## 《大数据技术基础综合实验》

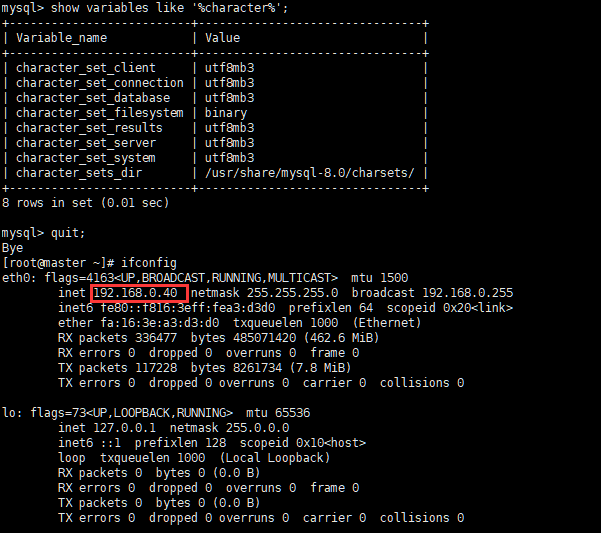
**姓名：李志毅**

**班级：2018211314**

**学号：2018211582**

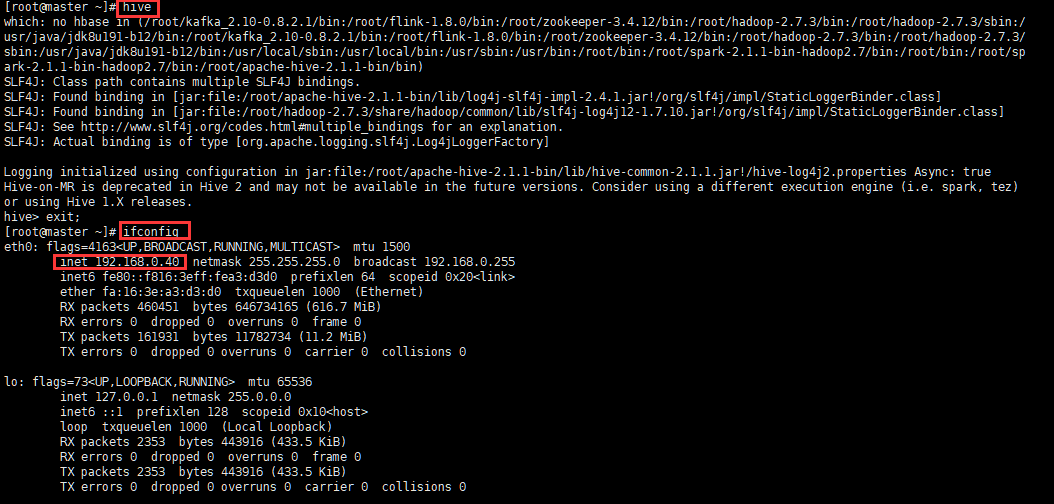
## 一、实验截图

### 实验截图1



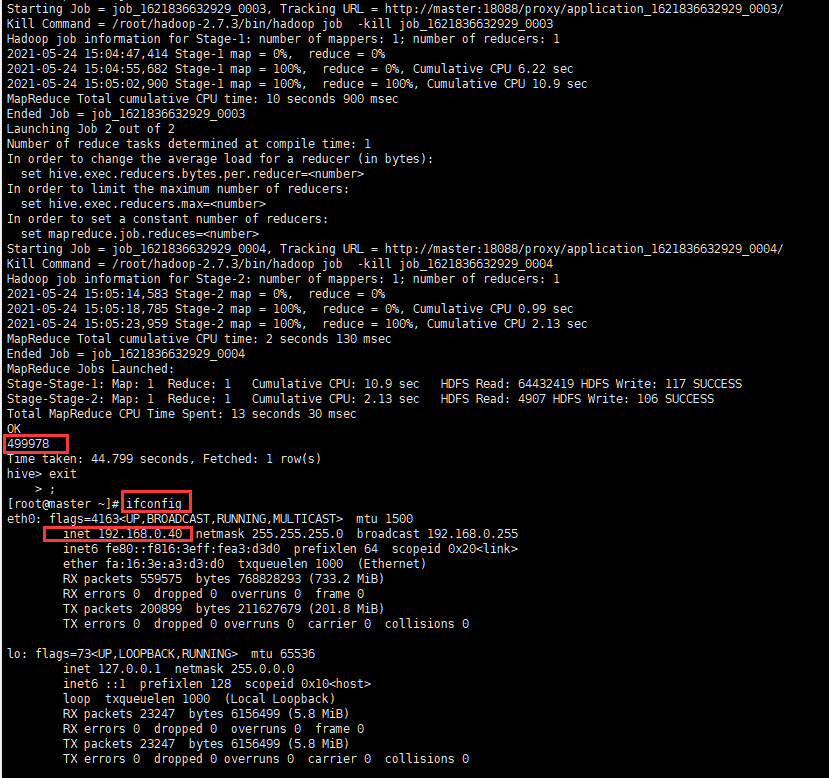
**实验截图 1:查看 mysql 编码**

### 实验截图2



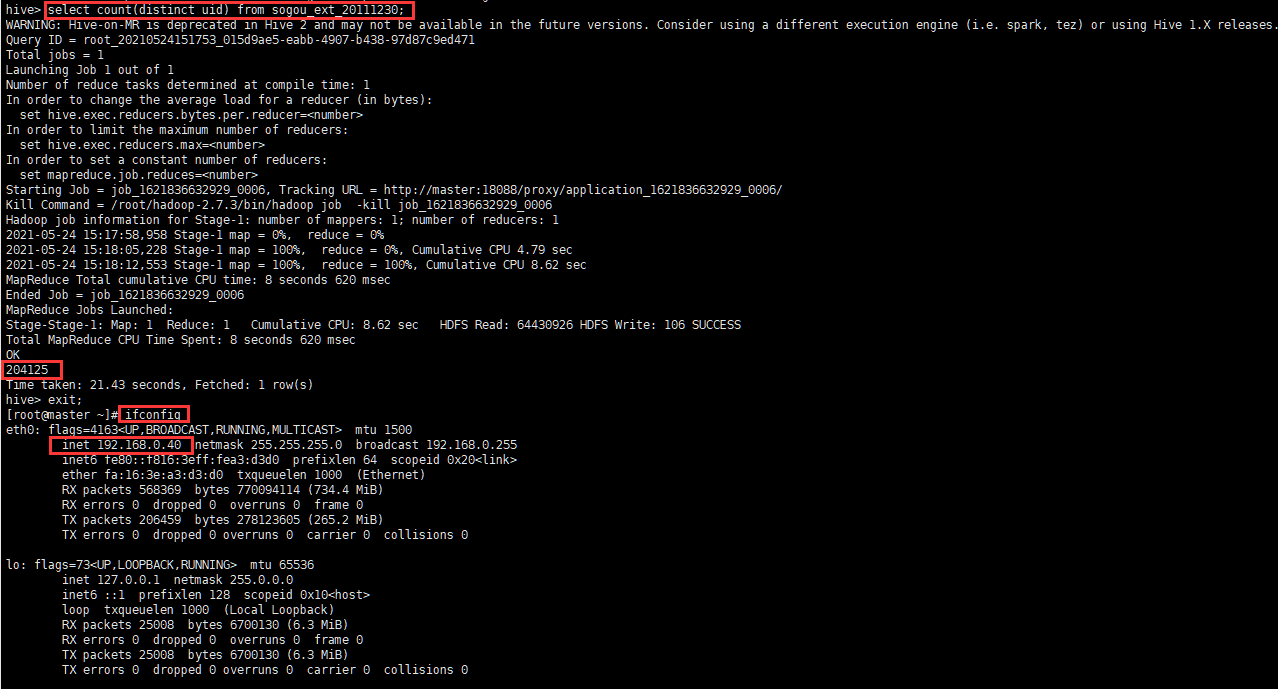
**实验截图 2:hive启动成功**

**实验截图3**



**实验截图 3:无重复总条数**

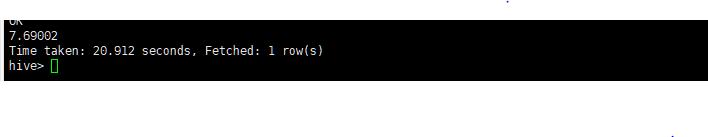
### 实验截图4



**实验截图 4:独立UID条数**

### 实验截图5第一种

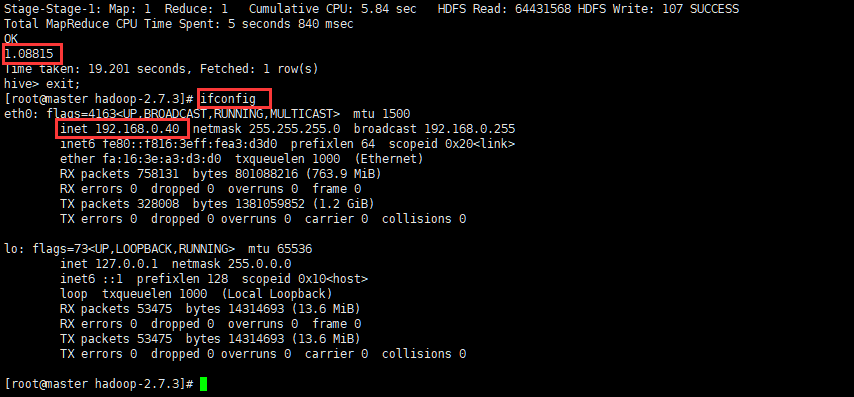
|  |
| --- |
