Code review - FLAML

Marcel Witas

May 2022

1 Wstęp

Niniejszy raport zawiera wnioski i uwagi dotyczące funkcji przygotowującej dowolny tabelaryczny zbiór danych opisujący problem predykcyjny do użycia frameworku do AutoML **FLAML**. Recenzowany kod został przygotowany przez grupę KTR.

2 Czy kod osiąga cel, który postawiono?

Funkcja przyjmuje zmienne objaśniające X, kolumne objaśnianą y, można podać również podział do kroswalidacji. Następnie, ponieważ **FLAML** nie oferuje preprocessingu, na danych jest wykonywane kilka jego kroków :

- usunięcie zmiennych, które prawdopodobnie są "id",
- usunięcie zmiennych o zerowej wariancji,
- uzupełnienie braków danych najczęstszą wartością,
- zastąpienie outlierów wartościami skrajnymi,
- upodobnienie rozkładu do rozkładu normalnego,
- przeskalowanie do rozkładu standardowego,
- zamiana kolumn typu object w ich WOE.

Uwagi dotyczące preprocessingu pojawią się jeszcze w dalszej części raportu. Zakładając jednak ich poprawność, na tak przygotowanych danych można wywołać już funkcję fit z frameworku, która trenuje kolejne modele poszukując optymalnego. Zatem kod osiąga cel, który postawiono.

3 Czy w kodzie są jakieś oczywiste błędy logiczne?

Pewne wątpliwości można mieć do jednej z funkcji związanej z preprocessingiem. Klasa *idRemover* teoretycznie ma usuwać zmienne, które mogą być indeksem. Mogłoby to mieć znaczenie, na przykład gdy funkcja podaczas wczytania zbioru danych kolumna z indeksami jako zwykłą lub po prostu w danych istnieje zmienna id. Taka kolumna nie powinna mieć wpływu na estymator.

Problem w tym, że w *idRemover* sprawdzane jest jedynie, czy rozmiar kolumny jest równy liczbie unikalnych wartości znajdujących się w niej. To oczywiście działa, gdy mamy do czynienia z kolumnami typu "id". Jednakże, możemy przez również innych zmiennych numerycznych, np. które również będą unikalne, ale mogą nieść istotną informację, np. kiedy dane są zapisane z dokładnością do kilku miejsc po przecinku.

Zatem być może należałoby najpierw sprawdzić typ kolumny, i jeśli jest to float, nie usuwać jej. Należałoby się również zastanowić nad typem int, ponieważ wtedy może być to zarówno kolumna "id", jak i kolumna numeryczna o pewnym rozkładzie, zatem usunięcie jej, może czasami również nie być optymalne.

Oczywiście nie jest to błąd, który uniemożliwi działanie funkcji, ale może usuwać za dużo kolumn, co ma wpływ na jakość modelu.

4 Czy patrząc na wymagania zawarte podczas prezentacji są one w pełni zaimplementowane?

W prezentacji wspominano m.in. o tym, że **FLAML** nie ma wbudowanego preprocessingu. Zaimplementowane zostało kilka uniwersalnych kroków, które przygotowują dane, zatem wymagania zostały spełnione.

5 Czy kod jest zgodny z istniejącymi wytycznymi stylistycznymi? (czy kod jest zgodny z PEP 8)

Kod jest czytelny i nie ma wielu błędów stylistycznych. Pojawiają się jedynie drobne niezgodności z PEP 8, takie jak niewłaściwa (zbyt duża lub zbyt mała) liczba pustych linijek.

6 Czy są jakieś obszary, w których kod mógłby zostać poprawiony? (skrócić, przyspieszyć, itp.)

Oprócz poprawy błędu związanego z *idRemover* kod jest raczej dobrze napisany, i trudno znaleźć miejsca, w którym mógłby zostać realnie poprawiony.

7 Czy dokumentacja i komentarze są wystarczające?

W większości przypadków dokumentacja i komentarze są zrozumiałe i wystarczające. Ponieważ niektóre komentarze są po polsku, inne po angielsku, może warto byłoby się zdecydować na jeden język.

8 Czy udało się odtworzyć zamieszone przykłady w kodzie?

Zamieszczone przykłady udało się odtworzyć z podobnym rezultatem. Najlepsze wyniki ROC AUC dla każdego zbioru różniły się o mniej niż 0.1.

W wynikach, dla każdego zbioru oprócz ROC AUC została przedstawiony czas trenowania najlepszej konfiguracji. Być może lepiej byłoby podać całkowity czas poszukiwania najlepszego modelu, ponieważ on wydaje się tu ważniejszy.

9 Czy udało się użyć przygotowanych kodów na nowych danych?

Kod udało się użyć na nowych danych. Sprawdzone zostały dane zawierające kolumnę unikalnych wartości typu "id", jak i kolumne zawierącą dane numeryczne mogące nieść informację. Zgodnie z oczekiwaniami, obie zostały usunięte, co potwierdza, że ta część wymaga poprawy. Poza tym, nie było problemów, a **FLAML** znajduje dobre modele dość szybko, szczególnie w porównaniu do np. *AutoPyTorcha*.

10 Podsumowanie

W kodzie przygotowującym dowolny tabelaryczny zbiór danych opisujący problem predykcyjny do użycia **FLAML** jest jeden wyraźny błąd dotyczący preprocessingu. Warto byłoby rozważyć również ujednolicenie komentarzy i lekkie poprawki stylistyczne. Pomimo tego, kod został napisany dobrze i spełnia swój cel.