

# Tunowalność algorytmów uczenia maszynowego

## PD 1 AutoML

Wojciech Kosiuk  
Politechnika Warszawska

Michał Mazuryk  
Politechnika Warszawska

Listopad 2023

## 1 Abstrakt

Praca zawiera podejście do analizy tunowalności algorytmów i hiperparametrów uczenia maszynowego na przykładzie Random Forest, XGBoost i LightGBM. Porównywane są w niej również różne metody samplingu, Grid Search, Random Search, Bayes Optimization.

## 2 Wstęp

W dziedzinie uczenia maszynowego, złożone algorytmy wymagają precyzyjnego dostrojenia hiperparametrów przed procesem treningu. Ich konfiguracja jest kluczowa dla osiągnięcia optymalnej wydajności. Wybór odpowiednich hiperparametrów może odbywać się poprzez użycie domyślnych wartości z pakietów bibliotek, manualne dostrojenie lub zastosowanie strategii optymalizacji hiperparametrów. Pojawia się pytanie jak podejście do wyboru parametrów będzie wpływało na końcowy wynik?

## 3 Powiązane prace

Artykuł Tunability: Importance of Hyperparameters of Machine Learning Algorithms przedstawia definicje tunowalności zarówno algorytmu, jak i hiperparametru. Opisuje wyznaczanie konfiguracji domyślnych. Tunowalność będzie różnicą między AUC na defaultowych parametrach dla datasetu, a średnią z najlepszych wyników otrzymanych na różnych ziarnach. Tunowalność dla hiperparametru definiujemy jako różnicę między defaultowym AUC, a średnią z najlepszych wyników po ziarnach przy zmianie wartości tylko tego jednego hiperparametru z ustawień defaultowych.

## 4 Zbiory danych do testów

Do wyznaczenia konfiguracji domyślnych oraz testowania algorytmów użyliśmy 4 zbiorów danych:

- **SPECT heart data** - binarna klasyfikacja, zbiór wyekstraktowanych feature'ów ze zdjęć pacjentów, zmienna objaśniana - diagnoza pacjenta (0,1)
- **shrutime** - binarna klasyfikacja, klasyfikacja, czy klient banku zamknie swoje konto
- **bank-marketing**: binarna klasyfikacja, dane z portugalskich banków, predykcja, czy klient złożył depozyt w banku
- **eeg-eye-state** - binarna klasyfikacja, dane pochodzące z badania mózgu EEG, klasyfikacja, czy pacjent w trakcie badania ma oczy otwarte, czy zamknięte

## 5 Znajdowanie konfiguracji domyślnych

Najpierw dla każdego modelu znaleziono zbiór parametrów domyślnych. Są to takie konfiguracje, dla których algorytm miał średnio największy wynik na przestrzeni testowanych datasetów. Jest to pewna symulacja mająca na celu wyłonienie najbardziej uniwersalnych parametrów, do których w następnych sekcjach będziemy się porównywać. Przestrzeń i zakres szukanych parametrów został wybrany na podstawie doświadczenia i literatury, a następnie zostały utworzone modele dla każdej konfiguracji używając Grid Search przy  $CV = 3$ .

### 5.1 Random Forest

Pierwszym testowanym modelem jest Random Forest, wybrane dla niego parametry do konfiguracji to:

- **n\_estimators**: [50, 100, 200],
- **max\_depth**: [5, 10, 20],
- **min\_samples\_split**: [2, 5, 10],
- **min\_samples\_leaf**: [2, 5, 8].

Najwyższy średnik wynik AUC otrzymany na zbiorach to 0.7569, dla **max\_depth**: 10, **min\_samples\_leaf**: 8, **min\_samples\_split**: 2, **n\_estimators**: 100. Będziemy je uznawać za defaultowe.

n_estim	max_depth	min_samples_split	min_samples_leaf	Średnie AUC	AUC dla 1. zbioru	AUC dla 2. zbioru	AUC dla 3. zbioru	AUC dla 4. zbioru
100	10	2	8	0.7569	0.8419	0.8618	0.8292	0.4947
50	5	10	2	0.7565	0.8471	0.8511	0.8105	0.5173
200	5	5	8	0.7562	0.8442	0.849526	0.814950	0.516149
200	10	2	5	0.7561	0.8440	0.862086	0.828217	0.489914

Tabela 1: Wyniki uczenia dla najlepszych (największe średnie AUC) konfiguracji parametrów z wybranej przestrzeni dla Random Forest.

