ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Lab 03 – Classification & Clustering

Data Mining - Term I/2020-2021

Thành phố Hồ Chí Minh, ngày 7 tháng 12 năm 2020

1. ***Mức độ hoàn thành:*** 100%
2. ***Mô tả tập dữ liệu:***

File "hawks.csv" mô tả về các mẫu diều hâu từ ba loài khác nhau. Dữ liệu gồm 908 quan sát với 19 thuộc tính trong bảng sau:

STT	Thuộc tính	Mô tả
1	Month	Tháng (ghi bằng số)
2	Day	Ngày trong tháng
3	Year	Năm (từ 1992-2003)
4	CaptureTime	Thời gian bắt
5	ReleaseTime	Thời gian thả
6	BandNumber	Dải ID của từng con diều hâu
7	Species	Loài (CH = Cooper's = diều hâu Cooper, RT = Red-tailed = diều hâu đuôi đỏ, SS = Sharp-shinned = diều hâu vuốt sắc)
8	Age	A = Adult (trưởng thành), I = Imature (chưa trưởng thành)
9	Sex	F = Female (cái), M = Male (đực)
10	Wing	Chiều dài cánh (mm)
11	Weight	Cân nặng (gm)
12	Culmen	Chiều dài đường sống mỏ (mm)
13	Hallux	Chiều dài ngón chân (mm)
14	Tail	Chiều dài đuôi (mm)
15	StandardTail	Chiều dài đuôi tiêu chuẩn (mm)
16	Tarsus	Chiều dài xương bàn chân (mm)
17	WingPitFat	Lượng mỡ trong cánh
18	KeelFat	Lượng mỡ phần xương ức

19	Crop	Lượng thức ăn trong diều (1=đầy, 0=rỗng)

3. *Tiền xử lí dữ liệu:*

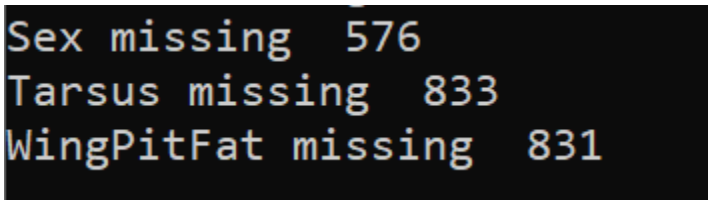
Ta cần tiền xử lí dữ liệu trước khi thực hiện các yêu cầu bên dưới.

Nếu thuộc tính phân lớp là *Species*, sẽ có một vài thuộc tính thừa không ảnh hưởng tới việc phân lớp, đó là *Month, Day, Year, CaptureTime, ReleaseTime, BandNumber*. Ta xoá bỏ các thuộc tính này đi.

Tiếp theo, ta xoá những thuộc tính bị thiếu dữ liệu trên 60%, sử dụng lại code python ở Lab01 (xem trong file Preprocessing) để thực hiện:

```
python3 delete_missing_col.py hawks.csv --missing_rate=60 --output=hawks.csv
```

Ta sẽ có thêm ba thuộc tính bị loại là *Sex, Tarsus, WingPitFat* với độ thiếu dữ liệu trên 60%



Cuối cùng, ta điền dữ liệu còn thiếu vào các ô trống bằng cách tính trung bình:

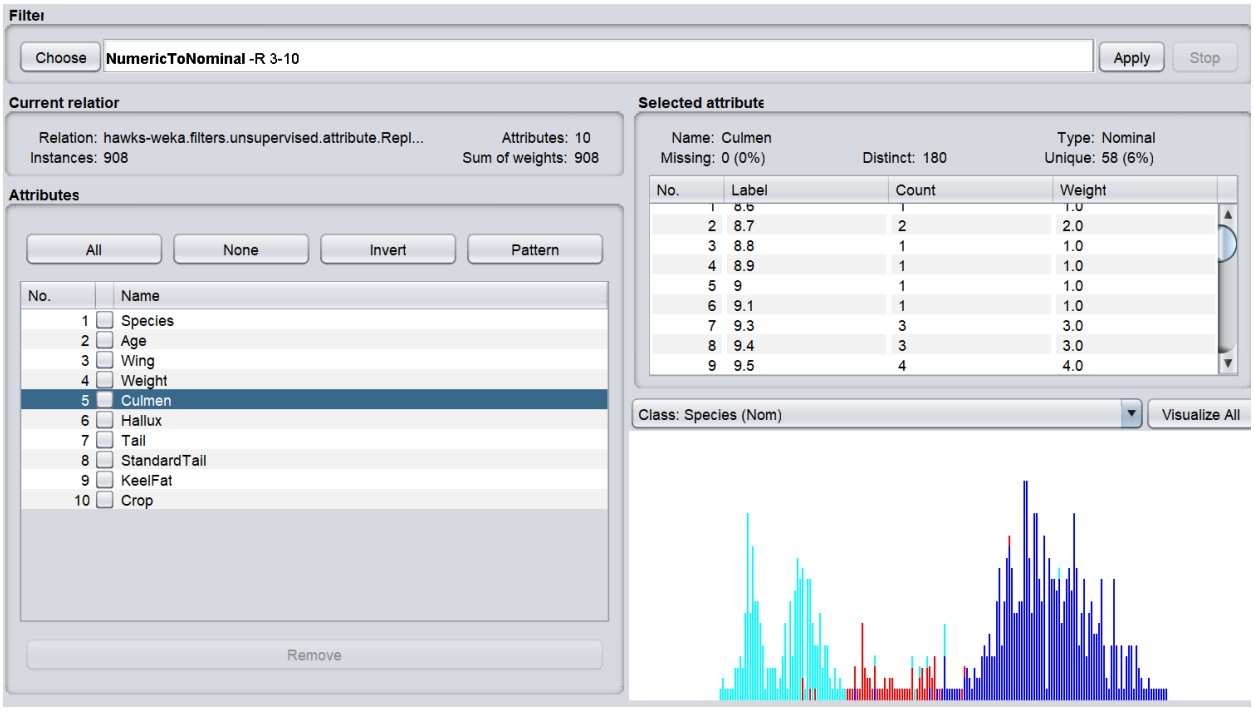
```
python3 impute.py hawks.csv --method=Means --columns Weight Culmen Hallux
StandardTail KeelFat Crop --output=hawks_output.csv
```

Như vậy ta đã hoàn thành xong việc tiền xử lí dữ liệu.

4. *Phân lớp dữ liệu bằng Weka Explorer:*

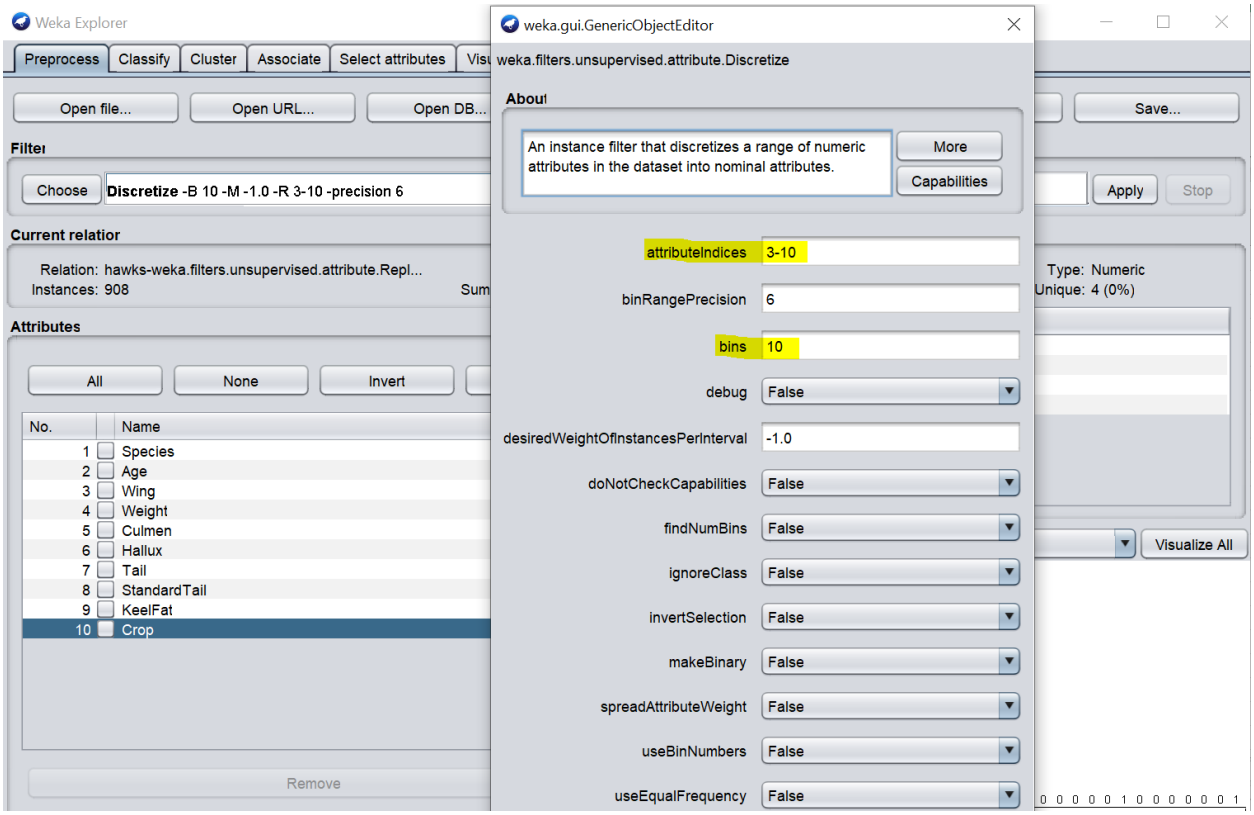
a) *Phân lớp dữ liệu:*

Ta cho dữ liệu vào Weka và dùng filter *NumericToNominal* cho các thuộc tính từ 3-10 để chuyển tất cả dữ liệu về dạng nominal. Sau đó tiến hành phân lớp.



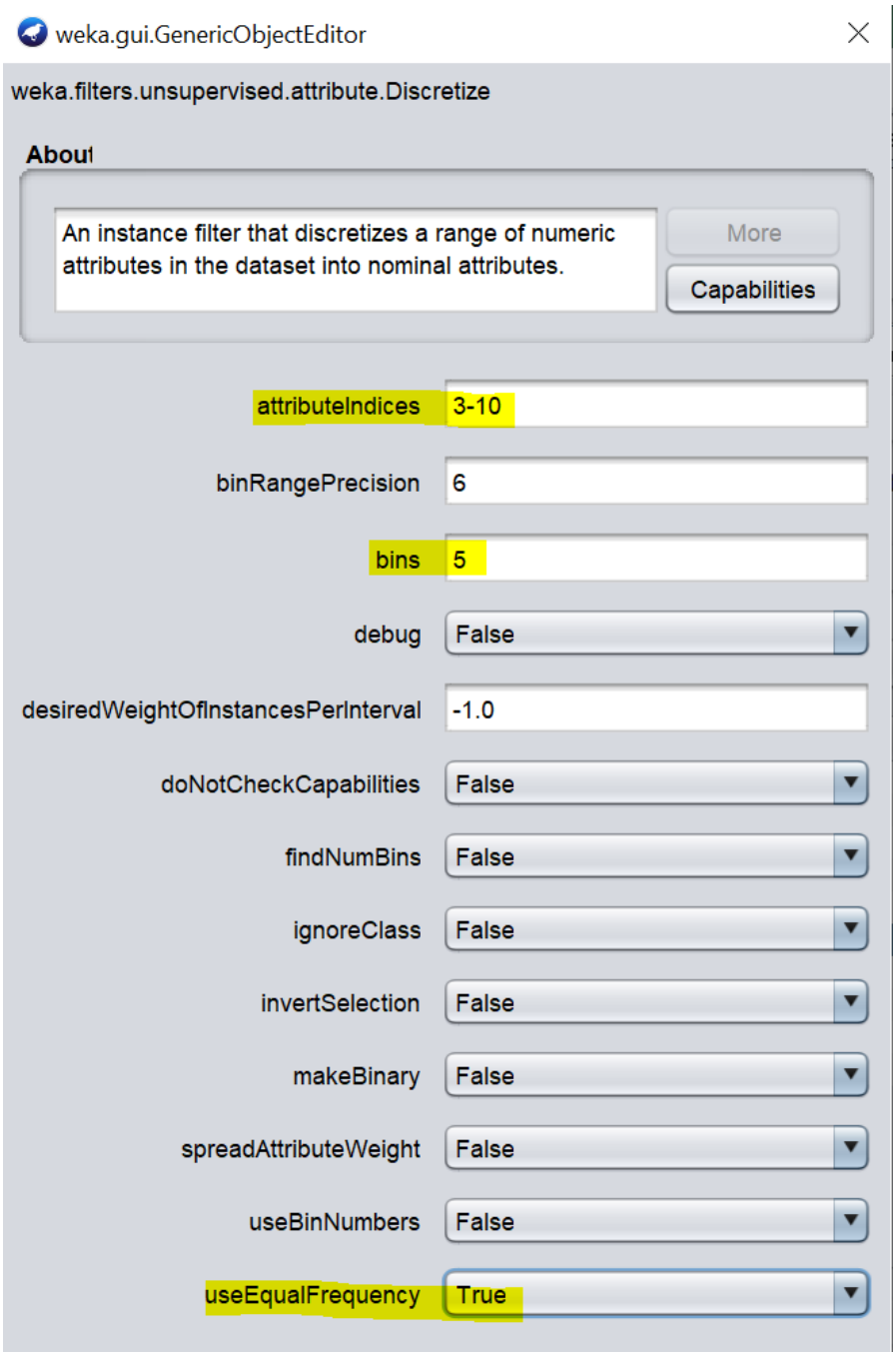
b) <u>Rời rạc hoá thành 10 giỏ dữ liệu:</u>

Tiến hành rời rạc hoá theo độ rộng bằng cách sử dụng filter Discretize cho thuộc tính từ 3-10, số giỏ là 10:



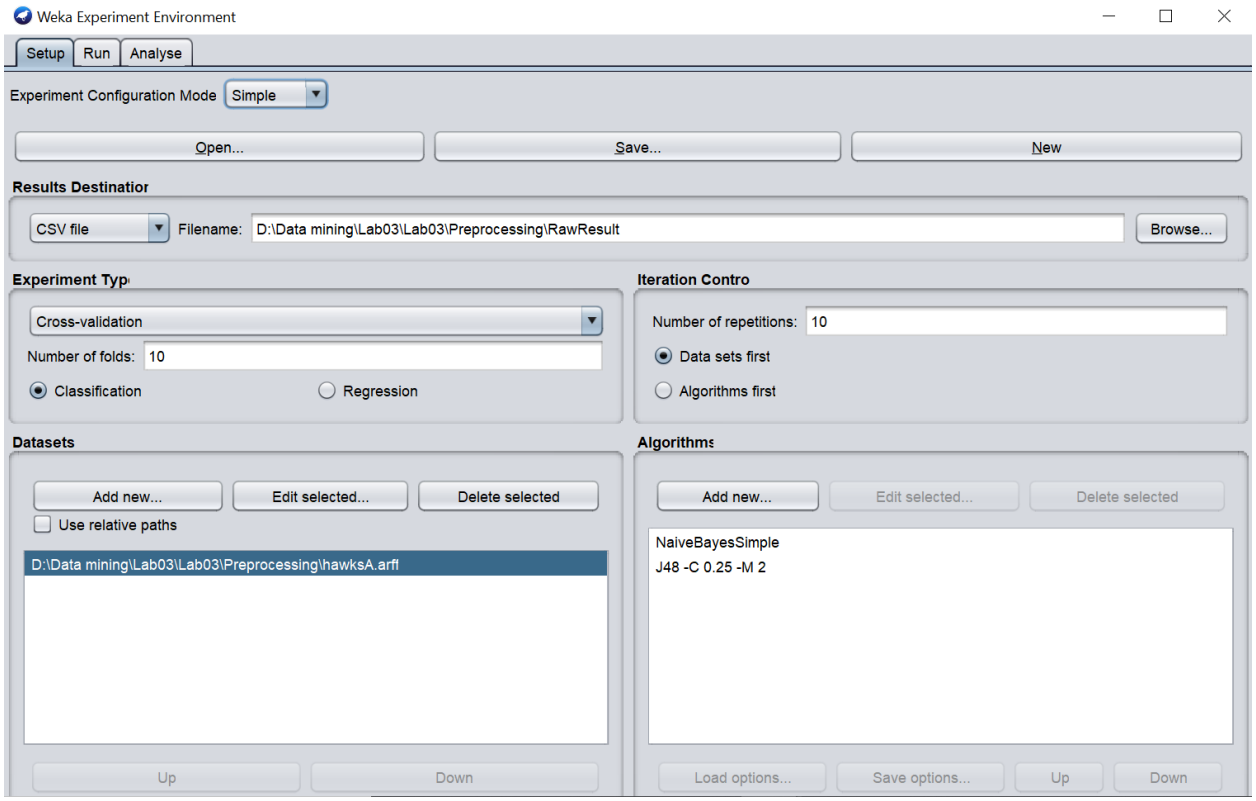
c) *<u>Rời rạc hoá thành 5 giỏ dữ liệu:</u>*

Tiến hành rời rạc hoá theo sâu bằng cách sử dụng filter *Discretize* cho thuộc tính từ 3-10, số giỏ là 5, chọn thuộc tính *useEqualFrequency = True*:

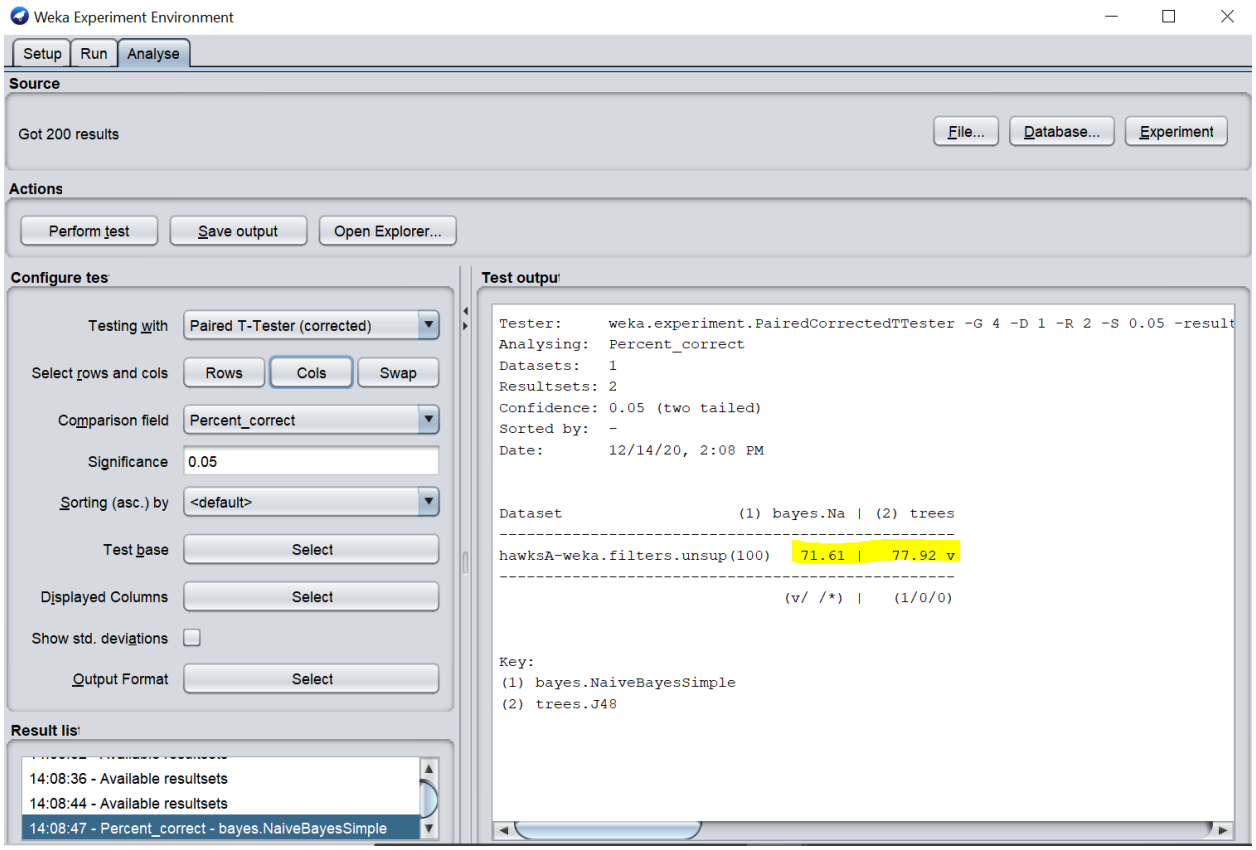


5. ***Phân lớp dữ liệu bằng Weka Experimenter:***

   Mở tab *Experimenter* của Weka, ta nhập dataset cần phân lớp và thuật toán vào, cho chạy 10 lần, sau đó lưu kết quả vào file *RawResults.csv*:



Qua tab *Analyse*, chọn dataset *RawResults.csv*. Tại *Row* ta chọn *Dataset*, tại *Cols* ta chọn *Scheme*. Bấm *Perform test* để chạy, kết quả như hình dưới:



Hai số 71.61 và 77.92 chính là tỉ lệ mẫu trung bình được phân lớp đúng của hai thuật toán NaiveBayesSimple và J48

6. *Đánh giá:*

- *Phương pháp phân lớp nào thường cho kết quả cao nhất?*

  Tính trung bình tỉ lệ mẫu được phân lớp chính xác của từng phương pháp, ta có kết quả sau:

NaiveBayesSimple	97.62188
ID3	94.44968
J48	96.55709

Phương pháp NaiveBayesSimple thường cho kết quả cao nhất.

- *Phương pháp nào không thực hiện tốt và tại sao?*

Phương pháp ID3 không thực hiện tốt, vì phương pháp này sử dụng *Information Gain* làm độ đo, nó ưu tiên những thuộc tính có số lượng lớn các giá trị mà ít xét tới những giá trị nhỏ hơn. Và tại một thời điểm nó chỉ xét một thuộc tính để đưa ra quyết định.

- *Tại sao ta sử dụng phiên bản đã rời rạc hóa của tập dữ liệu nếu tập dữ liệu đã được rời rạc hóa?*

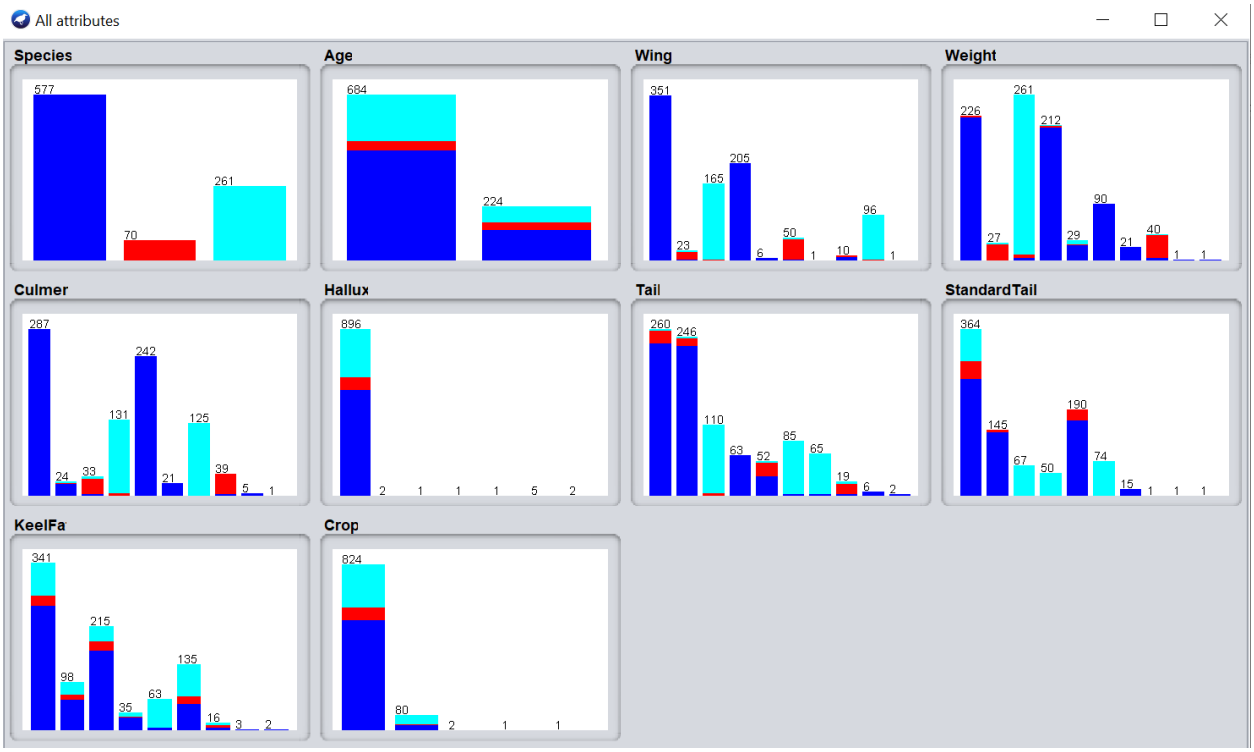
  Vì yêu cầu của các thuật toán là chỉ chạy được trên tập dữ liệu rời rạc, nên ta cần phải rời rạc hoá để có thể phân lớp đúng.

- *Việc rời rạc hóa và cách rời rạc hóa có ảnh hưởng đến kết quả phân lớp hay không, nếu có thì ảnh hưởng thể nào?*

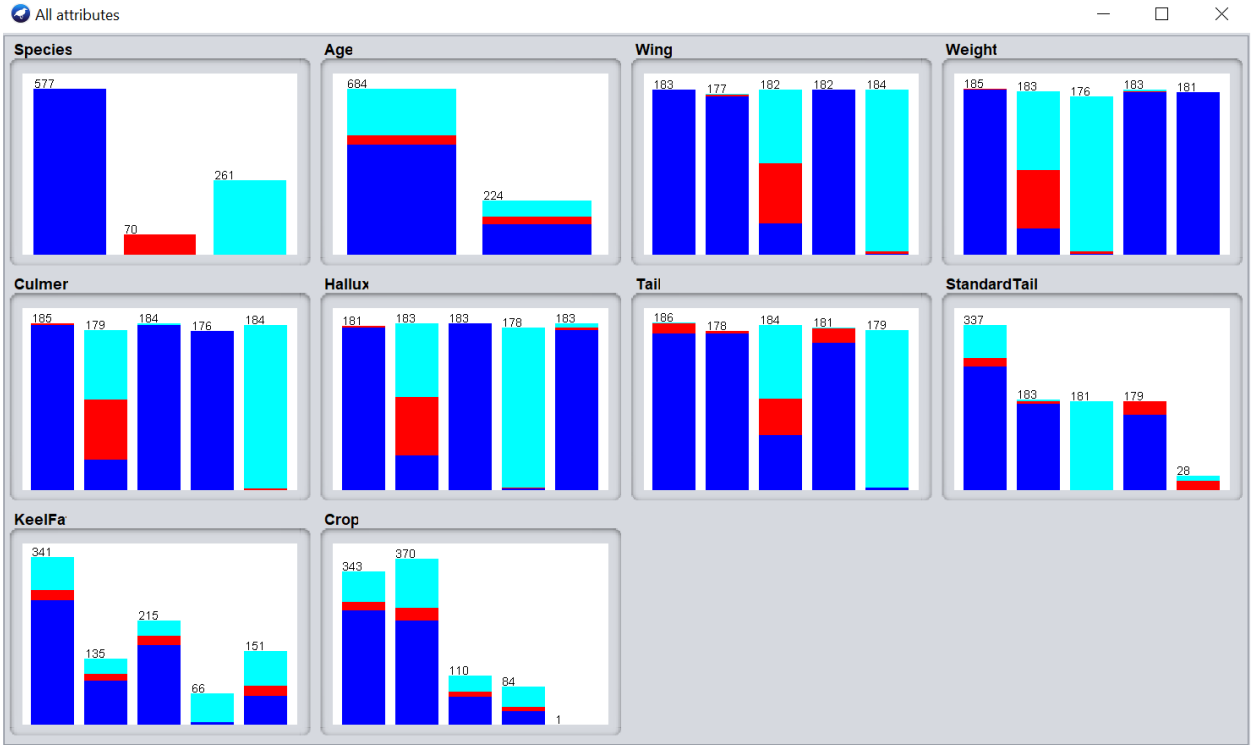
  Có ảnh hưởng, cụ thể là tỉ lệ mẫu được phân lớp đúng của các cách rời rạc hoá sẽ khác nhau (trong file Results.csv thể hiện rõ điều đó). Đây là kết quả trung bình cho ba cách rời rạc hoá:

A	93.49124
B	98.10834
C	96.89058

Ta thấy cách rời rạc B (chia 10 giỏ theo độ rộng) cho kết quả tốt nhất. Cách rời rạc A (rời rạc từng giá trị) cho ra quá nhiều nhánh dẫn đến sự hỗn loạn. Cách rời rạc C (chia 5 giỏ theo chiều sâu) chỉ phân chia đều các giá trị vào mỗi giỏ mà không quan tâm đến thuộc tính quyết định (hay độ thuần khiết của của thuộc tính không cao). Ta xem xét biểu đồ của cách rời rạc B và C sẽ thấy rõ điều đó, các cột ở phương pháp B có màu đồng nhất hơn ở phương pháp C:



Phương pháp B: chia 10 giỏ theo độ rộng



Phương pháp C: chia 5 giỏ theo chiều sâu

- *Chiến lược nào trong ba chiến lược đánh giá đã đánh giá quá cao (overestimate) độ chính xác và tại sao?*

  Chiến lược *use training-set* đã đánh giá quá cao độ chính xác, vì nó lấy ngay tập huấn luyện để làm tập kiểm tra, vậy nên lúc nào độ chính xác cũng cao và mang tính chủ quan.
- *Chiến lược nào đánh giá thấp (underestimate) độ chính xác và tại sao?*

  Chiến lược Cross-validation đánh giá thấp độ chính xác vì nó không tin tưởng vào kết quả của một lần chia tập huấn luyện và tập kiểm tra; thay vào đó nó phải đánh giá nhiều lần (trong bài là 10) với các tập huấn luyện và kiểm tra khác nhau mới cho ra kết quả cuối cùng.