Analiza tunowalności hiperparametrów wybranych algorytmów uczenia maszynowego

Krzysztof Sawicki Kacper Wnęk

November 2024

Spis treści

1 Zbiory dane oraz algorymy

1.1 Zbiory

Do przeprowadzenia eskperymentów będziemy korzystać z 4 zbiorów pochodzących z platformy kaggle.

- Flight price (30000 wierszy , 11 kolumn)
- Diamonds Price (10000 wierszy, 10 kolumn)
- Cellphone Price (162 wiersze, 18 kolumn)
- Salary Prediction (1000 wierszy, 7 kolumn)

Wszystkie wymienione zbiory służą do zadania regresji. Przed przeprowadzeniem eksperymentów zostały one przetworzone przy pomocy narzędzia Pipeline, w którym braki danych zostały zatępione medianą w przypadku danych liczbowych oraz constantem w przypadku danych tekstowych. Zmienne typu string zostały również poddane procesowi One Hot Encoding z parametrem *handleunknown* ustawionym na ignore.

1.2 Algorytmy

Do przeprowadzenia eksperymentów zostały wybrane następujące algorytmy:

- XGBRegressor z pakietu **xgboost**
- Random Forest z pakietu **scikit-learn**
- ElasticNett z pakietu **scikit-learn**

2 Eksperyment

2.1 Wybór siatki hiperparametrów

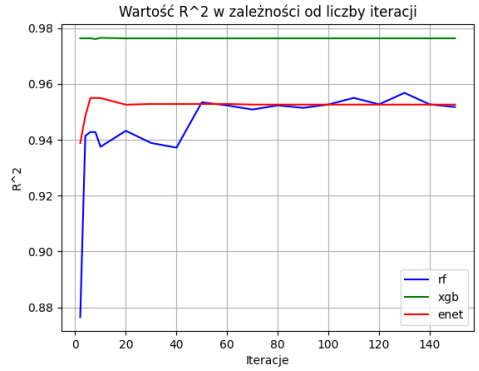
Zakres parametrów został dobrany na podstawie lektury artykułu [1]. Analizie zostały poddane tylko najpopularniejsze parametry, charakterystyczne dla danego modelu.

Model	**Parametr**	**Min**	**Max**	**Rozkład**
Random Forest	n_estimators	1	500	randint
Random Forest	min_samples_split	2	10	randint
Random Forest	min_samples_leaf	2	10	randint
Elastic Net	alpha	0	1	uniform
Elastic Net	l1_ratio	0	1	uniform
XGB	min_child_weight	2^0	2^7	uniform (2^x)
XGB	max_depth	1	15	randint
XGB	colsample_bytree	0	1	uniform

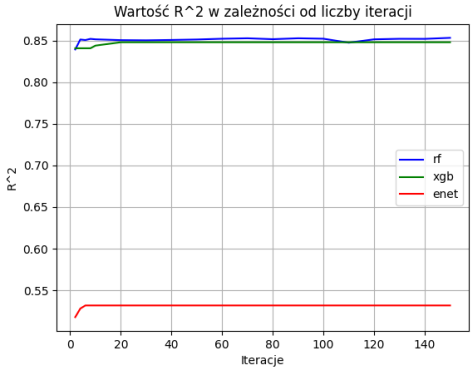
Tabela 1: Parametry modeli, ich zakresy i rozkłady

2.2 Eksperyment - Random Search

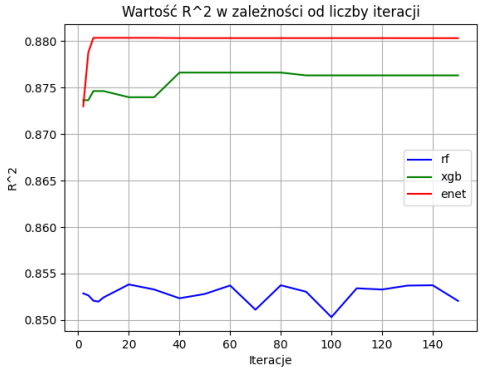
W ramach metody wyboru punktów z rozkładu jednostajnego zdecydowaliśmy się na Random Search. Jest to metoda optymalizacji hiperparametrów modeli, w której wartości parametrów są losowo wybierane z określonych przedziałów lub rozkładów. W porównaniu do Grid Search, Random Search może szybciej znaleźć optymalne rozwiązania, ponieważ bada większą różnorodność kombinacji przy mniejszej liczbie prób.
Kryterium optymalizacji była miara R^2. Została ona wybrana, ponieważ w odróżnieniu od **MSE** można jej użyć do porównywania wyników dla różnych zbiorów danych. W tym przypadku **MSE** jest za bardzo zależne od skali zmiennej objaśnianej. Liczba iteracji dla Random Search była stopniowo zwiększana. Najpierw co 2, a od 10 co 10 aż do maksymalnej wartości 150. Parametr **cv** był ustawiony na 5. W celu zapewnienia powtarzalności eksperymentu, parametr **random state** został ustawiony na 42. Wyniki maksymalnego R^2 dla każdego modelu i datasetu można zaobserwować na wykresach:



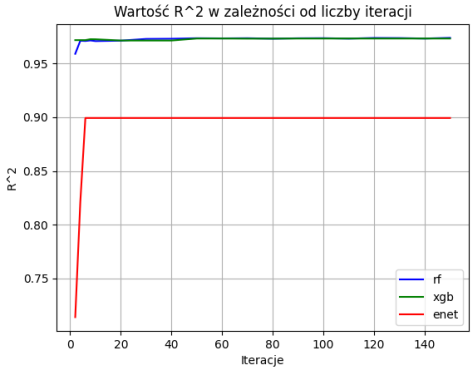
(a) Wykres dla zbioru phone



(b) Wykres dla zbioru flight



(c) Wykres dla zbioru salary



(d) Wykres dla zbioru diamonds

Rysunek 1: Wykresy zależności R^2 od liczby iteracji dla metody Random Search

Jak możemy zaobserwować na wykresach największy wzrost R^2 otrzymujemy w pierwszych kilku iteracjach algorytmu, chodź nie jest to regułą. Już po około 50 iteracjach osiągamy względną stabilność niezależnie od rozpatrywanego zbioru. Jest to dosyć dobry wynik ponieważ relatywnie szybko możemy uzyskać akceptowalne rezultaty niezależnie od zbioru danych. W przypadku modelu Elastic Net można zauważyć największą stabilność między kolejnymi rezultatami. Może to wynikać ze struktury modelu, gdyż jest to dosyć prosty algorytm oraz tego, że siatka parametrów była istotnie mniejsza niż w pozostałych przypadkach.
Patrząc na wyniki dla poszczególnych zbiorów danych:

- Dla zbioru 'phone' stabilność osiągana jest po około 50 iteracjach, z wyraźnym prowadzeniem modelu XGB, który jest stabilny od samego początku. ElasticNet wykazuje najmniejszą zmienność co oznacza, że przestrzeń parametrów została dobrze zeksplorowana. Dla modelu Random

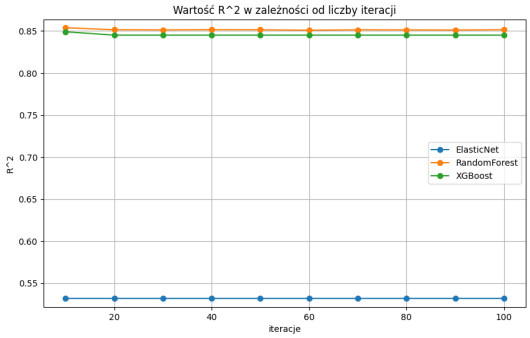
Forest można zaobserwować znaczący wzrost w pierwszych iteracjach.

- W zbiorze 'flight' wyniki stabilizują się bardzo szybko, bo już po około 10 iteracjach. Tutaj Random Forest i XGB osiągają niemal identyczne wyniki, podczas gdy ElasticNet, pozostaje na znacznie niższym poziomie R^2.
- W zbiorze 'salary' model Random Forest nie wykazuje tej samej stabilności co ElasticNet oraz XGB jednocześnie posiadając najgorsze wyniki. Świetnie poradził sobie model Elastic Net, który oprócz najlepszej stabilności wyróżnia się również wynikami.
- W zbiorze 'diamonds' różnice między iteracjami są minimalne, a stabilność najszybciej osiąga Elastic Net. Wykonuje on również bardzo duży skok w pierwszych iteracjach. Modele XGB oraz Random Forest są od początku lepiej przystosowane do tego zbioru.

2.3 Eksperyment - Bayes Search

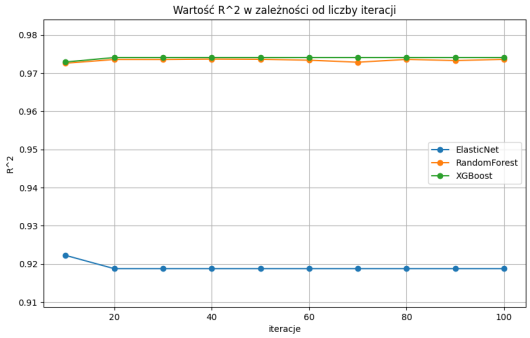
W naszych eksperymentach do optymalizacji hiperparametrów zdecydowaliśmy się wykorzystać **Bayes Search**. Jest to metoda, która zamiast losowego wyboru wartości, jak w Random Search, wykorzystuje podejście probabilistyczne. Dzięki temu Bayes Search koncentruje się na najbardziej obiecujących obszarach przestrzeni parametrów, co pozwala szybciej osiągnąć lepsze wyniki. Podobnie jak wcześniej, jako kryterium optymalizacji wybraliśmy miarę R^2. Liczbę iteracji stopniowo zwiększaliśmy co 10, zaczynając od 10 i dochodząc do 100. Dodatkowo zastosowaliśmy walidację krzyżową z parametrem **cv** równym 5, aby wyniki były bardziej stabilne. Aby zapewnić powtarzalność eksperymentów, ustawiliśmy **random state** na wartość 42. Każdy model był trenowany na zadanych przestrzeniach hiperparametrów, a wyniki R^2 dla różnych zbiorów danych można zaobserwować na wykresach:

(a) Wykres dla zbioru phone



(b) Wykres dla zbioru flight

(c) Wykres dla zbioru salary



(d) Wykres dla zbioru diamonds

Rysunek 2: Wykresy zależności R^2 od liczby iteracji dla metody Bayes Search

W przypadku zastosowania **Bayes Search** zauważamy, że największe wzrosty miary R^2 mają miejsce w początkowych iteracjach, podobnie jak w Random Search. Jednak w przeciwieństwie do niego, wyniki są bardziej stabilne i szybciej osiągają satysfakcjonujące wartości. Już po około 30 iteracjach większość modeli osiąga względną stabilność wyników.
Patrząc na wyniki dla poszczególnych zbiorów danych:

- Dla zbioru 'phone' stabilność osiągana jest po około 30 iteracjach, z wyraźnym prowadzeniem modelu XGB. ElasticNet wykazuje mniejszą zmienność, co sugeruje, że przestrzeń parametrów tego modelu została dobrze eksplorowana i nie wymaga dalszych zmian.
- W zbiorze 'flight' wyniki stabilizują się bardzo szybko, bo już po około 20 iteracjach. Tutaj Random Forest i XGB osiągają niemal identyczne wyniki, podczas gdy ElasticNet, choć mniej zmienny, pozostaje na niższym poziomie R^2.
- Zbiór 'salary' pokazuje większe fluktuacje, zwłaszcza dla modelu XGB, ale stabilność wyników osiągana jest po około 50 iteracjach. Ponownie, ElasticNet wykazuje najmniejszą zmienność, co potwierdza jego prostszą strukturę i mniejszą przestrzeń parametrów.
- W przypadku zbioru 'diamonds' różnice między iteracjami są minimalne, a stabilność osiągana jest najszybciej, bo już po 20 iteracjach. Modele Random Forest oraz XGB osiągają najwyższe wartości R^2, podczas gdy ElasticNet pozostaje na stałym poziomie.

3 Tunowalność

W celu przeanalizowania tunowalności poszczególnych algorytmów skorzystaliśmy z definicji zawartej w [1], przyjmując zbiór defaultowy jako zbiór domyślnych parametrów ustawionych przez pakiet z którego pochodzi dany model. Tabela przedstawia wartości tunnowalności poszczególnych parametrów dla każdego datasetu.

Model XGB

Dataset	Default params	Params rs	Params bayes	Tunability rs	Tunability bayes
Cellphone	0.979	0.976	0.977	0.003	0.004
Salary	0.825	0.876	0.877	0.051	0.052
Flight	0.845	0.848	0.849	0.003	0.004
Diamonds	0.972	0.974	0.974	0.002	0.002

Tabela 2: Tabela tunowalności dla modelu XGB

Model Elastic Net

Dataset	Default params	Params rs	Params bayes	Tunability rs	Tunability bayes
Cellphone	0.561	0.953	0.954	0.392	0.393
Salary	0.455	0.880	0.881	0.425	0.426
Flight	0.224	0.533	0.532	0.309	0.308
Diamonds	0.115	0.899	0.922	0.784	0.807

Tabela 3: Tabela tunowalności dla modelu Elastic Net

Model Random Forest

Dataset	Default params	Params rs	Params bayes	Tunability rs	Tunability bayes
Cellphone	0.952	0.956	0.955	0.004	0.003
Salary	0.846	0.854	0.854	0.008	0.008
Flight	0.840	0.854	0.854	0.014	0.014
Diamonds	0.825	0.856	0.974	0.031	0.149

Tabela 4: Tabela tunowalności dla modelu Random Forest

4 Wnioski

4.1 Tunowalność

Jak możemy zauważyć z tabel, model Elastic Net wyróżnia się największą tunowalnością. Nie jest to jednakże wynik jednoznaczny i łatwy do interpretowania, ponieważ ten sam model odznaczał się największymi skokami w R^2 w pierwszych iteracjach. Może to oznaczać, że analizowany model jest bardziej czuły na parametry z siatki i wymaga on tunowania, aby dopasować się do danych. Modele XGB i RF już w bazowych ustawieniach osiągają akceptowalne wyniki i można dojść do wniosku, że są bardziej uniwersalne i nie wymagają przesadnego tunowania. Podsumowując:

- Model Elastic Net - jest bardzo czuły w pierwszych iteracjach tunowania oraz najbardziej stabilny w późniejszych. Defaultowe parametry modelu nie sprawdzają się dla żadnego zbioru zatem można stwierdzić, że jest to model wymagający tunowania przynajmniej dla analizowanych parametrów.
- Model XGB - tylko w pojedynczych przypadkach reagował na tunowanie. Wykazuje on dobre przystosowanie do danych. Uśredniając miał najlepsze wyniki
- Random Forest - łączył w sobie zachowania modelu Elastic Net oraz XGB. Na ogół nie wymagał tunowania i nie reagował na nie znacznie, jednakże zdarzały się przypadki, gdzie tunowanie było konieczne, a nawet nie osiągał on pełnej stabilności w badanej liczbie iteracji.

4.2 Porównanie metod przeszukiwania siatki

Obie metody osiągnęły zbliżone wyniki z delikatną przewagą **Bayes Search**. Szybciej osiągał on stabilne wyniki jednakże jest on bardziej wymagający obliczeniowo. W przypadku **Random Search** nie został zaobserwowany bias sampling z racji, że wszystkie rozkłady w siatce parametrów były rozkładami jednostajnymi. W kontekście **Bayes Search**, algorytm ma tendencję do próbkowania z tych obszarów przestrzeni hiperparametrów, które wydają się najbardziej obiecujące na podstawie aktualnie dostępnych informacji zatem występuje tam bias sampling, gdyż dalsze iteracje prowadzą do eksploracji podobnych obszarów parametrów.

Bibliografia

[1] Anne-Laure Boulesteix Philipp Probst i Bernd Bischl. “Tunability: Importance of Hyperparameters of Machine Learning Algorithms”. W: *Journal of Machine Learning Research 20* (2019).