TRƯỜNG ĐẠI HỌC FPT

FPT UNIVERSITY

ASSIGNMENT 2
DATA STRUCTURE AND ALGORITHM

Người thực hiện: **Phạm Đông Đông – SE160168**

Lớp : **AI1705**

Khoá : **16**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

TRƯỜNG ĐẠI HỌC FPT

FPT UNIVERSITY

ASSIGNMENT 2
DATA STRUCTURE AND ALGORITHM

Người thực hiện: **Đỗ Tiến Đạt – SE160187**

Lớp : **AI1705**

Khoá : **16**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

PHẦN ĐÁNH GIÁ CỦA GIẢNG VIÊN

_____

_____

_____

_____

_____

_____

_____

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

MỤC LỤC

Trang

1. Bài toán
2. Giải quyết bài toán
    2.1 Mô tả cấu trúc dữ liệu
    2.2 Sơ đồ giải thuật
    2.3 Hiện thực
    2.4 Kết quả và thảo luận
3. Kết luận

Tự đánh giá

APRIORI ALOGORITHM

1. Bài toán

1.1 Giới thiệu bài toán

Cài đặt giải thuật Apriori và ứng dụng giải thuật trong giải quyết bài toán tìm tập luật phổ biến của tập hóa đơn bán hàng. So sánh tập luật tìm được với Công cụ Weka.

1.2 Ý nghĩa bài toán

Apriori là một thuật toán dùng để khai thác tập hợp mục thường xuyên và học quy tắc kết hợp trên tập dữ liệu đã cho. Nó hoạt động bằng cách xác định các mục riêng lẻ thường xuyên trong tập dữ liệu và mở rộng chúng thành các tập mục lớn hơn và lớn hơn miễn là các tập mục đó xuất hiện đủ thường xuyên trong tập dữ liệu.[1-2]

Ví dụ khi khánh hàng mua bánh mì thì thuật toán này sẽ chỉ ra được tỉ lệ phần trăm các món hàng sẽ được mua kèm với bánh mì như : mứt, bơ, sữa,… từ đó cửa hàng sẽ có thể sắp xếp các món hàng để tăng doanh thu.

2. Giải quyết bài toán

2.1 Mô tả cấu trúc dữ liệu

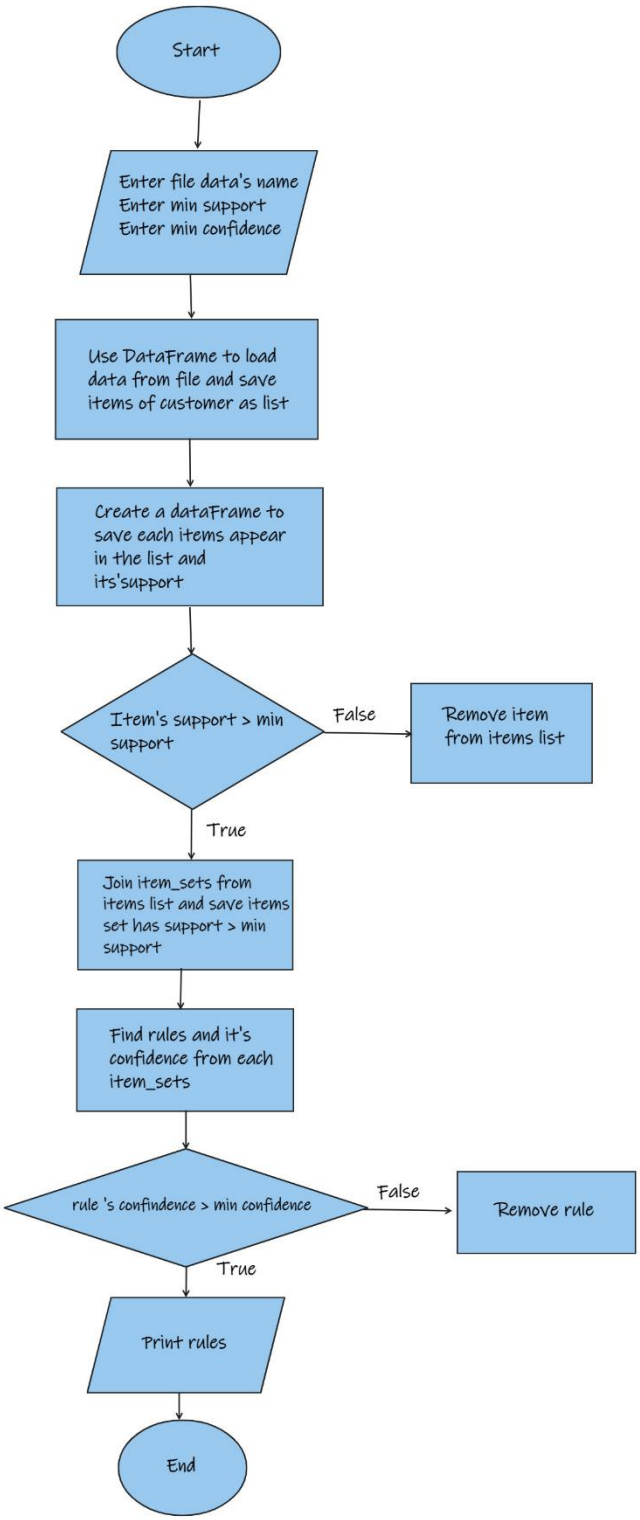
Cấu trúc dữ liệu được sử dụng trong bài toán là set và frozenset, dataframe.

