一、课前准备

二、课堂主题

业务中的数据分析案例，数据分析职场技能与状况；

三、课堂目标

1. 真实业务场景中的数据分析；
2. 数据分析职业状态与职业规划；

四、知识要点

1. 业务中的数据分析案例

1.1 电商网站行为分析

1、需求：

1. 用户的购物行为情况；
2. 统计每天各行为的访问次数，并以折线图显示；
3. 找出购买率最高的前二十个商品品类，并以柱状图展示；

2、数据集介绍：

> 数据集包含了2017年11月25日至2017年12月3日之间，有行为的约一百万随机用户的所有行为（行为包括点击、购买、加购、喜欢），数据集的每一行表示一条用户行为，由用户ID、商品ID、商品类目ID、行为类型和时间戳组成，并以逗号分隔；用户行为类型共有四种，它们分别是：

1. pv：商品详情页点击；
2. buy：购买；
3. cart：加入购物车；
4. fav：收藏商品；

3、需求的实现过程：

> 分析数据 —> 整理并清除异常数据 —> 实现需求

1、分析数据

> 数据有3个多G，全部读取耗费时间与内存，因此使用pandas的迭代读取方法，先获取前一万条数据，熟悉数据内容，并加入列名

```
import pandas as pd
user_behavior = pd.read_csv('UserBehavior.csv',header=None,iterator=True)
head_data = user_behavior.get_chunk(10000)
head_data.head()
```

数据如下：

	0	1	2	3	4
0	1	2268318	2520377	pv	1511544070
1	1	2333346	2520771	pv	1511561733
2	1	2576651	149192	pv	1511572885
3	1	3830808	4181361	pv	1511593493
4	1	4365585	2520377	pv	1511596146

2、对数据进行整理、清洗

2.1、添加列名，数据特征分别是：用户ID，商品ID，商品类目ID，行为类型，时间戳

```
head_data.columns = ['user_id','goods_id','category_id','behaviour','timestamp']
```

2.2、查看是否有缺失值，有两种方法：

> 两种方式对比：

> 当数据集较大时，推荐使用第一种方式，虽然有些麻烦，但是节省时间和空间，size是一个对象的属性，调用时获取值的时间复杂度是O(1)；
>
> 第二种方式会将所有的数据过滤一遍进行判断，很慢，而且占用内存较大
>
> 可以看出以上数据没有缺失值，不需要进行缺失值的处理

```
# 1 对比每一列数据的大小
print(head_data['user_id'].size)
print(head_data['goods_id'].size)
print(head_data['category_id'].size)
print(head_data['behaviour'].size)
print(head_data['timestamp'].size)
```

```
# 2 直接用pandas提供的接口
head_data.isnull().any()
```

2.3、将时间戳转为时间格式，并新加一列time

```
head_data['time'] = pd.to_datetime(head_data['timestamp'],unit='s')
```

2.4、时间戳这列数据下面不会用到，可以删除掉

```
head_data = head_data.drop(['timestamp'],axis=1)
```

2.5、将time字段设为索引，目的是为了方便清除异常时间的数据

```
head_data.set_index('time',inplace=True)
```

2.6、数据是2017年11月25日至2017年12月3日之间，将异常时间的数据清洗掉

```
head_data = head_data['2017-11-25':'2017-12-3']
```

3、数据整理完，开始解决需求

3.1、需求1：统计用户的每个购物行为，思路：使用groupby进行分组并统计

```
count_by_user_behav = head_data.groupby(['user_id','behaviour']).count()
count_by_user_behav.head()
```

统计后数据形式如下：

		goods_id	category_id	timestamp
user_id	behaviour			
1	pv	55	55	55
100	buy	8	8	8
	fav	6	6	6
	pv	84	84	84
1000	cart	2	2	2

只取一列的次数，作为画图是的Y轴的值就可以了

```
count_by_user_behav = count_by_user_behav['goods_id']
count_by_user_behav.head()
```

```
user_id  behaviour
1        pv           55
100      buy           8
         fav           6
         pv           84
1000     cart          2
Name: goods_id, dtype: int64
```

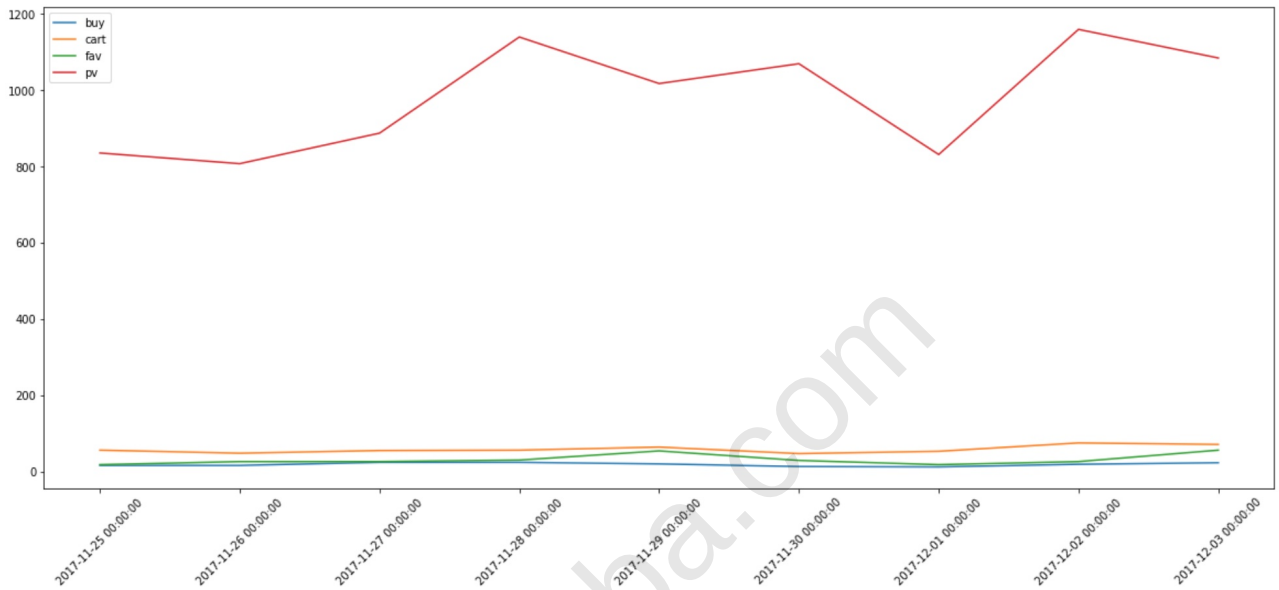
```
# 使用pandas自带的画图功能，因为用户很多，所以x轴就很长，双击图片可以放大
count_by_user_behav.plot(kind='bar', figsize=(150, 10))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x114b23198>
```

3.2、需求2 : 统计每天各行为的访问次数，并以折线图显示

```
# 接下来结合matplotlib画图
from matplotlib import pyplot as plt
%matplotlib inline
```

```
# 设置画布尺寸
plt.figure(figsize=(20, 8))

# 以行为类型进行分组
for group_name,group_data in head_data.groupby('behaviour'):

    # 对每天的行为进行统计，resample中的D表示天，也可以用H按小时统计
    count_by_day = group_data.resample('D').count()['behaviour']

    # 以日期作为X轴，以次数为y轴
    x = count_by_day.index
    y = count_by_day.values

    # 设置x，y轴数据以及每条线的标签名
    plt.plot(range(len(x)), y, label=group_name)

# 设置x轴的刻标以及对应的标签，rotation是设置标签的倾斜度
plt.xticks(range(len(x)), x, rotation=45)

# 对每条折线的含义进行标注,自动选择最佳位置
plt.legend(loc='best')
plt.show()
```



