User Behavior Analysis of TaoBao

7201 Project

MAJOR: Data Science
TEAM MEMBER:
Wang Xindi （MB955075）
Zhou Yiren （MB955512）
Wang Zhe （MB955343）

Dec.2019

1. Introduction

With the development of Internet, online shopping has become a fashion. We usually care about products and activities in online shopping platform but never pay attention to the data. A few weeks ago, thanks to the course requirement, we registered account of Alicloud and found a user behavior data of taobao released by alibaba. Then we took this as data set and used knowledge of python we learned to analyze the user behavior of Taobao during that time.By traking and record a series of user behaviors like clicks, collection and add to chart, order, payment,etc.to monitor and focus on user in the process of commodity purchase. Enterprises should also quickly lock user needs in order to make correct judgement and accurate operation. Data includes user's behavior of browsing, collecting, adding to shopping cart, and purchasing as well as the corresponding time and type of products.

2. Understand the data

There are about 1.04 million pieces of data in this data set, with a total of 6 columns as follows:

User_id:user identity;

Item_id:item ID;

Behavior_type: user behavior type(including click, favorite, add to shopping cart, and pay, represented by the Numbers 1, 2, 3, and 4);

User_geohash: geographic location;

Item_category: category ID (category of goods);

Time: the time when user behavior occurs

Libraries we introduced include pandas, numpy,datetime, matplotlib and seaborn

Seaborn is an extend to matplotlib, a data visualization library that provides more advanced API encapsulation for ease and flexibility in applications. It can be used to set the graphics display and has a color palette function.

3. Whole analysis:

3.1 Data cleaning

```
#对时间特征处理，添加每日时间段，星期特征
df.time = pd.to_datetime(df['time'])
df['daily'] = df['time'].dt.time
df['weekday'] = df['time'].dt.weekday
df['date'] = df['time'].dt.date
df.head()
```

```
df['date']=pd.to_datetime(df['date'])
df['time']=pd.to_datetime(df['time'])
df.dtypes
```

```
user_id                    int64
item_id                    int64
behavior_type              int64
user_geohash              object
item_category              int64
time              datetime64[ns]
daily                     object
weekday                    int64
date              datetime64[ns]
dtype: object
```

```
df=df.sort_values(by='time',ascending=True)#排序处理
df=df.reset_index(drop=True)#建立索引
df.describe()
```

	user_id	item_id	behavior_type	item_category
count	1.225691e+07	1.225691e+07	1.225691e+07	1.225691e+07
mean	7.170732e+07	2.023084e+08	1.105271e+00	6.846162e+03
std	4.122920e+07	1.167397e+08	4.572662e-01	3.809922e+03
min	4.913000e+03	6.400000e+01	1.000000e+00	2.000000e+00
25%	3.584965e+07	1.014130e+08	1.000000e+00	3.721000e+03
50%	7.292804e+07	2.021359e+08	1.000000e+00	6.209000e+03
75%	1.073774e+08	3.035405e+08	1.000000e+00	1.029000e+04
max	1.424559e+08	4.045625e+08	4.000000e+00	1.408000e+04

