Passion Point - PP Score

Tuan Tran

June 25, 2019

Giới Thiệu Trong tài liệu này sẽ giới thiệu định nghĩa, ý tưởng và giải thích các scripts dùng để tính toán điểm Passion Point (PP) score.

1 Giới Thiệu về Pasion Point score

Định Nghĩa PP score dùng để Đánh giá mức độ yêu thích của người dùng giúp các doanh nghiệp xác định được những người yêu thích (lovers) đối với thương hiệu, sản phẩm của họ. Từ đó có xác định được khách hàng tiềm năng của các sản phẩm mới cũng như những người có thể có những hoạt động tích cực cho sản phẩm hay đại diện cho thương hiệu [15]. Tùy vào loại người dùng có ảnh hưởng, các người có ảnh hưởng chung trên mạng xã hội thuê để quảng cáo cho nhãn hàng hoặc những người có kiến thức về sản phẩm của nhãn hàng hay có những bài chia sẻ và đánh giá về nhãn hàng, tùy từng mức độ nổi tiếng mà ta có những thang đo và tiêu chí để đo độ yêu thích:

- Đối với những người dùng có ảnh hưởng chung trên mạng xã hội có mức độ nổi tiếng rộng được thuê để quảng cáo: chúng ta đánh giá độ yêu thích của họ thông qua tận tâm của họ như thế nào với việc hoàn thành công việc của mình và mức độ lan tỏa sự yêu thích sản phẩm và sự chú ý của cộng đồng theo dõi người nổi tiếng này với sản phẩm.
- Đối với những đối tượng thuộc dạng có mức độ nổi tiếng bình thường hoặc nổi tiếng không phải được thuê để quảng cáo: Họ là những người có những bài viết hay chia sẻ đến cộng đồng về chi tiết sản phẩm của nhãn hàng. Chúng ta cần xác định được mức độ tốt của nội dung họ tạo ra và nội dung hướng tới sản phẩm của họ là khen ngợi hay chê bai, đánh giá sự phản ứng và sự quan tâm của cộng đồng đối với nội dung của họ.

Trong dự án này, chúng tôi nghiên cứu việc xác định mức độ yêu thích của một cá nhân thông qua các yếu tố:

- Số lượng bài viết tương tác của một cá nhân đó đủ lớn trong một khoảng thời gian nhất định: Điều này giúp chúng ta có đủ cơ sở tin cậy để kết luận cá nhân đó yêu thích sản phẩm.

- Mức độ người đó hoạt động trên một khoảng thời gian nhất định: Điều này thể hiện tính trung thành của người đó đối với sản phẩm. Người dùng càng chuyên cần thực hiện tương tác với sản phẩm sẽ thể hiện được mức độ yêu thích của người đó trong một khoảng thời gian nhất định.
- Hành vi của người đó khi thực hiện các hoạt động tương tác đến với sản phẩm có đi theo một chu kỳ: Ví dụ, trong một khoảng thời gian nhất định, sự tương tác của người đó có tính chu kỳ trùng với các thời điểm sản phẩm mới được ra mắt của nhãn hàng sẽ thể hiện sự quan tâm và yêu thích của người đó tới nhãn hàng. Điều này sẽ giá trị hơn đối với việc có hành vi tương tác với sản phẩm tại các thời điểm nhãn hàng đang thực hiện chiến dịch tiếp thị.

