

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 могут 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 [1]: 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 [2]: # 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 [3]: # 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 [4]: # 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 [5]: # 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 [6]: # 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 [7]: # 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 [8]: # 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 профилем
    """
    # 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 [9]: *# 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 [10]: # 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 [11]: # 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 [12]: # 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 [13]: # 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 [14]: # 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 [15]: # с какого профиля стартуем
# если ранее парсинг был прерван, продолжаем с того же места
start_idx = profiles.user_name.nunique()
start_idx
```

```
Out[15]: 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 [16]: # 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 [17]: 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 [18]: # оценим датафрейм с постами
posts.head(2)
```

Out[18]:

	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 [19]:

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

Out[19]:

	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 [20]:

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

Объединим датафреймы

In [21]:

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

In [22]:

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

In [23]:

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

In [24]:

```
# 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 [25]:

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

5970	ivan-mushavets	The long-awaited update. Only higher from here!#startup #IT #VC #project	1	0	0	Ivan Mushavets	Chief Business Development officer at Haiku Dev	SalAd Lab	Business Development Officer at Haiku Dev
------	----------------	--	---	---	---	----------------	---	-----------	---

6828	vladislav-popov-2021	Кожного разу кажу - дивіться більше референсів, особливо акцентуйте увагу на живих та працюючих продуктах. Рекомендую для мобільного дизайну заходити сюди. Поки що бібліотека	240	12	17	Vladislav Popov	Head of Design at Skeleton Crew	Skeleton Crew	Head of Design at Skeleton Crew
------	----------------------	--	-----	----	----	-----------------	---------------------------------	---------------	---------------------------------

user_id	post	likes	comments	reposts	user_name	user_head	user_work	user_pc
---------	------	-------	----------	---------	-----------	-----------	-----------	---------

безкоштовна —
<https://chamjo.design/>

In [26]: `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 [27]: # 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 [28]: dataset_from_team_3 = pd.read_csv(
        os.path.join(DATA_PATH, 'dataset_from_team_3.csv'), index_col=0
    )
```

```
In [29]: 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 [30]: # Проверим на наличие дубликатов
dataset_from_team_3.duplicated().sum()
```

```
Out[30]: 48
```

```
In [31]: # Устраняем их
dataset_from_team_3.drop_duplicates(inplace=True)
```

```
In [32]: # Проверка на пропущенные значения
dataset_from_team_3.isna().sum()
```

```
Out[32]: name             0
status           0
company          0
url              0
text             0
likes_cnt        7
reposts_cnt      0
comments_cnt     0
dtype: int64
```

```
In [33]: # 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 [34]: 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/dt9N6D7BСтатья 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-popov/	I'm happy to share that I'm starting a new position as Software Engineer at Tinkoff	0.0	0	

Датасет команды №4

```
In [35]: dataset_from_team_4 = pd.read_csv(
        os.path.join(DATA_PATH, 'dataset_from_team_4.csv'), delimiter=';'
    )
```

```
In [36]: 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 [37]: # Проверим на наличие дубликатов
dataset_from_team_4.duplicated().sum()
```

```
Out[37]: 84
```

```
In [38]: # Устраняем их
dataset_from_team_4.drop_duplicates(inplace=True)
```

```
In [39]: # Проверка на пропущенные значения
dataset_from_team_4.isna().sum()
```

```
Out[39]: url_user          0
        name            317
        job             317
        text_post        317
        react_per_user    317
        count_comments    317
        dtype: int64
```

```
In [40]: # Устраняем их
dataset_from_team_4.dropna(inplace=True)
```

```
In [41]: # 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 [42]: 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 [43]: 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 [44]: # Применение функции к замене колонки likes на кол-во лайков
dataset_from_team_4['likes'] = dataset_from_team_4['likes'].apply(
    calculate_median
)
```

```
In [45]: 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 [46]: dataset_from_team_8_1 = pd.read_csv(os.path.join(
        DATA_PATH, 'dataset_from_team_8_1.csv'
    ), delimiter=';', index_col=0)
```

```
In [47]: 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 [48]: # Проверим на наличие дубликатов
dataset_from_team_8_1.duplicated().sum()
```

```
Out[48]: 13
```

```
In [49]: # Устраняем их
dataset_from_team_8_1.drop_duplicates(inplace=True)
```

```
In [50]: # Проверка на пропущенные значения
dataset_from_team_8_1.isna().sum()
```

```
Out[50]: 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 [51]: # Устраняем их
dataset_from_team_8_1.dropna(inplace=True)
```

```
In [52]: # 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 [53]: 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'St 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.Overview'My		
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 [54]: dataset_from_team_8_2 = pd.read_csv(os.path.join(
    DATA_PATH, 'dataset_from_team_8_2.csv'
), delimiter=';', index_col=0)
```

```
In [55]: 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 [56]: # Проверим на наличие дубликатов
dataset_from_team_8_2.duplicated().sum()
```

```
Out[56]: 23
```

```
In [57]: # Устраняем их
dataset_from_team_8_2.drop_duplicates(inplace=True)
```

```
In [58]: # Проверка на пропущенные значения
dataset_from_team_8_2.isna().sum()
```

```
Out[58]: 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 [59]: # Устраняем их
dataset_from_team_8_2.dropna(inplace=True)
```

```
In [60]: # 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 [61]: 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'-'Индив 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 [62]: dataset_from_team_10 = pd.read_csv(
        os.path.join(DATA_PATH, 'dataset_from_team_10.csv')
    )
```

```
In [63]: 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 [64]: # Проверим на наличие дубликатов
dataset_from_team_10.duplicated().sum()
```

```
Out[64]: 3
```

```
In [65]: # Устраняем их
dataset_from_team_10.drop_duplicates(inplace=True)
```

```
In [66]: # Проверка на пропущенные значения
dataset_from_team_10.isna().sum()
```

```
Out[66]: 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 [67]: # Устраняем их
dataset_from_team_10.dropna(inplace=True)
```

```
In [68]: # Переименируем 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 [69]: dataset_from_team_10 = dataset_from_team_10.drop('posts_cnt', axis=1)
```

```
In [70]: 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	https://t.me/krupeshaaaTel 999 813 50 98Любозн 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 [71]: 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 [72]: # 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 [73]: # 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 [74]: # Объединяем датафреймы
df = pd.merge(df, dataset_from_team_8_1, how='outer')

print('Размер:', df.shape)
```

Размер: (10556, 16)

Часть 2

```
In [75]: # Объединяем датафреймы
df = pd.merge(df, dataset_from_team_8_2, how='outer')

print('Размер:', df.shape)
```

Размер: (10725, 16)

Датафрейм 10 команды

```
In [76]: # 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 [77]: 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 [78]: # 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 [79]: df.isna().sum()

```



```
Out[79]: 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 [80]: # 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 [81]: # 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 [82]: # 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 [83]: # 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 [84]: # 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 [85]: # удаляем слова, которые идут после хэш-тега
df['post'] = df['post'].apply(
    lambda x: re.sub(r'#[^\s]+', ' ', x)
)
```

```
In [86]: # производим замену дефиса на пробел
df["post"] = df["post"].str.replace("-", " ")
```

```
In [87]: # удаляем лишние текстовые символы (те, которые не состоят из букв русского алфавита)
# только русские буквы и пробелы
df['post'] = df['post'].str.replace(
    '^[а-яА-ЯёЁ\\s]', ' ', regex=True
)
```

```
In [88]: %%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: 19.5 s

Wall time: 32.5 s

```

In [89]: # 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 [90]: # 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 [91]: # определяем шаблон регулярного выражения для русских букв
         pattern = '^a-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 [92]: # оценим качество подготовки текста
         df.sample(1)

```

```

Out[92]:

```

	user_id	post	likes	comments	reposts	user_name	user_head	user_work	user_position
2384	nikolay-schwartz	Велком Нужны абсолютно все и разрабы и тестеры и аналитики и руководители проектов	5	1	0	Nikolai Shvarts	Hi-Tech leader with 20+ years of experience in the development of scalable software, support information infrastructure, and telecom.	TechComLab	Product Manager

```
In [93]: 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 [94]: # 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`.