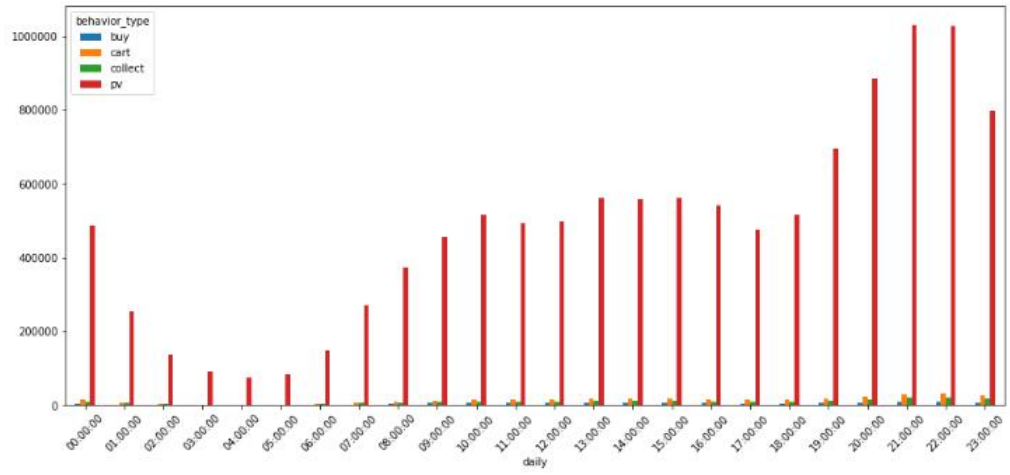
3.2 Time dimension analysis

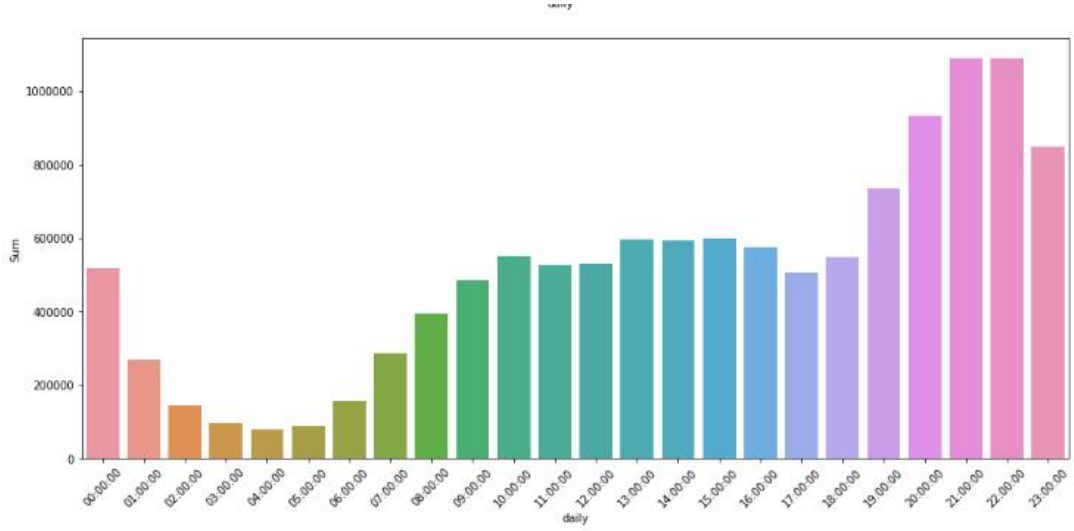
3.3.1 Analyze the whole situation of a day

```
In [27]: table_time = df.groupby(['daily','behavior_type']).size().unstack()
         table_time1 = df.pivot_table(index='daily',columns='behavior_type',aggfunc='count',margins=True,margins_name='Sum')
         table_time1 = table_time1['date']

         table_time.plot.bar(stacked=False,rot=45,figsize=(16,7))
         plt.figure(figsize=(16,7))
         sns.barplot(x=table_time1.index[:-1],y = table_time1['Sum'][:-1])
         plt.xticks(rotation=45)
```

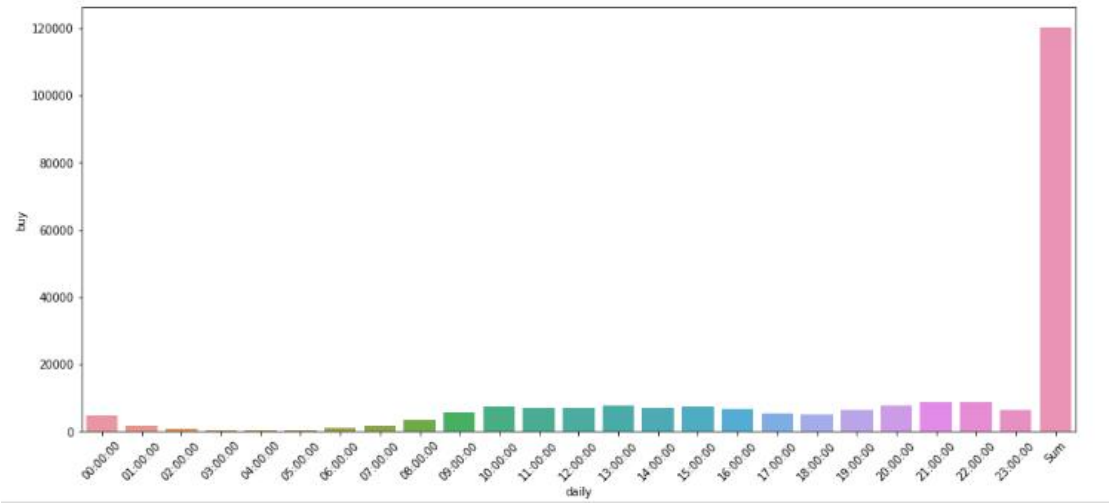
```
Out[27]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                 17, 18, 19, 20, 21, 22, 23]), <a list of 24 Text xticklabel objects>)
```