> 可以清晰看出四种购物行为的走势图

3.3需求3：找出购买率最高的前二十个商品品类，并以柱状图展示

> 思路：按商品种类进行分组，购买率 = 购买次数/(访问+购物车+收藏+购买)

```
# 取消以时间为索引
head_data = head_data.reset_index()

# 按商品种类进行分组
count_by_category_id = head_data.groupby('category_id')

# 分组后每组数据的数量
count_by_category_id.size()
```

```
5053508      4
5071267     16
5091223      2
5103246      1
5140516      1
5150761      1
Length: 942, dtype: int64
```

> length的意思是在这一万条数据中，出现了942个商品种类

看一下分组后的部分数据

```
for group_name,group_data in count_by_category_id:
    print('组名：',group_name)
    print('数据：')
    print(group_data)
    print(group_data.shape)
    break
```

```
组名：  8254
数据：
                      time  user_id  goods_id  category_id behaviour
7109 2017-11-25 01:18:30  1000326   4367195         8254         pv
7114 2017-11-25 02:55:26  1000326   4367195         8254         pv
(2, 5)
```

> 思路：通过shape看出这组数据的形状为两行五列，因此可以直接用shape[0]作为购物的总次数使用，接下来再以behaviour分组，通过size属性，获取buy这一组的次数，最后将商品种类与购买率一一对应保存在字典中，使用内置函数sorted对字典进行排序

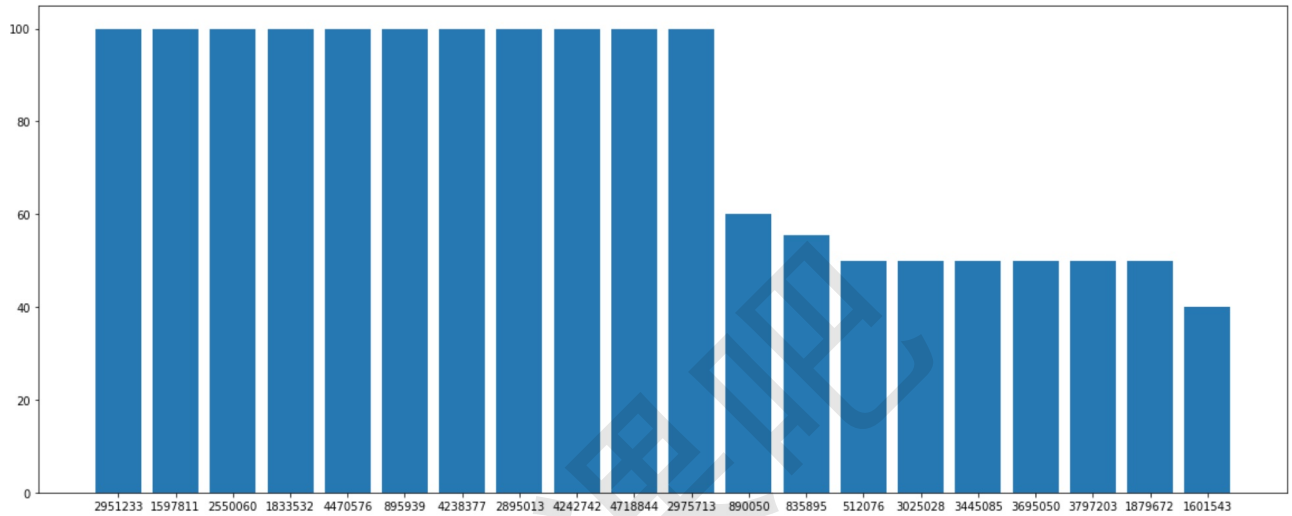
```
con_dict = {}
for group_name,group_data in count_by_category_id:
    # 总次数
    total = group_data.shape[0]
    buy = 0
    try:
        # 有些商品没有购买行为，以buy作为索引获取时会出错，使用异常语句捕捉一下
        buy = group_data.groupby('behaviour').size()['buy']
    except:
        pass
```

```
    # 转化率
    convention = buy/total*100

    # 类别名称对应转化率
    con_dict[group_name] = convention

# 排序
sort_con = sorted(con_dict.items(),key=lambda item:item[1],reverse=True)

# 对排好序的列表取值
sort_con_20 = sort_con[:20]
sort_con_20
```

> sorted的用法： 第一个参数是进行排序的可迭代对象； key：主要是用来比较的元素，lambda函数的参数取自要排序的可迭代对象，指定可迭代对象中的元素进行排序； reverse：排序规则，默认升序，True表示降序；
>
> 前二十个商品种类及对应购买率如下，拿到这个，就可以使用matplotlib进行绘图了

```
[(2951233, 100.0),
 (1597811, 100.0),
 (2550060, 100.0),
 (1833532, 100.0),
 (4470576, 100.0),
 (895939, 100.0),
 (4238377, 100.0),
 (2895013, 100.0),
 (4242742, 100.0),
 (4718844, 100.0),
 (2975713, 100.0),
 (890050, 60.0),
 (835895, 55.555555555555557),
 (512076, 50.0),
 (3025028, 50.0),
 (3445085, 50.0),
 (3695050, 50.0),
 (3797203, 50.0),
 (1879672, 50.0),
 (1601543, 40.0)]
```

```
# 获取所有的商品种类
tick_label = [i[0] for i in sort_con_20]

# 获取转化率
num_list = [i[1] for i in sort_con_20]

# 开始画图
plt.figure(figsize=(20, 8))
plt.bar(range(len(sort_con_20)), num_list,tick_label=tick_label)
plt.show()
```



小结

1.2 全局搜索业务搜索量指标统计

全局搜索业务介绍：

融合本地和在线搜索，为用户提供搜索结果，触达用户的需求；

业务目标：

更多人、更多次数使用全局搜索，在本地和网络查找自己需要的内容和功能。然后可以进行业务的商业变现。

指标体系：

> 无法衡量就无法优化

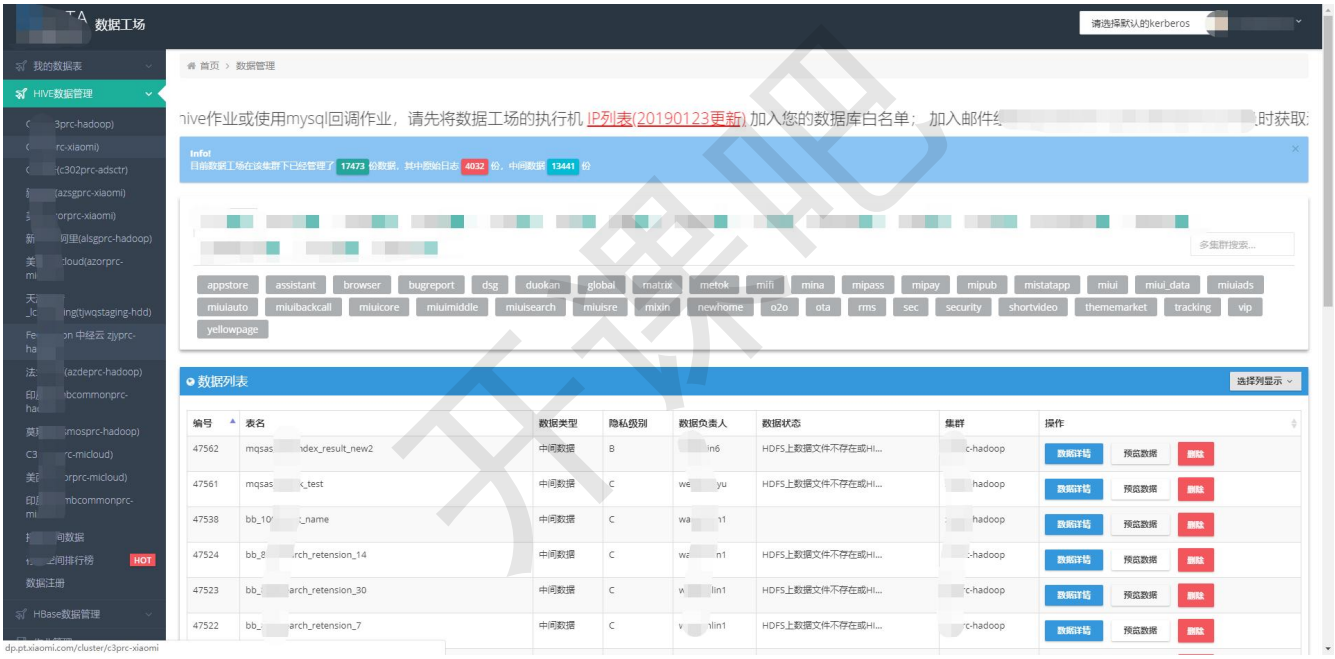
依据业务目标制定一组数据指标，作为量化的业务目标，可以用来进行衡量业务状况、进行业务拆解、制定考核标准等。全局搜索业务核心指标包含：日活跃用户量、活跃次数、总搜索量、搜索用户数等。