## 5.2 XGBoost

Kolejnym modelem jest XGBoost, wybrane dla niego parametry do konfiguracji to:

- `n_estimators`: [50, 100, 200],
- `max_depth`: [3, 5, 7, 10],
- `learning_rate`: [0.001, 0.005, 0.01, 0.02].

n_estim	max_depth	learning_rate	Średnie AUC	AUC dla 1. zbioru	AUC dla 2. zbioru	AUC dla 3. zbioru	AUC dla 4. zbioru
200	3	0.02	0.7525	0.8248	0.8610	0.8263	0.4979
200	3	0.01	0.7475	0.8131	0.8515	0.8152	0.5101
100	3	0.02	0.7467	0.8113	0.8514	0.8151	0.5090
100	5	0.005	0.7450	0.8099	0.8501	0.8214	0.4985

Tabela 2: Wyniki uczenia dla najlepszych (największe średnie AUC) konfiguracji parametrów z wybranej przestrzeni dla XGBoost.

Najwyższy średnik wynik AUC otrzymany na zbiorach to 0.7525, dla `n_estimators`: 200, `max_depth`: 3, `learning_rate`: 0.02. Będziemy je uznawać za defaultowe.

## 5.3 LightGBM

Ostatni porównywany przez nas model to LightGBM, wybrane dla niego parametry do konfiguracji to:

- `n_estimators`: [20, 100, 300],
- `max_depth`: [3, 7, 10],
- `learning_rate`: [0.01, 0.1, 0.3],
- `num_leaves`: [10, 31, 65, 100].

n_estim	max_depth	learning_rate	num_leaves	Średnie AUC	AUC dla 1. zbioru	AUC dla 2. zbioru	AUC dla 3. zbioru	AUC dla 4. zbioru
300	3	0.01	31	0.7508	0.8194	0.8567	0.8233	0.5037
300	3	0.01	10	0.7508	0.8194	0.8567	0.8233	0.5037
300	3	0.01	100	0.7508	0.8194	0.8567	0.8233	0.5037
300	3	0.01	65	0.7508	0.8194	0.8567	0.8233	0.5037

Tabela 3: Wyniki uczenia dla najlepszych (największe średnie AUC) konfiguracji parametrów z wybranej przestrzeni dla LightGBM.

Najwyższy średnik wynik AUC otrzymany na zbiorach to 0.7508, uzyskano go dla 4 zbiorów parametrów, więc za defaultowy będziemy przyjmować jeden z nich, `n_estimators: 300`, `max_depth: 3`, `learning_rate: 0.01`, `num_leaves: 31`.

## 6 Random Search

Kolejnym krokiem jest użycie metody Random Search i porównanie z wynikami defaultowymi. Przestrzenie parametrów, w których będziemy poszukiwać zostały rozszerzone do przestrzeni zawierających zakresy z Grid Search. Random Search został uruchamiany dla każdego zbioru 3 razy (na 3 różnych ziarnach), więc porównywanym wynikiem z defaultem jest średnia najlepszych uzyskanych AUC na przestrzeni ziaren. Dodatkowe ustawione opcje Random Searcha to `CV = 3` oraz liczba iteracji = 40.

### 6.1 Random Forest

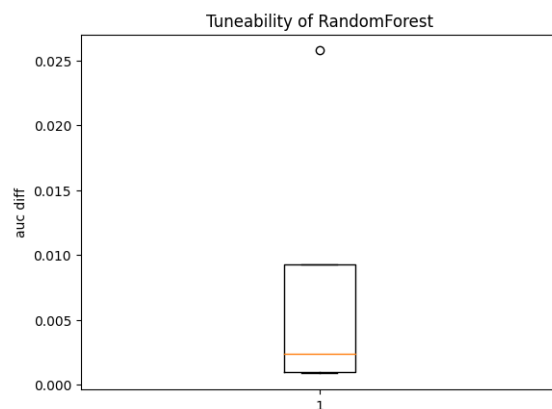
Przestrzeń parametrów:

- `n_estimators`: od 50 do 200,
- `max_depth`: od 5 do 20,
- `min_samples_split`: od 2 do 10,
- `min_samples_leaf`: od 2 do 8.

