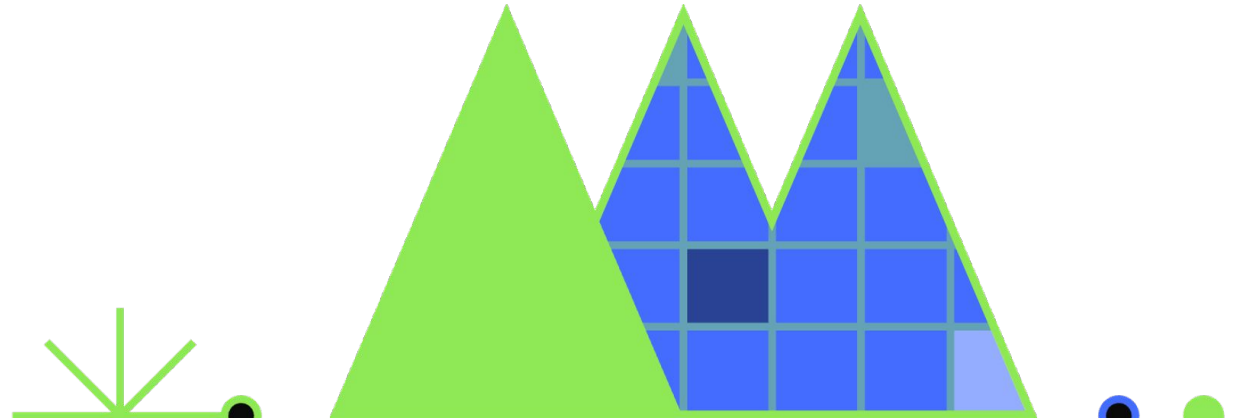
Тегирование тарифов

Подготовил презентацию:
Евгений Иванкин



Обо мне

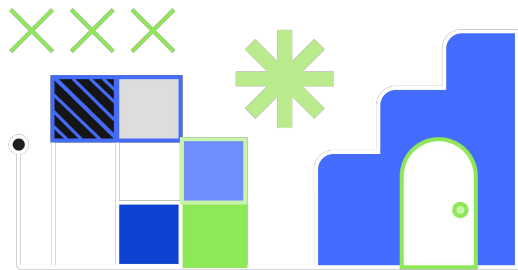
- Студент 4-го курса в Иннополисе
- Трек обучения - Data Science
- Разработчик в ООО “Инногеотех”
- Любитель котов

Проблемы с данными

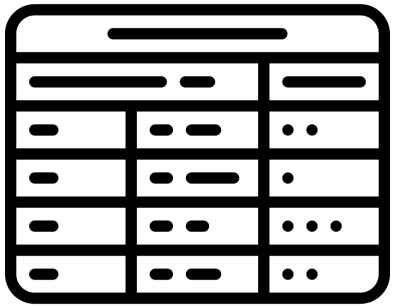
- Дубликаты
- Не все ожидаемые классы есть в данных и представлены в достаточном количестве
- Встречаются описания не на английском языке: на русском, французском, испанском и турецком
- Странные записи в одном из наборов: вместо значений - названия колонок
- Пустые описания

Зачем нам чистить дубликаты?

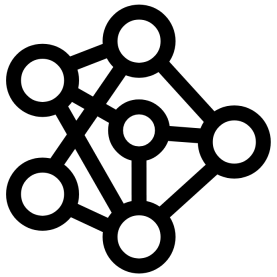
- Позволит нам понять, сколько действительно разных примеров у нас есть
- Так мы будем уверены, что у нас никакие записи из обучающей выборки не “утекли” в валидационную, и мы можем доверять значениям метрик
- Ещё раз про метрики: мы получим более оптимистичные значения, если у нас в валидационной выборке будут повторы, даже если они не пересекаются с обучающей



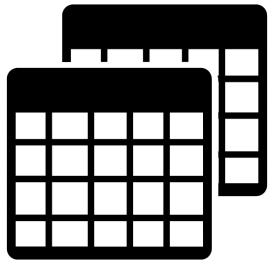
Синтетические данные



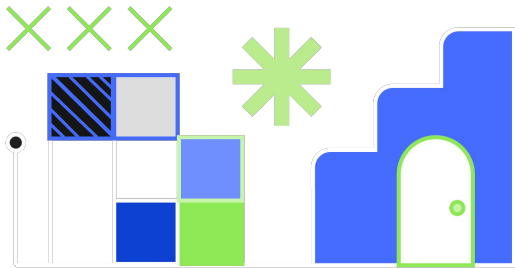
Берём удачные примеры из исходных данных



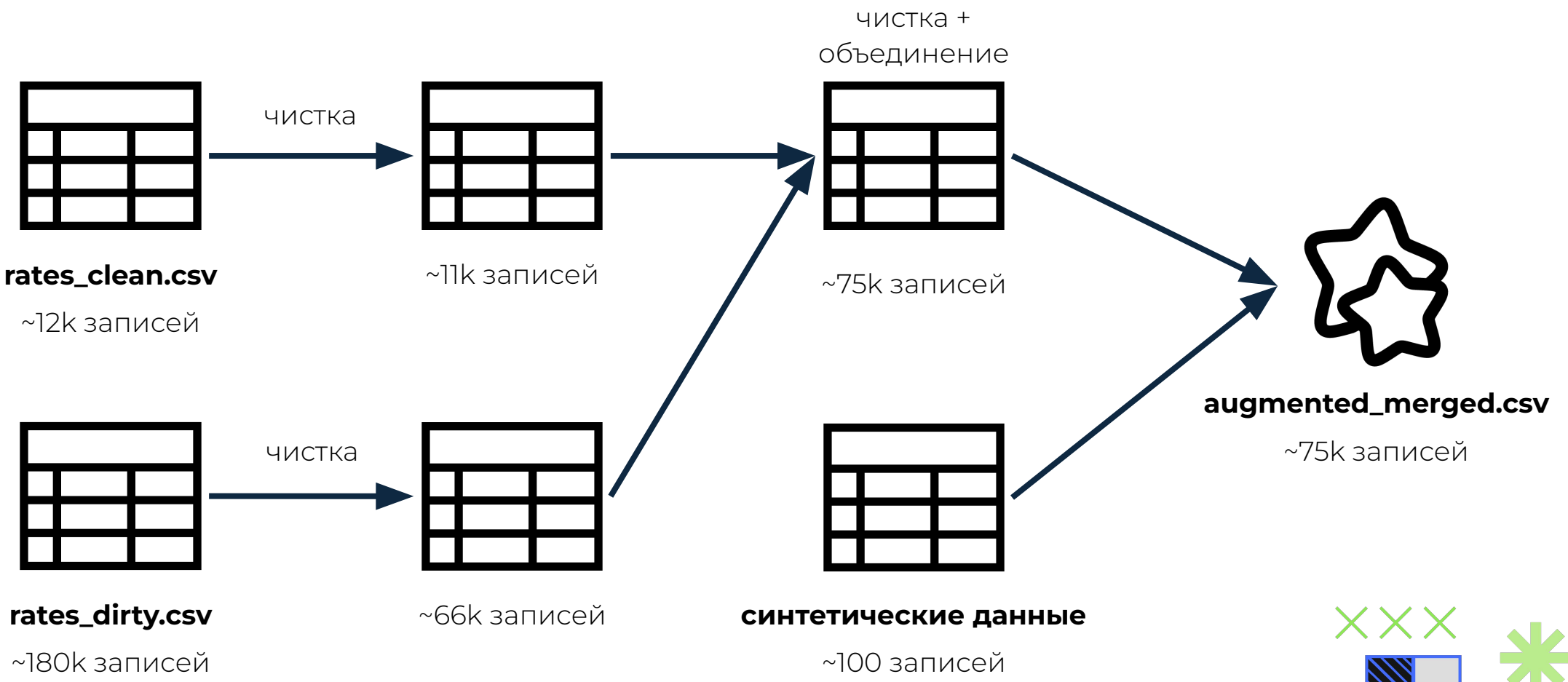
LLM на их основе генерирует примеры для нужных нам классов и языков



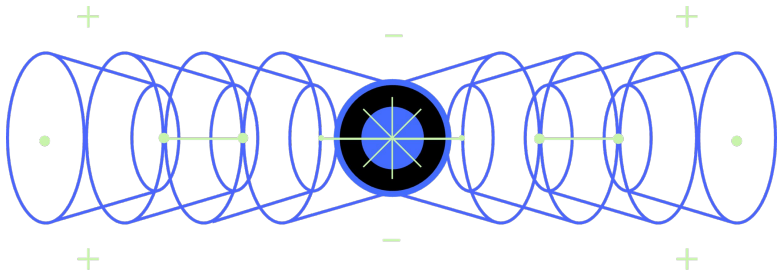
Убираем некорректные примеры, добавляем оставшиеся записи ко всем данным



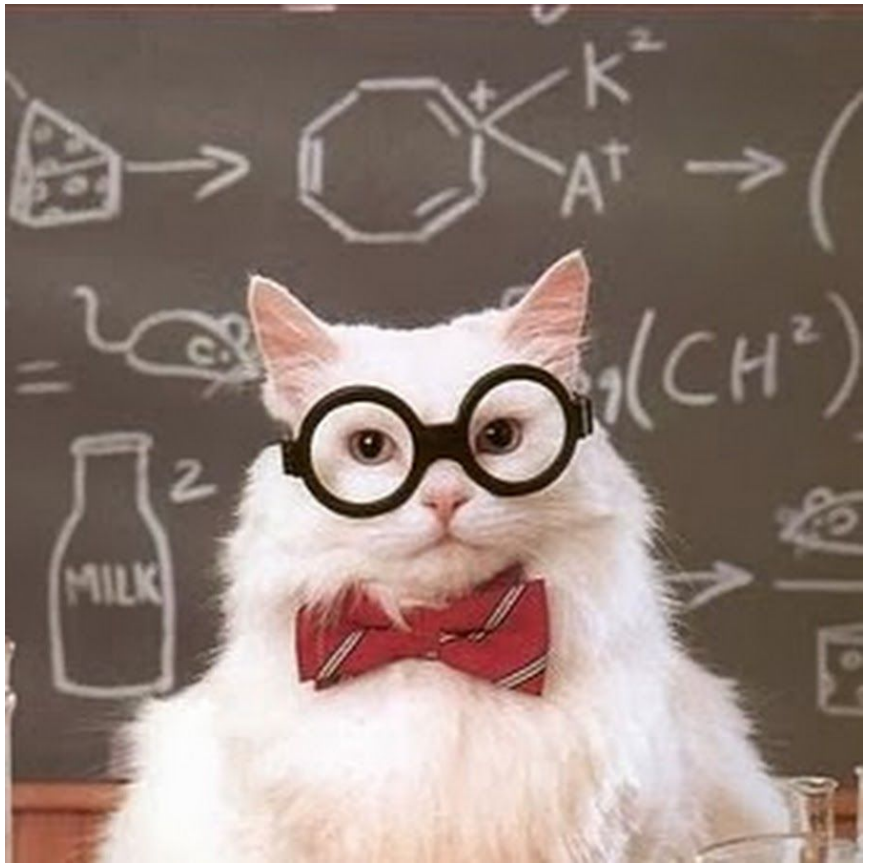
Что в итоге с данными?



Обучаем модели



- Для решения задачи были выбраны простые модели: они должны показать достаточно хорошую точность при сохранении высокой скорости
- В качестве baseline используется DummyClassifier выдающий в качестве предсказаний самый часто встречаемый вариант
- Для каждого предсказываемого признака обучалась своя модель, данные делились с соотношением 80/20 на обучающую и валидационные выборки со стратификацией по классам

