User Behavior Analysis for Taobao

• Project Background

The objective of this project is providing insights and suggestions based on the analysis of user behavior:

1. Analyze the common e-commerce metrics and establish a funnel model of user behavior conversion
2. Summarize user's behavior in time dimension
3. Explore the user's preference for different products and make marketing strategies
4. Identify core user groups and analyze the behavior of these users

This dataset contains about one million transaction records between November 25, 2017 and December 3, 2017. I did this project using Mysql and Tableau. A detailed description of each column :

| Column | Type |

| User ID | int |

|Product ID | int |

| Category | int |

| behavior | varchar including ('pv', 'buy', 'cart', 'fav') |

| time | int |

| behavior | description |

| pv | page view on product detail page|

| buy | purchase product|

| cart | add to shopping cart |

| fav | add to favorite|

• Data Cleaning

1. Delete duplicated records and checking missing value

```sql
SELECT * FROM `userbehavior`
group by user_id, item, category, time
having count(user_id)>1;
```

Message | Result 1 | Profile | Status

user_id	item	category	behavior	time
(N/A)	(N/A)	(N/A)	(N/A)	(N/A)

```sql
SELECT count(user_id), count(item), count(category),count(behavior), count(time)
FROM `userbehavior`;
```

Message | Result 1 | Profile | Status

count(user_id)	count(item)	count(category)	count(behavior)	count(time)
799999	799999	799999	799999	799999

2. Convert type of time

```sql
Alter Table userbehavior
ADD date VARCHAR(20),
ADD hour VARCHAR(20);

UPDATE userbehavior SET date = FROM_UNIXTIME(time,"%Y-%m-%d");
UPDATE userbehavior SET hour = FROM_UNIXTIME(time,"%H");
UPDATE usebehavior SET time = FROM_UNIXTIME(time);

```

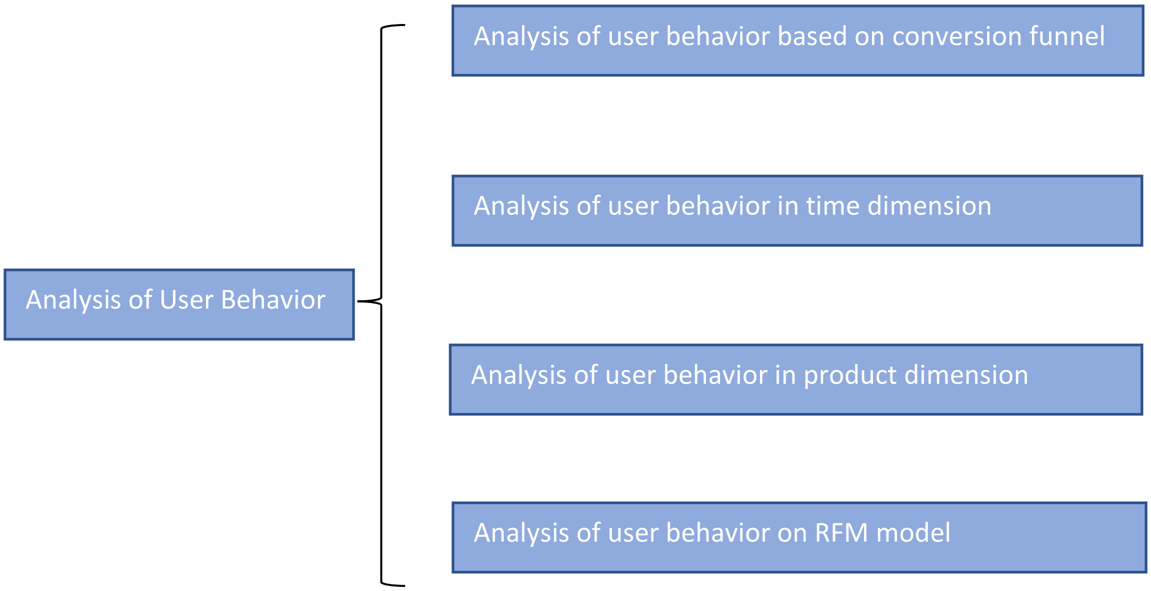
user_id	item	category	behavior	time	date	hour
1	2333346	2520771	pv	1511561733	2017-11-24	14
1	2576651	149192	pv	1511572885	2017-11-24	17
1	3830808	4181361	pv	1511593493	2017-11-24	23
1	4365585	2520377	pv	1511596146	2017-11-24	23
1	4606018	2735466	pv	1511616481	2017-11-25	05
1	230380	411153	pv	1511644942	2017-11-25	13
1	3827899	2920476	pv	1511713473	2017-11-26	08
1	3745169	2891509	pv	1511725471	2017-11-26	11
1	1531036	2920476	pv	1511733732	2017-11-26	14
1	2266567	4145813	pv	1511741471	2017-11-26	16
1	2951368	1080785	pv	1511750828	2017-11-26	18
1	3108797	2355072	pv	1511758881	2017-11-26	21

