Analiza Tunowalności Wybranych Algorytmów Nauczania Maszynowego

Optymalizacja Hiperparametrów

Andrii Voznesenskyi

14 listopada 2024

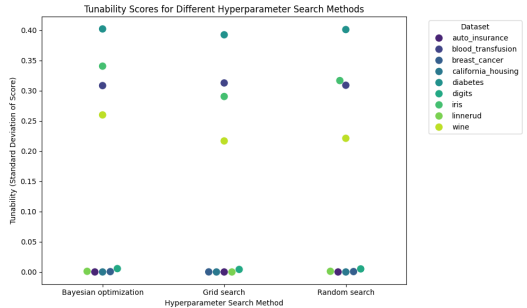
- Zbadanie tunowalności hiperparametrów algorytmów:
  - XGBoost
  - Random Forest
  - ElasticNet
  - Gradient Boosting
- Porównanie trzech metod tunowania:
  - **Grid Search**
  - **Random Search**
  - **Optymalizacja Bayesowska**
- Ocena stabilności i efektywności wyników.

Tabela: Wybrane algorytmy i zakresy hiperparametrów

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

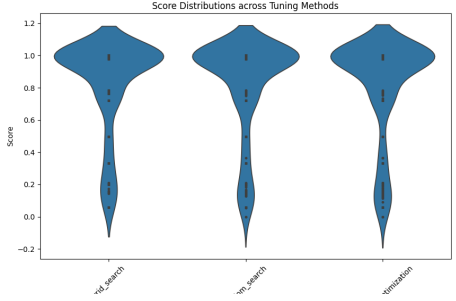
- **Grid Search** (z biblioteki `scikit-learn`):
  - Implementacja: `GridSearchCV` z `sklearn.model_selection`.
  - Przeszukiwanie pełnej siatki parametrów, testując wszystkie możliwe kombinacje.
  - Używana w przypadku modeli o niewielkiej liczbie hiperparametrów lub w początkowej fazie analizy.
- **Random Search** (z biblioteki `scikit-learn`):
  - Implementacja: `RandomizedSearchCV` z `sklearn.model_selection`.
  - Losowe próbkowanie przestrzeni parametrów, co pozwala na szybsze przeszukiwanie przy ograniczonej liczbie iteracji.
  - Używana w celu zidentyfikowania potencjalnych regionów przestrzeni parametrów.
- **Optymalizacja Bayesowska** (z biblioteki `scikit-optimize`):
  - Implementacja: `BayesSearchCV` z `skopt`.
  - Wykorzystuje wyniki poprzednich iteracji do inteligentnego wyboru kolejnych punktów.
  - Zapewnia szybszą konwergencję, szczególnie dla modeli o dużej liczbie hiperparametrów.

- Modele zostały dopasowane do rodzaju zadania:
  - **Regresja:**
    - `ElasticNet` (`sklearn.linear_model.ElasticNet`)
  - **Klasyfikacja:**
    - `GradientBoostingClassifier` (`sklearn.ensemble.GradientBoostingClassifier`)
    - `RandomForestClassifier` (`sklearn.ensemble.RandomForestClassifier`)
    - `XGBClassifier` (`xgboost.XGBClassifier`)
- Modele niezgodne z typem zadania (np. użycie modelu regresyjnego dla klasyfikacji) zostały pominięte.
- Dzięki temu analiza zapewnia:
  - Spójność wyników,
  - Unikanie błędów związanych z nieodpowiednim dopasowaniem modelu do danych.

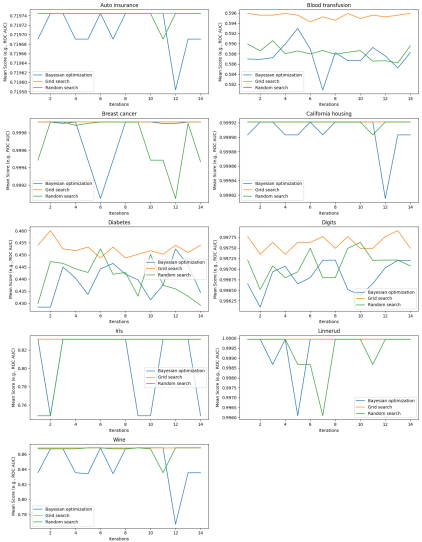


Wykres tunowalności algorytmów dla metod Grid Search, Random Search oraz Optymalizacji Bayesowskiej.

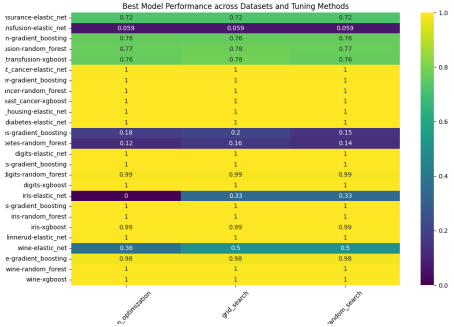
0.30



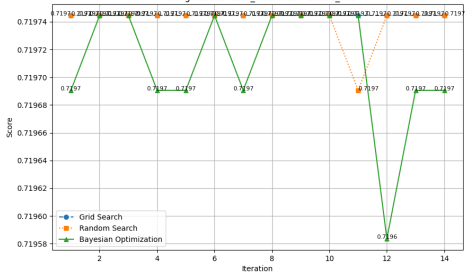
Rozkład wyników uzyskanych przez każdą z metod tunowania.



