



Решение для  
автоматизации  
"рутинных" задач  
сотрудников  
страховой компании  
с помощью AI»

Еремеев Дмитрий  
Шихалиев Фрэнк  
(tg: frankshikhaliev)



# Концепции, которых придерживаемся при развитии систем

1

Концепция фиче вектора-эмбединга  
вокруг сущности (Клиент, Адрес, Объект,  
История страхования и т.д.)

2

Демократизация за счет open source и  
transfer learning, fine-tuning

3

Supervised  
и unsupervised learning

4

Экономика и клиенты  
в первую очередь при определении  
приоритетов (impact / effort)

5

Венчурная модель  
в экспериментах  
по машинному обучению

6

Context engineering

# Команда RGS LAB и подходы к управлению

- OKR
- Agile / SAFe и команда
- Scrum и Канбан
- Подрядчики часть команды
- Подкоманды
  - Разметка данных
  - Управление данными
  - Тарификация и анализ
  - CV and NLP приложения
  - Внедрение сервисов и пайплайнов
- Подкоманды собраны из разных исторических департаментов (ОПЕРУ, ИТ, Аналитика, тарификация и пр)



# Задачи RGS Lab

Моделирование риска

Моделирование спроса  
Динамическое ценообразование

Anti-fraud

- Анализ графов, модели на графах, графовые БД
- Векторные БД, алгоритмы схожести текста

Распознавание документов

- Полисы
- Документы

Автоматизация колл-центра

- Чат-бот
- Анализ отзывов (текст, аудио)

Анализ изображений

- Фото с предосмотром авто
- Детекция повреждений

Гео-аналитика

Эмбеддинги локаций

Поиск по внутренним базам для ускорения  
процессов и Retrieval-Augmented Generation

Автоматизация андеррайтинга

- Автоматизация расчетов автопарков
- Автоматизация нестандартных котировок андеррайтеров

Анализ медицинских программ

- Вхождение услуги в программу
- Поддержка оператора МедКЦ

# Моделирование риска (авто)

## Каско

- Виновник
- Потерпевший
- Стекло
- ПДТЛ
- Тоталь
- Угон

## ОСАГО

Типы урегулирования

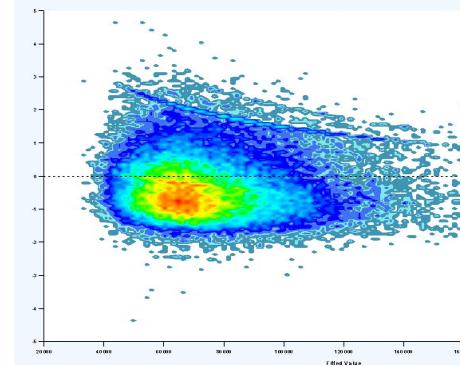
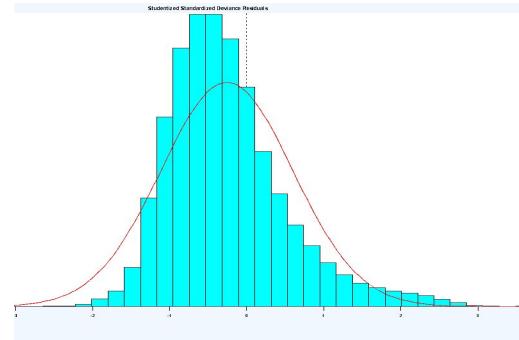
## Риск

Частота

Тяжесть

(Poisson regression)

(Gamma regression)



# Моделирование спроса

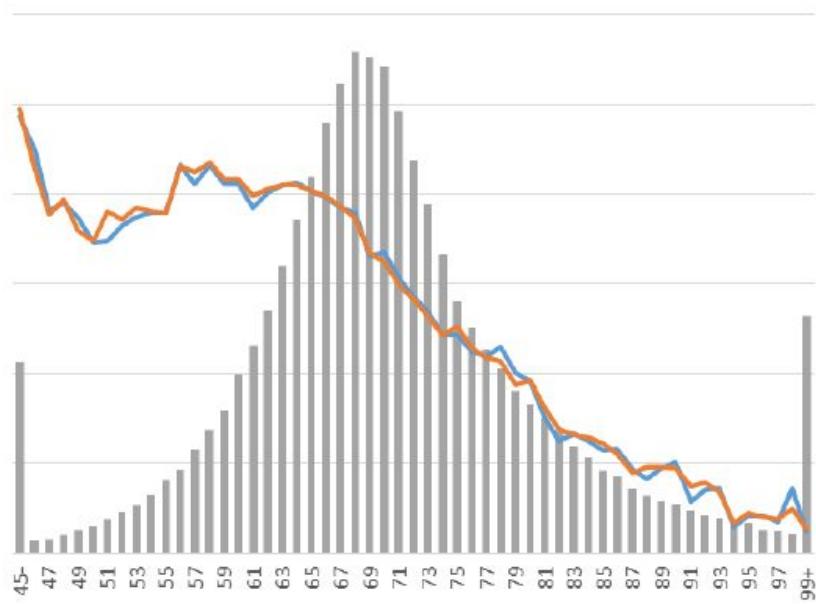
задача классификации в рамках одного события

## Модели спроса

- Пролонгация (расторжение)
- Новый бизнес (конверсия)
- Досрочное расторжение
- Смена покрытия / продукта
- Прогноз вероятности  
(logistic regression)

Зависимость конверсии от ценовой привлекательности компании

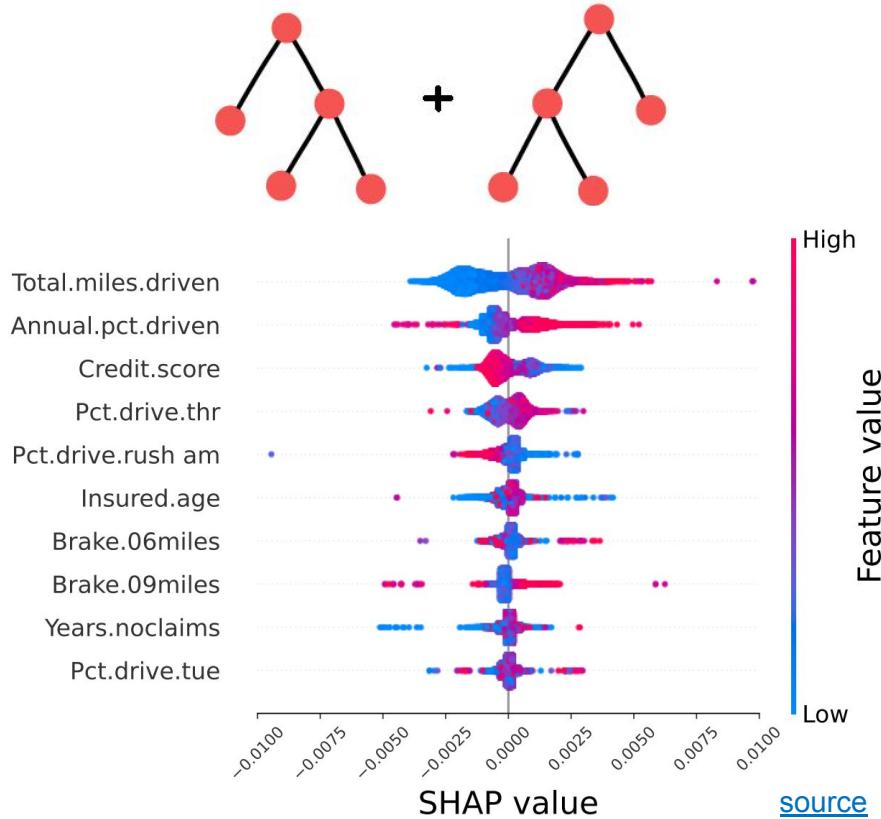
■ Exposure, %   ■ Observed Average   ■ Fitted Average



