

Исследовательский хакатон Яндекс Практикума

- [Описание задачи](#)
- [Сбор данных](#)
 - [Оценка результатов ручного поиска](#)
 - [Подключение библиотеки](#)
 - [1.2. Поиск и сбор целевых профилей](#)
 - [1.3. Парсинг постов и профилей](#)
- [Получение и объединение 5 датасетов с команды № 2, 3, 4, 8 и 10](#)
 - [Датасет нашей команды №2](#)
 - [Датасет команды №3](#)
 - [Датасет команды №4](#)
 - [Датасет команды №8](#)
 - [Часть 1](#)
 - [Часть 2](#)
 - [Датасет команды №10](#)
 - [Объединение датасетов](#)
- [Обработка данных](#)
 - [Предобработка](#)
 - [Подготовка текста](#)
 - [EDA](#)
 - [Выборка постов](#)
- [Моделирование](#)
 - [Векторизация текстов](#)
 - [3.2. LDA](#)
 - [Ключевые слова](#)
 - [Интерпретация тем для LDA](#)
 - [Типичные статьи](#)
 - [3.3. NMF](#)
 - [Ключевые слова](#)
 - [Интерпретация тем для NMF](#)
 - [Типичные статьи](#)
 - [ТОП-10 тем постов целевой аудитории](#)
 - [ТОП-10 тем, вызывающих наибольшую реакцию](#)
- [Выводы](#)

Описание задачи

По условиям Практикума исследование проводится командой из 5 человек. Всего в хакатоне принимают участие 10 команд.

Предлагаем ознакомиться с исследованием команды №2.

Состав участников:

- Менеджмент:
 - Давыдова Евгения
- Специалисты Data Science:
 - Папин Алексей
 - Балычева Ирина
 - Григорьев Александр
- IT рекрутер:
 - Карепанова Антонина

Бизнес-требования

1. Отрасль и направления деятельности: *EdTech*, сервис онлайн образования.
2. Общее описание задачи: провести исследование по теме наставничества и менторства на основании контента социальной сети LinkedIn, размещенного в открытом доступе, созданного целевой аудиторией.
3. Цели исследования:
 - Определить топ-10 тем в направлении наставничества на основании наибольшего охвата, используя теги *наставничество, менторство, коучинг, mentorship, mentor, coaching, buddy*.
 - Определить топ-10 популярных тем по просмотрам, реакциям: лайкам, комментариям, репостам среди IT-специалистов, подходящих под описание целевой аудитории исследования,
 - Дополнить профили целевой аудитории новыми параметрами.

В наше распоряжение предоставлен портрет целевой аудитории, в котором описаны роли наставника и ревьюера.

В данной тетрадке опишем процесс исследования, касающийся работы специалистов *Data Science*.

Обязательные требования для работы DS.

- Собрать датасет в виде CSV- или JSON-файла (не ссылки),
- Презентация в виде ссылки на *Google Slides*,
- Ссылка на код проекта размещенного на *GitHub* и оформленного по рекомендациям.

Общая задача для команды: провести исследование по теме наставничества, сформировать результат в виде презентации и выступить на демо.

Порядок исследования:

1. Соберём данные. С помощью действующих аккаунтов социальной сети *LinkedIn* выполним веб-скрейпинг и соберём данные аккаунтов людей и их постов, подходящих под целевую аудиторию.
2. Выполним обработку полученных данных и сформируем датасет для исследования. Подготовим текстовые данные постов для исследования. Выполним очистку текстов от ненужных символов и слов.

3. Сделаем токенизацию, векторизацию. Проведем исследование для достижения целей бизнеса. Исследуем датасет применив к текстам постов метод латентного размещения Дирихле (*LDA*) для выделения тематики постов. Выявим ТОП-10 тем постов целевой аудитории. Узнаем ТОП-10 тем, вызывающих наибольшую реакцию у аудитории соцсети.
4. Сделаем выводы по итогам исследования и оценим результаты.

Сбор данных

Получать данные из соцсети будем непосредственно со страниц сайта *www.linkedin.com*. Для этого воспользуемся двумя библиотеками:

- *BeautifulSoup* — это пакет *Python* для анализа документов HTML и XML,
- *Selenium WebDriver* — это инструмент для автоматизации действий веб-браузера.

Как будем выполнять сбор данных:

1. Сначала в ручном режиме постараемся найти профили пользователей соцсети подходящие под целевую аудиторию. Оценим какие поисковые запросы выдают наиболее релевантный результат.
2. Напишем код, который с помощью поисковых запросов соберёт максимально возможное число целевых профилей. Сохраним полученные профили в файл `profiles.csv`.
3. Далее итерируясь по найденным профилям будем парсить данные из профилей пользователей и их посты. Данные из профилей добавим в `profiles.csv`, а посты сохраним в `posts.csv`. Общим полем в обеих таблицах будет `user_id` - идентификатор пользователя в соцсети *LinkedIn*.

Оценка результатов ручного поиска

Попробовав выполнить ручной поиск, используя теги `наставничество`, `менторство`, `коучинг`, `mentorship`, `mentor`, `coaching`, `buddy`, стало понятно, что по данным запросам целевая аудитория очень низкая. Чаще попадают рекламные аккаунты либо аккаунты без контента.

EdTech прежде всего предполагает онлайн обучение IT специалистов. Поэтому было решено искать аккаунты IT специалистов. Именно данные специалисты скорее всего будут нашей целевой аудиторией. Конечно же не все, но часть точно.

Примеры запросов: `разработка ПО`, `devops`, `data science`, `project management`, `design ui ux` и т.д. Т.е. все те специалисты, которые могу и обучаются онлайн или делятся опытом.

Выполним поиск таких аккаунтов. А позже, выполним фильтрацию в соответствии с ключевыми словами.

Первым делом загрузим все необходимые для работы библиотеки.

Подключение библиотеки

```
In [2]: import time
import configparser
import random
```

```

import re
import os.path

import pandas as pd
import numpy as np
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
import pymorphy2
import nltk
from nltk.corpus import stopwords
from sklearn.decomposition import LatentDirichletAllocation, NMF
from sklearn.feature_extraction.text import TfidfVectorizer
import matplotlib.pyplot as plt
import seaborn as sns
from itertools import product

sns.set_theme(style='whitegrid', palette='Set2')
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
pd.set_option('display.max_colwidth', None)

SEED = 42

```

Загружаем конфиг

```

In [3]: # папка, куда будем сохранять данные
DATA_PATH = '../datasets/'

# путь к файлу расширения для Chrome "Доступ к LinkedIn"
EXTENSION_PATH = '1.5_0.crx'

# файл конфигурации
CFG_FILE = 'parser.ini'

"""
файл конфигурации необходимо предварительно создать,
формат файла parser.ini:
[LINKEDIN]
USER_LOGIN = эл_почта_без_кавычек
USER_PASSWORD = пароль_без_кавычек
""";

# загружаем данные из конфига
conf = configparser.ConfigParser()
try:
    conf.read(CFG_FILE)
    USER_LOGIN = conf['LINKEDIN']['USER_LOGIN']
    USER_PASSWORD = conf['LINKEDIN']['USER_PASSWORD']
except:
    print(f'Не удалось прочитать файл конфигурации: {CFG_FILE}')

```

Общие процедуры и функции

```

In [4]: # функция создания и открытия окна браузера
def chrome_start():
    # настройки браузера
    options = webdriver.ChromeOptions()

    # подключаем расширение к драйверу
    options.add_extension(EXTENSION_PATH)

    # меняем стратегию - ждать, пока свойство
    # document.readyState примет значение interactive
    options.page_load_strategy = 'eager'

```

```

# запускаем Chrome с расширением
driver = webdriver.Chrome(options=options)

return driver

```

```

In [5]: # процедура входа в свою учетную запись в LinkedIn
def linkedin_login(driver):
    try:
        # открываем страницу входа LinkedIn,
        # необходимо отключить двухфакторную аутентификацию
        driver.get("https://linkedin.com/uas/login")

        # ожидаем загрузку страницы
        time.sleep(3)

        # поле ввода имени пользователя
        username = driver.find_element(By.ID, "username")
        # вводим свой Email
        username.send_keys(USER_LOGIN)

        # поле ввода пароля
        pword = driver.find_element(By.ID, "password")
        # вводим пароль
        pword.send_keys(USER_PASSWORD)

        # нажимаем кнопку Войти
        driver.find_element(By.XPATH, "//button[@type='submit']").click()
    except:
        print('Не удалось открыть и войти в linkedin.com')

```

```

In [6]: # формируем запрос на поиск людей, по ключевым словам
def search_people_url(keywords, tags, page_num=1):
    """
    Функция на вход получает ключевые слова,
    список тем публикаций для поиска и номер страницы.
    Возвращает url для запроса страницы.
    """

    # преобразуем теги из списка в формат для запроса
    tags_str = str(tags).replace(" ", "").replace("'", '')

    # формируем строку запроса
    search_url = 'https://www.linkedin.com/search/results/people/'
    search_url += f'?keywords={keywords}'
    search_url += '&origin=FACETED_SEARCH'
    search_url += f'&page={page_num}'
    search_url += '&profileLanguage=["ru"]'
    # темы публикаций (хештеги)
    search_url += f'&talksAbout={tags_str}'

    return search_url

```

```

In [7]: # получаем список профилей на странице
def get_profiles(driver):
    """
    Функция получает драйвер открытой страницы,
    ищет ссылки на доступные профили пользователей и возвращает
    список id пользователей.
    """

    # список найденных профилей
    profiles = []

    # ищем на странице ссылки на профили
    finded_profiles = driver.find_elements(
        By.CSS_SELECTOR, "span.entity-result__title-text a.app-aware-link"
    )
    for profile in finded_profiles:
        # получаем url на профиль пользователя

```

```

url = profile.get_attribute("href")
# если url ссылается на доступный профиль
if 'linkedin.com/in' in url:
    # оставляем только id профиля
    profile_id = url.split('?')[0].split('/in/')[1]
    # добавляем id в список
    profiles.append(profile_id)

# избегаемся от дублей, если вдруг появятся
profiles = list(set(profiles))
return profiles

```

```

In [8]: # прокрутка страницы, для загрузки динамического контента
def get_scrolled_page(driver, num_scrolls=15, pause_time=0.5):
    """
    Функция прокручивает страницу, загруженную в экземпляр driver,
    num_scrolls раз, с pause_time паузами между прокрутками.
    Возвращает код страницы.
    """
    # текущая высота body
    last_height = driver.execute_script('return document.body.scrollHeight')
    for i in range(num_scrolls):

        # нажимаем кнопку PageDown 5 раз
        for _ in range(5):
            driver.find_element(By.TAG_NAME, 'body').send_keys(Keys.PAGE_DOWN)
            # делаем паузу для загрузки динамического контента
            time.sleep(random.uniform(pause_time, 3))

        # вычисляем новую высоту body
        new_height = driver.execute_script('return document.body.scrollHeight')
        if new_height == last_height:
            break
        last_height = new_height

    return driver

```

```

In [9]: # собираем информацию о пользователе
def get_user_info(driver, user_id):
    """
    Функция парсит со страницы профиля информацию о пользователе.
    На вход получает, драйвер и идентификатор пользователя.
    На выходе возвращает список с данным профиля
    """
    # прокручиваем страницу до конца что бы загрузился динамический контент
    driver = get_scrolled_page(driver, num_scrolls=3, pause_time=0.5)

    # извлекаем код страницы
    src = driver.page_source

    # передаём код страницы в парсер
    soup = BeautifulSoup(src, 'lxml')

    # извлекаем HTML содержащий имя и заголовок
    intro = soup.find('div', {'class': 'mt2 relative'})

    # получаем имя
    user_name = ''
    try:
        name_loc = intro.find("h1")
        user_name = name_loc.get_text().strip()
    except: ...

    # заголовок, обычно тут пишут, где работает или специальность или навыки
    user_head = ''
    try:
        head_at_loc = intro.find("div", {'class': 'text-body-medium'})
        user_head = head_at_loc.get_text().strip()

```

```

except: ...

# получаем теги
user_tags = ''
try:
    # темы публикаций
    tags_at_loc = intro.find(
        "div", {'class': 'text-body-small t-black--light break-words mt2'}
    )
    # уточняем
    tags_at_loc = tags_at_loc.find('span', {'aria-hidden': 'true'})
    # убираем лишние символы
    user_tags = tags_at_loc.get_text().split(':')[1].strip()
    user_tags = user_tags.replace('#', '').replace(' и', ',')
except: ...

# получаем локацию пользователя
user_location = ''
try:
    location_at_loc = intro.find(
        "div", {'class': 'pv-text-details__left-panel mt2'}
    )
    # уточняем
    location_at_loc = location_at_loc.find(
        'span', {'class': 'text-body-small'}
    )
    user_location = location_at_loc.get_text().strip()
except: ...

# место работы
user_work = ''
try:
    work_at_loc = intro.find("div", {'class': 'inline-show-more-text'})
    user_work = work_at_loc.get_text().strip()
except: ...

# количество отслеживающих и контактов
user_viewwers, user_contacts = '0', '0'
try:
    stat_at_loc = soup.find(
        "ul", {'class': 'pv-top-card--list pv-top-card--list-bullet'}
    )
    user_viewwers = stat_at_loc.find_all("span")[0].get_text().strip()
    user_contacts = stat_at_loc.find_all("span")[2].get_text().strip()
except: ...

# общие сведения
user_common_info = ''
try:
    common_at_loc = soup.find("div", {'class': 'display-flex ph5 pv3'})
    user_common_info = common_at_loc.find_all('span')[0].get_text().strip()
except: ...

# должность
user_position = ''
try:
    position_at_loc = soup.find("ul", {'class': 'pvs-list'})
    user_position = position_at_loc.find_all('span')[0].get_text().strip()
except: ...

return [
    user_name, user_head, user_work, user_position, user_tags,
    user_location, user_viewwers, user_contacts, user_common_info
]

```

```

In [10]: # парсим данные публикации
def get_post_info(post):
    """

```

```

Функция на вход получает блок кода с публикацией.
Возвращает список параметров публикации: текст и реакции.
"""

# текст поста
post_text = 'no text'
try:
    post_text = post.find(
        'span', {'class': 'break-words'})
    ).get_text().strip()
except: ...

# блок реакций на пост
likes, comments, reposts = '0', '0', '0'
try:
    reactions = post.find('ul', {'class': 'social-details-social-counts'})
    try:
        likes = reactions.find(
            'span', {'class': 'social-details-social-counts__reactions-count'})
        ).get_text().strip().replace('\xa0', ' ')

    except: ...
    try:
        comments = reactions.find(
            'li', {'class': 'social-details-social-counts__comments'})
        ).get_text().strip().replace('\xa0', ' ')
        comments = re.match('^\d+', comments)[0]
    except: ...
    try:
        reposts = reactions.find(
            'li', {'class': 'social-details-social-counts__item social-details-social-cou'})
        ).get_text().strip().replace('\xa0', ' ')
        reposts = re.match('^\d+', reposts)[0]
    except: ...
except: ...

return [post_text, likes, comments, reposts]

```

1.2. Поиск и сбор целевых профилей

Открываем в браузере LinkedIn

```

In [13]: # запускаем браузер
driver = chrome_start()

```

```

In [14]: # входим в LinkedIn
linkedin_login(driver)

```

Поисковые запросы и параметры парсинга

Результаты парсинга поисковых запросов будем сохранять в отдельные файлы, позже соберём в один.

```

In [15]: # параметры поисковых запросов, теги, темы публикаций

#KEYWORDS = 'разработка no'
#TAGS = ['softwaredevelopment', 'webdevelopment', 'startup', 'it', 'design']
#CSV_FILE_NAME = os.path.join(DATA_PATH, 'profiles_id_1.csv')

#KEYWORDS = 'devops'
#TAGS = ['devops', 'aws', 'python', 'cloud', 'kubernetes']
#CSV_FILE_NAME = os.path.join(DATA_PATH, 'profiles_id_2.csv')

#KEYWORDS = 'data science'
#TAGS = ['datascience', 'machinelearning', 'ai', 'artificialintelligence', 'dataanalytics']

```



```
#CSV_FILE_NAME = os.path.join(DATA_PATH, 'profiles_id_3.csv')

#KEYWORDS = 'project management'
#TAGS = ['projectmanagement', 'business', 'agile', 'scrum', 'it']
#CSV_FILE_NAME = os.path.join(DATA_PATH, 'profiles_id_4.csv')

#KEYWORDS = 'design ui ux'
#TAGS = ['design', 'webdesign', 'ux', 'ui', 'uxdesign', 'uidesign']
#CSV_FILE_NAME = os.path.join(DATA_PATH, 'profiles_id_5.csv')

KEYWORDS = 'data analyst'
TAGS = ['datascience', 'dataanalytics', 'machinelearning', 'data', 'analytics']
CSV_FILE_NAME = os.path.join(DATA_PATH, 'profiles_id_6.csv')
```

