

Praca domowa 1 - AutoML

Hubert Bujakowski
Mikołaj Gałkowski

Listopad 2023

Streszczenie

W niniejszym dokumencie przedstawiono eksperymenty mające na celu analizę tunowalności hiperparametrów wybranych algorytmów uczenia maszynowego, zgodnie z definicjami wprowadzonymi w artykule [1]. Eksperymenty skoncentrowano na ocenie wpływu zmiany poszczególnych hiperparametrów na skuteczność algorytmów w kontekście klasyfikacji binarnej. Otrzymane wyniki mogą stanowić cenne źródło informacji dla praktyków zajmujących się strojeniem modeli maszynowych w różnych dziedzinach zastosowań.

1 Wstęp

Celem pracy domowej było przeanalizowanie tunowalności hiperparametrów 3 wybranych algorytmów uczenia maszynowego na bazie definicji wprowadzonych w artykule [1]. Badania zostały przeprowadzone na czterech tabelarycznych zbiorach danych przeznaczonych do klasyfikacji binarnej, dostępnych na platformie [OpenML](#). Wybrane zbiory danych identyfikowane były poprzez następujące numery ID oraz ich wymiary:

- [44](#) (4601, 58)
- [1504](#) (1941, 34)
- [37](#) (768, 9)
- [1494](#) (1055, 42)

Wybraliśmy zbiory nie posiadające zbyt dużej ilości obserwacji w celu przyspieszenia obliczeń. Do analizy wybraliśmy następujące algorytmy:

- Decision Tree Classifier
- Random Forest Classifier
- XGBoost Classifier

W przypadku każdego z algorytmów przetestowaliśmy dwie metody samplingu punktów:

- Random search
- Bayes optimization

2 Wyniki eksperymentów

2.1 Siatka hiperparametrów

Wybór konkretnych parametrów był podyktowany analizą artykułu naukowego, który stanowił podstawę naszych badań. Dodatkowo zakres niektórych hiperparametrów wyznaczyliśmy empirycznie, aby zdążyć wykonać eksperymenty w czasie.

Tabela 1: Siatka hiperparametrów dla algorytmu RandomForestClassifier

Parametr	Wartości
n_estimators	1, 2, ..., 200
max_depth	10, 11, ..., 50
min_samples_split	2, 3, ..., 20
min_samples_leaf	1, 2, ..., 10
max_features	100 wartości z $U \sim (0.01, 1)$
criterion	"gini", "entropy"
class_weight	None, "balanced"
max_samples	100 wartości z $U \sim (0.01, 1)$

Tabela 2: Siatka hiperparametrów dla algorytmu XGBoostClassifier

Parametr	Wartości
n_estimators	1, 2, ..., 150
max_depth	1, 2, ..., 15
learning_rate	100 wartości z $U \sim (0.01, 1)$
booster	"gbtree", "gblinear", "dart"
gamma	1000 wartości z $U \sim (0.001, 1024)$
subsample	0.1, 0.2, ..., 1
colsample_bytree	10 wartości z $U \sim (0, 1)$
colsample_bylevel	10 wartości z $U \sim (0, 1)$
reg_alpha	1000 wartości z $U \sim (0.001, 1024)$
reg_lambda	1000 wartości z $U \sim (0.001, 1024)$

Tabela 3: Siatka hiperparametrów dla algorytmu DecisionTreeClassifier

Parametr	Wartości
criterion	"gini", "entropy"
splitter	"best", "random"
max_depth	1, 2, ..., 30
min_samples_split	2, 3, ..., 30
min_samples_leaf	1, 2, ..., 30
max_features	100 wartości z $U \sim (0.01, 1)$

W celu wybrania optymalnych ('defaultowych') hiperparametrów, użyty został RandomSearch ustawiony na 500 iteracji na powyższych siatkach ([1] [2] [3]). Wyniki według metryki ROC_AUC zostały uśrednione na wszystkich 4 zbiorach.

