Faculty of Mathematics and Information Science

WARSAW UNIVERSITY OF TECHNOLOGY

Piotr Robak, Agata Węglerska

AutoML
Homework1

November 22, 2023

1 Wprowadzenie

Dokument ten przedstawia wyniki starań dotyczących znalezienia optymalnych hiperparametrów dla różnych algorytmów klasyfikacji binarnej wykonane przy pomocy kodu napisanego w języku Python. W ramach prac wykorzystano dwie techniki przeszukiwania przestrzeni parametrów:

- Randomized search
- Bayesian Optimization

Przy pomocy tego pierwszego wyznaczono również parametry podstawowe, czyli średnio najlepsze na wszystkich testownaych zbiorach, dzięki czemu można było sprawdzić o ile parametry wyznaczane przez podane techniki okazują się być lepsze od takich które najlepiej się sprawdzają dla nieznanego zbioru danych.

Do przeszukiwania przestrzeni zostały wykorzystane metody RandomizedSearchCV() oraz BayesSearchCV(). Pierwsza należąca do biblioteki sklearn, druga do skopt.

Postanowiliśmy poszukiwać parametrów dla następujących algorytmów klasyfikacji:

1. Random Forest
2. Gradient Boosting Classifier
3. KNN

.

2 Wybór danych

Postanowiliśmy przeprowadzić testy, na 4 zbiorach danych uzyskanych z platformy OpenML. Dzięki temu uzyskane wyniki na pewnym stopniu są niezależne od konkretnego zbioru przez co wyznaczone parametry podstawowe, oraz wnioski można uogólnić. Każdy ze zbiorów dotyczy klasyfikacji binarnej.

- Zbiór zawierający informację o klasyfikacji maila jako spamu.
- Zbiór zawierający informację o występowaniu defektów.
- Zbiór zawierający informację o tym czy dana osoba oddała krew.
- Zbiór zawierający informację o tym czy dany klient przedłużył umowę.

3 Przebieg doświadczenia - opis ogólny

Na początku dane poddano preprocessingowi polegającemu na zastosowaniu metody MinMaxScaler() dla wartości numerycznych oraz metody OneHotEncoder() dla wartości kategorycznych. Następnie dla każdego z algorytmów przeprowadzono przeszukiwanie Randomized-Search dla 50 iteracji. Na postawie wyników dla każdego zbioru danych, wybrano parametry średnio najlepsze, to znaczy takie średnia po wszystkich zbiorach danych z wyników szkolenia była dla nich najlepsza.

Następnie dla każdego ze zbiorów danych wyznaczono parametry optymalne wykorzystując Randomized-Search dla tego konkretnego zbioru, po czym porównano wyniki z parametrami podstawowymi.

Wyznaczono parametry optymalne wykorzystując optymalziację bayesowską.

Następnie wyznaczono różnice w dokładności pomiędzy wyznaczonymi przy użyciu optymalizacji bayesowskiej parametrami optymalnymi,a wyznaczonymi parametrami podstawowymi.

Wyznaczono poziom serotoniny we krwi studentów, okazał się on spadać czasem wykonywania zadania, jednakże można było zaobserwować kilka znacznych chwilowych wzrostów.

4 Otrzymane wyniki

4.1 Wyniki dla przeszukiwania Randomized-Search

Ilość iteracji wraz z otrzymanymi średnimi poprawnościami klasyfikacji dla przeszukiwania Randomized-Search przedstawiają wykresy w folderze Wyniki/<nazwa algorytmu>/RandomSearch.

4.2 Wyniki dla przeszukiwania Bayesian-Search

Ilość iteracji wraz z otrzymanymi średnimi poprawnościami klasyfikacji dla przeszukiwania Bayesian-Search przedstawiają wykresy w folderze Wyniki/<nazwa algorytmu>/Bayes.

5 Zakresy wybranych hiperparametrów

Dla poszczególnych algorytmów zostały wybrane różne przestrzenie hiperparametrów. Intuicyjny wybór opierał się głównie na zrozumieniu co każdy z parametrów reprezentuje.

5.1 Random Forest Classifier

Zakresy hiperparametrów.

- n estimators = wektor z przedziału [10,200] co 10 punktów
- min samples leaf = wektor z przedziału [1,50] co 5 punktów
- random state = wektor z przedziału [1]
- max depth = wektor z przedziału [10,110] co 11 punktów
- min samples = wektor z przedziału [2,10] co 4 punktów

5.2 SVM

Zakresy hiperparametrów.

- svm c = wektor z przedziału [1,200] co 10 punktów
- coef0 = wektor z przedziału [1,50] co 5 punktów
- gamma = wektor z przedziału [1,50] co 5 punktów
- degree = wektor z przedziału [1,10] co 5 punktów

5.3 XGboost

Zakresy hiperparametrów.

- n neighbors = wektor z przedziału [5,100] co 10 punktów
- algorithm = wektor z przedziału ['ball tree', 'kd tree', 'brute']

6 Tunowalność wykorzystanych algorytmów

Dzięki przeszukiwaniu Bayesian Optimization można było dokonać tunowalności każdego z algorytmów. Wyniki tego przeszukiwania wyszły różne, w niektórych przypadkach nie udało się uzyskać lepszych hiperparametrów niż te defaultowe. Wyniki te przedstawiają wykresy znajdujące się w załącznikach w folderze Wyniki/<nazwa algorytmu>/boxplot.

7 Wpływ techniki losowania punktów na sposoby tunowania (hiperparametrów/algorytmów)

Technika losowania punktów ma wpływ ze względu na bias sampling, a zatem zagrożenie wynikające z możliwości pominięcia niektórych punktów, które byłyby dobrym wyborem. Random search losuje punkty i może okazać się, że niektóre właściwie pominął, wybierając gorsze. Podobnie grid search ze względu na dyskretne wartości może pewne punkty pominąć. Bayes search z kolei najbardziej jest odporny na tego typu zagrożenie.