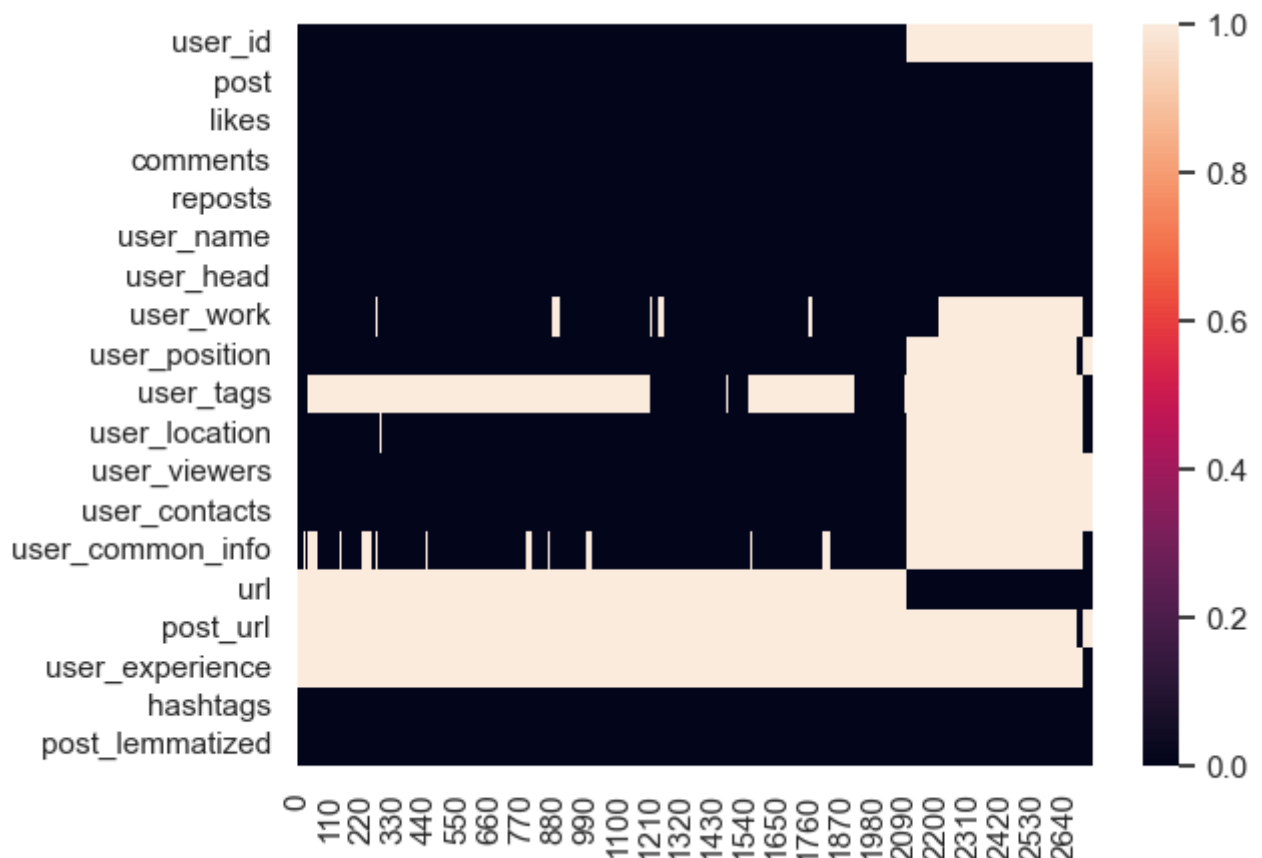
Возможно есть и другие проблемы. Рассмотрим подробнее.

```
In [95]: # проверим на дубли в post_lemmatized
df.post_lemmatized.duplicated().sum()
```

Out[95]: 233

```
In [96]: # удаляем дубликаты
df = df.drop_duplicates(subset='post_lemmatized', ignore_index=True)
```

```
In [97]: # 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 [98]: # % пропусков по полям датасета
round(df.isna().mean() * 100)
```

```
Out[98]: user_id      23.0
         post        0.0
         likes       0.0
         comments    0.0
         reposts     0.0
         user_name    0.0
         user_head    0.0
         user_work    21.0
         user_position 23.0
         user_tags    79.0
         user_location 22.0
         user_viewers 23.0
         user_contacts 23.0
         user_common_info 29.0
         url          77.0
         post_url     99.0
         user_experience 99.0
         hashtags     0.0
         post_lemmatized 0.0
         dtype: float64
```

```
In [99]: 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 [100... # 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 [101... # объединим пользовательские реакции в одну
df['reaction'] = df.likes + df.comments + df.reposts
```

```
In [102... # проверим содержимое поля числа фоловеров
df.user_viewers.unique()
```

```
Out[102]: 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', '1,732', '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', '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', '575', '8,817', '274', '1,074', '772', '13,844',  
      '12,066', '1,230', '725', '460', '2,067', '6,747', '370', '477',  
      '8,203', '1,538', '852', '1,053', '802', '1,160', '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\xa0495',  
      '739', '675', '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', '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', '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', '5\xa0815', '12\xa0836', 0], dtype=object)
```

```
In [103... # 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 [104... # проверим содержимое поля числа контактов  
df.user_contacts.unique()
```

```
Out[104]: array(['500+', '338', '405', '33', '53', '92', '58', '467', '402', '91',  
      '0', '233', '143', '184', '112', '9', '154', '297', '48', '257',  
      '451', '491', '369', '470', '270', '198', '80', '245', '433',  
      '209', '193', '345', '244', '124', '264', '460', '419', '250',  
      '96', '10', '396', '372', '305', nan], dtype=object)
```

```
In [105... # 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 [106... df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2733 entries, 0 to 2732
Data columns (total 20 columns):
#   Column                Non-Null Count  Dtype
---  -
0   user_id               2733 non-null   object
1   post                 2733 non-null   object
2   likes                2733 non-null   int32
3   comments             2733 non-null   int32
4   reposts              2733 non-null   int32
5   user_name            2733 non-null   object
6   user_head            2733 non-null   object
7   user_work            2733 non-null   object
8   user_position        2115 non-null   object
9   user_tags            2733 non-null   object
10  user_location        2733 non-null   object
11  user_viewers         2733 non-null   int32
12  user_contacts        2733 non-null   int32
13  user_common_info     2733 non-null   object
14  url                  2733 non-null   object
15  post_url             2733 non-null   object
16  user_experience      2733 non-null   float64
17  hashtags             2733 non-null   object
18  post_lemmatized      2733 non-null   object
19  reaction            2733 non-null   int32
dtypes: float64(1), int32(6), object(13)
memory usage: 363.1+ KB
```

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

Выборка постов

В соответствии с техническим заданием, нам необходимо найти посты, соответствующие набору ключевых слов. Постараемся выполнить наибольший охват по теме наставничество. В нашем датасете, кроме постов, ключевые слова могут встречаться в тегах и информации о пользователе.

Составим список ключевых слов и выполним поиск.

In [107...

```
# ключевые слова для фильтрации постов
keywords = '|'.join([
    'обучение', 'ментор', 'менторство', 'менторинг', 'тренер', 'советник',
    'наставник', 'наставничество', 'подопечный', 'знания', 'коуч', 'коучинг',
    'опыт', 'опытный', 'развитие', 'скилл', 'mentorship', 'mentor', 'coaching',
    'buddy', 'skills', 'itmentoring'
])

# ищем ключевые слова в постах, тегах пользователей,
# хештегах и информации о пользователе
keywords_filter = (
    (df.post_lemmatized.str.contains(keywords, case=False))
    | (df.user_tags.str.contains(keywords, case=False))
    | (df.hashtags.str.contains(keywords, case=False))
    | (df.user_common_info.str.contains(keywords, case=False))
)

print(
    'Число постов соответствующих наибольшему охвату, по ключевым словам:',
    keywords_filter.sum()
)
```

Число постов соответствующих наибольшему охвату, по ключевым словам: 1317

In [108...

```
# создаем копию датафрейма
df_full = df.copy()
```



```
# оставим только подходящие посты
df = df[keywords_filter]
```

In [109...

