数据挖掘作业：数据探索性分析与数据预处理

陈龙飞 1120190837

1. 问题描述

选择2个数据集进行探索性分析与预处理。

2. 数据集

本次作业选择的数据集为：Alzheimer Disease and Healthy Aging Data in US、Movies Dataset from Pirated Sites。

3. Python数据挖掘代码展示

```
# -*- encoding:utf-8 -*-

import pandas as pd
import numpy as np
import nltk
import matplotlib.pyplot as plt
from sklearn.impute import SimpleImputer, KNNImputer


def get_attribute_type(attribute):
    """
    判断属性的类型

    参数:
    attribute: pandas.Series，表示一个属性的数据列

    返回:
    字符串，表示属性的类型，可能的取值包括"数值属性"、"标称属性"、"文本型属性"、"其他"
    """
    if np.issubdtype(attribute.dtype, np.number):  # 如果数据类型是数值类型
        return "数值属性"
    elif attribute.dtype == 'object':  # 如果数据类型是字符串
        text = attribute.dropna().tolist()
        unique_tokens = set(text)  # 去重
        if len(unique_tokens) > 2000:  # 如果唯一词汇量超过2000个，则认为是文本型属性
            return "文本型属性"
        else:
            return "标称属性"
    else:  # 其他情况
        return "其他"


# 读取数据集
df = pd.read_csv('dataset.csv')

# 遍历每一列数据
num_attr = []
```

```
nom_attr = []
for col_name in df.columns:
    attr_type = get_attribute_type(df[col_name])
    print(col_name, "的类型是：", attr_type)
    if attr_type == "数值属性":
        num_attr.append(col_name)
        # 统计五数概括和缺失值个数
        attribute = df[col_name]
        five_num = attribute.describe()[['min', '25%', '50%', '75%', 'max']]
        missing_values = attribute.isnull().sum()
        print("  五数概括：\n", five_num)
        print("  缺失值个数：", missing_values)

        # 绘制盒图
        plt.boxplot(attribute.dropna().values)
        plt.title(col_name + "的盒图")
        plt.show()
    elif attr_type == "标称属性":
        nom_attr.append(col_name)
        # 统计每个可能取值的频数
        attribute = df[col_name]
        freq_count = attribute.value_counts()
        print("  频数统计：\n", freq_count)

        # 绘制直方图
        plt.hist(attribute.dropna().values, bins=len(freq_count))
        plt.xticks(rotation=90)
        plt.title(col_name + "的直方图")
        plt.show()

print("数值属性名称列表：", num_attr)
print("标称属性名称列表：", nom_attr)

# 策略1：将缺失部分剔除
df1 = df.dropna()

# 策略2：用最高频率值来填补缺失值
df2 = df.fillna(df.mode().iloc[0])

# 策略3：通过属性的相关关系来填补缺失值
imp_mean = SimpleImputer(missing_values=np.nan, strategy='mean')
imp_mean.fit(df)
df3 = pd.DataFrame(imp_mean.transform(df), columns=df.columns)

# 策略4：通过数据对象之间的相似性来填补缺失值
df4 = df.copy()
imp = KNNImputer(n_neighbors=5, weights='uniform')
df4.iloc[:, :] = imp.fit_transform(df4)

# 输出处理后的数据集
print("策略1：将缺失部分剔除")
print(df1)

print("\n策略2：用最高频率值来填补缺失值")
print(df2)
```

```
print("\n策略3：通过属性的相关关系来填补缺失值")
print(df3)

print("\n策略4：通过数据对象之间的相似性来填补缺失值")
print(df4)

# 导出新数据集
df1.to_csv('data_1.csv', index=False)
df2.to_csv('data_2.csv', index=False)
df3.to_csv('data_3.csv', index=False)
df4.to_csv('data_4.csv', index=False)
```

