



**Faculty of Mathematics
and Information Science**

WARSAW UNIVERSITY OF TECHNOLOGY

Automatyczne uczenie maszynowe

Tunowalność algorytmów ML

Szymon Pawlonka

Łucja Żukowska

WARSZAWA, 2023



1 Wstęp

Celem zadania była optymalizacja 3 wybranych algorytmów na 4 różnych zbiorach danych dla ustalonej siatki hiperparametrów w celu obliczenia tunowalności. Do optymalizacji wykorzystano dwie różne techniki losowania punktów - Random Search oraz optymalizację bayesowską. Dla każdej z nich ustalono stałą liczbę 500 iteracji.

2 Wybrane zbiory danych

Skupiono się na zbiorach danych dotyczących klasyfikacji binarnej zawierających około 5000 rekordów ze zróżnicowaną liczbą kolumn. Wybrane zbiory to abalone, spambase, ada_prior oraz phoneme. Wszystkie zostały zaczerpnięte z bibliotek OpenML. Więcej informacji na ich temat można znaleźć poprzez odnośniki zamieszczone w dodatku A.

3 Wybrane algorytmy oraz zakresy hiperparametrów

W ramach projektu wybrano następujące algorytmy: las losowy (RandomForestClassifier), wzmocnienie gradientowe (GradientBoostingClassifier) i sieci neuronowe (MLPClassifier). Przy implementacji korzystano z biblioteki sklearn.

Zakresy hiperparametrów przedstawiono w tabeli [1]. Wartości hiperparametrów dla lasu losowego oraz wzmocnienie gradientowe zaczerpnięto z artykułu^[1], a siatkę hiperparametrów dla sieci neuronowych z artykułu^[2].

4 Obliczenia tunowalności

W celu obliczenia tunowalności algorytmów skorzystano z definicji zamieszczonej w artykule^[1]. Tunowalnością algorytmu dla zbioru danych j określamy jako

$$d^{(j)} := R^{(j)}(\theta^*) - R^{(j)}(\theta^{(j)*}), \text{ dla } j = 1, \dots, m, \quad (1)$$

gdzie $R^{(j)}(\theta)$ to dokładność algorytmu dla danej konfiguracji hiperparametrów θ dla zbioru danych j . Ponadto θ^* to taka konfiguracja hiperparametrów, której średni wynik dla wybranych zbiorów danych był najwyższy, zaś $\theta^{(j)*}$ to taka konfiguracja, która os-



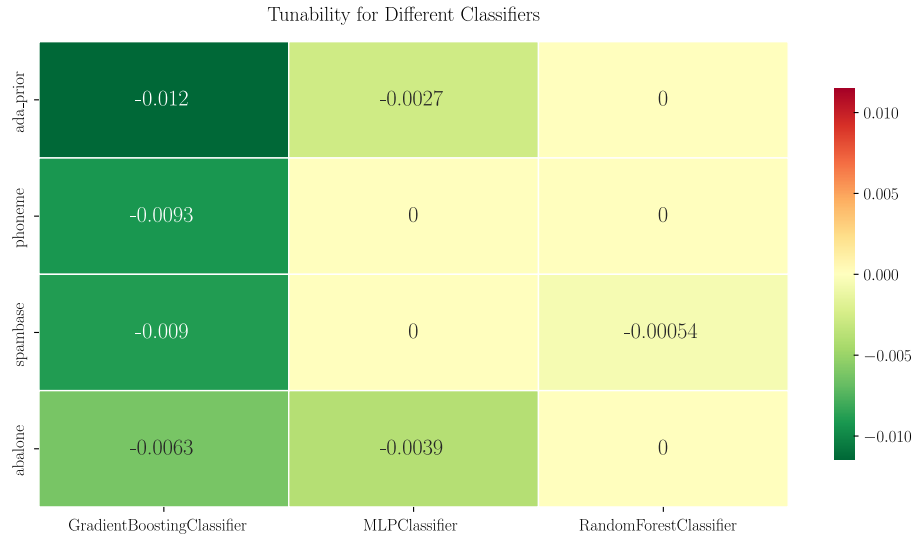
Algorytm	Hiperparametr	Typ	Od	Do	Funkcja
RandomForestClassifier	n_estimators	integer	1	2000	-
	max_depth	integer	1	20	-
	max_samples	numeric	0.1	1	-
	max_features	discrete	-	-	-
	min_samples_split	numeric	0.1	0.5	-
MLPClassifier	hidden_layer_sizes	integer	10	500	-
	activation	discrete	-	-	-
	learning_rate_init	numeric	-6	1	2^x
	alpha	numeric	-6	1	2^x
GradientBoostingClassifier	learning_rate	numeric	-10	0	2^x
	subsample	numeric	0.1	1	-
	loss	discrete	-	-	-
	max_depth	integer	1	15	-
	min_samples_split	integer	0	7	2^x
	max_features	numeric	0	1	-

