

ЛИДЕРЫ
ЦИФРОВОЙ
ТРАНСФОРМАЦИИ

Модель по построению оптимальных маршрутов
инкассации платежных терминалов

Algorithmistic Heist



ГАЗПРОМБАНК

Команда



Вольнов Сергей
Арсеньевич
Тг: @sergak0
Капитан, DS/Algo



Червов Никита
Романович
Тг: @chervovn04
Algorithm Master



Ильин Павел
Андреевич
Тг: @Pashteticus
Data Scientist

Решение

Концепт решения



Предсказание
притока наличности

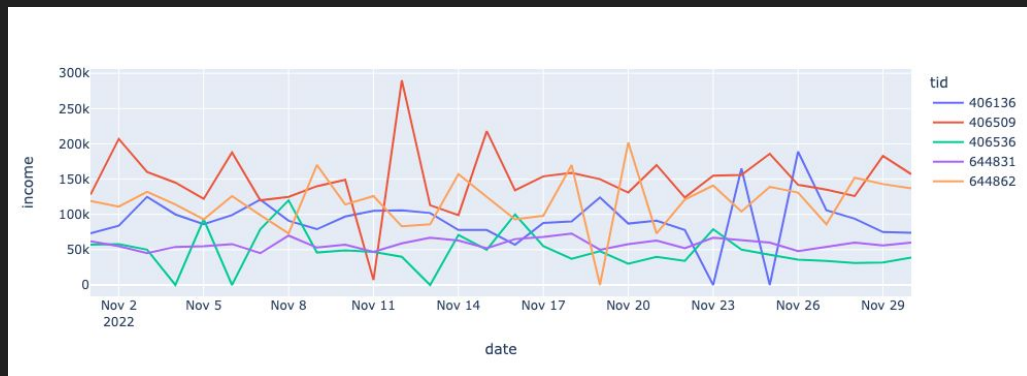


Выбор терминалов
для объезда

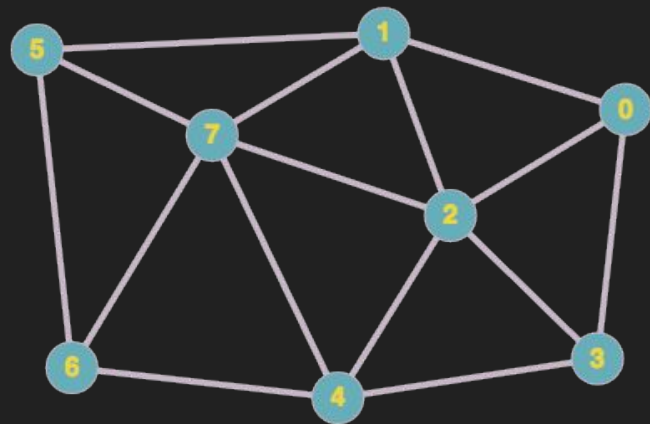


Построение
оптимальных путей

Данные



Ежедневный прирост
наличности в терминалах

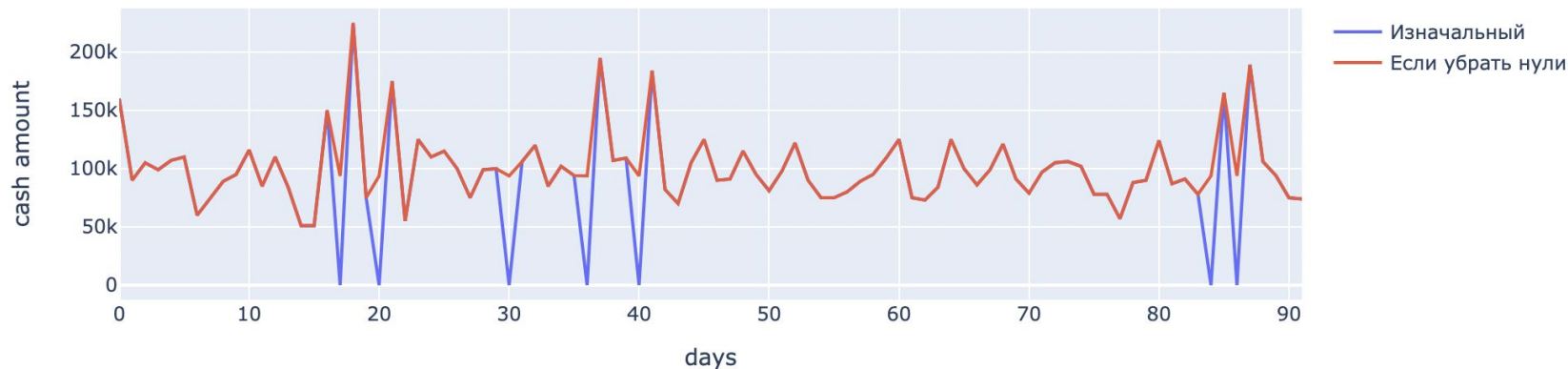


Граф расстояний между
вершинами

Предсказание объема наличности

Идея

- Предсказываем, когда терминалы не работают (в какие дни будут нули)
- В остальных днях используем среднее значение по данному терминалу



