狗狗数据集清洗过程

从清洗针对的问题类型来看，项目中的清洗按照这样的顺序进行：

1. 处理缺失值问题
2. 处理重复值问题
3. 处理数据集结构问题
4. 处理除了缺失值、重复值之外的数据集内容质量问题

对单个清洗动作来说，一般遵循着定义-编码-测试的流程。
下面是关于数据清洗过程的叙述。

处理缺失值

df_basic 表

- 有大量缺失值的列或删除整列，或过滤无效记录

处理重复值

df_pred 表

- `jpg_url` 列数据有重复，进行去重处理

处理结构问题

- 首先，三个数据集本来是对同一对象进行观察研究，根据对「数据整洁性」的定义，数据不应保存在三个表中，而应保存在同一表中。因此借助 `merge()` 函数将三个表合并为一个表
- 另外，数据集中还有些结构问题，比如推文内容和链接混杂、狗狗地位分隔四列等，项目中一般借助 pandas.Series 的 str 属性，对这些文本信息进行处理

处理其他内容质量问题

处理质量问题，遵循着定义-编码-测试的一般性过程。合并后的数据集存在很多源数据带来的质量问题，具体清洗过程如下：

- 将 `expanded_urls` 列的链接去重：该列中某些记录存在链接重复，比如某个记录中有两条一模一样的链接，处理方法是写一个对字符串的子串进行去重的函数，然后apply到Series中去
- 处理 `rating_numerator` 和 `rating_denominator` 列的异常值：项目外这两列是进行过初步清洗的，但是清洗存在一些疏漏，比如：忽略了浮点数评分、错误抓取了类似 7/11 这样的数据，在项目中对此进行了纠正
- 提取 `source` 列中元素内容并舍弃其余部分：这里用的是 pandas.Series.str 中的 replace() 方法，将无用内容识别处理再以空字符替换
- 将 `timestamp` 列时间戳数据改为字符串格式：使用了 pandas.to_datetime() 函数
- 重新提取 `name` 列狗狗的名称：项目外这列也是进行过初步清洗的，但是存在一些 'a'、'the' 这样的结果。原因是使用正则匹配模式的时候忽略了某些特殊情况。在项目中我仔细观察了这些模式，并反复测试，剔除了特殊情况下的匹配。这里使用了 re 模块，而不是 pandas.Series.str 中的方法
- 过滤掉 `retweeted_status_id` 列指示的转发数据：通过布尔数组过滤
- `tweet_id` 列数据改为字符串格式：借助 Series.astype() 处理