+ Set và frozenset được sử dụng để lưu các items.

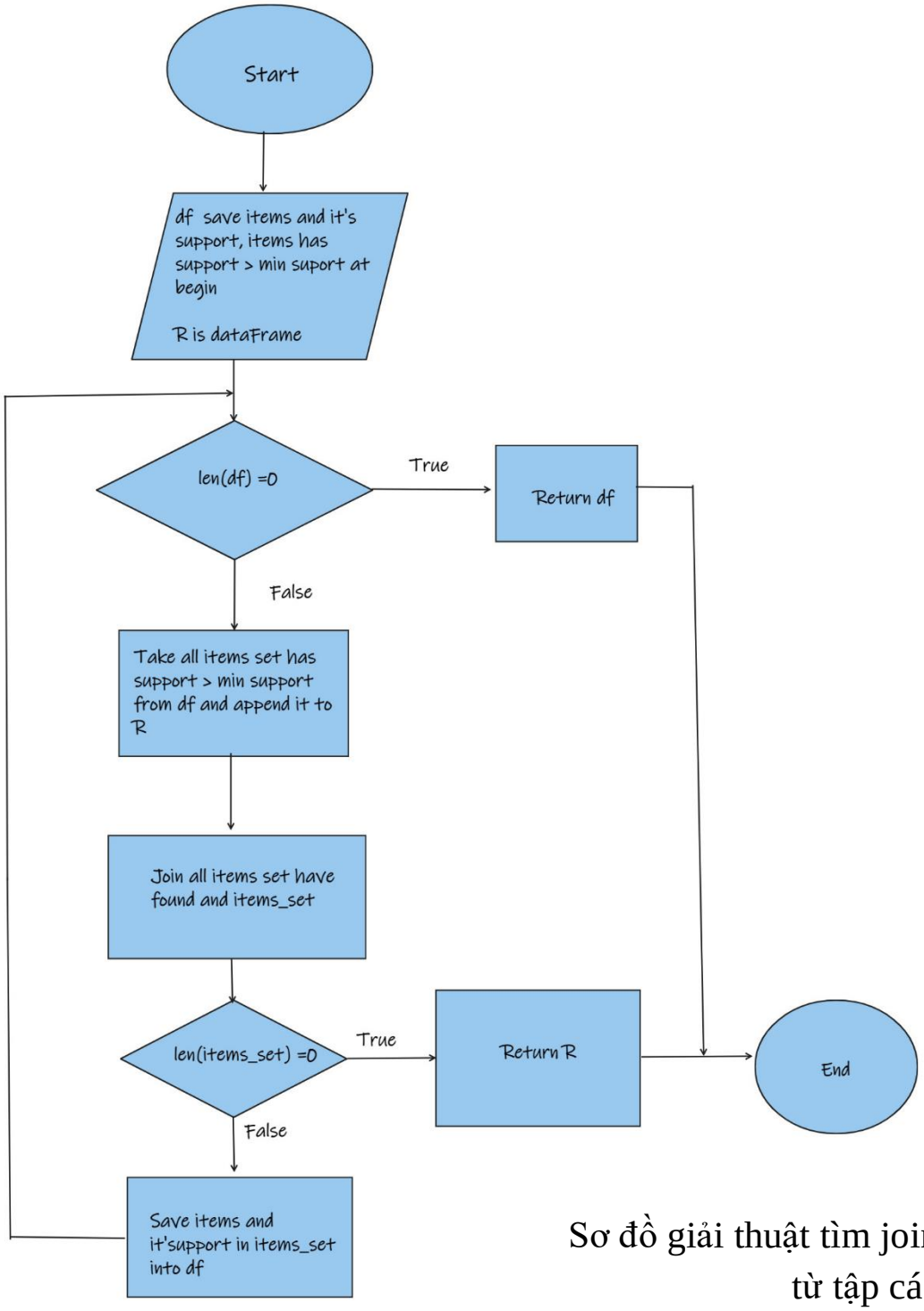
! Lưu ý: Set chứa tập các đối tượng không theo một thứ tự nào cả, các đối tượng là duy nhất và không thể thay đổi trong set. Tuy nhiên set vẫn có thể thay đổi, có thể thêm hoặc xóa bớt các đối tượng. Frozenset, cùng mang các đặc điểm như set nhưng chúng ta không thể thay đổi frozenset cũng như các phần tử trong nó.[4]

