**PHÂN TÍCH DỮ LIỆU**

**Họ tên:** Nguyễn Văn Vũ Em

**Mssv:** 3122410090

1. **Mô tả dữ liệu**

**Dataset 1: Chess**

* Giới thiệu về Dataset:
  + Dữ liệu này có thể là thông tin về các nước đi hoặc vị trí trên bàn cờ trong trò chơi cờ vua. Tuy nhiên, do thiếu thông tin chi tiết, không có cột mô tả cụ thể, các giá trị chỉ là số nguyên.
* Các cột chính:
  + 1, 3, 5, ..., 74: Các cột được đánh số từ 1 đến 74 (chỉ số lẻ), có thể đại diện cho các đặc điểm liên quan đến bàn cờ hoặc nước đi trong cờ vua.
  + Dữ liệu: Mỗi cột chứa số nguyên, có thể đại diện cho trạng thái hoặc hành động của một quân cờ, hoặc vị trí trên bàn cờ.
* Ứng dụng và mục tiêu:
  + Phân tích chiến thuật: Dữ liệu có thể được dùng để phân tích các chiến thuật cờ vua dựa trên nước đi hoặc trạng thái bàn cờ.
  + Huấn luyện mô hình dự đoán: Sử dụng các đặc điểm này để huấn luyện mô hình dự đoán kết quả trận đấu, hoặc phân loại trạng thái bàn cờ.
* Các bước tiền xử lý đề xuất:
  + Kiểm tra dữ liệu: Kiểm tra các giá trị thiếu hoặc không hợp lệ trong dữ liệu.
  + Chuyển đổi dữ liệu: Tạo ra các biến mới, chẳng hạn như các trạng thái quan trọng của bàn cờ (quân vua, quân hậu, v.v.), hoặc tổng kết quả của mỗi trò chơi.

**Dataset 2: Mushroom**

* Giới thiệu về Dataset:
  + Bộ dữ liệu này có thể liên quan đến các đặc điểm của nấm và có thể được dùng để phân loại nấm dựa trên các đặc tính khác nhau (mặc dù không có mô tả chi tiết về các cột).
* Các cột chính:
  + 1, 3, 9, 13, ..., 113: Các cột được đánh số từ 1 đến 113, mỗi cột chứa các giá trị số nguyên.
  + Dữ liệu: Mỗi hàng có thể đại diện cho một loại nấm với các đặc điểm được mã hóa thành số, chẳng hạn như kích thước, hình dạng, màu sắc, hoặc tính độc/hữu ích của nấm.
* Ứng dụng và mục tiêu:
  + Phân loại nấm: Dữ liệu có thể được sử dụng để phân loại các loại nấm khác nhau dựa trên các đặc điểm sinh học hoặc hóa học.
  + Xây dựng mô hình học máy: Sử dụng dữ liệu này để xây dựng mô hình dự đoán xem nấm có độc hay không, hoặc xác định các loại nấm dựa trên đặc điểm sinh thái.
* Các bước tiền xử lý đề xuất:
  + Xử lý các giá trị thiếu: Kiểm tra và loại bỏ hoặc thay thế các giá trị thiếu.
  + Chuyển đổi dữ liệu: Mã hóa lại các giá trị số thành các đặc tính dễ hiểu hơn (ví dụ: màu sắc, kích thước, trạng thái sinh trưởng).
  + Phân tích phân phối dữ liệu: Tạo biểu đồ phân phối để xem xét mối tương quan giữa các đặc điểm và loại nấm.

1. **Mô tả phân tích ưu điểm và nhược điểm**
   1. **Thư viện Apriori từ mlxtend**

* Ưu điểm:
* Đơn giản và dễ hiểu: Apriori là một thuật toán khai thác tập phổ biến (frequent itemset mining) cơ bản và phổ biến nhất, dễ hiểu và dễ triển khai.
* Tính ổn định: Thuật toán dựa trên việc quét dữ liệu nhiều lần để tìm các tập phổ biến, do đó cho ra kết quả chính xác và ổn định.
* Tính mở rộng: Nó có thể dễ dàng được tùy chỉnh với các tham số như ngưỡng hỗ trợ tối thiểu (min\_support) và nó có thể được áp dụng cho nhiều bài toán khác nhau như phân tích rổ hàng (market basket analysis).
* Nhược điểm:
* Hiệu suất kém với dữ liệu lớn: Vì thuật toán Apriori quét toàn bộ dữ liệu nhiều lần để tính toán các tập phổ biến, nó có thể trở nên chậm chạp và không hiệu quả khi áp dụng cho các tập dữ liệu lớn hoặc phức tạp.
* Tốn tài nguyên bộ nhớ: Do tính chất quét toàn bộ dữ liệu nhiều lần, thuật toán có thể tiêu thụ nhiều bộ nhớ, đặc biệt khi số lượng các mục lớn và mức độ kết hợp cao.
* Số lần quét nhiều: Mỗi mức kết hợp (1-itemset, 2-itemset,...) yêu cầu một lần quét dữ liệu, dẫn đến số lượng quét dữ liệu tăng dần khi số mục tăng.
  1. **Thư viện FP-Growth từ mlxtend**
     + Ưu điểm:
       - Hiệu suất cao: FP-Growth có khả năng xử lý tốt các tập dữ liệu lớn hơn vì nó chỉ cần quét qua dữ liệu hai lần (một lần để xây dựng cây FP, một lần để khai thác các tập phổ biến). Điều này làm cho FP-Growth nhanh hơn nhiều so với Apriori trong trường hợp dữ liệu lớn.
       - Tối ưu bộ nhớ: FP-Growth sử dụng cấu trúc cây FP để lưu trữ dữ liệu dưới dạng tần suất, giảm nhu cầu phải quét lại toàn bộ dữ liệu nhiều lần. Điều này giúp giảm bớt bộ nhớ sử dụng so với thuật toán Apriori.
       - Tính toán nhanh với dữ liệu dày đặc: Thuật toán FP-Growth đặc biệt hiệu quả trong việc khai thác tập phổ biến từ các tập dữ liệu dày đặc (có nhiều tập con phổ biến).
     + Nhược điểm:
       - Phức tạp hơn: FP-Growth có cơ chế phức tạp hơn so với Apriori, việc xây dựng và duy trì cây FP yêu cầu người dùng có hiểu biết sâu hơn về thuật toán.
       - Khó mở rộng và tùy chỉnh: Do sử dụng cấu trúc cây FP và cách tiếp cận phức tạp, việc tùy chỉnh thuật toán FP-Growth hoặc mở rộng nó cho các bài toán khác có thể khó hơn so với Apriori.
       - Hiệu suất không nhất quán với dữ liệu thưa: Khi dữ liệu thưa (sparse), hiệu suất của FP-Growth có thể giảm, do cần xây dựng và quản lý cấu trúc cây FP không tối ưu trong trường hợp này.
  2. **Tổng kể so sánh**
* Apriori: Phù hợp cho các bài toán nhỏ hoặc cần tính toán chính xác, nhưng có thể trở nên không hiệu quả khi áp dụng trên tập dữ liệu lớn do quét nhiều lần.
* FP-Growth: Thích hợp hơn cho các bài toán với dữ liệu lớn nhờ hiệu suất cao và sử dụng bộ nhớ tối ưu, nhưng yêu cầu sự hiểu biết sâu hơn về thuật toán và có thể kém hiệu quả với dữ liệu thưa.

1. **Hình ảnh trực quan**

**Dataset 1: Chess**

**A graph of blue and orange bars

Description automatically generated**

**A screenshot of a computer program

Description automatically generated**

**Dataset 2: Mushroom**

**A graph with blue and orange bars

Description automatically generated**

**A screenshot of a computer

Description automatically generated**