# Краудфандинг

исследование сборов с помощью методов компьютерной лингвистики

# **Данные** собственный датасет

завершенные благотворительные сборы на <u>planeta.ru</u>

2000+ строк

900.000+ словоупотреблений

30.000+ слов-признаков (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 улучить 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: 1
   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
         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 с
kf.get_n_splits(X) # разбиваем
#print(kf)
for train_index, test_index in kf.split(X): # учим и
    #print("TRAIN:", train_index, "TEST:", test_inde
    kf_X_train, kf_X_test = X[train_index], X[test_:
    kf_y_train, kf_y_test = y[train_index], y[test_i
    kf_accuracy = train_model_2(LogisticRegression(r
    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)

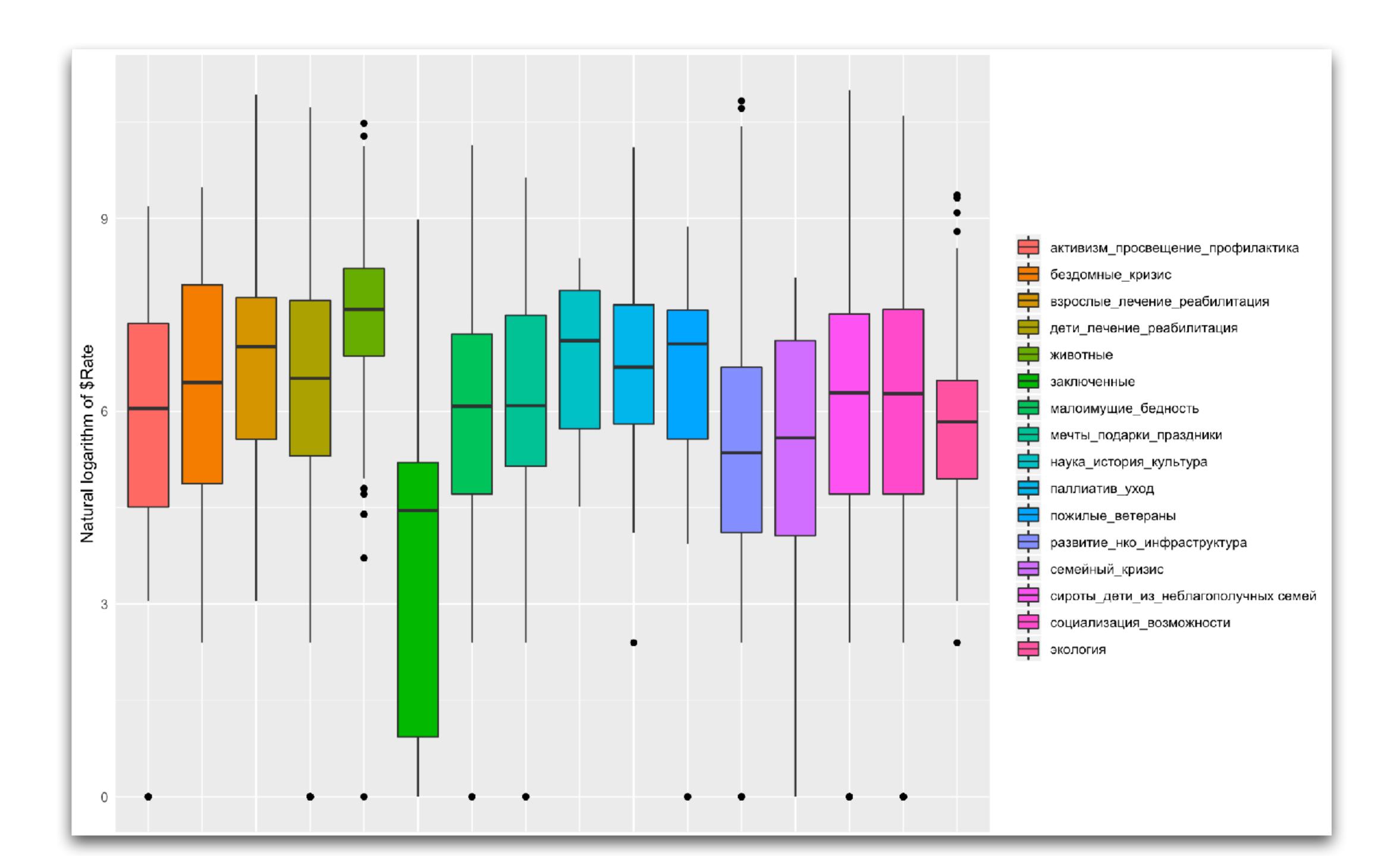
```
link
                                                    text
О Лариса Ермошина живет в городе Донской Тульско...