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1317 entries, 2 to 2732
Data columns (total 20 columns):
#   Column                Non-Null Count  Dtype
---  -
0   user_id               1317 non-null   object
1   post                  1317 non-null   object
2   likes                 1317 non-null   int32
3   comments              1317 non-null   int32
4   reposts              1317 non-null   int32
5   user_name             1317 non-null   object
6   user_head             1317 non-null   object
7   user_work             1317 non-null   object
8   user_position         1127 non-null   object
9   user_tags             1317 non-null   object
10  user_location         1317 non-null   object
11  user_viewers          1317 non-null   int32
12  user_contacts         1317 non-null   int32
13  user_common_info      1317 non-null   object
14  url                   1317 non-null   object
15  post_url              1317 non-null   object
16  user_experience        1317 non-null   float64
17  hashtags              1317 non-null   object
18  post_lemmatized        1317 non-null   object
19  reaction              1317 non-null   int32
dtypes: float64(1), int32(6), object(13)
memory usage: 185.2+ KB
```

In [110...

```
df_full.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2733 entries, 0 to 2732
Data columns (total 20 columns):
#   Column                Non-Null Count  Dtype
---  -
0   user_id               2733 non-null   object
1   post                  2733 non-null   object
2   likes                 2733 non-null   int32
3   comments              2733 non-null   int32
4   reposts              2733 non-null   int32
5   user_name             2733 non-null   object
6   user_head             2733 non-null   object
7   user_work             2733 non-null   object
8   user_position         2115 non-null   object
9   user_tags             2733 non-null   object
10  user_location         2733 non-null   object
11  user_viewers          2733 non-null   int32
12  user_contacts         2733 non-null   int32
13  user_common_info      2733 non-null   object
14  url                   2733 non-null   object
15  post_url              2733 non-null   object
16  user_experience        2733 non-null   float64
17  hashtags              2733 non-null   object
18  post_lemmatized        2733 non-null   object
19  reaction              2733 non-null   int32
dtypes: float64(1), int32(6), object(13)
memory usage: 363.1+ KB
```

Оценим размеры постов в количестве символов и количестве слов.

In [111...

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

```
return(len(text))
```

```
# подсчет числа слов
```

```
def count_words(text):  
    return(len(text.split()))
```

In [112...

```
# 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()
```



In [113...

```
# характеристики постов по символам  
df.num_chars.describe()
```

Out[113]:

```
count    1317.000000  
mean      418.823842  
std       386.803824  
min         9.000000  
25%      110.000000  
50%      298.000000  
75%      606.000000  
max     1847.000000  
Name: num_chars, dtype: float64
```

In [114...

```
# характеристики постов по словам  
df.num_words.describe()
```

Out[114]:

```
count    1317.000000  
mean       46.135156  
std        42.322398  
min         2.000000  
25%        12.000000  
50%        33.000000  
75%        66.000000  
max       191.000000  
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 [115... # 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 [116... # число записей, попадающих под ограничения
len(df[chars_filter | words_filter])
```

Out[116]: 271

```
In [117... # оценим содержание мелких текстов
df.query('num_chars < @min_chars and num_words < @min_words').post_lemmatized.head()
```

```
Out[117]: 6          эпизод подкаст теория рациональный выбор мешать рациональный
16    предыдущий статья опубликовать размышление цифровой зрелость гибкий разработка
17          статья технократия процесс изменение залетать
29          дописать статья
31          бизнес видеоигра
Name: post_lemmatized, dtype: object
```

```
In [118... # удаляем короткие посты
df = df.query('num_chars >= @min_chars and num_words >= @min_words')
```

```
In [119... # оценка датасета после фильтрации
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1046 entries, 2 to 2732
Data columns (total 22 columns):
#   Column                Non-Null Count  Dtype
---  -
0   user_id                1046 non-null   object
1   post                  1046 non-null   object
2   likes                 1046 non-null   int32
3   comments              1046 non-null   int32
4   reposts               1046 non-null   int32
5   user_name              1046 non-null   object
6   user_head              1046 non-null   object
7   user_work              1046 non-null   object
8   user_position          879 non-null    object
9   user_tags              1046 non-null   object
10  user_location           1046 non-null   object
11  user_viewers            1046 non-null   int32
12  user_contacts           1046 non-null   int32
13  user_common_info        1046 non-null   object
14  url                     1046 non-null   object
15  post_url                1046 non-null   object
16  user_experience         1046 non-null   float64
17  hashtags                1046 non-null   object
18  post_lemmatized         1046 non-null   object
19  reaction                1046 non-null   int32
20  num_chars               1046 non-null   int64
21  num_words               1046 non-null   int64
dtypes: float64(1), int32(6), int64(2), object(13)
memory usage: 163.4+ KB
```

Моделирование

Складываем все лемматизированные тексты в один список.

```
In [147... docs = df["post_lemmatized"].tolist()
docs_full = df_full["post_lemmatized"].tolist()
```

```
In [148... # первые пять элементов
docs[:5]
```

```
Out[148]: ['подкаст мираж платформа аудио инстаграм звук музыка картинка фильм формула любовь марк заха
ров',
'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 развитие продукт продукт сложный веб приложение веб приложение основа дизайн поддер
жка дизайн задача письмо оптимизация конверсия качество ожидать опыт разработка интерфейс сту
дия продуктовый живой дизайн сеть минимум желание вникать разбираться умение основной инструм
ент умение понадобиться предлагать белый заработный офис минута ходьба настольный теннис заня
тие тренер группа английский китайский рабочий мощный маки испытательный срок отклик почта',
'профессия менеджер часами поработать позиция взаимодействовать зрение подчинённый руководит
ель заказчик исполнитель поделиться мысль обменяться видение профессия восприятие',
'запретный плод сладкий удивиться посещаемость вырасти зато сеть крайний мера айтишник сюд
а']
```

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

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

```
In [167... # создаем модель векторизации
tfidf = TfidfVectorizer(min_df=20, max_df=0.9)
```

```
In [168... %%time
# обучим модель и получим векторное представление для каждого текста
x = tfidf.fit_transform(docs)
x_full = tfidf.fit_transform(docs_full)
```

```
CPU times: total: 31.2 ms
Wall time: 78.8 ms
```

```
In [169... # размер полученной матрицы
x.shape, x_full.shape
```

```
Out[169]: ((1046, 515), (2733, 862))
```

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

```

In [170... # список слов векторизатора
tf_feature_names = tfidf.get_feature_names_out()

In [171... # словарь
id2word = {i: token for i, token in enumerate(tf_feature_names)}

In [172... # примеры слов в словаре
id2word[0], id2word[1], id2word[2], id2word[200], id2word[420]

Out[172]: ('абсолютно', 'автоматизация', 'автоматизировать', 'иностраный', 'оценивать')

```

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 [173... # параметры
n_topic_list = [10, 15, 20] # число тем
iter_list = [50, 100, 150] # число итераций

In [174... %%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: 719 ms
Wall time: 15.3 s

```

```

In [175... 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[175]:

Сравнительная таблица качества моделирования

	n_topics	max_iter	lda_score
0	10	50	-31134.117720
1	10	100	-31134.117718
2	10	150	-31134.117718
3	15	50	-32019.004623
4	15	100	-32019.004623
5	15	150	-32019.004623
6	20	50	-32722.311906
7	20	100	-32722.311904
8	20	150	-32722.311904

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 [176...

```
%%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_full = LatentDirichletAllocation(
    n_components=n_topics,
    max_iter=n_iters,
    random_state=SEED
)

lda_topics = lda.fit_transform(x)
lda_topics_full = lda_full.fit_transform(x_full)
```

CPU times: total: 14.1 s

Wall time: 19.3 s

In [177...

