2018 年 12 月 5 日

实验二任务说明：

1. 采用 K 均值算法对给定数据集进行聚类，给出聚类结果。改变 K 的不同取值，研究 K 值改变给聚类结果所带来的变化。改变初始簇心，研究簇心变化给聚类结果所带来的变化。SPSS 和 WEKA 中均有 K-均值算法的实现，可自由选择你所熟悉的工具完成本实验。
2. 请将你的聚类结果和相应分析说明写在实验报告中（Word，格式不限），并在今天下课前发至邮箱：buptssedwkd2018@163.com

使用 python 实现聚类实验

1. 数据归一化

```
def normalizations(df):
    return df.apply(lambda x: (x - np.min(x)) / (np.max(x) - np.min(x)))
```

2. 同时发现数据中 592 行数据有缺失，将它删去
3. 使用 scikit-learn 机器学习包中的 kmeans 算法计算

```
def k_means(df):
    max = 0
    index = 0
    df = normalizations(df)
    for i in range(3, 16):
        y_pred = KMeans(n_clusters=i).fit_predict(df)
        r= metrics.calinski_harabaz_score(df, y_pred)
        print(f'k 为{i},评分为{r}')
        if r > max:
            max = r
            index = i
    print(f'最优时，k 为{index},评分为{max}')
```

4. 使用 calinski_harabaz 评价聚类结果他可以从簇内的稠密程度和簇间的离散程度来评估聚类的效果。

Calinski-Harabasz分数值s的数学计算公式是：

$$s(k) = \frac{tr(B_k)}{tr(W_k)} \frac{m-k}{k-1}$$

5. 输出结果

```
k 为 3,评分为 3473.5486295563533
k 为 4,评分为 3730.5767271906952
k 为 5,评分为 3354.9089637074076
k 为 6,评分为 3409.4564710202917
k 为 7,评分为 3393.4306292366223
k 为 8,评分为 3387.0017586198787
k 为 9,评分为 3373.187488926511
k 为 10,评分为 3360.211786616531
k 为 11,评分为 3172.4831732169378
k 为 12,评分为 3061.7701496409136
k 为 13,评分为 2962.7315137480814
k 为 14,评分为 2883.4755385386284
k 为 15,评分为 2804.833517635444
最优时，k 为 4,评分为 3730.5767271906952
```

6. 因此，当 k 为 4 时，聚类效果最好

7. 通过修改聚类初始点，观测对于 calinski_harabaz 分数的变化。

```
def change_init(df):
    y_pred = KMeans(n_clusters=4, random_state=9,init='k-means++').fit_predict(df)
    r = metrics.calinski_harabaz_score(df, y_pred)
    print(f'中心点方法：k-means++    分数：{r}')
    y_pred = KMeans(n_clusters=4, random_state=9, init='random').fit_predict(df)
    r = metrics.calinski_harabaz_score(df, y_pred)
    print(f'中心点方法：random    分数：{r}')
```

