1--lastest_data.py

file_model　　保存 model 中的文件名（我这里可能缺这些原文件）

根据上面的文件名新建一个文件名（只包含模型和日期如 4g_20160601_new），以此保存处理后的数据

某一文件名下面的数据如下

15358998669,无锡,手机|华为|荣耀畅玩 4C 运动版|SCL-AL00,2

提取号码和手机型号保存到新文件中

最后将保留的型号和号码去重，分别写入型号文件 models_all.txt 和号码文件 users_all 保存

时间戳

2--lastest_data_app.py

将 D:\Python27\dianxin\latest data\data 下所有文件中的所有 app 记录下来保存到 file_apps 中

做出来结果不一样

3--apps_class00.py

将 apps_all 里的 app 进行分类，将 data[1]=“00”的类别写入结果？但是 apps_all 里的数据只有 app 类型，**代码中分为两列，没有看明白**

4--apps_all_class.py

根据将 apps_all 里剩下全部的 data[1]项对 app 进行分类，并将结果写入文件，文件只有一项，即属于该类别的 app 名

分类中 0-18 表示什么？

4--model_3g+4g.py

合并 3g 和 4g 的数据（从 526 到 610），合并后的文件名如 model_20160528.txt，里面的数据有两项，号码和机型

5--top500_eachday.py

0526-0610

"app_20160" + date + "_new.txt"由第 6 个

apps_need 表示什么

南京|360 云盘|18013918761|0|182|60|0|0|12747879|1298437|0

```
app = data[1]
user = data[2]
flow = int(data[7]) + int(data[8]) + int(data[9]) + int(data[10])
```

第三行表示什么？------四个时间段的流量

首先更新 user（有则免之，无则加之），统计从 0526-0610 这段时间所有符合 apps_need 的 app 所耗流量并放入 list

```
need = round(app_need / flow, 2)
```

获得结果为前 500 个用户和所耗流量

时间戳 10000 行

6--lastest_data_app_new.py

处理 app 数据文件（D:\Python27\dianxin\latest data\data），即 app 使用数据，提取相关信息

```
city = data[0]
app = data[3]
user = data[4]
nums_1 = data[13]
nums_2 = data[14]
nums_3 = data[15]
nums_4 = data[16]
flow_1 = data[17]
flow_2 = data[18]
flow_3 = data[19]
flow_4 = data[20]
```

结果存入"app_" + data_time + "_new.txt"，如 app_20160528_new.txt

6--user_error_all.py

0526-0610

读取文件数据"model_20160" + date + "_error.txt"如 model_20160526_error.txt，（得到这步的程序没有——有，在下一步）数据内容如下：

18168558105|iPhone6Plus|Ascend GX1|麦芒 3S|M20-T|R8205|Galaxy A5|P8|红米 Note|R7|小米 Note|荣耀畅玩 4|Ascend Mate7|麦芒 4|红米 2

统计用户所用过的所有机型

被注释掉的用户换机那一段应该是没用的

类似更新用户所用机型，这个代码没有看的很明白

7--model_3g+4g_new.py

0526-0610

读取文件数据"model_20160" + date + ".txt"

过滤 ipad, PAD，电视（**还是应该为电脑**）机型

```
dict_users[user][pos] = max(model,model_old)
```

如果都是 iphone 机型，谁大保留谁（**如果旧机型为 iphone，新不为 iphone，也算换机，这个可以改吗**）

```
elif item in model:
    dict_users[user][pos] = model
    temp = 1
```

这一段的作用是什么？---因为一样，用现在值覆盖？

以上即为更新用过的机型

根据机型长度分为未换机用户（1），换机用户（2），异常用户（else）

时间戳

8--user_change_all.py

查看异常用户是否重复

```
model_ret = list(set(model) ^ set(model_old))
```

按位异或，得到什么？

处理 change 中异常用户，这个代码没看懂

9--user_error_all_new.py

0526-0619

异常用户更新机型，然后将结果写入"user_error_all_new.txt"

10--user_change_all_new.py

user_change_all.txt 这个文档的**处理代码好像没有**

打开"user_change_all.txt"，读取号码，换机的两个机型，日期

迭代打开"model_" + date + "_change_new.txt"

读取号码，换机的两个机型，日期

重新写入"user_change_all_new.txt"

