山东大学 计算机科学与技术 学院

大数据分析与实践 课程实验报告

学号：	姓名：于佳杭	班级：23 数据
实验题目：		
实验学时：2	实验日期： 2025.9.13	

实验步骤与内容：

1. 库的导入与数据的读入

```
Python
import pandas as pd
from pandas importDataFrame
importnumpyas np

primitive_data=pd.read_csv("data-sample-and-filter.csv")
primitive_data
```

读入结果如下图:

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
0	47.0	71.0	通辽	一般节点	1756.0	585.0	北京	网络核心	3.677962e+10	1.000000e+11
1	47.0	74.0	通辽	一般节点	1756.0	776.0	北京	网络核心	3.660713e+10	1.000000e+11
2	47.0	240.0	通辽	一般节点	1756.0	802.0	北京	网络核心	3.603489e+10	1.000000e+11
3	47.0	241.0	通辽	一般节点	1997.0	464.0	天津	网络核心	4.233391e+10	1.000000e+11
4	47.0	242.0	通辽	一般节点	474.0	672.0	哈尔滨	一般节点	1.130008e+10	1.000000e+11
...	...	...	...	...	...	...	...	...	...	...
1142	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1143	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1144	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1145	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1146	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

1147 rows × 10 columns

可以看到数据底部有较多的空行

2. 删除多余的空行并进行过滤

采用 dropna 方法并指定参数为 any 删除多余的空行

```
Python
primitive_data_1=primitive_data.dropna(how='any')
primitive_data_1
```

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
0	47.0	71.0	通辽	一般节点	1756.0	585.0	北京	网络核心	3.677962e+10	1.000000e+11
1	47.0	74.0	通辽	一般节点	1756.0	776.0	北京	网络核心	3.660713e+10	1.000000e+11
2	47.0	240.0	通辽	一般节点	1756.0	802.0	北京	网络核心	3.603489e+10	1.000000e+11
3	47.0	241.0	通辽	一般节点	1997.0	464.0	天津	网络核心	4.233391e+10	1.000000e+11
4	47.0	242.0	通辽	一般节点	474.0	672.0	哈尔滨	一般节点	1.130008e+10	1.000000e+11
...	...	...	...	...	...	...	...	...	...	...
1113	1129.0	546.0	上海	网络核心	2050.0	502.0	石家庄	网络核心	1.350000e+11	1.000000e+11
1114	1129.0	514.0	上海	网络核心	2473.0	946.0	吉林	一般节点	2.001232e+10	1.000000e+11
1115	36036.0	499.0	长春	一般节点	1257.0	178.0	上海	网络核心	4.117194e+10	1.000000e+11
1116	36422.0	346.0	天津	网络核心	1997.0	41.0	天津	网络核心	1.604818e+10	1.000000e+11
1117	2701.0	619.0	大连	网络核心	2549.0	1070.0	沈阳	网络核心	1.470004e+10	1.000000e+11

1118 rows × 10 columns

接下来过滤得到 traffic 不等于 0 且 from_level=一般节点的数据

```
Python
data_before_filter=primitive_data_1
data_after_filter_1=data_before_filter.loc[data_before_filter["traffic"]!=0]
data_after_filter_2=data_after_filter_1.loc[data_after_filter_1["from_level"]=='一般节点']
data_after_filter_2
```

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
0	47.0	71.0	通辽	一般节点	1756.0	585.0	北京	网络核心	3.677962e+10	1.000000e+11
1	47.0	74.0	通辽	一般节点	1756.0	776.0	北京	网络核心	3.660713e+10	1.000000e+11
2	47.0	240.0	通辽	一般节点	1756.0	802.0	北京	网络核心	3.603489e+10	1.000000e+11
3	47.0	241.0	通辽	一般节点	1997.0	464.0	天津	网络核心	4.233391e+10	1.000000e+11
4	47.0	242.0	通辽	一般节点	474.0	672.0	哈尔滨	一般节点	1.130008e+10	1.000000e+11
...	...	...	...	...	...	...	...	...	...	...
1097	2473.0	1460.0	吉林	一般节点	591.0	586.0	绥化	一般节点	9.165302e+10	1.000000e+11
1103	36036.0	18.0	长春	一般节点	3443.0	650.0	青岛	网络核心	4.350363e+10	1.000000e+11
1104	63.0	6.0	通辽	一般节点	36036.0	20.0	长春	一般节点	1.871659e+10	1.000000e+11
1107	36036.0	52.0	长春	一般节点	1129.0	171.0	上海	网络核心	2.760267e+10	1.000000e+11
1115	36036.0	499.0	长春	一般节点	1257.0	178.0	上海	网络核心	4.117194e+10	1.000000e+11

