# Часто задаваемые вопросы (MATH&ML)

## ▼ Почему у меня ответы не сходятся с ответами платформы?

В первую очередь, убедитесь, что у вас стоят последние версии библиотек pandas, numpy и scikit-learn. С обновлением библиотек алгоритмы машинного обучения в них могут немного по-другому работать, поэтому результаты могут не сходиться.

Посмотреть версии библиотек можно следующей командой в командной строке:

```
pip show pandas numpy scikit-learn
```

Либо же с помощью питона – вот так:

```
import pandas as pd import numpy as np import sklearn print(pd.__version__)
print(np.__version__) print(sklearn.__version__)
```

Сейчас последние версии – pandas==2.0.0, numpy==1.24.2, scikit-learn==1.2.2

Обновить библиотеки до последней версии можно следующей командной в командной строке:

```
pip install -U pandas numpy scikit-learn
```

❗ **Примечание:** если вы работаете в Google Colab, то после обновления библиотек нужно перезапустить ноутбук. Делается это потому, что при запуске ноутбука колаб автоматически импортирует все стандартные библиотеки – а по умолчанию они в нём старой версии.

Если же версии последние, а ответы все равно не сходятся, то тогда нужно задать вопрос ментору в канале соответствующего модуля.

### ▼ Задание X выполняется очень долго. Можно как-то ускорить обучение модели?

Задания, требующие высокой производительности, можно решать в Google Colab на GPU.

### ▼ Загрузка файлов в Google Colab занимает очень много времени. Можно ли как-то загружать их быстрее?

Да, можно – через Google Drive и библиотеку PyDrive (предустановлена в колабе). Для этого вам нужно:

1. Загрузить ваш файл на гуглдиск.
2. В колабе импортировать все необходимые библиотеки:

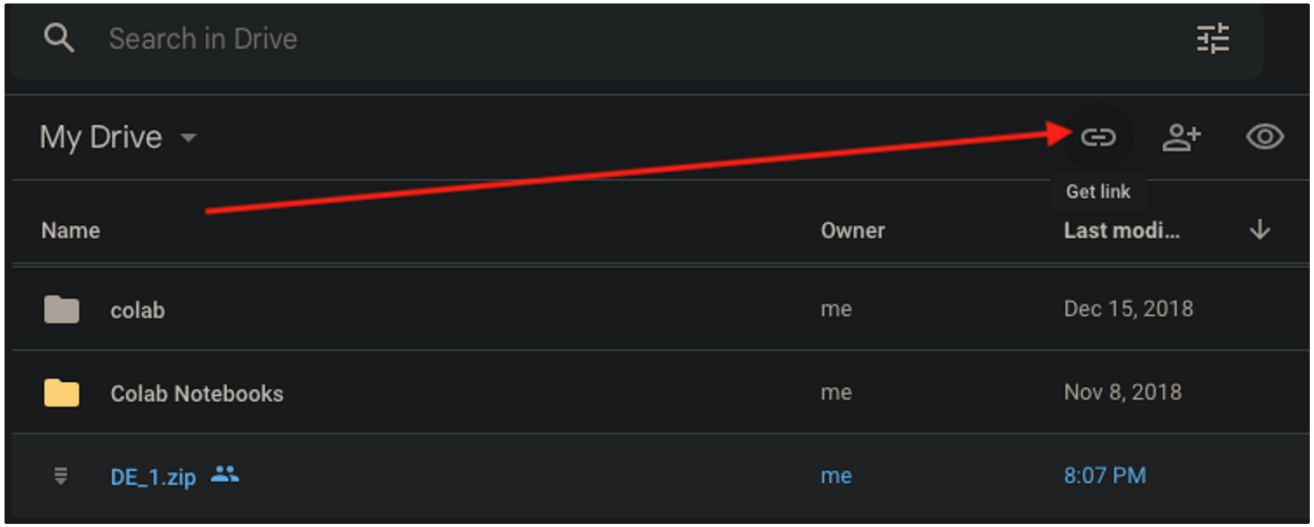
```
from pydrive.auth import GoogleAuth from pydrive.drive import GoogleDrive from
google.colab import auth from oauth2client.client import GoogleCredentials
```

