AI YP24 Команда 11

Тема: Выявление аномалий в данных

Руководитель:
Кирилл Малюшитский

Команда

- Егошин Юрий (@Yuri_Dmitrievich)
- Глеб Лысенко (@glebly)
- Силиневич Илья (@uJlbI0IILuH)
- Цыбакова Ольга (@olgasub57)

О проекте

Цель

- Реализовать сервис для определения фрода в транзакционных данных (банковские карты, USA)

Задачи

- Научиться проводить анализ данных и обучение модели на практике
- Реализовать сервис для определения фрода и задеплоить
- Определить ключевые пороговые метрики и сравнить качество модели на основе этих метрик
- Рассчитать бизнес ценность

Этапы проекта

- [x] Выбрать Dataset, провести EDA,
- [x] ML, выбрать и обучить наиболее подходящую модель
- [x] Реализовать сервис (BE API + FE) и развернуть его
- Усовершенствовать модель, внедрение DL-архитектуры

Датасет

Исходный датасет

	Unnamed: 0	trans_date_trans_time	cc_num	merchant	category	amt	first	last	gender	street	...	long	city_pop	job	dob	trans_num	unix_time	merch_lat	merch_long	is_fraud	merch_zipcode
0	0	2019-01-01 00:00:18	2703186189652095	fraud_Rippin, Kub and Mann	misc_net	4.97	Jennifer	Banks	F	561 Perry Cove	...	-81.1781	3495	Psychologist, counselling	1988-03-09	0b242abb623afc578575680df30655b9	1325376018	36.011293	-82.048315	0	28705.0
1	1	2019-01-01 00:00:44	630423337322	fraud_Heller, Gutmann and Zieme	grocery_pos	107.23	Stephanie	Gill	F	43039 Riley Greens Suite 393	...	-118.2105	149	Special educational needs teacher	1978-06-21	1f76529f8574734946361c461b024d99	1325376044	49.159047	-118.186462	0	NaN

Размер исходного датасета

```
data.shape
```

```
(1296675, 24)
```

```
Data columns (total 24 columns):
 #   Column                 Non-Null Count    Dtype
---  ------                 --------------    -----
 0   Unnamed: 0             1296675 non-null  int64
 1   trans_date_trans_time  1296675 non-null  object
 2   cc_num                 1296675 non-null  int64
 3   merchant               1296675 non-null  object
 4   category               1296675 non-null  object
 5   amt                    1296675 non-null  float64
 6   first                  1296675 non-null  object
 7   last                   1296675 non-null  object
 8   gender                 1296675 non-null  object
 9   street                 1296675 non-null  object
 10  city                   1296675 non-null  object
 11  state                  1296675 non-null  object
 12  zip                    1296675 non-null  int64
 13  lat                    1296675 non-null  float64
 14  long                   1296675 non-null  float64
 15  city_pop               1296675 non-null  int64
 16  job                    1296675 non-null  object
 17  dob                    1296675 non-null  object
 18  trans_num              1296675 non-null  object
 19  unix_time              1296675 non-null  int64
 20  merch_lat              1296675 non-null  float64
 21  merch_long             1296675 non-null  float64
 22  is_fraud               1296675 non-null  int64
 23  merch_zipcode          1100702 non-null  float64
dtypes: float64(6), int64(6), object(12)
```

Проведена предобработка данных:

- удалили столбцы Unnamed: 0, first, last
- перевели столбец trans_date_trans_time в datetime
- разделили общую датувремя на год месяц день и отдельно время
- заполнили в столбце ‘merch_zipcode’ пропущенные значения -1
- рассчитали возраст - столбец ‘age’