According to the summary statistics conducted at different time periods every day, user behaviors were relatively stable in the daytime, while it surged from 5 p.m. and continued until 11 p.m., indicating that it was the peak time for users to have user behaviors at night. However, since the pv volume is significantly higher than the occurrence of other user behaviors, it is necessary to conduct research according to different behaviors.

```
for i in table_time1.columns:
    plt.figure(figsize=(16,7))
    sns.barplot(x=table_time1.index,y = table_time1[i])
    plt.xticks(rotation=45)
```

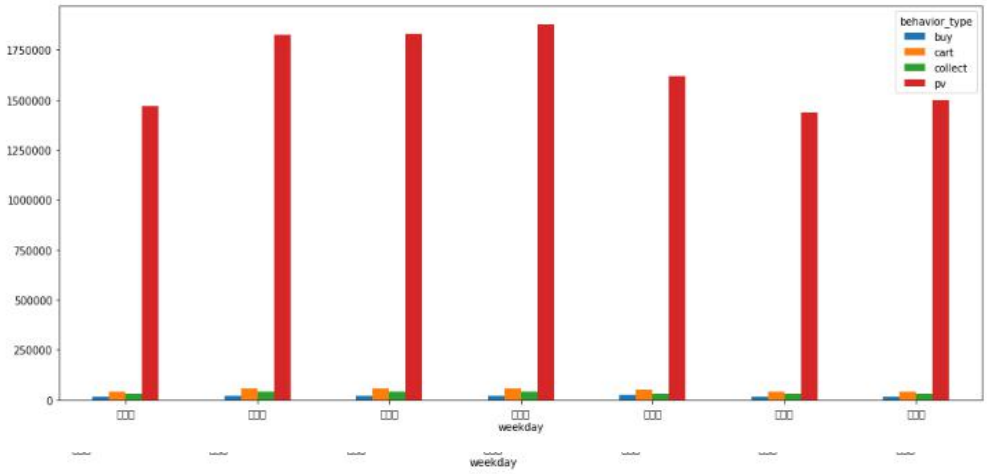


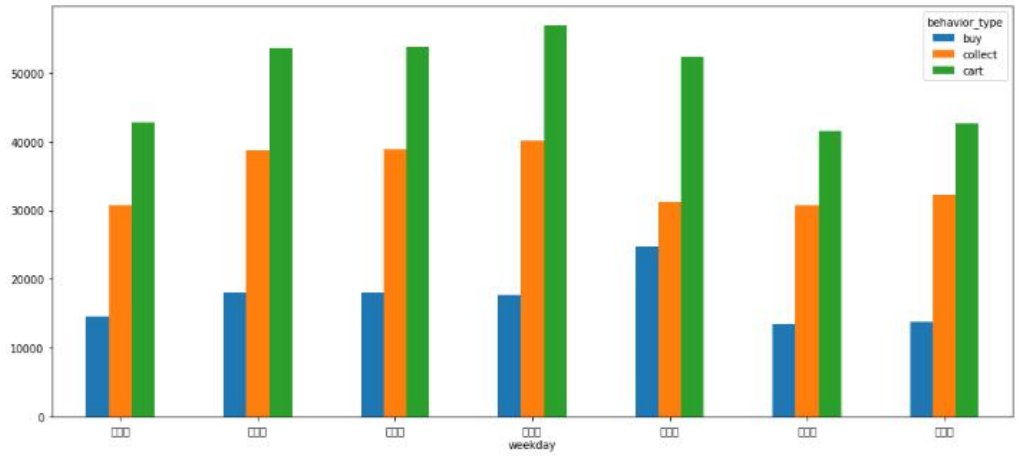
It can be seen that the number of clicks, favorites and additions to the shopping cart increased significantly at night, but the increase in purchase at night was not as significant as other user behaviors. In this aspect, following explanation may be applied: firstly, users may have a lot of time to browse the app in the evening, but they may not order and pay immediately; During the day, part of the demand comes from the daily needs of work, which can make the order quantity increase.

3.3.2Analyze the whole situation of a week

```
In [12]: table_week = df.groupby(['weekday','behavior_type']).size().unstack().reset_index()
         table_week['weekday'] = table_week['weekday'].replace([0,1,2,3,4,5,6],['星期日','星期一','星期二','星期三','星期四','星期五','星期六'])
         table_week = table_week.set_index('weekday')
         table_week.plot.bar(rot=0,figsize=(16,7))

         week_withoutpv = table_week.loc[:,['buy','collect','cart']]
         week_withoutpv.plot.bar(rot=0,figsize=(16,7))
Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x25e13c8c1c8>
```





We can see the pv volume is higher in weekdays than on weekends, which is different from the original expectation. The day of 12-12 was Thursday, which also had a certain influence on the result. It can be seen that the purchase volume of Thursday was significantly higher than that of other times, and the decrease of collection volume on that day was also related to the Thursday of 12-12.

