Automatyczne Uczenie Maszynowe - Projekt 1

Wiktoria Boguszewska i Mateusz Zacharecki

15 listopada 2024

1 Wprowadzenie

1.1 Cel

Celem projektu jest przeanalizowanie tunowalności hiperparametrów 3 wybranych algorytmów uczenia maszynowego na co najmniej 4 zbiorach danych. Do tunowania modeli należy wykorzystać min. 2 różne techniki losowania punktów.

1.2 Dane

W projekcie wykorzystujemy 4 różne zbiory danych dla problemu klasyfikacji binarnej.

1. **Breast Cancer**

   Pierwszy zbiór danych zawiera dane dotyczace raka piersi. Jest to zbiór zawierający 30 numerycznych zmiennych objaśniających typu ciągłego oraz zmienną objaśnianą typu binarnego. Zbiór służy analizie tego czy dana osoba zachorowała na raka piersi.

2. **Credit**

   Drugi zbiór danych klasyfikuje klientów jako dobre lub złe ryzyko kredytowe na podstawie 20 różnych atrybutów, takich jak stan konta oszczędnościowego, czas trwania kredytu, historia kredytowa, ilość zaległych rat oraz dane demograficzne. Zmienna objaśniana jest typu binarnego i określa, czy dany klient jest dobrym, czy złym ryzykiem kredytowym.

3. **Blood Transfusion**

   Trzeci zbiór danych analizuje czy dana osoba oddała krew w marcu 2007 r. Analiza dokonywana jest na bazie zmiennych objaśniających, które określają ile miesięcy minęło od ostatniejgo oddania krwi, ile razy łącznie dana osoba oddawała krew, ile łącznie krwi oddała oraz ile miesięcy minęło odkąd oddawała krew po raz pierwszy. Zbiór danych zawiera 4 zmienne objaśniające typu numerycznego, a zmienna objaśniana jest typu binarnego.

4. **Bank marketing**

   Czwarty zbiór danych zawiera informacje z kampanii marketingowych dla portugalskiej instytucji bankowej, których celem było przekonanie klientów do założenia lokaty terminowej. Zbiór zawiera dane dotyczące klientów, takie jak wiek, zawód, edukacja, status zadłużenia i szczegóły kontaktów podczas kampanii. W zbiorze jest 16 zmiennych objaśniających, a zmienna objaśniana jest typu binarnego i wskazuje na to czy klient założył lokatę terminową.

2 Metody samplingu

Główna część projektu polegała na zastosowaniu różnych metod tunowalności hiperparametrów. Wykonaliśmy to dla 3 algorytmów: regresja logistyczna, KNN oraz XGBoost. Zastosowane przez nas metody samplingu to:

1. **Random Search**

   Random Search polega na losowym próbkowaniu kombinacji hiperparametrów z predefiniowanego zakresu wartości. Jest to metoda skuteczna i stosunkowo szybka, ponieważ nie wymaga przeszukiwania całej siatki

hiperparametrów, co pozwala na efektywne znajdowanie optymalnych parametrów przy ograniczonej liczbie iteracji. W projekcie ta metoda pozwalała na eksplorację wielu kombinacji parametrów przy mniejszym obciążeniu obliczeniowym.

2. **Bayes Search**

   Bayes Search, znany również jako optymalizacja bayesowska, bazuje na probabilistycznym modelu, który wykorzystuje wcześniejsze wyniki do bardziej efektywnego wybierania kolejnych kombinacji hiperparametrów. Proces ten pozwala na skoncentrowanie się na obszarach przestrzeni parametrów o wyższej wydajności, co przyczynia się do szybszej konwergencji do optymalnych wartości. W projekcie użyto tej metody do bardziej precyzyjnego dostrajania modeli, ale czas obliczeniowy był w tym przypadku znacznie dłuższy.

3. **Grid Search**

   Grid Search to metoda systematycznego przeszukiwania przestrzeni parametrów poprzez próbki z siatki predefiniowanych wartości. Dla każdego hiperparametru ustala się z góry określone wartości, a następnie przeszukuje się wszystkie możliwe kombinacje. Choć metoda jest czasochłonna i obciążająca obliczeniowo, zapewnia gruntowną analizę przestrzeni parametrów.

3 Wyniki dla poszczególnych metod

3.1 Random Search

MODEL	**LOGISTYCZNA**				**KNN**				**XGBOOST**			
ZBIÓR	**cancer**	**credit**	**blood**	**bank**	**cancer**	**credit**	**blood**	**bank**	**cancer**	**credit**	**blood**	**bank**
TRAIN ACCURACY	0.9736	0.7175	0.7843	0.8827	0.967	0.7138	0.8027	0.8866	0.9363	0.7025	0.7792	0.8827
TEST ACCURACY	0.9825	0.68	0.72	0.8928	0.9912	0.705	0.7533	0.8961	0.9561	0.685	0.74	0.8928

Tabela 1: Wyniki otrzymane za pomocą metody Random Search.

3.2 Bayes Search

MODEL	**LOGISTYCZNA**				**KNN**				**XGBOOST**			
ZBIÓR	**cancer**	**credit**	**blood**	**bank**	**cancer**	**credit**	**blood**	**bank**	**cancer**	**credit**	**blood**	**bank**
TRAIN ACCURACY	0.9736	0.7213	0.786	0.8836	0.967	0.7175	0.8094	0.8866	0.9758	0.7188	0.796	0.8858
TEST ACCURACY	0.9825	0.685	0.72	0.8961	0.9737	0.685	0.7533	0.8961	0.9649	0.685	0.76	0.9028

Tabela 2: Wyniki otrzymane za pomocą metody Bayes Search.

3.3 Grid Search

MODEL	**LOGISTYCZNA**				**KNN**				**XGBOOST**			
ZBIÓR	**cancer**	**credit**	**blood**	**bank**	**cancer**	**credit**	**blood**	**bank**	**cancer**	**credit**	**blood**	**bank**
TRAIN ACCURACY	0.9692	0.72	0.786	0.8836	0.967	0.7113	0.8044	0.8852	0.967	0.7075	0.7994	0.8852
TEST ACCURACY	0.9912	0.69	0.72	0.895	0.9737	0.705	0.7467	0.8983	0.9649	0.72	0.7533	0.8961

Tabela 3: Wyniki otrzymane za pomocą metody Grid Search.

4 Analiza wyników

4.1 Liczba iteracji

Wszystkie 3 metody optymalizacji hiperparametrów zostały przetestowane poprzez wykonanie 50 iteracji każdej z nich. Ponadto każda z iteracji metody bayesian search została wykonana aktualizując optymalny zestaw hiperparametrów 30 razy.

Analizując wykresy najwyższych accuracy dla kolejnych iteracji oraz poprawy accuracy dla kolejnych interacji, najlepsza wydaje się metoda optymalizacji bayesian search, która stabilizuje się przeważnie już w kilku pierwszych iteracjach, natomiast nowe optymalne parametry zostawały znajdowane w 20 bądź więcej iteracjach zaledwie 2 razy. Dobra wydaje się również być metoda randomized search, która zbiegała często w pierwszych krokach, a niemal zawsze optimum było znajdowane w 30 iteracjach. W pojedynczych przypadkach lepsze hiperparametry zostawały znajdowane w więcej niż 30 krokach, natomiast wówczas poprawa wartości accuracy była nieznaczna. Metoda grid search osiągała zbieżność różnie w zależności od modelu i zbioru. Wpływ na to miał sposób definicji siatki hiperparametrów i wielokrotne testowanie tych hiperparametrów, które nie wpływały na jakość modelu.

4.2 Zakres hiperparametrów

Zakresy hiperparametrów dla poszczególnych modeli i dla metod optymalizacji randomized search oraz bayesian search określono bazując na definicjach opisanych w artykule *Tunability: Importance of Hyperparameters of Machine Learning Algorithms*. Z kolei siatka parametrów dla metody grid została utworzona również bazując na tym artykule, ale w taki sposób, żeby metoda wykonała się 50 razy dla każdego z algorytmów.

Algorytm	**Hiperparametr**	**Typ**	**Dolna granica**	**Górna granica**	**Rozkład**
LogisticRegression	C	numeric	10^{-10}	10^{10}	loguniform
	l1_ratio	numeric	0	1	uniform
KNeighborsClassifier	n_neighbors	integer	1	30	uniform
	weights	discrete	–	–	uniform
	p	integer	1	2	uniform
XGBClassifier	booster	discrete	–	–	uniform
	eta	numeric	10^{-10}	1	loguniform
	n_estimators	integer	1	500	uniform
	max_depth	integer	1	10	uniform
	subsample	numeric	0.1	1	uniform
	colsample_bytree	numeric	0	1	uniform
	colsample_bylevel	numeric	0	1	uniform
	lambda	numeric	10^{-5}	10^{2}	loguniform
	alpha	numeric	10^{-5}	10^{2}	loguniform
	min_child_weight	numeric	1	10^{7}	loguniform

Tabela 4: Zakresy hiperparametrów dla metody Random Search oraz Bayes Search.

Algorytm	**Hiperparametr**	**Wartości**
LogisticRegression	C	10^{-10}, 10^{-5}, 10^{-3}, 10^{-1}, 1, 10, 100, 10^{3}, 10^{5}, 10^{10}
	l1_ratio	0, 0.2, 0.6, 0.9, 1
KNeighborsClassifier	n_neighbors	$1, 2, \ldots, 20, 22, 24, 26, 28, 30$
	weights	*uniform*, *distance*
XGBClassifier	eta	10^{-3}, 10^{-2}, 10^{-1}, 0.3, 0.5
	n_estimators	$50, 100, 200, 300, 400$
	max_depth	$5, 10$

Tabela 5: Zakresy hiperparametrów dla metody Grid Search.

4.3 Tunowalność poszczególnych algorytmów

Celem zmierzenia tunowalności poszczególnych algorytmów stosując metodę randomized search, wyznaczono defaultowe zestawy hiperparametrów, tzn. takie zestawy hiperparametrów, które maksymalizują średnią wartość accuracy dla wszystkich rozważanych zbiorów danych. Następnie wyznaczono tunowalność algorytmów w sposób jaki został opisany w artykule *Tunability: Importance of Hyperparameters of Machine Learning Algorithms*. Wyniki przedstawia poniższa tabela.

Model	**Defaultowe hiperparametry**	**Tunowalność**
Regresja logistyczna	C: 6.563234996980466e-06, l1_ratio: 0.34345601404832493	$-3.5014 \cdot 10^{-6}$
kNN	weights: uniform, p: 2, n_neighbors: 18	-0.0004
XGBoost	alpha: 0.006066641915981054, booster: gbtree, colsample_bylevel: 0.141194832453028, colsample_bytree: 0.5588264346475861, eta: 2.4708290259874402e-08, lambda: 2.051302466178794, max_depth: 8, min_child_weight: 5.4627652310036785, n_estimators: 237, subsample: 0.2477228436396202	0

Tabela 6: Defaultowe hiperparametry i tunowalność dla metody Random Search.

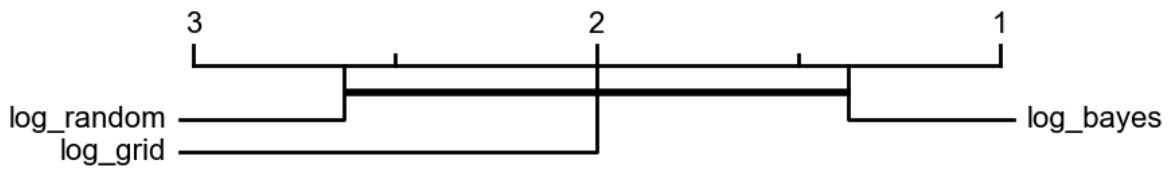
Tunowalność dla każdego z algorytmów jest bardzo bliska bądź równa 0. Może być to wynik doboru zbyt małej liczby zbiorów o niewielkich rozmiarach. W przypadku modeli regresji logistycznej i kNN, jedynym zbiorem, dla którego otrzymujemy niezerową tunowalność jest Blood Transfusion.

4.4 Testy statystyczne

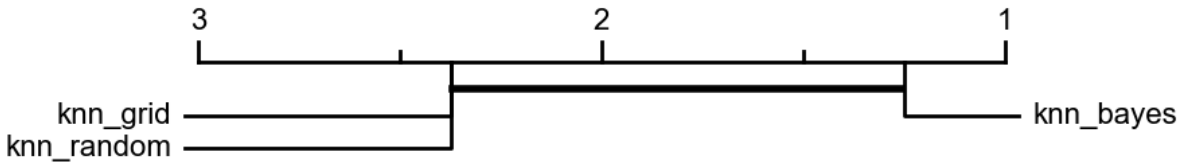
Zastosowano także test Manna-Whitneya do porównania różnic wyników pomiędzy technikami losowania hiperparametrów. W ten sposób porównano metody randomized search oraz bayesian search wykonując łącznie 12 testów statystycznych, osobno dla wyników uzyskanych z każdego modelu i dla każdego zbioru. Okazuje się, że w każdym przypadku odrzucono hipotezę zerową twierdząc, że różnica wyników z obu modeli jest istotna statystycznie i wyniki te pochodzą z różnych rozkładów. Można więc wnioskować, że występuje bias sampling.

4.5 Critical Difference Diagrams

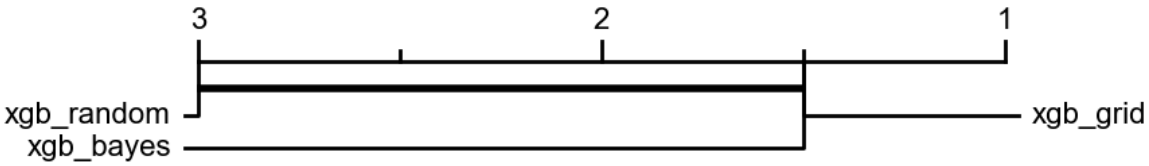
Diagramy Critical Difference sugerują zdecydowaną przewagę metody bayesian search w porównaniu z pozostałymi metodami. Z kolei opierając się na wizualizacjach, randomized search wydaje się działać najgorzej plasując się na każdym diagramie na ostatnim miejscu. Metoda grid search ma różną skuteczność w zależności od modelu, działając średnio dla regresji logistycznej, porównywalnie źle do randomized searcha dla kNN i porównywalnie dobrze do bayesian search dla XGBoosta.



Rysunek 1: Critical Difference Diagram dla metod Random Search, Bayes Search, Grid Search i modelu regresji logistycznej.



Rysunek 2: Critical Difference Diagram dla metod Random Search, Bayes Search, Grid Search i modelu kNN.



Rysunek 3: Critical Difference Diagram dla metod Random Search, Bayes Search, Grid Search i modelu XGBoost.