```
# размер полученной матрицы
lda_topics.shape, lda_topics_full.shape
```

Out[177]:

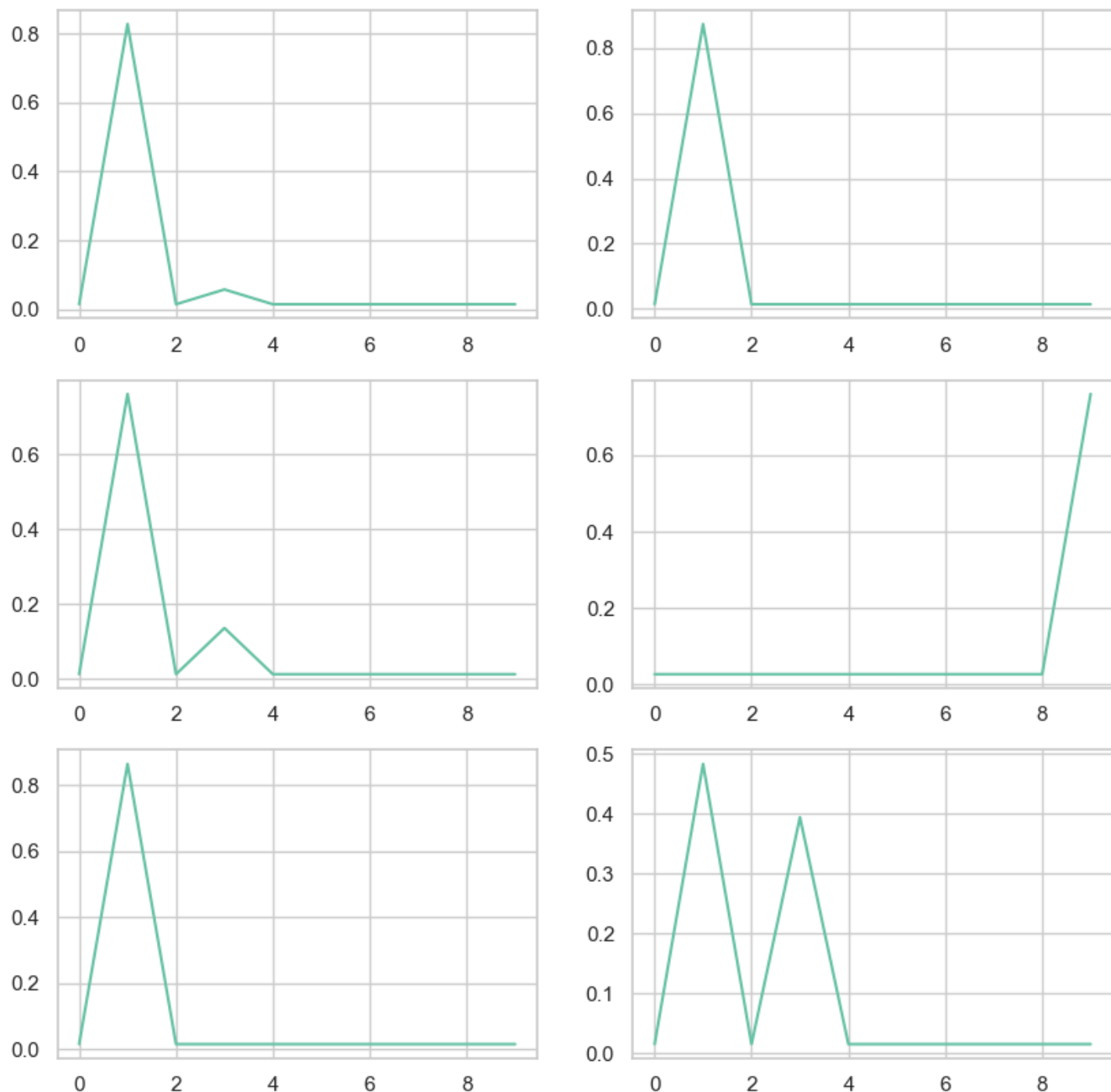
```
((1046, 10), (2733, 10))
```

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 [180...

```
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 [181...

```
# 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()

```

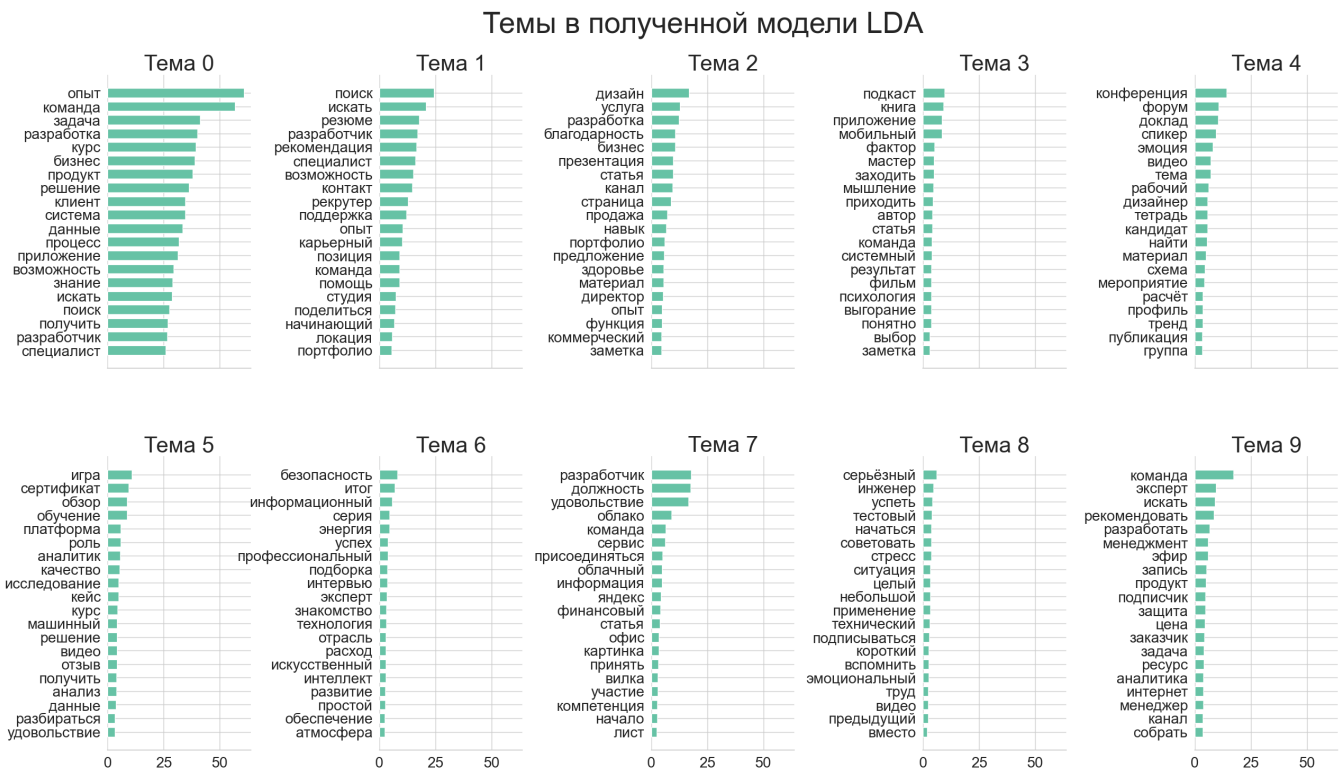
In [182...

```

# число ключевых слов в теме
n_top_words = 20

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

```



Тема 0 выделяется из 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: Опыт и разработка в команде
- Тема 1: Поиск работы и развитие карьеры разработчика
- Тема 2: Дизайн и предложения услуг
- Тема 3: Подкасты, книги и приложения для личного развития
- Тема 4: Конференции и форумы для обмена опытом
- Тема 5: Обучение и аналитика в сфере игр
- Тема 6: Информационная безопасность и профессиональное развитие
- Тема 7: Работа разработчика в облачных сервисах
- Тема 8: Стресс и ситуации в работе разработчика
- Тема 9: Рекомендации и разработка продуктов в команде

Типичные статьи

In [184...

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

Тема 0

Разрыв между линией соответствия и надежной оценкой угроз и рисков может привести к значительному уровню дезинформированных расходов. Специалист по управлению безопасностью и защитой данных выполняет неустанную миссию, чтобы прорваться сквозь мрак новых модных словечек и сделать нашу жизнь немного проще.

