

山东大学计算机科学与技术学院

大数据分析实践 课程实验报告

学号：202300130086	姓名：张珈恺	班级：23 数据
实验题目：数据采样方法实践		
实验学时：2	实验日期：2025/9/19	
实验目标： 利用 Pandas 库实现多种数据采样和过滤的方法		
实验环境： Python3.9, Jupyter notebook		

实验步骤与内容：

1 、库的导入与数据的读入

```
import pandas as pd
from pandas import DataFrame
import numpy as np

primitive_data = pd.read_csv("C:\\SDUicloudCache\\张珈恺\\大数据分析实践\\data.csv", encoding='gbk')
primitive_data
```

	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
...
1113	1129	546	上海	网络核心	2050	502	石家庄	网络核心	48731433404	1.000000e+11
1114	1129	514	上海	网络核心	2473	946	吉林	一般节点	50060666120	1.000000e+11
1115	36036	499	长春	一般节点	1257	178	上海	网络核心	50545082113	1.000000e+11
1116	36422	346	天津	网络核心	1997	41	天津	网络核心	50628787089	1.000000e+11
1117	2701	619	大连	网络核心	2549	1070	沈阳	网络核心	48753971761	1.000000e+11

1118 rows x 10 columns

2 、删除多余的空行并进行过滤

使用 dropna 方法并指定参数为'any' 删除多余的空行

```
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	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
...
1113	1129	546	上海	网络核心	2050	502	石家庄	网络核心	48731433404	1.000000e+11
1114	1129	514	上海	网络核心	2473	946	吉林	一般节点	50060666120	1.000000e+11
1115	36036	499	长春	一般节点	1257	178	上海	网络核心	50545082113	1.000000e+11
1116	36422	346	天津	网络核心	1997	41	天津	网络核心	50628787089	1.000000e+11
1117	2701	619	大连	网络核心	2549	1070	沈阳	网络核心	48753971761	1.000000e+11

1118 rows x 10 columns

使用loc 方法过滤得到'traffic'不等于0 且'from_level'='一般节点'的数据

```
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
```

✓ 0.0s

	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
...
1097	2473	1460	吉林	一般节点	591	586	绥化	一般节点	48409925693	1.000000e+11
1103	36036	18	长春	一般节点	3443	650	青岛	网络核心	48663350759	1.000000e+11
1104	63	6	通辽	一般节点	36036	20	长春	一般节点	50355678076	1.000000e+11
1107	36036	52	长春	一般节点	1129	171	上海	网络核心	49345226162	1.000000e+11
1115	36036	499	长春	一般节点	1257	178	上海	网络核心	50545082113	1.000000e+11

550 rows × 10 columns

3、对数据进行抽样

使用 sample 方法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数据采样权重之比为 1 : 5