4. 代码流程分析

首先，设计函数get_attribute_type(attribute)，它的功能是判断数据集中的某个属性是数值属性，还是标称属性，还是其它文本型属性。接下来，遍历数据集中的每一列，判断每个属性的类别。若一个属性为数值属性，则挖掘它的5数并统计缺失值个数，并使用盒图将数据可视化；若一个属性为标称属性，则给出每个可能取值的频数，并使用直方图将数据可视化。最后，实现四种策略对缺失数据进行处理：1）直接将缺失部分剔除；2）用最高频率值来填补缺失值；3）调用SimpleImputer方法找到缺失值相关性最高的值进行填补；4）调用KNNImputer找到缺失值相似性最高（最邻近）的值进行填补。

造成数据缺失的可能原因：1）有些信息暂时无法获取；2）有些信息被遗漏；3）有些对象的某个或某些属性不可用；4）有些信息被认为是不重要的；5）某些信息获取代价过高。

5. 运行结果举例展示

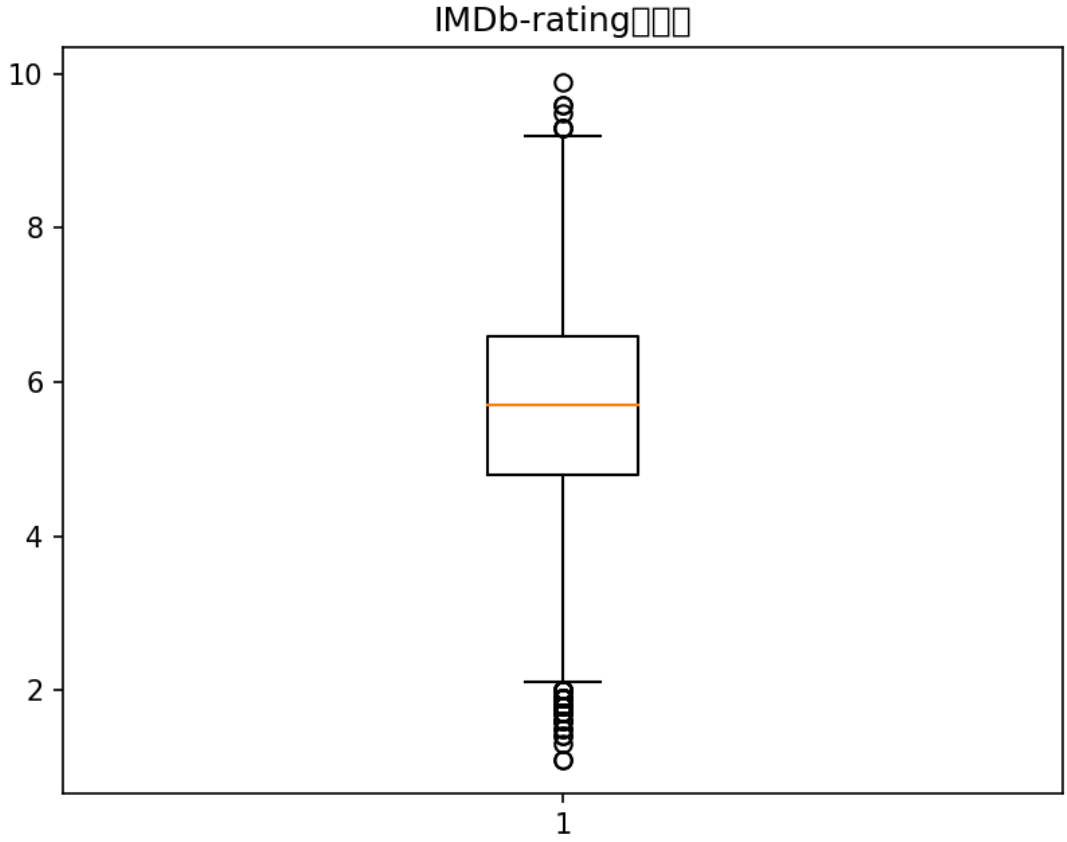
对于数据集Movies Dataset from Pirated Sites：

数值属性名称列表： ['IMDb-rating', 'downloads', 'id', 'run_time', 'views']
标称属性名称列表： ['appropriate_for', 'industry', 'language']

IMDb-rating数据分析：

IMDb-rating 的类型是： 数值属性
五数概括：
min 1.1
25% 4.8
50% 5.7
75% 6.6
max 9.9
Name: IMDb-rating, dtype: float64
缺失值个数： 841