数据分析需求：

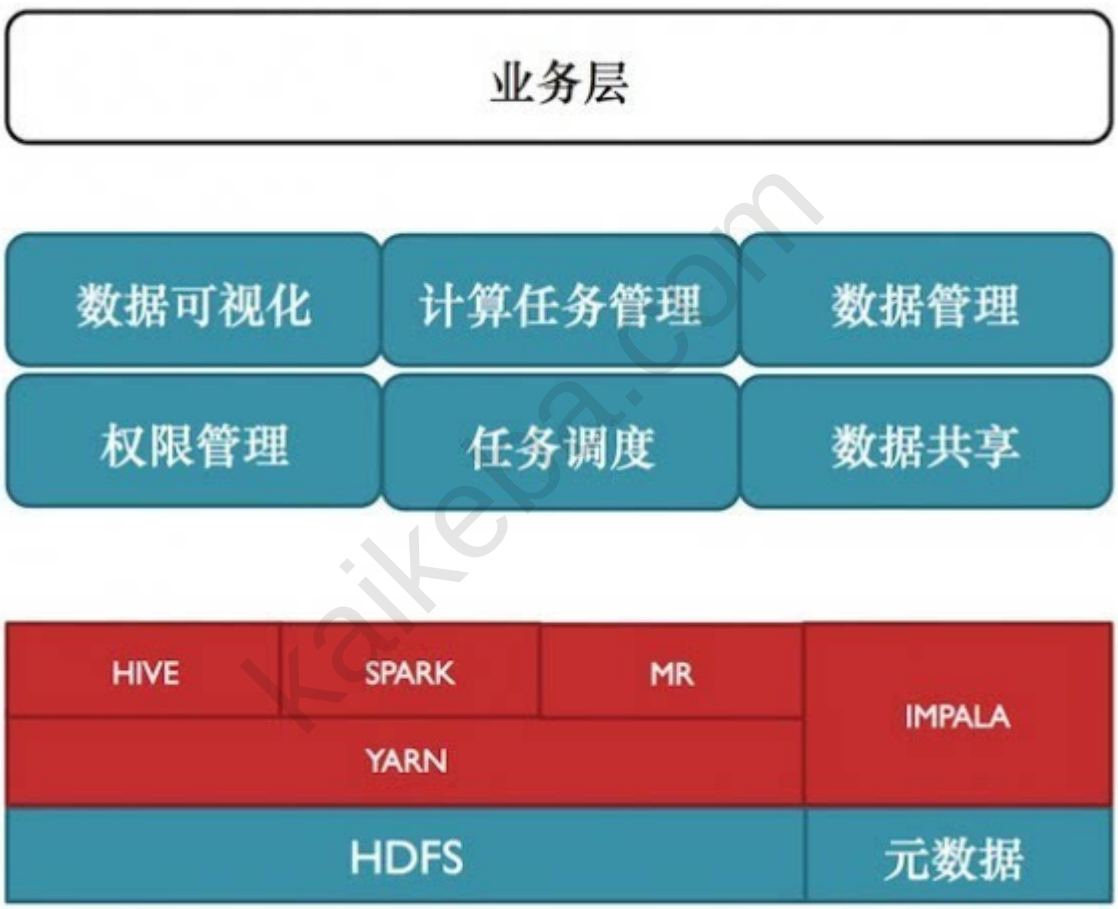
统计全局搜索业务的重要业务指标：总搜索量、总搜索用户数。

业务日志表管理平台：

业务上用数据工场平台，统一管理各业务的数据表；



数据工场架构：



全局搜索日志：

本地搜索和在线搜索分别计入不同的字段内，本地搜索和在线搜索字段内存储各用户整天搜索信息；

imei	用户id
search_events	本地搜索信息
olsearch_events	在线搜索信息
others	其他记录的字段
date	日期分区

用户id	本地搜索信息	在线搜索信息	其他记录的字段	日期分区
imei	**search_events**	**olsearch_events**	**others**	**date**
1	[{"query":"新闻","search_type":"lastsearch","time":"1559549464"},{"query":"小说","search_type":"imsearch","time":"1559553045"}]	null	**	2019/6/3
2	null	[{"query":"视频","search_type":"lastsearch","time":"1559549464"},{"query":"电影","search_type":"imsearch","time":"1559553045"},{"query":"音乐","search_type":"imsearch","time":"1559564233"}]	**	2019/6/3
3	[{"query":"字典","search_type":"lastsearch","time":"1559564233"}]	[{"query":"电视剧","search_type":"imsearch","time":"1559553045"},{"query":"翻译","search_type":"imsearch","time":"1559564233"}]	**	2019/6/3
4	[{"query":"地图","search_type":"lastsearch","time":"1559553045"}]	null	**	2019/6/3

搜索总量统计逻辑：

总搜索量=本地搜索量+在线搜索量；

总搜索用户量=发生本地搜索用户+发生在线搜索用户（去掉有重复的用户）；

SQL统计实现：

```
select "搜索" as name,count(1) as count1,count(distinct imei) as count2 from
(select imei
from
browser.global_search
lateral view explode (search_events) seventtable as sevent
where date = ${date - 1}
--and array_contains(sevent.search_type,'lastsearch')

union all

select imei
from
browser.global_search
lateral view explode (olsearch_events) olseventtable as olsevent
where date = ${date - 1}
--and array_contains(olsevent.olsearch_type,'lastsearch')
) as tmp;
```

语法要点：

元素展开lateral view explode详解：

1.单个lateral view

源表（table1）数据{A:string B:array<BIGINT>}

A	B
190	[1030,1031,1032,1033,1190]
191	[1030,1031,1032,1033,1190]

希望得到如下结果：

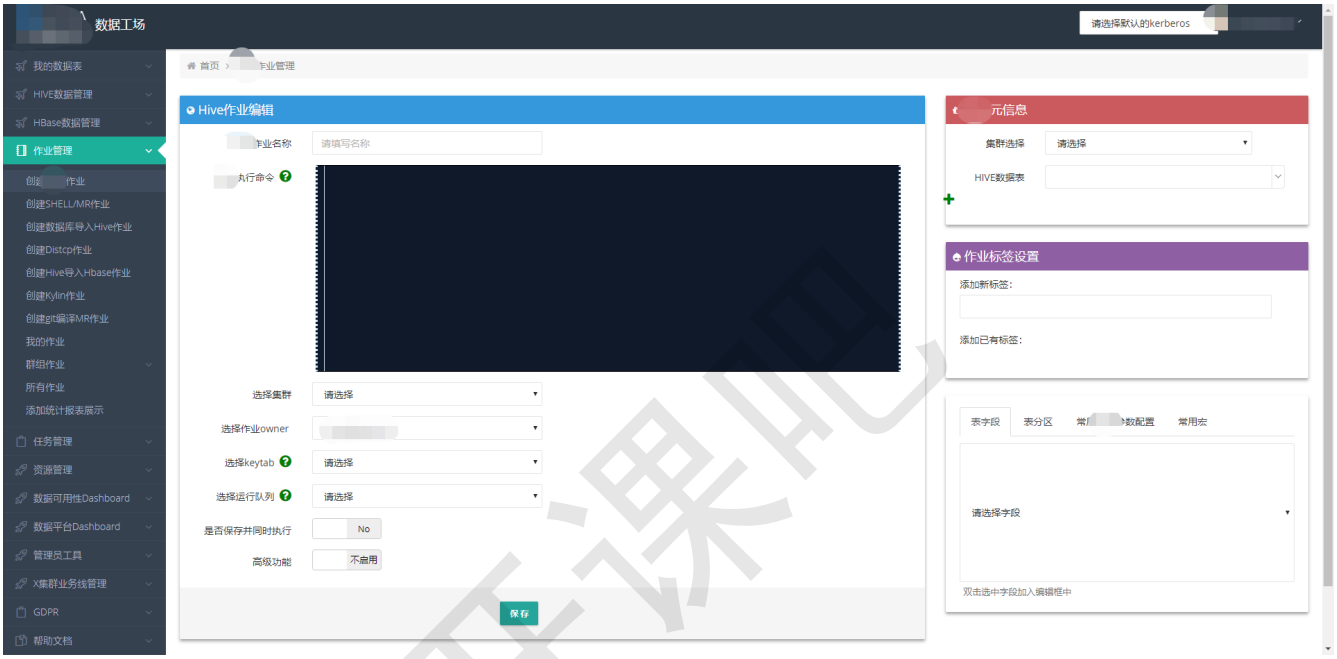
190	1030
190	1031
190	1032
190	1033
190	1190
191	1030
191	1031
191	1032
191	1033
191	1190

union all：将两个表中的 相同的字段拼接到一起，union all不去重，数据会重复；

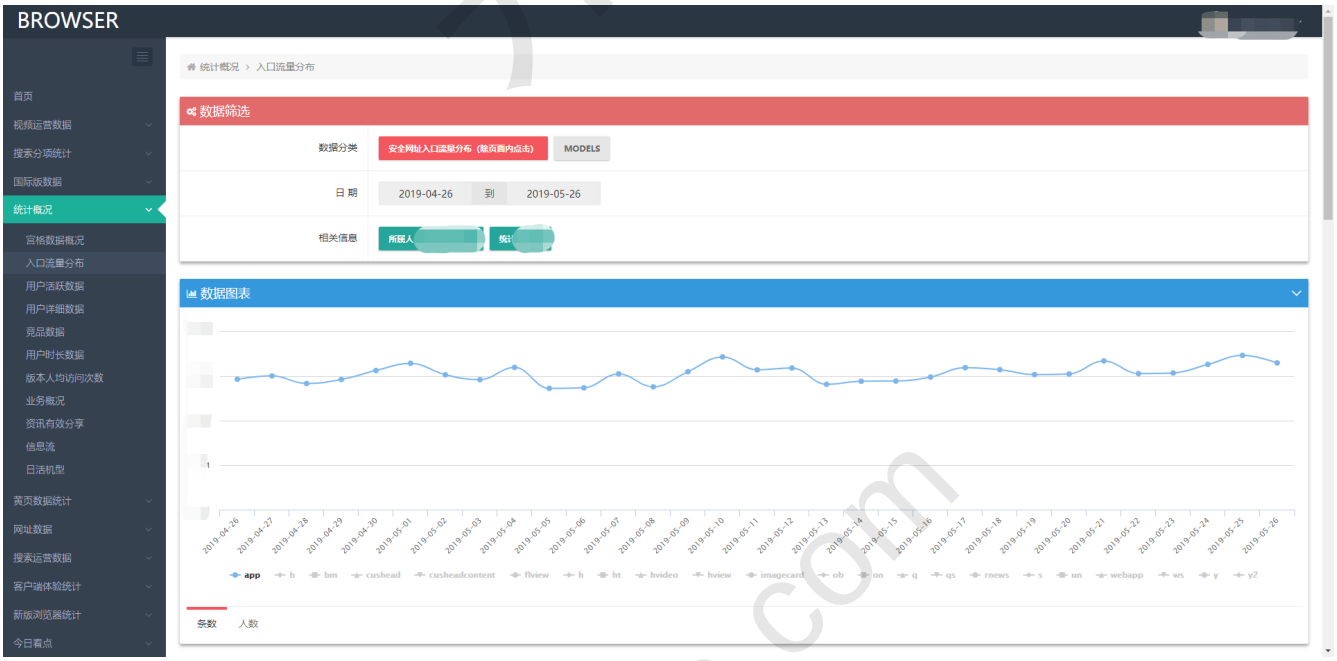
1		1
2		3
3		5
		6
1		
2		
3		
1		
3		
5		
6		

SQL作业提交平台：

设置每天自动执行SQL与数据依赖，可以每天自动生成统计图表



展示平台：



小结

1.3 资讯APP留存预测与生命周期价值分析

资讯APP业务介绍：

应用介绍

热点资讯实时推送 — 人工智能推荐你感兴趣的新闻资讯、每日热点头条新闻第一时间送达； 红包金币不设限制 — 刷新闻、刷视频、发布围观，邀请好友统统都给钱，收徒越多每日奖励越高。 新人直送6元红包 — 新手可领6元红包，邀请好友限时奖金翻倍哟最高可达千元收徒红包； 独家本地资讯围观 — 个人便捷发布专属视频，图片，文字让老乡一起围观，成为新媒体达人； 全网最全热门视频 — 影视搞笑娱乐游戏生活...唯一覆盖全网千万精彩短视频的新闻媒体平台；