EDA

- **Сильный дисбаланс классов**

is_fraud	proportion
0	99.421135
1	0.578865

count

morning	midday	evening	night	
0	0	1	0	4133
		0	1	2633
	1	0	0	466
1	0	0	0	274

- пик мошенничества - между 18 и 24 часами и также с 00 часов до 06 утра

EDA

- **сколько в среднем теряется денег за одну мош. тр. и медиана потери**

Признак `amt`

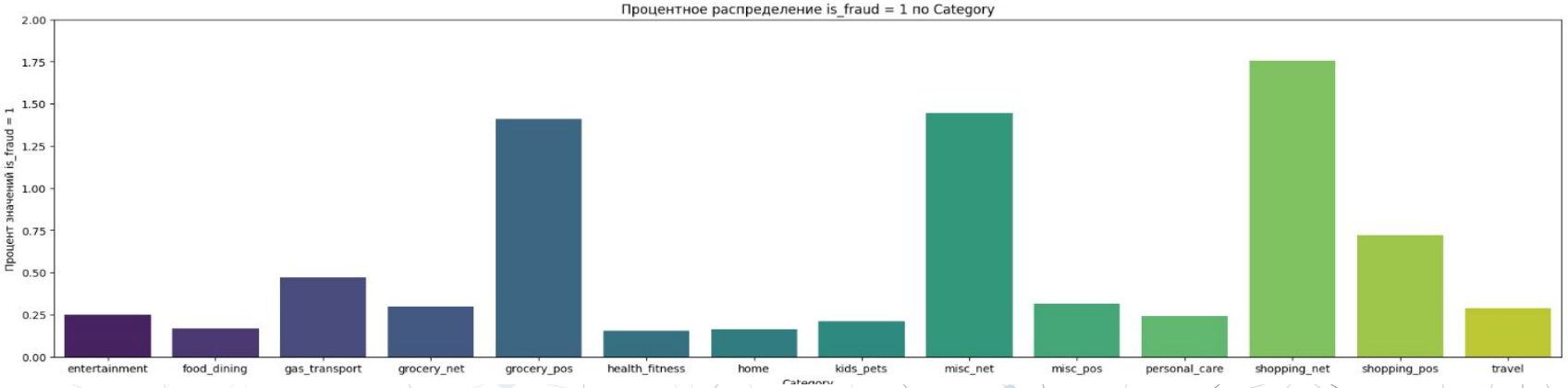
```
[ ]  data[data['is_fraud'] == 1]['amt'].mean() # сколько в среднем теряется денег за одну мош. тр.
```

```
531.3200919264589
```

```
data[data['is_fraud'] == 1]['amt'].median() # медианное значение за одну мош.тр.
```

```
396.505
```

- большинство мошенничества - бакалейные товары, сетевых магазинов



EDA

Финальный датасет

	cc_num	merchant	category	amt	gender	street	city	state	zip	lat	long	city_pop	job	dob	merch_lat	merch_long	is_fraud	merch_zipcode	name	trans_year	trans_month	trans_day	trans_time	age
0	2703186189652095	fraud_Rippin, Kub and Mann	misc_net	4.97	F	561 Perry Cove	Moravian Falls	NC	28654	36.0788	-81.1781	3495	Psychologist, counselling	1988-03-09	36.011293	-82.048315	0	28705.0	Jennifer Banks	2019	1	1	00:00:18	36
1	630423337322	fraud_Heller, Gutmann and Zieme	grocery_pos	107.23	F	43039 Riley Greens Suite 393	Orient	WA	99160	48.8878	-118.2105	149	Special educational needs teacher	1978-06-21	49.159047	-118.186462	0	-1.0	Stephanie Gill	2019	1	1	00:00:44	46

Размер финального датасета

```
data.shape
```

```
(1296675, 24)
```

Лидерство мошеннических операций по штатам

state	amt
NY	0.000428
TX	0.000369
PA	0.000353
CA	0.000251
OH	0.000248
FL	0.000217
IL	0.000191
MI	0.000184
AL	0.000166
MN	0.000160

Ключевые метрики

- **F1** - баланс между recall и precision - так как сильный дисбаланс классов
- **Целевой показатель F1>= 0.82**

- **Recall** - *максимальное количество мошеннических операций*
  Целевой показатель Recall >= 0.85

- **Precision** - точность *выявления мошенничества*
  Целевой показатель Precision >=0.80

Baseline, Метрики, расчет

Применили модель LogisticRegression
как базовую, так и с гиперпараметрами:
balanced, penalty: l2, C: 0.01, solver: sag, liblinear

