UDACITY

< Back to Machine Learning Engineer (Basic)

# Investigate Movie Dataset

审阅

代码审阅

HISTORY

## Meets Specifications

*"我对数据分析整个流程环节的理解正确吗？有没有缺少什么步骤？"*

这两个问题可以一起回答, 首先, 我觉的你的理解很好, 并没有缺什么步骤, 事实上, 我们项目设计呈现的流程基本就是完整的数据分析流程.

对于不同任务当然就会有所侧重, 如果对于机器学习, 你数据整理阶段会更加侧重于如何处理缺失, 让数据的分布更合理, 从而让模型能够更好地学习, 这个时候数据的可视化主要是研究特征分布啊, 时间序列的规律啊, 相关性, 是否有聚类啊, 甚至利用算法来得到特征重要性等

数据分析则更注重展示, 这就要求对数据更多的进行筛选, 整理等, 以提取出一些我们关心的统计学上的量化指标, 或者发掘数据的某种特点, 来支撑你的一些论点.

*"有没有系统的介绍特征工程的资料或者教材？"*

The Elements of Statistical Learning

特征工程其实更多靠经验, kaggle会有很多数据分析以及特征工程的例子, 就是别人经验的宝库, 一般都会集中在 kernels 这一块进行分享, 比如说, 这里是一个目前活跃比赛的一个数据分析与构建预测模型的例子: https://www.kaggle.com/willkoehrsen/start-here-a-gentle-introduction

这类的kernels有很多, 模仿他们可以让你学得很快, 另外kaggle blog 也是一个很好的站点

### 数据导入与处理

**学生成功导入所需要的库，并通过 `Pandas` 读取tmdb-movies.csv 中的数据。**

学生成功完成 5 个 tasks。

学员对数据表中的空值进行了处理。

## 根据指定要求读取数据

完成3个简单读取的tasks

学员完成逻辑读取中的2个任务

学员使用 Groupby 命令完成2个分组读取任务

## 绘图与可视化

对 popularity 最高的20名电影绘制其 popularity 值。

lovely!

