移动设备数据集分析

摘要：如今单纯的软件或硬件已经不被市场看好。想要卖出软硬件必须构建相应的生态链而软硬件是构成一个系统或者一个品牌生态链最重要的两大部分。硬件开发者针对当前最受欢迎的 app 继而推出相关高性能硬件。以及 app 设计者如何根具热门机型推出该机型优化策略已经成为软硬件厂商需要思考的问题。其中对 app 使用者年龄性别预测，app 地区活跃度，时段活跃度，对于 app 团队如何针对特定人群特定时段设计优化起着非常重要的作用。这些关系这个 app 的生命周期。以下详细介绍了基于移动设备数据构建的性别年龄预测模型，以及热门机型，热门 app 统计，相关地区 app 使用活跃度，活跃时段等。

关键词：移动设备数据集；热门 app；热门硬件；性别年龄预测

目　　录

一、数据分析目标与任务……

1.1 ……..1

1.2……

二、数据预处理……

2.1 ……..

2.2……

三、数据探索……

3.1 ……..

3.2……

四、数据分析模型……

4.1 ……..

4.2……

五、方案评估……

一、数据分析目标与任务

目标：实现对给定的移动设备数据集分析。预测使用者性别年龄。

任务：1、统计最多使用的手机品牌以及最受欢迎型号。

2、统计 app 最活跃日期及当前使用量。

3、统计 app 最活跃时间段及每个时间段使用情况。

4、分析 app 最活跃地区及该地区范围使用量情况。

5、将使用量最高的 app 统计显示其类别。

6、建立性别年龄预测模型并检验其准确率。

该实验用 python 编写在 pycharm 平台编译，运行于 x86/64Windows 平台

二、数据预处理

1. 数据说明（说明数据规模、数据文件、以及各字段等基本信息，并给出数据样本和样本说明。）

数据集名称	gender_age_train.csv
数据规模（row*colum）	74645*4
各字段基本息	device_id：仪器 id,用于标识移动设备。 Gender：性别 Age：年龄 Group：记录性别年龄并划分在相应组别
数据样本	device_id gender age group 8076087639492060000 M 35 M32-38
样本说明	设备 id:-8076087639492060000 使用者性别：男

	年龄：35 组别划分：男 32-38 岁
数据集名称	gender_age_test.csv
数据规模（row*colum）	112072*1
各字段基本息	device_id：仪器 id,用于标识移动设备。
数据样本	device_id 1002079943728930000
样本说明	设备 id: 1002079943728930000
数据集名称	events.csv
数据规模（row*colum）	3252950*5
各字段基本息	event_id:事件 id device_id：设备 id timestamp：时间戳-记录访问时间 longitude：经度 latitude：维度
数据样本	event_id device_id timestamp longitude latitude 1 2.92E+16 2016/5/1 0:55 121.38 31.24
样本说明	事件 id: 1 设备 id：29182687948017100 访问时间：2016/5/1 0:55 经度：121.38 纬度：31.24
数据集名称	app_labels.csv
数据规模（row*colum）	459943*2

各字段基本息	app_id：app 的 id label_id：该 app 对应的分类标签
数据样本	app_id label_id 7324884708820020000 251
样本说明	app 的 id：7324884708820020000 所属分类：251
数据集名称	label_categories.csv
数据规模（row*colum）	930*2
各字段基本息	label_id：app 分类标签 category：app id 对应的 app 类别
数据样本	label_id category 1 2 game-game type
样本说明	标签 id：1 所属分类：未知 标签 id： 所属分类：游戏-游戏型
数据集名称	phone_brand_device_model.csv
数据规模（row*colum）	187245*3
各字段基本息	device_id：设备 id phone_brand：手机品牌 device_model：手机型号
数据样本	device_id phone_brand device_model -8890648629457970000 小米 红米
样本说明	设备 id：-8890648629457970000 手机品牌：小米 手机型号：红米

数据集名称	app_events.csv
数据规模（row*colum）	1048576*4
各字段基本息	event_id：事件 id app_id：app 的 id is_installed：是否已经安装 is_active：当前是否运行
数据样本	event_id app_id is_installed is_active 2 5.93E+18 1 1
样本说明	事件 id：2 app id：5927333115845830000 是否已经安装：已经安装 当前是否运行：是

2. 数据清洗（说明对数据做了哪些清洗，效果如何？）

   经纬度数据经测试排序后出现在（0，0）处出现数据最多，明显是一个错误数据故在将其转换为另一个经纬度数据集 long_la_dic 时将（0，0）去掉。效果是，处理后的数据恢复正常。