Собираем ID пользователей

```
In [16]: # число страниц для парсинга, в бесплатном аккаунте доступно не более 100
# для примера работы скрипта установлены 2 страницы, при реальном парсинге
# нужно выставить максимальное значение
NUM_PAGES = 2

# пустой датафрейм для id пользователей
df = pd.DataFrame(columns=['id'])

for page_num in range(1, NUM_PAGES+1):

    # выводим номер страницы, в случае сбоя можно
    # будет начать новый парсинг с нее
    print(page_num, end=' ')

    # формируем url запроса
    people_url = search_people_url(KEYWORDS, TAGS, page_num=page_num)

    # запрашиваем и открываем страницу
    driver.get(people_url)

    # получаем и добавляем список найденных id профилей на странице
    profiles_id = get_profiles(driver)

    # добавляем данные в датафрейм
    df = pd.concat(
        [df, pd.DataFrame({'id': profiles_id})]
    ).reset_index(drop=True)

    # сохраняем в CSV
    df.to_csv(CSV_FILE_NAME)

    # быстро спим и за работу...
    time.sleep(random.uniform(3, 5))

1 2
```

```
In [17]: # закрываем браузер
driver.quit()
```

Собираем все id в один датафрейм

```
In [11]: # имя файла для сохранения профилей юзеров
CSV_PROFILES_FILE_NAME = os.path.join(DATA_PATH, 'profiles.csv')

# названия столбцов для хранения данных о пользователях
profile_columns = [
    'user_name', # имя
    'user_head', # заголовок
    'user_work', # последнее/текущее место работы
    'user_position', # должность
    'user_tags', # теги, интересы
```

```

'user_location', # адрес
'user_viewers', # число подписчиков
'user_contacts', # число контактов
'user_common_info' # общая информация
]

```

```

In [12]: # если файл с профилями уже существует
if os.path.exists(CSV_PROFILES_FILE_NAME):

    # загружаем датафрейм из файла
    profiles = pd.read_csv(CSV_PROFILES_FILE_NAME, index_col=0)

else:
    # список файлов с id пользователей
    list_csv_files = [
        'profiles_id_1.csv',
        'profiles_id_2.csv',
        'profiles_id_3.csv',
        'profiles_id_4.csv',
        'profiles_id_5.csv',
    ]
    # пустой DF
    profiles = pd.DataFrame(columns=['id'])

    # соберем все файлы в один DF
    for csv_file in list_csv_files:
        csv_file_name = os.path.join(DATA_PATH, csv_file)
        profiles = pd.concat(
            [profiles, pd.read_csv(csv_file_name, index_col=0)]
        ).reset_index(drop=True)

    # удаляем дубли
    profiles = profiles.drop_duplicates()

    profiles = profiles.reindex(
        columns = profiles.columns.tolist() + profile_columns
    )

print('Всего профилей:', len(profiles))

```

Всего профилей: 1709

Результат

```

In [13]: # профили
profiles.id.info()

<class 'pandas.core.series.Series'>
Index: 1709 entries, 0 to 1864
Series name: id
Non-Null Count  Dtype
-----
1709 non-null   object
dtypes: object(1)
memory usage: 26.7+ KB

```

Мы выполнили поиск различных IT специалистов на *Linkedin* и собрали идентификаторы их профилей. В нашем распоряжении оказалось 1709 идентификаторов. Можем приступить к сбору данных о людях и парсингу постов.

1.3. Парсинг постов и профилей

```

In [22]: # запускаем браузер
driver = chrome_start()

```

```
In [23]: # входим в LinkedIn
linkedin_login(driver)
```

Парсим профили и посты

```
In [14]: # имя файла для сохранения публикаций
CSV_POSTS_FILE_NAME = os.path.join(DATA_PATH, 'posts.csv')

# названия столбцов для хранения публикаций
posts_columns = [
    'user_id', # id профиля
    'text', # текст публикации
    'likes', # количество реакций
    'comments', # количество комментариев
    'reposts', # количество комментариев
]
```

```
In [15]: # если файл с профилями уже существует
if os.path.exists(CSV_POSTS_FILE_NAME):
    # загружаем датафрейм из файла
    posts = pd.read_csv(CSV_POSTS_FILE_NAME, index_col=0)
else:
    # пустой датафрейм для текстов публикаций
    posts = pd.DataFrame(columns=posts_columns)
```

Т.к. процесс парсинга может прерваться по разным причинам, например блокировка аккаунта или потеря связи с LinkedIn, то желательно запомнить позицию, на которой процесс парсинга остановился. Это даст возможность продолжить сбор данных с того места, где остановились.

```
In [16]: # с какого профиля стартуем
# если ранее парсинг был прерван, продолжаем с того же места
start_idx = profiles.user_name.nunique()
start_idx
```

Out[16]: 428

```
In [27]: # парсим данные из профилей
# для примера работы скрипта выборка сделана от start_idx до start_idx+1,
# в боевых условиях start_idx+1 нужно удалить
for profile_id in profiles.id[start_idx:start_idx+1]:

    # для контроля выводим на экран текущий ID профиля
    print(profile_id)

    # получаем url профиля пользователя
    profile_url = f'https://www.linkedin.com/in/{profile_id}/'

    # открываем ссылку profile_url
    driver.get(profile_url)

    # парсим информацию профиля
    user_info = get_user_info(driver, profile_id)

    # сохраняем данные в датафрейм
    profiles.loc[profiles.id == profile_id, profile_columns] = user_info

    # сохраняем данные профилей в CSV
    profiles.to_csv(CSV_PROFILES_FILE_NAME)

    # пауза
    time.sleep(random.uniform(10, 20))

    # URL на все публикации пользователя
    posts_url = f'https://www.linkedin.com/in/{profile_id}/recent-activity/all/'
```

```

driver.get(posts_url)

# получаем код проскроленной страницы
src = get_scrolled_page(driver, num_scrolls=25, pause_time=0.5).page_source

# передаем код страницы в парсер
soup = BeautifulSoup(src, 'lxml')

# получаем список постов
posts_block = soup.find_all(
    'li', {'class': 'profile-creator-shared-feed-update__container'})

print(f'posts: {len(posts_block)}')

count_posts = 1

# парсим посты
for post in posts_block:

    # номер поста для контроля
    print(count_posts, end=' ')
    count_posts += 1

    # получаем данные публикации
    post_info = get_post_info(post)

    if not post_info[0] == 'no text':
        # добавляем данные в датафрейм
        posts.loc[len(posts.index)] = [profile_id] + post_info

    # сохраняем в CSV
    posts.to_csv(CSV_POSTS_FILE_NAME)

print()

```

kamushken

posts: 169

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65
66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96
97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 1
44 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 16
7 168 169

```

In [28]: # закрываем браузер
driver.quit()

Результат

In [17]: # профили
profiles.info()

```
<class 'pandas.core.frame.DataFrame'>
Index: 1709 entries, 0 to 1864
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  -
0   id                     1709 non-null   object
1   user_name              428 non-null    object
2   user_head              428 non-null    object
3   user_work              398 non-null    object
4   user_position          428 non-null    object
5   user_tags              140 non-null    object
6   user_location          426 non-null    object
7   user_viewers           430 non-null    object
8   user_contacts          430 non-null    object
9   user_common_info       399 non-null    object
dtypes: object(10)
memory usage: 146.9+ KB
```

```
In [18]: posts.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 9504 entries, 0 to 9503
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   user_id     9504 non-null   object
1   text        9504 non-null   object
2   likes       9504 non-null   object
3   comments    9504 non-null   int64
4   reposts     9504 non-null   int64
dtypes: int64(2), object(3)
memory usage: 445.5+ KB
```

Вывод:

Мы собрали список аккаунтов пользователей сети *Linkedin* потенциально целевой аудитории. Выполнили сбор данных из профилей пользователей и их публикаций.

Нам не удалось получить информацию по всем запланированным профилям пользователей т.к. учетные записи, с помощью которых собирались данные, были заблокированы сервисом *Linkedin*.

Но, в результате мы смогли собрать данные на более чем 400 пользователей и более 9 тыс. постов.

Получение и объединение 5 датасетов с команды № 2, 3, 4, 8 и 10

В течение хакатона обменялись датасеты с разных команд в целях улучшения данных и повышения точности

Датасет нашей команды №2

```
In [19]: # оценим датафрейм с постами
posts.head(2)
```

Out[19]:

	user_id	text	likes	comments	reposts
0	ali-wodan	Кстати говоря. Теперь подкаст Миражи доступен в соцсети Вконтакте: https://lnkd.in/gKkrJX9Я наконец разобрался как туда прикрутить RSS :-) #podcast #миражи	1	0	0
1	ali-wodan	I'm #hiring. Know anyone who might be interested?	1	0	0

In [20]:

```
# оценим датафрейм с информацией о пользователях
profiles.head(2)
```

Out[20]:

	id	user_name	user_head	user_work	user_position	user_tags	user_location	user_viewers	us
0	ali-wodan	Ali Wodan	Head of Design	Performix	Head Of Design	podcast, it	Москва, Московская область, Россия	2 391	
1	ikotow	Игорь Котов	Директор по производству – Технократия	Технократия	Технократия	it, обучение, менеджмент, технологии, производство	Казань, Республика Татарстан, Россия	340	

In [21]:

```
# переименуем столбец text в post для лучшего отражения содержимого
posts = posts.rename(columns={'text': 'post'})
```

Объединим датафреймы

In [22]:

```
# переименуем столбец id в user_id в датафрейме profiles,
# для последующего объединения с posts
profiles = profiles.rename(columns={'id': 'user_id'})
```

In [23]:

```
# объединяем датафреймы
dataset_from_team_2 = pd.merge(posts, profiles, on='user_id')
```

In [24]:

```
# удаляем дубликаты
dataset_from_team_2.drop_duplicates(inplace=True)
```

In [25]:

```
# удаляем из столбца likes точки, запятые и пробелы
dataset_from_team_2["likes"] = dataset_from_team_2["likes"].replace(
    r'\.|\,|\s', '', regex=True
)

# меняем тип данных столбца likes на integer
dataset_from_team_2["likes"] = dataset_from_team_2["likes"].astype("int64")
```

In [26]:

```
# смотрим что получилось
dataset_from_team_2.sample(2)
```

Out[26]:

	user_id	post	likes	comments	reposts	user_name	user_head	user_w
8074	nicholasvasilkov	Thank you www.mesto.co for the great content with Nick Davidov https://lnkd.in/ggqkGsgB	2	0	0	Nicholas (Nick) Vasilkov	Sales & Marketing departments for business and entrepreneurs. IT-tools integration. CRM. Support.	Vasilkov.Dig

7185	nikolay-selin	Today is a special day for me, because my 7-year journey at Magora comes to an end. It's been a rocky road from a rookie to a sales professional and I still have so much to learn.I feel grateful for the smart, optimistic and curious people I've met along the way, both inside and outside the team. For good friends and long-term business relationships. For a new set of skills and a new city to call home. For the ability to take a small part in launching amazing tech products.Now, it's time for a new challenge — more on that in the next post. Wish me luck!	49	10	0	Nick Selin	Turning your ideas into a high-demand tech product in web, mobile and AI space	Roo
------	---------------	---	----	----	---	------------	--	-----

In [27]:

dataset_from_team_2.info()

```
<class 'pandas.core.frame.DataFrame'>
Index: 9412 entries, 0 to 9503
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   user_id                9412 non-null   object
1   post                   9412 non-null   object
2   likes                  9412 non-null   int64
3   comments               9412 non-null   int64
4   reposts                9412 non-null   int64
5   user_name              9412 non-null   object
6   user_head              9412 non-null   object
7   user_work              8880 non-null   object
8   user_position          9412 non-null   object
9   user_tags              3183 non-null   object
10  user_location          9374 non-null   object
11  user_viewers           9412 non-null   object
12  user_contacts          9412 non-null   object
13  user_common_info       9005 non-null   object
dtypes: int64(3), object(11)
memory usage: 1.1+ MB
```

```
In [28]: # Сохраняем датафрейм
dataset_from_team_2.to_csv(os.path.join(DATA_PATH, 'dataset_from_team_2.csv'))
```

Мы получили датасет, который содержит следующие поля:

- `user_id` - идентификатор пользователя *LinkedIn*,
- `post` - текст поста,
- `likes` - число лайков поста,
- `comments` - число комментариев к посту,
- `reposts` - число репостов,
- `user_name` - имя пользователя,
- `user_head` - подпись пользователя, обычно тут указывают специализацию, например Data Analyst,
- `user_work` - текущее или последнее место работы пользователя,
- `user_position` - должность,
- `user_tags` - теги, которые пользователь указал в своем профиле,
- `user_location` - место жительства,
- `user_viewers` - число фолловеров, т.е. других пользователей, отслеживающих активность данного пользователя,
- `user_contacts` - число контактов,
- `user_common_info` - информация пользователя о себе.

Датасет команды №3

```
In [29]: dataset_from_team_3 = pd.read_csv(
        os.path.join(DATA_PATH, 'dataset_from_team_3.csv'), index_col=0
    )
```

```
In [30]: dataset_from_team_3.info()
```



```
<class 'pandas.core.frame.DataFrame'>
Index: 304 entries, 0 to 487
Data columns (total 8 columns):
#   Column          Non-Null Count  Dtype
---  -
0   name             304 non-null    object
1   status           304 non-null    object
2   company          304 non-null    object
3   url              304 non-null    object
4   text             304 non-null    object
5   likes_cnt        297 non-null    float64
6   reposts_cnt      304 non-null    int64
7   comments_cnt     304 non-null    int64
dtypes: float64(1), int64(2), object(5)
memory usage: 21.4+ KB
```

```
In [31]: # Проверим на наличие дубликатов
dataset_from_team_3.duplicated().sum()
```

Out[31]: 48

```
In [32]: # Устраняем их
dataset_from_team_3.drop_duplicates(inplace=True)
```

```
In [33]: # Проверка на пропущенные значения
dataset_from_team_3.isna().sum()
```

Out[33]:

name	0
status	0
company	0
url	0
text	0
likes_cnt	7
reposts_cnt	0
comments_cnt	0

dtype: int64

```
In [34]: # Переименуем названия колонки под нашими названиями датасеты
dataset_from_team_3 = dataset_from_team_3.rename(columns={
    'text': 'post', 'name': 'user_name', 'status': 'user_head',
    'company': 'user_work', 'likes_cnt': 'likes',
    'reposts_cnt': 'reposts', 'comments_cnt': 'comments'
})
```

```
In [35]: display(dataset_from_team_3.head(2))

display(dataset_from_team_3.tail(2))
```

	user_name	user_head	user_work	url	post
0	Michil Egorov	Middle Software Engineer - Yandex	Yandex	https://www.linkedin.com/in/michilegorov	Всем привет!Выпустил свою первую статью на хабр! https://lnkd.in/dt9N6D7B Статья про историю и технологии разработки игры https://guess-word.com и как мы создали игру с элементами машинного обучения и вышли в ноль за 2 месяцаПри внимательном прочтении вы даже сможете запустить первую версию игры!
1	Michil Egorov	Middle Software Engineer - Yandex	Yandex	https://www.linkedin.com/in/michilegorov	Если вам интересно позалипать в слова, я запустил игру! https://guess-word.com/ Особенно понравится братьям NLP-шникам)

	user_name	user_head	user_work	url	post	likes	reposts	comments
453	Matvey Popov	Software Engineer at Yandex	Yandex	https://www.linkedin.com/in/matvey-popov	<p>My Russian speaking friends keep getting discriminated due to the language they speak. To all of my friends globally, please remember few things:1. Russian speaking does not equal Russian. There were 15 countries in the USSR. All of those countries still have Russian speaking minorities. It doesn't mean they are Russian or identify themselves as Russian. Just like Irish does not equal English or Spanish does not equal Mexican.2. Russian does not equal aggressor. None of the Russians support the war. Some just don't understand what is happening due to the limited information that they are getting. There is no free media left in Russia.3. Ukrainians also speak Russian, some just Russian. Next time you tell a Russian speaker you wont serve them, think about which side you're taking. The person approaching you might have a relative sitting in the bomb shelter right now. 4. Russian name also does not equal Russian. My name is Russian. I was born in Latvia, my parents were born in Latvia. I have Latvian, Ukrainian, Polish, Turkish, Romanian and probably many more ethnicities in me. I grew up in Ireland. My Russian surname was inherited from my great grandfather who was a pacifist, he went through the war and wouldn't ever stand by what's happening right now. Many Ukrainians have Russian surnames too. 5. The anger you're translating on to innocent people is not going to solve the problem, it's going to create more problems and hatred. There are</p>	0.0	0	

	user_name	user_head	user_work	url	post	likes	reposts	
					many ways to help, but hating on others is definitely not one of the ways.#StandWithUkraine UA			
454	Matvey Popov	Software Engineer at Yandex	Yandex	https://www.linkedin.com/in/matvey-popovv	I'm happy to share that I'm starting a new position as Software Engineer at Tinkoff	0.0	0	

Датасет команды №4

