**Nguyễn Ngọc Ánh**

**3120410041**

**BÀI THỰC HÀNH SỐ 4**

Thư Viện APRIORI

**Mô tả dữ liệu:**

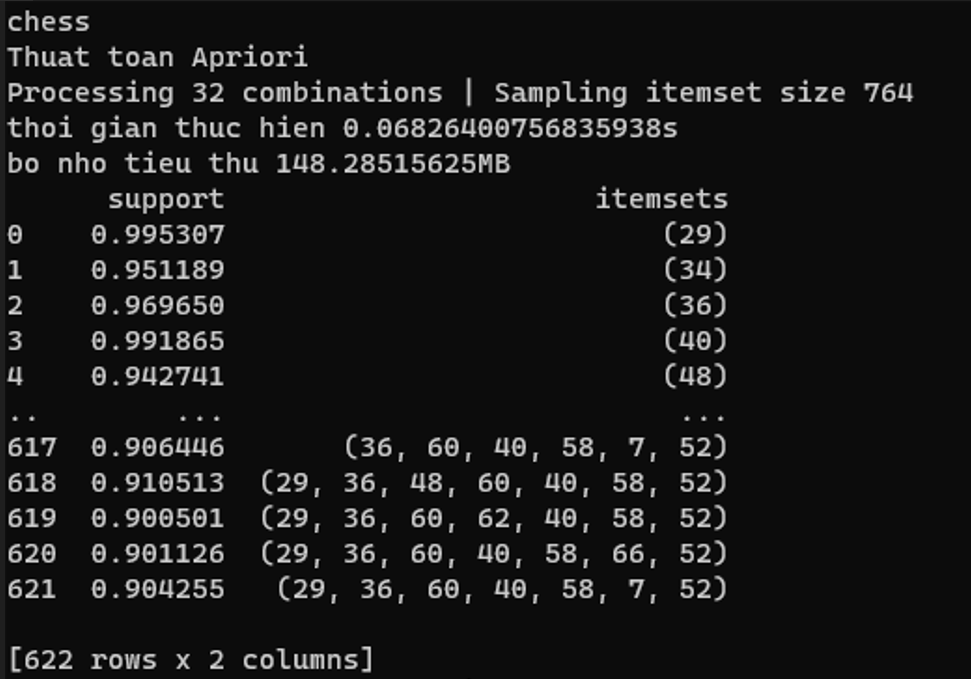
File Chess.dat là một tập dữ liệu chứa 3196 dòng × 37 cột về các transaction

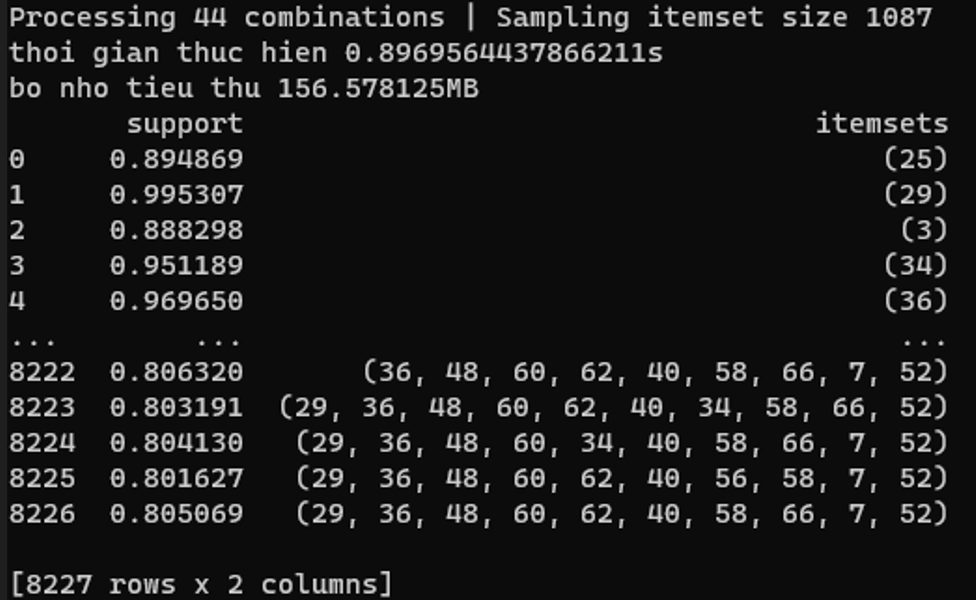
Fìle Mushroom một tập dữ liệu chứa 8124 dòng × 23 cột về các transaction