# GLM -> GBM

## Градиентный Бустинг на деревьях решений

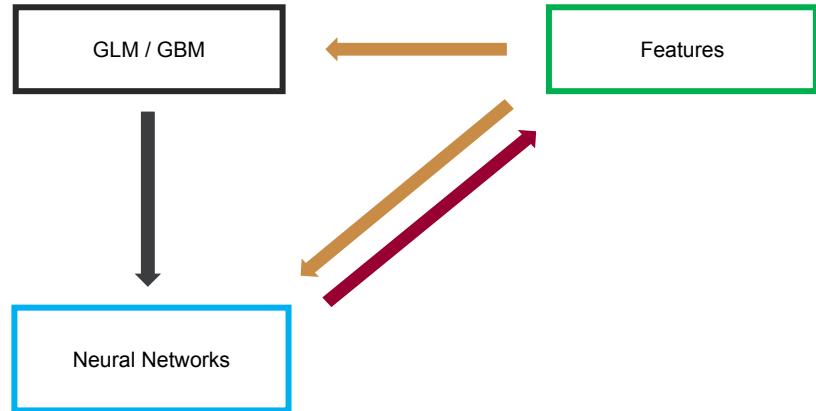
- Позволяет лучше учесть нелинейности особенно для признаков разделяемых деревьями, что влияет на точность предсказаний.
- Облегчает работу с большим количеством признаков.
- Даёт выигрыш в скорости построения моделей.
- Хуже интерпретируемость SHAP, Partial Dependence



[source](#)

# GBM -> Нейронные сети

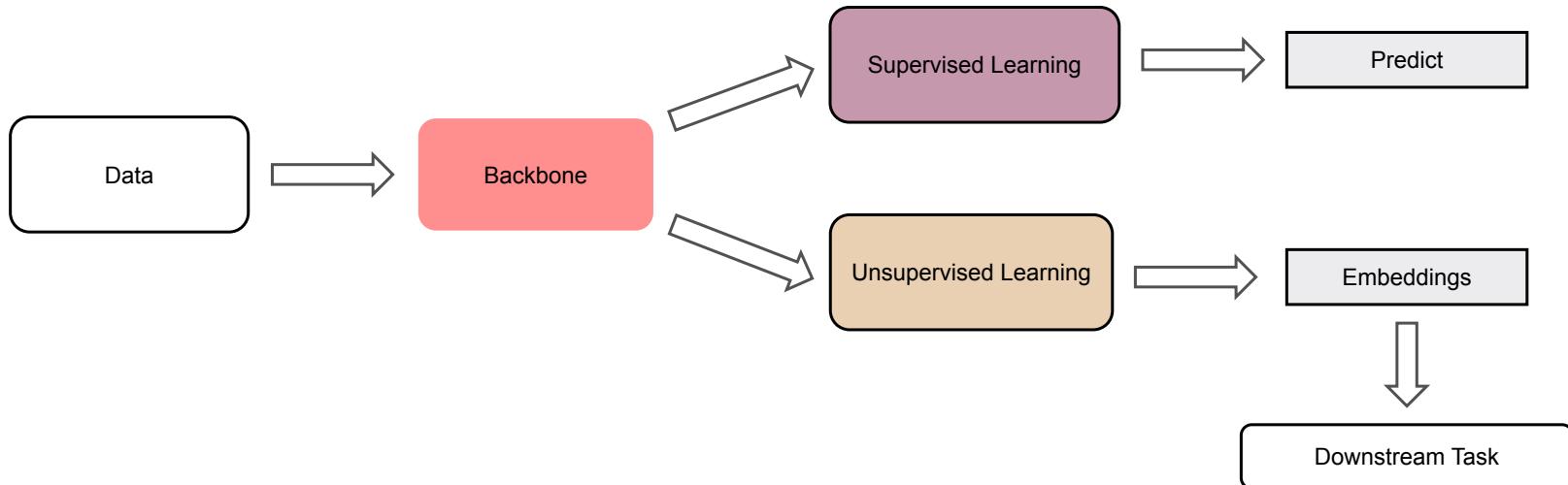
- Нейронные сети могут как заменять GLM / GBM, так и служить для усиления этих моделей.
- Замена рисковых моделей возможна с помощью «табличных» нейросетей.
- Усиление GLM / GBM возможно с помощью генерации признаков нейросетями.



Чем движимо - Цыбенко 89 и Хорнек 91 - теорема об универсальной аппроксимации нейронными сетями непрерывных и в некоторых случаях разрывных функций

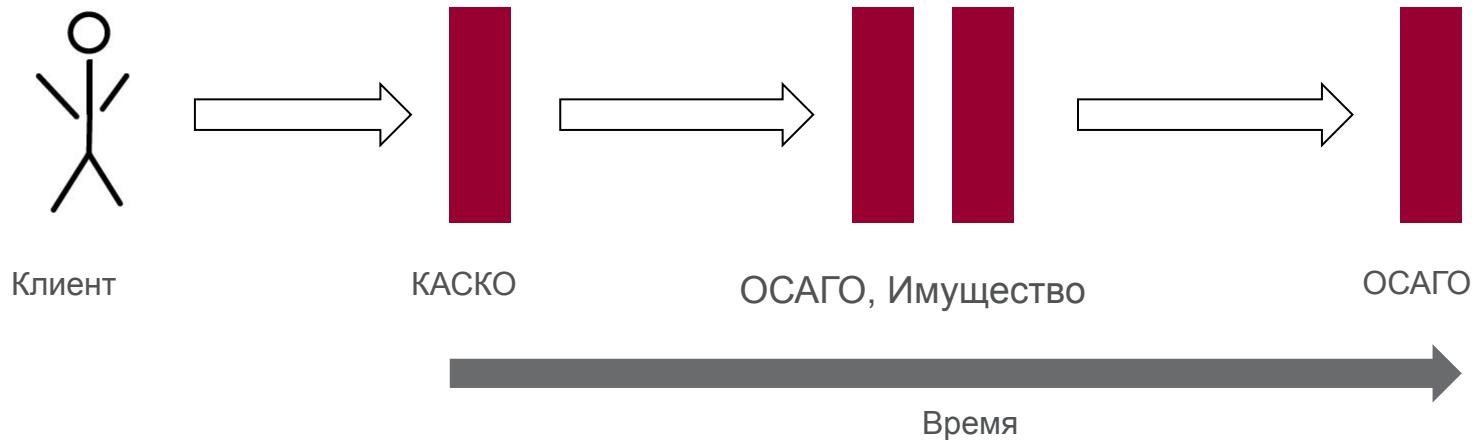
# Векторные представления

- Нейросети предоставляют удобные средства работы со структуризованными данными.
- Тексты, картинки, аудио, последовательности, графы представляются в виде обучаемых векторных представлений (эмбеддингов).



# История клиента

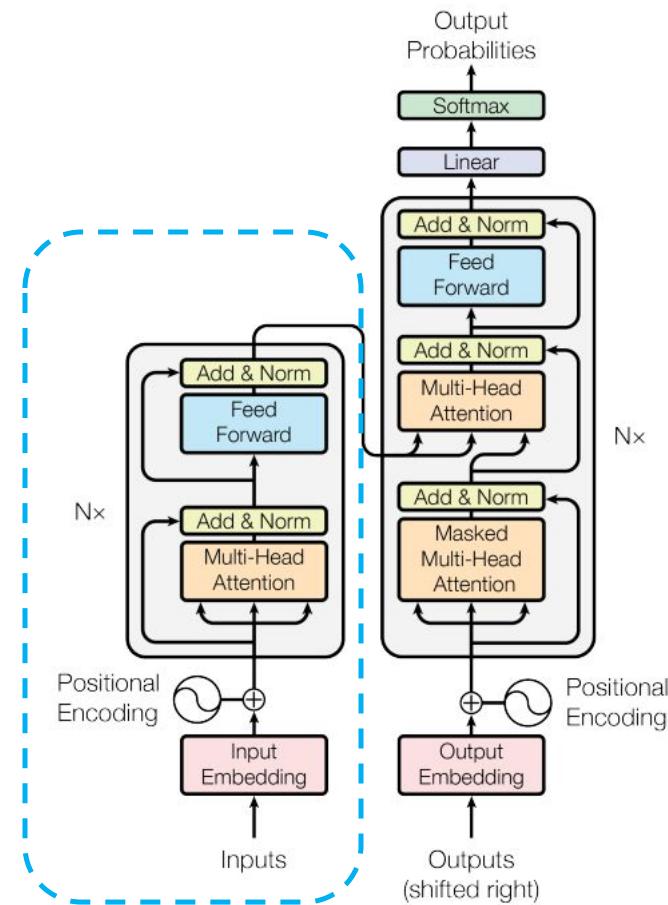
- Учесть влияние страховой истории клиента
- Event-sequence задача
- Каждый полис имеет свои характеристики и историю убытков