```
In [36]: dataset_from_team_4 = pd.read_csv(
        os.path.join(DATA_PATH, 'dataset_from_team_4.csv'), delimiter=';'
    )
```

```
In [37]: dataset_from_team_4.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1191 entries, 0 to 1190
Data columns (total 6 columns):
 #   Column                Non-Null Count  Dtype  
---  -
 0   url_user              1191 non-null   object 
 1   name                  796 non-null    object 
 2   job                   796 non-null    object 
 3   text_post             796 non-null    object 
 4   react_per_user        796 non-null    object 
 5   count_comments        796 non-null    float64
dtypes: float64(1), object(5)
memory usage: 56.0+ KB
```

```
In [38]: # Проверим на наличие дубликатов
dataset_from_team_4.duplicated().sum()
```

```
Out[38]: 84
```

```
In [39]: # Устраняем их
dataset_from_team_4.drop_duplicates(inplace=True)
```

```
In [40]: # Проверка на пропущенные значения
dataset_from_team_4.isna().sum()
```

```
Out[40]: url_user          0
name              317
job               317
text_post         317
react_per_user    317
count_comments    317
dtype: int64
```

```
In [41]: # Устраняем их
dataset_from_team_4.dropna(inplace=True)
```

```
In [42]: # Переименуем названия колонки под нашими названиями датасеты
dataset_from_team_4 = dataset_from_team_4.rename(columns={
    'url_user': 'url', 'name': 'user_name', 'job': 'user_head',
    'text_post': 'post', 'react_per_user': 'likes',
    'count_comments': 'comments'
})
```

```
In [43]: dataset_from_team_4['likes'] = dataset_from_team_4['likes'].str.replace(
        "''", ',')
    )
dataset_from_team_4['likes'] = dataset_from_team_4['likes'].str.replace(
        " ", '')
    )
dataset_from_team_4['likes'] = dataset_from_team_4['likes'].str.replace(
        '[\[\]]+', '', regex=True
    )
```

```
In [44]: def calculate_median(row):
    # Удаление всех символов, кроме цифр, из строки
    numbers = ''.join(filter(str.isdigit, row))

    # Проверка на пустой список
    if not numbers:
        return None

    # Преобразование строки с числами в список целочисленных значений
    numbers_list = list(map(int, numbers))

    # Расчет максимального значения
    max_value = np.max(numbers_list)

    return max_value
```

```
In [45]: # Применение функции к замене колонки Likes на кол-во лайков
dataset_from_team_4['likes'] = dataset_from_team_4['likes'].apply(
    calculate_median
)
```

```
In [46]: display(dataset_from_team_4.head(2))

display(dataset_from_team_4.tail(2))
```

	url	user_name	user_head	post	likes	comments
0	https://www.linkedin.com/in/artem-reshetnikov-925143251/	Artem Reshetnikov	Data Analyst	['I love SQL.']	5.0	0.0
1	https://www.linkedin.com/in/korenevich/	Pavel Karanevich	Growth Evangelist Entrepreneur US Marketer Advisor	['Приложение которое из голоса раскидывает задачи. Идея огонь!']	7.0	0.0

1189 https://www.linkedin.com/in/%D0%BE%D0%BB%D0%B5%D1%81%D1%8F-%D1%86%D0%B0%D1%80%D0%B5%D0%B3%D0%BE%D1%80%D0%BE%D0%B4%D1%86%D0%B5%D0%B2%D0%B0-748179237?miniProfileUrn=urn%3Ali%3Afs_miniProfile%3AACoAADrvc-gBOKDoqybZ93sYw_gTHsGQU27rlGw

1190 https://www.linkedin.com/in/%D0%BE%D0%BB%D0%B5%D1%81%D1%8F-%D1%86%D0%B0%D1%80%D0%B5%D0%B3%D0%BE%D1%80%D0%BE%D0%B4%D1%86%D0%B5%D0%B2%D0%B0-748179237?miniProfileUrn=urn%3Ali%3Afs_miniProfile%3AACoAADrvc-gBOKDoqybZ93sYw_gTHsGQU27rlGw

Датасет команды №8

В ходе получения датасеты с команды 8 были обнаружены неточности, в которой сообщается, что индексы не нумерируются должным образом, что и было решено разбить CSV файла на 2 части

Часть 1

```
In [47]: dataset_from_team_8_1 = pd.read_csv(os.path.join(
        DATA_PATH, 'dataset_from_team_8_1.csv'
    ), delimiter=';', index_col=0)
```

```
In [48]: dataset_from_team_8_1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 112 entries, 0 to 103
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  -
0   profile_url      98 non-null     object
1   name             98 non-null     object
2   works_at         98 non-null     object
3   exp_list         98 non-null     object
4   post             98 non-null     object
5   reactions_cnt    98 non-null     float64
6   comments_cnt     98 non-null     float64
7   post_url         98 non-null     object
8   posts_cnt        98 non-null     float64
dtypes: float64(3), object(6)
memory usage: 8.8+ KB
```

```
In [49]: # Проверим на наличие дубликатов
dataset_from_team_8_1.duplicated().sum()
```

```
Out[49]: 13
```

```
In [50]: # Устраняем их
dataset_from_team_8_1.drop_duplicates(inplace=True)
```

```
In [51]: # Проверка на пропущенные значения
dataset_from_team_8_1.isna().sum()
```

```
Out[51]: profile_url      1
name              1
works_at          1
exp_list          1
post              1
reactions_cnt     1
comments_cnt      1
post_url          1
posts_cnt         1
dtype: int64
```

```
In [52]: # Устраняем их
dataset_from_team_8_1.dropna(inplace=True)
```

```
In [53]: # Переименуем названия колонки под нашими названиями датасеты
dataset_from_team_8_1 = dataset_from_team_8_1.rename(columns={
    'profile_url': 'url', 'name': 'user_name', 'job': 'user_head',
    'works_at': 'user_head', 'exp_list': 'user_position',
    'reactions_cnt': 'likes', 'comments_cnt': 'comments',
    'posts_cnt': 'reposts'
})
```

```
In [54]: display(dataset_from_team_8_1.head(2))

display(dataset_from_team_8_1.tail(2))
```

	url	user_name	user_head	user_position	post	lik
0	https://www.linkedin.com/in/ruslandubrovin/	Руслан Дубровин	Software Developer – Yandex	['Software Developer'Yandex'март 2019 г. – настоящее время · 4\ха0г. 4\ха0мес.'Lead Software Developer'TheQuestion'июль 2018 г. - авг. 2021 г. · 3\ха0г. 2\ха0мес.'Software Developer'Технократия (worked as outstaff for redmadrobot)'сент. 2017 г. - июнь 2018 г. · 10 мес.'Golang developer'infotech.group'нояб. 2016 г. - сент. 2017 г. · 11 мес.'Python developer'Cinarra Systems'апр. 2016 г. - нояб. 2016 г. · 8 мес.']	нет постов	
1	https://www.linkedin.com/in/grigory-kostin-aaa16061/	Grigory Kostin	Developer at Yandex	['Developer'Yandex'январ. 2015 г. – настоящее время · 8\ха0лет 6\ха0мес.'HeadHunter Group'2\ха0г.\ха05\ха0мес.'Senior Developer'апр. 2014 г. - дек. 2014 г. · 9 мес.'Developer'авг. 2012 г. - апр. 2014 г. · 1\ха0г. 9\ха0мес.']	нет постов	
	url	user_name	user_head	user_position	post	lik
102	https://www.linkedin.com/in/ilias-iliasov-434a47251/	Ilias Iliasov	Senior Java Developer	['Senior Developer'Stack Overflow'Полный рабочий день'сент. 2018 г. – настоящее время · 10\ха0мес.'Гибкий формат работы'Developer'Russia · рабочий день'сент. 2018 г. - сент. 2018 г. · 11 мес.'My Overview'Моя карьера']		
103	https://www.linkedin.com/in/%D0%B0%D0%BD%D1%82%D0%BE%D0%BD-%D0%B3%D1%80%D0%B8%D1%88%D0%B8%D0%BD-2bb3a53a/	Антон Гришин	Frontend-developer	['experience']		

Часть 2

```
In [55]: dataset_from_team_8_2 = pd.read_csv(os.path.join(
        DATA_PATH, 'dataset_from_team_8_2.csv'
    ), delimiter=';', index_col=0)
```

```
In [56]: dataset_from_team_8_2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 193 entries, 0 to 149
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  -
0   profile_url      169 non-null    object
1   name             169 non-null    object
2   works_at         169 non-null    object
3   exp_list         169 non-null    object
4   post             169 non-null    object
5   reactions_cnt    169 non-null    float64
6   comments_cnt     169 non-null    float64
7   post_url         169 non-null    object
8   posts_cnt        169 non-null    float64
dtypes: float64(3), object(6)
memory usage: 15.1+ KB
```

```
In [57]: # Проверим на наличие дубликатов
dataset_from_team_8_2.duplicated().sum()
```

```
Out[57]: 23
```

```
In [58]: # Устраняем их
dataset_from_team_8_2.drop_duplicates(inplace=True)
```

```
In [59]: # Проверка на пропущенные значения
dataset_from_team_8_2.isna().sum()
```

```
Out[59]: profile_url      1
name              1
works_at          1
exp_list          1
post              1
reactions_cnt     1
comments_cnt      1
post_url          1
posts_cnt         1
dtype: int64
```

```
In [60]: # Устраняем их
dataset_from_team_8_2.dropna(inplace=True)
```

```
In [61]: # Переименуем названия колонки под нашими названиями датасеты
dataset_from_team_8_2 = dataset_from_team_8_2.rename(columns={
    'profile_url': 'url', 'name': 'user_name', 'job': 'user_head',
    'works_at': 'user_head', 'exp_list': 'user_position',
    'reactions_cnt': 'likes', 'comments_cnt': 'comments',
    'posts_cnt': 'reposts'
})
```

```
In [62]: display(dataset_from_team_8_2.head(2))

display(dataset_from_team_8_2.tail(2))
```


	url	user_name	user_head	use
				[iOS Dev Полный день'июн настоящее
				Developer'h Полный день'март 2022 2023 4\ха0мес.\u200e Passwords & Developer'Atlas Полный день'сент. 2021 2022 г. · 6 мес.\u
0	https://www.linkedin.com/in/cbelkin/	Constantine Belkin	iOS Developer at VK	Developer'amazi Полный день'сент. 2020 2021 г. · 10 ме developer'amazi Полный день'июль 20 2020 10
1	https://www.linkedin.com/in/%D0%B0%D1%80%D1%82%D0%B5%D0%BC-%D1%88%D0%BB%D1%8F%D1%85%D1%82%D0%B8%D0%BD-bb112390/	Артём Шляхтин	Senior iOS Developer at Sberbank	[Developer' Полный день'нояб настоящее врем 8\ха0мес.' Developer'IBM'i г. - нояб. 2018 5\х Developer'RosEur 2015 г. - июль г.'Разработчик Tech'май 20 мес.'Рекомен письмо'-'Индиви предприним 2013 г. - авг. 20

	url	user_name	user_head	user_position	post	likes
148	https://www.linkedin.com/in/ivan-sergunin-2676b8201/	Ivan Sergunin	iOS Developer at Sberbank	['iOS Developer'Sberbank · Полный рабочий день'январ. 2021 г. – настоящее время · 2\ха0г. 6\ха0мес.'iOS Developer'SPB TV · Полный рабочий день'нояб. 2014 г. – дек. 2020 г. · 6\ха0лет 2\ха0мес.]	нет постов	0.0
149	https://www.linkedin.com/in/igor-shvetsov-6a081713/	Igor Shvetsov	iOS Developer at Tinkoff Digital	['iOS Developer'Tinkoff Bank · Полный рабочий день'апр. 2020 г. – настоящее время · 3\ха0г. 3\ха0мес.'Developer'Noveo Group'окт. 2015 г. – сент. 2019 г. · 4 г.'iOs Developer'iOS Developer'Mail.ru Group'2019 · Менее года'MTS'9\ха0лет\ха011\ха0мес.'IT department'дек. 2005 г. – окт. 2015 г. · 9\ха0лет 11\ха0мес.'Senior Developer'дек. 2005 г. – окт. 2015 г. · 9\ха0лет 11\ха0мес.'Developer'ClearScale'2013 · Менее года']	нет постов	0.0

Датасет команды №10

```
In [63]: dataset_from_team_10 = pd.read_csv(
        os.path.join(DATA_PATH, 'dataset_from_team_10.csv')
    )
```

```
In [64]: dataset_from_team_10.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype  
---  -
 0   account_link          500 non-null    object  
 1   search_keywords       500 non-null    object  
 2   name                  500 non-null    object  
 3   title                 500 non-null    object  
 4   works_at              446 non-null    object  
 5   intro                 500 non-null    object  
 6   experience            500 non-null    float64 
 7   place                 500 non-null    object  
 8   posts_cnt             500 non-null    int64   
 9   post_text             500 non-null    object  
10   reaction_cnt          350 non-null    float64 
11   comments_cnt          164 non-null    float64 
12   repost_cnt            170 non-null    float64 
dtypes: float64(4), int64(1), object(8)
memory usage: 50.9+ KB
```

```
In [65]: # Проверим на наличие дубликатов
dataset_from_team_10.duplicated().sum()
```

```
Out[65]: 3
```

```
In [66]: # Устраняем их
dataset_from_team_10.drop_duplicates(inplace=True)
```

```
In [67]: # Проверка на пропущенные значения
dataset_from_team_10.isna().sum()
```

```
Out[67]: account_link      0
search_keywords      0
name                 0
title                0
works_at             52
intro                0
experience            0
place                0
posts_cnt            0
post_text            0
reaction_cnt         149
comments_cnt         335
repost_cnt           329
dtype: int64
```


```
In [68]: # Устраняем их
dataset_from_team_10.dropna(inplace=True)
```

```
In [69]: # Переименуем названия колонки под нашими названиями датасеты
dataset_from_team_10 = dataset_from_team_10.rename(columns={
    'account_link': 'url', 'search_keywords': 'user_head',
    'name': 'user_name', 'title': 'user_tags', 'works_at': 'user_work',
    'intro': 'user_common_info', 'experience': 'user_experience',
    'place': 'user_location', 'post_text': 'post', 'reaction_cnt': 'likes',
    'comments_cnt': 'comments', 'repost_cnt': 'reposts'
})
```

```
In [70]: dataset_from_team_10 = dataset_from_team_10.drop('posts_cnt', axis=1)
```

```
In [71]: display(dataset_from_team_10.head(1))

display(dataset_from_team_10.tail(1))
```

	url	user_head	user_name	user_tags	user_work
17	https://www.linkedin.com/in/dm-bychkov	frontend	Dmitrii Bychkov 	Frontend Web Developer JavaScript TypeScript React Redux HTML CSS Node.js SQL	SmartMechanica Всем привет!) разработчик.Я люблю время посвящаю изуче анализировать все обдумывать различные ве поиске самого эф Frontend: JavaScript, React Backend: Node.js, Express. Работа с REST API, Gi TypeScript, Priz развивающейся компани меня в: профессиональный рос 85jsmaildm@gmail.com привет!)\r\n\r\nМен разработчик.\r\n свободное вре стека.\r\nМне н происходит вокруг меня, сценариев моих действий решения.\r\n\r\nМой стек Redux Toolkit, HTML, Express.js, PostgreSQL, SQL API, Git, Webpack, Jes Prizma, Rec развиваю командой.\r\nДля меня в: профессиональный рост. 85\r\njsmaildm@gmail.com

	url	user_head	user_name	user_tags	user_work	user_com
						Любознательный разработчик с 3-х летни работы. 3 филологический фак специальности «учитель языка и литературы стилистически, орфограф грамматически п оставлять коммэ коде.Мой стек: J TypeScript, React, Red HTML5, CSS3, Node.js Sessions, Bcrypt, Pc Sequelize ORM, Slack, Tre настоящее время знако Vue.js, а также меня вдо работы Бруно Симона, г активно изучаю three.js
499	https://www.linkedin.com/in/alena-krupennikova-7b6376278	frontend	Alena Krupennikova	Frontend Dev • JavaScript • TypeScript • React • Redux • React Native	Smart Kids	Frontend-разработчик с 3 опытом работы. 3 филологический фак специальности «учитель языка и литературы стилистически, орфограф грамматически п оставлять коммэ коде.\r\n\r\nМой стек: J TypeScript, React, Red HTML5, CSS3, Node.js Sessions, Bcrypt, Pc Sequelize ORM, Sl Miro.\r\n\r\nВ настоящ знакомлюсь со Vue.js, а та вдохновляют рабс Симона, поэтому я активн three.js.\r\n\r\nhttps://t.me/krupeshaaa\r\n+7 999 813 50 98\r\n\r\n

Объединение датасетов

Примерный суммарный размер датасет

```
In [72]: shape_sum_dataset = (
    dataset_from_team_2.shape[0] + dataset_from_team_3.shape[0] + dataset_from_team_4.shape[0]
    dataset_from_team_2.shape[1] + dataset_from_team_3.shape[1] + dataset_from_team_4.shape[1]
)
print('Суммарный размер датасет:', shape_sum_dataset)

Суммарный размер датасет: (10810, 58)
```

Датафрейм 2 и 3 команды

```
In [73]: # Объединяем датафреймы
df = pd.merge(
    dataset_from_team_2, dataset_from_team_3,
    how='outer', suffixes=('_x', '_y')
)

print('Размер:', df.shape)
```

Размер: (9668, 15)

Датафрейм 4 команды

```
In [74]: # Объединяем датафреймы
df = pd.merge(df, dataset_from_team_4, how='outer', suffixes=('_x', '_y'))

print('Размер:', df.shape)
```

Размер: (10458, 15)

Датафрейм 8 команды

Часть 1

```
In [75]: # Объединяем датафреймы
df = pd.merge(df, dataset_from_team_8_1, how='outer')

print('Размер:', df.shape)
```

Размер: (10556, 16)

Часть 2

```
In [76]: # Объединяем датафреймы
df = pd.merge(df, dataset_from_team_8_2, how='outer')

print('Размер:', df.shape)
```

Размер: (10725, 16)

Датафрейм 10 команды

```
In [77]: # Объединяем датафреймы
df = pd.merge(df, dataset_from_team_10, how='outer')

print('Размер:', df.shape)
```

Размер: (10810, 17)

Обработка данных

Для дальнейшей работы с данными нам необходимо их подготовить, удалить из текста лишние символы, оставить только русскоязычные тексты, проверить все ли данные имеют правильный тип и т.д.