3. Filter records happened between '2017-11-25' and '2017-12-03'

```sql
#filter
delete from userbehavior
where date<'2017-11-25' or date> '2017-12-03';
```

- ## Data Analyzing

Analyzing Structure:



Analysis of User behavior on conversion funnel

a) Common e-commerce metrics: PV, UV, UV/PV

```sql
select count(distinct user_id) as UV,
       sum(case when behavior='pv' then 1 else 0 end) as PV,
       sum(case when behavior='buy' then 1 else 0 end) as Buy,
       sum(case when behavior='cart' then 1 else 0 end) as Cart,
       sum(case when behavior='fav' then 1 else 0 end) as Fav,
       sum(case when behavior='pv' then 1 else 0 end)/count(distinct user_id) as 'PV/UV'
from userbehavior;
```

Message | Result 1 | Profile | Status

UV	PV	Buy	Cart	Fav	PV/UV
7783	681492	15393	41720	20846	87.5616

The total number of user visit is 7783, page view is 681492, average page view per user is about 88.

b) Repurchase rate

```
# repurchase rate
select sum(case when buy_amount>1 then 1 else 0 end) as "number of user repurchase",
        count(user_id) as "total number of user",
        sum(case when buy_amount>1 then 1 else 0 end)/count(user_id) as "repurchase rate"

        from (select *, count(behavior) as buy_amount from userbehavior
             where behavior = 'buy' group by user_id)a;

```

Message | Result 1 | Profile | Status

number of user re	total number of us	repurchase rate
3385	5231	0.6471

The total number of repurchase user is 3385, total number of user is 5231 and repurchase rate is 64.71%. Generally speaking, the loyalty of user is high.

c) Bounce rate

```
#bounce rate
select count(*) as "number of user who only visit page one time"
from
(select user_id
from userbehavior
group by user_id
having count(behavior)=1)a;
```

Message | Result 1 | Profile | Status

number of user who only visit page one time
0

According to the result, in this nine-day period, no one left Taobao with only visiting the page once, and the bounce rate is 0. Bounce rate reflects whether the website or the app is attractive to the users and it is helpful hen we try to increase the retention rate. It is one of the most metrics to evaluate the quality of the website or the app. The result means that the product itself or the content in the product introduction page is sufficiently attractive to the users.

