Praca domowa 2 - AutoML

Hubert Bujakowski
Mikołaj Gałkowski

Styczeń 2024

Streszczenie

Praca koncentruje się na opracowaniu skutecznej metody klasyfikacji dla sztucznie wygenerowanego zbioru danych "artificial". Celem jest stworzenie modelu o maksymalnej mocy predykcyjnej, klasyfikującego obserwacje do dwóch klas. Ocena dokładności modelu będzie oparta na zrównoważonej dokładności. Badania obejmą różne algorytmy klasyfikacyjne i techniki przetwarzania danych w celu optymalizacji wyników.

1 Wstęp

Celem projektu jest opracowanie metody klasyfikacji, która umożliwi zbudowanie modelu o maksymalnej mocy predykcyjnej. W tym celu zostaną przygotowane dwa warianty modelu: ręcznie oraz z wykorzystaniem frameworków AutoMLowych. Jakość modeli będzie oceniana za pomocą zrównoważonej dokładności (balanced accuracy).

2 Zbiór Danych

Dane do projektu to sztucznie wygenerowany zbiór, który obejmuje 500 zmiennych objaśniających. Zbiór treningowy zawiera 2000 obserwacji, a zbiór testowy 600. Nie zawiera on braków danych.

3 Podejście Ręczne

W tym wariantu modelu dane zostały przygotowane ręcznie. Zbiór zawiera 500 zmiennych objaśniających, a zmienna wyjaśniana przyjmuje wartości -1, 1.

Kroki preprocessingu, które wykonaliśmy to:

- usunięcie silnie skorelowanych kolumn (`corr(kolumna_A, kolumna_B) > 0.75`
- skalowanie przy użyciu `MinMaxScaler'a`
- redukcja mało wpływowych kolumn poprzez `feature importance` z `Random Forest'a` – użyliśmy do tego klasy `SelectFromModel` i ustawiliśmy próg wyboru na 90% średniej ważności cech [`threshold='0.9*mean'`]

3.1 Opis algorytmów

Algorytmy, które wzieliśmy pod uwage w naszych eksperymentach:

- `CatBoost`
- `XGBoost`
- `SVM`
- `Naive Bayes`
- `Logistic Regression`
- `KNN`

Powyższe algorytmy poddaliśmy treningowi hiperparametrów, zarówno metodą `random_search` jak i `grid_search` przy użyciu 10-krotnej kros-walidacji.

3.2 Wyniki Eksperymentów

Jakość predykcji modelu ręcznego mierzona poprzez `best_score_` z danego search'a + `train_balanced_acc` dla każdego modelu o najlepszych hiperparametrach wytrenowanego na całym zbiorze treningowym.

Run Name	Created	Duration	best_score	train_balanced_acc	search_type
CatBoost_2024-01-14_20-59-36	44 minutes ago	25.4min	0.8595	1	RANDOM
XGBoost_2024-01-14_19-51-49	1 hour ago	8.9min	0.8335000000...	0.9935	GRID
XGBoost_2024-01-14_18-55-57	2 hours ago	23.7min	0.8315543494...	1	RANDOM
SVM_2024-01-14_20-01-41	1 hour ago	10.9min	0.6019999999...	0.757	GRID
LogisticRegression_2024-01-14_20-20-24	1 hour ago	9.3s	0.5965	0.668500000000...	-
KNN_2024-01-14_20-21-28	1 hour ago	34.8min	0.5745	0.6845	GRID
KNN_2024-01-14_20-20-47	1 hour ago	31.7s	0.5745	0.6845	RANDOM
NaiveBayes_2024-01-14_20-13-27	1 hour ago	5.0s	-	0.7275	None

Rysunek 1: Tabela przedstawia wyniki naszych modeli przy użyciu frameworka `MLflow`.

Wyniki 1 monitorowaliśmy przy pomocy narzędzia `MLflow`, które pozwoliło nam na analizę otrzymanych rezultatów i szybki dostęp do artefaktów najlepszych modeli.

Najlepszym modelem, który potencjalnie jest przetrenowany jest CatBoost z hiperparametrami wybranymi poprzez algoryt random search. Zdecydowaliśmy się na wykonanie finalnej predykcji na zbiorze testowym przy użyciu tego modelu.

Hiperparametry CatBoosta 1(pogrubione - wybrane po przeszukiwaniu):

Tabela 1: Siatka hiperparametrów dla algorytmu RandomForestClassifier oraz końcowe hiperparametry

Parametr	**Wartości**
`Depth`	[6, 7, 8, **9**, 10]
`Learning Rate`	`np.linspace(0.01, 0.1, 10)` – **0.05**
`Iterations`	`np.arange(100, 301, 100)` – **300**
`L2 leaf reg`	`np.arange(1, 10, 2)` – **9**
`Subsample`	`np.linspace(0.8, 1.0, 3)` – **1**
`Border count`	[32, 64, 128] – **32**
`Loss function`	['Logloss', 'CrossEntropy'] – **'Logloss'**
`Eval metric`	['Logloss', 'AUC'] – **'Logloss'**
`Bootstrap Type`	['Bayesian', 'Bernoulli', 'MVS'] – **"MVS"**

4 AutoML

W drugim podejściu modelu wykorzystano frameworki AutoMLowe na surowych danych:

- `AutoSklearn`
- `AutoGluon`

4.1 Opis eksperymentów

Wykorzystaliśmy wersję `1` AutoSklearna z kros-walidacją 7-krotną, gdzię wielkość zbioru treningowego w każdym foldzie wynosiła `75%` całych danych treningowych.

W przypadku AutoGluona zdecydowaliśmy się na domyślne parametry.

W obydwu frameworkach naszą funkcją względem, której optymalizowane były modele było `balanced accuracy`

4.2 Wyniki Eksperymentów

W poniższej tabeli (2) przedstawiają najlepsze wyniki po wytrenowaniu modeli przez `AutoSklearna` i `AutoGluona` na zbiorze walidacyjnym w ramach kros-walidacji.

Tabela 2: Wyniki frameworków AutoML

Framework	**best_score**
`AutoSklearn`	0.885
`AutoGluon`	0.860

4.2.1 Wyniki eksperymentów na 5% zbioru treningowego

Tabela 3: Wyniki frameworków AutoML

Framework	**test 5%**
`AutoSklearn`	0.933 (28/30)
`AutoGluon`	0.866 (26/30)

Zdecydowaliśmy się ostatecznie skorzystać z AutoSklearn do wykonania końcowych predykcji. Ze względu na niewielką liczbę dostępnych danych, trudno jednoznacznie przewidzieć, który model lepiej uogólni się na danych testowych. Nasza decyzja była oparta na intuicji oraz popularnym przekonaniu, że 'im więcej, tym lepiej'. Niemniej jednak, zdajemy sobie sprawę, że to podejście może nas zawieść w tym konkretnym przypadku.

5 Podsumowanie

Podsumowując, nasza praca skupiła się na opracowaniu różnych metod klasyfikacji dla sztucznie wygenerowanego zbioru danych "artificial". Porównaliśmy podejście ręczne z użyciem różnych algorytmów klasyczny m.in. CatBoost czy XGBoost oraz podejście typu AutoML (AutoSklearn). Ostatecznie, wraz z popularnym przekonaniem "im więcej, tym lepiej", wybraliśmy AutoSklearn do końcowych predykcji na zbiorze testowym, zdając sobie jednak sprawę, że w przypadku niewielkich danych to podejście może być ryzykowne.