

Auto ML - praca domowa 1

Maciej Malewicz, Antoni Chudy

1 Wstęp

Poniższy projekt jest związany z zagadnieniem tunowalności hiperparametrów wybranych algorytmów uczenia maszynowego. Problem ten jest opisany w artykule „Tunability: Importance of Hyperparameters of Machine Learning Algorithms”, którego autorami są Philipp Probst, Anne-Laure Boulesteix oraz Bernd Bischl. Do przeprowadzonego eksperymentu zostały wybrane 3 algorytmy:

- Elastic net z hiperparametrami `C` oraz `l1_ratio`,
- drzewo decyzyjne z hiperparametrami `max_depth`, `min_samples_split`, `min_samples_leaf`, `max_leaf_nodes`, `ccp_alpha`,
- las losowy z hiperparametrami `max_depth`, `min_samples_split`, `min_samples_leaf`, `max_leaf_nodes`, `ccp_alpha`, `bootstrap`, `warm_start`.

Celem eksperymentu jest porównanie, jak dla ustalonych zbiorów danych i rozważanych algorytmów uczenia maszynowego radzą sobie dwie różne metody szukania optymalnych hiperparametrów. Tymi metodami są: losowanie hiperparametrów z rozkładu jednostajnego (za pomocą **RandomizedSearchCV**) oraz technika bayesowska (za pomocą **BayesSearchCV**).

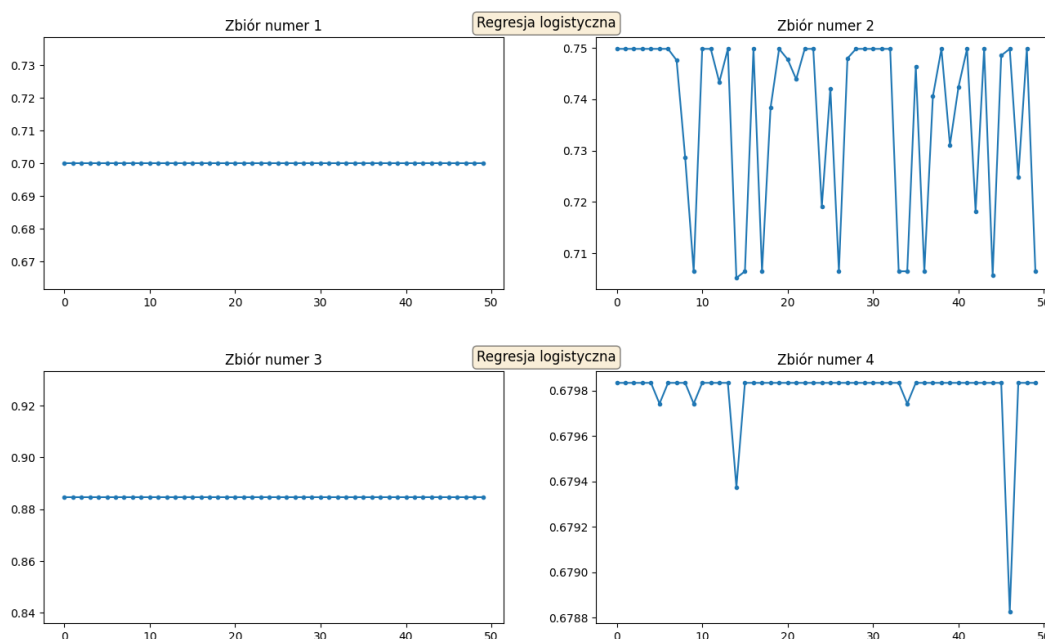
2 Przebieg eksperymentu

W pierwszej części eksperymentu punkty są wybierane z rozkładu jednostajnego. Dla każdego modelu liczymy średnią wartość *accuracy* z czterech zbiorów dla każdej iteracji. Następnie wyznaczamy θ^* , który jest maksimum po wszystkich obliczonych średnich wartościach *accuracy*. W drugiej części doświadczenia stosujemy podejście Bayesowskie. Wyznaczone tą metodą wartości *accuracy* dla poszczególnych modeli porównujemy z wartością θ^* odpowiednią dla danego modelu. Sprawdzimy również czy wyniki techniki bayesowskiej stabilizują się od pewnej iteracji, czy też nie. Na koniec zastanowimy się także czy dla poszczególnych modeli wyniki obu metod różnią się istotnie od siebie. Każdy algorytm na każdym zbiorze wykonuje po 50 iteracji dla obu metod.

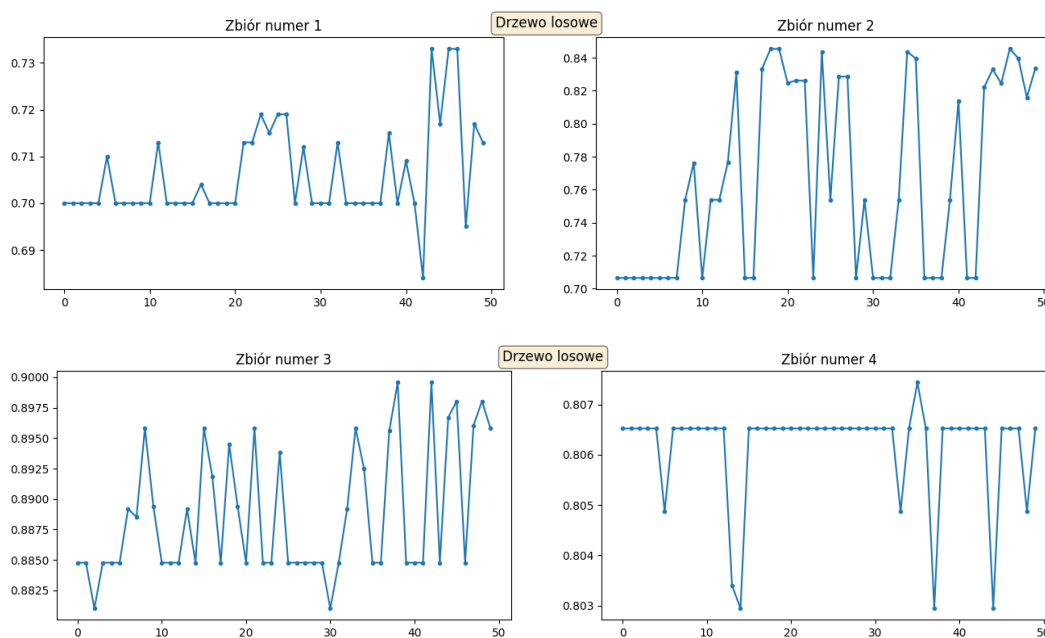
3 Uzyskane wyniki

3.1 Zbieżność wyników

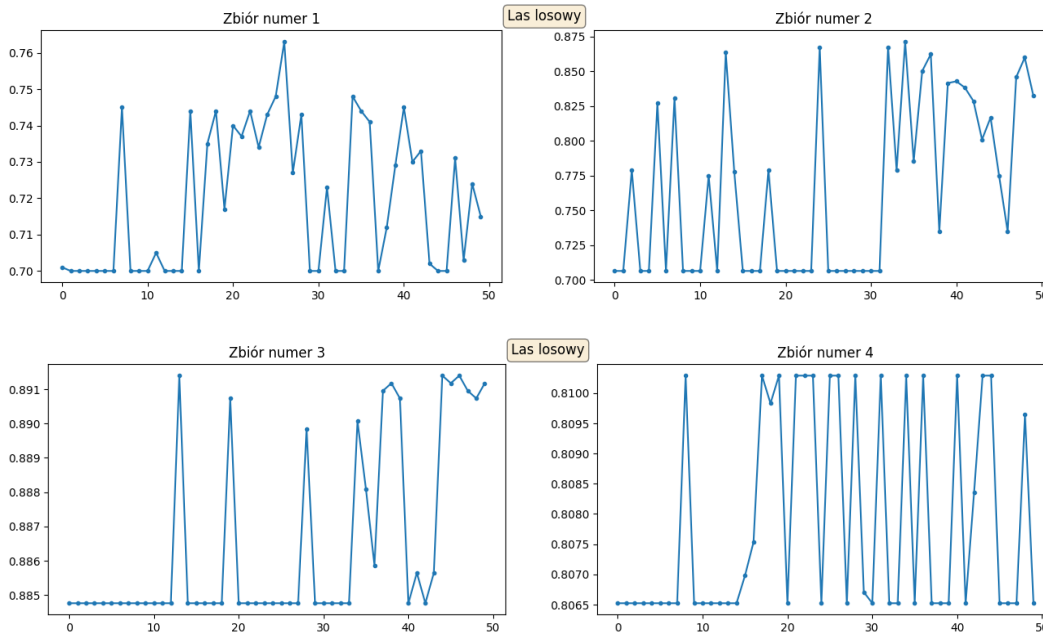
W tej podsekcji dla każdego algorytmu i dla każdego zbioru danych, wizualnie sprawdzimy czy rozważana ilość iteracji (50) na przeszukiwanie ustalonych siatek hiperparametrów jest wystarczająca, aby uzyskać stabilne wyniki optymalizacji dla przeszukiwania bayesowskiego. Innymi słowy, czy od pewnej iteracji funkcja **BayesSearchCV** zwraca bardzo zbliżone do siebie wyniki, czy jednak zwracane wyniki są „skokowe”. Poniższej, dla każdego zbioru ze względu na kolejne algorytmy przedstawione są wykresy z wynikami kros-walidacji na kolejnych iteracjach funkcji **BayesSearchCV**:



Dla regresji logistycznej jedynym „ciekawym” wynikiem jest wykres dot. zbioru numer 2 (przy pozostałych wykresach ciężko powiedzieć, czy uzyskane wykresy pokazują stabilność wyników, czy niemożność polepszenia ich za pomocą tuningu hiperparametrów), niestety wyniki są skokowe i nie widać żadnej stabilizacji.



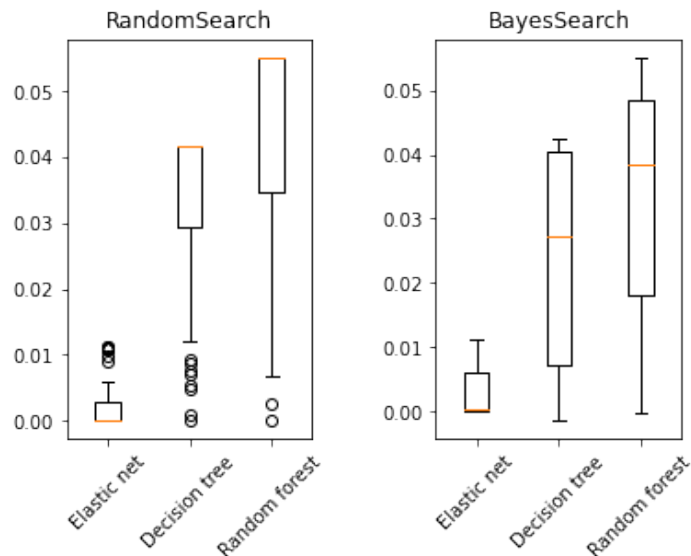
Jeśli chodzi o drzewo losowe, to tutaj również ciężko cokolwiek powiedzieć o stabilności, jednakże obiecujące wydaje się być to, że na zbiorze numer 1 i 2 algorytm przy końcowych iteracjach zwrócił przeciętnie wyższe wyniki niż we wcześniejszych fazach iteracji.



Dla lasu losowego: pewną stabilizację można dostrzec w pierwszym zbiorze, w środkowej fazie iteracji, niestety algorytm w późniejszych fazach znów wygląda jakby losowo błędził. Dla zbiorów o numerach 2 i 3, końcówki iteracji wydają się być ciekawe, ponieważ uzyskiwane wyniki wyglądają jakby były większe przy początkowych iteracjach. Z kolei wyniki na czwartym zbiorze są bardzo skokowe.

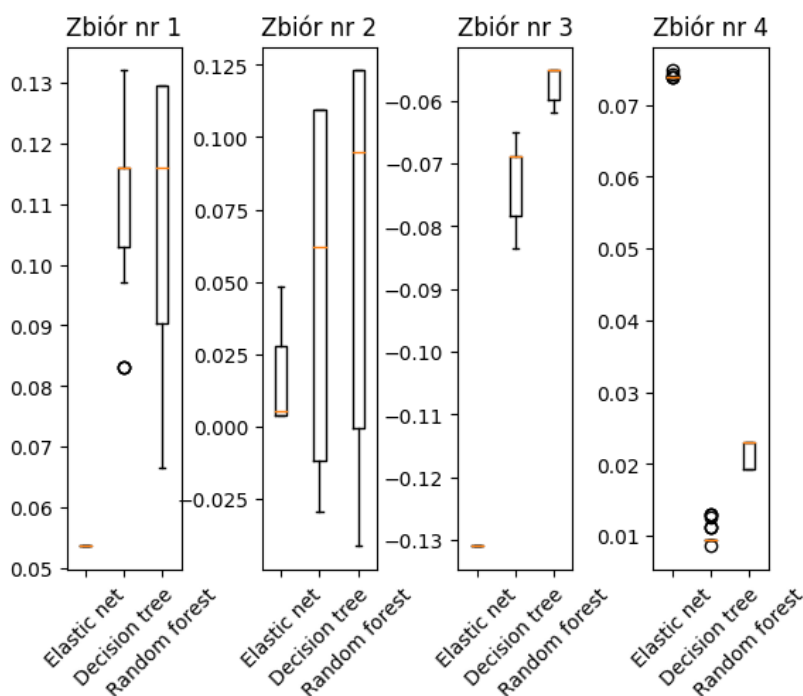
W wielu przypadkach nie dało się dostrzec nic konkretnego jeśli chodzi o zbieżność wyników. Natomiast w pojedynczych przypadkach, gdyby tylko rozważyć większą ilość iteracji, jest nadzieja, że wyniki zaczęły by się stabilizować.

3.2 Tunowalność dla obu metod



Zauważmy, że w podejściu Bayesowskim wyniki są bardziej rozrzucone. W przypadku algorytmu RandomSearch dla drzew i lasów losowych górna połowa masy prawdopodobieństwa znajduje się w jednym punkcie.

3.3 Różnorodność wyników modeli dla metody BayesSearch dla poszczególnych zbiorów



Powyższe wyniki wskazują, że w zależności od użytego zbioru, jego struktury, rozrzut wyników jest zauważalnie różny.

4 Testy statystyczne

Dla każdego modelu postawiliśmy hipotezę czy średnie wyniki uzyskane wyszukiwaniem za pomocą **RandomizedGridSearch** istotnie się różnią od wyników uzyskanych wyszukiwaniem **BayesSearchCV**. Innymi słowy, jeśli dla ustalonego modelu X oznacza średnie wyniki uzyskane za pomocą **RandomizedGridSearch**, a Y oznacza średnie wyniki uzyskane za pomocą **BayesSearchCV**, to za pomocą testu Wilcoxona **rank sums** zweryfikowaliśmy hipotezę zerową czy X i Y pochodzą z tego samego rozkładu przy dwustronnej hipotezie alternatywnej.

Dla poszczególnych modeli uzyskaliśmy następujące p -wartości: Regresja logistyczna: 0.345, drzewo decyzyjne: 0.005, las losowy: 0.010.

Zakładając poziom istotności 0.05 z uzyskanych wyników otrzymujemy, że dla regresji logistycznej nie mamy podstaw do odrzucenia hipotezy zerowej, natomiast dla drzewa decyzyjnego oraz lasu losowego podejmujemy decyzję o odrzuceniu hipotezy zerowej. Zauważmy, że w przypadku regresji logistycznej można powiedzieć, że wyniki są zbliżone do siebie (można to również potwierdzić patrząc na podobieństwa boxplotów dot. tunowalności dla poszczególnych metod).

5 Podsumowanie

Przeprowadzone doświadczenie pokazało, że dla drzew decyzyjnych i lasów losowych przy niewielkiej liczbie iteracji ciężko uzyskać stabilność wyników. Natomiast jeśli chodzi o tunowalność, to w boxplotach dotyczących tunowalności można zauważyć, że wyniki dla drzewa decyzyjnego oraz lasu losowego mają tendencję do przyjmowania bardziej różnorodnych wartości w metodzie bayesowskiej, w porównaniu do metody losowego doboru hiperparametrów. Zauważmy na koniec, że uzyskane wyniki mogłyby się różnić w zależności od zbiorów przyjętych do analizy. Lepsze wyniki uzyskalibyśmy wykonując większą liczbę iteracji, jest to jednak trudne z uwagi na ograniczoną moc obliczeniową.

6 Bibliografia

- Philipp Probst, Anne-Laure Boulesteix, Bernd Bischl Tunability, Importance of Hyperparameters of Machine Learning Algorithms
- <https://scikit-learn.org/stable/>
- <https://scikit-optimize.github.io/stable/>
- <https://scipy.org/>