d) Conversion funnel

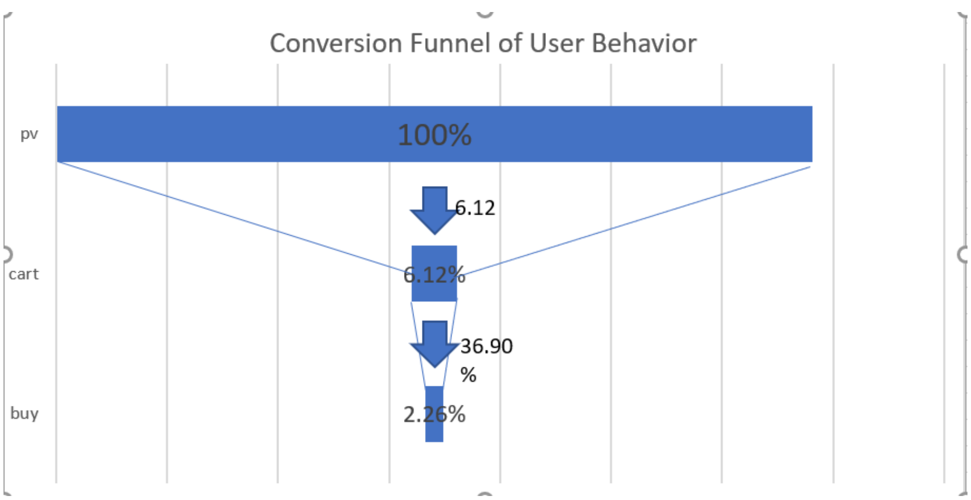
Considering the date we have, the conversion funnel here should be : Product detail page -> Shopping cart-> Payment page.

Conversion funnel of total:

```
#Conversion funnel
select behavior, count(*)
from userbehavior
group by behavior
order by behavior desc;
```

Message | Result 1 | Profile | Status

behavior	count(*)
pv	681492
fav	20846
cart	41720
buy	15393

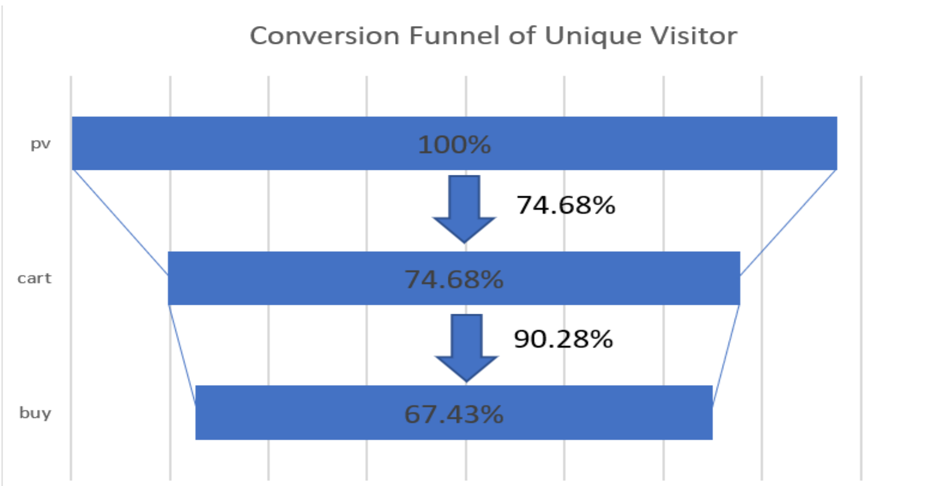


Conversion funnel of unique visitor:

```
#conversion funnel of unique visitor
select behavior, count(distinct user_id)
from userbehavior
group by behavior
order by behavior desc;
```

Message | Result 1 | Profile | Status

behavior	count(distinct user_
pv	7758
fav	3002
cart	5794
buy	5231



According to the conversion funnel, the conversion rate of total from product detail page to shopping cart page is only 6.12%. However, the conversion rate of unique visitor from product detail page to shopping cart page is 74.68%. This proves that users will click multiple times (681492/15392 = 44) of production detail page to compare before they add it to cart. The first improvement we can do here is trying to make accurate recommendation, which will decrease the user's cost in searching information.

For unique user, the number of users who make a payment takes place 67.43% of the numbers of unique user who click the product detail page. This proves that the purchasing conversion rate is pretty high and the products in Taobao can meets most users' requirements.

In sum, according to conversion funnel, we can provide two suggestions. One is optimizing recommendation system. Prove accurate recommendations according to users' preference and optimize rank of the searching results. The other one is highlight the key information which the user pay more focus on the product detail page which decreases the searching cost for the users.

