

Краудфандинг

**исследование с применением методов компьютерной
ЛИНГВИСТИКИ**

Лиза Кадетова, 2020

Данные

собственный датасет

завершенные благотворительные сборы на planeta.ru

2000+ строк

900.000+ словоупотреблений

30.000+ слов-признаков (леммы, без стоп-слов, включая ситуационные)

Структура

ранее выполненные задачи

- Сбор данных (Python)
- Описательная статистика (Python, R)
- Поиск зависимости между частотностью слов и результатом сбора (R)
- Кластеризация с помощью ML (Python)
- Разметка вручную поверх машинной разметки
- Обучение классификатора для определения темы текста (Python)

Вопросы

текущего этапа

1. Можно ли улучшить качество предсказания темы
2. Какие темы успешнее других
3. Есть ли сезонность
4. Можно ли предсказать точную сумму
5. Адресные и неадресные сборы: что успешнее
6. Географический охват сборов

Структура

задачи текущего этапа

1. Аугментация данных (Python, imblearn)
2. Дисперсионный анализ успеха по категориям и по месяцам (R)
3. Обучение предсказанию непрерывной переменной (Python, sklearn)
4. NER: извлечение имен собственных (Python, DeepPavlov)
5. Автоматическая разметка текстов на адресные и неадресные (Python, fuzzywuzzy)
6. Дисперсионный анализ успеха по критерию адресности (R)
7. NER: извлечение топонимов (Python, DeepPavlov)
8. Визуализация географического охвата (Google maps)

Аугментация данных

улучшит ли качество предсказания

- данные несбалансированы
- under-sampling: ↓
мажоритарного класса
- over-sampling: ↑
миноритарного класса,
создание доп. данных из
имеющихся

```
for k in np.unique(df['Category']):  
    volume = len(df[df['Category']==k])  
    volume_percent = round(volume*100/2038)  
    print(f'{volume} = {volume_percent}% {k}')
```

85 = 4%	активизм_просвещение_профилактика
37 = 2%	бездомные_кризис
71 = 3%	взрослые_лечение_реабилитация
535 = 26%	дети_лечение_реабилитация
245 = 12%	животные
6 = 0%	заключенные
66 = 3%	малоимущие_бедность
92 = 5%	мечты_подарки_праздники
10 = 0%	наука_история_культура
57 = 3%	паллиатив_уход
44 = 2%	пожилые_ветераны
84 = 4%	развитие_нко_инфраструктура
28 = 1%	семейный_кризис
262 = 13%	сироты_дети_из_неблагополучных семей
395 = 19%	социализация_возможности
21 = 1%	экология

Over-sampling

imblearn

- разные способы: случайное дублирование экземпляров, искусственная генерация
- RandomOverSampler, SMOTE, ADASYN
- f1_score weighted
- + k-fold кросс-валидация

```
kf = KFold(n_splits=15) # вызов KFold, количество сл
kf.get_n_splits(X) # разбиваем
#print(kf)
for train_index, test_index in kf.split(X): # учим
    #print("TRAIN:", train_index, "TEST:", test_index)
    kf_X_train, kf_X_test = X[train_index], X[test_index]
    kf_y_train, kf_y_test = y[train_index], y[test_index]
    kf_accuracy = train_model_2(LogisticRegression(n
    print(kf_accuracy)
```

```
0.7302526251017173
0.8273778167983612
0.8287180255217856
0.6985739750445632
0.8754106187929719
0.745771633358549
0.7648602268050531
0.7733937117781529
0.7829275681903968
0.7035525228995968
0.7740292652956718
0.6594350277697194
0.719034527150685
0.6897324170400265
0.5854170021258628
```

```
CPU times: user 18min 2s, sys: 2min, total: 20min 3s
Wall time: 7min 5s
```


Результат

влияние over-sampling на качество предсказания

- 15 раундов k-fold: weighted f1-score от 59% до 89% - независимо от over-sampling
- лучшие показатели (макс. 89%) случайны: в результате перезапуска одной и той же комбинации векторизатора, сплита и классификатора, а не за счет контролируемого изменения отдельных настроек, насколько модели “повезло на экзамене”
- итог: нужно работать с датасетом: добирать данные или пересматривать категории

Тест

на внешних данных

- 10 текстов с других площадок
- результат: 9 из 10
- ошибка: сбор на мемориальную доску А. С. Макаренко определен как помощь детям (кто бы не запутался)

	text	link
0	Лариса Ермошина живет в городе Донской Тульско...	https://dobro.mail.ru/projects/zhizn-dlya-lari...
1	Удар был такой силы, что Шанс встать уже не см...	https://dobro.mail.ru/projects/rejdom-vsem-bud...
2	«Доброе утро, Любовь Алексеевна! Вот молоко, т...	https://dobro.mail.ru/projects/ne-bojsya-ya-s-...
3	«Человек-радио» – так в шутку называют Сашу в ...	https://dobro.mail.ru/projects/podarit-dyihani...
4	«Почему листья зеленые?», «Что такое железо?»,...	https://dobro.mail.ru/projects/preodolet-izoly...
5	В марте 2017 года студент – медик Евгений Косо...	https://boomstarter.ru/projects/dm/drugaya_med...
6	2020 год Указом Президента РФ объявлен Годом п...	https://boomstarter.ru/projects/Serdze88/roves...
7	Меня зовут Максим Батырев, я автор книг-бестсе...	https://boomstarter.ru/projects/591685/memoria...
8	Всем привет! Мы – движение "Молодёжь за мир".\...	https://boomstarter.ru/projects/914463/obedy_d...
9	Здравствуй, я – Сергей Богдановский, руковож...	https://boomstarter.ru/projects/taganai89/sozd...

```
boom.shape
```

```
(10, 2)
```

```
# просим модель определить категории для всех текстов в новом датасете
```

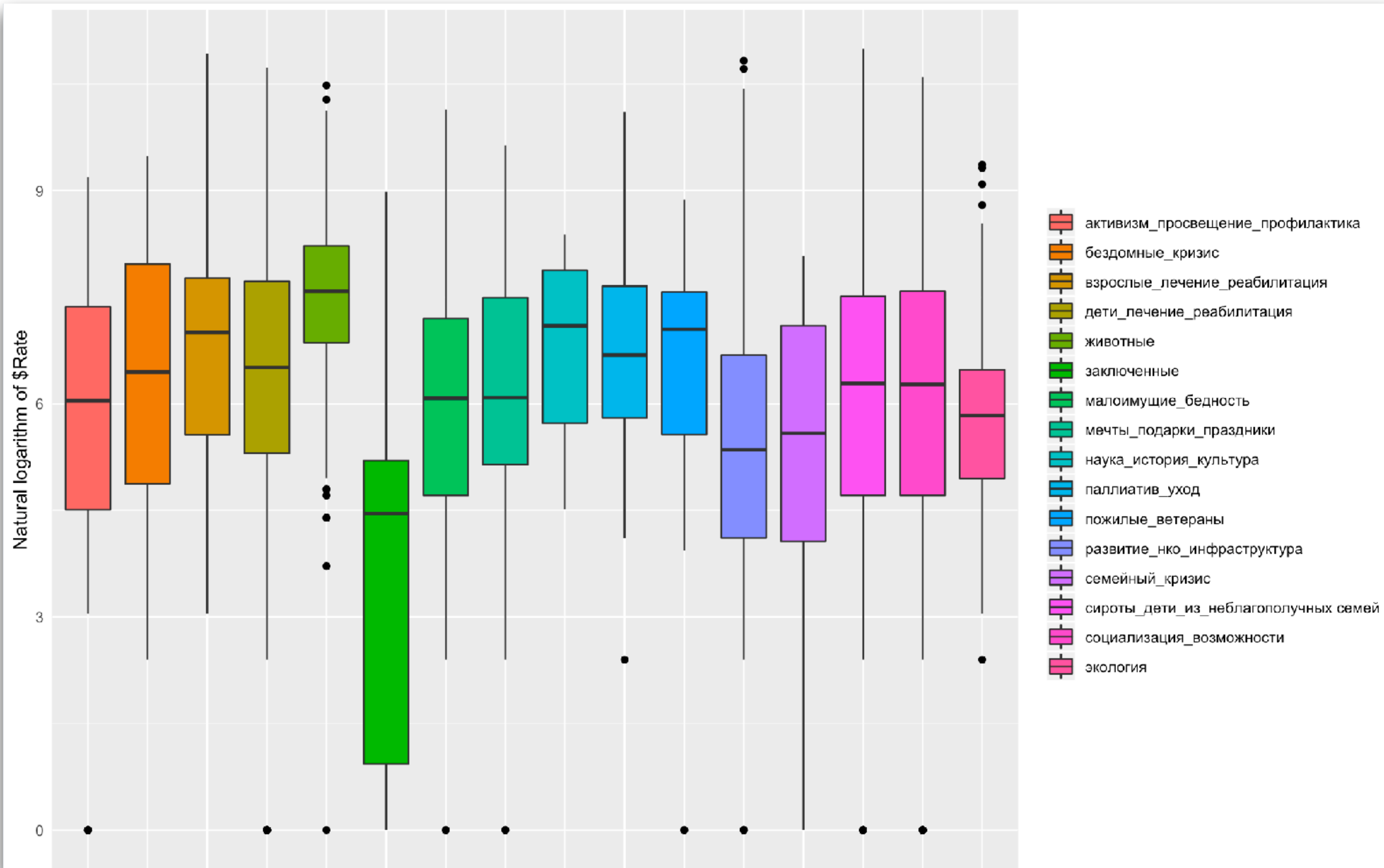
```
cats = clf.predict(boom['text'])  
cats
```

```
array(['взрослые_лечение_реабилитация', 'животные', 'пожилые_ветераны',  
      'дети_лечение_реабилитация', 'социализация_возможности',  
      'бездомные_кризис', 'пожилые_ветераны',  
      'дети_лечение_реабилитация', 'бездомные_кризис',  
      'социализация_возможности'], dtype=object)
```

16 категорий

какая лучше собирает деньги?

- визуализация
- однофакторный дисперсионный анализ:
 - ANOVA: есть требования к данным
 - критерий Краскела-Уоллиса - непараметрический аналог anova

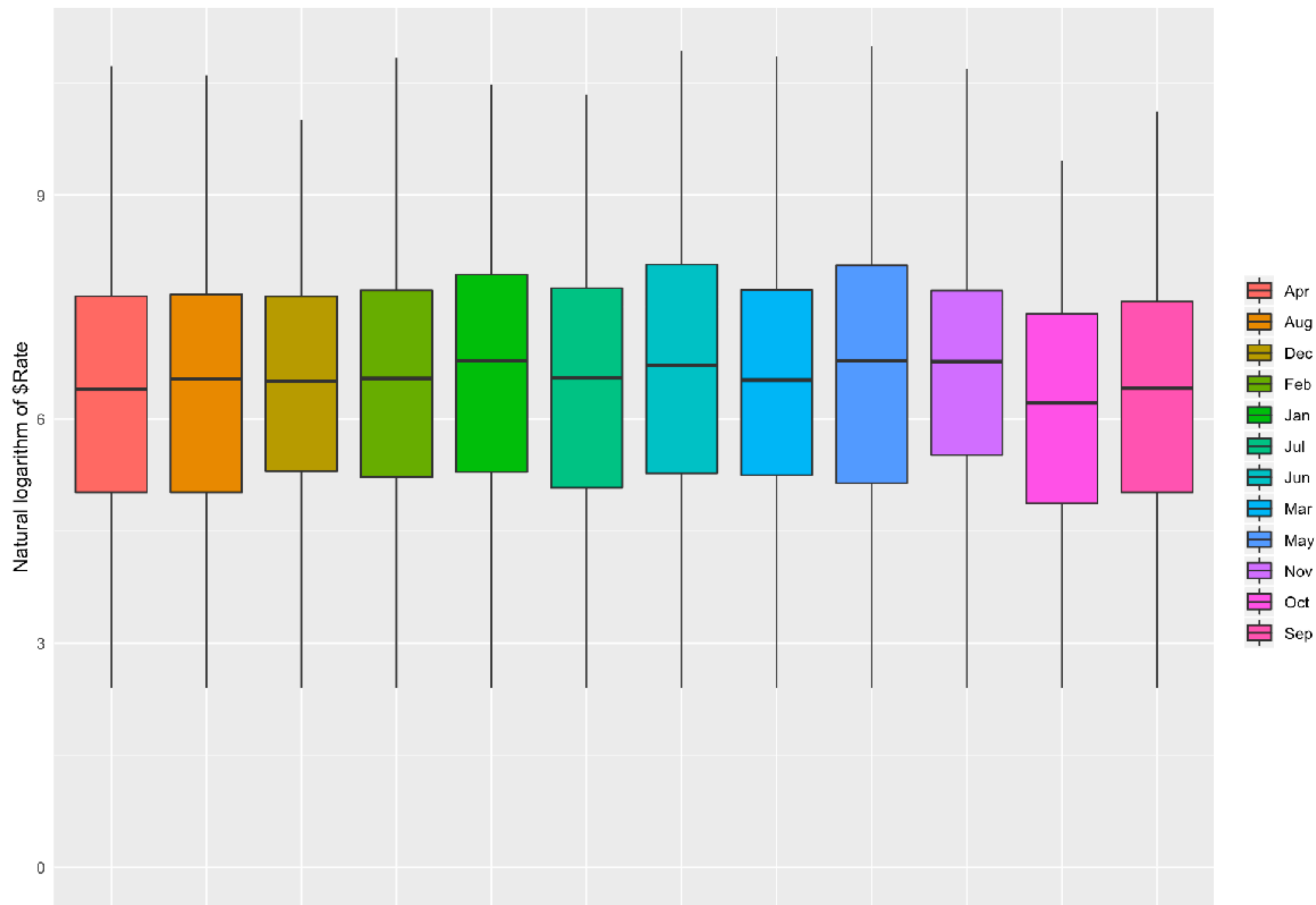


Категория-победитель	За чей счет	Степень значимо сти	anova\ tukey	kruskal\du nn without outliers	kruskal\ dunn with outliers
животные	сироты_дети_из_неблагополучных семей	****	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
животные	социализация_возможности	****	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
животные	развитие_нко_инфраструктура	****	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
животные	дети_лечение_реабилитация	****	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
животные	малоимущие_бедность	****	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
животные	мечты_подарки_праздники	***	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
животные	активизм_просвещение_профилактика	***	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
животные	экология	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
животные	семейный_кризис	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
дети_лечение_реабилитация	развитие_нко_инфраструктура	**	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
пожилые_ветераны	развитие_нко_инфраструктура	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
взрослые_лечение_реабилитация	развитие_нко_инфраструктура	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
паллиатив_уход	развитие_нко_инфраструктура	**	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
животные	паллатив_уход	*	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
животные	взрослые_лечение_реабилитация	*	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Сезонность

зависит ли успех от месяца

- нет
- пригодились навыки работы со строками



Сумма по тексту

непрерывная перем. и ранги

- Rate = Result/Days
- для точной суммы - линейные модели и ансамбли: LinearRegression, Lasso, Ridge, RandomForestRegressor
- + сокращение вектора признаков
- → результат плохой
- с рангами намного лучше, но все равно не очень
- вопрос не столько прикладной, сколько теоретический: связь между собственно текстом и успехом сбора

	predicted	fact	difference
958	2605.32	1720.0	885.32
1772	2190.51	30.0	2160.51
1734	1695.33	100.0	1595.33
1270	1712.48	1240.0	472.48
1839	2624.10	60.0	2564.10
442	3051.03	780.0	2271.03
618	1428.83	760.0	668.83
259	2852.13	13520.0	-10667.87
1632	2489.21	180.0	2309.21
375	2702.06	240.0	2462.06

	precision	recall	f1-score	support
fast	0.70	0.17	0.27	96
moderate	0.48	0.42	0.45	171
slow	0.58	0.82	0.68	222
accuracy			0.55	489
macro avg	0.59	0.47	0.47	489
weighted avg	0.57	0.55	0.52	489

Адресные сборы успешнее неадресных?

