

Курс по методам машинного обучения

Практическое задание № 2

Основы numpy + matplotlib + pandas

Крыжановская Светлана

1 Описание задания

Задание состоит из трех частей, посвященных работе с табличными данными с помощью библиотеки **pandas**, визуализации с помощью библиотек **matplotlib**, **seaborn**, **plotly** и векторным вычислениям с помощью библиотеки **numpy**. В каждой части Вам необходимо выполнить несколько заданий.

2 Формат задания

Задание состоит из двух различных видов подзаданий: **кросс-рецензирование** и **юнит-тестирование**. В этом задании за юнит-тестирование можно получить максимум **27 баллов**, а за кросс-рецензирование — **13 баллов**.

Юнит-тестирование — уже знакомый вам формат, в котором необходимо реализовать какие-либо функции. В данном задании вам необходимо реализовать функции, находящиеся в файлах `functions.py` и `functions_vectorised.py`. После реализации ваш код можно протестировать локально, а затем его необходимо сдать в проверяющую систему (вкладка `numpy-pandas-matplotlib`

(unit-tests)). Напомним, что такой вид заданий можно сдавать после дедлайна со штрафом 40%.

Кросс-рецензирование — в таких заданиях вам необходимо написать код, ответить на вопросы, что-то визуализировать в ноутбуке `numpy-pandas-matplotlib.ipynb`, который можно найти в дополнительных файлах во вкладке `numpy-pandas-matplotlib (unit-tests)` и в шаблоне решения во вкладке `numpy-pandas-matplotlib (ноутбук)`. Выполненный ноутбук `numpy-pandas-matplotlib.ipynb` необходимо **сдать в тестирующую систему во вкладку `numpy-pandas-matplotlib (ноутбук)`**. Данный вид задания принимается **только до дедлайна**. После окончания срока сдачи, у вас будет еще **неделя** на проверку решений как минимум **3х других студентов** — это **необходимое** условие для получения оценки за вашу работу. Если вы считаете, что вас оценили неправильно, можете писать на почту `ml.cmc@mail.ru` с темой письма `ВМК.ML[Задание 2][peer-review]` с просьбой перепроверить оценивание задания.

Важно: *перед сдачей, проверьте пожалуйста, что не оставили в ноутбуке где-либо свои ФИО, группу и т.д. — кросс-рецензирование проводится анонимно.*

В ноутбуке, а также здесь подробно описано, к какому типу заданий относится та или иная часть в ноутбуке.

2.1 pandas

В ноутбуке `numpy-pandas-matplotlib.ipynb` необходимо ответить на 10 вопросов по анализу табличных данных с помощью библиоте-

ки pandas . Многие из заданий можно выполнить несколькими способами. Не существуют единственно верного, но для решения так или иначе должен быть задействован арсенал pandas. Все задания в этой части оцениваются по системе **кросс-рецензирования**.

2.2 matplotlib

В ноутбуке `numpy-pandas-matplotlib.ipynb` необходимо построить 3 визуализации по табличным данным. При желании в решении заданий допустимо пользоваться любыми средствами для визуализации в ноутбуке — **главное, проверьте, что при конвертации ноутбука в html, а также при открытии сданного ноутбука из проверяющей системы, все графики по-прежнему видны**. Все графики будут оцениваться по системе **кросс-рецензирования** на содержательность и соответствие правилам, описанным в ноутбуке.

2.3 numpy

В файлах `functions.py` и `functions_vectorised.py` находятся шаблоны шести функций, которые необходимо реализовать в рамках задания. Формулировки заданий прописаны в ноутбуке `numpy-pandas-matplotlib.ipynb`. Библиотеками, не объявленными в импорте в файлах с шаблонами функций, пользоваться запрещено. Модули с реализованными функциями необходимо **сдать в систему для автоматической проверки**. Все тесты находятся в открытом доступе и предварительное тестирование может быть запущено локально на компьютере.

Помимо реализации функций, необходимо провести сравнение скорости работы функций в ноутбуке `numpy-pandas-matplotlib.ipynb`. Графики и выводы будут оцениваться по системе **кросс-рецензирования**.

Важно: задания, в которых есть решения, содержащие в каком-либо виде взлом тестов, дополнительные импорты и прочие нечестные приемы, будут автоматически оценены в 0 баллов без права передачи задания.

3 Тестирование

Для локального тестирования необходимо скачать из тестирующей системы папку `public_tests` с публичными тестами и скрипт для тестирования `run.py` и положить их в одну директорию вместе с файлами `functions.py` и `functions_vectorised.py` с реализованными функциями. Для тестирования необходимо запустить команду `python3 run.py public_tests`.

Каждая функция тестируется на 4-6 тестах на правильность, а функции из модуля `functions_vectorised.py` дополнительно тестируются на время выполнения. Входные тестовые данные для функций лежат в папках `[0-9]_task[1-6]_[vectorised/unvectorised]_input`, а правильные решения в папках `[0-9]_task[1-6]_[vectorised/unvectorised]_gt`. Входные тестовые данные для функций хранятся в NumPY файлах, а правильные результаты в формате `.pkl`. Примеры чтения входных данных и правильных ответов:

```
1 import numpy as np
2 x = np.load('public_tests/07_task1_vectorised_input
    /input_0/x.npy')
```

```
1 import pickle
2 with open('public_tests/07_task1_vectorised_gt/
    output_0.pkl', 'rb') as f:
3     data = pickle.load(f)
```