8. 输出结果

中心点方法：k-means++　　分数：4563.351498217524
中心点方法：random　　分数：2104.7693650359834

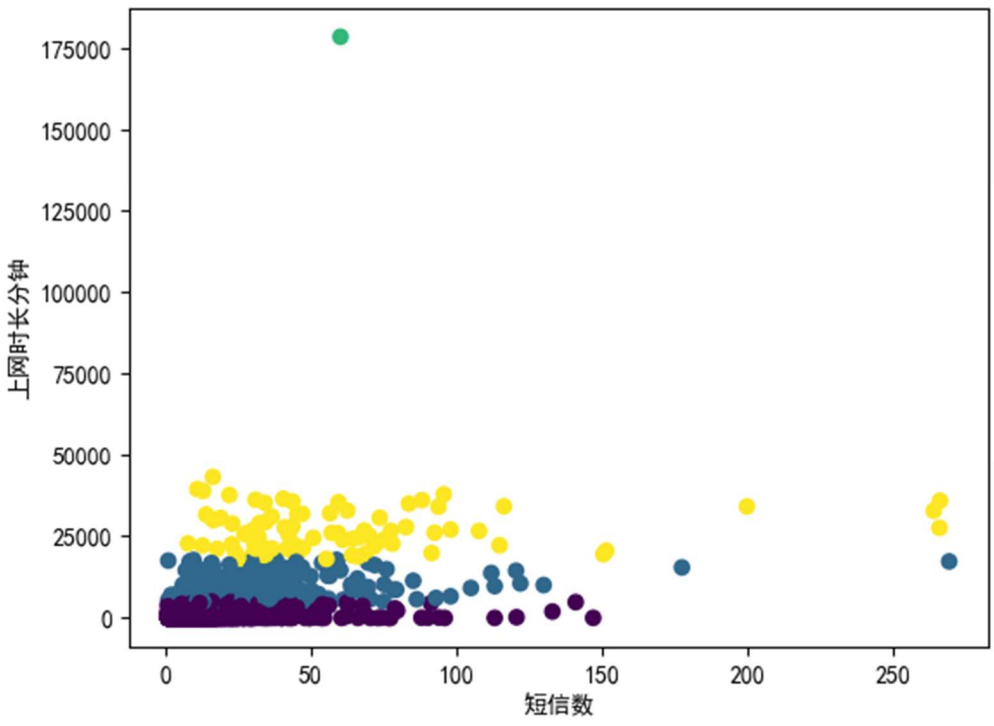
9. 观察到初始中心点选区方法不同，对于结果的影响也是巨大的。

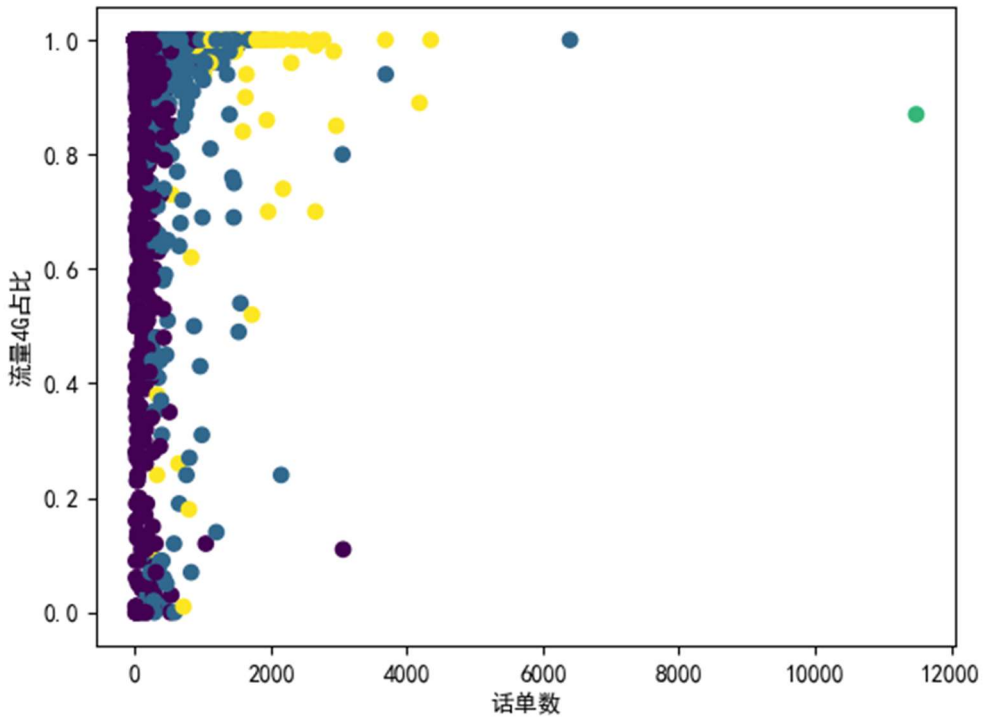
   传统的 k-means 算法随机选取中心点，而 k-means 算法对于初始点十分敏感。随机选取导致不同的结果。