# История клиента

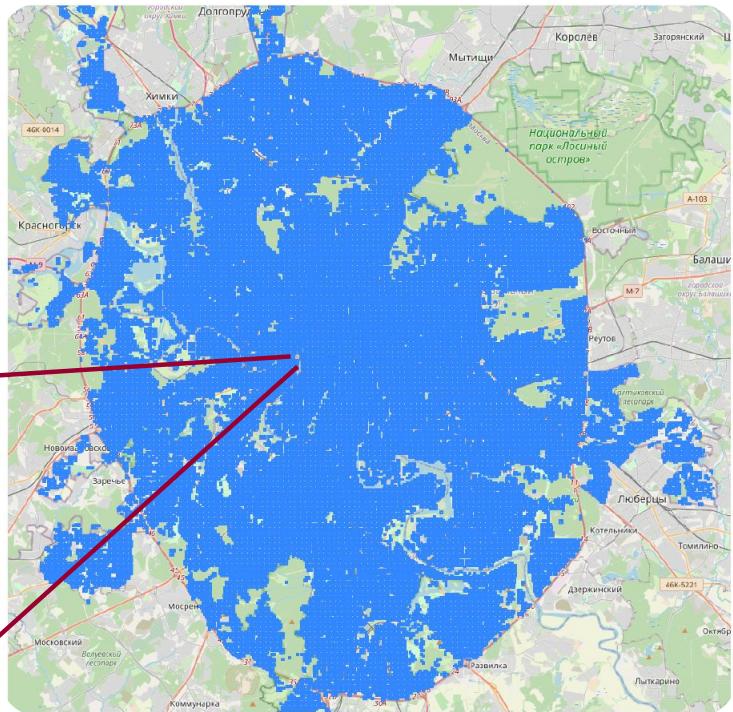
- Для получения векторного представления последовательности событий используем Transformer Encoder.
- Supervised learning:  
[CLS] токен + log-likelihood
- Self-supervised learning:
  - CoLES (Contrastive Learning for Event Sequences with Self-Supervision)
  - Основанные на подходах из LLM.

Transformer



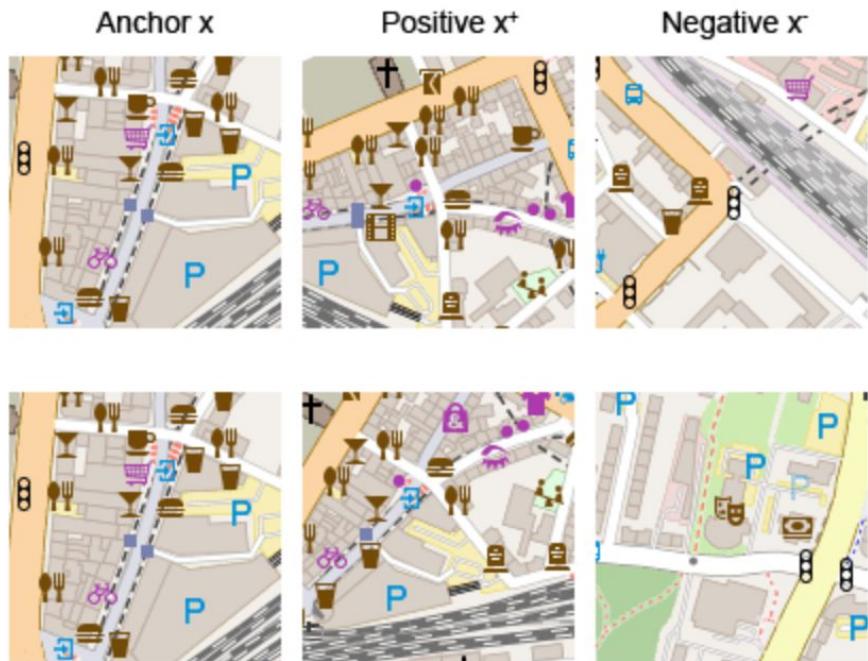
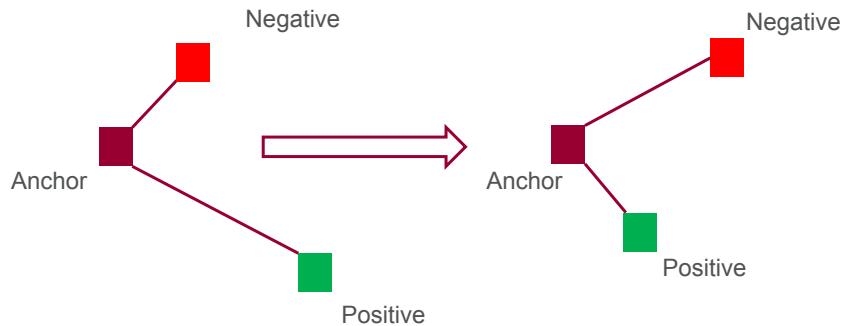
Geo

- Геосетка: полигоны 100 x 100 метров
  - Признаки: основаны на POI, местности в окрестности, свойствах городов
  - Слои в карты в стандартном стиле
  - Эмбеддинги полигонов
  - Геокодирование адресов



# Векторные представления полигонов

- Энкодер: сверточная нейросеть (CNN)
- Каналы изображения: слои карты
- Contrastive learning



[source](#)

# Автоматическая обработка запросов к андеррайтерам

- Классификация / типизация обращений.
- Формирование текста ответа на запрос
- Прогнозирование ответа на базе исторических согласований
- Автозаполнение форм на базе RPA

BERT

# Немного про будущее тарификации и в целом страховых продуктов

AI Тренд		Развитие систем страхования
1	Мультимодальные и большие языковые модели способные получить json а в перспективе и сразу расчет из изображений паспорта, прав, стс и птс	Новый вид интерфейсов (чаты, голосовые интерфейсы), постепенная замена привычных форм на чаты и естественное общение
2	Улучшение метрик моделей создания кода Qwen Coder и аналогов, создание Spec Mode reasoning - создание кода, формул, а значит и тарифов, масок полисов, также правил и условий страхований значительно упрощаются	Значительное ускорение деплоя новых продуктов и формул расчета, а также страховой документации, создание бизнес требований и архитектуры приложений
3	Развитие Agent to agent и model context protocols, интеграция api спецификаций в работу агентов на базе LLM	Языковые модели с большим контекстом смогут гибко вызывать наборы сервисов по 10-20 штук и по 20-30 признаков в каждом, что потенциально покроет систему тарификации страхования целиком

# Немного про будущее тарификации и в целом страховых продуктов

AI Тренд		Развитие систем страхования
4	Retrieval Augmented Generation по клиентской и внутренней информации, развитие модулей памяти агентов за счет большого контекста моделей	Кратное ускорение работы андеррайтинга, возникновение автоматизированного AI агента андеррайтера без RPA
5	Reasoning + rag, создание пайплайнов для работы с неполными данными	Возможность формирования моделей оценки риска не неполных параметрах с заложенными общими знаниями о мире и в перспективе в домене страхования с подключением знаний из баз
6	Мультиагентные системы	Единый полис клиента
7	Развитие систем создания кода	Упрощение создания страховых страховых продуктов за счет генеративного ИИ  Шаг к настоящим рекомендательным системам и полноценной персонализации

# Внутренняя платформа работы с ИИ ai.rgs.ru

- Развёрнута система доступа к LLM моделям для сотрудников компании
- Еженедельно 20 пользователей подключаются к платформе и начинают использовать LLM модели
- Работа идет на картах nvidia h100
- Подключены RAG системы, агенты по процессам и документам компании
- Самые популярные модели
  - qwen-coder 2.5 14b
  - gpt-oss 20b
  - phi4 mini

## РОСГОССТРАХ

Q Поиск в чатах

+ Новый чат

### ИСТОРИЯ

Напиши вычисление  $n!$  на python

Как работает гпт осс?

Текст 55 слов

Напиши код на python

по вычислению  $n$

Напиши код на python

вычисления  $n$

Напиши код на python

вычисления  $n$

Вакансия бизнес

аналитика

Вакансия бизнес

аналитика

Вакансия бизнес

аналитика

### НАСТРОЙКИ

👤 Администрирование

🗑 Очистить историю

✉️ Сообщить о проблеме

✖ Выйти из аккаунта

qwen2.5-coder:14b

читаемого вывода результата.

Я найдет ошибки в коде

```
def factorial(n): result = 1 for i in range(1, n + 1): result = i
```

```
# Форматирование с математической нотацией
print(f"\n! = {result};_") # Разделитель групп разрядов
print(f"\n! = {result};_") # Подчеркивание групп разрядов
```

```
return result
```

```
factorial(10)
```

+ Ваш код вычисляет факториал, но есть небольшая ошибка в цикле. Вместо умножения result на  $i+1$ , вам должно умножать его на просто  $i$ . Также ваша функция должна возвращать только результат вычисления факториала, а форматирование выводить внутри нее может быть излишним. Рассмотрите следующий исправленный вариант: `pythondoc factorial(n): result = 1 for i in range(2, n + 1): # Исправлено: начинаяем с 2, так как 1 умножать не нужно result *= i return result` Вывод форматированного результата вне функции `=10fact_result = factorial(10)print(f"\n! = {fact_result};_")` # Разделитель групп разрядов`print(f"\n! = {fact_result};_")` # Подчеркивание групп разрядов`factorial(10) # Вызов функции без вывода, если не нужно возвращать значение. Теперь ваш код корректно вычисляет факториал и форматирует его вывод вне функции.`

Спросите что-нибудь



Видео: <https://disk.yandex.ru/d/WrgcG0nOe6uSZQ>

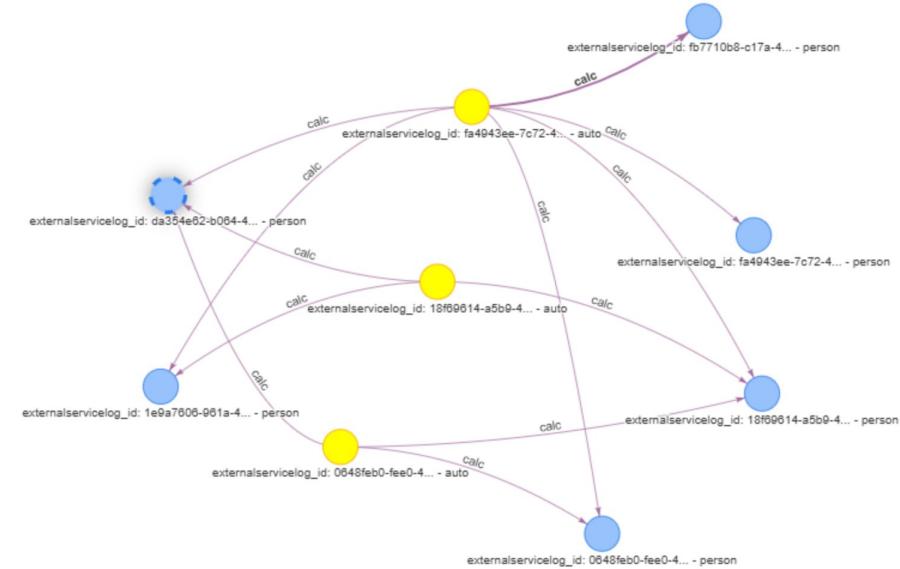
# Инсайты после анализа работы пользователей с ai.rgs.ru

- Разработчики основные пользователи системы
- Возможность быть удобным собеседником для проговаривания бизнес задачи (наведение на мысли)
- Помочь переписать и упростить сообщение
- Наиболее полезная модель по соотношению цена-качество из локальных пока gpt-oss 20b
- Не всегда привычный пользовательский путь для работы с моделями может вызвать вопросы
- Модели менее 14 млрд параметров практически бесполезны для общих задач и требует таргетированной настройки для применения



# Anti-Fraud

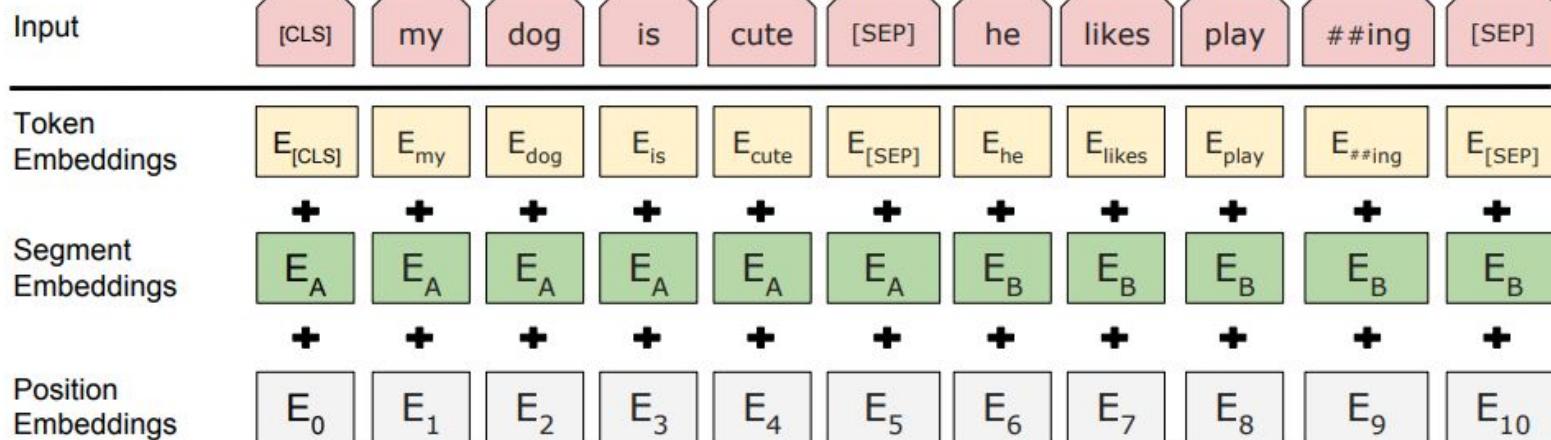
- Графы: убытки + котировки
- Вершины: персоны, авто + признаки
- Ребра: связи соответствующего типа
- Semi-Supervised / Unsupervised модели
- Графовые и векторные БД



# Обработка отзывов клиентов

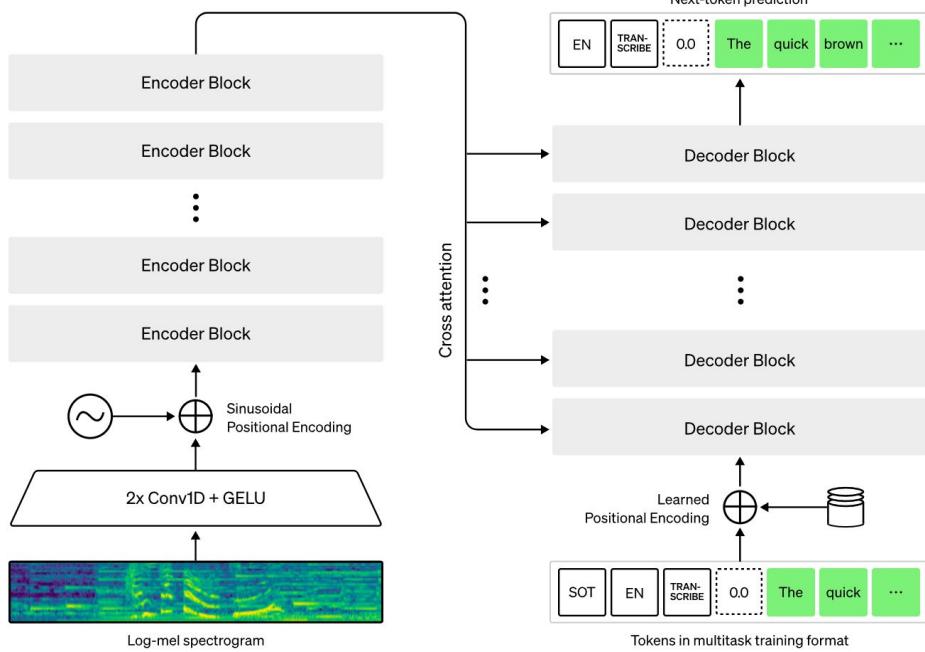
- Классификация / типизация / суммаризация отзывов, обращений.
- Удовлетворенность клиента, готовность рекомендовать компанию, удобство клиента.
- Текстовые отзывы □ Transformer-based модели

## BERT



# Обработка звука

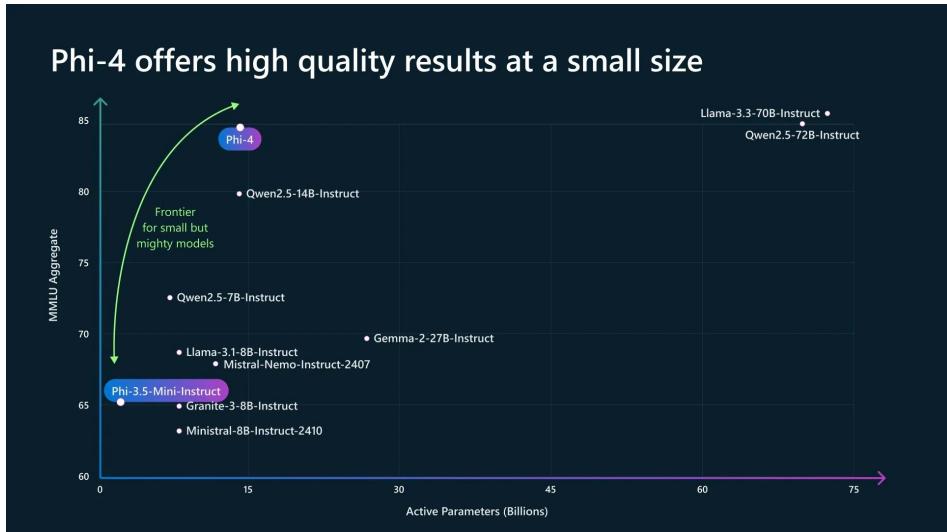
- Модели распознавания звука (ASR) позволяют перевести речь в текст
  - 15 тыс звонков в день
- Видные представители для домена Колл-Центра:
  - Whisper Turbo (WER - 0.44)
  - T-One (WER - 0.24)



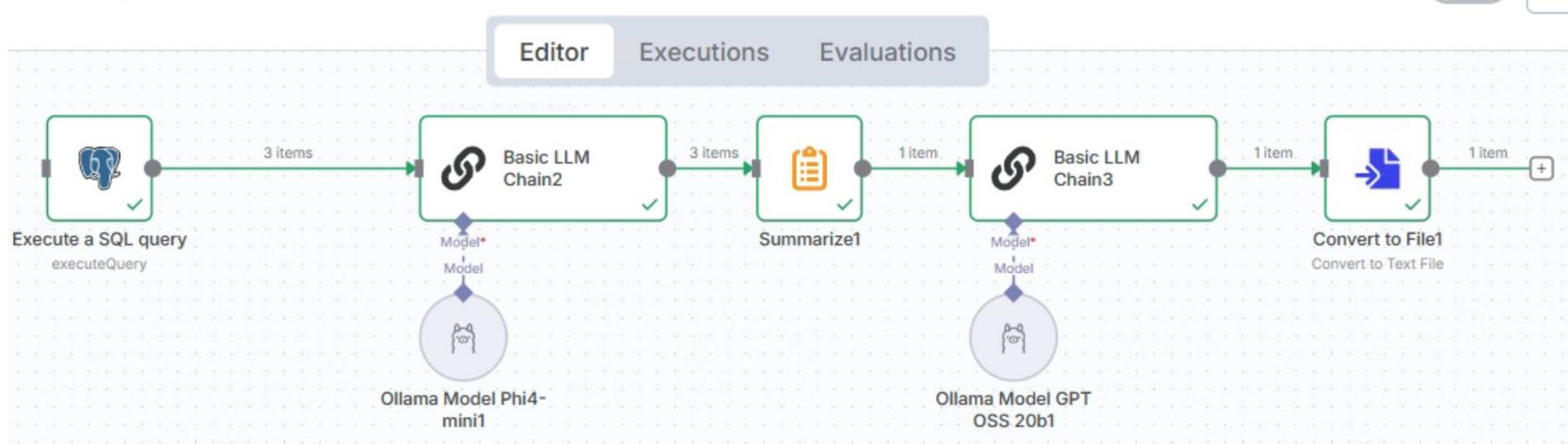
[Whisper](#)

# Речевая аналитика

- Модели анализа текста
  - phi4-mini:3.8b
- Приветствие, окончание, жалобы, типы звонков, вежливость, окончание, работа с возражениями, быстрое завершение, лингвистические особенности, негативные фразы, слэнг, разговорная речь, нецензурная лексика и споры, уменьшительно-ласкательные формы, замечания от клиента и тд (более 100 признаков в звонке)
- Суммаризация
- Инсайты отдавные на откуп модели



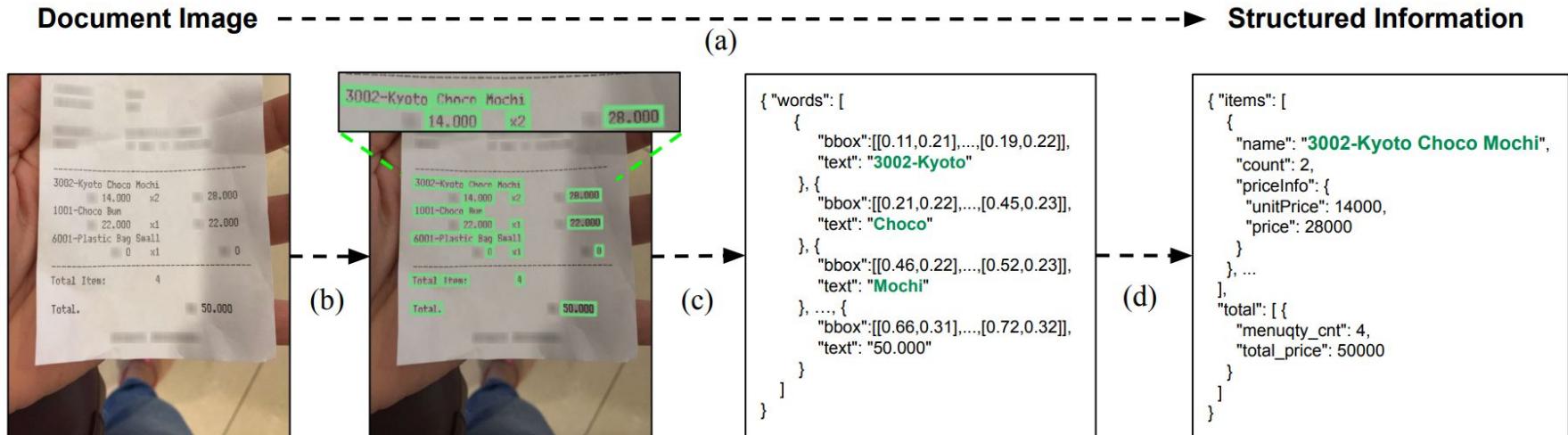
# n8n - low code решение



- Используем для анализа цепочки промптов
- Построение многоуровневых цепочек анализа группы звонков, особенно полезных для моделей на 4 млрд параметров
- Простые интеграции с существующими моделями для проработки пользовательских сценариев
- Упрощение доступа продвинутых пользователей к ИИ и автоматизации

# OCR

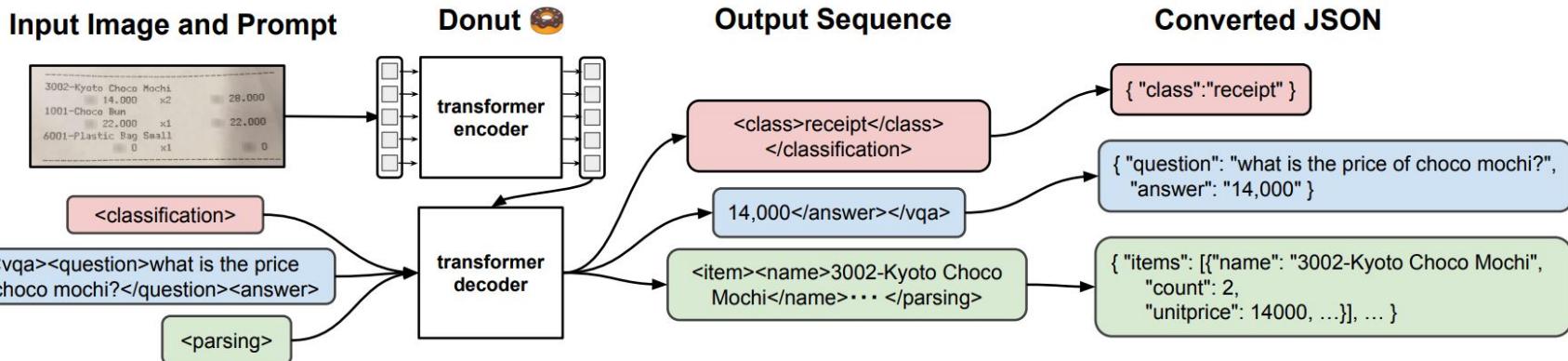
- Распознавание документов
- Паспорта, полисы, ...



# OCR

- Donut
- Phi-3-Vision

## Donut



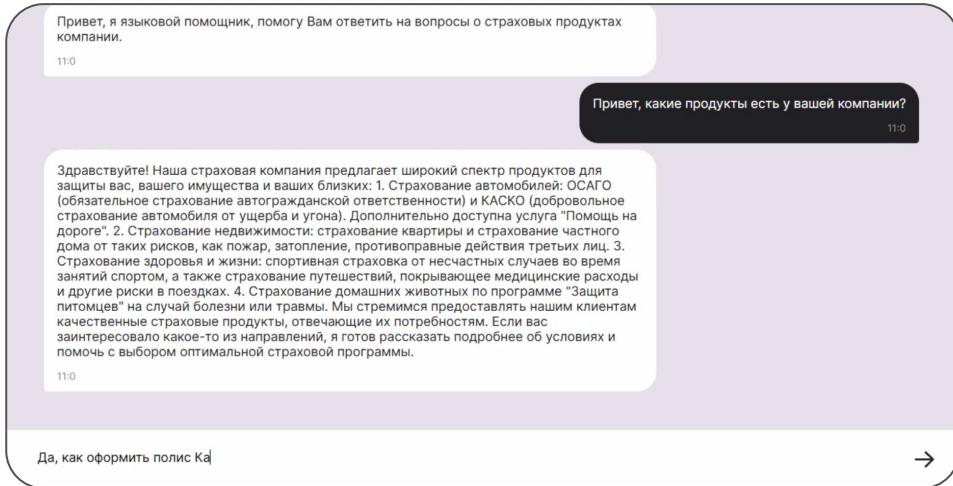
# Распознавание повреждений

- Детекция повреждений по фото.
- Можно использовать как непосредственно детекцию / оценку повреждений, так и оценивать влияние на риск при принятии на страхование.



# ИИ Агент помощник страхового агента

- Подсказки по продуктам компании на базе страховой документации
- Ссылки на продукты
- В будущем - проактивное предложения для клиента на основе чата с клиентом в том числе индивидуальных условий и тарифов



- e5 tokenizer
- gemma 4b, 12b
- GPT OSS 20b
- typesense vector database

# ИИ как новый интерфейс вместо структурированного API

- Традиционные интеграции через API требуют точно прописанных запросов и форматов.
- С использованием ИИ появляется возможность заменять жёсткие API на гибкий слой на базе LLM.
- Система получает запрос на естественном языке:
  - «рассчитай тариф для клиента X с такими-то условиями»
- LLM “понимает” его и обращается к нужным базам данных и сервисам.
- LLM выступает как универсальный интерфейс, способный по описанию задачи вызвать различные инструменты (по стандарту Model context protocol) и собрать ответ.

# Запросы и будущее

MCP сервисы для страховых компаний

Opensource проекты которые хотели бы провести пилоты с корпорациями

Система исходящего персонального взаимодействия на базе истории с применением LLM для снижения риска и роста продаж

Новые подходы борьбы с мошенничеством на базе графового и мультимодального ИИ

Корпоративная система АБ тестирования

Supervised fine tuning модели на страховой домен доступные по MCP, API, onprem

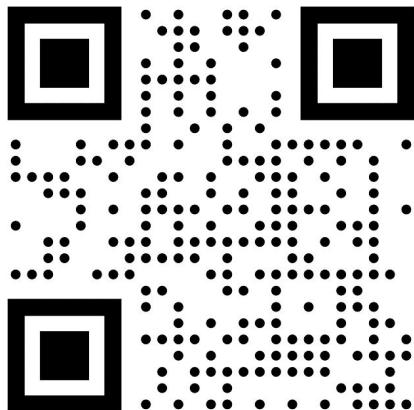
Системы контроля и тестирований gen AI пайплайном с моделями и серверами разных видов

Доказавшие эффективность цепочки агентов для решения бизнес задач

# Для партнеров

Предложения по ИИ  
от партнеров собираем  
централизованно:

[ai@rgs.ru](mailto:ai@rgs.ru)



- **Аутстаф** - достижения в страховании и ИИ, запрос на резюме свободных сотрудников для привлечения с опытом ИИ в страховании
- **Аутсорс** – примеры проектов в страховании в области ИИ, чем подрядчик гордится
- **Интеграторы и стартапы** - продукты по ИИ с подтвержденным эффектом и успешным кейсом внедрения, подходящим под страхование
- **Резюме сотрудников** - опыт работы с ИИ в страховании

# Приложение

# MCP example

The screenshot shows the PyCharm IDE interface with the following details:

- Project Structure:** The left sidebar shows a project named "TEMPLATE-MCP" with several files and folders:
  - lama\_index
  - servers
  - calculate\_policy\_price\_range
  - server.py
  - .python-version
  - cl.py
  - ollama\_client.py
  - pyproject.toml
  - README.md
  - server\_weather.py
  - test\_client\_sse.py
  - test\_client\_stdio.py
  - test.py
  - uv.lock
  - gitignore
  - README.md
- Code Editor:** The main window displays the content of `server.py`.

```
def calculate_policy_price_range():
    """
    Расчитывает диапазон цен страхового полиса.

    :param driver_experience: Стаж водителя (в годах)
    :param car_power: Мощность автомобиля (в лошадиных силах)
    :param region: Регион (наименование или код региона)
    :return: Диапазон цен полиса (min_price, max_price)
    """

    # Базовый тариф (БТ) – допустим, 5000–7000
    base_tariff_min = 5000
```
- Terminal:** Below the editor, a terminal window shows the command and its output:

```
� main: 0.11 0.0 5.5
lama_index > servers > calculate_policy_price_range > server.py > calculate_policy_price_range
9 def calculate_policy_price_range()
14
15
16
17
18
19
20
21
22
23
24

d:\d1\>/projects/template-mcp/Lama_Index/client uv run client_run.py --config configs/example.yaml
Context7 Documentation MCP Server running on stdio
Enter your message: Calculate the insurance policy for a driver from Moscow with 10 years of experience and a 180 horsepower car.
User: Calculate the insurance policy for a driver from Moscow with 10 years of experience and a 180 horsepower car.
Tool input: "calculate_policy_price_range" {'car_power': 180, 'driver_experience': 10, 'region': 'Moscow'}
Agent: Based on the information provided, the insurance policy for a driver from Moscow with 10 years of experience and an 180 horsepower car would cost between 2880.0 to 61740.0 rubles.
Enter your message: 
```
- Bottom Status Bar:** Shows the current file as `main.py`, the author as IgnatovD (18 hours ago), the line number as Ln 21, Col 23, and the file type as Python.

- У generic моделей доменных знаний маловато
  - Общие знания, представления и зависимости языковые модели вполне способны закладывать и это уже полезно
  - Первые успешные кейсы ожидаем в доменной адаптации модели и SFT (supervised fine tuning)
  - Api с тарифами пока никуда не уйдут, но взаимодействие может меняться особенно для простых продуктов с малым числом параметров

# Метрики

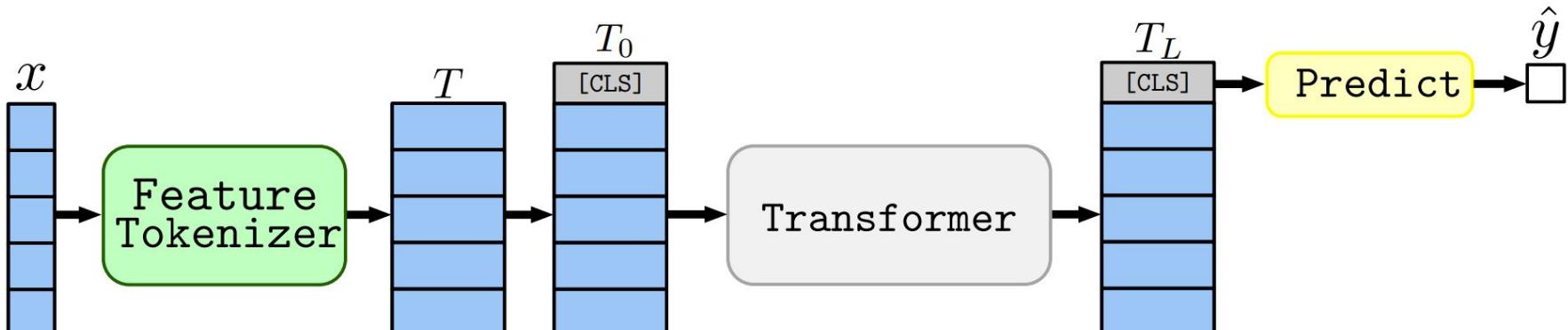
300



млн  
прибыли в год

# Табличные данные

- Модификации MLP
- Табличные трансформеры: TabTransformer, FT-Transformer, ...



[FT-Transformer](#)

# Мера персонализации тарифов - метрики машинного обучения

	Частота убытков (Пуассоновская регрессия)	Тяжесть убытка (Гамма регрессия)
Основная для замера качества персонализации	Poisson Deviance	Gamma Deviance
Для ранжирования в том числе при регрессиях	Модифицированные Gini / Lift / AUC	
Вспомогательные	MSE, MAE, MAPE, Pseudo R2 и пр	
Продуктовые и клиентские бизнес метрики	WP, LR, Conversion, Renewal, CF, AvClaim...	

# Разберемся с тематикой в рамках выступления

1

ИИ - многообразие методов обучения машин, все что входит в machine learning, large language models, multimodal models

2

Большие данные - непосредственно данные, во всем многообразии (изображения, звук, текст, таблицы, сущности и связи между ними и различные представления)

3

Персонализация - наиболее точная оценка риска с учетом многообразия представленных данных

4

Персонализация сервиса - сокращение времени на обслуживание по различным бизнес процессам

5

Тарификация - определение частоты и тяжести убытков и “ себестоимости” страхования

6

Страхование - все что связано с неопределенностью

# GLM

## Generalized Linear Model

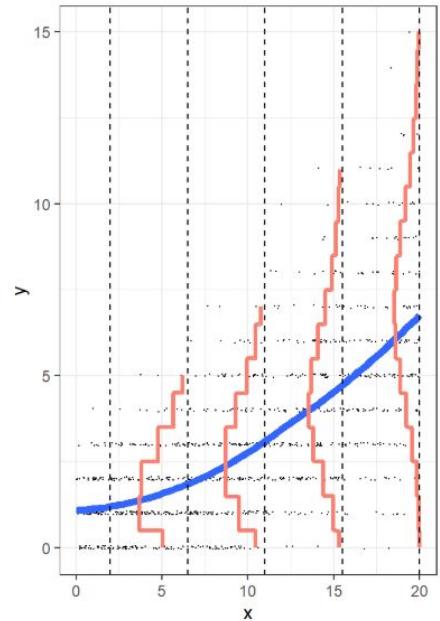
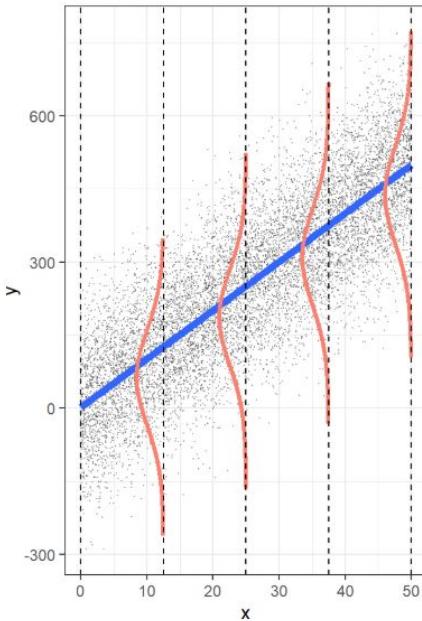
Наблюдения  $Y_i$  из экспоненциального распределения:

$$Y_i \sim p(y_i | \theta_i, \varphi) = c(y_i, \varphi) \exp\left(\frac{y_i \theta_i - a(\theta_i)}{\varphi}\right).$$

Мат. ожидание  $\mu = \mathbb{E}(Y_i)$ :

$$\mu = g^{-1}(\mathbf{w}_1 X_1 + \mathbf{w}_2 X_2 + \dots + \mathbf{w}_n X_n),$$

где  $g$ - функция связи.

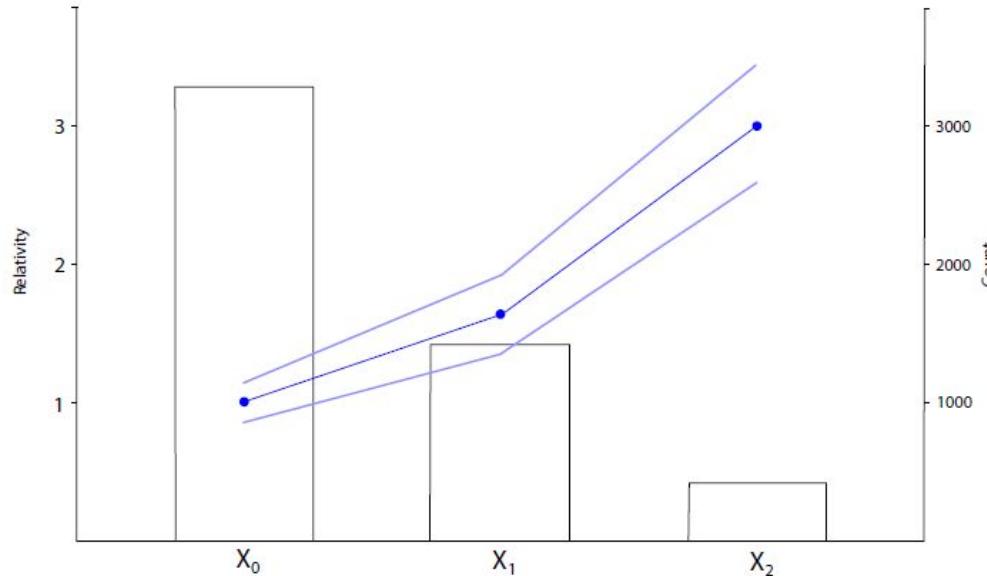


# Likelihood

Параметры  $\mathbf{w} = \{w_1, w_2, \dots, w_n\}$  определяются максимизацией правдоподобия:

$$l(\mathbf{w}) = \log p(\mathcal{D}|w) = \sum \log p(y_i|x_i, \theta_i, \varphi),$$

$$\hat{\mathbf{w}} = \operatorname{argmax} l(\mathbf{w}).$$



# Пример оценки частоты и среднего убытка

- На основе данных по случайной выборке из 5 наблюдений полугодовых экспозиций полисов ОСАГО требуется спрогнозировать:
- Частоту страховых случаев (используя Пуассоновское распределение).
- Средний размер убытка в случае наступления страхового события (с использованием Гамма-распределения).
- Цель — определить ожидаемую стоимость риска для сегмента «Ущерб до 200 тыс. рублей».

## Результаты прогнозов

LightGBM		ChatGPT o3		ChatGPT GPT-4.5		Gemini 2.5 Pro	
Частота	Средний убыток	Частота	Средний убыток	Частота	Средний убыток	Частота	Средний убыток
0,013	45 078	0,0284	43 992	0,042	64 850	0,055	80000
0,024	38 231	0,0203	51 516	0,027	79 620	0,012	70000
0,009	49 769	0,0267	44 064	0,036	59 430	0,065	75000
0,006	39 578	0,0185	47 700	0,019	85 410	0,03	95000
0,024	49 066	0,0328	44 100	0,046	69 780	0,018	72000

# Пример оценки частоты и среднего убытка

## Промпт

- Данна случайная выборка из 5 строк со значениями факторов по половинам полисов, разделенных по полгода по экспозиции.
- Нужно предложить оценки частоты и среднего убытка по ОСАГО для риска "ущерб до 200 тысяч рублей".
- Оцени прогнозные значения для величины частоты и среднего убытка, как если бы ты моделировал частоту с помощью распределения Пуассона и тяжесть убытка с использованием гамма-распределения.
- Дай оценки прогнозных значений для этих 5 наблюдений исходя из общих представлений о мире и факторах. Представь числовые значения прогнозов для частоты и среднего убытка, которые являются непрерывными величинами.

Думал на протяжении 2m 39s >

Ниже – ориентировочные point-прогнозы, если

- частоту  $N$  моделировать распределением Пуассона с параметром  $\lambda$  «убытков ≤ 200 тыс. руб.» на пол-года экспозиции;
- средний убыток  $S$  (при условии, что убыток произошёл) – гамма-распределением с математическим ожиданием  $\mu$ .

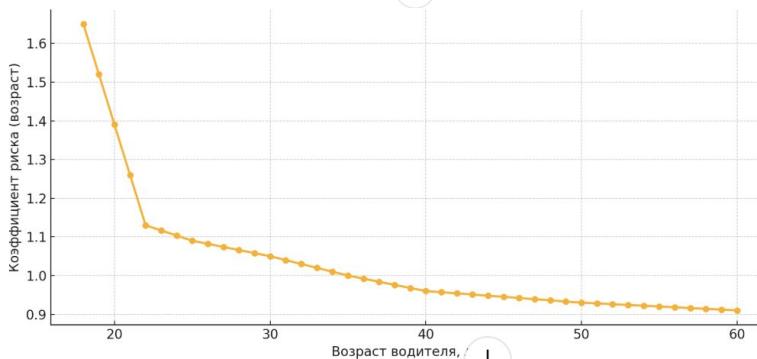
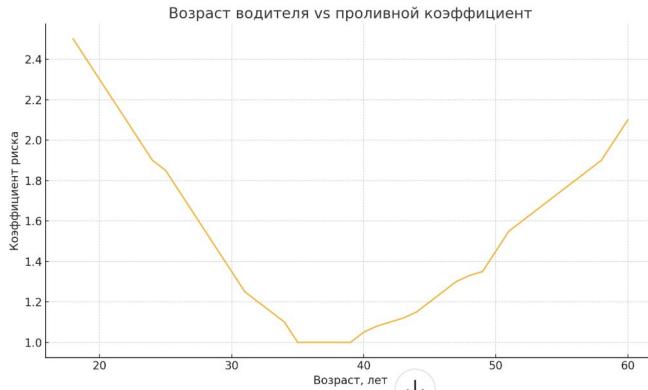
## Базовые гипотезы

Компонент	Принятая база	Как корректировал
Базовая частота	$\lambda_0 = 0.021$ на $\frac{1}{2}$ года ( $\approx 0.042$ в год) – соответствует порядка 7 % всех ОСАГО-случаев в год, из которых ~60 % укладываются в лимит 200 тыс.	Мультиплексивные коэффициенты по мощности автомобиля, возрасту ТС, числу водителей, индикаторам «рисковых» водителей и (если бы был) статус такси.
Базовая тяжесть	$\mu_0 = 45000$ руб. – медианный ремонт при мелких ДТП без totала	Коэффициенты по мощности (+), возрасту ТС (-: запчасти дешевле), «новизне» (самая молодая машина – дороже ремонта).

## Прогнозы по 5 наблюдениям

Полис-ID	$\hat{\lambda}$ за 6 мес.	$\hat{\mu}$ (руб.)
9173203	0.0284	43 992
17 192 313	0.0203	51 516
17 525 903	0.0267	44 064
3 694 827	0.0185	47 700
13 817 737	0.0328	44 100

# Задача проверки понимания базовых зависимостей (ChatGPT 4 и o3 + Deep Research)



- Базовый набор рассуждений без корректировок
- Модель при рассуждениях ухватила идею о росте тарифа к 60 годам написанную в тексте и заложила данную неверную концепцию
- После 3-5 наводящих вопросов все же удалось получить более «верную» зависимость

# Нейросети в тарификации на табличных данных

- Гибридные модели CANN (combined actuarial neural network)
  - Корректировка базового GLM/GBM с помощью нейронных сетей с прямой связью (Feed-forward NN)
  - Модель сохраняет интерпретируемость за счет skip-connection, при этом повышается точность
- Автоэнкодеры при работе с категориальными данными
  - Автоэнкодер обучается на one-hot-представлении категориальных данных, сокращая их размерность, но сохраняя важные характеристики данных
  - Обученный автоэнкодер может применяться на различных наборах данных, например, обученный на данных по частоте убытков автоэнкодер можно применить на данных по тяжести.
- Интерпретация нейронных сетей с помощью суррогатных моделей GLM
  - Для каждого фактора и каждой пары факторов рассчитываются partial dependence эффекты, на основе которых данные сегментируются на гомогенные группы
  - На сегментированных данных строится GLM модель, что дает понятную тарифную сетку без потери инсайтов исходной модели

# Компьютерное зрение для оценки ущерба авто

- Использование vision-LLM (модель GPT-4 с функцией анализа изображений), обученной на фотографиях поврежденных автомобилей.
- Модель научилась классифицировать типы повреждений и локализовать их на фото, показывая качество на уровне специализированных CV-моделей.
- Модель дает контекст – например, различает мелкую трещину на стекле от полностью разбитого лобового стекла.
- В эксперименте настроенный vision-LLM превзошел обычный CNN и «необученный» GPT-4 в задаче классификации и локализации повреждений автомобиля, что демонстрирует ценность сочетания языка и зрения для страховых задач.

[2]

[https://github.com/IAA-AITF/Actuarial-AI-Case-Studies/tree/main/case-studies/2025/car\\_damage\\_classification\\_and\\_localization](https://github.com/IAA-AITF/Actuarial-AI-Case-Studies/tree/main/case-studies/2025/car_damage_classification_and_localization)

# LLM для оценки убытков по текстам

- Внедрение LLM для анализа текстовых описаний страховых случаев позволяет выделить новые факторные признаки и существенно повысить точность модели прогнозирования выплат.
- Дополнение градиентного бустинга сгенерированными LLM признаками (по описанию травм) снизило RMSE примерно на 18% и почти удвоило  $R^2$  (с 0.27 до 0.51) при – то есть модель с AI значительно точнее предсказывает итоговую стоимость убытка.

[1]

[https://github.com/IAA-AITF/Actuarial-AI-Case-Studies/tree/main/case-studies/2025/claim\\_cost\\_prediction\\_with\\_LLM-extracted\\_features](https://github.com/IAA-AITF/Actuarial-AI-Case-Studies/tree/main/case-studies/2025/claim_cost_prediction_with_LLM-extracted_features)