                                                              https://dobro.mail.ru/projects/zhizn-dlya-lari...
                                                           https://dobro.mail.ru/projects/rejdom-vsem-bud...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

#### boom.shape

(10, 2)

# 16 категорий какая лучше собирает деньги?

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

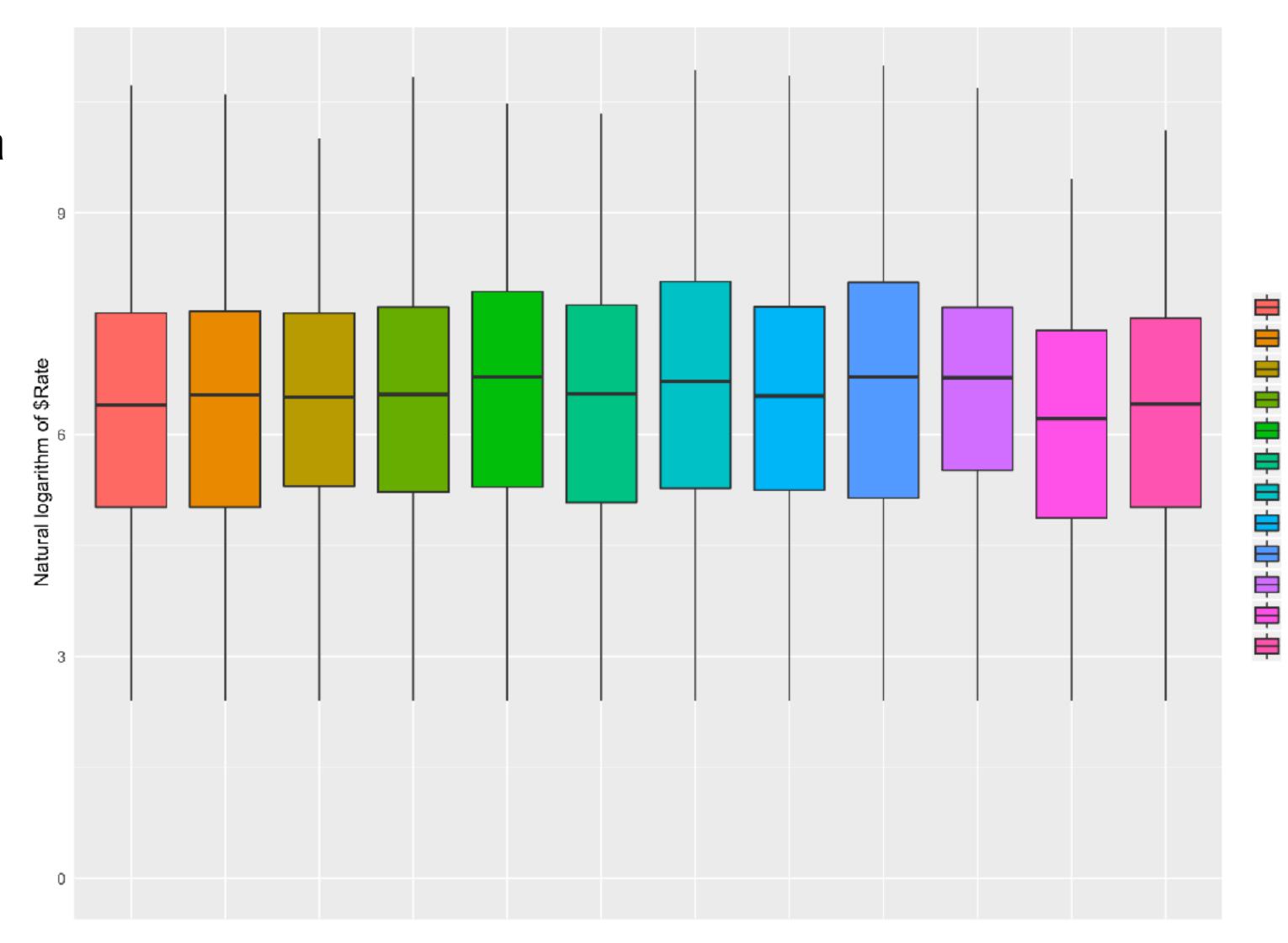


Категория-победитель	За чей счет	Степень значимо сти	anova\ tukey	kruskel\du nn without outliers	kruskel\ dunn with outliers
животные	сироты_дети_из_неблагополучных семей	***	<b>/</b>		<b>✓</b>
животные	социализация_возможности	****	<b>/</b>		<b>✓</b>
животные	развитие_нко_инфраструктура	****	<b>/</b>		<b>✓</b>
животные	дети_лечение_реабилитация	****	<b>/</b>		<b>✓</b>
животные	малоимущие_бедность	****	<b>✓</b>		<b>✓</b>
животные	мечты_подарки_праздники	***	<b>/</b>		<b>✓</b>
животные	активизм_просвещение_профилактика	***	<b>/</b>		<b>✓</b>
животные	экология	**	<b>/</b>		<b>✓</b>
животные	семейный_кризис	**	<b>✓</b>		<b>✓</b>
дети_лечение_реабилитация	развитие_нко_инфраструктура	**	<b>✓</b>		<b>✓</b>
пожилые_ветераны	развитие_нко_инфраструктура	**			
взрослые_лечение_реабилитация	развитие_нко_инфраструктура	**			
паллиатив_уход	развитие_нко_инфраструктура	**			
животные	паллатив_уход	*	<b>/</b>		
животные	взрослые_лечение_реабилитация	*	<b>/</b>		

## Сезонность

### зависит ли успех от месяца

- HeT
- 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 moderate slow	0.70 0.48 0.58	0.17 0.42 0.82	0.27 0.45 0.68	96 171 222
accuracy macro avg weighted avg	0.59 0.57	0.47 0.55	0.55 0.47 0.52	489 489 489

# **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
- ⇒ близость по расстоянию
   Левенштейна (fuzzywuzzy)
- результат хороший, хотя много случайных обстоятельств: порядок имен в тексте и т.д.: Сережа и мама Наташа

ил', 'Евгений', 'Дмитрий', 'Константина', 'Максим', 'Гарик', 'Сергей', 'Сергей', 'Винил', 'Драксея', 'Максим', 'Александр', 'Александра', 'Александр', 'Лиля', 'Лилечка', 'Ребенка', 'Лилер', 'Егорушка', 'Егора'], ['Сергей', 'Миша', 'Миша', 'Михаил', 'Михаил', 'Светлана', 'Татья 1ихаил'], ['Сергею', 'Сергей', 'Сергей', 'Сергей', 'Сергей', 'Сергей', 'Сергей', 'Сергей', 'Сергея', 'С цику', 'Вадику', 'Вадик', 'Вадима', 'Вадиму'], ['Стругацких'], [], ['Андрей', 'Аманом', 'Варк 'Варвара', 'Вари', 'Злата', 'Варваре', 'Варвара'], ['Дарья'], ['Нодежды'], ['Ванечка', 'Ваня' на', 'Марвину', 'Гринго', 'Гринго', 'Гринго', 'Гринго', 'Уринари', 'Роял', 'Граф', 'Уринари', ки', 'Лаки', 'Бумеранга'], [], ['Никиты', 'Никиты', 'Леонтьевых', 'Никита', 'Никита', 'Никита жрем', 'Медикомп', 'Октенисепт', 'Пронтосан', 'Пронтосан', 'Никите'], ['Диана', 'Диана', 'Диана', 'Марка', 'Аня', 'Анечка', 'Юра', 'Андрей', 'Сковорода', 'Ольге'], ['Деда'], ['Алеши', 'Алеша Іаргарита', 'Ириной', 'Александрой', 'Никитой', 'Оксаной'], ['Арсению', 'Арсению', 'Гармонь', ', 'Арсений', 'Арсений', 'Арсений', 'Арсений', 'Арсений', 'Арсения', 'Сыроватский', 'Арсений в', 'Клёпа'], [], ['Муравья', 'Муравья'], ['Дмитрий', 'Дмитрий', 'Дмитрий', 'Дмитрия', 'Дмитрия', 'Дмитрий', 'Дмитрий', 'Дмитрия', 'Александром', 'Игорем', 'Юрия', 'Андрея', 'Бориса', 'Егору', 'Лиле', 'Егору', 