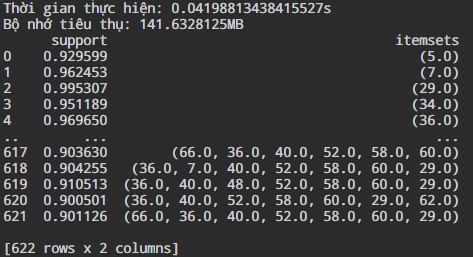
**Mô tả tập dữ liệu**

* **Chess.dat** là tập dữ liệu gồm 37 cột và 3196 bộ dữ liệu gồm các số tự nhiên.
* **Mushroom.dat** là tập dữ liệu gồm 23 cột và 8124 bộ dữ liệu gồm các số tự nhiên.

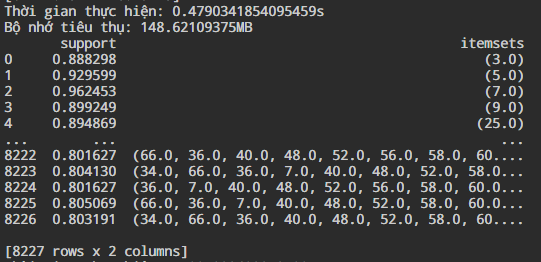
**Mô tả phân tích ưu và nhược điểm của từng hàm khai thác tập phổ biến trên giữa các thư viện**

Trong file báo cáo này em sử dụng 2 thư viện khai thác tập phổ biến là **mlxtend** và **pyfpgrowth** dựa trên 2 thuật toán là **Apriori** và **FP-Growth**

* **Thuật toán Apriori**
  + Chess.dat thực nghiệm trên 5 giá trị *minsup* khác nhau gồm:
    - *minsup* = 0.9
      * Số lượng tập phổ biến gồm: 622 tập
      * Thời gian thực hiện: 0.042s
      * Bộ nhớ tiêu thụ: 141.63MB

**

* + - *minsup* = 0.8
      * Số lượng tập phổ biến gồm: 8227 tập
      * Thời gian thực hiện: 0.479s
      * Bộ nhớ tiêu thụ: 148.62MB

**

* + - *minsup* = 0.7
      * Số lượng tập phổ biến gồm: 48731 tập
      * Thời gian thực hiện: 4.29s
      * Bộ nhớ tiêu thụ: 179.65MB

*A screenshot of a computer

Description automatically generated*

* + - *minsup* = 0.6
      * Số lượng tập phổ biến gồm: 254944 tập
      * Thời gian thực hiện: 30s
      * Bộ nhớ tiêu thụ: 328.45MB

*A screenshot of a computer

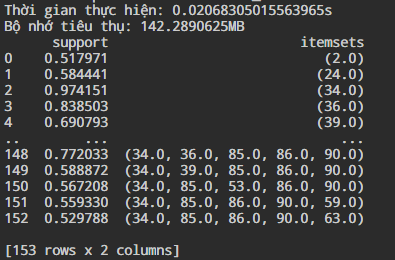
Description automatically generated*

* + - *minsup* = 0.55
      * Số lượng tập phổ biến gồm: 574998 tập
      * Thời gian thực hiện: 281.63s
      * Bộ nhớ tiêu thụ: 517.47 MB

*A screenshot of a computer

Description automatically generated*

* + Mushroom.dat thực nghiệm trên 5 giá trị *minsup* khác nhau gồm:
    - *minsup* = 0.5
      * Số lượng tập phổ biến gồm: 153 tập
      * Thời gian thực hiện: 0.02s
      * Bộ nhớ tiêu thụ: 142.28MB

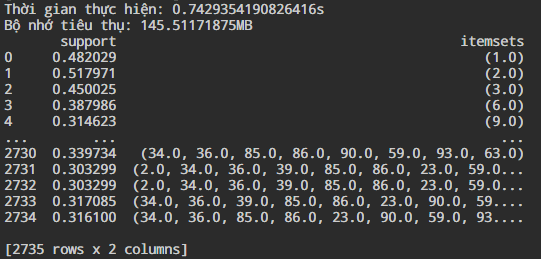
**

* + - *minsup* = 0.4
      * Số lượng tập phổ biến gồm: 565 tập
      * Thời gian thực hiện: 0.097s
      * Bộ nhớ tiêu thụ: 142.83MB

*A screenshot of a computer

Description automatically generated*

* + - *minsup* = 0.3
      * Số lượng tập phổ biến gồm: 2735 tập
      * Thời gian thực hiện: 0.74s
      * Bộ nhớ tiêu thụ: 145.51 MB

**

* + - *minsup* = 0.2
      * Số lượng tập phổ biến gồm: 53583 tập
      * Thời gian thực hiện: 16.16s
      * Bộ nhớ tiêu thụ: 184.01MB

