Анализ размещении реклам и жалоб

Данный проект связан с жалобами в сфере индустрии “площадок по размещению объявлении”, прототип “Авито”.

Представим у нас имеется БД на стеке PostgreSQL. Таблицы “adv” & “ads_pro”
Структура таблицы след:

adv

- user_id (уникальный id пользователя)
- category_id (id по категориям товаров)
- category_name (наименование категории)
- server_date (дата размещения объявления)

ads_pro

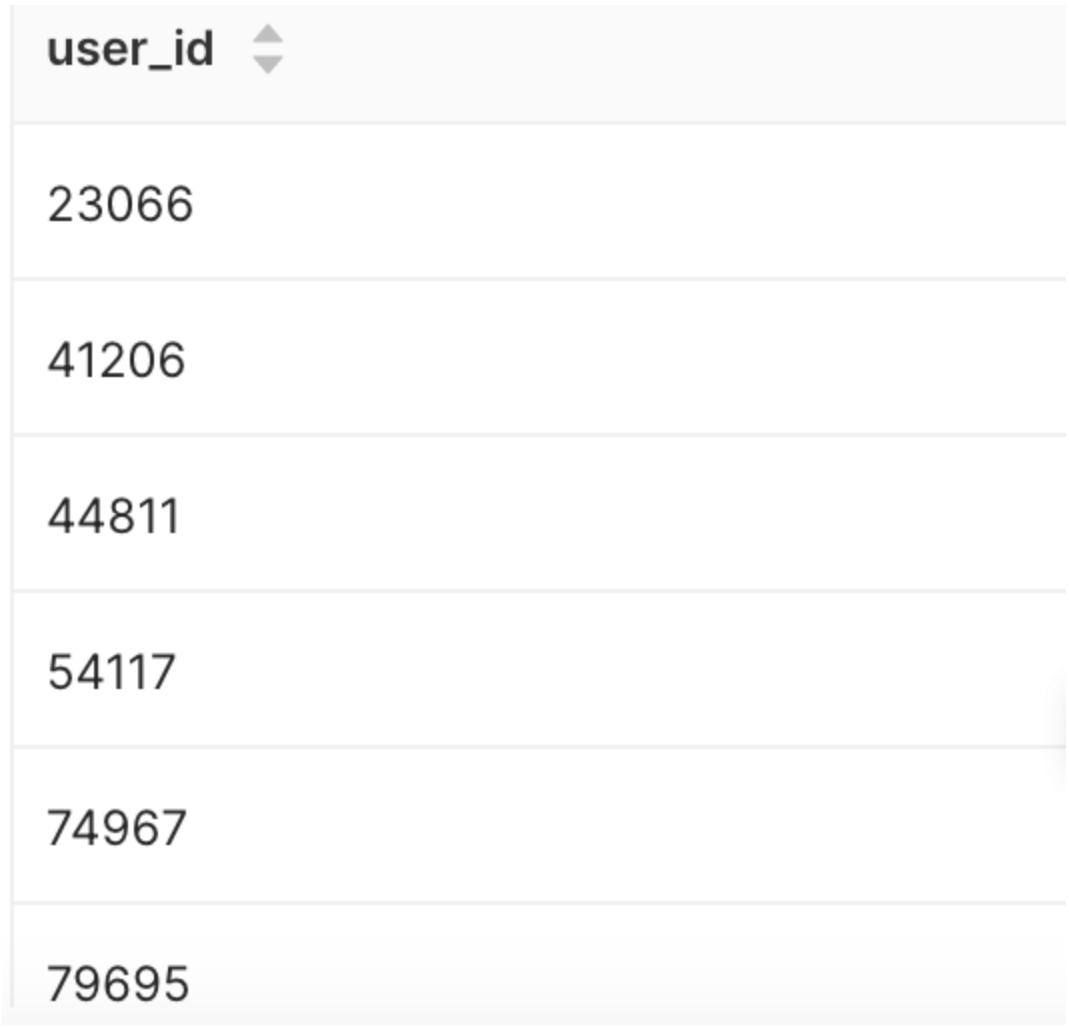
- id (уникальный номер жалобы
- ad_id (уникальный номер объявлении)
- content (комментарии к жалобам)
- type (тип жалоб)
- ip (Хэш по IP адресам)
- checked_at\checked_by (была ли проведена работа по жалобам)

Цель данного проекта - это выявить количество размещенных объявлении по категориям и построить по ним матрицу, также свободная аналитика (комменты) по поиску инсайтов в данных и предоставлении рекомендации для улучшения

пользовательского опыта. Задачи предстоят выполнять с помощью SQL запросов, использую также Python, а также поможет умение работать по интеграции с Google Tables и с библиотеками для соединения с БД.