+ Dataframe được sử dụng để duyệt dữ liệu từ file Excel.

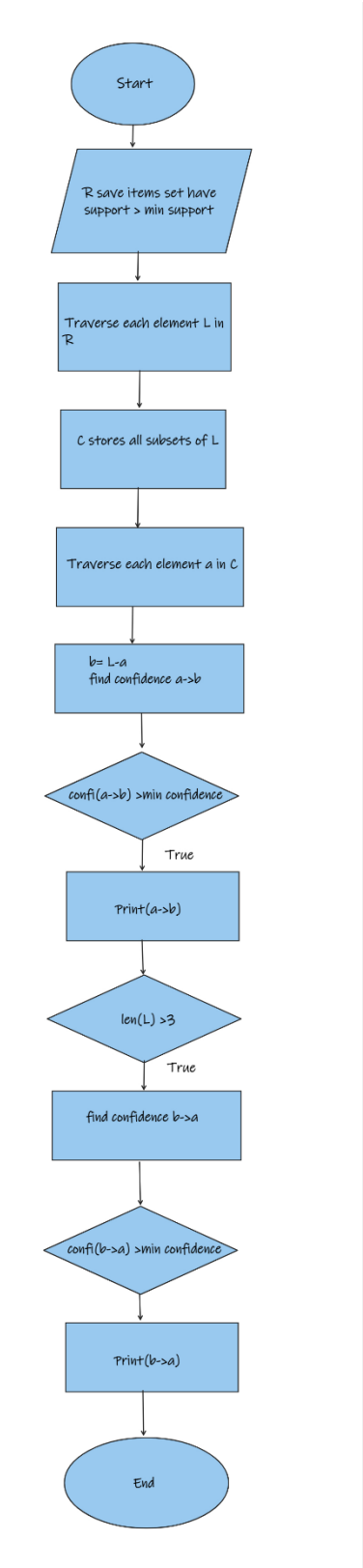
2.2 Sơ đồ giải thuật



Sơ đồ giải thuật toàn bài



Sơ đồ giải thuật tìm join items_set từ tập các items_set



Sơ đồ giải thuật in ra các Rules từ tập item_set phổ biến

2.3 Hiện thực

```
import pandas as pd
from itertools import combinations
# input:
#   TID        items
# 0  T1     I1,I2,I5
# 1  T2        I2,I4
# 2  T3        I2,I3
# 3  T4     I1,I2,I4
# 4  T5        I1,I3
# 5  T6        I2,I3
# 6  T7        I1,I3
# 7  T8  I1,I2,I3,I5
# 8  T9     I1,I2,I3

# Trả về từng list items của khách hàng
def dataSet(file_name):
    data_set=pd.read_excel(file_name+'.xlsx')
    items =data_set['items'].str.split(",")
    return items
# 0        [I1, I2, I5]
# 1            [I2, I4]
# 2            [I2, I3]
# 3        [I1, I2, I4]
# 4            [I1, I3]
# 5            [I2, I3]
# 6            [I1, I3]
# 7    [I1, I2, I3, I5]
# 8        [I1, I2, I3]

#Trả về số support của mỗi mặt hàng
def count_item(items):
    count_ind_item = {}
    for row in items:
        for i in range(len(row)):
            count_ind_item[row[i]]=count_ind_item.get(row[i],0)+1
    data = pd.DataFrame()
    data['item_sets'] = count_ind_item.keys()
    data['supp_count'] = count_ind_item.values()
    data = data.sort_values('item_sets')
    return data
```

```
#   item_sets  supp_count
# 0        I1           6
# 1        I2           7
# 4        I3           6
# 3        I4           2
# 2        I5           2

# Số support của các item_set
def count_itemset(items, itemsets):
    count_item = {}
    for item_set in itemsets:
        set_A = set(item_set)
        item_set=tuple(item_set)
        for row in items:
            set_B = set(row)
            if set_B.intersection(set_A) == set_A:
                count_item[item_set]=count_item.get(item_set,0)+1
    data = pd.DataFrame()
    data['item_sets'] = count_item.keys()
    data['supp_count'] = count_item.values()
    return data
# print(count_itemset(items,[['I1','I2'],['I2','I5']]))
#output:
#   item_sets  supp_count
# 0  (I1, I2)           4
# 1  (I2, I5)           2

#Kiểm tra xem item_set có supp lớn hơn min supp hay không
def check_support(data,supp):
    df = data[data.supp_count >= supp]
    return df
#Hàm tạo ra các item_set từ tập các items
def join(list_of_items):
    itemsets=[]
    i=1
    for entry in list_of_items:
        proceding_items=list_of_items[i:]
        for item in proceding_items:
            if (type(item) is str):
                if entry!=item:
