Tunowalność Hiperparametrów

Gaspar Sekula, Julia Kruk

1 Wprowadzenie

1.1 Cel projektu

Celem projektu było zbadanie tunowalności 3 algorytmów uczenia maszynowego na co najmniej 4 zbiorach danych z wykorzystaniem co najmniej 2 metod samplingu.

1.2 Zbiory danych

1.2.1 Wybór zbiorów

Przy wyborze zbiorów danych kierowaliśmy się następującymi kryteriami: dane musiały dotyczyć zagadnień przyrodniczych, zawierać co najmniej 1000 rekordów, a liczba predyktorów powinna mieścić się w przedziale od 10 do 90. Dodatkowo, zbiory danych musiały być odpowiednie dla problemów klasyfikacji binarnej. Ostatecznie wybraliśmy pięć zbiorów danych z platformy OpenML: mushroom, ozone-level-8hr, kc1, MagicTelescope oraz higgs. Aby usprawnić obliczenia, ograniczyliśmy liczbę rekordów do 5000 dla zbiorów przekraczających ten rozmiar, zachowując przy tym proporcje klas (Wykres 1).

1.2.2 Inżynieria cech

Ewentualne braki danych zastępowaliśmy medianą (w przypadku danych numerycznych) lub najczęściej występującą wartością (w przypadku cech kategorycznych), dane numeryczne skalowaliśmy przy pomocy `SimpleScaler`, a dane kategoryczne kodowaliśmy z użyciem `OneHotEncoder`.

1.3 Algorytmy samplingu i modele

1.3.1 Modele

W eksperymencie badaliśmy trzy modele klasyfikacji binarnej: `XGBoost`, `K-Nearest Neighbors` oraz `Logistic Regression`. `XGBoost` został wybrany w celu przetestowania tych samych hiperparametrów, które zostały użyte w artykule [4], choć zredukowaliśmy ich liczbę z powodu ograniczonych zasobów obliczeniowych. W przypadku `K-Nearest Neighbors` postanowiliśmy sprawdzić większą liczbę hiperparametrów niż we wspomnianym artykule, gdzie autorzy uwzględnili tylko jeden; my rozszerzyliśmy analizę do trzech. LogisticRegression natomiast nie był badany w artykule, co skłoniło nas do jego wyboru, aby poszerzyć zakres naszych badań.

1.3.2 Algorytmy samplingu

Do badania tunowalności użyliśmy metody `Random Search` oraz `Bayes Search`. Badaliśmy również stabilność i zbieżność wyników uzyskanych tymi metodami.

1.4 Zakres hiperparametrów

Hiperparametry do optymalizacji i ich zakresy wybraliśmy na podstawie dokumentacji modeli ([1], [2], [3]), artykułu [4] oraz publikacji [5]. Badane zakresy hiperparametrów znajdują się w Tabeli 1.

Hiperparametr	Rozkład / Wartości
xgb__n_estimators	randint(100, 1000)
xgb__learning_rate	uniform(0.01, 1)
xgb__reg_alpha	loguniform(2^{-10}, 2^{10})
xgb__reg_lambda	loguniform(2^{-10}, 2^{10})
xgb__min_child_weight	randint(1, 100)

Hiperparametr	Rozkład / Wartości
logreg__C	loguniform(2^{-10}, 2^{10})
logreg__max_iter	randint(7000, 8000)
logreg__penalty	{'l1', 'l2'}
logreg__solver	{'liblinear', 'saga'}
logreg__fit_intercept	{True, False}

Hiperparametr	Rozkład / Wartości
knn__n_neighbors	randint(1, 100)
knn__weights	{'uniform', 'distance'}
knn__metric	{'euclidean', 'manhattan', 'minkowski'}

Tabela 1: Zakresy hiperparametrów dla modeli XGBoost, Logistic Regression i K-Nearest Neighbors

2 Wyniki eksperymentu

2.1 Stabilność metody Bayes Search

Wykonaliśmy 60 iteracji metody `Bayes Search` dla każdego modelu na każdym zbiorze danych. Wyniki tych eksperymentów są przedstawione na Rysunku 2. Obserwowane wartości w kolejnych iteracjach wykazują pewne wahania, co jest zrozumiałe, ponieważ funkcja AUC nie jest znana z góry. Metoda bayesowska nie jest w stanie znaleźć dokładnego optimum, a jedynie przybliża jego lokalizację.

