Metrics_AntonNew_HomeWork_4.ipynb

Файл Изменить Вид Вставка Среда выполнения Инструменты Справка Изменения сохранены

Комментировать Поделиться

+ Код + Текст

ОЗУ Диск

Редактирование

Подготовка

[266]
```
# imports
import numpy as np
import pandas as pd
import matplotlib
import plotly.express as px
from sqlalchemy import create_engine
from datetime import datetime, timedelta
from plotly.subplots import make_subplots
```

[267]
```
# connection
HOST = '37.139.42.145'
DBNAME = 'game-analytics'
USER = 'analytics'
PASSWORD = 'BRtTaqYiJyr29WXN'
TABLE_SCHEMA = 'data_viz_1068.project_dataset'
engine = create_engine(f'postgresql://{USER}:{PASSWORD}@{HOST}/{DBNAME}')
```

[268]
```
# getting table from database
project_dataset = pd.read_sql(f"""
    SELECT *
    FROM {TABLE_SCHEMA}
""", con=engine)

# converting some types to datetime
project_dataset['event_date'] = pd.to_datetime(project_dataset['event_date'])
project_dataset['cohort_date'] = pd.to_datetime(project_dataset['cohort_date'])

# dropping full duplicates (if they exists)
project_dataset.drop_duplicates()

# printing
project_dataset.head(2)
```

	event_time	event_date	event_name	revenue_usd	region	country	device_type	platform	cohort_date	user_id	user_type	content_id	currency	re
0	2020-11-15 00:33:21	2020-11-14	LaunchApp	0.0	EU	UA	xiaomi-Redmi Note 7	android	2020-11-03	187230	organic	None	None	
1	2020-11-15 00:33:46	2020-11-14	LaunchApp	0.0	EU	RU	iPhone 8	ios	2020-10-24	170448	non_organic	None	None	

[269]
```
project_dataset = project_dataset.dropna(subset=['event_time'])
```

Дополнение основного датасета

[270]
```
# copy of main df
refined_df = project_dataset.copy()
```

[271]
```
# adding CIS region
refined_df['region'] = (np.where(
                        refined_df['country'].isin(['RU', 'UA', 'BE', 'AZ', 'AM', 'KZ', 'KG', 'MD', 'TJ', 'TM', 'UZ']),
                        'CIS',
                        refined_df['region']))
```

[272]
```
# adding install datetime
install_times = refined_df[refined_df['event_name']=='FirstLaunchApp']
install_times = install_times.drop_duplicates(subset=('user_id'))
install_times = install_times[['user_id', 'event_time']]
install_times = install_times.rename(columns={'event_time' : 'install_time'})
refined_df = refined_df.merge(install_times, on='user_id')
refined_df.head(1)
```

	event_time	event_date	event_name	revenue_usd	region	country	device_type	platform	cohort_date	user_id	user_type	content_id	currency	rev
0	2020-11-15 00:33:46	2020-11-14	LaunchApp	0.0	CIS	RU	iPhone 11 Pro Max	ios	2020-11-14	202013	organic	None	None	

```
[273]
# adding install month
refined_df['install_month'] = refined_df.install_time.dt.to_period('M')
```

```
[274]
# Removing events which happend before install times.
refined_df = refined_df[refined_df['event_time'] >= refined_df['install_time']].copy()
# adding living hours
refined_df['lifehour'] = ((refined_df['event_time'] - refined_df['install_time']).dt.total_seconds() / 3600).astype('int')
# adding living days
refined_df['lifeday'] = (refined_df['event_time'] - refined_df['install_time']).dt.days
refined_df.head(1)
```

	event_time	event_date	event_name	revenue_usd	region	country	device_type	platform	cohort_date	user_id	user_type	content_id	currency	rev
0	2020-11-15 00:33:46	2020-11-14	LaunchApp	0.0	CIS	RU	iPhone 11 Pro Max	ios	2020-11-14	202013	organic	None	None	

1. Процент платящих в первые 24 часа

```
[275]
df = refined_df.copy()
# leaving only payers
df = df[df['revenue_usd']>0]
```

```
[276]
def CalculatePercent(df):
    all_payers = df['user_id'].nunique()
    df = df[df['event_time'] <= df['install_time'] + timedelta(hours=24)]
    payers_24 = df['user_id'].nunique()
    percent = payers_24 / all_payers * 100
    return percent
```

```
[277]
result = pd.DataFrame([])

result.loc['CIS','android'] = CalculatePercent(df[(df['region']=='CIS') & (df['platform']=='android')])
result.loc['CIS','ios'] = CalculatePercent(df[(df['region']=='CIS') & (df['platform']=='ios')])
result.loc['CIS','organic'] = CalculatePercent(df[(df['region']=='CIS') & (df['user_type']=='organic')])
result.loc['CIS','non_organic'] = CalculatePercent(df[(df['region']=='CIS') & (df['user_type']=='non_organic')])

result.loc['NA','android'] = CalculatePercent(df[(df['region']=='NA') & (df['platform']=='android')])
result.loc['NA','ios'] = CalculatePercent(df[(df['region']=='NA') & (df['platform']=='ios')])
result.loc['NA','organic'] = CalculatePercent(df[(df['region']=='NA') & (df['user_type']=='organic')])
result.loc['NA','non_organic'] = CalculatePercent(df[(df['region']=='NA') & (df['user_type']=='non_organic')])

result.style.set_caption('24 hour payers percent').background_gradient(cmap='coolwarm', axis=1)
```

24 hour payers percent

	android	ios	organic	non_organic
CIS	48.22	47.01	48.59	40.26
NA	45.32	46.30	46.05	42.13

Рассмотрим процент пользователей которые совершили покупку в свои первые 24 часа.

В CIS регионе лидер - organic. (android чуть меньше).
В NA регионе лидером является IOS.

Не органические пользователи покупают в первые 24 часа намного хуже всех остальных. Видимо покупной трафик который приходит в игру не очень хороший с точки зрения монетизации.

2. Самый покупаемый пак

```
[278]
df = refined_df.copy()
# leaving only payers
df = df[df['revenue_usd']>0]
# leaving only first paymments
df = df.sort_values('event_time')
df = df.drop_duplicates('user_id', keep='first')
```

```
[279]
def GetTopSeller(df):
    content_id = df['content_id'].value_counts().index[0];
    revenue = df[df['content_id']==content_id]['revenue_usd'].median()
    return f"{content_id}_${revenue}"
```

```
[280]
result = pd.DataFrame([])

result.loc['CIS','android'] = GetTopSeller(df[(df['region']=='CIS') & (df['platform']=='android')])
result.loc['CIS','ios'] = GetTopSeller(df[(df['region']=='CIS') & (df['platform']=='ios')])
result.loc['CIS','organic'] = GetTopSeller(df[(df['region']=='CIS') & (df['user_type']=='organic')])
result.loc['CIS','non_organic'] = GetTopSeller(df[(df['region']=='CIS') & (df['user_type']=='non_organic')])

result.loc['NA','android'] = GetTopSeller(df[(df['region']=='NA') & (df['platform']=='android')])
result.loc['NA','ios'] = GetTopSeller(df[(df['region']=='NA') & (df['platform']=='ios')])
result.loc['NA','organic'] = "ta_starter_$10/" + GetTopSeller(df[(df['region']=='NA') & (df['user_type']=='organic')])
result.loc['NA','non_organic'] = GetTopSeller(df[(df['region']=='NA') & (df['user_type']=='non_organic')])

```

Топ селлеры среди первых покупок.

```
[281]
result.style.set_caption('Top sellers')
```

