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：202300130028	姓名： 苗雨健	班级： 数据 23
实验题目：数据质量实践		
实验学时：2	实验日期： 2025/9/26	

实验目的：

本次实验主要围绕宝可梦数据集进行分析，考察在拿到数据后如何对现有的数据进行预处理清洗操作，建立起对于脏数据、缺失数据等异常情况的一套完整流程的认识

硬件环境：

计算机一台

软件环境：

Linux 或 Windows

实验步骤与内容：

首先加载必要库，在正常导入时遇到问题，我们采用 latin-1 格式进行导入

```
# 导入必要的库
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```

df = pd.read_csv("C:\\Users\\33566\\Downloads\\Pokemon.csv", encoding='latin-1')

```

删去末尾空行

```python
# 2. 查看并删除末尾无意义数据
print("最后5行数据:")
print(df.tail())

# 删除最后四行无意义数据
df_clean = df.iloc[:-4].copy()
```

```
最后5行数据:
              #       Name     Type 1     Type 2      Total         HP  \
805         721  Volcanion       Fire      Water        600         80
806   undefined  undefined  undefined  undefined  undefined  undefined
807   undefined  undefined  undefined  undefined  undefined  undefined
808         NaN        NaN        NaN        NaN        NaN        NaN
809         NaN        NaN        NaN        NaN        NaN        NaN

          Attack    Defense    Sp. Atk    Sp. Def      Speed Generation  \
805          110        120        130         90         70          6
806   undefined  undefined  undefined  undefined  undefined  undefined
```

然后对 type2 进行处理

```python
# 3. 处理Type 2列的异常值
# 查看Type 2列的唯一值
print("Type 2列唯一值:")
print(df_clean['Type 2'].unique())

# 查看Type 2列的值分布
type2_counts = df_clean['Type 2'].value_counts()
print("\nType 2值分布:")
print(type2_counts)

# 将错误项替换为NaN
df_clean['Type 2'] = df_clean['Type 2'].replace(['A', '273', '0', 'BBB'], np.nan)
```

```
Type 2列唯一值:
['Poison' nan 'Flying' 'Dragon' '0' 'Ground' '273' 'Fairy' 'Grass'
 'Fighting' 'Psychic' 'Steel' 'Ice' 'A' 'Rock' 'Dark' 'Water' 'Electric'
 'Fire' 'Ghost' 'Bug' 'BBB' 'Normal']

Type 2值分布:
Type 2
Flying      98
Poison      37
Ground      35
Psychic     33
```

去除重复

```python
# 4. 检查并删除重复行
duplicates = df_clean.duplicated()
print(f"重复行数量: {duplicates.sum()}")

if duplicates.sum() > 0:
    print("重复的行:")
    print(df_clean[duplicates])
    # 删除重复行，保留第一个出现的数据
    df_clean = df_clean.drop_duplicates()

print(f"删除重复值后数据形状: {df_clean.shape}")
```

```
185  168    Ariados     Bug  Poison   390  7
186  168    Ariados     Bug  Poison   390  7
187  168    Ariados     Bug  Poison   390  7

     Speed Generation Legendary
15      30          1     FALSE
23      71          1     FALSE
185     40          2     FALSE
186     40          2     FALSE
187     40          2     FALSE
删除重复值后数据形状: (801, 13)
```

处理 attack 异常值

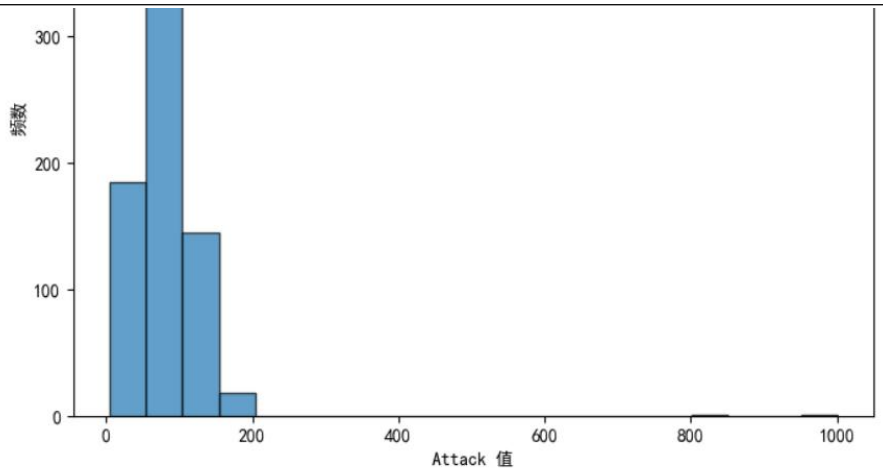
```python
# 5. 分析Attack属性分布并处理异常值
# 设置中文字体
plt.rcParams['font.sans-serif'] = ['SimHei', 'Microsoft YaHei', 'DejaVu Sans']
plt.rcParams['axes.unicode_minus'] = False

# 转换数据类型为数值型
df_clean['Attack'] = pd.to_numeric(df_clean['Attack'], errors='coerce')

# 绘制Attack属性直方图
plt.figure(figsize=(8, 6))
plt.hist(df_clean['Attack'].dropna(), bins=20, alpha=0.7, edgecolor='black')
plt.title('Attack 属性分布')
plt.xlabel('Attack 值')
plt.ylabel('频数')
plt.show()

# 使用IQR方法识别异常值
Q1 = df_clean['Attack'].quantile(0.25)
Q3 = df_clean['Attack'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

print(f"Q1: {Q1}, Q3: {Q3}, IQR: {IQR}")
print(f"异常值边界: [{lower_bound}, {upper_bound}]")
```




```
Q1: 55.0, Q3: 100.0, IQR: 45.0
异常值边界: [-12.5, 167.5]
Attack 异常值数量: 9
异常值数据:
                        Name  Attack
9                  Squirtle   840.0
140                  Tauros  1000.0
165      MewtwoMega Mewtwo X   190.0
237  HonedgeossMega Honedgeoss   185.0
```

修正 Generation 和 Legendary 属性的置换错误

```
10  # 6. 修正Generation和Legendary属性的置换错误
    # 查找并交换置换的属性值
    for idx, row in df_clean.iterrows():
        gen_val = str(row['Generation'])
        leg_val = str(row['Legendary'])
        # 判断是否存在属性置换（Generation为文本，Legendary为数字）
        if gen_val.isalpha() and leg_val.isdigit():
            print(f"发现置换行: {row['Name']}")
            print(f"  原始: Generation={gen_val}, Legendary={leg_val}")
            # 交换值
            df_clean.at[idx, 'Generation'] = leg_val
            df_clean.at[idx, 'Legendary'] = gen_val
            print(f"  修正后: Generation={leg_val}, Legendary={gen_val}")

    # 转换Generation列的数据类型为数值型
    df_clean['Generation'] = pd.to_numeric(df_clean['Generation'], errors='coerce')
```

```
发现置换行: Blastoise
  原始: Generation=FALSE, Legendary=1
  修正后: Generation=1, Legendary=FALSE
发现置换行: Pikachu
  原始: Generation=FALSE, Legendary=0
  修正后: Generation=0, Legendary=FALSE
```

结论分析与体会：

考察在拿到数据后如何对现有的数据进行预处理清洗操作，建立了对于脏数据、缺失数据等异常情况的一套完整流程的认识