



# **BIV Hack** **Challenge**

ekanam

2 0 2 4



# Проблематика

Классификация назначений платежей является рутинной задачей которую можно автоматизировать с помощью нейронных сетей

# Решение кейса

Мы решили сравнить 4 модели классификации:

## Решение без нейронных сетей:

1. Наивный Байесовский классификатор + TF-IDF векторизацию

## Нейросетевое решение: (Векторизация с RuBERT)

2. biGRU Classifier
3. RuBERT Classifier
4. FastText

# Стек

- Frontend: React + TypeScript
- Backend: Docker, compose + FastAPI + Dishka
- ML: Transformers, sklearn, pytorch, tensorflow, keras, Natasha, pandas

# Качество моделей

	F1-Score	Accuracy
<b>RuBERT</b>	<b>0.98</b>	<b>0.98</b>
<b>biGRU</b>	0.975	0.96
<b>FastText</b>	0.97	0.97
<b>NBC + TF-IDF</b>	0.96	0.98

# Валидация моделей

	Индекс Джини	p-value критерия Колмогорова- Смирнова	Статистика критерия Колмогорова- Смирнова
<b>NBC + TF-IDF</b>	-	-	-
<b>FastText</b>	0.9987	2e-8	0.9932
<b>biGRU</b>	0.9983	5e-11	0.9948
<b>RuBERT</b>	0.91	1e-13	0.9917

# Плюсы и минусы подходов

## Решение без нейронных сетей:

- Хуже качество
- + Дешевле
- + Быстрее
- + Не требует GPU
- + Легче найти специалиста

## Нейросетевое решение:

- Нужно GPU
- Дольше работает
- Дороже специалисты
- + Лучшее качество
- + Много исследований в области, больше потенциал роста

# Интерфейс

**BIV**

Классификация назначения платежа

Команда: Ekanam

Текст:

Оплата за кофе в зернах по счету No 123 от 15.09.2

Рассчитать

Результат:

SERVICE

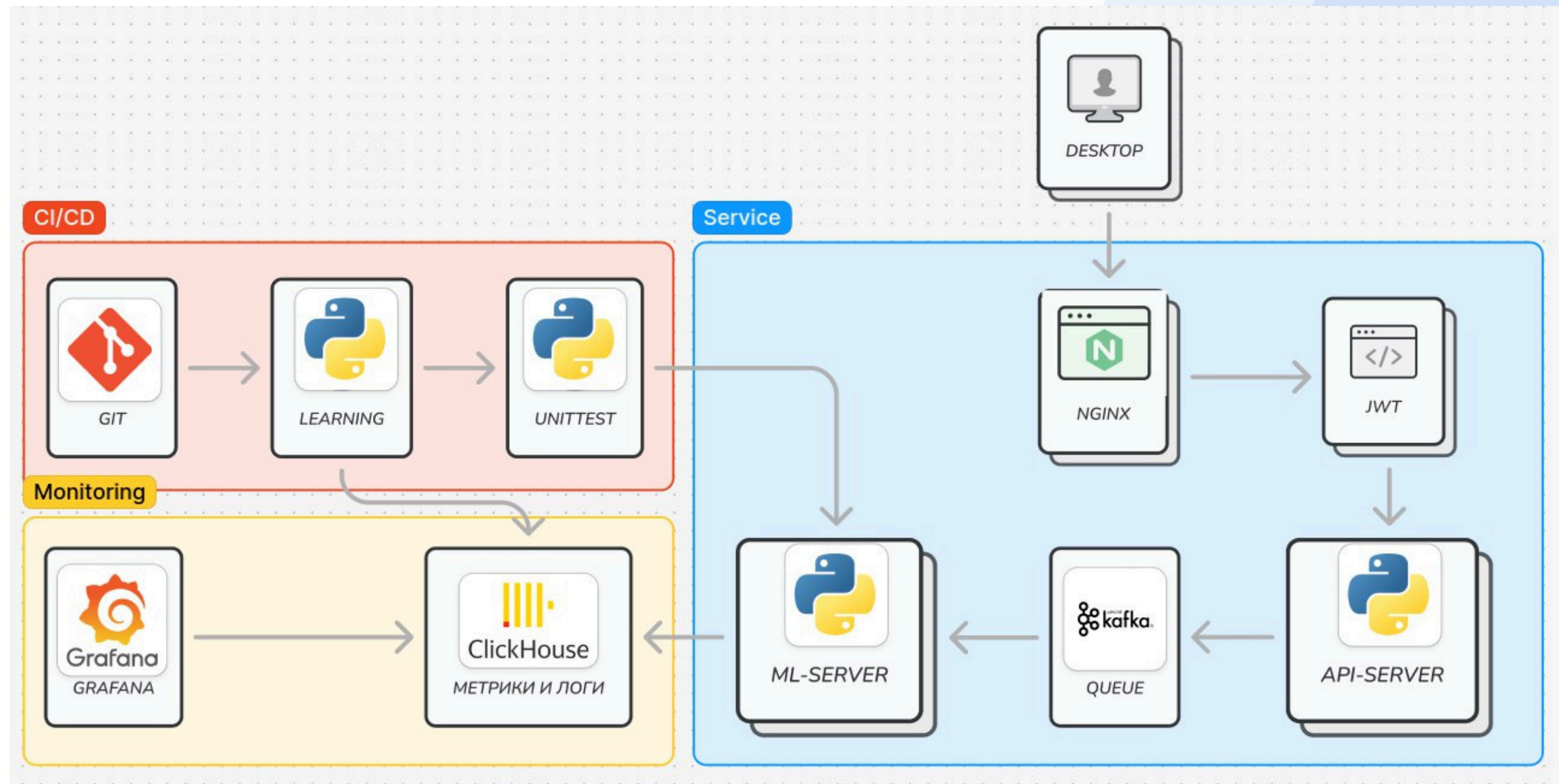


# Пайплайн



# Дальнейшее развитие

- Дообучение RuBERT на большем корпусе данных
- Распределенное решение, CPU-Bound архитектура
- Сбор метрик качества моделей и визуализация в Grafana





# Команда еканат



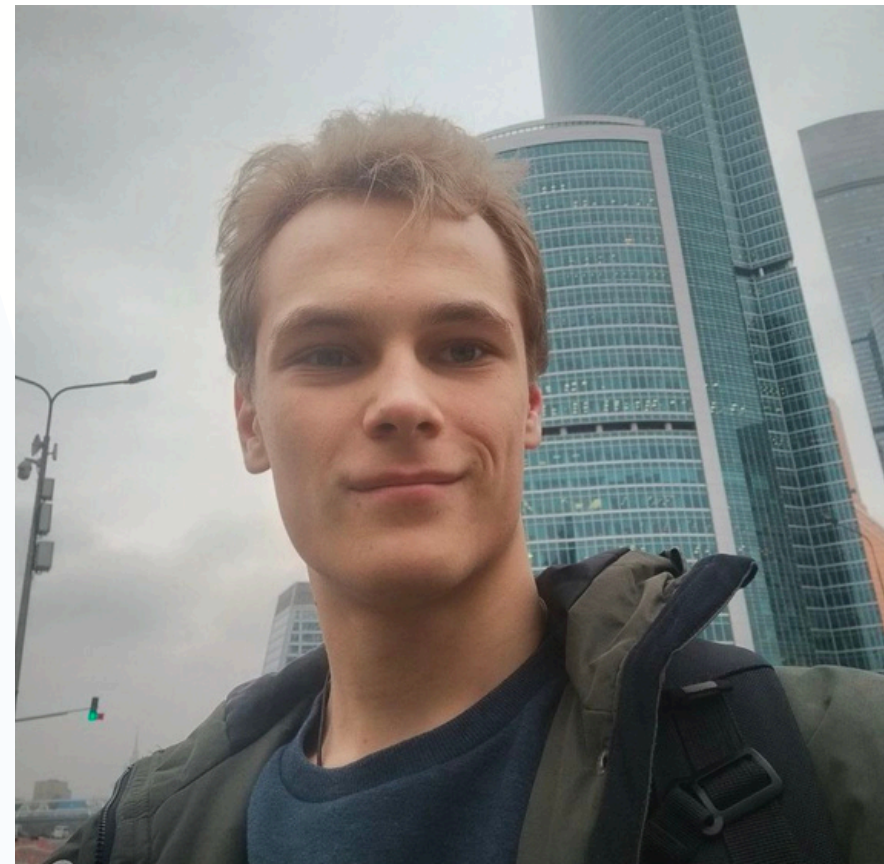
Научный сотрудник, Lead



ML-engineer



Backend разработчик



ML-engineer



Дизайнер



# Ссылка на гитхаб



ekanam

2 0 2 4

