ĐỒ ÁN GIỮA KỲ

Môn: Xử lý dữ liệu lớn

Thời gian làm bài: 02 tuần

I. Hình thức

- Đồ án giữa kỳ được thực hiện theo nhóm **02 – 03** sinh viên.
- Nhóm sinh viên thực hiện các yêu cầu và nộp bài theo hướng dẫn chi tiết bên dưới.

II. Yêu cầu

Cho tập tin **data.csv** chứa dữ liệu mua hàng của người dùng. Trong đó, dòng đầu tiên chứa tiêu đề (header), các dòng còn lại là dữ liệu tương ứng.

- **Member_number**: mã số khách hàng
- **Date**: ngày mua hàng dạng dd/mm/yyyy
- **itemDescription**: tên của 01 món hàng
- **year**: năm mua
- **month**: tháng mua
- **day**: ngày mua
- **day_of_week**: thứ trong tuần

Ví dụ

Member_number	Date	itemDescription	year	month	day	day_of_week
1249	01/01/2014	citrus fruit	2014	1	1	2
1249	01/01/2014	coffee	2014	1	1	2
1381	01/01/2014	curd	2014	1	1	2
1381	01/01/2014	soda	2014	1	1	2
1440	01/01/2014	other vegetables	2014	1	1	2
1440	01/01/2014	yogurt	2014	1	1	2
1659	01/01/2014	specialty chocolate	2014	1	1	2
1659	01/01/2014	frozen vegetables	2014	1	1	2
1789	01/01/2014	hamburger meat	2014	1	1	2
1789	01/01/2014	candles	2014	1	1	2

Dữ liệu trong data.csv (hiển thị trên Google Colab)

a) Câu 1 (2.0 điểm): Đếm món hàng

Sử dụng **textFile()** trong thư viện PySpark để đọc tập tin **data.csv** và đếm số lượng món hàng mỗi người khách mua trong một ngày. Kết quả xử lý ghi xuống thư mục **counters** sử dụng hàm **saveAsTextFile()**. Nội dung kết quả gồm: *Mã khách hàng, Ngày mua, Số lượng.*

Ví dụ

part-00000 ×

```
1 Member_number,Date,1
2 1249,01/01/2014,2
3 1381,01/01/2014,2
4 1440,01/01/2014,2
5 1659,01/01/2014,2
```

Kết quả đếm số lượng món hàng mỗi người mua trong một ngày.

Dòng đầu tiên phát sinh từ header ban đầu.

Sinh viên có thể đọc tập tin kết quả bằng **Dataframe** của **SQL Context** và hiển thị ra để kiểm tra với hàm **show()**.

Ví dụ

```
+-------------+----------+---+
|Member_number|      Date|  1|
+-------------+----------+---+
|         1249|01/01/2014|  2|
|         1381|01/01/2014|  2|
|         1440|01/01/2014|  2|
|         1659|01/01/2014|  2|
|         1789|01/01/2014|  2|
|         1922|01/01/2014|  2|
|         2226|01/01/2014|  2|
```

Ví dụ hiển thị kết quả Câu 1 bằng DataFrame

b) Câu 2 (2.0 điểm): Giỏ hàng

Sử dụng **textFile()** trong thư viện PySpark để đọc tập tin **data.csv** và tìm ra *danh sách món hàng mỗi người khách mua trong một ngày*. Kết quả xử lý ghi xuống thư mục **baskets** sử dụng hàm **saveAsTextFile()**. Nội dung kết quả gồm: *Mã khách hàng, Ngày mua, Danh sách món hàng*. Lưu ý các cột cách nhau bằng dấu "**;**" và các phần tử món hàng trong cột *Danh sách món hàng* cách nhau bằng dấu "**,**".

Ví dụ

part-00000 ×

```
1 Member_number;Date;itemDescription
2 1249;01/01/2014;citrus fruit,coffee
3 1381;01/01/2014;curd,soda
4 1440;01/01/2014;other vegetables,yogurt
5 1659;01/01/2014;specialty chocolate,frozen vegetables
6 1789;01/01/2014;hamburger meat,candles
7 1922;01/01/2014;tropical fruit,other vegetables
8 2226;01/01/2014;sausage,bottled water
9 2237;01/01/2014;bottled water,Instant food products
10 2351;01/01/2014;cleaner,shopping bags
```

Kết quả tìm danh sách món hàng mỗi người mua trong một ngày.