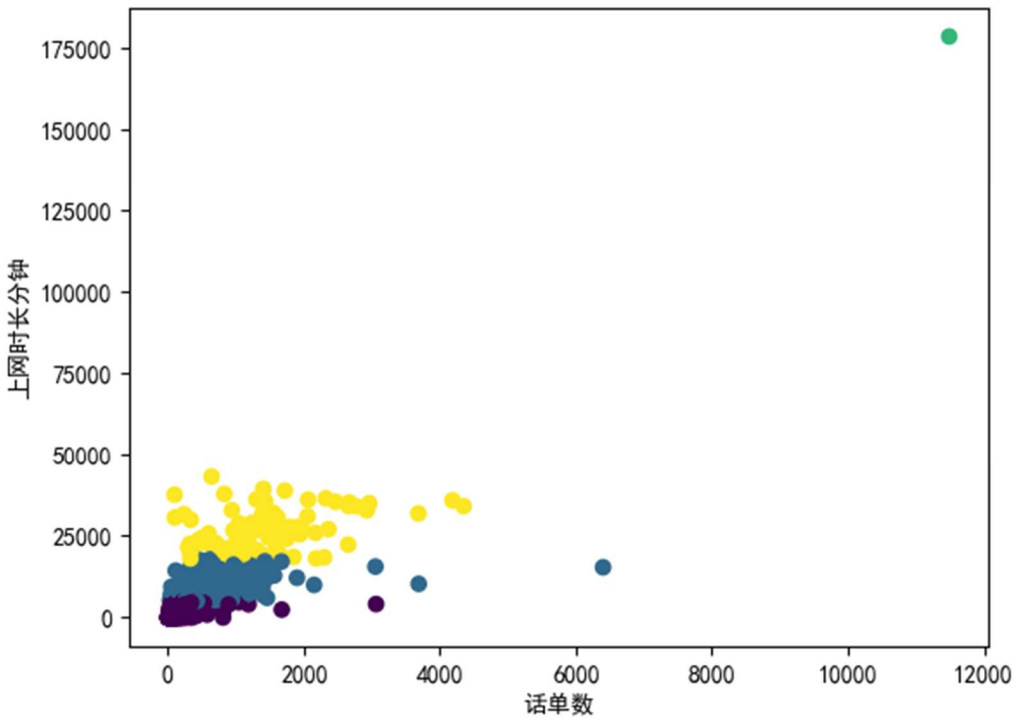
   而 K-means++算法使初始的聚类中心之间的相互距离要尽可能的远，符合直观的感觉。

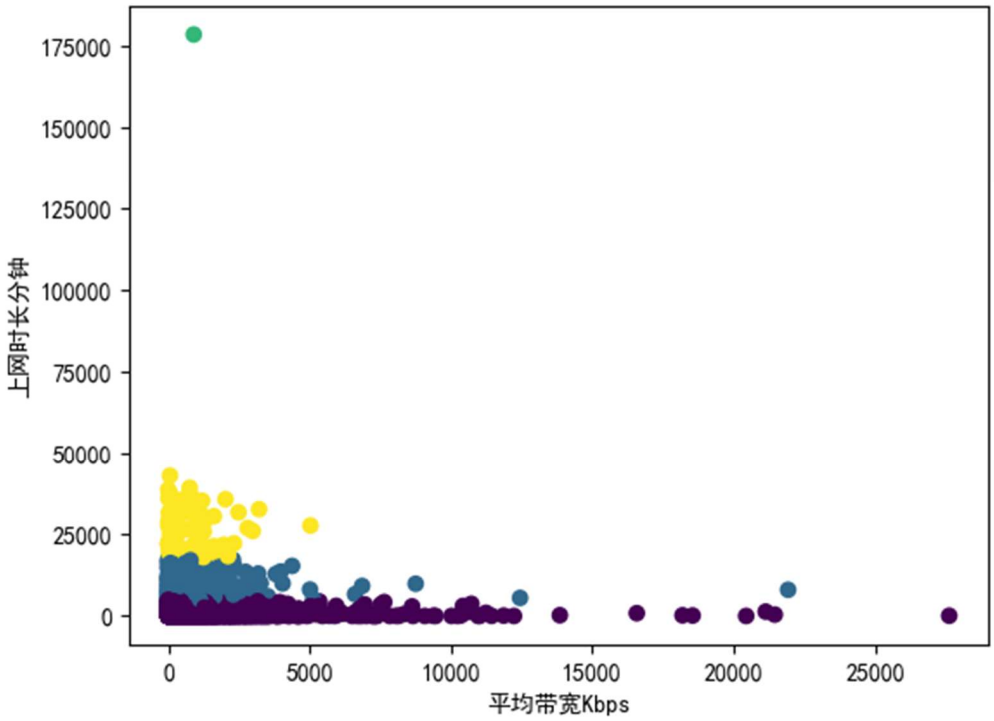
算法描述如下：
步骤一：随机选取一个样本作为第一个聚类中心 c1；
步骤二：
计算每个样本与当前已有类聚中心最短距离（即与最近一个聚类中心的距离），用 D(x)表示；
这个值越大，表示被选取作为聚类中心的概率较大；
最后，用轮盘法选出下一个聚类中心；
步骤三：重复步骤二，知道选出 k 个聚类中心。

