

# ЛЦТ 2025

Сервис для отслеживания динамики  
клиентских настроений и проблем  
по конкретным банковским продуктам  
в режиме реального времени

MISIS x MISIS

# Проблема



## 01

### Функциональность

Аналитикам важно получать всю необходимую информацию из данных и визуализировать ее. Мы создали сервис, наиболее подходящий под данные задачи

## 02

### Парсинг

Самостоятельный анализ отзывов увеличивает время реагирования на проблему. Наш сервис решает эту проблему через автоматизацию, фильтры и удобный интерфейс

# Решение



## 01

### Скорость

Использование модели по API обеспечивает быструю работу сервиса

## 02

### Удобство

Интуитивно понятный интерфейс и минималистичный дизайн, фильтры по отзывам, инфографика для аналитиков

## 03

### Качество

Использование LLM с 12-ю миллионами параметров для классификации показывает высокие метрики

# Продукт

## Сервис

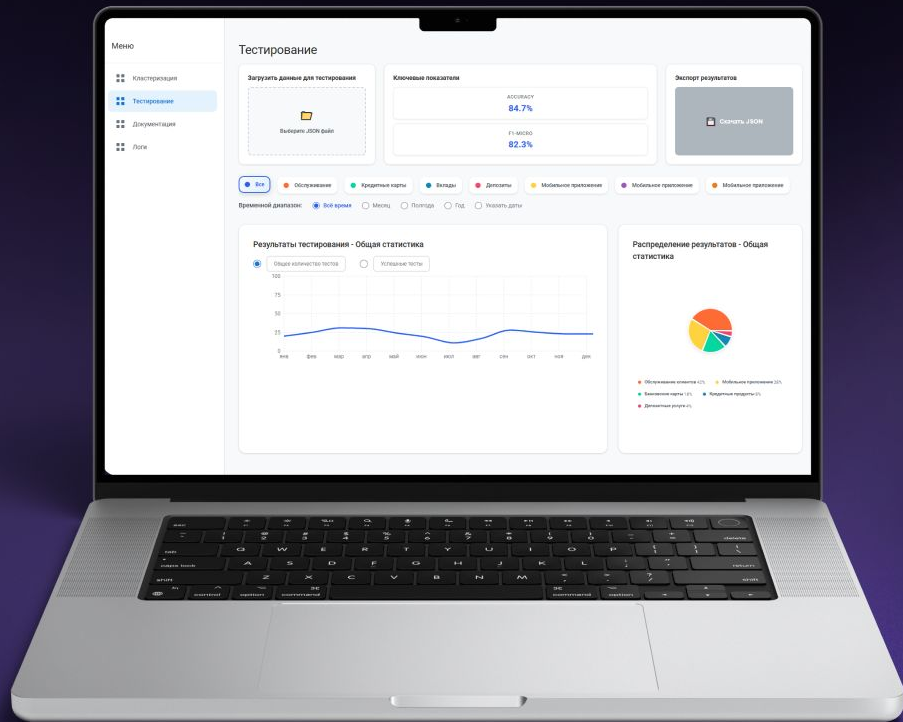
Предоставляет удобный интерфейс для анализа отзывов и отслеживания динамики их изменения

## Фильтры

Поддерживается фильтрация данных по кластерам, темам и временным периодам, что позволяет глубже понимать структуру обратной связи

## Тестирование

Система даёт возможность проверять корректность обработки данных и сценариев работы, обеспечивая точность аналитики



# ML модель и стек

2 отзыва  
в секунду

в среднем



## Проработанный датасет

banki.ru 24000

sравни.ru 1000

12 категорий  
для  
классификации



## AI агент

Построен с помощью 2-х  
фреймворков

LangChain

LangGraph

```
MISIS x MISIS

workflow = StateGraph(ClassificationState)

workflow.add_node("classify_category", classify_category)
workflow.add_node("classify_sentiments", classify_sentiments)

workflow.add_edge(START, "classify_category")
workflow.add_edge("classify_category", "classify_sentiments")
workflow.add_edge("classify_sentiments", END)

classification_agent = workflow.compile()
```

## Наименование задачи

Сервис для отслеживания динамики клиентских настроений и проблем по конкретным банковским продуктам в режиме реального времени

## Описание решения

Цель нашего решения – создать максимально удобный и полезный интерактивный сервис для аналитиков отдела маркетинга. Классификация отзывов по темам и тональностям выполняется с помощью большой языковой модели Gemma-3-27b-it, что показывает высокую точность

Использование LLM по API гарантирует независимость от вычислительных ресурсов, а также приемлемую скорость выполнения. Главное в нашем проекте — простота и удобство: интуитивно понятная архитектура и наглядная визуализация дашборда.

## Как планируем развивать

В дальнейшем мы планируем расширить возможности сервиса – добавить новые функции и улучшить старые. После внедрения продукта в экосистему компании, стоит собрать фидбек пользователей, что позволит нам выявить возможные проблемы и устранить их



# О команде

## Краткая история команды

Никита, Влад и Богдан участвую вместе на хакатонах не впервые. Они познакомились с самого поступления в НИТУ МИСИС, где обучались в одной академической группе. Участвуют в различных хакатонах и создают различные проекты студенты уже больше года.

Остальные участники команды были найдены благодаря общеуниверситетскому тематическому чату в телеграмме по участию в хакатонах

## Почему выбрали именно эту задачу

Задача нам показалась наиболее интересная с точки зрения нашего стека и опыта. ML-инженеры команды уже не раз имели дело с NLP и LLM, а остальные участники участвовали в похожих проектах до этого. Также данный кейс заинтересовал нас своей нестандартной постановкой задач. Например, самостоятельно определить список категорий для отзывов, визуализировать результаты двухэтапной классификации. Наиболее вдохновляющей мыслью для нас было желание создать продукт, который облегчил бы работу множества сотрудников

## Сложности и вызовы

Сначала мы пытались создать решение на базе легковесных моделей-классификаторов, но, не добившись желаемого результата, пришлось полностью переделывать ML-часть сервиса, когда уже прошло больше половины хакатона. Также было потрачено много сил на создание наиболее наглядного дашборда – мы рассматривали данную задачу со стороны пользователей и постоянно находили моменты, которые стоило доработать

# Команда



Python

LLM

## Team Lead, ML

**Никита**  
Мещеряков

@no\_status\_now



LangGraph

LangChain

Transformers

## ML

**Владислав**  
Елдашов

@avolxn



React

Product Analyst

## Frontend

**Артем**  
Зебелян

@goodzone\_z



FastAPI

SQL

Docker

## Backend

**Богдан**  
Камышников

@elsayanara



Figma

Adobe

## UI/UX Designer

**Артем**  
Брежнев

@dewerr