Для начала необходимо выявить уникальных пользователей из таблицы adv, которые размещали объявления в Марте 2020, но не размещали в Апреле 2020 года. Для решения задачи буду использовать SQL запрос.

```
select
  distinct user_id
from adv
where date_trunc('month', server_date) = '2020-03-01'
  and user_id not in (select
                      distinct user_id
                      from adv
                      where date_trunc('month', server_date) =
                      '2020-04-01')
-- примерно как выглядит результирующая таблица ниже на скрине
```



user_id
23066
41206
44811
54117
74967
79695

Далее пропишем запрос, используя таблицу adv, который будет демонстрировать сколько объявлении было размещено в разрезе категории и проставим по ним ранки для пользователей. Эта результирующая таблица поможет построить матрицу по размещении по категориям

```
select
  user_id,
  category_id,
  category_name,
  COUNT(*) as ads_cnt,
  rank () over (partition by user_id
                              order by count (*) DESC) as catego
from adv
```

```
group by user_id, category_id, category_name

-- для тестирования используем user_id = 82389
```

user_id	category_id	category_name	ads_cnt	category_rank
449	37	electronic	1	1
10327	1	real estate	1	1
20342	3	car parts	1	1
21747	891	fashion and style	3	1
23066	891	fashion and style	1	1
40419	37	electronic	1	1

user_id	category_id	category_name	ads_cnt	category_rank
82389	37	electronic	3	1
82389	899	home and garden	2	2
82389	3	car parts	1	3

Следующая задача, состоит в том чтобы подсоединиться к БД с помощью питона для поиска инсайтов и размещения файла в Google Tables

```
import psycopg2
import pandas as pd


conn = psycopg2.connect(
    dbname="data_base_name",
    user="user_name",
    password="password",
    host="example.examplehost.com",
```

```
    port="1234"
)
```

```
query = """
select
  user_id,
  category_id,
  category_name,
  COUNT(*) as ads_cnt,
  rank () over (partition by user_id
                              order by count (*) DESC) as categ
from adv
group by user_id, category_id, category_name
"""
```

```
df = pd.read_sql(query, conn)
df.head()
```

	user_id	category_id	category_name	ads_cnt	category_rank
0	449	37	electronic	1	1
1	10327	1	real estate	1	1
2	20342	3	car parts	1	1
3	21747	891	fashion and style	3	1
4	23066	891	fashion and style	1	1

я выгрузил таблицу, далее чтобы построить матрицу по категориям с учетом расчета количества пользователей, которые размещали объявления.

```
# для решения воспользуюсь свод таблицей со след параметрами
pivot_df = df.pivot_table(index='user_id',
```

```
    columns='category_name',
    values='ads_cnt', aggfunc='count', fill_value=0)
```

```
pivot_df
```

category_name	animals	business and services	car parts	children	electronic	fashion and style	hobby, rest and sport	home and garden	jobs	real estate	transport
user_id											
449	0	0	0	0	1	0	0	0	0	0	0
10327	0	0	0	0	0	0	0	0	0	1	0
20342	0	0	1	0	0	0	0	0	0	0	0
21747	0	0	0	0	0	1	0	0	0	0	0
23066	0	0	0	0	0	1	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...
221293830	0	0	0	0	0	0	0	1	0	0	0
221305183	0	0	0	0	0	1	0	0	0	0	0
221307731	0	0	0	0	1	0	0	0	0	0	0

```
# транспонирование матрицы и матрич.умножение
matrix_c = pivot_df.T.dot(pivot_df)
matrix_c
```

category_name	animals	business and services	car parts	children	electronic	fashion and style	hobby, rest and sport	home and garden	jobs	real estate	transport
category_name											
animals	23865	383	388	1022	1051	1427	799	1181	112	154	241
business and services	383	37285	1000	1178	2131	2000	1130	2342	796	367	566
car parts	388	1000	55184	1477	4424	2101	2017	3075	197	235	1920
children	1022	1178	1477	104731	6059	17466	5723	5725	336	512	457
electronic	1051	2131	4424	6059	146533	10283	6885	8488	637	689	1245
fashion and style	1427	2000	2101	17466	10283	150964	8518	8542	715	756	600
hobby, rest and sport	799	1130	2017	5723	6885	8518	61327	6041	261	330	529
home and garden	1181	2342	3075	5725	8488	8542	6041	77636	401	662	804

