



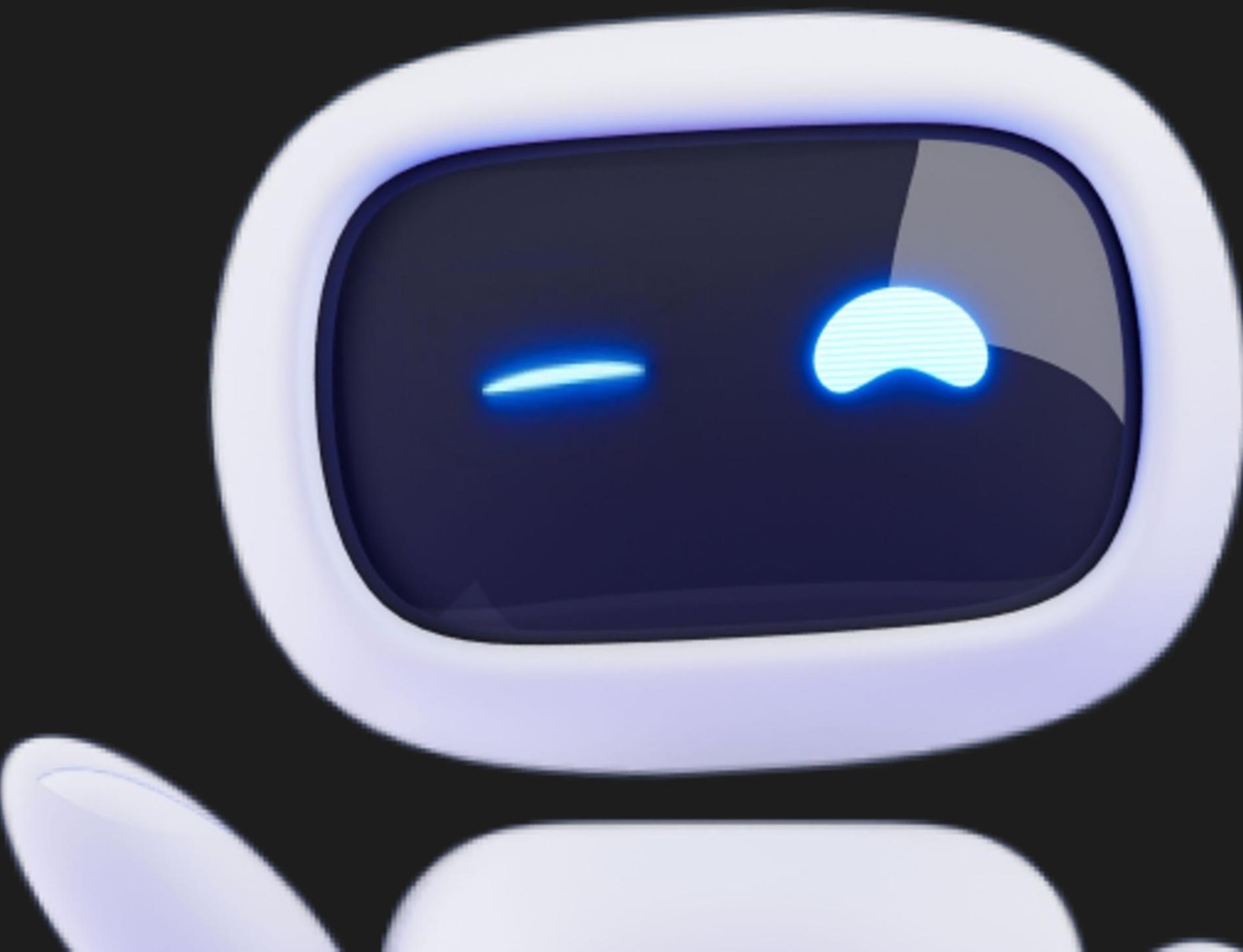
ГАЗПРОМБАНК



AsoFast

# Хакатон TechFriend AI Challenge

projectX MISIS

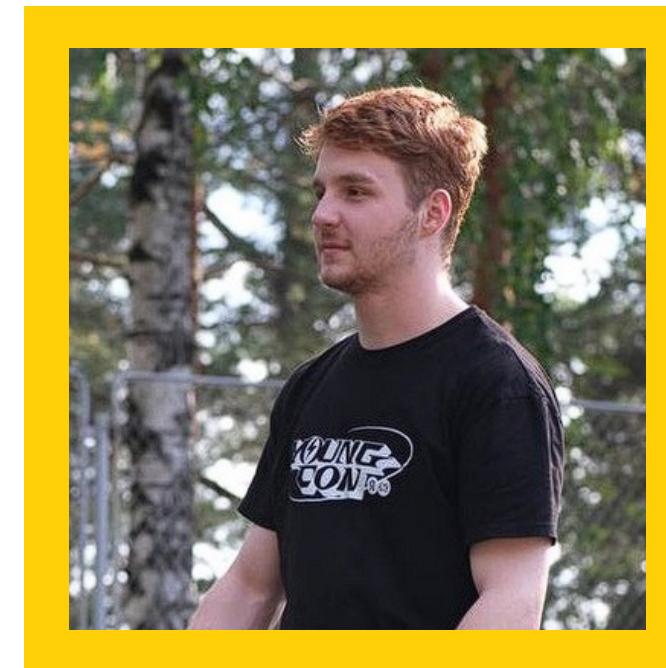


# Команда проекта



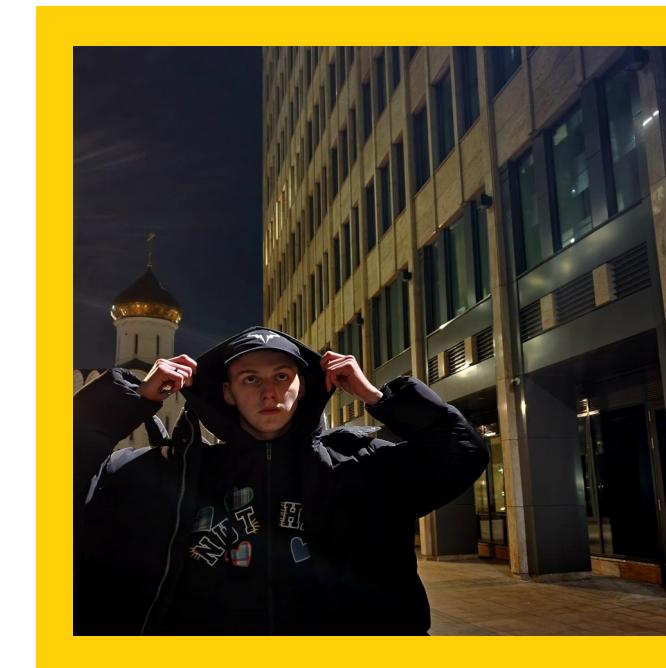
Дмитрий  
Уросов

СТУДЕНТ НИТУ МИСИС



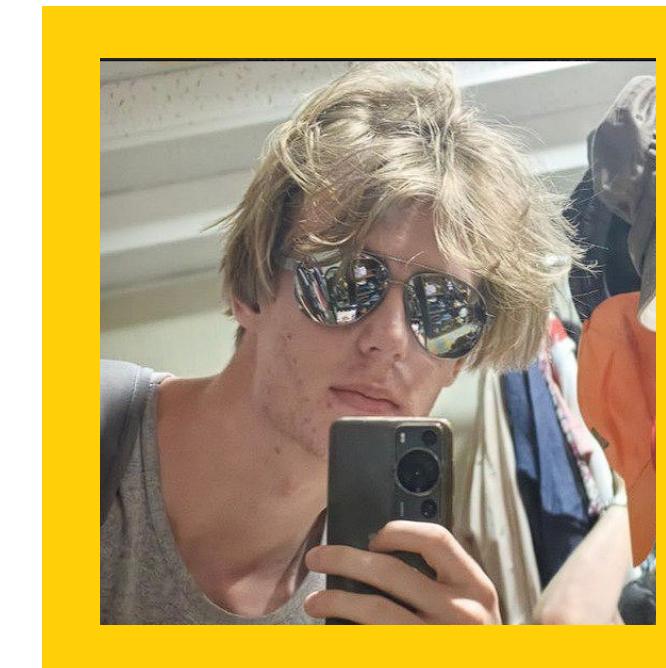
Айдар  