| **select avg(length(keyword)) from sogou\_ext\_20111230;** |



**实验截图 5:关键词平均长度统计**

### 实验截图5第二种

|  |
| --- |
| **select avg(a.cnt) from (select size(split(keyword,'\\s+')) as cnt from sogou\_ext\_20111230) a;** |

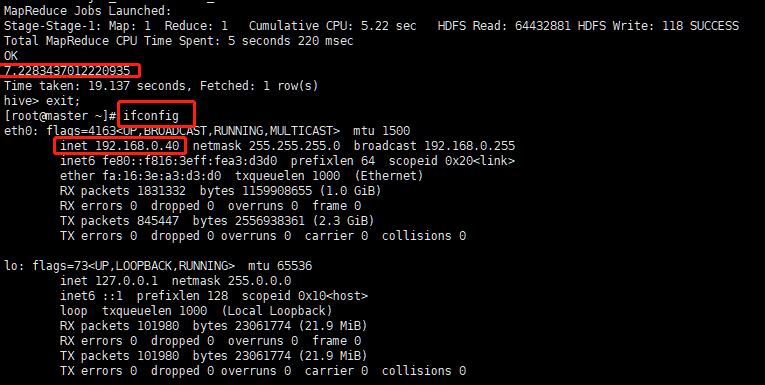


**实验截图 5:关键词平均长度统计**

### 实验截图5第三种

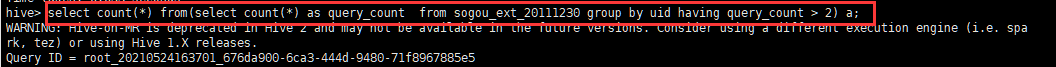
|  |
| --- |
| **select avg(a.len/a.cnt) from (select size(split(keyword,'\\s+')) as cnt,length(split(keyword,'\\s+')) as len from sogou\_ext\_20111230) a;** |

