收集

```
# 导入需要的库
import numpy as np
from aip import AipImageClassify
from bs4 import BeautifulSoup
import requests
import re
import pandas as pd
```

```
# 收集文件 1 保存为 twitter_archive_enhanced
# 读取数据
twitter_archive_enhanced = pd.read_csv('twitter-archive-enhanced.csv')

# 显示前两行
twitter_archive_enhanced.head(2)
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	source	text	retweeted_status_id	retweeted_status_user_id	retweeted
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	<a href="http://twitter.com/download/iphone" r...	This is Phineas. He's a mystical boy. Only eve...	NaN	NaN	NaN
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	<a href="http://twitter.com/download/iphone" r...	This is Tilly. She's just checking pup on you....	NaN	NaN	NaN

```
# 收集文件 2 保存为 tweet_json
# 显示数据前2行
tweet_json = pd.read_json("tweet_json.json", lines = True)

# 显示前两行
tweet_json.head(2)
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	contributors	coordinates	created_at	display_text_range	entities	extended_entities	favorite_count	favorited	full_text	geo	...	possibly_sensitive_appealable	quoted_status	quo
0	NaN	NaN	2017-08-01 16:23:56	[0, 85]	{'hashtags': [], 'symbols': [], 'user_mentions...	{'media': [{'id': 892420639486877696, 'id_str'...	39492	False	This is Phineas. He's a mystical boy. Only eve...	NaN	...	0.0	NaN	NaN
1	NaN	NaN	2017-08-01 00:17:27	[0, 138]	{'hashtags': [], 'symbols': [], 'user_mentions...	{'media': [{'id': 892177413194625024, 'id_str'...	33786	False	This is Tilly. She's just checking pup on you....	NaN	...	0.0	NaN	NaN

2 rows × 31 columns

```
# 收集文件 3 保存为 image_predictions
r = requests.get("https://raw.githubusercontent.com/udacity/new-dand-advanced-china/master/%E6%95%B0%E6%8D%AE%E6%B8%85%E6%B4%97/WeRateDogs%E9%A1%B9%E7%9B%AE/image-p

# # 新建空的文件image-predictions_byPythonDownload.tsv
fileobj = open("image-predictions_byPythonDownload.tsv", 'wb')

# # 将数据写入fileobj中
fileobj.write(r.content)
fileobj.close()

image_predictions = pd.read_csv("image-predictions_byPythonDownload.tsv", sep = '\t')
image_predictions.head(5)
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog	p2	p2_conf	p2_dog	p3	p3_c
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh_springer_spaniel	0.465074	True	collie	0.156665	True	Shetland_sheepdog	0.061
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1	redbone	0.506826	True	miniature_pinscher	0.074192	True	Rhodesian_ridgeback	0.072
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1	German_shepherd	0.596461	True	malinois	0.138584	True	bloodhound	0.116
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	1	Rhodesian_ridgeback	0.408143	True	redbone	0.360687	True	miniature_pinscher	0.222
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	1	miniature_pinscher	0.560311	True	Rottweiler	0.243682	True	Doberman	0.154

```
# 合并
```

评估

目测评估

```
# 目测评估三个数据集
```

twitter_archive_enhanced数据中有大量的空值
twitter_archive_enhanced数据中有些列是不需要用的比如 in_reply_to_status_id

tweet_json数据中包含了大量的空值

image_predictions数据中对狗狗品种认定，但是给出了三种结果，取其中最高的即可。

编程评估

```
# 使用 pandas 的各种方法评估三个数据集，比如 info value_counts 等

# 你需要添加更多的 code cell 和 markdown cell 来完成所有编程评估

twitter_archive_enhanced.info()

image_predictions.info()

tweet_json.info()

twitter_archive_enhanced['source'].unique()

twitter_archive_enhanced['name'].unique()

twitter_archive_enhanced['rating_denominator'].value_counts()

twitter_archive_enhanced['rating_numerator'].value_counts()
```

提示：

- 完成目测评估和编程评估之后，总结列出你发现的三个数据集中的所有问题；
- 每个问题都要有对应的一句话或几句话描述；
- 最终至少要包含 8 个质量问题和 2 个整洁度问题。

质量

`twitter_archive_enhanced` 表格

- 转发的(即retweets)的数据需要删除
- source 列应当只包含iphone web内容
- timestamp 列的时间数据不是datatime类型
- name列数据异常
- doggo floofer pupper puppo 列数据缺失
- rating_denominator列中有异常值（等于0）
- 没有图片的数据需要删除
- tweet_id应当为object类型

`tweet_json` 表格

`image_predictions` 表格

整洁度

- 将`favorite_count`和`retweet_count`合并到twitter_archive_enhanced数据集中
- doggo floofer pupper puppo 可以合并成一个列

清理

提示：

- 清理数据集之前需要先备份数据集；
- 按照下面示例的结构：**定义-代码-测试**，对提出的每个问题进行清洗。

