Mini-AutoML dla Danych Tabelarycznych

Hanna Szczerbińska Helena Wałachowska
Paula Wołkowska

Politechnika Warszawska
Wydział Matematyki i Nauk Informacyjnych
Warsztaty badawcze

29.01.2026

Cele projektu

Celem naszego projektu było:

1. stworzenie uproszczonego systemu AutoML, który umożliwiłby automatyczne wykonanie zadania klasyfikacji binarnej na dowolnym dostarczonym zbiorze danych,
2. skonstruowanie i wykorzystanie portfolio modeli.

Wybór modeli do portfolio

Faza 1 - grupa 1

- Pierwsze 50 konfiguracji obejmowało sześć typów modeli.
- Wybór konkretnych wartości hiperparametrów odbywał się poprzez losowanie zestawów hiperparametrów na podstawie ręcznie wybranych zakresów.
- Do wstępnego portfolio z każdego typu modelu wylosowano ustaloną liczbę konfiguracji.

Przestrzeń przeszukiwań dla grupy 1

Typ modelu	**Zakres wartości hiperparametrów**	**Liczba modeli**
RandomForestClassifier	$n\_estimators \in \{50, 100, 200, 500\}$ $max\_depth \in \{10, 20, \text{None}\}$ (stałe: $min\_samples\_split$=2, $random\_state$=42, n_jobs=−1)	10
XGBClassifier	$n\_estimators \in \{100, 200, 300\}$ $max\_depth \in \{5, 7, 10\}$ $learning\_rate \in \{0.05, 0.1\}$ (stałe: $random\_state$=42, n_jobs=−1, $eval\_metric$="logloss")	10
LGBMClassifier	$n\_estimators \in \{100, 200\}$ $num\_leaves \in \{31, 63, 127\}$ $learning\_rate \in \{0.05, 0.1\}$ (stałe: $random\_state$=42, n_jobs=−1, $verbose$=−1)	10
CatBoostClassifier	$iterations \in \{100, 200\}$ $depth \in \{6, 8\}$ $learning\_rate \in \{0.05, 0.1\}$ (stałe: $random\_state$=42, $verbose$=False, $allow\_writing\_files$=False)	8
SVC	$C \in \{1.0, 10.0\}$ $kernel \in \{\text{"rbf"}, \text{"poly"}\}$ $\gamma \in \{\text{"scale"}, \text{"auto"}\}$ (stałe: $random\_state$=42, $probability$=True)	6
ExtraTreesClassifier	$n\_estimators \in \{50, 100, 200\}$ $max\_depth \in \{10, \text{None}\}$ (stałe: $random\_state$=42, n_jobs=−1)	6
	Suma	**50**

Faza 1 - grupa 2

- Utworzono 28 modeli wraz z odpowiadającymi im konfiguracjami hiperparametrów.
- Procedura doboru wartości hiperparametrów była oparta na algorytmie Random Search o 15 iteracjach z trójkrotną walidacją krzyżową. D oceny jakości predykcji użyto balanced accuracy.
- Wybór hiperparametrów modeli przeprowadzono na podstawie dziesięciu zbiorów danych pochodzących z OpenML o identyfikatorach: 31, 1464, 334, 50, 1504, 3, 1494, 1510, 1489 oraz 37.
- Procedura kalibracji modeli polegała na losowym wyborze czterech zbiorów danych dla każdego typu modelu oraz optymalizacji hiperparametrów dla każdego zbioru danych w oparciu o zdefiniowane zakresy wartości.

Przestrzeń przeszukiwań dla grupy 2

Typ modelu	**Hiperparametr**	**Rozkład / zbiór wartości**
SVM	C	uniform$(0.01, 100)$
	$kernel$	{rbf, linear, poly}
	γ	{scale, auto} $\cup$ uniform$(0.001, 1)$
kNN	$n\_neighbors$	randint$(3, 20)$
	$weights$	{uniform, distance}
	$metric$	{euclidean, manhattan, minkowski}
	p	randint$(1, 3)$
Naive Bayes	$var\_smoothing$	uniform$(10^{-12}, 10^{-6})$
Gradient Boosting	$n\_estimators$	randint$(50, 300)$
	max_depth	randint$(3, 10)$
	$learning\_rate$	uniform$(0.01, 0.3)$
	$min\_samples\_split$	randint$(2, 20)$
	$min\_samples\_leaf$	randint$(1, 10)$
CatBoost	$n\_estimators$	randint$(50, 300)$
	max_depth	randint$(3, 10)$
	$learning\_rate$	uniform$(0.01, 0.3)$
	$min\_data\_in\_leaf$	randint$(1, 20)$
LightGBM	$n\_estimators$	randint$(50, 300)$
	max_depth	randint$(3, 15)$
	$learning\_rate$	uniform$(0.01, 0.3)$
	$min\_child\_samples$	randint$(5, 50)$
	$subsample$	uniform$(0.6, 1.0)$
AdaBoost	$n\_estimators$	randint$(50, 300)$
	$learning\_rate$	uniform$(0.1, 1.5)$

Faza 1 - grupa 3

- Dokonano ręcznego wyboru 22 modeli wraz z odpowiadającymi im konfiguracjami hiperparametrów, na podstawie wyników zewnętrznych eksperymentów opublikowanych na platformie OpenML.
- W szczególności uwzględniono wielowarstwowe sieci neuronowe MLP o architekturach charakteryzujących się malejącą liczbą neuronów w kolejnych warstwach, klasyczne klasyfikatory liniowe i nieliniowe (SVM, regresję logistyczną oraz klasyfikator kNN), a także metody zespołowe (Random Forest, AdaBoost oraz XGBoost) w kilku wariantach parametrów.

Faza 2

- Spośród wstępnie wybranych 100 modeli wyselekcjonowano 50 najlepszych konfiguracji.
- Selekcja została przeprowadzona na podstawie ewaluacji wszystkich 100 modeli na 15 zbiorach danych (5 małych, 5 średnich oraz 5 dużych), losowo wybranych z OpenML.
- Każdy model generował predykcje dla wszystkich 15 zbiorów danych, które były ocenianie za pomocą balanced accuracy. Na tej podstwie dla każdego zbioru danych utworzono ranking modeli.
- Dla każdego modelu obliczono średnią pozycję w rankingach uzyskanych dla wszystkich zbiorów danych.
- Do ostatecznego portfolio zakwalifikowano konfiguracje o najwyższej średniej pozycji rankingowej.

Metoda selekcji modeli z portfolio dla nowego zbioru danych

Działanie metody fit - etap 1

- W zależności od liczby obserwacji w zbiorze treningowym wybierana jest liczba foldów walidacji krzyżowej: pięć dla mniejszych zbiorów danych (<10000) oraz trzy dla większych. (zachowujemy proporcję klas)
- Pierwszym etapem selekcji modeli jest szybki screening, w którym każdy model z portfolio trenowany jest na części danych treningowych (80%), a następnie oceniany na wydzielonym zbiorze walidacyjnym (20%) przy użyciu miary ROC-AUC obliczanej na podstawie ciągłych wyników predykcji (prawdopodobieństw lub wartości funkcji decyzyjnej).
- Spośród wszystkich modeli do kolejnego etapu przekazywana jest jedynie ustalona liczba najlepszych konfiguracji (domyślnie 15).