3. 数据处理（说明为便于分析，还做了哪些数据处理？如数据集成、数据变换、特定数据提取、数据规约等。本项目数据预处理的难点在哪里？理由是什么？）

   1. 将手机品牌数据分组统计方便显示市场上已有手机种类
   2. 将排名前五的手机品牌以及各自型号分组统计方便显示热门手机
   3. 将最受欢迎设备型号以及使用数量统计显示最热门机型
   4. 将使用日期，当天使用量以及时间段，每时间段使用量以及经纬度，该地区使用量各自两两提取转化成 dataframe 方便后面统计使用
   5. 以 app_id 为索引将 category 转化为 dataframe 方便后面统计使用
   6. 将性别年龄分组统计放入列表方便后面使用
   7. 将经纬度数据清理并集成数据集以便绘制在地图上

三、数据探索分析

1. 结合可视化呈现，对数据进行探索性分析（详细说明从哪些方面对数据进行探索性的分析）

   1. 从已有手机品牌数据选出最受欢迎手机品牌进而在最受欢迎品牌中选出其各自最热门机型最终统计出所有手机中使用量最大的手机品牌和机型。

   2. 统计最受欢迎的手机设备型号进行可视化呈现

   3. 从时间日期找出 app 最活跃使用日以及当天使用情况在从当日分析 app 集中活跃时间段以及在此时间段使用量，并且找出哪些地区最活跃并找出该地区活动范围。

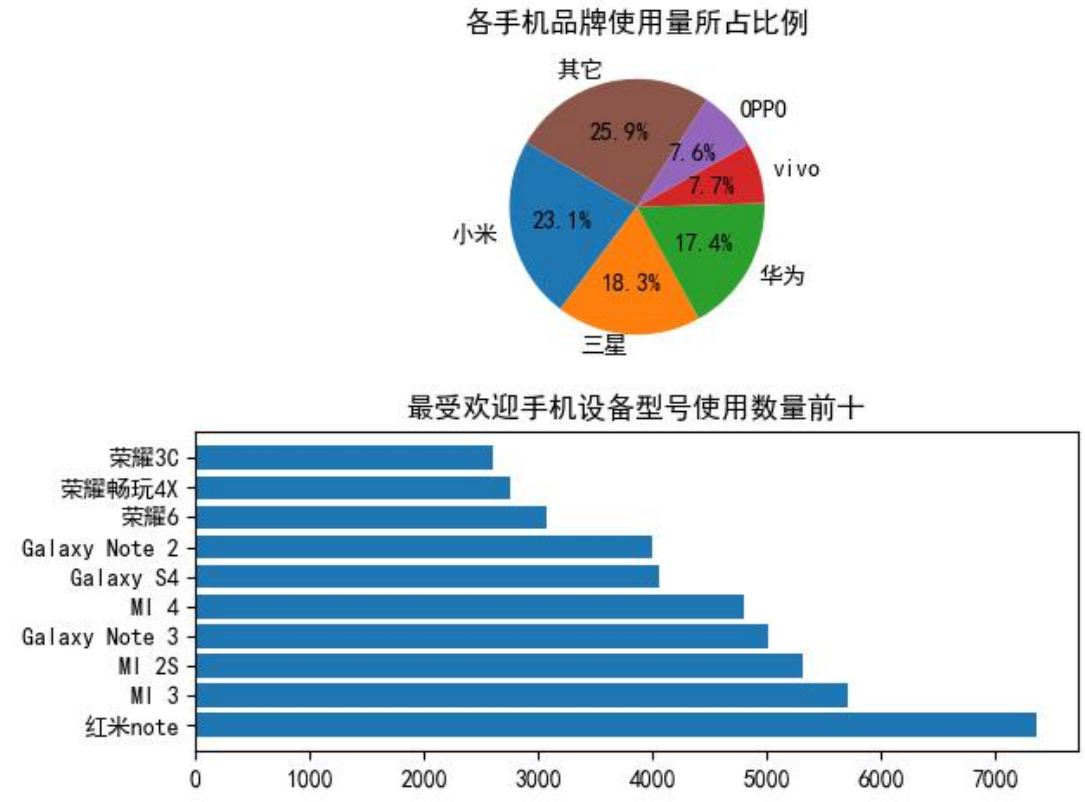
   4. 筛选出最受欢迎的 appid 及其类别

   5. 绘制 app 使用分部区域以便找出最活跃地区。

   6. 通过性别年龄筛选统计验证模型准确性

2. 可视化呈现结果（给出可视化呈现界面图，并分析结果做说明）

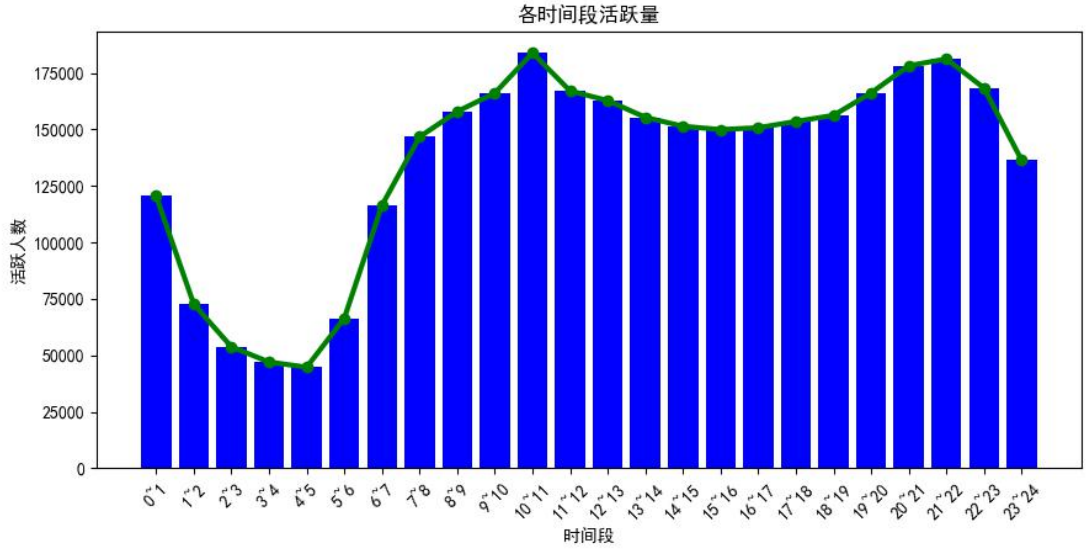
   1）：

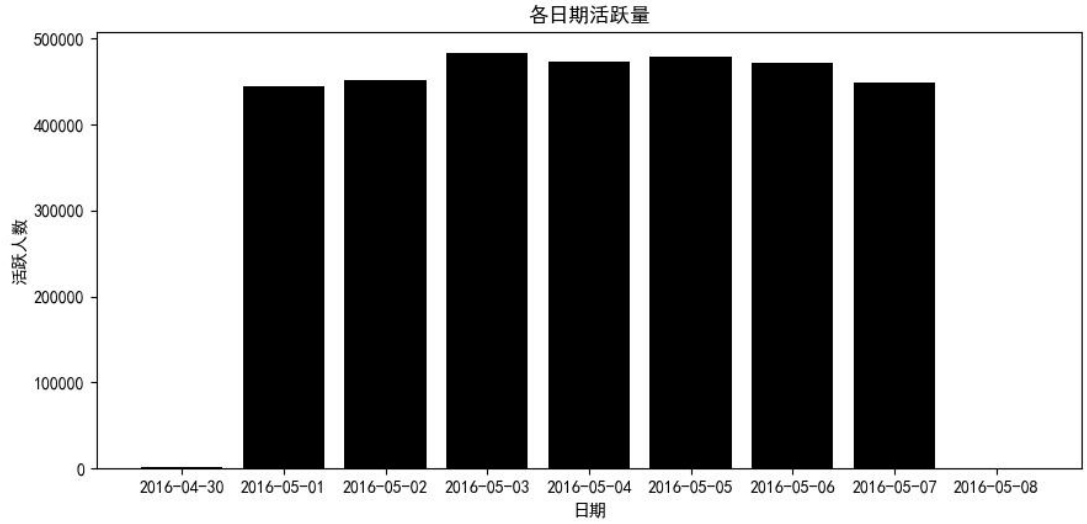


各品牌使用量表明小米手机是使用最多的品牌。前五分别是：小米，三星，华为，vivo，oppo 联合占比 74.1%其他品牌占比 25.9%所以 app 开发者厂商可以根据该图重点优化 app 在小米等前五品牌手机上的使用体验。而最受欢迎前 10 手机数据，数据显示红米 note 是最受欢迎的机型。该可以更进一步

