Assignment 1:

Memory-based CF for Rating Prediction

Overview

- Programming
  - UCF or ICF for rating prediction

- Requirements
  - Project report
  - Executable source code
  - Prediction results

Project Report

- Main content
  - Detailed implementation introduction
  - Key code explanation
  - Result analysis
    - Hyper-parameter influence
      - Table or plot
  - How to run your code
    - Environments (E.g., python 2.X or python 3.x, c++?)
    - Dependency libraries

Source Code

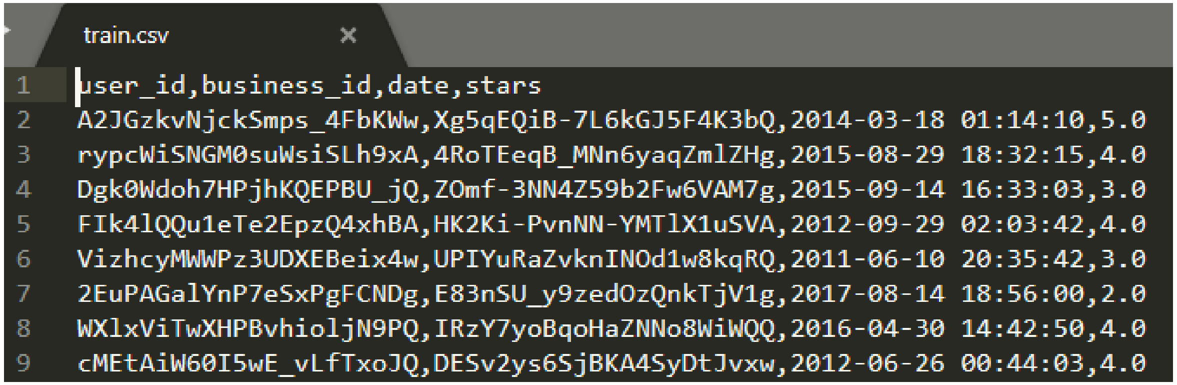
- Programming language
  - Python is preferred.

- Terminal command
  - E.g., python2.7 xxx.py test.file output.file

- Source code files
  - How the method is implemented

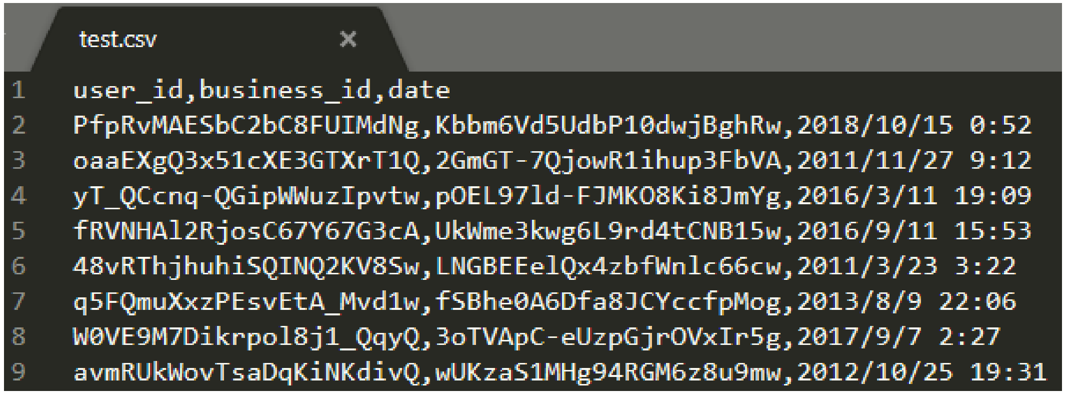
Data Format

- Train set
  - With rating scores



```
user_id,business_id,date,stars
A2JGzkvNjckSmps_4FbKWw,Xg5qEQiB-7L6kGJ5F4K3bQ,2014-03-18 01:14:10,5.0
rypcWiSNGM0suWsiSLh9xA,4RoTEeqB_MNn6yaqZmlZHg,2015-08-29 18:32:15,4.0
Dgk0Wdoh7HPjhKQEPBU_jQ,ZOmf-3NN4Z59b2Fw6VAM7g,2015-09-14 16:33:03,3.0
FIk4lQQu1eTe2EpzQ4xhBA,HK2Ki-PvnNN-YMTlX1uSVA,2012-09-29 02:03:42,4.0
VizhcyMWWPz3UDXEBeix4w,UPIYuRaZvknINOd1w8kqRQ,2011-06-10 20:35:42,3.0
2EuPAGalYnP7eSxPgFCNDg,E83nSU_y9zedOzQnkTjV1g,2017-08-14 18:56:00,2.0
WXlxViTwXHPBvhioljN9PQ,IRzY7yoBqoHaZNNo8WiWQQ,2016-04-30 14:42:50,4.0
cMEtAiW60I5wE_vLfTxoJQ,DESv2ys6SjBKA4SyDtJvxw,2012-06-26 00:44:03,4.0
```