```
In [78]: df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10810 entries, 0 to 10809
Data columns (total 17 columns):
#   Column                Non-Null Count  Dtype
---  -
0   user_id                9412 non-null   object
1   post                   10810 non-null  object
2   likes                  10778 non-null  float64
3   comments               10810 non-null  float64
4   reposts                10020 non-null  float64
5   user_name              10810 non-null  object
6   user_head              10810 non-null  object
7   user_work              9221 non-null   object
8   user_position          9679 non-null   object
9   user_tags              3268 non-null   object
10  user_location          9459 non-null   object
11  user_viewers           9412 non-null   object
12  user_contacts          9412 non-null   object
13  user_common_info       9090 non-null   object
14  url                    1398 non-null   object
15  post_url               267 non-null    object
16  user_experience         85 non-null     float64
dtypes: float64(4), object(13)
memory usage: 1.4+ MB

```

```

In [79]: # оценим датафрейм с постами
df.head(2)

```

	user_id	post	likes	comments	reposts	user_name	user_head	user_work	user_position
0	ali-wodan	Кстати говоря. Теперь подкаст Миражи доступен в соцсети Вконтакте: https://lnkd.in/gKkrJX9Я наконец разобрался как туда прикрутить RSS :-) #podcast #миражи	1.0	0.0	0.0	Ali Wodan	Head of Design	Performix	Head Of Design
1	ali-wodan	I'm #hiring. Know anyone who might be interested?	1.0	0.0	0.0	Ali Wodan	Head of Design	Performix	Head Of Design

```

In [80]: df.isna().sum()

```

```
Out[80]:
```

user_id	1398
post	0
likes	32
comments	0
reposts	790
user_name	0
user_head	0
user_work	1589
user_position	1131
user_tags	7542
user_location	1351
user_viewers	1398
user_contacts	1398
user_common_info	1720
url	9412
post_url	10543
user_experience	10725
dtype:	int64

```
In [81]: # Заполняем пропуски нулями
df[['comments', 'reposts', 'likes']] = df[['comments', 'reposts', 'likes']]
df.fillna(0)

# преобразуем тип данных
df[['comments', 'reposts', 'likes']] = df[['comments', 'reposts', 'likes']]
df.astype('int')
```

```
In [82]: # Проверим
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10810 entries, 0 to 10809
Data columns (total 17 columns):
#   Column                Non-Null Count  Dtype
---  -
0   user_id               9412 non-null   object
1   post                 10810 non-null  object
2   likes                10810 non-null  int32
3   comments             10810 non-null  int32
4   reposts              10810 non-null  int32
5   user_name            10810 non-null  object
6   user_head            10810 non-null  object
7   user_work            9221 non-null   object
8   user_position        9679 non-null   object
9   user_tags            3268 non-null   object
10  user_location         9459 non-null   object
11  user_viewers         9412 non-null   object
12  user_contacts        9412 non-null   object
13  user_common_info     9090 non-null   object
14  url                  1398 non-null   object
15  post_url             267 non-null    object
16  user_experience       85 non-null     float64
dtypes: float64(1), int32(3), object(13)
memory usage: 1.3+ MB
```

Предобработка

```
In [83]: # функция удаления эмодзи
def remove_emojis(text):
    emoji_pattern = re.compile("[
        u"\U0001F600-\U0001F64F" # смайлики
        u"\U0001F300-\U0001F5FF" # символы и пиктограммы
        u"\U0001F680-\U0001F6FF" # транспорт и символы на карте
        u"\U0001F1E0-\U0001F1FF" # флаги
        u"\U00002500-\U00002BEF" # китайские символы
```



```
In [84]: # удалим посты на украинском языке
# определяем шаблон для украинских символов (по специфичным для данного языка символам)
ukrainian_pattern = r'[ЄєІіїіґґ]'

# создаем маску, указывающую строки, в которых столбец "post" содержит текст на украинском яз
mask = df['post'].str.contains(ukrainian_pattern, regex=True, na=False)

# сохраняем в датафрейме только строки, в которых маска имеет значение False
df = df[~mask]
```

```
In [85]: # сохраняем хэштеги в отдельный столбец перед их удалением из постов
df['hashtags'] = df['post'].str.findall(r'#([\^\s]+)').apply(
    lambda x: ', '.join(x)
)
```

В дальнейшем нам предстоит анализировать тексты постов, поэтому сразу выполним лемматизацию текстов и сохраним результат в отдельном столбце `post_lemmatized`.

```
In [91]: %%time
# функция лемматизации текста
morph = pymorphy2.MorphAnalyzer()
def lemmatize_text(text):
    lemmatized words = [
```

```
morph.parse(word)[0].normal_form for word in text.split() if morph.word_is_known(word)
]
return ' '.join(lemmatized_words)
```

лемматизируем посты

```
df['post_lemmatized'] = df['post'].apply(lemmatize_text)
```

CPU times: total: 36.4 s

Wall time: 54.8 s

In [92]:

скачиваем стоп-слова

```
nlTK.download('stopwords')
```

```
stop_words = set(stopwords.words('russian'))
```

еще один список от bukvarix.com - список стоп-слов Яндекс Wordstat

(этот список можно дополнить/изменить)

```
file_path_words = os.path.join(DATA_PATH, 'stop_words.txt')
```

```
with open(file_path_words, 'r', encoding='utf-8') as file:
```

```
stop_words_buk = file.read()
```

[nlTK_data] Downloading package stopwords to

[nlTK_data] C:\Users\krasn\AppData\Roaming\nlTK_data...

[nlTK_data] Package stopwords is already up-to-date!

In [93]:

удаляем стоп-слова и слова-паразиты

```
df['post_lemmatized'] = df['post_lemmatized'].apply(
    lambda x: ' '.join([word for word in x.split() if word not in stop_words])
)
```

```
df['post_lemmatized'] = df['post_lemmatized'].apply(
    lambda x: ' '.join(
        [word for word in x.split() if word.lower() not in stop_words_buk]
    )
)
```

Оставляем только посты содержащие буквы русского алфавита. Избавляемся от постов исключительно на иностранных языках.

In [94]:

определяем шаблон регулярного выражения для русских букв

```
pattern = '[^а-яА-ЯёЁ]'
```

создаем маску, чтобы проверить, содержит ли каждая ячейка русские буквы

```
mask = df['post_lemmatized'].str.contains(pattern, regex=True)
```

фильтруем датафрейм, используя маску

```
df = df[mask]
```

In [100...]

оценим качество подготовки текста

```
df.sample(1)
```

							Cybersecurity expert with 10 years of experience. Journalist, creator, blogger in information security. Web and marketing professional with 16 years of experience. Startup promoter. MBA in Information Technologies		
6219	antonlenskiy	пециалисты компании обнаружили в процессорах и на платформах и годов и новую уязвимость которая получила обозначение	1	0	0	Anton Lenskiy	INFOSECHUB.net	Co-Fo	



In [101...

df.info()

```
<class 'pandas.core.frame.DataFrame'>
Index: 2966 entries, 0 to 10809
Data columns (total 19 columns):
#   Column                Non-Null Count  Dtype
---  -
0   user_id                2204 non-null   object
1   post                  2966 non-null   object
2   likes                 2966 non-null   int32
3   comments              2966 non-null   int32
4   reposts              2966 non-null   int32
5   user_name             2966 non-null   object
6   user_head             2966 non-null   object
7   user_work             2276 non-null   object
8   user_position         2237 non-null   object
9   user_tags             604 non-null    object
10  user_location         2233 non-null   object
11  user_viewers          2204 non-null   object
12  user_contacts         2204 non-null   object
13  user_common_info     2055 non-null   object
14  url                   762 non-null    object
15  post_url              33 non-null     object
16  user_experience       36 non-null     float64
17  hashtags              2966 non-null   object
18  post_lemmatized       2966 non-null   object
dtypes: float64(1), int32(3), object(15)
memory usage: 428.7+ KB
```

Из 10 тыс. постов, пригодных для использования, осталось менее трех тысяч.

Мы получили датасет, который содержит следующие поля:

- `user_id` - идентификатор пользователя *LinkedIn*,
- `post` - текст поста,
- `likes` - число лайков поста,
- `comments` - число комментариев к посту,
- `reposts` - число репостов,
- `hashtags` - хештеги взятые из текста поста,
- `post_lemmatized` - лемматизированный текст поста,
- `user_name` - имя пользователя,
- `user_head` - подпись пользователя, обычно тут указывают специализацию, например Data Analyst,
- `user_work` - текущее или последнее место работы пользователя,
- `user_position` - должность,
- `user_tags` - теги, которые пользователь указал в своем профиле,
- `user_location` - место жительства,
- `user_viewers` - число фолловеров, т.е. других пользователей, отслеживающих активность данного пользователя,
- `user_contacts` - число контактов,
- `user_common_info` - информация пользователя о себе,
- `url` - ссылка пользователя,
- `post_url` - ссылка на пост,
- `user_experience` - стаж.

Сохранение датасетов

In [102...

```
# Сохраняем датафрейм лемматизации
df.to_csv(os.path.join(DATA_PATH, 'unity_datasets.csv'))
```

EDA

Итоговый датасет имеет некоторые проблемы, которые необходимо обработать:

- числовые поля `comments` и `reports` имеют тип `object`,
- есть пропуски в `user_work`, `user_tags`, `user_location` и `user_common_info`,
- пользовательские реакции представлены тремя полями `likes`, `comments` и `reposts`.

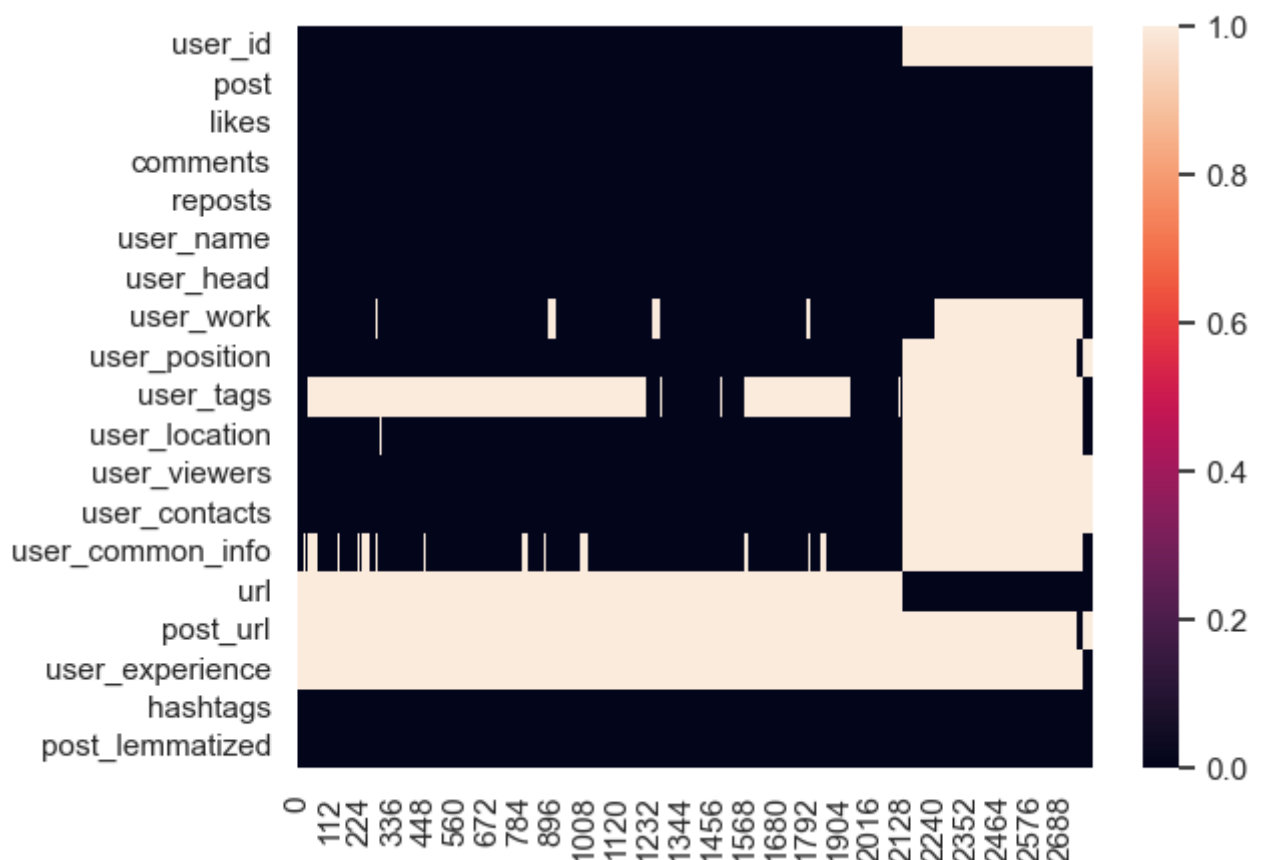
Возможно есть и другие проблемы. Рассмотрим подробнее.

```
In [103... # проверим на дубли в post_lemmatized
df.post_lemmatized.duplicated().sum()
```

```
Out[103]: 233
```

```
In [119... # удаляем дубликаты
df = df.drop_duplicates(subset='post_lemmatized', ignore_index=True)
```

```
In [120... # оценим визуально пропуски
sns.heatmap(df.isna().T);
```



Все поля, в которых имеются пропуски, просто не содержат информации, пользователи ее не указали, скрипт парсинга не смог корректно выявить эти данные на странице. В любом случае мы можем их заменить на знак "-" (минус или тире), это не должно повлиять на результаты анализа.

