UDACITY

< Return to "Data Analyst Nanodegree" in the classroom

Wrangle and Analyze Data

审阅

HISTORY

Meets Specifications

针对上一次审阅中提出的问题和建议进行了修改，非常棒！恭喜你完成项目！
在下方的审阅中又提了一些小建议，希望能对以后的学习和工作有所帮助~
数据整理是现实世界中数据分析师工作的一大部分，完成这个项目的你已经具备相当强的数据分析实战经验！
当然，我们完成数据整理的目的是进行数据分析，接下来，我们将学习探索性数据分析，在整理后的数据集中满足好奇心！
继续加油吧！期待你的下一个项目！
Learning by doing!

代码功能与可读性

Jupyter Notebook 具有直观清晰，易于遵循的逻辑结构。 该代码包含清晰有效的注释，并应用在 Jupyter Notebook 的 Markdown 单元格中。 数据清洗过程（即收集、评估和清理）的步骤也通过注释或 Markdown 单元格被清晰地标识出来。

项目中的所有代码都包含在名为 wrangle_act.ipynb 的 Jupyter Notebook 中，并且代码在运行时没有出现错误。

收集数据

学员成功收集数据：

- **在"项目详细信息"页面上使用至少三（3）个不同来源。**
- **在"项目详细信息"页面上，使用至少三（3）种不同的文件格式。**
  首先将每一条数据导入到一个单独的 pandas 数据框中。

评估数据

学员评估使用了两种方式：

- **可视化评估：每张收集的数据都显示在 Jupyter Notebook 中，以便进行可视化评估。 一旦显示出来，数据可以在外部应用程序（如 Excel，文本编辑器）中进行评估。**
- **编程评估：使用 pandas 的功能和/或方法来评估数据。**

学员能够检测到至少 八（8）个数据质量问题和两（2）个整洁度问题，包括待清理问题以满足项目要求。每一个问题用一到几句话记录下来。

清理数据

清理过程中的定义，编码和测试步骤都有明确的记录。

在清理之前，保存原始数据的副本。

（如果可能的话）评估阶段确定的所有问题都可以通过 Python 和 pandas 成功清理，并包括满足项目要求所需的清理任务。

学员需要创建一个整洁的主数据集（或者多个数据集，如果有必要的话）与所有收集的数据片段。

本次提交对报告中的代码进行了详细的解释，做的不错！

其他建议

- 对于这个数据集而言，其实评分是很重要的数据，其中也存在很多问题，我们可以具体了解一下出现质量问题的原因是什么，根据那个原因编程清理问题。在这个数据集中，评分存在以下问题：
  1）分子是小数，但是只提取了小数点后面的数字的情况，比如 11.26/10，提取为了 26/10；
  2）多只狗狗评的总分：99/90，规律是：分母是10的N倍，且分子可以被 N 整除；
  3）同一个推特中存在两处分数形式的数字，提取的是第一个，但是可能第二个才是正确的：`@jonnysun @Lin_Manuel ok jomny I know you're excited but 960/00 isn't a valid rating, 13/10 is tho`
  4）比较单独的错误，比如 24/7 指的是7天 24 小时，并不是一个评分，这条推文中也没有具体的评分，可以 drop 掉；
  我们可以使用正则表达式重新提取，减少质量问题：

```
# 提取出的分子是带有小数点的，分母是10的倍数
rating = final_Data_clean.text.str.extract('((?:\d+\.)?\d+)\/([1-9]+[0]+)', expand = True)
# 提取出来的结果是个 dataframe 数据集，有两列，分别命名为分子和分母
rating.columns = ['rating_numerator', 'rating_denominator']
# 用新提取的数据替换掉原有的数据，记得修改分子的类型
final_Data_clean['rating_numerator'] = rating['rating_numerator'].astype(float)
final_Data_clean['rating_denominator'] = rating['rating_denominator']
```