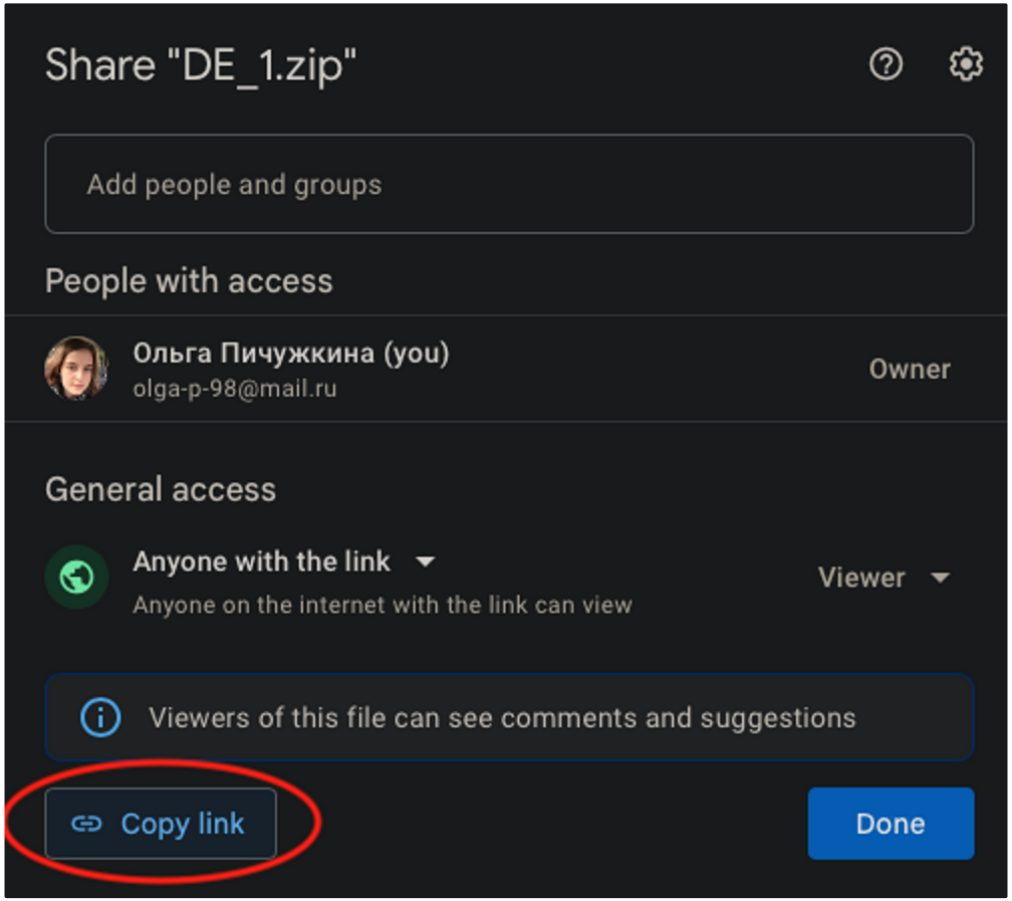
3. Привязать ваш гуглдиск к колабу:

```
auth.authenticate_user() gauth = GoogleAuth() gauth.credentials =
GoogleCredentials.get_application_default() drive = GoogleDrive(gauth)
```

(Выскочит окошко авторизации, там надо будет её подтвердить.)

4. Скопировать ссылку на ваш файл на гуглдиске – вот так:






5. Ваша ссылка будет выглядеть примерно так:

   https://drive.google.com/file/d/aaaaaa123456/view?usp=share_link

   Нам нужна только часть между /d/ и /view (здесь выделена красным) – это и будет id файла.

6. Загружаем файл, вставляя в строчку кода ниже наш id:

```
download = drive.CreateFile({"id": "aaaaaa123456"})
download.GetContentFile("test_csv")
```

Всё! Теперь наш файл лежит в колабе.

Может показаться слишком муторным проводить эти процедуры каждый раз перед началом работы в колабе. Но файлы так будут загружаться намного быстрее, чем вручную (вы же их один раз уже на гуглдиск загрузили). Можете сами попробовать и убедиться.

### ▼ У меня не ставится библиотека scikit-surpise на Windows. Что делать?

Рекомендуем сделать задание в Google Colab. С установкой scikit-surprise очень много проблем на Windows.

### ▼ Как лучше всего оформить README.md? Есть какие-то примеры или рекомендации?

Очень хорошо, что вы задумались об аккуратном оформлении readme!

Для начала – базовые вещи. Readme пишется на языке разметки Markdown. О нём вам рассказывали в бонусном модуле блока 1 (“Markdown и Git для создания портфолио”), но, если вы не проходили бонусный модуль или плохо его помните – ничего страшного. Вы уже сталкивались с языком Markdown, когда оформляли текстовые ячейки в юпитер ноутбуках.

Вот так выглядит текст на языке Markdown:

```markdown
# *MarkDown текст  в Jupyter Notebook (заголовок курсивным шрифтом)*

Обычный текст

**Текст полужирным шрифтом**

$ формула:  a_{i+1} = a_{i} + 10$

```

А так он отображается пользователю.