Tabela 1: Przedziały hiperparametrów dla wybranych algorytmów. Kolumna Funkcja opisuje, jakim transformacjom uległy wskazane przedziały. Wartości dyskretne dla max_features to 'sqrt', 'log2', dla activation to 'relu', 'identity', 'logistic', 'tanh', a dla loss: 'log_loss' i 'exponential'.

iągnęła najwyższy wynik dla zbioru j . Obliczone wartości tunowalności Random Search dla wybranych zbiorów danych prezentuje heatmapa 1.

Jak można zauważyć, w 5 na 12 przypadków średnia najlepsza konfiguracja hiperparametrów okazała się również tą najlepszą. Widać również, iż dla algorytmu Gradient Boosting otrzymano najniższe wartości tunowalności. Oznacza to, że spośród badanych algorytmów jego wynik można najbardziej polepszyć, patrząc wyłącznie na konkretny zbiór danych przy optymalizacji jego hiperparametrów.

Analogicznie, wyniki dla metody losowania punktów Bayes Search zebrano na heatmapie 2. Okazuje się, że tylko w jednym przypadku, dla zbioru phoneme wytrenowanego przy użyciu algorytmu RandomForest, najlepszy uzyskany wynik nie był lepszy od wyniku uzyskanego dla wyznaczonej przy użyciu Random Search konfiguracji hiperparametrów θ^* . Porównując ze sobą wyniki obu heatmap można również wyciągnąć wniosek, iż poza wspomnianym przypadkiem, wszystkie pozostałe pary (algorytm, zbiór danych) osiągnęły lepszy maksymalny wynik dla optymalizacji bayesowskiej. Można zauważyć, że technika losowania punktów może istotnie wpłynąć na samą wartość tunowalności, m.in. nie znaj-



Rysunek 1: Heatmapa tunowalności algorytmów dla wybranych zbiorów danych przy technice losowania punktów Random Search. Zielony kolor oznacza, że najlepszy wynik dla danego zbioru danych był lepszy od wyniku średniej konfiguracji hiperparametrów θ^* . Czerwony oznacza analogicznie wynik gorszy.

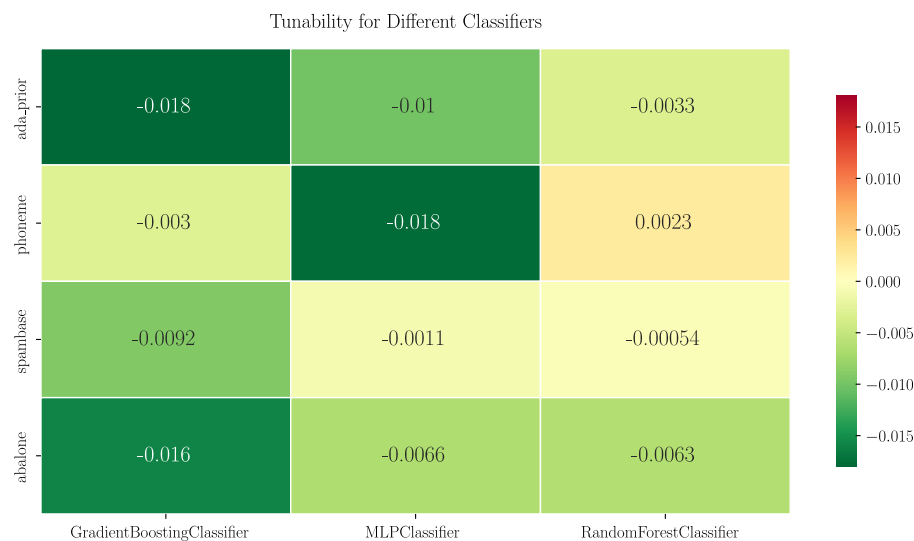
dując najlepszych możliwych konfiguracji dla danego zbioru rozwiązań.