Тема 1

Присоединяйтесь к нашей команде в качестве маркетолога в международную компанию. Мы ищем высококвалифицированного специалиста, который готов взять на себя миссию привлечения новых клиентов и увеличения продаж. Это компания, которая помогает своим клиентам прокачивать карьерные возможности за счет свободного английского и развития в международном бизнесе. Мы ищем ориентированного на результат маркетолога, который умеет запускать образовательные продукты онлайн, создавать и осуществлять маркетинговую стратегию, распределять маркетинговый бюджет по каналам, создавать высоко конверсионные воронки и выстраивать стратегию развития в медиа. Кроме того, вам предстоит провести с потенциальными потребителями составительские и запустить корпоративные тренинги по выходу на международный рынок, дополнить команду качественными ребятами под задачи бизнеса, разработать алгоритм и сделать объемный анализ конкурентов, а также постоянно анализировать сквозную аналитику для принятия управленческих решений. Если вы готовы взять на себя эти задачи и сделать маркетинговый прорыв, который обеспечит увеличение оборота компании, мы ждем ваших резюме и предложений. Мы ищем конкретных людей, которые готовы принимать вызовы и достигать высоких результатов. Присоединяйтесь к нашей команде и помогите нам развиваться и расти вместе. Переходите по ссылке, вам нужно заполнить анкету.

Тема 2

Откликайтесь красиво. Кому задизайнить? Всем привет. Пока нахожусь в поиске работы, но моя первая красная сама себя не купит. Готов сделать красиво, читаемо и удобно всего за российских рублях. Оплатить можно будет из любой точки планеты, даже получите фискальный чек, если кому то необходимо. Для связи буду благодарен за лайк, репост или любую другую реакцию. Картинка для привлечения внимания.

Тема 3

Всем привет в команду, требуются разработчики и дизайнер на Команду, активно развивает в Казахстане направление управление с пецтехникой и оборудованием в промышленных компаниях. Что мы обещаем: Все условия для комфортной работы, Достойную зарплату, которая ограничена только вашими стараниями, бонусы и опцион в проекте, Доступ к курсам и обучение за счет компании. Какие требования: Для отличных знаний, Фреймворк, Наличие опыта кроссбраузерной верстки для разных платформ, Для стек технологий для нас не на первом месте. Нужен человек, умеющий решать задачи, находить решения и постоянно улучшать свои результаты. Общие требования: Приветствуется знание методологии, Знание паттерна, Понимание методологии и принципы, Умение писать красивый структурированный код без мусора. Портфолио приветствуется. Обязательным требованием является умение закрывать задачи в срок. Умение планировать сроки исполнения в зависимости от объема работ. Желание развиваться и обостренное чувство прекрасного. Основной офис Алматы, можем рассмотреть удаленку.

Тема 4

Не самый этичный, но очень хитрый способ пройти собеседование. Программист сделал нейронку, которая может на лету генерировать текст из прямого эфира. И к тому же, во втором столбике дает ответы на вопросы, если кто-то на трансляции их задавал. Один инструмент сломал всю систему онлайн интервью.

Тема 5

Каким был прошлый год для хайтека и чего ждать теперь? Он стал для израильских технологий весьма неоднозначным годом, особенно его вторая половина. Хотя пандемия пошла отрасли на пользу, сейчас ее накрыли сокращения и снижение инвестиций.

Тема 6

Я так рада, что у меня есть работа. В которую можно уйти от всего происходящего вокруг и психовать из-за автотеста, с которым ты можешь бороться и победить. Чувствовать, что здесь всё зависит только от тебя и твоих усилий. Этого сейчас очень не хватает. Ещё я рада, что я нахожусь в правильной компании, которая понимает страхи сотрудников и старается помочь, ну насколько это возможно. Третий повод для радости, это пройденный учебник по

рекомендую И последний новое слово сказанное трехлетним сыном Давайте искать поводы для радости поводы для печали сами как то находятся сейчас

Тема 7

Где проверить свои скиллыувидеться с комьюнити и найти единомышленников На соревновании для опытных разработчиков инженеров и системных аналитиков Шесть направленийдва тура и весомые призы в финалекоторый пройдет в Москве апреля Кстатипобедителям от борочного этапа из городов РФ и Беларуси Тинькофф оплатит билеты на финал в Москву Участвовать в отборе можно онлайн или очно Для техкто предпочитает живое общениемы открыли площадки в городах Один из них Ростов на Дону Не забудьте указать его при регистрациичтобы получить приглашение на площадку и увидеться вживую Выбирайте трек и регистрируйтесь до апреля на странице соревнований

Тема 8

Какие же вопросы задают на собеседованиях аналитикам данных в Собрала в карусельку примеры вопросов по категориям сохраняйте к себе для подготовки к следующему собесу Общие вопросы о вас как аналитике данных Смысловые и бизнес вопросы в работе аналитика Об опыте и биографии Углублённые темы которые любят работодателиКаких вопросов как считаете не хватает в списке Какие хотите чтобы мы разобрали вместе Пишите в комменты Расскажу какой ответ ждёт от вас компания в моём тг канале

Тема 9

Отличный отчет где вы можете ознакомиться с примерами того как другие компании комбинируют маркетинговые решения в своих маркетинговых стратегиях Рекомендую также прочитать эту статью в которой можно обратить внимание на весь ландшафт маркетинговых технологий Ну и конечно сам Где можно узнать много полезной информации и новостей из мира маркетинга

Сохраним в датафрейм номер наиболее вероятной темы для каждого поста.

In [186...

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

# значения наиболее вероятных топиков для полного датасета
df_full['lda_topic'] = np.argmax(lda_topics_full, 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 [199...

```
%%time

# число тем
n_topics = 10
n_iters = 300

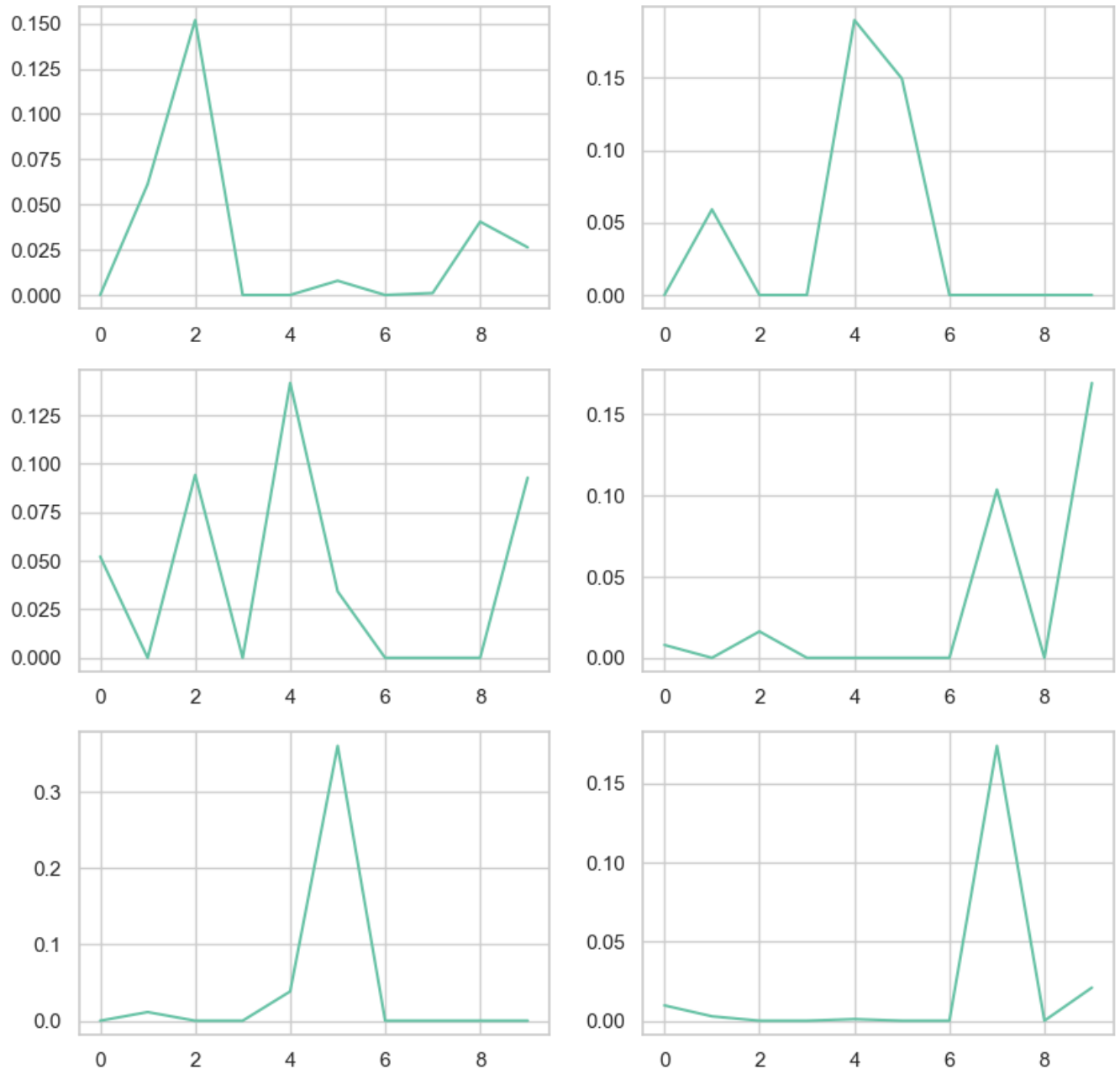
# создаем модель
nmf = NMF(n_components=n_topics, max_iter=n_iters, random_state=SEED)
nmf_full = NMF(n_components=n_topics, max_iter=n_iters, random_state=SEED)
```

```
# обучаемся
nmf_topics = nmf.fit_transform(x)
nmf_topics_full = nmf_full.fit_transform(x_full)
```