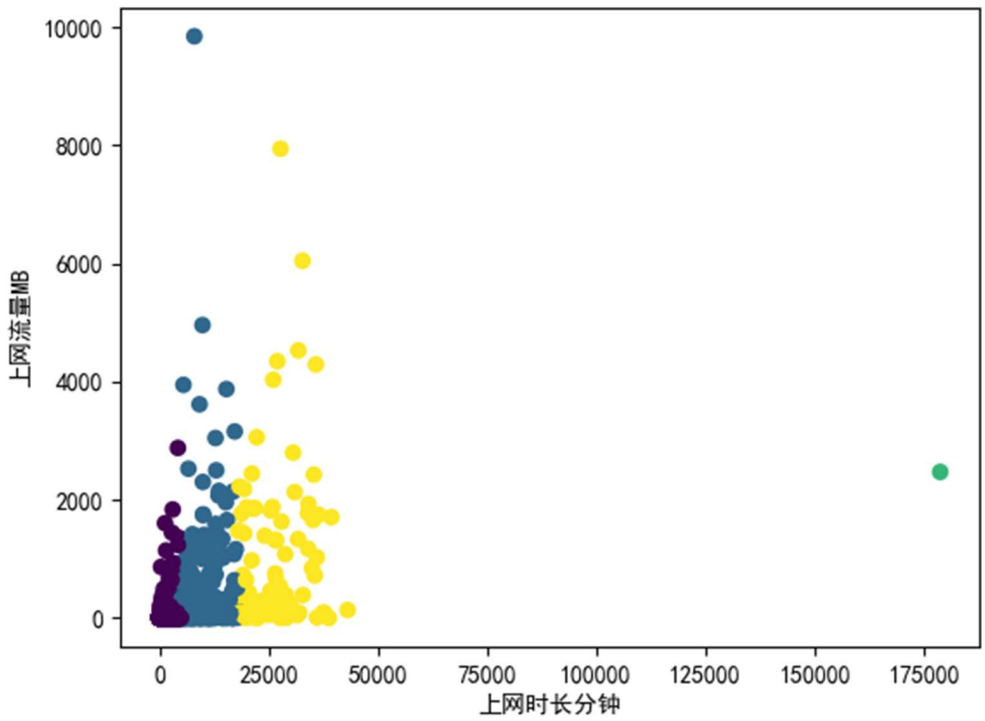
10. 根据分类结果，给出若干图像：

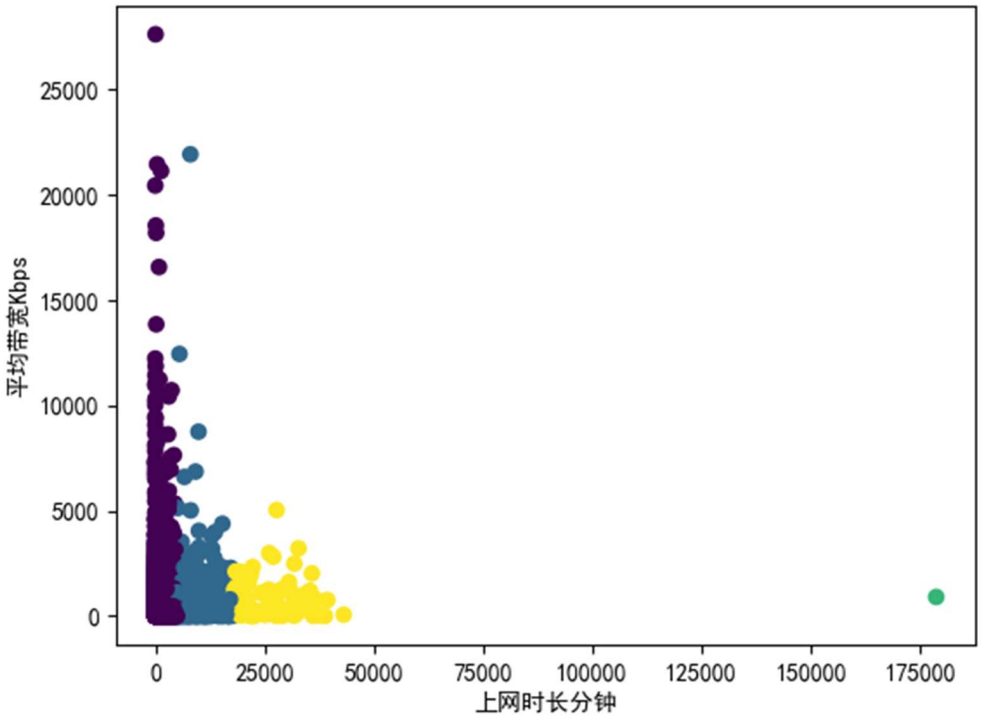


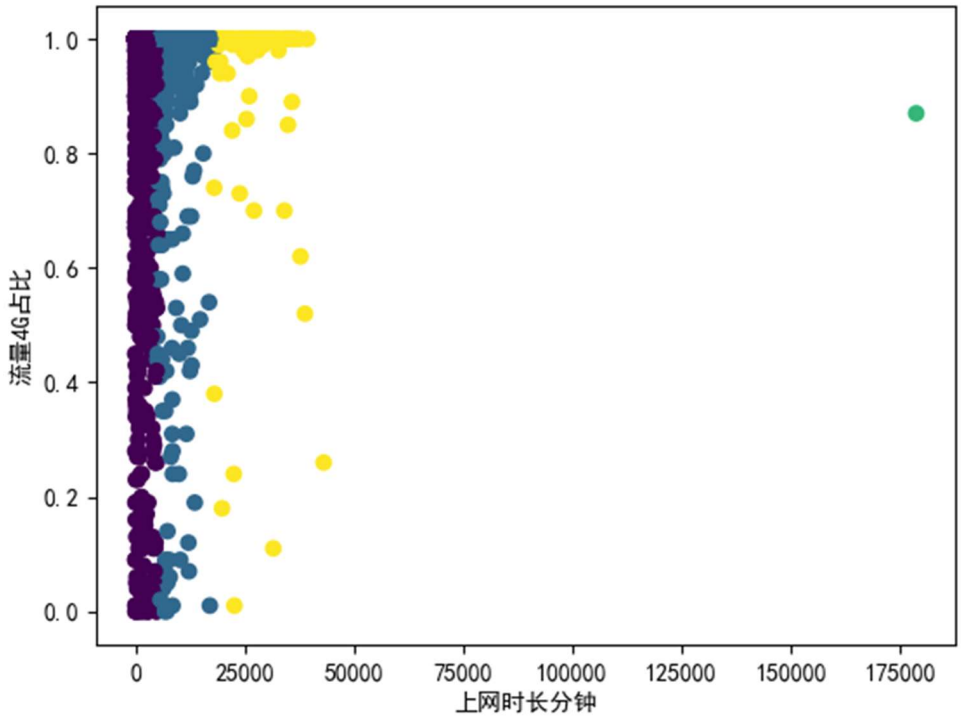

流量4G占比
1.0
0.8
0.6
0.4
0.2
0.0
0
2000
4000
6000
8000
10000
12000
话单数

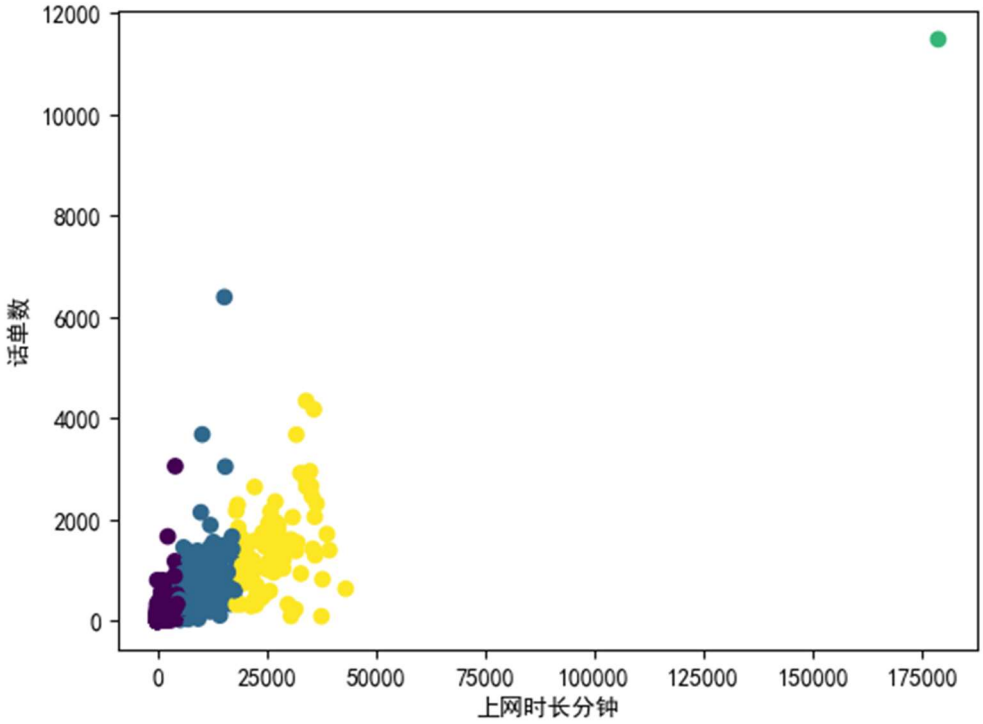

上网时长分钟
175000
150000
125000
100000
75000
50000
25000
0
0
2000
4000