Zbieżność optymalizacji bayesowskiej jest ściśle powiązana ze zbiorem danych. Nie udało się jasno określić liczby, która byłaby odpowiednia dla wszystkich wybranych zbiorów. Przykładowo dla zbioru abalone dla algorytmu Random Forest wystarczyło około 150 iteracji, przy czym dla tego samego algorytmu dla zbioru phoneme zmiany następowały aż do ostatniej, czyli 500. iteracji. Analogicznej obserwacji można dokonać przy metodzie Random Search. Przykładowo dla zbioru danych ada_prior przy uczeniu algorytmem GradientBoosting wystarczyło pierwszych kilka iteracji, aby znaleźć najlepsze rozwiązanie, podczas gdy dla RandomForest najlepsze rozwiązanie zostało znalezione dopiero w okolicach 450. iteracji. Po więcej szczegółów odsyłamy do dodatków C i D.

Określona została również tunowalność algorytmów dla każdej wykonanej iteracji. Można ją zapisać jako

$$d_i^{(j)} := R^{(j)}(\theta^*) - R^{(j)}(\theta_i), \text{ dla } j = 1, \dots, m, i = 1, \dots, n, \quad (2)$$

gdzie $R^{(j)}(\theta)$ to dokładność algorytmu dla danej konfiguracji hiperparametrów θ dla zbioru danych j , a i to numer iteracji.



Rysunek 2: Heatmapa tunowalności algorytmów dla wybranych zbiorów danych przy technice losowania punktów Bayes Search. Zielony kolor oznacza, że najlepszy wynik dla danego zbioru danych był lepszy od wyniku średniej konfiguracji hiperparametrów θ^* . Czerwony oznacza analogicznie wynik gorszy.

5 Wnioski

Na podstawie przeprowadzonych badań, zauważono, że średnio najlepsza konfiguracja dla wybranych zbiorów danych nie jest najlepszą konfiguracją dla tego zbioru. Widać to w sposób szczególny, porównując jej wynik do najlepszych wyników otrzymanych poprzez optymalizację bayesowską. Oznacza to, że wybranie uniwersalnego zbioru hiperparametrów dla każdego zbioru danych jest niełatwym zadaniem, pomimo tego, że w tym przypadku rozpatrywana liczba zbiorów była mała. Jednakże, niejednokrotnie jest to wystarczające przybliżenie, bardzo bliskie optimum dla konkretnego zbioru danych.

Ponadto, pośród badanych algorytmów najniższe wartości tunowalności odnotowano dla algorytmu Gradient Boosting, co świadczy o tym, że w jego przypadku można uzyskać największą poprawę wyniku, optymalizując hiperparametry ściśle pod uczone zbiory danych.

Warto zaznaczyć, że wynik tunowalności jest zależny od techniki losowania punktów. Może ona wpływać na znalezienie różnych najlepszych i średnio najlepszych konfiguracji. W przypadku Random Search wynik średnio najlepszy był często najlepszy, a możliwe polepszenie w pozostałych przypadkach było minimalne. Sytuacja prezentuje się inaczej w optymalizacji bayesowskiej, gdzie istniała większa możliwość poprawy hiperparametrów.



Bibliografia

- [1] Philipp Probst, Bernd Bischl, and Anne-Laure Boulesteix. Tunability: Importance of hyperparameters of machine learning algorithms, 2018.
- [2] Jozsef Suto. The effect of hyperparameter search on artificial neural network in human activity recognition. *Open Computer Science*, 11(1):411–422, 2021.

