Задание для Школы менеджеров Яндекса

Прослушивание Яндекс.Музыки

In [1]:

```
# воспользуемся модулем Pandas, чтобы поработать с таблицей
import pandas as pd
```

In [2]:

```
df = pd.read_csv('/Users/bns/Downloads/music_data.csv', index_col=0)
```

In [3]:

```
# проверим первые несколько строк таблицы
df.head(5)
```

Out[3]:

	audition_id	track_duration	track_genre	track_id	
0	3fecd60bf5564de7bb0064335f482b4d	336.629082	RAP	efd614e5-239a-418e-b39c-181b43719b62	a
1	8ae1703df8354ee6b8d39ce2ceae4508	428.797846	VOCAL	d4697e6e-698a-41e8-8e15-ec890c94751e	
2	44383125d98a4d939e7f294602674fc6	463.467143	METAL	e006afab-c832-4d07-8cd0-7d4a9c2aabe8	9
3	a90b74992c9f4046a68935cb83ced7ee	323.746259	HARDROCK	c2ea845c-fcba-480a-b41a-d58bf9493375	
4	0d98a88fba0a4dc3bb0444089a0ce16b	316.888403	POP	ad68738c-2a38-4e0f-910a-71bb0aa0a0c0	6

In [4]:

```
df['track_genre'].nunique()
```

Out[4]:

```
22
```

In [5]:

```
df['user_id'].nunique()
```

Out[5]:

Out[5]:

```
1727
```

Итак, у нас 22 жанра музыки, из которых нас интересует только один - TECHNO. Но при этом 1727 пользователей.

In [6]:

```
print(df.groupby('track_genre')['track_duration'].mean().sort_values())
```

```
track_genre
INDUSTRIAL      279.025634
HOUSE           300.008262
PUNK            306.771923
HARDROCK        325.856495
RAP             356.900853
POP             374.521012
DANCE           375.409102
RNB             380.362767
DISCO           395.095225
ROCK            415.521762
KPOP            418.328719
VOCAL           426.691425
METAL           452.393251
TECHNO          469.605097
DUBSTEP         500.153042
JAZZ            578.472750
BLUES           578.784888
ELECTRONICS     735.828505
CLASSICAL       933.281535
RELAX          1002.698880
POSTROCK       1290.964020
PODCASTS       4210.708248
Name: track_duration, dtype: float64
```

Теперь давайте введём предпосылку о том, как работет Яндекс.Музыка. Как мне кажется, помимо лайков и дизлайков (что очевидно, но не пишется в логах), Я.Музыка отслеживает какую часть трека мы прослушали. То есть, если трек нам не нравится - мы попросту его пропустим, а если нравится - прослушаем около 90% длины трека. Давайте добавим в таблицу длину прослушивания. Мы отбрасываем короткие прослушивания, потому что иногда алгоритмы яндекс-музыки могут посоветовать песню из Топ-Чарта (а там чаще всего POP жанр), и это может сдвинуть наши последующие выводы. Человек, которому POP не нравится, просто пропустит трек с начала.

In [7]:

```
df['len'] = df['utc_audition_start_dttm'].apply(lambda x: (len(x[:-8:])))
df.len.unique()
```

Out[7]:

```
array([19, 12])
```

Заметим, что не везде время указано в обыденной форме, проигнорируем это, так как возможно возникла ошибка при подсчете.

In [8]:

```
from tqdm.auto import tqdm #таблица весьма большая, поэтому прикрутим прог
рес-бар
from datetime import datetime
tqdm.pandas(desc="Статус вычислений")
def timeinsecs (x):
    try:
        y = (datetime.strptime(x[:-8:], '%Y-%m-%dT%H:%M:%S'))
        z = int(y.timestamp())
        return z
    except:
        return 0 #к сожалению, не все временные отметки в едином формате,
 где-то всего 12 символов вместо 19
df['utc_audition_end_secs']=pd.to_numeric(df['utc_audition_end_dttm'].prog
ress_apply(timeinsecs))
df['utc_audition_start_secs']=pd.to_numeric(df['utc_audition_start_dttm'].
progress_apply(timeinsecs))
df['audition_len_secs']=df['utc_audition_end_secs']-df['utc_audition_start
_secs'] #считаем, сколько времени слушали трек
drop_cols = ['utc_audition_end_secs', 'utc_audition_start_secs', 'len']
df.drop(drop_cols, axis=1, inplace=True) #выкидываем ненужные столбцы
df['listening_progress']=(df.audition_len_secs/df.track_duration)*100 #пос
читаем процент прослушивания
```

```
/opt/anaconda3/lib/python3.8/site-packages/tqdm/std.py:668: F
utureWarning: The Panel class is removed from pandas. Accessi
ng it from the top-level namespace will also be removed in th
e next version
  from pandas import Panel
```

In [9]:

```
df = df[df['listening_progress']>90] #выкидываем все недослушанные треки
```

Итак, мы оставили только треки, которые вероятнее всего понравились слушателям. Давайте теперь найдем самый популярный жанр у каждого слушателя среди тех треков, что ему вероятнее всего понравились.

In [10]:

```
groupby_df = (df.groupby(['user_id','track_genre'])['track_genre'].agg(['c
ount']).sort_values(by='count', ascending=False).reset_index().drop_duplic
ates('user_id', keep='first',inplace = False))
```

Увы, мы не знаем маржинальность, которой обладает один билет (его нам в задании не задали), потому мы не можем утверждать, что стоит отправлять приглашение тем, у кого жанр техно находится на втором месте по популярности, поэтому отправим приглашение лишь тем, кто по-настоящему фанатеет от техно:)

In [11]:

```
final_df = groupby_df [groupby_df['track_genre']=='TECHNO']
final_df
```

Out[11]:

	user_id	track_genre	count
1063	686a6f367b8040398af6	TECHNO	28
1208	64b6c65c4129495ba6a7	TECHNO	26
2832	573008e5938a49a0b732	TECHNO	19
3642	952f35e2706c4426ac9e	TECHNO	17
7016	d07ff9ccf12d41799459	TECHNO	12
7582	b7f34a9f0e24462caa94	TECHNO	12
8919	a0f3ea9a2ea047c286d8	TECHNO	11
11537	f8d3066844134fbe8cb7	TECHNO	9
11857	247b4efe742c413dbbf8	TECHNO	9

Итак, мы выбрали "истинных любителей" TECHNO, то есть тех, кто не проматывает треки техно, а слушает до конца, и тех, кто треки техно слушает чаще, чем любые другие.

Никита Битюцкий (nikbitoff@yandex.ru) для Школы Менеджеров Яндекса - 2021 г

P.S. Идея для развития модели: имея представления о маржинальности билета (то есть то, какую прибыль он приносит организаторам), мы можем рассмотреть для отправки письма в том числе тех, у кого жанр техно второй по популярности