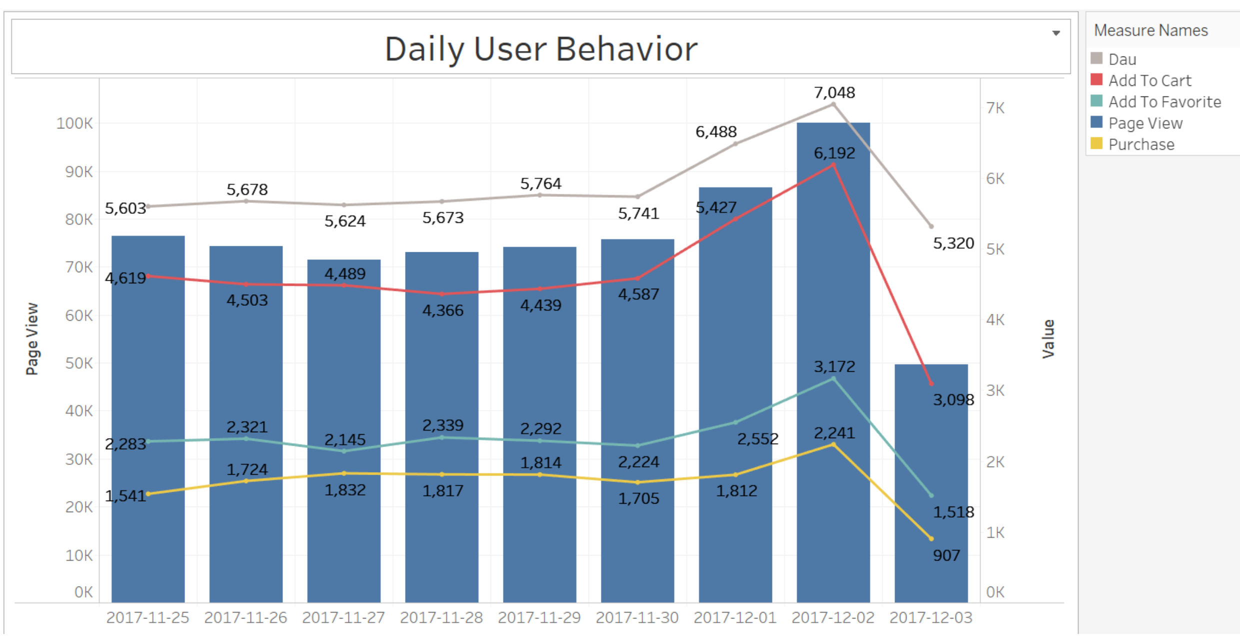
Analysis of user behavior in time dimension

a) Analysis of user behavior by day

```sql
#from the demension of time
select date,
        count(distinct user_id) as dau,
        sum(case when behavior ='pv' then 1 else 0 end) as 'page_view',
        sum(case when behavior ='cart' then 1 else 0 end) as
'add_to_cart',
        sum(case when behavior ='fav' then 1 else 0 end) as
'add_to_favorite',
        sum(case when behavior ='buy' then 1 else 0 end) as 'purchase'
from userbehavior
group by date;
```

Message | Result 1 | Profile | Status

date	dau	page_view	add_to_cart	add_to_favorite	purchase
2017-11-25	5603	76451	4619	2283	1541
2017-11-26	5678	74280	4503	2321	1724
2017-11-27	5624	71551	4489	2145	1832
2017-11-28	5673	73081	4366	2339	1817
2017-11-29	5764	74087	4439	2292	1814
2017-11-30	5741	75729	4587	2224	1705
2017-12-01	6488	86623	5427	2552	1812
2017-12-02	7048	99994	6192	3172	2241
2017-12-03	5320	49696	3098	1518	907



