大数据分析实践实验报告

实验二 数据质量实践

一、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

二、实验环境

python3，jupyter notebook

三、实验过程

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

data = pd.read_csv(r"C:\Users\边鑫磊\Desktop\Pokemon.csv", encoding='ISO-8859-1')
data
```

[1]:

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	FALSE
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	FALSE
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	FALSE
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	FALSE
4	4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	FALSE
...	...	...	...	...	...	...	...	...	...	...	...	...	...
805	721	Volcanion	Fire	Water	600	80	110	120	130	90	70	6	TRUE
806	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined
807	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined
808	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
809	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

810 rows × 13 columns

导入库&导入数据

[2]:
```
data = data.iloc[:-4]
data
```

[2]:

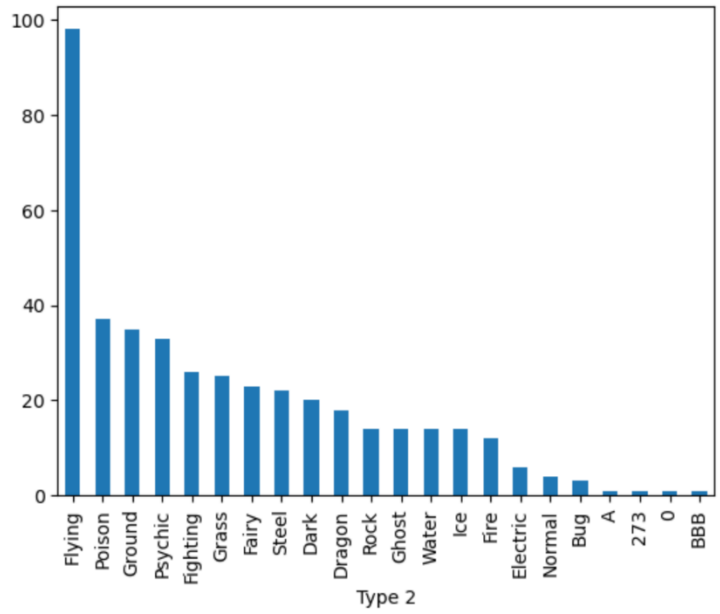
	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	FALSE
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	FALSE
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	FALSE
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	FALSE
4	4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	FALSE
...	...	...	...	...	...	...	...	...	...	...	...	...	...
801	719	Diancie	Rock	Fairy	600	50	100	150	100	150	50	6	TRUE
802	719	DiancieMega Diancie	Rock	Fairy	700	50	160	110	160	110	110	6	TRUE
803	720	HoopaHoopa Confined	Psychic	Ghost	600	80	110	60	150	130	70	6	TRUE
804	720	HoopaHoopa Unbound	Psychic	Dark	680	80	160	60	170	130	80	6	TRUE
805	721	Volcanion	Fire	Water	600	80	110	120	130	90	70	6	TRUE

806 rows × 13 columns

删除无意义的数据和多余的空行后（包括 4 行数据）

[4]:
```
data["Type 2"].value_counts().plot(kind='bar')
```

[4]:
```
<Axes: xlabel='Type 2'>
```



可以发现，type2 存在异常的数值取值，需要删除

[6]:
```
data=data[data["Type 2"]!='273']
data
```

[6]:

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	FALSE
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	FALSE
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	FALSE
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	FALSE
4	4	Charmander	Fire	NaN	309	39	52	43	60	50	65	1	FALSE
...	...	...	...	...	...	...	...	...	...	...	...	...	...
801	719	Diancie	Rock	Fairy	600	50	100	150	100	150	50	6	TRUE
802	719	DiancieMega Diancie	Rock	Fairy	700	50	160	110	160	110	110	6	TRUE
803	720	HoopaHoopa Confined	Psychic	Ghost	600	80	110	60	150	130	70	6	TRUE
804	720	HoopaHoopa Unbound	Psychic	Dark	680	80	160	60	170	130	80	6	TRUE
805	721	Volcanion	Fire	Water	600	80	110	120	130	90	70	6	TRUE

805 rows × 13 columns

删除 type2 取值异常的数据后（包括 1 行数据）

[8]:
```
data[data.duplicated()]
```

[8]:

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
15	11	Metapod	Bug	NaN	205	50	20	55	25	25	30	1	FALSE
23	17	Pidgeotto	Normal	Flying	349	63	60	55	50	50	71	1	FALSE
185	168	Ariados	Bug	Poison	390	70	90	70	60	60	40	2	FALSE
186	168	Ariados	Bug	Poison	390	70	90	70	60	60	40	2	FALSE
187	168	Ariados	Bug	Poison	390	70	90	70	60	60	40	2	FALSE

可以发现，数据集中存在重复值

[33]:
```
data=data.copy()
data.drop_duplicates(inplace=True)
data
```

[33]:

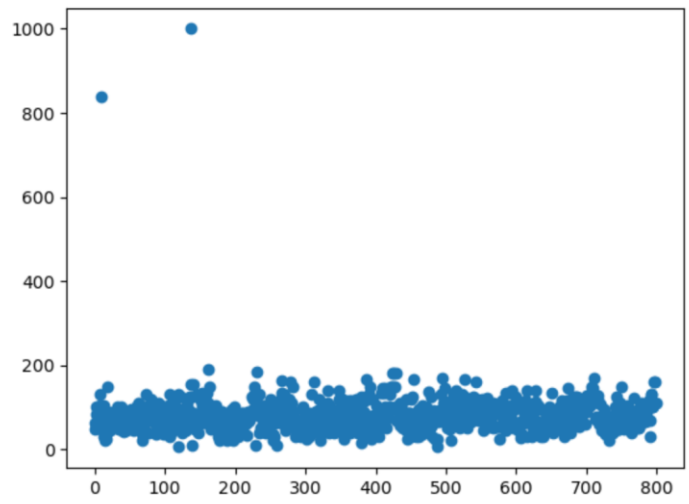
	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	318	45	49.0	49	65	65	45	1	FALSE
1	2	Ivysaur	Grass	Poison	405	60	62.0	63	80	80	60	1	FALSE
2	3	Venusaur	Grass	Poison	525	80	82.0	83	100	100	80	1	FALSE
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100.0	123	122	120	80	1	FALSE
4	4	Charmander	Fire	NaN	309	39	52.0	43	60	50	65	1	FALSE
...	...	...	...	...	...	...	...	...	...	...	...	...	...
801	719	Diancie	Rock	Fairy	600	50	100.0	150	100	150	50	6	TRUE
802	719	DiancieMega Diancie	Rock	Fairy	700	50	160.0	110	160	110	110	6	TRUE
803	720	HoopaHoopa Confined	Psychic	Ghost	600	80	110.0	60	150	130	70	6	TRUE
804	720	HoopaHoopa Unbound	Psychic	Dark	680	80	160.0	60	170	130	80	6	TRUE
805	721	Volcanion	Fire	Water	600	80	110.0	120	130	90	70	6	TRUE

800 rows × 13 columns

删除重复值后（包括 5 行数据）

[12]:
```
import matplotlib.pyplot as plt
data.iloc[:, 6] = pd.to_numeric(data.iloc[:, 6], errors='coerce')
plt.scatter(range(0,data.shape[0]),data.iloc[:,6])
```

[12]:
```
<matplotlib.collections.PathCollection at 0x21ee191ab10>
```



可以发现，存在 Attack 属性过高的异常值

