Raport o Mini-AutoML dla Danych Tabelarycznych

Elissa Hallak, Liwia Jankowska, Julia Tomaszkiewicz

Styczeń 2026

Wstęp

Celem projektu jest opracowanie uproszczonego systemu AutoML do klasyfikacji binarnej na dowolnym zbiorze danych tabelarycznych. System `MiniAutoML` umożliwia automatyczny wybór najlepszego modelu lub zestawu modeli z portfolio oraz wykorzystanie ensemblingu w celu poprawy dokładności i stabilności predykcji.

Etap 1: Wstępna selekcja modeli do portfolio

Podejście do wyboru modeli (Opcja B)

Przy doborze modeli do naszego portfolio zdecydowałyśmy się na screening na podstawie wyników zewnętrznych. Zamiast przeprowadzać czasochłonne eksperymenty, oparłyśmy się na wynikach z ogólnodostępnych źródeł i benchmarków.
Przeglądając literaturę [1, 3] zauważyłyśmy wyraźny trend: w przypadku typowych danych tabelarycznych algorytmy oparte na drzewach decyzyjnych sprawdzają się lepiej niż sieci neuronowe. Dlatego to właśnie na nich postanowiłyśmy oprzeć naszą strategię, oszczędzając czas obliczeniowy przy zachowaniu wysokiej skuteczności.

Skład naszego portfolio

Wykorzystując maksymalny dostępny limit, przygotowałyśmy plik `models.json` zawierający dokładnie 50 konfiguracji. Naszą strategią była maksymalizacja różnorodności modeli, co jest fundamentem skutecznego ensemblingu.
Nasze portfolio to:

- **Gradient Boosting**
  Główną grupę modeli stanowią **LightGBM (16)**, **XGBoost (7)** oraz **CatBoost (8)**. Są to obecnie najskuteczniejsze algorytmy dla danych tabelarycznych [2, 4]. Zróżnicowałyśmy je pod kątem głębokości drzew i tempa uczenia, aby wyłapać zarówno proste, jak i bardzo złożone zależności.
- **Drzewa Losowe**
  Oprócz klasycznego **Random Forest (8)**, dodałyśmy też algorytm **Extra Trees (2)**. Modele te wprowadzają większą losowość przy podziałach węzłów, co w połączeniu z boostingiem redukuje wariancję błędu.
- **Modele Geometryczne i Liniowe**
  Dla dopełnienia dodałyśmy:
  - **SVM (4):** Modelujące nieliniowe granice za pomocą jądra RBF.
  - **k-NN (2):** Algorytmy, które patrzą na lokalne sąsiedztwo punktów, a nie na globalne reguły.
  - **Regresję Logistyczną (3):** Jako stabilny punkt odniesienia (*baseline*).

Wszystkie modele zostały skonfigurowane tak, aby radzić sobie z nierównowagą klas (class_weight='balanced').

Etap 3: Implementacja Systemu Mini-AutoML

Celem etapu było opracowanie uproszczonego systemu AutoML do automatycznej klasyfikacji binarnej na danych tabelarycznych. Stworzono klasę `MiniAutoML` z trzema podstawowymi metodami: `fit`, `predict` oraz `predict_proba`.

Struktura klasy MiniAutoML

- **`__init__`**: przyjmuje konfigurację modeli z pliku JSON, opcjonalny parametr `max_models` (maksymalna liczba modeli do sprawdzenia) oraz `seed` do zapewnienia powtarzalności wyników.
- **fit**: wybiera i trenuje najlepsze modele na pełnym zbiorze treningowym. Procedura obejmuje:
  1. Podział cech na numeryczne i kategoryczne.
  2. Budowę pipeline z preprocessingiem (imputacja braków, skalowanie, one-hot encoding) oraz modelem klasyfikacyjnym.
  3. Walidację krzyżową (*StratifiedKFold*, 5-krotna) i obliczenie średniego *Balanced Accuracy*.
  4. Ranking modeli i wybór Top-5 do potencjalnego ensemblingu.
  5. Trenowanie wybranych modeli na całym zbiorze treningowym.
- **predict_proba**: zwraca prawdopodobieństwo przynależności do klasy pozytywnej; w przypadku ensemble stosowane jest uśrednianie wyników Top-5 modeli (soft voting).
- **predict**: zwraca etykiety klas binarnych.