CPU times: total: 109 ms
Wall time: 148 ms

In [200...

```
# 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])
```



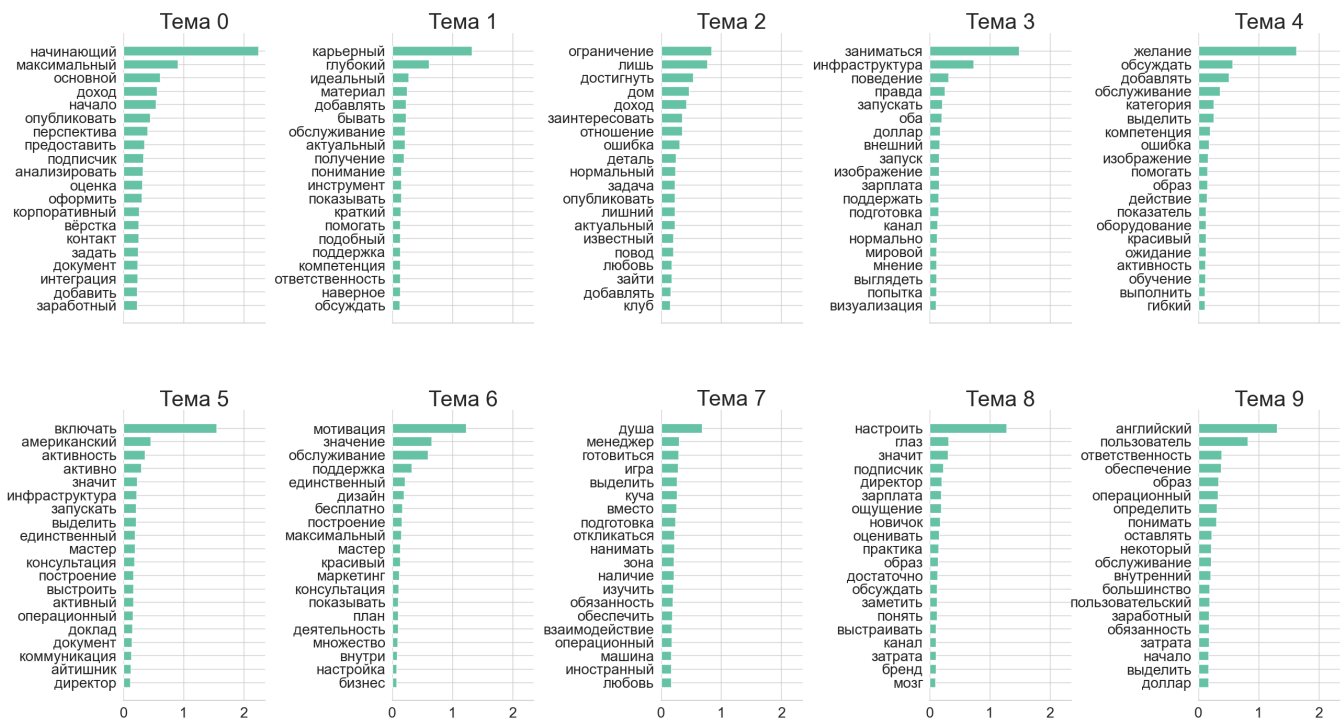
Как и в случае с LDA, публикации могут принадлежать одновременно нескольким темам.

Ключевые слова

In [201...

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

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



Интерпретация тем для NMF

- Тема 0: Начало бизнеса, максимальный доход, анализ и оценка.
- Тема 1: Карьерное развитие, глубокое понимание, поддержка и обсуждение.
- Тема 2: Преодоление ограничений, интерес к домашнему бизнесу.
- Тема 3: Занятие деятельностью, инфраструктура, визуализация.
- Тема 4: Желание и обсуждение, компетенции и гибкость.
- Тема 5: Активность и коммуникация, операционная инфраструктура.
- Тема 6: Мотивация и поддержка, дизайн и маркетинг.
- Тема 7: Личность менеджера, готовность и взаимодействие.
- Тема 8: Настройка и оценка, роль подписчиков и директора.
- Тема 9: Английский язык, ответственность и обслуживание