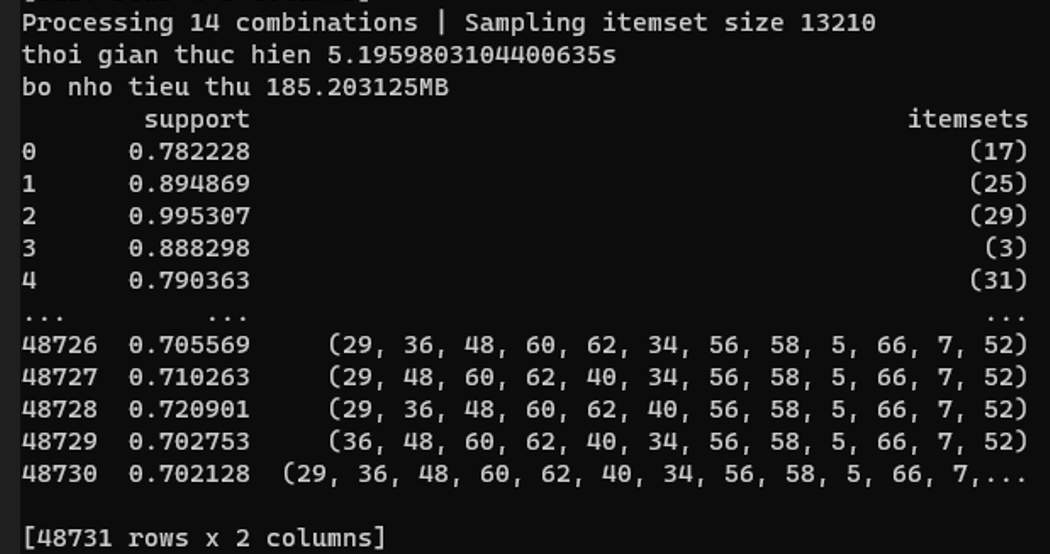
1. **Dữ liệu gốc:**

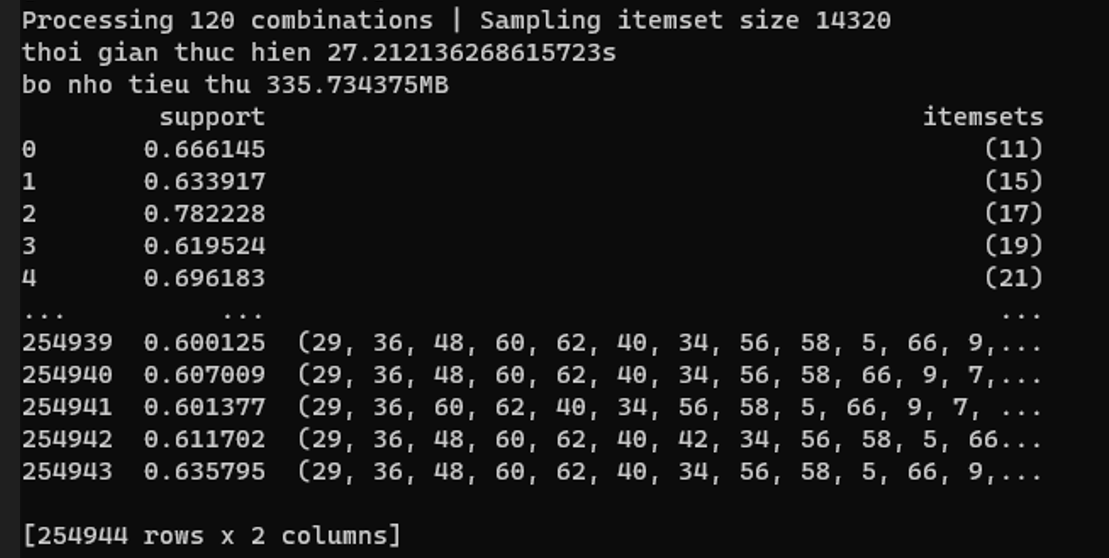
* Minsup lớn hơn thì tập phổ biến support càng lớn, giá trị cũng ít hơn, thời gian chạy và bộ nhớ tiêu thụ cũng ít hơn so với minsup nhỏ hơn