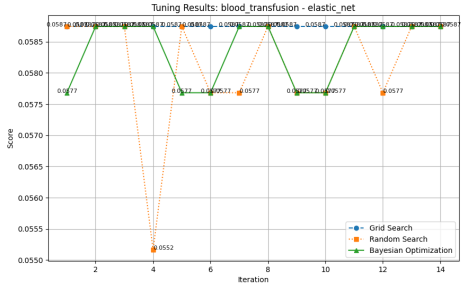
Szybkość zbieżności wyników dla trzech metod tunowania.



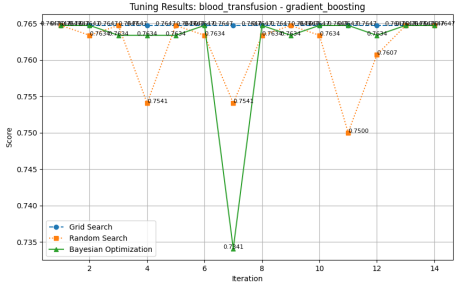
Mapa cieplna porównująca różne modele pod względem kluczowych miar wydajności, wskazując najlepsze konfiguracje.



Rysunek: Wyniki strojenia Elastic Net dla zbioru Auto Insurance.

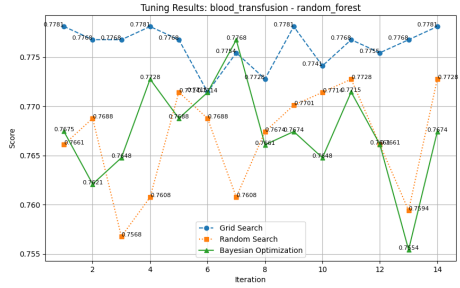


Rysunek: Wyniki strojenia Elastic Net dla zbioru Blood Transfusion.

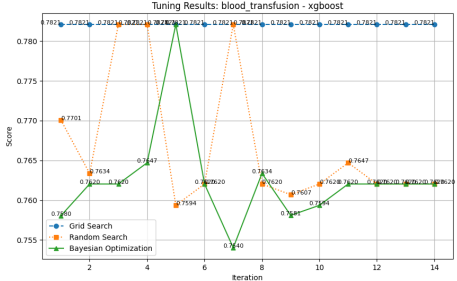


Rysunek: Wyniki strojenia Gradient Boosting dla zbioru Blood Transfusion.

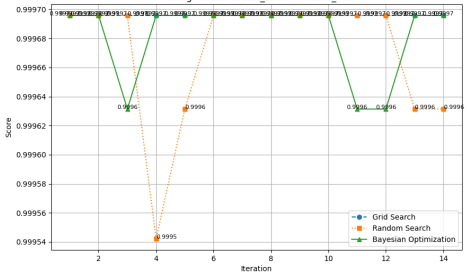
0.41



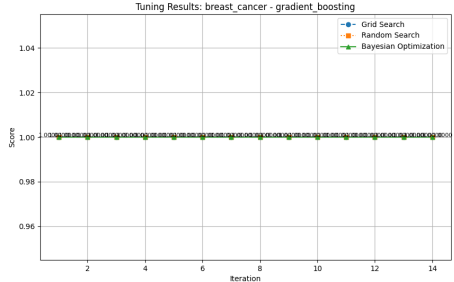
Rysunek: Wyniki strojenia Random Forest dla zbioru Blood Transfusion.



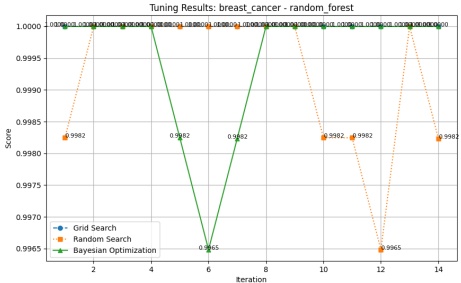
Rysunek: Wyniki strojenia XGBoost dla zbioru Blood Transfusion.



Rysunek: Wyniki strojenia Elastic Net dla zbioru Breast Cancer.

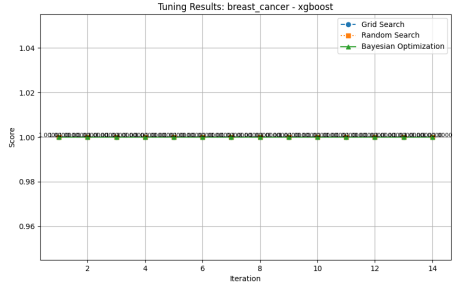


Rysunek: Wyniki strojenia Gradient Boosting dla zbioru Breast Cancer.



