $cv = 5$ oraz $n\_iter = 1000$ zapewnił, że dla każdego algorytmu testowaliśmy tę samą, 1000-punktową siatkę hiperparametrów na każdym zbiorze danych, a każda z tych kombinacji była oceniana przy użyciu 5-krotnej walidacji krzyżowej z metryką roc_auc. Proces ten dostarczył nam informację o tym, która kombinacja hiperparametrów jest lokalnie najlepsza, a wykorzystując pełne wyniki kroswalidacji oraz uśredniając pozycję w rankingu każdej kombinacji mogliśmy wybrać średnio najlepsze hiperparametry dla wszystkich czterech zbiorów danych.

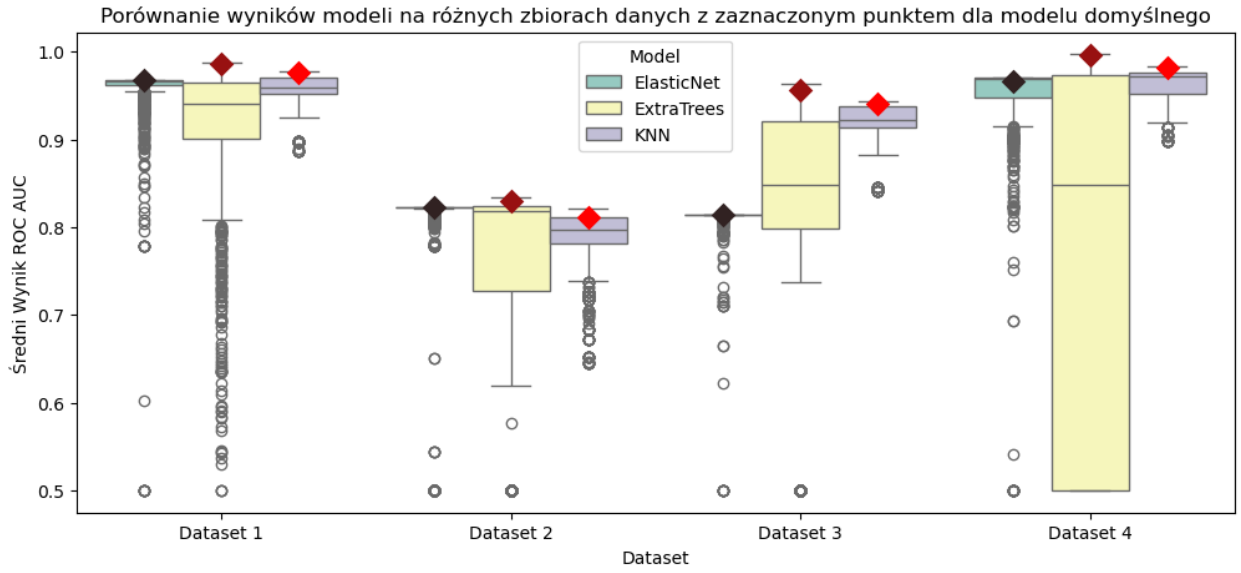
4 Tunowalność algorytmów

Tunowalność jest to potencjalny zysk wydajności (mierzony metryką roc_auc), jaki można osiągnąć poprzez modyfikacje kombinacji hiperparametrów w odniesieniu do ustalonej konfiguracji domyślnej. Niska tunowalność świadczy o tym, że algorytm jest stabilny oraz zysk z optymalizacji hiperparametrów w porównaniu do konfiguracji bazowej jest niewielki. Dla każdego algorytmu oraz datasetu tunowalność, zmierzyliśmy jako