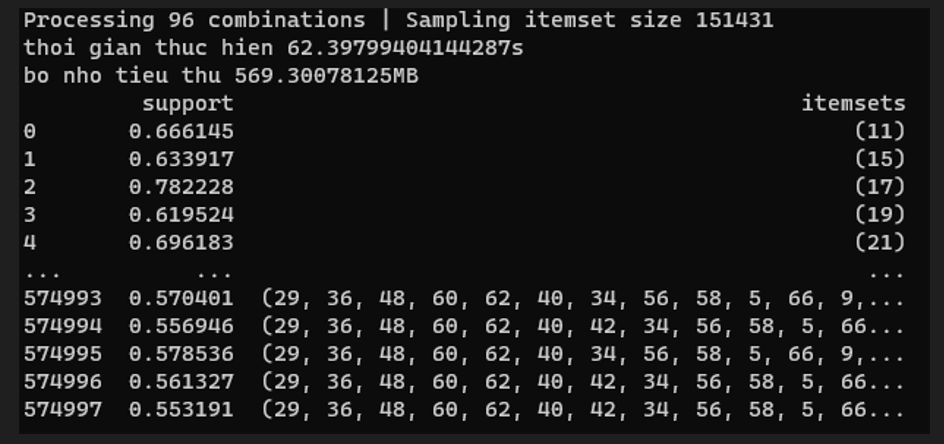
Dưới đây là sự so sánh giữa các 5 minsup khác nhau: 0.9, 0.8, 0.7, 0.6, 0.55 trên tệp Chess cùng dữ liệu gốc:



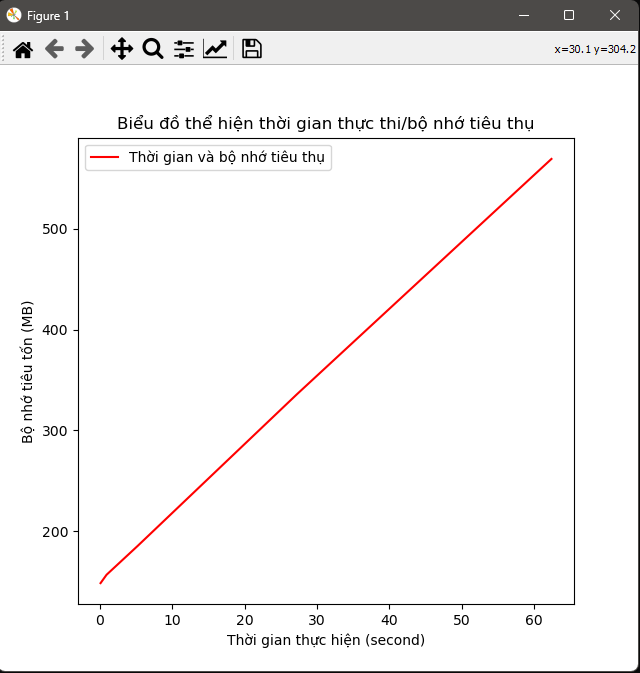


****

****

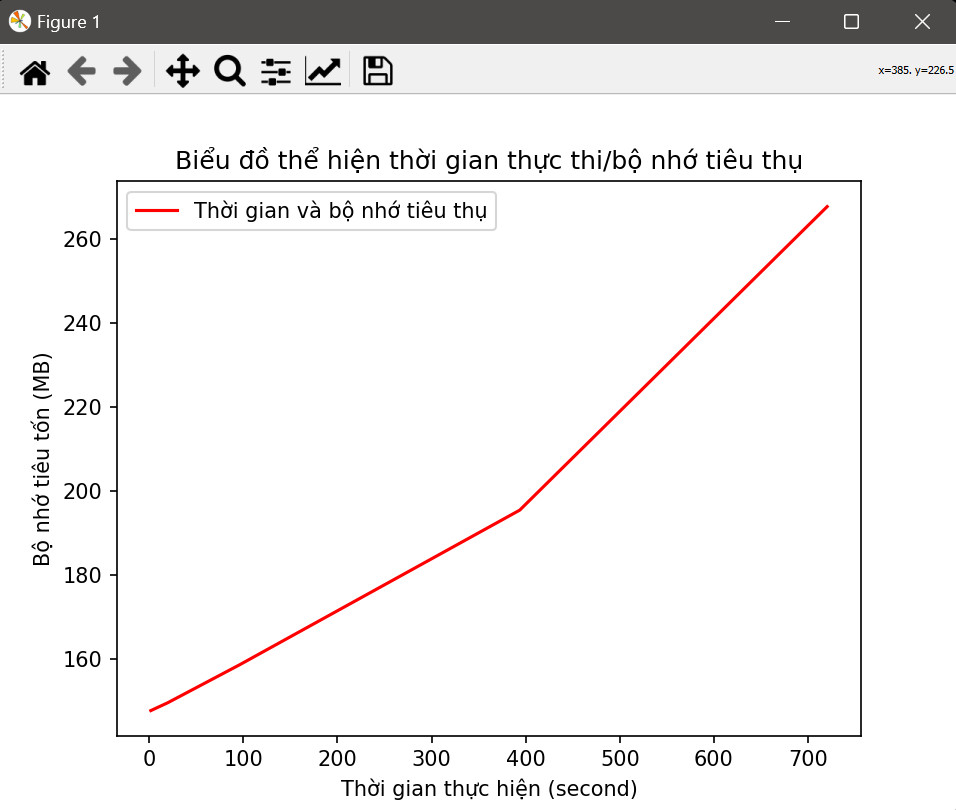
****

* Đồ thị thể hiện giữa thời gian chạy và bộ nhớ tiêu thụ khi dùng thư viện Apriori chạy

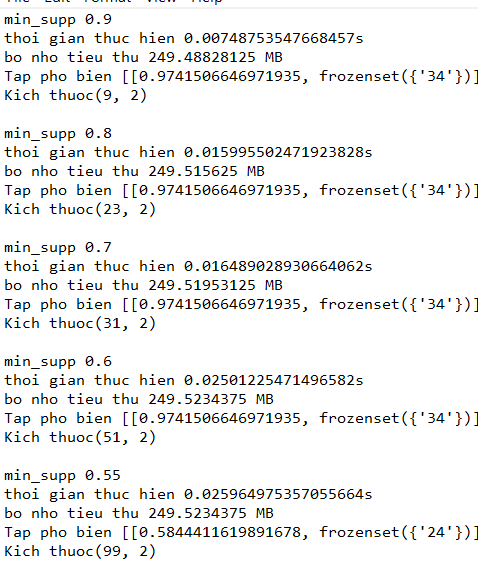


Dưới đây là sự so sánh giữa các 5 minsup khác nhau: 0.9, 0.8, 0.7, 0.6, 0.55 trên tệp Mushroom cùng dữ liệu gốc:

* Đồ thị thể hiện giữa thời gian chạy và bộ nhớ tiêu thụ khi dùng thư viện Apriori chạy



**3. Dataframe lọc item thoả ngưỡng minsup trước khi khai thác tập phổ biến**

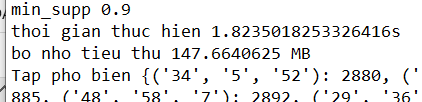


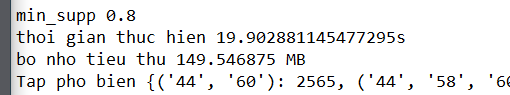
Với dữ liệu được lọc trước rồi tìm tập phổ biến thì thời gian chạy minsup rất nhỏ nhưng minsup nhỏ thì thời gian và bộ nhớ tiêu thụ vẫn nhiều hơn

