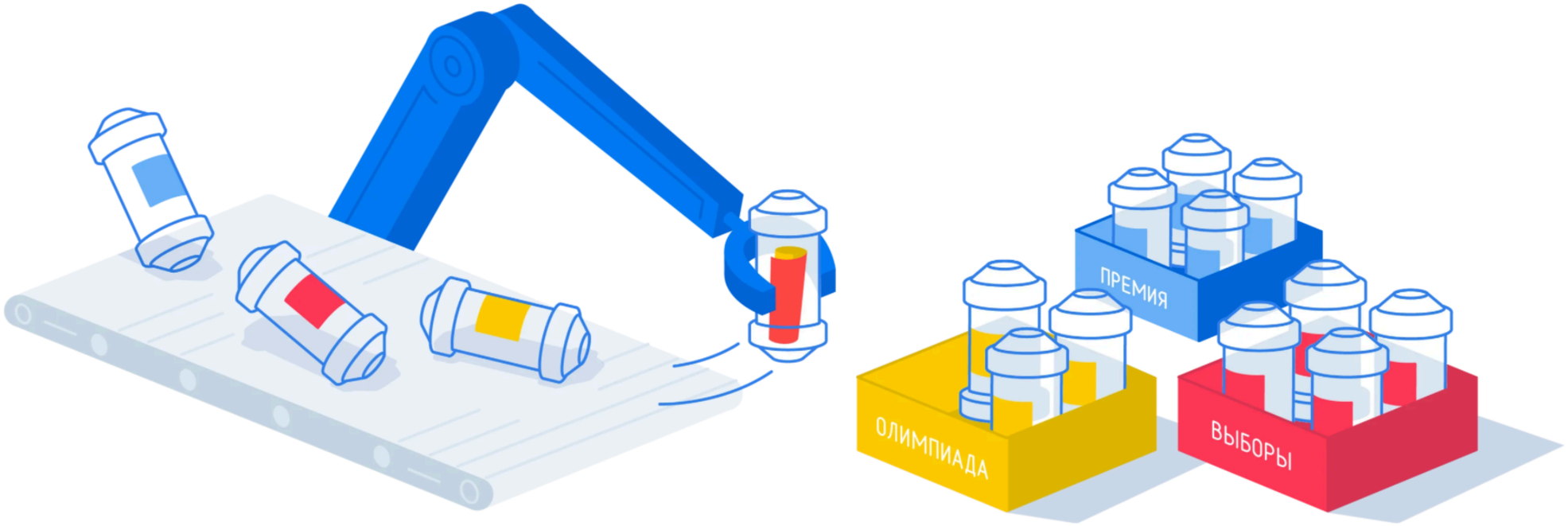
тему пожалуйста

Рекомендательная система актуальных тем для исследований и отчетов

Выпускной проект DS-School
Выполнил Мукасеев Евгений

Цель проекта

Создать робота-советника, который будет рекомендовать темы исследований



7519

Партнеров у Яндекс.Новостей
и нет редакции

более

100 000

Сообщений в будний день

Какие новости нужны?

Не все новости одинаково полезны

- Свежие - регулярное обновление
- Тематические - отбор источников
- Популярные - контроль просмотров

Подготовка данных

Data - новая нефть

- Регулярный парсинг сайтов:

banki.ru

alfabank.ru

vtb.ru

- Очистка данных
- Обновление просмотров

parser.ipynb + views.ipynb

Используемые библиотеки и кратко про код

- datetime - обработка времени
- requests - соединение с сайтом
- re - обработка регулярных выражений
- HTTPAdapter, Retry - обработка сброса соединений
- sleep - задержка выполнения
- BeautifulSoup - обработка html страниц