Nr zbioru	Średnie AUC	AUC dla 1. ziarna	AUC dla 2. ziarna	AUC dla 3. ziarna
1	0.8457	0.8446	0.8451	0.8474
2	0.8628	0.8627	0.8628	0.8628
3	0.8300	0.8296	0.8304	0.8301
4	0.5204	0.5214	0.5236	0.5162

Tabela 4: Najlepsze wyniki uczenia dla każdego z 3 testowanych ziaren w Random Forest dla wyliczenia tunowalności algorytmu.

Wyniki po wyborze parametrów z Random Search wyszły lepsze niż dla defaultów.



Rysunek 1: Wykres kwantylowy dla tunowalności Random Forest. Średnia tunowalność na przestrzeni zbiorów wynosi 0.00786.

## 6.2 XGBoost

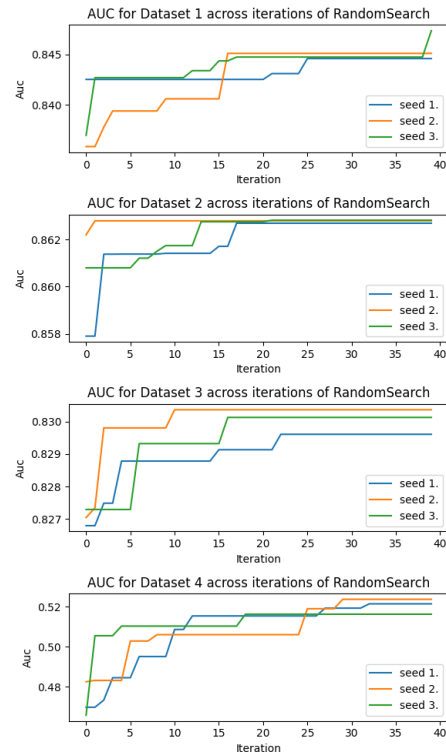
Przestrzeń parametrów:

- `n_estimators`: od 50 do 200,
- `max_depth`: od 3 do 10,
- `learning_rate`: od 0.001 do 0.02.

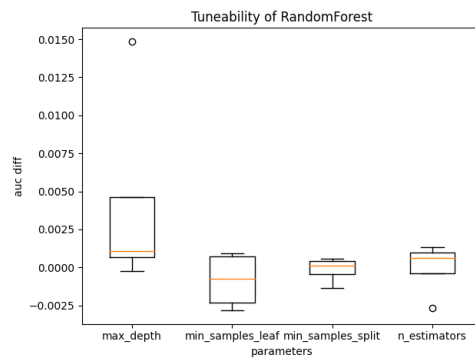
Nr zbioru	Średnie AUC	AUC dla 1. ziarna	AUC dla 2. ziarna	AUC dla 3. ziarna
1	0.8256	0.8295	0.8238	0.8235
2	0.8657	0.8659	0.8658	0.8652
3	0.8300	0.8307	0.8308	0.8283
4	0.4925	0.4984	0.5016	0.4775

Tabela 5: Najlepsze wyniki uczenia dla każdego z 3 testowanych ziaren w XGBoost dla wyliczenia tunowalności algorytmu.

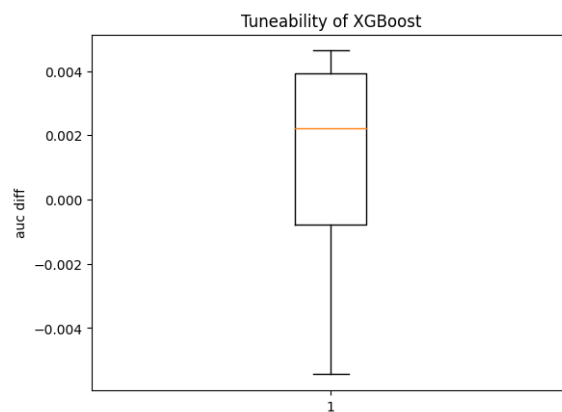
Wyniki po wyborze parametrów z Random Search wyszły lepsze niż dla defaultów.