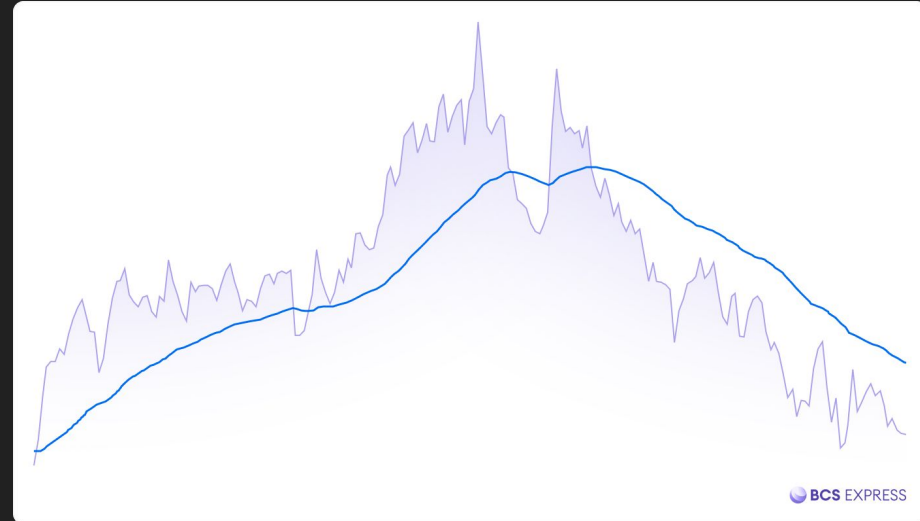
Временные признаки

- День недели
- Число
- Месяц
- Является ли праздником
- Расстояние до ближайшего праздника



Лаги/Скользящие средние

- Скользящие средние
- Лаги
- Экспоненциально взвешенные средние
- Окна: 2, 3, 7, 14 и 28 дней



Статистика по поломкам

- Кол-во поломок по терминалам
- Частота поломок
- Агрегации по времени между поломками
- Время до последней поломки
- Кол-во поломок в последнее время



Модель

```
cat_cols = ['tid']

params_cat = {'n_estimators' : 1000,
              'learning_rate': .03,
              'depth' : 4,
              'use_best_model': True,
              'cat_features' : cat_cols,
              'random_state': 42,
              'eval_metric' : 'AUC'}

cat_model = CatBoostClassifier(**params_cat)
```

Метрики прогнозирования притока денег

Предсказание нулей:

Precision = **0.80**

Recall = 0.28

Предсказание притока наличности:

MAE = **12400**

MAPE = 0.24

Построение оптимальных маршрутов

За что платим деньги?

- сумма процентов в неинкассированных терминалах
- стоимость обслуживания терминалов
- **стоимость броневиков**

Замечание: основную стоимость составляет аренда броневиков, так что их оптимизируем в первую очередь.

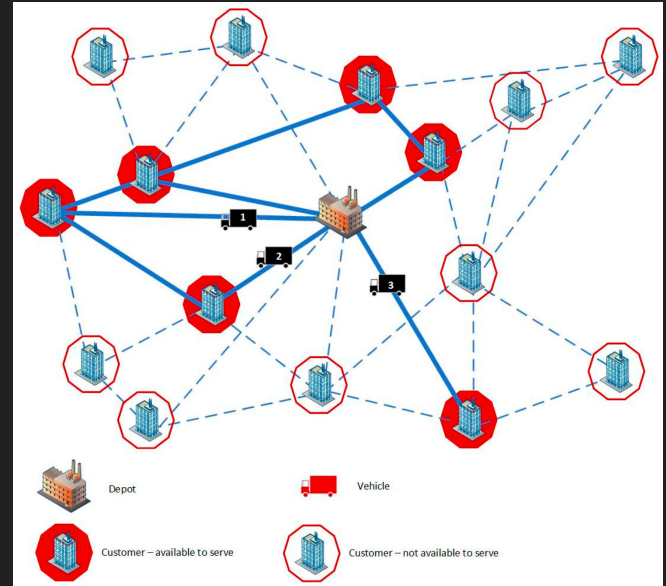


Vehicle Routing Problem with Costs

Задача: объехать вершины графа, так чтобы суммарный штраф (cost) за непосещенные вершины был наименьшим и каждая машина ездил не больше max_time времени.