具体到具体机型 app 厂商可以具此推出相关联名手机活动促进手机销售提高 app 知名度等。

2）：

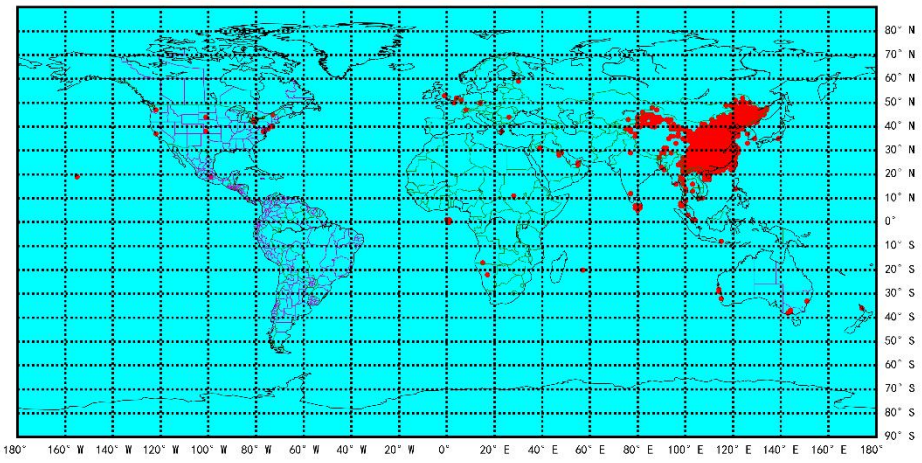




根据统计数据绘制出最活跃时间段与最活跃的日期。根据数据显示 app 使用最活跃时间段为上午 10-11 点晚上 21-22 点。符合人们的正常作息时间。根据这张图可以在使用最活跃阶段注意 app 服务器防止宕机以及在该时段推送广告盈利等。日期图显示在 5.3 日最为活跃。

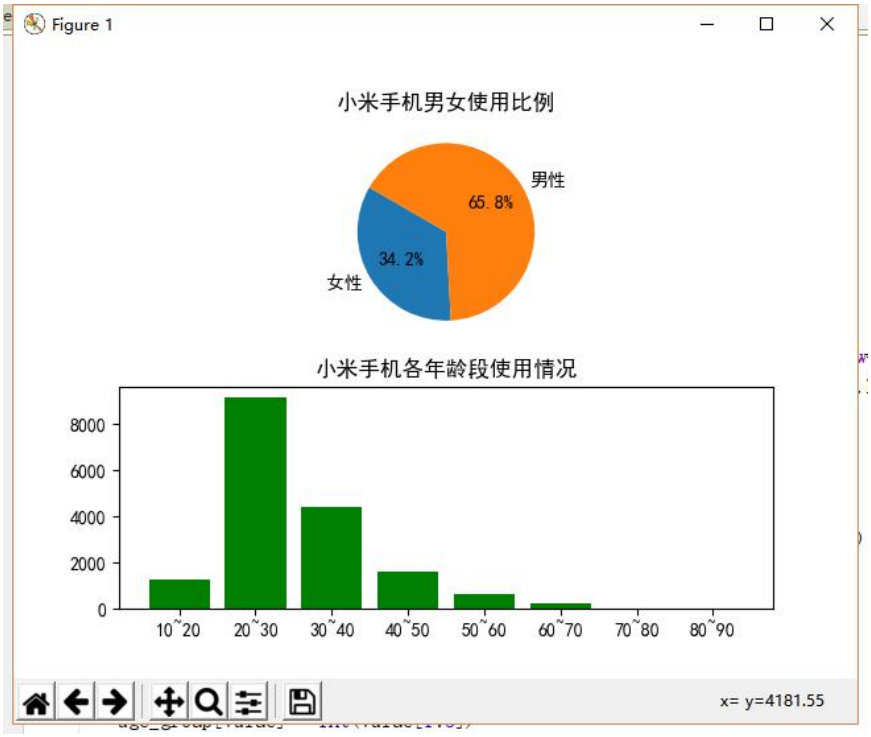
3）：

App使用区域图



将 app 使用记录下的经纬度数据绘制在地图上显示在中国使用的最多。这表明这应该是统计的中国区域数据集。而世界范围内也有少数其他地方有数据分散。通过此数据表明 app 开发者可根据中国人习惯为中国提供更好的服务