```
In [107... # % пропусков по полям датасета
round(df.isna().mean() * 100)
```

```
Out[107]: user_id      26.0
           post        0.0
           likes       0.0
           comments    0.0
           reposts     0.0
           user_name    0.0
           user_head    0.0
           user_work    23.0
           user_position 25.0
           user_tags    80.0
           user_location 25.0
           user_viewers 26.0
           user_contacts 26.0
           user_common_info 31.0
           url          74.0
           post_url     99.0
           user_experience 99.0
           hashtags     0.0
           post_lemmatized 0.0
           dtype: float64
```

```
In [108... columns_to_fill = [
            'user_id', 'user_work', 'user_tags', 'user_location',
            'user_common_info', 'url', 'post_url'
        ]

        columns_to_fill_dight = ['user_experience', 'user_viewers']

        # избавляемся от пропусков
        df[columns_to_fill] = df[columns_to_fill].fillna(value='-')

        # избавляемся от пропусков нулями
        df[columns_to_fill_dight] = df[columns_to_fill_dight].fillna(0)
```

```
In [109... # проверим результат
print(columns_to_fill)
display(df[columns_to_fill].isna().sum())

print('-'*100)

print(columns_to_fill_dight)
display(df[columns_to_fill_dight].isna().sum())

['user_id', 'user_work', 'user_tags', 'user_location', 'user_common_info', 'url', 'post_url']
user_id      0
user_work    0
user_tags    0
user_location 0
user_common_info 0
url          0
post_url     0
dtype: int64
-----
['user_experience', 'user_viewers']
user_experience    0
user_viewers      0
dtype: int64
```

```
In [110... # объединим пользовательские реакции в одну
df['reaction'] = df.likes + df.comments + df.reposts
```

```
In [111... # проверим содержимое поля числа фоловеров
df.user_viewers.unique()
```

```
Out[111]: array(['2\xa0391', '340', '540', '411', '40', '581', '66', '1,231',  
      '4,569', '2,840', '839', '3,547', '534', '103', '60', '478', '415',  
      '1,328', '1,344', '1,732', '116', '6,961', '1,211', '624', '6,750',  
      '1,738', '2,091', '1,378', '500+ connections', '253', '652', '172',  
      '884', '189', '1,678', '1,183', '456', '1,023', '119', '1,166',  
      '634', '1,663', '16', '155', '300', '1,272', '3,716', '1,312',  
      '660', '933', '789', '2,153', '2,875', '3,572', '1,076', '11,009',  
      '667', '83', '928', '6,197', '596', '239', '575', '8,817', '274',  
      '1,074', '772', '109', '13,844', '12,066', '1,230', '725', '460',  
      '2,067', '6,747', '370', '153', '477', '8,203', '1,538', '852',  
      '476', '1,053', '802', '1,160', '215', '7,371', '1,159', '781',  
      '3,327', '272', '1,296', '843', '2,856', '393 connections', '771',  
      '554', '216', '85', '1\xa0705', '500+ контактов', '2\xa0478',  
      '280', '944', '2\xa0872', '436', '287', '1\xa0035', '5\xa0492',  
      '10\xa0918', '275', '4\xa0609', '930', '1\xa0623', '1\xa0495',  
      '739', '675', '247', '198', '1\xa0195', '7\xa0559', '1\xa0453',  
      '381', '692', '2\xa0073', '1\xa0649', '1\xa0820', '1\xa0001',  
      '1,733', '1,977', '297', '905', '2,273', '1,170', '135', '4,409',  
      '1,130', '3,165', '642', '4,949', '746', '3,598', '1,916', '1,065',  
      '2,443', '703', '2,831', '2,934', '1,179', '604', '10,401', '796',  
      '313', '481', '8,893', '4,564', '2,003', '732', '29,597', '3,830',  
      '1,981', '2,952', '4,482', '5,508', '882', '424', '1,686', '2,301',  
      '3\xa0691', '1,488', '550', '255', '3,115', '778', '5,300', '0',  
      '112', '298 connections', '3,768', '12', '1\xa0613', '674',  
      '9\xa0885', '2\xa0667', '2\xa0366', '2\xa0797', '4\xa0439', '515',  
      '1\xa0063', '414', '372', '4\xa0169', '1\xa0779', '1\xa0167',  
      '349', '493 контакта', '15\xa0024', '5\xa0815', '12\xa0836', 0],  
      dtype=object)
```

```
In [112... # оставим только числа  
df.user_viewers = df.user_viewers.str.replace(r'\D', '', regex=True).fillna(0)  
  
# изменим тип данных  
df.user_viewers = df.user_viewers.astype('int')
```

```
In [113... # проверим содержимое поля числа контактов  
df.user_contacts.unique()
```

```
Out[113]: array(['500+', '338', '405', '33', '53', '92', '58', '467', '402', '91',  
      '0', '233', '143', '184', '452', '112', '9', '154', '297', '48',  
      '226', '257', '106', '451', '491', '369', '148', '470', '349',  
      '213', '270', '198', '80', '245', '433', '209', '236', '193',  
      '345', '244', '124', '264', '309', '460', '419', '250', '96', '10',  
      '396', '372', '305', nan], dtype=object)
```

```
In [114... # оставим только числа  
df.user_contacts = df.user_contacts.str.replace('[\D]', '', regex=True).fillna(0)  
  
# изменим тип данных  
df.user_contacts = df.user_contacts.astype('int')
```

```
In [115... df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2966 entries, 0 to 10809
Data columns (total 20 columns):
#   Column                Non-Null Count  Dtype
---  -
0   user_id                2966 non-null   object
1   post                   2966 non-null   object
2   likes                  2966 non-null   int32
3   comments               2966 non-null   int32
4   reposts                2966 non-null   int32
5   user_name              2966 non-null   object
6   user_head              2966 non-null   object
7   user_work              2966 non-null   object
8   user_position          2237 non-null   object
9   user_tags              2966 non-null   object
10  user_location           2966 non-null   object
11  user_viewers            2966 non-null   int32
12  user_contacts           2966 non-null   int32
13  user_common_info        2966 non-null   object
14  url                     2966 non-null   object
15  post_url                2966 non-null   object
16  user_experience          2966 non-null   float64
17  hashtags                2966 non-null   object
18  post_lemmatized         2966 non-null   object
19  reaction                2966 non-null   int32
dtypes: float64(1), int32(6), object(13)
memory usage: 417.1+ KB
```

Видимые проблемы устранены. Мы избавились от пропусков и количественные данные преобразовали в тип *int*.

Выборка постов

Оценим размеры постов в количестве символов и количестве слов.

In [119...

```
# подсчет числа символов
def count_chars(text):
    return(len(text))

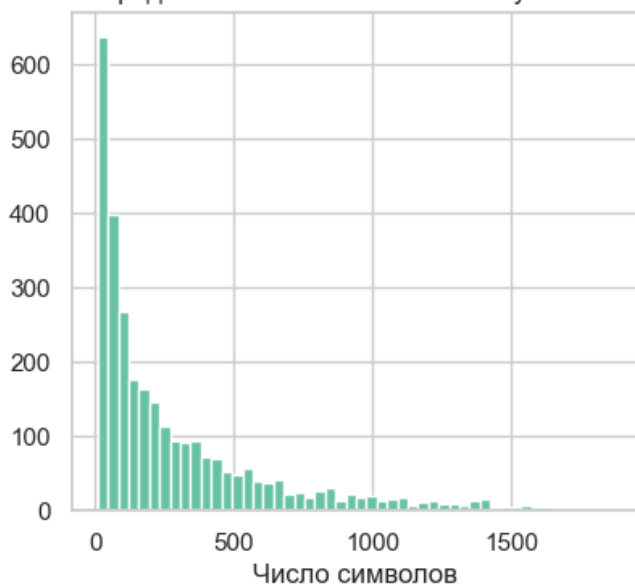
# подсчет числа слов
def count_words(text):
    return(len(text.split()))
```

In [120...

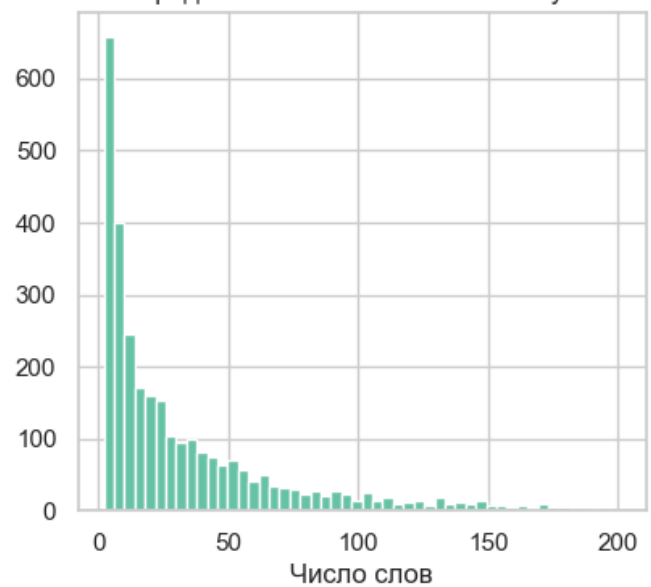
```
# посчитаем статистику и построим графики
df.loc[:, 'num_chars'] = df.post_lemmatized.apply(count_chars)
df.loc[:, 'num_words'] = df.post_lemmatized.apply(count_words)

plt.figure(figsize=(10, 4))
plt.subplot(1, 2, 1)
df['num_chars'].hist(bins=50)
plt.title('Распределение постов по количеству символов')
plt.xlabel('Число символов')
plt.subplot(1, 2, 2)
df['num_words'].hist(bins=50)
plt.title('Распределение постов по количеству слов')
plt.xlabel('Число слов')
plt.show()
```


Распределение постов по количеству символов



Распределение постов по количеству слов



In [121... *# характеристики постов по символам*
df.num_chars.describe()

Out[121]:

count	2966.000000
mean	293.454821
std	339.486465
min	9.000000
25%	56.000000
50%	158.000000
75%	402.750000
max	1870.000000

Name: num_chars, dtype: float64

In [122... *# характеристики постов по словам*
df.num_words.describe()

Out[122]:

count	2966.000000
mean	32.337829
std	37.123691
min	2.000000
25%	6.000000
50%	18.000000
75%	45.000000
max	201.000000

Name: num_words, dtype: float64

Большая часть постов короткие. Медианный размер поста 355 символов 39 слов. Есть смысл отбросить совсем короткие посты исключив их из анализа.

Оценим потери датасета, если отбросим посты короче 90 символов или 9 слов.

In [124... *# ограничения по количеству символов и слов*
min_chars = 90
min_words = 9

chars_filter = df.num_chars < min_chars
words_filter = df.num_words < min_words

In [125... *# число записей, попадающих под ограничения*
len(df[chars_filter | words_filter])

Out[125]: 1087

In [126... *# оценим содержание мелких текстов*
df.query('num_chars < @min_chars and num_words < @min_words').post_lemmatized.head()

```
Out[126]: 0    подкаст мираж доступный соцсеть вконтакте разобраться прикрутить
4          подкаст мираж эпизодошибка невозвратный затрата
5          эпизод эффект икеа платформа
7          эпизод подкаст теория рациональный выбор мешать рациональный
9          эпизод платформа
Name: post_lemmatized, dtype: object
```

```
In [127... # удаляем короткие посты
df = df.query('num_chars >= @min_chars and num_words >= @min_words')
```

```
In [128... # оценка датасета после фильтрации
df.info()

<class 'pandas.core.frame.DataFrame'>
Index: 1879 entries, 2 to 10803
Data columns (total 22 columns):
#   Column                Non-Null Count  Dtype
---  -
0   user_id               1879 non-null   object
1   post                  1879 non-null   object
2   likes                 1879 non-null   int32
3   comments              1879 non-null   int32
4   reposts               1879 non-null   int32
5   user_name             1879 non-null   object
6   user_head             1879 non-null   object
7   user_work             1879 non-null   object
8   user_position         1451 non-null   object
9   user_tags             1879 non-null   object
10  user_location         1879 non-null   object
11  user_viewers          1879 non-null   int32
12  user_contacts         1879 non-null   int32
13  user_common_info      1879 non-null   object
14  url                   1879 non-null   object
15  post_url              1879 non-null   object
16  user_experience       1879 non-null   float64
17  hashtags              1879 non-null   object
18  post_lemmatized       1879 non-null   object
19  reaction              1879 non-null   int32
20  num_chars             1879 non-null   int64
21  num_words             1879 non-null   int64
dtypes: float64(1), int32(6), int64(2), object(13)
memory usage: 293.6+ KB
```

Моделирование

Складываем все лемматизированные тексты в один список.

```
In [129... docs = df["post_lemmatized"].tolist()
```

```
In [130... # первые пять элементов
docs[:5]
```

```
Out[130]: ['честной народ искать векторный иллюстратор возможный длительный сотрудничество итог удалённ
ый уровень иллюстрация уметь рисовать',
'подкаст мираж платформа аудио инстаграм звук музыка картинка фильм формула любовь марк заха
ров',
'редкий случай правильный распространять призыв проявить активный позиция вмешаться политиче
ский государство подсказывать поправка творчество просветительство оказаться угроза конкретно
речь описать попов видео',
'искать команда дизайнер линейка продукт маркетинг райтер порядок интерфейсный текст английс
кий русский опыт способность глубоко разбираться технический деталь переводить человеческий у
словие вилка условие почта',
'команда найти откликнуться темп кек вкратце заняться самостоятельный собрать команда выстра
ивать процесс тратить курировать откликнуться герой продукт оплата чеканный монета откликатьс
я сюда']
```

Вывод:

- Мы выполнили предобработку полученных данных, удалили из текстов эмодзи и лишние символы, провели лемматизацию постов. Исключили посты без русских символов.
- Объединили таблицы постов и профилей пользователей и создали датасет. Устранили в датасете выявленные проблемы, избавились от пропусков и привели типы данных в соответствие.
- Выполнили поиск постов в соответствии с ключевыми словами для наибольшего охвата целевой аудитории.
- Исключили посты с небольшим числом символов и слов.

Наш датасет значительно сократился, но теперь наши данные готовы для анализа.

Векторизация текстов

Переведем тексты и слова, в числовое представление, т.е. выполним векторизацию. Для этого можно использовать метод Tf-idf.

```
In [131... # создаем модель векторизации
tfidf = TfidfVectorizer(min_df=20, max_df=0.9)
```

```
In [132... %%time
# обучим модель и получим векторное представление для каждого текста
x = tfidf.fit_transform(docs)
```

CPU times: total: 31.2 ms
Wall time: 72 ms

```
In [133... # размер полученной матрицы
x.shape
```

```
Out[133]: (1879, 843)
```

Составим словарь {id_токена: токен} - он пригодится нам позднее.

```
In [134... # список слов векторизатора
tf_feature_names = tfidf.get_feature_names_out()
```

```
In [135... # словарь
id2word = {i: token for i, token in enumerate(tf_feature_names)}
```

```
In [136... # примеры слов в словаре
id2word[0], id2word[1], id2word[2], id2word[200], id2word[420]
```

```
Out[136]: ('абсолютно', 'автоматизация', 'автоматизированный', 'интеллект', 'персонал')
```

3.2. LDA

Теперь можем запустить алгоритм LDA. Выполним подбор параметров. Качество модели будем оценивать с помощью метода `score()`. Посмотрим как меняется скор в зависимости от количества тем и числа итераций.

```
In [137... # параметры
n_topic_list = [10, 15, 20] # число тем
```

```
iter_list = [50, 100, 150] # число итераций
```

In [138...

```
%%time

# список для сохранения результатов
lda_results = []

# цикл подбора параметров
for n_topics, max_iter in product(n_topic_list, iter_list):

    # создаем модель
    lda = LatentDirichletAllocation(
        n_components=n_topics,
        max_iter=max_iter,
        n_jobs=-2,
        random_state=SEED
    )

    # обучаем модель на матрице векторизованных текстов
    lda.fit_transform(x)

    # метрика показывает приблизительное логарифмическое правдоподобие
    lda_score = lda.score(x)

    # сохраняем результаты
    lda_results.append([n_topics, max_iter, lda_score])
```

CPU times: total: 11.5 s
Wall time: 40.2 s

In [139...

```
pd.DataFrame(
    lda_results, columns=['n_topics', 'max_iter', 'lda_score']
).style.highlight_max(
    subset=['lda_score']
).set_caption('<h3>Сравнительная таблица качества моделирования</h3>')
```

Out[139]:

Сравнительная таблица качества моделирования

	n_topics	max_iter	lda_score
0	10	50	-61120.309714
1	10	100	-61099.060574
2	10	150	-61099.060500
3	15	50	-62803.297214
4	15	100	-62653.982571
5	15	150	-62629.169943
6	20	50	-63706.672917
7	20	100	-63706.672395
8	20	150	-63706.672395

Минимальное значение lda_score при n_topics = 10 и max_iter = 150.

Эксперимент показал, что с увеличением числа топиков, скор ухудшается, а увеличение числа итераций на скор влияет незначительно.

Получим модель с указанными параметрами.

In [140...