'Саше', 'Ал \', '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', 'РОМАН', 'АЛЕКСЕЙ', 'АЛЕКСЕЙ', 'АЛЕКСЕЙ', 'АЛЕКСЕЙ', 'АЛЕКСЕЙ', 'АЛЕКСЕЙ', 'АЛЕКСЕЙ', 'АЛЕКСЕЙ', 'АЛЕКСЕЙ', 'ВИКТОРА', 'ДМИТРИЙ', 'БРИГИ', 'КИРИЛИН', 'АНДРЕЯ', 'МАКСИМА', 'ЮРИЙ', 'ЮРИЙ', 'Чердакова', 'Чижа', 'ВЛАДИМИР', 'ДОРИЙ', 'ДОРИЙ', 'КОРИЗА', 'КОРШУН', 'ТУПАЛО', 'Ларье', 'Дарья', 'Мвану', 'Юры', 'Дарь', 'Дарье', 'Дарья', 'Мвану', 'Мвану', 'Дарь', 'Дарье', 'Дарья', 'Мвану', 'Мвану', 'Дарье', 'Дарья', 'Мвану', 'Дарье', 'Дарья', 'Дарья', 'Дарья', 'Дарье', 'Дарья', 'Дарья', 'Дарье', 'Дарья', 'Дарье', 'Дарье', 'Дарья', 'Дарье', 'Дарья', 'Дарье', 'Дарър', 'Дарър', 'Дарър', 'Дарър', 'Дарър', 'Дарър', 'Дарър', 'Дарър 'Дарье', 'Дарья', 'Ивану', 'Иван', 'Юры', 'Пан', 'Ивану', 'Бочча', 'Бакаидов', 'Ивана', 'Дар' 'Оксана', 'Роман', 'Вероника', 'Вероника', 'Димы', 'Ники', 'Диму', 'Димой', 'Дима', 'Максяков ['Север', 'Шани', 'Шаньку', 'Шаня', 'Шанька', 'Шани', 'Шаню', 'Шани'], ['Кольцова', 'Никиты поголовцевых', 'Элизабет', 'Марат', 'Андрей', 'Олеся', 'Мария', 'Алексей', 'Егор', 'Ирина', 'Максим', 'С', 'Сергей'], ['Лаки', 'Пильве', 'Трэвор', 'Люси', 'Серенити', 'Домра', 'Лаки', 'Поки', 'Ваксеровой', 'Поки', 'Б', 'Р', 'В', 'Власовой', 'Лиды', 'Алексеевской', 'Ириной', 'Катей', 'Панфиловой'], ['Егор', 'а', 'Фроловых', 'Егор', 'Тим'], ['Тимура', 'Софи', 'Софи', 'Софи', 'Андреасян', 'Софи', 'Рози', 'Дженнифер', 'Дарья', 'Софи', 'Софи', 'Софи', 'Софи', 'Софи', 'Софи'], [], ['Дэни', 'Галы 'Труфальдино', 'Даниэль', 'Олежка', 'Вела', 'Валерий', 'Олежке', 'Марта', 'Марточке', 'Макс за', 'Валера', 'Ливадоновых'], ['Маша', 'Маша', 'Маша', 