没看懂这个程序

11- user_unchange_all.py

dict_users[user]不是为空么，怎么赋值给 model_old？

```
try:
    model_old = dict_users[user]
    #dict_users[user]不是为空么，怎么赋值给 model_old?
    if model != model_old:
        if "iPhone" in model and "iPhone" in model_old:
            dict_users[user] = max(model,model_old)
            #假如前者为 iPhone4，后者为 iPhone5，也不算换机么？
        else:
            result = user + "|" + model_old + "|" + model + "|" + date
            file_change.write(result)
            file_change.write("\n")
            del dict_users[user]
            dict_user_noneed[user] = ""
except:
    dict_users[user] = model
```

这种结构的功能还是没怎么明白，哭

重新检查为换机的文档，查出有换机的更新换机文档

11-- dict_user&app_unchangel.py

读取"apps_all.txt"新建字典，读取"user_unchange_all.txt"，获取机型放入字典

循环读取文件"app_20160" + date + "_new.txt"，将 app 四个阶段的流量写入"dict_user&app_" + date + "_unchange.txt.txt"。

异常处理部分看不懂

12-- dict_user&app_change_all.py dict_user&app_average.py

dict_app = {}；dict_user = {}；user_526 = {}；user_527 = {}；user_528 = {}；user_529 = {}；user_530 = {}；user_531 = {}；user_601 = {}；user_602 = {}；user_603 = {}；user_604 =

{}；user_605 = {}；user_606 = {}；user_607 = {}；user_608 = {}；user_609 = {}；user_610 = {}

打开"user_change_all_new.txt"，读取号码（用户），机型，时间，以及 app 流量等数据

初始化每一天的字典

新建字典：user_label = {}

统计所有的用户，app 的总流量（0526-0610），用户|机型|四个阶段的流量

13-- dict_user&app_average.py

统计换机和未换机用户平均每次所用流量（分时间段）

14-- nmf.py

非负矩阵分解算法

15 example2.py

nmf（非负矩阵分解算法的）例子

16-- dict_user&app_unchange_all.py

从 0526-0610 所有未换机的用户文件操作

建立用户字典包括统计 app 流量（四个阶段）

17-- userchange_model_eachday.py

加上换机时间？

18--user_change_usage.py

```
for date in dates:
   dict_usage[user][date] = []
```

下一句这句是什么结构？dict_usage[user]里面再嵌套一个字典还是 list？ 是字典

操作文件："app_20160" + date + "_new.txt"

提取处理每天的数据，字典类型：user(号码)：dict_usage[user][date]（嵌套字典 dict_apps[app_id]），嵌套字典：app_id：app_id：流量（用分号隔开）

将以上结果写入"user_change_usage.txt","w"

19--user_change_usage_final.py

操作文件："user_change_usage.txt","r"

统计 app 所耗总流量：app:总流量

20--user_change_app&usage_top50_final.py

操作文件："user_change_app&usage_top50.txt","r"和"apps_all.txt","r"

将所有 app 的代号和 ID 统计好写入文档"user_change_app&usage_top50_final.txt","w"：

```
result = app_no + " " + app_id
```

21-- trend_top50_increase&decrease.py

操作文件："user_change_app&usage_top50_final.txt","r"，"user_change_app&trend.txt","r"
根据流量进行降序和升序排序

22-- user_change_app_class.py

操作文件："apps_all.txt","r"和"user_change_app&usage.txt","r"
读取文件 1：生成两个字典，apps_id[no] = app_id　　apps_class[no] = app_class
读取文件 2：apps_change[app_class].append(app_id)　　apps_change_times[app_class] += app_times
将以上结果写入文档("user_change_app_class.txt","w")

23--user_change_app_usage&trend.py

文件从 0526-0610
操作文件："user_change_model_eachday_final.txt","r"和"user_change_usage_final.txt","r"
dict_changedate[user] = data[1]　用户（号码）：日期
dict_app_usage[app] += 1　　app 编号：出现次数（在换机前）
before_first[app] = flow
before_last[app] = flow
after_ave[app] += int(flow)　　累计 app 流量
并计算平均 after_ave[app] = after_ave[app]/len_after

```
dict_app_trend[app] = []
dict_app_trend[app].append(first)
dict_app_trend[app].append(last)
dict_app_trend[app].append(after)
dict_app_trend[app].append(1)
```

没明白什么意思
结果写入："user_change_app&usage_new.txt","w"和"user_change_app&trend_new.txt","w"

24--change_app_after_classes.py

```
file_1 = open("user_change_app&trend_new.txt","r")
file_result_1 = open("user_change_app_increase_after.txt","w")
file_result_2 = open("user_change_app_decrease_after.txt","w")
file_result_3 = open("user_change_app_other_after.txt","w")
```