Theo nghiên cứu của Đắc, Đắc đề suất sử dụng công thức như dưới đây để tính PP score:

$$PPscore = ((p+\frac{z^2\alpha/2}{2n})/(1+\frac{z^2\alpha/2}{n})-\sqrt{(\frac{p(1-p)}{n}+\frac{z^2_{\alpha/2}}{4n^2})}/(1+\frac{z^2\alpha/2}{n}))*\beta+log_{10}(totalPosts) \tag{1.1}$$

- Với $p = \frac{k}{n}$, p là phần trăm số positive interaction trên tổng số interaction, k là tổng số positive interaction.
- $\beta = \frac{Activative.Day}{reporting.Day-start.Day}$, β là mức độ chuyên cần.
- $k = w_p \sum PositivePost$, k là tổng số positive interactions.
- $w_p = 0.86$, trọng số của post.
- $w_s = 0.5$, trọng số của Share.
- $w_c = 0.64$, trọng số của Comment.
- $n = \sum (Post + Share + Comment)$, n là tổng số interactions.
- $z^2_{\alpha/2} = 1.96$

Tuy nhiên trong quá trình tính PP score với data thực tế thì công thức trên có 1 vài vấn đề. Nên công thức đã được đơn giản hóa đi và cải thiện hơn như sau.

$$PPscore = (Wilson\ score + \sum \log_{10}(totalPosts + 1) + \log_{10}(mean\_like + 1)) * \beta$$

$$Wilson\ score = ((p+\frac{z^4_{\alpha/2}}{2*totalPosts})-z^2_{\alpha/2}*\sqrt{(\frac{p(1-p)+\frac{z^4_{\alpha/2}}{4totalPosts}}{totalPosts})}*\frac{1}{1+\frac{z^4_{\alpha/2}}{totalPosts}} \tag{1.2}$$

- Với $p = \frac{k}{n}$, p là phần trăm số positive interaction trên tổng số interaction, k là tổng số positive interaction.
- $\beta = \frac{Activative.Day}{reporting.Day - start.Day}$, β là mức độ chuyên cần.
- $totalPosts$ là tổng số posts.
- $mean\_like$ là trung bình số like của tất cả các bài posts của một người dùng.
- $z^2_{\alpha/2} = 1.96$

2 Diễn giải về script

Trước hết cần phải import các library cần thiết đặc biệt là pymongo bởi vì database là MongoDB. Và sau đó là tạo kết nối với database, trong đây cần phải kết nối 2 database vì sẽ cần sử dụng database đã đánh mentioned.

```
import pandas as pd
import numpy as np
import datetimedatabase
from pymongo import MongoClient


client1 = MongoClient('45.122.223.198:27017',       #IP address of database
                  username = 'kapiReadOnly',        #Username
                  password = 'pl2oieAt9#tnWV!Yc0',  #Password
                  authSource = 'kapi',              #name of database
                  authMechanism = 'SCRAM-SHA-1')
kapi = client1['kapi']


client = MongoClient('45.122.223.198:27017',       #IP address of database
                  username = 'kpi-v2R',             #Username
                  password = 'EecvKJxdTQ1JEK8J2FA', #Password
                  authSource = 'kpi-v2',            #name of database
                  authMechanism = 'SCRAM-SHA-1')
kpitest = client['kpi-v2']
```

Tiếp đến là query data cần thiết để tính, trong đó cần hai function query data. Function đầu tiên dùng để query thông tin cần thiết có liên quan tới tất cả các bài post của một người dùng trong một khoảng thời gian nhất định.

```
def get_gnl_pst(user_list, start_date, end_date):
    cluster = kapi['posts'].find({'to_user': {'$in': user_list},
```

```
        'created_date': {'$gte': start_date, '$lt':
            end_date},'sentiment':{'$ne':''}
                                    },{
        '_id': 0, 'from_user': 1, 'to_user': 1,
        'created_date': 1, 'user_id': 1, 'likes_count': 1
    })

    tb = pd.DataFrame()
    for a in cluster:
        tmp = pd.DataFrame([a])
        tb = tb.append(tmp, ignore_index=True)

    gc.collect()
    return tb
```