'Маше', 'Машиной', 'Машиной'], [], [ ія', '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'], писа'], ['Рустемом'], [], ['Фроська', 'Фроська', 'Фроське'], [], [], [], [], ['Мария', 'Аркал писа'], ['Рустемом'], [], ['Фроська', 'Фроська', 'Фроське'], [], [], [], [], ['Мария', 'Аркај 4'], ['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', 'Верднига', 'Вити', 'Вити', 'Вити', 'Константин', 'Илья', 'Илья', 'Илья', 'Илья', 'Илья', 'Илья', 'Илья', 'Илья', 'Илье', 'Ильв', 'Константин', 'Константин', 'Константин', 'Константин', 'Константин', 'Константин', 'Константин', 'Константин', 'Константина'], ['Леониде'], [], ['Женя', 'Жени', 'Жени', 'Оксана', 'Кени', 'Жени', 'Кени', 'К \лёша', 'Алёши', 'Алёша', 'Алёша', 'Эльвире', 'Алёше'], [], ['Иоанна', 'Нунан', 'Ретта', 'Мац 'Владик', 'Дима', 'ПедиаШур', 'НутриДринк', 'Дима', 'Дима', 'ПедиаШура', 'НутриДринка', 'Ну 'Маши', 'Владислава', 'Педиашура', 'НутриДринка'], [], [], [], [], ['Мурклуб', 'Гавпарка', ['Геркулес', 'Санокс'], [], ['Л', 'Королевой'], [], [], [], [], ['Николай'], [], ['Нелли', 'Надежда'], ['Юмашева', 'Вики', 'Вики', 'Вика', 'Вике'], [], [], ['Л', 'Миша', 'Миша', 'Миша', 'Пермяков'], ['Петуховой'], ['Йога', 'Йога', 'Джоанн'], [], [], [], ['Илья', 'Ксюшені 'Дипрей' 'Дипрей'] ['Колесник' 'Макарова' 'Шепаков' 'Парина' 'Рыженкова' 'Ры

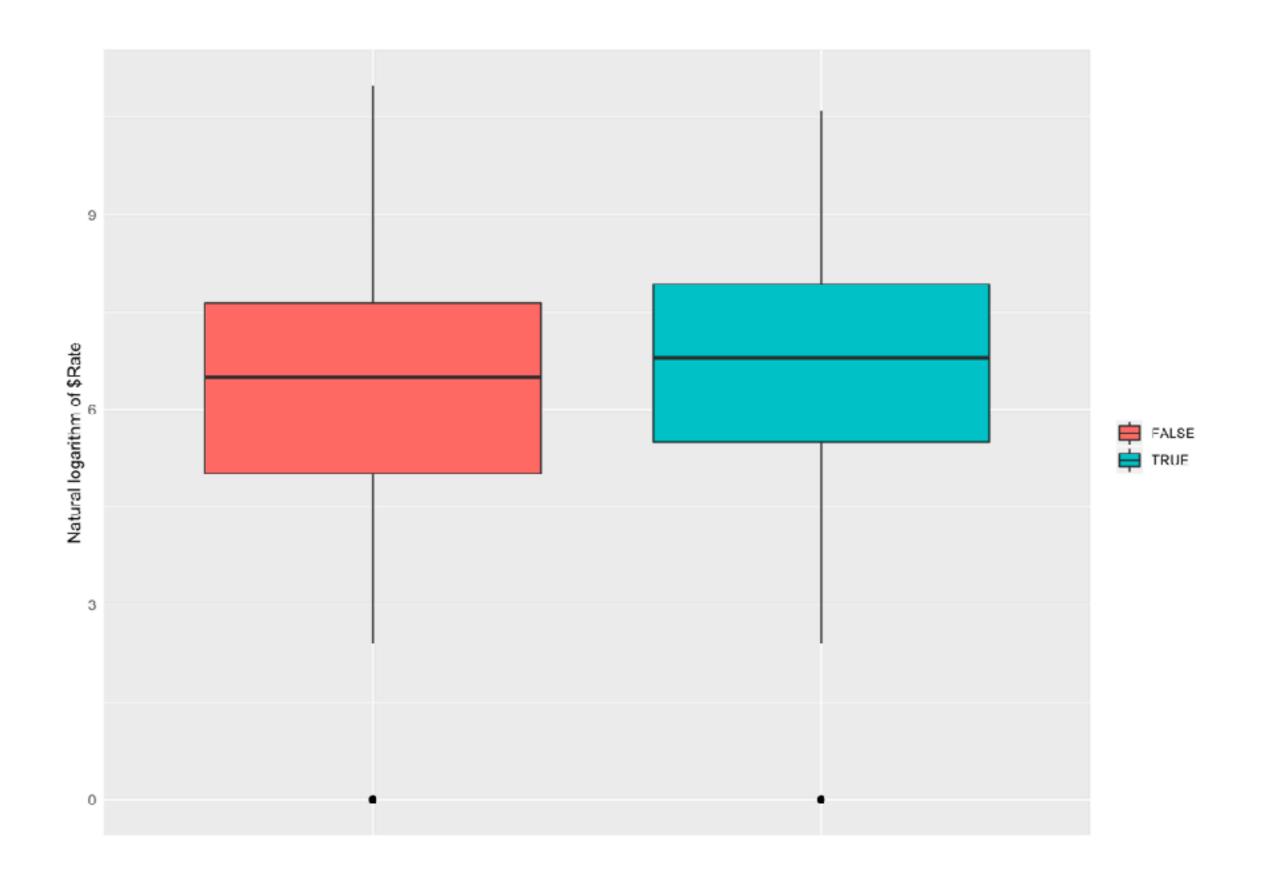
```
['Надежда', 'Островка', 'Островка']
similarity rate: 56
['Сергею', 'Сергей', 'Сергей', 'Сергей', 'Сергей', 'Сергей', 'Сергей', 'Сергея', 'Сергея', 'Сергей', 'Сергея', 'Сергея']
similarity rate: 94
['Вадима', 'Вадюшка', 'Вадику', 'Вадику', 'Вадик', 'Вадима', 'Вадиму'