Selekcja modeli

System wybiera najlepsze modele w portfolio na podstawie wyników walidacji krzyżowej. W eksperymentach do ensemblingu wybrano maksymalnie pięć najlepszych modeli.

Ensembling

Wykorzystano *soft voting*, czyli uśrednianie prawdopodobieństw przewidywanych przez Top-5 modeli. Końcowe predykcje systemu `MiniAutoML` są obliczane w ten sposób. Podejście to pozwala:

- zwiększyć dokładność w porównaniu z najlepszym pojedynczym modelem,
- zmniejszyć wariancję wyników przy różnych podziałach danych,
- zapewnić większą stabilność systemu.

Etap 4: Ocena i ewaluacja

W etapie oceny przetestowano system `MiniAutoML` na trzech zbiorach danych: `raisins`, `med` oraz `income`, aby sprawdzić skuteczność klasyfikatorów oraz powtarzalność wyników przy różnych ziarnach losowości.

Krótki opis zbiorów danych

- **raisins** [5]: zmienna celu `Class` określa rodzaj rodzynka (Kecimen/Besni); 900 próbek z 7 cechami wyekstrahowanymi z obrazów.
- **med** [7]: zmienna celu `mort30` wskazuje śmierć w ciągu 30 dni po zabiegu; dane pacjentów do klasyfikacji ryzyka zgonu.
- **income** [6]: zmienna celu `income_>50K` wskazuje, czy dochód przekracza 50 tys. dolarów; cechy demograficzne i zawodowe.

Dla każdego zbioru danych przeprowadzono podział na zbiory treningowe i testowe. Na zbiorach treningowych trenowano system `MiniAutoML`, po czym wyciągano ranking Top-5 modeli według średniej *Balanced Accuracy* uzyskanej w walidacji krzyżowej oraz obliczano wyniki predykcji na zbiorze testowym. Proces ten powtarzano wielokrotnie zarówno dla pojedynczego ziarna, jak i dla różnych ziaren losowości, aby ocenić powtarzalność i stabilność wyników.

Wyniki

- Ranking Top-5 modeli jest stabilny dla tego samego ziarna losowości – każde uruchomienie generuje identyczny zestaw modeli oraz te same wartości *Balanced Accuracy*. Z tego względu w tabeli **??** zaprezentowano jedynie pojedynczy zestaw wyników dla każdego zbioru danych, ponieważ wyniki powtarzały się przy każdym uruchomieniu.
- Różne zbiory preferują różne modele, co uzasadnia automatyczną selekcję modeli zamiast ręcznego wyboru jednego algorytmu.
- Wyniki najlepszego modelu (best CV) i ensemble są zbliżone. Nie zauważamy jednoznaczej poprawy wyników przy użyciu modelu ensemble.

Table 1: Top-5 modeli dla trzech zbiorów danych przy ziarnie 42

Dataset	Seed	Model	Balanced Accuracy	Rank
5*raisins	42	cat_150_deep	0.869	1
		logreg_l2_strong	0.868	2
		xgb_200_subsample	0.868	3
		svc_rbf_tight	0.868	4
		cat_500_very_slow	0.867	5
5*med	42	svc_linear	0.828	1
		logreg_l2_standard	0.823	2
		logreg_l1_selection	0.823	3
		logreg_l2_strong	0.822	4
		lgbm_300_dart_slow	0.774	5
5*income	42	lgbm_100_feature_fraction	0.844	1
		lgbm_100_leaves15	0.844	2
		lgbm_100_standard	0.844	3
		lgbm_200_reg_l1	0.843	4
		lgbm_200_reg_l2	0.843	5