                    tuples=(entry,item)
                    itemsets.append(tuples)
```

```
            else:
                if entry[0:-1] ==item[0:-1]:
                    tuples=entry+item[1:]
                    itemsets.append(set(tuples))
        i+=1
    if len(itemsets)==0:
        return None
    return itemsets
# VD1:
# l=['I1','I2','I3','I4','I5']
# print(join(l))
# Output : [('I1', 'I2'), ('I1', 'I3'), ('I1', 'I4'), ('I1', 'I5'),
#           ('I2', 'I3'), ('I2', 'I4'), ('I2', 'I5'), ('I3', 'I4'), ('I3', 'I5'),
('I4', 'I5')]
# VD2:
# s=[('I1','I2'),('I1','I3'),('I1','I5'),('I2','I3'),('I2','I4'),('I2','I5')]
# print(join(s))
# [('I1', 'I2', 'I3'), ('I1', 'I2', 'I5'), ('I1', 'I3', 'I5'), ('I2', 'I3', 'I4'),
('I2', 'I3', 'I5'), ('I2', 'I4', 'I5')]

def apriori(items,supp):
    freq=pd.DataFrame()
    R=pd.DataFrame()
    df=count_item(items)
    while len(df)!=0:
        df=check_support(df,supp)
        R=pd.concat([R,df])
        if len(df)>1 or (len(df)==1 and int(df.supp_count)>=supp):
            freq=df
        itemsets=join(df.item_sets)
        if (itemsets is None):
            return R
        df=count_itemset(items,itemsets)
    return df
#Tính số support của từng tập itemset kiểu frozenset
def find_supp(frozen_set_of_items):
    global items
    count=0
    for i in items:
        if frozen_set_of_items.issubset(i) :
            count+=1
    return count
```

```
if __name__=="__main__":
    items=dataSet(input("Enter file_data'names : "))
    min_supp=(int(input("Enter minimum support(%) :"))*len(items))/100
    min_confidence=int(input("Enter minimum confidence(%) :"))
    l=apriori(items,min_supp)
    l=l['item_sets']
    for L in l:
        if type(L) is not str:
            C = [frozenset(item) for item in combinations(L, len(L) - 1)]
            for a in C:
                b = frozenset(L) - a
                set_ab = frozenset(L)
                supp_ab = find_supp(set_ab)
                supp_a = find_supp(a)
                supp_b = find_supp(b)
                confidence_1 = supp_ab / supp_a * 100
                if (confidence_1 >= min_confidence):
                    print(str(list(a)) + ' -> ' + str(list(b)) + ' = ' +
str(round(confidence_1)) + '%')
                if len(L)>=3:
                    confidence_2 = supp_ab / supp_b * 100
                    if (confidence_2 >= min_confidence):
                        print(str(list(b)) + ' -> ' + str(list(a)) + ' = ' +
str(round(confidence_2)) + '%')
                print()
```