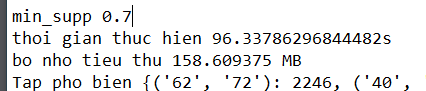
Thư Viện Pyfpgrowth

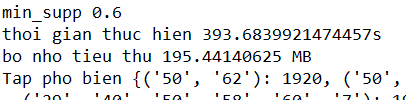
1. Dữ liệu gốc

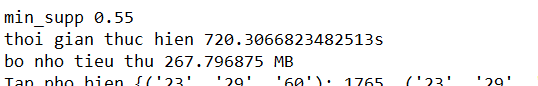
File chess



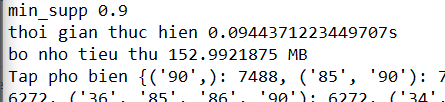


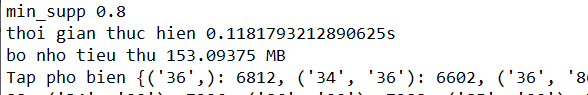


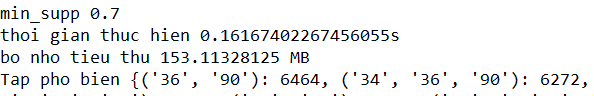


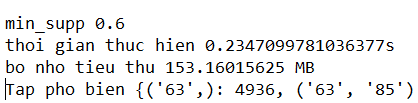


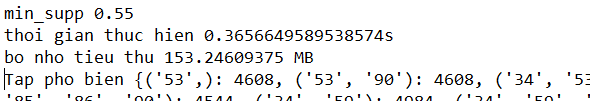
File Mushroom







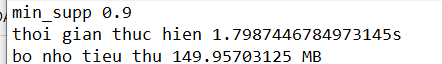


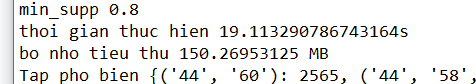


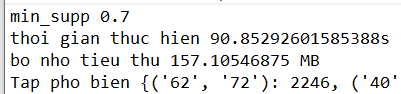
Khi sủ dụng thư viện này thì độ minsup càng nhỏ thời gian chạy cũng lâu hơn dữ liệu lớn hơn lại chạy nhanh hơn và sử dụng bộ nhớ thấp hầu như gần bằng nhau và tập phổ biến cũng không nhiều

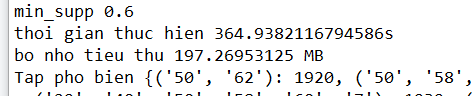
1. Dữ liệu được sắp xếp theo item tăng dần độ phổ biến

