Automatyczne uczenie maszynowe
Praca domowa 2

Laura Korona
Piotr Nieciecki

Styczeń 2024

1 Wstęp

Celem pracy domowej jest zaproponowanie metod klasyfikacji, które pozwolą stworzyć modele o jak największej mocy predykcyjnej - jeden z modeli ma być zbudowany ręcznie, zaś drugi z użyciem poznanych na zajęciach frameworków AutoML. Modele mają dokonywać klasyfikacji binarnej na sztucznie wygenerowanym zbiorze danych, w którym ukryte zostały istotne zmienne.

2 Model zbudowany ręcznie

2.1 Tworzenie modelu

Tworzenie modelu zbudowanego ręcznie rozpoczęto od wyszukania najważniejszych cech w zbiorze danych. Posłużono się w tym celu funkcją `permutation_importance` z pakietu `scikit-learn`, która poprzez przestawianie kolejności wartości badanej cechy określa jej ważność w danym modelu. Badanym modelem był las las losowy o następujących hiperparametrach:

- `n_estimators`=100
- `max_depth`=5

Pozostałe hiperparametry miały domyślne wartości. Model ten został wybrany do badania, gdyż stwierdzono, że dawał wystarczająco dobry wynik na zbiorze danych, w którym znajdowało się dużo nieistotnych cech. Wynik ten mierzony był jako *balanced accuracu* i wynosił w przybliżeniu 0.63.

Po zbadaniu modelu i zbioru danych funkcją `permutation_importance` wybrano 10 najważniejszych cech. Następnie przeprowadzono `RandomizedSearchCV` w celu znalezienia najlepszych hiperparametrów dla lasu losowego - badano następujące hiperparametry:

- `n_estimators`
- `max_depth`
- `criterion`

- `min_samples_split`.

Przeprowadzono 300 iteracji.

2.2 Ostateczny model

Najlepszy zestaw hiperparametrów znaleziony przez procedurę `RandomizedSearchCV` jest następujący:

- `n_estimators`=162
- `max_depth`=45
- `criterion`=`gini`
- `min_samples_split`=2.

Las losowy o powyższych hiperparametrach daje `balanced_accuracy` wynoszące w przybliżeniu 0.86 (zmierzone podczas kroswalidacji w `RandomizedSearchCV`). Podczas sprawdzenia poprawności działania modelu na 5% danych zbioru testowego otrzymano zaś wynik wynoszący około 0.93.

3 Model wykorzystujący framework AutoML

Do tworzenia modelu wykorzystującego metody z dziedziny AutoML użyliśmy technologię AutoGluon. Po wczytaniu danych z plików podzieliśmy je na zbiór testowy oraz zbiór do trenowania. Zdecydowaliśmy się, że 20% danych przeznaczymy do testowania modelu, podczas gdy pozostałem 80% będzie służyło do trenowania.

Do stworzenia modelu wykorzystaliśmy klasę `TabularPredictor`, która jest dedykowana do danych tabelarycznych, czyli takich z jakimi mieliśmy do czynienia przy omawianej pracy domowej. Podczas tworzenia instancji podaliśmy kolumnę, której wartości chcieliśmy przewidywać. Dodatkowo zdefiniowaliśmy metrykę, która najbardziej nas interesuje - `balanced accuracy`. Ze szczegółów dodaliśmy jeszcze ścieżkę pod którą model miał być zapisany zdefiniowaliśmy, aby framework zapisywał logi tworzone podczas działania do późniejszej analizy.

AutoGluon posiada różne predefiniowane ustawienia, które wpływają no to jak szybko model się uczy kosztem dokładności zwróconego modelu. Do naszego uczenia wybraliśmy ten o największej dokładności - `best quality`. Czas uczenia wyniósł w przybliżeniu jedną godzinę. Na końcu zmierzyliśmy nasz model na zbiorze testowym oraz uruchomiliśmy dla niego przygotowany przez prowadzące program walidacyjny. Uzyskane wyniki opisane zostały w rozdziale 3.2.

3.1 Opis modelu

Model, który uzyskaliśmy - `WeightedEnsemble_L3` był wagowym połączeniem dwóch innych modeli. W jego skład wchodził model `CatBoost_BAG_L2` z wagą wynoszącą 0.955 oraz `RandomForestGini_BAG_L2` z wagą 0.045.

3.2 Wyniki

Podczas swojego działania AutoGluon wyznaczył, że miara `balanced accuracy` dla uzyskanego modelu wynosi 0.8718. W naszych testach uzyskaliśmy wynik na poziomie 0.8445.

4 Podsumowanie

Z zebranych przez nas danych wynika, że model wytrenowany ręcznie ma duże szansę na uzyskanie lepszego wyniku od uczonego w sposób automatyczny. Jednocześnie warto podkreślić, że czas poświęcony na uczenie ręczne był znacznie większy. Z tego powodu uważamy, że technologię związane z automatycznym uczeniem maszynowym mają bardzo duży potencjał, aby wspomagać, ale jeszcze nie zastępować ludzi w procesie uczenia modeli.