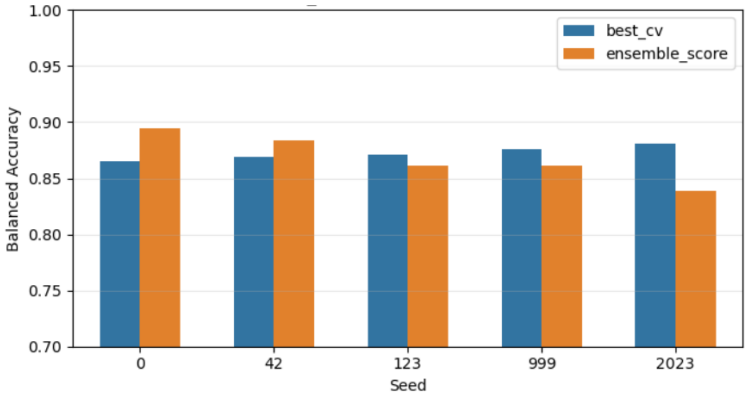

Figure 1: Porównanie najlepszego CV i wyniku ensemble dla zbioru raisins dla różych ziaren

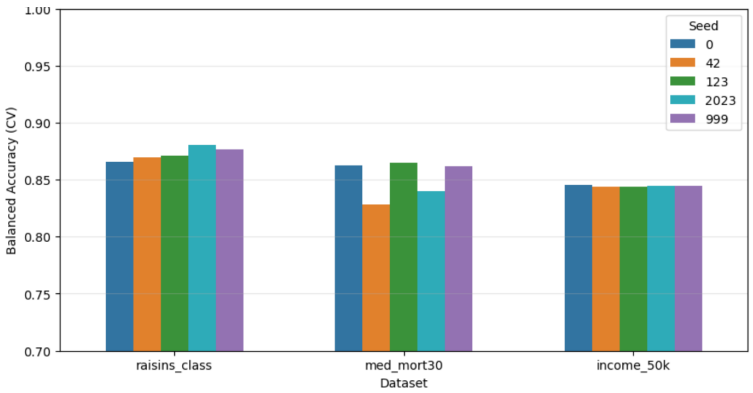

Figure 2: Najlepszy wynik CV dla każdego datasetu przy różnych ziarnach

Wnioski

Na podstawie przeprowadzonych eksperymentów można stwierdzić, że:

- System `MiniAutoML` skutecznie wybiera najlepsze modele dla różnych zbiorów danych binarnych, co potwierdzają wyniki walidacji krzyżowej.
- Zmiana ziaren losowości nie wpływa istotnie na jakość predykcji, co świadczy o stabilności systemu.
- Ensembling nie daje jednoznacznych wyników (polepszenie/pogorszenie balanced accuracy) w porównaniu do najlepszego pojedynczego modelu.
- Podejście jest uniwersalne i może być zastosowane do innych zbiorów danych binarnych, zachowując wysoką jakość predykcji i powtarzalność eksperymentów.
- Sprawdzanie różnych ziaren pokazuje, że faworytami danego zbioru danych są często modele z jednej grupy (*czest_wyboru_modeli.png*).

References

[1] B. Bischl et al.,
OpenML-CC18 Curated Classification benchmark

[2] L. Grinsztajn, E. Oyallon, G. Varoquaux,
Why do tree-based models still outperform deep learning on typical tabular data?

[3] R. Shwartz-Ziv, A. Armon,
Tabular data: Deep learning is not all you need

[4] W. Kretowicz, P. Biecek,
MementoML: Performance of selected machine learning algorithm configurations on OpenML100 datasets

[5] Raisin binary classification. Kaggle. `https://www.kaggle.com/datasets/nimapourmoradi/raisin-binary-classification`

[6] Income Dataset. Kaggle. `https://www.kaggle.com/datasets/mastmustu/income?select=train.csv`

[7] Dataset Surgical binary classification. Kaggle. `https://www.kaggle.com/datasets/omnamahshivai/surgical-dataset-binary-classification`