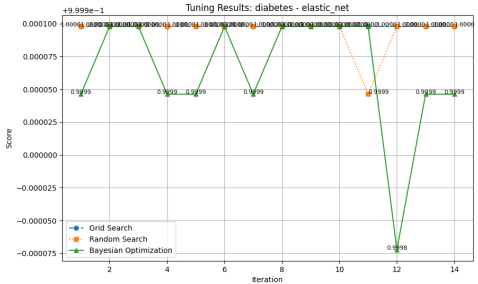
Rysunek: Wyniki strojenia Random Forest dla zbioru Breast Cancer.

0.41

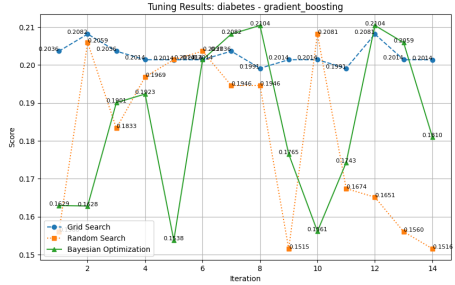


Rysunek: Wyniki strojenia XGBoost dla zbioru Breast Cancer.

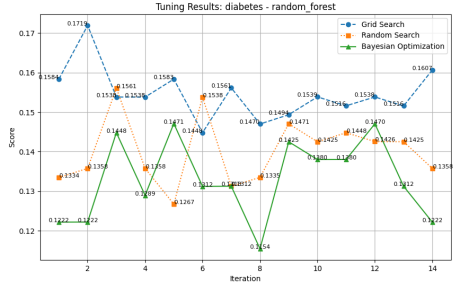
Rysunek: Wyniki strojenia Elastic Net dla zbioru California Housing.



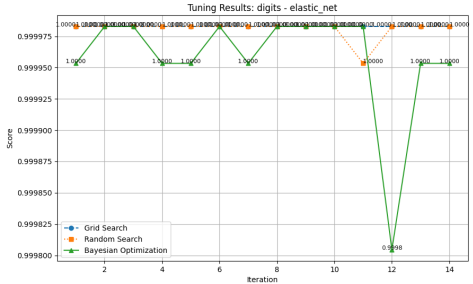
Rysunek: Wyniki strojenia Elastic Net dla zbioru Diabetes.



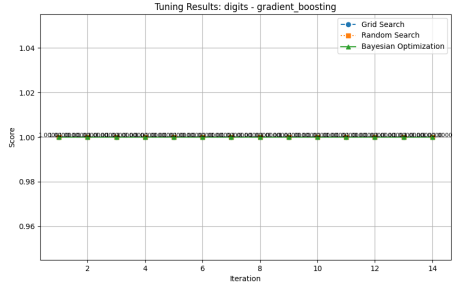
Rysunek: Wyniki strojenia Gradient Boosting dla zbioru Diabetes.



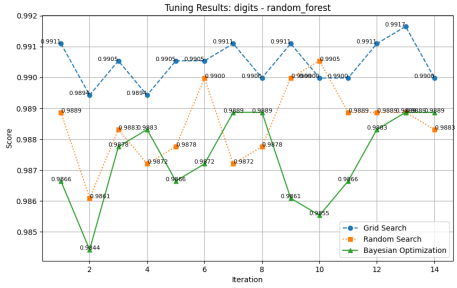
Rysunek: Wyniki strojenia Random Forest dla zbioru Diabetes.



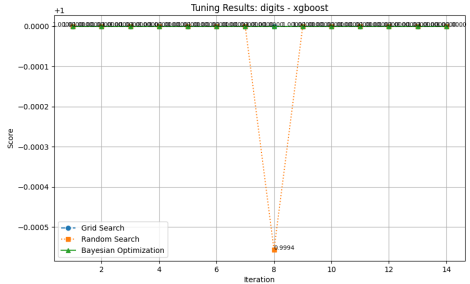
Rysunek: Wyniki strojenia Elastic Net dla zbioru Digits.



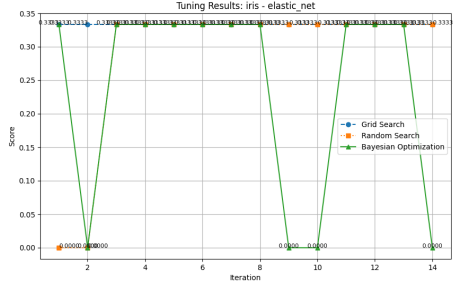
Rysunek: Wyniki strojenia Gradient Boosting dla zbioru Digits.



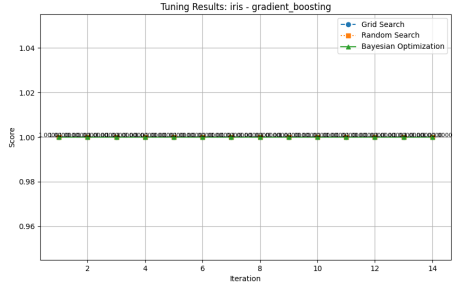
Rysunek: Wyniki strojenia Random Forest dla zbioru Digits.