业务目标：

不考虑需要盈利情况下，增加活跃用户量。

1. 计算不同渠道投入产出比，找到更好的渠道进行投放。
2. 提升整体用户留存率。

数据分析需求：

计算留存率，预测长期留存率。

计算用户生命周期。计算用户生命周期价值。

留存率计算：

使用用户的画像数据和日志数据，计算各天留存率，由于留存率的波动，求解多日留存率的平均值；

```
select channel,count(1) uv,count(b.deviceid) uv2,count(b2.deviceid) uv7,count(b3.deviceid)
uv14,count(b4.deviceid) uv30,
round(100*count(b.deviceid)/count(1),2) rate2,
round(100*count(b2.deviceid)/count(1),2) rate7,round(100*count(b3.deviceid)/count(1),2)
rate14,round(100*count(b4.deviceid)/count(1),2) rate30 from
(select a.deviceid deviceid,channel from
(select distinct deviceid from browser.xiangkan_user_profile_o2o_daily_dm where date=${date-
30} and first_day=${date-30}  and last_version is not null and last_version !='' and
last_version !=' ') a
join
(select distinct deviceid,channel from feeds.o2o_mifeeds_log_info where date=${date-30} and
event.category='APP' and event.action='激活'  and event.name='首次激活'
--and channel='xiaomi_classicnews'
) b
```

```
on a.deviceid=b.deviceid) a

left join
(select distinct deviceid from feeds.o2o_mifeeds_log_info where date=${date-29} and
((event.category='启动' and event.action='APP'  and event.name='应用启动') or
(event.category='Push' and event.action='点击'  and event.name='调起日活')
or (event.category='启动' and event.action='Deeplink'  and event.name='调起APP'))) b
on a.deviceid=b.deviceid

left join
(select distinct deviceid from feeds.o2o_mifeeds_log_info where date=${date-23} and
((event.category='启动' and event.action='APP'  and event.name='应用启动') or
(event.category='Push' and event.action='点击'  and event.name='调起日活')
or (event.category='启动' and event.action='Deeplink'  and event.name='调起APP'))) b2
on a.deviceid=b2.deviceid

left join
(select distinct deviceid from feeds.o2o_mifeeds_log_info where date=${date-16} and
((event.category='启动' and event.action='APP'  and event.name='应用启动') or
(event.category='Push' and event.action='点击'  and event.name='调起日活')
or (event.category='启动' and event.action='Deeplink'  and event.name='调起APP'))) b3
on a.deviceid=b3.deviceid

left join
(select distinct deviceid from feeds.o2o_mifeeds_log_info where date=${date} and
((event.category='启动' and event.action='APP'  and event.name='应用启动') or
(event.category='Push' and event.action='点击'  and event.name='调起日活')
or (event.category='启动' and event.action='Deeplink'  and event.name='调起APP'))) b4
on a.deviceid=b4.deviceid
group by channel;
```

SQL语法要点：

join可以关联不同的表格中的字段

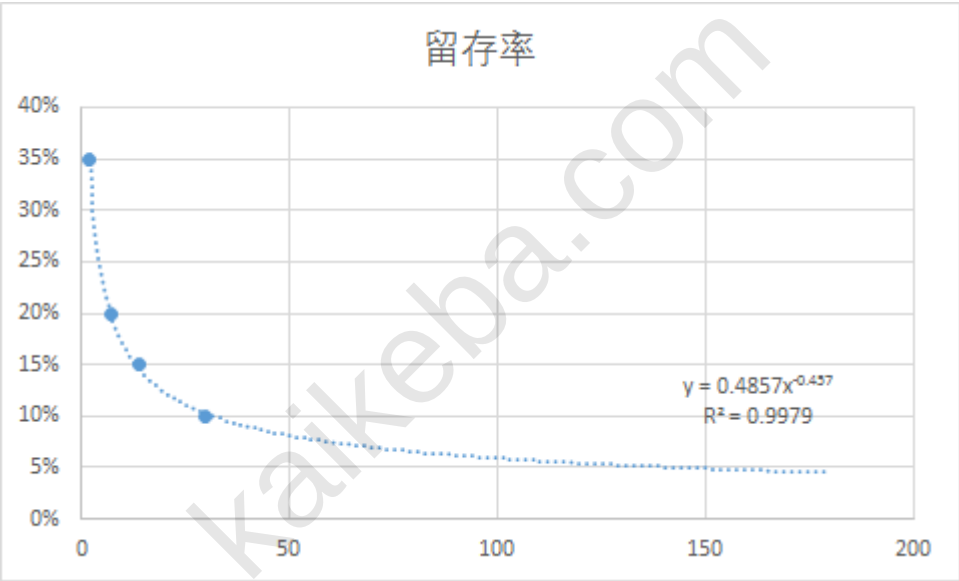
1		2
2		3
3		4
join		
2	2	
3	3	
left join		
1	null	
2	2	
3	3	
right join		
2	2	
3	3	
null	4	

180天留存趋势线：

月内留存情况（数据做了加工处理）：

留存周期	留存率
2	35%
7	20%
14	15%
30	10%

通过Excel的趋势线功能，做出180天留存趋势分布图：



用户生命周期与价值：

通过Excel的公式填充，计算每天的留存情况：

留存周期（天）	留存率	公式示例
1	100%	
2	35.4%	0.4857*A10^(-0.457)
3	29.4%	0.4857*A11^(-0.457)
4	25.8%	0.4857*A12^(-0.457)
5	23.3%	
6	21.4%	
7	20.0%	
8	18.8%	
9	17.8%	
10	17.0%	
170	4.6%	
171	4.6%	
172	4.6%	
173	4.6%	
174	4.6%	
175	4.6%	
176	4.6%	
177	4.6%	
178	4.5%	
179	4.5%	
180	4.5%	
	14.9	SUM(B9:B188)

用户生命周期：计算180天留存总和

用户生命周期价值：用户生命周期*日活跃ARPU

ARPU：*ARPU*(*ARPU*-AverageRevenuePerUser)即通过每用户平均可以获得的收入。

渠道投放策略决策：

渠道：app分发渠道大概可以看做app下载平台，

1. 线下推广：手机厂商，运营商预装app、相关的制作个人刷机ROM团队在整合自己的ROM时的预装以及二维码等线下宣传时推广；
2. 传统媒体：电视等广告媒体宣传；
3. 线上方面：应用的内置广告，弹窗。网页弹窗以及最为重要的应用平台的推广推荐；

决策依据：用户生命周期价值>渠道投放成本

后续可以进行渠道优化等分析工作；

小结

1.4 音乐新用户留存原因分析

音乐业务介绍：

系统级音乐APP，提供音乐及相关内容和功能；

业务目标：

日活跃量

提升新用户的留存率

留存原因分析思路：

通过音乐APP的日志表，把新用户在音乐APP中的各种行为统计出来。根据用户是否有各种行为，及用户是否留存下来，建立决策树模型，找到影响用户会否留存下来的关键因素；

数据准备：

通过SQL查询行为数据，关联用户是否留存确定用户类别；

In [2]:
```
import pandas as pd
file = pd.read_csv('D:/kkb/归档/md课件/data.txt', sep='\t')
file.head()
```

Out[2]:

iei	c.music	c.auto_next	c.force_next	c.play_all_list	c.song_click	c.song_like	c.list_like	c.download	c.share	c.comment	c.recommend	c.radio	c.video	c.se
if4	0	0	0	0	0	0	0	0	0	0	0	0	0	
if8	0	0	0	0	0	0	0	0	0	0	0	0	0	
8b	1	0	7	0	0	0	0	0	0	0	0	0	0	
20	0	0	0	0	0	0	0	0	0	0	0	0	0	
e6	0	0	0	0	0	0	0	0	0	0	0	0	0	

In [3]:
```
file.columns.values.tolist()
```

Out[3]:
```
['c.imei',
 'c.music',
 'c.auto_next',
 'c.force_next',
 'c.play_all_list',
 'c.song_click',
 'c.song_like',
 'c.list_like',
 'c.download',
 'c.share',
 'c.comment',
 'c.recommend',
 'c.radio',
 'c.video',
 'c.search',
 'c.search_click',
 'c.model',
 'c.acimei']
```