Top sellers

	android	ios	organic	non_organic
CIS	ta_low_offer_3_$0.653616	ta_low_offer_3_$1.302585	ta_low_offer_3_$0.664544	ta_low_offer_3_$1.277263
NA	ta_challenge_pass_$7.99	ta_starter_$10.0	ta_starter_$10/ta_low_offer_3_$1.0	ta_challenge_pass_$8.0

Для органики ta_low_offer_3 и ta_starter одинаково популярны.

```
[282]
display(df[(df['region']=='NA') & (df['user_type']=='organic')]['content_id'].value_counts().head(3))
```

```
ta_low_offer_3       149
ta_starter           149
ta_challenge_pass    142
Name: content_id, dtype: int64
```

Посмотрим цены на популярные паки в разных регионах. Возможно цены на одни и те же товары в регионах разные. И это может влиять на выбор.

Для андроида в CIS регионе цены ниже

```
[283]
display(df[(df['region']=='CIS') & (df['platform']=='android') & (df['content_id']=='ta_low_offer_3')]['revenue_usd'].median())
display(df[(df['region']=='NA') & (df['platform']=='android') & (df['content_id']=='ta_low_offer_3')]['revenue_usd'].median())
display(df[(df['region']=='CIS') & (df['platform']=='android') & (df['content_id']=='ta_challenge_pass')]['revenue_usd'].median())
display(df[(df['region']=='NA') & (df['platform']=='android') & (df['content_id']=='ta_challenge_pass')]['revenue_usd'].median())
```

```
0.653616
0.99
5.4239075
7.99
```

А для IOS цены в CIS наоборот выше

```
[284]
display(df[(df['region']=='CIS') & (df['platform']=='ios') & (df['content_id']=='ta_low_offer_3')]['revenue_usd'].median())
display(df[(df['region']=='NA') & (df['platform']=='ios') & (df['content_id']=='ta_low_offer_3')]['revenue_usd'].median())
display(df[(df['region']=='CIS') & (df['platform']=='ios') & (df['content_id']=='ta_starter')]['revenue_usd'].median())
display(df[(df['region']=='NA') & (df['platform']=='ios') & (df['content_id']=='ta_starter')]['revenue_usd'].median())
```

```
1.302585
1.0
11.611672
10.0
```

Для не органики
цены в CIS ta_low_offer_3 на андроид выше.
цены в CIS ta_challenge_pass на андроид наоборот ниже.

```
[285]
display(df[(df['region']=='CIS') & (df['user_type']=='non_organic') & (df['content_id']=='ta_low_offer_3')]['revenue_usd'].median())
display(df[(df['region']=='NA') & (df['user_type']=='non_organic') & (df['content_id']=='ta_low_offer_3')]['revenue_usd'].median())
display(df[(df['region']=='CIS') & (df['user_type']=='non_organic') & (df['content_id']=='ta_challenge_pass')]['revenue_usd'].median())
display(df[(df['region']=='NA') & (df['user_type']=='non_organic') & (df['content_id']=='ta_challenge_pass')]['revenue_usd'].median())
```

```
1.277263
1.0
6.6717275
8.0
```

То что товары разнятся в цене в одном типе когорты в разных регионах в данном случае не влияет на выбор пользовтелей.

Вывод

```
1  result.style.set_caption('Top sellers')
```

Top sellers

	android	ios	organic	non_organic
CIS	ta_low_offer_3_$0.653616	ta_low_offer_3_$1.302585	ta_low_offer_3_$0.664544	ta_low_offer_3_$1.277263
NA	ta_challenge_pass_$7.99	ta_starter_$10.0	ta_starter_$10/ta_low_offer_3_$1.0	ta_challenge_pass_$8.0

Заметно что в CIS регионе преобладают более дешевые товары чем в NA регионе.
В CIS регионе во всех кагортах топ селлером является ta_low_offer_3.
Видимо в CIS регионе пользователи склонны платить меньшую сумму за товар при первой покупке (возможно их финансовое положение хуже).
В NA пользователи готовы отдать более значительную сумму за первую покупку.