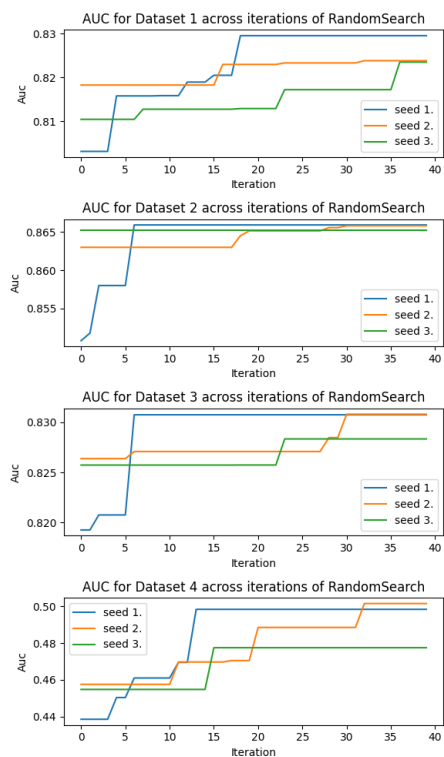
Rysunek 2: Wykres przedstawiający jak liczba iteracji Random Search wpływała na osiągnięte AUC. Aby osiągnąć stabilne wyniki potrzeba było 25-30 iteracji.



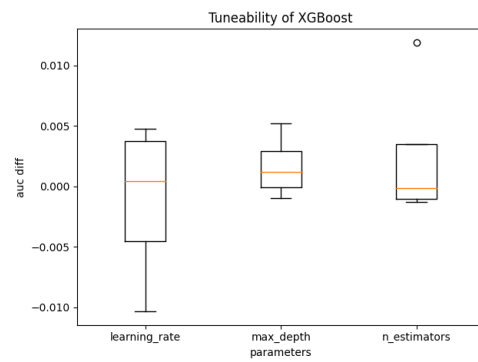
Rysunek 3: Wykres kwantylowy tunowalności hiperparametrów. Średnia tunowalność max\_depth to 0.0042, min\_samples\_leaf -0.0008, min\_samples\_split -0.0001, n\_estimators 0



Rysunek 4: Wykres kwantylowy dla tunowalności XGBoost. Średnia tunowalność na przestrzeni zbiorów wynosi 0.0010.



Rysunek 5: Wykres przedstawiający jak liczba iteracji Random Search wpływała na osiągnięte AUC. Aby osiągnąć stabilne wyniki potrzeba było 25-30 iteracji.



Rysunek 6: Wykres kwantylowy tunowalności hiperparametrów. Średnia tunowalność learning\_rate to -0.0012, max\_depth 0.0016, n\_estimators 0.0026



### 6.3 LightGBM

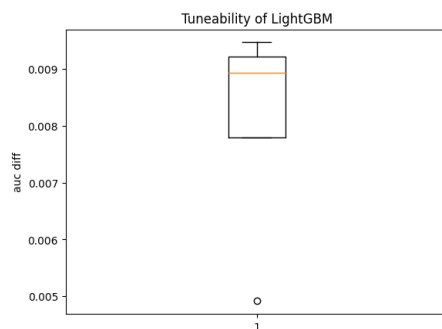
Przestrzeń parametrów:

- `n_estimators`: od 20 do 300,
- `max_depth`: od 3 do 10,
- `learning_rate`: od 0.01 do 0.3,
- `num_leaves`: od 10 do 100.