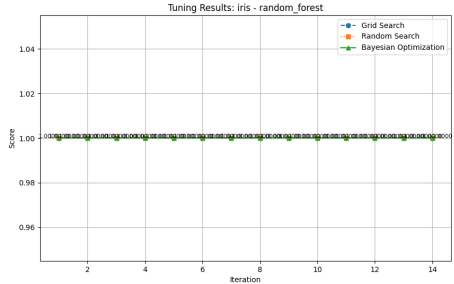
Rysunek: Wyniki strojenia XGBoost dla zbioru Digits.



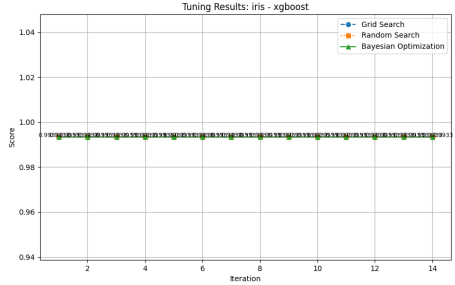
Rysunek: Wyniki strojenia Elastic Net dla zbioru Iris.



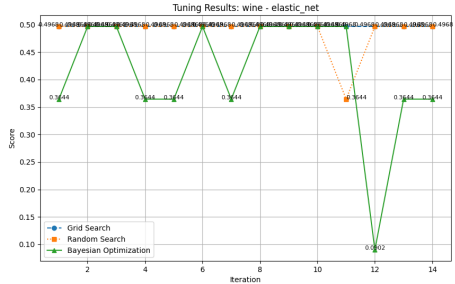
Rysunek: Wyniki strojenia Gradient Boosting dla zbioru Iris.



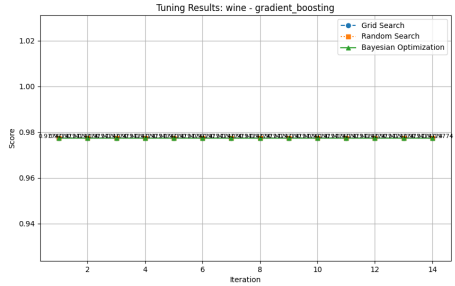
Rysunek: Wyniki strojenia Random Forest dla zbioru Iris.



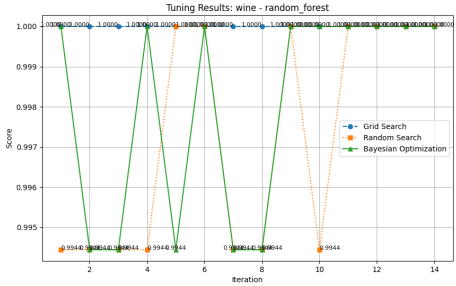
Rysunek: Wyniki strojenia XGBoost dla zbioru Iris.



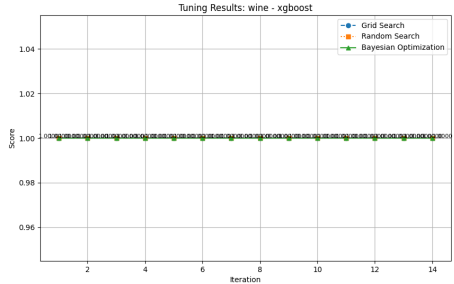
Rysunek: Wyniki strojenia Elastic Net dla zbioru Wine.



Rysunek: Wyniki strojenia Gradient Boosting dla zbioru Wine.



Rysunek: Wyniki strojenia Random Forest dla zbioru Wine.



Rysunek: Wyniki strojenia XGBoost dla zbioru Wine.

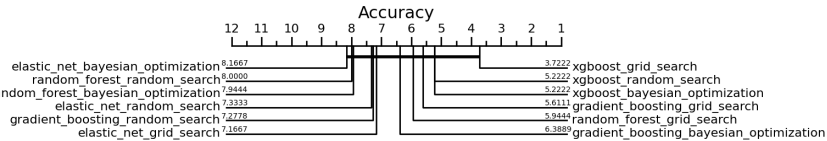
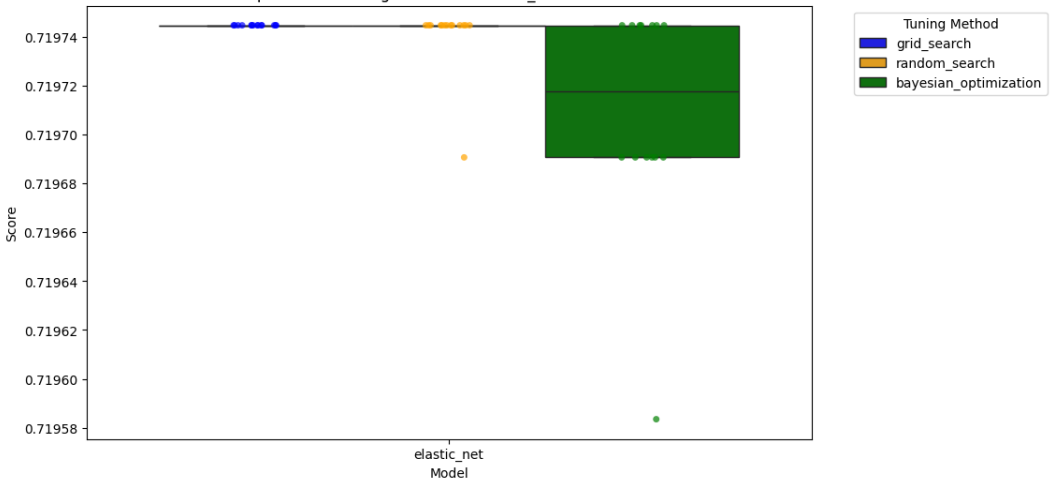


