第二章

1.实践案例一

1）留出法的各步骤python代码

#导入必要的库

import pandas as pd

from sklearn.model\_selection import train\_test\_split

#读取数据

data = pd.read\_excel(r"D:\研\编实验教材\data.xlsx")

#选择特征变量和目标变量

X = data[['X']]

Y = data['Y']

#划分数据集

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.3, random\_state=42)

#输出划分后的数据集大小

print(f"训练集大小: {len(X\_train)}")

print(f"测试集大小: {len(X\_test)}")

X\_train.to\_excel('train\_X.xlsx', index=False)

Y\_train.to\_excel('train\_Y.xlsx', index=False)

X\_test.to\_excel('test\_X.xlsx', index=False)

Y\_test.to\_excel('test\_Y.xlsx', index=False)

2）交叉验证法的各步骤python代码

#导入必要的库

import pandas as pd

from sklearn.model\_selection import KFold

#读取数据

data = pd.read\_excel(r"D:\研\编实验教材\data.xlsx")

#准备特征矩阵和目标变量

X = data[['X']]

y = data['Y']

#设置交叉验证参数

k = 5

kf = KFold(n\_splits=k, shuffle=True, random\_state=42)

#执行交叉验证划分

for train\_index, test\_index in kf.split(X):

X\_train, X\_test = X.iloc[train\_index], X.iloc[test\_index]

y\_train, y\_test = y.iloc[train\_index], y.iloc[test\_index]

print(f"训练集大小: {len(X\_train)}, 测试集大小: {len(X\_test)}")

3）自助法的各步骤python代码

#导入必要的库

import pandas as pd

import numpy as np

#读取数据

data = pd.read\_excel(r"D:\研\编实验教材\data.xlsx")

#设置采样大小

sample\_size = len(data)

#计算每个样本被选中的概率

prob = np.ones(sample\_size) / sample\_size

#生成训练集索引

train\_indices = np.random.choice(np.arange(sample\_size), size=sample\_size, p=prob, replace=True)

#生成测试集索引

test\_indices = np.setdiff1d(np.arange(sample\_size), train\_indices)

while len(test\_indices) == 0:

train\_indices = np.random.choice(np.arange(sample\_size), size=sample\_size,

p=prob, replace=True)

test\_indices = np.setdiff1d(np.arange(sample\_size), train\_indices)

#根据索引划分训练集和测试集

train\_data = data.iloc[train\_indices]

test\_data = data.iloc[test\_indices]

print("训练集大小:", len(train\_data))

print("测试集大小:", len(test\_data))

print("训练集数据:")

print(train\_data.head())

print("测试集数据:")

print(test\_data.head())

2．实践案例二

1）Bootstrap抽样的各步骤python代码

#导入必要的库

import pandas as pd

import numpy as np

#读取数据

data = pd.read\_excel(r"D:\研\编实验教材\data.xlsx")

#定义Bootstrap抽样函数

def bootstrap\_sample(data, n\_samples):

indices = np.arange(len(data))

bootstrap\_indices = np.random.choice(indices, size=n\_samples, replace=True)

bootstrap\_sample = data.iloc[bootstrap\_indices]

return bootstrap\_sample

#进行Bootstrap抽样

n\_iterations = 1000

n\_samples = len(data)

bootstrap\_samples = []

for \_ in range(n\_iterations):

sample = bootstrap\_sample(data, n\_samples)

bootstrap\_samples.append(sample)

#处理Bootstrap样本

bootstrap\_samples\_df = pd.concat(bootstrap\_samples, ignore\_index=True).reset\_index(drop=True)

for i in range(2):

print(f"Bootstrap Sample {i+1}:\n", bootstrap\_samples[i].head(3), "\n")

2）分层抽样的各步骤python代码

#导入必要的库

import pandas as pd

#读取数据

df = pd.read\_excel(r"D:\研\编实验教材\分层抽样.xlsx")

#查看分层依据的分布情况

print(df['组别'].value\_counts())

#初始化抽样结果存储

sampled\_data = {}

#遍历组别的唯一值，在每个层中进行抽样

for y\_value in df['组别'].unique():

layer\_df = df[df['组别'] == y\_value]

sample = layer\_df.sample(n=1, replace=False)

sampled\_data[y\_value] = sample

#合并抽样结果

sampled\_df = pd.concat(sampled\_data.values()).reset\_index(drop=True)

print(sampled\_df)

3）系统抽样的各步骤python代码

#导入必要的库

import pandas as pd

import numpy as np

#读取数据

data = pd.read\_excel(r"D:\研\编实验教材\data.xlsx")

#确定抽样比例和样本量

sample\_size = int(len(data) \* 0.1)

#计算抽样间隔

interval = len(data) // sample\_size

#修正后的抽样间隔计算

if len(data) % sample\_size != 0:

interval = len(data) // (sample\_size + 1)

#进行系统抽样

indices = np.arange(0, len(data), interval)

sampled\_data = data.iloc[indices]

print(sampled\_data)