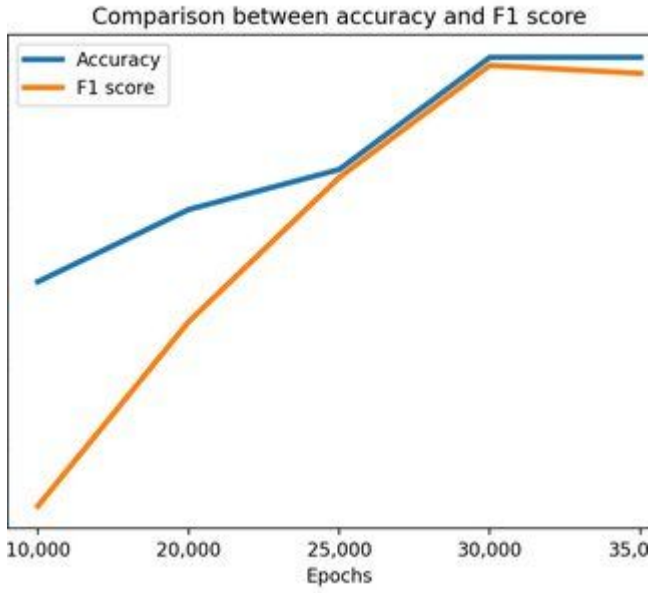


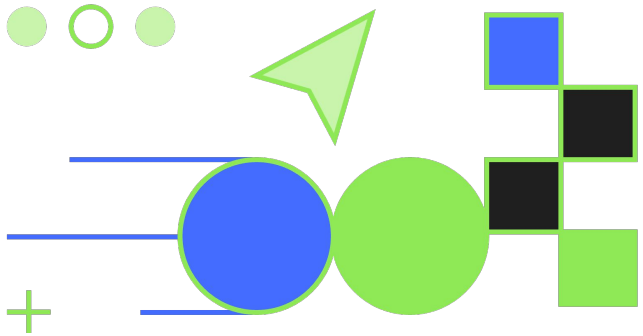
Сравниваем качество с F1 Macro

Classifier	class	quality	bathroom	bedding	capacity	club	balcony	view
DummyClassifier(strategy='stratified')	0.07	0.05	0.25	0.20	0.14	0.50	0.50	0.03
DummyClassifier(strategy='most_frequent')	0.06	0.03	0.25	0.15	0.11	0.50	0.49	0.03
ComplementNB()	0.80	0.89	0.53	0.38	0.59	0.55	0.71	0.65
RidgeClassifier()	0.97	0.94	0.91	0.51	0.83	0.83	0.96	0.80
LinearSVC()	0.99	0.98	0.99	0.63	0.89	0.89	0.98	0.90

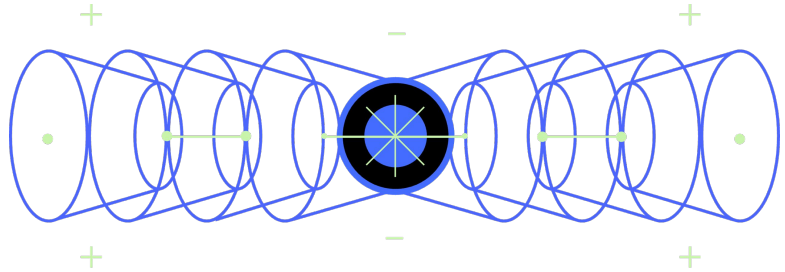
Почему F1, а не Accuracy?

- Из-за дисбаланса классов мы бы получили более оптимистичные результаты оценки, если использовали accuracy, сильнее всего улучшились бы результаты у DummyClassifier
- Мы бы получили более близкие по метрикам результаты, нам было бы сложнее сравнивать модели