Модель	Гипрепараметры	F1	Precision	Recall	ROC_AUC	Businees metric	Profit Percentage, %
LogisticRegression		0.24	0.74	0.14	0.90	3 236 176 000	124.78
LogisticRegression	balanced	0.089	0.05	0.83	0.95	2 962 146 000	114.22
LogisticRegression	balanced, penalty: l2, C: 0.01, solver: sag	0.089	0.047	0.83	0.95	2 962 122 000	114.22
LogisticRegression	balanced, penalty: l1, C: 0.01, solver: liblinear	0.089	0.047	0.83	0.95	2 962 154 000	114.22

Выбор лучшего результата

- Лучшие метрики на модели LogisticRegression без применения каких-либо параметров
  F1 = 0.24, Precision = 0.74, Recall = 0.14 - но это низкие показатели.

- SMOTE:
  F1 = 0.11, Precision = 0.06, Recall = 0.81 - но это низкие показатели.

- Перебор порога для более высоких показателей метрик:

Лучший порог: 0.15,
F1-Score: 0.44
Precision (Validation): 0.55
Recall (Validation): 0.36
Precision (Test): 0.58
Recall (Test): 0.39

F1 - score недостаточно высокая

- Лучший показатель F1 = 0.44

Бизнес метрики

В модель добавлены 2 бизнес метрики для оценки экономической эффективности модели:

1) **Операционная маржа в результате выявления мошеннических операций с помощью модели**

выручка затраты

$$ОМ = (T - TP) * K - A * (TP + FP),$$

T - число всех транзакций
TP - количество истинно полож. результатов
K - комиссия от 1й транзакции
A - затраты на работу с выявленными моделью показателями мошеннических операций (арбитраж)
FP - количество ложноположительных результатов

2) **Маржинальность выявления моделью мошеннических операций**

$$ОМ\_\% = ОМ / (T - TP) * K$$

Измеряется в % и отражает эффективность определения моделью мошеннических операций.

Структура проекта

1. backend/:

Содержит API-логику на базе **FastAPI**, включая обработку данных, модели и маршруты:

- `data/`: хранит входные данные (основной CSV-файл для обучения)
- `main.py`: главный файл запуска приложения
- `log_config.py`: файл с настройками логгера
- `preprocessing_data.py`: функция предобработки данных
- `sub_functions.py`: дополнительные функции для работы API
- `Dockerfile`: инструкция для сборки контейнера
- `logs/`: хранение логов backend

2. frontend/:

Содержит Streamlit-приложение для визуализации:

- `logs/`: хранит логи для анализа ошибок и запросов
- `app.py`: основной файл для запуска приложения
- `config.py`: файл, содержащий настройки проекта
- `Dockerfile`: инструкция для сборки контейнера

3. docker-compose.yml:

Описывает сборку и запуск обоих сервисов (backend и frontend)

4. Сопроводительные файлы README.md и report.pdf:

- Описывают структуру проекта и способы его запуска
- Показывают, что должен получить пользователь при работе с проектом