Dodatek

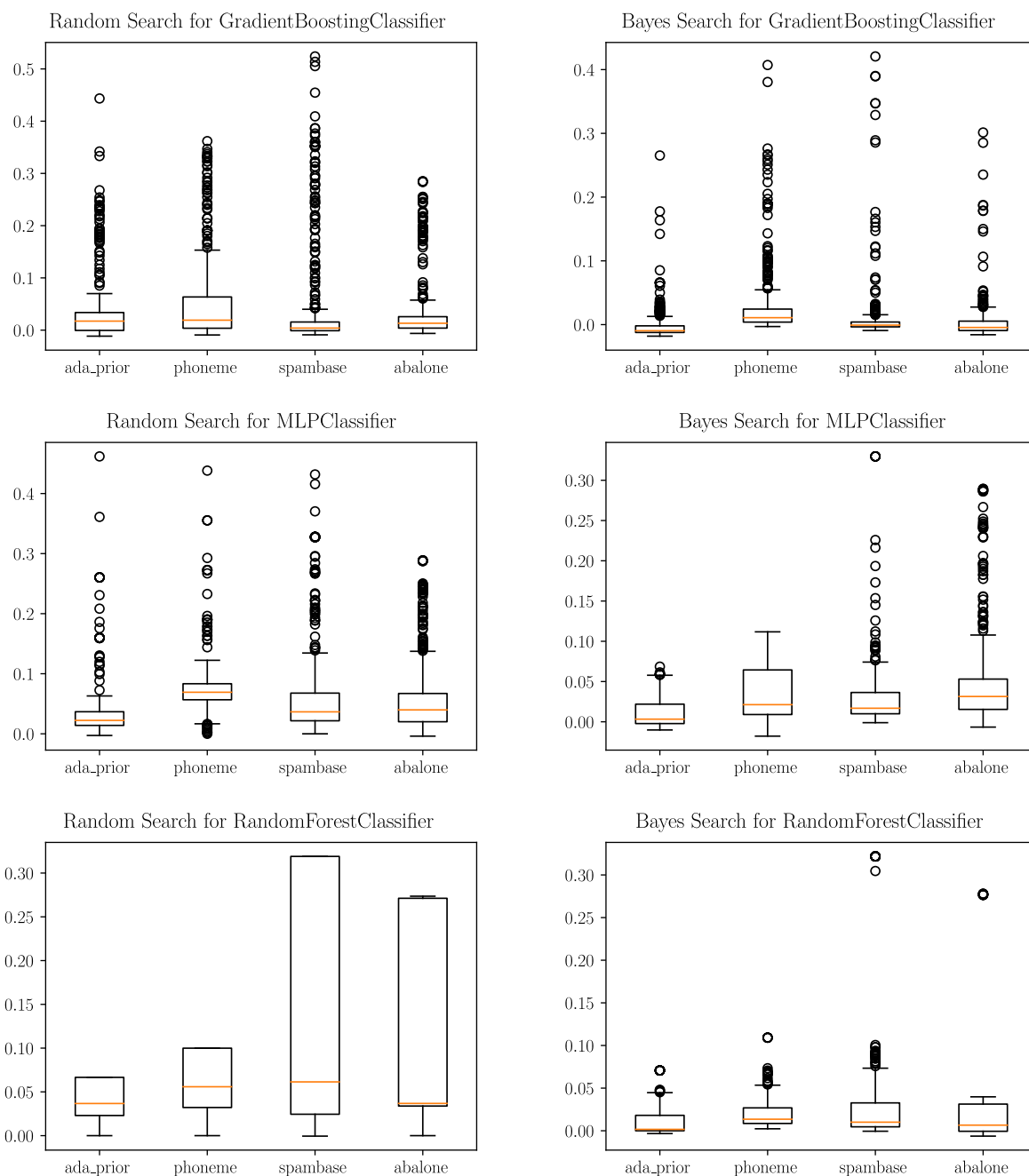
A Zbiory danych

Odnosińniki do wybranych zbiorów danych:

- abalone - 720 - <https://www.openml.org/search?type=data&id=720>
- ada_prior - 1037 - <https://www.openml.org/search?type=data&id=1037>
- phoneme - 1489 - <https://www.openml.org/search?type=data&id=1489>
- spambase - 44 - <https://www.openml.org/search?type=data&id=44>