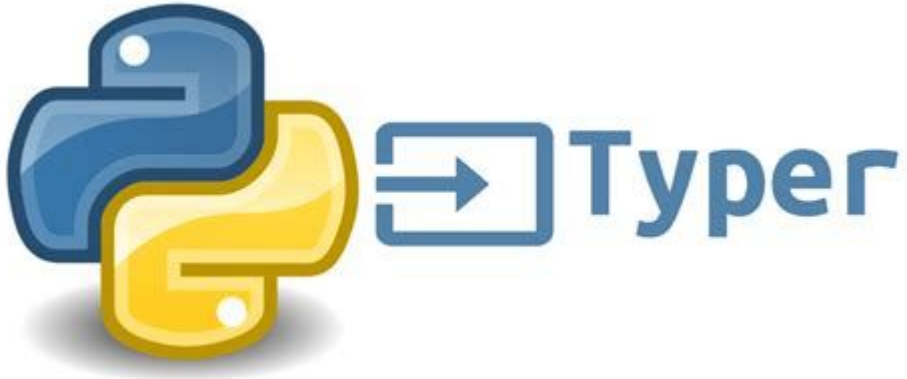




Модели в проде: CLI



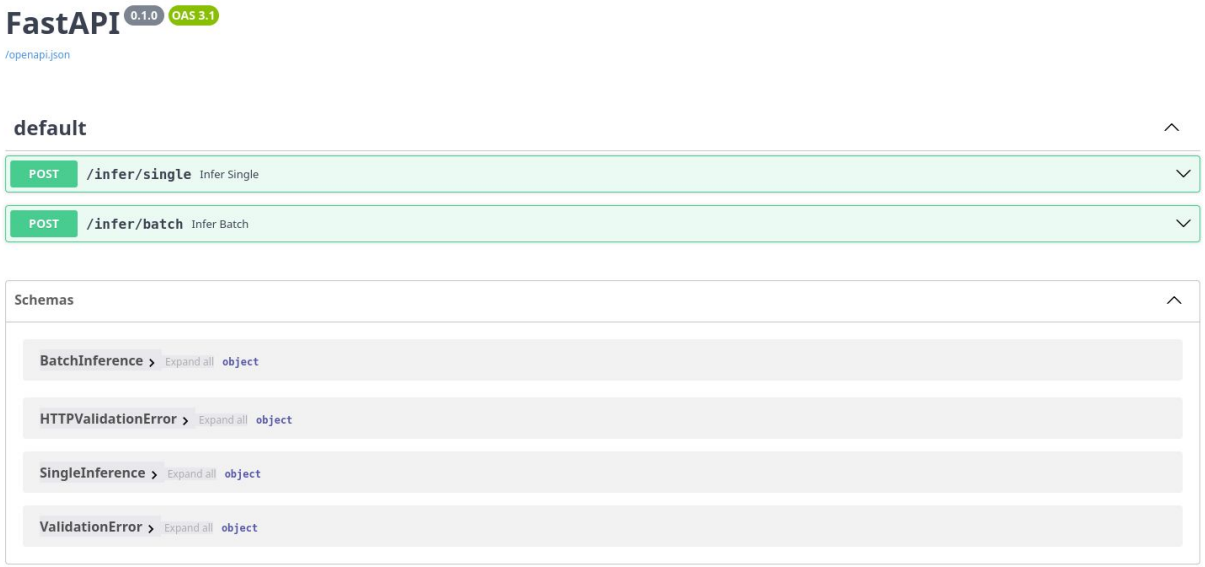
- При передаче большого количества данных за раз, решение обрабатывает около 26 тысяч записей в секунду
- Решение позволяет обновлять модели, в т. ч. только для одного признака, не меняя код



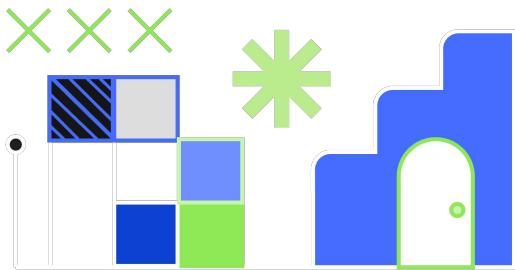

```
└─$ python -m cli_app --help

 Usage: cli_app.py [OPTIONS]

 Options
 *  --content                       PATH                          A path to rates CSV file
                                                                  [default: None]
                                                                  [required]
    --model                         [svc_pipeline|ridge_pipeline] [default: svc_pipeline]
    --install-completion                                          Install completion for the current
                                                                  shell.
    --show-completion                                             Show completion for the current
                                                                  shell, to copy it or customize the
                                                                  installation.
    --help                                                        Show this message and exit.
```

Модели в проде: Web API

- Предоставляет ручки для обработки одного запроса и батча из нескольких
- Модели предварительно загружаются при старте сервера
- По результатам нагрузочного тестирования, при обработке батчей достигается необходимая пропускная способность в 3000-4000 запросов



FastAPI



Вопросы?