Архитектура проекта

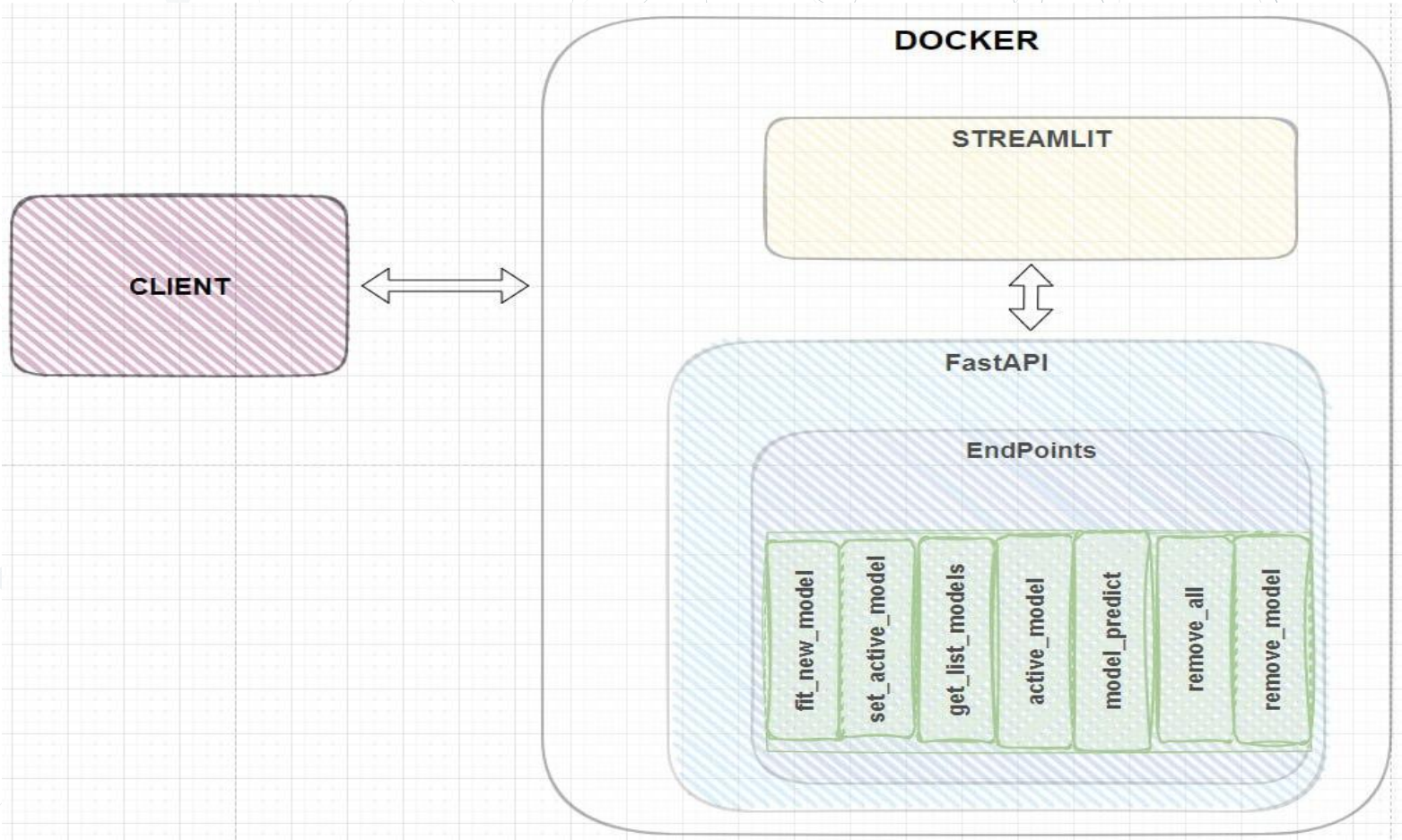
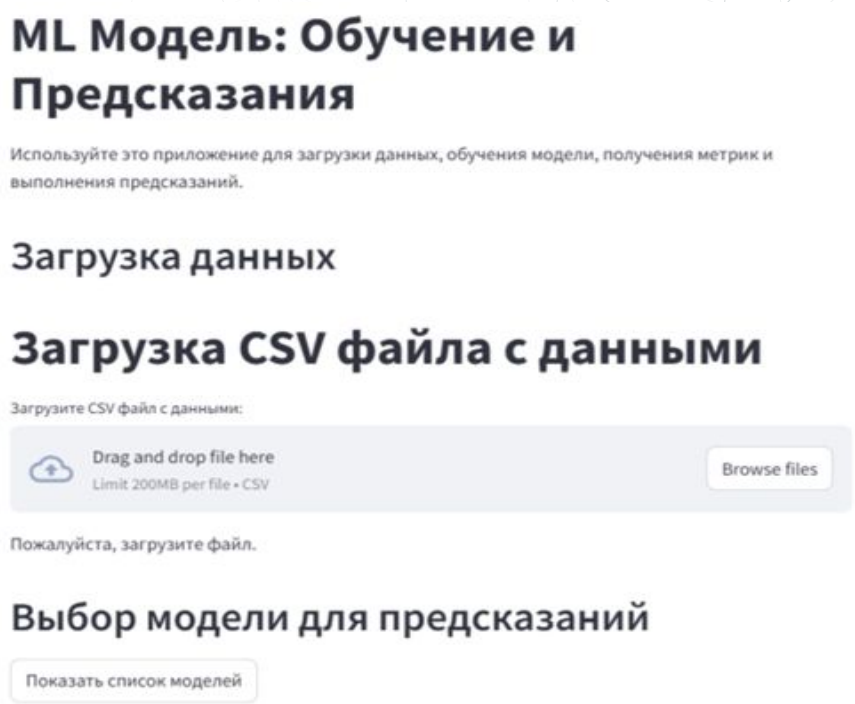