B Wykresy tunowalności

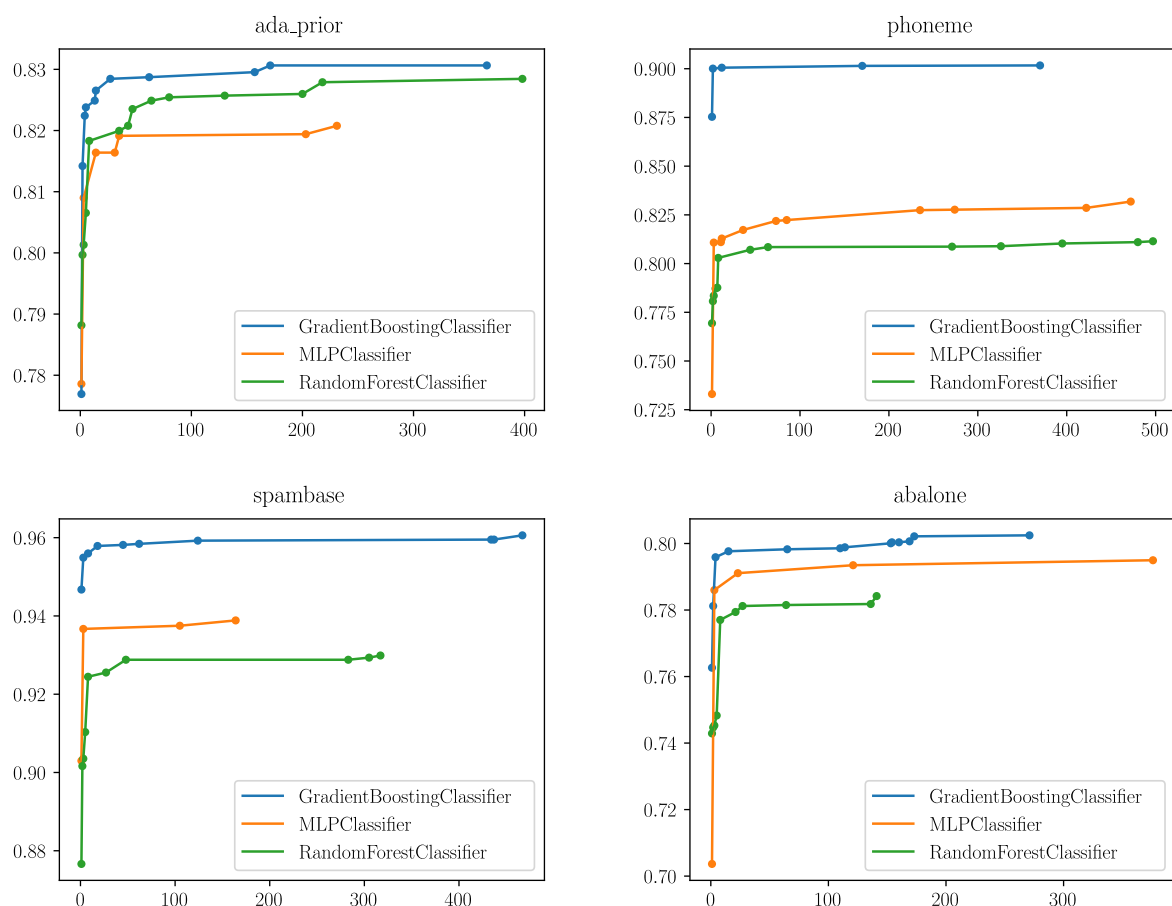
Boxploty 3 przedstawiają tunowalność poszczególnych algorytmów dla każdego zbioru według definicji (2) dla obu wybranych metod losowania punktów.



Rysunek 3: Wykresy tunowalności. W lewej kolumnie umiejscowione zostały wyniki dla metody Random Search, zaś w prawej dla optymalizacji bayesowskiej. Na osi Y zaprezentowano wynik tunowalności, a na osi X rozmieszczono kolejne zbiory danych.