Типичные статьи

In [203...

```
# 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

Добрый день Дорогие Друзья Прошлая моя публикация была посвященаТеме ИТ продакт менеджмент Вступление Коротко и простыми словами Оставил ссылку в В этот же раз Я хочу рассказать В ам о процессе зарождения ИТ продукта так называемой стадии Тема Зарождение ИТ продукта Минимально Жизнеспособный Продукт это тестовый ИТ продукт с минимальным и основополагающи м набором функций либо одной единственной функцией Цель проверка жизнеспособности и во стребованности создаваемого ИТ продукта на рынке На стадии Продакт менеджер получает обр атную связь от целевой аудитории и понимает стоит ли развивать ИТ продукт дальше либо отказа тся от задуманного какие изменения следует в него внести а что оставить как есть делит ся по способам создания основные из них Лэндинг пейдж ИТ продукт представляется в вид е посадочной страницы На странице размещается описание продукта информация о его преимуществах идет некий диалог с потенциальным пользователем а в конце добавляется кн опка Зарегистрироваться с целью получения новостей по данному продукту по е мейл Лэндинг п ейдж позволяет оценить заинтересованность в продукте собрать базу подписчиков и в дальнейше м развивать с ними диалог Консьерж Все операции проводятся командой разработчиков прод укта в ручную также как это делает Консьерж в отеле Чтобы не разрабатывать приложение трати ть на это деньги и время можно самим стать этим приложением на первом этапе Такими образом вы оцените спрос и обратную связь получите опыт живого общения с пользователем Волшеб ник страны Оз ИТ продукт имеет оболочку например сайт но все операции проводятся командой р азработчиков также в ручную завуалировано Вы как бы создаете у пользователя иллюзию искуст венного интеллекта как Волшебник Волшебник страны Оз позволяет не только проверить идею сэк ономить на разработке автоматизированного функционала но и узнать правильно ли сконструиров ан сайт оболочка с точки зрения пользователя какие созданы удобства и неудобства Од на функция ИТ продукт представлен в виде Одной единственной функции уникального товарного пр едложения УТП чаще всего и оформлен в виде мобильного или веб приложения бота например Обычно создается для инвесторов чтобы показать будет работать идея УТП или нет Пред варительный заказ Описание будущего ИТ продукта размещается на краудфандинговой платформе ан ализируется интерес привлекаются средства и первые пользователи Ну вот и все Стадия очень важна Грамотное ее проведение позволяет правильно оценить идею избежать массу ошибок финансовых потерь запустить крутой ИТ продукт себе и людям на радость До скорого

Тема 1

Всем привет Наша команда ищет инженера в крупный финтех проект инвестиции Фор мат работы по желанию удаленно офис гибрид если Москва Саратов Пенза Возможна работа вн е 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 настройки и по пыт работы с системами Одержании 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 Опы т понимание принципов работы высоконагруженных высокодоступных систем Опыт работы с Опыт работы с системами виртуализации Опыт работы с системами Ком пания 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 в нутренним учебным программам Возможности профессионального роста и развития Лучше сразу при ходите в Буду рада ответить на все вопросы и рассказать про детали

Тема 2

Как составить резюме для работы за границей В этом видео я разберу р езюме который планирует поиск работы за границей и поделюсь теми фишк ами которые важно учесть для того чтобы получать отклики и приглашения на собеседования ТА ЙМКоды Вступление Как участвовать в разборе резюме Что писать в разделе о себе Как заполнять раздел с контактами Как прописывать ключевые слова Как у казывать языки для общения Как отправлять резюме в разных странах напрямую Как оп исывать достижения и обязанности Как проходить под требования об опыте работы Где брат ь дополнительный опыт Какие шаблоны для резюме лучше всего использовать для поиска раб оты за границей Как настроить доступ к резюме Как проверить корректность текста в резюме на английском Что влияет на получение работы кроме резюме Что делать если не получается найти работу Где можно задать мне вопрос про поиск работы чтобы по лучить развернутый ответ

Тема 3

Продолжая собирать полезную информацию из группы в Машинное обучение Курс на Курсере по т ряд статей с примерам задачам и кодом Советую Курс от Курс от МФТИ Курс Линейная алгебра Курс линейной алгебры Преподает легендарный профессор Массачусетского технологического института Гилберт Стрэнг канал поможет понять линейную алгебру Проверено хороший ресурс

Тема 4

Срочно в эфир Две вакансии в две прекрасные компании Лучший книжный сервис в мире ищет в свою команду разработчика Отличная команда интересные задачи отсутствие занудной бюрократии и проект за который не стыдно Пишите мне в личку все расскажу а подробности по ссылке Британский стартап ищет к себе в команду разработчика Условия огонь перспективы манят своим размахом но и уровень тоже нужен соответствующий Очень желателен опыт с Подробности по этой вакансии тут Ну и тоже у меня в личке конечно же А еще компании выше ищут не только их но и бекендеров айосеров аналитиков и кого только не так что лайк шер репост и пусть все найдут компанию своей мечты

Тема 5

Всем привет Мы ищем опытного Аналитика данных а точнее даже двух аналитиков с хорошим английским опытом в и желательно со знанием Ниже перечислены все основные требования к вакансии наших партнеров работа не в международной аудиторской компании из Франции Условия полная удаленка работа частично на английском языке частично на русском официальное трудоустройство в компанию партнера нет треккинга времени заработная плата от до Требования года опыта с Отличное знание и языков Менеджмент баз данных отличное знание Работа на результат без трекинга времени работа с международными клиентами Опыт с будет плюсом Задачи Сбор и формализация требований клиентов к аналитике данных Анализ потоков данных в организациях и обеспечение качества данных Построение эффективных и масштабируемых моделей данных Подключение и сбор данных из разных источников базы данных облачные и локальные источники Разработка полностью автоматизированных отчетов высокой сложности Проверка работы младших аналитиков данных коучинг Откликайтесь и присылайте резюме здесь или мне в телеграм Дарья Все м отвечу

Тема 6

Тренды в мобильной разработке Современный мир невозможно представить без мобильных устройств и приложений Несмотря на уже достигнутые высоты мобильная разработка продолжает активно развиваться поэтому существуют определенные тенденции которые наиболее ярко проявляются в этой сфере Развитие технологий Современные мобильные приложения становятся все более сложными и функциональными что требует развития технологий Одной из главных тенденций является развитие и усовершенствование технологий таких как и которые позволяют создавать мобильные приложения для нескольких платформ одновременно Разработка без кода Одной из новых тенденций в мобильной разработке является разработка без кода Это подход который позволяет создавать приложения без необходимости писать код Вместо этого разработчики используют графические интерфейсы и инструменты для создания приложений Этот подход может ускорить процесс разработки и снизить затраты на создание приложения Искусственный интеллект и машинное обучение Искусственный интеллект и машинное обучение являются ключевыми направлениями развития мобильной разработки в настоящее время Многие компании уже внедрили технологии в свои приложения например голосовые помощники и распознавание текста Кроме того машинное обучение позволяет создавать персонализированные рекомендации и улучшать пользовательский опыт Безопасность С ростом количества мобильных устройств и приложений возрастает и угроза кибератак Поэтому безопасность является одной из главных тенденций в мобильной разработке Разработчики должны уделять большое внимание защите данных пользователей и использованию криптографии Интернет вещей С каждым годом увеличивается количество устройств подключенных к интернету Это открывает новые возможности для мобильных приложений которые могут управлять умными домами автомобилями и другой техникой Также интернет вещей позволяет собирать большое количество данных которые можно использовать для улучшения мобильного приложения Итак мобильная разработка продолжает активно развиваться и существует множество тенденций которые определяют ее направление Разработчики мобильных приложений должны следить за тенденциями и использовать новые технологии чтобы создавать более функциональные и безопасные приложения

Тема 7

Всем привет Хочу поделиться интересными идеями из книги Барбары Шер О чем мечтать Как понять что чего хочешь на самом деле и как этого добиться По названию думал что это очередная книга про успешный успех и это будет лёгкое чтение на раз все оказалось совсем наоборот Читал медленно вникая во все аспекты Что зацепило Очень актуальная тема и самый частый запрос от клиентов я не знаю чего я хочу Как в карьерном консультировании так и в психологии часто встречаюсь с я чем то занимаюсь и не знаю нравиться оно мне или нет или какие нибудь другие вариации этой темы Структура выстроены и объяснены причины и следствия часто психологические Доходчивость рассказано на простых примерах Не даётся волшебная таблетка Для чего то нужно выделить время что то нужно эмоционально прожить над чем то нужно работать каждый день Много диагностических упражнений для саморефлексии Похвалить не получается Сразу происходит отсев по критерию есть ли действительно мотивация и ресурсы на изменения сейчас С определенными вопросами рекомендуется обращаться к психологу так как не все можно решить самостоятельно В общем простая идея на которую не все обращают внимание Чтобы добиться чего ты хочешь для начала нужно понять чего ты хочешь а на это обычно нет времени потому что нужно добиваться а не сидеть думать Потом построить план и работать по нему Подытожив можно сказать что книга заставляет повышаться осознанность в жизни и карьере Со своей стороны как специалист могу быстро и качественно помочь вам вашим друзьям близким коллегам с аналогичными запросами подробнее про меня можно прочитать на моем сайте Спасибо за внимание

Тема 8

Коллеги я доделала сайт для а то столько лет уже делаем добро а единой точки входа нет Сайт будет и дальше наполняться а пока просто приглашаю в гости Там уже есть информация обо всех наших соцсетях о программах менторинга и ролевой модели Ну и программа менторинга уже в самом разгаре еще можно успеть стать ментором менти или спикером

Тема 9

Три года я развиваю терапию новое направление в области управления человеческим капиталом в основе которого системный подход построенный на знаниях и инструментах из нейробиологии и психологии бизнес коучинга и менеджмента За плечами несколько лет настоящей научной деятельности экспериментов исследований проб и ошибок И сегодня терапия уже по настоящему похожа на отдельную профессию суть и смысл которой помогать бизнесу достигать своих амбициозных целей через управление состоянием менеджеров и команд развитие глубоких доверительных отношений внутри бизнес партнерств в команде командой построение человекоцентричных корпоративных культур и бизнес процессов в развитие лидерских компетенций нового времени Это главные зоны ответственности для терапевта И если до сих пор я сопровождала управленцев и их бизнесы как внешний терапевт то теперь я вижу как это может развиваться дальше и для следующего шага очень хочу поделиться опытом терапевта Что ищу креативный бизнес в идеале средний и готов обсуждать полных дня в неделю подчинение первому лицу это критично в т.ч. для эффективности взаимодействия амбициозных провокаторов инноваторов открытых смелых предпринимателей понимающих суть и ценность такой поддержки для людей и бизнеса компании с продвинутой или стремящейся к этому и ко Вам точно нужен терапевт если у вас есть бизнес в активной фазе развития прямо сейчас вы переживаете бизнес или культурную трансформацию на вас и команду ожидается большая нагрузка а в ближайшее время что то не так с атмосферой в партнерстве в команде что то не так с состоянием менеджеров и или команд нездоровая культура слабый и незрелый менеджмент есть другие задачи на стыке бизнеса и психологии Итак Кто там до сих пор фантазирует о Венди Роудс дайте о себе знать Для обеих сторон это немного авантюра она обязательно даст свои плоды как для меня и вашего бизнеса так и для всего современного бизнес ландшафта Резюме и ответы на вопросы от потенциальных работодателей в лс Какими ещё могут быть задачи вашего терапевта в первом комментарии Дружите если вы знаете кого то для кого это может быть актуально поделитесь пожалуйста

In [209...]

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

# значения наиболее вероятных топиков для полного датасета
df_full['nmf_topic'] = np.argmax(nmf_topics_full, axis=1)
```

