Recommender System

Recommender System

背景补充阅读材料：

API : <https://requests.readthedocs.io/en/master/api/>

Python connect to MySQL Server:

\* <https://docs.sqlalchemy.org/en/13/dialects/mysql.html>

\* <https://dev.mysql.com/downloads/workbench/>

PySpark: <https://www.sicara.ai/blog/2017-05-02-get-started-pyspark-jupyter-notebook-3-minutes>

Recommendation Algorithms:

    \* Content Based: <https://en.wikipedia.org/wiki/Recommender_system#Content-based_filtering>

    \* Collaborative Filtering: <https://en.wikipedia.org/wiki/Collaborative_filtering>

FAQ:

1. 爬取数据的时候时间很长，这个正常么？能加速么?

A: 这个是正常的，因为我们的 API call 有流量限制，并且读取数据的速度也有限制，并且我们读取的数据量也比较大，所以速度慢是正常的。我这边测试的结果大概在20个小时左右，如果在这个时间上下都是正常的

2. Steam API 如何才能 access，我注册了账号显示 不能访问

A：Steam API 的获取，要先消费 5 块钱，之后可以申请开发者权限，在开发者权限里面就可以找到自己的 API key 了

3. Windows 如何 安装 Pyspark

A：对于使用 Windows 的同学，建议同学安装一个虚拟机，然后在虚拟机中安装一个 ubuntu 的系统，之后的操作在 ubuntu 的系统中进行

推荐系统小练习：

[data.xlsx](https://piazza.com/redirect/s3?bucket=uploads&prefix=paste%2Fiy3rm5kn9s35jk%2Fdfd626a99241c5c8d6d0907ba74a34c1a789f793781ae35b80b8a7bffe9c344f%2Fdata.xlsx)

[Recommender\_System\_Practice.ipynb](https://piazza.com/redirect/s3?bucket=uploads&prefix=paste%2Fiy3rm5kn9s35jk%2F43f8e624f921c67aa01171103f5cdd97320e0f65eb22c498f7bc0f5f1eaed217%2FRecommender_System_Practice.ipynb)

**Week 1:**

Materials: [W1.zip](https://d1b10bmlvqabco.cloudfront.net/attach/j3ug4vhvxnn762/iyks3spmq2d2nq/j94jqjrhvm7a/W1.zip)

[week\_1.ipynb](https://piazza.com/redirect/s3?bucket=uploads&prefix=attach%2Fk2y2hvaa3uv4lt%2Fiy3rm5kn9s35jk%2Fk70t1eanis13%2Fweek_1.ipynb)

本周我们主要介绍了游戏行业以及本项目的背景，version control和API基本知识。布置给大家的任务是：  
1) 安装source tree  
2) 熟悉 API 基本用法：快速过一遍YouTube API的两篇说明 - [网页1](https://developers.google.com/youtube/v3/docs/) 和 [网页2](https://developers.google.com/youtube/v3/docs/videos/list)  
3) 通过官方的[Steam Web API](https://developer.valvesoftware.com/wiki/Steam_Web_API" \l "GetOwnedGames_.28v0001.29" \t "_blank)，获得5000个用户的game inventory，需要注册steam developer account获得API key  
- 阅读 API 说明 (GetOwnedGames部分)  
- 注册账号，获取API key  
- build URL，使用Python的requests包，获得某个用户的game inventory信息  
- 将获取的信息保存到一个txt文件。要求：  
a. 每一行是一个 {user\_steam\_id : user\_game\_inventory} 格式的dictionary  
b. 5000个用户Id，对应5000行信息  
4) 通过非官方的[Steam API](https://wiki.teamfortress.com/wiki/User:RJackson/StorefrontAPI" \t "_blank)获得500个游戏的detail信息，注意该API每5分钟限制200次API call  
- 阅读API说明 (appdetail部分)  
- hint: 使用自带的time包可以设置等待时间，比如在for loop中使用 time.sleep(5) ，每执行完一次loop，都会等待5秒钟

**Week 2:**

Materials:

* [app\_detail.txt.zip](https://piazza.com/redirect/s3?bucket=uploads&prefix=attach%2Fk05n28kws0w7df%2Fiy3rm5kn9s35jk%2Fk4x1xh3dyy76%2Fapp_detail.txt.zip)
* [week\_2.ipynb](https://piazza.com/redirect/s3?bucket=uploads&prefix=attach%2Fk2y2hvaa3uv4lt%2Fiy3rm5kn9s35jk%2Fk70t0jwe1x70%2Fweek_2.ipynb)
* [Week\_2.pdf](https://piazza.com/redirect/s3?bucket=uploads&prefix=attach%2Fk05n28kws0w7df%2Fiy3rm5kn9s35jk%2Fk4x1y8l1vbgr%2FWeek_2.pdf)

本周给大家布置的任务是用附件中 app\_detail.txt.zip 这个文件，生成一张app info的表格，必须要有的information:

- app id

- app name

- initial price

- header image url

其他features请同学自己选择添加