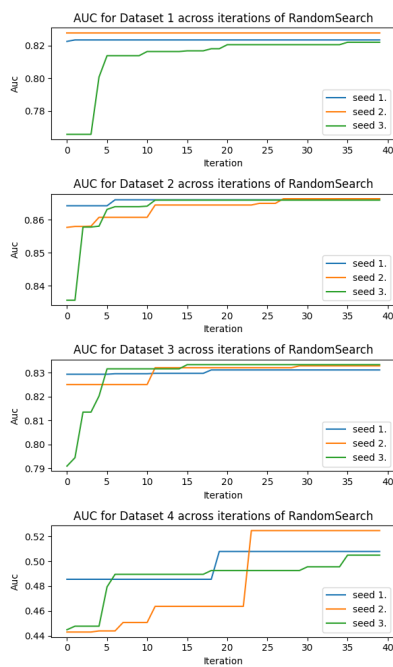
Nr zbioru	Średnie AUC	AUC dla 1. ziarna	AUC dla 2. ziarna	AUC dla 3. ziarna
1	0.8243	0.8234	0.8277	0.8219
2	0.8662	0.8663	0.8663	0.8660
3	0.8324	0.8311	0.8328	0.8333
4	0.5124	0.5078	0.5246	0.5049

Tabela 6: Najlepsze wyniki uczenia dla każdego z 3 testowanych ziaren w LightGBM dla wyliczenia tunowalności algorytmu.

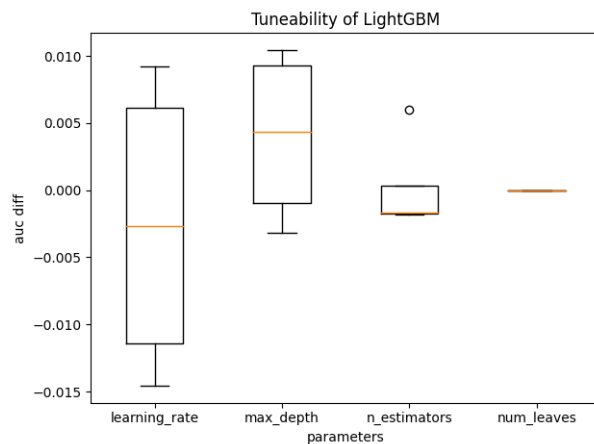
Wyniki po wyborze parametrów z Random Search wyszły lepsze niż dla defaultów.



Rysunek 7: Wykres kwantylowy dla tunowalności LightGBM. Średnia tunowalność na przestrzeni zbiorów wynosi 0.0081.



Rysunek 8: Wykres przedstawiający jak liczba iteracji Random Search wpływała na osiągnięte AUC. Aby osiągnąć stabilne wyniki potrzeba było 20-25 iteracji.



Rysunek 9: Wykres kwantylowy tunowalności hiperparametrów. Średnia tunowalność learning\_rate to -0.0027, max\_depth 0.0040, n\_estimators 0.0002, num\_leaves 0

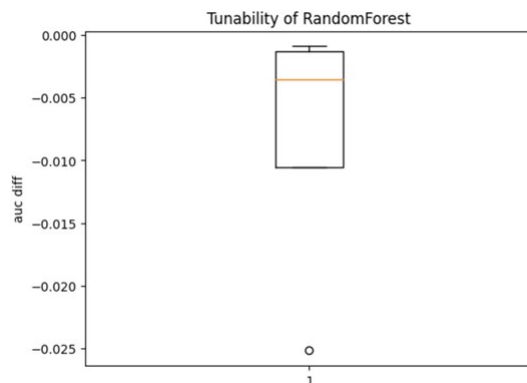
## 7 Optimizacja Bayesowska

Ostatnim krokiem było zastosowanie optymalizacji bayesowskiej do strojenia hiperparametrów i wyznaczenia tunowalności. Przestrzeń hiperparametrów pokrywa się z tą, użytą w Random Search. Eksperyment został wykończony dla każdego zbioru 3 razy, a miarą było AUC. Liczba iteracji została wyznaczona dla każdego algorytmu oddzielnie.

### 7.1 Random Forest

#### 7.1.1 Tunowalność algorytmu

Optymalizacja bayesowska została wykonana 3-krotnie (3 różne ziarna) na 4 datasetach z liczbą iteracji 10. Wyniki okazały się nieznacznie, ale jednak gorsze niż dla konfiguracji domyślnej.



Rysunek 10: Wykres kwantylowy dla tunowalności Random Forest.

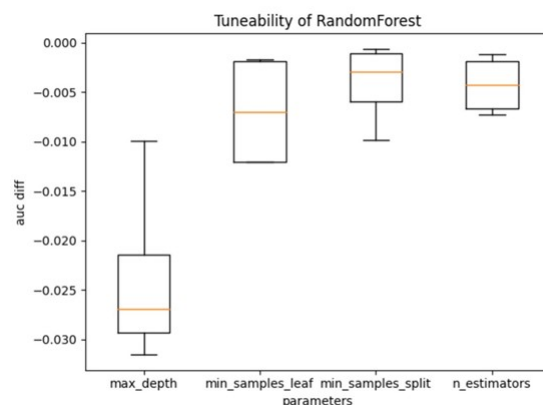
#### 7.1.2 Tunowalność hiperparametrów

Optymalizacja każdego hiperparametru została wyznaczona również za pomocą 3 różnych ziaren oraz 10 iteracji. Wszystkie parametry poza optymalizowanym zostały ustawione na te z konfiguracji domyślnej. Żaden z eksperymentów nie uzyskał wyników lepszych niż konfiguracja domyślna.

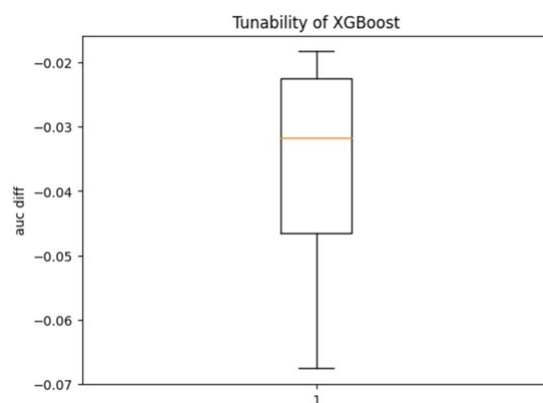
### 7.2 XGBoost

#### 7.2.1 Tunowalność algorytmu

Optymalizacja bayesowska została wykonana 3-krotnie (3 różne ziarna) na 4 datasetach z liczbą iteracji 100. Wyniki nie są satysfakcjonujące i są gorsze niż te dla konfiguracji domyślnej.



Rysunek 11: Wykres kwantylowy tunowalności hiperparametrów Random Forest.



Rysunek 12: Wykres kwantylowy dla tunowalności XGBoost.

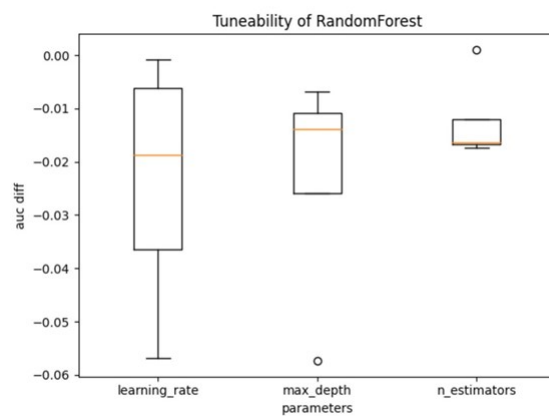
## 7.2.2 Tunowalność hiperparametrów

Optymalizacja każdego hiperparametru została wyznaczona również za pomocą 3 różnych ziaren oraz 20 iteracji. Wszystkie parametry poza optymalizowanym zostały ustawione na te z konfiguracji domyślnej. Żaden z eksperymentów nie uzyskał wyników lepszych niż konfiguracja domyślna.