2.2 Stabilność metody Random Search

Metodę `Random Search` wykonaliśmy 100 razy dla każdego zbioru i dla każdego modelu. Wykres 3 przedstawia skumulowane maksimum wartości AUC dla danych iteracji. Można zauważyć, że już w pierwszych 20 iteracjach uzyskaliśmy dobre wyniki, a dalsze iteracje rzadko przynosiły znaczące poprawy.
Na Rysunku 4 przedstawiono rozkład wyników AUC uzyskanych korzystając z metody Random Search na każdym ze zbiorów danych i na każdym modelu.

2.3 Tunowalność algorytmów

Tunowalność algorytmów dla każdego ze zbiorów została wyznaczona z poniższego wzoru (pochodzącego z artykułu [4]):

$$d^{(j)} := R^{(j)}(\theta^\star) - R^{(j)}(\theta^{(j)\star}),$$

gdzie j jest numerem zbioru, $R^{(j)}(\theta) = -\text{AUC}$, a $\theta^\star$ jest domyślnym zestawem hiperparametrów.
Otrzymane wyniki zostały przedstawione w Tabeli 2.

Zbiór danych \ Model	XGBoost	KNN	LogReg
mushroom	0.0000	-0.0001	-0.0020
ozone	0.0596	0.0174	0.0099
kc1	0.1401	0.0177	0.0080
MagicTelescope	0.0395	0.0003	0.0013
higgs	0.2942	0.0960	0.0091

Tabela 2: Tunowalność algorytmów

2.4 Optymalne hiperparametry

W celu wyznaczenia optymalnego zestawu hiperparametrów dla danego modelu wykorzystaliśmy wzór (pochodzący z artykułu [4]):

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

a za funkcję g przyjęliśmy średnią. Obliczone optymalne parametry przedstawione są w Tabeli 3.

Model	Hiperparametr	Optymalna wartość	Wartość domyślna	$\mathbf{AUC_t}$	$\mathbf{AUC_d}$
XGBoost	min_child_weight	15.0	1	0.8646	0.8563
	reg_alpha	8.85	0		
	reg_lambda	18.72	1		
	learning_rate	0.49	0.3		
	n_estimators	480.0	100		
KNN	n_neighbors	27.0	5	0.8468	0.8296
	weights	distance	uniform		
	metric	manhattan	minkowski		
LogReg	C	1.16	1.0	0.8362	0.8350
	max_iter	7535.0	100		
	penalty	l1	l2		
	solver	liblinear	lbfgs		

Tabela 3: Porównanie optymalnych wartości parametrów z wartościami domyślnymi. AUC_t oraz AUC_d to, odpowiednio, średni wynik AUC dla danej θ i wynik AUC uzyskany z domyślnymi hiperparametrami.

Jak można zauważyć, żadna z wartości się nie pokrywa. Jednak pomimo robieżności w wartościach hiperparametrów, różnice w wynikach nie są znaczące. Ponadto, należy wziąć pod uwagę, że nasze badanie odbywa się na niewielkiej liczbie zbiorów danych.

2.5 Testy statystyczne

Przeprowadziliśmy trzy testy: weryfikujący czy wyniki AUC po optymalizacji bayesowskiej lub `Random Search` się istotnie różnią oraz czy metoda `Random Search` zbiega istotnie szybciej niż `Bayes Search`.

2.5.1 Porównanie wyników

Niech próbką $X = X_1, ..., X_5$ będą najlepsze wyniki AUC na poszczególnych zbiorach danych uzyskanych jedną z metod. Natomiast próbką $Y = Y_1, ..., Y_5$ - uzyskanych inną metodą. Spośród metod mamy do dyspozycji: tuning metodą Bayesa, tuning metodą `Random Search` i brak metody (model z domyślnymi wartościami parametrów). Ponieważ próbka jest bardzo mała, z czym źle sobie radzi test Wilcoxona, zastosowaliśmy następującą strategię testowania:

- Przeprowadzamy test Shapiro Wilka w celu zweryfikowania normalności rozkładu $Z := X - Y$.
- Jeżeli tak jest, to przeprowadzamy t-test, w przeciwnym przypadku - test Wilcoxona.

W tym teście sprawdzamy hipotezę:

H_0: Wyniki uzyskane metodą 1. nie różnią się istotnie od wyników uzyskanych metodą 2. względem alternatywy.
H_1: Jest istotna różnica pomiędzy wynikami.

Wszystkie testy przeprowadziliśmy na poziomie istotności 0.05. W 8 z 9 testów (3 modele i 3 pary metod) nie mieliśmy podstaw do odrzucenia hipotezy H_0. Wyjątek stanowił model `Logistic Regression` z metodą `Random Search` i default. Możemy zatem stwierdzić, że w większości przypadków, wyniki uzyskane przy użyciu dobranych

parametrów nie różnią się istotnie od tych bez tuningu hiperparametrów. Ponadto wyniki dla `Bayes Search` i `Random Search` również nie różnią się istotnie. Szczegółowe wyniki znajdują się w Tabeli 4 .

