Исследовательский хакатон Яндекс Практикума

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.

1. 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*.

1.1. Оценка результатов ручного поиска

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

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

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

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

Первым делом загрузим все необходимые для работы библиотеки.

In [96]:
```
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)

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

In [7]:
```
# прокрутка страницы, для подгрузки 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 профиля
    """
    # прокручиваем страницу до конца что бы подгрузился 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 [10]:
```
# запускаем браузер
driver = chrome_start()
```

In [11]:
```
# 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 [12]:
```
# параметры поисковых запросов, теги, темы публикаций

#KEYWORDS = 'разработка по'
#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 [13]:
```
# 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 [14]:
```
# закрываем браузер
driver.quit()
```

Собираем все id в один датафрейм

In [15]:
```
# 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 [16]:
```
# 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 [17]:
```
# 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 приступать к сбору данных о людях и парсингу постов.

1.3. Парсинг постов и профилей

In [18]:
```
# запускаем браузер
driver = chrome_start()
```

In [19]:
```
# входим в LinkedIn
linkedin_login(driver)
```

Парсим профили и посты

In [20]:
```
# 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 [21]:
```
# 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 продолжит сбор данных с того мета, где остановились.

In [22]:
```
# с какого профиля стартуем
# если ранее парсинг был прерван, продолжаем с того же места
start_idx = profiles.user_name.nunique()
start_idx
```

Out[22]:
```
426
```

In [23]:
```
# 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()
```

```
hanna-tiselko-8b9a4825
posts: 41
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
```

In [24]:
```
# закрываем браузер
driver.quit()
```

Результат

In [25]:
```
# 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        427 non-null    object
 2   user_head        427 non-null    object
 3   user_work        397 non-null    object
 4   user_position    427 non-null    object
 5   user_tags        139 non-null    object
 6   user_location    425 non-null    object
 7   user_viewers     429 non-null    object
 8   user_contacts    429 non-null    object
 9   user_common_info  398 non-null    object
dtypes: object(10)
memory usage: 146.9+ KB
```

In [26]:
```
posts.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 9345 entries, 0 to 9344
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   user_id   9345 non-null   object
 1   text      9345 non-null   object
 2   likes     9345 non-null   object
 3   comments  9345 non-null   object
 4   reposts   9345 non-null   object
dtypes: object(5)
memory usage: 438.0+ 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 сбирались данные, были заблокированы сервисом Linkedin.

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

2. Обработка данных

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

In [27]:
```
# оценим датафрейм с постами
posts.head(2)
```

Out[27]:

	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 [28]:
```
# оценим датафрейм с информацией о пользователях
profiles.head(2)
```

Out[28]:

	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 [29]:
```
# переименуем столбец text в post для лучшего отражения содержимого
posts = posts.rename(columns={'text': 'post'})
```

2.1. Предобработка

In [30]:
```
# 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
                               # другие разные символы
                               u"\U00002702-\U000027B0"
                               u"\U00002702-\U000027B0"
                               u"\U000024C2-\U0001F251"
                               u"\U0001f926-\U0001f937"
                               u"\U00010000-\U0010ffff"
                               u"\u2640-\u2642"
                               u"\u2600-\u2B55"
                               u"\u200d"
                               u"\u23cf"
                               u"\u23e9"
                               u"\u231a"
                               u"\ufe0f"
                               u"\u3030"
                               "]+", flags=re.UNICODE)
    # Удаляем эмодзи, используя паттерны
    text_without_emojis = emoji_pattern.sub(r'', text)
    return text_without_emojis

# удаляем эмодзи из постов
posts['post'] = posts['post'].apply(
    lambda x: remove_emojis(x) if pd.notnull(x) else x
)
```

In [31]:
```
# удалим посты на украинском языке
# определяем шаблон для украинских символов
# (по специфичным для данного языка символам)
ukrainian_pattern = r'[ЄєІіЇїҐґ]'

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

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

Хештеги, которые встречаются в тексте поста, выносим в отдельный столбец.

In [32]:
```
# сохраняем хэштэги в отдельный столбец перед их удалением из постов
posts['hashtags'] = posts['post'].str.findall(r'#([^\s]+)').apply(lambda x: ', '.join(x))
```

2.2. Подготовка текста

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

In [33]:
```
%%time
# функция лемматизации текста
morph = pymorphy2.MorphAnalyzer()
def lemmatize_text(text):
    lemmatized_words = [morph.parse(word)[0].normal_form for word in text.split()]
    return ' '.join(lemmatized_words)

# лемматизируем посты
posts['post_lemmatized'] = posts['post'].apply(lemmatize_text)
```

```
CPU times: total: 24.7 s
Wall time: 29.6 s
```

In [34]:
```python
# удаляем слова, которые идут после хэш-тэга
posts['post_lemmatized'] = posts['post_lemmatized'].apply(
    lambda x: re.sub(r'#[^\s]+', ' ', x)
)
```

In [35]:
```python
# производим замену дефиса на пробел
posts["post_lemmatized"] = posts["post_lemmatized"].str.replace("-", " ")
```

In [36]:
```python
# удаляем лишние текстовые символы (те, которые не состоят из букв русского алфавита)
# только русские буквы и пробелы
posts['post_lemmatized'] = posts['post_lemmatized'].str.replace(
    '[^а-яА-ЯёЁ\s]', ' ', regex=True
)
```

In [37]:
```python
# 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 [38]:
```python
# удаляем стоп-слова и слова-паразиты
posts['post_lemmatized'] = posts['post_lemmatized'].apply(
    lambda x: ' '.join([word for word in x.split() if word not in stop_words])
)
posts['post_lemmatized'] = posts['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 [39]:
```python
# определяем шаблон регулярного выражения для русских букв
pattern = '[^а-яА-ЯёЁ]'
# создаем маску, чтобы проверить, содержит ли каждая ячейка русские буквы
mask = posts['post_lemmatized'].str.contains(pattern, regex=True)
# фильтруем датафрейм, используя маску
posts = posts[mask]
```

In [40]:
```python
# оценим качество подготовки текста
posts.sample(2)
```

Out[40]:

	user_id	post	likes	comments	reposts	
3683	yuliii	Как считаете, стресс-интервью вышло из моды еще в 2016 или вообще в нее не входило?Да-да, строки сегодняшние хочу посвятить именно этому и поделиться опытом прохождения интервью в таком формате.Дело было холодным осенним днем 2021 года. Рассматривала предложения о работе и молодой человек из кадрового агентства любезно предложил пообщаться с операционным директором одной компании, пилящей ПО для автобизнеса. Захожу на интервью в zoom и сидит передо мной представительный серьезный дядька. Начиналось все красиво: какие вакансии закрывала, чем отличается java от js, сколько лет опыт работы и проч. Неожиданно его волосатая рука падает на стол и он прикрикивает: "И ты серьезно думаешь, что с таким опытом подойдешь нашей компании?" Я думаю: "Ха, а он любит поиграть" и отвечаю: "А вы серьезно считаете, что сможете обеспечить меня интересными вакансиями для работы?" Он улыбнулся, откинулся в кресле и продолжил: "Историческое образование, да? Ну давай поговорим".На меня посыпались вопросы про Спарту, Помпею, Петра I, Ось (2-я мировая), Рузвельта, Трумэна и, мое любимое, "Кто же убил Кеннеди?" На удивление, у меня не возникло желание отключиться, а наоборот, было смешно смотреть на его вытягивающееся и белеющее лицо.Он давал мне какие-то задачи по теории вероятности, поиск 2/3 от числа, поиск кратчайшего пути для шарика и пара еще бредовых.Естественно все 2 (!) часа нашего общения он сидел с максимально надменным каменным лицом и временами кричал. Меня лишь веселил весь этот кринж.Через несколько дней я получила оффер от компании и попросила операционного директора никогда больше таких экспериментов над кандидатами не проводить.Слово своё мужское сдержал, насколько мне известно. #interview #joboffer #homeoffice #networking	73	33	2	interview, joboffer, home
1464	anna-zaytseva-hrst	Вы в поисках работы?#карьера Компании и сотрудники вышли из анабиоза после новогодних праздников, поэтому самое время возобновить поиск новых возможностей и работу по упаковке своего профессионального опыта. Приглашаю к себе на карьерную консультацию:Работаю как в консультационном так и в коучинновом (ICF) подходе; Проработаем всё, что связано с поиском новой работы, исходя из вашего профессионального бекграунда, пожеланий к работодателю и географии;Помогу разобраться что делать и куда идти, когда текущие задачи и роль «поперек горла», а чем еще можно заниматься - понимания нет;Работаю со страхами и ограничениями - поиск работы может быть комфортным; При необходимости - работа с резюме, вашим позиционированием, прохождение	31	0	0	карьерныйконсультант#по

user_id	post	likes	comments	reposts
	собеседований.По наблюдениям - самым сложным для профессионала является рассказ о собственном опыте, продажа своих компетенций работодателю. Проведу бережную, но экспертную оценку вашего опыта и подскажу, на чем делать акценты, чтобы цифра в оффере вас порадовалаДоговариваюсь о консультации через Телеграмм @HrOffer или почту az@hrst.ru Приходите, буду вам рада #карьерныйконсультант#поискработы#коуч #резюме			

In [41]:
```
posts.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2240 entries, 0 to 9253
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   user_id          2240 non-null   object
 1   post             2240 non-null   object
 2   likes            2240 non-null   object
 3   comments         2240 non-null   object
 4   reposts          2240 non-null   object
 5   hashtags         2240 non-null   object
 6   post_lemmatized  2240 non-null   object
dtypes: object(7)
memory usage: 140.0+ KB
```

Из 9 тыс. постов, пригодных для использования, осталось чуть более двух тысяч.

2.3. Создаём датасет

Объединим датафреймы.

In [42]:
```
# переименуем столбец id в user_id в датафрейме profiles,
# для последующего объединения с posts
profiles = profiles.rename(columns={'id': 'user_id'})
```

In [43]:
```
# объединяем датафреймы
df = pd.merge(posts, profiles, on='user_id')
```

In [44]:
```
# удаляем дубликаты
df.drop_duplicates(inplace=True)
```

In [45]:
```
# удаляем из столбца likes точки, запятые и пробелы
df["likes"] = df["likes"].replace(r'\.|\,|\s', '', regex=True)

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

In [46]:
```
# смотрим что получилось
df.sample()
```

Out[46]:

	user_id	post	likes	comments	reposts	hashtags	post_lemmatized
994	munis-ashrapov	The novelty is an uninterruptible power system with double energy conversion and completes the filling of the three-phase series Vanguard II 33 (VGD II).Новинка представляет собой систему бесперебойного питания с двойным преобразованием энергии и завершает наполнение трёхфазной серии Vanguard II 33 (VGD II).For more details / Подробнее:Facebook: https://lnkd.in/dCvgqpyLinkedIn: https://lnkd.in/d2fNDdJTwitter: https://lnkd.in/duJzmpuInstagram: https://lnkd.in/dY5Wb3f#pixeltj #news #digitaltransformation #hardware #UPS #POWERCOM	2	0	0	pixeltj, news, digitaltransformation, hardware, UPS, POWERCOM	новинка система бесперебойный питание двойной преобразование энергия завершать наполнение трёхфазный серия

In [47]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2230 entries, 0 to 2239
Data columns (total 16 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   user_id           2230 non-null   object
 1   post              2230 non-null   object
 2   likes             2230 non-null   int64 
 3   comments          2230 non-null   object
 4   reposts           2230 non-null   object
 5   hashtags          2230 non-null   object
 6   post_lemmatized   2230 non-null   object
 7   user_name         2230 non-null   object
 8   user_head         2230 non-null   object
 9   user_work         2142 non-null   object
 10  user_position     2230 non-null   object
 11  user_tags         575 non-null    object
 12  user_location     2223 non-null   object
 13  user_viewers      2230 non-null   object
 14  user_contacts     2230 non-null   object
 15  user_common_info  2043 non-null   object
dtypes: int64(1), object(15)
memory usage: 296.2+ KB
```

In [48]:
```
# Сохраняем датафрейм
df.to_csv(os.path.join(DATA_PATH, 'linkedin.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,
- `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 фоловеров, т.е. других пользователей, отслеживающих активность данного пользователя,
- `user_contacts` - число контактов,
- `user_common_info` - информация пользователя о себе.

2.4. 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 [49]:
```
# проверим на дубли в post_lemmatized
df.post_lemmatized.duplicated().sum()
```

Out[49]:
```
106
```

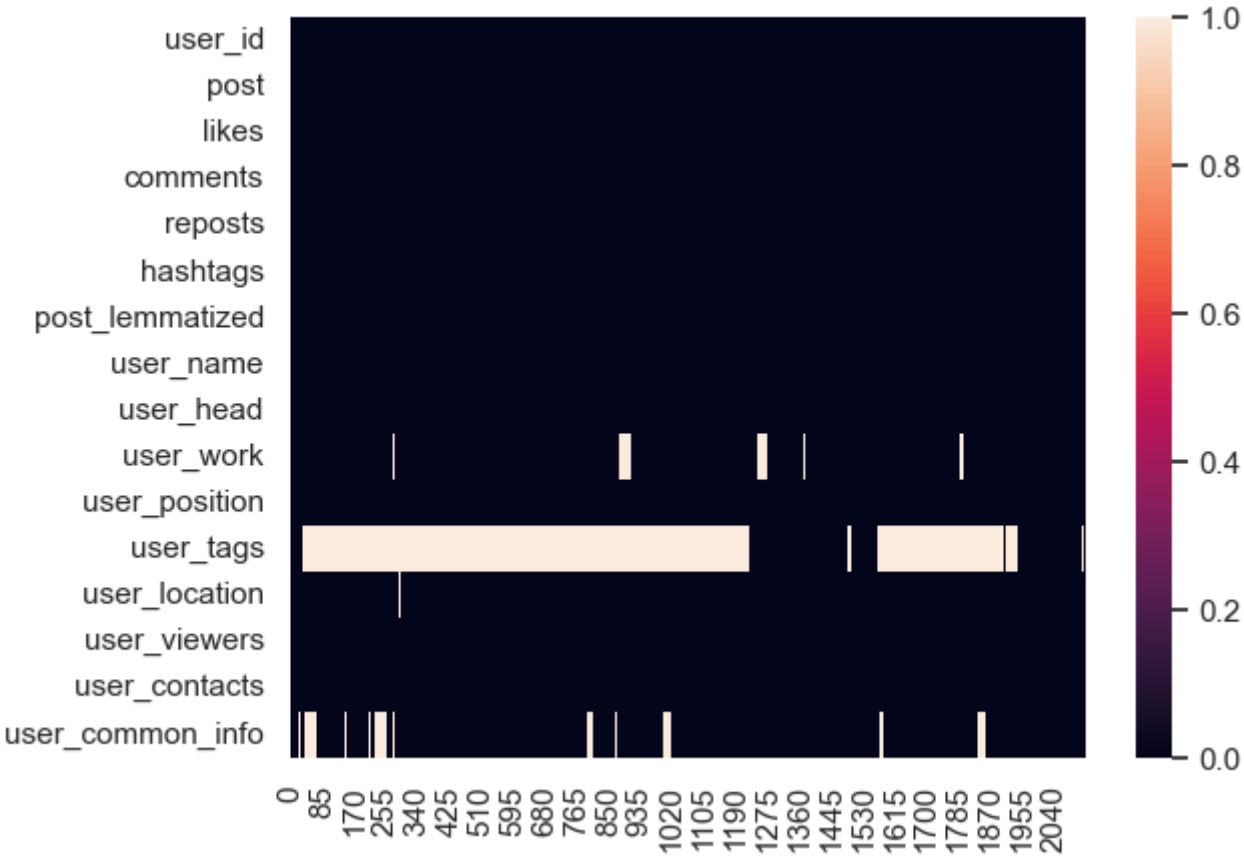
In [50]:
```
# удаляем дубликаты
df = df.drop_duplicates(subset='post_lemmatized', ignore_index=True)
```

In [51]:
```
# преобразуем тип данных
df[['comments', 'reposts']] = df[['comments', 'reposts']].astype(
    'int', errors='ignore'
)
```

In [52]:
```
# проверим изменения
df[['comments', 'reposts']].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2124 entries, 0 to 2123
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   comments  2124 non-null   int32
 1   reposts   2124 non-null   int32
dtypes: int32(2)
memory usage: 16.7 KB
```

In [53]:
```
# 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 [54]:
```
columns_to_fill = [
    'user_work', 'user_tags', 'user_location', 'user_common_info'
]

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

In [55]:
```
# проверим результат
df[columns_to_fill].isna().sum()
```

Out[55]:
```
user_work           0
user_tags           0
user_location       0
user_common_info    0
dtype: int64
```

In [56]:
```
# объединим пользовательские реакции в одну
df['reaction'] = df.likes + df.comments + df.reposts
```

In [57]:
```
# проверим содержимое поля числа фоловеров
df.user_viewers.unique()
```

Out[57]:
```
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', '1,118', '1,065',
       '2,443', '703', '2,831', '2,934', '1,179', '604', '10,401', '796',
       '313', '481', '8,893', '4,564', '2,003', '732', '29,597', '3,830',
       '1,981', '2,952', '4,482', '5,508', '882', '424', '1,686', '2,301',
       '3\xa0691', '1,488', '255', '3,115', '778', '5,300', '0', '21,858',
       '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'], dtype=object)
```

In [58]:
```
# оставим только числа
df.user_viewers = df.user_viewers.str.replace('[\D]', '', regex=True)

# изменим тип данных
df.user_viewers = df.user_viewers.astype('int')
```

In [59]:
```
# проверим содержимое поля числа контактов
df.user_contacts.unique()
```

Out[59]:
```
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', '309', '460', '419',
       '250', '96', '10', '396', '372', '305'], dtype=object)
```

In [60]:
```
# оставим только числа
df.user_contacts = df.user_contacts.str.replace('[\D]', '', regex=True)

# изменим тип данных
df.user_contacts = df.user_contacts.astype('int')
```

In [61]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2124 entries, 0 to 2123
Data columns (total 17 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   user_id           2124 non-null   object
 1   post              2124 non-null   object
 2   likes             2124 non-null   int64 
 3   comments          2124 non-null   int32 
 4   reposts           2124 non-null   int32 
 5   hashtags          2124 non-null   object
 6   post_lemmatized   2124 non-null   object
 7   user_name         2124 non-null   object
 8   user_head         2124 non-null   object
 9   user_work         2124 non-null   object
 10  user_position     2124 non-null   object
 11  user_tags         2124 non-null   object
 12  user_location     2124 non-null   object
 13  user_viewers      2124 non-null   int32 
 14  user_contacts     2124 non-null   int32 
 15  user_common_info  2124 non-null   object
 16  reaction          2124 non-null   int64 
dtypes: int32(4), int64(2), object(11)
memory usage: 249.0+ KB
```

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

2.5. 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 [62]:
```
# 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: 1126
```

In [63]:
```
# оставим только подходящие посты
df = df[keywords_filter]
```

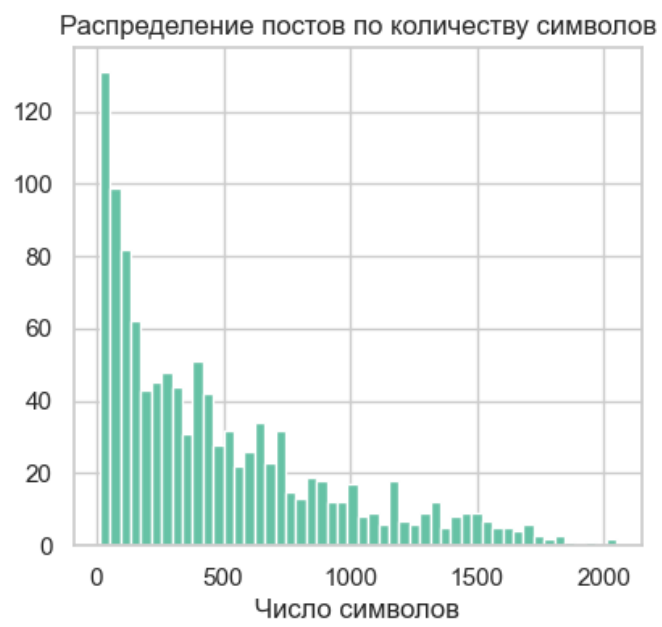
In [64]:
```
df.info()
```

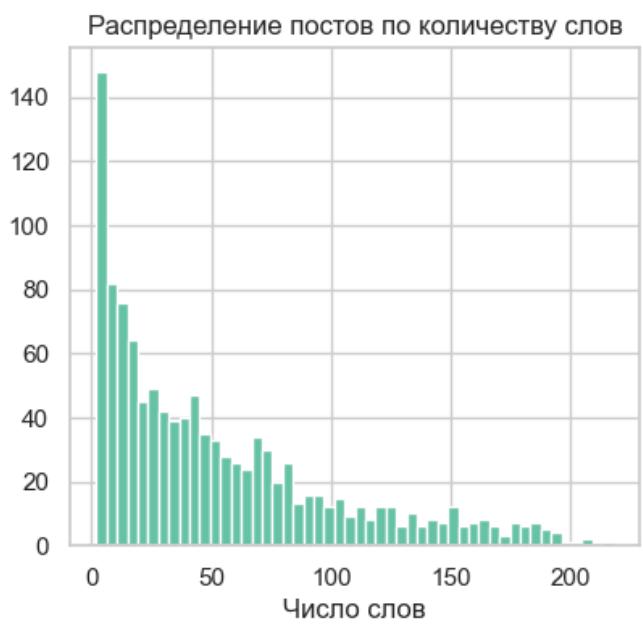
```
<class 'pandas.core.frame.DataFrame'>
Index: 1126 entries, 2 to 2123
Data columns (total 17 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   user_id           1126 non-null   object
 1   post              1126 non-null   object
 2   likes             1126 non-null   int64 
 3   comments          1126 non-null   int32 
 4   reposts           1126 non-null   int32 
 5   hashtags          1126 non-null   object
 6   post_lemmatized   1126 non-null   object
 7   user_name         1126 non-null   object
 8   user_head         1126 non-null   object
 9   user_work         1126 non-null   object
 10  user_position     1126 non-null   object
 11  user_tags         1126 non-null   object
 12  user_location     1126 non-null   object
 13  user_viewers      1126 non-null   int32 
 14  user_contacts     1126 non-null   int32 
 15  user_common_info  1126 non-null   object
 16  reaction          1126 non-null   int64 
dtypes: int32(4), int64(2), object(11)
memory usage: 140.8+ KB
```

Оценим размеры постов в количестве символов и количестве слов.

In [65]:
```python
# подсчет числа символов
def count_chars(text):
    return(len(text))

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

In [66]:
```python
# посчитаем статискику и постром 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');
```





In [67]:
```
# характеристики постов по символам
df.num_chars.describe()
```

Out[67]:
```
count    1126.000000
mean      478.726465
std       441.662632
min        11.000000
25%       119.000000
50%       349.500000
75%       705.000000
max      2049.000000
Name: num_chars, dtype: float64
```

In [68]:
```
# характеристики постов по словам
df.num_words.describe()
```

Out[68]:
```
count    1126.000000
mean       52.590586
std        48.386612
min         2.000000
25%        13.000000
50%        38.500000
75%        76.000000
max       218.000000
Name: num_words, dtype: float64
```

Большая часть постов короткие. Медианный размер поста 356 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 [69]:
```
# 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 [70]:
```
# число записей, попадающих под ограничения
len(df[chars_filter | words_filter])
```

Out[70]:
```
226
```

In [71]:
```
# оценим содержание мелких текстов
df.query(
```

```
    'num_chars < @min_chars and num_words < @min_words'
).post_lemmatized.head()
```

Out[71]:
```
6     эпизод подкаст теория рациональный выбор используют мешать рациональный
17                              статья технократия процесс изменений залетать
29                                                          дописать статью
31                                                 бизнес видеоигр интерактива
32                              обсуждение метод принятие оптимальный решение
Name: post_lemmatized, dtype: object
```

In [72]:
```
# удаляем короткие посты
df = df.query('num_chars >= @min_chars and num_words >= @min_words')
```

In [73]:
```
# оценка датасета после фильтрации
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 900 entries, 2 to 2123
Data columns (total 19 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   user_id           900 non-null    object
 1   post              900 non-null    object
 2   likes             900 non-null    int64 
 3   comments          900 non-null    int32 
 4   reposts           900 non-null    int32 
 5   hashtags          900 non-null    object
 6   post_lemmatized   900 non-null    object
 7   user_name         900 non-null    object
 8   user_head         900 non-null    object
 9   user_work         900 non-null    object
 10  user_position     900 non-null    object
 11  user_tags         900 non-null    object
 12  user_location     900 non-null    object
 13  user_viewers      900 non-null    int32 
 14  user_contacts     900 non-null    int32 
 15  user_common_info  900 non-null    object
 16  reaction          900 non-null    int64 
 17  num_chars         900 non-null    int64 
 18  num_words         900 non-null    int64 
dtypes: int32(4), int64(4), object(11)
memory usage: 126.6+ KB
```

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.

3. Моделирование

Складываем все лемматизированные тексты в один список.

In [74]:
```
docs = df["post_lemmatized"].tolist()
```

In [75]:
```
# первые пять элементов
docs[:5]
```

Out[75]:
```
['подкаст миражи платформах аудио инстаграме патреоне звуки музыка картинки аудиоцитат фильм формула любви марк захарова',
 'искать команда мидло продакт дизайнер основный линейка продукт маркетинг скил юай юикс райтер привести порядок интерфейсный тексты английский русский опыт способность глубоко разбираться технический деталь переводить айтишный человеческий условие классноподробности вилка условие почта',
 'команда редизайн развитие продукт промо продукт сложный веб приложение сопровождениеили веб приложение основа дизайн системыразработка поддержка кроссплатформенный дизайн системымаркетинговый задачи лэндинги письма оптимизация конверсия сайтовконтроль качество ожидаем опыт разработка интерфейс студия продуктовый летминимум живой дизайн минимум сложныйумение желание вникать разбираться умение основной инструмент возможный перейти фигма точно умение понадобиться предлагаем белый заработный платакомфортный офис минута ходьба тульскаяспортзал настольный теннис занятие тренером группа английский китайский офисесовременный рабочий мощный маки монитоpыдмс испытательный срок отклик почта',
 'профессия менеджер часами поработать позиция взаимодействовать зрение подчинённого руководителя заказчик исполнителя поделиться мысль обменяться видение профессия восприятие людьми',
 'запретный плод сладок удивлюсь посещаемость выросла зато соц сеть крайний мера айтишник точно знают сюда']
```

3.1. Векторизация текстов

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

In [76]:
```
# создаем модель векторизации
tfidf = TfidfVectorizer(min_df=20, max_df=0.9)
```

In [77]:
```
%%time
# обучим модель и получим векторное представление для каждого текста
x = tfidf.fit_transform(docs)
```

```
CPU times: total: 15.6 ms
Wall time: 34 ms
```

In [78]:
```
# размер полученной матрицы
x.shape
```

Out[78]:
```
(900, 424)
```

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

In [79]:
```
# список слов векторизатора
tf_feature_names = tfidf.get_feature_names_out()
tf_feature_names
```

Out[79]:
```
array(['активно', 'активный', 'актуальный', 'анализ', 'аналитик',
       'аналитика', 'английский', 'база', 'банк', 'безопасность',
       'бизнес', 'бизнеса', 'благодаря', 'близкий', 'бренд', 'будущее',
       'быстро', 'быстрый', 'важно', 'вариант', 'веб', 'вести',
       'взаимодействие', 'вид', 'видео', 'видеть', 'включать',
       'внедрение', 'внешний', 'внимание', 'внутренний', 'внутри',
       'возможность', 'возможный', 'вообще', 'вопросы', 'времени',
       'выбирать', 'выбор', 'выбрать', 'выполнять', 'выше', 'гибкий',
       'году', 'голова', 'график', 'группа', 'давать', 'далее',
       'дальнейший', 'данные', 'действие', 'делиться', 'дело', 'деталь',
       'деятельность', 'дизайн', 'дизайнер', 'директор', 'добавить',
       'довольно', 'долгий', 'долго', 'должность', 'достаточно',
       'достигнуть', 'достижение', 'доступ', 'доступный', 'думаю',
       'желание', 'жизни', 'зависеть', 'зависимость', 'задавать',
       'задание', 'задать', 'задач', 'задача', 'задачи', 'заказ',
       'заказчик', 'заниматься', 'занятие', 'запрос', 'запуск',
       'запускать', 'запустить', 'заработный', 'заявка', 'знание', 'знаю',
       'идея', 'изменение', 'изучать', 'изучение', 'инструмент',
       'интеграция', 'интервью', 'интернет', 'интерфейс',
       'информационный', 'информация', 'искать', 'использование',
       'исследование', 'история', 'итог', 'казаться', 'канал', 'кандидат',
       'карьера', 'карьерный', 'качественный', 'качество', 'кейс',
       'клиент', 'клиентов', 'ключевой', 'книга', 'код', 'команда',
       'команды', 'комментарий', 'коммерческий', 'коммуникация',
       'компании', 'компаний', 'компанию', 'конкретный', 'консультация',
       'контакт', 'контент', 'контроль', 'корпоративный', 'крупный',
       'курс', 'лайк', 'лично', 'лишь', 'люди', 'людьми', 'магазин',
       'маркетинг', 'материал', 'международный', 'менеджер', 'менеджмент',
       'минимум', 'мнение', 'множество', 'мобильный', 'модель',
       'мотивация', 'мысль', 'набор', 'навык', 'назад', 'наиболее',
       'найти', 'наличие', 'написание', 'направление', 'насколько',
       'научиться', 'начало', 'начинающий', 'небольшой', 'неделю',
       'некоторый', 'необходимый', 'ниже', 'нравиться', 'обеспечение',
       'область', 'облачный', 'оборудование', 'образование', 'обсудить',
       'обучение', 'общаться', 'общение', 'общий', 'объём', 'обязанности',
       'огромный', 'однако', 'около', 'онлайн', 'описание', 'опыт',
       'опыта', 'опытный', 'опытом', 'организация', 'основа', 'основной',
       'основный', 'ответственность', 'отвечать', 'отклик', 'открывать',
       'открытый', 'относиться', 'отношение', 'отправить', 'отправлять',
       'офис', 'официальный', 'оценка', 'очередь', 'ошибка', 'пакет',
       'партнёр', 'перейти', 'период', 'персональный', 'план',
       'планировать', 'плата', 'платформа', 'плюс', 'повысить',
       'повышение', 'подготовка', 'поддерживать', 'поддержка',
       'поделиться', 'позволить', 'позиция', 'поиск', 'пойти', 'получить',
       'получиться', 'пользователей', 'пользователь', 'помогать',
       'помощь', 'понимание', 'понимать', 'понятный', 'понять',
       'пообщаться', 'попробовать', 'портфолио', 'постоянный',
       'потенциальный', 'потребность', 'почта', 'появиться', 'правило',
       'правильный', 'практика', 'практический', 'предлагать',
       'предложение', 'предыдущий', 'привести', 'приводить', 'прийтись',
       'приложение', 'приложений', 'приложения', 'применение', 'примерно',
       'принимать', 'принцип', 'принять', 'приходить', 'причина',
       'проблема', 'проведение', 'провести', 'программа',
       'программирование', 'программный', 'продавать', 'продажа',
       'продолжение', 'продукт', 'продукта', 'продуктовый', 'проектами',
       'проектный', 'проектов', 'проекты', 'происходить', 'простой',
       'профессиональный', 'профиль', 'проходить', 'процесс', 'прочитать',
       'путь', 'пытаться', 'работе', 'работу', 'рабочий', 'развивать',
       'развиваться', 'развитие', 'развития', 'различный',
       'разрабатывать', 'разработать', 'разработка', 'разработки',
       'разработчик', 'разработчиков', 'ранний', 'расти', 'реализация',
       'реализовать', 'реальный', 'результат', 'резюме', 'рекомендация',
       'рекомендовать', 'ресурс', 'решать', 'решение', 'решения', 'роль',
       'россия', 'руководитель', 'рынок', 'самостоятельно', 'связь',
       'сервис', 'сеть', 'сила', 'сильный', 'система', 'системный',
       'ситуация', 'следовать', 'слово', 'сложно', 'сложный', 'случай',
       'смена', 'смысл', 'собеседование', 'собрать', 'совет',
       'современный', 'создавать', 'создание', 'составить', 'составлять',
       'сотрудник', 'сотрудников', 'специалист', 'специалистов', 'список',
```

```
       'способ', 'срок', 'становиться', 'стартап', 'статья', 'стоимость',
       'сторона', 'страна', 'страница', 'стратегия', 'строить', 'студент',
       'сфера', 'счёт', 'текст', 'текущий', 'тема', 'теория', 'тест',
       'тестирование', 'тестовый', 'техника', 'технический', 'технологии',
       'технологий', 'технология', 'тип', 'топ', 'точно', 'требования',
       'требовать', 'труд', 'увидеть', 'удалённый', 'уйти', 'улучшение',
       'улучшить', 'умение', 'уметь', 'управление', 'управлять',
       'уровень', 'условие', 'услуга', 'успех', 'успешный', 'участие',
       'участник', 'учиться', 'факт', 'финансовый', 'формат',
       'формирование', 'функция', 'хотя', 'целое', 'цель', 'центр',
       'часто', 'частый', 'чувство', 'чувствовать', 'шаг', 'эксперт',
       'электронный', 'этап', 'эффективность', 'эффективный', 'яндекс'],
      dtype=object)
```

In [80]:
```
# словарь
id2word = {i: token for i, token in enumerate(tf_feature_names)}
```

In [82]:
```
# примеры слов в словаре
id2word[0], id2word[1], id2word[2], id2word[200], id2word[420]
```

Out[82]:
```
('активно', 'активный', 'актуальный', 'отклик', 'этап')
```

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 [83]:
```
# параметры
n_topic_list = [10, 15, 20] # число тем
iter_list=[50, 100, 150] # число итераций
```

In [84]:
```
%%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: 1.3 s
Wall time: 14.9 s
```

In [85]:
```
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[85]:

Сравнительная таблица качества моделирования

	n_topics	max_iter	lda_score
0	10	50	-24700.275130
1	10	100	-24687.073079
2	10	150	-24687.073091
3	15	50	-25454.894627
4	15	100	-25454.895015
5	15	150	-25454.895015
6	20	50	-26814.154998
7	20	100	-26812.608458
8	20	150	-26812.608468

Минимальное значение lda_score при n_topics = 10 и max_iter = 100. Эксперимент показал, что с увеличением числа топиков, скор ухудшается, а увеличение числа итераций на скор влияет незначительно.

Получим модель с указанными параметрами.

In [86]:
```
%%time

# число тем
n_topics = 10
n_iters = 100

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

lda_topics = lda.fit_transform(x)
```

```
CPU times: total: 2.36 s
Wall time: 3.2 s
```

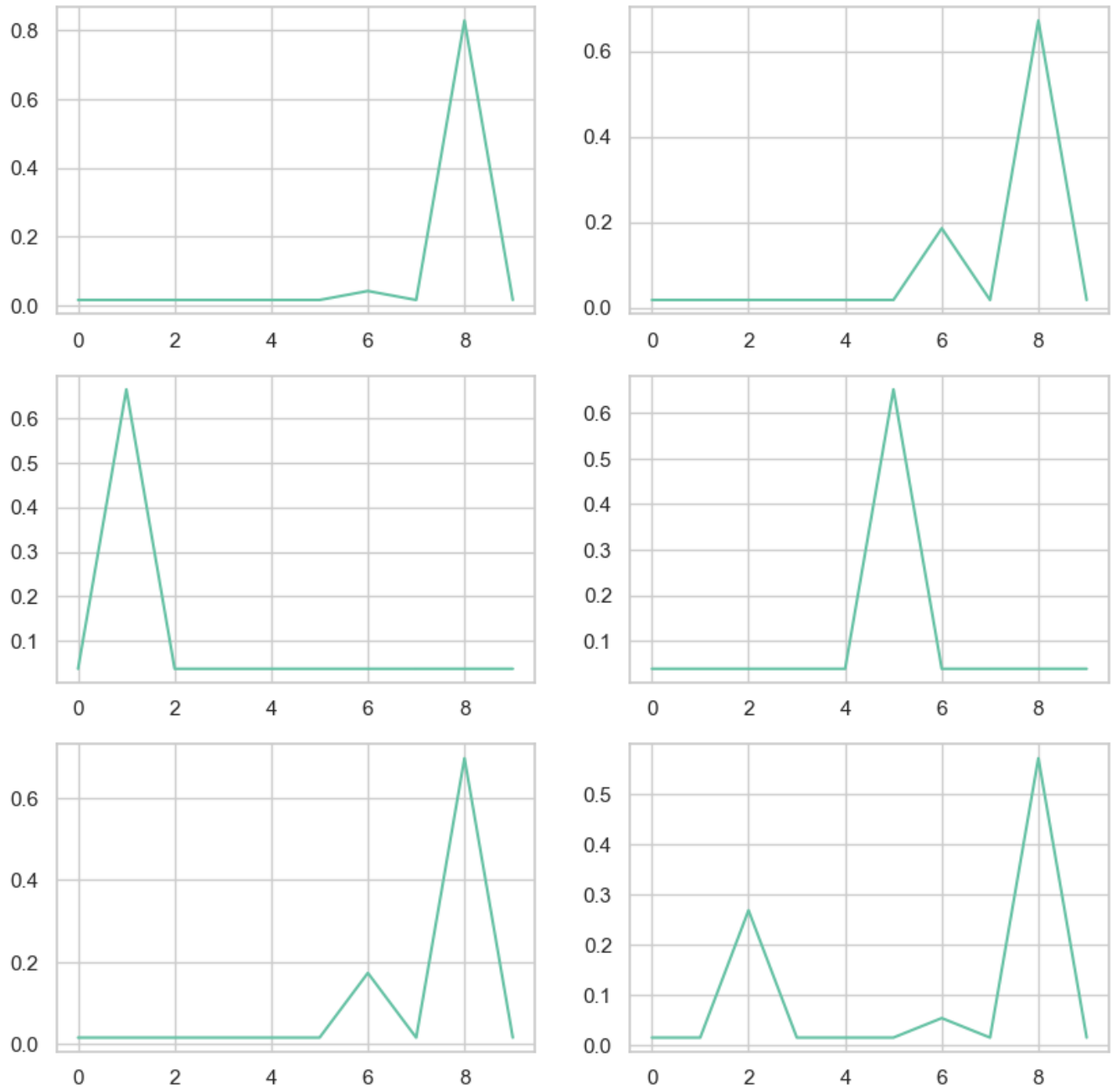
In [87]:
```
# размер полученной матрицы
lda_topics.shape
```

Out[87]:
```
(900, 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 [88]:
```
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.

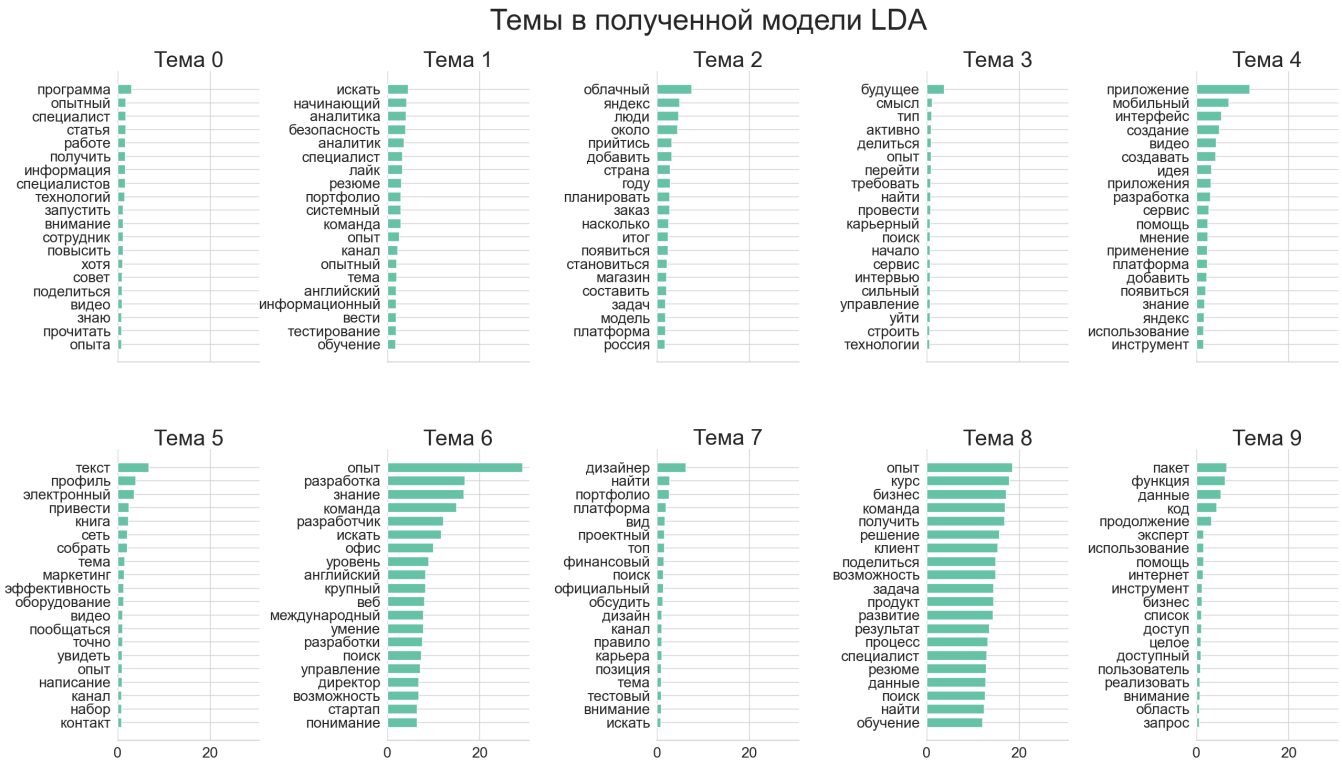
Ключевые слова

Теперь извлечём ключевые слова для каждой из тем.

In [89]:
```
# 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 [90]:
```
# число ключевых слов в теме
n_top_words = 20

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



Темы 6 и 8 выделятюся от остальных наибольшими значениями вероятности для ключевых слов.

Интерпретация тем для LDA

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

- Тема 0: "Программа для опытных специалистов: получение информации и обмен опытом"
- Тема 1: "Карьера начинающего аналитика безопасности: поиск работы, опыт и обучение"
- Тема 2: "Развитие облачной платформы Яндекс в России: рост, задачи и появление магазина"
- Тема 3: "Будущее карьерного поиска: активное деление опыта, переходы и требования в управлении технологиями"
- Тема 4: "Создание и разработка мобильных приложений: идеи, интерфейс и использование инструментов"
- Тема 5: "Эффективность электронного текста и видео в маркетинге: сбор информации, контакт и оборудование"
- Тема 6: "Развитие профессионального опыта в разработке: уровень знаний, командная работа и поиск возможностей"
- Тема 7: "Поиск и развитие карьеры дизайнера: портфолио, позиции и внимание к дизайну"
- Тема 8: "Развитие карьеры и обучение в бизнесе: опыт, команда и поиск возможностей"
- Тема 9: "Использование пакетов и инструментов для реализации функций и обработки данных в бизнесе"

Типичные статьи

In [93]:
```
for i in range(n_topics):
    doc_id = np.argmax(lda_topics[:, i])
    print("Тема ", i)
    print(df.iloc[doc_id]["post"])
    print("\n")
```

Тема 0
Как начать карьеру в игровой индустрии, разрабатывая свою игру или работая на компанию своей мечты?В этом видео https://lnkd.in/dxN-5ge я рассмотрю способы попадания в геймдев, как для опытных специалистов так и для новичков без опыта. Кстати, многие из советов применимы также в других сферах, так что взглянуть стоит даже, если ты ищешь работу в другой сфере.#геймдев #поискработы #игроваяиндустрия

Тема 1
Продакт менеджер или интересуешься ITПриглашаем тебя на онлайн вебинар с Senior Product Manager, где будем обсуждать: Технические навыки специалиста Опыт становления Senior Product Manager за 2 года Каких продактов ищут компании Первые 90 дней в качестве продакта в новой компании иЖдем тебя 4 и 11 мая в 19:30Переходи по ссылке в актуальных в Telegram канал, мы отправим ссылку на вебинар в чатhttps://t.me/product_bee

Тема 2
Ровно два месяца назад меня депортировали из Грузии. Без причины, объяснения ситуации, просто посадили обратно на самолет до Турции. Сидя тогда в Стамбуле я долго думал как вьехать обратно, в итоге логичным решением было поехать в Сербию, страну в которой я успел прожить год в далеком 2018-ом. Балканы встретили с душой, распростертыми объятиями и вкусной едой. Продолжаю интегрироваться в "новое" место, налаживать связи в местном IT сообществе. Планирую серию коротких постов о жизни на Балканах. #serbia #it #relocation

Тема 3
Делюсь своим мотивационным профилем. Краткая расшифровка по типам мотивации: инструментальный тип — самостоятельность, сильная направленность на материальную составляющую; профессиональный — свобода действий, нацеленность на профессиональное развитие; патриотический — общественное признание, достижение сверхидеи; хозяйский — инициативность, лидерские задатки, потребность в карьерном росте; люмпенизированный — отсутствие инициативы, избирательность в труде.Тест можно пройти по ссылке https://lnkd.in/dB5Z4keS #resume #Motype #motivated

Тема 4
Привет, меня зовут Алексей, я опять или снова исследую удобство сервиса Яндекс.Музыка. На основании исследования, участники неоднократно помогут улучшить сервис, сделать его удобным и лучше. Ты мне подходишь, если:1. Слушаешь музыку или подкасты через приложение Яндекс.Музыка2. Пользуешься другими похожими сервисамиНапиши мне с запросом на участие, мы определимся когда будет тебе удобно, интервью будет проходить через Zoom. Поскольку мы тестируем приложение Яндекс.Музыка, необходимо будет предварительно установить Zoom на телефон. Перед началом интервью я объясню все детали подробней.

Тема 5
7 recruitment companies in IsraelI've been searching through job openings on Telegram channels lately, and in one of them, called "high-tech jobs for olim" (https://lnkd.in/d6inhwan), I saw a recommendation about an additional channel for job seekers - recruitment agencies. They even published a list of such companies in Israel:cps.co.ilJobinfo.co.ilNisha.co.ilDialog.co.ilGotfriends.co.ilEthosia.co.ilurbanrecruits.co.ilOf course, I signed up with all of them, but I'm not sure if could work effectively for juniors...Have you ever had experience working with recruiting agencies? What do you think, could it be helpful / can lead to results??----------Сегодня просматривала телеграм-каналы, в которых публикуются актуальные вакансии. В одном из них под названием "high-tech jobs for olim" (https://lnkd.in/d6inhwan) увидела идею с дополнительным каналом для поиска работы - рекрутинговые агентства. Ребята опубликовали список таких компаний в Израиле:Cps.co.ilJobinfo.co.ilNisha.co.ilDialog.co.ilGotfriends.co.ilEthosia.co.ilurbanrecruits.co.ilЯ, конечно, в каждой из них зарегистрировалась, но не уверена, насколько это будет эффективно для джуна...Расскажите, был ли у вас опыт работы с рекрутинговыми агентствами? На ваш взгляд, это полезно/может привести к результатам?----------#jobsearch #recruitement #jobs #recruiting #jobseekers #joboopenings #tech

Тема 6
#москва #офис #фултаймИщу классных коллег к нам в команду AdGuard (adguard.com) Что делать:Редизайн и развитие отдельного продукта (промо-сайт + продукт, сложное веб-приложение) и сопровождениеилиUX/UI для веб-приложений и ПО с нуля и на основе дизайн-системыРазработка и поддержка кроссплатформенной дизайн-системыМаркетинговые задачи, лэндинги, письма, оптимизация конверсии сайтовКонтроль качества Что мы ожидаем:Опыт разработки интерфейсов в студии или продуктовых компаниях от 2 летМинимум 2 живых проекта с вашим дизайном в сети, минимум 1 из них сложн

ыйУмение и желание вникать и разбираться, умение думатьSketch как основной инструмент (возможно со временем перейдем на фигму но это не точно) + умение работать на всем что понадобится Что предлагаем:Белая заработная платаКомфортный офис в г. Москве в 5 минутах ходьбы от м. ТульскаяСпортзал, настольный теннис, занятия с тренером, группы по английскому и китайскому языкам в офисеСовременные и удобные рабочие места — мощные маки, хорошие мониторыДМС после испытательного срока Для вопросов и откликов пишите на почту hr@performix.ru или в телеграм @ksenia_hr

Тема 7

Сегодня, завершается регистрация на топ-интенсив "Политика в проектном управлении"!Приглашаю тебя присоединиться к разговору о политике как о своде негласных правил и норм, которые невидимо управляют всеми бизнес-процессами.Тема интенсива полезна абсолютно каждому, кто хочет не просто работать, а работать успешно и достигать поставленных целей.Разбор реальных кейсов, в том числе из твоей практики, наглядно покажет важность и ценность понимания, что политика существует в любом коллективе.Новые знания - это возможность пересмотреть свой подход к построению карьеры и достижению успеха!16 декабря в 19.00 мск интенсив "Политика в проектном управлении"Продолжительность 2 часа, онлайн.Регистрация на сайте https://lnkd.in/gmjvpfV Стоимость участия: 5 000р.Готова ответить на все вопросы в ЛС.До встречи!#консалтингвуправлении #управление #бизнес #управлениепроектами #проектноеуправление #развитиебизнеса #развитиекоманд

Тема 8

Почему-то в среде айтишников существует устойчивое мнение, что 1с-ники - это недопрограммисты. И мне это очень обидно слышать, как человеку, который посвятил этой профессии 10 лет. Я хотела хотела бы немного развеять этот миф. 1С специалист - это Fullstack разработчик. Именно так. 1С программисты работают с базой данных, проектируют интерфейсы, тестируют программный продукт, формируют и принимают файлы различного формата для обмена данными, работают с системой контроля версий, помимо этого сотрудничают с пользователями, работая как аналитик и call центр. И этого мало. Чтобы быть хорошим 1с специалистом, 1с-ник должен отлично знать законодательство и прогнозировать возможные проблемы в учёте в своей организации, ведь зачастую программа на предприятии допиливается под него. Также, основываясь на бизнес-процессах организации, 1с-внедренец должен подобрать идеальное программное обеспечение и настроить его под пользователей, обучить и направить. Помимо этого есть ещё такая вещь, как регулярно приходящие баги с обновлениями, а так же неверный результаты работы программы, по мнению бухгалтерии, при расчете себестоимости готовой продукции или зарплаты, например. И вот тут нужно очень хорошо уметь разбираться в очень сложном коде и если нужно - исправлять. И вот тогда, в отладке, познается, почему у 1С нет реальных конкурентов на рынке. Вы не задумывались, почему в такой прибыльной нише, в которой можно зарабатывать огромные деньги, до сих пор нет реального конкурента 1С и все предприятия, которые собираются ставить учет, выбирают эту программу? Чаще всего все описанные функции ложатся на плечи одного человека, это полная ответственность за принятые решения, интересные задачи с каждым годом и с каждым новым законом или управленческим решением.К чему я это всё. Мой муж долгое время работал 1С разработчиком, дорос до синьора. Большую часть времени - в одном холдинге. Учитывая всё вышеописанное, ему не было скучно и он не застрял на одном месте без развития, не тот случай.А сейчас я слышу мнение, что 1С-нику не по плечам стать фронтендером. Серьёзно? Тем более мне непонятно, когда ставят в упрек преданность одной компании. Не всегда нужно менять компании каждые полтора года чтобы развиваться.Мне не верится что на всём огромном рынке труда не найдётся место для фронтендера, который пишет на реакте, знает кучу технологий и умеет такие скиллы как у моего мужа. Я его знаю лучше других и поверьте это искренняя рекомендация. Если вам нужен frontend react developer, который точно знает чего он хочет, самостоятелен и устойчив, напишите, пожалуйста, Юрию. Yuri Koshelev Всем спасибо за лайки и репосты.

Тема 9

Манипуляция данными с помощью SQL запросов в RB целом я не особо приветствую использования SQL, внутри / вместо R, т.к. функционал самого R гораздо богаче. Тем не менее знать о такой возможности стоит. Ранее для манипуляции данными с помощью SQL запросов зачастую использовали пакет sqldf, который последний раз обновлялся ещё в далёком 2017 году. На смену устаревшему пакету пришел новый - tidyquery.Данный пакет содержит всего 2 функции: query() - реализует манипуляцию данными с помощью SQL запросов show_dplyr() - транслирует ваш SQL запрос в dplyr кодПродолжение с примерами кода по ссылке: https://lnkd.in/dmwpUGRQ#R #SQL #DataScience

Сохраним в датафрейм номер наиболее вероятной темы для каждого поста.

In [94]:
```
# значения наиболее вероятных топиков
df['lda_topic'] = np.argmax(lda_topics, axis=1)
```

Вывод:

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

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

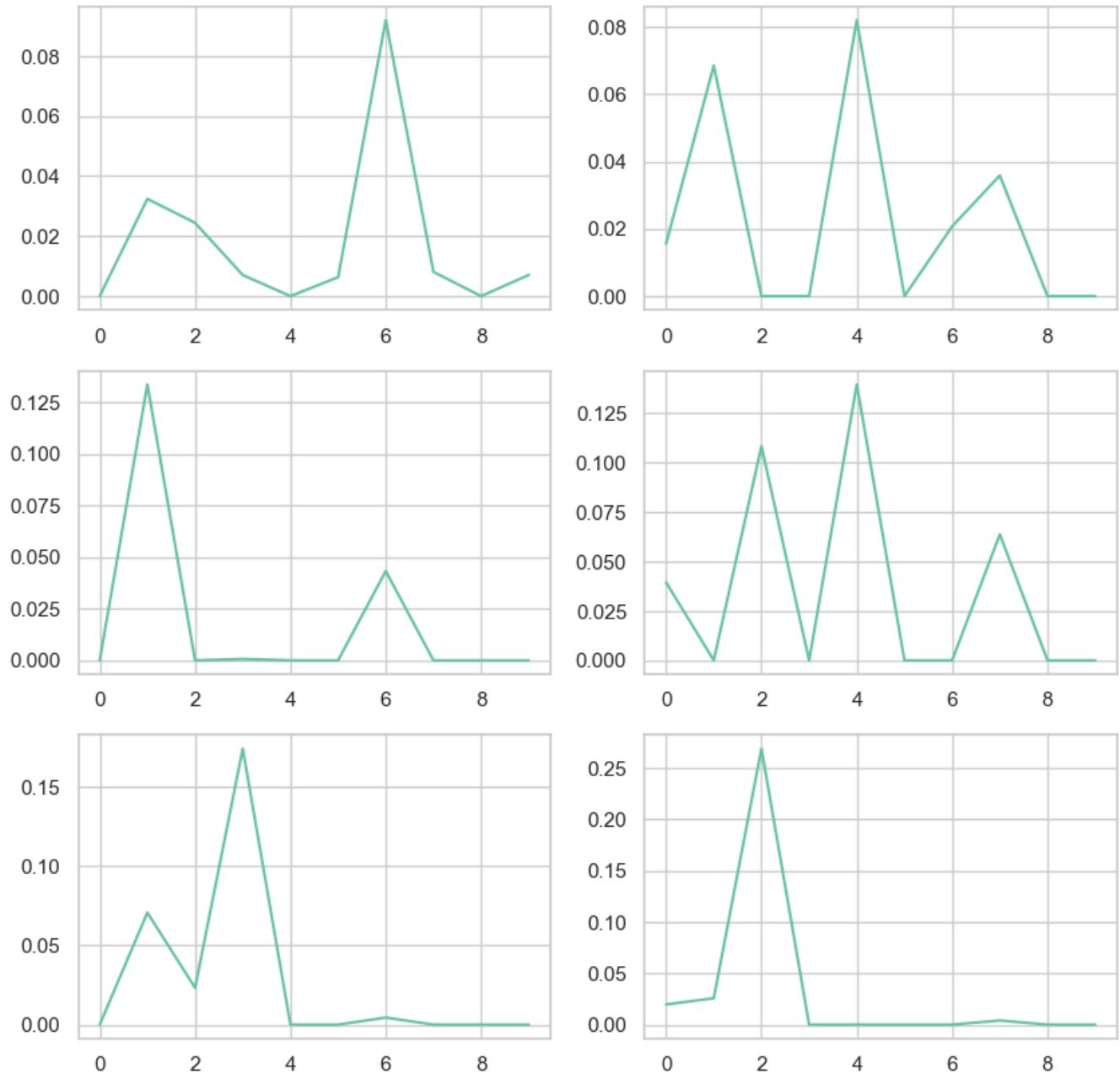
3.3. NMF

Неотрицательная матричная факторизация (NMF).

In [97]:
```
%%time

# число тем
n_topics = 10
n_iters = 300

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

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

```
CPU times: total: 15.6 ms
Wall time: 27 ms
```

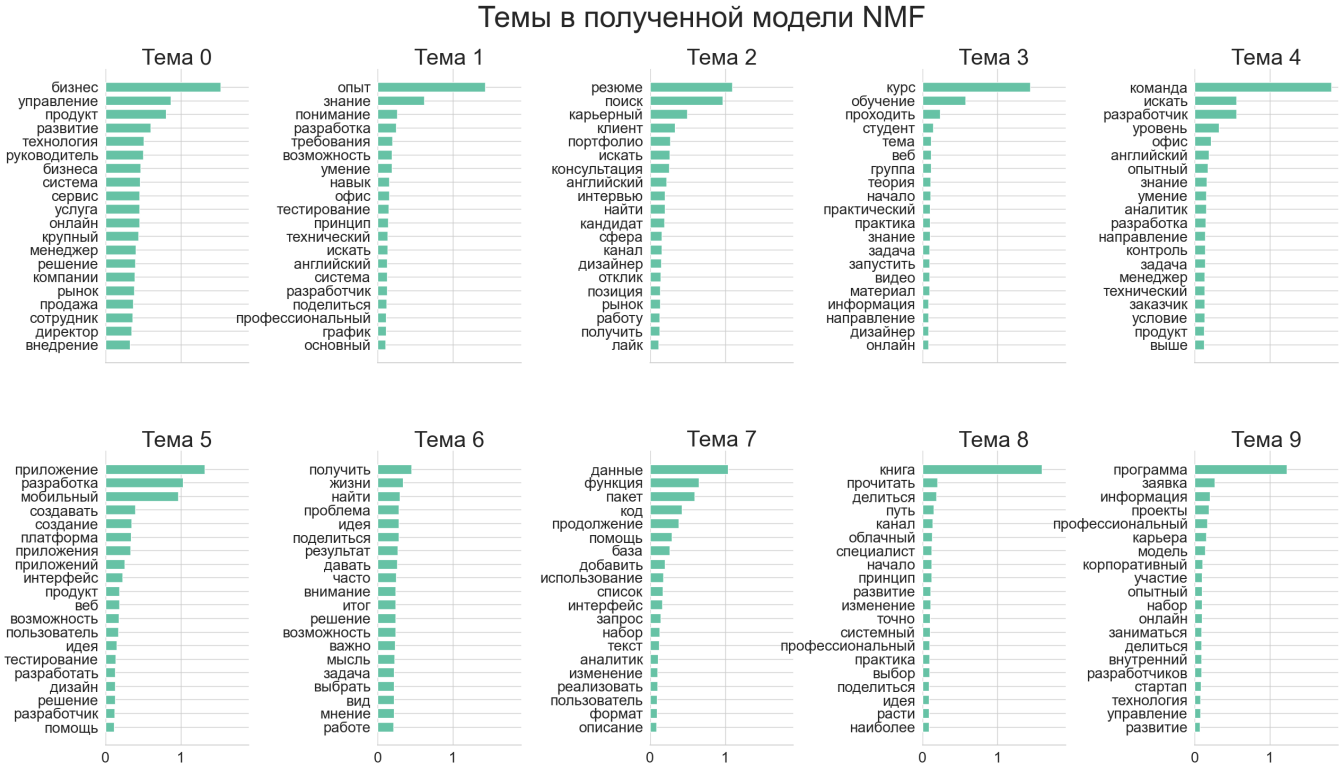
In [98]:
```
# графики полученных вероятностей принадлежности текста к топикам
plt.figure(figsize=(10,10))
for i in range(6):
    idx = np.random.randint(0, nmf_topics.shape[0])
    plt.subplot(3, 2, i+1)
    plt.plot(nmf_topics[idx])
```



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

Ключевые слова

In [99]:
```
# 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: "Управление бизнесом и развитие продукта в эпоху технологий"
- Тема 1: "Развитие профессиональных навыков и требования в разработке программного обеспечения"
- Тема 2: "Поиск работы в сфере дизайна и карьерное развитие"
- Тема 3: "Онлайн-курсы и обучение веб-дизайну: теория и практика для студентов"
- Тема 4: "Формирование эффективной команды разработчиков для успешной разработки технического продукта"
- Тема 5: "Разработка мобильных и веб-приложений: создание продукта с удобным интерфейсом и помощь разработчиков"
- Тема 6: "Важность обмена идеями и мнениями в работе: получение результатов и решение проблем"
- Тема 7: "Управление данными и их использование: функции, код и аналитика"
- Тема 8: "Профессиональное развитие и обмен знаниями: книги, практика и облачные каналы"
- Тема 9: "Управление профессиональным развитием и участие в проектах: программы, информация и деление опытом"

Типичные статьи

In [105…

```
# 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
Результативное и эффективное управление проектами принято считать стратегической компетенцией в организации.Эта компетенция позволяет компании: соединять результаты с бизнес-целями; более успешно конкурировать на своих рынках; добиваться большей устойчивости своей организации; реа гировать на воздействие изменений бизнес-среды, с помощью корректировки планов управления про ектами#управление #управлениепроектами

Тема 1
Всем привет! :) Наша команда ищет Devops-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: 200-250 тыс. р. на руки.Какой опыт требуется:•Понимание основных принципов и подходов IaC, методологии DevOps;•Опыт работы с Kub ernetes, Helm, Docker, Containerd;•Опыт работы и реализации решений для сборки и деплоя (Gitl ab CI, Teamcity, Octopus Deploy);•Опыт работы с системами Configuration Management (Ansible, Chief, Puppet);•Опыт настройки и поддержании систем мониторинга, логирования и визуализации (Zabbix, стек ELK, стек Prometheus -Grafana);•Понимание принципов работы сетевых протоколов;• Опыт написания запросов на SQL (TSQL как плюс);•Опыт написания автоматизаций на Bash, Python; •Опыт работы с Git;•Опыт взаимодействия с другими командами разработки, локализации и устране ния проблем;Будет плюсом, но не обязательно:•Опыт/понимание принципов работы высоконагруженны х/высокодоступных систем;•Опыт работы с Keycloak, Consul;Опыт работы с системами виртуализаци и (VMware, Proxmox);•Опыт работы с Windows системами; Компания предлагает вам:- Рабочую техни ку при необходимости : ноутбук, монитор и т.д.;- ДМС (или спорт) после испытательного срока;- Оплачиваемые профильные внешние курсы, а также доступ к внутренним учебным программам.- Возмо жности профессионального роста и развития.Лучше сразу приходите в telegram: @tatberezkaБуду р ада ответить на все вопросы и рассказать про детали :)#вакансия #devops #middle #удаленка #fu lltime #remote

Тема 2
Как составить резюме для работы за границей? https://lnkd.in/eSd3QMdxB этом видео я разберу р езюме Junior Data Scientist, 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]0:00 - Вступление0:19 - Как участвовать в разборе резюме0:25 - Что писать в разделе о себе2:55 - Как заполнять раздел с контактами3:57 - Как прописывать ключевые слова4:40 - Как у казывать языки для общения4:55 - Как отправлять резюме в разных странах напрямую5:28 - Как оп исывать достижения и обязанности9:10 - Как проходить под требования об опыте работы? Где брат ь дополнительный опыт10:48 - Какие шаблоны для резюме лучше всего использовать для поиска раб оты заграницей11:22 - Как настроить доступ к резюме11:53 - Как проверить корректность текста в резюме на английском13:20 - Что влияет на получение работы кроме резюме13:40 - Что делать, если не получается найти работу15:55 - Где можно задать мне вопрос про поиск работы, чтобы по лучить развернутый ответ

Тема 3
Продолжаю собирать полезную информацию из группы в Telegram "Работа ищет аналитиков".Часть 3. Машинное обучение:- Курс Andrew Ng на Курсере по ML.https://lnkd.in/di_Cf6wG- Курс на хабре о т ODS (ряд статей с примерам, задачам и кодом)https://lnkd.in/drjMPZ-a (Советую)- Курс от Goo gle https://lnkd.in/d67M2bQu- Курс Deep learning School от МФТИhttps://lnkd.in/drTb3Eza- Курс Applied Machine Learning Cornell Tech https://lnkd.in/dBj6yzE6Линейная алгебра:- Курс линейно й алгебры. Преподает легендарный профессор Массачусетского технологического института Гилберт Стрэнг. https://lnkd.in/dfHK7uYU- YouTube канал "3blue1brown" поможет понять линейную алгебр у https://lnkd.in/d6Ju4EcM. (Проверено, хороший ресурс).#machinelearning

Тема 4
Ценю командную работу. Поясню почему: В споре рождается истина, а в команде реализовываются г ениальные идеи! Команда – это единый организм, объединенный общей целью В команде объединяетс я группа людей, суммарная мощность и потенциал команды выше, чем у человека Командная работа регулирует усилия и умножает эффект Команда усиливает лидера, развивает компетенции и миними зирует его слабости (замещает их) В команде формируется больше идей и вариантов решений При к омандном подходе обеспечивается разнообразие взглядов Способы достижения конечной цели более разнообразны, чем при взгляде одного человека Победа делится на всех участников и проигрыш де лится на всех Команда проживают все человеческие эмоции, укрепляя и сплачивая коллектив Коман ды поддерживают в лидерах чувство ответственности за общее дело, потому что он ведет ее к рез ультату Команда может сделать больше, чем один человек! Командные чувства – движущая сила про екта #проектноеуправление #команды #работавкоманде

Тема 5

Тренды в мобильной разработкеСовременный 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 технологийСовременные мобильные приложения становятся все более сложными и функциональными, что требует развития технологий. Одной из главных тенденций является развитие и усовершенствование cross-platform-технологий, таких как React Native, Xamarin и Flutter, которые позволяют создавать мобильные приложения для нескольких платформ одновременно.Разработка без кодаОдной 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 обучениеИскусственный интеллект и машинное обучение являются ключевыми направлениями развития мобильной разработки в настоящее время. Многие компании уже внедрили AI-технологии в свои приложения, например, голосовые помощники и распознавание текста. Кроме того, машинное обучение позволяет создавать персонализированные рекомендации и улучшать пользовательский опыт.БезопасностьС 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 вещейС 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. #ai #react #nocode #lowcode #мобильныеприложения #разработка #новыетехнологии #безопасностьданных #интернетвещей #разработкабезкода

