Wydział Matematyki
i Nauk Informacyjnych
POLITECHNIKA WARSZAWSKA

# Sprawdzenie tunowalności hiperparametrów

Sieci neuronowe

Jakub Borek
Karol Denst

15 listopada 2024

# 1 Wstęp

Celem pracy jest zbadanie tunowalności poszczególnych hiperparametrów. Ocena tunowalności będzie się bazować na [4]. Wybrane modele to

- Klasyfikator wektorów nośnych
- Losowy las decyzyjny
- Gradient Boosting

Modele i ich hiperparametry są opisane w sekcji 2. Dla każdego z modeli wykorzystujemy jego istniejącą implementację w pakiecie SKLearn. Modele były testowane na poniższych zbiorach danych z OpenML:

- Diabetes [5] - zawiera dane różnych osób wraz z informacją czy mają cukrzycę.
- Banknote Authentication [1] - zawiera dane opisujące zdjęcia prawdziwych i sfałszowanych banknotów.
- Credit [2] - zawiera dane różnych osób wraz z informacją czy są dobrymi kandydatami na kredyt.
- Spambase [3] - zawiera dane opisujące email oraz informację czy jest to spam czy nie.

# 2 Wybrane modele i ich hiperparametry

W tej sekcji opisane są modele oraz testowane hiperparametry.

## 2.1 Klasyfikator wektorów nośnych

Klasyfikator wektorów nośnych (SVC w pakiecie SKLearn) to model klasyfikacyjny bazujący na wektorach nośnych. Działa poprzez znajdowanie optymalnej granicy decyzyjnej, która najlepiej oddziela klasy w danych. Hiperparametry i ich przedziały:

- **C** - [0.01, 100] - parametr regularyzujący.
- **kernel** - {**linear, sigmoid, rbf**} - jądro algorytmu.
- **gamma** - [0.001, 100] - współczynnik ziarna. (nie ma znaczenia dla ziarna liniowego)

## 2.2 Losowy las decyzyjny

Jest to algorytm klasyfikacji oparty na ensemble, który tworzy wiele drzew decyzyjnych. Ostateczna decyzja jest wynikiem głosowania większościowego. Hiperparametry i ich przedziały:

- **n_estimators** - [1, 2000] - liczba drzew w lesie.
- **max_depth** - [3, 15] - maksymalna głębokość drzewa.
- **min_samples_split** - [2, 10] - minimalna liczba próbek wymagana do podziału węzła.
- **min_samples_leaf** - [1, 10] - minimalna liczba próbek w liściu.
- **max_features** - {**sqrt, log2, 0.2, 0.4**} - liczba cech branych pod uwagę przy każdym podziale.

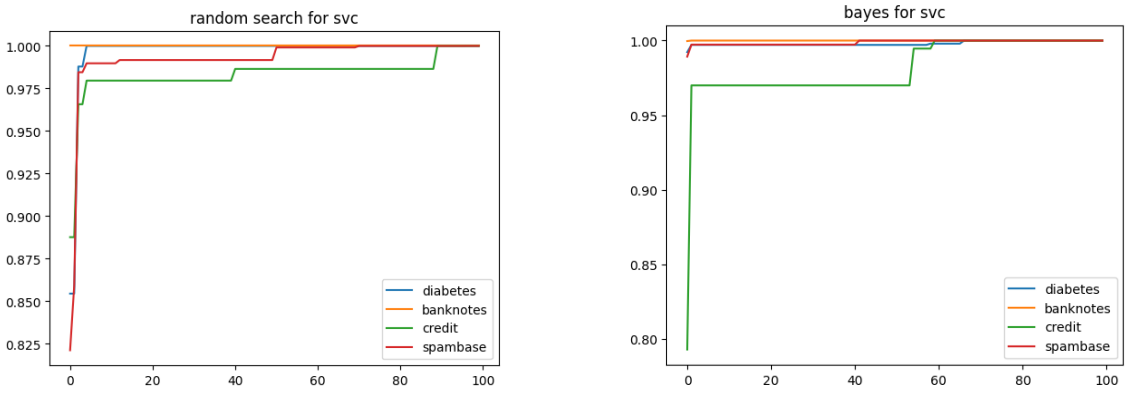
### 2.3 Gradient Boosting

Algorytm klasyfikacji, który buduje modele w sposób sekwencyjny, poprawiając błędy poprzednich modeli. Hiperparametry i ich przedziały:

- **n_estimators** - [1, 2000] - liczba drzew w lesie.
- **max_depth** - [3, 15] - maksymalna głębokość drzewa.
- **min_samples_split** - [2, 10] - minimalna liczba próbek wymagana do podziału węzła.
- **min_samples_leaf** - [1, 10] - minimalna liczba próbek w liściu.
- **max_features** - {**sqrt, log2, 0.2, 0.4**} - liczba cech branych pod uwagę przy każdym podziale.

## 3 Potrzebna liczba iteracji

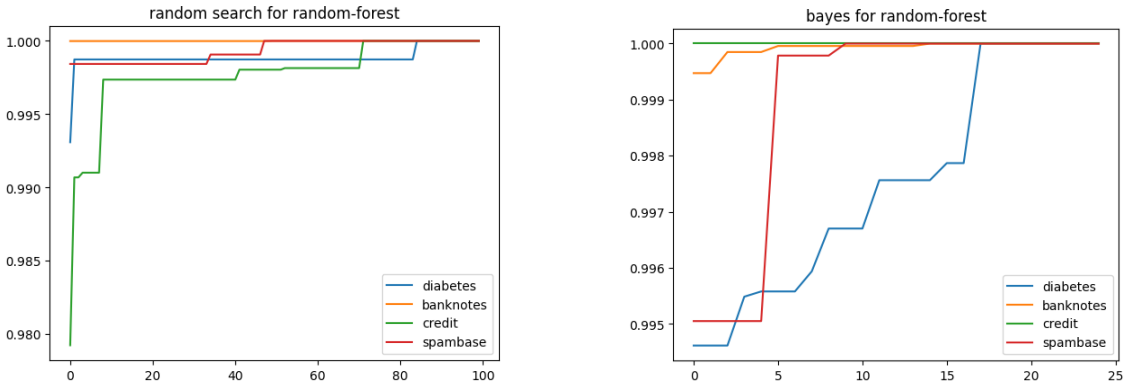
Różne algorytmy wymagają różnej liczby iteracji aby osiągnąć maksymalną efektywność. W tej sekcji przedstawione jest porównanie ile iteracji jest potrzeba do osiągnięcia najlepszej możliwej wydajności dla każdego algorytmu dla każdego zbioru danych.




(a) Wyniki dla optymalizacji losowej

(b) Wyniki dla optymalizacji Bayesowskiej

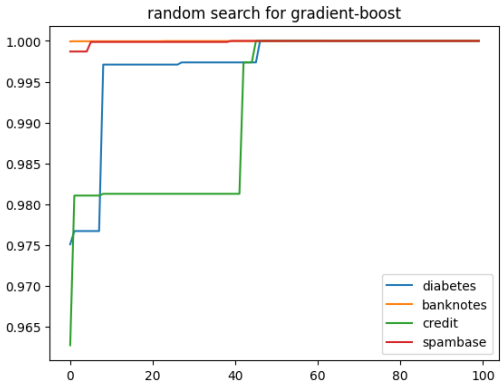
Rysunek 1: Wyniki dla Klasyfikator wektorów nośnych