```
data_before_sample=data_after_filter_2
columns=data_before_sample.columns
weight_sample=data_before_sample.copy()
weight_sample['weight']=0
for i in weight_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')
weight_sample_finish=weight_sample_finish[columns]
weight_sample_finish
```

✓ 0.0s

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
142	591	64	绥化	一般节点	36272	105	太原	网络核心	51256753219	1.000000e+11
321	96	135	呼和浩特	一般节点	2050	553	石家庄	网络核心	51921872375	1.000000e+11
56	96	346	呼和浩特	一般节点	1257	138	上海	网络核心	47759033178	1.000000e+11
147	591	526	绥化	一般节点	1129	514	上海	网络核心	49318922185	1.000000e+11
91	180	264	呼和浩特	一般节点	63	70	通辽	一般节点	50106121660	1.000000e+11
8	47	251	通辽	一般节点	2549	839	沈阳	网络核心	50755299504	1.000000e+11
22	63	60	通辽	一般节点	36422	258	天津	网络核心	49920786706	1.000000e+11
492	47	250	通辽	一般节点	4515	652	西安	网络核心	49014089485	1.000000e+11
929	4360	468	南京	一般节点	1997	464	天津	网络核心	49145116989	1.000000e+11
305	63	232	通辽	一般节点	2549	1066	沈阳	网络核心	49269663214	1.000000e+11
54	96	159	呼和浩特	一般节点	2360	266	太原	网络核心	51625089370	1.000000e+11
717	2473	1043	吉林	一般节点	36422	324	天津	网络核心	50594312992	1.000000e+11
443	787	52	玉溪	一般节点	2360	215	太原	网络核心	49322809158	1.000000e+11
18	63	10	通辽	一般节点	235	106	北京	网络核心	52195591947	1.000000e+11
847	47	252	通辽	一般节点	591	560	绥化	一般节点	51065218921	1.000000e+11
131	474	1473	哈尔滨	一般节点	2549	1461	沈阳	网络核心	53304989080	1.000000e+11

随机采样

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

✓ 0.0s

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
80	180	200	呼和浩特	一般节点	2701	300	大连	网络核心	51884294458	1.000000e+11
436	591	1266	绥化	一般节点	2050	505	石家庄	网络核心	51285397493	1.000000e+11
1039	180	264	呼和浩特	一般节点	36036	54	长春	一般节点	49124032697	1.000000e+11
75	180	84	呼和浩特	一般节点	1536	86	鄂尔多斯	网络核心	49100967003	1.000000e+11
131	474	1473	哈尔滨	一般节点	2549	1461	沈阳	网络核心	53304989080	1.000000e+11
40	96	117	呼和浩特	一般节点	2050	505	石家庄	网络核心	48814619370	1.000000e+11
780	96	391	呼和浩特	一般节点	180	205	呼和浩特	一般节点	50103206178	1.000000e+11
150	591	582	绥化	一般节点	2701	619	大连	网络核心	50838395442	1.000000e+11
276	47	74	通辽	一般节点	4561	1033	成都	网络核心	50819524115	1.000000e+11
824	47	252	通辽	一般节点	474	1389	哈尔滨	一般节点	51228405663	1.000000e+11
47	96	136	呼和浩特	一般节点	2360	215	太原	网络核心	49292630301	1.000000e+11
363	180	254	呼和浩特	一般节点	2360	62	太原	网络核心	50252917820	1.000000e+11
372	474	416	哈尔滨	一般节点	3227	512	济南	网络核心	49544939922	1.000000e+11
309	96	99	呼和浩特	一般节点	2360	76	太原	网络核心	49047882786	1.000000e+11
932	2473	1460	吉林	一般节点	1997	467	天津	网络核心	50151515116	1.000000e+11
57	96	379	呼和浩特	一般节点	1756	1187	北京	网络核心	49400869697	1.000000e+11
110	474	672	哈尔滨	一般节点	47	242	通辽	一般节点	51555817613	1.000000e+11
448	787	307	玉溪	一般节点	36422	258	天津	网络核心	51727332383	1.000000e+11

分层采样：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

```
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
```

✓ 0.0s

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
491	47	249	通辽	一般节点	36539	1140	杭州	一般节点	50888438116	1.000000e+11
388	474	678	哈尔滨	一般节点	96	124	呼和浩特	一般节点	49289354051	1.000000e+11
308	63	286	通辽	一般节点	47	258	通辽	一般节点	50067368970	1.000000e+11
122	474	1272	哈尔滨	一般节点	2473	1043	吉林	一般节点	49735704801	1.000000e+11
21	63	58	通辽	一般节点	36036	54	长春	一般节点	48363382095	1.000000e+11
546	63	60	通辽	一般节点	4360	468	南京	一般节点	47970715088	1.000000e+11
762	474	1374	哈尔滨	一般节点	180	18	呼和浩特	一般节点	48043608658	1.000000e+11
743	4069	1195	宁波	一般节点	96	134	呼和浩特	一般节点	50099141709	1.000000e+11
787	36036	54	长春	一般节点	180	256	呼和浩特	一般节点	51915256521	1.000000e+11
783	591	23	绥化	一般节点	180	218	呼和浩特	一般节点	50522034278	1.000000e+11
310	96	102	呼和浩特	一般节点	474	678	哈尔滨	一般节点	49006847943	1.000000e+11
294	63	12	通辽	一般节点	474	417	哈尔滨	一般节点	48667628783	1.000000e+11
779	96	152	呼和浩特	一般节点	180	202	呼和浩特	一般节点	51162997127	1.000000e+11
383	474	670	哈尔滨	一般节点	5058	144	南宁	一般节点	50998204735	1.000000e+11
530	47	249	通辽	一般节点	2473	799	吉林	一般节点	49803820036	1.000000e+11
812	180	52	呼和浩特	一般节点	474	682	哈尔滨	一般节点	50713290427	1.000000e+11
160	591	1258	绥化	一般节点	4448	127	无锡	一般节点	50322958171	1.000000e+11
1086	36539	1140	杭州	一般节点	235	1661	北京	网络核心	51411580502	1.000000e+11
452	787	325	玉溪	一般节点	2701	181	大连	网络核心	49891276242	1.000000e+11
168	787	52	玉溪	一般节点	3213	246	重庆	网络核心	50468642387	1.000000e+11

结论与体会：

本次实验成功用 Pandas 实现数据清洗与三种采样。删除空行并筛选出有效数据后，加权、随机、分层采样各有特点：加权突出网络核心样本，分层按比例保留类别特征，随机则无偏向性。实验让我掌握了数据预处理关键步骤，也认识到需依需求选合适采样方法，为后续分析打牢基础。