

Code Review PD3

Mikołaj Gałkowski

5 maja 2022

Code Review

Raport dotyczący kodu przygotowanego przez grupę **KTR** używającej framework'a **FLAML**.

1) Czy ten kod osiąga cel, który postawiono?

Tak, kod osiąga cel, który postawiono. Dane są odpowiednio poddane preprocessingowi i wywoływany jest na nich framework.

2) Czy w kodzie są jakieś oczywiste błędy logiczne?

Jedyną rzeczą, która może być niepoprawna jest `idRemover`, który może wyrzucać kolumny, które niekoniecznie będą kolumnami `id`, lecz kolumnami numerycznymi niosącymi pewną informację.

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

Tak, wszystkie wymagania zostały zaimplementowane. Grupa **KTR** wspomniała, że ich framework nie oferuje preprocessingu. Postanowili samemu zaimplementować go, dzięki czemu ich Pipeline jest kompletny.

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

Klasy `ignorer` oraz `idRemover` zostały nazwane z małych liter. Nie jest to zgodne z PEP8.

W wielu miejscach nie zostały zachowane odstępy, które ułatwiały by czytelność np. klasa `ZerVarRemover`

```
1  from time import perf_counter
2  from sklearn.base import TransformerMixin
3  import logging
4  from time import perf_counter
5  class ZeroVarRemover(TransformerMixin):
6      __slots__ = '_sus', '_logger'
7      def __init__(self):
8          super().__init__()
9          self._sus = None
10         self._logger = logging.getLogger(__name__)
11     def fit(self, x, y):
12         __start = perf_counter()
13         self._sus = x.loc[:, x.nunique() == 1]
14         self._logger.info(f"zerovarremover cols staged to be ...
                                removed: ...
                                {self._sus.columns.tolist()} ...
                                ")
15
16     return self
17     def transform(self, x):
18         return x.drop(self._sus, axis = 1)
```

A wystarczyło to wrzucić do **Black'a**, żeby zwiększyć czytelność kodu.

```
1  from time import perf_counter
2  from sklearn.base import TransformerMixin
3  import logging
4  from time import perf_counter
5
6
7  class ZeroVarRemover(TransformerMixin):
8      __slots__ = "_sus", "_logger"
9
10     def __init__(self):
11         super().__init__()
12         self._sus = None
13         self._logger = logging.getLogger(__name__)
14
15     def fit(self, x, y):
16         __start = perf_counter()
17         self._sus = x.loc[:, x.nunique() == 1]
18         self._logger.info(
19             "zerovarremover cols "
20             + "staged to be removed: "
21             + self._sus.columns.tolist()
22         )
23         return self
24
25     def transform(self, x):
26         return x.drop(self._sus, axis=1)
```

Ponadto w niektórych funkcjach/metodach brakuje odstępów w celu większej czytelności i możliwości dostrzeżenia poszczególnych kroków np. w `OutlierTransform'erze` w metodzie `fit()` brakuje odstępów co zmniejsza czytelność. Gdzieś tam brakuje również spacji np. w przypisaniu zmiennych w celu zwiększenia czytelności.

Niektóre komentarze tłumaczą oczywistości np. `SimpleImputer(strategy="most_frequent"), ...`
uzupełnienie braków danych. Moim zdaniem są zbędne.

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

Wydaje mi się, że nie.

6) Czy dokumentacja i komentarze są wystarczające?

Raczej tak, choć w niektórych miejscach brakuje krótkiego docstringa np. `ZerVarRemover` nie ma żadnego komentarza. Ponadto tak jak wcześniej wspominałem niektóre komentarze są zbędne. Język komentarzy również powinien zostać ujednol-

cony.

7) Czy udało się odtworzyć zamieszczone przykłady w kodzie?

Tak, udało się odtworzyć przykłady zamieszczone w kodzie.

8) Czy udało się użyć przygotowanych kodów na nowych danych?

Tak przetestowałem framework na 3 datasetach z platformy Kaggle:

1. [Stroke Prediction](#) **ROC AUC - 84.7%**
2. [Heart failure](#) **ROC AUC - 93.6%**
3. [Rain in Australia](#) **ROC AUC - 61.7%**

Podsumowanie

Grupie **KTR** udało się stworzyć w pełni działający kod, który potrzebuje małych poprawek (komentarze, odstępy, nazwy klas, `idRemover`).