Цикл парсинга

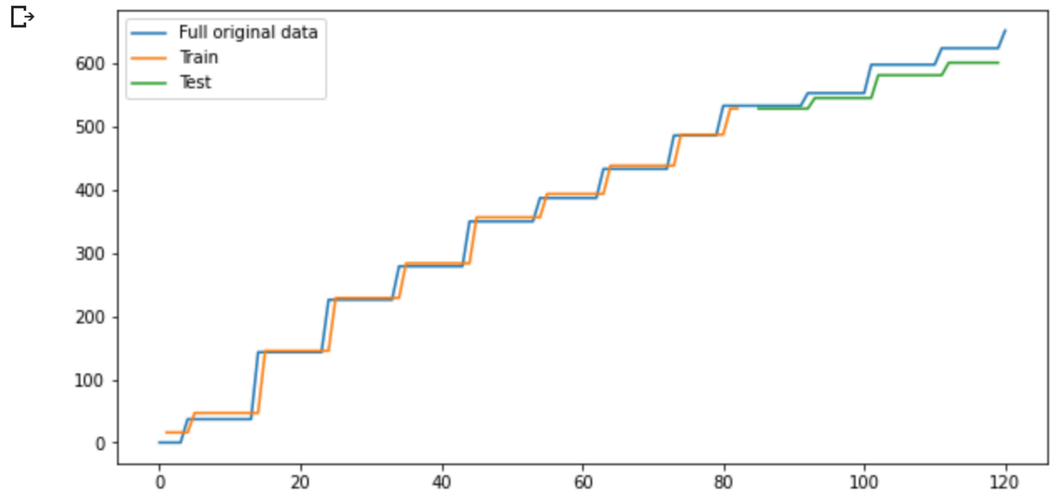
```
if sites['banki']['on']:
  for page in range(sites['banki']['first_page'], sites['banki']['last_
    count_add_articles = 0
    count_upd_views = 0
    count_upd_comments = 0
    count_u  Follow link (cmd + click)
    url = 'https://www.banki.ru/news/lenta/page' + str(page)
    response = load_page(url)
    if response == -1:
      continue
    soup = BeautifulSoup(response.text, 'lxml')
    articles = soup.find_all('ul', class_='text-list text-list--date te
    for article in articles:
      # Кол-во просмотров
      views = 0
      # Кол-во комментариев
      comments = 0
      # Источник новости
      source = ''
      article_url = article.find('a', class_='text-list-link')
      if article_url.get('href')[0] == 'h':
          # print('Error url - {}'.format(article_url.get('href')))
          continue
      article_id = str(article_url.get('href').split('=')[1])
      article_info = article.find_all('span', class_='news__info')
```

lstm.ipynb

Используемые библиотеки и кратко про код

- Sequential, Dense, LSTM - работа с нейронной сетью
- MinMaxScaler - нормализация
- mean_squared_error - MSE

