数据清理报告

一、数据说明

此次 WeRateDogs 推特数据分析用到三个数据集：

1. 主体数据集 WeRateDogs 的推特档案，载入为 DataFrame: twt_archive；
2. 补充数据：推特图像的预测数据，载入为 DataFrame: img_pre；
3. 补充数据：用 Tweepy 获取的推特补充数据，主要信息是转发数和喜爱数，载入为 DataFrame: twt_json。

二、数据评估过程

（一）目测评估

通过目测初步评估数据，了解三个数据集的结构和关系：以 twt_archive 为目标数据集，将 img_pre 所有信息和 twt_json 的转发数（retweet_count）、喜爱数（favorite_count）并入。

同时也发现：

1. twt_archive: name 可能有无效名字；
2. twt_archive & twt_json: source 列 lmxl 可以去掉标签，只保留来源类型信息
3. twt_archive: rating_numerator 和 rating_denominator 两列都为了描述评分信息，可以直接保留处理后的评分；部分 rating_numerator 和 ating_denominatorr 不正确，需重新提取；并且 rating_denominator 不统一，不利于比较分析，统一 rating_denominator 后可以得到标准化的评分信息 rating；
4. twt_archive: doggo, floofer, pupper, puppo 都是描述狗的“地位”，应合并为一列，其中，部分数据含有两个狗的“地位”的信息。

（二）编程评估

通过编程评估，除了进一步核实目测评估发现的问题，还进一步发现：

1. twt_archive: tweet_id, in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, timestamp, retweeted_status_ imestamp 的数据类型有误；
2. twt_archive: 一条数据的 rating_denominator 为 0，评分不合理；
3. img_pre: tweet_id, img_num, p1, p2, p3 的数据类型有误；
4. twt_json: id 和 id_str 都表达同样的信息，保留一列，且 id 为字符串格式。

三、数据处理过程

在数据处理过程中，主要用到几个核心方法：

1. 更改数据格式：pd.Series.astype()；
2. 删除多余的列：pd.DataFrame.drop(columns=[], inplace=True)；
3. 从一列中的单元格中提取部分文本：pd.Series.str.extract()；
4. 用某个函数/方法处理 DataFrame：pd.DataFrame.apply(fun, axis=1/0)；
5. 合并两个 DataFrame：pd.DataFrame.merge(DataFrame, how=’left’,

on=’’)

四、数据分析和可视化

此次分析探讨的问题：

1. 评分、转发数、喜爱数的分布;
2. 转发数、喜爱数与评分的关系;
3. 评分与地位的关系;
4. 转发数、喜爱数与地位的关系;
5. 转发数与喜爱数的关系。

分析和可视化方法：

1. 单一变量统计数据分析：pd.Series.describe();
2. 两个变量统计数据分析：pd.Series.groupby().describe();
3. 单一变量可视化分析：直方图、箱型图;
4. 两个变量统计数据分析：散点图、箱型图。