Тема 6

Софт скилы для менедежера / лида команды. Часть первая. Я часто вижу посты о хард скиллах для менеджера/лида команды, а вот софтовые вещи встречаю редко. О них мало кто говорит, хотя они важны не меньше. По этой причине я решил поделиться несколькими рекомендациями по этой теме, так как сам часто выступал на разное количество аудитории и вообще много общался с людьми за 26 лет жизни. Начать хочу с воодушевления. Говорить с воодушевлением — почему это важно? Умение правильно и красиво говорить для проектного менеджера открывает много дверей и возможностей в управлении командой. Воодушевление делает твою речь живой, помогает удержать интерес команды и пробуждает в них желание действовать. Прежде всего, если то, о чем ты говоришь, воодушевляет тебя самого, то и слушатели не останутся равнодушными. Научиться говорить с воодушевлением может каждый независимо от своего характера. Для этого обрати внимание на следующие моменты: 1. Говори с чувством Для того чтобы твоя речь прозвучала воодушевленно, мало хорошо продумать что сказать. Нужно увлечься темой, глубоко ее прочувствовать. Во время митинга / выступления ты должен думать не только о фактах, которые нужно изложить, но и о том, что тема беседы означает для тебя и для присутствующих. 2. Думай о тех кто тебя слушает Еще один немаловажный фактор, от которого зависит воодушевление— это твердая убежденность в том, что твоя речь очень важна для слушателей. Тщательно проанализируй, почему эта информация им нужна, какую пользу она принесет и как ее представить так, чтобы слушатели в полной мере увидели ее ценность. Работай с темой до тех пор, пока ты не найдешь что-то такое, что тебя по-настоящему восхитит. 3. Воодушевление = оживленность Воодушевление проявляется в оживленности, а оживленность бывает видна по выражению лица. Говорить нужно убежденно, но не догматично. Однако, не следует впадать в крайность. Некоторые люди готовы восхищаться всем. Таким людям нужно помочь понять, что когда выступающий говорит помпезно или чересчур эмоционально, то слушатели не столько слушают его, сколько думают о нем самом. Но если человек, наоборот, стесняется говорить, ему нужно быть раскованнее. 4. Воодушевление должно соответствовать содержанию речи Следи за тем, чтобы не говорить все время слишком воодушевленно, иначе твои слушатели в буквальном смысле устанут и все выступление и объяснение задач/целей будет напрасно. Как этого избежать? Необходимо составить план встречи так, чтобы разные части можно было изложить по-разному. Постарайся не сбиваться на равнодушный тон, будь увлечен своей речью. Естественно, какие-то мысли потребуют большего воодушевления, а какие-то — меньшего. Твоя задача — искусно переплести их друг с другом. Подытожим. Что тебе нужно, чтобы говорить воодушевлено и интересно для слушателя. #project management #management #softskills

Тема 7

Добавляем подграфики распределения данных по осям на ggplot2 Пакет ggside является расширением для ggplot2, и добавляет в него дополнительные геомы. Имена этих геомон начинаются с geom_xside* или geom_yside*. Продолжение со списком функций и примерами кода по ссылке: https://lnkd.in/eqdsqwST

```
Тема  8
8 книг за 40 дней? много или мало? Как часто вы ставите перед собой новые вызовы?  В конце пр
осьба! Я давно не делал такого и вот решил начать с "Чтения книг"!Цель: Прочитать 8 книг за 4
0 дней- 4 книги на тему hr/подбора- 2 книги на развитие своих компетенций- 2 книги из сфер, к
оторые мне интересны, но не знакомы - для всестороннего развитияГлавное правило: Читать не на
скорость, а с полным пониманием, возможными конспектами и новыми идеями Мне нужна помощь наше
го сообщества Linkedin с выбором книг для чайников об астрономии, журналистике и кино, если т
акие знаете#книги #hr


Тема  9
Коллеги, я доделала сайт для Women in Tech Russia (а то столько лет уже делаем добро, а едино
й 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 програ
мма менторинга уже в самом разгаре, еще можно успеть стать ментором, менти или спикером! http
s://women-in-tech.ru/ #womenintech #womenintechrussia
```

In [106…

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

Вывод:

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

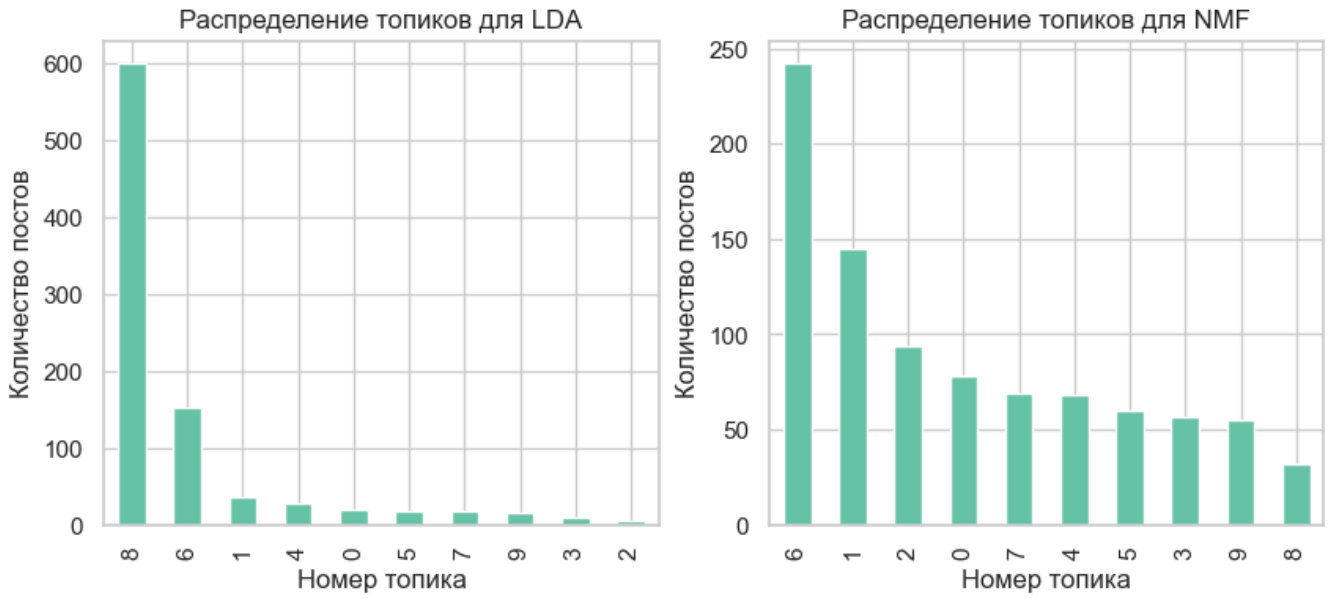
3.4. ТОП-10 тем постов целевой аудитории

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 [107…

```python
# 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 8. Это значит, что алгоритм плохо различает 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: 6, 1, 2, 0 и 7.

- **Тема 0: "Управление бизнесом и развитие продукта в эпоху технологий"**
- **Тема 1: "Развитие профессиональных навыков и требования в разработке программного обеспечения"**
- **Тема 2: "Поиск работы в сфере дизайна и карьерное развитие"**
- Тема 3: "Онлайн-курсы и обучение веб-дизайну: теория и практика для студентов"
- Тема 4: "Формирование эффективной команды разработчиков для успешной разработки технического продукта"
- Тема 5: "Разработка мобильных и веб-приложений: создание продукта с удобным интерфейсом и помощь разработчиков"
- **Тема 6: "Важность обмена идеями и мнениями в работе: получение результатов и решение проблем"**
- **Тема 7: "Управление данными и их использование: функции, код и аналитика"**
- Тема 8: "Профессиональное развитие и обмен знаниями: книги, практика и облачные каналы"
- Тема 9: "Управление профессиональным развитием и участие в проектах: программы, информация и деление опытом"

3.5. ТОП-10 тем, вызывающих наибольшую реакцию

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

Давайте посчитаем все типы реакций для каждой из тем.

In [108…

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

Out[108]:

nmf_topic	comments	likes	reaction	reposts
0	231	2536	3185	418
1	528	6788	7905	589
2	1179	12675	14739	885
3	692	3723	4851	436
4	162	1009	1245	74
5	48	603	706	55
6	1398	7088	8896	410
7	229	1405	1765	131
8	84	700	802	18
9	141	1507	1721	73

В целом видна корреляция между разными типами реакций. Из 10 тем, в качестве наиболее популярных и интересных можно отметить темы: 2, 6, 1, 3, 0.

- **Тема 0: "Управление бизнесом и развитие продукта в эпоху технологий"**
- **Тема 1: "Развитие профессиональных навыков и требования в разработке программного обеспечения"**
- **Тема 2: "Поиск работы в сфере дизайна и карьерное развитие"**
- **Тема 3: "Онлайн-курсы и обучение веб-дизайну: теория и практика для студентов"**
- Тема 4: "Формирование эффективной команды разработчиков для успешной разработки технического продукта"
- Тема 5: "Разработка мобильных и веб-приложений: создание продукта с удобным интерфейсом и помощь разработчиков"
- **Тема 6: "Важность обмена идеями и мнениями в работе: получение результатов и решение проблем"**
- Тема 7: "Управление данными и их использование: функции, код и аналитика"
- Тема 8: "Профессиональное развитие и обмен знаниями: книги, практика и облачные каналы"
- Тема 9: "Управление профессиональным развитием и участие в проектах: программы, информация и деление опытом"

Выводы:

- Т.к. мы получили всего 10 тем, ТОП пришлось сократить до 5.
- ТОП тематики постов целевой аудитории и ТОП тем вызывающих интерес, во многом совпадают. Но есть и различия, например по теме 7 есть публикации, но реакция на них ниже и наоборот, на тему 3 присутствует интерес, но публикаций недостаточно.

Выводы

Мы провели исследование для EdTech, сервиса онлайн образования. Для исследования собрали данные о пользователях и публикациях в социальной сети *Linkedin*. Тема исследования - наставничество и менторство. Для проведения исследования, собрали контент созданный целевой аудиторией социальной сети. В качестве контента использовали информацию из открытых

профилей пользователей и публикуемые ими сообщения. Собранные данные были обработаны и создан датасет.

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

- Тема 6: "Важность обмена идеями и мнениями в работе: получение результатов и решение проблем"
- Тема 1: "Развитие профессиональных навыков и требования в разработке программного обеспечения"
- Тема 2: "Поиск работы в сфере дизайна и карьерное развитие"
- Тема 0: "Управление бизнесом и развитие продукта в эпоху технологий"
- Тема 7: "Управление данными и их использование: функции, код и аналитика"

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

- Тема 2: "Поиск работы в сфере дизайна и карьерное развитие"
- Тема 6: "Важность обмена идеями и мнениями в работе: получение результатов и решение проблем"
- Тема 1: "Развитие профессиональных навыков и требования в разработке программного обеспечения"
- Тема 3: "Онлайн-курсы и обучение веб-дизайну: теория и практика для студентов"
- Тема 0: "Управление бизнесом и развитие продукта в эпоху технологий"

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 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.