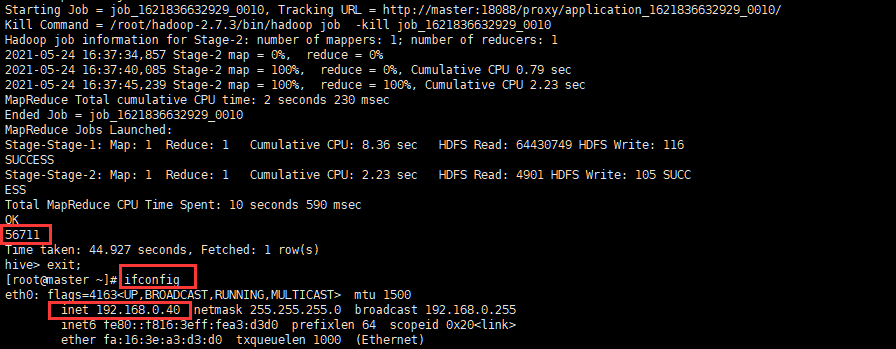




**实验截图 5:关键词平均长度统计**

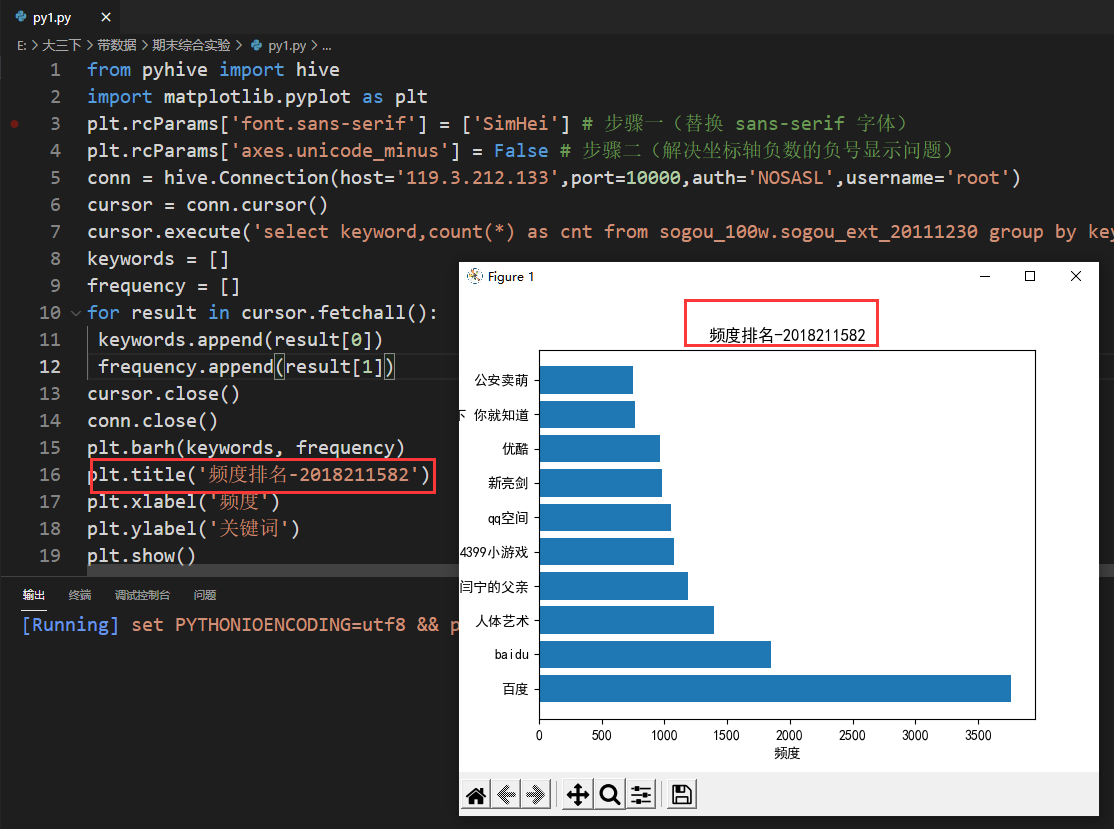
### 实验截图6





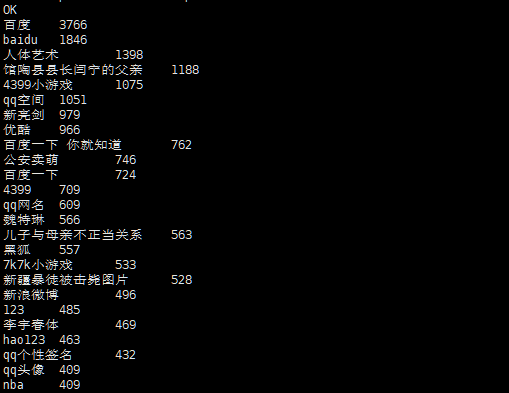
**实验截图 6:** **查询次数大于 2 次的用户总数**

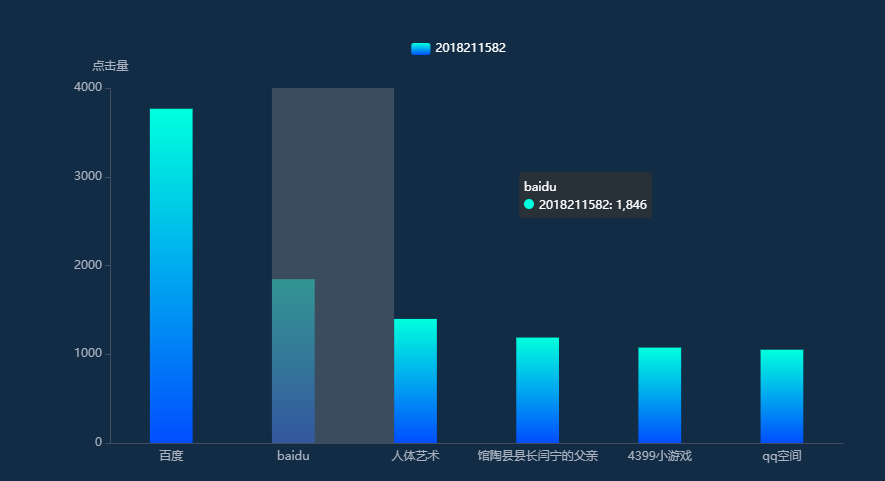
### 实验截图7



**实验截图 7:** **搜索前10的Python可视化**

### 实验截图8





**实验截图 8:** **华为DLV可视化截图**

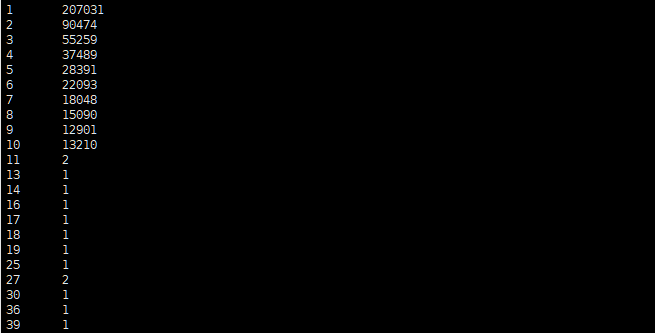
## 二、实验分析

**1.分析 rank 与用户点击次数之间的关系**

* **根据要求，设计HiveSQL语句：**根据rank进行分组，统计每组的总数，即为点击数，根据rank升序排序数据，因此HiveSQL语句如下：

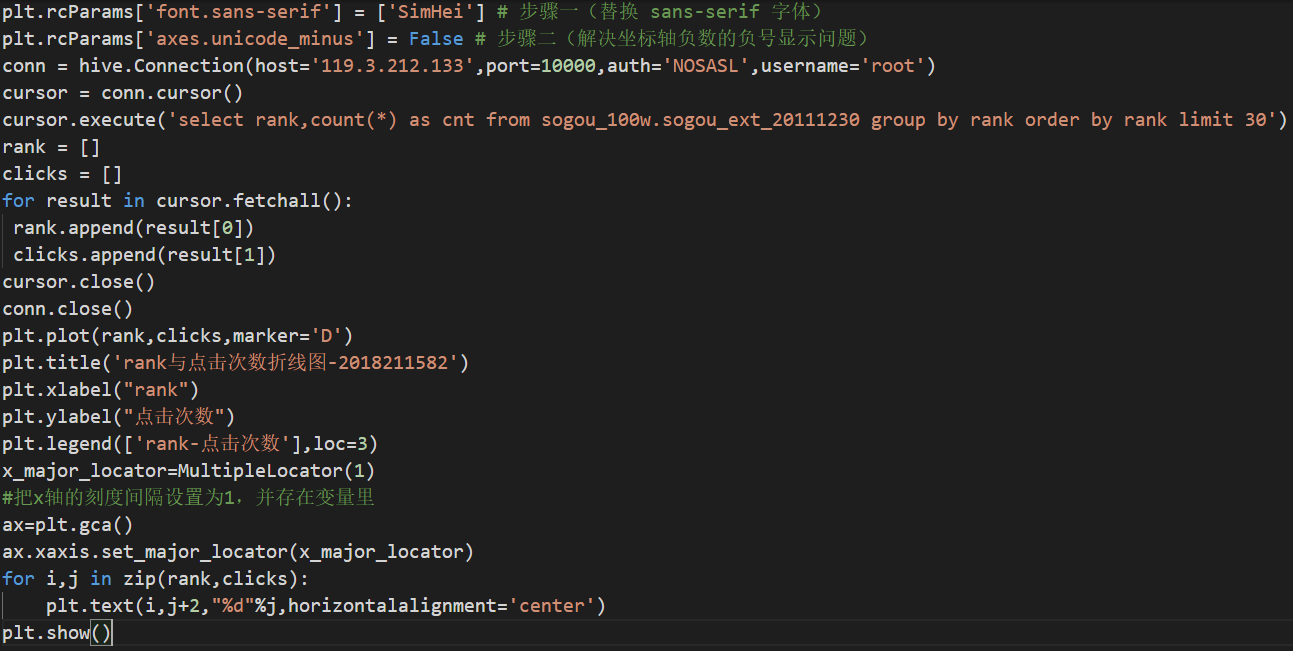
**SELECT** rank,**COUNT**(\*) **AS** cnt **FROM** sogou\_100w.sogou\_ext\_20111230 **GROUP** **BY** rank **ORDER** **BY** rank **LIMIT** 30

查询结果如下：



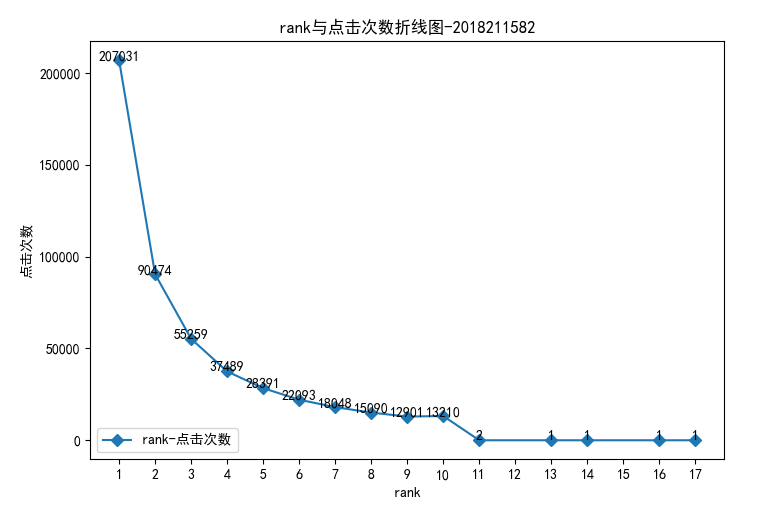
* **编写Python可视化代码：**

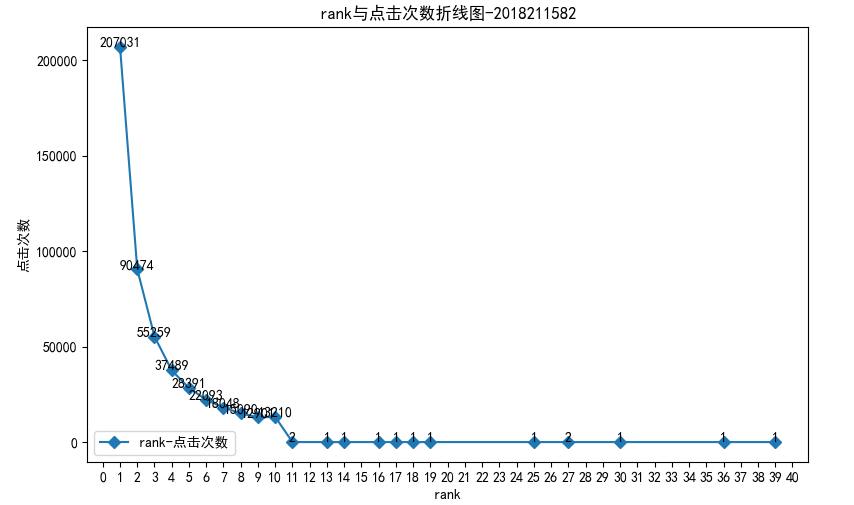
根据结果，可以使用PyHive连接hive并将查询到的数据进行可视化展示，python代码如下：



* **结果分析：**

用户提交一个查询后，搜索引擎可能会返回多页结果，但是这些结果并不一定都是用户所想要的，因此用户一般不会将这些结果逐个都点击浏览。因此，根据搜索数据，我设计了HiveSQL语句，查询到了不同rank对应的点击次数，使用python远程连接Hive得到查询到数据并，并根据这个数据，制作了如下的折线图：





由折线图我们可以清晰的看出，**99.9%**的用户只会翻看搜索引擎返回结果的前**10**个结果，即返回结果页面的第一页，只有极少数的用户会翻看rank10之后的结果，仅有**14**次，翻看rank数最大为**39**，仅有**1**次。这个用户行为决定了尽管搜索引擎返回的结果数目十分庞大，但真正可能被绝大部门用户所浏览的，只有排在最前面的很小一部分而已，这也是一个web系统或网页需要做好SEO优化的原因，排名更靠前的返回结果更加受到使用者的青睐，并且这一概率很大。