通过读文件，分为降序和升序，和其他趋势

25--change_app_after_classes_new.py

```
file_1 = open("user_change_app_decrease_after.txt","r")
file_2 = open("user_change_app_increase_after.txt","r")
file_3 = open("apps_all.txt","r")
```

```
file_result_1 = open("user_change_app_decrease_after_new.txt","w")
file_result_2 = open("user_change_app_increase_after_new.txt","w")
```

结果写入形式：result = line + "\t" + app_id

26--app_class_increase_after.py

```
file_1 = open("user_change_app_decrease_after_new.txt","r")
file_2 = open("app_classes.txt","r")
file_result_1 = open("app_class_decrease_after.txt","w")
```

字典 1app_classes[app_no].append(app_id)

字典 2 classes_name[app_no] = app_name

结果写入形式：app_no + " " + class_name + " " + str(number) + app_id + ","

26--user_class_usage_temporal.py

```
file_1 = open("user_change_usage_temporal.txt","r")
file_result_1 = open("user_class_usage_temporal.txt","w")
```

最大项大于 0.5，则加入，最大两项大于 0.5 则加入两个，否则最大项加入其他（这一项无）

27--temporal_class_change.py

```
file_1 = open("user_class_usage_temporal.txt","r")
file_result_1 = open("temporal_class_change.txt","w")
```

统计每个时间段组合的次数

28--model_brand&year&price.py

```
file_1 = open("change_model_before.txt","r")
file_result_1 = open("model_brand.txt","w")
file_result_2 = open("model_year.txt","w")
file_result_3 = open("model_price.txt","w")
```

所读文件数据类型：Y927|vivo|4G|Android|2014.11|1400

model_brands = {}——model_brands[brand] += 1 累计机型次数

model_years = {}——model_years[year] += 1 累计年份次数

model_prices = {} 累计价格阶段次数（以 500 为一个阶段，到 2000）

29--user_change_model_final.py

```
file_1 = open("user_change_model_eachday_final.txt","r")

file_result_1 = open("user_change_model_final_2.txt","w")
file_result_2 = open("change_model_before_old.txt","w")
file_result_3 = open("change_model_after_old.txt","w")
```

所读文件数据类型：

18012778818|608|Y1||Y1||Y1|Y1|Y1|Y1|Y1|Y1||Y1||麦芒 3S|麦芒 3S|麦芒 3S

models_before = {}——

models_after = {}——

file_result_1：user + "|"+ model_before + "," + model_after 最开始和最后不同的型号

由此更新后两个文件（**不知道有什么用**）

30--user_model_in_2years.py

```
file_1 = open("change_model_before.txt","r")
file_2 = open("user_change_model_final_2.txt","r")
file_result_1 = open("model_in_2years.txt","w")
file_result_2 = open("user_model_in_2years.txt","w")
twoyears = ["2014","2015","2016"]
```

文件 1 数据类型：Q802D|中兴|4G|Android|2014.10|700

文件 2 数据类型：

18915528725|iPhone,P8

17712218081|XT1085,iPhone

15370078119|C8817E,5263

字典 1：model_2years[model] = year

将文件 2 中属于最近两年的数据写入结果 2 文件中

将机型和年份写入结果 1 文件中

31-- user_model_2year_usage.py

```
file_1 = open("user_model_in_2years.txt","r")
file_2 = open("user_change_usage_final.txt","r")
file_result_1 = open("user_model_not_2year_usage.txt","w")
```

文件 1 数据类型：17712218081|XT1085,iPhone

将所有不是最近两年的数据写入文件

32-- user_change_model_eachday_final.py

```
file_1 = open("userchange_model_eachday.txt","r")
file_2 = open("user_model_in_2years.txt","r")
file_result = open("user_change_model_eachday_final_not_2years.txt","w")
```

文件 1 数据类型：

18151968052|527|5892|5892,5263|5892|5892|5892|5892|5892|5892|5892|5892|5892|5892|5892|5892|5892|5892|5892

文件 2 数据类型：17712218081|XT1085,iPhone

这个代码功能没有看懂

33-- model_2years_app_usage&trend.py

```
file_1 = open("user_change_model_eachday_final_not_2years.txt","r")
file_2 = open("user_model_not_2year_usage.txt","r")
```

```
file_result_1 = open("model_not_2years_app&usage.txt","w")
file_result_2 = open("model_not_2years_app&trend.txt","w")
```

文件 1：

18962298796|604||iPhone4S|iPhone4S,iPhone|iPhone4S|||||VIVO|VIVO||VIVO|VIVO||

字典 1：dict_changedate[user] = data[1]