Схема работы с сервисом

Рис1. Схема сервиса

Фронт (Streamlit)



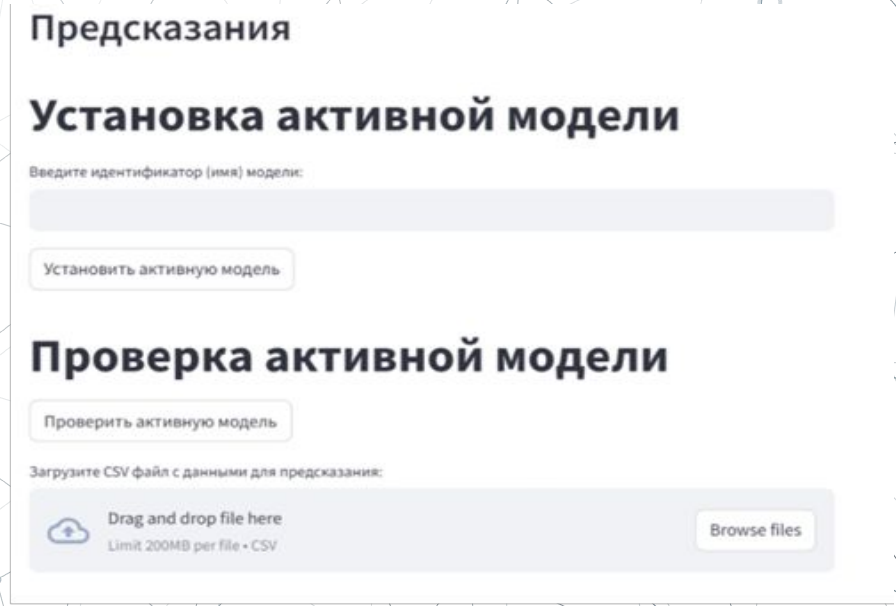


Рис2. Streamlit интерфейс

Развертывание и эксплуатация

Проект развернут на VPS с конфигурацией: 4 x 3.3 ГГц CPU • 8 ГБ RAM • 80 ГБ NVMe с публичным ipv4

- http://193.160.208.32:8501/ для работы с интерфейсом Streamlit
- http://193.160.208.32:8000/docs#/ для взаимодействия с API

