AutoML 2023 Praca Domowa 2

Wiktor Jakubowski, Mikołaj Piórczyński

1 Wstęp

Poniższy dokument jest raportem z eksperymentów przeprowadzonych w ramach pracy domowej nr. 2 w ramach przedmiotu Automatyczne Uczenie Maszynowe na Wydziale Matematyki i Nauk Informacyjnych Politechniki Warszawskiej w semestrze jesiennym 2023 roku. W ramach zadania należało przygotować modele dla zadania klasyfikacji binarnej o jak największej mocy predykcyjnej dla sztucznie wygenerowanych i dostarczonych na potrzeby zadania danych, w których zostały ukryte istotne zmienne. Modele należało przygotować w dwóch wariantach:

- ręcznie, czyli wybrać rodzaj modelu, hiperparametry, etc.
- wykorzystując dostępne frameworki AutoMLowe

Dane zawierały 500 zmiennych objaśniających oraz 2000 obserwacji w zbiorze treningowym i 600 w zbiorze testowym. We wszystkich eksperymetach dokładność modeli była mierzona za pomocą miary zrównoważonej dokładności balanced accuracy zdefiniowanej jako:

$$\text{BALANCED ACCURACY} = \frac{1}{2}\left(\frac{TP}{P} + \frac{TN}{N}\right)$$

2 Eksperymenty

Ze zbioru treningowego zostało wydzielonych 20% danych do utworzeni dodatkowego zbioru walidacyjnego wspólnego dla wszystkich eksperymentów.

2.1 ClassicML

W eksperymentach zdecydowaliśmy się na wybór modelu `XGBoost` [1] ze względu na jego dominującą jakość w stosunku do innych modeli na większości zbiorach danych. Dokonano następującego preprocessingu danych:

- usunięto kolumny zawierające pojedynczą wartość dla wszystkich danych treningowych
- dokonano imputacji brakujących danych wartością średnią w kolumnie
- dokonano standaryzacji zmiennych
- dokonano usunięcia outlierów za pomocą modelu `IsolationForest`

[1] `https://xgboost.readthedocs.io/en/stable/`

- usunięto silnie skorelowane liniowo zmienne, dla których wartość współczynnika korelacji liniowej Pearsona przekraczała $\rho = 0.9$
- dokonano selekcji zmiennych na podstawie `feature importance` zmiennych uzyskanych z modelu drzewa decyzyjnego wytrenowanego na zbiorze treningowym

Przetestowano zarówno algorytm z domyślnymi hiperparametrami jak również dokonano ich optymalizacji z pomocą algorytmu zaimplementowanego w pakiecie `Optuna`[2] dla budżetu 100 iteracji. Przy optymalizacji hiperparametrów zastosowano heurystykę balansującą BALANCED ACCURACY na zbiorze treningowym i zbiorze walidacyjnym mającą na celu swoistą regularyzację modelu i zapobieżenie przeuczeniu. Dokonano również analizy wpływu liczby zmiennych w modelu na jego moc predykcyjną.

Table 1: BALANCED ACCURACY na zbiorach treningowym i walidacyjnym dla modeli wytrenowanych 'ręcznie'.

Features	**HPO**	**Train**	**Validation**
10	-	0.9469	0.8479
20	-	0.9544	0.8351
50	-	0.9800	0.8254
100	-	0.9869	0.8128
10	+	0.9287	0.8296
20	+	0.9094	0.8130
50	+	0.8950	0.8402
100	+	0.9012	0.8206

Widzimy, że w przypadku domyślnych hiperparametrów algorytm miał wyraźne problemy z przeuczeniem na zbiorze treningowy, w przypadku zastosowania optymalizacji hiperparametrów wraz z dodaną heurystyką zjawisko to uległo nieznacznemu zmniejszeniu. Jako finalny algorytm wybrano model z 50 zmiennymi oraz hiperparaetrami dostrojonymi poprzez optymalizację.

2.2 AutoML

W eksperymentach zostały wykorzystane trzy wiodące pakiety AutoML:

- AutoGluon[3],
- TPOT[4],
- H2O[5],

Powyższe pakiety zostały poddane treningowi na dostarczonych danych treningowych i walidacyjnych. Każdy z modeli był trenowany przez 15 minut (mała ilość czasu wynika z mikroskopijnego rozmiaru danych wejściowych) w dwóch podejściach:

[2] `https://optuna.readthedocs.io/en/stable/`
[3] `https://auto.gluon.ai/`
[4] `https://epistasislab.github.io/tpot/`
[5] `https://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html`

1. Preprocessing danych ograniczony do imputacji brakujących danych i zmapowania zmiennej celu na wartości '0'/'1',
2. Preprocessing danych rozszerzony o skalowanie cech, redukcję wymiarowości, i usunięcie silnie skorelowanych kolumn

Imputacja brakujących danych została wykonana poprzez uzupełnienie średnią z każdej kolumny – każda cecha jest numeryczna.

Skalowanie cech zostało przeprowadzone przy pomocy `Standard Scaler`, który standaryzuje każdą wartość poprzez odjęcie średniej z kolumny i podzieleniu wyniku przez odchylenie standardowe.

Redukcja wymiarowości została zaaplikowana poprzez zastosowanie Analizę Głównych Składowych (Principal Component Analysis – PCA). Jako próg wyjaśnialnej wariancji przyjęto poziom 0.8.

Selekcję cech oparto o wartości współczynnika korelacji Pearsona; gdy jego wartość przekraczałą 0.8, cechę o mniejszej wariancji usuwano z rozważań.

Wyniki eksperymentów zostały przedstawione w tabeli 2.

Table 2: Balanced accuracy na zbiorach treningowym i walidacyjnym dla pakietów AutoML.

Pakiet AutoML	**Preprocessing**	**Train**	**Validation**	**5% test**
AutoGluon	+	1	0.73	0.80
TPOT	+	0.99	0.66	
H2O	+	1	0.68	
AutoGluon	**-**	**0.87**	**0.86**	**0.90**
TPOT	-	0.92	0.84	
H2O	-	1	0.84	

Tak jak z powyższej tabeli wynika, najlepsze wyniki na zbiorze walidacyjnym osiągnął pakiet AutoGluon bez zaawansowanego preprocessingu. Model, wytypowany przez ten pakiet za najlepszy to `Weighted Ensemble` z regularyzacją L2. Na komitet składają się następujące modele z odpowiednimi wagami:

- `CatBoost` - 0.619
- `LightGBM` - 0.286
- `LightGBMXT` - 0.095

Oprócz tego modelu, wszystkie pozostałe cechują się dość sporym przeuczeniem, co odwzorowuje spora różnica w wartościach metryki na zbiorze treningowym i walidacyjnym. Warty zauważenia jest również fakt, że dodatkowe transformacje i filtrowanie danych znacznie obniżyły osiągane wyniki przez modele.

3 Podsumowanie

Jak wynika z powyższych rozważań, zarówno modele zmodyfikowane ręcznie, jak i pakiety do automatycznego uczenia maszynowego osiągnęły wysokie wyniki na zadanych zbiorach. To powiedziawszy, pakiety AutoML poradziły sobie z zadaniem nieco lepiej, przy dużo mniejszym czasie

trenowania i braku potrzeby zaawansowanego preprocessingu i inżynierii cech. Na zaznaczenie zasługuje też fakt, że eksperymenty z przeprocesowaniem danych przed zaaplikowaniem na nich modelu AutoML nie powiodły się, jako że predykcje w ten sposób uzyskane okazały się znacznie gorsze od tych wykonywanych na nietransformowanych danych.