4）



可视化一个范例，描述手机品牌的男女使用比例及各年龄使用数量情况

四、数据分析模型

1. 结合分析的目标，拟采用哪一种模型（如聚类、分类、回归）开展分析（详细说明采用模型和方法有哪些？理由是什么？）
2. 采用的是分类模型，运用 SVM 模型

   步骤如下：

   1.把数据划分成特征和标签两类

   2.用 train_test_split 方法随机分成训练集数据和测试集数据

   3.用 MinMaxScaler()和 fit_transform()方法将特征数据进行归一化操作

   4.用 SVM 模型中的 SVC()方法训练特征，生成模型

   5.最后用测试集取测试数据
3. 模型评估(针对选用的模型，说明采用什么评估方法对模型进行评估)

   使用准确率评估

五、代码及运行结果

1、分析最受欢迎的手机品牌

分析排名前五的各手机品牌最受欢迎的手机型号

分析最受欢迎手机型号

代码：

```
import pandas as pd
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif']=['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus']=False #用来正常显示负号
phone = pd.read_csv('phone_brand_device_model.csv')
gender_age_train = pd.read_csv('gender_age_train.csv')
brand_count = phone.groupby('phone_brand')['phone_brand'].count()
# print(type(brand_count))
print('市场上已有手机品牌种类数为：%d，品牌名称如下：'%len(brand_count))
for i in brand_count.index:
    print(i,end = ',')
brand_sort = brand_count.sort_values(ascending = False)
print('\n其中使用最多即最受欢迎手机品牌前五如下：\n',brand_sort.head())
model_count = phone.groupby(['phone_brand','device_model'])['device_model'].count()
# print(model_count)
print('排名前五的手机品牌版本及各型号前五使用数量排序如下：\n')
for i in brand_sort.index[:5]:
    print(model_count.loc[i,:].sort_values(ascending = False).head())

device_model_count = phone.groupby('device_model')['device_model'].count()
device_model_sort = device_model_count.sort_values(ascending = False)
print('\n最受欢迎手机设备型号使用数量前五如下long_la_dic：\n',device_model_sort.head())
```

```
labels = list(brand_sort.index[0:5])
labels.append('其它')
values = brand_sort.head().values
values = list(values)
sum = 0
for i in values:
    sum = sum+i
others_brand_count = len(phone['phone_brand'])-sum
values.append(others_brand_count)
plt.figure()
plt.subplot(2,1,1)
plt.pie(values,labels=labels,autopct='%1.1f%%',shadow=False,startangle=150)
plt.title('各手机品牌使用量所占比例')

plt.subplot(2,1,2)
plt.barh(device_model_sort.index[0:10],device_model_sort[0:10])
plt.title('最受欢迎手机设备型号使用数量前十')
plt.show()
```

运行结果：

其中使用最多即最受欢迎手机品牌前五如下：

```
 phone_brand
小米       43210
三星       34286
华为       32564
vivo     14395
OPPO     14289
Name: phone_brand, dtype: int64
```

排名前五的手机品牌版本及各型号前五使用数量排序如下：

```
phone_brand  device_model
小米           红米note          7358
             MI 3            5712
             MI 2S           5308
             MI 4            4798
             红米1S            2479
Name: device_model, dtype: int64
phone_brand  device_model
三星           Galaxy Note 3    5019
             Galaxy S4       4059
             Galaxy Note 2    3993
             Galaxy S3       2415
             Galaxy S5       1758
Name: device_model, dtype: int64
phone_brand  device_model
华为           荣耀6             3076
             荣耀畅玩4X          2754
             荣耀3C            2598
             Mate 7          2406
             荣耀6 Plus         1690
Name: device_model, dtype: int64
```

```
phone_brand  device_model
vivo         X5Pro           891
             X3T             889
             X3L             694
             Xplay           596
             X5SL            576
Name: device_model, dtype: int64
phone_brand  device_model
OPPO         R7              1457
             R7 Plus         1058
             R7s              932
             Find 7           821
             R3               703
Name: device_model, dtype: int64
```