```
# 备份三个数据集
twitter_archive_enhanced.to_csv("twitter_archive_enhanced.csv")
tweet_json.to_csv("tweet_json.csv")
image_predictions.to_csv("image_predictions.csv")
```

```
twitter_data_clean = twitter_archive_enhanced.copy()
tweet_json_clean = tweet_json.copy()
image_predictions_clean = image_predictions.copy()
```

问题描述一

定义

in_reply_to_user_id 有78条，这些数据是回复。将其删除。

代码

```
# 解决问题一的代码
df_temp = twitter_data_clean[twitter_data_clean["in_reply_to_user_id"].isnull()==False]
twitter_data_clean = twitter_data_clean.drop(index=df_temp.index)
```

```
# 删除不用的列
col = ['in_reply_to_status_id', 'in_reply_to_user_id', 'retweeted_status_id', 'retweeted_status_user_id', 'retweeted_status_timestamp']
twitter_data_clean = twitter_data_clean.drop(columns = col)
```

测试

```
# 测试问题一是否正确清理完成
twitter_data_clean.info()
```

问题描述二

定义

source列应当只包含iphone web等内容

代码

```
# 解决问题二的代码
# 重置index列
twitter_data_clean.reset_index(drop=True, inplace=True)
# twitter_archive_enhanced
```

```
'''
    解析出每一个html标记语言中的内容
    <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>
    提取出Twitter for iPhone
'''
# html = twitter_archive_enhanced['source'][1]
# t = BeautifulSoup(html, 'lxml')
# t.a.contents

for i in range(len(twitter_data_clean)):
    html = twitter_data_clean.loc[i, 'source']
    try:
        t = BeautifulSoup(html, 'lxml') # html转为BeautifulSoup
        twitter_data_clean.loc[i, 'source'] = t.a.contents[0]
    except:
        twitter_data_clean.loc[i, 'source'] = html
# twitter_archive_enhanced['source']
```

测试

```
# 测试问题二是否正确清理完成
twitter_data_clean['source'].unique()
```

问题描述三

定义

timestamp 列的时间数据不是datatime类型

代码

```
twitter_data_clean['timestamp'] = pd.to_datetime(twitter_data_clean['timestamp'])
```

检测

```
twitter_data_clean.info()
```

问题描述四

定义

name列数据异常。根据text列修补

- 使用正则表达式匹配名字，并且名字第一个字应该是大写.
- 如果text中没有写出名字那么用nan代替

代码

```
# 引用 https://blog.csdn.net/u010606346/article/details/84778363
twitter_data_clean['name'] = twitter_data_clean['text'].str.extract(r'(?:This is|named|Meet|Say hello to|name is|Here we have|Here is)\s([A-Z][^\s.,]*)')
```

测试

```
twitter_data_clean['name'].unique()
```

问题描述五

定义

doggo floofer pupper puppo等 列数据缺失

- 查询text中是否存在对应的单词（doggo floofer pupper puppo）
- 存在则放入新的列nickname中

代码

```
# twitter_archive_enhanced[twitter_archive_enhanced['text'].str.find('puppo')!=-1]
# nickname
for i in range(len(twitter_data_clean)):
    if twitter_data_clean.loc[i, 'text'].find("puppo")!=-1:
        twitter_data_clean.loc[i, 'nickname'] = "puppo"

    if twitter_data_clean.loc[i, 'text'].find("doggo")!=-1:
        twitter_data_clean.loc[i, 'nickname'] = "doggo"

    if twitter_data_clean.loc[i, 'text'].find("floofer")!=-1:
        twitter_data_clean.loc[i, 'nickname'] = "floofer"

    if twitter_data_clean.loc[i, 'text'].find("pupper")!=-1:
        twitter_data_clean.loc[i, 'nickname'] = "pupper"
```

检测

```
twitter_data_clean['nickname'].head(10)
```

问题描述六

定义

rating_denominator数据缺失

代码

