# Projekt 1

## Automatyczne uczenie maszynowe

Marta Balcerzak Michał Dębski Maciej Koczorowski

## Wprowadzenie

Celem projektu jest przeanalizowanie tunowalności trzech algorytmów uczenia maszynowego: regresji logistycznej z karą elastic net, k najbliższych sąsiadów oraz lasu losowego na sześciu zbiorach danych pochodzących z platformy OpenML: diabetes, pizza, climate, spambase, pc1 i credit o numerach id, odpowiednio, 37, 1444, 1467, 44, 1068 oraz 29. Pojęcie tunowalności algorytmu definiujemy w oparciu o artykuł *Tunability: Importance of Hyperparameters of Machine Learning Algorithms*:

**Definicja.** *Najlepszą konfiguracją hiperparametrów* dla zbioru danych $j$ nazwiemy

$$\theta^{(j)\star} := \arg\min_{\theta\in\Theta} R^{(j)}(\theta),$$

gdzie $\Theta$ to badana przestrzeń hiperparametrów, a $R^{(j)}(\theta)$ jest funkcją ryzyka przy konfiguracji hiperparametrów $\theta$ dla zbioru $j$.

**Definicja.** *Najlepszą konfiguracją hiperparametrów* dla zbiorów danych $\{1, ..., m\}$ nazwiemy

$$\theta^{\star} := \arg\min_{\theta\in\Theta} g\left(R^{(1)}(\theta), ..., R^{(m)}(\theta)\right),$$

gdzie $g$ jest pewną funkcją agregującą.

**Definicja.** *Tunowalność algorytmu* na zbiorze danych $j$ określamy jako różnicę:

$$d^{(j)} := R^{(j)}\big(\theta^{(j)\star}\big) - R^{(j)}\big(\theta^{\star}\big).$$

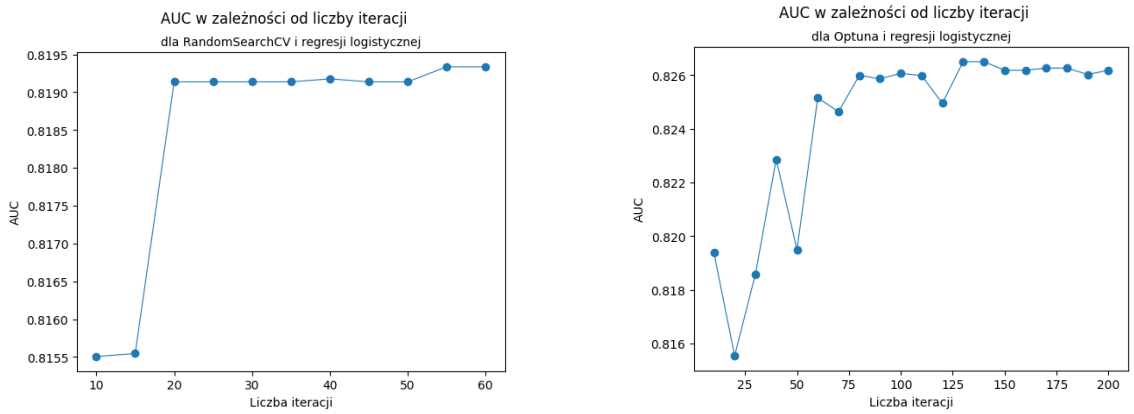
Powyższa definicja jest różna co do znaku od tej proponowanej w artykule. Jest ona dla nas wygodniejsza, ponieważ, w równoważny sposób, zamiast minimalizacji pewnej funkcji ryzyka, w projekcie będziemy maksymalizować miarę AUC, którą wybraliśmy ponieważ jest ona uniwersalna i odporna na dysproporcję klas.
Jako funkcję agregującą przyjmiemy średnią, w badaniach sprawdzaliśmy także, jak zachowa się $\theta^{\star}$, gdy przed obliczeniem średniej ustandaryzujemy otrzymane wartości $R^{(j)}(\theta)$ odejmując ich średnią oraz dzieląc przez odchylenie standardowe. Okazało się, że otrzymywane wyniki były takie same, więc pozostaliśmy przy średniej.
Podsumowując, zajmiemy się wyznaczeniem nowej domyślnej konfiguracji hiperparametrów (dla każdego z rozważanych algorytmów), która osiąga średnio najlepsze wartości AUC dla wszystkich zbiorów danych. Następnie przeanalizujemy różnice pomiędzy miarą otrzymaną dla takiej konfiguracji a miarą otrzymaną dla najlepszej konfiguracji, znalezionej dla każdego zbioru osobno, tzn. sprawdzimy tunowalność rozważanych algorytmów.
W badaniu wykorzystamy dwie techniki losowania punktów: Random Search z trzykrotną kroswalidacją oraz ustalonym ziarnem losowości, aby dla wszystkich zbiorów danych korzystać z tej samej siatki hiperparametrów dla każdego algorytmu, a także optymalizację bayesowską z wykorzystaniem biblioteki Optuna.

