数据清洗流程

---

1. **获取数据**

- 利用pandas库中read_csv方法读取文件twitter-archive-enhanced.csv
- 利用requests库通过编程方法下载文件image_predictions.tsv
- 利用json模块提取udacity给定文件tweet_json.txt中的相关信息，存入DataFrame

2. **评估数据**

- 通过对文件twitter-archive-enhanced.csv, image_predictions.tsv, tweet_json.txt进行目测评估与编程评估
- 经过不断的尝试与总结，发现对于一个数据集的评估，应该首先了解每个变量在现实世界中的真实意义，才能很好的理解/发现数据中存在的异常
- 对于一个表格，首先应该利用info()方法，观察变量是否有合适的数据类型，比如数值类变量是否为int或float类型，时间类变量是否为datetime类型，id类变量（需要身份标识的）不能为数值类型，以防止首位为0的情况下被省略
- 同样地，善于利用describe() 及 sort_values()很容易发现数值型变量中的异常值（通常为极大或极小到不符合常理）；利用value_counts()方法很容易找到str变量下的一些异常类别
- 通过sample(), head(), tail()等方法可以随机的抽查一些变量，说不定可以发现一些比较unique的问题，针对性解决

经过实际的目测评估与编程评估，数据集中存在的问题为：

质量

`twitter_archive_enhanced` 表格

- tweet_id不能是整型，应该为字符串
- name存在错误数据记录，'an'可能本应该代表英文名字‘Ann’
- timestamp 格式错误，应为时间格式
- expanded_urls存在多种格式（字符串和超链接）
- retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp部分数据缺失严重（无法清理）
- rating_numerator, rating_denominator存在字符串格式的数据，且部分评分为0
- rating_numerator 存在异常数据，最大值1776

`image_predictions` 表格

- tweet_id不能是整型，应该为字符串

清洁度

- `twitter_archive_enhanced` 表格中变量in_reply_to_status_id、in_reply_to_user_id多余
- `twitter_archive_enhanced` 表格中doggo、floofer、pupper、puppo内容多余
- `tweet_json` 中的内容retweet_count和favorite_count应归于tweet_archive_enhanced表格中
- `image_predicitons` 表格内容应合并twitter_archive_enhanced表格中，但有些数据不包含图片，因此采用inner的合并方法取其交集

3. 清理数据

按照Define、Code、Test三个步骤，对数据评估部分所发现的问题逐个解决，分为以下几大部分：

- 清除不合格数据
- 数据格式转换
- 清除多余变量
- 利用正则表达式重新提取评分
- 利用正则表达式重新提取评级
- 缺失及异常值处理
- 利用正则表达式重新提取名字
- 内容合并
- 再次迭代验
- 将清理后的数据保存到新的csv文件内

数据清理部分心得：

- 在清理数据初期可能会对pandas库内的很多数据清理工具不够熟悉，从而导致无从下手，个人总结为及时google、baidu、stack overflow往往都能找到想要的答案；
- 正则表达式在从非结构化数据中提取结构化数据时，往往可以发挥巨大的作用，在今后的学习过程中可以逐渐熟悉这个工具；
- 在进行数据清理的操作后再次做测试很重要，因为有些时候算法可能并没有按照预期的情况运行。

  —— 例如value_counts()方法显示了某列中有部分None，随后用replace(None, 'A')去替换，但是其实原始数据中的None可能是字符串类型的'None'，并不是python中的关键字None或者Numpy中的np.nan类型，当再次迭代时就会发现问题并没有被解决。