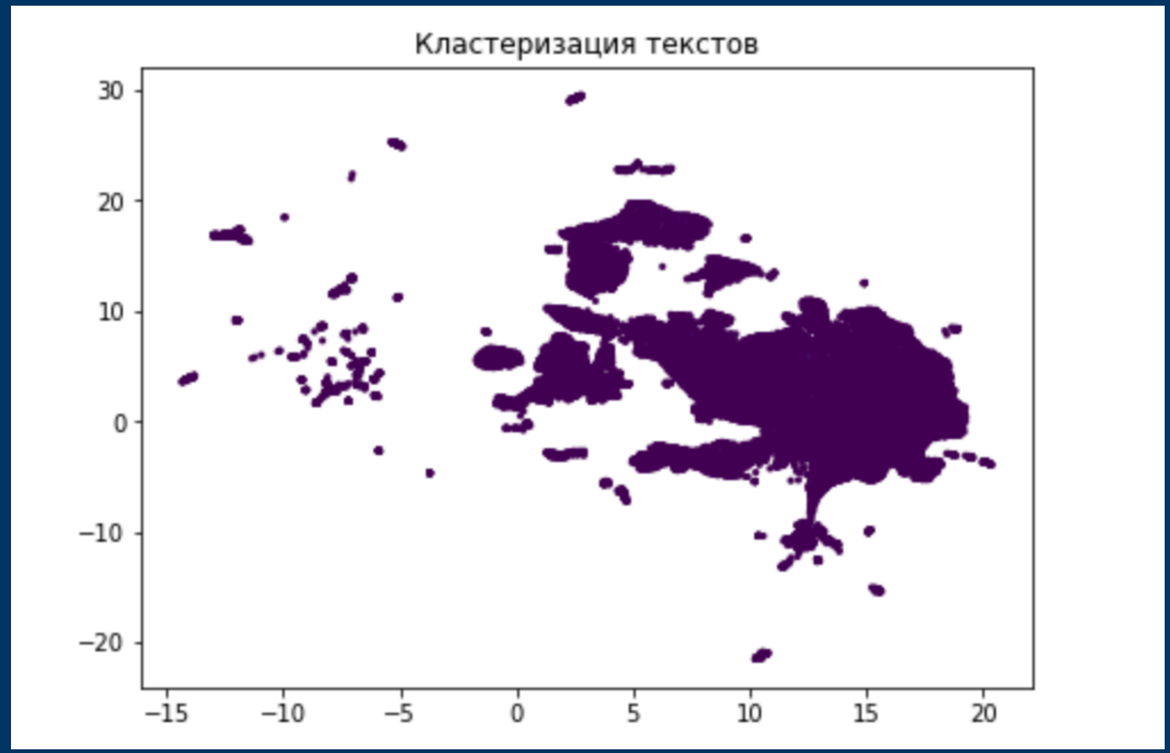
```
# plot baseline and predictions
plt.figure(figsize = (10,5))
plt.plot(scaler.inverse_transform(dataset), label = 'Full original data')
plt.plot(trainPredictPlot, label = 'Train')
plt.plot(testPredictPlot, label = 'Test')
plt.legend()
plt.show()
```



word_analysis.ipynb

Используемые библиотеки и кратко про код

- nltk - библиотека для обработки естественного языка
- matplotlib.pyplot, seaborn - построение графиков
- string - обработка строк
- umap, TruncatedSVD - снижение размерности
- TfidfVectorizer - TF-IDF для необработанных документов



