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：202300300198	姓名：朱宣玮	班级：数据 23
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：

Python 3.11

Jupyter Notebook

数据集：

数据集地址：http://storage.amesholland.xyz/data.csv

实验步骤与结果：

1. 库的导入与数据的读入

```python
# 1. 库的导入与数据的读入
import pandas as pd
import numpy as np

primitive_data = pd.read_csv("data.csv", encoding='gbk')
primitive_data
```

[1] ✓ 0.2s Python

	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 × 10 columns

2. 删除多余的空行并进行过滤

```python
# 2. 删除多余的空行并进行过滤

# 删除包含空值的行
primitive_data_1 = primitive_data.dropna(how='any')
primitive_data_1
```

[2] ✓ 0.0s Python

	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 × 10 columns

```
# 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
```

[3] ✓ 0.0s Python

	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

采取不同的采样方式采取 50 个样本并比较采样结果

（1）加权采样

```
# 3. 对数据进行抽样
# 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

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', random_state=42)
weight_sample_finish = weight_sample_finish[columns]
weight_sample_finish
```

[4] ✓ 0.0s Python

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
292	63	6	通辽	一般节点	2841	418	郑州	网络核心	51392218854	1.000000e+11
1005	36036	499	长春	一般节点	2050	502	石家庄	网络核心	49116324777	1.000000e+11
534	47	258	通辽	一般节点	1997	84	天津	网络核心	50060087433	1.000000e+11
411	591	19	绥化	一般节点	235	1506	北京	网络核心	50171685281	1.000000e+11
80	180	200	呼和浩特	一般节点	2701	300	大连	网络核心	51884294458	1.000000e+11
32	63	282	通辽	一般节点	36422	230	天津	网络核心	49455678350	1.000000e+11
706	2473	799	吉林	一般节点	1536	86	鄂尔多斯	网络核心	49550894885	1.000000e+11
412	591	23	绥化	一般节点	2701	71	大连	网络核心	50009822342	1.000000e+11
495	47	258	通辽	一般节点	235	1958	北京	网络核心	48574009525	1.000000e+11
11	47	259	通辽	一般节点	1756	245	北京	网络核心	50703793815	1.000000e+11
1041	180	20	呼和浩特	一般节点	36422	394	天津	网络核心	50353235399	1.000000e+11

（2）随机抽样

```python
# 随机抽样
random_sample = data_before_sample
random_sample_finish = random_sample.sample(n=50, random_state=42)
random_sample_finish = random_sample_finish[columns]
random_sample_finish
```

[5] ✓ 0.0s Python

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
296	63	58	通辽	一般节点	2549	922	沈阳	网络核心	49092144382	1.000000e+11
79	180	192	呼和浩特	一般节点	591	586	绥化	一般节点	49504348509	1.000000e+11
830	36036	54	长春	一般节点	591	11	绥化	一般节点	49794381448	1.000000e+11
113	474	678	哈尔滨	一般节点	1997	124	天津	网络核心	49044545927	1.000000e+11
997	36036	52	长春	一般节点	63	12	通辽	一般节点	50822505842	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
84	180	214	呼和浩特	一般节点	2701	135	大连	网络核心	48901190886	1.000000e+11
499	47	417	通辽	一般节点	2701	300	大连	网络核心	49964894755	1.000000e+11
136	591	19	绥化	一般节点	36036	18	长春	一般节点	49524524277	1.000000e+11

（3）分层抽样

```python
# 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
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(n=min(17, len(ybjd)), random_state=42),
    wlhx.sample(n=min(33, len(wlhx)), random_state=42)
])
after_sample
```

[6] ✓ 0.0s Python

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
622	180	20	呼和浩特	一般节点	36036	499	长春	一般节点	49636788433	1.000000e+11
913	2473	799	吉林	一般节点	47	243	通辽	一般节点	50993016382	1.000000e+11
966	36539	1146	杭州	一般节点	63	12	通辽	一般节点	49520418698	1.000000e+11
530	47	249	通辽	一般节点	2473	799	吉林	一般节点	49803820036	1.000000e+11
498	47	314	通辽	一般节点	591	586	绥化	一般节点	50043006782	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
1023	96	134	呼和浩特	一般节点	96	124	呼和浩特	一般节点	49523879533	1.000000e+11
136	591	19	绥化	一般节点	36036	18	长春	一般节点	49524524277	1.000000e+11

（4）系统抽样

```python
# 系统抽样：每隔 k 个样本取一个
def systematic_sampling(df, n):
    k = len(df) // n
    start = np.random.randint(0, k)
    indices = np.arange(start, len(df), k)
    return df.iloc[indices[:n]]

systematic_sample = systematic_sampling(data_before_sample, 50)
systematic_sample
```

[7] ✓ 0.0s Python

	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

（5）整群抽样

```python
# 整群抽样：随机选取一个群组（以 to_level 为群），再从中抽取 50 个样本
clusters = data_before_sample['to_level'].unique()
chosen_cluster = np.random.choice(clusters, 1)[0]
cluster_data = data_before_sample[data_before_sample['to_level'] == chosen_cluster]

# 若该群组样本数少于50，取全部，否则随机抽取50个
cluster_sample = cluster_data.sample(
    n=min(50, len(cluster_data)),
    random_state=42
)
print("随机选中的群组为：", chosen_cluster)
cluster_sample.head()
```

[12] ✓ 0.0s Python

```
随机选中的群组为： 一般节点
```

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	to_level	traffic	bandwidth
622	180	20	呼和浩特	一般节点	36036	499	长春	一般节点	49636788433	1.000000e+11
913	2473	799	吉林	一般节点	47	243	通辽	一般节点	50993016382	1.000000e+11
966	36539	1146	杭州	一般节点	63	12	通辽	一般节点	49520418698	1.000000e+11
530	47	249	通辽	一般节点	2473	799	吉林	一般节点	49803820036	1.000000e+11
498	47	314	通辽	一般节点	591	586	绥化	一般节点	50043006782	1.000000e+11

4. 汇总比较不同采样方法的结果

```python
# 汇总比较不同采样方法的结果
print("原始数据量：", len(data_before_sample))
print("加权采样样本数：", len(weight_sample_finish))
print("随机采样样本数：", len(random_sample_finish))
print("分层采样样本数：", len(after_sample))
print("系统采样样本数：", len(systematic_sample))
print("整群采样样本数：", len(cluster_sample))
```

[14] ✓ 0.0s Python

```
原始数据量： 550
加权采样样本数： 50
随机采样样本数： 50
分层采样样本数： 50
系统采样样本数： 50
整群采样样本数： 50
```

结论分析与体会：

本次实验通过 Pandas 库实现了多种数据采样与过滤方法，包括随机抽样、加权抽样、分层抽样和整群抽样。

实验过程中掌握了 dropna()、loc[]、sample() 等函数的使用方法，并理解了不同采样方式的适用场景。随机抽样能够快速获得代表性样本；加权抽样适合样本分布不均的情况；分层抽样可确保各类别均衡；整群抽样比较简单，提高抽样效率。

通过对比不同结果，体会到数据预处理和抽样方法对后续分析结果的影响。