ОС - Ubuntu 18.04, установленный Docker и Docker Compose

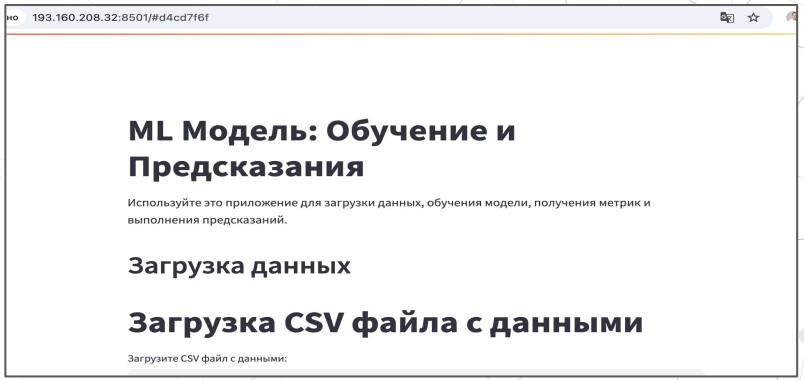
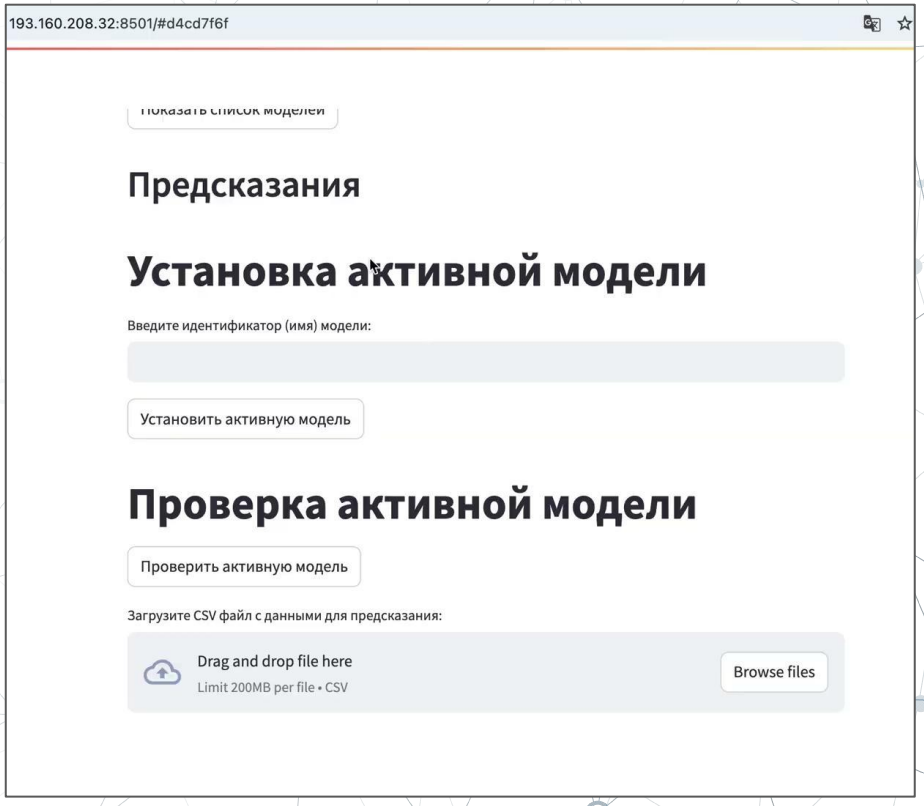


Рис3. Интерфейс Streamlit

Рис4. Серверная часть

Демо работы сервиса



Скринкаст. Демонстрация работы сервиса через интерфейс

Распределение выполненных работ

1. Выбор датасета и определение задачи - вместе
2. EDA, подготовка, пропуски, анализ данных, ML - вместе
3. Разработка BE, FE, доработка ML, развертывание - вместе
4. Презентация - вместе

Бизнес ценность

Для транзакционного банковского бизнеса

Маржа = Количество транзакций (Т) х Комиссия (К) - Затраты (А)

Ценность решения:

- снижение потерь на фрод операциях
- снижение рисков для клиентов (мерчантов) банка (имидж)

Расчет

- 0.5% от общего объема транзакций - фрод
- 530$ - в среднем потери на одну фрод операцию
- 100$ - затраты бизнеса на арбитраж фрод операции

Итого, при количество транзакций Т = 10 тыс в сутки

- фрод - 50 операций, потери 50*530 = 2650

Возьмем модель с качеством предсказаний фрода ~80% (целевое)

- **Ценность для клиентов** - 2650 * 0.8 = **2120 $** в сутки
- **Ценность для банка** - 50*0.8*100= **4000** $ в сутки

ROI

- **ROI =** ((4000*30*12 - 336000)/(336000)) * 100%= **320 % т.е затраты окупятся менее чем за год**

Цели по проекту на второе полугодие

- Усовершенствование ML
- Доработка сервисных решений
- Определение пороговых метрик
- Работа с пропусками данных и признаками с целью улучшения метрик
- Исследовать другие модели, оценить их эффект.
- Обучение нейронки и сравнение с результатами ML
- Расчет и оценка бизнес эффекта от внедрения проекта

Итого (саммари)

- Выбрали Dataset, провели EDA
- Реализовали Модель
- Спроектировали API, реализовали Front для работы с моделью
- Развернули решение
- Посчитали бизнес ценность, заземлили ее на метрики модели
- Наметили планы на следующие этапы