```
%%time

# число тем
```

```
n_topics = 10
n_iters = 150

# создаем модель
lda = LatentDirichletAllocation(
    n_components=n_topics,
    max_iter=n_iters,
    random_state=SEED
)

lda_topics = lda.fit_transform(x)
```

CPU times: total: 22.1 s

Wall time: 27.1 s

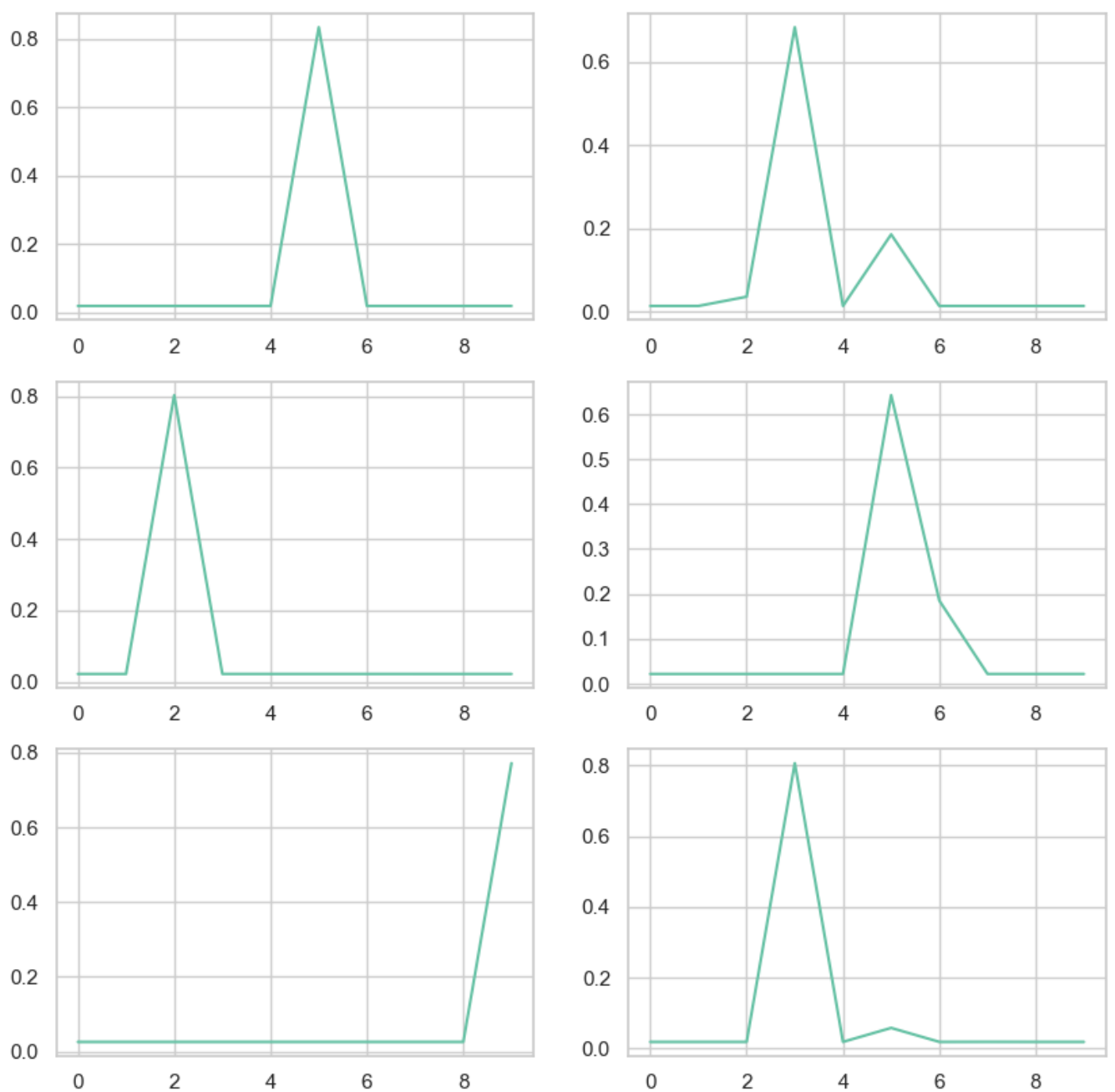
```
In [141... # размер полученной матрицы
lda_topics.shape
```

```
Out[141]: (1879, 10)
```

Номера строк матрицы соответствуют индексам текстов, а колонки выделенным темам. В каждой ячейке стоит вероятность того, что данный текст относится к данной теме.

Для наглядности, выберем несколько случайных записей и построим графики полученных вероятностей принадлежности текста к топикам.

```
In [142... plt.figure(figsize=(10,10))
for i in range(6):
    idx = np.random.randint(0, lda_topics.shape[0])
    plt.subplot(3, 2, i+1)
    plt.plot(lda_topics[idx])
```



Некоторые тексты могут принадлежать сразу нескольким темам.

Ключевые слова

Теперь извлечём ключевые слова для каждой из тем.

In [143...

```
# процедура строит график вероятностей ключевых слов по темам
def plot_top_words(model, feature_names, n_top_words, title):
    fig, axes = plt.subplots(2, 5, figsize=(30, 15), sharex=True)
    axes = axes.flatten()
    for topic_idx, topic in enumerate(model.components_):
        top_features_ind = topic.argsort()[::-n_top_words - 1 : -1]
        top_features = [feature_names[i] for i in top_features_ind]
        weights = topic[top_features_ind]

        ax = axes[topic_idx]
        ax.barh(top_features, weights, height=0.7)
        ax.set_title(f"Тема {topic_idx}", fontdict={"fontsize": 30})
        ax.invert_yaxis()
        ax.tick_params(axis="both", which="major", labelsize=20)
        for i in "top right left".split():
            ax.spines[i].set_visible(False)
        fig.suptitle(title, fontsize=40)
```

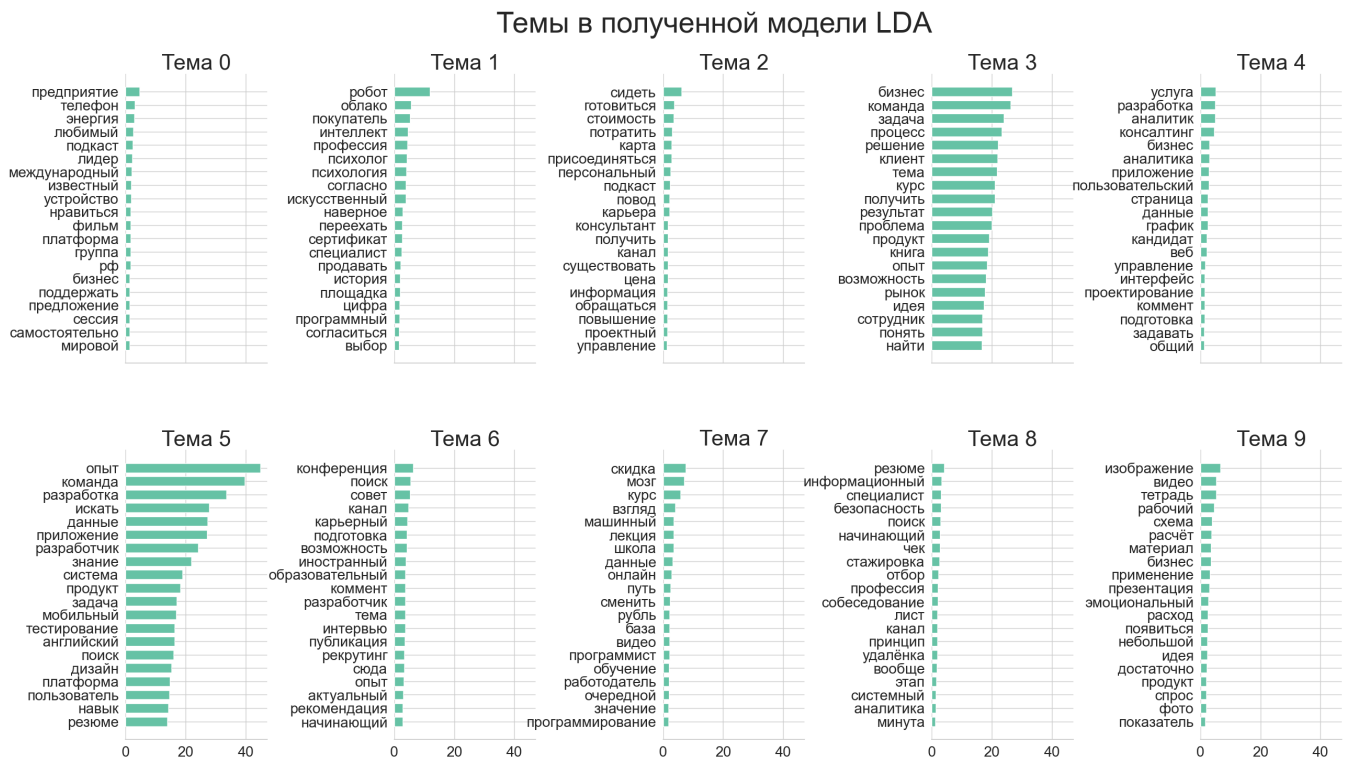
```
plt.subplots_adjust(top=0.90, bottom=0.05, wspace=0.90, hspace=0.3)
plt.show()
```

In [144...

```
# число ключевых слов в теме
```

```
n_top_words = 20
```

```
plot_top_words(
    lda, tf_feature_names, n_top_words, 'Темы в полученной модели LDA'
)
```



Темы 3 и 5 выделяются от остальных большими значениями вероятности для ключевых слов.

Интерпретация тем для LDA

Мы получили ключевые слова для каждой из тем и можно даже уловить смысл набора слов, но сформулировать тему более конкретно все равно затруднительно. Попробуем ключевые слова передать в ChatGPT и попросим уточнить тему.

- Тема 0: Технологии и влияние в современном мировом бизнесе.
- Тема 1: Влияние искусственного интеллекта на профессии и принятие решений.
- Тема 2: Подготовка и развитие карьеры в сфере консалтинга и управления проектами.
- Тема 3: Бизнес и управление задачами для достижения результатов и решения проблем на рынке.
- Тема 4: Разработка и управление пользовательским интерфейсом в сфере консалтинга и бизнес-аналитики.
- Тема 5: Опыт и навыки в разработке мобильных приложений и продуктов, поиск работы и составление резюме в этой области.
- Тема 6: Карьерные возможности, подготовка и советы для разработчиков и начинающих специалистов в области IT.
- Тема 7: Обучение программированию, онлайн-курсы и значение машинного обучения в современном мире.
- Тема 8: Поиск работы, отбор и собеседование для начинающих специалистов в области информационной безопасности.

- Тема 9: Применение изображений и видео в бизнесе, создание презентаций и эмоциональное воздействие на потребителя.

Типичные статьи

In [146...

```
for i in range(n_topics):  
    doc_id = np.argmax(lda_topics[:, i])  
    print("Тема ", i)  
    print(df.iloc[doc_id]["post"])  
    print("\n")
```


Тема 0
Вышел со мной в гостях В этом подкасте мы затронули множество интересных тем биоинформатику как стать и кому не стоит становиться инженером компьютерного зрения интересные проекты на старых местах работы танцы фотографию и много другого И конечно мы говорим о самом главном текущем месте работы в о том что делает вся компания и конкретно о нашем совершенно потрясающем сервисе распознавания объектов в фильмах и не только о Подкаст есть на почти всех платформах но привожу основные ссылки Яндекс Канал подкаста в Телеграме VK

Тема 1

Как выбрать куда релоцироваться Релокация сейчас горячая тема и вопрос о том с чего начать когда решаешь переехать самый популярный на моих консультациях Открытием на этих встречах для меня стало то насколько люди в гонке за оффером с визой готовы согласиться на любую страну и город лишь бы взяли Особенно текто никогда за границей не жил совсем забывают прикинуть насколько им будет в этой стране комфортно какой они смогут обеспечить себе уровень жизни Каким будет их круг общения хобби типичный день Мол само как то наверное разрулится Когда прошлой весной я получила два оффера в Испанию и Германию они были на удивление похожи по цифрам и обязательствам И выбор о том куда ехать я делала в первую очередь опираясь на город Но даже если вы ещё только начинаете искать работу за рубежом очень советую заранее представить как вам будет жить в другом городе обязательно использовать эти критерии при выборе В карточках рассказала простых шагов которые делала я Уже знаете куда хотите переехать фотка июнь когда после лет в Бразилии я переехала в Москву со всеми своими пожитками раскиданными в чемоданах Но это другая история всё ещё думаете минут были самыми полезными в моём рисёрче

Тема 2

прекрасная новость открываю свой телеграмм канал о проектном управлении Детали на моем премо сайте Присоединяйся сам приглашай своих коллег Сейчас тестовый режим идет формирование состава участников группы февраля официальное открытие Канала

Тема 3

Бесплатная консультация для тех кто ищет работу в но почему то не может дойти до оффера или хотя бы получить приглашение на собеседование Разработчики аналитики менеджеры дизайнеры и другие профессионалы любых уровней я вас всех приглашаю на звонок который изменит текущую ситуацию на рынке в вашу пользу Почему консультация бесплатная Дело в том что сейчас я создаю новый крутой продукт для ребят которые хотят построить карьеру в поэтому собираю информацию о тех проблемах с которыми они сталкиваются в процессе поиска работы а взамен даю много пользы которую вы сможете унести с собой совершенно бесплатно Что за продукт я разрабатываю Я готовлю практический онлайн курс который поможет вам найти работу мечты где задачи будут вас драйвить а команда помогать развиваться увеличить доход в раза чтобы вы смогли позволить себе больше наслаждаться жизнью и работать в кайф построить крутую карьеру составив план индивидуальной траектории развития на годы вперед Чтобы продукт получился реально классным мне необходимо пообщаться с вами и задать несколько вопросов Поэтому я хочу пригласить вас на звонок который займет примерно минут вашего времени но при этом сэкономит недели и месяцы поисков работы И не переживайте я не буду ничего продавать вам на этой встрече На звонке я поспрашиваю вас о том что у вас сейчас происходит какие цели вы перед собой сейчас ставите что вы уже делаете и почему у вас пока что не получается достигнуть результатов А затем я постараюсь вам прямо на месте подсказать несколько возможных решений вашей ситуации которые вы сможете испробовать сразу же после звонка Кто уже попробовал прийти на консультацию получили много полезных идей и ушли от меня мотивированными воплощать их в жизнь Так что я рекомендую воспользоваться возможностью пока она есть Пишите в личку Хочу на созвон и мы выберем удобное время для звонка чтобы разобрать вашу ситуацию также можно написать в или во ВКонтакте

Тема 4

С радостью сообщаю что я оказываю бизнес услуги в Просмотрите мою страницу услуг Бизнес аналитика Проектирование пользовательского интерфейса Разработка баз данных Разработка пользовательского ПО Разработка приложений Управление корпоративным контентом Управление информацией и Тестирование ПО

Тема 5

Всем привет Мы компания компания энтузиастов мира и фанатов разработки качественных программных решений Наша основная миссия создать с помощью своих программ интегри

рованную цифровую среду и повысить эффективность процессов наших клиентов Наша команда ищет на продукт ДБО ФЛ в банковский проект Формат работы по желанию удаленно офис гибридный если Москва Саратов Пенза коворкинг Занятость полная занятость ЗП до 300 тысяч рублей на руки Чем предстоит заниматься Техническое лидерство в реализации проекта ДБО физлиц Ревью программной архитектуры решения и инфраструктуры развертывания Участие в развитии архитектуры решения проработка интеграционных потоков Техническая координация внутренних и внешних команд Разбор инцидентов и методик недопущения Выработка решений по мониторингу и обеспечения отказоустойчивости планомерное увеличение доступности решения Координация всех ИТ служб банка для обеспечения бесперебойной работы и решения инцидентов Принятие ключевых технических решений проекта Анализ функциональных и нефункциональных требований в контексте архитектуры системы и платформы Участие в постановке задач аналитикам и разработчикам Аудит принятых системными аналитиками решений выбор оптимального способа реализации бизнес требований в соответствии с принятыми подходами Наши ожидания от кандидата Понимание концепций и ограничений распределенных систем Опыт разработки высоконагруженных приложений на архитектуру в качестве ведущего разработчика системного архитектора или тим лида Опыт управления командой разработки и Опыт проектирования с нуля или развития микросервисной платформы плюсом перевод с монолита Хорошее знание шаблонов проектирования и интеграции Экспертные знания языка программирования Опыт разработки архитектурной документации компонентная функциональная развертывания и т.д. Знания платформы Опыт работы с СУБД и построенное кластера Опыт работы с каким либо из списка Уверенное знание методологий и принципов разработки ПО Компания предлагает вам Рабочую технику при необходимости ноутбук монитор и т.д. ДМС или спорт после испытательного срока Оплачиваемые профильные внешние курсы а также доступ к внутренним учебным программам Высококвалифицированная команда гибкие методики разработки Возможности профессионального роста и развития Буду рада ответить на все вопросы и рассказать про детали

Тема 6

Продолжаем апгрейтить нашумевший Гайд для Джунов в от Мне дико понравилась инициатива и захотелось дополнить блок по развитию софт скиллов Этот гайд просто кладёшь супер полезной инфы всё что нужно знать джуну который хочет быстро и эффективно прокачаться в Смотрите сами какие там есть разделы По мощь Сайты для поиска работы ТГ каналы для поиска вакансий Подготовка к интервью в иностранную компанию Полезные советы ам Резюме и сопроводительное письмо Консультации и менторство для ов Бесплатные курсы Английский язык Самостоятельное и Подготовка к и Краткое руководство для начинающего в зучение Викторина Игнатенко Анастасия Гусева Карьерный консультант Нат алья Везломцева Буду благодарна за репост

Тема 7

Сколько на самом деле зарабатывают программисты на старте карьеры А главное как быстрее повысить свою зарплату на а то и за год Смотри видео до конца и узнаешь Кстати для тех кто только начинает свой путь в или хочет сменить направление ловите курсы от онлайн школы со скидкой до А кто хочет попробовать прежде чем покупать ловите бесплатный доступ на 3 дней к онлайн курсам и интенсивам

Тема 8

Неудобные вопросы на собеседованиях Я тоже выбираю компанию а не только о компания меня Это прям главный принцип за который я топлю на своих консультациях больше инфо тут по поиску работы И лучший км к способ этого принципа придерживаться задавать правильные в меру неудобные вопросы тем кто вас собеседует В карусели смотрите вопросы чтобы ещё на этапе отбора понять а вам то в ообщем надо в эту компанию Узнать ответы на эти вопросы можно в моём телеграм канале тут

Тема 9

Как протестировать Рассказываю о том как протестировать первую версию продукта чтобы подтвердить спрос на бизнес идею А также делюсь что полезно проанализировать после теста чтобы оптимизировать дальнейшие расходы и бизнес процессы До полнительные материалы рабочая тетрадь с формулами для расчетов и полная схема из видео

Сохраним в датафрейм номер наиболее вероятной темы для каждого поста.

In [147...