Diagram różnic krytycznych (Critical Difference Diagram) przedstawia wyniki porównania metod tunowania hiperparametrów dla różnych modeli na podstawie ich średnich rang skuteczności. Oś pozioma reprezentuje średnie rangi metod, gdzie niższe wartości wskazują na lepszą skuteczność danej metody.

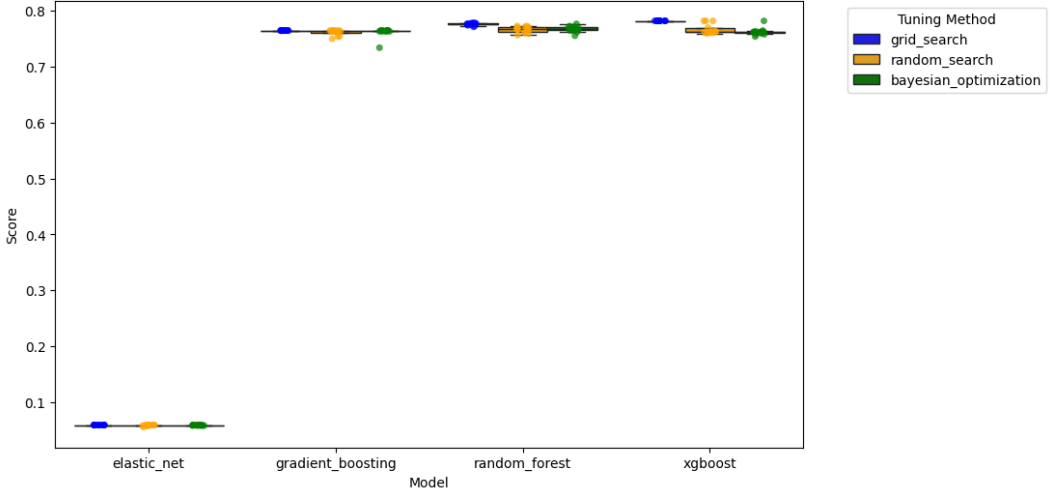
Metody, które są połączone grubymi poziomymi liniami, należą do grup, między którymi nie stwierdzono statystycznie istotnych różnic. Diagram opiera się na testach statystycznych: test Friedmana zastosowany jako globalny test różnic oraz test Wilcoxona z poprawką Holm-Bonferroniego dla porównań parowych.
Najlepsze wyniki uzyskały metody takie jak *xgboost_grid_search* i *xgboost_bayesian_optimization*, które osiągnęły najniższe średnie rangi. Z kolei metody takie jak *elastic_net_grid_search* wykazują wyższą średnią rangę, co wskazuje na ich niższą skuteczność w analizowanych przypadkach. Wyniki te powstały w oparciu o analizy wykonane na kilku zbiorach danych, a diagram ilustruje kluczowe wnioski z tych eksperymentów.

ANOVA?

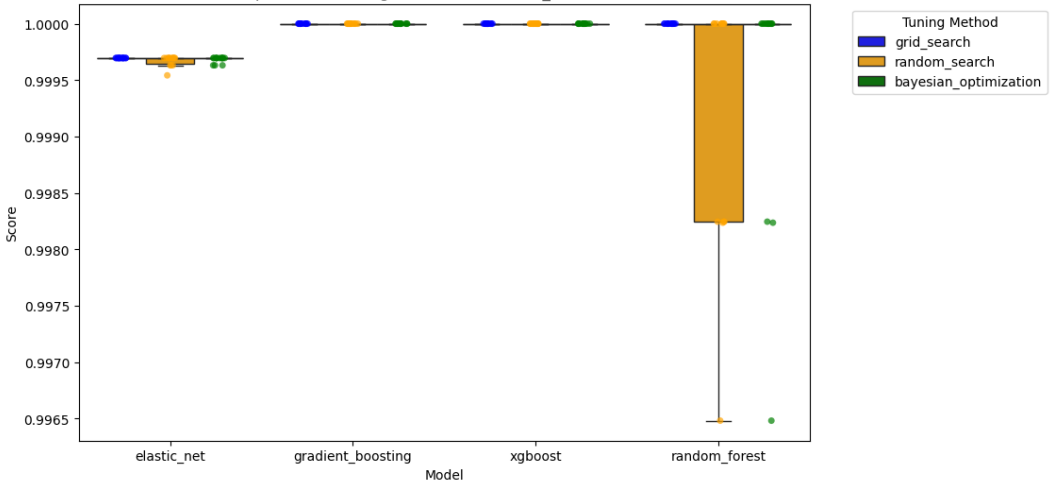
- ANOVA (*Analysis of Variance*) to metoda statystyczna służąca do porównania średnich wyników pomiędzy różnymi grupami.
- W naszym przypadku, grupami są różne metody strojenia hiperparametrów (`grid_search`, `random_search`, `bayesian_optimization`).
- ANOVA pozwala ocenić, czy różnice pomiędzy wynikami strojenia są statystycznie istotne.