Tunowalność każdego z hiperparametrów widoczna na wykresach RandomSearch w [1], [2], [3] nie odsta je od wartości optymalnych ('defaultowych') tj. zawiera się między 1 a 3 kwantylem w znacznej większości, wyjątkiem jest tylko algorytm XGBoost dla hiperparametrów, które miały dużą siatkę z rozkładu $U \sim (0.001, 1024)$ (**gamma**, **reg_alpha**, **reg_lambda**).

Zwiększenie ilości iteracji z 500 do 1000 nie zmieniła znacząco wyników (boxploty miały ten sam charakter). Nasze eksperymenty ograniczony były do naszych prywatnych maszyn przez co testy zakończyliśmy właśnie na maksymalnie 1000 iteracjach. Przez co każdy algorytm osiągał stabilność w okolicach 500 iteracji.

Następnie przeszliśmy do badania tunowalności hiperparametrów w poszczególnych algorytmach. Każdy hiperparametr badany był przez następującą liczbę iteracji:

$$no_iter = len(HYPERPARAMETERS_SPACE[hyperparameter]) \times no_datasets \times 0.8 \quad (1)$$

gdzie:

- *no_iter* - liczba operacji dla danego hiperparametru
- $len(HYPERPARAMETERS_SPACE[hyperparameter])$ - długość tablicy przechowującej potencjalne wartości danego hiperparametru
- *no_datasets* - liczba zbiorów danych - w naszym przypadku wynosi 4

Poszczególne wyniki przedstawiliśmy podobnie jak w artykule przy pomocy boxplotów przedstawionych w dodatku. ([1][2][3])

2.2 Wyniki - Decision Tree Classifier

W przypadku drzewa decyzyjnego każdy z hiperparametrów w tym modelu jest podatny na tunowalność [1]. Zarówno w przypadku losowego samplingu jak i Bayesowskiego możemy zauważyć, że jesteśmy w stanie otrzymać model, który da nam lepsze rezultaty niż default. Ze względu na małą liczbę zbiorów danych, niezbyt dużą liczbę iteracji oraz brak testów statystycznych nie jesteśmy w stanie w pełni wykluczyć sampling bias'u.

2.3 Wyniki - Random Forest Classifier

Podobnie jak w przypadku drzewa, las losowy jest również bardzo podatny na tunowalność [2]. Nietypowym zachowaniem jest tunowalność dla np. parametry `class_weight` czy też `criterion` gdzie podczas losowego samplingu różnica oscylowała w okolicy 0, zaś w przypadku bayesowskiej optymalizacji część wyników znajdowała się znacząco powyżej 0, co jest dla nas zaskakujące.

2.4 Wyniki - XGBoost Classifier

Dla algorytmu XGBoost'a możemy zauważyć, że defaultowe parametry w przypadku hiperparametrów: `gamma`, `reg_alpha` oraz `reg_lambda` nie zostały wyznaczone optymalnie. Na obydwu wykresach wszystkie wartości tunowalności dla tych hiperparametrów znajdują się znacząco na ujemnych wartościach. Pozostałe hiperparametry mają niską tunowalność, lecz mała zmiana w nich może nieco zmienić jakość predykcji modelu.