分析电影净利润（票房-成本）随着年份变化的情况，并简单进行分析。

选择最多产的10位导演（电影数量最多的），绘制他们排行前3的三部电影的票房情况，并简要进行分析。

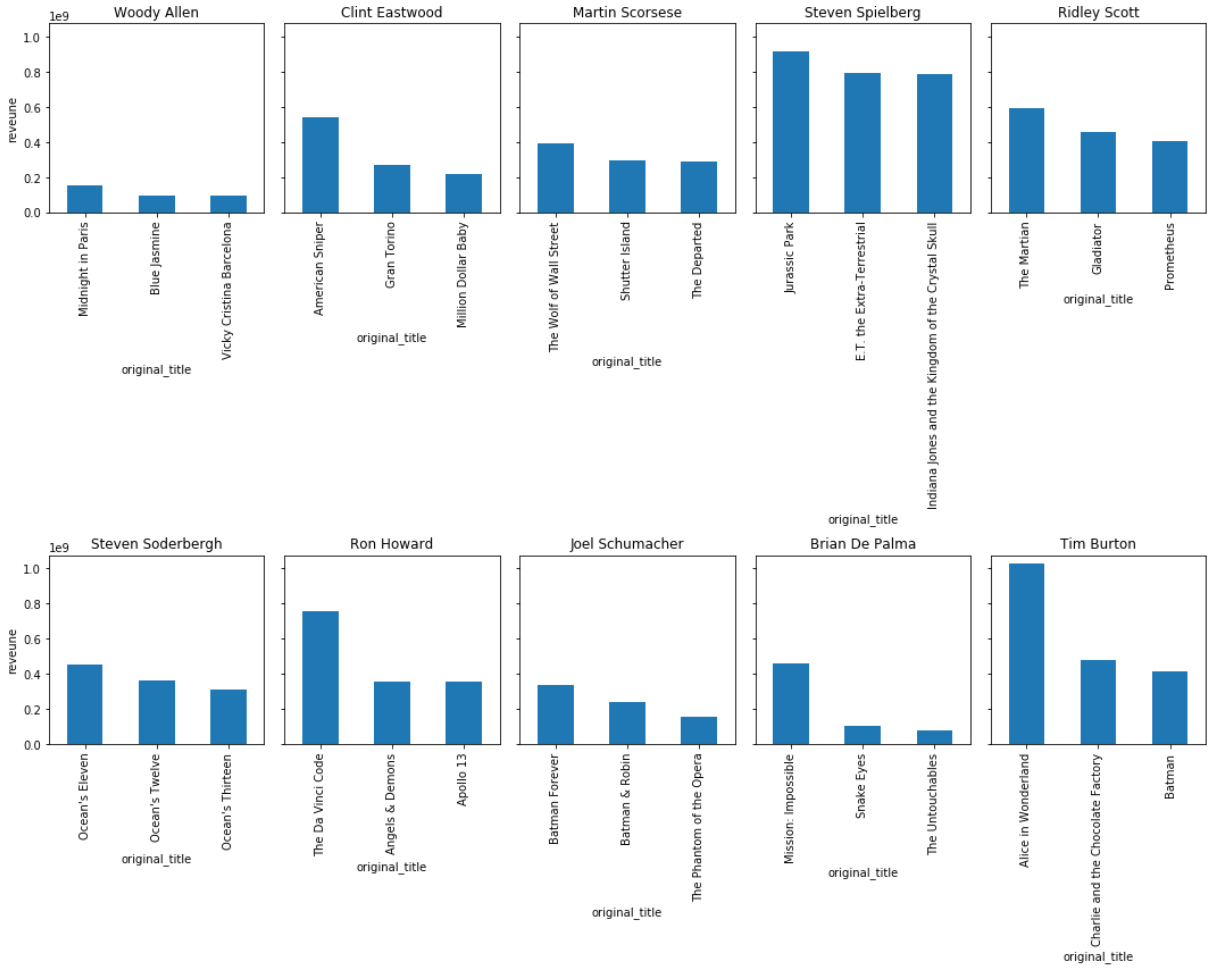
### 参考

以下是一个参考做法, 注意请不要公开这些代码

```
# 部分电影有多个导演, 用 '|' 分开
tmp = movie_data['director'].str.split('|', expand=True).stack().reset_ind
ex(level=1, drop=True).rename('director')
movie_data_split = movie_data[['original_title', 'revenue']].join(tmp)
```

```
# 得到目标数据，首先筛选全部满足要求的导演的电影，再按按收益排序
directors_top10 = tmp.value_counts().nlargest(10).index
target_data = movie_data_split[movie_data_split['director'].isin(directors
_top10)].sort_values('revenue', ascending=False)

# 作图
_, axes = plt.subplots(2, 5, figsize=(15, 12), sharey=True, tight_layout=T
rue)
for key, ax in zip(directors_top10, axes.flat):
    target_data[target_data['director']==key].set_index('original_title')
[:3].plot(kind='bar', ax=ax, legend=False)
    ax.set_ylabel('reveune')
    ax.set_title(key)
```




*提示: 第一行代码是拆分多个导演的, 这个博客能够帮助你理解*

**分析1968年~2015年六月电影的数量的变化。**

**分析1968年~2015年六月电影 Comedy 和 Drama 两类电影的数量的变化。**

## 参考

分享一下我的做法:

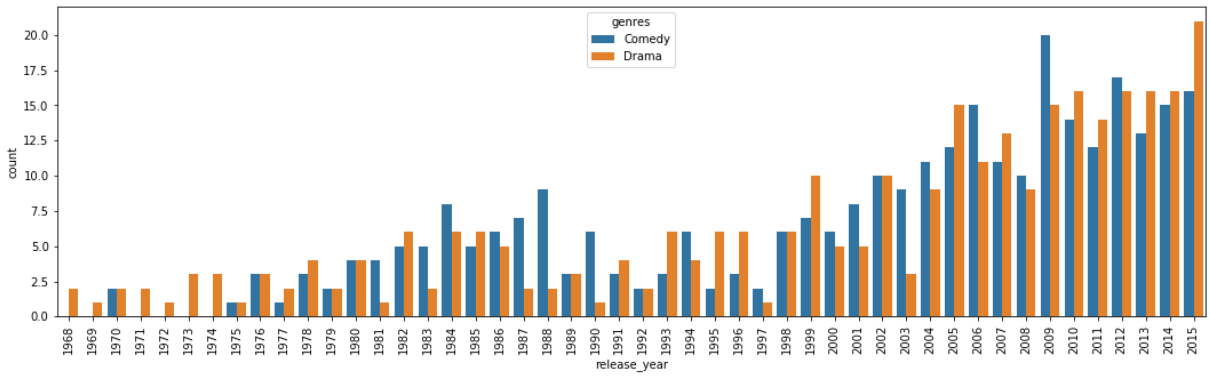
```
df_genres = movie_data.drop('genres', axis=1).join(movie_data['genres'].st
r.split('|', expand=True).stack().reset_index(level=1, drop=True).rename(
'genres'))

# 筛选条件1 年份
sel_year = df_genres['release_year'].between(1968, 2015)

# 筛选条件2 六月
sel_June = pd.to_datetime(df_genres['release_date']).dt.month == 6

# 筛选条件3 类型
sel_genre = df_genres['genres'].isin(['Drama', 'Comedy'])

# 筛选数据并作图(参考逻辑读取部分)
plt.figure(figsize=[18, 5])
sb.countplot(data=df_genres[sel_year&sel_June&sel_genre], x='release_year'
, hue='genres')
plt.xticks(rotation=90)
```




下载项目

返回 PATH

审阅者 FAQ Reviewer Agreement