2、

使用 App 最活跃的日期及当天 App 使用量情况

使用 App 最活跃的时间段及每个时间段使用量情况

使用 App 最活跃地区及该地区范围使用量情况

代码：

```
import pandas as pd
from mpl_toolkits.basemap import Basemap
import numpy as np
import matplotlib.pyplot as plt
app_events = pd.read_csv('app_events.csv')
app_labels = pd.read_csv('app_labels.csv')
label_categories = pd.read_csv('label_categories.csv')
events = pd.read_csv('events.csv')
times = events['timestamp'].values
longitudes = events['longitude'].values
latitudes = events['latitude'].values
dates = {}
dates1 = {}
time_span = {}
time_span1 = {}
longitudes_dic = {}
latitudes_dic = {}
long_la_dic = {}
lon=[]
lat=[]
```

```
for i in times:
    value = i.split(' ')
    date = value[0]
    date2int = int(date.replace('-',''))
    hour = value[1].split(':')[0]
    hour1 = int(hour)
    hour = hour + '~' + str(int(hour) + 1)
    dates[date] = 1 + dates.get(date, 0)
    time_span[hour] = 1 + time_span.get(hour, 0)
    time_span1[hour1] = 1 + time_span1.get(hour1, 0)
    dates1[date2int] = 1 + dates1.get(date2int, 0)
for index in range(len(longitudes)):
    i,j = int(longitudes[index]),int(latitudes[index])
    if i==0 and j==0:
        continue
    long_la = str(i) + ',' + str(j)
    long_la_dic[long_la] = 1 + long_la_dic.get(long_la, 0)
    lon.append(i)
    lat.append(j)

dates = sorted(dates.items(),key = lambda x:x[1],reverse = True)
dates1 = sorted(dates1.items(),key = lambda x:x[0])
dates = pd.DataFrame(dates,columns = ['使用日期','当天使用量'])
dates1 = pd.DataFrame(dates1,columns = ['使用日期','当天使用量'])
print('使用App最活跃的日期及当天App使用量情况为：\n',dates)
```

```
hours = sorted(time_span.items(),key = lambda x:x[1],reverse = True)
hours1 = sorted(time_span1.items(),key = lambda x:x[0])
hours = pd.DataFrame(hours,columns = ['时间段（小时）','每个时间段使用量'])
print('使用App最活跃的时间段及每个时间段使用量情况为：\n',hours)

hours1 = pd.DataFrame(hours1,columns = ['时间段（小时）','每个时间段使用量'])
long_la_dic = sorted(long_la_dic.items(),key = lambda x:x[1],reverse = True)
long_la_dic = pd.DataFrame(long_la_dic,columns = ['纬度，经度','该地区使用量'])
print('使用App最活跃地区及该地区范围使用量情况为：\n',long_la_dic)

# 画时间峰值图
a = []
x = []
for i in hours1['时间段（小时）']:
    x.append(str(i)+'~'+str(i+1))
y=hours1['每个时间段使用量'].values.tolist()
for value in dates1['使用日期'].values.tolist():
    value = list(str(value))
    value.insert(4, '-')
    value.insert(7, '-')
    a.append(''.join(value))
b=dates1['当天使用量'].values.tolist()
```

```
# 时间
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus']=False
p1 = plt.figure(figsize=(9,9))
ax1 = p1.add_subplot(2,1,1)
plt.title('各时间段活跃量')
plt.ylabel('活跃人数')
plt.xlabel('时间段')
plt.xticks(rotation = 45)
plt.plot(x,y,'ro-',lw=3,color = 'g')
plt.bar(x,y,color='b')
# 日期
ax2 = p1.add_subplot(2,1,2)
plt.title('各日期活跃量')
plt.ylabel('活跃人数')
plt.xlabel('日期')
plt.bar(a,b,color='k')
plt.show()
```

```
label_categories.dropna(axis = 0,how = 'any',inplace = True)
label_id = label_categories['label_id']
category = label_categories['category']
categories = pd.DataFrame(category,index = label_id,columns = ['category'])
app_id = app_events.groupby('app_id')['is_active'].sum()
app_popular = app_id.sort_values(ascending = False).head()
popular = {}
for i in app_popular.index:
    for j in range(len(app_labels['app_id'])):
        if i == app_labels.loc[j,'app_id']:
            label_id = app_labels.loc[j,'label_id']
            popular[i] = categories.loc[label_id,'category']
            break
print('排名前五的app_id及category是：')
for i in popular.items():
    print(i)
```

```
plt.rcParams['font.sans-serif']=['SimHei'] #用来正常显示中文标签
m = Basemap(width = 10000000000,height = 10000000000)  # 实例化一个map
m.drawcoastlines(linewidth=0.25)  # 画海岸线
m.drawstates(linewidth=0.25, color='m') #画出州界线
m.drawcountries(linewidth=0.25, color='g')# 画出国境线
m.drawmapboundary(fill_color = 'aqua') #整个地球蓝色
#m.fillcontinents(color = 'coral', lake_color = 'aqua') #大陆棕色江河海为蓝色
#m.shadedrelief()# 绘制阴暗的浮雕图
parallels = np.arange(-90., 90., 10.)  # 画纬度，范围为[-90,90]间隔为10
m.drawparallels(parallels, labels=[False, True, True, False],fontsize=5)
meridians = np.arange(-180., 180., 20.)  # 经度，范围为[-180,180]间隔为10
m.drawmeridians(meridians, labels=[True, False, False, True],fontsize=5)

# 绘制区域使用点图

lon, lat = m(lon, lat)
m.scatter(lon, lat, s=5,marker = '.', color = 'r')
plt.title('App使用区域图')
plt.savefig('map.png', dpi=1000)
plt.show()
```