IMDb-rating盒图：

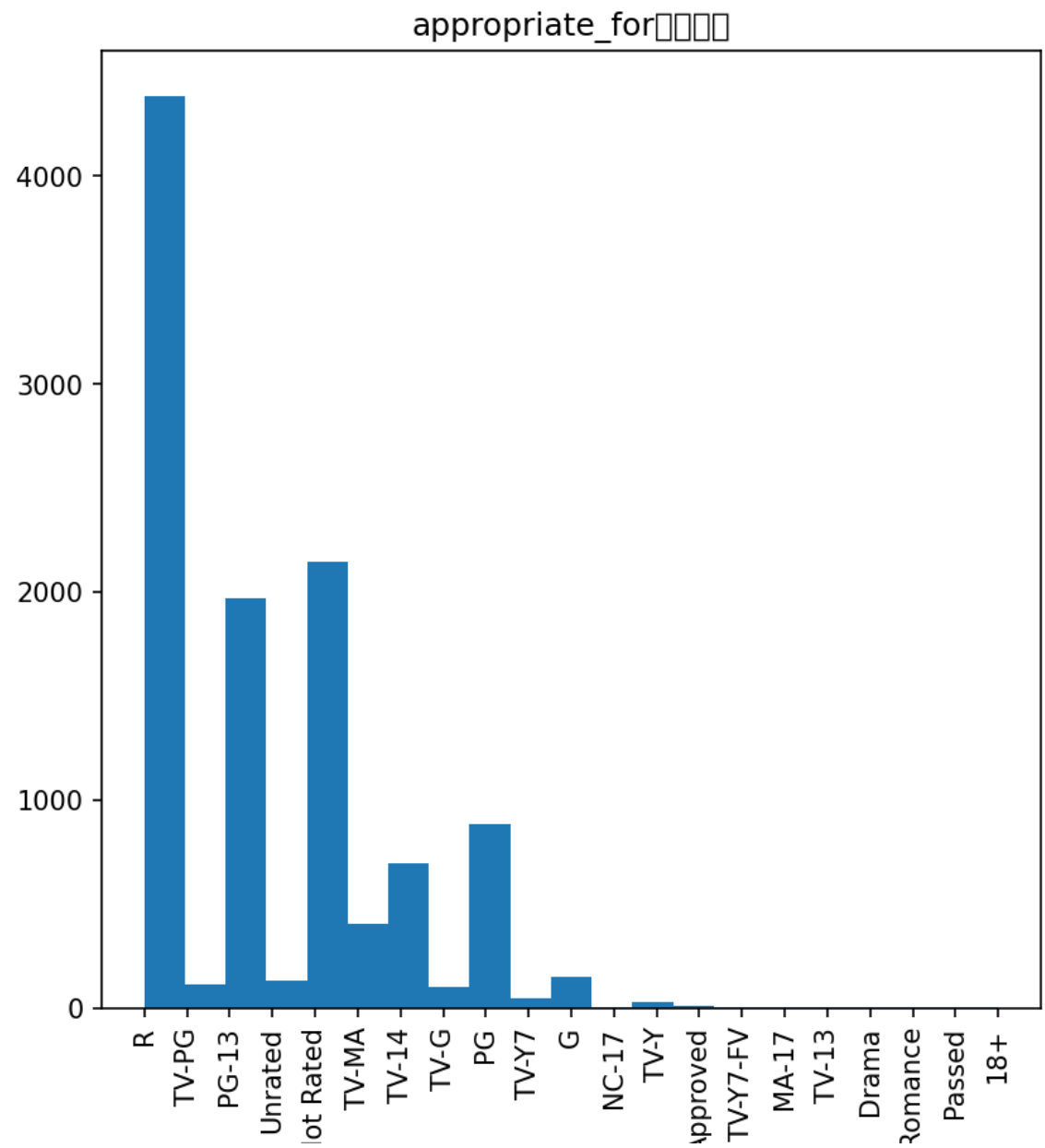


appropriate_for数据分析：

appropriate_for 的类型是： 标称属性
频数统计：

R	4384
Not Rated	2142
PG-13	1968
PG	886
TV-14	694
TV-MA	406
G	152
Unrated	132
TV-PG	115
TV-G	99
TV-Y7	45
TV-Y	25
Approved	9
NC-17	4
TV-Y7-FV	3
Passed	3
MA-17	1
TV-13	1
Drama	1
Drama, Romance	1
18+	1