2.5.2 Porównanie szybkości zbieżności

Chcemy porównać szybkość zbieżności wyników metodami `Bayes Search` oraz `Random Search`. Niech $X = X_1, ..., X_{60}$ będzie dotychczasowym najlepszym wynikiem AUC (cummulative max) dla ustalonego zbioru danych przy optymalizacji `Random Search`, a $Y = Y_1, ..., Y_{60}$ - dla tego samego zbioru i modelu co X, lecz przy `Bayes Search`. Weryfikujemy hipotezy jak w poprzednim akapicie, również na poziomie istotności 0.05. We wszystkich przypadkach wyniki się istotnie różniły. Rezultat badania znajduje się w Tabeli 5.

Ponieważ wiemy, że szybkość zbueżności jest istotnie różna, pojawia się chęć odpowiedzi na pytanie: "Która metoda optymalizacji zbiega szybciej?". W tym celu przeprowadzamy trzeci test, w którym X i Y są jak dotychczas, lecz zmianie ulegają hipotezy:

H_0: $\text{med}(X - Y) \leqslant 0$ (`Random Search` zbiega wolniej lub tak samo jak `Bayes Search`) wobec alternatywy
H_1: $\text{med}(X - Y) > 0$ (`Random Search` zbiega szybciej niż `Bayes Search`)

Wyniki badania znajdują się w Tabeli 6. Owoce naszego statystycznego przedsięwzięcia pozwalają stwierdzić, że w zbiorach danych kc1 oraz ozone metoda `Random Search` zbiega istotnie szybciej, a w przypadku pozostałych - `Bayes Search`. Na tak małej próbce nie jesteśmy w stanie określić ogólnych uwarrunkowań, kiedy optymalizacja bayesowska powinna być tą preferowaną.

2.6 Bias sampling

Na podstawie badania rozkładu rożnic wyników AUC uzyskanych przy `Random Search` i tych dla domyślnych wartości hiperparametrów (Wykres 5), wyciągamy następujące wnioski:

- Dla wszystkich modeli dla zbioru Mushroom stwierdzamy brak bias samplingu.
- Brak bias samplingu możemy ponadto stwierdzić dla: KNN w zbiorach Ozone i Kc1.
- W pozostałych przypadkach, ze względu na skośność (asymetrię) rozkładu, stwierdzamy bias sampling.

Podobnie dla wyników z `Bayes Search` (Wykres 6):

- Brak bias samplingu możemy stwierdzić dla: XGBoost na zbiorach Mushroom, Logistic Regression na wszytskich zbiorach poza Mushroom oraz dla KNN na zbiorach Mushroom i MagicTelescope.
- W pozostałych przypadkach, ze względu na skośność (asymetrię) rozkładu, stwierdzamy bias sampling.

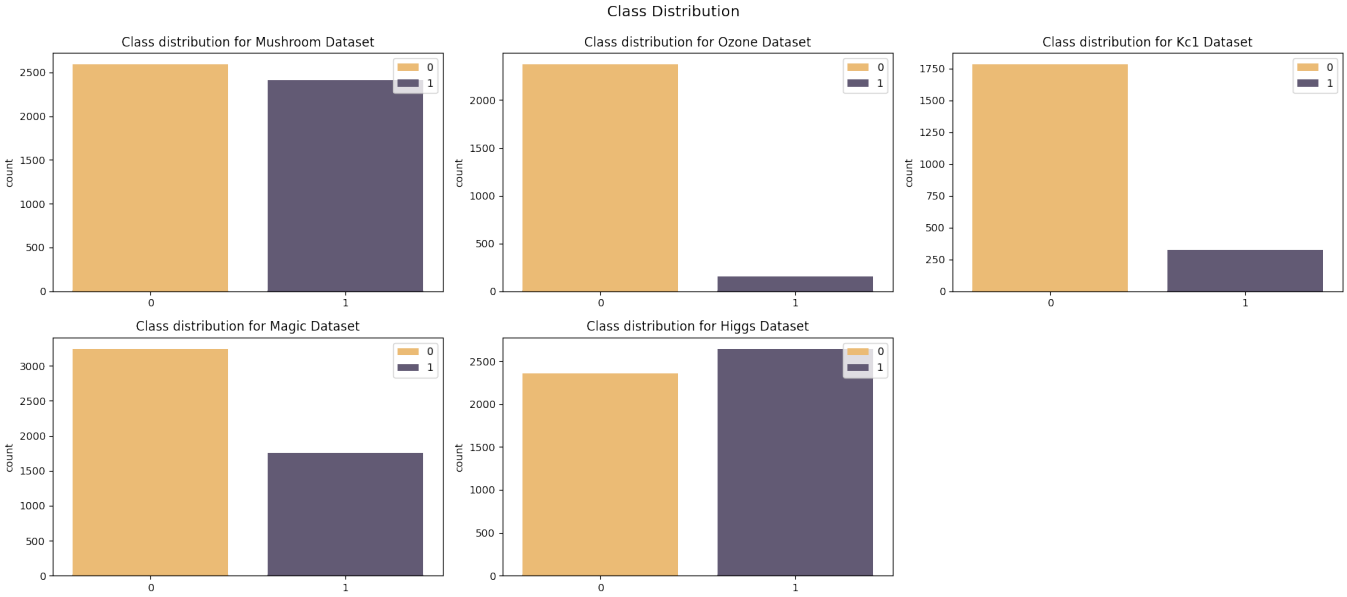
Wspólne dla obu wykresów są następujące wnioski:

- Wyniki różnią się w zależności od zbioru danych i modelu.
- Dla większości zbiorów danych najmniejszą skośnością charakteryzują się rozkłady dla modelu Logistic Regression.

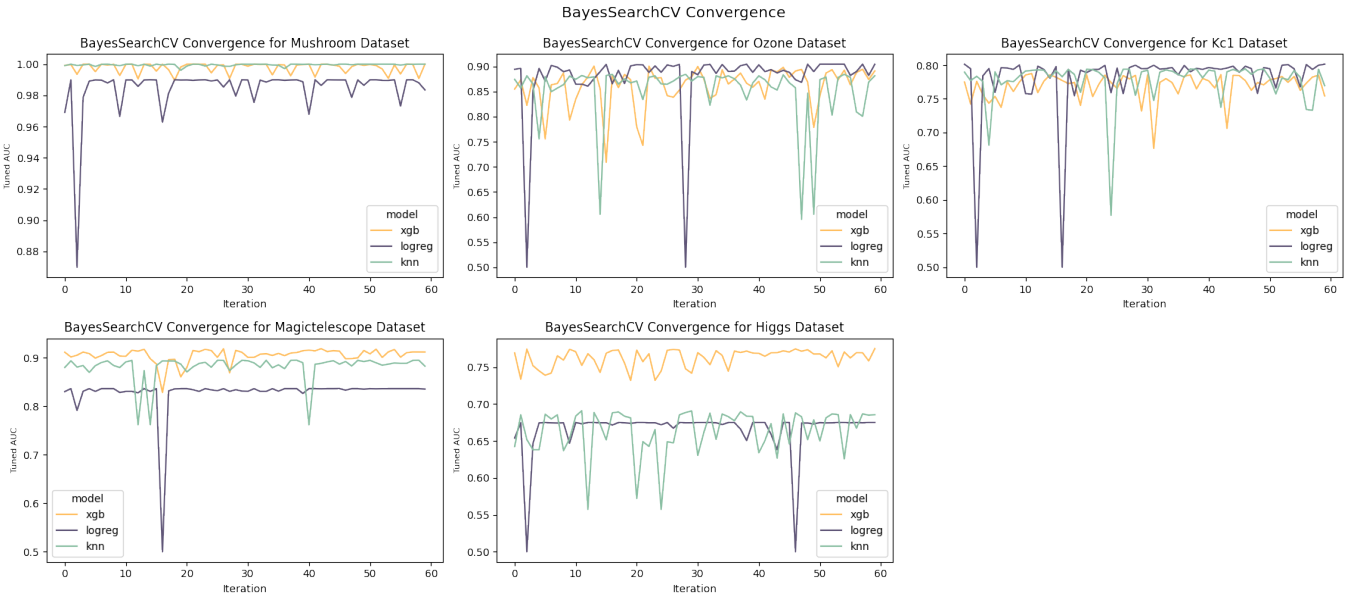
3 Wnioski z eksperymentu

Jak wynika z danych przedstawionych w Tabeli 2 i Tabeli 3, modyfikowanie zestawów hiperparametrów (w przypadku tych danych i tych modeli) nie wpływa znacząco na poprawę wyników. Najwiekszą tunowalnością wykazał się `XGBoost`, jednak wyniosła ona maksymalnie 0.2942. Potencjalną przyczyną małej tunowalności może być odpowiednie przygotowanie zbiorów danych, które pochodzą z platformy OpenML.
Testy statystyczne wykazały, że nie ma istotnych różnic między wynikami uzyskanymi przy użyciu `Random Search`, `Bayes Search` czy domyślnych hiperparametrów. Taki rezultat może mieć związek z charakterystyką zbiorów danych, na których przeprowadzono badanie. Co więcej, zbieżność metod `Random Search` i `Bayes Search` na podstawie naszego eksperymentu różni się istotnie, a to, który model szybcie uzyskuje optymalne wyniki - zależy od zbioru danych. W związku z tym nie można sformułować ogólnego twierdzenia, które jednoznacznie determinowałoby, która z badanych metod jest najlepsza.