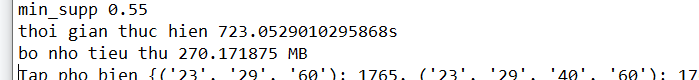
File Chess



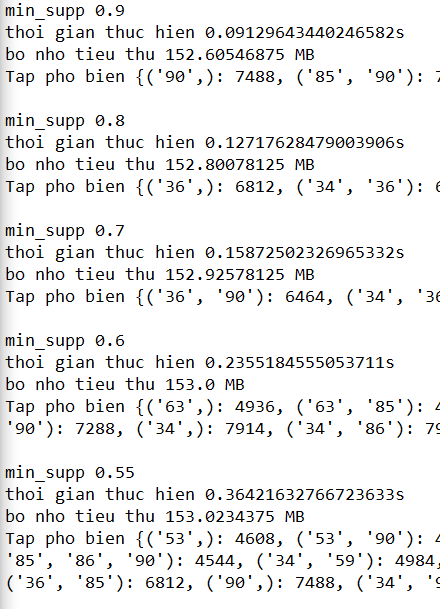








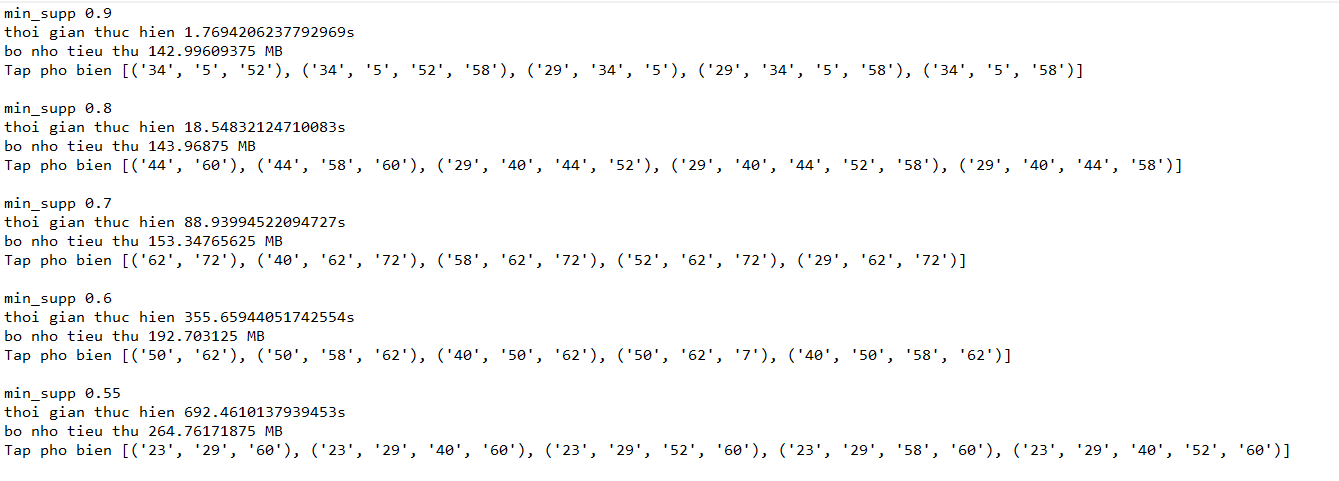
File Mushroom



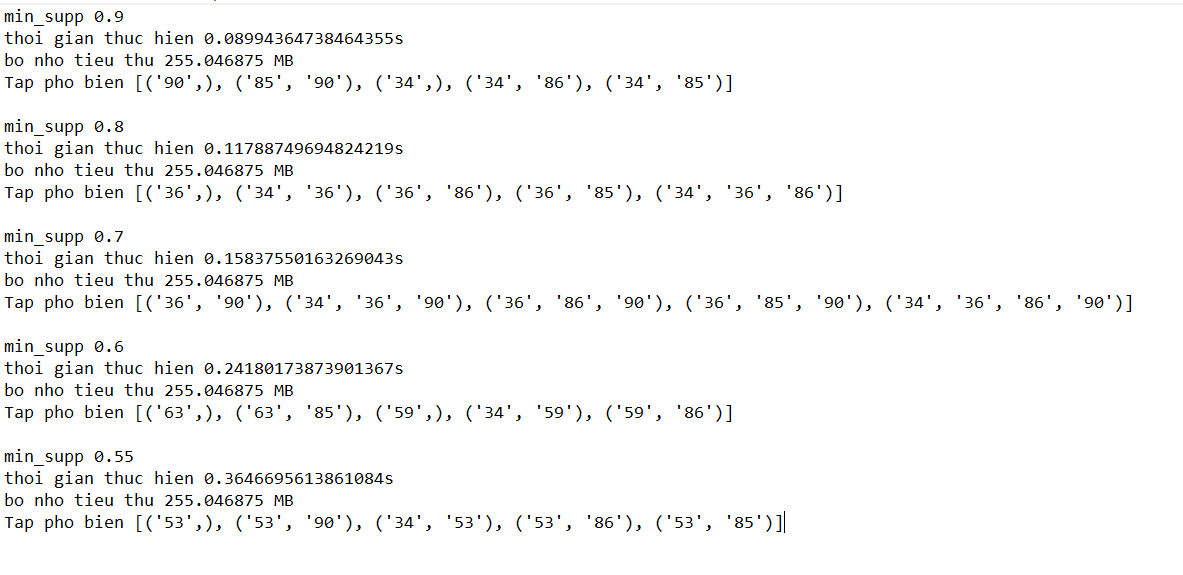
Với dữ liệu càng khó hơn thì thời gian chạy và bộ nhớ cũng tăng nhiều hơn so với dữ liệu gốc và dữ liệu nhiều hơn thì cho độ phổ biến ít hơn nên thời gian chạy nhanh hơn và bộ nhớ tiêu thụ cũng ít hơn

1. Dataframe lọc item thoả ngưỡng minsup trước khi khai thác tập phổ biến

File Chess



File Mushroom



Với dữ liệu càng khó hơn thì thời gian chạy và bộ nhớ cũng tăng nhiều hơn so với dữ liệu gốc và dữ liệu nhiều hơn thì cho độ phổ biến ít hơn nên thời gian chạy nhanh hơn và bộ nhớ tiêu thụ cũng ít hơn

* **So sánh ưu và nhược giữa hai thư viện:**

| Thư viện | Apriori | Pyfpgrowth |
| --- | --- | --- |
| Kỹ thuật khai thác | Lặp | FP-Tree |
| Ưu điểm | Đơn giản, hiệu quả cho tập dữ liệu có độ phổ biến cao | Hiệu quả cho tập dữ liệu có độ phổ biến thấp, sử dụng ít bộ nhớ |
| Nhược điểm | Có thể không hiệu quả cho tập dữ liệu có độ phổ biến thấp, sử dụng nhiều bộ nhớ | Phức tạp, có thể không hiệu quả cho tập dữ liệu có độ phổ biến cao |
| Thích hợp | Tập dữ liệu có độ phổ biến cao | Tập dữ liệu có độ phổ biến thấp |