appropriate_for直方图：



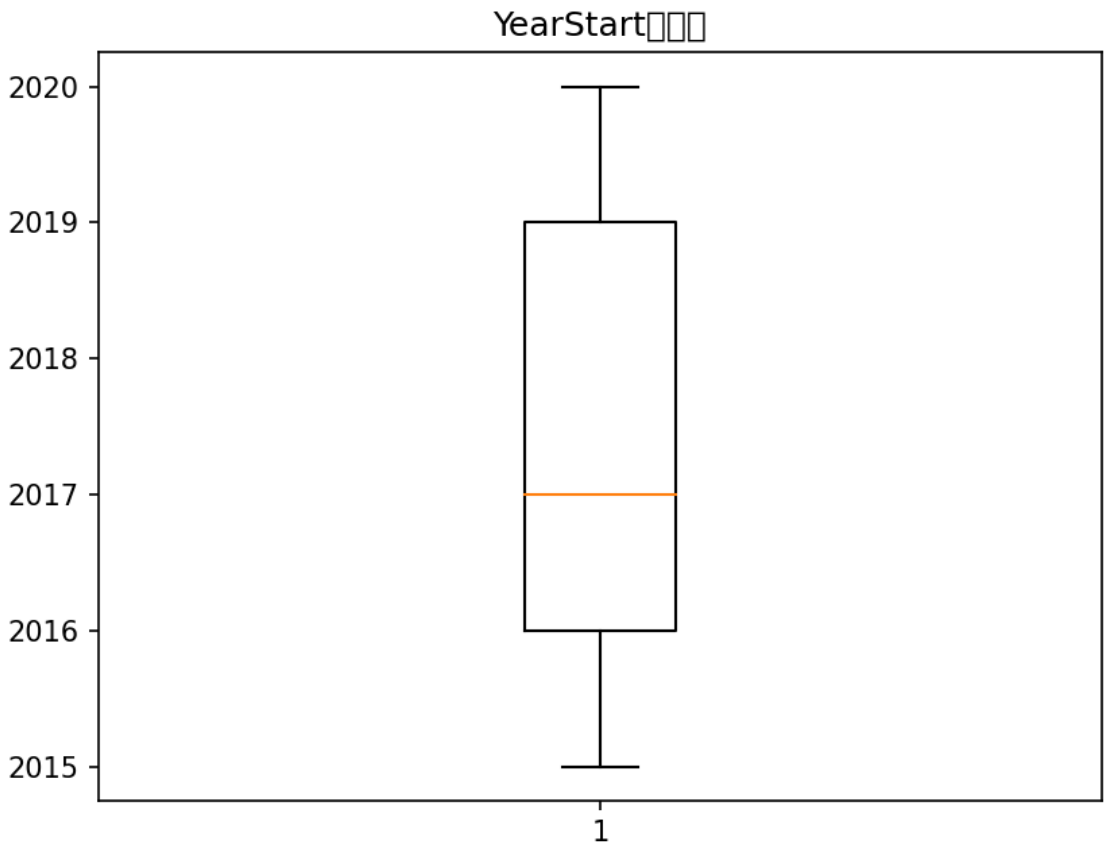
对于数据集Movies Dataset from Pirated Sites：

数值属性名称列表： ['YearStart', 'YearEnd', 'Data_Value', 'Data_Value_Alt', 'Low_Confidence_Limit', 'High_Confidence_Limit', 'LocationID']
标称属性名称列表： ['LocationAbbr', 'LocationDesc', 'Datasource', 'Class', 'Topic', 'Question', 'Data_Value_Unit', 'DataValueTypeID', 'Data_Value_Type', 'StratificationCategory1', 'Stratification1', 'StratificationCategory2', 'Stratification2', 'ClassID', 'TopicID', 'QuestionID', 'StratificationCategoryID1', 'StratificationID1', 'StratificationCategoryID2', 'StratificationID2']