6000
8000
10000
12000
话单数

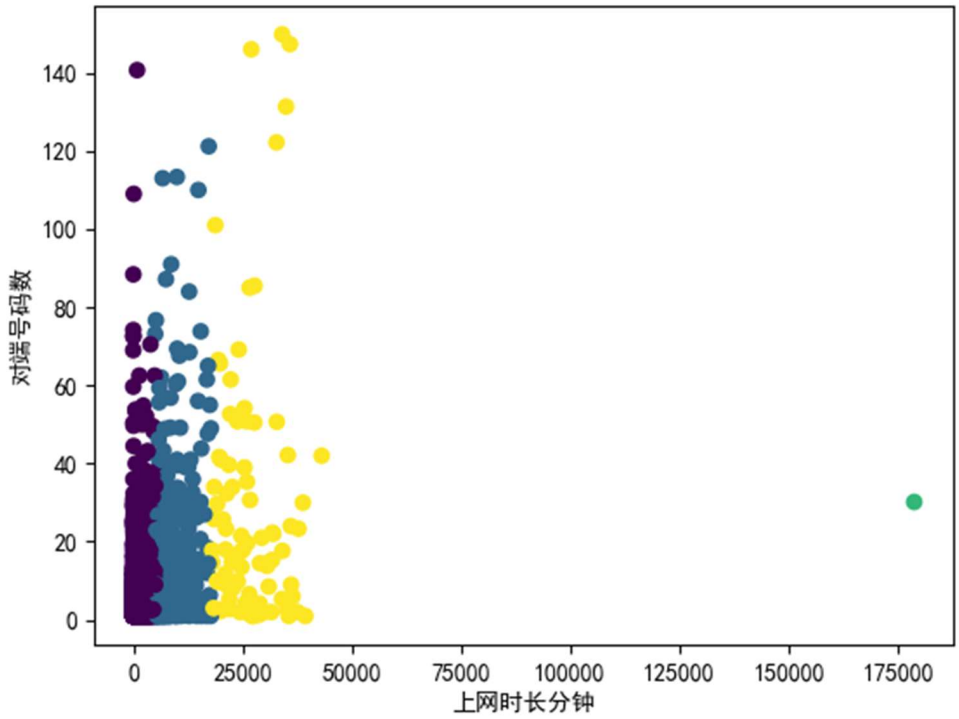

175000
150000
125000
100000
75000
50000
25000
0
上网时长分钟
0
5000
10000
15000
20000
25000
平均带宽Kbps