Dòng đầu tiên phát sinh từ header ban đầu. Các phần tử cách nhau bởi dấu ';'

Danh sách món hàng là dạng chuỗi, ngăn cách bởi dấu ','.

Sinh viên có thể đọc tập tin kết quả bằng **Dataframe** của **SQL Context** và hiển thị ra để kiểm tra với hàm **show()**.

```
+-------------+----------+--------------------+
|Member_number|      Date|     itemDescription|
+-------------+----------+--------------------+
|         1249|01/01/2014| citrus fruit,coffee|
|         1381|01/01/2014|           curd,soda|
|         1440|01/01/2014|other vegetables,...|
|         1659|01/01/2014|specialty chocola...|
|         1789|01/01/2014|hamburger meat,ca...|
```

Kết quả hiển thị Câu 2 bằng DataFrame

c) Câu 3 (2.0 điểm): Tập phổ biến

- Đọc tập tin kết quả ở mục b) lên thành **DataFrame** có tên ***dfBaskets***, chuyển đổi cột **itemDescription** thành dạng mảng các chuỗi và hiển thị ra màn hình với lệnh show(). Lưu ý với mỗi chuỗi **itemDescription**, mỗi món hàng xuất hiện không quá một lần (unique).
- Đổi tên các cột trong dataframe kết quả theo thứ tự là *Member_number*, *Date* và *Items*.

Ví dụ

```
+-------------+----------+--------------------+
|Member_number|      Date|               Items|
+-------------+----------+--------------------+
|         1249|01/01/2014|[coffee, citrus f...|
|         1381|01/01/2014|        [soda, curd]|
|         1440|01/01/2014|[yogurt, other ve...|
|         1659|01/01/2014|[specialty chocol...|
|         1789|01/01/2014|[hamburger meat, ...|
|         1922|01/01/2014|[other vegetables...|
|         2226|01/01/2014|[sausage, bottled...|
|         2237|01/01/2014|[bottled water, I...|
|         2351|01/01/2014|[shopping bags, c...|
```

Dataframe kết quả sau khi đổi tên theo yêu cầu.

- Sử dụng thư viện `pyspark.ml.fpm.FPGrowth` để xây dựng mô hình tập bổ biến với **minSupport=1%** và các luật liên kết với **minConfidence=10%**. Sinh viên hiển thị danh sách tập phổ biến và luật liên kết ra màn hình với lệnh show().

  Ví dụ:

```
+--------------------+----+
|               items|freq|
+--------------------+----+
|              [beef]| 508|
|             [sugar]| 265|
|               [oil]| 223|
|         [chocolate]| 353|
|        [white wine]| 175|
|             [candy]| 215|
|  [processed cheese]| 152|
|              [meat]| 252|
```

Tập phổ biến

```
+------------------+------------+--------------------+----
|        antecedent|  consequent|          confidence|
+------------------+------------+--------------------+----
|[other vegetables]|[whole milk]|0.12151067323481117|0.76
|          [yogurt]|[whole milk]|0.12996108949416343|0.82
|      [rolls/buns]|[whole milk]|0.12697448359659783|0.80
|            [soda]|[whole milk]|0.11975223675154852|0.75
```

Luật liên kết

d) Câu 4 (1.0 điểm): Giỏ hàng thành vector

- Đọc tập tin **data.csv** lên thành **DataFrame** có tên ***dfMembers*** gồm hai cột
  - *Member_number*: Mã khách hàng
  - *Items*: Danh sách món hàng, cách nhau bởi dấu ",", chứa tất cả món hàng được mua bởi 01 khách hàng.
- Sinh viên có thể hiển thị kết quả ra màn hình với hàm **show()**.

Ví dụ

```
+-------------+--------------------+
|Member_number|               Items|
+-------------+--------------------+
|         1249|citrus fruit,coff...|
|         1381|curd,soda,coffee,...|
|         1440|other vegetables,...|
|         1659|specialty chocola...|
```

Kết quả DataFrame dfMembers

- Tìm danh sách món hàng trong toàn bộ dữ liệu, lưu vào biến ***items***, trong đó các phần tử được sắp xếp theo thứ tự bảng chữ cái tăng dần. Sau đó, tạo ra một **dictionary** tên ***dictItems*** chứa phần tử dạng <tên món hàng>:<chỉ số mảng>. In giá trị của hai biến trên ra màn hình.

  Ví dụ

```
['Instant food products', 'UHT-milk', 'abrasive cleaner', 'artif.
{'Instant food products': 0, 'UHT-milk': 1, 'abrasive cleaner': 2,
```

*Kết quả in ra màn hình biến **items** và **dictItems***

- Viết hàm ***basket2vector(member, basket, dictItems)*** để chuyển đổi từ một mã khách hàng (***member***), một danh sách món hàng (***basket***) dạng chuỗi với các phần tử cách nhau bởi dấu "," và biến ***dictItems*** ở trên. Kết quả trả về là một **Vectors.sparse** (**pyspark.ml.linalg**) với độ dài bằng số lượng phần tử của ***dictItems***, giá trị là 0.0 hoặc 1.0 ứng với sự xuất hiện của mỗi món hàng trong ***basket*** *(ví dụ món hàng **it** có trong **basket** thì phần tử vector tại **dictItems[it]** sẽ là 1.0).*

*Ví dụ: gọi hàm **basket2vector** và truyền dòng đầu tiên trong **dfMembers** vào xử lý.*

```
print(basket2vector(dfMembers.first()['Member_number'],
                    dfMembers.first()['Items'],
                    dictItems))
```

```
(167,[11,30,34,61,138],[1.0,1.0,1.0,1.0,1.0])
```

*Kết quả **basket2vector** với dòng đầu tiên trong **dfMembers***

Sinh viên lưu ý **DataFrame *dfMembers*** được dùng cho mục e) và mục f).

e) Câu 5 (1.0 điểm): Giỏ hàng tương tự

- Tạo ra mô hình **MinHashLSH** trong thư viện **pyspark.ml.feature** để tìm các người dùng có thói quen mua sản phẩm giống nhau. Sử dụng ***dfMembers*** từ câu trên với
  - ***inputCol*** là "Items"
  - ***outputCol*** là "Hashes"
  - Số lượng bảng băm là 5
- Sinh viên huấn luyện mô hình với ***dfMembers*** sau đó dùng hàm transform() để chuyển đổi ***dfMembers*** và hiển thị **DataFrame** kết quả ra màn hình. Lưu ý không thay đổi giá trị của biến ***dfMembers*** vì còn dùng ở câu sau.

Ví dụ

```
+-------------+--------------------+--------------------+
|Member_number|               Items|              Hashes|
+-------------+--------------------+--------------------+
|         1249|(167,[11,30,34,61...|[[2.85001106E8], ...|
|         1381|(167,[1,10,11,28,...|[[3.9022841E7], [...|
|         1440|(167,[28,64,102,1...|[[3.41446049E8], ...|
|         1659|(167,[12,14,26,34...|[[1.02200615E8], ...|
```

DataFrame** kết quả chuyển đổi từ **dfMembers

- Sử dụng hàm **approxSimilarityJoin**() để tìm ra các cặp người dùng có thói quen mua giống nhau với độ do **JaccardDistance** không quá ***0.3***. Kết quả hiển thị ra màn hình gồm cặp mã khách hàng và **JaccardDistance**, trong đó chỉ lọc những dòng dữ liệu có *JaccardDistance lớn hơn 0.*

Ví dụ

```
+----+----+-------------------+
| idA| idB|    JaccardDistance|
+----+----+-------------------+
|3124|1063|               0.25|
|1643|4535|0.19999999999999996|
|1860|3605|               0.25|
|4342|1056| 0.2857142857142857|
|2911|3714|               0.25|
|3715|4805|               0.25|
```

Các cặp người dùng tìm ra

- Sử dụng hàm **approxNearestNeighbors**() để tìm ra **05** người khách hàng có thói quen mua sắm giống với người đầu tiên trong ***dfMembers***. Hiển thị kết quả tìm kiếm ra màn hình.