Porównanie wyników metod strojenia dla danych o ubezpieczeniach samochodowych. Z wykresu wynika, że `Bayesian Optimization` osiąga lepsze wyniki stabilności i dokładności niż `Grid Search` i `Random Search`. Różnice są istotne statystycznie.

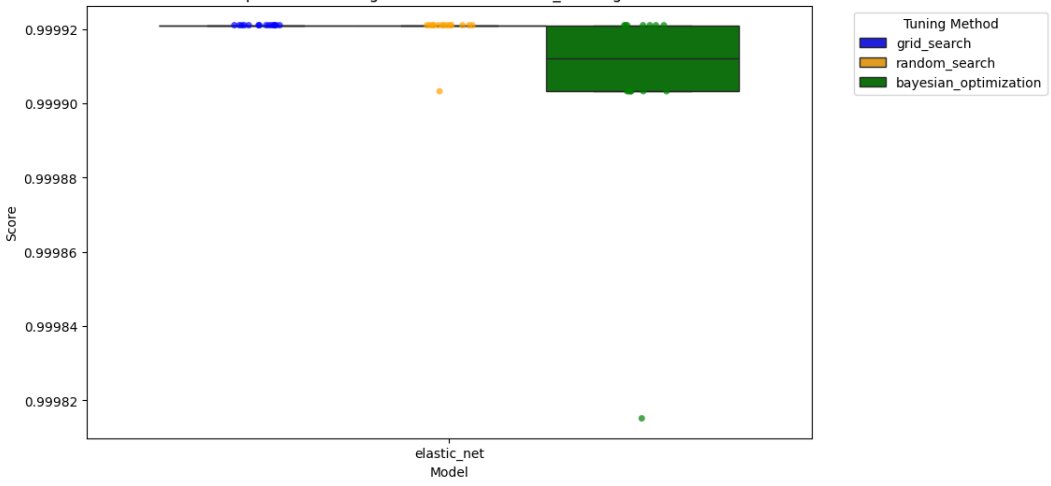


Porównanie wyników metod strojenia dla danych o transfuzjach krwi. `Grid Search` i `Bayesian Optimization` osiągają podobne wyniki, podczas gdy `Random Search` wykazuje większą zmienność. ANOVA wskazuje na znaczące różnice w wydajności między metodami.

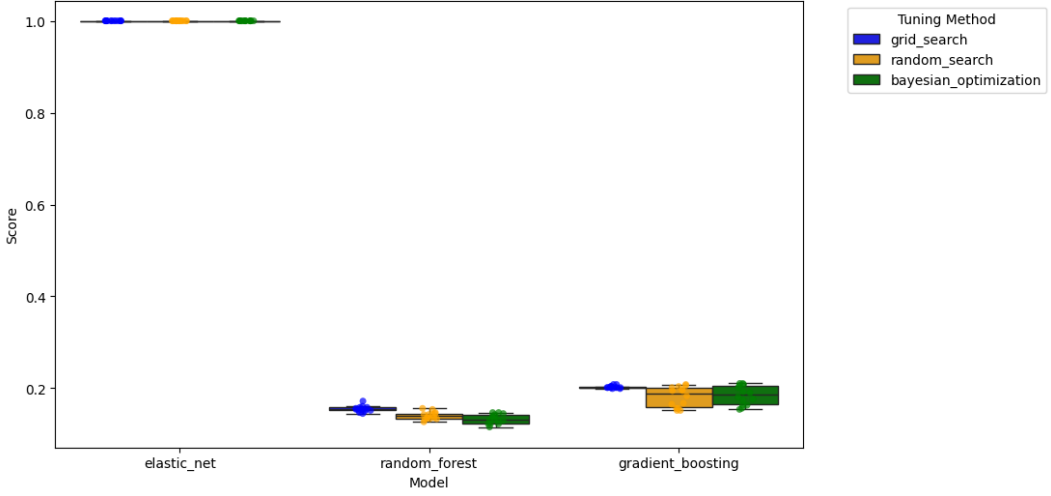


Porównanie wyników metod strojenia dla danych o raku piersi. Wykres pokazuje wyraźną przewagę `Bayesian Optimization` nad pozostałymi metodami pod względem dokładności i stabilności wyników.