字典 2：before_first[app] = flow

字典 3：before_last[app] = flow

预测两年的趋势，**但这个趋势表达的意思还是没看懂**

34-- model_2years_app_class.py

```
file_1 = open("apps_all.txt","r")
file_2 = open("model_not_2years_app&usage.txt","r")
file_result = open("model_not_2years_app_class.txt","w")
```

文件 1：彩票宝 7　　文件 2：362　20

字典 1：apps_id[no] = app_id

字典 2：apps_class[no] = app_class

字典 3 ：apps_change[app_class].append(app_id)

字典 4：apps_change_times[app_class] += app_times

```
result = "class_" + app_class + ": " + str(times) + "|" + ",".join(apps)（即 id）
```

对 app 使用次数进行统计，将结果写入

35-- model_price_2years.py

```
file_1 = open("change_model_before.txt","r")
file_result_3 = open("model_price_2years.txt","w")
```

文件 1：XT1085|摩托罗拉|4G|Android|2015.01|1000

将价格各阶段的值写入文件，结果如下：

>2000|9

1000-1500|15

1500-2000|18

500-1000|51

<500|20

36-- user_model_2year_usage_temporal.py

```
file_1 = open("user_model_in_2years.txt","r")
file_2 = open("user_change_usage.txt","r")
file_result_1 = open("user_model_not_2year_usage_temporal.txt","w")
```

文件 1：17712218081|XT1085,iPhone

读取文件 2 中不在文件 1 里的数据写入

37-- user_change_usage_temporal.py

```
file_1 = open("user_change_model_eachday_final_not_2years.txt","r")
file_2 = open("user_model_not_2year_usage_temporal.txt","r")
file_result = open("user_change_usage_temporal_not_2years.txt","w")
file_result_2 = open("user_change_usage_sum_not_2years.txt","w")
```

文件 1：

18915528725|605|iPhone5S,iPhone|iPhone5S,iPhone|iPhone5S,iPhone|iPhone5S,iPhone|iPhone5S,iPhone|iPhone5S,iPhone|iPhone5S,iPhone|iPhone5S,iPhone|iPhone5S,iPhone|iPhone5S,iPhone|iPhone5S,iPhone|iPhone5S,iPhone|iPhone5S,iPhone|P8|P8|P8|||

字典 1：dict_changedate[user] = date_pos

每行结果为，全部+after（换机后）的数据流量和次数

另外一个结构一样，但是是总平均累加？

结果 1：

18936128790|48034898,12162555,55823875,449853549,5|0,67301728,125072361,247008628,8

结果 2：18936128790|113174975|27134368

37-- user_model&price_in_2years.py

```
file_1 = open("user_model_in_2years.txt","r")
file_2 = open("change_model_before.txt","r")
file_result = open("user_model&price_in_2years.txt","w")
```

文件 1：17712218081|XT1085,iPhone

文件 2：Q802D|中兴|4G|Android|2014.10|700

字典 1：model_price[model] = price

功能：将文件 2 中的机型价格，写入到文件 1

38-- user_change_usage_temporal_2years_final.py

```
file_1 = open("user_change_usage_temporal_2years.txt","r")
file_2 = open("user_model&price_in_2years.txt","r")
file_result = open("user_change_usage_temporal_2years_final.txt","w")
```

文件 1：

18018358235|340479,78932,8210774,5047101,3|465940,8077858,36226299,35429243,8

文件 2：17712218081|XT1085,iPhone|1000

结果：

18018358235|340479,78932,8210774,5047101,3|465940,8077858,36226299,35429243,8|700

字典 1：user_price[user] = price

将价格写入后面

39-- model_in_2years_usage_price&appclass.py

```
file_1 = open("user_change_usage_temporal_2years_final.txt","r")
file_2 = open("model_2years_app&trend.txt","r")
file_3 = open("apps_all.txt","r")
file_result_1 = open("model_in_2years_usage_price500.txt","w")
file_result_2 = open("model_in_2years_usage_price1000.txt","w")
file_result_3 = open("model_in_2years_usage_price1500.txt","w")
file_result_4 = open("model_in_2years_usage_price2000.txt","w")
file_result_5 = open("model_in_2years_usage_priceother.txt","w")
file_result_6 = open("model_in_2years_usage_price.txt","w")
file_result_7 = open("model_2years_appclass_trend.txt","w")
```

文件 1：

18018358235|340479,78932,8210774,5047101,3|465940,8077858,36226299,35429243,8|700

文件 2：199　0,0,17756

文件 3：分期乐　7

将文件 1 按价格分类写入结果 12345，并将个文件中的流量分类累计写入结果 6

字典 1：app_class = apps_class[app_no]

将三个点的流量累加，并写入结果 7