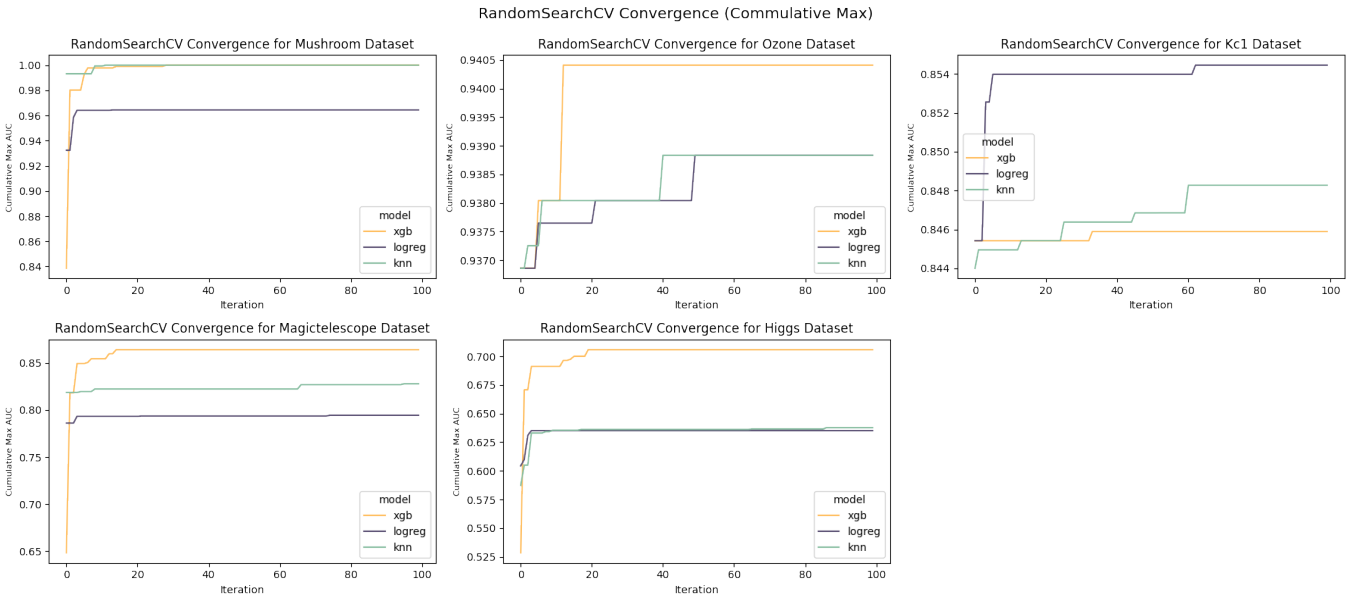
4 Wizualizacje i wyniki



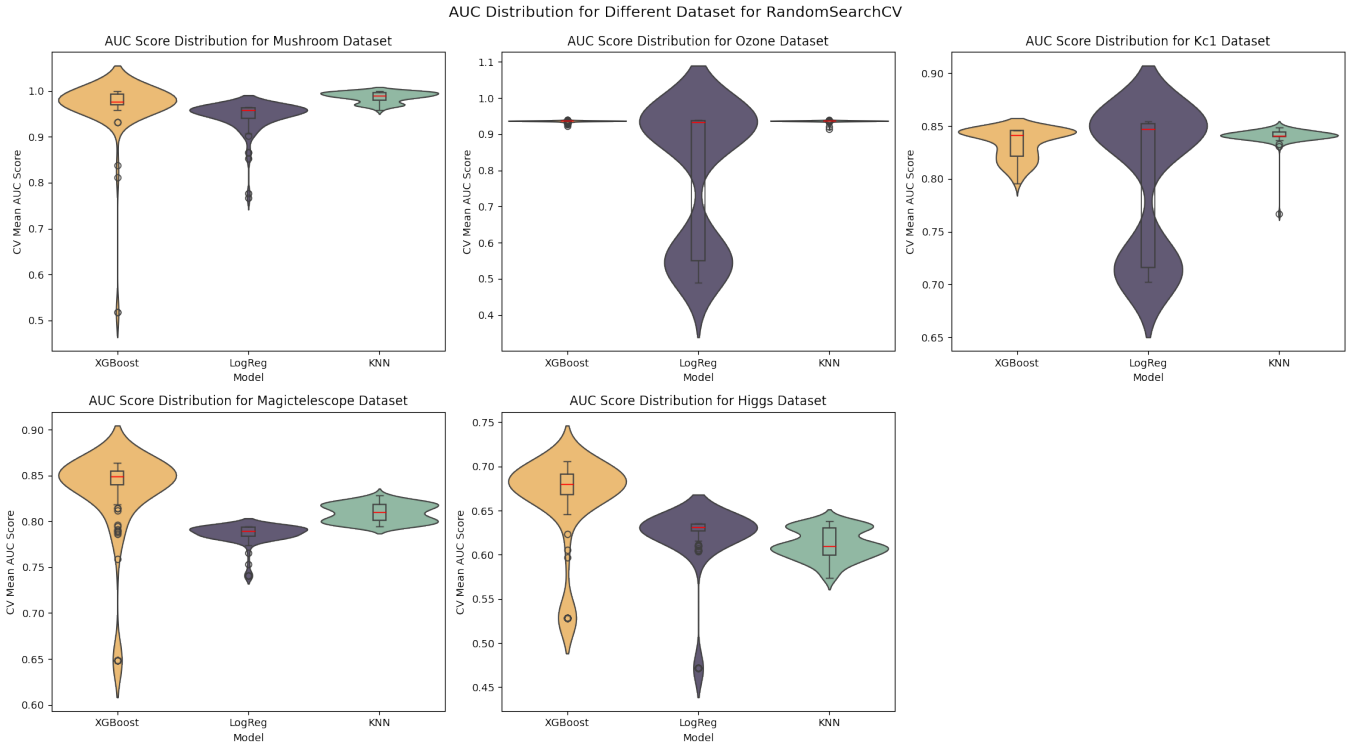
Rysunek 1: Rozkład klas



Rysunek 2: Zbieżność metody Bayes Search



Rysunek 3: Skumulowane maksimum wartości AUC dla metody Random Search



Rysunek 4: Rozkład AUC dla poszczególnych zbiorów danych dla metody Random Search

Model	Metoda 1	Metoda 2	Test	p-wartość	Hipoteza zerowa
xgb	random	bayes	t-test	0.845555	brak podstaw do odrzucenia
xgb	random	default	t-test	0.107512	brak podstaw do odrzucenia
xgb	bayes	default	t-test	0.413406	brak podstaw do odrzucenia
logreg	random	bayes	t-test	0.849031	brak podstaw do odrzucenia
logreg	random	default	t-test	0.023207	odrzucona
logreg	bayes	default	t-test	0.752208	brak podstaw do odrzucenia
knn	random	bayes	t-test	0.930449	brak podstaw do odrzucenia
knn	random	default	t-test	0.099877	brak podstaw do odrzucenia
knn	bayes	default	t-test	0.463025	brak podstaw do odrzucenia

