ВМК МГУ

Задание [Base][8]. Деревья решений

Курс по методам машинного обучения, 2024-2025, Юлиан Сердюк

1 Характеристики задания

- **Длительность:** 1 неделя
- **Base**
  - **Кросс-проверка:** 7 баллов; в течение 1 недели после дедлайна; нельзя сдавать после жесткого дедлайна
  - **Юнит-тестирование:** 3 балла; можно сдавать после дедлайна со штрафом в 40%; публичная и приватная части; PEP8
- **Почта:** ml.cmc@mail.ru
- **Темы для писем на почту:** ВМК.ML[Задание [Base][8]][peer-review], ВМК.ML[Задание [Base][8]][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 вопросы, можете писать на почту с соответствующей темой письма

2 Описание задания

Привет, ребятушки!
В этом файле описываются задание типа unit-test.

3 Кросс-проверка

Внимание! Отправлять задание нужно в систему во вкладку с пометкой (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 по проверяющей системе на сайте курса.

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 (Оценка разбиений)

В этом задании Вам нужно реализовать функцию, которая оценивает качество разбиения множества объектов. На лекции вам рассказали о трех метриках: gini, entropy и classification_error. В этом задании вам нужно реализовать их все. Для вычислений разрешается пользоваться библиотекой numpy.

Замечание: При вычислении этих метрик используйте **натуральный** логарифм!

4.1 Формат отправки

В шаблонном файле split_measures.py, вам необходимо реализовать функцию evaluate_measures. На вход эта функция получает список меток классов объектов, которые попали в одно из получившихся разбиений. Эта функция возвращает словарь, в котором содержатся значения всех трёх метрик. Вовращаемый словарь

должен содержать три ключа: “gini”, “entropy” и “error”, которым должны быть сопоставлены значения метрик gini, entropy и classification error в типе float ().

Ниже показан схематический пример такого скрипта.

```
def evaluate_measures(sample):
    return {"gini": float(np.sum(sample)),
            "entropy": float(np.min(sample)),
            "error": float(np.max(sample))}

print(evaluate_measures([1, 2, 3, 2, 3, 1, 2, 0]))
```

4.2 Используемая метрика

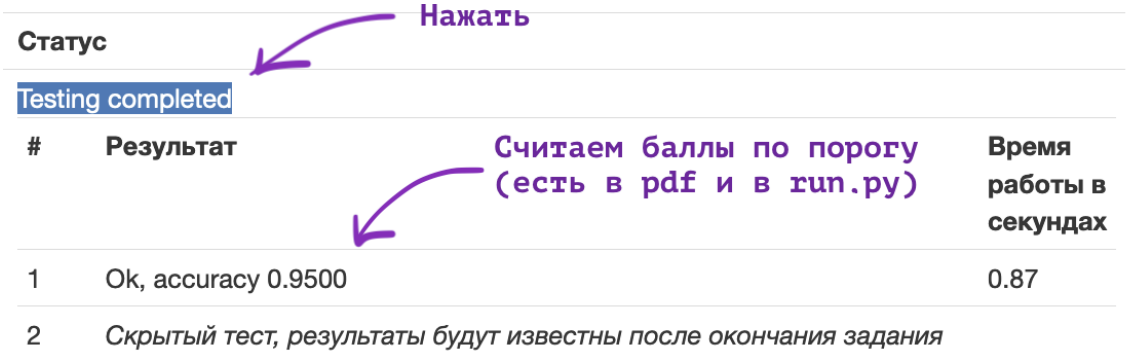
Для успешного выполнения задания необходимо, чтобы Ваше полученное значение отличалось от истинного не более, чем на 0.001 (абсолютное значение разности).

4.3 Оценивание

Задание разбито на две части: публичную (один пример) и приватную (содержащую несколько выборок). Для получения оценки в каждой из частей вам необходимо пройти все тесты без ошибок вычислить ответ и получить значение, удовлетворяющее метрике.

В данном задании два теста: публичный, который оценивается из **1 балла**, и приватный (недоступен вам), оцениваемый из **2 баллов**. За публичный тест Вы можете получить гарантированные баллы уже до жёсткого дедлайна, а баллы за приватный тест вам откроются после дедлайна.

В системе, из-за наличия закрытых тестов, вы не сможете увидеть свои баллы до дедлайна даже за открытый тест. Но вы сможете посчитать баллы за открытый тест способом, приведенным на картинке ниже:



4.4 Дополнительные материалы

Если вы хотите почитать что это за метрики и как их вычислять, Вы можете обратиться к лекциям Китова или Дьяконова. Воронцов, к сожалению, слишком поверхностно упоминает эти метрики :(

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

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

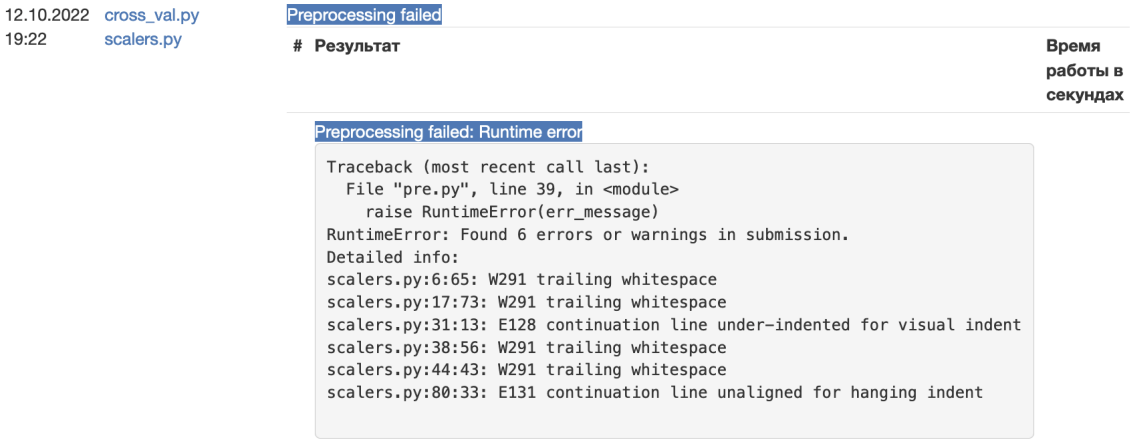
Python. Зачем мы это вводим? Хорошая читаемость кода – не менее важный параметр, чем работоспособность кода :) Единый стиль позволяет быстрее понимать код сокомандников (в командных проектах, например), упрощает понимание кода (как другим, так и вам). Также, привыкнув к какому-либо стилю программирования, 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.

Если всё сделано правильно, то при переходе в соответствующую папку в консоли и запуске команды ‘python run.py‘ Вы не должны получать сообщений об ошибках. Учтите, что после запуска скрипта будет создано несколько дополнительных файлов и директорий (это связано с работой тестирующей системы).

Запуск тестов производится командой

```
python run.py
```