**2.查询次数最多的前20个的用户ID和次数**

* **设计挖掘目标**

用户的查询次数一定程度上代表了用户对于一个搜索引擎产品的依赖程

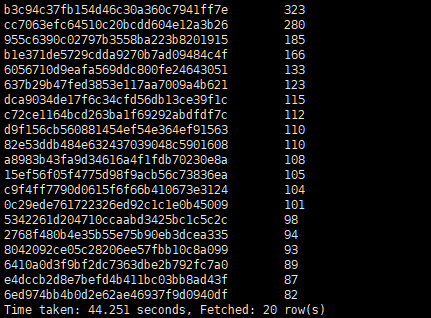
度和使用程度，同时也能够从侧面反映出搜索引擎的搜索算法的准确度，通过对于用户查询次数的分析，有利于优化搜索引擎，优化用户体验。因此，从庞大的数据中提取出查询次数最多的用户相关信息和次数是有必要做的一项。

* **完成HiveSQL代码**

**SELECT** uid,**COUNT**(\*) **AS** cnt **FROM** sogou\_ext\_20111230 **GROUP** **BY** uid **ORDER** **BY** cnt **DESC** **LIMIT** 20;

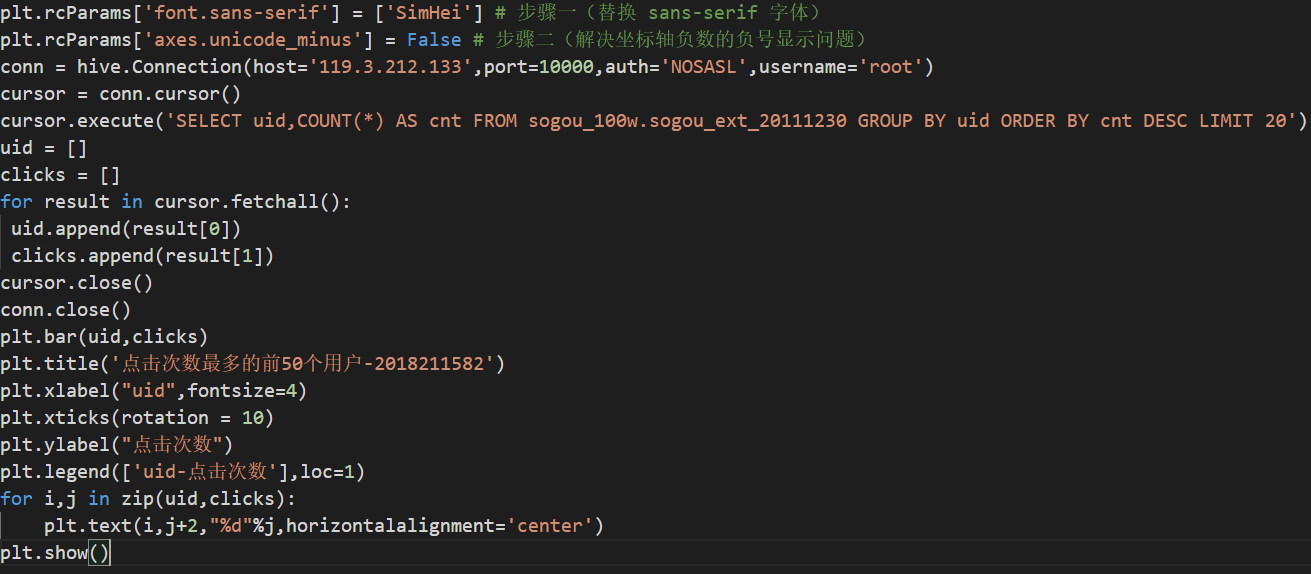
首先根据uid进行分组，统计出uid和每个uid的总共查询次数，再将

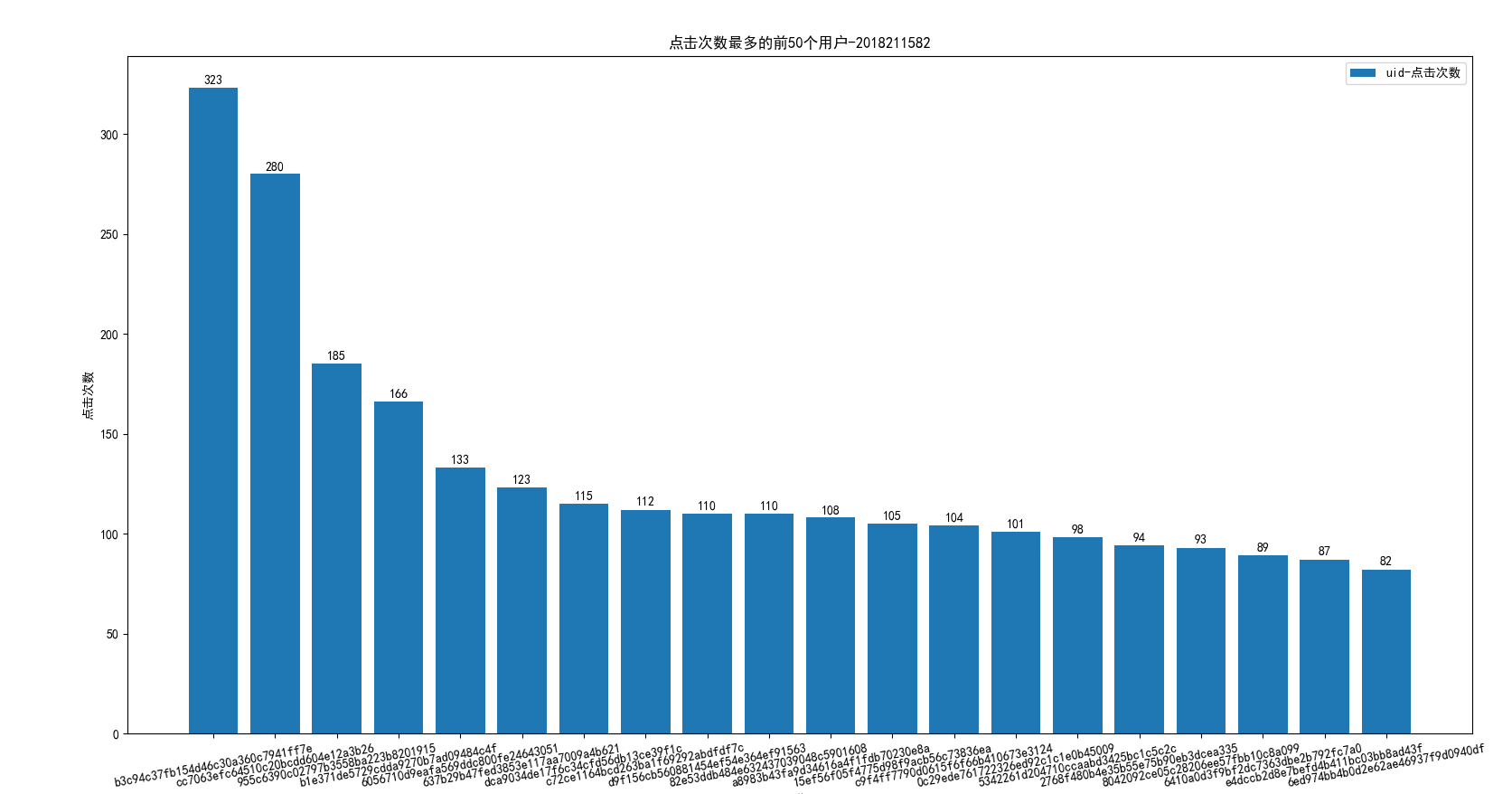
根据查询次数降序排列，取前20个数据，得到查询次数最多的前20个用户的用户ID



* **编写Python可视化代码：**

根据结果，可以编写python可视化代码如下，绘制柱状图





* **结果分析：**

由上述查询出的数据和柱状图展示，我们可以发现，查询总数前**20**的数

字范围在**80**次以上，大概有**14**个用户查询次数在**100**次以上，查询次数最多的为用户ID b3c94c37fb154d46c30a360c7941ff7e所带来的**323**次，查询次数排名第20的总次数为**82**次，这些数据可以为公司提取到特殊的用户群体和用户画像，用来说明产品的使用率和用户对于产品的依赖性。