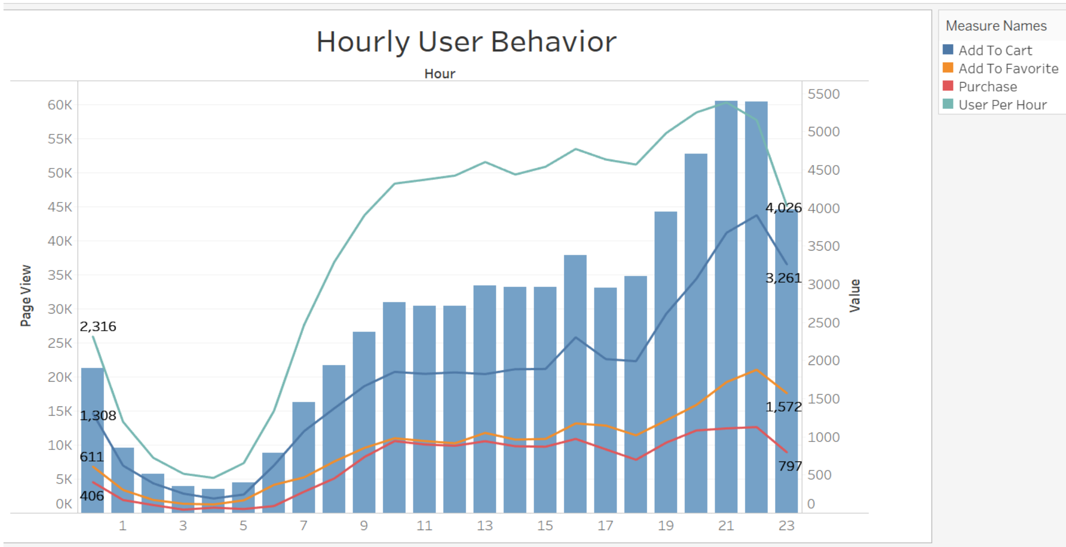
In this time period, 2017-11-25, 2017-11-26, 2017-12-02 and 2017-12-03 are weekends. The lines are stable between 11-25 and 12-01, but on 12-02, all the metrics increased obviously. (We don't extract all the information of 12-03, so we ignore it here). This may lead by the campaign we launched for Double 12 Festival.

b) Analysis of user behavior by hour

```
#from the demension of hour
select hour,
        count(distinct user_id) as user_per_hour,
        sum(case when behavior ='pv' then 1 else 0 end) as 'page_view',
        sum(case when behavior ='cart' then 1 else 0 end) as 
'add_to_cart',
        sum(case when behavior ='fav' then 1 else 0 end) as 
'add_to_favorite',
        sum(case when behavior ='buy' then 1 else 0 end) as 'purchase'
from userbehavior
group by hour;
```

Message | Result 1 | Profile | Status

hour	user_per_hour	page_view	add_to_cart	add_to_favorite	purchase
00	4774	37820	2305	1176	973
01	4636	33112	2019	1148	836
02	4570	34768	1992	1020	700
03	4983	44224	2613	1216	924
04	5253	52722	3072	1419	1084
05	5384	60446	3676	1718	1111
06	5152	60352	3905	1880	1128
07	4026	44506	3261	1572	797
08	2316	21308	1308	611	406
09	1197	9605	623	304	172
10	726	5753	391	173	106



According to the hourly user behavior chart. The lines are stable between 09:00 and 18:00. All the metrics arrives peak between 19:00 and 23:00. This conforms to users' daily routine. When we make strategies and campaigns, we may take the time regular into account.