similarity rate: 78
['Алиса', 'Рентген', 'Лаки', 'Лаки', 'Лаки', 'Лаки', 'Бумеранга']
similarity rate: 52
['Диана', 'Диана', 'Дианочка', 'Дианы', 'Пауль', 'Диана', 'Дианы']
similarity rate: 60
 ['Марасакин', 'Марк', 'Марка', 'Марка', 'Марка', 'Марка', 'Марка', 'Марк', 'Марк', 'Марка', 'Марк', 'Марк', 'Марку', 'Марка', 'Марку', 'Марка', 'Марку', 'Марку', 'Марка', 'Марку', 'Марка', 'Марку', 'Марку', 'Марка', 'Марку', 'Марка', 'Марку', 'Марка', 'Марку', 'Марка', 'Марку', 'Марка', 'Мар
similarity rate: 91
['Ани', 'Ани']
similarity rate: 100
['Алеши', 'Алеша', 'Алеша', 'Алеше', 'Лёше', 'Алеши']
similarity rate: 65
['Арсению', 'Арсению', 'Гармонь', 'Арсений', 'Арсений', 'Арсений', 'Арсению', 'Арсений', 'Арсений', 'Арсений', 'Арсений', 'Арсений', 'А
рсения', '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', 'Софи', 'Софи', 'Софи', 'Дженнифер', 'Дарья', 'Софи', 'Софи', 'Софи', '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

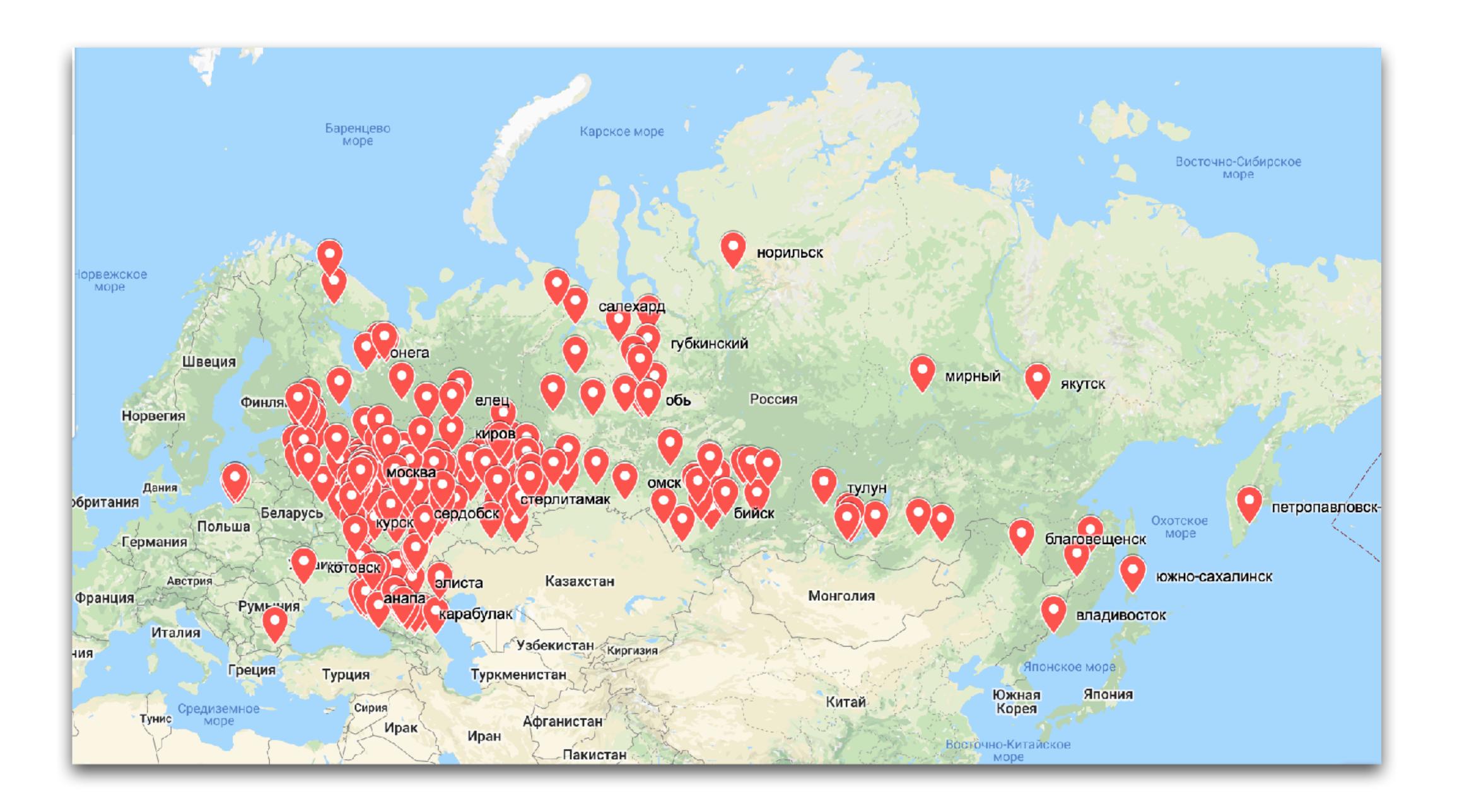
<b>.</b> у. <sup>‡</sup>	group1 ÷	group2 <sup>‡</sup>	effsize <sup>‡</sup>	n1 <sup>‡</sup>	n2 <sup>‡</sup>	magnitude <sup>‡</sup>
Rate	FALSE	TRUE	0.07250259	1592	439	small



# География по городам РФ

- извлечение топонимов с DeepPavlov
- 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 DeepPavlov извлек
def prepr(lst): # получает список названий из ячейки датасета
    #print(lst)
    str = " ".join(lst) # превращает его в строку для применения строковых методов дальше
    str = re.sub(r' - ', ''-'', str) # сшиваем составные названия типа Ростов-на-Дону, re.sub.
    str = " ".join(mystem.lemmatize(str)) # лемматизация. mystem возвращает с пустыми строка
