Độ đo hỗn loạn sau khi sử dụng biến ***A*** phân hoạch dữ liệu ***D*** có ***m*** phần tử thành ***v*** phân vùng kích thước tương ứng là m_1, m_2, ..., m_v được tính bởi (5):

$$Info_A(D) = -\sum_{j=1}^{v} \frac{m_j}{m} Info(D_j) \quad (5)$$

Độ lợi thông tin khi chọn biến ***A*** phân hoạch dữ liệu ***D*** thành ***v*** phần được tính bởi công thức (6):

$$Gain(A) = Info(D) - Info_A(D) \quad (6)$$

Giải thuật CART của Breiman và các cộng sự sử dụng chỉ số Gini để phân hoạch dữ liệu trong quá trình xây dựng cây. Giả sử p_c là xác suất mà phần tử trong dữ liệu ***D*** thuộc lớp y_c (c =1, C), chỉ số Gini được tính theo công thức (7):

$$Gini\ (D) = 1 - \sum_{c=1}^{k} p_c^2 \quad (7)$$

Hàm Gini nhỏ nhất khi lớp trong ***D*** bị lệch. Nếu sử dụng biến ***A*** phân hoạch ***D*** kích thước ***m*** thành 2 tập con D_1 (kích thước m_1) và D_2 (kích thước m_2), hàm Gini được tính như công thức (8). Biến được chọn phân hoạch dữ liệu là biến cho giá trị chỉ số Gini nhỏ nhất.

$$Gini_A(D) = \frac{m_1}{m} Gini(D_1) + \frac{m_2}{m} Gini(D_2) \quad (8)$$

Cho vấn đề hồi quy, độ đo hỗn loạn thông tin tại phân vùng ***D*** dựa trên độ lệch chuẩn như trong (9) với μ là giá trị trung bình của các giá trị ***y*** trong ***D***.

$$S(D) = \sum_{i=1}^{k} \frac{(y_i - \mu)^2}{k} \quad (9)$$

Nếu sử dụng biến ***A*** phân hoạch ***D*** kích thước ***m*** thành 2 tập con D_1 (kích thước m_1) và D_2 (kích thước m_2), độ hỗn loạn sau khi phân hoạch được tính như công thức (10).

$$S_A(D) = \frac{m_1}{m} S(D_1) + \frac{m_2}{m} S(D_2) \quad (10)$$

Biến được chọn phân hoạch dữ liệu là biến cho giá trị độ hỗn loạn trước khi phân hoạch trừ cho độ hỗn loạn sau khi phân hoạch là nhỏ nhất.

Mô hình cây quyết định sau khi xây dựng thường không mạnh với nhiễu và dễ dẫn đến học vẹt. Tức là mô hình có tính tổng quát thấp, chỉ cần dữ liệu kiểm tra có thay đổi một ít so với dữ liệu học thì cây quyết định dự báo sai. Để khắc phục khuyết điểm này, Quinlan cũng đề nghị các chiến lược cắt nhánh trong giải thuật C4.5. Có 2 lựa chọn hoặc postpruning (cắt nhánh cây sau khi xây dựng cây) hay prepruning (dừng sớm quá trình phân nhánh). Trong thực tế, postpruning được sử dụng nhiều hơn prepruning. Tuy nhiên độ phức tạp của việc cắt nhánh sau khi xây dựng cây rất phức tạp, sử dụng các chiến lược để ước lượng lỗi sinh ra bởi mô hình sau khi cắt nhánh.

3.4 Mô hình Bagging (BagDT)

Từ những năm 1990, cộng đồng máy học đã nghiên cứu cách để kết hợp nhiều mô hình phân loại yếu thành mô hình tập hợp phân loại mạnh cải thiện độ chính xác cao hơn so với chỉ một mô hình phân loại đơn yếu. Trong phân tích thành phần lỗi của giải thuật học, Breiman đã chỉ ra trong (Breiman, 1996), lỗi bao gồm 2 thành phần là bias và variance. Thành phần lỗi bias là khái niệm về lỗi của mô hình học (không liên quan đến dữ liệu học) và thành phần lỗi variance là lỗi do tính biến thiên của mô hình so với tính ngẫu nhiên của các mẫu dữ liệu học. Mục đích của các mô hình tập hợp là làm giảm variance và/hoặc bias của các giải thuật học. Dựa trên cách phân tích hiệu quả của giải thuật học dựa trên thành phần lỗi bias và variance, Breiman đã đề xuất giải thuật học Bagging (Bootstrap AGGregatING) nhằm giảm lỗi variance của giải thuật học nhưng không làm tăng lỗi bias quá nhiều. Giải thuật có thể được tóm tắt như sau:

– Từ tập dữ liệu học ***LS*** có ***m*** phần tử, xây dựng ***T*** mô hình cơ sở độc lập nhau.

– Mô hình thứ ***t*** được xây dựng trên tập mẫu Bootstrap thứ ***t*** (lấy mẫu ***m*** phần tử có hoàn lại từ tập học ***LS***).

– Kết thúc quá trình xây dựng ***T*** mô hình cơ sở, dùng chiến lược bình chọn số đông để phân lớp một phần tử ***x*** mới đến hoặc giá trị trung bình cho bài toán hồi quy.

Trong thực tế, giải thuật Bagging cải thiện rất tốt các mô hình đơn không ổn định như cây quyết định và thường có thành phần lỗi variance cao. Hình 4 là ví dụ của giải thuật Bagging được áp dụng cho mô hình cơ sở là cây quyết định.