Analysis of user behavior in product dimension

a) Top ten popular items by sales

```
#top ten items
select item, count(behavior) as "purchase_times"
from userbehavior
where behavior ='buy'
group by item
order by count(behavior) desc
limit 10;

```

Message | Result 1 | Profile | Status

item	purchase_times
3122135	12
2124040	11
4401268	10
2964774	8
4296993	7
11517	7
121226	7
3991727	7
1910706	7
1095113	7

According to the sell rank of products, we can see that the sales for each product is less than 12 and only three items' sales over 10. Therefor there is no hot sale product until now.

b) Top ten popular items by views

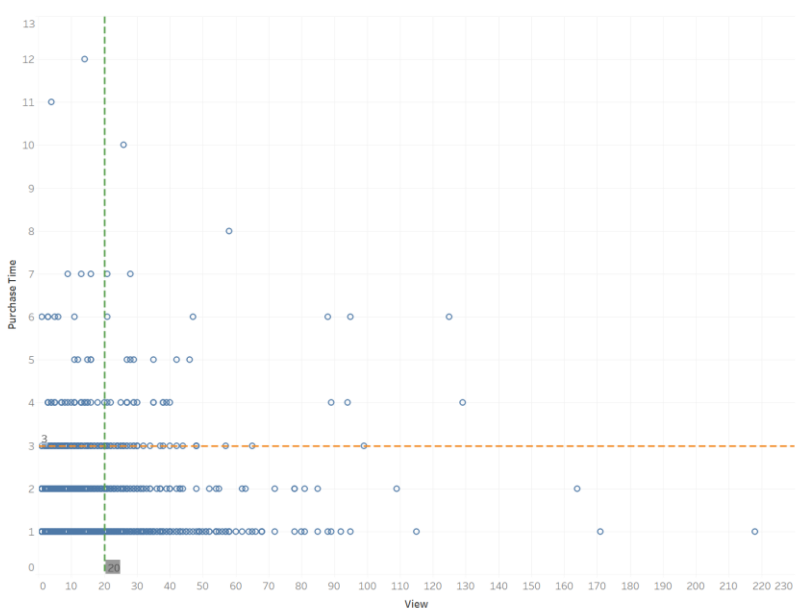
```
select item, count(behavior)
from userbehavior
where behavior = 'pv'
group by item
order by count(behavior)
limit 10;
```

Message | Result 1 | Profile | Status

item	count(behavior)
4606018	1
4092065	1
2903258	1
4666650	1
3682069	1
2266567	1
4615417	1
4973305	1
79715	1
2286574	1

Apparently, there is no union between hot items by views and hot items by sales. There is no positive relation between sales and views. So we should analyze from this two dimension at the same time.

c) Sales vs. Views



According to experience, we set views 20 and sales 3 as references. The first quadrant represents the products that with high views and high sales. These are popular products, and the purchase conversion rate is high. For these products, we can make more promotions and try to attract more potential users.

The second quadrant represents for the products with high sales but low views. Two possibilities, one is these products are necessities for some specific users. and the other one is lots of users would like to buy it but lack of traffic. We may make improvement based on customers information and products information. If the first possibility is true, then we may make recommendations to the specific group user. If the second suppose is true, we may increase exposure.

The third quadrant represents for the products with low views and low sales. One possibility for this is the traffic is low. And the other one is these products don't meet the requirements of users.

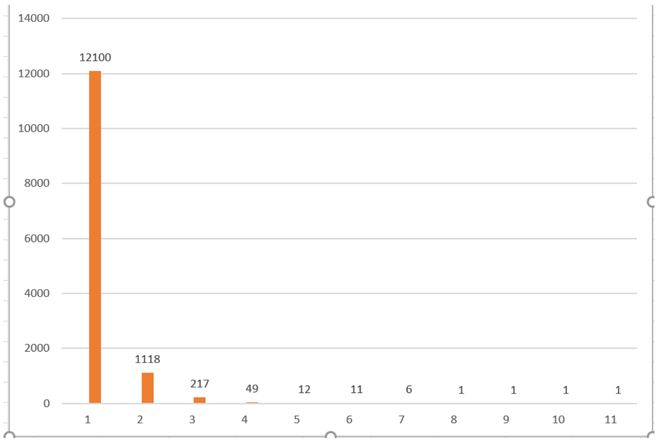
The fourth quadrant is the products with high views but low sales. We need to check whether we put the advertisement to the right user group, or the price of the products is too high, or the introduction on the page details page is poor. There are lots of possibilities, and we can pay more attentions on these products.