554 rows × 10 columns

3. 对数据进行抽样

采取不同的采样方式采取 50 个样本并比较采样结果

- 加权采样：to_level 的值为一般节点与网络核心的权重之比为 1 : 5

```
Python
data_before_sample=data_after_filter_2
columns=data_before_sample.columns
weight_sample=data_before_sample.copy()
weight_sample['weight']=0
foriinweight_sample.index:
if weight_sample.at[i,'to_level']=='一般节点':
weight=1
else:
```

```
weight=5
weight_sample.at[i,'weight']=weight

weight_sample_finish=weight_sample.sample(n=50,weights='weight')
#data_before_sample=data_before_sample[columns]
weight_sample_finish=weight_sample[columns]
weight_sample_finish
```

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
674	591.0	586.0	绥化	一般节点	47.0	243.0	通辽	一般节点	2.310000e+11	1.000000e+11
51	96.0	156.0	呼和浩特	一般节点	3227.0	103.0	济南	网络核心	3.504423e+10	1.000000e+11
16	47.0	427.0	通辽	一般节点	1997.0	213.0	天津	网络核心	4.349038e+10	1.000000e+11
309	96.0	99.0	呼和浩特	一般节点	2360.0	76.0	太原	网络核心	1.810000e+11	1.000000e+11
587	96.0	141.0	呼和浩特	一般节点	3213.0	246.0	重庆	网络核心	9.794152e+10	1.000000e+11
277	47.0	240.0	通辽	一般节点	3213.0	246.0	重庆	网络核心	9.794152e+10	1.000000e+11
365	180.0	260.0	呼和浩特	一般节点	1756.0	788.0	北京	网络核心	1.280000e+11	1.000000e+11
660	63.0	224.0	通辽	一般节点	2701.0	71.0	大连	网络核心	9.786992e+09	1.000000e+11
286	47.0	259.0	通辽	一般节点	4561.0	1087.0	成都	网络核心	1.140000e+11	1.000000e+11
349	180.0	52.0	呼和浩特	一般节点	3227.0	449.0	济南	网络核心	6.987232e+10	1.000000e+11
44	96.0	127.0	呼和浩特	一般节点	1756.0	1027.0	北京	网络核心	8.917187e+10	1.000000e+11
494	47.0	252.0	通辽	一般节点	1536.0	86.0	鄂尔多斯	网络核心	4.103025e+10	1.000000e+11
452	787.0	325.0	玉溪	一般节点	2701.0	181.0	大连	网络核心	9.501373e+09	1.000000e+11
1107	36036.0	52.0	长春	一般节点	1129.0	171.0	上海	网络核心	2.760267e+10	1.000000e+11
172	787.0	63.0	玉溪	一般节点	1536.0	1882.0	广州	网络核心	1.040000e+11	1.000000e+11

- 随机抽样

```
Python
random_sample=data_before_sample
random_sample_finish=random_sample.sample(n=50)
random_sample_finish=random_sample_finish[columns]
random_sample_finish
```

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
148	591.0	558.0	绥化	一般节点	36036.0	499.0	长春	一般节点	6.475345e+10	1.000000e+11
533	47.0	252.0	通辽	一般节点	1536.0	585.0	广州	网络核心	6.885369e+10	1.000000e+11
57	96.0	379.0	呼和浩特	一般节点	1756.0	1187.0	北京	网络核心	9.186782e+10	1.000000e+11
346	180.0	38.0	呼和浩特	一般节点	2549.0	1487.0	沈阳	网络核心	1.010000e+11	1.000000e+11
172	787.0	63.0	玉溪	一般节点	1536.0	1882.0	广州	网络核心	1.040000e+11	1.000000e+11
127	474.0	1399.0	哈尔滨	一般节点	4360.0	468.0	南京	一般节点	2.294405e+10	1.000000e+11
497	47.0	260.0	通辽	一般节点	36422.0	350.0	天津	网络核心	4.117194e+10	1.000000e+11
412	591.0	23.0	绥化	一般节点	2701.0	71.0	大连	网络核心	9.786992e+09	1.000000e+11
414	591.0	29.0	绥化	一般节点	235.0	1649.0	北京	网络核心	4.744260e+10	1.000000e+11
281	47.0	249.0	通辽	一般节点	1536.0	1882.0	广州	网络核心	1.040000e+11	1.000000e+11
775	96.0	134.0	呼和浩特	一般节点	180.0	98.0	呼和浩特	一般节点	4.473394e+10	1.000000e+11
313	96.0	111.0	呼和浩特	一般节点	2360.0	197.0	太原	网络核心	1.800000e+11	1.000000e+11
161	591.0	1266.0	绥化	一般节点	235.0	1950.0	北京	网络核心	5.840438e+10	1.000000e+11
7	47.0	250.0	通辽	一般节点	2473.0	762.0	吉林	一般节点	7.720147e+09	1.000000e+11