**3.直接搜索url的准确率分析**

* **设计挖掘目标**

有时候，用户会直接输入url或url的一部分到检索框中进行检索，而

此时能否对url进行特殊的识别也是考验搜索引擎的一个地方，因此我们可以通过对那些keyword中包含url的搜索数据进行分析，可以分析出目标：

**用户点击的URL结果与用户输入的网址匹配的比例**

* **完成HiveSQL代码**

**SELECT** **SUM**(**IF**(instr(url,keyword)<=0,1,0)),**SUM**(**IF**(instr(url,keyword)>0,1,0)) **FROM** (**SELECT** \* **FROM** sogou\_100w.sogou\_ext\_20111230 **WHERE** keyword **LIKE** '%www%') a;

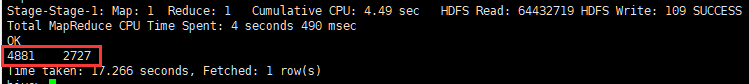
首先，使用LIKE语句筛选出keyword中包含’www’的数据，这些都是用

户直接输入网址进行查询的数据，之后我们使用instr子串检测函数，检测url中是否包含keyword，使用TF语句：

对于第一项，我们要统计不包含的数据项个数，因此若子串结果为负，则返回1，否则返回0，以此统计url中不含有关键词的数据个数。

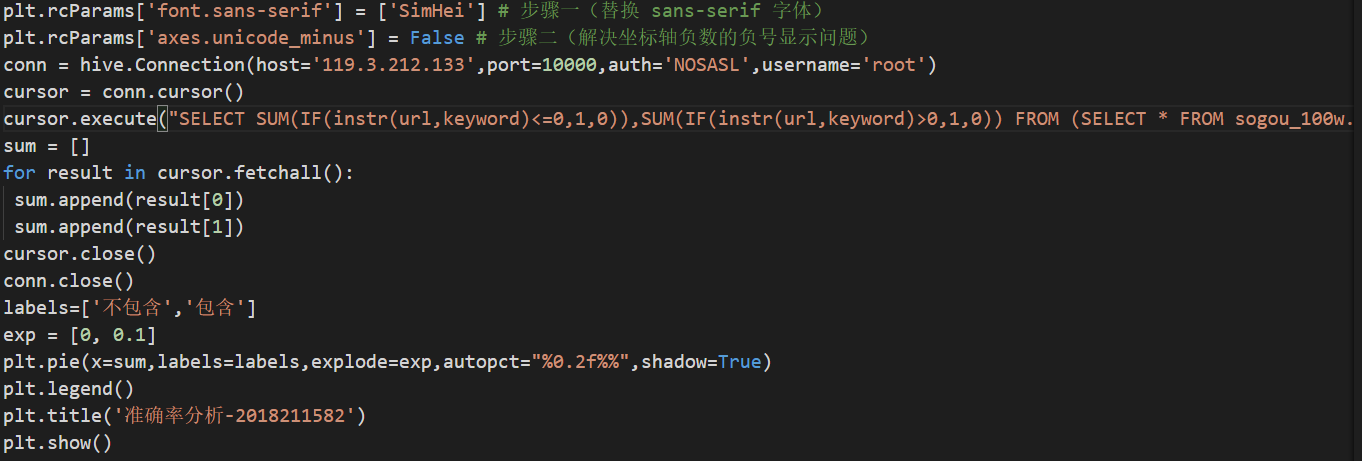
对于第二项，统计包含的数据项个数，如果子串检测结果为正，则返回1，否则返回0，以此统计url含有关键词的数据个数。