*A screenshot of a computer

Description automatically generated*

* + - *minsup* = 0.15
      * Số lượng tập phổ biến gồm: 98575 tập
      * Thời gian thực hiện: 57.28s
      * Bộ nhớ tiêu thụ: 168.64MB

***A screenshot of a computer

Description automatically generated***

* **Thuật toán FP-Growth**
  + Chess.dat thực nghiệm trên 5 giá trị *minsup* khác nhau gồm:
    - *minsup* = 0.9
      * Số lượng tập phổ biến gồm: 622 tập
      * Thời gian thực hiện: 0.078s
      * Bộ nhớ tiêu thụ: 141.07MB

*A computer screen shot of numbers and letters

Description automatically generated*

* + - *minsup* = 0.8
      * Số lượng tập phổ biến gồm: 8227 tập
      * Thời gian thực hiện: 0.119s
      * Bộ nhớ tiêu thụ: 151.72MB

*A computer screen shot of a black screen

Description automatically generated*

* + - *minsup* = 0.7
      * Số lượng tập phổ biến gồm: 48731 tập
      * Thời gian thực hiện: 0.66s
      * Bộ nhớ tiêu thụ: 204.06MB

*A screenshot of a computer

Description automatically generated*

* + - *minsup* = 0.6
      * Số lượng tập phổ biến gồm: 254944 tập
      * Thời gian thực hiện: 3.2s
      * Bộ nhớ tiêu thụ: 374.85MB

*A computer screen shot of numbers and letters

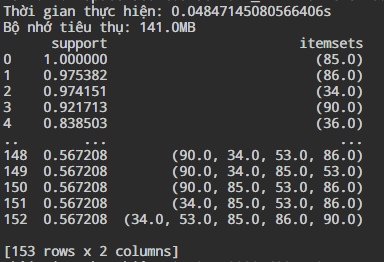
Description automatically generated*

* + - *minsup* = 0.55
      * Số lượng tập phổ biến gồm: 574998 tập
      * Thời gian thực hiện: 6.83s
      * Bộ nhớ tiêu thụ: 623.85MB

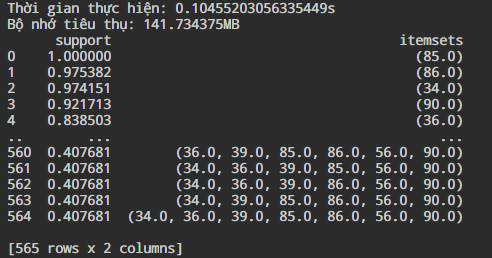
*A screenshot of a computer

Description automatically generated*

* + Mushroom.dat thực nghiệm trên 5 giá trị *minsup* khác nhau gồm:
    - *minsup* = 0.5
      * Số lượng tập phổ biến gồm: 153 tập
      * Thời gian thực hiện: 0.048s
      * Bộ nhớ tiêu thụ: 141MB

**

* + - *minsup* = 0.4
      * Số lượng tập phổ biến gồm: 565 tập
      * Thời gian thực hiện: 0.1s
      * Bộ nhớ tiêu thụ: 141.73MB

**

* + - *minsup* = 0.3
      * Số lượng tập phổ biến gồm: 2735 tập
      * Thời gian thực hiện: 0.08s
      * Bộ nhớ tiêu thụ: 144.92MB

*A screenshot of a computer

Description automatically generated*

* + - *minsup* = 0.2
      * Số lượng tập phổ biến gồm: 53583 tập
      * Thời gian thực hiện: 0.417s
      * Bộ nhớ tiêu thụ: 189.64MB

*A screenshot of a computer

Description automatically generated*

* + - *minsup* = 0.15
      * Số lượng tập phổ biến gồm: 98575 tập
      * Thời gian thực hiện: 0.826s
      * Bộ nhớ tiêu thụ: 225.47MB

***A screenshot of a computer

Description automatically generated***

* **Kết luận**

Vậy thông qua một số kết quả thực nghiệm trên, thuật toán FP-Growth cho thấy tốc độ xử lý nhanh và vượt trội hơn so với thuật toán Apriori ở ngưỡng các giá trị *minsup* thấp trong mảng *minsup* được cho. Tuy nhiên, thuật toán FP-Growth lại chiếm dụng nhiều bộ nhớ hơn đôi chút so với thuật toán Apriori. Từ đó em rút ra được kết luận về ưu và nhược điểm của cả 2 thuật toán như sau:

Thuật toán Apriori:

* Ưu điểm:
  + Đơn giản để hiểu và áp dụng. Dựa trên thuộc tính Apriori để giảm phạm vi tìm kiếm.
* Nhược điểm
  + Tốn kém thời gian để tìm độ phổ biến vì phải duyệt qua toàn tập dữ liệu. Sinh ra nhiều tập ứng cử viên không cần thiết

Thuật toán FP-Growth:

* Ưu điểm:
  + Thời gian xử lý nhanh hơn rất nhiều so với thuật toán Apriori do sử dụng cấu trúc cây nhỏ gọn. Không cần sinh tập ứng cử viên. Hiệu quả hơn khi độ phổ biến tối thiểu thấp.
* Nhược điểm:
  + Tiêu tốn nhiều bộ nhớ hơn đôi chút và có thể gặp khó khăn khi dữ liệu lớn.

**Hình ảnh trực quan hóa**

**Thuật toán Apriori**

* Chess.dat

A graph with red lines

Description automatically generated

* Mushroom.dat

A graph on a white background

Description automatically generated

**Thuật toán FP-Growth**

* Chess.dat

A screen shot of a graph

Description automatically generated

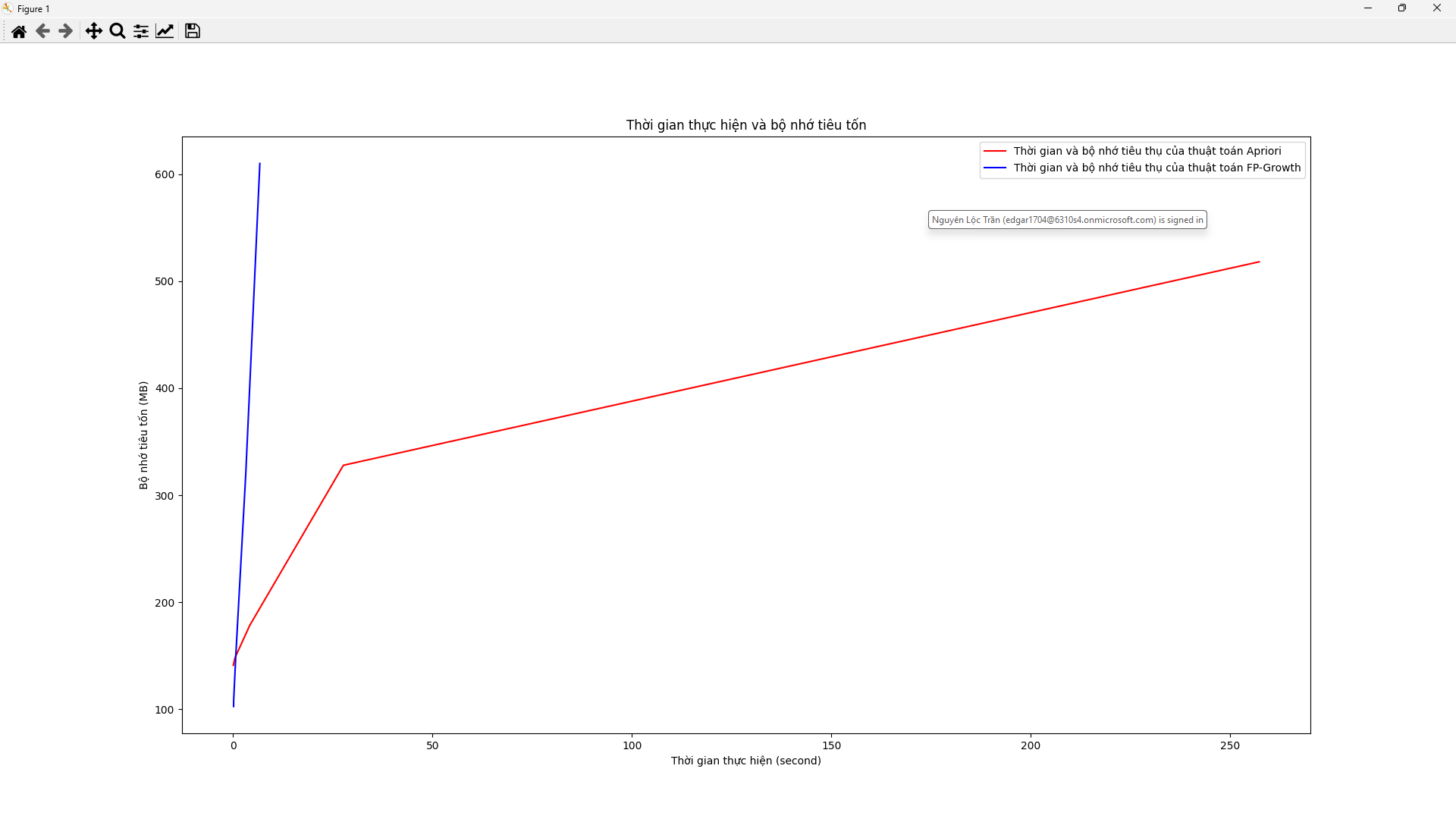
* Mushroom.dat

A graph with a red line

Description automatically generated

**So sánh giữa 2 thuật toán Apriori và FP-Growth**

* Trên tập chess.dat



* Trên tập mushroom.dat