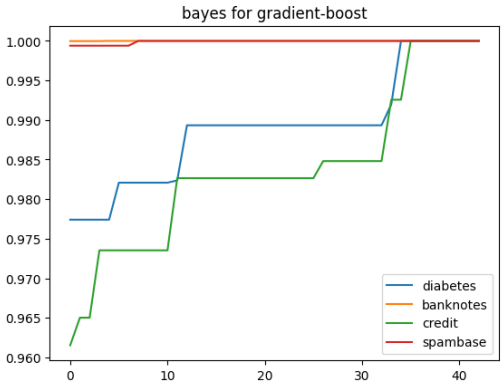

(a) Wyniki dla optymalizacji losowej

(b) Wyniki dla optymalizacji Bayesowskiej

Rysunek 2: Wyniki dla lasu losowego




(a) Wyniki dla optymalizacji losowej




(b) Wyniki dla optymalizacji Bayesowskiej

Rysunek 3: Wyniki dla Gradient Boostringu

# 4 Tunowalność algorytmów

Tunowalność hiperparametrów miarzy jak duże zmiany w wynikach algorytmu można osiągnąć poprzez zmiany hiperparametrów. Wzory mierzące tunowalność pochodzą z [4]. Tunowalność jest obliczana jako różnica pomiędzy wynikiem dla najlepszych parametrów, a wynikiem dla domyślnych parametrów SKLearn. Poprzez wynik rozumiemy $1-a$, gdzie $a$ to dokładność klasyfikacji na zbiorze testowym.

## 4.1 Wyniki dla optymalizacji losowej

| Algorytm | Zbiór danych | Tunowalność |
|---|---|---|
| Klasyfikator wektorów nośnych | Diabetes | 0.067 |
| Klasyfikator wektorów nośnych | Banknote Authentication | 0.0 |
| Klasyfikator wektorów nośnych | Credit | 0.045 |
| Klasyfikator wektorów nośnych | Spambase | 0.043 |
| Losowy las decyzyjny | Diabetes | 0.065 |
| Losowy las decyzyjny | Banknote Authentication | 0.005 |
| Losowy las decyzyjny | Credit | 0.017 |
| Losowy las decyzyjny | Spambase | 0.040 |
| Gradient Boosting | Diabetes | 0.076 |
| Gradient Boosting | Banknote Authentication | 0.005 |
| Gradient Boosting | Credit | 0.030 |
| Gradient Boosting | Spambase | 0.047 |

### 4.2 Wyniki dla optymalizacji bayesowskiej

| Algorytm | Zbiór danych | Tunowalność |
|---|---|---|
| Klasyfikator wektorów nośnych | Diabetes | 0.065 |
| Klasyfikator wektorów nośnych | Banknote Authentication | 0.0 |
| Klasyfikator wektorów nośnych | Credit | 0.047 |
| Klasyfikator wektorów nośnych | Spambase | 0.043 |
| Losowy las decyzyjny | Diabetes | 0.062 |
| Losowy las decyzyjny | Banknote Authentication | 0.005 |
| Losowy las decyzyjny | Credit | 0.028 |
| Losowy las decyzyjny | Spambase | 0.038 |
| Gradient Boosting | Diabetes | 0.073 |
| Gradient Boosting | Banknote Authentication | 0.005 |
| Gradient Boosting | Credit | 0.035 |
| Gradient Boosting | Spambase | 0.047 |

## 5 Podsumowanie

Celem pracy jest analiza tunowalności hiperparametrów trzech algorytmów klasyfikacyjnych: klasyfikatora wektorów nośnych, losowego lasu decyzyjnego oraz gradient boostingu. Modele te zostały zaimplementowane za pomocą pakietu SKLearn i przetestowane na zbiorach danych z OpenML: Diabetes, Banknote Authentication, Credit oraz Spambase. Tunowalność oceniono, porównując wyniki dla domyślnych i optymalnych wartości hiperparametrów, zgodnie z metodyką zaproponowaną w [4]. Przeprowadzone elementy pokazały, że jak bardzo tunowalny jest dany algorytm bardzo zależy od zbioru danych na podstawie którego przeprowadzamy optymalizację.

# Bibliografia


[1] Volker Lohweg. *Banknote Authentication*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C55P 2012.

[2] *OpenML Dataset: 31 (Credit Approval)*. https://www.openml.org/d/31. Accessed: 2024-11-13. 2023.

[3] *OpenML Dataset: 44 (Spambase)*. https://www.openml.org/d/44. Accessed: 2024-11-13. 2023.

[4] Philipp Probst, Bernd Bischl i Anne-Laure Boulesteix. *Tunability: Importance of Hyperparameters of Machine Learning Algorithms*. 2018. arXiv: 1802.09596 [stat.ML]. URL: https://arxiv.org/abs/1802.09596.

[5] Peter D. Turney. "Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm". W: *Journal of Artificial Intelligence Research* 2 (1995), s. 369–409. DOI: 10.1613/jair.120. URL: https://www.jair.org/index.php/jair/article/view/10129.