    str = str.strip() \# mystem пишет \ n в конце списка, удаляем
   str = str.split() # снова превращаем строку в список
    #print(str)
   а = [] # 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'}
```



## Чат-бот

### паралингвистический бонус

- чат-бот, который подбирает фонд для получения или оказания помощи по теме
- aimylogic без БД
- Виталий даже предложил податься в навыки Алисы :)

Nonprofit\_bot Q [] : www.teddyfood.ru, Ника www.fond-nika.ru, Юна www.yunacenter.ru, Фонд помощи городским животным www.urbananimal.ru и другие. Больше фондов вы найдете на сайте www.nuzhnapomosh.ru по фильтру Животным. Проблемами раздельного сбора мусора занимается проект Раздельный Сбор www.rsbor-msk.ru. Была рада помочь, Лиза! Обращайтесь еще и рекомендуйте Конечно. Слушаю. Если сложно выбрать, вы можете поддержать фонд Нужна помощь. Он не занимается одной конкретной проблемой, но поддерживает разные фонды: создает для них инструменты работы, освещает их работу, собирает важные данные. Это тоже некоммерческая организация, фонд не получает прибыли и работает благодаря пожертвованиям. www.nuzhnapomosh.ru/donate/ Нужна помощь Сто рублей в месяц спасут сто жизней 0 рублей в месяц спасут 100 жиз Оформите регулярное пожертвование всего в 100 рублей, и мы превратим ваше желание помочь в добрые дела Это подходящая информация? 15:42 это беслпатно? 15:43 🗸

15:43

Благотворительные фонды и центры, о которых знаю я,

оказывают помощь бесплатно.

## 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

# Все коды будут на гитхабе

с подробными комментариями