Function thứ hai là dùng để query các post có đánh sentiment. Trong đó, theo Đắc thì chỉ lấy những post có đánh sentiment bằng 1.

```
def get_smt_pst(user_list,industry, start_date, end_date):
    cluster = kpiv2['posts'].find({'to_user': {'$in': user_list},
        'industry': industry, 'parent_id': None, 'brand': {'$ne': None},
        'created_date': {'$gte': start_date, '$lt': end_date},'sentiment':
            {'$ne': ''}
                                    },{
        '_id': 0, 'from_user': 1, 'to_user': 1, 'created_date': 1,
        'brand': 1, 'sentiment': 1,
    })

    tb = pd.DataFrame()
    for a in cluster:
        tmp = pd.DataFrame([a])
        tb = tb.append(tmp, ignore_index=True)

    gc.collect()
    return tb
```

Bước tiếp theo là dùng hai function ở trên để query dat chuẩn bị tính toán. Function query tất cả post của một người dùng đc lưu về "gnl_tb". Còn Function query những post mà đã được đánh sentiment được lưu về "smt_tol".

```
b_list = ["100001997853167", "100003714391961", "567112096",

start_date = datetime.datetime(2018,7,1) #Year, Month, Day of start date
```

```
end_date = datetime.datetime(2018,12,31) #Year, Month, Day of end date
industry = 'baby_milk_powder'

smt_tb = get_smt_pst(b_list, industry, start_date, end_date)
gnl_tb = get_gnl_pst(b_list, start_date, end_date)
```

Từ bảng "gnl_tb" và "smt_tb", dùng các lệnh aggregate các lệnh để standardize các cột để tạo bảng "cmb_tb". Trong bảng mới này chứa các thông tin cần thiết để tính PP score theo công thức.

```
gnl_tb['created_date'] = pd.to_datetime(gnl_tb['created_date'])
gnl_tb['date'] = [x.strftime('%Y-%m-%d') for x in gnl_tb['created_date']]
gnl_tb = (gnl_tb.groupby('to_user')
         .agg({'likes_count':'mean','date': pd.Series.nunique})
         .reset_index()
         .rename({'likes_count':'mean_like_cnt','date':'beta'}, axis='columns'))
gnl_tb['beta'] = gnl_tb['beta']/(end_date - start_date).days
gc.collect()


smt_tb['date'] = [x.strftime('%Y-%m-%d') for x in smt_tb['created_date']]
smt_tb = (smt_tb.groupby(['to_user', 'brand', 'date'])
        .agg({'created_date':'count', 'sentiment': lambda x: np.count_nonzero(x
           == '1')})
        .reset_index())
smt_tb = (smt_tb.groupby(['to_user', 'brand'])
        .agg({'sentiment': 'sum', 'created_date': [lambda x:
           np.sum(np.log10(x+1)), 'sum']})
        .reset_index())
smt_tb.columns = ['to_user', 'brand', 'positive_pst', 'n_adj', 'total_pst']

cmb_tb = smt_tb.join(gnl_tb.set_index('to_user'), on='to_user')
cmb_tb['p'] = (cmb_tb.positive_pst/cmb_tb.total_pst)

del gnl_tb, smt_tb
gc.collect()
```

Tiếp theo là define "pp_fun" được dùng để tính PP score từ bảng "cmb_tb" có được ở phía trên.

```
def pp_fun(total_pst, n_adj, p, mean_like_cnt, beta):
    return (
           (((p + 1.96**2/(2*total_pst)) - 1.96*np.sqrt((p*(1-p) +
```

```
            1.96**2/(4*total_pst))/total_pst))
          / (1 + 1.96**2/total_pst))
          + n_adj
          + np.log10(mean_like_cnt + 1)*p
         )*beta
```

Cuối cùng là sử dụng "pp_fun"" để tính PP score trên bảng "cmb_tb". Sau đó là lưu kết quả tính toán dưới dạng file csv.

```
cmb_tb['PP_scr'] = cmb_tb.apply(lambda row: pp_fun(row.total_pst, row.n_adj,
    row.p,
                                                   row.mean_like_cnt, row.beta),
                                axis=1)

gc.collect()
cmb_tb.to_csv('PP_scr_new.csv')
```