TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

BÁO CÁO GIỮA KÌ MÔN HỌC XỬ LÍ DỮ LIỆU LỚN

BÀI TẬP TỔNG HỢP VỀ SPARK

Giáo viên hướng dẫn: Th.S Nguyễn Thành An
Sinh viên thực hiện: Dương Trọng Chí - MSSV: 52000742
Trần Gia Hoàng - MSSV: 52000759
Trần Khánh Duy - MSSV: 52000042

Hồ Chí Minh - 2022

Mục lục

Bảng Đánh Giá Mức Độ Hoàn Thành Công Việc

Mã Số Sinh Viên	Họ và tên	Phân chia công việc	Mức Độ Hoàn Thành
52000742	Dương Trọng Chí	Tìm hiểu câu 1 và 2	Hoàn thành tốt
52000759	Trần Gia Hoàng	Tìm hiểu câu 3 và 4	Hoàn thành tốt
52000042	Trần Khánh Duy	Tìm hiểu câu 5 và 6	Hoàn thành tốt

Chương 1

Bài tập giữa kì Xử Lí Dữ Liệu Lớn

1.1 Câu 1

Phương thức `def preprocessLine(x)` sẽ trả về một tuple với key là giá trị từng dòng của cột Member number và Date và values là 1.

Phương thức `def f(x)` sẽ trả về giá trị của cột Member number, Date và số lượng hàng theo từng dòng, các dòng này được ngăn cách bởi dấu ",".

Sau đó, ta dùng hàm map() gọi lại phương thức `def preprocessLine(x)` lúc này RDD mỗi dòng sẽ có dạng key là giá trị của Member number, Date và values là 1. Sau đó, ta dùng `reduceByKey()` để tiến hành cộng value của các RDD có key giống nhau lại. Cuối cùng, ta dùng hàm `map()` gọi phương thức `f(x)` để tiến hành định dạng lại từng cột Member number, Date, số lượng ngăn cách bởi dấu ",".

1.2 Câu 2

Phương thức `def get_key_value(x)` sẽ gọi hàm `split()` để lấy 3 cột Member number, Date, itemDescription. Sau đó, sẽ trả về một tuple với key là giá trị từng dòng của cột Member number và Date và values là giá trị theo từng dòng của itemDescription.

Phương thức `def f2(x)` sẽ trả về giá trị của cột Member number, Date và itemDescription theo từng dòng và ngăn cách ba cột này bởi dấu ",".

Sau đó, ta dùng hàm map() gọi lại phương thức `def get_key_value(x)` lúc này RDD mỗi dòng sẽ có dạng key là giá trị của Member number, Date và values là giá trị của itemDescription theo từng dòng. Sau đó, ta dùng reduceByKey để tiến hành cộng values của các RDD có key giống nhau lại và ngăn cách chúng bằng dấu ",". Cuối cùng, ta dùng hàm `map()` gọi phương thức `f(x)` để tiến hành định dạng lại từng cột Member number, Date, itemDescription ngăn cách bởi dấu ";".

1.3 Câu 3

Phương thức `def convertToItemsList()` sẽ chuyển các dòng trong cột itemDescription sang dạng mảng bằng phương thức `list()` và dùng phương thức `set()` để làm cho các phần tử trong itemDescription chỉ xuất hiện 1 lần. Tiếp theo, ta dùng hàm `map()` và truyền vào phương thức `convertToItemsList()` để xử lí từng dòng của cột itemDescription sang dạng list. Sau đó, ta tiến hành tạo một dataFrame bằng phương thức `createDataFrame()` truyền vào một RDD và 3 cột có tên lần lượt là Member number, Date, Items. Tiếp đến, gọi phương thức `FPGrowth()` truyền vào đó 3 tham số itemsCol="Items", minSupport=0.01, minConfidence=0.1 và gán giá trị của phương thức này vào trong biến fpGrowth. Cuối cùng, ta gọi phương thức `fpGrowth.fit()` để tiến hành tạo ra mô hình.