- Test set
  - Without rating scores



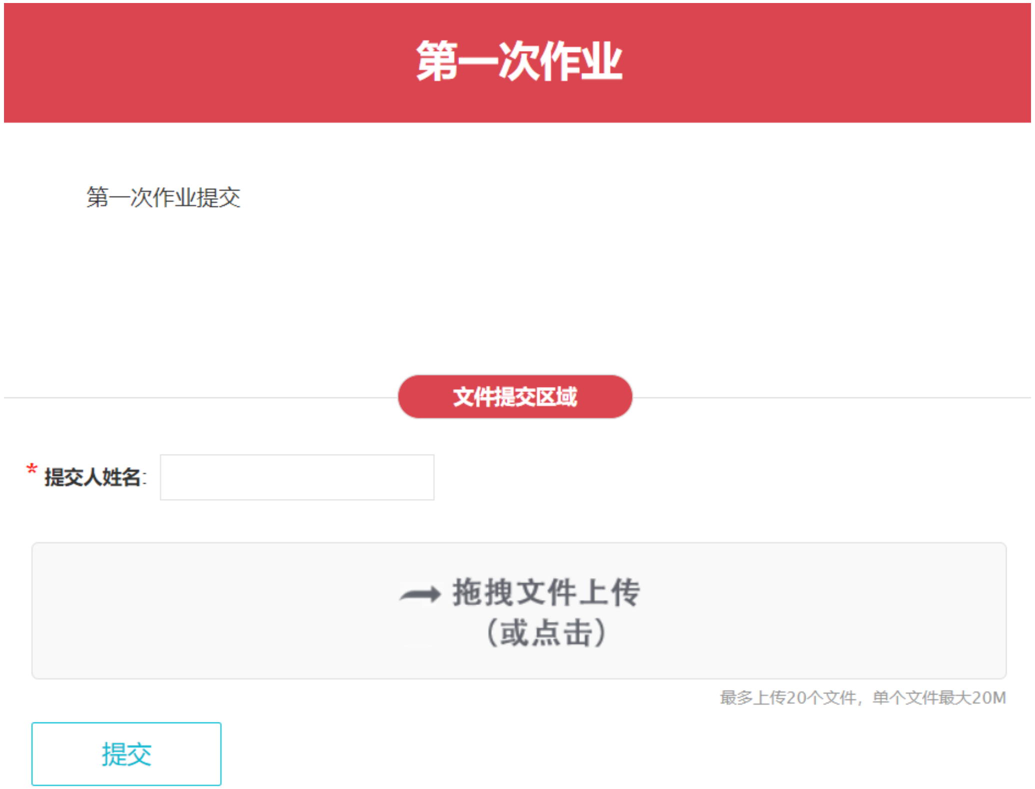
```
user_id,business_id,date
PfpRvMAESbC2bC8FUIMdNg,Kbbm6Vd5UdbP10dwjBghRw,2018/10/15 0:52
oaaEXgQ3x51cXE3GTXrT1Q,2GmGT-7QjowR1ihup3FbVA,2011/11/27 9:12
yT_QCcnq-QGipWWuzIpvtw,pOEL97ld-FJMKO8Ki8JmYg,2016/3/11 19:09
fRVNHAl2RjosC67Y67G3cA,UkWme3kwg6L9rd4tCNB15w,2016/9/11 15:53
48vRThjhuhiSQINQ2KV8Sw,LNGBEEelQx4zbfWnlc66cw,2011/3/23 3:22
q5FQmuXxzPEsvEtA_Mvd1w,fSBhe0A6Dfa8JCYccfpMog,2013/8/9 22:06
W0VE9M7Dikrpol8j1_QqyQ,3oTVApC-eUzpGjrOVxIr5g,2017/9/7 2:27
avmRUkWovTsaDqKiNKdivQ,wUKzaS1MHg94RGM6z8u9mw,2012/10/25 19:31
```

Prediction Results

- A file containing prediction results
  - Format: each line corresponds to one prediction tuple
    - User_id, Business_id, Rating_score
    - E.g., A2JGzkvNjckSmps_4FbKWw, Xg5qEQiB-7L6kGJ5F4K3bQ, 4.0

Submission

- File name
  - zip file, named with ‘Name+ID+Assignment No.’
- Submission URL
  - http://xzc.cn/J7knNdne0x