From the time dimension, we can draw the following conclusions:

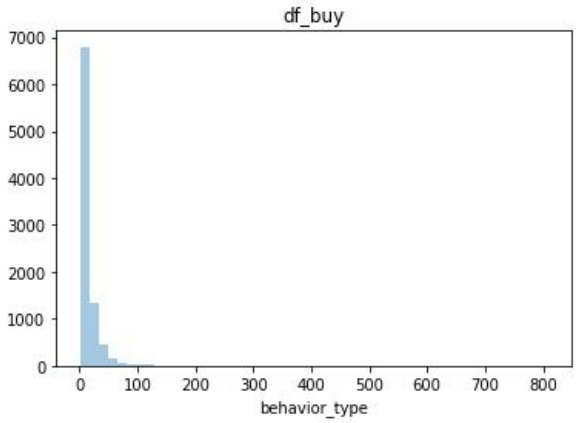
a. Users generally browse, collect and purchase the most commodities from 7 p.m. to 11 p.m., but the purchase volume during this period is not significantly higher than that during the day;

b. Promotion activity has significantly changed the user behavior in a period of time. According to the type of activity, browsing, collecting and adding to the shopping cart behaviors of the users in the days before 12-12 have increased dramatically, while the purchase volume on the day of 12-12 has doubled.

c. Purchase volume of non-working days is lower than that of working days.

3.3.3Analyze the daily user purchase times

```
In [70]: df_buy=df[df.behavior_type==4].groupby('user_id')['behavior_type'].count()
         sns.distplot(df_buy,kde=False)
         plt.title('df_buy')
Out[70]: Text(0.5, 1.0, 'df_buy')
```

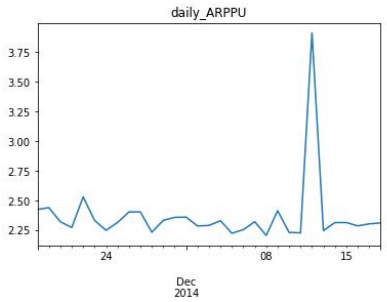


The chart shows that the purchase times of users are generally less than 10 times, so we need to focus on the purchase times of more than 10 times.

3.3.4Analyze Average Revenue Per User

Average revenue per user, which can be calculated by the Total Revenue /AU, which can measure the profitability and development vitality of the product.

```
In [74]: df_buy1=df[df.behavior_type==4].groupby(['date','user_id'])['behavior_type'].count().reset_index().rename(columns={'behavior_type':'total'})
         df_buy1.groupby('date').apply(lambda x:x.total.sum()/x.total.count()).plot()
         plt.title('daily_ARPPU')
Out[74]: Text(0.5, 1.0, 'daily_ARPPU')
```



Average number of purchases by active users=Total consumption/Number of active users

(Number of active users:Those who have operational behaviors every day are active)

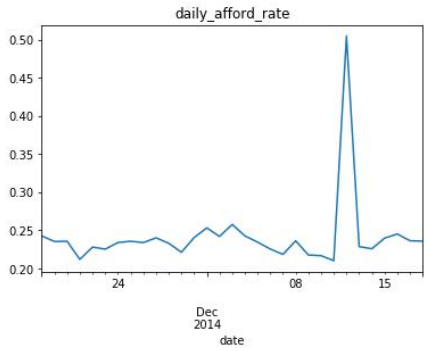
It shown that spending power peaks at 12 noon

3.3.5Analyze the Daily Pay Rate

Pay Rate=Number of real consumption/Number of active users

```
In [75]: df['operation']=1
         df_buy2=df.groupby(['date','user_id','behavior_type'])['operation'].count().reset_index().rename(columns={'operation':'total'})
         df_buy2.groupby('date').apply(lambda x:x[x.behavior_type==4].total.count()/len(x.user_id.unique())).plot()
         plt.title('daily_afford_rate')

Out[75]: Text(0.5, 1.0, 'daily_afford_rate')
```



It shown that Daily Pay Rates also peaked at 12 p.m

4. Conclusion:

1) Merchants should grasp users' behaviors and habits, conduct accurate operation according to different types of users, and make personalized recommendation;

2) Merchants should carry out new promotion activities and visit old users according to the time dimension. Stimulate users' continuous consumption according to the repurchase rate, monitor users' continuous user behavior according to the retention, prevent loss and stimulate users' behavior to some extent;

3) Preferential policies should be provided for high-value users to keep them active, specific strategies should be developed for different user behavior groups to stimulate further user behavior, and corresponding measures should be taken to stimulate or improve activity for inactive users and most low-purchase users.