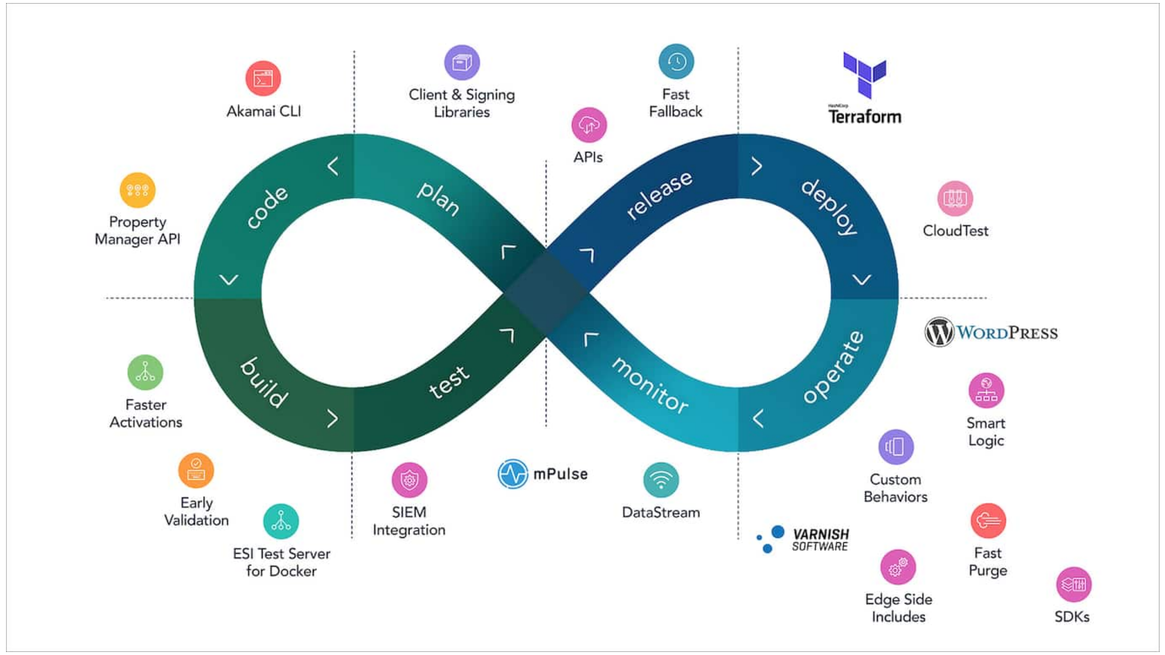
Преодолевая трудности

Трудное не есть невозможное

- Proxy
- Pagination
- Разное время публикации новостей
- Обработка текста занимает много времени и ресурсов
- Для прогноза популярности используются только данные о просмотрах

Предварительные итоги

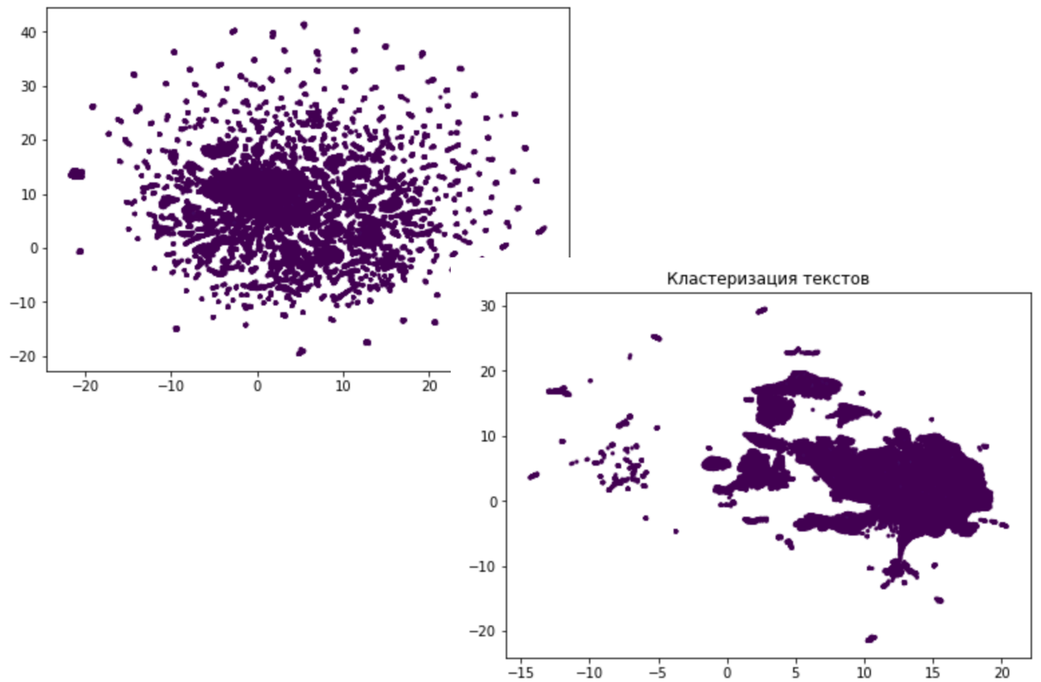
- Датасет новостей - более 120 тыс.
- Обновление просмотров - 2 раза в минуту
- Анализ текста - отобраны 100 ключевых слов
- Модель прогнозирования популярности - LSTM для регрессии с оконным методом (RMSE train score: 20.64, test score: 12.70, при значениях от 0 до 652)
- Модель кластеризации новостей



Выводы

- Заголовки в основном однородны, что нельзя сказать про тексты
- MVP можно сделать своими руками
- Для решения большинства задач достаточно публичных библиотек
- LSTM для регрессии с оконным методом позволяет предсказывать количество просмотров (RMSE train score: 20.64, test score: 12.70)
- Применение модели возможно новостными агрегаторами

```
plt.figure(figsize=(7,5))
plt.scatter(embedding[:, 0], embedding[:, 1],
c = df.views,
s = 10, # size
edgecolor='none'
)
plt.show()
```