第一次作业

第一次作业提交

文件提交区域

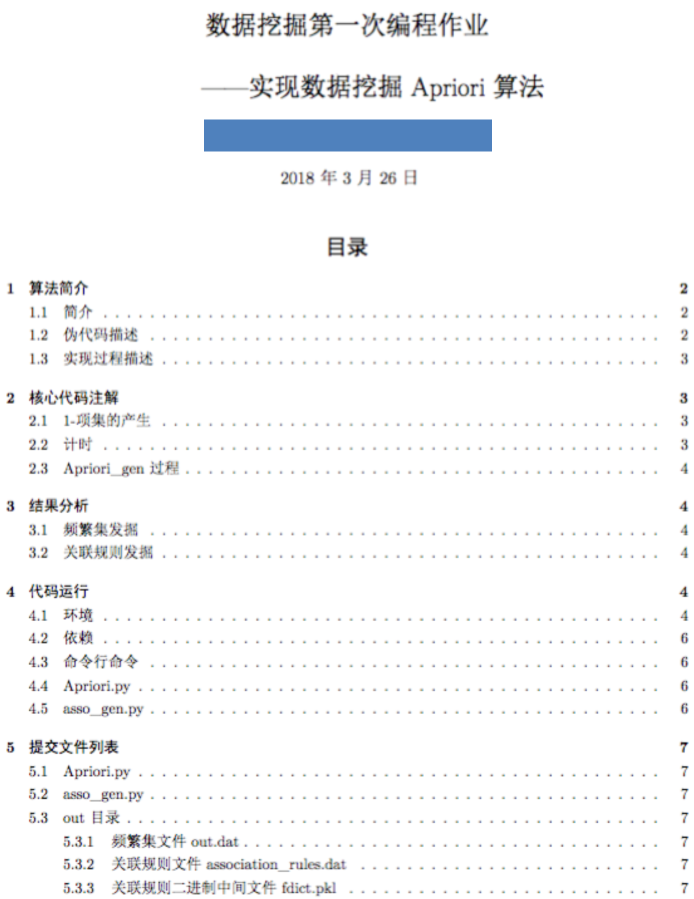
* 提交人姓名:

最多上传20个文件，单个文件最大20M

How to evaluate?

- Code
  - Executable
  - Clear, easy-to-understood
- Prediction results
  - Effectiveness
    - How is the predicted score compared with ground-truth?
    - RMSE
- Report
  - Integrity
  - Readability
  - Highlights

Report Example



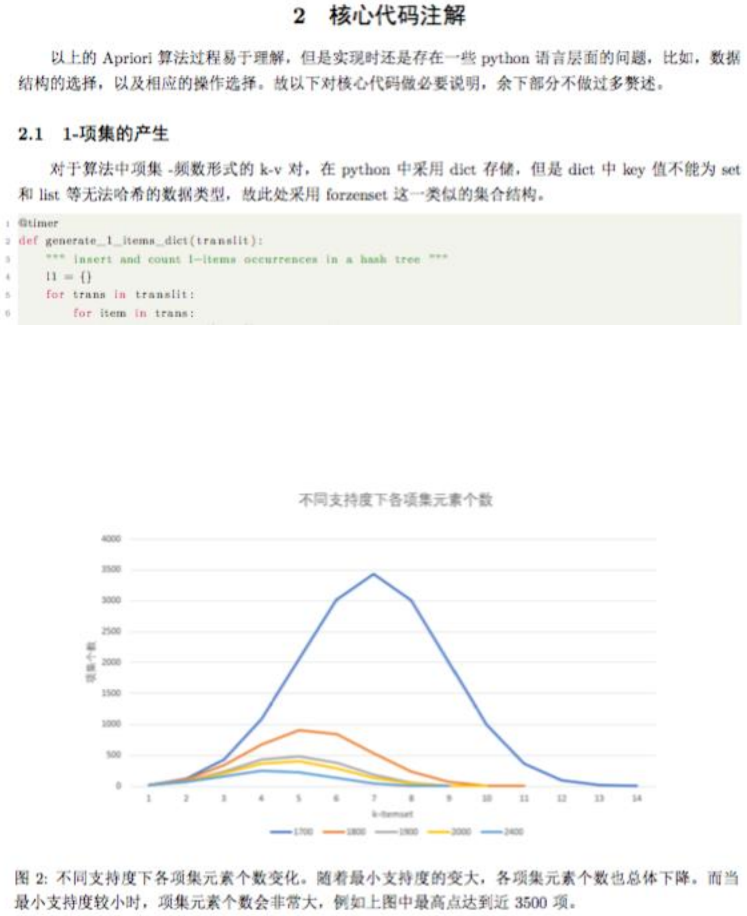
数据挖掘第一次编程作业

——实现数据挖掘 Apriori 算法

[redacted]

2018 年 3 月 26 日

目录



2 核心代码注解

以上的 Apriori 算法过程易于理解，但是实现时还是存在一些 python 语言层面的问题，比如，数据结构的选择，以及相应的操作选择。故以下对核心代码做必要说明，余下部分不做过多赘述。

2.1 1-项集的产生

对于算法中项集 -频数形式的 k-v 对，在 python 中采用 dict 存储，但是 dict 中 key 值不能为 set 和 list 等无法哈希的数据类型，故此处采用 forzenset 这一类似的集合结构。

```
@timer
def generate_1_items_dict(translit):
    """ insert and count 1-items occurrences in a hash tree """
    l1 = {}
    for trans in translit:
        for item in trans:
```

图 2: 不同支持度下各项集元素个数变化。随着最小支持度的变大，各项集元素个数也总体下降。而当最小支持度较小时，项集元素个数会非常大，例如上图中最高点达到近 3500 项。

Data and Deadline

- **Data**
  - https://pan.baidu.com/s/1nc63rbYsU58PyvoZJtLgBQ
  - 提取码: pgqe

- **Deadline**
  - 24:00, 2020-4-5
  - Submission after that time might be penalized