

Проект: взаимодействие пользователей с карточками Яндекс.Дзен

1. Описание проекта

Цель проекта:

автоматизация процесса анализа взаимодействия пользователей с карточками статей Яндекс.Дзен.

Метрики анализа взаимодействия пользователей с карточками статей Яндекс.Дзен:

- количество взаимодействий пользователей с карточками в разбивке по темам карточек;
- количество карточек, генерируемых источниками с разными темами;
- соотношение тем карточек с темами источников.

Источник данных: база данных ZEN.

План выполнения проекта:

- создать коннекция с базой данных ZEN;
- выполнить SQL запрос;
- создать csv файл с исходными данными.
- написать скрипт дашборда согласно утвержденному макету;
- создать дашборд с исходными данными;
- подготовить презентация с отчетом.

Дополнительные данные:

- Каждую карточку определяют её тема и источник.
- У каждого источника есть тема.
- Пользователи разделены на возрастные категории.
- Пайплайн будет брать данные из таблицы с сырыми данными, трансформировать данные и укладывать их в агрегирующую таблицу.

2. Подготовительный этап анализа

2.1. Загрузка библиотек

```
# импортируем библиотеки
import pandas as pd
from sqlalchemy import create_engine, text
from IPython.display import Image
from matplotlib import pyplot as plt
import seaborn as sns
```

▼ 2.2.Загрузка данных

```
#создаем коннекцию к базе
db_config = {'user': 'praktikum_student', # имя пользователя
            'pwd': 'Sdf4$2;d-d30pp', # пароль
            'host': 'rc1b-wcoijxj3yxfsf3fs.mdb.yandexcloud.net'
            'port': 6432, # порт подключения
            'db': 'data-analyst-zen-project-db'} # название баз
```

```
#запрашиваем сырые данные
connection_string = 'postgresql://{user}:{password}@{host}:{port}/{db}'.format(db_config['user'],
db_config['pwd']
db_config['host']
db_config['port']
db_config['db']
```

```
engine = create_engine(connection_string)
```

```
# напишем запрос и выгрузим данные в датасет
query = '''
```

```
        SELECT *
        FROM dash_visits
    '''
```

```
df = pd.read_sql_query(sql=text(query), con=engine.connect())
```

```
# изменим тип данных в столбце dt
df['dt'] = pd.to_datetime(df['dt'])
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30745 entries, 0 to 30744
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
dt      datetime64[ns]  30745         object
user_id  int64            30745         int64
visit_id int64            30745         int64
device   object           30745         object
page     object           30745         object
referrer object           30745         object
```

```
0    record_id      30745 non-null int64
1    item_topic     30745 non-null object
2    source_topic   30745 non-null object
3    age_segment    30745 non-null object
4    dt             30745 non-null datetime64[ns]
5    visits         30745 non-null int64
dtypes: datetime64[ns](1), int64(2), object(3)
memory usage: 1.4+ MB
```

▼ 3. Выгрузка данных

```
# вызрузим csv файл для создания дашборда
df.to_csv('zen-project.csv', index = False)
```

▼ 4. Дашборд

https://public.tableau.com/app/profile/anna.chaplin/viz/Project_Yandex_Dzen_16852901117180/Dashboard1

▼ 5. Презентация

<https://disk.yandex.ru/i/beTnaTITnXZ5bg>

✓ 0 сек. выполнено в 01:28