Загрузим получившую матрицу в гугл диск след образом

```
import os
import gspread

# потребуется для того чтобы авторизироваться
```

```
from oauth2client.service_account import ServiceAccountCredentia

# потребуется для того чтоыб  заливать пандавский датафрейм в гу
from df2gspread import df2gspread as d2g



# мои учетные данные
path_to_credential = os.getcwd() + '/my_json.json'

# лимит доступа для работы с API
scope = ['https://spreadsheets.google.com/feeds',
         'https://www.googleapis.com/auth/drive']

# авторизация
credentials = ServiceAccountCredentials.from_json_keyfile_name(
        path_to_credential, scope)
gs = gspread.authorize(credentials)
```

```
# название книги
workbook_name = f"Матрица_размещении_объявлении_по_категориям"

# создание новой книги
workbook = gs.create(workbook_name)

# добавляем лист Проект
sheet = workbook.add_worksheet(title="Проект", rows="1",
    cols="1")

# предоставляю доступ к данной гугл таблице
table_name = 'Матрица_размещении_объявлении_по_категориям'
sheet = gs.create(table_name)
my_mail = 'example@gmail.com'
sheet.share(my_mail, perm_type='user', role='writer')
```

```
df = matrix_c.copy() # копия матрицы

# Название гугл шита
spreadsheet_name = 'Матрица_размещении_объявлении_по_категориям

# Загружаем таблицу с матрицами в лист 'Проект'
sheet_name = 'Проект'
d2g.upload(df, spreadsheet_name, sheet_name,
    credentials=credentials,
    row_names=True)
```

Ad-hoc аналитика: Наибольшее количество взаимодействий наблюдается в категориях:
- “мода и стиль” (150,964)
- “Электроника” (146,533)
- “дети” (104,731)

Если обратить внимание на пересечения, то наибольшее количество приходится также на:
- “мода и стиль” и “дети” (17,466 и пересечение с “дети” составляет долю 16,68%)
- “электроника” и “автозапчасти” (4,424 и пересекается с “электроникой” на 8,02%))

Ну а наименее популярные категории могу выделить одно из наименьших это категории “бизнес” и “работа”.

Далее буду анализировать жалобу (используя таблицу ads_pro, к примеру отвечу на такие вопросы как:
- какие частые причины являются для жалоб
- как быстро работает “система работы над жалобами”
- имеется ли частые пользователи, которые жалуются.

```
# дополнительно импортирую библиотеки по визуализации
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# настройка plota
sns.set(rc={'figure.figsize':(16,6)}, style='whitegrid')
```