3 Podsumowanie

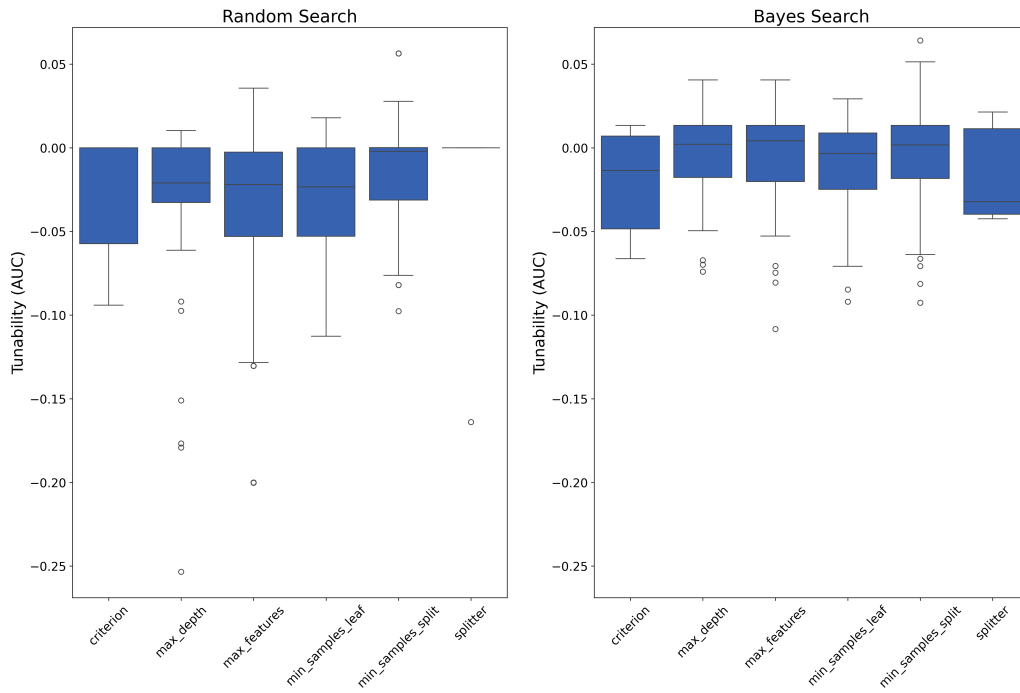
Podsumowując, eksperymenty nad tunowalnością hiperparametrów trzech różnych algorytmów uczenia maszynowego (Decision Tree Classifier, Random Forest Classifier, XGBoost Classifier) zostały szczegółowo przeanalizowane w kontekście ich wpływu na skuteczność modeli w zadaniu klasyfikacji binarnej. Badania obejmowały zastosowanie dwóch metod samplowania punktów: Random Search i Bayes Optimization na czterech różnych zbiorach danych z platformy OpenML.

Wyniki uzyskane z eksperymentów jednoznacznie wskazują na elastyczność hiperparametrów badanych algorytmów, co otwiera możliwość doskonalenia skuteczności modeli poprzez tuning parametrów. Zauważalne różnice w tunowalności między poszczególnymi hiperparametrami podkreślają istotność procesu strojenia modeli.