Каримов

СТУДЕНТ НИТУ МИСИС



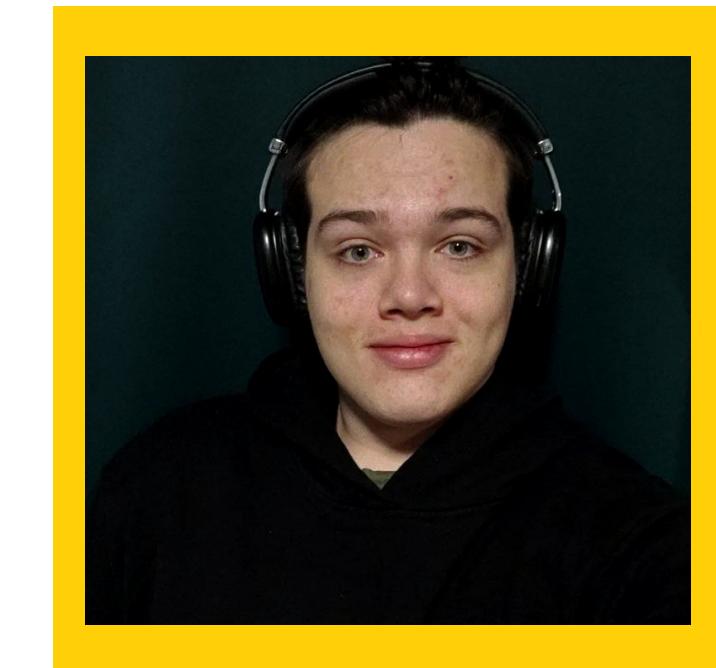
Тимофей  
Суник

СТУДЕНТ НИТУ МИСИС



Родион  
Оркин

СТУДЕНТ НИТУ МИСИС



Дамир  
Красильников

СТУДЕНТ СПО ИКТЗИ КИТ



Ирина  
Давыдова

СТУДЕНТ МГППУ

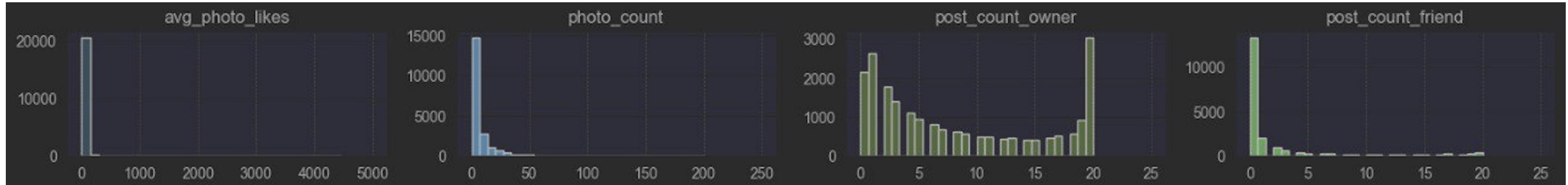
# Анализ данных: распределение



До:



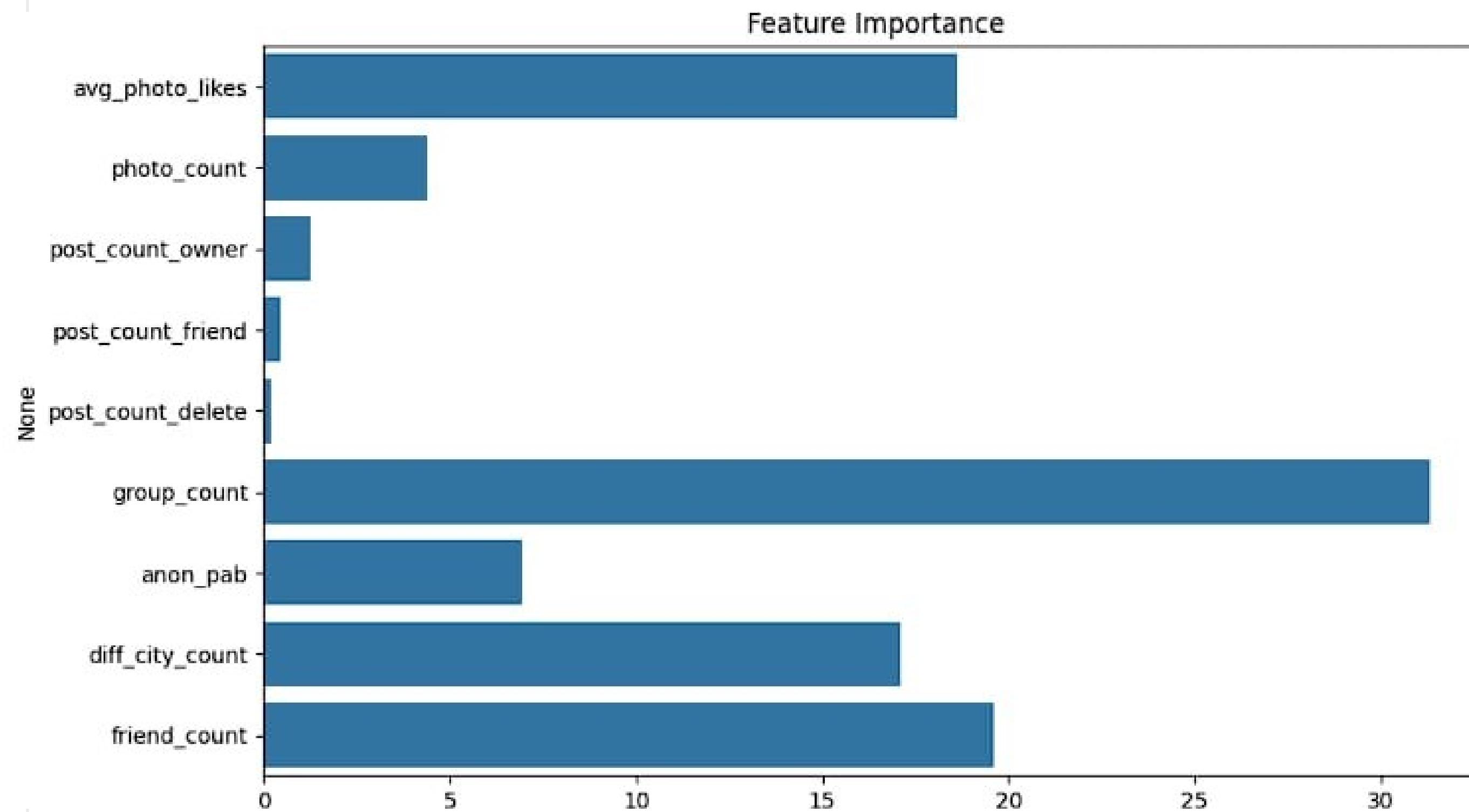
После:



# Разметка для экстравертов/интровертов

## Использование исходных данных

часть данных берется в изначальном виде, нормализуется и используется для обучения



## Создание новых признаков

Часть исходных данных используется для выведения новых признаков

## Обработка данных

Текстовые данные проходят очистку, нормализацию и переводятся в эмбеддинги  
Числовые данные обрабатываются. На их основе создаются новые признаки

## Ближайший кластер для пользователя

### Интроверт

Cluster 0 - Closest User:  
avg\_photo\_likes -0.491877  
photo\_count -0.178747  
post\_count\_owner 0.250000  
post\_count\_friend 0.000000  
post\_count\_delete 0.000000  
group\_count -1.112347  
anon\_pab 0.000000  
diff\_city\_count -1.276473  
friend\_count -1.205017

### Экстраверт

Cluster 1 - Closest User:  
avg\_photo\_likes 0.346022  
photo\_count 0.251930  
post\_count\_owner 0.300000  
post\_count\_friend 0.000000  
post\_count\_delete 0.000000  
group\_count 0.580301  
anon\_pab 1.000000  
diff\_city\_count 0.341617  
friend\_count 0.470334

# Типология

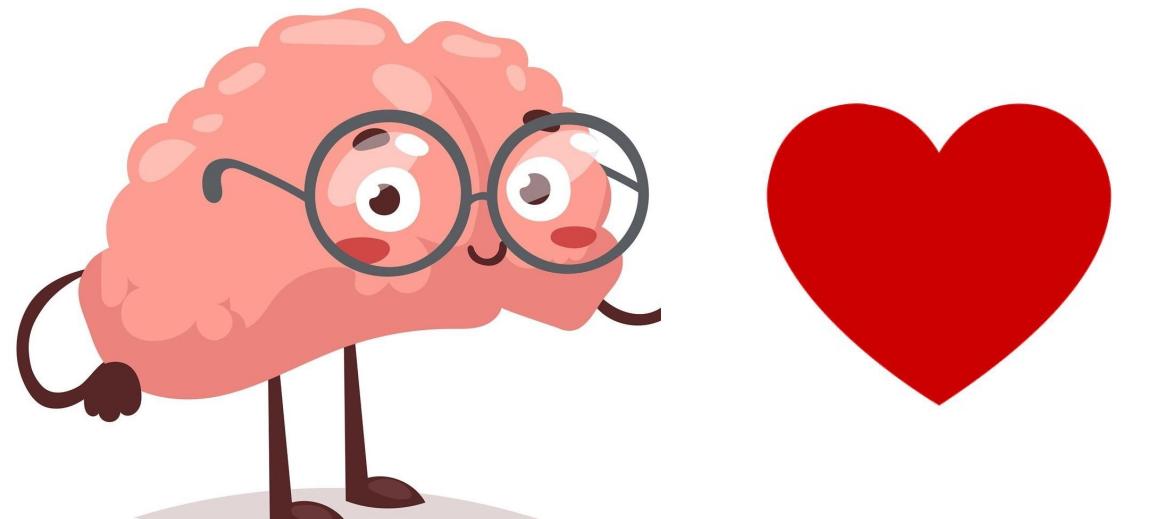
**Социальное  
направление:**

Экстраверт/ Интроверт



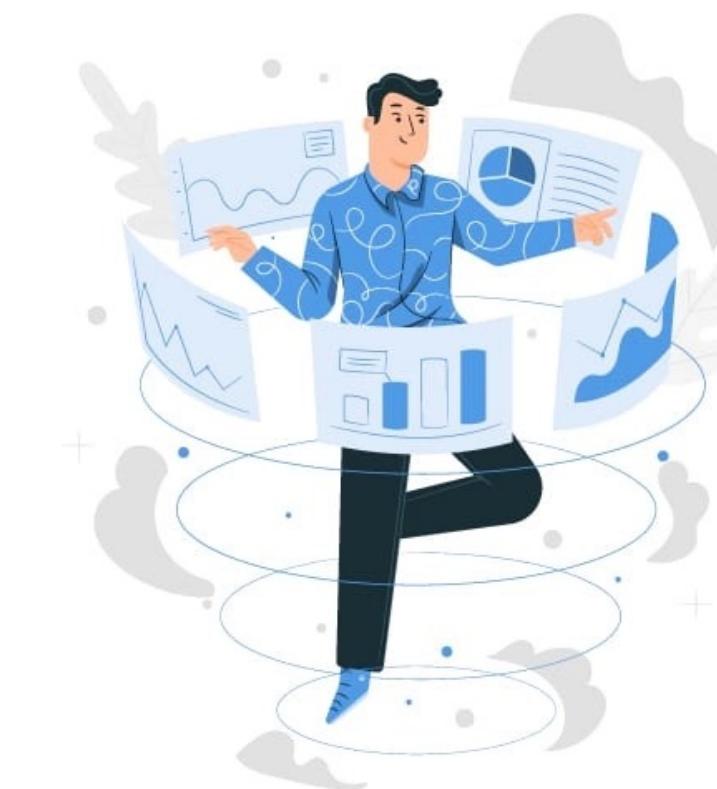
**Способ принятия  
решений:**

Аналитический/  
Чувственный



**Когнитивный стиль:**

Ригидность/Гибкость

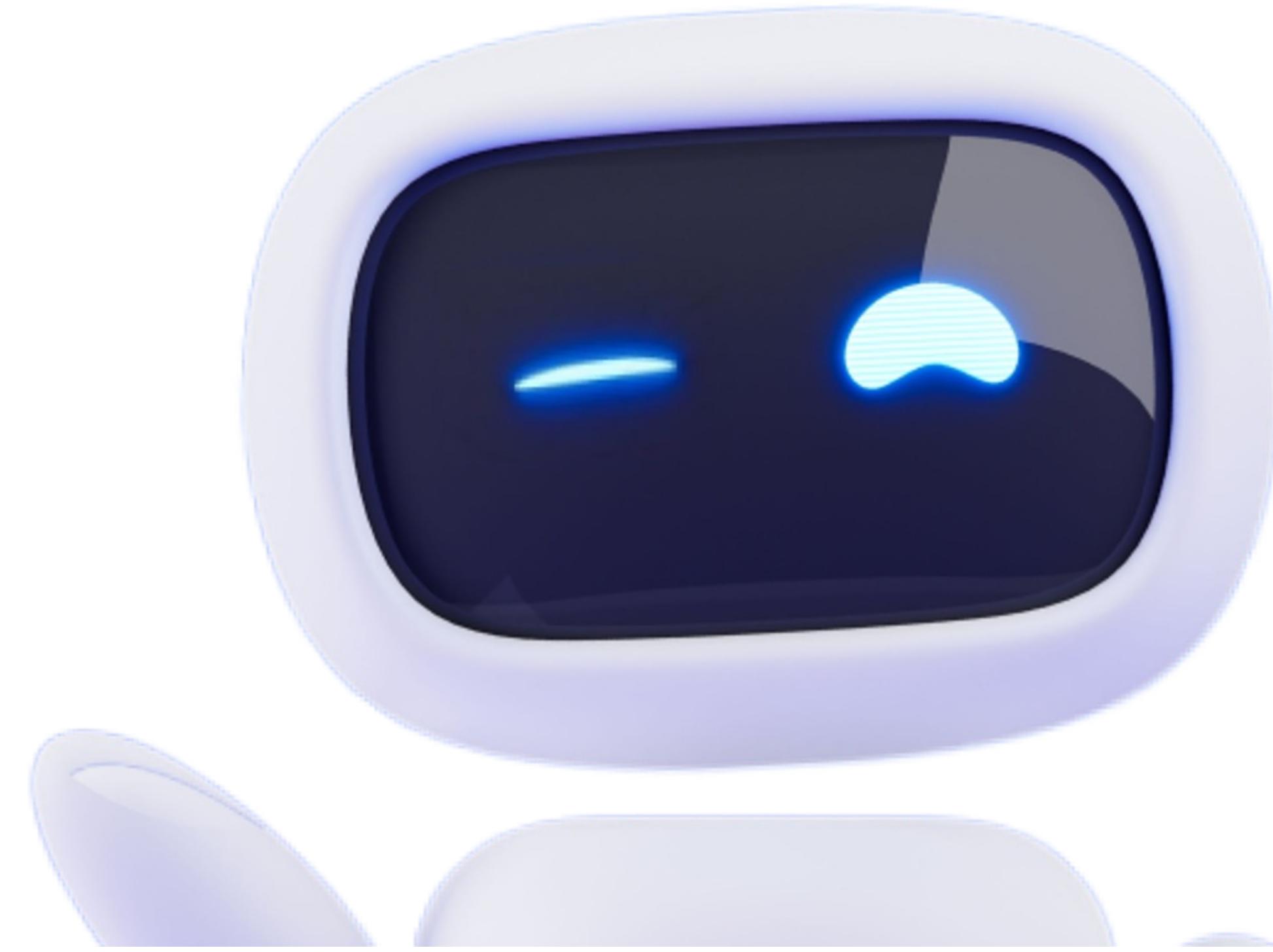


# Задача хакатона

Разработка алгоритма, способного предоставить глубокое понимание пользователя на основе его активности в социальных сетях, а именно, определить его психологический тип личности. Данная разработка позволит значительно персонализировать общение виртуального помощника в проекте Виртуальный друг.

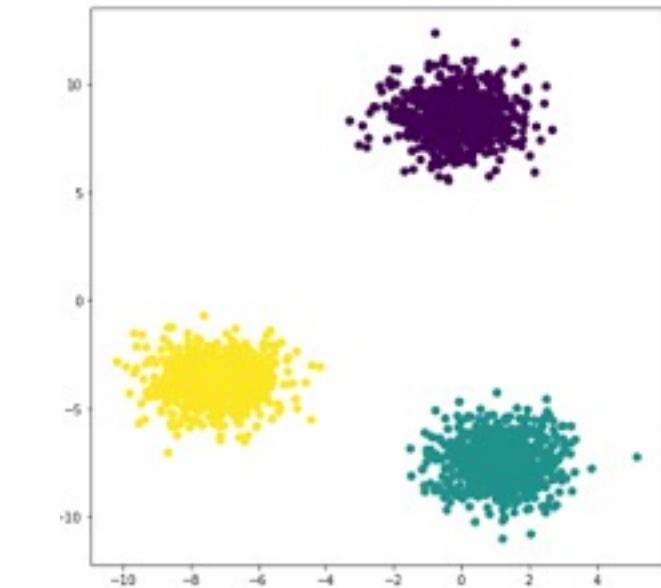
- Привет, я вчера победил на хакатоне!

-Отличная работа! Ты действительно справился с этой задачей на высоком уровне. Продолжай в том же духе!



# 3 варианта решений

## 1. кластеризация



### 3 целевые переменные

- социальное направление
- способ принятия решений
- когнитивный стиль

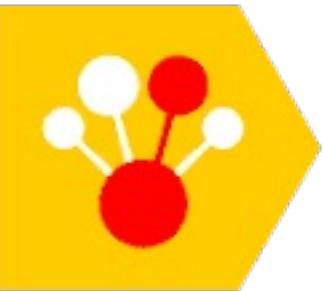
### Итоговый результат

1/2 + Э/И + А/Ч + Р/Г

### Время обучения

3 секунды

## 2. градиентный бустинг



CatBoost

catboost

### Итоговый результат:

91% Acc

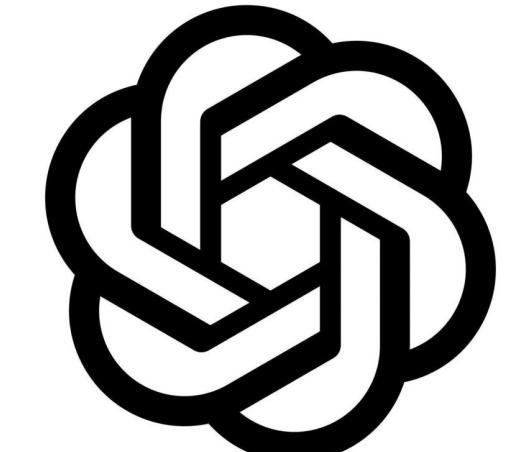
### Время обучения:

30 секунд

## 3. I'm using

Gemini

LLaMA  
by Meta



### предобученная модель

- высокая точность
- высокая стоимость



### Итоговый результат

98% Acc

### Время обработки данных:

100 юзеров/мин

# Неочевидные признаки

## Хештеги

часть данных берется в изначальном виде, нормализуется и используется для обучения

a - кол-во постов с #

b - общее кол-во постов

a1 - общее кол-во #

$$\Gamma(z) = (a)/(b) \cdot (1 - (a)/(a1)) .$$

## Эмоциональность

Анализ токсичности + эмоциональной тональности постов пользователя

## Активность пользователя

Среднее время между постами пользователя

```
# Загрузка токенизатора и модели
tokenizer = BertTokenizer.from_pretrained('s-nlp/russian_toxicity_classifier')
model = BertForSequenceClassification.from_pretrained('s-nlp/russian_toxicity_classifier')
model.to("cpu")
```

|   | user_id | mean     |
|---|---------|----------|
| 0 | 2       | 0.028571 |
| 0 | 3       | 0.027778 |
| 0 | 4       | 0.010000 |
| 0 | 5       | 0.011905 |
| 0 | 6       | 0.020833 |
| 0 | 7       | 0.009259 |

# Telegram bot



Бот разработан для  
помощи определения  
психотипа личности

## Стек

AIogram



```
@dp.message_handler(lambda message: message.text == "Аналитик - Чувствительный")
async def show_analytic_sensitive(message: types.Message):
    # Укажите путь к вашему изображению
    photo_path = "analytic_sensitive_chart.jpg"
    with open(photo_path, 'rb') as photo:
        await message.answer_photo(photo=photo, caption="График распределения Аналитика - Чувствительного")

# Обработка кнопки "Ригидный - Гибкий"
@dp.message_handler(lambda message: message.text == "Ригидный - Гибкий")
async def show_rigid_flexible(message: types.Message):
    # Укажите путь к вашему изображению
    photo_path = "rigid_flexible_chart.jpg"
    with open(photo_path, 'rb') as photo:
        await message.answer_photo(photo=photo, caption="График распределения Ригидного - Гибкого")

# Старт сессии - выбор метода проверки
@dp.message_handler(lambda message: message.text == "Старт сессии")
async def start_session(message: types.Message):
    await message.answer(text="Выберите метод проверки:", reply_markup=check_keyboard)

# Обработка кнопки "В главное меню"
@dp.message_handler(lambda message: message.text == "Главное меню")
async def back_to_menu(message: types.Message):
    await message.answer(text="Возвращение в главное меню.", reply_markup=menu_keyboard)

@dp.message_handler(lambda message: message.text == "Проверка по ID пользователя")
async def check_by_id(message: types.Message):
    await message.answer(text="Выберите модель для предсказания:", reply_markup=mode_keyboard)

@dp.message_handler(lambda message: message.text == "K-means")
async def check_by_id(message: types.Message):
    await message.answer(text="Введите ID пользователя (только числовой формат):", reply_markup=process_id)

model_type = 'None'
process_id() > elif (model_type == 'lstm') > try > if not user_data.empty:
```

# Разметка для 2(А/Ч) и 3(Р/Г) целевых

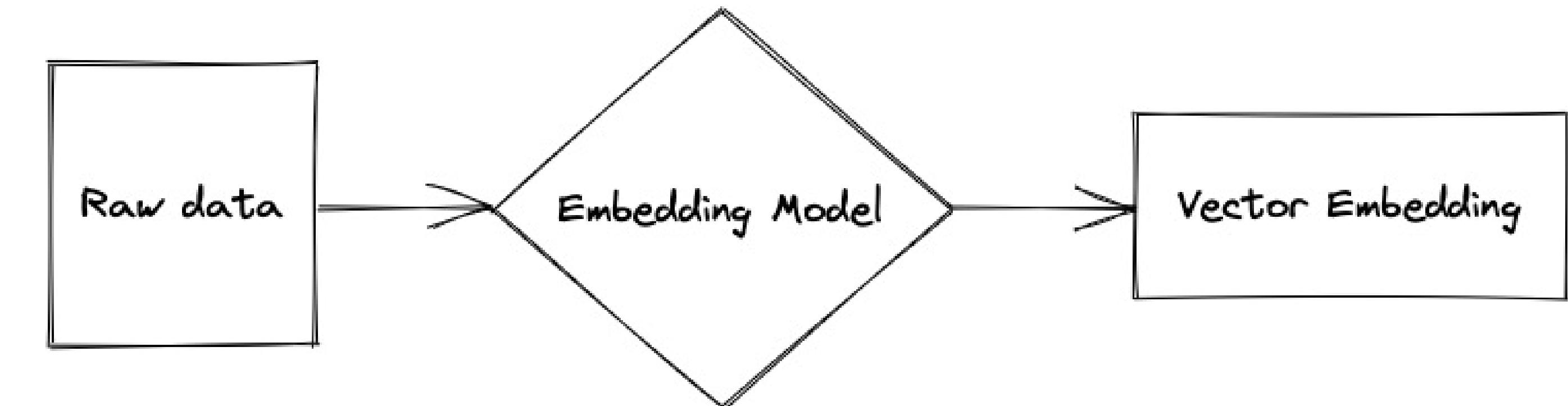
## Вторая целевая

возраст, пол, разница в интересах пользователя



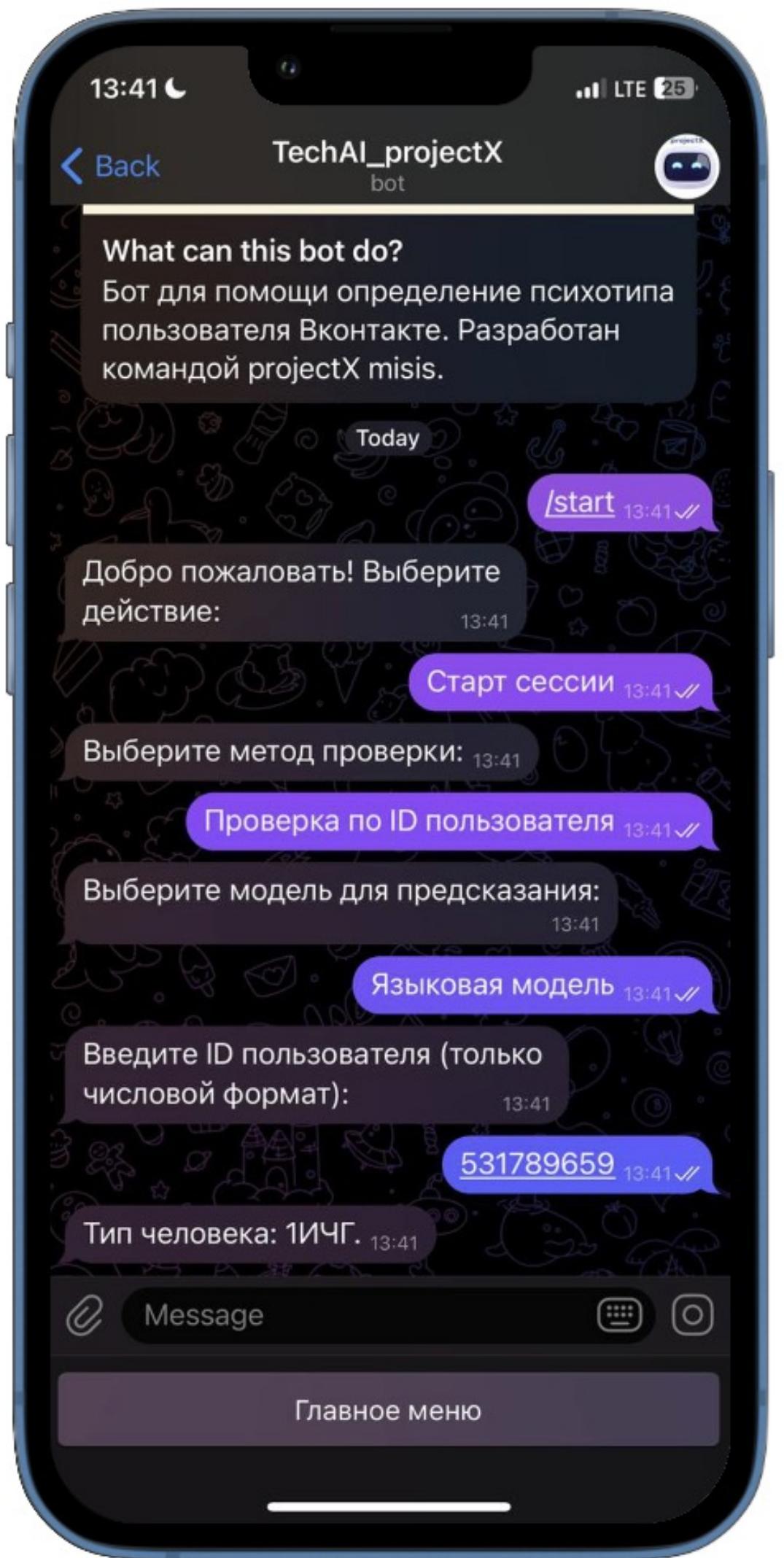
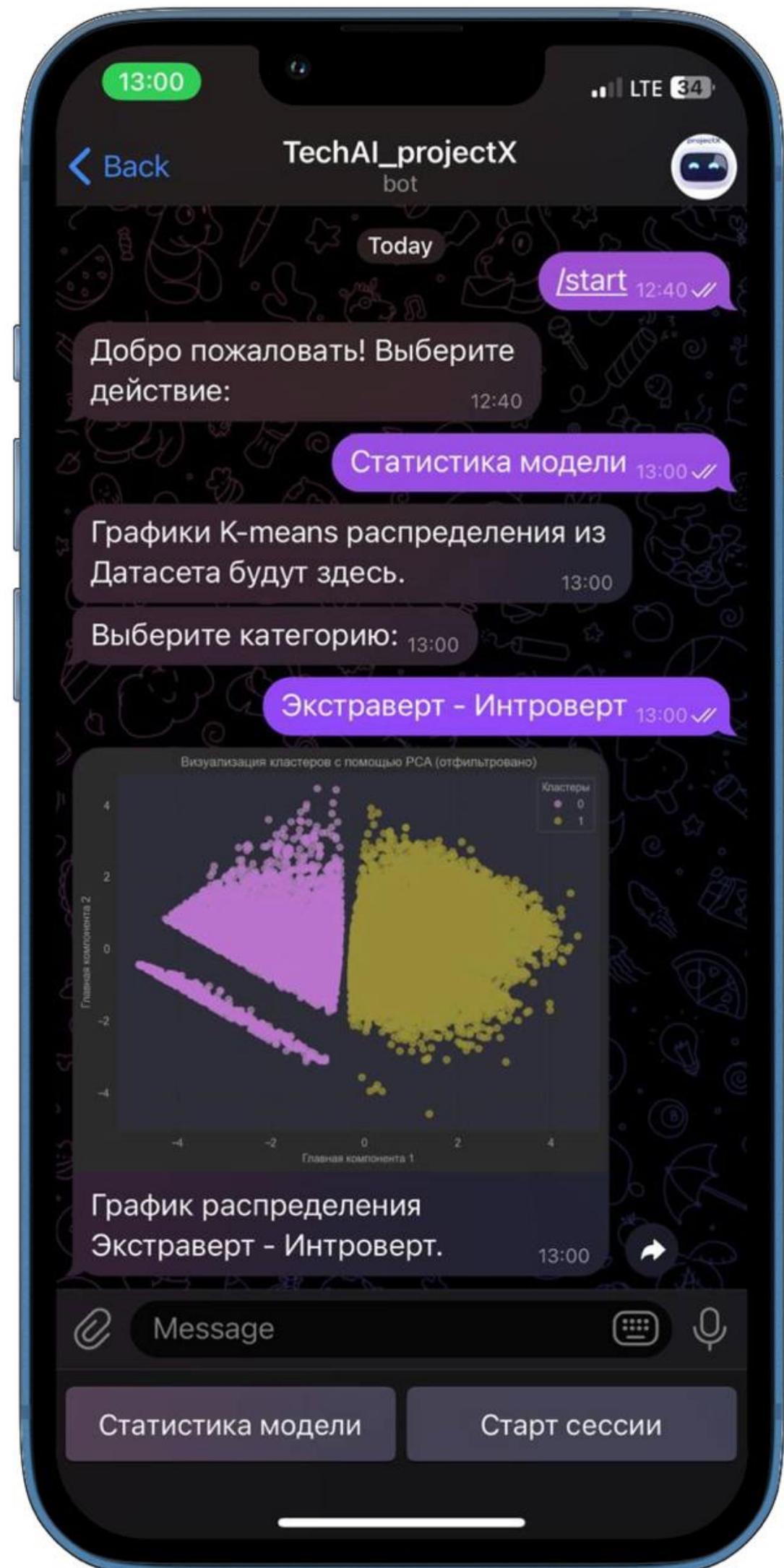
## Третья целевая

усредненные эмбеддинги групп пользователя, avg  
расстояние между группами пользователя, усредненные  
эмбеддинги постов пользователя, avg расстояние между  
постами пользователя

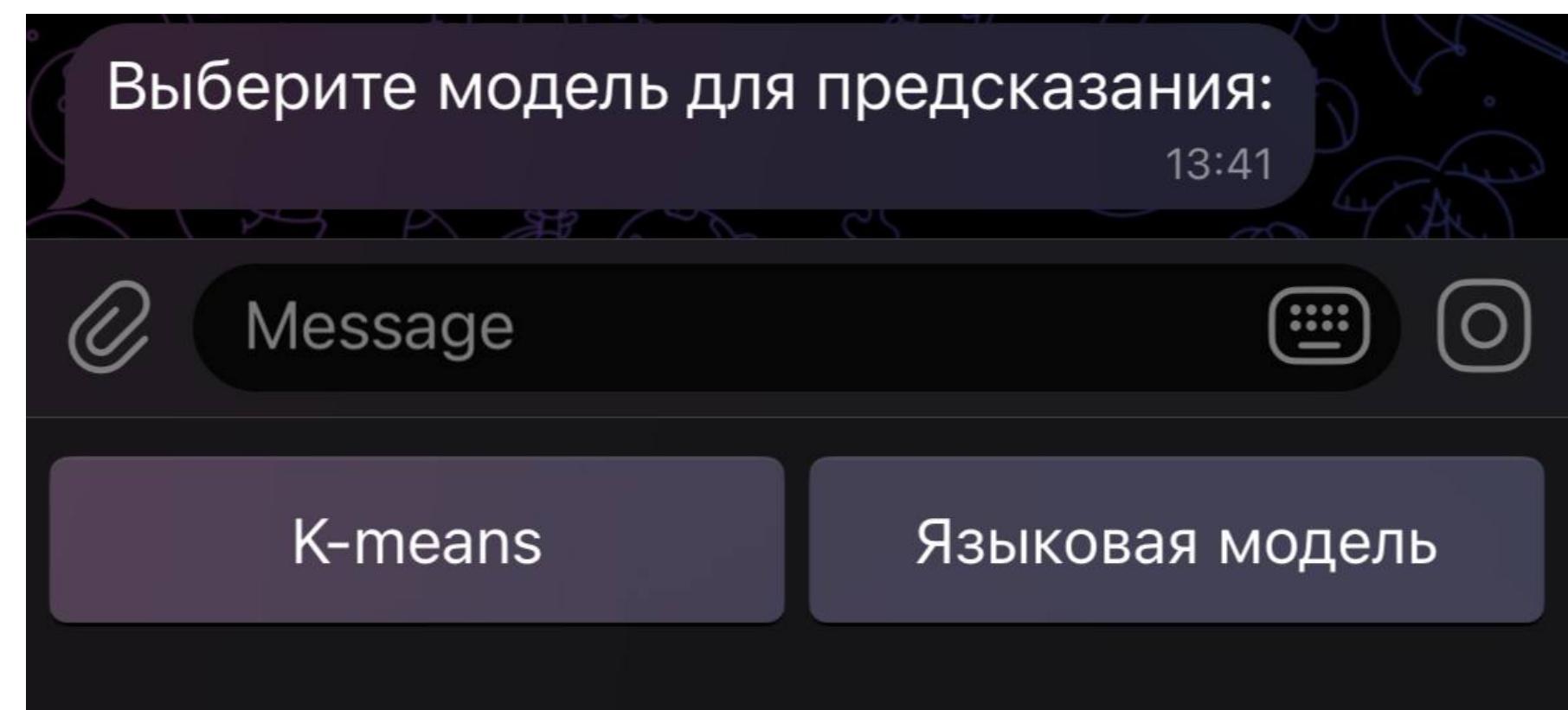


# Функционал бота

## Вывод различных статистик



## Выбор модели для предсказания



# Советы по улучшению