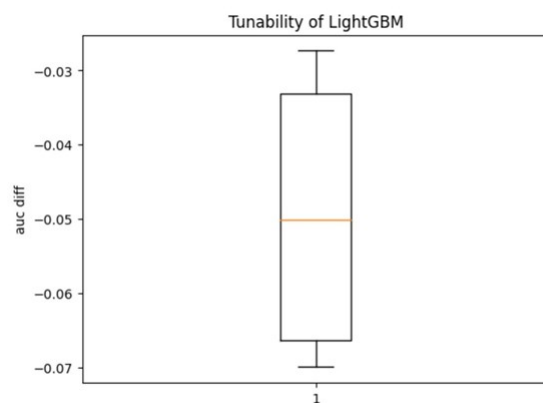
## 7.3 LightGBM

### 7.3.1 Tunowalność algorytmu

Optymalizacja bayesowska została wykonana 3-krotnie (3 różne ziarna) na 4 datasetach z liczbą iteracji 200. Ponownie wyniki są zróżnicowane w zależności od użytego ziarna, oraz nie przebijają wyników konfiguracji domyślnych.



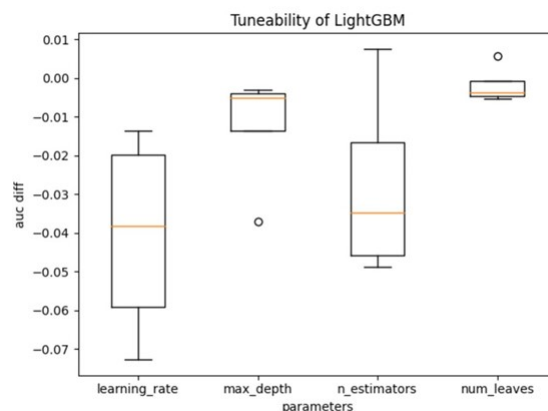
Rysunek 13: Wykres kwantylowy tunowalności hiperparametrów XGBoost.



Rysunek 14: Wykres kwantylowy dla tunowalności LightGBM.

### 7.3.2 Tunowalność hiperparametrów

Optymalizacja każdego hiperparametru została wyznaczona również za pomocą 3 różnych ziaren oraz 20 iteracji. Jest to jedyny przypadek kiedy optymalizacja bayesowska dała minimalnie lepsze wyniki niż konfiguracje domyślne dla hiperparametrów - num\_leaves oraz n\_estimators.



Rysunek 15: Wykres kwantylowy tunowalności hiperparametrów LightGBM.

## 8 Podsumowanie

### 8.1 Ile iteracji metody potrzebujemy, aby uzyskać stabilne wyniki optymalizacji?

Podczas stosowania metody Random Search, zarówno Random Forest, jak i XGBoost potrzebowały około 25-30 iteracji do osiągnięcia stabilnych wyników. Algorytm LightGBM potrzebował do tego 20-25 iteracji.

### 8.2 Tunowalność algorytmów

Brak zdecydowanych różnic pomiędzy algorytmami. Nieznacznie lepiej od 2 pozostałych poradził sobie Random Forest.

### 8.3 Tunowalność hiperparametrów

Dla metody Random Search, strojenie poszczególnych hiperparametrów działało gorzej niż tunowanie wszystkich naraz. Dla metody bayesowskiej metoda ta miała więcej sensu i dawała dla algorytmu LightGBM lepszy wynik niż konfiguracja domyślna. Porównywanie tunowalności hiperparametrów pomiędzy sobą ma sens i pozwala wywnioskować, którymi hiperparametrami najlepiej się zająć, bo mogą mieć największy wpływ na skuteczność algorytmu.

### 8.4 Czy technika losowania punktów wpływa na różnice

Metoda Random Search pozwala na uzyskanie wyników minimalnie lepszych od konfiguracji domyślnych. Optymalizacja bayesowska nie pozwoliła na uzyskanie pozytywnych wartości tunowalności. Metoda ta działa stosunkowo wolno, i wedle naszych eksperymentów, mało skutecznie. Jednymi z potencjalnych problemów, przez który optymalizacja bayesowska mogła w naszym przypadku nie okazać

się efektywna jest: użycie nieoptymalnego pakietu - kilka pakietów oferuje rozwiązanie do strojenia hiperparametrów za pomocą optymalizacji bayesowskiej, oraz sprawdzenie zbyt małej liczby konfiguracji - tu głównymi ograniczeniami były zasoby komputerowe i czasowe.