Code Review funkcji grupy Gakubu

Jan Kruszewski

5 maja 2022

1 Wprowadzenie

W poniższym dokumencie przeprowadzam Code Review funkcji przygotowanej przez grupe Gakubu, której praca dotyczy pakietu AutoGluon. Celem funkcji było przygotowanie dowolego tabularycznego zbioru danych opisujacego binarny, problem predykcyjny do użycia frameworku AutoGluon.

2 Czy kod osiaga postawiony cel?

Postawiony cel sprowadzał sie do 4 punktów: - wczytania zbiorów danych z benchmarku do przetestowania funkcji - przeprowadzenie preprocessingu - modelowanie

2.1 Wczytanie danych

W celu wczytania zbiorów zespół Gakub stworzył pomocnicza funkcje `create_data_frame_from_arff`, która przyjawszy na wejściu adres url zwraca ramke danych. Funkcja jest, krótka, czytelna i co najważniejsze spełnia swój cel.

2.2 Preprocessing

Preprocessing wykonywany jest wewnatrz głównej funkcji `make_kfold_cross_valiation`. Preprocessing obejmuje encoding zmiennych w formacie string za pomoca systemu Unicode UTF-8. Pozostały preprocessing i feature engineering został pozostawiony AutoGluon'owi, który wykonuje cześć pracy za użytkownika.

2.3 Modelowanie

Tworzenie modelów oraz ich walidacja wykonywana jest w funkcji `make_kfold_cross_valiation`. Na wejściu podaje poza argumentami funkcji `create_data_frame_from_arff` nazwe zbioru, nazwe zmiennej celu, ilość foldów do kroswalidacji oraz inne parametry dla kroswalidacji. Funkcja zwraca rezultaty kroswalidacji w postaci tablicy, zawierajacej rezultaty dla kolejnych foldów w postaci nazwy modelu, wyniku dla metryki roc_auc oraz nazwy zbioru. Funkcja po przetestowaniu działa bezbłednie, możnaby jednak moim zdaniem pokusić sie o umieszczenie w wynikowej tablicy wyników dla również innych metryk takich jak np. f1 Score czy accuracy.

3 Czy w kodzie sa jakieś oczywiste błedy logiczne?

Po dokładnej analizie kodu nie znalazłem żadnych błedów logicznych.

4 Czy wymagania z prezentacji zostały w pełni zaimplementowane?

W prezentacji grupa Gakubu przedstawiła wyniki swojej pracy w pełni wynikajace z pracy wykonanej nad kodem. Wszystkie założenia zostały zaimplementowane w rzetelny sposób.

5 Czy kod jest zgodny z wytycznymi PEP8?

Zdecydowanie można odpowiedzieć twierdzaco na zadane powyżej pytanie. Funkcje oraz zmienne zostały nazwana w odpowiedni sposób. Twórcy operuja pustymi przestrzeniami by zwiekszyć czytelność kodu. Nie przekraczaczane sa maksymalne długości linijek kodu(można to zauważyć np. przy deklaracji zmiennych przy funkcji `make_kfold_cross_valiation`. Co robi wrażenie autorzy kodu nie zapomnieli również o potrójnym cudzyslowiu przy opisywaniu funkcji oraz nie dodawania spacji przy znaku '=' wewnatrz funkcji. Jedyna rzecza do której możnaby sie przyczepić to drobne literówki.

6 Czy sa jakieś obszary, w których kod mógłby zostać poprawiony?

Jak wspominałem wcześniej sadze, że możnaby dodać inne metryki do oceny jakości modeli. Nie widze innych oczywistych sposobów poprawy kodu, ale z checia przeczytam co na ten temat do powiedzenia maja inne grupy.

7 Czy dokumentacja i komentarze sa wystarczajace?

Uważam, że objaśnienia funkcji oraz komentarze sa zdecydowana zaleta kodu zespołu Gakubu. Każda funkcja została opisana w jasny sposób. Komentarze wewnatrz kodu również pomagaja zrozumieć dane fragmenty kodu.

8 Czy udało sie odtworzyć zamieszone przykłady w kodzie?

Przetestowałem funkcje grupy Gakubu na moim komputerze odpalajac ja dla 4 zbiorów danych z benchamarku. Uważałem uruchamianie jej dla wszystkich zbiorów za zbedne. Wszystko działało w odpowiedni sposób opisany powyżej.

9 Czy udało sie użyć kodu na nowych danych?

Za pomoca kodu, co cieszy udało sie stowrzyć model dla innego binarnego zbioru danych 'liver disorders'.

10 Podsumowanie

Kod bedacy rezultatem kamienia milowego 2 jest przejrzysty oraz co najwazniejsze działa. Kod spełnia wytyczne PEP8 co ułatwia jego czytanie. Pomijajac drobne literówki w dokumentacji kod nie pozostawia dużego pola do krytyki.