Краудфандинг

исследование сборов с помощью методов компьютерной лингвистики

Лиза Кадетова, 2020

Данные

собственный датасет

завершенные благотворительные сборы на planeta.ru

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: ↑ миноритарного класса, создание доп. данных из имеющихся

```python
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 сп
kf.get_n_splits(X)  # разбиваем
#print(kf)
for train_index, test_index in kf.split(X): # учим м
    #print("TRAIN:", train_index, "TEST:", test_inde
    kf_X_train, kf_X_test = X[train_index], X[test_i
    kf_y_train, kf_y_test = y[train_index], y[test_i
    kf_accuracy = train_model_2(LogisticRegression(m
    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	Здравствуйте, я – Сергей Богдановский, руковож...	https://boomstarter.ru/projects/taganai89/sozd...

```
boom.shape
```

```
(10, 2)
```

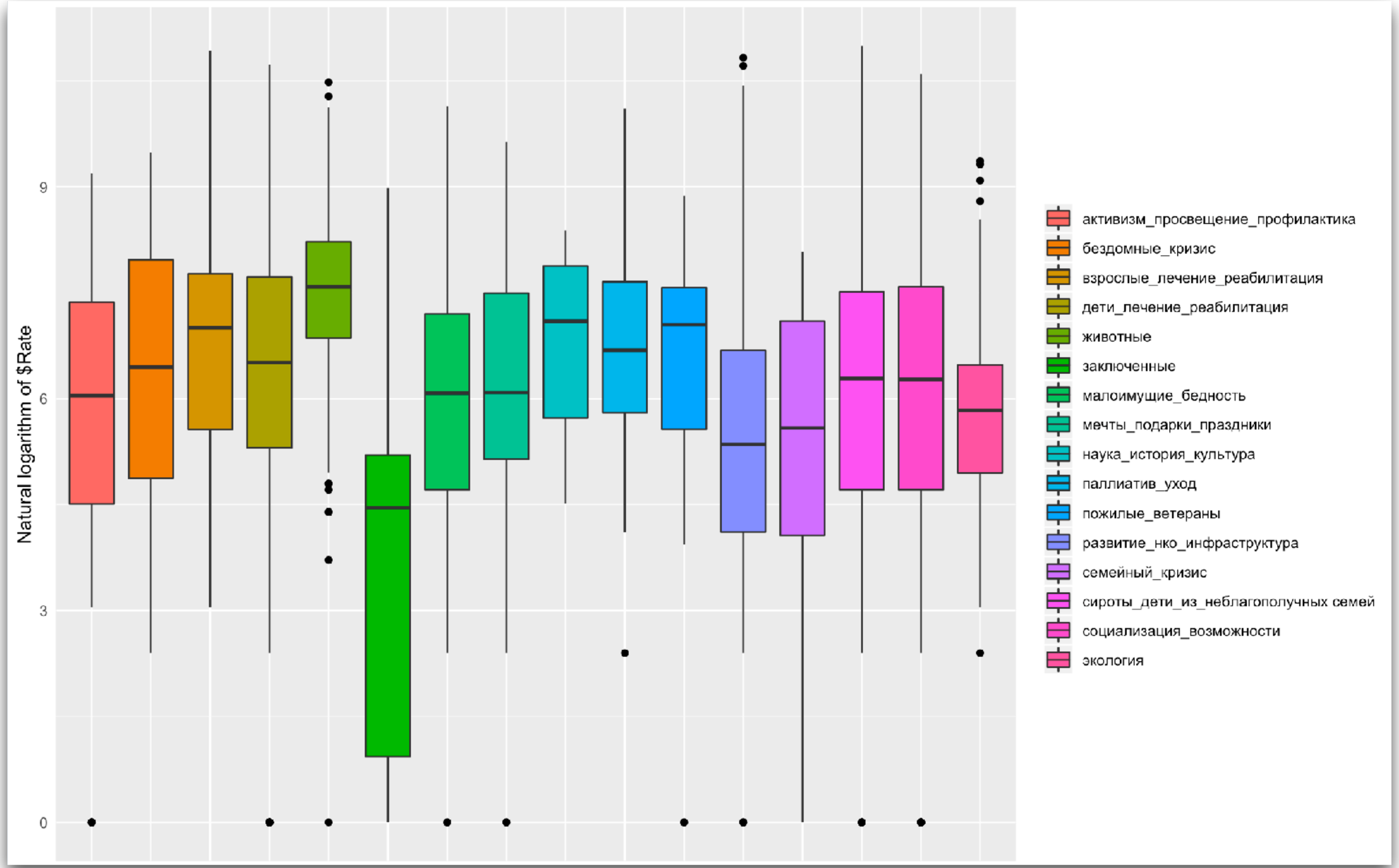
```
# просим модель определить категории для всех текстов в новом датасете