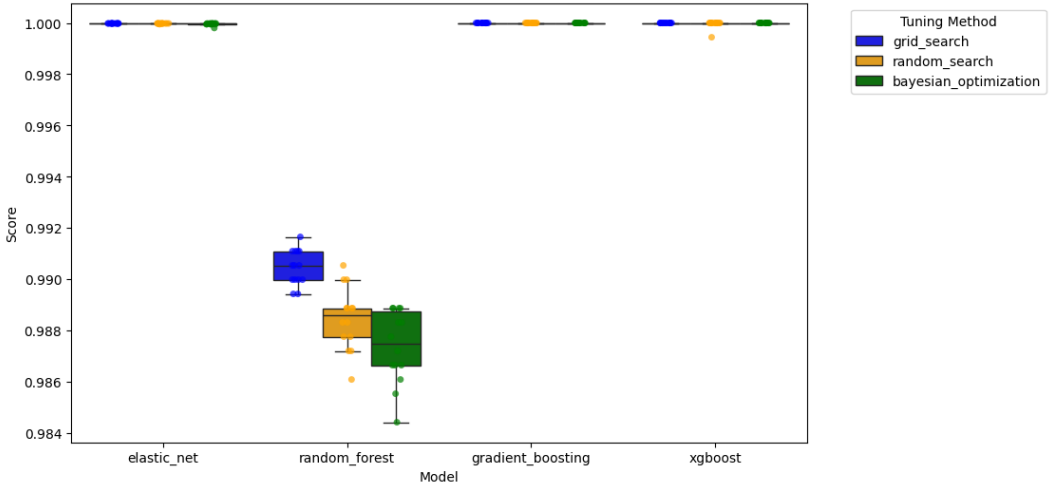
0.63



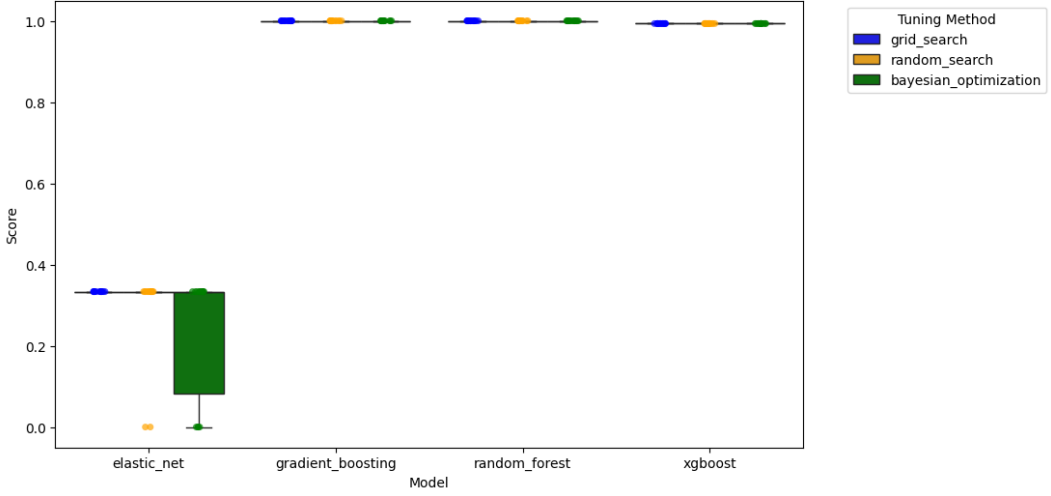
Porównanie wyników metod strojenia dla danych o rynku nieruchomości w Kalifornii. `Bayesian Optimization` osiąga najlepsze wyniki, podczas gdy `Random Search` ma większe odchylenie wyników. Wyniki te są zgodne z testami ANOVA.



Porównanie wyników metod strojenia dla danych o cukrzycy. `Bayesian Optimization` oferuje lepszą wydajność w porównaniu do `Grid Search` i `Random Search`, co potwierdza analiza ANOVA.

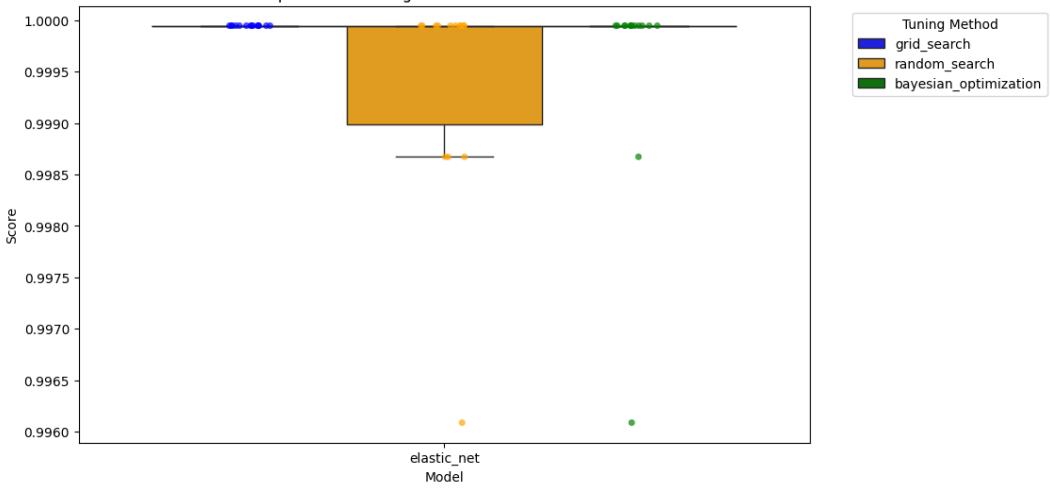


Porównanie wyników metod strojenia dla danych o cyfrach. Wykres wskazuje, że `Grid Search` jest bardziej stabilny, natomiast `Bayesian Optimization` osiąga najwyższe wyniki w większości konfiguracji.