d) Purchase times vs. Number of items

```sql
select a.purchase_time, count(a.item) as num_items
from
(select item, count(behavior) as 'purchase_time'
from userbehavior
where behavior ='buy'
group by item)a
group by a.purchase_time
order by count(a.item) desc;
```

Message | Result 1 | Profile | Status

purchase_time	num_items
1	12100
2	1118
3	217
4	49
6	12
5	11
7	6
12	1
8	1
10	1
11	1



The number of items purchased only once is 12100, which is about 89.52% of the total products. It proves that most e-commerce platform make profits based on long-tail products instead of hot style ones.

Analysis of user behavior based on RFM model

a) Recency Dimension

```
#R dimension
Create View r_value as
select user_id, min(time_difference) as R
from (
select user_id, DATEDIFF('2017-12-03', DATE) AS time_difference
from userbehavior
where behavior ='buy')a
group by user_id;

select user_id, R, case when R between 0 and 2 then 4
                        when R between 3 and 4 then 3
                        when R between 5 and 7 then 2
                        else 1 end  as R_Score
from r_value;
```

Message | Result 1 | Profile | Status

user_id	R	R_Score
100	6	2
1000001	1	4
1000011	8	1
100002	3	3
1000027	0	4
1000028	1	4

b) Frequency Dimension

```
#F dimension
Create View f_value as
select user_id, count(behavior) as F
from userbehavior
where behavior ='buy'
group by user_id;

select user_id, F, case when F between 1 and 10 then 1
                        when F between 10 and 20 then 2
                        when F between 20 and 30 then 3
                        else 4 end  as F_Score
from f_value;
```

Message | Result 1 | Profile | Status

user_id	F	F_Score
100	7	1
1000001	1	1
1000011	1	1
100002	1	1
1000027	2	1
1000028	4	1

c) Recency-Frequency Dimension

```
create view r_score as
select user_id, R,       case when R between 0 and 2 then 4
                         when R between 3 and 4 then 3
                         when R between 5 and 7 then 2
                         else 1 end  as R_Score
from r_value;

create view f_score as
select user_id, F,       case when F between 1 and 10 then 1
                         when F between 10 and 20 then 2
                         when F between 20 and 30 then 3
                         else 4 end  as F_Score
from f_value;

Create View rf_score as
select a.user_id, a.R_score,b.F_score, a.R_Score+b.F_Score as
RF_Score
from r_score a join f_score b on a.user_id = b.user_id;
```

user_id	R_score	F_score	RF_Score
100	2	1	3
1000001	4	1	5
1000011	1	1	2
100002	3	1	4
1000027	4	1	5
1000028	4	1	5
1000037	4	1	5
1000054	4	1	5

RF-Score User Group

F (4-6)	Retaining User	Loyal User
F (1-3)	Lost User	Promising User
	R (1-2)	R (3-4)

```
## user_group_count
select user_group, count(*) as user_amount
 from (select *, case when RF_Score between 2 and 3 then 'lost user'
               when RF_Score between 4 and 5 then ' retaining user'
               when RF_Score between 6 and 7 then 'promising user'
               else 'loyal user' end as 'user_group'
from rf_score)a
 group by user_group;
```

Message | Result 1 | Profile | Status

user_group	user_amount
lost user	1320
retaining user	3809
promising user	99
loyal user	3

As we don't have transaction fees for each record, we cannot analyze by Monetary. The proportion of promising user is only 1.89%. We could send e-mails or send coupons to these users to prompts them to buy. The ratio of loyal user is least, but these are the most valuable users for Taobao. We'd better make strategies target these users to keep user stickiness. The retaining user number is the largest, and we may inspire them purchase by price incentive, collocations recommendations and some other strategies to keep them. For the loss users, maybe they found some other platforms to replace Taobao or are not interested in the products in Taobao anymore. We need to design strategies to recall them back.