YearStart数据分析：

YearStart 的类型是： 数值属性
五数概括：
min 2015.0
25% 2016.0
50% 2017.0
75% 2019.0
max 2020.0
Name: YearStart, dtype: float64
缺失值个数： 0

YearStart 盒图：



LocationAbbr 数据分析：

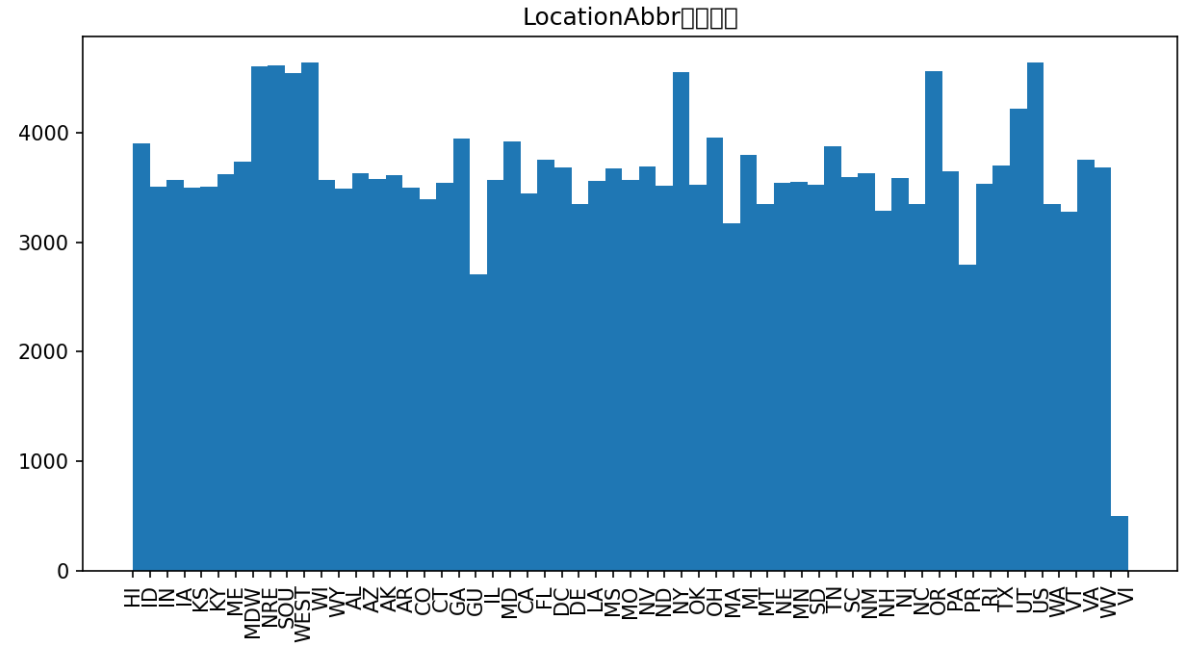
LocationAbbr 的类型是： 标称属性

频数统计：

US 4644
WEST 4638
NRE 4614
MDW 4611
OR 4565
NY 4557
SOU 4542
UT 4222
OH 3955
GA 3951
MD 3919
HI 3907
TN 3879
MI 3796
VA 3758
FL 3753
ME 3733
TX 3699
NV 3696
DC 3684
WV 3682
MS 3677
PA 3648
NM 3635

AL 3633
KY 3623
AK 3611
SC 3592
NJ 3589
AZ 3582
MO 3573
IL 3571
IN 3570
WI 3569
LA 3563
MN 3555
NE 3546
CT 3543
RI 3534
OK 3526
SD 3526
ND 3514
KS 3510
ID 3507
IA 3501
AR 3498
WY 3494
CA 3447
CO 3390
NC 3349
WA 3348
MT 3348
DE 3346
NH 3284
VT 3278
MA 3174
PR 2797
GU 2703
VI 503

LocationAbbr 直方图：



缺失值处理后的数据集不方便在报告中展示，运行代码后可获得合理结果（与原数据集相比）。

6.仓库地址

https://github.com/Smurflyiaa/-