ВМК МГУ

Задание 3. K ближайших соседей (kNN)

Курс по методам машинного обучения, 2023-2024, Драгунов Никита

1 Характеристики задания

- **Длительность:** 2 недели
- **Кросс-проверка:** 23 балла; в течение 1 недели после дедлайна; нельзя сдавать после жесткого дедлайна
- **Юнит-тестирование:** 17 баллов; Можно сдавать после дедлайна со штрафом в 40%; Публичная часть; PEP8
- **Почта:** ml.cmc@mail.ru
- **Темы для писем на почту:** ВМК.ML[Задание 3][peer-review], ВМК.ML[Задание 3][unit-tests]

Кросс-проверка: 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 или есть какие-то вопросы, можете писать на почту с соответствующей темой письма

2 Описание задания

В настоящем задании вы познакомитесь с алгоритмом k ближайших соседей (k-NN, KNN, kNN) для решения задач классификации и регрессии.

Приведем здесь краткое напоминание о принципе работы kNN. Пусть дана обучающая выборка $X = (x_i, y_i)$ и функция расстояния ρ. Требуется классифицировать новый объект u. Алгоритм k ближайших соседей относит объект u к тому классу, представителей которого окажется больше всего среди k его ближайших по ρ соседей: $a(u; X, k) = \underset{y \in Y}{\operatorname{argmax}} \sum_{i=1}^{k} w_i [y_u^{(i)} = y]$, где $y_u^{(i)}$ – метка класса i–го соседа объекта u. В классическом методе k ближайших соседей все объекты имеют единичные веса: $w_i = 1$. Альтернативой данному подходу служат веса, обратно пропорциональные расстоянию между объектами. При решении задачи регрессии ответом алгоритма служит средневзвешенное значение меток $y_u^{(i)}$ среди k ближайших соседей.

3 Кросс-проверка

- **Ссылка на задание:** ссылка тут

Внимание! Отправлять задание нужно в систему во вкладку **KNN (notebook).**

Внимание! Отправлять задание нужно только с расширением ipynb! После отправки проверьте корректность загруженного задания в систему, просмотрев глазами загруженное решение (оно автоматически сконвертируется в html). Как это сделать, можно найти в туториале тут

Внимание!: Перед сдачей проверьте, пожалуйста, что не оставили в ноутбуке где-либо свои ФИО, группу и так далее — кросс-рецензирование проводится анонимно.

4 Юнит-тестирование

- В ходе выполнения задания 1.1 из jupyter-notebook вам необходимо реализовать два вида нормализации признакового пространства в модуле `scalers.py`. Первый вид нормализации `StandardScaler` выполняет нормализацию к распределению с нулевым матожиданием и единичной дисперсией. Второй вид нормализации `MinMaxScaler` отображает признаковые описания в отрезок [0, 1].

- В ходе выполнения задания 2.1 из jupyter-notebook вам необходимо реализовать две функции в модуле `cross_val.py`. Первая функция `kfold_split` выполняет генерацию индексов обучающей и валидационной выборок для кросс-валидации. Вторая функция `knn_cv_score` выполняет кросс-валидацию для KNN—модели.

Файлы `scalers.py` и `cross_val.py` можно найти в архиве из **шаблона решения** во вкладке **KNN (unit-tests).** Более подробное описание входных и выходных данных вы найдете в этих файлах. После реализации ваш код можно протестировать локально (ссылка), а затем его необходимо сдать в проверяющую систему **(вкладка KNN (unit-tests))**.

Замечание: Запрещается пользоваться библиотеками, импорт которых не объявлен в файле с шаблонами функций.

Замечание: 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 пересдачи задания.

5 Стиль программирования

Внимание! Обновление!!!

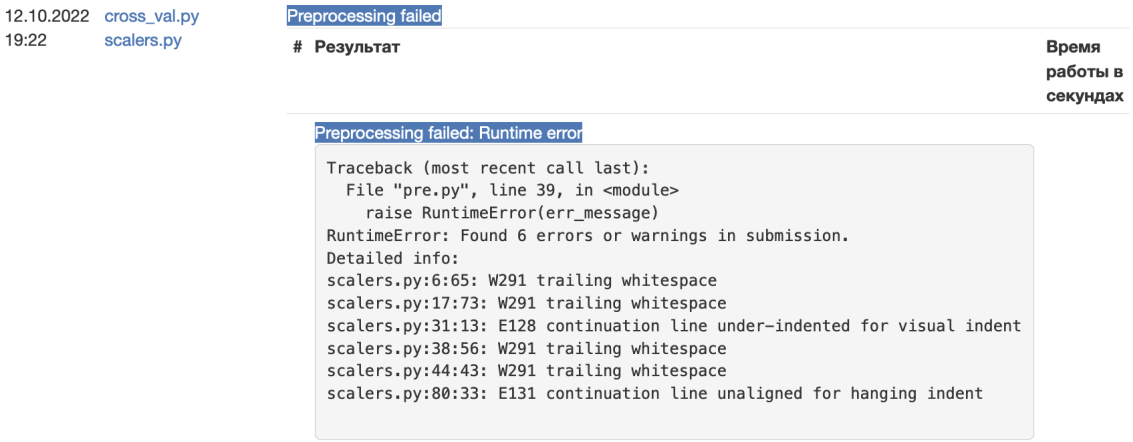
Начиная с этого задания при выполнении задач типа unit-tests, ML-задания вам необходимо будет соблюдать определенный стиль программирования (codestyle). В данном случае мы выбирали PEP8 как один из популярных стилей для языка Python. Зачем мы это вводим? Хорошая читаемость кода – не менее важный параметр, чем работоспособность кода :) Единый стиль позволяет быстрее понимать код сокомандников (в командных проектах, например), упрощает понимание кода (как другим, так и вам). Также, привыкнув к какому-либо стилю программирования, вам будет проще переориентироваться на другой.

Полезные при изучении PEP8 ссылки, если что-то непонятно, дополнительный материал можно найти самостоятельно в интернете:

- Официальный сайт PEP8, на английском
- Небольшое руководство по основам на русском

Требования к PEP8 мы вводим только для заданий с авто-тестами, требований к такому же оформлению ноутбуков нет. Но улучшение качества кода в соответствии с PEP8 в них приветствуется!

Внимание!!! В проверяющей системе, при несоответствии прикрепляемого кода PEP8, будет высвечиваться вердикт `Preprocessing failed`. Более подробно посмотреть на ошибки можно, нажав на них:



Также посылки, упавшие по code style, считаются за попытку сдачи и идут в счет общего количества посылок за день.

Проверить стиль программирования локально можно при помощи утилиты pycodestyle (в окружении, которое вы ставили, эта утилита уже есть) с параметром максимальной длины строки (мы используем 160 вместе дефолтных 79):

```
pycodestyle --max-line-length=160 your_file_with_functions.py
```

6 Тестирование

В cv-gml можно скачать все файлы, необходимые для тестирования, одним архивом. Для этого просто скачайте zip-архив во вкладке **шаблон решения** соответствующего задания и разархивируйте его. Далее следуйте инструкциям по запуску тестирования.

Тесты запускаются с помощью команд:

```
python run.py unittest scalers
python run.py unittest cv
```