прежде конечно начну с EDA

```
ads_pro.dtypes
ads_pro.info()
ads_pro.head()
```

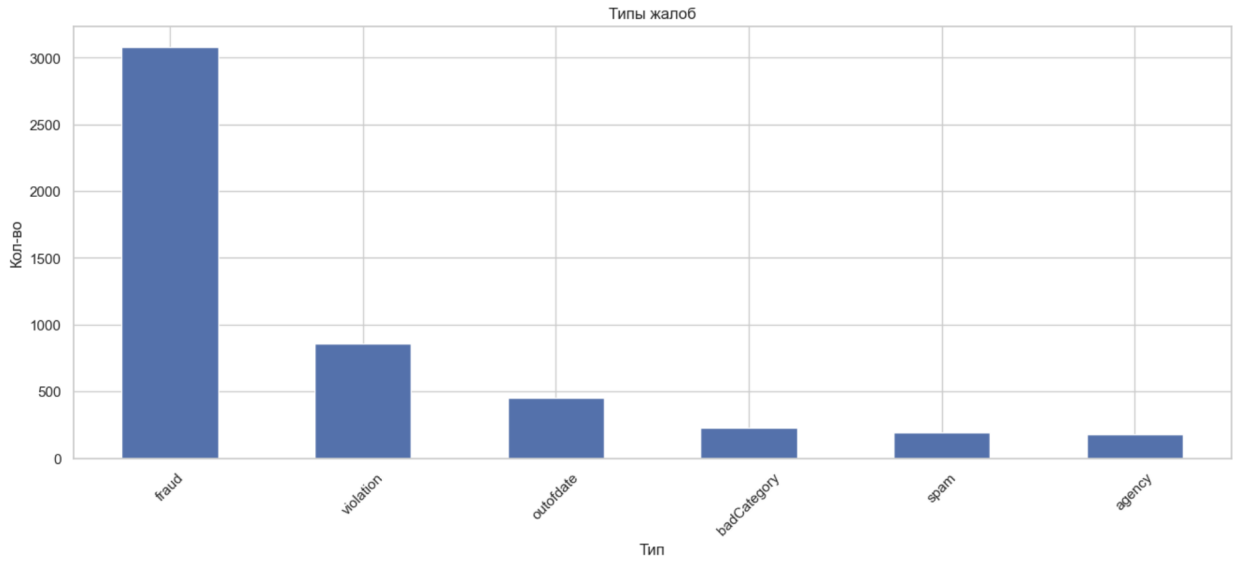
Далее сделав обзор примерно понимаю какие колонки исходя из задачи буду просматривать

```
ads_pro.type.value_counts()
```

```
fraud          3083
violation       862
outofdate       455
badCategory     230
spam            192
agency          178
Name: type, dtype: int64
```

вижу числа далее также визиализирую

```
ax = ads_pro.type.value_counts().plot(kind='bar')
ax.set_xticklabels(ax.get_xticklabels(), rotation=45)
plt.xlabel('Тип')
plt.ylabel('Кол-во')
plt.title('Типы жалоб')
plt.show()
```



точечно просматриваю каждый тип жалоб, тоже некий обзор датафрейма

```
ads_pro.query('type == "outofdate"')
ads_pro.query('type == "fraud"')
```

	operation_timestamp	id	ad_id	type	content	created_at	ip	checked	checked_by	checked_at	checked_as	ban_id
10	2019-06-13 12:05:47	15063235	598125386	outofdate	Хамит в смс	2019-06-13 15:05:32	0	0	0	NaT	None	0
14	2019-06-13 12:06:31	15063239	552197351	outofdate	Не відповідного змісту	2019-06-13 15:06:14	0	0	0	NaT	None	0
18	2019-06-13 12:07:10	15063243	592937823	outofdate	Ложь,работают абы как.	2019-06-13 15:06:41	5	0	0	NaT	None	0
34	2019-06-13 12:10:12	15063259	598651574	outofdate	Включили olx доставку, а цена не актуальна. Дл...	2019-06-13 15:09:44	0	0	0	NaT	None	0
44	2019-06-13 12:12:25	15063269	595814653	outofdate	Машеник проживає в республіці тому відправити ...	2019-06-13 15:11:54	60	0	0	NaT	None	0

	operation_timestamp	id	ad_id	type	content	created_at	ip	checked	checked_by	checked_at	checked_as	ban_id
0	2019-06-13 12:04:13	15063225	599051878	fraud	Несуществующий объект. Объявление-пустышка для...	2019-06-13 15:03:55	34	0	0	NaT	None	0
1	2019-06-13 12:04:14	15063226	599064676	fraud	Нет автомобиля в наличии! Вымогает деньги! Зан...	2019-06-13 15:03:57	0	0	0	NaT	None	0
3	2019-06-13 12:04:55	15063228	599072912	fraud	Скриншоты из сайтав зделоны на деньги кинуть х...	2019-06-13 15:04:29	5	0	0	NaT	None	0
4	2019-06-13 12:04:56	15063229	599071007	fraud	Обманюють людей	2019-06-13 15:04:33	0	0	0	NaT	None	0
5	2019-06-13 12:05:01	15063230	599064775	fraud	Мошенники нет в наличии цена не соответствует	2019-06-13 15:04:49	0	0	0	NaT	None	0

далее хочу посмотреть ad_id их количество в целом

```
ads_pro.ad_id.value_counts() # id по номеру 599140479 занимает л
```

```
599140479    530
599138135    199
599086897     21
599110832     19
599076558     19
             ...
598939075      1
597803413      1
579265434      1
435328420      1
550286958      1
Name: ad_id, Length: 3113, dtype: int64
```