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

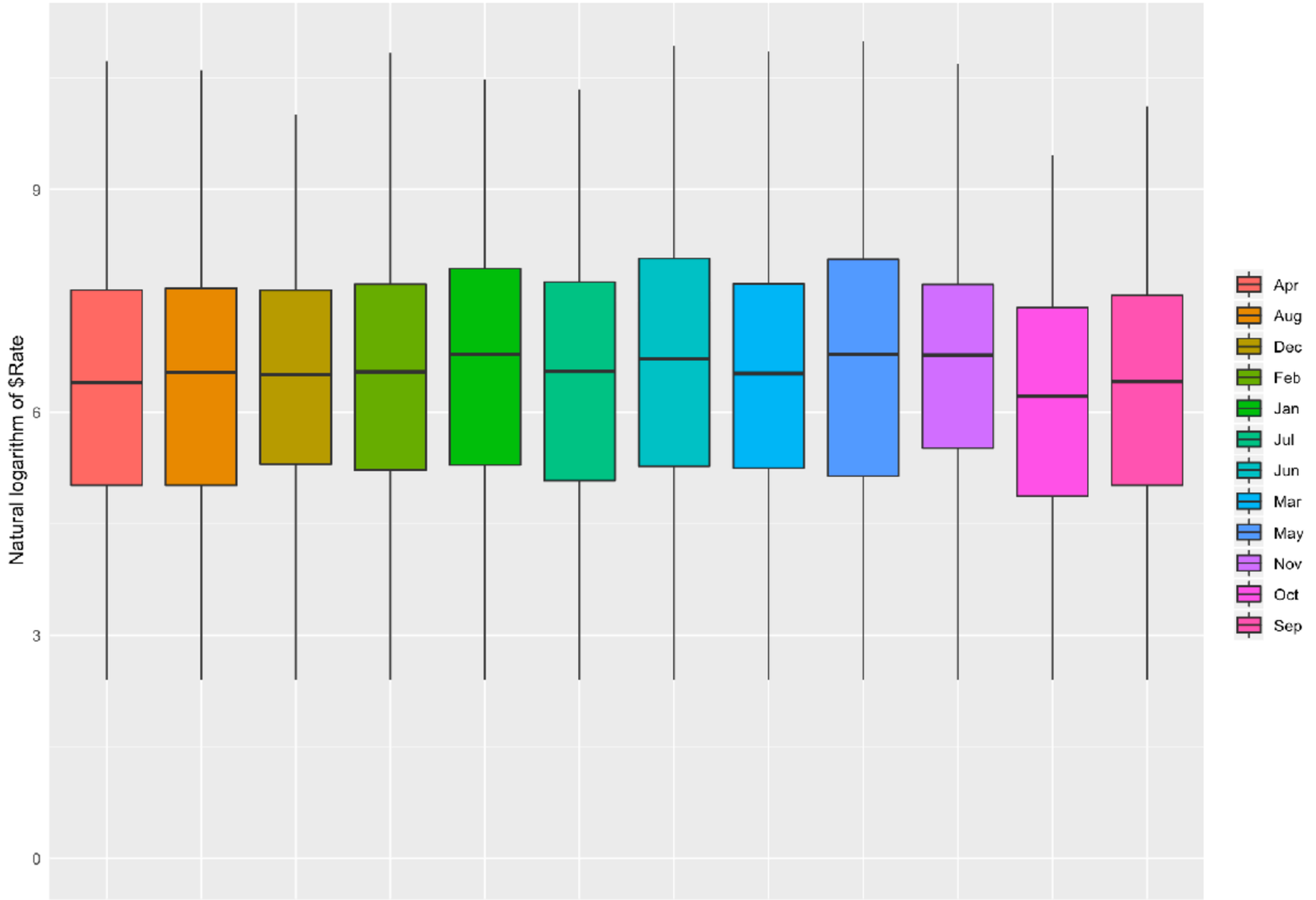

Natural logarithm of $Rate
0
3
6
9
активизм_просвещение_профилактика
бездомные_кризис
взрослые_лечение_реабилитация
дети_лечение_реабилитация
животные
заключенные
малоимущие_бедность
мечты_подарки_праздники
наука_история_культура
паллиатив_уход
пожилые_ветераны
развитие_нко_инфраструктура
семейный_кризис
сироты_дети_из_неблагополучных семей
социализация_возможности
экология