```
[17]: data_clean = data[data['Attack'] < 800].copy()
      data_clean
```

[17]:

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	318	45	49.0	49	65	65	45	1	FALSE
1	2	Ivysaur	Grass	Poison	405	60	62.0	63	80	80	60	1	FALSE
2	3	Venusaur	Grass	Poison	525	80	82.0	83	100	100	80	1	FALSE
3	3	VenusaurMega Venusaur	Grass	Poison	625	80	100.0	123	122	120	80	1	FALSE
4	4	Charmander	Fire	NaN	309	39	52.0	43	60	50	65	1	FALSE
...	...	...	...	...	...	...	...	...	...	...	...	...	...
801	719	Diancie	Rock	Fairy	600	50	100.0	150	100	150	50	6	TRUE
802	719	DiancieMega Diancie	Rock	Fairy	700	50	160.0	110	160	110	110	6	TRUE
803	720	HoopaHoopa Confined	Psychic	Ghost	600	80	110.0	60	150	130	70	6	TRUE
804	720	HoopaHoopa Unbound	Psychic	Dark	680	80	160.0	60	170	130	80	6	TRUE
805	721	Volcanion	Fire	Water	600	80	110.0	120	130	90	70	6	TRUE

797 rows × 13 columns

删除 Attack 属性过高的异常值后（包括 3 行数据）

```
[39]: data_clean.iloc[10:30]
```

[39]:

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
11	9	Blastoise	Water	NaN	530	79	83.0	100	85	105	78	FALSE	1
12	9	BlastoiseMega Blastoise	Water	NaN	630	79	103.0	120	135	115	78	1	FALSE
13	10	Caterpie	Bug	NaN	195	45	30.0	35	20	20	45	1	FALSE
14	11	Metapod	Bug	NaN	205	50	20.0	55	25	25	30	1	FALSE
16	12	Butterfree	Bug	Flying	395	60	45.0	50	90	80	70	1	FALSE
17	13	Weedle	Bug	Poison	195	NaN	35.0	30	20	20	50	1	FALSE
18	14	Kakuna	Bug	Poison	205	45	25.0	50	25	25	35	1	FALSE
19	15	Beedrill	Bug	Poison	395	65	90.0	40	45	80	75	1	FALSE
20	15	BeedrillMega Beedrill	Bug	Poison	495	65	150.0	40	15	80	145	1	FALSE
21	17	Pidgeotto	Normal	Flying	349	63	60.0	55	50	50	71	1	FALSE
22	16	Pidgey	Normal	Flying	251	40	45.0	40	35	35	56	1	FALSE
24	18	Pidgeot	Normal	Flying	479	83	80.0	75	70	70	101	1	FALSE
25	18	PidgeotMega Pidgeot	Normal	Flying	579	83	80.0	80	135	80	121	1	FALSE
26	19	Rattata	Normal	NaN	253	30	56.0	35	25	35	72	1	FALSE
27	20	Raticate	Normal	NaN	413	55	81.0	60	50	70	97	1	FALSE
28	21	Spearow	Normal	Flying	262	40	60.0	30	31	31	70	1	FALSE
29	22	Fearow	Normal	Flying	442	65	90.0	65	61	61	100	1	FALSE
30	23	Ekans	Poison	NaN	288	35	60.0	44	40	54	55	1	FALSE
31	24	Arbok	Poison	NaN	438	60	85.0	69	65	79	80	1	FALSE
32	25	Pikachu	Electric	NaN	320	35	55.0	40	50	50	90	FALSE	0

可以发现，有两条数据的 generation 与 Legendary 属性被置换（11 行和 32 行）