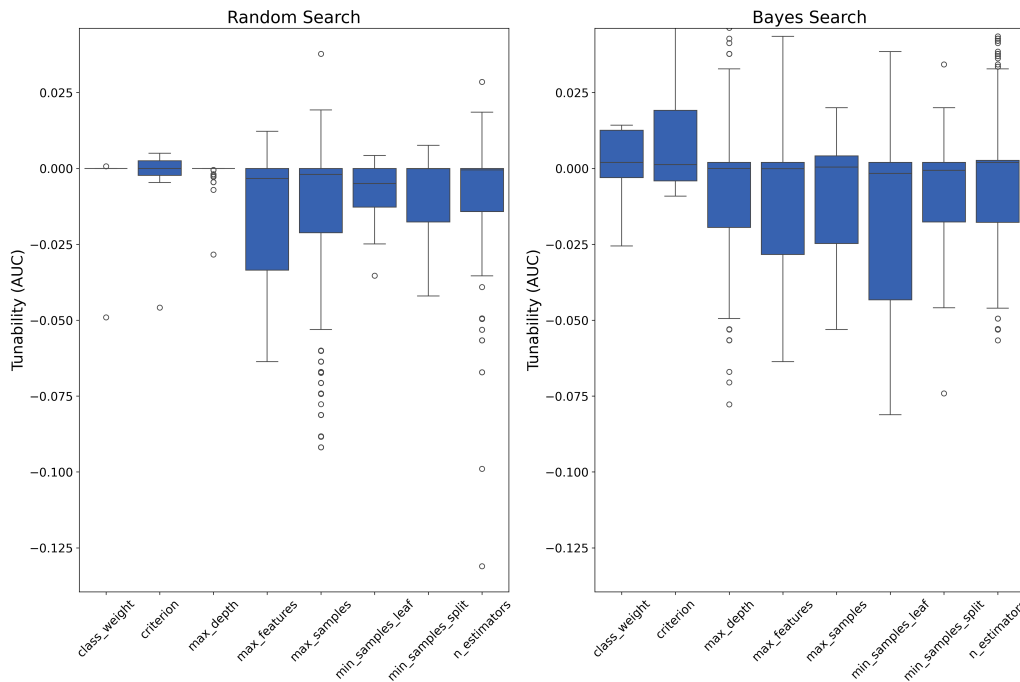
Bibliografia

- [1] Philipp Probst, Anne-Laure Boulesteix, and Bernd Bischl. "Tunability: Importance of Hyperparameters of Machine Learning Algorithms". In: *J. Mach. Learn. Res.* 20.1 (Jan. 2019), pp. 1934–1965. ISSN: 1532-4435.

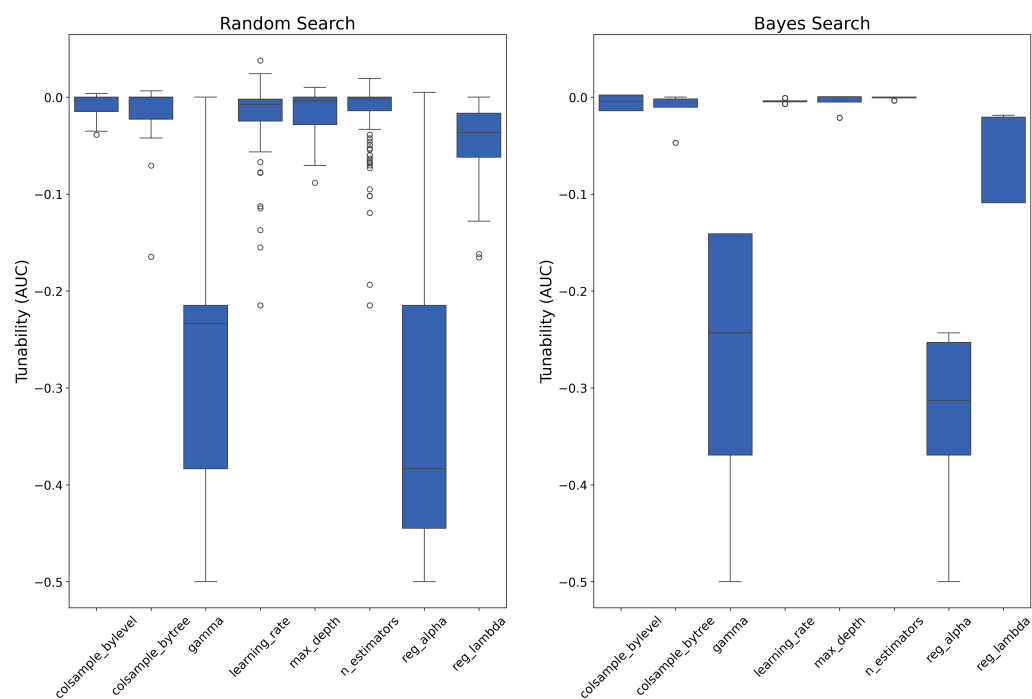
4 Dodatek



Rysunek 1: Boxploty dostosowalności hiperparametrów w algorytmie DecisionTreeClassifier w odniesieniu do optymalnych wartości domyślnych (różnica według metryki ROC.AUC). Wartości poniżej zera oznaczają lepsze AUC w porównaniu do optymalnych hiperparametrów.



Rysunek 2: Boxploty dostosowalności hiperparametrów w algorytmie RandomForestClassifier w odniesieniu do optymalnych wartości domyślnych (różnica według metryki ROC.AUC). Wartości poniżej zera oznaczają lepsze AUC w porównaniu do optymalnych hiperparametrów.



Rysunek 3: Boxploty dostosowalności hiperparametrów w algorytmie XGBoost Classifier w odniesieniu do optymalnych wartości domyślnych (różnica według metryki ROC_AUC). Wartości poniżej zera oznaczają lepsze AUC w porównaniu do optymalnych hiperparametrów.