Как я писал курсовую

Приступая к работе

Начало было обычным, ничего не предвещало беды, я спокойно скачал на свой мак 25гб датасета. Потом с помощью Dask я открыл этот датасет, склеил как-то. Импортировал из Dask ML лог. регрессию как базу, с чего стоит начать обучение, а она ругается, что некоторые кусочки датафрейма пустые ! Да, я не правильно склеил данные, но узнал об этом уже позднее. Я превратил этот датафрейм в пандасовский и разбил на четыре части, каждая по гигабайту, но уже в другой тетрадке, потому что 80гб свободного места на накопителе мне не хватило на это.

Обучение

Я не спроста разбил датасет на части, как я узнал об этом позже. Начал я в третьей тетрадке обучать деревья первого датасета и как-то это затянулось на 5 часов! Я несколько дней потратил на то, чтобы подобрать подходящие гиперпараметры для модели. Я взял обычные деревья решений с заданным параметром class_weight, потому что датасет дисбалансный. Но так как моделей, как и датасетов будет четыре, их предсказания (predict_proba) я усредню… Так я думал, пока не приплюснул все датасеты до 3 признаков и не взглянул на них. На scatterplot-е видно, что данные поделены на островки и на одном датасете там есть островок, а в остальных там пустота, поэтому усреднять предсказания всех деревьев не вариант. И поэтому я написал ансамбль с конкурсом. Побеждает та модель, чья вероятность первого класса наивысшая. Это должно поднять Recall. Поднимать надо именно Recall, а не Precision, потому что с каждого верно угаданного первого класса мы получаем real money, а с каждого неверно угаданного нулевого класса мы ничего не потеряем. И я еще довольно удачно подобрал аргументы каждого отдельного дерева. Recall там больше, чем Precision на тесте из трех других датасетов.

Качество

На десерт метрики. Вот F1 равный 0.5 это простое угадывание одного из двух на основе псевдослучайных чисел. А у меня… А у меня F1 macro = 0.57 на предсказаниях ансамбля из трех моделей на четвертом датасете.

Подготовка к запуску

Затем все это предстояло завернуть в пайплайн. Я схалтурил один большой пайплайн из множества маленьких, попытался сериализовать в pickle, а он не умеет в глубокое копирование и самодельные трансформеры тоже не сохраняет. Тьфу на него! Лучше заверну все пайплайны в скриптик, который запускается из под терминала и который использует уже не три модели, а все четыре.

Как запускать

Теперь инструкция, как все это запускать:

1. Открываем терминал в папке с кодом
2. Создаем папку data, в неё запихиваем датасет
3. Устанавливаем необходимые пакеты командой pip `install -r requirements.txt`
4. Запускаем скрипт `python3 predict.py`
5. Ждем 3-5 минут, подаем на стол!