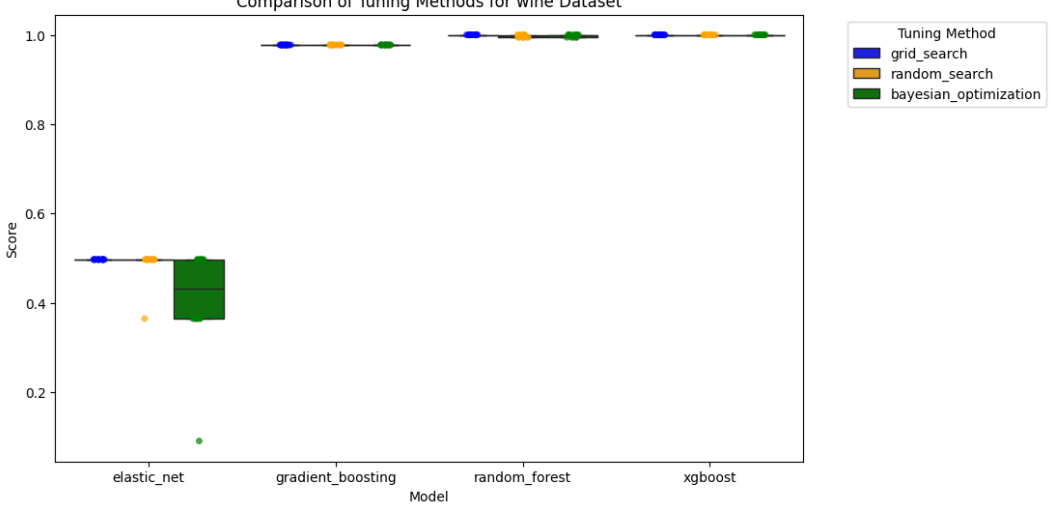
Porównanie wyników metod strojenia dla danych o irysach. Z wykresu wynika, że różnice między `Grid Search`, `Random Search` i `Bayesian Optimization` są niewielkie, ale istotne w sensie statystycznym.

0.78



Porównanie wyników metod strojenia dla danych Linneruda. `Grid Search` i `Bayesian Optimization` oferują podobne wyniki, podczas gdy `Random Search` wykazuje większą zmienność.

0.81



Porównanie wyników metod strojenia dla danych o winach. `Bayesian Optimization` przewyższa inne metody, co jest widoczne w wyższej stabilności wyników.

0.85

0.89

- Optymalizacja Bayesowska:
  - Najlepsza metoda pod względem efektywności i szybkości zbieżności.
- XGBoost i Gradient Boosting:
  - Wysoka tunowalność i poprawa wyników dzięki optymalizacji hiperparametrów.
- Grid Search:
  - Największy koszt obliczeniowy, wymaga wielu iteracji.
- Random Search:
  - Mniejsza dokładność wyników niż Optymalizacja Bayesowska.

- Ile iteracji potrzebujemy dla stabilnych wyników?
  - Optymalizacja Bayesowska wymaga najmniejszej liczby iteracji.
  - Zależy to od modelu, rozmiaru zbioru danych oraz rodzaju zbioru danych.
- Czy technika losowania punktów wpływa na wyniki?
  - Optymalizacja Bayesowska redukuje bias sampling w porównaniu do Random Search.
  - Wpływ zależy od modelu i charakterystyki zbioru danych.
- Jakie hiperparametry są kluczowe?
  - **XGBoost:**
    - `learning_rate`: kluczowy parametr kontrolujący szybkość uczenia modelu.
    - `max_depth`: istotny dla złożoności drzew decydujących.
  - **Random Forest:**
    - `n_estimators`: liczba drzew w modelu, wpływająca na stabilność i dokładność.
    - `max_depth`: ogranicza głębokość drzew, co pomaga w zapobieganiu przeuczeniu.
  - **Elastic Net:**
    - `alpha`: siła regularizacji, kluczowa dla ograniczenia przeuczenia.
    - `l1_ratio`: decyduje o proporcji między karami L1 (Lasso) i L2 (Ridge).
  - **Gradient Boosting:**
    - `learning_rate`: kontroluje wkład każdej iteracji w końcowy model.
    - `n_estimators`: liczba iteracji wzmacniania, wpływająca na dokładność modelu.

Analiza Tunowalności Wybranych Algorytmów Nauczania Maszynowego

Dziękuję za uwagę!
Czy są jakieś pytania?