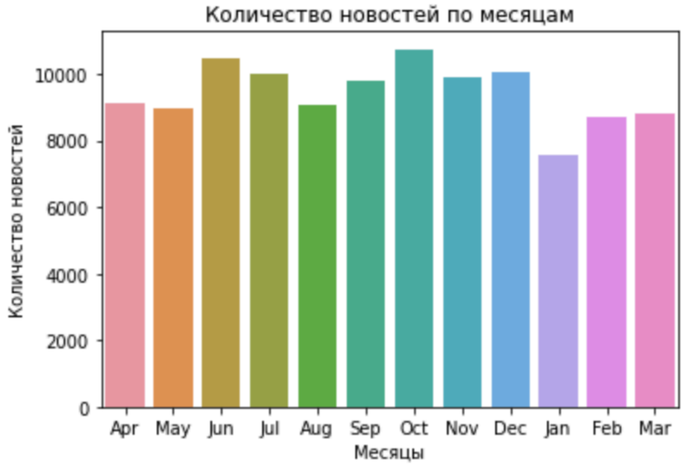
Что дальше?

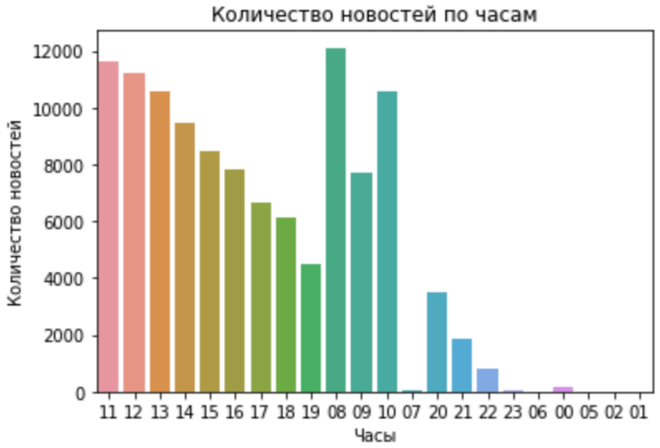
Вкалывают роботы, а не человек

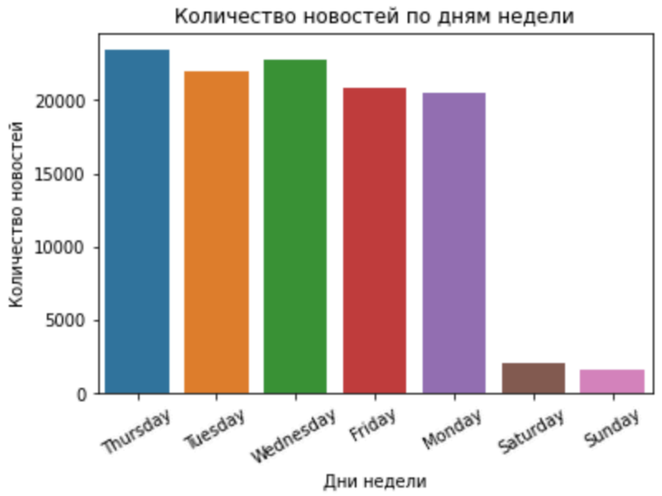
- Добавление новых источников (Открытие, Тиньков, Газпромбанк)
- Анализ вхождения отобранных слов
- Анализ выявленных кластеров
- Бот в Telegram с подпиской на новости
- Добавление продукта в SmartMarket от СБЕРа

Приложения


Количество новостей по годам
Количество новостей
0
5000
10000
15000
20000
2015
2016
2017
2018
2019
2020
2021
Годы


Количество новостей по месяцам
Количество новостей
0
2000
4000
6000
8000
10000
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Месяцы


Количество новостей по часам
Количество новостей
0
2000
4000
6000
8000
10000
12000
11 12 13 14 15 16 17 18 19 08 09 10 07 20 21 22 23 06 00 05 02 01
Часы


Количество новостей по дням недели
Количество новостей
0
5000
10000
15000
20000
Thursday
Tuesday
Wednesday
Friday
Monday
Saturday
Sunday
Дни недели

Спасибо за внимание

GitHub

https://github.com/mukaseevru/