- что это? а как определить?
- извлечение имен с DeepPavlov
- авторазметка по наличию имен плохая: мусор, нечеткие критерии, как посчитать количество употреблений имени
- \Rightarrow близость по расстоянию Левенштейна (fuzzywuzzy)
- результат хороший, хотя много случайных обстоятельств: порядок имен в тексте и т.д.: Сережа и мама Наташа

л', 'Евгений', 'Дмитрий', 'Константина', 'Максим', 'Гарик', 'Сергей', 'Сергей', 'Винил', 'Д
ксея', 'Максим', 'Александр', 'Александра', 'Александр', 'Лиля', 'Лилечка', 'Ребенка', 'Лил
р', 'Егорушка', 'Егора', ['Сергей', 'Миша', 'Миша', 'Михаил', 'Михаил', 'Светлана', 'Татья
ихаил'], ['Сергею', 'Сергей', 'Сергей', 'Сергей', 'Сергей', 'Сергей', 'Сергей', 'Сергея', 'С
дику', 'Вадику', 'Вадик', 'Вадима', 'Вадиму'], ['Стругацких'], [], ['Андрей', 'Аманом', 'Вари
Варвара', 'Вари', 'Злата', 'Варваре', 'Варвара'], ['Дарья'], ['Нодежды'], ['Ванечка', 'Ваня
а', 'Марвину', 'Гринго', 'Гринго', 'Гринго', 'Гринго', 'Уринари', 'Роял', 'Граф', 'Уринари',
си', 'Лаки', 'Бумеранга'], [], ['Никиты', 'Никиты', 'Леонтьевых', 'Никита', 'Никита', 'Никите
окрем', 'Медикомп', 'Октенисепт', 'Пронтосан', 'Пронтосан', 'Никите'], ['Диана', 'Диана', 'Д
, 'Марка', 'Марк', 'Марка', 'Марка', 'Марка', 'Марка', 'Марка', 'Марк', 'Марк', 'Марк', 'Марка', 'Ма
['Лёша', 'Аня', 'Анечка', 'Юра', 'Андрей', 'Сковорода', 'Ольге'], ['Деда'], ['Алеши', 'Алеша
аргарита', 'Ириной', 'Александрой', 'Никитой', 'Оксаной'], ['Арсений', 'Арсений', 'Гармонь',
, 'Арсений', 'Арсений', 'Арсений', 'Арсений', 'Арсений', 'Арсения', 'Сыроватский', 'Арсений',
, 'Клёпа'], [], ['Муравья', 'Муравья'], ['Дмитрий', 'Дмитрий', 'Дмитрий', 'Дмитрия', 'Дмитр
, ['Александром', 'Игорем', 'Юрия', 'Андрея', 'Бориса', 'Егору', 'Лиле', 'Егору', 'Саше', 'А
, 'Саши', 'Шевчук', 'Кинчев', 'Чернецкий', 'АЛИСА', 'КАЛИНОВ', 'ЧИЖ', 'ГАРКУША', 'АФФИНАЖ',
, 'СЕРГЕЙ', 'ЧИЖА', 'СЕРГЕЙ', 'АЛЕКСАНДР', 'А', 'С', 'С', 'А', 'ЛЕОНИД', 'ЛЕОНИДА', 'ЮРИЙ',
, 'ИЛЬИ', 'КНЯЗЯ', 'БАЛУ', 'АЛЕКСАНДР', 'БАЛУ', 'АРТЁМ', 'РОМАН', 'Романа', 'АЛЕКСЕЙ', 'Ал
антина', 'АНДРЕЙ', 'Андрея', 'ВАДИМ', 'Вадима', 'ОЛЕГ', 'АНДРЕЙ', 'Андрея', 'АЛЕКСЕЙ', 'Алек
'Алексея', 'Виктора', 'Алексея', 'АЛЕКСЕЙ', 'Максим', 'Олега', 'АЛЕКСАНДРА', 'ДМИТРИЙ', 'БРИ
IN', 'КИРИЛИН', 'Андрея', 'Максима', 'Вадим', 'ЮРИЙ', 'ЮРИЙ', 'Чердакова', 'Чижа', 'ВЛАДИМИР
, [], [], ['Аня'], [], ['Биллом'], ['Елена'], ['Деда'], [], ['Толмачев', 'Коршун', 'Тупало',
'Дарье', 'Дарья', 'Ивану', 'Иван', 'Юры', 'Пан', 'Ивану', 'Бочча', 'Бакаидов', 'Ивана', 'Дар
'Оксана', 'Роман', 'Вероника', 'Вероника', 'Димы', 'Ники', 'Диму', 'Димой', 'Дима', 'Максяко
'Север', 'Шани', 'Шаньку', 'Шаня', 'Шанька', 'Шани', 'Шаню', 'Шани'], ['Кольцова', 'Никите
оголовцевых', 'Элизабет', 'Марат', 'Андрей', 'Олеся', 'Мария', 'Алексей', 'Егор', 'Ирина',
'Максим', 'С', 'Сергей'], ['Лаки', 'Пильве', 'Трэвор', 'Люси', 'Серенити', 'Домра', 'Лаки',
'Б', 'Р', 'В', 'Власовой', 'Лиды', 'Алексеевской', 'Ириной', 'Катей', 'Панфиловой'], ['Егор',
а', 'Фроловых', 'Егор', 'Тим'], ['Тимура', 'Софи', 'Софи', 'Софи', 'Андреасян', 'Софи', 'Роз
'Дженнифер', 'Дарья', 'Софи', 'Софи', 'Софи', 'Софи', 'Софи', 'Софи'], [], ['Дэни', 'Гали
'Труфальдино', 'Даниэль', 'Олежка', 'Вела', 'Валерий', 'Олежке', 'Марта', 'Марточке', 'Макс
а', 'Валера', 'Ливадоновых'], ['Маша', 'Маша', 'Маша', 'Маше', 'Машиной', 'Машиной'], [], [
я', 'Валерий', 'Мария', 'Шалинцевых', 'Брайлем', 'Шалинцевых'], ['Воробьев', 'Павлик', 'Павл
'Снегурочкой', 'Булышев', 'Владимир', 'Владимир', 'Владимир', 'Владимира'], ['Исмаил', 'Елиза
['Деда'], ['Дима', 'Диму', 'Диминых', 'Димы', 'Дима', 'Димина', 'Димина', 'Сашу', 'Диминой'],
тиса'], ['Рустемом'], [], ['Фроська', 'Фроська', 'Фроське'], [], [], [], [], ['Мария', 'Арка
'], ['Александр', 'Алексей', 'Валера', 'Денис', 'Елена', 'Ирина', 'Ирина', 'Катя', 'Ольга',
ова'], [], [], [], ['Антон', 'Любови', 'Любовью', 'Наталья', 'Любовью', 'Антоном', 'Наталья
Андрею', 'Маше', 'Тане', 'Ивану', 'Илье'], ['Наташа', 'Вити', 'Вити', 'Вердника', 'Вити', 'В
Анна', 'Анну', 'Анна'], ['Илья', 'Илья', 'Илье', 'Илья', 'Илья', 'Илья', 'Илья', 'Илье', 'Иль
ьер', 'Грегори', 'Грегори', 'Грегори'], [], [], ['Константин', 'Константин', 'Константин',
антин', 'Константину', 'Константина'], ['Леониде'], [], ['Женя', 'Жени', 'Жени', 'Оксана',
'Жени', 'Жени'], ['Александр', 'Виктория', 'Маша', 'Вера', 'Полины', 'Коля', 'Аня'], ['Рябо
Улей', 'Ксюши', 'Ксюша', 'Юлю', 'Ксюша', 'Ксюши', 'Юли', 'Ксюши', 'Юли', 'Ксюша', 'Ксюше', 'И
'Ксюша', 'Юля', 'Малике', 'Егоре', 'Малика'], ['Света', 'Света'], ['Алёша', 'Эльвира', 'И
Алёша', 'Алёши', 'Алёша', 'Алёша', 'Эльвире', 'Алёше'], [], ['Иоанна', 'Нунан', 'Ретта', 'Ма
'Владик', 'Дима', 'ПедиаШур', 'НутриДринк', 'Дима', 'Дима', 'ПедиаШура', 'НутриДринка', 'Ну
'Маши', 'Владислава', 'Педиашура', 'НутриДринка'], [], [], [], [], ['Мурклуб', 'Гавпарка',
['Геркулес', 'Санокс'], [], ['Л', 'Королевой'], [], [], [], [], ['Николай'], [], ['Нелли
'Нелли', 'Надежда'], ['Юмашева', 'Вики', 'Вики', 'Вика', 'Вике'], [], [], ['Л', 'Миша', 'Миша
ускин', 'Пермяков'], ['Петуховой'], ['Йога', 'Йога', 'Джоанн'], [], [], [], ['Илья', 'Ксюшен
Андрей', 'Андрей', 'Андрей', 'Колесник', 'Макарова', 'Шепяков', 'Ларина', 'Рыженкова', 'Р

['Надежда', 'Островка', 'Островка']
similarity rate: 56



['Сергею', 'Сергей', 'Сергей', 'Сергей', 'Сергей', 'Сергей', 'Сергей', 'Сергей', 'Сергея', 'Сергея', 'Сергей', 'Сергея', 'Сергея']
similarity rate: 94

['Вадима', 'Вадюшка', 'Вадику', 'Вадику', 'Вадик', 'Вадима', 'Вадиму']
similarity rate: 78



['Алиса', 'Рентген', 'Лаки', 'Лаки', 'Лаки', 'Лаки', 'Бумеранга']
similarity rate: 52

['Диана', 'Диана', 'Дианочка', 'Дианы', 'Пауль', 'Диана', 'Дианы']
similarity rate: 60

['Марасакин', 'Марк', 'Марка', 'Марк', 'Марка', 'Марка', 'Марка', 'Марка', 'Марк', 'Марк', 'Марк', 'Марка', 'Марк', 'Марк', 'Марк', 'Марку', 'Марка', 'Марку']
similarity rate: 91

['Ани', 'Ани']
similarity rate: 100



['Алеши', 'Алеша', 'Алеша', 'Алеше', 'Лёше', 'Алеши']
similarity rate: 65

['Арсению', 'Арсению', 'Гармонь', 'Арсений', 'Арсения', 'Арсений', 'Арсения', 'Арсению', 'Арсений', 'Арсений', 'Арсений', 'Арсений', 'Арсений', 'Арсений', 'Арсений', 'Арсения', 'Сыроватский', 'Арсений', 'Арсению', 'Арсению', 'Арсения', 'Арсения']
similarity rate: 81

['Муравья', 'Муравья']
similarity rate: 100



['Дмитрий', 'Дмитрий', 'Дмитрий', 'Дмитрия', 'Дмитрий', 'Дарья', 'Дмитрий', 'Дмитрия', 'Дмитрия']
similarity rate: 78

['Север', 'Шани', 'Шаньку', 'Шаня', 'Шанька', 'Шани', 'Шаню', 'Шани']
similarity rate: 56

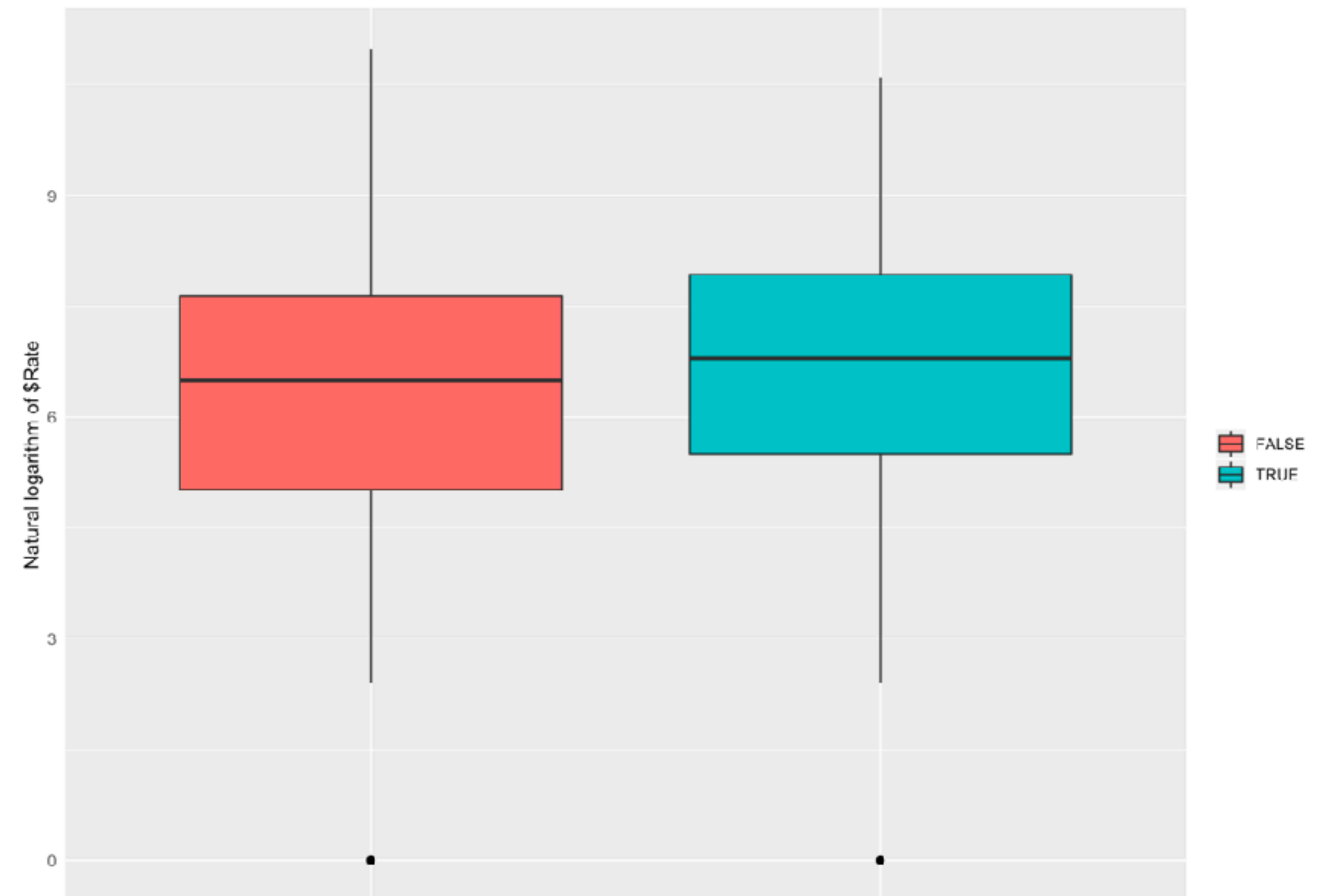
['Кольцова', 'Никиты', 'Никиты', 'Никита']
similarity rate: 61

['Тимура', 'Софи', 'Софи', 'Софи', 'Андреасян', 'Софи', 'Роза', 'Софи', 'Софи', 'Софи', 'Софи', 'Розы', 'Софи', 'Софи', 'Софи', 'Дженнифер', 'Дарья', 'Софи', 'Софи', 'Софи', 'Софи', 'Софи']
similarity rate: 62

Сравнение адресных и неадресных

- 1/4 данные
- критерий Вилкоксона (Манна-Уитни)
- да, адресные сборы успешнее неадресных
- но эффект небольшой

.y.	group1	group2	effsize	n1	n2	magnitude
Rate	FALSE	TRUE	0.07250259	1592	439	small



География по городам РФ

- извлечение топонимов с Deerpavlov
- проблемы: мусор, ошибки токенизации (Ростов-на-Дону, Великий Устюг и т.п.)
- чистка списка
- сопоставление со списком городов РФ

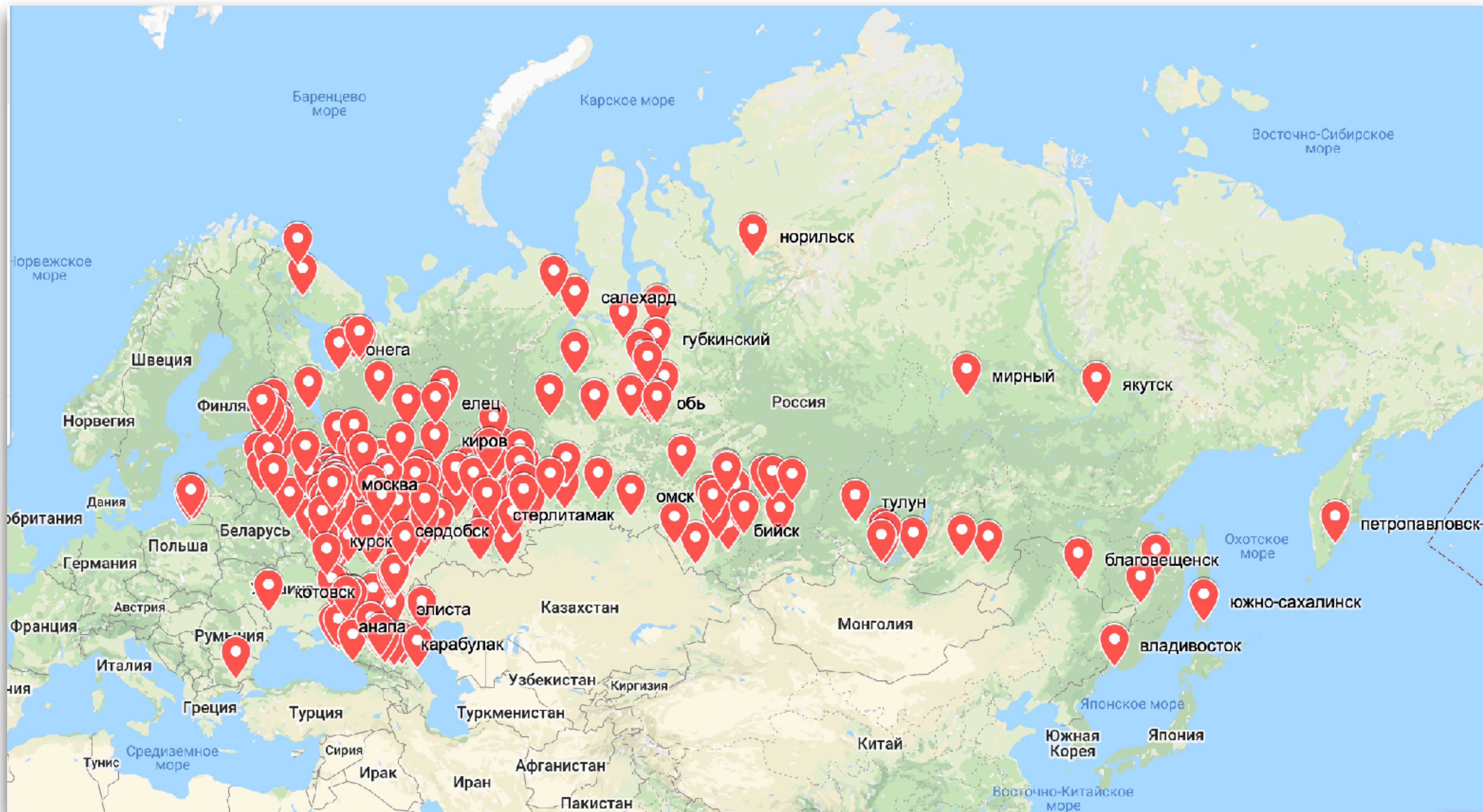
```
double = ['великий', 'нижний', 'верхний', 'старый', 'новый', 'ближний', 'дальний', 'южный',  
          'набережный', 'сосновый']
```

```
# функция для обработки того, что Deerpavlov извлек
```

```
def prepr(lst): # получает список названий из ячейки датасета  
    #print(lst)  
    str = " ".join(lst) # превращает его в строку для применения строковых методов дальше  
    #print(str)  
    str = re.sub(r' - ', "-", str) # сшиваем составные названия типа Ростов-на-Дону, re.sub .  
    str = " ".join(mystem.lemmatize(str)) # лемматизация. mystem возвращает с пустыми строками  
    str = str.strip() # mystem пишет \n в конце списка, удаляем  
    str = str.split() # снова превращаем строку в список  
    #print(str)  
    a = [] # сюда пойдут города после объединения составных названий  
    for i in range(len(str)):  
        if str[i] in double: # сверяемся со списком приставок, который задали выше  
            #print(str[i])  
            new_str = f'{str[i]} {str[i+1]}' # объединяем составные названия  
            str[i+1] = "substitute" # после объединения вторая часть остается, а если ее уда.  
            #print(new_str)  
            a.append(new_str) # пишем в список  
            #print(a)  
        else:  
            a.append(str[i]) # если название не совпало с приставкой, просто пишем его в тот  
    a = set(a) # убираем дубли  
    if "substitute" in a:  
        a.remove("substitute") # удаляем замену  
    print(a)  
    return a # получаем список уникальных объектов в нормальной форме
```

```
prepr(['Вышнего', 'Волочка', 'Ростов', '-', 'на', '-', 'Дону', 'Набережных', 'Челнах'])
```

```
{'набережная', 'вышний волочек', 'ростов-на-дону', 'челны'}  
{'вышний волочек', 'набережная', 'ростов-на-дону', 'челны'}
```

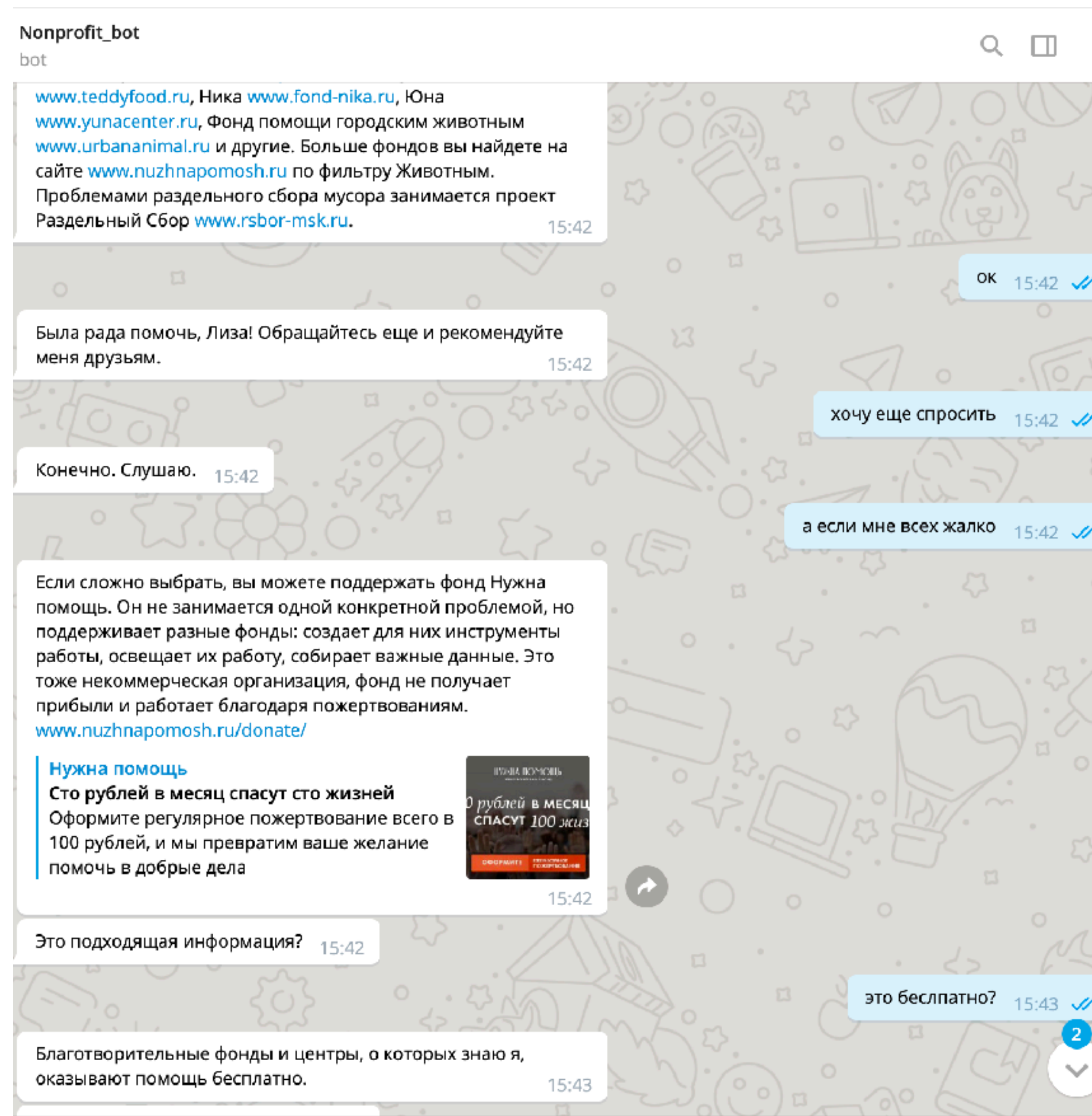



[Link to the map.](#)

Чат-бот

паралингвистический бонус

- чат-бот, который подбирает фонд для получения или оказания помощи по теме
- aimylogic



Что дальше

продолжение исследования

- увеличивать датасет
- улучшать качество моделей
- извлекать и сравнивать другие переменные
- прикладные инструменты: определение потенциала сбора по тексту, чат-бот для подбора благотворительной организации