смотрю что за ad_id и из чего она состоит

```
ads_pro.query('ad_id == 599140479')
```

	operation_timestamp	id	ad_id	type	content	created_at	ip	checked	checked_by	checked_at	checked_as	ban_id
1445	2019-06-13 20:33:56	15065666	599140479	violation	Предлагают порно	2019-06-13 23:33:36	0	0	0	NaT	None	0
1494	2019-06-13 20:33:56	15065667	599140479	violation	Сексуальні послуги	2019-06-13 23:33:36	660402	0	0	NaT	None	0
2341	2019-06-13 20:33:55	15065664	599140479	violation	Секс. (	2019-06-13 23:33:28	3131	0	0	NaT	None	0
2382	2019-06-13 20:29:00	15065603	599140479	violation	Она торгует секс услугами	2019-06-13 23:28:42	9	0	0	NaT	None	0
2386	2019-06-13 20:29:05	15065607	599140479	violation	Порно сайт	2019-06-13 23:29:00	0	0	0	NaT	None	0

вижу что она больше всего состоит из контента 18+ и относится к типу violation. то есть уже исходя из количество всего 530 штук таких объявлении или жалоб только это доля этого ad_id равна 62% (оставим и это на заметку).

при этом контент разный.
Далее хочу посмотреть на контент и их количество скажем так ТОП-20

```
ads_pro.content.value_counts().head(20)
```

```
Порнография
68
Агентство в рубрике от частных лиц
47
Секс услуги
43
Здравствуйте!\nЭто мошенник,каждый день с десятками новых аккаунтов!\nВы его удаляете,оно снова лезет...\nПримите м
еры!!!    37
Проституция
34
Неверная рубрика
33
Мошенничество
30
Интим услуги,боди массаж,секс проституция,эротический массаж.
20
Проблемные документы
17
Нет автомобиля в наличии! Вымогает деньги! Занижена цена!
15
Интим услуги
15
Неактуальное объявление
14
Мошенники нет в наличии цена не соответствует
13
Агенство недвижимости
12
Добрый день! Удалите пожалуйста это объявление! Это мошенники, требуют предоплату не показывая квартиры!
12
Развод на деньги
11
```

видно что порнография лидирует. пойдем точечно анализировать по такому же методу посмотрим контент Порнография

```
ads_pro[ads_pro['content'].str.contains("Порнография",
    na=False)]ad_id.value_counts()
```

```
599140479    61
599138135    11
599165800     1
599168384     1
Name: ad_id, dtype: int64
```

```
ads_pro.query('ad_id == 599140479')['content'].value_counts()
```

```
Порнография                                                  55
Секс услуги                                                  22
Проституция                                                  21
Проститутка                                                   7
Интим услуги                                                  6
                                                             ..
Еротика и порнография                                         1
Секс фото, порно фото                                         1
Вы в своем уме такое выставлять.Дети сидят в объявлениях.     1
разврат,порно                                                 1
Шикарно))) и так каждую ночь!                                 1
Name: content, Length: 388, dtype: int64
```

Исходя из предварительного обзора я бы предложил каждый тип (type) разделить еще на под категории исходя из контента (content) к примеру type = ‘violation’ и будет subtype = [’porn’, ‘sextoys’, ‘eskort’ etc]. То есть здесь также идет связка и для content то есть текстовый анализ (если в значении содержатся такие слова как порнография, порно их уже можно относить к под категории ‘porn’). Думаю это может привести к оперативному пересечения мер по отношении к пользователям (бан, блокировка, штрафы и прочие санкции исходя уже из подкатегории).