Категория-победитель	За чей счет	Степень значимости	anova\ tukey	kruskel\dunn without outliers	kruskel\ dunn with outliers
животные	сироты_дети_из_неблагополучных семей	****	☑	☑	☑
животные	социализация_возможности	****	☑	☑	☑
животные	развитие_нко_инфраструктура	****	☑	☑	☑
животные	дети_лечение_реабилитация	****	☑	☑	☑
животные	малоимущие_бедность	****	☑	☑	☑
животные	мечты_подарки_праздники	***	☑	☑	☑
животные	активизм_просвещение_профилактика	***	☑	☑	☑
животные	экология	**	☑	☑	☑
животные	семейный_кризис	**	☑	☑	☑
дети_лечение_реабилитация	развитие_нко_инфраструктура	**	☑	☑	☑
пожилые_ветераны	развитие_нко_инфраструктура	**	☐	☑	☐
взрослые_лечение_реабилитация	развитие_нко_инфраструктура	**	☐	☑	☐
паллиатив_уход	развитие_нко_инфраструктура	**	☐	☑	☐
животные	паллатив_уход	*	☑	☐	☐
животные	взрослые_лечение_реабилитация	*	☑	☐	☐

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

['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', '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', '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

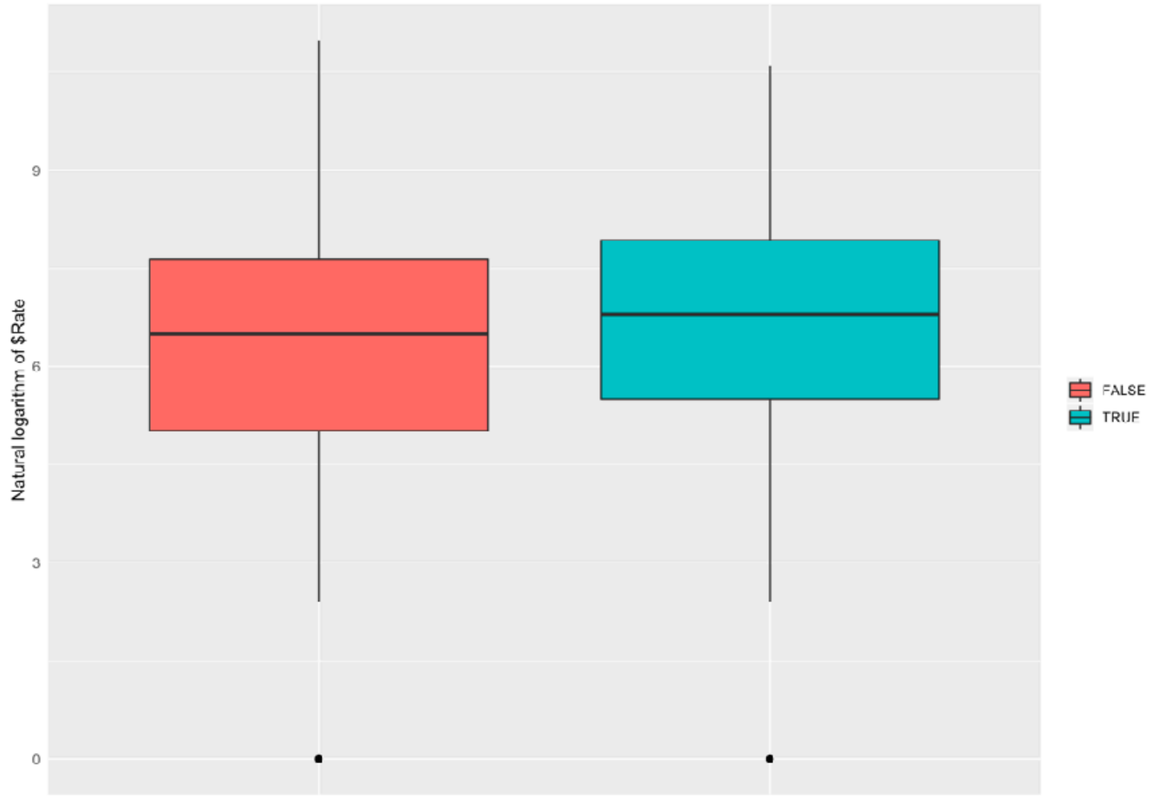
['Кольцова', 'Никиты', 'Никиты', 'Никита']
similarity rate: 61

['Тимура', 'Софи', 'Софи', 'Софи', 'Андреасян', 'Софи', 'Роза', 'Софи', 'Софи', 'Софи', 'Софи', 'Розы', 'Софи', 'Софи', 'Софи', 'Дженнифер', 'Дарья', '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

