

---

# Raport z Realizacji Projektu Badawczego Automatyczne Uczenie Maszynowe

## Optymalizacja i Porównanie Metod Strojenia Hiperparametrów dla Wybranych Algorytmów z Wykorzystaniem Techniek Samplingu Rok akademicki 2024/2025

Wersja 1.0  
Andrii Voznesenskyi

Opiekun naukowy:  
mgr Anna Kozak

14 listopada 2024

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>4</b>
<b>2</b>	<b>Zbiory danych</b>	<b>4</b>
2.1	Przetwarzanie danych . . . . .	4
<b>3</b>	<b>Metodyka</b>	<b>4</b>
3.1	Wybrane Algorytmy . . . . .	4
3.2	Techniki Tunowania . . . . .	5
<b>4</b>	<b>Wyniki</b>	<b>5</b>
4.1	Tunowalność Algorytmów . . . . .	5
4.2	Porównanie Metod Tunowania . . . . .	6
4.3	Konwergencja Wyników . . . . .	6
4.4	Diagram Różnic Krytycznych . . . . .	7
<b>5</b>	<b>Wnioski</b>	<b>7</b>

## **Streszczenie**

Niniejszy raport prezentuje wyniki analizy tunowalności hiperparametrów dla czterech algorytmów uczenia maszynowego: XGBoost, Random Forest, ElasticNet i Gradient Boosting. Wykorzystano trzy techniki tunowania: Grid Search, Random Search oraz Optymalizację Bayesowską. Przeanalizowano 10 zbiorów danych pod kątem efektywności, stabilności oraz tunowalności algorytmów. Wyniki wskazują, że Optymalizacja Bayesowska była najbardziej efektywna, osiągając stabilne rezultaty przy mniejszej liczbie iteracji. XGBoost i Gradient Boosting wykazały największą tunowalność, co sugeruje ich podatność na poprawę wydajności dzięki odpowiedniej konfiguracji hiperparametrów.

# 1 Wstęp

Raport dotyczy analizy tunowalności hiperparametrów algorytmów: XGBoost, Random Forest, ElasticNet oraz Gradient Boosting. Celem było porównanie trzech metod optymalizacji hiperparametrów (*Grid Search*, *Random Search*, *Optymalizacja Bayesowska*) pod względem efektywności, stabilności i tunowalności modeli.

Tunowalność odnosi się do wrażliwości algorytmu na zmiany konfiguracji hiperparametrów. W badaniu uwzględniono dziesięć zbiorów danych o zróżnicowanych charakterystykach, w tym klasyfikacyjnych (*binary*, *multiclass*) oraz regresyjnych, takich jak *breast cancer*, *iris* czy *california housing*.

Efektywność modeli oceniano na podstawie wartości kolumny **score**, która zawiera wyniki odpowiednie dla zadania (np. *F1-score dla klasyfikacji lub inna metryka dostosowana do zadania regresji*). Brak jednoznacznego podziału metryk w wynikach CSV nie pozwala na rozróżnienie, czy kolumna **score** reprezentuje np. **MSE** czy **F1-score** w analizie wizualnej. Jednakże, wyniki te odzwierciedlają jakość modelu w ramach zastosowanych metod tunowania.

## 2 Zbiory danych

Do analizy wykorzystano 10 zróżnicowanych zbiorów danych, obejmujących zadania klasyfikacyjne i regresyjne:

**Breast Cancer** (*scikit-learn*) to dane klasyfikacyjne dotyczące diagnostyki raka piersi, zawierające 30 cech histopatologicznych. **Iris** (*scikit-learn*) to klasyczny zbiór klasyfikacyjny opisujący cechy trzech gatunków irysów. **California Housing** (*scikit-learn*) obejmuje dane regresyjne dotyczące cen domów w Kalifornii z uwzględnieniem cech demograficznych. **Wine** (*scikit-learn*) to dane klasyfikacyjne dotyczące jakości wina na podstawie analizy chemicznej. **Digits** (*scikit-learn*) zawiera dane klasyfikacyjne z obrazami cyfr w formie macierzy pikseli. **Diabetes** (*scikit-learn*) to dane regresyjne o cukrzycy, uwzględniające cechy demograficzne i biomedyczne. **Linnerud** (*scikit-learn*) obejmuje dane wielowymiarowe dotyczące parametrów fitnessowych. **Auto MPG** (Kaggle) zawiera dane regresyjne o zużyciu paliwa samochodów na podstawie ich parametrów [1]. **Auto Insurance in Sweden** (Kaggle) to dane regresyjne o kosztach ubezpieczeń samochodowych w Szwecji [2]. **Blood Transfusion Dataset** (Kaggle) opisuje dane klasyfikacyjne dotyczące skłonności dawców do oddania krwi [3].

### 2.1 Przetwarzanie danych

Dane zostały znormalizowane przy użyciu *StandardScaler*, aby zapewnić, że wszystkie cechy mają średnią 0 i odchylenie standardowe 1, co zapobiega dominacji cech o dużej skali w procesie optymalizacji hiperparametrów. W przypadku zbiorów z Kaggle konieczne było dodatkowe wstępne przetwarzanie, takie jak usunięcie brakujących wartości i konwersja danych do formatu numerycznego.

## 3 Metodyka

### 3.1 Wybrane Algorytmy

Tabela 1 przedstawia zestawienie algorytmów uczenia maszynowego wykorzystanych w badaniu wraz z optymalizowanymi hiperparametrami i ich zakresami.

Tabela 1: Wybrane algorytmy i zakresy hiperparametrów

Algorytm	Hiperparametr	Zakres wartości
Random Forest	<b>n_estimators</b>	10, 50, 100, 200
	<b>max_depth</b>	5, 10, 20, 30
	<b>min_samples_split</b>	2, 5, 10
	<b>min_samples_leaf</b>	1, 2, 4
XGBoost	<b>n_estimators</b>	50, 100, 200
	<b>learning_rate</b>	0.01, 0.1, 0.2
	<b>max_depth</b>	3, 5, 7, 10
	<b>subsample</b>	0.6, 0.8, 1.0
ElasticNet	<b>alpha</b>	0.01, 0.1, 1.0
	<b>l1_ratio</b>	0.1, 0.5, 0.9
	<b>max_iter</b>	1000, 2000, 5000
Gradient Boosting	<b>n_estimators</b>	50, 100, 200
	<b>learning_rate</b>	0.01, 0.1, 0.2
	<b>max_depth</b>	3, 5, 7

## 3.2 Techniki Tunowania

W eksperymencie zastosowano trzy metody optymalizacji hiperparametrów:

**Grid Search** — Klasyczne przeszukiwanie siatki hiperparametrów, testujące wszystkie możliwe kombinacje zadanych wartości. Wykorzystano `GridSearchCV` z biblioteki `scikit-learn` z 3-krotną walidacją krzyżową.

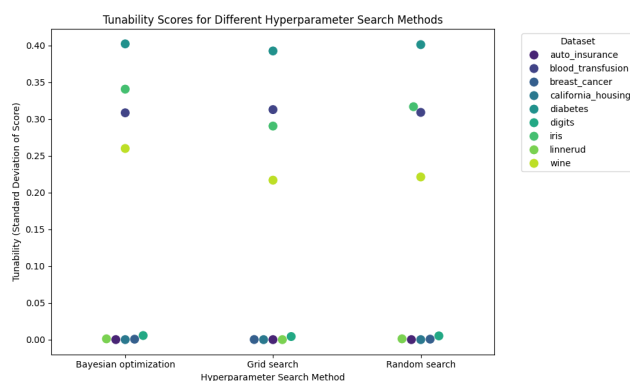
**Random Search** — Losowy wybór kombinacji hiperparametrów, pozwalający na szybsze przeszukiwanie większej przestrzeni parametrów. Użyto `RandomizedSearchCV` z `scikit-learn`.

**Optymalizacja Bayesowska** — Zaawansowana metoda wykorzystująca modele probabilistyczne do wyboru najbardziej obiecujących obszarów przestrzeni parametrów. Implementacja oparta na `BayesSearchCV` z `scikit-optimize` aktualizowała priorytety na podstawie wyników kolejnych iteracji.

## 4 Wyniki

Na rysunku 1 przedstawiono tunowalność algorytmów dla różnych metod tunowania hiperparametrów. Na rysunku 2 zilustrowano rozkład wyników uzyskanych przez każdą z metod. Zbieżność wyników w zależności od liczby iteracji pokazano na rysunku 3, a różnice krytyczne pomiędzy metodami tunowania hiperparametrów przedstawiono na rysunku 4. Na rysunkach przedstawiono wyniki analiz, ilustrujące tunowalność algorytmów, porównanie metod tunowania, zbieżność wyników oraz różnice krytyczne między metodami.

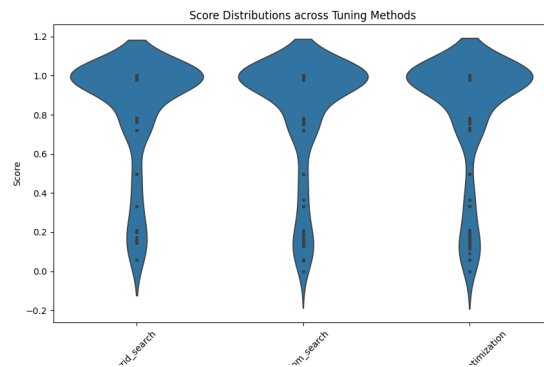
### 4.1 Tunowalność Algorytmów



Rysunek 1: Wykres przedstawia tunowalność algorytmów mierzonych jako odchylenie standardowe wyników modelu na różnych zbiorach danych dla trzech metod tunowania hiperparametrów: Optymalizacji Bayesowskiej, Grid Search i Random Search. Każdy punkt reprezentuje wynik dla jednej kombinacji algorytmu i zbioru danych. Pozycja pionowa punktu pokazuje stopień tunowalności, gdzie wyższe wartości oznaczają większą wrażliwość algorytmu na zmiany hiperparametrów. Wyniki wskazują, że Optymalizacja Bayesowska charakteryzuje się bardziej przewidywalnymi wynikami niż Grid Search i Random Search, które mają większe rozproszenie.

XGBoost i Gradient Boosting wykazują wyższą tunowalność w porównaniu z ElasticNet. Tunowalność była największa dla Optymalizacji Bayesowskiej.

## 4.2 Porównanie Metod Tunowania



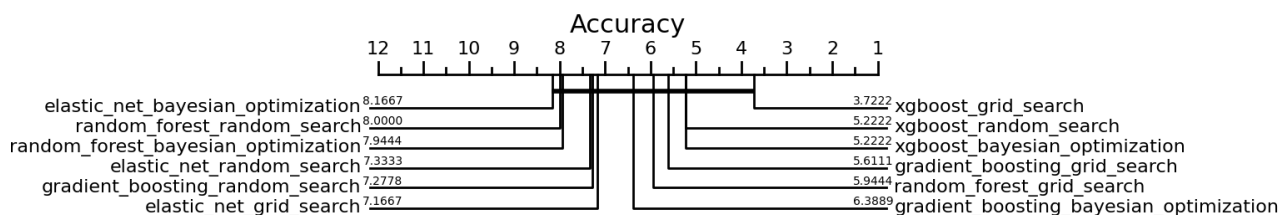
Rysunek 2: Wykres skrzypcowy przedstawia rozkład wyników dla trzech metod tunowania hiperparametrów: Grid Search, Random Search i Optymalizacji Bayesowskiej. Każdy "skrzypiec" reprezentuje zakres wyników (wartości metryk) uzyskanych przez daną metodę. Grubość skrzypca wskazuje częstotliwość wyników o określonej wartości, a punkty wewnętrzne reprezentują konkretne obserwacje. Wykres pokazuje, że Optymalizacja Bayesowska osiąga bardziej stabilne wyniki z mniejszym rozrzutem w porównaniu do Grid Search i Random Search, które wykazują większe zróżnicowanie wyników.

## 4.3 Konwergencja Wyników



Rysunek 3: Wykres przedstawia zbieżność wyników dla trzech metod tunowania hiperparametrów: Grid Search, Random Search oraz Optymalizacji Bayesowskiej, na różnych zbiorach danych. Oś pozioma reprezentuje liczbę iteracji, a oś pionowa przedstawia średni wynik modelu (np. ROC AUC). Krzywe dla poszczególnych zbiorów danych ilustrują, jak szybko wyniki stabilizują się w trakcie iteracji. Optymalizacja Bayesowska wykazuje szybszą zbieżność w porównaniu do pozostałych metod, co sugeruje jej większą efektywność w uzyskiwaniu stabilnych i optymalnych wyników. Grid Search wymaga najwięcej iteracji, co wiąże się z większym kosztem obliczeniowym.

## 4.4 Diagram Różnic Krytycznych



Rysunek 4: Diagram różnic krytycznych (CD-Diagram) przedstawia porównanie metod tunowania hiperparametrów dla różnych modeli na podstawie ich średnich rang. Oś pozioma reprezentuje średnie rangi poszczególnych metod, gdzie niższa wartość oznacza lepszą skuteczność. Linie łączące metody wskazują brak statystycznie istotnych różnic między nimi według testu Wilcoxona z poprawką Holm-Bonferroniego. Diagram ilustruje, że metody o najwyższych rangach, takie jak `xgboost_grid_search`, osiągają lepsze wyniki w porównaniu do mniej skutecznych metod, takich jak `elastic_net_grid_search`. Diagram został wygenerowany na podstawie wyników 10 zbiorów danych, z wykorzystaniem zaawansowanej analizy statystycznej.

Diagram różnic krytycznych przedstawia analizę post-hoc przeprowadzoną na podstawie testu Friedmana, który odrzucił hipotezę zerową o braku różnic między metodami. Następnie zastosowano test Wilcoxona z poprawką Holm-Bonferroniego do analizy parowych różnic między metodami. Grube linie poziome grupują metody, między którymi nie stwierdzono statystycznie istotnych różnic. Widać, że `xgboost_grid_search` oraz `xgboost_bayesian_optimization` należą do grupy o najwyższej skuteczności, podczas gdy `elastic_net_grid_search` osiąga najniższe rangi.

## 5 Wnioski

- Zakresy hiperparametrów dla poszczególnych algorytmów, takie jak `n_estimators` w XGBoost czy `l1_ratio` w ElasticNet, zostały dostosowane na podstawie literatury oraz wyników eksperymentalnych, co potwierdziło ich znaczenie w procesie optymalizacji.
- Algorytmy takie jak XGBoost i Gradient Boosting wykazały wyższą tunowalność, co wskazuje na ich większą podatność na dostosowanie parametrów, podczas gdy algorytmy takie jak ElasticNet wykazywały mniejszy wpływ zmian hiperparametrów na wyniki.
- Metody losowania punktów, takie jak Random Search i Grid Search, wprowadzały większe rozproszenie wyników w porównaniu do Optymalizacji Bayesowskiej. Stwierdzono, że technika samplingu może wprowadzać bias, szczególnie dla bardziej złożonych algorytmów, takich jak XGBoost.
  - Diagram różnic krytycznych potwierdza, że metody optymalizacji dla XGBoost (*grid search* i *bayesian optimization*) osiągają najlepsze średnie rangi, co wskazuje na ich wysoką skuteczność w strojeniu hiperparametrów.
  - W przypadku algorytmu ElasticNet różnice między metodami tunowania były mniej istotne, co sugeruje, że wpływ hiperparametrów na wydajność modelu jest w tym przypadku ograniczony.
  - Grube linie na diagramie wskazują grupy metod, które są statystycznie równoważne pod względem wyników, co oznacza brak znaczących różnic między np. `random_forest_bayesian_optimization` a `elastic_net_random_search`.
  - Wyniki sugerują, że bardziej zaawansowane metody tunowania, takie jak Optymalizacja Bayesowska, mogą znacząco poprawić wyniki dla złożonych modeli, takich jak XGBoost, w porównaniu do bardziej tradycyjnych metod, jak Grid Search.

## Literatura

- [1] UCI Machine Learning Repository. Auto mpg dataset, 2020.
- [2] Kaggle Community. Auto insurance in sweden, 2020.
- [3] Kaggle Community. Blood transfusion dataset, 2020.