```
# значения наиболее вероятных топиков
df['lda_topic'] = np.argmax(lda_topics, axis=1)
```

Вывод:

Мы выполнили тематическое моделирование с помощью алгоритма Латентного размещения Дирихле (LDA). Провели эксперимент и выяснили, что с увеличением числа топиков, скор ухудшается, а увеличение числа итераций на скор влияет незначительно.

Практически все тексты найденных типичных статей соответствуют темам топиков и ключевым словам. Но вероятности ключевых слов по темам распределены не равномерно.

3.3. NMF

Неотрицательная матричная факторизация (NMF).

In [148...

```
%%time

# число тем
n_topics = 10
n_iters = 300

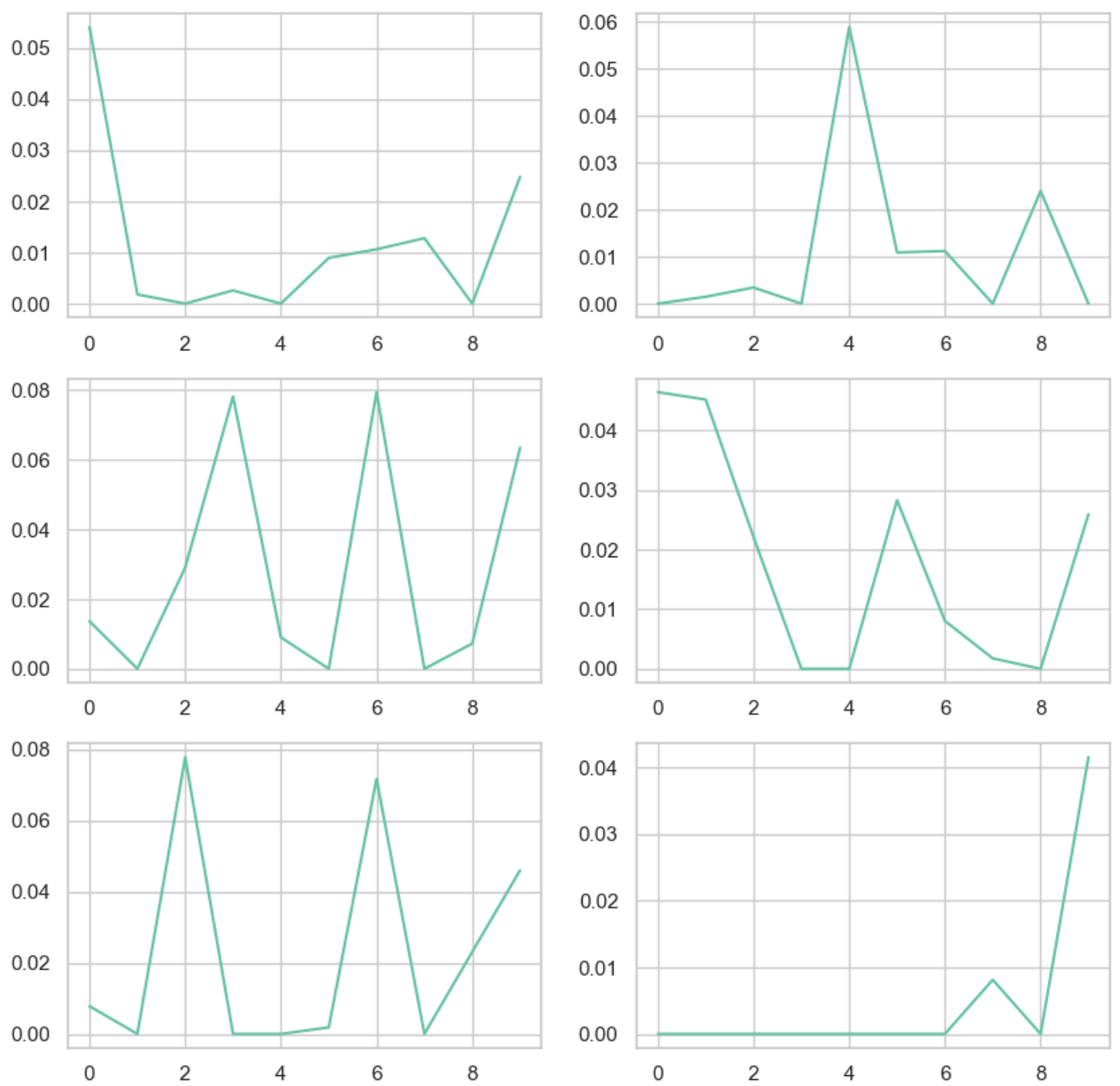
# создаем модель
nmf = NMF(
    n_components=n_topics,
    max_iter=n_iters,
    random_state=SEED
)

# обучаемся
nmf_topics = nmf.fit_transform(x)
```

CPU times: total: 31.2 ms
Wall time: 99.3 ms

In [149...

```
# графики полученных вероятностей принадлежности текста к топикам
plt.figure(figsize=(10,10))
for i in range(6):
    idx = np.random.randint(0, nmf_topics.shape[0])
    plt.subplot(3, 2, i+1)
    plt.plot(nmf_topics[idx])
```



Как и в случае с LDA, публикации могут принадлежать одновременно нескольким темам.

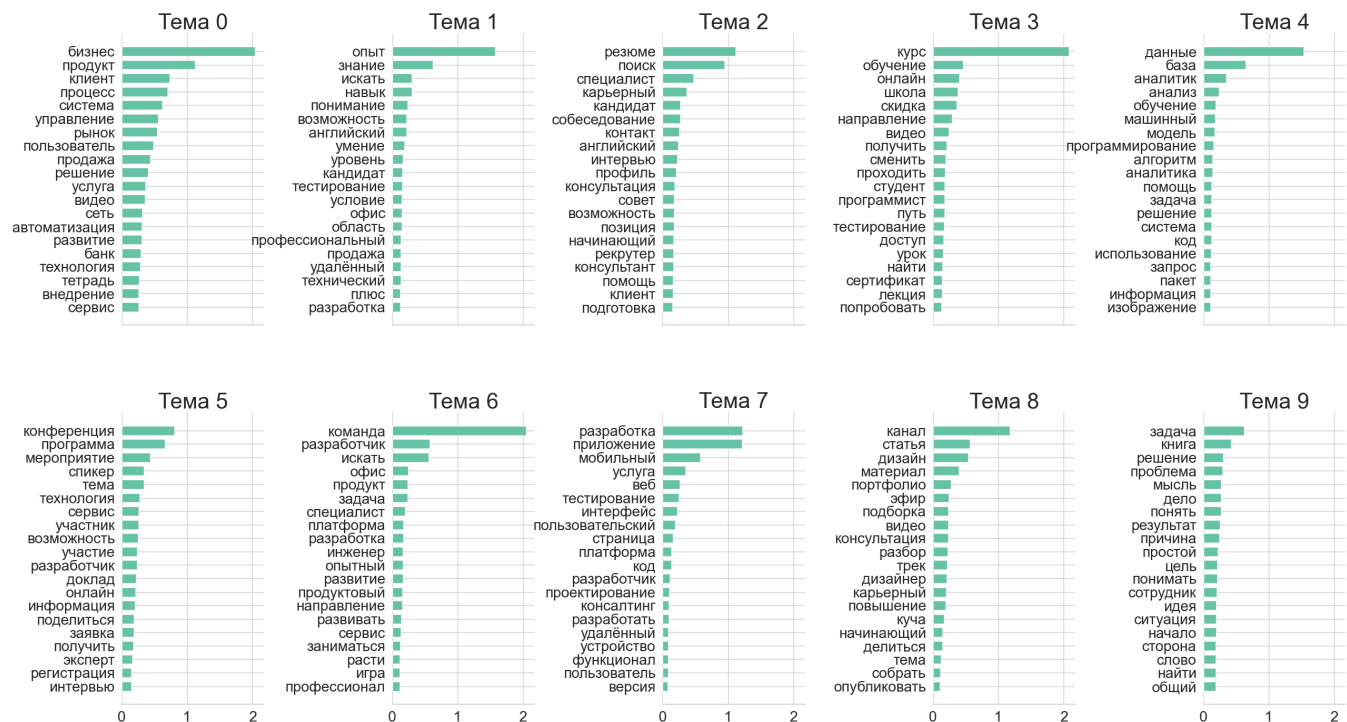
Ключевые слова

In [150...

```
# число ключевых слов в теме
n_top_words = 20

plot_top_words(
    nmf, tf_feature_names, n_top_words, 'Темы в полученной модели NMF'
)
```

Темы в полученной модели NMF



Интерпретация тем для NMF

- Тема 0: Управление бизнесом, разработка продукта и услуги для клиентов, автоматизация процессов и развитие на рынке.
- Тема 1: Поиск работы, развитие профессиональных навыков и знаний, уровень английского языка, тестирование и удаленная работа в технической области.
- Тема 2: Поиск работы, составление резюме, подготовка к собеседованию, карьерное развитие и консультации специалистов.
- Тема 3: Обучение и онлайн-курсы, получение сертификатов, выбор направления и путь становления программистом или тестировщиком.
- Тема 4: Анализ данных и машинное обучение: использование алгоритмов и моделей, программирование, решение задач и обработка информации.
- Тема 5: Участие в технологических конференциях и мероприятиях: программа, спикеры, онлайн-регистрация, получение информации и возможность поделиться экспертизой.
- Развитие и работа в команде разработчиков, поиск специалистов, разработка продукта, рост и развитие профессиональных навыков.
- Тема 7: Разработка мобильных и веб-приложений, тестирование пользовательского интерфейса, консалтинг, удаленная работа и функциональные возможности.
- Тема 8: Деление опытом и материалами в каналах и статьях, разработка дизайна и портфолио, консультации и разборы задач для дизайнеров.
- Тема 9: Анализ задач, поиск решений и понимание проблем, цели и идеи в рабочей ситуации, взаимодействие и общее понимание сотрудников.

Типичные статьи

In [153...

```
# оценим типичные статьи для каждой из тем
for i in range(n_topics):
    doc_id = np.argmax(nmf_topics[:, i])
    print("Тема ", i)
    print(df.iloc[doc_id]["post"])
    print("\n")
```

Тема 0

Конечный исполнитель и бизнес нужна ли волку телега Заметил тенденцию за знакомыми руководителями за собой в частности злоупотребление в изоляции подчиненных от бизнеса Полная защита от внешних негативных факторов обеспечение эмоционального штиля в команде фокус только на технологии и узкую специализацию избавление от общения со стейкхолдерами и т д По книжкам образцовый руководитель Но боюсь руководитель диод не всегда полезен Полная версия на моем канале

В больших компаниях разработчики имеют узкую специализацию и зону ответственности что может затруднять их понимание бизнес процессов и целей компании Однако это понимание является важным фактором для успешного выполнения задач и достижения общих целей При полной изоляции производства начальство может показаться оторванным от реальности и дающим сомнительные указания Часто рядовые сотрудники производства испытывают негативное отношение или даже отвращение к бизнесу хотя бы на интуитивном уровне Примеры негатива Заполнение ежедневных таймшитов Их вообще хоть кто то смотрит Убиваем большой модуль но мы же писали его месяц Нужно писать новый функционал так на старых еще не все паттерны проектирования перепробовал да и оптимизации еще оптимизировать Старый стек не модно куда бизнес смотрит Сейчас бы все на переписать Погрузив команды в детали бизнеса появится понимание тех или иных действий и требований В таком случае сотрудник если не проявит лояльность то как минимум сократится уровень напряжения Заодно повысится ориентированность на бизнес точность и правильность выполнения задач Некоторые шаги по решению проблемы осведомленности Закрывать базовые пробелы по процессам стратегии механизмам бизнеса на регулярных встречах о о Приглашать на фасилитационные встречи по стратегии открытым проблемам и тактике отдела или центра компании Способствовать межфункциональному сотрудничеству между другими отделами или центрами Предоставлять производству доступ к соответствующим бизнес данным и метрикам чтобы они могли видеть как их работа способствует успеху компании Для понимания деятельности компании важно передать информацию о ее структуре бизнес модели сегментации рынка стратегии и ключевых показателях производительности а также обратить внимание на взаимодействие бизнес процессов таких как маркетинг продажи и техническая поддержка В случае заказной разработки добавляется плюс знания о том как работает воронка продаж какие денежные потоки существуют как играют конкурсы какие есть роли и т д Идеально когда подчиненный имеет Т образные знания с пониманием тонкостей работы бизнеса Но в этом случае он скорее всего уже в менеджменте либо близок к этому Понимание бизнес процессов может быть драйвером по изменениям и улучшениям продукта вместо оппозиции с бизнесом Как считаете важно погружать производство в тонкости бизнеса

Тема 1

Всем привет Наша команда ищет инженера в крупный финтех проект инвестиции Формат работы по желанию удаленно офис гибридный если Москва Саратов Пенза Возможна работа вне РФ из некоторых стран Занятость полная занятость ЗП тыс р на руки Какой опыт требуется Понимание основных принципов и подходов методологии Опыт работы с Опыт работы и реализации решений для сборки и деплоя Опыт работы с системами Опыт настройки и поддержания систем мониторинга логирования и визуализации стек стек Понимание принципов работы сетевых протоколов Опыт написания запросов на как плюс Опыт написания автоматизаций на Опыт работы с Опыт взаимодействия с другим и командами разработки локализации и устранения проблем Будет плюсом но не обязательно Опыт понимания принципов работы высоконагруженных высокодоступных систем Опыт работы с Опыт работы с системами виртуализации Опыт работы с системами Компания предлагает вам Рабочую технику при необходимости ноутбук монитор и т д ДМС или спорт после испытательного срока Оплачиваемые профильные внешние курсы а также доступ к внутренним учебным программам Возможности профессионального роста и развития Лучше сразу приходите в Буду рада ответить на все вопросы и рассказать про детали

Тема 2

ПОЛЕЗНЫЙ СПИСОК ДЛЯ РЕКРУТЕРОВ И ТЕХ КТО В ПОИСКЕ РАБОТЫ Полный список телеграмм каналов для поиска вакансий и сотрудников Польша вакансии и резюме вакансии и резюме вакансии и запросы на поиск работы в в Польше живое общение вакансии курсы стажировки и митапы вакансии для и вакансии для и вакансии для начинающих специалистов в и вакансии и резюме специалистов в Польше аналитика и вакансии для вакансии с релокацией и резюме вакансии для начинающих специалист ов в СНГ вакансии и резюме вакансии с релокацией цией и удалёнкой вакансии и резюме резюме кандидатов в поиске резюме соискателей вакансии для начинающих специалистов на удаленку вакансии с релокацией стажировка и вакансии для молодых специалистов вакансии с релокацией

Тема 3

Очень сильно разочарован в политике. Если репостнитето поможет не только мне и всем инструкторам, которые создают там курсы. Ситуация следующая: забирает себе около 10% от дохода при прямой продаже курса через площадку или если принять их рекламное предложение, все курсы за 10% или 20%. В итоге от каждой продажи у автора остается от 10 до 20 долларов, так как средняя цена курсов на платформе 200-300\$. Там еще и налоги, так что еще меньше. Как можно изменить ситуацию? Использовать купон от инструктора. Если вы приобретаете курс с ним, то инструктор получает 10% от продажи. До недавних пор срабатывало, точно можно было указать информацию о купоне в описании курса, чтобы привлечь людей покупать через него. Там более он всегда предлагал скидку. Сейчас платформа стала массово удалять такие упоминания, ссылаясь на политику, в которой на странице курса НИГДЕ НЕЛЬЗЯ УПОМИНАТЬ ПРО КУПОН. В лекциях тоже нельзя. Купон можно использовать только в рекламных письмах для УЖЕ СУЩЕСТВУЮЩИХ СТУДЕНТОВ, которые когда-то покупали другие курсы или в БОНУСНОЙ ЛЕКЦИИ ПОСЛЕ ОКОНЧАНИЯ ДРУГОГО КУРСА. Очень кривой механизм, который никак не дает инструктору возможность сделать это до продажи целевого курса. У меня сейчас вообще сняли с публикации лекцию, где нет ни о одной ссылке на купоны, только на бесплатные материалы для изучения. По крайней мере я не нашел запретов на это в политиках. Буду разбираться. Это я к чему? Теперь актуальный купон на все мои курсы будет на моем сайте, а также на отдельных страницах для каждого курса. Буду надеяться, что все кто захочет их купить, будут их использовать. К тому же без купона цена будет намного выше. Например, основной курс будет за 200\$ вместо 100\$. Если вы следите за каким-то интересным автором, то обязательно уточните у него наличие купона. Возможно, так вы сможете ему заработать лишнюю копейку за труд, который он вложил и вкладывает много времени и сил. Я например за последние 3 месяца полностью обновил материалы основного курса и потратил на это много сил. Получать после такого марш-броска несколько долларов не в моих планах. Буду смотреть за аналитикой после такого нововведения. Может быть вообще покину площадку, благо есть хороший аналог лично для меня на платформе, где размещены те же самые курсы. Надеюсь на поддержку.

Тема 4

Любой дата-ориентированный проект начинается с базы данных, и если данные сравнимы с кровью, то база данных — сердце. Поэтому среди прочих услуг мы предлагаем разработку автоматизированных систем сбора и хранения данных. В процессе разработки мы определим структуру базы данных, подберем наиболее оптимальную технологию для Вашего бизнеса, развернем базу данных на сервере, решим вопрос настройки автоматизированного сбора данных, предоставим программный интерфейс для удобного взаимодействия с базой данных, разработаем интерфейс интерактивного графического анализа данных. Своевременный сбор данных позволит повысить эффективность Вашего бизнеса и вести глубокую аналитику в будущем.

Тема 5

Вторая конференция на Неделе мартов в СПб. Совсем скоро состоится ключевое событие в мире сообществ — конференция, которую делают разработчики для разработчиков. Хедлайнеры конференции — это те, кто ищет новые решения и ищет пути для развития. Не упустите возможность узнать мировые тренды программирования от ведущих специалистов BARC Group и других. Регистрация и программа конференции — свернуть.

Тема 6

Вакансий от БанкЦентрКредит. Разработчик в команду. Поддержка в команде. Разработчик в команду. Валютный Контроль. Тестирующий со знаниями банковских продуктов в команду. Валютный Контроль. Специалист Информационной Безопасности. За описаниями вакансий пишите мне в

Тема 7

С радостью сообщаю, что я оказываю бизнес-услуги в сфере веб-разработки, разработки облачных приложений, разработки мобильных приложений, разработки приложений и ИТ-консалтинг. Просмотрите мою страницу услуг.

Тема 8

В телеграм-каналах теперь можно делиться папками, и тут ребята из разных телеграм-каналов про

и дизайн собрались в одной такой папке я удивился когда оказалось что я подписан почти на все каналы из папки Очень полезные и интересные каналы где можно найти море информации Если вы начинающий дизайнер и вам все интересно то вам точно нужна эта папка Вот ссылочка на папку Скорее забираем себе и прокачиваем навыки Скорее переходим в мой телеграмм канал Там куча полезного в канале материалов для повышения ваших скиллов Розыгрыш консультаций по карьерному треку и набору портфолио Прямые эфиры с дизайном Полезные статьи

Тема 9

Только ленивый не писал про это за явление до конца не понятно поэтому ниже список базовых советов как пережить любые менеджерам Тимлидам и руководителям проектов и не только Больше общаться Если есть какое то неудобство на проекте или во взаимоотношениях с командой нужно обратиться к руководителю или аккаунту за советом Ну и всё таки важно выстраивать дружелюбный климат вокруг то есть не надо замыкаться чтобы самому себя не закапывать Будьте открытыми например в чём то просите помощи если не успеваете или поделитесь что какие то конкретные задачи вам нравятся больше Возможно это кому то сложно но надо стараться хотя бы начинать заниматься этим Когда присутствует взаимная поддержка выгорание случается реже или как минимум не в такой интенсивной форме Планировать и приоритизировать Внимательно относиться к распределению задач чтобы не загонять себя и не уходить в перегрузы Оценивание своего времени то чему надо постоянно учиться От проекта к проекту понимать почему в этой ситуации я недооценил сколько сложных задач подряд я могу решать без снижения эффективности Может быть такое что проект не требует напряжённой работы но вы овертаймите из за того что выделили на задачу часа а не сделали её и за Если вы менеджер и понимаете что никак не укладываетесь или ваш сотрудник то делегируйте задачи распределите их Поможет декомпозиция разбейте задачи на более мелкие раздайте их остальным участникам процесса и контролируйте выполнение Любить своё дело Если получаете удовольствие от работы вам это нравится это ваше хобби вы можете заниматься этим иногда почти круглосуточно Быть в тонусе Главное в этот момент не расслабляться чтобы принимать правильные решения которые позволят решить ситуацию очень помогает любая физическая активность Относиться с интересом Нужно попробовать воспринимать происходящее как уровень в игре который никак не поддаётся То есть смотреть на всё не только с позиции надо но и с желанием покорить эту гонку Включить соревновательный дух Принять вызов Не ограничиваться работой Наша работа часть жизни но только лишь часть Надо всегда понимать это и даже если на проекте что то получилось не так и вы по уши в огне вы способны изменить ситуацию у вас есть ваша семья друзья личная жизнь хобби Если обстоятельства складываются так что всё горит просто обратите внимание что это не смертельно и всё Зачастую это помогает решить психологические проблемы и высвобождает свежие мысли Очень важно быть эмпатичным и с уважением относиться к переживаниям других коллег заказчика подрядчика босса и т д Также важно профессионально объяснять причины любой ситуации и погружать в план действий например в изменения процесса контроля Тем самым вы показываете что обстоятельства не хаотичны и вы управляете ими работаете над конечным результатом А какие методы от используете Вы

In [154...