```
temp = twitter_data_clean['text'].str.extract(r'(\d+\.?\d*\/\d+)')
temp = temp[0].str.split("/")

for i in range(len(twitter_data_clean)):
    twitter_data_clean.loc[i, 'rating_numerator'] = temp[i][0]
    twitter_data_clean.loc[i, 'rating_denominator'] = temp[i][1]

twitter_data_clean['rating_numerator'] = twitter_data_clean['rating_numerator'].astype("float64")
twitter_data_clean['rating_denominator'] = twitter_data_clean['rating_denominator'].astype("float64")
```

检测

```
twitter_data_clean.info()
```

问题七

定义

没有图片的数据需要删除

代码

```
# 合并twitter_archive_enhanced 和 image_predictions数据集合
twitter_data_clean = pd.merge(twitter_data_clean, image_predictions_clean, how = "left", on = "tweet_id")
twitter_data_clean.info()
```

```
# 删除没有图片的数据行
tempdata = twitter_data_clean[pd.isnull(twitter_data_clean['img_num']) == True]
twitter_data_clean = twitter_data_clean.drop(index = tempdata.index)
```

```
# 删除不需要的列

col = ['p2', 'p2_conf', 'p2_dog', 'p3', 'p3_conf', 'p3_dog']
twitter_data_clean = twitter_data_clean.drop(columns = col)
```

```
# 数据索引从新排列
twitter_data_clean.reset_index(drop=True, inplace=True)
```

检测

```
twitter_data_clean.info()
```

问题八

定义

tweet_id应当为object类型

代码

```
twitter_data_clean['tweet_id'] = twitter_data_clean['tweet_id'].astype("object")
```

检测

```
twitter_data_clean.info()
```

问题描述 九

定义

twitter_archive_enhanced补充每一条推特数据的转发数和点赞数

代码

```
'''
    补充twitter_archive_enhanced数据集中retweet_count列和favorite_count列
'''
rtweet_count_list = []
favorite_count_list = []
for i in range(len(twitter_data_clean)):
    id = twitter_data_clean.loc[i]['tweet_id']
    try:
        t1 = tweet_json[tweet_json['id'] == id]["retweet_count"]
        t2 = tweet_json[tweet_json['id'] == id]["favorite_count"]
        rtweet_count_list.append(t1.iloc[0])
        favorite_count_list.append(t2.iloc[0])
    except:
        rtweet_count_list.append(np.nan)
        favorite_count_list.append(np.nan)
# rtweet_count_list
# favorite_count_list
```

```
twitter_data_clean['favorite_count'] = favorite_count_list
twitter_data_clean['retweet_count'] = rtweet_count_list
twitter_data_clean[['favorite_count', 'retweet_count']].head(2)
```

检测

```
# 查询id=890729181411237888的推特数据是否相等
print(tweet_json[tweet_json['id'] == 890729181411237888]['favorite_count'])
print(twitter_data_clean[twitter_data_clean['tweet_id'] == 890729181411237888]['favorite_count'])
```

```
# 查询id=890729181411237888的推特数据是否相等
print(tweet_json[tweet_json['id'] == 890729181411237888]['retweet_count'])
print(twitter_data_clean[twitter_data_clean['tweet_id'] == 890729181411237888]['retweet_count'])
```

```
twitter_data_clean.info()
```

问题描述 九（补充）

定义

favorite_count 列和retweet_count中有几个NaN

- 用均值填充

代码

```
favorite_count_mean = twitter_data_clean['favorite_count'].mean()
retweet_count_mean = twitter_data_clean['retweet_count'].mean()
```

```
twitter_data_clean['favorite_count'].fillna(favorite_count_mean, inplace = True)
twitter_data_clean['retweet_count'].fillna(retweet_count_mean, inplace = True)
```

检测

```
twitter_data_clean.info()
'''
favorite_count                2050 non-null float64
retweet_count                 2050 non-null float64
2050 - > 2052
'''
```

问题描述 十

定义

doggo floofer pupper puppo 可以合并成一个列

- 已经合并，直接删除doggo floofer pupper puppo列

代码

```
twitter_data_clean = twitter_data_clean.drop(columns=["doggo", "floofer", "pupper", "puppo"])
```

检测

```
twitter_data_clean.info()
```

存储清理后的主数据集

```
 twitter_data_clean.to_csv("twitter_data_clean.csv")
```