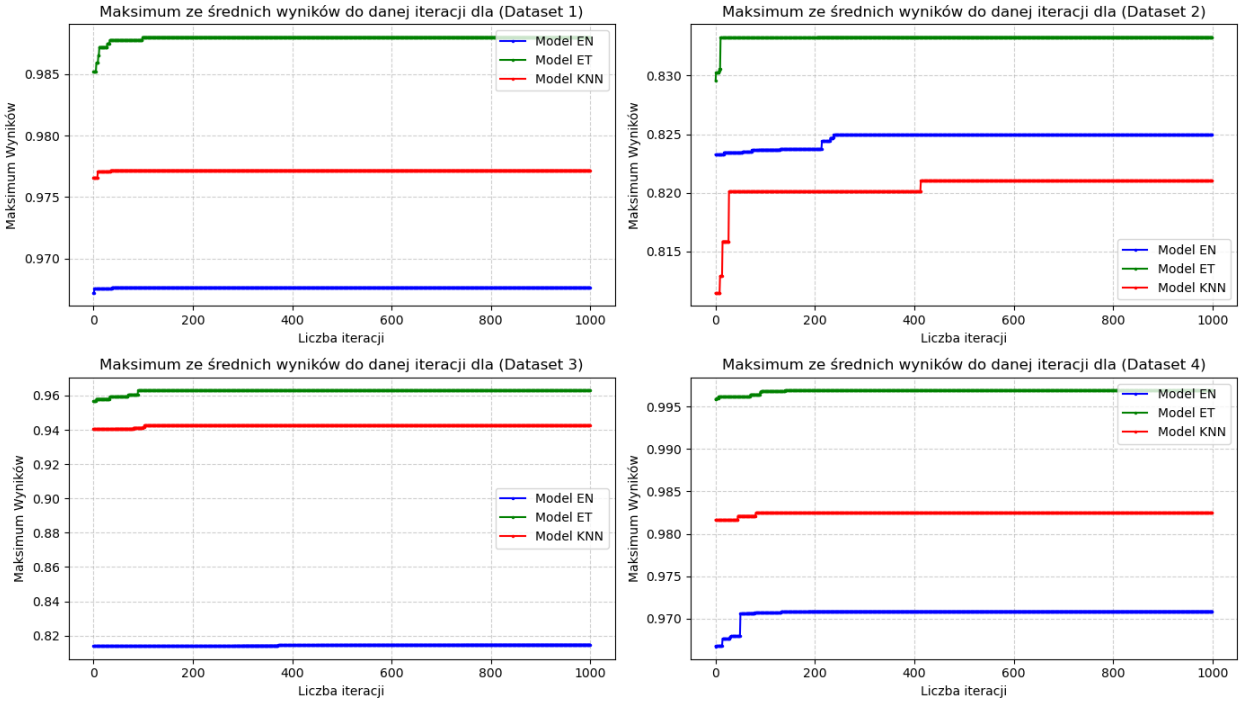
$$\textbf{tunability} = \textbf{best score} - \textbf{default score}.$$

Algorytm	Hiperparametr	Badany zakres	Wartość optymalna
Regresja Logistyczna	C	$[2^{-10}, 2^{10}]$ (300 wart. w skali log.)	≈ 0.00596
	l1_ratio	$[0, 1]$ (1000 wart. roz. liniowo)	≈ 0.001
Extra Trees Classifier	n_estimators	{100, 200, 500, 1000}	100
	criterion	{'gini', 'entropy'}	'entropy'
	max_depth	{None, 10, 20, 30, 100}	100
	min_samples_split	{2, 3, 5, 7, 11, 13}	11
	min_samples_leaf	{1, 2, 5}	1
	max_features	{'sqrt', 'log2', None, 1}	'log2'
	min_impurity_decrease	{0.0, 0.01, 0.1}	0.0
K-Nearest Neighbors	n_neighbors	$\{1, 2, 3, \ldots, 30\}$	1
	weights	{'uniform', 'distance'}	'distance'
	metric	{'minkowski', 'manhattan', 'chebyshev', 'euclidean'}	'chebyshev'
	algorithm	{'auto', 'ball_tree', 'kd_tree', 'brute'}	'kd_tree'
	leaf_size	$\{10, 100\}$	100
	p	{1,2}	1