1.4 Câu 4

Đầu tiên ta tiến hành đọc file csv bằng phương thức `read.csv()`. Tiếp theo, ta tiến hành dùng hàm `select()` chọn ra 2 cột Member number và itemDescription. Sau đó, ta chuyển chúng thành rdd bằng phương thức `rdd`. Kế tiếp, ta dùng `reduceByKey()` để tiến hành gôm các itemDescription thành môt cột và ngăn cách bởi dấu phẩy, ta lại chuyển chúng lại thành DataFrame bằng phương thức `toDF()`. Kế tiếp, ta dùng phương thức `selectExpr()` để tiến hành đổi tên 2 cột Member number, itemDescription thành Member number, Items và gán giá trị của hai cột này vào biến dfMembers.

Để tìm danh sách các món hàng trong toàn bộ dữ liệu ta tiến hành đọc dữ liệu từ cột itemDescription bằng lệnh `select()` sau đó chuyển chúng thành rdd. Để các phần tử không bị trùng lặp ta dùng hàm `distinct()` để loại bỏ các phần tử bị trùng. Tiếp theo, ta dùng hàm `flatMap()` để tiến hành làm phẳng dữ liệu.

Sau đó, để tạo ra một dictionary với các phần tử gồm tên món hàng:chỉ số mảng ta tiến hành dùng một vòng lặp và sử dụng phương thức `enumerate` để tạo ra chỉ số index.

Phương thức `basket2vector(member, basket, dictItems)` phương thức này đầu tiên sẽ duyệt qua một dictItems có dạng là một dictionary với key là tên món hàng và values là chỉ số index của biến dictItems. Phương thức này sẽ duyệt qua dictItems và kiểm tra xem nếu key của dictItems có tồn tại trong đối số basket truyền vào thì ta tiến hành thêm values của biến này vào một mảng, đồng thời ta cũng thêm giá trị 1 vào một mảng khác tương ứng. Cuối cùng, phương thức này trả về một vectors.sparse có độ dài là độ dài của dictItems, giá trị của mảng chứa value và mảng chứa giá trị 1 mà ta đã tiến hành xử lí ở vòng lặp trên.

1.5 Câu 5

Để tạo ra mô hình MinHashLSH đầu tiên ta tiến hành đọc dữ liệu từ dataFrame có tên là dfMembers và chuyển dataFrame này thành RDD. Sau đó, ta dùng hàm `map` xử lí mỗi dòng để trả về một cặp key là Member number và values là giá trị của phương thức

`basket2vector` mà ta đã tạo ra ở câu 4. Tiếp đến, ta chuyển lại RDD thành dataFrame bằng phương thức `toDF()` và đổi tên 2 cột `_1` và `_2` lần lượt thành Member number và Items sau đó ta lưu vào biến `new_dfMembers`.

Để tạo ra mô hình MinHashLSH ta tiến hành tạo đối tượng MinHashLSH và truyền vào 3 tham số inputCol và outputCol lần lượt là Items và Hashes và numHashTables là 5. Sau đó, ta tiến hành lưu đối tượng này vào biến `mh_lsh`.

Để tìm ra các cặp người dùng có thói quen mua giống nhau ta dùng phương thức `approxSimilarityJoin()` và dùng phương thức `filter()` để lọc các dòng dữ liệu có JaccardDistance lớn hơn 0.

Để tìm ra 5 người có thói quen mua sắm giống với người đầu tiên trong dfMember ta dùng phương thức `approxNearestNeighbors` truyền vào 3 đối số `new_dfMembers`, dòng đầu tiên của cột Items và giá trị 5.

1.6 Câu 6

Để phân cụm những người khách hàng có thói quen mua sắm giống nhau trong dfMembers với k=5. Ta tiến hành tạo một đối tượng `Kmeans()` với đối số truyền vào k = 5. Sau đó, ta dùng phương thức `alias()` tiến hành đổi tên các cột Member number, Items lần lượt thành label, feature và tiến hành lưu kết quả vào biến `new_dfMembers_km`. Để tiến hành huấn luyện mô hình ta gọi phương thức `fit()` và truyền vào phương thức này một dataFrame chứa các cột label và feature.

Để in kết quả ra màn hình ta tiến hành gọi phương thức transform truyền vào đối số là biến `new_dfMembers_km` vừa tạo và dùng lệnh `show()` để hiển thị thông tin ra mà hình.