ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO LAB 03:

CLASSIFICATION & CLUSTERING

Sinh viên thực hiên:

Trần Thanh Tùng - 18120258.

Trần Hữu Chí Bảo - 18120288.

Môn: Khai thác dữ liệu và ứng dụng

Thành phố Hồ Chí Minh - 2020

MỤC LỤC

MỤC LỤC 2
PHẦN I. TIỀN XỬ LÝ 3
1.1 Cột CaptureTime: 3
1.2 Cột ReleaseTime: 3
1.3 Cột BandNumber: 4
1.4 Cột Sex: 5
1.5 Các cột còn lại: 6
PHẦN II. PHÂN LỚP DỮ LIỆU BẰNG WEKA EXPLORER 7
1.1 Thực nghiệm A: 7
1.2 Thực nghiệm B: 7
1.3 Thực nghiệm C: 8
PHẦN III. PHÂN LỚP DỮ LIỆU BẰNG WEKA EXPERIMENT 10
PHẦN IV. ĐÁNH GIÁ 12
PHẦN V. TÀI LIỆU THAM KHỎA 14

PHẦN I. TIỀN XỬ LÝ

Nguyên nhân: Do dữ liệu thô ban đầu có nhiều thuộc tính có các giá trị missing.

1.1 Cột CaptureTime:

- Trong cột này, có 1 dòng dữ liệu bị bỏ trống.
- Ta đơn giản là tìm giá trị mean và điền vào giá trị bỏ trống đó.
- Đồng thời, ta đổi thời gian dạng Giờ: Phút sang Phút (phút thứ bao nhiêu trong ngày) do Weka không hiểu thuộc tính này là liên tục. Bước này để ta có thể rời rạc hóa dữ liệu của các yêu cầu B, C.

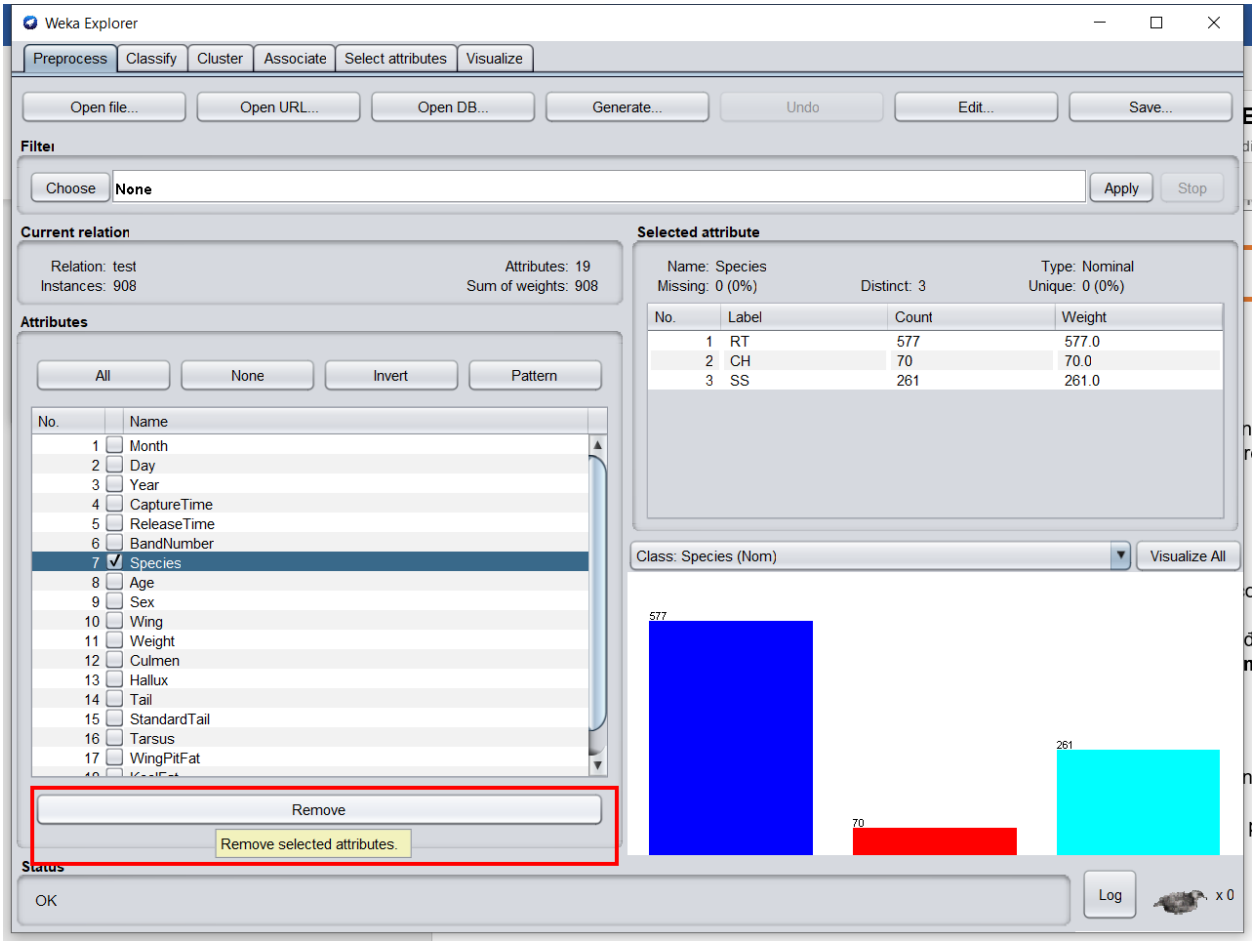
  Ta dùng thư viện pandas và numpy để xử lý việc này

```
import numpy as np
import pandas as pd
raw_df = pd.read_csv('hawks.csv') #doc file hawks.csv
#rut trich so phut cot CaptureTime
minute = pd.to_numeric(raw_df['CaptureTime'].str.extract(r'[:.](\d+)', expand = False),errors = 'coerce')
#rut trich so gio cot CaptureTime
hour = pd.to_numeric(raw_df['CaptureTime'].str.extract(r'(\d+)[:.]', expand = False),errors = 'coerce')
#chuyen cot CaptureTime ve phut
raw_df['CaptureTime'] = (minute + 60*hour).fillna((minute + 60*hour).mean()).astype(int)
```

Hình 1. Xử lý Cột CaptureTime bằng pandas

1.2 Cột ReleaseTime:

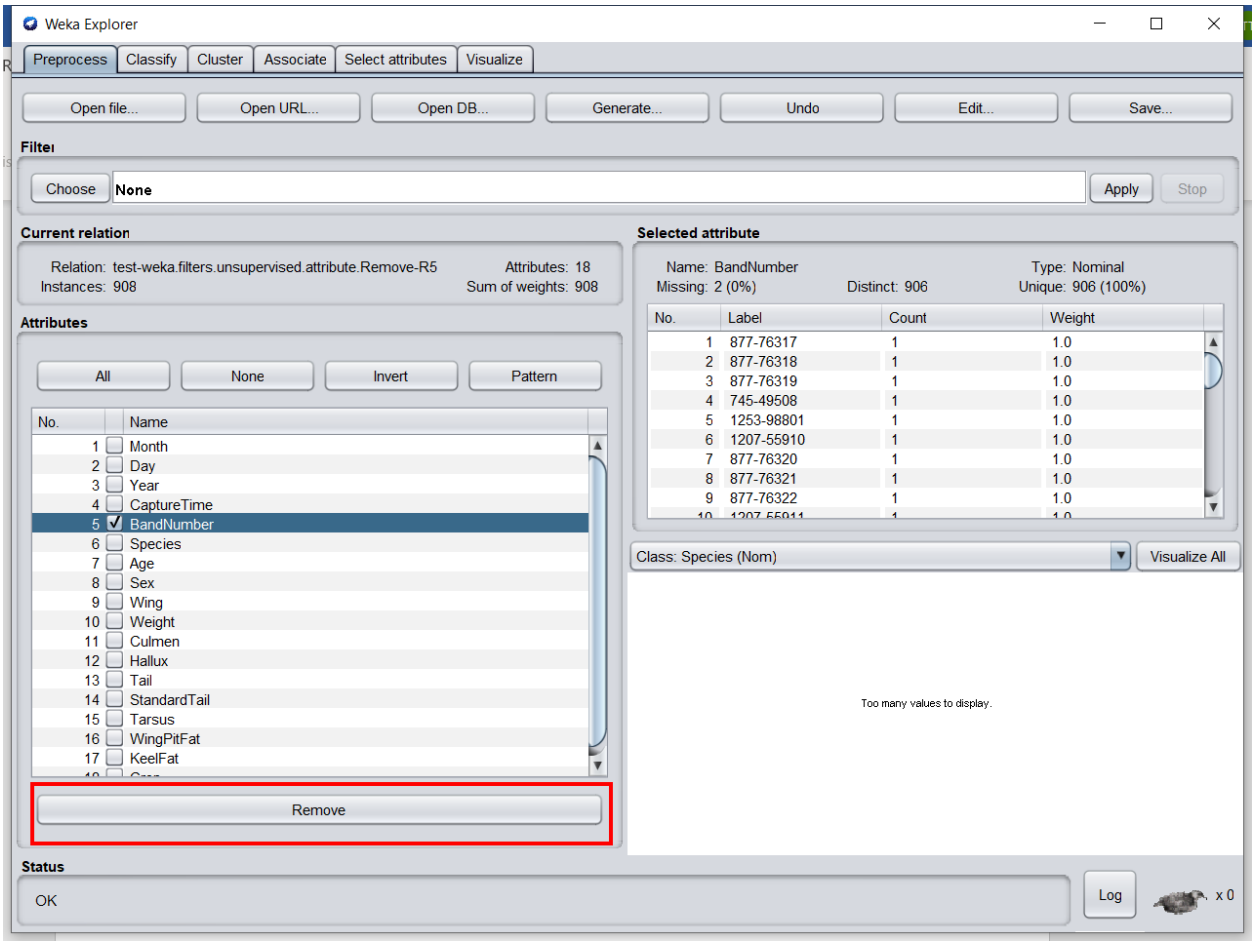
- Thông qua mô tả của tập dữ liệu, đây là cột ghi thời gian thả ra của những con chim ưng sau khi bắt được.
- Thuộc tính này có đến 93% missing, hơn nữa ta nhận thấy rằng sau khi bắt được người ta thường thả nó sau 10-30’ → Có thể suy ra từ thuộc tính **CaptureTime** nên ta có thể không xét đến thuộc tính này khi phân tích.
- Ta tiến hành Remove thuộc tính ReleaseTime bằng Weka:



Hình 2. Remove thuộc tính ReleaseTime bằng Weka

1.3 Cột BandNumber:

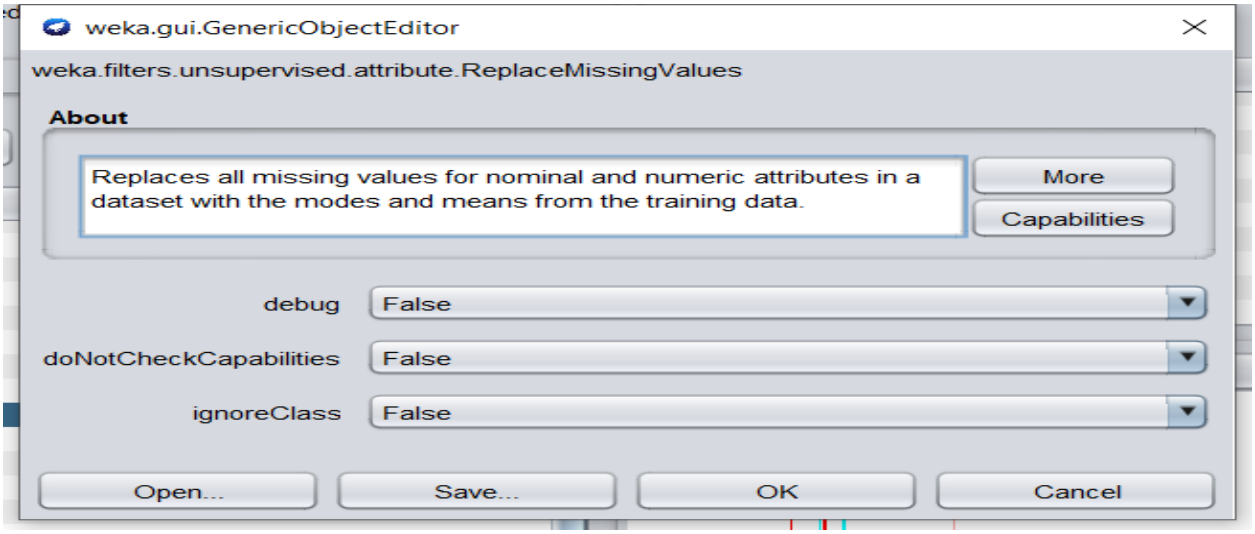
- Thông qua mô tả của tập dữ liệu, thì cột này chứa mã định danh của các con chim ưng được quan sát. Mỗi con mang một mã duy nhất.
- Do đó ta nhận thấy rằng có thể không xét đến thuộc tính này trong quá trình phân lớp.
- Ta tiến hành Remove thuộc tính BandNumber bằng Weka:



Hình 3. Remove thuộc tính BandNumber bằng Weka

1.4 Cột Sex:

- Trong cột này, có đến 63% dữ liệu bị bỏ trống.
- Ta sử dụng filter ReplaceMissingData của Weka để điền các giá trị trống cho thuộc tính:



Hình 4. Bảng thông số của thuộc tính Replaces all missing values

1.5 Các cột còn lại:

- Các cột **Wing, Weight, Culmen, Hallux, Tail, StandardTail, Tarsus, WingPitFat, KeelFat, Crop**
- Ta dùng chiến lược điền mean vào các dòng bị bỏ trống của các cột này nguyên nhân do các cột này là các thuộc tính liên tục.

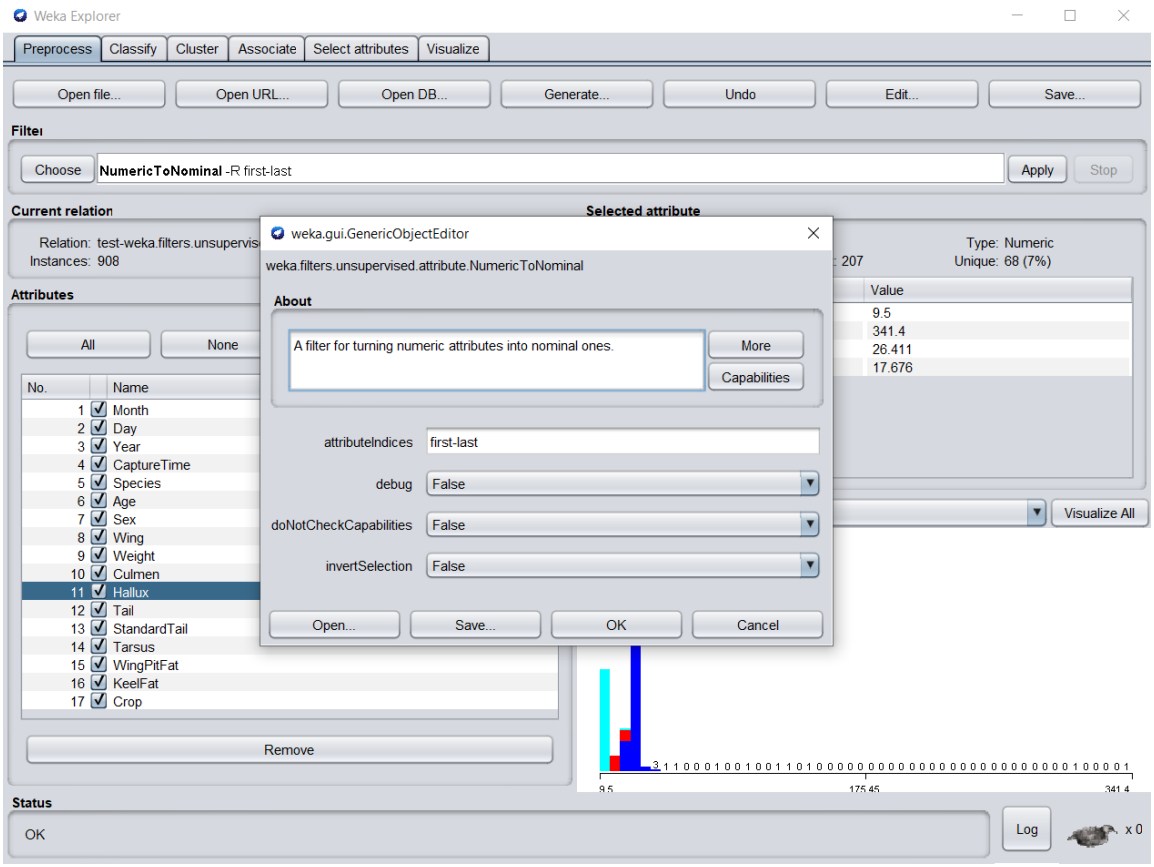
```
def fill_na_col(col):
    if  col.name in ['Wing', 'Weight', 'Culmen', 'Hallux', 'Tail', 'StandardTail', 'Tarsus','WingPitFat','KeelFat','Crop']:
        return col.fillna(round(col.mean(),2))
    return col
temp_df = raw_df.apply(fill_na_col)
raw_df = temp_df
raw_df.to_csv('test123.csv', index = False) #luu ket qua xuong file
```

Hình 5. Điền các dòng bị bỏ trống bằng giá trị mean

PHẦN II. PHÂN LỚP DỮ LIỆU BẰNG WEKA EXPLORER

1.1 Thực nghiệm A:

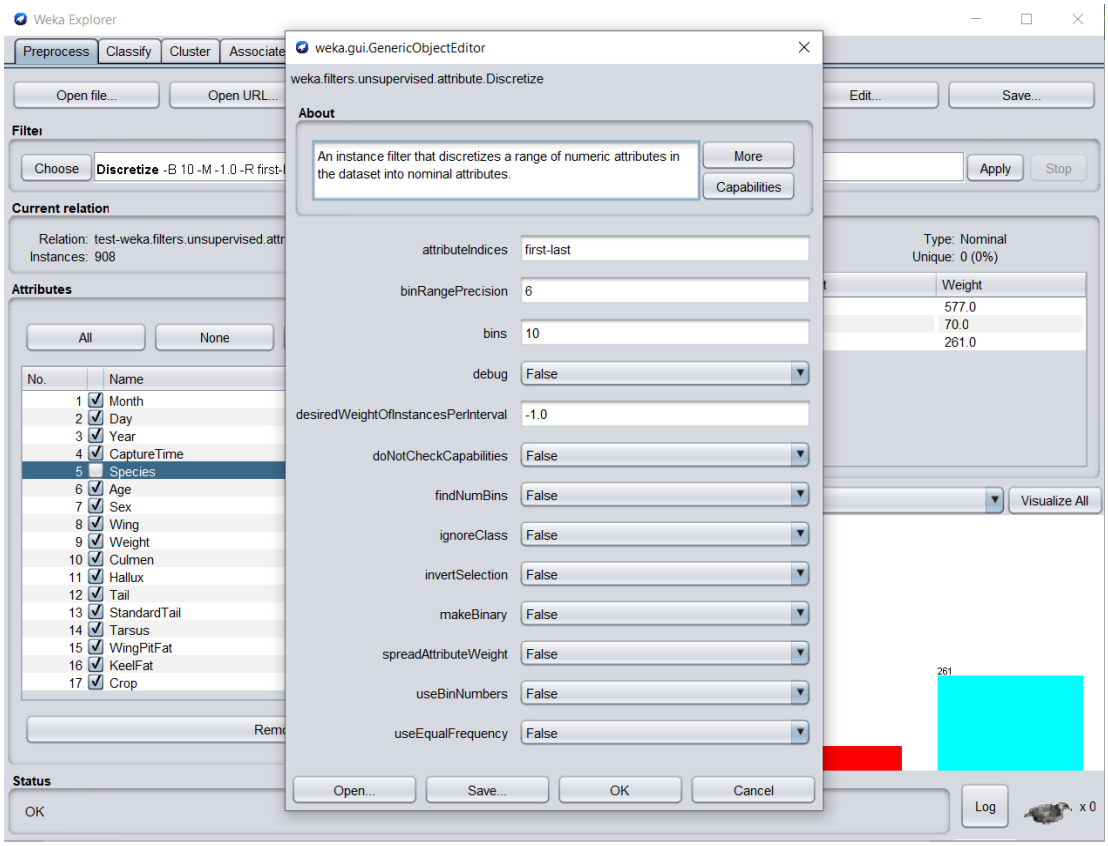
- Chuyển tất cả thuộc tính của dữ liệu đầu vào về nominal do yêu cầu của thuật toán.



Hình 6.Chuyển các thuộc tính thành nominal

1.2 Thực nghiệm B:

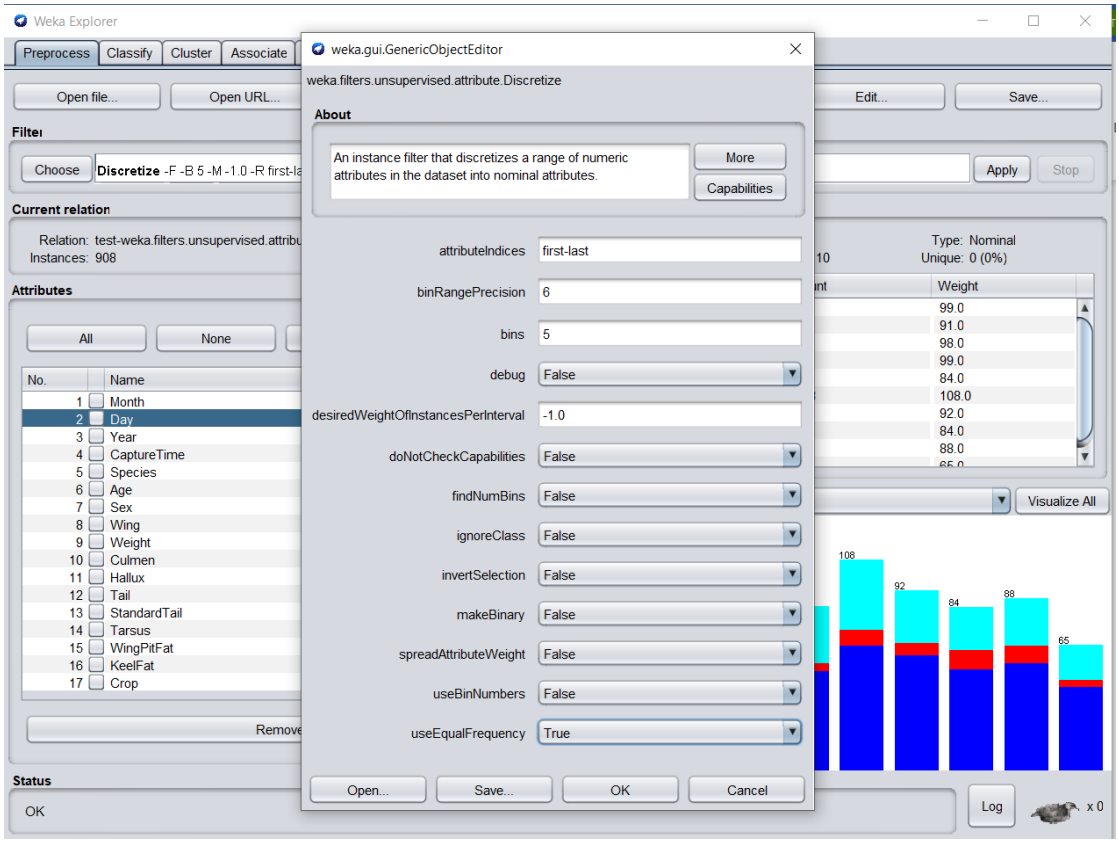
- Rời rạc hóa mọi thuộc tính không phải là lớp trong tập dữ liệu thành 10 giỏ có độ rộng bằng nhau.
- Sử dụng chức năng "Filter" trong cửa số "Preprocess" của Explorer, chọn 'filters' → 'unsupervised' → 'attribute' → 'Discretize'



Hình 7. Bảng thông số.

1.3 Thực nghiệm C:

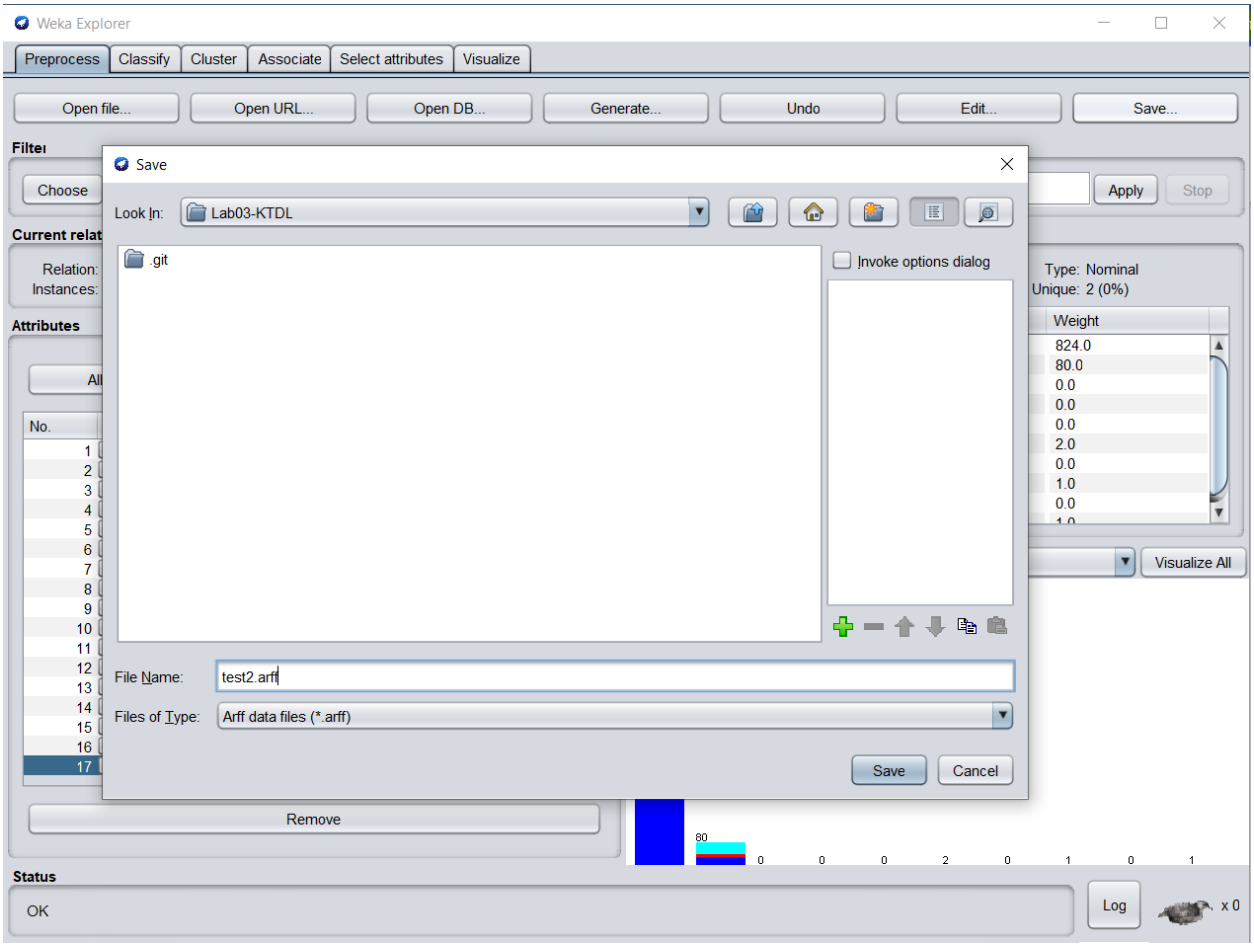
- Rời rạc hóa mọi thuộc tính không phải là lớp trong tập dữ liệu thành 5 giỏ có độ sâu bằng nhau:
- Sử dụng chức năng “Filter” trong cửa số “Preprocess” của Explorer, chọn ‘filters’ → ‘unsupervised’ → ‘attribute’ → ‘Discretize’
  - **Lưu ý:** Số bins chỉnh = 5 và useEqualFrequency = true.



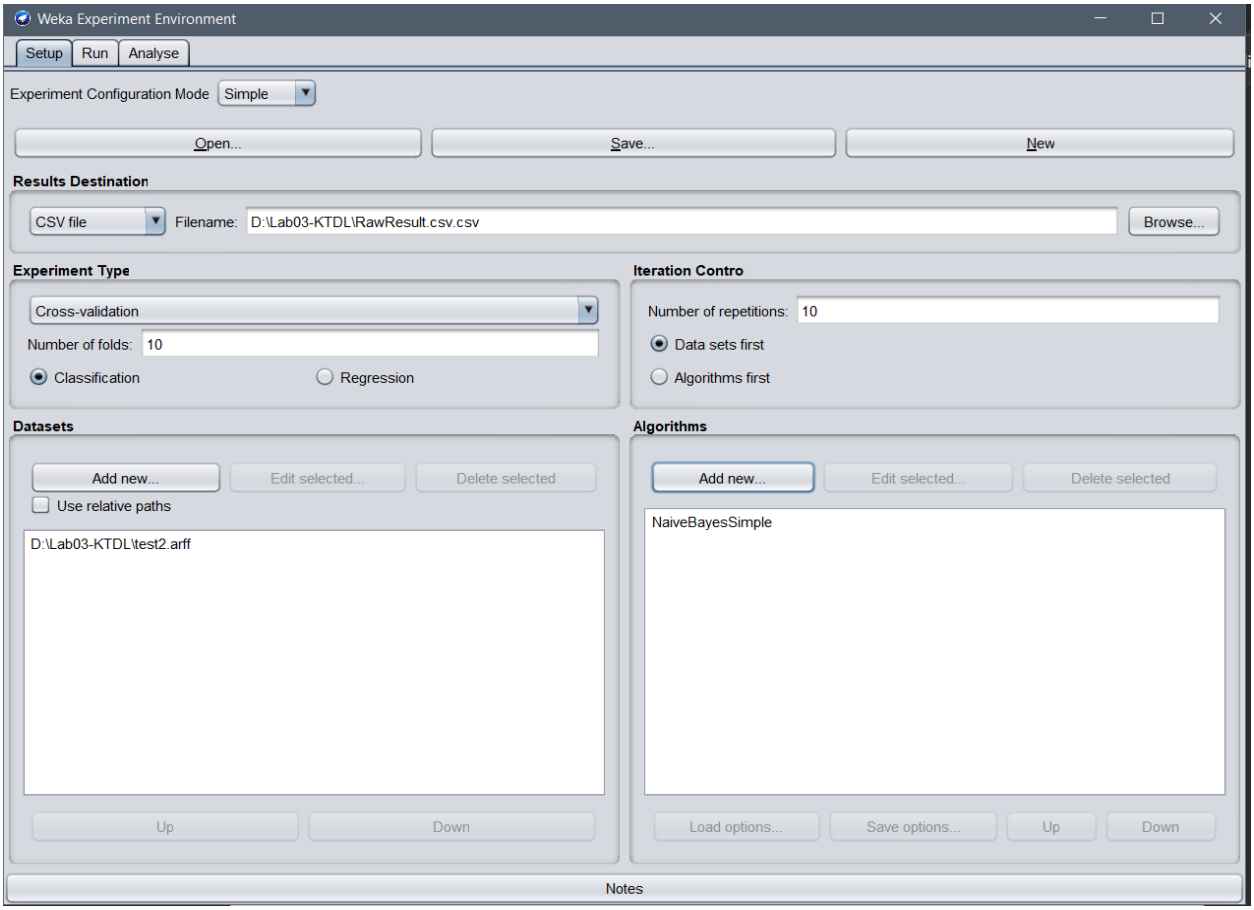
Hình 8. Bảng thông số.

PHẦN III. PHÂN LỚP DỮ LIỆU BẰNG WEKA EXPERIMENT

- Ở thực nghiệm này ta cần sử dụng file arff với các thuộc tính đều là nominal.
- Ta tạo ra file này bằng cách lưu lại file dữ liệu sau khi đã tiền xử lý ở thực nghiệm A dưới dạng file arff.

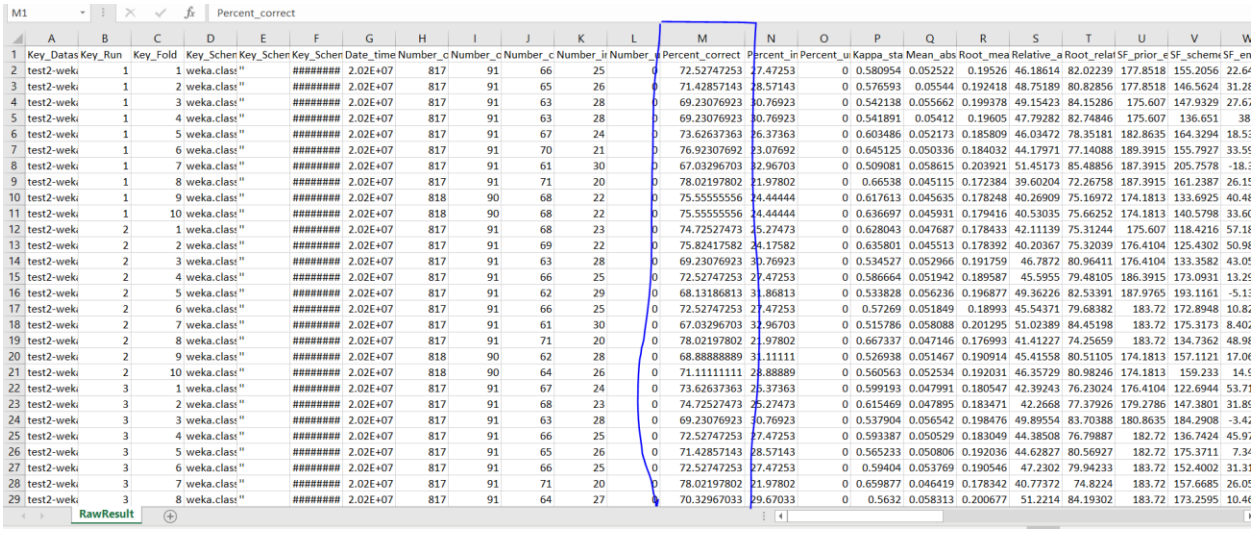


Hình 9. Lưu lại file đã tiền xử lý ở thực nghiệm A dưới dạng file arff



Hình 10. Giao diện trên weka Experiment

Sau khi chạy thực nghiệm xong, ta tính tỉ lệ trung bình của các mẫu được phân lớp đúng sau 10 × 10 lượt chạy bằng cách tính mean của cột Percent_correct trong tập tin ResultRaw.csv



HÌnh 11. Cột Percent_correct trong file ResultRaw.csv

PHẦN IV. ĐÁNH GIÁ

Từ kết quả thu được qua các thực nghiệm đánh giá, ta rút ra được một số kết luận:

- **Phương pháp phân lớp nào thường cho kết quả cao nhất?**
  - Phương pháp phân lớp NaiveBayesSimple thường cho kết quả cao nhất.
- **Phương pháp nào không thực hiện tốt và tại sao?**
  - 2 phương pháp ID3 và J48 khi sử dụng 2 chiến lược đánh giá "cross validation với 10 fold" và "Percentage split với tỷ lệ 66%" khi thực nghiệm với tập dữ liệu không được chia thành các bin thì cho ra kết quả không tốt, nguyên nhân do 2 thuật toán này là thuật toán phân lớp bằng cách tạo ra cây quyết định. Mà trong tập dữ liệu của ta, có khá nhiều thuộc tính missing (được điền bằng giá trị mean) vì thế cây quyết định tạo ra không mang tính hiệu quả cao.
  - Trong thực nghiệm D bằng weka Experimenter ta nhận thấy cả 2 phương pháp NaiveBayesSimple và J48 đều không mang kết quả khả quan khi mức độ phân lớp đúng của cả 2 chỉ hơn 70%. NaïveBayesSimple còn cho kết quả tệ hơn J48 khi kết quả phân lớp đúng trung bình chỉ là 72%, nguyên nhân vì NaiveBayesSimple dựa trên giả định rằng các thuộc tính độc lập với nhau, điều này là phi thực tế trong tập dữ liệu của ta.
- **Tại sao ta sử dụng phiên bản đã rời rạc hóa của tập dữ liệu nếu tập dữ liệu đã được rời rạc hóa?**
  - Ta sử dụng phiên bản đã rời rạc hóa của tập dữ liệu nếu tập dữ liệu đã được rời rạc hóa vì một số thuộc tính rời rạc mang quá nhiều giá, điều này đôi khi khiến cho các thuật toán phân lớp hoạt động kém hiệu quả. Để giải quyết vấn đề này, ta sử dụng phiên bản đã được rời rạc hóa của chúng.

- **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?**
  - Việc rời rạc hóa và cách rời rạc hóa có ảnh hưởng tới việc phân lớp dữ liệu. Đối với 2 thuật toán ID3 và J48, thì việc phân loại được thực hiện tốt hơn trong trường hợp các thuộc tính được rời rạc hóa. So sánh 2 cách rời rạc hóa dữ liệu là rời rạc theo chiều rộng giỏ và rời rạc theo chiều sâu của giỏ ta cũng thấy có sự khác biệt trong kết quả phân loại.

- **Chiến lược nào trong ba chiến lược đánh giá đã đánh giá quá cao (overestimate) độ chính xác và tại sao?**
  - Trong 3 chiến lược đánh giá, thì "Use training set" đã đánh giá quá cao (overestimate) độ chính xác. Nguyên nhân vì cách hoạt động của Use training set:

1. Weka lấy tất cả dữ liệu đã được gán nhãn
2. Dùng thuật toán được chọn để xây dựng mô hình phân lớp từ các dữ liệu này.
3. Sau đó, dùng mô hình này để phân loại lại các dữ liệu ban đầu.
4. Cho ra kết quả phân loại.

➢ Chính vì cách hoạt động như vậy mà chiến lược đánh giá này đánh giá quá cao độ chính xác.

- **Chiến lược nào đánh giá thấp (underestimate) độ chính xác và tại sao?**

- Chiến lược "Percentage Split với tỷ lệ 66% đánh giá thấp độ chính xác (underestimate) vì chiến lược này sẽ chia tập dữ liệu của chúng ta thành 66% training data và 34% testing data, và cách chia là ngẫu nhiên. Nên trong nhiều trường hợp khiến cho độ chính xác thấp hơn so với thực tế.

PHẦN V. TÀI LIỆU THAM KHẢO

[1] https://stackoverflow.com/questions/10437677/cross-validation-in-weka
[2] https://www.periyaruniversity.ac.in/ijcii/issue/marnew/2_mar_18.pdf
[3] https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputation-with-examples-6022d9ca0779
[4] https://www.w3schools.com/python/python_regex.asp
[5] https://stats.stackexchange.com/questions/419803/cross-validation-or-percentage-split
[6] https://towardsdatascience.com/5-reasons-why-you-should-use-cross-validation-in-your-data-science-project-8163311a1e79#:~:text=The%20classic%20approach%20is%20to,can%20create%20these%20folds%20with.
[7] https://www.cs.waikato.ac.nz/ml/weka/mooc/dataminingwithweka/transcripts/Transcript2-2.txt#:~:text=If%20we%20had%20just%20one,random%20split%20of%20the%20dataset