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



География
по городам РФ

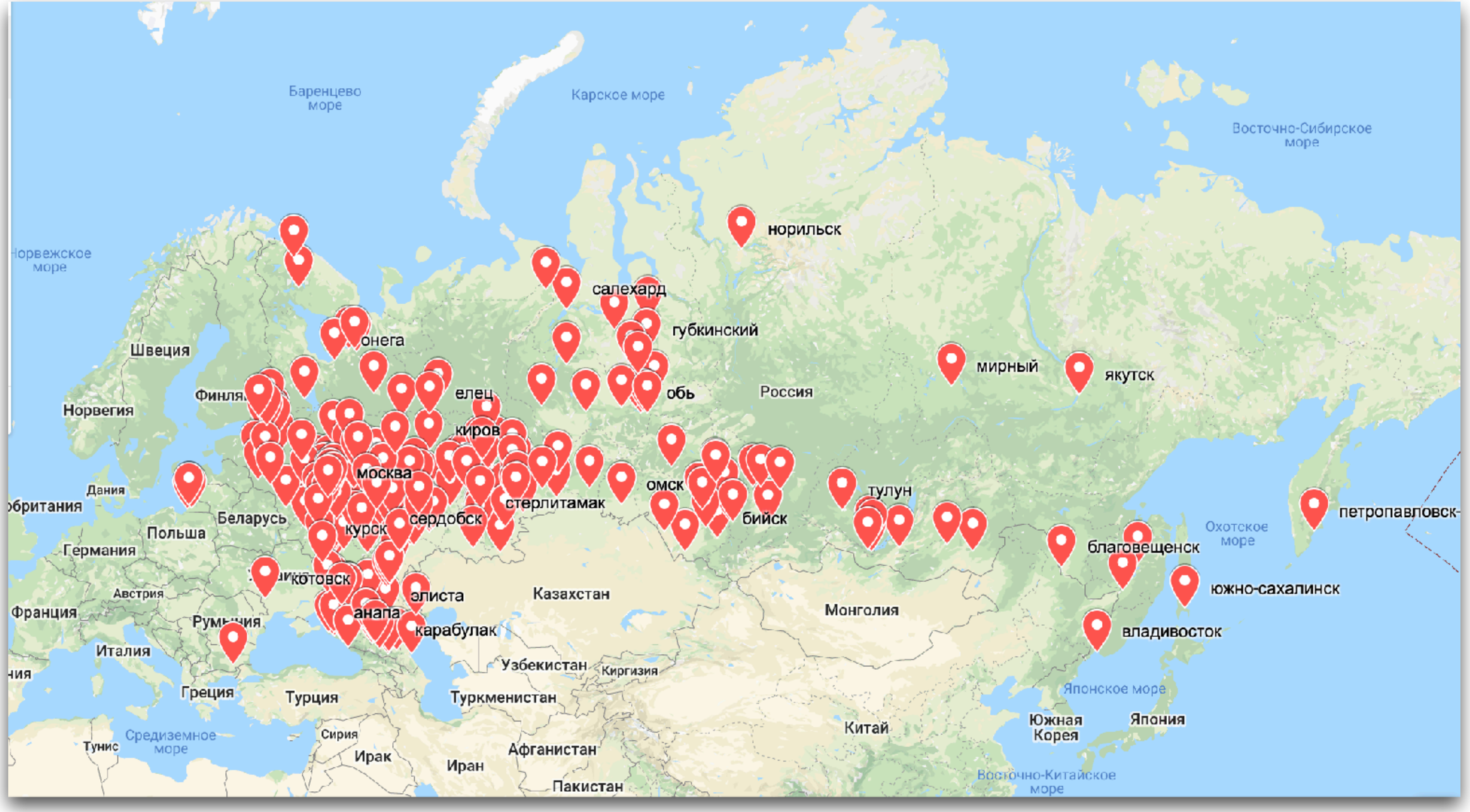
- извлечение топонимов с DeepPavlov
- проблемы: мусор, ошибки токенизации (Ростов-на-Дону, Великий Устюг и т.п.)
- чистка списка
- сопоставление со списком городов РФ

```
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
    #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 строка
    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'])
```