Далее я бы посмотрел некоторые моменты связанные со временами. К примеру рассчитал бы время реакции след образом

```
ads_pro['reaction_time'] =
    ads_pro['checked_at'] - ads_pro['created_at']
```

отобрал бы все ненулевые значения по новому столбцу (потому что мы знаем что есть нулевые значения в checked_at - выше на скрине где ads_pro.info()

```
  ads_pro.query('reaction_time.notnull()')['type'].value_counts
```

```
fraud          493
outofdate       13
spam            11
badCategory      8
agency           7
violation        6
Name: type, dtype: int64
```

видим что в основном обрабатывается типы “мошенничеств” или fraud. далее можно рассчитать среднее и медианное значение а также самую мин реакцию и макс реакцию.

```
ads_pro.loc[ads_pro['reaction_time'].idxmin()].to_frame()
ads_pro.loc[ads_pro['reaction_time'].idxmax()].to_frame()
```

	268
operation_timestamp	2019-06-13 13:03:25
id	15063492
ad_id	597953253
type	fraud
content	кинул на деньги, скрины,есть
created_at	2019-06-13 16:02:07
ip	0
checked	1
checked_by	0
checked_at	2019-06-13 16:03:21
checked_as	by_system
ban_id	0
reaction_time	0 days 00:01:14

	83
operation_timestamp	2019-06-14 08:52:38
id	15063306
ad_id	599075645
type	fraud
content	Мошенник, этой квартиры нет
created_at	2019-06-13 15:20:34
ip	7
checked	1
checked_by	0
checked_at	2019-06-14 11:52:32
checked_as	by_system
ban_id	0
reaction_time	0 days 20:31:58

```
print(f"Ср.время реакции: {ads_pro['reaction_time'].mean()}")
print(f"Медианное время реакции: {ads_pro['reaction_time'].media

Ср.время реакции: 0 days 05:39:06.797397769
Медианное время реакции: 0 days 03:02:44
```

```
''' также расчитаем соотношение общих значении из датафрейма к т
которые были рассмотрены'''

ads_pro.query('reaction_time.notnull()')['operation_timestamp']
    .value_counts() / ads_pro['operation_timestamp'].dt.day_name

output:
Friday      0.164067
Thursday    0.074180
Name: operation_timestamp, dtype: float64
```

исходя из результатов времени мы можем понять что система по обработке жалоб работает совсем не быстро. Думаю нужно проанализировать процессы по обработке жалоб и таким образом вывести почему так происходит что в пятницу было обработано 16% проверок из всех жалоб или претензии а в четверг 7%. надо больше информаци на счет дней и в целом работы системы для выявления причины. В любом случае надо оптимизировать рабочие процессы системы по обработке жалоб (by_system).

Также я бы посмотрел среднее время по типу

```
ads_pro.groupby('type', as_index=False)['reaction_time'].mean()
    .sort_values(by='reaction_time', ascending=False)
```

	type	reaction_time
0	agency	0 days 10:54:36
3	outofdate	0 days 06:49:51.769230769
1	badCategory	0 days 05:46:06.250000
2	fraud	0 days 05:39:39.131845841
4	spam	0 days 02:57:09.454545454
5	violation	0 days 01:01:04.333333333

Так как частые жалобы по обработке реакции приходилось на мошенничество надо сделать упор на сокращение реакции времени.

Также стоит обратить внимание на пользователей которые подают жалобы часто.

```
ads_pro['ip'].value_counts().head(20)
```

```
0             2124
4294967295     238
9              133
4              127
3              119
5              116
1              115
7              111
6              105
2               96
8               94
1487            20
85              18
31              15
95              14
41              14
21              12
81              12
4977            12
22              11
Name: ip, dtype: int64
```

ну а частые жалобы приходится на нулевой ip (думаю это значит что ip неизвестен). ну а далее за ним идет `4294967295`. в целом стоит отобрать тех пользователей количество которых больше 100 и начать работать с этими пользователями. к примеру проанализировать является ли жалоба обоснованной на основе того же конвента на которую жалуется пользователь. далее обратить внимание также на какие-то может аномалии то есть связи с категориями или может это однотипные нарушения

```
ads_pro.query('ip == 4294967295').filter(['type']).value_counts
```

```
type
fraud          138
violation       43
outofdate       25
badCategory     22
spam             9
agency           1
dtype: int64
```

к примеру данный пользователь по части жалоб больше обращается по типу fraud которую в свою очередь тоже стоит усилить меры по борьбе с мошенничеством (сюда можно отнести какие то информации для пользователей о признаках мошенничества). может даже стоит связаться с пользователем то есть работа в формате обратной связи чтобы понять причины частых жалоб и возможно этот же пользователь может дать какую то важную информацию. кстати также надо обратить внимание на такие жалобы по типу ourofdate то есть я как понимаю это старые объявления которых тоже стоит уделить внимание на может автоматическое устранение подобных объявлений.