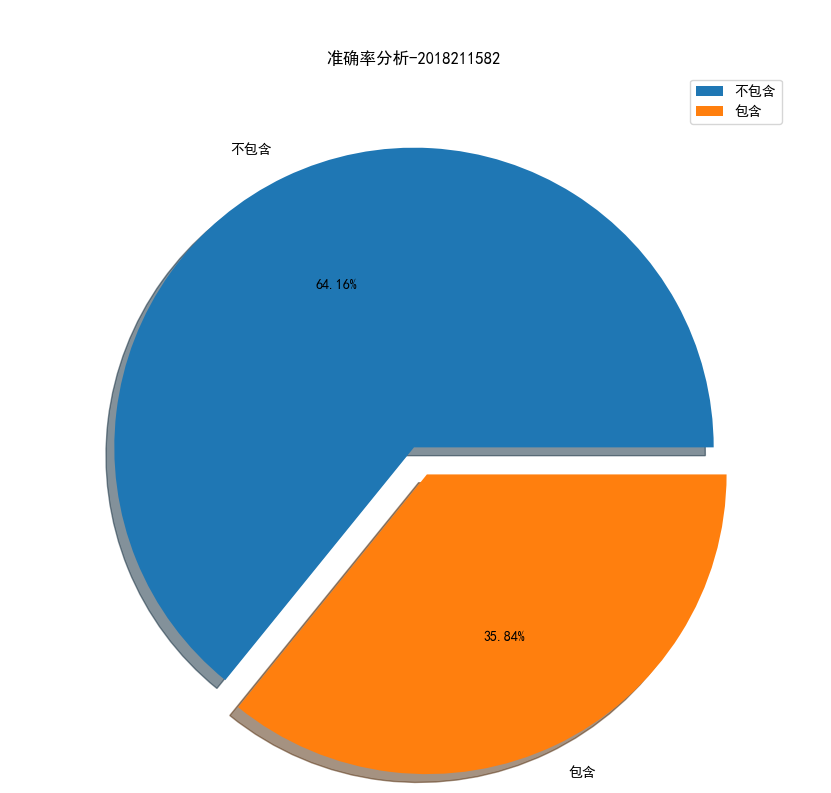
使用两个SUM函数统计总数并返回。



* **编写Python可视化代码：**

根据结果，可以编写python可视化代码如下，绘制饼图





* **结果分析：**

由上述饼状图和查询数据可以看出，用户查询关键词为网址时，用户最

后点击URL中包含该keyword的数据项共有**2727**项，占比**35.84%**，用户最后点击的URL中不包含该keyword的数据项共有**4881**项，占比**64.16%**。由此可以看出，若用户直接输入网址，以此为关键词查询，搜索出的结果似乎并不能够满足用户的需要，用户想搜索的网址很多并没有被检索到，这一方面可能是搜素引擎搜索算法的原因，另一方面也间接的表明，用户通过直接输入网址查询的方式并不理想，效果没有输入一些检索词而得到的准确。

**4.用户使用时间挖掘**

* **设计挖掘目标**

通过对用户使用时间的获取，将使用时间范围分为早上、中午、晚上和

午夜，分别代表8-12点、12-18点、18-23点、0点-8点。通过对使用时间的分析，可以挖掘出用户使用频率最高的时间段

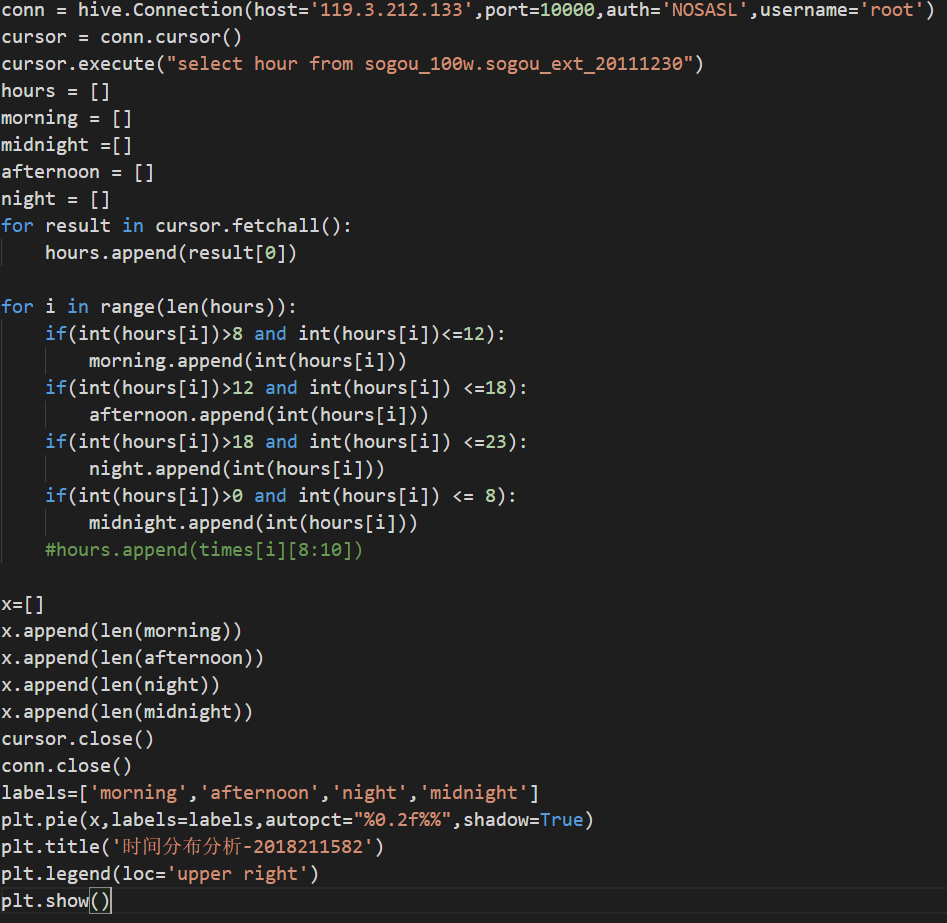
* **完成HiveSQL代码**

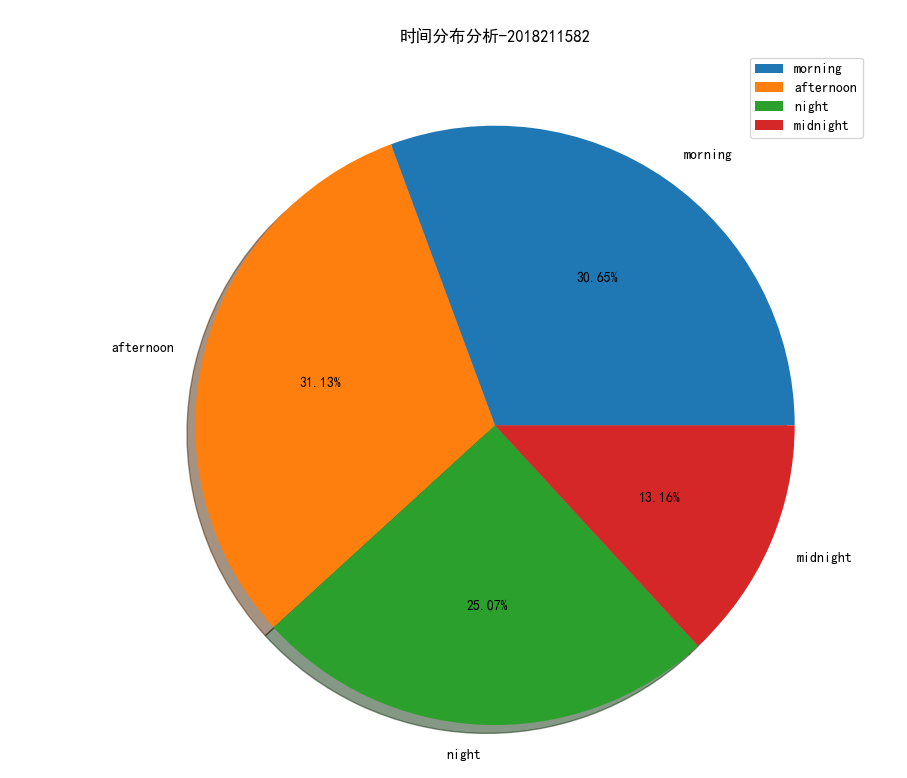
**SELECT** **HOUR** **FROM** sogou\_100w.sogou\_ext\_20111230