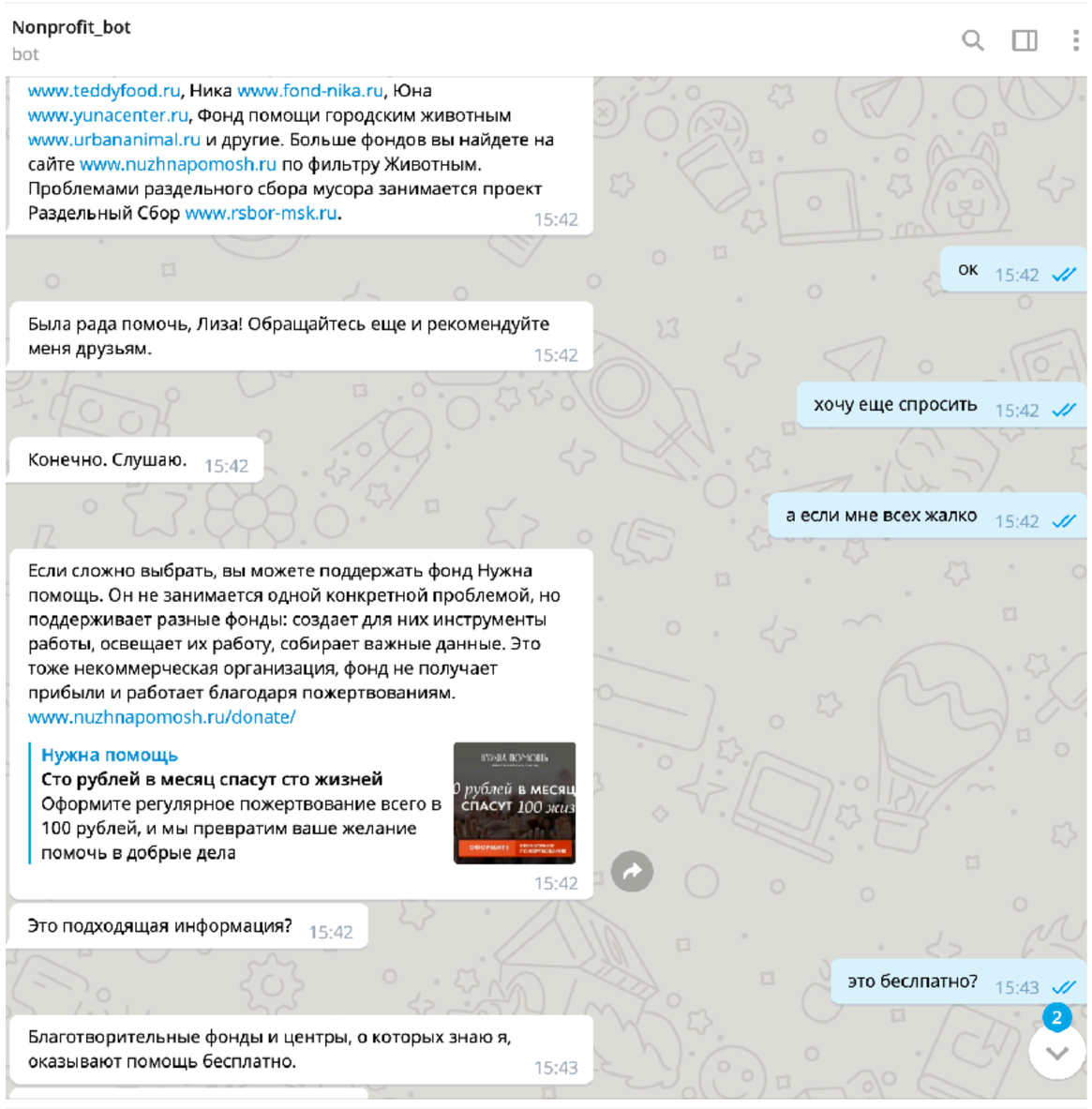
```
{'набережная', 'вышний волочек', 'ростов-на-дону', 'челны'}
{'вышний волочек', 'набережная', 'ростов-на-дону', 'челны'}
```


Баренцево море
Карское море
Восточно-Сибирское море
Норвежское море
норильск
салехард
губкинский
онега
мирный
якутск
Швеция
Финляндия
Норвегия
елец
обь
Россия
киров
москва
омск
тулун
Дания
Великобритания
стерлитамак
Беларусь
сердобск
бийск
петропавловск-
Польша
курск
Охотское море
Германия
благовещенск
котовск
элиста
Казахстан
южно-сахалинск
Австрия
Франция
анапа
Монголия
карабулак
владивосток
Румыния
Италия
Узбекистан
Киргизия
Греция
Турция
Туркменистан
Японское море
Китай
Южная Корея
Япония
Средиземное море
Тунис
Сирия
Ирак
Иран
Афганистан
Пакистан
Восточно-Китайское море

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 :)



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

Скоро :)