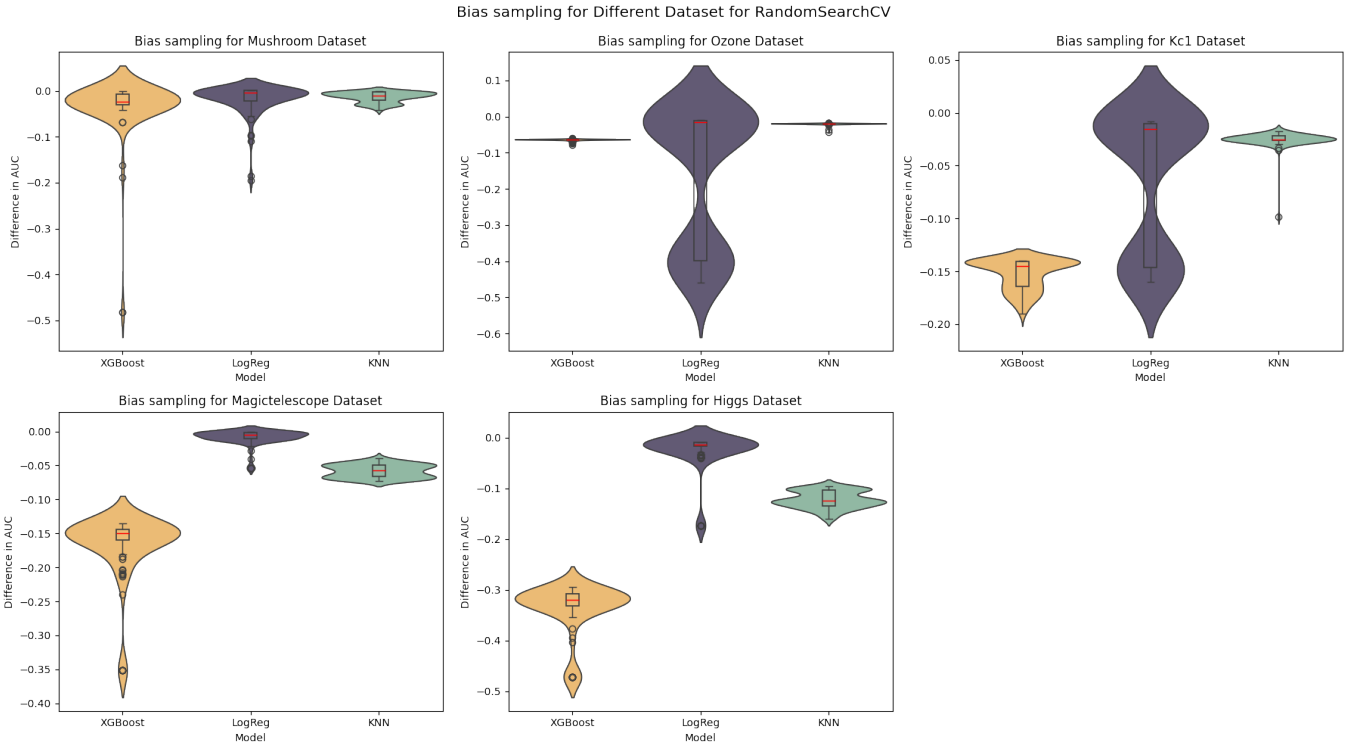
Tabela 4: Porównanie wyników przy użyciu testów statystycznych.

Model	Zbiór danych	Test	p-wartość	Hipoteza zerowa
xgb	higgs	wilcoxon test	8.879560e-12	odrzucona
xgb	kc1	wilcoxon test	7.076742e-12	odrzucona
xgb	magic	wilcoxon test	1.195110e-11	odrzucona
xgb	mushroom	wilcoxon test	2.530149e-06	odrzucona
xgb	ozone	wilcoxon test	7.534773e-13	odrzucona
logreg	higgs	wilcoxon test	5.987643e-13	odrzucona
logreg	kc1	wilcoxon test	1.099584e-13	odrzucona
logreg	magic	wilcoxon test	9.592416e-12	odrzucona
logreg	mushroom	wilcoxon test	9.301841e-13	odrzucona
logreg	ozone	wilcoxon test	1.435184e-11	odrzucona
knn	higgs	wilcoxon test	1.652376e-12	odrzucona
knn	kc1	wilcoxon test	1.493619e-11	odrzucona
knn	magic	wilcoxon test	1.120685e-11	odrzucona
knn	mushroom	wilcoxon test	3.116095e-03	odrzucona
knn	ozone	wilcoxon test	9.012469e-12	odrzucona