Działanie metody fit - etap 2

- W drugim etapie przeprowadzana jest pełna ewaluacja wyselekcjonowanych modeli z wykorzystaniem walidacji krzyżowej. Modele oceniane są do momentu wyczerpania budżetu czasowego.
- Jeżeli ensembling nie jest aktywny, wybierany jest pojedynczy model o najwyższej średniej wartości ROC-AUC. Model ten jest następnie trenowany ponownie na pełnym zbiorze treningowym.
- Na podstawie predykcji out-of-fold wyznaczany jest optymalny próg decyzyjny maksymalizujący wartość miary balanced accuracy, który zastępuje domyślny próg 0.5

Esembling

Esembling

- Do ensemble mogą wejść maksymalnie pięć modeli, przy czym wprowadzono dodatkowe ograniczenie różnorodności polegające na tym, że z każdej rodziny algorytmów (np. modele drzewiaste, boostingowe, SVM) mogą zostać wybrane co najwyżej dwa modele.
- Po zakończeniu selekcji wszystkie modele wchodzące w skład ensemble są trenowane ponownie na pełnym zbiorze treningowym. Predykcje out-of-fold poszczególnych modeli są następnie uśredniane, a na ich podstawie wyznaczany jest wspólny próg decyzyjny maksymalizujący wartość miary balanced accuracy. W fazie predykcji ensemble stosuje miękkie głosowanie (soft voting), polegające na uśrednianiu prawdopodobieństw generowanych przez poszczególne modele, a następnie porównaniu uzyskanej wartości z nauczonym progiem decyzyjnym.

Wnioski

Eksperyment ewaluacyjny

- Jakość predykcji możliwych do osiągnięcia przez stworzony system Mini-AutoML została oceniona na podstawie pięciu zbiorów testowych pochodzących z platformy OpenML oraz zbioru testowego X zaproponowanego w ramach zajęć.
- Eksperymenty przeprowadzono przy ograniczeniu czasowym wynoszącym 20 minut oraz przy pozostałych parametrach pozostawionych na wartościach domyślnych.
- Jakość predykcji uzyskanych na zbiorach danych pochodzących z platformy OpenML okazała się zadowalająca, osiągając wartości miary balanced accuracy bliskie 100% dla trzech z nich. Uzyskane wyniki są zgodne z rezultatami raportowanymi w literaturze oraz w niezależnych eksperymentach dostępnych dla tych zbiorów danych.

Wyniki eksperymentu

Zbiór danych	Tryb	Wybrany model / skład ensemble	Balanced Accuracy	Zakres prawdopodobieństw	Czas dopasowania [s]
Zbiór testowy X	Ensemble	ExtraTrees (2), RandomForest (2), SVC (1)	0.5586	$[0.2952, 0.7586]$	128.51
	Bez ensemble	ExtraTreesClassifier	0.5489	$[0.2500, 0.8050]$	137.03
credit-g (OpenML)	Ensemble	RandomForest (3), CatBoost (2)	0.7405	$[0.2052, 0.9777]$	330.54
	Bez ensemble	CatBoostClassifier	0.7333	$[0.1355, 0.9780]$	477.29
blood-transfusion-service-center (OpenML)	Ensemble	CatBoost (4), MLP (1)	0.6949	$[0.0440, 0.7367]$	32.96
	Bez ensemble	MLPClassifier	0.7027	$[0.0107, 0.6830]$	40.27
wdbc (OpenML)	Ensemble	SVC (1), ExtraTrees (1), CatBoost (1), XGBoost (2)	0.9453	$[0.0004, 0.9996]$	189.87
	Bez ensemble	SVC	0.9688	$[0.0001, 1.0000]$	106.56
monks-problems-2 (OpenML)	Ensemble	MLP (1), GradientBoosting (1), LightGBM (3)	0.9832	$[0.0000, 0.9899]$	108.09
	Bez ensemble	MLPClassifier	1.0000	$[0.0000, 0.9977]$	161.80
PhishingWebsites (OpenML)	Ensemble	GradientBoosting (1), LightGBM (4)	0.9722	$[0.0000, 1.0000]$	281.24
	Bez ensemble	LGBMClassifier	0.9733	$[0.0000, 1.0000]$	256.99