- 分层抽样：根据 to_level 的值进行分层采样

根据比例一般节点抽 17 个，网络核心抽 33 个

```
Python
ybjd=data_before_sample.loc[data_before_sample['to_level']=='一般节点']
wlhx=data_before_sample.loc[data_before_sample['to_level']=='网络核心']
after_sample=pd.concat([ybjd.sample(17),wlhx.sample(33)])
after_sample
```

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
376	474.0	460.0	哈尔滨	一般节点	3757.0	122.0	福州	一般节点	2.063768e+10	1.000000e+11
334	96.0	391.0	呼和浩特	一般节点	96.0	120.0	呼和浩特	一般节点	9.099477e+10	1.000000e+11
25	63.0	70.0	通辽	一般节点	180.0	264.0	呼和浩特	一般节点	2.318961e+10	1.000000e+11
347	180.0	42.0	呼和浩特	一般节点	4360.0	406.0	南京	一般节点	1.680000e+11	1.000000e+11
416	591.0	60.0	绥化	一般节点	180.0	52.0	呼和浩特	一般节点	2.274215e+10	1.000000e+11
764	2473.0	941.0	吉林	一般节点	180.0	26.0	呼和浩特	一般节点	4.103025e+10	1.000000e+11
408	591.0	13.0	绥化	一般节点	180.0	264.0	呼和浩特	一般节点	2.318961e+10	1.000000e+11
812	180.0	52.0	呼和浩特	一般节点	474.0	682.0	哈尔滨	一般节点	2.300000e+11	1.000000e+11
604	96.0	134.0	呼和浩特	一般节点	2473.0	1460.0	吉林	一般节点	3.736624e+10	1.000000e+11
674	591.0	586.0	绥化	一般节点	47.0	243.0	通辽	一般节点	2.310000e+11	1.000000e+11
129	474.0	1410.0	哈尔滨	一般节点	4069.0	1205.0	宁波	一般节点	2.961526e+10	1.000000e+11
379	474.0	473.0	哈尔滨	一般节点	474.0	1374.0	哈尔滨	一般节点	4.590714e+10	1.000000e+11
555	63.0	278.0	通辽	一般节点	36036.0	18.0	长春	一般节点	6.473446e+10	1.000000e+11
759	3757.0	122.0	福州	一般节点	96.0	407.0	呼和浩特	一般节点	2.760267e+10	1.000000e+11
804	180.0	264.0	呼和浩特	一般节点	474.0	475.0	哈尔滨	一般节点	9.186782e+10	1.000000e+11
780	96.0	391.0	呼和浩特	一般节点	180.0	205.0	呼和浩特	一般节点	1.909842e+10	1.000000e+11
39	96.0	114.0	呼和浩特	一般节点	2473.0	769.0	吉林	一般节点	1.162506e+10	1.000000e+11
304	63.0	230.0	通辽	一般节点	3227.0	77.0	济南	网络核心	9.868956e+10	1.000000e+11
1059	47.0	252.0	通辽	一般节点	1997.0	250.0	天津	网络核心	1.470004e+10	1.000000e+11
159	591.0	1250.0	绥化	一般节点	235.0	1749.0	北京	网络核心	5.757832e+10	1.000000e+11

- 还可以自行实现系统抽样，整群抽样等方法

整群抽样:

[11]:
```
def cluster_sampling(df, cluster_col, num_clusters):
    """
    对DataFrame进行整群抽样
    :param df: 数据框
    :param cluster_col: 用于分群的列名
    :param num_clusters: 要抽取的群数量
    :return: 抽样结果
    """
    # 获取所有唯一的群标签
    clusters = df[cluster_col].unique()

    if num_clusters > len(clusters):
        raise ValueError("抽取的群数量不能超过总群数")

    # 随机选择指定数量的群
    chosen_clusters = np.random.choice(clusters, size=num_clusters, replace=False)

    # 抽取所有属于这些群的样本
    return df[df[cluster_col].isin(chosen_clusters)]

# 执行整群抽样，随机抽取3个城市的全部数据
cluster_sample_finish = cluster_sampling(data_before_sample, 'from_city', 3)
cluster_sample_finish
```

[11]:

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
0	47	71	通辽	一般节点	1756	585	北京	网络核心	49636052613	1.000000e+11
1	47	74	通辽	一般节点	1756	776	北京	网络核心	50056871412	1.000000e+11
2	47	240	通辽	一般节点	1756	802	北京	网络核心	49453581081	1.000000e+11
3	47	241	通辽	一般节点	1997	464	天津	网络核心	49733361585	1.000000e+11
4	47	242	通辽	一般节点	474	672	哈尔滨	一般节点	50492573662	1.000000e+11
5	47	243	通辽	一般节点	96	124	呼和浩特	一般节点	49942713747	1.000000e+11
6	47	249	通辽	一般节点	1997	85	天津	网络核心	50499586948	1.000000e+11
7	47	250	通辽	一般节点	2473	762	吉林	一般节点	49108721007	1.000000e+11
8	47	251	通辽	一般节点	2549	839	沈阳	网络核心	50755299504	1.000000e+11
9	47	252	通辽	一般节点	96	134	呼和浩特	一般节点	50256475808	1.000000e+11
10	47	258	通辽	一般节点	1997	122	天津	网络核心	49594312223	1.000000e+11

系统抽样：

[10]:
```
def systematic_sampling(df, n):
    """
    对DataFrame进行系统抽样
    :param df: 数据框
    :param n: 抽样数量
    :return: 抽样结果
    """
    total_rows = len(df)
    # 计算抽样间隔 k
    k = total_rows // n
    if k == 0:
        raise ValueError("样本量过大，无法进行系统抽样")

    # 从 0 到 k-1 中随机选择一个起始点
    start_index = np.random.randint(0, k)

    # 生成抽样索引
    indices = np.arange(start_index, total_rows, k)

    return df.iloc[indices]

# 执行系统抽样
systematic_sample_finish = systematic_sampling(data_before_sample.reset_index(drop=True), 50)
systematic_sample_finish
```

[10]:

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
4	47	242	通辽	一般节点	474	672	哈尔滨	一般节点	50492573662	1.000000e+11
15	47	425	通辽	一般节点	1756	1018	北京	网络核心	50796899329	1.000000e+11
26	63	74	通辽	一般节点	2701	181	大连	网络核心	50364636480	1.000000e+11
37	96	108	呼和浩特	一般节点	2360	236	太原	网络核心	48210462086	1.000000e+11
48	96	141	呼和浩特	一般节点	474	422	哈尔滨	一般节点	49429192047	1.000000e+11
59	96	391	呼和浩特	一般节点	47	417	通辽	一般节点	51570663870	1.000000e+11
70	180	36	呼和浩特	一般节点	2194	406	唐山	网络核心	50973267302	1.000000e+11
81	180	202	呼和浩特	一般节点	36272	247	太原	网络核心	49867223584	1.000000e+11
92	180	272	呼和浩特	一般节点	3443	316	青岛	网络核心	52854391127	1.000000e+11
103	474	614	哈尔滨	一般节点	3227	724	济南	网络核心	51504522549	1.000000e+11
114	474	1238	哈尔滨	一般节点	1756	1008	北京	网络核心	51270474683	1.000000e+11
125	474	1410	哈尔滨	一般节点	4069	1205	宁波	一般节点	46523775334	1.000000e+11
136	591	56	绥化	一般节点	36036	52	长春	一般节点	48627355195	1.000000e+11
147	591	586	绥化	一般节点	180	192	呼和浩特	一般节点	49061517661	1.000000e+11
158	591	1290	绥化	一般节点	2194	180	唐山	网络核心	49758461056	1.000000e+11
169	787	324	玉溪	一般节点	1536	1941	广州	网络核心	48712502205	1.000000e+11

结论分析与体会：

本次实验让我深刻体会到，在数据预处理的基础上，必须根据具体的分析目标来选择最恰当的抽样方法，才能确保分析结果的有效性和针对性。