TMDB电影推荐数据分析--作业

项目介绍

本项目使用'movie_metadata.csv'数据集，
该数据集记录了IMDB网站爬取到的5043条电影数据，包含28个变量。
这些电影来自于66个国家，拍摄时间横跨100年，出自于2399位独特的电影导演，展现了数以千计的男女演员的风采。

每条记录包含的28列信息如下所示：

- **color**：画面颜色
- **director_name**：导演姓名
- **num_critic_for_reviews**：评论家评论的数量
- **duration**：电影时长
- **director_facebook_likes**：脸书喜欢该导演的人数
- **actor_3_facebook_likes**：脸书上喜爱3号男演员的人数
- **actor_2_name**：男二号姓名
- **actor_1_facebook_likes**：脸书上喜爱男一号的人数
- **gross**：总票房
- **genres**：电影题材
- **actor_1_name**：男一号姓名
- **movie_title**：电影片名
- **num_voted_users**：参与投票的用户数量
- **cast_total_facebook_likes**：脸书上投喜爱的总数
- **actor_3_name**：三号男演员姓名
- **facenumber_in_poster**：海报中的人脸数量
- **plot_keywords**：剧情关键字
- **movie_imdb_link**：imdb地址
- **num_user_for_reviews**：用户的评论数量
- **language**：语言
- **country**：国家
- **content_rating**：电影分级
- **budget**：制作成本
- **title_year**：电影年份
- **actor_2_facebook_likes**：脸书上喜爱男二号的人数
- **imdb_score**：imdb上的评分
- **aspect_ratio**：画布的比例
- **movie_facebook_likes**：脸书上被点赞的数量

让我们来一起探索一下这5000多条电影信息记录中蕴含的信息。

我们一起来完成以下代码过程。在如下有# TODO 提示的地方，将代码补全，实现注释中所要求的功能。

> **提示**：这样的文字将会指导你如何使用 jupyter Notebook 来完成项目。你可以通过单击代码区域，然后使用键盘快捷键 **Shift+Enter** 或 **Shift+Return** 来运行代码。或者在选择代码后使用**执行**（run cell）按钮执行代码。

In [1]:

```
# 加载项目中所用到的模块
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
```

In [2]:

```
# 读取数据
movies_df = pd.read_csv('movie_metadata_clean.csv')
```

数据分析

一. 评价最高的20部电影

在视频课时中我们讲解了如何获得评价最高的10部电影，并且打印电影的片名（movie_title),评分（imdb_score),导演名（'director_name'),发行时间（'title_year'),国家（'country')字段。

使用的代码如下：

top10_score_movies = movies_df.sort_values(by = 'imdb_score',ascending = False).head(10)

top10_score_movies[['movie_title','imdb_score','director_name','title_year','country']]

请在下方代码块中 # TODO 下方编写代码，完成以下功能：

筛选出评价最高的20部电影，
并且打印电影的片名（movie_title),评分（imdb_score),导演名（'director_name'),发行时间（'title_year'),国家（'country')

> **提示**：你将使用sort_values函数将数据根据imdb_score字段降序排序，然后使用head函数打印前20行数据。你需要做的是复制以上代码，将head函数中的10改为20

In [ ]:

```
# TODO
```

【运行结果展示】 你将会获得如下信息：

 title

二. 票房收入最高的10部电影

在视频课时中我们讲解了如何获得票房收入最高的10部电影，并且打印电影的片名（movie_title),评分（imdb_score),导演名（'director_name'),发行时间（'title_year'),国家（'country')字段。

使用的代码如下：

top10_gross_movies = movies_df.sort_values(by = 'gross',ascending = False).head(10)

top10_gross_movies[['movie_title','imdb_score','director_name','title_year','country']]

请在下方代码块中 # TODO 下方编写代码，完成以下功能：

筛选出票房收入最高的10部电影
并且打印电影的画面颜色（'color'），语言（'language')

> **提示**：你将使用sort_values函数将数据根据 gross 字段降序排序，然后使用head函数打印前10行数据。接下来使用得到的新数据输出画面颜色和语言两列数据。你需要做的是复制以上代码，将第二行代码中切片时使用到的字段列表修改为只含有'color'和'language'的列表。

In [5]:

```
# TODO
```

【运行结果展示】 你将会获得如下信息：

 title

三. imdb评分最高的10个导演

在视频课时中我们讲解了如何获得总票房收入最高的10个导演，并且绘制导演与总票房收入直方图。

使用的代码如下：

dir_grouped_data = movies_df.groupby('director_name', as_index = False)['gross'].sum()

top10_dir_grouped_data = dir_grouped_data.sort_values(by='gross', ascending=False).head(10)

top10_dir_grouped_data.plot(x = 'director_name', y = 'gross', kind = 'bar')

plt.show()

请在下方代码块中 # TODO 下方编写代码，完成以下功能：

筛选出imdb评分最高的10个导演,
接着，请绘制imdb评分最高的10个导演与其imdb评分直方图

> **提示**：你将会使用到groupby函数按照导演来聚合数据，然后使用sort_values将数据降序排序，使用head函数选择前10行数据，不要忘记使用plt.show()来展示图像哦。你需要做的是复制以上代码，从按照导演聚合之后的数据中提取imdb_score变量，即将'gross'修改为'imdb_score'。同时，之后的代码中'gross'都需要替换为'imdb_score'。

In [ ]:

```
# TODO
```

【运行结果展示】 你将会获得如下图：

 title

四. 导演点赞量与电影点赞量的关系散点图

在视频课时中我们讲解了如何绘制电影评分和总票房的关系散点图

使用的代码如下：

score_gross = movies_df[['imdb_score','gross']]

score_gross.plot(x = 'imdb_score', y = 'gross', kind = 'scatter')

plt.show()

请在下方代码块中 # TODO 下方编写代码，完成以下功能：

筛选出导演点赞量('director_facebook_likes')与电影点赞量('movie_facebook_likes')数据，
并且绘制导演点赞数与电影点赞数散点图。

> **提示**：你需要做的是复制以上代码，将第一行代码movies_df的方括号中变量名'imdb_score'和'gross'分别修改为'director_facebook_likes'和'movie_facebook_likes'，然后将第二行代码plot中x轴变量修改为'movie_facebook_likes'，y轴变量修改为'director_facebook_likes'。

In [ ]:

```
# TODO
```

【运行结果展示】 你将会获得如下图：

 title