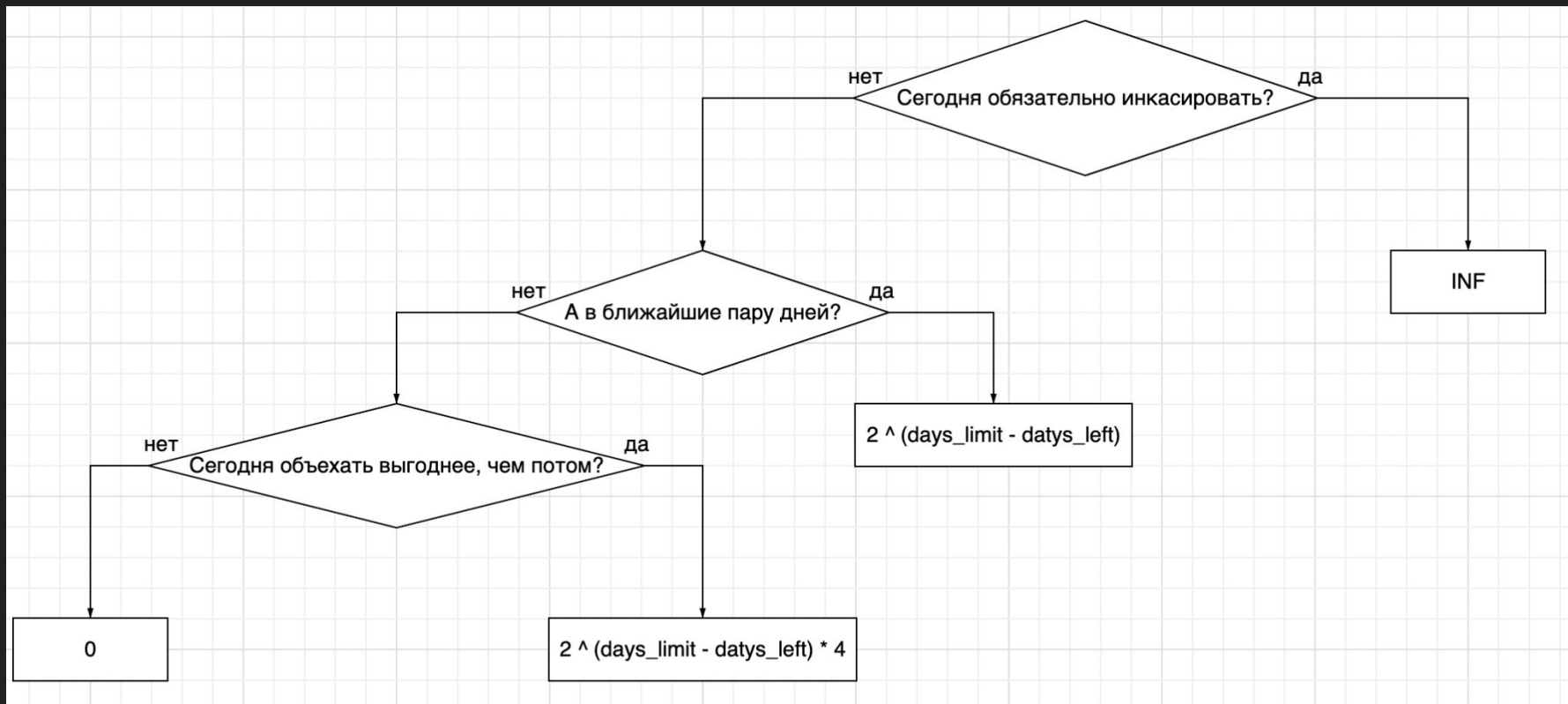


Google OR-Tools



Считаем штрафы для вершин

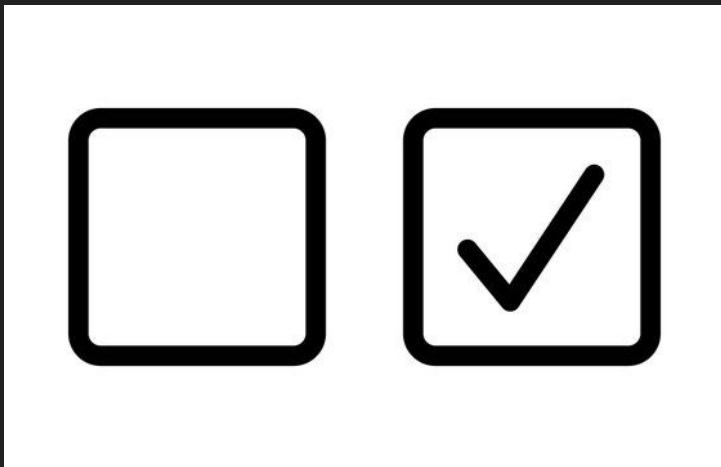
Какая логика за этим стоит?