运行结果：

使用App最活跃的日期及当天App使用量情况为：

	使用日期	当天使用量
0	2016-05-03	483293
1	2016-05-05	478999
2	2016-05-04	473487
3	2016-05-06	471730
4	2016-05-02	451546
5	2016-05-07	448345
6	2016-05-01	444589
7	2016-04-30	959
8	2016-05-08	2

使用App最活跃的时间段及每个时间段使用量情况为：

	时间段（小时）	每个时间段使用量
0	10~11	183839
1	21~22	181175
2	20~21	178179
3	22~23	168246
4	11~12	167025
5	19~20	166160
6	09~10	166061
7	12~13	162745
8	08~9	157896
9	18~19	156209
10	13~14	155337

使用App最活跃地区及该地区范围使用量情况为：

	纬度，经度	该地区使用量
0	116,39	99208
1	121,31	85581
2	113,23	82803
3	1,1	76933
4	104,30	72335
5	113,22	71549
6	114,22	61450
7	120,30	56829
8	120,31	51601
9	114,30	49863
10	116,40	39491

排名前五的app_id及category是：

(8693964245073640147, '6.0-5.6 inches')
(5927333115845830913, '6.0-5.6 inches')
(4348659952760821294, '6.0 inches above')
(3433289601737013244, '6.0 inches above')
(628020936226491308, '6.0 inches above')

代码：

```
plt.rcParams['font.sans-serif']=['SimHei'] #用来正常显示中文标签
plt.rcParams['axes.unicode_minus']=False #用来正常显示负号
gender_age_train = pd.read_csv('gender_age_train.csv')
phone = pd.read_csv('phone_brand_device_model.csv')

phone_categories = phone.groupby(['phone_brand','device_model'])['phone_brand'].count()
num = 1
num1 = 0
brand_dic = {}
model_dic = {}
temp = 'E人E本'
for i in phone_categories.index:
    if temp == i[0]:
        brand_dic[i[0]] = num
        num1 += 1
        model_dic[i[1]] = num + float('0.' + str(num1))
    else:
        num1 = 1
        num+=1
        brand_dic[i[0]] = num
        model_dic[i[1]] = num + float('0.' + str(num1))
    temp = i[0]
brand2num = []
model2num = []
for i in range(len(phone['phone_brand'])):
    brand_value = phone.loc[i,'phone_brand']
    model_value = phone.loc[i,'device_model']
    brand2num.append(brand_dic[brand_value])
    model2num.append(model_dic[model_value])
phone['brand2num'] = brand2num
phone['model2num'] = model2num
```

```
connect = gender_age_train.join(phone.set_index('device_id'),on='device_id',how='left')
print('gender_age_train表与phone表根据关键字device_id连接后表如下：\n',connect.head())
connect.dropna(axis = 0,how = 'any',inplace = True)
connect.drop_duplicates(subset=['device_id'],keep='first',inplace=True)
connect.reset_index(inplace = True)
connect.drop(['index'],axis = 1,inplace = True)
gender_count = connect.groupby(['phone_brand','gender'])['phone_brand'].count()
age_count = connect.groupby(['phone_brand','age'])['phone_brand'].count()
age_dup = connect['group'].drop_duplicates()
age_group = {}
for value in age_dup:
    if value.endswith('-') or value.endswith('+'):
        age_group[value] = int(value[1:3])
    else:
        age_group[value] = int((int(value[1:3])+int(value[4:6]))/2)
ages_reset = []
for i in connect['group']:
    ages_reset.append(age_group[i])
connect['ages_reset'] = ages_reset
```

```
mi_age = age_count['小米']
count = {'10~20':0,'20~30':0,'30~40':0,'40~50':0,'50~60':0,'60~70':0,'70~80':0,'80~90':0}
for i in mi_age.index:
    value = mi_age[i]
    if i<=20:
        count['10~20'] += value
    elif 20<i<=30:
        count['20~30'] += value
    elif 30 < i <= 40:
        count['30~40'] += value
    elif 40 < i <= 50:
        count['40~50'] += value
    elif 50 < i <= 60:
        count['50~60'] += value
    elif 60 < i <= 70:
        count['60~70'] += value
    elif 70 < i <= 80:
        count['70~80'] += value
    elif 80 < i <= 90:
        count['80~90'] += value
plt.figure()
plt.subplot(2,1,1)
plt.title('小米手机男女使用比例')
plt.pie([i for i in gender_count['小米']],labels =['女性','男性'],autopct='%1.1f%%',shadow=False,startangle=150)
plt.subplot(2,1,2)
plt.title('小米手机各年龄段使用情况')
plt.bar(count.keys(),count.values(),color ='g')
plt.show()
```

```
label = []
for i in connect['gender']:
    if i == 'M':
        label.append(0)
    else:
        label.append(1)
label = {'label':label[10000:20000]}
label = pd.DataFrame(label)
label1 = connect['ages_reset'][10000:20000]
data = connect[['brand2num','model2num']].iloc[10000:20000,:]
print('经过一系列处理后，connect表格最终形式如下：\n',connect.head())

train_data,test_data,train_label,test_label = train_test_split(data,label,test_size = 0.12,random_state = 10)
train_data_scaler = MinMaxScaler().fit_transform(train_data)
test_data_scaler = MinMaxScaler().fit_transform(test_data)
test_label = test_label.reset_index(drop = True)
svm = SVC().fit(train_data_scaler,train_label)
predict = svm.predict(test_data_scaler)
count = 0
```

```
for i in range(len(predict1)):
    if predict1[i] == test_label1[i]:
        count+=1
print('年龄预测准确率为：{:.2%}'.format(count/len(predict1)))

gender_age_test = pd.read_csv('gender_age_test.csv')
test = pd.concat([gender_age_test,phone],ignore_index=True)
test.dropna(axis = 0,how = 'any',inplace = True)
data = test[['brand2num','model2num']].iloc[10000:20000,:]
data = MinMaxScaler().fit_transform(data)
result = svm.predict(data)
result1 = svm1.predict(data)
print('性别预测结果如下：\n',result)
print('年龄预测结果如下：\n',result1)
```