  当然，这样重新提取后，还是可能存在一个推文中有多个评分的情况，针对有多个分数的情况，因为具体涉及到的问题不同，我们可以使用 findall 找出所有 text 中有多个分数的数据，然后将其筛选出来，查看具体是哪种情况，单独处理：

```
# 检查提取了两个 score 的情况
final_Data_clean['scores'] = final_Data_clean.text.str.findall('((?:\d+\.)?\d+\/[1-9]+[0]+)')
final_Data_clean['score_counts'] = final_Data_clean['scores'].apply(lambda x: len(set(x)))
final_Data_clean[final_Data_clean['score_counts']>1][['text','scores','score_counts']]
```

- name 的清理，其实现在清理的都是小写的名字，那么可以直接判断哪些行是小写，然后进行替换：

```
# 当 name 为小写或者 name 为 None 的时候，将这些数据的 name 修改为 np.nan
mask = (final_Data_clean.name.str.islower())|(final_Data_clean.name == 'None')
final_Data_clean.loc[mask, 'name'] = np.nan
```

有用链接

- Working with Text Data
- Pandas docs: Extracting Substrings
- pandas.Series.str.extract
- 【分享】项目可能会用到的正则表达式元字符、语法

存储并处理清洁过的数据

学员将他们收集、评估和清理过的主数据集保存到 CSV 文件或 SQLite 数据库中。

项目代码中，将收集、清理、整合后的主数据集保存到了一个 CSV 文件中。
这里有一点小问题，如果我们读取这个清理后的 CSV 文件，会发现出现一列 `'Unnamed: 0'` 。

要解决这个问题，可以参考 Stack Overflow 的解答.

```
twitter_archive_master= pd.read_csv("twitter_archive_master.csv")  #读取整理好的数据集中的数据
twitter_archive_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1628 entries, 0 to 1627
Data columns (total 27 columns):
Unnamed: 0                  1628 non-null int64
tweet_id                    1628 non-null int64
timestamp                   1628 non-null object
source                      1628 non-null object
text                        1628 non-null object
retweeted_status_id         0 non-null float64
retweeted_status_user_id    0 non-null float64
retweeted_status_timestamp  0 non-null float64
expanded_urls               1628 non-null object
rating_numerator            1628 non-null int64
rating_denominator          1628 non-null int64
name                        1166 non-null object
jpg_url                     1628 non-null object
img_num                     1628 non-null int64
p1                          1628 non-null object
p1_conf                     1628 non-null float64
p1_dog                      1628 non-null bool
p2                          1628 non-null object
p2_conf                     1628 non-null float64
p2_dog                      1628 non-null bool
p3                          1628 non-null object
```

使用 Jupyter Notebook 中的 pandas 或 SQL 分析主数据集，并生成至少三（3）个独立的结论。

在 Jupyter Notebook 中，使用 Python 绘图库或在 Tableau 中至少生成一（1）个标记的可视化对象。

学员必须在他们的清洗数据中明确他们之后分析和可视化所依据的数据是建立在评估和清理的基础上。

像狗的地位 stage 这样的列，可能一行数据中包含了不只一个值，对于这样的数据，有一些特殊的处理方法：将这一列中的多个值进行拆分，形成多行数据，虽然这样可能会产生 tweet_id 的重复项，但是这个拆分后的新数据集，却可以单独用来分析地位相关的聚合统计：

```
# 你可以打印一下每一步的值，看看每一步的操作产生了什么效果
split = twitter_archive_master['stage'].str.split('-',expand=True).stack()
split_reset = split.reset_index(level=1,drop=True).rename('stage')
df_stage = twitter_archive_master.drop('stage', axis=1).join(split_reset)
```

参考链接

- Pandas: 如何将一列中的文本拆分为多行?
- pandas visualization
- 《利用Python进行数据分析·第2版》第10章 数据聚合与分组运算

报告

学员需要言简意赅地介绍他们的数据清理。 这一文件（wrangle_report.pdf）大约只需要300-600字。

学员发现至少三（3）个结论，其中至少包含一个（1）可视化。这一文件（act_report.pdf）至少需要 250 个字。

项目文件

提交的文件包括如下：

- wrangle_act.ipynb
- wrangle_report.pdf
- act_report.pdf

并包括所有的数据集文件，如存储的主数据集，并使用在项目提交页面中指定的文件名和扩展名。

下载项目

返回 PATH