2.4 Kết quả và thảo luận

Kết quả test case 1:

```
Enter file_data'names : as1
Enter minimum support(%) :50
Enter minimum confidence(%) :50
['I1'] -> ['I3'] = 100%

['I3'] -> ['I1'] = 67%

['I2'] -> ['I3'] = 67%

['I3'] -> ['I2'] = 67%

['I2'] -> ['I5'] = 100%

['I5'] -> ['I2'] = 100%

['I3'] -> ['I5'] = 67%

['I5'] -> ['I3'] = 67%

['I5', 'I2'] -> ['I3'] = 67%
['I3'] -> ['I5', 'I2'] = 67%

['I5', 'I3'] -> ['I2'] = 100%
['I2'] -> ['I5', 'I3'] = 67%

['I2', 'I3'] -> ['I5'] = 100%
['I5'] -> ['I2', 'I3'] = 67%
```

Kết quả tool weka test case 1:

```
Best rules found:

 1. I5=yes 3 ==> I2=yes 3    <conf:(1)> lift:(1.33) lev:(0.19) [0] conv:(0.75)
 2. I2=yes 3 ==> I5=yes 3    <conf:(1)> lift:(1.33) lev:(0.19) [0] conv:(0.75)
 3. I1=yes 2 ==> I3=yes 2    <conf:(1)> lift:(1.33) lev:(0.13) [0] conv:(0.5)
 4. I3=yes I5=yes 2 ==> I2=yes 2    <conf:(1)> lift:(1.33) lev:(0.13) [0] conv:(0.5)
 5. I2=yes I3=yes 2 ==> I5=yes 2    <conf:(1)> lift:(1.33) lev:(0.13) [0] conv:(0.5)
 6. I3=yes 3 ==> I1=yes 2    <conf:(0.67)> lift:(1.33) lev:(0.13) [0] conv:(0.75)
 7. I3=yes 3 ==> I2=yes 2    <conf:(0.67)> lift:(0.89) lev:(-0.06) [0] conv:(0.38)
 8. I2=yes 3 ==> I3=yes 2    <conf:(0.67)> lift:(0.89) lev:(-0.06) [0] conv:(0.38)
 9. I5=yes 3 ==> I3=yes 2    <conf:(0.67)> lift:(0.89) lev:(-0.06) [0] conv:(0.38)
10. I3=yes 3 ==> I5=yes 2    <conf:(0.67)> lift:(0.89) lev:(-0.06) [0] conv:(0.38)
11. I2=yes I5=yes 3 ==> I3=yes 2    <conf:(0.67)> lift:(0.89) lev:(-0.06) [0] conv:(0.38)
12. I5=yes 3 ==> I2=yes I3=yes 2    <conf:(0.67)> lift:(1.33) lev:(0.13) [0] conv:(0.75)
13. I3=yes 3 ==> I2=yes I5=yes 2    <conf:(0.67)> lift:(0.89) lev:(-0.06) [0] conv:(0.38)
14. I2=yes 3 ==> I3=yes I5=yes 2    <conf:(0.67)> lift:(1.33) lev:(0.13) [0] conv:(0.75)
```

Kết quả test case 2:

```
Enter file_data'names : as2
Enter minimum support(%) :20
Enter minimum confidence(%) :50
['I1'] -> ['I2'] = 67%

['I2'] -> ['I1'] = 57%

['I1'] -> ['I3'] = 67%

['I3'] -> ['I1'] = 67%


['I5'] -> ['I1'] = 100%

['I2'] -> ['I3'] = 57%

['I3'] -> ['I2'] = 67%


['I4'] -> ['I2'] = 100%


['I5'] -> ['I2'] = 100%

['I1', 'I3'] -> ['I2'] = 50%

['I1', 'I2'] -> ['I3'] = 50%

['I3', 'I2'] -> ['I1'] = 50%

['I1', 'I5'] -> ['I2'] = 100%

['I1', 'I2'] -> ['I5'] = 50%
['I5'] -> ['I1', 'I2'] = 100%

['I5', 'I2'] -> ['I1'] = 100%
```

Kết quả tool weka test case 2:

```
Best rules found:

 1. 5=YES 2 ==> 1=YES 2    <conf:(1)> lift:(1.5) lev:(0.07) [0] conv:(0.67)
 2. 4=YES 2 ==> 2=YES 2    <conf:(1)> lift:(1.29) lev:(0.05) [0] conv:(0.44)
 3. 5=YES 2 ==> 2=YES 2    <conf:(1)> lift:(1.29) lev:(0.05) [0] conv:(0.44)
 4. 2=YES 5=YES 2 ==> 1=YES 2    <conf:(1)> lift:(1.5) lev:(0.07) [0] conv:(0.67)
 5. 1=YES 5=YES 2 ==> 2=YES 2    <conf:(1)> lift:(1.29) lev:(0.05) [0] conv:(0.44)
 6. 5=YES 2 ==> 1=YES 2=YES 2    <conf:(1)> lift:(2.25) lev:(0.12) [1] conv:(1.11)
 7. 1=YES 6 ==> 2=YES 4    <conf:(0.67)> lift:(0.86) lev:(-0.07) [0] conv:(0.44)
 8. 3=YES 6 ==> 1=YES 4    <conf:(0.67)> lift:(1) lev:(0) [0] conv:(0.67)
 9. 1=YES 6 ==> 3=YES 4    <conf:(0.67)> lift:(1) lev:(0) [0] conv:(0.67)
10. 3=YES 6 ==> 2=YES 4    <conf:(0.67)> lift:(0.86) lev:(-0.07) [0] conv:(0.44)
11. 2=YES 7 ==> 1=YES 4    <conf:(0.57)> lift:(0.86) lev:(-0.07) [0] conv:(0.58)
12. 2=YES 7 ==> 3=YES 4    <conf:(0.57)> lift:(0.86) lev:(-0.07) [0] conv:(0.58)
13. 2=YES 3=YES 4 ==> 1=YES 2    <conf:(0.5)> lift:(0.75) lev:(-0.07) [0] conv:(0.44)
14. 1=YES 3=YES 4 ==> 2=YES 2    <conf:(0.5)> lift:(0.64) lev:(-0.12) [-1] conv:(0.3)
15. 1=YES 2=YES 4 ==> 3=YES 2    <conf:(0.5)> lift:(0.75) lev:(-0.07) [0] conv:(0.44)
16. 1=YES 2=YES 4 ==> 5=YES 2    <conf:(0.5)> lift:(2.25) lev:(0.12) [1] conv:(1.04)
```

- Đánh giá kết quả:
  + Kết quả của chúng em thỏa mãn yêu cầu bài toán.
  + Kết quả của chúng em hoàn toàn trùng khớp với kết quả của tool weka.
- Thảo luận bài toán:
  Nhược điểm:
  + Phải duyệt CSDL nhiều lần.
  + Số lượng tập ứng viên rất lớn.
  + Thực hiện độ phổ biến nhiều, đơn điệu.
  + Chỉ được dùng để phát hiện các luật kết hợp dạng khẳng định (X => Y) chứ không thể phát hiện các luật kết hợp ở dạng phủ định (X => ⌐Y).
  Ưu điểm:
  + Đơn giản dễ hiểu và dễ áp dụng.
  + Được sử dụng để tính toán các tập phổ biến lớn.

3. Kết luận

Qua quá trình làm bài này nhóm em rút ra được khá nhiều bài học:

+ Nhận ra khả năng làm việc nhóm online còn chưa tốt và đã được cải thiện rất nhiều thông qua lần làm việc này.
+ Biết thêm được một giải thuật mới là apriori và các cấu trúc dữ liệu mới như dataframe.
+ Hiểu sâu hơn về các cấu trúc dữ liệu cơ bản như set, dict,… và các hàm cơ bản của python.
+ Biết thêm nhiều thư viện mới của python như pandas, combination,…
+ Rèn luyện khả năng tư duy, tìm kiếm và nghiên cứu tài liệu trên nền tảng internet.
+ Hiểu hơn về AI đặc biệt là data mining.
+ Giúp chúng em trở nên hứng thú và muốn khai thác nhiều hơn về lĩnh vực AI.

TỰ ĐÁNH GIÁ – ĐỀ TÀI NHÓM

Nội dung	Điểm	Ghi chú
Giới thiệu về bài toán (0.25 đ)	**0.25**	
Mô tả cấu trúc dữ liệu (1.25 đ)	**1.0**	
Sơ đồ giải thuật (1.5 đ)	**1.25**	
Hiện thực (5 đ)	**4.75**	
Kết quả và thảo luận (0.5 đ)	**0.5**	
Điểm nhóm (0.75 đ)	**0.5**	
Các điều rút ra cho bản thân (0.25 đ)	**0.25**	
Báo cáo đúng theo mẫu (0.5 đ)	**0.5**	
Tổng điểm	**9.0**	

TÀI LIỆU THAM KHẢO

1. Thuật toán Apriori khai phá luật kết hợp trong Data Mining

   Tác giả: Nguyen Minh Duc

2. Giải thích thuật toán Apriori

   Tác giả: Edureka

3. Apriori Algorithm from Scratch

   Tác giả: BISMA KHAN

4. Set và Frozenset

   Tác giả: Cuong Tran