python建模：

```
# -*- coding:utf-8 -*-
import pandas as pd
from sklearn import tree, metrics
import graphviz
```

```
# 数据读入
file = pd.read_csv('/Users/a/Documents/data.txt', sep='\t')
# file = pd.read_csv('D:/data.txt', sep='\t')


# 生成active 0-1变量
file['active'] = [(1 if file['c.imei'][i]==file['c.acimei'][i] else 0) for i in
range(len(file))]
# 生成auto_next 0-1变量
file['auto_next'] = [(1 if file['c.auto_next'][i]>0 else 0) for i in range(len(file))]
# 生成force_next 0-1变量
file['force_next'] = [(1 if file['c.force_next'][i]>0 else 0) for i in range(len(file))]
# 生成play_all_list 0-1变量
file['play_all_list'] = [(1 if file['c.play_all_list'][i]>0 else 0) for i in
range(len(file))]
# 生成song_click 0-1变量
file['song_click'] = [(1 if file['c.song_click'][i]>0 else 0) for i in range(len(file))]

# 数据集切片
dat1 = file[['c.music', 'c.radio', 'c.video', 'c.search', 'c.search_click']]

dat2 = file[['c.auto_next', 'c.force_next', 'play_all_list', 'c.song_click',
'c.song_like','c.list_like', 'c.download', 'c.share','c.comment','c.recommend','c.radio',
'c.video', 'c.search', 'c.search_click']]

dat3 =
file[['c.auto_next','c.force_next','c.play_all_list','c.song_click','c.song_like','c.list_li
ke', 'c.download', 'c.share', 'c.comment', 'c.recommend']]

dat4 = file[['auto_next', 'force_next', 'play_all_list', 'song_click', 'c.song_like',
             'c.list_like', 'c.download', 'c.share', 'c.comment', 'c.recommend']]

dat5 = file[['auto_next', 'force_next', 'play_all_list', 'song_click', 'c.song_like',
             'c.list_like', 'c.download', 'c.share', 'c.comment', 'c.recommend',
             'c.radio', 'c.video', 'c.search', 'c.search_click']]

y = file['active']

# 决策树模型
clf1 = tree.DecisionTreeClassifier(criterion='gini')
clf2 = tree.DecisionTreeClassifier(criterion='gini', splitter='best', min_samples_leaf=50,
max_depth=4)
clf3 = tree.DecisionTreeClassifier(criterion='entropy')
clf4 = tree.DecisionTreeClassifier(criterion='entropy', min_samples_leaf=50, max_depth=4)
# 建立决策树模型m1
m1 = clf1.fit(dat1, y)
print(m1.feature_importances_)
dot_data = tree.export_graphviz(m1, out_file=None, feature_names=['c.music', 'c.radio',
'c.video', 'c.search'], class_names=['0', '1'], filled=True,special_characters=True,
rounded=True)
graph = graphviz.Source(dot_data)
graph.render("tree")
```

```
# 建立决策树模型m2
m2 = clf1.fit(dat2, y)
print(m2.feature_importances_)
print(clf1.score(dat2, y))  # 0.8449
# m2.predict_proba(dat2)[:, 1]
metrics.roc_auc_score(y, m2.predict_proba(dat2)[:, 1])  # 0.715588
dot_data = tree.export_graphviz(m2, out_file=None, feature_names=['c.auto_next',
'c.force_next', 'c.play_all_list', 'c.song_click', 'c.song_like', 'c.list_like',
'c.download', 'c.share', 'c.comment', 'c.recommend', 'c.radio', 'c.video', 'c.search',
'c.search_click'],
                                class_names=['0', '1'], filled=True,
special_characters=True, rounded=True)
graph = graphviz.Source(dot_data)
graph.render("tree")

# 建立决策树模型m3
m3 = clf3.fit(dat3, y)
print(m3.feature_importances_)
dot_data = tree.export_graphviz(m3, out_file=None, feature_names=['c.auto_next',
'c.force_next', 'c.play_all_list', 'c.song_click', 'c.song_like', 'c.list_like',
'c.download', 'c.share', 'c.comment', 'c.recommend'],class_names=['0', '1'], filled=True,
special_characters=True, rounded=True)
graph = graphviz.Source(dot_data)
graph.render("tree")

# 建立决策树模型m4
m4 = clf3.fit(dat4, y)
print(m4.feature_importances_)
dot_data = tree.export_graphviz(m4, out_file=None, feature_names=['auto_next', 'force_next',
'play_all_list', 'song_click', 'c.song_like', 'c.list_like', 'c.download', 'c.share',
'c.comment', 'c.recommend'],class_names=['0', '1'], filled=True, special_characters=True,
rounded=True)
graph = graphviz.Source(dot_data)
graph.render("tree")

# 建立决策树模型m5
m5 = clf3.fit(dat5, y)
print(m5.feature_importances_)
dot_data = tree.export_graphviz(m5, out_file=None, feature_names=['auto_next', 'force_next',
'play_all_list', 'song_click', 'c.song_like', 'c.list_like', 'c.download', 'c.share',
'c.comment', 'c.recommend', 'c.radio', 'c.video', 'c.search', 'c.search_click'],class_names=
['0', '1'], filled=True, special_characters=True, rounded=True)
graph = graphviz.Source(dot_data)
graph.render("tree")
```

交付结果：

关键因素，可能的策略

小结

2. 数据分析职场

2.1 数据分析职场状况

优秀简历案例：

工作经历

2017.06—2018.11　　斗鱼直播（武汉）　　数据平台部　　数据挖掘工程师

- **个性化推荐**
  1) **推荐策略**：抽取用户的观看房间序列数据，基于 Word Embedding 计算房间之间的相似度，配合用户的观看偏好房间可用来作个性化推荐，配合 KNN 算法可用来修正房间的信息；
  2) **评价指标**：建立个性化推荐的评价体系（包括精确性和多样性指标等），并对关键指标（如转化率）的影响因素定时监控；
  3) **效果优化**：针对新用户推荐，调整策略中各召回集的优先级和比例等参数形成 AB test，比较不同推荐策略的效果优劣，并根据各召回集的实际转化来进一步优化。
- **用户画像**
  1) **流失标签**：根据用户的固有属性和行为属性，运用分类模型（如逻辑回归/ 随机森林/ GBDT 等）构建用户的流失标签体系，配合其他的用户标签对即将流失用户采取一定的挽留措施。
- **产品运营**
  1) **产品功能优化**：查看产品各功能使用率，撤掉流量低的功能以达到精简产品的目的；优化搜索流程；
  2) **产品性能优化**：实时监控视频站性能，若超过阈值实时报警，并将数据沉淀在 APM 产品上供查询；
  3) **用户运营**：从观看、付费、渠道来源等各维度划分用户，针对特定的用户群使用不同策略来运营；
  4) **主播运营**：做好可挖掘主播的调研分析，并在主播的生命周期各阶段评估主播的健康程度。

2016.12—2017.05　　百度　　在线管理部　　数据分析师

1) 内部反馈处理：采用贝叶斯算法对用户问题进行分类，问题类别的划分能提高客服人员的处理效率；
2) 外部舆情挖掘：跟踪产品线的舆情导向，对同一新闻分渠道统计，总体评估该新闻的传播热度，判断舆情等级并积极与产品线保持沟通，辅助以词云图展示外部用户的真实声音；
3) 自主产品调研：发起某产品（手机百度 app）的问卷调研，找出核心用户群体，并针对其开展深度访谈，确定产品的问题，同时与竞品作对比，对产品的进一步优化提出建议。

2014.12—2016.09　　京东商城　　通讯业务部　　数据分析师

1) 用户购买：运用漏斗模型对用户的行为路径进行分析，优化购买流程，对潜在的用户作相关性推荐；
2) 用户分析：构建京东通信用户的通信画像+网上购物画像，对其通信行为、购物行为进行分析；
3) 用户挽留：挖掘产品策略变化后，影响客户流失的重要因素（决策树算法），并根据用户的网上购物画像，采用聚类方法有针对性的选择几种中高频商品，跨部门合作，维系老用户、吸引新用户。

专业技能

计算机技能：熟练使用 python 进行数据处理、网络爬虫和算法建模，用 R 实现图表可视化

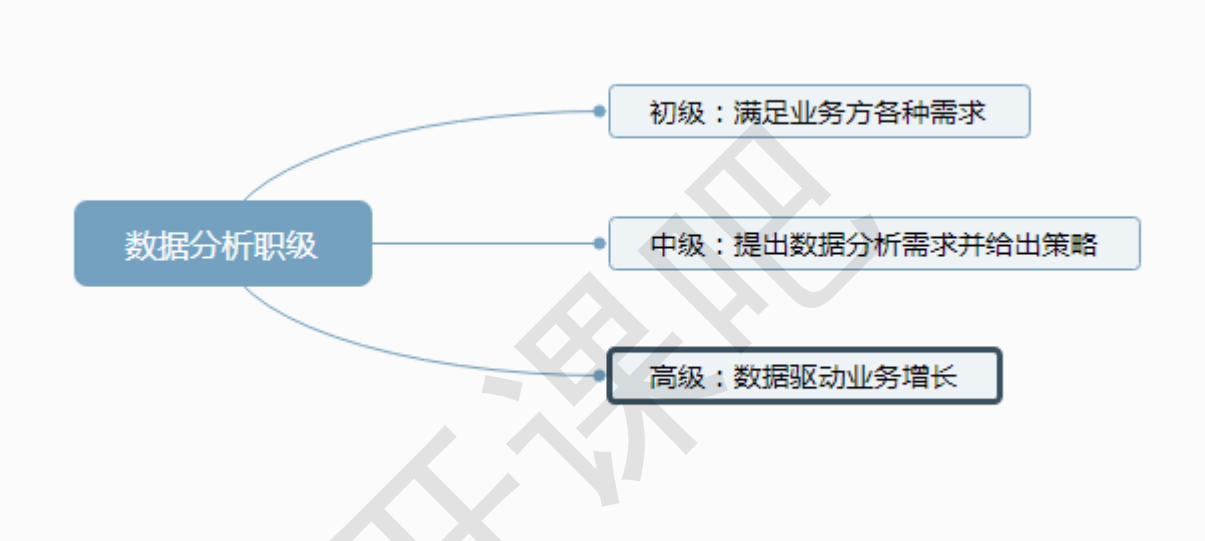
精通 excel（包括 VBA 编程），灵活运用 SQL，熟练使用 JAVA 编程

数据分析师职业路径：

懂业务，从业务转数据分析；

懂数据分析技能，学习业务；

职级体系：



层级1		助理数据分析师	数据分析师	资深数据分析师
概述		使命必达 （快速学习，执行配合）	无懈可击 （独立执行，小有成就）	无中生有 （发掘项目，主动执行）
业务分析能力	问题识别	在一定指导下准确识别问题	能够独立清晰地识别问题	发现问题并转化为分析目标
	分析规划	在一定指导下完成分析规划	明确范围并做好分析规划	提炼问题并做好分析规划
	数据获取	提取简单数据、了解分析方法	熟练提取数据、应用分析方法	熟练提取数据、指导员工分析
	展现演示	能够形成报告、传递分析结果	展现的条理、逻辑、表达清晰	结论突出清晰、指导员工
	价值应用	完成业务需求、提出建议	提出建议、推动建议被采纳	提出有效建议、独立主动
执行和管理能力		合理分配和安排、完成分析任务	作为核心成员、控制进度和质量	领导跨部门项目、寻求资源
胜任和影响力			推动建议被采纳、跨团队沟通协调	具备一定影响力、判断力
分析方法要求		1、数据预处理、检验、清洗 2、各种常用统计检验方法 3、描述性统计分析 4、对比分析 5、简单的多元统计分析方法 6、数据库知识	除助理数据分析师要求掌握的分析方法之外，还需要正确熟练地掌握和运用以下方法： 结构分析、趋势分析、关联分析等	比数据分析师增加以下分析方法： 数据挖掘模型，比如：回归、聚类、因子、神经网络、时间序列、关联规则、决策树等等

层级2		数据分析专家	高级数据分析专家	资深数据分析专家
概述		独挡一面 （推进业务，辅导团队）	红杏出墙 （统筹规划，名声在外）	诸葛连弩 （参与决策，指挥有度）
业务分析能力	问题识别	识别问题并推动解决问题	思考数据的价值并规划推进	提出对业务发展前瞻性建议
	分析规划	提炼问题并做好分析规划	提炼问题并做好分析规划	提炼问题并做好分析规划
	数据获取	熟练提取数据、指导员工分析	熟练提取数据、指导员工分析	熟练提取数据、指导员工分析
	展现演示	结论突出清晰、指导员工	结论突出清晰、指导员工	结论突出清晰、指导员工
	价值应用	提出有效建议、独立主动	提出有效建议、独立主动	提出有效建议、独立主动
执行和管理能力		完成影响大的复杂项目	独立主动完成影响大的复杂项目	独立主动完成影响大的复杂项目
胜任和影响力		具备较强影响力、判断力	分享和指导、在公司层面有影响力	在专业领域有一定的影响力
分析方法要求		与资深数据分析师相同	与资深数据分析师相同	与资深数据分析师相同

级别	P4	P5	P6	P7
总宗旨	要求：做事让人放心，能在指导下完成分配的任务 工作独立性：明确What/How，偶尔指导	两年以上工作经验 要求：个人能独当一面，能独立完成分配的任务 工作独立性：明确What 自行How	要求：独立负责一条业务线，组织小团队独当一面 工作独立性：自行What 自行How	要求：组织中型团队独挡一面 工作独立性：自行What 自行How
数据提取及处理能力	1.SQL能力：能写基本SQL（join、group by、order by、distinct、sum、count、主键），在数据提取平台上高效准确提取分析所需数据 2.Excel能力：会使用EXCEL常用功能（透视表）、基本函数（vlookup）	1.SQL能力：能写基本SQL（join、内外连接区别、group by、order by、distinct、sum、count、主键、索引、分区、统计函数、中位数、分位数、窗口函数），在数据提取平台上高效准确提取分析所需数据 2.Excel能力：会使用EXCEL常用功能（透视表）、基本函数（vlookup）、动态图表 3.统计基本知识：了解统计、方差、标准差、正态分布	1.SQL能力：能写基本SQL（join、内外连接区别、group by、order by、distinct、sum、count、主键、索引、分区、统计函数、中位数、分位数、窗口函数），在数据提取平台上高效准确提取分析所需数据 2.Excel能力：会使用EXCEL常用功能（透视表）和基本函数（vlookup），动态图表 3.统计基本知识：了解统计、方差、标准差、正态分布	1.SQL能力：能写基本SQL（join、内外连接区别、group by、order by、distinct、sum、count、主键、索引、分区、统计函数、中位数、分位数、窗口函数），在数据提取平台上高效准确提取分析所需数据 2.Excel能力：会使用EXCEL常用功能（透视表）和基本函数（vlookup），动态图表 3.统计基本知识：了解统计、方差、标准差、正态分布
数据体系建设能力	1.业务逻辑：了解负责业务线的基本业务逻辑，并在指导下与业务部门提炼指标框架与核心指标 2.对数据生产流程的理解：了解所负责业务线的数据规范和数据生产流程，对于已有规范可以正确的维护新需求，对于未包含的规范可以明确提出规范需求 3.产品分析能力：能与业务沟通确认	1.业务逻辑：了解负责业务线及相关业务线基本业务逻辑，能独立与业务部门提炼指标框架与核心指标 2.对数据生产流程的理解：了解所负责业务线的数据规范和数据生产流程，对于已有规范可以正确的维护新需求，对于未包含的规范可以明确提出规范建议 3.产品分析能力：能与业务沟通确认需求，并独立转化成底	1.业务逻辑：了解负责业务线及相关业务线复杂交叉的业务逻辑，能独立与业务部门提炼指标框架与核心指标 2.对数据生产流程的理解：了解相关业务线的数据规范和数据生产流程，独立完成并能规划基本的数据规范与流程 3.产品分析能力：能与业务沟通确认需求，独立转化成底层数据方案和产品需	1.业务逻辑：精通多业务线复杂交叉的业务逻辑，能独立与业务部门提炼指标框架与核心指标，并提出优化方案 2.对数据生产流程的理解：熟悉多业务线的数据规范和数据生产流程，独立完成并规划数据规范与流程，并推广应用 3.产品分析能力：能与业务沟通确认需求，独立转化成底层数据方案和产品需求，并能规划升级产品功能