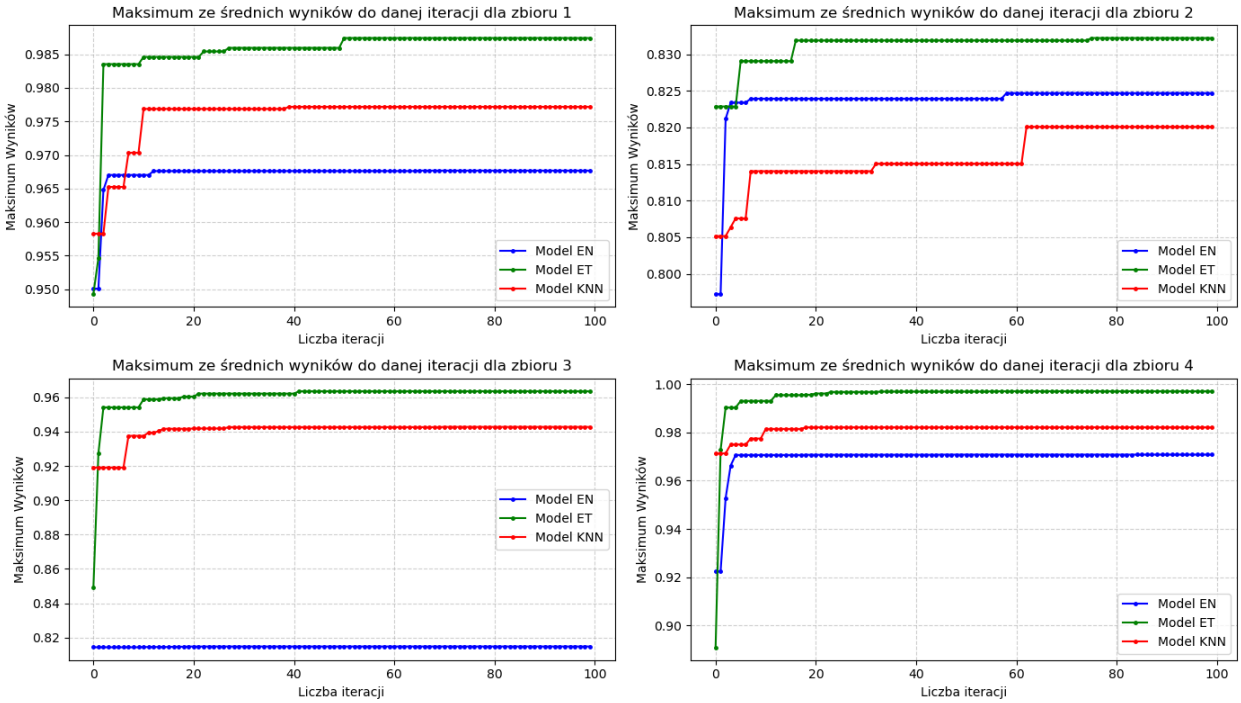
Tabela 1: Opis użytych hiperparametrów i znalezionych wartości optymalnych



Rysunek 1: Wykres porównujący średnie wyniki uzyskane dla danych kombinacji hiperparametrów badanych metodą RandomSearchCV



Rysunek 2: Wykres porównujący średnie wyniki uzyskane do danej iteracji metody RandomSearchCV



Rysunek 3: Wykres porównujący średnie wyniki uzyskane do danej iteracji metody bayesowskiej

5 Optymalizacja Bayesowska

Podobny eksperyment wykonany został przy użyciu bayesowskiej optymalizacji hiperparametrów (funkcja BayesSearchCV z pakietu scikit-learn). Wykorzystana została ta sama co wcześniej szeroka siatka hiperparametrów, $cv = 5$ i $n\_iter = 100$. Jako, że w optymalizacji bayesowskiej ciężko kontrolować siatkę hiperparametrów używaną przez modele, to nie da się znaleźć średnio najlepszej kombinacji dla 4 zbiorów danych. Ciekawy jest jednak rozkład wyników w zależności od liczby iteracji, zobrazowany na Rysunku 3.

6 Wnioski

Dzięki naszemu eksperymentowi znalezione zostały nowe optymalne siatki hiperparametrów, ich wartości można znaleźć w Tabeli 1. Wiemy również, że od pewnego momentu wyniki stabilizują się. Moment ten jest zależny od zbioru danych, ale na ogół dla RandomSearchCV było to około setnej iteracji, a dla metody bayesowskiej około sześćdziesiątej piątej iteracji. Tunowalność poszczególnych algorytmów przedstawia się następująco:

	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Elastic Net	0.0005	0.0017	0.0010	0.0041
Extra Trees	0.0028	0.0037	0.0063	0.0010
KNN	0.0006	0.0096	0.0021	0.0008

Tabela 2: Tabela różnic między wynikiem modelu znalezionego metodą RandomSearch a wynikiem domyślnej siatki hiperparametrów

	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Elastic Net	0.0005	0.0014	0.0011	0.0041
Extra Trees	0.0022	0.0027	0.0068	0.0012
KNN	0.0006	0.0086	0.0021	0.0005

Tabela 3: Tabela różnic między wynikiem modelu znalezionego metodą bayesowską a wynikiem domyślnej siatki hiperparametrów

Z Tabel 2 i 3 wynika, że tunowalność modeli jest niska i nie ma wielkich różnic między metodami optymalizacji wyboru hiperparametrów, czyli nie występuje bias sampling.

Z Rysunku 1 wynika, że Elastic Net wyróżnia się stabilnymi wynikami niezależnie od dobranej siatki hiperparametrów a Extra Trees ma duży rozrzut średnich wyników kroswalidacji.