WeRateDogs 推特数据整理过程

一， 收集数据

1， 资料来源：手头文件

直接用 pandas 包的 read_csv 读取。

```
twitter_archive = pd.read_csv("twitter-archive-enhanced.csv")
```

2， 资料来源：通过 API 获取

通过 twitter 的 api 获取数据，首先在 https://apps.twitter.com/app/new 中创建申请，创建后单击 Keys and Access Tokens，然后单击 Create my Access token 按钮。

之后就可以导入 tweepy 包，并按以下格式通过 API 获取 twitter 数据，将获取的数据存入 tweet_json.txt。

```
import tweepy
import json as js
```

```
CONSUMER_KEY = "**************"
CONSUMER_SECRET = "**************
OAUTH_TOKEN = "1**************"
OAUTH_TOKEN_SECRET = "**************"
```

```
auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(OAUTH_TOKEN, OAUTH_TOKEN_SECRET)
api = tweepy.API(auth,wait_on_rate_limit=True,wait_on_rate_limit_notify=True)
none_id = []
with open("tweet_json.txt",'a') as f:
    for id in twitter_archive["tweet_id"]:
        try:
            tweet = api.get_status(id,tweet_mode='extended')._json
            tweet = js.dumps(tweet)
            f.write(tweet)
            f.write("\n")
        except:
            none_id.append(id)
```

导入 json 包，使用 json.loads 函数逐条解析 json 数据，将 tweet_id,retweet_count,favorite_count 数据存入 pandas.DataFrame。

3， 资料来源：从互联网下载文件

导入 request 包，使用提供的 url 使用 request 的 get 函数获取 url 内容，存入 image-predictions.tsv 文件中，使用 pandas 的 read_csv 函数并将分隔符设置为“\t”读取数据为 DataFrame。

二， 评估数据

1，在 pandas 中直接读取数据集，进行目测评估。

2， 在 Excel 中打开数据集，进行目测评估。

3，使用 pandas 的 info()函数获取列名，数据集数量，缺失值数量，数据类型，数据集大小等信息。

4， 使用 duplicated()功能查找是否存在重复值。

5， 使用 value_counts()函数查看某一列值的信息。

6， 随机查看某些具体的值。

7，将观察到的问题整理记录。

质量

twitter_archive 表格

- tweet_id是整型，而不是字符串
- in_reply_to_status_id和in_reply_to_user_id为float数据类型，应为str。
- in_reply_to_status_id和in_reply_to_user_id为回复的Twitter，应删除。
- retweeted_status_id和retweeted_status_user_id和retweeted_status_timesstamp为转发的Twitter应删除。
- retweeted_status_id和retweeted_status_user_id和retweeted_status_timesstamp为float数据类型，应为str。
- 存在一组数据的rating_denominator = 0,0为分母。该条数据正好也是回复微博，故可以直接删除。
- 在通过API获取数据时，有几组tweet_id获取不到数据，原twitter也查找不到，应删除。
- text的内容后面存在链接内容
- 存在同一个狗狗有多个地位的情况

tweet_json 表格

- 因tweet_archive中存在回复转发的twitter和非@dog_rates发的twitter，应将这些数据同样处理，进行删除。

image_predictions 表格

- tweet_id是整型，而不是字符串
- 数据集数量小于twitter_archive，存在无图片的twitter。

清洁度

- twitter_archive应将rating_numerator/rating_denominator计算出具体评分。
- 应将三个数据集依据tweet_id进行合并
- twitter_archive应将 doggo，floofer，pupper，puppo合并到一列

三， 清理数据

1. 使用 astype()将数据类型转换为合适的数据类型。

```
twitter_archive_clean.tweet_id = twitter_archive_clean.tweet_id.astype(str)
```

```
image_predictions_clean.tweet_id = image_predictions_clean.tweet_id.astype(str)
```

2. 筛选排除不符合要求的数据。

```
twitter_archive_clean = twitter_archive_clean[twitter_archive_clean.in_reply_to_status_id.isnull()]
```

3. 使用 drop()删除不符合要求的数据。

```
index = twitter_archive_clean[twitter_archive_clean.retweeted_status_id.notnull()].index
```

```
index
```

```
Int64Index([  19,   32,   36,   68,   73,   74,   78,   91,   95,   97,
            ...
            926,  937,  943,  949, 1012, 1023, 1043, 1242, 2259, 2260],
           dtype='int64', length=181)
```

```
twitter_archive_clean.drop(index, inplace = True)
```

```
none_id
```

```
[888202515573088257,
 873697596434513921,
 869988702071779329,
 866816280283807744,
 861769973181624320,
 845459076796616705,
 842892208864923648,
 837012587749474308,
 827228250799742977,
 802247111496568832,
 775096608509886464,
 754011816964026368]
```

```
for i in none_id:
    index = twitter_archive_clean[twitter_archive_clean['tweet_id'] ==i].index
    twitter_archive_clean.drop(index,inplace = True)
```

4. 使用 apply() 和 split()清除不必要的文本。

```
f = lambda x :x.split(' https://')[0]
twitter_archive_clean.text = twitter_archive_clean.text.apply(f)
```

5. 向数据集中通过运算添加需要的列。

```
twitter_archive_clean['rating'] = twitter_archive_clean.rating_numerator/twitter_archive_clean.rating_denominator
```

6. 使用 merge()方法合并数据集。

```
twitter_archive_clean=pd.merge(twitter_archive_clean,tweet_json_clean,on = 'tweet_id',how = 'left')
```

```
twitter_archive_clean=pd.merge(twitter_archive_clean,image_predictions_clean,on = 'tweet_id')
```

7. 将同属性的数据合并到一列，并删除不需要的列。

```
twitter_archive_clean.doggo.replace('None','',inplace = True)
```

```
twitter_archive_clean.floofer.replace('None','',inplace = True)
```

```
twitter_archive_clean.pupper.replace('None','',inplace = True)
```

```
twitter_archive_clean.puppo.replace('None','',inplace = True)
```

```
twitter_archive_clean['stage'] = (twitter_archive_clean.doggo+twitter_archive_clean.floofer+
                                  twitter_archive_clean.pupper+twitter_archive_clean.puppo)
```

```
twitter_archive_clean['stage'].value_counts()
```

```
              1668
pupper         201
doggo           63
puppo           22
doggopupper      8
floofer          7
doggofloofer     1
doggopuppo       1
Name: stage, dtype: int64
```

```
twitter_archive_clean['stage'].replace(['doggopupper','doggofloofer','doggopuppo',''],
                                       ['doggo,pupper','doggo,floofer','doggo,puppo',np.nan],inplace = True)
```

```
twitter_archive_clean = twitter_archive_clean.drop(['doggo', 'floofer', 'pupper', 'puppo'],axis = 1)
```

四， 存储数据

将数据整理后的数据存储到 twitter_archive_master.csv。

```
twitter_archive_clean.to_csv("twitter_archive_master.csv",index = False)
```

使用 pandas 的 read_csv()功能读取数据集便可进行分析。