*MarkDown текст в Jupyter Notebook (заголовок курсивным шрифтом)*

Обычный текст

**Текст полужирным шрифтом**

формула : $a_{i+1} = a_i + 10$

Краткую шпаргалку по синтаксису Markdown можно посмотреть здесь.

Markdown-файлы, как и код, лучше всего создавать и редактировать в IDE – потому что IDE подсвечивает синтаксис Markdown.

А теперь о хороших практиках оформления readme. Вот здесь есть огромное количество примеров хороших readme, статей с рекомендациями по их составлению и готовых шаблонов. Изучайте, выбирайте, экспериментируйте, решайте, что вам подойдёт лучше. Успехов!

### ▼ Можно ли как-то регулировать отображение таблиц в pandas?

Да, можно. Можно настраивать максимальное количество строк/столбцов, которые могут отображаться в таблице, максимальную высоту/ширину таблицы, кодировку, форматирование чисел с плавающей точкой, формат дат и многое другое. Все это настраивается с помощью функции `pd.set_option` (обычно она вызывается в самом начале юпитер ноутбука). Например, вот так можно убрать ограничение на ширину таблицы (чтобы ячейки с объёмным содержанием отображались аккуратнее):

```
pd.set_option("display.max_rows", None)
```

Полностью посмотреть список настроек отображения таблиц можно, вызвав функцию `pd.describe_option("display")`. `"display"` в данном случае означает, что нам нужны настройки отображения таблиц, но вообще в pandas и многое другое настраивается. Подробнее о настройке поведения pandas можно почитать здесь.

Всё, что относится к ***стилям*** отображения таблиц (размер шрифта, сам шрифт, его цвет, цвет фона и многое другое) настраивается уже чуть-чуть сложнее (подробнее можно почитать здесь). Стиль есть не у всего ноутбука целиком, а у каждой таблицы по отдельности. С помощью метода df.style.set_table_styles можно задать атрибуты стиля – их названия и значения берутся из языка CSS (это язык описания внешнего вида веб-страниц). Например, вот так:

```
df.style.set_table_styles([ {"selector": "th", "props": [("font-size", "18px")]},
{"selector": "td", "props": [("font-size", "16px")]}, ])
```

В метод подаётся список из словарей, в каждом словаре хранится стиль, который применяется к каким-то определённым элементам. У каждого словаря есть ключи “selector” и “props”. Значение ключа “selector” – это то, к каким именно элементам применяется стиль (полный список доступных значений можно посмотреть здесь и здесь), а значение ключа “props” – это собственно список атрибутов стиля. Каждый атрибут – это кортеж из двух элементов: первый элемент название атрибута, а второй его значение. Полный список доступных атрибутов можно посмотреть здесь.

В примере выше мы установили для заголовков таблиц (тэг th) размер шрифта 18 пикселей, а для ячеек (тэг td) – размер шрифта 16 пикселей.

Также можно установить отдельные стили для отдельных столбцов. Для этого в метод нужно подать словарь, в котором ключами будут названия столбцов, а значениями – стили:

```
df.style.set_table_styles({ # красим столбец A в синий цвет "A": ["selector": "",
props=[("background-color", "blue")]], # а столбец B – в красный "B": ["selector":
"", props=[("background-color", "red")]], })
```

Чтобы сделать то же самое для строк, необходимо задать значение аргумента `axis=1`. Например, вот так можно сделать, чтобы при наведении курсора мыши на строку 5, размер шрифта в ячейках увеличивался до 25 пикселей:

```
df.style.set_table_styles([{'selector': 'td:hover', 'props': [('font-size',
'150%')]}], axis=1)
```

### Можете порекомендовать полезную литературу по ML для дальнейшего развития?

☐ в процессе работы

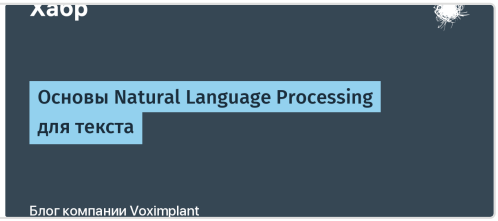
### Можете порекомендовать полезную литературу по NLP для дальнейшего развития?

Вот хорошая вводная статья по NLP:

Основы Natural Language Processing для текста

Обработка естественного языка сейчас не используются разве что в совсем консервативных отраслях. В большинстве технологических

https://habr.com/ru/company/Voximplant/blog/446738/




Если же вам нужно что-то посложнее по современным методам NLP, то рекомендую это список лекций:

CMU Advanced NLP 2022

https://www.youtube.com/playlist?list=PL8PYTP1V4I8D0UkqW2fEhgLr...

И отдельно по трансформерам (архиктектура нейросетей, чаще всего применяющаяся в задачах NLP) есть очень хорошая книга Transformers for Natural Language Processing (с примерами в ноутбуках).