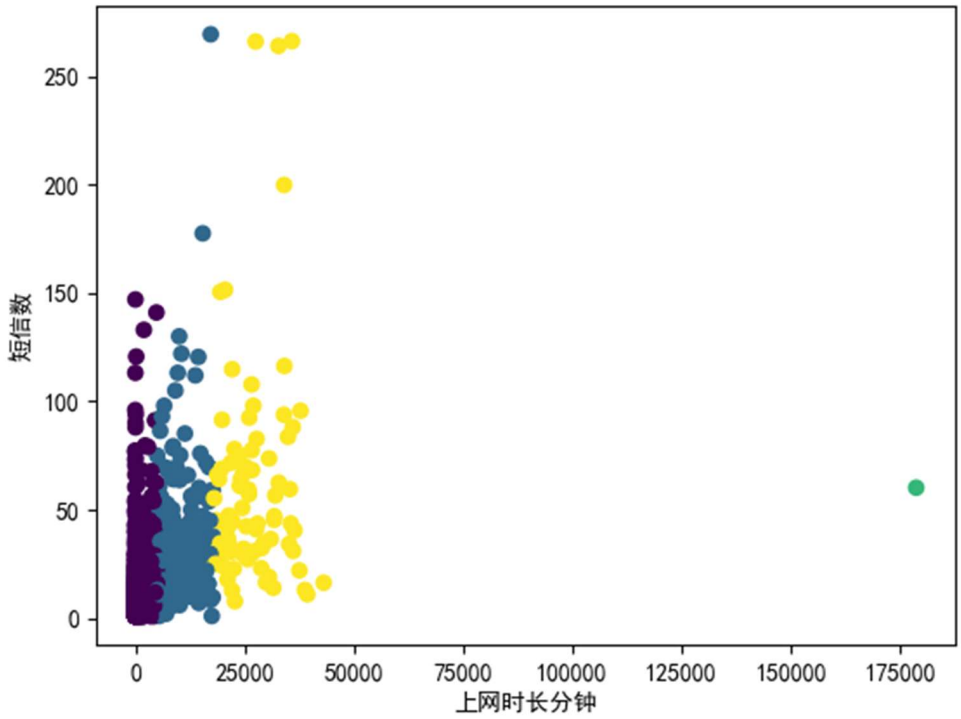

10000
8000
6000
4000
2000
0
上网流量MB
0
25000
50000
75000
100000
125000
150000
175000
上网时长分钟

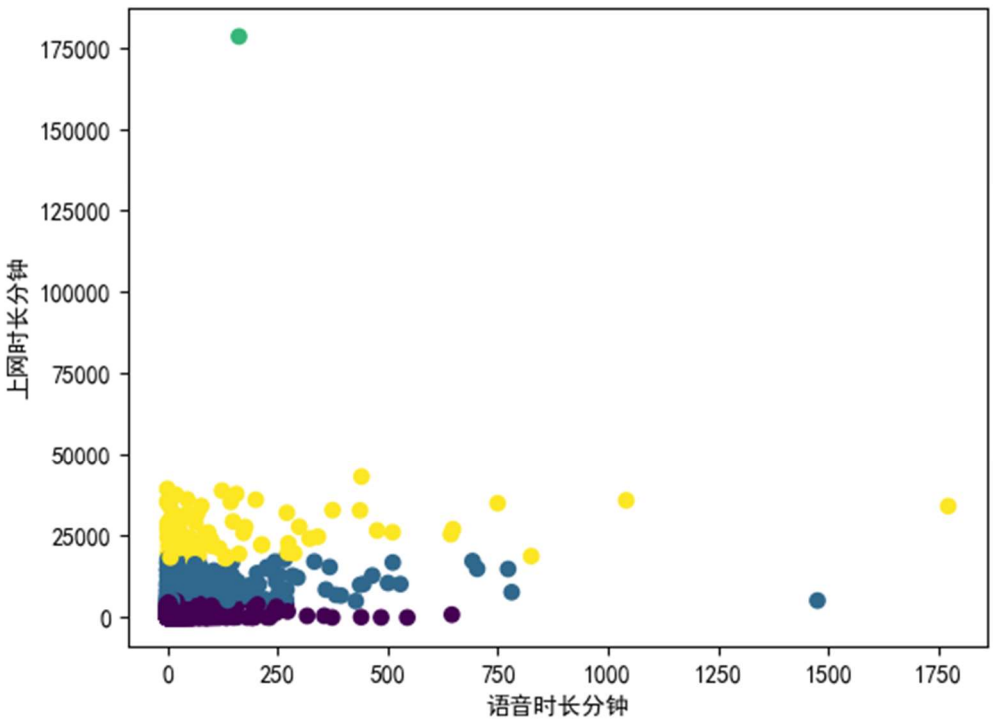

平均带宽Kbps
25000
20000
15000
10000
5000
0
0
25000
50000
75000
100000
125000
150000
175000
上网时长分钟

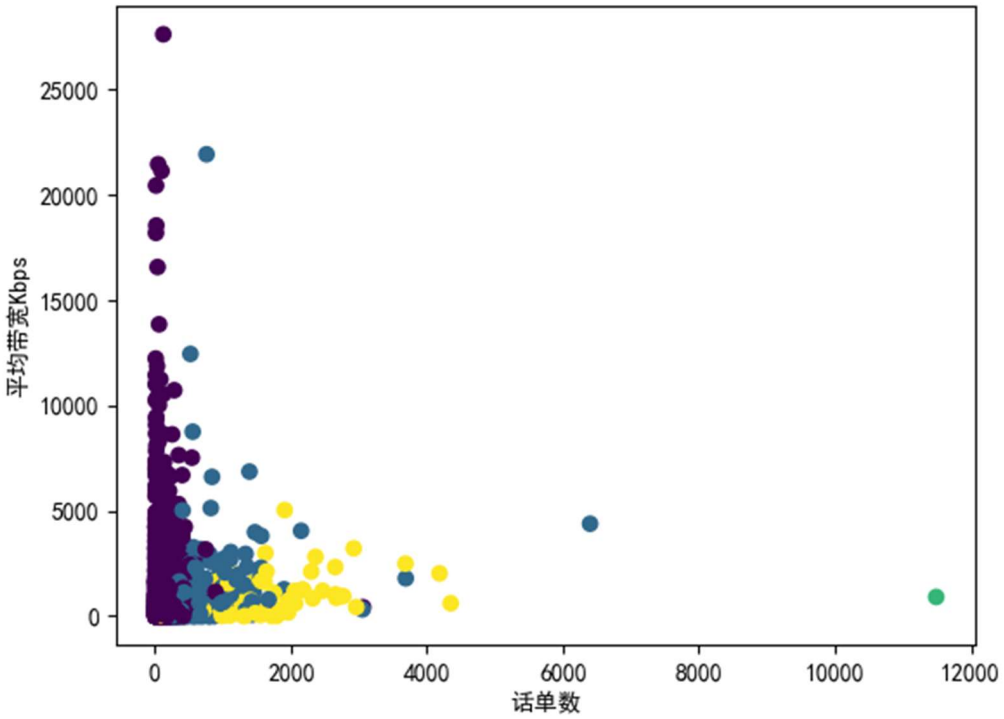

流量4G占比
1.0
0.8
0.6
0.4
0.2
0.0
0
25000
50000
75000
100000
125000
150000
175000
上网时长分钟


话单数
12000
10000
8000
6000
4000
2000
0
0
25000
50000
75000
100000
125000
150000
175000
上网时长分钟


对端号码数
140
120
100
80
60
40
20
0
0
25000
50000
75000
100000
125000
150000
175000
上网时长分钟


短信数
250
200
150
100
50
0
0
25000
50000
75000
100000
125000
150000
175000
上网时长分钟


上网时长分钟
175000
150000
125000
100000
75000
50000
25000
0
0
250
500
750
1000
1250
1500
1750
语音时长分钟


25000
20000
15000
10000
5000
0
平均带宽Kbps
0
2000
4000
6000
8000
10000
12000
话单数