ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

LAB 3 - Classification & Clustering

MÔN: KHAI THÁC DỮ LIỆU VÀ ỨNG DỤNG

| Giáo viên hướng dẫn |

Thầy: Dương Nguyễn Thái Bảo

Sinh viên thực hiện:

CAO TẤT CƯỜNG – 18120296

HÀ VĂN DUY – 18120339

Chuyên ngành: Khoa học máy tính

Thành phố Hồ Chí Minh – 2020

MỤC LỤC

Phần I: Thông tin chung

Danh sách thành viên:

- Cao Tất Cường 18120296
- Hà Văn Duy 18120339

Tên file: Observations.pdf

Nội dung: Mô tả câu trả lời của nhóm về các quan sát, đánh giá mô hình, quá trình phân lớp dữ liệu bằng công cụ Weka.

Phần II: Tiền xử lý dữ liệu

Nhóm thực hiện tiền xử lý bộ dữ liệu hawks bằng cả 2 cách: Tiền xử lý bằng python và tiền xử lý trực tiếp trên Weka.

Cả 2 phương pháp tiền xử lý bổ trợ cho nhau để thực hiện được các yêu cầu, đều cùng điền dữ liệu bị thiếu bằng giá trị mean của mỗi thuộc tính.

2.1. Tiền xử lý bằng Python

Các giá trị categorical mặc dù có thể điền khuyết bằng giá trị "mode", nhưng vì nó thiếu cũng không ảnh hưởng tới kết quả (ví dụ như thuộc tính Release Time, Sex, ...), nên ở đây chỉ điền khuyết các thuộc tính là kiểu numeric để phục vụ cho quá trình rời rạc hóa dữ liệu thôi.

```
In [1]: import pandas as pd
        df = pd.read_csv('hawks.csv')
```

```
In [2]: df.describe().loc[['mean']]
```

Out[2]:

	Month	Day	Year	Wing	Weight	Culmen	Hallux	Tail	StandardTail	Tarsus	WingPitFat	KeelFat	Crop
mean	9.842511	15.740088	1998.253304	315.637486	772.080178	21.801498	26.410865	198.831498	199.182137	71.950667	0.792208	2.184303	0.23454

```
In [3]: vaild_cols = ['Wing', 'Weight', 'Culmen', 'Hallux', 'Tail', 'StandardTail', 'Tarsus', 'WingPitFat', 'KeelFat', 'Crop']
        def custom_fillna(col):
            if col.name in vaild_cols:
                col = col.fillna(round(col.mean(), 2))

            return col
        res = df.apply(custom_fillna)
```

```
In [4]: # move Class col to the end
        res = res[ [ col for col in res.columns if col != 'Species' ] + ['Species'] ]
```

```
In [5]: res.to_csv('hawks_fillna.csv', index=False)
```

```
In [6]: res
```

Out[6]:

	Month	Day	Year	CaptureTime	ReleaseTime	BandNumber	Age	Sex	Wing	Weight	Culmen	Hallux	Tail	StandardTail	Tarsus	WingPitFat	KeelFat
0	9	19	1992	13:30	NaN	877-76317	I	NaN	385.0	920.0	25.7	30.10	219	199.18	71.95	0.79	2.18
1	9	22	1992	10:30		877-76318	I	NaN	376.0	930.0	21.8	26.41	221	199.18	71.95	0.79	2.18

Tập dữ liệu được lưu dưới tên "hawks_fillna.csv", và nó sẽ là tên khởi đầu của các "Relation" (nằm trong phần tên tập tin đầu vào) của mỗi kết quả phân lớp của yêu cầu A, B, C và D.

2.2. Tiền xử lý trực tiếp bằng Weka

Điền dữ liệu bị thiếu bằng giá trị mean của các thuộc tính có missing values chưa xử lý ở phần Python: sử dụng chức năng "Filter" trong cửa số "Preprocess" của Explorer, chọn 'filters' → 'unsupervised' → 'attribute' → 'ReplaceMissingValues'.

Để chạy được các thuật toán phân lớp: NaiveBayesSimple, Id3, J48 thì mọi thuộc tính đều phải là kiểu dữ liệu nominal: sử dụng chức năng "Filter" trong cửa số "Preprocess" của Explorer, chọn 'filters' → 'unsupervised' → 'attribute' → 'NumericToNominal'.

Phần III: Đánh giá kết quả phân lớp

3.1. Phương pháp phân lớp nào thường cho kết quả cao nhất?

Dựa theo file kết quả "result.xls", ta có thể thấy được phương pháp NaiveBayesSimple cho ra kết quả cao hơn đối với cả 3 chiến lược đánh giá Use training set, Cross-validation (10 fold) và Percentage split (66%).

Lưu ý rằng phương pháp này chỉ đang cho ra kết quả tốt trên bộ dữ liệu cùng với các mẫu hiện tại, chứ chưa thể khẳng định nó luôn tốt nhất khi ta tiến hành phân lớp 1 bộ mẫu mới.

3.2. Phương pháp nào không thực hiện tốt và tại sao?

Dựa theo file kết quả "result.xls", ta có thể thấy được phương pháp Id3 cho ra kết quả chính xác thấp nhất (xấp xỉ 0%) đối với 2 chiến lược đánh giá Cross-validation (10 fold) và Percentage split (66%) nhưng với chiến lược đánh giá Use training set thì kết quả chính xác luôn luôn 100%. Chứng tỏ thuật toán Id3 bị overfitting trên tập dữ liệu này vì không thực hiện việc cắt tỉa. Đây là vấn đề của cây quyết định, nó chia nhỏ dữ liệu cho đến khi tạo thành các tập thuần túy (pure sets).

3.3. Tại sao ta sử dụng phiên bản đã rời rạc hóa của dữ liệu nếu dữ liệu đã được rời rạc hóa?

Rời rạc hoá dữ liệu như vậy cũng giúp giảm miền giá trị, từ đó giúp thuật toán phân lớp chạy nhanh hơn trong khi vẫn đảm bảo tính chính xác.

Ngoài ra, một số thuật toán đòi hỏi phải có các thuộc tính đã được rời rạc hóa thì mới cho phép gom cụm, cho nên việc rời rạc hóa này cũng được coi như là một phần của bước tiền xử lý dữ liệu.

3.4. Việc rời rạc hóa và cách rời rạc hóa có ảnh hưởng đến kết quả phân lớp hay không, nếu có thì ảnh hưởng thế nào?

Việc rời rạc hóa có ảnh hưởng đến kết quả phân lớp. Tùy thuộc vào cách thức rời rạc hóa mà kết quả phân lớp sẽ tốt hơn. Cụ thể, đối với phương pháp rời rạc bằng cách chia giỏ, thực nghiệm B chia thành 10 giỏ cho ra kết quả phân lớp tốt hơn so với chia thành 5 giỏ với độ sâu bằng nhau như thực nghiệm C. Với số giỏ phù hợp, ta có thể nhận được kết quả phân lớp tốt hơn. Tuy nhiên, ta có thể thấy việc rời rạc bằng cách chia giỏ đó cũng đã cho kết quả tốt hơn so với việc không chia giỏ (thực nghiệm A).

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

Chiến lược đánh giá use training set đã đánh giá quá cao độ chính xác bởi vì với phương pháp này, ta lấy hết toàn bộ dữ liệu cho tập huấn luyện để đánh giá. Như vậy, nhiễm nhiên kết quả đánh giá sẽ cao, nhưng nó chỉ phản ánh đúng cho bộ dữ liệu đó thôi. Trên thực tế, nếu đem mô hình đó đem ra để đánh giá các bộ test khác ở ngoài, kết quả thường sẽ không khả quan, thậm

chí độ chính xác đôi khi giảm xuống chỉ còn chưa tới 30%. Trường hợp này còn gọi là mô hình bị overfit.

3.6. Chiến lược nào đã đánh giá thấp (underestimate) độ chính xác và tại sao?

Chiến lược percentage split đã đánh giá thấp độ chính xác. Bản thân cách đánh giá này hoạt động chính là chia bộ dữ liệu huấn luyện ra làm 2 phần riêng biệt, một phần chỉ dùng để huấn luyện và một phần chỉ dùng để đánh giá. Điều đó khắc phục được điểm yếu của phương pháp use training set, tuy nhiên nó vẫn tồn tại điểm yếu là không thể tận dụng tối đa bộ dữ liệu để huấn luyện. Thông thường, dữ liệu cần thiết để huấn luyện cho một mô hình phân lớp là khá lớn trong khi ở bộ dữ liệu hawks chỉ có 908 mẫu huấn luyện, nhưng nó lại bị mất đi 33% để làm bộ kiểm tra. Như vậy, sẽ có một số mẫu lạ ở trong bộ kiểm tra chưa từng được huấn luyện để tìm ra đặc trưng, từ đó sẽ phân lớp sai, và cuối cùng dẫn đến độ chính xác của mô hình này sẽ thấp hơn các mẫu còn lại.

TÀI LIỆU THAM KHẢO