

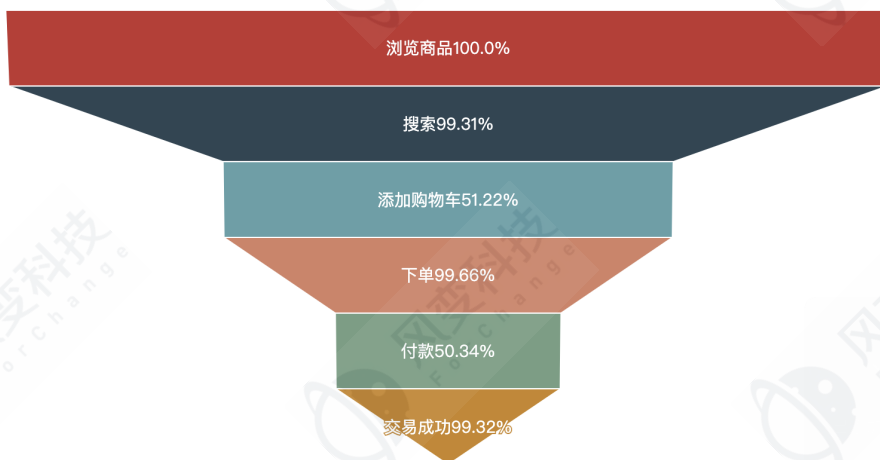
漏斗分析法

一、简介

漏斗分析法，从字面上看就是用一种类似漏斗的框架对事物进行分析的一种方法。

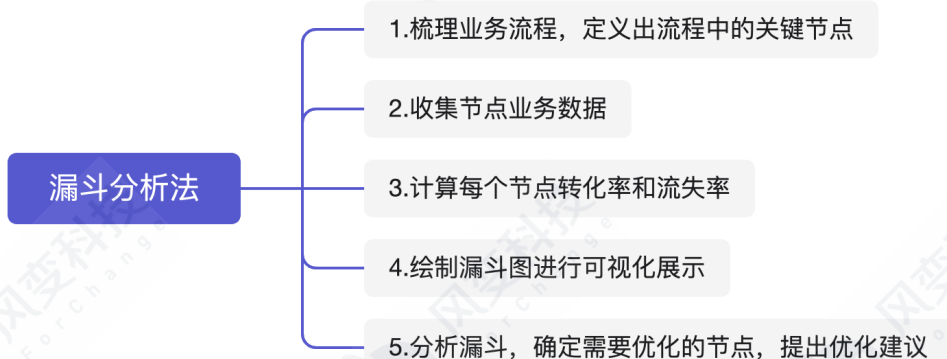
它在互联网行业中被广泛运用，能够科学直观地反映用户行为状态，以及从起点到终点各阶段的用户行为数据，主要用于电商购物、APP获客与增长、用户消费决策分析等领域。

转化漏斗 交易成功99.32% 添加购物车51.22% 搜索99.31% 浏览商品100.0% 下单99.66% 付款50.34%



漏斗分析法主要将业务流程按照客观事实拆解成多个步骤，通过分析各个步骤之间的关系来分析我们实际业务流程中的问题，帮助我们找到流程中的指标弱项，并且针对该指标弱项进行流程优化，最终提升整个项目的收益。

二、分析流程



构建一个漏斗分析模型，通常分为以下5个步骤：

- 1.梳理业务流程，定义出流程中的关键节点
- 2.收集节点业务数据
- 3.计算每个节点转化率和流失率
- 4.绘制漏斗图进行可视化展示
- 5.分析漏斗，确定需要优化的节点，提出优化建议

其中第一点，梳理业务流程，应该实际按照公司业务的实际情况，根据用户的路径和时间的先后顺序进行梳理。

三、案例讲解

接下来，我们将按照上面的分析流程，使用Python进行数据分析和数据可视化，来对一个电商平台的业务流程进行分析。

1.梳理业务流程，定义出流程中的关键节点

相信大家都有在淘宝、京东等购物网站上购物的经历，那我们就尝试着来拆解一下一个电商平台的业务流程，并且提取出关键节点。

首先，假设一个用户进入电商平台是没有目的的，一般来说他会先【浏览一下平台推荐的商品】，通过平台推荐的商品对某个商品产生兴趣。建立起兴趣之后，用户将会使用平台的搜索功能对同类型商品进行【搜索】，通过不断的搜索挑选出一个满意的产品，并【添加购物车】。

在添加购物车之后，用户就会进行【下单】和【付款】，最后【交易成功】。

通过我们刚刚梳理的业务流程，我们便可以将一个用户从进入平台到交易成功拆解成6个行为阶段（也就是节点）：浏览商品-搜索-添加购物车-下单-付款-交易成功。



2.获取节点业务数据

节点（也就是用户的行为阶段）的业务数据一般由开发部门在产品中进行获取，这里我们准备了一份数据给大家。



漏斗转化分析.xlsx
0.1MB



漏斗转化讲解.ipynb
20.4KB

3.计算每个节点转化率

在计算转化率之前，我们先来看看转化率的一些基础知识。

转化率是上一个流程节点到下一个流程节点的比率，它为了表示相邻环节的承接效果。它的计算方法是：

$$1 \quad \text{转化率} = \text{当前节点人数} / \text{上个节点人数}$$

$$\text{转化率} = \text{当前节点人数} / \text{上个节点人数}$$

我们使用Pands库来计算转化率

(1) 读取完数据后，我们使用value_counts()方法统计每个行为阶段的数量，也就是我们每个流程节点的人数。

```
[1] > %> R
import pandas as pd

[2] > %> R
# 读取文件
df = pd.read_excel('...', '漏斗转化分析.xlsx')
df

# 统计每个行为阶段的数据，并设置索引
action_df = df['行为阶段'].value_counts().reset_index()
# 更改列名为“行为阶段”和“数量”
action_df.columns = ['行为阶段', '数量']
action_df
```

行为阶段	数量
浏览商品	588
搜索	576
添加购物车	295
下单	294
付款	148
交易成功	147

(2) 根据转化率公式，我们还需要将上一个节点的数量进行整理。这里我们使用for循环和if语句来构造。

要注意的是，由于第一个节点（浏览商品）没有上一个节点，所以我们以它自己的数量填入。



```
[4] ▶ ML  
# 获取每个行为阶段的数量  
action_list = list(action_df['数量'])  
action_list
```

```
[580, 576, 295, 294, 148, 147]
```

```
[5] ▶ ML  
# 计算上一个行为阶段的数量  
action_list2 = []  
for a in range(len(action_list)):  
    if a == 0:  
        action_list2.append(action_list[0])  
    else:  
        action_list2.append(action_list[a-1])  
action_list2
```

```
[580, 580, 576, 295, 294, 148]
```

```
[6] ▶ ML  
# 将上个行为阶段数量添加进DataFrame  
action_df['上个行为阶段数量'] = action_list2  
action_df
```

	行为阶段	数量	上个行为阶段数量
0	浏览商品	580	580
1	搜索	576	580
2	添加购物车	295	576
3	下单	294	295
4	付款	148	294
5	交易成功	147	148

(3) 整理完上一个节点的数量后，我们就可以根据转化率公式进行计算，获得当前节点的转化率了。

```
[7] ▶ ML  
# 计算转化率  
action_df['转化率'] = action_df['数量'] / action_df['上个行为阶段数量']  
action_df
```

	行为阶段	数量	上个行为阶段数量	转化率
0	浏览商品	580	580	1.000000
1	搜索	576	580	0.993103
2	添加购物车	295	576	0.512153
3	下单	294	295	0.996610
4	付款	148	294	0.503401
5	交易成功	147	148	0.993243

到这里，我们就获得了我们需要的转化率了。

但是我们的分析还是为了展示并用于优化业务的，单纯从数字看不直观，所以我们来对数据进行数据可视化的操作，绘制漏斗图，使数据更直观。

4.绘制漏斗图

由于我们学习的pandas库和matplotlib库都不能比较方便地画出漏斗图，所以我们将使用一个新的可视化库pyecharts库来绘制漏斗图。

这里我们不对pyecharts库做详细讲解，有兴趣的同学可以点击下方链接，进入官网学习。

https://pyecharts.org/#/zh-cn/basic_charts?id=funnel%ef%bc%9a%e6%bc%8f%e6%96%97%e5%9b%be

这里我们构造画图所需要的数据和标签。

```
[8] ▶ ML
# 导入用于设置的方法
from pyecharts import options as opts
# 导入漏斗图类
from pyecharts.charts import Funnel

[9] ▶ ML
# 构造标签字典
action = list(action_df['行为阶段'])
conversion = list(action_df['转化率'])
label = []
# 将转化率构造百分比形式
for i in range(len(action_list2)):
    l = action[i] + str(round(conversion[i] * 100, 2)) + '%'
    label.append(l)
label

['浏览商品100.0%', '搜索99.31%', '添加购物车51.22%', '下单99.66%', '付款50.34%', '交易成功99.32%']

[10] ▶ ML
# 组装标签和数据
data = list(zip(label, list(action_df['数量'])))
data

[('浏览商品100.0%', 580),
 ('搜索99.31%', 576),
 ('添加购物车51.22%', 295),
 ('下单99.66%', 294),
 ('付款50.34%', 148),
 ('交易成功99.32%', 147)]
```

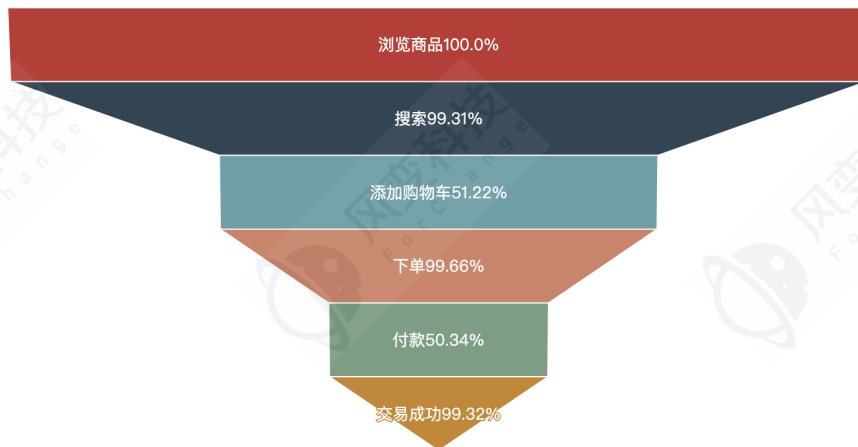
使用画图代码绘制漏斗图。

[11]

ML

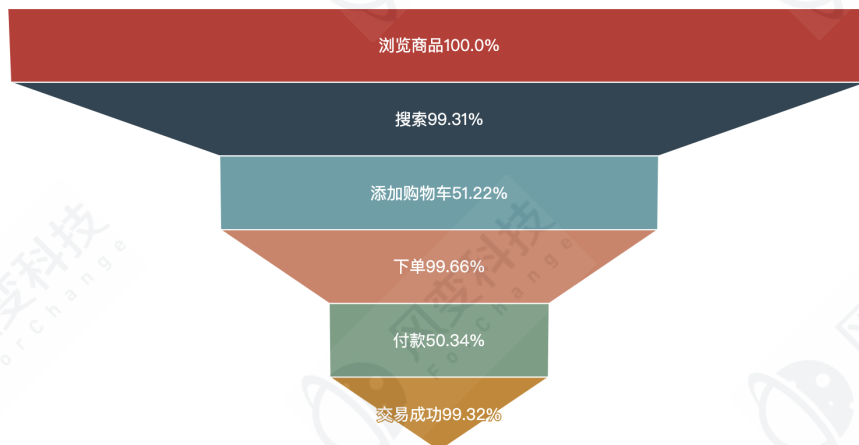
```
# 实例化漏斗图对象
funnel = Funnel()
# 添加数据
funnel.add(series_name='行为', data_pair=data)
# 设置图表标签位置
funnel.set_series_opts(label_opts=opts.TooltipOpts(is_show=True, position='inside'))
# 设置标题
funnel.set_global_opts(title_opts=opts.TitleOpts(title='转化漏斗'))
# 生成图表
# funnel.render('./转化漏斗pye.html')
funnel.render_notebook()
```

转化漏斗 交易成功99.32% 添加购物车51.22% 搜索99.31% 浏览商品100.0% 下单99.66% 付款50.34%



5.分析异常节点，提出优化建议

转化漏斗 交易成功99.32% 添加购物车51.22% 搜索99.31% 浏览商品100.0% 下单99.66% 付款50.34%

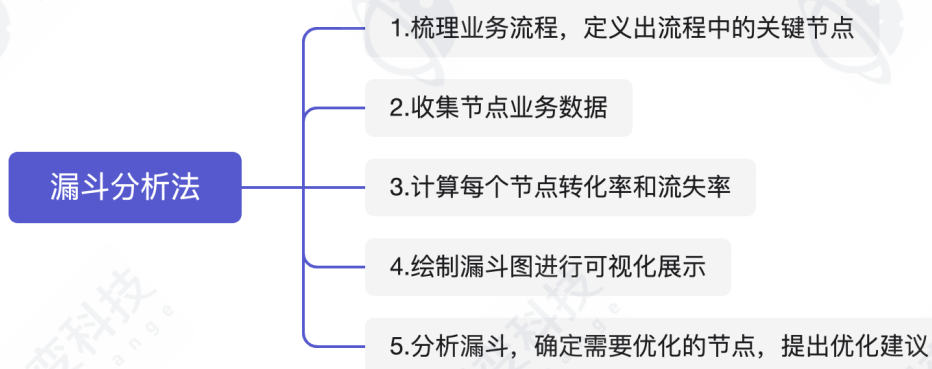


通过我们上一步绘制的漏斗图，我们可以看出从下单到付款这一步的漏斗偏大，也就是客户在这一步流失严重。

我们可以思考是否是由于系统问题导致下单后没有付款，或者是其他的问题导致。

四、总结

通过这个电商用户的漏斗分析，我们掌握了漏斗分析的全流程。



同时我们使用Python进行数据处理，并且使用Pyecharts库来绘制漏斗图。

漏斗分析法主要是一个分析流程，重点在于业务流程的梳理，需要对业务十分熟悉才可以准确梳理出关键节点。

【特别推荐】——风变Python学堂公众号

有Python知识干货、明星讲师直播、Python应用案例讲解等，帮大家学好Python，用好Python！
现在关注【风变Python学堂】，还可领取专属【资料包】，快扫下方二维码领取福利吧！

