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：202300130205	姓名：李尚远	班级：23 级数据班
实验题目：**数据采样方法实践**		
实验学时：2	实验日期：2025. 9. 15	

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：

一、

导入数据集

```
import pandas as pd
# from pandas import DataFrame
import numpy as np
[1]
```

```
#读取数据
primitive_data=pd.read_csv("data.csv",encoding="gbk")
print(primitive_data)
[2]
```

```
14          47         417          通辽          一般节点          96         391       呼和浩特
15          47         425          通辽          一般节点        1756        1018         北京
16          47         427          通辽          一般节点        1997         213         天津
17          63           6          通辽          一般节点         591          23         绥化
18          63          10          通辽          一般节点         235         106         北京
19          63          12          通辽          一般节点         180         252       呼和浩特
20          63          54          通辽          一般节点         235         100         北京
21          63          58          通辽          一般节点       36036          54         长春
22          63          60          通辽          一般节点       36422         258         天津
23          63          62          通辽          一般节点       36422         394         天津
24          63          66          通辽          一般节点         235         112         北京
25          63          70          通辽          一般节点         180         264       呼和浩特
26          63          74          通辽          一般节点        2701         181         大连
27          63         224          通辽          一般节点         180          20       呼和浩特
```

二、删除具有缺失值的单元格

```
#去除有缺失值的单元格
primitive_data_1=primitive_data.dropna(how='any') #how='all'表示选择那些所有列都是缺失值的行
print(primitive_data_1)
```

[3]

1106	网络核心	49364931565	1.000000e+11
1107	网络核心	49345226162	1.000000e+11
1108	网络核心	50812501057	1.000000e+11
1109	一般节点	49686438362	1.000000e+11
1110	网络核心	49750726117	1.000000e+11
1111	网络核心	49757975617	1.000000e+11
1112	网络核心	49349239913	1.000000e+11
1113	网络核心	48731433404	1.000000e+11
1114	一般节点	50060666120	1.000000e+11
1115	网络核心	50545082113	1.000000e+11
1116	网络核心	50628787089	1.000000e+11
1117	网络核心	48753971761	1.000000e+11

[1118 rows x 10 columns]

三、

```
#数据筛选，选择traffic=0的数据，和from_level=一般节点的数据
data_after_filter_1=primitive_data_1.loc[primitive_data_1["traffic"]!=0]
data_after_filter_2=data_after_filter_1.loc[data_after_filter_1["from_level"]=="一般节点"]
print(data_after_filter_2)
```

[4]

1062	网络核心	49590902097	1.000000e+11
1063	一般节点	49900452417	1.000000e+11
1073	网络核心	49459363742	1.000000e+11
1075	网络核心	50488255524	1.000000e+11
1079	一般节点	50209459772	1.000000e+11
1086	网络核心	51411580502	1.000000e+11
1093	网络核心	47929885030	1.000000e+11
1097	一般节点	48409925693	1.000000e+11
1103	网络核心	48663350759	1.000000e+11
1104	一般节点	50355678076	1.000000e+11
1107	网络核心	49345226162	1.000000e+11
1115	网络核心	50545082113	1.000000e+11

[550 rows x 10 columns]

四、随机抽样

```python
#随机采样前的预处理
data_before_sample=data_after_filter_2
random_sample=data_before_sample
columns=data_before_sample.columns  #用于后续回复数据列结构
```

[5]

```python
#随机抽样
random_sample_finish=random_sample.sample(n=50)
print(random_sample_finish)
random_sample_finish=random_sample_finish[columns]
# print(random_sample_finish)
```

[6]

	from_dev	from_port	from_city	from_level	to_dev	to_port	to_city	\
799	180	52	呼和浩特	一般节点	474	460	哈尔滨	
51	96	156	呼和浩特	一般节点	3227	103	济南	
44	96	127	呼和浩特	一般节点	1756	1027	北京	
423	591	558	绥化	一般节点	180	20	呼和浩特	
124	474	1311	哈尔滨	一般节点	2549	1570	沈阳	
9	47	252	通辽	一般节点	96	134	呼和浩特	
41	96	120	呼和浩特	一般节点	1997	250	天津	

五、

```python
#分层抽样
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"]
print(ybjd.shape[0]/(ybjd.shape[0]+wlhx.shape[0]),wlhx.shape[0]/(ybjd.shape[0]+wlhx.shape[0]))
after_sample=pd.concat([ybjd.sample(17),wlhx.sample(33)])
print(after_sample)
```

[17]

373	网络核心	50339382092	1.000000e+11
452	网络核心	49891276242	1.000000e+11
142	网络核心	51256753219	1.000000e+11
431	网络核心	49411244329	1.000000e+11
950	网络核心	50524728588	1.000000e+11
398	网络核心	50733378641	1.000000e+11
277	网络核心	47105527982	1.000000e+11
116	网络核心	48505909225	1.000000e+11
333	网络核心	50062726803	1.000000e+11
114	网络核心	50262691915	1.000000e+11
325	网络核心	48893583868	1.000000e+11
409	网络核心	50469487601	1.000000e+11
422	网络核心	48492868383	1.000000e+11
15	网络核心	50796899329	1.000000e+11

六、

```python
#加权随机采样 1 5
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]
print(weight_sample_finish)
```

[18]

1097	一般节点	48409925693	1.000000e+11
0	网络核心	49636052613	1.000000e+11
378	网络核心	50470657254	1.000000e+11
393	网络核心	49693039378	1.000000e+11
302	网络核心	50870996562	1.000000e+11
691	网络核心	48978564669	1.000000e+11
1103	网络核心	48663350759	1.000000e+11
660	网络核心	50555895575	1.000000e+11
366	网络核心	47435896137	1.000000e+11
701	网络核心	48906180396	1.000000e+11
66	网络核心	51023900961	1.000000e+11
447	一般节点	49557001334	1.000000e+11
929	网络核心	49145116989	1.000000e+11

结果图片：