```
]: df1=data_clean.copy()
last_two_cols = df.columns[-2:]

row11_col1 = df1.loc[11, last_two_cols[0]]
row11_col2 = df1.loc[11, last_two_cols[1]]
df1.loc[11, last_two_cols[0]] = row11_col2
df1.loc[11, last_two_cols[1]] = row11_col1

row30_col1 = df1.loc[32, last_two_cols[0]]
row30_col2 = df1.loc[32, last_two_cols[1]]
df1.loc[32, last_two_cols[0]] = row30_col2
df1.loc[32, last_two_cols[1]] = row30_col1

df1.iloc[10:30]
```

]:

	#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
11	9	Blastoise	Water	NaN	530	79	83.0	100	85	105	78	1	FALSE
12	9	BlastoiseMega Blastoise	Water	NaN	630	79	103.0	120	135	115	78	1	FALSE
13	10	Caterpie	Bug	NaN	195	45	30.0	35	20	20	45	1	FALSE
14	11	Metapod	Bug	NaN	205	50	20.0	55	25	25	30	1	FALSE
16	12	Butterfree	Bug	Flying	395	60	45.0	50	90	80	70	1	FALSE
17	13	Weedle	Bug	Poison	195	NaN	35.0	30	20	20	50	1	FALSE
18	14	Kakuna	Bug	Poison	205	45	25.0	50	25	25	35	1	FALSE
19	15	Beedrill	Bug	Poison	395	65	90.0	40	45	80	75	1	FALSE
20	15	BeedrillMega Beedrill	Bug	Poison	495	65	150.0	40	15	80	145	1	FALSE
21	17	Pidgeotto	Normal	Flying	349	63	60.0	55	50	50	71	1	FALSE
22	16	Pidgey	Normal	Flying	251	40	45.0	40	35	35	56	1	FALSE
24	18	Pidgeot	Normal	Flying	479	83	80.0	75	70	70	101	1	FALSE
25	18	PidgeotMega Pidgeot	Normal	Flying	579	83	80.0	80	135	80	121	1	FALSE
26	19	Rattata	Normal	NaN	253	30	56.0	35	25	35	72	1	FALSE
27	20	Raticate	Normal	NaN	413	55	81.0	60	50	70	97	1	FALSE
28	21	Spearow	Normal	Flying	262	40	60.0	30	31	31	70	1	FALSE
29	22	Fearow	Normal	Flying	442	65	90.0	65	61	61	100	1	FALSE
30	23	Ekans	Poison	NaN	288	35	60.0	44	40	54	55	1	FALSE
31	24	Arbok	Poison	NaN	438	60	85.0	69	65	79	80	1	FALSE
32	25	Pikachu	Electric	NaN	320	35	55.0	40	50	50	90	0	FALSE

置换后

四、实验总结

数据质量是数据分析有效性的核心前提，它通过完整性、准确性、一致性、唯一性、有效性等维度衡量数据是否满足应用需求，比如说在本实验中，宝可梦数据集中存在的 HP 值缺失、Attack 值异常、Type 2 字段出现异常取值、重复记录以及 Generation 与 Legendary 属性置换等等问题，都直接破坏了数据质量，可能导致后续分析结果失真。而数据清洗是保障数据质量的关键手段，需通过数据探查定位问题，再针对性处理缺失值（如填补合理数值或保留合理空值）、修正异常值、删除重复记录、统一数据格式与取值规则，从而将“脏数据”转化为“可用数据”

数据质量与数据清洗相辅相成，数据质量的评估结果指导清洗方向，清洗操作则直接提升数据质量，二者共同构成从原始数据到可用数据的转化过程。比如在这次实践中，我们先明确了存在哪些数据质量问题，然后通过系统化清洗修复，才能为后续的分析、统计等工作提供可靠基础，避免“垃圾数据进，垃圾结果出”的情况，因此数据清洗是整个数据分析流程中不可或缺的环节