d) High business value user

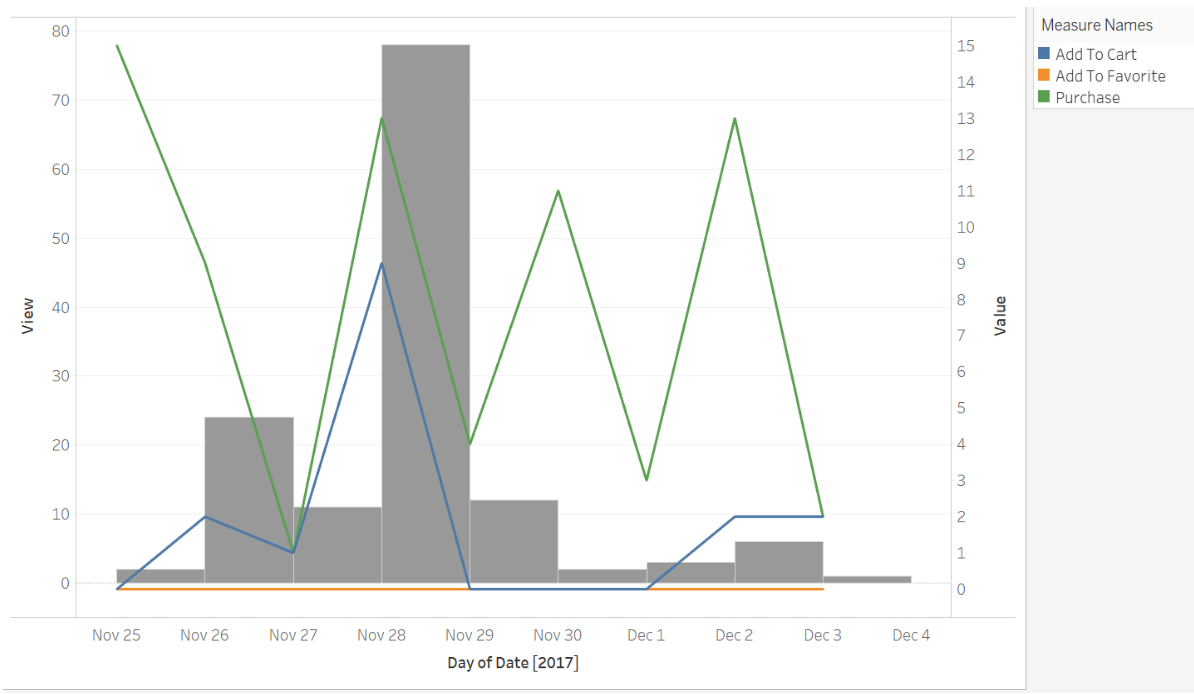
We take the user '107932' as an example.

```
#high user
select date,
sum(case when behavior='pv' then 1 else 0 end) as 'view',
sum(case when behavior='cart' then 1 else 0 end) as 'add_to_cart',
sum(case when behavior='fav' then 1 else 0 end) as 'add_to_favorite',
sum(case when behavior='buy' then 1 else 0 end) as 'purchase',
sum(case when behavior='buy' then 1 else 0 end)/sum(case when behavior='pv' then 1 else 0 end) as 'ourchase conversion'

from userbehavior
where user_id =107932
group by date;
```

Message Result 1 Profile Status

date	view	add_to_cart	add_to_favorite	purchase	ourchase conversic
2017-11-25	2	0	0	15	7.5000
2017-11-26	24	2	0	9	0.3750
2017-11-27	11	1	0	1	0.0909
2017-11-28	78	9	0	13	0.1667
2017-11-29	12	0	0	4	0.3333
2017-11-30	2	0	0	11	5.5000
2017-12-01	3	0	0	3	1.0000
2017-12-02	6	2	0	13	2.1667
2017-12-03	1	2	0	2	2.0000



This user almost purchased on Taobao every day, but he never added to favorite. For the features that are not used often, we should make exploration and make a decision on whether try to improve it or abandon it.