### Wstępne przetwarzanie danych

Zanim przejdziemy do kluczowej części projektu, poddamy nasze zbiory wstępnemu przetwarzaniu. Zmienne o silnie skośnym rozkładzie przekształcimy przy użyciu logarytmu. Braki danych numerycznych uzupełnimy średnią z pozostałych obserwacji, która, po przetestowaniu, okazuje się dawać lepsze wyniki niż użycie metody k najbliższych sąsiadów. W przypadku danych kategorycznych zastosujemy uzupełnianie najczęściej występującą wartością. Ponadto zmienne numeryczne przeskalujemy standardowym sposobem, tzn. odejmując średnią i skalując w celu uzyskania jednostkowej wariancji. Zmienne kategoryczne zostaną przekształcone za pomocą kodowania one-hot, tzn. zmiany każdej kategorii na oddzielną cechę binarną.

# 1 Ile iteracji?

Przed przystąpieniem do badania tunowalności algorytmów sprawdzimy, ile iteracji każdej metody potrzebujemy, aby uzyskać stabilne wyniki optymalizacji. Wyniki miary AUC dla każdej z technik losowania przy algorytmie elastic net w zależności od liczby iteracji przedstawiamy na rysunku 1. W przypadku przeszukiwania losowego zauważamy największy wzrost AUC przy liczbie iteracji równej 20. Podniesienie jej do 30 nie wpływa znacząco na czas wykonywania obliczeń, zatem w celu zachowania ostrożności wybieramy tę wartość. Przy optymalizacji bayesowskiej duży skok AUC występuje, gdy liczba iteracji wynosi 60. Kontynuacja zwiększania liczby iteracji przynosi niewielką poprawę, a czas działania algorytmu istotnie się zwiększa. Będziemy zatem korzystać z tej wartości podczas dalszych badań.




(a) Random Search

(b) optymalizacja bayesowska

Rysunek 1: AUC w zależności od liczby iteracji dla poszczególnych algorytmów

# 2 Zakresy hiperparametrów

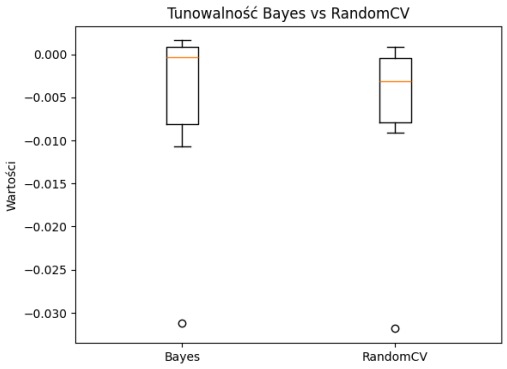
Hiperparametry badane podczas projektu dla każdego z algorytmów dobieramy zgodnie ze wspomnianym wcześniej artykułem. W tabeli 1 przedstawiamy wszystkie hiperparametry oraz ich zakresy, których używamy przy badaniu tunowalności algorytmów.

| Algorytm | Hiperparametr | Granica dolna | Granica górna |
|---|---|---|---|
| Elastic Net | l1_ratio | 0 | 1 |
| | C | $10^{-10}$ | $10^{10}$ |
| kNN | n_neighbors | 1 | 30 |
| | p | 1 | 3 |
| Random Forest | n_estimators | 1 | 2000 |
| | min_samples_leaf | 1 | 2000 |
| | max_features | 0.1 | 1 |
| | max_samples | 0.1 | 1 |
| | criterion | "gini", "entropy" | |

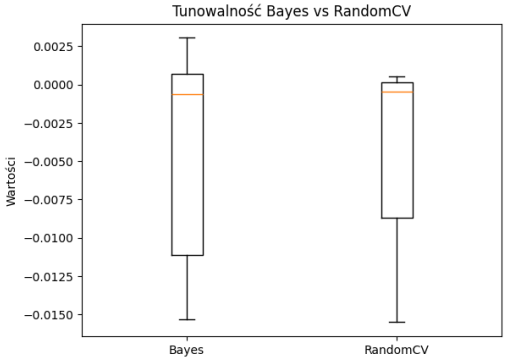
Tabela 1: Użyte zakresy hiperparametrów dla poszczególnych algorytmów