Определяем выгодно ли сегодня инкасировать

Отдельно для каждого терминала хотим определить выгодно ли нам его сегодня объезжать или нет.

$\text{delta_loss} = \text{smallest_use_expenses} - \text{smallest_not_use_expenses}$



Два подхода

Отдельно для каждого терминала хотим определить в какие дни его оптимально инкассировать, чтобы потерять как можно меньше.

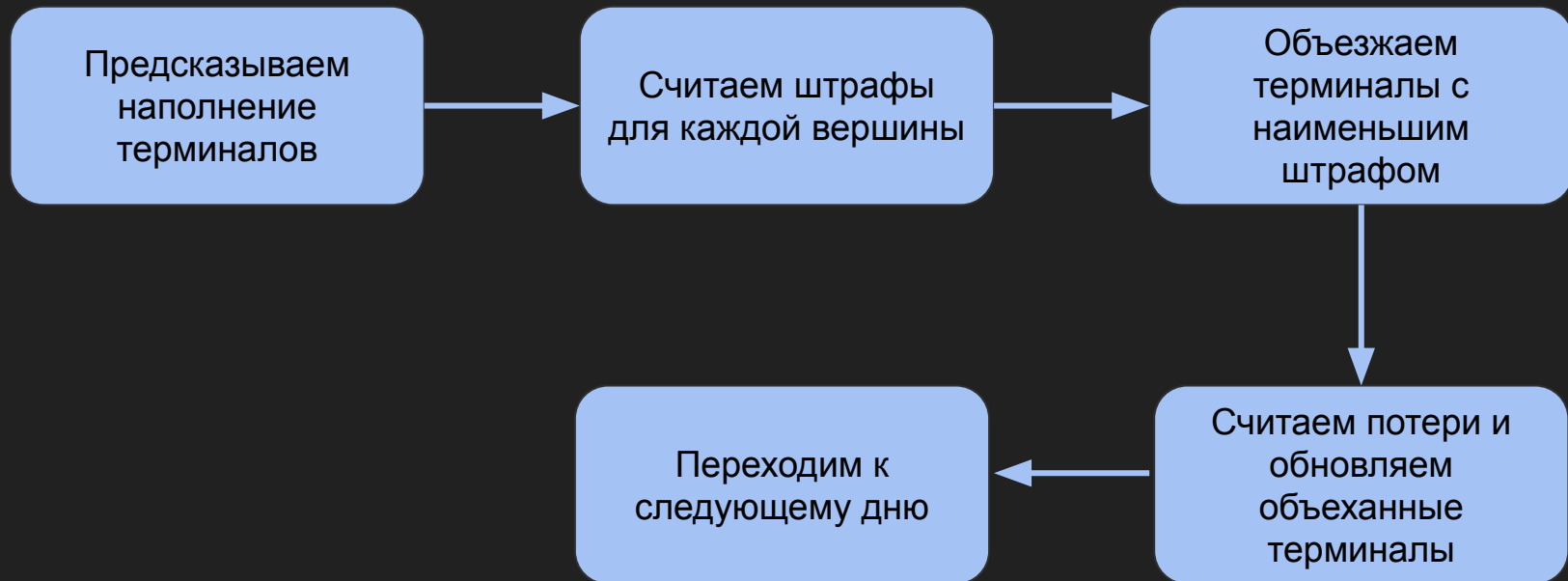
1) Простой, но долгий способ: перебрать $2^{(\text{num_days})}$ вариантов.

Время работы: $O(\text{num_days} * 2^{\text{num_days}})$

2) Сложнее писать, но сильно быстрее (остановились на нем): применить динамическое программирование.

Время работы: $O(\text{days_limit} * \text{num_days}^2)$

Общий пайплайн



Финальные метрики за 3 месяца

- Суммарная стоимость фондирования = 2.22M
- Суммарная стоимость инкассации терминалов = 1.35M
- Кол-во используемых броневиков = 4
- Стоимость аренды броневиков на все дни = 7.28M
- Общие расходы = 10.85M



Масштабируемость

- Время создания предсказания на один день ~30 сек.
- Время работы алгоритма растет полиномиально в зависимости от размера входных данных.
- Весь код задокументирован и легко воспроизводим.



Что еще пробовали

- Генетические алгоритмы
- Метод отжига
- Optuna



Спасибо за внимание!