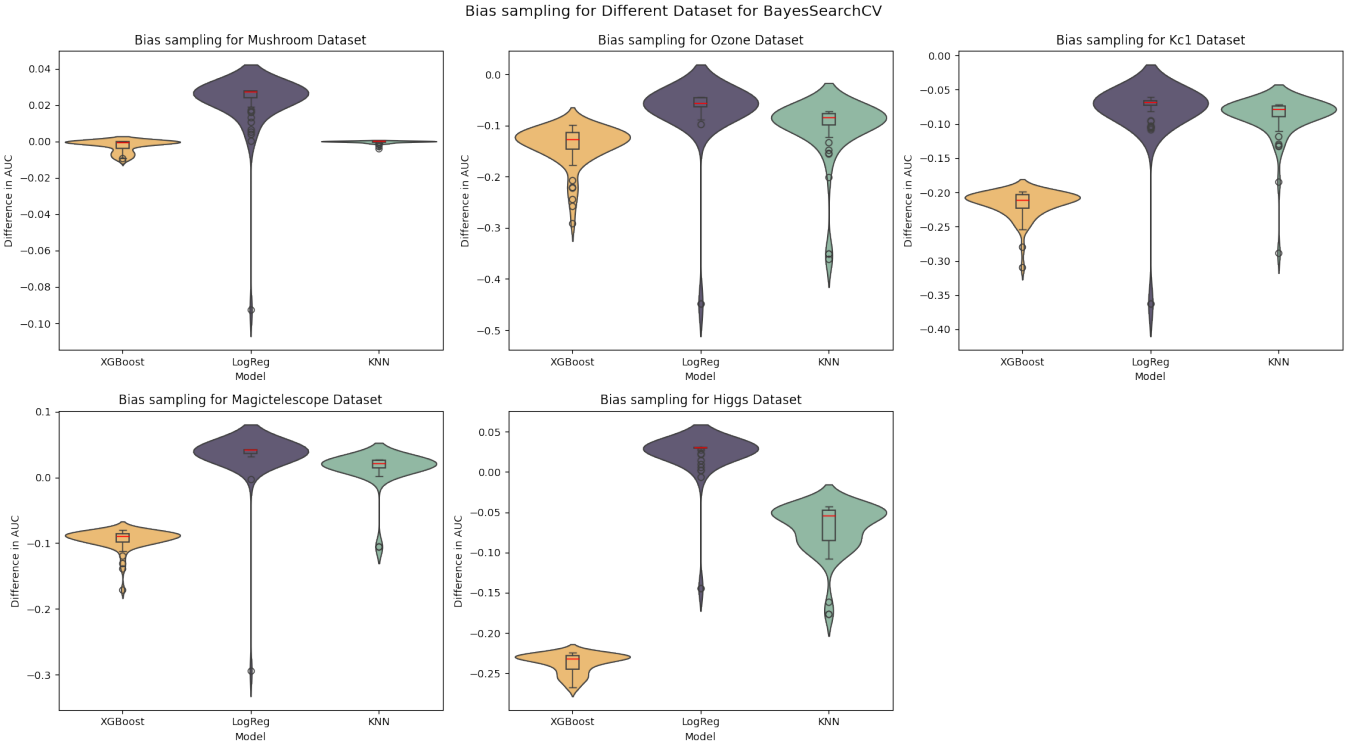
Tabela 5: Porównanie zbieżności przy uzyciu testów statystycznych.

Model	Zbiór danych	p-wartość	Hipoteza zerowa	Szybciej zbiega
xgb	higgs	1.000000e+00	brak podstaw do odrzucenia	Bayes Search
xgb	kc1	3.538371e-12	odrzucona	Random Search
xgb	magic	1.000000e+00	brak podstaw do odrzucenia	Bayes Search
xgb	mushroom	9.999987e-01	brak podstaw do odrzucenia	Bayes Search
xgb	ozone	3.767386e-13	odrzucona	Random Search
logreg	higgs	1.000000e+00	brak podstaw do odrzucenia	Bayes Search
logreg	kc1	5.497918e-14	odrzucona	Random Search
logreg	magic	1.000000e+00	brak podstaw do odrzucenia	Bayes Search
logreg	mushroom	1.000000e+00	brak podstaw do odrzucenia	Bayes Search
logreg	ozone	7.175920e-12	odrzucona	Random Search
knn	higgs	1.000000e+00	brak podstaw do odrzucenia	Bayes Search
knn	kc1	7.468095e-12	odrzucona	Random Search
knn	magic	1.000000e+00	brak podstaw do odrzucenia	Bayes Search
knn	mushroom	9.984420e-01	brak podstaw do odrzucenia	Bayes Search
knn	ozone	4.506235e-12	odrzucona	Random Search

Tabela 6: Ustalenie, która metoda szybciej zbiega przy użyciu testów statystycznych.



Rysunek 5: Rozkład różnicy AUC w iteracjach metodą Radnom Search i AUC dla domyślnych prarametrów.



Rysunek 6: Rozkład różnicy AUC w iteracjach metodą Bayes Search i AUC dla domyślnych prarametrów.

Literatura

[1] K nearest neighbours classifier official documentation.

[2] Logistic regression official documentation.

[3] Xgboost official documentation.

[4] Bernd Bischl, Anne-Laure Boulesteix, and Philipp Probst. Tunability: Importance of hyperparameters of machine learning algorithms. *Journal of Machine Learning Research*, 2019.

[5] Sateesh Ambesange; A. Vijayalaxmi; S. Sridevi; Venkateswaran; B. S. Yashoda. Multiple heart diseases prediction using logistic regression with ensemble and hyper parameter tuning techniques. *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*.