2.2 数据分析职场与未来

某硅谷最大互联网公司总数据科学家：数据分析师，究竟是做什么的？

数据分析师在市场上是近些年出现的一个新的职能，比起研发、算法、产品、运营等等这些已经演进二三十年的职能，我们还是在非常年轻的阶段。

> 我们就像一把枪上的准星，没有准星也能开枪，但是准星能使这把枪更加有用。公司没有任何人做数据分析，短期也依然能运行，只是很多地方运行地会不太好；如果有一天公司里做数据分析的人都消失了，公司短时间内也不会垮掉，但是时间长一些肯定会有影响。

who

我们是一群在相关量化领域受过专业的训练，并且希望应用自己的量化能力，在数据中挖掘对业务有用的信息，并且通过这些信息为业务发展提供助力但是同时又保持数据的中立性的人。

what

我总结我们做的事情，可以抽象成三类 (1) 描述现状 (2) 寻找规律 (3) 推动改进。这三类事是逐层推进地，但是都很重要。

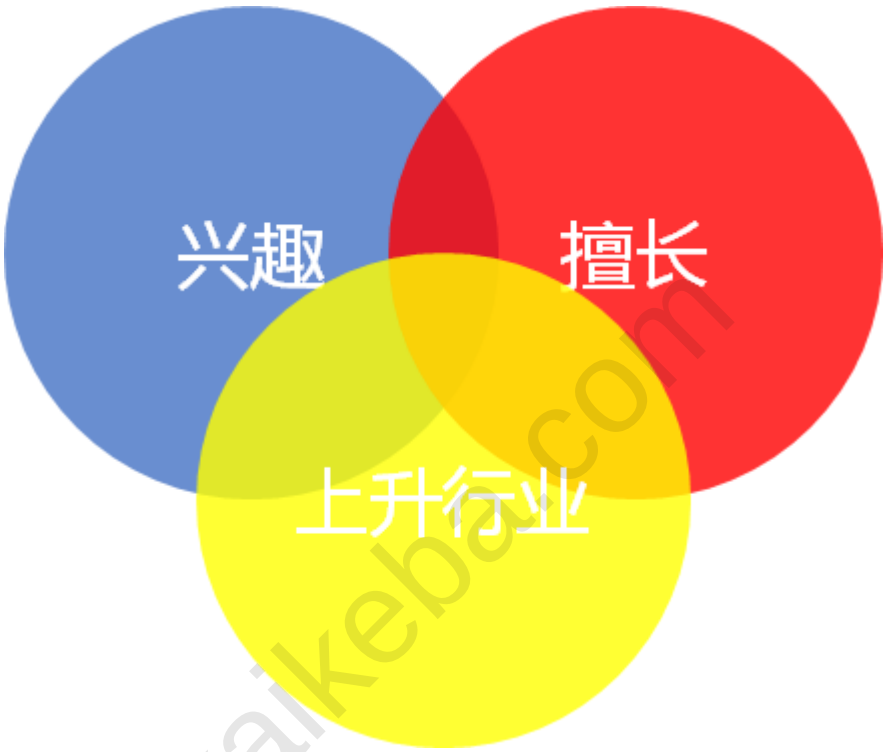
应用在取数上就是 “取什么数”、“为什么取”往往比 “怎么取”、“是多少”更重要。

where

我把这个问题总结成两个方面 “能力建设”和 “文化建设”。在**能力建设**方向，打铁还需自身硬。

跟能力建设同等重要甚至更重要的是**文化建设**。我们改变环境 (同事、公司、行业) 怎么看待数据分析师，首先要坚定我们自己怎么看待自己。这里有自信的问题。

2.3 职场技能与职业规划



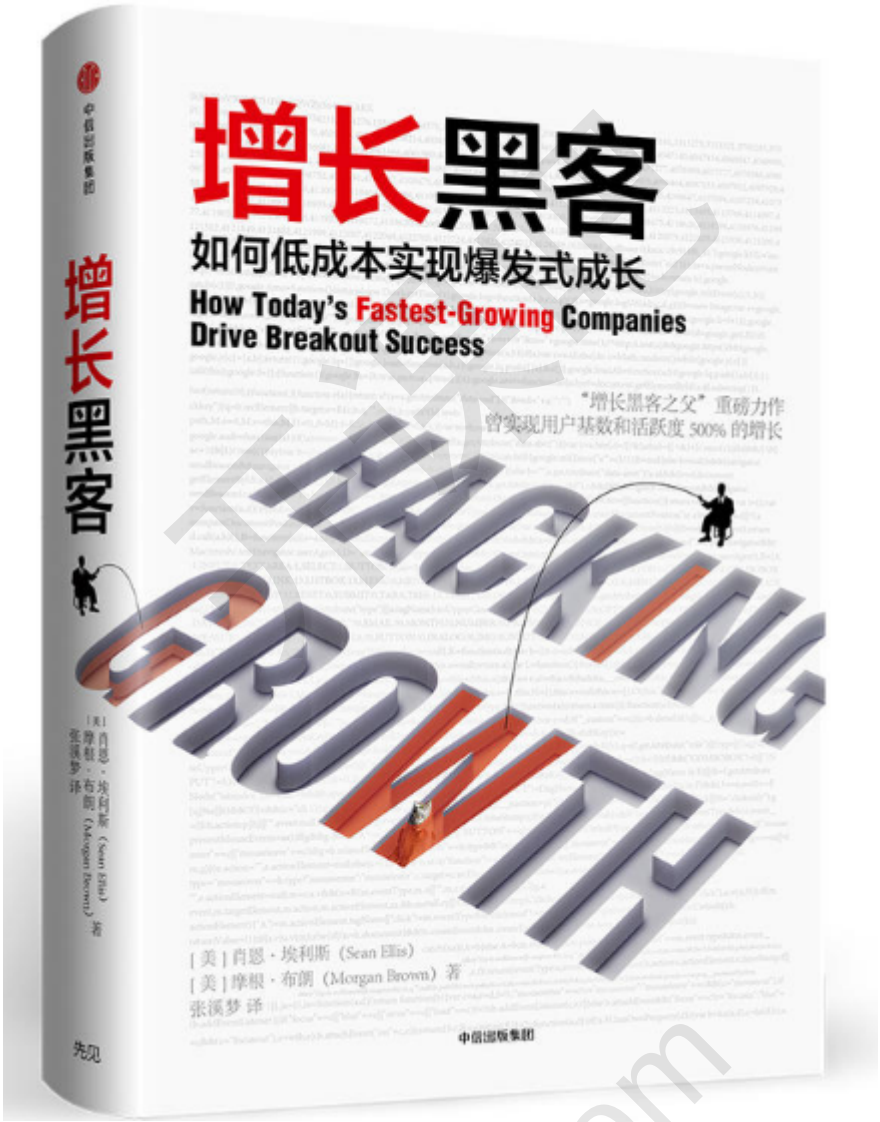
认清自身的优劣势和兴趣爱好，从而帮助自己考虑清楚想要从事的岗位。

性格测量：百度 mbti

五、拓展点、未来计划、行业趋势

书籍推荐：

《增长黑客》



六、总结

数据分析的目的是理解业务，满足数据分析需求，驱动业务增长；

数据分析流程：

接收需求

获取数据

分析数据

汇报结果

后续课程：大数据分析技术，hive等。业务逻辑，数据分析方法论。