C Wykresy zbieżności optymalizacji bayesowskiej

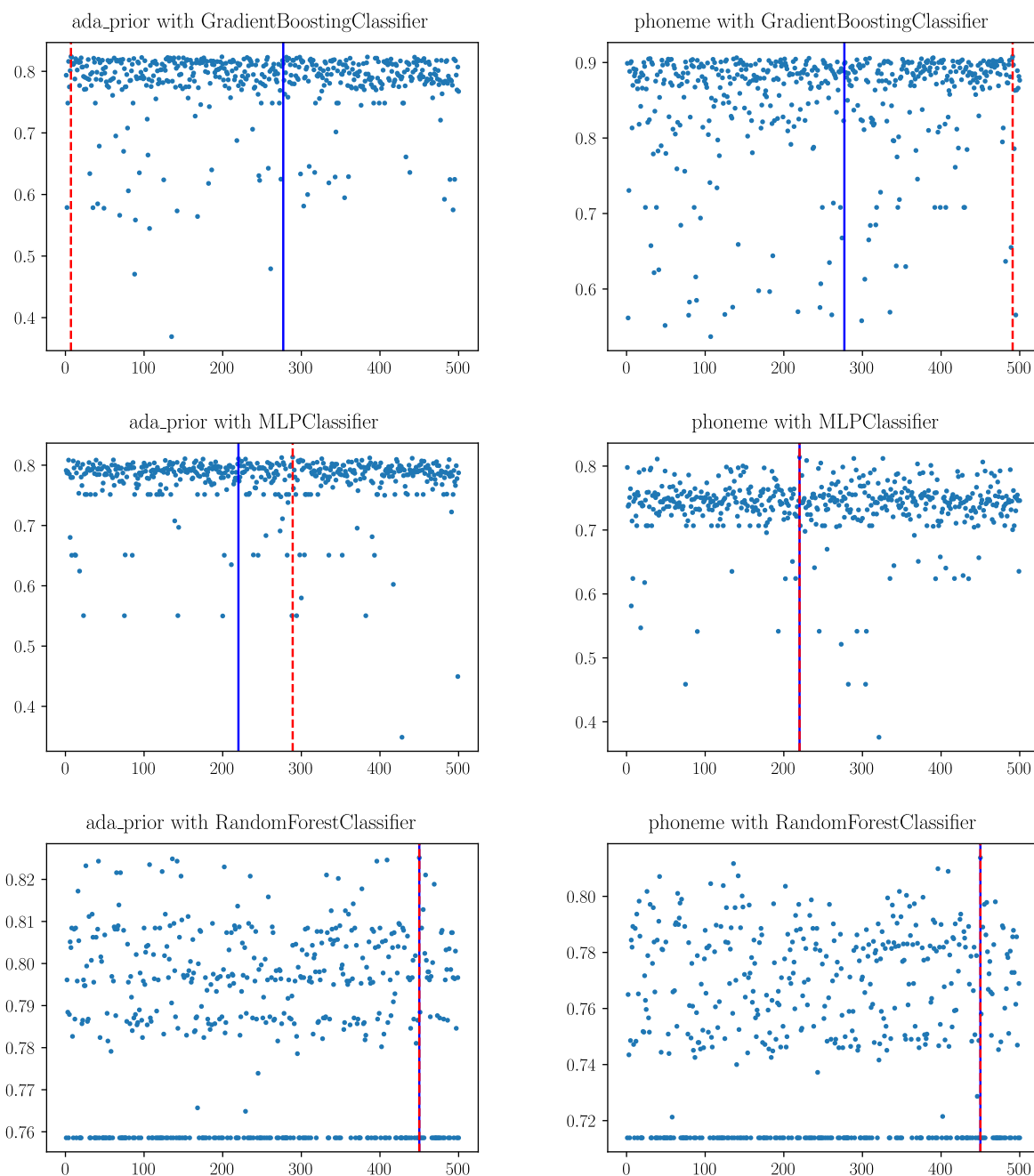
Na wykresach 4 przedstawiono zbieżność optymalizacji bayesowskiej przy optymalizacji wybranych algorytmów podczas uczenia wybranych zbiorów danych.



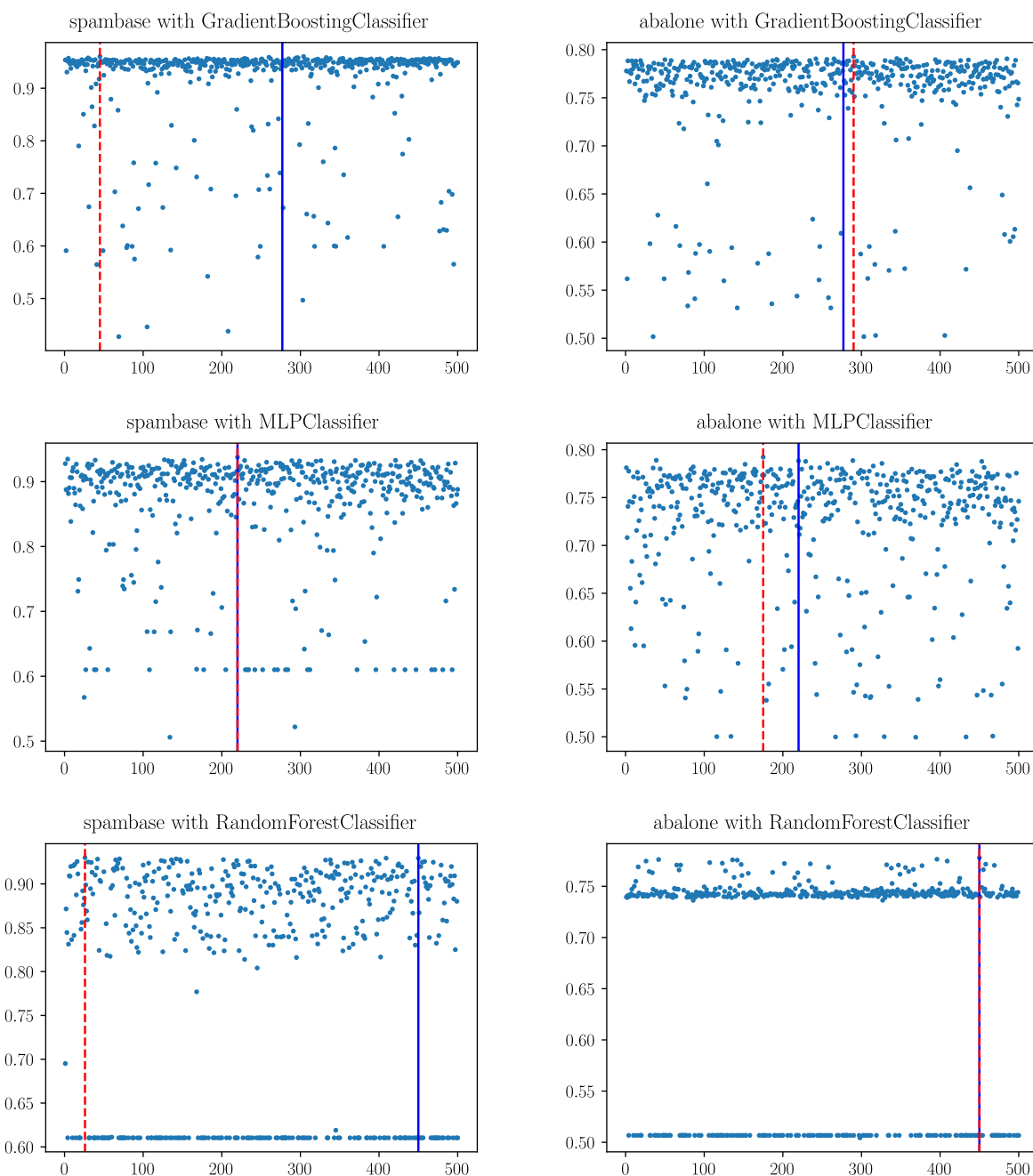
Rysunek 4: Wykres zbieżności optymalizacji bayesowskiej dla wybranych zbiorów przy trenowaniu danych algorytmów metodą „best so far” - najlepszy wynik osiągnięty do dodanej iteracji. Na osi X umieszczony jest numer iteracji, zaś Y prezentuje jej dokładność.

D Wyniki poszczególnych iteracji Random Search

Na wykresach 5 i 6 przedstawiono wyniki osiągnięte w poszczególnych iteracjach każdego spośród wybranych algorytmów dla metody losowania punktów Random Search.



Rysunek 5: Wykres wyników osiągniętych w poszczególnych iteracjach dla każdego algorytmu dla zbiorów danych `ada_prior` i `phoneme`. Na osi X umieszczony jest numer iteracji, zaś Y prezentuje jej dokładność. Czerwona linia prezentuje wynik najlepszy, zaś niebieskie wyniki średnio najlepsze dla danego algorytmu.



Rysunek 6: Wykres wyników osiągniętych w poszczególnych iteracjach dla każdego algorytmu dla zbiorów danych spambase i abalone. Na osi X umieszczony jest numer iteracji, zaś Y prezentuje jej dokładność. Czerwona linia prezentuje wynik najlepszy, zaś niebieskie wyniki średnio najlepsze dla danego algorytmu.