直接返回huor结果，之后使用python进行时间划分并展示。

* **编写Python可视化代码：**

根据结果，可以编写python可视化代码如下，绘制饼图





* **结果分析：**

可以看到30.65%的用户会选择在早上使用搜索引擎，31.13%的用户选择

在下午只用搜索引擎，25.07%的用户选择在晚上使用搜索引擎，13.16%的用户选择在午夜使用搜索引擎。整体来看，用户偏向早上和下午即白天时间使用，午夜使用的用户较少。

**附 录**

**实验分析python代码：**

**一、分析 rank 与用户点击次数之间的关系**

|  |
| --- |
| **from** matplotlib **import** lines  **from** pyhive **import** hive  **import** matplotlib.pyplot **as** plt  **from** matplotlib.pyplot **import** MultipleLocator  plt.rcParams['font.sans-serif'] = ['SimHei'] *# 步骤一（替换 sans-serif 字体）*  plt.rcParams['axes.unicode\_minus'] = False *# 步骤二（解决坐标轴负数的负号显示问题）*  conn = hive.Connection(host='119.3.212.133',port=10000,auth='NOSASL',username='root')  cursor = conn.cursor()  cursor.execute('select rank,count(\*) as cnt from sogou\_100w.sogou\_ext\_20111230 group by rank order by rank limit 30')  rank = []  clicks = []  **for** result **in** cursor.fetchall():  rank.append(result[0])  clicks.append(result[1])  cursor.close()  conn.close()  plt.plot(rank,clicks,marker='D')  plt.title('rank与点击次数折线图-2018211582')  plt.xlabel("rank")  plt.ylabel("点击次数")  plt.legend(['rank-点击次数'],loc=3)  x\_major\_locator=MultipleLocator(1)  *#把x轴的刻度间隔设置为1，并存在变量里*  ax=plt.gca()  ax.xaxis.set\_major\_locator(x\_major\_locator)  **for** i,j **in** zip(rank,clicks):  plt.text(i,j+2,"%d"%j,horizontalalignment='center')  plt.show() |

**二、查询次数最多的前20个的用户ID和次数**

|  |
| --- |
| **from** matplotlib **import** lines  **from** pyhive **import** hive  **import** matplotlib.pyplot **as** plt  **from** matplotlib.pyplot **import** MultipleLocator  plt.rcParams['font.sans-serif'] = ['SimHei'] *# 步骤一（替换 sans-serif 字体）*  plt.rcParams['axes.unicode\_minus'] = False *# 步骤二（解决坐标轴负数的负号显示问题）*  conn = hive.Connection(host='119.3.212.133',port=10000,auth='NOSASL',username='root')  cursor = conn.cursor()  cursor.execute('SELECT uid,COUNT(\*) AS cnt FROM sogou\_100w.sogou\_ext\_20111230 GROUP BY uid ORDER BY cnt DESC LIMIT 20')  uid = []  clicks = []  **for** result **in** cursor.fetchall():  uid.append(result[0])  clicks.append(result[1])  cursor.close()  conn.close()  plt.bar(uid,clicks)  plt.title('点击次数最多的前50个用户-2018211582')  plt.xlabel("uid",fontsize=4)  plt.xticks(rotation = 10)  plt.ylabel("点击次数")  plt.legend(['uid-点击次数'],loc=1)  **for** i,j **in** zip(uid,clicks):  plt.text(i,j+2,"%d"%j,horizontalalignment='center')  plt.show() |

**三.直接搜索url的准确率分析**

|  |
| --- |
| **from** matplotlib **import** lines  **from** pyhive **import** hive  **import** matplotlib.pyplot **as** plt  **from** matplotlib.pyplot **import** MultipleLocator  plt.rcParams['font.sans-serif'] = ['SimHei'] *# 步骤一（替换 sans-serif 字体）*  plt.rcParams['axes.unicode\_minus'] = False *# 步骤二（解决坐标轴负数的负号显示问题）*  conn = hive.Connection(host='119.3.212.133',port=10000,auth='NOSASL',username='root')  cursor = conn.cursor()  cursor.execute("SELECT SUM(IF(instr(url,keyword)<=0,1,0)),SUM(IF(instr(url,keyword)>0,1,0)) FROM (SELECT \* FROM sogou\_100w.sogou\_ext\_20111230 WHERE keyword LIKE '%www%') a")  sum = []  **for** result **in** cursor.fetchall():  sum.append(result[0])  sum.append(result[1])  cursor.close()  conn.close()  labels=['不包含','包含']  exp = [0, 0.1]  plt.pie(x=sum,labels=labels,explode=exp,autopct="%0.2f%%",shadow=True)  plt.legend()  plt.title('准确率分析-2018211582')  plt.show() |

**四、用户使用时间点分布**

|  |
| --- |
| **from** matplotlib **import** lines  **from** pyhive **import** hive  **import** matplotlib.pyplot **as** plt  **from** matplotlib.pyplot **import** MultipleLocator  plt.rcParams['font.sans-serif'] = ['SimHei'] *# 步骤一（替换 sans-serif 字体）*  plt.rcParams['axes.unicode\_minus'] = False *# 步骤二（解决坐标轴负数的负号显示问题）*  conn = hive.Connection(host='119.3.212.133',port=10000,auth='NOSASL',username='root')  cursor = conn.cursor()  cursor.execute("select hour from sogou\_100w.sogou\_ext\_20111230")  hours = []  morning = []  midnight =[]  afternoon = []  night = []  **for** result **in** cursor.fetchall():  hours.append(result[0])    **for** i **in** range(len(hours)):  **if**(int(hours[i])>8 **and** int(hours[i])<=12):  morning.append(int(hours[i]))  **if**(int(hours[i])>12 **and** int(hours[i]) <=18):  afternoon.append(int(hours[i]))  **if**(int(hours[i])>18 **and** int(hours[i]) <=23):  night.append(int(hours[i]))  **if**(int(hours[i])>0 **and** int(hours[i]) <= 8):  midnight.append(int(hours[i]))  *#hours.append(times[i][8:10])*    x=[]  x.append(len(morning))  x.append(len(afternoon))  x.append(len(night))  x.append(len(midnight))  cursor.close()  conn.close()  labels=['morning','afternoon','night','midnight']  plt.pie(x,labels=labels,autopct="%0.2f%%",shadow=True)  plt.title('时间分布分析-2018211582')  plt.legend(loc='upper right')  plt.show() |