Ví dụ

```
+-------------+--------------------+--------------------+------------------+
|Member_number|               Items|              Hashes|           distCol|
+-------------+--------------------+--------------------+------------------+
|         1249|(167,[11,30,34,61...|[[2.85001106E8], ...|               0.0|
|         1321|(167,[11,30,138],...|[[2.85001106E8], ...|               0.4|
|         1263|(167,[11,30,61,10...|[[2.85001106E8], ...|               0.5|
|         1794|(167,[11,30,138,1...|[[2.85001106E8], ...|0.5714285714285714|
|         4327|(167,[30,34,63,76...|[[3.43690326E8], ...|0.5714285714285714|
+-------------+--------------------+--------------------+------------------+
```

*Kết quả 05 người có thói quen mua sắm giống với người đầu tiên trong **dfMembers**.*

f) Câu 6 (1.0 điểm): Phân cụm người dùng theo giỏ hàng

- Tạo ra mô hình **pyspark.ml.clustering.KMeans** để phân cụm những người khách hàng có thói quen mua sắm giống nhau trong ***dfMembers*** với **k=5.**
- Sau khi hoàn thành huấn luyện, sinh viên tính toán và in DataFrame kết quả ra màn hinh sử dụng hàm **transform**().

Ví dụ

```
+-------------+--------------------+----------+
|Member_number|               Items|prediction|
+-------------+--------------------+----------+
|         1249|(167,[11,30,34,61...|         2|
|         1381|(167,[1,10,11,28,...|         2|
|         1440|(167,[28,64,102,1...|         3|
```

Kết quả phân cụm với k-Means

g) Câu 7 (1.0 điểm): Báo cáo

- Sinh viên viết báo cáo, xuất thành tập tin ***report.pdf***. Trong đó, bao gồm các nội dung
  - Danh sách sinh viên: MSSV, Họ tên, Email, Phân công công việc, Mức độ hoàn thành.
  - Với mỗi câu 1 – 6, sinh viên giải thích ngắn gọn các viết lệnh hay quá trình xử lý của các hàm, ví dụ từng dòng dữ liệu được hàm map() …, sau đó hàm reduce() ….
  - **Độ dài tối đa cho báo cáo là 2 trang A4, font Times New Roman, fontSize 13, giãn dòng 1.5**
  - Sinh viên có thể chép mã nguồn vào phần giải thích hoạt động từng câu nhưng chú ý không trình bày dài dòng, cẩu thả.

III. Hướng dẫn nộp bài

- Tạo thư mục với tên theo cú pháp <MSSV1>_<MSSV2> (tương tự cho nhóm gồm 03 sinh viên), trong đó gồm:
  - <MSSV1>_<MSSV2>.ipynb chứa mã nguồn đồ án
  - <MSSV1>_<MSSV2>.pdf kết xuất pdf mã nguồn đồ án từ Google Colab.
  - report.pdf: báo cáo đồ án.
- Lưu ý giữ lại kết quả thực thi của các ô trong cả hai tập tin .ipynb và .pdf.
- Nén thư mục thành <MSSV1>_<MSSV2>.zip và nộp theo deadline.

IV. Thời gian nộp bài

- Thời lượng cho đồ án là 02 tuần kể từ ngày công bố.
- Sinh viên nộp trễ hơn hạn trên đồng nghĩa với việc **0.0 điểm** giữa.

V. Quy định

- **Mọi hành vi sao chép code trên mạng, chép bài bạn hoặc cho bạn chép bài nếu bị phát hiện đều sẽ bị điểm 0.0.**
- **Nếu bài làm của sinh viên có dấu hiệu sao chép trên mạng hoặc sao chép nhau, sinh viên sẽ được gọi lên phỏng vấn code để chứng minh bài làm là của mình.**

-- HẾT --