Вывод:

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

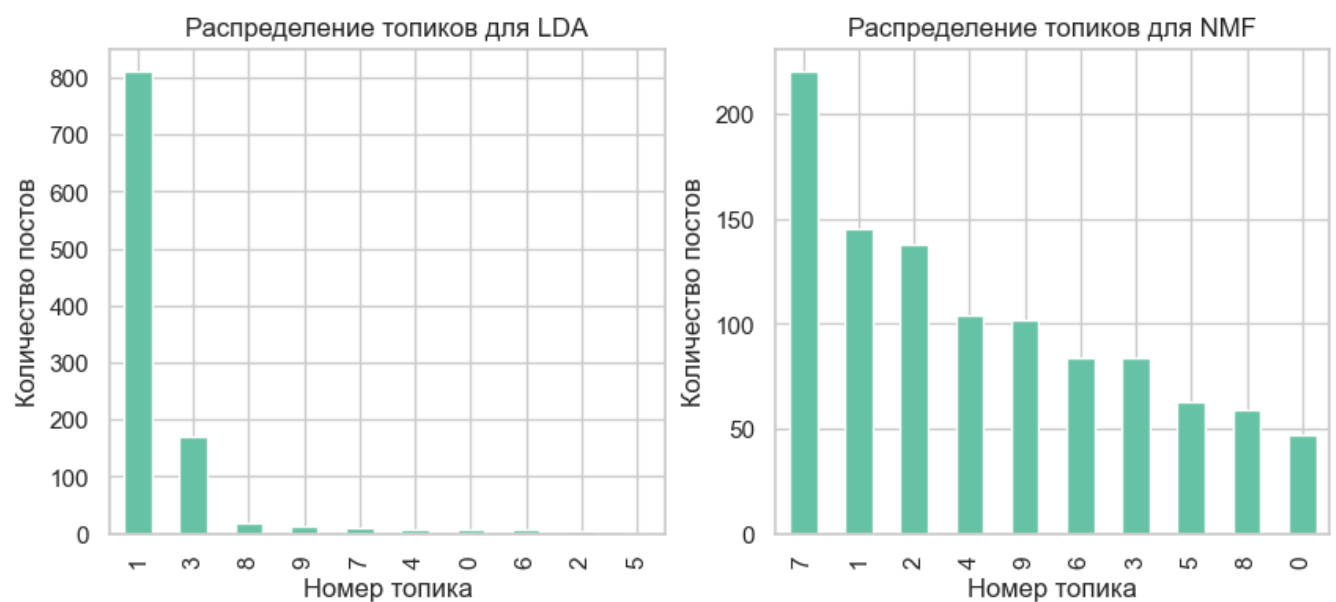
ТОП-10 тем постов целевой аудитории

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 [210...

```
# распределение топиков для LDA
plt.figure(figsize=(10,4))
plt.subplot(1,2,1)
df.lda_topic.value_counts().plot(
    kind='bar', xlabel='Номер топика', ylabel='Количество постов',
    title='Распределение топиков для LDA'
)
plt.subplot(1,2,2)
df.nmf_topic.value_counts().plot(
    kind='bar', xlabel='Номер топика', ylabel='Количество постов',
    title='Распределение топиков для NMF'
);
```



Алгоритм LDA отдает предпочтение топику под номером 1 и 3. 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: 7, 1, 2, 4 и 9.

- Тема 0: Начало бизнеса, максимальный доход, анализ и оценка.
- **Тема 1: Карьерное развитие, глубокое понимание, поддержка и обсуждение.**
- **Тема 2: Преодоление ограничений, интерес к домашнему бизнесу.**
- Тема 3: Занятие деятельностью, инфраструктура, визуализация.
- **Тема 4: Желание и обсуждение, компетенции и гибкость.**
- Тема 5: Активность и коммуникация, операционная инфраструктура.
- Тема 6: Мотивация и поддержка, дизайн и маркетинг.
- **Тема 7: Личность менеджера, готовность и взаимодействие.**

- Тема 8: Настройка и оценка, роль подписчиков и директора.
- **Тема 9: Английский язык, ответственность и обслуживание**

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 на полном датасете.

Выведем интерпретацию тем для полного датасета.

- Тема 0: Бизнес-развитие и управление продуктом
- Тема 1: Развитие команды и благодарность
- Тема 2: Поиск работы и карьерное развитие
- Тема 3: Опыт и навыки в разработке
- Тема 4: Работа разработчиком и конференции
- Тема 5: Каналы и материалы о дизайне
- Тема 6: Курсы и обучение онлайн
- Тема 7: Поиск специалистов и рекомендации
- Тема 8: Удовольствие от работы и персональное развитие
- Тема 9: Разработка приложений и услуг

In [208...

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

Out[208]:

	comments	likes	reaction	reposts
nmf_topic				
0	1757	16169	18704	778
1	344	3724	4396	328
2	1580	21821	26296	2895
3	1012	14441	16820	1367
4	232	1253	1564	79
5	504	7030	8085	551
6	2026	23402	27377	1949
7	664	3537	4392	191
8	542	7707	8299	50
9	305	1949	2573	319

В целом видна корреляция между разными типами реакций.

Из 10 тем, в качестве наиболее популярных и интересных можно отметить темы: 6, 2, 0, 3, 8.

- **Тема 0: Бизнес-развитие и управление продуктом**
- Тема 1: Развитие команды и благодарность
- **Тема 2: Поиск работы и карьерное развитие**

- **Тема 3: Опыт и навыки в разработке**
- Тема 4: Работа разработчиком и конференции
- Тема 5: Каналы и материалы о дизайне
- **Тема 6: Курсы и обучение онлайн**
- Тема 7: Поиск специалистов и рекомендации
- **Тема 8: Удовольствие от работы и персональное развитие**
- Тема 9: Разработка приложений и услуг

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.

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 7: Личность менеджера, готовность и взаимодействие.
- Тема 1: Карьерное развитие, глубокое понимание, поддержка и обсуждение.
- Тема 2: Преодоление ограничений, интерес к домашнему бизнесу.
- Тема 4: Желание и обсуждение, компетенции и гибкость.
- Тема 9: Английский язык, ответственность и обслуживание

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

- Тема 6: Курсы и обучение онлайн
- Тема 2: Поиск работы и карьерное развитие
- Тема 0: Бизнес-развитие и управление продуктом
- Тема 3: Опыт и навыки в разработке
- Тема 8: Удовольствие от работы и персональное развитие

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 []: