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：202320130203	姓名：李姿含	班级：数据 23
实验题目：数据质量实践		
实验学时：2	实验日期：20251008	

实验目标：

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。

作品描述（实验背景、数据集来源、描述思路）：

一、实验背景

在大数据分析流程中，原始数据常存在格式错误、重复记录、异常值等“脏数据”问题，这些问题会直接影响后续分析结果的准确性与可靠性。宝可梦数据集包含 721 只宝可梦的基础信息与能力属性，涵盖名称、属性类型、生命值（HP）、攻击力（Attack）等关键字段，适合作为数据清洗实践的样本。通过对该数据集的预处理，可掌握数据质量优化的核心方法。

二、数据集来源

Pokeman Dataset：721 Pokemon, including their number, name, first and second type, and basic stats: HP, Attack, Defense, Special Attack, Special Defense, and Speed

数据集下载链接：http://storage.ameshol land.xyz/Pokemon.csv

三、描述思路

① 删除无意义数据：原始数据集末尾存在 2 行无业务意义的无效记录，通过索引切片保留有效数据

```
#删除最后两行无意义数据
df_clean=df.iloc[:-2].copy()#保留除最后两行外的所有数据
print(df_clean)
```

② 清理 Type2 列异常值：通过 value_counts 查看 Type2 列取值分布，定位异常值并剔除对应记录

```
print(df_clean["Type 2"].value_counts(dropna=False))#查看 Type2 的取值分布
df_clean=df_clean[df_clean["Type 2"]!="A"].copy()
df_clean=df_clean[df_clean["Type 2"]!="273"].copy()
df_clean=df_clean[df_clean["Type 2"]!="0"].copy()
df_clean=df_clean[df_clean["Type 2"]!="BBB"].copy()
```

③ 删除重复值：检测重复数据行，保留首次出现的记录，删除后续重复项

```
duplicate_count=df_clean.duplicated().sum()
if duplicate_count>0:
    print(df_clean[df_clean.duplicated(keep=False)])#显示所有重复行
df_clean=df_clean.drop_duplicates(keep="first").copy()
print(df_clean)
```

④ 处理 Attack 列异常高值：用四分位距法定义异常值阈值，识别并处理攻击字段的极端值

```
Q1 = df_clean["Attack"].quantile(0.25)  # 第一四分位数
Q3 = df_clean["Attack"].quantile(0.75)  # 第三四分位数
IQR = Q3 - Q1  # 四分位距
upper_bound = Q3 + 1.5 * IQR  # 上异常值阈值（超过则为异常）
lower_bound = Q1 - 1.5 * IQR  # 下异常值阈值（宝可梦 Attack 无负值，可忽略）
attack_outliers = df_clean[df_clean["Attack"] > upper_bound]
#print(f"\nAttack 异常高值记录数量：{len(attack_outliers)}")
if len(attack_outliers) > 0:
    print("Attack 异常高值详情：")
    print(attack_outliers[["Name", "Type 1", "Type 2", "Attack", "Total"]])
df_clean = df_clean[df_clean["Attack"] <= upper_bound].copy()
```

⑤ 修复 Generation 和 Legendary 字段置换：定义字段值置换的异常行，交换两列值并规范数据类型

```
cond1=df_clean["Generation"].isin([True,False])
cond2=pd.to_numeric(df_clean["Legendary"],errors="coerce").notna()
swap_rows=df_clean[cond1|cond2]#满足任一条件即为置换行
#交换两列的值
df_clean.loc[cond1|cond2,["Generation","Legendary"]]=df_clean.loc[cond1|cond2,
["Legendary","Generation"]].values
df_clean["Generation"]=pd.to_numeric(df_clean["Generation"],errors="coerce").a
stype("Int64")
df_clean["legendary"]=df_clean["Legendary"].map({True:True,False:False,1:True,
0:False})
print("\n 修复后 Generation 列取值分布：")
print(df_clean["Generation"].value_counts(dropna=False))
print("\n 修复后 Legendary 列取值分布：")
print(df_clean["Legendary"].value_counts(dropna=False))
```

结果图片：

初始数据集：

```
D:\anaconda3\python.exe C:\Users\86134\PycharmProjects\knn\.venv\big_
             #                   Name  ... Generation  Legendary
0            1              Bulbasaur  ...          1      FALSE
1            2                Ivysaur  ...          1      FALSE
2            3               Venusaur  ...          1      FALSE
3            3  VenusaurMega Venusaur  ...          1      FALSE
4            4             Charmander  ...          1      FALSE
..         ...                    ...  ...        ...        ...
805        721              Volcanion  ...          6       TRUE
806  undefined              undefined  ...  undefined  undefined
807  undefined              undefined  ...  undefined  undefined
808        NaN                    NaN  ...        NaN
809        NaN                    NaN  ...        NaN

[810 rows x 13 columns]

进程已结束，退出代码为 0
```

删除最后两行无意义数据：

```
             #                   Name  ... Generation  Legendary
0            1              Bulbasaur  ...          1      FALSE
1            2                Ivysaur  ...          1      FALSE
2            3               Venusaur  ...          1      FALSE
3            3  VenusaurMega Venusaur  ...          1      FALSE
4            4             Charmander  ...          1      FALSE
..         ...                    ...  ...        ...        ...
803        720    HoopaHoopa Confined  ...          6       TRUE
804        720     HoopaHoopa Unbound  ...          6       TRUE
805        721              Volcanion  ...          6       TRUE
806  undefined              undefined  ...  undefined  undefined
807  undefined              undefined  ...  undefined  undefined

[808 rows x 13 columns]

进程已结束，退出代码为 0
```

查看 Type2 列取值分布：

```
Type 2
NaN          384
Flying        98
Poison        37
Ground        35
Psychic       33
Fighting      26
Grass         25
Fairy         23
Steel         22
Dark          20
Dragon        18
Rock          14
Ghost         14
Water         14
Ice           14
Fire          12
Electric       6
Normal         4
Bug            3
undefined      2
A              1
273            1
0              1
BBB            1
Name: count, dtype: int64
```

Type2 列删除异常值后数据：

```
            #                   Name  ... Generation  Legendary
0           1              Bulbasaur  ...          1      FALSE
1           2                Ivysaur  ...          1      FALSE
2           3               Venusaur  ...          1      FALSE
3           3  VenusaurMega Venusaur  ...          1      FALSE
4           4             Charmander  ...          1      FALSE
..        ...                    ...  ...        ...        ...
803       720    HoopaHoopa Confined  ...          6       TRUE
804       720     HoopaHoopa Unbound  ...          6       TRUE
805       721              Volcanion  ...          6       TRUE
806  undefined              undefined  ...  undefined  undefined
807  undefined              undefined  ...  undefined  undefined

[804 rows x 13 columns]

进程已结束，退出代码为 0
```

Type2 列修复后取值分布：

```
Type 2
NaN          384
Flying        98
Poison        37
Ground        35
Psychic       33
Fighting      26
Grass         25
Fairy         23
Steel         22
Dark          20
Dragon        18
Water         14
Ghost         14
Ice           14
Rock          14
Fire          12
Electric       6
Normal         4
Bug            3
undefined      2
Name: count, dtype: int64

进程已结束，退出代码为 0
```

删除重复值：

```
              #                   Name  ... Generation  Legendary
0             1              Bulbasaur  ...          1      FALSE
1             2                Ivysaur  ...          1      FALSE
2             3               Venusaur  ...          1      FALSE
3             3  VenusaurMega Venusaur  ...          1      FALSE
4             4             Charmander  ...          1      FALSE
..          ...                    ...  ...        ...        ...
802         719    DiancieMega Diancie  ...          6       TRUE
803         720    HoopaHoopa Confined  ...          6       TRUE
804         720     HoopaHoopa Unbound  ...          6       TRUE
805         721              Volcanion  ...          6       TRUE
806   undefined              undefined  ...  undefined  undefined

[798 rows x 13 columns]

进程已结束，退出代码为 0
```

删除 Attack 列的异常高值：

```
Attack异常值检测阈值：
第一四分位数(Q1)：55.00
第三四分位数(Q3)：100.00
上阈值（异常值分界）：167.50

Attack异常高值记录数量：9
Attack异常高值详情：
                        Name   Type 1    Type 2  Attack Total
9                   Squirtle    Water       NaN   840.0   314
140                   Tauros   Normal       NaN  1000.0   490
165      MewtwoMega Mewtwo X  Psychic  Fighting   190.0   780
237  HeracrossMega Heracross      Bug  Fighting   185.0   600
430    GroudonPrimal Groudon   Ground      Fire   180.0   770
432    RayquazaMega Rayquaza   Dragon    Flying   180.0   780
435       DeoxysAttack Forme  Psychic       NaN   180.0   600
500    GarchompMega Garchomp   Dragon    Ground   170.0   700
717       KyuremBlack Kyurem   Dragon       Ice   170.0   700
处理Attack异常值后数据维度：(787, 13)

进程已结束，退出代码为 0
```

修复 Generation 和 Legendary 属性置换：

```
修复前Generation列取值分布：
Generation
5            164
1            158
3            156
4            120
2            105
6             81
FALSE          2
undefined      1
Name: count, dtype: int64

修复前legendary列取值分布：
Legendary
FALSE     721
TRUE       59
0           3
1           1
Poison      1
Ground      1
NaN         1
Name: count, dtype: int64
```

```
修复后Generation列取值分布：
Generation
5       164
1       157
3       156
4       120
2       105
6        81
0         3
<NA>      1
Name: count, dtype: Int64

修复后Legendary列取值分布：
Legendary
FALSE     723
TRUE       59
1           2
Poison      1
Ground      1
NaN         1
Name: count, dtype: int64

进程已结束，退出代码为 0
```