运行结果：

```
gender_age_train表与phone表根据关键字device_id连接后表如下：
             device_id gender  age  ... device_model brand2num model2num
0 -8076087639492063270      M   35  ...          MI 2        52     52.30
1 -2897161552818060146      M   35  ...          MI 2        52     52.30
2 -8260683887967679142      M   35  ...          MI 2        52     52.30
3 -4938849341048082022      M   30  ...        红米note          52     52.25
4   245133531816851882      M   30  ...          MI 3        52     52.70

[5 rows x 8 columns]
经过一系列处理后，connect表格最终形式如下：
             device_id gender  age  ... brand2num model2num ages_reset
0 -8076087639492063270      M   35  ...        52     52.30         35
1 -2897161552818060146      M   35  ...        52     52.30         35
2 -8260683887967679142      M   35  ...        52     52.30         35
3 -4938849341048082022      M   30  ...        52     52.25         30
4   245133531816851882      M   30  ...        52     52.70         30


性别预测准确率为：64.58%
C:\Users\1938108903\AppData\Roaming\Python\Python37\site-packages\sklearn\svm\base.py:193:
  "avoid this warning.", FutureWarning)
年龄预测准确率为：15.33%
```

```
性别预测结果如下：
 [0 0 0 ... 0 0 0]
年龄预测结果如下：
 [30 30 24 ... 30 30 30]
```

六、方案评估

（定性评估以下方面：本次课程设计是否达到预期目标、完成了所有设计任务？课程设计难度如何？还有哪些尚待解决之处？）

该次设计基本达到了预期目标能完成所设计的任务。

由于性别预测所给出的测试集只有一列数据故不知道如何通过测试集来评估训练集，所以引入了手机型号品牌来预测性别年龄。这个方法准确率也不高，但是已经想不出其他办法来测试训练集，这是本课程最难的地方。不知道选哪些数据作为测试集，预测准确率低也是我们尚待解决的问题。

其中不足的地方还在于未将可视化做的更加美观。比如我们想结合 HTML 和 echarts 做出网页版数据可视化但是由于学识技术限制未能完成非常遗憾

还有便是我觉的将数据绘制在地图上虽然不如网页版可以点击选取但是也能算该次设计的亮点。