# 3 Tunowalność algorytmów

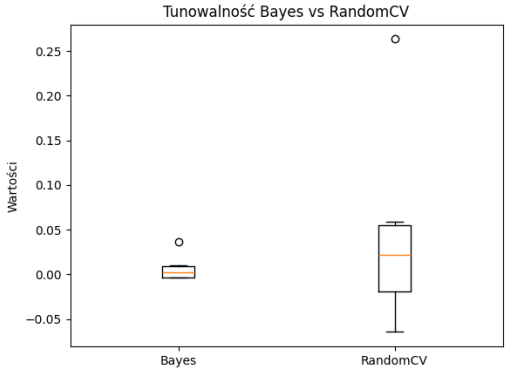
Na rysunkach 2 oraz 3 możemy przyjrzeć się wykresom skrzynkowym wyników tunowalności algorytmów na rozważanych zbiorach z wyróżnieniem techniki losowania. Dokładne wartości średnich i odchyleń standardowych wyników tunowalności umieszczamy w tabeli 2.




(a) Elastic Net

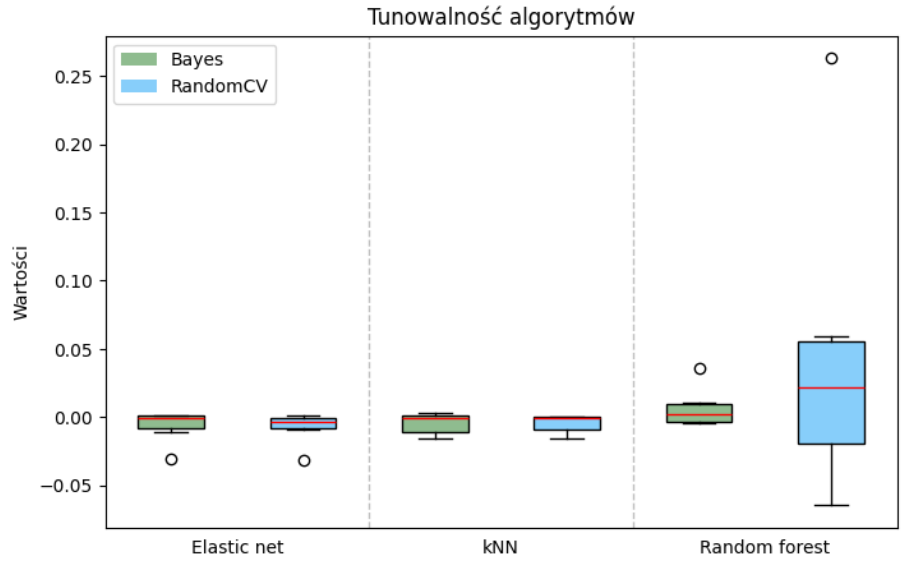



(b) kNN




(c) Random Forest

Rysunek 2: Tunowalność poszczególnych algorytmów z różnymi technikami losowania




Rysunek 3: Wykres zbiorczy tunowalności poszczególnych algorytmów

| Klasyfikator | **Elastic Net** | | **kNN** | | **Random Forest** | |
|---|---|---|---|---|---|---|
| | RandomCV | Bayes | RandomCV | Bayes | RandomCV | Bayes |
| Średnia | -0.0077 | -0.0066 | -0.0045 | -0.0045 | 0.0459 | 0.0070 |
| Odchylenie standardowe | 0.0113 | 0.0117 | 0.0064 | 0.0074 | 0.1056 | 0.0141 |

Tabela 2: Średnie i odchylenia standardowe tunowalności poszczególnych algorytmów

Możemy zauważyć, że wyniki oscylują w pobliżu zera. Algorytm lasu losowego z optymalizacją bayesowską zdecydowanie wyróżnia się pod względem rozrzutu. Widzimy także kilka wartości zakwalifikowanych jako odstające, może to oznaczać, że pewne zbiory są mniej lub bardziej podatne na tuning przy zastosowaniu konkretnych algorytmów. Początkowo mogą zaskakiwać wartości dodatnie pojawiające się na wykresach, wynika to z faktu, że najlepsze hiperparametry wybieraliśmy w oparciu o wyniki na zbiorach treningowych, natomiast wyniki końcowe miary AUC obliczyliśmy dla zbiorów testowych.

# 4 Porównanie różnic

W celu porównania różnic wyników pomiędzy technikami losowania hiperparametrów sprawdzamy, dla każdego algorytmu, czy rozkłady wyników tunowalności różnią się pomiędzy tymi technikami. Wykorzystamy do tego nieparametryczny test Wilcoxona. Dla algorytmów regresji logistycznej z karą elastic net, k najbliższych sąsiadów oraz lasu losowego p-wartość wynosi, odpowiednio, 0.3125, 1.0 oraz 0.5625. Wskazuje to na brak istotnych różnic w porównywanych rozkładach.