```
# значения наиболее вероятных топиков
df['nmf_topic'] = np.argmax(nmf_topics, axis=1)
```

Вывод:

Определенно есть соответствие между темами, ключевыми словами и текстами. Вероятности ключевых слов в темах распределены равномерно.

ТОП-10 тем постов целевой аудитории

Мы рассмотрели два алгоритма для моделирования тем. Оба алгоритма показали достаточно интерпретируемые результаты. Сделать однозначный выбор между ними достаточно сложно.

Проверим как распределились топики для разных алгоритмов в датасете.

In [155...

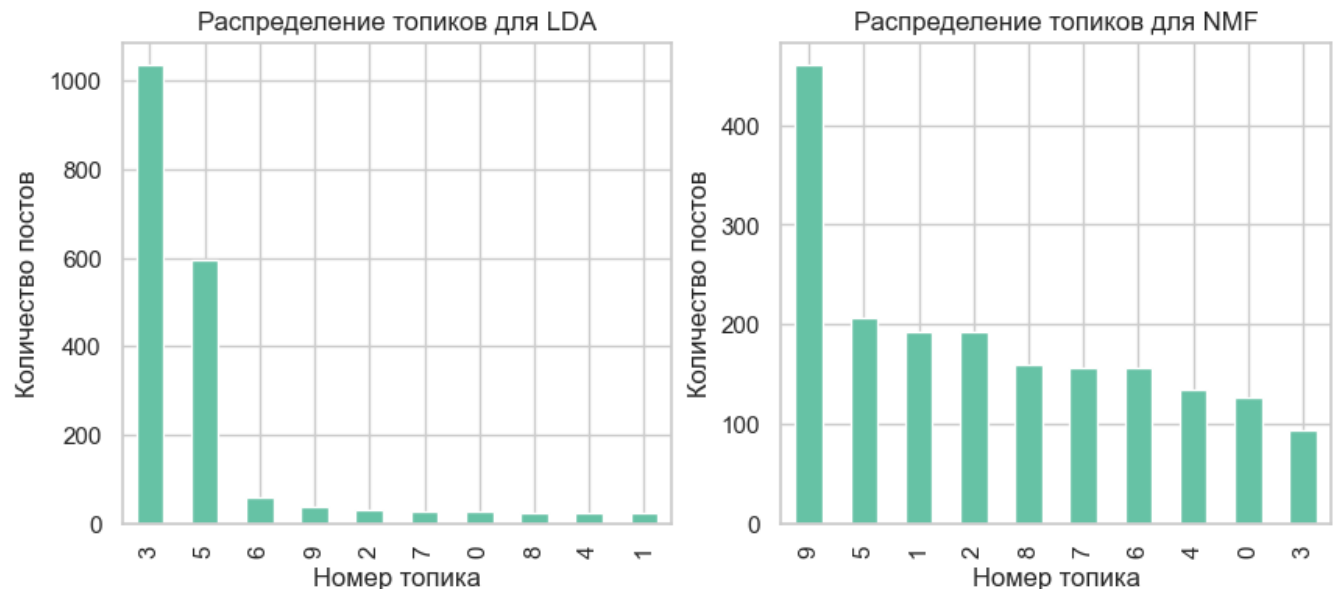
```
# распределение топиков для LDA
plt.figure(figsize=(10,4))
plt.subplot(1,2,1)
```



```

df.lda_topic.value_counts().plot(
    kind='bar', xlabel='Номер топики', ylabel='Количество постов',
    title='Распределение топики для LDA'
)
plt.subplot(1,2,2)
df.nmf_topic.value_counts().plot(
    kind='bar', xlabel='Номер топики', ylabel='Количество постов',
    title='Распределение топики для NMF'
);

```



Алгоритм LDA отдает предпочтение топику под номером 3 и 5. Это значит, что алгоритм хуже различает другие темы.

Алгоритм NMF выглядит предпочтительней. Поэтому в качестве ТОП-10 тем в направлении наставничества на основании наибольшего охвата, можно предложить темы на основе ключевых слов, полученных с помощью алгоритма NMF.

Но так как мы классифицировали всего 10 тем, то, пожалуй, стоит сократить ТОП до 5 позиций. В таком случае, можем отметить, что наибольшее число публикаций наблюдается для тем: 9, 5, 1, 2 и 8.

- Тема 0: Управление бизнесом, разработка продукта и услуги для клиентов, автоматизация процессов и развитие на рынке.
- **Тема 1: Поиск работы, развитие профессиональных навыков и знаний, уровень английского языка, тестирование и удаленная работа в технической области.**
- **Тема 2: Поиск работы, составление резюме, подготовка к собеседованию, карьерное развитие и консультации специалистов.**
- Тема 3: Обучение и онлайн-курсы, получение сертификатов, выбор направления и путь становления программистом или тестировщиком.
- Тема 4: Анализ данных и машинное обучение: использование алгоритмов и моделей, программирование, решение задач и обработка информации.
- **Тема 5: Участие в технологических конференциях и мероприятиях: программа, спикеры, онлайн-регистрация, получение информации и возможность поделиться экспертизой.**
- Развитие и работа в команде разработчиков, поиск специалистов, разработка продукта, рост и развитие профессиональных навыков.
- Тема 7: Разработка мобильных и веб-приложений, тестирование пользовательского интерфейса, консалтинг, удаленная работа и функциональные возможности.
- **Тема 8: Деление опытом и материалами в каналах и статьях, разработка дизайна и портфолио, консультации и разборы задач для дизайнеров.**

- **Тема 9: Анализ задач, поиск решений и понимание проблем, цели и идеи в рабочей ситуации, взаимодействие и общее понимание сотрудников.**

ТОП-10 тем, вызывающих наибольшую реакцию

Наш датасет содержит данные по разным реакциям пользователей на публикации: лайки, комментарии и репосты. Так же мы создали новый параметр - суммарная реакция.

Давайте посчитаем все типы реакций для каждой из тем.

In [156...]

```
# посчитаем суммарные реакции для топиков
df.pivot_table(
    index='nmf_topic', values=['likes', 'comments', 'reposts', 'reaction'],
    aggfunc='sum'
).style.background_gradient()
```

Out[156]:

	comments	likes	reaction	reposts
nmf_topic				
0	56	883	971	32
1	855	11985	14084	1244
2	1953	21468	26643	3222
3	339	5941	7087	807
4	273	2250	2628	105
5	345	3110	3629	174
6	330	3273	3912	309
7	656	3210	4182	316
8	218	5101	6016	697
9	2886	30108	34539	1545

В целом видна корреляция между разными типами реакций.

Из 10 тем, в качестве наиболее популярных и интересных можно отметить темы: 9, 2, 1, 3, 8.

- Тема 0: Управление бизнесом, разработка продукта и услуги для клиентов, автоматизация процессов и развитие на рынке.
- **Тема 1: Поиск работы, развитие профессиональных навыков и знаний, уровень английского языка, тестирование и удаленная работа в технической области.**
- **Тема 2: Поиск работы, составление резюме, подготовка к собеседованию, карьерное развитие и консультации специалистов.**
- **Тема 3: Обучение и онлайн-курсы, получение сертификатов, выбор направления и путь становления программистом или тестировщиком.**
- Тема 4: Анализ данных и машинное обучение: использование алгоритмов и моделей, программирование, решение задач и обработка информации.
- Тема 5: Участие в технологических конференциях и мероприятиях: программа, спикеры, онлайн-регистрация, получение информации и возможность поделиться экспертизой.
- Развитие и работа в команде разработчиков, поиск специалистов, разработка продукта, рост и развитие профессиональных навыков.
- Тема 7: Разработка мобильных и веб-приложений, тестирование пользовательского интерфейса, консалтинг, удаленная работа и функциональные возможности.

- **Тема 8: Деление опытом и материалами в каналах и статьях, разработка дизайна и портфолио, консультации и разборы задач для дизайнеров.**
- **Тема 9: Анализ задач, поиск решений и понимание проблем, цели и идеи в рабочей ситуации, взаимодействие и общее понимание сотрудников.**

Выводы:

- Т.к. мы получили всего 10 тем, ТОП пришлось сократить до 5.
- ТОП тематики постов целевой аудитории и ТОП тем вызывающих интерес, во многом совпадают. Но есть и различия, например по теме 5 есть публикации, но реакция на них ниже и наоборот, на тему 3 присутствует интерес, но публикаций недостаточно.

Выводы

Мы провели исследование для EdTech, сервиса онлайн образования. Для исследования собрали данные о пользователях и публикациях в социальной сети *Linkedin*. Тема исследования - наставничество и менторство. Для проведения исследования, собрали контент созданный целевой аудиторией социальной сети. В качестве контента использовали информацию из открытых профилей пользователей и публикуемые ими сообщения. Собранные данные были обработаны и создан датасет.

На полученном датасете мы провели анализ и тематическое моделирование. Моделирование выполнено на Latent Dirichlet Allocation (LDA) и Non-Negative Matrix Factorization (NMF). В результате анализа качества моделей, мы выбрали NMF. Нам удалось определить следующий ТОП тем в направлении наставничества на основании наибольшего охвата (в порядке убывания важности):

- Тема 9: Анализ задач, поиск решений и понимание проблем, цели и идеи в рабочей ситуации, взаимодействие и общее понимание сотрудников.
- Тема 5: Участие в технологических конференциях и мероприятиях: программа, спикеры, онлайн-регистрация, получение информации и возможность поделиться экспертизой.
- Тема 1: Поиск работы, развитие профессиональных навыков и знаний, уровень английского языка, тестирование и удаленная работа в технической области.
- Тема 2: Поиск работы, составление резюме, подготовка к собеседованию, карьерное развитие и консультации специалистов.
- Тема 8: Деление опытом и материалами в каналах и статьях, разработка дизайна и портфолио, консультации и разборы задач для дизайнеров.

и ТОП популярных тем по просмотрам и реакциям среди IT-специалистов, подходящих под описание целевой аудитории (в порядке убывания важности):

- Тема 9: Анализ задач, поиск решений и понимание проблем, цели и идеи в рабочей ситуации, взаимодействие и общее понимание сотрудников.
- Тема 2: Поиск работы, составление резюме, подготовка к собеседованию, карьерное развитие и консультации специалистов.
- Тема 1: Поиск работы, развитие профессиональных навыков и знаний, уровень английского языка, тестирование и удаленная работа в технической области.
- Тема 3: Обучение и онлайн-курсы, получение сертификатов, выбор направления и путь становления программистом или тестировщиком.
- Тема 8: Деление опытом и материалами в каналах и статьях, разработка дизайна и портфолио, консультации и разборы задач для дизайнеров.

Не секрет, что соцсеть LinkedIn у русскоязычных пользователей чаще всего используется для поиска работы зарубежом. Это подтверждается темами 1 и 2. По данным темам есть и публикации и реакции на них.

Темы 8 и 9 вполне можно отнести к менторству. Обе темы присутствуют в топе публикаций и в топе по реакциям пользователей.

Тема 5 находится в топе публикаций. Можно предположить, что к данному топику относятся публикации рекламного и информационного характера. По числу реакций аудитории рейтинг низкий.

Тема 3 относится непосредственно к онлайн образованию. Ретинг реакций у данной темы достаточно высок, а вот число публикаций не на высоте. Т.е. можно сделать вывод - есть спрос.

Данная информация может помочь сервису онлайн образования, понять какие темы на рынке представлены в достаточной мере, а какие не очень. Эта информация поможет эффективнее принимать бизнес-решения.

Что, можно улучшить в данном проекте:

Учитывая жесткие временные рамки проекта и технические сложности, связанные со сбором данных, мы не смогли ещё собрать датасет для более качественного исследования. В результате, общее количество смоделированных тем сократилось до десяти.

Для исправления ситуации, можно продолжить сбор данных. Это позволит расширить число тем и улучшить качество тематического моделирования. Так же не исчерпаны возможности по тестированию других алгоритмов машинного обучения.

In []: